ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

AutoDiscretizer
Διαδικτυακή εφαρμογή διακριτοποίησης δεδομένων με αυτόματη επιλογή μεθόδου

Φοιτητής:
Παυλίδης Δημήτριος
Student ID: 144242

Επιβλέπων:
Ουγιάρογλου Στέφανος
Επ. Καθηγητής

20 Ιανουαρίου 2024

Τίτλος Π.Ε. Διαδικτυακή εφαρμογή διακριτοποίησης δεδομένων
με αυτόματη επιλογή μεθόδου
Κωδικός Π.Ε. 23311
Ονοματεπώνυμο φοιτητή Παυλίδης Δημήτριος
Ονοματεπώνυμο εισηγητή Ουγιάρογλου Στέφανος
Ημερομηνία ανάληψης Π.Ε. 03-11-2023
Ημερομηνία περάτωσης Π.Ε. 28-01-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Αφιέρωση

Θα ήθελα να ευχαριστήσω τον κ. Ουγιάρογλου Στέφανο, για την ανεκτίμητη βοήθεια και την υποστήριξη που μου παρείχε κατά τη διάρκεια της εκπόνησης αυτής της πτυχιακής εργασίας. Επιπλέον, θα ήθελα να εκφράσω την ευγνωμοσύνη μου προς την οικογένειά μου, για τη σταθερή τους υποστήριξη και ενθάρρυνση καθ' όλη τη διάρκεια αυτής της πρόκλησης.

Πρόλογος

Η επιλογή του θέματος για την παρούσα εργασία είναι βασισμένη σε συγκεκριμένους λόγους που συνδυάζουν τόσο την προσωπική μου εμπειρία όσο και τις επιδιώξεις μου για μάθηση και εξέλιξη. Αρχικά, η εμπειρία μου στην ανάπτυξη διαδικτυακών εφαρμογών με οδήγησε στην απόφαση να εξερευνήσω βαθύτερα το πεδίο αυτό. Η άνεση και η εξοικείωση με τη διαδικασία ανάπτυξης εφαρμογών, μου παρέχουν ένα στέρεο θεμέλιο για την περαιτέρω εξερεύνηση και καινοτομία. Παράλληλα, η επιθυμία μου να αναπτύξω περαιτέρω τις γνώσεις μου στον τομέα της μηχανικής μάθησης και της επεξεργασίας δεδομένων αποτελεί έναν ακόμη κρίσιμο λόγο για την επιλογή αυτής της θεματικής. Η τεχνολογία, εξελίσσεται με ταχύτατους ρυθμούς και η κατανόηση των προηγμένων τεχνικών στην επεξεργασία δεδομένων, έχει γίνει πλέον απαραίτητη για την ανάπτυξη αποδοτικών λύσεων.

Περίληψη

Στην ανάλυση δεδομένων, η διακριτοποίηση είναι ένας θεμελιώδης μετασχηματισμός, μέσω του οποίου συνεχή χαρακτηριστικά μετατρέπονται σε διακριτές κατηγορίες. Αυτή η διαδικασία, βρίσκει εφαρμογή σε πολλούς τομείς όπου είναι απαραίτητη η απλοποίηση των δεδομένων για ευκολότερη ερμηνεία και ανάλυση. Πέρα από την απλοποίηση, η διακριτοποίηση μπορεί επίσης να ενισχύσει την απόδοση μοντέλων μηχανικής μάθησης σε ορισμένα σύνολα δεδομένων. Στην πρακτική εφαρμογή, η διακριτοποίηση εμπλέκει τη διαίρεση συνεχών τιμών σε ομάδες ή διαστήματα. Για παράδειγμα, μια συνεχής μεταβλητή όπως η θερμοκρασία μπορεί να διακριτοποιηθεί σε κατηγορίες όπως χαμηλή, μέση, υψηλή. Κάθε διάστημα τιμών αντιστοιχίζεται σε μια συγκεκριμένη κατηγορία, καθιστώντας την επεξεργασία και την ανάλυση των δεδομένων πιο διαχειρίσιμη και ερμηνεύσιμη. Η έρευνά μας, μας αποκάλυψε, ότι υπάρχει έλλειψη διαθέσιμων διαδικτυακών εφαρμογών για τη διακριτοποίηση δεδομένων. Στόχος της πτυχιακής εργασίας, είναι η ανάπτυξη της διαδικτυακής εφαρμογής AutoDiscretizer, όπου ο χρήστης θα μπορεί να ανεβάζει το σύνολο δεδομένων που περιέχει αριθμητικά γνωρίσματα, θα του δίνει τη δυνατότητα να επιλέγει τα γνωρίσματα που επιθυμεί να μετατραπούν σε κατηγορικά μέσω διακριτοποίησης, θα επιλέγει τη μέθοδο διακριτοποίησης που επιθυμεί, θα εκτελεί τη διακριτοποίηση και θα λαμβάνει το σύνολο δεδομένων μετά την προεπεξεργασία που θα έχει υποστεί. Επιπλέον, η εφαρμογή θα διαθέτει έναν μηχανισμό αυτόματης επιλογής της καταλληλότερης για τα συγκεκριμένα δεδομένα μεθόδου μέσω εκπαίδευσης και αξιολόγησης μοντέλων μηχανικής μάθησης.

Abstract

In the field of data processing, discretization is a fundamental transformation that converts continuous features into discrete categories. This process finds application in many areas where simplification of data for easier interpretation and analysis is necessary. Beyond simplification, discretization can also enhance the performance of machine learning models in certain data sets. In practical application, discretization involves dividing continuous values into groups or intervals. For example, a continuous variable such as temperature can be discretized into categories like low, medium, high. Each range of values is associated with a specific category, making the processing and analysis of data more manageable and interpretable. Our research revealed that there is a lack of available online applications for data discretization. The goal of this thesis is to develop an online application where the user can upload a dataset containing numerical attributes, will have the option to select the attributes they wish to convert to categorical through discretization, choose the desired method of discretization, execute the discretization, and receive the dataset after the preprocessing it has undergone. Additionally, the application will feature an automatic mechanism for selecting the most suitable method for the specific data through the training and evaluation of machine learning models.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

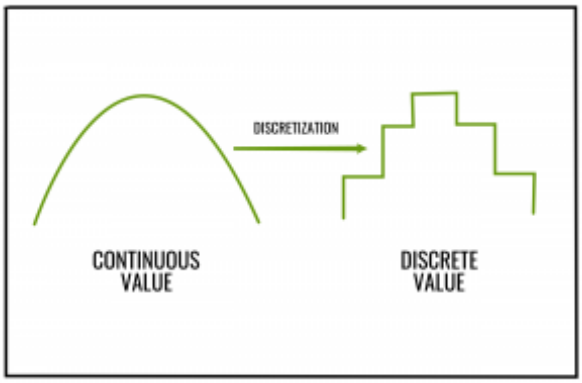
1.1 Διακριτοποίηση δεδομένων

Η σημασία της αποτελεσματικής διαχείρισης και επεξεργασίας των δεδομένων είναι κρίσιμη στη σημερινή ψηφιακή εποχή, καθώς αφορά τη συλλογή, την οργάνωση, την προστασία και την αποθήκευση δεδομένων, ώστε να μπορούν να αναλυθούν για τη λήψη επιχειρηματικών αποφάσεων. Οι οργανισμοί, δημιουργούν και καταναλώνουν δεδομένα με πρωτοφανείς ρυθμούς, ενώ οι λύσεις διαχείρισης δεδομένων γίνονται ουσιώδεις για την κατανόηση και εκμετάλλευση αυτών των απεριόριστων όγκων πληροφορίας [1]. Η ανάλυση δεδομένων, είναι απαραίτητη στον επιχειρηματικό κόσμο, επειδή επιτρέπει στην ηγεσία να δημιουργήσει στρατηγικές βασισμένες σε στοιχεία, να κατανοήσει καλύτερα τους πελάτες για να στοχεύσει πιο αποτελεσματικά τις ενέργειες μάρκετινγκ, και να αυξήσει τη συνολική παραγωγικότητα. Οι εταιρείες, που εκμεταλλεύονται την ανάλυση δεδομένων, απολαμβάνουν ανταγωνιστικό πλεονέκτημα επειδή μπορούν να κάνουν γρηγορότερες αλλαγές που αυξάνουν τα έσοδα, μειώνουν τα κόστη και προωθούν την καινοτομία [2].

Η διακριτοποίηση, είναι μια διαδικασία επεξεργασίας δεδομέμων, η οποία μετατρέπει αριθμητικά χαρακτηριστικά σε κατηγορικά [3], βελτιώνοντας έτσι την ανάλυση και ερμηνεία των δεδομένων. Στην πράξη, αυτό συνήθως σημαίνει την κατηγοριοποίηση ενός συνεχούς εύρους τιμών σε περιορισμένο αριθμό τμημάτων ή ομάδων. Τα δεδομένα συνήθως παρουσιάζονται σε μικτή μορφή: ονομαστικά, διακριτά, και/ή συνεχή. Τα διακριτά και συνεχή δεδομένα είναι ταξινομικά, με μια σειρά μεταξύ των τιμών τους, ενώ οι ονομαστικές τιμές δεν διαθέτουν καμία τάξη μεταξύ τους [4]. Για παράδειγμα, η ηλικία είναι μια συνεχής μεταβλητή που μπορεί να λάβει οποιαδήποτε τιμή σε ένα εύρος. Η ηλικία μπορεί να κατηγοριοποιηθεί σε ομάδες, όπως 0-18, 19-30, 31-60, και άνω των 60. Μια σημαντική πτυχή της διακριτοποίησης είναι η επιλογή των κατάλληλων σημείων κοπής, η οποία μπορεί να επηρεάσει σημαντικά την ποιότητα των τελικών κατηγοριών και την απόδοση των αλγορίθμων μηχανικής μάθησης. Στην επεξεργασία δεδομένων, η διακριτοποίηση είναι σημαντική για διάφορους λόγους. Καταρχάς, μπορεί να βελτιώσει την απόδοση ορισμένων αλγορίθμων μηχανικής μάθησης. Αυτό συμβαίνει επειδή οι αλγόριθμοι αυτοί ενδέχεται να λειτουργούν πιο αποτελεσματικά με διακριτές παρά με συνεχείς τιμές [5]. Επιπλέον, η διακριτοποίηση κάνει τα δεδομένα πιο εύκολα στην ερμηνεία, καθώς μετατρέπει τις συνεχείς τιμές σε μια πιο απλή και κατανοητή μορφή. Αυτό είναι

ιδιαίτερα χρήσιμο στην ανάλυση δεδομένων, όπου η σαφήνεια και η ακρίβεια είναι κρίσιμης σημασίας.

