ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη μοντέλων μηχανικής μάθησης για τον εντοπισμό καταπόνησης σε γεωργικές καλλιέργειες

Πατεράκης Νικόλαος

Επιβλέπων Καθηγητής: Σαρηγιαννίδης Παναγιώτης,
Αναπληρωτής Καθηγητής Π.Δ.Μ

Μάρτιος, 2022 , Κοζάνη

HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL
& COMPUTER ENGINEERING

Development Of Machine Learning Models For Crops Stress Detection In Smart Farming Domain

MASTER THESIS

Paterakis Nikolaos

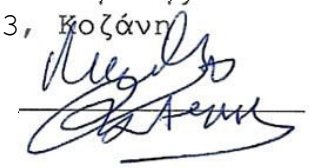
SUPERVISOR: Associate Professor,

Panagiotis Sarigiannidis

March, 2022, Kozani

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

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 "Ανάπτυξη Μοντέλων Μηχανικής Μάθησης για τον εντοπισμό καταπόνησης σε γεωργικές καλλιέργειες" 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 και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Παναγιώτη Σαρηγιαννίδη 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: 

Περίληψη

Με τη πάροδο του χρόνου η έννοια του Δικτύου των Πραγμάτων γίνεται όλο και πιο δημοφιλής, καθώς πλέον η δημιουργία ενός ασύρματου δικτύου αισθητήρων απαιτεί λιγότερο κόστος και χρόνο από ότι χρειαζόταν. Όλο ένα και περισσότερα ασύρματα δίκτυα αισθητήρων κάνουν την εμφάνιση τους στον βιομηχανικό τομέα αλλά και σε τομείς που απασχολούν τον άνθρωπο καθημερινά. Ένας από αυτούς τους τομείς είναι και η Γεωργία.

Η ένταξη των Σύγχρονων Τεχνολογιών Πληροφορίας και Επικοινωνιών στον τομέα της γεωργίας έχει ως αποτέλεσμα την δημιουργία μιας καινούργια έννοιας, της «Έξυπνης» γεωργίας (Smart Farming). Κάποιοι την ονομάζουν και ως Τέταρτη Πράσινη Επανάσταση. Η «έξυπνη» Γεωργία συνδυάζει έννοιες όπως το Δίκτυο των Πραγμάτων τα Μεγάλα Δεδομένα (Big Data) και την Μηχανική Μάθηση (Machine Learning) για την αύξηση της γεωργικής παραγωγής μέσα από μια πιο ακριβής και αποδοτική χρήση των πόρων.

Η παρούσα διπλωματική εργασία, έχει ως σκοπό την μελέτη και ανάπτυξη αλγορίθμων μηχανικής μάθησης, οι οποίοι επεξεργάζονται τα δεδομένα που συλλέχθηκαν από τα τους αισθητήρες και προσπαθούν να προβλέπουν την κατάσταση υγείας του δέντρου που παρακολουθείται. Αξιολογώντας τα αποτελέσματα που συλλέχθηκαν από το πειραματικό στάδιο, παρατηρούνται και σημειώνονται οι ιδιαιτερότητες του κάθε αλγορίθμου και στη συνέχεια με την συνολική σύγκριση όλων των μοντέλων γίνεται η επιλογή των ιδανικότερων. Τα δεδομένα συλλέχθηκαν από μικρά δίκτυα αισθητήρων, τα οποία παρακολουθούσαν δέντρα που ήταν ήδη γνωστή η κατάσταση υγείας τους. Οι αλγόριθμοι αξιολογήθηκαν σύμφωνα με την επίδοση και τον χρόνο που χρειάζονται για να κάνουν μια πρόβλεψη.

Στη συνέχεια γίνεται η ανακεφαλαίωση και η κατάληξη των συμπερασμάτων από όλα τα αποτελέσματα που παρουσιάστηκαν στο πειραματικό στάδιο. Επίσης προτείνονται μελλοντικές επεκτάσεις για την δημιουργία πιο αξιόπιστων και ρεαλιστικών μοντέλων και η έρευνα επιπλέον παραγόντων ώστε τα μοντέλα να έχουν μεγαλύτερη ποικιλία για την καλύτερη μελέτη των δεδομένων. Τέλος προτείνεται η δοκιμή των ίδιων πειραμάτων σε ένα μεγαλύτερο εύρος ποικιλίας δέντρων και περιβαλλόντων.

Λέξεις κλειδιά: Δίκτυο των Πραγμάτων, Δίκτυο ασύρματων αισθητήρων, «Έξυπνη» Γεωργία, Μεγάλα δεδομένα, Μηχανική Μάθηση

Abstract

As time passes by, the “Internet of Things” is becoming more popular. That is due to the fact that, nowadays, the cost and time to create a wireless network of sensors is less demanding. More wireless networks of sensors appear not only in the Industrial field but also in a variety of other fields that constitute part of people’s everyday life. One of these fields, is the Agriculture field.

The inclusion of the Modern Technology of Information and Communication in the Agriculture field results in the creation of a new concept: “Smart” Agriculture. Some people call this new concept the “4th Green Revolution”. “Smart” Agriculture combines concepts such as the Internet of Things, Big Data and Machine Learning to increase agricultural production through a more accurate and efficient use of resources.

The purpose of this Thesis is to study and develop machine learning algorithms, that process the data collected by the sensors and try to predict the health status of the monitored tree. Evaluating the results collected by the experimental stage, the peculiarities of each algorithm are observed and noted, and then following the overall comparison of all models, the most ideal ones are selected. The data was collected by small networks of sensors, that monitored trees whose health status was already known. Algorithms were evaluated according to performance and the time it took to make a prediction.

Then the recapitulation and conclusion of all the results presented in the experimental stage was completed. Finally, future extensions are suggested in order to create more reliable and realistic models, and in order to study additional factors that could make the models more diverse. Last but not least, the testing of the same experiments in a wider

variety of trees and environments is recommended.

Keywords: Internet of Things, Wireless network of sensors, Smart Agriculture, Big Data, Machine Learning

Περιεχόμενα

Λίστα Εικόνων

Πίνακας Πινάκων

Αγγλικά Ακρωνύμια

IoT - Internet of Things

IIoT - Industrial Internet of Things

WSN - Wireless Sensor Network

KNN - K Nearest Neighbors

DT - Decision Tree

LR - Logistic Regression

QDA - Quadratic Discriminant Analysis

SGD - Stochastic Gradient Descent

RFID - Radio Frequency identification

IAB - Internet Architecture Board

M2M - Machine to Machine

M2S – Machine to Server

ALG - Application-Layer-Gateway

SVM - Support Vector Machines

RFID - Radio Frequency Identification

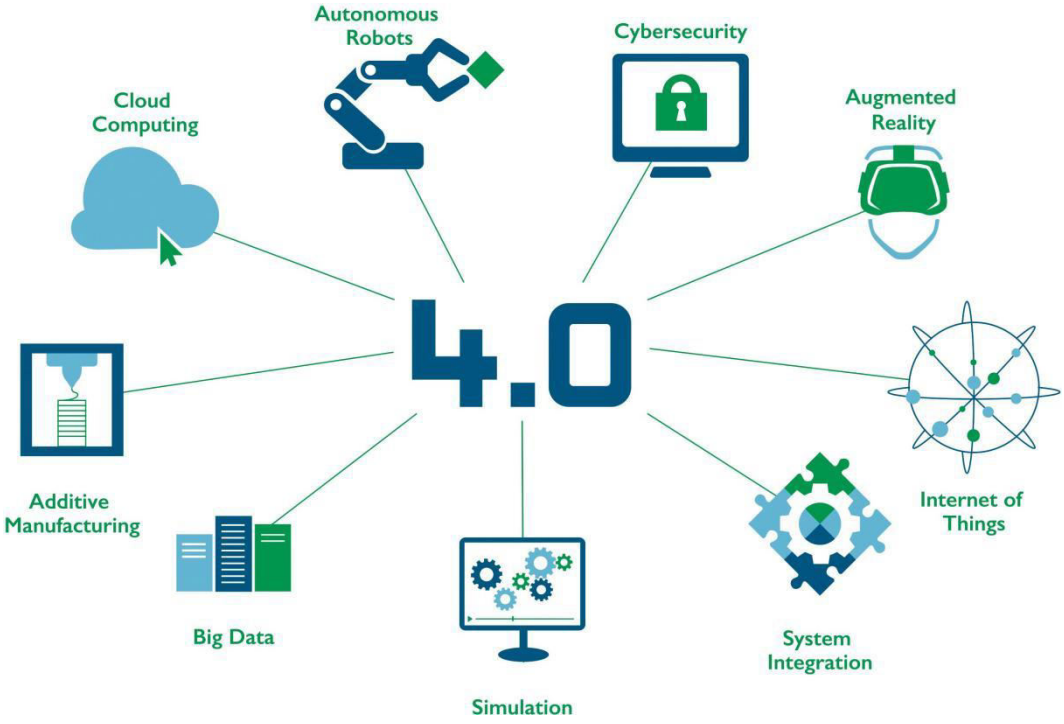
IP - Internet Protocol

Κεφάλαιο 1ο - Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται μια εισαγωγική ιστορική αναδρομή στο διαδίκτυο των πραγμάτων και στα μοντέλα επικοινωνίας που μπορούν να χρησιμοποιούν οι συσκευές σε ένα δίκτυο που περιλαμβάνει το διαδίκτυο των πραγμάτων. Στη συνέχεια θα ακολουθήσει μια σύντομη αναφορά στο πως το διαδίκτυο των πραγμάτων επηρέασε τον τομέα της γεωργίας. Τέλος θα γίνει μια σύντομη αναφορά στον ρόλο που έχει η μηχανική μάθηση στην «έξυπνη» γεωργία.

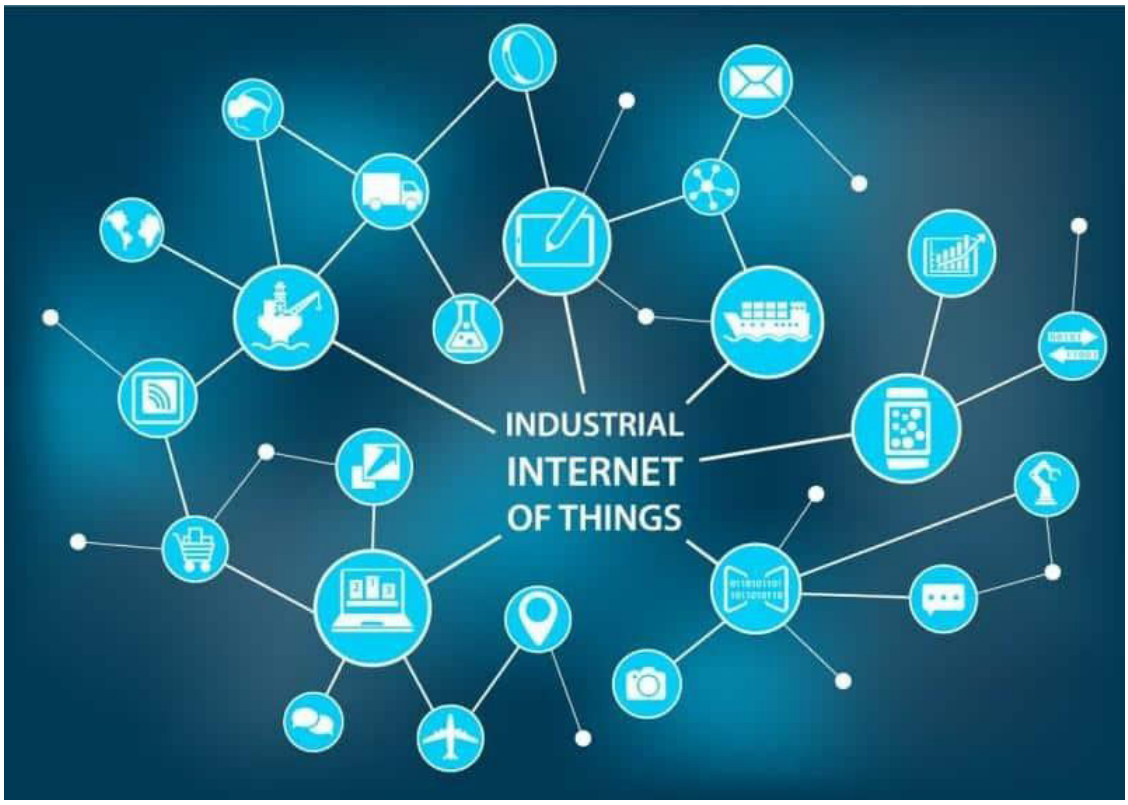
1.1 Το Διαδίκτυο των Πραγμάτων

Ο όρος «Διαδίκτυο» των πραγμάτων (Internet of Things - IoT) [**1**] αναφέρθηκε για πρώτη φορά από τον πρωτοπόρο της Βρετανικής Τεχνολογίας Kevin Ashton το 1999 για να περιγράψει ένα σύστημα από φυσικές οντότητες, οι οποίες θα μπορούσαν να συνδεθούν στο διαδίκτυο. Αρχικά ο Ashton επινόησε τον συγκεκριμένο όρο για να μπορέσει να παρουσιάσει την δύναμη που θα είχε η χρήση της ταυτοποίησης μέσω ραδιοσυχνοτήτων (RFID) στο κομμάτι του ανεφοδιασμού των εταιριών, με σκοπό την καταμέτρηση και τον εντοπισμό των παραγγελιών δίχως να χρειάζεται ανθρώπινη παρέμβαση. Πλέον στον όρο Διαδίκτυο των πραγμάτων έχει απονεμηθεί παραπάνω από ένας ορισμό [2] και παρόλο που η ιδέα του Διαδικτύου των πραγμάτων υπήρχε από τα τέλη του 1970, χρειάστηκαν αρκετά χρόνια για την ανάπτυξη της τεχνολογίας ώστε να μπορεί να υποστηρίξει αυτή την ιδέα. Σήμερα το Διαδίκτυο των πραγμάτων έχει εισχωρήσει σε όλους τους τομείς της ανθρώπινης καθημερινότητας όπως φαίνεται και στην *Εικόνα 1*. Ένα από τα μεγαλύτερα αποτελέσματα που είχε αυτή η ένταξη του Διαδικτύου των πραγμάτων είναι το λεγόμενο «Βιομηχανικό Διαδίκτυο των πραγμάτων» (Industrial Internet of Things - IIoT), ή αλλιώς τέταρτη βιομηχανική επανάσταση (Industry 4.0).



Εικόνα 1: Διαδίκτυο των πραγμάτων 4.0 [3].

Όπως φαίνεται και στην *Εικόνα 2*, οι τομείς που έχουν επηρεαστεί από το βιομηχανικό διαδίκτυο των πραγμάτων είναι πάρα πολλοί αν όχι όλοι. Ένας από αυτούς είναι και ο τομέας της γεωργίας, ο οποίος θα αναφερθεί σε βάθος στα επόμενα κεφάλαια της διπλωματικής.



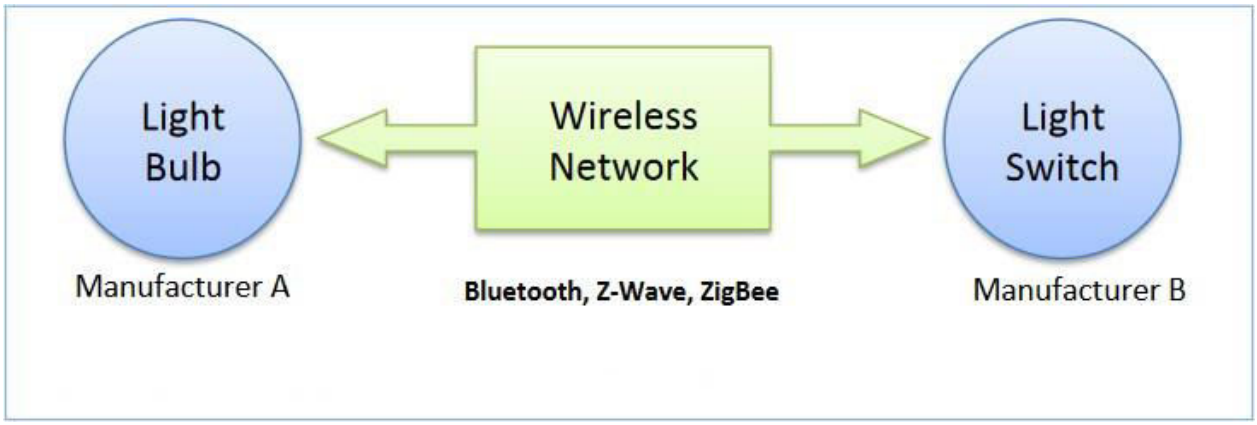
Εικόνα 2: Βιομηχανικό Διαδίκτυο των πραγμάτων [4].

1.2 Μοντέλα επικοινωνίας του Διαδικτύου των Πραγμάτων

Τον Μάρτιο του 2015 το συμβούλιο Αρχιτεκτονικής διαδικτύου (Internet Architecture Board - IAB) δημοσίευσαν ένα αρχιτεκτονικό εγχειρίδιο οδηγιών για την δημιουργία διαδικτύου σε «έξυπνες» συσκευές (RFC 7452), το οποίο παρουσιάζει τη δομή από 4 βασικά μοντέλα [5] που χρησιμοποιούνται από συσκευές του Διαδικτύου των πραγμάτων. Στη συνέχεια αυτής της υπό ενότητας παρουσιάζονται τα 4 βασικά μοντέλα, καθώς αναφέρονται και τα χαρακτηριστικά «κλειδιά» από τη δομή του κάθε μοντέλου.

1.2.1 Συσκευή-με-Συσκευή μοντέλο επικοινωνίας

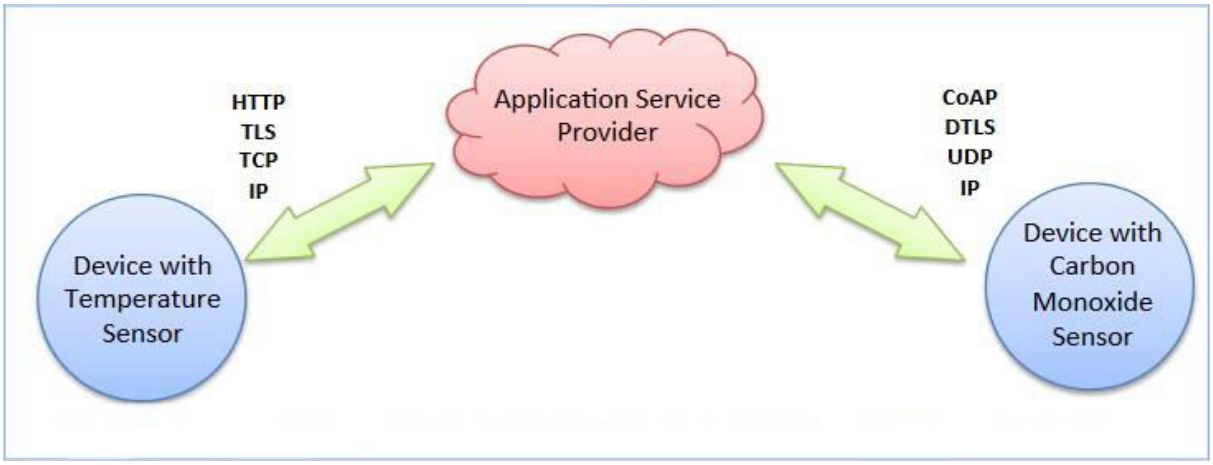
Το μοντέλο επικοινωνίας Συσκευή-με-Συσκευή (Machine to Machine – M2M) αποτελείται από δύο ή περισσότερες συσκευές οι οποίες συνδέονται και επικοινωνούν κατευθείαν η μία με την άλλη χωρίς να χρειάζονται κάποιον μεσολαβητή. Η σύνδεση μεταξύ των συσκευών μπορεί να πραγματοποιηθεί μέσα από πολλά είδη δικτύων, ένα εκ των οποίων είναι το διαδίκτυο και το τοπικό IP δίκτυο. Παρόλα αυτά τα πιο συχνά πρωτόκολλα που χρησιμοποιούν οι συσκευές για την δημιουργία μιας σύνδεσης μεταξύ τους είναι τα πρωτόκολλα Bluetooth, το ZigBee42 ή το Z-Wave, στην *Εικόνα 3* παρουσιάζεται και το σχετικό παράδειγμα.



Εικόνα 3: Μοντέλο επικοινωνίας Συσκευή - με - Συσκευή [6].

1.2.2 Συσκευή-με-Διακομιστή μοντέλο επικοινωνίας

Στο μοντέλο επικοινωνίας Συσκευή-με-Διακομιστή (Machine to Server– M2S) οι συσκευές του Διαδικτύου των πραγμάτων δεν επικοινωνούν απευθείας μεταξύ τους όπως στο προηγούμενο παράδειγμα, αλλά συνδέονται με το ανάλογο υπολογιστικό νέφος μέσω του διαδικτύου, έτσι ώστε να μπορεί ο Διακομιστής να ελέγχει την κίνηση των μηνυμάτων και να παρέχει στις συσκευές τα δεδομένα που επιθυμούν. Το συγκεκριμένο μοντέλο επικοινωνίας εκμεταλλεύεται αρκετά συχνά μηχανισμούς επικοινωνίας που ήδη υπάρχουν, όπως την ασύρματη σύνδεση Wi-Fi ή μια σύνδεση Ethernet ώστε να δημιουργήσει μια σύνδεση μεταξύ της συσκευής με το IP δίκτυο και στην συνέχεια την σύνδεση της συσκευής με τον διακομιστή, όπως φαίνεται και στην *Εικόνα 4*.

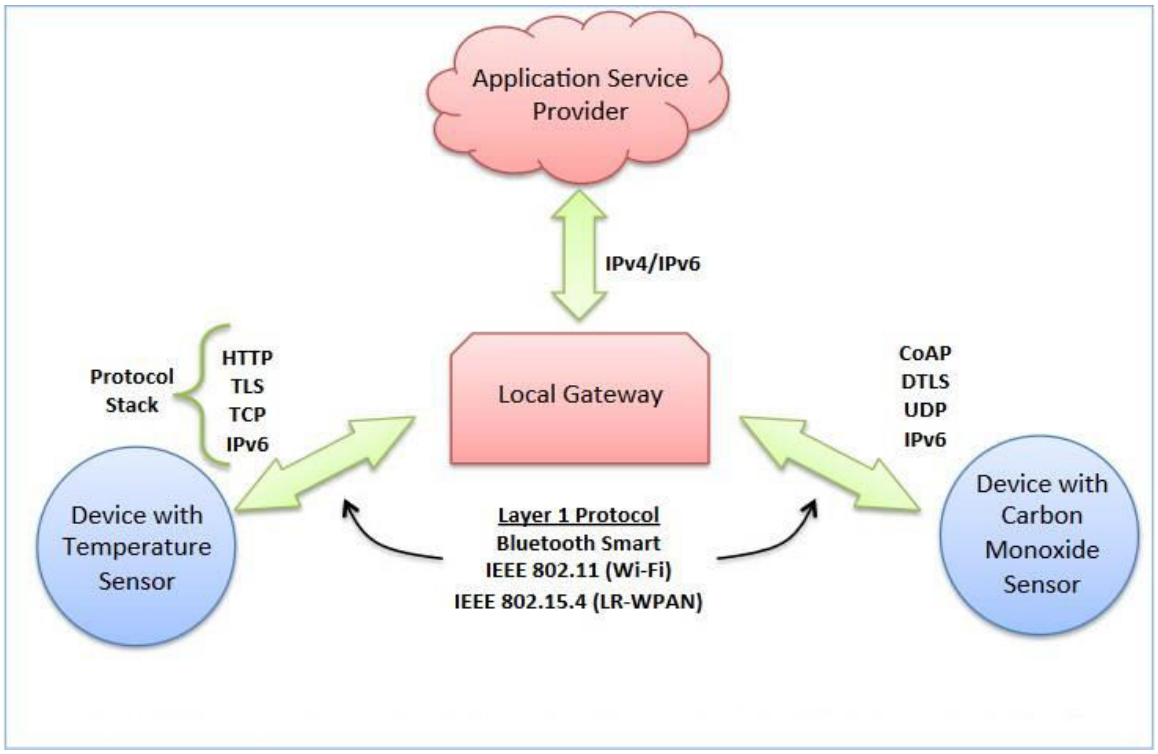


Εικόνα 4: Μοντέλο επικοινωνίας Συσκευή - με - Διακομιστή [6].

1.2.3 Συσκευή-με-Πύλη Δικτύου μοντέλο επικοινωνίας

Στο μοντέλο επικοινωνίας Συσκευή-με-Πύλη Δικτύου, οι συσκευές του Διαδικτύου των πραγμάτων χρησιμοποιούν το λογισμικό Application-Layer-Gateway (ALG) ως αγωγό για την σύνδεση τους με το ανάλογο υπολογιστικό νέφος. Με απλά λόγια, στο συγκεκριμένο μοντέλο επικοινωνίας υπάρχει ο σκοπός της σύνδεσης των συσκευών με το υπολογιστικό νέφος, όπως παρατηρήθηκε και στο προηγούμενο παράδειγμα του μοντέλου επικοινωνίας Συσκευή-με-Διακομιστή, αλλά προτού συνδεθούν με το υπολογιστικό νέφος πρέπει πρώτα να απευθυνθούν στην τοπική πύλη Δικτύου τους. Στη συνέχεια, στη τοπική πύλη Δικτύου του χρήστη έχει τοποθετηθεί και λειτουργεί ένα λογισμικό

για να παρέχει ασφάλεια και ότι άλλο μπορεί να χρειαστεί, όπως την μετάφραση των δεδομένων ή κάποιου πρωτοκόλλου, στην *Εικόνα 5* φαίνεται και το αντίστοιχο παράδειγμα.

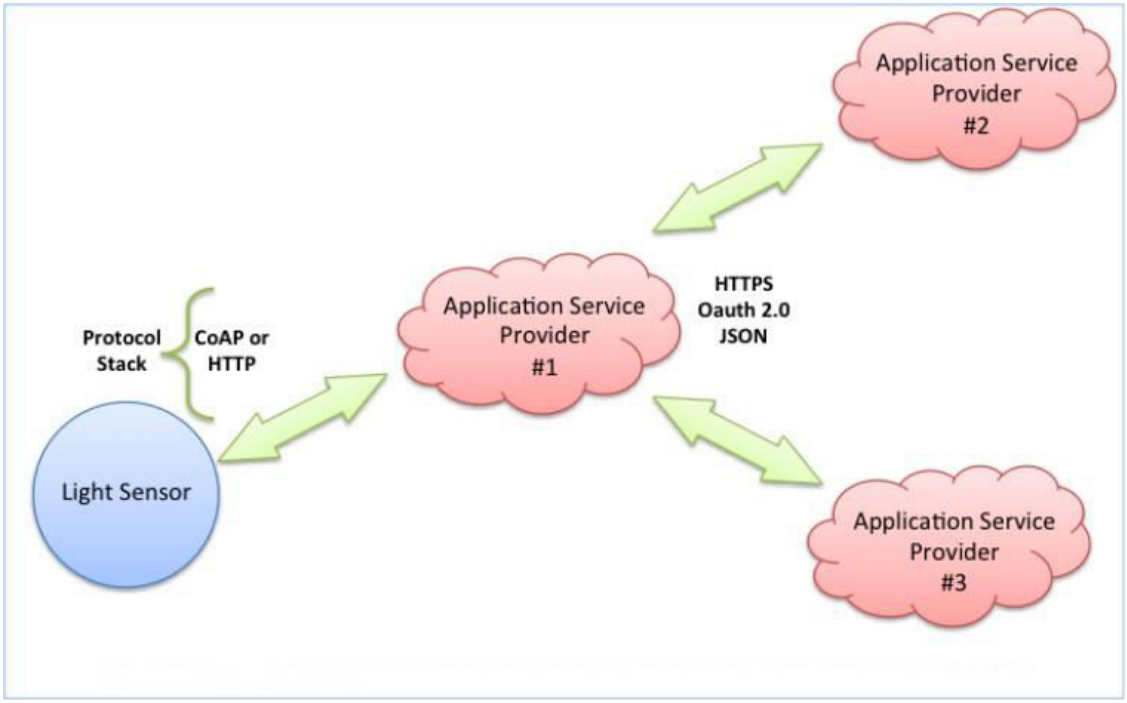


Εικόνα 5: Μοντέλο επικοινωνίας Συσκευή - με - Πύλη Δικτύου [6].

1.2.4 Back-end data-sharing μοντέλο επικοινωνίας

Ουσιαστικά το back-end data-sharing μοντέλο αναφέρεται στην αρχιτεκτονική ενός μοντέλου επικοινωνίας, στο οποίο ο χρήστης θα μπορεί να επιλέγει δεδομένα για να αναλύσει από «έξυπνες» συσκευές, οι οποίες θα υπάρχουν σε κάποιο υπολογιστικό νέφος αποθηκευμένες, συνδυάζοντας τες με δεδομένα και από άλλες πηγές. Το συγκεκριμένο μοντέλο επικοινωνίας πραγματοποιεί την επιθυμία του χρήστη, αλλά και την γενική επιθυμία της πρόσβασης δεδομένων που έχουν αποθηκεύσει τρίτα πρόσωπα. Επίσης χαρακτηρίζεται και ως προέκταση του μοντέλου επικοινωνίας Συσκευή-με-Διακομιστή, το οποίο μπορεί να οδηγήσει σε σιλό δεδομένων όπου όλες οι συσκευές του Διαδικτύου των πραγμάτων «ανεβάζουν» τα δεδομένα τους στο ίδιο υπολογιστικό νέφος. Για παράδειγμα ο υπεύθυνος μιας εταιρείας θα ήθελε να συλλέξει και να αναλύσει την ενέργεια που καταναλώνουν τα δεδομένα που

χρησιμοποιούν όλοι οι αισθητήρες IoT και όλα τα συστήματα που χρησιμοποιούν το διαδίκτυο, σε όλο το κτήριο της εταιρείας. Συνήθως σε ένα μοντέλο επικοινωνίας Συσκευή-με-Διακομιστή τα δεδομένα που συλλέγει ο κάθε IoT αισθητήρες και το κάθε σύστημα, αποθηκεύονται μόνα τους στο δικό τους προσωπικό σιλό δεδομένων. Με την αρχιτεκτονική back-end data sharing η εταιρεία θα μπορεί να έχει εύκολη πρόσβαση και να αναλύει τα δεδομένα από όλες τις συσκευές του κτηρίου μέσα στο υπολογιστικό νέφος. Επίσης η συγκεκριμένη αρχιτεκτονική δίνει λύση στο «φράγμα» των παραδοσιακών σιλό δεδομένων, διότι οι χρήστες έχουν την δυνατότητα να μετακινούν τα δεδομένα τους σε διάφορες IoT συσκευές. Στην *Εικόνα 6* παρουσιάζεται και το σχετικό παράδειγμα.



Εικόνα 6: Back-end data-sharing μοντέλο επικοινωνίας [6].

1.3 Η ένταξη του Διαδικτύου των πραγμάτων στο τομέα της γεωργίας

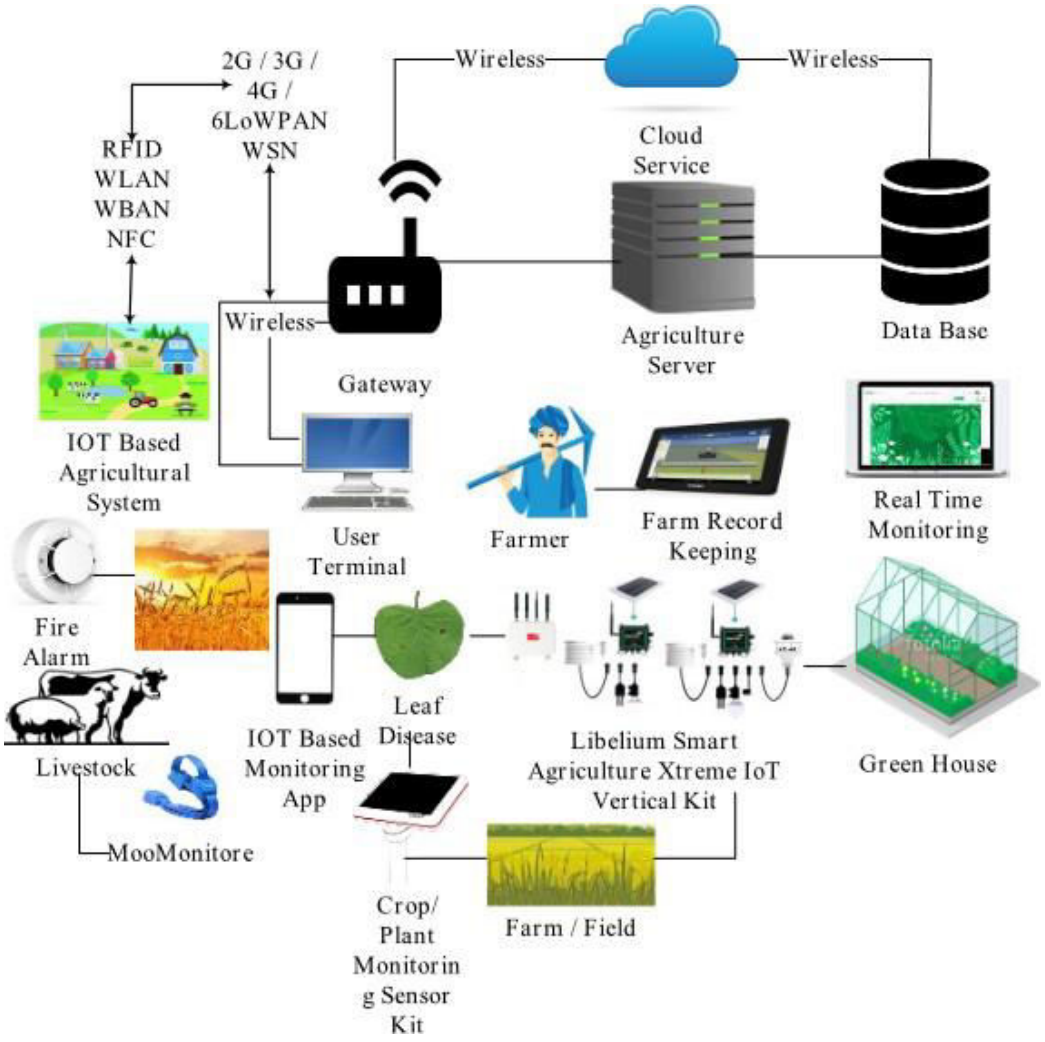
Όπως προαναφέρθηκε νωρίτερα στην αρχή της ενότητας, ένας από τους τομείς που επηρέασε αισθητά το Διαδίκτυο των πραγμάτων είναι και ο τομέας της γεωργίας. Μέχρι το 2050 προβλέπεται οι γεωργικές καλλιέργειες που χρησιμοποιούν την τεχνολογία του Διαδικτύου των πραγμάτων, να έχουν έως

και 70% περισσότερη παραγωγή τροφής και να μπορούν να παρέχουν φαγητό μέχρι και σε 9,6 δισεκατομμύρια ανθρώπους. Επίσης, υπολογίζεται δύο δισεκατομμύρια αισθητήρες να χρησιμοποιούνται σε 525 εκατομμύρια φάρμες [7]. Λόγω αυτής της πρόβλεψης, η χρήση της τεχνολογίας του διαδικτύου των πραγμάτων στην γεωργία γίνεται όλο ένα και περισσότερο γνωστή, με αποτέλεσμα περισσότεροι γεωργοί να στρέφονται προς το Διαδίκτυο των πραγμάτων, ψάχνοντας τρόπους για την βελτίωση των γεωργικών καλλιεργειών τους. Η ένταξη του Διαδικτύου των πραγμάτων στον τομέα της γεωργίας δημιούργησε καινούργιες έννοιες, όπως την έννοια της "έξυπνης γεωργίας" (Smart Farming) και την έννοια της γεωργίας ακριβείας (Precision Agriculture) [8]. Η «έξυπνη» γεωργία αναφέρεται ως η ένταξη διάφορων τεχνολογιών και συσκευών, όπως το Διαδίκτυο ή συσκευές που κάνουν την χρήση του Διαδικτύου των πραγμάτων, στον τομέα της γεωργίας [9], όπως φαίνεται και στην *Εικόνα 7*.

