ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Αυτοματοποιημένη Οπτικοποίηση Δεδομένων με Μηχανική Μάθηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΝΔΡΕΑΣ ΧΑΤΖΗΣΑΒΒΑΣ

Επιβλέπων : **Γρηγόρης Μεντζας**
Καθηγητής Ε.Μ.Π.

Αθήνα, Φλεβάρης 2025

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Αυτοματοποιημένη Οπτικοποίηση Δεδομένων με Μηχανική Μάθηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΝΔΡΕΑΣ ΧΑΤΖΗΣΑΒΒΑΣ

Επιβλέπων : Γρηγόρης Μέντζας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24η Φεβρουαρίου 2025.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γρηγόρης Μέντζας Καθηγητής Ε.Μ.Π.	Δημήτριος Ασκούνης Καθηγητής Ε.Μ.Π.	Ευάγγελος Μαρινάκης Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Φλεβάρης 2025

(Υπογραφή)

.....

Ανδρέας Χατζησάββας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ένα σύστημα σύστασης οπτικοποίησης δεδομένων είναι εξαιρετικά σημαντικό στη σύγχρονη εποχή, όπου ο όγκος των δεδομένων αυξάνεται συνεχώς λόγω της ψηφιοποίησης και της τεχνολογικής προόδου. Οι επιχειρήσεις, οι οργανισμοί και οι ερευνητές συλλέγουν τεράστιες ποσότητες δεδομένων από διάφορες πηγές (π.χ. κοινωνικά δίκτυα, αισθητήρες IoT, διαδικτυακές πλατφόρμες), καθιστώντας δύσκολη την επεξεργασία και την εξαγωγή χρήσιμων πληροφοριών. Η οπτικοποίηση δεδομένων επιτρέπει την παρουσίαση πολύπλοκων δεδομένων με τρόπο που είναι εύκολα κατανοητός.

Η μηχανική μάθηση παίζει καθοριστικό ρόλο στην ενίσχυση τέτοιων συστημάτων, καθώς μπορεί να αυτοματοποιήσει την ανάλυση τεράστιων όγκων δεδομένων και να προτείνει τις κατάλληλες οπτικοποιήσεις με βάση τα χαρακτηριστικά των δεδομένων και τις ανάγκες των χρηστών. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να αναγνωρίζουν μοτίβα που μπορεί να μην είναι άμεσα ορατά και να προτείνουν τρόπους παρουσίασης των πληροφοριών που μεγιστοποιούν την κατανόηση.

Στην παρούσα διπλωματική εργασία, μελετάται η αυτοματοποίηση της διαδικασίας σύστασης οπτικοποίησης δεδομένων με την χρήση μηχανικής μάθησης. Εξετάζονται τα κύρια χαρακτηριστικά που συμβάλλουν στην βέλτιστη οπτικοποίηση δεδομένων μέσω διαδικασιών εξαγωγής χαρακτηριστικών. Αναπτύσσονται μοντέλα βασισμένα σε διάφορους αλγορίθμους, με στόχο την κατανόηση της φύσης του προβλήματος και την πρόταση της καταλληλότερης προσέγγισης. Η μεθοδολογία αξιοποιεί διαφορετικά σύνολα δεδομένων από την online πλατφόρμα του plotly, η οποία εξυπηρετεί ως ένα σημείο συγκέντρωσης δεδομένων από ποικίλες πηγές. Στην συνέχεια, διεξάγεται μια σειρά πειραμάτων ώστε να καταλήξουμε σε μοντέλα που ανταπεξέρχονται στις απαιτήσεις μας και να αξιολογήσουμε τις μεθόδους μας. Τέλος, υλοποιείται μια web εφαρμογή που επιτρέπει την γραφική αλληλεπίδραση με το χρήστη και την οπτικοποίηση των αποτελεσμάτων της μεθόδου.

Λέξεις Κλειδιά: <<Αναλυτική, Εξαγωγή Χαρακτηριστικών, Μηχανική Μάθηση, Οπτικοποίηση Αναλυτικής Δεδομένων, Σύσταση Οπτικοποίησης>>

Abstract

A data visualization recommendation system is extremely important in the modern era, where the volume of data is constantly increasing due to digitization and technological advancements. Businesses, organizations, and researchers collect vast amounts of data from various sources (e.g., social networks, IoT sensors, online platforms), making it difficult to process and extract useful information. Data visualization allows for the presentation of complex data in a way that is easily understandable.

Machine learning plays a crucial role in enhancing such systems, as it can automate the analysis of large data sets and suggest the appropriate visualizations based on the data's characteristics and user needs. Machine learning algorithms can identify patterns that may not be immediately visible and propose ways of presenting the information that maximize understandability.

In this thesis, we study the modeling of the data visualization recommendation process using machine learning. We aim to investigate the key features that lead to the optimal visualization of data in a feature extraction and engineering process. We develop~~ed~~ models from a variety of available algorithms that can provide insights into the nature of the problem, allowing us to suggest the ideal approach to achieve our goal. This approach is evaluated on a real dataset from a reliable source. We conduct~~ed~~ a series of experiments to develop models that meet our requirements and evaluate our methods. Lastly, we develop~~ed~~ a web app to present our recommendations to the user and communicate with the recommendation system intuitively.

Keywords: <<Analytics, Feature Extraction, Machine Learning, Data Visualization, Visualization Recommendation>>

Ευχαριστίες

Δεν θα μπορούσα να ολοκληρώσω αυτή την εργασία χωρίς τη στήριξη όλων όσων με στήριξαν καθ'όλη τη διάρκεια αυτής της προσπάθειας.

Καταρχάς θα ήθελα να ευχαριστήσω τον κ. Γρηγόρη Μέντζα για την εμπιστοσύνη που μου έδειξε με την ανάθεση του συγκεκριμένου θέματος, μέσα από το οποίο απέκτησα πολύτιμες γνώσεις και δεξιότητες, οι οποίες θα συμβάλλουν σημαντικά στη μελλοντική μου ακαδημαϊκή και επαγγελματική πορεία. Επίσης θα ήθελα να ευχαριστήσω την διδακτορικό Κατερίνα Λεπενιώτη για την άψογη συνεργασία, τον πολύτιμο χρόνο που αφιέρωσε και την προθυμία της να με βοηθήσει ανά πάσα χρονική στιγμή. Η καθοδήγηση της ήταν καθοριστικής σημασίας για την περάτωση της παρούσας εργασίας.

Επιπλέον θα ήθελα να ευχαριστήσω τους γονείς μου, Κατερίνα και Γιώργο, και τις δύο μου αδερφές, Κάλια και Κοραλία, που με στήριξαν ανελλιπώς σε όλα τα φοιτητικά μου χρόνια. Ιδιαίτερες ευχαριστίες αξίζουν και οι φίλοι μου για όλες τις πολύτιμες στιγμές που ζήσαμε μαζί. Ευχαριστώ τους Αλέξανδρο, Θοδωρή και Άνδρεα.

Πίνακας περιεχομένων

Πίνακας Πινάκων

Πίνακας Εικόνων

1 *Εισαγωγή*

1.1 *Οπτικοποίησης Δεδομένων*

Η κατανόηση αριθμητικών δεδομένων μπορεί να είναι δύσκολη για πολλούς, καθώς οι αριθμοί από μόνοι τους συχνά δεν προσφέρουν μια άμεση και ξεκάθαρη εικόνα. Αντίθετα, η οπτικοποίηση δεδομένων καθιστά την πληροφορία πιο διαισθητική και ευκολότερη στην κατανόηση, εκμεταλλευόμενη την ικανότητα του ανθρώπινου εγκεφάλου να αναγνωρίζει γρήγορα πρότυπα, συσχετισμούς και τάσεις. Αυτός είναι και ο λόγος που οι οπτικοποιήσεις δεδομένων χρησιμοποιούνται ευρέως σε πολλούς επαγγελματικούς τομείς, διευκολύνοντας όχι μόνο την ανάλυση αλλά και την αποτελεσματική επικοινωνία ευρημάτων.

Η οπτικοποίηση δεδομένων είναι ιδιαίτερα χρήσιμη γιατί επιτρέπει την απεικόνιση σύνθετων πληροφοριών με τρόπο που αποκαλύπτει κρίσιμες τάσεις και μοτίβα, τα οποία μπορεί να παραβλεφθούν σε αμιγώς αριθμητικές παρουσιάσεις. Υποστηρίζει τη λήψη αποφάσεων, διευκολύνοντας τους επαγγελματίες να εξάγουν γρήγορα και αξιόπιστα συμπεράσματα, ενώ παράλληλα ενισχύει την επικοινωνία ευρημάτων με σαφή και κατανοητό τρόπο, ακόμη και για μη εξειδικευμένο κοινό. Επιπλέον, βοηθά στην αναγνώριση πιθανών λαθών ή αντιφάσεων στα δεδομένα, προωθώντας την ακρίβεια στην ανάλυση. Η ευρεία της εφαρμογή σε τομείς όπως η επιστήμη, τα οικονομικά, η ιατρική και το μάρκετινγκ, καταδεικνύει την προσαρμοστικότητά της στις διαφορετικές ανάγκες κάθε κλάδου.

Ωστόσο, τα υπάρχοντα εργαλεία οπτικοποίησης δεδομένων συχνά παρουσιάζουν σημαντικούς περιορισμούς. Πολλά από αυτά απαιτούν εξειδικευμένες τεχνικές γνώσεις, όπως προγραμματισμό ή πολύπλοκες διαδικασίες ρύθμισης παραμέτρων, καθιστώντας τα δυσπρόσιτα για χρήστες χωρίς σχετικό υπόβαθρο. Επιπλέον, η ανάγκη για χειρωνακτική προσαρμογή μπορεί να περιορίσει την ευελιξία και την αποδοτικότητα στη λήψη γρήγορων αποφάσεων. Η οπτικοποίηση δεδομένων, επομένως, αν και αποτελεί ένα καθοριστικό εργαλείο στην επιστημονική και επιχειρηματική διαδικασία, χρειάζεται βελτιώσεις ώστε να γίνει πιο προσιτή, ευέλικτη και αποδοτική. Αποτελεσματικές οπτικοποιήσεις επιτυγχάνουν την ισορροπία μεταξύ πληροφορικής αποδοτικότητας και ευελιξίας, επιτρέποντας την εξαγωγή πολύτιμων συμπερασμάτων από σύνθετα δεδομένα με μεγαλύτερη ευκολία και ακρίβεια.

1.2 *Αντικείμενο διπλωματικής*

Ένα συχνό πρόβλημα στην οπτικοποίηση αναλυτικών δεδομένων είναι η επιλογή του καταλληλότερου γραφικού εργαλείου για την αναπαράσταση των δεδομένων, ώστε να διευκολυνθεί η κατανόηση σύνθετων πληροφοριών και να επιτευχθεί η αποτελεσματική επικοινωνία τους προς το κοινό-στόχο. Στη σύγχρονη εποχή των δεδομένων, η οπτικοποίηση διαδραματίζει καθοριστικό ρόλο τόσο στην ανάλυση όσο και στη μετάδοση γνώσης, βοηθώντας στην ανάδειξη τάσεων, συσχετίσεων και μοτίβων που διαφορετικά μπορεί να παραβλεφθούν.

Ωστόσο, η διαδικασία επιλογής της βέλτιστης μορφής οπτικοποίησης, λαμβάνοντας υπόψη τα χαρακτηριστικά των δεδομένων (όπως ο τύπος, η δομή και η κατανομή τους) και τους στόχους της ανάλυσης (όπως η επισήμανση τάσεων, συσχετίσεων ή ακραίων τιμών), παραμένει μη τυποποιημένη. Παρά την ύπαρξη πληθώρας εργαλείων και τεχνικών οπτικοποίησης, όπως γραφήματα, διαγράμματα και πίνακες, η επιλογή της κατάλληλης μεθόδου βασίζεται σε μεγάλο βαθμό στην εμπειρία και την υποκειμενική κρίση του αναλυτή, αντί να υποστηρίζεται από ένα σαφές και αυτοματοποιημένο σύστημα προτάσεων

Παρά τις προσπάθειες που έχουν γίνει για την επίλυση αυτού του προβλήματος, παρατηρείται ότι πολλές από τις υπάρχουσες προσεγγίσεις απαιτούν έναν βαθμό εξειδικευμένης εμπειρίας από τους χρήστες. Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας μεθόδου που απαλλάσσει πλήρως τον χρήστη από τη διαδικασία επιλογής της κατάλληλης οπτικοποίησης για τα δεδομένα του, αξιοποιώντας τεχνικές μηχανικής μάθησης. Επιδιώκουμε να γεφυρώσουμε αυτό το κενό, καθιστώντας την επιλογή της βέλτιστης, αυτοματοποιημένης

οπτικοποίησης δεδομένων προσβάσιμη σε όλους, χωρίς την ανάγκη ειδικής τεχνικής κατάρτισης.

Το σύστημα που αναπτύξαμε εστιάζει στην εξαγωγή βασικών χαρακτηριστικών των δεδομένων προς οπτικοποίηση, όπως το μέγεθος, τα πρότυπα, οι τάσεις και άλλες κρίσιμες ιδιότητες. Στη συνέχεια, μέσω μιας σειράς πειραμάτων, αξιολογούμε διάφορα μοντέλα μηχανικής μάθησης προκειμένου να προσδιορίσουμε το καταλληλότερο σύστημα αυτόματης σύστασης οπτικοποιήσεων, το οποίο προσαρμόζεται δυναμικά στα χαρακτηριστικά των δεδομένων και στους στόχους της εκάστοτε ανάλυσης. Τέλος, το τελικό μας μοντέλο ενσωματώθηκε σε μια web εφαρμογή, προσφέροντας μια φιλική γραφική διεπαφή που επιτρέπει στους χρήστες να αξιοποιούν την προτεινόμενη προσέγγιση χωρίς να απαιτείται εξοικείωση με τον προγραμματισμό.

1.3 *Οργάνωση κειμένου*

Η παρούσα διπλωματική εργασία αποτελείται από οκτώ κεφάλαια. Στο Κεφάλαιο 1 (Εισαγωγή) παρουσιάζεται ο τομέας της Οπτικοποίησης Δεδομένων, στον οποίο εντάσσεται η έρευνά μας, καθώς και το πλαίσιο του προβλήματος που διερευνούμε. Το Κεφάλαιο 2 περιλαμβάνει την επισκόπηση της σχετικής βιβλιογραφίας, όπου αναλύονται προηγούμενες εργασίες που σχετίζονται με το αντικείμενο της διπλωματικής και παρέχουν το απαραίτητο υπόβαθρο για την κατανόηση της έρευνας. Στο Κεφάλαιο 3 παρουσιάζεται το θεωρητικό υπόβαθρο πάνω στο οποίο βασίζεται η εργασία. Εξετάζονται οι βασικές έννοιες της οπτικοποίησης αναλυτικών δεδομένων και της μηχανικής μάθησης, οι οποίες είναι θεμελιώδεις για την κατανόηση της μεθοδολογίας μας. Το Κεφάλαιο 4 επικεντρώνεται στην προτεινόμενη προσέγγιση της διπλωματικής εργασίας, περιγράφοντας τη διαδικασία εξαγωγής χαρακτηριστικών, τις τεχνικές προεπεξεργασίας, τους αλγόριθμους που δοκιμάστηκαν και τη μεθοδολογία αξιολόγησής τους. Στο Κεφάλαιο 5 παρουσιάζεται η διαδικασία υλοποίησης της προτεινόμενης μεθόδου. Αναλύονται τα εργαλεία που χρησιμοποιήθηκαν, οι πηγές και η διαδικασία συλλογής των δεδομένων, τα τελικά χαρακτηριστικά (features) που παρήχθησαν, οι αρχιτεκτονικές που εφαρμόστηκαν, καθώς και το demo που αναπτύχθηκε για τη διευκόλυνση της χρήσης. Το Κεφάλαιο 6 περιλαμβάνει την αξιολόγηση των ευρημάτων μέσω μιας σειράς πειραμάτων, όπου εξετάζονται οι επιδόσεις των μοντέλων και εξάγονται χρήσιμα συμπεράσματα. Στο Κεφάλαιο 7 συνοψίζονται τα βασικά συμπεράσματα της έρευνας και προτείνονται ιδέες για μελλοντικές επεκτάσεις και βελτιώσεις της μεθοδολογίας. Τέλος, στο Κεφάλαιο 8 παρατίθεται η βιβλιογραφία που αξιοποιήθηκε κατά τη διάρκεια της διπλωματικής εργασίας, προσφέροντας το απαραίτητο θεωρητικό υπόβαθρο και υποστηρίζοντας τα ευρήματα της μελέτης.

2 *Σχετικές εργασίες*

2.1 *Οπτικοποίηση Αναλυτικής Δεδομένων*

Οι τεχνικές Οπτικοποίησης Αναλυτικής Δεδομένων προσφέρουν μια ποικιλία από προτεινόμενες προσεγγίσεις. Στο [1] προτείνεται μια καινοτόμα μέθοδος ταξινόμησης οπτικοποίησης υψηλού επιπέδου. Ενώ συνήθως επιχειρείται η ταξινόμηση των δεδομένων, στην προσέγγιση αυτή ταξινομούνται οι αλγόριθμοι οπτικοποίησης, τα λεγόμενα design models. Επειδή η ταξινομία που εισηγούνται βασίζεται στα design models και δίνει έμφαση στην ανθρώπινη πτυχή της οπτικοποίησης, είναι πιο ευέλικτη από ήδη υπάρχουσες ταξινομίες. Τα design models κατηγοριοποιούνται με βάση το αν είναι διακριτά ή συνεχή και ανάλογα με το πώς ο σχεδιαστικός αλγόριθμος επιλέγει να εμφανίσει χαρακτηριστικά όπως το εύρος των δεδομένων, τη χρονικότητά τους, το χρώμα και τη διαφάνεια. Αυτή η πρωτοποριακή προσέγγιση παρέχει μια εναλλακτική οπτική στο πεδίο της οπτικοποίησης, βοηθώντας να εξηγηθεί πώς παραδοσιακές κατηγοριοποιήσεις (π.χ. πληροφοριακή και επιστημονική οπτικοποίηση) σχετίζονται και επικαλύπτονται, ενώ μπορεί να εμπνεύσει ερευνητικές ιδέες σε τομείς υβριδικής απεικόνισης. Η πρότασή τους διαφέρει σημαντικά από τη δική μας, καθώς εμείς στοχεύουμε στην ταξινόμηση των δεδομένων και όχι των αλγορίθμων. Ωστόσο, μας παρέχει μια βαθύτερη κατανόηση των ειδών οπτικοποίησης, της αποτελεσματικότητάς τους και του τρόπου βέλτιστης αξιοποίησής τους.

Οι προσεγγίσεις βασισμένες σε κανόνες (rule-based approaches) είναι συχνά αποδοτικές σε απλά προβλήματα, αποφεύγοντας την ανάγκη για πιο περίπλοκες τεχνικές. Τέτοιες έρευνες έχουν παρουσιαστεί στο [18], όπου παρέχονται στους χρήστες δυνατότητες ορισμού παραμέτρων, με το σύστημα να προτείνει αντίστοιχες οπτικοποιήσεις με βάση τα δεδομένα που εισάγονται. Στο [19] επιδιώκεται η συμπλήρωση της χειροκίνητης κατασκευής οπτικοποιήσεων μέσω διαδραστικής πλοήγησης σε συλλογές αυτόματα παραγόμενων οπτικοποιήσεων. Το σύστημα αυτό υποστηρίζει μικτή πρωτοβουλία, επιτρέποντας πολυδιάστατη εξερεύνηση προτεινόμενων γραφημάτων που επιλέγονται βάσει στατιστικών

και αντιληπτικών μέτρων. Τα στατιστικά μέτρα αξιολογούν τη σημαντικότητα των δεδομένων μέσω συναρτήσεων όπως ο μέσος όρος και η ομαδοποίηση, ενώ τα αντιληπτικά μέτρα βασίζονται σε αρχές οπτικής αντίληψης, προτιμώντας γραφήματα που διευκολύνουν την αναγνώριση προτύπων και τάσεων. Παρόμοιες προσεγγίσεις έχουν παρουσιαστεί και στα [20] και [21].

Παρόλο που προσφέρουν ικανοποιητικά αποτελέσματα, οι rule-based προσεγγίσεις παρουσιάζουν ορισμένα σημαντικά μειονεκτήματα. Υποθέτουν ότι τα χαρακτηριστικά των δεδομένων έχουν σαφείς προδιαγραφές που παραπέμπουν σε μία ή περισσότερες συγκεκριμένες οπτικοποιήσεις. Επιπλέον, ο ορισμός κανόνων είναι μια χρονοβόρα διαδικασία που απαιτεί εκτενή μελέτη, ενώ όσο αυξάνεται ο όγκος των δεδομένων, οι υπάρχοντες κανόνες δυσκολεύονται να προσδιορίσουν μια μοναδικά αποδοτική πρόταση.

2.2 *Μηχανική Μάθηση και Οπτικοποίηση Αναλυτική Δεδομένων*

Τα εργαλεία που προσφέρει η μηχανική μάθηση δίνουν λύσεις σε αρκετές προκλήσεις που αντιμετωπίζουν οι rule-based προσεγγίσεις. Στο [8], οι ερευνητές στοχεύουν να επιλύσουν τρία βασικά προβλήματα στην αυτόματη οπτικοποίηση δεδομένων: πώς να αξιολογηθεί αν μια οπτικοποίηση είναι «καλή» ή «κακή», ποια από δύο οπτικοποιήσεις είναι «καλύτερη» και πώς να εντοπιστούν οι k βέλτιστες οπτικοποιήσεις για ένα συγκεκριμένο σύνολο δεδομένων. Για την αντιμετώπιση αυτών των προκλήσεων, προτείνεται η χρήση ενός μοντέλου εκμάθησης κατάταξης με είσοδο τα δεδομένα και τα χαρακτηριστικά τους. Εξάγονται 14 χαρακτηριστικά από τα δεδομένα και δημιουργείται μια μερική κατάταξη των οπτικοποιήσεων μέσω ενός κατευθυνόμενου γράφου, γνωστού ως Hasse διάγραμμα. Η αξιολόγηση των γραφικών βασίζεται σε ένα σύστημα βαθμολόγησης που χρησιμοποιεί κανόνες, ενώ για την τελική κατάταξη εφαρμόζεται τοπολογική ταξινόμηση. Αν και η προσέγγιση είναι αποδοτική, απαιτεί σημαντική γνώση στον τομέα της ανάλυσης δεδομένων και τα εξαγόμενα χαρακτηριστικά καλύπτουν περιορισμένο φάσμα πληροφοριών.

Πιο κοντά στην προσέγγιση που ακολουθούμε βρίσκεται η εργασία στο [14], όπου περιγράφεται μια μέθοδος σύστασης οπτικοποιήσεων βασισμένη στη μηχανική μάθηση. Το σύστημα μαθαίνει επιλογές σχεδίασης οπτικοποιήσεων από ένα μεγάλο σύνολο δεδομένων και αντίστοιχες οπτικοποιήσεις. Η μελέτη καλύπτει ζητήματα όπως η επιλογή του κατάλληλου τύπου οπτικοποίησης και η τοποθέτηση μεταβλητών στους άξονες X και Y. Εκπαιδεύονται μοντέλα με τη χρήση νευρωνικών δικτύων και αξιολογείται η γενικευσιμότητα και η αβεβαιότητα των προβλέψεων, αξιοποιώντας δεδομένα από πραγματικούς χρήστες. Ένα

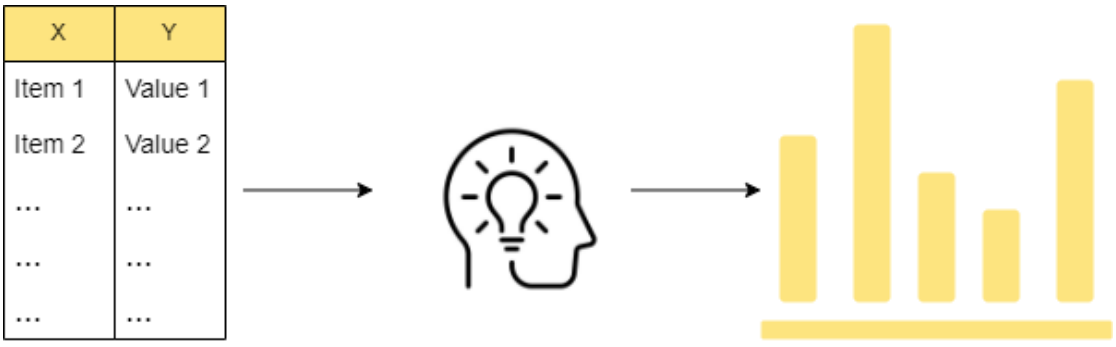
σημαντικό στοιχείο που υιοθετήσαμε και στην παρούσα διπλωματική είναι η χρήση δεδομένων από το Plotly Community Feed, το οποίο αποτελεί αξιόπιστη πηγή για την προσομοίωση της ανθρώπινης κρίσης. Στην εργασία προτείνονται 81 μοναδικά χαρακτηριστικά για κάθε στήλη και 30 συνδυαστικά χαρακτηριστικά μεταξύ στηλών, με τη χρήση 16 συναρτήσεων συνάθροισης, οδηγώντας σε ένα σύνολο 841 χαρακτηριστικών. Παρά την πολύτιμη συμβολή αυτής της εργασίας, διαπιστώσαμε ότι δεν εστιάζει σε βάθος στη μοντελοποίηση της διαδικασίας. Βασιζόμενοι σε αυτή την προσέγγιση, επιδιώξαμε να την επεκτείνουμε με τη χρήση προηγμένων τεχνικών και state-of-the-art μεθόδων που συνεχώς εξελίσσονται.

3 *Θεωρητικό υπόβαθρο*

3.1 *Οπτικοποίηση Αναλυτικής Δεδομένων*

Η οπτικοποίηση πληροφοριών ανέκαθεν βασιζόταν σε βασικά γεωμετρικά στοιχεία, όπως σημεία, γραμμές, καμπύλες και σχήματα, για την αναπαράσταση δεδομένων και την απεικόνιση των σχέσεων μεταξύ τους. Η **Οπτικοποίηση Αναλυτικής Δεδομένων** αποτελεί την πρακτική του σχεδιασμού και της δημιουργίας γραφικών ή οπτικών αναπαραστάσεων που διευκολύνουν την κατανόηση και την επικοινωνία μεγάλων όγκων πληροφοριών. Αυτές οι αναπαραστάσεις συμβάλλουν σημαντικά στην ανάλυση δεδομένων, καθώς επιτρέπουν τον εντοπισμό περιττής πληροφορίας, ελλιπών τιμών, σφαλμάτων στρογγυλοποίησης ή συσσώρευσης, σιωπηρών ορίων, ακραίων τιμών και ασυνήθιστων ομάδων. Επιπλέον, διευκολύνουν την εξερεύνηση της δομής των δεδομένων, την αναγνώριση τάσεων και συσχετίσεων, τον εντοπισμό τοπικών μοτίβων, την αξιολόγηση της απόδοσης μοντέλων και την αποτελεσματική παρουσίαση των αποτελεσμάτων.

Η διαδικασία αυτή απεικονίζεται συνοπτικά στην Εικόνα 3.1, όπου παρουσιάζεται η ροή από τα ακατέργαστα δεδομένα (σε μορφή πινάκων) προς την εξαγωγή γνώσης και, τελικά, την παραγωγή γραφικών αναπαραστάσεων. Το σχήμα αποτυπώνει την κεντρική ιδέα ότι η οπτικοποίηση δεν είναι απλώς μια διαδικασία απεικόνισης δεδομένων, αλλά μια γνωστική διεργασία που υποστηρίζει την ανάλυση και την εξαγωγή πολύτιμων πληροφοριών από σύνθετα δεδομένα.



Εικόνα 3.1: Μετάφραση Δεδομένων σε γραφικές οπτικοποίησης

Η οπτικοποίηση αποτελεί αναπόσπαστο εργαλείο για την εξερευνητική ανάλυση δεδομένων (EDA) και την εξόρυξη γνώσης από δεδομένα (Data Mining), ενώ είναι καθοριστική για τον έλεγχο της ποιότητας των δεδομένων και την εξοικείωση των αναλυτών με τα χαρακτηριστικά τους. Μέσα από την οπτικοποίηση, αποκαλύπτονται πρότυπα και σχέσεις που δεν είναι άμεσα ορατές μέσω αμιγώς στατιστικών μεθόδων, συμβάλλοντας στη διαμόρφωση νέων ερευνητικών υποθέσεων και ιδεών.

Παρά τα πλεονεκτήματα που προσφέρει, η ορθή ερμηνεία των γραφικών αναπαραστάσεων απαιτεί εμπειρία. Ο αναλυτής πρέπει να είναι σε θέση να αναγνωρίζει σημαντικά πρότυπα και συσχετίσεις, αποφεύγοντας τον κίνδυνο υπερ-ερμηνείας. Αυτός ο κίνδυνος προκύπτει όταν αποδίδεται υπερβολική σημασία σε τυχαία μοτίβα ή σε συσχετίσεις που δεν υποστηρίζονται επαρκώς από τα δεδομένα, οδηγώντας σε παραπλανητικά συμπεράσματα.

Η οπτικοποίηση δεδομένων αντιμετωπίζει πολυδιάστατες προκλήσεις, οι οποίες μπορούν να αναλυθούν ως εξής:

1. **Σύλληψη της Ανθρώπινης Αντίληψης:**
   Πώς επιλέγει ένας άνθρωπος να αναπαραστήσει οπτικά τα δεδομένα; Με ποια κριτήρια αξιολογεί μια οπτικοποίηση ως αποτελεσματική και γιατί προτιμά μια συγκεκριμένη μέθοδο παρουσίασης έναντι άλλης; Η κατανόηση αυτών των παραμέτρων είναι κρίσιμη για τη βελτιστοποίηση της διαδικασίας οπτικοποίησης.
2. **Αντίληψη και Κατανόηση Μεγάλων Όγκων Δεδομένων:**
   Τα ακατέργαστα δεδομένα, ειδικά σε μεγάλα σύνολα, είναι δύσκολο να ερμηνευτούν χωρίς την κατάλληλη οπτικοποίηση. Η απλή απεικόνιση των δεδομένων όπως είναι μπορεί να μην προσφέρει ευδιάκριτα αποτελέσματα, ιδιαίτερα για μη εξειδικευμένους χρήστες. Η πρόκληση έγκειται στην ανάπτυξη τεχνικών που καθιστούν τα δεδομένα πιο κατανοητά και λειτουργικά.
3. **Έλλειψη Αντικειμενικής Αλήθειας:**
   Η εξεύρεση της «βέλτιστης» οπτικοποίησης είναι μια υποκειμενική και χρονοβόρα διαδικασία, καθώς δεν υπάρχει κάποιο απόλυτο σημείο αναφοράς ή μια αντικειμενική αλήθεια που να καθορίζει ποια οπτικοποίηση είναι η καλύτερη. Η επιλογή της κατάλληλης απεικόνισης εξαρτάται συχνά από το πλαίσιο, τον σκοπό της ανάλυσης και το κοινό στο οποίο απευθύνεται.

Υπάρχουν πολλές σημαντικές ευκαιρίες για μελλοντική έρευνα στον τομέα της Οπτικοποίησης Αναλυτικής Δεδομένων. Απαιτείται η ανάπτυξη αρχών και κατευθυντήριων γραμμών που θα καθοδηγούν την επιλογή των κατάλληλων γραφικών αναπαραστάσεων από το πλήθος των διαθέσιμων επιλογών. Το ζητούμενο δεν είναι ο σχεδιασμός μιας μοναδικής, ‘ιδανικής’ οπτικοποίησης—εάν κάτι τέτοιο είναι εφικτό—αλλά η επιλογή ενός συνόλου

οπτικοποιήσεων που, σε συνδυασμό, προσφέρουν μια πιο ολοκληρωμένη και πολύπλευρη κατανόηση των δεδομένων. Παρομοίως με τη φωτογράφιση ενός πολύπλοκου αντικειμένου, όπου μία και μόνο φωτογραφία δεν επαρκεί για την πλήρη αποτύπωσή του, ενώ η λήψη φωτογραφιών από κάθε πιθανή γωνία είναι περιττή, έτσι και στην οπτικοποίηση δεδομένων απαιτείται μια ισορροπημένη προσέγγιση που αποκαλύπτει τις κρίσιμες πτυχές της πληροφορίας χωρίς υπερβολές.

3.2 *Μηχανική Μάθηση*

Η Μηχανική Μάθηση (Machine Learning) είναι ένας κλάδος της Τεχνητής Νοημοσύνης (Artificial Intelligence) και της Επιστήμης Υπολογιστών (Computer Science) που εστιάζει στη χρήση δεδομένων και αλγορίθμων, με στόχο να επιτρέψει σε συστήματα τεχνητής νοημοσύνης να μιμηθούν τον τρόπο που οι άνθρωποι μαθαίνουν και να εκπληρώσουν συγκεκριμένα έργα χωρίς την ανάγκη σαφών και ρητών οδηγιών. Η μηχανική μάθηση βρίσκει ευρεία εφαρμογή σε διάφορους τομείς, όπως η επεξεργασία φυσικής γλώσσας (Natural Language Processing), η όραση υπολογιστή (Computer Vision), η αναγνώριση ομιλίας (Speech Recognition), το φιλτράρισμα ηλεκτρονικής αλληλογραφίας (Email Filtering), καθώς και σε κλάδους όπως η γεωργία, η ιατρική, και η χρηματοοικονομική ανάλυση.

Ας αναλύσουμε το μαθησιακό σύστημα ενός μοντέλου μηχανικής μάθησης σε τρία κύρια στάδια:

1. **Διαδικασία Απόφασης (Decision Process):**
   Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται κυρίως για να πραγματοποιούν προβλέψεις ή ταξινομήσεις. Με βάση τα δεδομένα εισόδου (input data), τα οποία μπορεί να φέρουν ή όχι ετικέτες (labels), ο αλγόριθμος επιχειρεί να αναγνωρίσει πρότυπα και να δημιουργήσει εκτιμήσεις που αντικατοπτρίζουν τα χαρακτηριστικά των δεδομένων.
2. **Συνάρτηση Σφάλματος (Loss Function):**
   Η συνάρτηση σφάλματος αξιολογεί την ακρίβεια των προβλέψεων του μοντέλου. Σε περιπτώσεις όπου υπάρχουν γνωστά παραδείγματα (labeled data), η συνάρτηση συγκρίνει την πρόβλεψη του μοντέλου με την πραγματική τιμή, μετρώντας το σφάλμα μεταξύ τους. Αυτή η διαδικασία βοηθά στη βελτίωση της απόδοσης του μοντέλου.
3. **Διαδικασία Βελτιστοποίησης Μοντέλου (Model Optimization Process):**
   Η βελτιστοποίηση στοχεύει στη ρύθμιση των παραμέτρων του μοντέλου (όπως τα βάρη – weights) ώστε να μειωθεί το σφάλμα πρόβλεψης. Ο αλγόριθμος εφαρμόζει

μια επαναληπτική διαδικασία αξιολόγησης και βελτίωσης, προσαρμόζοντας αυτόματα τα βάρη έως ότου επιτευχθεί το επιθυμητό επίπεδο ακρίβειας.

Κατηγορίες Μηχανικής Μάθησης

Η μηχανική μάθηση διακρίνεται κυρίως σε δύο βασικές κατηγορίες: **Επιβλεπόμενη Μάθηση** (Supervised Learning) και **Μη Επιβλεπόμενη Μάθηση** (Unsupervised Learning). Η διάκριση αυτή βασίζεται στον τρόπο με τον οποίο ο αλγόριθμος "εκπαιδεύεται" χρησιμοποιώντας τα δεδομένα.

Επιβλεπόμενη Μάθηση (Supervised Learning)

Η επιβλεπόμενη μάθηση αποτελεί μια προσέγγιση κατά την οποία το μοντέλο εκπαιδεύεται με δεδομένα που φέρουν ετικέτες (labeled data). Κάθε παράδειγμα στο σύνολο εκπαίδευσης περιλαμβάνει δεδομένα εισόδου και την αντίστοιχη σωστή έξοδο (output label), επιτρέποντας στο μοντέλο να μάθει τη σχέση μεταξύ τους. Ο όρος «επιβλεπόμενη» υποδηλώνει ότι η διαδικασία εκμάθησης καθοδηγείται από αυτά τα γνωστά δεδομένα.

Κύριες κατηγορίες αλγορίθμων επιβλεπόμενης μάθησης είναι:

- **Ταξινόμηση (Classification):** Χρησιμοποιείται όταν οι έξοδοι ανήκουν σε ένα πεπερασμένο σύνολο κατηγοριών. Παράδειγμα αποτελεί η ταξινόμηση email ως "ανεπιθύμητο" ή "μη ανεπιθύμητο".
- **Παλινδρόμηση (Regression):** Εφαρμόζεται όταν η έξοδος είναι μια συνεχής αριθμητική τιμή. Παράδειγμα αποτελεί η πρόβλεψη της τιμής ενός ακινήτου βάσει συγκεκριμένων χαρακτηριστικών (μέγεθος, τοποθεσία κ.λπ.).

Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

Αντίθετα, στη μη επιβλεπόμενη μάθηση, το μοντέλο εκπαιδεύεται με δεδομένα που δεν φέρουν ετικέτες (unlabeled data). Ο στόχος είναι να ανακαλύψει κρυμμένα πρότυπα, σχέσεις και δομές μέσα στα δεδομένα χωρίς την καθοδήγηση προκαθορισμένων εξόδων. Αυτό καθιστά τη μη επιβλεπόμενη μάθηση ιδιαίτερα χρήσιμη για την ανάλυση μεγάλων και μη δομημένων συνόλων δεδομένων.

Κύριες κατηγορίες αλγορίθμων μη επιβλεπόμενης μάθησης είναι:

- **Ομαδοποίηση (Clustering):** Ανάθεση δεδομένων σε ομάδες (clusters) με βάση την ομοιότητά τους. Παράδειγμα αποτελεί η τμηματοποίηση πελατών σε ομάδες με κοινά χαρακτηριστικά.

- **Μείωση Διαστάσεων (Dimensionality Reduction):** Μείωση του πλήθους των μεταβλητών (features) σε ένα σύνολο δεδομένων για τη διευκόλυνση της οπτικοποίησης ή της επεξεργασίας.

4 *Αυτοματοποιημένη Οπτικοποίηση Αναλυτικής Δεδομένων*

4.1 *Προτεινόμενη Προσέγγιση*

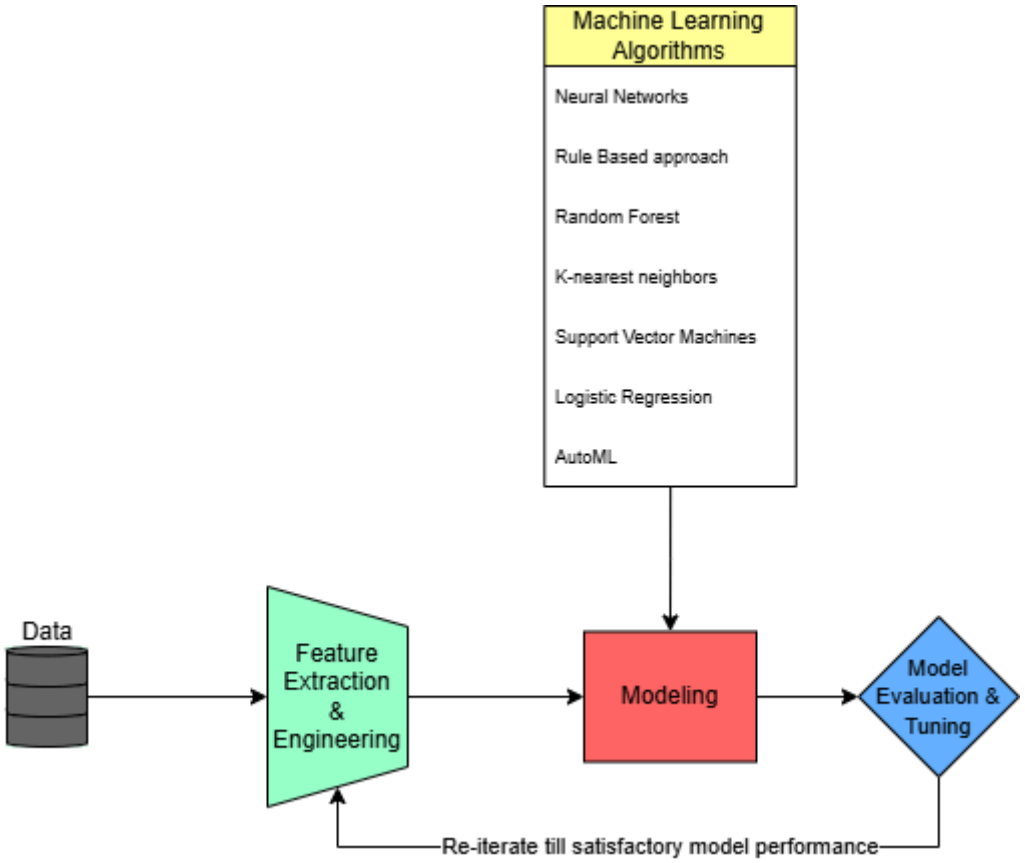
Στην παρούσα ενότητα προτείνετε μια προσέγγιση για την αυτοματοποίηση της διαδικασίας επιλογής της βέλτιστης οπτικοποίησης αναλυτικών δεδομένων. Η επιλογή κατάλληλων οπτικοποιήσεων απαιτεί εξειδικευμένη γνώση στην ανάλυση δεδομένων και προγραμματισμό, γεγονός που μπορεί να αποτελεί εμπόδιο για επαγγελματίες χωρίς τεχνικό υπόβαθρο. Η παραδοσιακή διαδικασία απαιτεί εμπειρία στην ανάλυση δεδομένων και εξοικείωση με κώδικα, γεγονός που αποτελεί εμπόδιο για πολλούς επαγγελματίες. Ως εκ τούτου, είναι σημαντικό να αναπτυχθούν μέθοδοι που καθιστούν τη διαδικασία πιο προσιτή, μειώνοντας την ανάγκη εξειδικευμένης παρέμβασης.