Ωστόσο, η διακριτοποίηση δεδομένων αντιμετωπίζει μια σειρά από προκλήσεις και περιορισμούς. Κατά τη διαδικασία μετατροπής συνεχών μεταβλητών σε διακριτές τιμές, ο προσδιορισμός των ορίων ή των κατωφλιών που ορίζουν τα διαστήματα στα οποία θα ταξινομηθούν οι συνεχείς τιμές είναι μια κρίσιμη πρόκληση. Υπάρχουν διάφορες μέθοδοι διακριτοποίησης, όπως η ομοιόμορφη διακριτοποίηση, η ποσοστιαία διακριτοποίηση και η διακριτοποίηση βάσει k-means, καθεμία με τα πλεονεκτήματα και τις ελλείψεις της. Η επιλογή της κατάλληλης μεθόδου διακριτοποίησης είναι σημαντική, καθώς επηρεάζει άμεσα την ακρίβεια και την εγκυρότητα των αναλυτικών μοντέλων [6].



Σχήμα 1.1: Διακριτοποίηση δεδομένων

1.2 Automated Machine Learning (AutoML)

Η φράση “Θα ήθελα να χρησιμοποιήσω τη μηχανική μάθηση, αλλά δεν μπορώ να επενδύσω πολύ χρόνο“ είναι κάτι που ακούμε συχνά στη βιομηχανία και από ερευνητές σε άλλες επιστημονικές περιοχές [7]. Το Automated Machine Learning (AutoML), είναι ένα από τα πιο κομβικά εργαλεία στον τομέα της μηχανικής μάθησης. Το AutoML αποσκοπεί στην απλοποίηση της διαδικασίας ανάπτυξης μοντέλων μηχανικής μάθησης, κάνοντας την προσβάσιμη ακόμη και σε μη ειδικούς, μέσω της αυτοματοποίησης κρίσιμων βημάτων όπως η επιλογή αλγορίθμων, η βελτιστοποίηση παραμέτρων και η επεξεργασία δεδομένων.

Η ιστορική του εξέλιξη, αντανακλά μια σημαντική μετάβαση από τις παραδοσιακές μεθόδους στην αυτοματοποιημένη διαδικασία, επιτρέποντας τις επιχειρήσεις και τους ερευνητές να εστιάζουν περισσότερο στην ερμηνεία και την εφαρμογή των αποτελεσμάτων παρά στην επίπονη διαδικασία δημιουργίας των μοντέλων [8]. Ωστόσο, παρά την εξέλιξη του, το AutoML παραμένει ένα πεδίο με πολλές προκλήσεις, κυρίως στην επίτευξη ισορροπίας μεταξύ αυτοματοποίησης και προσαρμοστικότητας σε ποικίλες συνθήκες δεδομένων.

Στην καρδιά του AutoML, βρίσκονται οι αρχές που αποσκοπούν στην ελαχιστοποίηση της ανθρώπινης παρέμβασης και στην αυξημένη αυτοματοποίηση της διαδικασίας της μηχανικής μάθησης. Οι βασικές αυτές αρχές περιλαμβάνουν:

- Βελτιστοποίηση Υπερπαραμέτρων (Hyperparameter Optimization): Η βελτιστοποίηση υπερπαραμέτρων είναι η διαδικασία εύρεσης των βέλτιστων τιμών για τις παραμέτρους ενός μοντέλου μηχανικής μάθησης. Οι υπερπαράμετροι είναι σημαντικοί για την απόδοση του μοντέλου, και η αυτοματοποίηση αυτής της διαδικασίας εξοικονομεί χρόνο και προσπάθεια. Οι μέθοδοι περιλαμβάνουν την Grid Search, τη Random Search και τη Bayesian Optimization [9].

- Πλατφόρμες AutoML (AutoML Frameworks): Οι πλατφόρμες AutoML είναι εργαλεία που παρέχουν έτοιμες λύσεις για πολλές εργασίες μηχανικής μάθησης. Παρέχουν αυτόματη επιλογή μοντέλων, βελτιστοποίηση υπερπαραμέτρων και αυτόματη παραγωγή κώδικα για εκπαίδευση μοντέλων [10].

- Βελτιστοποίηση Χαρακτηριστικών (Feature Engineering): Η βελτιστοποίηση χαρακτηριστικών επικεντρώνεται στην επιλογή και μετασχηματισμό των χαρακτηριστικών των δεδομένων για να βελτιώσει την απόδοση του μοντέλου. Αυτόματες διαδικασίες μπορούν να αναγνωρίσουν τα σημαντικά χαρακτηριστικά και να εφαρμόσουν αυτόματα μετασχηματισμούς [11].

- Επιλογή Μοντέλου (Model Selection): Αυτόματες μεθόδους επιλέγουν το καλύτερο μοντέλο μηχανικής μάθησης για ένα δεδομένο πρόβλημα. Αξιολογούν διάφορα μοντέλα και επιλέγουν αυτό με την υψηλότερη απόδοση [12].

- Μέθοδοι Ensemble (Ensemble Methods): Οι μέθοδοι ensemble συνδυάζουν προβλέψεις από πολλά μοντέλα για να βελτιώσουν την ακρίβεια και την αξιοπιστία των προβλέψεων [13].

- Αυτόματη Προεπεξεργασία Δεδομένων (Automated Data Preprocessing): Η αυτόματη προεπεξεργασία δεδομένων αναλαμβάνει τη διαχείριση και την επεξεργασία των δεδομένων, συμπεριλαμβανομένης της αντιμετώπισης απουσιάζουσων τιμών, της κωδικοποίησης χαρακτηριστικών και της κλιμάκωσης [14].

- Αυτόματες Παραγωγές Μηχανικής Μάθησης (Automatic Machine Learning Pipelines): Οι αυτόματες παραγωγές μηχανικής μάθησης δημιουργούν πλήρεις διαδικασίες από την προεπεξεργασία των δεδομένων έως την αξιολόγηση των μοντέλων και την εφαρμογή τους στην πράξη [15].

- AutoML για Ανάλυση Χρονοσειρών (AutoML for Time Series Analysis): Ειδικές μέθοδοι AutoML αφορούν την αυτόματη επιλογή μοντέλων και υπερπαραμέτρων για προβλήματα χρονοσειρών [16].

Κάθε μία από αυτές τις μεθόδους, στοχεύει στην απλοποίηση και αυτοματοποίηση συγκεκριμένων ετερόκλητων καθηκόντων στη διαδικασία ανάπτυξης μοντέλων μηχανικής μάθησης, επιτρέποντας στους χρήστες να επικεντρώνονται στην ερμηνεία των αποτελεσμάτων και στην

εφαρμογή των μοντέλων σε πραγματικά προβλήματα.

Επιπλέον, ενα βασικό στοιχείο του AutoML είναι η αυτόματη αξιολόγηση και σύγκριση των διαφόρων μοντέλων μηχανικής μάθησης. Κατά τη διάρκεια αυτής της διαδικασίας, το AutoML αξιολογεί την απόδοση των μοντέλων σε ένα σύνολο δεδομένων ελέγχου χρησιμοποιώντας διάφορες μετρικές, όπως η ακρίβεια, η ανάκληση και άλλες, ανάλογα με τον τύπο του προβλήματος. Οι πληροφορίες από αυτή τη διαδικασία αξιοποιούνται στη συνέχεια για να επιλεγεί το καλύτερο μοντέλο για το συγκεκριμένο πρόβλημα. Αυτό επιτρέπει στους ερευνητές και τους επαγγελματίες της μηχανικής μάθησης να επιλέξουν την καλύτερη λύση χωρίς την ανάγκη να διαθέτουν ειδικές γνώσεις στη στατιστική ή τη μηχανική μάθηση. Επίσης, η αυτόματη σύγκριση μοντέλων μπορεί να βοηθήσει στην ανακάλυψη τυχόν προβλημάτων ή αδυναμιών στα μοντέλα, επιτρέποντας την περαιτέρω βελτίωσή τους.

Μετά την αυτόματη αξιολόγηση και σύγκριση των μοντέλων, το AutoML παρέχει τη δυνατότητα αυτόματης εξαγωγής αποτελεσμάτων και δημιουργίας αναφορών. Αυτό είναι σημαντικό για τη διαφάνεια και την κατανόηση των αποτελεσμάτων της αυτοματοποιημένης διαδικασίας. Οι αναφορές αυτές μπορούν να περιλαμβάνουν γραφήματα, πίνακες και περιγραφές της απόδοσης των μοντέλων σε διάφορες μετρικές. Επίσης, μπορούν να περιλαμβάνουν συστάσεις για την επιλογή του καλύτερου μοντέλου και τη βελτίωση των αποτελεσμάτων. Αυτή η αυτόματη δημιουργία αναφορών εξοικονομεί χρόνο και διευκολύνει τη συνεργασία μεταξύ των επαγγελματιών δεδομένων και επιστημόνων των δεδομένων.