Εικόνα 7: «Έξυπνη» γεωργία [10].

Παρόμοια η γεωργία ακριβείας κάνει χρήση της τεχνολογίας των Διαδικτύων των πραγμάτων και την συνδυάζει με τεχνολογίες όπως τα Μεγάλα Δεδομένα [11] για να παρέχει μια αποδοτική διαχείριση των πηγών [12], όπως φαίνεται στην *Εικόνα 8*. Γενικά και οι δύο έννοιες αποτελούν αποτελέσματα της ένταξης των σύγχρονων τεχνολογιών στον γεωργικό τομέα, με σκοπό την αύξηση παραγωγής, ενώ παράλληλα γίνεται μείωση της γεωργικής έκτασης και των εργατικών χεριών [13]. Παρόλο που η γεωργία ακριβείας μπορεί να δηλωθεί ως μια υπό κατηγορία της έξυπνης γεωργίας, η γεωργία ακριβείας έχει μια ιδιαίτερη

διαφορά. Αναφέρεται πολύ συγκεκριμένα σε IoT τεχνολογίες που έχουν ως στόχο, την βελτίωση αποτελεσματικότητας από δεδομένα και μετρήσεις που χρησιμοποιούνται στη λήψη αποφάσεων. Δηλαδή η γεωργία ακριβείας είναι τόσο ευαίσθητη και ακριβής, όπου η ελάχιστη αλλαγή μετρήσεων μπορεί να επιφέρει διαφορετικό αποτέλεσμα και αυτό με την σειρά του την εκτέλεση μια εντολής που μπορεί να κάνει ζημιά σε σημαντικές πηγές της γεωργικής καλλιέργεια.

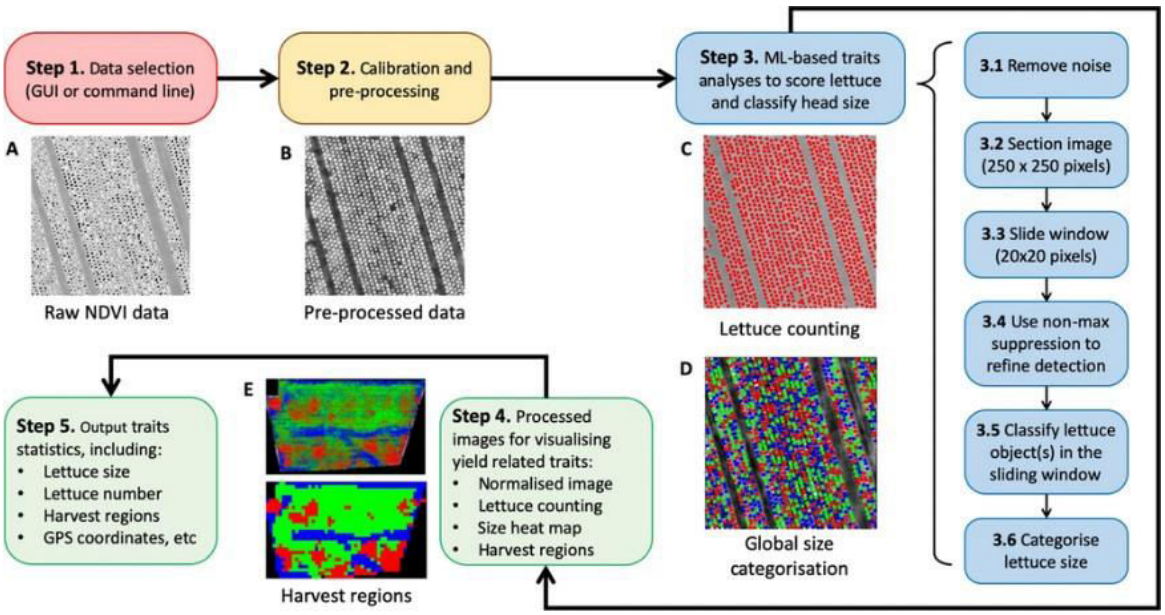


Εικόνα 8: Εφαρμογές του Διαδικτύου των πραγμάτων [14].

1.3.1 Μηχανική μάθηση στον Γεωργικό τομέα

Η τεχνητή νοημοσύνη και το Διαδίκτυο των πραγμάτων [15] μπόρεσαν να αφαιρέσουν τον παράγοντα της τύχης και βοήθησαν τους σύγχρονους αγρότες στη βελτίωση της φάρμας τους [16]. Η γεωργία ακριβείας βασίστηκε ιδιαίτερα στην μηχανική μάθηση, η οποία ανήκει στο γενικό σύνολο της τεχνητής νοημοσύνης. Η μηχανική μάθηση παίζει σημαντικό ρόλο στα συστήματα αποφάσεων που έχουν αναπτυχθεί, διότι εντοπίζει και αναλύει πολύπλοκες

συμπεριφορές που μπορεί να εμφανιστούν στα δεδομένα [17]. Η αύξηση της ακρίβειας των δεδομένων για την παραγωγή περισσότερης και καλύτερης σοδειάς πλέον έχει γίνει απαραίτητη [18]. Στην *Εικόνα 9* παρουσιάζεται ένα παράδειγμα ανάλυσης εικόνων από αλγόριθμο μηχανικής μάθησης για την αξιολόγηση ποιότητας ενός λάχανου.



Εικόνα 9: Βήματα αλγορίθμου αξιολόγησης ενός λάχανου [19].

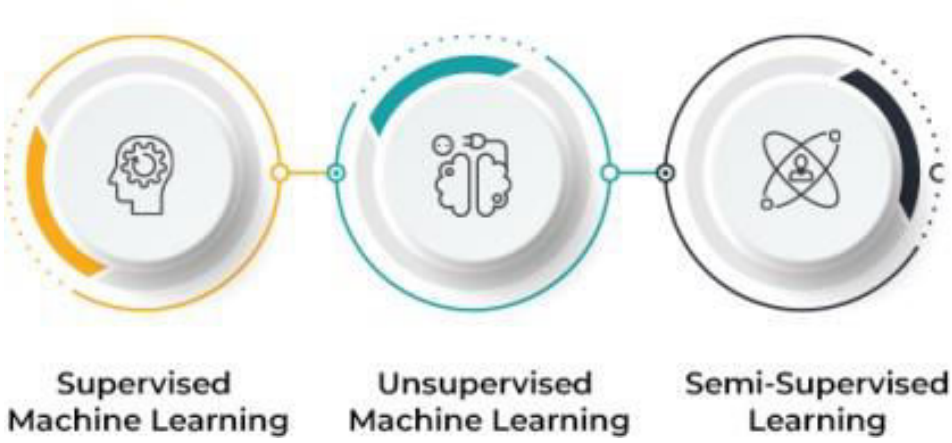
Κεφάλαιο 2 - Ανάλυση Αλγορίθμων Μηχανικής Μάθησης

Το δεύτερο κεφάλαιο περιγράφει την ένταξη της μηχανικής μάθησης στην ανάπτυξη των αλγορίθμων για την μελέτη και επίλυση προβλημάτων που προκύπτουν στην καθημερινότητα των ανθρώπων. Στη συνέχεια ακολουθεί μια συνοπτική αναφορά και επεξήγηση των αλγορίθμων επιτηρούμενης μάθησης (supervised learning). Τέλος αναφέρονται έξη γενικές κατηγορίες αλγορίθμων επιτηρούμενης μάθησης οι οποίες επιλέχθηκαν για την μελέτη και ανάπτυξη των μοντέλων για το πειραματικό στάδιο.

2.1 Αλγόριθμοι μηχανικής μάθησης

Οι αλγόριθμοι μηχανικής μάθησης είναι αλγόριθμοι που έχουν ως στόχο να δώσουν την δυνατότητα εκμάθησης σε έναν υπολογιστή. Λέγοντας ότι οι αλγόριθμοι δίνουν την δυνατότητα εκμάθησης σε έναν υπολογιστή, δεν σημαίνει απαραίτητα την δημιουργία κάποιας συνείδησης, αλλά την διαδικασία εντοπισμού μοτίβων στα δεδομένα ανάλυσης [20]. Γενικότερα η μηχανική μάθηση ανήκει στον ευρύτερο όρο της τεχνητής νοημοσύνης [21] και σκοπός της είναι η μελέτη, βελτιστοποίηση και χρήση μαθηματικών μοντέλων, τα οποία έχουν την δυνατότητα να εκπαιδευτούν μέσα από δεδομένα για την λήψη βέλτιστων μελλοντικών αποφάσεων [22]. Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται σε πολλές περιστάσεις, όπως την ανάλυση εικόνων, την ανάλυση δεδομένων και την πρόβλεψη καιρικών συνθηκών [23]. Ένα κύριο πλεονέκτημα των αλγόριθμων μηχανικής μάθησης είναι η δυνατότητα τους να λειτουργούν αυτόματα, από την στιγμή που αντιλαμβάνονται πως να διαχειρίζονται τα δεδομένα που τους δίνονται [24]. Οι τεχνικές επίλυσης μηχανικής μάθησης χρησιμοποιούνται για την επίλυση ενός τεράστιου εύρους προβλημάτων [25]. Οι τρεις κύριες κατηγορίες προβλημάτων που αντιμετωπίζουν οι αλγόριθμοι μηχανικής μάθησης είναι, προβλήματα

ταξινόμησης (classification), τα προβλήματα ομαδοποίησης (clustering) και τα προβλήματα παλινδρόμησης (regression). Ενώ οι τρεις κατηγορίες αλγόριθμων μηχανικής μάθησης είναι, οι αλγόριθμοι επιτηρούμενης μάθησης, μη επιτηρούμενης μάθησης (unsupervised learning) και οι ημι-επιτηρούμενη μάθηση (semi-supervised learning) [26] (*Εικόνα 10*). Στην παρούσα διπλωματική εργασία οι αλγόριθμοι επιτηρούμενης μάθησης χρησιμοποιήθηκαν για την περάτωση των πειραμάτων.



Εικόνα 10: Οι τρεις κατηγορίες αλγορίθμων μηχανικής μάθησης [27].

2.2 Αλγόριθμοι επιτηρούμενης μάθησης

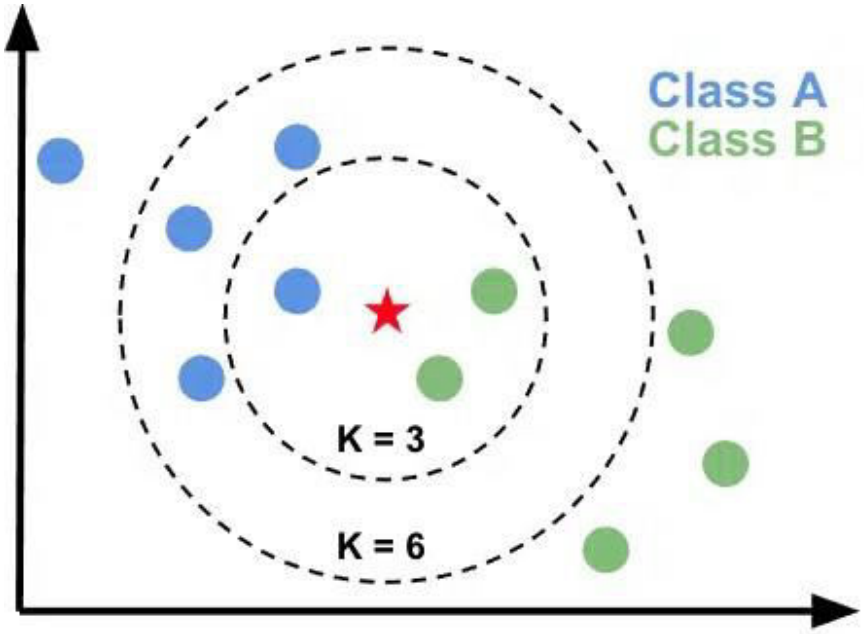
Εάν τα δεδομένα που δίνονται στον αλγόριθμο είναι ήδη κατηγοριοποιημένα, τότε η διαδικασία της μάθησης είναι επιτηρούμενη [28]. Ο σκοπός των αλγόριθμων μηχανικής μάθησης είναι να βρουν κάποιο συμπέρασμα από δεδομένα που είναι κατηγοριοποιημένα [29]. Οι διάφοροι αλγόριθμοι επιτηρούμενης μάθησης [30] καταγράφουν διαδρομές που δημιουργούνται ανάμεσα στα δεδομένα εισόδου εκ- παίδευσης και στα αποτελέσματα εξόδου εκπαίδευσης και επιλέγουν την καταλληλότερη διαδρομή [31]. Στη συνέχεια παίρνουν τη διαδρομή που επέλεξαν και την εισάγουν σε μία συνάρτηση που δημιουργούν. Αφού δημιουργηθεί η συνάρτηση κατά τη φάση εκπαίδευσης, τότε περνάμε στην φάση της δοκιμής, όπου εκεί δοκιμάζεται η συνάρτηση που δημιούργησε ο αλγόριθμος. Τέλος γίνεται αξιολόγηση των αποτελεσμάτων της συνάρτησης.

2.2.1 Αλγόριθμος Κ-πλησιέστερων γειτόνων

Ο αλγόριθμος Κ-πλησιέστερων γειτόνων (KNN) είναι από τους πιο διαδεδομένους μηχανισμούς εξόρυξης δεδομένων σε προβλήματα ταξινόμησης και στον εντοπισμό μοτίβων [32]. Πολλοί τον αποκαλούν και ως "τεμπέλη", διότι κατά τη φάση εκπαίδευσης δεδομένων, δεν γίνεται πραγματικά εκπαίδευση δεδομένων, αλλά αποθήκευση των δεδομένων που του έχουν δοθεί. Παρόλο που ο αλγόριθμος είναι εύκολος στη χρήση και πολύ αποτελεσματικός στις περισσότερες περιπτώσεις, η επίδοση του εξαρτάται ιδιαίτερα από την κατανομή των δεδομένων εκπαίδευσης. Η τιμή που θα δοθεί στην παράμετρο k του αλγορίθμου, παίζει πολύ σημαντικό ρόλο [33]. Η αυξομείωση της τιμής k επηρεάζει την ακρίβεια του αλγορίθμου, είτε θετικά, είτε αρνητικά σε κάθε πρόβλημα. Από την άλλη πλευρά όταν υπάρχει αύξηση της ακρίβειας του αλγορίθμου, παρατηρείται μείωση της ασάφειας του. Η μεταβλητή k αντιπροσωπεύει τον αριθμό των πιο κοντινών δεδομένων εκπαίδευσης σε ένα αποτέλεσμα δοκιμής, εάν η μεταβλητή k ισούται με 1, τότε η αναζήτηση γίνεται στον πιο κοντινό γείτονα [34]. Ο αλγόριθμος ακολουθεί την ιδέα της εγγύτητας που είναι βασισμένη στον μαθηματικό τύπο της απόστασης του Ευκλείδη, η οποία υπολογίζει την απόσταση μεταξύ δύο ή περισσότερων σημείων του επιπέδου [35].

$$\mathbf{d(i,j)} = \sqrt{|\mathbf{X_{i1} - x_{j1}}|^2 + |\mathbf{X_{i2} - x_{j2}}|^2 + |\mathbf{X_{i3} - x_{j3}}|^2 + \ldots + |\mathbf{X_{ip} - x_{jp}}|^2} \qquad (2.1)$$

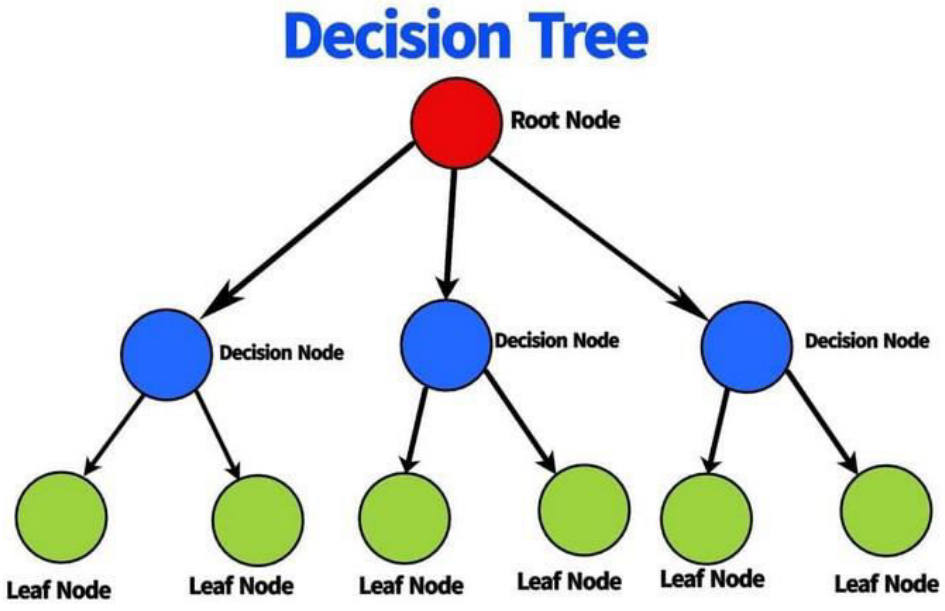
Τα δεδομένα που του δίνονται είναι της μορφής (x,y). Στόχος του KNN αλγορίθμου είναι να δημιουργήσει μια συνάρτηση h(x) στην οποία θα ισχύει $\mathbf{x} \rightarrow \mathbf{y}$. Η απλή δομή του αλγορίθμου αποδεικνύει ότι δεν χρειάζεται η κατασκευή κάποιου μαθηματικού μοντέλου ή η παραμετροποίηση πολλών μεταβλητών για την εύρεση του σωστού αποτελέσματος. Στην *Εικόνα 11* απεικονίζεται ένα απλό παράδειγμα δύο κλάσεων και δύο διαδικασιών αναζήτησης, η μία με τιμή k = 3 και η δεύτερη με τιμή k = 6.



Εικόνα 11: Αποτελέσματα αλγορίθμου KNN [36]

2.2.2 Decision Trees Αλγόριθμοι

Ένα δέντρο απόφασης (Decision Tree), είναι ένα διάγραμμα ροής με την δομή ενός δέντρου. Ο πρώτος κόμβος που συναντάει κάποιος στην κορυφή του δέντρου είναι ο "ριζικός" κόμβος, στην συνέχεια οι εσωτερικοί κόμβοι αντιπροσωπεύουν την δοκιμή μιας από τις μεταβλητές του προβλήματος και οι εξωτερικοί κόμβοι ή αλλιώς οι λεγόμενοι κόμβοι "φύλλα" οι οποία αντιπροσωπεύουν μία κλάση. Στην *Εικόνα 12* παρουσιάζεται η δομή ενός decision tree αλγόριθμου [37].

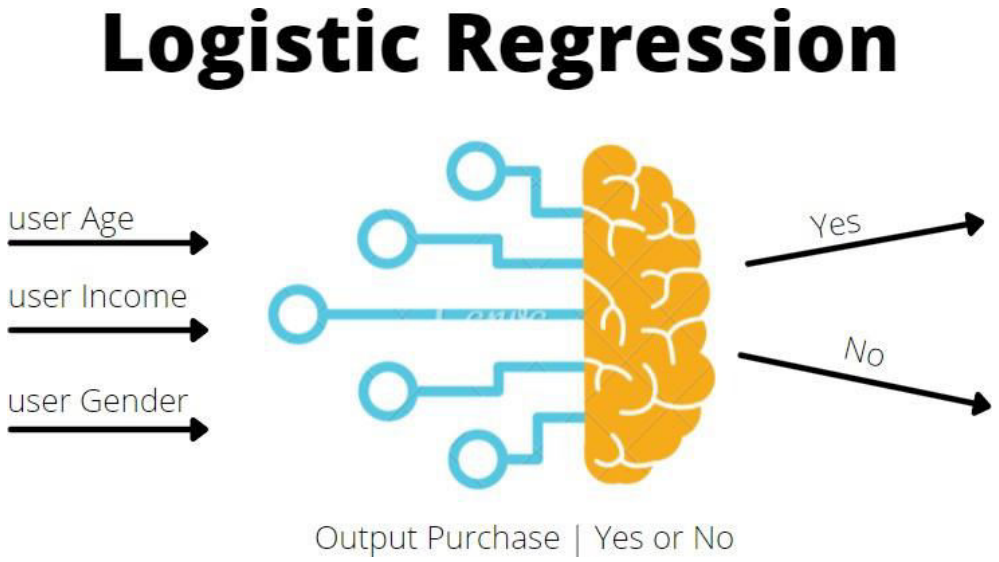


Εικόνα 12: Δέντρο Απόφασης [38].

Τα δέντρα απόφασης είναι από τις πιο ισχυρές και αποτελεσματικές μεθόδους και χρησιμοποιούνται συχνά στον τομέα της μηχανικής μάθησης για αναλύσεις εικόνων, εντοπισμό μοτίβων [39] και ανάλυση δεδομένων [40]. Λόγω της πολύ εύκολης δομής τους και της μεγάλης ακρίβειας που έχουν σε δεδομένα ποικίλων μορφών, τα δέντρα απόφασης έχουν χρησιμοποιηθεί σε πολλούς τομείς. Τα δέντρα απόφασης κατασκευάζονται κατά τη διάρκεια της φάσης εκπαίδευσης, δημιουργούνται αναδρομικά [41] εφαρμόζοντας σε κάθε εσωτερικό κόμβο ερωτήματα στα οποία ανάλογα την απάντηση που θα δεχθούν σε κατευθύνουν στον ανάλογο επόμενο εσωτερικό κόμβο ή στον επόμενο "φύλλο" κόμβο. Τα περισσότερα δέντρα κατασκευάζονται από πάνω προς τα κάτω, δηλαδή από τον "ριζικό" κόμβο προς τους κόμβους "φύλλα". Τέλος υπάρχουν περιπτώσεις όπου κάποια δέντρα αποφάσεων εκτός από την φάση της δημιουργίας και της ταξινόμησης, περιέχουν και μια τρίτη φάση, την φάση του "κλαδέματος" [42] στην οποία αφαιρούνται κόμβοι οι οποίο θεωρούνται αχρείαστοι για να αυξηθεί η συνολική επίδοση του δέντρου απόφασης.

2.2.3 Linear Models

Στην κατηγορία των γραμμικών μοντέλων [43] ανήκει ένα μεγάλο εύρος αλγορίθμων, όπου ο καθένας από αυτούς τους ακολουθεί την δικιά του λογική για την πρόβλεψη των αποτελεσμάτων [44]. Για αυτό το λόγο στην συγκεκριμένη ενότητα θα αναφερθεί η μεθοδολογία που χρησιμοποιεί ο αλγόριθμος Logistic Regression (LR), διότι είναι εκείνος που χρησιμοποιήθηκε στην πειραματική φάση. Ο LR αλγόριθμος χρησιμοποιεί μια πολύ γνωστή τεχνική που ονομάζεται μέθοδος ελαχίστων τετραγώνων. Αυτό που κάνει όμως τον συγκεκριμένο αλγόριθμο να διαφέρει από τους υπόλοιπους, δεν είναι η χρήση της μεθόδου των ελαχίστων τετραγώνων, αλλά την ιδιότητα του αλγορίθμου να απαντάει μόνο με αν η πρόβλεψη είναι αληθής ή ψευδής (*Εικόνα 13*).



Εικόνα 13: Πιθανές προβλέψεις του LR αλγορίθμου[45].

2.2.4 Linear and Quadratic Discriminant Analysis

Ο αλγόριθμος Linear Discriminant Analysis (LDA) και ο Quadratic Discriminant analysis (QDA) αλγόριθμος βασίζονται στην τεχνική της διακριτικής ανάλυσης (Discriminant Analysis) [46], για την δημιουργία μιας εξίσωσης δύο ή περισσότερων διαφορετικών ομάδων δεδομένων. Οι εξισώσεις και των δύο αλγορίθμων προκύπτουν από απλά μοντέλα πιθανοτήτων, τα οποία με την χρήση της θεωρίας πιθανοτήτων αναζητούν την κλάση των άγνωστων δεδομένων [47]. Ακολουθώντας τον τύπο, για κάθε κλάση k και χρησιμοποιώντας τους Bayesian κανόνες, οι δύο αλγόριθμοι καταλήγουν στις εξής εξισώσεις:

Για τον QDA :

$$\mathbf{logP(y = k|x) = -\frac{1}{}log|\sum_k| - \frac{1}{}(x - \mu_k)^t\sum_k^{-1}(x - \mu_k) + logP(y = k) + C_{st}} \quad (2.2)$$

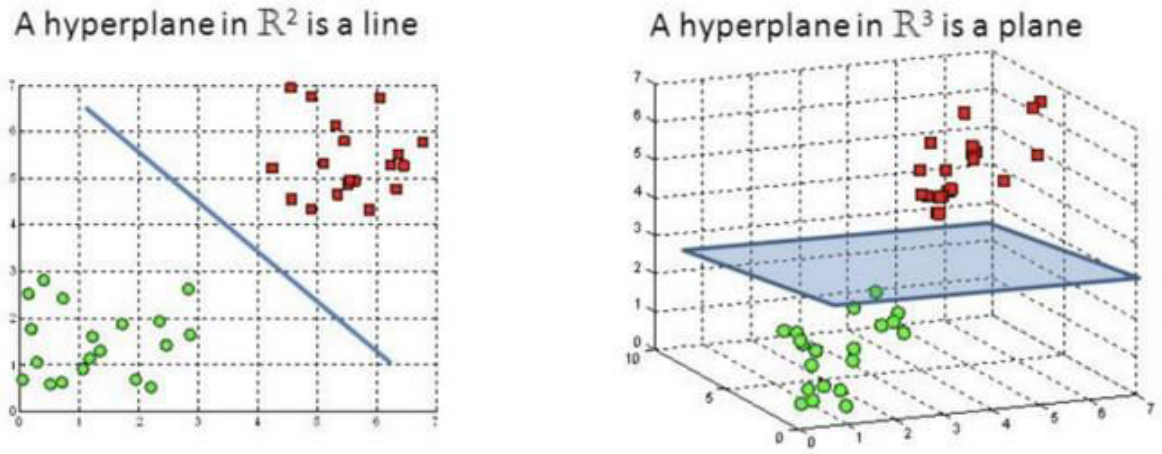
Για τον LDA :

$$\mathbf{logP(y = k|x) = -\frac{1}{}(x - \mu_k)^t\sum_k^{-1}(x - \mu_k) + logP(y = k) + C_{st}} \quad (2.3)$$

Η διαφορά των δύο αλγορίθμων είναι, ότι ο αλγόριθμος LDA θεωρεί πως ο πίνακας διακυμάνσεων, δηλαδή το είναι ίδιο για όλες τις k κλάσεις. Αυτή η θεώρηση όμως δεν δίνει πάντοτε το μεγαλύτερο ποσοστό ευστοχίας σε όλα τα προβλήματα. Για αυτό το λόγο ο QDA ακολουθεί μια πιο χαλαρή προσέγγιση και θεωρεί πως κάθε κλάση έχει το δικό της πίνακα διακυμάνσεων.

2.2.5 Support Vector Machines

Οι αλγόριθμοι Support Vector Machines (SVM) έκαναν την εμφάνιση τους το 1992 από τον ρωσικής προέλευσης επιστήμονα Vladimir Vapnik [48]. Οι αλγόριθμοι SVM χρησιμοποιούν την συνάρτηση kernel η οποία μπορεί και υπολογίζει συστηματικά τα διανύσματα υποστήριξης ταξινόμησης και σε περισσότερες διαστάσεις. Δηλαδή μπορεί και υπολογίζει τις σχέσεις μεταξύ των δεδομένων αν βρισκόντουσαν σε περισσότερες διαστάσεις μέσα από μια τεχνική που ονομάζεται kernel trick [49]. Οι αλγόριθμοι SVM βρίσκουν το ιδανικό υπερ-επίπεδο σε ένα χώρο πολλών διαστάσεων χρησιμοποιώντας μια μη γραμμική kernel συνάρτηση για να μεγαλώσουν τα όρια μεταξύ των κλάσεων [50]. Στην *Εικόνα 14* παρουσιάζεται η εφαρμογή ενός υπερ-επιπέδου σε δυο και σε τρεις διαστάσεις.

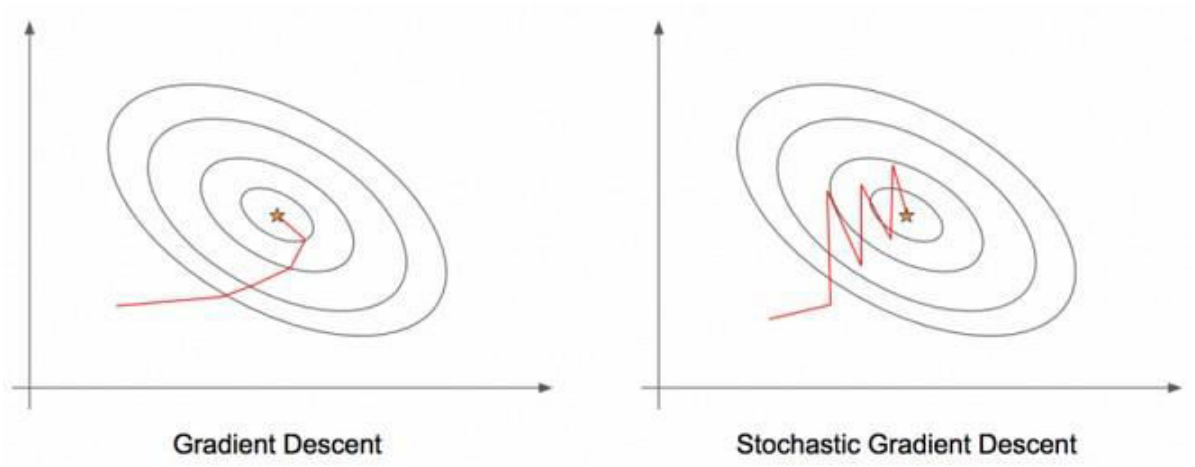


Εικόνα 14: Εφαρμογή υπέρ επιπέδου σε δύο και τρείς διαστάσεις [51].

Τα περισσότερα ρεαλιστικά προβλήματα περιλαμβάνουν μη διαχωρισμένες μεταβλητές στα οποία δεν υπάρχει κάποιο υπερ-επίπεδο [52] που να μπορεί να δια- χωρίζει τα θετικά με τα αρνητικά. Μία ωφέλιμη στρατηγική επίλυση για αυτό το είδος προβλημάτων είναι η χαρτογράφηση των δεδομένων σε περισσότερες διαστάσεις και η εύρεση ενός υπερ-επιπέδου. Το μεγαλύτερο πλεονέκτημα των αλγόριθμων SVM είναι ότι το στάδιο εκπαίδευσης είναι πάρα πολύ εύκολο, αλλά έχουν και άλλα πλεονεκτήματα, όπως πάρα πολύ μεγάλη ακρίβεια, πολύ καλή επίδοση σε δεδομένα που δεν έχουν ξανά δει αλλά και ο εύκολος εντοπισμός της μη γραμμικής σχέσης των δεδομένων.

2.2.6 Stochastic Gradient Descent

Ο stochastic gradient descent αλγόριθμος είναι μια αναβάθμιση του gradient descent αλγόριθμου. Με τον όρο στοχαστικό [53] που του προσθέτουμε, εννοούμε μία μέθοδο ή μια τεχνική που οδηγεί σε ένα τυχαίο αποτέλεσμα. Με αυτό τον τρόπο δίνετε η δυνατότητα της επιλογής τυχαίων παραδειγμάτων αντί να ληφθούν υπόψιν όλα τα δεδομένα, κάτι που οδηγεί στην αύξηση της ταχύτητας υπολογισμού και στην μείωση των λαθών αφού η προσέγγιση γίνεται πλέον για ένα μέρος των δεδομένων και όχι του συνόλου. Στην *Εικόνα 15* φαίνεται η σύγκριση μεταξύ των δυο αλγορίθμων [54] στην οποία παρατηρήθηκε ότι, παρόλο που ο stochastic gradient descent αλγόριθμος προσθέτει περισσότερο θόρυβο, λόγω της μικρής επιλογής δεδομένων, καταλήγει πιο γρήγορα στο ελάχιστο από τον gradient descent αλγόριθμο.



Εικόνα 15: Σύγκριση αποτελεσμάτων του SGD αλγορίθμου με τον GD αλγόριθμο [54].

Κεφάλαιο 3 - Πειραματικό στάδιο

Σε αυτό το κεφάλαιο γίνεται η παρουσίαση των δεδομένων που θα χρησιμοποιηθούν για την εκπαίδευση και δοκιμή των αλγορίθμων και η επεξήγηση των τριών ειδών δεδομένων που συλλέχθηκαν. Στη συνέχεια παρουσιάζεται συνοπτικά τα μηχανήματα και οι αισθητήρες που χρησιμοποιήθηκαν για την συλλογή των δεδομένων.

3.1 Ανάλυση δεδομένων εκπαίδευσης και δοκιμής

Τα δεδομένα που χρησιμοποιήθηκαν σε αυτή τη διπλωματική εργασία, είναι δεδομένα που συλλέχθηκαν από τους αισθητήρες υγρασίας και θερμοκρασίας των μηχανημάτων Libelium. Τα δεδομένα έχουν συλλεχθεί και αποθηκευτεί σε ένα φύλλο excel με όνομα R_filter το οποίο παρουσιάζεται και στους πίνακες (*Πίνακας 1*) και (*Πίνακας 2*) που ακολουθούν.

1	SM 50	SM 20	TEMP	TARGET
2	13.60816	16.54619	10.62	healthy
3	13.9522	16.72359	10.63	healthy
4	14.03788	16.81119	10.63	healthy
5	13.78103	16.72359	10.63	healthy
6	13.9522	16.63604	10.63	healthy
7	13.9522	16.54856	10.63	healthy
8	13.69748	16.55093	10.64	healthy
9	13.78299	16.63842	10.64	healthy
10	13.86856	16.55093	10.64	healthy
11	13.86856	16.55093	10.64	healthy
12	13.78299	16.55093	10.64	healthy
13	13.78299	16.46349	10.64	healthy

14	13.69942	16.37846	10.65	healthy
15	13.70137	16.3808	10.66	healthy
16	13.7869	16.3808	10.66	healthy
17	13.61782	16.47056	10.67	healthy
18	13.61782	16.38315	10.67	healthy
19	13.61976	16.29812	10.68	healthy
20	13.53431	16.21082	10.68	healthy

Πίνακας 1: Excel με δεδομένα εκπαίδευσης και δοκιμής 1

22327	43.18696	50.47257	3.48	unhealthy
22328	43.24408	50.58918	3.47	unhealthy
22329	43.3139	50.65508	3.47	unhealthy
22330	43.24408	50.721	3.47	unhealthy
22331	43.38376	50.721	3.47	unhealthy
22332	43.37102	50.77175	3.46	unhealthy
22333	43.37102	50.77175	3.46	unhealthy
22334	43.35828	50.82248	3.45	unhealthy
22335	43.35828	50.75654	3.45	unhealthy
22336	43.35828	50.69063	3.45	unhealthy
22337	43.34554	50.60959	3.44	unhealthy
22338	43.34554	50.60959	3.44	unhealthy
22339	43.26304	50.59443	3.43	unhealthy
22340	43.26304	50.66026	3.43	unhealthy
22341	43.3328	50.66026	3.43	unhealthy
22342	43.26304	50.66026	3.43	unhealthy
22343	43.3328	50.66026	3.43	unhealthy
22344	43.32006	50.64508	3.42	unhealthy
22345	43.32006	50.64508	3.42	unhealthy
22346	43.30732	50.6299	3.41	unhealthy

Πίνακας 2: Excel με δεδομένα εκπαίδευσης και δοκιμής 2

Ο συνολικός αριθμός δεδομένων είναι 22.346. Η πρώτη κατηγορία είναι οι μετρήσεις από τον αισθητήρα των πενήντα εκατοστών, η δεύτερη είναι οι μετρήσεις από τον αισθητήρα των είκοσι εκατοστών, η τρίτη κατηγορία είναι οι μετρήσεις από τους αισθητήρες θερμοκρασίας και τέλος στην τέταρτη κατηγορία έχουμε τον τύπο του δέντρου. Δηλαδή αν είναι άρρωστο ή υγιές. Στη συνέχεια από τα δεδομένα που συλλέχθηκαν και χρησιμοποιήθηκαν για την εκπαίδευση και δοκιμή των αλγορίθμων, δημιουργήθηκαν δύο επιπλέον excel. Το πρώτο excel που δημιουργήθηκε ονομάζεται R_filter_balanced και περιέχει ίσα δεδομένα μη υγιή δέντρων με τα δεδομένα των υγιή δέντρων, σκοπός αυτής της δημιουργίας, είναι η δοκιμή των επιδόσεων των αλγορίθμων όταν υπάρχουν ακριβώς οι ίδιοι αριθμοί δεδομένων και από τις δύο κλάσεις. Το δεύτερο excel που δημιουργήθηκε ονομάζεται R_filter_without_temperature και σκοπός είναι η δοκιμή των επιδόσεων των αλγορίθμων χωρίς να ληφθεί υπόψιν οι μετρήσεις των αισθητήρων θερμοκρασίας.

