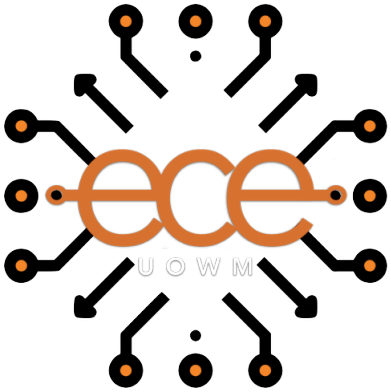


Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Αλγόριθμοι μηχανικής μάθησης υπό αβεβαιότητα

Μιχαήλ Παπαποστόλου (ΑΜ: 891)

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

Εργαστήριο Ευφυών Συστημάτων & Βελτιστοποίησης

28 Φεβρουαρίου 2024

Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη του πως υλοποιήσεις αλγορίθμων χαμηλής ασυμφωνίας βοηθούν στο πρόβλημα της αβεβαιότητας σε εναν χώρο δεδομένων και η σύγκριση της απόδοσης τριών κλασικών γραμμικών μοντέλων παλινδρόμησης, με αλλαγές σε δύο κύριες υπερ-παραμέτρους: την επιλογή πέντε διαφορετικών αλγορίθμων δειγματοληψίας και την επιλογή διαφορετικού αριθμού σημείων (500, 1,000 & 2,500). Οι αλγόριθμοι δειγματοληψίας χρησιμοποιούν τις ακολουθίες αυτές για την παραγωγή των δεδομένων και τα αποτελέσματα που προκύπτουν σχολιάζονται ως προς τις μετρικές R^2, RMSE, MSE και MAE. Συνολικά στην εργασία αυτή θα εξετάσουμε 45 συνδυασμούς μοντέλων και υπερ-παραμέτρων, ώστε να αξιολογήσουμε την απόδοση τους στα 579 προβλήματα που χρησιμοποιήθηκαν. Μελετώντας το σύνολο των αποτελεσμάτων μπορούμε να συμπεράνουμε με μεγάλη βεβαιότητα ότι η απόδοση των μοντέλων βελτιώνεται με την αύξηση του αριθμού των σημείων, ενώ σε όλες τις περιπτώσεις η παλινδρόμηση Lasso φαίνεται να έχει τη μεγαλύτερη διασπορά στις τιμές του R^2.

Λέξεις κλειδιά: Μηχανική μάθηση, γραμμικά μοντέλα παλινδρόμησης, τεχνικές δειγματοληψίας, υπερπαράμετροι.

Abstract

The purpose of this particular thesis is to study how implementations of known low-discrepancy sequences help with uncertainty issues in a set of data and to compare the efficacy of three classic linear regression model , when two hyperparameters of the problem are adjusted: namely the choice between one of five low discrepancy sequences for generating the dataset and the number of points to be generated (either 500, 1,000 or 2,500 points). These models are then evaluated using know regression metrics such as R^2, RMSE , MSE and MAE. In this paper, 45 possible combinations of models and hyperparameters will be examined, in order to judge their effectiveness on 579 mixed integer problems. By studying all the findings we deduced that model efficacy is increased when the number of generated points is also increased, and in all cases the LASSO regression model has greater variance of R^2 scores.

Keywords: Machine learning, linear regression models, sampling techniques, hyperparameter

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου κ. Πλόσκα Νικόλαο καθώς και τον υποψήφιο διδάκτορα Καραντουμάνη Εμμανουήλ για τη συμβολή τους στην εκπόνηση της διπλωματικής εργασίας. Υπήρξαν αρωγοί στις προσπάθειες μου με καίριες υποδείξεις καθ' όλη τη διάρκεια της εργασίας. Επιπλέον θα ήθελα να ευχαριστήσω θερμά τους γονείς μου οι οποίοι με τη στήριξη, τις αρχές και τους κόπους τους μου παρείχαν τη δύναμη και τα εφόδια για να ολοκληρώσω με επιτυχία τη διπλωματική εργασία και τις σπουδές μου.

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 ''Αλγόριθμοι μηχανικής μάθησης υπό αβεβαιότητα'' 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. Πλόσκα Νικόλαου αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε 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

Η αβεβαιότητα στη διακύμανση ενός συνόλου δεδομένων αποτελεί ένα μεγάλο πρόβλημα στον κλάδο της μηχανικής μάθησης. Η αβεβαιότητα αυτή έχει αντίκτυπο στη σταθερότητα και στην απόδοση των μοντέλων. Οι ερευνητές έχουν προτείνει διάφορες μεθόθους αντιμετώπισης του προβλήματος όπως bootstraping, Bayesian μεθόδους κ.α. Παρόλα αυτά η ποσοτικοποίηση της αβεβαιότητας στη διακύμανση σε ένα σύνολο δεδομένων παραμένει μια μεγάλη πρόκληση στον τομέα αυτό. Οι ημι-τυχαίες (quasi-random) μεθόδοι δειγματοληψίας στοχεύουν στο να παράξουν δεδομένα με ομαλή διακύμανση σε έναν δεδομένο χώρο και έτσι να βελτιώσουν την απόδοση και την αξιοπιστία των μοντέλων που χρησιμοποιούνται.

1.2 Κίνητρα και Στόχοι Υλοποίησης

Το κίνητρο για την ανάληψη και τη μελέτη ενός τέτοιου θέματος πηγάζει απο την επιθυμία βαθύτερης κατανόησης για τα εν γένει προβλήματα που δημιουργεί η αβεβαιότητα στη μηχανική μάθηση. Στόχος υλοποίησης αποτελεί η μελέτη και εξαγωγή συμπερασμάτων για την απόδοση γραμμικών μοντέλων μηχανικής μάθησης, χρησιμοποιώντας συγκεκριμένες ακολουθίες χαμηλής ασυμφωνίας για την παραγωγή των δεδομένων έτσι ώστε να ελαχιστοποιηθεί η αβεβαιότητα στη διακύμανση τους. Συγκεκριμένα θα αναζητήσουμε την ιδανική μέθοδο δειγματοληψίας για κάθε ένα απο τα τρία γραμμικά μοντέλα που θα χρησιμοποιήσουμε και θα μελετήσουμε εάν και πως η ύπαρξη διαφορετικών βαθμών θορύβου στα δεδομένα επηρεάζει την απόδοση των μοντέλων αυτών.

1.3 Διάρθρωση Κειμένου

Τα υπόλοιπα κεφάλαια της διπλωματικής εργασίας οργανώνονται ως εξής: Στο Κεφάλαιο 2 θα κάνουμε τη βιβλιογραφική ανασκόπηση της εργασίας, όπου θα παρουσιάσουμε όλες τις έννοιες που σχετίζονται με το θεωρητικό και το πρακτικό της υπόβαθρο, ξεκινώντας απο τις πιο γενικές και συνεχίζοντας στις ειδικότερες. Θα εξηγήσουμε λοιπόν τις επιστημονικές μεθόδους που αποτελούν τη βάση του πειράματος, τους τρόπους και τις μετρικές με τις οποίες θα αξιολογήσουμε τα αποτελέσματά μας.

Στο Κεφάλαιο 3 θα γίνει μία βιβλιογραφική ανασκόπηση μεθόδων που ελαχιστοποιούν την αβεβαιότητα.

Στο Κεφάλαιο 4 θα αναλύσουμε την υλοποίηση του πειράματος μας. Θα ξεκινήσουμε περιγράφοντας τα συγκεκριμένα εργαλεία και περιβάλλοντα ανάπτυξης που χρησιμοποιήθηκαν καθώς και το που χρησιμοποιούνται. Έπειτα θα δούμε πως υλοποιήσαμε αλγοριθμικά τους μαθηματικούς αλγορίθμους που χρειάστηκαν στο τμήμα της δειγματοληψίας, καθως και τη συνολική διαρρύθμιση του αλγορίθμου που μοντελοποιεί τα προβλήματά μας.

Στο Κεφάλαιο 5 θα εκτελέσουμε την υπολογιστική μελέτη της διπλωματικής εργασίας. Θα αναπαραστήσουμε γραφικά τα αποτελέσματα μας και θα περιγράψουμε αναλυτικά την ερμηνεία τους, παρέχοντας πληροφορίες και συμπεράσματα που ίσως δεν είναι ορατά εκ πρώτης όψεως.

Στο Κεφάλαιο 6 θα συνοψίσουμε τα αποτελέσματα της υπολογιστικής μελέτης, καθώς και τις επιλογές της υλοποίησης που οδήγησαν σε αυτά. Θα εξάγουμε ένα συνολικό πόρισμα για τα ευρήματα μας και θα προτείνουμε μια πιθανή κατεύθυνση που μπορεί να λάβει μια παρόμοια μελέτη μελλοντικά.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Σε αυτό το κεφάλαιο της διπλωματικής εργασίας θα παρατεθεί όλο το θεωρητικό υπόβραθρο που θα επικαλεσθεί τόσο για την υλοποίηση όσο και την αξιολόγηση του πειράματος. Θα εξηγηθούν εισαγωγικές έννοιες της Μηχανικής Μάθησης και στη συνέχεια θα αναλυθούν οι πιο συγκεκριμένες έννοιες που αποτελούν τον ακρογωνιαίο λίθο του προβλήματος που μελετάται.

2.1 Μηχανική Μάθηση (ΜΜ)

Η Μηχανική Μάθηση (ΜΜ) αποτελεί μια υποκατηγορία της Τεχνητής Νοημοσύνης, και επικεντρώνεται στην προσπάθεια απομίμησης του ανθρώπινου τρόπου εκμάθησης με τη χρήση αλγορίθμων και δεδομένων. Συγκεκριμένα, με τη χρήση της ΜΜ, κανείς μπορεί να προγραμματίσει έναν ηλεκτρονικό υπολογιστή να μάθει από διαθέσιμα σύνολα δεδομένων την επίλυση κάποιου προβλήματος. Ένας γενικός ορισμός της είναι:

> Η Μηχανική Μάθηση είναι το πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν προγραμματιστεί ρητά. **Arthur Samuel, 1959** [5]

Στη συνέχεια, παρατίθεται ενας ορισμός που είναι πιο προσανατολισμένος σε μηχανικούς:

> Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E. **Tom Mitchell, 1997** [6]

Ας εξετάσουμε τον δεύτερο ορισμό λίγο παραπάνω, χρησιμοποιώντας μια κοινή αλλά συχνά αμελητέα από πολλούς εφαρμογή της Μηχανικής Μάθησης, το φίλτρο ανεπιθύμητης αλληλογραφίας (spam filter). Ας υποθέσουμε ότι η εργασία "Τ" που πρέπει να πραγματοποιήσει το πρόγραμμα του φίλτρου είναι να χαρακτηρίζει νέα εισερχόμενη αλληλογραφία στο email μας ως επιθυμητή ή ανεπιθύμητη. Το πρόγραμμα θα πρέπει να χρησιμοποιήσει παραδείγματα που εμείς του παρέχουμε (π.χ. τύπους email που θεωρούμε spam), τα οποία ως σύνολο τα αποκαλούμε πειραματικά δεδομένα ή στην προκειμένη περίπτωση "εμπειρία Ε". Τέλος, πρέπει να ορίσουμε ένα μέτρο "Ρ" βάση του οποίου αξιολογούμε την επίδοση του προγράμματος, όταν εμείς του παρέχουμε νέα δεδομένα (μη-υπάρχοντα στα πειραματικά). Ένα τέτοιο μέτρο ενδεικτικά θα μπορούσε να είναι η αναλογία σωστά ταξινομημένων email. Το μέτρο αυτό ονομάζεται ευστοχία και χρησιμοποιείται συχνά σε προβλήματα ταξινόμησης (classification) [1].

2.1.1 Χρησιμότητα της ΜΜ

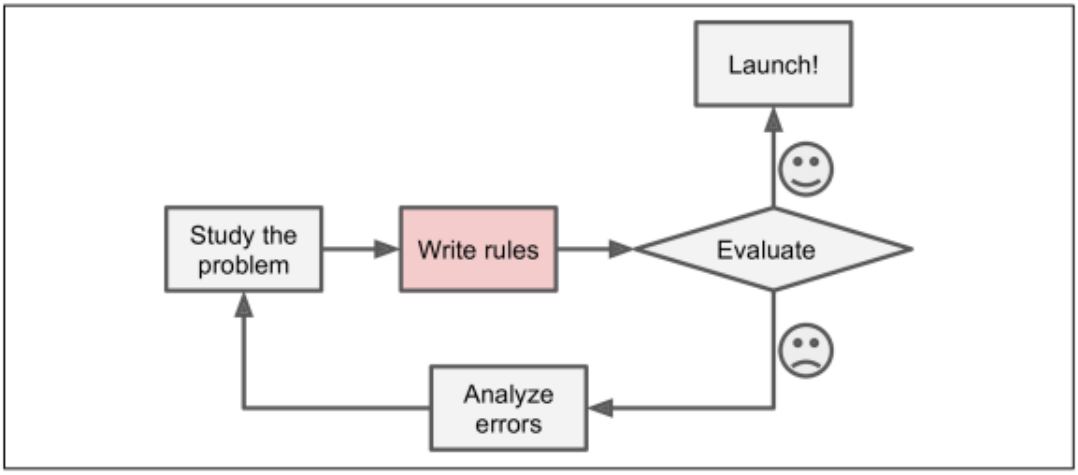
Για να κατανοήσουμε τη χρησιμότητα της ΜΜ, ας προσπαθήσουμε να δούμε πως θα λύναμε το πρόβλημα του φίλτρου spam με πιο κλασικές προγραμματιστικές μεθόδους (Σχήμα 2.1). Πιο συγκεκριμένα, θα ακολουθήσουμε τα παρακάτω βήματα:

1. Θα προσπαθήσουμε να ταυτοποιήσουμε διάφορες κοινές λέξεις ή φράσεις που έχουν τα email και χαρακτηρίζονται ως spam απο τους χρήστες (π.χ. "τρομερή ευκαιρία", "δωρεάν πιστωτική", κ.ο.κ.) ή άλλα κοινά χαρακτηριστικά (συνηθισμένος αποστολέας, συνηθισμένη δομή σώματος του email).

2. Θα γράψουμε έναν αλγόριθμο που να ανιχνεύει όλα αυτά τα λεκτικά μοτίβα και χαρακτηριστικά που ταυτοποιήσαμε και να χαρακτηρίζει τα συγκεκριμένα email ως spam ή μη spam.

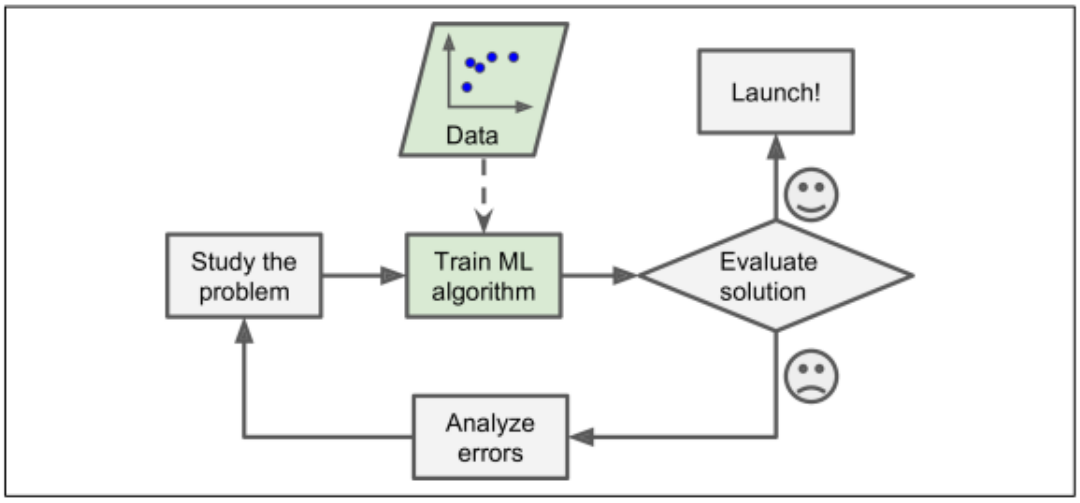
3. Θα ελέγχουμε την απόδοση του αλγορίθμου και θα τον αναβαθμίζουμε συνεχώς, μέχρι να φτάσει σε ένα ικανοποιητικό επίπεδο, επαναλαμβάνοντας τα βήματα 1 και 2.

Αν αναλογιστούμε τις πιθανές παγίδες και δυσκολίες ενός τέτοιου αλγορίθμου, προκύπτει ότι η δομή του θα καταλήξει να αποτελείται από μια μακροσκελή λίστα



Σχήμα 2.1: Διάγραμμα ροής της κλασικής αντιμετώπισης ενός προβλήματος [1]

από πολύπλοκους λεκτικούς κανόνες και περιπτώσεις, κάτι που τον καθιστά αρκετά δύσκολο και χρονοβόρο να συντηρηθεί και να αναβαθμιστεί. Αντίθετα, ένα φίλτρο spam που βασίζεται σε τεχνικές ΜΜ μαθαίνει αυτόματα ποιες λέξεις ή φράσεις προβλέπουν εύστοχα τον τύπο του email, ανιχνεύοντας ασυνήθιστα μοτίβα λέξεων στις περιπτώσεις spam, σε αντίθεση με τα επιθυμητά email (Σχήμα 2.2). Ο αλγόριθμος αυτός θα είναι αρκετά πιο σύντομος, κατανοητός και εύκολος στο να συντηρηθεί.



Σχήμα 2.2: Προσέγγιση μηχανικής μάθησης [1]

2.1.2 Επιβλεπόμενη και Μη-Επιβλεπόμενη Μάθηση

Με τη σειρά της και η ΜΜ διαχωρίζεται σε περαιτέρω κατηγορίες, οι οποίες επικεντρώνονται σε συγκεκριμένους τύπους προβλημάτων. Τα προβλήματα που επιδιώκει να λύσει η ΜΜ ποικήλουν και εξαρτώνται κυρίως από τη μορφή των

δεδομένων των οποίων μπορούμε να χρησιμοποιήσουμε για την εκμάθηση και το τελικό αποτέλεσμα που επιδιώκουμε. Ο πρώτος μεγάλος διαχωρισμός γινεται στη μορφή των δεδομένων εκπαίδευσης, για το αν γνωρίζουμε το επιθυμητό αποτέλεσμα των δεδομένων ή για το αν δεν το γνωρίσουμε και επιθυμούμε να ανακαλύψουμε τη καλύτερη δυνατή λύση. Αυτες οι δύο υποομάδες ονομάζονται επιβλεπόμενη και μη-επιβλεπόμενη μάθηση αντίστοιχα.

Η **Επιβλεπόμενη Μάθηση (Supervised Learning)** είναι μια ευρέως διαδεδομένη κατηγορία ΜΜ, στην οποία τα δεδομένα εκπαίδευσης είναι ένα σύνολο παραδειγμάτων, τα οποία ο αλγόριθμος αναλύει με σκοπό να παράγει ένα μοντέλο. Κάθε παράδειγμα αποτελείται από ένα σύνολο εισόδου, συνήθως ένα διάνυσμα των διαφόρων χαρακτηριστικών και την επιθυμητή έξοδο. Το μοντέλο αυτό θα χρησιμοποιηθεί από το πρόγραμμα για να μπορέσει να χαρακτηρίσει νέα δεδομένα που θα του χορηγηθούν. Η επιθυμητή έξοδος που συνοδεύει το διάνυσμα εισόδου ονομάζεται ταμπέλα (label) και το είδος των τιμών μπορεί να λάβει η ταμπέλα διαιρεί περαιτέρω τους τύπους προβλημάτων Επιβλεπόμενης Μάθησης (ΕΜ). Συγκεκριμένα, υπάρχουν δύο κλασικοί τύποι προβλημάτων Επιβλεπόμενης Μάθησης, η ταξινόμηση και η παλινδρόμηση, των οποίων η διαφορά έγκειται στο εάν η τιμή της ταμπέλας είναι διακριτή ή συνεχής [3]. Παρακάτω, αναλύεται ο κάθε τύπος προβλήματος ΕΜ αναλυτικά. Η ΕΜ έχει πολλές σημαντικές πρακτικές εφαρμογές που βελτιώνουν την καθημερινότητα μας, σε διάφορους τομείς όπως για παράδειγμα ο τομέας της υγείας.Ένα σωστά εκπαιδευμένο μοντέλο ΕΜ πάνω σε δεδομένα σχετικά με όγκους ασθενών μπορεί να δώσει μία έγκυρη πρόβλεψη σχετικά με το εάν ο όγκος ενός νέου ασθενή είναι καλοήθης ή κακοήθης βασισμένο σε διάφορα χαρακτηριστικά όπως η διαστάσεις του , το ιστορικό υγείας του ασθενούς κ.ο.κ.

Η **Μη-Επιβλεπόμενη Μάθηση (Unsupervised Learning)** είναι μια κατηγορία ΜΜ η οποία δεν έχει ταμπέλες στα δεδομένα εκπαίδευσης, σε αντίθεση με την Επιβλεπόμενη Μάθηση. Σκοπός της είναι να εντοπίσει ο υπολογιστής ένα υποκείμενο μοτίβο στα δεδομένα χωρίς να γνωρίζει τα δεδομένα εξόδου (labels). Οι πιο συνηθισμένες κατηγορίες της είναι οι παρακάτω:

1. Μη-επιβλεπόμενη μεταποίηση (Unsupervised Transformation)
2. Συσταδοποίηση (Clustering)

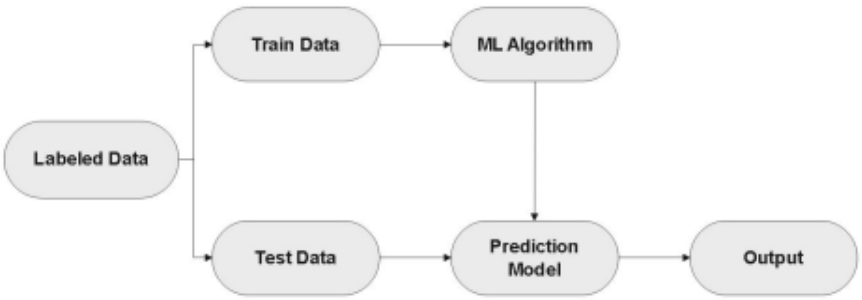
Η μη-επιβλεπόμενη μεταποίηση ενός συνόλου δεδομένων είναι τυπικά ένας αλγόριθμος που δημιουργεί μια νέα αναπαράσταση των δεδομένων, η οποία μπορεί να κατανοηθεί και να ερμηνευτεί ευκολότερα είτε από άλλους αλγορίθμους ΜΜ είτε από τον άνθρωπο. Μια κλασική εφαρμογή είναι η μείωση διαστάσεων (Dimensionality reduction), η οποία παίρνει μια σειρά δεδομένων με πληθώρα χαρακτηριστικών και βρίσκει ένα νέο τρόπο να αναπαραστήσει την πεμπτουσία τους, μειώνοντας το πλήθος των χαρακτηριστικών που χρειάζεται για να επιτευχθεί αυτό. Μια χρήση της μείωσης διαστάσεων είναι η μείωση σε δύο διαστάσεις, για να καταστεί δυνατή η γραφική αναπαράσταση ενός προβλήματος [3]. Για παράδειγμα, ένα IoT σύστημα θερμοκηπίου μπορεί να έχει πολλαπλά είδη αισθητήρων τα οποία δημιουργούν το σύνολο δεδομένων. Αυτό το σύνολο δεδομένων μπορεί εν τέλει να περιέχει πολλά χαρακτηριστικά N όπως θερμοκρασία, υγρασία κ.α. Αυτό μπορεί να δημιουργήσει πρόβλημα στη καλύτερη κατανόηση των δεδομένων, είτε από άλλους αλγορίθμους, ή από τον άνθρωπο. Έτσι η μείωση διαστάσεων μπορεί να εκφράσει τα ίδια σημεία των N διαστάσεων σε λιγότερες διαστάσεις, όπου η κάθε διασταση δεν εκφράζει ένα χαρακτηριστικό (π.χ. θερμοκρασία) αλλά εκφράζει όλα τα χαρακτηριστικά σε διαφορετικά επίπεδα. Από τους πιο γνωστούς αλγορίθμους μείωσης διαστάσεων είναι μέθοδος PCA (Principal Component Analysis), η οποία αρχικά βρίσκει τις ιδιοτιμές και τα ιδιοδιανύσματα του συνολου δεδομένων και χρησιμοποιεί τα ταξινομημένα (κατά ιδιοτιμή) ιδιοδιανύσματα ώς νέες διαστάσεις, και έπειτα μπορούμε να επιλέξουμε σε πόσες διαστάσεις θέλουμε να αναπαραστήσουμε τα δεδομένα.

Η συσταδοποίηση είναι μια κατηγορία αλγορίθμων που χρησιμοποιούνται με σκοπό την αναγνώριση ομάδων(συστάδων) στα δεδομένα. Στόχος είναι οι συστάδες αυτές να περιέχουν σημεία δεδομένων με παρόμοια χαρακτηριστικά και να διαφοροποιούνται σημαντικά με σημεία άλλων συστάδων. Ένα παράδειγμα είναι η ομαδοποίηση βιβλίων της βιβλιοθήκης ενός πανεπιστημίου με βάση τον κλάδο στον οποίο ανήκουν. Ένα παράδειγμα είναι η προσπάθεια ομαδοποίησης ταινιών σε πλατφόρμα παρακολούθησης, βάσει τη διάρκεια τους και τον αριθμό προβολών. Ένας αλγόριθμος ομαδοποίησης θα μπορούσε να μας βοηθήσει να ανακαλύψουμε τις συστάδες ταινιών, που μοιάζουν σε αυτά τα χαρακτηριστικά, έτσι ώστε να μπορέσουμε να κάνουμε καλύτερες προτάσεις στους χρήστες.