Επιπρόσθετα, το AutoML μπορεί να προσφέρει εξειδικευμένες λύσεις για συγκεκριμένα θέματα. Για παράδειγμα, στην ανάλυση εικόνων, το AutoML μπορεί να περιλαμβάνει αυτόματη εξαγωγή χαρακτηριστικών από εικόνες και αυτόματη επιλογή μοντέλων με βάση την ανίχνευση αντικειμένων. Σε προβλήματα ανάλυσης χρονοσειρών, το AutoML μπορεί να προσφέρει εξειδικευμένες μεθόδους για την αυτόματη επιλογή μοντέλων και υπερπαραμέτρων που λειτουργούν καλύτερα για χρονοσειρές δεδομένων [17].

Τέλος, το Automated Machine Learning (AutoML) έχει καταστεί κρίσιμο εργαλείο διότι συνεισφέρει σε διάφορους τομείς, όπως:

1. Εξοικονόμηση Χρόνου: Επιτρέπει στους επαγγελματίες να αναπτύσσουν γρήγορα μοντέλα μηχανικής μάθησης χωρίς την ανάγκη για εκτεταμένη εμπειρία στην περιοχή, εξοικονομώντας χρόνο και πόρους.

2. Εύκολη Πρόσβαση: Με το AutoML, ακόμη και άτομα χωρίς ειδίκευση στη μηχανική μάθηση μπορούν να δημιουργήσουν μοντέλα, δημιουργώντας ένα περιβάλλον που είναι πιο προσιτό.

3. Βελτιστοποίηση Απόδοσης: Εκτελεί αυτόματη βελτιστοποίηση υπερπαραμέτρων και επιλογή μοντέλων, βελτιώνοντας την απόδοση των μοντέλων.

4. Ευρεία Εφαρμογή: Το AutoML εφαρμόζεται σε πολλούς τομείς, όπως η ανάλυση δεδομένων, η ρομποτική, η υγεία, η χρηματοοικονομία, η εκπαίδευση και άλλους.

Ορισμένες βιβλιοθήκες και εργαλεία που σχετίζονται με το AutoML και μπορεί να χρησιμοποιηθούν για αυτές τις εφαρμογές περιλαμβάνουν:

- Auto-Sklearn: Μια αυτόματη βιβλιοθήκη για το Scikit-Learn που προσφέρει εργαλεία για την αυτόματη επιλογή μοντέλων και βελτιστοποίηση υπερπαραμέτρων [18].
- TPOT (Tree-Based Pipeline Optimization Tool): Ένα εργαλείο που χρησιμοποιεί γενετικούς αλγορίθμους για την αναζήτηση και βελτιστοποίηση αλγορίθμων προεπεξεργασίας και μοντέλων [19].
- H2O.ai: Μια πλατφόρμα AutoML που παρέχει εργαλεία για αυτόματη επιλογή μοντέλων, βελτιστοποίηση υπερπαραμέτρων και διαχείριση δεδομένων [20].
- Google AutoML: Μια υπηρεσία του Google Cloud που προσφέρει αυτόματη επιλογή και εκπαίδευση μοντέλων για διάφορες εφαρμογές [21].

Συνοψίζοντας, το Automated Machine Learning (AutoML) αναδεικνύεται ως μια σημαντική καινοτομία στον τομέα της μηχανικής μάθησης, προσφέροντας τη δυνατότητα ανάπτυξης αποδοτικών λύσεων. Με τη συνεχή εξέλιξη και εφαρμογή του AutoML, διαγράφεται ένα εντυπωσιακό μέλλον για την αυτοματοποίηση στον χώρο της μηχανικής μάθησης, ανοίγοντας τεράστιες προοπτικές για την περαιτέρω εξέλιξη τόσο της τεχνολογίας όσο και της κοινωνίας μας.

1.3 Κίνητρο

Τα δεδομένα αποτελούν τον θησαυρό της επιχειρηματικής και επιστημονικής κοινότητας. Οι εταιρείες, οι ερευνητικοί φορείς, οι κυβερνήσεις και οι ιδιώτες χρησιμοποιούν τα δεδομένα για να αντλήσουν σημαντικές πληροφορίες και να προβλέψουν τάσεις. Ωστόσο, με την αυξανόμενη ποσότητα δεδομένων που διατίθεται, προκύπτουν νέες προκλήσεις. Η ανάγκη για αποτελεσματική ανάλυση των δεδομένων είναι σημαντική. Η διακριτοποίηση δεδομένων αποτελεί μια κεντρική διαδικασία στην επεξεργασία και ανάλυση μεγάλων συνόλων δεδομένων. Η μετατροπή αριθμητικών χαρακτηριστικών σε κατηγορικά δεν είναι μόνο ένα τεχνικό εγχείρημα αλλά μια κρίσιμη διαδικασία που ενισχύει την ευκρίνεια, την κατανόηση και την εφαρμοσιμότητα των δεδομένων σε διάφορους τομείς. Η αξία αυτής της μετατροπής είναι αδιαμφισβήτητη σε πεδία όπως η μηχανική μάθηση, η στατιστική ανάλυση, και η διαχείριση βάσεων δεδομένων.

Παρόλα αυτά, οι τρέχουσες μέθοδοι διακριτοποίησης απαιτούν συχνά εκτενείς χειρωνακτικές παρεμβάσεις, ανάλυση και διαμόρφωση από τον χρήστη, καθιστώντας τη διαδικασία χρονοβόρα και συχνά απαιτητική από πλευράς πόρων. Αυτό συνεπάγει σημαντική επένδυση χρόνου και προσπάθειας, πράγμα που μπορεί να αποτελέσει αποτρεπτικό παράγοντα, ιδιαίτερα σε περιπτώσεις μεγάλου όγκου δεδομένων. Επιπλεον, για να επιτευχθεί μια αποδοτική διακριτοποίηση, απαιτούνται ειδικές γνώσεις στατιστικής, μηχανικής μάθησης και προγραμματισμού.

Αυτή η ανάγκη για εξειδικευμένη τεχνογνωσία μπορεί να αποτελέσει σημαντικό εμπόδιο για ερευνητές ή επαγγελματίες που δεν διαθέτουν τέτοιες γνώσεις, περιορίζοντας την πρόσβαση

σε αυτή την κρίσιμη διαδικασία. Επιπροσθετα, η πλειοψηφία των τρέχουσων μεθόδων διακριτοποίησης είναι στατικές και δεν είναι σχεδιασμένες να προσαρμόζονται εύκολα στις διαρκώς μεταβαλλόμενες ανάγκες των δεδομένων ή στις ειδικές απαιτήσεις κάθε εφαρμογής. Αυτό συνεπάγεται ότι οι χρήστες πρέπει συχνά να αναζητούν ή να αναπτύσσουν εναλλακτικές λύσεις, πράγμα που καθιστά τη διαδικασία ακόμη πιο περίπλοκη. Η επιλογή της ιδανικής μεθόδου για τη διακριτοποίηση ενός συγκεκριμένου συνόλου δεδομένων είναι μια περίπλοκη απόφαση. Δεδομένου ότι διαφορετικά δεδομένα απαιτούν διαφορετικές προσεγγίσεις και τεχνικές, οι χρήστες πρέπει να αναλύσουν και να κρίνουν ποια μέθοδος είναι η πλέον κατάλληλη, αυξάνοντας την πολυπλοκότητα της διαδικασίας. Η προσφορά διαδικτυακών εφαρμογών που εκτελούν αυτό το έργο είναι περιορισμένη. Οι λίγες υπάρχουσες εφαρμογές διακριτοποίησης δεδομένων είναι συχνά επι πληρωμή ή περίπλοκες στη χρήση, ενώ η πλειοψηφία τους δεν διαθέτει τη δυνατότητα αυτόματης επιλογής της πιο κατάλληλης μεθόδου. Συνεπώς, η ανάγκη για μια διαδικτυακή εφαρμογή διακριτοποίησης δεδομένων με αυτόματη επιλογή μεθόδου είναι απαραίτητη.

1.4 Συνεισφορά

Στόχος της πτυχιακής εργασίας, είναι η ανάπτυξη της διαδικτυακής εφαρμογής AutoDiscretizer, η οποία θα προσφέρει έναν σύγχρονο και εύχρηστο τρόπο για τη διακριτοποίηση δεδομένων. Η εφαρμογή αυτή θα επιτρέπει στους χρήστες να εφαρμόζουν διακριτοποίηση σε δεδομένα της επιλογής τους, διαθέτοντας παράλληλα την δυνατότητα αυτόματης επιλογής της καταλληλότερης μεθόδου διακριτοποίησης. Αυτό επιτυγχάνεται μέσω της ενσωμάτωσης προηγμένων αλγορίθμων μηχανικής μάθησης, οι οποίοι αναλύουν τα δεδομένα και προτείνουν τις βέλτιστες παραμέτρους για την αποδοτική διακριτοποίησή τους. Με αυτόν τον τρόπο, η εφαρμογή στοχεύει στη μεγιστοποίηση της απόδοσης και της ακρίβειας στη διαχείριση και επεξεργασία δεδομένων, καθιστώντας την ένα αναντικατάστατο εργαλείο για ερευνητές, αναλυτές δεδομένων και επαγγελματίες πληροφορικής. Μέσω της χρήσης της, ερευνητές και επαγγελματίες μπορούν να εστιάσουν σε πιο προηγμένες φάσεις της ανάλυσης δεδομένων, ενώ ταυτόχρονα εξασφαλίζουν την υψηλή ποιότητα των εισαγωγικών δεδομένων.