3.2 Σύντομη Παρουσίαση του Μηχανήματος Libelium

Οι συσκευές που χρησιμοποιήθηκαν για την μέτρηση και συλλογή των πληροφοριών, ονομάζονται Smart Agriculture PRO και είναι κατασκευασμένες από την εταιρεία libelium. Η εταιρεία σχεδιάζει και κατασκευάζει ασύρματες συσκευές δικτύων αισθητήρων. Το μοντέλο Smart Agriculture PRO (*Εικόνα 16*) είναι ειδικά κατασκευασμένο για την παρακολούθηση δεδομένων όπως την θερμοκρασία και την υγρασία σε ένα χωράφι.

Εικόνα 16: Μηχάνημα Libelium [55].

3.2.1 Παρουσίαση Αισθητήρων

Οι δύο αισθητήρες υγρασίας που χρησιμοποιήθηκαν για την μέτρηση των δεδομένων υγρασίας, είναι οι watermark 200SS. Ο αισθητήρας watermark 200SS (*Εικόνα 17*) είναι σχεδιασμένος να μετράει την τάση του νερού στο έδαφος, όσο μεγαλύτερη είναι η ποσότητα του νερού που υπάρχει μέσα στο χώμα του

εδάφους, τόσο μικρότερη είναι η ηλεκτρική αντίσταση που καταγράφει ο αισθητήρας, παρόμοια αν το χώμα του εδάφους είναι ξηρό, η ηλεκτρική αντίσταση που θα καταγράψει ο αισθητήρας θα έχει πολύ μεγαλύτερες τιμές.

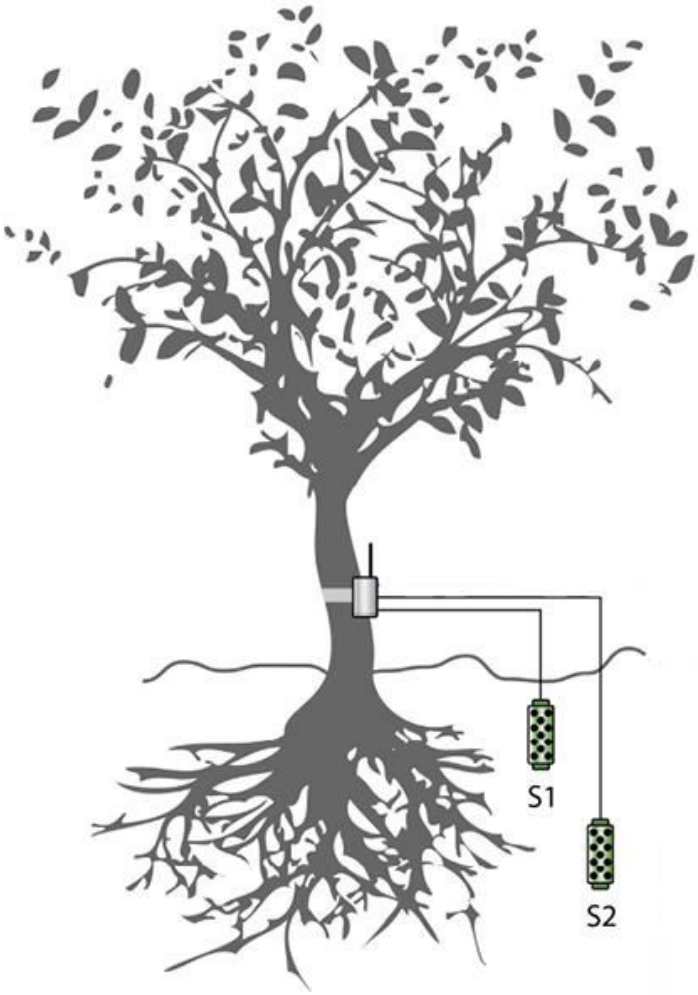
Εικόνα 17: Αισθητήρας watermark 200SS.

Ο αισθητήρας θερμοκρασίας που χρησιμοποιήθηκε για την μέτρηση των δεδομένων θερμοκρασίας είναι ο Pt-1000 (*Εικόνα 18*). Το Pt στο όνομα του αισθητήρα, υποδηλώνει ότι ο αισθητήρας είναι φτιαγμένος από λευκόχρυσο (Platinum - Pt), ενώ το 1000 αναφέρεται στο ότι ο αισθητήρας σε θερμοκρασία μηδέν βαθμών κελσίου έχει αντίσταση 1000Ω.

Εικόνα 18: Αισθητήρας Pt-1000.

3.2.2 Εγκατάσταση των αισθητήρων

Η εγκατάσταση των δύο αισθητήρων υγρασίας και του αισθητήρα θερμοκρασίας έγινε ως εξής. Σε πολύ κοντινή απόσταση από τα δέντρα που μελετήθηκαν, ανοίχτηκαν δύο τρύπες στο έδαφος, η μία στα είκοσι εκατοστά και η δεύτερη στα πενήντα εκατοστά. Στην πρώτη τρύπα των είκοσι εκατοστών τοποθετήθηκε ο ένας αισθητήρας υγρασίας και ο αισθητήρας θερμοκρασίας, ενώ στην δεύτερη τρύπα των πενήντα εκατοστών τοποθετήθηκε ο δεύτερος αισθήτηρας υγρασίας. Στην *Εικόνα 19* δίνεται και η αναπαράσταση του τελικού αποτελέσματος, αφού τελειώσει το στάδιο εγκατάστασης των αισθητήρων.



Εικόνα 19: Αποτέλεσμα Μετά την Εγκατάσταση των Αισθητήρων [55].

Κεφάλαιο 4 - Σύγκριση και ανάλυση αποτελεσμάτων

Σε αυτό το κεφάλαιο παρουσιάζεται και εξηγείται ο γενικός κώδικα που χρησιμοποιήθηκε κατά την διάρκεια της εκπαίδευσης και δοκιμής των αλγορίθμων. Στη συνέχεια γίνεται η ανάλυση όλων των αποτελεσμάτων του κάθε κώδικα ξεχωριστά με σκοπό την εύρεση της ιδανικής τιμής για την μεταβλητή παραμετροποίησης του κάθε αλγορίθμου. Τέλος γίνεται η ολική σύγκριση όλων των αλγορίθμων και επιλέγεται ο καταλληλότερος αλγόριθμος για τις τρεις βάσεις δεδομένων που εκ- παιδεύτηκαν και δοκιμάστηκαν οι αλγόριθμοι.

4.1 Ανάλυση του γενικού κώδικα

Στο πρώτο μέρος έχουμε την εισαγωγή βιβλιοθηκών όπως φαίνεται και στην *Εικόνα 20*. Η πρώτη βιβλιοθήκη ονομάζεται pandas και χρησιμοποιείται για να μπορεί να γίνεται πιο εύκολη η μελέτη των δεδομένων από το xlsx αρχείο. Η δεύτερη βιβλιοθήκη matplotlib δίνει την δυνατότητα σχεδίασης σχημάτων για την καλύτερη οπτικοποίηση των αποτελεσμάτων.

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3  #Gia na diaxwriseis analoga ta dedomena
4  from sklearn.model_selection import train_test_split
5  from sklearn.neighbors import KNeighborsClassifier
6  from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, f1_score
7  import time
```

Εικόνα 20: Ανάλυση γενικού κώδικα 1.

Στην επόμενη γραμμή, γίνεται η εισαγωγή της συνάρτησης train_split_test με την οποία εκπαιδεύτηκαν και δοκιμάστηκαν τα δεδομένα. Στη πέμπτη σειρά θα

υπάρχει συνήθως η εισαγωγή των αλγορίθμων που πρόκειται να χρησιμοποιηθούν, στην συγκεκριμένη φωτογραφία. Για παράδειγμα, ο αλγόριθμος KNeighbors Classifier, ο οποίος θα είναι και ο πρώτος αλγόριθμος που θα εξεταστεί. Στην έκτη σειρά υπάρχει η εισαγωγή των συναρτήσεων, οι οποίες δίνουν την δυνατότητα δημιουργίας των metrics. Τέλος στην έβδομη σειρά γίνεται η εισαγωγή της συνάρτησης time με την οποία θα μπορεί να μετρηθεί ο χρόνος που χρειάζεται ο αλγόριθμος για την λειτουργία του. Στην παρακάτω εικόνα φαίνεται το κομμάτι της εισαγωγής των δεδομένων και της κατάλληλης διαμόρφωσης τους.

```
# Create a dataframe
df = pd.read_excel('R_Filter.xlsx')
df['target'] = df['target'].replace(['healthy', 'unhealthy'],['1','2'])
#Bazeis ti einai to ka8ena se diaforetiko dataframe gt xreiazetai sth synarthsh train_test_split
target = df.target
#Kai to bgazeis apo to main dataframe
df = df.drop('target',axis='columns')
#Oi times average kai sum einai gia na bgaloume to meso oro apo ta 100 paradeigmata pou 8a treksei
average = 0
gg_worst = 1
```

Εικόνα 21: Ανάλυση γενικού κώδικα 2.

Αρχικά με την βοήθεια της βιβλιοθήκης panda, γίνεται πιο εύκολη η μελέτη και ανάθεση των δεδομένων στην *Εικόνα 21* που υπάρχουν στο xlsx σε ένα dataframe. Στη συνέχεια αντικαθίστανται οι ταμπέλες των δεδομένων “healthy” και “unhealthy” με “1” και “2” για να είναι πιο ευανάγνωστη. Έπειτα δημιουργείτε μια καινούργια μεταβλητή με όνομα “target” στην οποία εισάγεται μόνο η στήλη με τις ταμπέλες των δεδομένων και διαγράφεται η στήλη με τα δεδομένα από την κύρια dataframe. Η συγκεκριμένη διαδικασία είναι απαραίτητη, διότι η συνάρτηση train_test_split() απαιτεί την εισαγωγή δυο πινάκων, όπου ο ένας πίνακας είναι τα δεδομένα και ο δεύτερος είναι οι πληροφορίες που δηλώνουν σε ποια κατηγορία ανήκει η κάθε σειρά του dataframe. Τέλος γίνεται η δήλωση των δύο μεταβλητών average και gg_worst οι οποίες θα εξηγηθούν στη συνέχεια. Στην *Εικόνα 22* φαίνεται ο πρώτος βρόγχος του κώδικα, ο οποίος εκτελείται είκοσι φορές κατά τη διάρκεια του προγράμματος. Ο σκοπός του συγκεκριμένου βρόγχου είναι να χωρίζει κάθε φορά τα δεδομένα σε τυχαία μορφή για να μπορεί να παρατηρεί αν υπάρχει

κάποια συγκεκριμένη μορφή διαχωρισμού των δεδομένων η οποία δίνει το βέλτιστο αποτέλεσμα.

```
19  for i in range(20):
20      print("Try " + str(i + 1))
21      X_train, X_test, y_train, y_test = train_test_split(df, target, test_size=0.2)
22      all_average = []
23      all_total_times = []
24      neighbors = 0
25      perfect = 0
26      # Oi times best , best_value , worst kai worst_value
27      best = 0
28      best_value = 0
29      worst = 0
30      worst_value = 0
31      worst_score = 1
32      best_score = 0
```

Εικόνα 22: Ανάλυση γενικού κώδικα 3.

Στην γραμμή 19 ξεκινάει ο πρώτος βρόγχος, στην γραμμή 20 την εντολή print για την εκτύπωση μιας εντολής, η οποία έχει ως σκοπό να βοηθάει να αντιλαμβάνεται κάποιος πιο εύκολα κάθε πότε τελειώνει μια επανάληψη. Τέλος εκτός από τις μεταβλητές που δηλώνονται στο συγκεκριμένο κομμάτι κώδικα και θα εξηγηθούν στη συνέχεια, αξίζει να αναφερθεί ότι ο ρόλος της μεταβλητής test_size που υπάρχει μέσα στη συνάρτηση train_test_split είναι να κρατάει ένα τυχαίο μέρος των δεδομένων για να τα χρησιμοποιήσει έπειτα στο κομμάτι της δοκιμής. Στην *Εικόνα 23* το 0.2 που του έχει ανατεθεί συμβολίζει το 20% των δεδομένων. Ο δεύτερος βρόγχος που ακολουθεί, έχει ως στόχο την παροδική αύξηση των μεταβλητών που θα χρησιμοποιηθούν για την παραμετροποίηση του κάθε αλγορίθμου. Επίσης ξεκινάει και η καταμέτρηση του χρόνου με την συνάρτηση time().

```
34      for i in range(20):
35          start = time.time()
36          sum = 0
37          #Yparxei ena bug me thn deyterh
38          index = i
39          neighbors = neighbors + 1
```

Εικόνα 23: Ανάλυση γενικού κώδικα 4.

Στην *Εικόνα 24* εμφανίζεται ο τρίτος και τελευταίος βρόγχος του κώδικα, ο οποίος έχει ως κύριο στόχο την συνεχή εκπαίδευση και δοκιμή του αλγορίθμου. Αρχικά η εντολή print της γραμμής 43, έχει ως σκοπό την παρακολούθηση του βρόγχου και την εκτύπωση του μηνύματος που της έχει ανατεθεί στο τέλος κάθε επανάληψης του βρόγχου. Στη συνέχεια στην μεταβλητή model δηλώνεται ο αλγόριθμος που επιλέγεται να χρησιμοποιηθεί, στην συγκεκριμένη περίπτωση είναι ο αλγόριθμος KNeighborsClassifier. Ύστερα με την συνάρτηση fit() εκπαιδεύεται ο αλγόριθμος και με την συνάρτηση score() υπολογίζεται η ακρίβεια του αλγορίθμου με τα δεδομένα δοκιμής που έχουν κρατηθεί νωρίτερα, και έπειτα εισάγεται το αποτέλεσμα στη μεταβλητή sum η οποία έχει ως σκοπό την συλλογή και των εκατό αποτελεσμάτων των φάσεων εκπαίδευσης και δοκιμής.

Τέλος ακολουθεί μια συνθήκη και μια υποσυνθήκη (*Εικόνα 26*), οι οποίες έχουν ως σκοπό να καταγράφουν το χειρότερο ή το καλύτερο αποτέλεσμα αντίστοιχα ακρίβειας. Στην περίπτωση που καταγραφεί νέο χειρότερο αποτέλεσμα ακρίβειας, υπολογίζονται και τα αντίστοιχα metrics για περαιτέρω μελέτη.

```
41          for i in range(100):
42              print(int(i) * " " + ".")
43              model = KNeighborsClassifier(n_neighbors=neighbors)
44              model.fit(X_train, y_train)
45              sum = sum + model.score(X_test, y_test)
46              if (model.score(X_test, y_test) < worst_score):
47                  #confusion matirx
48                  knn_predictions = model.predict(X_test)
49                  cm = confusion_matrix(y_test, knn_predictions)
50                  # F1
51                  recall = cm[0][0] / (cm[0][0] + cm[1][0])
52                  precision = cm[0][0] / (cm[0][0] + cm[0][1])
53                  y_pred_knn = model.predict(X_test)  # _proba(X_test)
54                  f1 = f1_score(y_test, y_pred_knn, average=None)
55                  worst_score = model.score(X_test, y_test)
56                  # print(cm)
57              elif (model.score(X_test, y_test) > best_score):
58                  best_score = model.score(X_test, y_test)
```

Εικόνα 24: Ανάλυση γενικού κώδικα 5.

Στη συνέχεια (*Εικόνα 25*), μετά το τέλος του τρίτου βρόγχου υπολογίζεται ο μέσος όρος των εκατό μεταβλητών ακρίβειας και εισάγεται στη λίστα all_average με την εντολή append. Στις επόμενες τρεις σειρές γίνεται η παύση της καταγραφής του χρόνου και της πρόσθεσης του στη λίστα all_total_times με την εντολή append. Έπειτα ακολουθούν τρεις συνθήκες, όπου στην πρώτη γίνεται η καταγραφή του τέλειου αποτελέσματος, το οποίο εμφανίζεται όταν ο αλγόριθμος μπορεί και προβλέπει και στις εκατό φάσεις δοκιμής όλα τα σωστά αποτελέσματα. Στην δεύτερη και στην τρίτη επανάληψη γίνεται ο έλεγχος, για το αν το average είναι το νέο καλύτερο ή το νέο χειρότερο αποτέλεσμα και γίνεται η αντίστοιχη ανάθεση.

```
59          average = sum/100
60          all_average.append(average)
61          end = time.time()
62          total_time = (end - start)
63          all_total_times.append(total_time)
64          #Yparxei ena bug me thn deyterh 8esh tou pinaka ,gi
65          if(index == 1):
66              if(average == 1):
67                  perfect = perfect + 1
68          elif (average > best):
69              best = average
70              best_value = neighbors
71              if(average == 1):
72                  perfect = 1
73          elif (average < worst or worst == 0):
74              worst = average
75              worst_value = neighbors
```

Εικόνα 25: Ανάλυση γενικού κώδικα 6.

Εάν το average είναι χειρότερο από το προϋπάρχον τότε γίνεται ένας ακόμα έλεγχος, για το αν το average είναι χειρότερο από όλα τα worst που έχουν περάσει από την έναρξη του κώδικα. Εάν ισχύει και αυτό τότε το πρόγραμμα ζητάει από τον χρήστη αν θέλει να του εμφανίσει επιπλέον στοιχεία και metrics για το συγκεκριμένο αποτέλεσμα (*Εικόνα 26*). Στο τέλος της εικόνας υπάρχει μια ακόμα υποσυνθήκη η οποία ελέγχει αν το average είναι perfect, δηλαδή αν ο

αλγόριθμος έχει ποσοστό ευστοχίας 100% σε όλες τις δοκιμές που έκανε.

```
        if ( worst < gg_worst):
            gg_worst = worst
            print("There is a new worst result , do you want to print it? ( " + str(gg_worst) + " ) ")
            choice = input()
            if(str(choice) == "y"):
                df_show_train_worst = pd.DataFrame(X_train)
                df_show_train_worst2 = df_show_train_worst.assign(predictions= y_train.values)
                df_show_train_worst2.to_excel("WorstResultData.xlsx")
                # Confusion Matrix
                print("The worst score of this loop was: " + str(worst_score))
                print("And the best was: " + str(best_score))
                print("The F1 score is: " + str(f1))
                print("The precision was: " + str(precision))
                print("The recall was: " + str(recall))
                disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=model.classes_)
                disp.plot()
                plt.show()
    elif (average == 1 and average == best):
        perfect = perfect + 1
    average = 0
```

Εικόνα 26: Ανάλυση γενικού κώδικα 7.

Στη συνέχεια εκτυπώνεται το καλύτερο και χειρότερο αποτέλεσμα (*Εικόνα 27*) και δίνεται η δυνατότητα εκτύπωσης και εμφάνισης των metrics. Αν ο χρήστης εισάγει το “y” τότε εμφανίζουμε τα metrics F1 score και τα αντίστοιχα Confusion Matrix.

```
print(all_average)
print(all_total_times)
if(perfect == 20):
    print("Perfect Result!")
else:
    print("The best value was: " + str(best))
    print("And the worst was: " + str(worst))
    print("The n_neighbors i for best was: " + str(best_value) + " and for the worst value was: "
    print("Do you want to metrics and keep the result in an excel? (Write 'y')")
    print("The perfect result was: " + str(perfect))
choice = input()
#Ama 8eloujme na kanoume ta metrics ths sygkekrimenhs periptwshs , patame edw Y
if(str(choice) == "y"):
    #F1
    y_pred_knn = model.predict(X_test)  # _proba(X_test)
    print(f1_score(y_pred_knn, y_test, average=None))
    knn_predictions = model.predict(X_test)
    cm = confusion_matrix(y_test, knn_predictions)
    print(cm)
    #Confusion Matrix
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=model.classes_)
    disp.plot()
    plt.show()
    # Gia na deis an yparxei kapoia NaN metablhth mesa
    # df.columns[df.isna().any()]
    print("Decide between best or worst (For Best press 'b' and for Worst 'w')")
    best_worst = input()
```

Εικόνα 27 : Ανάλυση γενικού κώδικα 8.

Στο τέλος του κώδικα (*Εικόνα 28*) ζητάμε από τον χρήστη αν θέλει να μεταφέρει τα δεδομένα εκπαίδευσης και τα δεδομένα δοκιμής σε δύο φύλλα xlsx για τον εντοπισμό μοτίβων. Ο χρήστης μπορεί να απαντήσει με “b” ή “w” για να τα αναθέσει στα αντίστοιχα φύλλα.