Η **Ενισχυτική Μάθηση (Reinforcement Learning)** είναι ένας κλάδος της μηχανικής μάθησης όπου ένας ευφυής πράκτορας αποσκοπεί να βελτιστοποιήσει τις ενέργειές του με βάση την ανατροφοδότηση από το περιβάλλον. Αυτό είναι κάτι που συνήθως μοντελοποιείται με τη χρήση διαδικασιών στοχαστικών αποφάσεων με κόστη και αποφάσεις, όπως οι μαρκοβιανές αλυσίδες, που σημαίνει ότι η μελλοντική κατάσταση εξαρτάται αποκλειστικά από την τρέχουσα κατάσταση και την ενέργεια, και όχι από την ακολουθία των γεγονότων που την προηγήθηκαν [7]. Ένα παράδειγμα της ενισχυτικής μάθησης είναι η εκπαίδευση ενός ευφυής πράκτορας για να παίξει σκάκι. ο πράκτορας πραγματοποιεί ενέργειες (κινήσεις) σε ένα περιβάλλον (σκακιέρα), και ο στόχος είναι να μεγιστοποιήσει την ανταμοιβή (να κερδίσει το παιχνίδι). Κάθε κίνηση στο παιχνίδι εξαρτάται από την τωρινή κατάσταση, και μπορεί να επιφέρει κάποιο κόστος (να χάσει κάποιο πιόνι) ή κέρδος (να διεκδικήσει πιόνι του αντιπάλου).

2.1.3 Τι είναι η Επιβλεπόμενη Μάθηση

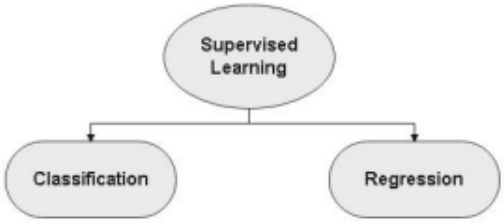
Η Επιβλεπόμενη Μάθηση βασίζεται στην ύπαρξη χαρακτηρισμένων δεδομένων, τα οποία θα υποστούν επεξεργασία με βάση το διάγραμμα ροής που βλέπουμε στο Σχήμα 2.3.



Σχήμα 2.3: Βασική αρχιτεκτονική Επιβλεπόμενης Μάθησης.[2]

Μετά τη συλλογή των δεδομένων, διαχωρίζονται στα δεδομένα εκμάθησης (training data) και στα δεδομένα αξιολόγησης (testing data). Έπειτα από την επεξεργασία των δεδομένων εκμάθησης από έναν αλγόριθμο το μοντέλο μαθαίνει να αναγνωρίζει τα χαρακτηριστικά που σχετίζονται με κάθε ταμπελα. Στη συνέχεια, παρέχουμε στο μοντέλο τα δεδομένα αξιολόγησης και το βάζουμε να μας προβλέψει την ταμπέλα για κάθε ένα από αυτά. Τέλος, συγκρίνουμε τα αποτελέσματα του μοντέλου

με τα πραγματικά αποτελέσματα (ταμπέλες) . Με αυτόν τον τρόπο μπορούμε να μετρήσουμε την απόδοση του μοντέλου με χρήση μετρικών αξιολόγησης όπως ορθότητα, ακρίβεια, ευαισθησία, εξειδίκευση. Στο Σχήμα 2.4 βλέπουμε τις δύο ευρείες υποκατηγορίες της Επιβλεπόμενης Μάθησης, την ταξινόμηση (Classification) και την παλινδρόμηση (Regression).



Σχήμα 2.4: Υποκατηγορίες Επιβλεπόμενης Μάθησης [2]

2.1.4 Ταξινόμηση (Classification)

Η Ταξινόμηση είναι μια βασική εργασία στην ΕΜ της οποίας ο στόχος είναι να χαρακτηρίσει με μια ταμπέλα(label) μια είσοδο βασιζόμενη στα χαρακτηριστικά της. Τα προβλήματα ταξινόμησης μπορούν να είναι δυαδικά, δηλαδή οι πιθανές διακριτές λύσεις να είναι ακριβώς δύο ή multiclass(πολλών κλάσεων), με περισσότερες από δύο λύσεις[8]. Μερικά κλασικά μοντέλα ταξινόμησης είναι επιγραμματικά:

1. Δέντρα Αποφάσεων (Decision Trees)
2. Κ-Πλησιέστεροι Γείτονες (K-Nearest Neighbors)
3. Τυχαίο Δάσος (Random Forest)
4. Support Vector Machines

Οι μετρικές που χρησιμοποιούνται για αυτά τα μοντέλα διαφέρουν από αυτές τις παλινδρόμησης και εστιάζουν πιο πολύ στο πώς το μοντέλο χωρίζει τον χώρο των δεδομένων σε εύστοχα διαμερίσματα σε αντίθεση με αυτά της παλινδρόμησης, τα οποία θα εξηγηθούν αναλυτικά αργότερα.

2.1.5 Παλινδρόμηση (Regression)

Η Παλινδρόμηση είναι μια από τις δύο κατηγορίες της ΕΜ στην οποία αφού εκπαιδεύσουμε το μοντέλο στα δεδομένα, αυτό μας δίνει προβλέψεις συνεχών τιμών

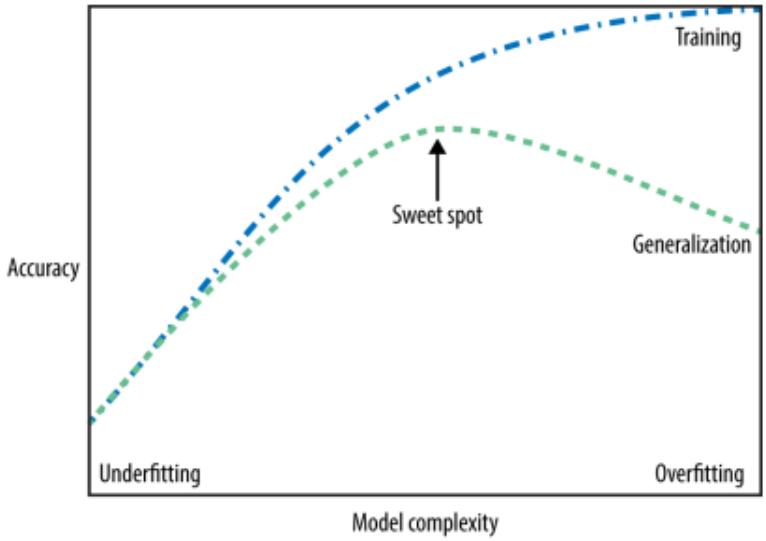
σε αντίθεση με την Ταξινόμηση που δίνει διακριτών. Για να αποσαφηνιστεί αυτή η πρόταση περαιτέρω, με τους όρους συνεχείς ή διακριτές δεν αναφερόμαστε απαραίτητα σε πραγματικούς και ακέραιους αριθμούς, αλλά ότι η πρόβλεψη βρίσκεται σε έναν συνεχές σύνολο τιμών και μπορεί να λάβει μια οποιαδήποτε απο αυτες.Ένα παράδειγμα παλινδρόμησης είναι η πρόβλεψη της ημερήσιας θερμοκρασίας, χρησιμοποιώντας τις τιμές των τελευταίων ημερών ή το ετήσιο εισόδημα ενός ανθρώπου βάσει του επαγγελματικού του ιστορικού.

2.1.6 Υπερπροσαρμογή - Υποπροσαρμογή

Στην ΜΜ πρέπει να δημιουργούνται μοντέλα τα οποία γενικέυουν όσο πιο εύστοχα γίνεται. Με τον όρο γενίκευση χαρακτηρίζεται η ικανότητα του μοντέλου να προβλέπει καλά τα νέα δεδομένα, έχοντας εξασκηθεί σε δεδομένα που του έχουν ήδη εισαχθεί. Θέλοντας να πετύχει αυτό το αποτέλεσμα κάποιος μπορεί να υποπέσει στις παγίδες είτε της υπερπροσαρμογής (Overfitting) είτε της υποπροσαρμογής (Underfitting) του μοντέλου στα δεδομένα. Στην πρώτη περίπτωση, το μοντέλο είναι υπερβολικά πολύπλοκο για τον όγκο των δεδομένων εξάσκησης που έχει δεχθεί, το οποίο σημαίνει ότι θα τα πάει εξαιρετικά καλά στα δεδομένα εξάσκησης έχοντας εστιάσει πολύ στις ιδιαιτερότητές τους, αλλά θα αποτύχει στα δεδομένα αξιολόγησης, διότι δεν θα έχει συνυπολογίσει τη φυσική διακύμανση που έχουν τα στοιχεία οποιουδήποτε συνόλου δεδομένων. Στη δεύτερη περίπτωση, επιλέγεται ένα υπερβολικά απλό μοντέλο το οποίο αποτυγχάνει πλήρως να καταλάβει τη συσχέτιση της εισόδου με την έξοδο και έχει κακή απόδοση τόσο στα δεδομένα εξάσκησης όσο και σε αυτά της αξιολόγησης. Για παράδειγμα είναι αδύνατον να περιμένουμε ένα γραμμικό μοντέλο να έχει καλή απόδοση σε δεδομένα που γνωρίζουμε ότι δεν έχουν γραμμική συσχέτιση μεταξύ τους. Αποτελεί λοιπόν απαραίτητη πρακτική η επιλογή ενός μοντέλου το οποίο βρίσκεται στη χρυσή τομή ύπαρξης και απουσίας πολυπλοκότητας, για να έχει τη βέλτιστη απόδοση όταν γενικεύει (Σχήμα 2.5).

2.1.7 Γραμμικά Μοντέλα - Linear Regression

Το πιο γνωστό και βασικό γραμμικό μοντέλο στην ΕΜ είναι η απλή Γραμμική Παλινδρόμηση (Simple Linear Regression). Η χρήση της είναι η πρόβλεψη μιας ποσότητας Υ η οποία σχετίζεται με μια ανεξάρτητη μεταβλητή Χ. Σαν μέθοδος λαμβάνει



Σχήμα 2.5: Η ιδανική αναλογία πολυπλοκότητας του μοντέλου με την απόδοση στα δεδομένα εξάσκησης και αξιολόγησης [3]

ως δεδομένο ότι το Υ με το Χ συσχετίζονται γραμμικά. Η μαθηματική έκφραση της απλής γραμμικής παλινδρόμησης φαίνεται στην Εξίσωση 2.1 και αποτελεί μια απλή ευθεία.

$$Y = \beta_0 + \beta_1 X \tag{2.1}$$

όπου β_1 και β_0 είναι οι δύο παράμετροι που μας είναι άγνωστοι. Αυτές τις δύο παραμέτρους τις ονομάζουμε κλίση της ευθείας (β_1) και κατακόρυφη μετατόπισή της (β_0) [9]. Αλγοριθμικά η Γραμμική Παλινδρόμηση θα προσπαθεί με βάση τα δοθέντα σε αυτή δεδομένα να δώσει μια πρόβλεψη για τις τιμές των αγνώστων παραμέτρων β_1 και β_0 έτσι ώστε να μπορεί να μαντέψει την τιμή του Y όταν της δοθούν άγνωστα δεδομένα. Φυσικά τα περισσότερα προβλήματα στον πραγματικό κόσμο περιέχουν παραπάνω από ένα ανεξάρτητο χαρακτηριστικό, όπου και χρησιμοποιούμε την εκτεταμένη εκδοχή της μεθόδου που ονομάζεται Πολλαπλή Γραμμική Παλινδρόμηση (Multiple Linear Regression). Η μαθηματική της έκφραση φαίνεται στην Εξίσωση 2.2.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \tag{2.2}$$

Η σημαντική διαφορά αυτού του τύπου είναι ότι ερμηνεύουμε το εκάστοτε β_p ως τη μέση επίδραση του χαρακτηριστικού X_p στο Y κρατώντας τα υπόλοιπα χαρακτη-

ριστικά σταθερά [9]. Αυτό γίνεται διότι αναγνωρίζουμε την επιρροή που ασκούν όλα τα χαρακτηριστικά ταυτόχρονα σε ένα πρόβλημα, επομένως θα ήταν άτοπο να τα μελετήσουμε ξεχωριστά (μοντελοποιώντας π.χ. μια απλή γραμμική παλινδρόμηση για κάθε ένα X_p σύμφωνα με την Εξίσωση 2.1).

2.1.8 Κανονικοποιημένα Γραμμικά Μοντέλα

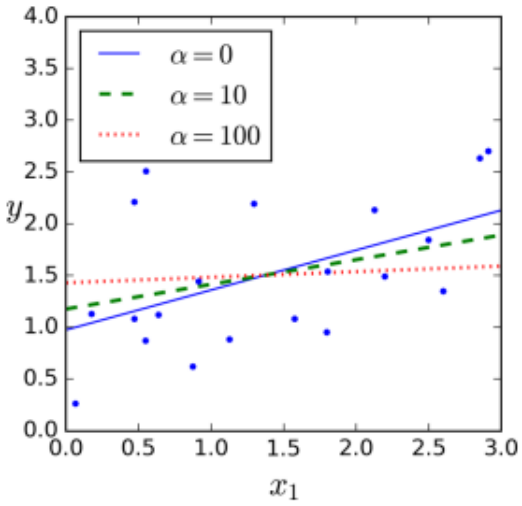
Μια βασική τεχνική για να αποφευχθεί η υπερπροσαρμογή του μοντέλου στα δεδομένα είναι η κανονικοποίηση. Με αυτή την τεχνική μειώνεται το μέγεθος της κάθε παραμέτρου ενός προβλήματος τείνοντας προς το μηδέν, με αποτέλεσμα να ελαχιστοποιείται η επιρροή οποιασδήποτε μοναδικής παραμέτρου στο αποτέλεσμα. Αυτός ο περιορισμός, ανάλογα με τον τρόπο που μετατρέπει την εξίσωση της Γραμμικής Παλινδρόμησης, μας οδηγεί στις δύο κανονικοποιημένες εκδοχές της, την Παλινδρόμηση Ράχης (Ridge Regression) και την Παλινδρόμηση LASSO (Lasso Regression).

Παλινδρόμηση Ράχης (Ridge Regression)

Η **Παλινδρόμηση Ridge (Ridge Regression)**, γνωστή επίσης και ως Κανονικοποίηση Tikhonov, είναι η πρώτη τεχνική η οποία προσθέτει εναν όρο κανονικοποίησης στη συνάρτηση κόστους της Γραμμικής Παλινδρόμησης. Προγραμματιστικά στην Python αυτό αναπαριστάται με μια παραπάνω παράμετρο 'α' η οποία λαμβάνει μη-αρνητικές τιμές και ελέγχει το βαθμό στον οποίο κανονικοποιείται το μοντέλο. Αν το α ισούται με μηδέν τότε ουσιαστικά έχουμε απλά Γραμμική Παλινδρόμηση, ενώ αν το α ισούται με έναν πολύ μεγάλο αριθμό τότε όλες οι παράμετροι συγκλίνουν προς το μηδέν. Ως αποτέλεσμα προκύπτει μια ευθεία που περνά από τη μέση τιμή των δεδομένων (Σχήμα 2.6).

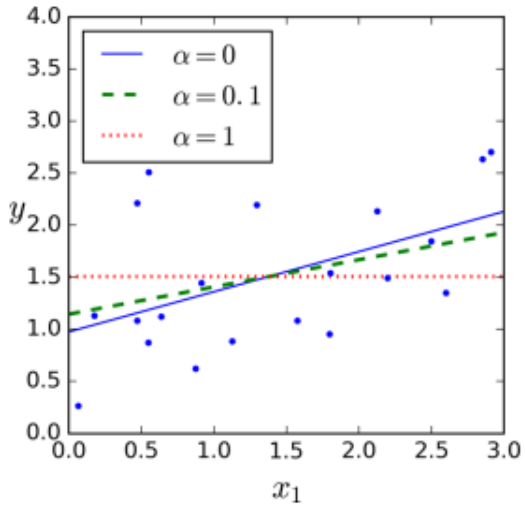
Παλινδρόμηση LASSO (Lasso Regression)

Η **Παλινδρόμηση LASSO (Least Absolute Shrinkage and Selection Operator Regression)** όπως και η ΠΡ αποτελεί μια κανονικοποιημένη εκδοχή της Γραμμικής Παλινδρόμησης, η οποία διαφέρει απλά ως προς τον όρο κανονικοποίησης που προσθέτει. Προγραμματιστικά λειτουργεί, όπως και στη ΠΡ, με μια παράμετρο 'α' όμως η διαφορά είναι ότι τείνει να θέσει τις παραμέτρους των λιγότερο σημαντικών



Σχήμα 2.6: Παλινδρόμηση Ράχης [1]

χαρακτηριστικών στο μηδέν, δημιουργώντας ένα πιο αραιό μοντέλο. Η κανονικοποίηση αυτή όπως και στην περίπτωση της Ridge Regression έχει ως αποτέλεσμα τον περιορισμό της υπερπροσαρμογής.



Σχήμα 2.7: Παλινδρόμηση Lasso [1]

Αυτά τα τρία μοντέλα θα χρησιμοποιηθούν αργότερα, στην υλοποίηση του πειράματος.

2.1.9 Μετρικές Αξιολόγησης

Οι μετρικές αξιολόγησης αλγορίθμων ΜΜ ποικίλουν και μας προσφέρουν πλήθος πληροφοριών ως προς τις διαφορές πτυχές της απόδοσης τους. Στο πείραμα θα μας απασχολήσουν τρεις. Οι R^2 **(Coefficient of Determination)**, **RMSE (Root Mean**

Squared Error), **MSE (Mean Squared Error)**, και **MAE (Mean Absolute Error)**. Συγκεκριμενα:

R^2 (Coefficient of Determination)

Το R^2 είναι ένα στατιστικό μέτρο που αντιπροσωπεύει τη βαθμίδα της διακύμανσης στην εξαρτημένη μεταβλητή η οποία εξηγείται απο τις ανεξάρτητες μεταβλητές σε ένα μοντέλο παλινδρόμησης. Το σύνολο τιμών του ειναι το [0,1] όπου το 0 υποδεικνύει ότι το μοντέλο αποτυγχάνει πλήρως στην εργασία που εξηγήθηκε μόλις , ενώ το 1 το αντίθετο. Επομένως επιθυμητό είναι το υψηλό R^2 στα μοντέλα μας [8].

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}}, \tag{2.3}$$

$$= 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{2.4}$$

MSE (Mean Squared Error)

Το MSE αντιπροσωπεύει την τετραγωνισμένη διαφορά των τιμών που έχουν προβλεφθεί από το μοντέλο με τις πραγματικές τιμές σε ένα πρόβλημα παλινδρόμησης. Ποσοτικοποιεί την απόδοση ενός μοντέλου πρόβλεψης όπως και το rmse. Το MSE χρησιμοποιείται συχνά σε διαδικασίες βελτιστοποίησης καθώς παρέχει μια διαφορίσιμη και συνεχή μέτρηση της απόδοσης του μοντέλου [8]

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.5}$$

Όπου n το πλήθος των δεδομένων $_i$ οι πραγματικές τιμές και $\hat{Y}_i$ οι τιμές που προβλέφθηκαν.

RMSE (Root Mean Squared Error)

Το RMSE πρόκειται ουσιαστικά για το MSE με μια τετραγωνική ρίζα και είναι ένα μέτρο της μέσης απόκλισης των δύο τιμών. Υπολογίζεται παίρνοντας την τετραγωνική ρίζα της μέσης τιμής της τετραγωνισμένης διαφοράς μεταξύ της τιμής που

προβλέφθηκε και της πραγματικής. Εκφράζεται στη ίδια μονάδα με την εξαρτημένη μεταβλητή κάτι που την κάνει εύκολο μέτρο στην ερμηνεία. Οι μικρότερες δυνατές τιμές είναι επιθυμητές εδώ.

$$RMSE = \sqrt{MSE} \tag{2.6}$$

MAE (Mean Absolute Error)

Το MAE είναι η μετρική που μας ενημερώνει για τη μέση απόκλιση των προβλέψεων του μοντέλου, έναντι των πραγματικών τιμών. Όπως είναι λογικό, επιθυμούμε χαμηλές τιμές MAE από το μοντέλο μας.

$$MAE = \sum_{i=1}^{n} |Y_i - \hat{Y}_i| \tag{2.7}$$

2.2 Ακολουθίες Χαμηλής Ασυμφωνίας

Οι ακολουθίες χαμηλής ασυμφωνίας (Low-Discrepancy Sequences) γνωστές και ως ημι-τυχαίες η υπο-τυχαίες ακολουθίες είναι ακολουθίες των οποίων η χαρακτηριστική ιδιότητα ότι έχουν ομοιόμορφη κατανομή των όρων τους στον χώρο τον οποίο ορίζονται. Συχνά χρησιμοποιούνται ως μέθοδοι δειγματοληψίας οταν πραγματοποιείται βελτιστοποίηση δίχως παραγώγους (Derivative Free Optimization) διότι η ομοιόμορφη κατανομή τους βοηθάει στην αποτελεσματική έρευνα του χώρου τον παραμέτρων και γρηγορότερη σύγκλιση προς την βέλτιστη λύση του προβλήματος [10].

Για να γίνει κατανοητός ο τρόπος με τον οποίο οι ακολουθίες αυτές συμπεριφέρονται μαθηματικά, παρατίθεται ο ορισμός της συνάρτησης βάση της οποίας θα επεκταθεί η έννοια της κάθε ακολουθίας παρακάτω.

Έστω για ακέραιους $i \geq 0$ και $b \geq 2$, ορίζεται η συνάρτηση $\phi_b(i)$ ως εξής:
Αν $i = \sum_{\alpha=1}^{\infty} i_\alpha b^{\alpha-1}$, όπου $i_\alpha \in \{0, 1, \ldots, b-1\}$, τότε $\phi_b(i) \coloneqq \sum_{a=1}^{\infty} \frac{i_\alpha}{b_\alpha}$
Επομένως, αν $i = (\ldots i_2 i_1)_b$ συμβολίζει την αναπαράσταση σε βαση-b του i, τότε $\phi_b(i) = (0.i_1 i_2 \ldots..)_b$, οπότε η συνάρτηση αντιστρέφει την αναπαράσταση.

2.2.1 Ακολουθία Van der Corput

Η ακολουθία Van der Corput είναι η απλούστερη μονοδιάστατη ακολουθία χαμηλής ασυμφωνίας που προτάθηκε από τον ομώνυμο Ολλανδό μαθηματικό το 1935 [11]. Η ακολουθία αυτή αντιστρέφει την αναπαράσταση σε βάση-b των αριθμών της ακολουθίας. Ουσιαστικά είναι η ακολουθία

$$\phi_b(0), \quad \phi_b(1), \quad \phi_b(2), \quad \dots \tag{2.8}$$

Για παράδειγμα, έχοντας βάση $b = 2$, γράφονται οι φυσικοί αριθμοί $0, 1, 2, ...$ σε βάση 2.

$$0, \quad 1_2, \quad 10_2, \quad \dots \tag{2.9}$$

Μετά εφαρμόζουμε τη συνάρτηση ϕ_2 σε κάθε αριθμό και προκύπτει η ακολουθία:

$$0, \quad 0.1_2, \quad 0.01_2, \quad \dots \tag{2.10}$$

Η οποία σε δεκαδικό σύστημα είναι:

$$0, \quad 0.5, \quad 0.25, \quad \dots \tag{2.11}$$

2.2.2 Ακολουθία Halton

Η ακολουθία Halton είναι η υλοποίηση της Van der Corput σε υψηλότερο αριθμό διαστάσεων. Συνήθως τέτοιες ακολουθίες αποδίδουν σε χαμηλό αριθμό διαστάσεων, αλλα προβλήματα συσχέτισης έχουν παρατηρηθεί μεταξύ ακολουθιών που παράγονται από μεγαλύτερους πρώτους αριθμούς [12]. Για να παραχθεί η ακολουθία Halton χρειάζονται τα εξής:

Έστω $p_1, \quad p_2, \quad \dots, \quad p_s$ οι πρώτοι s σε σειρά πρώτοι αριθμοί. Η ακολουθία Halton $t_0, \quad t_1, \quad \dots$ σε s διαστάσεις δίνεται από:

$$t_i = (\phi_{p_1}(i), \phi_{p_2}(i), \dots, \phi_{p_s}(i)), \quad i = 0, 1, \dots, \tag{2.12}$$

Όπου κάθε ένα από τα i στοιχεία της ακολουθίας έχει s διαστάσεις [13].

$$
\begin{aligned}
t_0 &= (0, \quad 0, \quad 0, \quad \ldots, \quad 0), \\
t_1 &= (0.1_2, \quad 0.1_3, \quad 0.1_5, \quad \ldots, \quad 0.1_{p_s}), \\
t_2 &= (0.01_2, \quad 0.2_3, \quad 0.2_5, \quad \ldots, \quad 0.2_{p_s}), \\
t_3 &= (0.11_2, \quad 0.01_3, \quad 0.3_5, \quad \ldots, \quad 0.3_{p_s}), \\
&\vdots
\end{aligned}
$$

2.2.3 Ακολουθία Hammersley

Η ακολουθία Hammersley είναι αρκετά παρόμοια με τη Halton με μοναδική διαφορά την απόδοση της σε υψηλότερο αριθμό διαστάσεων. Αυτό προκύπτει από τον τρόπο κατασκευής της:

Έστω $p_1, \quad p_2, \quad \ldots, \quad p_{s-1}$ οι πρώτοι $s-1$ σε σειρά πρώτοι αριθμοί. Η ακολουθία Hammersley $t_0, \quad t_1, \quad \ldots, \quad t_{n-1}$ με n ορους σε s διαστάσεις δίνεται από:

$$
t_i = (\frac{i}{n}, \phi_{p_1}(i), \phi_{p_2}(i), \ldots, \phi_{p_{s-1}}(i)), \quad i = 0, 1, \ldots, n-1 \tag{2.13}
$$

Εδώ φαίνεται ότι για κάθε όρο της ακολουθίας, στην πρώτη του διάσταση δεν εφαρμόζεται η συνάρτηση φp(i) και ο όρος προκύπτει από μια απλή διαίρεση. Έτσι πετυχαίνουμε όρους n όρους s διαστάσεων με n-1 περάσματα της ακολουθίας [13].

$$
\begin{aligned}
t_0 &= (0, \quad 0, \quad \ldots, \quad 0), \\
t_1 &= (\frac{1}{n}, \quad 0.1_2, \quad 0.1_3, \quad \ldots, \quad 0.1_{p_{s-1}}), \\
t_2 &= (\frac{2}{n}, \quad 0.01_2, \quad 0.2_3, \quad \ldots, \quad 0.2_{p_{s-1}}), \\
&\vdots \\
t_3 &= (\frac{n-1}{n}, \quad \ldots)
\end{aligned}
$$

2.2.4 Ακολουθία Sobol

Η ακολουθία Sobol είναι μια μέθοδος που χρησιμοποιείται για να παράγει μια ακολουθία αριθμών που είναι ομοιόμορφα διαμοιραζόμενοι σε έναν πολυδιάστατο

χώρο. Η κεντρική ιδέα της ακολουθίας είναι η στοχευμένη επιλογή σημείων με τρόπο έτσι ώστε ο χώρος να γεμίσει με τον πιο ομοιόμορφο τρόπο. Αυτό επιτυγχάνεται χρησιμοποιώντας ένα σύνολο αριθμών διεύθυνσης και μια μέθοδο που ονομάζεται ταξινόμηση κώδικα Gray, η οποία βεβαιώνει οτι τα διπλανά σημεία της ακολουθίας είναι όσο πιο ανόμοια γίνεται. Οι αριθμοί διεύθυνσης μας δίνουν τις αποστάσεις των σημείων σε κάθε διάσταση ενώ η ταξινόμηση κώδικα Gray φροντίζει οι δείκτες των διπλανών στοιχείων να διαφέρουν κατα ένα bit ακριβώς,το οποίο με τη σειρά του αποτρέπει την δημιουργία συστάδων μεταξύ σημείων μειώνοντας τη συσχέτισή τους [14].