1.5 Οργάνωση εργασίας

Σε αυτήν την ενότητα, θα αναλύσουμε την οργάνωση των επτά βασικών κεφαλαίων της εργασίας. Ο στόχος μας είναι να αποκαλύψουμε τον κεντρικό σκοπό και τα βασικά σημεία που περιέχει κάθε κεφάλαιο. Στο κεφάλαιο 1, έχουμε καλύψει κάποιες βασικές εισαγωγικες έννοιες σχετικά με τη διακριτοποίηση δεδομένων και το Automated Machine Learning (AutoML).

Στο κεφάλαιο 2, θα παρουσιάσουμε μια πιο εμπεριστατωμένη ανάλυση της διακριτοποίησης, εμπλουτίζοντας την με πληθώρα παραδειγμάτων. Επιπλέον, θα εξετάσουμε διάφορες μεθόδους διακριτοποίησης, όπως η ομοιόμορφη διακριτοποίηση, η ποσοστιαία διακριτοποίηση, και η διακριτοποίηση μέσω της μεθόδου k-means. Πρόσθετα, θα αναλύσουμε τη διαδικασία διακριτοποίησης, συμπεριλαμβάνοντας τους κανόνες διακριτοποίησης.

Στο κεφάλαιο 3, θα ασχοληθούμε με την αυτόματη επιλογή παραμέτρων για τη διακριτοποίηση. Θα εξηγήσουμε ειδικότερα τη μέθοδο της αυτόματης επιλογής στρατηγικής μέσω της κατηγοριοποίησης με τον αλγόριθμο Naive Bayes, καθώς και τον αυτόματο προσδιορισμό του αριθμού των διαμερισμάτων (bins) χρησιμοποιώντας την ίδια μέθοδο. Παράλληλα, θα εισαγάγουμε την έννοια του Naive Bayes, εξηγώντας τις βασικές του αρχές και την εφαρμογή του στον τομέα της διακριτοποίησης.

Στο κεφάλαιο 4, θα παρουσιάσουμε τις τεχνολογίες που ενσωματώθηκαν στην ανάπτυξη της διαδικτυακής εφαρμογής AutoDiscretizer. Θα εστιάσουμε στις τεχνολογίες που χρησιμοποιήθηκαν τόσο στον server side όσο και στον client side, αναλύοντας επίσης τις βιβλιοθήκες και τις εξαρτήσεις (dependencies) που αξιοποιήθηκαν. Αυτό θα δώσει μια ολοκληρωμένη εικόνα του τεχνικού υπόβαθρου της εφαρμογής.

Στο κεφάλαιο 5, θα εστιάσουμε στην πλήρη ανάλυση της υλοποίησης του AutoDiscretizer. Θα διερευνήσουμε τις προδιαγραφές, την αρχιτεκτονική και τις τεχνικές του προγραμματισμού. Η ανάλυση θα περιλαμβάνει λεπτομερή εξέταση του κώδικα και των αναπτυξιακών μεθόδων που χρησιμοποιήθηκαν. Επιπρόσθετα, θα συμπεριλάβουμε πληροφορίες για το GitHub repository της εφαρμογής, παρέχοντας πρόσβαση στον κώδικα πηγής και στα σχετικά αρχεία.

Στο κεφάλαιο 6, θα παρουσιάσουμε την εφαρμογής AutoDiscretizer με έμφαση στη διεπαφή χρήστη. Θα εξηγήσουμε πώς η διεπαφή ενισχύει την αλληλεπίδραση με την εφαρμογή και προσφέρει μια ευχάριστη και εύχρηστη εμπειρία για τον χρήστη, διευκολύνοντας τη διαδικασία της διακριτοποίησης δεδομένων. Επίσης, θα αναλύσουμε τις κύριες λειτουργίες και τα χαρακτηριστικά της διεπαφής, όπως την παρουσίαση δεδομένων, τις επιλογές διακριτοποίησης και τις δυνατότητες προσαρμογής από τον χρήστη.

Στο κεφάλαιο 7, θα συνοψίσουμε τα κύρια συμπεράσματα της εργασίας και θα εξετάσουμε τις δυνατότητες για μελλοντικές επεκτάσεις και βελτιώσεις. Θα αναλύσουμε τις προκλήσεις που αντιμετωπίσαμε κατά την ανάπτυξη, τις δυνατότητες που παρουσιάστηκαν και τα βήματα για περαιτέρω εξέλιξη της εφαρμογής. Επιπλέον, θα προτείνουμε ιδέες για την ενσωμάτωση νέων λειτουργιών, τη βελτίωση της χρηστικότητας και την επέκταση των δυνατοτήτων.

Κεφάλαιο 2

Διακριτοποίηση δεδομένων

2.1 Ανάγκη για διακριτοποίηση

Στην ανάλυση δεδομενων, η κατανόηση των διαφορετικών τύπων δεδομένων είναι σημαντική. Κάθε τύπος δεδομένων έχει τις δικές του ιδιαιτερότητες και απαιτεί συγκεκριμένες μεθόδους επεξεργασίας και ανάλυσης. Συγκεκριμένα, υπάρχουν δύο βασικοί τύποι δεδομένων που συναντώνται συχνά: τα ποιοτικά και τα ποσοτικά δεδομένα, τα οποία ταξινομούνται περαιτέρω σε:

1. **Ονομαστικά Δεδομένα (Nominal data)**:
   Είναι δεδομένα που κατηγοριοποιούνται με βάση ονόματα ή ετικέτες χωρίς να υπάρχει κάποια φυσική σειρά μεταξύ τους. Παραδείγματα περιλαμβάνουν το φύλο, την εθνικότητα, ή το είδος ενός προϊόντος.

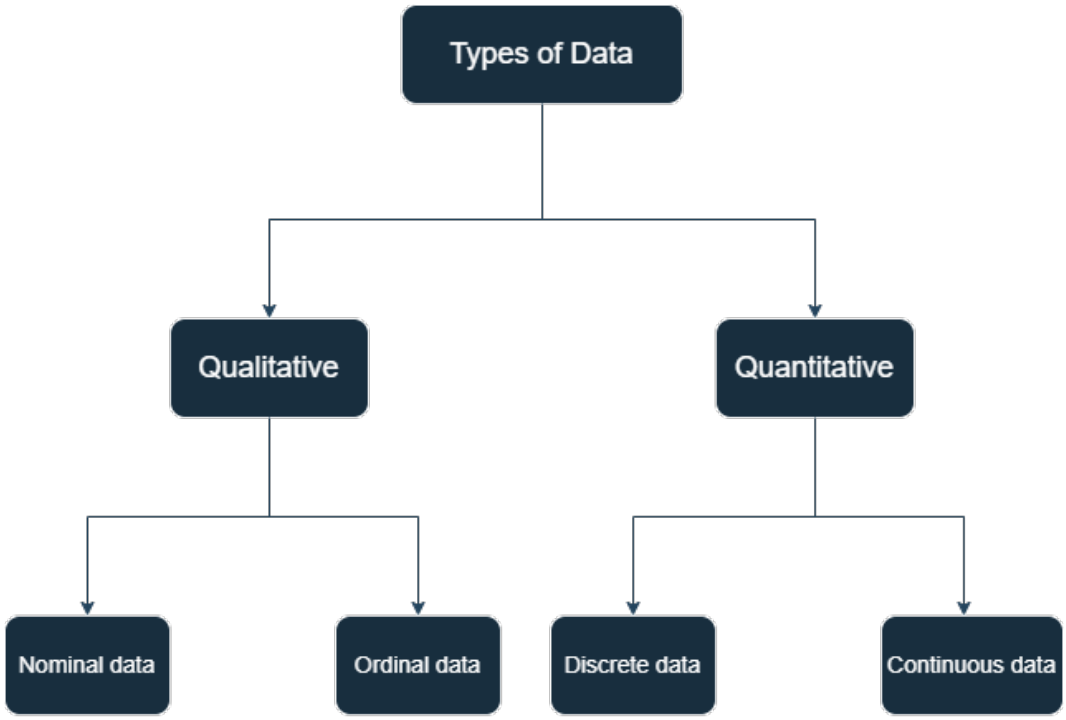
2. **Διατακτικά Δεδομένα (Ordinal data)**:
   Αυτοί οι τύποι δεδομένων έχουν μια φυσική σειρά ή ιεραρχία, αλλά δεν απέχουν ομοιόμορφα ο ένας από τον άλλο. Ένα κλασικό παράδειγμα είναι οι βαθμοί στις έρευνες ικανοποίησης (π.χ., απογοητευμένος, ουδέτερος, ικανοποιημένος).

3. **Διακριτά Δεδομένα (Discrete data)**:
   Τα διακριτά δεδομένα προκύπτουν από μετρήσεις που λαμβάνονται σε ξεχωριστές τιμές ή κατηγορίες. Για παράδειγμα, ο αριθμός των παιδιών σε μια οικογένεια ή ο αριθμός των πωλήσεων ενός προϊόντος.

4. **Συνεχή Δεδομένα (Continuous data)**:
   Αυτά είναι δεδομένα που μπορούν να λάβουν οποιαδήποτε τιμή εντός ενός συνεχούς εύρους. Τα συνεχή δεδομένα συνήθως προκύπτουν από μετρήσεις, όπως το βάρος, η θερμοκρασία ή η ταχύτητα.

Κάθε ένας από αυτούς τους τύπους δεδομένων παρέχει διαφορετικές πληροφορίες και προκαλεί διαφορετικές αναλυτικές προκλήσεις. Η κατανόηση και η σωστή χρήση αυτών των τύπων είναι κρίσιμη για την επεξεργασία και την ανάλυση δεδομένων. [22]

Στην κατηγορία των Ποιοτικών ή Κατηγορικών Δεδομένων (Qualitative or Categorical Data) εντάσσονται τα Ονομαστικά και τα Διατακτικά Δεδομένα. Αυτή η κατηγορία χαρακτηρίζεται από την έλλειψη ποσοτικών σχέσεων μεταξύ των τιμών της. Αντιθέτως, στην κατηγορία των Ποσοτικών Δεδομένων (Quantitative Data) ανήκουν τα Διακριτά και τα Συνεχή Δεδομένα, τα οποία εκφράζουν ποσότητες και επιτρέπουν αριθμητικές πράξεις.