Μια πιθανή προσέγγιση σε αυτό το πρόβλημα είναι η ανάπτυξη ενός συστήματος μηχανικής μάθησης, το οποίο μπορεί να εκπαιδευτεί ώστε να προτείνει κατάλληλες οπτικοποιήσεις βάσει των χαρακτηριστικών των δεδομένων. Σε αυτό το πλαίσιο, μια διαδικασία εξαγωγής και επεξεργασίας χαρακτηριστικών (feature extraction & engineering) μπορεί να χρησιμοποιηθεί για να περιγράψει τις ιδιότητες των δεδομένων και να παρέχει την απαιτούμενη πληροφορία για τη λήψη αποφάσεων.

Στη συνέχεια, τα εξαγόμενα χαρακτηριστικά μπορούν να αξιοποιηθούν για την εκπαίδευση αλγορίθμων μηχανικής μάθησης, με στόχο την αναγνώριση προτύπων που καθορίζουν ποια οπτικοποίηση είναι πιο κατάλληλη σε κάθε περίπτωση. Η διαδικασία μπορεί να περιλαμβάνει αξιολόγηση των μοντέλων μέσω πειραμάτων και επαναληπτική βελτιστοποίηση (model evaluation & tuning) έως ότου το σύστημα επιτύχει ένα ικανοποιητικό επίπεδο ακρίβειας.

Η ροή εργασίας της προτεινόμενης μεθοδολογίας απεικονίζεται στο Σχήμα 4.1, το οποίο περιλαμβάνει τα βασικά στάδια της διαδικασίας: από τη συλλογή και προεπεξεργασία

δεδομένων, την εξαγωγή χαρακτηριστικών, τη μοντελοποίηση (modeling) και την αξιολόγηση και βελτιστοποίηση των μοντέλων. Η διαδικασία αυτή μπορεί να είναι επαναληπτική, καθώς η βελτιστοποίηση του μοντέλου μπορεί να συνεχίζεται μέχρι να επιτευχθεί το επιθυμητό επίπεδο απόδοσης. Τα επιμέρους βήματα αναλύονται εκτενώς στα επόμενα υποκεφάλαια.



Εικόνα 4.1: Σχήμα προσέγγισης

4.2 Εξαγωγή χαρακτηριστικών - Feature Extraction

Η εξαγωγή χαρακτηριστικών (feature extraction) αποτελεί βασικό στάδιο στην ανάπτυξη ενός συστήματος που βασίζεται σε μηχανική μάθηση, καθώς επιτρέπει τη μετατροπή των δεδομένων σε μια πιο συμπαγή και ενημερωτική αναπαράσταση. Μια βέλτιστη επιλογή χαρακτηριστικών μπορεί να βελτιώσει την αποδοτικότητα του συστήματος, να μειώσει τις απαιτήσεις αποθήκευσης και να ενισχύσει την ποιότητα της ανάλυσης. Η κατασκευή διανυσμάτων χαρακτηριστικών (feature vectors) αποτελεί μια από τις πιο συνηθισμένες και αποτελεσματικές μεθόδους αναπαράστασης δεδομένων, ειδικά σε προβλήματα ταξινόμησης

(classification) και παλινδρόμησης (regression). Κάθε χαρακτηριστικό μπορεί να είναι ποσοτικό ή ποιοτικό και αντιπροσωπεύει μια ιδιότητα ή μεταβλητή που περιγράφει μια σημαντική πτυχή των δεδομένων.

Μια προσέγγιση για την εξαγωγή χαρακτηριστικών στην περίπτωση δισδιάστατων γραφικών οπτικοποιήσεων (2D visualizations) θα μπορούσε να βασίζεται στην ανάλυση των δεδομένων σε δύο άξονες, τον X (οριζόντιο) και τον Y (κατακόρυφο). Για την κατασκευή ενός πινακοειδούς συνόλου χαρακτηριστικών (tabular feature set), προτείνεται η εξαγωγή:

- **Ξεχωριστών χαρακτηριστικών** για κάθε στήλη (άξονα),
- **Κοινών χαρακτηριστικών,** που συνδυάζουν πληροφορίες και από τους δύο άξονες (X και Y).

Μια πιθανή ταξινόμηση των χαρακτηριστικών περιλαμβάνει:

- **35 ξεχωριστά χαρακτηριστικά** για κάθε στήλη (X και Y),
- Και **13 συνδυαστικά χαρακτηριστικά** που περιγράφουν συσχετίσεις μεταξύ των δύο στηλών.

Συνολικά, ένα τέτοιο σύστημα θα μπορούσε να περιλαμβάνει **83 χαρακτηριστικά** (35 × 2 + 13), που καλύπτουν διάφορες διαστάσεις των δεδομένων. Για την διευκόλυνση της ανάλυσης και της κατανόησης αυτών των χαρακτηριστικών, είναι χρήσιμο να ταξινομούνται σε κατηγορίες, βάσει κοινών γνωρισμάτων και του ρόλου τους στη διαδικασία ανάλυσης δεδομένων. Μια πιθανή κατηγοριοποίηση περιλαμβάνει τις εξής ομάδες: Type, Uniqueness, Categorical, Quantitave, Sequence, General Pairwise και Statistical Pairwise. Ακολουθεί αναλυτική περιγραφή των χαρακτηριστικών που προτείνεται να εξαχθούν, ταξινομημένων σύμφωνα με τις παραπάνω κατηγορίες.

Type Features

1. Το Type είναι ένα μοναδικό χαρακτηριστικό για κάθε στήλη. Παίρνει κατηγορηματική τιμή (‘c’ ή ‘q’) και περιγράφει αν η στήλη περιέχει κατηγορηματικά (categorical) ή αριθμητικά (quantitative) δεδομένα, αντίστοιχα.

Uniqueness Features

Τα χαρακτηριστικά Uniqueness αναφέρονται σε κάθε στήλη και αφορούν την ποικιλομορφία των δεδομένων, τις διαστάσεις και την παρουσία ελλιπών τιμών:

2. num_unique: Ο αριθμός των μοναδικών στοιχείων στη στήλη.
3. num_entries: Ο συνολικός αριθμός των στοιχείων στην στήλη.

4. num_none: Ο αριθμός των ελλιπών στοιχείων στη στήλη (τιμές όπως ‘None’ ή κενές τιμές ‘’).
5. percentage_none: Το ποσοστό των ελλιπών στοιχείων (υπολογίζεται ως num_none / num_entries).
6. has_none: Λογική τιμή (‘True’ ή ‘False’) που υποδεικνύει αν υπάρχουν ελλιπείς τιμές.

Categorical Features

Αυτά τα χαρακτηριστικά εφαρμόζονται σε στήλες με κατηγορηματικά δεδομένα (categorical):

7. c_entropy: Η εντροπία της στήλης, που μετρά την ποικιλία ή την αβεβαιότητα των κατηγοριών.
8. mean_c_length: Ο μέσος όρος του μήκους των κατηγοριών, δηλαδή ο μέσος αριθμός χαρακτήρων που περιέχουν οι κατηγορίες στη στήλη. Για παράδειγμα, οι κατηγορίες "Yes", "No", "Maybe" έχουν μήκη 3, 2, και 5 αντίστοιχα.
9. median_c_length: Η διάμεσος του μήκους των κατηγοριών, δηλαδή η μεσαία τιμή στο σύνολο των μηκών.
10. min_c_length/max_c_length: Το μικρότερο και το μεγαλύτερο μήκος κατηγοριών στη στήλη.
11. std_c_length: Η τυπική απόκλιση του μήκους των κατηγοριών, που δείχνει πόσο ποικίλλουν τα μήκη μεταξύ τους.
12. percentage_mode_c: Το ποσοστό των παρατηρήσεων που ανήκουν στη συνηθέστερη (mode) κατηγορία της στήλης.
13. unique_categories: Ο αριθμός των μοναδικών κατηγοριών που υπάρχουν στην στήλη.

Quantitative Features

Αυτά τα χαρακτηριστικά εφαρμόζονται σε στήλες με αριθμητικά δεδομένα (quantitative):

14. sample_mean: Ο μέσος όρος των τιμών στη στήλη.
15. sample_median: Η διάμεσος των τιμών, δηλαδή η τιμή που χωρίζει τις τιμές σε δύο ίσα μέρη.
16. sample_var: Η διακύμανση των τιμών, που δείχνει πόσο πολύ απλώνονται οι τιμές γύρω από τον μέσο όρο.
17. sample_min/sample_max: Η ελάχιστη και η μέγιστη τιμή στη στήλη.
18. sample_std: τυπική απόκλιση, που μετρά τη μεταβλητότητα των τιμών.
19. coeff_var: Ο συντελεστής μεταβλητότητας (CV), που είναι το πηλίκο της τυπικής απόκλισης προς τον μέσο όρο.
20. range: Το εύρος των τιμών, δηλαδή η διαφορά μεταξύ της μέγιστης και της ελάχιστης τιμής.
21. q_entropy: Η εντροπία των αριθμητικών τιμών, που μετρά την τυχαιότητα ή αβεβαιότητα στην κατανομή τους.

22. kurt: Η κύρτωση (kurtosis), που μετρά πόσο “αιχμηρή” ή “επίπεδη” είναι η κατανομή σε σύγκριση με μια κανονική κατανομή. Υψηλή κύρτωση δείχνει απότομες κορυφές.
23. skewness: Η ασυμμετρία, που δείχνει αν η κατανομή των τιμών είναι συμμετρική γύρω από τον μέσο όρο ή έχει κλίση προς μια κατεύθυνση.
24. percent_outliers_15iqr/3iqr: Το ποσοστό των ακραίων τιμών (outliers) που εντοπίζονται χρησιμοποιώντας τον κανόνα του 1.5 × IQR ή 3 × IQR.
    - **IQR (Interquartile Range):** Είναι το διάστημα μεταξύ του 1ου (25%) και 3ου (75%) τεταρτημορίου.
    - **Κανόνας 1.5 × IQR:** Ένα στοιχείο θεωρείται ακραία τιμή αν βρίσκεται έξω από το διάστημα
    - Q1−1.5×IQR, Q3+1.5×IQR
    - Q1−1.5×IQR, Q3+1.5×IQR

      .
    - **Κανόνας 3 × IQR:** Παρόμοια μέθοδος αλλά πιο αυστηρή, που θεωρεί ακραίες τιμές εκείνες που είναι πολύ πιο μακριά από το μέσο.
25. percent_outliers_1_99: Το ποσοστό τιμών που βρίσκονται εκτός του εύρους που ορίζεται από το 1ο και το 99ο εκατοστημόριο.
26. percent_outliers_3std: Το ποσοστό των ακραίων τιμών που βρίσκονται πέρα από τις 3 τυπικές αποκλίσεις (3σ) από τον μέσο όρο.
27. has_outliers15iqr/has_outliers3iqr/has_outliers_1_99/has_outliers_3std: Λογικές τιμές (True/False) που δείχνουν αν υπάρχουν ακραίες τιμές, σύμφωνα με τα αντίστοιχα κριτήρια

Sequence Features

Αναφέρονται στην ταξινόμηση των δεδομένων σε μια στήλη:

28. sortedness: Μετρά τον βαθμό στον οποίο τα δεδομένα είναι ταξινομημένα (σε αύξουσα ή φθίνουσα σειρά).
29. is_sorted: Λογική τιμή (True ή False) που υποδεικνύει αν τα δεδομένα είναι πλήρως ταξινομημένα αριθμητικά ή αλφαβητικά.

General Pairwise Features

Αυτά τα χαρακτηριστικά αναφέρονται σε σχέσεις μεταξύ δύο στηλών, συγκεκριμένα μεταξύ των αξόνων X και Y στις δισδιάστατες οπτικοποιήσεις:

30. num_identical_elements: Ο αριθμός στοιχείων που είναι ακριβώς ίδια και στην ίδια θέση και στις δύο στήλες (X και Y).
31. has_shared_elements: Λογική τιμή (True/False) που δείχνει αν υπάρχουν κοινά στοιχεία στις δύο στήλες (αν num_identical_elements > 0).
32. percent_shared_elements: Το ποσοστό των κοινών στοιχείων σε σύγκριση με το συνολικό αριθμό στοιχείων.

33. num_shared_unique_elements: Ο αριθμός μοναδικών στοιχείων που εμφανίζονται και στις δύο στήλες, ανεξαρτήτως θέσης.
34. has_shared_unique_elements: Λογική τιμή που δείχνει αν υπάρχουν κοινά μοναδικά στοιχεία στις δύο στήλες (αν num_shared_unique_elements > 0).
35. percent_shared_unique_elements: Το ποσοστό των κοινών μοναδικών στοιχείων σε σύγκριση με τον μέγιστο αριθμό μοναδικών στοιχείων είτε της X είτε της Y στήλης.
36. identical_unique: Λογική τιμή που δείχνει αν τα σύνολα των μοναδικών στοιχείων στις δύο στήλες είναι ακριβώς ίδια, ανεξάρτητα από τη σειρά.

Statistical Pairwise Features

Τα Statistical Pairwise Features είναι χαρακτηριστικά που περιγράφουν στατιστικές σχέσεις μεταξύ δύο στηλών με αριθμητικά δεδομένα (quantitative) σε δισδιάστατες οπτικοποιήσεις. Αυτά τα χαρακτηριστικά βοηθούν στην κατανόηση της γραμμικής συσχέτισης και των διαφορών στην κατανομή μεταξύ των δύο μεταβλητών.

37. correlation_value: Η τιμή του συντελεστή συσχέτισης Pearson, ο οποίος μετρά τον βαθμό της γραμμικής σχέσης μεταξύ δύο συνόλων δεδομένων. Ο συντελεστής Pearson (r) κυμαίνεται από -1 έως 1:
    - Τιμή **+1**: τέλεια θετική συσχέτιση (καθώς αυξάνονται τα δεδομένα στον X άξονα, αυξάνονται και στον Y).
    - Τιμή **-1**: τέλεια αρνητική συσχέτιση (καθώς αυξάνονται τα δεδομένα στον X, μειώνονται στον Y).
    - Τιμή **0**: καμία γραμμική συσχέτιση μεταξύ των δεδομένων.

    Σημείωση: Η συσχέτιση Pearson εξετάζει μόνο γραμμικές σχέσεις. Αν η σχέση είναι μη-γραμμική, η τιμή μπορεί να είναι κοντά στο 0 ακόμα και αν υπάρχει ισχυρή συσχέτιση.
38. correlation_p: Η τιμή p για τον έλεγχο σημαντικότητας της συσχέτισης Pearson. Η τιμή p δείχνει την πιθανότητα η παρατηρούμενη συσχέτιση να είναι αποτέλεσμα τυχαίας διακύμανσης.
    - Αν $p < 0.05$, θεωρούμε ότι η συσχέτιση είναι στατιστικά σημαντική, δηλαδή είναι απίθανο να προέκυψε τυχαία.
    - Αν $p > 0.05$, η συσχέτιση μπορεί να οφείλεται σε τύχη και δεν είναι στατιστικά ισχυρή.
39. ks_statistic: Η στατιστική τιμή του Kolmogorov-Smirnov (KS), η οποία χρησιμοποιείται για τη σύγκριση των κατανομών δύο συνόλων δεδομένων. Ο έλεγχος KS μετρά τη μέγιστη απόκλιση μεταξύ των σωρευτικών (cumulative) κατανομών δύο δειγμάτων.
    - Τιμή κοντά στο 0 υποδεικνύει ότι οι κατανομές είναι παρόμοιες.
    - Υψηλότερες τιμές δείχνουν μεγαλύτερη απόκλιση μεταξύ των δύο κατανομών.

40. ks_p: Η τιμή p για τη στατιστική KS. Όπως και με τη συσχέτιση Pearson, η τιμή p δείχνει αν η διαφορά που παρατηρείται μεταξύ των κατανομών είναι στατιστικά σημαντική.
    - Αν $p < 0.05$, θεωρούμε ότι υπάρχει σημαντική διαφορά μεταξύ των δύο κατανομών.
    - Αν $p > 0.05$, δεν υπάρχουν επαρκή στοιχεία για να απορρίψουμε την υπόθεση ότι οι δύο κατανομές είναι παρόμοιες.
41. correlation_significant_005: Λογική τιμή (True ή False) που δείχνει αν η συσχέτιση Pearson είναι στατιστικά σημαντική, με βάση επίπεδο σημαντικότητας 0.05 (δηλαδή, 5% πιθανότητα σφάλματος).
    - Αν η τιμή $p < 0.05$, τότε η συσχέτιση είναι σημαντική (True).
    - Αν η τιμή $p \geq 0.05$, δεν είναι σημαντική (False).

    Σημείωση: Αυτό το χαρακτηριστικό βοηθά να διαπιστώσουμε γρήγορα αν η συσχέτιση που υπολογίσαμε είναι αξιόπιστη.
42. ks_significant_005: Λογική τιμή (True ή False) που δείχνει αν η διαφορά μεταξύ των κατανομών είναι στατιστικά σημαντική σύμφωνα με τη στατιστική KS.
    - Αν η τιμή $p < 0.05$, θεωρείται ότι υπάρχει σημαντική διαφορά (True).
    - Αν $p \geq 0.05$, δεν παρατηρείται στατιστικά σημαντική διαφορά (False).

    Σημείωση: Όπως και στο correlation_significant_005, αυτή η παράμετρος απλοποιεί την κατανόηση της στατιστικής σημαντικότητας χωρίς να χρειάζεται να εξετάσουμε κάθε φορά τις τιμές p.

Κατά την εξαγωγή χαρακτηριστικών, ορισμένες κατηγορίες εξαρτώνται από τον τύπο των δεδομένων που περιέχει κάθε στήλη, όπως αυτός ορίζεται από το χαρακτηριστικό Type. Συγκεκριμένα, αν μια στήλη περιέχει αριθμητικά δεδομένα (quantitative), τότε τα χαρακτηριστικά που σχετίζονται με κατηγορηματικά δεδομένα (categorical) δεν είναι εφαρμόσιμα. Αντίστοιχα, αν μια στήλη περιέχει κατηγορηματικά δεδομένα, τότε τα χαρακτηριστικά που αφορούν αριθμητικές τιμές ή στατιστικές σχέσεις μεταξύ αριθμητικών μεταβλητών (Statistical Pairwise) δεν είναι συναφή.

Μια πιθανή προσέγγιση σε αυτές τις περιπτώσεις είναι η διατήρηση των αντίστοιχων τιμών ως κενές ή η εφαρμογή συγκεκριμένων στρατηγικών διαχείρισης, ανάλογα με το εκάστοτε μοντέλο μηχανικής μάθησης και τον τρόπο που αξιοποιεί τα χαρακτηριστικά κατά την εκπαίδευση. Οι πιθανές μέθοδοι διαχείρισης αυτών των τιμών και η επίδρασή τους στην απόδοση των μοντέλων αναλύονται στα επόμενα υποκεφάλαια.

Ετικέτες - Labels

Στην επιβλεπόμενη μηχανική μάθηση (Supervised Learning), το μοντέλο εκπαιδεύεται με δεδομένα που φέρουν ετικέτες (labels), οι οποίες αντιπροσωπεύουν την επιθυμητή έξοδο για

κάθε δεδομένη είσοδο. Όπως αναλύθηκε στο προηγούμενο κεφάλαιο, η διαδικασία μάθησης περιλαμβάνει την αναγνώριση προτύπων στα δεδομένα, την αξιολόγηση μέσω μιας συνάρτησης σφάλματος και τη βελτιστοποίηση του μοντέλου με στόχο τη βελτίωση της ακρίβειάς του.

Μια προτεινόμενη προσέγγιση για την αυτοματοποίηση της επιλογής της κατάλληλης γραφικής αναπαράστασης μπορεί να βασιστεί στην ταξινόμηση των δεδομένων σε κατηγορίες, με στόχο την εύρεση της πιο κατάλληλης οπτικοποίησης. Η εκπαίδευση ενός μοντέλου μηχανικής μάθησης μπορεί να διαμορφωθεί ώστε να μαθαίνει συσχετίσεις μεταξύ χαρακτηριστικών των δεδομένων και των προτιμώμενων οπτικοποιήσεων, επιτρέποντας έτσι στο σύστημα να προτείνει το κατάλληλο γράφημα βάσει των ιδιοτήτων του εκάστοτε συνόλου δεδομένων.

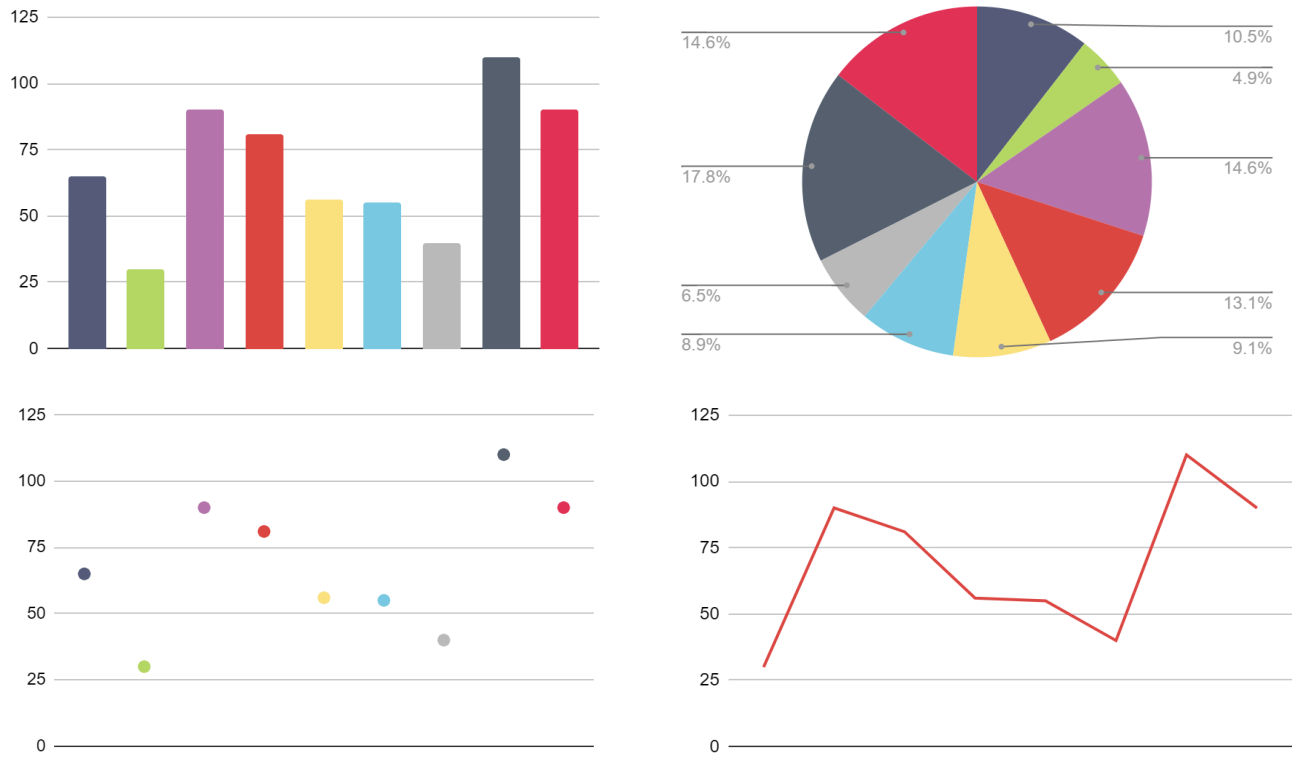
Ένα σύστημα σύστασης οπτικοποιήσεων μπορεί να αξιοποιεί καθιερωμένες και ευρέως χρησιμοποιούμενες κατηγορίες διαγραμμάτων, όπως:

1. Bar Chart (Ραβδόγραμμα): Κατάλληλο όταν ένας από τους δύο άξονες περιέχει κατηγορηματικά δεδομένα και ο στόχος είναι η σύγκριση ποσοτήτων μεταξύ κατηγοριών (π.χ. σύγκριση πωλήσεων μεταξύ προϊόντων).
2. Line Chart (Γραμμικό διάγραμμα): Χρησιμοποιείται σε συνεχή δεδομένα, ειδικά όταν υπάρχει χρονική διάσταση, επιτρέποντας την ανίχνευση τάσεων και μοτίβων (π.χ. μεταβολή θερμοκρασίας στη διάρκεια ενός έτους).
3. Pie Chart (Κυκλικό διάγραμμα): Αναπαριστά τα δεδομένα ως τμήματα ενός συνόλου, συνήθως ως ποσοστά, και είναι χρήσιμο όταν υπάρχει περιορισμένος αριθμός κατηγοριών (π.χ. ποσοστιαία συμμετοχή διαφημιστικών καμπανιών στην επισκεψιμότητα ενός ιστότοπου).
4. Scatter Plot (Διάγραμμα διασποράς): Εντοπίζει συσχετίσεις και μοτίβα μεταξύ δύο αριθμητικών μεταβλητών, καθιστώντας το χρήσιμο για στατιστική ανάλυση και διερεύνηση σχέσεων μεταξύ δεδομένων (π.χ. ύψος και βάρος ατόμων).

Η Εικόνα 4.2 παρουσιάζει παραδείγματα αυτών των τεσσάρων κατηγοριών γραφικών οπτικοποίησης, δίνοντας μια οπτική αναπαράσταση του τρόπου με τον οποίο τα διαφορετικά είδη δεδομένων αποτυπώνονται σε κάθε τύπο διαγράμματος.

Παρότι τα παραπάνω διαγράμματα είναι από τα πιο διαδεδομένα στην ανάλυση δεδομένων, μπορούν να χρησιμοποιηθούν και άλλες δισδιάστατες (2D) γραφικές οπτικοποιήσεις, ανάλογα με τα χαρακτηριστικά των δεδομένων και τις ανάγκες της ανάλυσης. Παραδείγματα τέτοιων γραφημάτων είναι τα Histogram (Ιστόγραμμα), που χρησιμοποιούνται για την απεικόνιση της

κατανομής συχνοτήτων, καθώς και τα Box Plots (Διαγράμματα Box-and-Whisker), τα οποία παρέχουν σύνοψη στατιστικών χαρακτηριστικών μιας αριθμητικής μεταβλητής. Η ενσωμάτωση επιπλέον τύπων οπτικοποιήσεων μπορεί να επεκτείνει τις δυνατότητες ενός συστήματος σύστασης, καθιστώντας το πιο ευέλικτο και προσαρμόσιμο σε διαφορετικά σύνολα δεδομένων.



Εικόνα 4.2: Τα είδη γραφικών παραστάσεων που έχουν επιλεγεί ως ετικέτες bar, pie, scatter και line chart

4.3 Επεξεργασία Χαρακτηριστικών Εισόδου - Feature Processing and Engineering

Τα δεδομένα σε ανεπεξέργαστη μορφή είναι συχνά ακατάστατα, ασυνεπή και ατελή, γεγονός που καθιστά απαραίτητη τη διαδικασία προεπεξεργασίας πριν από οποιαδήποτε ανάλυση ή μοντελοποίηση. Ανάλογα με την πηγή τους και τον σκοπό χρήσης τους, μπορεί να απαιτούνται διορθώσεις και μετασχηματισμοί, ώστε να βελτιωθεί η ποιότητά τους και να αξιοποιηθούν αποδοτικότερα στα επόμενα στάδια μιας μεθοδολογίας.

Μια τυπική διαδικασία προεπεξεργασίας δεδομένων μπορεί να περιλαμβάνει την αντιμετώπιση των εξής προκλήσεων:

1. Θόρυβος στα δεδομένα: Πολλά σύνολα δεδομένων περιέχουν ακραίες τιμές (outliers), που μπορεί να προκύψουν από σφάλματα κατά την καταγραφή, πληκτρολόγηση ή μεταφορά των δεδομένων. Τέτοιες ανωμαλίες εμφανίζονται συχνά λόγω ανθρώπινων λαθών, τεχνικών δυσλειτουργιών ή συνθηκών που δεν αντιπροσωπεύουν την πραγματικότητα.
2. Διαχείριση κατηγορηματικών χαρακτηριστικών (Categorical Features): Ορισμένοι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να επεξεργαστούν κατηγορηματικά δεδομένα (categorical features) και απαιτούν μετατροπή σε αριθμητική μορφή. Αυτή η διαδικασία μπορεί να περιλαμβάνει τεχνικές όπως one-hot encoding, ordinal encoding ή target encoding, ανάλογα με τη φύση των δεδομένων και το πρόβλημα που επιλύεται.
3. Διαφορά στην κλίμακα των χαρακτηριστικών (Feature Scaling): Τα χαρακτηριστικά ενός συνόλου δεδομένων μπορεί να έχουν διαφορετικές μονάδες μέτρησης και κλίμακες, γεγονός που μπορεί να επηρεάσει την απόδοση αλγορίθμων που είναι ευαίσθητοι σε διαφορές κλίμακας (π.χ. γραμμική παλινδρόμηση, SVM, k-NN). Συνήθεις τεχνικές αντιμετώπισης περιλαμβάνουν τη κανονικοποίηση (min-max scaling) και την τυποποίηση (standardization) ώστε να εξισορροπηθεί η επιρροή κάθε χαρακτηριστικού.
4. Αντιμετώπιση ελλιπών δεδομένων: Σε πολλές περιπτώσεις, οι πίνακες χαρακτηριστικών περιλαμβάνουν κενές τιμές, είτε λόγω έλλειψης δεδομένων στη συλλογή είτε λόγω στρατηγικής εξαγωγής χαρακτηριστικών (feature extraction). Ορισμένοι αλγόριθμοι δεν μπορούν να εκπαιδευτούν με ελλιπή δεδομένα, γεγονός που καθιστά απαραίτητη την αντικατάσταση (imputation) των κενών τιμών μέσω τεχνικών όπως μέσος όρος, διάμεσος, προχωρημένες μέθοδοι όπως KNN imputation ή μοντέλα πρόβλεψης τιμών.

Η επιλογή της κατάλληλης τεχνικής προεπεξεργασίας εξαρτάται από τη φύση των δεδομένων, τον αλγόριθμο που χρησιμοποιείται και τις απαιτήσεις του προβλήματος. Η σωστή διαχείριση αυτών των προκλήσεων συμβάλλει στην αύξηση της ακρίβειας και της αξιοπιστίας των μοντέλων μηχανικής μάθησης και στη βελτίωση της συνολικής ποιότητας της ανάλυσης δεδομένων.

Για τη βελτίωση της ποιότητας των δεδομένων και την αντιμετώπιση των προκλήσεων που αναφέρθηκαν, μπορούν να αξιοποιηθούν διάφορες μέθοδοι προεπεξεργασίας. Αυτές στοχεύουν στη μείωση του θορύβου στα δεδομένα, την αντιμετώπιση των ελλιπών τιμών, τη

μετατροπή κατηγορηματικών χαρακτηριστικών σε αριθμητικά και την κλιμάκωση των μεταβλητών για την ομαλή λειτουργία των αλγορίθμων μηχανικής μάθησης.

Διαχείριση Ακραίων Τιμών - Handling Outliers

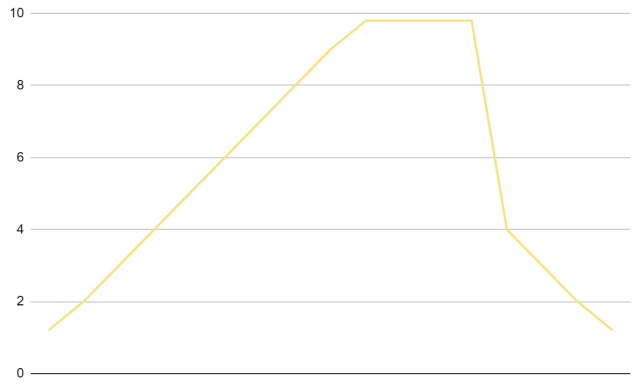
Οι ακραίες τιμές (outliers) μπορούν να επηρεάσουν σημαντικά την ακρίβεια και την αξιοπιστία της ανάλυσης δεδομένων, καθώς συχνά οφείλονται σε σφάλματα καταγραφής, μη ρεαλιστικές μεταβολές ή ειδικές συνθήκες που δεν αντιπροσωπεύουν το γενικό μοτίβο των δεδομένων.

Για την αποτελεσματική διαχείρισή τους, απαιτείται η επιλογή μιας κατάλληλης τεχνικής που να ελαχιστοποιεί την επίδρασή τους χωρίς να απομακρύνει σημαντικές πληροφορίες. Αυτή η διαδικασία μπορεί να περιλαμβάνει στατιστικές μεθόδους, όπως διαγνωστικούς κανόνες ανίχνευσης outliers (π.χ. interquartile range - IQR, Z-score), ή πιο σύνθετες τεχνικές, όπως μετασχηματισμούς δεδομένων και μοντέλα προσαρμογής ακραίων τιμών.

Η Εικόνα 4.3.1 παρουσιάζει ένα σύνολο δεδομένων πριν από την επεξεργασία, όπου παρατηρούνται ακραίες τιμές που επηρεάζουν τη συνολική κατανομή. Αντίστοιχα, η Εικόνα 4.3.2 απεικονίζει τη βελτίωση της ποιότητας των δεδομένων μετά την εφαρμογή μιας κατάλληλης μεθόδου αντιμετώπισης outliers, διατηρώντας τα βασικά χαρακτηριστικά της κατανομής.

Η επιλογή της καταλληλότερης τεχνικής εξαρτάται από τη φύση των δεδομένων και τον σκοπό της ανάλυσης, διασφαλίζοντας ότι οι μεταγενέστερες μοντελοποιήσεις βασίζονται σε πιο αντιπροσωπευτικά και αξιόπιστα δεδομένα.

Εικόνα 4.3.1: Line γραφική με ακραίες τιμές λόγω θορύβου



Εικόνα 4.3.2: Line γραφική μετά την αφαίρεση του θορύβου

Κωδικοποίηση Κατηγορηματικών Δεδομένων – One-Hot Encoding

Πολλοί αλγόριθμοι μηχανικής μάθησης δεν μπορούν να επεξεργαστούν κατηγορηματικές μεταβλητές (categorical features) και απαιτούν τη μετατροπή τους σε αριθμητική μορφή. Μία προτεινόμενη μέθοδος είναι η One-Hot Encoding, κατά την οποία κάθε κατηγορία ενός χαρακτηριστικού αναπαρίσταται από ένα νέο δυαδικό χαρακτηριστικό. Στην κατηγορία στην οποίο ανήκει το δεδομένο παίρνει τιμή 1 και στις υπόλοιπες παίρνει την τιμή 0. Η εναλλακτική μέθοδος, label encoding, δεν εισάγει νέα χαρακτηριστικά αλλά αντικαθιστά κάθε κατηγορία με έναν αυξανόμενο αριθμό. Αυτό έχει σαν αποτέλεσμα να προσθέτει μια προτεραιότητα(ordinality) στις κατηγορίες με τον μεγαλύτερο αριθμό.

Για παράδειγμα, αν το αρχικό χαρακτηριστικό είναι το *"Φρούτο"* με κατηγορίες *"Μήλο"*, *"Μάνγκο"*, *"Πορτοκάλι"* (πίνακας 4.3.1), δημιουργούνται τρία νέα χαρακτηριστικά.

Πίνακας 4.3.1: Δεδομένα φρούτων και αντίστοιχων τιμών τους

Fruit	Price
apple	5
mango	10

apple	15
orange	20

Εφαρμόζοντας one-hot encoding παράγεται το αποτέλεσμα που φαίνεται στον πίνακα 4.3.2:

Πίνακας 4.3.2: Δεδομένα φρούτων και αντίστοιχων τιμών τους μετά την εφαρμογή του One-Hot Encoding

Fruit_apple	Fruit_mango	Fruit_orange	Price
1	0	0	5
0	1	0	10
1	0	0	15
0	0	1	20

Κανονικοποίηση και Κλιμάκωση Δεδομένων – Data Scaling

Τα χαρακτηριστικά των δεδομένων συχνά έχουν διαφορετικές κλίμακες, γεγονός που μπορεί να επηρεάσει την απόδοση πολλών αλγορίθμων μηχανικής μάθησης. Για παράδειγμα, μια μεταβλητή μπορεί να παίρνει τιμές από 1 έως 1000, ενώ μια άλλη μπορεί να είναι κανονικοποιημένο ποσοστό μεταξύ 0 και 1. Σε τέτοιες περιπτώσεις, οι αλγόριθμοι όπως k-Nearest Neighbors (k-NN), Support Vector Machines (SVM) και Νευρωνικά Δίκτυα τείνουν να δίνουν μεγαλύτερη βαρύτητα στις μεταβλητές με μεγαλύτερη αριθμητική κλίμακα, επηρεάζοντας τα αποτελέσματα της ανάλυσης.

Για την αντιμετώπιση αυτού του ζητήματος, εφαρμόζεται Κανονικοποίηση (Standardization), μια μέθοδος που μετασχηματίζει τα δεδομένα έτσι ώστε να έχουν μηδενικό μέσο όρο (0) και τυπική απόκλιση (1). Ο μετασχηματισμός πραγματοποιείται ως εξής:

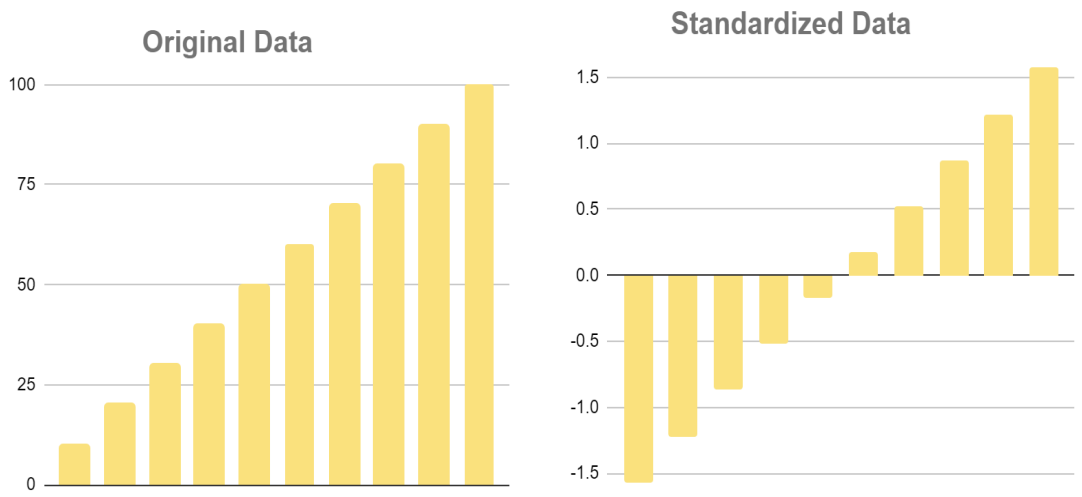
$$X' = \frac{X - \mu}{\sigma}$$

όπου:

- X’ είναι η κλιμακούμενη τιμή,
- X είναι η αρχική τιμή,
- μ είναι ο μέσος όρος των τιμών του χαρακτηριστικού,
- σ είναι η τυπική απόκλιση των τιμών του χαρακτηριστικού.

Αυτή η προσέγγιση διασφαλίζει ότι όλα τα χαρακτηριστικά βρίσκονται στην ίδια σχετική κλίμακα, επιτρέποντας στους αλγόριθμους να λειτουργούν βέλτιστα χωρίς να επηρεάζονται από τις απόλυτες τιμές των δεδομένων.

Στην Εικόνα 4.3.3, παρουσιάζεται μια σύγκριση μεταξύ των αρχικών δεδομένων (original data) και των κανονικοποιημένων δεδομένων (standardized data). Όπως φαίνεται, πριν από την κανονικοποίηση, τα χαρακτηριστικά παρουσιάζουν μεγάλες διακυμάνσεις στις τιμές τους, ενώ μετά τον μετασχηματισμό αποκτούν ομοιόμορφη κλίμακα με κεντρική τιμή το 0 και κατανομή που αντικατοπτρίζει τη σχετική απόκλιση των δεδομένων. Αυτή η μετατροπή συμβάλλει σημαντικά στη σταθερότητα των υπολογισμών και στην καλύτερη απόδοση των μοντέλων μηχανικής μάθησης.



Εικόνα 4.3.3: Standardization Δεδομένων πριν και μετά

Αντιμετώπιση Ελλιπών Δεδομένων – Data Imputation

Η απουσία δεδομένων είναι ένα συνηθισμένο φαινόμενο, καθώς τα χαρακτηριστικά μπορεί να έχουν κενές τιμές λόγω απώλειας πληροφοριών ή σκόπιμης εξαγωγής χαρακτηριστικών που δεν ισχύουν για όλους τους τύπους δεδομένων. Πολλοί αλγόριθμοι δεν μπορούν να εκπαιδευτούν με ελλιπή δεδομένα, επομένως απαιτείται αντικατάσταση των κενών τιμών (imputation).