2.2.5 Ακολουθία Latin Random

Η ακολουθία Latin Random, αναφέρεται στην ακολουθία η οποία αποτελείται από νούμερα που μοιάζουν να είναι τυχαία αλλά παράγονται από μια ντετερμινιστική διαδικασία. Η διαδικασία, με τη χρήση μιας αρχικής (seed) τιμής παράγει αυτές τις Latin ακολουθίες. Μια Latin ακολουθία προέρχεται από ένα Latin τετράγωνο. Ένα Latin τετράγωνο της τάξης n, με στοιχεία ενός συνόλου , είναι ένας πίνακας L μεγέθους $n \times n$ όπου κάθε στοιχείο του πίνακα περιέχει κάποιο στοιχείο του συνόλου X, έτσι ώστε κάθε γραμμή του L να είναι μια παραλλαγή του x και κάθε στήλη του L να είναι μια παραλλαγή του L [15].

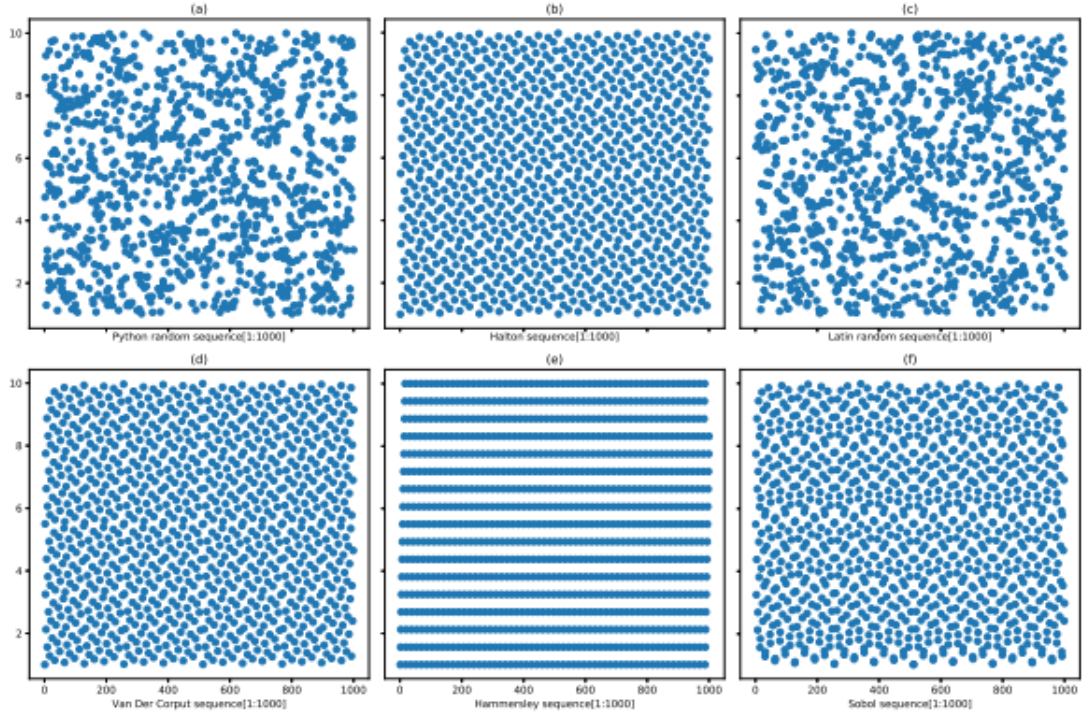
2.2.6 Παρουσίαση Ακολουθιών

Η ακολουθίες αυτές είναι όλες ακολουθίες χαμηλής ασυμφωνίας, οι οποίες χρησιμοποιούνται για να παράγουν στοιχεία, τα οποία κατανεμονται ομοιόμορφα. Η κάθε ακολουθία έχει τα μειονεκτήματα και τα πλεονεκτήματά της, καθώς προσπαθούν να γεμίσουν τον χώρο με διαφορετικές στρατηγικές. Όπως μπορούμε να δούμε στο Σχήμα 2.8 είναι ξεκάθαρες οι διαφορές και ομοιότητες των μεθόδων.

Αρχικά, μπορούμε να διακρίνουμε τις μεθόδους σε τρεις ομάδες βάσει τις ομοιότητες τους. Η πρώτη ομάδα που διακρίνεται είναι η τυχαία παραγωγή σημείων και η Latin random. Εδώ φαίνεται ξεκάθαρα ότι πράγματι η Latin random, μοιάζει να παράγει τυχαία σημεία, εφόσον μοιάζει τόσο με τη random, παρόλα αυτά γνωρίζουμε ότι παράγονται με ντετερμινιστικό τρόπο. Η άλλη ομάδα ακολουθιών που διακρίνεται βάσει των ομοιοτήτων, είναι οι Van der Corput, Sobol, και Halton. Η

Halton και η Van der Corput, είναι λογικό να μοιάζουν αρκετά καθώς η μια προέρχεται από την άλλη, ενώ είναι ενδιαφέρων, ότι, τουλάχιστον για δύο διαστάσεις και η Sobol φαίνεται να μοιάζει αρκετά με τις άλλες δύο. Τέλος, στην τελευταία ομάδα που διακρίνουμε, η Hammersley μοιάζει να είναι πολύ διαφορετική από τις υπόλοιπες, ωστόσο, πρέπει να σημειώσουμε ότι χρησιμοποιείται σε περιπτώσεις με πολλές διαστάσεις και γνωρίζουμε ότι η μία διάστασή της ακολουθεί κανονική ομοιόμορφη κατανομή. Στην περίπτωση αυτή, με δύο διαστάσεις αυτό είναι ξεκάθαρο.

Συμπεραίνουμε λοιπόν, ότι η κάθε μέθοδος δημιουργίας ακολουθιών, έχει πλεονεκτήματα και μειονεκτήματα, και είναι σημαντικό να αντιλαμβανόμαστε τις διαφορές τους όταν τις χρησιμοποιούμε και για τι είδους προβλήματος τις χρησιμοποιούμε.



Σχήμα 2.8: Σύγκριση της διασποράς του random package της Python και ημι-τυχαίων μεθόδων δειγματοληψίας: (a) Python random package, (b) Ακολουθία Halton, (c) Ακολουθία, (d) Ακολουθία Van der Corput, (e) Ακολουθία Hammersley, (f) Ακολουθία Sobol [4]

2.3 Βελτιστοποίηση Δίχως Παραγώγους (Derivative Free Optimization)

Η Βελτιστοποίηση δίχως παραγώγους (Derivative Free Optimization) ή DFO για συντομία, είναι μια πρακτική της μαθηματικής βελτιστοποίησης η οποία δε χρησιμοποιεί παραγώγους από την αντικειμενική συνάρτηση f του εκάστοτε προβλή-

ματος είτε επειδή αυτά δεν είναι αναλυτικά διαθέσιμα (Black-Box Optimization) ή επειδή είναι αναξιόπιστα. Το εύρος των εφαρμογών αυτής της πρακτικής είναι πραγματικά εντυπωσιακό, από προβλήματα σχεδιασμού ροτόρων και ευαίσθητων κυκλωμάτων έτσι ώστε να αποδίδουν βέλτιστα μέχρι την μελέτη της γεωμετρίας ατόμων, υπόγειων υδάτων κ.ο.κ. Μέχρι πρόσφατα η βελτιστοποίηση προβλημάτων με παραπάνω απο δέκα ανεξάρτητα χαρακτηριστικά θεωρούταν κοστοβόρα, παρόλα αυτά νέες μέθοδοι [16] φαίνεται να μπορούν να διαχειριστούν προβλήματα με εκατοντάδες μεταβλητες [17]. Στην εργασία θα μελετήσουμε το πως διαφορετικές πρακτικές δειγματοληψίας που χρησιμοποιούνται στην DFO επιρρεάζουν την αβεβαιότητα στα δεδομένα.

2.4 Αβεβαιότητα

Η Αβεβαιότητα στην ΜΜ αναφέρεται στην έλλειψη παντελούς σιγουριάς ως προς τις προβλέψεις η αποφάσεις που λαμβάνει ένα μοντέλο. Η αβεβαιότητα αυτη μπορεί να πηγάζει απο διάφορα σημεία όπως δεδομένα με πολύ θόρυβο ή περιορισμένες πληροφορίες. Στα μοντέλα παλινδρόμησης, η αβεβαιότητα στην διακύμανση των δεδομένων στον χώρο δημιουργεί πρόβλημα στον χαρακτηρισμό ενός μοντέλου ως έμπιστο ή οχι [18]. Οι ακολουθίες χαμηλής ασυμφωνίας στην διαδικασία της δειγματοληψίας έχουν προταθεί ως ικανή μέθοδος ελαχιστοποίησης του ζητήματος [19].

Κεφάλαιο 3

Βιβλιογραφική Ανασκόπηση

Στη μελέτη των Robinson et al. [19] μελετάται μια εναλλακτική στις ψευδοτυχαίες (pseudo-random) μεθόδους δειγματοληψίας οι οποίες αποτελούν τη γνωστική βάση της μεθόδου Monte Carlo. Προτείνουν μια ντετερμινιστική εκδοχή της μεθόδου γνωστή και ως ημί-τυχαίες (quasi-random) μέθοδοι Monte Carlo. Για την μέθοδο αυτή χρησιμοποιούν υπαρκτές ακολουθίες χαμηλής ασυμφωνίας για να παράξουν τα δεδομένα τους. Αυτή η μελέτη εστιάζει κυρίως στην χρήση της ακολουθίας Halton η οποία προτείνουν οτι δείχνει πολυ πιθανό να έχει εξαιρετικά ομαλή διασπορά στα δεδομένα. Συμπέραναν οτι η ακολουθία αυτή καθώς και οι παραλαγές οι οποίες προτάθηκαν έχουν αρκετά γρήγορο χρόνο εκτέλεσης σε προβλήματα όπου ο όγκος των δειγμάτων καθώς και ο αρθιμός διαστάσεων είναι μικρός. Ισχυρίζονται οτι υπάρχει θέση για τις ημι-τυχαίες μεθόδους Monte Carlo στην ανάλυση αβεβαιότητας.

Εξίσου ενδιαφέρουσα είναι και η μελέτη των Kocis et al. [20] στην οποία μελετάται η απόδοση των ακολουθιών χαμηλής ασυμφωνίας όταν τα δεδομένα που αυτές πρέπει να παράξουν έχουν υψηλό αριθμό διαστάσεων (από 40 έως και 400). Συμπεραίνουν οτι η ακολουθία Halton έχει καλή απόδοση μόνο σε χαμηλό αριθμό διαστάσεων οπως και στην προηγούμενη μελέτη που αναφέρθηκε, προτείνουν δε κάποιες αλλαγές σε μια προσπάθια βελτίωσης της. Οι αλλαγές αυτές φαίνεται να αυξάνουν δραματικά την απόδοση της. Ακολουθίες όπως η Sobol εχουν σταθερά καλή απόδοση και διακυμανση στον χώρο ασχέτως του αριθμού διαστάσεων. Τέλος υπολογίζεται το μέγιστο σφάλμα της ολοκλήρωσης Quasi Monte Carlo σε εννέα δοκιμαστικές συναρτήσεις σε ψηλό αριθμό διαστάσεων, και συγκρίνεται το σφάλμα που έχουν οι ως τότε γνωστές μέθοδοι και οι δυό νέες που έχουν προταθεί σε αυτή

τη μελέτη και συμπαιρένεται οτι παρότι οι νέες μέθοδοι τα πηγαίνουν καλύτερα, η απόδοση τους εξακολουθεί να εξαρτάται αρκετά απο τις ιδιατερότητες και χαρακτηριστικά (μονοτονία, ακρότατα κ.ο.κ) της εκάστοτε συνάρτησης.

Στην μελέτη των Bratley et. al [21] μελετάνε τις υλοποίησεις ακολουθιών χαμηλής ασυμφωνίας εστιάζοντας στις Halton, Sobol', Faure, και Niederreiter.Τα αποτελέσματα των δοκιμών σε πολυδιάστατα ολοκληρώματα δείχνουν την ακρίβεια των εφαρμοζόμενων ακολουθιών. Η σύγκριση με τις ακολουθίες του Sobol αναδεικνύει τα πλεονεκτήματα των ακολουθιών του Niederreiter για μεγαλύτερες διαστάσεις.

Ο Harald [22] μελετάει τεχνικές Monte Carlo οι οποίες χρησιμοποιούνται συνήθως για την εκτέλεση αναλύσεων αβεβαιότητας και ευαισθησίας. Υποστηρίζει οτι ένα βασικό στοιχείο των μεθόδων MC είναι η δειγματοληψία των παραμέτρων εισόδου για την προσομοίωση, όπου ο στόχος είναι να εξερευνηθεί ολόκληρος ο χώρος εισόδου με ένα λογικό μέγεθος δείγματος (N) και οτι το μέγεθος του δείγματος καθορίζει το υπολογιστικό κόστος της ανάλυσης αφού το N είναι ίσο με τον απαιτούμενο αριθμό εκτελέσεων προσομοίωσης. Συγγρίνει την δειγματοληψία που βασίζεται σε ακολουθίες Sobol με άλλες τυπικές διαδικασίες δειγματοληψίας σε σχέση με τυπικές εφαρμογές προσομοίωσης κτιρίου. Τέλος, καταλήγει ότι για τις περισσότερες από τις πτυχές που αναλύθηκαν η δειγματοληψία με βάση τις ακολουθίες Sobol αποδίδει καλύτερα από τις άλλες τεχνικές δειγματοληψίας που διερεύνησαν.

Η μελέτη σχετικά με τις τεχνικές δειγματοληψίας Hammersley και Halton από τους Wong, Luk και Heng [23] διερευνά την αποτελεσματικότητα αυτών των ακολουθιών χαμηλής απόκλισης στα γραφικά υπολογιστών. Μέσω των πειραμάτων ανίχνευσης ακτίνων, παρατηρήθηκε ότι ενώ τα σημεία Halton προσφέρουν σταδιακά πλεονεκτήματα δειγματοληψίας, τα σημεία Hammersley παρέχουν ανώτερα μοτίβα. Ο μετασχηματισμός αυτών των σημείων σε μια σφαίρα έδειξε ότι ο ισημερινός και ο πόλος έγιναν δυσδιάκριτοι, υπογραμμίζοντας την ομοιόμορφη κατανομή που επιτεύχθηκε. Συνολικά, η έρευνα δίνει έμφαση στην πρακτική χρησιμότητα αυτών των συνόλων σημείων στη δημιουργία οπτικά ευχάριστων και ομοιόμορφα κατανεμημένων μοτίβων δειγματοληψίας για διάφορες εφαρμογές στα γραφικά υπολογιστών.

Οι Ahmed et al. [24] παρουσίασαν μια νέα τεχνική που παράγει δισδιάστατα σύνολα σημείων μπλε θορύβου χαμηλής διαφοράς (LD) για δειγματοληψία. Αξιοποιώντας μονοδιάστατες δυαδικές ακολουθίες van der Corput, δημιουργούν δισδιά-

στατα σύνολα σημείων LD και τα αναδιατάσσουν έτσι ώστε να αντιστοιχούν σε έναν επιθυμητό στόχο, διατηρώντας το φασματικό προφίλ τους με χαμηλή απόκλιση. Οι πληροφορίες αναδιάταξης αποθηκεύονται σε έναν συμπαγή πίνακα αναζήτησης, ο οποίος μπορεί να χρησιμοποιηθεί για τη δημιουργία αυθαίρετα μεγάλων συνόλων σημείων.

Η μελέτη των Deutsch και Deutsch [25] εισάγει μια νέα επέκταση της λατινικής δειγματοληψίας υπερκύβου, το LHSMDU, με στόχο την ενίσχυση της πολυδιάστατης ομοιομορφίας των παραμέτρων εισόδου για σύνθετες υλοποιήσεις μοντέλων. Εξετάζοντας τις συσχετίσεις στον πίνακα δειγματοληψίας χρησιμοποιώντας την αποσύνθεση Cholesky, το LHSMDU βελτιώνει σημαντικά την αποτελεσματικότητα υλοποίησης για μεγάλα πολυμεταβλητά προβλήματα. Τα αποτελέσματα δείχνουν ότι το LHSMDU υπερέχει των παραδοσιακών μεθόδων όπως η προσομοίωση Monte Carlo και η δειγματοληψία με λατινικούς υπερκύβους, ειδικά σε σενάρια με συσχετισμένες μεταβλητές. Η μελέτη παρουσιάζει ότι η αυξημένη πολυδιάστατη ομοιομορφία οδηγεί σε ακριβέστερη αναπαραγωγή των κατανομών πιθανοτήτων, με τον αλγόριθμο να αποδίδει αποτελεσματικά ανεξάρτητα από τον αριθμό των πραγματοποιήσεων και τη διάσταση του πίνακα δειγματοληψίας. Αυτά τα ευρήματα υπογραμμίζουν τις δυνατότητες του LHSMDU στη βελτιστοποίηση της αποτελεσματικότητας και της ακρίβειας των σύνθετων προσομοιώσεων μοντέλων.

Τέλος, η μελέτη των Hou et. al [26] εξέτασε την διάδοση αβεβαιότητας (propagation of uncertainty) όταν χρησιμοποιείται η μέθοδος Quasi Monte Carlo. Μελέτησαν την απόδοση δειγματοληψίας τεσσάρων qmc μεθόδων δειγματοληψίας (Optimized Latin hypercube, ακολουθία Sobol, ακολουθία Niederreiter–Xing και ακολουθία lattice), καθώς και τους παράγοντες που επηρεάζουν την επίδοση των Quasi Monte Carlo μεθόδων. Μέσα από δύο πειράματα που πραγματοποίησαν δείχνουν ότι το Quasi Monte Carlo φαίνεται να έχει καλύτερες επιδόσεις από την απλή Monte Carlo. Επίσης συμπέραναν ότι οι μέθοδοι Quasi Monte Carlo μπουρούν να καταφέρουν καλύτερες επιδόσεις από τη Monte Carlo όταν η συνάρτηση στόχος (target function) είναι αρκετά ομαλή και εξαρτάται σε περιορισμένο αριθμό παραμέτρων.

Κεφάλαιο 4

Υλοποίηση

Σκοπός αυτού του κεφαλαίου αποτελεί η εξήγηση όλων τον εργαλείων και τεχνικών που χρησιμοποιήθηκαν για την κατασκευή του πειράματος και το πως αυτά συνδέονται λογικά μεταξύ τους για να οδηγηθούμε στη μελέτη των αποτελεσμάτων και στα τελικά συμπεράσματα.

4.1 Εργαλεία

Στο γρήγορα εξελισσόμενο κόσμο της μηχανικής μάθησης, όπου οι αλλαγές είναι ραγδαίες, οι σύγχρονοι προγραμματιστές εξοπλίζονται με ποικίλα εργαλεία για να πλοηγηθούν στον κλάδο τους. Από ισχυρές βιβλιοθήκες και πλαίσια εργασίας έως περιβάλλοντα ανάπτυξης και οπτικοποίησης, οι προγραμματιστές στον κλάδο της μηχανικής μάθησης διαθέτουν ό,τι χρειάζονται για να αναπτύξουν λογισμικά με τα οποία θα μπορέσουν να έχουν αντίκτυπο στον κόσμο. Στο κεφάλαιο αυτό θα μιλήσουμε για τα ουσιώδη εργαλεία που αποτελούν τη βάση των σύγχρονων προσπαθειών στη μηχανική μάθηση. Από το εκτεταμένο οικοσύστημα της Python και τις βιβλιοθήκες όπως το Scikit-Learn και το NumPy, μέχρι τα αλληλεπιδραστικά περιβάλλοντα ανάπτυξης όπως το Jupyter Notebook και τα ισχυρά εργαλεία οπτικοποίησης όπως το Matplotlib και το Seaborn, θα περιγράψουμε τα εργαλεία και τις τεχνολογίες που κινούν την καινοτομία και την ανακάλυψη στον τομέα της μηχανικής μάθησης.

4.1.1 C

Η C είναι μια ευρέως διαδεδομένη γλώσσα προγραμματισμού γνωστή κυρίως για την αποδοτικότητα καθώς και την ευελιξία της. Παρότι είχε αναπτυχθεί η αρ-

χική της έκδοση περίπου το 1970 παραμένει ακόμη ένα θεμελιώδες εργαλείο στην ανάπτυξη λογισμικού. Βασικό της χαρακτηριστικό είναι ότι επιτρέπει στον προγραμματιστή χειρισμούς χαμηλού επιπέδου τόσο στο υλικό όσο και στην μνήμη του ηλεκτρονικού υπολογιστή, γεγονός που την καθιστά χρήσιμη για εφαρμογές που απαιτούν γρήγορη απόδοση ή μικρό όγκο δεδομένων. Επιπλέον, ο κώδικας της μπορεί να μεταγλωττιστεί σε ένα ευρύ φάσμα από πλατφόρμες κι έτσι θεωρείται αρκετά "φορητή". Τέλος, οι πολλές βιβλιοθήκες που μπορούν να προστεθούν στη βασική βάση κώδικα της, της επιτρέπουν να κρατάει το μέγεθος ενός αλγορίθμου σχετικά μικρό, επαναχρησιμοποιώντας τα διάφορα έτοιμα τμήματα κώδικα που αυτές προσφέρουν.

4.1.2 Python

Η Python είναι μια γενικής χρήσης γλώσσα προγραμματισμού φιλική προς τον αρχάριο προγραμματιστή, η οποία λόγω της ευελιξίας της χρησιμοποιείται σε πολλούς παράταιρους τομείς όπως web development, software development, data science, αυτοματισμούς κ.ο.κ. Είναι επίσης σημαντικό να επισημάνουμε ότι η Python είναι από τις πιο διαδεδομένες γλώσσες προγραμματισμού. Η πληθώρα βιβλιοθηκών που μπορεί να υποστηρίξει την καθιστούν απαραίτητο εργαλείο οποιουδήποτε μοντέρνου προγραμματιστή, ανεξαρτήτως επιπέδου. Συγκεκριμένα θα μιλήσουμε για τις βιβλιοθήκες NumPy, Pandas, Scikit-Learn και Matplotlib/Seaborn οι οποίες χρησιμοποιήθηκαν στην εργασία αυτή.

4.1.3 Προγραμματιστικό Περιβάλλον

Ένα πρόγραμμα εγγραφής κώδικα ήταν απαραίτητο για την υλοποίηση της εργασίας. Παρόλο που χρησιμοποιήθηκαν και συνηθισμένα IDE όπως το PyCharm, ήταν αναγκαίο να γίνει χρήση ενός εργαλείου που θα μας προσέφερε την άμεση αλληλεπίδραση με τον κώδικα.

Το Jupyter Notebook είναι ένας τύπος αρχείου που υποστηρίζει πάνω από 40 γλώσσες προγραμματισμού και βοηθά τον χρήστη να φτιάξει ένα αρχείο κώδικα με λιγότερους περιορισμούς στη μορφοποίηση του σε σχέση με αυτούς που έχει ένα IDE (Ολοκληρωμένο Περιβάλλον Ανάπτυξης). Συγκεκριμένα καθε notebook δίνει στον προγραμματιστή τη δυνατότητα να έχει πλεγμένα μεταξύ τους κελιά καθαρού

κώδικα και μορφοποιημένου κειμένου, δημιουργώντας έτσι μια πιο ευανάγνωστη και κατανοητή περιήγηση στον κώδικα. Τα jupyter notebooks είναι προσβάσιμα μέσω εφαρμογών που συνήθως τρέχουν στον browser, όπως το jupyterLab.