Σχήμα 2.1: Τύποι Δεδομένων

Διαφορές μεταξύ Ονομαστικών και Διατακτικών Δεδομένων:
Τα ονομαστικά δεδομένα αφορούν κατηγορίες χωρίς φυσική σειρά ή ιεραρχία, όπως το φύλο ή η εθνικότητα. Αντίθετα, τα διατακτικά δεδομένα έχουν μια φυσική ιεραρχία ή σειρά, αλλά χωρίς ομοιόμορφη απόσταση μεταξύ των κατηγοριών, όπως βαθμοί ικανοποίησης σε έρευνες. Η κύρια διαφορά βρίσκεται στην ιεραρχική σχέση των διατακτικών δεδομένων, η οποία απουσιάζει στα ονομαστικά δεδομένα.

Διαφορά μεταξύ Διακριτών και Συνεχών Δεδομένων:
Τα διακριτά δεδομένα αποτελούνται από ξεχωριστές και διακριτές τιμές, όπως η καταμέτρηση αντικειμένων. Αντιθέτως, τα συνεχή δεδομένα μπορούν να λάβουν οποιαδήποτε τιμή εντός ενός εύρους και προέρχονται συνήθως από μετρήσεις, όπως το μήκος ή το βάρος. Η κύρια διαφορά εντοπίζεται στο εύρος των τιμών που μπορούν να πάρουν: τα διακριτά δεδομένα έχουν συγκεκριμένο αριθμό πιθανών τιμών, ενώ τα συνεχή δεδομένα είναι θεωρητικά απεριόριστα στο εύρος τιμών τους. [22]

Βάσει των πιο πρόσφατων εκτιμήσεων, καθημερινά δημιουργούνται δεδομένα που φτάνουν τα 328.77 εκατομμύρια terabytes, γεγονός που υπογραμμίζει την ασύλληπτη ποσότητα και ποικιλία των δεδομένων στη σύγχρονη εποχή [23]. Αυτή η αλματώδης αύξηση στον όγκο των δεδομένων επιβάλλει την ανάγκη για λεπτομερή επεξεργασία ώστε να επιτευχθεί βελτιωμένη κατανόηση και ερμηνεία τους. Εδώ ακριβώς εντοπίζεται η κρίσιμη σημασία της διακριτοποίησης, η οποία δρα ως καταλύτης για την απλοποίηση και την αποδοτικότερη ανάλυση των δεδομένων, καθιστώντας τα πιο διαχειρίσιμα για ενδελεχή ανάλυση.

Στην ουσία, η διακριτοποίηση μετατρέπει τα συνεχή χαρακτηριστικά σε κατηγορικά, καθιστώντας τα δεδομένα πιο προσιτά και κατανοητά για αναλύσεις και εφαρμογές μηχανικής μάθησης [4]. Για παράδειγμα, πολλά μοντέλα μηχανικής μάθησης όπως ο Naive Bayes, λειτουργούν καλύτερα με διακριτές τιμές. Η διακριτοποίηση συνεχών γνωρισμάτων μπορεί να επιταχύνει τη διαδικασία εκπαίδευσης. Επιπλέον, διευκολύνει την κατανόηση των δεδομένων και μπορεί να κατανείμει πιο ομοιόμορφα τις παραμορφωμένες τιμές. Επίσης, μειώνει την επιρροή των ακραίων τιμών, πράγμα που μπορεί να κάνει τα μοντέλα πιο ακριβή [24]. Παρακάτω αναφέρονται μερικά παραδείγματα για να κατανοήσουμε καλύτερα την ανάγκη για διακριτοποίηση:

Παράδειγμα 1:

Σενάριο: Ας υποθέσουμε ότι εργαζόμαστε σε έναν αλγόριθμο μηχανικής μάθησης Naive Bayes για να προβλέψουμε την πιθανότητα μιας συγκεκριμένης νόσου βάσει διάφορων ιατρικών αποτελεσμάτων. Μια από τις κύριες μεταβλητές είναι το επίπεδο σακχάρου στο αίμα, το οποίο είναι μια συνεχής μεταβλητή με ευρεία γκάμα τιμών.

Πρόκληση: Ο Naïve Bayes υποθέτει ότι τα χαρακτηριστικά είναι ανεξάρτητα της δεδομένης κλάσης, αλλά συχνά δυσκολεύεται με τα συνεχή δεδομένα, καθώς βασίζεται στον υπολογισμό πιθανοτήτων από τα δεδομένα εκπαίδευσης (training set). Αν τα συνεχή δεδομένα δεν διακριτοποιηθούν, το μοντέλο μπορεί να υπολογίσει εσφαλμένα τις πιθανότητες, ειδικά αν συναντήσει μια τιμή που δεν υπάρχει στο σετ εκπαίδευσης.

Λύση:
Βήμα 1: Διαχωρισμός του συνεχούς επιπέδου σακχάρου στο αίμα σε διακριτές κατηγορίες.
Για παράδειγμα:
Χαμηλό: Επίπεδο σακχάρου στο αίμα λιγότερο από 70 mg/dL
Κανονικό: 70 έως 140 mg/dL
Υψηλό: Πάνω από 140 mg/dL
Βήμα 2: Χρήση αυτών των κατηγοριών στο μοντέλο Naïve Bayes. Αυτό απλοποιεί τον υπολογισμό των πιθανοτήτων και καθιστά το μοντέλο πιο ανθεκτικό στις μεταβολές των επιπέδων σακχάρου.

Με τη διακριτοποίηση των δεδομένων, μπορούμε να υπολογίσουμε με μεγαλύτερη ακρίβεια την πιθανότητα κάθε κατηγορίας, το οποίο είναι κρίσιμο για τον Naïve Bayes. Αν ένα νέο δεδομένο πέσει σε ένα εύρος τιμών για το οποίο δεν υπάρχουν δεδομένα εκπαίδευσης, η διακριτοποιημένη προσέγγιση μπορεί ακόμα να το ταξινομήσει βάσει της κατηγορίας στην οποία ανήκει. Η διακριτοποίηση μπορεί να απλοποιήσει το μοντέλο μειώνοντας την πολυπλοκότητα της δια-

χείρισης συνεχών μεταβλητών.

Στο παραπάνω παράδειγμα, η διακριτοποίηση βοηθά τον ταξινομητή Naïve Bayes να διαχειριστεί πιο αποτελεσματικά τη μεταβλητότητα και τη συνεχή φύση των ιατρικών δεδομένων, οδηγώντας σε πιθανώς πιο ακριβείς προβλέψεις νόσου.

Παράδειγμα 2:

Φανταστείτε ότι εργαζόμαστε με ένα σύνολο δεδομένων για να προβλέψουμε τις δαπάνες των πελατών βάσει της ηλικίας τους και του ετήσιου εισοδήματός τους. Θα αποκαλέσουμε αυτά τα χαρακτηριστικά x_0 (ηλικία) και x_1 (ετήσιο εισόδημα).

Στην αρχική τους μορφή, η ηλικία και το εισόδημα είναι συνεχείς μεταβλητές. Εάν εφαρμόσουμε ένα μοντέλο γραμμικής παλινδρόμησης (linear regression), η πρόβλεψη των συνηθειών δαπάνης μπορεί να φαίνεται κάπως έτσι:

Δαπάνες = θ_0 + θ_1 x Ηλικία + θ_2 x Ετήσιο Εισόδημα

Εδώ, η πρόβλεψη εξαρτάται γραμμικά από τις συγκεκριμένες τιμές της ηλικίας και του ετήσιου εισοδήματος. Για παράδειγμα, το μοντέλο μπορεί να υποδείξει ότι ένα άτομο ηλικίας 30 ετών με ετήσιο εισόδημα 50.000 ευρώ έχει ένα συγκεκριμένο μοτίβο δαπάνης, το οποίο μπορεί να διαφέρει σημαντικά από κάποιον ηλικίας 31 ετών με εισόδημα 51.000 ευρώ, ακόμη και αν αυτές οι διαφορές μπορεί να είναι πρακτικά ασήμαντες στον πραγματικό κόσμο.

Ωστόσο, συνήθως οι δαπάνες των πελατών δεν αλλάζουν δραματικά με μικρές διαφορές στην ηλικία ή το εισόδημα. Για να το αντιμετωπίσουμε αυτό, μπορούμε να διακριτοποιήσουμε και τις δύο μεταβλητές σε ομάδες με μεγαλύτερο νόημα.

Ηλικία: Μπορούμε να κατηγοριοποιήσουμε την ηλικία σε ομάδες όπως 18-26, 27-35, 36-45 κλπ. Κάθε ηλικιακή ομάδα μπορεί στη συνέχεια να θεωρηθεί ως ξεχωριστό χαρακτηριστικό, επιτρέποντας στο μοντέλο να μάθει διακριτά μοτίβα δαπάνης που σχετίζονται με διαφορετικές ηλικιακές ομάδες.

Ετήσιο Εισόδημα: Το εισόδημα μπορεί να διακριτοποιηθεί σε κατηγορίες όπως Χαμηλό, Μεσαίο, Υψηλό και Πολύ Υψηλό. Αυτές οι κατηγορίες επιτρέπουν στο μοντέλο να κατανοήσει τις δαπάνες που σχετίζονται με διαφορετικά επίπεδα εισοδήματος αντί για το ακριβές ποσό του εισοδήματος.

