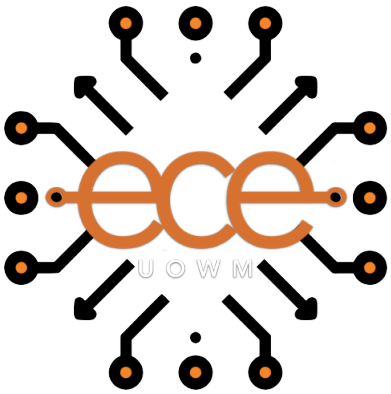


Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Μέθοδοι επιλογής χαρακτηριστικών για αλγόριθμους μηχανικής μάθησης

Αριστείδης Τσακνής (ΑΜ: 1744)

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

Εργαστήριο Ευφυών Συστημάτων & Βελτιστοποίησης

17 Ιουνίου 2024

Περίληψη

Στον τομέα της μηχανικής μάθησης και ειδικότερα στο κομμάτι της κατασκευής μοντέλων πρόβλεψης, η προεπεξεργασία των δεδομένων είναι ένα από τα κυριότερα βήματα που επηρεάζουν σημαντικά την απόδοση των μοντέλων. Ένα συχνό φαινόμενο στα σύγχρονα σύνολα δεδομένων είναι το τεράστιο μέγεθος, το οποίο τις περισσότερες φορές συνεπάγεται με περιττή και επαναλαμβανόμενη πληροφορία. Σημαντικό βήμα της προεπεξεργασίας είναι η αντιμετώπιση τέτοιων φαινομένων με σκοπό την καλύτερη προετοιμασία των δεδομένων για την εκπαίδευση των τελικών μοντέλων πρόβλεψης. Η αντιμετώπιση αυτών των φαινομένων επιτυγχάνεται μέσω των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Η συγκεκριμένη διπλωματική εργασία στοχεύει στην ανάλυση και σύγκριση διαφόρων τέτοιων μεθόδων και επικεντρώνεται στο πως τα μετασχηματισμένα δεδομένα που επιστρέφουν οι μέθοδοι επηρεάζουν την ακρίβεια διαφορετικών ταξινομητών. Η εργασία επεκτείνει επίσης πολλές από τις υπάρχουσες υλοποίησης των μεθόδων εισάγοντας τεχνικές που προσπαθούν να εντοπίσουν αυτόματα τον βέλτιστο αριθμό μείωσης διαστάσεων. Για την εξαγωγή των συμπερασμάτων και την αντικειμενική αξιολόγηση των αποτελεσμάτων των μεθόδων χρησιμοποιούνται επτά σύνολα δεδομένων με διαφορετικό αριθμό χαρακτηριστικών και ιδιοτήτων μεταξύ των στοιχείων. Τα αποτελέσματα δείχνουν ότι τόσο οι μέθοδοι μείωσης διαστάσεων όσο και πολλές από τις μεθόδους επιλογής χαρακτηριστικών τείνουν να διατηρούν ή ακόμα και να αυξάνουν την τελική απόδοση των τελικών μοντέλων, απλοποιώντας παράλληλα σε μεγάλο βαθμό την πολυπλοκότητα των συνόλων δεδομένων. Συμπερασματικά, η παρούσα εργασία υπογραμμίζει τη σημασία της προεπεξεργασίας των δεδομένων και ειδικότερα την απλοποίηση των περίπλοκων συνόλων δεδομένων που συγκροτούνται κυρίως σε σενάρια του πραγματικού κόσμου και προσφέρει καθοδήγηση για την επιλογή των καταλληλότερων μεθόδων με στόχο τη βελτίωση των αποτελεσμάτων των ταξινομητών.

Λέξεις κλειδιά: Python, Μηχανική μάθηση, Προεπεξεργασία, Μέθοδοι μείωσης διαστάσεων, Μέθοδοι εξαγωγής χαρακτηριστικών, Μέθοδοι επιλογής χαρακτηριστικών

Abstract

In the field of machine learning, particularly in the construction of predictive models, data preprocessing is one of the most critical steps that significantly affects model performance. A common phenomenon in modern datasets is their enormous size, which often entails redundant and repetitive information. Addressing such phenomena is a crucial step in preprocessing to better prepare the data for training the final predictive models. This is achieved through dimensionality reduction and feature selection methods. This thesis aims to analyze and compare various such methods, focusing on how the transformed data returned by these methods affect the accuracy of different classifiers. The work extends many of the existing implementations of these methods by introducing techniques that attempt to automatically identify the optimal number of dimensions for reduction. Seven datasets with different numbers of features and properties among the elements are used to draw conclusions and objectively evaluate the results of the methods. The results show that both dimensionality reduction methods and many feature selection methods tend to maintain or even increase the final performance of the models while significantly simplifying the complexity of the datasets. In conclusion, the research underscores the importance of data preprocessing, particularly the simplification of complex datasets that are primarily found in real-world scenarios, and offers guidance on selecting the most appropriate methods to improve classifier outcomes.

Keywords: Python, Machine learning, Preprocessing, Dimensionality reduction methods, Feature extraction methods, Feature selection methods

Δήλωση Πνευματικών Δικαιωμάτων

Δήλωση Πνευματικών Δικαιωμάτων Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο ''Μέθοδοι επιλογής χαρακτηριστικών για αλγόριθμους μηχανικής μάθησης'' καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Νικόλαου Πλόσκα αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δεν που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Υπογραφή Φοιτητή

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος αλγορίθμων

Κατάλογος πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός του προβλήματος

Στη σημερινή εποχή η ραγδαία ανάπτυξη της τεχνολογίας έχει φέρει τεράστιες μεταρρυθμίσεις στον τομέα των δεδομένων. Ειδικότερα, πλέον είμαστε σε θέση να συλλέγουμε και να αποθηκεύουμε τεράστιες ποσότητες δεδομένων από διάφορες πηγές, όπως μέσα κοινωνικής δικτύωσης, αισθητήρες, συναλλαγές και άλλα. Αυτή η έκρηξη δεδομένων έχει δημιουργήσει νέες ευκαιρίες και προκλήσεις για την εξαγωγή ουσιαστικών γνώσεων και τη λήψη τεκμηριωμένων αποφάσεων μέσα από αυτά. Η τεχνητή νοημοσύνη και η μηχανική μάθηση οδηγούν την καινοτομία σε πολλούς κλάδους τόσο της επιστήμης όσο και της επιχειρηματικότητας. Η αποτελεσματικότητα των μοντέλων μηχανικής μάθησης και τεχνητής νοημοσύνης εξαρτάται σε μεγάλο βαθμό από την ποιότητα των δεδομένων. Επομένως, το κομμάτι της διαχείρισης και ανάλυσης των δεδομένων είναι καίριας σημασίας. Τα σύνολα δεδομένων του πραγματικού κόσμου τις περισσότερες φορές αντιμετωπίζουν προβλήματα, όπως ο τεράστιος αριθμός χαρακτηριστικών, η επαναλαμβανόμενη πληροφορία, και ο τεράστιος όγκος δεδομένων. Η αξιοποίηση των δεδομένων σε αυτήν τη μορφή αυξάνει το υπολογιστικό κόστος σε τεράστιο βαθμό, κάνει το κομμάτι της εκπαίδευσης χρονοβόρο και το τελικό μοντέλο καταλήγει να είναι αναποτελεσματικό χωρίς ιδιαίτερη προγνωστική ικανότητα. Οι πιο γνωστές, ευέλικτες και πιθανότατα πιο αποτελεσματικές μέθοδοι για την αντιμετώπιση των παραπάνω προβλημάτων είναι οι μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών, οι οποίες αποτελούν απαραίτητα εργαλεία για τους επιστήμονες και αναλυτές δεδομένων. Οι συγκεκριμένες μέθοδοι συμβάλλουν στη μείωση του αριθμού των χαρακτηριστικών και κατ' επέ-

κταση στην απλοποίηση της πολυπλοκότητας του συνόλου δεδομένων, διατηρώντας παράλληλα τις πιο σημαντικές πληροφορίες σε αυτό. Το αποτέλεσμα της διαδικασία αυτής είναι μικρότερα σύνολα δεδομένων, τα οποία διατηρούν την κύρια πληροφορία των αρχικών δεδομένων, χωρίς να επιβαρύνονται από επαναλαμβανόμενη και περιττή πληροφορία ή θόρυβο. Παράλληλα, η εφαρμογή τέτοιων μεθόδων πολλές φορές οδηγεί τα μοντέλα μηχανικής μάθησης στην καλύτερη καταγραφή των εσωτερικών δομών και ιδιοτήτων των δεδομένων επιτυγχάνοντας έτσι υψηλότερη απόδοση.

1.2 Κίνητρα και στόχοι υλοποίησης

Η εργασία αποσκοπεί στην εις βάθος διερεύνηση των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών και τον αντίκτυπο εφαρμογής των συγκεκριμένων μεθόδων σαν βήμα προεπεξεργασίας κατά την ανάπτυξη μοντέλων μηχανικής μάθησης. Επιπλέον, σημαντικό κομμάτι της μελέτης αποτελεί και η συγκριτική αξιολόγηση των μεθόδων πάνω σε διάφορα σύνολα δεδομένων, με διαφορετικά χαρακτηριστικά και ιδιότητες το καθένα. Ειδικότερα, κρίνεται απαραίτητη η πειραματική αξιολόγηση των μεθόδων σε πραγματικά σύνολα δεδομένων, ο εντοπισμός των πιο αποτελεσματικών μεθόδων αν υπάρχουν καθώς και η αναγνώριση των μεθόδων που συνεισφέρουν περισσότερο στη βελτίωση της απόδοσης των τελικών μοντέλων μηχανικής μάθησης. Παράλληλα, αναλύοντας τους εσωτερικούς μηχανισμούς των μεθόδων η εργασία έχει σαν στόχο να αναπτυχθούν τεχνικές οι οποίες θα προσπαθήσουν να καθορίσουν αυτόματα τον βέλτιστο αριθμό διαστάσεων για ένα σύνολο δεδομένων, απλοποιώντας περαιτέρω τη διαδικασία προεπεξεργασίας δεδομένων.

Αναλυτικότερα, τα κίνητρα πίσω από αυτή την εργασία είναι:

1. **Βελτίωση της απόδοσης των τελικών μοντέλων πρόβλεψης**: Τα δεδομένα υψηλών διαστάσεων συχνά οδηγούν σε υπερπροσαρμογή και αυξημένο υπολογιστικό κόστος. Με την αποτελεσματική μείωση των διαστάσεων, ο στόχος είναι να βελτιωθεί η ακρίβεια και η αποτελεσματικότητα των τελικών μοντέλων.

2. **Απλοποίηση της προεπεξεργασίας**: Η ανάπτυξη αυτοματοποιημένων τεχνικών για τον προσδιορισμό του βέλτιστου αριθμού διαστάσεων θα απλοποιή-

σει το κομμάτι της προεπεξεργασίας, εξοικονομώντας χρόνο και υπολογιστική ισχύ.

3. **Συγκριτική Ανάλυση**: Η διεξαγωγή μιας συγκριτικής ανάλυσης μεταξύ όλων των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών θα προσφέρει χρήσιμες πληροφορίες για τα πλεονεκτήματα, τα μειονεκτήματα αλλά και τις δυνατότητες κάθε μεθόδου, καθοδηγώντας τη μελλοντική έρευνα και τις πρακτικές εφαρμογές.

4. **Πειραματική Ανάλυση**: Η αξιολόγηση αυτών των μεθόδων σε διαφορετικά σύνολα δεδομένων διασφαλίζει την εγκυρότητα των θεωρητικών ιδιοτήτων των μεθόδων και την επιβεβαίωση ή απόρριψη τους μέσω της πειραματικής διαδικασίας, αντιμετωπίζοντας πρακτικές προκλήσεις που συναντούν οι επιστήμονες του τομέα της ανάλυσης δεδομένων.

Οι στόχοι υλοποίησης αυτής της διπλωματικής εργασίας περιλαμβάνουν:

1. **Ανάπτυξη και βελτίωση αλγορίθμων**: Ένας από τους κύριους στόχους της εργασίας είναι η ανάπτυξη των διαφόρων μεθόδων στη γλώσσα προγραμματισμού Python και η βελτίωση των μεθόδων με σκοπό τον αυτόματο προσδιορισμό του βέλτιστου αριθμού διαστάσεων, βελτιώνοντας τη χρηστικότητα και την αποτελεσματικότητα των υπάρχουσων υλοποιήσεων.

2. **Πειραματική αξιολόγηση**: Διεξαγωγή αναλυτικών πειραμάτων σε πολλαπλά σύνολα δεδομένων και ταξινομητές για την πειραματική αξιολόγηση της απόδοσης των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών και επιβεβαίωση ή απόρριψη των θεωρητικών τους ιδιοτήτων.

Συνοψίζοντας, αυτή η εργασία στοχεύει να ενισχύσει τις υπάρχουσες γνώσεις και πρακτικές που υπάρχουν στο πεδίο της μηχανικής μάθησης και ειδικότερα στο κομμάτι της προεπεξεργασίας, παρέχοντας μια βαθύτερη κατανόηση των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών και συμβάλλοντας τελικά σε πιο αποτελεσματικές τεχνικές προεπεξεργασίας των δεδομένων.

1.3 Διάρθρωση κειμένου

Η παρούσα διπλωματική εργασία αποτελείται από πέντε επιπλέον κεφάλαια. Το δεύτερο κεφάλαιο με τίτλο θεωρητικό υπόβαθρο αποτελεί μια εκτεταμένη ανάλυση όλων των μεθόδων που χρησιμοποιήθηκαν στην εργασία. Ειδικότερα, γίνεται μια θεωρητική ανάλυση κάθε μεθόδου όπου αναλύεται θεωρητικά η προσέγγιση που ακολουθεί η κάθε μέθοδος. Ακολουθείται η μαθηματική ανάλυση κάθε μεθόδου όπου επεξηγούνται τα βήματα που ακολουθεί η μέθοδος για τη μείωση διαστάσεων ή την επιλογή των σημαντικότερων χαρακτηριστικών. Τέλος, δίνονται μερικά παραδείγματα από τις εφαρμογές τις εκάστοτε μεθόδου σε διάφορα επιστημονικά πεδία τονίζοντας τη συμβολή της μεθόδου στην επίτευξη του στόχου της εκάστοτε έρευνας.

Στο τρίτο κεφάλαιο βρίσκεται η βιβλιογραφική ανασκόπηση, όπου γίνεται μια πλήρης και μεθοδική ανάλυση της υπάρχουσας βιβλιογραφίας. Παραθέτονται και σχολιάζονται μελέτες που συγκρίνουν μεθόδους μείωσης διαστάσεων με κριτήριο αν είναι γραμμικές ή όχι, έρευνες που χρησιμοποιούν σαν κριτήριο αν η μέθοδος είναι εποπτευόμενη ή όχι και μελέτες που συγκρίνουν κλασσικές μεθόδους μείωσης διαστάσεων με μεθόδους που βασίζονται στα νευρωνικά δίκτυα. Παράλληλα, αναλύονται έρευνες οι οποίες επικεντρώνονται στις μεθόδους επιλογής χαρακτηριστικών και ειδικότερα πως αυτές επηρεάζουν την απόδοση των τελικών μοντέλων συγκριτικά με τις μεθόδους μείωσης διαστάσεων. Στο τελικό κομμάτι της βιβλιογραφίας γίνεται μια ανάλυση των υπαρχουσών ερευνών που σχετίζονται με τεχνικές για την εύρεση του βέλτιστου αριθμού μείωσης διαστάσεων για ένα σύνολο δεδομένων, τους περιορισμούς, τα ευρήματα και την αποτελεσματικότητα τους.

Το τέταρτο κεφάλαιο αφορά το κομμάτι των υλοποιήσεων. Στο συγκεκριμένο κεφάλαιο γίνεται η ανάλυση του κώδικα που αναπτύχθηκε σε Python. Για την καλύτερη κατανόηση και ανάλυση των εσωτερικών μηχανισμών κάθε μεθόδου αναπτύχθηκαν κώδικες για κάθε μέθοδο οι οποίες υλοποιούν στη γλώσσα Python τη θεωρητική προσέγγιση κάθε μεθόδου. Επιπλέον σε πολλές από τις μεθόδους προστέθηκαν συναρτήσεις οι οποίες επιτρέπουν την αυτόματη επιλογή ενός αριθμού διαστάσεων με στόχο την αυτόματη εύρεση του βέλτιστου αριθμού διαστάσεων. Κάθε κώδικας αποτέλεσε μια κλάση και όλοι οι κώδικες συγκρότησαν ένα πακέτο

Python. Στο συγκεκριμένο κεφάλαιο υπάρχουν οι ψευδοκώδικες για κάθε μέθοδο μαζί με μια αναλυτική επεξήγηση της κάθε υλοποίησης.

Στο πέμπτο κεφάλαιο πραγματοποιείται η υπολογιστική μελέτη. Στο συγκεκριμένο κεφάλαιο αναλύεται ο τρόπος και τα πειράματα που έγιναν για τη σύγκριση των μεθόδων. Επιπλέον, παραθέτονται τα αποτελέσματα των πειραμάτων, ποιες μέθοδοι είχαν τα καλύτερα αποτελέσματα και πως λειτούργησαν οι τεχνικές αυτόματης εύρεσης του αριθμού μείωσης διαστάσεων συγκριτικά με τα βέλτιστα αποτελέσματα.

Το έκτο κεφάλαιο είναι το κεφάλαιο των συμπερασμάτων. Εκεί γίνεται μια εκτενής ανάλυση και σχολιασμός των αποτελεσμάτων. Παράλληλα, γίνεται μια προσπάθεια εξήγησης των αιτιών που οδήγησαν σε αυτά τα αποτελέσματα. Τέλος, υπογραμμίζονται οι περιορισμοί της παρούσας εργασίας και προτείνονται ιδέες για μελλοντικές έρευνες οι οποίες θα ρίξουν παραπάνω φως στο συγκεκριμένο κομμάτι της επιστήμης των δεδομένων.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Μείωση διαστάσεων και επιλογή χαρακτηριστικών

Στον τομέα της ανάλυσης δεδομένων, η μείωση διαστάσεων (Dimensionality Reduction) και η επιλογή χαρακτηριστικών (Feature Selection) είναι δύο βασικές προσεγγίσεις για την αντιμετώπιση των προκλήσεων που τίθενται στα σύνολα δεδομένων που έχουν μεγάλο αριθμό χαρακτηριστικών με αποτέλεσμα η επεξεργασία τους να προϋποθέτει τεράστια υπολογιστική ισχύ. Σύνολα δεδομένων με πολυάριθμο πλήθος χαρακτηριστικών προκύπτουν σε διάφορους τομείς της επιστήμης συμπεριλαμβανομένης της Ψηφιακής Επεξεργασίας Εικόνας, Βιοπληροφορικής, Αστρονομίας, Υπολογιστικής Μοριακής Δυναμικής, Υπολογιστικής Δυναμικής Ρευστών και Φασματοσκοπίας [1] [2] [3] παρουσιάζοντας μοναδικές προκλήσεις στην επεξεργασία δεδομένων και την απόδοση των τελικών μοντέλων μηχανικής μάθησης που κατασκευάζονται από αυτά.

Η προσέγγιση της μείωσης διαστάσεων στοχεύει στη μείωση του αριθμού των μεταβλητών εισόδου ενός συνόλου δεδομένων. Αυτό επιτυγχάνεται μετασχηματίζοντας τα δεδομένα από έναν χώρο πολλών διαστάσεων σε έναν με λιγότερες διαστάσεις, διατηρώντας παράλληλα όσο το δυνατόν μεγαλύτερο ποσοστό της αρχικής πληροφορίας. Ο μετασχηματισμός αυτός είναι κύριας σημασίας για την απλοποίηση του τελικού μοντέλου, τη βελτίωση της ερμηνείας του συνόλου δεδομένων και τη μείωση των υπολογιστικών απαιτήσεων. Με τη συγκεκριμένη προσέγγιση αντιμετωπίζεται επίσης και η «κατάρα της διαστατικότητας» [4] όπου σύμφωνα με αυτήν η αποτελεσματικότητα των μοντέλων μηχανικής μάθησης μειώνεται με την αύξηση του αριθμού των διαστάσεων.

Η επιλογή χαρακτηριστικών, από την άλλη πλευρά, περιλαμβάνει την επιλογή ενός υποσυνόλου των πιο σχετικών χαρακτηριστικών για χρήση στην κατασκευή του τελικού μοντέλου. Σε αντίθεση με τη μείωση διαστάσεων, η οποία δημιουργεί νέους συνδυασμούς μεταβλητών, η επιλογή χαρακτηριστικών διατηρεί τις αρχικές μεταβλητές, επιλέγοντας αυτές που συμβάλλουν περισσότερο στην προγνωστική ικανότητα του μοντέλου [5]. Η συγκεκριμένη τεχνική όχι μόνο συμβάλλει στην αύξηση της ακρίβειας του μοντέλου, αλλά ταυτόχρονα μειώνει τον χρόνο εκπαίδευσης του μοντέλου και την υπερπροσαρμογή (Overfitting) [6].

Οι δύο παραπάνω προσεγγίσεις είναι καθοριστικής σημασίας για την καταπολέμηση της κατάρας της διαστατότητας, ενισχύοντας έτσι την ακρίβεια και την αποτελεσματικότητα του μοντέλου. Επιπλέον, οι χώροι πολλαπλών διαστάσεων είναι πιθανών να οδηγήσουν σε αραιές κατανομές δεδομένων, περιπλέκοντας την εξαγωγή σημαντικών μοτίβων από τα τελικά μοντέλα [4]. Παράλληλα, ο μεγάλος αριθμός διαστάσεων κλιμακώνει την υπολογιστική πολυπλοκότητα και τους πόρους που απαιτούνται για την ανάλυση των δεδομένων.

Οι μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών εφαρμόζονται σε πολλούς τομείς. Στη φασματοσκοπία χρησιμοποιούνται για την ανάλυση και επεξεργασία φασματικών δεδομένων τα οποία χαρακτηρίζονται από τον τεράστιο αριθμό διαστάσεων μιας και περιλαμβάνουν πληροφορίες από διαφορετικά μήκη κύματος ή συχνότητες. Στη βιοπληροφορική, χρησιμοποιούνται για την ανάλυση γενετικών δεδομένων, εντοπίζοντας τα γονίδια που σχετίζονται περισσότερο με ορισμένες ασθένειες. Στα χρηματοοικονομικά, βοηθούν στη διαχείριση κινδύνου και τον εντοπισμό απάτης απομονώνοντας τους βασικούς παράγοντες που επηρεάζουν τα οικονομικά αποτελέσματα [7]. Η ανάλυση κειμένου και η όραση υπολογιστή επωφελούνται επίσης από τη μείωση διαστάσεων και την επιλογή χαρακτηριστικών, επιτρέποντας την αποτελεσματική επεξεργασία μεγάλων συνόλων δεδομένων και βελτιώνοντας την απόδοση της επεξεργασίας φυσικής γλώσσας και των μοντέλων αναγνώρισης εικόνας [8].

Συνοψίζοντας και οι δύο προσεγγίσεις στοχεύουν στη μείωση του αριθμού των χαρακτηριστικών ενός συνόλου δεδομένων, η μείωση των διαστάσεων είναι προτιμότερη όταν οι νέοι συνδυασμοί χαρακτηριστικών αποτυπώνουν πιο αποτελεσματικά τα βασικά μοτίβα του συνόλου δεδομένων. Ενώ η επιλογή χαρακτηριστικών χρησι-

μοποιείται για να διατηρήσει το υποσύνολο των χαρακτηριστικών που προσφέρουν την καλύτερη ερμηνεία και περιέχουν το μεγαλύτερο μέρος της αρχικής πληροφορίας.

2.2 Principal Component Analysis

2.2.1 Θεωρητική ανάλυση

Η Ανάλυση Πρωταρχικών Συνιστωσών (Principal Component Analysis) είναι μια θεμελιώδης μέθοδος στην ανάλυση δεδομένων και στην προγνωστική μοντελοποίηση και έχει εφαρμοστεί εκτενώς σε διάφορους επιστημονικούς κλάδους για την ανάλυση και την ερμηνεία των δεδομένων. Είναι μια στατιστική διαδικασία που χρησιμοποιεί ορθογώνιο μετασχηματισμό για να μετατρέψει ένα σύνολο παρατηρήσεων, οι οποίες πιθανώς είναι μεταξύ συσχετιζόμενων μεταβλητών σε ένα σύνολο γραμμικά ασύνδετων μεταβλητών που ονομάζονται Πρωταρχικές Συνιστώσες (Principal Components) [9]. Η μείωση των διαστάσεων του αρχικού συνόλου δεδομένων και κατ' επέκταση η κατασκευή των πρωταρχικών συνιστωσών γίνεται με βάση τη διακύμανση. Ειδικότερα γίνεται μέσω της προβολής των αρχικών διαστάσεων σε έναν χώρο λιγότερων διαστάσεων ο οποίος καταγράφει το μεγαλύτερο μέρος της διακύμανσης τους [10].

2.2.2 Μαθηματική προσέγγιση

Τυποποίηση του συνόλου δεδομένων

Το αρχικό βήμα που ακολουθεί η Ανάλυση Πρωταρχικών Συνιστωσών (PCA) περιλαμβάνει την τυποποίηση του εύρους των συνεχών μεταβλητών για να διασφαλιστεί η ίση συμβολή των μεταβλητών στην ανάλυση. Το βήμα αυτό είναι αρκετά κρίσιμο για τη συγκεκριμένη μέθοδο καθώς έτσι αντιμετωπίζονται οι διαφορές στην κλίμακα διακύμανσης μεταξύ των μεταβλητών, οι οποίες διαφορετικά θα μπορούσαν να αλλοιώσουν τα αποτελέσματα της [9].

Υπολογισμός του πίνακα συνδιακύμανσης

Μετά την τυποποίηση, γίνεται ο υπολογισμός του πίνακα συνδιακύμανσης των δεδομένων, ένα βασικό βήμα που αποκαλύπτει τις συσχετίσεις μεταξύ των διαφόρων

μεταβλητών. Ο πίνακας συνδιακύμανσης, συμβολίζεται συνήθως ως C και υπολογίζεται από τον τυποποιημένο πίνακα δεδομένων X μέσω της εξίσωσης $C = X^T X$. Αυτός ο πίνακας αποτελεί τη βάση για την εξαγωγή των πρωταρχικών συνιστωσών [9].

Εύρεση ιδιοτιμών και ιδιοδιανυσμάτων

Το κύριο σημείο της ανάλυσης PCA έγκειται στον υπολογισμό των ιδιοτιμών λ και των ιδιοδιανυσμάτων v από τον πίνακα συνδιακύμανσης. Οι τιμές τους προκύπτουν με την επίλυση της εξίσωσης ιδιοτιμής $Cv = \lambda v$. Τα ιδιοδιανύσματα αντιπροσωπεύουν τις κατευθύνσεις των κύριων αξόνων στον χώρο των χαρακτηριστικών, υποδεικνύοντας έτσι τις κατευθύνσεις της μέγιστης διακύμανσης. Επιπρόσθετα, οι ιδιοτιμές ποσοτικοποιούν το μέγεθος αυτών των κατευθύνσεων, προσδιορίζοντας τη σημασία κάθε κύριου στοιχείου για την καταγραφή της μεταβλητότητας του συνόλου δεδομένων [9].

Επιλογή κύριων συνιστωσών

Η επιλογή των κύριων συνιστωσών πραγματοποιείται με βάση τις τιμές των ιδιοτιμών. Κάθε ιδιοτιμή σχετίζεται με ένα ιδιοδιάνυσμα. Τα ιδιοδιανύσματα που σχετίζονται με υψηλότερες ιδιοτιμές έχουν μεγαλύτερη προτεραιότητα. Αυτή η διαδικασία επιλογής περιλαμβάνει την ταξινόμηση των ιδιοδιανυσμάτων βάση των ιδιοτιμών και την επιλογή των κορυφαίων ιδιοδιανυσμάτων τα οποία θα σχηματίσουν τον τελικό πίνακα ιδιοδιανυσμάτων. Αυτός ο πίνακας στη συνέχεια χρησιμοποιείται για να γίνει η μετατροπή του αρχικού συνόλου δεδομένων στο νέο. Ο πίνακας πολλαπλασιάζεται με το σύνολο των αρχικών δεδομένων με αποτέλεσμα να κατασκευάζεται ο νέος χώρος μειωμένων διαστάσεων ο οποίος διατηρεί τις πιο ενημερωτικές πτυχές του αρχικού συνόλου δεδομένων [9].

2.2.3 Πεδία εφαρμογής

Ιατρική

Πέρα από το πεδίο της παραδοσιακής ανάλυσης δεδομένων και της μηχανικής μάθησης η Ανάλυση Πρωταρχικών Συνιστωσών έχει διεισδύσει σε διάφορους επιστημονικούς τομείς όπου τα δεδομένα υψηλών διαστάσεων θέτουν μοναδικές προ-

κλήσεις. Στην ιατρική, σε έρευνα σχετικά με την ανάλυση παραγόντων κινδύνου για ισχαιμικές καρδιακές παθήσεις (IHD) μέσω χημικών εξετάσεων αίματος [11], χρησιμοποιήθηκε η μέθοδος PCA. Ειδικότερα, η PCA αξιοποιήθηκε για τον εντοπισμό των συσχετίσεων και της συνδιακύμανσης μεταξύ των χαρακτηριστικών σε δεδομένα εξετάσεων αίματος για τον προσδιορισμό των παραγόντων που εξηγούν ένα σημαντικό μέρος της διακύμανσης των δεδομένων. Τα αποτελέσματα της μελέτης δείχνουν ότι η εφαρμογή της μεθόδου είχε θετικά αποτελέσματα, υπογραμμίζοντας τη χρησιμότητα της στη μείωση των διαστάσεων και στον εντοπισμό βασικών παραγόντων σε μεγάλα σύνολα δεδομένων, κάτι που είναι απαραίτητο για την κατανόηση πολύπλοκων ιατρικών καταστάσεων όπως η IHD.

Ανάλυση τροφίμων

Στην επιστήμη της χημείας και των τροφίμων, η PCA βοηθά στην ερμηνεία φασματοσκοπικών δεδομένων, επιτρέποντας στους επιστήμονες να προσδιορίζουν τις χημικές συνθέσεις και τις ιδιότητες του υλικού απλοποιώντας τα τεράστια σύνολα δεδομένων που λαμβάνονται από τα πειράματα. Σε έρευνα σχετικά με την ανίχνευση νοθείας του μελιού με χρήση φασματοσκοπίας [12] χρησιμοποιήθηκε η μέθοδος PCA στο κομμάτι της προεπεξεργασίας με στόχο τη μείωση των διαστάσεων των φασματικών δεδομένων. Η μέθοδος εντόπισε τα στοιχεία που περιείχαν τη μεγαλύτερη διακύμανση, υποδεικνύοντας το υποσύνολο των δεδομένων που πρέπει να επικεντρωθεί η έρευνα.

Διάγνωση σφαλμάτων

Η PCA χρησιμοποιείται σε μεγάλο βαθμό για τη διάγνωση σφαλμάτων σε βιομηχανικές διαδικασίες. Στην έρευνα των Ding et al. [13] γίνεται μια αξιολόγηση της μεθόδου στην ανίχνευση σφαλμάτων και προτείνονται τροποποιήσεις για τη βελτίωση της απόδοσης της κλασσικής μεθόδου. Η PCA είναι ιδιαίτερα αποτελεσματική στον συγκεκριμένο τομέα διότι προβάλλοντας τα δεδομένα διεργασίας σε άξονες που αντιπροσωπεύουν τις πιο σημαντικές παραλλαγές επιτρέπει τον ευκολότερο εντοπισμό ακραίων τιμών ή ανωμαλιών.

2.3 Singular Value Decomposition

2.3.1 Θεωρητική ανάλυση

Η μέθοδος Αποσύνθεσης Ιδιάζουσων τιμών (Singular Value Decomposition) αποτελεί μια από τις θεμελιώδεις μεθόδους στο πεδίο της γραμμικής άλγεβρας. Η μέθοδος παρέχει έναν ισχυρό μηχανισμό παραγοντοποίησης πινάκων που βρίσκει εκτεταμένη χρήση σε διάφορους τομείς, συμπεριλαμβανομένης της επεξεργασίας σήματος, των γραφικών υπολογιστών [14] και της μηχανικής μάθησης. Το βασικό στοιχείο της μεθόδου SVD είναι ότι έχει τη δυνατότητα να αποσυνθέτει οποιαδήποτε αριθμητικό πίνακα σε τρεις ξεχωριστούς πίνακες, αποκαλύπτοντας έτσι την εγγενή δομή και τα χαρακτηριστικά που μπορεί να μην είναι εμφανή στο αρχικό σύνολο δεδομένων.

2.3.2 Μαθηματική ανάλυση

Η μέθοδος Αποσύνθεσης Ιδιάζουσων τιμών αποσυνθέτει κάθε δεδομένο ενός πίνακα A με διαστάσεις $m \times n$ σε τρεις διακριτούς πίνακες U, Σ και V^T, όπου ο U έχει διαστάσεις $m \times m$, ο Σ είναι ένας διαγώνιος πίνακας $m \times n$ και ο V^T έχει διαστάσεις $n \times n$. Μεταξύ των πινάκων ισχύει η σχέση $A = U\Sigma V^T$ [15]. Αυτή η διάσπαση του αρχικού πίνακα αποκαλύπτει τις εγγενείς γεωμετρικές και αλγεβρικές ιδιότητες του A.

Μετασχηματισμός του αρχικού πίνακα δεδομένων

Το πρώτο βήμα της μεθόδου Singular Value Decomposition (SVD), περιλαμβάνει τον υπολογισμό $A^T A$, ο οποίος έχει σαν αποτέλεσμα έναν πίνακα μεγέθους $n \times n$ και του πίνακα AA^T ο οποίος έχει σαν αποτέλεσμα έναν πίνακα μεγέθους $m \times m$. Αυτό το βήμα είναι αρκετά σημαντικό καθώς μετατρέπει το αρχικό σύνολο δεδομένων σε δύο τετραγωνικούς πίνακες και προετοιμάζει το έδαφος για τον υπολογισμό των ιδιοτιμών και των ιδιοδιανυσμάτων, μέσω των οποίων γίνεται η κατασκευή των πινάκων V και Σ.

Υπολογισμός δεξιών ιδιάζουσων ιδιοδιανυσμάτων και ιδιοτιμών, Πίνακες V και Σ

Το επόμενο βήμα είναι ο υπολογισμός των ιδιοδιανυσμάτων και ιδιοτιμών του πίνακα A^TA. Τα ιδιοδιανύσματα του πίνακα A^TA σχηματίζουν τις στήλες του V, και οι τετραγωνικές ρίζες των ιδιοτιμών είναι οι ιδιάζουσες τιμές σ_i, τοποθετημένες κατά μήκος της διαγωνίου του πίνακα Σ. Αυτή η διαδικασία είναι κρίσιμη για την κατανόηση των δομικών και γεωμετρικών ιδιοτήτων που κωδικοποιούνται στον αρχικό πίνακα A.

Υπολογισμός αριστερών Ιδιάζουσων ιδιοδιανυσμάτων, Πίνακας U

Στη συνέχεια πραγματοποιείται ο υπολογισμός του πίνακα U. Ο πίνακα U εξάγεται από τα ιδιοδιανύσματα του πίνακα AA^T. Τα συγκεκριμένα ιδιοδιανύσματα σχηματίζουν τις στήλες του U. Οι ιδιοτιμές τόσο του πίνακα AA^T όσο και του A^TA είναι κοινές επομένως ο πίνακας Σ μπορεί να εξαχθεί σε οποιοδήποτε από τα δύο βήματα.

Συναρμολόγηση του πίνακα ιδιάζουσων τιμών

Με τα U, Σ, και V καθορισμένα, υπάρχει η δυνατότητα έκφρασης του πίνακα A ως το γινόμενο αυτών των πινάκων: $A = U\Sigma V^T$. Οι πίνακες U και V αντιπροσωπεύουν περιστροφές ή ανακλάσεις του αρχικού πίνακα. Αυτή η ιδιότητα είναι κρίσιμη για τη σταθερότητα και την ερμηνεία της αποσύνθεσης. Ο διαγώνιος πίνακας Σ που περιέχει τις ιδιοτιμές παρέχει έναν συντελεστή κλίμακας για καθένα από τα ορθογώνια στοιχεία που προσδιορίζονται στους πίνακες U και V και οι τιμές του υποδεικνύουν τη σημασία ή το «βάρος» κάθε στοιχείου στην ανακατασκευή του αρχικού πίνακα A.

Επιλογή κύριων ιδιάζουσων τιμών

Στο τελικό βήμα, πραγματοποιείται η ταξινόμηση των σ_i του πίνακα Σ σε φθίνουσα σειρά. Παράλληλα ταξινομούνται και τα ιδιοδανύσματα των πινάκων U και V σύμφωνα με την ταξινόμηση του πίνακα Σ. Επιλέγονται οι k μεγαλύτερες ιδιάζουσες τιμές, μέσω των οποίων προκύπτουν οι πίνακες U_k, Σ_k, και V_k^T. Για το νέο σύνολο δεδομένων με μειωμένο αριθμό διαστάσεων ισχύει $A' = U_k\Sigma_kV_k^T$.

Ερμηνεία αποτελεσμάτων

Η διαδικασία μετασχηματισμού SVD επεκτείνει τη χρησιμότητα της σε διάφορες εφαρμογές, μεταξύ άλλον και στη μείωση διαστάσεων. Μοιάζει αρκετά με τη μέθοδο της Ανάλυσης Πρωταρχικών Συνιστωσών με κύρια διαφορά ότι εφαρμόζεται απευθείας στον πίνακα δεδομένων A και δεν απαιτεί τον υπολογισμό του πίνακα συνδιακύμανσης.

2.3.3 Πεδία εφαρμογής

Συμπίεση δεδομένων

Η ικανότητα της μεθόδου Αποσύνθεσης Ιδιάζουσων τιμών να αποσυνθέτει πίνακες σε απλούστερα και ερμηνεύσιμα στοιχεία την καθιστά ανεκτίμητο εργαλείο τόσο για θεωρητικές όσο και για πρακτικές εφαρμογές. Στον τομέα της ανάλυσης δεδομένων και ειδικότερα στη διαδικασία συμπίεσης εικόνων η μέθοδος SVD είναι ιδιαίτερα αποτελεσματική. Σε σχετική έρευνα [16] έγινε αξιολόγηση της αποτελεσματικότητας της μεθόδου στη συμπίεση εικόνων, καθώς επίσης και η πρόταση τροποποίησης της μεθόδου για την αυτόματη εξαγωγή των K πρώτων ιδιάζουσων τιμών που διατηρούν μια ισορροπία μεταξύ του λόγου συμπίεσης και της ποιότητας της εικόνας. Η έρευνα καταλήγει στο συμπέρασμα, ότι η μέθοδος SVD είναι μια αποτελεσματική μέθοδος για τη συμπίεση εικόνων.

Βιοπληροφορική

Εξίσου σημαντική συνεισφορά παρέχει και στον τομέα της βιοπληροφορικής όπου συνεισφέρει στην ανάλυση πινάκων γονιδιακής έκφρασης, βελτιώνοντας έτσι την κατανόηση των λειτουργιών των γονιδίων, τις αλληλεπιδράσεις τους και τη συνολική αρχιτεκτονική των γενετικών δεδομένων. Στην έρευνα [17], η μέθοδος SVD χρησιμοποιήθηκε συνδυαστικά με μια άλλη γνωστή μέθοδο την Ανεξάρτητη Ανάλυση Στοιχείων (Independent Component Analysis) για τη βελτίωση των δεδομένων σε βιομοριακές βάσεις δεδομένων. Ειδικότερα, η SVD εφαρμόστηκε ως βήμα μείωσης των διαστάσεων των δεδομένων πριν από την εφαρμογή της ICA, καθιστώντας το μοντέλο πιο ανθεκτικό έναντι των ακραίων τιμών και μειώνοντας την υπερπροσαρμογή. Το τελικό μοντέλο πρόβλεψης σύμφωνα με το άρθρο παρουσίασε βελτίωση

στην ακρίβεια πρόβλεψης, υπογραμμίζοντας την αποτελεσματικότητα των δύο μεθόδων.

Διάγνωση σφαλμάτων σε περιστρεφόμενα μηχανήματα

Η εξαγωγή των ιδιάζουσων τιμών μέσω της μεθόδου SVD έχει τη δυνατότητα αποτύπωσης της υποκείμενης δομής των δεδομένων, αποκαλύπτοντας μοτίβα και σχέσεις μεταξύ των δεδομένων που υπό άλλες συνθήκες δεν θα ήταν διακριτά. Σε σχετική έρευνα [18] που επικεντρώνεται στη διάγνωση σφαλμάτων σε περιστρεφόμενα μηχανήματα η μέθοδος SVD έπαιξε σημαντικό ρόλο. Ειδικότερα η εφαρμογή της μεθόδου SVD σαν βήμα προεπεξεργασίας, οδήγησε σε βελτίωση των αποτελεσμάτων του τελικού μοντέλου ταξινομητή τονίζοντας, την αποτελεσματικότητα της στη διάγνωση διαφόρων βλαβών.

2.4 Μέθοδοι εκμάθησης συνόλου ως μηχανισμοί επιλογής χαρακτηριστικών

2.4.1 Θεωρητική ανάλυση

Οι μέθοδοι εκμάθησης συνόλου (ensemble learning) αποτελούν έναν ευρύτερο τομέα της μηχανικής μάθησης στον οποίο χρησιμοποιούνται πολλά ομοειδή μοντέλα (συχνά αποκαλούμενα «αδύναμοι μαθητές») για να λύσουν το ίδιο πρόβλημα και συνδυάζονται τα αποτελέσματα τους για να εξαχθούν οι τελικές προβλέψεις. Η βασική ιδέα πίσω από αυτές τις μεθόδους είναι ότι μια ομάδα «αδύναμων μαθητών» μπορεί να συγχωνευθεί για να δημιουργήσει έναν «ισχυρό μαθητή» ο οποίος θα έχει μεγαλύτερη αποτελεσματικότητα. Η ισχύς αυτών των μεθόδων αποδίδεται στην ικανότητα τους να κατανοούν περίπλοκες σχέσεις μεταξύ εισόδων και εξόδων και να μειώνουν την υπερπροσαρμογή. Τεχνικές όπως ο Random Forest, ο Gradient Boosting, ο LightGBM και ο CatBoost είναι μερικές από τις πιο γνωστές μεθόδους αυτής της κατηγορίας.

Ένα από τα σημαντικά πλεονεκτήματα των μεθόδων εκμάθησης συνόλου, ειδικά των μεθόδων που βασίζονται σε δέντρα, είναι η εγγενής τους ικανότητα επιλογής χαρακτηριστικών. Αυτό επιτυγχάνεται μέσω της ικανότητας τους να υπολογίζουν τη σημαντικότητα κάθε χαρακτηριστικού κατά την περίοδο εκπαίδευσης τους. Ειδι-

κότερα οι έμφυτες διαδικασίες και μετρικές που χρησιμοποιούν για την κατασκευή των δέντρων, μπορούν εύκολα να χρησιμοποιηθούν και για την επιλογή των πιο συναφών χαρακτηριστικών σε ένα σύνολο δεδομένων. Αυτό έχει ως αποτέλεσμα μέσω των συγκεκριμένων μεθόδων να πραγματοποιείται μια αξιολόγηση των χαρακτηριστικών που αποτελούν το σύνολο δεδομένων και να προσδιορίζονται τα πιο σχετικά χαρακτηριστικά που συμβάλλουν περισσότερο στην προγνωστική ισχύ του μοντέλου, επιτρέποντας έτσι τη μείωση των συνολικών χαρακτηριστικών χωρίς σημαντική απώλεια πληροφοριών.

2.4.2 Ensemble learning τεχνικές βασιζόμενες σε δέντρα

Μια από τις πιο δημοφιλής κατηγορίες εκμάθησης συνόλου είναι οι τεχνικές που βασίζονται σε δέντρα. Η πιο γνωστή μέθοδος αυτής της κατηγορίας είναι ο Random Forest. Ο Random Forest αποτελείται από πολλά μεμονωμένα δέντρα αποφάσεων, καθένα από τα οποία εκπαιδεύεται σε ένα τυχαίο υποσύνολο του συνόλου εκπαίδευσης. Κάθε δέντρο στη συνέχεια χρησιμοποιείται για να δώσει μια πρόβλεψη, κάθε πρόβλεψη προσμετράται και η πρόβλεψη με τις περισσότερες ψήφους γίνεται η τελική πρόβλεψη του μοντέλου.

Μια παρόμοια μέθοδος, με διαφορετική προσέγγιση είναι ο αλγόριθμος Gradient Boosting. Ο συγκεκριμένος αλγόριθμος δημιουργεί ένα σύνολο δέντρων αποφάσεων αλλά σε αντίθεση με τον Random Forest τα δέντρα χρησιμοποιούνται διαδοχικά. Κάθε δέντρο διορθώνει τα λάθη που έγιναν από τα προηγούμενα. Η σημαντικότητα των χαρακτηριστικών προκύπτει από το πόσο συμβάλλει κάθε χαρακτηριστικό στη βελτίωση της απόδοσης του μοντέλου.

Άλλοι τρεις αρκετά γνωστοί αλγόριθμοι που κατατάσσονται στις μεθόδους εκμάθησης συνόλου είναι οι XGBoost, LightGBM και CatBoost. Πρόκειται για αναβαθμισμένες εκδόσεις της μεθόδου gradient boosting, γνωστών για τους ταχείς ρυθμούς εκπαίδευσης και την αυξημένη αποτελεσματικότητά τους. Αξιοποιούν πολύπλοκους αλγόριθμους για να εξακριβώσουν τη σημαντικότητα των χαρακτηριστικών, καθιερώνοντας τους ως ισχυρά εργαλεία στον τομέα της επιλογής χαρακτηριστικών.

Τέλος, υπάρχει και ο αλγόριθμος AdaBoost ο οποίος κατασκευάζει και συνδυάζει δέντρα απόφασης ενός επιπέδου για να δημιουργήσει έναν ισχυρό ταξινομητή. Ειδικότερα, κατά τη λειτουργία του προσπαθεί να προσαρμόσει τα βάρη των

εσφαλμένα ταξινομημένων περιπτώσεων, έτσι ώστε οι επόμενοι αδύναμοι μαθητές να επικεντρωθούν περισσότερο σε αυτές τις περιπτώσεις που είναι δύσκολο να ταξινομηθούν.

2.4.3 Πεδία εφαρμογής

Οι μέθοδοι εκμάθησης συνόλου έχουν εφαρμογές σε πολλά πεδία και τομείς της επιστήμης, επιδεικνύοντας την προσαρμοστικότητα και την αποτελεσματικότητα τους στην επιλογή χαρακτηριστικών. Ακολουθούν ορισμένες πρακτικές εφαρμογές στις οποίες αξιοποιήθηκαν μέθοδοι εκμάθησης συνόλου:

Ιατρική

Μέθοδοι εκμάθησης συνόλου για την επιλογή χαρακτηριστικών χρησιμοποιήθηκαν στον τομέα της ιατρικής, για τον εντοπισμό βιοδεικτών για τη νόσο του Alzheimer σε κλινικά δεδομένα. Αυτή η προσέγγιση βοήθησε στην αποκάλυψη πιθανών πρώιμων δεικτών της νόσου, αναδεικνύοντας την ικανότητα των μεθόδων να χειρίζονται πολύπλοκα, πολλών διαστάσεων δεδομένα και να εξάγουν τα σημαντικότερα χαρακτηριστικά [19].

Μέθοδοι εκμάθησης συνόλου έχουν χρησιμοποιηθεί για την ενίσχυση της απόδοσης μοντέλων διάγνωσης του καρκίνου του προστάτη. Τα σύνολα δεδομένων έκφρασης γονιδίων μικροσυστοιχιών, έχουν τεράστιο αριθμό χαρακτηριστικών και αποτελούν σημαντική πρόκληση για πολλούς αλγόριθμους μηχανικής μάθησης για τον εντοπισμό των μοτίβων και σχέσεων μεταξύ των δεδομένων. Με την εξάλειψη των άσχετων χαρακτηριστικών μέσω των μεθόδων εκμάθησης συνόλου, τα μοντέλα πρόβλεψης επικεντρώνονται στα σημαντικά χαρακτηριστικά, δημιουργώντας ένα ισχυρό πλαίσιο για την ενίσχυση της απόδοσης του μοντέλου. Αυτή η προσέγγιση όχι μόνο αυξάνει την προγνωστική ακρίβεια, αλλά βοηθά επίσης στην απόκτηση βαθύτερης κατανόησης των γενετικών δεικτών που συνδέονται με τον καρκίνο του προστάτη [20].

Ταξινόμηση κειμένων

Στον τομέα της ταξινόμησης κειμένων (text classification), κατά την ανάπτυξη μοντέλων ταξινόμησης κειμένων, έχουν εφαρμοστεί μέθοδοι εκμάθησης συνόλου για

τον εντοπισμό λέξεων-κλειδιών και φράσεων. Η χρήση αυτών των μεθόδων βοήθησε στον εντοπισμό λέξεων-κλειδιά και φράσεων αυξάνοντας έτσι την ικανότητα των μοντέλων να ταξινομούν τα κείμενα με ακρίβεια [21] [22].

2.5 Συντελεστής συσχέτισης Κατάταξης Spearman

2.5.1 Θεωρητική ανάλυση

Ο συντελεστής συσχέτισης κατάταξης του Spearman είναι μια μη παραμετρική μετρική της συσχέτισης που αξιολογεί πόσο καλά μπορεί να περιγραφεί η σχέση μεταξύ δύο μεταβλητών αξιολογώντας τον βαθμό στον οποίο η σχέση τους μπορεί να απεικονιστεί μέσω μιας μονοτονικής συνάρτησης. Σε αντίθεση με τη συσχέτιση Pearson, η οποία αξιολογεί μόνο γραμμικές σχέσεις μεταξύ συνεχών μεταβλητών, η συσχέτιση του Spearman έχει σχεδιαστεί για να προσδιορίζει τόσο μονοτονικές σχέσεις, όσο γραμμικές και μη γραμμικές. Αυτό την καθιστά ιδιαίτερα χρήσιμη για την ανάλυση μεταβλητών που δεν έχουν απαραίτητα κανονική κατανομή ή όταν η σχέση μεταξύ των μεταβλητών δεν αναμένεται να είναι γραμμική [23].

Η συσχέτιση κατάταξης Spearman παίζει καθοριστικό ρόλο στο κομμάτι της επιλογής χαρακτηριστικών, επιτρέποντας την ανίχνευση τόσο γραμμικών όσο και μη γραμμικών σχέσεων μεταξύ των χαρακτηριστικών. Αυτή η ικανότητα είναι ιδιαίτερα σημαντική στον τομέα της ανάλυσης δεδομένων γιατί επιτρέπει στους αναλυτές να εντοπίσουν και να κατανοήσουν τις σχέσεις που έχει ένα χαρακτηριστικό τόσο με το χαρακτηριστικό κλάσης, όσο και με τα υπόλοιπα χαρακτηριστικά του συνόλου δεδομένων. Ενα κύριο πλεονέκτημα της μεθόδου είναι ότι εστιάζει στις τάξεις των τιμών των χαρακτηριστικών και όχι στις ακατέργαστες τιμές τους. Επιπλέον μπορεί να αποκαλύψει κρυφά μοτίβα στα δεδομένα που πιθανών να είχαν παραληφθεί από άλλες γραμμικές μεθόδους, βοηθώντας έτσι στην επιλογή των πιο σχετικών χαρακτηριστικών για την εκπαίδευση του τελικού μοντέλου. Αυτή η διαδικασία όχι μόνο ενισχύει την ερμηνευτικότητα του μοντέλου εστιάζοντας σε ένα μικρότερο σύνολο σημαντικών χαρακτηριστικών, αλλά επίσης βελτιώνει την απόδοση του μοντέλου μειώνοντας τον κίνδυνο υπερπροσαρμογής και την υπολογιστική πολυπλοκότητα.

2.5.2 Μαθηματική ανάλυση

Πριν τον ορισμό του συντελεστή συσχέτισης κατάταξης Spearman είναι απαραίτητο να γίνει ο ορισμός της κατάταξης (rank). Στη στατιστική, με τον όρο κατάταξη ορίζεται η αριθμητική τιμή που παίρνει ένα στοιχείο σε σχέση με την κατάταξη του, στο ταξινομημένο σύνολο του. Για τον υπολογισμό των ranks ενός χαρακτηριστικού απαιτείται η ταξινόμηση των στοιχείων σε αύξουσα σειρά, ο ορισμός των τιμών κατάταξης για κάθε στοιχείο με μικρότερο αριθμό κατάταξης το ένα, το οποίο ορίζεται σαν τιμή κατάταξης του πρώτου στοιχείου του ταξινομημένου συνόλου. Στην περίπτωση ισοπαλίας σαν τιμή κατάταξης των κοινών στοιχείων ορίζεται ο μέσος όρος τους.

Ο γενικός μαθηματικός ορισμός του συντελεστή κατάταξης Spearman δίνεται ως ο συντελεστής συσχέτισης Pearson μεταξύ των τιμών κατάταξης, $R(X)$ και $R(Y)$, των δύο μεταβλητών X και Y:

$$\rho = \frac{\sum_i (R(X_i) - \overline{R(X)})(R(Y_i) - \overline{R(Y)})}{\sqrt{\sum_i (R(X_i) - \overline{R(X)})^2 \sum_i (R(Y_i) - \overline{R(Y)})^2}}$$

Ο συντελεστής ρ έχει τιμές στο πεδίο $-1 \leq \rho \leq 1$. Ο βαθμός συσχέτισης μεταξύ δύο χαρακτηριστικών καθορίζεται από την απόλυτη τιμή του ρ. Όσο μεγαλύτερη είναι τόσο ισχυρότερη είναι και η συσχέτιση. Όταν η τιμή ρ είναι θετική, σημαίνει ότι μια αύξηση σε μια μεταβλητή αντιστοιχεί σε αύξηση και στην άλλη. Αντίθετα, μια αρνητική τιμή υποδηλώνει ότι μια αύξηση σε μια μεταβλητή συνδέεται με μείωση στην άλλη, υπογραμμίζοντας μια αντίστροφη σχέση μεταξύ τους [23].

2.5.3 Πεδία εφαρμογής

Διερευνητική ανάλυση δεδομένων

Ο συντελεστής συσχέτισης κατάταξης του Spearman, χρησιμοποιείται ευρέως σε πολλούς τομείς της επιστήμης, σε ζητήματα που αφορούν την καταγραφή της συσχέτισης μεταξύ δεδομένων. Ένα πεδίο εφαρμογής της μεθόδου είναι η διερευνητική ανάλυση δεδομένων που περιλαμβάνει μεγάλα σύνολα δεδομένων. Στην έρευνα των Xiao et al. [24] γίνεται μια συγκριτική ανάλυση των συντελεστών συσχέτισης Pearson, Spearman και Kendall. Στην έρευνα εξετάζεται η σχέση μεταξύ της κατά-

στασης λειτουργίας μιας αντλίας και διαφόρων καταγεγραμμένων μεταβλητών από τα δεδομένα κραδασμών της. Στα ευρήματα της έρευνας υπογραμμίζεται η χρησιμότητα των συντελεστών συσχέτισης για την εξαγωγή των ισχυρότερων συσχετίσεων μεταξύ των δεδομένων, καθώς επίσης τονίζεται και η πρακτική χρησιμότητα των μη παραμετρικών στατιστικών μετρήσεων όπως ο συντελεστής συσχέτισης κατάταξης Spearman στον χειρισμό πολύπλοκων και μεγάλων συνόλων δεδομένων.

Πρότυπα ανάμειξης σε πολύπλοκα δίκτυα

Στην έρευνα των Zhang et al. [25] γίνεται μια ανάλυση του συντελεστή συσχέτισης Spearman ως εργαλείο για τη μέτρηση της παρεκτικής και δυσπαρεκτικής ανάμειξης σε σύνθετα δίκτυα. Επιπλέον γίνεται σύγκριση της αποτελεσματικότητας με τον συντελεστή συσχέτισης Pearson. Η σύγκριση έγινε τόσο σε εμπειρικά όσο και σε τεχνητά δίκτυα. Τα αποτελέσματα της έρευνας καταλήγουν στο ότι ο συντελεστής συσχέτισης κατάταξης Spearman προσφέρει σημαντική βελτίωση σε σχέση με τις παραδοσιακές μεθόδους για τη μέτρηση της ανάμειξης σε πολύπλοκα δίκτυα. Τέλος, τονίζεται ότι είναι ιδιαίτερα αποτελεσματικός σε μεγάλα δίκτυα ή δίκτυα χωρίς κλίμακα, όπου άλλα μέτρα συσχέτισης ενδέχεται να αποτύχουν.

2.6 Συντελεστής συσχέτισης κατάταξης Kendall

2.6.1 Θεωρητική ανάλυση

Ο συντελεστής συσχέτισης κατάταξης Kendall είναι επίσης μια μη παραμετρική μετρική της συσχέτισης που αξιολογεί την ισχύ και την κατεύθυνση της συσχέτισης μεταξύ δύο μεταβλητών. Υπολογίζεται με βάση τον αριθμό των σύμφωνων ζευγών (concordant pairs) και των ασύμφωνων ζευγών (discordant pairs) προς τον αριθμό όλων των ζευγών δειγμάτων. Τα σύμφωνα ζεύγη είναι εκείνα όπου οι τάξεις και για τα δύο στοιχεία συμφωνούν με τη σειρά τους, ενώ τα ασύμφωνα ζεύγη έχουν τάξεις που διαφωνούν. Αυτή η μέθοδος είναι ιδιαίτερα αποτελεσματική για δεδομένα που δεν τηρούν τις προϋποθέσεις κανονικής κατανομής ή όταν το μέγεθος του δείγματος είναι μικρό [23].

2.6.2 Μαθηματική ανάλυση

Ο μαθηματικός τύπος που προσδιορίζει τον συντελεστή συσχέτισης κατάταξης Kendall μεταξύ στοιχείων που δεν έχουν ισόπαλες τιμές κατάταξης είναι :

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{1}{2}n(n-1)}$$

Στην περίπτωση που υπάρχουν ισοπαλίες ο τύπος του Kendall (τ_b) μετατρέπεται ως εξής:

$$\tau_b = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}$$

οπου:

- n_c είναι ο αριθμός των σύμφωνων ζευγών,
- n_d είναι ο αριθμός των ασύμφωνων ζευγών,
- $n_0 = n(n-1)/2$ είναι ο συνολικός αριθμός των ζευγών,
- $n_1 = \sum t_i(t_i - 1)/2$ ο συνολικός αριθμός ισοπαλιών στην πρώτη μεταβλητή,
- $n_2 = \sum u_i(u_i - 1)/2$ ο συνολικός αριθμός ισοπαλιών στη δεύτερη μεταβλητή,

Όπως και ο συντελεστής συσχέτισης κατάταξης του Spearman, ο συντελεστής συσχέτισης κατάταξης του Kendall κυμαίνεται μεταξύ των τιμών -1 έως 1, όπου οι τιμές κοντά στο 1 ή -1 υποδεικνύουν ισχυρή θετική ή αρνητική συσχέτιση και οι τιμές γύρω από το 0 δεν δείχνουν καμία συσχέτιση [23].

2.6.3 Πεδία εφαρμογής

Συσχέτιση των ειδών

Ο συντελεστής συσχέτισης κατάταξης Kendall, έχει παρόμοια λειτουργία με τον συντελεστή συσχέτισης κατάταξης Spearman, έχει εφαρμογή σε πολλά πεδία της επιστήμης, κυρίως, σε ζητήματα που αφορούν την καταγραφή της συσχέτισης μεταξύ δεδομένων. Ένα από αυτά τα πεδία είναι και η συσχέτιση και κατανομή των ειδών σε ομάδες. Ειδικότερα, στην έρευνα του LEGENDRE [26] για τον προσδιορισμό σημαντικά συνδεδεμένων ομάδων ειδών, χρησιμοποιήθηκε ο συντελεστής

συσχέτισης κατάταξης Kendall για την καταγραφή της συσχέτισης των δεδομένων και την αποκάλυψη συσχετίσεων μεταξύ των ειδών. Στην έρευνα υπογραμμίζεται η χρησιμότητα του και η αποτελεσματικότητα του στην αποκάλυψη των συσχετίσεων. Κάτι τέτοιο δείχνει ότι αποτελεί ένα σημαντικό εργαλείο για την ανάλυση δεδομένων.

Αξιολόγηση της δυστονίας

Στην έρευνα των Comella et al. [27] ο συντελεστής συσχέτισης κατάταξης Kendall έπαιξε σημαντικό ρόλο στην εξαγωγή των αποτελεσμάτων και την κατανόηση τους. Η έρευνα επικεντρώνεται στην αξιολόγηση της δυστονίας χρησιμοποιώντας τρεις κλίμακες αξιολόγησης, την κλίμακα Fahn-Marsden (F-M), την ενοποιημένη κλίμακα αξιολόγηση δυστονίας (UDRS) και τη παγκόσμια κλίμακα αξιολόγηση δυστονίας (GDS). Ο συντελεστής συσχέτισης κατάταξης Kendall χρησιμοποιήθηκε για την αξιολόγηση της συνέπειας ή της συμφωνίας μεταξύ των αξιολογητών. Συντέλεσε σημαντικά στην εξαγωγή των συμπερασμάτων καθώς εφαρμόζοντας τον, οι ερευνητές μπόρεσαν να εντοπίσουν ποιες συγκεκριμένες περιοχές του σώματος είχαν χαμηλότερη ή μεγαλύτερη συμφωνία μεταξύ των αξιολογητών.

2.7 Multidimensional Scaling

2.7.1 Θεωρητική ανάλυση

Η Πολυδιάστατη Κλιμάκωση είναι μια από τις πιο γνωστές και θεμελιώδεις μεθόδους μείωσης διαστάσεων. Μπορεί να θεωρηθεί και ως μια οικογένεια μεθόδων διότι πολλές τεχνικές βασίζονται πάνω σε αυτήν με μικρές διαφοροποιήσεις. Χαρακτηρίζεται ως μια στατιστική τεχνική η οποία στοχεύει να απεικονίσει την ομοιότητα ή την ανομοιότητα μεταξύ των διαφόρων στοιχείων ενός συνόλου δεδομένων μέσω της αναπαράστασης σε ένα χώρο λιγότερων διαστάσεων. Το κύριο στοιχείο της μεθόδου βάση του οποίου πραγματοποιείται η μείωση των διαστάσεων είναι η διατήρηση των αποστάσεων ανά ζεύγη μεταξύ των στοιχείων που βρίσκονται πλησιέστερα κατά τη χαρτογράφηση. Έτσι, μεταφέρονται όσο το δυνατόν πιο ακριβή οι αποστάσεις των σημείων, διατηρώντας κατά τη μείωση των διαστάσεων όσο το δυνατόν περισσότερο τη μορφολογία του συνόλου δεδομένων [28].

Τρεις είναι οι πιο γνωστές κατηγορίες MDS. Η κλασική μέθοδος MDS γνωστή και ως Principal Coordinates Analysis (PCoA), που εστιάζει στις ανομοιότητες μεταξύ των στοιχείων. Ειδικότερα οι ανομοιότητες εκφράζονται σαν πίνακας αποστάσεων που αντικατοπτρίζουν τις ευκλείδειες αποστάσεις μεταξύ των σημείων του συνόλου δεδομένων. Η διαμόρφωση του οποίου ελαχιστοποιεί μια συνάρτηση απώλειας που ονομάζεται strain [29]. Η μετρική μέθοδος MDS, η οποία επιτρέπει τη χρήση διαφόρων μετρικών απόστασης και παράλληλα στοχεύει στη διατήρηση της σειράς κατάταξης των αποστάσεων. Η Μη μετρική MDS, η οποία επιδιώκει να διατηρήσει περισσότερο τη σχέση των αποστάσεων παρά τις ακριβείς τιμές τους. Αυτό το επιτυγχάνει μέσω της εύρεσης μιας μη παραμετρικής μονοτονικής σχέσης μεταξύ των διαφορών στον πίνακα του συνόλου δεδομένων και των ευκλείδειων αποστάσεων μεταξύ των στοιχείων, σε συνάρτηση της θέσης κάθε στοιχείου στον χώρο μειωμένων διαστάσεων [30].

2.7.2 Μαθηματική ανάλυση μεθόδου

Το αρχικό βήμα της πολυδιάστατης κλιμάκωσης αποτελείται από την κατασκευή του πίνακα αποστάσεων. Ειδικότερα, σε αυτό το βήμα πραγματοποιείται ο υπολογισμός των ομοιοτήτων ή ανομοιοτήτων μεταξύ των στοιχείων. Μερικές από τις πιο κλασικές μεθόδους υπολογισμού περιλαμβάνουν την ευκλείδεια απόσταση και την απόσταση Manhattan. Η επιλογή του μέτρου απόστασης, εξαρτάται από τη φύση των δεδομένων και τον τύπο της μεθόδου MDS. Έτσι, σε ένα σύνολο δεδομένων με n στοιχεία, ο τελικός πίνακας αποστάσεων D, είναι ένας πίνακας όπου σε κάθε θέση του το στοιχείο d_{ij} ποσοτικοποιεί την απόσταση μεταξύ των οντοτήτων (i) και (j).

Το επόμενο βήμα περιλαμβάνει την εφαρμογή της τεχνικής "διπλό κεντράρισμα (double centering)" [31], η οποία μετασχηματίζει τον πίνακα αποστάσεων μετατρέποντας τον σε αποδεκτή μορφή για την αποσύνθεση ιδιοτιμών.

$$B = -\frac{1}{2} J D^2 J$$

όπου B είναι ο μετασχηματισμένος πίνακας αποστάσεων, D^2 είναι ο πίνακας αποστάσεων και J είναι ένας πίνακας κεντραρίσματος που ορίζεται ως $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, με το I να είναι ο πίνακας ταυτότητας και το $\mathbf{1}$ ένα διάνυσμα μονάδων.

Ο πίνακας B είναι κατάλληλος για την αποσύνθεση ιδιοτιμών από την οποία

εξάγονται τα ιδιοδανύσματα και οι ιδιοτιμές του. Τα ιδιοδανύσματα ταξινομούνται με βάση την ταξινόμηση των ιδιοτιμών, οι οποίες ταξινομούνται σε φθίνουσα σειρά.

Το τελικό βήμα, περιλαμβάνει την τελική διαμόρφωση. Σε αυτό το βήμα εφαρμόζεται ο εξής μετασχηματισμός:

$$X = Q\Lambda^{1/2}$$

όπου X είναι ο πίνακας συντεταγμένων στο νέο χώρο, ο Q είναι ο πίνακας που περιέχει τα επιλεγμένα ιδιοδιανύσματα και $\Lambda^{1/2}$ είναι ο διαγώνιος πίνακας των τετραγωνικών ριζών των επιλεγμένων ιδιοτιμών .

2.7.3 Πεδία εφαρμογής

Η Πολυδιάστατη Κλιμάκωση (MDS) έχει εφαρμοστεί εκτενώς σε διάφορους τομείς της επιστήμης, επιδεικνύοντας την ικανότητα και την ευελιξία της μετατρέποντας και αναλύοντας δεδομένα υψηλών διαστάσεων σε πιο απλοποιημένες μορφές.

Βιολογία

Στον τομέα της βιολογίας και ειδικότερα της γενετικής, η μέθοδος MDS χρησιμοποιείται για τη διερεύνηση της γεωγραφικής γενετικής δομής, αποκαλύπτοντας αποτελεσματικά εγγενή μοτίβα τόσο ιεραρχικά όσο και μη ιεραρχικά. Σε σχετική έρευνα [32] η μέθοδος MDS συνεισφέρει στην καλύτερη ανάλυση της γενετικής δομής των ειδών με βάση τη γεωγραφική τους θέση (γενετική γεωγραφική δομή). Στην έρευνα αναφέρεται ότι η μέθοδος MDS προσφέρει μια νέα προοπτική εντοπίζοντας πιθανές αλληλεπιδράσεις που δεν είναι εμφανείς μέσω των παραδοσιακών τεχνικών ομαδοποίησης. Τέλος η έρευνα καταλήγει ότι η MDS είναι ένα πολύτιμο εργαλείο για την ανάλυση της γεωγραφικής γενετικής δομής, προσφέροντας γνώσεις που μπορεί να παραληφθούν από πιο παραδοσιακές μεθόδους μείωσης διαστάσεων.

Έρευνα αγοράς

Η Πολυδιάστατη Κλιμάκωση (MDS) έχει χρησιμοποιηθεί σε μεγάλο βαθμό στον τομέα της έρευνας αγοράς. Χρησιμοποιείται για την ανάλυση της τιμολόγησης των προϊόντων, στη μελέτη των καναλιών διανομής και των επιπτώσεων της διαφήμισης.

Αποτελεί επίσης ένα βασικό εργαλείο στο μάρκετινγκ για την αποκρυπτογράφηση των προτιμήσεων των καταναλωτών. Οπτικοποιεί τον τρόπο με τον οποίο τα προϊόντα και οι επωνυμίες τοποθετούνται στην αγορά, σε σχέση με τις καταναλωτικές κλίσεις και τις ανταγωνιστικές προσφορές. Επιτρέπει έτσι στις εταιρείες να εντοπίσουν τις τάσεις της αγοράς και να δημιουργήσουν στρατηγικές μάρκετινγκ που ακολουθούν με μεγαλύτερη ακρίβεια τις επιθυμίες των καταναλωτών [33].

Ιατρική

Στην Ιατρική και ειδικότερα σε δεδομένα μαγνητικής τομογραφίας, η μέθοδος MDS χρησιμοποιείται ευρέως. Η πολυπλοκότητα των δεδομένων που εξάγονται είναι τεράστια και η ανάλυση τους πολύ περίπλοκη. Έτσι όπως αναφέρεται και στην έρευνα [34] χρησιμοποιούνται μέθοδοι μείωσης διαστάσεων και συγκεκριμένα η Πολυδιάστατη Κλιμάκωση ώστε να γίνει απλοποίηση των δεδομένων με αποτέλεσμα η οπτική ερμηνεία των δεδομένων να γίνει ευκολότερη.

2.8 Isomap

2.8.1 Θεωρητική ανάλυση

Ο ισομετρικός χάρτης (Isomap) είναι μια μέθοδος μείωσης διαστάσεων χωρίς επίβλεψη η οποία συνδυάζει τα βασικά αλγοριθμικά χαρακτηριστικά της μεθόδου PCA και MDS, επιτυγχάνοντας έτσι υπολογιστική απόδοση, παγκόσμια βελτιστοποίηση (global optimality) και ασυμπτωτικές εγγυήσεις σύγκλισης, με την ευελιξία για εκμάθηση μιας ευρείας κατηγορίας μη γραμμικών πολλαπλών (Manifold) [35]. Μπορεί να χαρακτηριστεί ως μια εξειδικευμένη παραλλαγή της μεθόδου MDS η οποία έχει σχεδιαστεί για να διατηρεί τις γεωδαιτικές αποστάσεις (geodesic distances) [31] μεταξύ των σημείων του συνόλου δεδομένων. Σε αντίθεση με τις γραμμικές προσεγγίσεις, η μέθοδος Isomap επιδιώκει να συλλάβει και να διατηρήσει τις εγγενείς γεωμετρικές ιδιότητες των δεδομένων που βρίσκονται σε μια πολλαπλή (manifold) σε ένα χώρο πολλών διαστάσεων. Αυτή η προσέγγιση αποκλίνει από τις παραδοσιακές μεθόδους εστιάζοντας στη διατήρηση της εγγενής διαστατικής δομής των δεδομένων.

2.8.2 Μαθηματική ανάλυση μεθόδου

Η μεθοδολογία που ακολουθεί ο αλγόριθμος Isomap περιλαμβάνει πέντε βασικά βήματα: Το πρώτο βήμα είναι η κατασκευή του γραφήματος γειτονιάς. Κάθε στοιχείο συνδέεται με τους πλησιέστερους γείτονες του. Οι πλησιέστεροι γείτονες καθορίζονται είτε από τον κανόνα K (καθορίζοντας τον αριθμό των γειτόνων για κάθε σημείο) είτε από τον κανόνα ε (που συνδέει όλα τα σημεία σε μια σταθερή ακτίνα ε). Το συγκεκριμένο βήμα είναι κρίσιμο για την αποτύπωση της τοπικής δομής της πολλαπλής.

Στη συνέχεια, προσδιορίζονται τα βάρη μεταξύ των στοιχείων του γραφήματος, χρησιμοποιώντας την ευκλείδεια απόσταση και κατασκευάζεται ένας πίνακας βαρών. Αυτά τα βάρη είναι καθοριστικά καθώς προσεγγίζουν τις εγγενείς γεωμετρικές ιδιότητες της πολλαπλής.

Έπειτα χρησιμοποιείται ένας αλγόριθμος εύρεσης ελαχίστων μονοπατιών όπως ο αλγόριθμος Dijkstra για την εύρεση των συντομότερων μονοπατιών μεταξύ όλων των ζευγών των σημείων του γραφήματος. Με τη συγκεκριμένη μέθοδο επιτυγχάνεται ο υπολογισμός των γεωδαιτικών αποστάσεων μεταξύ όλων των στοιχείων.

Αφότου υπολογιστούν οι γεωδαιτικές αποστάσεις, τετραγωνίζονται και αποθηκεύονται σε έναν πίνακα αποστάσεων D. Ο πίνακας αποστάσεων D στη συνέχεια μετασχηματίζεται σε έναν πίνακα Gram S [36] μέσω της διαδικασίας "double centering" [31].

Η διαδικασία ολοκληρώνεται με την εφαρμογή της φασματικής αποσύνθεσης στον πίνακα S από την οποία γίνεται η εξαγωγή των ιδιοτιμών και των ιδιοδιανυσμάτων του ($S = UU^T$), διευκολύνοντας την προβολή των δεδομένων στον νέο χώρο μειωμένων διαστάσεων.

2.8.3 Πεδία εφαρμογής

Η εφαρμογή της μεθόδου Isomap εκτείνεται σε πολλούς κλάδους και αποτελεί ένα σημαντικό εργαλείο σε πολλούς τομείς της επιστήμης.

Ρομποτική

Η πλοήγηση στον κόσμο της κινηματικής χαρτογραφίας και του ρομποτικού σχεδιασμού κίνησης εξαρτάται από τον ακριβή ποσοτικό προσδιορισμό της "απόστα-

σης" μεταξύ των διαμορφώσεων ή του "μήκους" των τροχιών στο χώρο διαμόρφωσης. Η μέθοδος Isomap μπορεί να βελτιστοποιήσει τις παραμετροποιήσεις του συστήματος συνδέοντας γειτονικούς κόμβους σε ένα γράφημα και χρησιμοποιώντας τις συντομότερες διαδρομές του γραφήματος για να ορίσει τις αποστάσεις μεταξύ των σημείων. Σε σχετική έρευνα [37], η μέθοδος Isomap χρησιμοποιήθηκε σε ένα ρομπότ κολυμβητή τριών συνδέσμων. Μέσω των μεθόδων μείωσης διαστάσεων γίνεται προσπάθεια για τη δημιουργία καλύτερων αντιστοιχίσεων του χώρου διαμόρφωσης, κάνοντας την οπτική αναπαράσταση των κινήσεων του ρομπότ να αντιστοιχεί περισσότερο στην πραγματική προσπάθεια που απαιτείται. Τα αποτελέσματα δείχνουν την αποτελεσματικότητα της μεθόδου Isomap στην παροχή ακριβέστερης αναπαράστασης των προσπαθειών ρομποτικής μετακίνησης σε διαφορετικές διαμορφώσεις, ενισχύοντας έτσι τον σχεδιασμό και την οπτικοποίηση της κίνησης στον τομέα της ρομποτικής.

Ιατρική

Σε μελέτη που έγινε σχετικά με τον καρκίνο του πνεύμονα [38], ο οποίος είναι η κύρια αιτία θανάτων που σχετίζονται με καρκίνο παγκοσμίως, οι ερευνητές αξιοποίησαν τον αλγόριθμο Isomap για να εμβαθύνουν σε δεδομένα γονιδιακής έκφρασης. Τα δεδομένα γονιδιακής έκφρασης είναι πολύπλοκα και πολυδιάστατα δεδομένα, που χρειάζονται προεπεξεργασία για να ανακαλυφθούν οι δομικές συσχετίσεις και τα εγγενή χαρακτηριστικά τους. Η μέθοδος Isomap αξιοποιήθηκε για την αποκάλυψη αυτών των χαρακτηριστικών των δεδομένων, υποδεικνύοντας μέσα από τα αποτελέσματα της ότι τα επίπεδα γονιδιακής έκφρασης συσχετίζονται καλά με παθολογικά χαρακτηριστικά. Στα αποτελέσματα της έρευνας υπογραμμίζεται η αποτελεσματικότητα της μεθόδου στην αποκάλυψη της εγγενούς δομής και στην αποτελεσματική διάκριση μεταξύ διαφορετικών ιατρικών καταστάσεων.

Αναγνώριση υποκείμενων στοιχείων σε εικόνες

Άλλο ένας τομέας της τεχνολογίας στον οποίο έχει εφαρμογή η μέθοδος Isomap είναι στην αναγνώριση εικόνων. Σύμφωνα με σχετική έρευνα [35] χρησιμοποιείται με επιτυχία για τον εντοπισμό των πραγματικών υποκείμενων παραγόντων σε σύνολα δεδομένων υψηλών διαστάσεων, όπως συνθετικές εικόνες προσώπου,

εικόνες χειρονομιών και χειρόγραφα ψηφία. Οι συνθετικές εικόνες προσώπων, οι οποίες αρχικά αναπαριστώνται σε έναν μεγάλο χώρο με pixels, ουσιαστικά ενσωματώνουν παραλλαγές στις εκφράσεις του προσώπου, τους προσανατολισμούς και τις συνθήκες φωτισμού. Η μέθοδος Isomap έχει την ικανότητα να ξετυλίγει αυτές τις μεταβλητές, διευκολύνοντας έτσι εργασίες όπως η αναγνώριση προσώπου και η ανάλυση έκφρασης. Ομοίως, εικόνες χειρονομιών οι οποίες αποτυπώνονται σε υψηλό αριθμό διαστάσεων μπορούν να απλοποιηθούν αποτελεσματικά καθώς οι πραγματικοί βαθμοί ελευθερίας που εμπλέκονται στην ανθρώπινη κίνηση είναι περιορισμένοι. Η μέθοδος Isomap επιτρέπει τη μείωση αυτού του χώρου υψηλών διαστάσεων σε μια πιο διαχειρίσιμη μορφή όπου διατηρείται η ουσία της χειρονομίας. Τέλος, η αναγνώριση χειρόγραφων ψηφίων επωφελείται επίσης από τη δυνατότητα μείωσης διαστάσεων της μεθόδου Isomap. Μέσω της εφαρμογής της μεθόδου, οι περιπλοκές στις κατηγορίες ψηφίων γίνονται πιο ευδιάκριτες, καθιστώντας την ταξινόμηση πιο εύκολη.

2.9 Local Linear embedding

2.9.1 Θεωρητική ανάλυση

Η τοπική γραμμική ενσωμάτωση (LLE) είναι μια μη γραμμική, χωρίς επίβλεψη μέθοδος μείωσης διαστάσεων που στοχεύει στη διατήρηση των γεωμετρικών χαρακτηριστικών του συνόλου δεδομένων. Έχει αναγνωριστεί για την αποτελεσματικότητα της στην απλοποίηση πολύπλοκων, πολλαπλών δομών (manifold structures) σε δεδομένα υψηλών διαστάσεων. Η βασική ίδεα της μεθόδου αξιοποιεί την αρχή ότι κάθε στοιχείο στον πολυδιάστατο χώρο και οι άμεσοι γείτονες του μπορούν να ανακατασκευαστούν γραμμικά σε ένα χώρο λιγότερων διαστάσεων με την προϋπόθεση ότι βρίσκονται σε ένα τοπικά γραμμικό τμήμα της πολλαπλής (manifold).

2.9.2 Μαθηματική ανάλυση

Η μαθηματική μεθοδολογία για την εφαρμογή της μεθόδου τοπικής γραμμικής ενσωμάτωσης ακολουθεί τα εξής βήματα: Στο πρώτο βήμα, γίνεται ο υπολογισμός των k πλησιέστερων γειτόνων κάθε στοιχείου x_i στο σύνολο δεδομένων. Οι πλησιέστεροι γείτονες καθορίζονται είτε από τον κανόνα K (καθορίζοντας τον αριθμό των

γειτόνων για κάθε σημείο) είτε από τον κανόνα ε (που συνδέει όλα τα σημεία σε μια σταθερή ακτίνα ε). Αυτό το βήμα προσπαθεί να καταγράψει την τοπική γεωμετρία του κάθε στοιχείου.

Στο επόμενο βήμα, γίνεται η κατασκευή του πίνακα βαρών κάθε στοιχείου. Εδώ υπολογίζονται τα βάρη που ανακατασκευάζουν καλύτερα με γραμμικό τρόπο κάθε στοιχείο του συνόλου δεδομένων από τους γείτονες του, υπό τον περιορισμό ότι το άθροισμα των βαρών ισούται με τη μονάδα. Ειδικότερα, επιλύοντας ένα πρόβλημα περιορισμένων ελαχίστων τετραγώνων:

$$\min_{w_i} \left\| x_i - \sum_{j \in J_i} w_{ij} x_j \right\|^2, \quad \text{s.t.} \sum_{j \in J_i} w_{ij} = 1.$$

Η μορφή του πίνακα αυτού του προβλήματος βελτιστοποίησης περιλαμβάνει τον υπολογισμό του $Q_i = G_i^T G_i$, όπου $G_i = [x_{i1} - x_i, \ldots, x_{ik} - x_i]$. Έτσι ο στόχος γίνεται:

$$\min_{w_i} w_i^T Q_i w_i, \quad \text{s.t.} \, w_i^T \mathbf{1} = 1,$$

όπου $\mathbf{1}$ είναι ένα διάνυσμα μονάδων. Η λύση, δεδομένου του περιορισμού, μπορεί να βρεθεί χρησιμοποιώντας έναν πολλαπλασιαστή Lagrange, που οδηγεί στα βέλτιστα βάρη w_i^*.

Το τελευταίο βήμα περιλαμβάνει τον υπολογισμό της ενσωμάτωσης μειωμένων διαστάσεων των στοιχείων του συνόλου δεδομένων. Πραγματοποιείται η προβολή των αρχικών στοιχείων σε ένα χώρο λιγότερων διαστάσεων όπου διατηρείται καλύτερα η τοπική γεωμετρία που υποδεικνύεται βάση των βαρών ανακατασκευής, που υπολογίστηκαν στο προηγούμενο βήμα. Για να το επιτύχει αυτό η μέθοδος χρησιμοποιεί τη βελτιστοποίηση μιας συνάρτησης κόστους που έχει σχεδιαστεί για να διατηρεί τις σχετικές αποστάσεις που εξάγονται από τα βάρη ανακατασκευής.

$$\min_{Y} \sum_{i=1}^{N} \left\| y_i - \sum_{j \in J_i} w_{ij} y_j \right\|^2, \quad \text{s.t.} \, \frac{1}{N} \sum_{i=1}^{N} y_i = 0, \; \frac{1}{N} \sum_{i=1}^{N} y_i y_i^T = I.$$

Ένας αραιός, συμμετρικός και θετικός πίνακας $M = (I - W)^T (I - W)$ κατασκευάζεται από τον πίνακα βάρους W και η ενσωμάτωση υπολογίζεται μέσω των ιδιοδιανυσμάτων του M που αντιστοιχεί στις μικρότερες ιδιοτιμές, με εξαίρεση το πρώτο ιδιοδιάνυσμα το οποίο σχετίζεται με τη μηδενική ιδιοτιμή, το οποίο συνήθως παρα-

λείπεται.

2.9.3 Πεδία εφαρμογής

Αναγνώριση προσώπων

Η μέθοδος της τοπικής γραμμικής ενσωμάτωσης (LLE) χρησιμοποιείται ευρέως στον τομέα της αναγνώρισης προσώπων. Η αναγνώριση προσώπων και η ταξινόμηση τους αποτελεί μια περίπλοκη διαδικασία που τις περισσότερες φορές παραδοσιακές τεχνικές μείωσης διαστάσεων όπως η PCA και η MDS αδυνατούν να τις διαχειριστούν. Αντίθετα, η LLE λόγω της εγγενής δυνατότητας της να καταγράφει την τοπική δομή των δεδομένων είναι πιο αποτελεσματική. Τα παραπάνω επικυρώνει και η έρευνα [39], στην οποία οι συγγραφείς εφάρμοσαν τη μέθοδο LLE για να μειώσουν τις διαστάσεις εικόνων προσώπων και να βελτιώσουν την απόδοση των μοντέλων αναγνώρισης προσώπου.

Ταξινόμηση σημάτων

Η Τοπική Γραμμική Ενσωμάτωση χρησιμοποιείται επίσης και στον τομέα της ταξινόμησης σημάτων. Ειδικότερα όπως φαίνεται και από την έρευνα [40] στον τομέα της αυτοματοποιημένης, ταχείας ταξινόμησης σημάτων ραδιοσυχνοτήτων, η εφαρμογή της μεθόδου φαίνεται να έχει πολύ καλά αποτελέσματα ενισχύοντας σημαντικά την απόδοση του ταξινομητή SVM που χρησιμοποιήθηκε στα πειράματα.

Διάγνωση σφαλμάτων σε μηχανήματα

Άλλη μια πτυχή της τεχνολογίας στην οποία έχει εφαρμογή η μέθοδος LLE είναι η διάγνωση σφαλμάτων. Μια προσέγγιση της τοπικής γραμμικής ενσωμάτωσης προσαρμοσμένη για τη διάγνωση σφαλμάτων σε μηχανήματα έχει διερευνηθεί. Στο κομμάτι της έρευνας [41] χρησιμοποιήθηκαν τεχνολογίες παλινδρόμησης ελάχιστης γωνίας και ελαστικού δικτύου για τον υπολογισμό της τοπικής δομής. Τα αποτελέσματα της έρευνας έδειξαν ότι η χρήση της μεθόδου τοπικής γραμμικής ενσωμάτωσης βελτιώνει σημαντικά την ακρίβεια διάγνωσης σφαλμάτων σε μηχανήματα.

2.10 Linear Discriminant Analysis

2.10.1 Θεωρητική ανάλυση

Η Γραμμική Διακριτική Ανάλυση (LDA) είναι μια στατιστική μέθοδος που χρησιμοποιείται τόσο για την επίλυση προβλημάτων ταξινόμησης όσο και σαν μέθοδος για τη μείωση των διαστάσεων ενός συνόλου δεδομένων. Αποσκοπεί στην εύρεση γραμμικών συνδυασμών μεταξύ των χαρακτηριστικών με σκοπό τον καλύτερο δυνατό διαχωρισμό των κλάσεων βάση αυτών των συνδυασμών. Οι παραδοχές που κάνει είναι ότι τα δεδομένα ακολουθούν κανονική κατανομή, και ότι οι τιμές κάθε χαρακτηριστικού έχουν την ίδια διακύμανση. Η μέθοδος προβάλει τα δεδομένα σε ένα χώρο λιγότερων διαστάσεων που μεγιστοποιεί τη διαχωριστικότητα μεταξύ των κλάσεων. Αυτό επιτυγχάνεται με την εύρεση των γραμμικών διακρίσεων (linear discriminants) που ως στόχο έχουν να μεγιστοποιήσουν την αναλογία της διακύμανσης μεταξύ των κλάσεων προς τη διακύμανση εντός της κλάσης, διασφαλίζοντας έτσι ότι οι κλάσεις είναι όσο το δυνατόν πιο διακριτές. Είναι αρκετά αποδοτική μέθοδος για δεδομένα με γραμμικές σχέσεις και δεν χρειάζεται μεγάλη υπολογιστική ισχύ. Παράλληλα μπορεί να διαχειριστεί την πολυσυγγραμμικότητα (multicollinearity). Μια από τις κύριες αδυναμίες της μεθόδου οφείλεται στις δύο παραδοχές που κάνει. Ειδικότερα η μέθοδος θεωρεί ότι τα δεδομένα ακολουθούν την κανονική κατανομή και είναι γραμμικώς διαχωρίσιμα. Αυτές οι δύο παραδοχές πολλές φορές δεν ισχύουν και ιδιαίτερα σε δεδομένα του πραγματικού κόσμου.

2.10.2 Μαθηματική ανάλυση

Η LDA ακολουθεί ένα σύνολο από διακριτά βήματα. Το πρώτο βήμα είναι ο υπολογισμός του μέσου για κάθε αριθμητικό χαρακτηριστικό. Δεδομένου ενός συνόλου δεδομένων με n διαστάσεις και d δείγματα, όπου τα δείγματα χωρίζονται σε δύο κλάσεις, c_1 και c_2, το πρώτο βήμα είναι να υπολογιστούν τα διανύσματα των μέσων για κάθε κατηγορία . Εάν τα u_1 και u_2 είναι τα μέσα διανύσματα για τις κλάσεις c_1 και c_2 αντίστοιχα, ισχύει οτι:

$$u_1 = \frac{1}{n_1} \sum_{x_i \in c_1} x_i, \quad u_2 = \frac{1}{n_2} \sum_{x_i \in c_2} x_i$$

όπου n_1 και n_2 είναι ο αριθμός των δειγμάτων στις κλάσεις c_1 και c_2, αντίστοιχα.

Στη συνέχεια το επόμενο βήμα είναι ο υπολογισμός των πινάκων διασποράς (Scatter matrices). Οι πίνακες διασποράς ποσοτικοποιούν το πόσο εξαπλώνονται τα δεδομένα μιας συγκεκριμένης κατηγορίας από τον μέσο όρο. Υπάρχουν δύο είδη πινάκων διασποράς: εντός κλάσης (S_w) και μεταξύ κλάσης (S_b). Ο πίνακας διασποράς εντός κλάσης (S_w) ορίζεται ως το άθροισμα των πινάκων διασποράς για κάθε ετικέτα του χαρακτηριστικού κλάσης:

$$S_w = S_1 + S_2$$

Οι πίνακες S_1 και S_2 ορίζονται ως

$$S_1 = \sum_{x_i \in c_1} (x_i - u_1)(x_i - u_1)^T, \quad S_2 = \sum_{x_i \in c_2} (x_i - u_2)(x_i - u_2)^T$$

Για τον υπλογισμό του πίνακα διασποράς μεταξύ των κλάσεων (S_b) ισχύει ο τύπος

$$S_b = (u_1 - u_2)(u_1 - u_2)^T$$

Ο στόχος της μεθόδου όπως προαναφέρθηκε είναι να προβάλλει τα δεδομένα σε ένα χώρο που μεγιστοποιεί τη διασπορά μεταξύ των κλάσεων και παράλληλα ελαχιστοποιεί τη διασπορά εντός της ίδιας κλάσης. Αυτό επιτυγχάνεται με τη μεγιστοποίηση του ακόλουθου κριτηρίου:

$$J(v) = \frac{v^T S_b v}{v^T S_w v}$$

Η μεγιστοποίηση του $J(v)$ οδηγεί στην επίλυση του γενικευμένου προβλήματος της ιδιοτιμής:

$$S_w^{-1} S_b v = \lambda v$$

Μέσω της λύσης του γενικευμένου προβλήματος αποκτώνται οι ιδιοτιμές λ και τα ιδιοδιανύσματα v. Μέσω των ιδιοτιμών γίνεται φθίνουσα ταξινόμηση των ιδιοδιανυσμάτων. Γενικότερα ισχύει ότι το ιδιοδιάνυσμα που σχετίζεται με τη μεγαλύτερη ιδιοτιμή προσδιορίζουν τις κατευθύνσεις που μεγιστοποιούν τον διαχωρισμό μεταξύ των κλάσεων.

Το τελικό βήμα είναι η προβολή των σημείων στον νέο χώρο λιγότερων διαστάσεων. Αυτό γίνεται επιλέγοντας τα k πρώτα ιδιοδιανύσματα και υπολογίζοντας το εσωτερικό γινόμενο μεταξύ του πίνακα ιδιοδιανυσμάτων και των στοιχείων του συνόλου δεδομένων.

$$y_i = v^T x_i$$

2.10.3 Πεδία εφαρμογής

Αυτόματη αναγνώριση ομιλίας

Στον τομέα της αυτόματης αναγνώρισης ομιλίας από βίντεο, η μέθοδος LDA επέφερε ιδιαίτερα θετικά αποτελέσματα, και βελτίωσε τα υπάρχοντα μοντέλα. Ειδικότερα σε σχετική έρευνα που πραγματοποιήθηκε [42] η μέθοδος LDA χρησιμοποιήθηκε για την εξαγωγή χαρακτηριστικών από μια τρισδιάστατη περιοχή ενδιαφέροντος (ROI) γύρω από το στόμα του ομιλητή σε διαδοχικά καρέ βίντεο. Τα συμπεράσματα της έρευνας καταλήγουν στο συμπέρασμα ότι η LDA, μεγιστοποιώντας τη δυνατότητα διαχωρισμού κλάσεων, παρέχει μια αποτελεσματική μέθοδο για την εξαγωγή χαρακτηριστικών σε συστήματα ανάγνωσης ομιλίας, έχοντας συγκριτικά με άλλες μεθόδους πολύ καλύτερα αποτελέσματα.

Αναγνώριση προσώπου

Ο τομέας της αναγνώρισης προσώπου έχει επωφεληθεί επίσης από τη μέθοδο LDA. Έχουν αναπτυχθεί διάφορες παραλλαγές της βασικής μεθόδου που χρησιμοποιούνται σε μεγάλο βαθμό και με θετικά αποτελέσματα σε ζητήματα που αφορούν την ανάλυση και την αναγνώριση προσώπων. Σε σχετική έρευνα [43] αποδεικνύεται η αποτελεσματικότητα της μεθόδου LDA στον τομέα της αναγνώρισης προσώπων.

Ιατρική

Στον τομέα της ιατρικής έχει χρησιμοποιηθεί η μέθοδος LDA για την προεπεξεργασία πολυδιάστατων δεδομένων. Ειδικότερα στην παρούσα έρευνα [44] γίνεται πρόβλεψη της στεφανιαίας νόσου. Η έρευνα επικεντρώνεται σε έναν πληθυσμό 10.265 ατόμων που αξιολογήθηκαν για ισχαιμία του μυοκαρδίου, από τα δεδομένα εξήχθησαν 22 χαρακτηριστικά για ανάλυση. Η έρευνα αποσκοπεί στην ταξινόμηση

των δειγμάτων ως φυσιολογικών ή παθολογικών και η συμβολή της LDA στην προεπεξεργασία των δεδομένων επέφερε σημαντικές βελτιώσεις.

2.11 Independent Component Analysis

2.11.1 Θεωρητική ανάλυση

Η Ανεξάρτητη Ανάλυση Συστατικών (ICA) είναι μια στατιστική μέθοδος. Χρησιμοποιείται στον τομέα της ανάλυσης δεδομένων για την αποκάλυψη κρυμμένων στοιχείων μέσα σε σύνολα τυχαίων μεταβλητών, μετρήσεων ή σημάτων. Η βασική αρχή της μεθόδου είναι η υπόθεση ότι τα δεδομένα που παρατηρούνται είναι συνδυασμοί άγνωστων και κρυφών παραγόντων και ο τρόπος με τον οποίο αναμειγνύονται είναι επίσης άγνωστος. Στόχος της μεθόδου είναι να ανακτήσει τα αρχικά σήματα πηγές βάση των οποίων προέκυψαν τα αναμειγμένα δεδομένα. Οι κύριες υποθέσεις που κάνει η μέθοδος είναι ότι τα συστατικά (σήματα πηγές) είναι στατιστικά ανεξάρτητα μεταξύ τους και επίσης ότι τα δεδομένα ακολουθούν μια μη κανονική κατανομή. Ειδικότερα λόγω του θεωρήματος κεντρικού ορίου (Central Limit Theorem) [45], το άθροισμα των ανεξάρτητων μεταβλητών τείνει προς μια κανονική κατανομή.

Παρόλο που η μέθοδος αυτήν δεν είναι εν γένει μια μέθοδος μείωσης διαστάσεων χρησιμοποιείται στον τομέα ευρέως. Αναζητώντας και εξάγοντας τα ανεξάρτητα στοιχεία ενός συνόλου δεδομένων καθίσταται εφικτή η αποκάλυψη της υποκείμενης δομής των δεδομένων. Επιπλέον μπορεί να βοηθήσει στη μείωση του θορύβου και του πλεονασμού στα δεδομένα. Τέλος εστιάζοντας στα ανεξάρτητα στοιχεία που δεν αντιπροσωπεύουν θόρυβο ή ανεπιθύμητα σήματα γίνεται απλοποίηση του συνόλου δεδομένων.

2.11.2 Μαθηματική ανάλυση

Το θεμελιώδες μαθηματικό μοντέλο της μεθόδου ICA έχει ως εξής:

$$X = AS$$

Όπου το X είναι ένας πίνακα διαστάσεων $m \times n$ και αντιπροσωπεύει τα αναμιγμένα σήματα. Ο A είναι ένας άγνωστος πίνακας συντελεστών ανάμειξης $m \times m$ διαστάσεων και ο S είναι ένας πίνακας $m \times n$ διαστάσεων που περιέχει τα ανεξάρτητα σήματα που πρόκειται να εξαχθούν μετά την εφαρμογή της μεθόδου.

Ο βασικός στόχος της μεθόδου είναι η λύση της εξίσωσης:

$$S = A^{-1}X$$

Επομένως, απαιτείται ο υπολογισμός τόσο του πίνακα S όσο και του πίνακα A. Για να γίνει αυτό η μέθοδος θεωρεί ότι τα σήματα ακολουθούν μια μη κανονική κατανομή και είναι ανεξάρτητα μεταξύ τους. Χρησιμοποιώντας αυτές τις δύο παραδοχές η μέθοδος ICA μπορεί να εκτιμήσει τόσο τον πίνακα A όσο και τον S. Μια κοινή προσέγγιση για να επιτευχθεί αυτό είναι με τη μεγιστοποίηση της μη-Γκαουσιανότητας (Maximizing Non-Gaussianity), η οποία υπολογίζεται χρησιμοποιώντας την κύρτωση (kurtosis) ή την Negentropy [46]. Η κύρτωση είναι ένα μέτρο της «ουράς» της κατανομής πιθανοτήτων και ορίζεται για μια τυχαία μεταβλητή Y ως:

$$\text{Excess Kurtosis} = \frac{E[(X-\mu)^4]}{(E[(X-\mu)^2])^2}$$

Ενώ η Negentropy ορίζεται ως:

$$J(Y) = H(Y_{gauss}) - H(Y)$$

όπου $H(Y)$ είναι η εντροπία του Y, και Y_{gauss} είναι μια γκαουσιανή μεταβλητή με την ίδια συνδιακύμανση με την Y. Η μεγιστοποίηση της Negentropy οδηγεί σε περισσότερες μη κανονικές κατανομές, καθώς οι κατανομές Gauss έχουν τη μέγιστη εντροπία για μια δεδομένη συνδιακύμανση [47].

Υπάρχουν διάφορες υλοποιήσεις της μεθόδου. Η περισσότερο γνωστή και ευρέως χρησιμοποιούμενη είναι η FastICA ή οποία προσπαθεί προσεγγιστικά μέσω μιας επανάληψης να μεγιστοποιήσει τη μη-Γκαουσιανότητα (Non-Gaussianity) για την εκτίμηση των ανεξάρτητων στοιχείων.

2.11.3 Πεδία εφαρμογής

Χημεία

Στον τομέα της χημείας και ειδικότερα στο κομμάτι της ανάλυσης μιγμάτων και φασματικών δεδομένων η μέθοδος (ICA) έχει χρησιμοποιηθεί σε μεγάλο βαθμό και αποτελεί ένα αρκετά χρήσιμο εργαλείο για τους επιστήμονες. Στην παρούσα έρευνα [48] η οποία αφορά τη χημειομετρία, τονίζεται η ικανότητα της μεθόδου να αναλύει και να ξεχωρίζει στατιστικά ανεξάρτητα συστατικά από φασματικά σήματα μιγμάτων, μια εργασία που αποδεικνύεται πρόκληση όταν τα φασματικά σήματα μοιράζονται ομοιότητες ή εξαρτήσεις, όπως συχνά παρατηρείται με χημικές ενώσεις.

Ανάλυση σημάτων και αφαίρεση θορύβου

Η κύρια λειτουργία της μεθόδου ICA είναι ο διαχωρισμός αναμιγνυόμενων σημάτων. Όπως υπογραμμίζεται και από την έρευνα [49] η μέθοδος ICA είναι καίριας σημασίας για την επεξεργασία δεδομένων που καταγράφονται από πολλαπλούς αισθητήρες, καθώς κάθε αισθητήρας καταγράφει συχνά ένα θορυβώδες μείγμα αρχικών σημάτων πηγής. Με τη χρήση της μεθόδου επιτυγχάνεται βελτίωση στην ποιότητα του σήματος φιλτράροντας επιλεκτικά τα στοιχεία θορύβου που προσδιορίζονται μέσω της μεθόδου, επιτρέποντας έτσι την ανακατασκευή των καθαρισμένων αρχικών σημάτων από τα βελτιωμένα ανεξάρτητα στοιχεία.

Βιοϊατρική

Στον τομέα της βιοϊατρικής η μέθοδος ICA έχει προσφέρει σημαντικό έργο. Ιδιαίτερα μέσω των εφαρμογών της στην επεξεργασία σήματος για διάφορα βιοϊατρικά δεδομένα όπως ERP, EEG, fMRI και στην οπτική απεικόνιση. Μία από τις βασικές εφαρμογές της μεθόδου είναι στην ανάλυση EEG, σύμφωνα με την έρευνα [50] η μέθοδος ICA έπαιξε καθοριστικό ρόλο στην εξαγωγή σημαντικών πληροφοριών από δεδομένα EEG. Τα αποτελέσματα της έρευνας τονίζουν την αποτελεσματικότητα της μεθόδου και υπογραμμίζουν την ευρεία εφαρμογή της στον τομέα της νευροεπιστήμης.

2.12 Kernel PCA

2.12.1 Θεωρητική ανάλυση

Η μέθοδος Ανάλυση Πρωταρχικών Συνιστωσών πυρήνα (Kernel PCA) είναι μια επέκταση της κλασσικής μεθόδου (PCA). Η κλασσική μέθοδος PCA είναι μια γραμμική μέθοδος η οποία παρουσιάζει αδυναμία στην καταγραφή μη γραμμικών σχέσεων που πολλές φορές υπάρχουν ανάμεσα σε πολύπλοκα, μη γραμμικά σύνολα δεδομένων που συναντιούνται συνήθως σε εφαρμογές του πραγματικού κόσμου. Η επέκταση της μεθόδου έρχεται να δώσει λύση σε αυτό το πρόβλημα εισάγοντας τις συναρτήσεις πυρήνα και το τρικ πυρήνα (kernel trick). Αυτή η προσθήκη στη μέθοδο έρχεται να δώσει λύση στον αρχικό περιορισμό, σχετικά με την ανάγκη της γραμμικότητας των δεδομένων. Ουσιαστικά με την προσθήκη των συναρτήσεων πυρήνα στα αρχικά βήματα της Kernel PCA το σύνολο δεδομένων μεταφέρεται σε ένα χώρο περισσότερων διαστάσεων. Έτσι, Διευκολύνεται ο εντοπισμός των πρωταρχικών συνιστωσών στον χώρο των υψηλών διαστάσεων, επιτρέποντας τη καταγραφή και μη γραμμικών σχέσεων στα δεδομένα. Για την ορθή λειτουργία της μεθόδου απαιτείται η κατάλληλη επιλογή συνάρτησης πυρήνα, καθώς μέσω αυτής ορίζεται ο χώρος στον οποίο θα αναλυθούν τα δεδομένα. Ανάλογα με τη μορφή των δεδομένων χρησιμοποιούνται διαφορετικοί τύποι πυρήνα, οι πιο συνηθισμένοι τύποι πυρήνα είναι ο πολυωνυμικός πυρήνας, ο πυρήνας της συνάρτησης ακτινικής βάσης (RBF) και ο σιγμοειδές πυρήνας. Ανάλογα τον τύπο του πυρήνα υπάρχουν και οι υπερπαράμετροι του οι οποίοι πρέπει να οριστούν κατάλληλα.

2.12.2 Μαθηματική ανάλυση

Η μαθηματική ανάλυση της μεθόδου δεν αποκλίνει σε μεγάλο βαθμό από την κλασσική μέθοδο. Τα βήματα που ακολουθεί είναι τα εξής: Αρχικά, επιλέγεται μια συνάρτηση πυρήνα $K(x_i, x_j)$ για να υπολογιστεί η ομοιότητα ανά ζεύγη μεταξύ όλων των σημείων x_i και x_j στο σύνολο δεδομένων. Αυτή η συνάρτηση πυρήνα αντιστοιχίζει έμμεσα τα δεδομένα σε έναν χώρο περισσότερων διαστάσεων από τον αρχικό. Σε πολλές εφαρμογές του πραγματικού κόσμου, τα δεδομένα ενδέχεται να μην μπορούν να διαχωριστούν γραμμικά στον αρχικό χώρο. Έτσι, μέσω της συνάρτησης πυρήνα τα δεδομένα μεταφέρονται σε έναν χώρο περισσότερων διαστάσεων, με κύ-

ριο στόχο τον εντοπισμό γραμμικών σχέσεων στον νέο χώρο. Η βασική ιδέα είναι ότι στον νέο χώρο υψηλότερων διαστάσεων, τα δεδομένα θα είναι ευκολότερο να διαχωριστούν γραμμικά και να εντοπιστούν γραμμικές σχέσεις μεταξύ τους.

Στο επόμενο βήμα γίνεται η κατασκευή του πίνακα πυρήνα K, ο οποίος κατασκευάζεται από τις ομοιότητες κατά ζεύγη, όπου για κάθε στοιχείο ισχύει ότι $K_{ij} = K(x_i, x_j)$. Αφότου κατασκευαστεί ο πίνακας πυρήνα K κεντραρίζεται χρησιμοποιώντας τον τύπο:

$$K' = K - 1_n K - K 1_n + 1_n K 1_n$$

όπου 1_n είναι ένας πίνακας $n \times n$ που αποτελείται από $\frac{1}{n}$ ενώ το n είναι ο αριθμός των δεδομένων. Αυτό το βήμα διασφαλίζει ότι τα δεδομένα βρίσκονται στο κέντρο του νέου χώρου χαρακτηριστικών.

Στη συνέχεια, η μέθοδος ακολουθεί τα βήματα της κλασσικής μεθόδου. Αναλυτικότερα, γίνεται η αποσύνθεση των ιδιοτιμών στον κεντραρισμένο πίνακα K' μέσω της οποίας ανακτούνται οι ιδιοτιμές λ_i και τα ιδιοδιανύσματα α_i. Πραγματοποιείται η ταξινόμηση των ιδιοδιανυσμάτων βάση των ιδιοτιμών σε φθίνουσα σειρά, και γίνεται η επιλογή των k πρώτων ιδιοδιανυσμάτων.

Το τελικό βήμα είναι η προβολή των δεδομένων μέσω των επιλεγμένων ιδιοδιανυσμάτων, υπολογίζοντας το γινόμενο μεταξύ των ιδιοδιανυσμάτων α_i και του πυρήνα ενός σημείου δεδομένων με όλα τα άλλα σημεία του συνόλου δεδομένων.

$$\text{Προβολή του } x_i \text{ βάση των } k \text{ πρωταρχικών συνιστωσών} = \sum_{j=1}^{n} \alpha_{kj} K(x_i, x_j)$$

2.12.3 Πεδία εφαρμογής

Αναγνώριση προτύπων

Η μέθοδος Kernel PCA επιδιώκει να δώσει λύσεις σε προβλήματα τα οποία αντιμετωπίζει η κλασσική μέθοδος PCA. Σε σχετική έρευνα που πραγματοποιήθηκε [51] , κατά την οποία έγινε σύγκριση των μεθόδων PCA και Kernel PCA σε σύνολα δεδομένων που αποτελούνται από εικόνες, τονίζεται η υπεροχή της μεθόδου Kernel PCA έναντι της κλασσικής μεθόδου στην εξαγωγή ουσιαστικών μοτίβων και χαρακτηριστικών από πολύπλοκα μη γραμμικά σύνολα δεδομένων. Τα αποτελέσματα των δύο μεθόδων σύμφωνα με την έρευνα χρησιμοποιήθηκαν από τον ίδιο ταξινομητή και

μέσω των αποτελεσμάτων του ταξινομητή έγινε η τελική σύγκριση. Χρησιμοποιώντας τη μέθοδο Kernel PCA, η μελέτη δείχνει βελτιωμένη απόδοση στις εργασίες ταξινόμησης και στην κατασκευή μοντέλων ανθρώπινου προσώπου, χαρακτηρίζοντας τη μέθοδο Kernel PCA ως ικανή να αποκαλύπτει πολύπλοκες κρυφές δομές μέσα στα δεδομένα τις οποίες δεν μπορεί να εντοπίσει η κλασσική μέθοδος.