Μια βασική προσέγγιση είναι η αντικατάσταση με στατιστικές τιμές, όπως:

- Ο μέσος όρος (mean) για αριθμητικά δεδομένα.
- Η διάμεσος (median) για να μειωθεί η επίδραση ακραίων τιμών.
- Η συχνότερη τιμή (mode) για κατηγορηματικά δεδομένα.

Αυτή η διαδικασία επιτρέπει την πληρότητα του συνόλου δεδομένων χωρίς να αλλοιώνει σημαντικά τις πληροφορίες. Στο παράδειγμα στον πίνακα 4.3.3 υπάρχουν κατηγορηματικά δεδομένα, το Φύλλο(Gender), και αριθμητικά, η Ηλικία(Age). Και στις δύο στήλες υπάρχουν ελλιπής τιμές.

Αυτή η διαδικασία επιτρέπει την πληρότητα του συνόλου δεδομένων χωρίς να αλλοιώνει σημαντικά τις πληροφορίες. Στο παράδειγμα στον πίνακα 4.3.3 υπάρχουν κατηγορηματικά δεδομένα, το Φύλλο(Gender), και αριθμητικά, η Ηλικία(Age). Και στις δύο στήλες υπάρχουν ελλιπής τιμές.

Πίνακας 4.3.3: Δεδομένα Φύλου και Ηλικίας, με κόκκινο χρώμα τα δεδομένα που λείπουν

Gender	Age
Male	25
Female	NaN
Female	30
Male	22
Female	NaN
NaN	27

Πιο κάτω, στον πίνακα 4.3.4, φαίνεται το αποτέλεσμα της αντιμετώπισης ελλιπών δεδομένων(data imputation) με την αντικατάσταση των κατηγορηματικών(categorical) πεδίων με την συχνότερη τιμή(mode) και των αριθμητικών(quantitative) δεδομένων με το μέσο όρο(mean).

Πίνακας 4.3.4: Δεδομένα Φύλου και Ηλικίας, με κίτρινο χρώμα τα δεδομένα που έχουν συμπληρωθεί με την χρήση Data Imputation

Gender	Age
Male	25
Female	26
Female	30
Male	22

Female	26
Female	27

4.4 Μοντελοποίηση - Modeling

Η μοντελοποίηση (modeling) αποτελεί βασικό στάδιο στη διαδικασία της προτεινόμενης μεθοδολογίας, καθώς επιτρέπει την εκμάθηση μοτίβων από τα εξαγόμενα χαρακτηριστικά των δεδομένων και την αυτόματη επιλογή της καταλληλότερης οπτικοποίησης. Το πρόβλημα διαμορφώνεται ως πρόβλημα ταξινόμησης (classification), στο οποίο κάθε είσοδος πρέπει να αντιστοιχιστεί σε μία από τις διαθέσιμες κατηγορίες οπτικοποιήσεων που έχουν επιλεγεί (*bar, line, scatter, pie etc.*).

Για τη διαμόρφωση μιας αποτελεσματικής και γενικεύσιμης προσέγγισης, προτείνεται η χρήση έξι διαφορετικών αλγορίθμων μηχανικής μάθησης, που καλύπτουν ένα ευρύ φάσμα τεχνικών ταξινόμησης και είναι κατάλληλοι για δεδομένα σε μορφή πίνακα (tabular data). Οι αλγόριθμοι που προτείνονται είναι οι εξής:

- Νευρωνικά Δίκτυα (Neural Networks): Μοντέλα με βαθιές αρχιτεκτονικές, ικανά να ανιχνεύουν πολύπλοκες σχέσεις στα δεδομένα.
- Τυχαία Δάση (Random Forest): Συνδυασμός πολλαπλών δέντρων απόφασης (decision trees) για την ενίσχυση της ακρίβειας και της γενίκευσης.
- k-Nearest Neighbors (k-NN): Ταξινόμηση με βάση την ομοιότητα μεταξύ των δειγμάτων, όπου κάθε νέο δείγμα ταξινομείται σύμφωνα με τους κοντινότερους γείτονές του.
- Διανυσματικές Μηχανές Υποστήριξης (Support Vector Machines - SVM): Δημιουργία υπερεπιπέδων (hyperplanes) για τη διάκριση κατηγοριών με υψηλή ακρίβεια.
- Λογιστική Παλινδρόμηση (Logistic Regression): Γραμμικό μοντέλο που εκτιμά την πιθανότητα κάθε κατηγορίας βάσει των χαρακτηριστικών εισόδου.
- AutoML: Αυτόματη επιλογή και βελτιστοποίηση μοντέλων μέσω μετα-μάθησης (meta-learning) και εξερεύνησης διαφορετικών αρχιτεκτονικών.

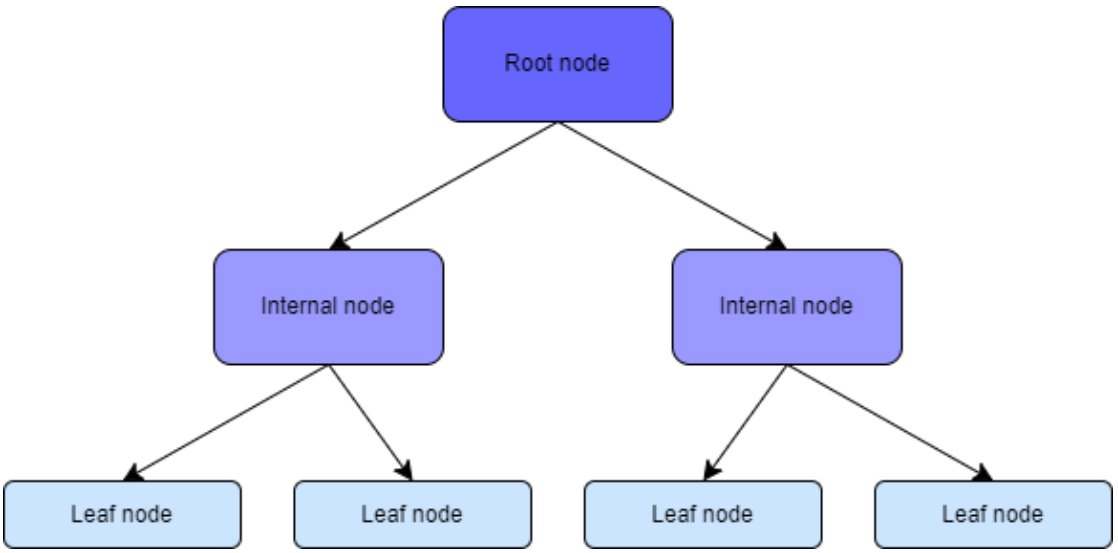
Η επιλογή αυτών των αλγορίθμων βασίζεται στη δυνατότητά τους να προσφέρουν διαφορετικά πλεονεκτήματα όσον αφορά την ακρίβεια, την ερμηνευσιμότητα και τη γενίκευση των αποτελεσμάτων. Η χρήση διαφορετικών αλγορίθμων επιτρέπει τη σύγκριση

των αποτελεσμάτων και την επιλογή της πιο αποδοτικής μεθόδου για το πρόβλημα, βελτιώνοντας τη συνολική απόδοση του συστήματος.

4.4.1 Random Forest

Ο αλγόριθμος Decision Tree βασίζεται στη διαδικασία ιεραρχικής λήψης αποφάσεων, όπου τα δεδομένα διαχωρίζονται διαδοχικά σε μικρότερα υποσύνολα. Ξεκινά με έναν ριζικό κόμβο (root node), ο οποίος δεν έχει εισερχόμενους κλάδους και αποτελεί το σημείο εκκίνησης της διαδικασίας ταξινόμησης. Στη συνέχεια, οι εξερχόμενοι κλάδοι από τον ριζικό κόμβο κατευθύνονται σε εσωτερικούς κόμβους απόφασης (decision nodes), όπου πραγματοποιούνται αξιολογήσεις βασισμένες σε συγκεκριμένα χαρακτηριστικά των δεδομένων.

Ο στόχος αυτών των κόμβων είναι η δημιουργία ομοιογενών υποσυνόλων, τα οποία τελικά οδηγούν σε κόμβους φύλλων (leaf nodes) ή τερματικούς κόμβους, όπου δεν πραγματοποιούνται περαιτέρω διαχωρισμοί. Οι κόμβοι φύλλων αντιπροσωπεύουν τις τελικές προβλέψεις του μοντέλου, δηλαδή τις κατηγορίες ταξινόμησης ή τις αριθμητικές τιμές σε περιπτώσεις παλινδρόμησης. Η Εικόνα 4.4.1.1 απεικονίζει τη δομή ενός Decision Tree, όπου παρουσιάζεται η διαδικασία διαδοχικού διαχωρισμού των δεδομένων.



Εικόνα 4.4.1.1: Decision Tree

Το Random Forest αποτελεί μια επέκταση της προσέγγισης των Decision Trees, η οποία ενσωματώνει πολλαπλά δέντρα απόφασης για τη βελτίωση της απόδοσης του μοντέλου. Πρόκειται για έναν ευέλικτο αλγόριθμο που μπορεί να εφαρμοστεί τόσο σε προβλήματα

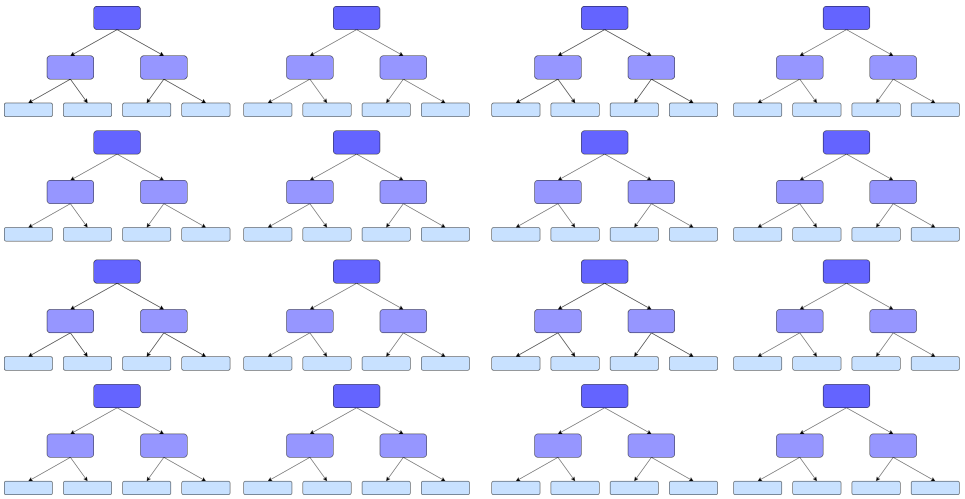
ταξινόμησης (classification) όσο και σε προβλήματα παλινδρόμησης (regression). Η βασική του λειτουργία περιλαμβάνει την κατασκευή ενός συνόλου από Decision Trees, όπου η τελική απόφαση λαμβάνεται με βάση τις προβλέψεις πολλών διαφορετικών δέντρων. Στα προβλήματα ταξινόμησης, η έξοδος του Random Forest καθορίζεται από τη δημοφιλέστερη κατηγορία (majority vote) μεταξύ των δέντρων, ενώ στα προβλήματα παλινδρόμησης, η πρόβλεψη προκύπτει από τον μέσο όρο των αποτελεσμάτων των επιμέρους δέντρων.

Η διαδικασία κατασκευής του Random Forest περιλαμβάνει τα εξής βήματα:

1. Bootstrap Aggregating (Bagging): Δημιουργία τυχαίων υποσυνόλων δεδομένων (bootstrapped datasets) από το αρχικό σύνολο δεδομένων. Το ίδιο δείγμα μπορεί να επιλεγεί περισσότερες από μία φορές.
2. Κατασκευή Decision Trees: Εκπαίδευση πολλαπλών Decision Trees, όπου κάθε δέντρο βασίζεται σε διαφορετικό τυχαίο υποσύνολο δεδομένων και ένα τυχαίο υποσύνολο χαρακτηριστικών (features).
3. Συνδυασμός Αποτελεσμάτων: Για κάθε νέο δείγμα, το Random Forest συνδυάζει τις προβλέψεις από όλα τα δέντρα για να παράγει το τελικό αποτέλεσμα.

Αυτή η μέθοδος ενισχύει τη γενίκευση του μοντέλου, καθιστώντας το πιο ανθεκτικό στην υπερπροσαρμογή (overfitting) σε σχέση με ένα μεμονωμένο Decision Tree. Επιπλέον, μέσω της χρήσης του Out-Of-Bag Dataset (OOB Dataset), το οποίο αποτελείται από τα δείγματα που δεν επιλέχθηκαν κατά τη διαδικασία bootstrap, είναι δυνατή η εκτίμηση της απόδοσης του μοντέλου χωρίς τη χρήση ξεχωριστού συνόλου επικύρωσης. Το Out-Of-Bag Error (OOB Error) αντιπροσωπεύει το ποσοστό των OOB δειγμάτων που ταξινομήθηκαν λανθασμένα και χρησιμοποιείται για τη ρύθμιση των παραμέτρων του μοντέλου, όπως ο αριθμός των χαρακτηριστικών που χρησιμοποιούνται σε κάθε διαχωρισμό.

Η Εικόνα 4.4.1.2 παρουσιάζει τη δομή ενός Random Forest, αποτελούμενου από πολλαπλά Decision Trees, καθένα από τα οποία μαθαίνει από διαφορετικά υποσύνολα δεδομένων.



Εικόνα 4.4.1.2: Random Forest

Ένα από τα βασικά πλεονεκτήματα των Random Forests είναι η αυξημένη ακρίβεια και ανθεκτικότητα στο θόρυβο. Καθώς η τελική απόφαση βασίζεται στη συνδυασμένη "ψήφο" πολλών Decision Trees, η παρουσία θορυβωδών δεδομένων σε μεμονωμένα δέντρα έχει ελάχιστη επίδραση στο συνολικό αποτέλεσμα. Επιπλέον, το Random Forest επιτρέπει την αποτίμηση της σημασίας των χαρακτηριστικών (feature importance), υπολογίζοντας τη σχετική συνεισφορά κάθε χαρακτηριστικού στη συνολική απόδοση του μοντέλου.

Λόγω των δυνατοτήτων αυτών, η χρήση του Random Forest προτείνεται ως μέθοδος μοντελοποίησης, καθώς παρέχει αξιόπιστες προβλέψεις, μειώνει το πρόβλημα της υπερ προσαρμογής και επιτρέπει την ανάλυση της σημασίας των χαρακτηριστικών.

4.4.2 Αλγόριθμος K-Nearest Neighbors

Ο k-Nearest Neighbors (k-NN) είναι ένας από τους πιο θεμελιώδεις αλγορίθμους της μηχανικής μάθησης, ο οποίος εφαρμόζεται σε προβλήματα ταξινόμησης (classification) και παλινδρόμησης (regression). Βασίζεται στην αρχή ότι παρόμοια δεδομένα βρίσκονται κοντά μεταξύ τους σε έναν διανυσματικό χώρο χαρακτηριστικών.

- Στην ταξινόμηση (k-NN classification), η πρόβλεψη της κλάσης ενός νέου δεδομένου γίνεται βάσει της πλειοψηφίας των k κοντινότερων γειτόνων.
- Στην παλινδρόμηση (k-NN regression), η έξοδος υπολογίζεται ως ο μέσος όρος των τιμών των k κοντινότερων γειτόνων.

Η διαδικασία του αλγορίθμου περιλαμβάνει τα εξής βήματα:

1. Ορισμός της παραμέτρου κ, που καθορίζει τον αριθμό των γειτονικών σημείων που θα ληφθούν υπόψη,

2. Υπολογισμός της απόστασης μεταξύ του νέου δεδομένου και όλων των σημείων στο σύνολο εκπαίδευσης. Οι πιο κοινές μετρικές απόστασης είναι:

   Ευκλείδια απόσταση (Euclidean Distance)

   Η απλή καρτεσιανή απόσταση μεταξύ δύο σημείων.

$$d(x, X_i) = \sqrt{\sum_{j=1}^{d} (x_j - X_{i_j})^2}$$

   Απόσταση Manhattan (Manhattan Distance)

   Υπολογίζεται ως το άθροισμα των απόλυτων διαφορών των συντεταγμένων των δύο σημείων.

$$d(x, y) = \sum_{i=1}^{n} \left| x_i - y_i \right|$$

   Απόσταση Minkowski (Minkowski Distance)

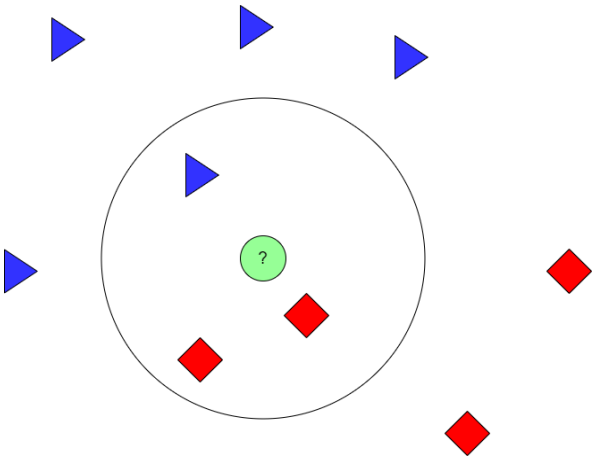
   Η Euclidean και η Manhattan είναι ειδικές περιπτώσεις της απόστασης Minkowski.

$$d(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^p \right)^{\frac{1}{p}}$$

   όπου όταν p = 2 τότε είναι η ίδια formula με την Euclidean και όταν p = 1 τότε παίρνουμε την εξίσωση για την απόσταση Manhattan.

3. Ταξινόμηση του νέου δεδομένου βάσει των ετικετών των k κοντινότερων γειτόνων. Στην ταξινόμηση, η ετικέτα που εμφανίζεται συχνότερα μεταξύ των γειτόνων καθορίζει την κατηγορία του νέου δεδομένου.

Η Εικόνα 4.4.2 παρουσιάζει ένα παράδειγμα όπου ένα άγνωστο σημείο κατηγοριοποιείται με βάση τους τρείς κοντινότερους γείτονές του. Στο παράδειγμα, υπάρχουν δύο κλάσεις: κόκκινοι ρόμβοι και μπλε τρίγωνα. Οι τρεις πλησιέστεροι γείτονες περιλαμβάνουν δύο ρόμβους και ένα τρίγωνο. Σύμφωνα με τον αλγόριθμο k-NN, το νέο σημείο ταξινομείται στην κλάση του ρόμβου, καθώς αποτελεί την πλειοψηφία των k γειτόνων.



Εικόνα 4.4.2: Κατηγοριοποίηση ενός αγνώστου δεδομένου εισόδου ελέγχοντας 3 κοντινότερους γείτονες

Ένα από τα βασικά πλεονεκτήματα του k-NN είναι ότι δεν απαιτεί υπόθεση για τη σχέση μεταξύ των μεταβλητών. Σε αντίθεση με άλλα μοντέλα που στηρίζονται σε μαθηματικές εξισώσεις, ο αλγόριθμος βασίζεται αποκλειστικά στη γειτνίαση των σημείων στον χώρο των χαρακτηριστικών, καθιστώντας τον ιδιαίτερα ευέλικτο. Για τον λόγο αυτό, προτείνεται η χρήση του ως μία από τις μεθόδους μοντελοποίησης, καθώς μπορεί να παρέχει αποδοτικά αποτελέσματα σε περιπτώσεις όπου η σχέση μεταξύ των δεδομένων δεν είναι αυστηρά γραμμική.

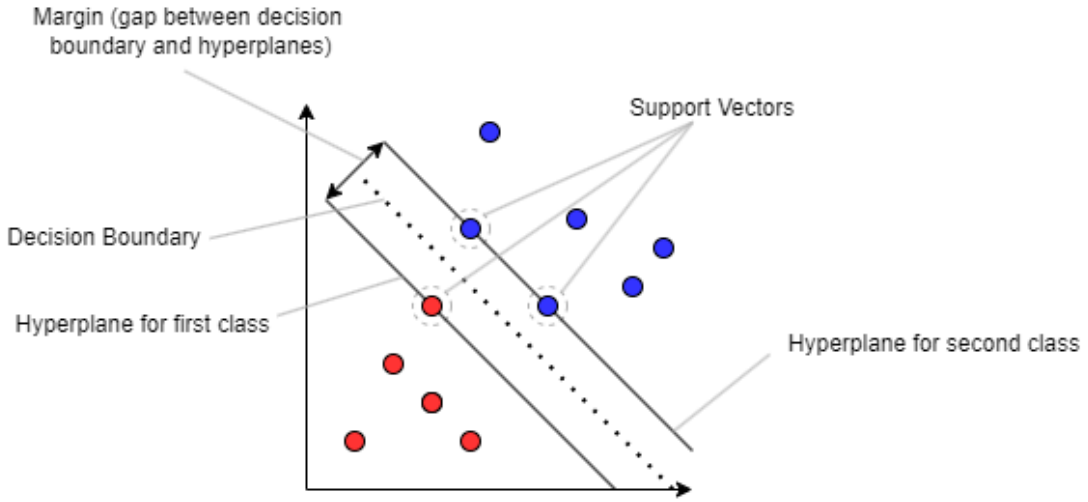
4.4.3 Support Vector Machines

Ο αλγόριθμος Support Vector Machines (SVM) είναι μία από τις πιο ισχυρές μεθόδους επιβλεπόμενης μηχανικής μάθησης (supervised learning), η οποία εφαρμόζεται σε προβλήματα ταξινόμησης (classification), παλινδρόμησης (regression) και εντοπισμού ακραίων τιμών (outlier detection). Παρόλο που είναι ευέλικτος, χρησιμοποιείται κυρίως για ταξινόμηση, καθώς είναι ικανός να διαχωρίζει τις κατηγορίες δεδομένων με τον βέλτιστο δυνατό τρόπο.

Το SVM επιδιώκει να βρει το υπερεπίπεδο (hyperplane) που διαχωρίζει με τον μεγαλύτερο δυνατό περιθώριο (margin) τις διαφορετικές κλάσεις δεδομένων. Σε έναν δισδιάστατο (2D) χώρο, το υπερεπίπεδο είναι μία ευθεία γραμμή, ενώ σε έναν τρισδιάστατο (3D) χώρο είναι ένα επίπεδο. Σε υψηλότερες διαστάσεις, αποτελεί ένα γενικευμένο υπερεπίπεδο που χωρίζει τα δεδομένα στον χώρο των χαρακτηριστικών.

Η Εικόνα 4.4.3.1 απεικονίζει ένα SVM σε δύο διαστάσεις (2D), όπου τα δεδομένα δύο κατηγοριών διαχωρίζονται με ένα υπερεπίπεδο (decision boundary). Τα support vectors

(υποστηρικτικά διανύσματα) είναι τα πλησιέστερα σημεία στο όριο της απόφασης, τα οποία καθορίζουν το περιθώριο μεταξύ των κατηγοριών.



Εικόνα 4.4.3.1: Support Vector Machine

Σε πολλές περιπτώσεις, τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα στον αρχικό τους χώρο. Για την αντιμετώπιση αυτού του προβλήματος, το SVM χρησιμοποιεί τη μέθοδο kernel trick, η οποία μετασχηματίζει τα δεδομένα σε έναν χώρο υψηλότερων διαστάσεων, όπου μπορεί να βρεθεί ένας γραμμικός διαχωρισμός.

Μερικά από τα πιο διμοφιλή kernel functions που χρησιμοποιούνται είναι:

- Linear Kernel: Χρησιμοποιείται όταν τα δεδομένα μπορούν να διαχωριστούν γραμμικά.
- Polynomial Kernel: Κατάλληλο για πιο πολύπλοκες σχέσεις μεταξύ των χαρακτηριστικών.
- Radial Basis Function (RBF) Kernel: Χρησιμοποιεί Gaussian συναρτήσεις για να ενισχύσει τον μη γραμμικό διαχωρισμό.
- Sigmoid Kernel: Χρησιμοποιείται σε εφαρμογές που προσομοιώνουν τη λειτουργία των νευρωνικών δικτύων.

Μία κρίσιμη υπερπαράμετρος του SVM είναι η παράμετρος C, η οποία ελέγχει το βαθμό αυστηρότητας του μοντέλου απέναντι στα σφάλματα κατάταξης:

- Μεγάλη τιμή C: Το μοντέλο προσπαθεί να ταξινομήσει όλα τα δεδομένα χωρίς λάθη, οδηγώντας σε μικρότερο περιθώριο και πιθανό υπερπροσαρμογή (overfitting).

- Μικρή τιμή C: Επιτρέπει περισσότερα σφάλματα ταξινόμησης, δημιουργώντας ένα μεγαλύτερο περιθώριο, το οποίο μπορεί να γενικεύει καλύτερα σε νέα δεδομένα.

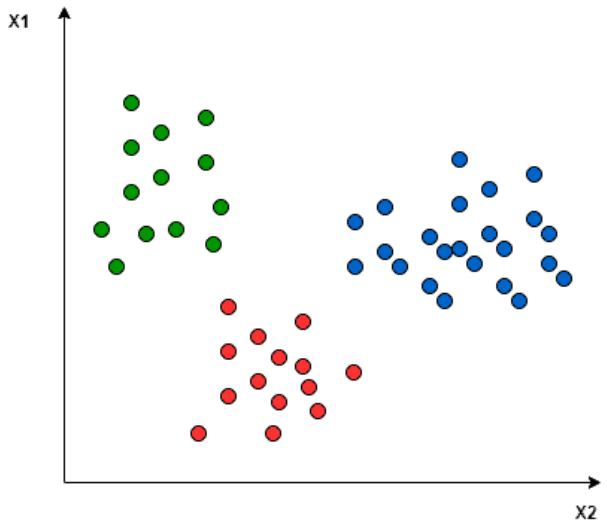
Το SVM είναι εγγενώς ένας δυαδικός ταξινομητής, καθώς διαχωρίζει τα δεδομένα σε δύο κατηγορίες. Ωστόσο, για προβλήματα ταξινόμησης με περισσότερες από δύο κλάσεις, εφαρμόζονται δύο στρατηγικές:

- One-vs-One (OvO):

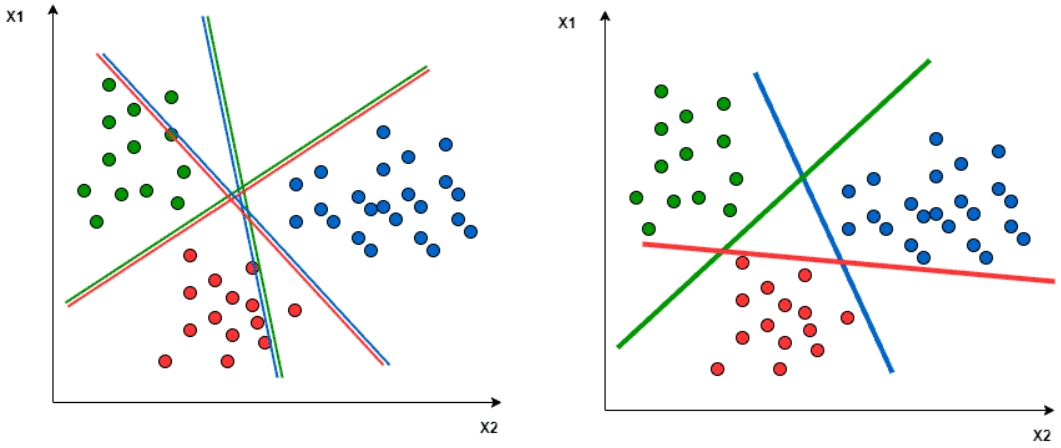
  Δημιουργείται ένας ταξινομητής SVM για κάθε ζεύγος κλάσεων. Αν υπάρχουν m κλάσεις, τότε δημιουργούνται $m(m-1)/2$ μοντέλα SVM. Η τελική ταξινόμηση γίνεται με βάση τη δημοφιλέστερη επιλογή ανάμεσα στα αποτελέσματα όλων των δυαδικών ταξινομητών. Η Εικόνα 4.4.3.2 απεικονίζει την εφαρμογή αυτής της προσέγγισης, όπου κατασκευάζονται πολλαπλά υπερεπίπεδα για τον διαχωρισμό των δεδομένων.
- One-vs-Rest (OvR):

  Δημιουργείται ένας ταξινομητής SVM για κάθε κλάση, όπου η κλάση συγκρίνεται με όλες τις υπόλοιπες μαζί. Αν υπάρχουν m κλάσεις, τότε δημιουργούνται m μοντέλα SVM. Η τελική κατηγορία καθορίζεται από τον ταξινομητή με την υψηλότερη εμπιστοσύνη πρόβλεψης.

Η Εικόνα 4.4.3.2 απεικονίζει την εφαρμογή των One-vs-One και One-vs-Rest προσεγγίσεων, όπου οι κλάσεις διαχωρίζονται με διαφορετικά υπερεπίπεδα.



a) Πρόβλημα ταξινόμησης 3 κλάσεων: κόκκινη, μπλε και πράσινη κλάση



b) Εφαρμογή της One-to-One προσέγγισης όπου δημιουργείται ένα υπερεπίπεδο μεταξύ κάθε ζεύγος κλάσης χωρίς να λαμβάνονται υπ όψιν οι άλλες κλάσεις

c) Εφαρμογή της One-to-Rest προσέγγισης όπου δημιουργείται ένα υπερεπίπεδο μεταξύ μίας κλάσης και όλων των υπολοίπων

Εικόνα 4.4.3.2: Support Vector Machine με 3 κλάσεις

Το SVM είναι ιδιαίτερα αποδοτικό σε περιπτώσεις με μικρά ή μεσαίου μεγέθους σύνολα δεδομένων, καθώς μπορεί να προσδιορίσει τα όρια μεταξύ των κατηγοριών με ακρίβεια. Ωστόσο, για μεγάλα σύνολα δεδομένων, το κόστος υπολογισμών μπορεί να είναι αυξημένο.

Λόγω των δυνατοτήτων του στην εύρεση των βέλτιστων διαχωριστικών ορίων, το SVM προτείνεται ως μία από τις μεθόδους ταξινόμησης για το υπό μελέτη πρόβλημα.

4.4.4 Αλγόριθμος Logistic Regression για Προβήματα Ταξινόμησης

Το Logistic Regression είναι ένα στατιστικό μοντέλο που χρησιμοποιείται κυρίως για προβλήματα ταξινόμησης (classification), όπου στόχος είναι ο υπολογισμός της πιθανότητας ένα δείγμα να ανήκει σε μία κλάση ή όχι. Σε αντίθεση με τη γραμμική παλινδρόμηση (Linear Regression), το Logistic Regression εξάγει πιθανότητες και χρησιμοποιείται κυρίως για δυαδική ταξινόμηση (binary classification).

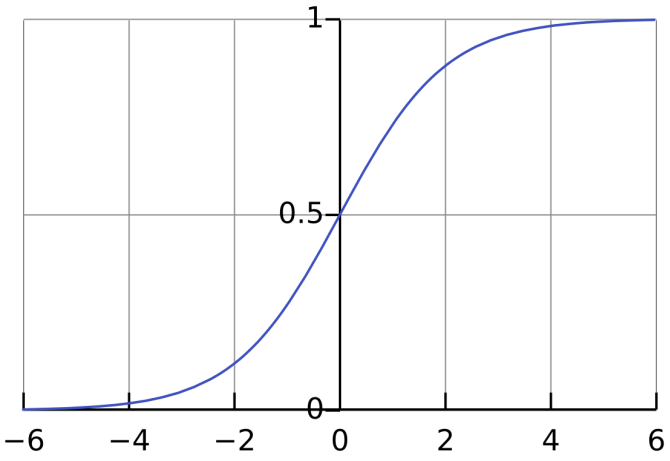
Το Logistic Regression βασίζεται στη σιγμοειδή συνάρτηση (sigmoid function), η οποία μετασχηματίζει οποιαδήποτε πραγματική τιμή σε πιθανότητα εντός του εύρους [0,1]. Η μαθηματική της εξίσωση είναι:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

όπου:

- z είναι ο γραμμικός συνδυασμός των χαρακτηριστικών εισόδου.

- Η έξοδος της σιγμοειδούς συνάρτησης παίρνει τιμές μεταξύ 0 και 1, σχηματίζοντας μια S-μορφή καμπύλη.
- Η τιμή κατωφλιού (threshold) συνήθως ορίζεται στο 0.5. Εάν η πιθανότητα που υπολογίζεται από το μοντέλο είναι μεγαλύτερη από 0.5, το δείγμα ταξινομείται στην κατηγορία 1, διαφορετικά στην κατηγορία 0. Η Εικόνα 4.4.4 απεικονίζει την μορφή της σιγμοειδής συνάρτησης.



Εικόνα 4.4.4: Logistic Regression

Το Logistic Regression σχεδιάστηκε αρχικά για δυαδική ταξινόμηση, όπου η εξαρτημένη μεταβλητή έχει δύο κατηγορίες. Ωστόσο, μπορεί να επεκταθεί και σε πολυταξική ταξινόμηση (multiclass classification) χρησιμοποιώντας την τεχνική One-vs-Rest (OvR):

- Εκπαιδεύεται ένα ξεχωριστό δυαδικό Logistic Regression μοντέλο για κάθε κατηγορία.
- Κάθε μοντέλο προβλέπει την πιθανότητα το δείγμα να ανήκει στη συγκεκριμένη κλάση έναντι όλων των άλλων.
- Η τελική ταξινόμηση γίνεται επιλέγοντας την κλάση με τη μεγαλύτερη πιθανότητα.

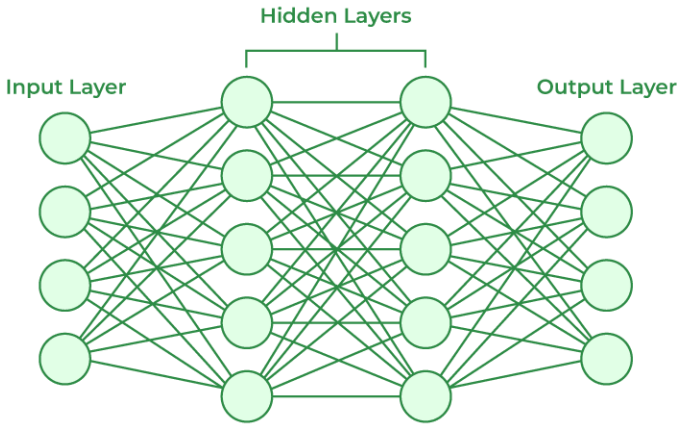
Λόγω αυτής της ευελιξίας, το Logistic Regression προτείνεται ως μία από τις μεθόδους ταξινόμησης στο προτεινόμενο σύστημα. Παρέχει απλότητα, ερμηνευσιμότητα και είναι αποτελεσματικό για προβλήματα με καλά διαχωρίσιμες κλάσεις.

4.4.5 Neural Networks

Τα νευρωνικά δίκτυα (Neural Networks - NN) είναι μοντέλα της μηχανικής μάθησης που εμπνέονται από τη βιολογική δομή και λειτουργία των νευρώνων του ανθρώπινου εγκεφάλου.

Αποτελούνται από διασυνδεδεμένους τεχνητούς νευρώνες, οι οποίοι επεξεργάζονται δεδομένα μέσω αριθμητικών υπολογισμών και παράγουν εξόδους που μπορούν να χρησιμοποιηθούν για ταξινόμηση (classification), παλινδρόμηση (regression) και άλλες εφαρμογές.

Η Εικόνα 4.4.5 απεικονίζει την αρχιτεκτονική ενός τυπικού νευρωνικού δικτύου, παρουσιάζοντας τη ροή πληροφοριών από τις εισόδους προς τις εξόδους μέσω ενδιάμεσων επιπέδων επεξεργασίας.



Εικόνα 4.4.5: Αρχιτεκτονική ενός τυπικού Νευρωνικού Δικτύου

Ένα νευρωνικό δίκτυο αποτελείται από τρεις κύριους τύπους νευρώνων (neurons):

- Νευρώνες εισόδου (Input neurons): Μεταφέρουν τα δεδομένα από το περιβάλλον προς το δίκτυο χωρίς να πραγματοποιούν υπολογισμούς.
- Υπολογιστικοί νευρώνες ή κρυμμένοι νευρώνες (Hidden neurons): Πραγματοποιούν υπολογισμούς, ανιχνεύουν μοτίβα και συσχετίσεις και διαμορφώνουν τα αποτελέσματα της ανάλυσης.
- Νευρώνες εξόδου (Output neurons): Παράγουν την τελική απόκριση του δικτύου, η οποία μπορεί να είναι μια κατηγορία ταξινόμησης, μια αριθμητική πρόβλεψη κ.λπ.

Η μαθηματική διατύπωση της εξόδου ενός νευρώνα δίνεται από την εξίσωση:

$$y_k = \varphi(\sum_{i=0}^{N} x_{ki} w_{ki})$$

όπου:

- x_{ki} είναι η είσοδος στον kkk-οστό νευρώνα,

- w_{ki} είναι το συναπτικό βάρος,
- $\varphi(\cdot)$ είναι η συνάρτηση ενεργοποίησης, η οποία καθορίζει την έξοδο του νευρώνα.

Ρόλος των Νευρωνικών Δικτύων στην Οπτικοποίηση Δεδομένων:

- Αυτόματη Αναγνώριση Κατάλληλης Οπτικοποίησης: Τα νευρωνικά δίκτυα μπορούν να μάθουν ποια χαρακτηριστικά των δεδομένων καθορίζουν τη βέλτιστη επιλογή γραφικής αναπαράστασης. Μέσω της ανάλυσης μεγάλου όγκου δεδομένων, εντοπίζουν συσχετίσεις ανάμεσα σε ιδιότητες των δεδομένων (π.χ. αριθμητικά ή κατηγορηματικά χαρακτηριστικά, συσχετίσεις μεταξύ στηλών) και τις οπτικοποιήσεις που τις αναδεικνύουν καλύτερα.
- Βελτιωμένη Κατηγοριοποίηση Δεδομένων: Χρησιμοποιώντας χαρακτηριστικά των δεδομένων, ένα νευρωνικό δίκτυο μπορεί να ταξινομεί αυτόματα τις δομές των δεδομένων σε κατάλληλες κατηγορίες (π.χ. καταλληλότητα για ραβδόγραμμα, διασπορά, γραμμικό διάγραμμα κ.λπ.).
- Ανίχνευση Μοτίβων σε Δεδομένα: Δεδομένου ότι πολλές γραφικές παραστάσεις έχουν διαφορετικές χρήσεις ανάλογα με το μοτίβο των δεδομένων, τα νευρωνικά δίκτυα μπορούν να εντοπίζουν περιοδικότητες, τάσεις ή κατανομές στα δεδομένα και να προτείνουν τις πιο κατάλληλες οπτικοποιήσεις.
- Αντιμετώπιση Περίπλοκων Δεδομένων: Όταν οι σχέσεις μεταξύ των μεταβλητών δεν είναι προφανείς ή γραμμικές, ένα νευρωνικό δίκτυο μπορεί να αναγνωρίσει κρυμμένα μοτίβα και να προσαρμόσει δυναμικά τις προτάσεις του. Αυτό το καθιστά ιδιαίτερα χρήσιμο σε περιπτώσεις όπου οι παραδοσιακές μεθοδολογίες αποτυγχάνουν να εντοπίσουν ποια οπτικοποίηση είναι η βέλτιστη.
- Προσαρμοστικότητα σε Νέα Δεδομένα: Σε αντίθεση με πιο στατικές μεθόδους, τα νευρωνικά δίκτυα βελτιώνονται με την πάροδο του χρόνου, καθώς εκπαιδεύονται σε νέο σύνολο δεδομένων, αναπροσαρμόζοντας την επιλογή της κατάλληλης οπτικοποίησης με βάση τα χαρακτηριστικά των εκάστοτε δεδομένων.

Για αυτόν τον λόγο, τα νευρωνικά δίκτυα προτείνονται ως μία από τις μεθόδους μοντελοποίησης του προβλήματος, καθώς επιτρέπουν προηγμένη μάθηση από τα χαρακτηριστικά των δεδομένων και εξαγωγή ακριβών προβλέψεων.

4.4.6 AutoML (Automated Machine Learning)

Οι μέθοδοι Βαθιάς Μάθησης (Deep Learning - DL) έχουν σημειώσει σημαντικές προόδους σε διάφορους τομείς, όπως η αναγνώριση εικόνας (image recognition), η ανίχνευση αντικειμένων (object detection) και η μοντελοποίηση φυσικής γλώσσας (language modeling).

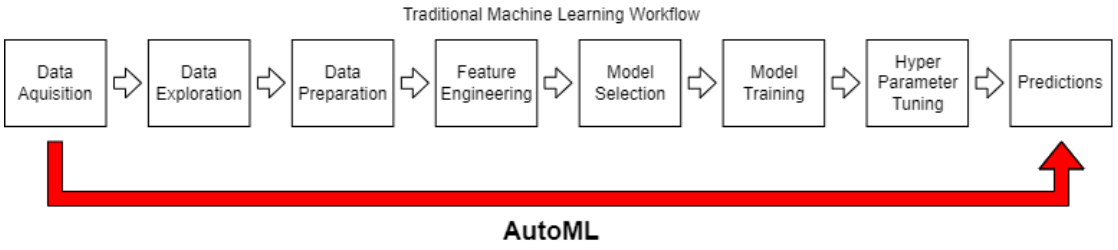
Ωστόσο, η ανάπτυξη ενός συστήματος μηχανικής μάθησης υψηλής ποιότητας απαιτεί εξειδικευμένη γνώση και εμπειρία, καθιστώντας τη διαδικασία χρονοβόρα και περιορίζοντας την ευρεία εφαρμογή της.