```
        if (str(best_worst) == "b"):
            print("Best it is")
            df_show_train_best = pd.DataFrame(X_train)
            df_show_train_best2 = df_show_train_best.assign(predictions=y_train.values)
            df_show_train_best2.to_excel("BestTrainData.xlsx")
            df_show_test_best = pd.DataFrame(X_test)
            df_show_test_best2 = df_show_test_best.assign(predictions=y_test.values)
            df_show_test_best.to_excel("BestTestData.xlsx")
        #Gia to xeirotero apotelesma meta
        elif (str(best_worst) == "w"):
            print("Worst it is")
            df_show_train_worst = pd.DataFrame(X_train)
            df_show_train_worst2 = df_show_train_worst.assign(predictions=y_train.values)
            df_show_train_worst2.to_excel("WorstTrainData.xlsx")
            df_show_test_worst = pd.DataFrame(X_test)
            df_show_test_worst2 = df_show_test_worst.assigm(predictions=y_test.values)
            df_show_test_worst2.to_excel("WorstTestData.xlsx")
    else:
        continue

```

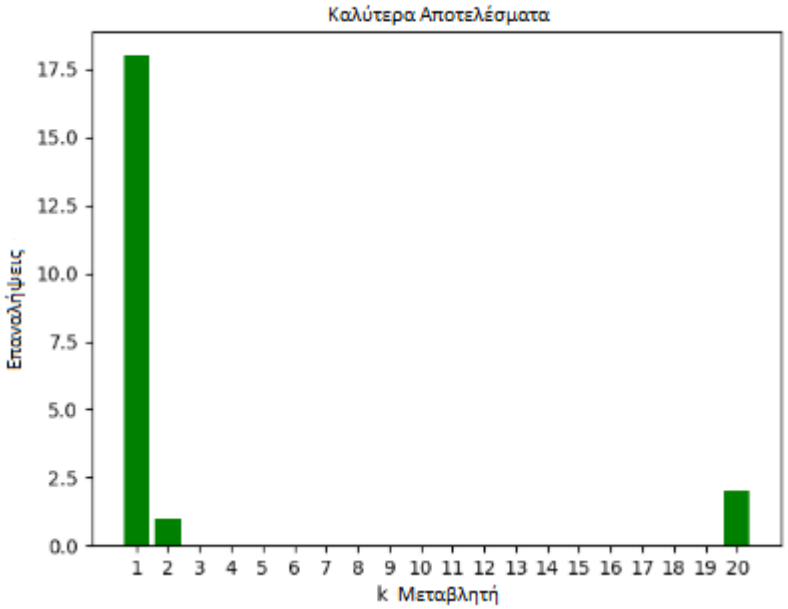
Εικόνα 28: Ανάλυση γενικού κώδικα 9.

4.2 Αλγόριθμος KNeighborsClassifier

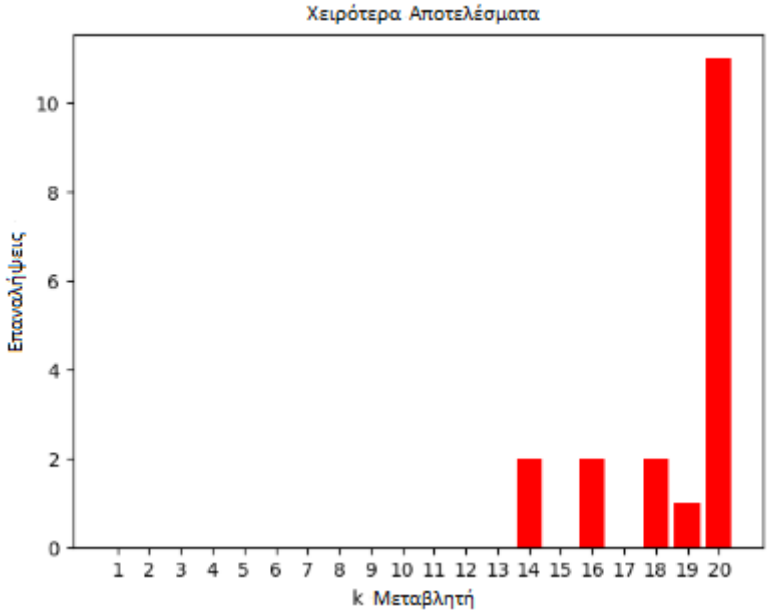
Ο αλγόριθμος KNeighborsClassifier ανήκει στην κατηγορία των K-nearest neighbors. Στον συγκεκριμένο αλγόριθμο δοκιμάστηκε για την παραμετροποίηση του αλγορίθμου, τιμές από το 1 έως το 20 για τη μεταβλητή k. Η μεταβλητή k στον αλγόριθμο KNeighborsClassifier δηλώνει πόσες τιμές θα κοιτάξει γύρω από την τιμή που αναζητάει. Η πρώτη δοκιμή δεδομένων έγινε στα R_filter δεδομένα και παρατηρήθηκαν τα εξής:

Το καλύτερο αποτέλεσμα εμφανίστηκε δεκαοχτώ φορές για k = 1, μία φορά στο k = 2 και δύο φορές για k = 20. Να σημειωθεί ότι συνολικά είναι είκοσι μία φορές διότι τη μία φορά που βρέθηκε το καλύτερο αποτέλεσμα στο k = 20 υπήρχε το ίδιο ακριβώς αποτέλεσμα και για k = 1 και είχαν και τα δύο k ποσοστό ευστοχίας 100%. Ενώ τα χειρότερα αποτελέσματα, έγιναν δύο φορές για k = 14, k = 16, και k = 18, μία φορά για k = 19 και έντεκα φορές

για k = 20. Στις εικόνες 29 και *Εικόνα 30* γίνεται η γραφική τους αναπαράσταση.

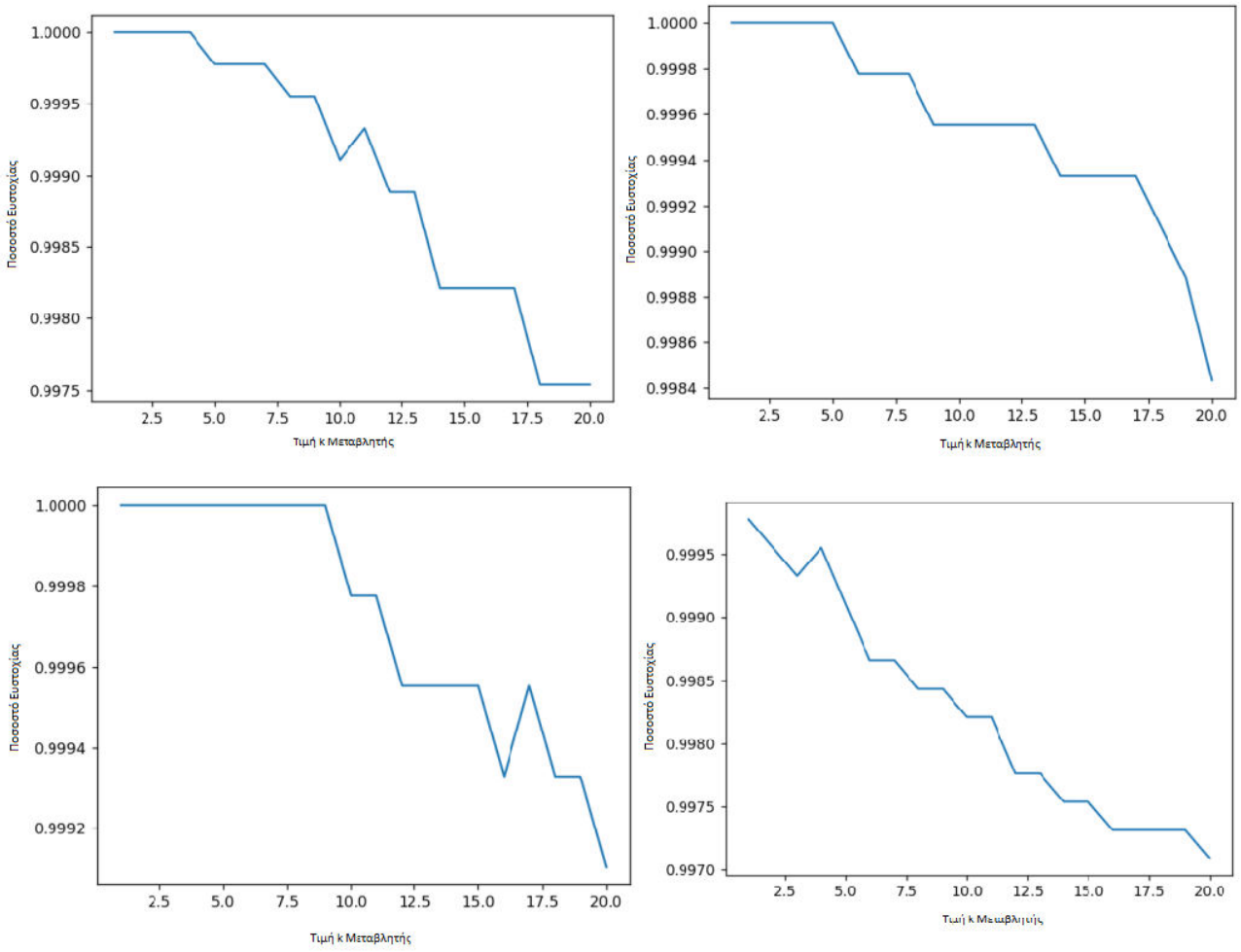


Εικόνα 29: Διάγραμμα Καλύτερων Αποτελεσμάτων στα δεδομένα R_filter του KNN αλγορίθμου.

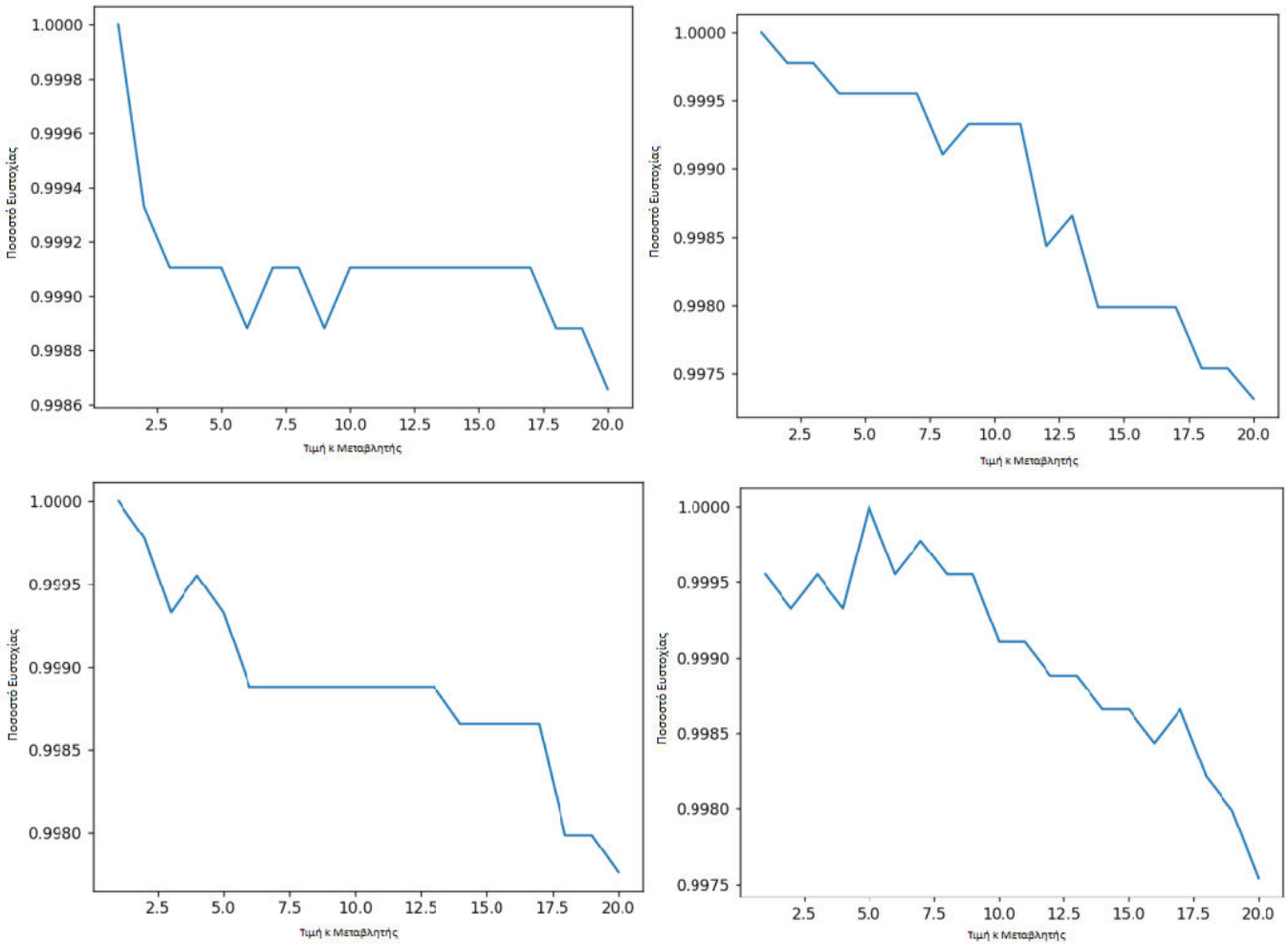


Εικόνα 30: Διάγραμμα Χειρότερων Αποτελεσμάτων στα δεδομένα R_filter του KNN αλγορίθμου.

Στις εικόνες *Εικόνα 31* και *Εικόνα 32* που ακολουθούν, παρουσιάζονται οκτώ από τις είκοσι δοκιμές του αλγορίθμου. Αυτό που συμπεραίνεται, είναι ότι όσο μεγαλώνει η μεταβλητή k τόσο μικραίνει και η ευστοχία του αλγορίθμου με κάποιες ελάχιστες εξαιρέσεις. Ενώ σε όλες τις περιπτώσεις το k = 1 έχει την καλύτερη ευστοχία με ποσοστό 100% και το k = 20 έχει την χειρότερη ευστοχία.

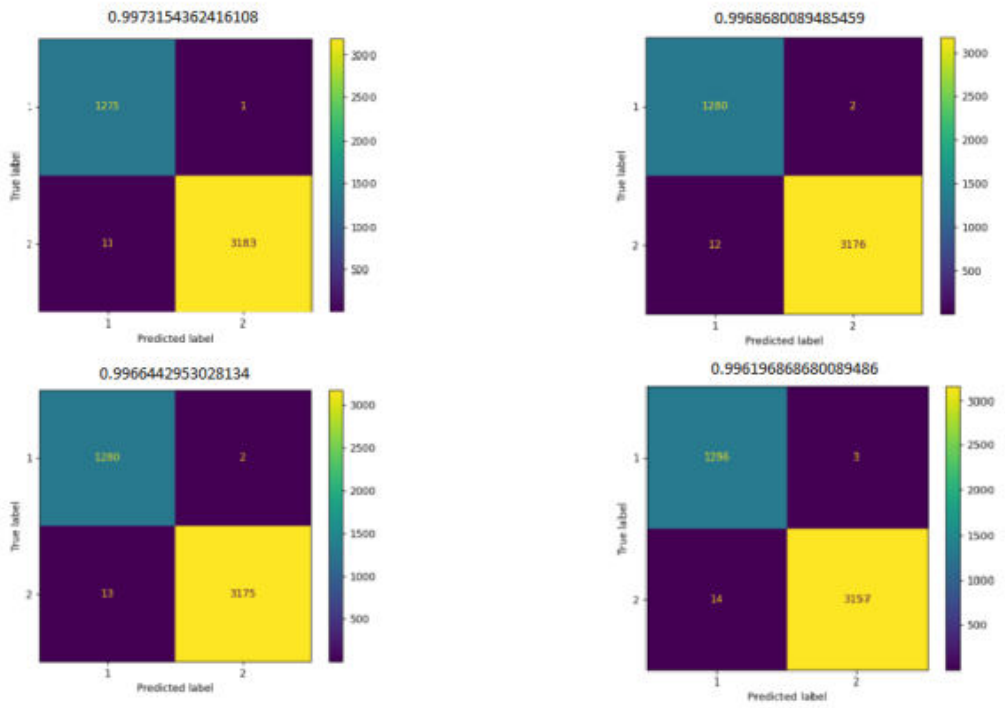


Εικόνα 31: Διάγραμμα Ποσοστού ευστοχίας του αλγορίθμου KNN στα δεδομένα R_filter 1.



Εικόνα 32: Διάγραμμα Ποσοστού ευστοχίας του αλγορίθμου KNN στα δεδομένα R_filter 2.

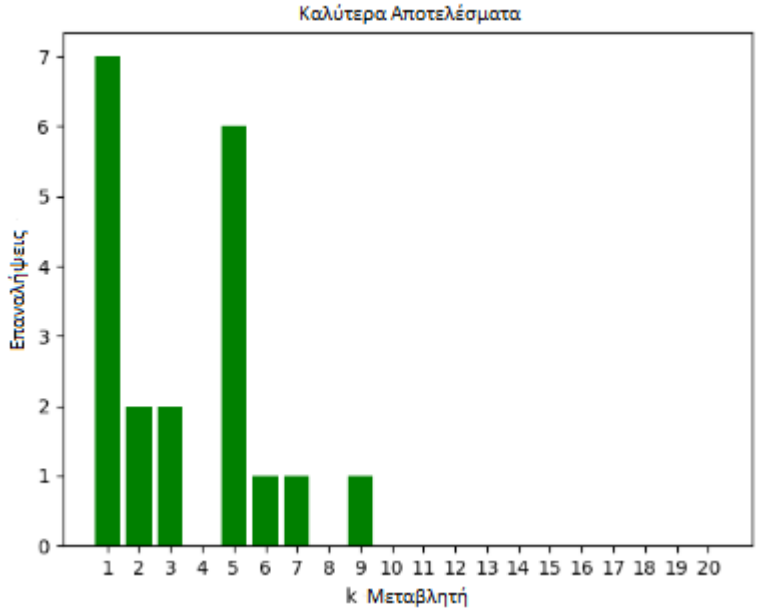
Στην *Εικόνα 33* παρουσιάζονται τα τέσσερα χειρότερα αποτελέσματα με τα confusion matrix τους και το ποσοστό ευστοχία τους ακριβώς από πάνω από τα σχήματα, από τα οποία παρατηρείται ότι τα περισσότερα λάθη που έχει κάνει ο αλγόριθμος είναι στις περιπτώσεις που προέβλεψε ότι ένα δέντρο είναι άρρωστο ενώ δεν ήταν.



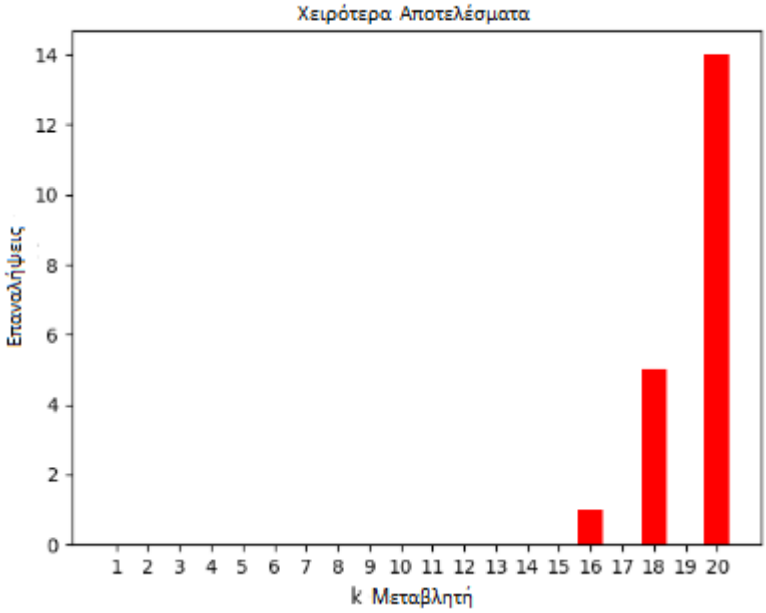
Εικόνα 33: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου KNN στα δεδομένα R_filter.

Η δεύτερη δοκιμή δεδομένων έγινε στα R_filter_without_temp δεδομένα και παρατηρήθηκαν τα εξής:

Τα καλύτερα αποτελέσματα παρατηρήθηκαν κατά κύριο λόγο στα k = 1 και k = 5 με εφτά και έξη φορές αντίστοιχα. Επίσης υπήρξαν για δύο φορές τα καλύτερα αποτελέσματα στο k = 2 και k = 3 και τέλος στο k = 8 και k = 9 εμφανίστηκε μία φορά το καλύτερο αποτέλεσμα. Από την άλλη, τα χειρότερα αποτελέσματα εμφανίστηκαν στο k = 20 με δεκατέσσερις επαναλήψεις, στο k = 18 με πέντε επαναλήψεις και τέλος στο k = 16 με μία μόνο επανάληψη. Στις εικόνες *Εικόνα 34* και *Εικόνα 35* γίνεται η απεικόνιση των παραπάνω.



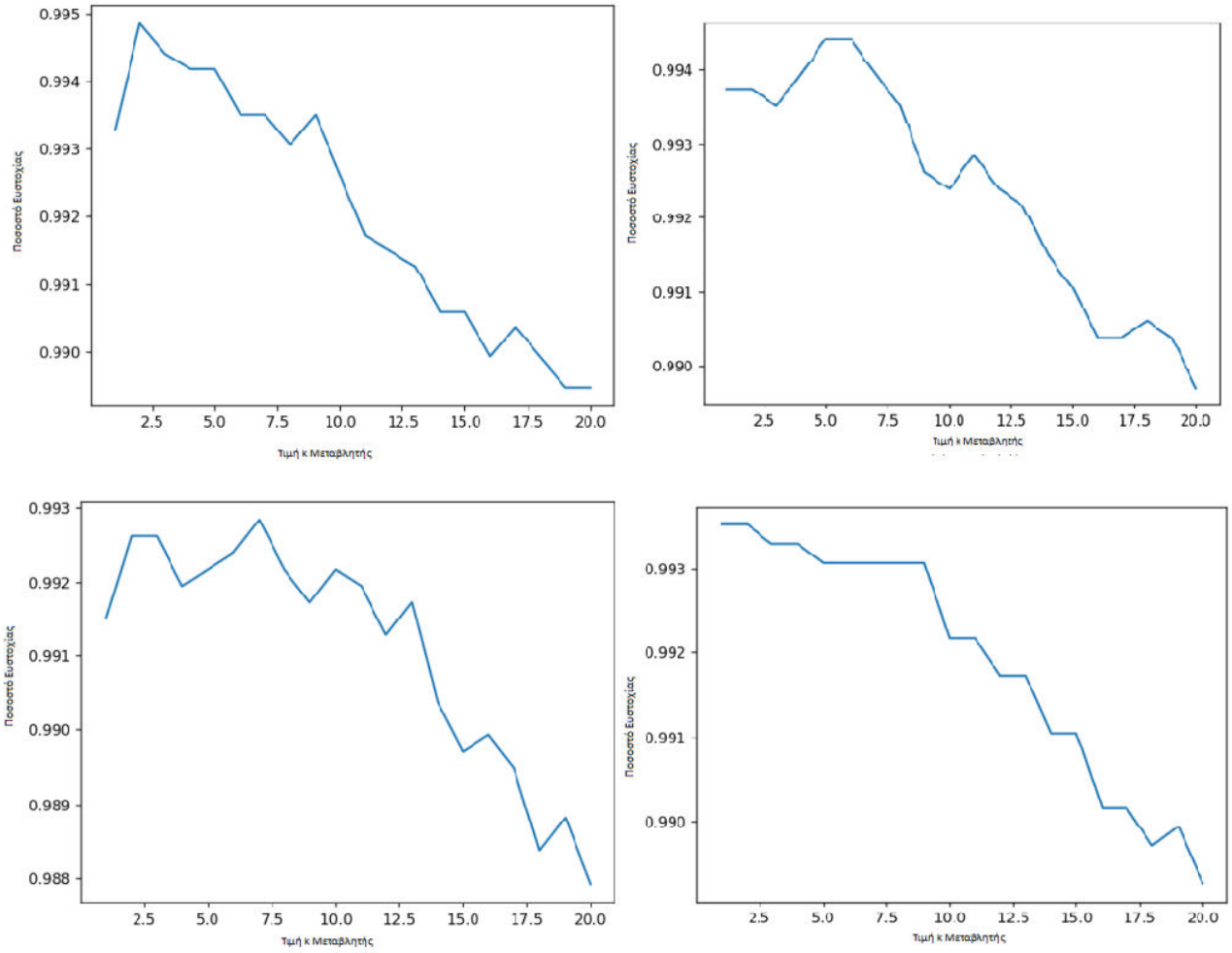
Εικόνα 34: Διάγραμμα Καλύτερων Αποτελεσμάτων στα δεδομένα R_filter_no_temp του KNN αλγορίθμου.



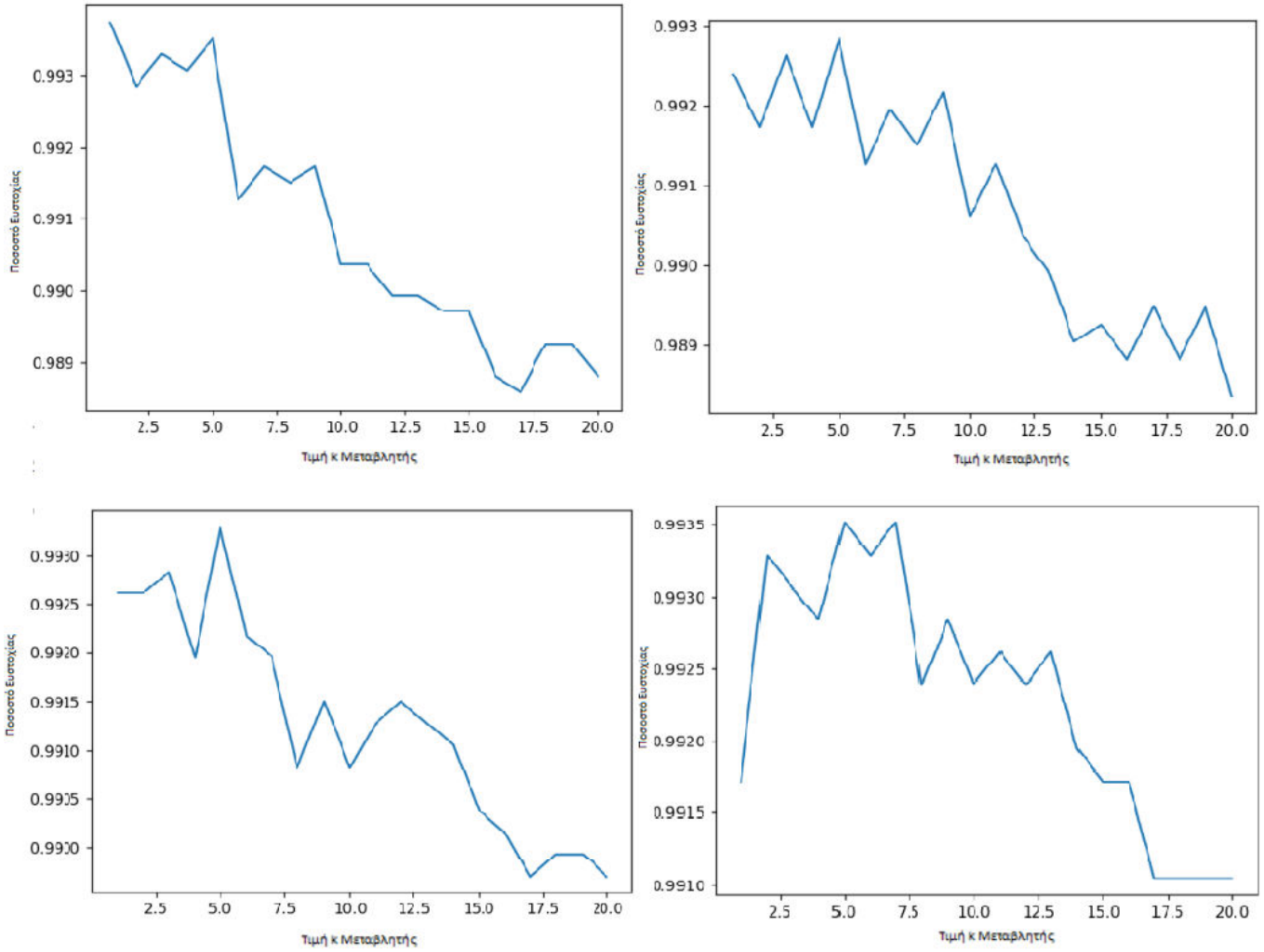
Εικόνα 35: Διάγραμμα Χειρότερων Αποτελεσμάτων στα δεδομένα R_filter_no_temp του KNN αλγορίθμου.

Στις εικόνες *Εικόνα 36* και *Εικόνα 37* που ακολουθούν, παρουσιάζονται οκτώ από τις είκοσι γραφικές παραστάσεις και φαίνεται ότι όπως και στην

πρώτη δοκιμή του αλγορίθμου με τα δεδομένα R_filter, ότι όσο αυξάνεται το k τόσο αρχίζει και μειώνεται η ευστοχία του αλγορίθμου. Στην συγκεκριμένη δοκιμή όμως αντί να υπάρχει το k = 1 ως καλύτερο αποτέλεσμα, τα πρώτα έξι με εφτά k παρουσιάζουν τις καλύτερες τιμές, ενώ πάλι το k = 20 και το k = 18 παρουσιάζουν την χειρότερη ευστοχία

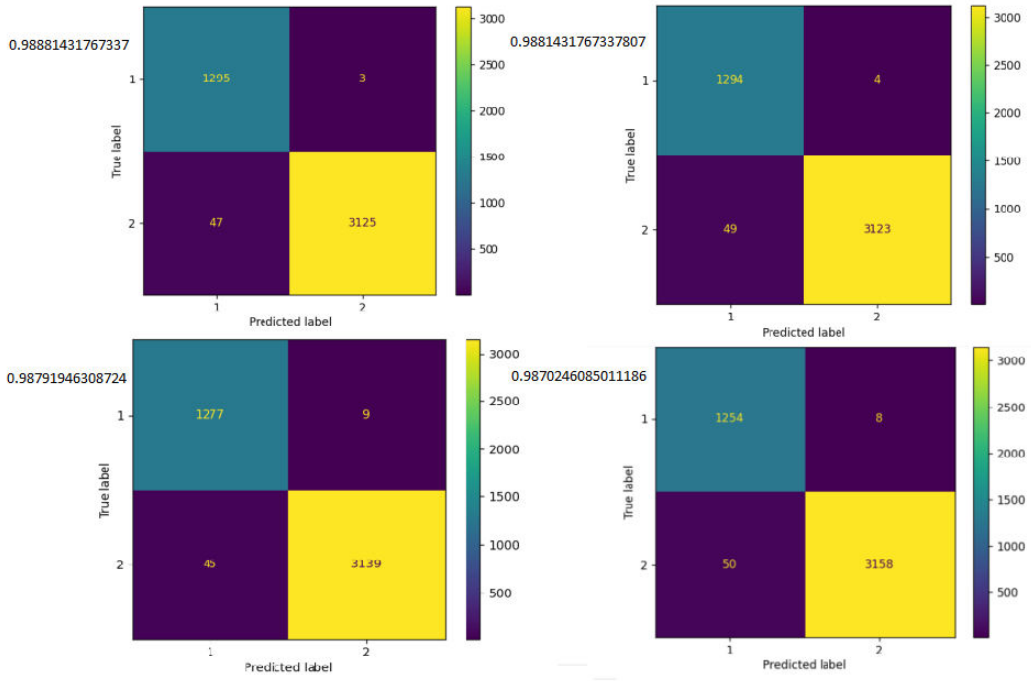


Εικόνα 36: Διάγραμμα Ποσοστού ευστοχίας του αλγορίθμου KNN στα δεδομένα R_filter_no_temp 1.



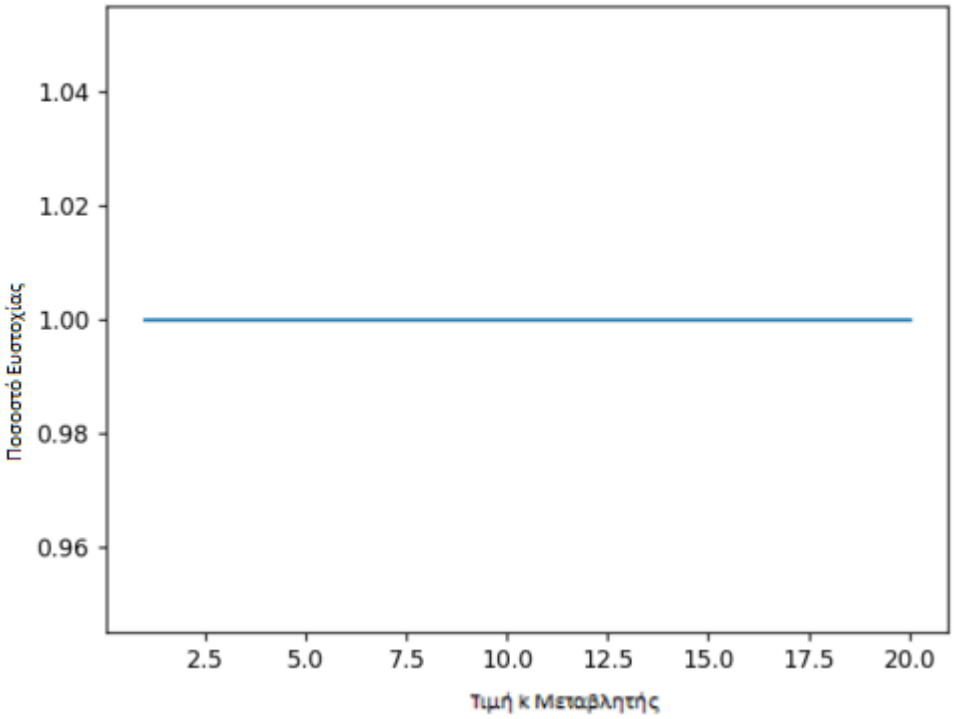
Εικόνα 37: Διάγραμμα Ποσοστού ευστοχίας του αλγορίθμου KNN στα δεδομένα R_filter_no_temp 2 .

Στην *Εικόνα 38* παρουσιάζονται τα τέσσερα χειρότερα αποτελέσματα με τα confusion matrix τους και το ποσοστό ευστοχίας τους ακριβώς αριστερά από τα σχήματα, από τα οποία παρατηρούμε ότι τα περισσότερα λάθη που έχει κάνει ο αλγόριθμος είναι στις περιπτώσεις που προέβλεψε ότι ένα δέντρο είναι άρρωστο ενώ δεν ήταν, όπως έγινε και στην πρώτη δοκιμή με τα R_filter δεδομένα.



Εικόνα 38: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου KNN στα δεδομένα R_filter_no_temp.

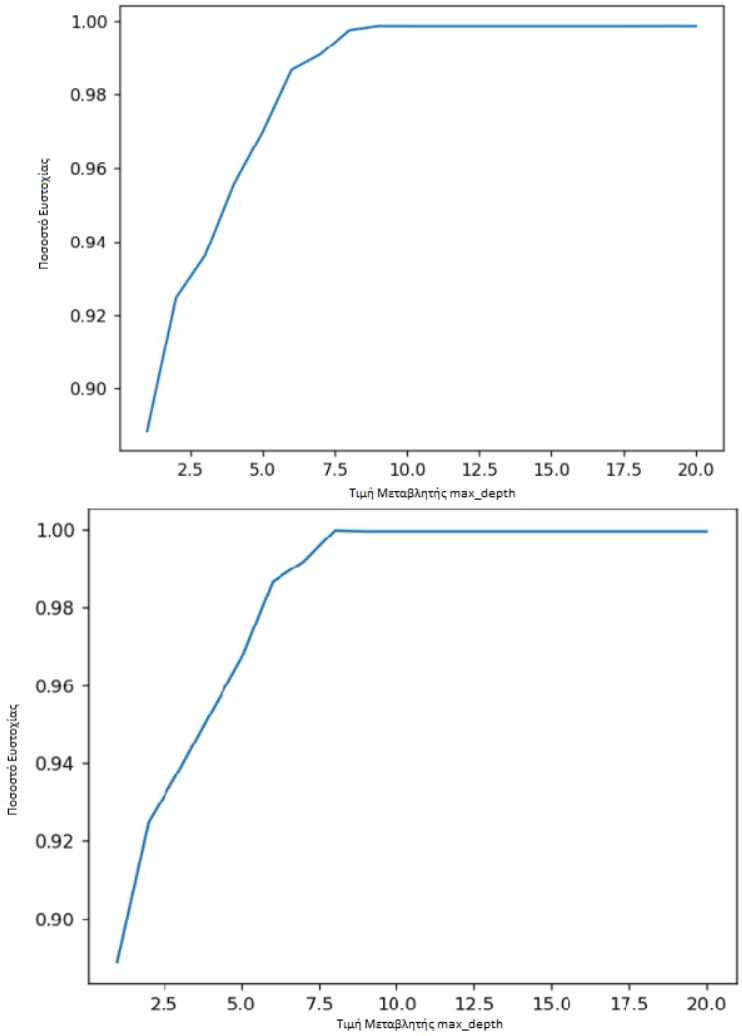
Στην τρίτη δοκιμή δεδομένων που έγινε στα R_filter_balanced δεδομένα , τα ποσοστά ευστοχίας ήταν 100%, ανεξαρτήτως από την k μεταβλητή, όπως φαίνεται και στην *Εικόνα 39* που ακολουθεί.



Εικόνα 39: Διάγραμμα Ποσοστού ευστοχίας του αλγορίθμου KNN στα δεδομένα R_filter_balanced.

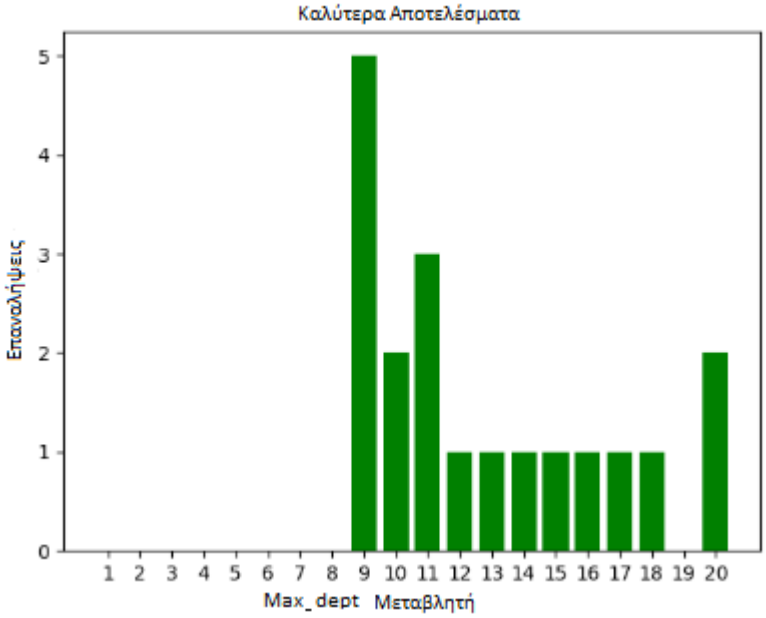
4.2.1 Αλγόριθμος DecisionTreeClassifier

Ο αλγόριθμος DecisionTreeClassifier ανήκει στην κατηγορία των Decision Trees αλγορίθμων. Η μεταβλητή που χρησιμοποιήθηκε για την παραμετροποίηση του αλγορίθμου είναι η μεταβλητή max_depth, η οποία είναι εκείνη που δηλώνει το μέγιστο βάθος που επιτρέπεται να φτάσει ο αλγόριθμος. Για την πρώτη δοκιμή που έγινε στα δεδομένα R_filter χρησιμοποιήθηκαν τιμές για το max_depth από το 1 έως το 20 και παρατηρήθηκε ότι για τα πρώτα εφτά max_depth τα ποσοστά ευστοχίας ήταν πάντα τα πιο χαμηλά, ενώ στη συνέχεια για τα επόμενα max_depth το ποσοστό ευστοχίας κυμαίνεται περίπου το ίδιο με το καλύτερο αποτέλεσμα να είναι 100% ποσοστό ευστοχίας Εικόνα 40.

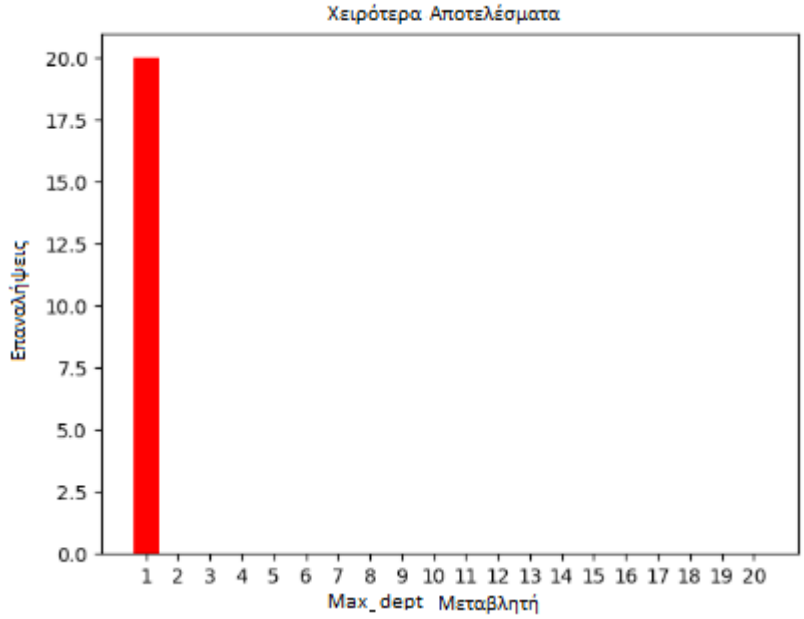


Εικόνα 40: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου DT στα R_filter δεδομένα.

Στις εικόνες *Εικόνα 41* και *Εικόνα 42* μπορεί κανείς να παρατηρήσει και καλύτερα ότι τα αποτελέσματα με την μεγαλύτερη ευστοχία μοιράζονται ανάμεσα στις k τιμές που ισούνται με μεγαλύτερο από εννιά, με εξαίρεση το k = 19. Ενώ το χειρότερο αποτέλεσμα εμφανίζεται συνέχεια στο k = 1.

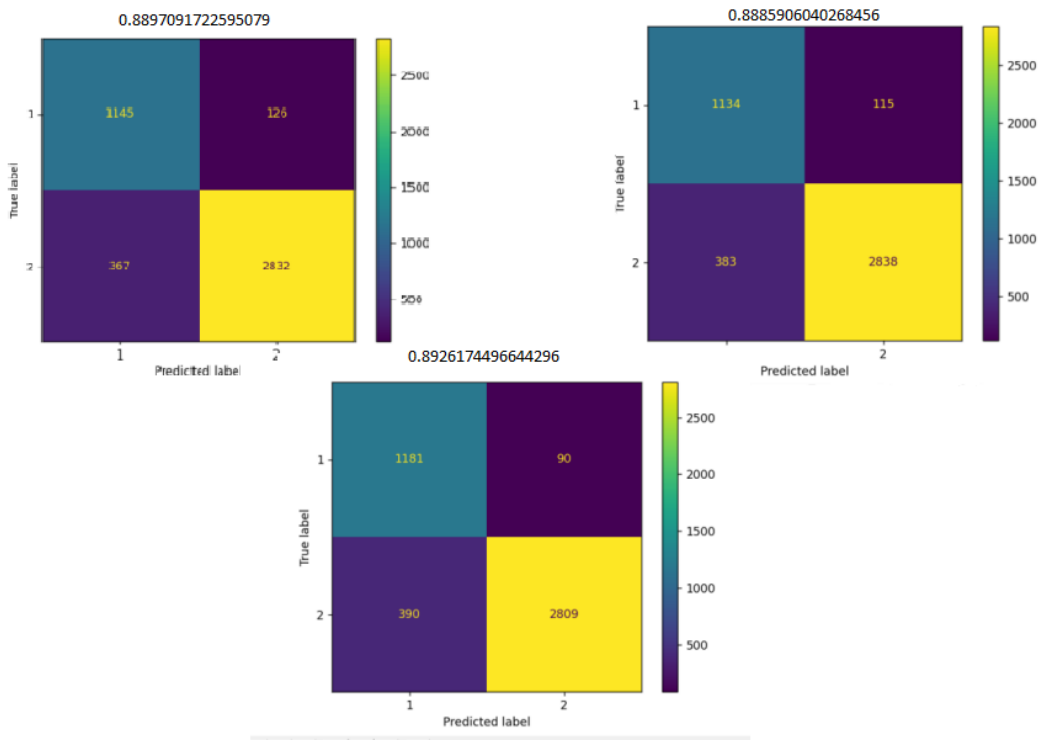


Εικόνα 41: Διάγραμμα Καλύτερων Αποτελεσμάτων στα δεδομένα R_filter του DT αλγορίθμου.

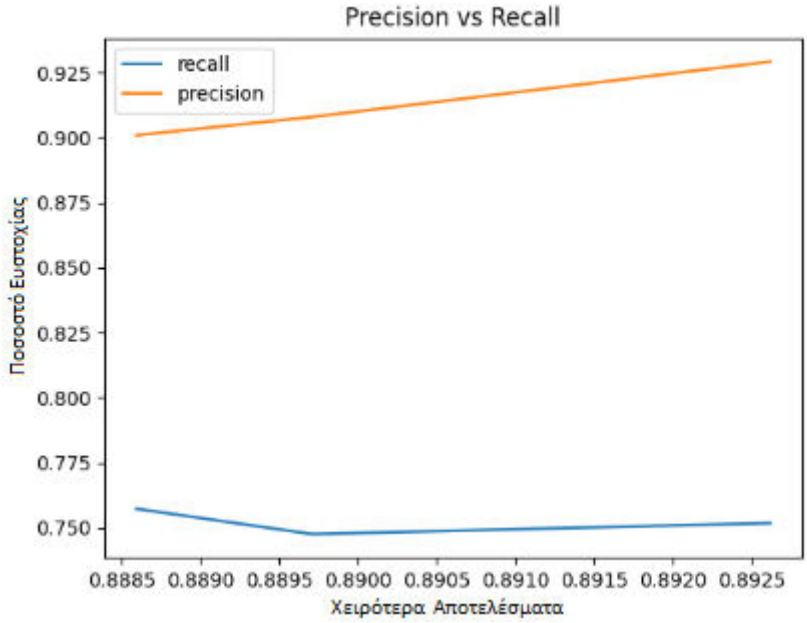


Εικόνα 42: Διάγραμμα Χειρότερων Αποτελεσμάτων στα δεδομένα R_filter του DT αλγορίθμου.

Τρία χειρότερα αποτελέσματα παρουσιάζονται στην *Εικόνα 43*. Και σε αυτόν τον αλγόριθμο παρατηρείται ότι οι περισσότερες λανθασμένες προβλέψεις έγιναν για την πρόβλεψη του άρρωστου δέντρου όπως φάνηκε και στον KNeighborsClassifier, αν η μεταβλητή recall είναι αισθητά χαμηλότερη σε σχέση με την μεταβλητή precision του αλγορίθμου, όπως φαίνεται και στην *Εικόνα 46*.



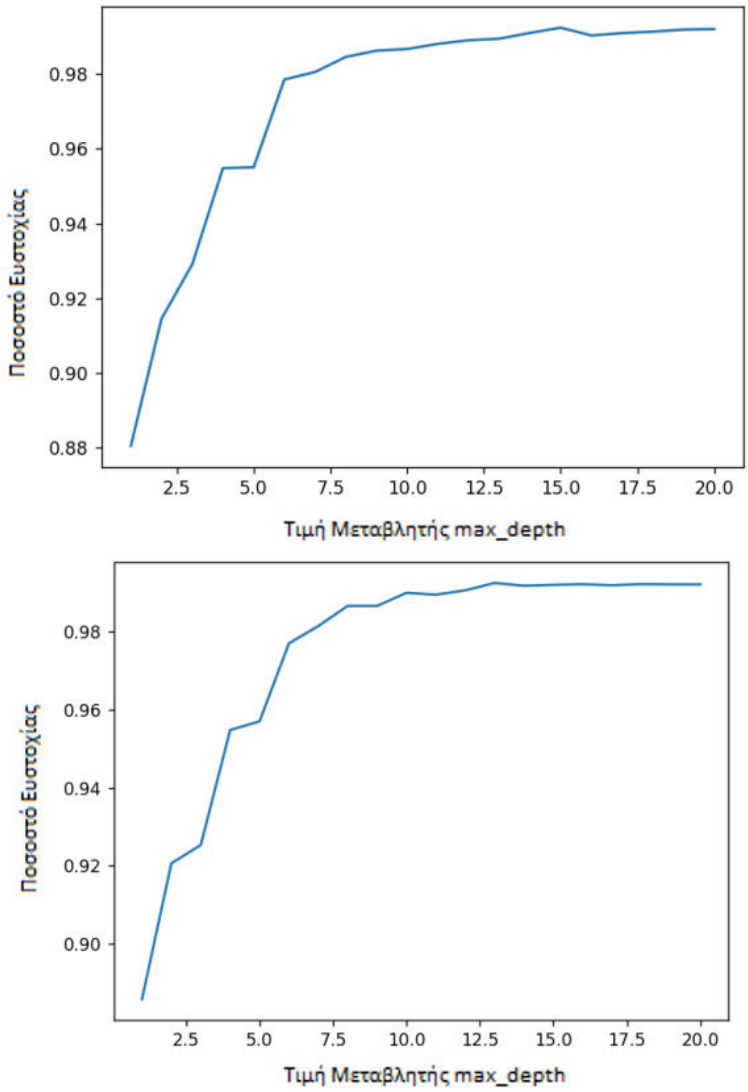
Εικόνα 43: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου DT στα δεδομένα R_filter.



Εικόνα 44: Διάγραμμα Precision vs Recall του αλγορίθμου DT στα δεδομένα R_filter.

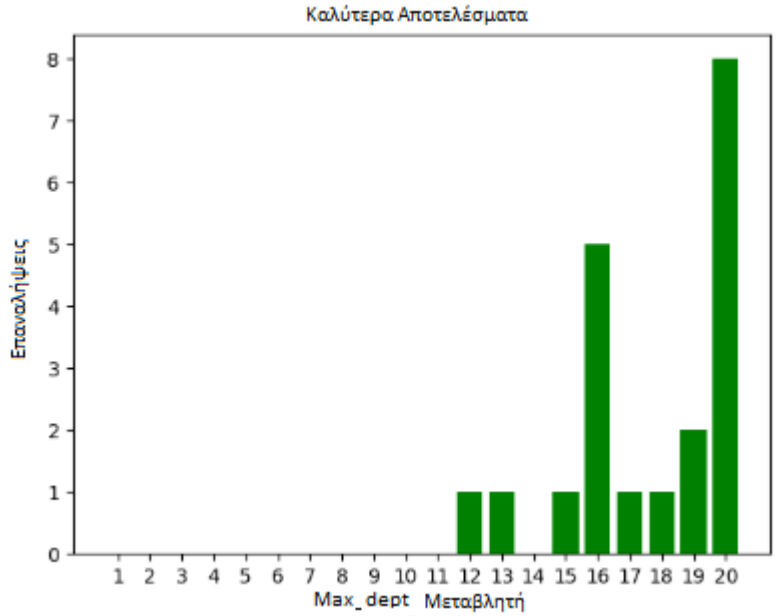
Για την δεύτερη δοκιμή που έγινε στα δεδομένα R_filter_without_temp χρησιμοποιήθηκαν το ίδιο εύρος τιμών όπως και στη πρώτη δοκιμή για τη μεταβλητή max_depth. Παρόμοια με την πρώτη δοκιμή, τα καλύτερα αποτελέσματα εμφανίστηκαν όταν η τιμή του max_depth ήταν πάνω από εφτά

(*Εικόνα 45*).

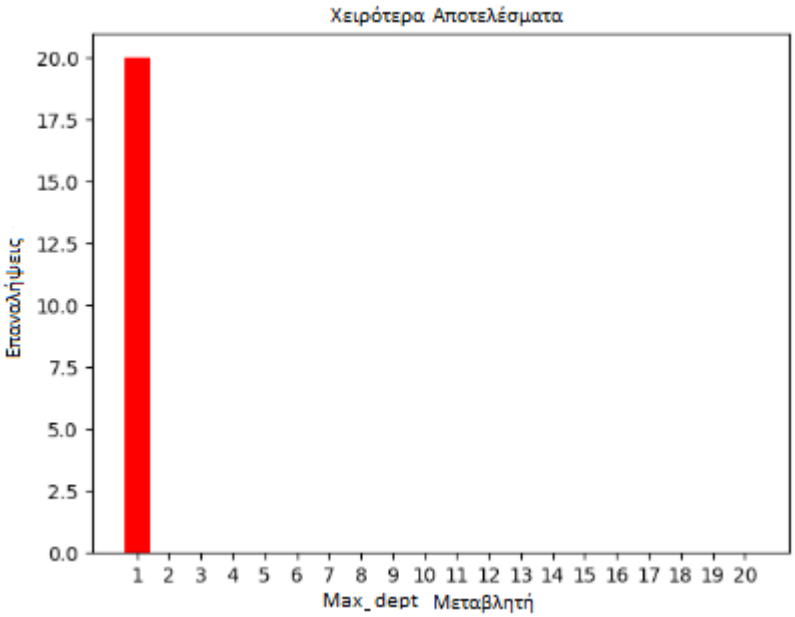


Εικόνα 45: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου DT στα R_filter_no_temp δεδομένα.

Στις εικόνες Εικόνα *46* και Εικόνα *47* παρατηρείται ότι όπως και στην πρώτη δοκιμή σε όλες τις περιπτώσεις το max_depth = 1 έχει το λιγότερο ποσοστό ευστοχίας, ενώ τα αποτελέσματα από max_depth = 12 και πάνω εμφάνισαν όλα τουλάχιστον μια φορά το μεγαλύτερο ποσοστό ευστοχίας με εξαίρεση το max_depth = 14.

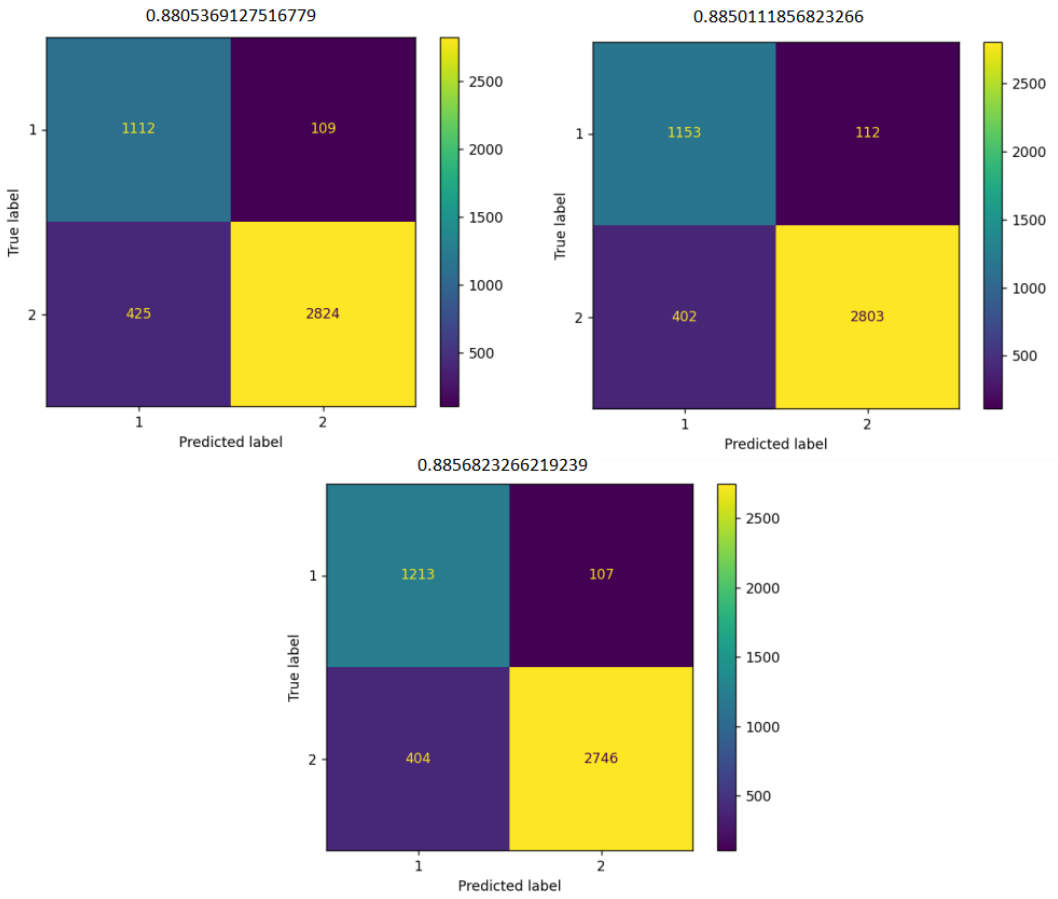


Εικόνα 46: Διάγραμμα Καλύτερων Αποτελεσμάτων στα δεδομένα R_filter_no_temp του DT αλγορίθμου.



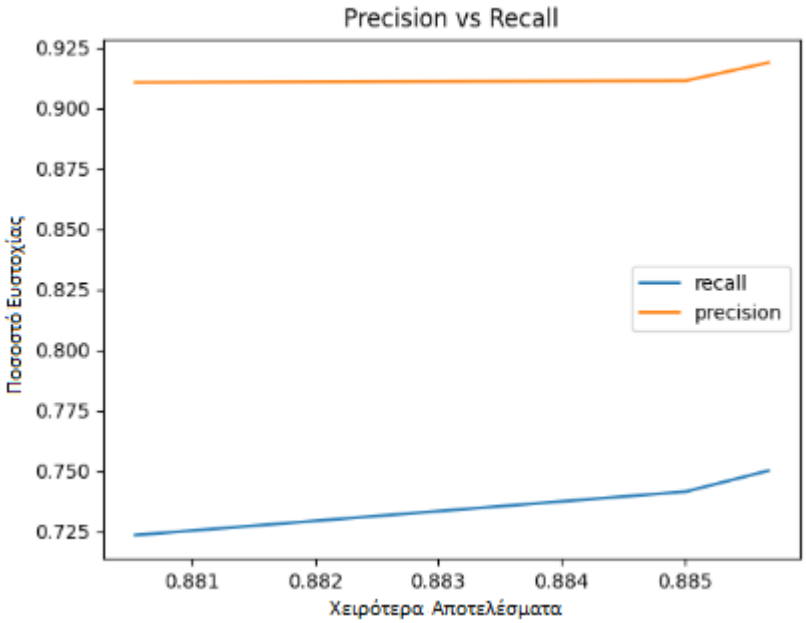
Εικόνα 47: Διάγραμμα Χειρότερων Αποτελεσμάτων στα δεδομένα R_filter_no_temp του DT αλγορίθμου.

Τα τρία χειρότερα παρουσιάζονται στην *Εικόνα 48*. Τα ποσοστά είναι λίγο χαμηλότερα από την πρώτη δοκιμή με τα δεδομένα R_filter.



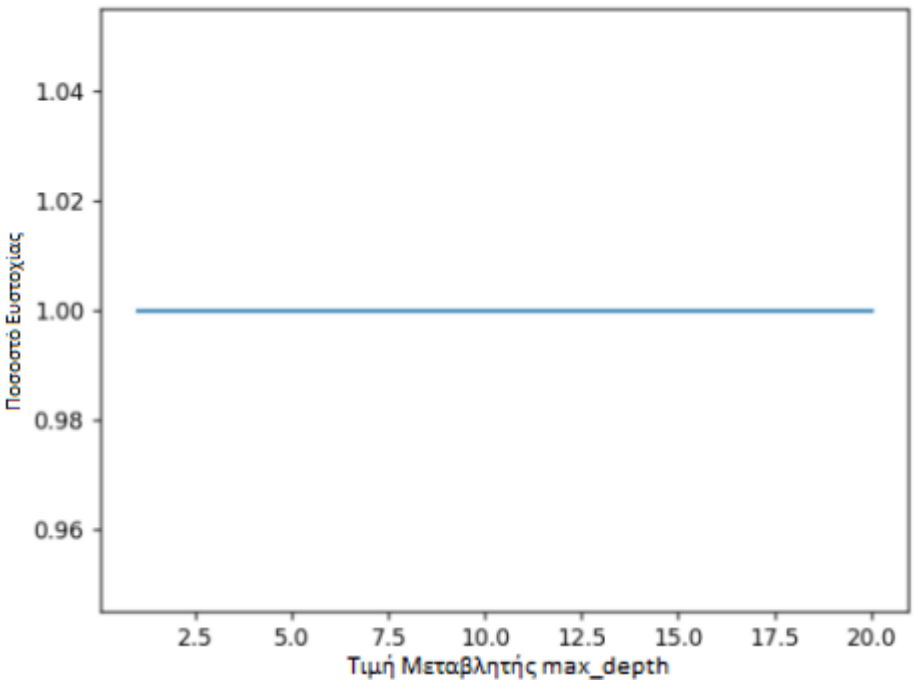
Εικόνα 48: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου DT στα δεδομένα R_filter_no_temp.

Η διαφορά στις τιμές με του precision και του recall έχει αυξηθεί ελάχιστα στην δεύτερη δοκιμή (Εικόνα *49*).



Εικόνα 49: Διάγραμμα Precision vs Recall του αλγορίθμου DT στα δεδομένα R_filter_no_temp.

Στην τρίτη δοκιμή, η δοκιμή έγινε στα δεδομένα R_filter_balanced, όπου και ο DecisionTreeClassifier είχε σε όλες τις δοκιμές του 100 (*Εικόνα 50*).

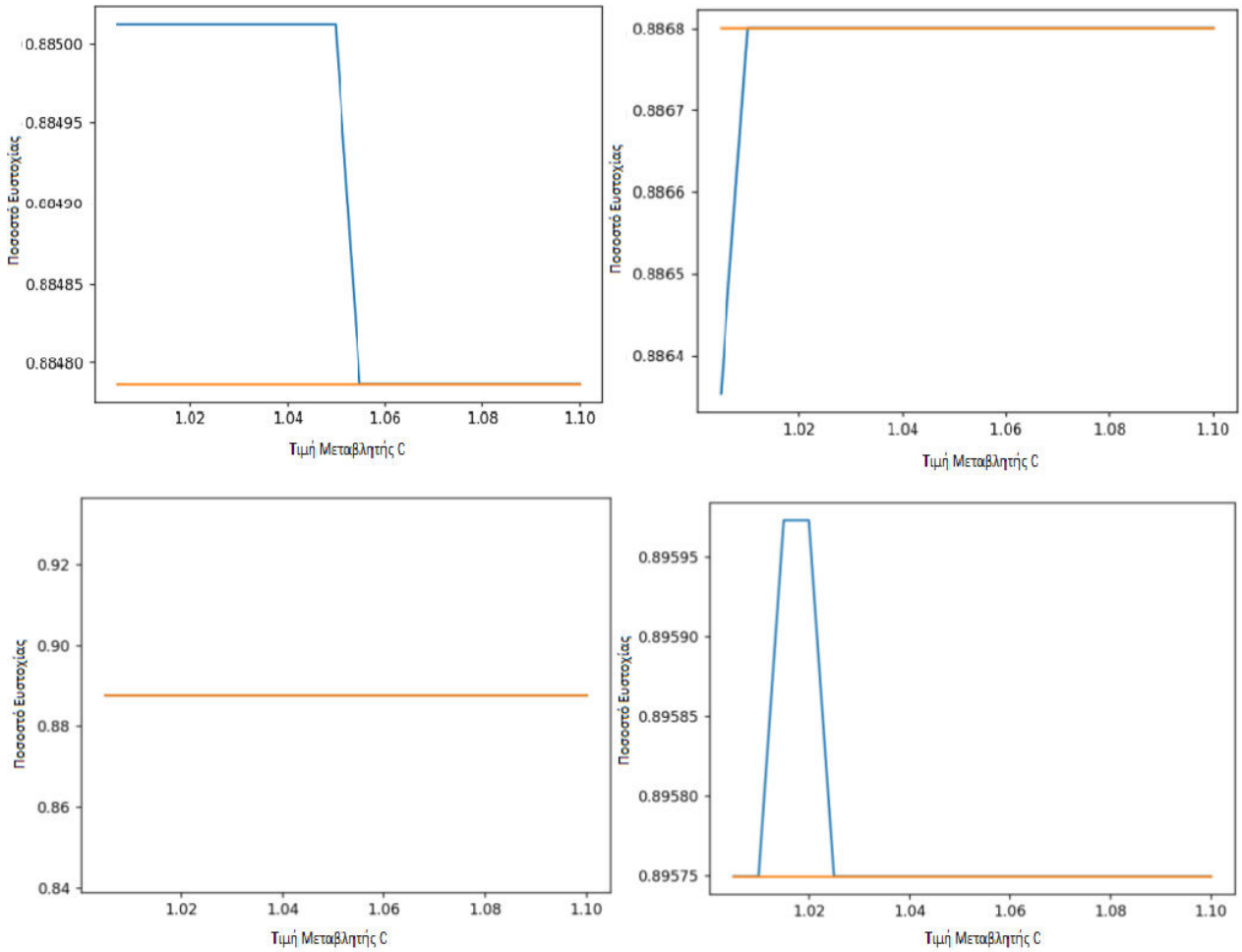


Εικόνα 50 : Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου DT στα R_filter_balanced δεδομένα

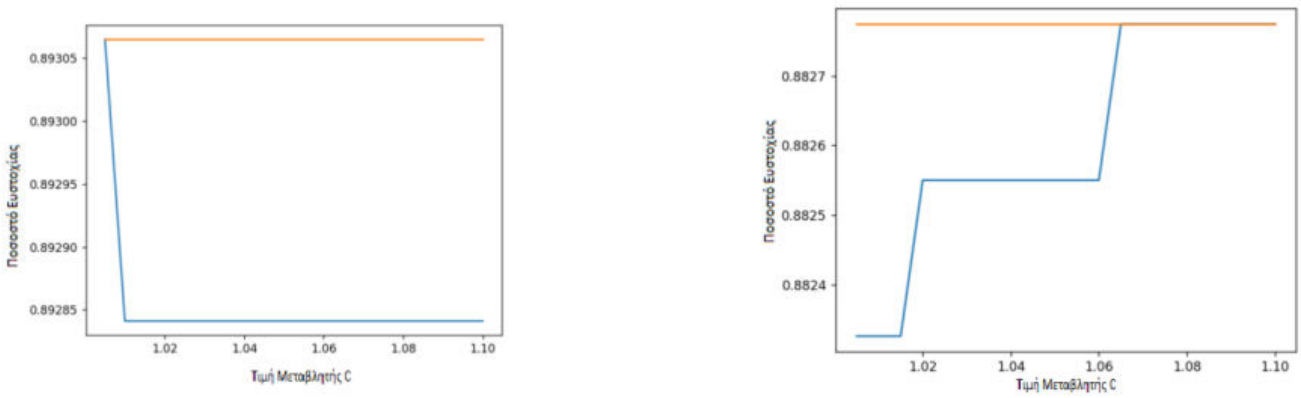
4.2.2 Αλγόριθμος Logistic Regression

Ο αλγόριθμος Logistic Regression ανήκει στην κατηγορία των Γραμμικών αλγόριθμων. Στον συγκεκριμένο αλγόριθμο δοκιμάστηκαν για την παραμετροποίηση του αλγορίθμου, τιμές από το 1.005 έως το 1,095 για τη μεταβλητή C με βήμα 0,005. Η μεταβλητή C στον Logistic Regression αλγόριθμο έχει την ιδιότητα της "τακτοποίησης" (regularization), αλλά αντί οι μεγαλύτερες τιμές να δηλώνουν πιο αυστηρή τακτοποίηση, στην μεταβλητή C δηλώνει πιο χαλαρή. Η σχέση που τους συνδέει είναι $C = \frac{1}{\lambda}$.