Αφαίρεση θορύβου

Η μέθοδος Kernel PCA είναι αποτελεσματική και στην αφαίρεση θορύβου. Σε σχετική έρευνα [52], τονίζεται η αποτελεσματικότητα της μεθόδου να παράγει πιο ακριβείς προ-εικόνες (Pre-Images: ανακατασκευή των αρχικών εικόνων βάση των αποτελεσμάτων των μεθόδων μείωσης διαστάσεων) από την κλασσική PCA, ειδικά σε περιπτώσεις όπου τα δεδομένα και η δομή τους ήταν εγγενώς μη γραμμική. Σε πειράματα που αφορούσαν σύνολα δεδομένων με χειρόγραφα ψηφία, η Kernel PCA έδειξε καλύτερη απόδοση στην αφαίρεση θορύβου, επιτυγχάνοντας σημαντικά καλύτερα αποτελέσματα σε σύγκριση με την κλασσική PCA. Η βελτίωση αυτήν αποδίδεται κυρίως στην ικανότητα της Kernel PCA να εξάγει μεγαλύτερο αριθμό χαρακτηριστικών που μεταφέρουν πληροφορίες σχετικά με τη δομή των δεδομένων, μια εργασία στην οποία η γραμμική PCA αποτυγχάνει εάν η δομή είναι μη γραμμική.

2.13 Boruta algorithm

2.13.1 Θεωρητική ανάλυση

Ο αλγόριθμος Boruta είναι ένας αλγόριθμος επιλογής χαρακτηριστικών, ο οποίος χρησιμοποιείται ευρέως στον τομέα της ανάλυσης δεδομένων για την αξιολόγηση της σημαντικότητας των χαρακτηριστικών που απαρτίζουν ένα σύνολο δεδομένων. Αξιοποιεί την εγγενή δυνατότητα των μεθόδων εκμάθησης συνόλου για την αξιολόγηση των χαρακτηριστικών και τον υπολογισμό της σημαντικότητας του εκάστοτε χαρακτηριστικού.

2.13.2 Μαθηματική ανάλυση

Τα βήματα τα οποία ακολουθεί ο αλγόριθμος για την αξιολόγηση των χαρακτηριστικών ενός συνόλου δεδομένων είναι τέσσερα. Αρχικά γίνεται η κατασκευή των

χαρακτηριστικών σκιάς. Τα συγκεκριμένα χαρακτηριστικά είναι ίδια με τα κανονικά χαρακτηριστικά αλλά οι τιμές κάθε χαρακτηριστικού είναι σε τυχαία θέση και όχι στην κανονική. Αυτή η διαδικασία διασφαλίζει ότι οποιοδήποτε προγνωστικό σήμα στα χαρακτηριστικά σκιάς οφείλεται καθαρά στην τύχη.

Στη συνέχεια εκπαιδεύεται ένας αλγόριθμος εκμάθησης συνόλου στο σύνολο δεδομένων που έχει επαυξηθεί με τα χαρακτηριστικά σκιάς και υπολογίζεται η σημαντικότητα κάθε χαρακτηριστικού. Ο αλγόριθμος βρίσκει το χαρακτηριστικό σκιάς με τη μεγαλύτερη σημαντικότητα και το συγκρίνει με τη σημαντικότητα κάθε κανονικού χαρακτηριστικού. Όσα από τα χαρακτηριστικά έχουν σημαντικότητα μικρότερη από αυτήν του χαρακτηριστικού σκιάς σηματοδοτούνται ως μη σημαντικά χαρακτηριστικά.

Ο αλγόριθμος επαναλαμβάνει επαναληπτικά τη διαδικασία αξιολόγησης, κάθε φορά σηματοδοτώντας τα χαρακτηριστικά που ταξινομούνται ως ασήμαντα. Με κάθε επανάληψη, τα χαρακτηριστικά σκιάς ανακατασκευάζονται και η σύγκριση γίνεται ξανά. Ανάλογα την υλοποίηση του αλγορίθμου τα χαρακτηριστικά που θεωρούνται ασήμαντα αφαιρούνται απευθείας ή μετά από κάποιο αριθμό επαναλήψεων κατά τις οποίες έχουν θεωρηθεί πολλές φορές ασήμαντα.

Η διαδικασία συνεχίζεται έως ότου όλα τα χαρακτηριστικά να ταξινομηθούν ως σημαντικά ή ασήμαντα ή έως ότου επιτευχθεί ένας καθορισμένος αριθμός επαναλήψεων. Το αποτέλεσμα είναι ένα υποσύνολο χαρακτηριστικών που έχει επικυρωθεί στατιστικά ότι έχουν περισσότερη προγνωστική ισχύ από την τυχαία πιθανότητα που προκύπτει από τα χαρακτηριστικά σκιάς, διασφαλίζοντας έτσι, ότι είναι πραγματικά σχετικά και συνεισφέρουν στην απόδοση του μοντέλου.

2.13.3 Πεδία εφαρμογής

Πρόβλεψη κατανάλωσης ενέργειας για τη θέρμανση κτιρίων

Η επιλογή χαρακτηριστικών έχει ευρεία χρήση σε πολλές πτυχές της ανάλυσης δεδομένων. Σε έρευνα που αναπτύχθηκε στον τομέα της θέρμανσης κτιρίων και ειδικότερα στην πρόβλεψη της κατανάλωσης [53], ενσωματώθηκε σαν πρωταρχικό βήμα ο αλγόριθμος Boruta. Η ενσωμάτωση του αλγορίθμου σαν βήμα προεπεξεργασίας αποδείχθηκε κρίσιμης σημασίας για την απλούστευση του συνόλου δεδομένων, αντιμετωπίζοντας έτσι προκλήσεις όπως ασυνεπή χαρακτηριστικά, θόρυβο και ακραίες

τιμές. Η αποτελεσματικότητα του αλγόριθμου Boruta να επιλέγει όλα τα σημαντικά χαρακτηριστικά, διασφαλίζοντας ότι διατηρούνται μόνο τα χαρακτηριστικά που σχετίζονται με τη μεταβλητή απόφασης, συντέλεσαν στην αύξηση της απόδοσης του τελικού μοντέλου και οδήγησαν σε καλύτερη προσαρμογή του μοντέλου σε πραγματικά δεδομένα εκτός του συνόλου εκπαίδευσης.

Βιολογία

Σε έρευνα που πραγματοποιήθηκε σχετικά με την αποτελεσματικότητα του αλγορίθμου Boruta [54] εξήχθησαν θετικά αποτελέσματα και τονίστηκε η αποτελεσματικότητα της μεθόδου. Η έρευνα επικεντρώθηκε στον εντοπισμό σημαντικών μοτίβων αλληλουχιών εντός σύντομων αλληλουχιών RNA. Η ανάλυση των δεδομένων μέσω του αλγορίθμου Boruta έδειξε υψηλό βαθμό συνέπειας μεταξύ των μοτίβων που επιλέχθηκαν και των πειραματικών δεδομένων, υπογραμμίζοντας την ακρίβεια και την αποδοτικότητα του αλγορίθμου. Η μελέτη υπογραμμίζει επίσης την αποτελεσματικότητα του αλγορίθμου Boruta στον εντοπισμό σχετικών μοτίβων ακολουθιών, αποδεικνύοντας έτσι τη χρησιμότητα του στον τομέα της βιοπληροφορικής και της μοριακής βιολογίας.

Ιατρική

Στον τομέα της ιατρικής ο αλγόριθμος Boruta έχει χρησιμοποιηθεί για την έγκαιρη πρόβλεψη της νόσου αλτσχάιμερ. Ειδικότερα, σε σχετική έρευνα πάνω στην πρόβλεψη της νόσου μέσω αλγορίθμων μηχανικής μάθησης [55], η εφαρμογή της μεθόδου σαν βήμα προεπεξεργασίας οδήγησε σε αύξηση της απόδοσης των τελικών μοντέλων πρόβλεψης. Ο αλγόριθμος Boruta έπαιξε καθοριστικό ρόλο στον εντοπισμό των σημαντικών χαρακτηριστικών του συνόλου δεδομένων. Τα επιλεγμένα χαρακτηριστικά συνέβαλαν καθοριστικά στην ενίσχυση της προγνωστικής ακρίβειας των μοντέλων μηχανικής μάθησης που αξιολογήθηκαν στη μελέτη, καταλήγοντας στο συμπέρασμα ότι ο αλγόριθμος Boruta σαν βήμα προεπεξεργασίας αποτελεί μια αξιόπιστη προσέγγιση για την έγκαιρη πρόβλεψη της νόσου του Αλτσχάιμερ.

2.14 t-distributed stochastic neighbor embedding

2.14.1 Θεωρητική ανάλυση

Η μέθοδος μείωσης διαστάσεων στοχαστική ενσωμάτωση γειτόνων τ-κατανομής, t-distributed stochastic neighbor embedding (t-SNE) αποτελεί ένα χρήσιμο εργαλείο στην ανάλυση δεδομένων και ειδικότερα στον τομέα της μηχανικής μάθησης. Η μέθοδος είναι ιδιαίτερα αποτελεσματική στην οπτικοποίηση δεδομένων υψηλών διαστάσεων σε δύο ή τρεις διαστάσεις. Ευρεία χρήση της μεθόδου παρουσιάζεται σε διάφορους τομείς της επιστήμης και ειδικότερα εκεί που υπάρχει η ανάγκη οπτικοποίησης περίπλοκων και σύνθετων συνόλων δεδομένων διατηρώντας την τοπική δομή αυτών σε έναν χώρο λιγότερων διαστάσεων. Η βασική αρχή της μεθόδου είναι η καταγραφή των ομοιοτήτων μεταξύ των χαρακτηριστικών και η μετατροπή αυτών σε κοινές πιθανότητες (joint probabilities). Έπειτα ελαχιστοποιείται η απόκλιση Kullback-Leibler (KL) μεταξύ αυτών των joint probabilities και εκείνων που υπολογίζονται στον χώρο λιγότερων διαστάσεων. Μέσω αυτής της διαδικασίας η μέθοδος διατηρεί αποτελεσματικά τις τοπικές ομοιότητες μεταξύ των σημείων, ενώ τα ενσωματώνει σε ένα χώρο λιγότερων διαστάσεων, συνήθως δύο ή τριών, κατάλληλο για οπτικοποίηση.

2.14.2 Μαθηματική ανάλυση

Η μέθοδος απαρτίζεται από τέσσερα διαδοχικά βήματα. Το πρώτο βήμα αφορά τον υπολογισμό των ομοιοτήτων μεταξύ των στοιχείων στον χώρο πολλών διαστάσεων. Για οποιαδήποτε δύο σημεία x_i και x_j στον αρχικό χώρο υψηλών διαστάσεων, η ομοιότητα τους μοντελοποιείται από μια υπό όρους πιθανότητα $p_{j|i}$, η οποία αντιπροσωπεύει την πιθανότητα ότι το x_i θα επέλεγε το x_j ως γείτονα του εάν οι γείτονες επιλέγονταν σε αναλογία με την πυκνότητα πιθανοτήτων κάτω από μια κανονική κατανομή με κέντρο στο x_i. Μαθηματικά, αυτό ορίζεται ως:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

όπου σ_i είναι η διακύμανση της κανονικής κατανομής με κέντρο στο σημείο δεδομένων x_i.

Στη συνέχεια γίνεται η μετατροπή των ομοιοτήτων και η κατανομή τους σε πιθανότητες. Ο υπολογισμός της joint probability p_{ij} ορίζεται ως ο μέσος όρος των πιθανοτήτων των όρων $p_{j|i}$ και $p_{i|j}$, και δίνεται από τον εξής τύπο:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

όπου N είναι ο συνολικός αριθμός των στοιχείων του συνόλου δεδομένων.

Το επόμενο βήμα αφορά τον χώρο μειωμένων διαστάσεων. Σε αυτό το βήμα η ομοιότητα μεταξύ δύο σημείων y_i και y_j που αντιστοιχούν στα σημεία του χώρου υψηλών διαστάσεων x_i και x_j αντιπροσωπεύεται με παρόμοιο τρόπο μέσω των joint probabilities q_{ij}. Η διαφορά μεταξύ των δύο τρόπων υπολογισμού των joint probabilities μεταξύ του χώρου υψηλών διαστάσεων και του χώρου μειωμένων διαστάσεων έγκειται στο γεγονός ότι στον χώρο μειωμένων διαστάσεων η κανονική κατανομή αντικαθίσταται από μια κατανομή t Student με έναν βαθμό ελευθερίας (που μοιάζει με κατανομή Cauchy)

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

Έτσι αποφεύγεται το πρόβλημα του συνωστισμού των στοιχείων στον χώρο μειωμένων διαστάσεων.

Τέλος η μέθοδος t-SNE επιδιώκει την ελαχιστοποίηση της διαφοράς μεταξύ του χώρου υψηλών διαστάσεων και του χώρου των μειωμένων διαστάσεων. Ειδικότερα προσπαθεί να ελαχιστοποιήσει την απόκλιση Kullback-Leibler (KL) μεταξύ των κοινών κατανομών πιθανότητας P στον υψηλών διαστάσεων χώρο και του Q στον χώρο των μειωμένων διαστάσεων:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

2.14.3 Πεδία εφαρμογής

Η μέθοδος t-SNE επικεντρώνεται κατά κύριο λόγο στη μείωση διαστάσεων ενός συνόλου δεδομένων με σκοπό την οπτικοποίηση των δεδομένων.

Βιολογία και γενετική

η μέθοδος t-SNE αποτελεί ένα σημαντικό και χρήσιμο εργαλείο οπτικοποίησης σε πολλούς τομείς της επιστήμης. Σε έρευνα που έγινε σχετικά με την ανάλυση ανθρώπινων γενετικών δεδομένων [56] και ειδικότερα εστιάζοντας στην αποκάλυψη πολύπλοκων γενετικών δομών, οι ερευνητές εντόπισαν ότι η μέθοδος t-SNE μπορεί να διαφοροποιήσει τα δείγματα από διαφορετικές ηπείρους, παρέχοντας μια σαφή οπτική αναπαράσταση της διαστρωμάτωσης του πληθυσμού. Επίσης στην έρευνα έγινε σύγκριση της μεθόδου με παραδοσιακές μεθόδους μείωσης διαστάσεων όπως την PCA και παρατηρήθηκε ότι έχει μεγαλύτερη ευρωστία στο χειρισμό των ακραίων τιμών. Επιπλέον τονίστηκε η δυνατότητα της μεθόδου στη καταγραφή των τοπικών χαρακτηριστικών των δεδομένων διατηρώντας παράλληλα τις παγκόσμιες δομές των δεδομένων.

2.14.4 Σεισμική μηχανική

Στον τομέα της σεισμικής μηχανικής σε σχετική έρευνα που πραγματοποιήθηκε [57] καταδεικνύεται η ικανότητα της μεθόδου t-SNE στην αποτελεσματική μείωση των διαστάσεων πολύπλοκων συνόλων δεδομένων σεισμικής μηχανικής με στόχο τη βελτιωμένη οπτικοποίηση και ερμηνεία για την ανάλυση των προβλέψεων σεισμικής ζημιάς. Σύμφωνα με την έρευνα τονίζεται η χρησιμότητα της μεθόδου στον χειρισμό μη ισορροπημένων συνόλων δεδομένων, κάτι που συμβαίνει συχνά στον συγκεκριμένο τομέα καθώς ο αριθμός των προσομοιώσεων που υποδεικνύουν ασφαλείς συνθήκες υπερβαίνει σημαντικά εκείνους που προβλέπουν αστοχία

2.14.5 Πρότυπα ανθρώπινης δραστηριότητας

Σε σχετική έρευνα που αφορά τη βελτίωση της απεικόνισης και της ανάλυσης δεδομένων αισθητήρων που συλλέγονται από έξυπνα σπίτια [58] χρησιμοποιήθηκε ο αλγόριθμος t-SNE. Οι ερευνητές πρότειναν μια τροποποιημένη προσέγγιση της μεθόδου με σκοπό την αντιμετώπιση της μη ντετερμινιστικής φύσης της, λόγω της μη κυρτής συνάρτησης κόστους. Τα αποτελέσματα της έρευνας έδειξαν ότι η μέθοδος t-SNE αποτελεί ένα αποτελεσματικό εργαλείο για τη μείωση των διαστάσεων σύνθετων συνόλων δεδομένων και την οπτικοποίηση προτύπων ανθρώπινης δραστηριότητας.

2.15 Factor analysis

2.15.1 Θεωρητική ανάλυση

Η παραγοντική ανάλυση είναι μια στατιστική και εύρεως γνωστή μέθοδος που χρησιμοποιείται σε πολλούς τομείς της επιστήμης, καθώς και σαν εργαλείο μείωσης διαστάσεων. Στις αρχές του 1900, με την ανάπτυξη του συντελεστή συσχέτισης Pearson τα θεμέλια για την ανάπτυξη της παραγοντικής ανάλυσης είχαν τεθεί. Η βασική ιδέα στην οποία βασίζεται η παραγοντική ανάλυση είναι ότι πολλά δεδομένα παράγονται από μερικούς υποκείμενους παράγοντες που δεν είναι άμεσα μετρήσιμοι σε ένα σύνολο δεδομένων. κατ' επέκταση ένα σύνολο δεδομένων κρύβει χαρακτηριστικά τα οποία επηρεάζουν τις παρατηρούμενες μεταβλητές του. Κύριος σκοπός της παραγοντικής ανάλυσης είναι ο εντοπισμός αυτών των κρυφών μοτίβων εντοπίζοντας έτσι έναν μικρότερο αριθμό αόρατων παραγόντων που επηρεάζουν αυτά τα χαρακτηριστικά.

2.15.2 Μαθηματική ανάλυση

Θεωρώντας ότι υπάρχουν τα χαρακτηριστικά ενός συνόλου δεδομένων τα οποία συμβολίζονται $X_1, X_2, ..., X_n$, και έστω ότι αυτά τα χαρακτηριστικά παρουσιάζουν γραμμικές συσχετίσεις μεταξύ τους οι οποίες χαρακτηρίζονται ως $F_1, F_2, ..., F_m$ συν ένα μικρό μέρος κάθε μεταβλητής το οποίο αντιπροσωπεύει τη διακύμανση του σφάλματος, τότε η σχέση μπορεί να αναπαρασταθεί ως

$$X = \lambda F + \Psi$$

Όπου:

- X είναι ο πίνακας των χαρακτηριστικών,
- λ είναι ο πίνακας ιδιοδιανυσμάτων (loadings), τα οποία αντιπροσωπεύουν τις σχέσεις των παραγόντων και των παρατηρούμενων μεταβλητών, υποδεικνύοντας τον βαθμό στον οποίο ένας παράγοντας εξηγεί τη διακύμανση σε μια παρατηρούμενη μεταβλητή.
- F ο πίνακας των λανθάνων παραγόντων.

- Ψ (psi) είναι το διάνυσμα του σφάλματος που σχετίζεται με κάθε παρατηρούμενη μεταβλητή.

Για την εξαγωγή των παραγόντων το πρώτο βήμα της μεθόδου είναι ο υπολογισμός του πίνακα συσχετίσεων μεταξύ των χαρακτηριστικών του συνόλου δεδομένων. Για τον υπολογισμό του πίνακα αξιοποιείται ο συντελεστής συσχέτισης του Pearson ο οποίος δίνεται από τον τύπο :

$$\rho_{XY} = \frac{E[(X - E(X))(Y - E(Y))]}{\sqrt{E[(X - E(X))^2]E[(Y - E(Y))^2]}}$$

Το επόμενο βήμα είναι η εξαγωγή των παραγόντων από τον πίνακα συσχετίσεων. Το συγκεκριμένο βήμα μπορεί να πραγματοποιηθεί με διάφορους τρόπους, οι πιο γνωστοί μέθοδοι είναι η εφαρμογή της μεθόδου PCA ή μέσω της μεθόδου SVD.

Σύμφωνα με τη μέθοδο, το προτελευταίο βήμα είναι η περιστροφή των παραγόντων με σκοπό την επίτευξη κάποιας πιο απλής διάταξης. Η περιστροφή μπορεί να είναι είτε ορθογώνια, διασφαλίζοντας έτσι ότι οι παράγοντες θα παραμείνουν ασυσχέτιστοι είτε πλάγια όπου οι παράγοντες θα συσχετίζονται.

Το τελικό βήμα είναι η επιλογή του αριθμού των παραγόντων που θα διατηρηθούν και ο μετασχηματισμός των δεδομένων βάση αυτών των παραγόντων. Ειδικότερα, για τον μετασχηματισμό των δεδομένων ισχύει ο τύπος:

$$F = X\Lambda^*$$

Όπου:

- Το F αντιπροσωπεύει τον πίνακα των βαθμολογιών παραγόντων.
- X είναι ο πίνακας των παρατηρούμενων μεταβλητών.
- Λ^* είναι ο πίνακας παραγόντων μετά την περιστροφή (εάν εφαρμόζεται περιστροφή).

2.15.3 Πεδία εφαρμογής

Βιοπληροφορική

Στον τομέα της βιοπληροφορικής υπάρχουν σύνολα δεδομένων που απαρτίζονται από εκατοντάδες χαρακτηριστικά. Για την αξιοποίηση αυτών των συνόλων δεδομένων, την ανάλυση και την κατασκευή αξιόπιστων και χρήσιμων μοντέλων μηχανικής μάθησης απαιτείται η εφαρμογή μεθόδων μείωσης διαστάσεων. Στο άρθρο [59] γίνεται εφαρμογή της μεθόδου Factor analysis σε σύνολα δεδομένων που σχετίζονται με τη λευχαιμία. Η συγκεκριμένη μέθοδος εφαρμόζεται για την ανακάλυψη της μοναδικότητας μεταξύ των χαρακτηριστικών, εξάγοντας τα πιο σημαντικά χαρακτηριστικά από το πολυδιάστατο σύνολο δεδομένων. Βοηθά στον εντοπισμό παραγόντων που εξηγούν τις παρατηρούμενες αποκλίσεις μεταξύ των μεταβλητών, απλοποιώντας την πολυπλοκότητα των δεδομένων. Τα αποτελέσματα του άρθρου υπογραμμίζουν την αποτελεσματικότητα της μεθόδου και τονίζουν τη βελτίωση της επίδοσης των τελικών μοντέλων μηχανικής μάθησης.

Ιατρική

Στον τομέα της ιατρικής η χρήση των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών είναι ένα απαραίτητο βήμα για την ανάλυση και εξαγωγή συμπερασμάτων από τα πολύπλοκα ιατρικά δεδομένα. Στην έρευνα [60] η μέθοδος Factor Analysis χρησιμοποιήθηκε με ιδιαίτερη αποτελεσματικότητα στη διάγνωση της νόσου του Πάρκινσον. Ειδικότερα, η έρευνα επικεντρώθηκε στην ανάλυση της δυσφωνίας, μιας μορφής διαταραχής του λόγου που οφείλεται στη νόσο του Πάρκινσον. Τα δεδομένα ομιλίας είναι πολυδιάστατα με αποτέλεσμα τα μοντέλα μηχανικής μάθησης να επηρεάζονται από την κατάρα της διαστατότητας. Η εφαρμογή της Factor Analysis για τη μείωση των διαστάσεων των δεδομένων αποδείχθηκε αποτελεσματική και συντέλεσε και στη βελτίωση της απόδοσης του τελικού μοντέλου μηχανικής μάθησης.

2.16 Laplacian Eigenmaps

2.16.1 Θεωρητική ανάλυση

Οι λαπλασιανοί ιδιοχάρτες (Laplacian Eigenmaps) είναι μια πολύ γνωστή μη γραμμική μέθοδος μείωσης διαστάσεων στον τομέα της μηχανικής μάθησης και ανάλυσης δεδομένων. Είναι μια μη εποπτευόμενη μέθοδος, που σαν βασικό στόχο έχει τη διατήρηση της τοπικής δομής των δεδομένων. Η χαρτογράφηση των δεδομένων υψηλών διαστάσεων σε έναν χώρο λιγότερων διαστάσεων γίνεται με τέτοιο τρόπο ώστε τα στοιχεία που βρίσκονται κοντά το ένα στο άλλο στον υψηλών διαστάσεων χώρο να παραμένουν κοντά στην αναπαράσταση μειωμένων διαστάσεων. Αυτήν η προσέγγιση στοχεύει στην καταγραφή και διατήρηση των εγγενών γεωμετρικών και τοπολογικών ιδιοτήτων του συνόλου δεδομένων στον χώρο λιγότερων διαστάσεων.

2.16.2 Μαθηματική ανάλυση

Η μέθοδος αποτελείται από πέντε βήματα. Το πρώτο βήμα είναι η κατασκευή του πίνακα των σχέσεων μεταξύ των δεδομένων $G = (V, E)$. Έστω ένα σύνολο δεδομένων $X = \{x_1, x_2, ..., x_n\}$ όπου κάθε x_i βρίσκεται στον χώρο υψηλών διαστάσεων. Τότε η κατασκευή του πίνακα $G = (V, E)$ καθορίζεται είτε από τον κανόνα K (καθορίζοντας τον αριθμό των γειτόνων για κάθε σημείο), είτε από τον κανόνα ε (που συνδέει όλα τα σημεία σε μια σταθερή ακτίνα ε). Για την αρχικοποίηση των βαρών, μεταξύ των σημείων χρησιμοποιείται το Heat Kernel $W_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{2t})$ για συνδεδεμένες κορυφές v_i και v_j, και $W_{ij} = 0$ για μη συνδεδεμένες κορυφές. Η παράμετρος t ελέγχει το πλάτος της γειτονιάς. Ο συγκεκριμένος πίνακας περιέχει όλες τις πληροφορίες που αφορούν την τοπολογία των στοιχείων σε σχέση με τα γειτονικά στοιχεία.

Το επόμενο βήμα είναι η εξαγωγή του λαπλασιανού πίνακα. Πραγματοποιείται ο υπολογισμός του πίνακα μοιρών D, ο οποίος είναι ένας διαγώνιος πίνακας όπου κάθε στοιχείο D_{ii} είναι το άθροισμα των βαρών όλων των ακμών που συνδέονται με την κορυφή v_i , $D_{ii} = \sum_j W_{ij}$. Ο λαπλασιανός πίνακας εξάγεται μέσω του τύπου $L = D - W$ και ουσιαστικά καταγράφει τη διαφορά μεταξύ του βαθμού μιας κορυφής και του αθροίσματος των βαρών των συνδεδεμένων ακμών της, ενσωματώνοντας έτσι την τοπολογία του γραφήματος.

Το τρίτο βήμα αποτελεί την επίλυση του προβλήματος ιδιοτιμών.

$$L\mathbf{u} = \lambda D\mathbf{u}$$

Από την παραπάνω εξίσωση εξάγονται οι ιδιοτιμές λ και τα αντίστοιχα ιδιοδιανύσματα τους $\mathbf{u}$. Οι ιδιοτιμές σε αυτό το βήμα είναι ταξινομημένες από τη μικρότερη στη μεγαλύτερη και ισχύει ότι τα ιδιοδιανύσματα που σχετίζονται με τις μικρότερες μη μηδενικές ιδιοτιμές παρέχουν τις συντεταγμένες στο χώρο μειωμένων διαστάσεων.

Το τελευταίο βήμα είναι ο μετασχηματισμός των δεδομένων στον νέο χώρο λιγότερων διαστάσεων. Σε αυτό το βήμα επιλέγονται οι k πρώτες ιδιοτιμές και τα αντίστοιχα ιδιοδιανύσματα τους. Συνήθως, η πρώτη ιδιοτιμή αγνοείται καθώς είναι πάντα πολύ κοντά στο μηδέν. Μέσω των ιδιοδιανυσμάτων που αντιστοιχούν στις επόμενες ιδιοτιμές εξάγονται τα μετασχηματισμένα δεδομένα στον νέο χώρο λιγότερων διαστάσεων.

2.16.3 Πεδία εφαρμογής

Σαν μέθοδος μείωσης διαστάσεων η Laplacian Eigenmaps έχει ευρεία εφαρμογή σε πολλούς τομείς της επιστήμης. Χρησιμοποιείται τόσο για απλοποίηση των συνόλων δεδομένων μειώνοντας έτσι τον χρόνο εκπαίδευσης των τελικών μοντέλων μηχανικής μάθησης όσο και για τη βελτίωση της ποιότητας των δεδομένων με αποτέλεσμα την αύξηση της αποτελεσματικότητας των τελικών μοντέλων πρόβλεψης.

Ταξινόμηση πολωσιμετρικών συνθετικών εικόνων

Στον τομέα της ταξινόμησης εικόνων και ειδικότερα σε ένα σύνολο δεδομένων με πολωσιμετρικές συνθετικές εικόνες ραντάρ διαφράγματος κάλυψη γης (PolSAR) εφαρμόστηκε η μέθοδος σαν βήμα προεπεξεργασίας. Τα μετασχηματισμένα δεδομένα λειτούργησαν ευεργετικά στον ταξινομητή αυξάνοντας την αποτελεσματικότητα των τελικών προβλέψεων, τονίζοντας έτσι τα θετικά αποτελέσματα και την αποτελεσματικότητα της μεθόδου [61].

Οπτικοποίηση ιχνών ινών στον ανθρώπινο εγκέφαλο

Σε έρευνα που πραγματοποιήθηκε πάνω σε δεδομένα μαγνητικού συντονισμού τανυστή διάχυσης (DT-MRI) αξιοποιήθηκε η μέθοδος Laplacian Eigenmaps για τη μείωση των διαστάσεων του χώρου δεδομένων με κύριο σκοπό την αναπαράσταση των ινών στον τρισδιάστατο χρωματικό χώρο RGB. Η αναπαράσταση αποσκοπεί στην ενίσχυση της αντίληψης των δεσμίδων ινών και τη συνδεσιμότητα μέσα στον ανθρώπινο εγκέφαλο. Η μέθοδος Laplacian Eigenmaps βοήθησε ιδιαίτερα στη συγκεκριμένη έρευνα μέσω της ικανότητας της να διατηρεί τη γειτονική δομή των δεδομένων όταν αυτά μετασχηματίζονται σε ένα χώρο λιγότερων διαστάσεων. Ως αποτέλεσμα, όταν έγινε η μεταφορά στον χώρο RGB παρόμοια ίχνη τοποθετήθηκαν το ένα κοντά στο άλλο στον χρωματικό χώρο RGB. Τα αποτελέσματα της έρευνας δείχνουν ότι η εφαρμογή της μεθόδου συντελεί στη βελτίωση της αντίληψης των δεσμίδων ινών και στην καλύτερη οπτικοποίηση της συνδεσιμότητας του εγκεφάλου, υπογραμμίζοντας την αποτελεσματικότητα της μεθόδου στην αναπαράσταση δεδομένων υψηλών διαστάσεων σε χώρους λιγότερων διαστάσεων [62].

Αναγνώριση προσώπου

Η εγγενής ικανότητα της μεθόδου Laplacian Eigenmaps να διατηρεί την τοπική δομή των δεδομένων κατά τη μείωση διαστάσεων αποτελεί ένα σημαντικό πλεονέκτημα έναντι κλασσικών μεθόδων όπως η PCA ειδικότερα, σε τομείς που απαιτείται οπτικοποίηση ή μεταφορά συγκεκριμένων χαρακτηριστικών από τον χώρο υψηλών διαστάσεων στον νέο χώρο λιγότερων διαστάσεων. Σε σχετική έρευνα [63] σχετικά με την αναγνώριση προσώπων μια προσέγγιση της μεθόδου Laplacian Eigenmaps η οποία ονομάζεται Incremental Laplacian Eigenmaps, έδειξε αξιοσημείωτα αποτελέσματα και συγκριτικά με την PCA και τη Locality preserving projections (LPP) έδειξε μεγαλύτερη αποτελεσματικότητα στην αναγνώριση. Αυτό τονίζει τα πλεονεκτήματα της διατήρησης της τοπικής δομής των δεδομένων, ειδικά σε σύνολα δεδομένων πολλών διαστάσεων όπως οι εικόνες προσώπων.

2.17 Πλεονεκτήματα και μειονεκτήματα μεθόδων

Πίνακας 2.1: Πίνακας πλεονεκτημάτων και μειονεκτημάτων

	Πλεονεκτήματα	Μειονεκτήματα
PCA	• Απλοποιεί τη διαδικασία ανάλυσης των δεδομένων διατηρώντας τα χαρακτηριστικά που έχουν τη μεγαλύτερη διακύμανση. • Μειώνει τον αριθμό των διαστάσεων του συνόλου δεδομένων, με αποτέλεσμα το σύνολο δεδομένων να γίνεται πιο ερμηνεύσιμο και διαχειρίσιμο χωρίς να επιφέρει σημαντική απώλεια πληροφοριών. • Μετασχηματίζει τα δεδομένα με βάση τις πρωταρχικές συνιστώσες και βοηθά στην αποκάλυψη μοτίβων που μπορεί να μην είναι άμεσα εμφανή, διευκολύνοντας την καλύτερη κατανόηση των υποκείμενων δομών του συνόλου δεδομένων. • Φιλτράρει αποτελεσματικά το θόρυβο από το σύνολο δεδομένων απομονώνοντας τα στοιχεία που καταγράφουν τη μεγαλύτερη διακύμανση, τα οποία συχνά αντιπροσωπεύουν το σήμα και όχι το θόρυβο.	• Καταγράφει γραμμικές σχέσεις μεταξύ των δεδομένων. Πολλά σύνολα δεδομένων έχουν εγγενείς μη γραμμικές σχέσεις, οδηγώντας δυνητικά σε χάσιμο σημαντικής πληροφορίας. • Εξισώνει τη διακύμανση με την πληροφορία, με αποτέλεσμα να δίνει προτεραιότητα στις κατευθύνσεις με τη μέγιστη διακύμανση παραβλέποντας άλλα λιγότερο μεταβλητά χαρακτηριστικά τα οποία μπορεί να είναι σημαντικά, οδηγώντας σε απώλεια σχετικών πληροφοριών. • Παρουσιάζει ευαισθησία σε ακραίες τιμές. Οι ακραίες τιμές μπορούν να επηρεάσουν σημαντικά την κατεύθυνση και τη διακύμανση που βάση της οποίας υπολογίζονται οι πρωταρχικές συνιστώσες, οδηγώντας σε μη αποτελεσματική μείωση των διαστάσεων. • Παρουσιάζει μεγάλη ευαισθησία στην κλιμάκωση των τιμών των χαρακτηριστικών.Τα χαρακτηριστικά με μεγαλύτερες κλίμακες κυριαρχούν έναντι εκείνων με μικρότερες κλίμακες.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
SVD	• Οδηγείται στο αποτέλεσμα αποκλειστικά από τα δεδομένα, είναι σταθερή μέθοδος, αρκετά ευέλικτη και μπορεί να εφαρμοστεί ανεξάρτητα από τις διαστάσεις του αρχικού πίνακα [15]. Επομένως, αποτελεί ένα πολύ χρήσιμο εργαλείο. • Η ανάλυση του αρχικού πίνακα στους επιμέρους τρεις πίνακες και η ταξινόμηση των ιδιοδιανυσμάτων βάση των ιδιοτιμών μπορεί να διαχωρίσει το σήμα από το θόρυβο [64], βελτιώνοντας την ποιότητα των δεδομένων και καθιστώντας την περαιτέρω ανάλυση πιο ακριβή και ουσιαστική. • Εντοπίζοντας και εξαλείφοντας τα λιγότερο σημαντικά χαρακτηριστικά του συνόλου δεδομένων, επιτυγχάνεται συμπίεση των δεδομένων χωρίς να χάνεται σημαντικό μέρος της πληροφορίας [65]. • Παρέχει την καλύτερη προσέγγιση χαμηλής κατάταξης ενός πίνακα ως προς το σφάλμα ελαχίστων τετραγώνων [15]. Αυτή η ιδιότητα είναι ιδιαίτερα ωφέλιμη στη μείωση διαστάσεων, όπου είναι επιθυμητή η διατήρηση των πιο σημαντικών χαρακτηριστικών των δεδομένων με ελάχιστη απώλεια πληροφοριών.	• Το υπολογιστικό κόστος εκτέλεσης της μεθόδου αυξάνεται αρκετά όσο αυξάνονται οι διαστάσεις του πίνακα του συνόλου δεδομένων. • Πολλές φορές, ειδικά σε πολύπλοκα σύνολα δεδομένων η ανάλυση του πίνακα του συνόλου δεδομένων στους επιμέρους τρεις πίνακες δεν εγγυάται ότι θα αποφέρει κάποια φυσική ή πρακτική ερμηνεία. • Δεν είναι πολύ αποτελεσματική μέθοδος για αραιούς πίνακες στοιχείων. • Λόγω της διάσπασης του πίνακα δεδομένων σε τρεις πίνακες, μπορεί να προκληθούν προβλήματα μνήμης, ειδικά σε πολύ μεγάλα σύνολα δεδομένων.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
Isomap	• Εγγυάται υπολογιστική αποτελεσματικότητα και ασυμπτωτική σύγκλιση [66] καθώς και παγκόσμια βελτιστοποίηση (global optimality) [67]. • Καταγράφει μη γραμμικές σχέσεις μεταξύ των στοιχείων του συνόλου δεδομένων, κάτι το οποίο καθιστά τη μέθοδο ιδιαίτερα χρήσιμη όταν γραμμικές μέθοδοι όπως η PCA δεν μπορούν να λειτουργήσουν αποτελεσματικά. • Επικεντρώνεται στη διατήρηση της συνολικής (Global) δομής δεδομένων λαμβάνοντας υπόψη τις γεωδαισιακές αποστάσεις. Αυτό έχει ως αποτέλεσμα να προσφέρει ικανοποιητικά αποτελέσματα ακόμη και σε σύνολα δεδομένων με περίπλοκα μοτίβα. • Είναι κατάλληλη για manifold learning καθώς βοηθά στην αποκάλυψη της υποκείμενης δομής των συνόλων δεδομένων που βρίσκονται σε μια πολλαπλή, όπως σχήματα, εικόνες και δεδομένα αισθητήρων.	• Παρουσιάζει ευαισθησία στην επιλογή παραμέτρων. Στο πρώτο βήμα της μεθόδου γίνεται η καταγραφή της συνδεσιμότητας των στοιχείων με βάση τους κοντινότερους γείτονες στον χώρο υψηλών διαστάσεων. Η συγκεκριμένη διαδικασία είναι ευαίσθητη στην επιλογή του αριθμού των κοντινότερων γειτόνων. Μια υπερβολικά ευρεία γειτονιά μπορεί να προκαλέσει σφάλματα βραχυκυκλώματος, κυρίως σε περιοχές όπου η πολλαπλή διπλώνει πάνω στον εαυτό της [68]. • Επηρεάζεται σε μεγάλο βαθμό από τον θόρυβο των δεδομένων. Κατά τον υπολογισμό των κοντινότερων γειτόνων η ύπαρξη θορύβου επηρεάζει το τελικό αποτέλεσμα προκαλώντας πολλές φορές σφάλματα βραχυκυκλώματος, όπου ο αλγόριθμος λανθασμένα υποθέτει μια κοντινή γεωδαιτική απόσταση μεταξύ σημείων που δεν είναι πραγματικά κοντά στην πολλαπλή [68] [66]. • Βασίζεται στην ακριβή συνδεσιμότητα των στοιχείων κατά το πρώτο βήμα της μεθόδου και η απουσία της συνεπάγεται σε τοπολογική αστάθεια [69].

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
Ensemble Learning	• Τα μοντέλα εκμάθησης συνόλου είναι ανθεκτικά στο θόρυβο και στις ακραίες τιμές, γεγονός που τα καθιστά αξιόπιστα για την επιλογή των σημαντικότερων χαρακτηριστικών. • Η επιλογή των χαρακτηριστικών γίνεται με βάση τις μετρικές που χρησιμοποιούν περισσότερο τα μοντέλα για την κατασκευή των δέντρων τους. Επομένως τα χαρακτηριστικά που επιλέγουν συμβάλλουν όντως στη διαδικασία λήψης αποφάσεων. • Είναι ανθεκτικά στην υπερπροσαρμογή παρέχοντας έτσι πιο αξιόπιστα αποτελέσματα σχετικά με τη σημαντικότητα των χαρακτηριστικών και μειώνοντας τον κίνδυνο επιλογή θορυβωδών χαρακτηριστικών. • Καταγράφουν τόσο γραμμικές όσο και μη γραμμικές σχέσεις, καθιστώντας τα μοντέλα αποτελεσματικά και σε σύνολα δεδομένων με μη γραμμικές σχέσεις μεταξύ των χαρακτηριστικών.	• Τα μοντέλα εκμάθησης συνόλου έχουν συνήθως μεγάλο υπολογιστικό κόστος. • Αντιμετωπίζουν προβλήματα όταν υπάρχει πολυσυγγραμμικότητα στο σύνολο δεδομένων, δηλαδή όταν πολλά χαρακτηριστικά συσχετίζονται σε μεγάλο βαθμό μεταξύ τους. Σε αυτές τις περιπτώσεις, η σημαντικότητα των χαρακτηριστικών κατανέμεται μεταξύ των χαρακτηριστικών και πολλές φορές τα μοντέλα αδυνατούν να πάρουν τις καλύτερες αποφάσεις. • Η ρύθμιση των υπερπαραμέτρων για καλύτερη απόδοση απαιτεί γνώση τόσο του μοντέλου όσο και των χαρακτηριστικών του συνόλου δεδομένων και πολλές φορές είναι δύσκολη διαδικασία.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
ICA	• Έχει τη δυνατότητα να διαχωρίσει τυφλά ένα πολύπλοκο σήμα σε σήματα πηγής. • Παρουσιάζει ανθεκτικότητα στον θόρυβο. Με την προϋπόθεση ότι τα σήματα πηγής είναι στατιστικά ανεξάρτητα, η μέθοδος ICA μπορεί να τα διαχωρίσει από σήματα θορύβου. • Μπορεί να καταγράψει περίπλοκες σχέσεις, καθιστώντας την πιο κατάλληλη για δεδομένα του πραγματικού κόσμου.	• Έχει μεγάλο υπολογιστικό κόστος και όσο μεγαλύτερο είναι το σύνολο δεδομένων αυτό κλιμακώνεται. • Κάνει δύο παραδοχές, ότι τα δεδομένα δεν ακολουθούν την κανονική κατανομή και ότι είναι στατιστικά ανεξάρτητα μεταξύ τους. Όταν στο σύνολο δεδομένων δεν ισχύουν αυτές οι παραδοχές, η αποτελεσματικότητα της μεθόδου μειώνεται. • Απαιτείται γνώση του συνόλου δεδομένων για τον προσδιορισμό του αριθμού των σημάτων προς εξαγωγή και πολλές φορές αυτό είναι μια περίπλοκη διαδικασία. • Η ερμηνεία των εξαγόμενων ανεξάρτητων σημάτων δεν συνεπάγεται απαραίτητα με κάποια ουσιαστική ερμηνεία στο αρχικό σύνολο δεδομένων.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
Factor Analysis	• Απλοποιεί το συνόλο δεδομένων και μείωνει τις διαστάσεις του, καθιστώντας ευκολότερη την ερμηνεία των υποκείμενων δομών και μοτίβων. • Βοηθά στην αντιμετώπιση της πολυσυγγραμμικότητας μεταξύ των χαρακτηριστικών συνδυάζοντας χαρακτηριστικά υψηλής συσχέτισης σε έναν μόνο παράγοντα.	• Χρειάζεται συνήθως μεγάλο αριθμό δειγμάτων για να δώσει αξιόπιστα αποτελέσματα. • Κάνει την υπόθεση ότι οι σχέσεις μεταξύ των χαρακτηριστικών του συνόλου δεδομένων είναι γραμμικές, κάτι που δεν ισχύει στα περισσότερα σύνολα δεδομένων του πραγματικού κόσμου. • Η απόφαση για την κατάλληλη μέθοδο περιστροφής (ορθογώνια ή λοξή) μπορεί να είναι πολύπλοκη και επηρεάζει την ερμηνεία των παραγόντων. Η επιλογή μιας μη κατάλληλης περιστροφής μπορεί να επηρεάσει τη σαφήνεια και τη χρηστικότητα των αποτελεσμάτων. • Βασίζεται στην παρουσία συσχέτισης μεταξύ των δεδομένων, η απουσία της καθιστά τη μέθοδο αναποτελεσματική. • Είναι ευαίσθητη σε ακραίες τιμές οι οποίες μπορούν να παραμορφώσουν τα αποτελέσματα και να οδηγήσουν σε λανθασμένα συμπεράσματα.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
Boruta	• Η ύπαρξη των χαρακτηριστικών σκιάς συντελούν στην αξιολόγηση των χαρακτηριστικών και στον διαχωρισμό τους ανάμεσα σε σημαντικά χαρακτηριστικά και θόρυβο, διασφαλίζοντας ότι τα χαρακτηριστικά που επιλέγονται είναι όντως σημαντικά. • Η επαναληπτική λειτουργία που χρησιμοποιεί ο αλγόριθμος, εξασφαλίζει τη διεξοδική ανάλυση του συνόλου δεδομένων και την εξαγωγή των πιο σημαντικών χαρακτηριστικών. • Οι μέθοδοι εκμάθησης συνόλου που χρησιμοποιούνται από τον αλγόριθμο εξασφαλίζουν την ανθεκτικότητα στην υπερπροσαρμογή και την καταγραφή τόσο γραμμικών όσο και μη γραμμικών σχέσεων.	• Απαιτείται μεγάλη υπολογιστική ισχύς και ο αλγόριθμος λόγω της επαναληπτικής διαδικασίας είναι αρκετά χρονοβόρος. • Η επίδοση του εξαρτάται από τη μέθοδο εκμάθησης συνόλου. Σε περίπτωση που δεν έχουν ρυθμιστεί σωστά οι υπερπαραμέτροι της μεθόδου, ο αλγόριθμος δεν θα είναι αποτελεσματικός. • Οι μέθοδοι εκμάθησης συνόλου αντιμετωπίζουν προβλήματα με την πολυσυγγραμμικότητα, με αποτέλεσμα και ο αλγόριθμος Boruta να επηρεάζεται από αυτήν.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
Laplacian Eigenmaps	• Διατηρεί αποτελεσματικά την τοπική δομή των δεδομένων, καθιστώντας τη μέθοδο χρήσιμο εργαλείο όταν τα δεδομένα έχουν εγγενείς γεωμετρικές ή πολλαπλές δομές. • Έχει τη δυνατότητα καταγραφής μη γραμμικών σχέσεων μεταξύ των δεδομένων. Έχει σχεδιαστεί για να συλλαμβάνει την πολλαπλή δομή στα δεδομένα, παρέχοντας μια διαφοροποιημένη αναπαράσταση. • Η καταγραφή της τοπικής δομής των δεδομένων βοηθάει στην καλύτερη αναπαράσταση της οπτικοποίησης στον χώρο μειωμένων διαστάσεων. Αυτό επιτρέπει στους αναλυτές να εντοπίσουν μοτίβα που δεν είναι εμφανή στον χώρο υψηλότερων διαστάσεων.	• Ο υπολογισμός των ιδιοτιμών και των ιδιοδιανυσμάτων που τελεί η μέθοδος έχει μεγάλο υπολογιστικό κόστος, το οποίο αυξάνεται σε μεγάλο βαθμό όσο μεγαλύτερα είναι τα σύνολα δεδομένων. • Η επιλογή τόσο του αριθμού των γειτόνων όσο και η μετρική για τη μέτρηση των αποστάσεων μεταξύ των στοιχείων επηρεάζει σε μεγάλο βαθμό την ποιότητα της αναπαράστασης των δεδομένων. • Η εφαρμογή της μεθόδου σε νέα στοιχεία δεδομένων (δηλαδή, επεκτάσεις εκτός του συνόλου εκπαίδευσης) δεν είναι απλή και απαιτούνται περίπλοκες τεχνικές για την προσέγγιση των νέων στοιχείων. • Έχει μεγάλες απαιτήσεις σε μνήμη και επομένως είναι δύσκολο να εφαρμοστεί σε μεγάλα σύνολα δεδομένων.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
MDS	• Μπορεί να μετασχηματίσει πολύπλοκα και πολυδιάστατα σύνολα δεδομένων στον δισδιάστατο ή τρισδιάστατο χώρο κάνοντας έτσι ευκολότερη την οπτικοποίηση των δεδομένων και την ερμηνεία των υποκείμενων δομών και μοτίβων που μπορεί να υπάρχουν σε αυτά. • Μπορεί να καταγράψει αποτελεσματικά τόσο γραμμικές όσο και μη γραμμικές σχέσεις μεταξύ των δεδομένων. • Υπολογίζει την ομοιότητα ή την ανομοιότητα μεταξύ των δεδομένων. Επομένως μπορεί να προσαρμοστεί ώστε να λειτουργεί για πολλαπλούς τύπους δεδομένων, γεγονός που την καθιστά χρήσιμο εργαλείο σε πολλές εφαρμογές.	• Κατασκευάζει τον πλήρη πίνακα ομοιοτήτων/ανομοιοτήτων μεταξύ όλων των δεδομένων. Αυτό επιφέρει μεγάλο υπολογιστικό κόστος και δέσμευση μεγάλου ποσού μνήμης. • Η παρουσία ακραίων τιμών μπορεί να παραμορφώσει σημαντικά τα αποτελέσματα της MDS επειδή η μέθοδος προσπαθεί να διατηρήσει τις αποστάσεις μεταξύ όλων των ζευγών σημείων. • Η ερμηνεία των αποτελεσμάτων μπορεί πολλές φορές να είναι δύσκολη, ειδικά όταν η διαμόρφωση δεν έχει σαφές μοτίβο. • Η εφαρμογή της μεθόδου σε νέα στοιχεία δεδομένων (δηλαδή, επεκτάσεις εκτός του συνόλου εκπαίδευσης) δεν είναι απλή και απαιτούνται περίπλοκες τεχνικές για την προσέγγιση των νέων στοιχείων.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
Spearman's Coefficient	• Δεν προϋποθέτει ότι τα στοιχεία ακολουθούν την κανονική κατανομή, γεγονός που την καθιστά αποτελεσματική και σε δεδομένα του πραγματικού κόσμου που συνήθως δεν ακολουθούν την κανονική κατανομή. • Είναι λιγότερο ευαίσθητη σε ακραίες τιμές σε σχέση με τη συσχέτιση Pearson. • Μπορεί να εντοπίζει μονοτονικές σχέσεις. Η ικανότητα εντοπισμού τόσο των αυξανόμενων όσο και των φθίνουσων τάσεων, αποτελεί σημαντικό πλεονέκτημα στη διερευνητική ανάλυση δεδομένων.	• Δεν μπορεί να ανιχνεύσει πιο σύνθετες σχέσεις (όπως τετραγωνικές ή κυβικές σχέσεις) εκτός και αν είναι αυστηρά αυξανόμενες ή φθίνουσες. • Βασίζεται στις τάξεις των δεδομένων, οποιοδήποτε λάθος στην κατάταξη ή την ανάθεση βαθμών μπορεί να επηρεάσει σημαντικά το αποτέλεσμα. Αυτό το ζήτημα επηρεάζει κυρίως σύνολα δεδομένων με πολλές ισόβαθμες βαθμίδες.
Kendall's Coefficient	• Όπως η συσχέτιση Spearman, έτσι και η συσχέτιση του Kendall δεν απαιτεί τα δεδομένα να ακολουθούν μια κανονική κατανομή. • Είναι ανθεκτική έναντι των ακραίων τιμών επειδή βασίζεται στην κατάταξη των δεδομένων και όχι στις πραγματικές τους τιμές. • Συχνά θεωρείται πιο ερμηνεύσιμη από τη συσχέτιση του Spearman επειδή μετράει απευθείας τον αριθμό των σύμφωνων και ασύμφωνων ζευγών μεταξύ των δεδομένων. • Είναι αποτελεσματική και σε μικρά μεγέθη δειγμάτων σε σύγκριση με άλλους συντελεστές συσχέτισης.	• Απαιτεί μεγαλύτερη υπολογιστική ισχύ σε σχέση με τους συντελεστές Pearson και Spearman, ειδικά όσο αυξάνεται το μέγεθος του συνόλου δεδομένων. • Η παρουσία πολλών ισόβαθμων βαθμίδων μπορεί να επηρεάσει την ακρίβεια της μέτρησης συσχέτισης. • Δεν μπορεί να μετρήσει ή να ανιχνεύσει αποτελεσματικά πολύπλοκες μη μονοτονικές σχέσεις. • Για μεγάλα μεγέθη δείγματος, ο συντελεστής συσχέτισης του Kendall μπορεί να είναι λιγότερο αποτελεσματικός στατιστικά από τον συντελεστή συσχέτισης του Pearson όταν ισχύει η υπόθεση της κανονικότητας.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
LDA	• Μεγιστοποιεί την αναλογία της διακύμανσης μεταξύ κλάσεων προς τη διακύμανση εντός κλάσης στο σύνολο δεδομένων. Έτσι, η LDA διασφαλίζει ότι οι κλάσεις είναι όσο το δυνατόν πιο διακριτές, ενισχύοντας θεωρητικά την απόδοση των ταξινομητών. • Δεν απαιτείται μεγάλος αριθμός υπολογιστικών πόρων για τη λειτουργία της μεθόδου, καθιστώντας την κατάλληλη για εφαρμογές σε πραγματικό χρόνο. • Βασίζεται σε υποθέσεις που είναι λιγότερο περιοριστικές σε σύγκριση με άλλες στατιστικές μεθόδους. Υποθέτει ότι τα χαρακτηριστικά ακολουθούν κανονικές κατανομές με ίσους πίνακες συνδιακύμανσης για κάθε κλάση, αν οι υποθέσεις πληρούνται, έστω σε ένα βαθμό και όχι πλήρως, είναι πιθανόν η μέθοδος να δώσει πολύ καλά αποτελέσματα.	• Εάν τα δεδομένα δεν ακολουθούν την κανονική κατανομή, η αποτελεσματικότητα της μεθόδου μειώνεται. • Σε περιπτώσεις όπου ο αριθμός των χαρακτηριστικών είναι σχετικά υψηλός σε σύγκριση με τον αριθμό των παρατηρήσεων (δεδομένα υψηλών διαστάσεων),η μέθοδος LDA μπορεί να υπερπροσαρμοστεί στα δεδομένα και να έχει κακή γενίκευση σε νέα δεδομένα. • Εντοπίζει γραμμικές σχέσεις ανάμεσα στα δεδομένα, γεγονός που την καθιστά αναποτελεσματική σε πολύπλοκα σύνολα δεδομένων με εγγενείς μη γραμμικές σχέσεις. • Παρουσιάζει ευαισθησία σε ακραίες τιμές.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
Kernel PCA	• Μπορεί να καταγράψει μη γραμμικές σχέσεις μεταξύ των δεδομένων. • Προβάλλοντας τα δεδομένα σε ένα χώρο υψηλότερων διαστάσεων μέσω της συνάρτησης πυρήνα, η μέθοδος μπορεί να αποκαλύψει μοτίβα και σχέσεις που δεν είναι εμφανείς στον αρχικό χώρο. • Υποστηρίζει πολλές συναρτήσεις πυρήνα με αποτέλεσμα να μπορεί να προσαρμοστεί στις ιδιότητες και τα χαρακτηριστικά του εκάστοτε συνόλου δεδομένων. • Μπορεί να βοηθήσει στη μείωση του θορύβου, εστιάζοντας στις πρωταρχικές συνιστώσες και απορρίπτοντας τον θόρυβο.	• Το υπολογιστικό κόστος της μεθόδου είναι μεγάλο, καθώς επίσης και οι απαιτήσεις μνήμης για την αποθήκευση του πίνακα υψηλών διαστάσεων μετά την εφαρμογή του τρικ πυρήνα. • Η αποτελεσματικότητα της μεθόδου εξαρτάται σε μεγάλο βαθμό από τη συνάρτηση πυρήνα, επομένως χρειάζεται προσεκτική επιλογή του τύπου της συνάρτησης πυρήνα για την εξαγωγή θετικών αποτελεσμάτων. • Η μετατροπή σε χώρο υψηλών διαστάσεων μέσω της συνάρτησης πυρήνα μπορεί να καταστήσει τα αποτελέσματα της μεθόδου δύσκολο να ερμηνευτούν ως προς τα αρχικά χαρακτηριστικά.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
t-SNE	• Είναι ιδιαίτερα αποτελεσματική μέθοδος για την οπτικοποίηση των δεδομένων σε έναν χώρο λιγότερων διαστάσεων. • Διατηρεί αποτελεσματικά την τοπική δομή των δεδομένων, καθιστώντας την καλή στην καταγραφή και ομαδοποίηση των τοπικών ομοιοτήτων μεταξύ των σημείων. • Καταγράφει τόσο γραμμικές όσο και μη γραμμικές σχέσεις μεταξύ των δεδομένων. • Είναι αποτελεσματική και σε σύνολα δεδομένων με μεγάλο αριθμό διαστάσεων όπου άλλες μέθοδοι δεν είναι τόσο αποτελεσματικές.	• Εχει μεγάλη υπολογιστική πολυπλοκότητα, ειδικά σε μεγάλα σύνολα δεδομένων. Κλιμακώνεται τετραγωνικά με τον αριθμό των σημείων δεδομένων, το οποίο την καθιστά πολύ αργή για μεγάλα σύνολα δεδομένων. • Ένα μεγάλο κομμάτι της αποτελεσματικότητας της μεθόδου προέρχεται από τις υπερπαραμέτρους της, οι οποίες πρέπει να αρχικοποιηθούν σωστά για να είναι αποτελεσματική η μέθοδος. • Δεν διατηρεί τόσο αποτελεσματικά την καθολική δομή των δεδομένων, με αποτέλεσμα να χάνονται πληροφορίες που μπορεί να είναι ζωτικής σημασίας για την ανάλυση των δεδομένων.

Table 2.1 Συνέχεια πίνακα προηγούμενης σελίδας

	Πλεονεκτήματα	Μειονεκτήματα
LLE	• Διατηρεί και μεταφέρει αποτελεσματικά την τοπική δομή των δεδομένων στον χώρο μειωμένων διαστάσεων, γεγονός που την καθιστά αποτελεσματική σε εργασίες σχετικά με την οπτικοποίηση των δεδομένων. • Δεν απαιτείται τα δεδομένα να παρουσιάζουν γραμμικές σχέσεις μεταξύ τους. Αντίθετα, μπορεί να καταγράψει αποτελεσματικά ακόμα και πολύπλοκες μη γραμμικές σχέσεις που μπορεί να έχουν τα δεδομένα. • Εστιάζει στην τοπική καταγραφή της δομής των δεδομένων, το οποίο καθιστά τη μέθοδο ανθεκτική στο θόρυβο και τις ακραίες τιμές. • Είναι αποτελεσματική και σε σύνολα δεδομένων με μεγάλο αριθμό διαστάσεων όπου άλλες μέθοδοι δεν είναι τόσο αποτελεσματικές.	• Η αποτελεσματικότητα της μεθόδου βασίζεται σε μεγάλο βαθμό στην αρχικοποίηση των παραμέτρων της και ειδικότερα στους κοντινότερους γείτονες. • Η μέθοδος απαιτεί μεγάλη υπολογιστική ισχύ, ιδιαίτερα όσο αυξάνεται ο αριθμός των δεδομένων. • Αν υπάρχουν μη συνδεδεμένες γειτονιές κατά τον υπολογισμό των κοντινότερων γειτόνων είναι πολύ πιθανό η μεταφορά των δεδομένων στον χώρο λιγότερων διαστάσεων να μην είναι αποτελεσματική.

Κεφάλαιο 3

Βιβλιογραφική Ανασκόπηση

3.1 Εισαγωγή

Στην επιστήμη της ανάλυσης δεδομένων και ειδικότερα στον τομέα της μηχανικής μάθησης υπάρχει πληθώρα μεθοδολογιών και προσεγγίσεων για την ανάπτυξη και βελτίωση των μοντέλων πρόβλεψης. Ένα αναπόσπαστο κομμάτι αυτής της διαδικασίας είναι η προεπεξεργασία των δεδομένων. Πολλά σύνολα δεδομένων αντιμετωπίζουν προβλήματα όπως ελλιπείς τιμές, ακραίες τιμές και πλεόνασμα χαρακτηριστικών που μπορεί όχι μόνο να είναι περιττά με μηδενική συνεισφορά στη βελτίωση του τελικού μοντέλου, αλλά αντίθετα να αποτελούν τροχοπέδη σε αυτήν. Στα προβλήματα αυτά επιδιώκουν να δώσουν λύση οι μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Ο τομέας των μεθόδων αυτών είναι πλούσιος σε περιεχόμενο και ποικιλομορφία. Κατά τη μακροχρόνια ανάπτυξη του τομέα της μηχανικής μάθησης, έχουν αναπτυχθεί παράλληλα ολοένα και περισσότερες τεχνικές για τη μείωση των διαστάσεων ενός συνόλου δεδομένων και την επιλογή των καταλληλότερων χαρακτηριστικών σε αυτό. Υπάρχουν διάφορες προσεγγίσεις που έχουν ακολουθηθεί για την ανάπτυξη νέων και διαφορετικών τεχνικών. Αυτές οι προσεγγίσεις κυμαίνονται από εκείνες που βασίζονται στη γραμμική άλγεβρα και τη στατιστική, οι οποίες στοχεύουν να οριοθετήσουν τις ομοιότητες των χαρακτηριστικών και τη δομή των δεδομένων σε έναν μετασχηματισμένο χώρο, έως πιο διαφοροποιημένες μεθόδους. Μεταξύ αυτών, υπάρχουν εποπτευόμενες μέθοδοι που αξιοποιούν τις ετικέτες του χαρακτηριστικού κλάσης καθώς και μη εποπτευόμενες μέθοδοι που αξιοποιούν μετρικές για τον μετασχηματισμό των δεδομένων. Επιπλέον, διακρίνονται μεταξύ τεχνικών που προσδιορίζουν γραμμικές σχέσεις και πραγματοποιούν

γραμμικούς μετασχηματισμούς στα σύνολα δεδομένων και εκείνων που αποκαλύπτουν μη γραμμικές συσχετίσεις, με ορισμένες από τις πιο προηγμένες στρατηγικές να αξιοποιούν τα νευρωνικά δίκτυα για την αποκρυπτογράφηση πολύπλοκων μη γραμμικών σχέσεων. Όμως υπάρχει άραγε κάποια μέθοδος που να αποτελεί πανάκεια λύση και να λειτουργεί για όλα τα σύνολα δεδομένων; Ποιος τύπος μεθόδων έχει τα καλύτερα αποτελέσματα στη βελτίωση της απόδοσης των τελικών μοντέλων μηχανικής μάθησης και ποια μέθοδος είναι η πιο αποδοτική; Ποιος είναι ο βέλτιστος αριθμός διαστάσεων και πως υπολογίζεται; Είναι καλύτερη η μείωση των διαστάσεων ή η επιλογή χαρακτηριστικών;

3.2 Μη γραμμικές μέθοδοι μείωσης διαστάσεων έναντι γραμμικών

Στη βιβλιογραφία υπάρχει πληθώρα ερευνών που πραγματεύεται θέματα σχετικά με τις μεθόδους μείωσης διαστάσεων, την ανάλυση και τη σύγκριση των μεθόδων προσπαθώντας να δώσει απαντήσεις στα παραπάνω ερωτήματα. Ένα μεγάλο κομμάτι της έρευνας επικεντρώνεται στη σύγκριση γραμμικών και μη γραμμικών μεθόδων. Στην έρευνα των De Backer et al. [70] παρουσιάζεται μια τέτοια σύγκριση. Ειδικότερα στο άρθρο γίνεται μια εκτεταμένη ανάλυση των μεθόδων της πολυδιάστατης κλιμάκωσης (multidimensional scaling), χαρτογράφησης Sammon (Sammon's mapping), χάρτες αυτό-οργάνωσης (self-organizing maps), και αυτοσυσχετιστικά νευρωνικά δίκτυα τροφοδοσίας (auto-associative feedforward neural networks). Ο κύριος στόχος είναι ο μετασχηματισμός των δεδομένων από σύνολα δεδομένων υψηλών διαστάσεων με σκοπό τη διευρεύνηση της ποιότητας των νέων δεδομένων. Για τον έλεγχο και την εκτίμηση των αποτελεσμάτων χρησιμοποιήθηκε ένας ταξινομητής k-Nearest Neighbor (k-NN). Έγινε σύγκριση των μη γραμμικών μεθόδων έναντι γραμμικών μεθόδων και συγκεκριμένα με τη γραμμική μέθοδο PCA. Τα σύνολα δεδομένων που χρησιμοποιήθηκαν για την εξαγωγή των αποτελεσμάτων περιλάμβαναν τεχνητά σύνολα τόσο αριθμητικά όσο και με εικόνες. Τα αποτελέσματα της έρευνας καταλήγουν στο συμπέρασμα ότι οι μη γραμμικές τεχνικές είναι ιδιαίτερα αποτελεσματικές στην εξαγωγή χαρακτηριστικών και στη μείωση διαστάσεων και πολλές φορές ξεπερνούν και γραμμικές μεθόδους όπως

η PCA. Ωστόσο, η έρευνα για την αξιολόγηση των αποτελεσμάτων χρησιμοποίησε ένα μοναδικό τύπο ταξινομητή το οποίο δημιουργεί ερωτήματα σχετικά με την εγκυρότητα των αποτελεσμάτων. Περισσότερο φως στο κομμάτι της σύγκρισης των γραμμικών και των μη γραμμικών μεθόδων έρχεται να δώσει το άρθρο των Anowar et al. [71], όπου παρουσιάζεται μια ανασκόπηση και σύγκριση διαφόρων μεθόδων μείωσης διαστάσεων. Στο άρθρο εξετάζονται και συγκρίνονται γραμμικές μέθοδοι έναντι μη γραμμικών, εποπτευόμενες έναντι μη εποπτευόμενων και τυχαίας προβολής έναντι πολλαπλότητας (manifold-based). Ειδικότερα, ελέγχθηκαν οι μέθοδοι Principal Component Analysis (PCA), Kernel PCA (KPCA), Linear Discriminant Analysis (LDA), Multi-Dimensional Scaling (MDS), Singular Value Decomposition (SVD), Locally Linear Embedding (LLE), Isometric Mapping (ISOMAP), Laplacian Eigenmap (LE), Independent Component Analysis (ICA), και t-Distributed Stochastic Neighbor Embedding (t-SNE). Χρησιμοποιήθηκαν τρία πραγματικά σύνολα δεδομένων διαφορετικών διαστάσεων μέσω των οποίων ελέγχθηκε η αποτελεσματικότητα των μεθόδων μείωσης διαστάσεων στη βελτίωση της ταξινόμησης. Στο κομμάτι της ταξινόμησης χρησιμοποιήθηκε ο ταξινομητής Support Vector Machine (SVM). Τα αποτελέσματα στα οποία καταλήγει το άρθρο είναι ότι η εφαρμογή της κατάλληλης μεθόδου μείωσης διαστάσεων διατηρεί ή και βελτιώνει την απόδοση του ταξινομητή μειώνοντας παράλληλα τον χρόνο εκπαίδευσης. Τεχνικές όπως η PCA και η KPCA έδειξαν σημαντικές βελτιώσεις στην ακρίβεια και στην ταχύτητα του μοντέλου. Επιπλέον καμία μέθοδος δεν αποδείχθηκε ανώτερη από τις υπόλοιπες σε όλες τις περιπτώσεις. Αντίθετα, η επιλογή της καλύτερης μεθόδου φαίνεται να εξαρτάται από τα ειδικά χαρακτηριστικά του συνόλου δεδομένων (π.χ. γραμμικότητα, μέγεθος). Οι εποπτευόμενες μέθοδοι όπως η LDA παρείχαν πλεονεκτήματα στις εργασίες ταξινόμησης αξιοποιώντας πληροφορίες από το χαρακτηριστικό κλάσης, ενώ οι μη εποπτευόμενες μέθοδοι όπως οι PCA, KPCA και t-SNE ήταν πιο ευέλικτες σε διαφορετικούς τύπους συνόλων δεδομένων. Οι μη γραμμικές μέθοδοι (π.χ. t-SNE, Isomap) υπερτερούσαν σε σύνολα δεδομένων με πολύπλοκες δομές, αλλά συχνά συνοδεύονταν με αυξημένο υπολογιστικό κόστος. από την παραπάνω έρευνα παρατηρείται ότι σημαντικό ρόλο έχει και η επιλογή του κατάλληλου ταξινομητή, καθώς επίσης και η μορφή του συνόλου δεδομένων για την αξιολόγηση της αποτελεσματικότητας μιας μεθόδου μείωσης διαστάσεων. Από τις έρευνες προκύπτει ότι

διαφορετικές τεχνικές λειτουργούν καλύτερα σε διαφορετικά σύνολα δεδομένων, καθώς επίσης ότι σημαντικό ρόλο έχει και ο ταξινομητής στην όλη διαδικασία.

Μια πιο αναλυτική ανασκόπηση τόσο γραμμικών όσο και μη γραμμικών μεθόδων παρουσιάζεται στο άρθρο των Sumithra V.S & Surendran [72], όπου γίνεται μια αξιολόγηση διαφόρων μεθόδων μείωσης διαστάσεων και αναλύονται οι δυνατότητες και οι αδυναμίες της κάθε μεθόδου. Ειδικότερα, οι μέθοδοι που αξιολογήθηκαν ήταν η Principal Component Analysis (PCA), Independent Component Analysis (ICA), Singular Value Decomposition (SVD), CUR Matrix Decomposition, Compact Matrix Decomposition (CMD), Non Negative Matrix Factorization (NMF), Linear Discriminant Analysis (LDA), Multidimensional Scaling, Kernel PCA, FastMap, Isomap, Locally Linear Embedding, Laplacian Eigenmaps και Local Tangent Space Alignment. Στα συμπεράσματα της έρευνας τονίζεται ότι πολλές φορές τα δεδομένα είναι μη γραμμικά, απαιτώντας προσεκτική επιλογή μεταξύ γραμμικών και μη γραμμικών μεθόδων μείωσης διαστάσεων με βάση τα συγκεκριμένα χαρακτηριστικά των δεδομένων. Παρά την πληθώρα πλεονεκτημάτων που προσφέρουν γενικότερα οι μέθοδοι μείωσης διαστάσεων σαν βήμα της προεπεξεργασίας η έρευνα υπογραμμίζει και διάφορα προβλήματα που είναι πιθανών να προκύψουν. Οι τεχνικές μείωσης διαστάσεων αντιμετωπίζουν προκλήσεις όπως πιθανή απώλεια πληροφοριών κατά τη διαδικασία μείωσης των διαστάσεων και δυσκολία στην ερμηνεία των μετασχηματισμένων χαρακτηριστικών. Το άρθρο καταλήγει στο συμπέρασμα ότι τόσο οι γραμμικές όσο και οι μη γραμμικές μέθοδοι είναι αποτελεσματικές αλλά απαιτείται προσεκτική επιλογή μεταξύ των δύο κατηγοριών με κύριο κριτήριο τα συγκεκριμένα χαρακτηριστικά των δεδομένων.

Στα παραπάνω έρχεται να συμφωνήσει και να ενισχύσει τις παρατηρήσεις η έρευνα των der Maaten et al. [73] όπου παρουσιάζεται μια εκτεταμένη έρευνα στον τομέα της μείωσης διαστάσεων και μια αναλυτική σύγκριση μεταξύ πολλών κλασσικών και μη μεθόδων. Ειδικότερα, αξιολογήθηκαν οι μέθοδοι Multidimensional Scaling (MDS), Isomap, Maximum Variance Unfolding (MVU), Kernel PCA, Diffusion Maps, Multilayer Autoencoders, Locally Linear Embedding (LLE), Laplacian Eigenmaps, Hessian LLE, Local Tangent Space Analysis (LTSA), Locally Linear Coordination (LLC), Manifold Charting. Ο κύριος στόχος της μελέτης ήταν η αξιολόγηση αυτών των μεθόδων σε σύνολα δεδομένων του πραγματικού κόσμου και ο εντοπισμός των

εγγενών πλεονεκτημάτων και αδυναμιών της κάθε μεθόδου. Στα σύνολα δεδομένων που χρησιμοποιήθηκαν υπήρχαν τόσο τεχνητά όσο και πραγματικά σύνολα δεδομένων. Η έρευνα δείχνει ότι ενώ οι μη γραμμικές τεχνικές συχνά υπερτερούν της PCA σε τεχνητές εργασίες που έχουν σχεδιαστεί για να επιδεικνύουν τα δυνατά τους σημεία, δεν ξεπερνούν σταθερά την PCA σε σύνολα δεδομένων του πραγματικού κόσμου. Μερικές από τις αδυναμίες των μεθόδων που παρατηρήθηκαν κατά την έρευνα είναι η ευαισθησία στις επιλογές υπερπαραμέτρων, η υπολογιστική πολυπλοκότητα και, σε ορισμένες περιπτώσεις, η αδυναμία αποτελεσματικής αποτύπωσης της υποκείμενης δομής ενός πιο περίπλοκου ή θορυβώδους συνόλου δεδομένων του πραγματικού κόσμου. Αυτά τα ευρήματα υπογραμμίζουν την αναγκαιότητα αξιολόγησης των μεθόδων μείωσης διαστάσεων όχι μόνο ως προς τις θεωρητικές τους ικανότητες αλλά και για την πρακτική εφαρμογή και την απόδοση τους σε σενάρια του πραγματικού κόσμου.

Στο άρθρο του Mendez [74], παρουσιάζεται μια έρευνα που πραγματοποιήθηκε πάνω σε μεθόδους μείωσης διαστάσεων στον τομέα της μηχανικής ρευστών. Αποτελεί ένα χαρακτηριστικό παράδειγμα της ευρείας χρήσης και εφαρμογής των μεθόδων μείωσης διαστάσεων σαν εργαλεία σε διάφορους τομείς και πεδία της επιστήμης. Η έρευνα διαχωρίζει τις μεθόδους σε γραμμικές και μη γραμμικές. Συγκεκριμένα, η γραμμική μέθοδος που χρησιμοποιήθηκε είναι η Principal Component Analysis (PCA), ενώ οι μη γραμμικές μέθοδοι είναι η Kernel Principal Component Analysis (kernel PCA), Locally Linear Embedding (LLE), Isomap. Οι μέθοδοι αξιολογήθηκαν σε τρία προβλήματα στον τομέα της ρευστοδυναμικής, στο φιλτράρισμα, στην αναγνώριση ταλαντωτικών μοτίβων και στη συμπίεση δεδομένων. Τα αποτελέσματα του άρθρου δεν καταλήγουν σε κάποια συγκεκριμένη ομάδα μεθόδων ως καλύτερη από την άλλη. Αντίθετα, παραθέτονται πλεονεκτήματα και χαρακτηριστικά των διαφόρων μεθόδων που υποδεικνύουν ότι η επιλογή της καλύτερης μεθόδου είναι πολυπαραγωντική και εξαρτάται τόσο από το σύνολο δεδομένων όσο και από τη διεργασία πάνω σε αυτό. Τέλος, υπογραμμίζεται η αποτελεσματικότητα των μη γραμμικών τεχνικών στην αποκάλυψη περίπλοκων μοτίβων σε δεδομένα δυναμικής ρευστών που ενδέχεται να μην είναι δυνατόν να καταγραφούν από τις γραμμικές μεθόδους. Το τελευταίο εύρημα του άρθρου δείχνει ότι ένας κύριος παράγοντας για την επιλογή της κατάλληλης μεθόδου μείωσης διαστάσεων είναι η πολυπλοκότητα

των δεδομένων και ειδικότερα, η ύπαρξη μη γραμμικών σχέσεων και μοτίβων στα δεδομένα. Οι γραμμικές μέθοδοι αδυνατούν να καταγράψουν τέτοιου είδους σχέσεις δίνοντας ένα σημαντικό πλεονέκτημα στις μη γραμμικές. από όλες τις προηγούμενες έρευνες προκύπτει ο ισχυρισμός ότι οι μη γραμμικές μέθοδοι δεν είναι απλώς θεωρητικά ανώτερες στη διαχείριση πολύπλοκων δομών δεδομένων αλλά επιδεικνύουν επίσης αξιοσημείωτη πρακτική αποτελεσματικότητα σε πολλά πραγματικά σενάρια. Αυτός ο ισχυρισμός επικυρώνεται εμπειρικά από τα ευρήματα των Rastogi et al. [75], όπου αναλύεται η αποτελεσματικότητα των διαφόρων αλγορίθμων εξαγωγής χαρακτηριστικών (μέθοδοι μείωσης διαστάσεων) τόσο σε θεωρητικό όσο και εμπειρικό επίπεδο, με σκοπό τη βελτίωση της ποιότητας των δεδομένων και τη βελτίωση της απόδοσης των αλγορίθμων μηχανικής μάθησης. Για την εμπειρική ανάλυση χρησιμοποιήθηκαν τρία σύνολα δεδομένων του πραγματικού κόσμου με διάφορες διαστάσεις. Τα αποτελέσματα που εξήχθησαν μέσα από αυτήν την έρευνα δείχνουν βελτιώσεις στην ποιότητα των δεδομένων και στην ακρίβεια ταξινόμησης, οι μέθοδοι που βασίζονται σε πολλαπλή (manifold) παρουσιάζουν ανώτερη απόδοση σε σύγκριση με τις μεθόδους που βασίζονται σε τυχαία προβολή, οι μη γραμμικές μέθοδοι ξεπερνούν τις περισσότερες φορές τις γραμμικές μεθόδους, επιδεικνύοντας την αποτελεσματικότητά τους στην αντιμετώπιση πολύπλοκων δομών δεδομένων. Τέλος σε σύνολα δεδομένων με πολλαπλές ετικέτες του χαρακτηριστικού κλάσης οι εποπτευόμενες τεχνικές φαίνεται να παρουσιάζουν καλύτερη αποτελεσματικότητα.

3.3 Εποπτευόμενες έναντι μη εποπτευόμενων μεθόδων

Μέσα από την προηγούμενη έρευνα προκύπτει ένας διαφορετικός διαχωρισμός των μεθόδων ανάλογα με το αν η μέθοδος είναι εποπτευόμενη ή όχι. Ο συγκεκριμένος τύπος διαχωρισμού των μεθόδων κρίνεται σημαντικό να διερευνηθεί περεταίρω ώστε η συνολική ανασκόπηση να συμπεριλαμβάνει το ευρύτερο τοπίο των τεχνικών μείωσης διαστάσεων και μέσω της σκοπιάς της εποπτευόμενης και μη εποπτευόμενης μάθησης. Στο άρθρο των Archana & Sachin [76], γίνεται μια συγκριτική ανάλυση της αποτελεσματικότητας της μη εποπτευόμενης μεθόδου PCA και της εποπτευόμενης μεθόδου LDA. Οι συγκρίσεις έγιναν σε σύνολα δεδομένων που περιέχουν εικόνες από βάσεις δεδομένων COIL-20, UMIST και YALE-B. Για την αξιολόγηση των μεθόδων και το πως αυτές επηρεάζουν την απόδοση των ταξινομητών χρησι-

μοποιήθηκαν δύο κλασσικοί ταξινομητές, ο K-Nearest Neighbors (KNN) και ο Naive Bayes. Η έρευνα καταλήγει στο συμπέρασμα ότι η PCA σε συνδυασμό είτε με τον ταξινομητή KNN είτε με τον ταξινομητή Naive Bayes, αποδίδει καλά στο σύνολο δεδομένων COIL-20. Ωστόσο, η LDA δείχνει καλύτερη απόδοση στα σύνολα δεδομένων UMIST και YALE-B, πιθανότατα λόγω της ικανότητας της να χρησιμοποιεί πληροφορίες των ετικετών του χαρακτηριστικού κλάσης. Τέλος σύμφωνα με το άρθρο και οι δύο μέθοδοι είναι ικανές και αποτελεσματικές στη μείωση διαστάσεων σε σύνολα δεδομένων εικόνων. Τα αποτελέσματα της έρευνας δεν δείχνουν μόνο την αποτελεσματικότητα των γραμμικών μεθόδων σε σύνολα δεδομένων με εικόνες αλλά ταυτόχρονα τονίζουν και τις διαφορές μεταξύ των μεθόδων χωρίς εποπτεία (PCA) και των μεθόδων με εποπτεία (LDA). Παρόμοια αποτελέσματα παρουσιάζει και η μελέτη των Ahmmed et al. [77], όπου γίνεται μια ανάλυση και αξιολόγηση των μεθόδων PCA και LDA στην ταξινόμηση υπερφασματικής εικόνας (HSI). Το σύνολο δεδομένων που χρησιμοποιήθηκε ήταν το Indian Pines που αποτελείται από 145x145 εικονοστοιχεία και 220 φασματικές ζώνες. Η αξιολόγηση των μεθόδων έγινε μέσω του ταξινομητή SVM. Η έρευνα εκτός από σύγκριση και αξιολόγηση των δύο μεθόδων, πάει ένα βήμα παραπέρα συνδυάζοντας τις δύο μεθόδους και ενώνοντας τα αποτελέσματα τους. Τα πειραματικά αποτελέσματα δείχνουν ότι η συνδυασμένη προσέγγιση είχε τα χειρότερα αποτελέσματα υποδηλώνοντας ότι ενώ ο συνδυασμός χαρακτηριστικών και από τις δύο μεθόδους αυξάνει τον χώρο των χαρακτηριστικών, δεν οδηγεί απαραίτητα σε καλύτερη ακρίβεια ταξινόμησης. Τέλος, καλύτερη απόδοση είχε η LDA από τις τρεις μεθόδους. Ωστόσο, τόσο η LDA όσο και η PCA βελτίωσαν την αποτελεσματικότητα του ταξινομητή. Επομένως από τις δύο μεθόδους φαίνεται να υπερτερεί τις περισσότερες φορές η μέθοδος LDA. Οι δύο προηγούμενες έρευνες διερεύνησαν το κομμάτι των εποπτευόμενων έναντι των μη εποπτευόμενων μεθόδων μέσω της σύγκρισης των μεθόδων PCA και LDA. Κρίνεται απαραίτητο για την εξαγωγή πιο έγκυρων συμπερασμάτων να γίνει μια ευρύτερη εξέταση αυτών των τεχνικών σε διαφορετικούς τομείς και χρησιμοποιώντας μεγαλύτερο αριθμό μεθόδων μείωσης διαστάσεων.

Στο άρθρο των Halpern et al. [78], πραγματοποιείται μια σύγκριση και αξιολόγηση διαφόρων μεθόδων μείωσης διαστάσεων. Η έρευνα επικεντρώνεται σε κλινικά κείμενα από τμήματα επειγόντων περιστατικών. Χρησιμοποιήθηκαν τόσο επο-

πτευόμενες όσο και μη εποπτευόμενες μέθοδοι. Οι εποπτευόμενες μέθοδοι ήταν η supervised Latent Dirichlet Allocation (sLDA) και η Maximum Entropy Discrimination Latent Dirichlet Allocation (MedLDA) ενώ οι μη εποπτευόμενες ήταν η Latent Dirichlet Allocation (LDA) και η Singular Value Decomposition (SVD). Στόχος της έρευνας είναι η βελτίωση των μοντέλων μηχανικής μάθησης με απώτερο σκοπό την καλύτερη πρόβλεψη κλινικών αποτελεσμάτων. Τα δεδομένα πάνω στα οποία εφαρμόστηκαν οι μέθοδοι συγκροτήθηκαν από σημειώσεις επισκέψεων στο τμήμα επειγόντων περιστατικών εντός μιας συγκεκριμένης περιόδου, συνολικού αριθμού 94.973 αρχείων ασθενών. Για τη σύγκριση των μεθόδων χρησιμοποιήθηκε ένας γραμμικός ταξινομητής SVM. Τα αποτελέσματα της έρευνας έδειξαν ότι όλες οι μέθοδοι ενίσχυσαν την ικανότητα ταξινόμησης του ταξινομητή. Επίσης παρατηρήθηκε ότι οι εποπτευόμενες μέθοδοι ήταν αποτελεσματικότερες σε σχέση με τις μη εποπτευόμενες μεθόδους, ιδιαίτερα για αναπαραστάσεις με πολύ λίγες διαστάσεις. Παρόλα αυτά, όταν οι διαστάσεις άρχισαν να ξεπερνούν τις πενήντα, η απόδοση των εποπτευόμενων και των μη εποπτευόμενων μεθόδων έγινε συγκρίσιμη. Όλες οι προηγούμενες έρευνες, τείνουν στο συμπέρασμα ότι οι μη εποπτευόμενες έρευνες έχουν το πλεονέκτημα, ειδικότερα όταν χρειάζεται να μειωθούν σε μεγάλο βαθμό οι διαστάσεις αλλά αυτό ισορροπείται όσο αυξάνονται οι διαστάσεις. Αξιοσημείωτο κομμάτι είναι ο έλεγχος των μεθόδων υπό την παρουσία θορύβου.

Στο άρθρο των Balachander et al. [79], παρουσιάζεται μια ανάλυση και σύγκριση μεθόδων μείωσης διαστάσεων με και χωρίς επίβλεψη σε συνθήκες όπου τα σύνολα δεδομένων περιέχουν θόρυβο. Το πλαίσιο στο οποίο πραγματοποιήθηκε η έρευνα είναι στην κυτταροδιάγνωση του λεμφώματος. Τα σύνολα δεδομένων που χρησιμοποιήθηκαν συγκροτήθηκαν από κύτταρα σε κυτταρολογικά παρασκευάσματα, όπου εννέα χαρακτηριστικά εξήχθησαν από κάθε κύτταρο. Οι μέθοδοι μείωσης διαστάσεων που χρησιμοποιήθηκαν ήταν τέσσερις, όπου δύο από αυτές ήταν χωρίς επίβλεψη, η PCA και η Self-Organizing Feature Maps και δύο ήταν με επίβλεψη Fisher's Linear Discriminants και Feed-Forward Neural Networks. Οι προαναφερόμενες μέθοδοι αξιολογήθηκαν με κριτήριο τη δυνατότητα τους να μειώνουν τη διάσταση του χώρου των χαρακτηριστικών από εννιά σε ένα χαρακτηριστικά, με και χωρίς την παρουσία θορύβου (0%, 1% και 5% θορύβου) και διερευνήθηκε και η ανθεκτικότητα αυτών των μεθόδων στην παρουσία θορύβου. Τα αποτελέσματα της έρευνας έδει-

ξαν ότι όλες οι μέθοδοι παρουσίασαν ευαισθησία στο θόρυβο με την απόδοση τους να μειώνεται όσο αυξανόταν ο θόρυβος. Τέλος τα Feed-Forward Neural Networks φαίνεται να έχουν την καλύτερη ισορροπία μεταξύ της μείωσης διαστάσεων και ακρίβεια ταξινόμησης προτύπων, ακόμη και με την παρουσία θορύβου.

3.4 Νευρωνικά έναντι κλασσικών μεθόδων

Για άλλη μια φορά τα νευρωνικά δίκτυα δείχνουν τις πολυάριθμες δυνατότητες τους στον τομέα της μηχανικής μάθησης. Τα νευρωνικά δίκτυα έχουν ευρεία χρήση και ραγδαία ανάπτυξη τα τελευταία χρόνια. Μέθοδοι μείωσης διαστάσεων που βασίζονται σε νευρωνικά δίκτυα έχουν αναπτυχθεί και δείχνουν να είναι ιδιαίτερα αποτελεσματικές. Τις παραπάνω παρατηρήσεις με τη σύγκριση νευρωνικών δικτύων έναντι κλασσικών μεθόδων έρχεται να αναλύσει σε βάθος η έρευνα των Fournier & Aloise [80]. Η έρευνα πραγματεύεται ζητήματα που αφορούν την αποτελεσματικότητα κλασσικών μεθόδων μείωσης διαστάσεων έναντι των αυτοκωδικοποιητών (autoencoders). Η συγκεκριμένη μέθοδος βασίζεται σε νευρωνικά δίκτυα και έχει δείξει πολύ καλά αποτελέσματα. Η έρευνα πραγματοποιείται στο πλαίσιο της ταξινόμησης εικόνων. Χρησιμοποιήθηκαν τα σύνολα δεδομένων MNIST, Fashion-MNIST και CIFAR-10, που περιέχουν εικόνες αντικειμένων και ανθρώπινων προσώπων και ο ταξινομητής K-Nearest Neighbors (KNN). Οι κλασσικές μέθοδοι που χρησιμοποιήθηκαν για τη σύγκριση μεταξύ νευρωνικών ήταν η PCA και η Isomap. Για την αξιολόγηση των τριών μεθόδων λήφθηκε υπόψιν κατά κύριο λόγο η απόδοση του ταξινομητή καθώς επίσης και ο χρόνος που χρειάστηκε κάθε μέθοδος να ολοκληρωθεί. Σύμφωνα με τα αποτελέσματα της έρευνας παρόλο που η PCA χρειάστηκε πολύ λιγότερο χρόνο παρείχε ανταγωνιστική ακρίβεια σε σχέση με τις υπόλοιπες δύο μεθόδους και στα τρία σύνολα δεδομένων, με ελάχιστες διαφορές στην απόδοση. Ωστόσο, για μικρές διαστάσεις, η μέθοδος autoencoders ξεπέρασε την PCA, πιθανότατα λόγω της ικανότητας της να μαθαίνει εξαιρετικά μη γραμμικούς μετασχηματισμούς. Επίσης στην έρευνα τονίζεται ότι μέσω της PCA μπορεί να θεωρηθεί ότι υπάρχει ένα βέλτιστο μέγεθος διαστάσεων το οποίο είναι ιδιαίτερα εμφανές στο σύνολο δεδομένων CIFAR-10, όπου οι πολύ υψηλές διαστάσεις θα μπορούσαν να δημιουργήσουν περισσότερο θόρυβο παρά χρήσιμες πληροφορίες. Τέλος οι συγγραφείς καταλήγουν στο συμπέρασμα ότι παρόλο που η PCA είναι μια από τις πιο

απλές και κλασσικές μεθόδους παραμένει ανταγωνιστική με τις σύγχρονες μεθόδους στις εργασίες ταξινόμησης εικόνων.

3.5 Επιλογή χαρακτηριστικών και μείωση διαστάσεων

Παίρνοντας μια πιο μακροσκοπική σκοπιά και προσθέτοντας στη σφαίρα της ανάλυσης τόσο μεθόδους μείωσης διαστάσεων όσο και μεθόδους επιλογής χαρακτηριστικών, κρίνεται απαραίτητη η σύγκριση των δύο ομάδων μεθόδων. Το άρθρο των Khalid et al. [81], πραγματοποιεί μια πλήρης ανασκόπηση διαφόρων μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Οι μέθοδοι επιλογής χαρακτηριστικών που αξιολογήθηκαν ήταν η Prediction Analysis of Microarray (PAM), Minimal Redundancy and Maximal Relevance (mRMR), I-RELIEF, Conditional Mutual Information Maximization (CMIM), Correlation Coefficient, Between-Within Ratio (BW-Ratio), INTERACT, Genetic Algorithms (GA), Support Vector Machine Recursive Feature Elimination (SVM-RFE), οι μέθοδοι μείωσης διαστάσεων που χρησιμοποιήθηκαν ήταν η PCA, ICA, Non-Linear PCA, Probabilistic PCA (PPCA), Kernel PCA (KPCA), probabilistic kernel principle component analysis (PKPCA). Η αξιολόγηση των μεθόδων έγινε με βάση τη βελτίωση της απόδοσης μοντέλων μηχανικής μάθησης όπου χρησιμοποιήθηκαν οι ταξινομητές Naive Bayes και LIBSVM και την καταλληλότητα των μεθόδων για συγκεκριμένες καταστάσεις όπου τονίζεται ότι μερικές μέθοδοι είναι πιο κατάλληλες για συγκεκριμένους τύπους και σχέσεις μεταξύ των χαρακτηριστικών (όπως γραμμικές έναντι μη γραμμικές σχέσεις). Τα αποτελέσματα της έρευνας καταλήγουν στο συμπέρασμα ότι δεν υπάρχει κάποια μέθοδος η οποία να υπερτερεί πάντα έναντι των υπολοίπων. Αντίθετα, η καλύτερη μέθοδος κάθε φορά εξαρτάται από συγκεκριμένα χαρακτηριστικά των δεδομένων και την προβλεπόμενη εφαρμογή. Η μέθοδοι μείωσης διαστάσεων είχαν μεγαλύτερη αποτελεσματικότητα, λόγω της ύπαρξης θορύβου στα δεδομένα. Τέλος, οι εποπτευόμενες μέθοδοι φαίνεται να επέστρεφαν πιο σχετικά υποσύνολα των δεδομένων για εργασίες πρόβλεψης συγκριτικά με τις μη εποπτευόμενες μεθόδους. Στα ευρήματα του προηγούμενου άρθρου έρχεται να επεκτείνει την έρευνα το άρθρο των Ray et al. [82], στο οποίο γίνεται μια παρουσίαση και αναλυτική ανασκόπηση διαφόρων μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Οι αλγόριθμοι που χρησιμοποιήθηκαν χωρίστηκαν σε μεθόδους επιλογής χαρακτηριστικών και μεθόδους μείωσης διαστά-

σεων. Οι μέθοδοι μείωσης διαστάσεων ήταν η Principal component analysis (PCA), Local fsher's discriminate analysis (LFDA), Canonical correlation analysis (CCA), Non-negative matrix factorization (NMF), Isomap, Locally linear embedding (LLE), Laplacian eigenmap (LE). Οι μέθοδοι επιλογής χαρακτηριστικών ήταν η Hybridised genetic algorithm and particle swarm optimization (HGAPSO), Genetic algorithm (GA), Particle swarm optimization (PSO), Hgapso, ReliefF, Minimum redundancy maximum relevance (MRMR), Recursive feature elimination (RFE), Simultaneous perturbation stochastic approximation (SPSA). Τα ευρήματα της έρευνας ήταν ότι τόσο οι μέθοδοι μείωσης διαστάσεων όσο και οι μέθοδοι επιλογής χαρακτηριστικών αποτελούν ένα πολύ σημαντικό βήμα στο κομμάτι της προεπεξεργασίας των δεδομένων καθώς τείνουν να βελτιώνουν την ανάλυση της πρόβλεψης των μοντέλων μηχανικής μάθησης, κάνοντας καλύτερη την οπτικοποίηση και παράλληλα διατηρώντας όσο το δυνατόν μεγαλύτερο μέρος της πληροφορίας των δεδομένων. Στη μελέτη χρησιμοποιήθηκαν σύνολα δεδομένων μικροσυστοιχιών όγκων και για την ταξινόμηση και σύγκριση των μεθόδων χρησιμοποιήθηκαν οι ταξινομητές Support Vector Machine (SVM) και k-Nearest Neighbors (kNN), Logistic Regression, Decision Tree, Naive Bayes, Random Forest. από τα μοντέλα μηχανικής μάθησης χρησιμοποιήθηκε τόσο το μέτρο της απόδοσης όσο και οι καμπύλες ROC για την καλύτερη αξιολόγηση. Οι συγγραφείς καταλήγουν στο συμπέρασμα ότι οι εποπτευόμενες μέθοδοι παρείχαν γενικά καλύτερες επιδόσεις σε σχέση με τις μη εποπτευόμενες μεθόδους, ειδικά όταν υπήρχε η απαίτηση να μειωθούν οι διαστάσεις σε μεγάλο βαθμό. Επίσης τονίστηκε ότι τόσο οι μέθοδοι μείωσης διαστάσεων με την επιλογή του κατάλληλου αριθμού διαστάσεων όσο και οι μέθοδοι επιλογής χαρακτηριστικών επιτρέπουν την ανάπτυξη πιο ακριβών και υπολογιστικά αποδοτικών μοντέλων μηχανικής μάθησης, ιδιαίτερα στην επεξεργασία και ανάλυση πολύπλοκων συνόλων δεδομένων όπως αυτά που υπάρχουν στην υγειονομική περίθαλψη.

3.6 Μείωση της υπερπροσαρμογής

Μια άλλη πτυχή που συνδέεται έμμεσα με την αύξηση της αποτελεσματικότητας ενός μοντέλου μηχανικής μάθησης είναι η υπερπροσαρμογή. Η υπερπροσαρμογή οδηγεί τα μοντέλα μηχανικής μάθησης να προσαρμόζονται σε υπερβολικό βαθμό στα δεδομένα εκπαίδευσης με αποτέλεσμα όταν έρθουν αντιμέτωπα με άγνωστα δεδο-

μένα να έχουν χαμηλή απόδοση, ακριβώς επειδή προσαρμόστηκαν στα δεδομένα εκπαίδευσης σε υπερβολικό βαθμό. Οι μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών μπορούν να αντιμετωπίσουν αυτό το πρόβλημα, ειδικότερα στο άρθρο των Salam et al. [83], αναλύονται διάφορες μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών με στόχο τον μετριασμό του ζητήματος της υπερπροσαρμογής στα μοντέλα μηχανικής μάθησης. Η συγκεκριμένη έρευνα μπορεί να μη σχετίζεται άμεσα με τη σύγκριση των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών βάση της απόδοσης μοντέλων μηχανικής μάθησης, όπως προαναφέρθηκε σε πολλές έρευνες. Αντίθετα, έμμεσα μέσω της σύγκρισης των μεθόδων στην αποτελεσματικότητα αντιμετώπισης του ζητήματος της υπερπροσαρμογής, η έρευνα προσπαθεί να ρίξει φως στο ποια μέθοδος θα κάνει το μοντέλο να λειτουργεί αποτελεσματικότερα σε άγνωστα δεδομένα κάτι το οποίο αντικατοπτρίζει την αποτελεσματικότητα του μοντέλου. Οι μέθοδοι που αναλύθηκαν είναι η Missing-Values Ratio (MVR), Low-Variance Filter (LVF), High-Correlation Filter (HCF), Random Forest, PCA, LDA, Backward Feature Elimination (BFE), Forward Feature Construction (FFC), Rough Set Theory (RS). Για την αξιολόγηση των μεθόδων χρησιμοποιήθηκαν οι ταξινομητές Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest Classifier (RFC) εξάγοντας μετρήσεις σχετικά με το accuracy, precision, recall, και F1 score. Τα σύνολα δεδομένων που χρησιμοποιήθηκαν ήταν το Congressional Voting Records Dataset και το Bands Dataset. Τα αποτελέσματα παρουσίασαν μια διαφοροποίηση της αποτελεσματικότητας μεταξύ των μοντέλων μηχανικής μάθησης (ταξινομητών) και των συνόλων δεδομένων. Ωστόσο, οι μέθοδοι Missing-Values Ratio και Low Variance Filter επέφεραν σημαντική βελτίωση στον μετριασμό της υπερπροσαρμογής. Η μέθοδος LDA έδειξε αξιοσημείωτη απόδοση καθώς και επίσης και το μοντέλο ταξινομητή Random forest επέδειξε σημαντική μείωση στο ζήτημα της υπερπροσαρμογής μέσω των μεθόδων μείωσης διαστάσεων. Έτσι οι συγγραφείς καταλήγουν στο ότι οι μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών μπορούν να μειώσουν αποτελεσματικά την υπερπροσαρμογή σε μοντέλα μηχανικής μάθησης, διατηρώντας ή ακόμα και βελτιώνοντας την απόδοση του τελικού μοντέλου. Η επιλογή της κατάλληλης μεθόδου πρέπει να εξετάζεται μεθοδικά τόσο με γνώμονα τα χαρακτηριστικά του συνόλου δεδομένων όσο και με το μοντέλο μηχανικής μάθησης που θα χρησιμοποιηθεί.

3.7 Βέλτιστος αριθμός διαστάσεων

Η βιβλιογραφία ως τώρα έχει δείξει ότι δεν υπάρχει κάποια μέθοδος που να ξεπερνάει όλες τις υπόλοιπες και να λειτουργεί αποτελεσματικά σε όλα τα σύνολα δεδομένων. Αντίθετα, η επιλογή της κατάλληλης μεθόδου αποτελεί ένα πολυπαραγωντικό ζήτημα που σχετίζεται με τη μορφή των δεδομένων, τις εγγενείς σχέσεις μεταξύ των δεδομένων αλλά και το τελικό μοντέλο μηχανικής μάθησης. Το ζήτημα που προκύπτει είναι, αφού έχει επιλεχθεί η κατάλληλη τεχνική, υπάρχει κάποιος τρόπος για την εύρεση του βέλτιστου αριθμού διαστάσεων, διατηρώντας μια ισορροπία μεταξύ του αριθμού των διαστάσεων και της πληροφορίας που διατηρείται μέσα από αυτές. Το άρθρο των Plastria et al. [84], παρουσιάζει την επίδραση διαφορετικών μεθόδων μείωσης διαστάσεων με κύριο σκοπό τη βελτίωση της απόδοσης της διαδικασίας της ταξινόμησης. Οι μέθοδοι που χρησιμοποιήθηκαν ήταν η Principal Separation, Principal Mean Components, Linear Mean Discriminants, Principal Mean Separation Components. Για την αξιολόγηση των μεθόδων χρησιμοποιήθηκε ο ταξινομητής Optimal Distance Separating Hyperplane και ο Eigenvalue-based Classification Tree. Οι δύο ταξινομητές χρησιμοποιήθηκαν σε έξι σύνολα αριθμητικών δεδομένων. Τα αποτελέσματα του άρθρου έδειξαν ότι καμία από τις μεθόδους δεν είναι σταθερά καλύτερη από την άλλη. Παρόλα αυτά, τονίζουν ότι τόσο η μέθοδος μείωσης διαστάσεων όσο και ο αριθμός των νέων διαστάσεων παίζουν σημαντικό ρόλο στην ταξινόμηση. Σε ορισμένες περιπτώσεις η μείωση των διαστάσεων των δεδομένων βελτίωσε τα αποτελέσματα ταξινόμησης, ενώ σε άλλες, η διατήρηση περισσότερων διαστάσεων ήταν πιο αποτελεσματική. Επίσης οι ερευνητές συμπεραίνουν ότι ο βέλτιστος αριθμός διαστάσεων εξαρτάται συχνά από τη δομή των δεδομένων και τον στόχο ταξινόμησης παρά από την ίδια τη μέθοδο μείωσης διαστάσεων που χρησιμοποιείται. Αυτό υποδηλώνει ότι η προσέγγιση ”ένα μέγεθος για όλους” για τη μείωση των διαστάσεων είναι αναποτελεσματική. Αντίθετα, η σε βάθος κατανόηση του συνόλου δεδομένων και των στόχων ταξινόμησης είναι επιτακτική για να καθοριστεί εάν θα μειωθούν οι διαστάσεις και, εάν ναι, σε ποιο βαθμό. Υπάρχουν όμως εργαλεία ή μεθοδολογίες που να καθοδηγούν τη διαδικασία της ανάλυσης δείχνοντας το σωστό μονοπάτι και επιδεικνύοντας ένα εύρος αριθμού διαστάσεων που επιφέρει μια ισορροπία μεταξύ της μείωσης διαστάσεων και της διατήρησης της

πληροφορίας;

3.8 Κριτήρια εύρεσης βέλτιστου αριθμού διαστάσεων

Το άρθρο των On et al. [85], προσπαθεί να δώσει απάντηση στο παραπάνω ερώτημα. Επικεντρώνεται στην ανάλυση της μεθόδου SVD και ειδικότερα στοχεύει στην αξιολόγηση της αποτελεσματικότητας της. Γίνεται αξιοποίηση των κριτηρίων Scree plot, Kaiser Gutman, Cumulative Variance με στόχο την αποτελεσματικότητα των κριτηρίων στην εύρεση ενός αριθμού διαστάσεων, που διατηρεί την ισορροπία μεταξύ του αριθμού των διαστάσεων και της διατήρησης της πληροφορίας. Για την αξιολόγηση της μεθόδου SVD χρησιμοποιείται ένας ταξινομητής τεχνητού νευρωνικού δικτύου (ANN). Τα σύνολα δεδομένων στα οποία εφαρμόζεται η μέθοδος είναι το Page Blocks Classification Dataset, το QSAR Biodegradation Dataset, το Spam Base Dataset, το Thoracic Surgery Data και το Climate Model Simulation Crashes Dataset. Η εφαρμογή της μεθόδου SVD είχε θετική επίδραση στην ικανότητα ταξινόμησης του μοντέλου αυξάνοντας την αποτελεσματικότητα του. Ωστόσο, τα κριτήρια έδειξαν ασυνέπειες στον προσδιορισμό του βέλτιστου αριθμού διαστάσεων. Οι συγγραφείς υπογραμμίζουν ότι απαιτείται πρόσθετη διερεύνηση για τον προσδιορισμό του βέλτιστου αριθμού διαστάσεων. Τα αποτελέσματα της έρευνας ενισχύουν την υπόθεση ότι ο προσδιορισμός του βέλτιστου αριθμού διαστάσεων είναι πολυπαραγωντικό ζήτημα και εξαρτάται τόσο από τα δεδομένα και τον ταξινομητή όσο και από την εμπειρική αξιολόγηση του ίδιου του αναλυτή. Τα τρία κριτήρια του άρθρου μπορεί να μην επιστρέφουν το βέλτιστο αριθμό διαστάσεων αλλά μπορούν να αποτελέσουν μια αφετηρία, μειώνοντας το πλάτος της ανάλυσης που πρέπει πραγματοποιηθεί, βελτιστοποιώντας έτσι την υπολογιστική αποτελεσματικότητα των επόμενων αναλύσεων. Παράλληλα, μπορούν να χρησιμοποιηθούν και σαν εργαλεία, παρέχοντας πολύτιμες γνώσεις για τη φύση των δεδομένων και την αποτελεσματικότητα των τεχνικών μείωσης διαστάσεων. Μέσω της εφαρμογής τους, οι αναλυτές μπορούν να κατανοήσουν καλύτερα την υποκείμενη δομή του συνόλου δεδομένων και την επίδραση της μεθόδου μείωσης διαστάσεων σε αυτά, διευκολύνοντας την περαιτέρω ανάλυση. Χρησιμοποιώντας τα παραπάνω δεδομένα το άρθρο των Ledesma et al. [86], εστιάζει στην ανάλυση της χρησιμότητας του Scree plot στον προσδιορισμό του βέλτιστου αριθμού διαστάσεων και στην περαιτέρω βελτίωση της μεθόδου,

ώστε να οδηγεί σε καλύτερα αποτελέσματα. Οι μέθοδοι μείωσης διαστάσεων που χρησιμοποιούνται για την αξιολόγηση του κριτηρίου Scree plot είναι η PCA και η διερευνητική παραγοντική ανάλυση (exploratory factor analysis). Το σύνολο δεδομένων που χρησιμοποιείται είναι το Driving Style Dataset. Η έρευνα υποστηρίζει ότι πολλές φορές η γραφική παράσταση Scree plot παρουσιάζει υποκειμενικότητα και κάποια ασάφεια. Οι συγγραφείς του άρθρου προτείνουν βελτιώσεις οι οποίες θα συνεισφέρουν στην αύξηση της αποτελεσματικότητας του Scree plot και στην αντιμετώπιση των αδυναμιών της μεθόδου. Οι βελτιώσεις χωρίζονται σε εσωτερικές και εξωτερικές. Οι εσωτερικές βελτιώσεις επικεντρώνονται στο να κάνουν το ίδιο το Scree plot πιο ενημερωτικό. Αυτό επιτυγχάνεται ενσωματώνοντας την παράλληλη ανάλυση στη γραφική παράσταση Scree plot, προσφέροντας ένα πιο αντικειμενικό κριτήριο για τον καθορισμό του αριθμού των παραγόντων (νέων διαστάσεων) συγκρίνοντας τις ιδιοτιμές με τις ιδιοτιμές από προσομοιωμένα, μη συσχετισμένα δεδομένα που κατασκευάζονται τυχαία από το αρχικό σύνολο δεδομένων. Οι εξωτερικές βελτιώσεις περιλαμβάνουν τον εμπλουτισμό της γραφικής παράστασης Scree plot συνδυάζοντας στοιχεία άλλων γραφικών παραστάσεων. Ειδικότερα προτείνονται μέθοδοι συνδυασμού της γραφικής παράστασης των παραγόντων (loading factors) με τη γραφική παράσταση Scree plot ή άλλες σχετικές απεικονίσεις με απώτερο σκοπό να προσφέρουν μια πιο ολοκληρωμένη κατανόηση των αποτελεσμάτων της παραγοντικής ανάλυσης.

Κεφάλαιο 4

Υλοποίηση

4.1 PCA

Η υλοποίηση της μεθόδου PCA πραγματοποιείται μέσω της κλάσης PCA. Η υλοποίηση ακολουθεί τη μαθηματική ανάλυση της μεθόδου και επικεντρώνεται στην επέκταση της μεθόδου με σκοπό την αυτοματοποιημένη εύρεση του βέλτιστου αριθμού διαστάσεων. Περιέχει τρεις μεταβλητές κλάσης, τη μεταβλητή eigenvectors, eigenvalues και transformed_data. Οι τρεις μεταβλητές αντιστοιχούν στα ιδιοδιανύσματα, ιδιοτιμές και το μετασχηματισμένο σύνολο δεδομένων. Η κύρια συνάρτηση της μεθόδου είναι η PCA_fit_transform, η οποία υλοποιεί τη μαθηματική μέθοδο της PCA σε κώδικα, δέχεται δύο ορίσματα, το σύνολο δεδομένων και τον αριθμό μείωσης διαστάσεων. Ο αριθμός μείωσης διαστάσεων μπορεί να πάρει τιμές από $1 < K <$ Αριθμός χαρακτηριστικών προσδιορίζοντας τον αριθμό των διαστάσεων που θα διατηρηθούν ή $0 < K < 1$, προσδιορίζοντας το ποσοστό της διακύμανσης που θα διατηρηθεί. Για τα βήματα της μεθόδου υπάρχουν οι συναρτήσεις calc_covariance_matrix, calc_eigenvector_eigenvalues, sort_eigenvectors_eigenvalues, transform. Οι τέσσερις μέθοδοι κατασκευάζουν τον πίνακα συνδιασποράς, υπολογίζουν τις ιδιοτιμές και τα ιδιοδιανύσματα, τα ταξινομούν σε αύξουσα σειρά και μετασχηματίζουν τα αρχικά δεδομένα. Υπάρχει επίσης η δυνατότητα της αυτοματοποιημένης εύρεσης του 'βέλτιστου' αριθμού διαστάσεων. Στην περίπτωση που δεν έχει οριστεί αριθμός διαστάσεων στη μέθοδο PCA_fit_transform, τότε η μέθοδος προσπαθεί να βρει αυτόματα τον βέλτιστο αριθμό διαστάσεων. Ειδικότερα, ακολουθεί κανονικά τα βήματα της μαθηματικής μεθόδου, υπολογίζοντας τα ιδιοδιανύσματα και τις ιδιοτιμές μέσω του πίνακα συνδιασποράς. Στη συνέχεια, καλεί τη συνάρτηση find_optimal_n_components

η οποία μέσω των ιδιοτιμών υπολογίζει την αθροιστική διακύμανση και κατασκευάζει ένα γράφημα με αυτήν και τη συνεισφορά κάθε νέας διάστασης στην ολική διακύμανση. Έπειτα αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει και το σημείο στο οποίο η σχέση της αθροιστικής διακύμανσης και της προσθήκης περισσότερων διαστάσεων προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα. Τέλος, υπάρχουν και οι βοηθητικές συναρτήσεις is_data_numerical και standardize_data, για τον έλεγχο και την τυποποίηση των δεδομένων.