Η παραδοσιακή προσέγγιση για την ανάπτυξη ενός μοντέλου Machine Learning (ML) περιλαμβάνει τα εξής διαδοχικά στάδια:

- Συλλογή και εξερεύνηση δεδομένων (Data Acquisition & Exploration)
- Προετοιμασία και επεξεργασία δεδομένων (Data Preparation & Feature Engineering)
- Επιλογή και εκπαίδευση μοντέλου (Model Selection & Training)
- Βελτιστοποίηση υπερπαραμέτρων (Hyperparameter Tuning)
- Δοκιμή και παραγωγή προβλέψεων (Predictions)

Αυτή η διαδικασία απαιτεί πολλαπλές παρεμβάσεις από ειδικούς, καθιστώντας την δύσκολη στην εφαρμογή από μη τεχνικούς χρήστες.
Μια πολλά υποσχόμενη λύση στο παραπάνω πρόβλημα είναι το Automated Machine Learning (AutoML), το οποίο αποσκοπεί στη βελτιστοποίηση και αυτοματοποίηση των διαδικασιών που απαιτούνται για τη δημιουργία ενός μοντέλου ML, χωρίς ανθρώπινη παρέμβαση. Οι AutoML βιβλιοθήκες μπορούν να μειώσουν δραματικά τον απαιτούμενο χρόνο και πόρους, ενώ επιτρέπουν σε αναλυτές χωρίς εξειδικευμένες τεχνικές γνώσεις να αξιοποιήσουν προηγμένες τεχνικές μηχανικής μάθησης.

Η Εικόνα 4.4.6 απεικονίζει τη διαφορά μεταξύ της παραδοσιακής ροής εργασιών στη μηχανική μάθηση και του AutoML. Παρατηρούμε ότι στη συμβατική προσέγγιση, κάθε στάδιο απαιτεί ξεχωριστή ανθρώπινη παρέμβαση, ενώ με τη χρήση AutoML, πολλά από τα ενδιάμεσα στάδια ενοποιούνται και αυτοματοποιούνται, επιταχύνοντας τη διαδικασία ανάπτυξης και βελτιστοποίησης του μοντέλου.



Εικόνα 4.4.6: Στάδια παραδοσιακής διαδικασίας ανάπτυξης machine learning μοντέλου και η συντόμευση AutoM.

Τα εργαλεία που κάνουν την AutoML μια καινοτόμα και ενδιαφέρουσα προσέγγιση ώστε να τη δοκιμάσουμε στον τομέα της **Αυτόματης Οπτικοποίησης Αναλυτικής Δεδομένων** είναι:

1. Αυτοματοποιημένη Επεξεργασία Δεδομένων (Data Preparation & Feature Engineering): Οι AutoML βιβλιοθήκες μπορούν να εξερευνήσουν και να προεπεξεργαστούν τα δεδομένα αυτόματα, περιλαμβάνοντας:
    a. Διαχείριση ελλειπόντων τιμών
    b. Κανονικοποίηση χαρακτηριστικών (Feature Scaling)
    c. Δημιουργία νέων χαρακτηριστικών από υπάρχοντα δεδομένα
    d. Κωδικοποίηση κατηγορικών μεταβλητών σε αριθμητικές τιμές

    Ορισμένες βιβλιοθήκες μπορούν ακόμη και να ανιχνεύσουν το είδος του προβλήματος (classification ή regression) χωρίς παρέμβαση από τον χρήστη.
2. Βελτιστοποίηση Υπερπαραμέτρων (Hyperparameter Optimization - HPO): Η βελτίωση της απόδοσης ενός μοντέλου εξαρτάται από την επιλογή των κατάλληλων υπερπαραμέτρων. Το AutoML χρησιμοποιεί τεχνικές όπως:
    a. Grid Search και Random Search
    b. Bayesian Optimization
    c. Γενετικούς Αλγόριθμους (Genetic Algorithms)

    Μέσω αυτών, το AutoML αυτοματοποιεί τη διαδικασία εύρεσης των βέλτιστων υπερπαραμέτρων, ελαχιστοποιώντας την ανάγκη για δοκιμές από τον χρήστη.
3. Συνδυασμένη Επιλογή Αλγορίθμου & Βελτιστοποίηση Υπερπαραμέτρων (CASH - Combined Algorithm Selection & Hyperparameter Optimization): Το CASH αποτελεί την καρδιά της σύγχρονης AutoML προσέγγισης, καθώς δεν περιορίζεται στη βελτιστοποίηση των υπερπαραμέτρων αλλά:
    a. Επιλέγει αυτόματα τον καταλληλότερο αλγόριθμο για το εκάστοτε πρόβλημα (π.χ., Decision Trees, Random Forest, SVM).
    b. Ρυθμίζει τις υπερπαραμέτρους για κάθε αλγόριθμο, προκειμένου να μεγιστοποιηθεί η απόδοσή του.

    Αυτό εξαλείφει την ανάγκη για χειροκίνητη επιλογή μοντέλου, επιτρέποντας την αυτόματη αναζήτηση της βέλτιστης λύσης.
4. Αυτόματη Αναζήτηση Βέλτιστης Αρχιτεκτονικής Νευρωνικών Δικτύων (Neural Architecture Search – NAS): Για βαθιά μάθηση, το Neural Architecture Search (NAS) είναι μία τεχνική που εξερευνά διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων, βρίσκοντας την ιδανική για το εκάστοτε πρόβλημα. Περιλαμβάνει:
    a. Αναζήτηση στο χώρο των πιθανών αρχιτεκτονικών
    b. Στρατηγική εξερεύνησης για την επιλογή της βέλτιστης λύσης
    c. Αξιολόγηση απόδοσης κάθε πιθανού δικτύου

Η NAS τεχνολογία μειώνει δραματικά την ανάγκη για χειροκίνητο σχεδιασμό πολύπλοκων νευρωνικών δικτύων, αυτοματοποιώντας τον καθορισμό της δομής, του μεγέθους και των συνδέσεων του δικτύου.

5. Στοίβαγμα Μοντέλων (Multi-Layer Stack Ensembling): Η τεχνική αυτή χρησιμοποιείται για τη βελτίωση της ακρίβειας των προβλέψεων συνδυάζοντας πολλαπλά μοντέλα σε διαφορετικά "επίπεδα" (layers). Μια ομάδα από "base" μοντέλα εκπαιδεύονται μεμονωμένα με τις κλασικές μεθόδους. Έπειτα οι προβλέψεις τροφοδοτούνται μαζικά ως features μαζί με τα original δεδομένα σε νέα μοντέλα (stacker models) στο δεύτερο επίπεδο τα οποία με τη σειρά τους προσπαθούν να μάθουν απο τα σφάλματα των προηγούμενων μοντέλων. Στον τελικό stacking layer συγχωνεύονται οι προβλέψεις από τα προηγούμενα μοντέλα με την χρήση βαρών για κάθε μοντέλο.

Δεδομένης της ικανότητάς του να βελτιστοποιεί αυτόματα τη διαδικασία μοντελοποίησης, το AutoML αποτελεί ιδανική προσέγγιση για την Αυτόματη Οπτικοποίηση Αναλυτικής Δεδομένων. Όπως συζητήθηκε στην Ενότητα 4.3, η προεπεξεργασία των χαρακτηριστικών (feature preprocessing) αποτελεί ένα σημαντικό βήμα για τη βελτίωση της απόδοσης των μοντέλων. Η AutoML προσέγγιση υπόσχεται να αυτοματοποιήσει αυτό το βήμα, εξαλείφοντας πιθανά ανθρώπινα λάθη και εξασφαλίζοντας μια αποδοτικότερη και βελτιστοποιημένη διαδικασία.

Η χρήση του AutoML στην προτεινόμενη μεθοδολογία μας επιτρέπει να συγκρίνουμε τις αυτόματες επιλογές του συστήματος με τις παραδοσιακές τεχνικές μηχανικής μάθησης, αναλύοντας την απόδοσή τους στην επιλογή της ιδανικής οπτικοποίησης δεδομένων.

4.5 Αξιολόγηση Μοντέλων - Model Evaluation

Η επιλογή της σωστής μετρικής είναι ζωτικής σημασίας κατά την αξιολόγηση των μοντέλων μηχανικής μάθησης. Ωστόσο, η πραγματική ισχύς ενός μοντέλου συνειδητοποιείται μόνο όταν έχουμε τη δυνατότητα να μετρήσουμε ποσοτικά την απόδοσή του. Οι μετρικές αξιολόγησης παίζουν κρίσιμο ρόλο στη διαδικασία αυτή, καθώς μας επιτρέπουν να κρίνουμε την ποιότητα και την αξιοπιστία των προβλέψεων του μοντέλου.

Δεν υπάρχει μία ενιαία μετρική που να ταιριάζει σε κάθε πρόβλημα, αφού η κατάλληλη επιλογή εξαρτάται από το πλαίσιο του προβλήματος και το είδος των δεδομένων. Στην περίπτωση της Αυτόματης Οπτικοποίησης Αναλυτικών Δεδομένων, το πρόβλημα είναι ένα πολυκατηγορικό classification πρόβλημα, όπου στόχος είναι η ακριβής κατηγοριοποίηση των δεδομένων σε συγκεκριμένους τύπους γραφημάτων.

Για την αξιολόγηση ενός ταξινομητή, χρησιμοποιούμε τον πίνακα σύγχυσης (confusion matrix), ο οποίος αποτελείται από τέσσερις βασικές κατηγορίες προβλέψεων:

1. True Positive (TP): Δείγματα με θετικές ετικέτες όπου το μοντέλο τις προβλέπει σωστά ως θετικές.
2. True Negative (TN): Δείγματα με αρνητικές ετικέτες όπου το μοντέλο τις προβλέπει σωστά ως θετικές.
3. False Positive (FP): Δείγματα με αρνητικές ετικέτες όπου το μοντέλο τις προβλέπει λανθασμένα ως θετικές.
4. False Negative (FN): Δείγματα με θετικές ετικέτες όπου το μοντέλο τις προβλέπει λανθασμένα ως αρνητικές.

Οι πιο διαδεδομένες μετρικές αξιολόγησης στην κατηγοριοποίηση είναι οι Accuracy, Precision, Recall και F1 Score.

Accuracy

Η ακρίβεια (accuracy) μετρά το ποσοστό των σωστών προβλέψεων σε σχέση με το σύνολο των προβλέψεων. Είναι η πιο βασική μετρική και χρησιμοποιείται ιδιαίτερα σε ισορροπημένα σύνολα δεδομένων.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Πλεονεκτήματα: Εύκολη στην κατανόηση, χρήσιμη όταν έχουμε ισορροπημένο dataset.

Μειονεκτήματα: Δεν είναι αξιόπιστη όταν υπάρχει ανισορροπία στις κατηγορίες (class imbalance).

Precision

Η precision δείχνει πόσο ακριβείς είναι οι θετικές προβλέψεις του μοντέλου. Δηλαδή, από τις περιπτώσεις που ταξινομήθηκαν ως θετικές, πόσες είναι πραγματικά σωστές.

$$Accuracy = \frac{TP}{TP + FP}$$

Πλεονεκτήματα: Χρήσιμη σε περιπτώσεις όπου τα False Positives έχουν υψηλό κόστος (π.χ., στην ανίχνευση ανεπιθύμητης αλληλογραφίας).

Μειονεκτήματα: Δεν λαμβάνει υπόψη τα False Negatives.

Recall

Η recall δείχνει πόσες από τις πραγματικά θετικές περιπτώσεις ανιχνεύτηκαν σωστά από το μοντέλο.

$$Recall = \frac{TP}{TP + FN}$$

Πλεονεκτήματα: Χρήσιμη όταν τα False Negatives έχουν μεγάλο κόστος (π.χ., σε ιατρικές διαγνώσεις).
Μειονεκτήματα: Δεν λαμβάνει υπόψη τα False Positives, με αποτέλεσμα να μπορεί να υπερεκτιμήσει τις προβλέψεις.

F1 Score

Η F1 Score είναι ένας συνδυασμός Precision και Recall και υπολογίζεται ως ο αρμονικός μέσος των δύο:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Πλεονεκτήματα: Χρήσιμη όταν θέλουμε ισορροπία μεταξύ Precision και Recall.
Μειονεκτήματα: Δεν λαμβάνει υπόψη τα True Negatives, κάτι που μπορεί να επηρεάσει την αξιολόγηση σε ορισμένα προβλήματα.

Η επιλογή της κατάλληλης μετρικής αξιολόγησης είναι κρίσιμη για την επιτυχημένη εκπαίδευση και αξιολόγηση ενός συστήματος μηχανικής μάθησης. Διαφορετικές μετρικές παρέχουν διαφορετικές οπτικές γωνίες της απόδοσης ενός ταξινομητή, και η καταλληλότητα κάθε μετρικής εξαρτάται από το εκάστοτε πρόβλημα και τη φύση των δεδομένων. Για την αξιολόγηση ενός μοντέλου στην αυτόματη οπτικοποίηση δεδομένων, η καταλληλότερη μετρική εξαρτάται από τη δομή του συνόλου δεδομένων. Αν οι κλάσεις είναι ισορροπημένες, η ακρίβεια (Accuracy) μπορεί να χρησιμοποιηθεί ως βασική μετρική. Σε περιπτώσεις ανισορροπίας, το F1 Score είναι πιο αξιόπιστο, καθώς λαμβάνει υπόψη τόσο την ακρίβεια όσο και την ανάκληση. Αν το ζητούμενο είναι η αποφυγή λανθασμένων προτάσεων οπτικοποίησης, τότε η Precision μπορεί να είναι η προτιμότερη επιλογή, ενώ αν προτεραιότητα είναι η εύρεση όλων των πιθανών καλών οπτικοποιήσεων, η Recall μπορεί να είναι πιο χρήσιμη. Η τελική επιλογή της μετρικής θα πρέπει να καθορίζεται από τις απαιτήσεις του εκάστοτε συστήματος και τους στόχους της αυτόματης οπτικοποίησης.

5 *Υλοποίηση*

5.1 *Εργαλεία και Βιβλιοθήκες*

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την κατασκευή του αλγορίθμου μας είναι η **python** και η πλατφόρμα στην οποία έγινε ο προγραμματισμός είναι το **Visual Studio Code** και το **Google Colab**. Το VSCode είναι ένας ανοικτού κώδικα επεξεργαστής κώδικα που έχει σχεδιαστεί για να είναι εύκολο στην χρήση και να παρέχει ένα ισχυρό περιβάλλον για την ανάπτυξη κώδικα σε διάφορες γλώσσες προγραμματισμού. Το Google Colab είναι ένα διαδικτυακό περιβάλλον προγραμματισμού που βασίζεται στο Jupyter Notebooks. Επιτρέπει στους χρήστες να γράφουν και να εκτελούν κώδικα Python απευθείας στο πρόγραμμα περιήγησης τους χωρίς να χρειάζεται να εγκαταστήσουν λογισμικό στον υπολογιστή τους. Η Python αποτελεί μια ερμηνευόμενη, αντικειμενοστραφής, γλώσσα προγραμματισμού υψηλού επιπέδου με δυναμική σημασιολογία. Η απλή, εύκολη στη μάθηση σύνταξη της Python δίνει έμφαση στην αναγνωσιμότητα, μειώνοντας έτσι το κόστος συντήρησης του προγράμματος. Η Python υποστηρίζει modules και packages, γεγονός που ενθαρρύνει τη δημιουργία δομοστοιχειωμένων (modular) προγραμμάτων και την επαναχρησιμοποίηση κώδικα. Φυσικά ενδείκνυται για την επίλυση προβλημάτων που κατηγοριοποιούνται ομοίως με το δικό μας πρόβλημα.

Τα δεδομένα μας τα πήραμε από το web και ήταν αρχικά σε **JSON (JavaScript Object Notation)** files. Είναι μια μορφή βασισμένη σε κείμενο για την αποθήκευση και ανταλλαγή δομημένων δεδομένων. Χρησιμοποιείται συνήθως για την διάδοση δεδομένων σε web εφαρμογές μεταξύ πελάτη και διακομιστή, και είναι ανεξάρτητο από γλώσσα προγραμματισμού, πράγμα που σημαίνει ότι μπορεί να χρησιμοποιηθεί με πολλές διαφορετικές γλώσσες, όπως Python.

Για την καλύτερη διαχείριση των δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη **Pandas** της Python. Παρέχει ισχυρά εργαλεία για ανάλυση, καθαρισμό, εξερεύνηση και διαχείριση

δομημένων δεδομένων. Είναι δημοφιλής στην επιστήμη δεδομένων, καθώς έχει πολλές συναρτήσεις που είναι εύκολες στη χρήση.

Για περαιτέρω στατιστική ανάλυση χρησιμοποιήθηκε το υπο-πακέτο **spicy.stats** της βιβλιοθήκης **SpiPy**, το οποίο περιέχει μια συλλογή από χρήσιμες συναρτήσεις που αξιοποιήσαμε για να υπολογίσουμε στατιστικές τιμές όπως entropy, kurtosis, skewness και pearson correlation coefficient.

Μέσο του **Scikit-learn (sklearn)** αξιοποιήσαμε τους αλγόριθμους ταξινόμησης random forest, support vector machines, logistic regression και k-nearest neighbors. Είναι μια δημοφιλής βιβλιοθήκη στην Python για μηχανική μάθηση και ανάλυση δεδομένων. Παρέχει πολλά εργασία προεπεξεργασίας που μας βοήθησαν μεταξύ των οποίων ήταν το StandardScaler για την κανονικοποίηση των δεδομένων, το OneHotEncoder για κωδικοποίηση κατηγορηματικών χαρακτηριστικών σε numerical και train_test_split για να μοιράσουμε το σύνολο δεδομένων σε test και train σύνολα. Μας παρείχε επίσης εργαλεία για αξιολόγηση της απόδοσης των μοντέλων όπως cross-validation, accuracy_score και confusion matrix.

Τα πλαίσια (frameworks) είναι εργαλεία λογισμικού που παρέχουν μια δομή και προκαθορισμένο κώδικα για την ανάπτυξη εφαρμογών. Αποτελούνται από πακέτα και βιβλιοθήκες που βοηθούν στην απλοποίηση της συνολικής εμπειρίας προγραμματισμού για τους προγραμματιστές ώστε να χτίσουν εφαρμογές γρήγορα και αποτελεσματικά, χωρίς να χρειάζεται να ξεκινήσουν από το μηδέν.

Το **Tensorflow** είναι μια ανοιχτού κώδικα πλατφόρμα για μηχανική μάθηση. Μπορεί να χρησιμοποιηθεί σε μια μεγάλη γκάμα εργασιών αλλά εστιάζει στην εκπαίδευση και την εξαγωγή συμπερασμάτων των βαθιών νευρωνικών δικτύων. Εξυπηρετεί ως πλατφόρμα για έρευνα και για ανάπτυξη συστημάτων μηχανικής μάθησης σε πολλούς τομείς όπως speech recognition, computer vision, robotics, information retrieval, and natural language processing. Είναι εύκολη στην χρήση αφού προσφέρει πολλαπλά εργαλεία για την κατασκευή και εκπαίδευση μοντέλων και στιβαρή παραγωγή ML οπουδήποτε, αφού επιτρέπει την εκπαίδευση και εύκολη ανάπτυξη του μοντέλου, ανεξάρτητα από τη γλώσσα ή την πλατφόρμα που χρησιμοποιεί ο χρήστης.

Το **Keras** είναι μια βιβλιοθήκη που παρέχει εξαιρετικά ισχυρά και αφηρημένα δομικά στοιχεία για τη δημιουργία δικτύων βαθιάς μάθησης. Αυτά τα δομικά στοιχεία που προσφέρει το Keras είναι χτισμένα χρησιμοποιώντας Theano και το TensorFlow. Το Keras υποστηρίζει και τα δύο, CPU και GPU υπολογισμούς και είναι εξαιρετικό εργαλείο για την γρήγορη πρωτοτυποποίηση ιδεών. Μερικά από τα κύρια χαρακτηριστικά τα οποία το κάνουν να ξεχωρίζει είναι ότι είναι απλό στην χρήση, και εύκολο στη εκμάθηση αφού έχει ευανάγνωστη

και απλή διεπαφή, βελτιστοποιημένη για κοινές περιπτώσεις χρήσης, η οποία παρέχει σαφή και ενεργή ανατροφοδότηση για σφάλματα χρήστη.

Το **AutoKeras** είναι μία βιβλιοθήκη λογισμικού ανοιχτού κώδικα για αυτοματοποιημένη μηχανική μάθηση (**AutoML**) χτισμένο πάνω από το Keras και το Tensorflow. Το AutoKeras απλοποιεί την διαδικασία σχεδίασης και εκπαίδευσης βαθιών νευρωνικών δικτύων κάνοντας το προσιτό σε όλα τα επίπεδα χρηστών εξοικονομώντας ταυτόχρονα σημαντικό υπολογιστικό χρόνο. Το **KerasTuner**, είναι μια βιβλιοθήκη ρύθμισης υπερπαραμέτρων για το Keras που παρέχει την υποδομή για την υλοποίηση του χώρου αναζήτησης και του αλγορίθμου αναζήτησης. Το κυριότερο χαρακτηριστικό του AutoKeras είναι ο **Neural Architecture Search (NAS)** αλγόριθμος του. Πραγματοποιεί μια αποδοτική αναζήτηση στο χώρο της νευρωνικής αρχιτεκτονικής και βρίσκει την ιδανικότερη αρχιτεκτονική για ένα δοσμένο dataset και learning task. Η βασική ροή εργασιών του AutoKeras περιλαμβάνει τα εξής βήματα. Πρώτον, το AutoKeras αναλύει τα δεδομένα εκπαίδευσης για να προσδιορίσει αν, για παράδειγμα, μια δεδομένη χαρακτηριστική παράμετρος πίνακα είναι κατηγορική ή αριθμητική, αν τα δεδομένα εικόνας περιλαμβάνουν διάσταση καναλιού ή αν οι ετικέτες ταξινόμησης χρειάζονται κωδικοποίηση. Δεύτερον, χρησιμοποιεί αυτές τις πληροφορίες για να κατασκευάσει έναν κατάλληλο χώρο αναζήτησης που περιλαμβάνει τόσο πρότυπα νευρωνικής αρχιτεκτονικής όσο και κοινές υπερπαραμέτρους. Τέλος, ο αλγόριθμος αναζήτησης βρίσκει τις υψηλών επιδόσεων υπερπαραμέτρους. Ο χώρος αναζήτησης του AutoKeras περιλαμβάνει μοντέλα βαθιάς μάθησης τελευταίας τεχνολογίας για τις υποστηριζόμενες εργασίες.

Το **AutoGluon** είναι ακόμα μια βιβλιοθήκη για την αυτοματοποιημένη μηχανική μάθηση (**AutoML**). Είναι χτισμένο πάνω από γνωστές machine learning βιβλιοθήκες όπως XGBoost, LightGBM, CatBoost, και deep learning frameworks όπως MXNet and PyTorch, έχει σχεδιαστεί για να είναι γενικό πλαίσιο AutoML. Είναι ένα πολύ ισχυρό εργαλείο το οποίο απαλλάσει τους χρήστες από πολλά βήματα της παραδοσιακής ροής εργασιών στην μηχανική μάθηση.

Για την ανάπτυξη της Web εφαρμογής μας χρησιμοποιούμε το **Streamlit**. Το Streamlit είναι ένα πλαίσιο ανοικτού κώδικα για την γρήγορη δημιουργία και ανάπτυξη εφαρμογών web για ανάλυση δεδομένων, μηχανική μάθηση και επιστήμη των δεδομένων. Σχεδιάστηκε για να διευκολύνει τους μηχανικούς δεδομένων, τους επιστήμονες δεδομένων και τους ερευνητές στη δημιουργία απλών αλλά ισχυρών εφαρμογών web, χωρίς να απαιτούνται γνώσεις σε frontend development καθώς δεν απαιτεί τον προγραμματισμό HTML, CSS ή JavaScript. Ολόκληρη η σύνταξη είναι στην Python, επιτρέποντας τη χρήση κοινων βιβλιοθηκών όπως Numpy, Pandas και Matplotlib. Δημιουργεί αυτόματα το User Interface (UI) ελαχιστοποιώντας την ανάγκη για παραδοσιακό frontend development. Παρέχει επίσης

διαδραστικότητα αφού υποστηρίζει διάφορους τύπους widget, όπως slider, buttons, checkboxes και text inputs, που μπορούν να χρησιμοποιηθούν για την αλληλεπίδραση με τον χρήστη και την εμφάνιση αποτελεσμάτων.

Η υλοποίηση βασίστηκε σε ένα ισχυρό οικοσύστημα βιβλιοθηκών και frameworks που επέτρεψαν την αποτελεσματική διαχείριση των δεδομένων, την εκπαίδευση αλγορίθμων μηχανικής μάθησης και βαθιάς μάθησης, καθώς και τη δημιουργία μιας εύχρηστης εφαρμογής για αυτοματοποιημένη οπτικοποίηση δεδομένων.

Η συνδυαστική χρήση εργαλείων όπως **scikit-learn, TensorFlow, AutoML frameworks και Streamlit** διασφάλισε μια ολοκληρωμένη και ευέλικτη προσέγγιση για την ανάλυση δεδομένων και την ανάπτυξη του συστήματος.

5.2 *Συλλογή Δεδομένων - Data Collection*

Στην έρευνα μας επικεντρωθήκαμε στην επιλογή συνόλων δεδομένων που προσφέρουν πολυμορφικά δεδομένα, όπου δηλαδή χαρακτηρίζονται από ποικιλία διαστάσεων, περιεχομένου και διαφόρων χρήσεων. Κατά την διαδικασία της αναζήτησης θέσαμε ως κριτήριο την αναζήτηση δεδομένων εισόδου φτιαγμένα από άλλους αναλυτές δεδομένων. Οι γραφικές που ψάξαμε πρέπει να περιέχουν δεδομένα σε μορφή X και Y στηλών μαζί με την ετικέτα που χαρακτηρίζει την γραφική παράσταση που απεικονίζεται. Έτσι χρησιμοποιήσαμε το Plotly Community Feed, μία πλατφόρμα που παρέχεται από την κοινότητα του Plotly, όπου οι users μπορούν να μοιραστούν, να ανταλλάξουν απόψεις και να συνεργαστούν σε Plotly projects. Η πλατφόρμα αυτή ικανοποιούσε τα κριτήρια μας εφόσον οι χρήστες του Community αξιολογούν και παρέχουν ανατροφοδότηση στα projects-γραφικές άλλων χρηστών, αλλά επίσης συμμετέχουν ενεργά στην τροποποίηση και βελτίωση αυτών των projects. Αυτή η διαδραστική προσέγγιση εξασφαλίζει ότι οι γραφικές οπτικοποιήσεις των χρηστών εξελίσσονται και βελτιώνονται συνεχώς.

Το Plotly Community Feed προσφέρει ένα API, μέσω του οποίου οι εγγεγραμμένοι χρήστες μπορούν να αναζητούν διαγράμματα και σύνολα δεδομένων (grids) χρησιμοποιώντας συγκεκριμένους όρους. Μέσω αιτήσεων (requests) στο API, αντλήθηκαν δεδομένα για τέσσερις διαφορετικούς τύπους γραφικών παραστάσεων, Bar Chart, Scatter Plot, Line Chart και Pie Chart. ο API επέστρεψε πίνακες αποτελεσμάτων σε μορφή URLs, τα οποία οδηγούσαν στις σελίδες του Plotly Community Feed όπου ήταν διαθέσιμα τα δεδομένα των διαγραμμάτων. Από εκεί, κατεβάσαμε τα αρχεία JSON που περιείχαν τις πληροφορίες κάθε γραφήματος. Ο συνολικός αριθμός των διαθέσιμων γραφημάτων ανά κατηγορία ήταν:

- 14.070 για Bar Charts

- 9.890 για Line Charts
- 10.050 για Pie Charts
- 9.810 για Scatter Plots

Συνολικά, αποκτήθηκε ένα σύνολο 43.820 JSON αρχείων.

Κάθε JSON αρχείο περιείχε πληροφορίες σχετικά με τη γραφική απεικόνιση, οι οποίες ήταν οργανωμένες στο top-level κλειδί ‘data’ όπου περιείχε μια ή περισσότερες σειρές δεδομένων, το κλειδί ‘layout’ που ορίζει τις οπτικές και δομικές ιδιότητες του γραφήματος, όπως τίτλους, χαρακτηριστικά αξόνων κτλ. και το κλειδί ‘frames’ ένας πίνακας που χρησιμοποιείται για τον ορισμό πλαισίων κινούμενων εικόνων. Εστιάζοντας στο κλειδί **data**, τα κύρια πεδία που μας ενδιέφεραν ήταν:

- Το ‘uid’ αναφέρεται σε ένα μοναδικό username – id (uid) του δημιουργού του project.
- Το ‘name’ είναι η ετικέτα-όνομα του γραφήματος.
- Το ‘type’ είναι ο τύπος του γραφήματος.
- Το ‘x’ και ‘y’ ή “labels” και “values” είναι πίνακες που περιέχουν τις τιμές του άξονα των x και y αντίστοιχα.
- Προαιρετικά το ‘markers’ που ορίζει κάποια χαρακτηριστικά της εμφάνισης της γραφικής όπως χρώμα και περίγραμμα.

Τα κλειδιά layout, frames και markers δεν θεωρήθηκαν χρήσιμα για τη διαδικασία και απορρίφθηκαν. Επίσης απορρίφθηκε το πεδίο name εφόσον μπορεί να περιέχει στον τίτλο το όνομα της γραφικής που χρησιμοποιήθηκε. Στην εικόνα 5.2 φαίνεται ένα παράδειγμα JSON αρχείου μιας bar γραφικής με τίτλος “yearly returns in %” μαζί με την γραφική οπτικοποίηση χρησιμοποιόντας το Plotly.

```
{
  "data": [
    {
      "uid": "07a8c2",
      "name": "yearly returns in %",
      "type": "bar",
      "x": [
        2008,
        2009,
        2010,
        2011,
        2012,
        2013,
        2014,
        2015
      ],
      "y": [
        154.7425,
        155.4425,
        75.4675,
        135.10750000000002,
        92.0075,
        207.3225,
        115.41,
        160.26,
        1095.7599999999998
      ]
    }
  ],
  "layout": {
    "title": "Nifty Algo Yearly Returns",
    "width": 1061,
    "xaxis": {
      "type": "linear",
      "range": [
        2007.5,
        2015.5
      ],
      "title": "year",
      "autorange": true
    },
    "yaxis": {
      "type": "linear",
      "range": [
        0,
        218.2342105263158
      ],
      "title": "Returns in %",
      "autorange": true
    },
    "height": 572,
    "autosize": true
  },
  "frames": []
}
```


Εικόνα 5.2: παράδειγμα γραφικής και json file όπως παρέχεται από το Ploply. Πηγή: https://plotly.com/~squareoff/282/nifty-algo-yearly-returns/#plot

Δεδομένου ότι η πηγή των δεδομένων δεν ήταν σχεδιασμένη για συγκεκριμένη χρήση στην αυτόματη οπτικοποίηση, εντοπίστηκαν διάφορα προβλήματα που απαιτούσαν επεξεργασία:

Διπλότυπα δεδομένα απο τον ίδιο χρήστη: Πολλά JSON αρχεία είχαν το ίδιο ‘uid’, γεγονός που υποδήλωνε ότι προέρχονται από τον ίδιο χρήστη. Ορισμένοι χρήστες είχαν δημιουργήσει πολλά διαγράμματα χρησιμοποιώντας τα ίδια δεδομένα, γεγονός που θα μπορούσε να προκαλέσει προκατάληψη στα μοντέλα. Για την αντιμετώπιση του προβλήματος, ελέγχθηκαν όλα τα αρχεία με το ίδιο uid και διατηρήθηκε μόνο ένα, απορρίπτοντας τα υπόλοιπα εφόσον περιείχαν τα ίδια δεδομένα.

Απομάκρυνση Dashboards: Μερικά αρχεία περιείχαν dashboards, δηλαδή πολλαπλά γραφήματα σε μία οθόνη, απεικονίζοντας τα ίδια δεδομένα με διαφορετικούς τύπους

διαγραμμάτων. Δεν υπήρχε κριτήριο επιλογής της σημαντικότερης γραφικής απεικόνισης, οπότε όλα τα dashboards αφαιρέθηκαν.

Ανισορροπία στις κενές τιμές των αξόνων X και Y: Σε ορισμένα αρχεία υπήρχαν πολλές κενές τιμές στον έναν άξονα, ενώ ο άλλος άξονας είχε σημαντικά περισσότερες καταχωρήσεις. Αν και τα κενά δεδομένα είναι αναμενόμενα σε πραγματικές συνθήκες, η μεγάλη δυσαναλογία μεταξύ των αξόνων προκαλεί θόρυβο και δυσχεραίνει την εκπαίδευση των μοντέλων. Ως λύση, αποφασίστηκε η απομάκρυνση γραφημάτων όπου οι κενές τιμές του ενός άξονα υπερέβαιναν το 10% του συνόλου των τιμών του άλλου άξονα. Για τα γραφήματα που παρέμειναν διαχειριστήκαμε τις κενές τιμές με τις μεθόδους που αναλύουμε στην ενότητα 5.4 Επεξεργασία χαρακτηριστικών εισόδου.

Εξισορρόπηση του dataset: Για να αποφευχθεί η προκατάληψη υπέρ των πιο κοινών τύπων γραφικών παραστάσεων, εξισορροπήθηκε ο αριθμός των γραφημάτων ανά κατηγορία. Κρατήθηκαν 4.635 αρχεία ανά τύπο γραφικής παράστασης, διαμορφώνοντας ένα τελικό σύνολο 18.540 αρχείων.

Η διαδικασία συλλογής και προεπεξεργασίας δεδομένων ολοκληρώθηκε με στόχο τη δημιουργία ενός ποιοτικού dataset κατάλληλου για την εκπαίδευση μοντέλων ταξινόμησης. Το dataset διαμορφώθηκε ώστε να είναι ισορροπημένο, καθαρό από θόρυβο και περιείχε δεδομένα τα οποία αντικατόπτριζαν πραγματικές επιλογές αναλυτών δεδομένων. Στα επόμενα στάδια, αυτά τα δεδομένα χρησιμοποιήθηκαν για την εξαγωγή χαρακτηριστικών (feature extraction) και την ανάπτυξη μοντέλων μηχανικής μάθησης για την αυτόματη επιλογή γραφικής οπτικοποίησης.

5.3 *Εξαγωγή χαρακτηριστικών - Feature Extraction*

Τα δεδομένα που συλλέχθηκαν από το Plotly Community Feed περιλαμβάνουν πληροφορίες για κάθε γραφική απεικόνιση με τη μορφή των X και Y στηλών. Ωστόσο, οι στήλες αυτές από μόνες τους δεν αρκούν για τη μοντελοποίηση ταξινομητών, καθώς δεν αποκαλύπτουν άμεσα κάποιο μοτίβο που θα μπορούσε να αξιοποιηθεί από έναν αλγόριθμο μηχανικής μάθησης. Για να ενισχυθεί η πληροφόρηση που παρέχουν τα δεδομένα, εφαρμόστηκε εξαγωγή χαρακτηριστικών (feature extraction), μια διαδικασία που αποσκοπεί στη μετατροπή των αρχικών δεδομένων σε ένα πιο δομημένο σύνολο αριθμητικών και κατηγορηματικών ιδιοτήτων που περιγράφουν τις ιδιότητες των δεδομένων.

Η εξαγωγή χαρακτηριστικών βασίστηκε στη μεθοδολογία που παρουσιάστηκε στην Ενότητα 4.2: Εξαγωγή Χαρακτηριστικών της Προτεινόμενης Προσέγγισης, με στόχο να διατηρηθούν και να αξιολογηθούν όλα τα χαρακτηριστικά στο Πείραμα 6.2: Αξιολόγηση Χαρακτηριστικών με Random Forest και Feature Importance. Ο αλγόριθμος υπολόγισε τη

σημασία κάθε χαρακτηριστικού μετρώντας τη συμβολή του στη βελτίωση της ακρίβειας των ταξινομήσεων. Η ανάλυση αυτή επέτρεψε να εντοπιστούν τα πιο κρίσιμα χαρακτηριστικά, τα οποία θα μπορούσαν να διατηρηθούν σε ενδεχόμενη μελλοντική βελτιστοποίηση του συστήματος.

Η διαδικασία εξαγωγής χαρακτηριστικών ξεκινά με την κατηγοριοποίηση των δεδομένων στις στήλες X και Y, καθώς ο τύπος κάθε στήλης καθορίζει το είδος των χαρακτηριστικών που μπορούν να εξαχθούν. Κάθε στήλη χαρακτηρίστηκε ως ένας από τους παρακάτω τύπους:

- Αριθμητική - Quantitative ('q'): Περιλαμβάνει αριθμητικά δεδομένα (ακέραιους, δεκαδικούς, ποσοστά).
- Κατηγορηματική - Categorical ('c'): Περιλαμβάνει κατηγορίες, ετικέτες, κείμενα ή σύμβολα.
- Ημερομηνία - Datetime ('d'): Περιλαμβάνει ημερομηνίες και χρονικά δεδομένα.

Η κατηγοριοποίηση των datetime δεδομένων έγινε με προσαρμοσμένη συνάρτηση που αναγνωρίζει διαφορετικές μορφές ημερομηνιών, όπως φαίνεται στον Πίνακα 5.3.1.

Πίνακας 5.3.1: Datetime Formats

Datetime Formats	'yyyy-m-d' 'yyyy-mm-d', 'yyyy-m-dd', 'yyyy-mm-dd', 'yyyy-mm-dd xx:xx:xx', 'yyyy-yy', 'yyyy-yyyy', 'yyyy-mm-dd', 'mm/dd/yyyy', 'mm/dd/yy', 'm/d/yyyy'

Έπειτα με βάση των τύπο της στήλης εξάγουμε αντίστοιχα όπως έχουμε ομαδοποίηση στη Ενότητα 4 categorical ή quantitative χαρακτηριστικά. Κάθε entry στο dataset περιέχει όλα τα χαρακτηριστικά και των δύο στηλών ανεξαρτήτως κατηγορίας. Αν δηλαδή μια στήλη σε ένα entry κατηγοριοποιήθηκε σε categorical, τα αντίστοιχα quantitative χαρακτηριστικά για αυτή την στήλη δεν έχουν παραλειφθεί απο το dataset αλλά επιλέχθηκε να παραμείνουν κενά ώστε να τα διαχειριστούμε αργότερα κατα την επεξεργασία τους. Έπειτα για κάθε στήλη ανεξάρτητα από το τύπο της εξάγουμε τα sequence και uniqueness features. Τελευταία παίρνουμε τα general pairwise και statistical pairwise χαρακτηριστικά τα οποία δεν είναι μοναδικά για κάθε στήλη όπως τα προηγούμενα αλλα περιέχουν πληροφορία που συσχετίζουν τις δύο στήλες. Τα statistical pairwise χαρακτηριστικά έχουν την ιδιαιτερότητα

ότι αφορούν μόνο στήλες με “quantitative” τύπο άρα για ζευγάρια στηλών όπου έστω ένα από αυτά είναι “categorical” παρέμειναν κενά.

Το τελικό σύνολο χαρακτηριστικών αποτελείται από όλα τα χαρακτηριστικά που προτάθηκαν στην Ενότητα 4.2, χωρίς καμία αφαίρεση σε αυτό το στάδιο. Για κάθε εγγραφή στο dataset διατηρούνται όλα τα χαρακτηριστικά ανεξάρτητα από τον τύπο της στήλης. Αν κάποιο χαρακτηριστικό δεν είναι σχετικό με τον τύπο της στήλης (π.χ. ένα quantitative χαρακτηριστικό για μια categorical στήλη), τότε αφήνεται κενό και αντιμετωπίζεται σε επόμενο στάδιο κατά την προεπεξεργασία των δεδομένων. Στον Πίνακα 5.3.2 παρουσιάζεται η ταξινόμηση των χαρακτηριστικών σε κατηγορίες.

Πίνακας 5.3.2: Τα Χαρακτηριστικά κατηγοριοποιημένα. Με κίτρινο το είδος, καφέ uniqueness, μπλε numerical, γαλάζιο categorical, ροζ general pairwise, κόκκινο statistical pairwise και πράσινο τα sequence features.

#	Feature	Category	#	Feature	Category
1	Type		26	Has Shared Unique	
2	# Entries		27	% Shared Unique Elements	
3	Range		28	Identical Unique	
4	Unique Elem		29	Correlation Value	
5	Has None		30	Correlation P	
6	# None		31	Ks Statistic	
7	% None		32	Ks P	
8	Mean		33	Correlation Significant 005	
9	Median		34	Ks Significant 005	
10	Sample Var		35	Entropy Q	
11	Sample Std		36	Entropy C	
12	Coeff Var		37	Percentage Mode C	
13	Min		38	Kurtosis	
14	Max		39	Skewness	

15	Mean Length		40	% Outliers (1.5 IQR)	
16	Median Length		41	% Outliers (3 IQR)	
17	Min Length		42	% Outliers (1-99 Percentile)	
18	Max Length		43	% Outliers (3 Std Dev)	
19	Std Length		44	Has Outliers (1.5 IQR) (X)	
20	Unique Labels		45	Has Outliers (3 IQR)	
21	# Identical Elements		46	Has Outliers (1-99 Percentile) (X)	
22	Has Shared Elements		47	Has Outliers (3 Std Dev)	
23	% Shared Elements		48	Sortedness	
24	Identical		49	Is Sorted	
25	# Shared Unique Elements		59	Plot Type	

Τα χαρακτηριστικά στις κατηγορίες Type, Categorical, Quantitative, Sequence και Uniqueness είναι σύνολο 35 ανά στήλη και τα χαρακτηριστικά στις κατηγορίες General Pairwise και Statistical Pairwise είναι 13 και αφορούν και τις δύο στήλες. Άρα σύνολο εξάγονται 83 (= 35x2 + 13) χαρακτηριστικά για κάθε γραφική οπτικοποίηση.