Επειδή όμως η μεταβλητή C στον συγκεκριμένο αλγόριθμο δεν είναι αναγκαία για να μπορεί να λειτουργήσει όπως ήταν και στους δυο προηγούμενους αλγορίθμους. Ο συγκεκριμένος αλγόριθμος θα δοκιμαστεί ταυτόχρονα στον ίδιο τρόπο διαχωρισμού δεδομένων εκπαίδευσης και δοκιμής και τα αποτελέσματα από τις δύο περιπτώσεις με και χωρίς την μεταβλητή παραμετροποίησης θα συγκριθούν για την επιλογή της ιδανικότερης τιμής που θα πρέπει να έχει ο αλγόριθμος για να δίνει τα μεγαλύτερα ποσοστά ευστοχίας στις περισσότερες περιπτώσεις. Στις παρακάτω εικόνες παρουσιάζονται οι έξη διαφορετικές περιπτώσεις που εμφανίστηκαν κατά την περίοδο ταυτόχρονης δοκιμής. Με την πορτοκαλί γραμμή απεικονίζεται το ποσοστό ευστοχίας του αλγορίθμου χωρίς παραμετροποίηση, ενώ η μπλε το ποσοστό ευστοχίας με την μεταβλητή παραμετροποίησης (εικόνα *Εικόνα 51* και *Εικόνα 52*).



Εικόνα 51: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου LR στα R_filter δεδομένα 1.



Εικόνα 52: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου LR στα R_filter δεδομένα 2.

Η απόφαση της καλύτερης επιλογής θα γίνει σύμφωνα με την επαναληπτικότητα της κάθε περίπτωσης. Όπως φαίνεται και στην εικόνα 51, υπάρχουν τρεις διαφορετικές περιπτώσεις :

Α) Την περίπτωση και οι δύο αλγόριθμοι να έχουν τα ίδια ακριβώς αποτελέσματα. Σε αυτή την περίπτωση η γραφική παράσταση παρουσιάζεται ως μια ευθεία πορτοκαλί γραμμή, δηλαδή το σχήμα (c) στην *Εικόνα 51*.

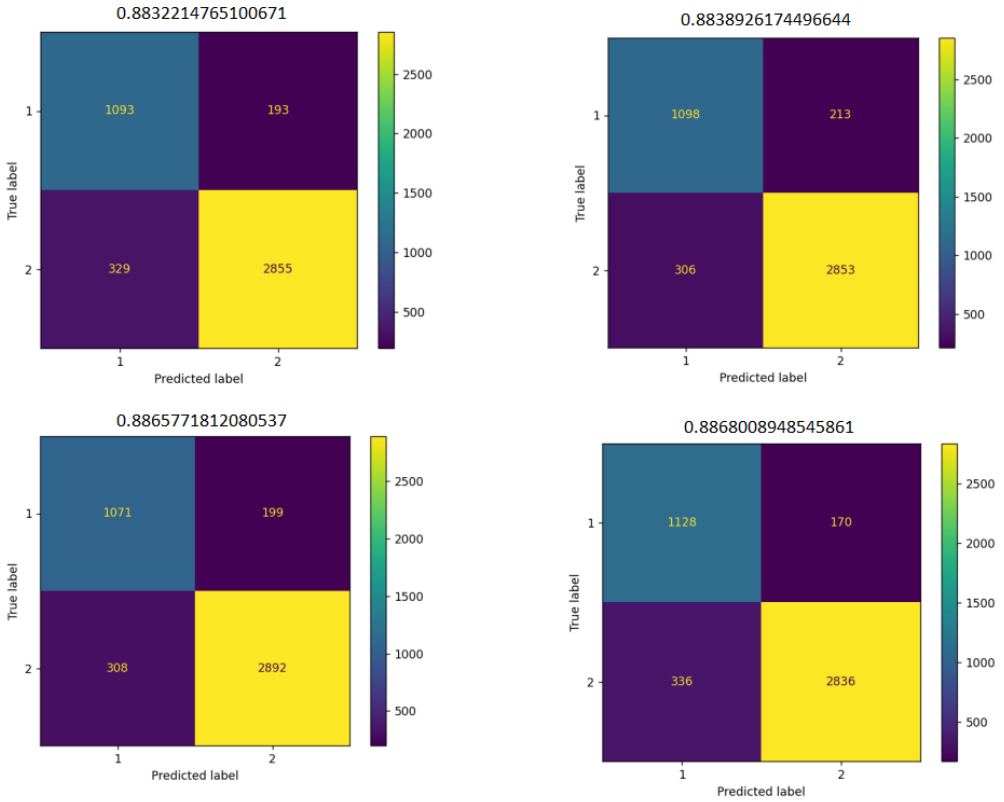
Β) Την περίπτωση ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης να έχει υψηλότερα ποσοστά ευστοχίας για κάποιες τιμές από τον αλγόριθμο χωρίς μεταβλητή παραμετροποίησης. Σε αυτή την περίπτωση έχουμε σχήματα όπως το (α) και (δ) της *Εικόνα 51*.

Γ) Την περίπτωση ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης να έχει χαμηλότερα ποσοστά ευστοχίας για κάποιες τιμές από τον αλγόριθμο χωρίς μεταβλητή παραμετροποίησης. Σε αυτή την περίπτωση έχουμε σχήματα όπως το (β), (ε) και (στ) της *Εικόνα 52*.

Η πρώτη περίπτωση εμφανίστηκε τρεις φορές από τις είκοσι δοκιμές που σημαίνει ότι η πιθανότητα να έχουμε την ίδια επίδοση και από τους δύο αλγορίθμους είναι 15%. Η δεύτερη περίπτωση εμφανίστηκε εννιά από τις είκοσι φορές, το οποίο μας δίνει ποσοστό εμφάνισης 45%. Και τέλος η τελευταία περίπτωση εμφανίστηκε οκτώ φορές. Άρα το ποσοστό εμφάνισης της τρίτης περίπτωσης είναι 40%.

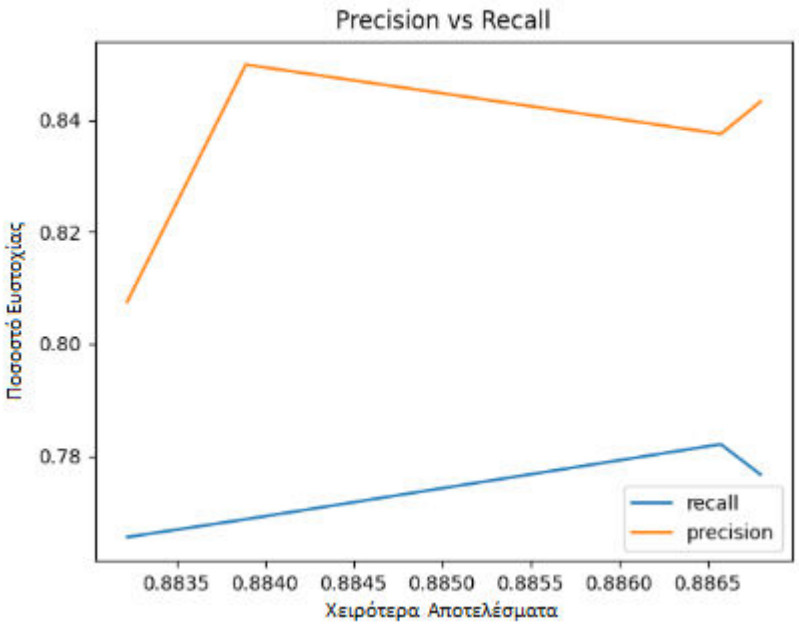
Από τα παραπάνω λοιπόν συμπεραίνεται, ότι για τον αλγόριθμο LogisticRegression στα δεδομένα R_filter, η παραμετροποίηση του αλγορίθμου έχει τις περισσότερες πιθανότητες μεγαλύτερης ακρίβειας. Όμως δεν είναι όλες οι τιμές οι ιδανικές. Η ιδανικότερη τιμή θα είναι ή η 1.015 ή 1.02 διότι είναι οι πιθανότητες που παρουσιάζουν καλύτερη απόδοση συχνότερα.

Στις εικόνες *Εικόνα 53* και *Εικόνα 54* παρουσιάζονται τα confusion matrix από τα τέσσερα χειρότερα ποσοστά ευστοχίας τους με τα ποσοστά ακριβώς πάνω από το καθένα και οι μεταβλητές precision και recall.



Εικόνα 53: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου LR στα δεδομένα R_filter.

Ο συγκεκριμένος αλγόριθμος σε σχέση με τον DecisionTreeClassifier εμφανίζει τα ίδια χαμηλά ποσοστά ευστοχίας με χαμηλότερο precision και υψηλότερο recall.



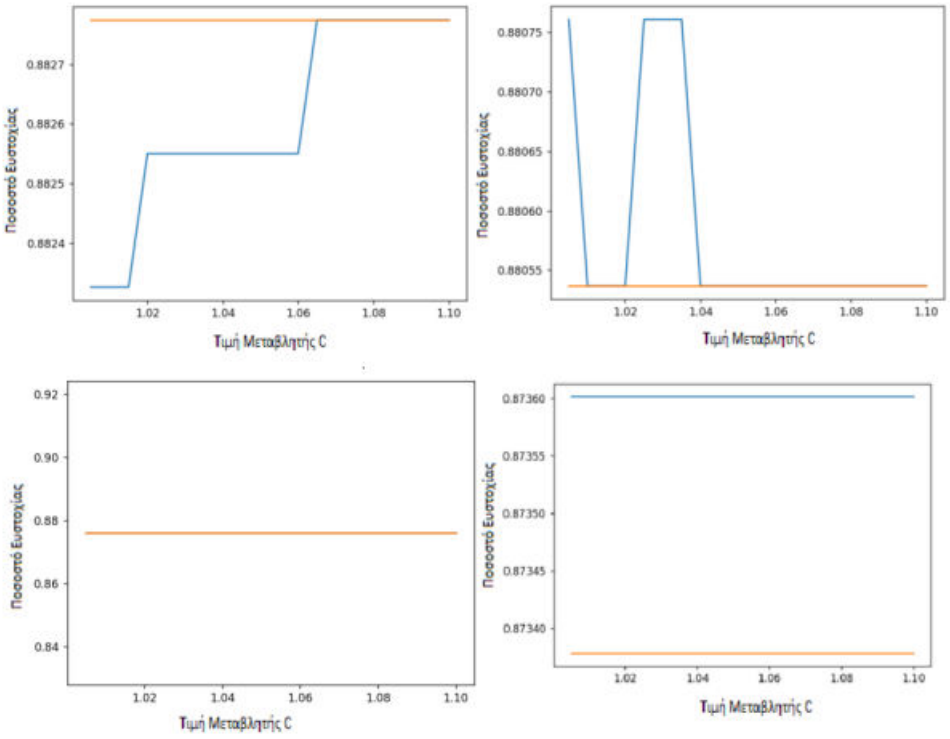
Εικόνα 54: Διάγραμμα Precision vs Recall του αλγορίθμου LR στα δεδομένα R_filter

Στη δεύτερη δοκιμή που έγινε στα δεδομένα R_filter_without_temp εφαρμόστηκε η ίδια ταυτόχρονη εξέταση του αλγορίθμου με παραμετροποίηση και χωρίς. Και σε αυτή τη δοκιμή εμφανίστηκαν τρεις διαφορετικές περιπτώσεις:

Α) Την περίπτωση και οι δύο αλγόριθμοι να έχουν τα ίδια ακριβώς αποτελέσματα. Σε αυτή την περίπτωση η γραφική παράσταση παρουσιάζεται ως μια ευθεία πορτοκαλί γραμμή, δηλαδή το σχήμα (c) στην *Εικόνα 55*.

Β) Την περίπτωση ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης να έχει υψηλότερα ποσοστά ευστοχίας για κάποιες τιμές από τον αλγόριθμο χωρίς μεταβλητή παραμετροποίησης. Σε αυτή την περίπτωση έχουμε σχήματα όπως το (β) και (δ) της *Εικόνα 55*.

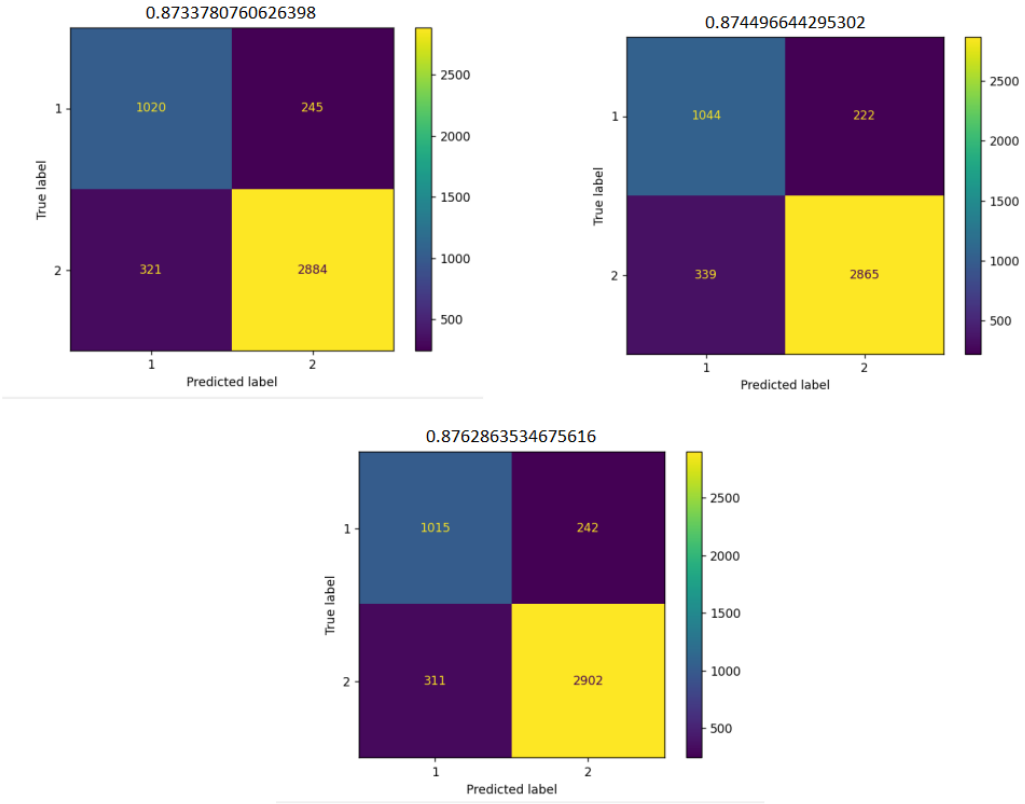
Γ) Την περίπτωση ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης να έχει χαμηλότερα ποσοστά ευστοχίας για κάποιες τιμές από τον αλγόριθμο χωρίς μεταβλητή παραμετροποίησης. Σε αυτή την περίπτωση έχουμε σχήματα όπως το (α) της *Εικόνα 55*.



Εικόνα 55: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου LR στα R_filter_no_temp δεδομένα.

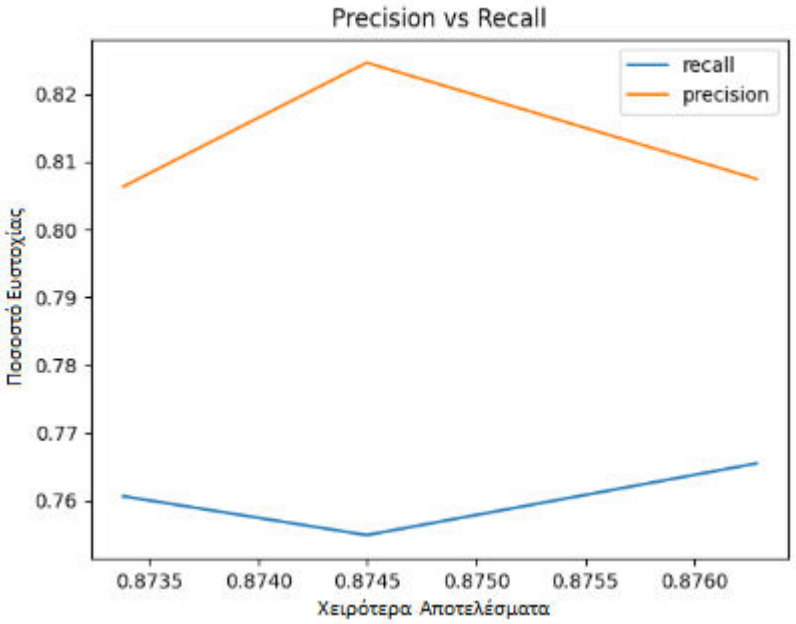
Η πρώτη περίπτωση όμως αυτή τη φορά εμφανίστηκε δώδεκα από τις είκοσι φορές, το οποίο μας δίνει πιθανότητα εμφάνισης 60%. Η δεύτερη περίπτωση

εμφανίστηκε μόνο τρεις φορές από τις είκοσι, δηλαδή πιθανότητα εμφάνισης 15%. Και τέλος η τρίτη περίπτωση είχε πιθανότητα εμφάνισης 25% με πέντε στα είκοσι. Άρα σύμφωνα με τα παραπάνω, παρατηρήθηκε ότι χωρίς να ληφθεί υπόψιν η θερμοκρασία, ο αλγόριθμος χωρίς παραμετροποίηση έχει την καλύτερη ευστοχία στις περισσότερες περιπτώσεις. Στις εικόνες Εικόνα 56 και Εικόνα *57* εμφανίζονται οι τρεις χειρότερες περιπτώσεις ευστοχίας με τα ποσοστά τους πάνω από τους πίνακες και οι μεταβλητές precision και recall.



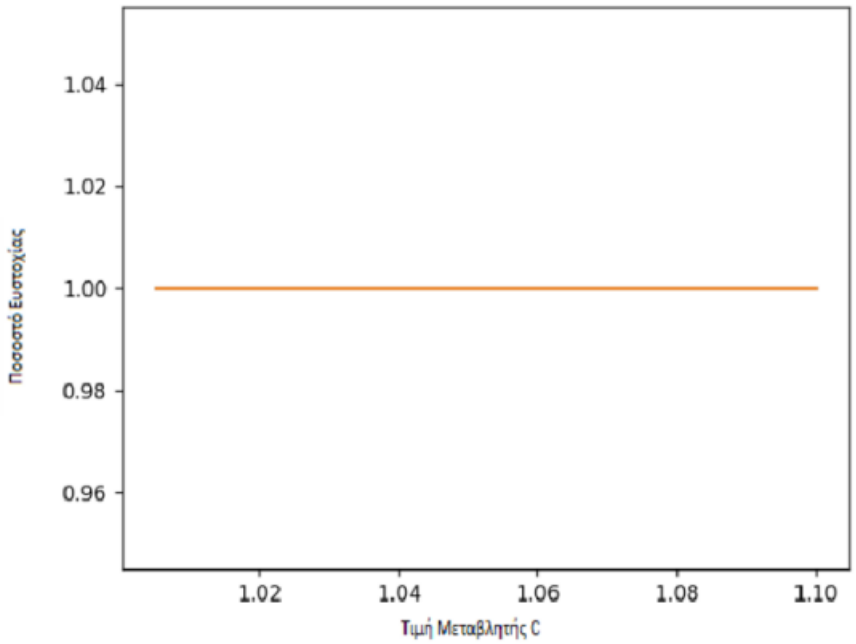
Εικόνα 56: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου LR στα δεδομένα R_filter_no_temp.

Παρατηρήθηκε ότι χωρίς να εκληφθεί υπόψιν η θερμοκρασία , ο αλγόριθμος κάνει λιγότερα λάθη στην πρόβλεψη των άρρωστων δέντρων σε σχέση με την πρώτη δοκιμή και περισσότερα λάθη στην πρόβλεψη των υγιή δέντρων. Παρόλα αυτά και σε αυτή τη δοκιμή η μεταβλητή recall είναι χαμηλότερη από την precision.



Εικόνα 57: Διάγραμμα Precision vs Recall του αλγορίθμου LR στα δεδομένα R_filter_no_temp.

Στη τρίτη δοκιμή που έγινε στα δεδομένα R_filter_balanced και αυτός το αλγόριθμος είχε ποσοστό ευστοχίας 100% (*Εικόνα 58*).



Εικόνα 58: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου LR στα R_filter_balanced δεδομένα.

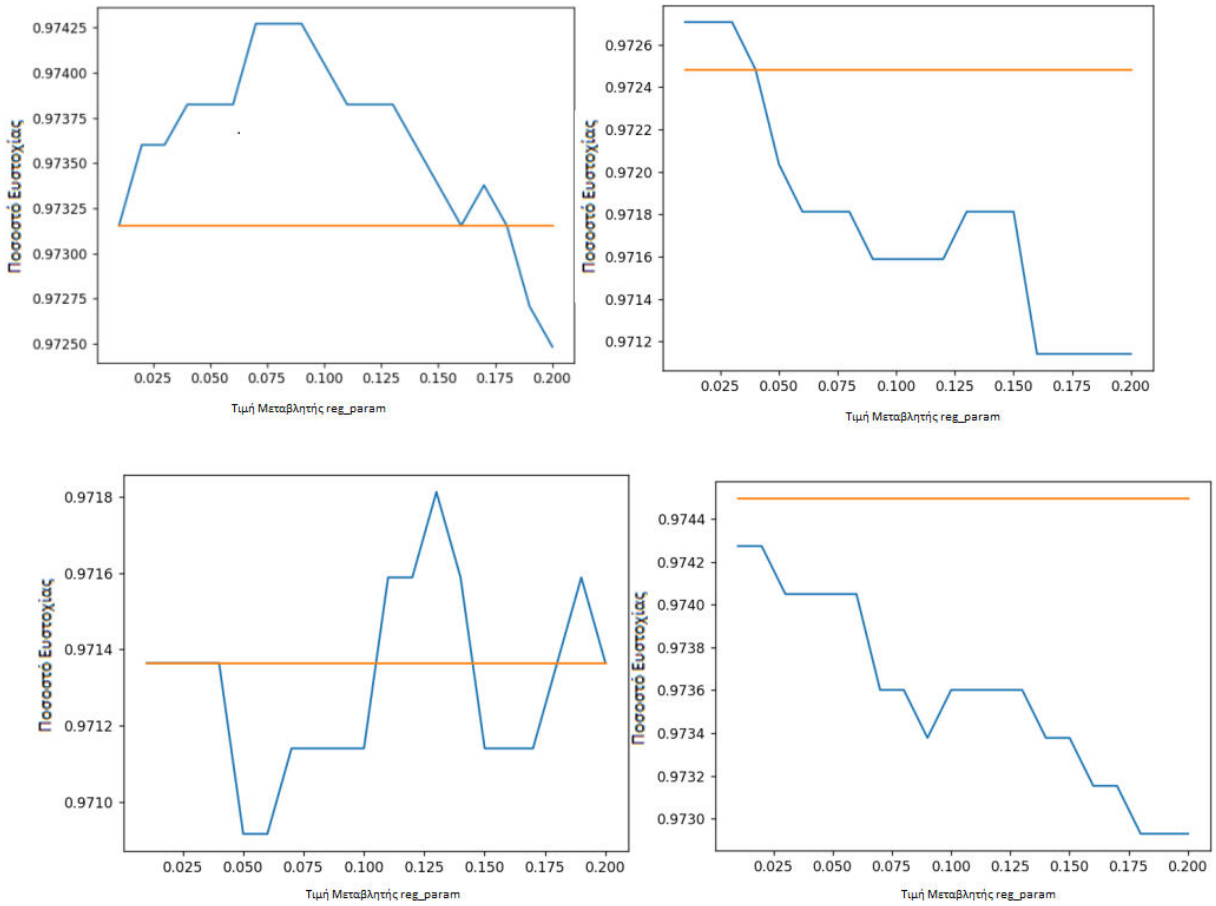
4.2.3 Αλγόριθμος Quadratic Discriminant Analysis

Στον Quadratic Discriminant Analysis αλγόριθμο δοκιμάστηκαν για την παραμετροποίηση του αλγορίθμου, τιμές από το 0.01 έως το 0.2 για τη μεταβλητή reg_param με βήμα 0,01. Η μεταβλητή reg_param στον Quadratic Discriminant Analysis αλγόριθμο έχει την μία πιο περίπλοκη ιδιότητα. Η reg_param είναι μέλος μιας συνάρτησης που ονομάζεται S2.

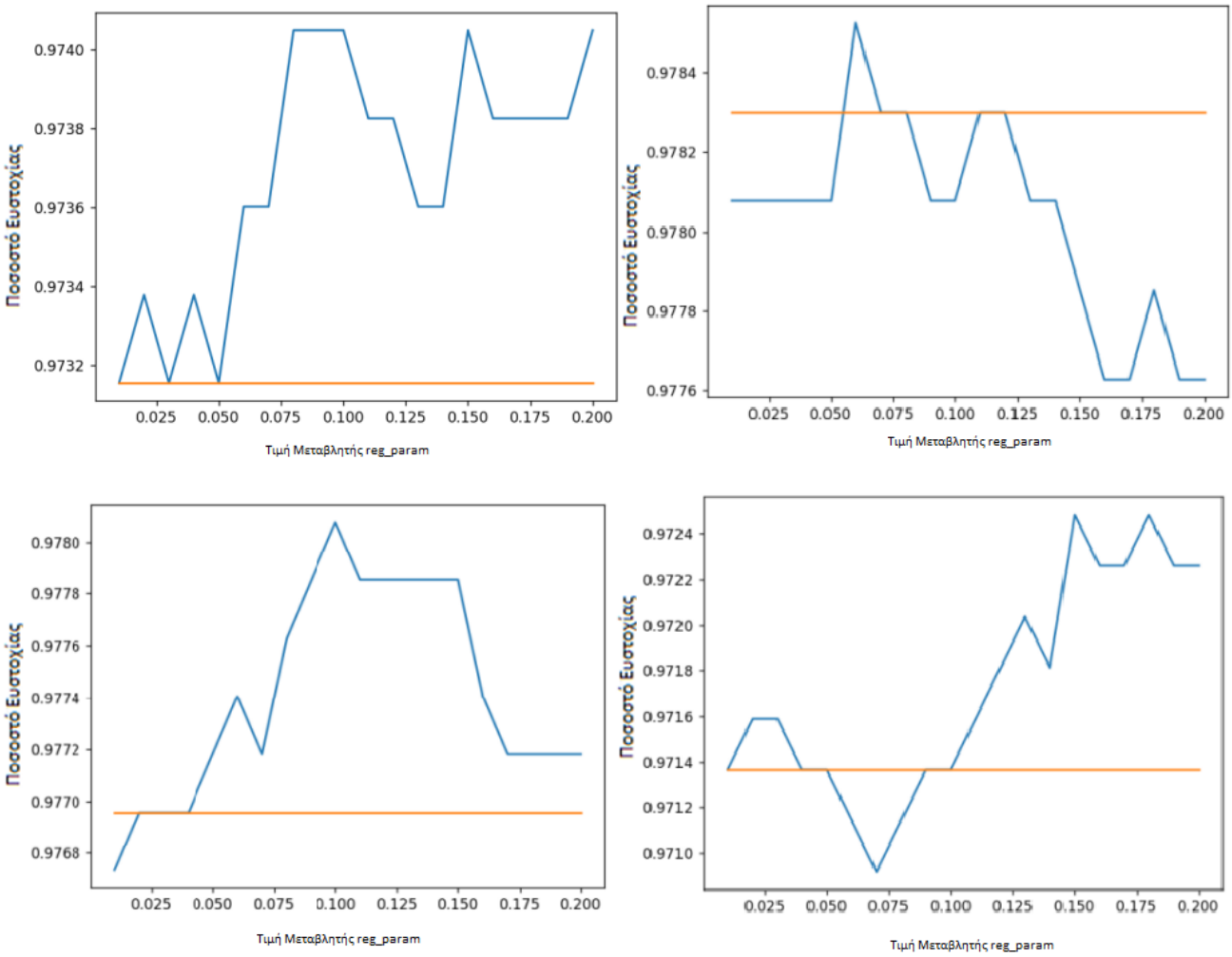
$$\mathbf{S2 = (1 - reg\_param) * S2 + reg\_param * np.eyes(n\_features)} \quad \textbf{(4.1)}$$

Η S2 συμβολίζει μια μεταβλητή του αλγορίθμου QDA που ονομάζεται scalings_.

Η αύξηση της reg_param μεταβλητής έχει ως αποτέλεσμα τη μείωση της μεταβλητής S2. Όπως και στον Logistic Regression η μεταβλητή που επιλέχθηκε για παραμετροποίηση του αλγορίθμου δεν χρειάζεται να είναι διάφορη του μηδενός για να λειτουργήσει ο αλγόριθμος. Για αυτό το λόγο θα δοκιμαστεί στον αλγόριθμο ταυτόχρονα ο ίδιος τρόπος διαχωρισμού δεδομένων εκπαίδευσης και δοκιμής και θα συγκριθούν τα αποτελέσματα των δύο περιπτώσεων με και χωρίς μεταβλητή παραμετροποίησης, όπως έγινε και στον προηγούμενο αλγόριθμο που αναφέρθηκε νωρίτερα. Στις εικόνες 59 και *Εικόνα 60* παρουσιάζονται οι οκτώ διαφορετικές περιπτώσεις που εμφανίστηκαν κατά την περίοδο ταυτόχρονης δοκιμής. Με την πορτοκαλί γραμμή παρουσιάζεται το ποσοστό ευστοχίας του αλγορίθμου χωρίς παραμετροποίηση, ενώ με το μπλε το ποσοστό ευστοχίας με την μεταβλητή παραμετροποίησης.



Εικόνα 59: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου QDA στα R_filter δεδομένα 1.



Εικόνα 60: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου QDA στα R_filter δεδομένα 2.

Για να αποφασιστεί στον αλγόριθμο Quadratic Discriminant Analysis αν χρειάζεται παραμετροποίηση ή όχι, χρειάζεται περισσότερη ανάλυση των περιπτώσεων του από ότι χρειάστηκε στον Logistic Regression. Για αυτό το λόγο τα αποτελέσματα που προέκυψαν από τον αλγόριθμο χωρίστηκαν σε οκτώ κατηγορίες:

Α) Στην πρώτη κατηγορία δίνεται η περίπτωση ο αλγόριθμος με την μεταβλητή παραμετροποίησης να ξεκινάει να έχει μεγαλύτερη ευστοχία από όταν δεν έχει μεταβλητή παραμετροποίησης για τις περισσότερες τιμές της μεταβλητής reg_param και καταλήγει από μια τιμή και μετά να έχει χαμηλότερη ευστοχία, όπως φαίνεται και στο διάγραμμα (α) στην *Εικόνα 59*.

Β) Στην δεύτερη κατηγορία δίνεται η περίπτωση ο αλγόριθμος με την

μεταβλητή παραμετροποίησης να ξεκινάει να έχει μεγαλύτερη ευστοχία από όταν δεν έχει μεταβλητή παραμετροποίησης για τις πρώτες τιμές της μεταβλητής reg_param και καταλήγει από μια τιμή και μετά να έχει χαμηλότερη ευστοχία, όπως φαίνεται και στο διάγραμμα (β) στην *Εικόνα 59*.

Γ) Στην τρίτη κατηγορία δίνεται η περίπτωση ο αλγόριθμος με την μεταβλητή παραμετροποίησης να έχει την ίδια ευστοχία με τον αλγόριθμο χωρίς μεταβλητή παραμετροποίησης. Ύστερα θα πρέπει να υπάρχουν μερικές τιμές για τις οποίες ο αλγόριθμος με τη μεταβλητή παραμετροποίησης να έχει χαμηλότερη ευστοχίας και μερικές τιμές που να έχει μεγαλύτερη ευστοχία από τον αλγόριθμο χωρίς μεταβλητή παραμετροποίησης, όπως φαίνεται και στο σχήμα (γ) της Εικόνα 59.

Δ) Στην τέταρτη περίπτωση δίνεται η ευστοχία του αλγορίθμου με την μεταβλητή παραμετροποίησης να είναι για όλες τις τιμές της μεταβλητής χαμηλότερη από αυτή του αλγορίθμου χωρίς την μεταβλητή. Αυτή η περίπτωση παρουσιάζεται στο διάγραμμα (δ) της *Εικόνα 59*.

Ε) Στην πέμπτη περίπτωση δίνεται η ευστοχία του αλγορίθμου με την μεταβλητή παραμετροποίησης να είναι για όλες τις τιμές της μεταβλητής υψηλότερη από αυτή του αλγορίθμου χωρίς την μεταβλητή. Αυτή η περίπτωση παρουσιάζεται στο διάγραμμα (ε) της *Εικόνα 60*.

ΣΤ) Στην έκτη περίπτωση δίνεται ο αλγόριθμος με την μεταβλητή παραμετροποίησης να ξεκινάει με χαμηλότερη ευστοχία από αυτή του αλγορίθμου χωρίς τη μεταβλητή παραμετροποίησης, να εμφανίζει υψηλότερη για λίγες τιμές της μεταβλητής και να ξανά επιστρέφει στην χαμηλότερη ευστοχία. Στο διάγραμμα (στ) της *Εικόνα 60* παρουσιάζεται αυτή η περίπτωση.

Ζ) Στην έβδομη περίπτωση δίνεται ο αλγόριθμος με την μεταβλητή παραμετροποίησης να ξεκινάει με χαμηλότερη ευστοχία από αυτή του αλγορίθμου χωρίς τη μεταβλητή παραμετροποίησης για τις αρχικές τιμές της μεταβλητής και καθώς υπάρχει αύξηση της μεταβλητής παραμετροποίησης να εμφανίζεται μεγαλύτερη ευστοχία από μία τιμή και μετά, όπως στο διάγραμμα (ζ) της *Εικόνα 60*.

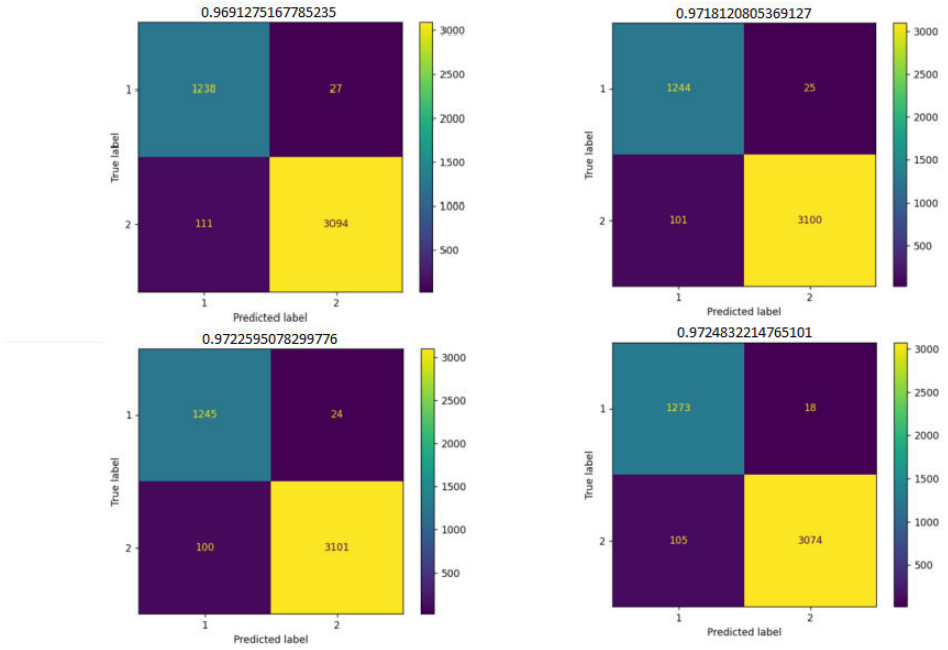
Η) Στην όγδοη και τελευταία περίπτωση δίνεται ο αλγόριθμος με την μεταβλητή παραμετροποίησης να ξεκινάει με υψηλότερη ευστοχία από αυτή του