4.1.1 Ψευδοκώδικας υλοποίησης PCA

Result: Initialize eigenvectors, eigenvalues, and transformed data to None
Initialization: eigenvectors ← None, eigenvalues ← None, transformed_data ← None;

Αλγόριθμος 1: PCA: __init__

Data: dataset, k (optional)

Result: Transformed data after PCA computation

if *!is_data_numerical(dataset)* **then**

    Print "Data must be numerical!";

    return *None*;

else

    dataset ← standardize_data(dataset);

    cov_matrix ← calc_covariance_matrix(dataset);

    eigenvectors, eigenvalues ← calc_eigenvector_eigenvalues(cov_matrix);

    eigenvectors, eigenvalues ← sort_eigenvectors_eigenvalues(eigenvalues, eigenvectors);

    if *k is None* **then**

        return *find_optimal_n_components(eigenvalues)*;

    else

        if *k < 1* **then**

            k ← find_k_based_on_variance_rate(eigenvalues, k);

        end

        if *len(eigenvalues) < k* **then**

            Print "K must be smaller than the number of attributes in the dataset";

            return *None*;

        end

        self.eigenvectors ← eigenvectors[:, :k];

        self.eigenvalues ← eigenvalues[:k];

        transformed_data ← transform(dataset);

        return *transformed_data*;

    end

end

Αλγόριθμος 2: PCA: fit_transform

Data: dataset

Result: Standardized dataset

mean ← calculate mean of dataset along each feature;

std_deviation ← calculate standard deviation of dataset along each feature;

std_deviation ← replace 0 with 1 in std_deviation;

standardized_dataset ← (dataset - mean) / std_deviation;

return *standardized_dataset*;

Αλγόριθμος 3: PCA: standardize_data

Data: data

Result: Covariance matrix of the data

covariance_matrix ← calculate the covariance matrix of data with columns as attributes;

return *covariance_matrix*;

Αλγόριθμος 4: PCA: calc_covariance_matrix

Data: covariance matrix

Result: Eigenvectors and eigenvalues of the covariance matrix

eigenvalues, eigenvectors ← compute eigenvalues and eigenvectors of the covariance matrix;

return *eigenvectors*, *eigenvalues*;

Αλγόριθμος 5: PCA: calc_eigenvector_eigenvalues

Data: eigenvalues, eigenvectors

Result: Sorted eigenvectors and eigenvalues in descending order of eigenvalues

sorted_indices ← argsort eigenvalues in descending order;

sorted_eigenvalues ← eigenvalues[sorted_indices];

sorted_eigenvectors ← eigenvectors[:, sorted_indices];

return *sorted_eigenvectors*, *sorted_eigenvalues*;

Αλγόριθμος 6: PCA: sort_eigenvectors_eigenvalues

Data: eigenvalues, variance_rate

Result: k value where cumulative variance first exceeds variance_rate

total_var ← sum(eigenvalues);

cumulative_var ← cumsum(eigenvalues) / total_var;

k ← 1;

while *cumulative_var[k-1] < variance_rate* **do**

    k ← k + 1;

end

return *k*;

Αλγόριθμος 7: PCA: find_k_based_on_variance_rate

Data: dataset

Result: True if data is numerical, False otherwise

return *dataset.dtype is numerical*;

Αλγόριθμος 8: PCA: is_data_numerical

Data: standardized dataset

Result: Dataset transformed into the PCA space

if *dimension mismatch between eigenvectors and dataset* **then**

    Print "Number of features in eigenvectors must match the number of columns in the dataset.";

    return *None*;

end

transformed_data ← project dataset onto the space spanned by the eigenvectors;

return *transformed_data*;

Αλγόριθμος 9: PCA: transform

Data: eigenvalues, use_savgol_filter (default True)

Result: Knee point determining the optimal number of components

eigenvalues ← filter eigenvalues greater than 0;

total_var ← sum(eigenvalues);

cumulative_var ← cumsum(eigenvalues) / total_var;

if *use_savgol_filter* **then**

    window_length ← minimum of 5 or half the length of cumulative_var minus one;

    polyorder ← 2;

    cumulative_var ← apply savgol_filter to smooth cumulative_var;

end

k ← range from 1 to length of cumulative_var;

knee_locator ← find knee using concave curve, increasing direction;

if *knee_locator.knee is None* **then**

    Decrease sensitivity and retry until knee found or 20 attempts;

end

if *knee_locator.knee* **then**

    Plot and display graph marking knee point;

end

return *knee_locator.knee*;

Αλγόριθμος 10: PCA: find_optimal_n_components

4.2 SVD

Η υλοποίηση της μεθόδου SVD πραγματοποιείται μέσω της κλάσης SVD. Η υλοποίηση ακολουθεί τη μαθηματική ανάλυση της μεθόδου και επικεντρώνεται στην επέκταση της μεθόδου με σκοπό την αυτοματοποιημένη εύρεση του βέλτιστου αριθμού διαστάσεων. Περιέχει τέσσερις μεταβλητές κλάσης, τρεις μεταβλητές για τον διαχωρισμό του αρχικού πίνακα στους τρεις υπό πίνακες της μεθόδου, U, sigma, VT και μια μεταβλητή για την αποθήκευση των μετασχηματισμένων δεδομένων. Η συνάρτηση που υλοποιεί τη μαθηματική μέθοδο σε κώδικα είναι η fit_transform. Δέχεται σαν όρισμα το σύνολο δεδομένων και τον αριθμό μείωσης διαστάσεων. Ο αριθμός μείωσης διαστάσεων μπορεί να πάρει τιμές από $1 < K <$ Αριθμός χαρακτηριστικών προσδιορίζοντας τον αριθμό των διαστάσεων που θα διατηρηθούν ή $0 < K < 1$, προσδιορίζοντας το ποσοστό της διακύμανσης που θα διατηρηθεί. Η ανάλυση του αρχικού πίνακα δεδομένων σε τρεις πίνακες γίνεται μέσω της συνάρτησης calc_VT_U_Sigma. Στην περίπτωση που δεν οριστεί αριθμός μείωσης διαστάσεων στη μέθοδο fit_transform, γίνεται αυτόματος υπολογισμός του ΄βέλτιστου' αριθμού διαστάσεων. Αυτό επιτυγχάνεται μέσω της συνάρτησης find_optimal_n_components. Αφότου η μέθοδος fit_transform υπολογίσει τους τρεις υποπίνακες U, sigma, VT, καλείται η συνάρτηση find_optimal_n_components και μέσω του πίνακα sigma υπολογίζεται η αθροιστική διακύμανση. Στη συνέχεια, η μέθοδος κατασκευάζει ένα γράφημα με αυτήν και τη συνεισφορά κάθε νέας διάστασης στην ολική διακύμανση του συνόλου δεδομένων. Αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει και το σημείο στο οποίο η σχέση της αθροιστικής διακύμανσης και της προσθήκης περισσότερων διαστάσεων προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα.

4.2.1 Ψευδοκώδικας υλοποίησης SVD

Result: Initialize U, sigma, VT, and transformed data to None

Initialization: U ← None, sigma ← None, VT ← None, transformed_data ← None;

Αλγόριθμος 11: SVD: __init__

```
Data: dataset, k (optional)
Result: Transformed data after SVD computation
if !is_data_numerical(dataset) then
    Print "Data must be numerical!";
    return None;
else
    calc_VT_U_Sigma(dataset);
    if k is None then
        return find_optimal_n_components(sigma);
    else
        if len(sigma) < k then
            Print "K must be smaller than the number of attributes in the
             dataset";
            return None;
        end
        if k < 1 then
            k ← find_k_based_on_variance_rate(sigma, k);
        end
        sigma ← sigma[:k];
        U ← U[:, :k];
        VT ← VT[:k, :];
        transformed_data ← transform(dataset);
        return transformed_data;
    end
end
```

Αλγόριθμος 12: SVD: fit_transform

```
Data: dataset
Result: Calculate U, sigma, and VT using SVD
(U, sigma, VT) ← compute SVD of dataset, full_matrices = False;
sigma ← sigma;
```

Αλγόριθμος 13: SVD: calc_VT_U_Sigma

Result: Sort U, sigma, and VT based on descending order of sigma

indices ← argsort sigma in descending order;

sigma ← sigma[indices];

VT ← VT[:, indices];

U ← U[:, indices];

Αλγόριθμος 14: SVD: sort_matrices

Data: dataset

Result: True if data is numerical, False otherwise

return *dataset.dtype is numerical*;

Αλγόριθμος 15: SVD: is_data_numerical

Data: dataset

Result: Dataset transformed using VT

transformed_data ← dot product of dataset and VT transposed;

return *transformed_data*;

Αλγόριθμος 16: SVD: transform

Data: eigenvalues, variance_rate

Result: k value where cumulative variance first exceeds variance_rate

total_var ← sum(eigenvalues);

cumulative_var ← cumsum(eigenvalues) / total_var;

k ← 1;

while *cumulative_var[k-1]* $<$ *variance_rate* **do**

  k ← k + 1;

end

return *k*;

Αλγόριθμος 17: SVD: find_k_based_on_variance_rate

Data: eigenvalues, use_savgol_filter (default True)

Result: Knee point determining the optimal number of components

eigenvalues ← filter eigenvalues greater than 0;

total_var ← sum(eigenvalues);

cumulative_var ← cumsum(eigenvalues) / total_var;

if *use_savgol_filter* **then**

    window_length ← minimum of 5 or half the length of cumulative_var minus one;

    polyorder ← 2;

    cumulative_var ← apply savgol_filter to smooth cumulative_var;

end

k ← range from 1 to length of cumulative_var;

knee_locator ← find knee using concave curve, increasing direction;

if *knee_locator.knee is None* **then**

    Decrease sensitivity and retry until knee found or 20 attempts;

end

if *knee_locator.knee* **then**

    Plot and display graph marking knee point;

end

return *knee_locator.knee*;

Αλγόριθμος 18: SVD: find_optimal_n_components

4.3 Ensemble learning

Η υλοποίηση της μεθόδου εκμάθησης συνόλου για επιλογή χαρακτηριστικών βασίζεται στην έμφυτη ικανότητα των μεθόδων εκμάθησης συνόλου να υπολογίζουν τη σημαντικότητα των χαρακτηριστικών, κατά την περίοδο εκμάθησης. Η συνάρτηση __init__ της κλάσης δέχεται σαν όρισμα τον ταξινομητή συνόλου που θα χρησιμοποιηθεί, και αν κρίνεται σκόπιμο τον συνολικό αριθμό χαρακτηριστικών που πρέπει να διατηρηθούν. Περιέχει τέσσερις μεταβλητές κλάσης την estimator και n_features για την αποθήκευση του μοντέλου ταξινομητή και του συνολικού αριθμού των χαρακτηριστικών που κρίνεται απαραίτητο να διατηρηθούν, τη μεταβλητή feature_importance η οποία διατηρεί τη σημαντικότητα κάθε χαρακτηριστικού και τη μεταβλητή selected_indices που αποθηκέυει τον δείκτη κάθε χαρακτηριστικού ώστε να επιστρέφονται οι δείκτες των σημαντικών χαρακτηριστικών μετά το πέρας της μεθόδου. Η βασική συνάρτηση της μεθόδου είναι η συνάρτηση fit. Δέχεται σαν όρισμα το σύνολο δεδομένων, τον πίνακα του χαρακτηριστικού κλάσης, και μια boolean μεταβλητή που καθορίζει αν θα χρησιμοποιηθεί η κλασσική μέθοδος ή αν θα χρησιμοποιηθεί η μέθοδος της μέσης μείωσης ακρίβειας (Mean Decrease Accuracy). Μέσω της συνάρτησης fit καλείται ο ταξινομητής και εκπαιδεύεται στο σύνολο δεδομένων. Έπειτα υπολογίζεται η σημαντικότητα κάθε χαρακτηριστικού, ταξινομούνται τα χαρακτηριστικά και οι δείκτες με βάση τη σημαντικότητα τους και καλείται η συνάρτηση find_optimal_features ή find_optimal_features_MDA ανάλογα. Στη συνάρτηση find_optimal_features υπολογίζεται η αθροιστική σημαντικότητα και κατασκευάζεται η γραφική παράσταση των χαρακτηριστικών σε σχέση με το πόσο συνεισφέρουν στη συνολική σημαντικότητα το καθένα. Έπειτα αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει το σημείο στο οποίο η σχέση της αθροιστικής σημαντικότητας και της προσθήκης περισσότερων χαρακτηριστικών προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα. Στη συνάρτηση find_optimal_features_MDA ακολουθείται παρόμοια προσέγγιση. Ειδικότερα, χωρίζεται το σύνολο δεδομένων σε σύνολο εκπαίδευσης και σύνολο τεστ, εκπαιδεύεται το μοντέλο στο σύνολο εκπαίδευσης και υπολογίζεται το

permutation importance μέσω του τεστ συνόλου. Τέλος, αφαιρούνται τα χαρακτηριστικά που έχουν permutation importance μικρότερο του μηδέν και ακολουθείται η ίδια διαδικασία με τη συνάρτηση find_optimal_features για τον καθορισμό των σημαντικών χαρακτηριστικών.

4.3.1 Ψευδοκώδικας υλοποίησης Ensemble Learning

Result: Initialize estimator, n_features, feature_importance, selected_indices
Initialization: estimator ← RandomForestClassifier(), n_features ← None, feature_importance ← None, selected_indices ← None;

Αλγόριθμος 19: Ensemble learning: __init__

Data: X, y, use_mda (default False)
Result: List of selected feature indices
estimator.fit(X, y);
feature_importance ← estimator.feature_importances_; selected_indices ← argsort(feature_importance) in descending order;
if *n_features is not None* **then**
    selected_indices ← selected_indices[:n_features];
else
    if *use_mda is False* **then**
        num ← find_optimal_features();
    else
        num ← find_optimal_features_MDA(X, y);
    end
end
return *selected_indices[:num].tolist()*;

Αλγόριθμος 20: Ensemble learning: fit

Data: X
Result: Data projected onto selected features
return *X[:, selected_indices]*;

Αλγόριθμος 21: Ensemble learning: transform

Data: use_savgol_filter (default True)

Result: Optimal number of features using knee detection

total_importance ← sum(feature_importance);

cumulative_importance ← cumsum(feature_importance[selected_indices]) / total_importance;

if *use_savgol_filter* **then**

    window_length ← minimum of 5 or half the length of cumulative_importance minus one;

    polyorder ← 2;

    cumulative_importance ← apply savgol_filter to smooth cumulative_importance;

end

k ← range from 1 to length of cumulative_importance;

knee_locator ← find knee using concave curve, increasing direction;

if *knee_locator.knee is None* **then**

    Decrease sensitivity and retry until knee found or 20 attempts;

end

if *knee_locator.knee* **then**

    Plot and display graph marking knee point;

end

return *knee_locator.knee*;

Αλγόριθμος 22: Ensemble learning: find_optimal_features

Data: X, y, use_savgol_filter (default True)

Result: Optimal number of features based on permutation importance

Split X, y into X_train, X_test, y_train, y_test using train_test_split;

estimator.fit(X_train, y_train);

result ← permutation_importance of estimator on X_test, y_test;

feature_importance ← mean of result.importances;

sorted_indices ← argsort(feature_importance) in descending order;

negative_indices ← where feature_importance[sorted_indices] < 0;

positive_sorted_indices ← sorted_indices[:len(sorted_indices) -
len(negative_indices)];

if *negative_indices is not empty* **then**

    Print "These features have zero importance, better leave them out",
    negative_indices;

end

sorted_indices ← positive_sorted_indices;

sorted_importances ← feature_importance[sorted_indices];

total_importance ← sum(sorted_importances);

cumulative_importance ← cumsum(sorted_importances) / total_importance;

selected_indices ← sorted_indices;

if *use_savgol_filter* **then**

    window_length ← minimum of 5 or half the length of
    cumulative_importance minus one;

    polyorder ← 2;

    cumulative_importance ← apply savgol_filter to smooth
    cumulative_importance;

end

k ← range from 1 to length of cumulative_importance;

knee_locator ← find knee using concave curve, increasing direction;

if *knee_locator.knee is None* **then**

    Decrease sensitivity and retry until knee found or 20 attempts;

end

if *knee_locator.knee* **then**

    Plot and display graph marking knee point;

end

return *knee_locator.knee*;

Αλγόριθμος 23: Ensemble learning: find_optimal_features_MDA

4.4 Factor Analysis

Η υλοποίηση της μεθόδου Factor Analysis πραγματοποιείται μέσω της κλάσης Factor_analysis. Η υλοποίηση ακολουθεί τη μαθηματική ανάλυση της μεθόδου και επικεντρώνεται στην επέκταση της μεθόδου με σκοπό την αυτοματοποιημένη εύρεση του βέλτιστου αριθμού διαστάσεων. Περιέχει πέντε μεταβλητές κλάσης, μια μεταβλητή για την αποθήκευση των ιδιοδιανυσμάτων ή όπως ονομάζονται από τη μέθοδο loadings, μια για την αποθήκευση των ιδιοτιμών, μια για την αποθήκευση των μετασχηματισμένων δεδομένων, μια για τη διακύμανση θορύβου και μια για το μέσο κάθε χαρακτηριστικού. Η συνάρτηση που υλοποιεί τη μαθηματική μέθοδο σε κώδικα είναι η fit_transform. Δέχεται σαν όρισμα το σύνολο δεδομένων, τον αριθμό μείωσης διαστάσεων, τον τύπο μοντέλου SVD για την επανάληψη σύγκλισης (standard SVD ή randomized SVD), την ανοχή, τον μέγιστο αριθμό επαναλήψεων, την τυχαία κατάσταση, και την περιστροφή. Ο αριθμός μείωσης διαστάσεων μπορεί να πάρει τιμές από $1 < K <$ Αριθμός χαρακτηριστικών προσδιορίζοντας τον αριθμό των διαστάσεων που θα διατηρηθούν. Στη συνέχεια, η συνάρτηση μπαίνει στο κομμάτι της επανάληψης όπου χρησιμοποιείται η μέθοδος SVD, κλιμακώνονται τα δεδομένα, υπολογίζεται η διακύμανση θορύβου και γίνεται έλεγχος αν υπάρχει σύγκλιση. Στη περίπτωση που δεν οριστεί αριθμός μείωσης διαστάσεων στη μέθοδο fit_transform, γίνεται αυτόματος υπολογισμός του ΄βέλτιστου' αριθμού διαστάσεων. Αυτό επιτυγχάνεται μέσω της συνάρτησης find_optimal_n_components. Διατηρούνται οι πρώτες n ιδιοτιμές όπου n είναι ο αριθμός των χαρακτηριστικών του αρχικού συνόλου και υπολογίζεται η αθροιστική διακύμανση. Στη συνέχεια, η μέθοδος κατασκευάζει ένα γράφημα με αυτήν και τη συνεισφορά κάθε νέας διάστασης στην ολική διακύμανση του συνόλου δεδομένων. Αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει και το σημείο στο οποίο η σχέση της αθροιστικής διακύμανσης και της προσθήκης περισσότερων διαστάσεων προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα. Το τελευταίο βήμα της μεθόδου είναι η περιστροφή. Στη περίπτωση που έχει οριστεί περιστροφή καλείται η συνάρτηση rotate_matrix η οποία τελεί τον ανάλογο τύπο περιστροφής (varimax, promax,

oblimin, quartimax).

4.4.1 Ψευδοκώδικας υλοποίησης Factor Analysis

Result: Initialize internal variables for factor analysis

Initialization: loadings ← None, eigenvalues ← None, transformed_data ← None, noise_variance ← None, mean ← None;

Αλγόριθμος 24: Factor analysis: __init__

Data: X

Result: Correlation matrix of X

correlation_matrix ← compute the correlation matrix of X, considering columns as variables;

return *correlation_matrix*;

Αλγόριθμος 25: Factor analysis: calc_correlation_matrix

Data: X

Result: Eigenvectors and eigenvalues of the matrix X

eigenvalues, eigenvectors ← compute eigenvalues and eigenvectors of matrix X using Hermitian matrix properties;

return *eigenvectors*, *eigenvalues*;

Αλγόριθμος 26: Factor analysis: calc_eigenvector_eigenvalues

Data: eigenvalues, eigenvectors

Result: Sorted eigenvectors and eigenvalues

sorted_indices ← argsort eigenvalues in descending order;

sorted_eigenvalues ← eigenvalues[sorted_indices];

sorted_eigenvectors ← normalize eigenvectors[:, sorted_indices];

return *sorted_eigenvectors*, *sorted_eigenvalues*;

Αλγόριθμος 27: Factor analysis: sort_eigenvectors_eigenvalues

Data: dataset

Result: Standardized dataset

mean ← calculate mean of dataset along each feature;

std_deviation ← calculate standard deviation of dataset along each feature;

std_deviation ← replace zeros in std_deviation with 1.0 to avoid division by zero;

standardized_dataset ← (dataset - mean) / std_deviation;

return *standardized_dataset*;

Αλγόριθμος 28: Factor analysis: standardize_data

Data: dataset

Result: True if data is numerical, False otherwise

return *dataset.dtype is numerical*;

Αλγόριθμος 29: Factor analysis: is_data_numerical

Data: X

Result: Dataset transformed based on factor loadings and noise variance

Wpsi ← loadings / (noise_variance + 1e-12);

Ih ← identity matrix of size loadings.shape[0];

cov_z ← inverse(Ih + dot(Wpsi, loadings.T));

tmp ← dot(X, Wpsi.T);

X_transformed ← dot(tmp, cov_z);

return *X_transformed*;

Αλγόριθμος 30: Factor analysis: transform

Data: X, k (optional), svd_type (optional), tol, max_iter, random_state, rotation (optional)

Result: Transformed data or the number of components

Adjust X by subtracting its mean;

Initialize parameters for convergence checking and iterations;

for *each iteration until max_iter* **do**

  Update and check convergence criteria;

  If converged or if iterations exceed max_iter, break;

end

Apply rotations if specified;

return *transformed data or number of components based on eigenvalues*;

Αλγόριθμος 31: Factor analysis: fit_transform

Data: rotation (optional)

Result: Rotated loadings if a rotation method is specified

if *rotation is None* **then**

    return *None*;

else

    return *rotate loadings using specified rotation method*;

end

Αλγόριθμος 32: Factor analysis: rotate_matrix

Data: eigenvalues, features_number, use_savgol_filter (optional)

Result: Optimal number of components based on knee point detection

Filter and adjust eigenvalues for valid numerical operations;

Calculate cumulative variance and apply smoothing if specified;

Detect knee point using cumulative variance curve;

return *detected knee point or indicate failure to detect knee*;

Αλγόριθμος 33: Factor analysis: find_optimal_n_components

4.5 Fast ICA

Η συγκεκριμένη υλοποίηση βασίζεται στη μέθοδο ICA και αποτελεί μια παραλλαγή της βασικής μεθόδου. Η συνάρτηση __init__ δέχεται σαν όρισμα τον αριθμό μείωσης διαστάσεων και την ανοχή. Περιέχει επτά μεταβλητές κλάσης, τη n_components και tol που αποθηκεύουν τα δύο ορίσματα του αριθμού μείωσης διαστάσεων και της ανοχής, τη mean που αποθηκεύει τη μέση τιμή κάθε χαρακτηριστικού, τη W η οποία αντιπροσωπεύει τον πίνακα των ανεξάρτητων σημάτων, τη whiten_solver που καθορίζει τη μέθοδο καθαρισμού του σήματος και τις Vt και Vs που αποθηκεύουν τους υποπίνακες της μεθόδου SVD. Η κύρια συνάρτηση είναι η fit_transform η οποία τελεί τον μετασχηματισμό των δεδομένων. Δέχεται σαν όρισμα το σύνολο δεδομένων, τον μέγιστο αριθμό επαναλήψεων (για τη σύγκλιση), και τη συνάρτηση μη-γραμμικότητας που θα χρησιμοποιηθεί. Στη συνέχεια, η μέθοδος κάνει τον καθαρισμό μέσω της μεθόδου SVD ή της PCA. Ακολουθείται μια μορφή επανάληψης στην οποία ενημερώνεται ο πίνακας των ανεξάρτητων σημάτων έως να επιτευχθεί σύγκλιση ή ξεπεραστεί ο μέγιστος αριθμός επαναλήψεων. Κατά τη διάρκεια της επανάληψης χρησιμοποιούνται οι συναρτήσεις sym_decorrelation που εξασφαλίζει ότι ο πίνακας των ανεξάρτητων σημάτων παραμένει ορθογώνιος και η συνάρτηση update η οποία εφαρμόζει τη συνάρτηση μη γραμμικότητας και υπολογίζει τη διαβάθμιση για την ενημέρωση του πίνακα ανεξάρτητων σημάτων.

4.5.1 Ψευδοκώδικας υλοποίησης Fast ICA

Result: Initialize internal variables for Fast ICA

Initialization: n_components ← 2, tol ← 1e-4, mean_ ← None, W ← None, whiten ← None, whiten_solver ← 'svd', Vt_ ← None, S_ ← None;

Αλγόριθμος 34: Fast ICA: __init__

Data: W

Result: Decorrelated W

K ← W @ W.T;

s, U ← eigenvalue decomposition of K;

W_new ← (U @ diag(1.0 / sqrt(s)) @ U.T) @ W;

return *W_new*;

Αλγόριθμος 35: Fast ICA: sym_decorrelation

Data: data, n_components, max_iter, g_type

Result: Transformed source signals matrix S

mean_ ← calculate mean of data along features;

X_centered ← data - mean_; **if** *whiten_solver == 'svd'* **then**

    U, S, Vt ← perform SVD on X_centered, full_matrices=False;

    S_ ← S[:n_components];

    Vt_ ← Vt[:n_components, :];

    K ← (U / S)[:,:n_components];

    X_whitened ← U[:,:n_components] * sqrt(n_samples);

else

    X_whitened ← fit and transform X_centered using PCA with whitening;

end

W ← initialize randomly(n_components, size of X_whitened);

for *iteration from 1 to max_iter* **do**

    W_old ← copy W;

    for *i from 1 to n_components* **do**

        W[i, :] ← update(W[i, :], X_whitened, g_type);

    end

    W ← sym_decorrelation(W);

    if *has_converged(W_old, W, tol)* **then**

        break;

    end

end

S ← compute sources (W @ X_whitened.T).T;

return *S*;

Αλγόριθμος 36: Fast ICA: fit_transform

Data: X

Result: Transformed data S

X_centered ← X - mean_; **if** *whiten_solver == 'svd'* **then**

  X_whitened ← (X_centered @ Vt_.T) / S_;

end

X_whitened ← transform X_centered using stored whitening transform;

S ← compute sources (W @ X_whitened.T).T;

return *S*;

Αλγόριθμος 37: Fast ICA: transform

Data: W_old, W_new, tol

Result: Boolean indicating if the algorithm has converged

delta_W ← compute Frobenius norm of (W_old - W_new);

return *delta_W < tol*;

Αλγόριθμος 38: Fast ICA: has_converged

Data: n_components, random_state

Result: Randomly initialized matrix W

rng ← initialize random number generator with seed random_state;

w_init ← generate normal random numbers for matrix (n_components, n_components);

w_init ← normalize each column of w_init;

return *w_init*;

Αλγόριθμος 39: Fast ICA: initialize_randomly

Data: $wi, X\_white,$ type

Result: Updated weight vector wi

$wiX \leftarrow X\_white@wi$;

switch *type* **do**

  case *'logcosh'* **do**

    $g \leftarrow \tanh(wiX)$;

    $g\_prime \leftarrow 1 - g^2$;

  end

  case *'exp'* **do**

    $g \leftarrow \exp(-wiX^2/2)$;

    $g\_prime \leftarrow wiX \times g$;

  end

  case *'cube'* **do**

    $g \leftarrow wiX^3$;

    $g\_prime \leftarrow 3 \times wiX^2$;

  end

end

$grad\_wi \leftarrow (X\_white^\top @g - \text{mean}(g\_prime) \times wi)/\text{size of } X\_white$;

return *normalize*$(grad\_wi)$;

Αλγόριθμος 40: Fast ICA: Update

4.6 Isomap

Η υλοποίηση της μεθόδου Isomap πραγματοποιείται μέσω της κλάσης isomap. Η συγκεκριμένη υλοποίηση ακολουθεί την υλοποίηση της βιβλιοθήκης Scikit learn η οποία χρησιμοποιεί ένα μοντέλο Kernel PCA για την επίλυση του προβλήματος ιδιοτιμών. Περιέχει τέσσερις μεταβλητές κλάσης, τη nbrs που αποθηκεύει το μοντέλο των κοντινότερων γειτόνων, τη scaler για την κανονικοποίηση των δεδομένων την g η οποία αποθηκεύει τον πίνακα για τον μετασχηματισμό νέων άγνωστων δεδομένων και την kernel_pca για την αποθήκευση του μοντέλου Kernel PCA. Η κύρια συνάρτηση είναι η fit_transform η οποία ακολουθεί τη θεωρητική προσέγγιση της μεθόδου. Δέχεται σαν όρισμα το σύνολο δεδομένων, τον αριθμό μείωσης διαστάσεων και τον αριθμό των κοντινότερων γειτόνων. Βρίσκει τους κοντινότερους γείτονες και κατασκευάζει το γράφημα κοντινότερων γειτόνων, ελέγχοντας παράλληλα ότι το γράφημα είναι συνδεδεμένο. Στην περίπτωση που δεν είναι συνδεδεμένο το γράφημα αυξάνει τον αριθμό των γειτόνων και επαναλαμβάνει τη διαδικασία μέχρι να κατασκευάσει ένα συνδεδεμένο γράφημα. Έπειτα μέσω του μοντέλου Kernel PCA υπολογίζει τον μετασχηματισμό. Στη συνέχεια, επιστρέφεται από τη μέθοδο το μετασχηματισμένο σύνολο δεδομένων. Στην περίπτωση που δεν οριστεί αριθμός μείωσης διαστάσεων στη μέθοδο fit_transform, γίνεται αυτόματος υπολογισμός του ΄βέλτιστου' αριθμού διαστάσεων. Αυτό επιτυγχάνεται μέσω της συνάρτησης find_optimal_n_components. Ειδικότερα, καλείται η Kernel PCA με αριθμό μείωσης διαστάσεων ίσο με τον αριθμό των χαρακτηριστικών. από αυτήν εξάγονται οι ιδιοτιμές και τα ιδιοδιανύσματα. Τέλος, μέσω της συνάρτησης find_optimal_n_components υπολογίζεται η αθροιστική διακύμανση και κατασκευάζεται ένα γράφημα με αυτήν και τη συνεισφορά κάθε νέας διάστασης στην ολική διακύμανση του συνόλου δεδομένων. Αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει το σημείο στο οποίο η σχέση της αθροιστικής διακύμανσης και της προσθήκης περισσότερων διαστάσεων προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα.

4.6.1 Ψευδοκώδικας υλοποίησης Isomap

Result: Initialize internal variables for Isomap

Initialization: imputer ← SimpleImputer(strategy='mean'), nbrs ← None,
scaler ← StandardScaler(), kernel_pca ← None, g ← None;

Αλγόριθμος 41: Isomap: __init__

Data: X, n_components, n_neighbors, variance_threshold,
best_num_of_components
Result: Embedded coordinates or number of components
scaler ← fit and transform X;
nbrs ← fit NearestNeighbors to X;
W ← kneighbors graph of X;
distances ← compute shortest path in W;
if *distances contain inf or NaN* **then**
Correct by adjusting n_neighbors until a connected graph is achieved;
else
C
end
ompute double centered matrix K using distances;
if *n_components is None* **then**
Fit KernelPCA and find optimal n_components;
return *n_components*;
else
if *variance_threshold is not None* **then**
Determine n_components using variance threshold;
else
if *n_components* $<$ *1* **then**
Determine n_components using proportion of variance;
else
Fit KernelPCA with n_components;
end
end
end
embedding ← transform K using KernelPCA;
return *embedding*;

Αλγόριθμος 42: Isomap: fit_transform

Data: X

Result: Transformed coordinates using fitted Isomap

Transform X using scaler and nbrs;

Compute geodesic distances for the new points;

Transform distances into new coordinates using KernelPCA;

return *new coordinates*;

Αλγόριθμος 43: Isomap: transform

Data: eigenvalues, variance_threshold

Result: Optimal number of components

Compute cumulative variance;

Determine number of components where increase in variance falls below threshold;

return *number of components*;

Αλγόριθμος 44: Isomap: find_components_with_variance_threshold

Data: eigenvalues, variance

Result: Number of components

Compute cumulative variance;

Determine number of components necessary to cover the specified variance;

return *number of components*;

Αλγόριθμος 45: Isomap: find_components_with_variance

Data: eigenvalues, use_savgol_filter

Result: Optimal number of components based on knee detection

Apply filter and detect knee in cumulative variance;

return *detected knee as the optimal number of components*;

Αλγόριθμος 46: Isomap: find_optimal_n_components

4.7 Kendall's Tau Correlation

Η υλοποίηση της μεθόδου Kendall Tau Correlation πραγματοποιείται μέσω της κλάσης Kendalls_Tau_Correlation. Η συνάρτηση correlation_coefficient υλοποιεί τη θεωρητική προσέγγιση της μεθόδου σε κώδικα. Δέχεται σαν όρισμα το σύνολο δεδομένων και τον πίνακα του χαρακτηριστικού κλάσης. Μέσω της συνάρτησης calculate_concordant_discordant υπολογίζονται τα σύμφωνα και ασύμφωνα ζεύγη μεταξύ των στοιχείων του συνόλου δεδομένων. Η μέθοδος ties_count υπολογίζει τις ισοπαλίες μεταξύ των τάξεων των ζευγών των στοιχείων. Μια άλλη σημαντική συνάρτηση είναι η feature_selection η οποία υπολογίζει τον συντελεστή κατάταξης συσχέτισης μεταξύ κάθε χαρακτηριστικού με το χαρακτηριστικό κλάσης. Εκτός από το σύνολο δεδομένων και τον πίνακα στοιχείων του χαρακτηριστικού κλάσης δέχεται σαν όρισμα και μια τρίτη μεταβλητή η οποία καθορίζει το κατώφλι για την επιλογή των χαρακτηριστικών. Ειδικότερα, υπολογίζεται για κάθε χαρακτηριστικό και το χαρακτηριστικό κλάσης ο συντελεστής Kendall Tau και έπειτα συγκρίνεται με τη μεταβλητή κατωφλίου για την απόρριψη ή όχι του χαρακτηριστικού. Τέλος υπάρχει η συνάρτηση calculate_feature_correlations η οποία υπολογίζει τόσο τον συντελεστή κατάταξης συσχέτισης μεταξύ κάθε χαρακτηριστικού με το χαρακτηριστικό κλάσης όσο και όλων των χαρακτηριστικών μεταξύ τους.

4.7.1 Ψευδοκώδικας υλοποίησης Kendall Tau Correlation

Initialization: This is a placeholder class initialization.;

Αλγόριθμος 47: Kendall's Tau Correlation: __init__

```
Data: x, y
Result: Returns the number of concordant and discordant pairs
if length of x is not equal to length of y then
    Print "Features and class feature should have the same length";
else
    Initialize concordant and discordant to 0;
    for i from 0 to n-2 do
        for j from i+1 to n-1 do
            diff ← (x[i] - x[j]) * (y[i] - y[j]);
            if diff > 0 then
                concordant ← concordant + 1;
            else
                if diff < 0 then
                    discordant ← discordant + 1;
                end
            end
        end
    end
    return concordant, discordant;
end
```

Αλγόριθμος 48: Kendall's Tau Correlation: calculate_concordant_discordant

```
Data: values
Result: Count of ties in the data
Initialize val_count as an empty dictionary;
for value in values do
    if value is in val_count then
        val_count[value] ← val_count[value] + 1;
    else
        val_count[value] ← 1;
    end
end
Initialize tie_count to 0;
for count in val_count.values do
    if count > 1 then
        tie_count ← tie_count + count * (count - 1) / 2;
    end
end
return tie_count;
```

Αλγόριθμος 49: Kendall's Tau Correlation: ties_count

Data: x, y

Result: Kendall's Tau correlation coefficient

if *length of x is not equal to length of y* **then**

    Print "Features and class feature should have the same length";

else

    if *y is not numerical* **then**

        Encode y using OrdinalEncoder;

    else

        c

    end

    oncordant, discordant ← calculate_concordant_discordant(x, y);

    n1 ← ties_count(x);

    n2 ← ties_count(y);

    n0 ← n * (n - 1) / 2;

    tau ← (concordant - discordant) / sqrt((n0 - n1) * (n0 - n2));

    return *tau*;

end

Αλγόριθμος 50: Kendall's Tau Correlation: correlation_coefficient

Data: X, y, threshold

Result: List of selected features based on correlation threshold

Initialize correlations as an empty list;

for *i from 0 to number of columns in X* **do**

    feature ← X[:, i];

    tau ← correlation_coefficient(feature, y);

    if *absolute value of tau > threshold* **then**

        Append i to selected_features;

        Append "correlation for feature i: tau" to correlations;

    end

end

return *selected features, correlations*;

Αλγόριθμος 51: Kendall's Tau Correlation: feature_selection

Data: features

Result: Correlation matrix of features

Initialize correlations as a zero matrix of shape (n, n);

for *i from 0 to n-1* **do**

    for *j from i to n-1* **do**

        if *i* == *j* **then**

            correlations[i, j] ← 1.0;

        else

            x ← features[:, i];

            y ← features[:, j];

            coef ← correlation_coefficient(x, y);

            correlations[i, j] ← coef;

            correlations[j, i] ← coef;

        end

    end

end

return *correlations*;

Αλγόριθμος 52: Kendall's Tau Correlation: calculate_feature_correlations

4.8 Kernel PCA

Η υλοποίηση της μεθόδου Kernel PCA πραγματοποιείται μέσω της κλάσης Kernel_PCA. Η υλοποίηση ακολουθεί τη μαθηματική ανάλυση της μεθόδου και επικεντρώνεται στην επέκταση της μεθόδου με σκοπό την αυτοματοποιημένη εύρεση του βέλτιστου αριθμού διαστάσεων. Περιέχει επτά μεταβλητές κλάσης, δύο για την αποθήκευση των ιδιοδιανυσμάτων και ιδιοτιμών, μια για την αποθήκευση των μετασχηματισμένων δεδομένων, μια για τη συνάρτηση πυρήνα και τρεις μεταβλητές οι οποίες αποτελούν υπερπαραμέτρους που χρειάζονται οι συναρτήσεις πυρήνα. Η συνάρτηση που υλοποιεί τη μαθηματική μέθοδο σε κώδικα είναι η fit_transform. Δέχεται σαν όρισμα το σύνολο δεδομένων και τον αριθμό μείωσης διαστάσεων. Ο αριθμός μείωσης διαστάσεων μπορεί να πάρει τιμές από $1 < K <$ Αριθμός χαρακτηριστικών προσδιορίζοντας τον αριθμό των διαστάσεων που θα διατηρηθούν ή $0 < K < 1$, προσδιορίζοντας το ποσοστό της διακύμανσης που θα διατηρηθεί. Ο μετασχηματισμός των δεδομένων μέσω της συνάρτησης πυρήνα πραγματοποιείται μέσω της συνάρτησης kernel_functions, ο υπολογισμός και το κεντράρισμα του πίνακα gram γίνεται μέσω της συνάρτησης calc_gram_matrix και center_gram_matrix. Υπάρχουν επίσης συναρτήσεις για τον υπολογισμό των ιδιοδυνασμάτων και ιδιοτιμών καθώς επίσης και η συνάρτηση transform για τον μετασχηματισμό των υπάρχοντων ή και νέων δεδομένων. Στην περίπτωση που δεν οριστεί η παράμετρος αριθμός μείωσης διαστάσεων γίνεται αυτόματος υπολογισμός του ΄βέλτιστου' αριθμού διαστάσεων. Αυτό επιτυγχάνεται μέσω της συνάρτησης find_optimal_n_components. Η συνάρτηση υπολογίζει την αθροιστική διακύμανση. Στη συνέχεια, κατασκευάζει ένα γράφημα με αυτήν και τη συνεισφορά κάθε νέας διάστασης στην ολική διακύμανση του συνόλου δεδομένων. Αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει το σημείο στο οποίο η σχέση της αθροιστικής διακύμανσης και της προσθήκης περισσότερων διαστάσεων προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα.

4.8.1 Ψευδοκώδικας υλοποίησης Kernel PCA

Result: Initialize Kernel PCA with specified parameters

Initialization: kernel_type ← 'linear', gamma ← None, coef0 ← 1, degree ← 3, eigenvectors ← None, eigenvalues ← None, transformed_data ← None;

Αλγόριθμος 53: Kernel PCA: __init__

Data: X, y

Result: Compute kernel function based on the specified kernel type

if *kernel_type == 'linear'* **then**
    return *np.dot(X, y)*;
else
    if *kernel_type == 'polynomial'* **then**
        Set gamma to 1.0 / X.shape[0] if gamma is None;
        return *(gamma * np.dot(X, y) + coef0) ** degree*;
    else
        if *kernel_type == 'rbf'* **then**
            Set gamma to 1.0 / X.shape[0] if gamma is None;
            return *np.exp(-gamma * np.linalg.norm(X - y) ** 2)*;
        else
            if *kernel_type == 'sigmoid'* **then**
                Set gamma to 1.0 / X.shape[0] if gamma is None;
                return *np.tanh(gamma * np.dot(X, y) + coef0)*;
            else
                Print "Please provide a valid kernel type";
            end
        end
    end
end

Αλγόριθμος 54: Kernel PCA: kernel_functions

Data: X, Y (optional)

Result: Gram matrix based on the kernel function

if *Y is None* **then**

    Y ← X;

end

n_samples_X ← X.shape[0];

n_samples_Y ← Y.shape[0];

Initialize gram_matrix to zeros of shape (n_samples_X, n_samples_Y);

for *i from 0 to n_samples_X-1* **do**

    for *j from 0 to n_samples_Y-1* **do**

        gram_matrix[i, j] ← kernel_functions(X[i], Y[j]);

    end

end

return *gram_matrix*;

Αλγόριθμος 55: Kernel PCA: calc_gram_matrix

Data: X, k (optional), use_savgol_filter

Result: Transformed data or number of components

gram_matrix ← calc_gram_matrix(X);

gram_matrix_centered ← center_gram_matrix(gram_matrix);

eigenvectors, eigenvalues ←
calc_eigenvector_eigenvalues(gram_matrix_centered);

eigenvectors, eigenvalues ← sort_eigenvectors_eigenvalues(eigenvalues,
eigenvectors);

if *k is None* **then**

    k ← find_optimal_n_components(eigenvalues, use_savgol_filter);

    return *k*;

else

    if *k < 1* **then**

        k ← find_k_based_on_variance_rate(eigenvalues, k);

    end

    if *len(eigenvalues) < k* **then**

        Raise Error "K must be smaller than the number of attributes that the dataset has";

    end

    eigenvectors ← eigenvectors[:, :k];

    eigenvalues ← eigenvalues[:k];

    transformed_data ← transform_data(gram_matrix_centered);

    return *transformed_data*;

end

Αλγόριθμος 56: Kernel PCA: fit_transform

Data: X

Result: New coordinates in the transformed space

gram_matrix ← calc_gram_matrix(X, train_data);

gram_matrix_centered ← center_gram_matrix(gram_matrix, training=False);

X_new ← compute new coordinates using the trained kernel PCA model;

return *X_new*;

Αλγόριθμος 57: Kernel PCA: transform

Data: K, training

Result: Centered gram matrix

if *training* **then**

    Compute and subtract row means and column means, add total mean;

else

    Subtract training means from new gram matrix;

end

return *centered gram matrix*;

Αλγόριθμος 58: Kernel PCA: center_gram_matrix

Data: dataset

Result: Standardized dataset

dataset ← fit and transform dataset using StandardScaler;

return *dataset*;

Αλγόριθμος 59: Kernel PCA: standardize_data

Data: cov_matrix

Result: Covariance matrix of the dataset

covariance_matrix ← np.cov(cov_matrix, rowvar=False);

return *covariance_matrix*;

Αλγόριθμος 60: Kernel PCA: calc_covariance_matrix

Data: dataset

Result: Eigenvectors and eigenvalues of the dataset

eigenvalues, eigenvectors ← np.linalg.eig(dataset);

return *eigenvectors*, *eigenvalues*;

Αλγόριθμος 61: Kernel PCA: calc_eigenvector_eigenvalues

Data: eigenvalues, eigenvectors

Result: Real parts of eigenvectors and eigenvalues

eigenvalues_real ← np.real(eigenvalues);

eigenvectors_real ← np.real(eigenvectors);

return *eigenvectors_real*, *eigenvalues_real*;

Αλγόριθμος 62: Kernel PCA: discard_im_part

Data: eigenvalues, eigenvectors

Result: Sorted eigenvectors and eigenvalues in descending order of eigenvalues

sorted_indices ← np.argsort(eigenvalues)[::-1];

sorted_eigenvalues ← eigenvalues[sorted_indices];

sorted_eigenvectors ← eigenvectors[:, sorted_indices];

return *sorted_eigenvectors*, *sorted_eigenvalues*;

Αλγόριθμος 63: Kernel PCA: sort_eigenvectors_eigenvalues

Data: eigenvalues, variance_rate

Result: Number of components based on a specified variance rate

total_var ← np.sum(eigenvalues);

cumulative_var ← np.cumsum(eigenvalues) / total_var;

k ← 1;

while *cumulative_var[k - 1] < variance_rate* **do**

  k ← k + 1;

end

return *k*;

Αλγόριθμος 64: Kernel PCA: find_k_based_on_variance_rate

Data: eigenvalues, features_number, use_savgol_filter

Result: Optimal number of components based on knee detection

eigenvalues ← eigenvalues[:features_number];

total_var ← np.sum(eigenvalues);

cumulative_var ← np.cumsum(eigenvalues) / total_var;

if *use_savgol_filter* **then**

    window_length ← min(5, len(cumulative_var) //2 * 2 - 1);

    polyorder ← 2;

    cumulative_var ← savgol_filter(cumulative_var, window_length,

     polyorder);

end

k ← np.arange(1, len(cumulative_var) + 1);

knee_locator ← KneeLocator(k, cumulative_var, curve='concave',

 direction='increasing');

if *knee_locator.knee is None* **then**

    Decrease sensitivity and retry until knee found or 20 attempts;

end

if *knee_locator.knee* **then**

    Plot knee point with vertical line at knee_locator.knee;

end

return *knee_locator.knee*;

Αλγόριθμος 65: Kernel PCA: find_optimal_n_components

4.9 LDA

Η υλοποίηση της μεθόδου LDA πραγματοποιείται μέσω της κλάσης LDA. Η υλοποίηση ακολουθεί τη μαθηματική ανάλυση της μεθόδου και επικεντρώνεται στην επέκταση της μεθόδου με σκοπό την αυτοματοποιημένη εύρεση του βέλτιστου αριθμού διαστάσεων. Περιέχει τρεις μεταβλητές κλάσης, δύο για την αποθήκευση των ιδιοδιανυσμάτων και ιδιοτιμών και μια για την αποθήκευση των μετασχηματισμένων δεδομένων. Η συνάρτηση που υλοποιεί τη μαθηματική προσέγγιση σε κώδικα είναι η fit_transform. Δέχεται σαν όρισμα το σύνολο δεδομένων, τον πίνακα των στοιχείων του χαρακτηριστικού κλάσης και τον αρθμό μείωσης διαστάσεων. Ο αριθμός μείωσης διαστάσεων μπορεί να πάρει τιμές από $1 < K <$ Αριθμός χαρακτηριστικών προσδιορίζοντας τον αριθμό των διαστάσεων που θα διατηρηθούν ή $0 < K < 1$, προσδιορίζοντας το ποσοστό της γραμμικής διακριτότητας που θα διατηρηθεί. Μέσω των συναρτήσεων calc_within_class_scatter_matrix και calc_between_class_scatter_matrix υπολογίζονται οι πίνακες διασποράς και μέσω των calc_eigenvector_eigenvalues και sort_eigenvectors_eigenvalues υπολογίζονται τα ιδιοδιανύσματα και οι ιδιοτιμές. Όταν ισχύει ότι $0 < K < 1$ τότε καλείται η συνάρτηση find_k_based_on_discriminant_power_rate για την εφαρμογή της μεθόδου. Στην περίπτωση που δεν οριστεί η παράμετρος αριθμός μείωσης διαστάσεων γίνεται αυτόματος υπολογισμός του ΄βέλτιστου' αριθμού διαστάσεων. Αυτό επιτυγχάνεται μέσω της συνάρτησης find_optimal_components. Η συνάρτηση υπολογίζει την αθροιστική διακριτότητα. Στη συνέχεια, κατασκευάζει ένα γράφημα με αυτήν και τη συνεισφορά κάθε νέας διάστασης στην ολική διακριτότητα του συνόλου δεδομένων. Αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει και το σημείο στο οποίο η σχέση της αθροιστικής διακριτότητας και της προσθήκης περισσότερων διαστάσεων προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα.

4.9.1 Ψευδοκώδικας υλοποίησης LDA

Result: Initialize LDA class variables

Initialization: eigenvectors ← None, eigenvalues ← None, transformed_data ← None, explained_variance_ratio ← None;

Αλγόριθμος 66: LDA: __init__

```
Data: dataset, labels, k
Result: Transformed data or return optimal k
if not is_data_numerical(dataset) then
    Print "Data must be numerical!";
    return;
end
if k is None then
    return My_LDA_fit_best_k(dataset, labels);
else
    if k > min(dataset.shape[1], length(unique(labels))) - 1 then
        Print "k must be smaller than the number of features - 1";
        return;
    end
    if not np.issubdtype(type(labels), np.number) then
        Print "Encoding labels";
    else
        Print "Labels are already numeric";
    end
    if k < 1 then
        k ← find_k_based_on_discriminant_power_rate(eigenvalues, k,
          dataset.shape[1], labels);
    else
        S_W ← calc_within_class_scatter_matrix(dataset, labels);
        S_B ← calc_between_class_scatter_matrix(dataset, labels);
        matrix ← inverse(S_W) * S_B;
        eigenvectors, eigenvalues ← calc_eigenvector_eigenvalues(matrix);
        eigenvectors, eigenvalues ←
          sort_eigenvectors_eigenvalues(eigenvectors, eigenvalues);
    end
    eigenvectors ← eigenvectors[:, :k];
    eigenvalues ← eigenvalues[:k];
    transformed_data ← transform(dataset, eigenvectors);
    return transformed_data;
end
```

Αλγόριθμος 67: LDA: fit_transform

Data: features, labels

Result: Within-class scatter matrix S_W

S_W ← zero matrix of shape (n_features, n_features);

for *label in unique(labels)* **do**

    class_samples ← select samples with label;

    class_mean ← mean of class_samples;

    deviations ← class_samples - class_mean;

    covariance_matrix ← deviations.T * deviations;

    S_W ← S_W + covariance_matrix;

end

return *S_W*;

Αλγόριθμος 68: LDA: calc_within_class_scatter_matrix

Data: features, labels

Result: Between-class scatter matrix S_B

overall_mean ← mean of features;

S_B ← zero matrix of shape (n_features, n_features);

for *label in unique(labels)* **do**

    class_samples ← select samples with label;

    class_mean ← mean of class_samples;

    n ← number of samples with label;

    mean_diff ← class_mean - overall_mean;

    S_B ← S_B + n * (mean_diff * mean_diff.T);

end

return *S_B*;

Αλγόριθμος 69: LDA: calc_between_class_scatter_matrix

Data: dataset

Result: Transformed dataset using LDA

if *number of features in eigenvectors does not match number of columns in dataset* **then**

    Print "Number of features in eigenvectors must match the number of columns in the dataset.";

    return;

end

transformed_data ← dataset * eigenvectors;

return *transformed_data*;

Αλγόριθμος 70: LDA: transform

Data: use_savgol_filter

Result: Optimal number of components based on cumulative variance

cumulative_var ← cumsum(explained_variance_ratio);

if *use_savgol_filter* **then**

    Apply smoothing to cumulative_var;

end

knee_locator ← find knee in cumulative_var;

return *knee_locator.knee*;

Αλγόριθμος 71: LDA: find_optimal_components

Data: dataset

Result: Boolean indicating if dataset is numerical

return *np.issubdtype(dataset.dtype, np.number)*;

Αλγόριθμος 72: LDA: is_data_numerical

Data: discriminant_power_rate, n_features, class_labels

Result: Optimal number of components k based on discriminant power rate

k ← 1;

while *k <= length of explained_variance_ratio and explained_variance_ratio[k-1] < discriminant_power_rate* **do**

    if *k > min(n_features, length(unique(class_labels))) - 1* **then**

        k ← k - 1;

        break;

    else

        k ← k + 1;

    end

end

return *k*;

Αλγόριθμος 73: LDA: find_k_based_on_discriminant_power_rate

Data: eigenvalues, eigenvectors

Result: Sorted eigenvectors and eigenvalues in descending order of eigenvalues

sorted_indices ← np.argsort(eigenvalues)[::-1];

sorted_eigenvalues ← eigenvalues[sorted_indices];

sorted_eigenvectors ← eigenvectors[:, sorted_indices];

return *sorted_eigenvectors, sorted_eigenvalues*;

Αλγόριθμος 74: LDA: sort_eigenvectors_eigenvalues

Data: dataset, labels

Result: Number of components k that maximizes the discriminant power

if *not is_data_numerical(dataset)* **then**

    Print ”Data must be numerical!”;

    return;

end

S_W ← calc_within_class_scatter_matrix(dataset, labels);

S_B ← calc_between_class_scatter_matrix(dataset, labels);

S_W_inv ← np.linalg.pinv(S_W);

matrix ← S_W_inv * S_B;

eigenvectors, eigenvalues ← calc_eigenvector_eigenvalues(matrix);

eigenvectors, eigenvalues ← sort_eigenvectors_eigenvalues(eigenvalues, eigenvectors);

total ← sum(eigenvalues);

explained_variance_ratio ← [(i / total) for i in sorted(eigenvalues, reverse=True)];

k ← find_optimal_components();

eigenvectors ← eigenvectors[:, :k];

eigenvalues ← eigenvalues[:k];

transformed_data ← transform(dataset);

return *k*;

Αλγόριθμος 75: LDA: My_LDA_fit_best_k

Data: dataset

Result: calc_eigenvector_eigenvalues

eigenvalues, eigenvectors ← np.linalg.eig(dataset);

return *eigenvectors*, *eigenvalues*;

Αλγόριθμος 76: LDA: Calculate Eigenvectors and Eigenvalues

4.10 LLE

Η υλοποίηση της μεθόδου LLE πραγματοποιείται μέσω της κλάσης Locally_Linear_Embedding. Η υλοποίηση ακολουθεί τη μαθηματική προσέγγιση της μεθόδου και επικεντρώνεται στην επέκταση της μεθόδου με σκοπό την αυτοματοποιημένη εύρεση του βέλτιστου αριθμού διαστάσεων. Περιέχει τέσσερις μεταβλητές κλάσης, δύο για την αποθήκευση των κοντινότερων γειτόνων και τον αριθμό μείωσης διαστάσεων, μια για την αποθήκευση των μετασχηματισμένων δεδομένων και μια για την αποθήκευση του μοντέλου εύρεσης των κοντινότερων γειτόνων. Η συνάρτηση που υλοποιεί τη μαθηματική μέθοδο σε κώδικα είναι η fit_transform και δέχεται σαν ορίσματα το σύνολο των χαρακτηριστικών και μια παράμετρο κανονικοποίησης. Ειδικότερα, καλείται η barycenter_kneighbors_graph που κατασκευάζει τον πίνακα βαρών μεταξύ των στοιχείων και η compute_embedding που υπολογίζει τις ιδιοτιμές και τα ιδιοδιανύσματα. Στην περίπτωση που δεν οριστεί η παράμετρος αριθμός μείωσης διαστάσεων γίνεται αυτόματος υπολογισμός του βέλτιστου' αριθμού διαστάσεων. Αυτό επιτυγχάνεται μέσω της συνάρτησης find_optimal_n_components. Η συνάρτηση υπολογίζει την αθροιστική τιμή όλων των ιδιοτιμών. Στην προκειμένη περίπτωση γίνεται αναζήτηση των ιδιοτιμών που είναι όσο το δυνατόν πιο κοντά στο μηδέν. Στη συνέχεια, κατασκευάζει ένα γράφημα με αυτήν και τη συνεισφορά κάθε νέας διάστασης στην ολική αθροιστική τιμή των ιδιοτιμών του συνόλου δεδομένων. Ειδικότερα, η προσθήκη κάθε νέας ιδιοτιμής αυξάνει το ποσοστό μεταξύ των επιλεγμένων ιδιοτιμών προς τη συνολική αθροιστική τιμή όλων των ιδιοτιμών. Έτσι, αξιοποιώντας τη βιβλιοθήκη KneeLocator υπολογίζεται το σημείο αγκώνας (elbow point), το οποίο αντικατοπτρίζει και το σημείο στο οποίο η σχέση της συνολικής αθροιστικής τιμής και της προσθήκης περισσότερων διαστάσεων (δηλαδή η προσθήκη επιπλέον ιδιοτιμών) προσφέρει δυσανάλογο κέρδος. Για τον εντοπισμό του σημείου αγκώνα εφαρμόζεται το φίλτρο Savitzky–Golay για να βελτιώσει τη γραφική παράσταση, με σκοπό τον καλύτερο υπολογισμό του σημείου αγκώνα.

4.10.1 Ψευδοκώδικας υλοποίησης LLE

Result: Initialize LLE class variables

Initialization: n_neighbors ← 12, n_components ← None, embedding ← None, nbrs ← None;

Αλγόριθμος 77: LLE: __init__

Data: X, reg

Result: Embedding of the data

W ← barycenter_kneighbors_graph(X, n_neighbors, reg);

eigenvalues, eigenvectors ← compute_embedding(X, W, n_components);

if *n_components is None* **then**

    num_of_features ← X.shape[1];

    eigenvalues ← eigenvalues[1:num_of_features + 1];

    n_components ← find_optimal_components(eigenvalues, True);

    return *n_components*;

else

    embedding ← eigenvectors[:, 1:n_components + 1];

    return *embedding*;

end

Αλγόριθμος 78: LLE: fit_transform

Data: X, n_neighbors, reg, n_jobs

Result: Weight matrix W

knn ← NearestNeighbors(n_neighbors=n_neighbors + 1, n_jobs=n_jobs).fit(X);

X ← knn._fit_X;

n_samples ← X.shape[0];

ind ← knn.kneighbors(X, return_distance=False)[:, 1:];

Y_neighbors ← X[ind];

data ← barycenter_weights(X, Y_neighbors, reg);

indptr ← np.arange(0, n_samples * n_neighbors + 1, n_neighbors);

return *csr_matrix((data.ravel(), ind.ravel(), indptr), shape=(n_samples, n_samples))*;

Αλγόριθμος 79: LLE: barycenter_kneighbors_graph

Data: X, W, n_components, random_state

Result: Eigenvalues and eigenvectors

n_samples ← X.shape[0];

I ← eye(n_samples, format='csr');

I_minus_W ← I - W;

if *I_minus_W is not csr_matrix* **then**

    I_minus_W ← csr_matrix(I_minus_W);

end

M ← I_minus_W.T.dot(I_minus_W);

if *n_components is None* **then**

    n_components ← X.shape[1];

else

    k

end

← n_components + 1;

v0 ← random_state.uniform(-1, 1, M.shape[0]);

return *eigsh(M, k, sigma=0.0, tol=1e-6, maxiter=100, v0=v0)*;

Αλγόριθμος 80: LLE: compute_embedding

Data: eigenvalues, use_savgol_filter

Result: Optimal number of components based on eigenvalues

total_var ← np.sum(eigenvalues);

cumulative_var ← np.cumsum(eigenvalues) / total_var;

if *use_savgol_filter* **then**

    Apply savgol_filter to smooth the cumulative variance curve;

end

k ← knee_locator on cumulative_var;

return *k*;

Αλγόριθμος 81: LLE: find_optimal_components

Data: X_new, reg

Result: Transformed new data

distances, indices ← nbrs.kneighbors(X_new, n_neighbors=12);

neighbors ← nbrs._fit_X[indices];

weights ← barycenter_weights(X_new, neighbors, reg);

X_transformed ← np.zeros((X_new.shape[0], n_components));

for *i from 0 to X_new.shape[0]-1* **do**

    X_transformed[i] ← np.dot(weights[i], embedding[indices[i]]);

end

return *X_transformed*;

Αλγόριθμος 82: LLE: transform

Data: X, neighbors, reg

Result: Weights for reconstructing each point from its neighbors

n_samples ← X.shape[0];

n_neighbors ← neighbors.shape[1];

Initialize weights as zero matrix of shape (n_samples, n_neighbors);

for *i from 0 to n_samples-1* **do**

    Z ← neighbors[i] - X[i];

    C ← np.dot(Z, Z.T);

    if *C.dtype is integer and reg is float* **then**

        C ← convert C to float64;

    end

    C ← C + np.eye(n_neighbors) * reg;

    w ← solve linear system Cw = 1;

    weights[i] ← w / sum(w);

end

return *weights*;

Αλγόριθμος 83: LLE: barycenter_weights

4.11 Spearman's Rank Correlation

Η υλοποίηση της μεθόδου Spearman's Rank Correlation πραγματοποιείται μέσω της κλάσης Spearman_Rank_Correlation. Η υλοποίηση ακολουθεί τη μαθηματική προσέγγιση της μεθόδου. Η μέθοδος calculate_rank αναθέτει σε κάθε στοιχείο των δεδομένων το ν αριθμό κατάταξης (rank) του και στην περίπτωση ισοπαλίας αναθέτει τον μέσο όρο μεταξύ των δύο ranks. Η συνάρτηση correlation_coefficient δέχεται σαν όρισμα το σύνολο δεδομένων και τον πίνακα στοιχείων του χαρακτηριστικού κλάσης και είναι αυτή που υπολογίζει τον συντελεστή εφαρμόζοντας τον μαθηματικό τύπο της μεθόδου μέσω κώδικα. Η feature_selection είναι η συνάρτηση επιλογής των χαρακτηριστικών, εκτός από τα κοινά ορίσματα με τη feature_selection δέχεται ένα επιπλέον όρισμα το οποίο αποτελεί το κατώφλι για την επιλογή του εκάστοτε χαρακτηριστικού ή όχι. Ειδικότερα υπολογίζει τους συντελεστές κατάταξης Spearman μεταξύ κάθε χαρακτηριστικού και του χαρακτηριστικού κλάσης και βάση του κατωφλίου απορρίπτει ή όχι κάθε χαρακτηριστικό. Τέλος η συνάρτηση calculate_feature_correlations υπολογίζει τους συντελεστές κατάταξης Spearman μεταξύ όλων των χαρακτηριστικών αλλά και μεταξύ του κάθε χαρακτηριστικού και του χαρακτηριστικού κλάσης.

4.11.1 Ψευδοκώδικας υλοποίησης Spearman's Rank Correlation

Initialization: ;

Αλγόριθμος 84: Spearman's Rank Correlation: __init__

```
Data: data
Result: Ranks of the data
data_np ← convert data to array;
sorted_indices ← argsort(data_np);
ranks ← zero array of size equal to data_np;
i ← 0;
while i < length of data_np do
    j ← i;
    while j + 1 < length of data_np and data_np[sorted_indices[j]] ==
     data_np[sorted_indices[j + 1]] do
        j ← j + 1;
    end
    avg_rank ← 1 + (i + j) / 2.0;
    for k from i to j do
        ranks[sorted_indices[k]] ← avg_rank;
    end
    i ← j + 1;
end
return ranks;
```

Αλγόριθμος 85: Spearman's Rank Correlation: calculate_rank

```
Data: x, y
Result: Spearman rank correlation coefficient
if length of x is not equal to length of y then
    Print "x and y must have the same length";
end
rank_x ← calculate_rank(x);
rank_y ← calculate_rank(y);
squared_sum ← sum((rank_x - rank_y) ** 2);
n ← length of x;
rank_coeff ← 1 - (6 * squared_sum) / (n * (n**2 - 1));
return rank_coeff;
```

Αλγόριθμος 86: Spearman's Rank Correlation: correlation_coefficient

Data: X, y, threshold

Result: Indices of selected features based on threshold

if *threshold is None* **then**

    threshold ← 0;

else

end

for $i \leftarrow 0$ **to** `numcols(`X`) - 1` **do**

    feature ← X[:, i];

    feature_cor ← correlation_coefficient(feature, y);

    if *abs(feature_cor)* $>$ *threshold* **then**

        selected_features[*Append*]i;

        correlations[*Append*]correlation for feature i: feature_cor;

    end

end

Print correlations;

return selected_features;

Αλγόριθμος 87: Spearman's Rank Correlation: feature_selection

```
Data: features
Result: Correlation matrix of features
correlations ← zero matrix of size (n, n);
for i from 0 to n - 1 do
    for j from i to n - 1 do
        if i == j then
            correlations[i, j] ← 1.0;
        else
            coef ← correlation_coefficient(features[:, i], features[:, j]);
            correlations[i, j] ← coef;
            correlations[j, i] ← coef;
        end
    end
end
return correlations;
```

Αλγόριθμος 88: Spearman's Rank Correlation: calculate_feature_correlations

4.12 Boruta

Η υλοποίηση της μεθόδου Boruta πραγματοποιείται μέσω της κλάσης Boruta. Η υλοποίηση ακολουθεί τη θεωρητική προσέγγιση της μεθόδου. Περιέχει πέντε μεταβλητές κλάσης, μια για την αποθήκευση του μοντέλου ταξινομητή εκμάθησης συνόλου που θα χρησιμοποιηθεί, μια για το ποσοστό κατωφλίου σημαντικότητας των χαρακτηριστικών και μια για την αποθήκευση της σημαντικότητας κάθε χαρακτηριστικού. Η συνάρτηση που υλοποιεί τη μαθηματική μέθοδο σε κώδικα είναι η fit. Δέχεται σαν ορίσματα το σύνολο των στοιχείων των χαρακτηριστικών, τα στοιχεία του χαρακτηριστικού κλάσης, τον μέγιστο αριθμό επαναλήψεων, δύο παραμέτρους που λειτουργούν σαν παράμετροι για τη διαδικασία της διωνυμικής δοκιμής (binomial test) και μια παράμετρο που καθορίζει πια στατιστική μέθοδος θα χρησιμοποιηθεί για την αξιολόγηση των χαρακτηριστικών. Η κατασκευή των χαρακτηριστικών σκιάς πραγματοποιείται μέσω της create_shadow_features, η εκπαίδευση του συνόλου δεδομένων γίνεται μέσω της συνάρτησης train_tree_estimator. Υπάρχουν δύο στατιστικές μέθοδοι για την αξιολόγηση των χαρακτηριστικών η binomial_test_with_bh_correction και η binomial_test_with_bonferroni_correction. Η binomial_test_with_bh_correction χρησιμοποιεί τη μέθοδο Bonferroni correction για την αξιολόγηση των χαρακτηριστικών, ενώ η binomial_test_with_bh_correction χρησιμοποιεί την Benjamini-Hochberg correction. Τέλος για την ενημέρωση της κατάταξης των χαρακτηριστικών σύμφωνα με τη σημαντικότητα τους χρησιμοποιείται η update_ranking.

4.12.1 Ψευδοκώδικας υλοποίησης Boruta

Data: estimator, perc

Result: Initialize Boruta class variables

if *estimator is None* **then**

  estimator ← RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42);

end

Initialize perc ← perc, estimator ← estimator, support_ ← empty array, ranking ← empty array, feature_importances ← empty array;

Αλγόριθμος 89: Boruta: __init__

Data: X, y, max_iter, p, alpha, two_step

Result: Significant features after multiple hypothesis testing

n_features ← X.shape[1];

feature_importances ← zero array of length n_features;

ranking ← zero array of length n_features, type integer;

for *i from 0 to max_iter-1* **do**

    X_updated ← create_shadow_features(X);

    importances ← train_tree_estimator(X_updated, y);

    shadow_feature_indices ← range(n_features, 2 * n_features);

    feature_importances += importances[:n_features];

    shadow_importances ← importances[shadow_feature_indices];

    update_ranking(shadow_importances, i);

end

if *two_step* **then**

    significant_features ← binomial_test_with_bh_correction(max_iter, p, alpha);

else

    significant_features ← binomial_test_with_bonferroni_correction(max_iter, p, alpha);

end

return *significant_features*;

Αλγόριθμος 90: Boruta: fit

Data: X

Result: Updated X with shadow features

X_shadow ← shuffle(X);

X_updated ← hstack(X, X_shadow);

return *X_updated*;

Αλγόριθμος 91: Boruta: create_shadow_features

Data: X, y

Result: Feature importances from the estimator

estimator ← clone(estimator);

estimator.fit(X, y);

return *estimator.feature_importances_*;

Αλγόριθμος 92: Boruta: train_tree_estimator

Data: shadow_importances, iter

Result: Update ranking of features

threshold ← percentile of shadow_importances at perc;

for *i from 0 to length of feature_importances - 1* **do**

    if *feature_importances[i] / (iter + 1) > threshold* **then**

        ranking[i] += 1;

    end

end

Αλγόριθμος 93: Boruta: update_ranking

Data: max_iter, p, alpha

Result: Significant features after Bonferroni correction

significant_features ← empty list;

p_values ← empty list;

for *i, successes in enumerate(ranking)* **do**

    result ← binomtest(successes, max_iter, p, alternative='greater');

    p_values.append(result.pvalue);

end

p_values ← np.array(p_values);

adjusted_alpha ← alpha / length of p_values;

for *i, p_value in enumerate(p_values)* **do**

    if *p_value < adjusted_alpha* **then**

        significant_features.append(i);

    end

end

return *significant_features*;

Αλγόριθμος 94: Boruta: binomial_test_with_bonferroni_correction

```
Data: max_iter, p, alpha
Result: Significant features after BH correction
significant_features ← empty list;
p_values ← empty list;
for i, successes in enumerate(ranking) do
    result ← binomtest(successes, max_iter, p, alternative='greater');
    p_values.append((i, result.pvalue));
end
p_values.sort by second element;
m ← length of p_values;
prev_bh_value ← 1;
for i, (original_index, p_value) in enumerate(p_values) do
    bh_threshold ← (i + 1) / m * alpha;
    if p_value <= bh_threshold then
        prev_bh_value ← p_value;
        significant_features.append(original_index);
    end
    else if p_value > bh_threshold then
        break;
    end
end
significant_features ← [idx for idx, p in p_values if p <= prev_bh_value];
return sorted(significant_features);
```

Αλγόριθμος 95: Boruta: binomial_test_with_bh_correction

Κεφάλαιο 5

Υπολογιστική μελέτη

Η αυξημένη πολυπλοκότητα και ο μεγάλος όγκος δεδομένων αποτελούν συχνά προβλήματα των συνόλων δεδομένων και ειδικότερα των συνόλων δεδομένων που εξάγονται από πραγματικά πειράματα. Μια από τις αποτελεσματικότερες μεθόδους διαχείρισης αυτών των προβλημάτων είναι οι μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Ο κύριος στόχος της παρούσας υπολογιστικής μελέτης είναι η βαθύτερη ανάλυση των μεθόδων αυτών, ο έλεγχος της αποτελεσματικότητας τους και οι επιδράσεις που έχουν στην ενίσχυση της επίδοσης των τελικών μοντέλων πρόβλεψης.

5.1 Μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών

Στη εργασία χρησιμοποιήθηκαν τόσο μέθοδοι μείωσης διαστάσεων όσο και μέθοδοι επιλογής χαρακτηριστικών. Ειδικότερα, οι μέθοδοι μείωσης διαστάσεων που χρησιμοποιήθηκαν ήταν οι μέθοδοι PCA, SVD, Factor Analysis, LDA, Kernel PCA, ICA, Isomap και LLE. Από τον τομέα της επιλογής χαρακτηριστικών χρησιμοποιήθηκαν ο αλγόριθμος Boruta, μέθοδοι εκμάθησης συνόλου για επιλογή χαρακτηριστικών, ο συντελεστής συσχέτισης κατάταξης Spearman και ο συντελεστής συσχέτισης κατάταξης Kendall. Οι συγκεκριμένες μέθοδοι επιλέχθηκαν λόγω της ευρείας εφαρμογής και αποτελεσματικότητας τους σε προηγούμενες έρευνες και λόγω ότι προσεγγίζουν τη μείωση διαστάσεων και επιλογή χαρακτηριστικών με διαφορετικά κριτήρια και μεθοδολογίες.

5.2 Σύνολα δεδομένων

Για την αξιολόγηση των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών χρησιμοποιήθηκαν επτά διαφορετικά σύνολα δεδομένων. Όλα τα σύνολα δεδομένων που χρησιμοποιήθηκαν είναι δημόσια, προέρχονται από δημοφιλή αποθετήρια, αποτελούνται από αριθμητικά δεδομένα, δεν έχουν ελλιπείς τιμές και αφορούν προβλήματα ταξινόμησης. Ειδικότερα, χρησιμοποιήθηκε το Load Digits Dataset [87] το οποίο περιέχει 64 χαρακτηριστικά και 1,797 στοιχεία, το Wine Dataset [88] το οποίο περιέχει 13 χαρακτηριστικά και 178, το Breast Cancer Dataset [89] το οποίο περιέχει 30 χαρακτηριστικά και 569 στοιχεία, το Dry Bean Dataset [90] το οποίο περιέχει 16 χαρακτηριστικά και 13,611 στοιχεία, το Ionosphere Dataset [91] το οποίο περιέχει 34 χαρακτηριστικά και 351 στοιχεία, το Connectionist Bench Dataset [92] το οποίο περιέχει 60 χαρακτηριστικά και 208 στοιχεία και το Musk Dataset [93] το οποίο περιέχει 168 χαρακτηριστικά και 476 στοιχεία.

5.3 Αλγόριθμοι ταξινόμησης

Στο κομμάτι της ταξινόμησης, ο στόχος ήταν να χρησιμοποιηθούν απλοί ταξινομητές, οι οποίοι αξιοποιούν διαφορετικές προσεγγίσεις για την εξαγωγή των τελικών αποτελεσμάτων. Αναλυτικότερα χρησιμοποιήθηκαν από τη βιβλιοθήκη Scikit Learn έξι ταξινομητές, ο KNN (K-Nearest Neighbors), ο SVM (Support Vector Machine), Decision Tree, ο Random Forest, ο Naive Bayes και ο MLP (Multi-Layer Perceptron). Σχετικά με τις παραμέτρους των ταξινομητών ο KNN λάμβανε υπόψιν τους πέντε κοντινότερους γείτονες και η μετρική της απόστασης μεταξύ των γειτόνων ήταν η ευκλείδεια απόσταση. Ο SVM είχε παράμετρο C = 1, ως συνάρτηση πυρήνα τον πυρήνα rbf, και η παράμετρος gamma είχε τιμή scale. Στον ταξινομητή Decision Tree το κριτήριο για την κατασκευή του δέντρου ήταν το gini impurity. Ο Random Forest είχε 100 ταξινομητές και σαν κριτήριο οι ταξινομητές χρησιμοποιούσαν το gini impurity. Για το μοντέλο του Naive Bayes χρησιμοποιήθηκε το μοντέλο κανονικής κατανομής (GaussianNB) και η παράμετρος var smoothing είχε τιμή 10^{-8}. Τέλος, το μοντέλο του πολυεπίπεδου Perceptron αποτελούνταν από δύο κρυφά επίπεδα, το πρώτο με 100 νευρώνες και το δεύτερο με 50, είχε συνάρτηση ενεργοποίησης τη relu, αλγόριθμο βελτιστοποίησης τον adam, παράμετρο alpha = 0.001, ρυθμό

εκμάθησης = 0.01 και μέγιστο αριθμό επαναλήψεων 200.

5.4 Πειραματική διαδικασία

Η πειραματική διαδικασία χωρίστηκε σε 2 κομμάτια. Στο πρώτο κομμάτι κατασκευάστηκαν 2 αρχεία τεστ, ένα για τις μεθόδους μείωσης διαστάσεων και ένα για τις μεθόδους επιλογής χαρακτηριστικών. Στο αρχείο που αφορά τη μείωση διαστάσεων ορίστηκαν οι έξι ταξινομητές με τις υπερπαραμέτρους τους, επίσης ορίστηκαν τα μονοπάτια για καθένα από τα σύνολα δεδομένων μαζί με τους κατάλληλους μετασχηματισμούς ώστε τα δεδομένα να είναι συμβατά με όλες τις μεθόδους. Για παράδειγμα η μετατροπή των ονομαστικών τιμών του συνόλου δεδομένων Dry Bean σε αριθμητικές για την εφαρμογή της μεθόδου LDA. Ορίστηκαν αντικείμενα από όλες τις μεθόδους μείωσης διαστάσεων και κατασκευάστηκε μια σύνθετη δομή επανάληψης. Στη δομή επανάληψης μελετήθηκαν για κάθε σύνολο δεδομένων, κάθε ταξινομητή και κάθε μέθοδο μείωσης διαστάσεων όλες οι διαστάσεις πλην της τελευταίας. Ειδικότερα, για κάθε σύνολο δεδομένων χρησιμοποιήθηκε η μέθοδος k cross validation (με k = 4) και σε κάθε διαχωρισμό των δεδομένων εφαρμόστηκε η εκάστοτε μέθοδος μείωσης διαστάσεων στο υποσύνολο εκπαίδευσης. Έπειτα βάση των δεδομένων εκπαίδευσης έγινε ο μετασχηματισμός των δεδομένων τεστ. Στη συνέχεια, έγινε η εκπαίδευση του κάθε ταξινομητή με το υποσύνολο εκπαίδευσης και υπολογίστηκαν μέσω του υποσυνόλου τεστ το accuracy, recall, precision και f1 score. Η ίδια διαδικασία επαναλήφθηκε για κάθε διαχωρισμό του cross validation, κατά τον οποίο επαναλήφθηκαν και τα βήματα εφαρμογής της μεθόδου μείωσης διαστάσεων, εκπαίδευσης των ταξινομητών και υπολογισμού των μετρικών. Η παραπάνω διαδικασία έγινε για όλες τις μεθόδους μείωσης διαστάσεων και για όλα τα σύνολα δεδομένων. Τα τελικά αποτελέσματα αποτελούνταν από τις αποδόσεις των έξι ταξινομητών για κάθε διάσταση και για κάθε σύνολο δεδομένων.

Παρόμοιο αρχείο τεστ κατασκευάστηκε και για την αξιολόγηση των μεθόδων επιλογής χαρακτηριστικών. Συγκεκριμένα χρησιμοποιήθηκαν τέσσερις μέθοδοι, ο αλγόριθμος Boruta, μέθοδοι εκμάθησης συνόλου ως μέθοδοι επιλογής χαρακτηριστικών, ο συντελεστής συσχέτισης κατάταξης του Spearman και ο συντελεστής συσχέτισης κατάταξης του Kendall. Οι μέθοδοι Boruta και εκμάθησης συνόλου έχουν την ιδιότητα ότι χρησιμοποιούν ένα μοντέλο εκμάθησης συνόλου για την αξιολόγηση

των χαρακτηριστικών σε σημαντικά και μη σημαντικά. Στο συγκεκριμένο πείραμα χρησιμοποιήθηκαν δοκιμαστικά τα μοντέλα της βιβλιοθήκης Scikit Learn, Random Forest, Gradient Boosting, AdaBoost, XGBoost, LightGBM και CatBoost. Κατασκευάστηκε η ίδια σύνθετη μορφή επανάληψης με το πρώτο αρχείο. Τα δεδομένα κάθε συνόλου δεδομένων χωριζότανε σε δεδομένα εκπαίδευσης και τεστ. Η κύρια διαφορά στο συγκεκριμένο πείραμα ήταν ότι μετά την επιλογή χαρακτηριστικών γινόταν τροφοδότηση των υποσυνόλων με τα σημαντικά χαρακτηριστικά στους ταξινομητές και εκπαιδεύονταν βάση αυτών. Η αξιολόγηση γινόταν μέσω του υποσυνόλου τεστ όπου και πάλι οι ταξινομητές αξιοποιούσαν μόνο τα σημαντικά χαρακτηριστικά.

Στο δεύτερο κομμάτι της πειραματικής διαδικασίας χρησιμοποιήθηκαν οι τεχνικές αυτόματης εύρεσης του βέλτιστου αριθμού διαστάσεων για κάθε σύνολο δεδομένων όπως περιγράφτηκαν στο κομμάτι των υλοποιήσεων. Ειδικότερα οι μέθοδοι για τις οποίες αναπτύχθηκαν τεχνικές αυτόματης εύρεσης του 'βέλτιστου' αριθμού διαστάσεων ήταν οι PCA, SVD, LDA, Kernel PCA Isomap, LLE, Factor Analysis. Το αρχείο είχε παρόμοια μορφή με αυτήν του πρώτου αρχείου αλλά με μικρές διαφοροποιήσεις. Χρησιμοποιήθηκε η ίδια σύνθετη μορφή επανάληψης. Εσωτερικά χρησιμοποιήθηκε επίσης το k cross validation με το ίδιο seed ώστε ο διαχωρισμός των δεδομένων να είναι ακριβώς ίδιος με αυτόν των δύο προηγούμενων πειραμάτων. Σε κάθε διαχωρισμό υπολογίστηκε ο αριθμός των βέλτιστων διαστάσεων. Τα αποτελέσματα κάθε μεθόδου τροφοδοτήθηκαν στους ταξινομητές μέσω των οποίων έγινε η διαδικασία της εκπαίδευσης και της αξιολόγησης.

5.5 Αποτελέσματα μεθόδων μείωσης διαστάσεων

Τα αποτελέσματα του Πίνακα 5.1 δείχνουν την απόδοση των έξι ταξινομητών στα επτά σύνολα δεδομένων χωρίς να έχει εφαρμοστεί κάποια μέθοδος μείωσης διαστάσεων ή επιλογής χαρακτηριστικών στα δεδομένα. από τα αποτελέσματα παρατηρείται ότι ο ταξινομητής SVM άγγιξε πέντε φορές τη μέγιστη ακρίβεια στα σύνολα δεδομένων Digits, Wine, Breast Cancer, Ionosphere και Dry Bean. Στις επόμενες θέσεις ακολουθούν οι ταξινομητές MLP και Random Forest οι οποίοι έδειξαν επίσης πολύ καλά αποτελέσματα τα οποία είναι αρκετά κοντά στον SVM. Ειδικότερα, ο MLP είχε την ίδια απόδοση με τον SVM στα σύνολα δεδομένων Digits και Breast Cancer, ενώ είχε παράλληλα τη μέγιστη απόδοση στα σύνολα δεδομένων

Accuracy						
Dataset/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
Digits	0.97	**0.98**	0.85	0.97	0.81	**0.98**
Wine	0.97	**0.98**	0.88	**0.98**	**0.98**	0.97
Breast Cancer	0.96	**0.97**	0.94	0.96	0.93	**0.97**
Ionosphere	0.85	**0.94**	0.87	0.93	0.88	0.92
Connectionist Bench	0.77	0.82	0.75	0.83	0.68	**0.85**
Dry Bean	0.92	**0.93**	0.90	0.92	0.90	0.92
Musk	0.87	0.89	0.80	0.90	0.74	**0.93**

Πίνακας 5.1: Πίνακας αποτελεσμάτων χωρίς προεπεξεργασία δεδομένων

Connectionist Bench και Musk τα οποία είχαν και τον μεγαλύτερο αριθμό από χαρακτηριστικά. Ο Random Forest έφτασε τη μέγιστη απόδοση μόνο στο Wine Dataset όμως σε όλα τα σύνολα δεδομένων είχε πολύ μικρές αποκλίσεις συγκριτικά με τις μέγιστες αποδόσεις. Ο ταξινομητής με τη χαμηλότερη απόδοση είναι ο Naive Bayes ο οποίος φαίνεται να έχει σχεδόν σε όλα τα σύνολα δεδομένων χαμηλή απόδοση και ιδιαίτερα στο Connectionist Bench Dataset όπου η απόδοση του πέφτει στο 0.68. Η μεταβλητότητα της απόδοσης των ταξινομητών μεταξύ των συνόλων δεδομένων δείχνει τη σημασία επιλογής του κατάλληλου ταξινομητή ανάλογα με τα χαρακτηριστικά και τις ιδιότητες του συνόλου δεδομένων. Οι ταξινομητές SVM, MLP και Random Forest δείχνουν αρκετά αξιόπιστες επιλογές τόσο για σύνολα με λίγα χαρακτηριστικά όσο και για σύνολα με πολλά χαρακτηριστικά και πιο πολύπλοκες σχέσεις. Πρόκληση φαίνεται να αποτελεί το σύνολο δεδομένων Connectionist Bench, υποδεικνύοντας ότι μπορεί να έχει πιο περίπλοκα ή θορυβώδη δεδομένα. Τέλος, η χαμηλή απόδοση του Naive Bayes δείχνει ότι ο ταξινομητής δεν είναι η καλύτερη επιλογή για σύνολα δεδομένων με πολύπλοκα μοτίβα ή δεδομένα υψηλότερων διαστάσεων.

Accuracy						
Dataset/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
Digits	0.94 SVD (24)	**0.95** SVD (24)	0.85 LLE (38)	0.92 PCA (20)	0.91 LDA (8)	0.94 SVD (20)
Wine	0.98 LDA (2)	**0.99** PCA (5)	**0.99** ICA (5)	0.98 ICA (5)	0.98 Factor Analysis (12)	0.98 Factor Analysis (10)
Breast Cancer	0.97 SVD (12)	**0.98** SVD (12)	0.96 LDA (1)	0.96 LDA (1)	0.97 LDA (1)	0.97 PCA (7)
Ionosphere	0.85 PCA (11)	0.93 Factor Analysis (11)	0.88 Factor Analysis (11)	0.94 Factor Analysis (11)	0.87 PCA (11)	**0.97** Factor Analysis (11)
Connectionist Bench	0.57 PCA (15)	0.59 Factor Analysis (15)	0.59 Isomap (6)	**0.69** Kernel PCA (15)	0.58 Isomap (6)	0.68 LDA (24)
Dry Bean	0.83 SVD (6)	0.83 LLE (12)	0.77 LLE (12)	0.82 SVD (6)	**0.85** SVD (6)	0.80 LDA (5)
Musk	0.75 PCA (29)	**0.77** PCA (29)	0.71 PCA (29)	0.74 PCA (29)	0.75 PCA (29)	**0.77** Kernel PCA (29)

Πίνακας 5.2: Πίνακας αποτελεσμάτων μέσω τεχνικών αυτόματης εύρεσης βέλτιστων διαστάσεων.

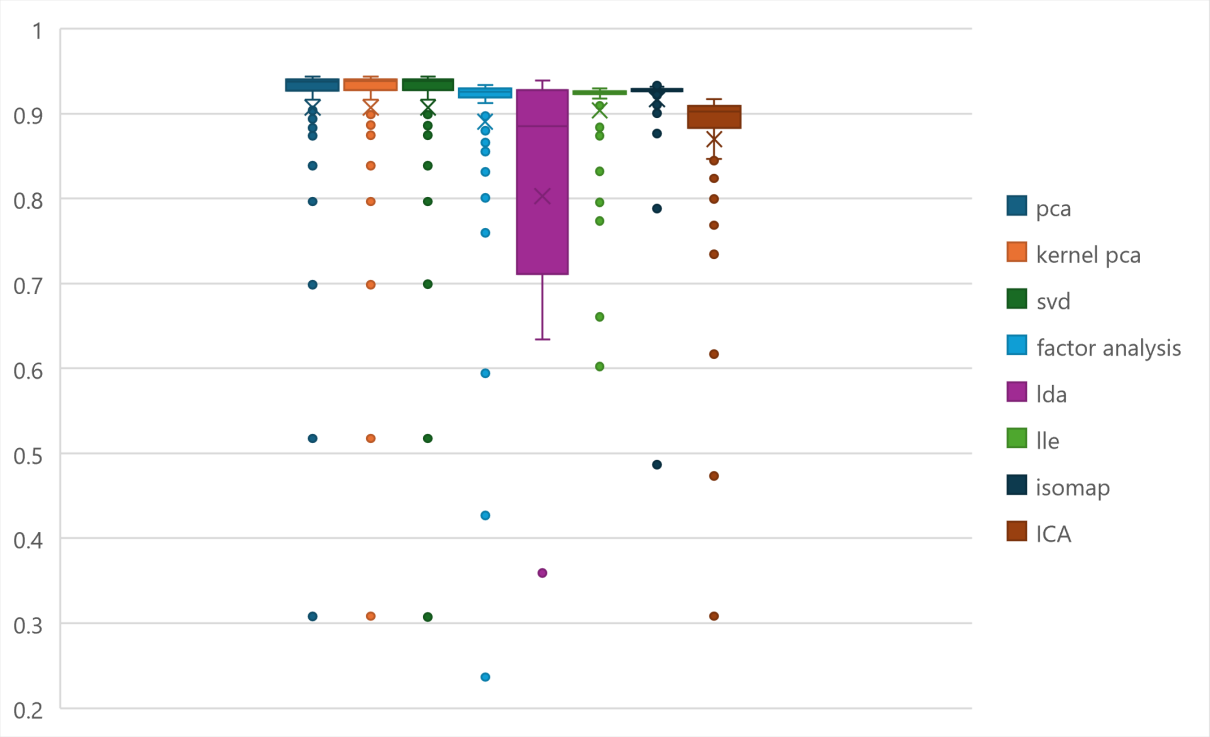
Ο Πίνακας 5.2 περιέχει τις τιμές ακρίβειας που επιτυγχάνει κάθε ταξινομητής μέσω των αυτόματων τεχνικών επιλογής βέλτιστου αριθμού διαστάσεων, σε κάθε σύνολο δεδομένων. από τα αποτελέσματα φαίνεται ότι διαφορετικές μεθοδοι μείωσης διαστάσεων λειτουργούν καλύτερα σε διαφορετικά σύνολα δεδομένων. Γενικότερα συγκριτικά τόσο με τα αποτελέσματα του Πίνακα 5.1 όσο και με τα αποτελέσματα των πινάκων βέλτιστων διαστάσεων για κάθε σύνολο δεδομένων, όπου έγινε επαναληπτική ανάλυση μέσω κάθε μεθόδου για κάθε αριθμό διάστασης μέχρι των αριθμό των χαρακτηριστικών μείον ένα, φαίνεται τα αποτελέσματα του πίνακα να είναι αρκετά κοντά στα βέλτιστα με μικρές αποκλίσεις. Ο ταξινομητής SVM κατέχει και πάλι την κορυφαία απόδοση σε τέσσερα από τα επτά σύνολα δεδομένων, στα Digits, Wine, Breast Cancer και Musk. Μετά την εφαρμογή των μεθόδων μείωσης διαστάσεων σημαντική βελτίωση παρουσιάζει και ο ταξινομητής Naive Bayes συγκριτικά με τα υπόλοιπα αποτελέσματα του πίνακα όπου παρατηρείται ότι οι τιμές ακρίβειας είναι αρκετά κοντά στις τιμές των υπόλοιπων ταξινομητών. Για το σύνολο δεδομένων Digits, ο ταξινομητής SVM με την εφαρμογή της μεθόδου SVD

(24 διαστάσεις) πέτυχε την υψηλότερη ακρίβεια (0,95). Τα σύνολα δεδομένων Wine και Breast Cancer έδειξα υψηλή ακρίβεια σε όλους τους ταξινομητές, με τον SVM μέσω της μεθόδου SVD (12 διαστάσεις) να επιτυγχάνει την υψηλότερη ακρίβεια (0,98) στο Breast Cancer και τους SVM και Decision Tree να μοιράζονται την κορυφή με (0.99) στο Wine Dataset όπου εφαρμόστηκαν οι μέθοδοι PCA (5) και ICA (5), γεγονός που υποδεικνύει ότι μειώνοντας τις διαστάσεις σε λιγότερες από τις μισές διατηρείται σχεδόν όλη η πληροφορία και επιτυγχάνονται πάρα πολύ καλές αποδόσεις. Για το σύνολο δεδομένων Ionosphere, ο ταξινομητής MLP με εφαρμογή της Factor Analysis (11 διαστάσεις) πέτυχε την υψηλότερη ακρίβεια (0,97), υποδεικνύοντας την αποτελεσματικότητα αυτού του συνδυασμού. Χαμηλές αποδόσεις καταγράφτηκαν στο σύνολο δεδομένων Connectionist Bench όπου ο Random Forest πέτυχε την υψηλότερη ακρίβεια (0.69) μέσω της μεθόδου Kernel PCA. Η πολυπλοκότητα των δεδομένων φαίνεται να δυσκολεύει τόσο τους ταξινομητές όσο και τις μεθόδους μείωσης διαστάσεων, οι οποίες φαίνεται να μην μπορούν να καταγράψουν αποτελεσματικά τα χαρακτηριστικά και τις ιδιότητες των δεδομένων με αποτέλεσμα να μην μεταφέρεται μεγάλο ποσοστό της πληρφορίας στο μετασχηματισμένο σύνολο δεδομένων.

Οι μέθοδοι PCA και SVD επιστρέφουν ικανοποιητικά αποτελέσματα και εμφανίζονται πολλές φορές ως οι κορυφαίες μέθοδοι σε πολλούς από τους ταξινομητές στα διάφορα σύνολα δεδομένων. Η Factor Analysis φαίνεται να λειτουργεί αρκετά αποτελεσματικά στο σύνολο δεδομένων Ionosphere όπου είχε τα βέλτιστα αποτελέσματα σε τέσσερις από τους έξι ταξινομητές. Η μέθοδος LDA εμφανίζεται στα πρώτα σύνολα δεδομένων προσφέροντας ικανοποιητικά αποτελέσματα αλλά όσο αυξάνεται ο αριθμός των χαρακτηριστικών και η πολυπλοκότητα των συνόλων αρχίζει να γίνεται λιγότερο αποτελεσματική. Η συγκεκριμένη παρατήρηση μπορεί να οφείλεται στην πολυπλοκότητα των δεδομένων όμως αυτό δεν είναι απόλυτα σίγουρο, πιθανό σενάριο είναι επίσης η χαμηλότερη απόδοση της μεθόδου να οφείλεται και στο χαρακτηριστικό κλάσης και την κατανομή των ετικετών του χαρακτηριστικού μιας και η μέθοδος αξιοποιεί και το χαρακτηριστικό κλάσης για να εξάγει τα τελικά αποτελέσματα.

Best Accuracy						
Method/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
PCA	0.94 (32)	**0.96** (38)	0.78 (17)	0.93 (39)	0.86 (51)	0.95 (44)
Kernel PCA	0.94 (34)	**0.96** (42)	0.78 (22)	0.93 (39)	0.87 (39)	0.95 (45)
SVD	0.94 (34)	**0.96** (42)	0.77 (19)	0.93 (31)	0.86 (39)	0.95 (21)
Factor Analysis	0.93 (28)	**0.96** (46)	0.78 (31)	0.93 (40)	0.88 (50)	0.94 (31)
LDA	**0.94** (9)	0.93 (9)	0.84 (8)	0.91 (9)	0.92 (9)	0.92 (9)
PCA	0.91 (20)	**0.94** (29)	0.75 (22)	0.89 (16)	0.84 (15)	0.93 (17)
LLE	0.93 (52)	**0.94** (42)	0.92 (33)	**0.94** (60)	0.87 (49)	**0.94** (20)
Isomap	0.93 (9)	**0.94** (43)	0.89 (14)	0.93 (63)	0.91 (17)	0.93 (56)

Πίνακας 5.3: Πίνακας αποτελεσμάτων βέλτιστων τιμών Digits Dataset



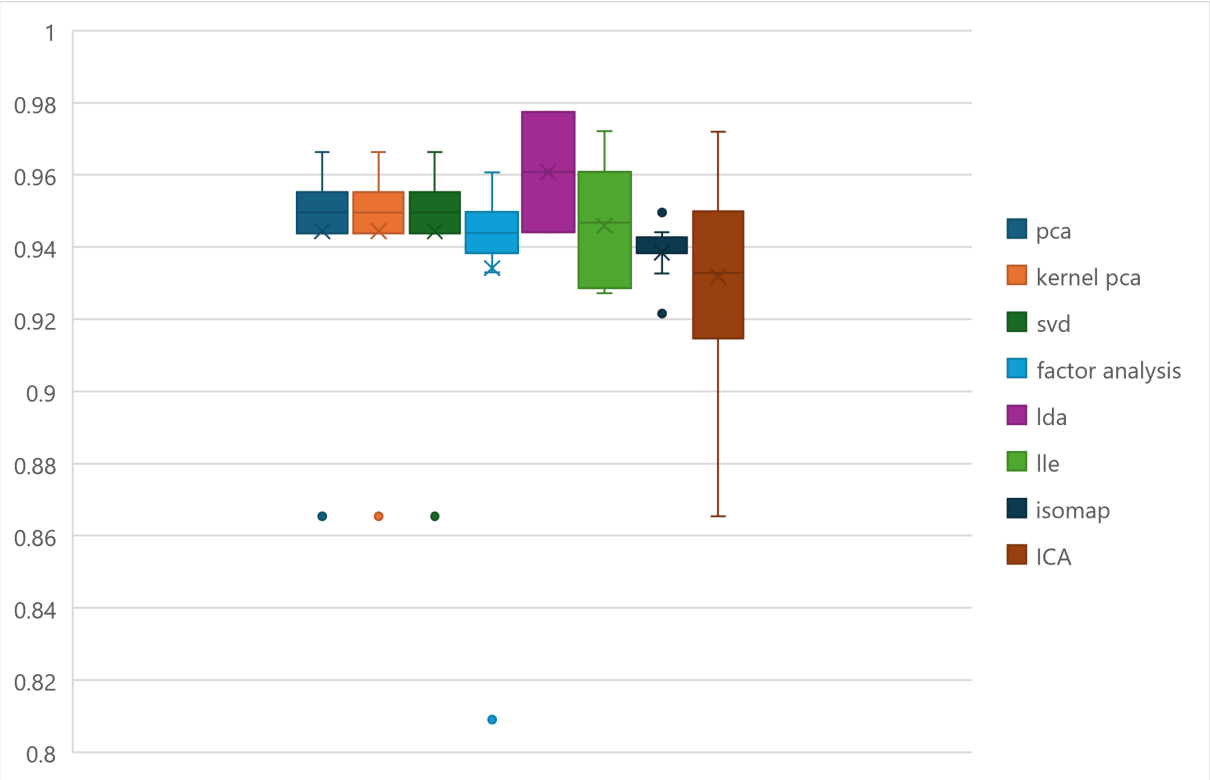
Σχήμα 5.1: Digits Dataset Box Plot

Ο Πίνακας 5.3 περιέχει τις βέλτιστες τιμές ακρίβειας και τον αριθμό διαστάσεων για κάθε μέθοδο και κάθε ταξινομητή στο σύνολο δεδομένων Digits. από τα αποτελέσματα είναι εμφανές ότι την καλύτερη ακρίβεια έχει ο ταξινομητής SVM, ο οποίος έχει την υψηλότερη ακρίβεια σε επτά από τις οκτώ μεθόδους ταξινόμησης. Πολύ κοντά επίσης βρίσκεται και ο ταξινομητής MLP και με ελάχιστα μικρότερες τιμές ακρίβειας ακολουθεί ο ταξινομητής KNN. Τα αποτελέσματα μετά τη χρήση των μεθόδων μείωσης διαστάσεων αν και μείωσαν κατά πολύ την πολυπλοκότητα ελάττωσαν σε μικρό βαθμό και την ακρίβεια των ταξινομητών στην πληθώρα των περι-

πτώσεων. Εξαιρέσεις αποτελούν οι ταξινομητές Decision Tree και Naive Bayes. Στον ταξινομητή Decision Tree οι μέθοδοι LLE και Isomap εκτός από την απλοποίηση του συνόλου δεδομένων βελτίωσαν την απόδοση του ταξινομητή. Όμοια συμπεριφορά παρατηρείται και στον ταξινομητή Naive Bayes όπου όλες οι μέθοδοι αύξησαν την ακρίβεια ταξινόμησης. από το γράφημα των Box plots για κάθε μέθοδο παρατηρείται ότι οι μέθοδοι PCA, Kernel PCA και SVD είχαν τη μεγαλύτερη μέση ακρίβεια συγκριτικά με τις υπόλοιπες μεθόδους, γεγονός που τονίζει τη σταθερότητα των μεθόδων στον μετασχηματισμό των δεδομένων. Αντίθετα, η μέθοδος ICA φαίνεται να είχε τη μικρότερη μέση ακρίβεια, το οποίο έχει μια λογική βάση μιας και το σύνολο δεδομένων απαρτίζεται από τιμές που σχηματίζουν 8x8 pixel εικόνες και η μέθοδος αναζητεί να βρει ανεξάρτητα σήματα μέσω των οποίων σχηματίζονται τα δεδομένα.

Best Accuracy						
Method/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
PCA	0.97 (2)	**0.99** (5)	0.94 (4)	0.97 (7)	0.96 (2)	0.97 (6)
Kernel PCA	0.97 (2)	**0.99** (5)	0.94 (3)	0.96 (5)	0.96 (2)	0.97 (6)
SVD	0.97 (2)	**0.99** (5)	0.94 (4)	0.96 (7)	0.96 (2)	0.97 (6)
Factor Analysis	0.96 (5)	**0.99** (11)	0.94 (5)	0.97 (8)	0.98 (12)	0.98 (10)
LDA	**0.98** (2)	0.97 (2)	0.92 (2)	0.97 (2)	0.97 (2)	0.97 (2)
LLE	**0.97** (4)	**0.97** (4)	0.95 (12)	0.96 (12)	**0.97** (1)	0.96 (2)
Isomap	0.95 (4)	**0.98** (6)	0.95 (7)	0.96 (11)	0.97 (3)	0.97 (6)
ICA	0.97 (2)	0.98 (5)	**0.99** (5)	0.98 (5)	0.98 (5)	0.97 (11)

Πίνακας 5.4: Πίνακας αποτελεσμάτων βέλτιστων τιμών Wine Dataset

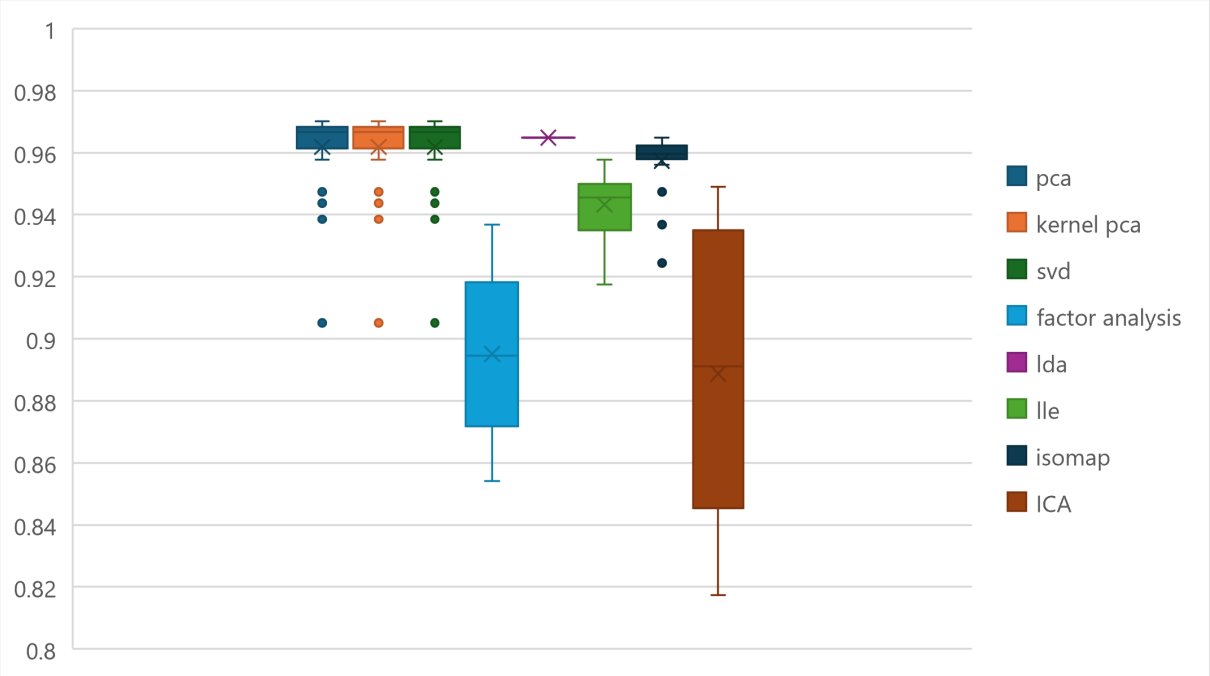


Σχήμα 5.2: Wine Dataset Box Plot

Ο Πίνακας 5.4 περιέχει τις βέλτιστες τιμές ακρίβειας και τον αριθμό διαστάσεων για κάθε μέθοδο και κάθε ταξινομητή στο σύνολο δεδομένων Wine. Στο συγκεκριμένο σύνολο δεδομένων παρατηρείται καλύτερα το στοιχείο της βελτίωσης της ποιότητας των δεδομένων και κατ' επέκταση της απόδοσης των ταξινομητών μετά την εφαρμογή μεθόδων μείωσης διαστάσεων. Ειδικότερα όλοι οι ταξινομητές έχουν ίδια ή και καλύτερη ακρίβεια σε σχέση με τα αποτελέσματα του Πίνακα 5.1. Όπως και στα προηγούμενα σύνολα δεδομένων και εδώ φαίνεται ότι την υψηλότερη απόδοση την έχει ο ταξινομητής SVM, ο οποίος επιτυγχάνει τη βέλτιστη απόδοση σε έξι από τις οκτώ μεθόδους. Οι υπόλοιποι ταξινομητές αν και έχουν ελαφρώς μικρότερη απόδοση παρουσιάζουν πολύ καλά αποτελέσματα μεταξύ των μεθόδων. Παράλληλα, οι διαστάσεις στις οποίες μειώθηκαν τα χαρακτηριστικά είναι λιγότερες από τις μισές. Το γεγονός αυτό δείχνει ότι πολλές φορές η απλοποίηση των περιττών δεδομένων αποτελεί το κλειδί για την κατασκευή ενός πολύ καλού μοντέλου, με ελάχιστες απαιτήσεις σε μνήμη. από το σχεδιάγραμμα των Box plots παρατηρείται ότι τη μεγαλύτερη μέση ακρίβεια είχε η μέθοδος LDA, ενώ για άλλη μια φορά τη μικρότερη είχε η μέθοδος ICA.