Η εξαγωγή των χαρακτηριστικών πραγματοποιήθηκε μέσω Python script, και τα δεδομένα αποθηκεύτηκαν σε ένα CSV αρχείο, ώστε να μπορούν να χρησιμοποιηθούν στα επόμενα

στάδια της ανάλυσης. Η συγκεκριμένη διαδικασία επιτρέπει τυποποίηση της εξαγωγής χαρακτηριστικών, διευκολύνοντας την αναπαραγωγή των αποτελεσμάτων και ευκολία αξιολόγησης μέσω διαφορετικών μοντέλων μηχανικής μάθησης.

5.4 Επεξεργασία Χαρακτηριστικών Εισόδου - Feature Processing and Engineering

Μετά την εξαγωγή των χαρακτηριστικών, ακολουθεί η διαδικασία επεξεργασίας των δεδομένων, η οποία περιλαμβάνει τον διαχωρισμό τους σε κατηγορίες, τη διαχείριση ακραίων τιμών, την κανονικοποίηση, την αντιμετώπιση κενών τιμών και την κωδικοποίηση κατηγορηματικών μεταβλητών.

Τα δεδομένα φορτώθηκαν από το CSV αρχείο και μετατράπηκαν σε Pandas DataFrames. Αρχικά, χωρίστηκαν σε τρεις κατηγορίες. Numerical (Αριθμητικά χαρακτηριστικά): Περιλαμβάνουν συνεχείς αριθμητικές τιμές, όπως το πλήθος εγγραφών, τη διακύμανση και τον συντελεστή συσχέτισης. Categorical (Κατηγορηματικά χαρακτηριστικά): Περιλαμβάνουν χαρακτηριστικά που αναπαριστούν κατηγορίες, όπως ο τύπος των δεδομένων και η ύπαρξη ακραίων τιμών. Boolean (Δυαδικά χαρακτηριστικά): Περιλαμβάνουν χαρακτηριστικά που παίρνουν τιμές 0 ή 1 (False/True).

Για τον διαχωρισμό των δεδομένων, χρησιμοποιήθηκε η ακόλουθη μεθοδολογία:

- Αν μια στήλη ήταν τύπου object, καταχωρήθηκε ως categorical.
- Αν μια στήλη ήταν τύπου bool, μετατράπηκε σε 0 (False) ή 1 (True) και καταχωρήθηκε ως boolean.
- Όλες οι υπόλοιπες στήλες θεωρήθηκαν numerical.

Τα χαρακτηριστικά χωρίστηκαν όπως φαίνεται στον πίνακα 5.4:

Πίνακας 5.4: Τύπος dataframe και τα αντίστοιχα χαρακτηριστικά

Τύπος Dataframe	Χαρακτηριστικά
Numerical	'Num_Entries_X', 'Num_Entries_Y', 'Normalized Range (X)', 'Normalized Range (Y)', 'Num_Unique_X', 'Num_Unique_Y', 'Num_None_X', 'Percentage_None_X', 'Num_None_Y', 'Percentage_None_Y', 'Normalized Mean (X)', 'Normalized Median (X)', 'Sample Variance (X)', 'Sample Standard Deviation (X)', 'Coefficient of Variation (X)', 'Minimum Value (X)', 'Maximum Value (X)', 'Normalized Mean (Y)', 'Normalized Median (Y)', 'Sample Variance (Y)', 'Sample Standard Deviation (Y)', 'Coefficient of Variation (Y)', 'Minimum Value (Y)', 'Maximum Value (Y)', 'Mean C (X)', 'Median C (X)', 'Min C (X)', 'Max C (X)', 'Std C (X)', 'Unique C (X)', 'Mean C (Y)', 'Median C (Y)', 'Min C (Y)', 'Max C (Y)', 'Std C

	(Y)', 'Unique C (Y)', 'Num Identical Elements', 'Percent Shared Elements', 'Identical', 'Num Shared Unique Elements', 'Percent Shared Unique Elements', 'Correlation Value', 'Correlation P', 'Ks Statistic', 'Ks P', 'Q Entropy (X)', 'Q Entropy (Y)', 'C Entropy (X)', 'C Entropy (Y)', 'Perc Mode (X)', 'Perc Mode (Y)', 'Kurtosis (X)', 'Skewness (X)', 'Kurtosis (Y)', 'Skewness (Y)', '% Outliers (1.5 IQR) (X)', '% Outliers (3 IQR) (X)', '% Outliers (1-99 Percentile) (X)', '% Outliers (3 Std Dev) (X)', '% Outliers (1.5 IQR) (Y)', '% Outliers (3 IQR) (Y)', '% Outliers (1-99 Percentile) (Y)', '% Outliers (3 Std Dev) (Y)', 'Is Sorted (X)'
Categorical	'X_Type', 'Y_Type', 'Has Outliers (1.5 IQR) (X)', 'Has Outliers (3 IQR) (X)', 'Has Outliers (1-99 Percentile) (X)', 'Has Outliers (3 Std Dev) (X)', 'Has Outliers (1.5 IQR) (Y)', 'Has Outliers (3 IQR) (Y)', 'Has Outliers (1-99 Percentile) (Y)', 'Has Outliers (3 Std Dev) (Y)', 'Sortedness (X)'
Boolean	'Has_None_X', 'Has_None_Y', 'Has Shared Elements', 'Has Shared Unique Elements', 'Identical Unique', 'Correlation Significant 005', 'Ks Significant 005'

Επεξεργασία Numerical Χαρακτηριστικών

Για την προετοιμασία των numerical χαρακτηριστικών, εφαρμόστηκε μία διαδικασία εξομάλυνσης και μετασχηματισμού με στόχο τη βελτίωση της ποιότητας των δεδομένων και τη μείωση των ακραίων τιμών.

Προκειμένου να αντιμετωπίσουμε τις ακραίες τιμές που μπορεί να οφείλονται σε θόρυβο, σφάλματα καταγραφής ή ασυνήθιστες μεταβολές, εφαρμόστηκε αποκοπή τιμών (clipping) στα όρια του 1ου και 99ου εκατοστημορίου (1st and 99th percentile cutoff). Για την εξαγωγή αυτών των ορίων, αξιοποιήθηκε η έτοιμη συνάρτηση quantile() της βιβλιοθήκης Pandas, μέσω της οποίας υπολογίστηκαν οι τιμές που αντιστοιχούν στο 1ο και 99ο εκατοστημόριο για κάθε numerical χαρακτηριστικό. Αφού καθορίστηκαν τα όρια, οι ακραίες τιμές περιορίστηκαν ως εξής:

- Αν μια τιμή υπερβαίνει το 99ο εκατοστημόριο, αντικαθίσταται από την τιμή του 99ου εκατοστημορίου.
- Αν μια τιμή είναι μικρότερη από το 1ο εκατοστημόριο, αντικαθίσταται από την τιμή του 1ου εκατοστημορίου.

Αυτή η τεχνική επέτρεψε τον περιορισμό των ακραίων τιμών χωρίς να επηρεάσουμε σημαντικά την κατανομή των δεδομένων, διατηρώντας τις βασικές πληροφορίες που εμπεριέχουν.

Μετά την αποκοπή των ακραίων τιμών, εφαρμόστηκε κανονικοποίηση (standardization) στα numerical χαρακτηριστικά, ώστε όλα τα χαρακτηριστικά να έχουν παρόμοια κλίμακα και να αποφευχθεί η υπερβολική επιρροή χαρακτηριστικών με μεγάλες αριθμητικές τιμές.

Η διαδικασία πραγματοποιήθηκε με την StandardScaler της βιβλιοθήκης scikit-learn, η οποία μετασχηματίζει κάθε numerical χαρακτηριστικό ως εξής:

$$X' = \frac{X - \mu}{\sigma}$$

όπου:

- Χ είναι η αρχική τιμή,
- μ είναι ο μέσος όρος των τιμών του χαρακτηριστικού,
- σ είναι η τυπική απόκλιση.

Με αυτόν τον μετασχηματισμό, κάθε χαρακτηριστικό αποκτά μέσο όρο 0 και τυπική απόκλιση 1, γεγονός που διευκολύνει τη σύγκριση μεταξύ των χαρακτηριστικών και συμβάλλει στη σταθερότητα των αλγορίθμων μηχανικής μάθησης.

Τέλος, για την αντιμετώπιση των κενών τιμών (NaN), εφαρμόστηκε αντικατάσταση (imputation) με τη χρήση της μέσης τιμής (mean imputation). Ενώ εφαρμόστηκε και αντικατάσταση με διάμεσο (median) προτιμήθηκε ο μέσος όρος (mean) εφόσον εμφάνισε καλύτερα αποτελέσματα. Συγκεκριμένα, για κάθε numerical χαρακτηριστικό υπολογίστηκε η μέση τιμή των μη κενών τιμών, και όπου υπήρχαν NaN τιμές, αυτές αντικαταστάθηκαν με την αντίστοιχη μέση τιμή του χαρακτηριστικού. Αυτή η διαδικασία διασφαλίζει τη διατήρηση όλων των διαθέσιμων δεδομένων χωρίς την ανάγκη αφαίρεσης δειγμάτων, που θα μπορούσε να οδηγήσει σε απώλεια πληροφορίας.

Με την ολοκλήρωση αυτών των βημάτων, τα numerical χαρακτηριστικά είναι έτοιμα προς χρήση στα επόμενα στάδια ανάλυσης και εκπαίδευσης μοντέλων μηχανικής μάθησης.

Επεξεργασία Categorical Χαρακτηριστικών

Η προεπεξεργασία των κατηγορηματικών χαρακτηριστικών περιλάμβανε αντικατάσταση κενών τιμών και μετατροπή τους σε αριθμητική μορφή (One-Hot Encoding).

Για την αναπλήρωση αυτών των κενών τιμών, εφαρμόστηκε αντικατάσταση με την πιο συχνά εμφανιζόμενη κατηγορία (mode imputation). Η τάση (mode) ενός χαρακτηριστικού είναι η κατηγορία που εμφανίζεται περισσότερες φορές μέσα στο dataset. Αυτή η μέθοδος επιτρέπει τη διατήρηση όλων των δεδομένων χωρίς να απαιτείται αφαίρεση εγγραφών, κάτι που θα μπορούσε να οδηγήσει σε μείωση του συνολικού μεγέθους του dataset. Η διαδικασία πραγματοποιήθηκε ως εξής:

1. Για κάθε κατηγορηματικό χαρακτηριστικό, υπολογίστηκε η πιο συχνή τιμή (mode).
2. Όπου υπήρχαν NaN τιμές στο χαρακτηριστικό, αυτές αντικαταστάθηκαν με την υπολογισμένη mode τιμή του χαρακτηριστικού.

Οι περισσότεροι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να επεξεργαστούν απευθείας κατηγορηματικά χαρακτηριστικά, καθώς απαιτούν αριθμητικές τιμές ως είσοδο. Για αυτόν τον λόγο, εφαρμόστηκε One-Hot Encoding, μία τεχνική που μετατρέπει τις κατηγορηματικές μεταβλητές σε δυαδικές αριθμητικές τιμές. Χρησιμοποιήθηκε ο OneHotEncoder της βιβλιοθήκης scikit-learn, ο οποίος εφαρμόζει την εξής διαδικασία:

1. Δημιουργεί μία νέα στήλη για κάθε διαφορετική τιμή της κατηγορηματικής μεταβλητής.
2. Στη νέα αναπαράσταση, κάθε δείγμα παίρνει την τιμή **1** στη στήλη που αντιστοιχεί στην αρχική του κατηγορία και **0** σε όλες τις υπόλοιπες.

Αυτή η προσέγγιση επιτρέπει στο μοντέλο να χρησιμοποιήσει τις κατηγορηματικές πληροφορίες χωρίς να δημιουργείται σχέση τάξης (ordinality) μεταξύ των κατηγοριών, όπως θα συνέβαινε με απλές αριθμητικές τιμές.

Επεξεργασία Boolean Χαρακτηριστικών

Τα boolean χαρακτηριστικά δεν απαιτούν περαιτέρω επεξεργασία, καθώς δεν περιέχουν κενές τιμές (NaN), επομένως δεν απαιτείται διαδικασία αντικατάστασης, είναι ήδη αριθμητικά, αφού μπορούν να αναπαρασταθούν με τις τιμές 0 (False) και 1 (True) και έχουν φυσική κλιμάκωση μεταξύ 0 και 1, επομένως δεν χρειάζεται επιπλέον διαδικασία κανονικοποίησης ή μετασχηματισμού τιμών. Αυτό τα καθιστά άμεσα αξιοποιήσιμα από τους αλγορίθμους μηχανικής μάθησης, χωρίς την ανάγκη για μετατροπή ή προσαρμογή.

Τα numerical, categorical και boolean χαρακτηριστικά συγχωνεύτηκαν σε ένα ενιαίο DataFrame. Το dataset χωρίστηκε σε τρία υποσύνολα: 60% Training Set (για την εκπαίδευση των μοντέλων), 20% Test Set (για την αρχική αξιολόγηση των μοντέλων) και 20% Validation Set (για την τελική επικύρωση της απόδοσης). Η συνολική διαδικασία χωρισμού των κατηγορηματικών και αριθμητικών χαρακτηριστικών, η επεξεργασία τους, η συγχώνευση και ο χωρισμος σε train,test και val απεικονίζεται στην εικόνα 5.4.

Δυναμική Διαμόρφωση Επεξεργασίας

Κατά την ανάπτυξη της διαδικασίας προεπεξεργασίας των δεδομένων, επιχειρήσαμε να τυποποιήσουμε τη διαδικασία μέσω μιας γενικής συνάρτησης, ώστε να μπορεί να εφαρμοστεί ανάλογα με το μοντέλο που χρησιμοποιείται. Όπως περιγράψαμε και στην ενότητα 4 ανάλογα με το μοντέλο που εκπαιδεύουμε κάποια επεξεργασία στα δεδομένα ενδέχεται να

παρακαμφθεί. Η ανάγκη για ευελιξία στην προετοιμασία των δεδομένων προέκυψε από το γεγονός ότι διαφορετικοί αλγόριθμοι μηχανικής μάθησης έχουν διαφορετικές απαιτήσεις προεπεξεργασίας.

Για τον σκοπό αυτό, δημιουργήθηκε η συνάρτηση preprocess_data(), η οποία επιτρέπει τον δυναμικό καθορισμό των βημάτων προεπεξεργασίας που θα εφαρμοστούν.

```
preprocess_data(X, apply_percentile_cutoff, apply_scaling,
impute_missing, apply_one_hot_encoding)
```

Η συνάρτηση αυτή δέχεται ως είσοδο το dataset X και τέσσερις Boolean παραμέτρους που καθορίζουν ποια στάδια επεξεργασίας θα εκτελεστούν:

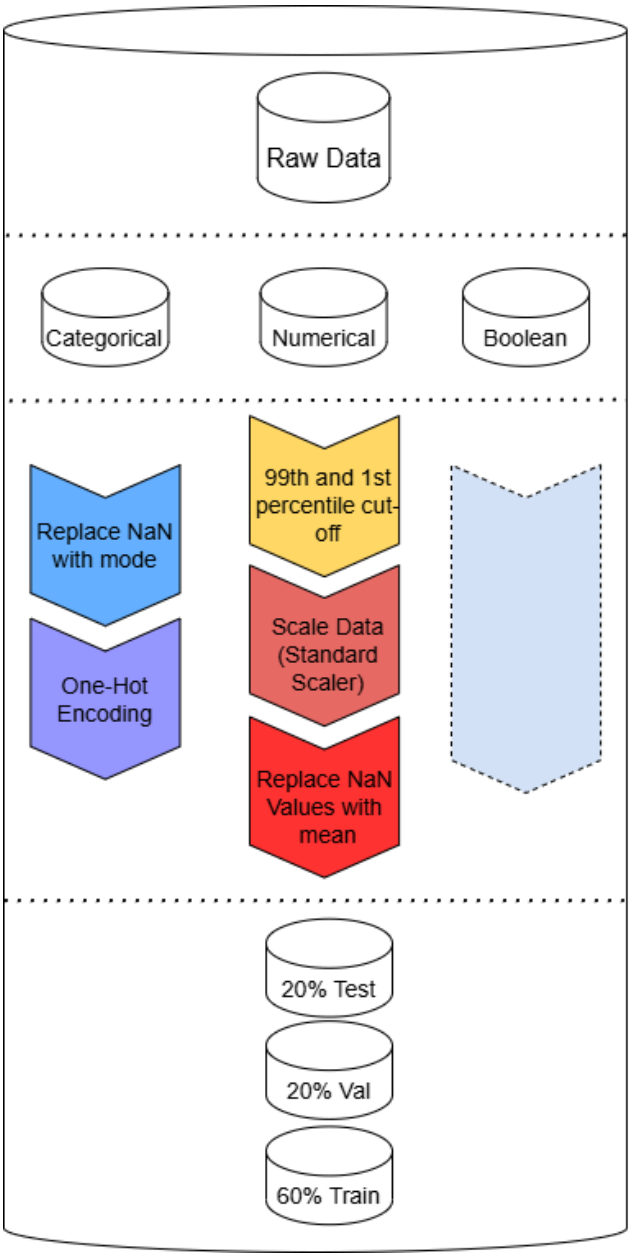
- apply_percentile_cutoff: Εφαρμόζει αποκοπή των ακραίων τιμών στο 1ο και 99ο εκατοστημόριο.
- apply_scaling: Κανονικοποιεί τα αριθμητικά χαρακτηριστικά ώστε να έχουν μέσο όρο 0 και τυπική απόκλιση 1.
- impute_missing: Αντικαθιστά τις κενές τιμές με την κατάλληλη μέθοδο (mean/mode imputation).
- apply_one_hot_encoding: Μετατρέπει τις κατηγορηματικές μεταβλητές σε αριθμητική μορφή μέσω One-Hot Encoding.

Η χρήση αυτής της συνάρτησης επέτρεψε τη δημιουργία μιας ενιαίας και επαναχρησιμοποιήσιμης ροής επεξεργασίας δεδομένων, δίνοντας τη δυνατότητα να προσαρμόσουμε τη διαδικασία ανάλογα με το μοντέλο που εκπαιδεύουμε.

Για παράδειγμα, στην περίπτωση του Random Forest, όπου το μοντέλο δεν είναι ευαίσθητο στην κλίμακα των αριθμητικών χαρακτηριστικών, δεν εφαρμόστηκε κανονικοποίηση (apply_scaling=False). Αντίστοιχα, επειδή τα δέντρα απόφασης μπορούν να διαχειριστούν απευθείας κενές τιμές, δεν εφαρμόστηκε imputation (impute_missing=False). Η κλήση της συνάρτησης για το Random Forest έγινε ως εξής:

```
X = preprocess_data(X, apply_percentile_cutoff=True,
apply_scaling=False, impute_missing=False,
apply_one_hot_encoding=True)
```

Με αυτόν τον τρόπο, διασφαλίστηκε ότι η προεπεξεργασία προσαρμόζεται στις απαιτήσεις κάθε μοντέλου, διατηρώντας παράλληλα μια τυποποιημένη διαδικασία που μπορεί εύκολα να επαναχρησιμοποιηθεί και να επεκταθεί για διαφορετικές προσεγγίσεις.



Εικόνα 5.4: Διαδικασία Επεξεργασίας Δεδομένων

5.5 *Μοντελοποίηση - Modeling*

Μετά την επεξεργασία των δεδομένων εισόδου, ακολουθεί η τελική υλοποίηση και ο καθορισμός της αρχιτεκτονικής του ταξινομητή. Για τη μοντελοποίηση, επιλέξαμε να πειραματιστούμε με έξι διαφορετικούς αλγορίθμους ταξινόμησης: Random Forest, k-Nearest

Neighbors (k-NN), Support Vector Machines (SVM), Logistic Regression, Neural Networks και AutoML μέσω των AutoGluon και AutoKeras.

Η βασική μας επιδίωξη είναι ο ταξινομητής να μπορεί να προσαρμόζεται στις ετικέτες που επιθυμεί να ταξινομήσει ο χρήστης μέσω της εφαρμογής. Δεδομένου ότι οι ετικέτες δεν είναι πολλές, επιλέξαμε να χρησιμοποιήσουμε μια multiclass one-vs-one (OvO) στρατηγική αντί να εκπαιδεύσουμε έναν ενιαίο ταξινομητή που θα αναγνωρίζει όλες τις κατηγορίες ταυτόχρονα.

Συγκεκριμένα, εκπαιδεύουμε ξεχωριστά μοντέλα για κάθε πιθανό συνδυασμό ετικετών, επιτρέποντας στην προσέγγισή μας να προσαρμόζεται ευέλικτα στις ανάγκες του χρήστη. Για παράδειγμα, αν ο χρήστης επιθυμεί να ταξινομήσει δεδομένα μεταξύ LINE-BAR-PIE, η εφαρμογή χρησιμοποιεί το αντίστοιχο μοντέλο που έχει εκπαιδευτεί για αυτόν τον συνδυασμό. Συνολικά, δημιουργούμε 11 διαφορετικά μοντέλα, δηλαδή 6 μοντέλα για δυαδική ταξινόμηση (C=2), 4 μοντέλα για ταξινόμηση τριών κατηγοριών (C=3) και 1 μοντέλο για όλες τις ετικέτες μαζί (C=4). Για λόγους απλοποίησης, αναφερόμαστε στους ταξινομητές με βάση τις ετικέτες που περιλαμβάνουν. Για παράδειγμα ο ταξινομητής με τις ετικέτες ‘line’, ‘scatter’, ‘bar’ και ‘pie’ αναφέρεται ως 'lsbp' και ο ταξινομητής με τις ετικέτες ‘bar’ και ‘pie’ αναφέρεται ως ‘bp’. Αυτή η στρατηγική προσφέρει μεγαλύτερη ευελιξία, καθώς ο χρήστης μπορεί να επιλέξει δυναμικά τον συνδυασμό ετικετών που θέλει να προβλέψει, χωρίς να περιορίζεται από ένα προκαθορισμένο ενιαίο μοντέλο.

5.5.1 Random Forest

Ο αλγόριθμος Random Forest βασίζεται στον συνδυασμό πολλών τυχαιοποιημένων Decision Trees, με την τελική πρόβλεψη να προκύπτει μέσω της πλειοψηφικής ψήφου όταν πρόκειται για ταξινόμηση ή μέσω του μέσου όρου των προβλέψεων στην περίπτωση παλινδρόμησης. Αυτή η προσέγγιση καθιστά το Random Forest ανθεκτικό στον θόρυβο και στην υπερεκπαίδευση (overfitting), ειδικά σε περιπτώσεις όπου ο αριθμός των μεταβλητών είναι σημαντικά μεγαλύτερος από τον αριθμό των παρατηρήσεων, όπως συμβαίνει στο dataset που χρησιμοποιούμε, το οποίο περιλαμβάνει 83 χαρακτηριστικά και 4 κλάσεις. Επιπλέον, ο συγκεκριμένος αλγόριθμος παρέχει τη δυνατότητα υπολογισμού της σχετικής σημασίας των χαρακτηριστικών, γεγονός που συμβάλλει στην καλύτερη κατανόηση των δεδομένων και στην επιλογή των πιο κρίσιμων μεταβλητών. Η ανάλυση της συνεισφοράς κάθε χαρακτηριστικού περιγράφεται λεπτομερώς στην Ενότητα 6.2 – Πείραμα 2: Αξιολόγηση χαρακτηριστικών με Random Forest και Feature Importance, όπου διερευνήσαμε ποια χαρακτηριστικά παίζουν σημαντικότερο ρόλο στη διαδικασία της ταξινόμησης.

Προκειμένου να διασφαλίσουμε ότι τα δεδομένα μας ήταν κατάλληλα για την εκπαίδευση του Random Forest Classifier, πραγματοποιήθηκε συγκεκριμένη διαδικασία προετοιμασίας. Αρχικά, εφαρμόσαμε αποκοπή ακραίων τιμών στα αριθμητικά χαρακτηριστικά χρησιμοποιώντας το 1st και 99th percentile cutoff ώστε να μειώσουμε την επίδραση πιθανών εκτροπών ή σφαλμάτων στα δεδομένα. Στη συνέχεια, οι κατηγορηματικές μεταβλητές κωδικοποιήθηκαν αριθμητικά με τη μέθοδο One-Hot Encoding, αφού το Random Forest δεν είναι σχεδιασμένο να επεξεργάζεται άμεσα μη αριθμητικές τιμές. Όσον αφορά τις ελλιπείς τιμές, δεν εφαρμόστηκε κάποια τεχνική αντικατάστασης, καθώς τα Random Forests έχουν την ικανότητα να διαχειρίζονται εγγενώς τα missing values χωρίς να απαιτείται προεπεξεργασία. Μετά την επεξεργασία των δεδομένων, τα χωρίσαμε σε τρία σύνολα: 60% training set, 20% test set και 20% validation set, ώστε να διασφαλιστεί η ορθότητα της αξιολόγησης του μοντέλου. Η συνάρτηση preprocess_data() τροποποιήθηκε κατάλληλα ώστε να περιλαμβάνει μόνο τα απαραίτητα βήματα επεξεργασίας για το συγκεκριμένο μοντέλο, παραλείποντας το scaling και την αναπλήρωση των ελλιπών τιμών. Ο τελικός τρόπος κλήσης της συνάρτησης ήταν ο εξής:

```
X = preprocess_data(X, apply_percentile_cutoff = True, apply_scaling
= False, impute_missing = False, apply_one_hot_encoding = True)
```

Για τη βελτιστοποίηση του μοντέλου, χρησιμοποιήσαμε Grid Search, μια μέθοδο εξαντλητικής αναζήτησης όλων των πιθανών συνδυασμών των υπερπαραμέτρων, με στόχο την εύρεση του ιδανικού συνόλου ρυθμίσεων που μεγιστοποιούν την απόδοση του ταξινομητή. Αυτή η διαδικασία περιγράφεται στην Ενότητα 6.1.1 – Πείραμα Grid Search για Random Forest με 4 κλάσεις, όπου πειραματιστήκαμε με διάφορες τιμές για τις βασικές υπερπαραμέτρους. Συγκεκριμένα, εξετάσαμε τον αριθμό των δέντρων (n_estimators), το μέγιστο βάθος του δέντρου (max_depth), το ελάχιστο απαιτούμενο μέγεθος δείγματος για τη διάσπαση ενός κόμβου (min_samples_split), τον αριθμό χαρακτηριστικών που λαμβάνονται υπόψη σε κάθε διάσπαση (max_features), καθώς και τη συνάρτηση κριτηρίου διαχωρισμού (criterion).

Μετά την εκτέλεση του Grid Search, το μοντέλο που επιλέχθηκε ως βέλτιστο περιλάμβανε τις εξής ρυθμίσεις: n_estimators=200, το οποίο σημαίνει ότι το δάσος αποτελείται από 200 δέντρα, max_depth=20, που επιτρέπει στα δέντρα να έχουν μέγιστο επιτρεπτό βάθος μέχρι 20 επίπεδα, min_samples_split=2, που καθορίζει ότι ένας κόμβος μπορεί να διασπαστεί αν περιέχει τουλάχιστον 2 δείγματα, max_features=sqrt, που ορίζει ότι σε κάθε διάσπαση εξετάζεται ένας αριθμός χαρακτηριστικών ίσος με την τετραγωνική ρίζα του συνολικού αριθμού των χαρακτηριστικών, και criterion='gini', το οποίο χρησιμοποιείται για την απόφαση διαχωρισμού των κόμβων στα δέντρα.

Πίνακας 5.5.1: Υπερπαράμετροι του Random Forest

n_estimators	200
criterion	“gini”
max_depth	20
min_samples_split	2
max_features	“sqrt”

5.5.2 K-Nearest Neighbors

Στην επόμενη μέθοδο που εφαρμόστηκε, το k-nearest neighbors (kNN), χρησιμοποιήσαμε έναν απλό αλλά αποτελεσματικό αλγόριθμο μηχανικής μάθησης για την ταξινόμηση των δεδομένων. Το kNN λειτουργεί με την ταξινόμηση των unlabeled δεδομένων αναθέτοντάς τους την κλάση των πιο κοντινών labeled δειγμάτων στον πολυδιάστατο χώρο των χαρακτηριστικών. Η βασική ιδέα είναι ότι τα δεδομένα με παρόμοια χαρακτηριστικά τείνουν να βρίσκονται κοντά μεταξύ τους.

Για να προετοιμάσουμε τα δεδομένα πριν από την εκπαίδευση, πραγματοποιήσαμε κατάλληλο data preprocessing. Όπως και πριν, εφαρμόσαμε αποκοπή ακραίων τιμών (1st και 99th percentile cut-off) στα αριθμητικά χαρακτηριστικά ώστε να περιορίσουμε την επίδραση των outliers. Ο αλγόριθμος kNN είναι ευαίσθητος στην κλίμακα των δεδομένων, καθώς βασίζεται στον υπολογισμό αποστάσεων. Για τον λόγο αυτό, εφαρμόσαμε κανονικοποίηση (Standard Scaler) ώστε όλα τα χαρακτηριστικά να έχουν μηδενικό μέσο όρο και τυπική απόκλιση ίση με 1, εξασφαλίζοντας ομοιόμορφη απόσταση μεταξύ τους.

Επιπλέον, το kNN δεν υποστηρίζει απουσιάζουσες τιμές, επομένως εφαρμόστηκε αντικατάσταση (imputation), αντικαθιστώντας τις κενές τιμές στα numerical χαρακτηριστικά με τον μέσο όρο και στα categorical με την πιο συχνή τιμή (mode). Στη συνέχεια, για να καταστήσουμε τα κατηγορηματικά χαρακτηριστικά κατάλληλα για τον αλγόριθμο, εφαρμόσαμε One-Hot Encoding, μετατρέποντας τις κατηγορικές τιμές σε δυαδικές αναπαραστάσεις. Όπως και στις προηγούμενες περιπτώσεις, τα δεδομένα χωρίστηκαν σε 60% train, 20% test και 20% validation sets. Η συνάρτηση preprocess_data() τροποποιήθηκε κατάλληλα ως εξής:

```
X = preprocess_data(X, apply_percentile_cutoff = True, apply_scaling
= True, impute_missing = True, apply_one_hot_encoding = True)
```

Για την ρύθμιση των υπερ παραμέτρων χρησιμοποιήσαμε grid search, όπως και στην περίπτωση του Random Forest.. Οι υπερπαράμετροι που εξετάστηκαν ήταν n_neighbors, ο αριθμός των k-nearest neighbors που λαμβάνονται υπόψη για την ταξινόμηση ενός νέου δείγματος, weights, ορίζει το πώς επηρεάζουν την απόφαση οι γείτονες, δηλαδή αν όλοι συνεισφέρουν εξίσου ή αν δίνεται μεγαλύτερη βαρύτητα στους κοντινότερους και metric, η μετρική απόστασης που χρησιμοποιείται για τον υπολογισμό της ομοιότητας μεταξύ των σημείων δεδομένων.

Το βελτιστοποιημένο μοντέλο kNN που προέκυψε από το Grid Search για KNN με 4 κλάσεις, είχε τις εξής τελικές υπερπαραμέτρους: n_neighbors = 6, metric = Manhattan Distance, weights = Uniform.

Πίνακας 5.5.2: Υπερπαράμετροι του K-Nearest Neighbors

n_neighbors	6
metric	“manhattan”
weights	“uniform”

5.5.3 Support Vector Machines

Η τρίτη μέθοδος που δοκιμάσαμε είναι το Support Vector Machines (SVM), ένας ισχυρός αλγόριθμος μηχανικής μάθησης που στηρίζεται στον διαχωρισμό των δεδομένων μέσω hyperplanes. Ο κύριος στόχος του SVM είναι να εντοπίσει την καλύτερη δυνατή γραμμή (ή επίπεδο σε υψηλότερες διαστάσεις) που διαχωρίζει τα δεδομένα σε διαφορετικές κατηγορίες. Αυτό επιτυγχάνεται με τη χρήση των Support Vectors, δηλαδή των σημείων δεδομένων που βρίσκονται κοντά στο hyperplane και καθορίζουν τη θέση και τον προσανατολισμό του.

Για την προετοιμασία των δεδομένων, ακολουθήσαμε παρόμοια διαδικασία με αυτή του kNN, δεδομένου ότι και το SVM επηρεάζεται από την κλίμακα των χαρακτηριστικών. Εφαρμόσαμε αποκοπή ακραίων τιμών (1st και 99th percentile cutoff), κανονικοποίηση (Standard Scaler) για να διασφαλίσουμε ότι όλα τα χαρακτηριστικά έχουν συγκρίσιμη κλίμακα, αντικατάσταση (imputation) των κενών τιμών και One-Hot Encoding για τη μετατροπή των κατηγορικών

χαρακτηριστικών σε αριθμητικές τιμές. Η συνάρτηση preprocess_data() τροποποιήθηκε ως εξής:

```
X = preprocess_data(X, apply_percentile_cutoff = True, apply_scaling = True, impute_missing = True, apply_one_hot_encoding = True)
```

Για τη βελτιστοποίηση των υπερπαραμέτρων, εφαρμόσαμε Grid Search, όπως και στα προηγούμενα μοντέλα. Οι υπερπαράμετροι που εξετάστηκαν ήταν οι εξής. C (regularization parameter): Ελέγχει την ισορροπία μεταξύ της μεγιστοποίησης του περιθωρίου (margin) και της ελαχιστοποίησης του σφάλματος κατά την εκπαίδευση. Ένα μεγαλύτερο C προσπαθεί να ταξινομήσει όλα τα σημεία σωστά, ενώ ένα μικρότερο C επιτρέπει περισσότερα λάθη ώστε το μοντέλο να γενικεύει καλύτερα. gamma: Ρυθμίζει την επιρροή που έχει κάθε σημείο εκπαίδευσης στον τελικό διαχωρισμό. Ένα υψηλό gamma σημαίνει ότι κάθε σημείο έχει μεγαλύτερη επιρροή, ενώ ένα χαμηλό gamma σημαίνει ότι το μοντέλο επηρεάζεται από ένα πιο γενικό μοτίβο. kernel: Καθορίζει τον τύπο της συνάρτησης πυρήνα που χρησιμοποιείται για τον μετασχηματισμό των δεδομένων σε έναν υψηλότερης διάστασης χώρο. Έτσι, ακόμα και αν τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα στον αρχικό τους χώρο, μπορούν να διαχωριστούν σε έναν μετασχηματισμένο χώρο.

Μετά την εκτέλεση του Grid Search για SVM με 4 κλάσεις, επιλέχθηκαν οι εξής υπερπαράμετροι για το βέλτιστο μοντέλο SVM: C = 100, ισχυρή έμφαση στην ταξινόμηση κάθε δείγματος σωστά., kernel = “rbf”, ο Radial Basis Function (RBF) kernel είναι ένας από τους πιο ισχυρούς και ευέλικτους πυρήνες, επιτρέποντας στο μοντέλο να συλλάβει μη γραμμικά μοτίβα στα δεδομένα και gamma = “auto”, η τιμή του gamma ρυθμίστηκε ώστε να καθορίζεται αυτόματα από το μέγεθος του dataset, εξασφαλίζοντας ισορροπημένη γενίκευση.

Αυτή η επιλογή υπερπαραμέτρων απέδωσε το καλύτερο αποτέλεσμα στο πείραμα 6.1.3 Grid Search για SVM με 4 κλάσεις, αποδεικνύοντας την ικανότητα του SVM να προσαρμόζεται σε πολύπλοκα μοτίβα και να εκμεταλλεύεται τις ιδιότητες των Support Vectors για ακριβή ταξινόμηση.

Πίνακας 5.5.3: Υπερπαράμετροι του SVM

c	100
kernel	“rbf”
gamma	“auto”

5.5.4 Logistic Regression

Το Logistic Regression είναι μια στατιστική μέθοδος ταξινόμησης, η οποία χρησιμοποιείται συχνά για πολυκατηγορικά (multiclass) προβλήματα μέσω της one-vs-rest (OvR) στρατηγικής. Σε αυτή την προσέγγιση, το μοντέλο εκπαιδεύεται πολλαπλές φορές, μία για κάθε κλάση, προβλέποντας την πιθανότητα ότι ένα δείγμα ανήκει σε μια συγκεκριμένη κλάση έναντι όλων των άλλων. Τελικά, η κλάση με τη μεγαλύτερη προβλεπόμενη πιθανότητα επιλέγεται ως η τελική πρόβλεψη.

Για την προετοιμασία των δεδομένων, εφαρμόσαμε παρόμοιο preprocessing όπως και στους αλγόριθμους kNN και SVM. Εφαρμόσαμε αποκοπή ακραίων τιμών (1st και 99th percentile cutoff), κανονικοποίηση (Standard Scaler) για να διασφαλίσουμε ότι όλα τα χαρακτηριστικά έχουν συγκρίσιμη κλίμακα, αντικατάσταση (imputation) των κενών τιμών και One-Hot Encoding για τη μετατροπή των κατηγορικών χαρακτηριστικών σε αριθμητικές τιμές. Η συνάρτηση preprocess_data() τροποποιήθηκε ως εξής:

```
X = preprocess_data(X, apply_percentile_cutoff = True, apply_scaling
= True, impute_missing = True, apply_one_hot_encoding = True)
```

Για την επιλογή των υπερπαραμέτρων, χρησιμοποιήσαμε Grid Search, το οποίο δοκίμασε διαφορετικούς συνδυασμούς υπερπαραμέτρων, με στόχο τη βελτιστοποίηση της απόδοσης του Logistic Regression. Οι κύριες υπερπαράμετροι που εξετάστηκαν ήταν οι εξής: Το C (regularization strength) που ρυθμίζει το επίπεδο κανονικοποίησης του μοντέλου. Μια υψηλότερη τιμή μειώνει την επίδραση της κανονικοποίησης, επιτρέποντας στο μοντέλο να προσαρμόζεται περισσότερο στα δεδομένα.. Το solver που ορίζει τον αλγόριθμο που χρησιμοποιείται για την εκπαίδευση του Logistic Regression. Διαφορετικοί solvers είναι πιο κατάλληλοι για συγκεκριμένους τύπους δεδομένων και μεγέθη datasets. Και το max_iter που καθορίζει το μέγιστο αριθμό επαναλήψεων που θα εκτελέσει ο αλγόριθμος μέχρι να συγκλίνει σε λύση. Αυτή η παράμετρος είναι σημαντική για μεγάλα datasets, καθώς μπορεί να επηρεάσει τη χρόνο σύγκλισης.

Μετά την εκτέλεση του Grid Search για LR με 4 κλάσεις (πείραμα 6.1.4), το τελικό Logistic Regression μοντέλο διαμορφώθηκε με τις εξής υπερπαραμέτρους: C = 100, επιτρέπει στο μοντέλο να προσαρμόζεται περισσότερο στα δεδομένα, μειώνοντας τη δύναμη της κανονικοποίησης, solver = "newton-cg", ο Newton Conjugate Gradient solver επιλέχθηκε καθώς είναι αποτελεσματικός για μεγάλα datasets και πολυκατηγορικά προβλήματα και max_iter = 1000, επιτρέπει στον αλγόριθμο περισσότερες επαναλήψεις, διασφαλίζοντας ότι θα συγκλίνει σωστά.

Πίνακας 5.5.4: Υπερπαράμετροι του Logistic Regressor

c	100
solver	“newton-cg”
max_iter	1000

5.5.5 Neural Networks

Στην υλοποίηση των νευρωνικών δικτύων, επιλέξαμε μια πλήρως συνδεδεμένη αρχιτεκτονική εμπρόσθιας τροφοδότησης (fully-connected feedforward neural network). Αυτή η επιλογή βασίστηκε στην ανάγκη εκπαίδευσης ενός ταξινομητή που μπορεί να μάθει περίπλοκες σχέσεις στα δεδομένα εισόδου και να βελτιστοποιήσει την απόδοσή του μέσω των σωστών υπερπαραμέτρων.