αλγορίθμου χωρίς τη μεταβλητή παραμετροποίησης, να εμφανίζει χαμηλότερη για λίγες τιμές της μεταβλητής και να ξανά επιστρέφει στην υψηλότερη ευστοχία. Στο διάγραμμα (η) της *Εικόνα 60* παρουσιάζεται αυτή η περίπτωση.

Η πρώτη περίπτωση έχει ποσοστό εμφάνισης 25%, αφού εμφανίστηκε πέντε από τις είκοσι επαναλήψεις. Η δεύτερη και η πέμπτη περίπτωση εμφανίστηκαν τρεις από τις είκοσι φορές, άρα τα ποσοστά εμφάνισης τους είναι 15%. Η τρίτη περίπτωση με τέσσερις στις είκοσι φορές, έχει ποσοστό εμφάνισης 20%. Η έκτη περίπτωση έχει ποσοστό εμφάνισης 10% με δύο στις είκοσι φορές. Και τέλος η τέταρτη, έβδομη και όγδοη περίπτωση εμφανίστηκαν μόνο μια φορά με τα ποσοστά τους να είναι στο 5%.

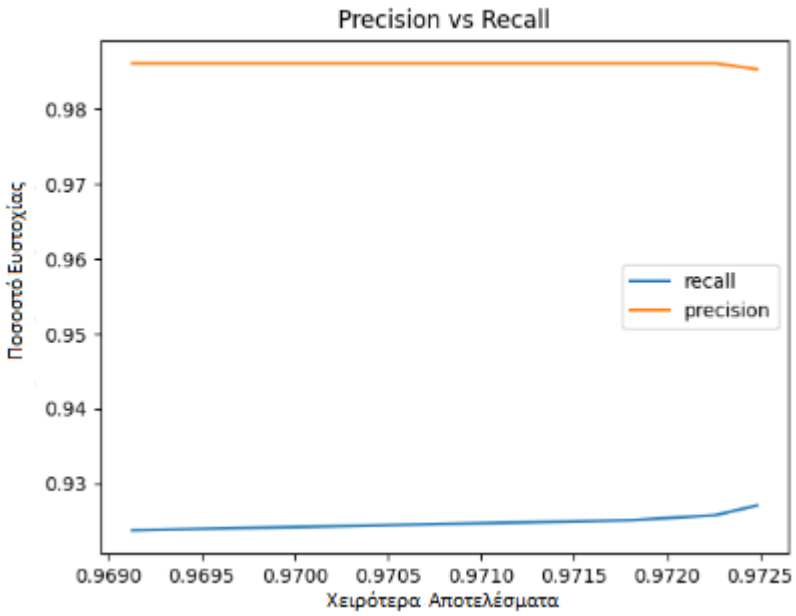
Σύμφωνα με τα διαγράμματα και τα ποσοστά εμφάνισης, συμπεραίνεται ότι ο συγκεκριμένος αλγόριθμος στην πρώτη δοκιμή έχει υψηλότερα ποσοστά ευστοχίας όταν χρησιμοποιείται η μεταβλητή παραμετροποίησης με τιμή 0,03. Διότι στις εννιά από τις είκοσι φορές ο αλγόριθμος με τιμή παραμετροποίησης 0,03 είχε υψηλότερη ευστοχία από όταν δεν είχε μεταβλητή παραμετροποίησης και 8 φορές που είχαν την ίδια ευστοχία. Δηλαδή ο αλγόριθμος με τιμή παραμετροποίησης 0,03 έχει ποσοστό πιθανότητας να έχει ίσο ή μεγαλύτερο ποσοστό ευστοχίας από όταν δεν χρησιμοποιεί τιμή παραμετροποίησης 85%.

Στις εικόνες Εικόνα 61 και Εικόνα *62* εμφανίζονται οι τρεις χειρότερες περιπτώσεις ευστοχίας με τα ποσοστά τους πάνω από τους πίνακες και οι μεταβλητές precision και recall.



Εικόνα 61: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου QDA στα δεδομένα R_filter.

Μέχρι στιγμής και αυτός ο αλγόριθμος έχει την τάση να κάνει περισσότερα λάθη όταν πρόκειται να προβλέψει αν ένα δέντρο είναι άρρωστο. Ενώ βρίσκει πάνω από το 98% των περιπτώσεων όταν πρόκειται για ένα υγιή δέντρο.



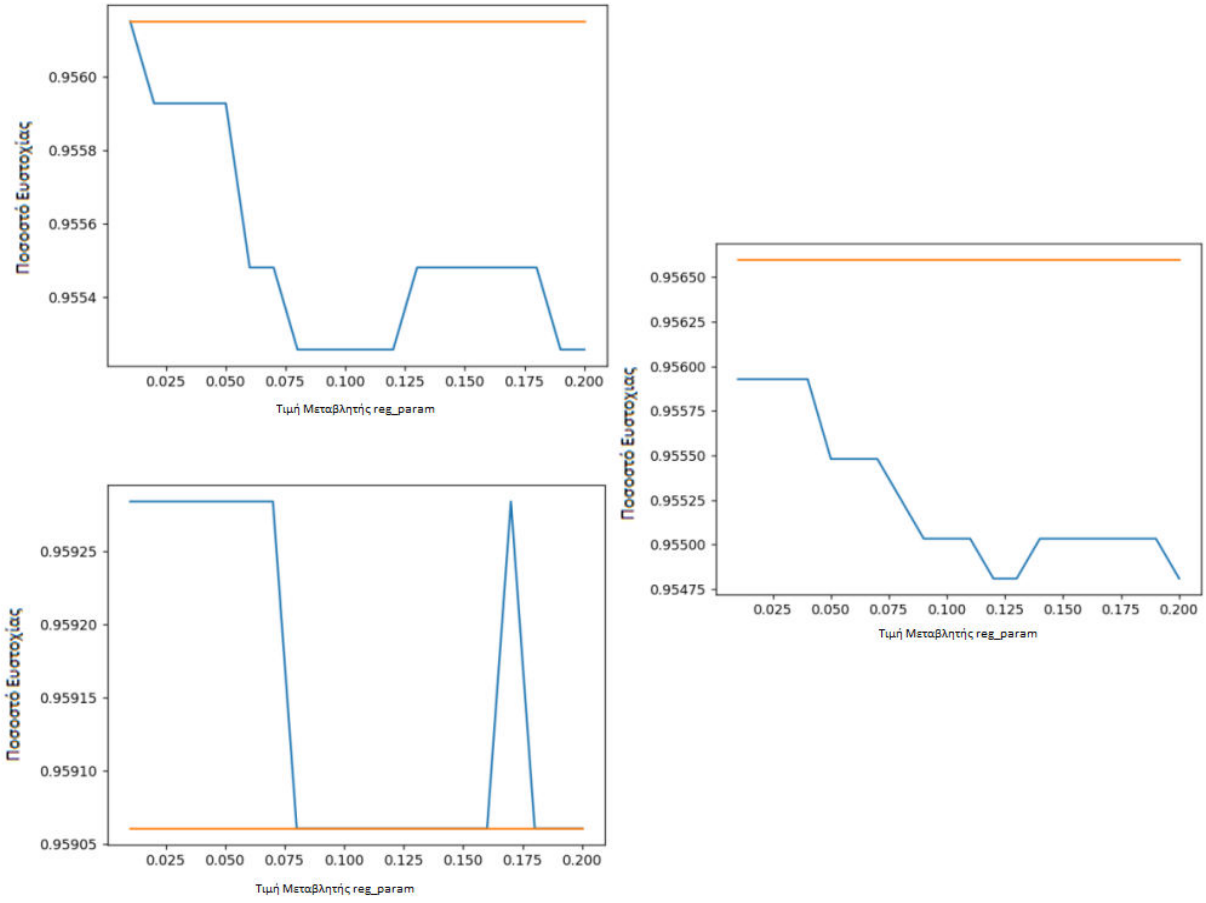
Εικόνα 62: Διάγραμμα Precision vs Recall του αλγορίθμου QDA στα δεδομένα R_filter.

Στην δεύτερη δοκιμή που έγινε στα R_filter_without_temp δεδομένα οι περιπτώσεις που εμφανίστηκαν ήταν μόνο τρεις. Οι οποίες είναι οι εξής:

Α) Ο αλγόριθμος με την μεταβλητή παραμετροποίησης να έχει για την πρώτη τιμή της μεταβλητής το ίδιο ποσοστό ευστοχίας, με τον αλγόριθμο δίχως την μεταβλητή παραμετροποίησης και για όλες τις υπόλοιπες τιμές να έχει χαμηλότερη. Το αντίστοιχο παράδειγμα, παρουσιάζεται στην *Εικόνα 63* διάγραμμα (α).

Β) Ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης να έχει μεγαλύτερη ή ίση ευστοχία με εκείνον δίχως μεταβλητή για όλες τις τιμές της μεταβλητής, όπως φαίνεται στο διάγραμμα (β) της *Εικόνα 63*.

Γ) Ο αλγόριθμος με την μεταβλητή παραμετροποίησης να έχει για όλες τις τιμές της μεταβλητής χαμηλότερη ευστοχία από αυτόν δίχως την μεταβλητή παραμετροποίησης. Ένα τέτοιο παράδειγμα παρουσιάζεται στο διάγραμμα (γ) της *Εικόνα 63*.

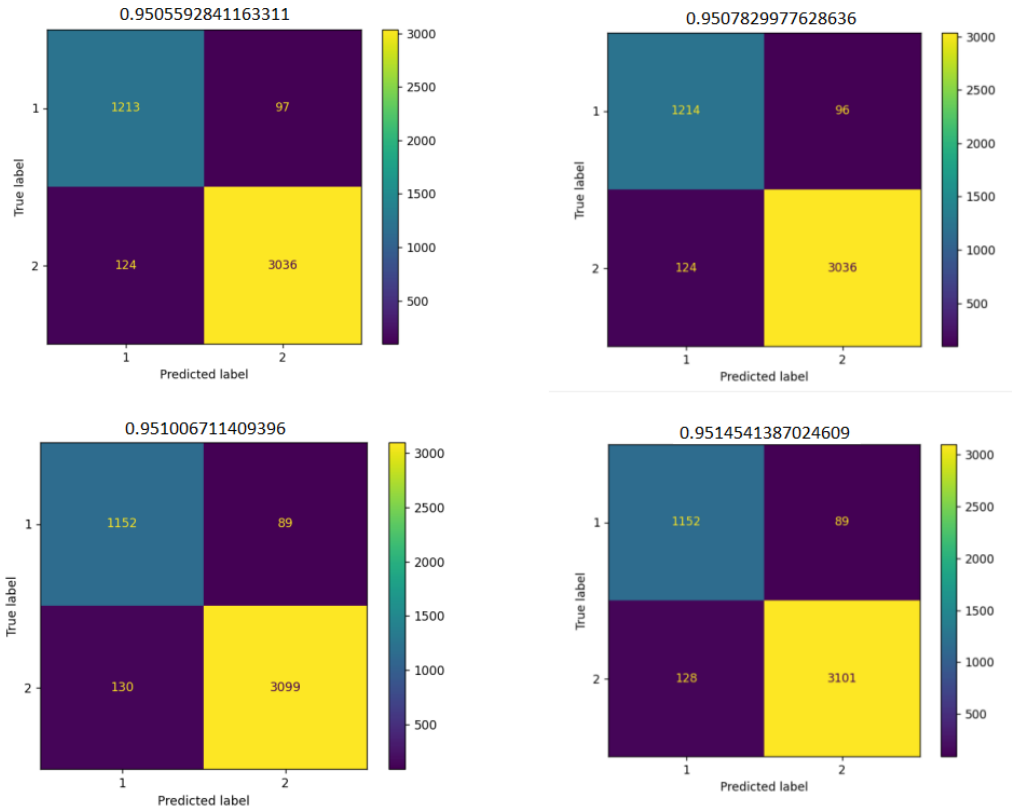


Εικόνα 63: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου QDA στα R_filter_no_temp δεδομένα

Η πρώτη περίπτωση εμφανίστηκε τις δεκατρείς από τις είκοσι φορές, το οποίο μας δίνει ποσοστό εμφάνισης 65%. Η τρίτη περίπτωση είχε ποσοστό εμφάνισης 30%, δηλαδή εμφανίστηκε έξη από τις είκοσι φορές. Και τέλος η δεύτερη περίπτωση εμφανίστηκε μόνο μία φορά από τις είκοσι και έχει ποσοστό εμφάνισης 5%.

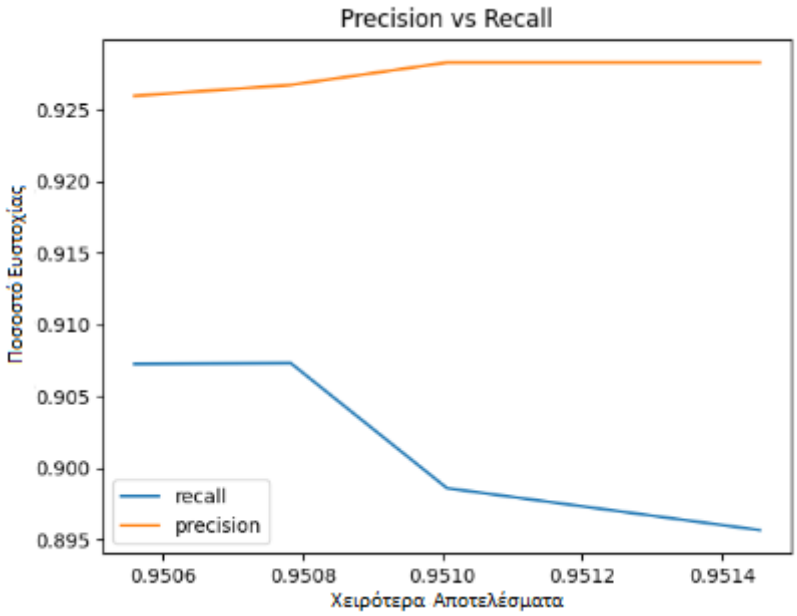
Από τα διαγράμματα και τα ποσοστά εμφάνισης , αντιλαμβάνεται κανείς ότι δίχως να ληφθεί υπόψιν η θερμοκρασία, ο αλγόριθμος δίχως μεταβλητή παραμετροποίησης έχει υψηλότερα ποσοστά ευστοχίας.

Στις εικόνες *Εικόνα 64* και *Εικόνα 65* εμφανίζονται οι τρεις χειρότερες περιπτώσεις ευστοχίας με τα ποσοστά της κάθε μίας πάνω από τους αντίστοιχους πίνακες και οι μεταβλητές precision και recall.



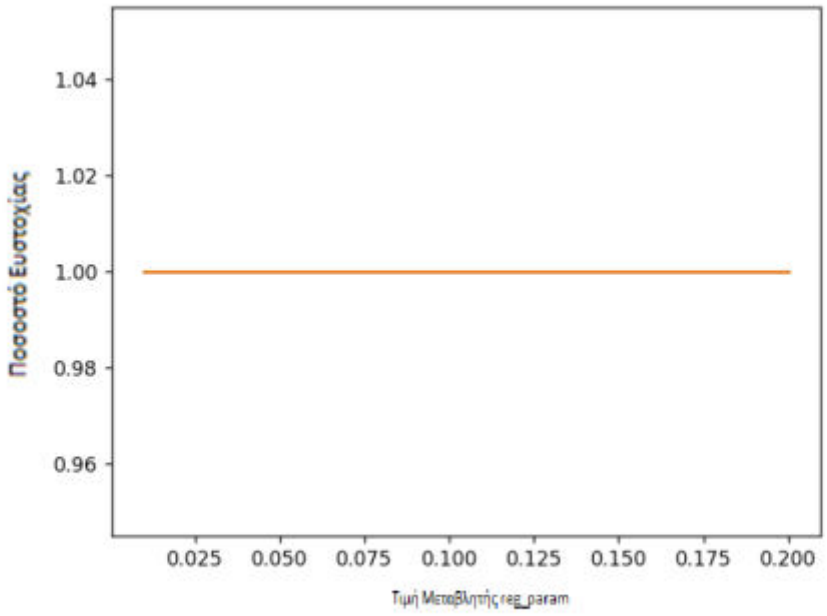
Εικόνα 64: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου QDA στα δεδομένα R_filter_no_temp.

Σύμφωνα με τις εικόνες *Εικόνα 64* και *Εικόνα 65* , μπορεί κανείς να αντιληφθεί ότι δίχως να εκληφθεί υπόψιν η θερμοκρασία ο αλγόριθμος δυσκολεύεται αρκετά περισσότερο να αντιληφθεί αν ένα δέντρο είναι υγιές αφού η μεταβλητή precision έχει πέσει στο 0.92 από το 0.97 - 0.98 που βρισκόταν όταν του δινόταν και τα δεδομένα θερμοκρασίας. Η μεταβλητή recall μειώνεται και εκείνη εξίσου, αλλά η μείωση της είναι στο 0.01, ενώ της precision στο 0.05 με 0.06.



Εικόνα 65: Διάγραμμα Precision vs Recall του αλγορίθμου QDA στα δεδομένα R_filter_no_temp.

Στη τρίτη δοκιμή που έγινε στα δεδομένα R_filter_balanced και αυτός ο αλγόριθμος είχε ποσοστό ευστοχίας 100% είτε με μεταβλητή παραμετροποίησης είτε χωρίς όπως φαίνεται στην *Εικόνα 66*.

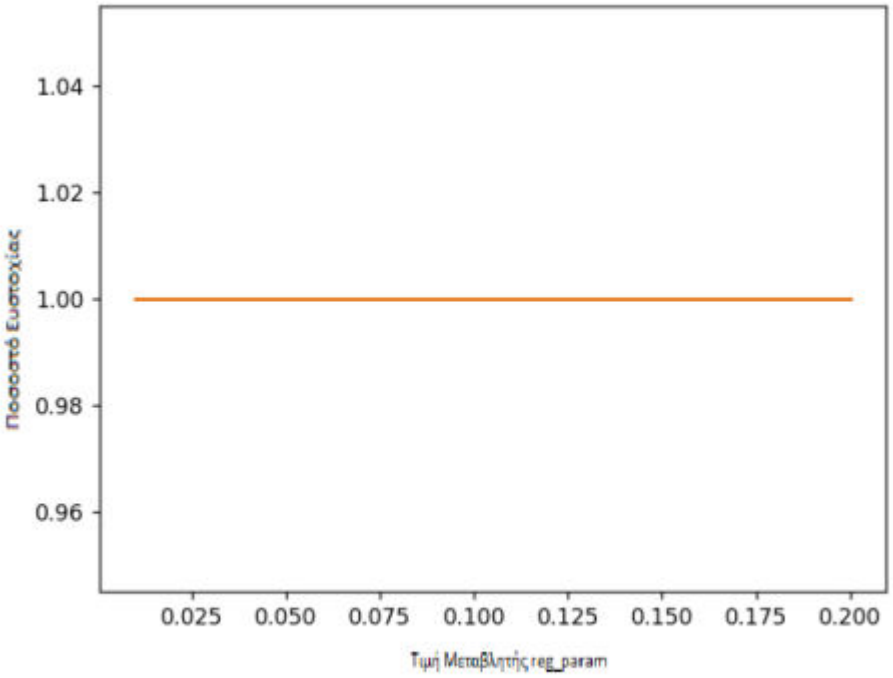


Εικόνα 66: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου QDA στα R_filter_balanced δεδομένα.

4.2.4 Αλγόριθμος Stochastic Gradient Descent Classifier

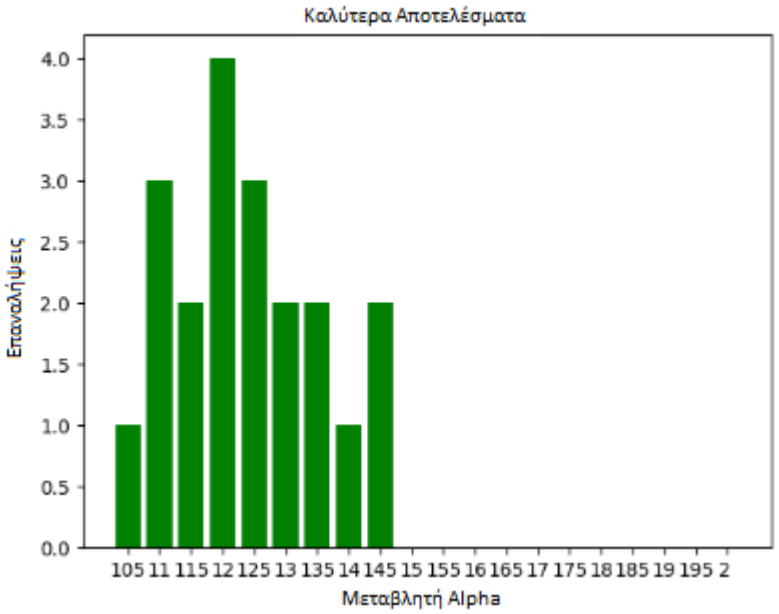
Ο αλγόριθμος Stochastic Gradient Descent Classifier ανήκει στην κατηγορία των stochastic gradient descent αλγόριθμων. Στον συγκεκριμένο αλγόριθμο δοκιμάστηκε για την παραμετροποίηση του αλγορίθμου, τιμές από το 0.01 έως το 0.02 για τη μεταβλητή alpha με βήμα 0,0005. Η μεταβλητή alpha στον Stochastic Gradient Descent Classifier αλγόριθμο έχει την ιδιότητα να πολλαπλασιάζει την "τακτοποίησης¨ (regularization). Όσο αυξάνεται η μεταβλητή alpha, τόσο πιο ¨αυστηρή" γίνεται η τακτοποίηση. Και σε αυτόν τον αλγόριθμο όπως και στους δύο προηγούμενους αλγορίθμους που αναλύθηκαν, η μεταβλητή που επιλέχθηκε για παραμετροποίηση του αλγορίθμου δεν χρειάζεται να είναι διάφορη του μηδενός για να λειτουργήσει ο αλγόριθμος. Για αυτό το λόγο θα δοκιμασθεί ο αλγόριθμος ταυτόχρονα με τον ίδιο τρόπο διαχωρισμού δεδομένων εκπαίδευσης και δοκιμής όπως έγινε και στους δύο προηγούμενους αλγορίθμους και θα συγκριθούν τα αποτελέσματα του αλγορίθμου και των δύο περιπτώσεων. Στην *Εικόνα 67* που ακολουθεί παρουσιάζονται δύο από τις είκοσι περιπτώσεις, όπου και στις δύο περιπτώσεις ο αλγόριθμος με μεταβλητή παραμετροποίησης

έχει μεγαλύτερη ευστοχία από αυτόν χωρίς. Γενικά και στις είκοσι περιπτώσεις ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης έχει μεγαλύτερη ευστοχία από όταν δεν χρησιμοποιεί κάποια μεταβλητή.



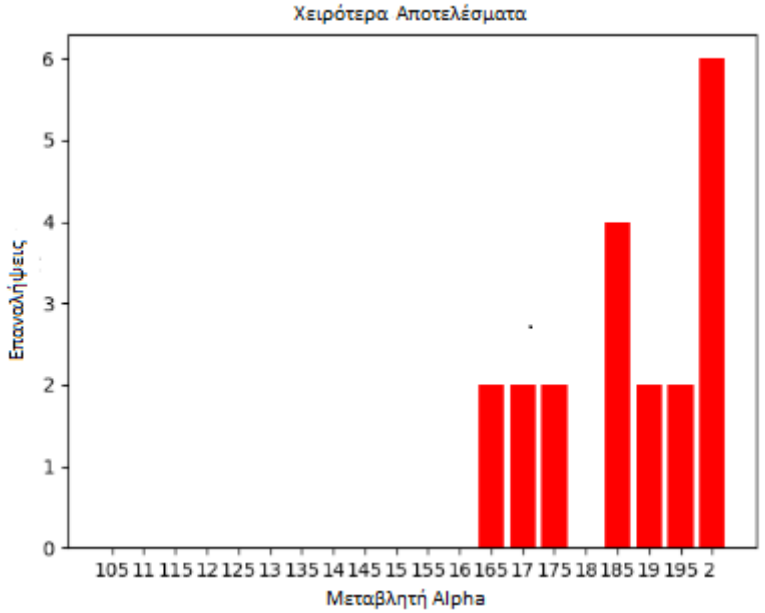
Εικόνα 67: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου SGD στα R_filter δεδομένα.

Στις εικόνες 68 και 69 παρουσιάζονται οι συχνότητες της βέλτιστης και χειρότερης ευστοχίας των τιμών της μεταβλητής παραμετροποίησης. Στην *Εικόνα 68* παρατηρήθηκε ότι τα βέλτιστα ποσοστά ευστοχίας τα κάνανε οι πρώτες εννιά τιμές με την τιμή 0,012 να έχει τις περισσότερες επαναλήψεις, που στην συγκεκριμένη περίπτωση είναι τέσσερα. Στην *Εικόνα 69*, παρουσιάζονται οι συχνότητες των χειρότερων αποτελεσμάτων ευστοχίας από τις τιμές της μεταβλητής. Παρατηρήθηκε λοιπόν, ότι οι οκτώ τελευταίες τιμές εκτός του 0,018, εμφάνισαν τα χειρότερα αποτελέσματα από όλες τις τιμές , με την τιμή 0,02 να είχε έξη φορές το χειρότερο αποτέλεσμα.



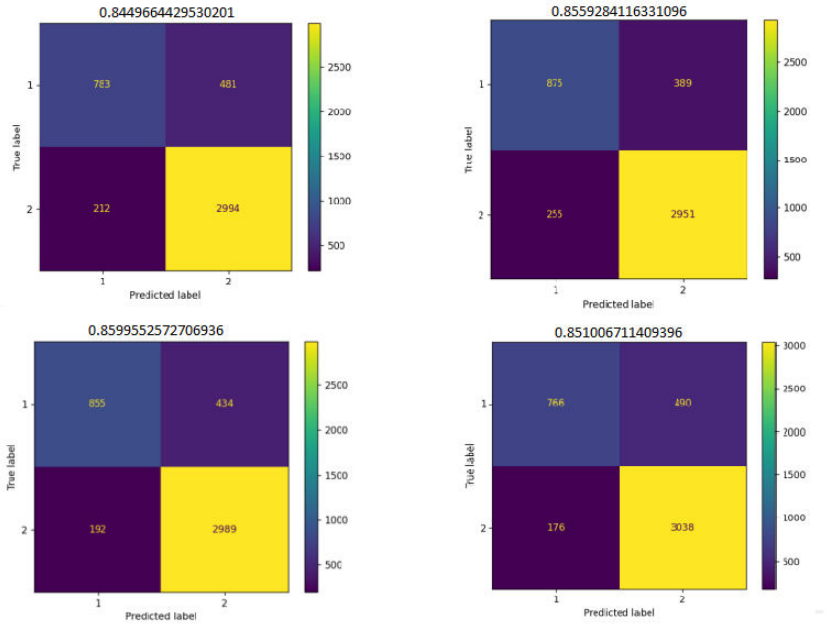
Εικόνα 68: Διάγραμμα Καλύτερων Αποτελεσμάτων στα δεδομένα R_filter του SGD αλγορίθμου.

Σύμφωνα με τις εικόνες 68 και 69, η τιμή που επιλέχθηκε για την μεταβλητή παραμετροποίησης του αλγορίθμου είναι η 0,012.



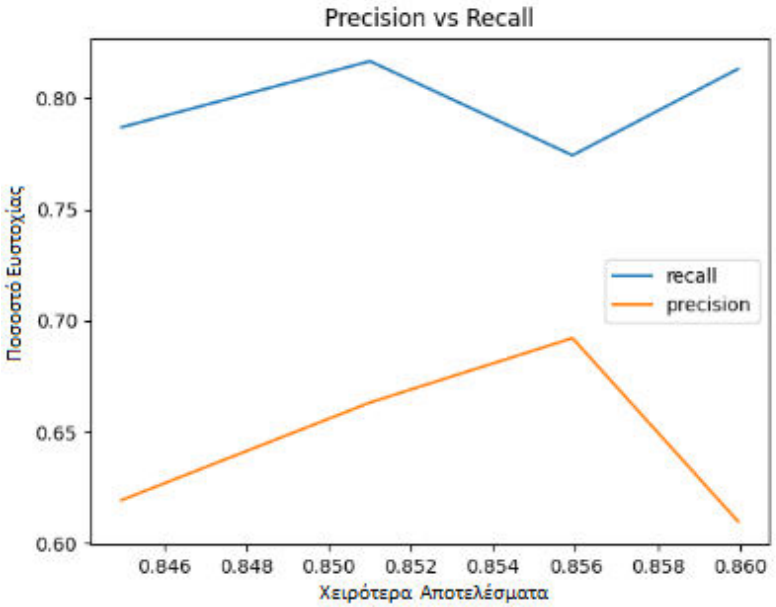
Εικόνα 69: Διάγραμμα Χειρότερων Αποτελεσμάτων στα δεδομένα R_filter του SGD αλγορίθμου.

Στις εικόνες Εικόνα 70 και Εικόνα *71* εμφανίζονται τα τέσσερα χειρότερα αποτελέσματα ευστοχίας με το ποσοστό ευστοχίας τους πάνω από τα matrix και οι μεταβλητές precision και recall.



Εικόνα 70: Τα Confusion Matrix Των Χειρότερων Αποτελεσμάτων του αλγορίθμου SGD στα δεδομένα R_filter.

Από τις εικόνες Εικόνα 70 και Εικόνα *71* παρατηρήθηκε ότι, ο αλγόριθμος Stochastic Gradient Descent , είναι ο μοναδικός που κάνει περισσότερα λάθη στις περιπτώσεις των υγιή δέντρων. Η μεταβλητή precision είναι πάρα πολύ χαμηλή σε σχέση με όλους τους προηγούμενους αλγορίθμους που μελετήθηκαν, όπως φάνηκε και στα confusion matrix της Εικόνα 70, οι λανθασμένες προβλέψεις των υγιή δέντρωνείναι το ένα τρίτο και παραπάνω από τον συνολικό αριθμό προβλέψεων.



Εικόνα 71: Διάγραμμα Precision vs Recall του αλγορίθμου SGD στα δεδομένα R_filter.

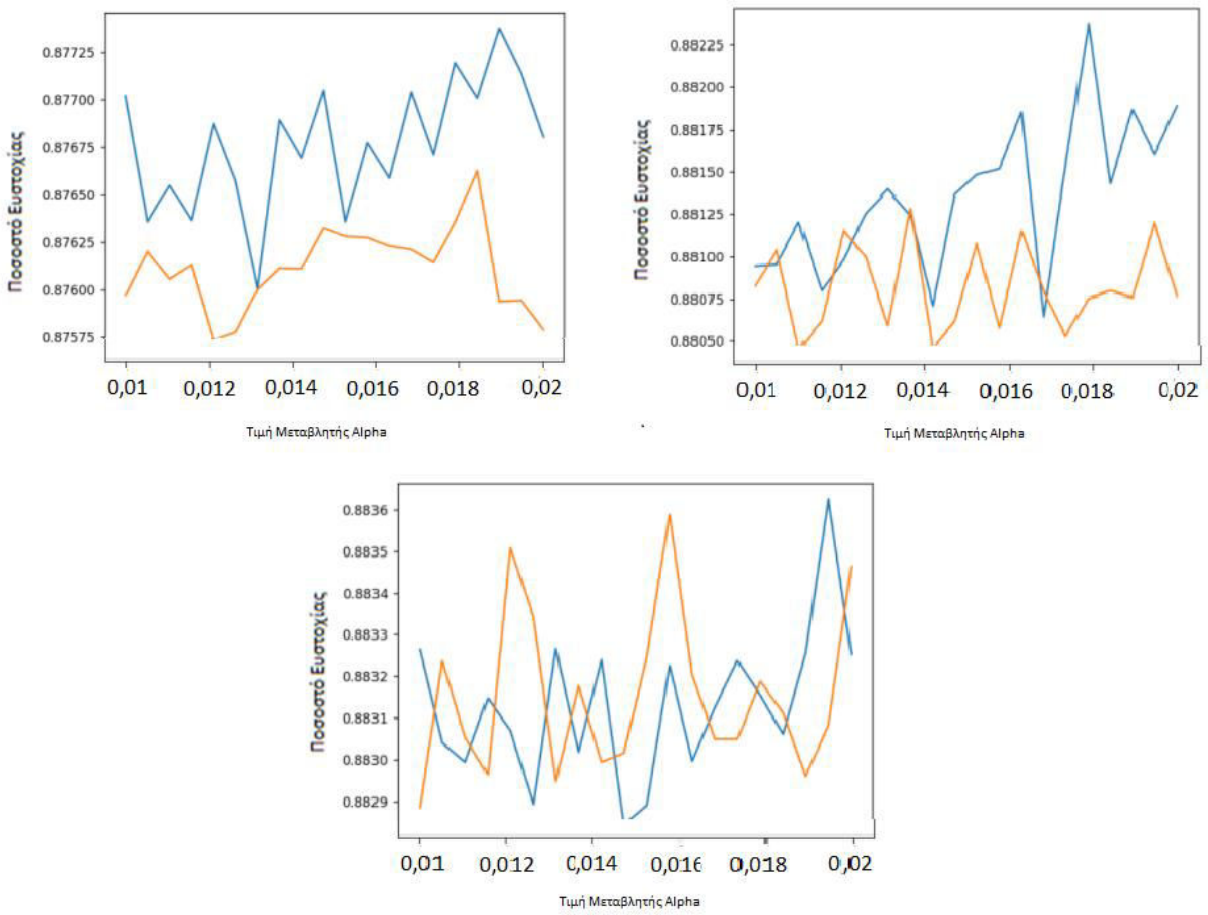
Στην δεύτερη δοκιμή που έγιναν με τα δεδομένα R_filter_without_temp ,τα αποτελέσματα χωρίστηκαν σε τρεις περιπτώσεις:

Α) Ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης να έχει μεγαλύτερη ευστοχία για όλες τις τιμές της μεταβλητής σε σχέση με εκείνον που δεν χρησιμοποιεί μεταβλητή. Στην *Εικόνα 72*, σχήμα (α) παρουσιάζεται το σχετικό παράδειγμα.

Β) Ο αλγόριθμος που χρησιμοποιεί μεταβλητή παραμετροποίησης να έχει και πάλι μεγαλύτερη ευστοχία από εκείνον χωρίς μεταβλητή παραμετροποίησης, αλλά να υπάρχουν κάποιες τιμές της μεταβλητής που να εμφανίζει χαμηλότερη από εκείνον χωρίς, όπως στο διάγραμμα (β) της *Εικόνα 72*.

Γ) Σαν τρίτη περίπτωση μεγάλη αστοχία του αλγορίθμου, διότι δεν υπάρχει ακριβώς κάποιος αλγόριθμος που να υπερισχύει, αλλά για κάποιες τιμές ο αλγόριθμος με μεταβλητή παραμετροποίηση έχει μεγαλύτερη ευστοχία και για κάποιες εκείνος δίχως μεταβλητή. Ένα τέτοιο παράδειγμα παρατηρείται στην εικόνα [72] του διαγράμματος (γ).

Η πρώτη περίπτωση εμφανίστηκε εφτά στις είκοσι φορές , το οποία μας δίνει πιθανότητα εμφάνισης 35%. Η δεύτερη περίπτωση έχει πιθανότητα εμφάνισης 40%, αφού εμφανίστηκε οκτώ στις είκοσι φορές. Και η τρίτη και τελευταία περίπτωση εμφανίστηκε πέντε στις είκοσι φορές και μας δίνει πιθανότητα εμφάνισης 25%.

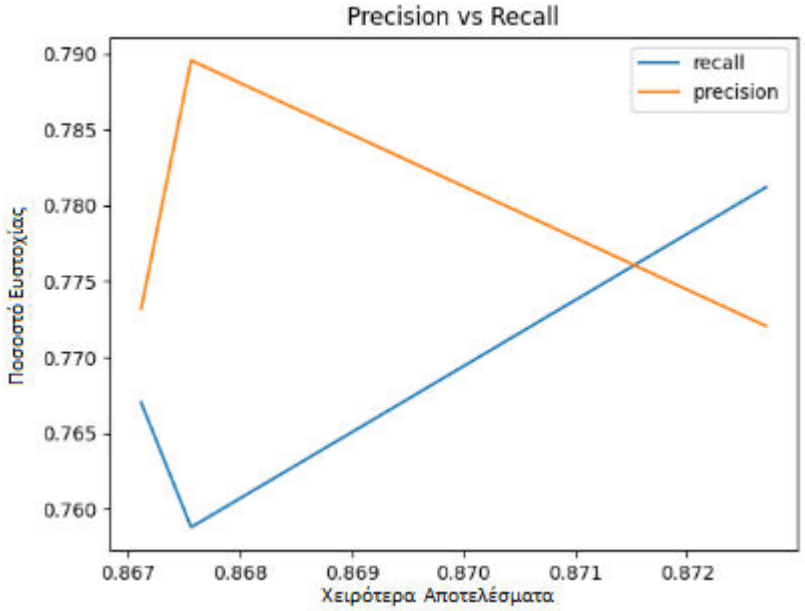


Εικόνα 72: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου SGD στα R_filter_no_temp δεδομένα.

Σύμφωνα με τα ποσοστά και διαγράμματα που μελετήθηκαν, συμπεραίνεται ότι η καλύτερη επιλογή για να μας δίνει ο αλγόριθμος Stochastic Gradient Descent το καλύτερο δυνατό αποτέλεσμα, είναι να χρησιμοποιηθεί μεταβλητή βελτιστοποίησης με τιμή 0,19, διότι σε όλα τα αποτελέσματα που δέχθηκε η μεταβλητή με τιμή 0,19 ήταν εκείνη που έδωσε σε όλες τις περιπτώσεις υψηλότερη ευστοχία από τον αλγόριθμο δίχως μεταβλητή παραμετροποίησης. Στις εικόνες *Εικόνα 73* και *Εικόνα 74* εμφανίζονται οι τρεις χειρότερες περιπτώσεις ευστοχίας και οι μεταβλητές precision και recall.

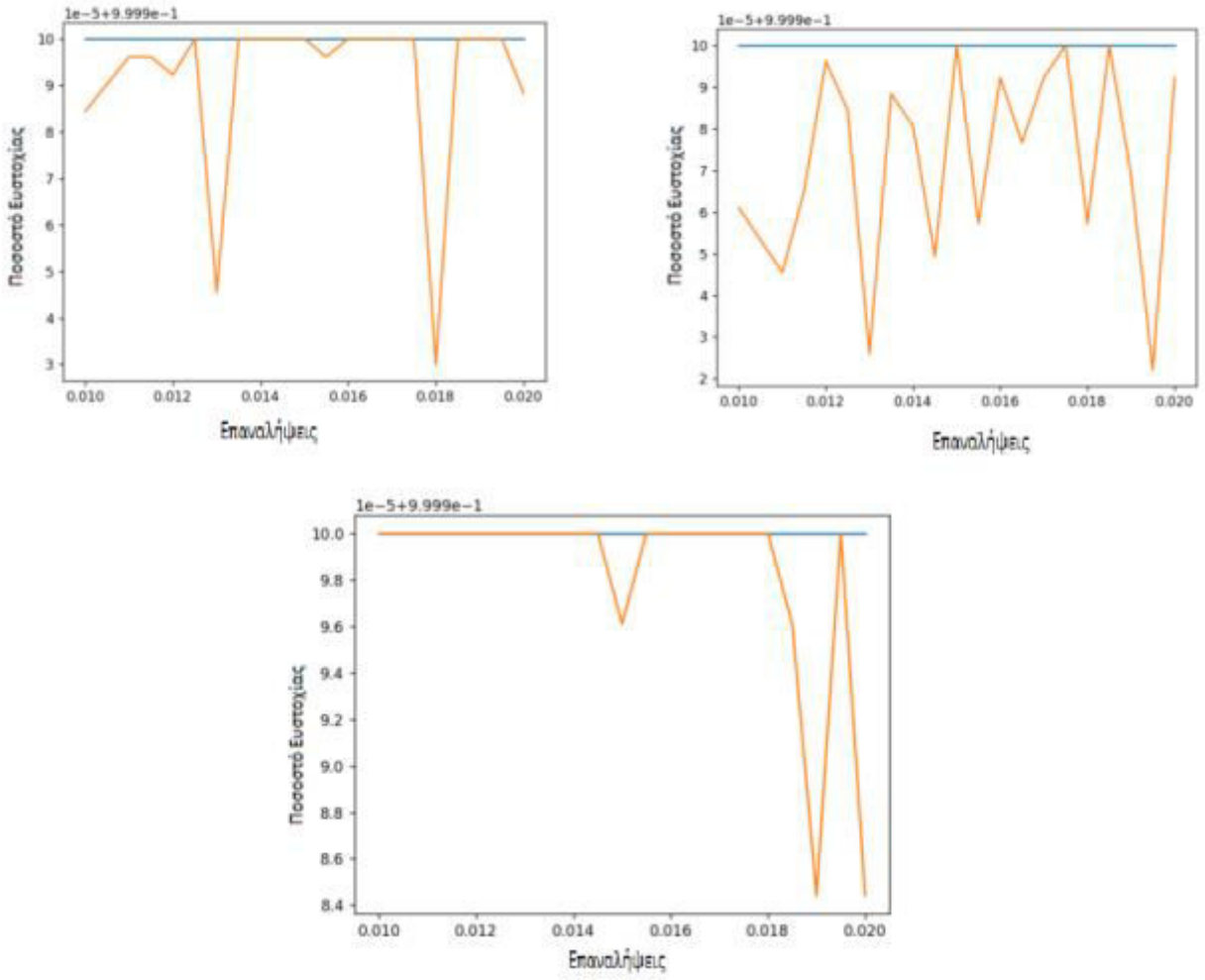
Από τα confusion matrix του αλγορίθμου και τις μεταβλητές precision και recall, παρατηρούμε ότι ο συγκεκριμένος αλγόριθμος είναι ο μοναδικός αλγόριθμος που, όταν δεν λαμβάνει υπόψιν του τα δεδομένα θερμοκρασίας τα χειρότερα αποτελέσματα ευστοχίας του είναι καλύτερα από όταν υπολογίζει και