4.1.4 NumPy

Η NumPy (Numerical Python) είναι μια βιβλιοθηκη της Python η οποία χρησιμοποιείται κυρίως όταν θέλουμε να δουλέψουμε συγκεκριμένα με πίνακες. Κάποια από τα κομμάτια του κώδικα της είναι γραμμένα σε C ή C++, πράγμα το οποίο την καθιστά πολύ ισχυρή και γρήγορη σε διαδικασίες γραμμικής άλγεβρας, όπως για παράδειγμα ο πολλαπλασιασμός πινάκων που έχουν παραδοσιακά μεγάλο χρονικό κόστος .Για αυτό το λόγο χρησιμοποιείται και στη ΜΜ, όπου η ταχύτητα και η αποτελεσματική χρήση των πόρων ενός συστήματος είναι κρίσιμοι παράγοντες.

4.1.5 Pandas

Η Pandas είναι μια βιβλιοθήκη της Python η οποία παρέχει δυνατότητες για την καταχώρηση και επεξεργασία δεδομένων. Αυτό το επιτυγχάνει με τις δομές δεδομένων Series(1-D), Dataframe(2-D) και Panel(3-D). Τα εργαλεία που περιλαμβάνει δίνουν την δυνατότητα ενός ευρέως φάσματος ενεργειών που μπορούν να τελεστούν στις δομές αυτές, όπως προσπέλαση και μετασχηματισμό δεδομένων, ταξινόμηση, συνδυασμός κ.α. Η διαφορά της με την NumPy είναι ότι σε μια δεδομένη δομή της Pandas μπορούν να υπάρχουν ετερογενή δεδομένα.

4.1.6 Scikit-Learn

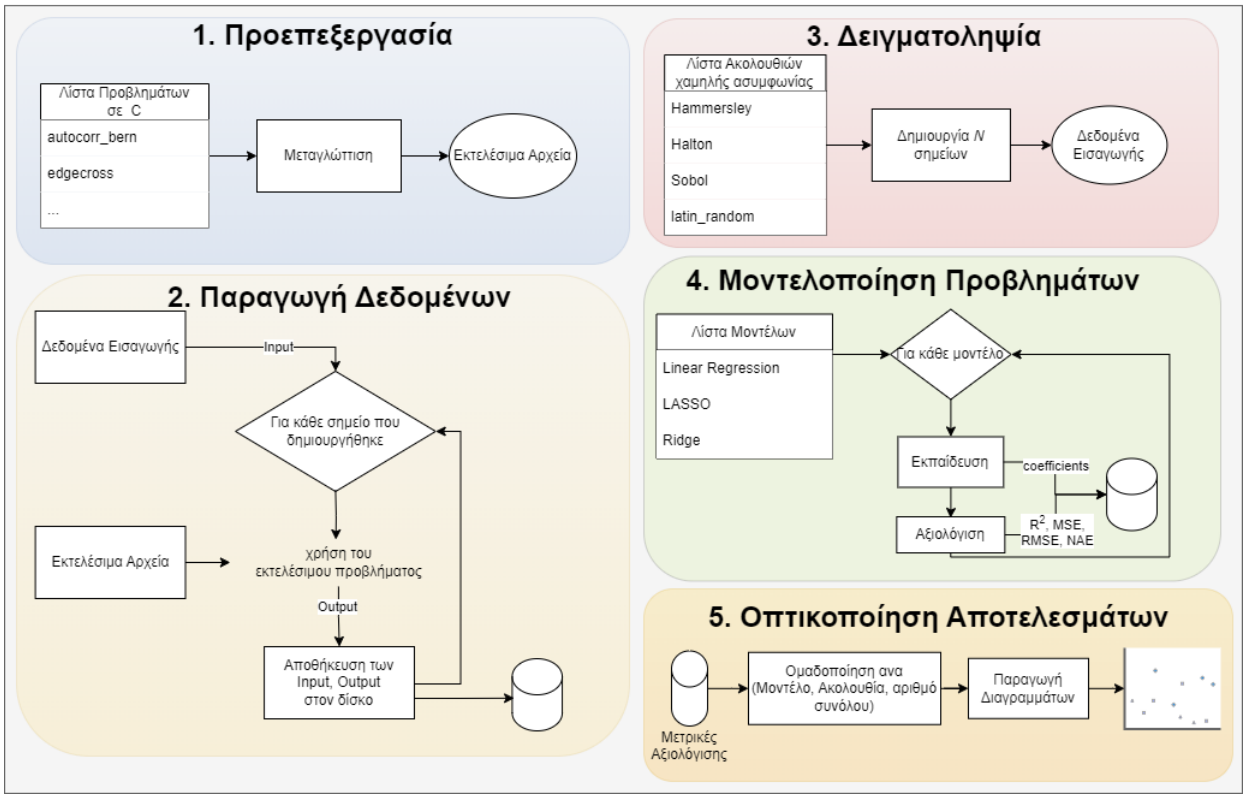
Η Scikit-Learn είναι μια βιβλιοθήκη μηχανικής μάθησης ανοιχτού κώδικα που υποστηρίζει επιτηρούμενη και μη-επιτηρούμενη μάθηση. Είναι σχεδιασμένη για να αλληλεπιδρά καλά με τις βιβλιοθήκες Pandas, NumPy και Matplotlib. Επειδή ακριβώς το Scikit-Learn υποστηρίζει τόσες πολλές δυνατότητες, όπως προεπεξεργασία δεδομένων, αλγόριθμους και μοντέλα ΜΜ, τρόπους σαφούς αναπαράστασης δεδομένων και σύγκρισης αποτελεσμάτων, αποτελεί μια από τις δημοφιλέστερες βιβλιοθήκες ανοιχτού κώδικα στο GitHub.

4.1.7 Matplotlib και Seaborn

Η Matplotlib είναι η μια βιβλιοθήκη της Python σχεδιασμένη για την απεικόνιση μαθηματικών δεδομένων της βιβλιοθήκης NumPy. Παρέχει ένα αντικειμενοστραφές API για να ενσωματώνει γραφήματα σε εφαρμογές. Με αντίστοιχο τρόπο δουλεύει και η βιβλιοθήκη Seaborn, που λειτουργεί ως μια επέκταση της matplotlib.

4.2 Διάρθρωση Πειράματος

Η γενική διάρθρωση του πειράματος είναι σχετικά απλή εκ πρώτης όψεως, ωστόσο διάφορες ιδιαιτερότητες εμφανίστηκαν κατά τη διάρκεια της ανάπτυξης, λόγω των διαφόρων τεχνολογιών και τεχνικών που χρησιμοποιήθηκαν για να επιτευχθεί το τελικό αποτέλεσμα. Παρακάτω παρατίθεται το συνολικό διάγραμμα ροής του πειράματος (Σχήμα 4.1), του οποίου τις υποενότητες θα εξηγήσουμε αναλυτικά στις επόμενες σελίδες.



Σχήμα 4.1: Γενικό διάγραμμα πειραματικής μεθόδου

Η κεντρική ιδέα του αλγορίθμου αποτελείται από τα εξής βήματα:

1. Προεπεξεργασία: Στην προεπεξεργασία τα αρχεία των προβλημάτων μετατράπηκαν στην κατάλληλη γλώσσα προγραμματισμού και τακτοποιήθηκαν σε ένα

σύστημα φακέλων με συγκεκριμένες προδιαγραφές. Έτσι ήταν εφικτό να αυτοματοποιηθούν πλήρως οι μετέπειτα διαδικασίες της Δειγματοληψίας, της Παραγωγής εξόδων και της Μοντελοποίησης προβλημάτων. Έπειτα, μεταγλωττίστηκε κάθε αρχείο πηγαίου κώδικα.

2. Δειγματοληψία: Στη δειγματοληψία αρχικά παρουσιάζεται ο τρόπος αλγοριθμικής υλοποίησης της κάθε ακολουθίας χαμηλής ασυμφωνίας. Έπειτα εξετάστηκαν οι προδιαγραφές του καθενός από τα Χ προβλήματα ξεχωριστά, κρατώντας στη μνήμη του ηλεκτρονικού υπολογιστή τα δεδομένα εισόδου ανά ακολουθία που είναι εφικτή με βάση τους περιορισμούς (Κάποιες από αυτές δεν δύναται να υλοποιηθούν για προβλήματα με διαστάσεις έξω από το δυνατό εύρος της καθεμιάς). Είναι σημαντικό να τονιστεί ότι τα βήματα 2-4 εκτελούνται αρκετές φορές, για διαφορετικό πλήθος σημείων ανά πρόβλημα.

3. Παραγωγή Εξόδων: Στην παραγωγή εξόδων δώσαμε ως είσοδο στο εκτελέσιμο του κάθε προβλήματος το δεδομένα εισόδου που παρήχθησαν για το πρόβλημα αυτό ανά ακολουθία, σώζοντας ένα αρχείο με την κάθε έξοδο δίπλα στην είσοδο με την οποία σχετίζεται, για κάθε ακολουθία δειγματοληψίας όπως προαναφέραμε.

4. Μοντελοποίηση Προβλημάτων: Αφού υπάρχουν πλέον όλα τα δεδομένα αποθηκευμένα τοπικά και διαχωρισμένα ανά πρόβλημα και ακολουθία δειγματοληψίας, εφαρμόστηκαν για κάθε ένα από αυτά οι τρεις γραμμικές μεθόδους παλινδρόμησης που αναφέρθηκαν στη βιβλιογραφική ανασκόπηση (Linear Regression, Ridge Regression και LASSO Regression) και συσσωρεύτηκαν ξεχωριστά αρχεία μετρικών όπως R^2 , MSE , RMSE και MAE, τα οποία χρησιμοποιήθηκαν αμέσως μετά. Επίσης, συλλέχθηκαν και οι προβλέψεις των μοντέλων για τις παραμέτρους της αντικειμενικής συνάρτησης του καθενός από αυτά κι έτσι διαπιστώθηκε η ικανότητα των μοντέλων να προβλέψουν τις συναρτήσεις αυτές σε προβλήματα με μικρό αριθμό διαστάσεων στα δεδομένα.

5. Οπτικοποίηση Αποτελεσμάτων: Σε αυτό το στάδιο της εργασίας αξιοποιήσαμε όλο τον όγκο μετρικών που έχουμε ήδη συσσωρεύσει για να αναπαραστήσουμε διαγραμματικά τα ευρήματά μας. Οι τρεις τύποι διαγραμμάτων που χρησιμοποιήθηκαν είναι θηκόγραμμα (box plot), ραβδόγραμμα (bar plot) και διάγραμμα συσχέτισης (Correlation plots).

4.2.1 Προεπεξεργασία

Τα προβλήματα προήλθαν απο την βάση MINLP Instance Database σε μορφή αρχείου .gms. Αφού μετατράπηκαν σε πηγαίο κώδικα c (αρχείο .c), τακτοποιήθηκαν ανα πρόβλημα σε ένα φάκελο που ονομάστηκε με το όνομα του προβλήματος. Ο φάκελος αυτός περιέχει τον πηγαίο κώδικα σε c, καθώς κι ένα απλό αρχείο κειμένου (.txt) με όνομα "problemdata" το οποίο περιέχει τις προδιαγραφές των δεδομένων του προβλήματος, με την εξής μορφοποίηση:

- 1η σειρά: Αριθμός διαστάσεων του κάθε σημείου
- 2η σειρά: Κατώτατο όριο τιμής για κάθε διάσταση
- 3η σειρά: Ανώτατο όριο τιμής για κάθε διάσταση
- 4η σειρά: Τύπος τιμης διάστασης (1 = ακέραιος, 0 = πραγματικός)

Παρακάτω φαίνεται ένα παράδειγμα του αρχείου problemdata (Σχήμα 4.2), όπου το κάθε σημείο έχει 25 διαστάσεις, η πρώτη έχει κατώτατη τιμή 0 και ανώτατη τιμή 1 και είναι ακέραιος αριθμός κ.ο.κ.

```
25
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Σχήμα 4.2: Παράδειγμα περιεχομένων αρχείου problemdata.txt

Ο πηγαίος κώδικας γράφτηκε έτσι ώστε να παίρνει ως είσοδο ένα απλό αρχείο κειμένου (.txt) με όνομα "input" το οποίο περιέχει ένα σημείο με βάση τις προδιαγραφές του προβλήματος, να περνάει την κάθε διάσταση του σημείου ως έναν όρο της αντικειμενικής του συνάρτησης (η συνάρτηση αυτή έχει τόσους ανεξάρτητους όρους όσες είναι και οι διαστάσεις) και να γυρνάει την εξαρτημένη μεταβλητή ως έξοδο σε ένα αρχείο .txt με όνομα "output". Παρότι η πειραματική διαδικασία υλοποιήθηκε ως επί το πλείστον στη γλώσσα Python, το συγκεκριμένο τμήμα υλοποιήθηκε σε C αφενός για να αυξηθεί η συνολική ταχύτητα εκτέλεσης της διαδικασίας και αφετέρου για να μην είναι αναλυτικά διαθέσιμη η αντικειμενική συνάρτηση του κάθε προβλήματος. Όπως θα δούμε παρακάτω η Python δεν αλληλεπιδρά με τα

περιεχόμενα του εκτελέσιμου αρχείου, αλλά μόνο με τα περιεχόμενα των αρχείων problemdata, input και output.

4.2.2 Δειγματοληψία

Αρχικά βλέπουμε τον τρόπο με τον οποίο υλοποιήθηκαν αλγοριθμικά οι ακολουθίες χαμηλής ασυμφωνίας που μελετήσαμε στη βιβλιογραφική ανασκόπηση. Να σημειωθεί ότι στην περίπτωση της ακολουθίας Sobol προϋπήρχε βιβλιοθήκη στην Python (sobol-seq) που να υποστηρίζει την παραγωγή σημείου σύμφωνα με τη θεωρητική μέθοδο η οποία χρησιμοποιήθηκε (υλοποιήθηκε μόνο η προσθήκη των σημείων αυτών σε μια ακολουθία). Στην περίπτωση των υπολοίπων ακολουθιών δημιουργήθηκε μια συνάρτηση παραγωγής ενός σημείου, η οποία καλείται από μια συνάρτηση που επαναληπτικά παράγει τη συνολική ακολουθία.

Result: seq
if *dimension* $\geq$ *1* & *dimension* $\leq$ *40* **then**
    Xdata = sobol.seq.i4.sobol.generate(1, num);
    ratio = uBound - lBound;
    seq = [i * ratio + lBound for i in Xdata];
    seq=transformToIntegerOrFloat(seq);
else
    seq = None;
end

Αλγόριθμος 1: Sobol Sequence

Στον Αλγόριθμο 1 βλέπουμε τον τρόπο με τον οποίο υλοποιήθηκε η ακολουθία Sobol. Αρχικά τόσο σε αυτή όσο και στις υπόλοιπες ακολουθίες, υπάρχει ένας σχετικός έλεγχος για το αποδεκτό πλήθος διαστάσεων που μπορεί να είχε το εκάστοτε πρόβλημα (στην περίπτωση αυτή $dimension \in [1, 40]$). Εάν δεν βρίσκεται στο αποδεκτό πλήθος η συνάρτηση απλα γυρνάει None και η διαδικασία της δειγματοληψίας συνεχίζει κανονικά, χωρίς να συμπεριληφθεί αυτή η ακολουθία για αυτό το πρόβλημα. Έπειτα καθορίζεται η μεταβλητή ratio η οποία σε όλες τις υλοποιήσεις υποδεικνύει το διάστημα που οι αριθμοί της εκάστοτε ακολουθίας πρέπει να κατανεμηθούν με χαμηλή ασυμφωνία. Με βάση την μεταβλητή αυτή εισάγονται τα στοιχεία με τη σειρά στη μεταβλητή που αποθηκεύεται η ακολουθία και τέλος καλείται η συνάρτηση transformToIntegerOrFloat η οποία μετατρέπει κάθε στοιχείο της ακολουθίας σε ακέραιο ή πραγματικό , ανάλογα με τις προδιαγραφές του αρ-

χείου problemdata. Επειδή η υλοποίηση είναι αρκετά παρόμοια και στις υπόλοιπες ακολουθίες εστιάζουμε περισσότερο στις διαφορές στον αλγόριθμο που πηγάζουν απο τον μαθηματικό ορισμό τους.

Μια σημαντική διαφορά που πρέπει να αναφερθεί είναι οτι οι επόμενες ακολουθίες χρησιμοποιούν πρώτους αριθμούς για την υλοποίηση τους. Για να επιτευχθεί αυτό εμείς έχουμε ορίσει την λίστα prime η οποία περιέχει ένα μεγάλο πλήθος πρώτων αριθμών με τη σωστή σειρά. Γνωρίζοντας οτι οι πρώτοι 10 πρώτοι αριθμοί είναι οι (2, 3, 5, 7, 11, 13, 17, 19, 23, 29) λέμε οτι ο prime(3) = 5 κ.ο.κ.

```
Result: point
i = index of sequence
m = dimension number
t = m-size array of 1s
r = m-size array of 0s
prime = m-size array m prime numbers
prime_inv = 1/prime
while 0 ≤ sum(t) do
    for i=0, i≤m, i+=1 do
        d = t[j] MOD prime(j)
        r[j] = r[j] + d * prime_inv[j]
        prime_inv[j] = prime_inv[j]/prime(j)
        t[j] = t[j] // prime(j)
    end
end
```

Αλγόριθμος 2: Παραγωγή σημείου Halton

Στον Αλγόριθμο 2 βλέπουμε την παραγωγή ενός σημείου m διαστάσεων της ακολουθίας Halton. Εδώ γίνεται εκτενής χρήση της βιβλιοθήκης NumPy για να δημιουργηθούν πίνακες γεμάτοι 0 ή 1 έτσι ώστε να τελεστούν πράξεις πινάκων με αυτούς αργότερα. Αρχικά δημιουργούμε τον ανάστροφο πίνακα των πρώτων m πρώτων αριθμών. Έπειτα οσο το άθροισμα των στοιχείων του t δεν είναι αρνητικό τελούμε m επαναλήψεις όπου κρατάμε το υπόλοιπο της διαίρεσης του t[j] με τον j-στο πρώτο αριθμό και το χρησιμοποιούμε για να δημιουργήσουμε τον όρο r[j]. Αναπροσαρμόζουμε τα prime_inv και το στοιχείο t[j] και συνεχίζουμε. Τα στοιχεία του πίνακα r καταλήγουν να το ένα σημείο m διαστάσεων της ακολουθίας Halton.

Στον Αλγόριθμο 3 η μέθοδος που χρησιμοποιούμε είναι ακριβώς η ίδια με τον Αλγόριθμο 1. Το διάστημα των αποδεκτών διαστάσεων για την ακολουθία Halton είναι το $dimension \in [1, 1600]$.

Στον Αλγόριθμο 4 βλέπουμε πάλι μια παρόμοια υλοποίηση με αυτή του Αλγο-

```
Result: seq
if dimension ≥ 1  &  dimension ≤ 1600  then
    ratio = uBound - lBound
    for i=0, i≤points, i+=1 do
        Halton_point = halton(i, dim)
        seq.append(Halton_point * ratio + lBound)
    end
    seq=transformToIntegerOrFloat(seq)
else
    seq = None;
end
```

Αλγόριθμος 3: Υλοποίηση Ακολουθίας Halton

```
Result: r
j = index of sequence
t = (m-1)-size array of 1s
t=i*t
prime_inv = (m-1)-size array of 0s
for j=0, j ≤ m-1, j++ do
    prime_inv[j] = 1.0/ float(prime(j))
end
r = np.zeros(m)
r[0] = float(iMOD(n + 1)) / float(n)
while 0 ≤ np.sum(t) do
    for j=0, j ≤ m-1, j++ do
        d = (t[j]MODprime(j))
        r[j + 1] = r[j + 1] + float(d) * prime_inv[j]
        prime_inv[j] = prime_inv[j] / prime(j)
        t[j] = (t[j] // prime(j))
    end
end
```

Αλγόριθμος 4: Παραγωγή σημείου Hammersley

ρίθμου 2. Η διαφορά είναι οτι για m-1 περάσματα έχουμε σημείο m διαστάσεων, κάνοντας την πρώτη διάσταση του κάθε σημείου να είναι ίση με το πηλίκο δύο όρων. Του υπολοίπου της διαίρεσης του μετρητή i με τον συνολικό αριθμό σημείων n αυξημένο κατα ενα και του ιδιου του αριθμού n.

Στον Αλγόριθμο 5 βλέπουμε ότι η μόνη διαφορά είναι στον τύπο του ratio, ο οποίος αναπροσαρμόστηκε για να διαιρεί τον χώρο όπως η ακολουθία Hammersley.

Στην υλοποίηση της ακολουθίας Latin Random θα παραθέσουμε ενα περιορισμένο τμήμα του αλγορίθμου χάριν συντομίας. Τηρώντας το μοτίβο με το οποίο υλοποιούμε τις ακολουθίες αυτές έχουμε τους Αλγορίθμους 6 & 7. Ο Αλγόριθμος 7 καλεί τον Αλγόριθμο 6 ο οποίος γυρνάει το σημείο dim διαστάσεων, το οποίο

```
Result: np.array(seq)
if dimension ≥ 1  &  dimension ≤ 100  then
    seq = []
    for i=0, i ≤ points, i++ do
        hh=np.array(hammersley(i,dim,points))
        val = (hh*(uBound - lBound)) + lBound
        seq.append(val)
    end
    seq = update_with_is_integer(seq, is_integer)
else
    seq = None
end
```

Αλγόριθμος 5: Υλοποίηση Ακολουθίας Hammersley

```
Result: x
x, seed = r8mat_uniform_01 ( dim_num, point_num, seed )
for i = 0, i≤dim_num, i++ do
    perm, seed = perm_uniform ( point_num, seed )
end
for j = 0, j≤dim_num, j++ do
    x[i,j] = ( perm[j] + x[i,j] ) / point_num
end
```

Αλγόριθμος 6: Υλοποίηση Πίνακα Σημείων Latin Random

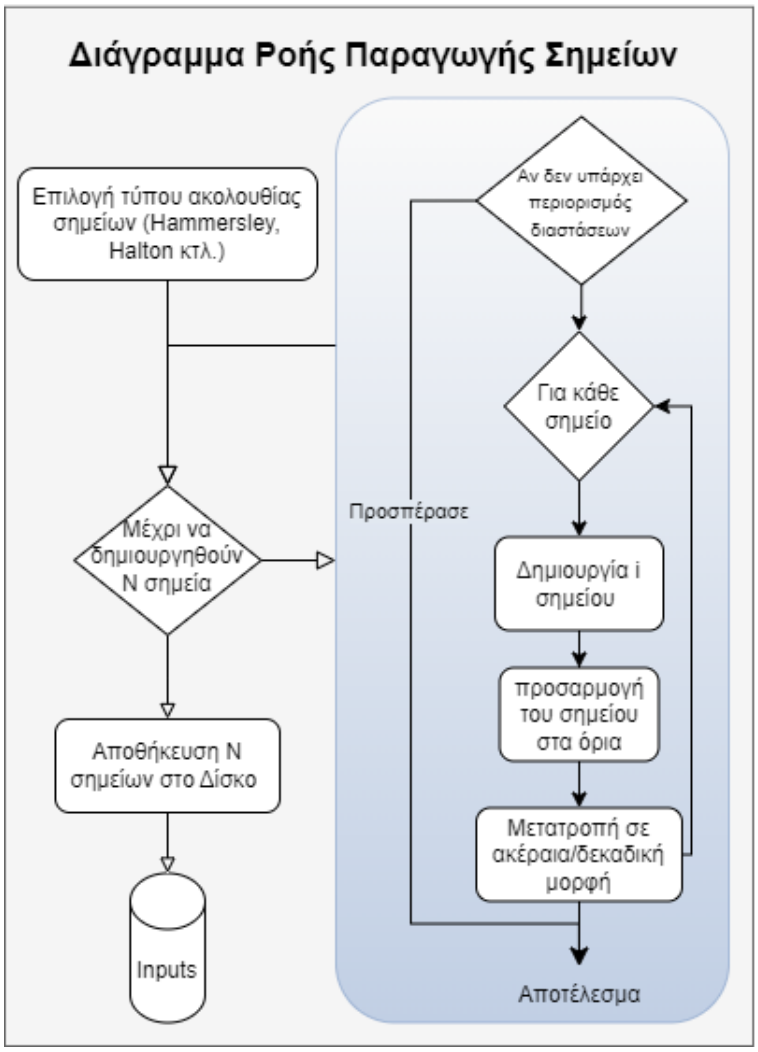
έπειτα προσαρμόζεται στον χώρο σύμφωνα με το ratio.