Πριν την εκπαίδευση, τα δεδομένα επεξεργάστηκαν κατάλληλα. Τα νευρωνικά δίκτυα δεν μπορούν να χειριστούν κενές τιμές, ούτε κατηγορηματικά χαρακτηριστικά. Παρόλο που μπορούν να προσαρμόσουν τα βάρη τους ώστε να διαχειρίζονται διαφορετικές κλίμακες χαρακτηριστικών, η εφαρμογή κανονικοποίησης (scaling) επιταχύνει τη διαδικασία εκπαίδευσης και βελτιώνει τη σταθερότητα του μοντέλου. Έτσι, εφαρμόσαμε τα πλήρη στάδια προεπεξεργασίας: αποκοπή ακραίων τιμών (1st and 99th percentile cut-off), κανονικοποίηση (scaling), αντικατάσταση κενών τιμών (data imputation) και κωδικοποίηση κατηγορηματικών χαρακτηριστικών (one-hot encoding). Η τροποποιημένη συνάρτηση preprocess_data χρησιμοποιήθηκε ως εξής:

```
X = preprocess_data(X, apply_percentile_cutoff = True, apply_scaling = True, impute_missing = True, apply_one_hot_encoding = True)
```

Η αποτελεσματικότητα των νευρωνικών δικτύων εξαρτάται σε μεγάλο βαθμό από τις σωστές επιλογές υπερπαραμέτρων. Μετά από δοκιμές, η τελική αρχιτεκτονική των μοντέλων μας περιλαμβάνει ένα επίπεδο κρυφών νευρώνων (hidden layer) με 150 νευρώνες, χρησιμοποιώντας τη συνάρτηση ενεργοποίησης ReLU. Ο αλγόριθμος εκπαίδευσης (optimizer) που χρησιμοποιήθηκε είναι ο Adam, ενώ η συνάρτηση ενεργοποίησης εξόδου είναι η softmax, ώστε να παράγονται πιθανότητες για κάθε κλάση. Η συνάρτηση απωλειών (loss function) που χρησιμοποιήθηκε είναι η categorical cross-entropy. Ο ρυθμός εκμάθησης (learning rate) ξεκίνησε από 5×10^{-4} και προσαρμόστηκε δυναμικά με την τεχνική ReduceLROnPlateau, ενώ εφαρμόστηκε dropout-rate 0.5 για την αποφυγή υπερπροσαρμογής (overfitting). Τα μοντέλα εκπαιδεύτηκαν για μέγιστο αριθμό 200 εποχών (epochs) με μέγεθος batch 256. Αναλυτικότερα, οι δοκιμές υπερπαραμέτρων περιγράφονται στο πείραμα **6.1.5 Hyperparameter Tuning για Neural Network με 4 κλάσεις**.

Κατά τη διαδικασία εκπαίδευσης, εισάγαμε callbacks για τη βελτιστοποίηση της απόδοσης. Χρησιμοποιήσαμε το EarlyStopping callback το οποίο διακόπτει την διαδικασία της εκπαιδευσης όταν ικανοποιηθούν συγκεκριμένες συνθήκες. Στόχος αυτού είναι η εξοικονόμηση χρόνου και πόρων ώστε να μην εκτελεστούν όλες οι εποχές που κρίθηκε ότι δεν χρειάστηκαν και σημαντικότερα να μην γίνει overfitting στα training data, όπου το μοντέλο αποδίδει εξαιρετικα σε αυτά αλλα έχει χαμηλή απόδοση σε αόρατα δεδομένα. Επιλέχθηκε να γίνεται monitor το validation accuracy των δεδομένων και όταν αυτό δεν βελτιωθεί κατ'ελάχιστο 10^{-3} σε διάστημα 50 εποχών τότε η εκπαίδευση του μοντέλου σταματά. Ταυτόχρονα εισάγαμε το ReduceLROnPlateau callback το οποίο μειώνει το learning rate κατά την εκπαίδευση όταν η απόδοση δεν βελτιώνεται για έναν συγκεκριμένο αριθμο εποχών. Σκοπός είναι να βελτιωθεί η διαδικασία της εκπαίδευσης επιτρέποντας στο μοντέλο να κάνει μικρότερες προσαρμογές στα βάρη ώστε να γίνονται πιο ακριβείς προσαρμογές. Συγκεκριμένα γίνεται ξανα monitor το validation accuracy και όταν αυτό δεν βελτιωθεί ,σε διάστημα 15 εποχών, κατά 10^{-4} τότε το learning rate μειώνεται κατα παράγονα 10%. Επίσης θέσαμε για το learning rate κατώφλι 10^{-6}. Η εκπαίδευση σταματάει όταν συναντηθεί 1 απο τις 3 συνθήκες: όταν το learning rate ξεπεράσει το κατώφλι 10^{-6}, όταν κληθεί το callback EarlyStopping ή όταν τελειώσουν και οι 200 εποχές. Το τελικό Νευρωνικό Δίκτυο έχει την αρχιτεκτονική που φαίνονται στον πιο κάτω πίνακα 5.5.5:

Πίνακας 5.5.5: Αρχιτεκτονική των Νευρωνικών Δικτύων

Συνάρτηση ενεργοποίησης (activation function)	ReLU (hidden layers) Softmax (Output layer)
Συνάρτηση απωλειών (loss function)	Categorical Cross-entropy
Βελτιστοποιητής (optimizer)	Adam
Ρυθμός εκπαίδευσης (learning rate)	5×10^{-4} with ReduceLROnPlateau
Εποχές (epochs)	200
Νευρώνες (neurons)	150
Batch Size	256
Dropout Rate	0.5

5.5.6 AutoML (Automated Machine Learning)

Για την αυτοματοποιημένη μηχανική μάθηση (AutoML), δοκιμάσαμε δύο διαφορετικά frameworks: AutoGluon και AutoKeras. Αυτά τα δύο εργαλεία διαφοροποιούνται ως προς τη φιλοσοφία και τα εργαλεία που χρησιμοποιούν για την εκπαίδευση των μοντέλων.

Το AutoGluon είναι ένα γενικό πλαίσιο AutoML που ενσωματώνει τόσο παραδοσιακούς αλγορίθμους μηχανικής μάθησης όσο και μοντέλα βαθιάς μάθησης. Χρησιμοποιεί βιβλιοθήκες όπως XGBoost, LightGBM, CatBoost για παραδοσιακά ML μοντέλα και MXNet, PyTorch για deep learning μοντέλα. Είναι σχεδιασμένο για να παρέχει έναν ευέλικτο και εύκολο τρόπο αυτοματοποίησης πολλών διαδικασιών της μηχανικής μάθησης, όπως επιλογή μοντέλου, ρύθμιση υπερπαραμέτρων (hyperparameter tuning) και διαχείριση των χαρακτηριστικών των δεδομένων.

Το AutoKeras, από την άλλη, είναι μια εξειδικευμένη AutoML βιβλιοθήκη χτισμένη πάνω από το TensorFlow και Keras, εστιάζοντας αποκλειστικά στη βελτιστοποίηση και εκπαίδευση νευρωνικών δικτύων. Σε αντίθεση με το AutoGluon, το οποίο χρησιμοποιεί ένα ευρύ φάσμα αλγορίθμων, το AutoKeras επικεντρώνεται στην αναζήτηση βέλτιστης αρχιτεκτονικής νευρωνικών δικτύων (Neural Architecture Search - NAS) και στη βελτιστοποίηση υπερπαραμέτρων αποκλειστικά για deep learning μοντέλα.

Τα δύο frameworks έχουν διαφορετική προσέγγιση στη διαδικασία AutoML, όπως φαίνεται στον παρακάτω πίνακα:

Πίνακας 5.5.6: AutoML Tools

Tool	AutoGluon	AutoKeras
Βασική τεχνολογία	XGBoost, LightGBM, CatBoost, PyTorch, MXNet	TensorFlow, Keras
Τύπος Μοντέλων	ML και Deep Learning	Deep Learning μόνο
Εξειδίκευση	Γενική AutoML πλατφόρμα	Νευρωνικα Δίκτυα (NAS)
Υποστήριξη Παραδοσιακών ML αλγορίθμων	Ναι	Όχι
Υποστήριξη	Όχι	Ναι

Αναζήτησης Αρχιτεκτονικής Νευρωνικών Δικτύων (NAS)		
Βελτιστοποίηση Υπερπαραμέτρων - Hyperparameter Optimization (HPO)	Grid Search, Bayesian Optimization	Neural Architecture Search (NAS)
Συνδυασμένη Επιλογή Αλγορίθμου & Βελτιστοποίηση Υπερπαραμέτρων (CASH)	Ναι, υποστηρίζει επιλογή αλγορίθμων και βελτιστοποίηση	Όχι, μόνο νευρωνικά δίκτυα
Αυτοματοποιημένη Επεξεργασία Δεδομένων (Data Preparation & Feature Engineering)	Ναι, περιλαμβάνει normalization, encoding, feature selection	Όχι, απαιτεί προεπεξεργασία από τον χρήστη
Στοίβαγμα Μοντέλων (Multi-Layer Stack Ensembling)	Ναι, συνδυάζει μοντέλα από διαφορετικούς αλγορίθμους	Όχι, επικεντρώνεται σε ατομικά δίκτυα
Ευκολία Χρήσης	Αυτόματο αλλά απαιτεί ρύθμιση	Πολύ Αυτοματοποιημένο

5.5.6.1 AutoGluon

Το AutoGluon παρέχει μια αυτοματοποιημένη προσέγγιση στη μηχανική μάθηση, εκτελώντας πολλές διαδικασίες που διαφορετικά απαιτούν χειροκίνητη ρύθμιση. Σημαντικό στοιχείο της λειτουργίας του είναι η αυτόματη διάγνωση του τύπου πρόβλεψης, η επεξεργασία δεδομένων, η βελτιστοποίηση υπερπαραμέτρων και η στοίβαξη μοντέλων. Το AutoGluon καθορίζει αυτόματα τον τύπο του προβλήματος βάσει των τιμών στη label στήλη. Στην περίπτωση μας, αναγνωρίστηκε ως multiclass classification, καθώς η στήλη ετικετών περιείχε πολλαπλές διακριτές κατηγορίες. Αν το πρόβλημα ήταν regression, το AutoGluon θα το αναγνώριζε εάν οι τιμές ήταν συνεχείς και δεκαδικές με πολλές μοναδικές τιμές. Για binary ή multiclass classification εξετάζει αν υπάρχουν δύο ή περισσότερες διακριτές κατηγορίες. Επίσης

Το AutoGluon αναγνωρίζει τον τύπο δεδομένων και έχει διαφορετικές μεθόδους επεξεργασίας για το κάθε ένα. Επεξεργάζεται τα δεδομένα με έναν γενικό τρόπο, κατάλληλο

για οποιοδήποτε μοντέλο. Στον πίνακα 5.5.6.1.1 φαίνεται ο τρόπος αναγνώρισης και επεξεργασίας των χαρακτηριστικών.

Πίνακας 5.5.6.1.1: Τύπος Στήλης, Τρόπος ανίχνευσης και επεξεργασία που κάνει το AutoGluon

Τύπος Δεδομένων Στήλης	Αναγνώριση Τύπου Στήλης	Αυτόματη Επεξεργασία Χαρακτηριστικών
Boolean	Οποιαδήποτε στήλη έχει 2 μοναδικές τιμές	Καμία τροποποίηση
Numerical	Στήλες με float ή int. Αυτή τη στιγμή δεν δοκιμάζονται για να καθοριστεί αν είναι πιθανό να είναι κατηγορικές	Integers και float τιμές δεν θα υποστούν κάποια τροποποίηση
Categorical	Οποιαδήποτε στήλη έχει μέσα string (εκτός και αν είναι text). Κάποια μοντέλα αποδίδουν καλύτερα όταν ορίζεται ποιές στήλες είναι categorical και ποιες continuous	Δεδομένου ότι πολλά μοντέλα απαιτούν οι κατηγορίες να κωδικοποιούνται ως ακέραιοι αριθμοί, κάθε κατηγορικό χαρακτηριστικό αντιστοιχεί σε μονοτονικά αυξανόμενους ακέραιους αριθμούς.
datetime	Αναγνωρίζονται προσπαθώντας να μετατραπούν σε Pandas datetime. Το Pandas αναγνωρίζει μια μεγάλη ποικιλία απο datetime μορφές. Αν αρκετές από τις τιμές στη στήλη μετατραπούν επιτυχώς σε datetime, τότε ο τύπος της στήλης ορίζεται ως datetime	Datetime μετατρέπονται σε Pandas datetime και σε στήλες [year, month, day, dayofweek]. Κενές, μη έγκυρες και εκτός ορίων τιμές μετατρέπονται στην μέση τιμή των έγκυρων τιμών.

Για την προετοιμασία των δεδομένων πριν από την εκπαίδευση, εκτελέσαμε μόνο το 1st και 99th percentile cut-off. Στη συνέχεια, το AutoGluon ανέλαβε να εφαρμόσει μετασχηματισμούς δεδομένων μέσω των Generators, τα οποία είναι προκαθορισμένα στάδια που μετατρέπουν τα αρχικά δεδομένα σε πιο κατάλληλα χαρακτηριστικά για την εκπαίδευση των μοντέλων. Οι Generators που χρησιμοποιήθηκαν φαίνονται στον Πίνακα 5.5.6.1.2.

Πίνακας 5.5.6.1.2: Generators και χρήσεις

Stage	Generator	Functionality

1	AsTypeFeatureGenerator	Επιβάλλει τη μετατροπή τύπων στα δεδομένα, ώστε να ταιριάζουν με τους τύπους που παρατηρήθηκαν κατά την εκπαίδευση.
2	FillNaFeatureGenerato	Το οποίο συμπληρώνει κενές τιμές στα δεδομένα.
3	IdentityFeatureGenerator	Προωθεί τα δεδομένα χωρίς τροποποιήσεις.
	CategoryFeatureGenerator	Μετατρέπει χαρακτηριστικά τύπου object (όπως strings) σε τύπους category. Αυτή η μετατροπή βελτιστοποιεί τη χρήση μνήμης και ενισχύει την απόδοση του μοντέλου κατά την εκπαίδευση. Επιπλέον, μπορεί να αφαιρεί σπάνιες κατηγορίες για να αποτρέψει την υπερεκπαίδευση και να βελτιώσει την απόδοση του μοντέλου.
	CategoryMemoryMinimize FeatureGenerator	Ελαχιστοποιεί τη χρήση μνήμης των κατηγορικών χαρακτηριστικών μετατρέποντας τις τιμές κατηγορίας σε μονοτονικά αυξανόμενες ακέραιες τιμές.
4	DropUniqueFeatureGenera tor	Αφαιρεί χαρακτηριστικά τα οποία έχουν μόνο 1 μοναδική τιμή ή σχεδόν καμία επαναλαμβανόμενη τιμή (βάσει του max_unique_ratio) και είναι τύπου κατηγορίας ή αντικειμένου.
5	DropDuplicatesFeatureGen erator	Αφαιρεί χαρακτηριστικά που είναι ακριβή αντίγραφα άλλων χαρακτηριστικών, διατηρώντας μόνο ένα από αυτά.

Το AutoGluon επιτρέπει στους χρήστες να ρυθμίσουν χειροκίνητα τις υπερπαραμέτρους, είτε ορίζοντας σταθερές τιμές είτε καθορίζοντας εύρος τιμών για αναζήτηση. Εάν δεν δοθούν χειροκίνητες τιμές, τότε οι υπερπαράμετροι καθορίζονται από το AutoGluon, βασιζόμενες σε προκαθορισμένα εύρη. Για τα νευρωνικά δίκτυα, δοκιμάστηκαν υπερπαράμετροι με αριθμό εποχών από 10 μέχρι 200, learning rate από 0.01 μέχρι 0.0005, activation functions ReLU, SoftReLU και Tanh, καθώς και dropout probability από 0.0 μέχρι 0.5. Για τα LightGBM

gradient boosted trees, το number of boosting rounds ορίστηκε σταθερά σε 100, ενώ το number of leaves είχε εύρος αναζήτησης από 10 μέχρι 50.

Για να επιταχύνουμε τη διαδικασία εκπαίδευσης, ορίσαμε συνολικό χρονικό όριο 20 λεπτών, ενώ επιτρέψαμε 20 διαφορετικούς συνδυασμούς υπερπαραμέτρων ανά μοντέλο. Η στρατηγική αναζήτησης υπερπαραμέτρων ορίστηκε ως auto, ενώ η μετρική αξιολόγησης που χρησιμοποιήθηκε ήταν το accuracy.

Για να αξιοποιηθεί η τεχνική του Model Ensembling μέσω stacking και bagging, θέσαμε την auto_stack παράμετρο σε True κατά το fit(). Αυτό οδήγησε στην αυτόματη ρύθμιση των επιπέδων stacking και των διπλών στο bagging. Ειδικότερα, επιλέχθηκε num_stack_levels ίσο με 1, num_bag_folds ίσο με 8 και num_bag_sets ίσο με 5. Το bagging χρησιμοποιήθηκε για να βελτιώσει τη γενίκευση του μοντέλου, ενώ το stacking επέτρεψε τον συνδυασμό μοντέλων από διαφορετικούς αλγορίθμους, αυξάνοντας την ακρίβεια της πρόβλεψης. Αυτή η διαδικασία αύξησε σημαντικά τον χρόνο εκπαίδευσης, αλλά οδήγησε σε καλύτερα αποτελέσματα.

Κατά τη χειροκίνητη τροποποίηση των υπερπαραμέτρων, το AutoGluon εκπαιδεύει λιγότερα μοντέλα, σε σύγκριση με την εκπαίδευση χωρίς ορισμό των υπερπαραμέτρων. Αυτό συμβαίνει επειδή εκπαιδεύονται μόνο μοντέλα που πληρούν τις προϋποθέσεις που τέθηκαν, ενώ μοντέλα με υπερπαραμέτρους εκτός του εύρους αναζήτησης παραλείπονται. Παρόλο που ο χώρος αναζήτησης που τέθηκε χειροκίνητα ήταν μεγάλος, παρατηρήθηκε μείωση των δοκιμασμένων μοντέλων. Συνεπώς, στην τελική εκπαίδευση αποφασίστηκε να μην εφαρμοστούν χειροκίνητες ρυθμίσεις υπερπαραμέτρων, επιτρέποντας στο AutoGluon να καθορίσει αυτόματα τις βέλτιστες τιμές.

5.5.6.2 AutoKeras

Το AutoKeras χρησιμοποιεί νευρωνικά δίκτυα ως τη βάση για τα machine learning μοντέλα που δημιουργεί, ακολουθώντας μια αυτοματοποιημένη διαδικασία εύρεσης της βέλτιστης αρχιτεκτονικής μέσω του Neural Architecture Search (NAS). Για να διασφαλίσουμε ότι τα μοντέλα θα εκπαιδευτούν αποδοτικά από το dataset, πραγματοποιήσαμε όλα τα βήματα της προεπεξεργασίας των δεδομένων, όπως και στην προηγούμενη υλοποίηση των νευρωνικών δικτύων που κατασκευάσαμε χειροκίνητα.

Για την εκπαίδευση, αρχικοποιήσαμε το AutoKeras Structured Data Classifier, καθορίζοντας τον μέγιστο αριθμό διαφορετικών μοντέλων που θα δοκιμάσει κατά τη διαδικασία εύρεσης της βέλτιστης αρχιτεκτονικής σε 25 max_trials, ενώ ο αριθμός των κλάσεων στην έξοδο (num_classes) ορίστηκε σε 4, ώστε να ταιριάζει με τις ετικέτες ταξινόμησης του προβλήματος. Στη διαδικασία fitting, ορίσαμε αριθμό εποχών 100, ώστε το AutoKeras να έχει επαρκή χρόνο εκπαίδευσης για κάθε προτεινόμενο μοντέλο.

Κατά τη διάρκεια της αναζήτησης της αρχιτεκτονικής και της εκπαίδευσης, το AutoKeras εξερευνά διαφορετικούς συνδυασμούς υπερπαραμέτρων και αρχιτεκτονικών για να βελτιστοποιήσει την απόδοση του μοντέλου. Το NAS καθορίζει ένα σύνολο πιθανών αρχιτεκτονικών που περιλαμβάνουν παραμέτρους όπως ο αριθμός επιπέδων, ο αριθμός νευρώνων ανά επίπεδο, οι τύποι ενεργοποιήσεων και οι συνδέσεις μεταξύ των νευρώνων.

Στον Πίνακα 5.5.6.2 παρουσιάζεται ένα συγκεκριμένο trial κατά τη διαδικασία fitting. Σε αυτή τη δοκιμή, οι υπερπαράμετροι που επιλέχθηκαν οδήγησαν σε ακρίβεια 62.16% στα validation δεδομένα, η οποία δεν ήταν η καλύτερη μέχρι εκείνη τη στιγμή, καθώς το βέλτιστο αποτέλεσμα σε προηγούμενο trial ήταν 62.70%. Το AutoKeras συνέχισε την εξερεύνηση των υπερπαραμέτρων, αναζητώντας καλύτερους συνδυασμούς μέσω επιπλέον trials.

```
Search: Running Trial #12

Value             |Best Value So Far |Hyperparameter
True              |True              |structured_data_block_1/normalize
False             |False             |structured_data_block_1/dense_block_1/use_batchnorm
2                 |2                 |structured_data_block_1/dense_block_1/num_layers
128               |32                |structured_data_block_1/dense_block_1/units_0
0                 |0                 |structured_data_block_1/dense_block_1/dropout
64                |64                |structured_data_block_1/dense_block_1/units_1
0                 |0                 |classification_head_1/dropout
adam              |adam              |optimizer
0.001             |0.001             |learning_rate

Epoch 1/50
372/372 [==============================] - 2s 3ms/step - loss: 0.9090 - accuracy: 0.5835 - val_loss: 89.7595 - val_accuracy: 0.6117
Epoch 2/50
372/372 [==============================] - 1s 2ms/step - loss: 0.8319 - accuracy: 0.6195 - val_loss: 128.2788 - val_accuracy: 0.6151
Epoch 3/50
372/372 [==============================] - 1s 2ms/step - loss: 0.8127 - accuracy: 0.6283 - val_loss: 169.3429 - val_accuracy: 0.6165
Epoch 4/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7991 - accuracy: 0.6354 - val_loss: 167.2715 - val_accuracy: 0.6141
Epoch 5/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7876 - accuracy: 0.6383 - val_loss: 208.7925 - val_accuracy: 0.6134
Epoch 6/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7782 - accuracy: 0.6433 - val_loss: 258.7956 - val_accuracy: 0.6158
Epoch 7/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7699 - accuracy: 0.6475 - val_loss: 266.0924 - val_accuracy: 0.6175
Epoch 8/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7621 - accuracy: 0.6515 - val_loss: 261.4309 - val_accuracy: 0.6192
Epoch 9/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7554 - accuracy: 0.6552 - val_loss: 252.0209 - val_accuracy: 0.6206
Epoch 10/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7483 - accuracy: 0.6573 - val_loss: 247.4962 - val_accuracy: 0.6182
Epoch 11/50
372/372 [==============================] - 1s 2ms/step - loss: 0.7424 - accuracy: 0.6594 - val_loss: 245.3779 - val_accuracy: 0.6216
Trial 12 Complete [00h 00m 22s]
val_accuracy: 0.6215847134590149

Best val_accuracy So Far: 0.6270492076873779
Total elapsed time: 00h 05m 04s
```

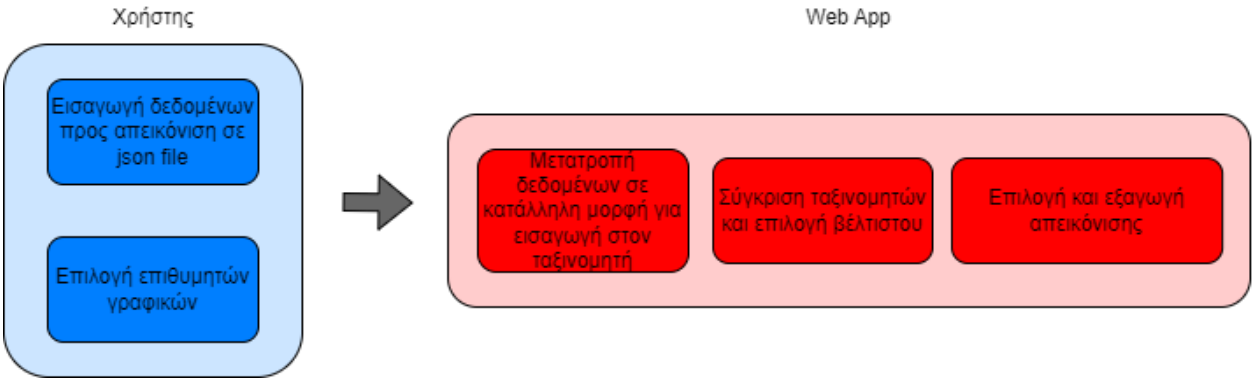
Εικόνα 5.5.6.2: #12 Trial από AutoKeras κατά την διαδικασία fitting

5.6 *Ανάπτυξη εφαρμογής - Web App - Demo*

Η ολοκλήρωση της διαδικασίας εκπαίδευσης των μοντέλων ακολουθείται από την ενσωμάτωσή τους στην Web εφαρμογή, όπου αξιοποιούνται για την παροχή προτάσεων στον χρήστη σχετικά με την ιδανικότερη οπτικοποίηση των δεδομένων του. Κατά την ανάπτυξη της μεθόδου, όπως περιγράφηκε στην ενότητα 4.1, αρχικά

προτάθηκε η χρήση ταξινομητών για την αυτόματη επιλογή γραφικής αναπαράστασης. Ωστόσο, κρίθηκε απαραίτητο να επεκταθεί αυτή η προσέγγιση με την ανάπτυξη μιας Web εφαρμογής, ώστε το σύστημα να είναι λειτουργικό και προσβάσιμο από τον χρήστη σε πραγματικό χρόνο. Χωρίς την ενσωμάτωση των ταξινομητών σε ένα διαδραστικό περιβάλλον, η χρήση τους θα παρέμενε θεωρητική, χωρίς πρακτική εφαρμογή.

Στην τελική έκδοση της εφαρμογής, χρησιμοποιήθηκαν τα μοντέλα που εκπαιδεύτηκαν με το AutoGluon, καθώς αυτά παρουσίασαν τις καλύτερες επιδόσεις, όπως θα δούμε αναλυτικά στην ενότητα 6: Αξιολόγηση. Για την ανάπτυξη της Web εφαρμογής επιλέχθηκε το Streamlit, ένα εύχρηστο framework που επιτρέπει τη δημιουργία διαδραστικών εφαρμογών με ελάχιστο κώδικα. Η εφαρμογή παρέχει ένα φιλικό προς τον χρήστη περιβάλλον, μέσω του οποίου μπορεί να πειραματιστεί με τα δικά του δεδομένα και να λάβει συστάσεις από τα εκπαιδευμένα μοντέλα. Η συνολική μεθοδολογία που ακολουθήθηκε στην ανάπτυξη της εφαρμογής αποτυπώνεται στην Εικόνα 5.6.1.



Εικόνα 5.6.1: Workflow Web εφαρμογής

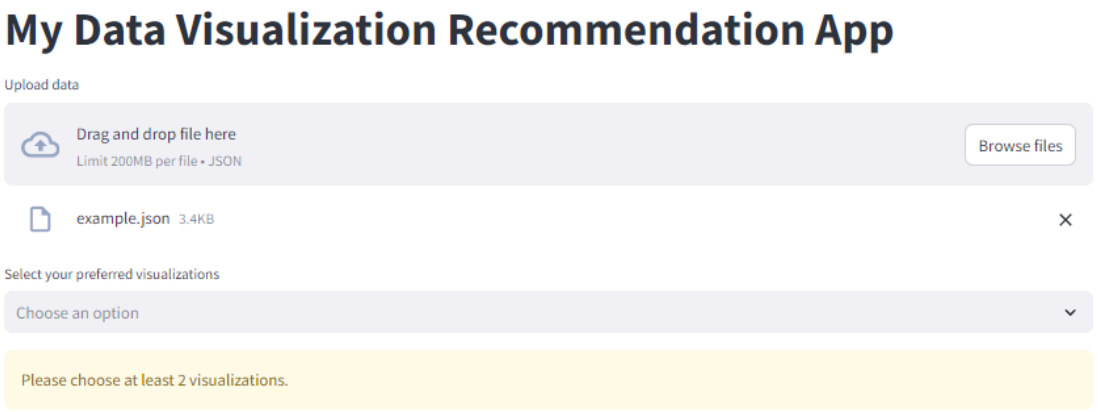
Η διαδικασία ξεκινά με τον χρήστη να εισάγει τα δεδομένα του στην εφαρμογή, επιλέγοντας την κατάλληλη μορφή. Στη συνέχεια, αποφασίζει ποιο ταξινομητή επιθυμεί να χρησιμοποιήσει, επιλέγοντας μεταξύ των διαθέσιμων ετικετών “bar”, “line”, “scatter” και “pie”. Από το σημείο αυτό, η εφαρμογή αναλαμβάνει αυτόματα την προετοιμασία των δεδομένων, την επιλογή του κατάλληλου μοντέλου και την απεικόνιση των αποτελεσμάτων.

Εισαγωγή Δεδομένων:

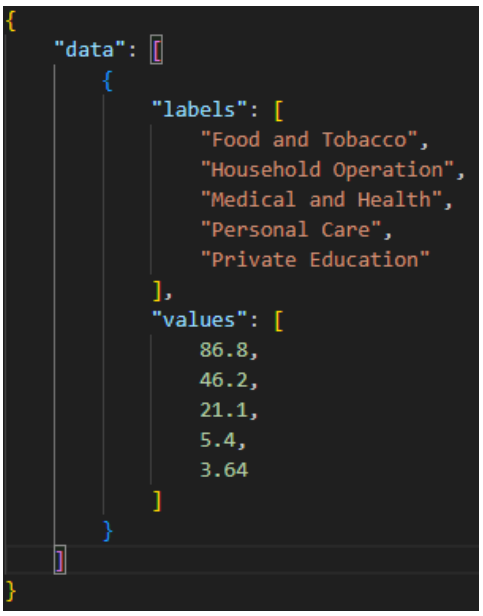
Η πρώτη φάση περιλαμβάνει την εισαγωγή των δεδομένων στην εφαρμογή μέσω drag and drop (εικόνα 5.6.2) ή επιλέγοντας αρχεία από το σύστημα του χρήστη. Για να διασφαλιστεί η ορθότητα της διαδικασίας, τα δεδομένα πρέπει να είναι κατάλληλα δομημένα. Επιλέξαμε ως

μορφή εισόδου αρχεία CSV ή JSON, καθώς αυτά επιτρέπουν την αποθήκευση δομημένων δεδομένων που μπορούν εύκολα να χρησιμοποιηθούν για την εξαγωγή χαρακτηριστικών.

Κάθε JSON file που εισάγεται στην εφαρμογή περιέχει ένα top-level key “data”, το οποίο περιλαμβάνει τις στήλες ‘x’ και ‘y’ με τα δεδομένα που θα απεικονιστούν. Αν τα δεδομένα είναι σε μορφή κειμένου, πρέπει να είναι σε string format (εισαγωγικά), ενώ αν είναι αριθμητικά, να αποθηκεύονται ως integers ή floats χωρίς εισαγωγικά. Στην Εικόνα 5.6.3, φαίνεται ένα παράδειγμα αποδεκτής δομής ενός JSON αρχείου.



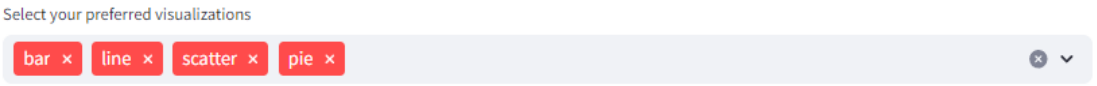
Εικόνα 5.6.2: Είσοδος δεδομένων



Εικόνα 5.6.3: Μορφή json file

Επιλογή επιθυμητών Οπτικοποιήσεων

Αφού τα δεδομένα εισαχθούν στην εφαρμογή, παρέχεται στον χρήστη η δυνατότητα προσαρμογής του ταξινομητή στις δικές του ανάγκες. Έχουν εκπαιδευτεί μοντέλα για όλους τους πιθανούς συνδυασμούς ετικετών με εξόδους 2, 3 ή 4 κλάσεων (bar, line, scatter, pie). Αυτό επιτρέπει στον χρήστη να πειραματιστεί και να συγκρίνει μεταξύ τους διαφορετικές επιλογές, ώστε να επιλέξει το καταλληλότερο μοντέλο για τα δεδομένα του. Στην Εικόνα 5.6.4, παρουσιάζεται το περιβάλλον της εφαρμογής κατά την επιλογή της οπτικοποίησης.



Εικόνα 5.6.4: Επιλογή οπτικοποίησης

Σύγκριση Μοντέλων

Αφού ο χρήστης επιλέξει τις οπτικοποιήσεις που τον ενδιαφέρουν, η εφαρμογή αναζητά τον ταξινομητή που ταιριάζει καλύτερα στις επιλεγμένες ετικέτες. Το μοντέλο που θα χρησιμοποιηθεί επιλέγεται βάσει του υψηλότερου validation accuracy, όπως αυτό υπολογίστηκε στην ενότητα 6: Αξιολόγηση. Για παράδειγμα, αν ο χρήστης επιλέξει και τις τέσσερις ετικέτες, το AutoGluon με 4 κλάσεις επιλέγεται ως ο ταξινομητής, καθώς έχει την υψηλότερη ακρίβεια μεταξύ των διαθέσιμων μοντέλων. Η επιλογή του μοντέλου είναι κρίσιμη, καθώς κάθε ταξινομητής απαιτεί διαφορετική προεπεξεργασία των δεδομένων πριν την εισαγωγή τους για πρόβλεψη.

Μετατροπή Δεδομένων

Μετά την επιλογή του μοντέλου, ακολουθεί η κατάλληλη προετοιμασία των δεδομένων πριν αυτά χρησιμοποιηθούν για πρόβλεψη. Κάθε μοντέλο έχει συγκεκριμένες απαιτήσεις για τα δεδομένα εισόδου. Για τη διασφάλιση της συμβατότητας, η εφαρμογή επιστρέφει στον χρήστη τα αρχικά δεδομένα και τα επεξεργασμένα χαρακτηριστικά που εισάγονται στο μοντέλο. Οι απαραίτητες μετατροπές γίνονται με βάση παραμέτρους που αποθηκεύσαμε από τη διαδικασία εκπαίδευσης. Αυτές είναι το 1st and 99th percentile, ο μέσος όρος (mean) των numerical χαρακτηριστικών για την αντικατάσταση κενών τιμών, το mode των categorical χαρακτηριστικών για συμπλήρωση ελλιπών δεδομένων και ο μετασχηματισμός των κατηγορηματικών χαρακτηριστικών σε αριθμητικές τιμές μέσω mapping. Για παράδειγμα, αν επιλεγεί το AutoGluon ως ταξινομητής για 4 κλάσεις, η μόνη προεπεξεργασία που εκτελείται είναι η αποκοπή ακραίων τιμών (percentile cut-off), καθώς το AutoGluon αναλαμβάνει αυτόματα τις υπόλοιπες μετατροπές. Στην Εικόνα 5.6.5, παρουσιάζονται τα δεδομένα εισόδου, τα αρχικά χαρακτηριστικά και οι αντίστοιχες τροποποιήσεις τους.

Input:

	x_values	y_values
0	Food and Tobac	86.8
1	Household Ope	46.2
2	Medical and He	21.1
3	Personal Care	5.4
4	Private Educatic	3.64

Features:

	X_Type	Y_Type	Num_Entries_X	Num_Entries_Y	Normalized Range (X)	Normalized Range (Y)	Num_Unique_X	Num_Unique_Y	Has_None_X	Num_None_X	Perce
0	c	q	5	5	None	2.5487	5	5	☐	0	

Processed Features:

	Num_Entries_X	Num_Entries_Y	Normalized Range (Y)	Num_Unique_X	Num_Unique_Y	Num_None_X	Percentage_None_X	Num_None_Y	Percentage_None_Y	No
0	5	5	2.5487	5	5	0	0	0	0	

Εικόνα 5.6.5: Δεδομένα εισόδου σε 'X' και 'Y' στήλες, αρχικά χαρακτηριστικά και τροποποιημένα χαρακτηριστικά

Απεικόνιση Δεδομένων

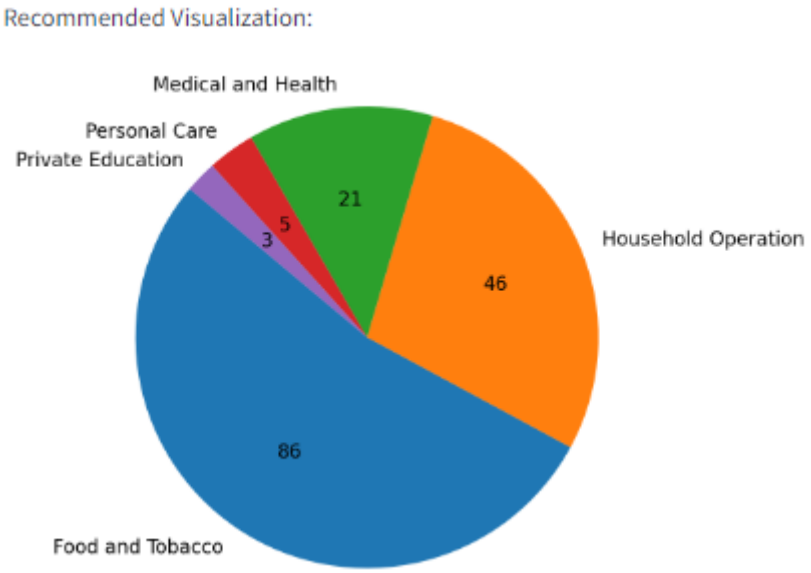
Στο τελευταίο στάδιο, τα επεξεργασμένα δεδομένα τροφοδοτούνται στο μοντέλο και πραγματοποιείται η πρόβλεψη της καταλληλότερης οπτικοποίησης. Το μοντέλο υπολογίζει τις πιθανότητες για κάθε κατηγορία και επιλέγει την ετικέτα με τη μεγαλύτερη εκτίμηση. Στην Εικόνα 5.6.6, παρουσιάζονται τα scores που επέστρεψε το μοντέλο για κάθε πιθανή κατηγορία. Στο συγκεκριμένο παράδειγμα, η ετικέτα "pie" είχε τη μεγαλύτερη πιθανότητα, επομένως το σύστημα πρότεινε την απεικόνιση των δεδομένων με μια pie chart. Στην Εικόνα 5.6.7, φαίνεται η τελική οπτικοποίηση των δεδομένων, όπως προτάθηκε από το μοντέλο.

Η ανάπτυξη της Web εφαρμογής ενσωματώνει όλα τα βήματα της διαδικασίας AutoML, επιτρέποντας στον χρήστη να αλληλεπιδράσει άμεσα με τα δεδομένα του και να λάβει προτάσεις για την ιδανικότερη γραφική αναπαράστασή τους.

Predicted Probabilities:

	Probability
bar	0.28403589129447937
line	0.02057832106947899
pie	0.6670913100242615
scatter	0.028294501826167107

Εικόνα 5.6.6: Scores από ταξινομητή



Εικόνα 5.6.7: Οπτικοποίηση όπως έχει προταθεί από το μοντέλο

6 *Αξιολόγηση*

Η διαδικασία αξιολόγησης είναι καθοριστική για την κατανόηση της απόδοσης των μοντέλων μας και την εξαγωγή χρήσιμων συμπερασμάτων για τη βελτίωσή τους. Ο στόχος της αξιολόγησης είναι να συγκρίνουμε διαφορετικές προσεγγίσεις ταξινόμησης, να εντοπίσουμε τις ισχυρότερες μεθόδους και να κατανοήσουμε ποια χαρακτηριστικά επηρεάζουν περισσότερο την τελική απόδοση. Τα πειράματα που ακολουθούν έχουν σχεδιαστεί για να απαντήσουν σε κρίσιμα ερωτήματα σχετικά με την επιλογή υπερπαραμέτρων, τη σημασία των χαρακτηριστικών, τη σύγκριση κλασικών αλγορίθμων ταξινόμησης, καθώς και την αποτελεσματικότητα των AutoML εργαλείων.

Αρχικά, στο Πείραμα 1 αναζητούμε τις βέλτιστες υπερπαραμέτρους για κάθε μοντέλο μέσω διαδικασίας εξαντλητικής αναζήτησης (Grid Search). Αυτό μας επιτρέπει να δημιουργήσουμε ισχυρότερα και πιο αξιόπιστα μοντέλα πριν προχωρήσουμε στην περαιτέρω αξιολόγηση τους. Στη συνέχεια, στο Πείραμα 2, αναλύουμε τη σημασία των χαρακτηριστικών χρησιμοποιώντας τη μέθοδο Feature Importance ενός Random Forest ταξινομητή, προκειμένου να εντοπίσουμε ποια χαρακτηριστικά έχουν τη μεγαλύτερη επίδραση στην απόδοση των μοντέλων μας.

Στο Πείραμα 3, συγκρίνουμε κλασικές μεθόδους μοντελοποίησης (Random Forest, k-NN, SVM, Logistic Regression) για να διαπιστώσουμε ποια είναι η πιο αποτελεσματική για το συγκεκριμένο πρόβλημα ταξινόμησης γραφικών παραστάσεων. Κατόπιν, στο Πείραμα 4, επεκτείνουμε τη σύγκριση, ενσωματώνοντας πιο προηγμένες προσεγγίσεις, όπως Νευρωνικά Δίκτυα και AutoML frameworks (AutoGluon, AutoKeras), για να δούμε αν τα αυτοματοποιημένα εργαλεία επιτυγχάνουν καλύτερες επιδόσεις σε σχέση με τις χειροκίνητα ρυθμισμένες προσεγγίσεις.

Τέλος, στην ενότητα 6.5, συνοψίζουμε τα αποτελέσματα των πειραμάτων, εστιάζοντας στα συμπεράσματα που προκύπτουν σχετικά με την αποδοτικότητα των μοντέλων και τις βέλτιστες πρακτικές που εντοπίσαμε.