με τη θερμοκρασία. Η μεταβλητή precision κάνει μια πολύ μεγάλη βελτίωση από το 0,65 περίπου στο 0,775 και ενώ υπάρχει μείωση της μεταβλητής recall, στο γενικό σύνολο υπάρχει βελτίωση των χειρότερων αποτελεσμάτων ευστοχίας.



Εικόνα 73: Διάγραμμα Precision vs Recall του αλγορίθμου SGD στα δεδομένα R_filter_no_temp.

Στη τρίτη δοκιμή που διεξάχθηκε στα δεδομένα R_filter_balanced ο αλγόριθμος Stochastic Gradient Descent είναι ο μοναδικός αλγόριθμος από όσους αναλύσαμε, που δεν εμφανίζει ποσοστό ευστοχίας 100% και στις δύο περιπτώσεις. Στην *Εικόνα 74* μπορεί να παρατηρηθεί η ασυνέπεια του αλγορίθμου δίχως μεταβλητή, όσο αναφορά τα τέλεια ποσοστά ευστοχίας.

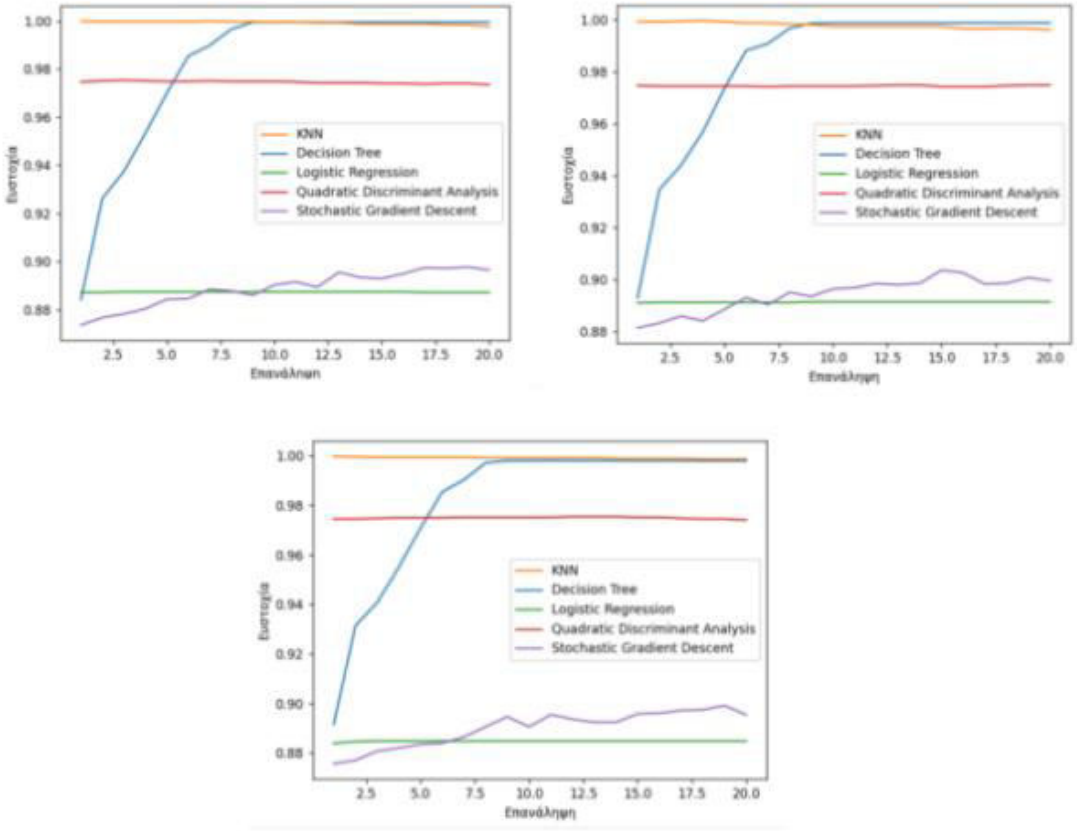


Εικόνα 74: Διάγραμμα Ποσοστού Ευστοχίας του Αλγορίθμου SGD στα R_filter_balanced δεδομένα.

4.2.5 Σύγκριση Όλων των Αλγορίθμων

Τέλος η τελευταία σύγκριση έγινε ανάμεσα σε όλους του αλγορίθμους, με σκοπό την επιλογή του ιδανικότερου αλγορίθμου σε κάθε περίσταση. Η διαδικασία ελέγχου ακολουθεί την ίδια λογική και των προηγούμενων δοκιμών. Δηλαδή η πρώτη σύγκριση έγινε στα δεδομένα R_filter, έπειτα στα δεδομένα R_filter_without_temp τα οποία δεν περιέχουν τα δεδομένα θερμοκρασίας και τέλος στα ισορροπημένα δεδομένα R_filter_balanced. Στην πρώτη δοκιμή παρατηρήθηκαν και ξεχώρισαν δύο αλγόριθμοι, ο KNNClassifier αλγόριθμος και ο DecisionTreeClassifier. Οι δύο αυτοί αλγόριθμοι , είναι οι μόνοι με ποσοστό ευστοχίας πάνω από 99%, όπως φαίνεται και στην *Εικόνα 75* οι δύο αλγόριθμοι όταν δέχονται την κατάλληλη τιμή στην μεταβλητή παραμετροποίησης τους πλησιάζουν το 100% ευστοχίας. Η επιλογή όμως του

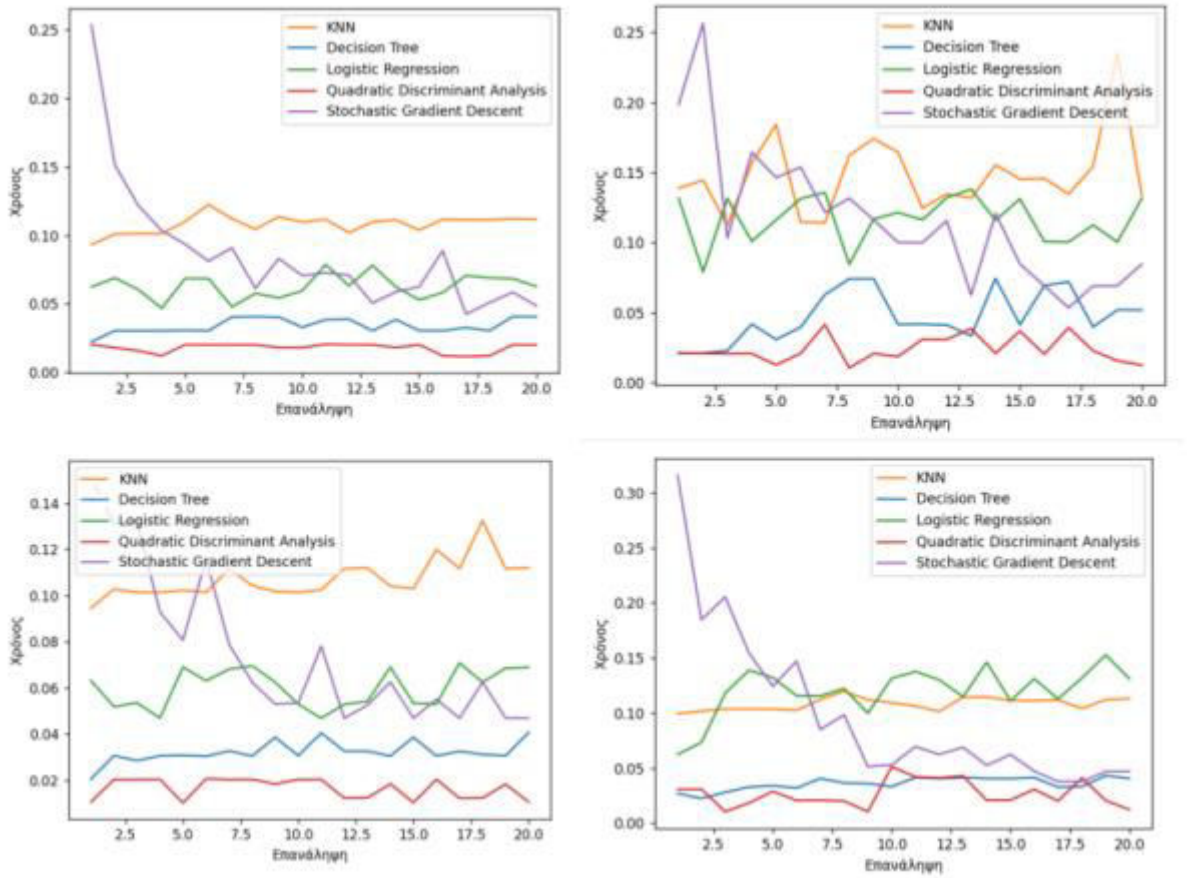
ιδανικότερου αλγορίθμου δεν θα γίνει στη συγκεκριμένη περίπτωση μόνο από το ποσοστό ευστοχίας των αλγορίθμων παρόλο που υπάρχουν περιπτώσεις που κάποιος από τους δύο μπορεί να έχει λίγο μεγαλύτερο ποσοστό ευστοχίας.



Εικόνα 75: Διάγραμμα Ποσοστών ευστοχίας από την σύγκριση όλων των αλγορίθμων στα δεδομένα R_filter.

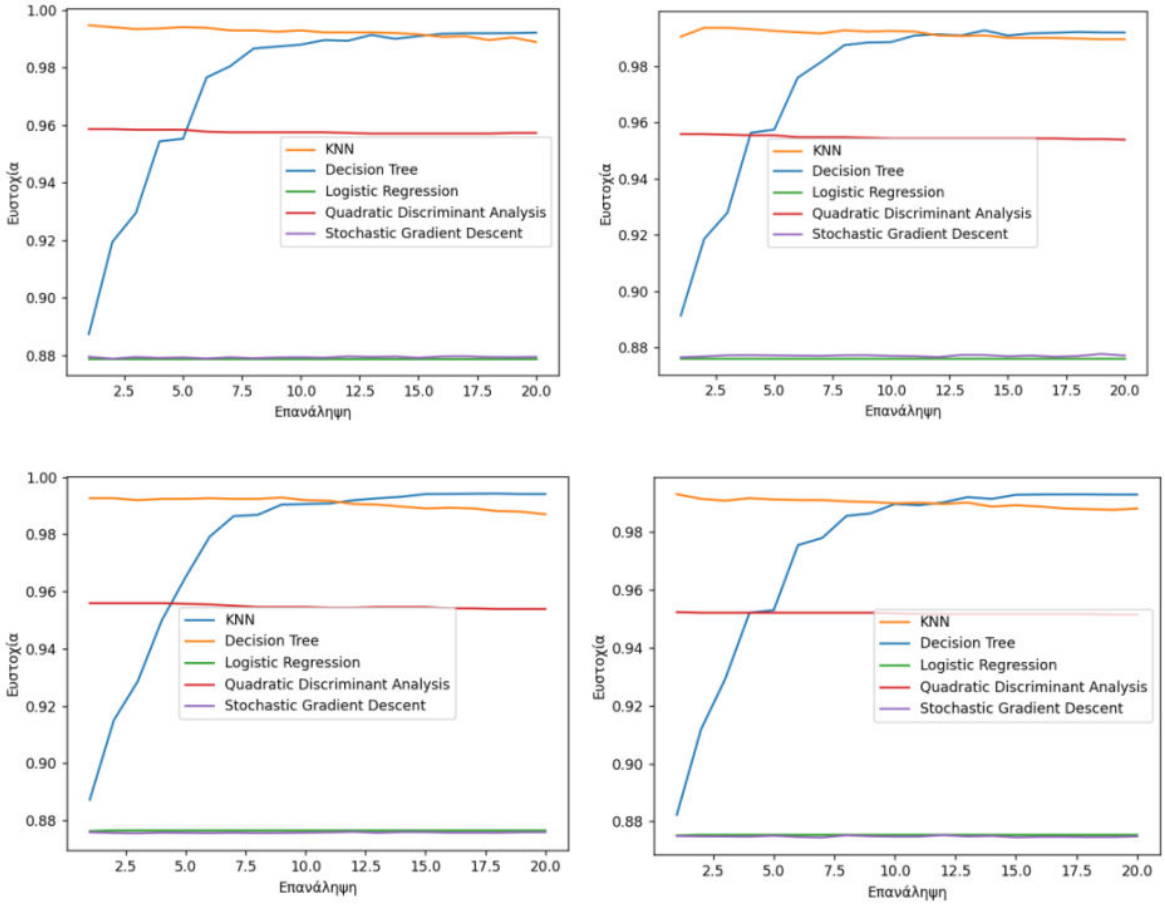
Ο δεύτερος παράγοντας που θα συμβάλει στην σύγκριση όλων των αλγορίθμων είναι ο χρόνος υπολογισμού του κάθε αλγορίθμου. Στην *Εικόνα 76* γίνεται η γραφική αναπαράσταση του χρόνου υπολογισμού των αλγορίθμων. Από την εικόνα μπορεί κανείς να αντιληφθεί, ότι ο αλγόριθμος KNNClassifier μπορεί να έχει πολύ μεγάλο ποσοστό ευστοχίας, έχει όμως στις περισσότερες επαναλήψεις τον μεγαλύτερο χρόνο πρόβλεψης των αποτελεσμάτων. Από την άλλη, ο αλγόριθμος DecisionTreeClassifier είναι από τους πιο γρήγορους αλγορίθμους, έχοντας μόνο τον αλγόριθμο QDA να έχει λίγο μικρότερο χρόνο υπολογισμού. Λαμβάνοντας υπόψιν και τον χρόνο υπολογισμού πρόβλεψης των αποτελεσμάτων, συμπεραίνεται ότι ο ιδανικότερος αλγόριθμος για την πρόβλεψη άρρωστου ή υγιούς δέντρου στα δεδομένα R_filter, είναι ο αλγόριθμος DecisionTreeClassifier, διότι παρόλο που το ποσοστό ευστοχίας του αλγορίθμου

KNNClassifier κινείται στο ίδιο ποσοστό ευστοχίας με το ποσοστό του DecisionTreeClassifier, ο χρόνος του DecisionTreeClassifier είναι ο μισός ή και λιγότερος από αυτόν του KNNClassifier για μια πρόβλεψη.



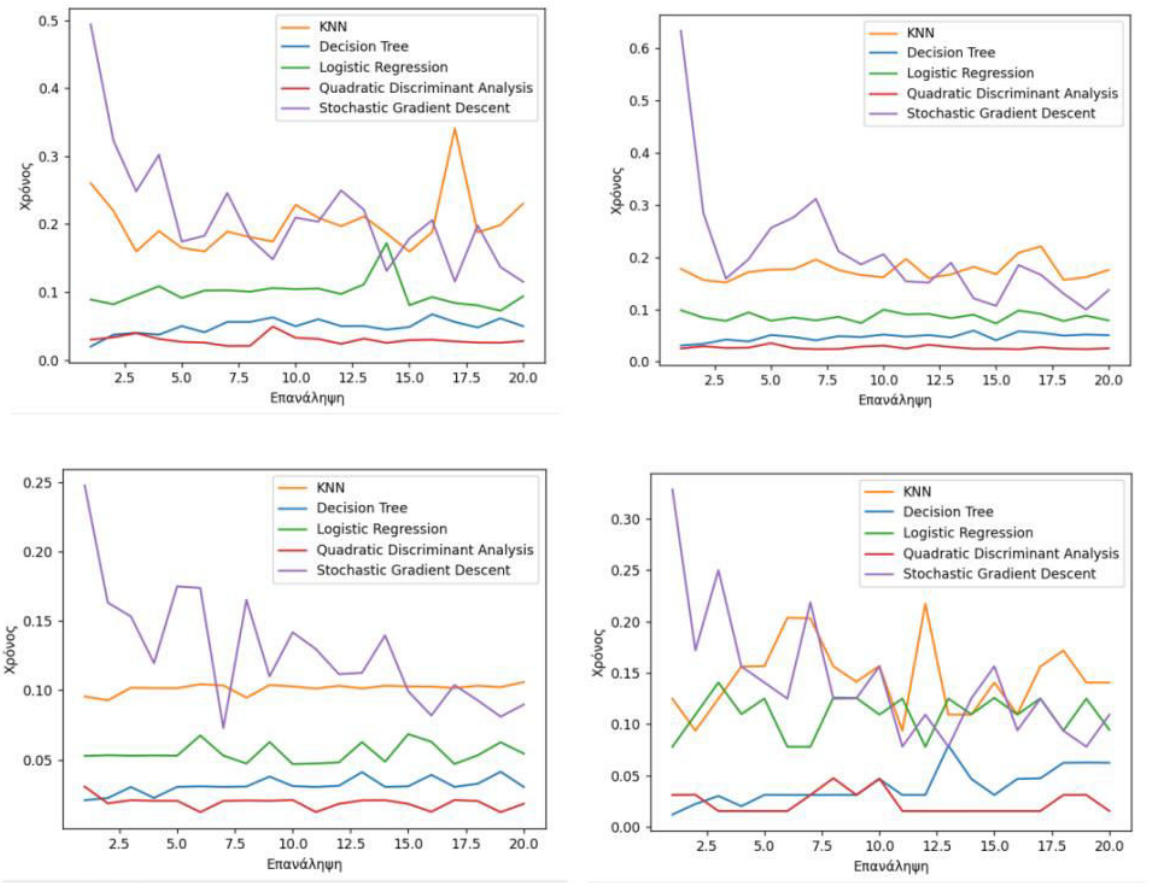
Εικόνα 76: Διάγραμμα Χρόνου από την σύγκριση όλων των αλγορίθμων στα δεδομένα R_filter.

Στην δεύτερη δοκιμή που έγινε στα δεδομένα R_filter_without_temp, τα αποτελέσματα εξελίχθηκαν παρόμοια με την πρώτη δοκιμή. Η επίδοση όλων των αλγορίθμων μειώθηκε, αφού τα ποσοστά ευστοχίας όλων των αλγορίθμων μειώθηκαν. Οι αλγόριθμοι όμως KNNClassifier και DecisionTreeClassifier, όπως και στην πρώτη δοκιμή είναι οι δύο με τα μεγαλύτερα ποσοστά ευστοχίας και παρόλο που υπάρχουν περιπτώσεις και σε αυτή τη δοκιμή κάποιος αλγόριθμος να έχει λίγο μεγαλύτερο ποσοστό ευστοχίας από τον άλλον, θα ληφθεί υπόψιν και αυτή τη φορά ο χρόνος που χρειάζονται οι δύο αλγόριθμοι για να κάνουν μία πρόβλεψη.



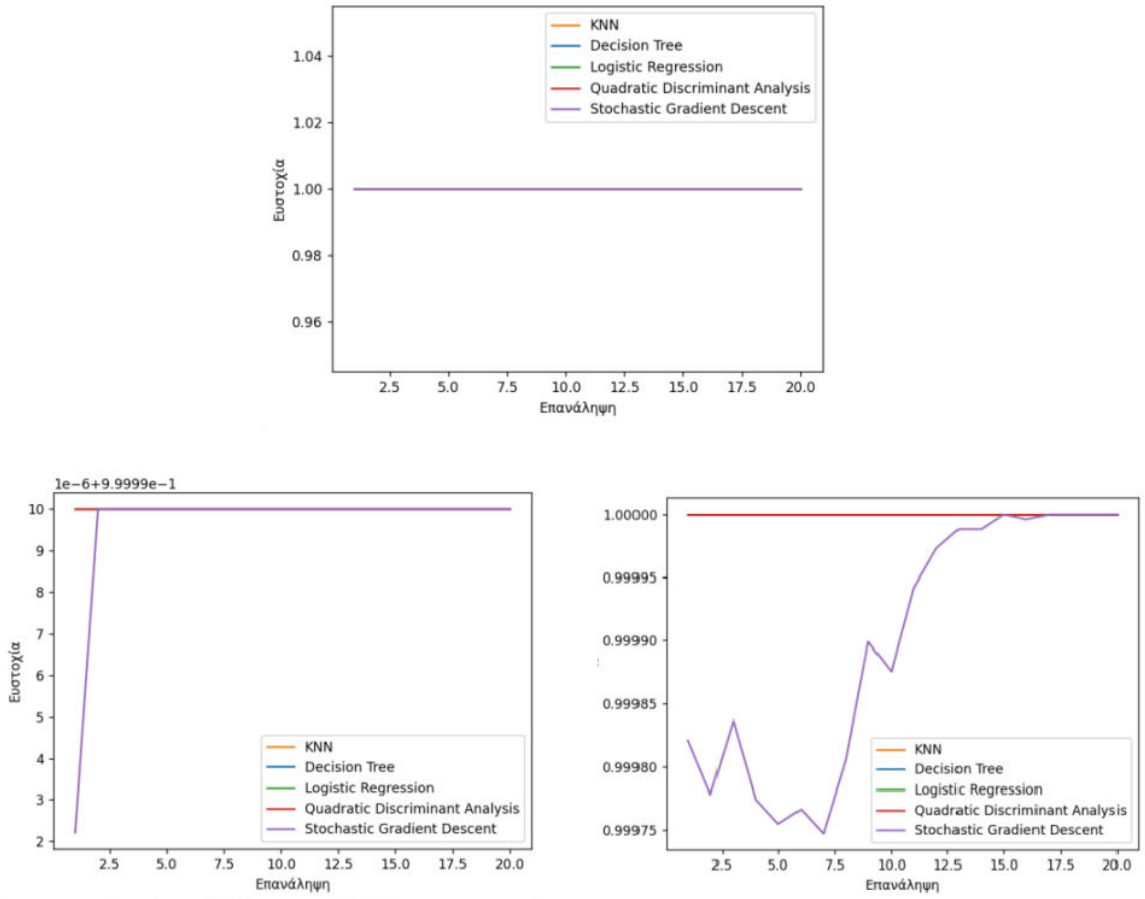
Εικόνα 77: Διάγραμμα Ποσοστών ευστοχίας από την σύγκριση όλων των αλγορίθμων στα δεδομένα R_filter_no_temp.

Όπως θα περίμενε κάποιος και στην συγκεκριμένη δοκιμή, οι χρόνοι κυμαίνονται όπως και στην πρώτη δοκιμή. Για τους ίδιους λόγους λοιπόν και στην περίπτωση που δεν περιλαμβάνεται υπόψιν η θερμοκρασία, ο ιδανικότερος αλγόριθμος είναι ο DecisionTreeClassifier, όπως είδαμε και στην πρώτη δοκιμή ο χρόνος που χρειάζεται ο DecisionTreeClassifier για να τελειώσει μία πρόβλεψη, είναι ο μισός ή λιγότερο από το μισό από τον χρόνο του KNNClassifier αλγόριθμου.



Εικόνα 78: Διάγραμμα Χρόνου από την σύγκριση όλων των αλγορίθμων στα δεδομένα R_filter_no_temp.

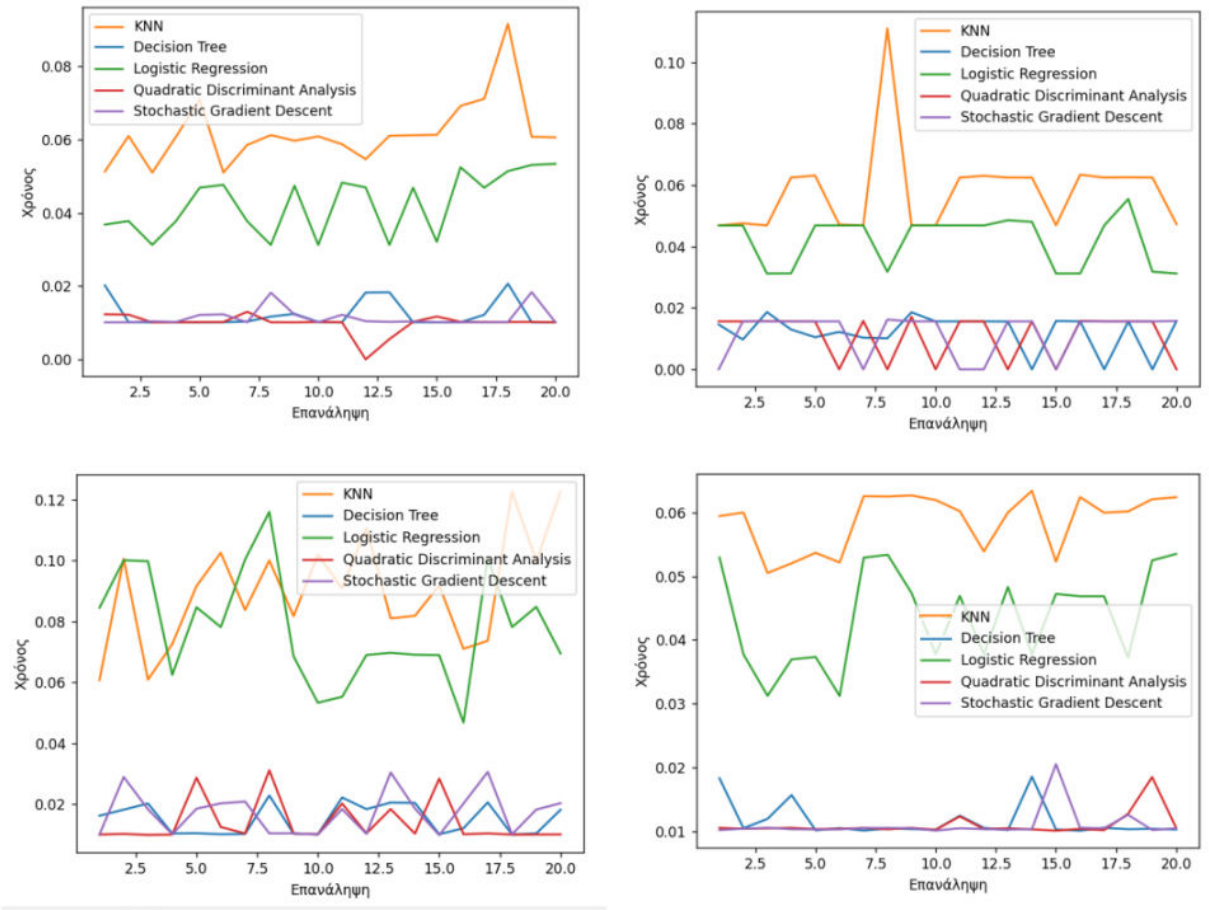
Στη τρίτη δοκιμή που έγινε στα δεδομένα R_filter_balanced όλοι οι αλγόριθμοι είχανε ποσοστό ευστοχίας 100%, εκτός από τον Stochastic Gradient Descent αλγόριθμο. Στην *Εικόνα 79* παρουσιάζονται δύο από τις είκοσι περιπτώσεις που ο SGD αλγόριθμος δεν εμφάνισε 100% ποσοστό ευστοχίας. Και σε αυτή τη περίπτωση δεν μπορεί να επιλεχθεί ο ιδανικότερος αλγόριθμος μόνο από το ποσοστό ευστοχίας των αλγορίθμων. Το μόνο που μπορεί να συμπεράνει κάποιος με σιγουριά από τα γραφήματα ευστοχίας, είναι ότι ο SGD αλγόριθμος δεν είναι η ιδανικότερη επιλογή. Για αυτό το λόγο, η απόφαση του ιδανικότερου αλγορίθμου θα γίνει και σε αυτή τη περίπτωση με τον συνδυασμό του ποσοστού ευστοχίας αλλά και του χρόνου που χρειάζεται για να κάνουν μια πρόβλεψη οι αλγόριθμοι.



Εικόνα 79: Διάγραμμα Ποσοστών ευστοχίας από την σύγκριση όλων των αλγορίθμων στα δεδομένα R_filter_balanced

Στην τρίτη δοκιμή οι καλύτεροι χρόνοι ήταν ανάμεσα στους εξής τρεις αλγορίθμους, στον DecisionTreeClassifier, τον Quadratic Disciminant Analysis και στον Stochastic Gradient Descent, ο SDG αλγόριθμος δεν είναι η ιδανικότερη επιλογή όπως αναφέρθηκε και νωρίτερα, διότι είναι ο μόνος που δεν εμφανίζει σε όλες τις περιπτώσεις ποσοστό ευστοχίας 100%. Στην *Εικόνα 80* δίνονται οι τέσσερις διαφορετικές περιπτώσεις που εμφανίστηκαν σε όλες τις δοκιμές που έγιναν για την εύρεση του καλύτερου χρόνου ανάμεσα στον DecisionTreeClassifier και στον QDA αλγόριθμο. Στις τέσσερις διαφορετικές περιπτώσεις που εμφανίστηκαν, μόνο στη μία ο αλγόριθμος DecisionTreeClassifier είχε στις περισσότερες επαναλήψεις μικρότερο χρόνο από τον QDA αλγόριθμο. Στα υπόλοιπα τρία ο αλγόριθμος QDA εμφανίζει

περισσότερες φορές μεγαλύτερο ποσοστό ευστοχίας, λαμβάνοντας υπόψιν την επίδοση του αλγορίθμου και στον χρόνο αλλά και στην ευστοχία, ο ιδανικότερες αλγόριθμος στην τρίτη δοκιμή είναι ο αλγόριθμος QDA.



Εικόνα 80: Διάγραμμα Χρόνου από την σύγκριση όλων των αλγορίθμων στα δεδομένα R_filter_balanced.

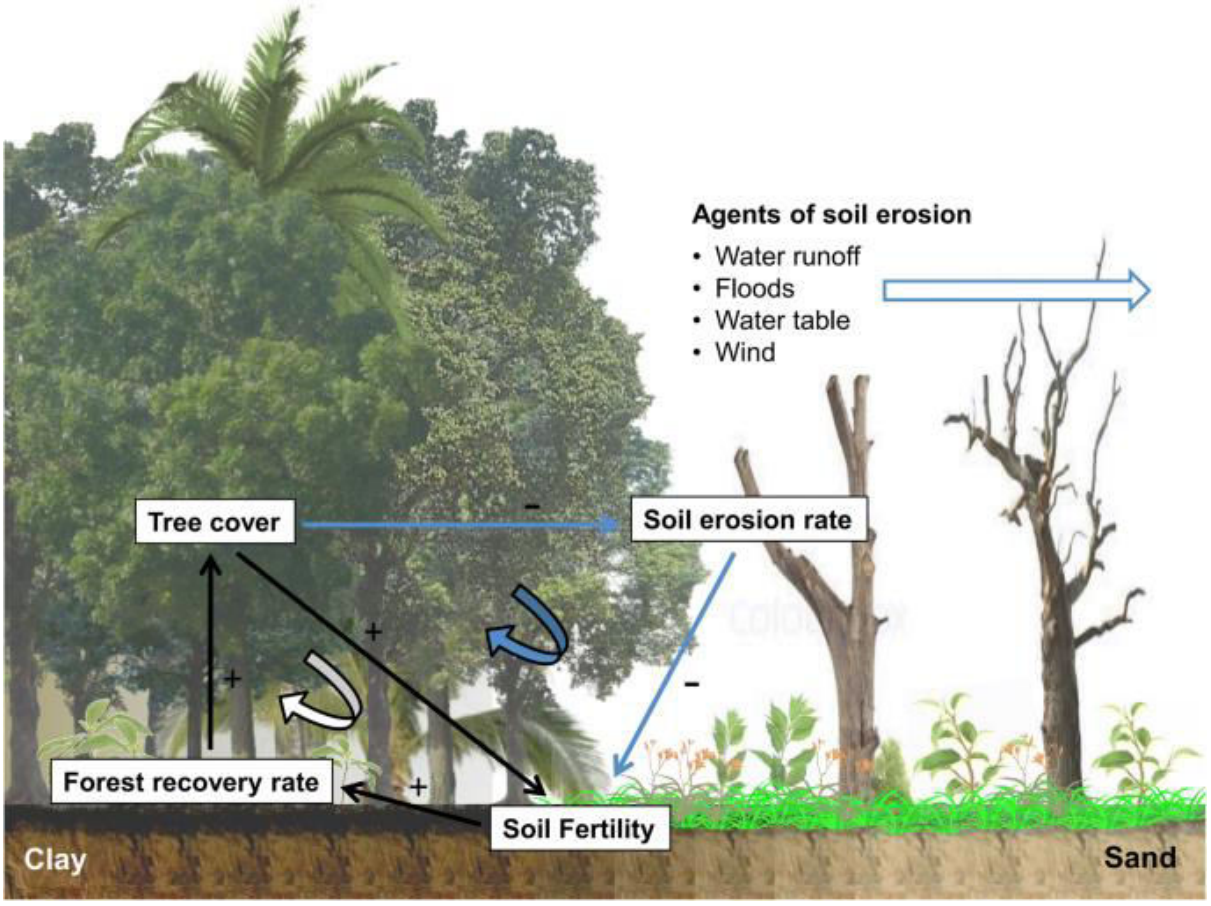
Κεφάλαιο 5 - Συμπεράσματα και Μελλοντικές Επεκτάσεις

Το διαδίκτυο των πραγμάτων μέρα με τη μέρα αρχίζει και εντάσσεται όλο ένα και περισσότερο στην καθημερινότητα του ανθρώπου. Τα δίκτυα αισθητήρων αποδεικνύουν καθημερινά ότι έχουν την δυνατότητα να πετύχουν πράγματα που κάποιος μπορεί να μην έχει φανταστεί παλαιότερα. Ενώ τα μοντέλα μηχανικής μάθησης εδώ και αρκετά χρόνια έχουν βρει θέση στη ζωή του ανθρώπου. Στην παρούσα διπλωματική εργασία έγινε ο συνδυασμός όλων των παραπάνω εννοιών, με σκοπό την αναγνώριση των υγιή και άρρωστων δέντρων. Τα πειράματα που διεξάχθηκαν στο πειραματικό στάδιο για την εκπαίδευση των αλγορίθμων, κατάφεραν και δημιούργησαν μοντέλα τα οποία είχαν ευστοχία πάνω από 99% και στην περίπτωση της ισορροπημένης συλλογής δεδομένων σχεδόν όλα τα μοντέλα εμφάνισαν 100% ποσοστό ευστοχίας. Αυτά τα πειράματα όμως εφαρμόστηκαν σε ήδη αναγνωρισμένα δέντρα από ειδικούς του κλάδου, το οποίο δεν είναι κάτι που θα δίνεται και στην περίπτωση μιας ρεαλιστικής εφαρμογής.Ο σκοπός των μοντέλων που αναπτύξαμε, είναι να μπορούν να ξεχωρίσουν τα υγιή με τα άρρωστα δέντρα όσο πιο γρήγορα γίνεται δίχως να χρειάζεται η παρέμβαση κάποιου ειδικού. Στην πραγματικότητα ένα δέντρο μπορεί να είναι στρεσαρισμένο για πάρα πολλούς λόγους. Τα μοντέλα που αναπτύχθηκαν ελέγχουν μόνο τον εδαφικό παράγοντα του δέντρου αλλά ακόμα και όταν πρόκειται για την μελέτη του εδάφους, σίγουρα δεν είναι αρκετή μόνο η παρακολούθηση υγρασίας και θερμοκρασίας για να εκπαιδευτεί σωστά ένα μοντέλο και να μπορεί να ανταπεξέλθει σε ρεαλιστικά προβλήματα. Από τα πειράματα που διεξάχθηκαν όμως παρατηρήθηκε, ότι ο ρόλος της θερμοκρασίας παίζει σημαντικό παράγοντα για την πρόβλεψη της καταπόνησης, κάτι το οποίο δεν θα περίμενε κανείς να συμβαίνει κοιτώντας απλά τις μετρήσεις. Επίσης παρατηρήθηκε ότι τα περισσότερα μοντέλα, είχαν την τάση να αναγνωρίζουν ένα δέντρο άρρωστο, δίχως να είναι. Αυτή η τάση των μοντέλων όμως αν υπήρχε σε ένα μελλοντικό σύστημα το οποίο θα ειδοποιούσε τον αγρότη με σκοπό την πρόληψη των

δέντρων, δεν θα ήταν κάτι το οποίο θα είχε κακές επιπτώσεις ούτε για τον αγρότη αλλά ούτε για το ίδιο το δέντρο. Επομένως δεν είναι κάτι το οποίο θα μπορούσε να χαρακτηρίσει κανείς ακόμα ως ελάττωμα.

Έχοντας συλλέξει και μελετήσει όλα τα δεδομένα και τα αποτελέσματα από το πειραματικό στάδιο, μελλοντικά θα ήταν επιθυμητή η ανάπτυξη μοντέλων με περισσότερες παραμέτρους. Όπως αναφέρθηκε και νωρίτερα οι μετρήσεις μόνο της θερμοκρασίας και της υγρασίας δεν είναι αρκετές για την εφαρμογή αυτών των μοντέλων σε ρεαλιστικά προβλήματα, για αυτό και επιπλέον παράμετροι όπως η σύσταση του χώματος και η κλίση του εδάφους προτείνεται να ληφθεί υπόψιν στην μελλοντική ανάπτυξη παρόμοιων μοντέλων έρευνας.

Επίσης η μελέτη και άλλων παραγόντων πέραν του εδάφους, όπως τα καιρικά φαινόμενα θα μπορούσε να προστεθεί και να ενισχύσει σημαντικά την ευστοχία των αλγορίθμων σε ρεαλιστικά προβλήματα. Η *Εικόνα 81* παρουσιάζει πως η διαφορετική σύσταση του χώματος μπορεί να επηρεάσει την υγεία ενός δέντρου.



Εικόνα 81: Αποτελέσματα της διαφορετικής σύστασης του χώματος [56].

Τέλος να σημειωθεί ότι κάθε δέντρο έχει τις δικές του ιδιαιτερότητες, για αυτό το λόγο για την επίτευξη των ιδανικότερων μοντέλων μηχανικής ανάπτυξης θα πρέπει να γίνει η εκπαίδευση και η δοκιμή τους σε κάθε δέντρο διαφορετικά.

Βιβλιογραφία

[1] A. Rayes and S. Salam, Internet of Things From Hype to Reality: The Road to Digitization. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-319-99516-8.

[2] Samuel Greengard. *The internet of things*. MIT press, 2021.

[3] foodmanufacture.co.uk, 'Smart food factories move even closer', foodmanufacture.co.uk. https://www.foodmanufacture.co.uk/Article/2017/06/15/Smart-food-factories-move-even-closer (accessed Jan. 31, 2023).