```
Result: np.array(seq)
ratio = (uBound-lBound)/1.0
xdata,_ = latin_random(points, dim,seed)
seq = np.array([num*ratio+lBound for num in xdata])
seq = update_with_is_integer(seq, is_integer)
```

Αλγόριθμος 7: Υλοποίηση Ακολουθίας Latin Random

Όπως παρατηρείται από τους περιορισμούς των ακολουθιών κάποιες δεν είναι δυνατόν να τρέξουν για ορισμένα από τα προβλήματα, λόγω περιορισμών στον αριθμό διαστάσεων που μπορούν να διαχειριστούν. Στην περίπτωση αυτή, συλλέγονται τα δεδομένα που δύναται να παραχθούν. Σε αυτό το σημείο της διαδικασίας εισάγεται και ο αριθμός των σημείων που επιθυμούμε να παράξουμε. Ενδεικτικά ο μεγαλύτερος αριθμός είναι 2,500 σημεία, πράγμα που σημαίνει ότι για κάθε πρόβλημα θα κρατηθούν σε λίστα στη μνήμη του ηλεκτρονικού υπολογιστή τα 2,500 σημεία που παρήχθησαν από κάθε ακολουθία, σύμφωνα με τις προδιαγραφές του αρχείου problemdata.

Στο Σχήμα 4.3 βλέπουμε το διάγραμμα ροής της δειγματοληψίας.

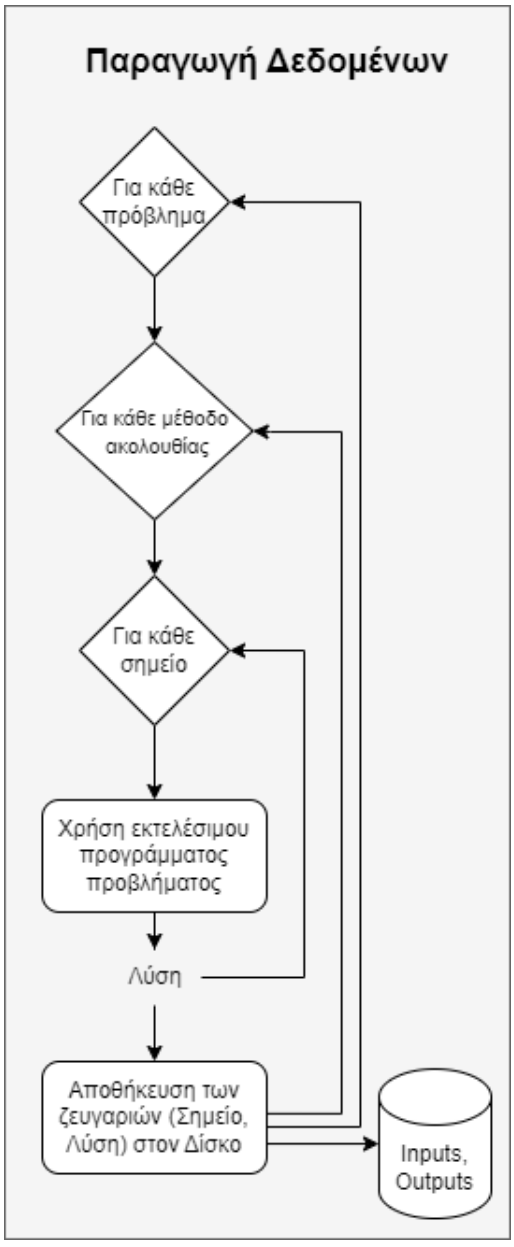


Σχήμα 4.3: Διαδικασία παραγωγής σημείων (Δειγματοληψία)

4.2.3 Παραγωγή Εξόδων

Έχοντας κρατήσει τη λίστα με τις εισόδους για το εκάστοτε πρόβλημα και μέθοδο δειγματοληψίας στη μνήμη, προτού περάσουμε στην επόμενη μέθοδο ή στο επόμενο πρόβλημα, δίνουμε μια μια τις γραμμές της λίστας ως είσοδο στο εκτελέσιμο αρχείο (a.out). Αυτο με τη σειρά του αποθηκεύεται στο τέλος κάθε σειράς, ώστε να σχετίζεται με τη σειρά που μπήκε ως είσοδος και όταν η λίστα εισόδων προσπελαστεί πλήρως τότε μεταφέρουμε όλα τα περιεχόμενά της σε ένα Pandas Dataframe, το οποίο μετατρέπεται άμεσα σε αρχείο .csv με τη σχετική μέθοδο της βιβλιοθήκης Pandas. Το αρχείο αυτό έχει ονομασία "ακολουθία.csv", όπου "ακολουθία" το όνομα της ακολουθίας που χρησιμοποιήθηκε για να παραχθούν τα δεδομένα εισόδου (π.χ. hammersley.csv) (Σχήμα 4.4). Αυτά τα αρχεία .csv θα χρησιμοποιηθούν για την εξάσκηση των μοντέλων μας. Αφού ολοκληρωθεί η μοντελοποίηση σε αυτά

τα δεδομένα, εισάγουμε θόρυβο στις εξόδους με την χρήση της μεθόδου random της Python. Ο τρόπος που επιτυγχάνεται αυτό είναι προσθέτοντας τυχαίους αριθμούς στην κάθε έξοδο. Αυτή η διαδικασία γίνεται για δύο διαστήματα αριθμών το $[-1, 1]$ και το $[-5, 5]$ τα οποία αποκαλούνται χαμηλός και υψηλός θόρυβος αντίστοιχα στα πλαίσια της εργασίας.

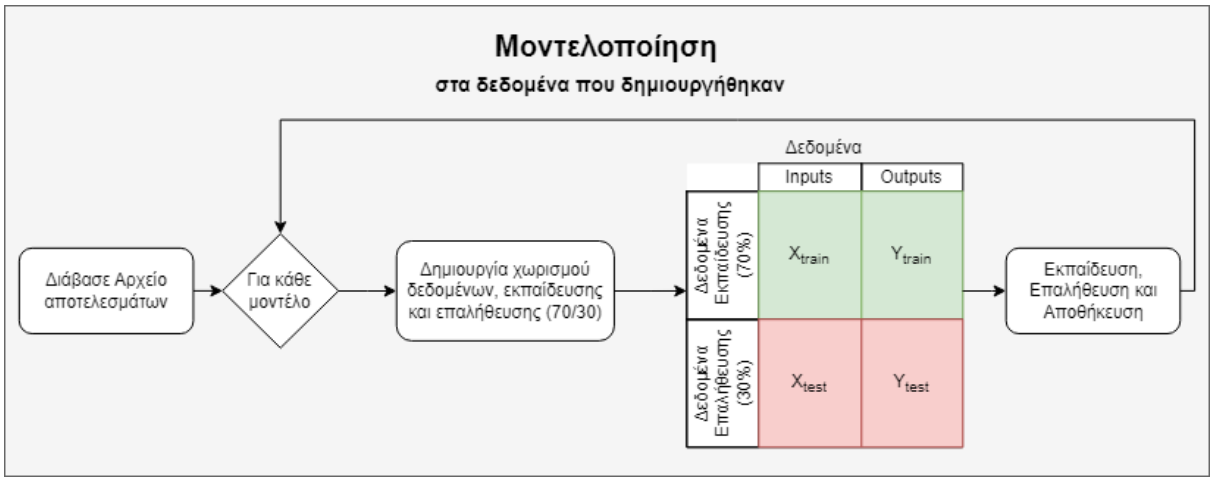


Σχήμα 4.4: Διαδικασία παραγωγής δεδομένων

4.2.4 Μοντελοποίηση Προβλημάτων

Στη μοντελοποίηση των προβλημάτων εφαρμόσαμε κάθε ένα από τα τρία γραμμικά μοντέλα (Linear Regression, Ridge Regression, LASSO Regression) στα δεδο-

μένα κάθε αρχείου .csv κάθε προβλήματος. Πρώτο βήμα είναι ο διαχωρισμός των σειρών κάθε αρχείου σε δεδομένα εξάσκησης και δεδομένα αξιολόγησης. Πραγματοποιήθηκε ένας διαχωρισμός 70-30, όπου το 70 τοις εκατό των δεδομένων θα αποτελούν την εξάσκηση και το υπόλοιπο 30 τοις εκατό την αξιολόγηση (Σχήμα 4.5).

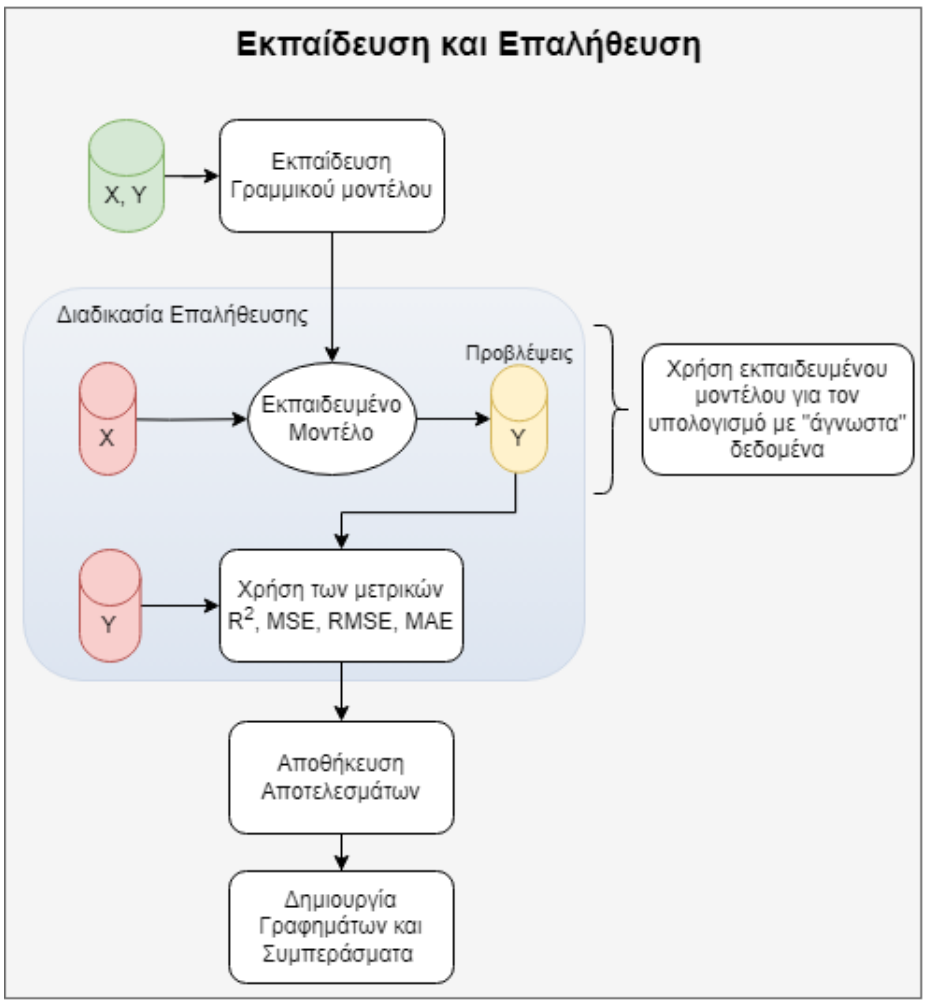


Σχήμα 4.5: Διαδικασία Μοντελοποίησης

Έπειτα εξάγαμε τις μετρικές που αναφέραμε (R^2 , MSE, RMSE και MAE), καθώς και μια πρόβλεψη για τις τιμές των παραμέτρων της αντικειμενικής συνάρτησης του κάθε προβλήματος. Αυτό θα γίνει με τη χρήση της μεθόδου coeff η οποία υποστηρίζεται και από τα 3 μοντέλα μας. Τέλος, για τις περιπτώσεις των Ridge και LASSO προσαρμόσαμε την παράμετρο "α" στις τιμες $\alpha_r = 150$ και $\alpha_l = 30$ αντίστοιχα. Οι τιμές των παραμέτρων α_r και α_l τέθηκαν σε αυτές βάσει των προτάσεων στη βιβλιογραφία [3] καθώς και αποτελέσματα που λάβαμε σε προηγούμενο σημείο της εκπόνησης της διπλωματικής εργασίας (Μοντελοποίηση με τα ιδια μοντέλα σε αυτή τη συλλογή δεδομένων (Auto-Mpg-Dataset). Τέλος, στο Σχήμα 4.6 βλέπουμε μια πιο αναλυτική αναπαράσταση της διαδικασίας που ακολουθήσαμε αλγοριθμικά για να εκπαιδεύθουν τα μοντέλα και να εξάχθουν τα αποτελέσματα τα οποία οπτικοποιήσαμε αργότερα.

4.2.5 Οπτικοποίηση Αποτελεσμάτων

Για την οπτικοποίηση των αποτελεσμάτων χρησιμοποιήθηκαν αρχικά διαγράμματα Box plots. Ένα Box plot ή box and whisker plot αναπαριστά τη διανομή ενός συνόλου δεδομένων. Αν χωρίζαμε το εύρος των δεδομένων σε τεταρτημόρια, τότε



Σχήμα 4.6: Διαδικασία Εκπαίδευσης, Επαλήθευσης και Αποθήκευσης των αποτελεσμάτων

τα περιεχόμενα του κουτιού θα ήταν όλα τα σημεία τα οποία συμπτύσσονται μεταξύ πρώτου (Q1) και τρίτου (Q3) τεταρτημορίου αντίστοιχα. Το εύρος αυτό το ονομάζουμε IQR (Interquartile Range) και η μέση τιμή (median value) του εύρους συμβολίζεται με ένα διάστημα εντός του κουτιού. Οι άκρες του διαστήματος που επεκτείνονται από τις πλευρές του κουτιού ονομάζονται whiskers και σημαδεύουν την κατώτατη και ανώτατη τιμή που μπορούν να λάβουν τα δεδομένα (συνηθίζεται να επεκτείνουν το εύρος στη μιάμιση φορά του εύρους του IQR ή μέχρι τα πιο ακραία σημεία του συνόλου δεδομένων). Τέλος, οποιεσδήποτε τιμές πέφτουν εκτός αυτού του εύρους ονομάζονται ακραίες (outliers) και αναπαριστώνται ατομικά στο διάγραμμα. Στα Box plots είναι αρκετά εύκολο να συγκρίνουμε τις κατανομές μεταξύ διαφορετικών συνόλων, όπως και θα κάνουμε παρακάτω. Επιπλέον, παρέχουν πληροφορίες σχετικά με την κεντρική τάση και τη διασπορά των δεδομένων. Το Scatter plot (διάγραμμα διασποράς) είναι διαγράμματα τα οποία πάνω τους έχουν

απλά σημεία, όπως αυτά ενός καρτεσιανού επιπέδου. Χρησιμεύουν στην υπόδειξη της σχέσης μεταξύ δύο μεγεθών. Στην παρούσα διπλωματική και οι δύο άξονες θα αναπαριστούν το R^2 αλλά για διαφορετικό πλήθος σημείων οπότε θα χαραχθεί επι του διαγράμματος η διχοτόμος του τεταρτημορίου για να είναι ευδιάκριτο το πλήθος των σημείων για το οποίο οι περισσότερες εκτελέσεις τα πάνε καλύτερα. Τα Bar plots (γράφημα ράβδων) είναι ίσως ο διασημότερος τρόπος να αναπαραστήσει κανείς με απλό και ευανάγνωστο τρόπο την σύγκριση δύο η περισσοτέρων κατηγοριών πάνω στην ίδια μετρική.

Κεφάλαιο 5

Υπολογιστική Μελέτη

Στο κεφάλαιο αυτό γίνεται η παρουσίαση και η ανάλυση των αποτελεσμάτων και σημειώνονται όλες οι παρατηρήσεις που μπορούν να σημειωθούν. Το κεφάλαιο αυτό χωρίζεται σε τρία μέρη. Το πρώτο μέρος αναφέρεται στα διαθέσιμα δεδομένα, είτε εισόδου, είτε εξόδου και στο σχεδιασμό του πειράματος. Το δεύτερο μέρος αναφέρεται στις μεθόδους οπτικοποίησης των δεδομένων έτσι ώστε να μπορέσουμε να εξηγήσουμε τα αποτελέσματα καλύτερα. Τέλος, το τρίτο μέρος περιέχει την ανάλυση των δεδομένων.

5.1 Πείραμα και Δεδομένα

5.1.1 Το Πείραμα

Στόχος της εργασίας είναι να μελετήσουμε κατά πόσο η ημι-τυχαίες μέθοδοι δειγματοληψίας βοηθούν στη μείωση της αβεβαιότητας στα δεδομένα έτσι ώστε τα μοντέλα παλινδρόμησης που χρησιμοποιούμε να αποδώσουν καλύτερα. Συγκεκριμένα, ποιο γραμμικό μοντέλο παλινδρόμησης μπορεί να αποδώσει καλύτερα. Εδώ είναι σημαντικό να αναφέρουμε για άλλη μια φορά εν συντομία τον σχεδιασμό του πειράματος και να εστιάσουμε σε συγκεκριμένα στοιχεία του σχεδιασμού του.

Ξεκινώντας απο το ζητούμενο της εργασίας, στόχος είναι να πραγματοποιήσουμε μια σύγκριση μοντέλων. Παρόλο που με μια ματιά αυτό φαίνεται απλό στην πραγματικότητα γίνεται περίπλοκο, καθώς πρέπει να συλλογιστούμε ότι οι υπερ-παράμετροι του πειράματος επηρεάζουν σημαντικά τα αποτελέσματα. Έτσι στην πραγματικότητα πρέπει η ανάλυση να συμπεριλάβει και όλους τους πιθανούς συνδυασμούς των υπερ-παραμέτρων αυτών. Στο πλαίσιο της εργασίας έγινε η επιλογή

δύο βασικών υπερ-παραμέτρων, για τις οποίες γνωρίζουμε ότι έχουν σημαντική επίδραση στα αποτελέσματα. Αυτές είναι ο αριθμός των σημείων που παράγονται για κάθε εκπαίδευση και επαλήθευση των μοντέλων, και ο τύπος ακολουθίας χαμηλής ασυμφωνίας που χρησιμοποιείται για την παραγωγή των δεδομένων αυτών.

Επομένως, οι συγκρίσεις των μοντέλων πρέπει να συμπεριλαμβάνουν και κάθε δυνατό συνδυασμό αυτών των υπερ-παραμέτρων. Τελικά, καταλήξαμε με τρία γραμμικά μοντέλα (γραμμική παλινδρόμηση, lasso, και ridge), πέντε διαφορετικές μεθόδους παραγωγής δεδομένων (τυχαία, Hammersley, Sobol, Halton και latin random) και τρία διαφορετικά μεγέθη δεδομένων. Τα μεγέθη που επιλέχθηκαν ήταν 500, 1,000 και 2,500 σημεία, καθώς η επιλογή αυτή μπορεί να κάνει αρκετά διακριτές τις διαφορές όσων αφορά το μέγεθος των δεδομένων. Συνολικά, στο πείραμα αυτό, εξετάσαμε 45 συνδυασμούς μοντέλων με τις δύο αυτές υπερ-παραμέτρους και μπορέσαμε να αναγνωρίσουμε τη συμπεριφορά τους στα 579 προβλήματα που χρησιμοποιήθηκαν.

5.1.2 Τα Δεδομένα

Όπως ήδη έχει αναφερθεί προηγουμένως στην εργασία, τα δεδομένα που χρησιμοποιήθηκαν παράχθηκαν με τη χρήση αλγορίθμων οι οποίοι παράγουν N σημεία. Οι αλγόριθμοι αυτοί είναι προγραμματιστικές υλοποιήσεις των ακολουθιών χαμηλής ασυμφωνίας, που αναφέρθηκαν προηγουμένως. Αυτό σημαίνει ότι για N σημεία ο αλγόριθμος παραγωγής δεδομένων, θα χρησιμοποιήσει κάποια από της ακολουθίες N φορές για να παράξει τα δεδομένα εισαγωγής (input) και ύστερα θα καλέσει το εκτελέσιμο αρχείο του αντίστοιχου προγράμματος για να υπολογίσει τη λύση του κάθε σημείου εισαγωγής (output). Αυτό θα πραγματοποιηθεί P φορές, όπου P είναι ο αριθμός των προβλημάτων που θα χρησιμοποιηθούν. Επομένως, για κάθε προσπέλαση του πειράματος θα παραχθούν $N * P$ μέγεθος δεδομένων. Επιπλέον, πρέπει να σημειωθεί ότι δεν είναι δυνατόν να δημιουργηθούν δεδομένα με τον μέγιστο αριθμό σημείων που επιθυμούμε και ύστερα να διχοτομηθεί αυτό το σύνολο έτσι ώστε να έχουμε το σύνολο δεδομένων με μικρότερο αριθμό σημείων, για να γλιτώσουμε υπολογιστικό χρόνο. Αυτό οφείλεται στη φύση των ακολουθιών χαμηλής ασυμφωνίας, καθώς, παρότι είναι ντετερμινιστικές ακολουθίες, τα σημεία που παράγουν εξαρτώνται από το συνολικό αριθμό σημείων που τελικά θα παραχθούν. Έτσι, για τις τρεις