6.1 Πείραμα 1: Πειράματα για εύρεση βέλτιστων υπερ παραμέτρων

Ένα machine learning μοντέλο ορίζεται ως ένα μαθηματικό μοντέλο με διάφορους παραμέτρους που μαθαίνει από τα δεδομένα. Κατά την εκπαίδευση, το μοντέλο προσαρμόζει τις εσωτερικές του παραμέτρους, ωστόσο υπάρχουν υπερπαράμετροι που δεν προσαρμόζονται αυτόματα αλλά πρέπει να καθοριστούν πριν την εκπαίδευση. Οι υπερπαράμετροι αυτές καθορίζουν βασικές ιδιότητες του μοντέλου, όπως η πολυπλοκότητά του και η ταχύτητα εκμάθησης.

Για την εύρεση των βέλτιστων υπερπαραμέτρων, χρησιμοποιήσαμε τον ταξινομητή ‘lsbp’, και στη συνέχεια εφαρμόσαμε την καλύτερη αρχιτεκτονική που προέκυψε και στους υπόλοιπους ταξινομητές. Ο λόγος που επιλέξαμε το ‘lsbp’ είναι επειδή περιλαμβάνει και τις τέσσερις κατηγορίες γραφημάτων που εξετάζουμε (line, scatter, bar, pie), επιτρέποντας τη βελτιστοποίηση των υπερπαραμέτρων σε ένα πλήρες και αντιπροσωπευτικό σύνολο δεδομένων.

Για τα scikit-learn μοντέλα, χρησιμοποιήθηκε η μέθοδος GridSearchCV για την επιλογή των υπερπαραμέτρων. Η Grid Search είναι μια διαδικασία εξαντλητικής αναζήτησης όπου ο χρήστης καθορίζει ένα πλέγμα (grid) τιμών για κάθε υπερπαράμετρο και δοκιμάζονται όλοι οι δυνατοί συνδυασμοί. Στο τέλος, επιλέγεται το καλύτερο μοντέλο με βάση τη μετρική αξιολόγησης που καθορίσαμε – στην περίπτωσή μας, το validation accuracy.

Η διαδικασία της βελτιστοποίησης των υπερπαραμέτρων μέσω Grid Search περιγράφεται αναλυτικά στα ακόλουθα υποκεφάλαια για κάθε μοντέλο ξεχωριστά:

- Grid Search για Random Forest με 4 κλάσεις (Ενότητα 6.1.1)
- Grid Search για KNN με 4 κλάσεις (Ενότητα 6.1.2)
- Grid Search για SVM με 4 κλάσεις (Ενότητα 6.1.3)
- Grid Search για Logistic Regression με 4 κλάσεις (Ενότητα 6.1.4)

Για τα νευρωνικά δίκτυα (Neural Networks), που αναπτύχθηκαν με τη βιβλιοθήκη Keras, δεν χρησιμοποιήσαμε κάποια έτοιμη συνάρτηση βελτιστοποίησης των υπερπαραμέτρων, όπως το GridSearchCV. Αντίθετα, εκτελέσαμε χειροκίνητες δοκιμές (manual tuning) σε διαφορετικούς συνδυασμούς παραμέτρων, προκειμένου να προσδιορίσουμε το βέλτιστο μοντέλο. Οι δοκιμές αυτές αναλύονται στην ενότητα 6.1.5 Hyperparameter Tuning για Neural Network με 4 κλάσεις.

Τέλος, τα AutoML μοντέλα (AutoGluon, AutoKeras) εκτελούν αυτόματα τη διαδικασία προσαρμογής των υπερπαραμέτρων και δεν απαιτούν παρεμβάσεις από εμάς. Οι μέθοδοι AutoML αναλαμβάνουν την επιλογή της βέλτιστης αρχιτεκτονικής, των παραμέτρων και των μοντέλων, με βάση τα δεδομένα εκπαίδευσης, καθιστώντας τη διαδικασία βελτιστοποίησης πλήρως αυτοματοποιημένη.

Οι τιμές των υπερπαραμέτρων που προέκυψαν από το πείραμα αυτό εφαρμόστηκαν στα τελικά μοντέλα που χρησιμοποιήθηκαν στη Web εφαρμογή, όπως περιγράφεται στο Κεφάλαιο 5.6.

6.1.1 Grid Search για Random Forest με 4 κλάσεις

Στην παρακάτω ανάλυση εφαρμόστηκε η μέθοδος Grid Search για τη βελτιστοποίηση των υπερπαραμέτρων του μοντέλου Random Forest σε ένα πρόβλημα ταξινόμησης με τέσσερις κλάσεις. Οι υπερπαράμετροι που εξετάστηκαν, καθώς και οι τιμές που δοκιμάστηκαν, παρουσιάζονται στον Πίνακα 6.1.1.1.

Πίνακας 6.1.1.1: Υπερ παράμετροι και τιμές που δοκιμάστηκαν με την χρήση του GridSearch για το Random Forest

n_estimators	100,200,300
max_depth	10,20,30
min_samples_split	2,5,10
max_features	‘sqrt’, ‘log2’
criterion	‘gini’, ‘entropy’

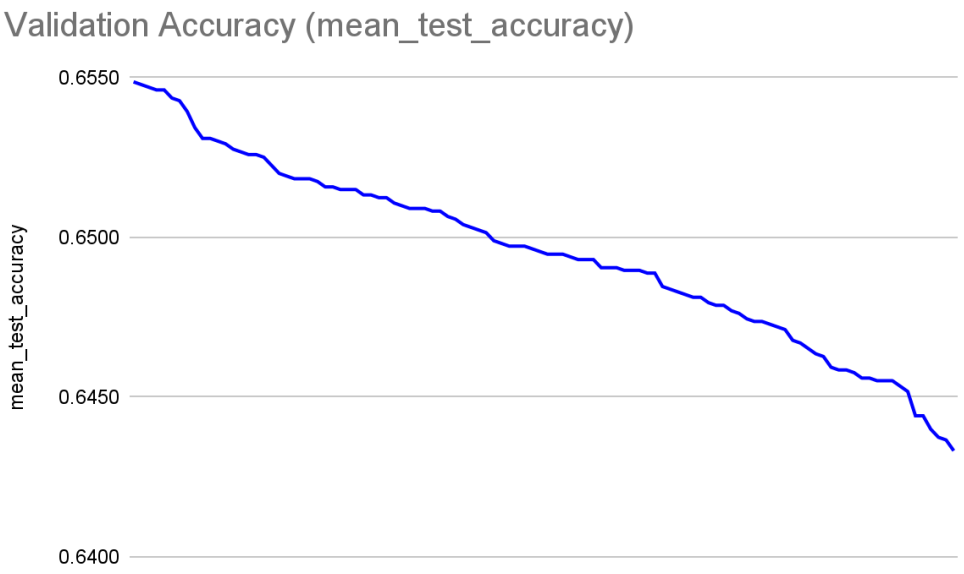
Μετά την εκτέλεση του Grid Search, πραγματοποιήθηκαν συνολικά **108 πειράματα**. Από αυτά, αναλύονται τα 10 καλύτερα αποτελέσματα, τα οποία επιλέχθηκαν με βάση το validation accuracy. Η βέλτιστη αρχιτεκτονική για το μοντέλο Random Forest προέκυψε με τις εξής υπερπαραμέτρους: criterion = ‘gini’, max_depth = 20, max_features = ‘sqrt’, min_sample_split = 2 και n_estimators = 200. Ο Πίνακας 6.1.1.2 παρουσιάζει τους δέκα καλύτερους συνδυασμούς υπερπαραμέτρων, ταξινομημένους με βάση το validation accuracy (mean_test_accuracy).

Πίνακας 6.1.1.2: Οι top 10 συνδιασμοί υπερ παράμετρων ταξινομημένοι με βάση το validation accuracy για το Random Forest

param_criterion	param_max_depth	param_max_features	param_min_samples_split	param_n_estimators	mean_train_accuracy	mean_test_accuracy	mean_train_loss	mean_test_loss
gini	20	sqrt	2	200	0.9655	0.6549	0.0345	0.3451
gini	20	sqrt	2	300	0.9655	0.6548	0.0345	0.3452

gini	30	log2	2	300	0.9679	0.6547	0.0321	0.3453
entropy	20	log2	2	300	0.9659	0.6546	0.0341	0.3454
gini	30	sqrt	2	300	0.9679	0.6546	0.0321	0.3454
entropy	30	sqrt	2	300	0.9679	0.6544	0.0321	0.3456
entropy	30	log2	2	300	0.9679	0.6543	0.0321	0.3457
gini	30	sqrt	2	200	0.9679	0.6539	0.0321	0.3461
gini	30	log2	2	200	0.9679	0.6534	0.0321	0.3466
entropy	20	log2	2	200	0.9660	0.6531	0.0340	0.3469

Επιπλέον, η Εικόνα 6.1.1 απεικονίζει τη συνολική απόδοση του μοντέλου για όλους τους συνδυασμούς υπερπαραμέτρων που δοκιμάστηκαν. Η ανάλυση αυτών των αποτελεσμάτων επιβεβαιώνει ότι η απόδοση του Random Forest μπορεί να βελτιστοποιηθεί σημαντικά μέσω της προσεκτικής επιλογής των υπερπαραμέτρων.



Εικόνα 6.1.1: Validation Accuracy για όλα τα πειράματα RF

6.1.2 Grid Search για KNN με 4 κλάσεις

Στην παρακάτω ανάλυση εφαρμόστηκε η μέθοδος Grid Search για τη βελτιστοποίηση των υπερπαραμέτρων του μοντέλου k-Nearest Neighbors (KNN) σε ένα πρόβλημα ταξινόμησης με τέσσερις κλάσεις. Οι υπερπαράμετροι που εξετάστηκαν, καθώς και οι αντίστοιχες τιμές που δοκιμάστηκαν, παρουσιάζονται στον Πίνακα 6.1.2.1.

Πίνακας 6.1.2.1: Υπερ παράμετροι και τιμές που δοκιμάστηκαν με την χρήση του GridSearch για το KNN

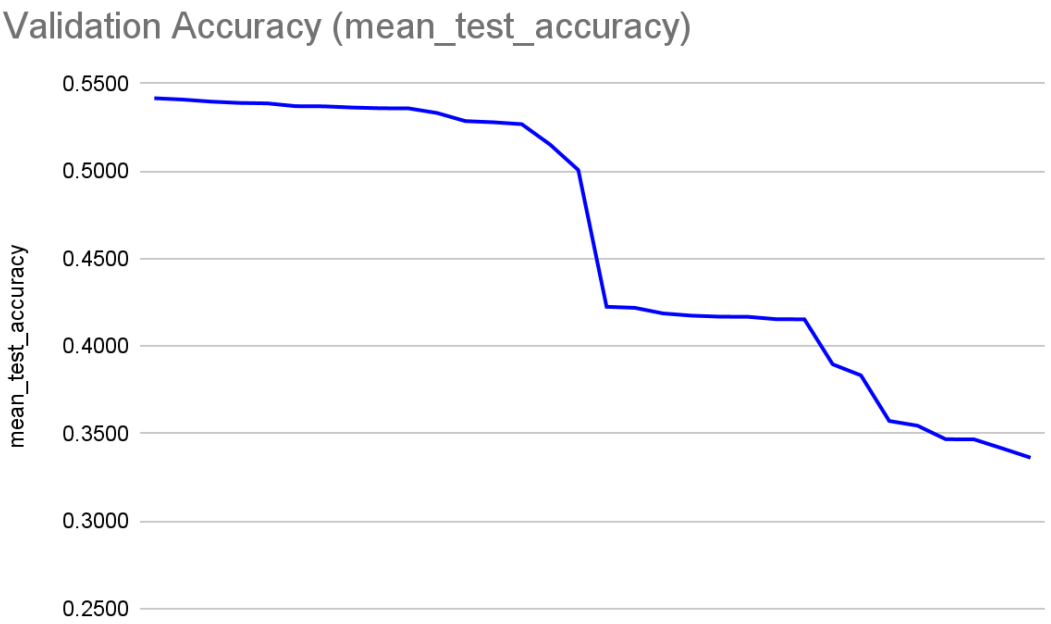
n_neighbors	2,3,4,5,6,7,8,9
weights	‘uniform’, ‘distance’
metric	‘euclidean’, ‘manhattan’

Μετά την εκτέλεση του Grid Search, πραγματοποιήθηκαν συνολικά **32 πειράματα**. Από αυτά, παρουσιάζονται οι 10 καλύτεροι συνδυασμοί υπερπαραμέτρων, επιλεγμένοι βάσει του validation accuracy. Η βέλτιστη αρχιτεκτονική του μοντέλου KNN προέκυψε με τις εξής υπερπαραμέτρους metric = ‘manhattan’, n_neighbors = 8 και weights = ‘uniform’. Ο Πίνακας 6.1.2.2 περιλαμβάνει τους δέκα καλύτερους συνδυασμούς υπερπαραμέτρων, ταξινομημένους με βάση το validation accuracy.

Πίνακας 6.1.2.2: Οι top 10 συνδιασμοί υπερ παράμετρων ταξινομημένοι με βάση το validation accuracy για το KNN

param_metric	param_n_neighbors	param_weights	mean_train_accuracy	mean_test_accuracy	mean_train_loss	mean_test_loss
manhattan	8	uniform	0.6412	0.5417	0.3588	0.4583
manhattan	9	uniform	0.6344	0.5409	0.3656	0.4591
manhattan	9	distance	0.9680	0.5397	0.0320	0.4603
manhattan	6	uniform	0.6648	0.5390	0.3352	0.4610
manhattan	8	distance	0.9680	0.5387	0.0320	0.4613
manhattan	5	distance	0.9680	0.5371	0.0320	0.4629
manhattan	5	uniform	0.6821	0.5370	0.3179	0.4630
manhattan	7	uniform	0.6519	0.5364	0.3481	0.4636
manhattan	6	distance	0.9680	0.5359	0.0320	0.4641
manhattan	7	distance	0.9680	0.5359	0.0320	0.4641

Η Εικόνα 6.1.2 απεικονίζει την απόδοση του μοντέλου για όλους τους συνδυασμούς υπερπαραμέτρων που δοκιμάστηκαν.



Εικόνα 6.1.2: Validation Accuracy για όλα τα πειράματα KNN

Τα αποτελέσματα δείχνουν ότι η χρήση της Manhattan distance παρήγαγε υψηλότερη ακρίβεια από την Euclidean distance. Αυτό υποδηλώνει ότι τα δεδομένα μας είναι πιθανώς υψηλών διαστάσεων, όπου κάθε χαρακτηριστικό συμβάλλει πιο ανεξάρτητα, χωρίς ισχυρή συσχέτιση με τα υπόλοιπα.

Στις υψηλές διαστάσεις, η Euclidean distance μπορεί να επηρεάζεται δυσανάλογα από μεγάλες αποκλίσεις, καθώς βασίζεται στο τετράγωνο των διαφορών. Αυτό μπορεί να μην αντικατοπτρίζει την πραγματική ομοιότητα μεταξύ των σημείων δεδομένων. Αντίθετα, η Manhattan distance, βασιζόμενη στην απόλυτη διαφορά μεταξύ των χαρακτηριστικών, αντιμετωπίζει πιο ισότιμα τη συμβολή κάθε χαρακτηριστικού. Έτσι, αναδεικνύει καλύτερα τα μοτίβα σε δεδομένα με χαμηλότερη συσχέτιση μεταξύ των χαρακτηριστικών, καθιστώντας την καταλληλότερη επιλογή για το συγκεκριμένο dataset.

6.1.3 Grid Search για SVM με 4 κλάσεις

Στην παρούσα ανάλυση εφαρμόστηκε η μέθοδος Grid Search για τη βελτιστοποίηση των υπερπαραμέτρων του μοντέλου Support Vector Machine (SVM) σε ένα πρόβλημα ταξινόμησης με τέσσερις κλάσεις. Οι υπερπαράμετροι που εξετάστηκαν, καθώς και οι αντίστοιχες τιμές που δοκιμάστηκαν, παρουσιάζονται στον Πίνακα 6.1.3.1.

Πίνακας 6.1.3.1: Υπερ παράμετροι και τιμές που δοκιμάστηκαν με την χρήση του GridSearch για το SVM

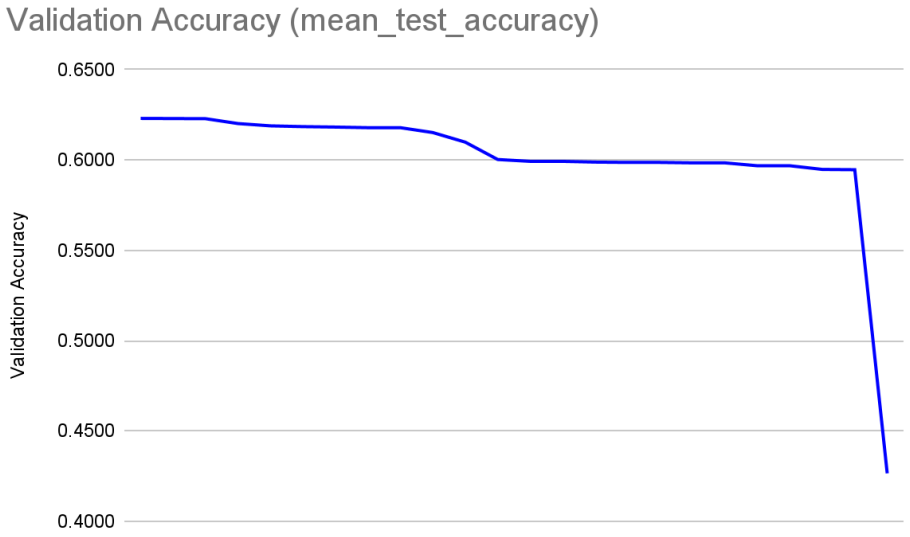
C	0.1, 1, 10, 100
kernel	‘linear’, ‘rbf’, ‘poly’
gamma	‘scale’, ‘auto’

Μετά την εκτέλεση του Grid Search, πραγματοποιήθηκαν συνολικά **24 πειράματα**. Από αυτά, παρουσιάζονται οι 10 καλύτεροι συνδυασμοί υπερπαραμέτρων, ταξινομημένοι με βάση το validation accuracy. Η βέλτιστη αρχιτεκτονική του μοντέλου SVM προέκυψε με τις εξής υπερπαραμέτρους: C = 10, gamma = ‘auto’ και kernel = ‘rbf’. Ο Πίνακας 6.1.3.2 περιλαμβάνει τους δέκα καλύτερους συνδυασμούς υπερπαραμέτρων, ταξινομημένους με βάση το validation accuracy.

Πίνακας 6.1.3.2: Οι top 10 συνδιασμοί υπερ παράμετρων ταξινομημένοι με βάση το validation accuracy για το SVM

param_C	param_gamma	param_kernel	mean_train_accuracy	mean_test_accuracy	mean_train_loss	mean_test_loss
10	auto	rbf	0.6820	0.6230	0.3180	0.3770
100	auto	rbf	0.7249	0.6229	0.2751	0.3771
10	scale	rbf	0.7082	0.6228	0.2918	0.3772
1	scale	rbf	0.6598	0.6201	0.3402	0.3799
10	auto	poly	0.6764	0.6189	0.3236	0.3811
1	scale	poly	0.6710	0.6185	0.3290	0.3815
100	auto	poly	0.7154	0.6182	0.2846	0.3818
1	auto	rbf	0.6393	0.6179	0.3607	0.3821
10	scale	poly	0.7087	0.6179	0.2913	0.3821
100	scale	rbf	0.7611	0.6152	0.2389	0.3848

Η Εικόνα 6.1.3 απεικονίζει την απόδοση του μοντέλου για όλους τους συνδυασμούς υπερπαραμέτρων που δοκιμάστηκαν.



Εικόνα 6.1.3: Validation Accuracy για όλα τα πειράματα SVM

Η συγκεκριμένη διαμόρφωση του μοντέλου δείχνει ότι το dataset πιθανώς περιέχει μη γραμμικούς διαχωρισμούς και ότι μια προσεκτική επιλογή των παραμέτρων C και gamma μπορεί να οδηγήσει σε ένα μοντέλο που διατηρεί καλή ισορροπία μεταξύ ακρίβειας και γενίκευσης.

6.1.4 Grid Search για LR με 4 κλάσεις

Στην παρούσα ανάλυση εφαρμόστηκε η μέθοδος Grid Search για τη βελτιστοποίηση των υπερπαραμέτρων του μοντέλου Logistic Regression (LR) σε ένα πρόβλημα ταξινόμησης με τέσσερις κλάσεις. Οι υπερπαράμετροι που εξετάστηκαν, καθώς και οι αντίστοιχες τιμές που δοκιμάστηκαν, παρουσιάζονται στον Πίνακα 6.1.4.1.

Πίνακας 6.1.4.1: Υπερ παράμετροι και τιμές που δοκιμάστηκαν με την χρήση του GridSearch για το LR

C	0.1, 1, 10, 100
solver	‘newton-cg’, lbfgs’, ‘liblinear’, ‘sag’, ‘saga’
max_iter	1000, 1500, 2000

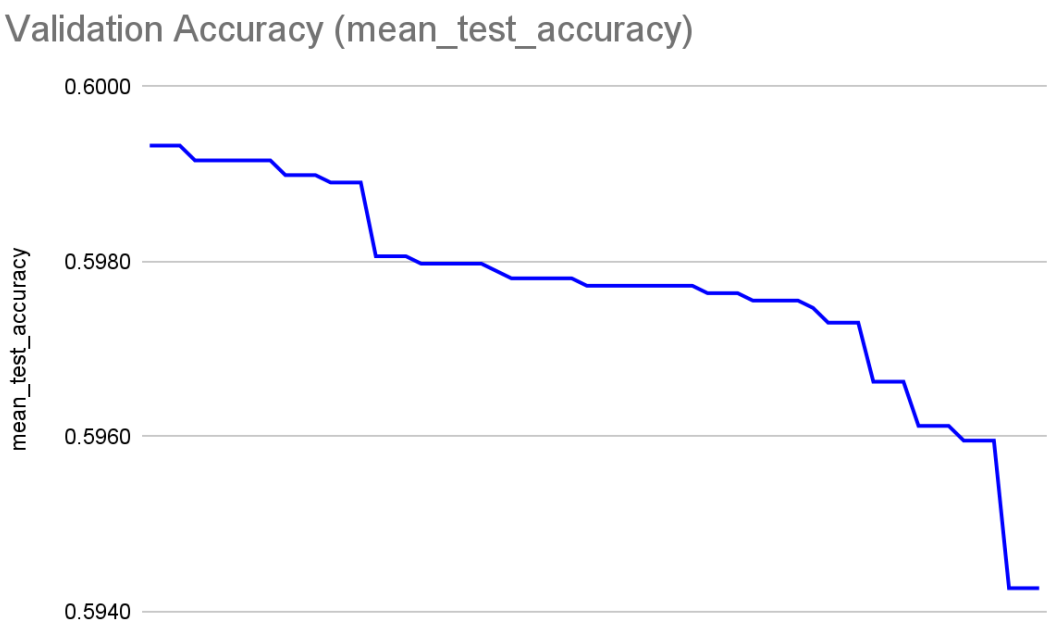
Μετά την εκτέλεση του Grid Search, πραγματοποιήθηκαν συνολικά **60 πειράματα**. Από αυτά, παρουσιάζονται οι 10 καλύτεροι συνδυασμοί υπερπαραμέτρων, ταξινομημένοι βάσει του validation accuracy. Η βέλτιστη αρχιτεκτονική του μοντέλου Logistic Regression προέκυψε με τις εξής υπερπαραμέτρους: C = 1, max_iter = 2000 και solver = ‘newton-cg’. Ο

Πίνακας 6.1.4.2 περιλαμβάνει τους δέκα καλύτερους συνδυασμούς υπερπαραμέτρων, ταξινομημένους με βάση το validation accuracy.

Πίνακας 6.1.4.2: Οι top 10 συνδιασμοί υπερ παράμετρων ταξινομημένοι με βάση το validation accuracy για το LR

param_C	param_max_iter	param_solver	mean_train_accuracy	mean_test_accuracy	mean_train_loss	mean_test_loss
1	2000	newton-cg	0.6068	0.5993	0.3932	0.4007
1	1500	newton-cg	0.6068	0.5993	0.3932	0.4007
1	1000	newton-cg	0.6068	0.5993	0.3932	0.4007
1	1000	saga	0.6070	0.5992	0.3930	0.4008
1	2000	saga	0.6070	0.5992	0.3930	0.4008
1	1500	saga	0.6070	0.5992	0.3930	0.4008
1	2000	sag	0.6070	0.5992	0.3930	0.4008
1	1500	sag	0.6070	0.5992	0.3930	0.4008
1	1000	sag	0.6070	0.5992	0.3930	0.4008
1	1500	lbfgs	0.6071	0.5990	0.3929	0.4010

Η Εικόνα 6.1.4 απεικονίζει την απόδοση του μοντέλου για όλους τους συνδυασμούς υπερπαραμέτρων που δοκιμάστηκαν.



Εικόνα 6.1.4: Validation Accuracy για όλα τα πειράματα LR

Η συγκεκριμένη διαμόρφωση υποδηλώνει ότι το dataset είναι αρκετά πολύπλοκο ώστε να απαιτεί σημαντικό αριθμό επαναλήψεων, ενώ το Newton-CG solver ήταν πιο αποδοτικό στη σύγκλιση των παραμέτρων συγκριτικά με άλλες επιλογές (π.χ. lbfgs, sag, saga).

6.1.5 Hyperparameter Tuning για Neural Network με 4 κλάσεις

Στην παρούσα ανάλυση πραγματοποιήθηκε χειροκίνητη αναζήτηση υπερπαραμέτρων (manual hyperparameter tuning) για τη βελτιστοποίηση ενός Νευρωνικού Δικτύου (Neural Network) σε ένα πρόβλημα ταξινόμησης με τέσσερις κλάσεις. Οι υπερπαράμετροι που εξετάστηκαν, καθώς και οι αντίστοιχες τιμές που δοκιμάστηκαν, παρουσιάζονται στον Πίνακα 6.1.5.1.

Πίνακας 6.1.5.1: Υπερ παράμετροι και τιμές που δοκιμάστηκαν χειροκίνητα για τα Νευρωνικά Δίκτυα

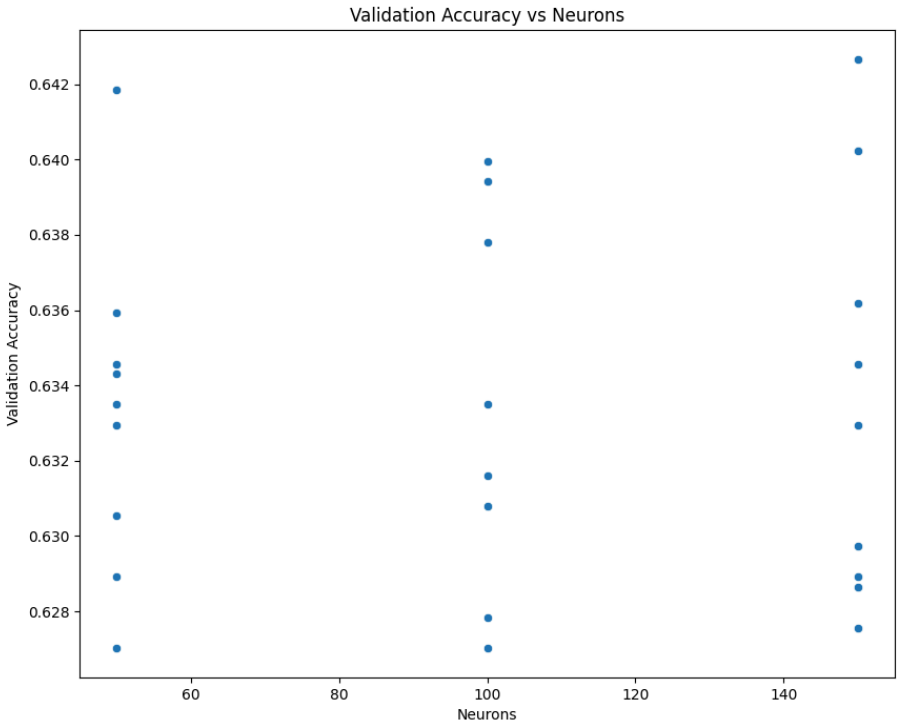
Number of Neurons	50,100,150
Batch Size	64, 128, 256
Dropout	0, 0.1, 0.5

Μετά την εκτέλεση της διαδικασίας tuning, πραγματοποιήθηκαν συνολικά **27 πειράματα**. Από αυτά, επιλέχθηκαν οι 10 καλύτερες αρχιτεκτονικές, ταξινομημένες με βάση το validation accuracy (mean_test_accuracy). Η βέλτιστη αρχιτεκτονική του μοντέλου προέκυψε με τις εξής υπερπαραμέτρους: Neurons = 150, Dropout Rate = 0.5 και Batch Size = 256. Ο Πίνακας 6.1.5.2 περιλαμβάνει τους δέκα καλύτερους συνδυασμούς υπερπαραμέτρων, ταξινομημένους με βάση το validation accuracy.

Πίνακας 6.1.5.2: Οι top 10 συνδιασμοί υπερ παράμετρων ταξινομημένοι με βάση το validation accuracy για τα Νευρωνικά Δίκτυα

Neurons	Dropout Rate	Batch Size	Validation Loss	Validation Accuracy
150	0.5	256	0.8132	0.6427
50	0.5	128	0.8139	0.6419
150	0.5	128	0.8149	0.6402
100	0.5	128	0.8081	0.6400
100	0.5	256	0.8086	0.6394
100	0.5	64	0.8188	0.6378
150	0.5	64	0.8312	0.6362
50	0.5	64	0.8195	0.6359
50	0.1	256	0.8132	0.6346
150	0.1	128	0.8257	0.6346

Εικόνα 6.1.5.1: Heatmap με τους συνδυασμούς Dropout Rate και Batch Size για το Validation Accuracy



Εικόνα 6.1.5.2: Scatter plot με τον αριθμό Νευρώνων και το Validation Accuracy

Η Εικόνα 6.1.5.1 απεικονίζει έναν heatmap που δείχνει τον αντίκτυπο του Dropout Rate και του Batch Size στο Validation Accuracy. Επιπλέον, η Εικόνα 6.1.5.2 παρουσιάζει ένα scatter plot που απεικονίζει τη σχέση μεταξύ του αριθμού των νευρώνων και του Validation Accuracy, επιτρέποντας την οπτικοποίηση της τάσης απόδοσης του μοντέλου σε σχέση με το πλήθος των νευρώνων στο δίκτυο.

6.2 Πειραμα 2: Αξιολόγηση χαρακτηριστικών με Random Forest και Feature Importance

Το feature importance στο Random Forest αποτελεί ένα μέτρο που υποδεικνύει τη σημασία κάθε χαρακτηριστικού στη διαδικασία πρόβλεψης του μοντέλου. Το Random Forest αποτελείται από πολλά δέντρα απόφασης, όπου κάθε δέντρο εκπαιδεύεται με τυχαία δείγματα δεδομένων και τυχαία υποσύνολα χαρακτηριστικών. Σε κάθε κόμβο του δέντρου, επιλέγεται το χαρακτηριστικό που ελαχιστοποιεί την αβεβαιότητα, η οποία μετριέται μέσω του Gini impurity. Η μείωση της αβεβαιότητας ονομάζεται Information Gain ή Gini Gain.

Στη συνέχεια, το Random Forest υπολογίζει τη σημασία κάθε χαρακτηριστικού αθροίζοντας το Information Gain σε όλους τους κόμβους όπου χρησιμοποιήθηκε το χαρακτηριστικό, σε όλα τα δέντρα του μοντέλου. Το άθροισμα αυτό κανονικοποιείται, ώστε το feature importance να αποδίδει ένα score μεταξύ 0 και 1.

Η ανάλυση του feature importance επιτρέπει την αναγνώριση των χαρακτηριστικών που επηρεάζουν περισσότερο τις αποφάσεις του μοντέλου, βοηθώντας στην εκτίμηση της ανάγκης για αφαίρεση λιγότερο σημαντικών χαρακτηριστικών. Ο στόχος είναι η μείωση της πολυπλοκότητας του μοντέλου και του χρόνου εκπαίδευσης, διατηρώντας παράλληλα υψηλή απόδοση. Για την αξιολόγηση των χαρακτηριστικών, χρησιμοποιήθηκε ο ταξινομητής lsbp με Random Forest. Όπως αναφέραμε και στην ενότητα 5.3 Εξαγωγή Χαρακτηριστικών, κρατήσαμε όλα τα προτεινόμενα χαρακτηριστικά με σκοπό να αξιολογηθούν σε αυτό το πείραμα και να αποφασιστεί αν θα τα κρατήσουμε στο τελικό σύνολο.

Πίνακας 6.2: Top 10 χαρακτηριστικά και το feature importance τους για random forest μοντέλο 4 κλάσεων

	Unfiltered		**Importance >= 1%**		**Importance >= 2%**	
	Feature	Importance	Feature	Importance	Feature	Importance
	Q Entropy (Y)	0.05	Q Entropy (Y)	0.06	Q Entropy (Y)	0.07

	Std C (X)	0.04	Std C (X)	0.05	Std C (X)	0.06
	Normalized Range (Y)	0.04	Normalized Range (Y)	0.05	Normalized Range (Y)	0.06
	Normalized Mean (Y)	0.04	X_Type_c	0.05	Normalized Mean (Y)	0.05
	Normalized Median (Y)	0.03	Normalized Mean (Y)	0.04	X_Type_c	0.05
	X_Type_c	0.03	Normalized Median (Y)	0.04	Normalized Median (Y)	0.05
	C Entropy (X)	0.03	Minimum Value (Y)	0.04	Minimum Value (Y)	0.05
	Coefficient of Variation (Y)	0.03	Coefficient of Variation (Y)	0.04	Coefficient of Variation (Y)	0.05
	Minimum Value (Y)	0.03	Skewness (Y)	0.04	Unique C (X)	0.05
	Sample Standard Deviation (Y)	0.03	Unique C (X)	0.04	Skewness (Y)	0.05
Accuracy	65.99%		64.89%		62.00%	

Στον Πίνακα 6.2 παρουσιάζονται τα δέκα σημαντικότερα χαρακτηριστικά σύμφωνα με το feature importance για το Random Forest μοντέλο τεσσάρων κλάσεων. Αρχικά, στο πρώτο πείραμα χρησιμοποιήθηκαν και τα 83 χαρακτηριστικά για την εκπαίδευση του μοντέλου. Ωστόσο, διαπιστώθηκε ότι πολλά από αυτά ήταν παράγωγα άλλων χαρακτηριστικών και πιθανώς εισήγαγαν πλεοναστική πληροφορία ή θόρυβο.

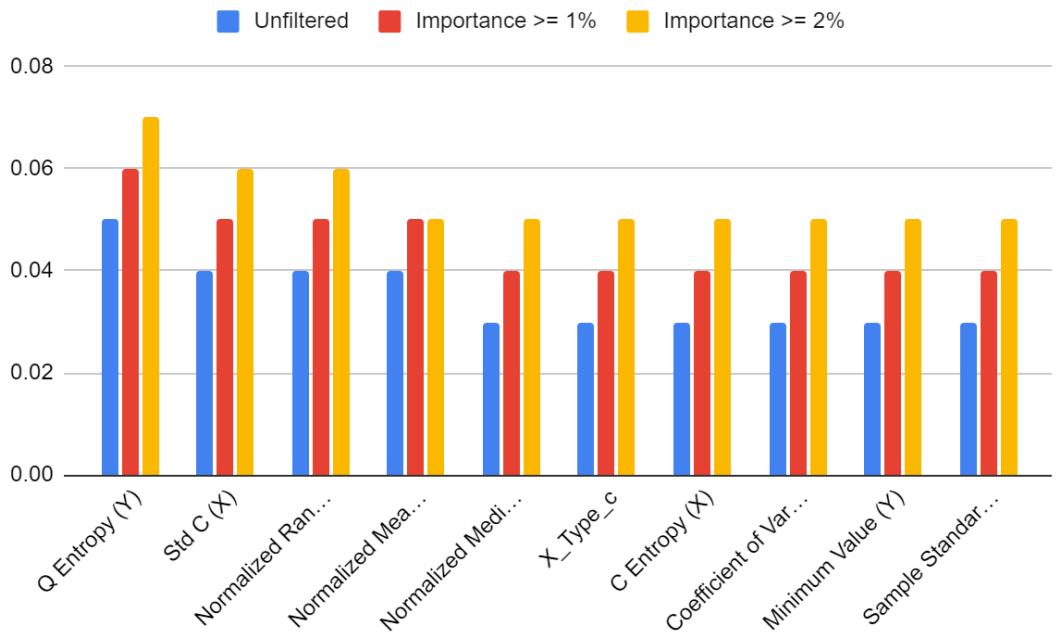
Για να αξιολογηθεί η επίδραση της μείωσης των χαρακτηριστικών, πραγματοποιήθηκαν δύο επιπλέον πειράματα. Στο δεύτερο πείραμα αγνοήθηκαν χαρακτηριστικά με feature importance μικρότερο του 1%, με αποτέλεσμα να παραμείνουν 30 χαρακτηριστικά. Στο τρίτο πείραμα αγνοήθηκαν χαρακτηριστικά με feature importance μικρότερο του 2%, μειώνοντας τα σε 22. Τα αποτελέσματα της μείωσης των χαρακτηριστικών έδειξαν ότι η απόδοση του μοντέλου μειώθηκε, καθώς το validation accuracy από 65.99% στο πρώτο πείραμα μειώθηκε στο 64.89% στο δεύτερο και στο 62.00% στο τρίτο.

Η ανάλυση των αποτελεσμάτων έδειξε ότι η μείωση των χαρακτηριστικών δεν βελτίωσε την απόδοση του μοντέλου. Αντίθετα, η ποικιλία των χαρακτηριστικών φαίνεται να είναι πιο σημαντική για την ταξινόμηση από την απόλυτη σημασία των μεμονωμένων χαρακτηριστικών. Λόγω αυτής της διαπίστωσης, στις επόμενες δοκιμές κρατήθηκε το αρχικό σύνολο των χαρακτηριστικών.

Η διερεύνηση του feature importance έδειξε ότι χαρακτηριστικά που σχετίζονται με στατιστικά μέτρα, όπως Entropy, Range, Mean και Median, είχαν τη μεγαλύτερη επιρροή στην ταξινόμηση. Τα περισσότερα σημαντικά χαρακτηριστικά προέρχονται από τη στήλη "Y", γεγονός που υποδηλώνει ότι περιέχει τη μεγαλύτερη πληροφορία για την τελική ταξινόμηση. Το χαρακτηριστικό με το υψηλότερο feature importance ήταν το Q Entropy (Y), το οποίο εκφράζει την εντροπία των ποσοτικών τιμών της στήλης Y.

Η Εικόνα 6.2 παρουσιάζει τη σημασία των χαρακτηριστικών για τις τρεις διαφορετικές περιπτώσεις: αρχικά για το πλήρες σύνολο των 83 χαρακτηριστικών, στη συνέχεια για τα χαρακτηριστικά με feature importance τουλάχιστον 1% και τέλος για εκείνα με feature importance τουλάχιστον 2%. Η απεικόνιση επιβεβαιώνει την αύξηση της σχετικής σημασίας των χαρακτηριστικών μετά τη μείωση του πλήθους τους.

Τα αποτελέσματα δείχνουν ότι η αφαίρεση χαρακτηριστικών με χαμηλό feature importance δεν οδήγησε σε βελτίωση της απόδοσης. Αντιθέτως, η ποικιλία των χαρακτηριστικών αποδείχθηκε πιο σημαντική για την ταξινόμηση. Για τον λόγο αυτό, επιλέχθηκε να διατηρηθεί το πλήρες σύνολο των 83 χαρακτηριστικών, καθώς προσέφερε τη βέλτιστη ακρίβεια στο μοντέλο.



Εικόνα 6.2: Top 10 features και το importance τους για το unfiltered δεδομένα, για importance >= 1% και για importance >= 2%

6.3 *Πειραμα 3: Σύγκριση Κλασικών Μεθόδων Μοντελοποίησης*

Το παρόν πείραμα επικεντρώνεται στη σύγκριση κλασικών αλγορίθμων ταξινόμησης, εξετάζοντας την απόδοσή τους σε διαφορετικούς συνδυασμούς ετικετών. Όπως αναφέρθηκε στην ενότητα 5.5, το σύστημα έχει σχεδιαστεί ώστε να χρησιμοποιεί ξεχωριστούς ταξινομητές για κάθε πιθανό σύνολο ετικετών, επιτρέποντας μεγαλύτερη ευελιξία στην επιλογή του μοντέλου ανάλογα με τις ανάγκες του χρήστη.