Best Accuracy						
Method/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
PCA	0.97 (11)	**0.98** (10)	0.93 (11)	0.95 (19)	0.92 (8)	**0.98** (12)
Kernel PCA	0.97 (11)	**0.98** (10)	0.94 (7)	0.95 (4)	0.93 (5)	**0.98** (12)
SVD	0.97 (10)	**0.98** (10)	0.93 (8)	0.95 (10)	0.92 (5)	**0.98** (13)
Factor Analysis	0.97 (12)	**0.98** (10)	0.96 (12)	0.96 (12)	0.96 (12)	**0.98** (9)
LDA	0.96 (1)	0.97 (1)	0.96 (1)	0.96 (1)	**0.97** (1)	**0.97** (1)
LLE	0.96 (14)	0.96 (22)	0.95 (26)	0.95 (19)	0.94 (18)	**0.97** (24)
Isomap	**0.96** (21)	**0.96** (24)	0.93 (14)	0.95 (19)	0.93 (5)	**0.96** (4)
PCA	0.95 (5)	0.96 (9)	0.92 (2)	0.94 (2)	0.93 (5)	**0.97** (12)

Πίνακας 5.5: Πίνακας αποτελεσμάτων βέλτιστων τιμών Breast Cancer Dataset



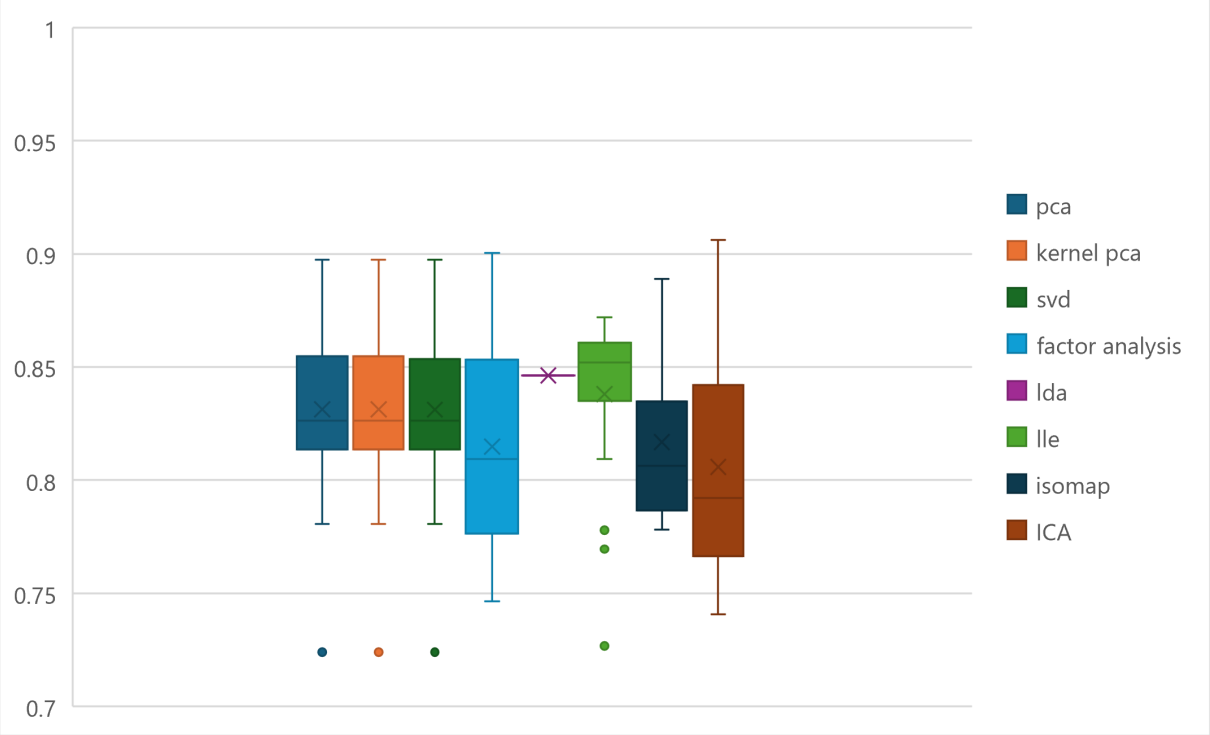
Σχήμα 5.3: Breast Cancer Dataset Box Plot

Ο Πίνακας 5.5 περιέχει τις βέλτιστες τιμές ακρίβειας και τον αριθμό διαστάσεων για κάθε μέθοδο και κάθε ταξινομητή στο σύνολο δεδομένων Breast Cancer. Στο συγκεκριμένο σύνολο δεδομένων παρουσιάζεται επίσης βελτίωση της απόδοσης σε πολλούς από τους ταξινομητές και σε διάφορες μεθόδους μείωσης διαστάσεων. Τονίζεται η αποτελεσματικότητα του ταξινομητή MLP ο οποίος παρουσιάζει τη βέλτιστη απόδοση σε όλες τις μεθόδους, με τον ταξινομητή SVM να έχει οριακά μικρότερη απόδοση σε μερικές από τις μεθόδους. Μικρές βελτιώσεις με την εφαρμογή των μεθόδων PCA, Kernel PCA και SVD παρουσιάζουν οι SVM, KNN και MLP, υποδεικνύοντας ότι οι συγκεκριμένες μέθοδοι είναι επωφελείς για αυτούς τους ταξι-

νομητές. Οι ταξινομητές Decision Tree και Naive Bayes εποφελούνται κυρίως από τη μέθοδο Factor Analysis, ενώ ο Random Forest διατηρεί σχεδόν ίδια απόδοση ανεξάρτητα από την εφαρμογή των μεθόδων μείωσης διαστάσεων. Μέσα από το γράφημα των Box plots παρατηρείται ότι την πιο υψηλή μέση απόδοση παρουσιάζουν οι μέθοδοι PCA, Kernel PCA και SVD, ενώ τη χειρότερη μέση απόδοση παρουσιάζουν οι μέθοδοι Factor Analysis και ICA.

Best Accuracy						
Method/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
PCA	0.90 (5)	**0.95** (29)	0.90 (8)	0.94 (27)	0.88 (23)	0.92 (10)
Kernel PCA	0.90 (5)	**0.95** (29)	0.90 (8)	0.94 (32)	0.88 (5)	0.92 (10)
SVD	0.90 (5)	**0.95** (29)	0.90 (10)	0.93 (33)	0.88 (23)	0.92 (10)
Factor Analysis	0.90 (7)	**0.95** (12)	0.91 (16)	0.94 (22)	0.89 (23)	0.92 (13)
LDA	**0.85** (1)	0.84 (1)	0.82 (1)	0.82 (1)	0.83 (1)	0.84 (1)
LLE	0.87 (33)	0.88 (27)	0.83 (31)	0.87 (30)	0.77 (13)	**0.90** (21)
Isomap	0.89 (6)	**0.92** (9)	0.89 (14)	**0.92** (17)	0.83 (26)	0.91 (10)
ICA	0.90 (5)	**0.95** (16)	0.88 (11)	0.94 (11)	0.91 (17)	0.92 (21)

Πίνακας 5.6: Πίνακας αποτελεσμάτων βέλτιστων τιμών Ionosphere Dataset



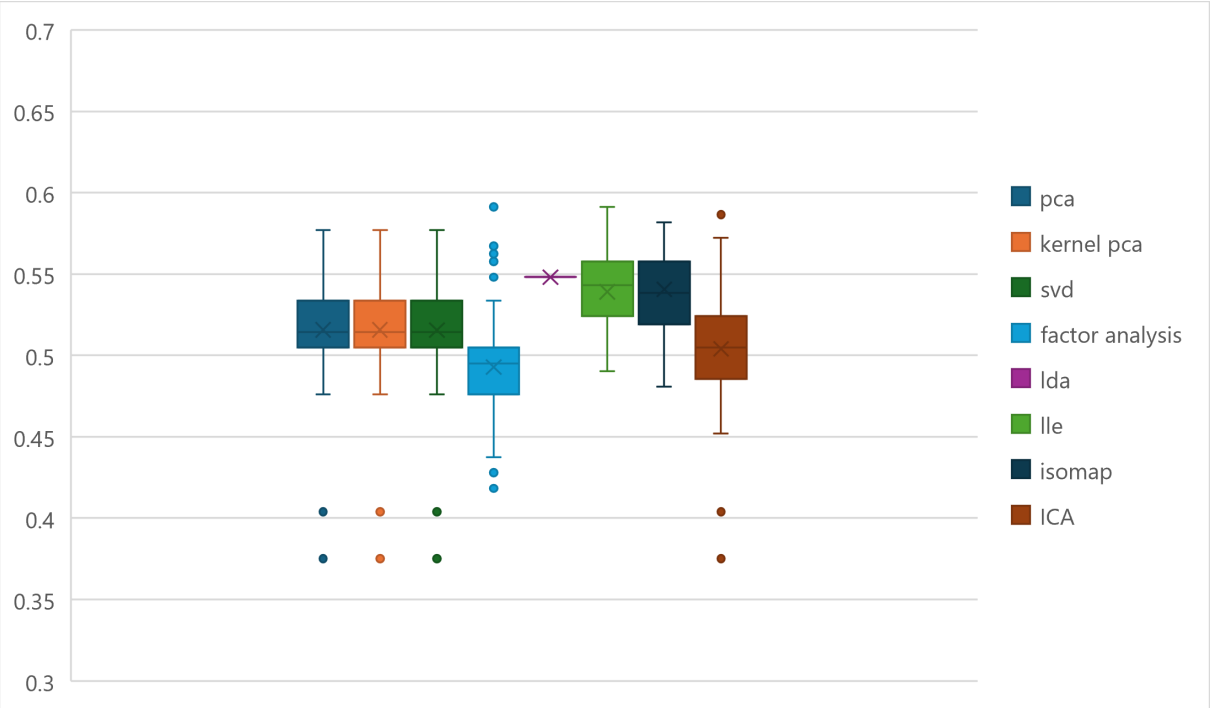
Σχήμα 5.4: Ionosphere Dataset Box Plot

Ο Πίνακας 5.6 περιέχει τις βέλτιστες τιμές ακρίβειας και τον αριθμό διαστάσεων

για κάθε μέθοδο και κάθε ταξινομητή στο σύνολο δεδομένων Ionosphere. Σημειώνεται βελτίωση στη βέλτιστη ακρίβεια μετά την εφαρμογή των μεθόδων σε όλους τους ταξινομητές με εξαίρεση τον ταξινομητή MLP ο οποίος σημειώνει ίδια ή ελάχιστα μικρότερη απόδοση σε σχέση με τα αρχικά αποτελέσματα. Αξιοσημείωτο είναι το φαινόμενο που παρατηρείται στην απόδοση των μεθόδων PCA, Kernel PCA και SVD, όπου σημειώνεται τεράστια μείωση διαστάσεων και παράλληλα βελτιώνεται και η τελική ακρίβεια των ταξινομητών. Επίσης μεγάλη σχετικά βελτίωση παρατηρείται και κατά την εφαρμογή της μεθόδου Factor Analysis από την οποία επωφελούνται αρκετά ο ταξινομητής Decision Tree και Naive Bayes. από το γράφημα των Box plots φαίνεται ότι η μέση ακρίβεια των μεθόδων είναι σχετικά χαμηλή και μόνο η μέθοδος LLE έχει μέση ακρίβεια μεγαλύτερη από 0.85. Χειρότερη ακρίβεια παρουσιάζει η μέθοδος ICA και ακολουθούν οι μέθοδοι Isomap και Factor Analysis.

Best Accuracy						
Method/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
PCA	0.58 (16)	0.59 (14)	0.59 (35)	**0.69** (27)	0.65 (3)	0.67 (15)
Kernel PCA	0.58 (16)	0.59 (14)	0.58 (41)	**0.67** (10)	0.65 (3)	**0.67** (15)
SVD	0.58 (16)	0.59 (14)	0.6 (56)	**0.66** (33)	0.65 (3)	**0.66** (15)
Factor Analysis	0.59 (5)	0.62 (8)	**0.70** (6)	0.69 (8)	0.63 (3)	0.69 (7)
LDA	0.55 (1)	0.54 (1)	**0.57** (1)	**0.57** (1)	0.55 (1)	0.55 (1)
LLE	0.57 (1)	0.58 (4)	**0.64** (44)	0.60 (49)	0.57 (37)	0.60 (59)
Isomap	0.58 (4)	0.63 (3)	**0.65** (44)	0.61 (9)	0.64 (3)	0.63 (4)
ICA	0.57 (17)	0.63 (9)	0.64 (12)	**0.68** (4)	0.65 (5)	0.66 (20)

Πίνακας 5.7: Πίνακας αποτελεσμάτων βέλτιστων τιμών Connectionist Bench Dataset

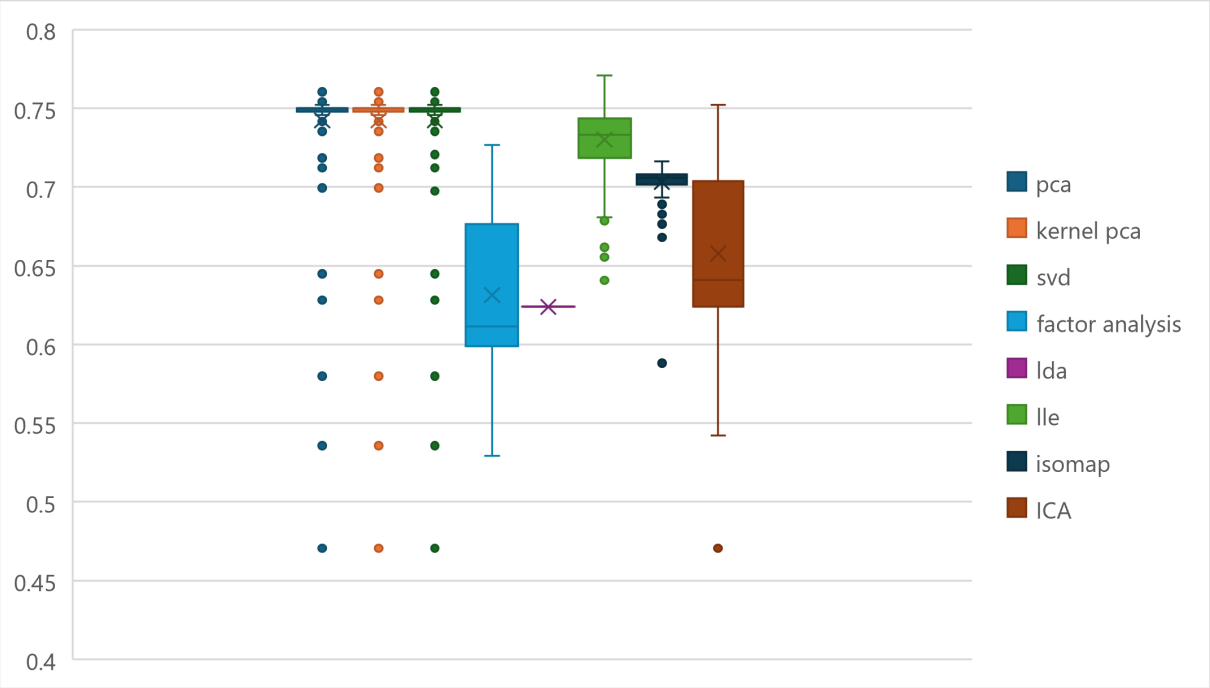


Σχήμα 5.5: Connectionist Bench Dataset Box Plot

Ο Πίνακας 5.7 περιέχει τις βέλτιστες τιμές ακρίβειας και τον αριθμό διαστάσεων για κάθε μέθοδο και κάθε ταξινομητή στο σύνολο δεδομένων Connectionist Bench. Το συγκεκριμένο σύνολο δεδομένων έχει τη χειρότερη απόδοση σε σχέση με τα υπόλοιπα σύνολα δεδομένων. Με την εφαρμογή των μεθόδων μείωσης διαστάσεων παρατηρείται μείωση της απόδοσης σε όλους τους ταξινομητές. Παρόλο που μειώνεται η ακρίβεια των ταξινομητών η μέθοδος η οποία έχει τα καλύτερα αποτελέσματα είναι η Factor Analysis. Αξιοσημείωτο είναι επίσης ότι και η μέση ακρίβεια των ταξινομητών όπως φαίνεται και στο σχεδιάγραμμα Box plot του Σχήματος 5.5 είναι αρκετά χαμηλή και καμία μέθοδος δεν ξεπερνάει τη μέση ακρίβεια 0.55.

Best Accuracy						
Method/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
PCA	0.76 (10)	0.79 (29)	0.70 (46)	0.77 (27)	0.78 (32)	**0.80** (8)
Kernel PCA	0.76 (10)	0.79 (29)	0.70 (42)	0.77 (28)	0.78 (32)	**0.80** (8)
SVD	0.76 (30)	0.79 (29)	0.70 (39)	0.77 (29)	0.77 (29)	**0.80** (8)
Factor Analysis	0.73 (11)	**0.76** (34)	0.72 (10)	0.74 (9)	0.72(36)	**0.76** (32)
LDA	0.62 (1)	**0.64** (1)	0.61 (1)	0.61 (1)	0.63 (1)	0.63 (1)
LLE	0.77 (158)	**0.78** (109)	0.77 (158)	0.77 (160)	0.77 (158)	0.77 (157)
Isomap	0.71 (108)	**0.80** (125)	0.75 (100)	0.76 (153)	0.71 (108)	0.79 (113)
ICA	0.75 (12)	0.79 (32)	0.75 (10)	**0.81** (10)	0.76 (9)	0.80 (9)

Πίνακας 5.8: Πίνακας αποτελεσμάτων βέλτιστων τιμών Musk Dataset



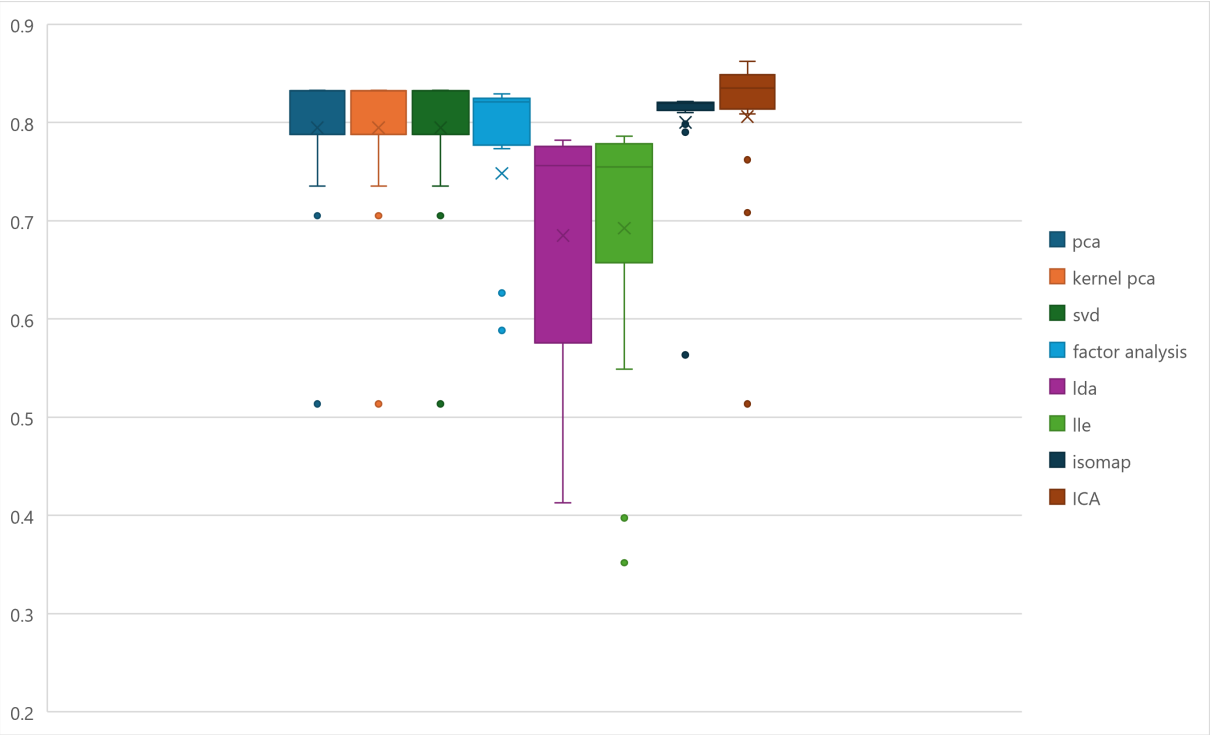
Σχήμα 5.6: Musk Dataset Box Plot

Ο Πίνακας 5.8 περιέχει τις βέλτιστες τιμές ακρίβειας και τον αριθμό διαστάσεων για κάθε μέθοδο και κάθε ταξινομητή στο σύνολο δεδομένων Musk. Σε αυτό το σύνολο δεδομένων σημειώνεται μια μείωση της απόδοσης των ταξινομητών η οποία όμως δεν είναι τόσο μεγάλη όσο στο σύνολο δεδομένων Connectionist Bench. Μοναδική εξαίρεση σε αυτό το γεγονός αποτελεί ο ταξινομητής Naive Bayes ο οποίος σημειώνει μια βελτίωση στην απόδοση του. Παράλληλα με τη μείωση της μέγιστης ακρίβειας σημειώνεται και τεράστια μείωση των διαστάσεων σε πολλές από τις μεθόδους, υποδεικνύοντας ότι μερικές φορές ίσως να αξίζει να θυσιαστεί ένα κομμάτι της ακρίβειας για την απλοποίηση πολύ περίπλοκων και πολυδιάστατων συνόλων

δεδομένων. Από τις μεθόδους μείωσης διαστάσεων την καλύτερη απόδοση έχουν οι μέθοδοι ICA και Factor Analysis. Στο διάγραμμα των Box plots παρατηρείται ότι η μέση ακρίβεια των μεθόδων PCA, Kernel PCA και SVD είναι αρκετά υψηλές συγκριτικά με τις υπόλοιπες μεθόδους. Μικρότερη μέση ακρίβεια αυτήν τη φορά σημειώνεται από τη μέθοδο Factor Analysis και αμέσως μετά με λίγο μεγαλύτερη μέση ακρίβεια έρχεται η LDA.

Best Accuracy						
Method/Classifier	KNN	SVM	Decision Tree	Random Forest	Naive Bayes	MLP
PCA	**0.88** (8)	0.83 (6)	0.77 (6)	0.83 (6)	0.86 (5)	0.78 (13)
Kernel PCA	0.83 (8)	0.83 (6)	0.76 (6)	0.83 (8)	**0.86** (5)	0.77 (13)
SVD	0.83 (8)	0.83 (6)	0.77 (6)	0.83 (6)	**0.85** (5)	0.78 (13)
Factor Analysis	0.83 (15)	0.83 (6)	0.68 (15)	0.74 (15)	**0.84** (6)	0.75 (15)
LDA	0.78 (5)	0.82 (5)	0.72 (5)	0.74 (5)	**0.85** (5)	0.78 (5)
ICA	**0.86** (6)	**0.86** (6)	0.78 (6)	0.84 (6)	0.83 (6)	0.76 (6)
LLE	0.78 (6)	0.78 (6)	0.74 (6)	0.79 (6)	0.56 (6)	**0.80 (6)**
Isomap	0.82 (7)	**0.84** (7)	0.79 (6)	0.83 (15)	**0.84** (7)	0.83 (3)

Πίνακας 5.9: Πίνακας αποτελεσμάτων βέλτιστων τιμών Dry bean Dataset



Σχήμα 5.7: Dry Bean Dataset Box Plot

Ο Πίνακας 5.9 περιέχει τις βέλτιστες τιμές ακρίβειας και τον αριθμό διαστάσεων για κάθε μέθοδο και κάθε ταξινομητή στο σύνολο δεδομένων Dry bean. Παρόμοια

πτωτική πορεία στην απόδοση των ταξινομητών παρατηρείται και σε αυτό το σύνολο δεδομένων. Η πτώση βέβαια, συγκριτικά με τα αποτελέσματα των μεθόδων χωρίς προεπεξεργασία είναι μικρότερη σε σχέση με τα προηγούμενα δύο σύνολα δεδομένων. Παράλληλα, αν και υπάρχει σχετική μείωση της απόδοσης των ταξινομητών υπάρχει και αισθήτη μείωση του αριθμού διαστάσεων σε σχεδόν στις μισές ή και λιγότερες. Οι μέθοδοι που έδωσαν τα καλύτερα αποτελέσματα παρόλη την πτωτική τάση της απόδοσης των ταξινομητών ήταν η μέθοδος Isomap και ICA. Στο γράφημα των Box plots η μέση απόδοση όλων των μεθόδων είναι αρκετά υψηλή και αγγίζει μερικές φορές ακόμα και τη βέλτιστη απόδοση. Αυτό υποδεικνύει ότι ανεξάρτητα τον αριθμό μείωσης των διαστάσεων το κύριο μέρος της πληροφορίας διατηρείται οδηγώντας τους ταξινομητές σε σχετικά υψηλές απόδοσης και παράλληλα απλοποιείται και το σύνολο δεδομένων.

5.6 Αποτελέσματα μεθόδων επιλογής χαρακτηριστικών

Best Accuracy (model used)				
Method/Classifier	**Boruta**	**Ensemble learning**	**Kendall's Tau**	**Spearman's**
KNN	**0.95** (LightGBM)	0.94 (LightGBM)	0.21	0.66
SVM	**0.97** (Gradient Boost)	**0.97** (LightGBM)	0.25	0.69
Decision Tree	**0.80** (XGBoost)	0.79 (XGBoost)	0.24	0.56
Random Forest	**0.94** (Random Forest)	**0.94** (LightGBM)	0.24	0.70
Naive Bayes	**0.86** (Gradient Boost)	0.85 (LightGBM)	0.22	0.31
MLP	**0.95** (Random Forest)	**0.95** (LightGBM)	0.24	0.68

Πίνακας 5.10: Πίνακας αποτελεσμάτων μεθόδων βέλτιστων τιμών επιλογής χαρακτηριστικών

Digits Dataset

Ο Πίνακας 5.10 δείχνει τα αποτελέσματα των ταξινομητών μετά την εφαρμογή των μεθόδων επιλογής χαρακτηριστικών για το σύνολο δεδομένων Digits. Στον πίνακα έχει γίνει επιλογή των βέλτιστων αποτελεσμάτων για τα διαφορετικά μοντέλα των μεθόδων Boruta και εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Αν και τα υποσύνολα δεδομένων απλοποιούν τα αρχικά δεδομένα αφαιρώντας τις περισσότερες φορές αρκετά χαρακτηριστικά, η ακρίβεια των ταξινομητών έχει πτωτική πορεία σε σχέση τα αποτελέσματα χωρίς προεπεξεργασία. Ειδικότερα, ο ταξινομητής SVM παρουσιάζει τα βέλτιστα αποτελέσματα κατά την εφαρμογή όλων των μεθόδων με μοναδική εξαίρεση κατά την εφαρμογή της μεθόδου Spearman's Rank Correlation Coefficient. Παράλληλα, η μέθοδος Boruta προσφέρει την καλύτερη επιλογή χαρακτηριστικών καθώς μετά την εφαρμογή της μεθόδου όλοι οι ταξινομητές έχουν τα βέλτιστα αποτελέσματα συγκριτικά με τις υπόλοιπες μεθόδους. Ακολουθεί με αρκετά κοντινά αποτελέσματα η μέθοδος εκμάθησης συνόλου κατά την οποία άλλοτε οι ταξινομητές έχουν την ίδια ακρίβεια με αυτήν που είχαν κατά την εφαρμογή της μεθόδου Boruta και άλλοτε ελάχιστα μικρότερη. Οι δύο συντελεστές συσχέτισης δεν κατάφεραν να καταγράψουν αποδοτικά τις σχέσεις μεταξύ των δεδομένων με αποτέλεσμα οι τελικοί ταξινομητές να έχουν κακή απόδοση. Πιθανότατα αυτό

οφείλεται στις ιδιότητες μεταξύ των χαρακτηριστικών που πιθανώς δεν είναι μονοτονικές.

Best Accuracy (model used)				
Method/Classifier	**Boruta**	**Ensemble learning**	**Kendall's Tau**	**Spearman's**
KNN	0.96 (XGBoost)	**0.97** (Random Forest)	0.95	0.94
SVM	**0.98** (Random Forest)	0.97 (CatBoost)	0.97	0.97
Decision Tree	0.92 (LightGBM)	**0.93** (Random Forest)	0.84	0.89
Random Forest	**0.98** (Random forest)	0.96 (Random Forest)	0.94	0.96
Naive Bayes	**0.96** (Random forest)	**0.96** (Random forest)	0.94	0.93
MLP	**0.97** (Random Forest)	**0.97** (LightGBM)	0.96	0.96

Πίνακας 5.11: Πίνακας αποτελεσμάτων μεθόδων βέλτιστων τιμών επιλογής χαρακτηριστικών

Wine Dataset

Ο Πίνακας 5.11 δείχνει τα αποτελέσματα των ταξινομητών μετά την εφαρμογή των μεθόδων επιλογής χαρακτηριστικών για το σύνολο δεδομένων Wine. Στον πίνακα έχει γίνει επιλογή των βέλτιστων αποτελεσμάτων για τα διαφορετικά μοντέλα των μεθόδων Boruta και εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Τα υποσύνολα δεδομένων που επιλέχθηκαν από τις μεθόδους διατηρούν ή και αυξάνουν πολλές φορές την ακρίβεια των ταξινομητών με μοναδική εξαίρεση τον ταξινομητή MLP όπου η ακρίβεια του είναι ελάχιστα μικρότερη από την αρχική. Ειδικότερα, ο ταξινομητής SVM παρουσιάζει τα βέλτιστα αποτελέσματα κατά την εφαρμογή όλων των μεθόδων. Οι μέθοδοι Boruta και εκμάθησης συνόλου είναι πολύ κοντά μεταξύ τους και άλλοτε υπερέχει η μια μέθοδος άλλοτε η άλλη. Οι δύο συντελεστές συσχέτισης λειτουργούν ιδιαίτερα αποτελεσματικά διατηρώντας τα κατάλληλα χαρακτηριστικά ώστε οι ταξινομητές να έχουν πολύ καλά αποτελέσματα, αρκετά κοντά σε αυτά των υπολοίπων δύο μεθόδων.

Best Accuracy (model used)				
Method/Classifier	**Boruta**	**Ensemble learning**	**Kendall's Tau**	**Spearman's**
KNN	**0.97** (Random Forest)	**0.97** (LightGBM)	**0.97**	0.96
SVM	0.97 (CatBoost)	**0.98** (Gradient Boost)	0.97	0.97
Decision Tree	0.92 (XGBoost)	0.92 (Adaboost)	**0.93**	0.91
Random Forest	**0.97** (LightGBM)	0.96 (Catboost)	**0.97**	0.96
Naive Bayes	**0.96** (LightGBM)	**0.96** (Gradient Boost)	0.93	0.93
MLP	0.97 (Catboost)	0.97 (Adaboost)	**0.98**	0.97

Πίνακας 5.12: Πίνακας αποτελεσμάτων μεθόδων βέλτιστων τιμών επιλογής χαρακτηριστικών Breast Cancer Dataset

Ο Πίνακας 5.12 δείχνει τα αποτελέσματα των ταξινομητών μετά την εφαρμογή των μεθόδων μείωσης διαστάσεων για το σύνολο δεδομένων Breast Cancer. Στον πίνακα έχει γίνει επιλογή των βέλτιστων αποτελεσμάτων για τα διαφορετικά μοντέλα των μεθόδων Boruta και εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Οι μέθοδοι έκαναν αποτελεσματική επιλογή χαρακτηριστικών, τα υποσύνολα που επέστρεψαν όχι μόνο απλοποίησαν το σύνολο δεδομένων αλλά βελτίωσαν και την ακρίβεια των ταξινομητών. Μοναδική εξαίρεση αποτελεί ο ταξινομητής Decision Tree ο οποίος είχε βέλτιστη ακρίβεια 0.93 ενώ κατά την εφαρμογή των μεθόδων χωρίς προεπεξεργασία είχε ακρίβεια 0.94. Οι μέθοδοι Boruta και εκμάθησης συνόλου λειτούργησαν αποτελεσματικά. Στο συγκεκριμένο σύνολο δεδομένων τόσο ο ταξινομητής SVM όσο και οι ταξινομητές MLP και KNN είχαν ικανοποιητικές αποδόσεις, οι οποίες ήταν πολύ κοντά μεταξύ τους. Επίσης, τόσο ο συντελεστής συσχέτισης κατάταξης του Kendall όσο και ο συντελεστής συσχέτισης κατάταξης του Spearman είχαν αρκετά καλά αποτελέσματα, με τον συντελεστή συσχέτισης κατάταξης του Kendall να είναι η μέθοδος που οδηγεί τους περισσότερους ταξινομητές

στη βέλτιστη απόδοση.

Best Accuracy (model used)				
Method/Classifier	**Boruta**	**Ensemble learning**	**Kendall's Tau**	**Spearman's**
KNN	**0.90** (XGBoost)	0.87 (Gradient Boost)	0.86	0.83
SVM	**0.95** (Random Forest)	**0.95** (LightGBM)	0.90	0.94
Decision Tree	0.86 (Random Forest)	0.87 (Random Forest)	0.88	**0.90**
Random Forest	0.93 (Random Forest)	**0.94** (Random Forest)	0.91	**0.94**
Naive Bayes	0.87 (Random Forest)	**0.90** (LightGBM)	0.88	0.89
MLP	0.89 (Random Forest)	**0.90** (Random Forest)	0.88	0.87

Πίνακας 5.13: Πίνακας αποτελεσμάτων μεθόδων βέλτιστων τιμών επιλογής χαρακτηριστικών Ionosphere Dataset

Ο Πίνακας 5.13 δείχνει τα αποτελέσματα των ταξινομητών μετά την εφαρμογή των μεθόδων μείωσης διαστάσεων για το σύνολο δεδομένων Ionosphere. Στον πίνακα έχει γίνει επιλογή των βέλτιστων αποτελεσμάτων για τα διαφορετικά μοντέλα των μεθόδων Boruta και εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Ο ταξινομητής SVM έχει τις καλύτερες τιμές ακρίβειας. Η μέθοδος εκμάθησης συνόλου για επιλογή χαρακτηριστικών προσφέρει τη μεγαλύτερη απόδοση συγκριτικά με τις υπόλοιπες μεθόδους και πολύ κοντά σε αυτήν είναι και η μέθοδος Boruta. Ιδιαίτερα αποτελεσματικοί είναι και οι δύο συντελεστές κατάταξης συσχέτισης με τον συντελεστή κατάταξης συσχέτισης του Spearman να παρουσιάζει τη μεγαλύτερη ακρίβεια στους ταξινομητές Decision Tree και Random Forest και τον συντελεστή κατάταξης συσχέτισης Kendall να οδηγεί τους ταξινομητές σε πολύ καλές αποδόσεις.

Best Accuracy (model used)				
Method/Classifier	**Boruta**	**Ensemble learning**	**Kendall's Tau**	**Spearman's**
KNN	0.55 (Adaboost)	0.55 (AdaBoost)	**0.63**	0.60
SVM	**0.60** (Adaboost)	**0.60** (AdaBoost)	**0.60**	**0.60**
Decision Tree	0.62 (CatBoost)	**0.64** (AdaBoost)	0.61	0.60
Random Forest	0.62 (LightGBM)	**0.66** (LightGBM)	0.62	0.61
Naive Bayes	0.57 (LightGBM)	0.57 (LightGBM)	**0.58**	0.57
MLP	**0.64** (Adaboost)	**0.64** (AdaBoost)	0.63	0.60

Πίνακας 5.14: Πίνακας αποτελεσμάτων μεθόδων βέλτιστων τιμών επιλογής χαρακτηριστικών Connectionist Bench Dataset

Ο Πίνακας 5.14 δείχνει τα αποτελέσματα των ταξινομητών μετά την εφαρμογή των μεθόδων μείωσης διαστάσεων για το σύνολο δεδομένων Connectionist Bench. Στον πίνακα έχει γίνει επιλογή των βέλτιστων αποτελεσμάτων για τα διαφορετικά μοντέλα των μεθόδων Boruta και εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Τα ποσοστά ακρίβειας των ταξινομητών έχουν πτωτική πορεία σε σχέση με τις μετρήσεις του πίνακα αποτελεσμάτων χωρίς προεπεξεργασία. Μοναδική εξαίρεση αποτελεί ο Random Forest ο οποίος παρουσίασε μια μικρή αύξηση στην ακρίβεια του. Η μέθοδος που οδήγησε τους ταξινομητές στην καλύτερη απόδοση ήταν η μέθοδος εκμάθησης συνόλου και αρκετά κοντά σε αυτήν ήταν και η μέθοδος Boruta. Οι δύο συντελεστές συσχέτισης κατάταξης επιστρέφουν επίσης αρκετά καλά αποτελέσματα συγκριτικά με τις υπόλοιπες μεθόδους, επιτυγχάνοντας και μέσω της μεθόδου του Kendall τη βέλτιστη απόδοση στα μοντέλα KNN, SVM και Naive Bayes.

Best Accuracy (model used)				
Method/Classifier	**Boruta**	**Ensemble learning**	**Kendall's Tau**	**Spearman's**
KNN	**0.86** (LightGBM)	0.84 (LightGBM)	0.73	0.79
SVM	**0.85** (LightGBM)	0.79 (XGBoost)	0.80	0.82
Decision Tree	0.57 (CatBoost)	**0.58** (LightGBM)	0.42	0.56
Random Forest	**0.63** (LightGBM)	0.60 (LightGBM)	0.39	0.52
Naive Bayes	**0.84** (LightGBM)	0.79 (Random forest)	0.72	0.77
MLP	**0.80** (XGBoost)	0.76 (LightGBM)	0.78	0.71

Πίνακας 5.15: Πίνακας αποτελεσμάτων μεθόδων βέλτιστων τιμών επιλογής χαρακτηριστικών

Dry bean Dataset

Ο Πίνακας 5.15 δείχνει τα αποτελέσματα των ταξινομητών μετά την εφαρμογή των μεθόδων μείωσης διαστάσεων για το σύνολο δεδομένων Dry Bean. Στον πίνακα έχει γίνει επιλογή των βέλτιστων αποτελεσμάτων για τα διαφορετικά μοντέλα των μεθόδων Boruta και εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Οι μέθοδοι επιλογής χαρακτηριστικών ενώ απλοποίησαν την πολυπλοκότητα των συνόλων δεδομένων δεν συντέλεσαν στην αύξηση της ακρίβειας των ταξινομητών, αντίθετα τα αποτελέσματα ακρίβειας είχαν πτωτική τάση. Τα αποτελέσματα δείχνουν ότι η μέθοδος Boruta οδηγεί τους ταξινομητές στα καλύτερα αποτελέσματα σε πέντε από τους έξι ταξινομητές ενώ πολύ κοντά σε ποσοστά ακρίβειας ήταν και η μέθοδος εκμάθησης συνόλου. Οι ταξινομητές που είχαν τα καλύτερα ποσοστά ακρίβειας ήταν οι KNN και SVM, ο KNN αξιοποίησε καλύτερα τα δεδομένα των μεθόδων Boruta και της μεθόδου εκμάθησης συνόλου, ενώ ο SVM πέτυχε τη βέλτιστη απόδοση μέσω των κατά την εφαρμογή των μεθόδων Kendall και Spearman. Σε ένα γενικότερο πλαίσιο οι συντελεστές κατάταξης συσχέτισης αν και είχαν λίγο χαμηλότερη απόδοση ήταν αρκετά κοντά στις δύο πρώτες μεθόδους.

Best Accuracy (model used)				
Method/Classifier	**Boruta**	**Ensemble learning**	**Kendall's Tau**	**Spearman's**
KNN	0.70 (Random Forest)	**0.74** (AdaBoost)	0.64	0.72
SVM	0.72 (CatBoost)	**0.75** (AdaBoost)	0.69	0.74
Decision Tree	0.62 (AdaBoost)	**0.68** (XGBoost)	0.65	0.66
Random Forest	0.70 (Gradient Boost)	0.70 (Gradient Boost)	0.62	**0.75**
Naive Bayes	0.60 (LightGBM)	**0.62** (AdaBoost)	0.60	0.51
MLP	0.76 (Random Forest)	**0.78** (LightGBM)	0.60	0.73

Πίνακας 5.16: Πίνακας αποτελεσμάτων μεθόδων βέλτιστων τιμών επιλογής χαρακτηριστικών Musk Dataset

Ο Πίνακας 5.16 παρουσιάζει τα αποτελέσματα των ταξινομητών μετά την εφαρμογή των μεθόδων μείωσης διαστάσεων για το σύνολο δεδομένων Musk. Στον πίνακα έχει γίνει επιλογή των βέλτιστων αποτελεσμάτων για τα διαφορετικά μοντέλα των μεθόδων Boruta και εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Οι μέθοδοι επιλογής χαρακτηριστικών ενώ απλοποίησαν την πολυπλοκότητα των συνόλων δεδομένων δεν συντέλεσαν στην αύξηση της ακρίβειας των ταξινομητών, αντίθετα τα αποτελέσματα ακρίβειας είχαν πτωτική τάση. Τα αποτελέσματα δείχνουν ότι η μέθοδος Boruta οδηγεί τους ταξινομητές στα καλύτερα αποτελέσματα σε πέντε από τους έξι ταξινομητές ενώ πολύ κοντά σε ποσοστά ακρίβειας ήταν και η μέθοδος εκμάθησης συνόλου. Ο ταξινομητής με τη βέλτιστη απόδοση ήταν ο SVM και ο MLP. Ο SVM είχε καλύτερη απόδοση κατά την εφαρμογή των συντελεστών κατάταξης συσχέτισης του Kendall και του Spearman, ενώ ο MLP κατά την εφαρμογή των μεθόδων Boruta και εκμάθησης συνόλου. Οι συντελεστές κατάταξης συσχέτισης είχαν λίγο χαμηλότερη απόδοση συγκριτικά με τις άλλες δύο μεθόδους κατά μέσω όρο μεταξύ των ταξινομητών, αλλά χωρίς να παρατηρείται κάποια μεγάλη απόκλιση.

Κεφάλαιο 6

Συμπεράσματα

Στη εργασία πραγματοποιήθηκε μια πλήρης συγκριτική ανάλυση μεταξύ διαφόρων μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Οι μέθοδοι εφαρμόστηκαν σε επτά σύνολα δεδομένων με διαφορετικό αριθμό χαρακτηριστικών. Ειδικότερα, ο αριθμός των χαρακτηριστικών κάθε συνόλου κυμαινόταν από 13 - 168 χαρακτηριστικά. Επιπλέον, η σύγκριση των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών έγινε υπό το πρίσμα της επίδρασης των συγκεκριμένων μεθόδων στην αύξηση, διατήρηση ή μείωση της αποτελεσματικότητας των μοντέλων πρόβλεψης. Για τον παραπάνω λόγο χρησιμοποιήθηκαν έξι διαφορετικά είδη ταξινομητών. Οι ταξινομητές που χρησιμοποιήθηκαν ήταν ο k-Nearest neighbors (KNN), ο Support Vector Machine (SVM), ο Decision Tree, ο Random Forest, ο Naive Bayes και ο Multilayer Perceptron (MLP). Οι έξι ταξινομητές εφαρμόστηκαν και στα επτά αρχικά σύνολα δεδομένων χωρίς να χρησιμοποιηθεί καμία από τις μεθόδους μείωσης διαστάσεων και επιλογής χαρακτηριστικών ώστε να γίνει η καταγραφή της απόδοσης τους.

Στη συνέχεια πραγματοποιήθηκαν δύο πειράματα. Το πρώτο πείραμα αφορούσε τη διευρεύνηση των μεθόδων μείωσης διαστάσεων, όπου χρησιμοποιήθηκαν οι μέθοδοι PCA, SVD, LDA, Kernel PCA, Isomap, LLE, Factor Analysis, ICA. Ειδικότερα, για κάθε σύνολο δεδομένων εφαρμόστηκαν όλες οι μέθοδοι μείωσης διαστάσεων και τα μετασχηματισμένα δεδομένα τροφοδοτήθηκαν στους έξι ταξινομητές από όπου καταγράφτηκε η απόδοσή τους. Αντί για κάποιο συγκεκριμένο αριθμό διαστάσεων χρησιμοποιήθηκε το εύρος των χαρακτηριστικών μείον ένα, με αποτέλεσμα για κάθε σύνολο δεδομένων να υπάρχει καταγεγραμμένη η απόδοση για κάθε διάσταση μέχρι τον αρχικό αριθμό διαστάσεων μείον ένα για όλες τις μεθόδους και όλους τους

ταξινομητές σε κάθε σύνολο δεδομένων.

Το δεύτερο πείραμα αφορούσε την αξιολόγηση των μεθόδων επιλογής χαρακτηριστικών. Στο συγκεκριμένο πείραμα εφαρμόστηκαν οι μέθοδοι Boruta, εκμάθησης συνόλου, Kendall's Rank Correlation Coefficient και Spearman's Rank Correlation Coefficient. Οι μέθοδοι Boruta και εκμάθησης συνόλου έχουν την ιδιαιτερότητα ότι χρησιμοποιούν μοντέλα εκμάθησης συνόλου για την επιλογή των σημαντικών χαρακτηριστικών. Στο συγκεκριμένο πείραμα χρησιμοποιήθηκαν έξι διαφορετικά μοντέλα στις δύο μεθόδους, τα μοντέλα ήταν ο Random Forest, Gradient Boosting, AdaBoost, XGBoost, LightGBM και CatBoost. Τα συγκεκριμένα μοντέλα διαφέρουν μεταξύ τους στον τρόπο με τον οποίο αξιολογούν τα χαρακτηριστικά, επομένως χρησιμοποιώντας τα στις δύο μεθόδους η επιλογή των σημαντικών χαρακτηριστικών θα διαφέρει ανάλογα το μοντέλο. Στο πείραμα εφαρμόστηκαν όλες οι μέθοδοι επιλογής χαρακτηριστικών στα επτά σύνολα δεδομένων, έγινε η επιλογή των σημαντικών χαρακτηριστικών από κάθε μέθοδο και τα αποτελέσματα τροφοδοτήθηκαν στους έξι ταξινομητές από όπου έγινε η εκπαίδευση των ταξινομητών και καταγράφτηκε η απόδοση τους.

Παράλληλα με τα δύο πειράματα έγινε και μια ανάλυση της λειτουργίας των διαφόρων μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών ως προς την εύρεση ενός μηχανισμού για τον αυτόματο εντοπισμό του βέλτιστου αριθμού διαστάσεων. Ειδικότερα, στις κλασσικές υλοποιήσεις των μεθόδων προστέθηκαν συναρτήσεις οι οποίες αξιοποιούν τα ενδιάμεσα βήματα των μεθόδων με σκοπό την αυτόματη εύρεση του βέλτιστου αριθμού διαστάσεων. Οι συγκεκριμένες τεχνικές αναπτύχθηκαν στις μεθόδους PCA, LDA, SVD, Kernel PCA, Isomap, LLE, Factor Analysis και εκμάθησης συνόλου. Αξιοποιήθηκαν χαρακτηριστικά όπως η διακύμανση και οι ιδιοτιμές, που υπολογίζονται εσωτερικά στις μεθόδους καθώς και ο βαθμός σημαντικότητας που προσδίδουν άλλες μέθοδοι στα χαρακτηριστικά και συνδυαστικά, κατασκευάζοντας τις γραφικές τους παραστάσεις, αξιοποιώντας μεθόδους εύρεσης του σημείου όπου το κέρδος μεταξύ της προσθήκης νέων χαρακτηριστικών ή διαστάσεων είναι δυσανάλογο της πληροφορίας που προσφέρουν αναπτύχθηκαν συναρτήσεις που εντοπίζουν και επιστρέφουν αυτό το σημείο. Για την περαιτέρω ανάλυση και δοκιμή της απόδοσης αυτών των τεχνικών πραγματοποιήθηκε ένα τρίτο πείραμα. Στο τρίτο πείραμα χρησιμοποιήθηκαν τα επτά σύ-

νολα δεδομένων και εφαρμόστηκαν οι μέθοδοι αυτόματης εύρεσης των βέλτιστων διαστάσεων ή χαρακτηριστικών. Τα αποτελέσματα των μεθόδων τροφοδοτήθηκαν στους έξι ταξινομητές και καταγράφτηκε η απόδοση τους. Ο σκοπός του τρίτου πειράματος ήταν η αξιολόγηση και κατανόηση κατά πόσο οι συγκεκριμένες τεχνικές επιστρέφουν τα βέλτιστα αποτελέσματα ή πόσο κοντά σε αυτά βρίσκονται.

Από τα τρία πειράματα τα αποτελέσματα που εξήχθησαν δείχνουν ότι οι μέθοδοι μείωσης διαστάσεων είχαν θετικά αποτελέσματα κατά κύριο λόγο στα περισσότερα σύνολα δεδομένων. Ειδικότερα, παρατηρήθηκε ότι η απόδοση των περισσότερων ταξινομητών αυξήθηκε ή παρέμεινε ίδια. Σε μερικές περιπτώσεις, κατά κύριο λόγο στο σύνολο δεδομένων Dry Bean, Connectionist Bench και Musk δεν παρατηρήθηκε βελτίωση των αποτελεσμάτων, αλλά μία σχετική μείωση της απόδοσης των ταξινομητών. Τα αποτελέσματα αυτά κατά κύριο λόγο οφείλονται στη δομή και τα στοιχεία των συνόλων δεδομένων. Στα πρώτα τέσσερα σύνολα δεδομένων είναι πιθανό να υπήρχε αρκετή επαναλαμβανόμενη πληροφορία ή κάποιο ποσοστό θορύβου το οποίο εξαλείφθηκε μέσω των μεθόδων μείωσης διαστάσεων οδηγώντας έτσι τους ταξινομητές σε καλύτερα αποτελέσματα. Αντίθετα, στα τελευταία τρία σύνολα δεδομένων είναι πιθανόν να ισχύουν οι αντίστροφες συνθήκες. Ειδικότερα, από τα αποτελέσματα φαίνεται ότι τα τρία σύνολα δεδομένων δεν παρουσίαζαν κάποιο μεγάλο ποσοστό θορύβου ή κάποια επαναλαμβανόμενη και κατ' επέκταση περιττή πληροφορία. Η μείωση των διαστάσεων σε τέτοιες περιπτώσεις είναι πιθανό να οδηγήσει σε απώλεια χρήσιμης πληροφορίας την οποία θα μπορούσαν να αξιοποιήσουν οι ταξινομητές και κατ' επέκταση σε μείωση της απόδοσης τους.

Στο δεύτερο πείραμα τα αποτελέσματα δεν ήταν εντελώς ξεκάθαρα. Κατά κύριο λόγο οι μέθοδοι επιλογής χαρακτηριστικών διατήρησαν ίδια ή μείωσαν σε μικρό βαθμό τη μέγιστη απόδοση των ταξινομητών. Σε συγκεκριμένες περιπτώσεις παρατηρήθηκε αύξηση της απόδοσης κάποιων ταξινομητών, αυτό κυρίως παρουσιάστηκε στον ταξινομητή Naive Bayes. Ο συγκεκριμένος ταξινομητής δέχεται σαν παραδοχή ότι τα χαρακτηριστικά είναι ανεξάρτητα μεταξύ τους. Επομένως, αφαιρώντας τα λιγότερο σημαντικά χαρακτηριστικά, πιθανώς να είχε ως αποτέλεσμα τη διατήρηση των περισσότερο ανεξάρτητων χαρακτηριστικών γεγονός που με τη σειρά του οδήγησε στη βελτίωση της απόδοσης του ταξινομητή. Τέλος, το ίδιο φαινόμενο με τα τρία τελευταία σύνολα δεδομένων Dry Bean Dataset, Connectionist Bench Dataset,

Musk Dataset, παρατηρήθηκε και σε αυτό το πείραμα όπου φαίνεται ότι η μείωση των χαρακτηριστικών δεν βοήθησε τους ταξινομητές. Το γεγονός αυτό ενισχύει την ιδέα ότι τα τρία σύνολα δεδομένων δεν έχουν περιττές πληροφορίες ή τουλάχιστον ο βαθμός των περιττών πληροφοριών είναι πολύ μικρός και ότι όλα τα χαρακτηριστικά συνεισφέρουν στην αποτελεσματικότητα των ταξινομητών.

Στο τρίτο πείραμα όπου χρησιμοποιήθηκαν οι τεχνικές για την εύρεση των βέλτιστων διαστάσεων τα αποτελέσματα ήταν θετικά. Γενικότερα τα αποτελέσματα των τεχνικών δεν οδήγησαν στην εύρεση της διάστασης η οποία οδηγεί τους ταξινομητές στα βέλτιστα αποτελέσματα σε τακτική βάση. Παρόλα αυτά οι διαστάσεις που επιλέχθηκαν οδήγησαν τους ταξινομητές σε ικανοποιητικά αποτελέσματα τα οποία είναι αρκετά κοντά στα βέλτιστα. Επιπλέον, σε σχέση με τις αποδόσεις των ταξινομητών στα αρχικά δεδομένα, οι τεχνικές εύρεσης των βέλτιστων διαστάσεων οδήγησαν σε αποτελέσματα που άλλοτε είναι πολύ κοντά στα αρχικά και άλλοτε τα ξεπερνάνε. Η εύρεση της βέλτιστης διάστασης πολλές φορές είναι ένα πολυπαραγοντικό και υποκειμενικό ζήτημα. Ειδικότερα, η βέλτιστη διάσταση κρίνεται σε μεγάλο βαθμό από τον ίδιο τον αναλυτή του μοντέλου, τους πόρους που διαθέτει και τη θέληση να θυσιάσει ένα μικρό ποσοστό απόδοσης ώστε να μειώσει σημαντικά την πολυπλοκότητα χώρου και χρόνου του μοντέλου. από τα αποτελέσματα φαίνεται ότι οι συγκεκριμένες τεχνικές πράγματι οδηγούν σε ένα χώρο μειωμένων διαστάσεων ο οποίος όντως μειώνει την πολυπλοκότητα του μοντέλου σε μεγάλο βαθμό αντισταθμίζοντας και τη διατήρηση της πληροφορίας, πράγμα που φαίνεται από την απόδοση των ταξινομητών.

Εμβαθύνοντας περισσότερο στις μεθόδους μείωσης διαστάσεων και αναλύοντας τόσο συγκριτικά όσο και συνολικά τα αποτελέσματα της κάθε μεθόδου τα συμπεράσματα που προκύπτουν είναι ότι η μέθοδος PCA οδηγεί τους ταξινομητές κοντά στη μέγιστη απόδοση με μια απόκλιση της τάξης του 10% με περίπου το 1/5 των διαστάσεων του αρχικού συνόλου. Εξαίρεση αποτελούν ως ένα βαθμό τα τρία τελευταία σύνολα δεδομένων όπου εκεί η απόκλιση ήταν λίγο μεγαλύτερη και οι διαστάσεις περισσότερες. Σε ένα γενικότερο πλαίσιο η μέθοδος ήταν αποτελεσματική επιτυγχάνοντας τον απώτερο σκοπό μείωσης των διαστάσεων διατηρώντας σε μεγάλο βαθμό την πληροφορία του συνόλου δεδομένων και τις αποδόσεις των ταξινομητών.

Η μέθοδος Kernel PCA είχε σχεδόν όμοια αποτελέσματα με τη μέθοδο PCA. Σε ελάχιστα σημεία υπήρχαν μικρές διαφορές μεταξύ των αποτελεσμάτων των δύο μεθόδων. Το κύριο πλεονέκτημα της μεθόδου Kernel PCA έναντι της κλασσικής PCA που παρατηρήθηκε ήταν ότι ενώ η απόδοση των ταξινομητών ήταν σχεδόν όμοια πολλές φορές η Kernel PCA οδηγούσε τους ταξινομητές σε αυτήν την απόδοση με πολύ μικρότερο αριθμό διαστάσεων. Το γεγονός αυτό πιθανότατα προκύπτει λόγω της δυνατότητας της μεθόδου Kernel PCA να καταγράφει μη γραμμικές σχέσεις μεταξύ των δεδομένων.

Παρόμοια αποτελέσματα έχει και η μέθοδος SVD. Αυτό ως ένα βαθμό είναι αναμενόμενο μιας και η μέθοδος SVD ακολουθεί σε μεγάλο βαθμό τη λειτουργία της μεθόδου PCA με κύρια διαφορά ότι χρησιμοποιεί αυτούσιο τον πίνακα των δεδομένων σε αντίθεση με την PCA που χρησιμοποιεί τον πίνακα συνδιακύμανσης. Τόσο ο αριθμός βέλτιστων διαστάσεων όσο και η απόδοση των ταξινομητών μετά την εφαρμογή της μεθόδου είναι πολύ κοντά και οριακά κοινά με αυτά της PCA.

Η μέθοδος Factor Analysis από την άλλη πλευρά φαίνεται ότι η απόδοση της εξαρτάται κατά κύριο λόγο από το σύνολο των δεδομένων. Τόσο στο Digits Dataset όσο και στο Wine Dataset φαίνεται να έχει υψηλές αποδόσεις με αρκετά μικρό αριθμό διαστάσεων και αποτελέσματα κοντά σε αυτά των προηγούμενων μεθόδων. Στα τρία επόμενα σύνολα δεδομένων Breast Cancer Dataset, Ionosphere Dataset και κατά κύριο λόγο στο Connectionist Bench Dataset φαίνεται να οδηγεί τους ταξινομητές σε ελαφρώς καλύτερα αποτελέσματα σε σχέση με τις υπόλοιπες μεθόδους. Αντίθετα, στα δύο τελευταία σύνολα δεδομένων Dry Bean Dataset και Musk Dataset παρατηρείται μια μικρή πτώση στα αποτελέσματα της μεθόδου. Κατά κύριο λόγο, οι συγκεκριμένες παρατηρήσεις πιθανώς προέρχονται από τη θεωρητική προσέγγιση της μεθόδου, η οποία ορίζει ότι η μέθοδος είναι αποτελεσματική όταν υπάρχει πολύσυγγραμικότητα και όταν υπάρχει κάποια λανθάνουσα υποκείμενη δομή που επηρεάζει τα στοιχεία του συνόλου δεδομένων.

Η γραμμική μέθοδος μείωσης διαστάσεων LDA φαίνεται να είναι ιδιαίτερα αποτελεσματική καθώς λειτουργεί ικανοποιητικά και για τα επτά σύνολα δεδομένων. Δεν έχει τα βέλτιστα αποτελέσματα συγκριτικά με τις υπόλοιπες μεθόδους, αλλά στα περισσότερα σύνολα δεδομένων οδηγεί τους ταξινομητές σε ικανοποιητικά αποτελέσματα. Το κύριο προτέρημα της μεθόδου είναι ότι μειώνει σε πολύ μεγάλο

βαθμό τις διαστάσεις και παρόλα αυτά διατηρεί την απόδοση των ταξινομητών. Λόγω ότι η μέθοδος ψάχνει να βρει μια αναπαράσταση των δεδομένων κατά την οποία προκύπτει η μέγιστη διαχωριστικότητα μεταξύ των στοιχείων του χαρακτηριστικού κλάσης, οι διαστάσεις που μπορεί να μειώσει το σύνολο δεδομένων ειδικά σε προβλήματα ταξινόμησης είναι αρκετά μικρές.

Στο κομμάτι των μη γραμμικών μεθόδων η μέθοδος LLE είχε εξίσου καλά αποτελέσματα συγκριτικά με τις υπόλοιπες μεθόδους. Στα πρώτα σύνολα δεδομένων λειτούργησε αποτελεσματικά. Όμως, στα 2 ενδιάμεσα σύνολα δεδομένων το Ionoshere Dataset και το Connectionist Bench Dataset η απόδοση της άρχισε να πέφτει ελαφρώς. Μια αδυναμία της μεθόδου που παρατηρείται από τα αποτελέσματα είναι ότι όσο αυξάνεται ο αριθμός των χαρακτηριστικών των δεδομένων τόσο αυξάνεται και ο αριθμός των διαστάσεων που απαιτείται από την LLE για τη βέλτιστη λειτουργία των ταξινομητών. Αυτό το συμπέρασμα δείχνει ότι η προσπάθεια της μεθόδου να καταγράψει την τοπική δομή των δεδομένων απαιτεί αρκετές διαστάσεις για να επιτευχθεί αποτελεσματικά.

Η Isomap από την άλλη πλευρά φαίνεται να λειτουργεί αποτελεσματικά στα πρώτα 3 σύνολα δεδομένων Digits Dataset, Wine Dataset, Breast Cancer Dataset βέβαια και αυτήν η μέθοδος χρειάζεται μεγαλύτερο αριθμό διαστάσεων για να επιτύχει τα βέλτιστα αποτελέσματα συγκριτικά με τις υπόλοιπες μεθόδους όπως την PCA και Kernel PCA. Στα επόμενα 2 σύνολα δεδομένων το Ionosphere Dataset και το Connectionist Bench Dataset η μέθοδος λειτουργεί ιδιαίτερα αποτελεσματικά αγγίζοντας πολλές φορές τις βέλτιστες τιμές μεταξύ όλων των μεθόδων. Στο Musk Dataset η μέθοδος είχε ελαφρώς χαμηλότερη απόδοση συγκριτικά με τις υπόλοιπες μεθόδους και χρειαζόταν επίσης μεγάλος αριθμός διαστάσεων για την επίτευξη της. Το γεγονός αυτό ενισχύει το προηγούμενο συμπέρασμα ότι πολλές φορές μέθοδοι που προσπαθούν να αναπαραστήσουν τη δομή των δεδομένων σε ένα χώρο μειωμένων διαστάσεων χρειάζονται μεγάλο αριθμό διαστάσεων για να το επιτύχουν αυτό αποτελεσματικά.

Τέλος, η μέθοδος μείωσης διαστάσεων ICA φαίνεται να είναι μια από τις σταθερότερες μεθόδους. Λειτουργεί αποτελεσματικά σε όλα τα σύνολα δεδομένων επιτυγχάνοντας πολλές φορές και τη μέγιστη απόδοση συγκριτικά με τις υπόλοιπες μεθόδους. Ένα από τα κύρια κριτήρια για την αποδοτική λειτουργία της μεθόδου

είναι η επιλογή του κατάλληλου αριθμού διαστάσεων. από τα γραφήματα Box plots φαίνεται ότι η μέση ακρίβεια των ταξινομητών μετά την εφαρμογή της μεθόδου είναι αρκετά χαμηλή, όμως παρόλα αυτά σε συγκεκριμένους αριθμούς επιλογής διαστάσεων οδηγεί τους ταξινομητές στα υψηλότερα ποσοστά ακρίβειας. Επιπλέον επιτυγχάνει αυτές τις υψηλές αποδόσεις με σχετικά χαμηλό αριθμό διαστάσεων, γεγονός που την καθιστά ακόμα πιο αποτελεσματική. Ο στόχος της μεθόδου είναι η εύρεση ανεξάρτητων σημάτων εισόδου, το οποίο φαίνεται ότι οδηγεί τους ταξινομητές γρήγορα και εύκολα σε καλές αποδόσεις. Επιπλέον ένα χαρακτηριστικό της ICA είναι ότι προσπαθεί να αφαιρέσει τον θόρυβο, κάτι το οποίο πιθανότατα συντελεί επίσης στην αυξημένη αποτελεσματικότητα των ταξινομητών.

Οι μέθοδοι επιλογής χαρακτηριστικών από την άλλη πλευρά προσπαθούν να επιλέξουν τα βέλτιστα χαρακτηριστικά για την καλύτερη εκπαίδευση των ταξινομητών. Ο αλγόριθμος Boruta προσπαθεί να επιλέξει τα πιο σημαντικά χαρακτηριστικά χρησιμοποιώντας όσο το δυνατόν πιο αντικειμενικά κριτήρια. από τα βέλτιστα αποτελέσματα των μεθόδων επιλογής χαρακτηριστικών φαίνεται ότι έχει κατά κύριο λόγο τα καλύτερα αποτελέσματα σε όλα τα σύνολα δεδομένων με ελάχιστες εξαιρέσεις. Η αποτελεσματικότητα της μεθόδου πιθανώς οφείλεται στο γεγονός ότι επαναληπτικά ελέγχει και ταξινομεί τα χαρακτηριστικά με βάση τυχαία χαρακτηριστικά. Έτσι, μετά από πολλές επαναλήψεις επιλέγει τα χαρακτηριστικά τα οποία πολλές φορές χαρακτηρίστηκαν ως σημαντικά, μειώνοντας έτσι σημαντικά την πιθανότητα κάποιου χαρακτηριστικού να επιλεχθεί εσφαλμένα ως σημαντικό. Σχετικά με το βέλτιστο μοντέλο κατά τη λειτουργία της μεθόδου Boruta, όλα τα μοντέλα αν και υπήρχε μια διαφοροποίηση στα χαρακτηριστικά που επέλεγε κάθε μοντέλο, κατά κύριο λόγο οδήγησαν τους τελικούς ταξινομητές σε παρόμοια αποτελέσματα. Ξεχωρίζουν ως ένα βαθμό τα μοντέλα LightGBM και XGBoost τα οποία φαίνεται να είχαν ελαφρώς καλυτερα αποτέλεσματα κατά μέσω όρο μεταξύ των συνόλων δεδομένων.

Η μέθοδος επιλογής χαρακτηριστικών εκμάθησης συνόλου ήταν αρκετά πιο γρήγορη από τη μέθοδο Boruta και οι επιλογές των σημαντικών στοιχείων δεν γινόταν μετά από κάποια επαναληπτική διαδικασία αλλά από τη μοναδική επιλογή της εκάστοτε μοντέλου. Παρόλα αυτά η μέθοδος φαίνεται να οδηγεί τους ταξινομητές σε ικανοποιητικά αποτελέσματα αναλογιζόμενοι κατά κύριο λόγο ότι οι αποδόσεις συγκριτικά με τις αποδόσεις της μεθόδου Boruta ειναι σχετικά κοντά και ότι κατά

την εφαρμογή της μεθόδου εκμάθησης συνόλου επιλεγόταν τις περισσότερες φορές λιγότερα χαρακτηριστικά συγκριτικά με τη μέθοδο Boruta.

Ο συντελεστής κατάταξης συσχέτισης Spearman μέσα από την πειραματική διαδικασία αποδείχθηκε ότι είναι μια μέθοδος της οποίας η αποτελεσματικότητα εξαρτάται σε μεγάλο βαθμό από τις ιδιότητες και τις σχέσεις μεταξύ των στοιχείων του συνόλου δεδομένων. Ειδικότερα, κατά την πειραματική διαδικασία η συσχέτιση κατωφλίου για τον συντελεστή συσχέτισης κατάταξης Spearman ορίστηκε ίση με 0.2. Σύμφωνα με τα αποτελέσματα η μέθοδος φαίνεται να λειτουργεί αποτελεσματικά σε σχεδόν σε όλα τα σύνολα δεδομένων έχοντας ίση ή και μερικές φορές ακόμα και καλύτερα αποτελέσματα συγκριτικά με τις υπόλοιπες μεθόδους. Μοναδικές εξαιρέσεις παρουσιάζονται στα σύνολα δεδομένων Dry Bean Dataset και Digits Dataset όπου οι αποδόσεις των ταξινομητών είναι ελαφρώς χαμηλώτερες συγκριτικά με τις μεθόδους Boruta και εκμάθησης συνόλου. Τα παραπάνω επιβεβαιώνουν τα θεωρητικά χαρακτηριστικά της μεθόδου όπου τονίζεται η δυνατότητα αποτελεσματικής καταγραφής μονοτονικών και μη γραμμικών σχέσεων συσχέτισης μεταξύ των χαρακτηριστικών, αλλά όχι πιο σύνθετων σχέσεων που πιθανώς να παρουσιάστηκαν στα δύο σύνολα δεδομένων που η μέθοδος είχε φτωχή απόδοση.

Παρόμοια μέθοδος με τον συντελεστή κατάταξης συσχέτισης Spearman είναι και η μέθοδος του Kendall. Ο συγκεκριμένος συντελεστής σύμφωνα με το θεωρητικό υπόβαθρο της μεθόδου λειτουργεί αποτελεσματικά και σε σύνολα δεδομένων όπου κατά την ανάθεση των τιμών κατάταξης (ranks) προκύπτουν πολλές ισοπαλίες. Κατά την πειραματική διαδικασία η συσχέτιση κατωφλίου για τον συντελεστή συσχέτισης κατάταξης Kendall ορίστηκε ίση με 0.2. Σύμφωνα με τα πειραματικά αποτελέσματα λειτουργεί ελαφρώς χειρότερα από τον συντελεστή κατάταξης συσχέτισης Spearman. Ειδικότερα, στα σύνολα δεδομένων Dry Bean Dataset και Digits Dataset όπου ο συντελεστής συσχέτισης κατάταξης Spearman υπολειτουργεί ο συντελεστής του Kendall λειτουργεί ακόμα χειρότερα. Στα υπόλοιπα σύνολα δεδομένων λειτουργεί επιστρέφοντας παρόμοια ή ελαφρώς χειρότερα αποτελέσματα. Οι διαφορές στην απόδοση των δύο συντελεστών συσχέτισης πιθανώς να οφείλονται στους εσωτερικούς μηχανισμούς καταγραφής της συσχέτισης που ακολουθούν οι δύο μέθοδοι. Ειδικότερα, ο συντελεστής κατάταξης συσχέτισης Spearman υπολογίζει τη μονοτονική σχέση μεταξύ δύο μεταβλητών ενώ, ο συντελεστής κατάταξης

συσχέτισης Kendall υπολογίζει την τακτική συσχέτιση μεταξύ δύο μεταβλητών, το οποίο σύμφωνα με τα αποτέλεσματα είναι πιθανών να μην μπορεί να καταγράψει τις σχέσεις όσο αποτελεσματικά το κάνει ο συντελεστής κατάταξης συσχέτισης Spearman.

Τα παραπάνω συμπεράσματα, τονίζουν την αποτελεσματικότητα των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών κατά την ανάπτυξη μοντέλων μηχανικής μάθησης. Επιπλέον, δείχνουν πως πολλές φορές τεράστια σύνολα δεδομένων μπορούν να απλοποιηθούν σε μεγάλο βαθμό διατηρώντας παράλληλα την κύρια πληροφορία που οδηγεί τους ταξινομητές σε υψηλή απόδοση. Βέβαια, η παρούσα εργασία αναλύει και ελέγχει ένα μικρό κομμάτι του συγκεκριμένου πεδίου της επιστήμης. Ένας από τους μεγαλύτερους περιορισμούς της εργασίας ήταν το μικρό εύρος συνόλων δεδομένων. Τα επτά σύνολα δεδομένων αν και περιείχαν διαφορετικά εύρη χαρακτηριστικών και αριθμό στοιχείων, δεν μπορούν να αποτελέσουν ένα ικανοποιητικό δείγμα μέσω του οποίου θα ήταν δυνατό να εξαχθούν οι γενικότερες τάσεις μεταξύ των συνόλων δεδομένων και των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Επιπλέον, ένα άλλο κομμάτι που δεν μελετήθηκε σε μεγάλο βαθμό ήταν το πως τα μοντέλα εκμάθησης συνόλου επηρεάζουν τα αποτελέσματα των μεθόδων Boruta και της μεθόδου εκμάθησης συνόλου για επιλογή χαρακτηριστικών. Ειδικότερα τα διαφορετικά μοντέλα αξιοποιούν διαφορετικές τεχνικές εσωτερικά για την αξιολόγηση των χαρακτηριστικών το οποίο είναι σχεδόν σίγουρο ότι επηρεάζει και τα αποτελέσματα των μεθόδων επιλογής χαρακτηριστικών. Εμβαθύνοντας στο κομμάτι των παραμέτρων, πολλές από τις μεθόδους τόσο μείωσης διαστάσεων όσο και επιλογής χαρακτηριστικών απαιτούν την αρχικοποίηση υπερπαραμέτρων οι οποίες συμβάλλουν σε μεγάλο βαθμό στην αποδοτικότητα της μεθόδου, κάτι που δεν αναλύθηκε στη συγκεκριμένη εργασία εις βάθος. Τέλος αν και χρησιμοποιήθηκε εύρος ταξινομητών για την αξιολόγηση των μεθόδων, δεν αναλύθηκαν εις βάθος οι μέθοδοι νευρωνικών δικτύων εκτός από ένα απλό μοντέλο MLP.

Για την επέκταση των ευρημάτων και την αντιμετώπιση των περιορισμών της παρούσας εργασίας προτείνεται η μελλοντική έρευνα να επικεντρωθεί στην ανάλυση περισσότερων και μεγαλύτερων συνόλων δεδομένων. Στη διερεύνηση και αξιολόγηση της αποτελεσματικότητας των διαφορετικών μοντέλων εκμάθησης συνόλου σε

μεγαλύτερο βάθος, πως επηρεάζονται τα μοντέλα των Boruta και εκμάθησης συνόλου κατά την επιλογή των σημαντικών χαρακτηριστικών από το εκάστοτε μοντέλο και σε τι αποτελέσματα καταλήγουν οι ταξινομητές. Αν και η εργασία περιλάμβανε πολλές από τις πιο γνωστές μεθόδους μείωσης διαστάσεων και επιλογής χαρακτηριστικών, χρήσιμη θα ήταν η ανάλυση και πιο σύνθετων και νέων μεθόδων όπως οι Autoencoders και γενικότερα μοντέλα που βασίζονται στη βαθιά μάθηση. Σημαντικό κομμάτι είναι επίσης το κομμάτι της ταξινόμησης. Στην έρευνα χρησιμοποιήθηκαν πολλοί ταξινομητές αλλά δεν διερευνήθηκε η βέλτιστη παραμετροποίηση των ταξινομητών ώστε να επιτύχουν την καλύτερη απόδοση, σε μελλοντική έρευνα κρίνεται σκόπιμο να πραγματοποιηθούν πειράματα στα οποία οι ταξινομητές αρχικοποιούνται στις βέλτιστες υπερπαραμέτρους τους. Άξιο κομμάτι για μελλοντική μελέτη αποτελεί επίσης η συνδυαστική ανάλυση των μεθόδων μείωσης διαστάσεων και επιλογής χαρακτηριστικών. Ειδικότερα, πως επηρεάζονται τα μοντέλα όταν πρώτα εφαρμοστεί μια μέθοδος επιλογής των σημαντικότερων χαρακτηριστικών και έπειτα εφαρμοστεί μια μέθοδος μείωσης διαστάσεων στο υποσύνολο των χαρακτηριστικών ή και το αντίθετο. Τέλος, μια διαφορετική πτυχή που δεν ερευνήθηκε είναι μέθοδοι και τρόποι προεπεξεργασίας των δεδομένων με σκοπό τη μεγιστοποίηση της αποτελεσματικότητας των μεθόδων μείωσης διαστάσεων, δηλαδή τρόποι οι οποίοι θα προετοιμάσουν καλύτερα τα δεδομένα ώστε όταν εφαρμοστούν οι μέθοδοι μείωσης διαστάσεων και επιλογής χαρακτηριστικών να εξάγουν ακόμα καλύτερα αποτελέσματα.

Βιβλιογραφία

[1] C. Ding and H. Peng, "MINIMUM REDUNDANCY FEATURE SELECTION FROM MICROARRAY GENE EXPRESSION DATA."

[2] C. Lai, M. J. Reinders, L. J. Van'T Veer, and L. F. Wessels, "A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets," vol. 7, no. 1, p. 235. [Online]. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-235

[3] X. Zheng, Y. Yuan, and X. Lu, "Dimensionality reduction by spatial–spectral preservation in selected bands," vol. 55, no. 9, pp. 5185–5197. [Online]. Available: http://ieeexplore.ieee.org/document/7954794/

[4] N. Venkat, "The curse of dimensionality: Inside out."

[5] D. L. Padmaja and B. Vishnuvardhan, "Comparative study of feature subset selection methods for dimensionality reduction on scientific data," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. IEEE, pp. 31–34. [Online]. Available: http://ieeexplore.ieee.org/document/7544805/

[6] X. Ying, "An overview of overfitting and its solutions."

[7] B. Baesens, S. Höppner, and T. Verdonck, "Data engineering for fraud detection," vol. 150, p. 113492. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167923621000026

[8] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," vol. 23, no. 1, pp. 241–250. [Online]. Available: http://ieeexplore.ieee.org/document/7536217/

[9] F. L. Gewers, G. R. Ferreira, H. F. de Arruda, F. N. Silva, C. H. Comin, D. R. Amancio, and L. d. F. Costa, "Principal component analysis: A natural approach to data exploration," vol. 54, no. 4, pp. 1–34. [Online]. Available: http://arxiv.org/abs/1804.02502

[10] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," vol. 374, no. 2065, p. 20150202. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202

[11] N. A. Qureshi, V. Suthar, H. Magsi, M. J. Sheikh, M. Pathan, and B. Qureshi, "Application of principal component analysis (pca) to medical data," *Indian Journal*

of Science and Technology, vol. 10, no. 20, May 2017. [Online]. Available: https://ischolar.sscldl.in/index.php/indjst/article/view/157283

[12] D. Dimakopoulou-Papazoglou, N. Ploskas, S. Serrano, C. S. Silva, V. Valdramidis, K. Koutsoumanis, and E. Katsanidis, "Application of UV–vis spectroscopy for the detection of adulteration in mediterranean honeys," vol. 249, no. 12, pp. 3043–3053. [Online]. Available: https://link.springer.com/10.1007/s00217-023-04347-1

[13] S. Ding, P. Zhang, E. Ding, A. Naik, P. Deng, and W. Gui, "On the application of pca technique to fault diagnosis," *Tsinghua Science and Technology*, vol. 15, no. 2, pp. 138–144, 2010.

[14] S. Lahabar and P. J. Narayanan, "Singular value decomposition on gpu using cuda," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, 2009, pp. 1–10.

[15] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 1st ed. Cambridge University Press. [Online]. Available: https://www.cambridge.org/core/product/identifier/9781108380690/type/book

[16] S. J. Li, H. Pang, P. Y. Li, Y. N. Li, and Z. X. Liu, "Image compression based on svd algorithm," in *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, 2021, pp. 306–309.

[17] M. Said, K. Mahar, and K. Eskaf, "Gene annotations prediction using singular value decomposition and independent component analysis," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017, pp. 1957–1961.

[18] J. Cheng, D. Yu, J. Tang, and Y. Yang, "Application of SVM and SVD technique based on EMD to the fault diagnosis of the rotating machinery," vol. 16, no. 1, pp. 89–98. [Online]. Available: http://www.hindawi.com/journals/sv/2009/519502/

[19] A. Spooner, G. Mohammadi, P. S. Sachdev, H. Brodaty, A. Sowmya, and for the Sydney Memory and Ageing Study and the Alzheimer's Disease Neuroimaging Initiative, "Ensemble feature selection with data-driven thresholding for alzheimer's disease biomarker discovery," vol. 24, no. 1, p. 9. [Online]. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-022-05132-9

[20] A. Gumaei, R. Sammouda, M. Al-Rakhami, H. AlSalman, and A. El-Zaart, "Feature selection with ensemble learning for prostate cancer diagnosis from microarray gene expression," vol. 27, no. 1, p. 146045822198940. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1460458221989402

[21] S. Van Landeghem, T. Abeel, Y. Saeys, and Y. Van De Peer, "Discriminative and informative features for biomolecular text mining with ensemble feature selection," vol. 26, no. 18, pp. i554–i560. [Online]. Available: https://academic.oup.com/bioinformatics/article/26/18/i554/205975

[22] B. Shravankumar and V. Ravi, "Text classification using ensemble features selection and data mining techniques," 07 2015, pp. 176–186.

[23] M.-T. Puth, M. Neuhäuser, and G. D. Ruxton, "Effective use of spearman's and kendall's correlation coefficients for association between two measured traits," *Animal Behaviour*, vol. 102, pp. 77–84, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003347215000196

[24] C. Xiao, J. Ye, R. M. Esteves, and C. Rong, "Using spearman's correlation coefficients for exploratory data analysis on big dataset," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 14, pp. 3866–3878, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3745

[25] W.-Y. Zhang, Z.-W. Wei, B.-H. Wang, and X.-P. Han, "Measuring mixing patterns in complex networks by spearman rank correlation coefficient," *Physica A: Statistical Mechanics and its Applications*, vol. 451, pp. 440–450, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437116001023

[26] P. Legendre, "Species associations: the kendall coefficient of concordance revisited," *Journal of Agricultural, Biological, and Environmental Statistics*, vol. 10, no. 2, pp. 226–245, 6 2005. [Online]. Available: https://doi.org/10.1198/108571105X46642

[27] C. L. Comella, S. Leurgans, J. Wuu, G. T. Stebbins, T. Chmura, , and T. D. S. Group, "Rating scales for dystonia: A multicenter assessment," *Movement Disorders*, vol. 18, no. 3, pp. 303–312, 2003. [Online]. Available: https://movementdisorders.onlinelibrary.wiley.com/doi/abs/10.1002/mds.10377

[28] M. C. Hout, M. H. Papesh, and S. D. Goldinger, "Multidimensional scaling," vol. 4, no. 1, pp. 93–103. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/10.1002/wcs.1203

[29] M. L. L. N. D. H. H. Andreas Buja, Deborah F Swayne and L. Chen, "Data visualization with multidimensional scaling," *Journal of Computational and Graphical Statistics*, vol. 17, no. 2, pp. 444–472, 2008. [Online]. Available: https://doi.org/10.1198/106186008X318440

[30] Multidimensional scaling - wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Multidimensional_scaling

[31] J. A. N. Lee and M. Verleysen, *Nonlinear dimensionality reduction*, ser. Information science and statistics. Springer Science + Business Media, specific pages consulted: 98-110.

[32] E. P. Lessa, "Multidimensional analysis of geographic genetic structure," vol. 39, no. 3, p. 242. [Online]. Available: https://academic.oup.com/sysbio/article-lookup/doi/10.2307/2992184

[33] L. G. Cooper, "A review of multidimensional scaling in marketing research," vol. 7, no. 4, pp. 427–450. [Online]. Available: http://journals.sagepub.com/doi/10.1177/014662168300700404

[34] S. V. Shinkareva, J. Wang, and D. H. Wedell, "Examining similarity structure: Multidimensional scaling and related approaches in neuroimaging," vol. 2013, pp. 1–9. [Online]. Available: http://www.hindawi.com/journals/cmmm/2013/796183/

[35] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," vol. 290, no. 5500, pp. 2319–2323. [Online]. Available: https://www.science.org/doi/10.1126/science.290.5500.2319

[36] V. Sreeram and P. Agathoklis, "On the properties of gram matrix," vol. 41, no. 3, pp. 234–237. [Online]. Available: http://ieeexplore.ieee.org/document/273922/

[37] T. Dear, R. L. Hatton, and H. Choset, "Nonlinear dimensionality reduction for kinematic cartography with an application toward robotic locomotion," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3604–3609. [Online]. Available: http://ieeexplore.ieee.org/document/6943067/

[38] S. Weng, C. Zhang, Z. Lin, and X. Zhang, "Mining the structural knowledge of high-dimensional medical data using isomap," vol. 43, no. 3, pp. 410–412. [Online]. Available: http://link.springer.com/10.1007/BF02345820

[39] Z. Shenglin and Z. Shan-an, "Face recognition by lle dimensionality reduction," in *2011 Fourth International Conference on Intelligent Computation Technology and Automation*, vol. 1, 2011, pp. 121–123.

[40] J. Nichols, F. Bucholtz, and B. Nousain, "Automated, rapid classification of signals using locally linear embedding," vol. 38, no. 10, pp. 13 472–13 474. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417411006737

[41] Y. Zhang, D. Ye, and Y. Liu, "Robust locally linear embedding algorithm for machinery fault diagnosis," vol. 273, pp. 323–332. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217313528

[42] G. Potamianos and H. Graf, "Linear discriminant analysis for speechreading," in *1998 IEEE Second Workshop on Multimedia Signal Processing (Cat. No.98EX175)*, 1998, pp. 221–226.

[43] S. K. Bhattacharyya and K. Rahul, "FACE RECOGNITION BY LINEAR DISCRIMINANT ANALYSIS," pp. 1–5. [Online]. Available: https://www.interscience.in/cgi/viewcontent.cgi?article=1087&context=ijcns

[44] C. Ricciardi, A. S. Valente, K. Edmund, V. Cantoni, R. Green, A. Fiorillo, I. Picone, S. Santini, and M. Cesarelli, "Linear discriminant analysis and principal component analysis to predict coronary artery disease," vol. 26, no. 3, pp. 2181–2192. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1460458219899210

[45] S. G. Kwak and J. H. Kim, "Central limit theorem: the cornerstone of modern statistics," *Korean Journal of Anesthesiology*, vol. 70, no. 2, pp. 144–156, 4 2017.

[46] J. Stone, "Independent component analysis: An introduction," *Trends in cognitive sciences*, vol. 6, pp. 59–64, 03 2002.

[47] T.-W. Lee, "Independent component analysis," in *Independent component analysis: Theory and applications*. Springer US, pp. 27–66. [Online]. Available: https://doi.org/10.1007/978-1-4757-2851-4_2

[48] Y. B. Monakhova, S. A. Astakhov, S. P. Mushtakova, and L. A. Gribov, "Methods of the decomposition of spectra of various origin in the analysis of complex mixtures," vol. 66, no. 4, pp. 351–362. [Online]. Available: http://link.springer.com/10.1134/S1061934811040137

[49] S. Vorobyov and A. Cichocki, "Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis," vol. 86, no. 4, pp. 293–303. [Online]. Available: http://link.springer.com/10.1007/s00422-001-0298-6

[50] J. V. Stone, "Independent component analysis: an introduction," *Trends in Cognitive Sciences*, vol. 6, no. 2, pp. 59–64, 2002. [Online]. Available: https://doi.org/10.1016/s1364-6613(00)01813-1

[51] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models." [Online]. Available: http://arxiv.org/abs/1207.3538

[52] S. Mika, B. Scholkopf, A. Smola, K.-R. Muller, M. Scholz, and G. Riitsch, "Kernel peA and de-noising in feature spaces."

[53] T. Szul, S. Tabor, and K. Pancerz, "Application of the BORUTA algorithm to input data selection for a model based on rough set theory (RST) to prediction energy consumption for building heating," vol. 14, no. 10, p. 2779. [Online]. Available: https://www.mdpi.com/1996-1073/14/10/2779

[54] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, "Boruta – a system for feature selection," vol. 101, no. 4, pp. 271–285. [Online]. Available: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/FI-2010-288

[55] L. K. Leong and A. A. Abdullah, "Prediction of alzheimer's disease (AD) using machine learning techniques with boruta algorithm as feature selection method," vol. 1372, no. 1, p. 012065. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1372/1/012065

[56] W. Li, J. E. Cerise, Y. Yang, and H. Han, "Application of t-SNE to human genetic data," vol. 15, no. 4, p. 1750017. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0219720017500172

[57] P. Hajibabaee, F. Pourkamali-Anaraki, and M. A. Hariri-Ardebili, "An empirical evaluation of the t-SNE algorithm for data visualization in structural engineering," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 1674–1680. [Online]. Available: https://ieeexplore.ieee.org/document/9680055/

[58] R. A. Hamad, E. Jarpe, and J. Lundstrom, "Stability analysis of the t-SNE algorithm for human activity pattern data," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 1839–1845. [Online]. Available: https://ieeexplore.ieee.org/document/8616314/

[59] M. Usman, S. Ahmed, J. Ferzund, A. Mehmood, and A. Rehman, "Using pca and factor analysis for dimensionality reduction of bio-informatics data," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 5, 2017. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2017.080551

[60] I. El Moudden, M. Ouzir, and S. ElBernoussi, "Automatic speech analysis in patients with parkinson's disease using feature dimension reduction," in *Proceedings of the 3rd International Conference on Mechatronics and Robotics Engineering*, ser. ICMRE 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 167–171. [Online]. Available: https://doi.org/10.1145/3068796.3068813

[61] S. T. Tu, J. Y. Chen, W. Yang, and H. Sun, "Laplacian eigenmaps-based polarimetric dimensionality reduction for sar image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 1, pp. 170–179, 2012.

[62] A. Brun, H.-J. Park, H. Knutsson, and C.-F. Westin, "Coloring of dt-mri fiber traces using laplacian eigenmaps," vol. 2809, 02 2003, pp. 518–529.

[63] W. Luo, "Face recognition based on laplacian eigenmaps," in *2011 International Conference on Computer Science and Service System (CSSS)*, 2011, pp. 416–419.

[64] Lin Hu, Hong Ma, and Li Cheng, "Method of noise reduction based on SVD and its application in digital receiver front-end," in *2012 18th Asia-Pacific Conference on Communications (APCC)*. IEEE, pp. 511–515. [Online]. Available: http://ieeexplore.ieee.org/document/6388246/

[65] J. Guo, R. Xie, and G. Jin, "An efficient method for NMR data compression based on fast singular value decomposition," vol. 16, no. 2, pp. 301–305. [Online]. Available: https://ieeexplore.ieee.org/document/8491389/

[66] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," vol. 35, no. 6, pp. 1098–1107. [Online]. Available: http://ieeexplore.ieee.org/document/1542257/

[67] S. Nanga, A. T. Bawah, B. A. Acquaye, M.-I. Billa, F. D. Baeta, N. A. Odai, S. K. Obeng, and A. D. Nsiah, "Review of dimension reduction methods," vol. 09, no. 3, pp. 189–231. [Online]. Available: https://www.scirp.org/journal/doi.aspx?doi=10.4236/jdaip.2021.93013

[68] M. Balasubramanian and E. L. Schwartz, "The isomap algorithm and topological stability," vol. 295, no. 5552, pp. 7–7. [Online]. Available: https://www.science.org/doi/10.1126/science.295.5552.7a

[69] M. Yousaf, T. U. Rehman, and L. Jing, "An extended isomap approach for nonlinear dimension reduction," vol. 1, no. 3, p. 160. [Online]. Available: https://link.springer.com/10.1007/s42979-020-00179-y

[70] S. De Backer, A. Naud, and P. Scheunders, "Non-linear dimensionality reduction techniques for unsupervised feature extraction," vol. 19, no. 8, pp. 711–720. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016786559800049X

[71] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE)," vol. 40, p. 100378. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1574013721000186

[72] S. V.S and S. Surendran, "A review of various linear and non linear dimensionality reduction techniques," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:44831898

[73] L. van der Maaten, E. Postma, and H. Herik, "Dimensionality reduction: A comparative review," *Journal of Machine Learning Research - JMLR*, vol. 10, 01 2007.

[74] M. A. Mendez, "Linear and nonlinear dimensionality reduction from fluid mechanics to machine learning," vol. 34, no. 4, p. 042001. [Online]. Available: https://iopscience.iop.org/article/10.1088/1361-6501/acaffe

[75] A. K. Rastogi, S. Taterh, and B. S. Kumar, "Dimensionality reduction algorithms in machine learning: A theoretical and experimental comparison," in *RAiSE-2023*. MDPI, p. 82. [Online]. Available: https://www.mdpi.com/2673-4591/59/1/82

[76] T. ArchanaH. and D. Sachin, "Dimensionality reduction and classification through PCA and LDA," vol. 122, no. 17, pp. 4–8. [Online]. Available: http://research.ijcaonline.org/volume122/number17/pxc3905104.pdf

[77] A. A. Joy, M. A. M. Hasan, and M. A. Hossain, "A comparison of supervised and unsupervised dimension reduction methods for hyperspectral image classification," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8679360/

[78] Y. Halpern, S. Horng, L. A. Nathanson, N. I. Shapiro, and D. Sontag, "A comparison of dimensionality reduction techniques for unstructured clinical text."

[79] T. Balachander, R. Kothari, and H. Cualing, "An empirical comparison of dimensionality reduction techniques for pattern classification," in *Artificial Neural Networks — ICANN'97*, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Eds. Springer Berlin Heidelberg, vol. 1327, pp. 589–594, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/BFb0020218

[80] Q. Fournier and D. Aloise, "Empirical comparison between autoencoders and traditional dimensionality reduction methods," in *2019 IEEE Second International Conference on Artificial*

Intelligence and Knowledge Engineering (AIKE). IEEE, pp. 211–214. [Online]. Available: https://ieeexplore.ieee.org/document/8791727/

[81] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*. IEEE, pp. 372–378. [Online]. Available: https://ieeexplore.ieee.org/document/6918213

[82] P. Ray, S. S. Reddy, and T. Banerjee, "Various dimension reduction techniques for high dimensional data analysis: a review," vol. 54, no. 5, pp. 3473–3515. [Online]. Available: https://link.springer.com/10.1007/s10462-020-09928-0

[83] M. A. Salam, A. Taher, M. Samy, and K. Mohamed, "The effect of different dimensionality reduction techniques on machine learning overfitting problem," vol. 12, no. 4. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=12&Issue=4&Code=IJACSA&SerialNo=80

[84] F. Plastria, S. De Bruyne, and E. Carrizosa, "Dimensionality reduction for classification," in *Advanced Data Mining and Applications*, C. Tang, C. X. Ling, X. Zhou, N. J. Cercone, and X. Li, Eds. Springer Berlin Heidelberg, vol. 5139, pp. 411–418, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-540-88192-6_38

[85] F. R. On, R. Jailani, S. L. Hassan, and N. M. Tahir, "Analysis of singular value decomposition using high dimensionality data," in *2015 IEEE 11th International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE, pp. 186–191. [Online]. Available: http://ieeexplore.ieee.org/document/7225643/

[86] R. D. Ledesma, P. Valero-Mora, and G. Macbeth, "The scree test and the number of factors: a dynamic graphics approach," vol. 18, p. E11. [Online]. Available: https://www.cambridge.org/core/product/identifier/S113874161500013X/type/journal_article

[87] sklearn.datasets.load_digits — scikit-learn 1.4.2 documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html

[88] S. Aeberhard and M. Forina, "Wine," UCI Machine Learning Repository, 1991, DOI: https://doi.org/10.24432/C5PC7J.

[89] M. O. S. N. Wolberg, William and W. Street, "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1995, DOI: https://doi.org/10.24432/C5DW2B.

[90] "Dry Bean," UCI Machine Learning Repository, 2020, DOI: https://doi.org/10.24432/C50S4B.

[91] W. S. H. L. Sigillito, V. and K. Baker, "Ionosphere," UCI Machine Learning Repository, 1989, DOI: https://doi.org/10.24432/C5W01B.

[92]

[93] D. Chapman and A. Jain, "Musk (Version 1)," UCI Machine Learning Repository, 1994, DOI: https://doi.org/10.24432/C5ZK5B.

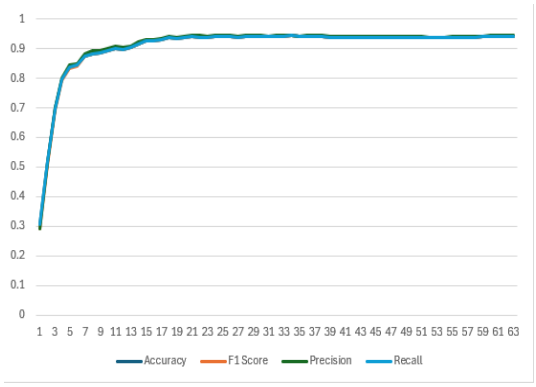
Παραρτήματα

Παράρτημα Α΄

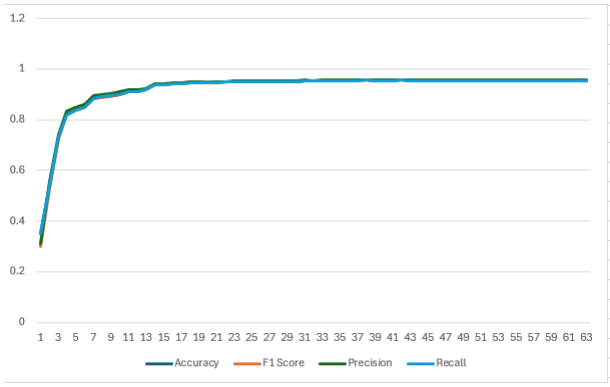
Αποτελέσματα αλγορίθμων

Α΄.1 Αποτελέσματα PCA

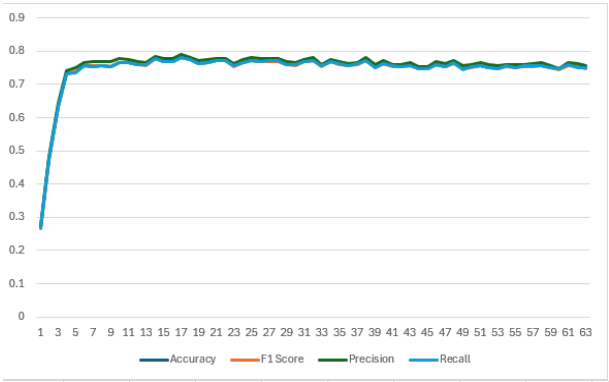
Α΄.1.1 Digits Dataset



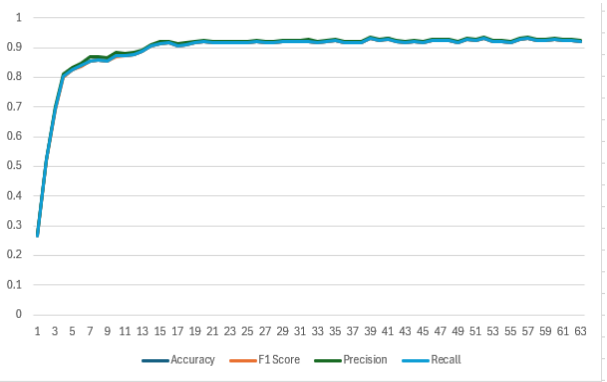
(α΄) KNN



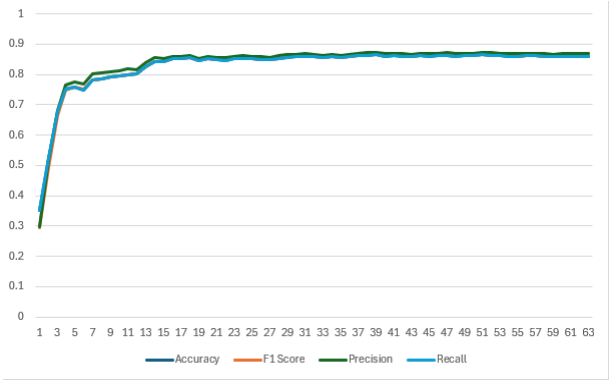
(β΄) SVM



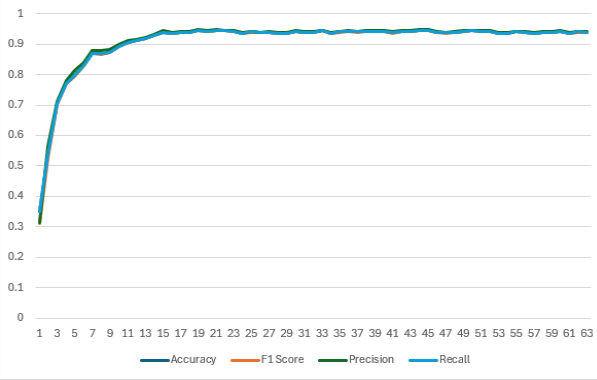
(γ΄) Decision Tree



(δ΄) Random Forest



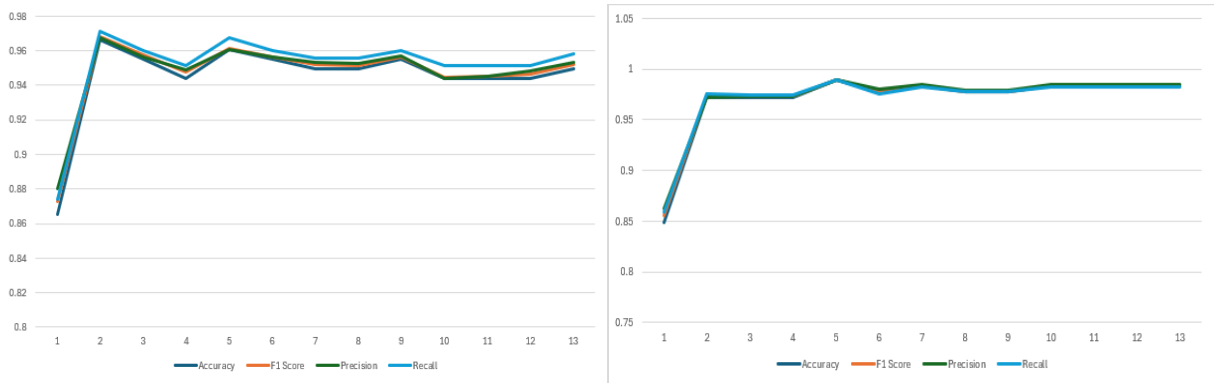
(ε΄) Naive Bayes



(στ΄) MLP

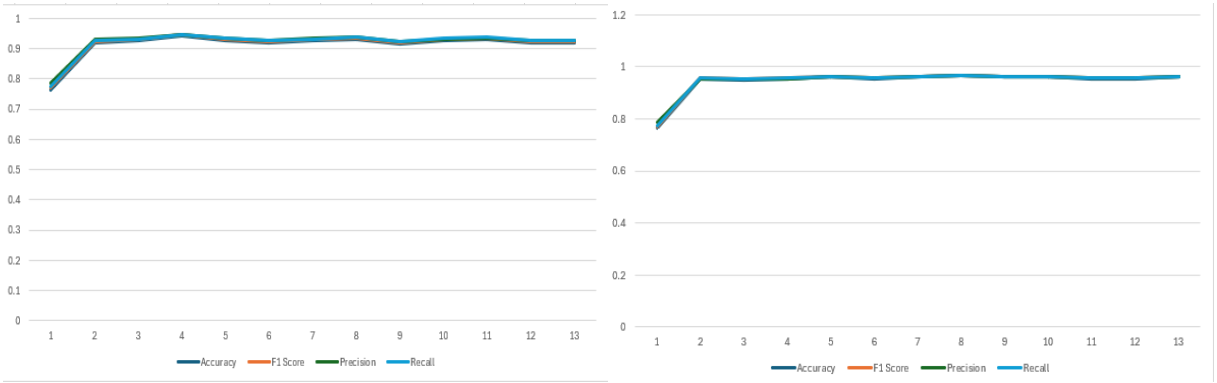
Σχήμα Α΄.1: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο PCA

A΄.1.2 Wine Dataset



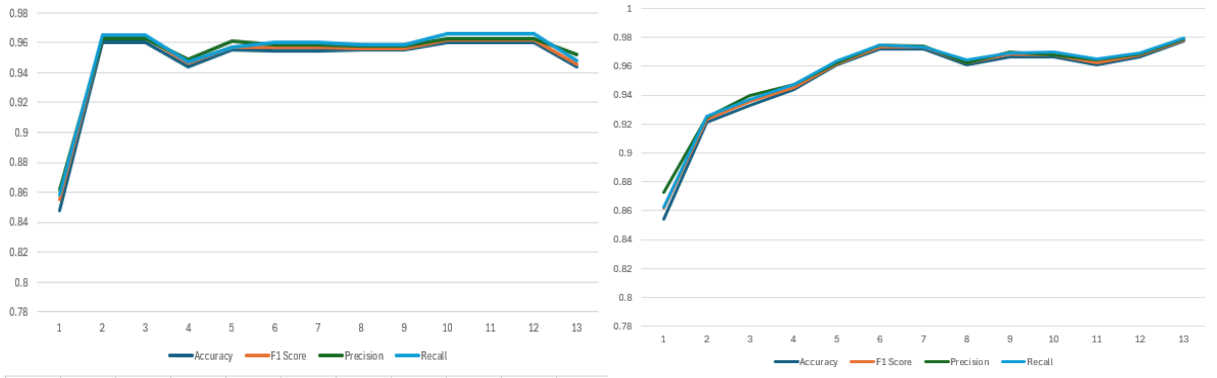
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest

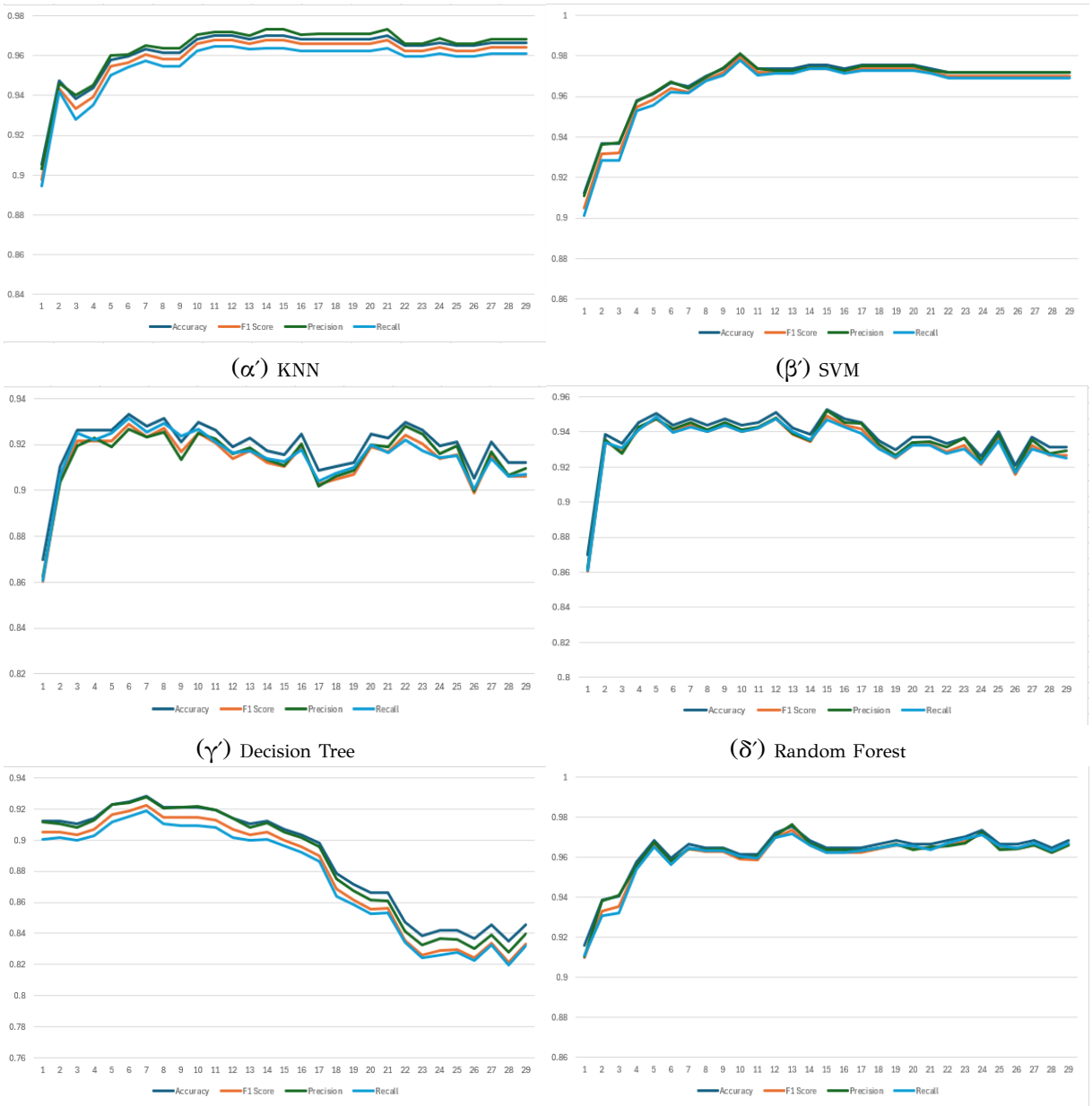


(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.2: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο PCA

Α΄.1.3 Breast Cancer Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

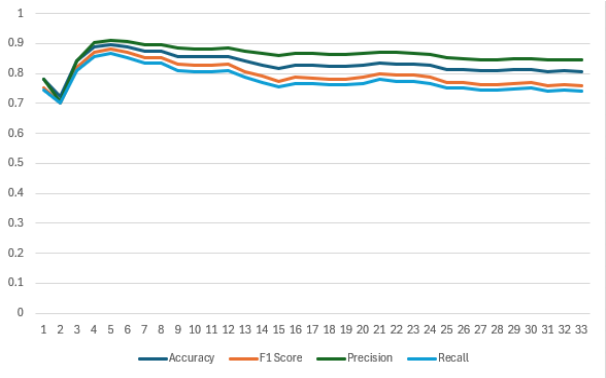
(δ΄) Random Forest

(ε΄) Naive Bayes

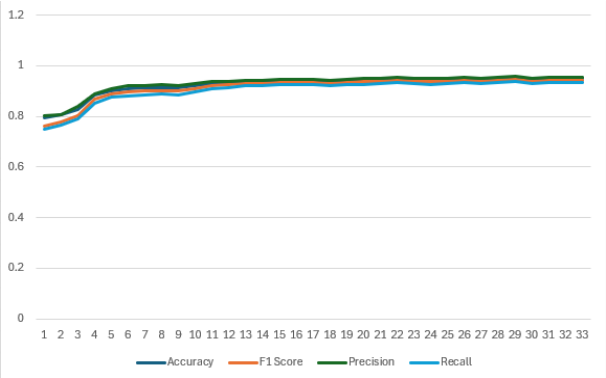
(στ΄) MLP

Σχήμα Α΄.3: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο PCA

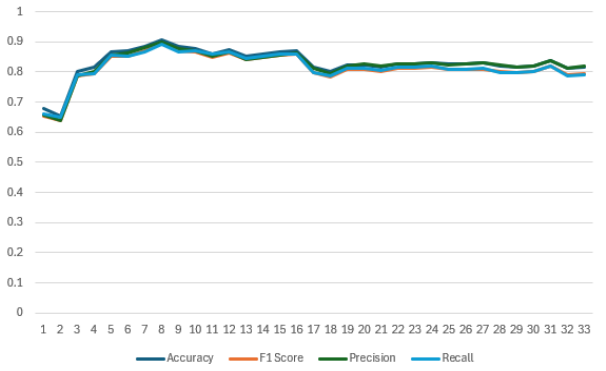
Α΄.1.4 Ionosphere Dataset



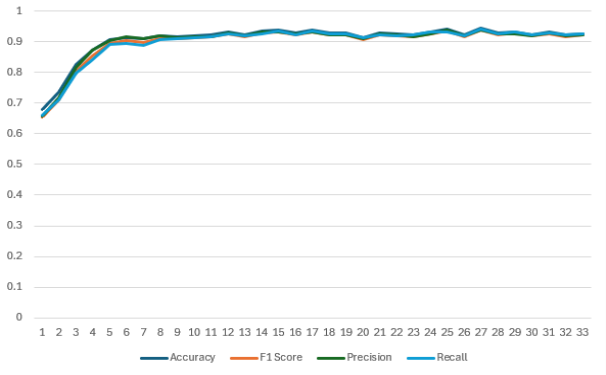
(α΄) KNN



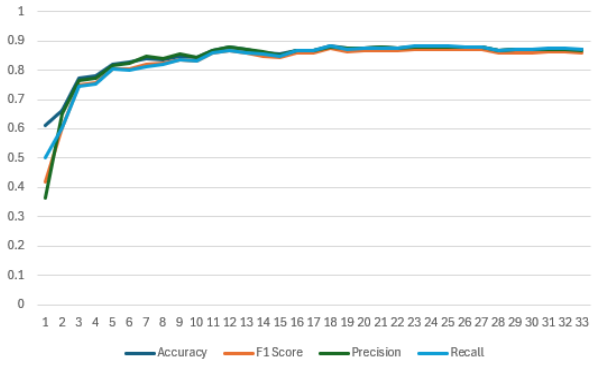
(β΄) SVM



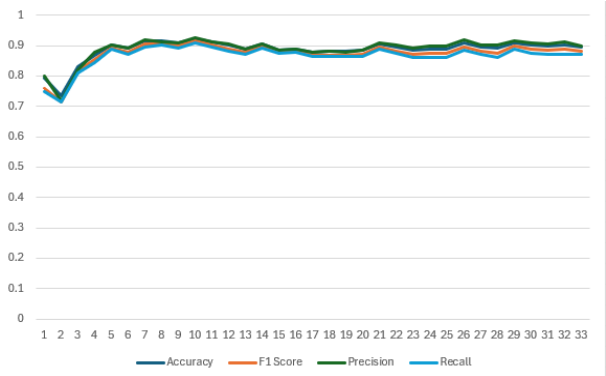
(γ΄) Decision Tree



(δ΄) Random Forest



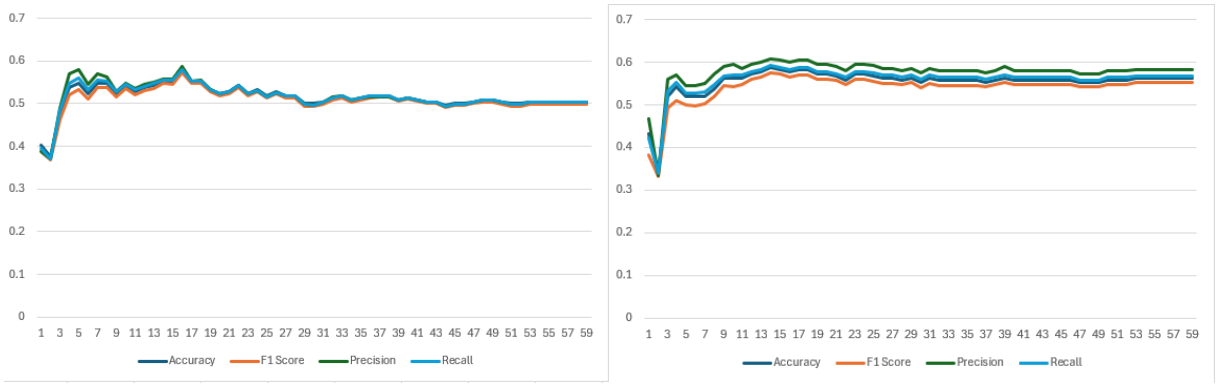
(ε΄) Naive Bayes



(στ΄) MLP

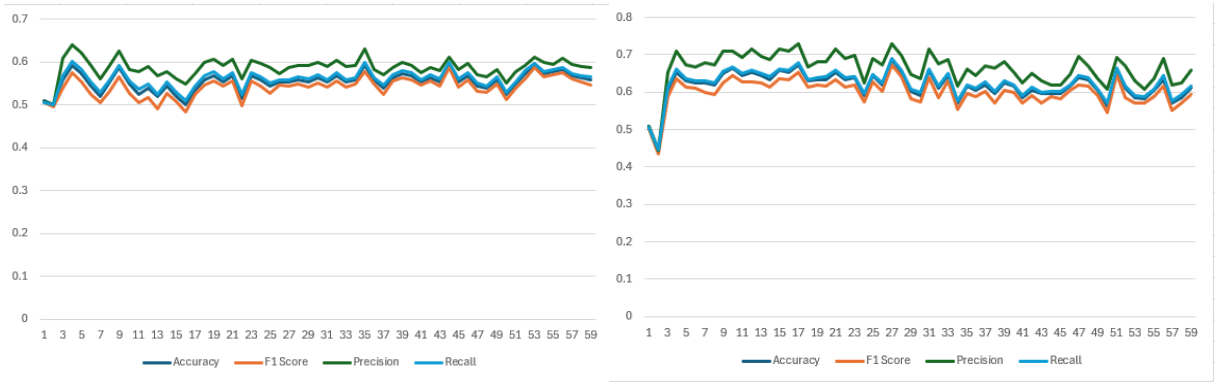
Σχήμα Α΄.4: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο PCA