προσπελάσεις του πειράματος θα δημιουργηθούν $[N \times (M+1)] * P$ πίνακες δεδομένων, όπου ο αριθμός σημείων (500, 1,000, 2,500), $M+1$ ο αριθμός των διαστάσεων του προβλήματος και η λύση του, και P ο αριθμός των προβλημάτων. Όπως είναι προφανές, ο χρόνος παραγωγής των δεδομένων αυτών ήταν εξαιρετικά μεγάλος, με την ανάγκη δεκάδων ωρών εκτέλεσης του κώδικα (ακόμα και με τη παραλληλοποίησή του) και τα δεδομένα που παρήχθησαν χρειάστηκαν ιδιαίτερη μεταχείριση τόσο στη μεταποίηση και ανάλυσή τους, όσο και στην παρουσίασή τους.

5.2 Οπτικοποίηση Δεδομένων

Όπως έχει ήδη αναφερθεί στη προηγούμενη ενότητα, το πείραμα δημιουργεί πολλαπλές περιπτώσεις που πρέπει να μελετήσουμε και με τη σειρά τους, ένα μεγάλο πλήθος δεδομένων. Η αναπαράσταση των πολλαπλών GB δεδομένων που παρήχθησαν δεν είναι δυνατόν να αναπαρασταθούν στο κείμενο της εργασίας αυτής. Για αυτό το λόγο θα γίνει η παρουσίαση των αποτελεσμάτων και των δεδομένων σε συγκεντρωμένη (aggregated) μορφή. Αυτό επιτυγχάνεται με τη χρήση γραφημάτων και μεθόδων συγκέντρωσης των αποτελεσμάτων, όπως ο υπολογισμός μέσης τιμής. Οι τύποι διαγραμμάτων που θα χρησιμοποιηθούν είναι Bar plots , Box plots και Scatter plots , των οποίων ο τρόπος ερμηνείας έχει εξηγηθεί στο κεφάλαιο της Υλοποίησης.

5.3 Αποτελέσματα

Τα δεδομένα που δημιουργήθηκαν και τα αντίστοιχα αποτελέσματά τους, έχουν ομαδοποιηθεί σε 45 ομάδες όπως έχουμε ήδη αναφέρει (μοντέλο παλινδρόμησης, αριθμός σημείων, ακολουθία χαμηλής ασυμφωνίας). Κατά συνέπεια οι συγκρίσεις που ακολουθούν γίνονται μεταξύ αυτών των 45 συνδυασμών.

5.3.1 Αξιολόγηση Μοντελοποίησης

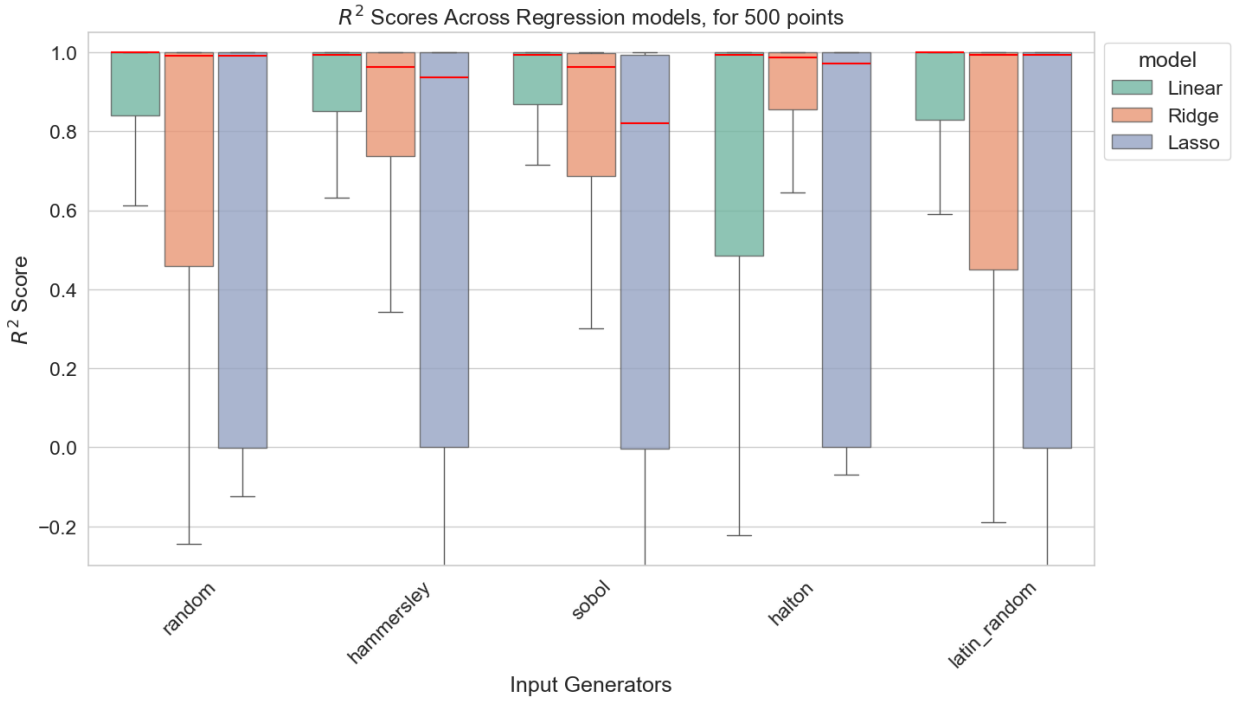
Αρχικά, πραγματοποιήθηκε η αξιολόγηση των μοντέλων παλινδρόμησης που χρησιμοποιήθηκαν για αυτούς τους 45 συνδυασμούς. Η αξιολόγηση έγινε με τη χρήση τεσσάρων μετρικών, ευρέως γνωστών, οι οποίοι χρησιμοποιούνται για την αξιολόγηση μοντέλων παλινδρόμησης, τα R^2, MSE, $RMSE$, και MAE. Οι τρεις τελευταίες μετρικές όπως μπορούμε να συμπεράνουμε και από τους τύπους τους (2.3, 2.5,

2.6, 2.7) είναι αρκετά αντίστοιχες και όπως παρατηρήθηκε, στη χρήση τους, δεν υπήρχαν σημαντικές αποκλίσεις ώστε να μπορέσουν να γίνουν εύκολες συγκρίσεις. Όλες οι συγκρίσεις πραγματοποιήθηκαν με τη χρήση του R^2 με το οποίο διακρίνονταν αρκετά καλά οι διαφορές των μοντέλων παλινδρόμησης. Έτσι, στο κείμενο θα χρησιμοποιηθούν μόνο τα γραφήματα που σχετίζονται με το R^2 και τα υπόλοιπα θα παρατεθούν στο παράρτημα της εργασίας.

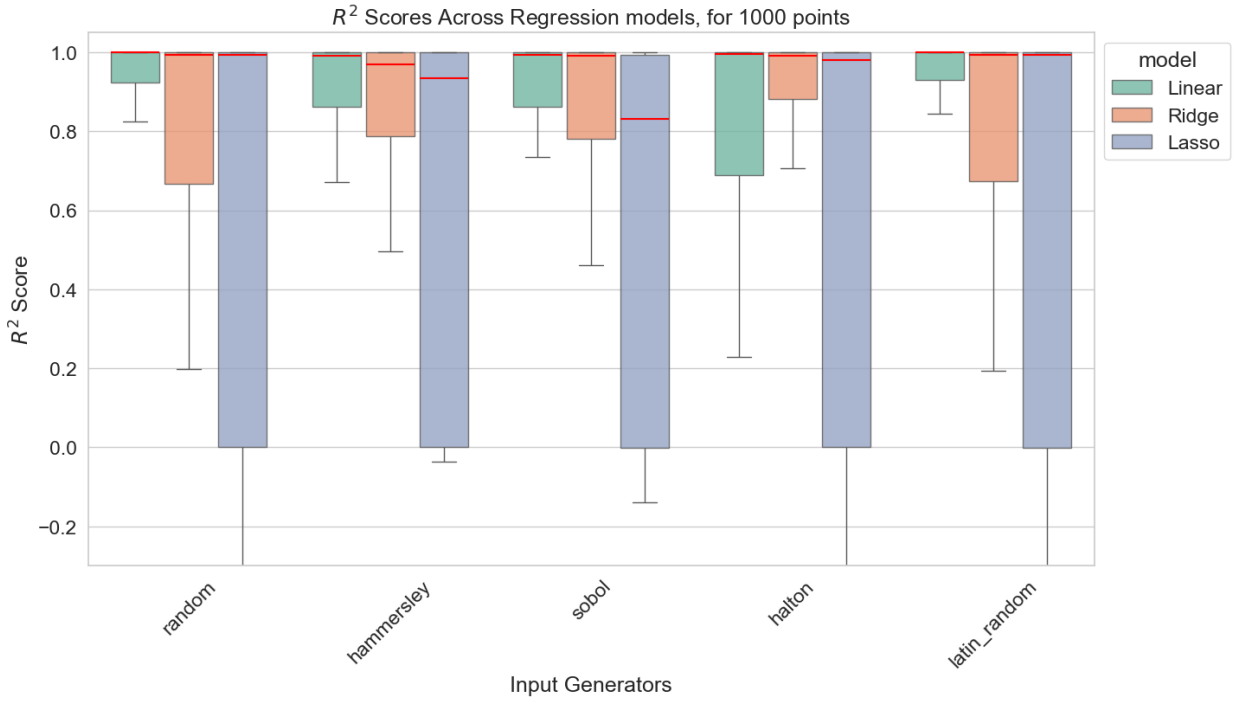
Για τη σύγκριση των R^2 μετρικών των μοντέλων στα 579 προβλήματα γίνεται χρήση των διαγραμμάτων box plot. Οι 45 συγκρίσεις που απαιτούνται χωρίστηκαν σε τρία διαγράμματα των 15 box plot, ομαδοποιημένα σύμφωνα με τον αριθμό σημείων που δημιουργήθηκαν. Αυτά τα βλέπουμε στα παρακάτω σχήματα (Σχήματα 5.1, 5.2, 5.3), όπου είναι περαιτέρω ομαδοποιημένα ανά τύπου ακολουθίας που χρησιμοποιήθηκε στον άξονα-x και ανά μοντέλο που χρησιμοποιήθηκε με τον χρωματισμό των κουτιών, όπως είναι φανερό και από το υπόμνημα των γραφημάτων.

Αρχικά, παρατηρούμε ότι σχεδόν σε κάθε περίπτωση, η διασπορά των R^2 είναι μικρότερη με τη χρήση του γραμμικού μοντέλου και μεγαλύτερη με τη χρήση του μοντέλου lasso. Παράλληλα, στην περίπτωση της χρήσης της ακολουθίας Halton, βλέπουμε ότι η διασπορά των R^2 του μοντέλου ridge είναι η μικρότερη απο τα τρια. Επίσης παρατηρούμε ότι όσο λιγότερα σημεία δεδομένων εισαγωγής έχουμε δημιουργήσει τόσο μεγαλύτερη διασπορά έχουν οι τιμές R^2, κάτι που είναι σταθερό για όλους τους συνδυασμούς. Όσον αφορά την κατανομή των τιμών αυτών μπορούμε τα διακρίνουμε ότι σε κάθε συνδυασμό, το μοντέλο κάνει αρκετά καλό fit καθώς στο 50 τοις εκατο των προβλημάτων οι τιμές του R^2 είναι μεγαλύτερες από 0.9. Η μόνη εξαίρεση είναι η Sobol στο μοντέλο lasso, η οποία έχει ενδιάμεση τιμή κάτω από 0.9 αλλά μεγαλύτερη του 0.8, κάτι αρκετά θετικό. Ωστόσο, δεν μπορούμε να παραλείψουμε ότι το υπόλοιπο 50 τοις εκατο των προβλημάτων έχει πιο χαμηλές τιμές, ειδικά στην περίπτωση της lasso για κάθε ακολουθία όπου στο 25 τοις εκατο των περιπτώσεων το R^2 είναι αρνητικό, ένδειξη ότι το μοντέλο δεν κατάφερε να κάνει fit. Τέτοιες περιπτώσεις αρνητικού R^2 υπάρχουν στα υπόλοιπα μοντέλα και αλγορίθμους ακολουθιών που χρησιμοποιήθηκαν, αλλά παρατηρούμε ότι με την αύξηση των σημείων που παράγουμε, το γραμμικό μοντέλο και το ridge έχουν πάντα θετικές τιμές στο R^2 (Σχήματα 5.2, 5.3).

Λαμβάνοντας υπόψιν την επίδοση των μοντέλων σε όλες τις περιπτώσεις, μπο-

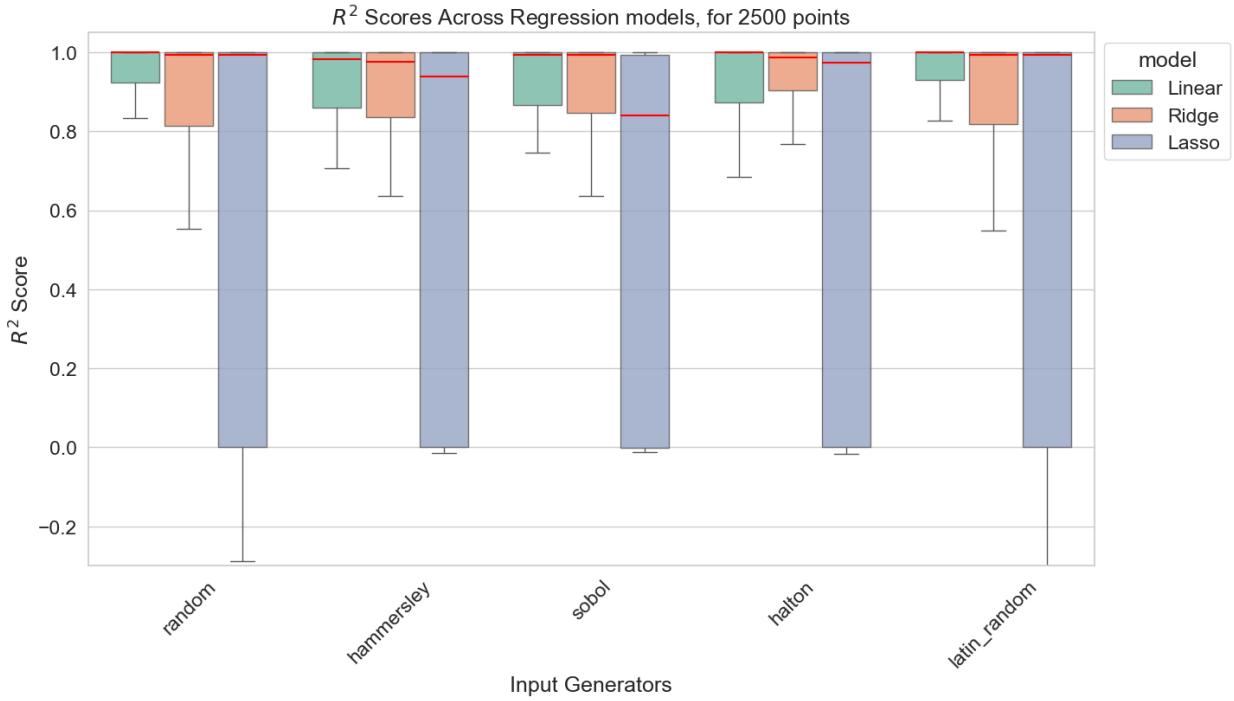


Σχήμα 5.1: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 500 σημείων



Σχήμα 5.2: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 1,000 σημείων

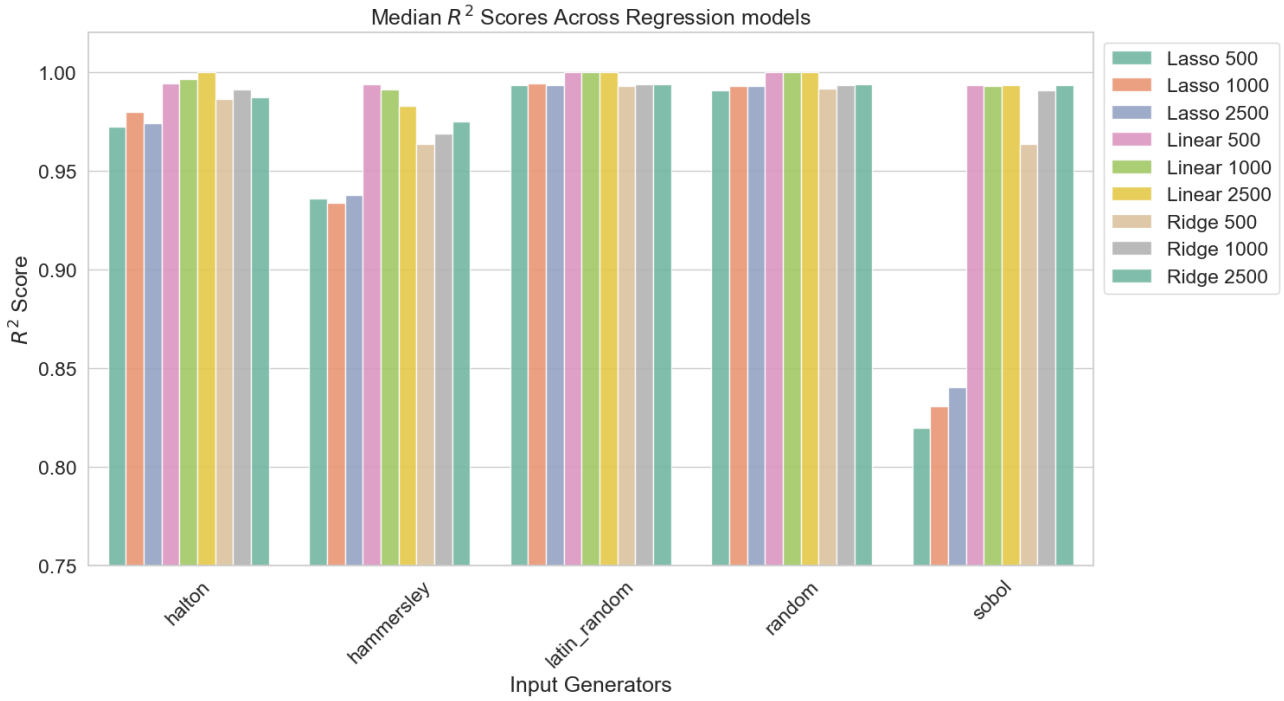
ρούμε να μελετήσουμε τη μέση περίπτωση. Όπως παρατηρήθηκε από τα θηκογράμματα (Σχήματα 5.1, 5.2, 5.3), το R^2 έχει ενδιάμεσο τιμή αρκετά κοντά στο 1, που σημαίνει ότι η απόδοση των μοντέλων είναι αρκετά καλή. Στο Σχήμα 5.4 βλέπουμε τις ενδιάμεσες τιμές για κάθε περίπτωση σε ένα ραβδοδιάγραμμα, έτσι ώστε να



Σχήμα 5.3: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 2,500 σημείων

μπορέσουμε να κάνουμε καλύτερη σύγκριση των τιμών. Μπορούμε να διακρίνουμε διαφορές μεταξύ διαφορετικών ακολουθιών και μεταξύ μοντέλων, παρόλα αυτά, δεν υπάρχει αξιόλογη διαφορά μεταξύ του αριθμού των σημείων που δημιουργήθηκαν. Βλέπουμε ότι η latin random και η random έχουν σταθερά την ενδιάμεση τιμή τους κοντά στο 1, ανεξαρτήτως μοντέλου, ενώ στις υπόλοιπες ακολουθίες μπορούμε να δούμε μια μικρή πτώση στην ενδιάμεση τιμή στη χρήση του lasso. Κατα τα άλλα, αυτές οι διαφορές είναι εξαιρετικά μικρές καθώς όλες οι τιμές είναι άνω του 0.8.

Βέβαια η ενδιάμεση τιμή δεν είναι η πιο κατάλληλη μέθοδος από μόνη της για να έχουμε μια ολοκληρωμένη εικόνη για τη μέση περίπτωση. Στο ραβδόγραμμα (Σχήμα 5.5) μπορούμε να πάρουμε μια καλύτερη εικόνα για τις τιμές του R^2. Αμέσως διακρίνουμε ότι η μέση τιμή για του R^2 αρκετές ράβδους είναι κατω του μηδενος. Αυτό παρατηρήθηκε μόνο στη χρήση των ακολουθιών Halton και Hammersley και κυρίως στο γραμμικό μοντέλο και στο μοντέλο lasso. Για το μοντέλο lasso ήταν αναμενόμενο αν αναλογιστούμε ότι το 25 τοις εκατό των τιμών είναι αρνητικό, ωστόσο, για το γραμμικό μοντέλο, κάτι τέτοιο έρχεται σε αντιπαράθεση με τις προηγούμενες παρατηρήσεις μας. Συνδυάζοντας τη πληροφορία που μας δίνει το θηκόγραμμα και το ραβδόγραμμα της μέσης τιμής, μπορούμε να συμπεράνουμε ότι οι αρνητικές μέσες τιμές του R^2, τουλάχιστον για το γραμμικό μοντέλο, οφείλονται αποκλειστικά

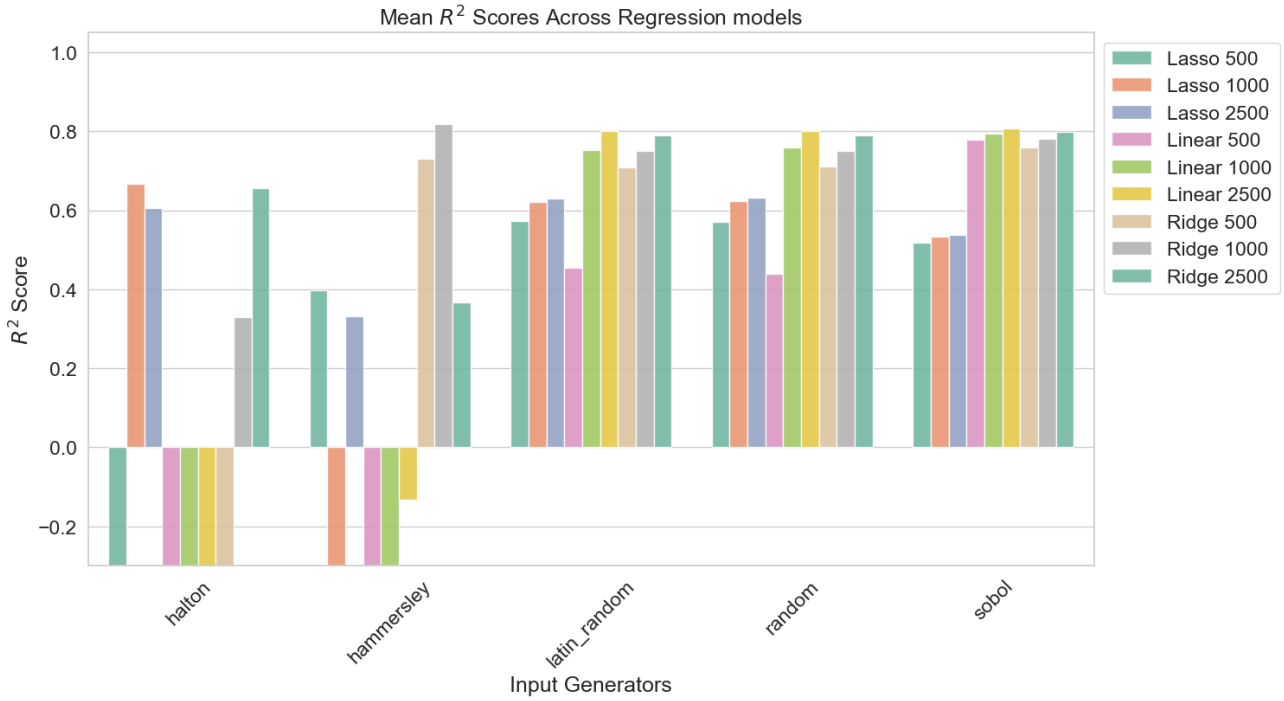


Σχήμα 5.4: Διάμεσος R^2 τιμή των μοντέλων ανά ομάδα

σε περιπτώσεις ακροτάτων τιμών. Έτσι, μπορούμε να καταλήξουμε ότι δεν σημαίνει ότι τα μοντέλα αυτά δεν μπορούν να κάνουν fit στη μέση περίπτωση. Ακόμα, πρέπει να σημειωθεί ότι ο αριθμός των σημείων βελτιώνει αρκετά αυτές τις μέσες τιμές και επανέρχονται άνω του μηδενός. Μπορούμε να δούμε ότι ο αριθμός των σημείων βελτιώνει τη μέση τιμή του R^2 σε κάθε συνδυασμό μοντέλου και ακολουθίας. Καταλήγοντας, είναι φανερό ότι η μέση τιμή από μόνη της μπορεί να είναι παραπλανητική, και είναι ωφέλιμο να συμπεριλάβουμε και τη διασπορά των τιμών αυτών, καθώς οι ακρότατες τιμές μπορεί να επηρεάσει τα αποτελέσματα.

5.3.2 Επίδραση μεγέθους δεδομένων στην απόδοση της μοντελοποίησης

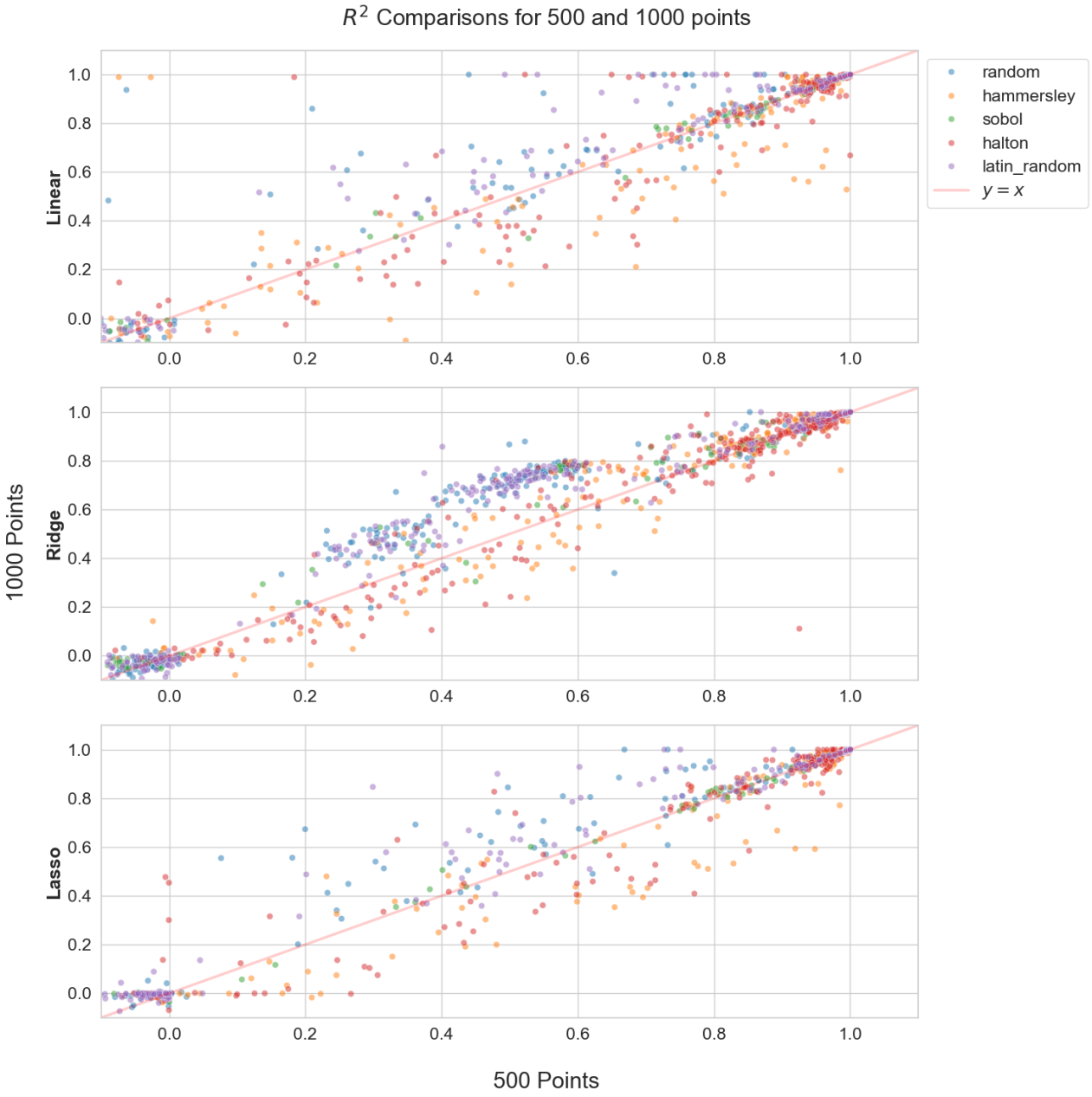
Στην προηγούμενη ενότητα πραγματοποιήσαμε συγκρίσεις κυρίως με γνώμονα τον τύπο της ακολουθίας που χρησιμοποιήθηκε. Σε αυτή την ενότητα αναλύσαμε τη διαφορά της επίδοσης των μοντέλων σχετικά με το μέγεθος των δεδομένων που χρησιμοποιήθηκαν. Έγιναν τρεις εκδοχές του πειράματος, έτσι ώστε να δημιουργηθούν τρία διαφορετικά σύνολα δεδομένων με 500, 1,000 και 2,500 σημεία. Προκειμένου να γίνει αυτή η σύγκριση, δημιουργήθηκαν τα διαγράμματα συσχέτισης (Σχήματα 5.6, 5.7, 5.8) ώστε να μπορέσουμε να συγκρίνουμε τη μετατόπιση του R^2 με την αλλαγή του πλήθους των δεδομένων.



Σχήμα 5.5: Μέση τιμή R^2 των μοντέλων ανά ομάδα

Στα σχήματα κάθε σημείο αναπαριστά έναν συγκεκριμένο συνδυασμό (μοντέλο M, ακολουθία A, πρόβλημα P) στον χώρο R^2 δύο διαφορετικών μεγεθών σημείων. Επομένως, για τον άξονα-y αν ο αριθμός σημείων του συνόλου δεδομένου ειναι Υ και αντίστοιχα για τον άξονα-x ο αριθμός σημείων του συνόλου δεδομένου ειναι Χ, τότε η απόδοση ενός συνδυασμού (M, A, P) αναπαριστάται στο χώρο ΧΥ ώς (R^2_X, R^2_Y). Παράλληλα, διαγράφεται η ευθεία $y = x$ για να μας βοηθήσει στην ανάγνωση του διαγράμματος. Σε κάθε διάγραμμα ο άξονας- y εκφράζει τη διάσταση με το μεγαλύτερο αριθμό σημείων. Επομένως, τα σημεία που βρίσκονται πάνω από την $y = x$ μας ενημερώνουν ότι το R^2 βελτιώθηκε με την αύξηση των δεδομένων, ενώ για τα σημεία που βρίσκονται κάτω από την ευθεία, σημαίνει το αντίθετο. Επίσης, Καθώς είναι δύσκολο να διακρίνουμε τη πυκνότητα των σημείων στις διάφορες περιοχές του χώρου, δεν θα αναφερθούμε ιδιαίτερα στις τιμές τους και τη διακύμανσή τους τους, καθώς για αυτό πρέπει να συμβουλευτούμε τα θηκογράμματα (Σχήματα 5.1, 5.2, 5.3). Επιπλέον, τα διαγράμματα αυτά χωρίστηκαν ανα μοντέλο που χρησιμοποιήθηκε και τα σημεία τους χρωματίστηκαν ανα τύπο ακολουθίας.

Στο Σχήμα 5.6 παρατηρούμε τη μεγαλύτερη αύξηση των τιμών του R^2 στη χρήση του μοντέλου ridge. Αυτή η βελτίωση συμβαίνει κυρίως στις ακολουθίες σημείων που παρήχθησαν από τις latin random και random. Αυτό το φαινόμενο το βλέπουμε