Αυτή η προσέγγιση διακριτοποίησης βοηθά το μοντέλο να αντιληφθεί τις ευρύτερες τάσεις στις δαπάνες των πελατών που σχετίζονται με διαφορετικές ηλικιακές ομάδες και επίπεδα εισοδήματος, αντί να επικεντρώνεται σε μικρές παραλλαγές εντός αυτών των συνεχών μεταβλητών. Αντανακλά την πραγματικότητα του μάρκετινγκ, όπου η συμπεριφορά του πελάτη συχνά αλλάζει πιο σημαντικά μεταξύ αυτών των ευρύτερων κατηγοριών [25].

Διαδικασία διακριτοποίησης:

Αρχικά, θα παρουσιάσουμε μερικούς βασικούς όρους για τη διακριτοποίηση:

Χαρακτηριστικό (Feature): Ένα χαρακτηριστικό (feature) αναφέρεται σε μία ατομική μετρήσιμη ιδιότητα. Με απλούστερους όρους, είναι ένα συγκεκριμένο στοιχείο των δεδομένων που χρησιμοποιείται για ανάλυση. Τα χαρακτηριστικά μπορεί να είναι διακριτά, συνεχή ή ονομαστικά. Κατά τη διαδικασία διακριτοποίησης δεδομένων, τα συνεχή χαρακτηριστικά, που είναι συνήθως αριθμητικά και αντιπροσωπεύουν μια ευρεία γκάμα τιμών, μετατρέπονται σε διακριτά χαρακτηριστικά. Αυτή η μετατροπή περιλαμβάνει την κατηγοριοποίηση ή τη διαίρεση αυτών των συνεχών τιμών σε διακριτές ζώνες ή διαστήματα [4].

Παράδειγμα (Instance): Ένα "παράδειγμα" αναφέρεται συνήθως σε μία μεμονωμένη, συγκεκριμένη παρατήρηση ή παραδείγματος εντός ενός συνόλου δεδομένων. Κάθε παράδειγμα αναπαρίσταται συνήθως ως μία γραμμή (row) σε ένα σύνολο δεδομένων και αποτελείται από μια συλλογή χαρακτηριστικών (γνωρισμάτων ή μεταβλητών) που το περιγράφουν. Συνήθως ένα σύνολο δεδομένων βρίσκεται σε μορφή πίνακα, όπου μια γραμμή αντιστοιχεί σε ένα παράδειγμα (Instance) και μια στήλη αντιστοιχεί σε ένα χαρακτηριστικό (Feature). Σε γενικότερο επίπεδο, ένα παράδειγμα είναι ένα επιμέρους σημείο δεδομένων. Το παράδειγμα (Instance) είναι σημαντικό επειδή η διαδικασία της διακριτοποίησης εφαρμόζεται στα χαρακτηριστικά κάθε παραδείγματος. Για παράδειγμα, σε ένα σύνολο δεδομένων όπου κάθε παράδειγμα (Instance) αντιπροσωπεύει ένα άτομο και ένα από τα χαρακτηριστικά είναι η ηλικία (μια συνεχής μεταβλητή), η διακριτοποίηση δεδομένων μπορεί να περιλαμβάνει τη μετατροπή του χαρακτηριστικού της ηλικίας σε διακριτές ηλικιακές ομάδες όπως παιδί, έφηβος, ενήλικας και υπερήλικας. Κάθε άτομο (Instance) στο σύνολο δεδομένων θα κατατάσσεται σε μία από αυτές τις διακριτές ηλικιακές ομάδες βάσει της ηλικίας του [4].

Σημείο κοπής (Cut-point): ένα "σημείο κοπής" αναφέρεται σε μια συγκεκριμένη τιμή που λειτουργεί ως ένα όριο ή κατώφλι στη διαδικασία διαίρεσης των συνεχών δεδομένων σε διακριτά διαστήματα ή κατηγορίες. Αυτή η έννοια είναι ιδιαίτερα σημαντική σε μεθόδους διακριτοποίησης που περιλαμβάνουν την τμηματοποίηση μιας συνεχούς μεταβλητής σε ένα σετ διακριτών εύρων. Για παράδειγμα, σκεφτείτε μια συνεχή μεταβλητή όπως η ηλικία σε ένα σύνολο δεδομένων. Αν αποφασίσετε να διακριτοποιήσετε την ηλικία σε κατηγορίες όπως Παιδί (0-12 ετών), Έφηβος (13-19 ετών) και Ενήλικας (20 ετών και άνω), τα σημεία κοπής σε αυτή την περίπτωση θα ήταν τα 12 και 19. Αυτές οι τιμές ορίζουν τα όρια όπου τελειώνει η μία κατηγορία και αρχίζει η άλλη. Ο καθορισμός των βέλτιστων σημείων κοπής είναι κρίσιμος καθώς επηρεάζει την ποιότητα της διακριτοποίησης. Η επιλογή των σημείων κοπής μπορεί να βασίζεται σε διάφορα κριτήρια όπως διαστήματα ίσου πλάτους (ομοιόμορφη διακριτοποίηση), διαστήματα ίσης συχνότητας (ποσοστιαία διακριτοποίηση) ή διακριτοποίηση βάσει k-means.

Αρτιότητα (Arity ή Bins): Η "αρτιότητα" αναφέρεται στον αριθμό των διακριτών κατηγοριών στα οποία διαιρείται μια συνεχής μεταβλητή. Ουσιαστικά δείχνει πόσες διακριτές τιμές μπορεί να λάβει ένα διακριτοποιημένο χαρακτηριστικό. Για παράδειγμα, αν μια συνεχής μεταβλητή όπως η ηλικία διακριτοποιηθεί σε τρεις κατηγορίες όπως Παιδί, Ενήλικας και Ηλικιωμένος, τότε η αρτιότητα αυτής της διακριτοποιημένης μεταβλητής είναι τρία. Η επιλογή της αρτιότητας

είναι σημαντική επειδή καθορίζει το επίπεδο λεπτομέρειας και την ευαισθησία της ανάλυσης. Μια υψηλότερη αρτιότητα σημαίνει περισσότερες κατηγορίες και δυνητικά λεπτότερες διακρίσεις μεταξύ των δεδομένων, ενώ μια χαμηλότερη αρτιότητα οδηγεί σε ευρύτερες κατηγορίες και μια πιο γενικευμένη άποψη των δεδομένων. Ο καθορισμός της κατάλληλης αρτιότητας είναι ένα κρίσιμο στοιχείο της διακριτοποίησης. Συχνά περιλαμβάνει μια ισορροπία μεταξύ της διευκόλυνσης των δεδομένων για ορισμένους αλγορίθμους (που μπορεί να προτιμούν ή να απαιτούν διακριτά δεδομένα) και της διατήρησης αρκετής λεπτομέρειας για να αποτυπώσουν σημαντικά πρότυπα και σχέσεις μέσα στα δεδομένα. Η επιλογή της αρτιότητας μπορεί να επηρεάζεται από παράγοντες όπως η φύση των δεδομένων, οι στόχοι της ανάλυσης και οι απαιτήσεις των επόμενων τεχνικών επεξεργασίας δεδομένων ή μοντελοποίησης που θα εφαρμοστούν [4].

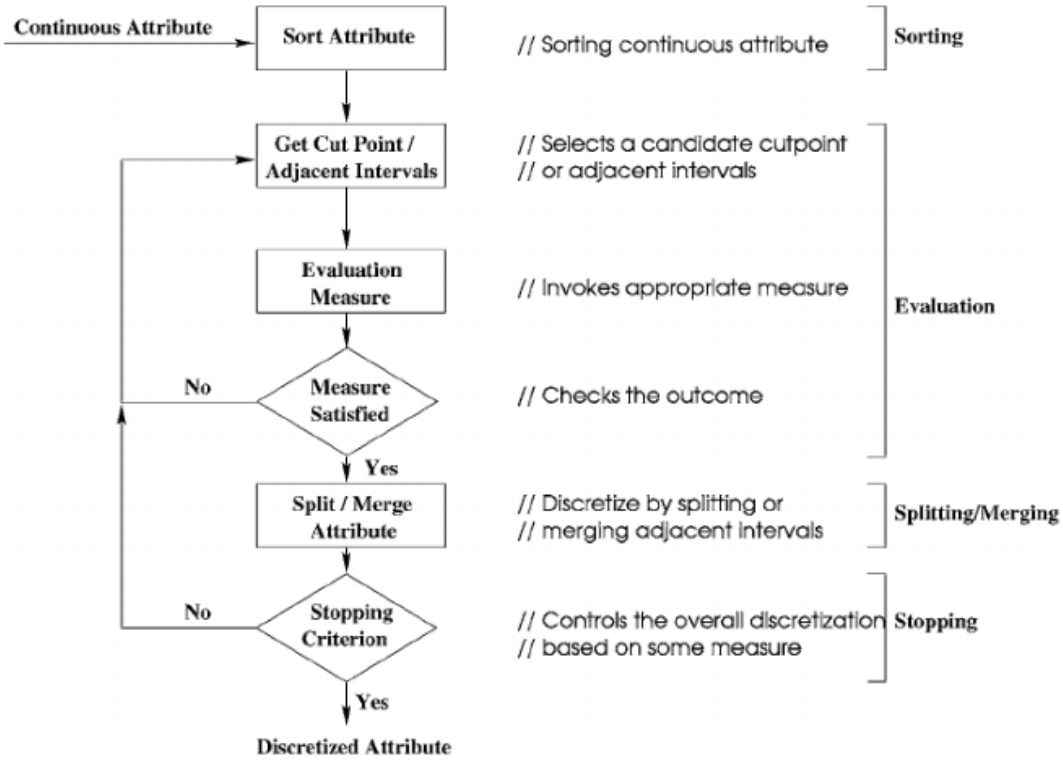
Η διαδικασία διακριτοποίησης περιλαμβάνει τέσσερα βασικά στάδια (σχήμα 2.2). Το πρώτο στάδιο αφορά την ταξινόμηση των συνεχών τιμών του χαρακτηριστικού που υπόκειται σε διακριτοποίηση. Στο δεύτερο στάδιο, εστιάζουμε στην αξιολόγηση ενός σημείου κοπής για τη διαίρεση ή συγχώνευση διπλανών διαστημάτων. Το τρίτο στάδιο περιλαμβάνει την πραγματική διάσπαση ή συγχώνευση αυτών των διαστημάτων, καθοδηγούμενη από ένα συγκεκριμένο κριτήριο. Τέλος, το τέταρτο και τελικό στάδιο είναι η απόφαση πότε να σταματήσουμε τη διαδικασία [4] [26].