Α΄.1.5 ok Connectionist Bench Dataset



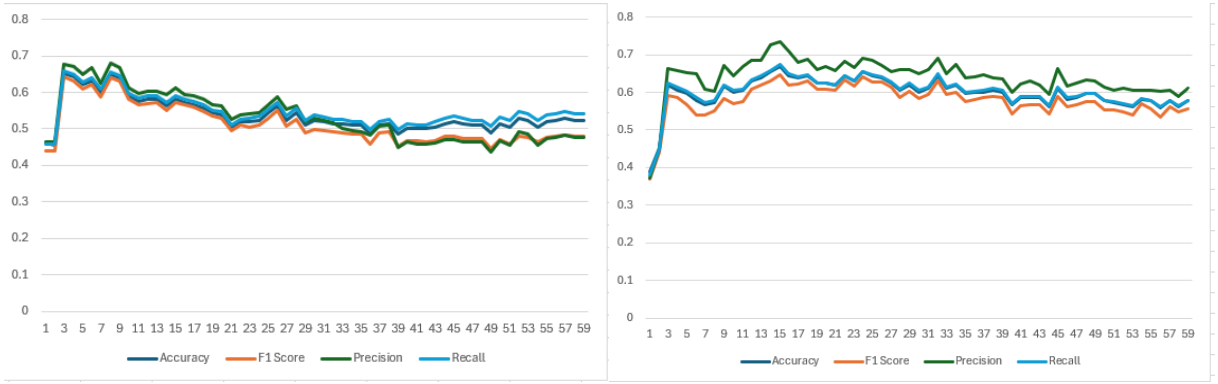
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest

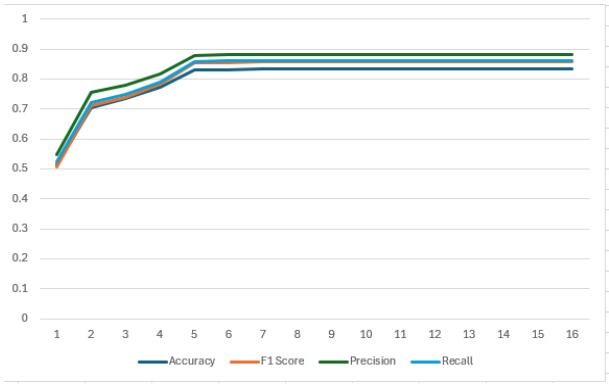


(ε΄) Naive Bayes

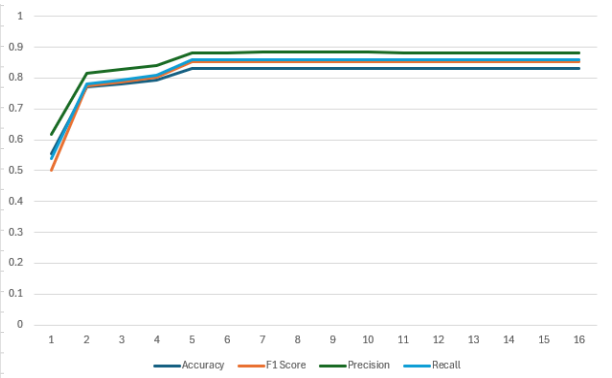
(στ΄) MLP

Σχήμα Α΄.5: Ανάλυση διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο PCA

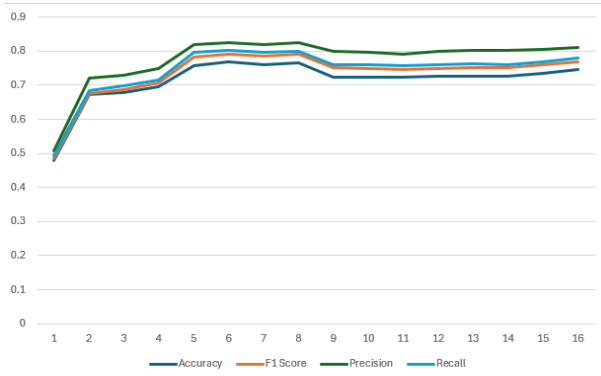
Α΄.1.6 Dry Bean Dataset



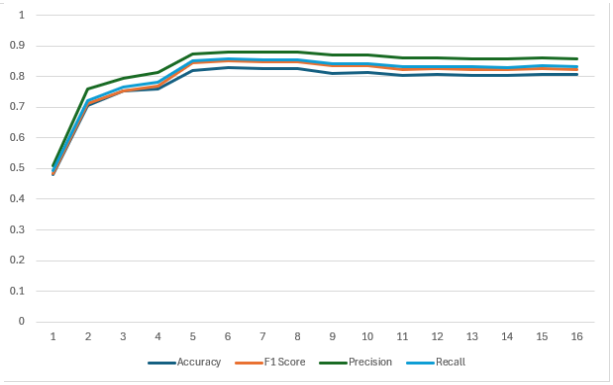
(α΄) KNN



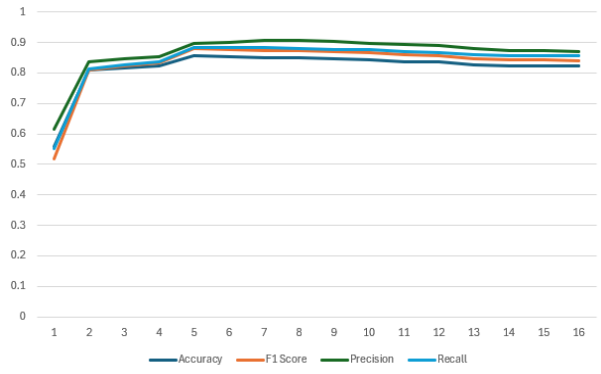
(β΄) SVM



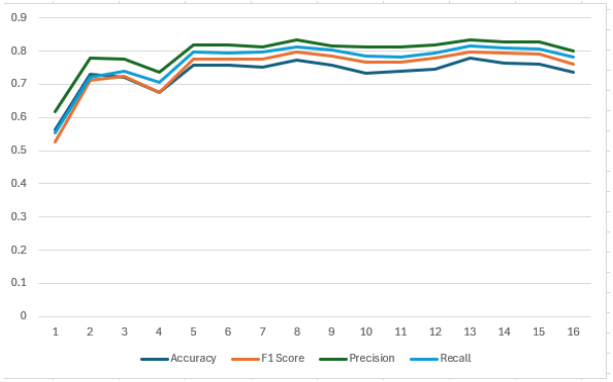
(γ΄) Decision Tree



(δ΄) Random Forest



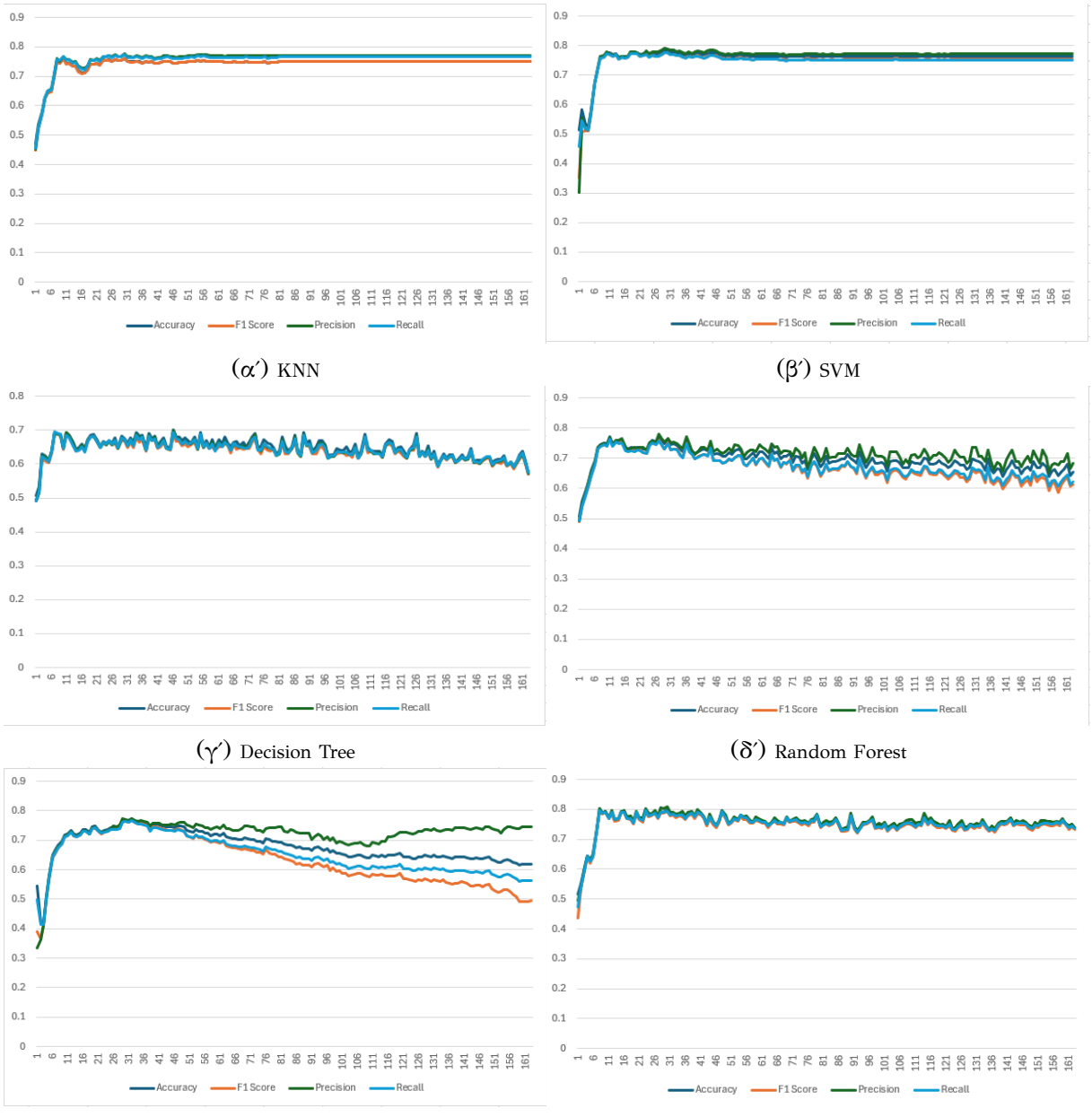
(ε΄) Naive Bayes



(στ΄) MLP

Σχήμα Α΄.6: Ανάλυση διαστάσεων στο Dry Beans Dataset με τη μέθοδο PCA

Α΄.1.7 Musk Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

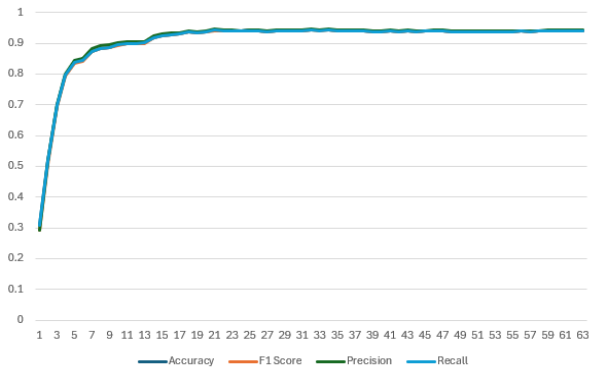
(ε΄) Naive Bayes

(στ΄) MLP

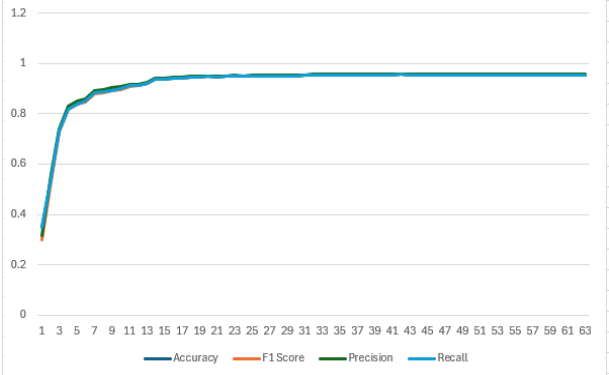
Σχήμα Α΄.7: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο PCA

Α΄.2 Αποτελέσματα Kernel PCA

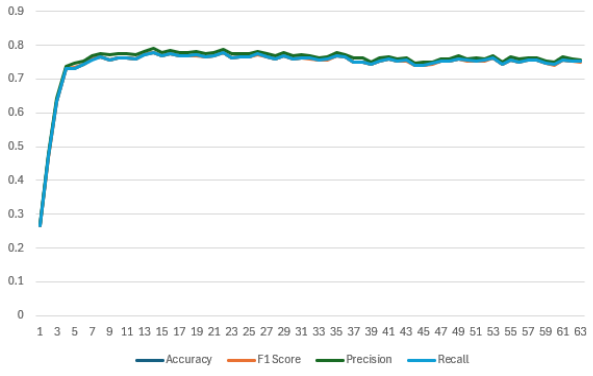
Α΄.2.1 Digits Dataset



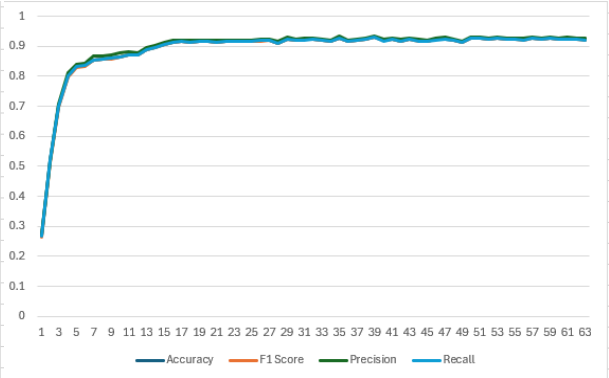
(α΄) KNN



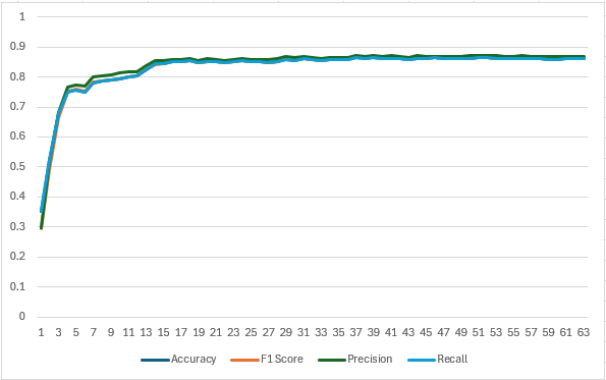
(β΄) SVM



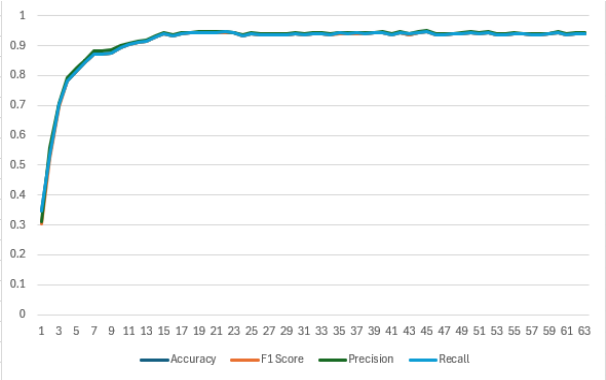
(γ΄) Decision Tree



(δ΄) Random Forest



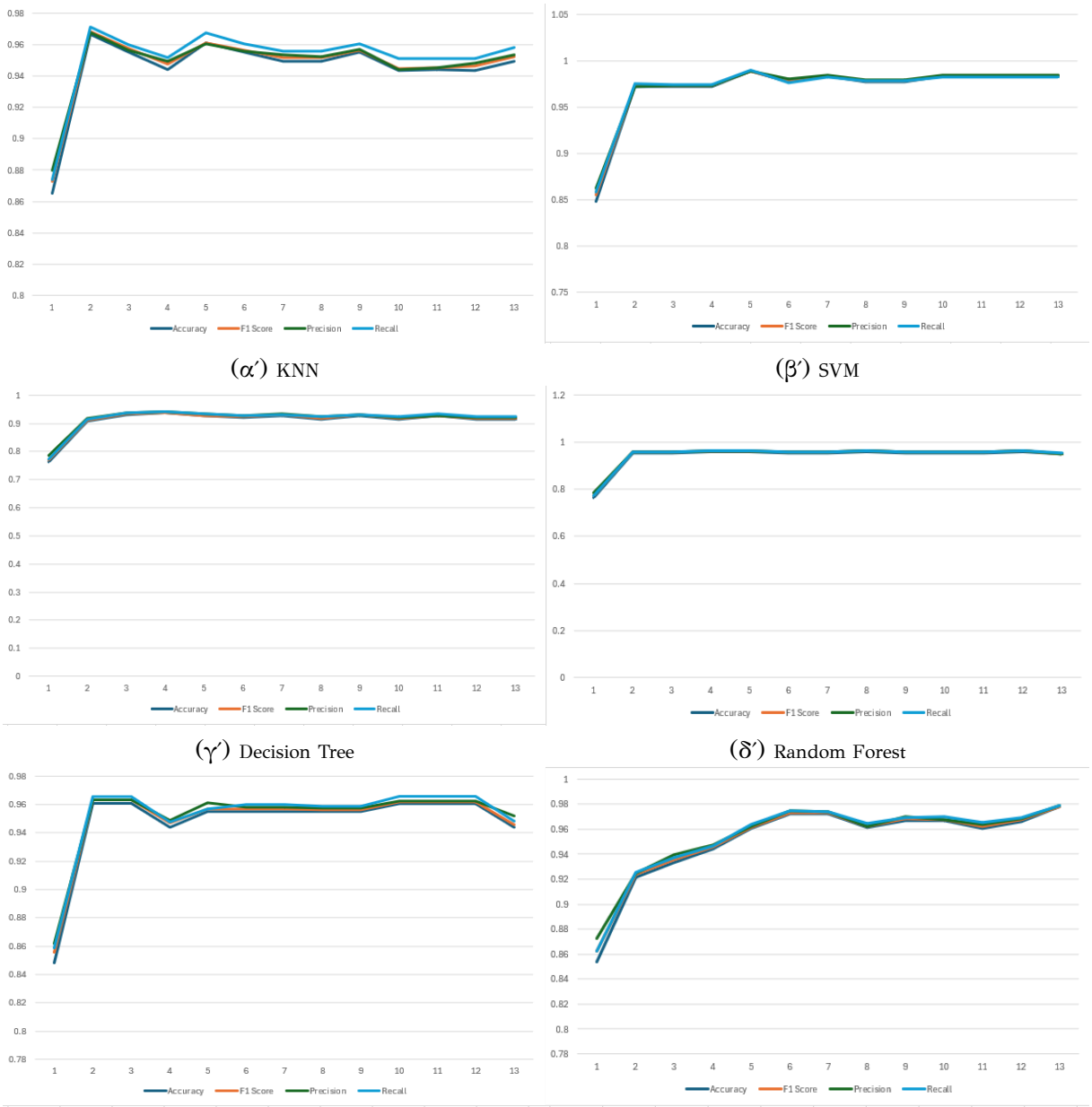
(ε΄) Naive Bayes



(στ΄) MLP

Σχήμα Α΄.8: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο Kernel PCA

Α΄.2.2 Wine Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

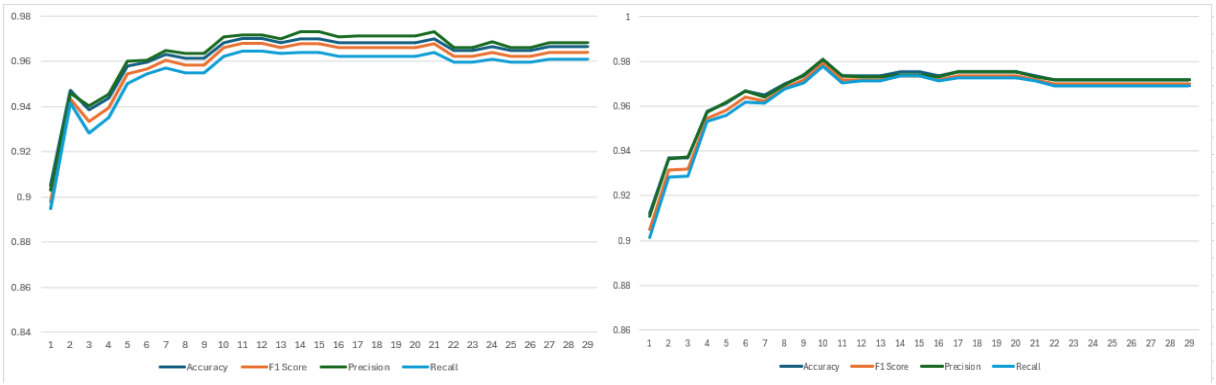
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

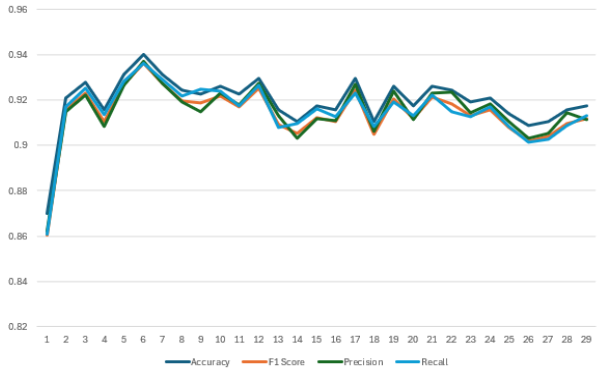
Σχήμα Α΄.9: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο Kernel PCA

A′.2.3 Breast Cancer Dataset

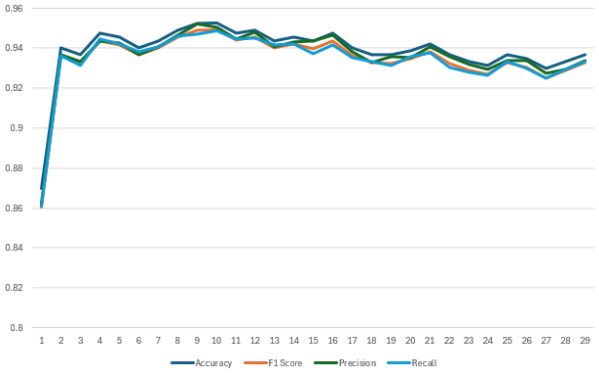


(α′) KNN

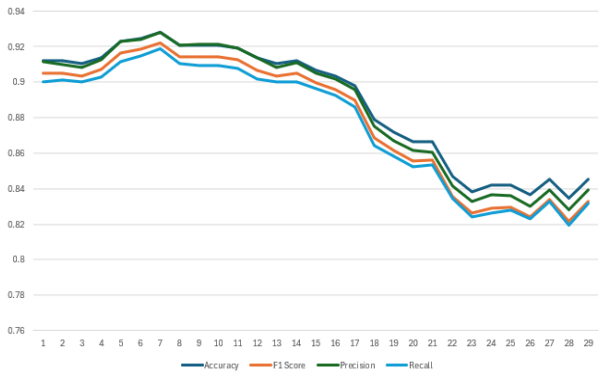
(β′) SVM



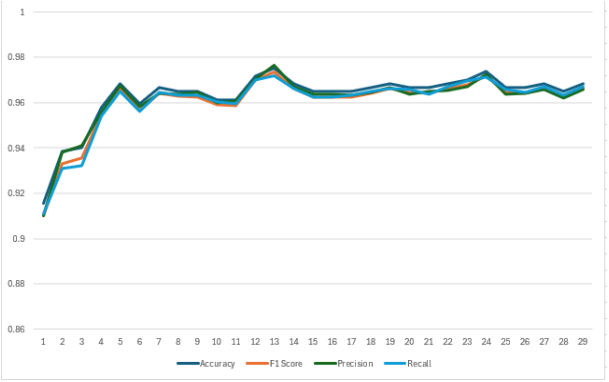
(γ′) Decision Tree



(δ′) Random Forest



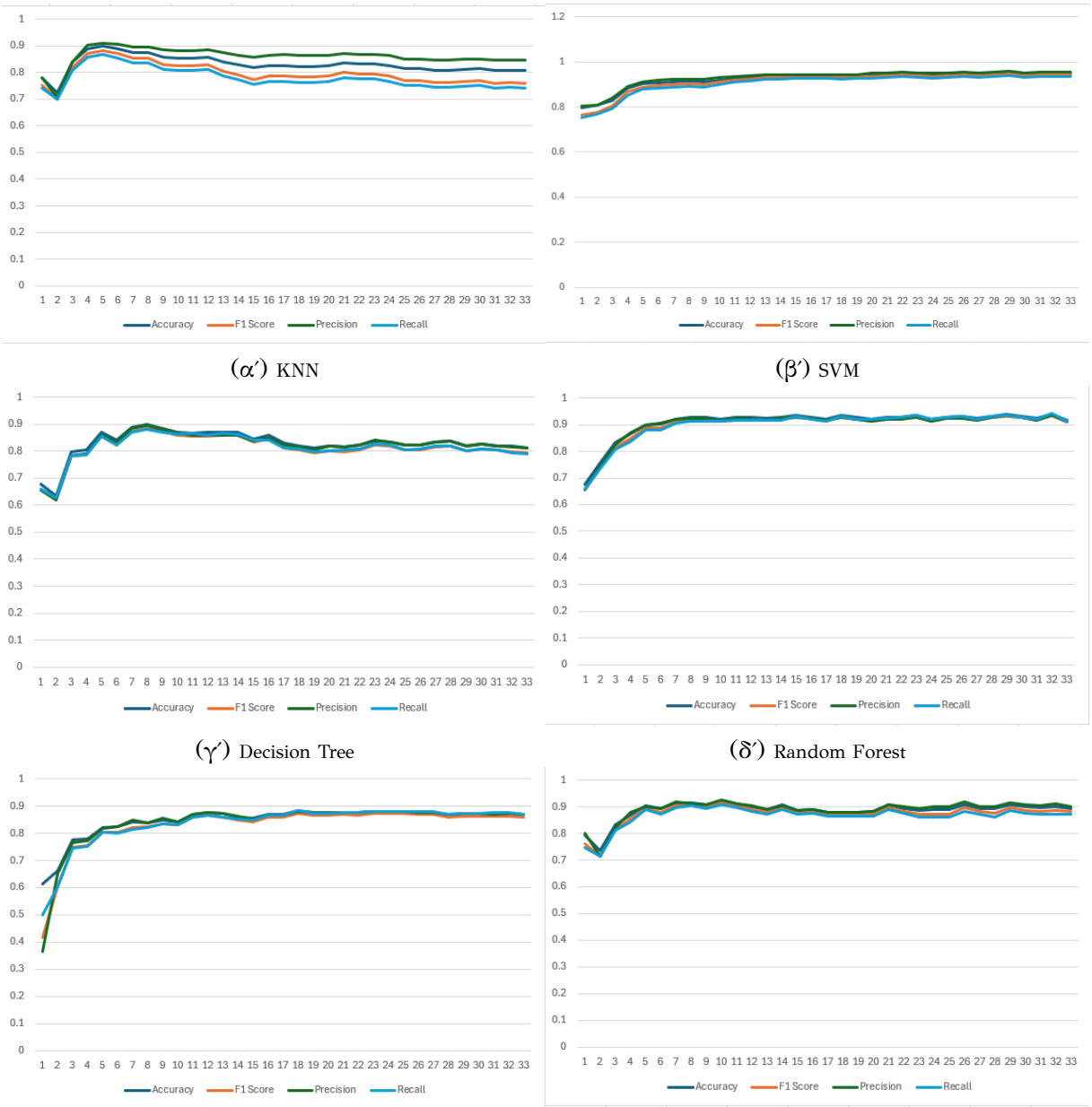
(ε′) Naive Bayes



(στ′) MLP

Σχήμα Α′.10: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο Kernel PCA

Α΄.2.4 Ionosphere Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.11: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο Kernel PCA

Α΄.2.5 ok Connectionist Bench Dataset

(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

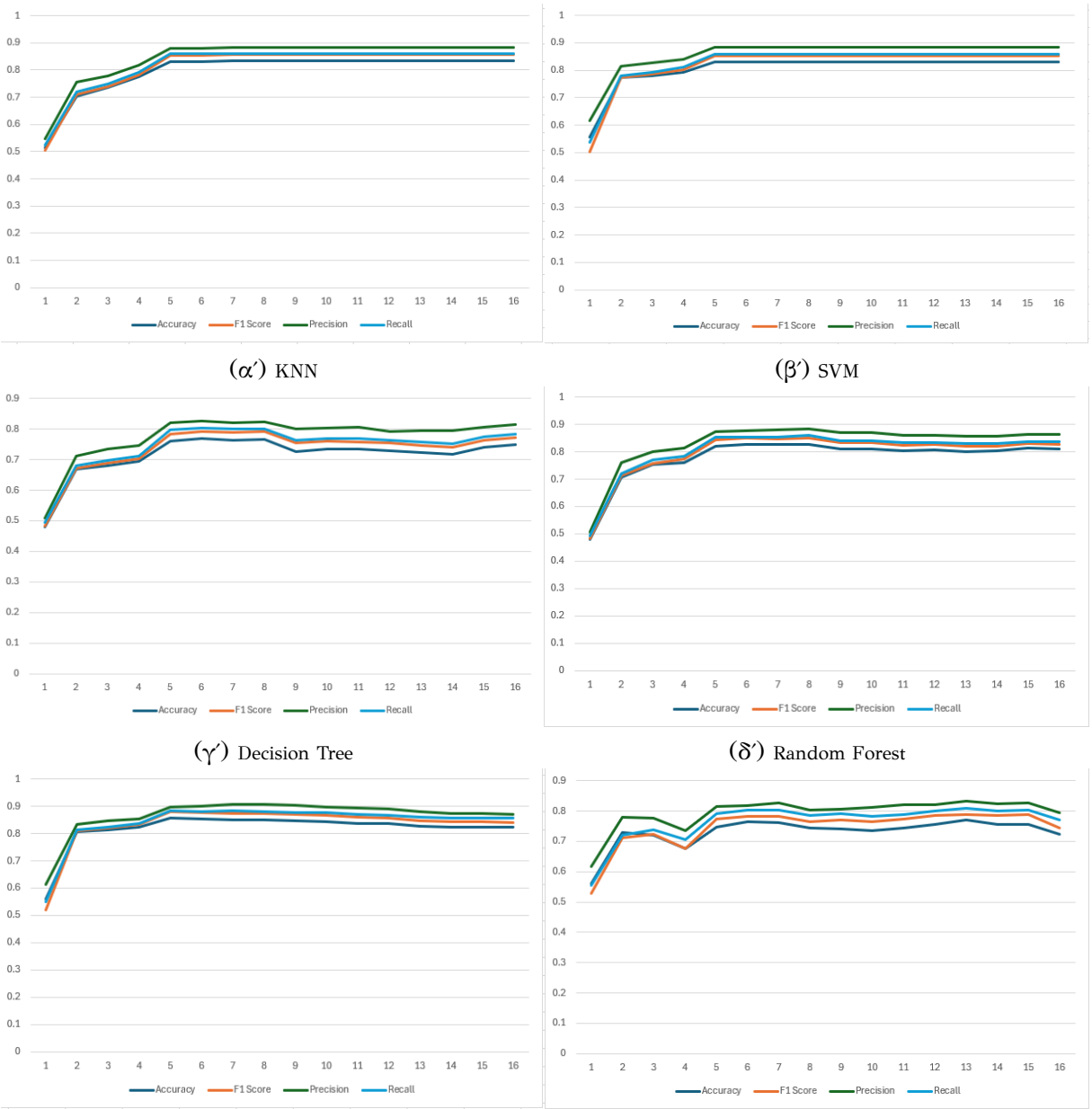
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.12: Ανάλυση διαστάσεων στο connectionist Bench Dataset με τη μέθοδο Kernel PCA

Α΄.2.6 Dry Bean Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

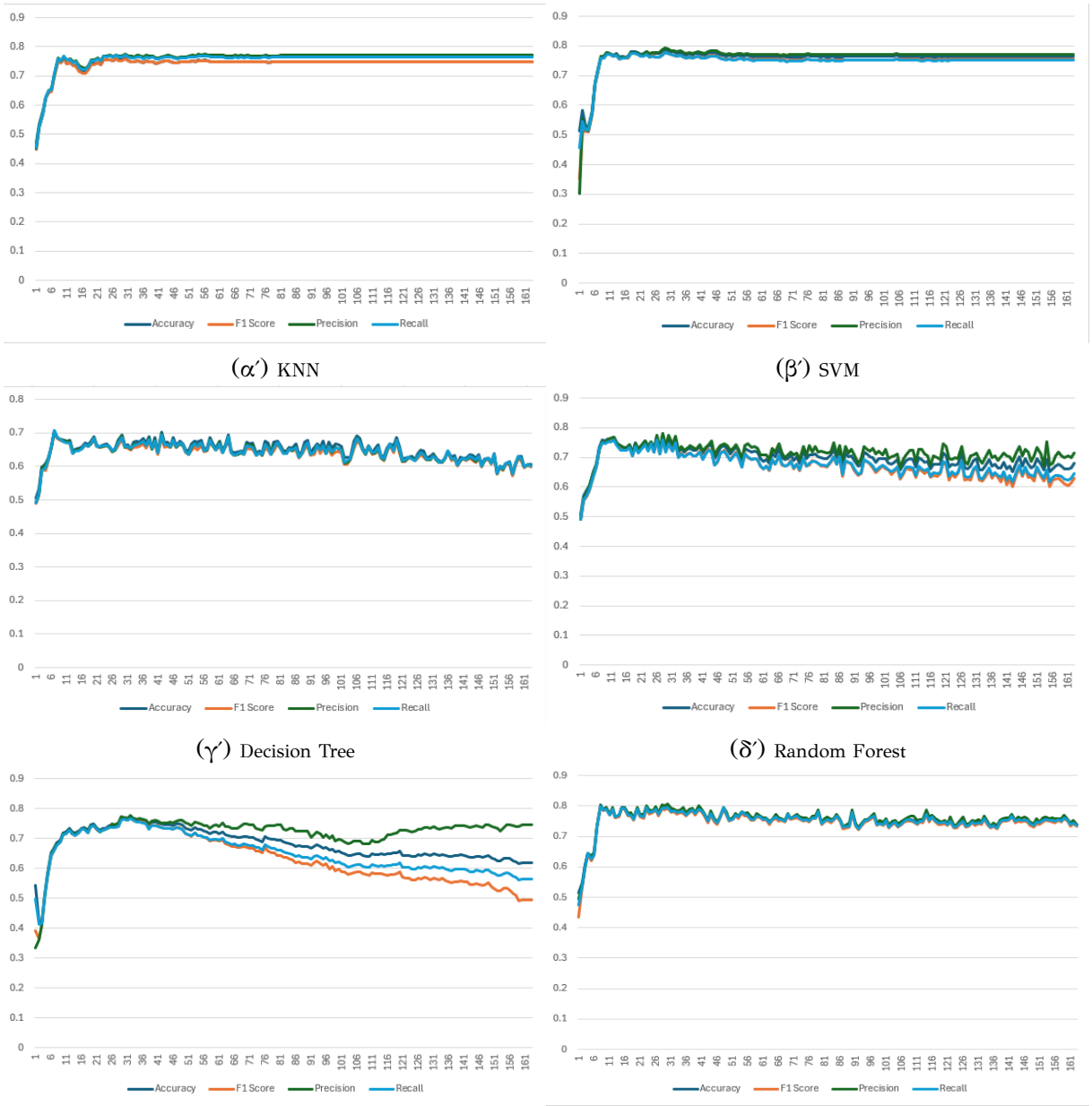
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.13: Ανάλυση διαστάσεων στο Dry Bean Dataset με τη μέθοδο Kernel PCA

Α΄.2.7 Musk Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

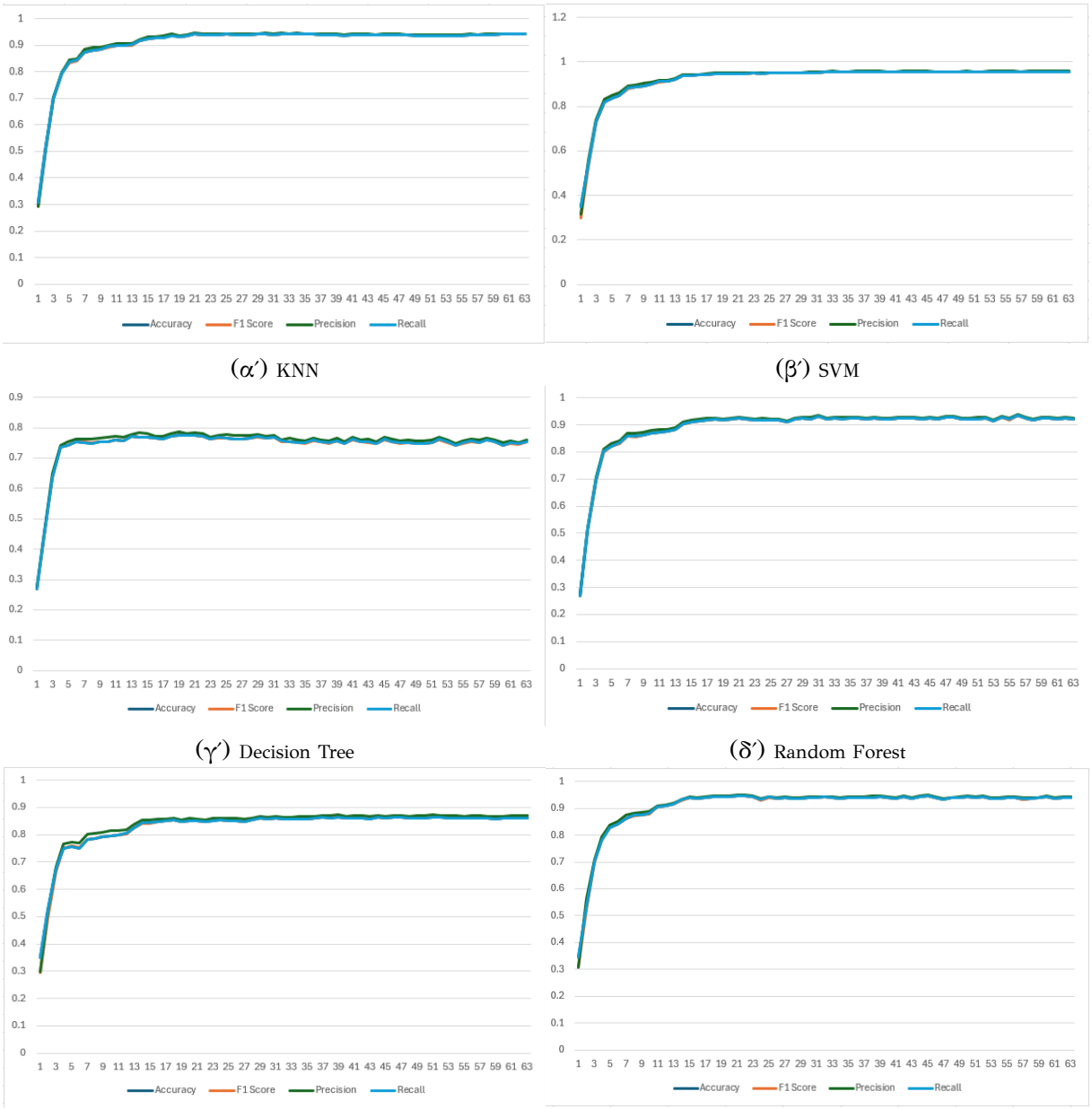
(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.14: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο Kernel PCA

Α΄.3 Αποτελέσματα SVD

Α΄.3.1 Digits Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

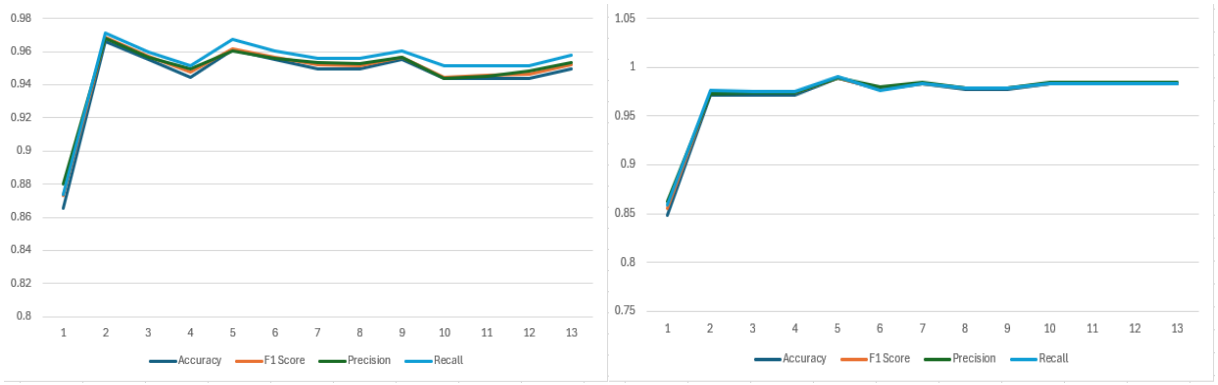
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

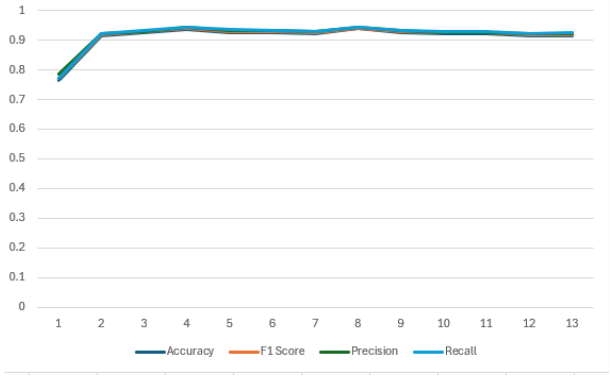
Σχήμα Α΄.15: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο SVD

Α΄.3.2 Wine Dataset

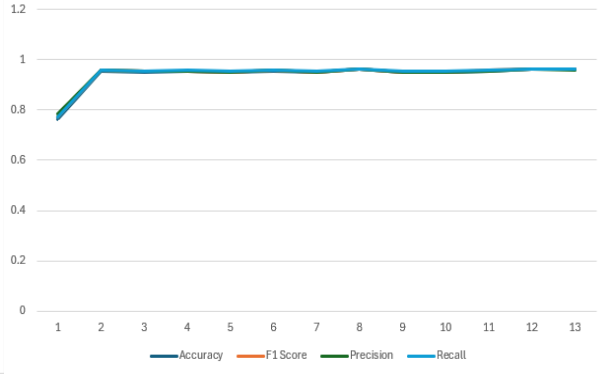


(α΄) KNN

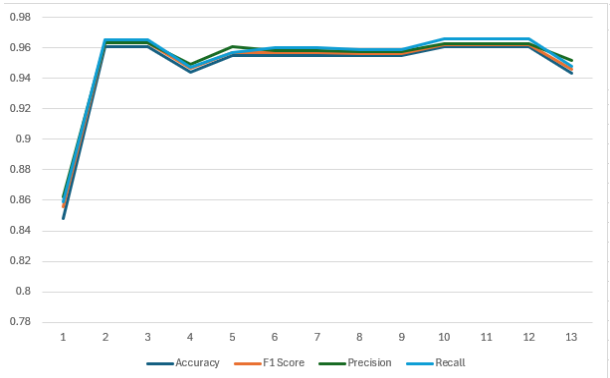
(β΄) SVM



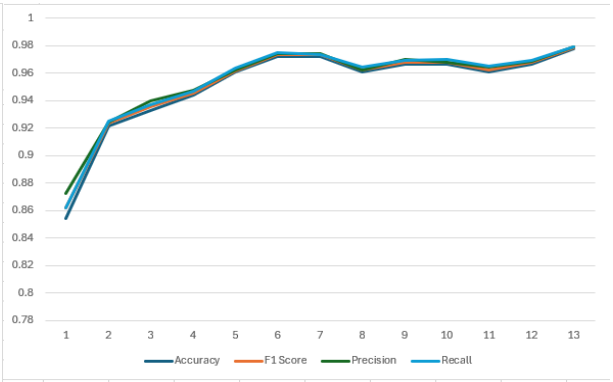
(γ΄) Decision Tree



(δ΄) Random Forest



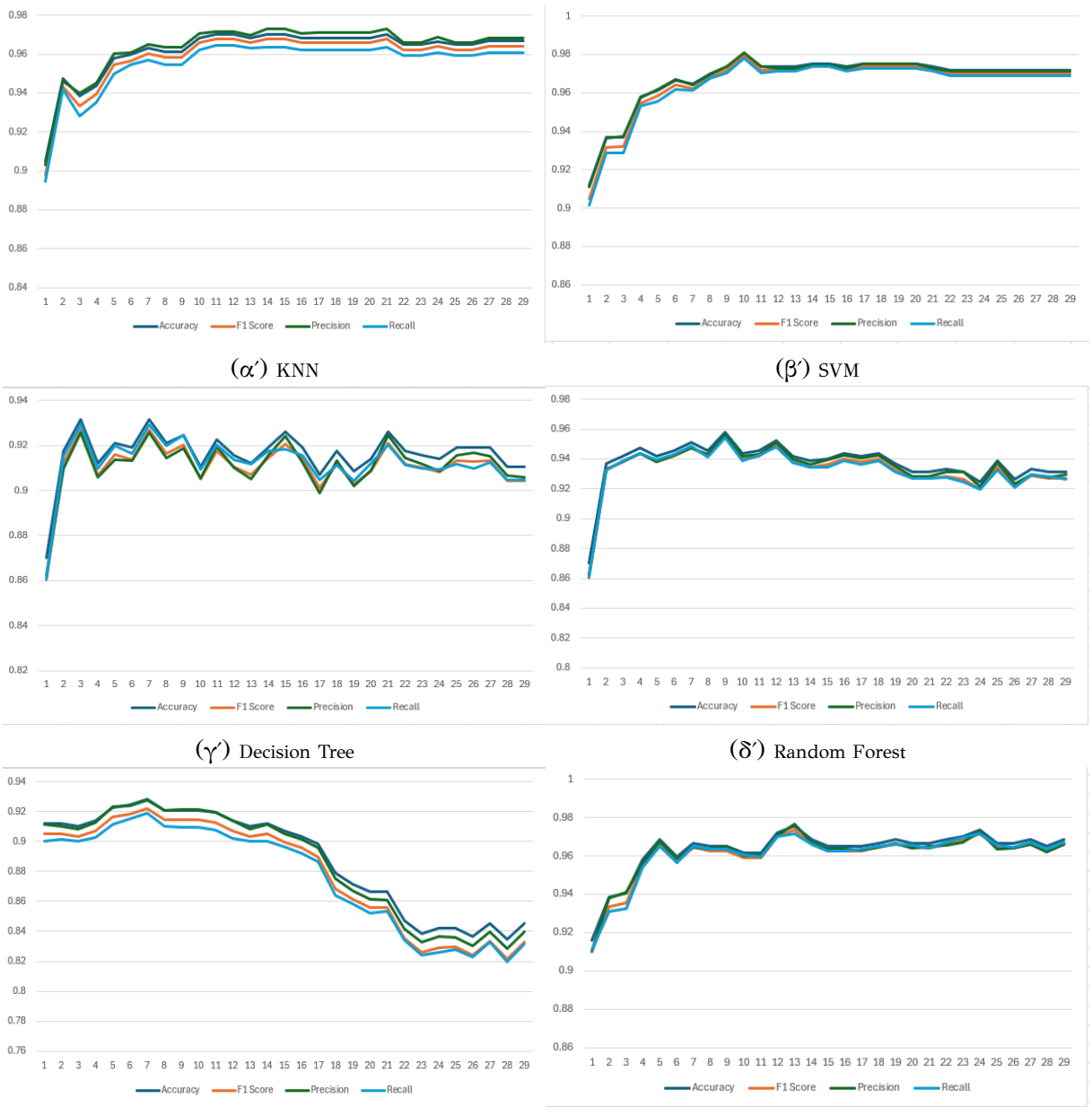
(ε΄) Naive Bayes



(στ΄) MLP

Σχήμα Α΄.16: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο SVD

Α΄.3.3 Breast Cancer Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

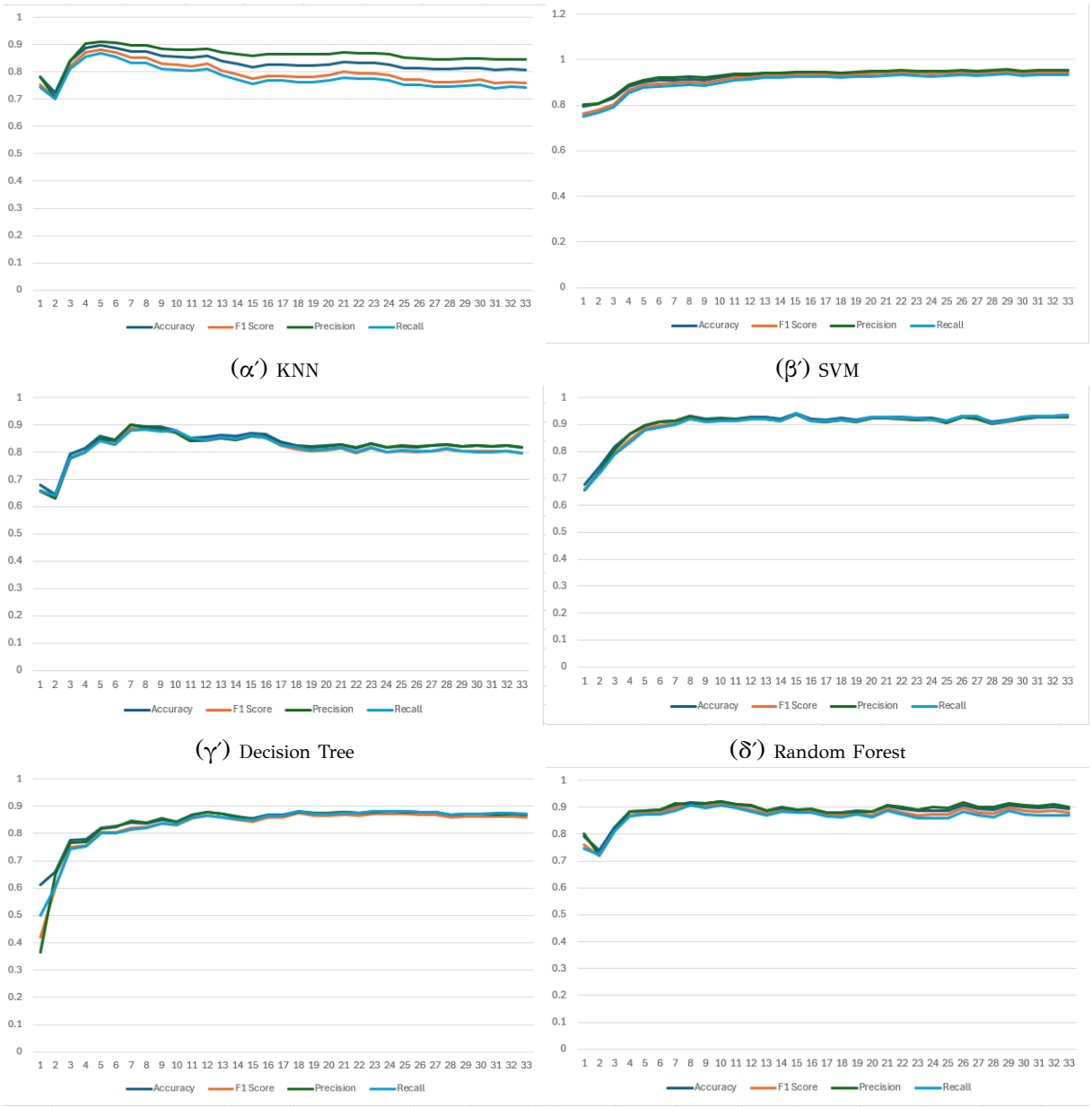
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.17: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο SVD

Α΄.3.4 Ionosphere Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

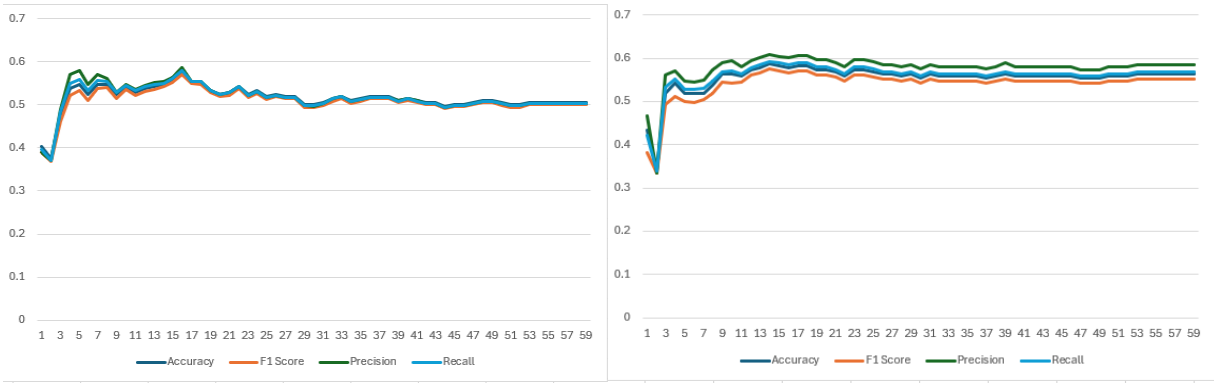
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

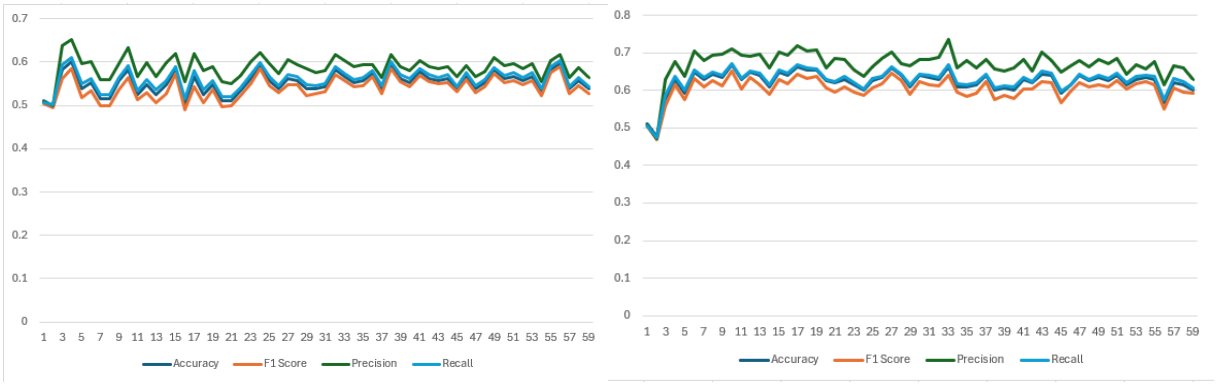
Σχήμα Α΄.18: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο SVD

Α΄.3.5 ok Connectionist Bench Dataset



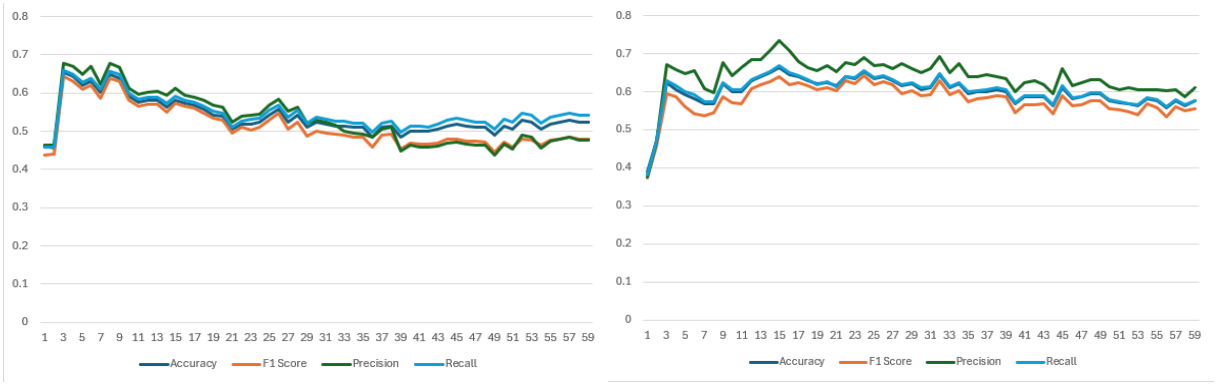
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest

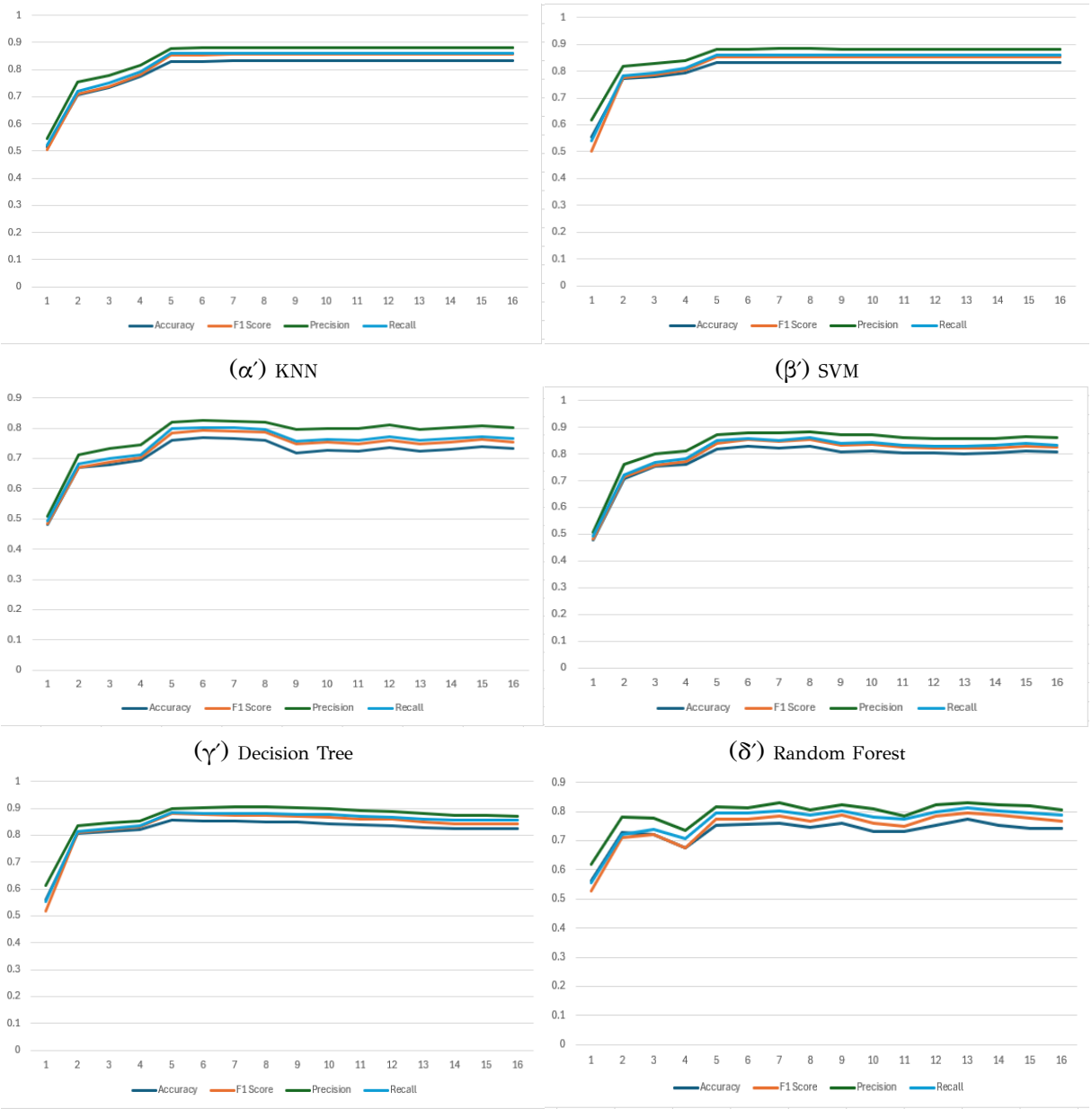


(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.19: Ανάλυση διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο SVD

Α΄.3.6 Dry Bean Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

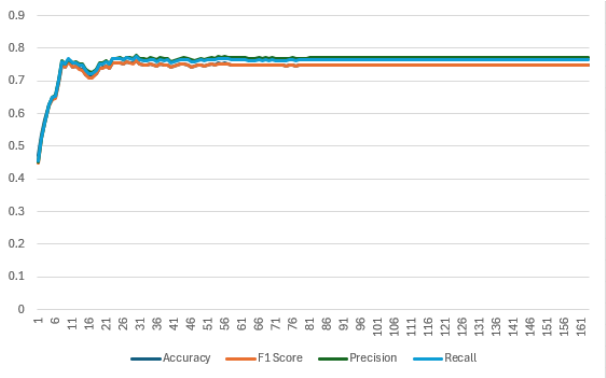
(δ΄) Random Forest

(ε΄) Naive Bayes

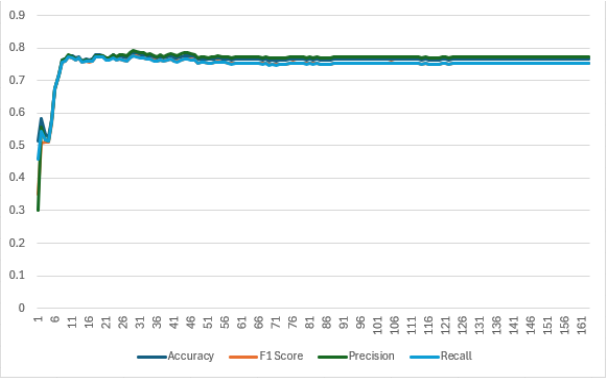
(στ΄) MLP

Σχήμα Α΄.20: Ανάλυση διαστάσεων στο Dry Bean Dataset με τη μέθοδο SVD

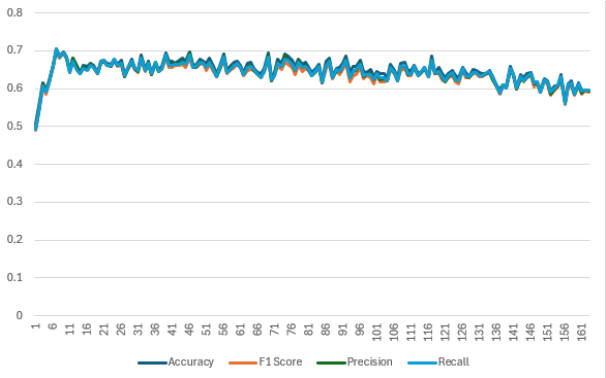
Α΄.3.7 Musk Dataset



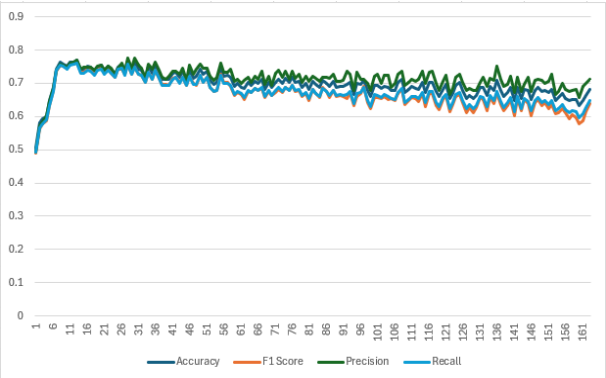
(α΄) KNN



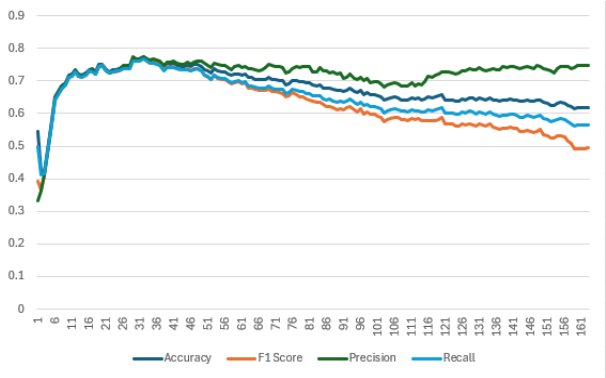
(β΄) SVM



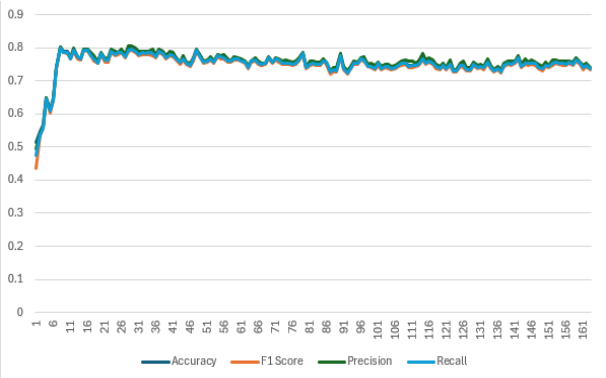
(γ΄) Decision Tree



(δ΄) Random Forest



(ε΄) Naive Bayes

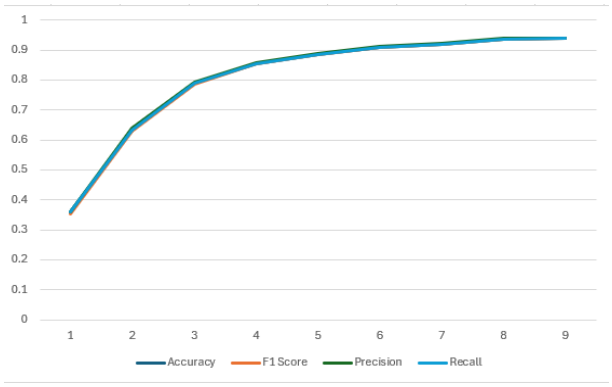


(στ΄) MLP

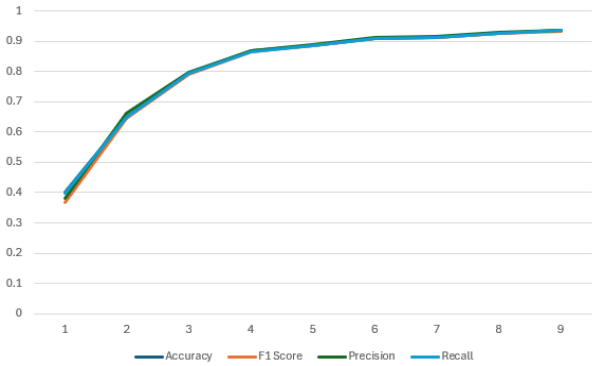
Σχήμα Α΄.21: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο SVD

Α΄.4 Αποτελέσματα LDA

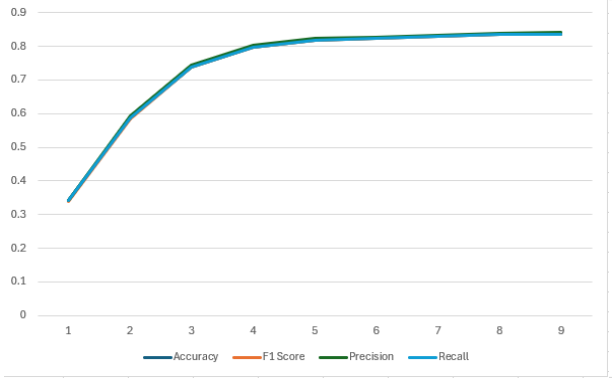
Α΄.4.1 Digits Dataset



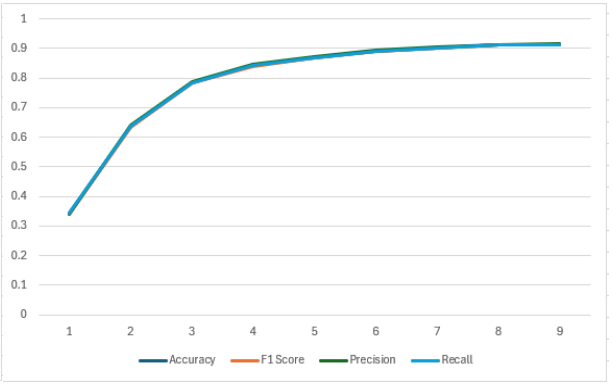
(α΄) KNN



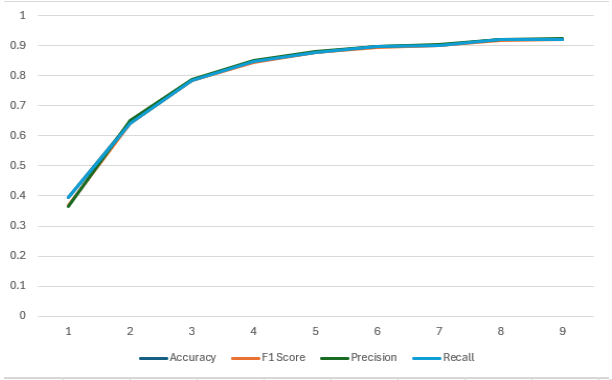
(β΄) SVM



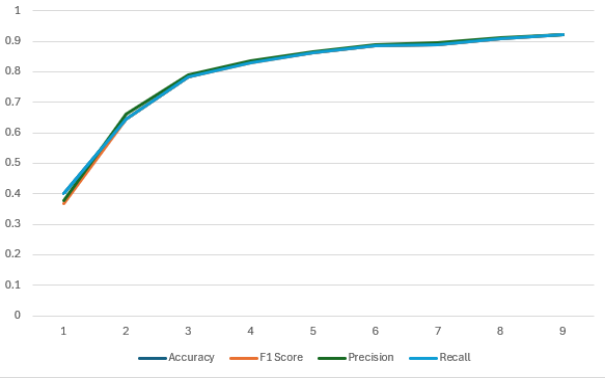
(γ΄) Decision Tree



(δ΄) Random Forest



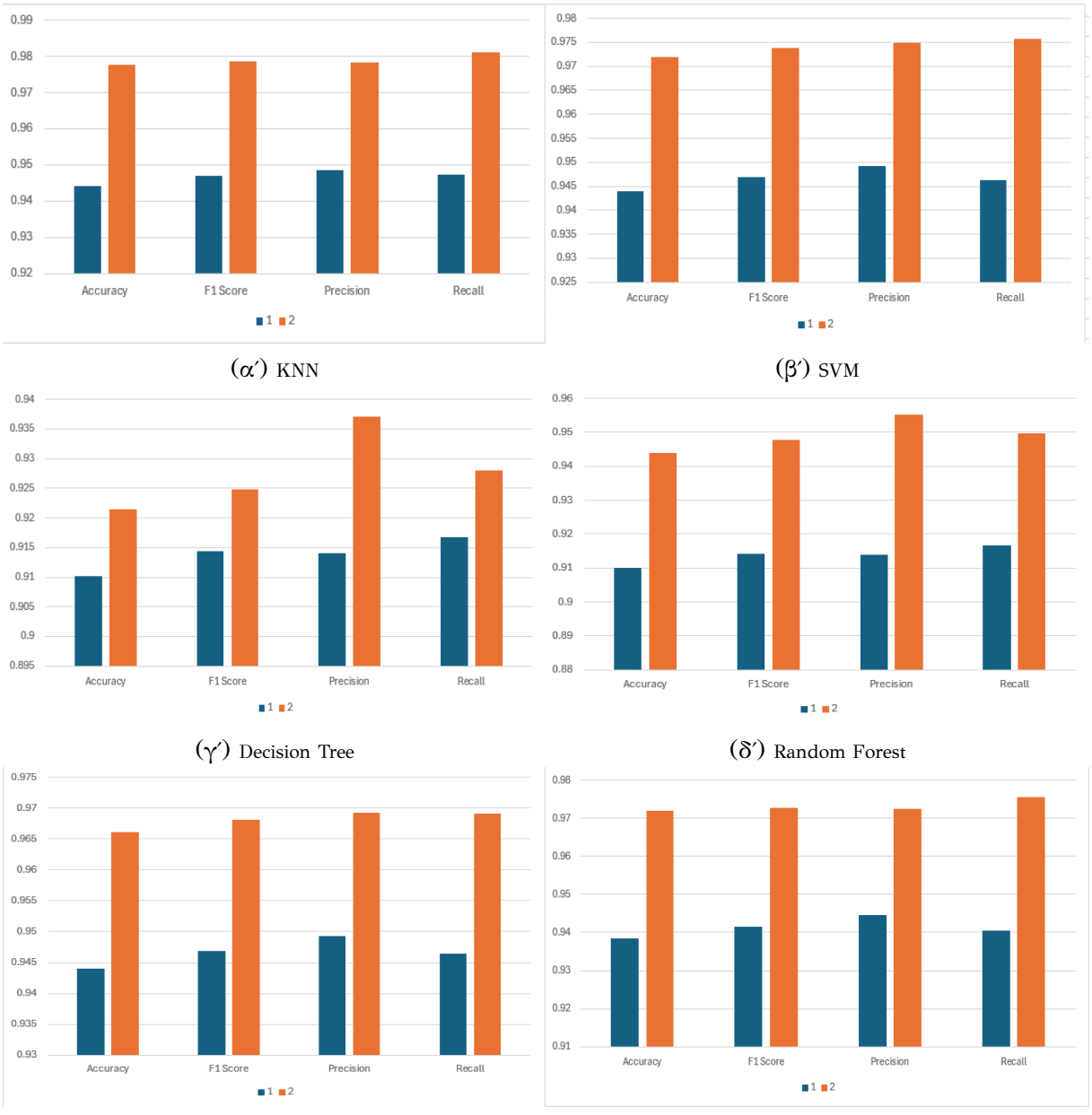
(ε΄) Naive Bayes



(στ΄) MLP

Σχήμα Α΄.22: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο LDA

Α΄.4.2 Wine Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

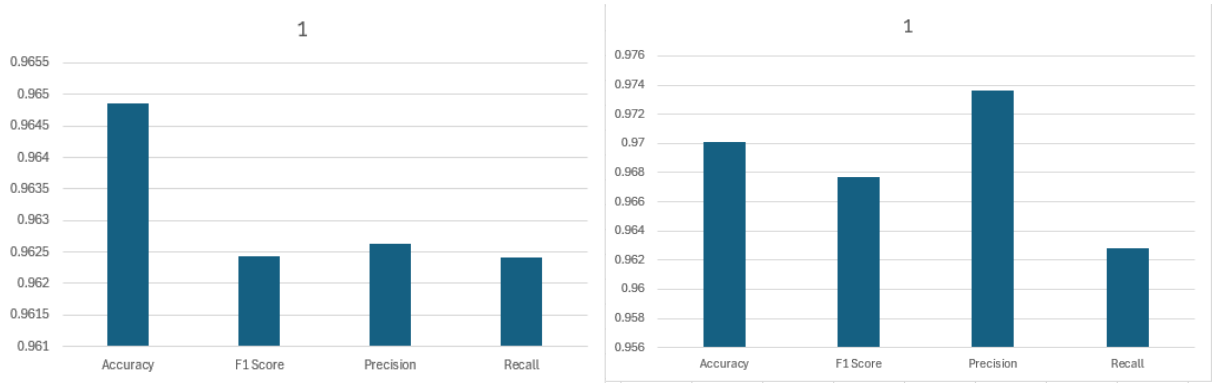
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

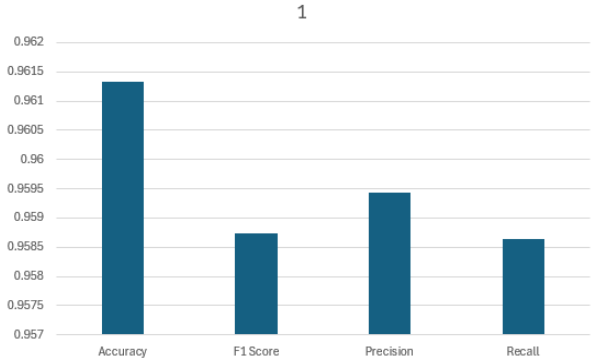
Σχήμα Α΄.23: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο LDA

Α΄.4.3 Breast Cancer Dataset

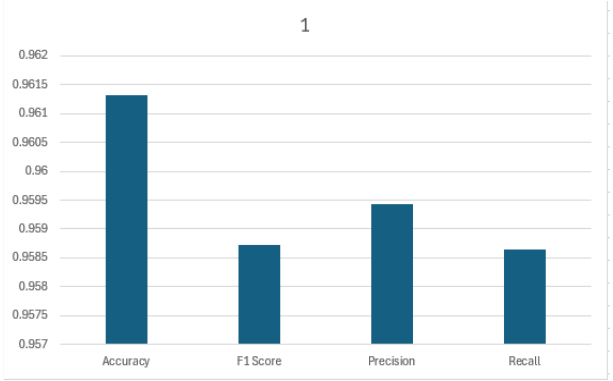


(α΄) KNN

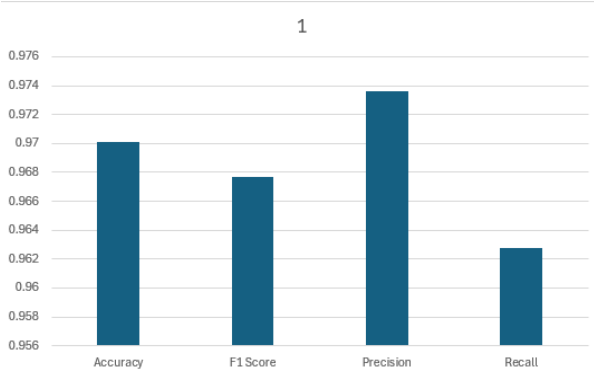
(β΄) SVM



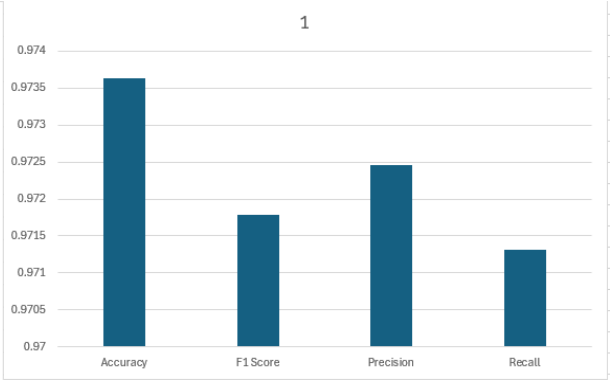
(γ΄) Decision Tree



(δ΄) Random Forest



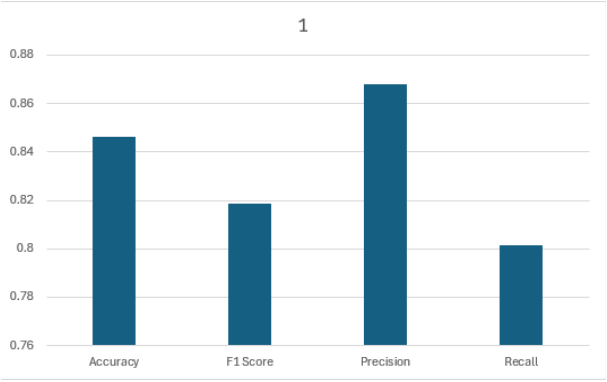
(ε΄) Naive Bayes



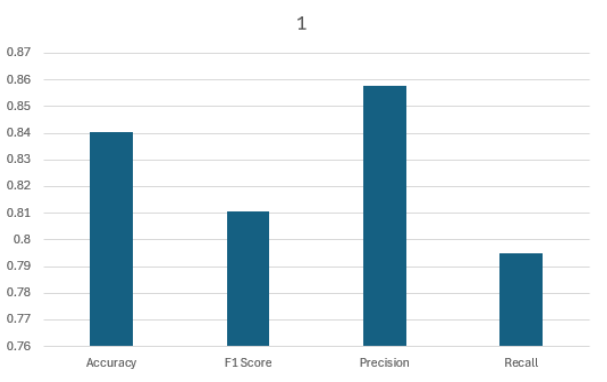
(στ΄) MLP

Σχήμα Α΄.24: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο LDA

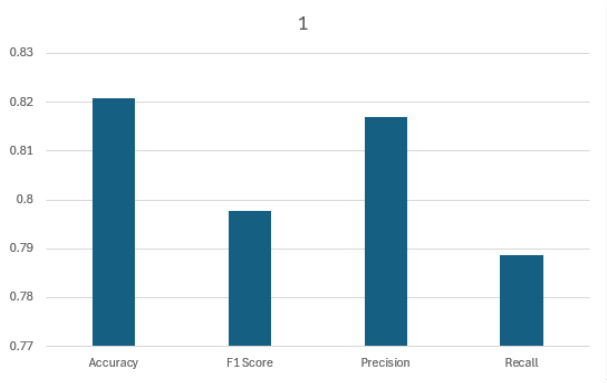
Α΄.4.4 Ionosphere Dataset



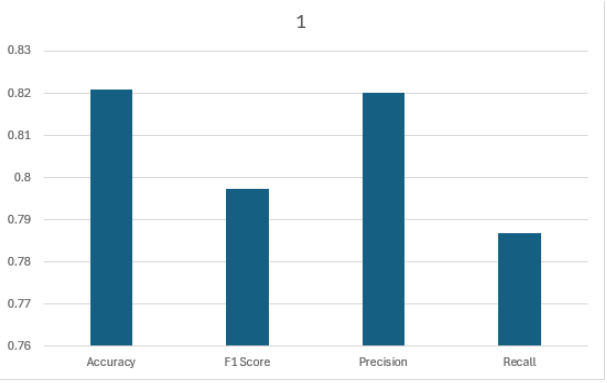
(α΄) KNN



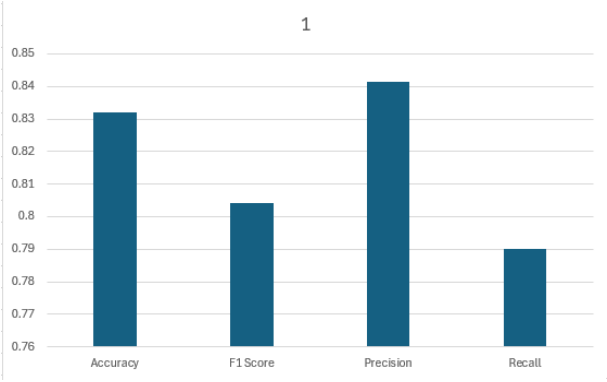
(β΄) SVM



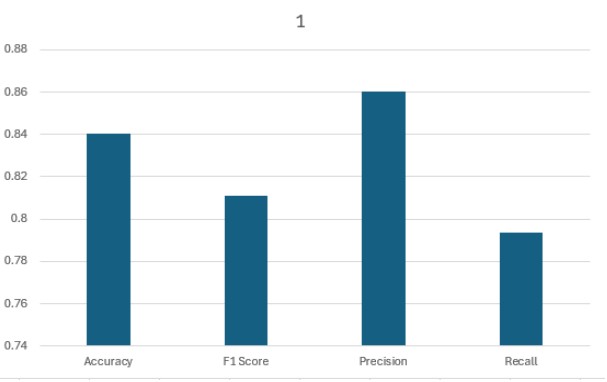
(γ΄) Decision Tree



(δ΄) Random Forest



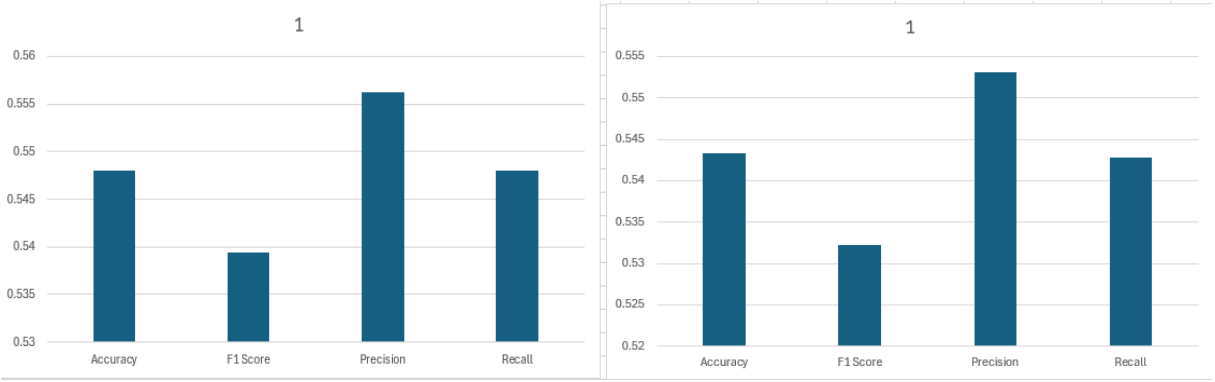
(ε΄) Naive Bayes



(στ΄) MLP

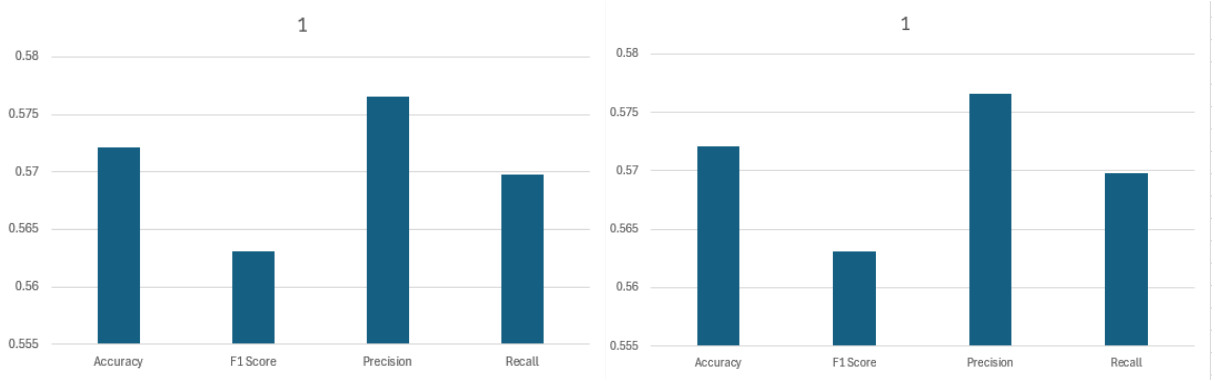
Σχήμα Α΄.25: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο LDA

Α΄.4.5 Connectionist Bench Dataset



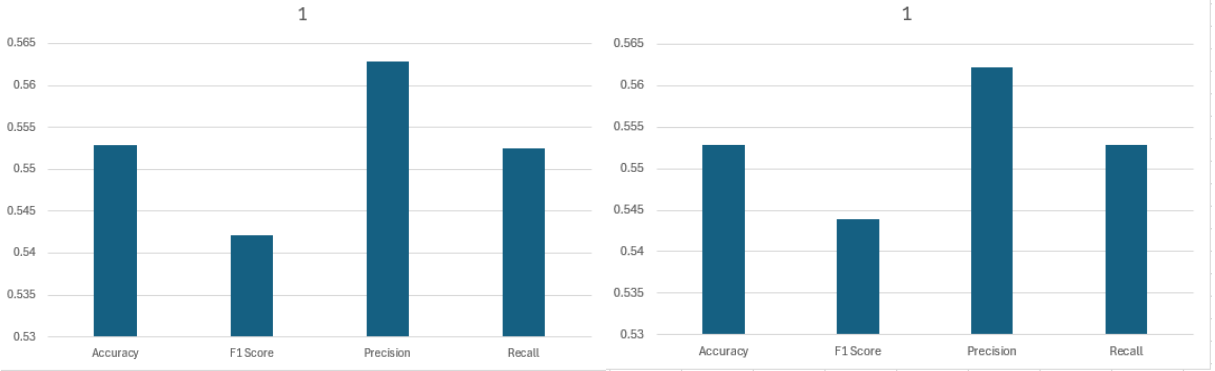
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest

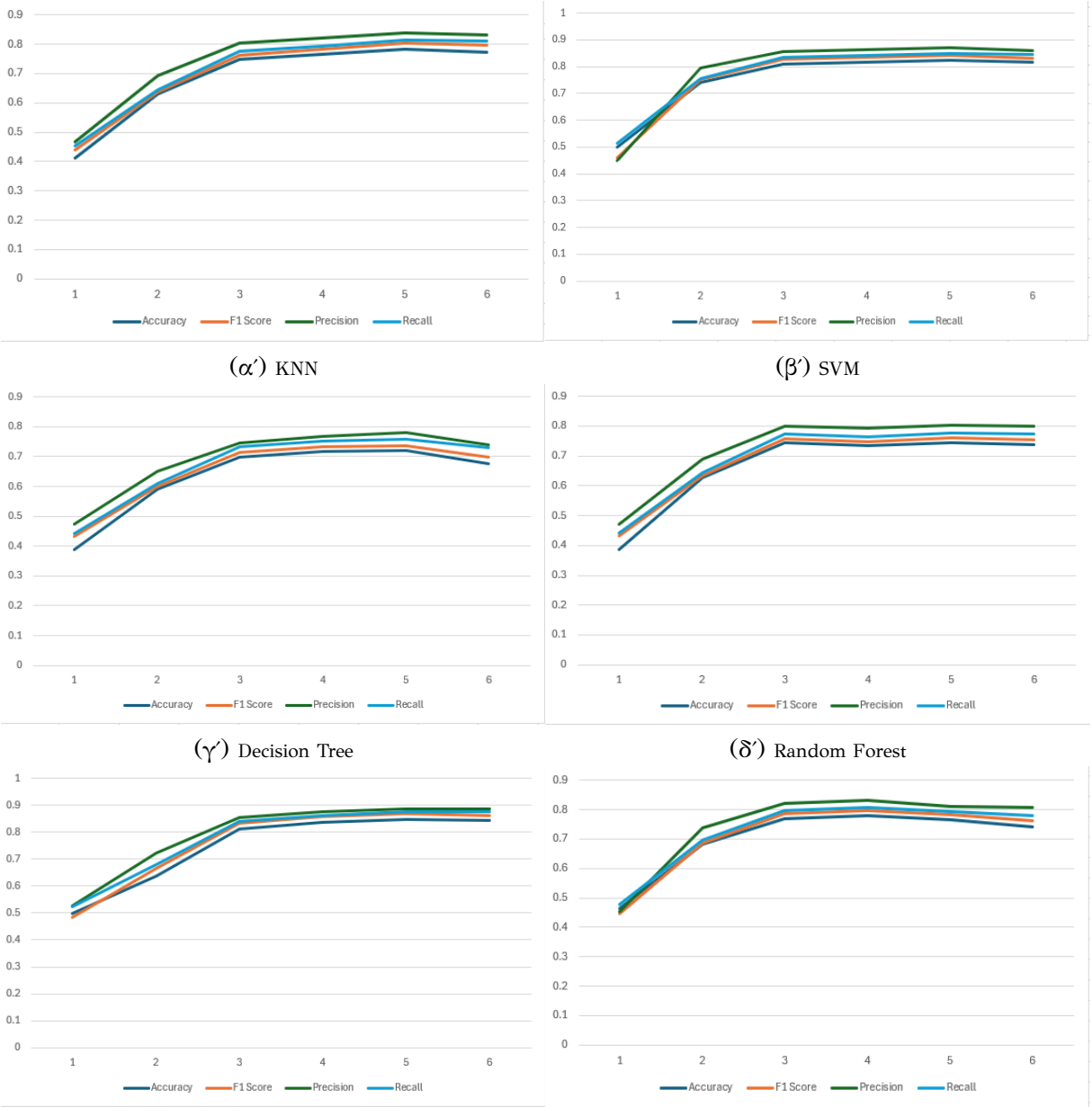


(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.26: Ανάλυση διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο LDA

Α΄.4.6 Dry Bean Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

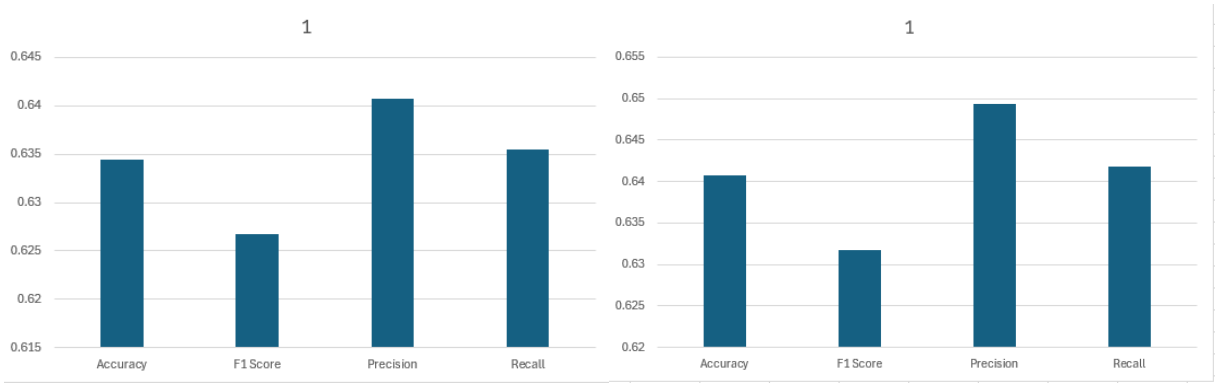
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

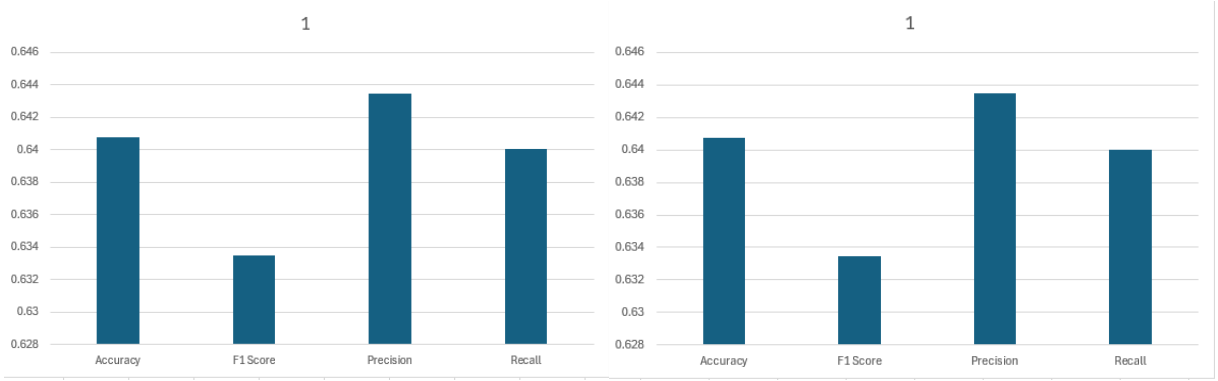
Σχήμα Α΄.27: Ανάλυση διαστάσεων στο Dry Bean Dataset με τη μέθοδο LDA

Α΄.4.7 Musk Dataset



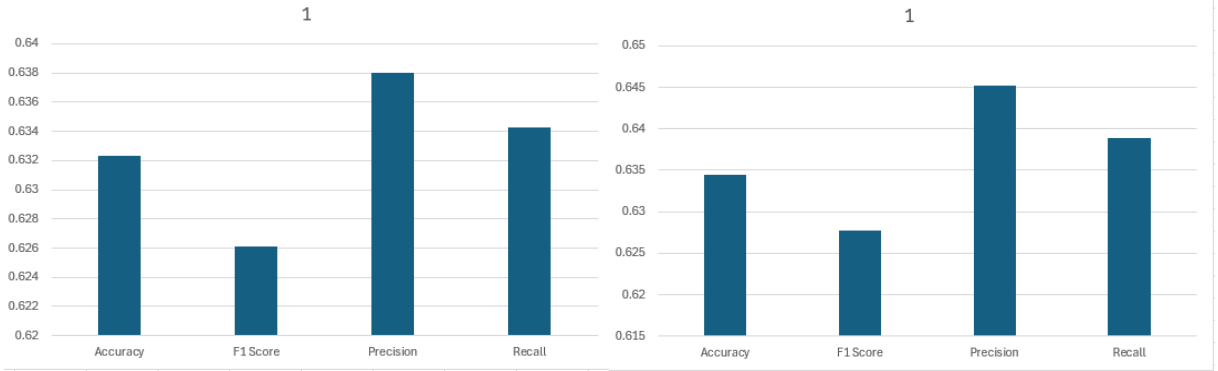
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest



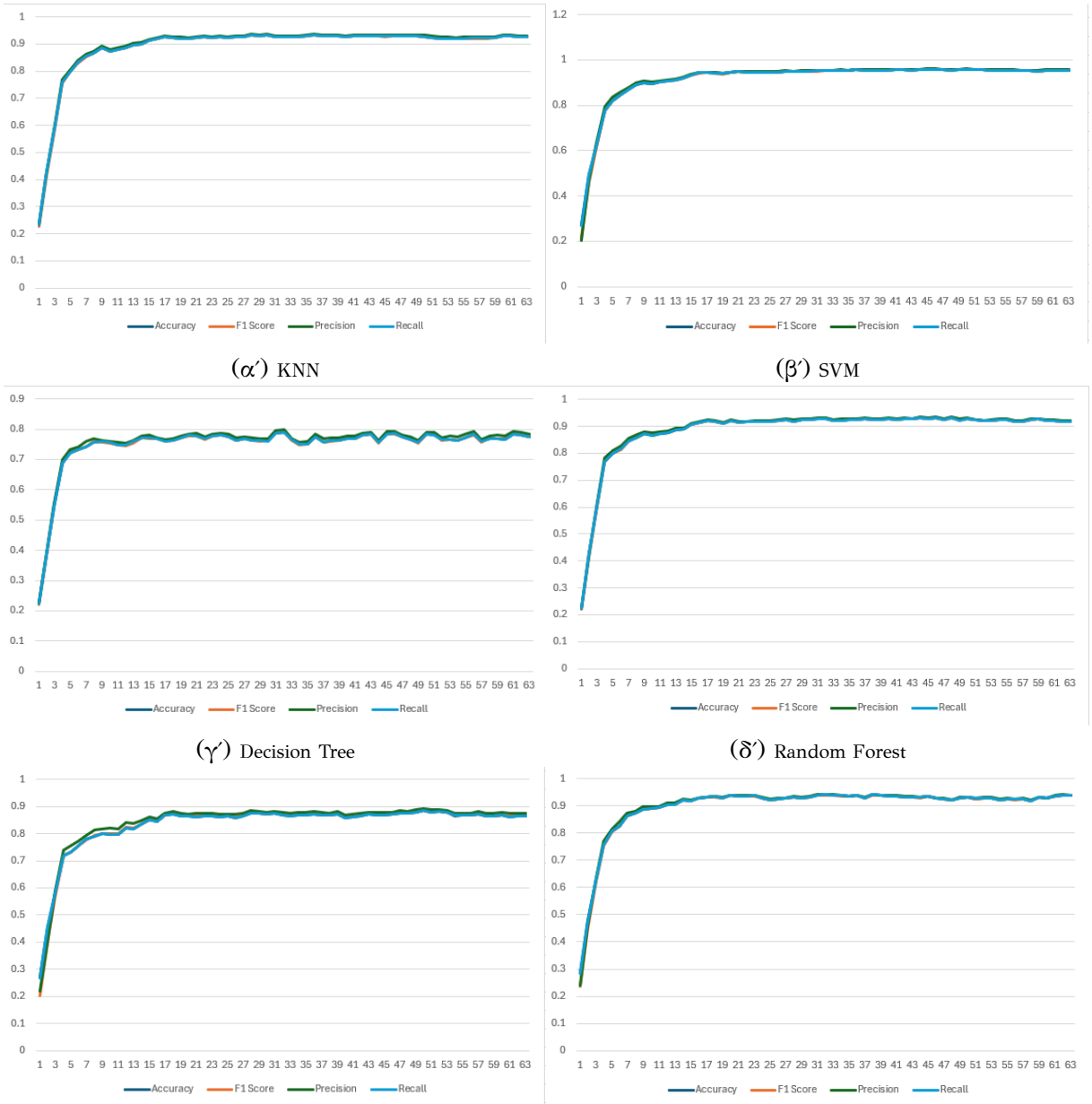
(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.28: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο LDA

Α΄.5 Αποτελέσματα Factor Analysis

Α΄.5.1 Digits Dataset



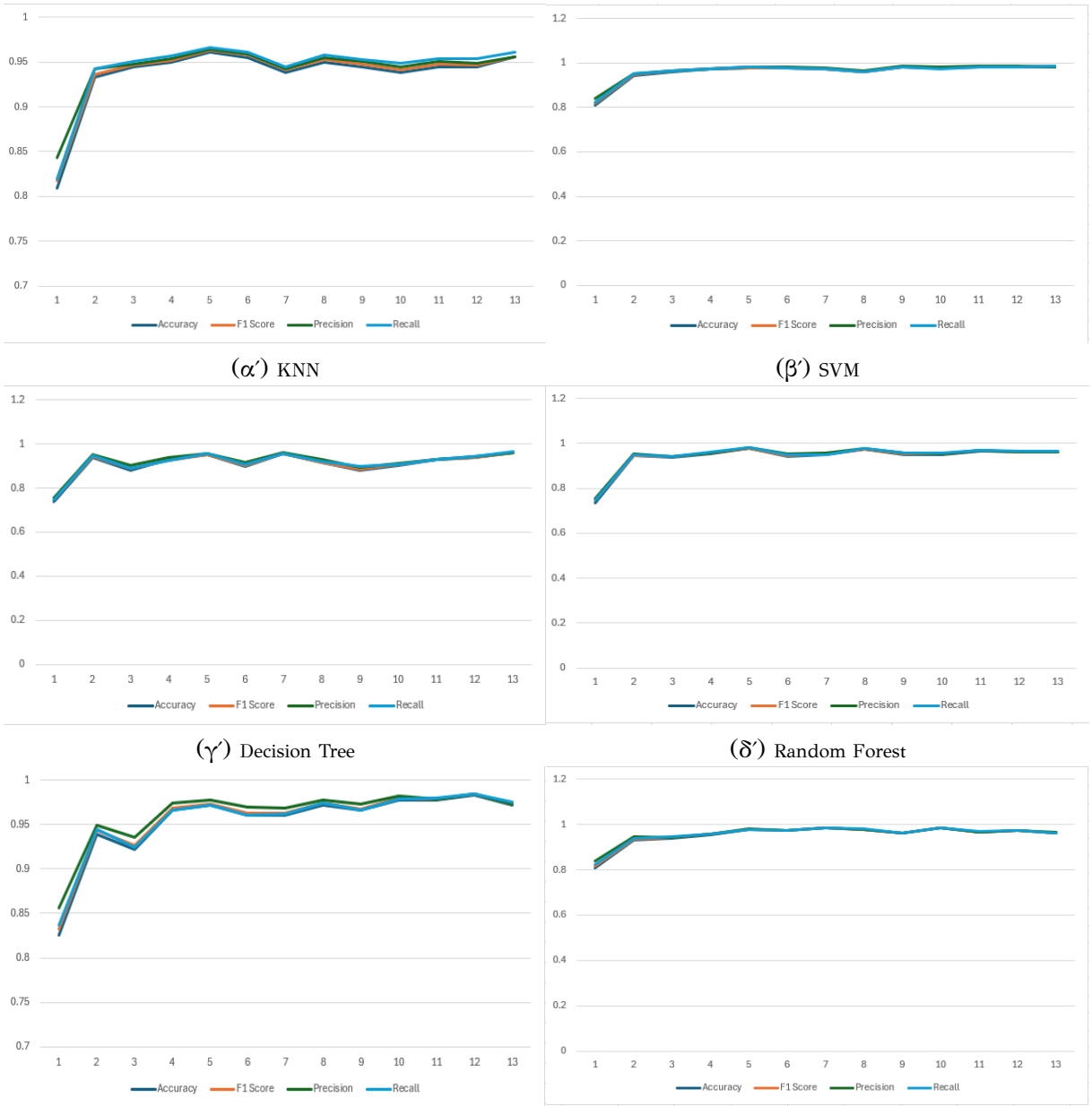
(α΄) KNN (β΄) SVM

(γ΄) Decision Tree (δ΄) Random Forest

(ε΄) Naive Bayes (στ΄) MLP

Σχήμα Α΄.29: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο Factor Analysis

Α΄.5.2 Wine Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

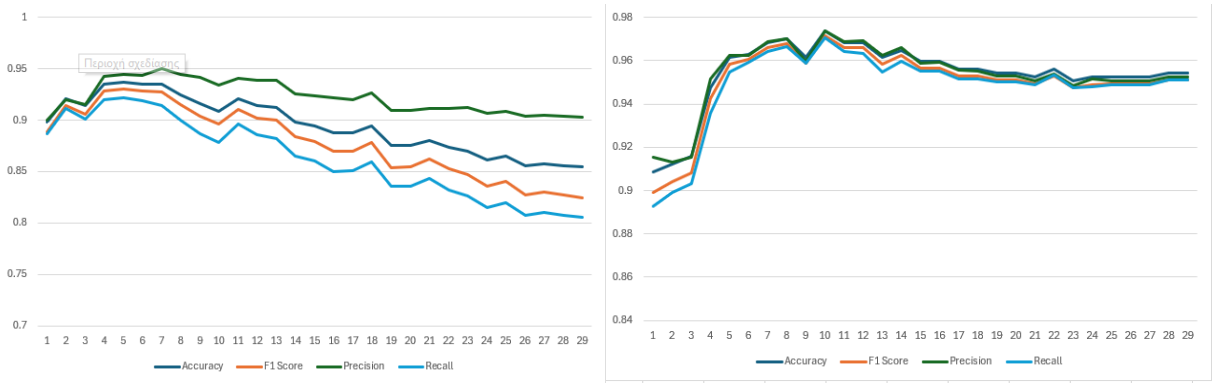
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