Σχήμα 5.6: Γράφημα σημείων σύγκρισης των R^2 τιμών μεταξύ 500 και 1,000 σημείων δεδομένων

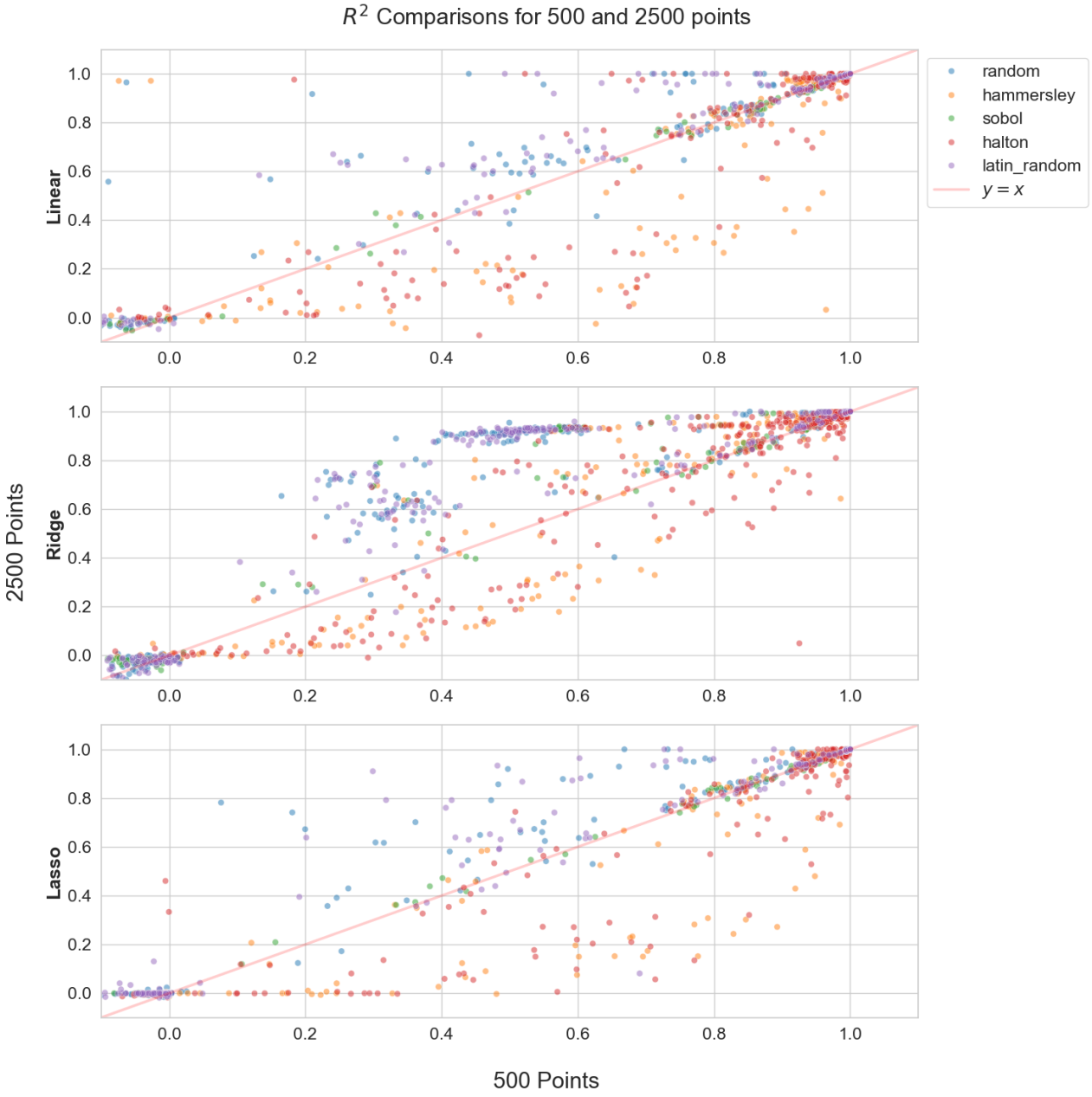
ακόμα πιο έντονα στη σύγκριση 500/2,500 στο Σχήμα 5.7 καθώς επίσης και στο Σχήμα 5.8.

Επιπλέον, παρατηρούμε ότι η Halton και η Hammersley, έχουν αρκετά σημεία κοντά στο (1, 1), καθώς επίσης βλέπουμε ότι κυρίως αυτές τείνουν να έχουν σημεία κάτω από την $y = x$, που σημαίνει ότι παρατηρείται μείωση του R^2 με την αύξηση του συνόλου δεδομένων. Η αυξητική συμπεριφορά των latin random και random και η κατηφορική συμπεριφορά των Hammersley και Halton, φαίνεται να είναι σταθερό μοτίβο, σε κάθε σύγκριση και ανεξαρτήτως μοντέλου. Με τη μόνη διαφορά ότι η

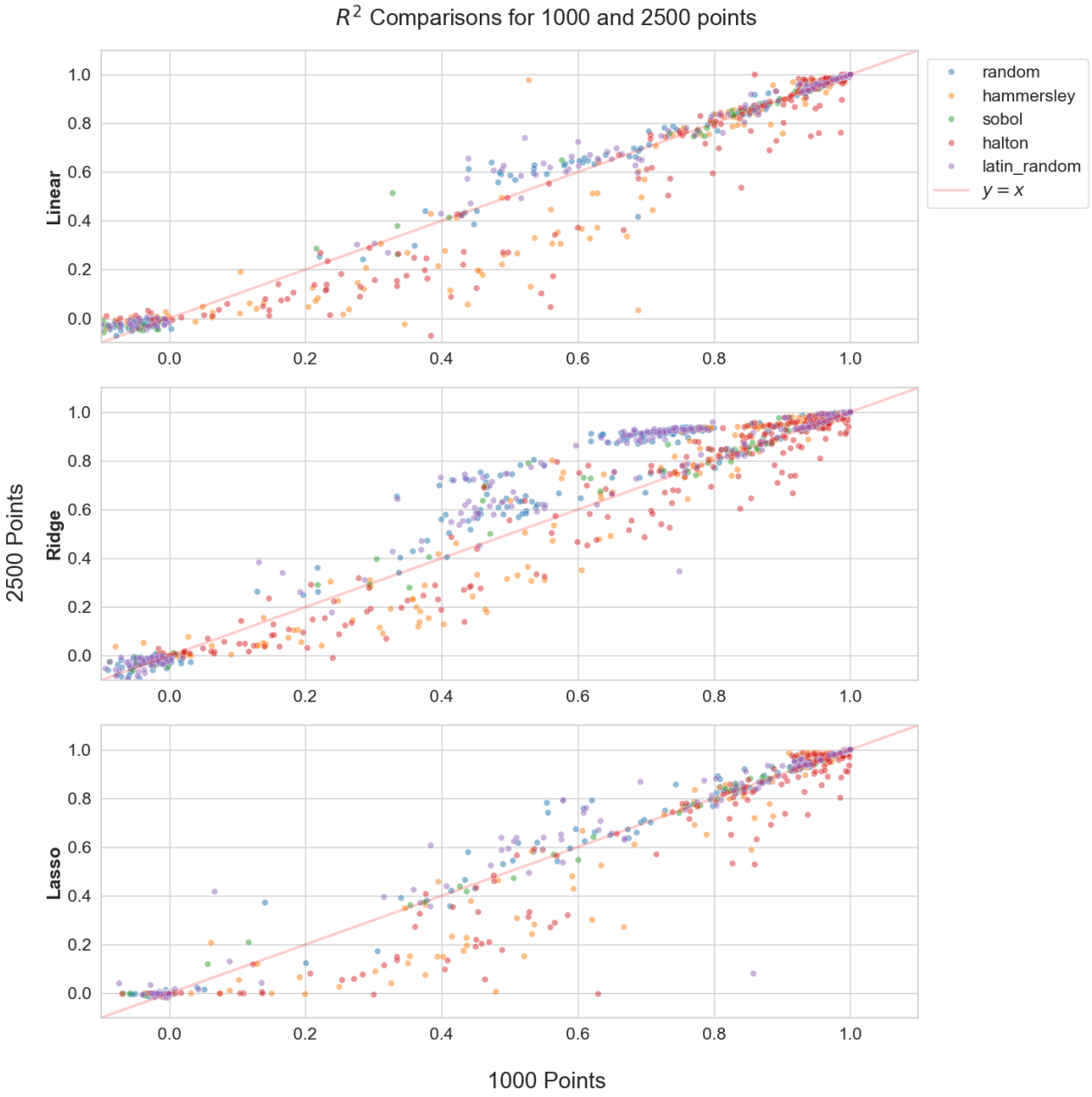
ridge είναι πιο ευαίσθητη στις αυξήσεις των latin random και random.

Τέλος, διακρίνουμε μια σταθερότητα στη Sobol, καθώς όλα τα σημεία της είναι πολύ κοντά στο $y = x$.

Ακόμα, μεταξύ των Σχημάτων 5.7 και 5.8 παρατηρούμε ότι τα σημεία που δεν είναι κοντά στο (0, 0) και στο (1, 1), τείνουν να πλησιάζουν την ευθεία $y = x$. Αυτό ίσως είναι ένδειξη ότι από ένα μέγεθος δεδομένων και πάνω, δεν θα υπάρχουν σημαντικές διαφορές στην επίδοση των μοντέλων.



Σχήμα 5.7: Γράφημα σημείων σύγκρισης των R^2 τιμών μεταξύ 500 και 2,500 σημείων δεδομένων

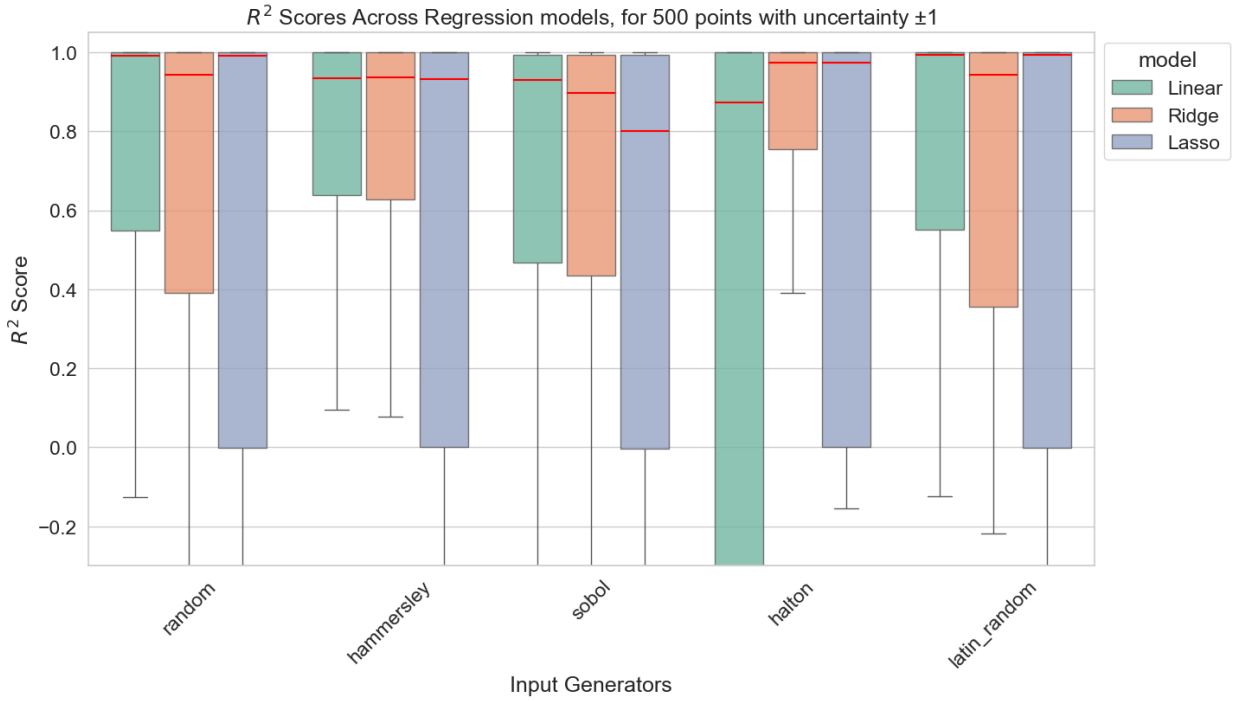


Σχήμα 5.8: Γράφημα σημείων σύγκρισης των R^2 τιμών μεταξύ 1,000 και 2,500 σημείων δεδομένων

5.3.3 Ανάλυση Αβεβαιότητας

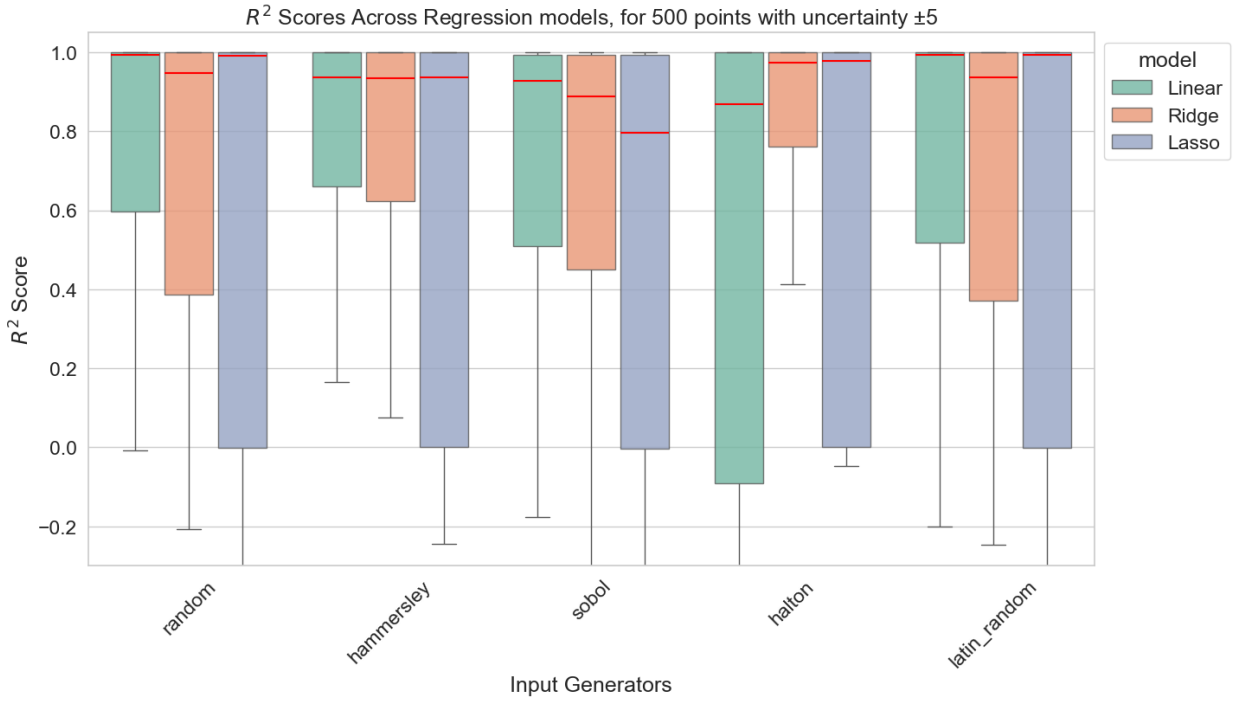
Σε αυτή την ενότητα, παρουσιάζονται τα αποτελέσματα των μετρικών αξιολόγησης των μοντέλων παλινδρόμησης, σε τροποποιημένα δεδομένα που έχει εισαχθεί αβεβαιότητα με τη μορφή θορύβου σε δύο επίπεδα (υψηλό θόρυβο και χαμηλό θόρυβο). Στη συνέχεια θα σημειωθούν παρατηρήσεις σε θηκογράμματα που παρουσιάζουν την R^2 μετρική στα δύο επίπεδα αβεβαιότητας και για κάθε σύνολο δεδομένων που χρησιμοποιήθηκε (500, 1,000, 2,500).

Στα Σχήματα 5.9 και 5.10, μπορούμε να παρατηρήσουμε τις διαφορές στην επίδοση των μοντέλων όταν έχουμε εισάγει θόρυβο σε σύνολο δεδομένων 500 σημείων. Μεταξύ των δύο δεν παρατηρούνται ιδιαίτερες διαφορές εκτός από την γενική επίδοση του γραμμικού μοντέλου με τη μέθοδο δειγματοληψίας Halton, όπου φαίνεται το θηκόγραμμα να μικραίνει προς τους θετικούς αριθμούς όταν η αβεβαιότητα μεγαλώνει. Κάτι που σημαίνει ότι υπήρχαν περισσότερες περιπτώσεις αυτού του συνδυασμού όπου το μοντέλο κατάφερε να προσεγγίσει τη λύση των προβλημάτων, καλύτερα όταν υπήρχε μεγαλύτερη αβεβαιότητα. Αξιοσημείωτο είναι επίσης, να γίνει σύγκριση με τις επιδόσεις των μοντέλων στα δεδομένα δίχως θόρυβο. Εδώ παρατηρούμε ότι το μοντέλο Lasso, μοιάζει να μην επηρεάζεται σχεδόν καθόλου όταν υπάρχει θόρυβος στα δεδομένα, κάτι που δεν μας προσφέρει κάποιο ιδιαίτερο αποτέλεσμα, καθώς στο Σχήμα 5.1 μπορούμε να δούμε ότι ήταν το μοντέλο που είχε τη χαμηλότερη επίδοση. Αντιθέτως, στις περιπτώσεις του γραμμικού μοντέλου και του Ridge, παρατηρούμε καθαρή πτώση των επιδόσεων. Όλες οι ενδιάμεσες τιμές του R^2 φαίνεται να καθαρά να μειώνονται με την εξαίρεση του γραμμικού μοντέλου στη random και τη latin random.



Σχήμα 5.9: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 500 σημείων με χαμηλό θόρυβο

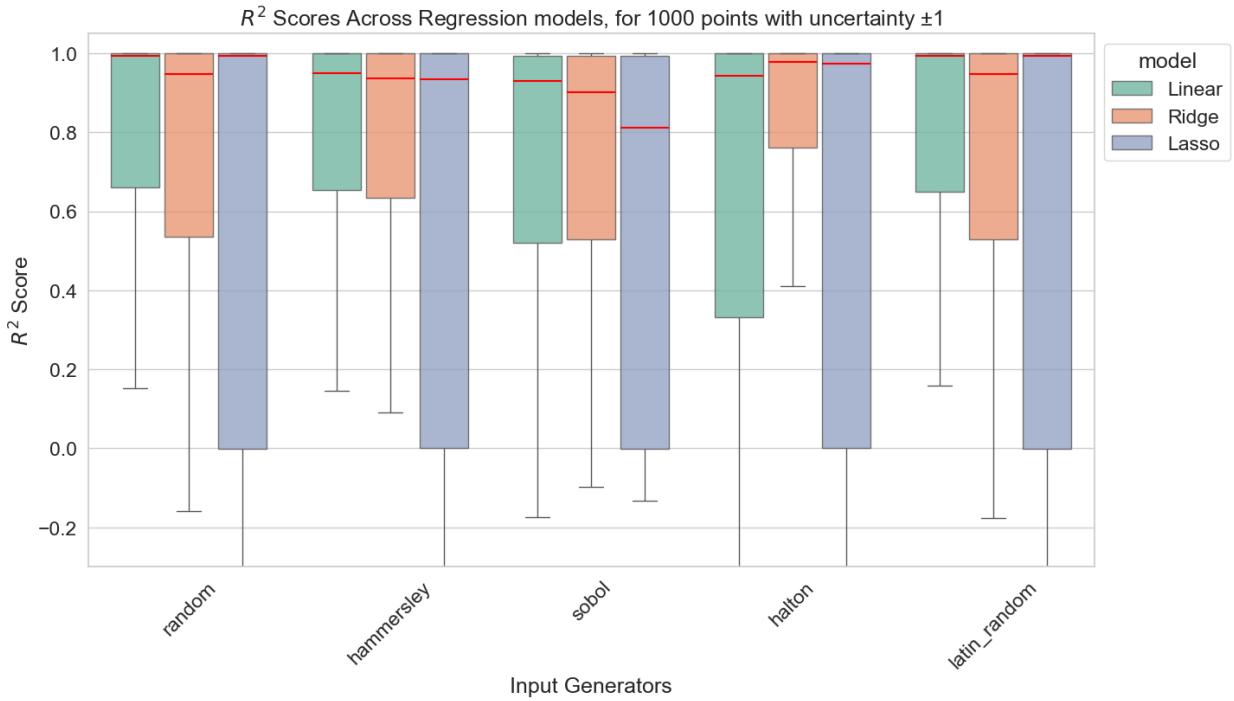
Εν συνεχεία, στις περιπτώσεις των δεδομένων των 1,000 σημείων στα Σχήματα 5.11 και 5.12, και των δεδομένων των 2,500 σημείων, στα Σχήματα 5.13 και 5.14,



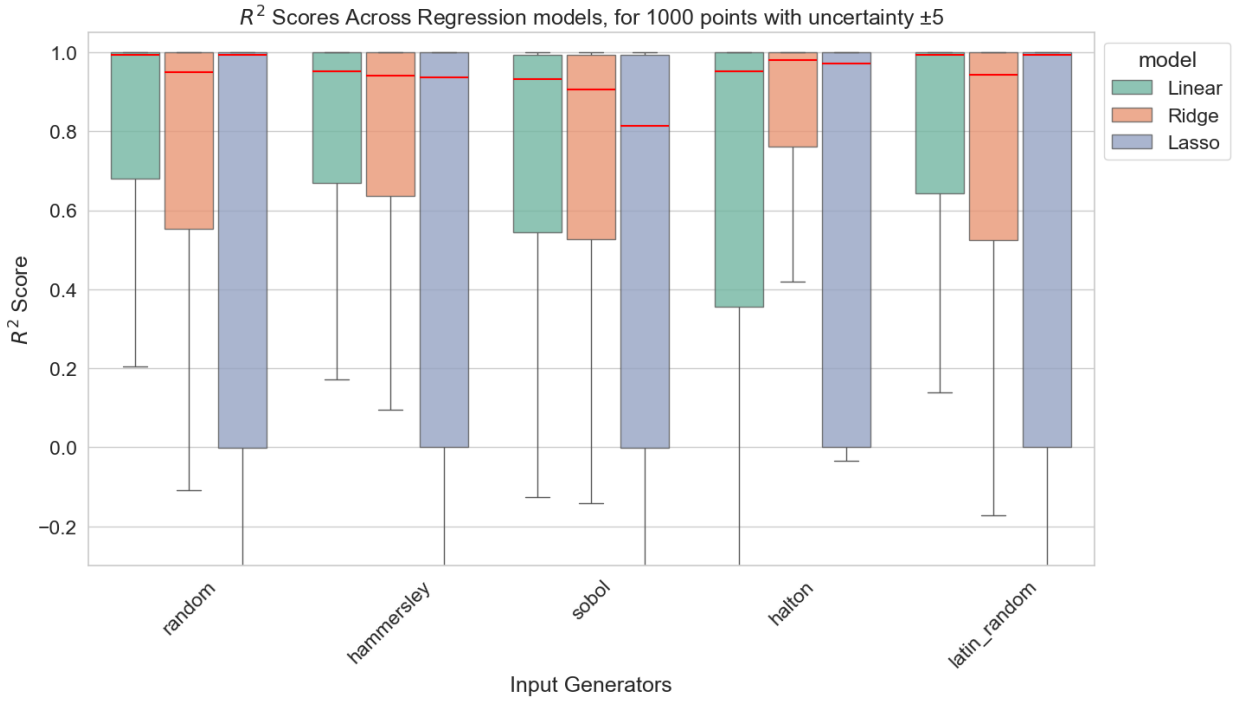
Σχήμα 5.10: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 500 σημείων με υψηλό θόρυβο

βλέπουμε το ίδιο μοτίβο διαφορών και ομοιοτήτων όπως και στα δεδομένα 500 σημείων.

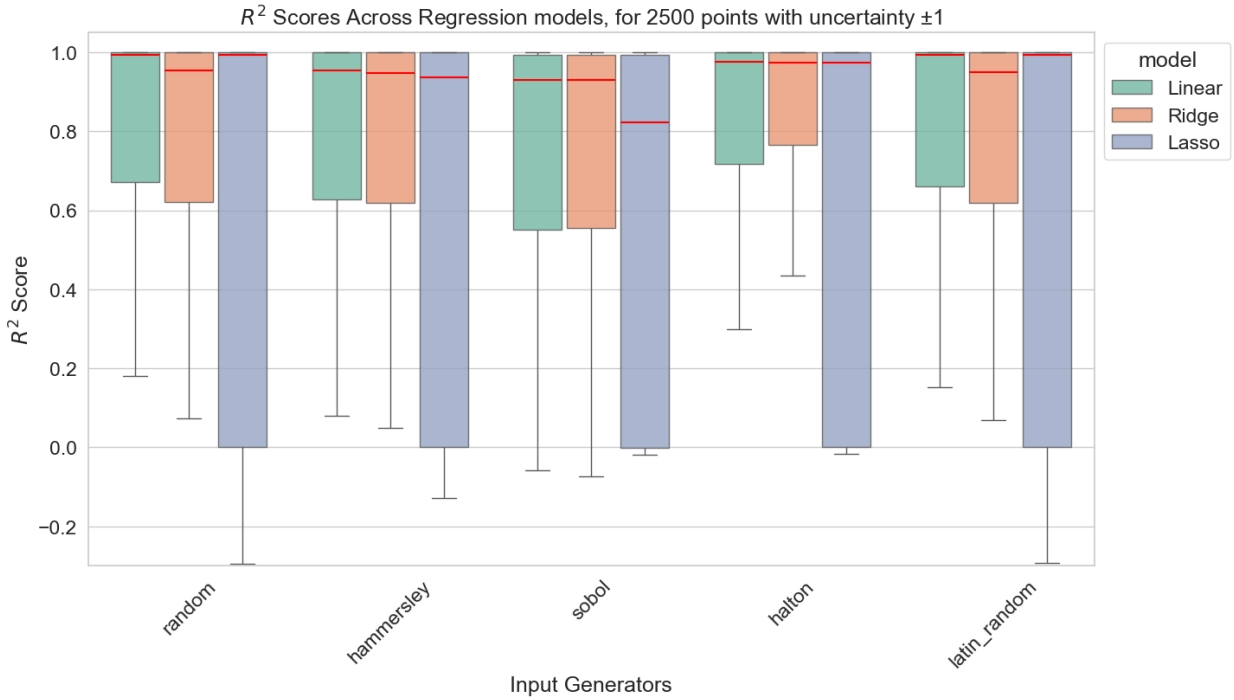
Επιπλέον, είναι φανερό ότι και στις περιπτώσεις που εισάγουμε θόρυβο, όσο περισσότερα σημεία υπάρχουν τόσο καλύτερη επίδοση έχουν τα μοντέλα. Στην περίπτωση των δεδομένων υπό αβεβαιότητα μπορούμε να διακρίνουμε ότι το μοντέλο Ridge με τη χρήση της μεθόδου δειγματοληψίας Halton, φαίνεται να έχει σταθερά την καλύτερη επίδοση όλων, ανεξαρτήτως υψηλού ή χαμηλού θορύβου. Αυτό είναι κάτι που έρχεται σε αντίθεση με τα αποτελέσματα των δεδομένων χωρίς θόρυβο, όπου μοιάζει ότι επί το πλείστων το γραμμικό μοντέλο είχε την καλύτερη επίδοση.



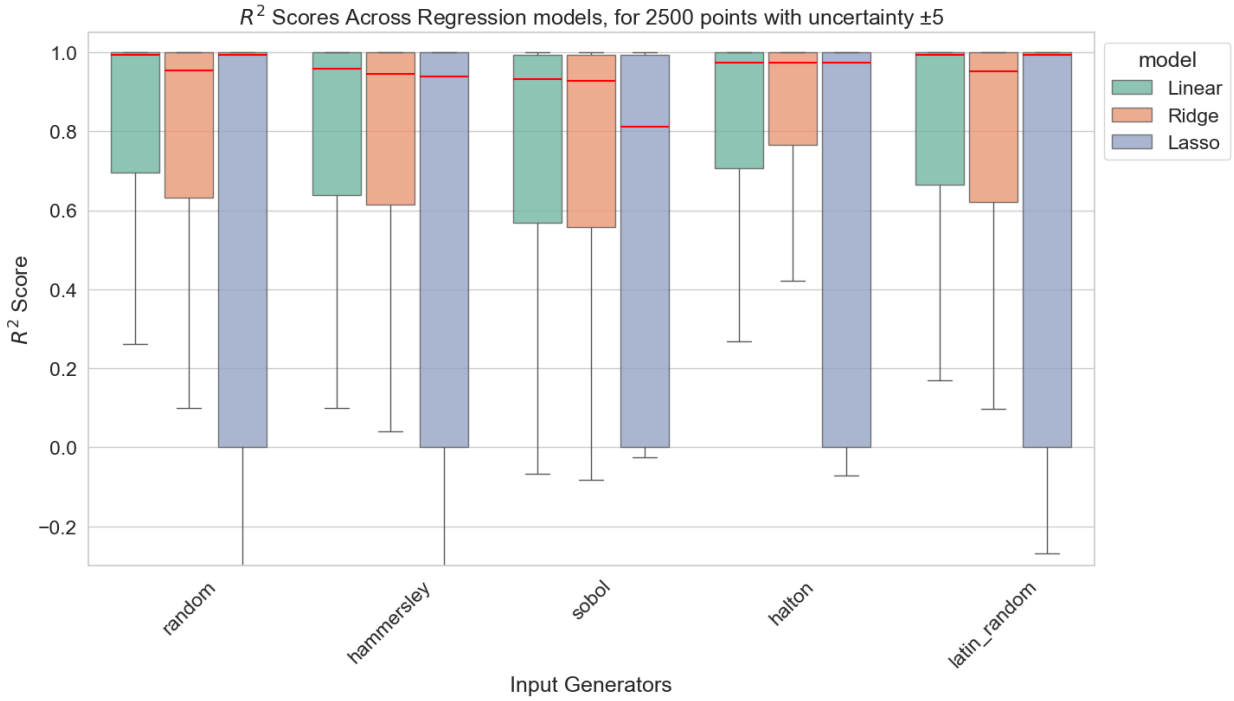
Σχήμα 5.11: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 1,000 σημείων με χαμηλό θόρυβο



Σχήμα 5.12: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 1,000 σημείων με υψηλό θόρυβο



Σχήμα 5.13: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 2,500 σημείων με χαμηλό θόρυβο



Σχήμα 5.14: Boxplot παρουσίασης των R^2 τιμών σε δεδομένα 2,500 σημείων με υψηλό θόρυβο

Κεφάλαιο 6

Συμπεράσματα

Σε αυτή τη διπλωματική εργασία διερευνήσαμε την επιρροή που είχε η επιλογή υπερ-παραμέτρων σε γραμμικά μοντέλα μηχανικής μάθησης στην απόδοση τους , μειώνοντας την αβεβαιότητα στα δεδομένα. Δημιουργήσαμε ένα πρόγραμμα με το οποίο ρυθμίζοντας τις επιθυμητές υπερ-παραμέτρους (αριθμό σημείων N και επιλογή ακολουθίας χαμηλής ασυμφωνίας) εξάγουμε κλασικές μετρικές μοντέλων παλινδρόμησης (R^2, MSE, RMSE και MAE). Τις μετρικές αυτές τις χρησιμοποιήσαμε για να οδηγηθούμε στα εξής συμπεράσματα.

Αρχικά, παρατηρήσαμε ότι τα μοντέλα, ανεξαρτήτως του αριθμού των σημείων που χρησιμοποιήθηκαν και του τύπου της ακολουθίας, φαίνεται να κάνουν αρκετά καλό "fit" στα περισσότερα προβλήματα, με την εξαίρεση των μοντέλων lasso για τα οποία σχεδόν το 50 τοις εκατο των πειραμάτων είχαν πολύ χαμηλές ή αρνητικές τιμες.

Η επιλογή ακολουθίας χαμηλής ασυμφωνίας για τη δημιουργία των δεδομένων εισαγωγής φαίνεται να είναι σχετικά δύσκολη και να μην μπορούμε να καταλήξουμε σε ξεκάθαρο συμπέρασμα για το αν μια ακολουθία είναι καλύτερη από την άλλη. Παρόλα αυτά καταλήξαμε ότι η χρήση της κάθε ακολουθίας έχει διαφορετική επίδραση στην απόδοση των μοντέλων, όταν αλλάζει ο αριθμός των σημείων που δημιουργούν, στις περιπτώσεις προβλημάτων που το R^2 δεν είναι κοντά στο 1. Πιο συγκεκριμένα, η απόδοση των ακολουθιών latin random και random φαινεται να βελτιώνεται με τη δημιουργία περισσότερων σημείων, ενώ στην περίπτωση της Hammersley και της Halton η απόδοση φαίνεται να χειροτερεύει. Σε αντίθεση με τις υπόλοιπες, η Sobol φαίνεται να παραμένει σταθερή.

Μελετώντας την απόδοση των μοντέλων, μπορούμε να συμπεράνουμε με μεγάλη

βεβαιότητα ότι η απόδοση των μοντέλων βελτιώνεται σε όλες τις περιπτώσεις με την αύξηση των σημείων που δημιουργήσαμε. Επομένως, σε περίπτωση κακής απόδοσης ενός μοντέλου που προσπαθεί να κάνει "fit" σε ένα πρόβλημα , για να το βελτιώσουμε μπορούμε να αυξήσουμε τον αριθμό των δεδομένων.

Τέλος, από την ανάλυση των αποτελεσμάτων των δεδομένων υπό αβεβαιότητα, μπορούμε να συμπεράνουμε ότι ο συνδυασμός του μοντέλου Ridge με τη μέθοδο δειγματοληψίας Halton, μπορεί να μας προσφέρει σταθερά καλύτερες επιδόσεις και καλύτερο "fit" από τα υπόλοιπα μοντέλα και μεθόδους. Κάτι που είναι σταθερό ανεξαρτήτως αβεβαιότητας, και ανεξαρτήτως αριθμού σημείων δεδομένων. Ο συνδυασμός Ridge και Halton παρατηρήθηκε επίσης ότι έχει εξίσου καλές αποδόσεις και στα δεδομένα χωρίς θόρυβο με μοναδικό ανταγωνιστή το γραμμικό μοντέλο. Επομένως, μπορούμε να καταλήξουμε ότι ο συνδυασμός αυτός είναι ο πιο κατάλληλος, ειδικά όταν γνωρίζουμε ότι στα δεδομένα υπάρχει σε κάποιο βαθμό αβεβαιότητα.

Με βάση τα συμπεράσματα αυτά το λογικό επόμενο βήμα θα ήταν να ερευνηθεί περαιτέρω ο ρόλος του αριθμού σημείων στην απόδοση των μοντέλων. Συγκεκριμένα εάν μετά από κάποιο πλήθος η ανοδική τάση της απόδοσης είτε σταματάει είτε συγκλίνει προς μια τιμή. Ένα τέτοιο συμπέρασμα θα βοηθούσε πολύ στη μείωση της πολυπλοκότητας ενός αλγορίθμου και θα παρέχει ένα πιο τυποποιημένο τρόπο βελτιστοποίησης για τέτοιου είδους προβλήματα και για αυτή την υπερπαράμετρο. Τέλος ένα εξίσου ενδιαφέρον βήμα θα ήταν να διερευνηθούν ξεχωριστά οι ακολουθίες χαμηλής ασυμφωνίας που χρησιμοποιήσαμε, ώστε να διαπιστωθεί κατηγορηματικά το εύρος πλήθους σημείων στα οποία αυτές παρέχουν τη βέλτιστα ομοιόμορφη κατανομή του χώρου που μελετάται. Αυτό θα έχει ως αποτέλεσμα να γίνεται πιο σωστή επιλογή με βάσει τις ιδιαιτερότητες του κάθε προβλήματος.

Βιβλιογραφία

[1] A. Geron, *Hands-on machine learning with scikit-learn, keras, and TensorFlow*. Sebastopol, CA: O'Reilly Media, 2 ed., Oct. 2019.

[2] S. Dridi, "Supervised learning-a systematic literature review," 2021.

[3] S. Guido and A. C. Mueller, *Introduction to machine learning with python*. Sebastopol, CA: O'Reilly Media, Oct. 2016.

[4] X. Wang, W. Zhang, J. Yan, X. Yuan, and H. Zha, "On the flexibility of block coordinate descent for large-scale optimization," *Neurocomputing*, vol. 272, pp. 471–480, 2018.

[5] P. Simon, *Too big to ignore*. Wiley and SAS Business Series, Nashville, TN: John Wiley & Sons, Apr. 2013.

[6] R. Hierons, "Machine learning. tom m. mitchell. published by McGraw-Hill, maidenhead, U.K., international student edition, 1997. ISBN: 0-07-115467-1, 414 pages. price: U.K. £22.99, soft cover," *Softw. Test. Verif. Reliab.*, vol. 9, pp. 191–193, Sept. 1999.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Adaptive Computation and Machine Learning series, Cambridge, MA: Bradford Books, 2 ed., Nov. 2018.

[8] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer texts in statistics, New York, NY: Springer, 1 ed., June 2013.