[4] Industrial Internet of Things (IIoT) solving business needs', DATAQUEST, Jan. 03, 2020. https://www.dqindia.com/industrial-internet-things-iiot-solving-business-needs/ (accessed Jan. 31, 2023).

[5] D. Thaler, D. McPherson, H. Tschofenig, and J. Arkko. Architectural considerations in smart object networking. shorturl.at/KQZ89, 2015.

[6] S. Kulkarni, and S. Kulkarni. Communication models in internet of things: a survey. *International Journal of Science Technology & Engineering*, 3(11):87–91, 2017.

[7] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas," IEEE Internet of Things Journal, vol. 5, no. 6. Institute of Electrical and Electronics Engineers (IEEE), pp. 4890–4899, Dec. 2018. doi: 10.1109/jiot.2018.2879579.

[8] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture," Computers and Electronics in Agriculture, vol. 157. Elsevier BV, pp. 218–231, Feb. 2019. doi: 10.1016/j.compag.2018.12.039.

[9] R. Dagar, S. Som, and S. K. Khatri, "Smart Farming – IoT in Agriculture," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, Jul. 2018. doi: 10.1109/icirca.2018.8597264.

[10] 5G: the gamechanger for precision agriculture | United Nations Development Programme', UNDP. https://www.undp.org/policy-centre/singapore/blog/5g-gamechanger-precision-agriculture (accessed Jan. 31, 2023).

[11] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big Data in Smart Farming – A review," Agricultural Systems, vol. 153. Elsevier BV, pp. 69–80, May 2017. doi: 10.1016/j.agsy.2017.01.023.

[12] M. Dholu and K. A. Ghodinde, "Internet of Things (IoT) for Precision Agriculture Application," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, May 2018. doi: 10.1109/icoei.2018.8553720.

[13] A. Yazdinejad et al., "A Review on Security of Smart Farming and Precision Agriculture: Security Aspects, Attacks, Threats and Countermeasures," Applied Sciences, vol. 11, no. 16. MDPI AG, p. 7518, Aug. 16, 2021. doi: 10.3390/app11167518.

[14] M. Balasubramaniyan and C. Navaneethan, "Applications of Internet of Things for smart farming – A survey," Materials Today: Proceedings, vol. 47. Elsevier BV, pp. 18–24, 2021. doi: 10.1016/j.matpr.2021.03.480.

[15] Y. Mekonnen, S. Namuduri, L. Burton, A. Sarwat, and S. Bhansali, "Review—Machine Learning Techniques in Wireless Sensor Network Based Precision Agriculture," Journal of The Electrochemical Society, vol. 167, no. 3. The Electrochemical Society, p. 037522, Dec. 19, 2019. doi: 10.1149/2.0222003jes.

[16] A. Sharma, A. Jain, P. Gupta, and V. Chowdary, "Machine Learning Applications for Precision Agriculture: A Comprehensive Review," IEEE Access, vol. 9. Institute of Electrical and Electronics Engineers (IEEE), pp. 4843–4873, 2021. doi: 10.1109/access.2020.3048415.

[17] S. Condran, M. Bewong, M. Z. Islam, L. Maphosa, and L. Zheng, "Machine Learning in Precision Agriculture: A Survey on Trends, Applications and Evaluations Over Two Decades," IEEE Access, vol. 10. Institute of Electrical and Electronics Engineers (IEEE), pp. 73786–73803, 2022. doi: 10.1109/access.2022.3188649.

[18] R. Akhter and S. A. Sofi, "Precision agriculture using IoT data analytics and machine learning," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 8. Elsevier BV, pp. 5602–5618, Sep. 2022. doi: 10.1016/j.jksuci.2021.05.013.

[19] A. Bauer et al., "Combining computer vision and deep learning to enable ultra-scale aerial phenotyping and precision agriculture: A case study of lettuce production," Horticulture Research, vol. 6, no. 1. Oxford University Press (OUP), Jun. 01, 2019. doi: 10.1038/s41438-019-0151-5.

[20] Y. Zhang. New advances in machine learning. BoD–Books on Demand, 2010.

[21] D. Sharma, and N. Kumar. A review on machine learning algorithms, tasks and applications. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 6(10):2278–1323, 2017.

[22] G. Bonaccorso. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[23] M. M. Ali, B. K. Paul, K. Ahmed, F. M. Bui, J. M. W. Quinn, and M. A. Moni, "Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison," Computers in Biology and Medicine, vol. 136. Elsevier BV, p. 104672, Sep. 2021. doi: 10.1016/j.compbiomed.2021.104672.

[24] B. Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.

[25] C. Crisci, B. Ghattas, and G. Perera, "A review of supervised machine learning algorithms and their applications to ecological data," Ecological Modelling, vol. 240. Elsevier BV, pp. 113–122,

Aug. 2012. doi: 10.1016/j.ecolmodel.2012.03.001.

[26] S. Ray, "A Quick Review of Machine Learning Algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE, Feb. 2019. doi: 10.1109/comitcon.2019.8862451.

[27] Top 10 Machine Learning Algorithms in 2022', Spiceworks. https://www.spiceworks.com/tech/artificial-intelligence/articles/top-ml-algorithms/ (accessed Jan. 31, 2023).

[28] I. Maglogiannis, K. Karpouzis, and M. Wallace, Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies. Amsterdam: IOS Press, 2007..

[29] M Praveena and V Jaiganesh. A literature review on supervised machine learning algorithms and boosting process. International Journal of Computer Applications, 169(8):32–35, 2017.

[30] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi. Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3):128–138, 2017.

[31] V. Nasteski, "An overview of the supervised machine learning methods," HORIZONS.B, vol. 4. University St. Kliment Ohridski - Bitola, pp. 51–62, Dec. 15, 2017. doi: 10.20544/horizons.b.04.1.17.p05.

[32] D. Abdulqader, A. M. Abdulazeez, and D. Q. Zeebaree. Machine learning supervised algorithms of gene selection: A review. Machine Learning, 62(03):233–244, 2020.

[33] A. Moldagulova and R. Bte. Sulaiman, "Using KNN algorithm for classification of textual documents," 2017 8th International Conference on Information Technology (ICIT). IEEE, May 2017. doi: 10.1109/icitech.2017.8079924.

[34] M. Mahmudur, R. Khan, R. Bente, A. B. Siddique, and M. Oishe. Study and observation of the variation of accuracies of knn, svm, lmnn, enn algorithms on eleven different datasets from uci machine learning repository. In *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, pages 124–129. IEEE, 2018.

[35] C. Shravya, K. Pravalika, and S. Subhani. Prediction of breast cancer using supervised machine learning techniques. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8(6):1106–1110, 2019.

[36] J.-C. Chouinard, 'k-Nearest Neighbors (KNN) in Python', JC Chouinard. https://www.jcchouinard.com/k-nearest-neighbors/ (accessed Jan. 31, 2023).

[37] D. Gupta, A. Malviya, and S. Singh. Performance analysis of classification tree learning algorithms. *International Journal of Computer Applications*, 55(6), 2012.

[38] Decision Tree Algorithm, Explained', KDnuggets. https://www.kdnuggets.com/decision-tree-algorithm-explained.html (accessed Jan. 31, 2023).

[39] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," Journal of Applied Science and Technology Trends, vol. 2, no. 01. Interdisciplinary Publishing Academia, pp. 20–28, Mar. 24, 2021. doi: 10.38094/jastt20165.

[40] D. Thangamani, and P. Sudha. Identification of malnutrition with use of supervised datamining techniques–decision trees and artificial neural networks. *Int J Eng Comput Sci*, 3(09), 2014.

[41] K. Bernat and W. Drzewiecki, 'Two-stage subpixel impervious surface coverage estimation: comparing classification and regression trees and artificial neural networks', Amsterdam, Netherlands, Oct. 2014, p. 92441I. doi: 10.1117/12.2067308.

[42] Y. Bouzida, and F. Cuppens. Neural networks vs. decision trees for intrusion detection. In *IEEE/IST workshop on monitoring, attack detection and mitigation (MonAM)*, volume 28, page 29. Citeseer, 2006.

[43] A. Agresti. *Foundations of linear and generalized linear models*. John Wiley & Sons, 2015.

[44] E. W. Frees, R. A. Derrig, and G. Meyers. *Predictive modeling applications in actuarial science*, volume 1. Cambridge University Press, 2014.

[45] H. Nawaz, 'Develop a Logistic Regression Machine Learning Model', Medium, Mar. 31, 2022. https://blog.devgenius.io/develop-a-logistic-regression-machine-learning-model-64d2be403ba3 (accessed Feb. 01, 2023).

[46] H. A. N., A. H. H., R. Siti, and L. C.. Discriminant analysis: An illustrated example. *African Journal of Business Management*, 4(9):1654– 1667, 2010.

[47] K. S. Kim, H. H. Choi, C. S. Moon, and C. W. Mun, "Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions," Current Applied Physics, vol. 11, no. 3. Elsevier BV, pp. 740–745, May 2011. doi: 10.1016/j.cap.2010.11.051.

[48] A. E. Mohamed, "Comparative study of four supervised machine learning techniques for classification," International Journal of Applied, vol. 7, no. 2, pp. 1–15, 2017.

[49] S. Chowdhury and M. P. Schoen, "Research Paper Classification using Supervised Machine Learning Techniques," 2020 Intermountain Engineering, Technology and Computing (IETC). IEEE, Oct. 02, 2020. doi: 10.1109/ietc47856.2020.9249211.

[50] P. Jain, J. M. Garibaldi, and J. D. Hirst, "Supervised machine learning algorithms for protein structure classification," Computational Biology and Chemistry, vol. 33, no. 3. Elsevier BV, pp. 216–223, Jun. 2009. doi: 10.1016/j.compbiolchem.2009.04.004.

[51] Medium. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-

learning-algorithms-934a444fca47ple?gi=e172516666a6 (accessed Feb. 01, 2023).

[52] I. Muhammad and Z. Yan, "SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY.," ICTACT Journal on Soft Computing, vol. 5, no. 3, 2015.

[53] S. Hikmat Haji and A. M. Abdulazeez, "COMPARISON OF OPTIMIZATION TECHNIQUES BASED ON GRADIENT DESCENT ALGORITHM: A REVIEW", J Arch.Egyptol, vol. 18, no. 4, pp. 2715-2743, Feb. 2021.

[54] M. Deshpande, 'Complete Guide to Deep Neural Networks – Part 2', GameDev Academy, Sep. 20, 2017. https://gamedevacademy.org/complete-guide-to-deep-neural-networks-part-2/ (accessed Feb. 01, 2023).

[55] Libelium Comunicaciones Distribuidas S.L. Agriculture 2.0 technical guide, 2013.

[56] B. M. Flores, A. Staal, C. C. Jakovac, M. Hirota, M. Holmgren, and R. S. Oliveira, "Soil erosion as a resilience drain in disturbed tropical forests," Plant and Soil, vol. 450, no. 1–2. Springer Science and Business Media LLC, pp. 11–25, May 08, 2019. doi: 10.1007/s11104-019-04097-8.