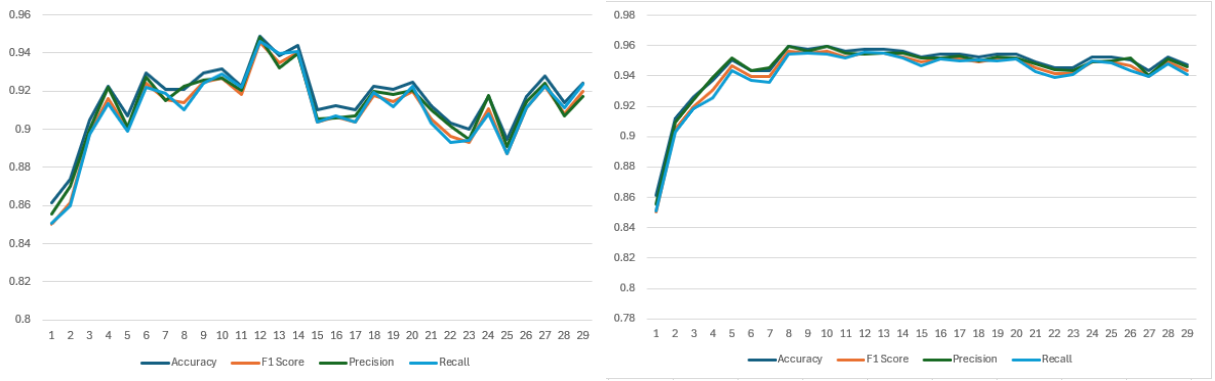
Σχήμα Α΄.30: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο Factor Analysis

Α΄.5.3 Breast Cancer Dataset



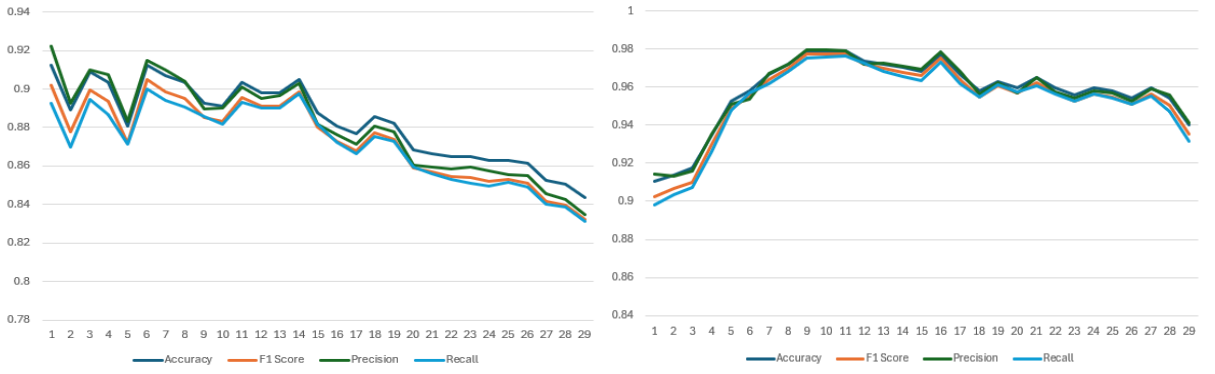
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest

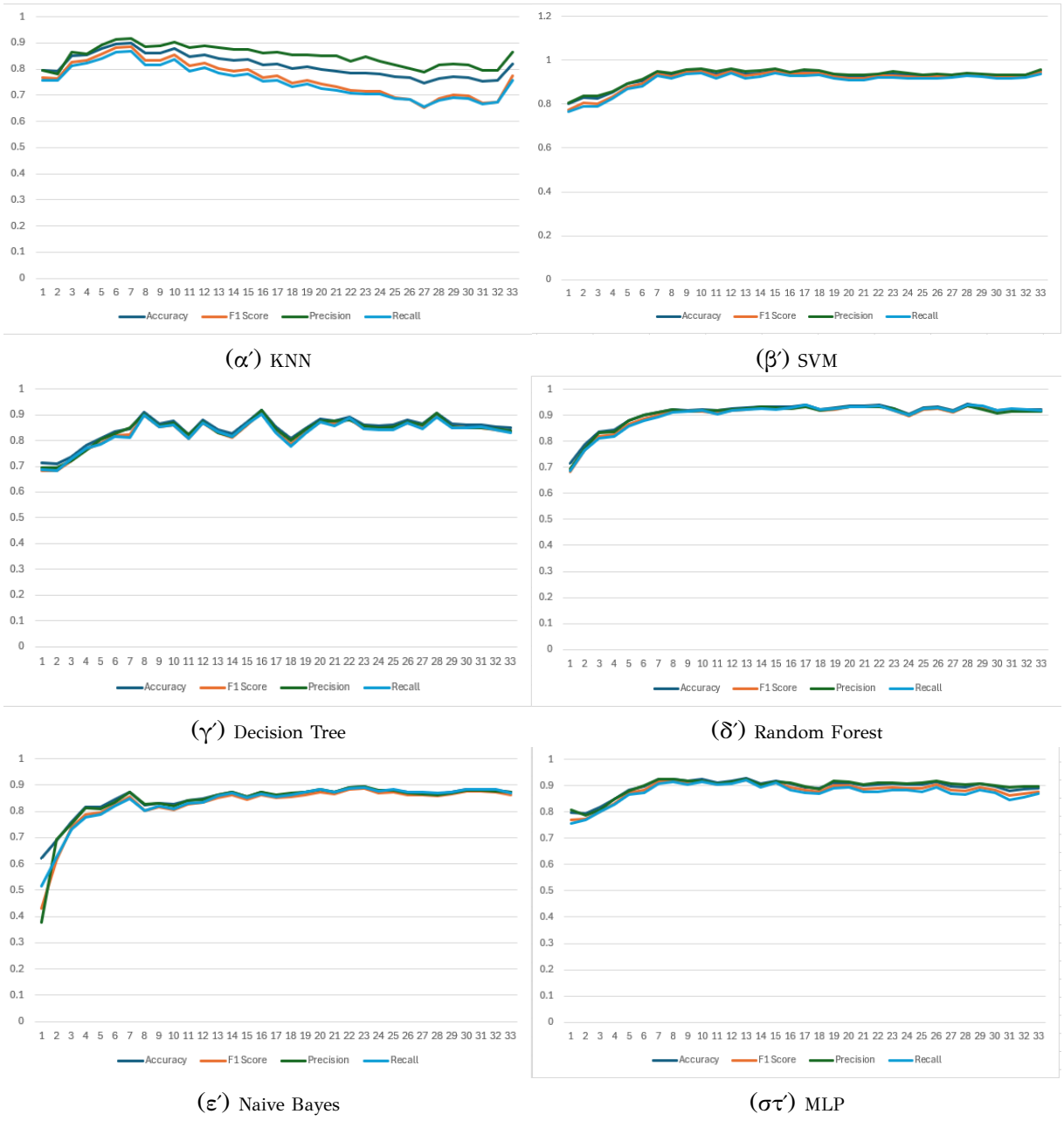


(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.31: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο Factor Analysis

Α΄.5.4 Ionosphere Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

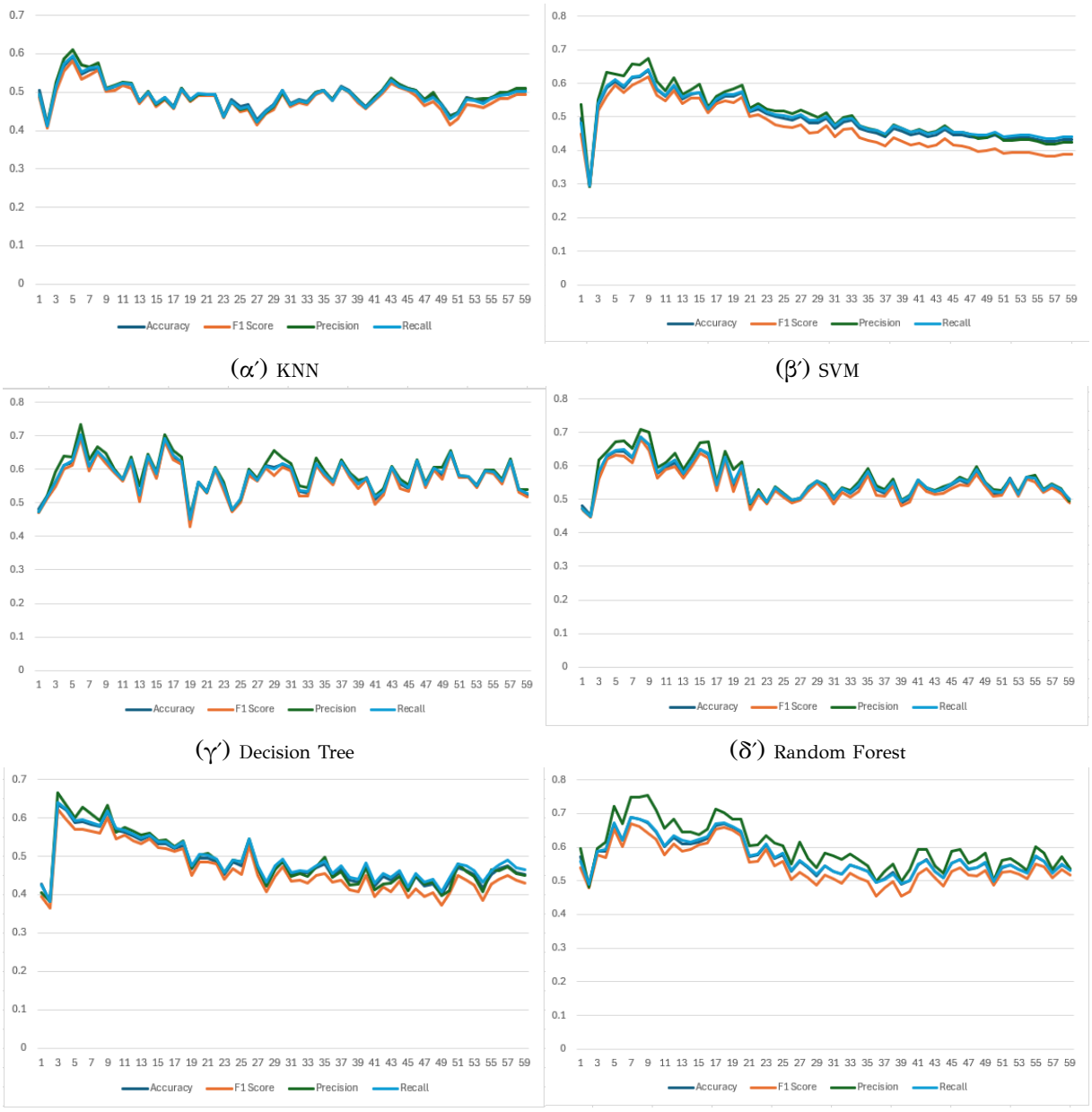
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.32: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο Factor Analysis

Α΄.5.5 ok Connectionist Bench Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

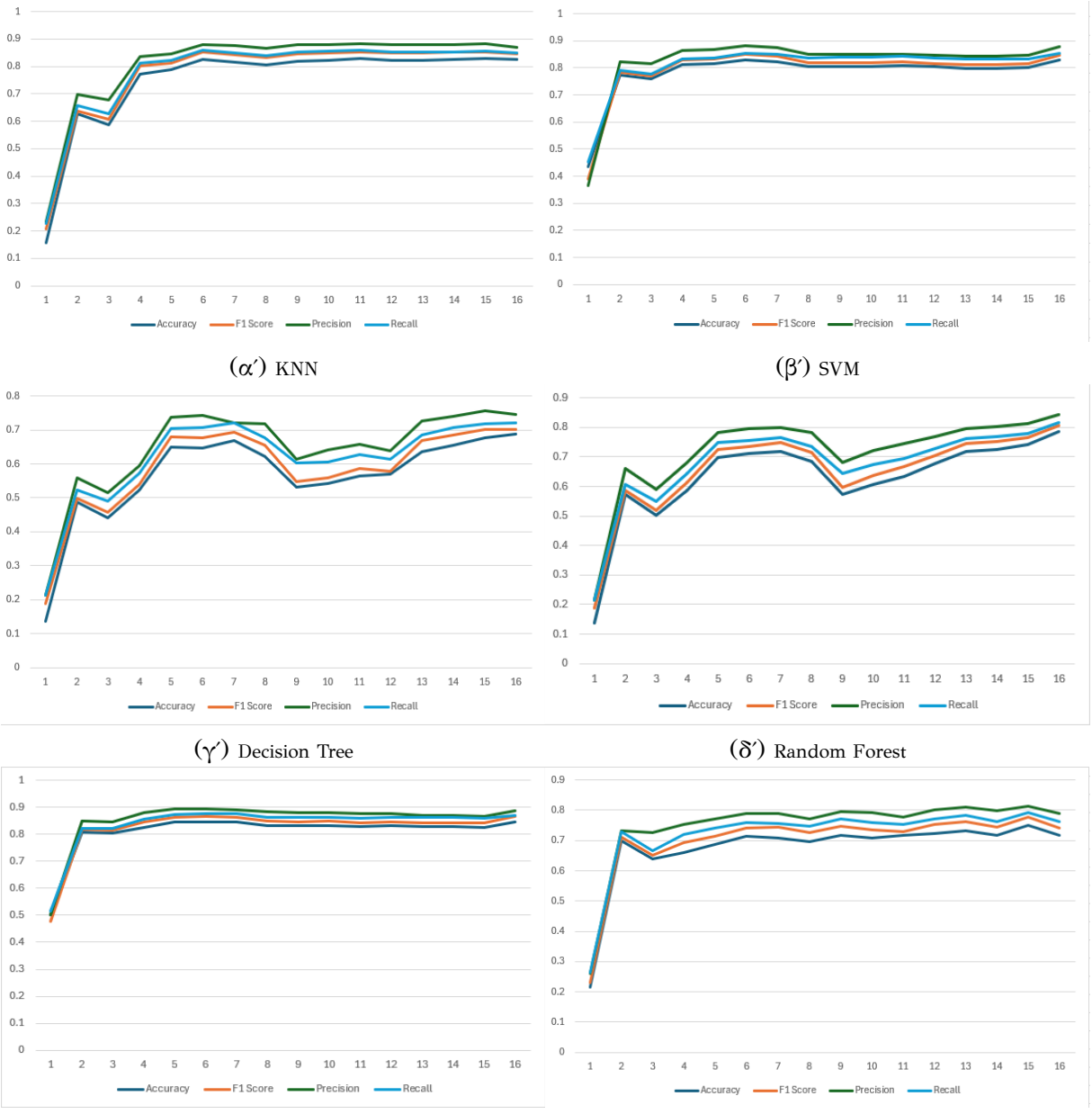
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.33: Ανάλυση διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο Factor Analysis

Α΄.5.6 Dry Bean Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

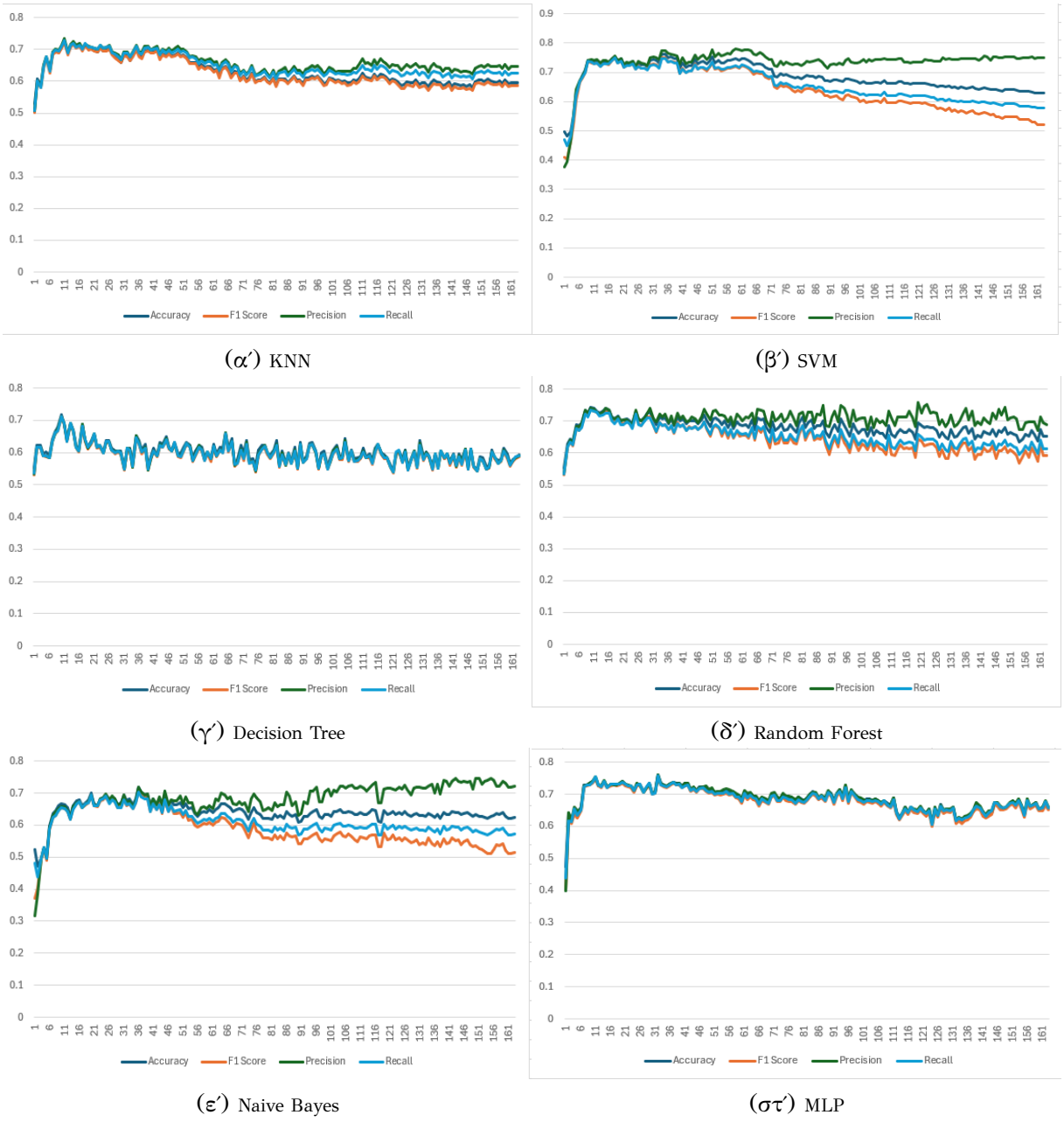
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.34: Ανάλυση διαστάσεων στο Dry Bean Dataset με τη μέθοδο Factor Analysis

Α΄.5.7 Musk Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

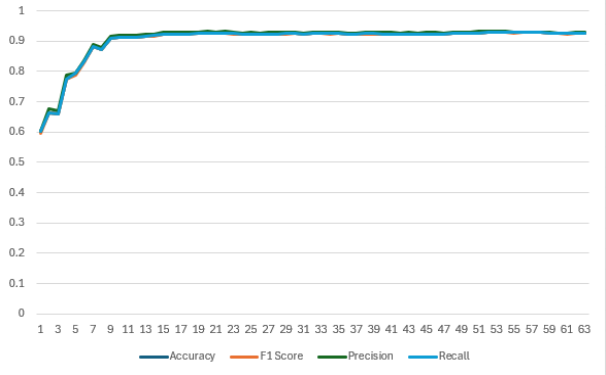
(ε΄) Naive Bayes

(στ΄) MLP

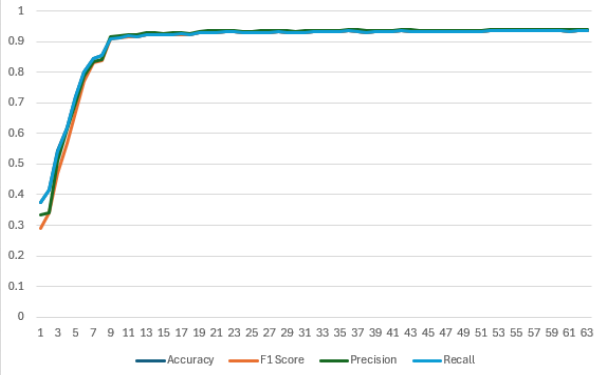
Σχήμα Α΄.35: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο Factor Analysis

Α΄.6 Αποτελέσματα LLE

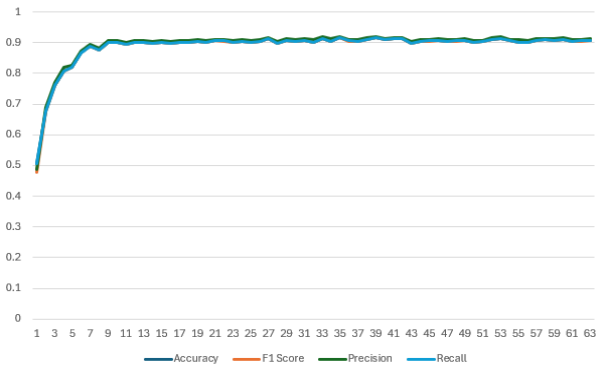
Α΄.6.1 Digits Dataset



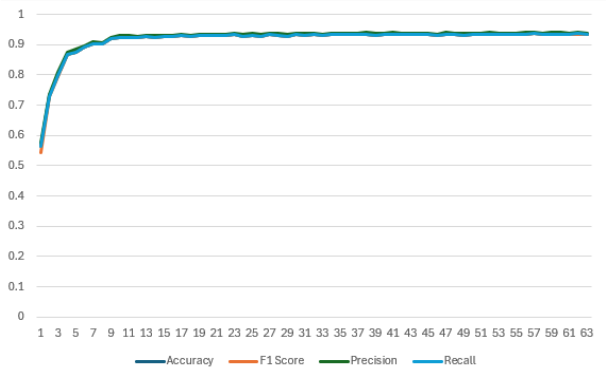
(α΄) KNN



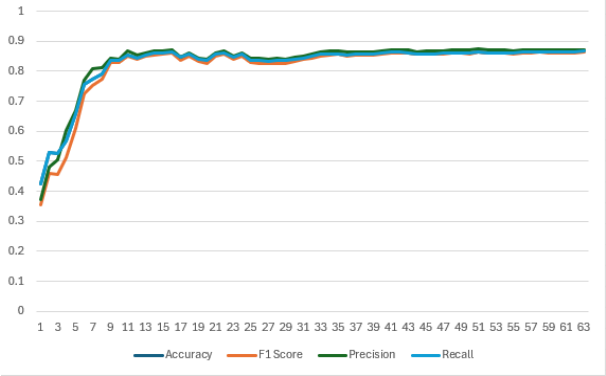
(β΄) SVM



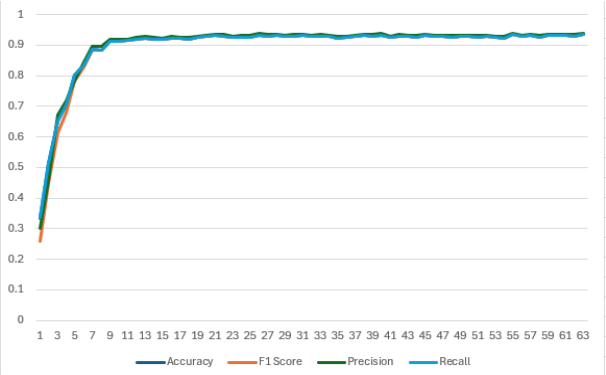
(γ΄) Decision Tree



(δ΄) Random Forest



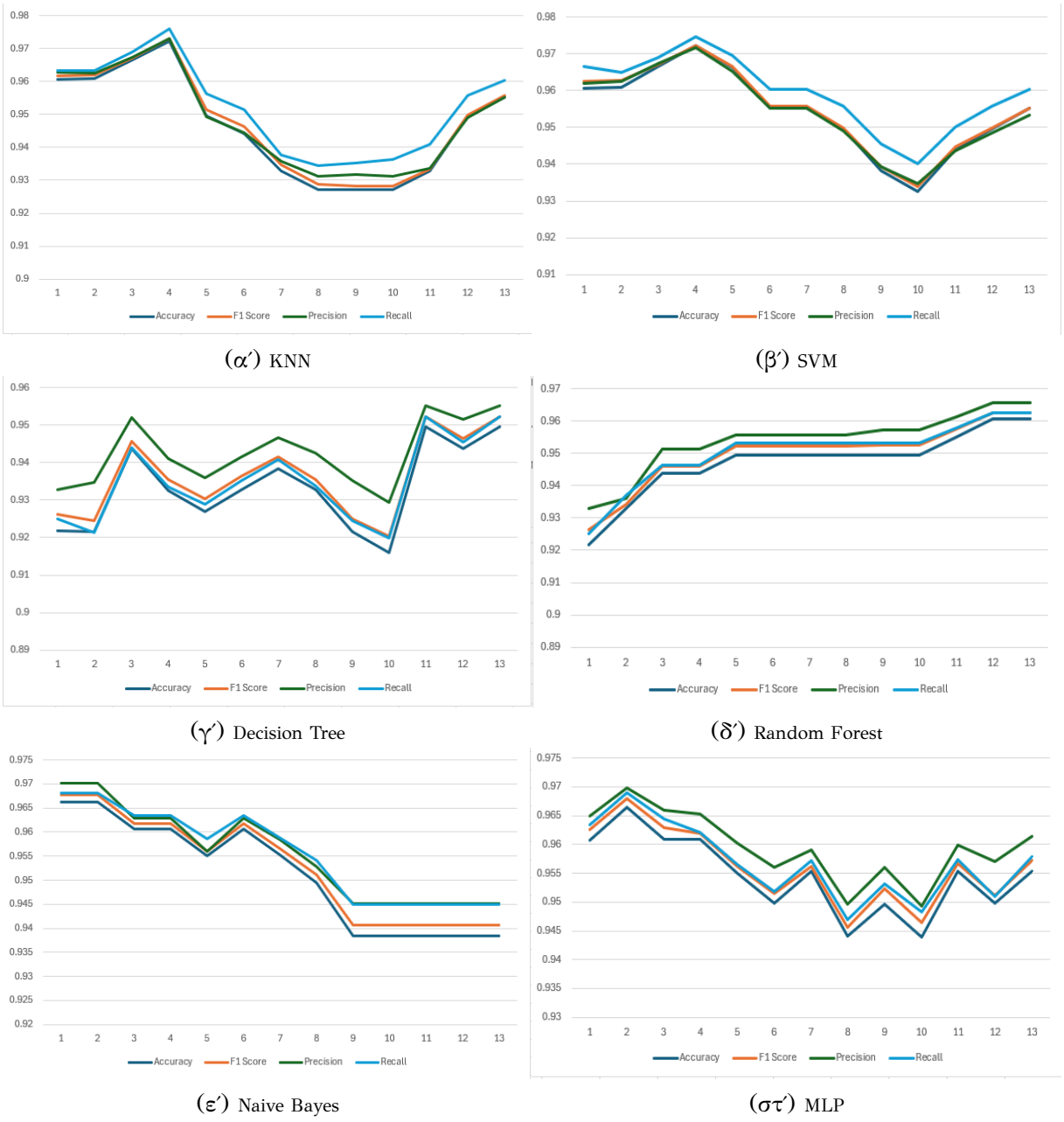
(ε΄) Naive Bayes



(στ΄) MLP

Σχήμα Α΄.36: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο LLE

Α΄.6.2 Wine Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

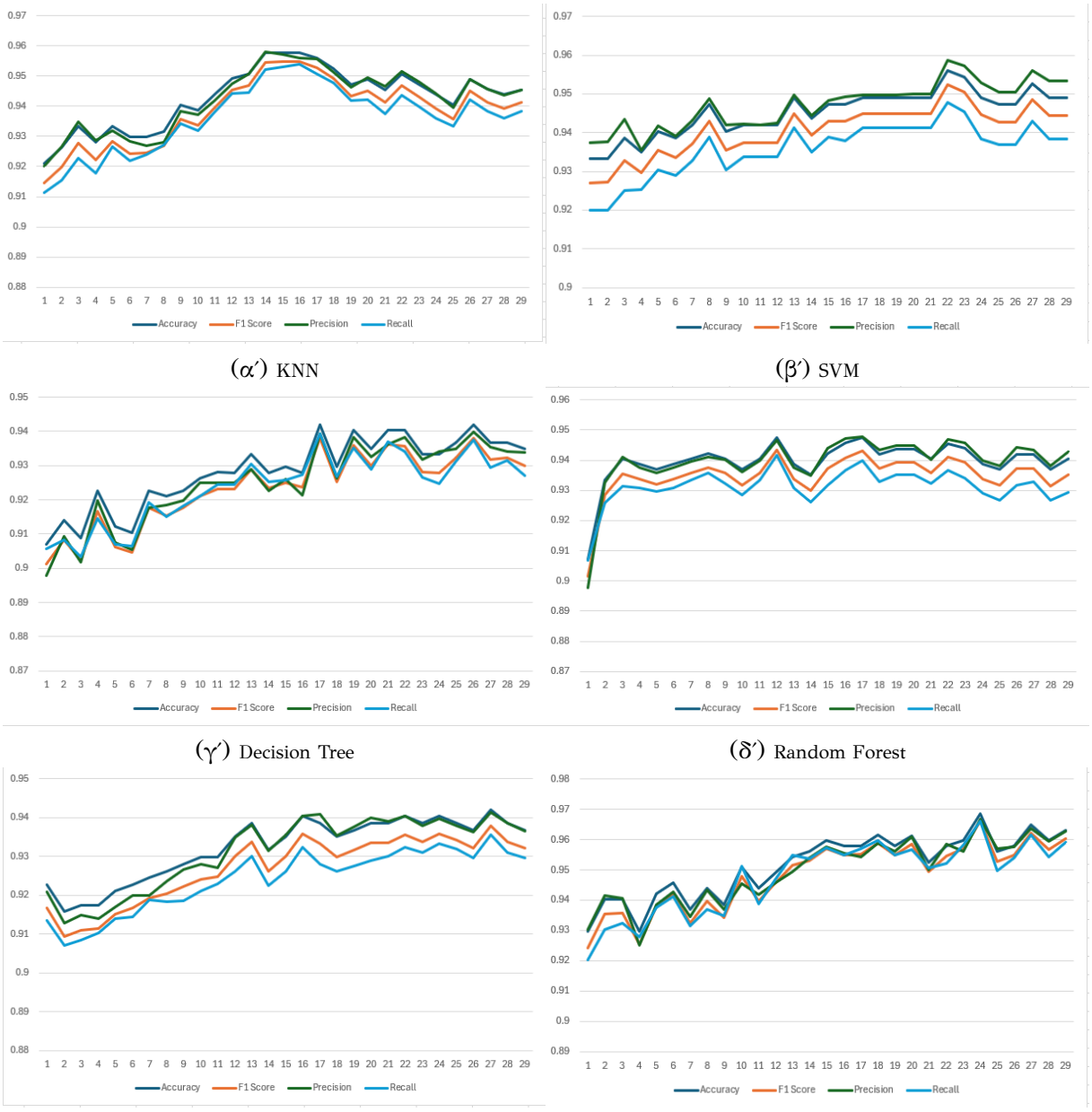
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.37: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο LLE

Α΄.6.3 Breast Cancer Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

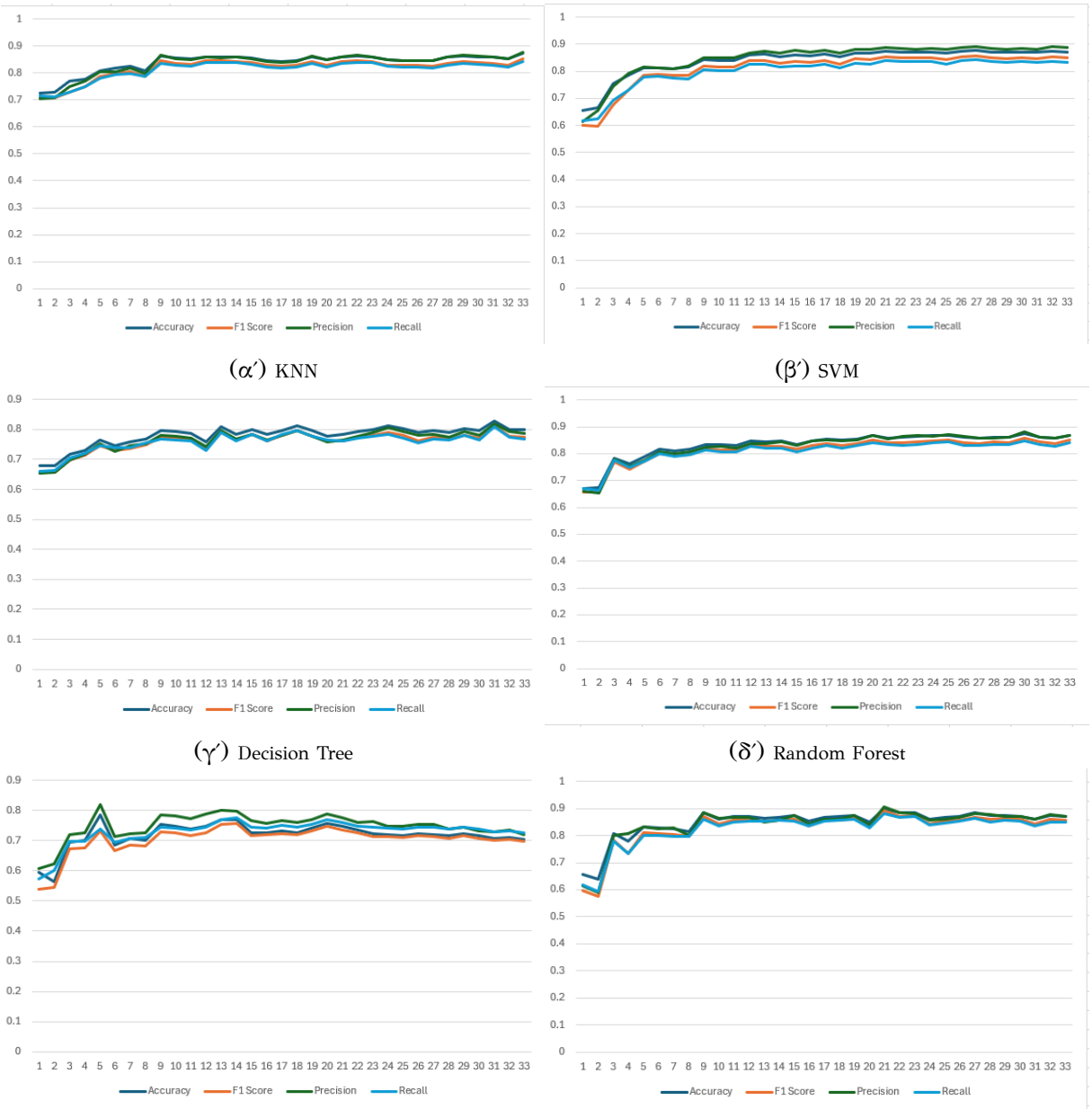
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.38: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο LLE

Α΄.6.4 Ionosphere Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.39: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο LLE

Α΄.6.5 Connectionist Bench Dataset

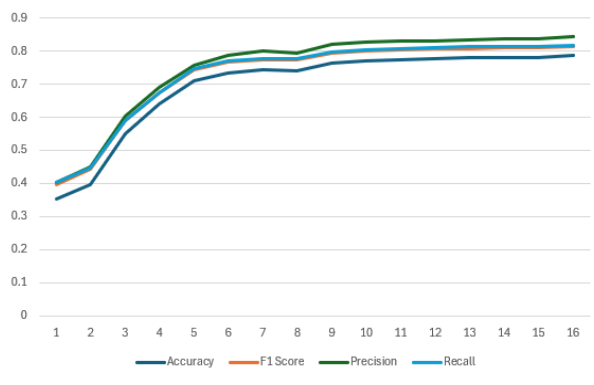
(α΄) KNN (β΄) SVM

(γ΄) Decision Tree (δ΄) SVD

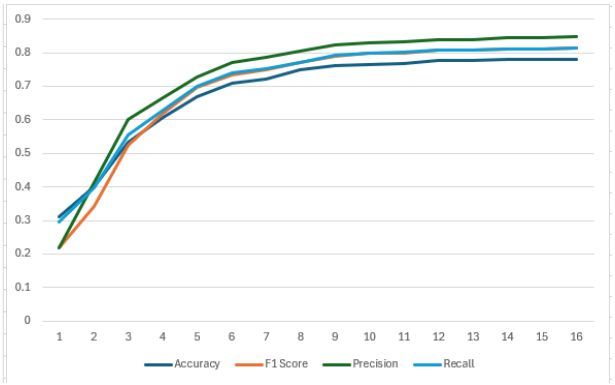
(ε΄) Naive Bayes (στ΄) MLP

Σχήμα Α΄.40: Ανάλυση διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο LLE

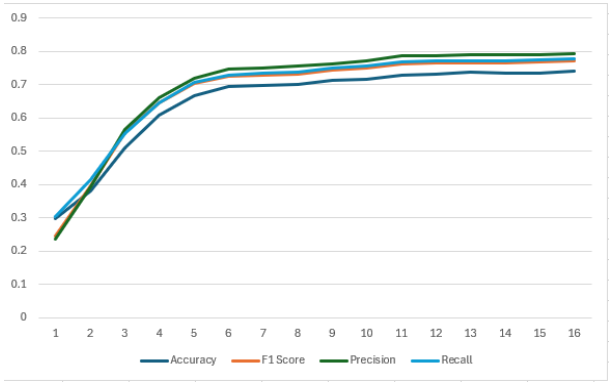
Α΄.6.6 Dry Bean Dataset



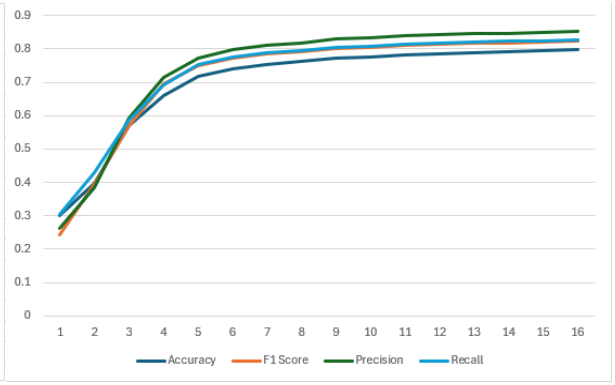
(α΄) KNN



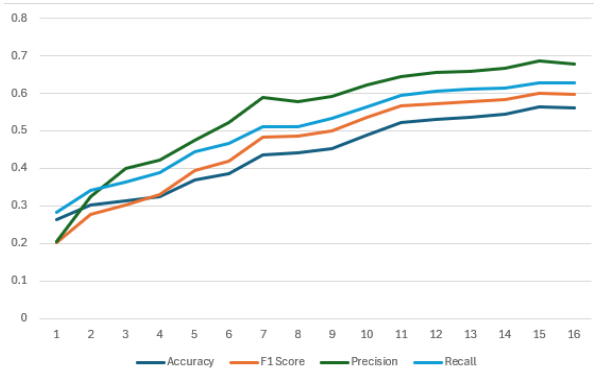
(β΄) SVM



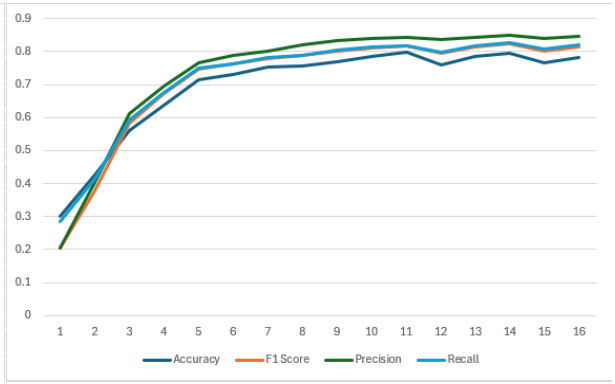
(γ΄) Decision Tree



(δ΄) Random Forest



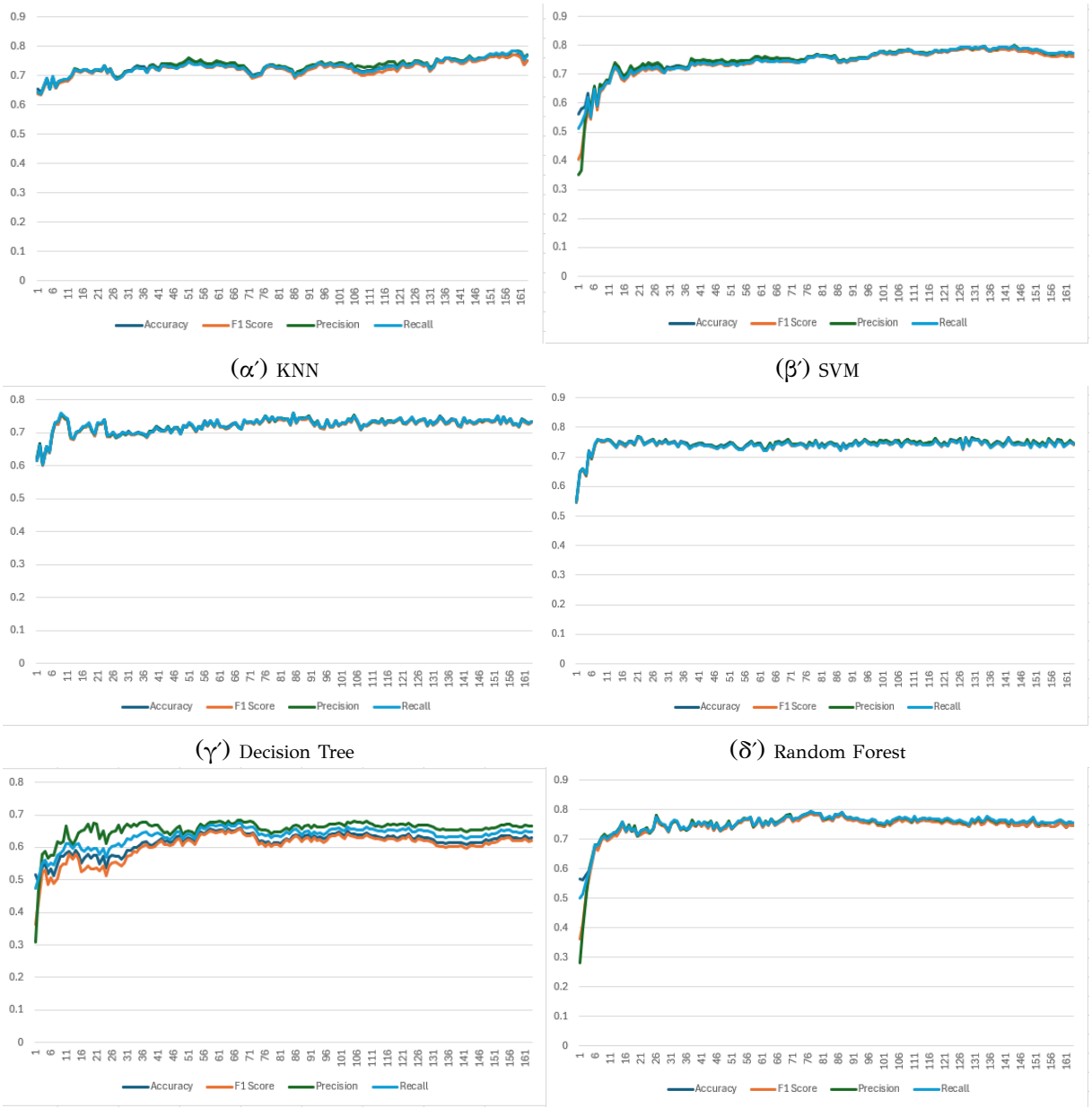
(ε΄) Naive Bayes



(στ΄) MLP

Σχήμα Α΄.41: Ανάλυση διαστάσεων στο Dry Bean Dataset με τη μέθοδο LLE

Α΄.6.7 Musk Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

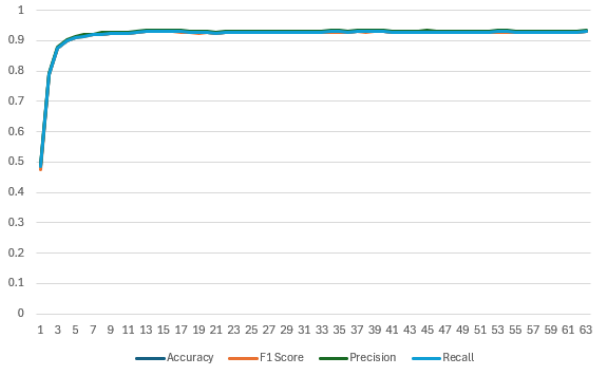
(ε΄) Naive Bayes

(στ΄) MLP

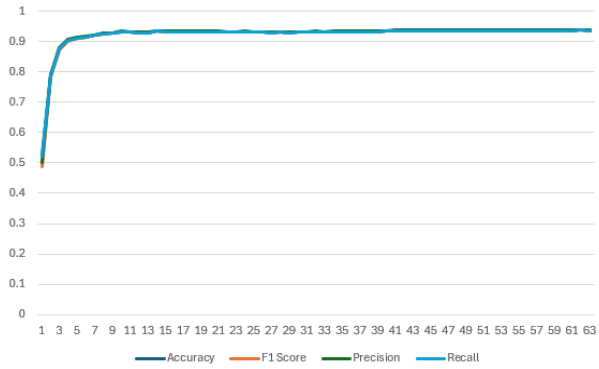
Σχήμα Α΄.42: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο LLE

Α΄.7 Αποτελέσματα Isomap

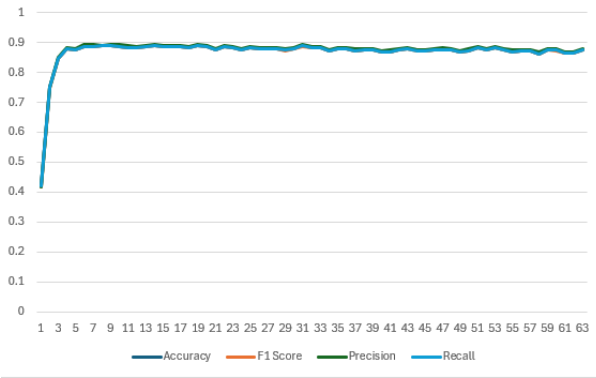
Α΄.7.1 Digits Dataset



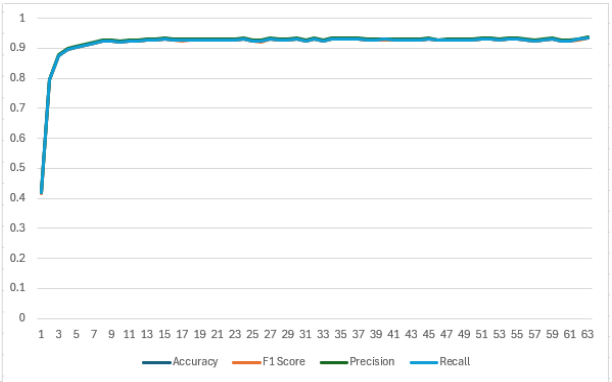
(α΄) PCA



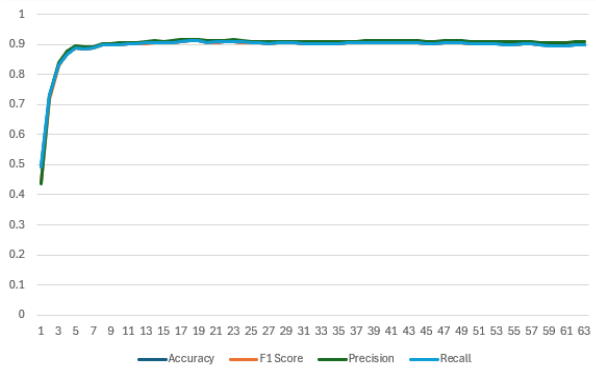
(β΄) LDA



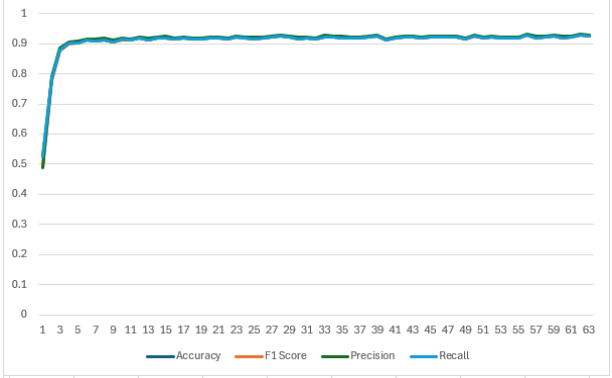
(γ΄) ICA



(δ΄) SVD



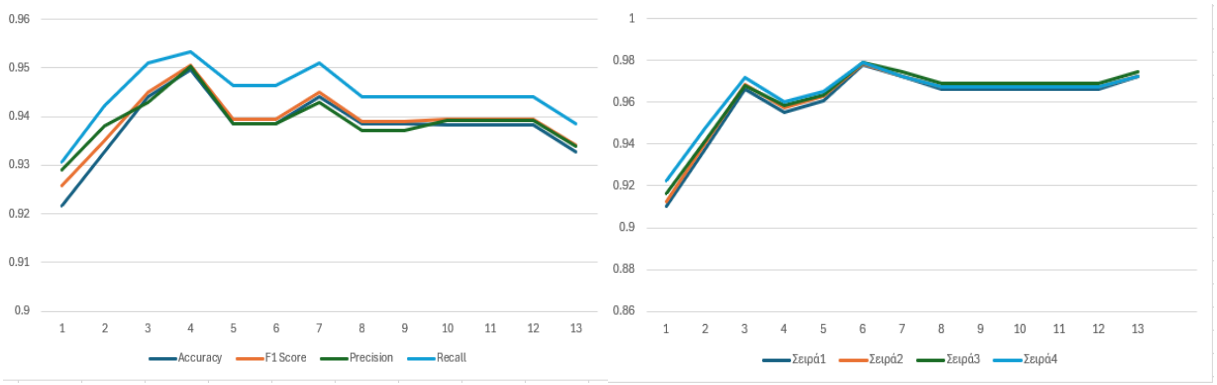
(ε΄) ICA



(στ΄) SVD

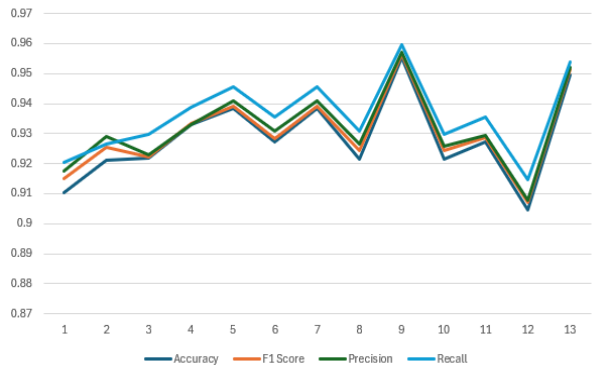
Σχήμα Α΄.43: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο Isomap

Α΄.7.2 Wine Dataset

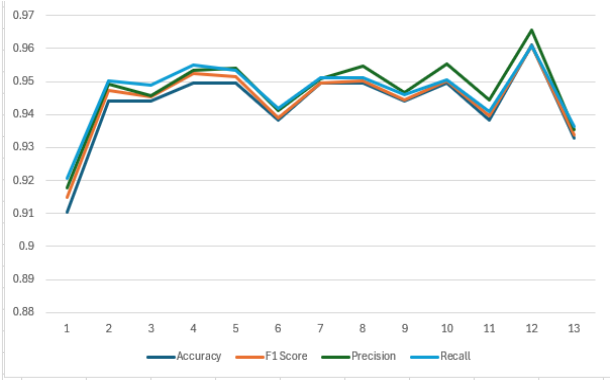


(α΄) PCA

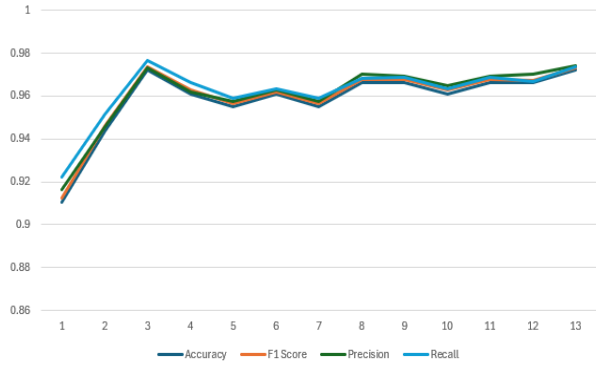
(β΄) LDA



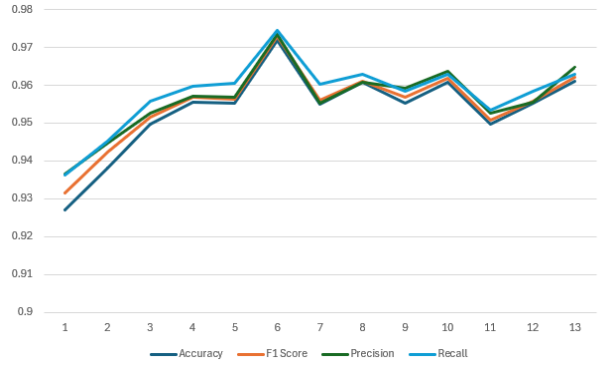
(γ΄) ICA



(δ΄) SVD



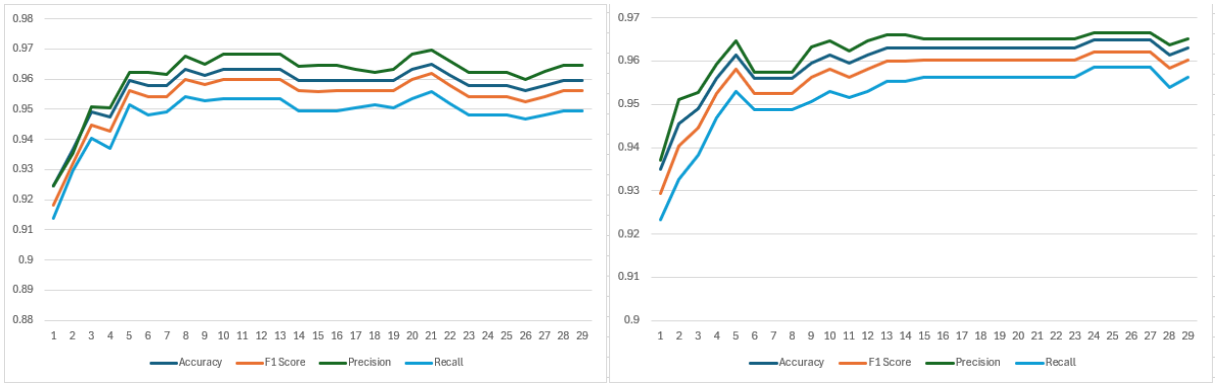
(ε΄) ICA



(στ΄) SVD

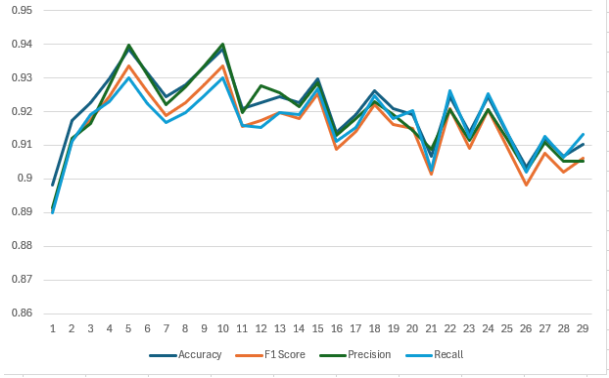
Σχήμα Α΄.44: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο Isomap

Α΄.7.3 Breast Cancer Dataset

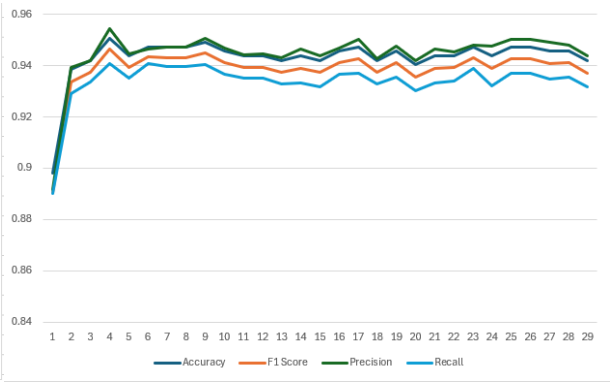


(α΄) KNN

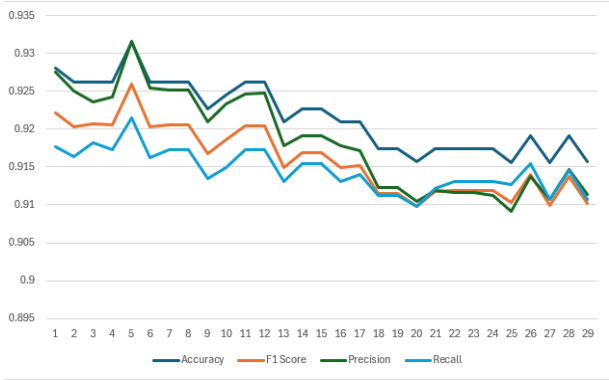
(β΄) SVM



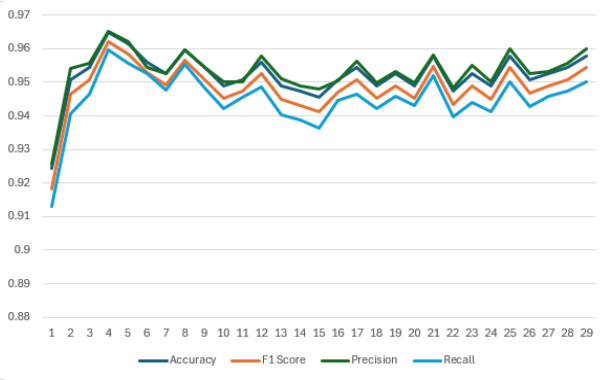
(γ΄) Decision Tree



(δ΄) Random Forest



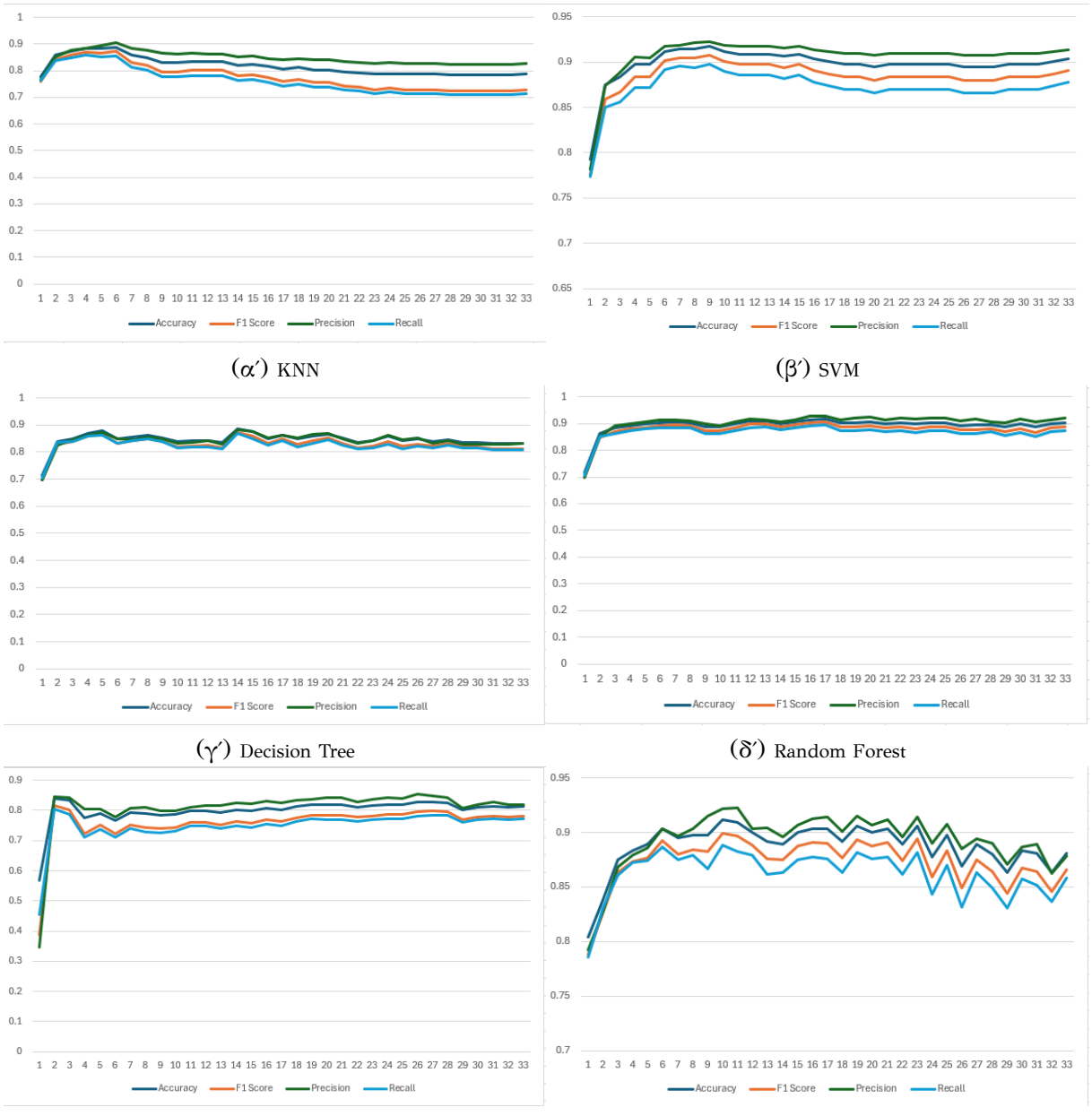
(ε΄) Naive Bayes



(στ΄) MLP

Σχήμα Α΄.45: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο Isomap

Α΄.7.4 Ionosphere Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.46: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο Isomap

Α΄.7.5 Connectionist Bench Dataset

(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

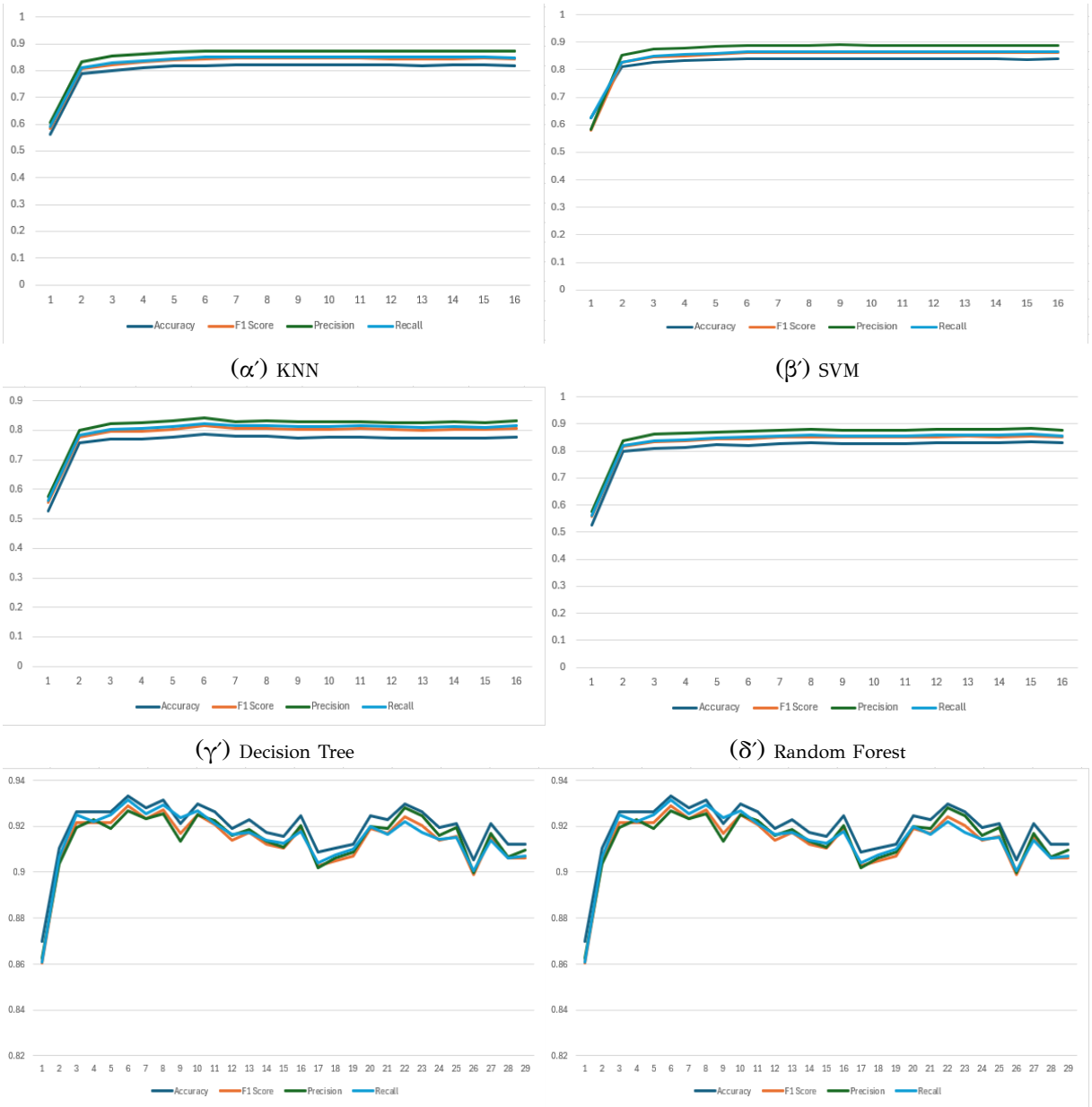
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.47: Ανάλυση διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο Isomap

Α΄.7.6 Dry Bean Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

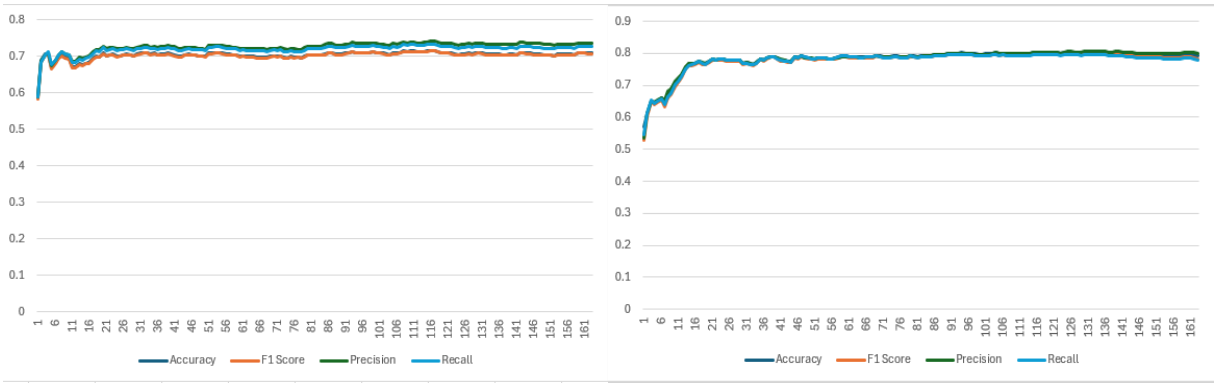
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

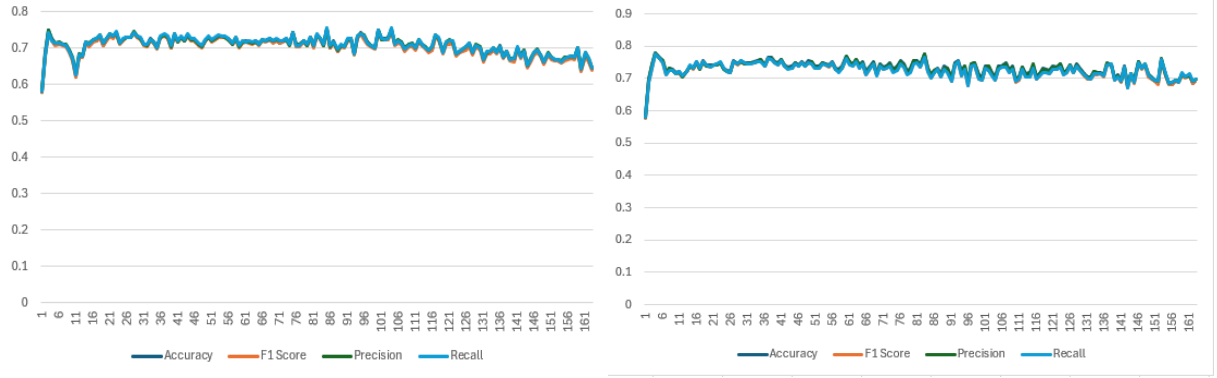
Σχήμα Α΄.48: Ανάλυση διαστάσεων στο Dry Bean Dataset με τη μέθοδο Isomap

Α΄.7.7 Musk Dataset



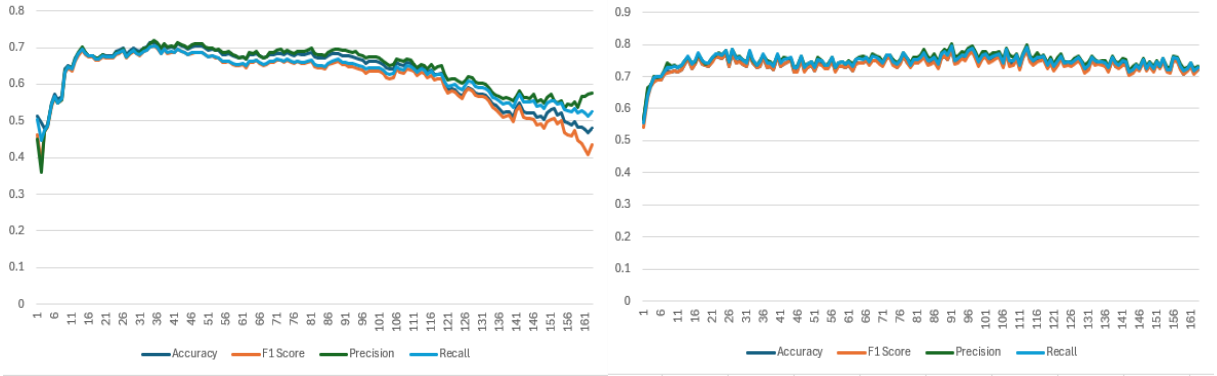
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest



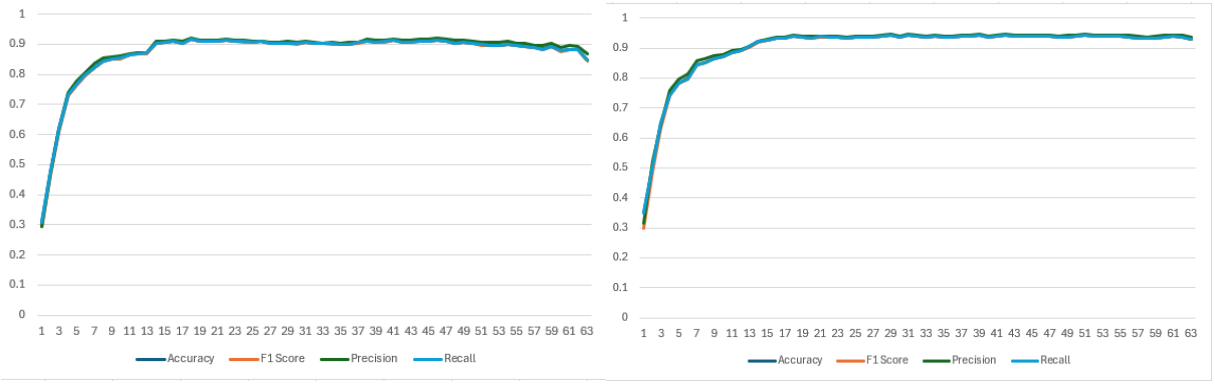
(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.49: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο Isomap

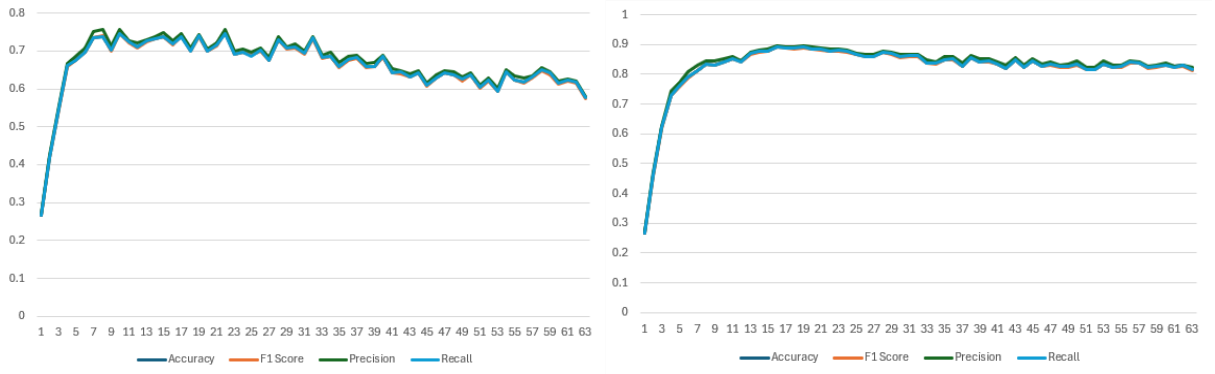
Α΄.8 Αποτελέσματα ICA

Α΄.8.1 Digits Dataset



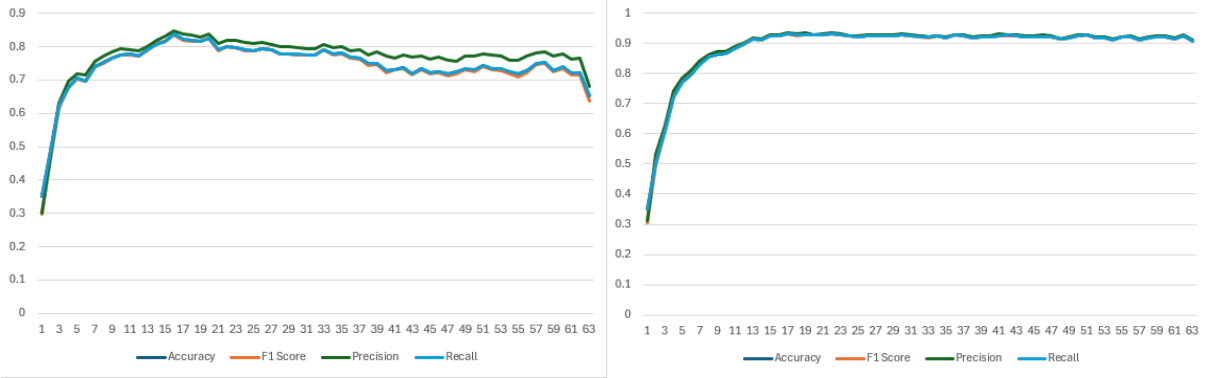
(α΄) KNN

(β΄) SVM



(γ΄) Decision Tree

(δ΄) Random Forest

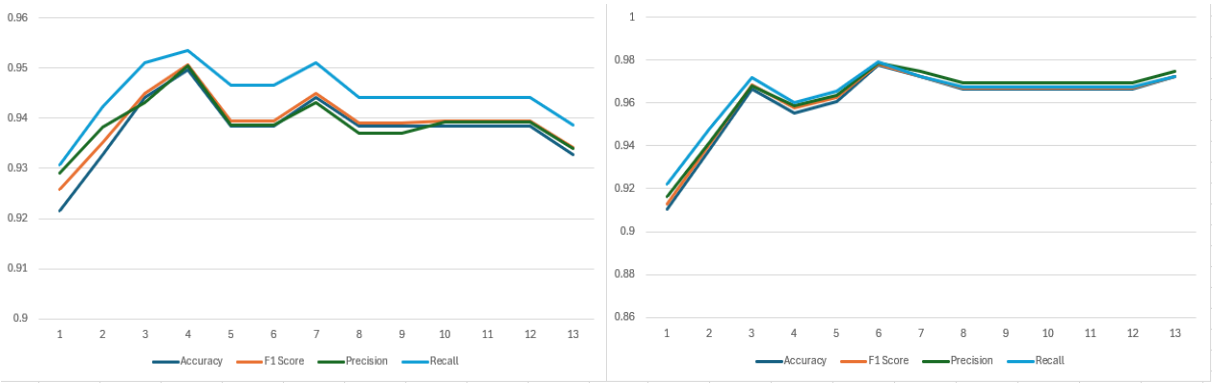


(ε΄) Naive Bayes

(στ΄) MLP

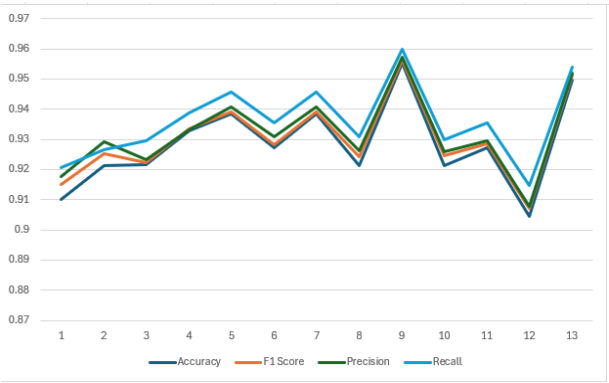
Σχήμα Α΄.50: Ανάλυση διαστάσεων στο Digits Dataset με τη μέθοδο ICA

A′.8.2 Wine Dataset

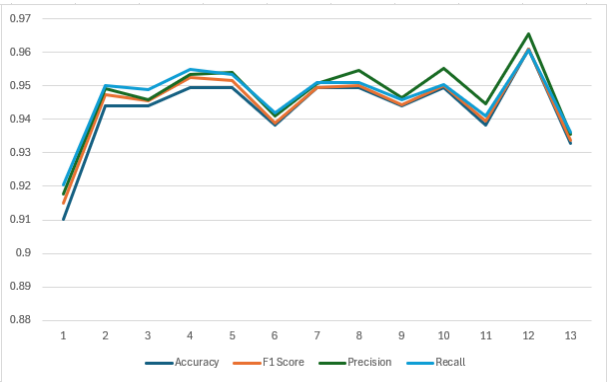


(α′) KNN

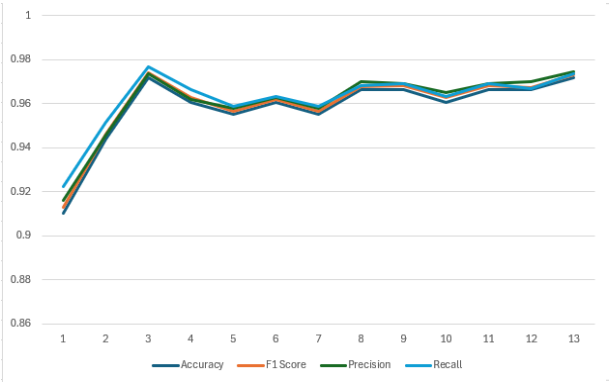
(β′) SVM



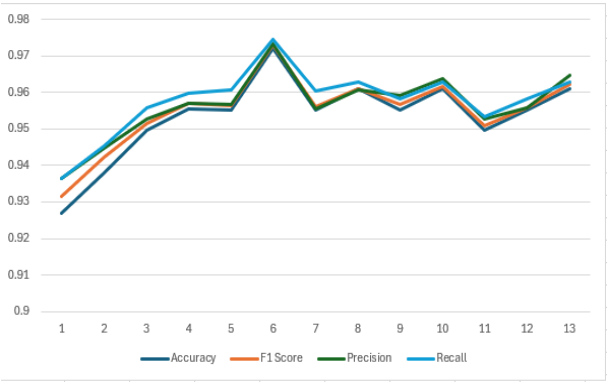
(γ′) Decision Tree



(δ′) Random Forest



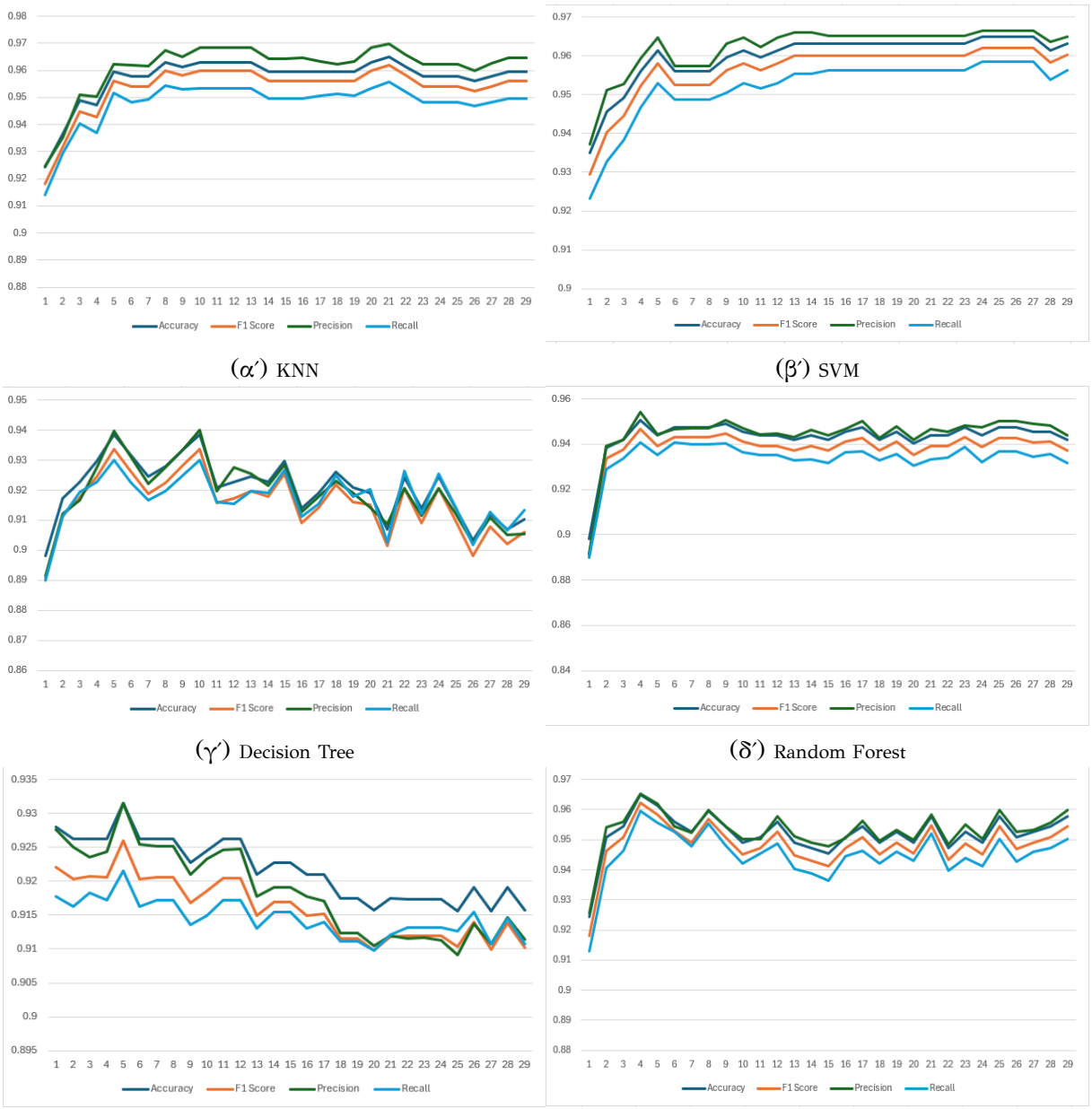
(ε′) Naive Bayes



(στ′) MLP

Σχήμα Α′.51: Ανάλυση διαστάσεων στο Wine Dataset με τη μέθοδο ICA

A΄.8.3 Breast Cancer Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

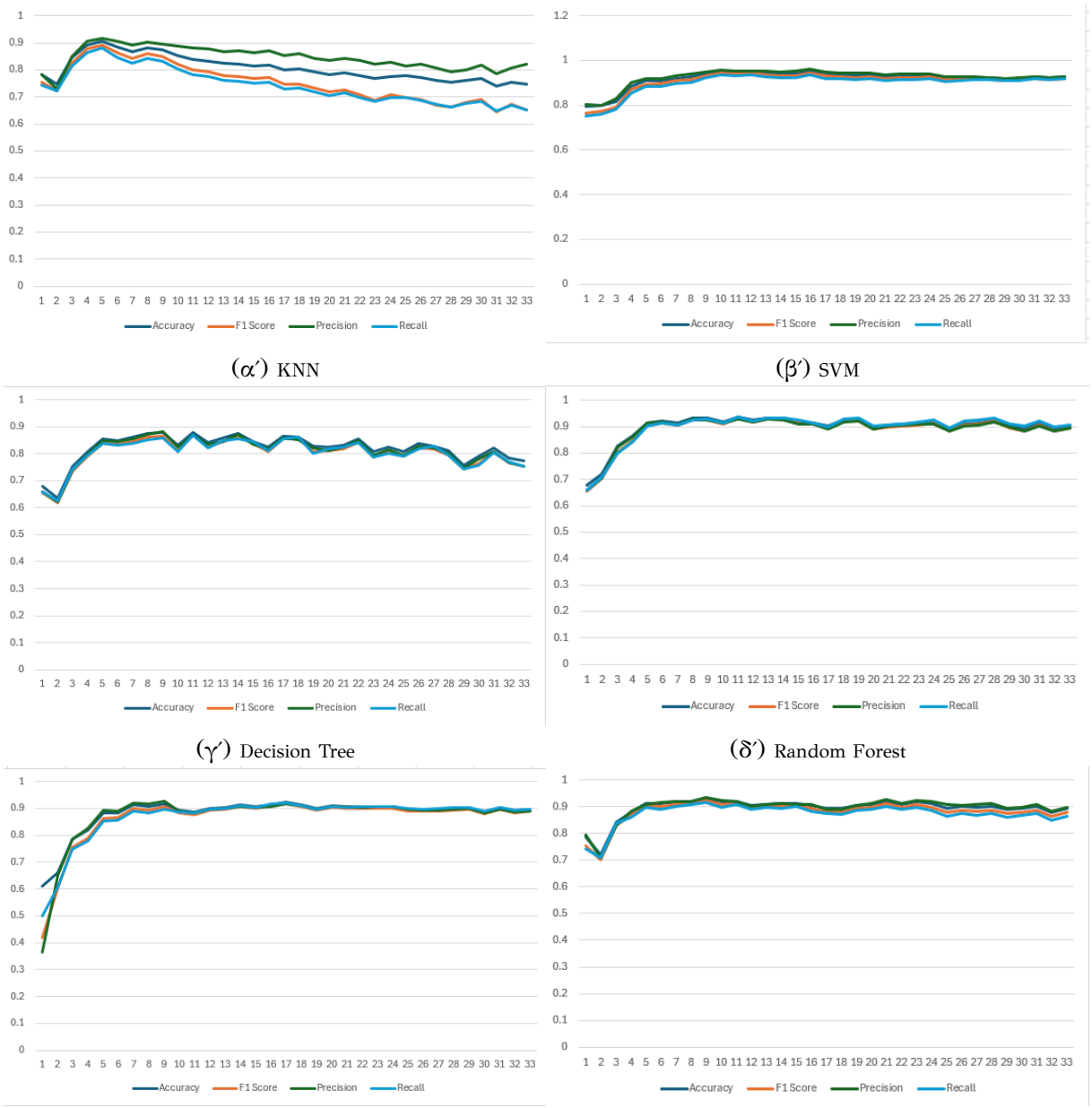
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.52: Ανάλυση διαστάσεων στο Breast Cancer Dataset με τη μέθοδο ICA

Α΄.8.4 Ionosphere Dataset



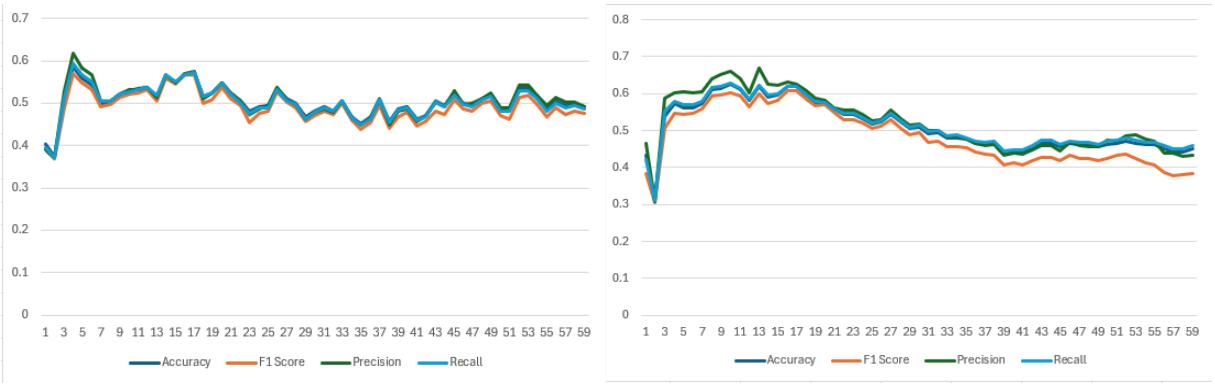
(α΄) KNN (β΄) SVM

(γ΄) Decision Tree (δ΄) Random Forest

(ε΄) Naive Bayes (στ΄) MLP

Σχήμα Α΄.53: Ανάλυση διαστάσεων στο Ionosphere Dataset με τη μέθοδο ICA

Α΄.8.5 ok Connectionist Bench Dataset

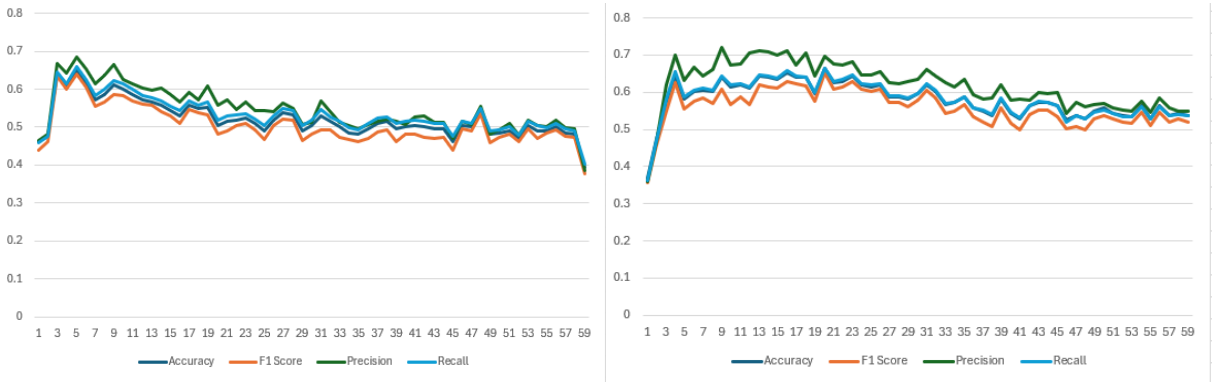


(α΄) KNN

(β΄) SVM

(γ΄) Decision tree

(δ΄) Random Forest

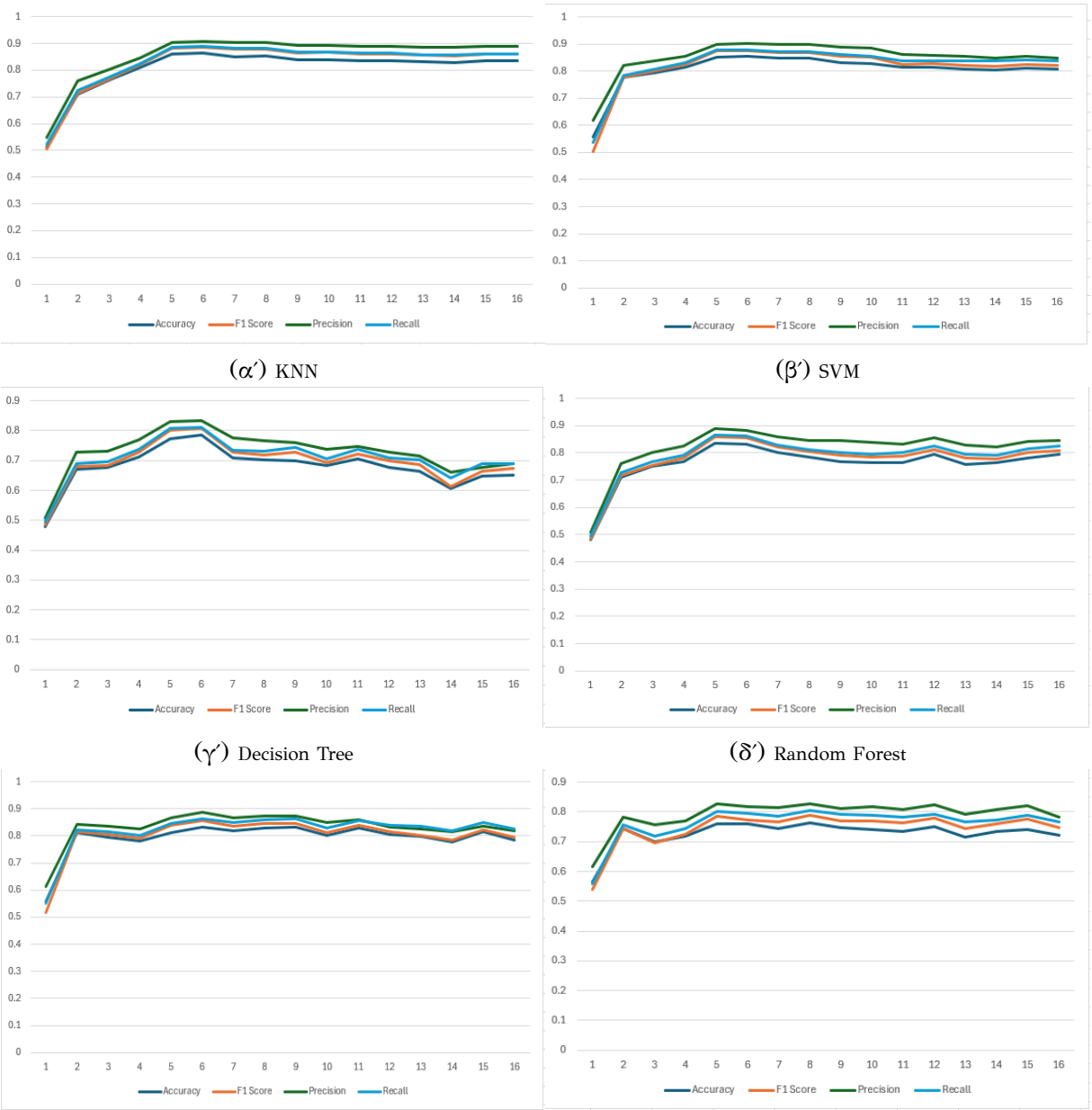


(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.54: Ανάλυση διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο ICA

Α΄.8.6 Dry Bean Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

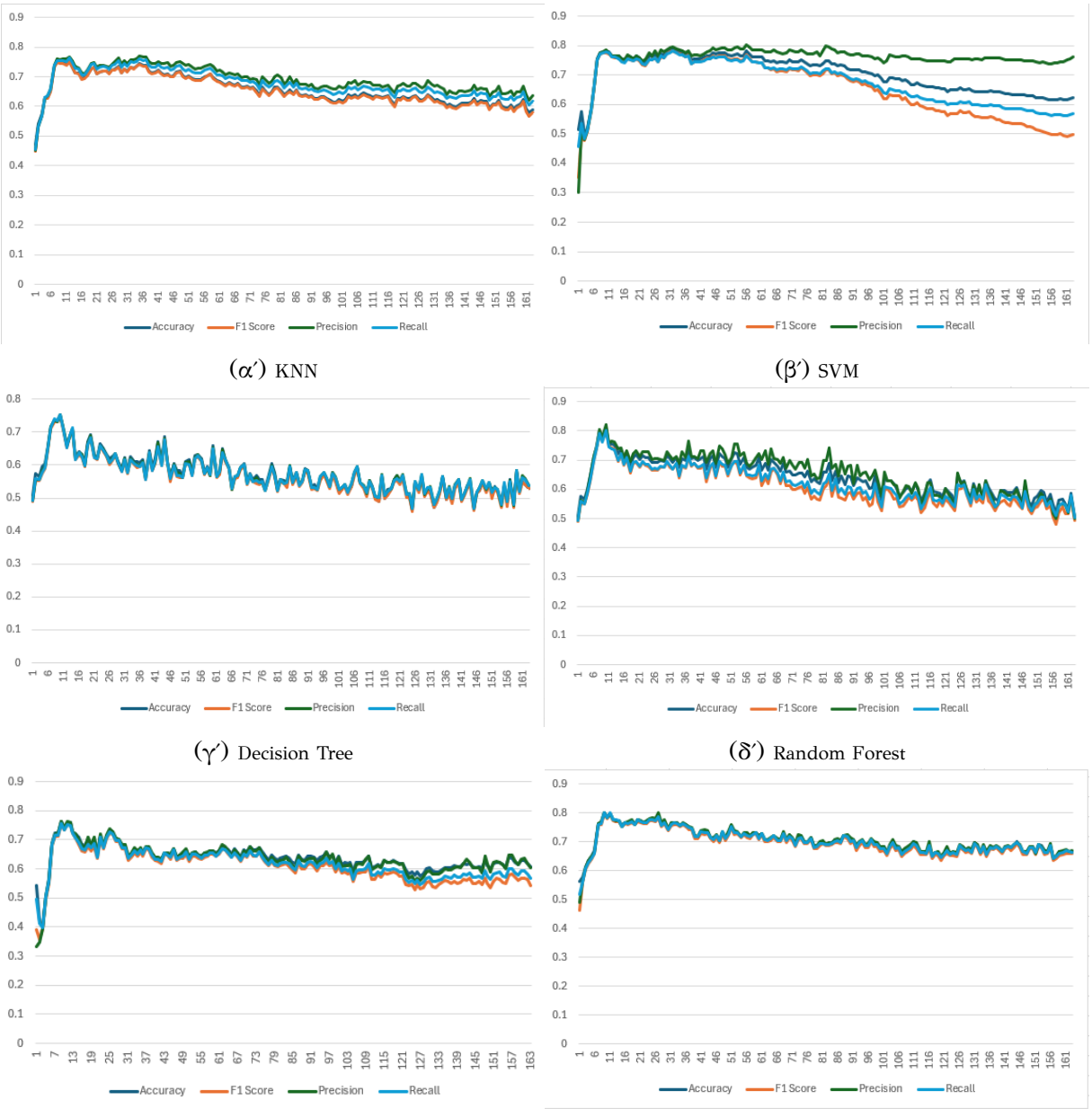
(δ΄) Random Forest

(ε΄) Naive Bayes

(στ΄) MLP

Σχήμα Α΄.55: Ανάλυση διαστάσεων στο Dry Bean Dataset με τη μέθοδο ICA

Α΄.8.7 Musk Dataset



(α΄) KNN

(β΄) SVM

(γ΄) Decision Tree

(δ΄) Random Forest

(ε΄) Naive Bayes

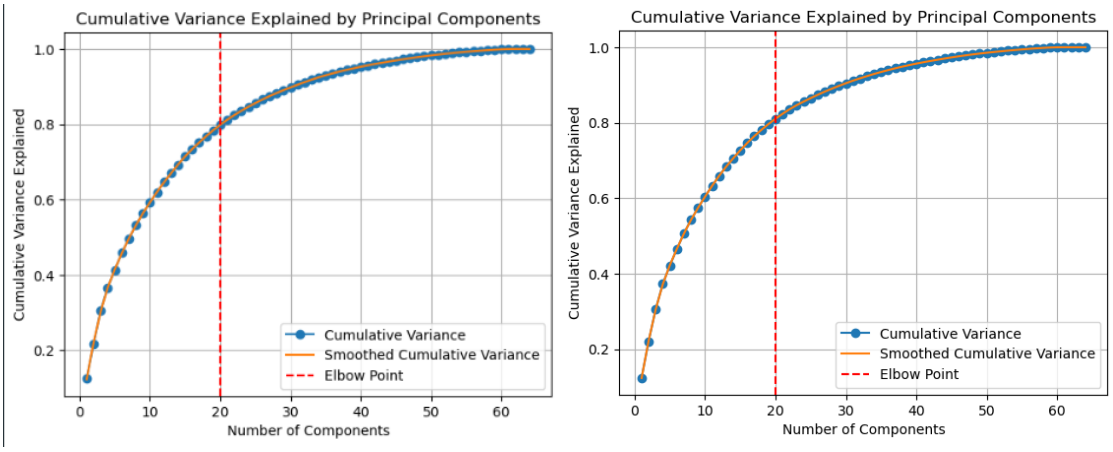
(στ΄) MLP

Σχήμα Α΄.56: Ανάλυση διαστάσεων στο Musk Dataset με τη μέθοδο ICA

Α΄.9 Αποτελέσματα τεχνικών αυτόματης εύρεσης του αριθμού διαστάσεων

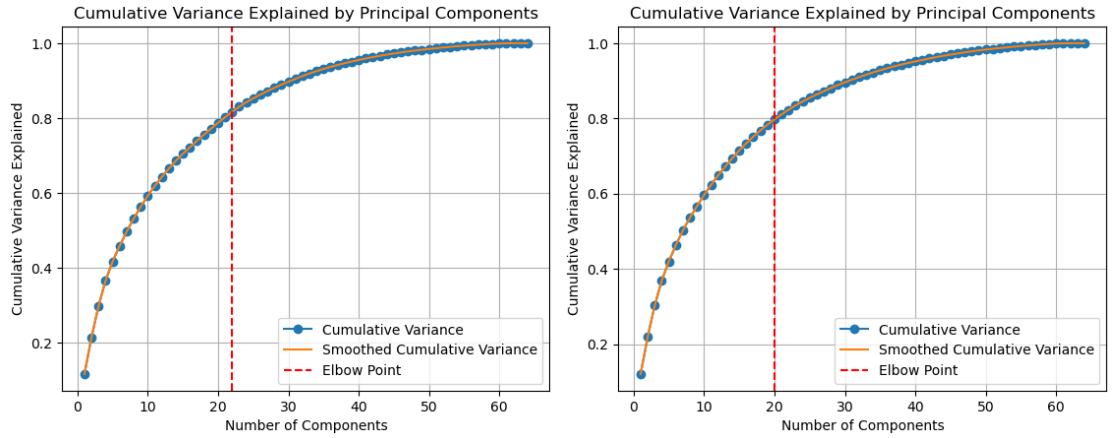
Α΄.9.1 PCA

Digits Dataset



(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3

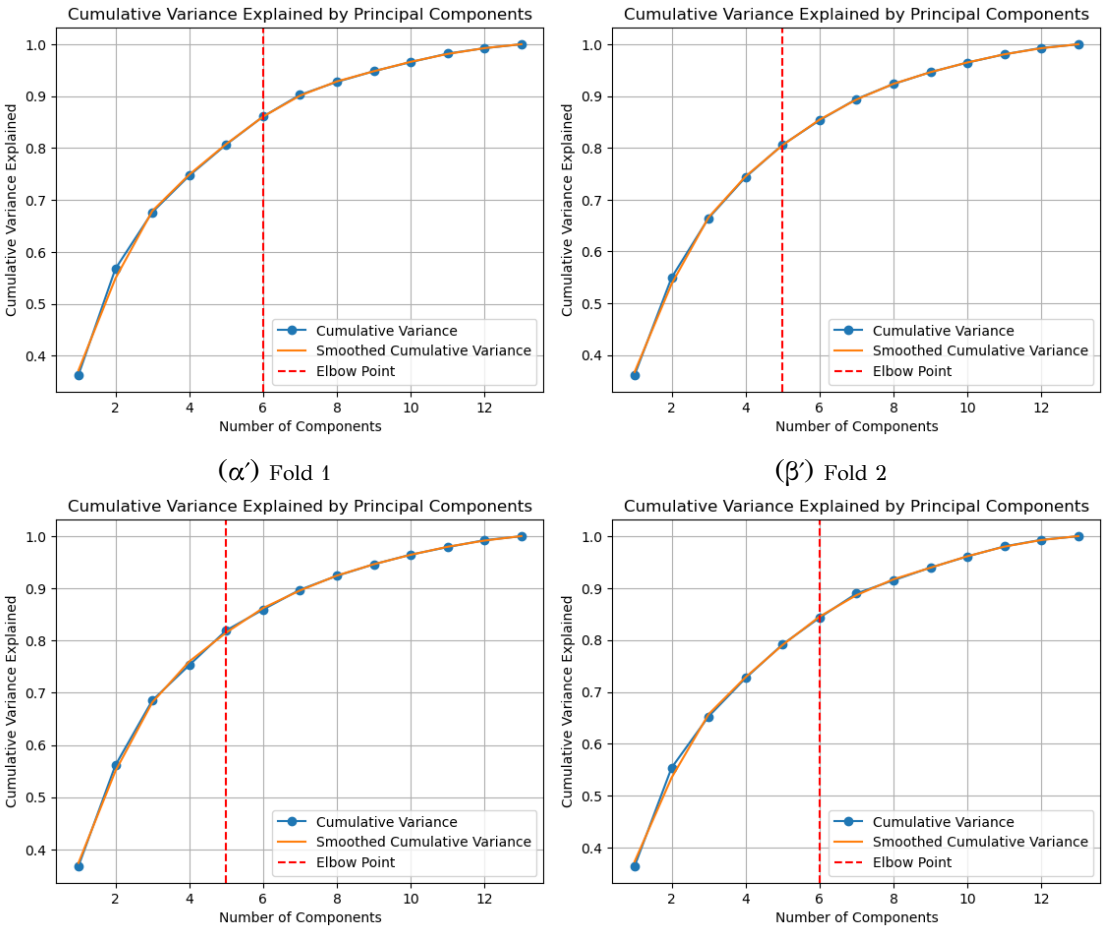
(δ΄) Fold 4

Σχήμα Α΄.57: Ανάλυση βέλτιστων διαστάσεων στο Digits Dataset με τη μέθοδο PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.94	0.94	0.94	0.94
Average SVM	0.95	0.95	0.95	0.95
Average Decision Tree	0.77	0.76	0.78	0.77
Average Random Forest	0.92	0.92	0.92	0.92
Average Naive Bayes	0.85	0.85	0.86	0.85
Average MLP	0.94	0.94	0.94	0.94

Πίνακας Α΄.1: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων PCA Digits Dataset.

Wine Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

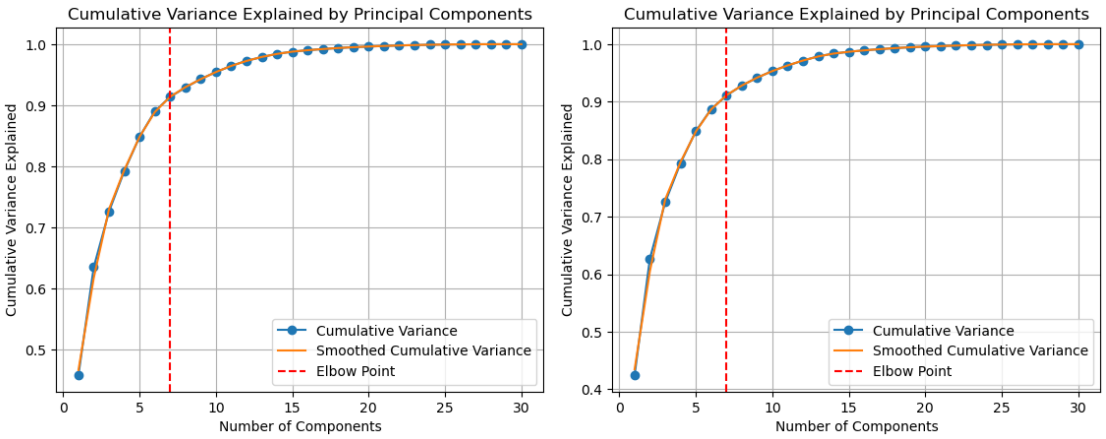
(δ΄) Fold 4

Σχήμα Α΄.58: Ανάλυση βέλτιστων διαστάσεων στο Wine Dataset με τη μέθοδο PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.98	0.98	0.98	0.98
Average Decision Tree	0.93	0.93	0.94	0.94
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.95	0.95	0.95	0.95
Average MLP	0.96	0.96	0.96	0.96

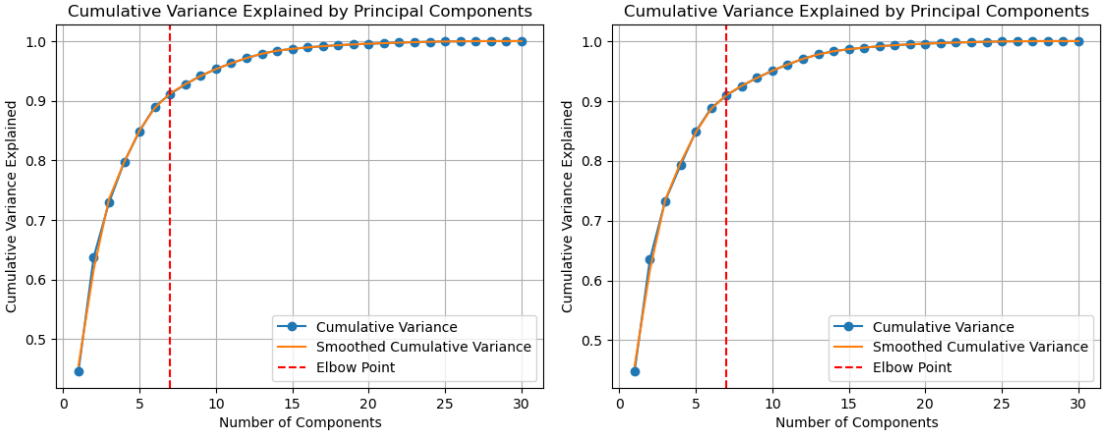
Πίνακας Α΄.2: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων PCA Wine Dataset.