Ταξινόμηση: Η Ταξινόμηση είναι το αρχικό βήμα όπου οι συνεχείς τιμές ενός χαρακτηριστικού τακτοποιούνται σε μια συγκεκριμένη σειρά, συνήθως σε αύξουσα ή φθίνουσα αριθμητική σειρά. Αυτό το βήμα είναι κρίσιμο για τη διαδικασία διακριτοποίησης καθώς θέτει τις βάσεις για τον προσδιορισμό των διαστημάτων ή κατηγοριών στα οποία θα κατηγοριοποιηθούν τα συνεχή δεδομένα.

Επιλογή σημείου κοπής: Η Επιλογή ενός σημείου κοπής είναι ενα κρίσιμο βήμα όπου προσδιορίζεται μια συγκεκριμένη τιμή ή κατώφλι (threshold) για να διαιρέσει μια συνεχή μεταβλητή σε διακριτά διαστήματα ή κατηγορίες. Το σημείο κοπής είναι η τιμή στην οποία τα συνεχή δεδομένα 'κόβονται' ή διαχωρίζονται σε διαφορετικές κατηγορίες. Αυτό το βήμα επηρεάζει άμεσα τον τρόπο αναπαράστασης των συνεχών δεδομένων στη διακριτοποιημένη τους μορφή και μπορεί να επηρεάσει σημαντικά την ανάλυση που ακολουθεί [4].

Διάσπαση/Συγχώνευση: Μετά την επιλογή των σημείων κοπής, η συνεχής περιοχή δεδομένων διαιρείται ή 'διασπάται' σε αυτά τα σημεία. Αυτή η διαδικασία δημιουργεί διακριτά διαστήματα ή κατηγορίες. Κάθε σημείο δεδομένων ταξινομείται σε μία από αυτές τις κατηγορίες βάσει της τιμής του. Η διάσπαση χρησιμοποιείται συχνά για να αυξηθεί η λεπτομέρεια της αναπαράστασης των δεδομένων, καθιστώντας τα διακριτοποιημένα δεδομένα πιο συγκεκριμένα και λεπτομερή. Αντίστροφα, η συγχώνευση περιλαμβάνει τον συνδυασμό διπλανών διαστημάτων ή κατηγοριών σε μεγαλύτερα. Αυτό το βήμα μπορεί να γίνει εάν τα δεδομένα σε δύο ή περισσότερα διαστήματα βρεθούν αρκετά παρόμοια ή αν τα μικρότερα διαστήματα δεν παρέχουν σημαντική διαφοροποίηση. Η συγχώνευση μειώνει τον αριθμό των κατηγοριών, κάτι που μπορεί να απλοποιήσει τα δεδομένα και να είναι επωφελές για ορισμένες αναλύσεις ή αλγορίθμους που προτιμούν λιγότερες, ευρύτερες κατηγορίες. Είναι ένας τρόπος μείωσης της πολυπλοκότητας των δεδομένων διατηρώντας την ουσιαστική τους δομή [4].

Κριτήρια Διακοπής: Τα Κριτήρια Διακοπής στη διακριτοποίηση δεδομένων αναφέρονται στις οδηγίες ή τις συνθήκες που χρησιμοποιούνται για να καθοριστεί πότε πρέπει να ολοκληρωθεί η διαδικασία διακριτοποίησης. Αυτό είναι ένα κρίσιμο σημείο γιατί αποτρέπει το overfitting (όπου το μοντέλο προσαρμόζεται υπερβολικά στα συγκεκριμένα δεδομένα στα οποία εκπαιδεύτηκε, εις βάρος της γενίκευσης) και το underfitting (όπου το μοντέλο είναι υπερβολικά απλό για να αποτυπώσει τα υποκείμενα πρότυπα των δεδομένων). Τα κριτήρια διακοπής εξασφαλίζουν ότι η διακριτοποίηση επιτυγχάνει μια ισορροπία μεταξύ λεπτομέρειας και πρακτικότητας [4] [26].



Σχήμα 2.2: Διαδικασία Διακριτοποίησης

Μειονεκτήματα διακριτοποίησης:
Η διακριτοποίηση δεδομένων, ενώ είναι χρήσιμη σε πολλά πλαίσια, ιδιαίτερα στην προετοιμασία δεδομένων για μηχανική μάθηση και στατιστική ανάλυση, μπορεί επίσης να έχει μερικά μειονεκτήματα. Κάποια από τα κύρια μειονεκτήματα είναι:

Απώλεια Πληροφορίας: Είναι ένα από τα σημαντικότερα μειονεκτήματα. Κατηγοριοποιώντας τα συνεχή δεδομένα σε διακριτά διαστήματα, μερικές λεπτομερείς πληροφορίες και των αρχικών δεδομένων μπορεί να χαθούν. Για παράδειγμα, η διακριτοποίηση της ηλικίας σε κατηγορίες όπως παιδί, ενήλικας, ηλικιωμένος αφαιρεί την ακριβή πληροφορία της ηλικίας [24] [27].

Ευαισθησία στα Όρια: Η επιλογή των ορίων για τη διακριτοποίηση μπορεί να επηρεάσει σημαντικά τα αποτελέσματα. Σημεία δεδομένων που βρίσκονται κοντά στα όρια μπορεί να κατηγοριοποιηθούν διαφορετικά λόγω μικρών παραλλαγών, επηρεάζοντας το αποτέλεσμα της ανάλυσης.

Υπεραπλούστευση: Η διακριτοποίηση μπορεί μερικές φορές να απλοποιήσει υπερβολικά τα δεδομένα, οδηγώντας σε απώλεια σημαντικών στοιχείων και σχέσεων που είναι πιο εμφανή στη συνεχή μορφή.

Κίνδυνος Overfitting ή Underfitting: Στη μηχανική μάθηση, η ακατάλληλη διακριτοποίηση μπορεί να οδηγήσει σε μοντέλα που είτε υπερπροσαρμόζονται (Overfitting) είτε υποπροσαρμόζονται (Underfitting).

Στη συνέχεια, θα προχωρήσουμε στην ανάλυση των διαφόρων στρατηγικών διακριτοποίησης.

2.2 Ομοιόμορφη διακριτοποίηση

Η ομοιόμορφη διακριτοποίηση (Equal-Width Discretization) είναι μια τεχνική που χρησιμοποιείται στην επεξεργασία δεδομένων και στη μηχανική μάθηση για να μετατρέψει συνεχείς ή πραγματικούς αριθμούς σε διακριτές κατηγορίες. Αυτή η μέθοδος είναι ιδιαίτερα χρήσιμη όταν δουλεύουμε με αλγόριθμους μηχανικής μάθησης που λειτουργούν καλύτερα με διακριτές εισόδους ή όταν χρειάζεται να απλοποιήσουμε τα δεδομένα για ευκολότερη κατανόηση [24]. Η βασική ιδέα πίσω από τη ομοιόμορφη διακριτοποίηση είναι να μετατραπεί ένα συνεχές εύρος τιμών σε ένα περιορισμένο αριθμό διακριτών διαστημάτων με ίδιο πλάτος [28]. Αυτό γίνεται μέσω των εξής βημάτων:

Καθορισμός του Εύρους Δεδομένων: Πρώτα, υπολογίζεται το εύρος των συνεχών δεδομένων που θα διακριτοποιηθούν. Αυτό γίνεται με τον υπολογισμό της διαφοράς μεταξύ της μέγιστης και της ελάχιστης τιμής των δεδομένων [29].

Διαίρεση σε Διακριτά Διαστήματα: Στη συνέχεια, το εύρος των δεδομένων διαιρείται σε έναν προκαθορισμένο αριθμό διαστημάτων ή κατηγοριών, όπου κάθε διάστημα έχει το ίδιο πλάτος. Για παράδειγμα, αν θέλουμε να χωρίσουμε τα δεδομένα σε τέσσερις κατηγορίες, το εύρος των δεδομένων διαιρείται σε τέσσερα ίσα μέρη [24] [29].

Ανάθεση των Τιμών σε Κατηγορίες: Κάθε τιμή στα αρχικά δεδομένα ανατίθεται σε μια από τις δημιουργημένες κατηγορίες, με βάση το διάστημα στο οποίο πέφτει [29].

Παράδειγμα:

Έστω ότι έχουμε το εξής σετ δεδομένων που περιέχει τις τιμές ενός συνεχούς χαρακτηριστικού (σχήμα 2.3) και θέλουμε να διακριτοποιήσουμε αυτές τις τιμές σε 3 κατηγορίες.