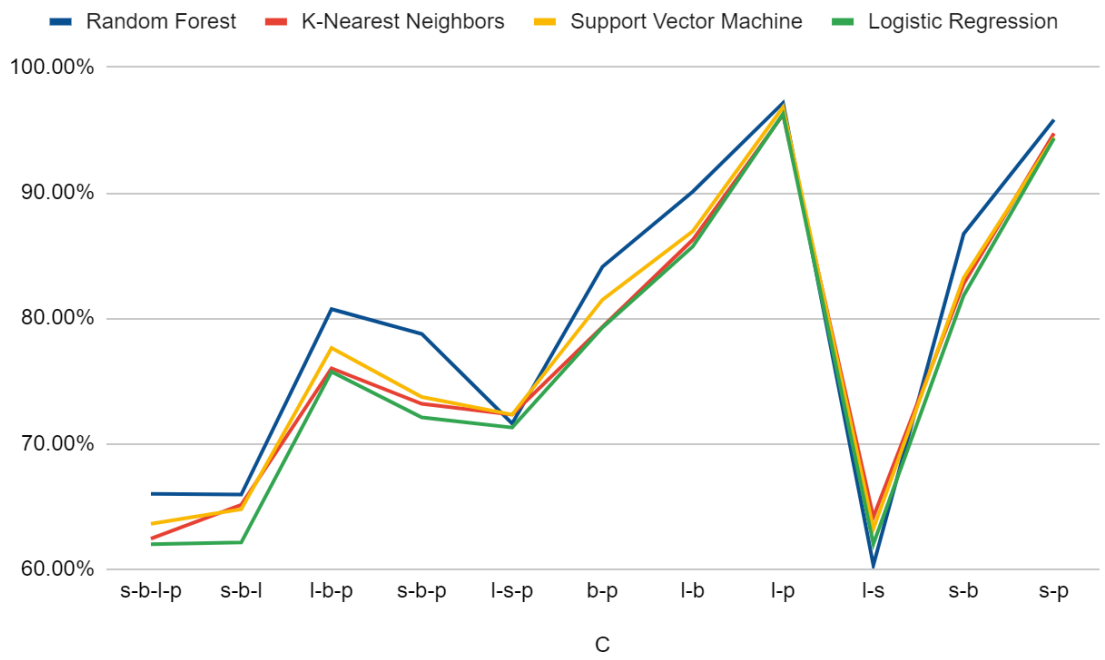
α την αξιολόγηση, δοκιμάστηκαν οι μέθοδοι Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM) και Logistic Regression. Συνολικά πραγματοποιήθηκαν 44 πειράματα, καθώς κάθε μία από τις τέσσερις μεθόδους εφαρμόστηκε σε 11 διαφορετικούς συνδυασμούς ετικετών. Ως μετρική απόδοσης χρησιμοποιήθηκε το Validation Accuracy σε unseen δεδομένα, παρέχοντας μια αντικειμενική εκτίμηση της γενίκευσης των μοντέλων.

Τα αποτελέσματα συνοψίζονται στον Πίνακα 6.3, όπου καταγράφεται η επίδοση κάθε μεθόδου ταξινόμησης. Επιπλέον, η Εικόνα 6.3.1 παρουσιάζει συνολικά τη διακύμανση του validation accuracy μεταξύ των μοντέλων, ενώ η Εικόνα 6.3.2 εστιάζει ειδικά στην απόδοση του Random Forest.

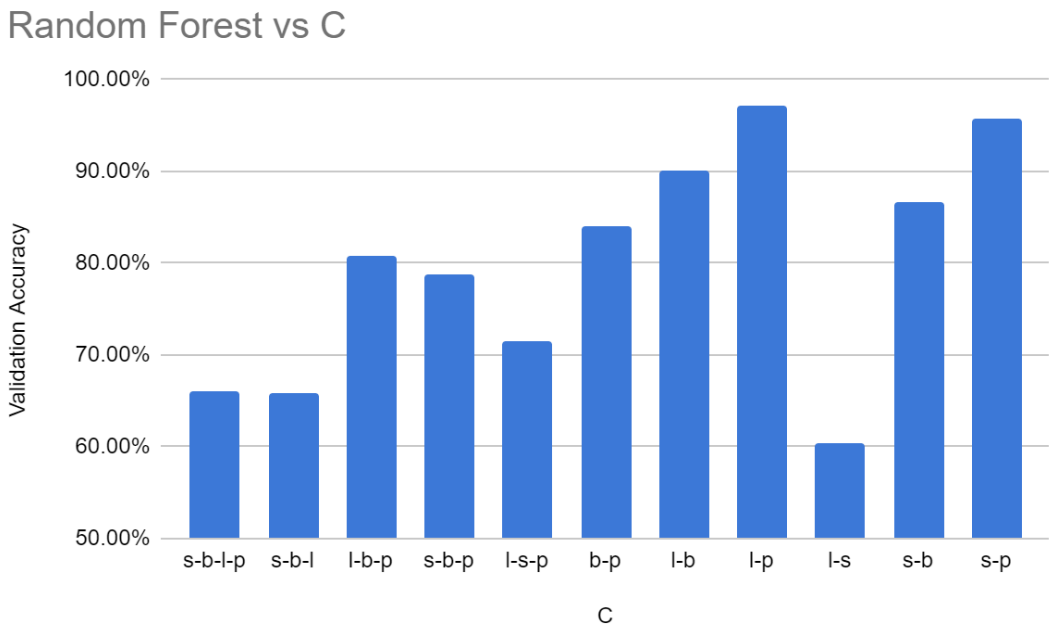
Πίνακας 6.3: Επίδοση Random Forest, K-Nearest Neighbors, Support Vector Machines και Logistic Regression

	Model			
C	Random Forest	K-Nearest Neighbors	Support Vector Machine	Logistic Regression
s-b-l-p	**65.99%**	62.42%	63.60%	61.98%
s-b-l	**65.93%**	65.12%	64.76%	62.11%
l-b-p	**80.72%**	76.00%	77.62%	75.73%
s-b-p	**78.74%**	73.17%	73.71%	72.09%
l-s-p	71.60%	**72.31%**	**72.31%**	71.28%
b-p	**84.10%**	79.31%	81.47%	79.25%
l-b	**90.09%**	86.25%	86.93%	85.71%
l-p	**97.17%**	96.23%	96.83%	96.29%
l-s	60.31%	**64.15%**	63.21%	61.99%
s-b	**86.73%**	82.75%	83.22%	81.81%

s-p	**95.82%**	94.74%	94.41%	94.34%



Εικόνα 6.3.1: Validation Accuracy για κλασικά μοντέλα



Εικόνα 6.3.2: Validation Accuracy για Random Forest

Από την ανάλυση των αποτελεσμάτων, προκύπτει ότι οι ταξινομητές αντιμετωπίζουν μεγαλύτερη δυσκολία στην ταξινόμηση ετικετών που αφορούν τις κατηγορίες ‘line’ και ‘scatter’. Στο Random Forest, όπως φαίνεται στην Εικόνα 6.3.2, οι συνδυασμοί ετικετών s-b-l-p, s-b-l, l-s-p και l-s παρουσιάζουν σημαντικά χαμηλότερη ακρίβεια σε σχέση με άλλες κατηγορίες. Αυτό μπορεί να οφείλεται στο ότι οι δύο αυτές επιλογές (line και scatter) χρησιμοποιούνται συχνά με παρόμοιο τρόπο, καθιστώντας πιο δύσκολο για το μοντέλο να τις διαχωρίσει.

Αναλύοντας τα αποτελέσματα των διαφορετικών αλγορίθμων, το Random Forest αναδεικνύεται ως η πιο αποτελεσματική επιλογή, επιτυγχάνοντας την υψηλότερη ακρίβεια στα περισσότερα σύνολα ετικετών. Η ισχυρή απόδοση του Random Forest οφείλεται στη δυνατότητά του να συλλαμβάνει πολύπλοκες σχέσεις και αλληλεπιδράσεις μεταξύ των χαρακτηριστικών, αξιοποιώντας τη φύση του ως ensemble μοντέλου. Η χρήση πολλαπλών δέντρων απόφασης επιτρέπει στο Random Forest να μοντελοποιεί πιο σύνθετες δομές δεδομένων, οδηγώντας σε καλύτερη γενίκευση.

Αντίθετα, το Logistic Regression σημείωσε τη χαμηλότερη απόδοση, γεγονός που υποδηλώνει ότι η υπόθεση μιας γραμμικής σχέσης μεταξύ των χαρακτηριστικών και της εξαρτημένης μεταβλητής δεν ισχύει στα συγκεκριμένα δεδομένα. Αυτή η διαπίστωση ενισχύεται από τα αποτελέσματα του Πειράματος 6.1.2, όπου η χρήση της Manhattan distance στο KNN έδειξε καλύτερη απόδοση, υποδηλώνοντας ότι τα χαρακτηριστικά δεν είναι ιδιαίτερα συσχετισμένα. Συνεπώς, το Logistic Regression φαίνεται να μην μπορεί να αποτυπώσει τις πιθανές μη γραμμικές σχέσεις στα δεδομένα.

Οι μέθοδοι KNN και SVM πέτυχαν μέτρια αποτελέσματα, δείχνοντας ότι έχουν τη δυνατότητα να διαχειριστούν τα δεδομένα αποτελεσματικά, αλλά πιθανώς απαιτούν καλύτερη ρύθμιση υπερπαραμέτρων ή καλύτερη επεξεργασία των χαρακτηριστικών ώστε να μεγιστοποιηθεί η απόδοσή τους.

Συμπερασματικά, τα αποτελέσματα υποδεικνύουν ότι το Random Forest αποτελεί την πιο αξιόπιστη επιλογή για το συγκεκριμένο πρόβλημα ταξινόμησης, επιτυγχάνοντας υψηλή απόδοση και αξιοπιστία. Αντίθετα, το Logistic Regression αποδεικνύεται ανεπαρκές για την καταγραφή της πολυπλοκότητας των δεδομένων, ενώ τα μοντέλα KNN και SVM δείχνουν προοπτικές βελτίωσης μέσω βελτιστοποίησης των ρυθμίσεων τους.

Η στρατηγική one-vs-one (OvO) που περιγράφηκε στην ενότητα 5.5 αποδεικνύεται χρήσιμη στην παρούσα μελέτη, καθώς η δυνατότητα επιλογής διαφορετικών ταξινομητών για κάθε υποσύνολο ετικετών επιτρέπει την καλύτερη προσαρμογή στις ανάγκες του εκάστοτε ταξινομητικού προβλήματος.

6.4 Πειραμα 4: Σύγκριση Random Forest - Neural Network - AutoGluon - Autokeras

Στο τελικό στάδιο της πειραματικής διαδικασίας, δοκιμάστηκαν οι πιο αποδοτικοί αλγόριθμοι ταξινόμησης με σκοπό την επιλογή του καταλληλότερου για την υλοποίηση του προβλήματος. Όπως αναφέρθηκε και στην ενότητα 5.5, η μοντελοποίηση βασίζεται στη δυνατότητα επιλογής διαφορετικών ταξινομητών για κάθε συνδυασμό ετικετών, προκειμένου να επιτευχθεί η μέγιστη δυνατή απόδοση.

Για τη σύγκριση, εκτελέστηκαν 11 πειράματα για κάθε συνδυασμό δύο, τριών και τεσσάρων κλάσεων, χρησιμοποιώντας τα Random Forest, Neural Networks, AutoGluon και AutoKeras. Ως μετρική αξιολόγησης επιλέχθηκε το Validation Accuracy σε unseen δεδομένα, παρέχοντας μια αντικειμενική εκτίμηση της γενίκευσης των μοντέλων. Τα αποτελέσματα συνοψίζονται στον Πίνακα 6.4, όπου αποτυπώνεται η απόδοση κάθε μοντέλου.

Πίνακας 6.4: Επίδοση Random Forest, Neural Networks, AutoGluon και AutoKeras

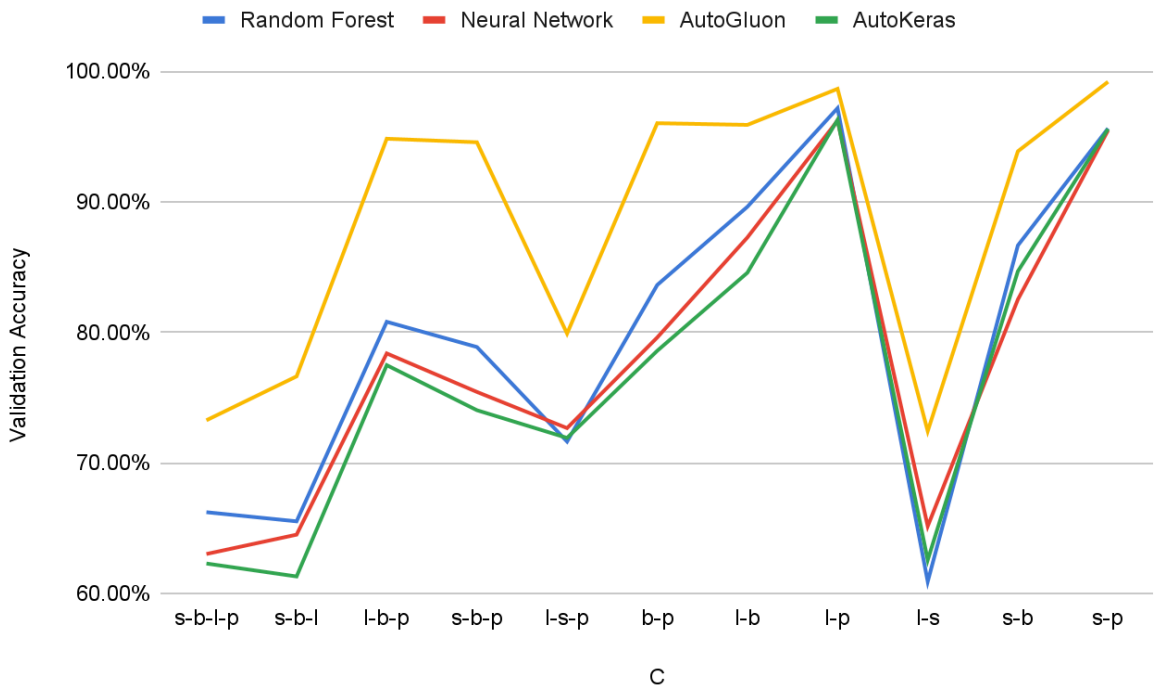
	Model			
C	Random Forest	Neural Network	AutoGluon	Autokeras
s-b-l-p	65.99%	63.03%	**73.27%**	62.30%
s-b-l	66.27%	64.51%	**76.63%**	61.31%
l-b-p	80.81%	78.39%	**94.83%**	77.49%
s-b-p	78.88%	75.44%	**94.56%**	74.04%
l-s-p	71.64%	72.67%	**79.91%**	71.92%
b-p	83.63%	79.61%	**96.02%**	78.59%
l-b	89.62%	87.27%	**95.89%**	84.57%
l-p	97.17%	96.22%	**98.65%**	96.28%
l-s	60.92%	65.16%	**72.44%**	62.57%
s-b	86.66%	82.52%	**93.87%**	84.68%
s-p	95.62%	95.47%	**99.19%**	95.58%

Η σύγκριση των αποτελεσμάτων δείχνει ότι το AutoGluon πέτυχε με διαφορά τις καλύτερες αποδόσεις σε όλα τα classification μοντέλα που εκπαιδεύτηκαν. Ειδικότερα, στο πείραμα με τέσσερις κλάσεις, το οποίο παρουσίασε τις χαμηλότερες συνολικά επιδόσεις, το AutoGluon

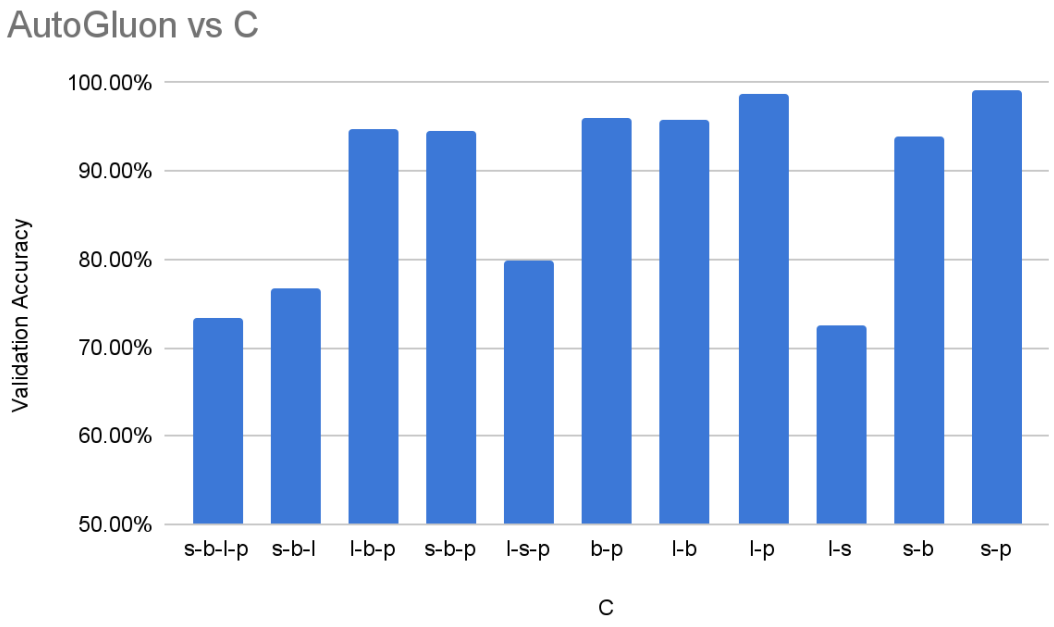
πέτυχε validation accuracy 73.27%, σημειώνοντας αισθητή διαφορά από το δεύτερο καλύτερο μοντέλο, το Random Forest, το οποίο έφτασε στο 66.99%.

Τα Random Forest, Neural Networks και AutoKeras παρουσίασαν σχετικά ικανοποιητική απόδοση, ωστόσο δεν κατάφεραν να φτάσουν την αποδοτικότητα του AutoGluon. Σε όλα τα πειράματα, τα μοντέλα φαίνεται να δυσκολεύονται ιδιαίτερα στη διάκριση μεταξύ των ετικετών “scatter” και “line”, οι οποίες παρουσιάζουν υψηλό βαθμό ομοιότητας μεταξύ τους. Αυτό επιβεβαιώνεται από την Εικόνα 6.4.2, όπου φαίνεται ότι η απόδοση του AutoGluon σε αυτές τις ετικέτες είναι αισθητά χαμηλότερη, καθώς και από τα ευρήματα του προηγούμενου Πειράματος 6.3.

Παρόλο που το AutoKeras ανήκει επίσης στην κατηγορία των AutoML μοντέλων, δεν πέτυχε αντίστοιχα υψηλές επιδόσεις. Συγκριτικά με τα υπόλοιπα μοντέλα, παρουσιάζει τις χαμηλότερες επιδόσεις σχεδόν σε όλα τα classification προβλήματα, υποδηλώνοντας ότι η προσέγγισή του μπορεί να μην είναι τόσο αποδοτική για το συγκεκριμένο σύνολο δεδομένων.



Εικόνα 6.4.1: Validation Accuracy για τα καλύτερα μοντέλα



Εικόνα 6.4.2: Validation Accuracy για AutoGluon

Βάσει των αποτελεσμάτων, επιλέχθηκε το AutoGluon ως ο μοναδικός ταξινομητής που θα χρησιμοποιηθεί για τη σύσταση των οπτικοποιήσεων στην ανάπτυξη του Web App. Η ανωτερότητά του έναντι των υπολοίπων μοντέλων, ειδικά σε προβλήματα πολλαπλών κλάσεων, το καθιστά την ιδανική επιλογή για την υλοποίηση της εφαρμογής, εξασφαλίζοντας υψηλή ακρίβεια και προσαρμοστικότητα σε διαφορετικούς συνδυασμούς ετικετών.

Η Εικόνα 6.4.1 παρέχει μια συνολική σύγκριση του Validation Accuracy για τα καλύτερα μοντέλα, ενώ η Εικόνα 6.4.2 εστιάζει ειδικά στην απόδοση του AutoGluon, αποδεικνύοντας τη σημαντική υπεροχή του σε σχέση με τις άλλες μεθόδους ταξινόμησης.

6.5 *Σύνοψη συμπερασμάτων αξιολόγησης*

πό την ανάλυση και τα πειράματα που πραγματοποιήθηκαν, προκύπτουν σημαντικά συμπεράσματα σχετικά με τη φύση των δεδομένων και την απόδοση των διαφορετικών ταξινομητών:

- Τα αποτελέσματα του πειράματος 6.1.2 δείχνουν ότι η απόσταση Manhattan είχε καλύτερη απόδοση από την Euclidean, υποδεικνύοντας ότι τα δεδομένα μας είναι υψηλών διαστάσεων και χαρακτηρίζονται από χαμηλή συσχέτιση μεταξύ των

χαρακτηριστικών. Αυτό σημαίνει ότι κάθε χαρακτηριστικό συνεισφέρει πιο ανεξάρτητα στη συνολική πληροφορία του dataset, γεγονός που μπορεί να εξηγεί γιατί η Euclidean απόσταση, η οποία επηρεάζεται περισσότερο από μεγάλες τιμές λόγω της τετραγωνικής της φύσης, δεν απέδωσε εξίσου καλά. Η προτίμηση του μοντέλου προς τη Manhattan απόσταση υποδηλώνει ότι η σχέση μεταξύ των χαρακτηριστικών είναι τέτοια που η απλή άθροιση των αποστάσεων παράγει καλύτερα αποτελέσματα από την τετραγωνική απόσταση που χρησιμοποιείται στην Euclidean metric.

- Το πείραμα 6.2 εξετάζει την αφαίρεση χαρακτηριστικών βάσει του feature importance στο Random Forest, και τα αποτελέσματα δείχνουν ότι η μείωση των χαρακτηριστικών με αυτό το κριτήριο οδηγεί σε μείωση της ακρίβειας του μοντέλου. Αυτό δείχνει ότι η ποικιλία των χαρακτηριστικών είναι πιο σημαντική από την απόλυτη σημασία τους για τη μοντελοποίηση του προβλήματος. Αντί να εξαρτώνται από λίγα μόνο ισχυρά χαρακτηριστικά, τα δεδομένα φαίνεται να ωφελούνται από μια μεγαλύτερη γκάμα πληροφοριών. Επιπλέον, η ίδια ανάλυση αποκάλυψε ότι τα quantitative χαρακτηριστικά συνέβαλαν περισσότερο στην ταξινόμηση, γεγονός που ενισχύει την υπόθεση ότι στατιστικές μετρήσεις, όπως μέσες τιμές, τυπικές αποκλίσεις και εντροπία, περιέχουν ουσιαστική πληροφορία για τη διάκριση των κατηγοριών.
- Στο πείραμα 6.3, το Random Forest παρουσίασε την καλύτερη απόδοση σε σύγκριση με τα υπόλοιπα μοντέλα. Αυτό μπορεί να υποδηλώνει ότι τα δεδομένα περιέχουν πολύπλοκες σχέσεις και αλληλεπιδράσεις μεταξύ των χαρακτηριστικών, τις οποίες το Random Forest μπορεί να διαχειριστεί αποτελεσματικά λόγω της φύσης του ως ensemble μοντέλου. Συνδυάζοντας τα αποτελέσματα από πολλά ανεξάρτητα δέντρα απόφασης, το Random Forest μπορεί να απορροφήσει τη φυσική αβεβαιότητα που υπάρχει στα δεδομένα και να δημιουργήσει έναν πιο σταθερό ταξινομητή. Αντίθετα, το Logistic Regression παρουσίασε τη χαμηλότερη απόδοση, γεγονός που ενισχύει την υπόθεση ότι τα δεδομένα δεν ακολουθούν μια απλή γραμμική σχέση μεταξύ χαρακτηριστικών και ετικετών. Δεδομένου ότι το Logistic Regression υποθέτει γραμμικότητα, τα αποτελέσματά του δείχνουν ότι οι σχέσεις στα δεδομένα είναι μη γραμμικές, απαιτώντας πιο πολύπλοκες μεθόδους μοντελοποίησης.
- Τα KNN και SVM σημείωσαν ικανοποιητικές επιδόσεις, υποδεικνύοντας ότι μπορούν να λειτουργήσουν αποτελεσματικά όταν τα δεδομένα είναι λιγότερο πολύπλοκα ή όταν υπάρχουν συγκεκριμένες συσχετίσεις που μπορούν να εκμεταλλευτούν. Παρόλα αυτά, για να ανταγωνιστούν πλήρως το Random Forest, πιθανώς απαιτούν καλύτερη ρύθμιση υπερπαραμέτρων ή πιο λεπτομερή προεπεξεργασία των χαρακτηριστικών. Στο ίδιο πείραμα έγινε εμφανές ότι η διάκριση μεταξύ των ετικετών "line" και

"scatter" είναι ιδιαίτερα δύσκολη, κάτι που ενισχύεται και από τα αποτελέσματα του πειράματος 6.4. Αυτό δείχνει ότι αυτές οι δύο κατηγορίες έχουν πολύ παρόμοια χαρακτηριστικά, γεγονός που δυσκολεύει τη διάκρισή τους από τα μοντέλα ταξινόμησης.

- Το πειράμα 6.4 εισάγει την αξιολόγηση του AutoGluon και του AutoKeras, μαζί με το Random Forest και τα Neural Networks. Τα αποτελέσματα δείχνουν ότι, παρόλο που τα Random Forest, Neural Networks και AutoKeras παρουσίασαν σχετικά καλή απόδοση, δεν κατάφεραν να φτάσουν την αποδοτικότητα του AutoGluon. Το γεγονός ότι το AutoGluon ξεπέρασε τις παραδοσιακές προσεγγίσεις μπορεί να υποδηλώνει ότι, παρά την ισχυρή ικανότητα των Random Forest και Neural Networks να αποτυπώνουν πολύπλοκες σχέσεις, οι AutoML τεχνικές μπορούν να βελτιστοποιήσουν τις ρυθμίσεις τους με πιο αποδοτικό τρόπο. Η χρήση του AutoGluon φαίνεται να εκμεταλλεύεται την αυτοματοποιημένη επιλογή μοντέλων και τις κατάλληλες ρυθμίσεις υπερπαραμέτρων, κάτι που πιθανώς του επέτρεψε να προσαρμοστεί καλύτερα στις ανάγκες του dataset.
- Από όλα τα πειράματα προκύπτει ότι το Random Forest είναι η καλύτερη επιλογή μεταξύ των κλασικών αλγορίθμων, καθώς επιτυγχάνει σταθερά υψηλές επιδόσεις σε πολυδιάστατα δεδομένα με μη γραμμικές σχέσεις. Παρόλα αυτά, το AutoGluon υπερέχει σε απόδοση, κάτι που οδήγησε στην επιλογή του ως βασικού ταξινομητή για την ανάπτυξη του Web App.
- Αυτή η απόφαση βασίζεται στο γεγονός ότι το AutoGluon μπορεί να βελτιστοποιεί αυτόματα τις παραμέτρους του μοντέλου, να συνδυάζει διαφορετικές μεθόδους ταξινόμησης και να παρέχει γενικά καλύτερη ακρίβεια χωρίς την ανάγκη εκτεταμένης παραμετροποίησης από τον χρήστη. Αντίθετα, τα Neural Networks και το AutoKeras απαιτούν περισσότερη λεπτομερή διαμόρφωση για να επιτύχουν υψηλές επιδόσεις, ενώ τα κλασικά μοντέλα όπως το Random Forest, το KNN και το SVM αν και αποδεικνύονται ικανά, δεν καταφέρνουν να φτάσουν το επίπεδο απόδοσης του AutoGluon.
- Το γενικό συμπέρασμα είναι ότι τα μοντέλα ταξινόμησης αποδίδουν διαφορετικά ανάλογα με τη φύση των δεδομένων, αλλά η επιλογή ενός AutoML μοντέλου όπως το AutoGluon μπορεί να προσφέρει το καλύτερο δυνατό αποτέλεσμα χωρίς την ανάγκη εξαντλητικής βελτιστοποίησης από τον χρήστη.

7 *Επίλογος*

7.1 *Σύνοψη και συμπεράσματα*

Η επιλογή του κατάλληλου γραφικού εργαλείου αποτελεί σημαντική πρόκληση στην ανάλυση δεδομένων, καθώς επηρεάζει τόσο την κατανόηση των πληροφοριών από τον αναλυτή όσο και την αποτελεσματικότητα της επικοινωνίας με το κοινό. Στην παρούσα διπλωματική εργασία, προτείνεται μια αυτοματοποιημένη λύση για την επιλογή της κατάλληλης οπτικοποίησης στην αναλυτική δεδομένων, βασισμένη σε δεδομένα από το Plotly Community Feed, μια πλατφόρμα όπου οι χρήστες μοιράζονται τις οπτικοποιήσεις τους.

Η προσέγγιση που αναπτύχθηκε περιλαμβάνει τη διαδικασία εξαγωγής χαρακτηριστικών, την προ-επεξεργασία των δεδομένων, την αναζήτηση του καταλληλότερου ταξινομητή, την αξιολόγηση των αποτελεσμάτων και την ανάπτυξη της web εφαρμογής. Στη φάση της εξαγωγής χαρακτηριστικών, τα δεδομένα αναλύθηκαν και χαρτογραφήθηκαν σε 83 χαρακτηριστικά, αποτελούμενα από ατομικά χαρακτηριστικά για κάθε στήλη καθώς και συνδυαστικά χαρακτηριστικά μεταξύ δύο στηλών. Η προεπεξεργασία περιλάμβανε τεχνικές όπως αποκοπή ακραίων τιμών, one-hot encoding, κανονικοποίηση και αντικατάσταση κενών τιμών, ώστε να διασφαλιστεί η ομοιογένεια των δεδομένων.

Για την ταξινόμηση, εξετάστηκαν διάφοροι κλασικοί και σύγχρονοι αλγόριθμοι μηχανικής μάθησης, συμπεριλαμβανομένων των Random Forest, K-Nearest Neighbors, Support Vector Machines, Logistic Regression, Neural Networks και AutoML εργαλείων όπως το AutoGluon και το AutoKeras. Έγιναν δοκιμές για την εύρεση βέλτιστων υπερπαραμέτρων, ενώ η ταξινόμηση υλοποιήθηκε με one-vs-one στρατηγική, όπου εκπαιδεύτηκαν ξεχωριστά μοντέλα για κάθε συνδυασμό ετικετών, επιτρέποντας μεγαλύτερη προσαρμοστικότητα στις απαιτήσεις της εφαρμογής.

Ως τελική λύση, αναπτύχθηκαν 11 διαφορετικά μοντέλα, έξι για δυαδική ταξινόμηση, τέσσερα για τριπλή ταξινόμηση και ένα για το σύνολο των ετικετών. Η αξιολόγηση των

ταξινομητών έδειξε ότι το AutoGluon παρουσίασε την καλύτερη απόδοση, καθιστώντας το την ιδανική επιλογή για την ανάπτυξη της εφαρμογής. Παράλληλα, η σημασία των χαρακτηριστικών αξιολογήθηκε μέσω του Feature Importance του Random Forest, παρέχοντας πολύτιμες πληροφορίες για τη συνεισφορά κάθε χαρακτηριστικού στην ταξινόμηση.

Το τελευταίο στάδιο της εργασίας περιλάμβανε την ανάπτυξη του Web App, το οποίο επιτρέπει τη χρήση του ταξινομητή χωρίς την ανάγκη γραφής κώδικα, καθιστώντας την επιλογή οπτικοποιήσεων προσβάσιμη και εύχρηστη για όλους.

Η ολοκλήρωση της παρούσας μελέτης ανέδειξε ότι η επιλογή γραφικής αναπαράστασης δεδομένων δεν είναι μια ευθύγραμμη διαδικασία. Αν και η προσέγγιση που αναπτύχθηκε καλύπτει ένα συγκεκριμένο μέρος των ζητημάτων που σχετίζονται με την οπτικοποίηση, υπάρχουν περιθώρια βελτίωσης. Οι datetime στήλες αντιμετωπίστηκαν ως κατηγορικές, αν και θα μπορούσαν να αξιοποιηθούν επιπλέον χαρακτηριστικά για την ανάλυσή τους. Επιπλέον, το Feature Importance εμφάνισε χαμηλές τιμές συνολικά, γεγονός που υποδηλώνει ότι τα χαρακτηριστικά δεν διαχωρίζουν απόλυτα τις κατηγορίες.

Παρότι οι κλασικές μέθοδοι παρουσίασαν αρχικά ικανοποιητικά αποτελέσματα, αποδείχθηκε ότι τα σύγχρονα AutoML εργαλεία, όπως το AutoGluon, υπερέχουν τόσο σε ακρίβεια όσο και σε ευκολία χρήσης, καθιστώντας τα την πλέον κατάλληλη επιλογή για τέτοιου είδους εφαρμογές.

7.2 *Μελλοντικές επεκτάσεις*

Η Αυτόματη Οπτικοποίηση Αναλυτικής Δεδομένων αποτελεί ένα αναδυόμενο και ανεξερεύνητο πεδίο με πολλές προοπτικές εξέλιξης. Με την παρούσα έρευνα επιχειρήθηκε να τεθούν τα θεμέλια για την περαιτέρω ανάπτυξη του τομέα, ωστόσο υπάρχουν αρκετές βελτιώσεις και προτάσεις που μπορούν να εφαρμοστούν στο πρόβλημα.

Ένα από τα βασικά ζητήματα που αναδείχθηκαν κατά την ανάπτυξη της μεθόδου ήταν η διαχείριση δεδομένων με περισσότερες διαστάσεις. Η μελέτη επικεντρώθηκε σε δισδιάστατα δεδομένα και στην ανάπτυξη χαρακτηριστικών που αφορούν αποκλειστικά 2D γραφικές παραστάσεις, όπως scatter, bar, line και pie charts. Η εισαγωγή επιπλέον διαστάσεων θα μπορούσε να επιτρέψει τη χρήση περισσότερων τύπων οπτικοποιήσεων, καλύπτοντας ένα ευρύτερο φάσμα αναγκών. Δεδομένα με μία μόνο μεταβλητή θα μπορούσαν να αναπαρασταθούν μέσω dot plots και histograms, ενώ δεδομένα με τρεις μεταβλητές θα μπορούσαν να αξιοποιήσουν 3D scatter plots, 3D bar charts, volume rendering ή heatmaps.

Σε ακόμη περισσότερες διαστάσεις, θα ήταν δυνατή η χρήση animated 3D visualizations, όπου ο χρόνος θα μπορούσε να λειτουργήσει ως τέταρτη διάσταση, ή 4D scatter plots, όπου η επιπλέον διάσταση θα μπορούσε να εκφραστεί μέσω χρώματος, μεγέθους ή διαφάνειας.

Μια ακόμη σημαντική βελτίωση αφορά τη μετάβαση από τη one-vs-one στρατηγική που υιοθετήθηκε στην παρούσα έρευνα, προς μια multiclass προσέγγιση. Καθώς ο αριθμός των ετικετών αυξάνεται, η one-vs-one στρατηγική απαιτεί τη δημιουργία και εκπαίδευση μεγάλου αριθμού μοντέλων, καθιστώντας τη διαδικασία χρονοβόρα και πολύπλοκη. Αντίθετα, ένα multiclass ταξινομητικό μοντέλο μπορεί να εκπαιδεύεται σε όλες τις κλάσεις ταυτόχρονα, μειώνοντας τον αριθμό των απαιτούμενων μοντέλων και επιτρέποντας καλύτερη γενίκευση σε άγνωστες περιπτώσεις. Παράλληλα, η εκπαίδευση σε ένα ενιαίο σύνολο δεδομένων μειώνει τον κίνδυνο overfitting, προσφέροντας μια πιο αποδοτική λύση.

Ένας ακόμη περιοριστικός παράγοντας της παρούσας μελέτης ήταν το σύνολο δεδομένων από το Plotly Community Feed. Αν και το dataset ήταν επαρκές για τους σκοπούς της έρευνας, η παρουσία αντιγράφων γραφικών παραστάσεων ή δεδομένων που δημιουργήθηκαν από bots επηρέασε την ποιότητα των αποτελεσμάτων. Η χρήση τέτοιων δεδομένων ενδέχεται να δημιουργήσει biased μοντέλα, τα οποία δεν θα μπορούν να γενικεύσουν ικανοποιητικά σε πραγματικά σενάρια. Για μελλοντικές επεκτάσεις, ένα πιο ποικιλόμορφο και μεγαλύτερο dataset θα μπορούσε να βελτιώσει την απόδοση του μοντέλου και να εξασφαλίσει μεγαλύτερη αξιοπιστία στις προτεινόμενες οπτικοποιήσεις.

Μια επιπλέον κατεύθυνση για μελλοντική εξέλιξη είναι η ενσωμάτωση Explainability στη διαδικασία ταξινόμησης. Ένα επεξηγηματικό σύστημα θα επέτρεπε στο μοντέλο να εξηγεί με σαφήνεια γιατί προτείνει μια συγκεκριμένη οπτικοποίηση και ποιοι παράγοντες επηρέασαν αυτήν την απόφαση. Για παράδειγμα, αν το σύστημα επιλέξει ένα scatter plot, θα μπορούσε να εξηγήσει ότι η επιλογή αυτή οφείλεται στην παρουσία ακραίων τιμών στο dataset. Επιπλέον, η χρήση τεχνικών όπως association rules ή decision trees θα μπορούσε να αναδείξει σημαντικά μοτίβα, τάσεις και ανωμαλίες στα δεδομένα, βελτιώνοντας την κατανόηση των προτεινόμενων επιλογών. Μια ακόμη προοπτική είναι η αυτόματη δημιουργία περιλήψεων που θα συνοδεύουν τις οπτικοποιήσεις, παρέχοντας στον χρήστη μια σαφή και κατανοητή ερμηνεία των αποτελεσμάτων.

Οι παραπάνω βελτιώσεις μπορούν να επεκτείνουν σημαντικά το πεδίο της Αυτόματης Οπτικοποίησης Αναλυτικής Δεδομένων, καθιστώντας τα συστήματα πιο ευέλικτα, αποδοτικά και κατανοητά για τον χρήστη.

8 *Βιβλιογραφία*

[1] M. Tory and T. Moller, "Rethinking Visualization: A High-Level Taxonomy," IEEE Symposium on Information Visualization, Austin, TX, USA, 2004, pp. 151-158, doi: 10.1109/INFVIS.2004.59.

[2] Biau, G., Scornet, E. A random forest guided tour. TEST 25, 197–227 (2016). https://doi.org/10.1007/s11749-016-0481-7

[3] Q. Wang, C. Zhu-Tian, Y. Wang and H. Qu, "A Survey on ML4VIS: Applying Machine Learning Advances to Data Visualization," in IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 12, pp. 5134-5153, 1 Dec. 2022, doi: 10.1109/TVCG.2021.3106142.

[4] Haifeng Jin, Qingquan Song, and Xia Hu. 2019. Auto-Keras: An Efficient Neural Architecture Search System. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). Association for Computing Machinery, New York, NY, USA, 1946–1956. https://doi.org/10.1145/3292500.3330648

[5] Erickson, Nick, et al. "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data." arXiv preprint arXiv:2003.06505 (2020).

[6] Karmaker, Shubhra Kanti, Md Mahadi Hassan, Micah J. Smith, Lei Xu, Chengxiang Zhai, and Kalyan Veeramachaneni. "Automl to date and beyond: Challenges and opportunities." ACM Computing Surveys (CSUR) 54, no. 8 (2021): 1-36.

[7] He, Xin, Kaiyong Zhao, and Xiaowen Chu. "AutoML: A survey of the state-of-the-art." Knowledge-based systems 212 (2021): 106622.

[8] Luo, Yuyu, Xuedi Qin, Nan Tang, and Guoliang Li. "Deepeye: Towards automatic data visualization." In 2018 IEEE 34th international conference on data engineering (ICDE), pp. 101-112. IEEE, 2018.

[9] Donders, A. Rogier T., Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. "A gentle introduction to imputation of missing values." Journal of clinical epidemiology 59, no. 10 (2006): 1087-1091.

[10] Song, Qinbao, and Martin Shepperd. "Missing data imputation techniques." International journal of business intelligence and data mining 2, no. 3 (2007): 261-291.

[11] Cover, Thomas, and Peter Hart. "Nearest neighbor pattern classification." IEEE transactions on information theory 13, no. 1 (1967): 21-27.

[12] Cortes, Corinna. "Support-Vector Networks." *Machine Learning* (1995).

[13] Ho, Tin Kam. "Random decision forests." In Proceedings of 3rd international conference on document analysis and recognition, vol. 1, pp. 278-282. IEEE, 1995.

[14] Verdonck, Tim, Bart Baesens, María Óskarsdóttir, and Seppe vanden Broucke. "Special issue on feature engineering editorial." Machine learning 113, no. 7 (2024): 3917-3928.

[15] Hu, Kevin, Michiel A. Bakker, Stephen Li, Tim Kraska, and César Hidalgo. "Vizml: A machine learning approach to visualization recommendation." In Proceedings of the 2019 CHI conference on human factors in computing systems, pp. 1-12. 2019.

[16] Unwin, Antony. "Why is data visualization important? what is important in data visualization?." Harvard Data Science Review 2, no. 1 (2020): 1.

[17] Pathak, Shreyans, and Shashwat Pathak. "Data visualization techniques, model and taxonomy." Data Visualization and Knowledge Engineering: Spotting Data Points with Artificial Intelligence (2020): 249-271.

[18] Wongsuphasawat, Kanit, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. "Towards a general-purpose query language for visualization recommendation." In *Proceedings of the workshop on human-in-the-loop data analytics*, pp. 1-6. 2016.

[19] Wongsuphasawat, Kanit, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. "Voyager: Exploratory analysis via faceted browsing of visualization recommendations." *IEEE transactions on visualization and computer graphics* 22, no. 1 (2015): 649-658.

[20] Hu, Kevin, Diana Orghian, and César Hidalgo. "DIVE: A mixed-initiative system supporting integrated data exploration workflows." In *Proceedings of the workshop on human-in-the-loop data analytics*, pp. 1-7. 2018.

[21] Mackinlay, Jock, Pat Hanrahan, and Chris Stolte. "Show me: Automatic presentation for visual analysis." *IEEE transactions on visualization and computer graphics* 13, no. 6 (2007): 1137-1144.

[22] Choo, Jaegul, and Shixia Liu. "Visual analytics for explainable deep learning." *IEEE computer graphics and applications* 38, no. 4 (2018): 84-92.