[9] G. Seber and A. Lee, *Linear Regression Analysis*. Wiley Series in Probability and Statistics, Wiley, 2012.

[10] A. Georgieva and I. Jordanov, "A hybrid meta-heuristic for global optimisation using low-discrepancy sequences of points," *Comput. Oper. Res.*, vol. 37, pp. 456–469, Mar. 2010.

[11] J. v. d. Corput, "Verteilungsfunktionen i," *Proc. Kon. Ned. Akad. v. Wetensch.*, vol. 38, p. 8, 1935.

[12] J. H. Halton, "Algorithm 247: Radical-inverse quasi-random point sequence," *Commun. ACM*, vol. 7, pp. 701–702, Dec. 1964.

[13] J. Dick, F. Y. Kuo, and I. H. Sloan, "High-dimensional integration: The quasi-monte carlo way," *Acta Numer.*, vol. 22, pp. 133–288, May 2013.

[14] S. Joe and F. Y. Kuo, "Remark on algorithm 659," *ACM Trans. Math. Softw.*, vol. 29, pp. 49–57, Mar. 2003.

[15] D. Stinson, *Combinatorial Designs*. New York, NY: Springer, 2004 ed., Oct. 2003.

[16] M. J. D. Powell, "Least frobenius norm updating of quadratic models that satisfy interpolation conditions," *Math. Program.*, vol. 100, May 2004.

[17] A. R. a. R. . Conn, *Introduction to derivative-free optimization*. MPS-SIAM series on optimization, Philadelphia, MS: Society for Industrial and Applied Mathematics/Mathematical Programming Society, 2009.

[18] D. Kaplan, "On the quantification of model uncertainty: A bayesian perspective," *Psychometrika*, vol. 86, pp. 215–238, Mar. 2021.

[19] D. Robinson and C. Atcitty, "Comparison of quasi- and pseudo-monte carlo sampling for reliability and uncertainty analysis," in *40th Structures, Structural Dynamics, and Materials Conference and Exhibit*, (Reston, Virigina), American Institute of Aeronautics and Astronautics, Apr. 1999.

[20] L. Kocis and W. J. Whiten, "Computational investigations of low-discrepancy sequences," *ACM Trans. Math. Softw.*, vol. 23, pp. 266–294, June 1997.

[21] P. Bratley, B. L. Fox, and H. Niederreiter, "Implementation and tests of low-discrepancy sequences," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 2, no. 3, pp. 195–213, 1992.

[22] H. Niederreiter, "Low-discrepancy and low-dispersion sequences," *Journal of number theory*, vol. 30, no. 1, pp. 51–70, 1988.

[23] T.-T. Wong, W.-S. Luk, and P.-A. Heng, "Sampling with hammersley and halton points," *Journal of graphics tools*, vol. 2, no. 2, pp. 9–24, 1997.

[24] A. G. Ahmed, H. Perrier, D. Coeurjolly, V. Ostromoukhov, J. Guo, D.-M. Yan, H. Huang, and O. Deussen, "Low-discrepancy blue noise sampling," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–13, 2016.

[25] J. L. Deutsch and C. V. Deutsch, "Latin hypercube sampling with multidimensional uniformity," *Journal of Statistical Planning and Inference*, vol. 142, no. 3, pp. 763–772, 2012.

[26] T. Hou, D. Nuyens, S. Roels, and H. Janssen, "Quasi-Monte carlo based uncertainty analysis: Sampling efficiency and error estimation in engineering applications," *Reliab. Eng. Syst. Saf.*, vol. 191, p. 106549, Nov. 2019.

Παραρτήματα

Παράρτημα Α΄

Αποτελέσματα αλγορίθμων

Πίνακας Α΄.1: R^2 scores

model	Sequence Algorithms	Npoints	Mean	Median
Lasso	halton	500	-2.486	0.972
		1000	0.667	0.980
		2500	0.606	0.974
	hammersley	500	0.397	0.936
		1000	-0.576	0.934
		2500	0.332	0.938
	latin_random	500	0.572	0.993
		1000	0.622	0.994
		2500	0.630	0.993
	random	500	0.571	0.991
		1000	0.623	0.993
		2500	0.631	0.993
	sobol	500	0.517	0.820
		1000	0.534	0.831
		2500	0.538	0.841
Linear	halton	500	8.3×10^{21}	0.994
		1000	8.0×10^{19}	0.997
		2500	6.3×10^{18}	1.000
	hammersley	500	-10.843	0.994
		1000	-0.996	0.991
		2500	-0.134	0.983
	latin_random	500	0.455	1.000
		1000	0.752	1.000
		2500	0.802	1.000
	random	500	0.438	1.000
		1000	0.760	1.000
		2500	0.800	1.000
	sobol	500	0.778	0.993
		1000	0.794	0.993
		2500	0.806	0.994
Ridge	halton	500	-2.347	0.986
		1000	0.330	0.991
		2500	0.655	0.987
	hammersley	500	0.730	0.963
		1000	0.817	0.969
		2500	0.367	0.975
	latin_random	500	0.709	0.993
		1000	0.749	0.994
		2500	0.789	0.994
	random	500	0.710	0.992
		1000	0.751	0.993
		2500	0.789	0.994
	sobol	500	0.759	0.964
		1000	0.781	0.991
		2500	0.799	0.993

Πίνακας Α΄.2: MAE scores

model	Sequence Algorithms	Npoints	Mean	Median
Lasso	halton	500	2.4×10^{27}	265.860
		1000	2.4×10^{27}	106.522
		2500	2.3×10^{27}	94.603
	hammersley	500	3.8×10^{27}	65.326
		1000	3.8×10^{27}	53.125
		2500	3.8×10^{27}	48.221
	latin_random	500	2.3×10^{27}	65.326
		1000	2.3×10^{27}	65.700
		2500	2.3×10^{27}	66.347
	random	500	2.3×10^{27}	68.417
		1000	2.3×10^{27}	65.946
		2500	2.3×10^{27}	65.472
	sobol	500	6.0×10^{27}	18.274
		1000	6.3×10^{27}	18.323
		2500	6.2×10^{27}	17.998
Linear	halton	500	2.5×10^{27}	467.574
		1000	2.3×10^{27}	10.749
		2500	2.4×10^{27}	3.222
	hammersley	500	3.8×10^{27}	5.279
		1000	3.6×10^{27}	4.894
		2500	3.7×10^{27}	4.735
	latin_random	500	2.3×10^{27}	3.390
		1000	2.3×10^{27}	2.130
		2500	2.3×10^{27}	2.159
	random	500	2.2×10^{27}	3.538
		1000	2.3×10^{27}	2.100
		2500	2.4×10^{27}	2.183
	sobol	500	6.6×10^{27}	2.707
		1000	6.1×10^{27}	2.617
		2500	6.2×10^{27}	2.571
Ridge	halton	500	2.2×10^{27}	31.900
		1000	2.3×10^{27}	22.353
		2500	2.3×10^{27}	13.696
	hammersley	500	3.8×10^{27}	31.395
		1000	3.6×10^{27}	24.317
		2500	3.6×10^{27}	13.186
	latin_random	500	2.4×10^{27}	50.183
		1000	2.3×10^{27}	35.662
		2500	2.3×10^{27}	19.143
	random	500	2.3×10^{27}	51.769
		1000	2.4×10^{27}	34.715
		2500	2.4×10^{27}	19.151
	sobol	500	6.4×10^{27}	13.474
		1000	6.3×10^{27}	10.260
		2500	6.1×10^{27}	7.784

Πίνακας Α΄.3: MSE scores

model	Sequence Algorithms	Npoints	Mean	Median
Lasso	halton	500	2.3×10^{57}	1.0×10^{5}
		1000	2.4×10^{57}	1.9×10^{4}
		2500	2.3×10^{57}	1.5×10^{4}
	hammersley	500	3.5×10^{57}	6888.553
		1000	3.8×10^{57}	4679.428
		2500	3.6×10^{57}	3800.502
	latin_random	500	2.0×10^{57}	6642.768
		1000	2.2×10^{57}	6760.599
		2500	2.2×10^{57}	6820.175
	random	500	2.2×10^{57}	7457.347
		1000	2.1×10^{57}	6845.471
		2500	2.2×10^{57}	6668.422
	sobol	500	5.6×10^{57}	504.263
		1000	5.9×10^{57}	521.674
		2500	5.8×10^{57}	519.548
Linear	halton	500	2.3×10^{57}	1.1×10^{6}
		1000	2.4×10^{57}	218.010
		2500	2.2×10^{57}	16.883
	hammersley	500	3.8×10^{57}	48.090
		1000	3.4×10^{57}	41.656
		2500	3.7×10^{57}	41.286
	latin_random	500	2.2×10^{57}	17.994
		1000	2.1×10^{57}	7.160
		2500	2.2×10^{57}	6.674
	random	500	1.9×10^{57}	20.066
		1000	2.1×10^{57}	6.886
		2500	2.3×10^{57}	6.696
	sobol	500	6.5×10^{57}	11.714
		1000	5.7×10^{57}	10.706
		2500	6.0×10^{57}	10.360
Ridge	halton	500	2.1×10^{57}	1581.704
		1000	2.2×10^{57}	899.159
		2500	2.2×10^{57}	339.625
	hammersley	500	3.6×10^{57}	1662.425
		1000	3.4×10^{57}	1005.253
		2500	3.4×10^{57}	317.883
	latin_random	500	2.3×10^{57}	3821.199
		1000	2.2×10^{57}	1971.604
		2500	2.1×10^{57}	572.188
	random	500	2.1×10^{57}	4023.874
		1000	2.3×10^{57}	1873.805
		2500	2.3×10^{57}	571.330
	sobol	500	6.3×10^{57}	278.409
		1000	6.0×10^{57}	157.886
		2500	5.6×10^{57}	107.878

Πίνακας Α΄.4: R^2 scores

model	Sequence Algorithms	Npoints	Mean	Median
Lasso	halton	500	-2.486	0.972
		1000	0.667	0.980
		2500	0.606	0.974
	hammersley	500	0.397	0.936
		1000	-0.576	0.934
		2500	0.332	0.938
	latin_random	500	0.572	0.993
		1000	0.622	0.994
		2500	0.630	0.993
	random	500	0.571	0.991
		1000	0.623	0.993
		2500	0.631	0.993
	sobol	500	0.517	0.820
		1000	0.534	0.831
		2500	0.538	0.841
Linear	halton	500	8.3×10^{21}	0.994
		1000	8.0×10^{19}	0.997
		2500	6.3×10^{18}	1.000
	hammersley	500	-10.843	0.994
		1000	-0.996	0.991
		2500	-0.134	0.983
	latin_random	500	0.455	1.000
		1000	0.752	1.000
		2500	0.802	1.000
	random	500	0.438	1.000
		1000	0.760	1.000
		2500	0.800	1.000
	sobol	500	0.778	0.993
		1000	0.794	0.993
		2500	0.806	0.994
Ridge	halton	500	-2.347	0.986
		1000	0.330	0.991
		2500	0.655	0.987
	hammersley	500	0.730	0.963
		1000	0.817	0.969
		2500	0.367	0.975
	latin_random	500	0.709	0.993
		1000	0.749	0.994
		2500	0.789	0.994
	random	500	0.710	0.992
		1000	0.751	0.993
		2500	0.789	0.994
	sobol	500	0.759	0.964
		1000	0.781	0.991
		2500	0.799	0.993

Πίνακας Α΄.5: RMSE scores

model	Sequence Algorithms	Npoints	Mean	Median
Lasso	halton	500	2.9×10^{27}	319.355
		1000	2.9×10^{27}	141.379
		2500	2.8×10^{27}	125.708
	hammersley	500	4.4×10^{27}	82.995
		1000	4.5×10^{27}	68.405
		2500	4.5×10^{27}	61.648
	latin_random	500	2.7×10^{27}	81.503
		1000	2.7×10^{27}	82.223
		2500	2.7×10^{27}	82.584
	random	500	2.8×10^{27}	86.356
		1000	2.7×10^{27}	82.737
		2500	2.8×10^{27}	81.660
	sobol	500	7.3×10^{27}	22.456
		1000	7.5×10^{27}	22.840
		2500	7.4×10^{27}	22.794
Linear	halton	500	2.9×10^{27}	1077.474
		1000	2.9×10^{27}	14.765
		2500	2.8×10^{27}	4.109
	hammersley	500	4.6×10^{27}	6.935
		1000	4.3×10^{27}	6.441
		2500	4.5×10^{27}	6.409
	latin_random	500	2.8×10^{27}	4.242
		1000	2.7×10^{27}	2.676
		2500	2.8×10^{27}	2.583
	random	500	2.6×10^{27}	4.480
		1000	2.7×10^{27}	2.624
		2500	2.8×10^{27}	2.588
	sobol	500	7.8×10^{27}	3.423
		1000	7.3×10^{27}	3.272
		2500	7.5×10^{27}	3.219
Ridge	halton	500	2.7×10^{27}	39.771
		1000	2.8×10^{27}	29.986
		2500	2.8×10^{27}	18.429
	hammersley	500	4.4×10^{27}	40.773
		1000	4.3×10^{27}	31.692
		2500	4.3×10^{27}	17.829
	latin_random	500	2.8×10^{27}	61.816
		1000	2.8×10^{27}	44.403
		2500	2.7×10^{27}	23.920
	random	500	2.7×10^{27}	63.434
		1000	2.8×10^{27}	43.287
		2500	2.8×10^{27}	23.903
	sobol	500	7.7×10^{27}	16.686
		1000	7.5×10^{27}	12.565
		2500	7.3×10^{27}	10.386