Breast Cancer Dataset



(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3

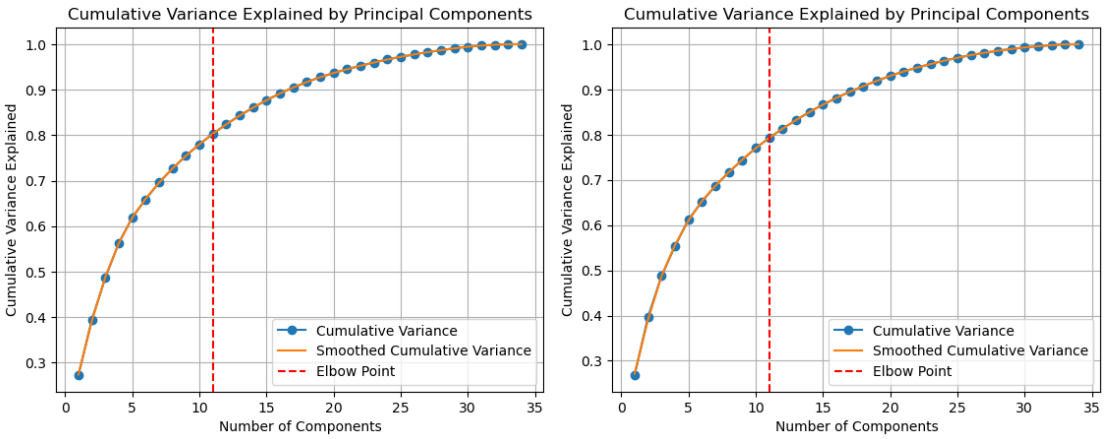
(δ΄) Fold 4

Σχήμα Α΄.59: Ανάλυση βέλτιστων διαστάσεων στο Breast Cancer Dataset με τη μέθοδο PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.96	0.96	0.96	0.96
Average Decision Tree	0.94	0.93	0.93	0.93
Average Random Forest	0.94	0.94	0.94	0.94
Average Naive Bayes	0.93	0.92	0.93	0.92
Average MLP	0.97	0.97	0.97	0.97

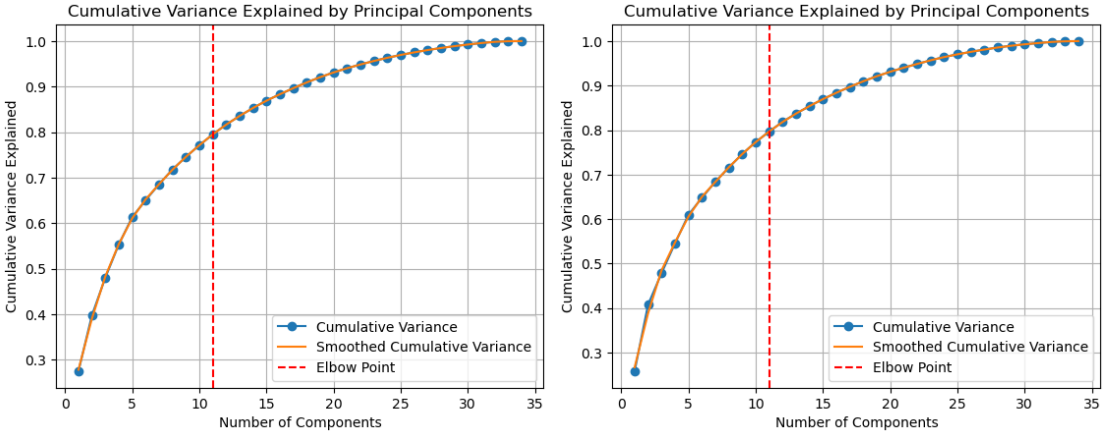
Πίνακας Α΄.3: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων PCA Breast Cancer Dataset.

Ionosphere Dataset



(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3

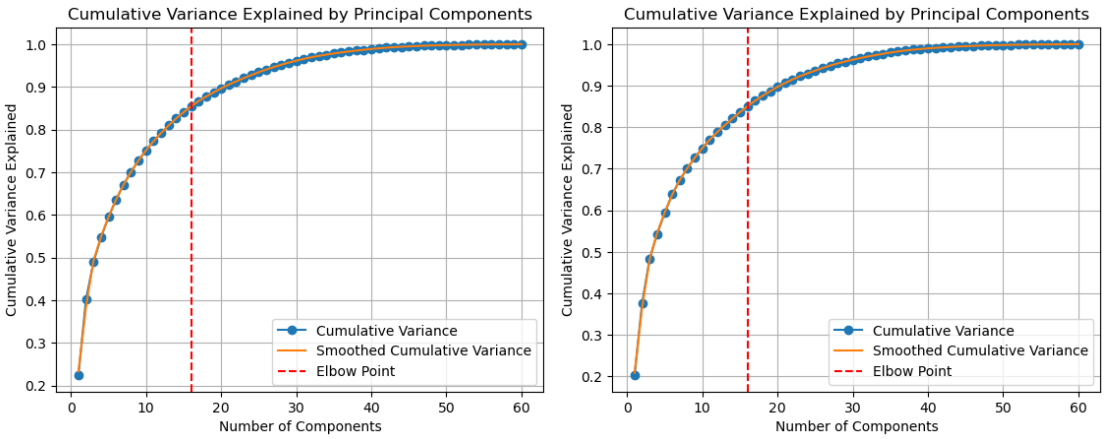
(δ΄) Fold 4

Σχήμα Α΄.60: Ανάλυση βέλτιστων διαστάσεων στο Ionosphere Dataset με τη μέθοδο PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.85	0.83	0.88	0.81
Average SVM	0.93	0.92	0.94	0.91
Average Decision Tree	0.87	0.86	0.87	0.87
Average Random Forest	0.93	0.93	0.93	0.92
Average Naive Bayes	0.87	0.86	0.87	0.86
Average MLP	0.91	0.90	0.90	0.89

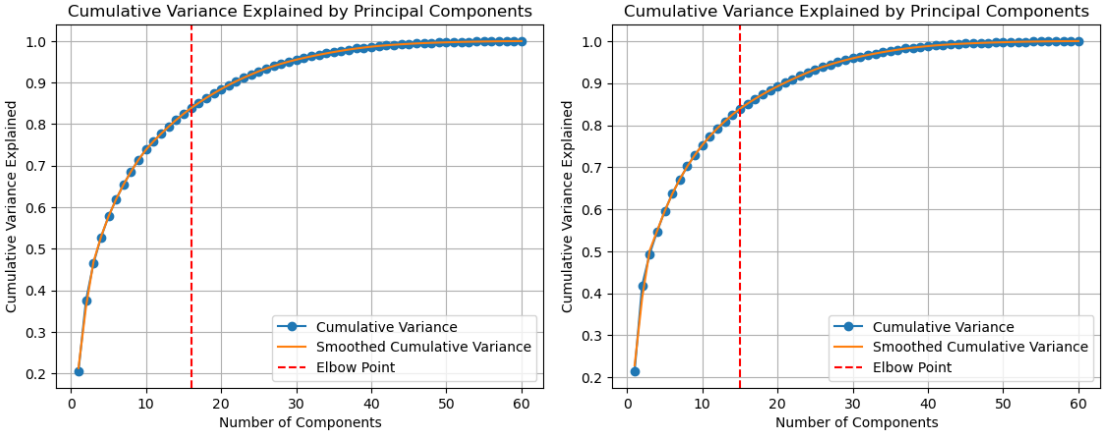
Πίνακας Α΄.4: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων PCA Ionosphere Dataset.

Connectionist Bench Dataset



(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3

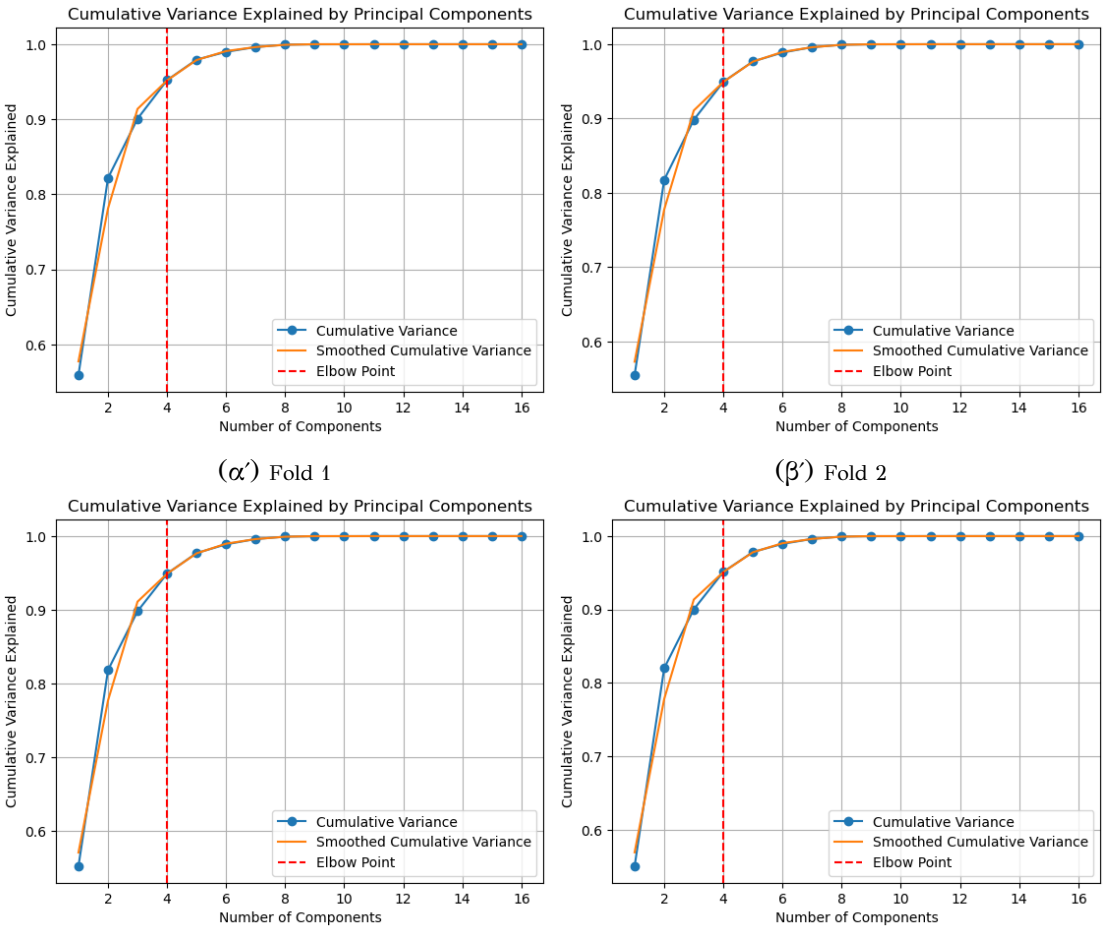
(δ΄) Fold 4

Σχήμα Α΄.61: Ανάλυση βέλτιστων διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.57	0.57	0.58	0.58
Average SVM	0.58	0.57	0.60	0.58
Average Decision Tree	0.55	0.53	0.60	0.56
Average Random Forest	0.65	0.63	0.70	0.65
Average Naive Bayes	0.57	0.56	0.59	0.58
Average MLP	0.63	0.62	0.67	0.64

Πίνακας Α΄.5: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων PCA Connectionist Bench Dataset.

Dry Bean Dataset



(α΄) Fold 1 (β΄) Fold 2

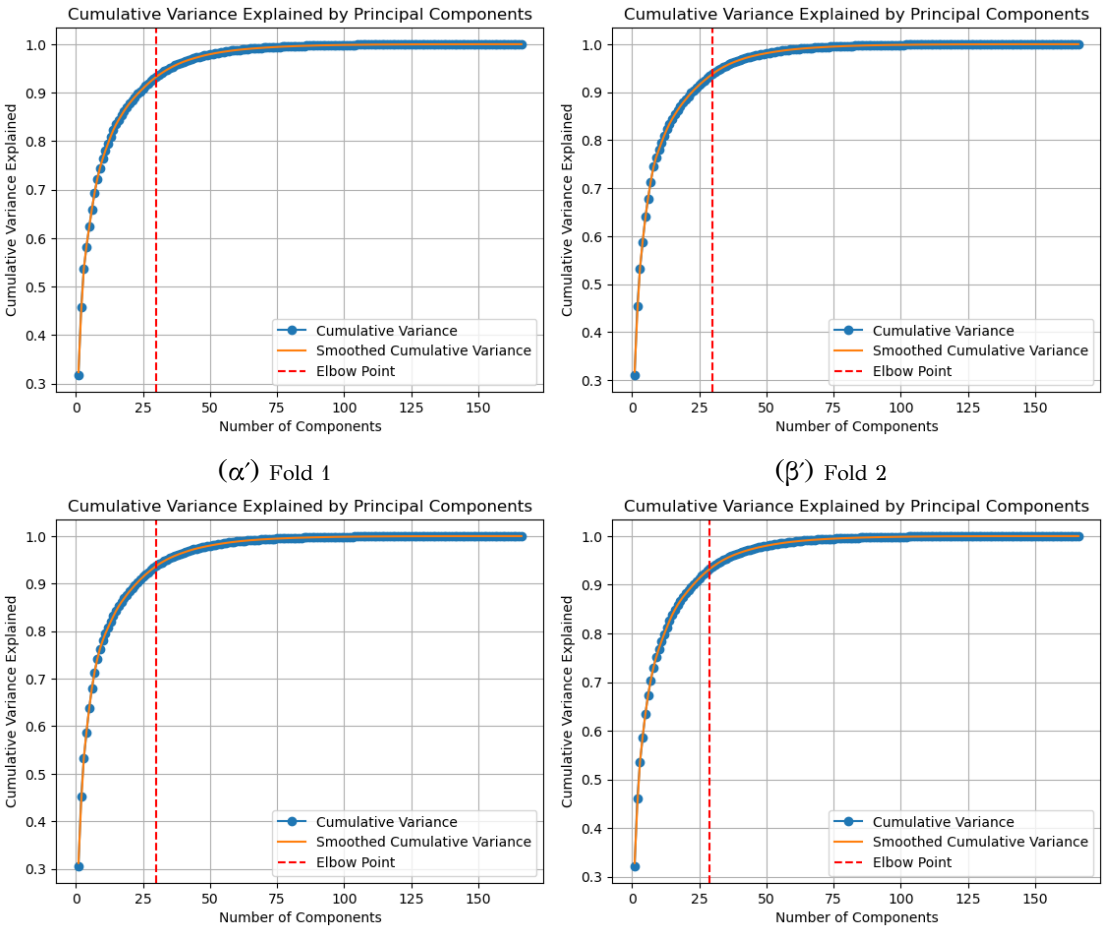
(γ΄) Fold 3 (δ΄) Fold 4

Σχήμα Α΄.62: Ανάλυση βέλτιστων διαστάσεων στο Dry Bean Dataset με τη μέθοδο PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.77	0.78	0.82	0.79
Average SVM	0.79	0.80	0.84	0.81
Average Decision Tree	0.70	0.70	0.75	0.71
Average Random Forest	0.76	0.77	0.81	0.78
Average Naive Bayes	0.82	0.83	0.85	0.84
Average MLP	0.70	0.72	0.78	0.74

Πίνακας Α΄.6: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων PCA Dry Bean Dataset.

Musk Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

(δ΄) Fold 4

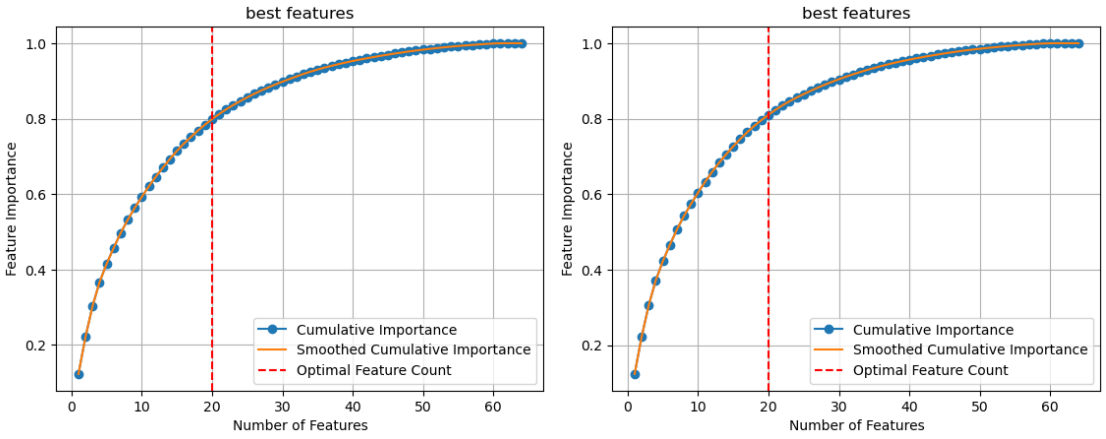
Σχήμα Α΄.63: Ανάλυση βέλτιστων διαστάσεων στο Musk Dataset με τη μέθοδο PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.75	0.75	0.76	0.76
Average SVM	0.77	0.76	0.78	0.76
Average Decision Tree	0.71	0.71	0.71	0.71
Average Random Forest	0.74	0.73	0.75	0.73
Average Naive Bayes	0.75	0.74	0.75	0.74
Average MLP	0.78	0.78	0.79	0.78

Πίνακας Α΄.7: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων PCA Musk Dataset.

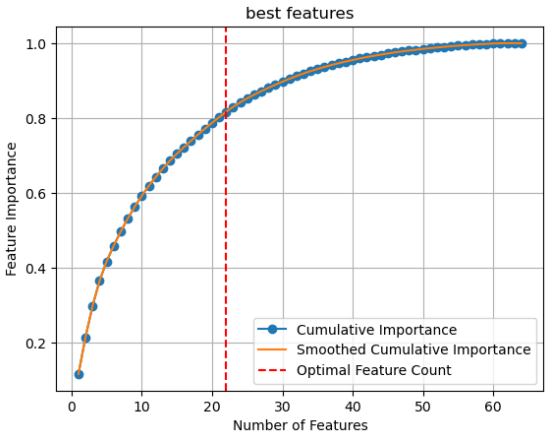
Α΄.9.2 Kernel PCA

Digits Dataset

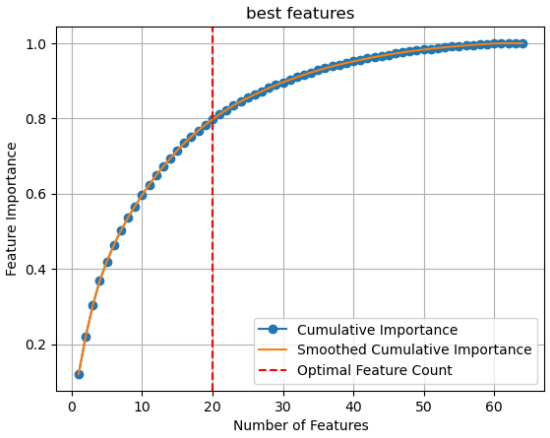


(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3



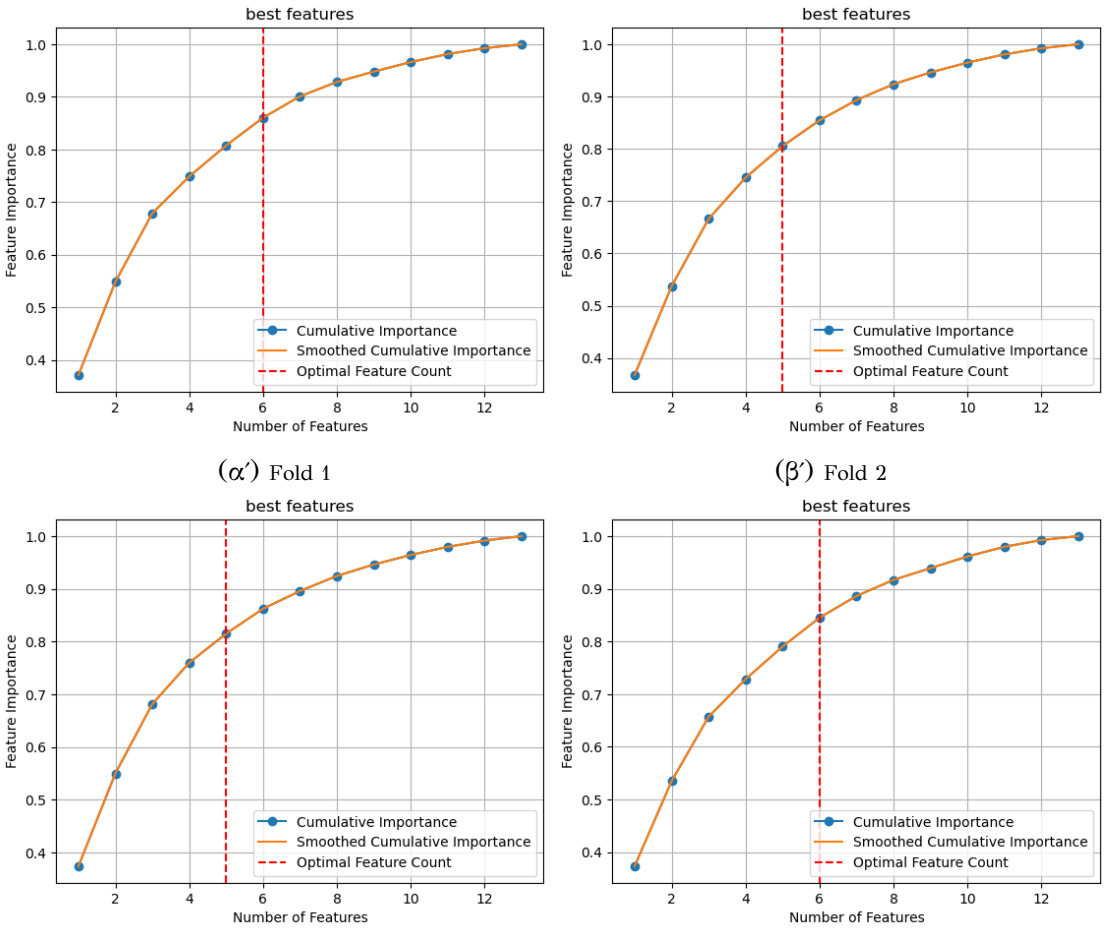
(δ΄) Fold 4

Σχήμα Α΄.64: Ανάλυση βέλτιστων διαστάσεων στο Digits Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.94	0.94	0.94	0.94
Average SVM	0.95	0.95	0.95	0.95
Average Decision Tree	0.77	0.77	0.78	0.77
Average Random Forest	0.92	0.92	0.92	0.92
Average Naive Bayes	0.85	0.85	0.86	0.85
Average MLP	0.94	0.94	0.94	0.94

Πίνακας Α΄.8: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Kernel PCA Digits Dataset.

Wine Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

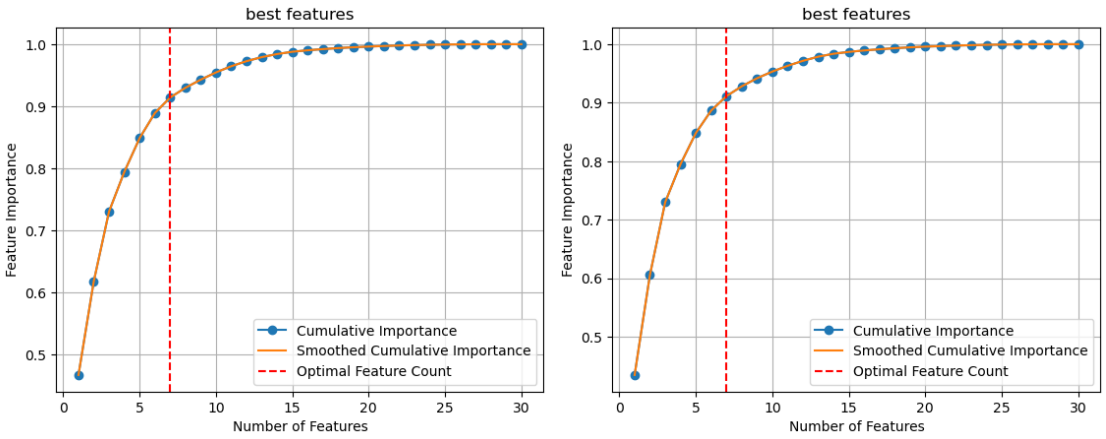
(δ΄) Fold 4

Σχήμα Α΄.65: Ανάλυση βέλτιστων διαστάσεων στο Wine Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.98	0.98	0.98	0.98
Average Decision Tree	0.93	0.93	0.94	0.94
Average Random Forest	0.95	0.95	0.95	0.95
Average Naive Bayes	0.95	0.95	0.95	0.95
Average MLP	0.97	0.97	0.97	0.97

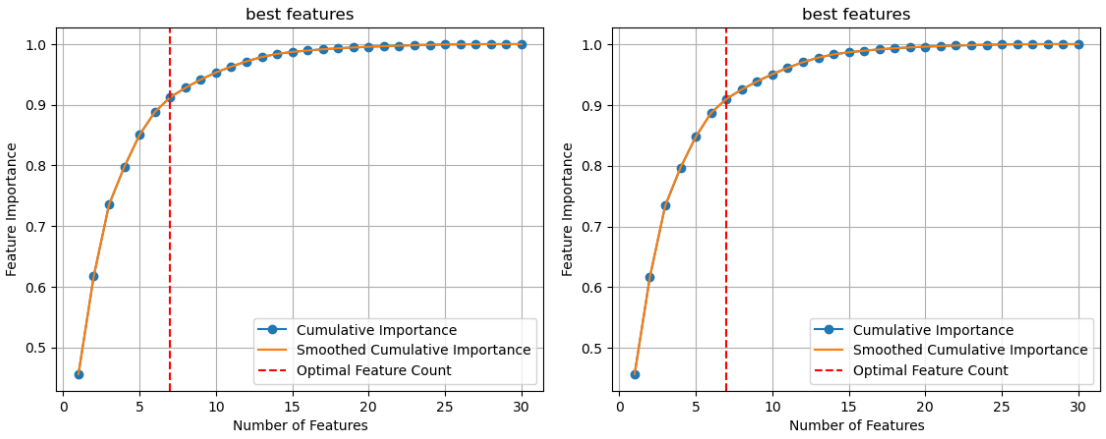
Πίνακας Α΄.9: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Kernel PCA Wine Dataset.

Breast Cancer Dataset



(α′) Fold 1

(β′) Fold 2



(γ′) Fold 3

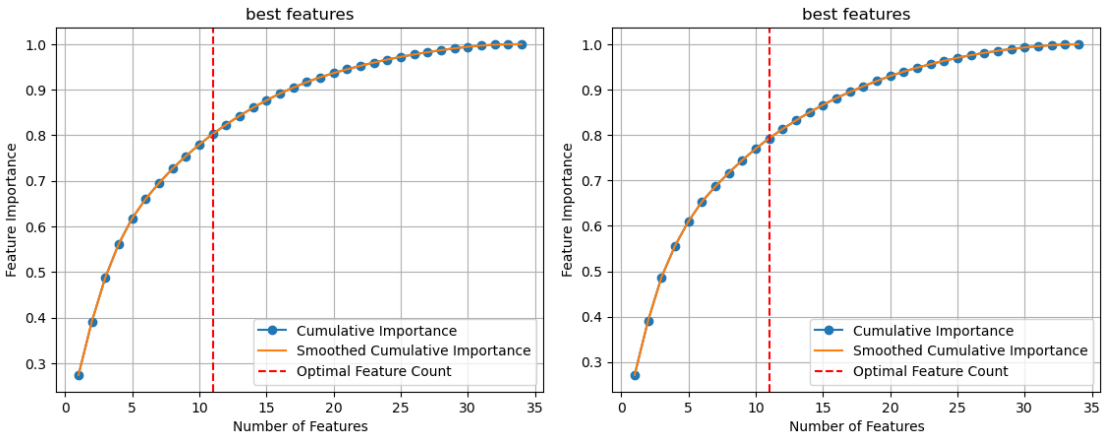
(δ′) Fold 4

Σχήμα Α′.66: Ανάλυση βέλτιστων διαστάσεων στο Breast Cancer Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.96	0.96	0.96	0.96
Average Decision Tree	0.94	0.94	0.94	0.94
Average Random Forest	0.95	0.95	0.95	0.95
Average Naive Bayes	0.93	0.92	0.93	0.92
Average MLP	0.97	0.97	0.97	0.97

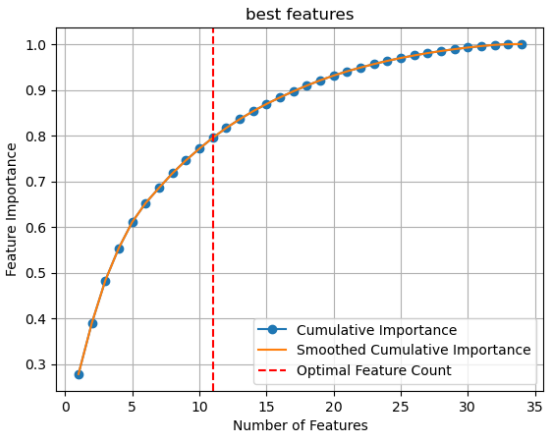
Πίνακας Α′.10: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Kernel PCA Breast Cancer Dataset.

Ionosphere Dataset

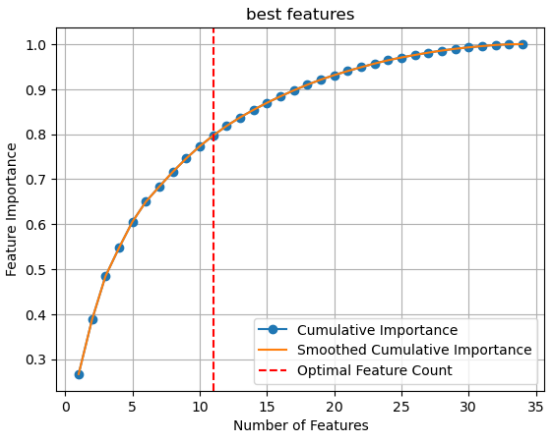


(α′) Fold 1

(β′) Fold 2



(γ′) Fold 3



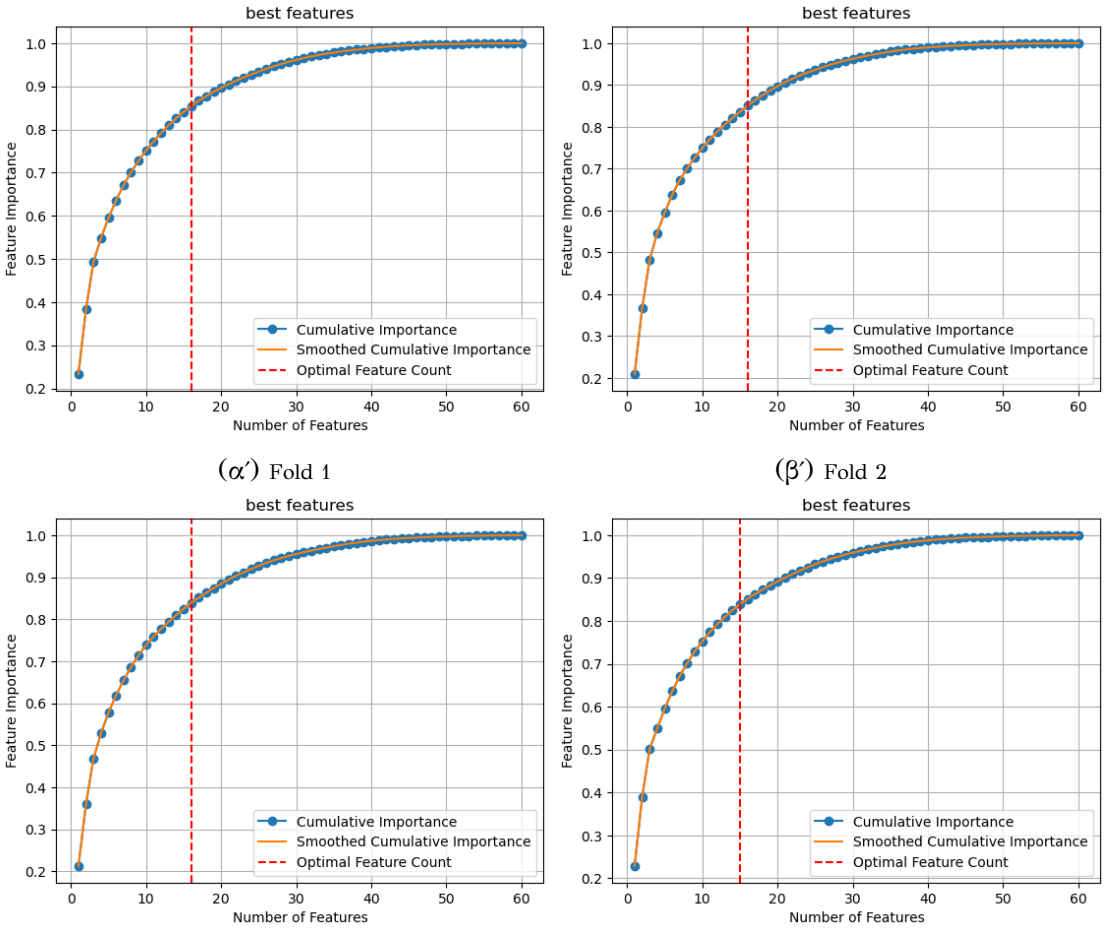
(δ′) Fold 4

Σχήμα Α′.67: Ανάλυση βέλτιστων διαστάσεων στο Ionosphere Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.85	0.83	0.88	0.81
Average SVM	0.93	0.92	0.94	0.91
Average Decision Tree	0.86	0.85	0.85	0.86
Average Random Forest	0.92	0.91	0.91	0.91
Average Naive Bayes	0.87	0.86	0.87	0.86
Average MLP	0.91	0.90	0.90	0.90

Πίνακας Α′.11: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Kernel PCA Ionosphere Dataset.

Connectionist Bench Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

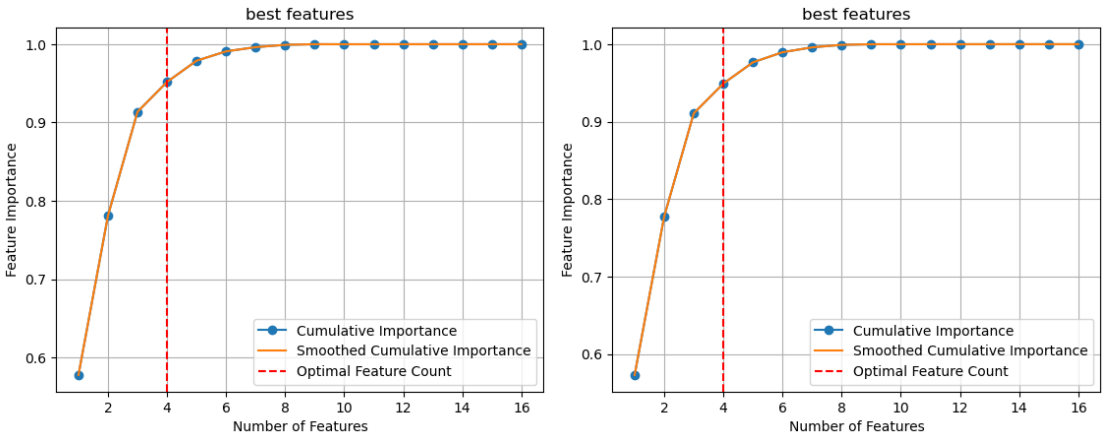
(δ΄) Fold 4

Σχήμα Α΄.68: Ανάλυση βέλτιστων διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.57	0.57	0.58	0.58
Average SVM	0.58	0.57	0.60	0.58
Average Decision Tree	0.54	0.53	0.58	0.55
Average Random Forest	0.69	0.68	0.74	0.70
Average Naive Bayes	0.57	0.56	0.59	0.58
Average MLP	0.64	0.63	0.69	0.65

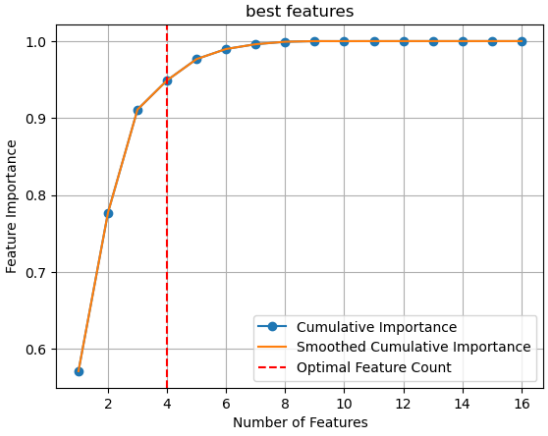
Πίνακας Α΄.12: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Kernel PCA Connectionist Bench Dataset.

Dry Bean Dataset

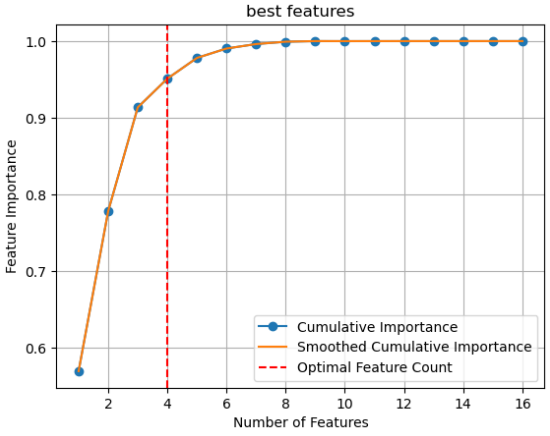


(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3



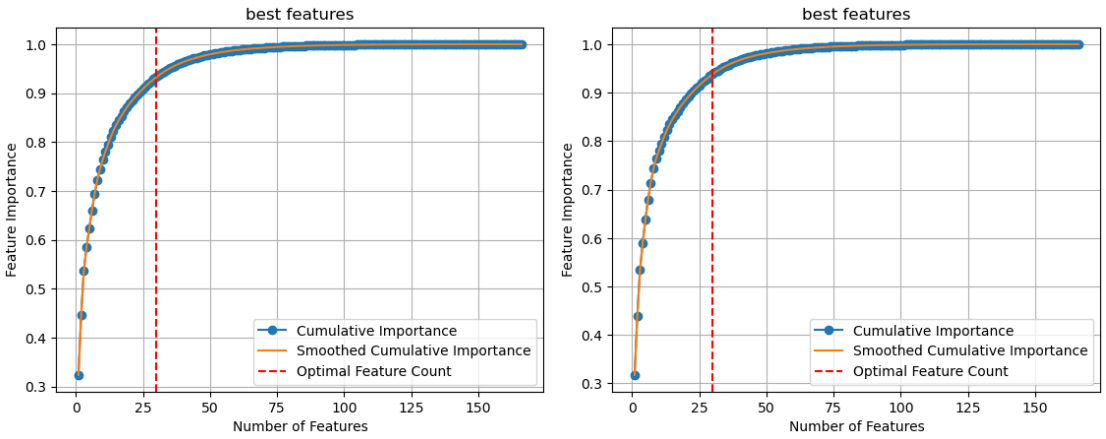
(δ΄) Fold 4

Σχήμα Α΄.69: Ανάλυση βέλτιστων διαστάσεων στο Dry Bean Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.77	0.78	0.82	0.79
Average SVM	0.79	0.80	0.84	0.81
Average Decision Tree	0.70	0.71	0.75	0.71
Average Random Forest	0.76	0.77	0.81	0.78
Average Naive Bayes	0.82	0.83	0.85	0.84
Average MLP	0.73	0.73	0.78	0.75

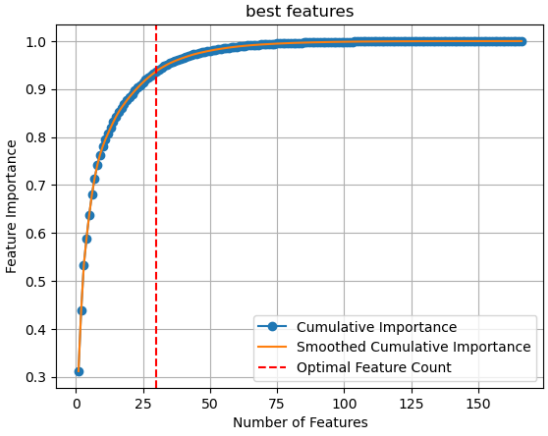
Πίνακας Α΄.13: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Kernel PCA Dry Bean Dataset.

Musk Dataset

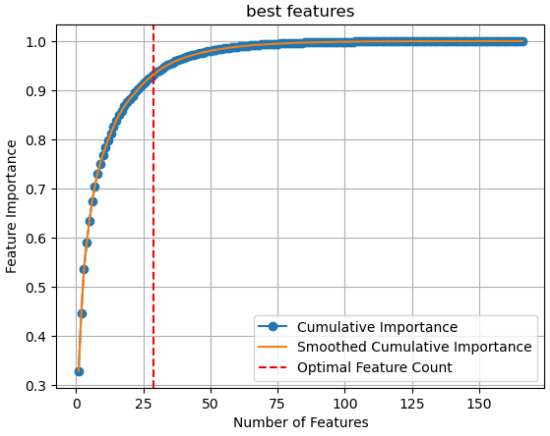


(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3



(δ΄) Fold 4

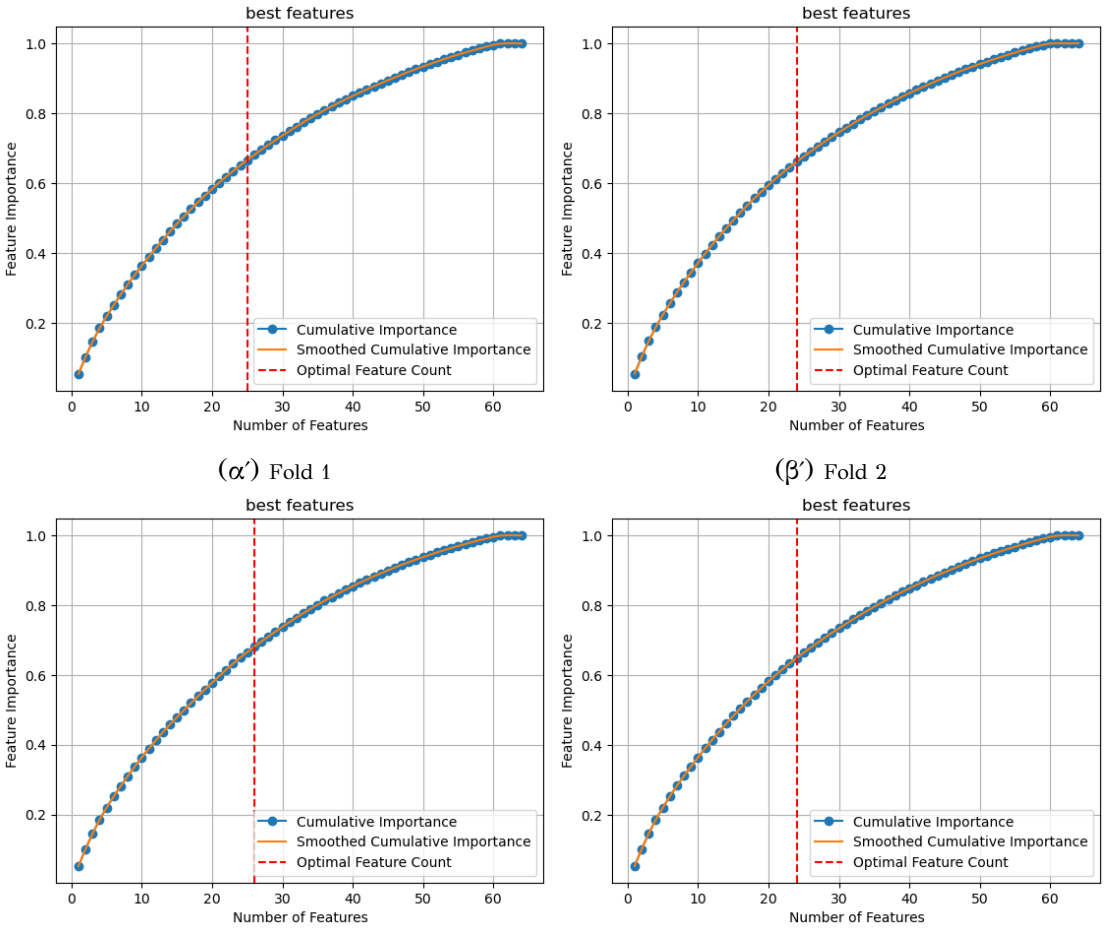
Σχήμα Α΄.70: Ανάλυση βέλτιστων διαστάσεων στο Musk Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.75	0.75	0.76	0.76
Average SVM	0.77	0.76	0.78	0.76
Average Decision Tree	0.71	0.70	0.71	0.71
Average Random Forest	0.74	0.73	0.75	0.73
Average Naive Bayes	0.75	0.74	0.75	0.74
Average MLP	0.80	0.79	0.81	0.80

Πίνακας Α΄.14: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Kernel PCA Musk Dataset.

Α΄.9.3 SVD

Digits Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

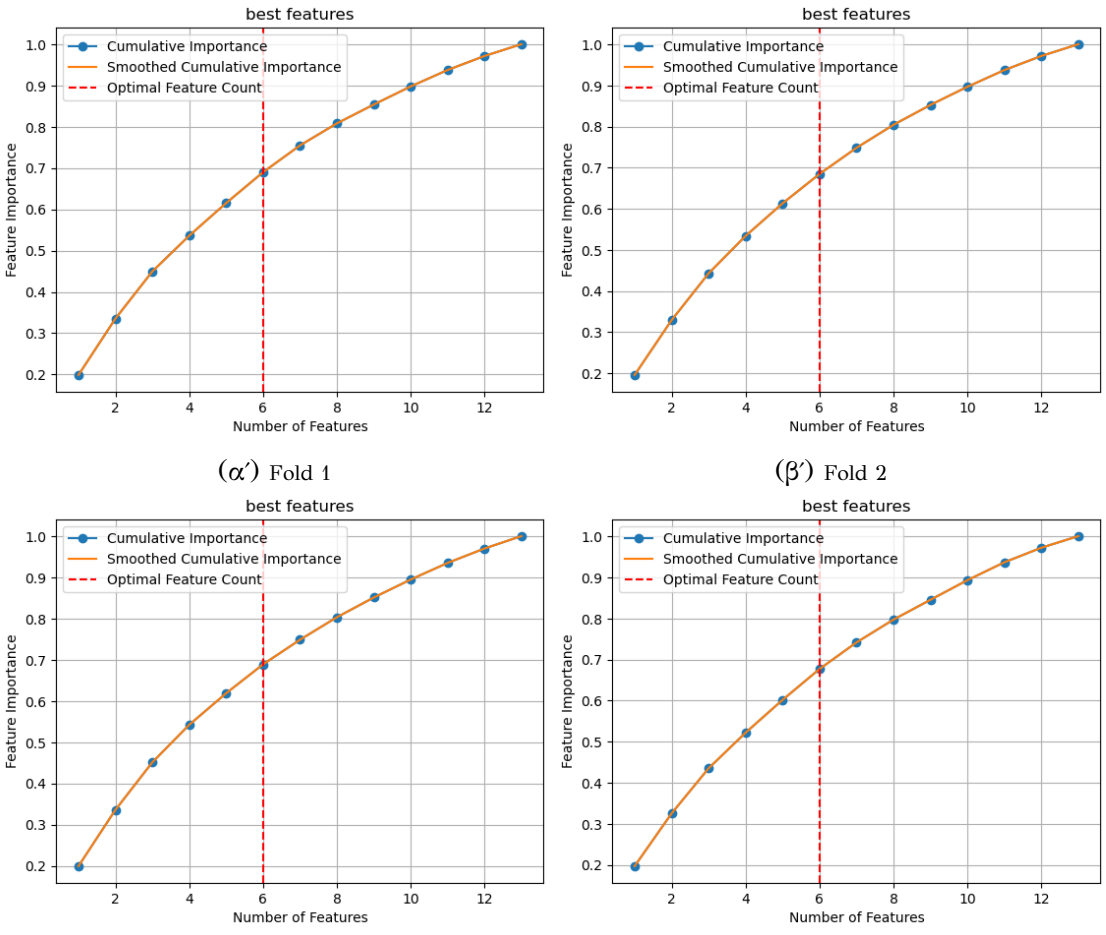
(δ΄) Fold 4

Σχήμα Α΄.71: Ανάλυση βέλτιστων διαστάσεων στο Digits Dataset με τη μέθοδο SVD

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.94	0.94	0.94	0.94
Average SVM	0.95	0.95	0.95	0.95
Average Decision Tree	0.77	0.76	0.77	0.76
Average Random Forest	0.92	0.92	0.92	0.92
Average Naive Bayes	0.85	0.85	0.86	0.85
Average MLP	0.93	0.93	0.94	0.93

Πίνακας Α΄.15: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων SVD Digits Dataset.

Wine Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

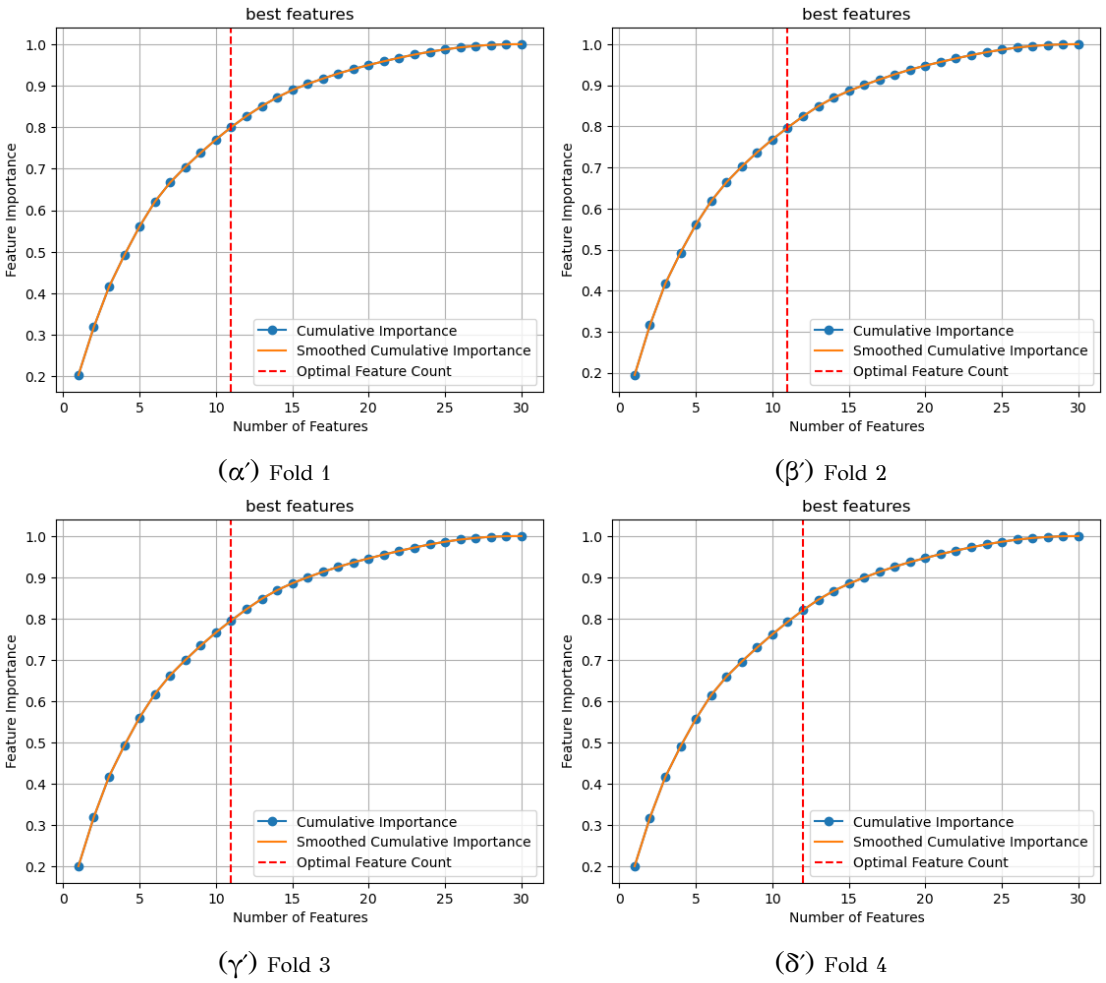
(δ΄) Fold 4

Σχήμα Α΄.72: Ανάλυση βέλτιστων διαστάσεων στο Wine Dataset με τη μέθοδο SVD

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.98	0.98	0.98	0.98
Average Decision Tree	0.92	0.92	0.93	0.93
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.95	0.96	0.96	0.96
Average MLP	0.96	0.96	0.96	0.96

Πίνακας Α΄.16: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων SVD Wine Dataset.

Breast Cancer Dataset



(α′) Fold 1

(β′) Fold 2

(γ′) Fold 3

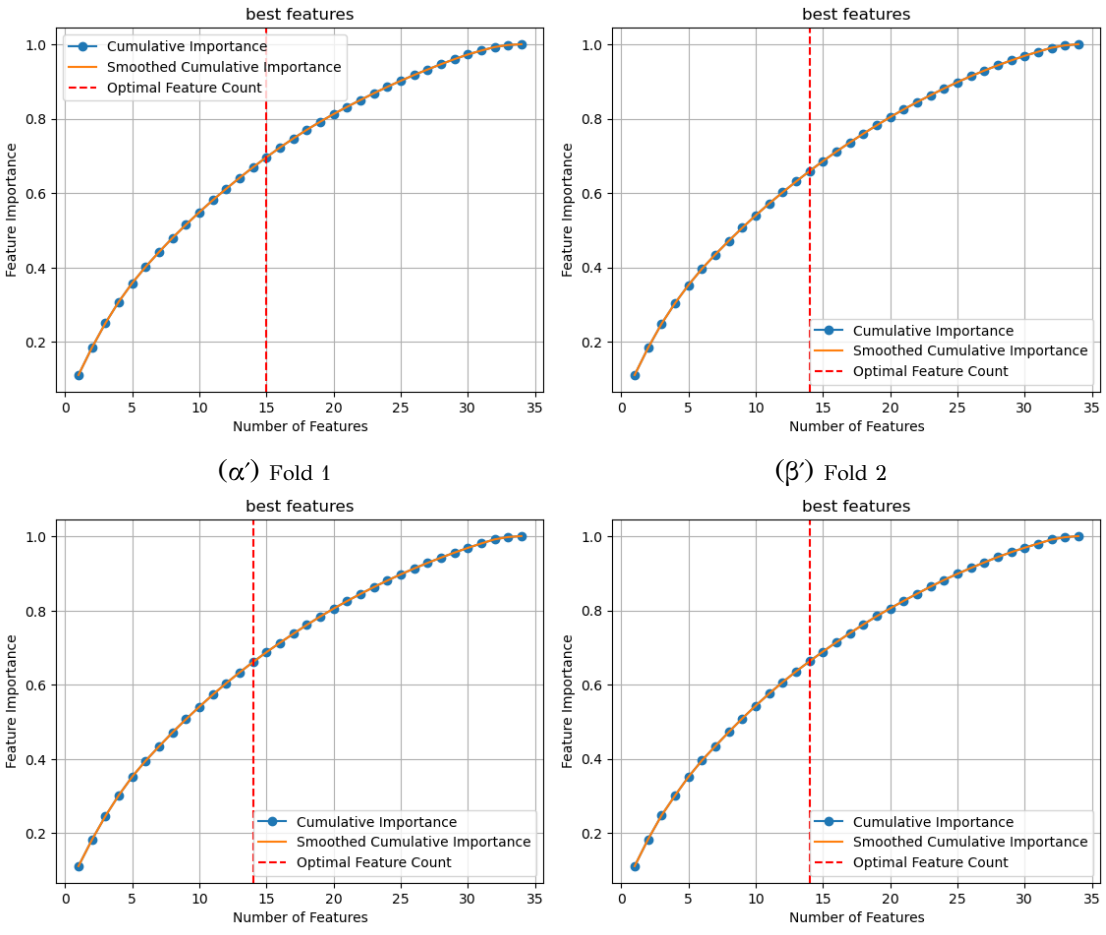
(δ′) Fold 4

Σχήμα Α′.73: Ανάλυση βέλτιστων διαστάσεων στο Breast Cancer Dataset με τη μέθοδο SVD

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.97	0.97	0.97	0.96
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.91	0.91	0.91	0.91
Average Random Forest	0.95	0.94	0.95	0.94
Average Naive Bayes	0.92	0.91	0.92	0.91
Average MLP	0.97	0.97	0.97	0.97

Πίνακας Α′.17: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων SVD Breast Cancer Dataset.

Ionosphere Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

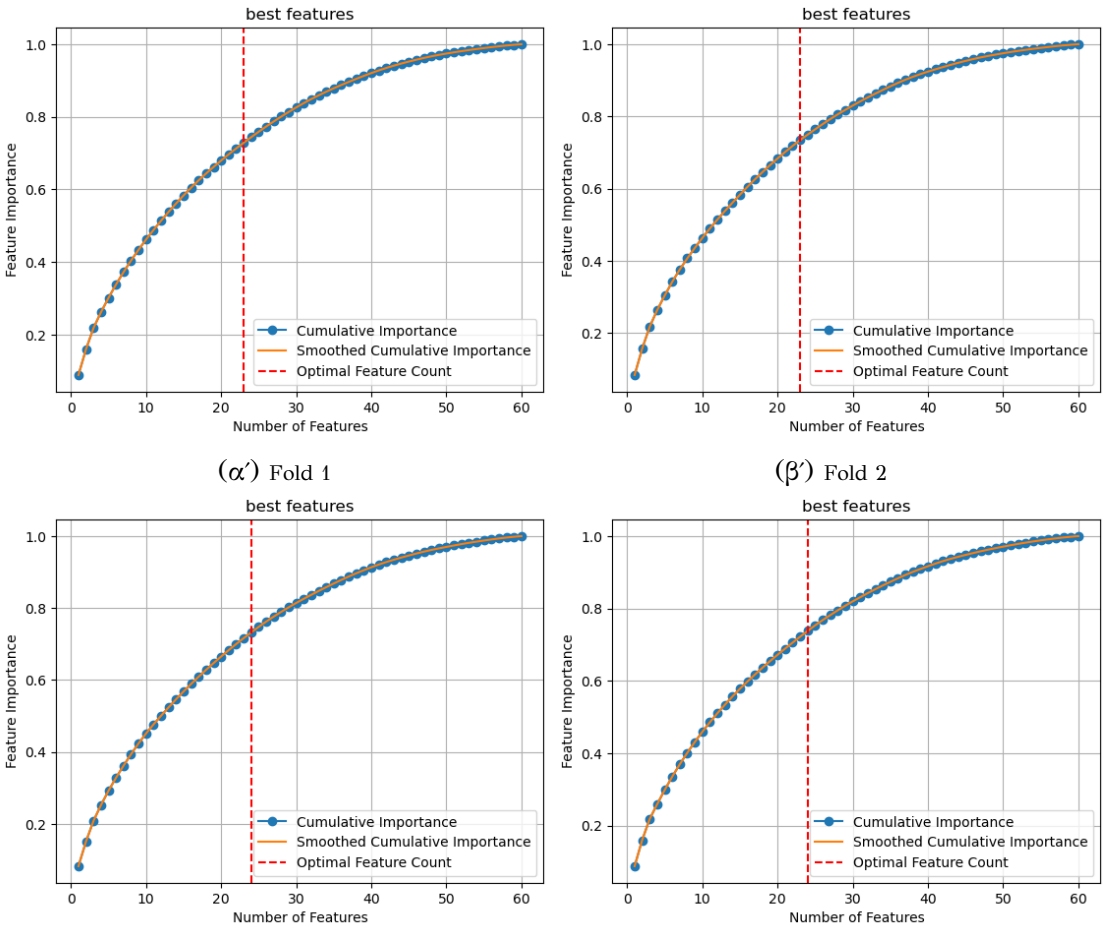
(δ΄) Fold 4

Σχήμα Α΄.74: Ανάλυση βέλτιστων διαστάσεων στο Ionosphere Dataset με τη μέθοδο SVD

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.83	0.79	0.86	0.77
Average SVM	0.94	0.93	0.94	0.92
Average Decision Tree	0.86	0.85	0.85	0.85
Average Random Forest	0.93	0.93	0.93	0.93
Average Naive Bayes	0.86	0.85	0.86	0.85
Average MLP	0.88	0.87	0.88	0.87

Πίνακας Α΄.18: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων SVD Ionosphere Dataset.

Connectionist Bench Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

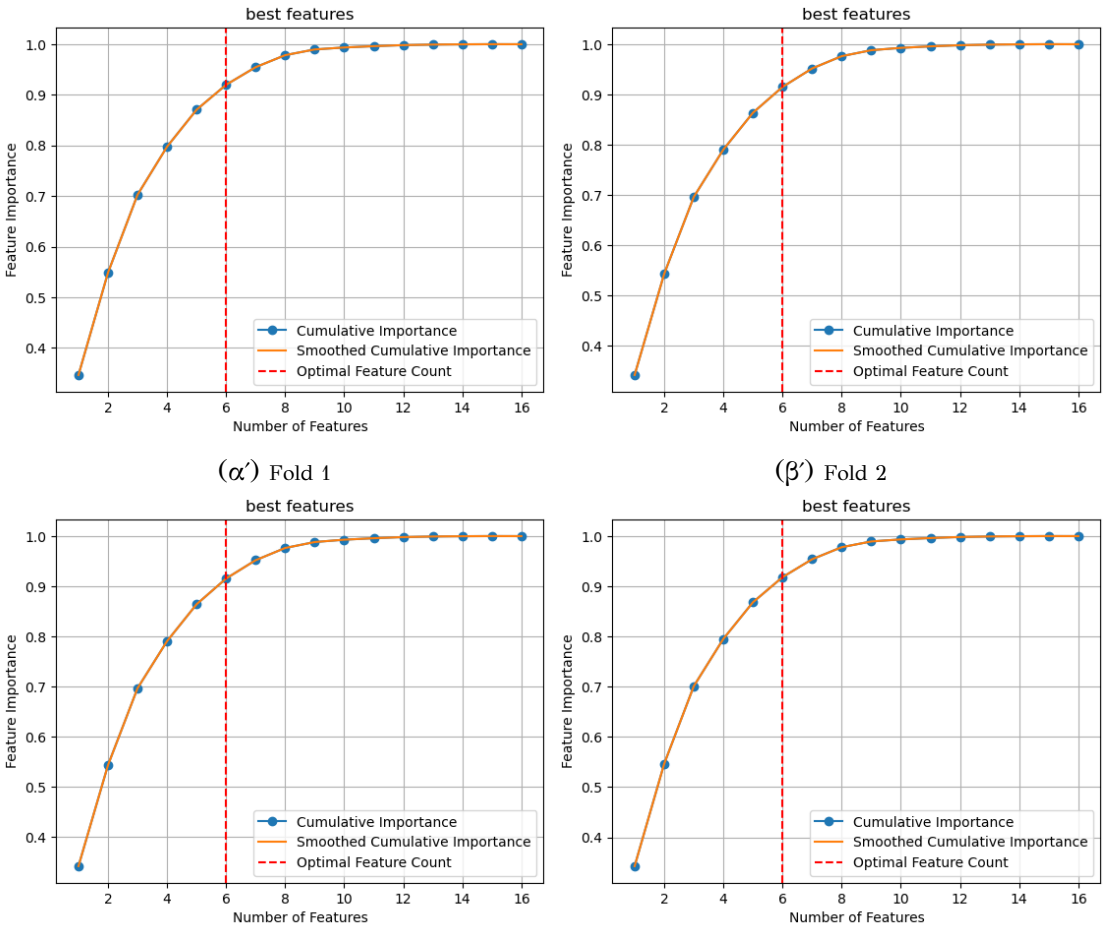
(δ΄) Fold 4

Σχήμα Α΄.75: Ανάλυση βέλτιστων διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο SVD

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.52	0.50	0.51	0.51
Average SVM	0.57	0.56	0.60	0.58
Average Decision Tree	0.56	0.55	0.60	0.57
Average Random Forest	0.62	0.60	0.65	0.62
Average Naive Bayes	0.52	0.51	0.54	0.54
Average MLP	0.68	0.67	0.71	0.68

Πίνακας Α΄.19: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων SVD Connectionist Bench Dataset.

Dry Bean Dataset



(α′) Fold 1 (β′) Fold 2

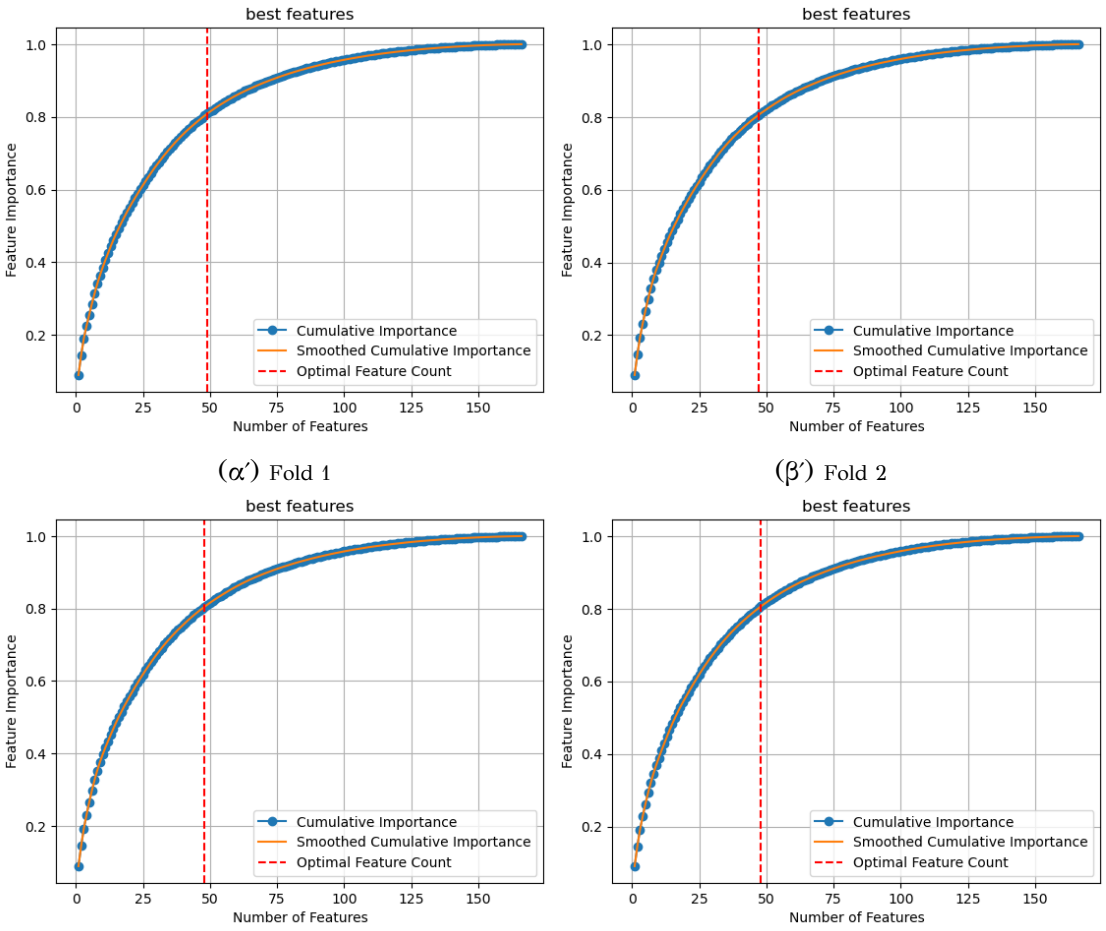
(γ′) Fold 3 (δ′) Fold 4

Σχήμα Α′.76: Ανάλυση βέλτιστων διαστάσεων στο Dry Bean Dataset με τη μέθοδο SVD

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.83	0.85	0.88	0.86
Average SVM	0.83	0.85	0.88	0.86
Average Decision Tree	0.77	0.79	0.83	0.80
Average Random Forest	0.83	0.85	0.88	0.86
Average Naive Bayes	0.85	0.88	0.90	0.88
Average MLP	0.77	0.78	0.81	0.80

Πίνακας Α′.20: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων SVD Dry Bean Dataset.

Musk Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

(δ΄) Fold 4

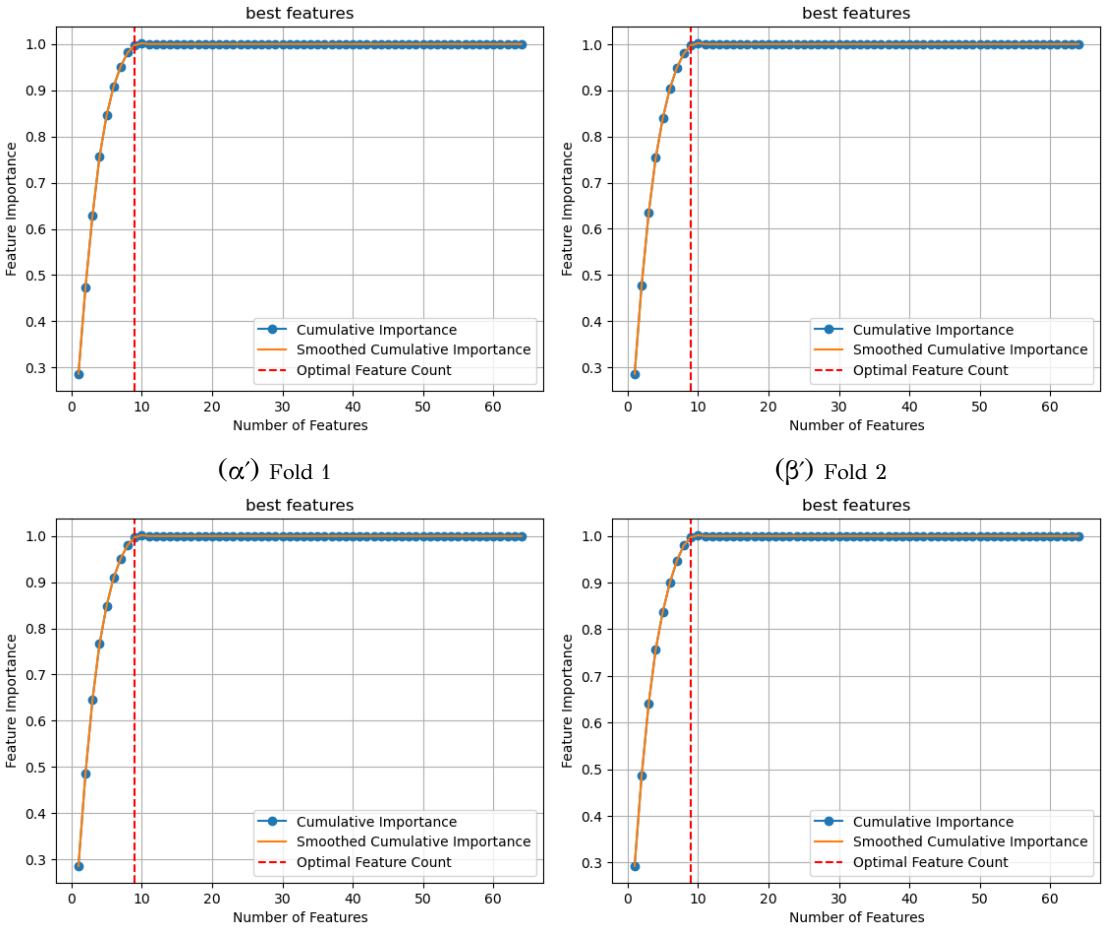
Σχήμα Α΄.77: Ανάλυση βέλτιστων διαστάσεων στο Musk Dataset με τη μέθοδο SVD

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.74	0.74	0.76	0.76
Average SVM	0.76	0.75	0.77	0.75
Average Decision Tree	0.68	0.67	0.68	0.67
Average Random Forest	0.74	0.72	0.75	0.72
Average Naive Bayes	0.72	0.70	0.73	0.71
Average MLP	0.76	0.76	0.77	0.76

Πίνακας Α΄.21: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων SVD Musk Dataset.

Α΄.9.4 LDA

Digits Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

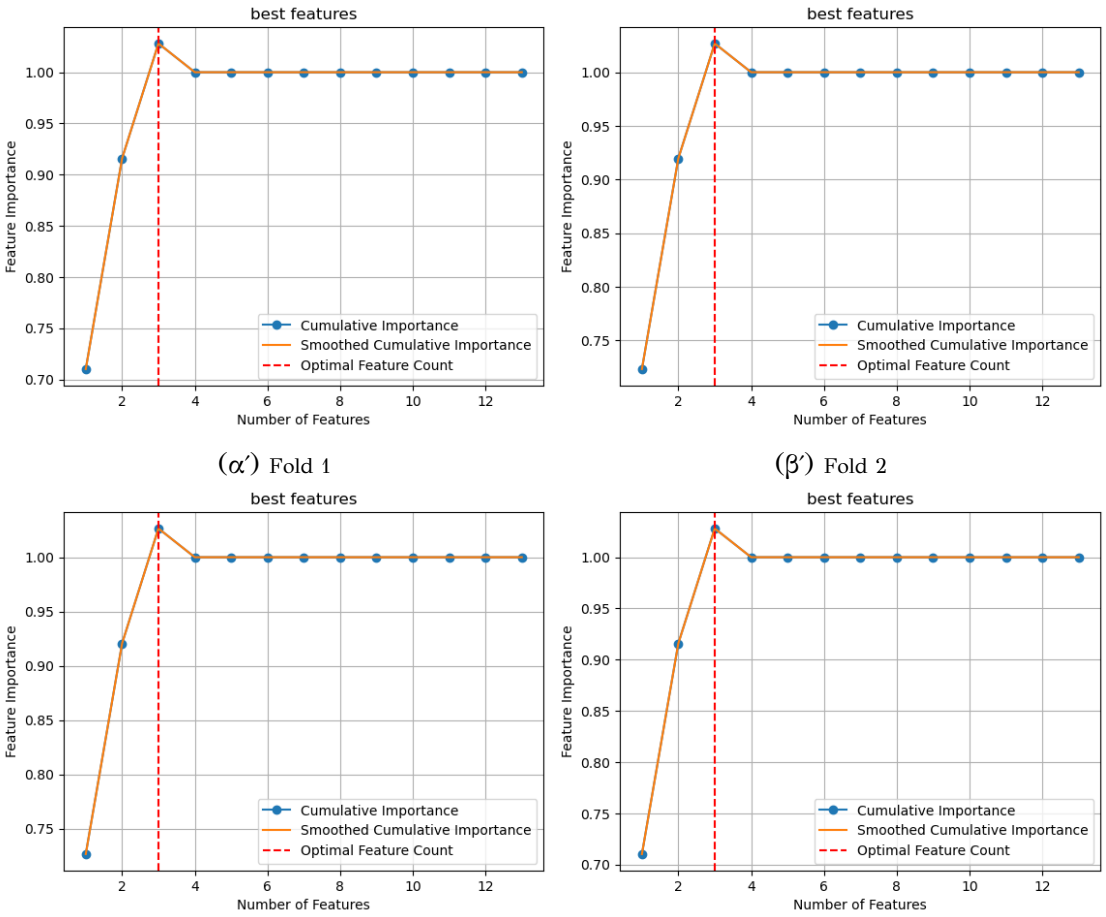
(δ΄) Fold 4

Σχήμα Α΄.78: Ανάλυση βέλτιστων διαστάσεων στο Digits Dataset με τη μέθοδο LDA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.93	0.93	0.94	0.93
Average SVM	0.93	0.93	0.93	0.93
Average Decision Tree	0.83	0.83	0.83	0.83
Average Random Forest	0.92	0.92	0.92	0.92
Average Naive Bayes	0.92	0.92	0.93	0.92
Average MLP	0.91	0.91	0.91	0.91

Πίνακας Α΄.22: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LDA Digits Dataset.

Wine Dataset



(α΄) Fold 1 (β΄) Fold 2

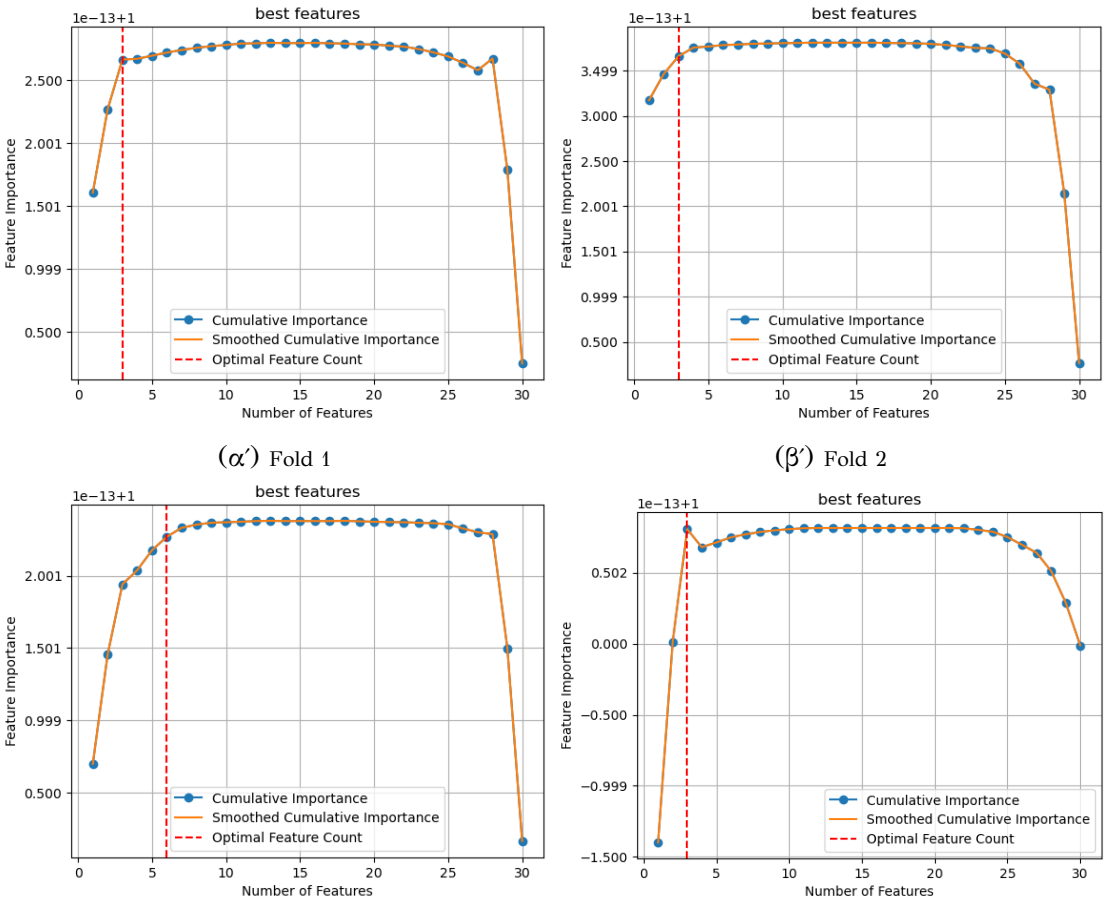
(γ΄) Fold 3 (δ΄) Fold 4

Σχήμα Α΄.79: Ανάλυση βέλτιστων διαστάσεων στο Wine Dataset με τη μέθοδο LDA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.97	0.97	0.97	0.98
Average SVM	0.97	0.97	0.97	0.98
Average Decision Tree	0.92	0.92	0.94	0.93
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.97	0.97	0.97	0.97
Average MLP	0.97	0.97	0.97	0.97

Πίνακας Α΄.23: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LDA Wine Dataset.

Breast Cancer Dataset



(α′) Fold 1

(β′) Fold 2

(γ′) Fold 3

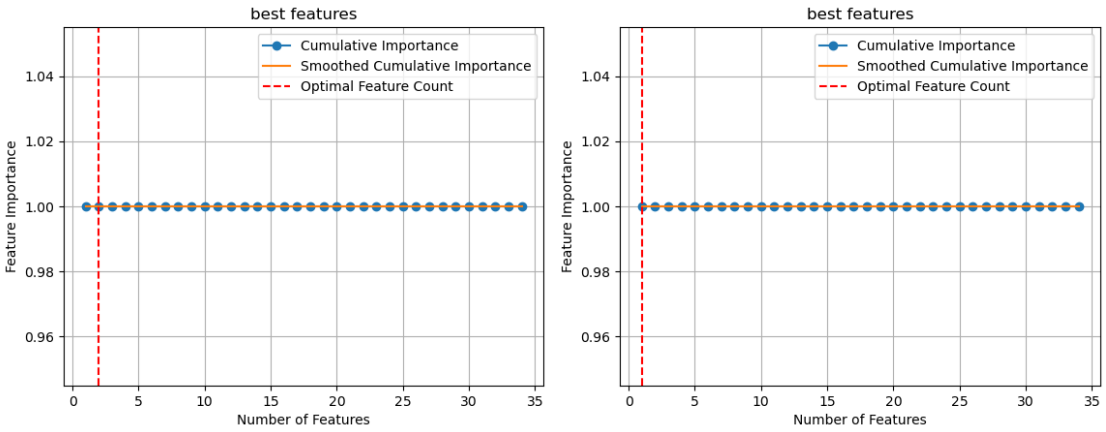
(δ′) Fold 4

Σχήμα Α′.80: Ανάλυση βέλτιστων διαστάσεων στο Breast Cancer Dataset με τη μέθοδο LDA.

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.97	0.97	0.97	0.96
Average Decision Tree	0.96	0.96	0.96	0.96
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.97	0.97	0.97	0.96
Average MLP	0.96	0.96	0.96	0.96

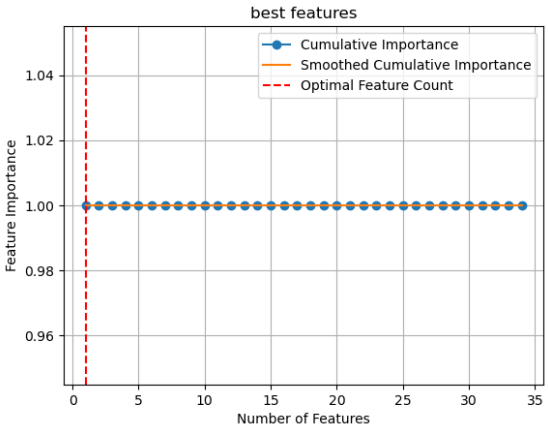
Πίνακας Α′.24: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LDA Breast Cancer Dataset.

Ionosphere Dataset

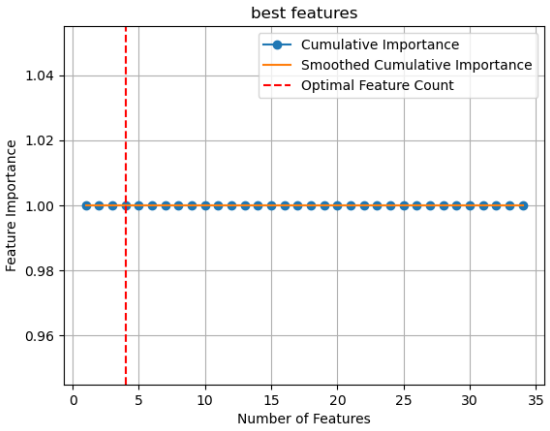


(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3



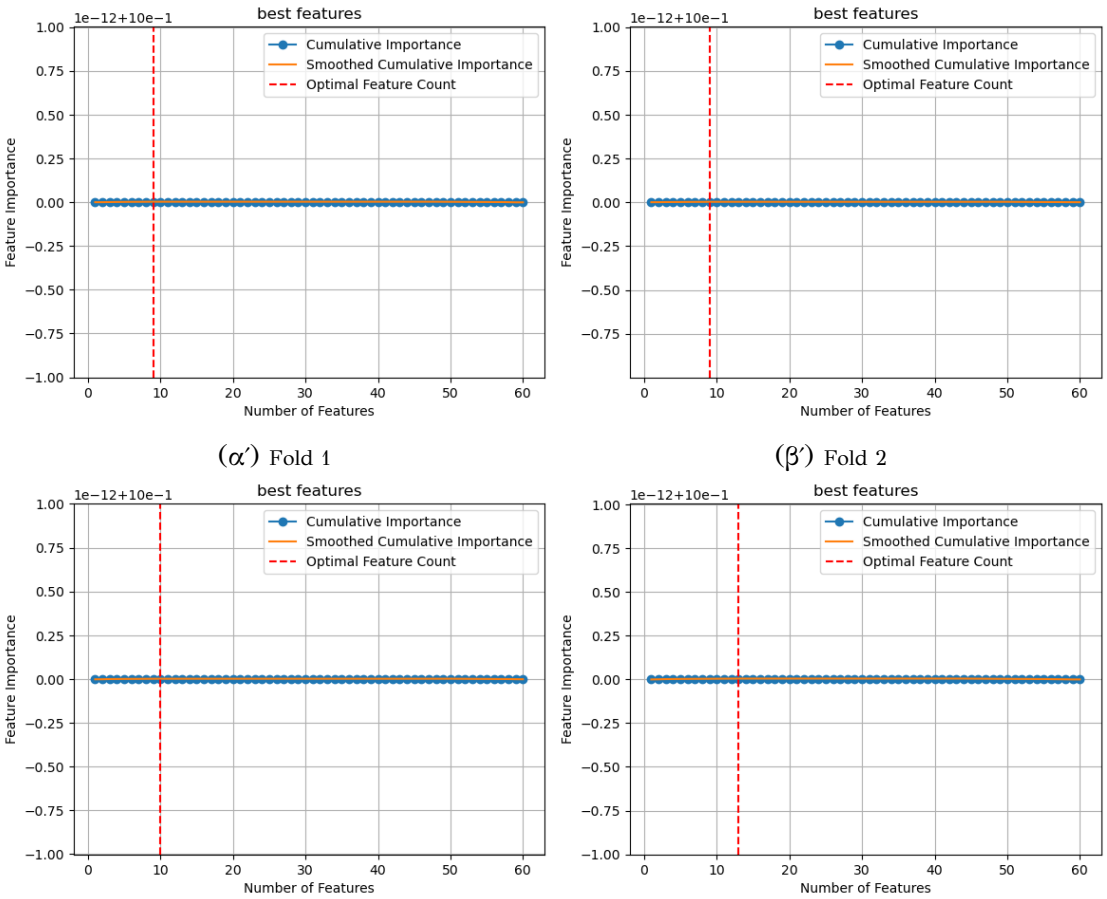
(δ΄) Fold 4

Σχήμα Α΄.81: Ανάλυση βέλτιστων διαστάσεων στο Ionosphere Dataset με τη μέθοδο LDA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.85	0.82	0.87	0.80
Average SVM	0.84	0.81	0.86	0.80
Average Decision Tree	0.82	0.80	0.82	0.79
Average Random Forest	0.82	0.80	0.82	0.79
Average Naive Bayes	0.83	0.80	0.84	0.79
Average MLP	0.84	0.81	0.86	0.80

Πίνακας Α΄.25: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LDA Ionosphere Dataset.

Connectionist Bench Dataset



(α′) Fold 1

(β′) Fold 2

(γ′) Fold 3

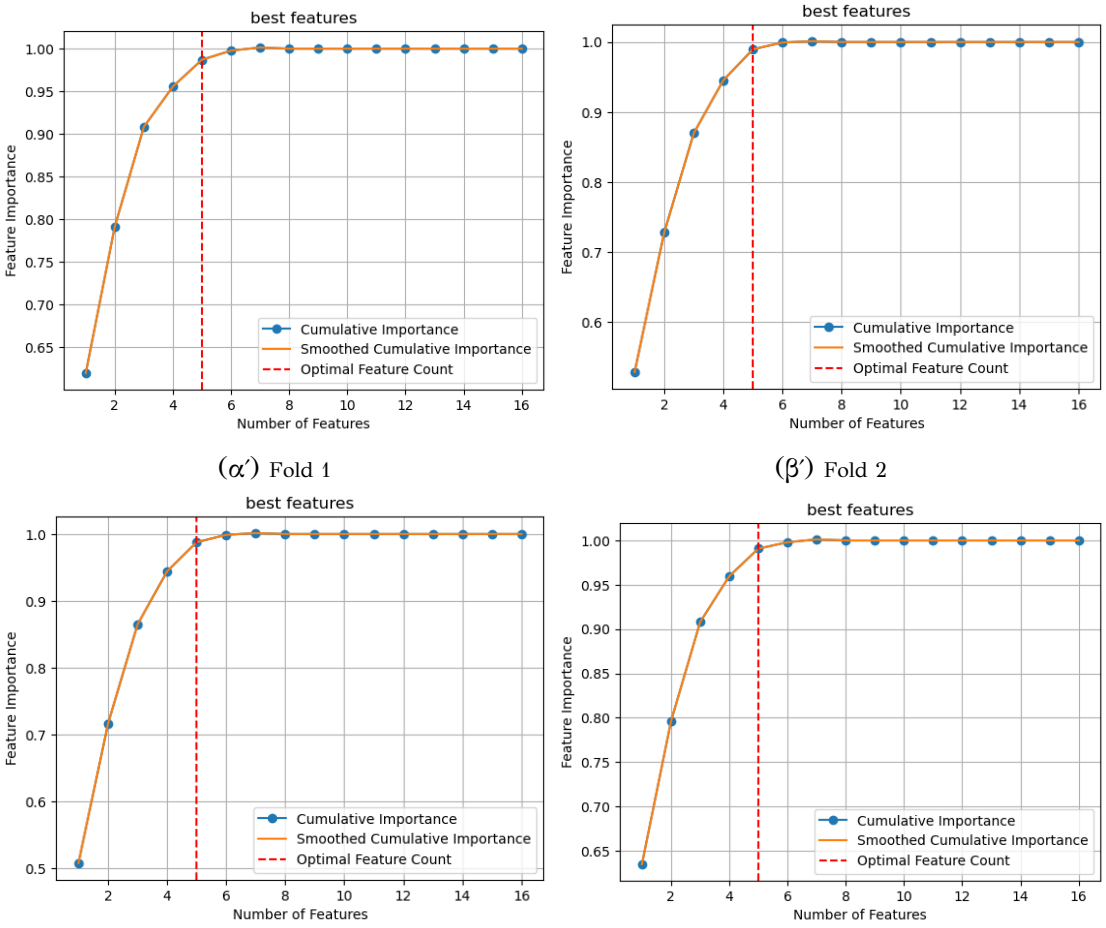
(δ′) Fold 4

Σχήμα Α′.82: Ανάλυση βέλτιστων διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο LDA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.55	0.54	0.56	0.55
Average SVM	0.54	0.53	0.55	0.54
Average Decision Tree	0.57	0.56	0.58	0.57
Average Random Forest	0.57	0.56	0.58	0.57
Average Naive Bayes	0.55	0.54	0.56	0.55
Average MLP	0.56	0.55	0.57	0.56

Πίνακας Α′.26: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LDA Connectionist Bench Dataset.

Dry Bean Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

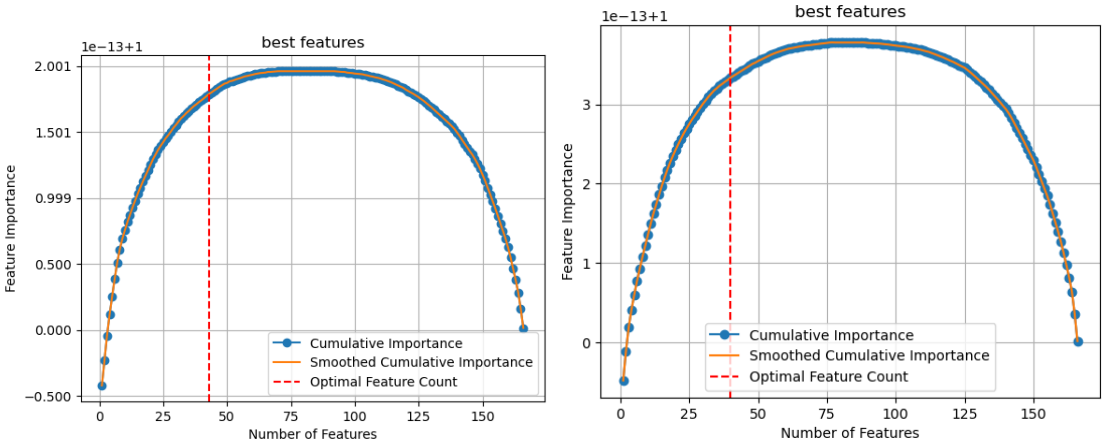
(δ΄) Fold 4

Σχήμα Α΄.83: Ανάλυση βέλτιστων διαστάσεων στο Dry Bean Dataset με τη μέθοδο LDA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.76	0.78	0.82	0.79
Average SVM	0.82	0.83	0.86	0.84
Average Decision Tree	0.71	0.73	0.76	0.75
Average Random Forest	0.74	0.76	0.80	0.78
Average Naive Bayes	0.85	0.87	0.89	0.88
Average MLP	0.81	0.83	0.86	0.84

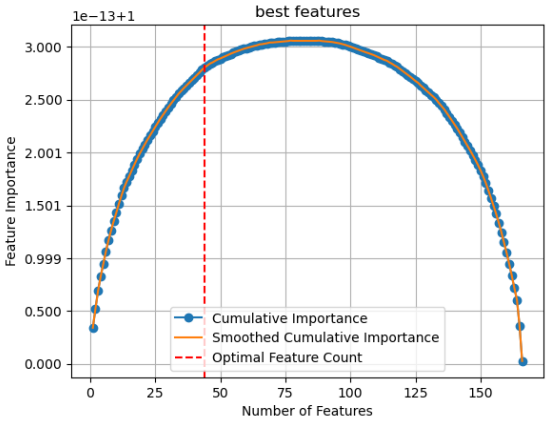
Πίνακας Α΄.27: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LDA Dry Bean Dataset.

Musk Dataset

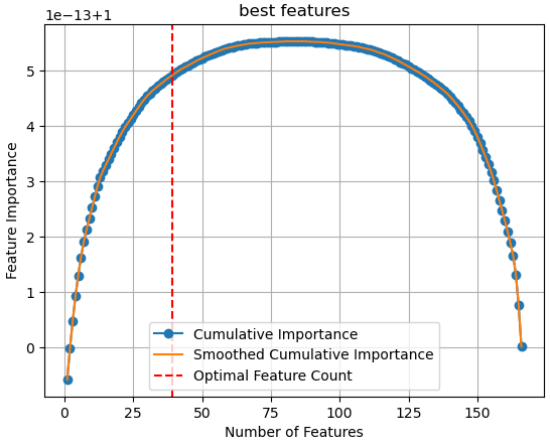


(α′) Fold 1

(β′) Fold 2



(γ′) Fold 3



(δ′) Fold 4

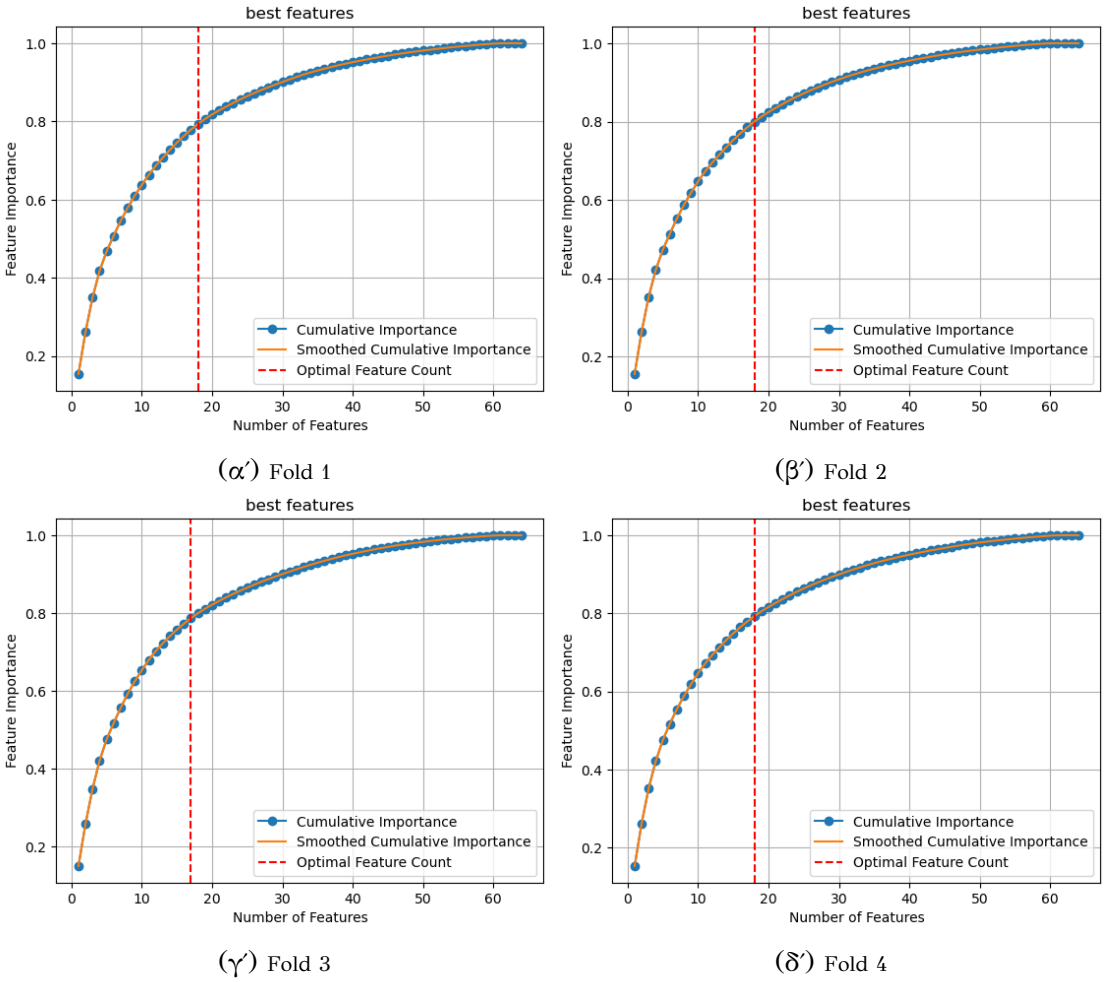
Σχήμα Α′.84: Ανάλυση βέλτιστων διαστάσεων στο Musk Dataset με τη μέθοδο LDA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.63	0.63	0.64	0.64
Average SVM	0.64	0.63	0.65	0.64
Average Decision Tree	0.64	0.63	0.64	0.64
Average Random Forest	0.64	0.63	0.64	0.64
Average Naive Bayes	0.63	0.63	0.64	0.63
Average MLP	0.63	0.63	0.64	0.64

Πίνακας Α′.28: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LDA Musk Dataset.

Α΄.9.5 Factor Analysis

Digits Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

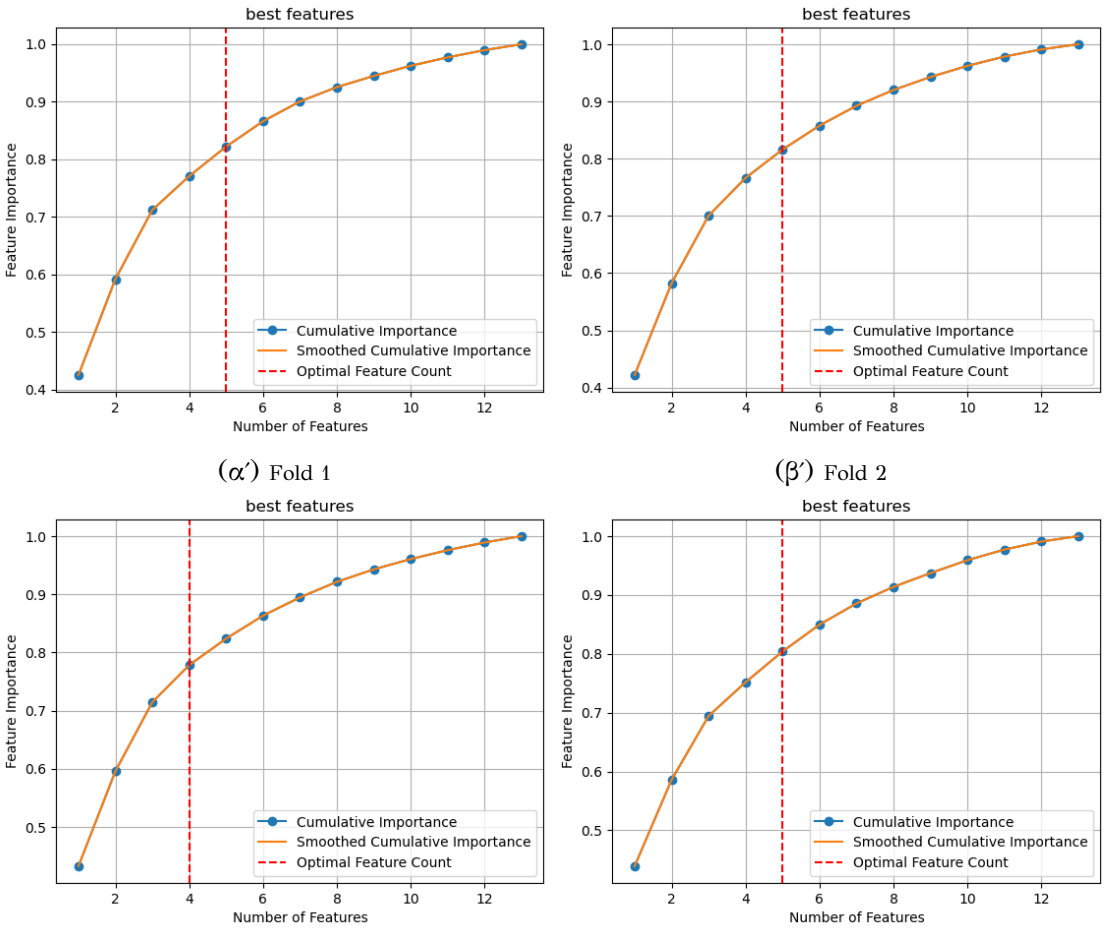
(δ΄) Fold 4

Σχήμα Α΄.85: Ανάλυση βέλτιστων διαστάσεων στο Digits Dataset με τη μέθοδο Factor Analysis

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.92	0.92	0.92	0.92
Average SVM	0.94	0.94	0.94	0.94
Average Decision Tree	0.78	0.78	0.78	0.78
Average Random Forest	0.91	0.91	0.92	0.91
Average Naive Bayes	0.86	0.86	0.87	0.86
Average MLP	0.93	0.93	0.93	0.93

Πίνακας Α΄.29: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Factor Analysis Digits Dataset.

Wine Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

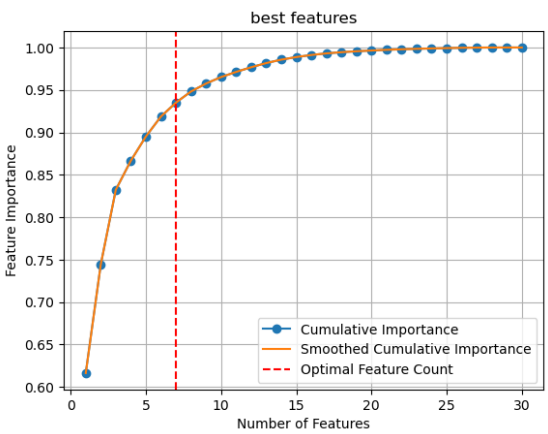
(δ΄) Fold 4

Σχήμα Α΄.86: Ανάλυση βέλτιστων διαστάσεων στο Wine Dataset με τη μέθοδο Factor Analysis

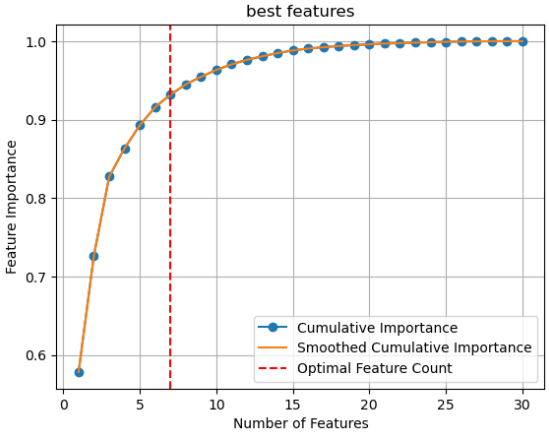
Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.98	0.98	0.98	0.98
Average Decision Tree	0.93	0.93	0.94	0.94
Average Random Forest	0.97	0.97	0.97	0.98
Average Naive Bayes	0.98	0.98	0.98	0.98
Average MLP	0.97	0.97	0.98	0.97

Πίνακας Α΄.30: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Factor Analysis Wine Dataset.

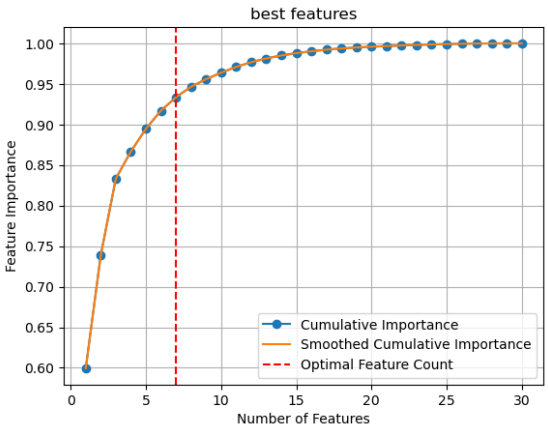
Breast Cancer Dataset



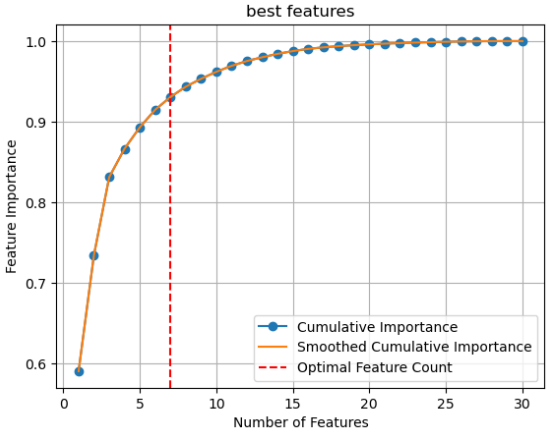
(α΄) Fold 1



(β΄) Fold 2



(γ΄) Fold 3



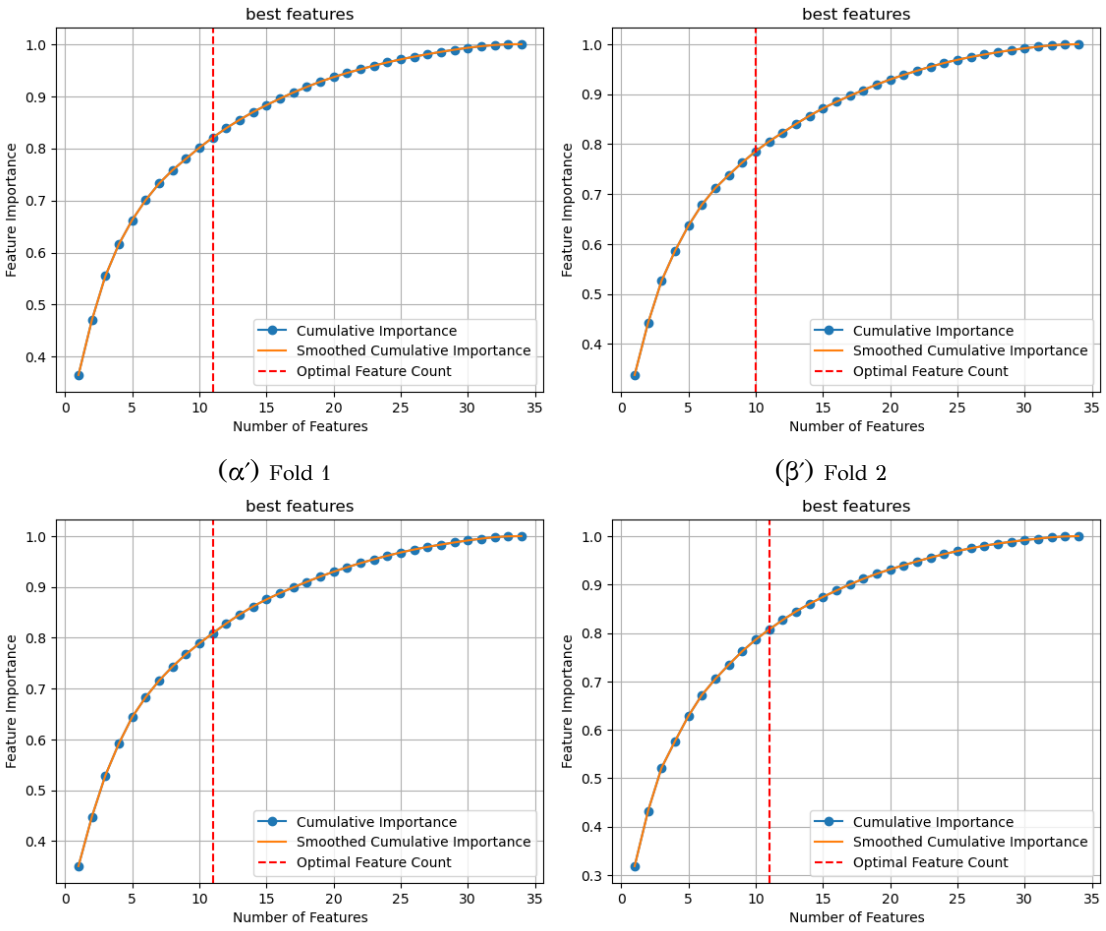
(δ΄) Fold 4

Σχήμα Α΄.87: Ανάλυση βέλτιστων διαστάσεων στο Breast Cancer Dataset με τη μέθοδο Factor Analysis

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.94	0.93	0.95	0.92
Average SVM	0.97	0.97	0.97	0.96
Average Decision Tree	0.92	0.91	0.91	0.92
Average Random Forest	0.95	0.94	0.95	0.94
Average Naive Bayes	0.90	0.90	0.90	0.90
Average MLP	0.97	0.97	0.97	0.97

Πίνακας Α΄.31: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Factor Analysis Breast Cancer Datase

Ionosphere Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

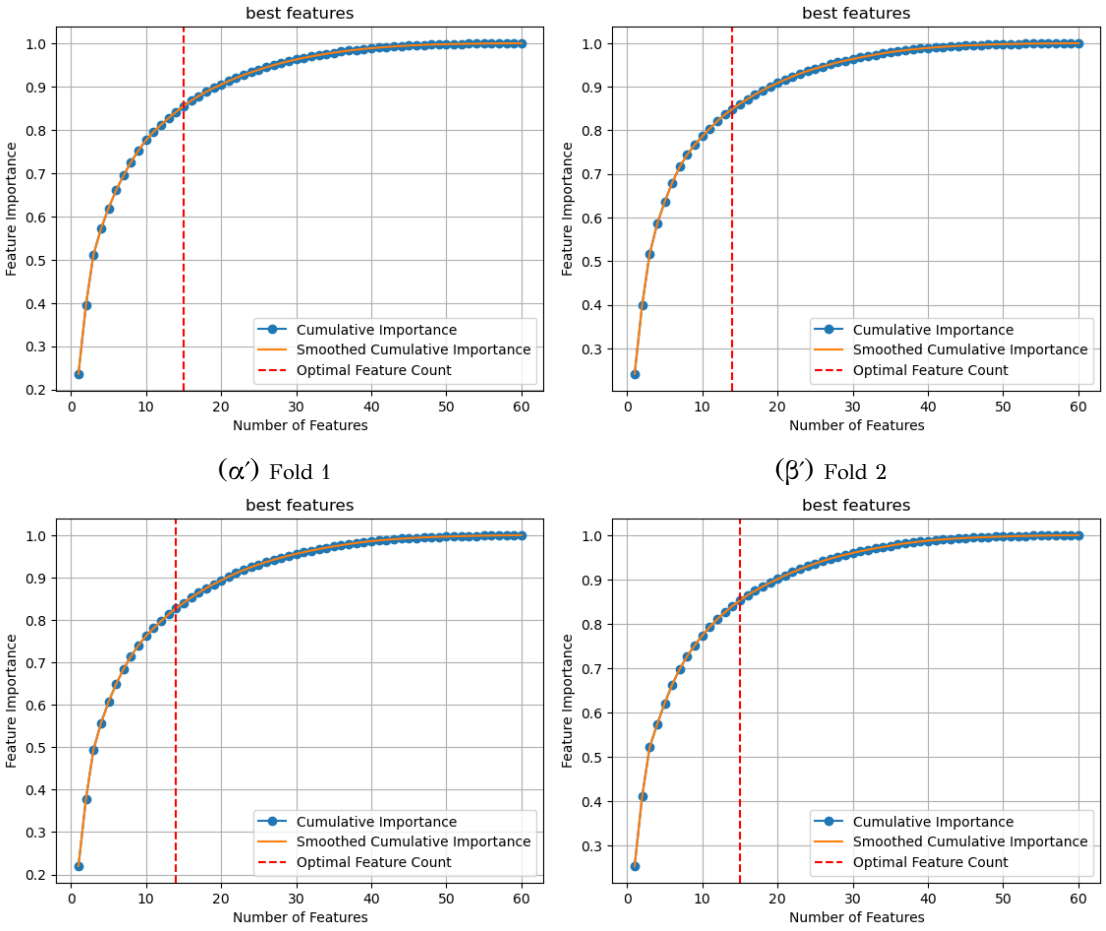
(δ΄) Fold 4

Σχήμα Α΄.88: Ανάλυση βέλτιστων διαστάσεων στο Ionosphere Dataset με τη μέθοδο Factor Analysis

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.85	0.82	0.88	0.80
Average SVM	0.94	0.93	0.95	0.92
Average Decision Tree	0.88	0.87	0.88	0.86
Average Random Forest	0.94	0.93	0.94	0.93
Average Naive Bayes	0.83	0.81	0.83	0.81
Average MLP	0.93	0.93	0.93	0.92

Πίνακας Α΄.32: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Factor Analysis Ionosphere Dataset.

Connectionist Bench Dataset



(α′) Fold 1

(β′) Fold 2

(γ′) Fold 3

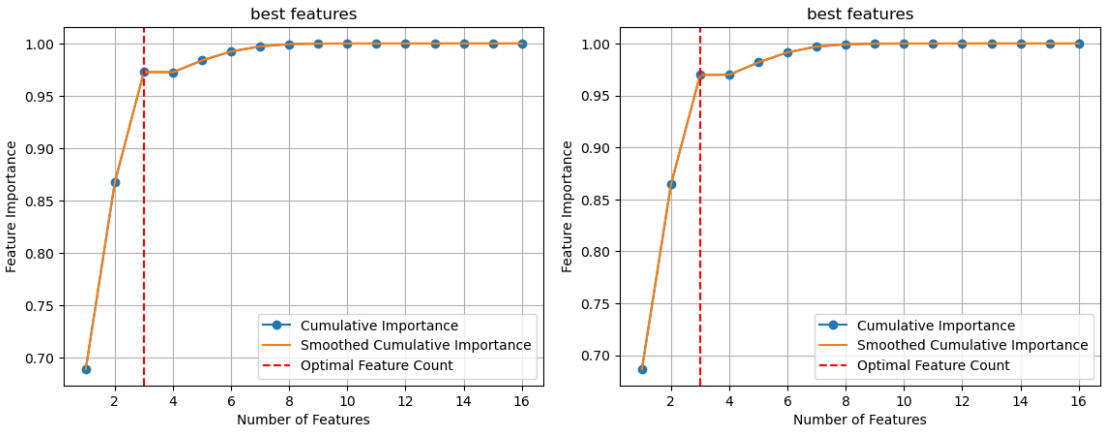
(δ′) Fold 4

Σχήμα Α′.89: Ανάλυση βέλτιστων διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο Factor Analysis

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.48	0.47	0.47	0.48
Average SVM	0.59	0.57	0.62	0.59
Average Decision Tree	0.59	0.57	0.61	0.59
Average Random Forest	0.63	0.61	0.68	0.63
Average Naive Bayes	0.56	0.55	0.58	0.56
Average MLP	0.63	0.61	0.66	0.63

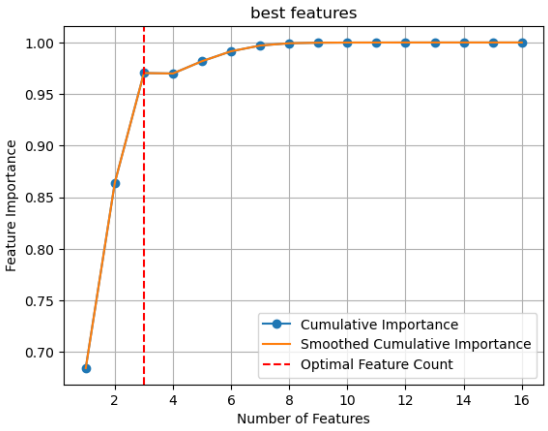
Πίνακας Α′.33: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Factor Analysis Connectionist Bench Dataset.

Dry Bean Dataset

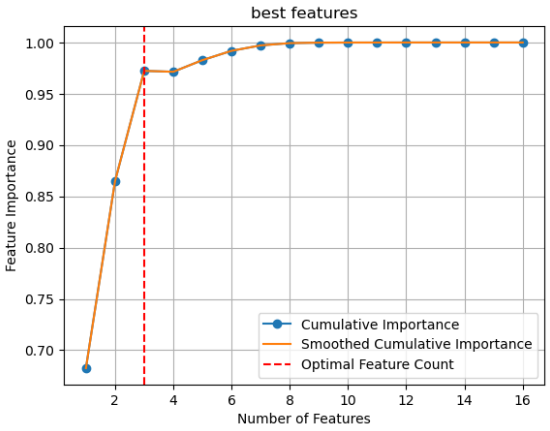


(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3



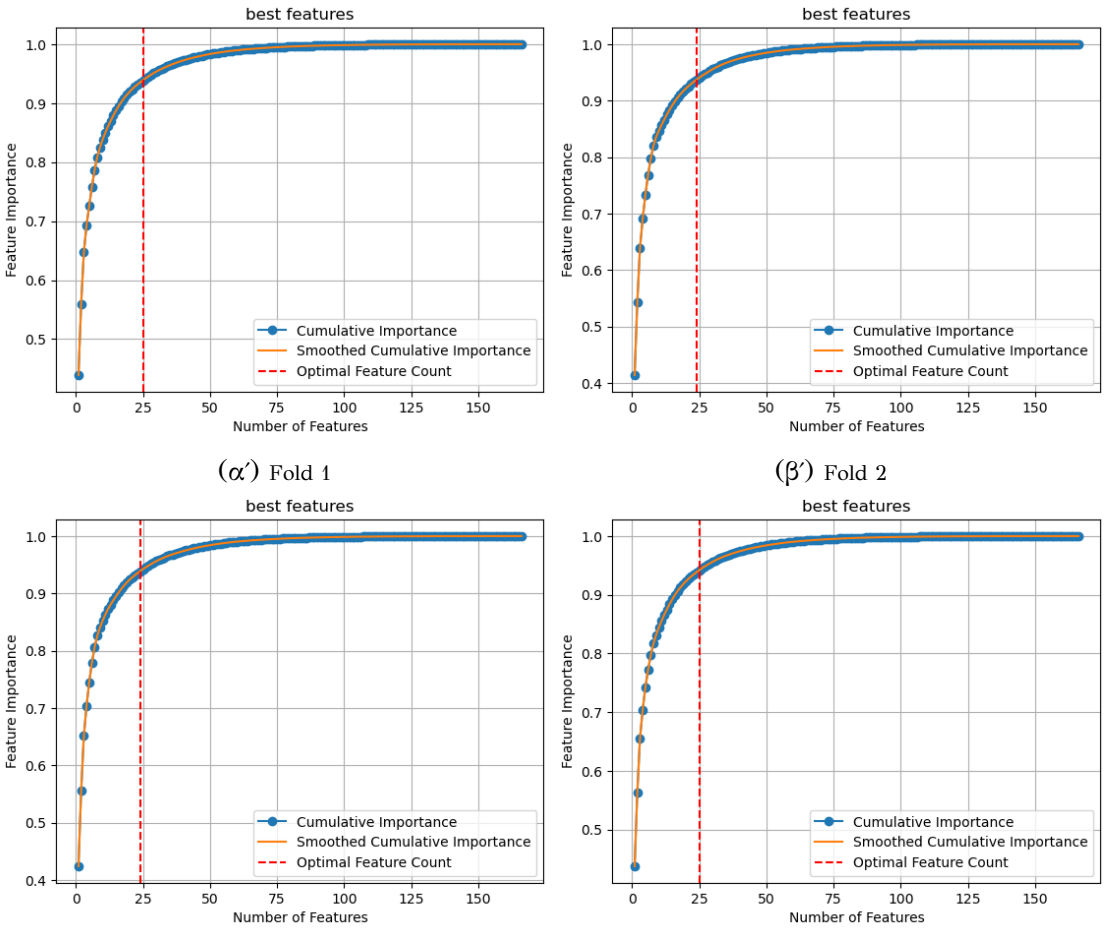
(δ΄) Fold 4

Σχήμα Α΄.90: Ανάλυση βέλτιστων διαστάσεων στο Dry Bean Dataset με τη μέθοδο Factor Analysis

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.59	0.61	0.68	0.63
Average SVM	0.76	0.77	0.82	0.78
Average Decision Tree	0.44	0.45	0.51	0.48
Average Random Forest	0.50	0.52	0.60	0.54
Average Naive Bayes	0.80	0.82	0.85	0.82
Average MLP	0.64	0.65	0.73	0.67

Πίνακας Α΄.34: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Factor Analysis Dry Bean Dataset.

Musk Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

(δ΄) Fold 4

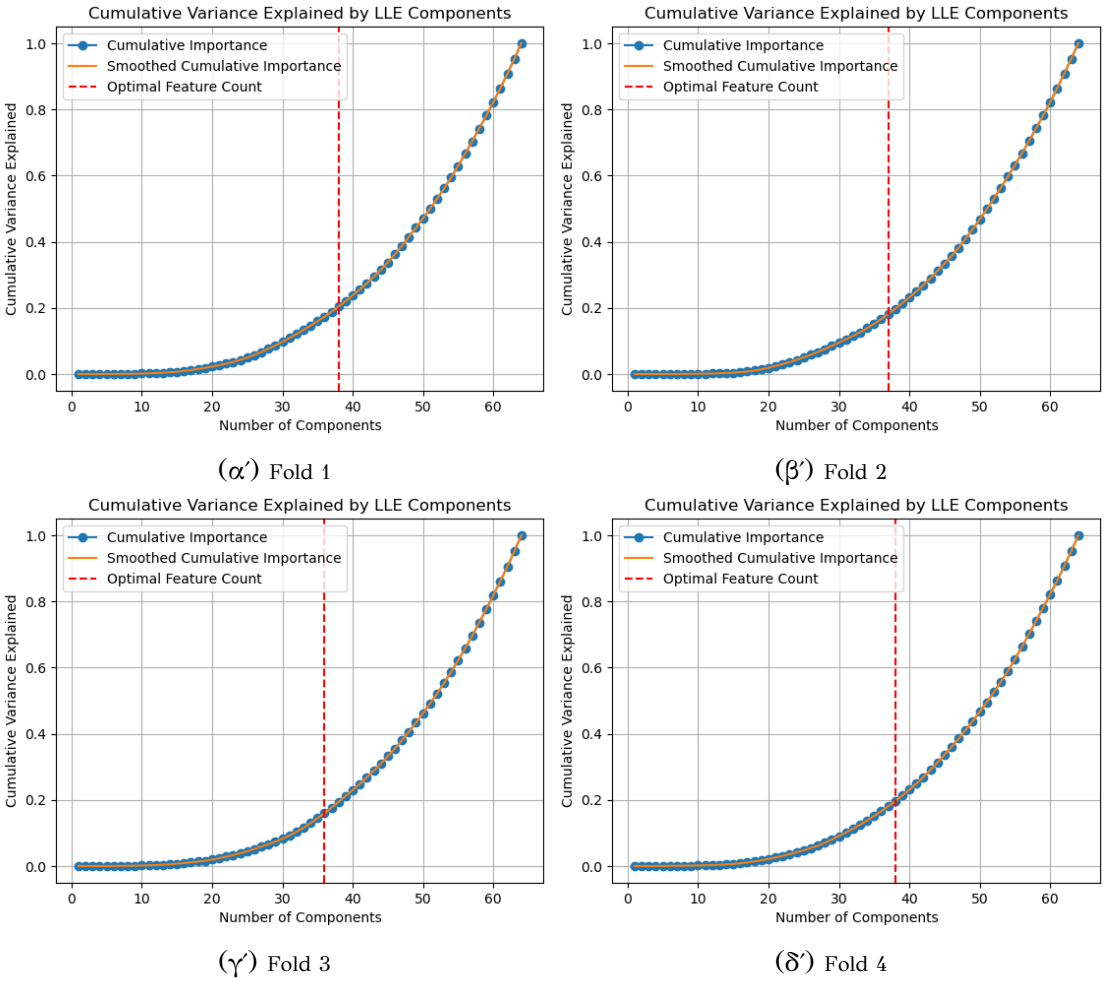
Σχήμα Α΄.91: Ανάλυση βέλτιστων διαστάσεων στο Musk Dataset με τη μέθοδο Factor Analysis

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.70	0.69	0.71	0.70
Average SVM	0.73	0.73	0.73	0.73
Average Decision Tree	0.60	0.60	0.60	0.60
Average Random Forest	0.70	0.69	0.70	0.69
Average Naive Bayes	0.68	0.68	0.68	0.68
Average MLP	0.71	0.71	0.72	0.72

Πίνακας Α΄.35: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Factor Analysis Musk Dataset.

Α΄.9.6 LLE

Digits Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

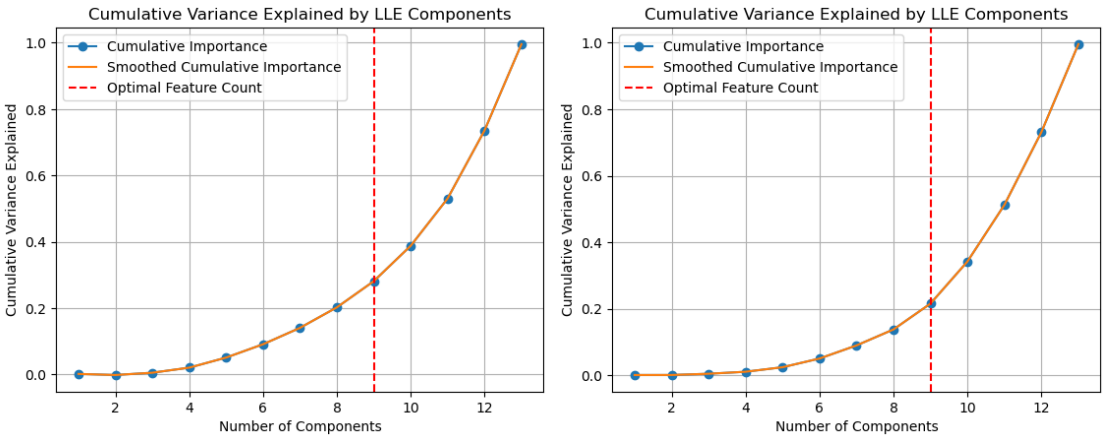
(δ΄) Fold 4

Σχήμα Α΄.92: Ανάλυση βέλτιστων διαστάσεων στο Digits Dataset με τη μέθοδο LLE

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.91	0.91	0.91	0.91
Average SVM	0.92	0.92	0.92	0.92
Average Decision Tree	0.85	0.85	0.85	0.84
Average Random Forest	0.91	0.91	0.92	0.91
Average Naive Bayes	0.88	0.88	0.88	0.88
Average MLP	0.90	0.90	0.91	0.90

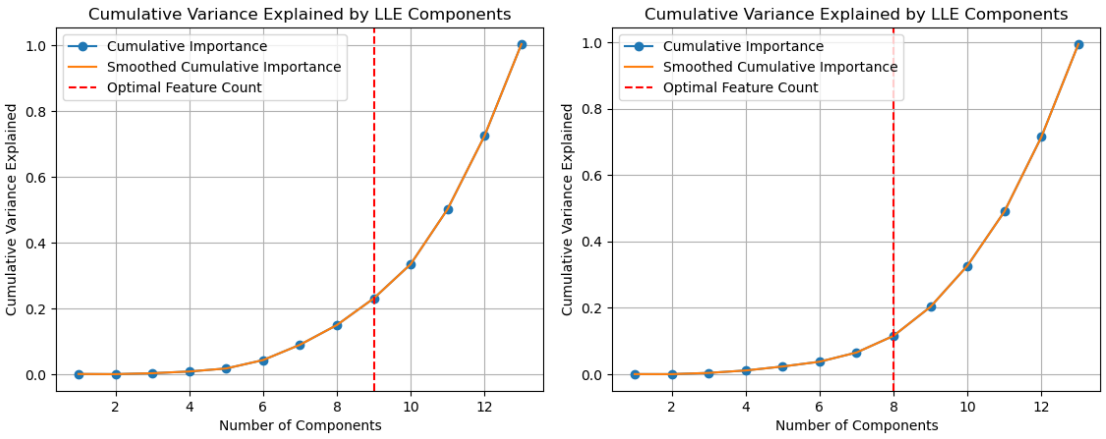
Πίνακας Α΄.36: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LLE Digits Dataset.

Wine Dataset



(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3

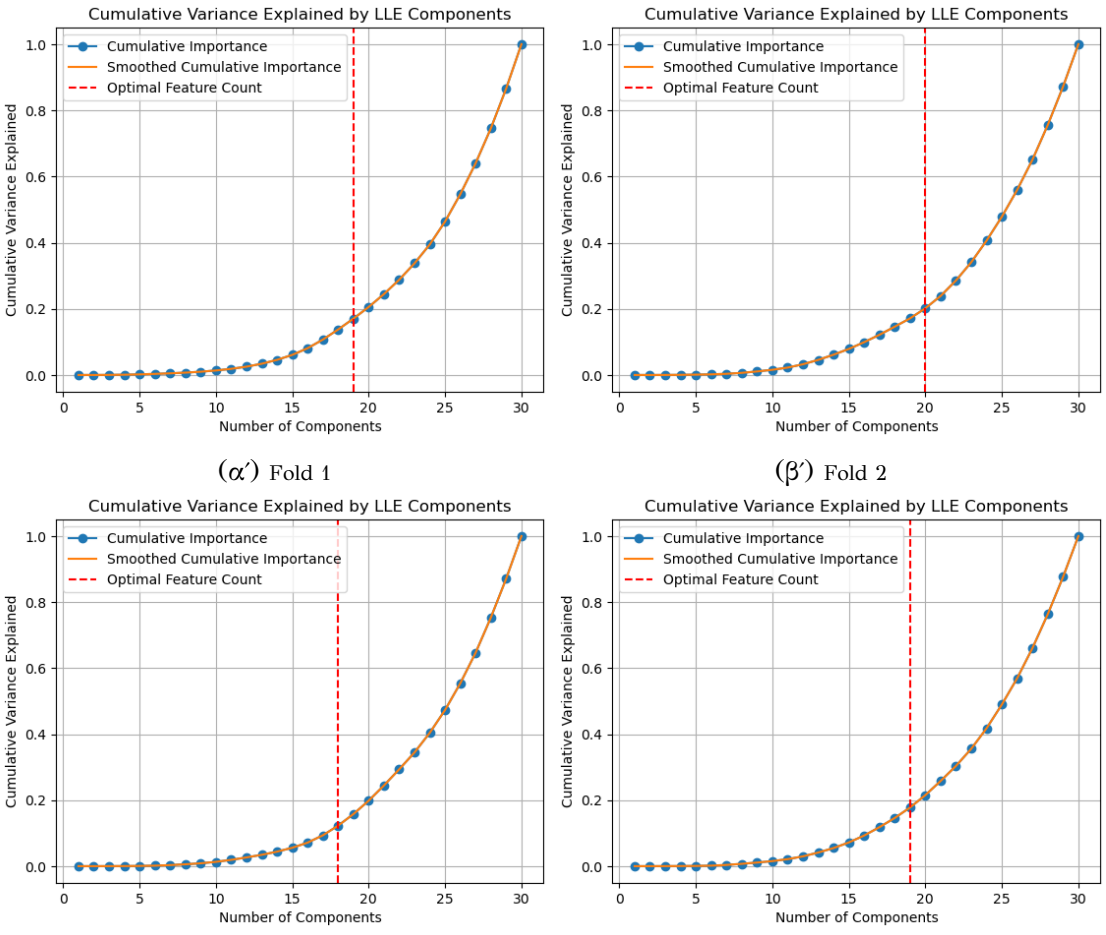
(δ΄) Fold 4

Σχήμα Α΄.93: Ανάλυση βέλτιστων διαστάσεων στο Wine Dataset με τη μέθοδο LLE

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.85	0.86	0.88	0.87
Average SVM	0.85	0.85	0.89	0.84
Average Decision Tree	0.80	0.80	0.82	0.82
Average Random Forest	0.90	0.90	0.91	0.91
Average Naive Bayes	0.83	0.83	0.85	0.85
Average MLP	0.90	0.91	0.92	0.91

Πίνακας Α΄.37: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LLE Wine Dataset.

Breast Cancer Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

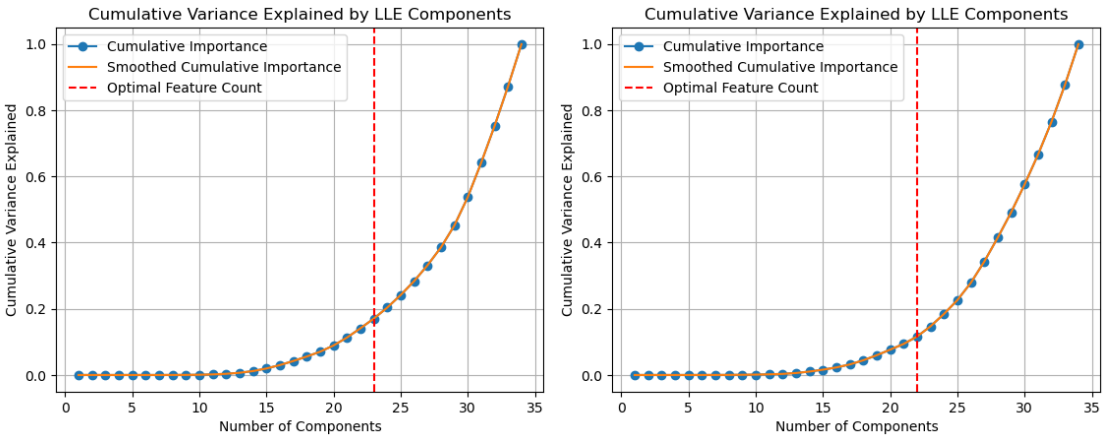
(δ΄) Fold 4

Σχήμα Α΄.94: Ανάλυση βέλτιστων διαστάσεων στο Breast Cancer Dataset με τη μέθοδο LLE

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.94	0.93	0.94	0.92
Average SVM	0.93	0.93	0.93	0.92
Average Decision Tree	0.89	0.89	0.89	0.89
Average Random Forest	0.93	0.93	0.94	0.92
Average Naive Bayes	0.86	0.85	0.86	0.84
Average MLP	0.94	0.94	0.94	0.93

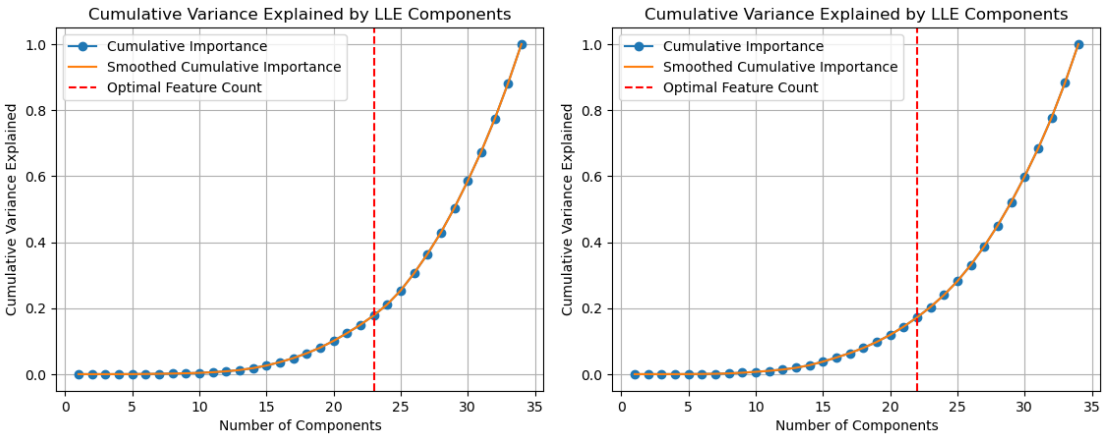
Πίνακας Α΄.38: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LLE Breast Cancer Dataset.

Ionosphere Dataset



(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3

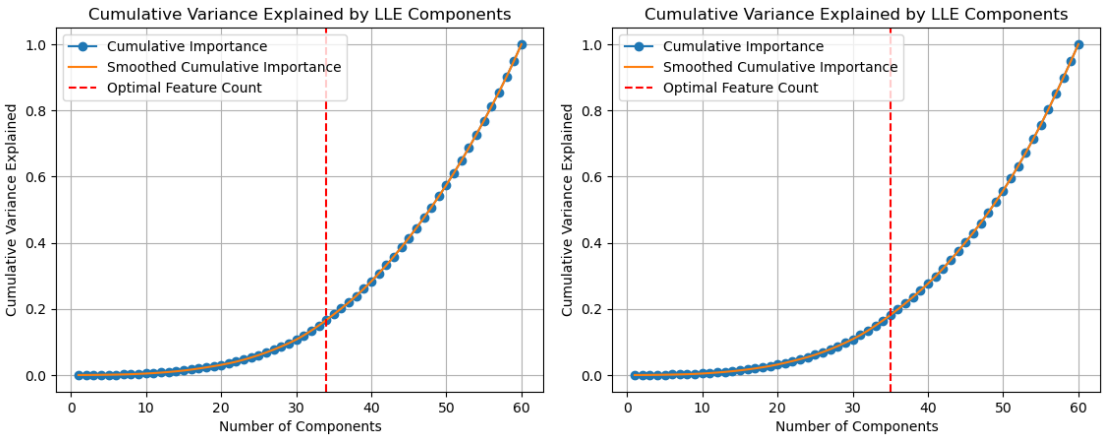
(δ΄) Fold 4

Σχήμα Α΄.95: Ανάλυση βέλτιστων διαστάσεων στο Ionosphere Dataset με τη μέθοδο LLE

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.85	0.82	0.88	0.80
Average SVM	0.89	0.88	0.88	0.89
Average Decision Tree	0.80	0.79	0.80	0.80
Average Random Forest	0.92	0.91	0.92	0.91
Average Naive Bayes	0.81	0.78	0.81	0.77
Average MLP	0.87	0.86	0.86	0.87

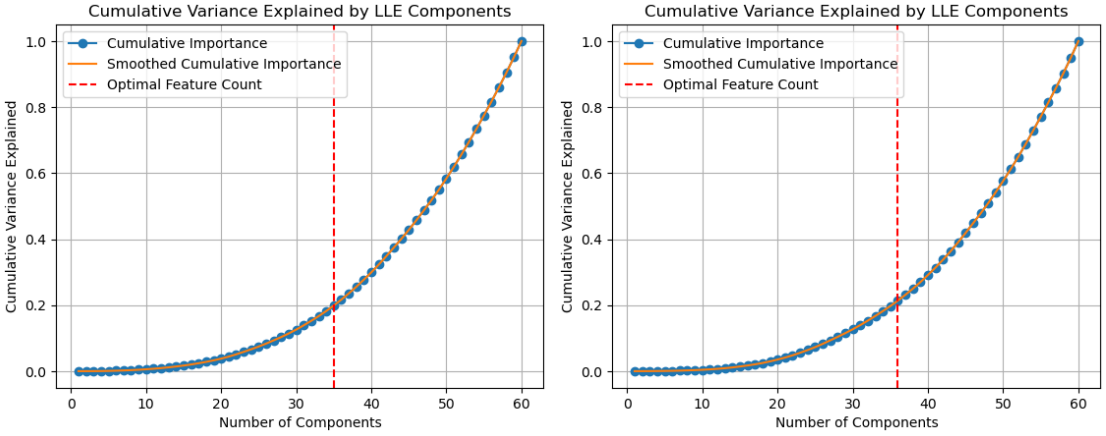
Πίνακας Α΄.39: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LLE Ionosphere Dataset.

Connectionist Bench Dataset



(α΄) Fold 1

(β΄) Fold 2



(γ΄) Fold 3

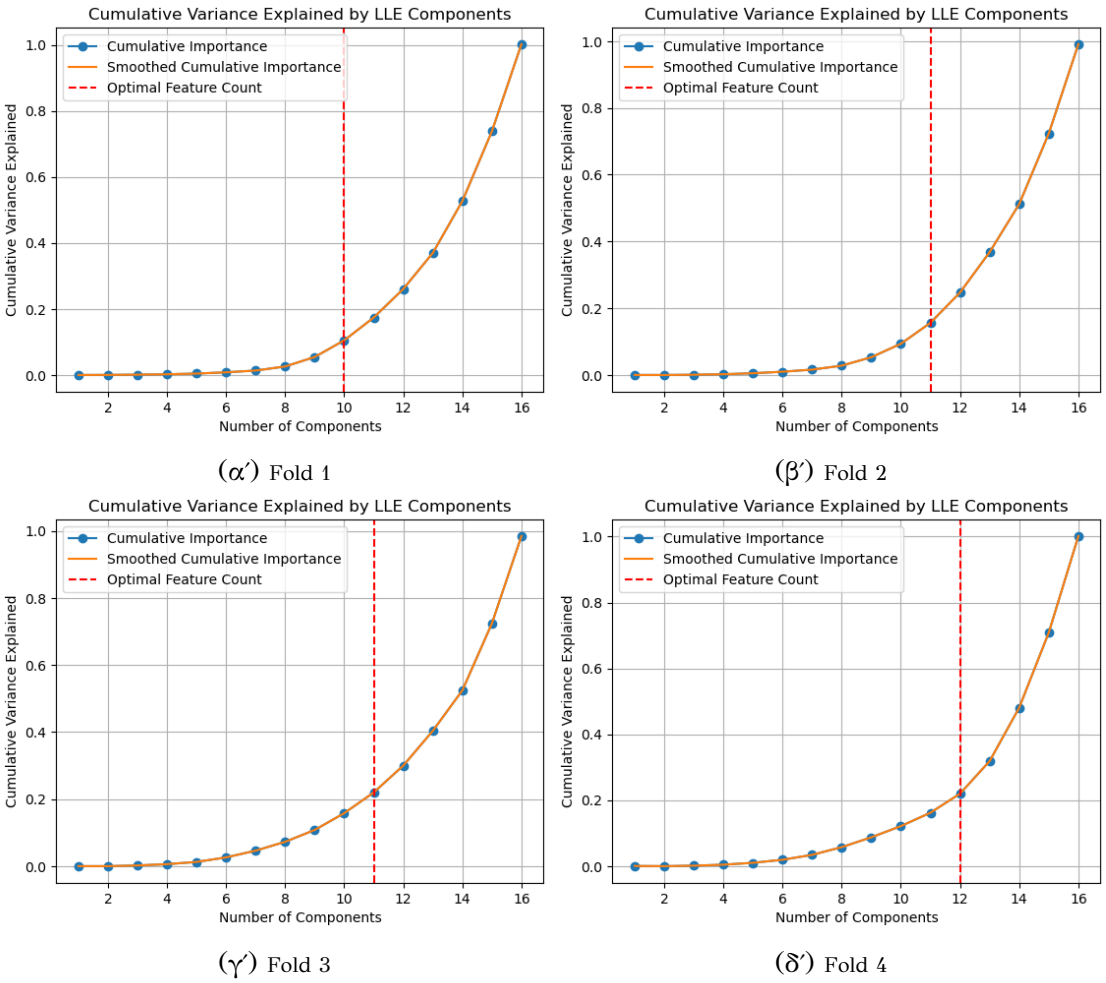
(δ΄) Fold 4

Σχήμα Α΄.96: Ανάλυση βέλτιστων διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο LLE

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.56	0.55	0.56	0.56
Average SVM	0.56	0.55	0.57	0.56
Average Decision Tree	0.47	0.47	0.47	0.47
Average Random Forest	0.51	0.51	0.52	0.52
Average Naive Bayes	0.57	0.56	0.58	0.57
Average MLP	0.54	0.52	0.57	0.55

Πίνακας Α΄.40: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LLE Connectionist Bench Dataset.

Dry Bean Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

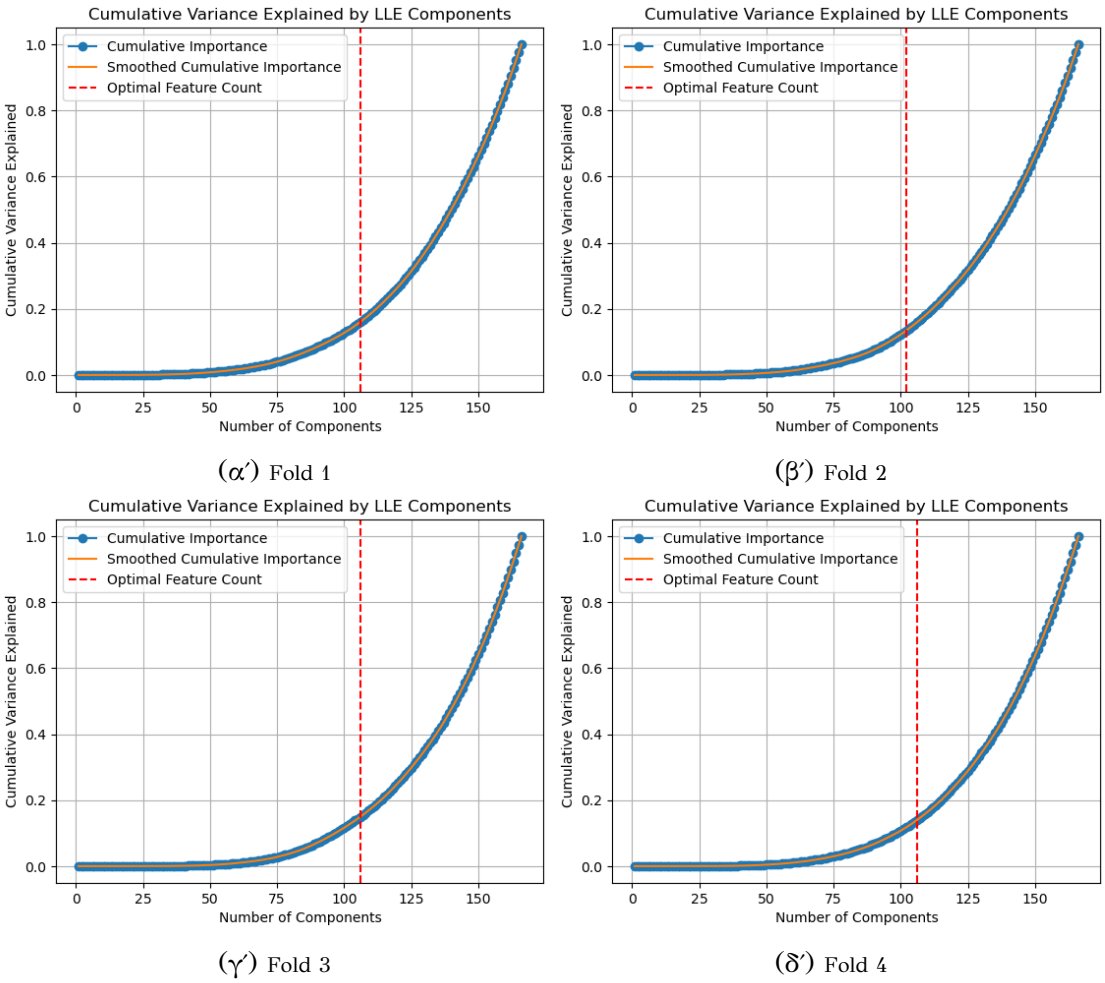
(δ΄) Fold 4

Σχήμα Α΄.97: Ανάλυση βέλτιστων διαστάσεων στο Dry Bean Dataset με τη μέθοδο LLE

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.83	0.85	0.88	0.85
Average SVM	0.83	0.85	0.88	0.86
Average Decision Tree	0.77	0.80	0.83	0.80
Average Random Forest	0.83	0.85	0.88	0.85
Average Naive Bayes	0.78	0.82	0.85	0.82
Average MLP	0.79	0.81	0.86	0.82

Πίνακας Α΄.41: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LLE Dry Bean Dataset.

Musk Dataset



(α′) Fold 1

(β′) Fold 2

(γ′) Fold 3

(δ′) Fold 4

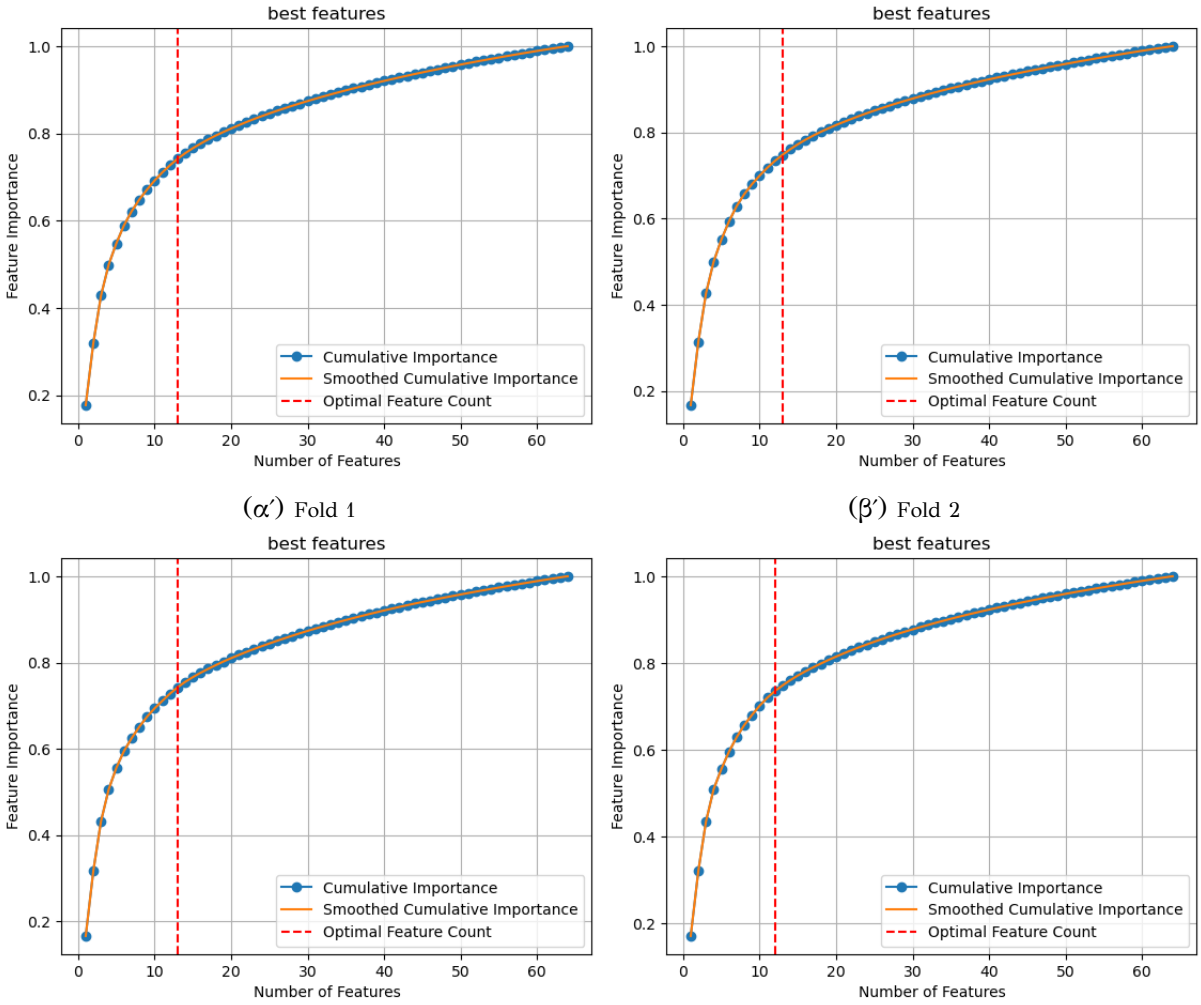
Σχήμα Α′.98: Ανάλυση βέλτιστων διαστάσεων στο Musk Dataset με τη μέθοδο LLE.

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.71	0.71	0.71	0.71
Average SVM	0.76	0.75	0.76	0.75
Average Decision Tree	0.64	0.64	0.65	0.65
Average Random Forest	0.72	0.71	0.72	0.71
Average Naive Bayes	0.58	0.57	0.61	0.59
Average MLP	0.76	0.76	0.77	0.76

Πίνακας Α′.42: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων LLE Musk Dataset.

Α΄.9.7 Isomap

Digits Dataset



(α΄) Fold 1 (β΄) Fold 2

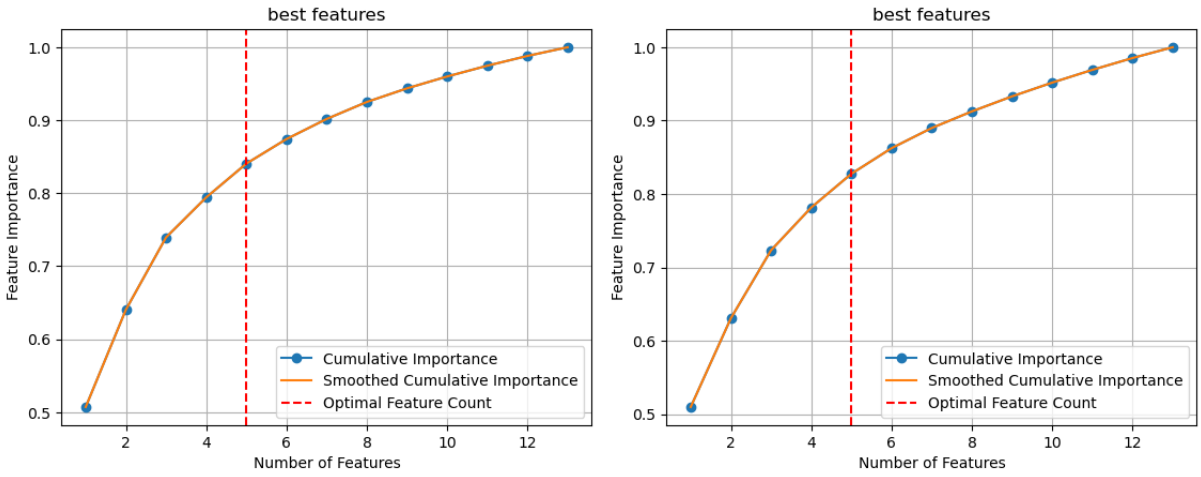
(γ΄) Fold 3 (δ΄) Fold 4

Σχήμα Α΄.99: Ανάλυση βέλτιστων διαστάσεων στο Digits Dataset με τη μέθοδο Kernel PCA

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.86	0.86	0.88	0.86
Average SVM	0.84	0.84	0.88	0.84
Average Decision Tree	0.65	0.65	0.69	0.65
Average Random Forest	0.83	0.83	0.86	0.83
Average Naive Bayes	0.69	0.68	0.79	0.69
Average MLP	0.70	0.70	0.84	0.70

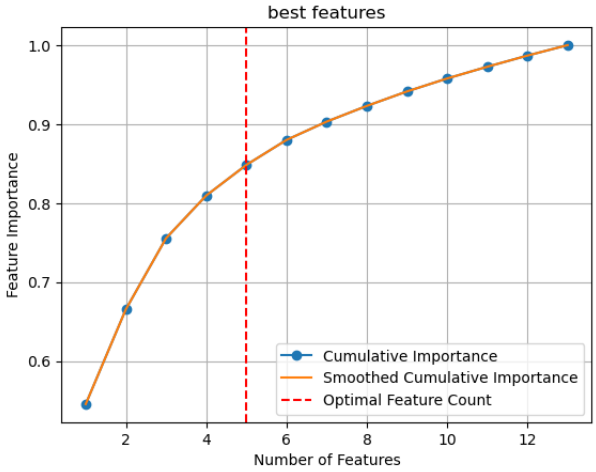
Πίνακας Α΄.43: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Isomap Digits Dataset.

Wine Dataset

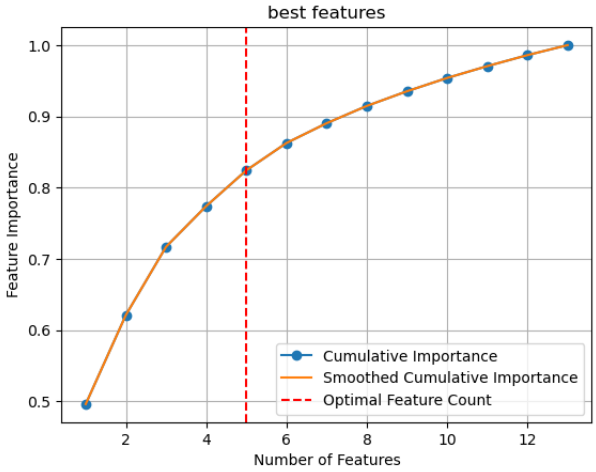


(α′) Fold 1

(β′) Fold 2



(γ′) Fold 3



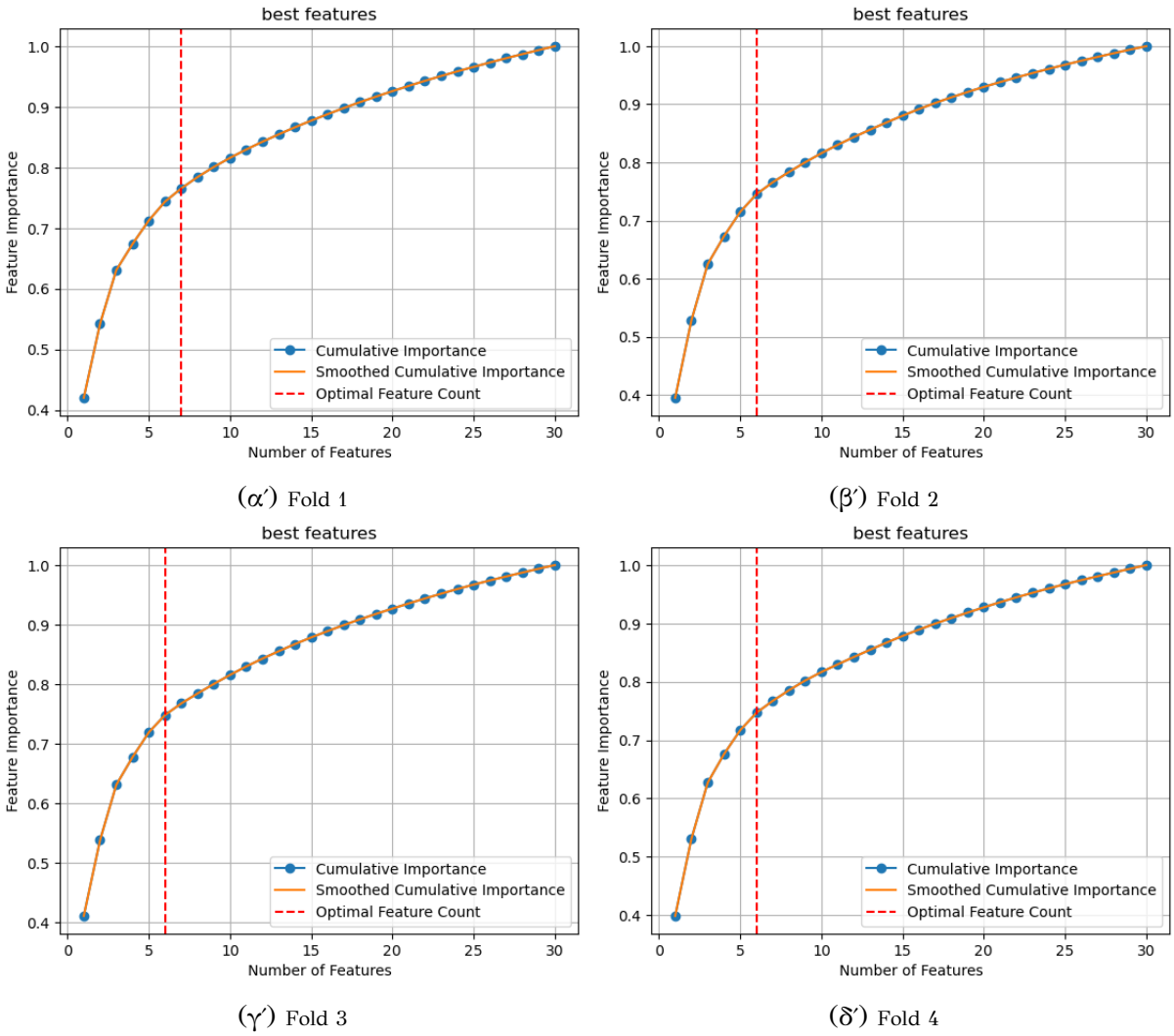
(δ′) Fold 4

Σχήμα Α′.100: Ανάλυση βέλτιστων διαστάσεων στο Wine Dataset με τη μέθοδο Isomap

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.90	0.90	0.94	0.89
Average SVM	0.86	0.85	0.92	0.85
Average Decision Tree	0.75	0.72	0.85	0.73
Average Random Forest	0.78	0.75	0.89	0.75
Average Naive Bayes	0.88	0.88	0.93	0.87
Average MLP	0.81	0.77	0.80	0.78

Πίνακας Α′.44: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Isomap Wine Dataset.

Breast Cancer Dataset



(α΄) Fold 1

(β΄) Fold 2

(γ΄) Fold 3

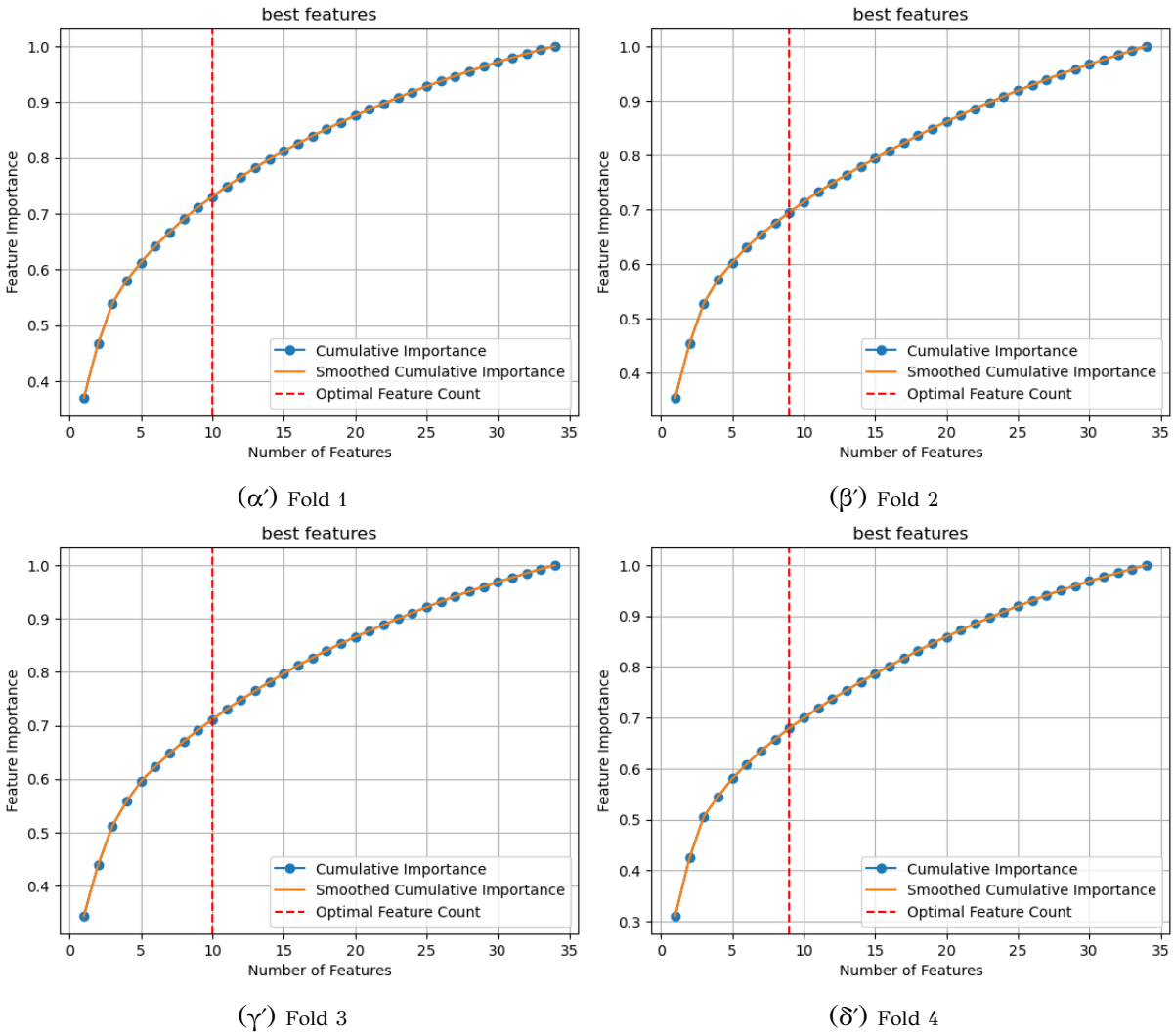
(δ΄) Fold 4

Σχήμα Α΄.101: Ανάλυση βέλτιστων διαστάσεων στο Breast Cancer Dataset με τη μέθοδο Isomap

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.93	0.92	0.95	0.91
Average SVM	0.91	0.91	0.92	0.90
Average Decision Tree	0.75	0.73	0.81	0.76
Average Random Forest	0.91	0.90	0.93	0.88
Average Naive Bayes	0.85	0.83	0.87	0.83
Average MLP	0.87	0.86	0.88	0.88

Πίνακας Α΄.45: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Isomap Breast Cancer Dataset.

Ionosphere Dataset



(α′) Fold 1 (β′) Fold 2

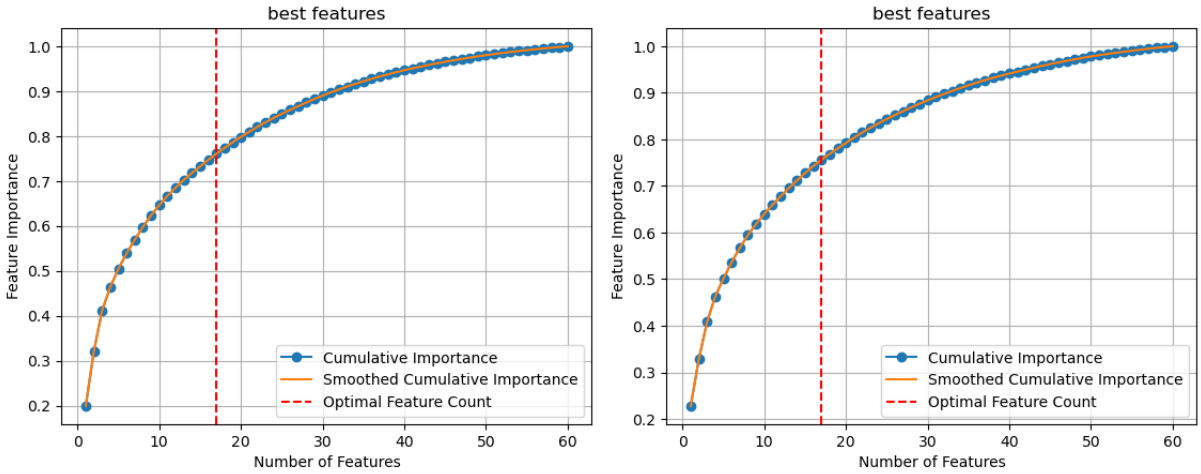
(γ′) Fold 3 (δ′) Fold 4

Σχήμα Α′.102: Ανάλυση βέλτιστων διαστάσεων στο Ionosphere Dataset με τη μέθοδο Isomap

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.76	0.67	0.82	0.67
Average SVM	0.85	0.82	0.85	0.82
Average Decision Tree	0.50	0.49	0.59	0.58
Average Random Forest	0.64	0.61	0.76	0.71
Average Naive Bayes	0.71	0.70	0.74	0.74
Average MLP	0.46	0.42	0.54	0.56

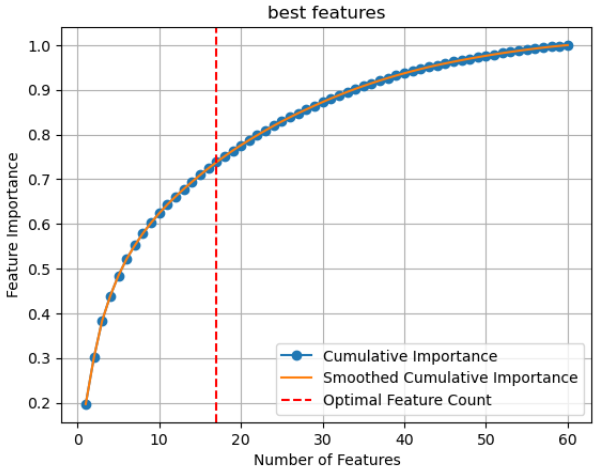
Πίνακας Α′.46: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Isomap Ionosphere Dataset.

Connectionist Bench Dataset

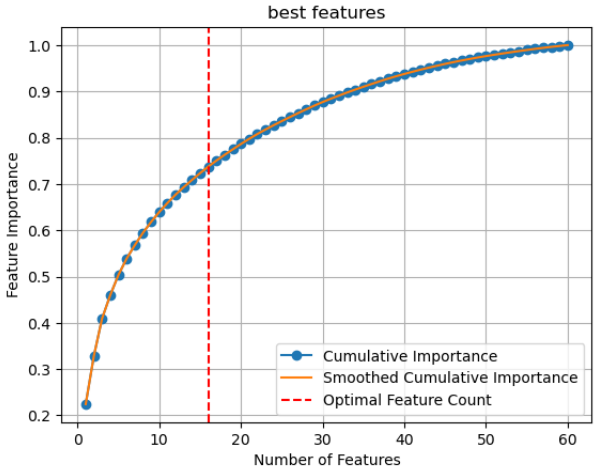


(α′) Fold 1

(β′) Fold 2



(γ′) Fold 3



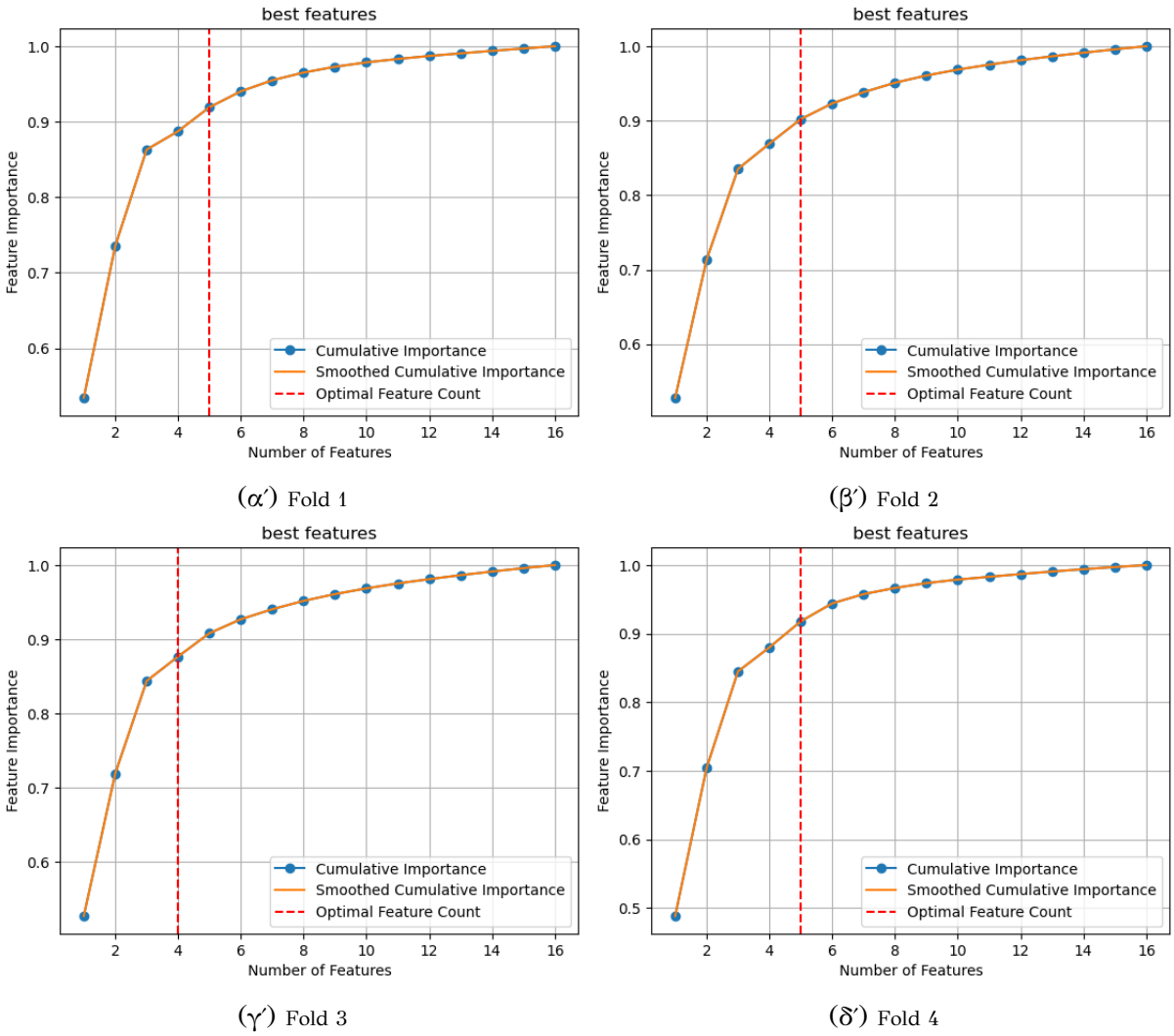
(δ′) Fold 4

Σχήμα Α′.103: Ανάλυση βέλτιστων διαστάσεων στο Connectionist Bench Dataset με τη μέθοδο Kernel PCA.

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.54	0.54	0.54	0.54
Average SVM	0.51	0.49	0.57	0.52
Average Decision Tree	0.54	0.48	0.63	0.55
Average Random Forest	0.57	0.52	0.68	0.57
Average Naive Bayes	0.53	0.50	0.56	0.54
Average MLP	0.58	0.58	0.60	0.59

Πίνακας Α′.47: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Isomap Connectionist Bench Dataset.

Dry Bean Dataset



(α′) Fold 1

(β′) Fold 2

(γ′) Fold 3

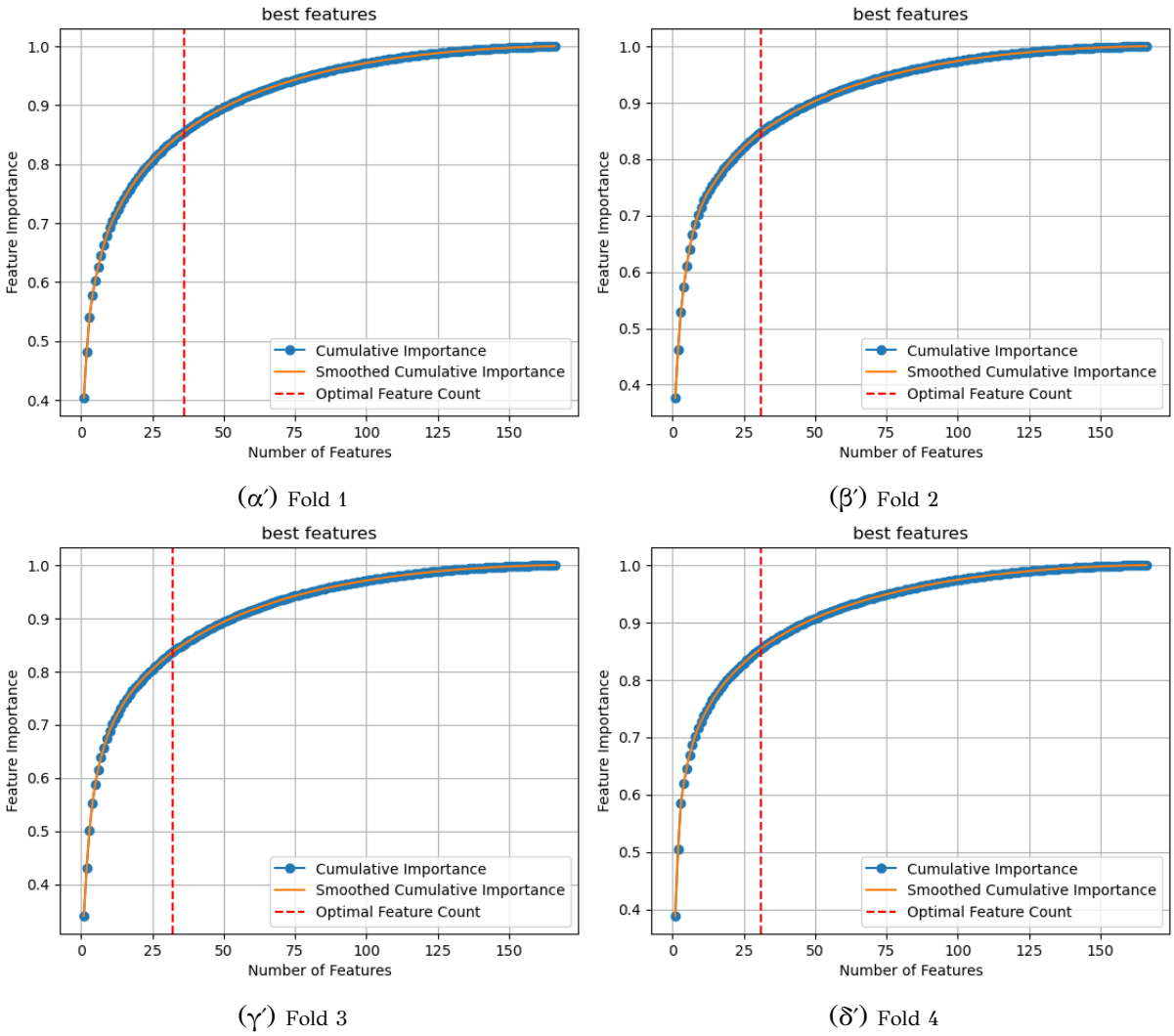
(δ′) Fold 4

Σχήμα Α′.104: Ανάλυση βέλτιστων διαστάσεων στο Dry Bean Dataset με τη μέθοδο Isomap

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.52	0.40	0.54	0.48
Average SVM	0.50	0.38	0.44	0.46
Average Decision Tree	0.35	0.25	0.33	0.32
Average Random Forest	0.43	0.33	0.47	0.39
Average Naive Bayes	0.33	0.26	0.35	0.37
Average MLP	0.46	0.31	0.38	0.39

Πίνακας Α′.48: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Isomap Dry Bean Dataset.

Musk Dataset



(α′) Fold 1

(β′) Fold 2

(γ′) Fold 3

(δ′) Fold 4

Σχήμα Α′.105: Ανάλυση βέλτιστων διαστάσεων στο Musk Dataset με τη μέθοδο Isomap.

Classifier	Accuracy	F1 Score	Precision	Recall
Average KNN	0.67	0.67	0.72	0.69
Average SVM	0.67	0.63	0.69	0.64
Average Decision Tree	0.56	0.55	0.56	0.55
Average Random Forest	0.70	0.68	0.70	0.68
Average Naive Bayes	0.65	0.55	0.61	0.60
Average MLP	0.64	0.63	0.69	0.66

Πίνακας Α′.49: Πίνακας αποτελεσμάτων βέλτιστου αριθμού διαστάσεων Isomap Musk Dataset

Α΄.10 Αποτελέσματα μεθόδων επιλογής χαρακτηριστικών

Α΄.10.1 Boruta algorithm

Wine Dataset

Random Forest				
Features selected (13/13), (13/13), (13/13), (13/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.96
Average SVM	0.98	0.98	0.98	0.98
Average Decision Tree	0.89	0.90	0.91	0.90
Average Random Forest	0.98	0.98	0.98	0.98
Average Naive Bayes	0.96	0.96	0.96	0.97
Average MLP	0.97	0.97	0.97	0.97
AdaBoost				
Features selected (3/13), (4/13), (5/13), (3/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.94	0.94	0.95	0.95
Average SVM	0.94	0.94	0.94	0.95
Average Decision Tree	0.91	0.91	0.92	0.91
Average Random Forest	0.94	0.95	0.95	0.95
Average Naive Bayes	0.92	0.92	0.94	0.93
Average MLP	0.91	0.91	0.92	0.92

Gradient Boost				
Features selected (5/13), (5/13), (6/13), (7/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.96
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.90	0.90	0.91	0.91
Average Random Forest	0.94	0.94	0.95	0.95
Average Naive Bayes	0.92	0.92	0.93	0.93
Average MLP	0.94	0.95	0.95	0.95
XGBoost				
Features selected (7/13), (4/13), (6/13), (6/13)				
Average KNN	0.96	0.96	0.96	0.96
Average SVM	0.94	0.94	0.95	0.95
Average Decision Tree	0.88	0.89	0.90	0.89
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.91	0.91	0.92	0.92
Average MLP	0.95	0.95	0.96	0.95
LightGBM				
Features selected (7/13), (8/13), (8/13), (8/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.94	0.94	0.95	0.95
Average SVM	0.97	0.97	0.97	0.97

Average Decision Tree	0.92	0.92	0.93	0.92
Average Random Forest	0.95	0.95	0.95	0.96
Average Naive Bayes	0.96	0.96	0.96	0.96
Average MLP	0.96	0.96	0.96	0.96
CatBoost				
Features selected (9/13), (12/13), (11/13), (13/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.93	0.93	0.94	0.94
Average SVM	0.97	0.97	0.97	0.98
Average Decision Tree	0.88	0.88	0.90	0.89
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.96	0.96	0.96	0.96
Average MLP	0.96	0.96	0.96	0.97

Πίνακας Α΄.50: Πίνακας αποτελεσμάτων Boruta Wine Dataset

Digits Dataset

Random Forest				
Features selected (58/64), (58/64), (57/64), (59/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.94	0.94	0.95	0.94

Average SVM	0.96	0.96	0.97	0.96
Average Decision Tree	0.79	0.79	0.79	0.79
Average Random Forest	0.94	0.94	0.94	0.94
Average Naive Bayes	0.83	0.83	0.84	0.83
Average MLP	0.95	0.95	0.95	0.95
AdaBoost				
Features selected (3/64), (3/64), (2/64), (2/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.41	0.39	0.40	0.41
Average SVM	0.42	0.40	0.40	0.42
Average Decision Tree	0.39	0.39	0.39	0.39
Average Random Forest	0.40	0.39	0.39	0.40
Average Naive Bayes	0.39	0.34	0.36	0.39
Average MLP	0.44	0.43	0.44	0.44
Gradient Boost				
Features selected (48/64), (46/64), (48/64), (49/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.95
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.78	0.78	0.78	0.78

Average Random Forest	0.94	0.94	0.94	0.94
Average Naive Bayes	0.86	0.86	0.87	0.86
Average MLP	0.94	0.94	0.94	0.94
XGBoost				
Features selected (44/64), (44/64), (45/64), (50/64)				
Average KNN	0.95	0.95	0.95	0.95
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.80	0.80	0.81	0.80
Average Random Forest	0.94	0.94	0.94	0.94
Average Naive Bayes	0.85	0.85	0.86	0.85
Average MLP	0.95	0.95	0.95	0.95
LightGBM				
Features selected (45/64), (45/64), (47/64), (50/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.95
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.79	0.79	0.79	0.79
Average Random Forest	0.94	0.94	0.94	0.94
Average Naive Bayes	0.83	0.83	0.84	0.83
Average MLP	0.95	0.95	0.95	0.95

CatBoost				
Features selected (46/64), (46/64), (47/64), (50/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.95
Average SVM	0.96	0.96	0.97	0.96
Average Decision Tree	0.79	0.79	0.80	0.79
Average Random Forest	0.94	0.94	0.94	0.94
Average Naive Bayes	0.84	0.83	0.85	0.84
Average MLP	0.94	0.94	0.95	0.95

Πίνακας Α΄.51: Πίνακας αποτελεσμάτων Boruta Digits Dataset

Breast Cancer Dataset

Random Forest				
Features selected (26/30), (27/30), (26/30), (27/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.97	0.97	0.96
Average SVM	0.97	0.97	0.97	0.96
Average Decision Tree	0.91	0.91	0.90	0.91
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.93	0.93	0.93	0.92

Average MLP	0.97	0.97	0.97	0.97
AdaBoost				
Features selected (4/30), (6/30), (7/30), (10/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.95
Average SVM	0.96	0.96	0.96	0.96
Average Decision Tree	0.93	0.93	0.93	0.93
Average Random Forest	0.96	0.96	0.96	0.95
Average Naive Bayes	0.94	0.94	0.94	0.93
Average MLP	0.96	0.95	0.96	0.95
Gradient Boost				
Features selected (9/30), (8/30), (10/30), (11/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.96	0.96	0.97	0.96
Average SVM	0.97	0.97	0.98	0.97
Average Decision Tree	0.92	0.92	0.91	0.92
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.95	0.95	0.95	0.95
Average MLP	0.96	0.96	0.96	0.96
XGBoost				
Features selected (8/30), (11/30), (9/30), (10/30)				
Average KNN	0.95	0.95	0.95	0.95

Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.92	0.92	0.92	0.92
Average Random Forest	0.96	0.95	0.95	0.95
Average Naive Bayes	0.95	0.95	0.95	0.94
Average MLP	0.96	0.96	0.96	0.96
LightGBM				
Features selected (7/30), (7/30), (9/30), (10/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.96	0.96	0.96	0.95
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.91	0.91	0.91	0.91
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.96	0.95	0.96	0.95
Average MLP	0.96	0.95	0.95	0.96
CatBoost				
Features selected (14/30), (13/30), (11/30), (13/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.97	0.97	0.96
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.92	0.91	0.91	0.91

Average Random Forest	0.97	0.97	0.97	0.97
Average Naive Bayes	0.95	0.95	0.95	0.94
Average MLP	0.97	0.97	0.97	0.97

Πίνακας Α΄.52: Πίνακας αποτελεσμάτων Boruta Breast Cancer Dataset

Ionosphere Dataset

Random Forest				
Features selected (29/34), (33/34), (30/34), (33/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.82	0.78	0.86	0.76
Average SVM	0.95	0.94	0.95	0.94
Average Decision Tree	0.86	0.84	0.86	0.83
Average Random Forest	0.93	0.92	0.93	0.92
Average Naive Bayes	0.87	0.85	0.87	0.85
Average MLP	0.89	0.88	0.90	0.87
AdaBoost				
Features selected (4/34), (4/34), (2/34), (4/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.80	0.78	0.79	0.78
Average SVM	0.87	0.86	0.88	0.84

Average Decision Tree	0.83	0.82	0.81	0.83
Average Random Forest	0.87	0.86	0.86	0.85
Average Naive Bayes	0.81	0.79	0.80	0.78
Average MLP	0.83	0.81	0.82	0.80
Gradient Boost				
Features selected (5/34), (8/34), (11/34), (6/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.87	0.85	0.89	0.84
Average SVM	0.91	0.90	0.91	0.89
Average Decision Tree	0.87	0.86	0.87	0.85
Average Random Forest	0.91	0.91	0.91	0.90
Average Naive Bayes	0.87	0.85	0.88	0.84
Average MLP	0.90	0.89	0.90	0.89
XGBoost				
Features selected (4/34), (8/34), (5/34), (5/34)				
Average KNN	0.90	0.89	0.91	0.88
Average SVM	0.91	0.90	0.91	0.89
Average Decision Tree	0.84	0.83	0.84	0.83
Average Random Forest	0.92	0.92	0.92	0.91

Average Naive Bayes	0.87	0.85	0.87	0.85
Average MLP	0.91	0.90	0.91	0.89
LightGBM				
Features selected (10/34), (10/34), (9/34), (9/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.88	0.86	0.90	0.84
Average SVM	0.94	0.93	0.94	0.93
Average Decision Tree	0.90	0.89	0.89	0.89
Average Random Forest	0.93	0.93	0.93	0.92
Average Naive Bayes	0.91	0.90	0.90	0.91
Average MLP	0.90	0.89	0.90	0.89
CatBoost				
Features selected (16/34), (14/34), (17/34), (16/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.83	0.79	0.87	0.78
Average SVM	0.95	0.94	0.95	0.93
Average Decision Tree	0.87	0.85	0.86	0.84
Average Random Forest	0.94	0.93	0.94	0.93
Average Naive Bayes	0.87	0.85	0.87	0.86
Average MLP	0.88	0.87	0.88	0.86

Πίνακας Α΄.53: Πίνακας αποτελεσμάτων Boruta Ionosphere Dataset

Connectionist Bench Dataset

Random Forest				
Features selected (20/60), (23/60), (23/60), (23/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.54	0.53	0.54	0.54
Average SVM	0.52	0.50	0.54	0.52
Average Decision Tree	0.59	0.58	0.60	0.59
Average Random Forest	0.61	0.59	0.63	0.61
Average Naive Bayes	0.54	0.52	0.58	0.55
Average MLP	0.60	0.58	0.64	0.60
AdaBoost				
Features selected (26/60), (27/60), (29/60), (24/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.55	0.55	0.56	0.55
Average SVM	0.60	0.59	0.61	0.60
Average Decision Tree	0.60	0.59	0.62	0.61
Average Random Forest	0.60	0.58	0.62	0.60
Average Naive Bayes	0.54	0.53	0.59	0.56

Average MLP	0.64	0.63	0.68	0.64
Gradient Boost				
Features selected (13/60), (16/60), (21/60), (15/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.54	0.53	0.54	0.54
Average SVM	0.56	0.54	0.62	0.57
Average Decision Tree	0.58	0.56	0.60	0.58
Average Random Forest	0.60	0.58	0.64	0.60
Average Naive Bayes	0.50	0.48	0.53	0.50
Average MLP	0.63	0.60	0.68	0.63
XGBoost				
Features selected (11/20), (27/28), (21/21), (22/22)				
Average KNN	0.52	0.51	0.53	0.52
Average SVM	0.52	0.49	0.56	0.52
Average Decision Tree	0.57	0.56	0.60	0.58
Average Random Forest	0.56	0.54	0.59	0.56
Average Naive Bayes	0.50	0.48	0.54	0.50
Average MLP	0.54	0.52	0.60	0.55
LightGBM				
Features selected (17/60), (23/60), (24/60), (23/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.52	0.51	0.52	0.52

Average SVM	0.58	0.56	0.61	0.58
Average Decision Tree	0.57	0.55	0.59	0.57
Average Random Forest	0.62	0.59	0.67	0.62
Average Naive Bayes	0.57	0.55	0.63	0.58
Average MLP	0.61	0.58	0.66	0.61
CatBoost				
Features selected (18/60), (23/60), (24/60), (21/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.55	0.54	0.55	0.55
Average SVM	0.60	0.58	0.63	0.60
Average Decision Tree	0.62	0.60	0.64	0.62
Average Random Forest	0.61	0.59	0.64	0.61
Average Naive Bayes	0.55	0.52	0.60	0.56
Average MLP	0.58	0.55	0.64	0.59

Πίνακας Α΄.54: Πίνακας αποτελεσμάτων Boruta Connectionist Bench Dataset

Dry Bean Dataset

Random Forest				
Features selected (16/16), (16/16), (16/16), (16/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.83	0.86	0.88	0.86

Average SVM	0.83	0.85	0.88	0.86
Average Decision Tree	0.56	0.58	0.64	0.63
Average Random Forest	0.54	0.56	0.61	0.62
Average Naive Bayes	0.82	0.84	0.87	0.85
Average MLP	0.76	0.78	0.83	0.80
AdaBoost				
Features selected (7/16), (4/16), (7/16), (6/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.57	0.59	0.66	0.61
Average SVM	0.77	0.78	0.82	0.79
Average Decision Tree	0.39	0.41	0.45	0.45
Average Random Forest	0.39	0.40	0.45	0.45
Average Naive Bayes	0.66	0.66	0.69	0.68
Average MLP	0.65	0.68	0.74	0.70
Gradient Boost				
Features selected (16/16), (16/16), (16/16), (16/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.83	0.86	0.88	0.86
Average SVM	0.83	0.85	0.88	0.86
Average Decision Tree	0.56	0.59	0.64	0.64

Average Random Forest	0.54	0.56	0.61	0.62
Average Naive Bayes	0.82	0.84	0.87	0.85
Average MLP	0.76	0.78	0.83	0.80
XGBoost				
Features selected (12/16), (13/16), (13/16), (13/16)				
Average KNN	0.83	0.85	0.88	0.86
Average SVM	0.83	0.85	0.88	0.86
Average Decision Tree	0.56	0.59	0.65	0.64
Average Random Forest	0.57	0.60	0.66	0.64
Average Naive Bayes	0.82	0.84	0.87	0.85
Average MLP	0.80	0.82	0.84	0.84
LightGBM				
Features selected (9/16), (7/16), (6/16), (6/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.86	0.88	0.90	0.88
Average SVM	0.85	0.87	0.90	0.88
Average Decision Tree	0.55	0.57	0.61	0.62
Average Random Forest	0.63	0.65	0.72	0.69
Average Naive Bayes	0.84	0.86	0.88	0.87
Average MLP	0.78	0.79	0.82	0.81

CatBoost				
Features selected (16/16), (16/16), (16/16), (16/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.83	0.86	0.88	0.86
Average SVM	0.83	0.85	0.88	0.86
Average Decision Tree	0.57	0.60	0.65	0.65
Average Random Forest	0.54	0.56	0.61	0.62
Average Naive Bayes	0.82	0.84	0.87	0.85
Average MLP	0.76	0.78	0.83	0.80

Πίνακας Α΄.55: Πίνακας αποτελεσμάτων Boruta Dry Bean Dataset

Musk Dataset

Random Forest				
Features selected (66/168), (75/168), (66/168), (77/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.70	0.70	0.71	0.70
Average SVM	0.72	0.70	0.73	0.71
Average Decision Tree	0.65	0.65	0.65	0.65
Average Random Forest	0.66	0.64	0.66	0.65
Average Naive Bayes	0.59	0.57	0.58	0.57

Average MLP	0.76	0.75	0.76	0.75
AdaBoost				
Features selected (5/168), (6/168), (3/168), (3/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.62	0.60	0.63	0.61
Average SVM	0.61	0.59	0.61	0.60
Average Decision Tree	0.62	0.61	0.61	0.61
Average Random Forest	0.61	0.60	0.61	0.61
Average Naive Bayes	0.53	0.53	0.54	0.54
Average MLP	0.60	0.58	0.60	0.59
Gradient Boost				
Features selected (18/168), (35/168), (24/168), (14/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.63	0.62	0.63	0.62
Average SVM	0.69	0.66	0.69	0.66
Average Decision Tree	0.62	0.61	0.62	0.62
Average Random Forest	0.70	0.67	0.70	0.68
Average Naive Bayes	0.59	0.57	0.58	0.58
Average MLP	0.71	0.70	0.71	0.70
XGBoost				
Features selected (9/168), (20/168), (4/168),(7/168)				
Average KNN	0.63	0.61	0.62	0.62

Average SVM	0.67	0.63	0.69	0.65
Average Decision Tree	0.61	0.60	0.60	0.60
Average Random Forest	0.65	0.63	0.65	0.63
Average Naive Bayes	0.56	0.54	0.55	0.55
Average MLP	0.65	0.64	0.64	0.64
LightGBM				
Features selected (19/168), (28/168), (19/168), (11/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.66	0.65	0.66	0.66
Average SVM	0.70	0.67	0.71	0.68
Average Decision Tree	0.59	0.59	0.60	0.60
Average Random Forest	0.68	0.67	0.69	0.67
Average Naive Bayes	0.61	0.59	0.60	0.59
Average MLP	0.70	0.69	0.71	0.70
CatBoost				
Features selected (35/168), (43/168), (36/168), (24/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.70	0.69	0.71	0.70
Average SVM	0.72	0.70	0.72	0.70
Average Decision Tree	0.62	0.61	0.61	0.62

Average Random Forest	0.70	0.68	0.70	0.68
Average Naive Bayes	0.59	0.58	0.59	0.58
Average MLP	0.74	0.72	0.74	0.73

Πίνακας Α΄.56: Πίνακας αποτελεσμάτων Boruta Musk Dataset

Α΄.11 Ensemble learning

Wine Dataset

Random Forest				
Features selected (6/13), (6/13), (6/13), (6/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.97	0.97	0.98
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.93	0.93	0.94	0.93
Average Random Forest	0.96	0.96	0.97	0.96
Average Naive Bayes	0.96	0.96	0.96	0.96
Average MLP	0.97	0.97	0.97	0.98
AdaBoost				
Features selected (3/13), (3/13), (3/13), (3/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.92	0.92	0.92	0.93

Average SVM	0.94	0.94	0.94	0.95
Average Decision Tree	0.89	0.89	0.91	0.89
Average Random Forest	0.91	0.91	0.92	0.92
Average Naive Bayes	0.91	0.90	0.92	0.91
Average MLP	0.91	0.91	0.91	0.92
Gradient Boost				
Features selected (4/13), (4/13), (4/13), (4/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.96	0.96	0.96	0.97
Average SVM	0.94	0.95	0.95	0.95
Average Decision Tree	0.93	0.93	0.94	0.93
Average Random Forest	0.96	0.96	0.97	0.96
Average Naive Bayes	0.93	0.93	0.95	0.94
Average MLP	0.96	0.96	0.97	0.96
XGBoost				
Features selected (5/13), (4/13), (4/13), (5/13)				
Average KNN	0.93	0.93	0.93	0.93
Average SVM	0.95	0.95	0.95	0.96
Average Decision Tree	0.88	0.88	0.90	0.88
Average Random Forest	0.95	0.95	0.96	0.95

Average Naive Bayes	0.91	0.91	0.92	0.92
Average MLP	0.93	0.93	0.93	0.93
LightGBM				
Features selected (7/13), (4/13), (6/13), (7/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.97	0.97	0.97
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.88	0.89	0.90	0.89
Average Random Forest	0.93	0.93	0.94	0.93
Average Naive Bayes	0.96	0.96	0.96	0.96
Average MLP	0.97	0.97	0.97	0.98
CatBoost				
Features selected (5/13), (6/13), (6/13), (6/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.97	0.97	0.97
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.91	0.91	0.92	0.91
Average Random Forest	0.94	0.94	0.95	0.95
Average Naive Bayes	0.94	0.94	0.95	0.95
Average MLP	0.96	0.96	0.96	0.96

Πίνακας Α΄.57: Πίνακας αποτελεσμάτων ensemble learning Wine Dataset Dataset

Digits Dataset

Random Forest				
Features selected (29/64), (30/64), (28/64), (28/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.94	0.94	0.94	0.94
Average SVM	0.95	0.95	0.96	0.95
Average Decision Tree	0.79	0.79	0.80	0.79
Average Random Forest	0.93	0.93	0.93	0.93
Average Naive Bayes	0.84	0.84	0.86	0.84
Average MLP	0.93	0.93	0.93	0.93
AdaBoost				
Features selected (3/64), (3/64), (3/64), (3/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average SVM	0.43	0.41	0.41	0.43
Average Decision Tree	0.41	0.40	0.41	0.41
Average Random Forest	0.43	0.42	0.42	0.43
Average Naive Bayes	0.36	0.30	0.33	0.36
Average MLP	0.44	0.42	0.44	0.44

Gradient Boost				
Features selected (19/64), (21/64), (23/64), (22/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.93	0.93	0.93	0.93
Average SVM	0.94	0.94	0.94	0.94
Average Decision Tree	0.78	0.78	0.78	0.78
Average Random Forest	0.91	0.91	0.92	0.91
Average Naive Bayes	0.82	0.82	0.85	0.82
Average MLP	0.92	0.92	0.93	0.92
XGBoost				
Features selected (24/64), (25/64), (27/64), (25/64)				
Average KNN	0.92	0.92	0.92	0.92
Average SVM	0.94	0.94	0.95	0.94
Average Decision Tree	0.79	0.79	0.80	0.79
Average Random Forest	0.92	0.92	0.92	0.92
Average Naive Bayes	0.77	0.77	0.81	0.77
Average MLP	0.92	0.92	0.92	0.92
LightGBM				
Features selected (34/64), (36/64), (36/64), (36/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.95
Average SVM	0.97	0.97	0.97	0.97

Average Decision Tree	0.78	0.78	0.79	0.78
Average Random Forest	0.94	0.94	0.94	0.94
Average Naive Bayes	0.85	0.85	0.87	0.85
Average MLP	0.95	0.95	0.95	0.95
CatBoost				
Features selected (18/64), (22/64), (21/64), (24/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.93	0.93	0.93	0.93
Average SVM	0.95	0.95	0.95	0.95
Average Decision Tree	0.79	0.79	0.80	0.79
Average Random Forest	0.91	0.91	0.92	0.91
Average Naive Bayes	0.83	0.82	0.84	0.83
Average MLP	0.92	0.92	0.92	0.92

Πίνακας Α΄.58: Πίνακας αποτελεσμάτων ensemble learning Digits Dataset Dataset

Breast Cancer Dataset

Random Forest				
Features selected (10/30), (10/30), (11/30), (11/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.94

Average SVM	0.95	0.95	0.95	0.94
Average Decision Tree	0.92	0.92	0.92	0.92
Average Random Forest	0.95	0.94	0.95	0.94
Average Naive Bayes	0.93	0.93	0.93	0.92
Average MLP	0.96	0.95	0.95	0.95
AdaBoost				
Features selected (15/30), (13/30), (12/30), (14/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.96	0.96	0.97	0.96
Average SVM	0.98	0.97	0.98	0.97
Average Decision Tree	0.92	0.92	0.92	0.92
Average Random Forest	0.96	0.95	0.96	0.95
Average Naive Bayes	0.93	0.93	0.93	0.92
Average MLP	0.97	0.97	0.97	0.97
Gradient Boost				
Features selected (8/30), (8/30), (9/30), (8/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.96	0.97	0.96
Average SVM	0.98	0.97	0.98	0.97
Average Decision Tree	0.92	0.92	0.92	0.92

Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.96	0.95	0.96	0.95
Average MLP	0.96	0.95	0.95	0.96
XGBoost				
Features selected (8/30), (7/30), (6/30), (10/30)				
Average KNN	0.95	0.95	0.95	0.95
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.92	0.92	0.92	0.91
Average Random Forest	0.96	0.95	0.96	0.95
Average Naive Bayes	0.95	0.94	0.95	0.94
Average MLP	0.95	0.95	0.95	0.96
LightGBM				
Features selected (9/30), (11/30), (10/30), (12/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.97	0.97	0.97
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.91	0.91	0.91	0.91
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.96	0.95	0.95	0.95
Average MLP	0.97	0.96	0.96	0.97

CatBoost				
Features selected (12/30), (11/30), (10/30), (11/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.96	0.97	0.96
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.91	0.91	0.91	0.91
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.95	0.95	0.95	0.94
Average MLP	0.96	0.96	0.96	0.96

Πίνακας Α΄.59: Πίνακας αποτελεσμάτων ensemble learning Breast Cancer Dataset Dataset

Ionosphere Dataset

Random Forest				
Features selected (12/34), (11/34), (15/34), (10/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.86	0.83	0.89	0.81
Average SVM	0.92	0.91	0.93	0.91
Average Decision Tree	0.87	0.85	0.87	0.83
Average Random Forest	0.94	0.93	0.94	0.92
Average Naive Bayes	0.89	0.88	0.92	0.86

Average MLP	0.90	0.89	0.89	0.89
AdaBoost				
Features selected (17/34), (17/34), (18/34), (19/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.83	0.80	0.87	0.78
Average SVM	0.93	0.92	0.93	0.91
Average Decision Tree	0.87	0.85	0.87	0.84
Average Random Forest	0.93	0.92	0.93	0.91
Average Naive Bayes	0.83	0.82	0.82	0.83
Average MLP	0.89	0.88	0.89	0.87
Gradient Boost				
Features selected (8/34), (7/34), (9/34), (8/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.87	0.85	0.89	0.83
Average SVM	0.92	0.91	0.92	0.90
Average Decision Tree	0.89	0.87	0.89	0.87
Average Random Forest	0.93	0.92	0.93	0.91
Average Naive Bayes	0.89	0.87	0.89	0.87
Average MLP	0.89	0.88	0.88	0.88
XGBoost				
Features selected (12/34), (9/34), (15/34), (13/34)				
Average KNN	0.85	0.83	0.88	0.81

Average SVM	0.93	0.92	0.94	0.91
Average Decision Tree	0.82	0.80	0.81	0.80
Average Random Forest	0.92	0.92	0.92	0.91
Average Naive Bayes	0.87	0.85	0.88	0.84
Average MLP	0.89	0.88	0.88	0.88
LightGBM				
Features selected (14/34), (16/34), (16/34), (13/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.82	0.78	0.85	0.76
Average SVM	0.95	0.95	0.96	0.94
Average Decision Tree	0.87	0.86	0.87	0.86
Average Random Forest	0.93	0.93	0.93	0.93
Average Naive Bayes	0.91	0.90	0.90	0.90
Average MLP	0.89	0.87	0.89	0.86
CatBoost				
Features selected (15/34), (10/34), (15/34), (11/34)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.86	0.84	0.89	0.82
Average SVM	0.95	0.94	0.96	0.93
Average Decision Tree	0.86	0.85	0.86	0.84

Average Random Forest	0.93	0.92	0.93	0.91
Average Naive Bayes	0.88	0.87	0.89	0.86
Average MLP	0.88	0.87	0.88	0.87

Πίνακας Α΄.60: Πίνακας αποτελεσμάτων ensemble learning Ionosphere Dataset Dataset

Connectionist Bench Dataset

Random Forest				
Features selected (20/60), (23/60), (23/60), (23/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.54	0.53	0.54	0.54
Average SVM	0.52	0.50	0.54	0.52
Average Decision Tree	0.59	0.58	0.60	0.59
Average Random Forest	0.61	0.59	0.64	0.61
Average Naive Bayes	0.54	0.52	0.58	0.55
Average MLP	0.60	0.58	0.64	0.60
AdaBoost				
Features selected (26/60), (27/60), (29/60), (24/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.55	0.55	0.56	0.55
Average SVM	0.60	0.59	0.61	0.60

Average Decision Tree	0.64	0.64	0.67	0.65
Average Random Forest	0.58	0.56	0.60	0.58
Average Naive Bayes	0.54	0.53	0.59	0.56
Average MLP	0.64	0.63	0.68	0.64
Gradient Boost				
Features selected (13/60), (16/60), (21/60), (15/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.54	0.53	0.54	0.54
Average SVM	0.56	0.54	0.62	0.57
Average Decision Tree	0.58	0.57	0.60	0.58
Average Random Forest	0.58	0.56	0.62	0.58
Average Naive Bayes	0.50	0.48	0.53	0.50
Average MLP	0.63	0.60	0.68	0.63
XGBoost				
Features selected (19/60), (27/60), (21/60), (23/60)				
Average KNN	0.52	0.51	0.53	0.52
Average SVM	0.52	0.49	0.56	0.52
Average Decision Tree	0.54	0.51	0.56	0.54
Average Random Forest	0.60	0.58	0.64	0.60

Average Naive Bayes	0.50	0.48	0.54	0.50
Average MLP	0.54	0.52	0.60	0.55
LightGBM				
Features selected (23/60), (25/60), (22/60), (25/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.52	0.51	0.52	0.52
Average SVM	0.58	0.56	0.61	0.58
Average Decision Tree	0.59	0.58	0.61	0.59
Average Random Forest	0.66	0.63	0.72	0.66
Average Naive Bayes	0.57	0.55	0.63	0.58
Average MLP	0.61	0.58	0.66	0.61
CatBoost				
Features selected (17/60), (24/60), (24/60), (23/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.55	0.54	0.55	0.55
Average SVM	0.60	0.58	0.63	0.60
Average Decision Tree	0.59	0.57	0.60	0.59
Average Random Forest	0.62	0.61	0.64	0.62
Average Naive Bayes	0.55	0.52	0.60	0.56
Average MLP	0.58	0.55	0.64	0.59

Πίνακας Α΄.61: Πίνακας αποτελεσμάτων ensemble learning Connectionist Bench Dataset

Dry Bean Dataset

Random Forest				
Features selected (10/16), (10/16), (11/16), (11/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.64	0.66	0.72	0.68
Average SVM	0.79	0.80	0.84	0.81
Average Decision Tree	0.41	0.42	0.46	0.48
Average Random Forest	0.42	0.43	0.49	0.48
Average Naive Bayes	0.79	0.80	0.84	0.81
Average MLP	0.71	0.73	0.78	0.76
AdaBoost				
Features selected (4/16), (3/16), (4/16), (4/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.56	0.58	0.66	0.60
Average SVM	0.75	0.75	0.80	0.76
Average Decision Tree	0.33	0.35	0.39	0.39
Average Random Forest	0.42	0.43	0.53	0.47
Average Naive Bayes	0.66	0.66	0.73	0.67

Average MLP	0.63	0.64	0.71	0.66
Gradient Boost				
Features selected (6/16), (8/16), (7/16), (7/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.69	0.71	0.77	0.73
Average SVM	0.77	0.79	0.83	0.80
Average Decision Tree	0.46	0.48	0.52	0.54
Average Random Forest	0.49	0.50	0.58	0.56
Average Naive Bayes	0.80	0.81	0.85	0.82
Average MLP	0.68	0.69	0.76	0.71
XGBoost				
Features selected (6/16), (7/16), (7/16), (8/16)				
Average KNN	0.63	0.65	0.71	0.67
Average SVM	0.79	0.80	0.84	0.81
Average Decision Tree	0.40	0.42	0.46	0.47
Average Random Forest	0.40	0.41	0.47	0.46
Average Naive Bayes	0.77	0.78	0.83	0.79
Average MLP	0.69	0.71	0.76	0.73
LightGBM				
Features selected (10/16), (10/16), (8/16), (8/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.84	0.87	0.89	0.87

Average SVM	0.84	0.86	0.89	0.86
Average Decision Tree	0.58	0.60	0.65	0.65
Average Random Forest	0.60	0.63	0.70	0.67
Average Naive Bayes	0.83	0.85	0.87	0.86
Average MLP	0.76	0.78	0.81	0.80
CatBoost				
Features selected (7/16), (6/16), (8/16), (8/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.67	0.69	0.75	0.71
Average SVM	0.77	0.79	0.83	0.79
Average Decision Tree	0.48	0.51	0.57	0.56
Average Random Forest	0.54	0.56	0.62	0.59
Average Naive Bayes	0.77	0.78	0.83	0.79
Average MLP	0.69	0.72	0.77	0.73

Πίνακας Α΄.62: Πίνακας αποτελεσμάτων ensemble learning Dry Bean Dataset

Musk Dataset

Random Forest				
Features selected (58/168), (48/168), (50/168), (53/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.73	0.72	0.73	0.73
Average SVM	0.73	0.71	0.74	0.72
Average Decision Tree	0.66	0.65	0.65	0.65
Average Random Forest	0.69	0.67	0.69	0.67
Average Naive Bayes	0.56	0.54	0.55	0.55
Average MLP	0.74	0.73	0.75	0.73
AdaBoost				
Features selected (64/168), (66/168), (62/168), (68/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.74	0.74	0.76	0.75
Average SVM	0.75	0.73	0.76	0.73
Average Decision Tree	0.60	0.59	0.60	0.60
Average Random Forest	0.69	0.67	0.70	0.68
Average Naive Bayes	0.62	0.61	0.62	0.61
Average MLP	0.75	0.74	0.76	0.75
Gradient Boost				
Average KNN	0.67	0.66	0.66	0.66
Average SVM	0.72	0.70	0.74	0.70

Average Decision Tree	0.64	0.63	0.63	0.63
Average Random Forest	0.70	0.68	0.71	0.69
Average Naive Bayes	0.58	0.57	0.57	0.57
Average MLP	0.73	0.72	0.74	0.72
XGBoost				
Features selected (58/168), (59/168), (57/168), (62/168)				
Average KNN	0.74	0.73	0.74	0.74
Average SVM	0.75	0.74	0.76	0.74
Average Decision Tree	0.68	0.67	0.68	0.68
Average Random Forest	0.71	0.69	0.72	0.70
Average Naive Bayes	0.59	0.58	0.58	0.58
Average MLP	0.77	0.76	0.77	0.76
LightGBM				
Features selected (64/168), (54/168), (59/168), (56/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.71	0.70	0.73	0.72
Average SVM	0.73	0.71	0.75	0.72
Average Decision Tree	0.63	0.62	0.63	0.63
Average Random Forest	0.71	0.70	0.72	0.70

Average Naive Bayes	0.59	0.58	0.59	0.58
Average MLP	0.78	0.77	0.78	0.77
CatBoost				
Features selected (59/168), (65/168), (52/168), (56/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.72	0.72	0.74	0.74
Average SVM	0.75	0.74	0.75	0.74
Average Decision Tree	0.66	0.64	0.65	0.65
Average Random Forest	0.71	0.70	0.71	0.70
Average Naive Bayes	0.62	0.61	0.62	0.62
Average MLP	0.77	0.77	0.77	0.77

Πίνακας Α΄.63: Πίνακας αποτελεσμάτων ensemble learning Musk Dataset

Α΄.12 Kendall's Rank Correlation

Wine Dataset				
Features selected (10/13), (10/13), (10/13), (3/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.95	0.95	0.95	0.96
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.84	0.83	0.86	0.84

Average Random Forest	0.94	0.94	0.94	0.95
Average Naive Bayes	0.94	0.94	0.95	0.94
Average MLP	0.96	0.96	0.96	0.97
Digits Dataset				
Features selected (1/64), (2/64), (2/64), (2/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.21	0.19	0.20	0.21
Average SVM	0.26	0.21	0.22	0.25
Average Decision Tree	0.24	0.23	0.24	0.24
Average Random Forest	0.24	0.23	0.24	0.24
Average Naive Bayes	0.22	0.16	0.15	0.22
Average MLP	0.24	0.21	0.22	0.24
Breast Cancer Dataset				
Features selected (25/30), (25/30), (25/30), (25/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.97	0.96	0.97	0.96
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.93	0.92	0.92	0.92
Average Random Forest	0.97	0.96	0.97	0.96
Average Naive Bayes	0.93	0.92	0.93	0.92

Average MLP	0.98	0.97	0.97	0.97
Ionosphere Dataset				
Features selected (7/30), (6/30), (6/30), (6/30)				
Average KNN	0.86	0.84	0.87	0.83
Average SVM	0.91	0.89	0.93	0.87
Average Decision Tree	0.88	0.87	0.89	0.86
Average Random Forest	0.91	0.91	0.92	0.90
Average Naive Bayes	0.88	0.86	0.92	0.84
Average MLP	0.88	0.87	0.88	0.87
Connectionist Bench Dataset				
Features selected (23/60), (18/60), (17/60), (30/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.63	0.60	0.64	0.62
Average SVM	0.60	0.57	0.66	0.60
Average Decision Tree	0.61	0.60	0.63	0.61
Average Random Forest	0.62	0.59	0.66	0.62
Average Naive Bayes	0.58	0.56	0.62	0.59
Average MLP	0.63	0.60	0.71	0.63
Dry Bean Dataset				
(7/16), (7/16), (7/16), (3/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.64	0.62	0.63	0.63

Average SVM	0.69	0.65	0.71	0.66
Average Decision Tree	0.65	0.64	0.65	0.65
Average Random Forest	0.62	0.61	0.61	0.61
Average Naive Bayes	0.61	0.60	0.60	0.60
Average MLP	0.61	0.59	0.60	0.59
Musk Dataset				
(11/168), (16/168), (6/168), (19/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.64	0.62	0.63	0.63
Average SVM	0.69	0.65	0.71	0.66
Average Decision Tree	0.65	0.64	0.65	0.65
Average Random Forest	0.62	0.61	0.61	0.61
Average Naive Bayes	0.61	0.60	0.60	0.60
Average MLP	0.61	0.59	0.60	0.59

Πίνακας Α΄.64: Πίνακας αποτελεσμάτων μέσω της μεθόδου Kendall's rank Correlation

Α΄.13 Spearman's Rank Correlation

Wine Dataset				
Features selected (10/13), (10/13), (10/13), (10/13)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.94	0.94	0.94	0.95
Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.89	0.89	0.90	0.89
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.93	0.93	0.94	0.94
Average MLP	0.96	0.96	0.96	0.96
Digits Dataset				
Features selected (29/64), (24/64), (27/64), (24/64)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.66	0.65	0.66	0.66
Average SVM	0.69	0.69	0.70	0.69
Average Decision Tree	0.56	0.56	0.57	0.57
Average Random Forest	0.71	0.71	0.71	0.71
Average Naive Bayes	0.31	0.22	0.27	0.31
Average MLP	0.68	0.67	0.69	0.68
Breast Cancer Dataset				
Features selected (21/30), (21/30), (21/30), (22/30)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.96	0.96	0.97	0.96

Average SVM	0.97	0.97	0.97	0.97
Average Decision Tree	0.91	0.91	0.90	0.91
Average Random Forest	0.96	0.96	0.96	0.96
Average Naive Bayes	0.94	0.93	0.93	0.93
Average MLP	0.97	0.97	0.97	0.97
Ionosphere Dataset				
Features selected (20/34), (25/34), (24/34), (27/34)				
Average KNN	0.83	0.79	0.87	0.78
Average SVM	0.95	0.94	0.96	0.93
Average Decision Tree	0.90	0.89	0.90	0.89
Average Random Forest	0.94	0.93	0.94	0.93
Average Naive Bayes	0.89	0.87	0.90	0.86
Average MLP	0.87	0.85	0.87	0.84
Connectionist Bench Dataset				
Features selected (17/60), (9/60), (10/60), (14/60)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.61	0.60	0.62	0.61
Average SVM	0.60	0.58	0.64	0.60
Average Decision Tree	0.61	0.60	0.62	0.61
Average Random Forest	0.61	0.59	0.67	0.61

Average Naive Bayes	0.57	0.56	0.57	0.57
Average MLP	0.60	0.58	0.63	0.60
Dry Bean Dataset				
Features selected (11/16), (11/16), (11/16), (11/16)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.64	0.62	0.63	0.63
Average SVM	0.69	0.65	0.71	0.66
Average Decision Tree	0.65	0.64	0.65	0.65
Average Random Forest	0.62	0.61	0.61	0.61
Average Naive Bayes	0.61	0.60	0.60	0.60
Average MLP	0.61	0.59	0.60	0.59
Musk Dataset				
Features selected (45/168), (33/168), (43/168), (43/168)				
Classifier	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Average KNN	0.72	0.72	0.73	0.72
Average SVM	0.74	0.73	0.75	0.73
Average Decision Tree	0.66	0.65	0.66	0.66
Average Random Forest	0.75	0.74	0.76	0.74
Average Naive Bayes	0.51	0.50	0.53	0.53
Average MLP	0.73	0.72	0.74	0.72

Πίνακας Α΄.65: Πίνακας αποτελεσμάτων μέσω της μεθόδου Spearman's rank Correlation