Α/Α	Τιμή
1	5
2	15
3	25
4	35
5	45
6	55
7	65
8	75
9	85
10	95

Σχήμα 2.3: Equal-Width Παράδειγμα

Υπολογισμός Πλάτους Διαστημάτων: Το πλάτος κάθε διαστήματος υπολογίζεται ως εξής:

Μέγιστη τιμή = 95, Ελάχιστη τιμή = 5, Αριθμός κατηγοριών = 3

$$\text{Πλάτος} = \frac{\text{Μέγιστη Τιμή} - \text{Ελάχιστη Τιμή}}{\text{Αριθμός Κατηγοριών}} = \frac{95 - 5}{3} = 30 \tag{2.1}$$

Δημιουργία Διαστημάτων:
Διάστημα 1: 5 έως 34 (περιλαμβάνει τις τιμές 5, 15, 25)
Διάστημα 2: 35 έως 64 (περιλαμβάνει τις τιμές 35, 45, 55)
Διάστημα 3: 65 έως 94 (περιλαμβάνει τις τιμές 65, 75, 85, 95 - το τελευταίο διάστημα συχνά προσαρμόζεται για να περιλαμβάνει τη μέγιστη τιμή)

Τώρα το σετ δεδομένων μας έχει μετατραπεί ως εξής:

Α/Α	Τιμή	Διακριτή Κατηγορία
1	5	1
2	15	1
3	25	1
4	35	2
5	45	2
6	55	2
7	65	3
8	75	3
9	85	3
10	95	3

Σχήμα 2.4: Equal-Width Παράδειγμα Αποτελέσματα

Προκλήσεις και Περιορισμοί:

Επιλογή Αριθμού των Διαστημάτων (Bins):
Ένας σημαντικός παράγοντας στη διακριτοποίηση είναι ο αριθμός των διαστημάτων που θα χρησιμοποιηθούν. Η επιλογή του αριθμού αυτών των διαστημάτων επηρεάζει τον βαθμό λεπτομέρειας και την ευαισθησία της ανάλυσης. Ένας μικρός αριθμός διαστημάτων μπορεί να αποκρύπτει λεπτομέρειες, ενώ ένας πολύ μεγάλος αριθμός μπορεί να οδηγήσει σε υπερβολική εξειδίκευση [30].

Στατιστική Σημασία των Διαστημάτων:
Όταν χρησιμοποιείται η διακριτοποίηση σε περιβάλλον ανάλυσης δεδομένων, είναι σημαντικό να καθοριστεί εάν τα διαστήματα που δημιουργούνται έχουν στατιστική σημασία. Αυτό συμβαίνει όταν τα διαστήματα αντικατοπτρίζουν πραγματικά διαφορές στα δεδομένα, αντί να αποτελούν απλώς αριθμητικές κατατμήσεις χωρίς πρακτική σημασία [31].

Η ομοιόμορφη διακριτοποίηση είναι ευρέως χρησιμοποιημένη σε πολλούς τομείς όπως η βιοστατιστική, οικονομική ανάλυση και μηχανική μάθηση [32]. Παρ' όλα αυτά, η επιλογή της σωστής μεθόδου διακριτοποίησης εξαρτάται από τη φύση των δεδομένων και τον ειδικό σκοπό της ανάλυσης. Σε περιπτώσεις όπου η κατανομή των δεδομένων είναι σκεδαστική ή έχει ακραίες τιμές, άλλες μέθοδοι όπως η ποσοστιαία διακριτοποίηση ή άλλες τεχνικές μπορεί να είναι πιο κατάλληλες.

2.3 Ποσοστιαία διακριτοποίηση

Η βασική λειτουργία της ποσοστιαία διακριτοποίησης (Equal-Frequency Discretization) είναι η μετατροπή συνεχών ή πολύ λεπτομερών δεδομένων σε μια πιο χονδρική και διαχειρίσιμη μορφή. Η βασική ιδέα είναι να χωρίσει τα δεδομένα σε διακριτές κατηγορίες, ώστε κάθε κατηγορία να περιέχει περίπου τον ίδιο αριθμό παραδειγμάτων ή παρατηρήσεων [28] [29]. Αυτό γίνεται μέσω των εξής βημάτων:

Ταξινόμηση Δεδομένων: Αρχικά, τα δεδομένα ταξινομούνται βάσει της τιμής τους.

Διαίρεση σε Διαστήματα: Στη συνέχεια, το εύρος των ταξινομημένων τιμών διαιρείται σε N ίσα τμήματα, όπου N είναι ο αριθμός των επιθυμητών κατηγοριών [29].

Κατανομή σε Κατηγορίες: Κάθε δεδομένο κατατάσσεται στην κατηγορία ανάλογα με το διάστημα στο οποίο ανήκει η τιμή του [29].

Παράδειγμα:

Έστω ότι έχουμε το εξής σετ δεδομένων που περιέχει τις τιμές ενός συνεχούς χαρακτηριστικού (σχήμα 2.5) και θέλουμε να διακριτοποιήσουμε αυτές τις τιμές σε 3 κατηγορίες.

A/A	Τιμή
1	5
2	15
3	25
4	35
5	45
6	55
7	65
8	75
9	85
10	95

Σχήμα 2.5: Equal-Frequency Παράδειγμα

Ταξινόμηση: Τα δεδομένα είναι ήδη ταξινομημένα.

Διαίρεση σε Διαστήματα: Με 10 παρατηρήσεις και 3 κατηγορίες, θέλουμε κάθε κατηγορία να περιέχει περίπου 10/3 ≈ 3-4 παρατηρήσεις.

Κατανομή σε Κατηγορίες:
Κατηγορία 1: Τιμές 5, 15, 25, 35 (A/A 1-4)
Κατηγορία 2: Τιμές 45, 55, 65 (A/A 5-7)
Κατηγορία 3: Τιμές 75, 85, 95 (A/A 8-10)

Τώρα το σετ δεδομένων μας έχει μετατραπεί ως εξής:

A/A	Τιμή	Διακριτή Κατηγορία
1	5	1
2	15	1
3	25	1
4	35	1
5	45	2
6	55	2
7	65	2
8	75	3
9	85	3
10	95	3

Σχήμα 2.6: Equal-Frequency Παράδειγμα Αποτελέσματα

Η ποσοστιαία διακριτοποίηση είναι μια σημαντική μέθοδος στην προεπεξεργασία δεδομένων, παρέχοντας έναν άλλο απλό τρόπο για τη μετατροπή συνεχών δεδομένων σε διακριτές τιμές.

Βοηθά στην αντιμετώπιση περιπτώσεων όπου υπάρχει υψηλή διασπορά στις τιμές των δεδομένων. Παρά τις προκλήσεις που μπορεί να παρουσιάσει, όπως η απώλεια πληροφορίας, προσφέρει σημαντικά οφέλη στην ανάλυση και επεξεργασία μεγάλων σετ δεδομένων.

2.4 Διακριτοποίηση βάσει k-means

Η διακριτοποίηση με βάση τον αλγόριθμο k-means αποτελεί ακόμα μια αποτελεσματική μέθοδο για τη μετατροπή συνεχών τιμών σε διακριτές κατηγορίες. Στην ουσία, ο k-means είναι ένας αλγόριθμος συσταδοποίησης που χρησιμοποιείται για να ομαδοποιήσει τα δεδομένα σε καθορισμένο αριθμό συστάδων, βασιζόμενος στην ελαχιστοποίηση της απόστασης μεταξύ των σημείων των δεδομένων και των κέντρων κάθε συστάδας. Αυτή η προσέγγιση επιτρέπει την εύκολη και οργανωμένη κατηγοριοποίηση συνεχών δεδομένων σε διακριτά σύνολα, καθιστώντας τα πιο διαχειρίσιμα για επεξεργασία και ανάλυση. Η διαδικασία του αλγορίθμου k-means περιγράφεται μέσω των εξής βημάτων [28] [33]:

Επιλογή Αρχικών Κέντρων: Ο αλγόριθμος αρχίζει με την τυχαία επιλογή K σημείων ως αρχικά κέντρα των συστάδων.

Ανάθεση Σημείων σε Συστάδες: Κάθε σημείο δεδομένων ανατίθεται στη συστάδα του κοντινότερου κέντρου.

Ενημέρωση Κέντρων Συστάδων: Τα κέντρα των συστάδων ενημερώνονται ώστε να είναι ο μέσος όρος των σημείων που τους έχουν ανατεθεί.

Επανάληψη: Τα βήματα 2 και 3 επαναλαμβάνονται μέχρι η θέση των κέντρων να μην αλλάξει σημαντικά ή μέχρι να επιτευχθεί κάποιος αριθμός επαναλήψεων.

Παράδειγμα:

Έστω ότι έχουμε το εξής σετ δεδομένων που αποτελείται από μία σειρά τιμών:

A/A	Τιμή
1	2
2	3
3	10
4	12
5	20
6	25
7	30
8	31
9	35
10	40

Σχήμα 2.7: K-means Παράδειγμα

Θα εφαρμόσουμε τον αλγόριθμο k-means για να διακριτοποιήσουμε αυτές τις τιμές σε 3 συστάδες (κατηγορίες).

Επιλογή Αρχικών Κέντρων: Επιλέγουμε τυχαία 3 τιμές ως αρχικά κέντρα των συστάδων. Ας πούμε ότι αυτά είναι τα 3, 20, και 35.

Ανάθεση στις Συστάδες: Κάθε τιμή ανατίθεται στη συστάδα με το κοντινότερο κέντρο. Για παράδειγμα, η τιμή 2 ανατίθεται στη συστάδα με κέντρο 3, ενώ η τιμή 30 ανατίθεται στη συστάδα με κέντρο 35.

Ενημέρωση Κέντρων Συστάδων: Υπολογίζουμε το μέσο όρο των τιμών σε κάθε συστάδα και τοποθετούμε το κέντρο της συστάδας σε αυτή την τιμή.

Επανάληψη: Επαναλαμβάνουμε τα βήματα 2 και 3 μέχρι να σταθεροποιηθούν τα κέντρα των συστάδων.

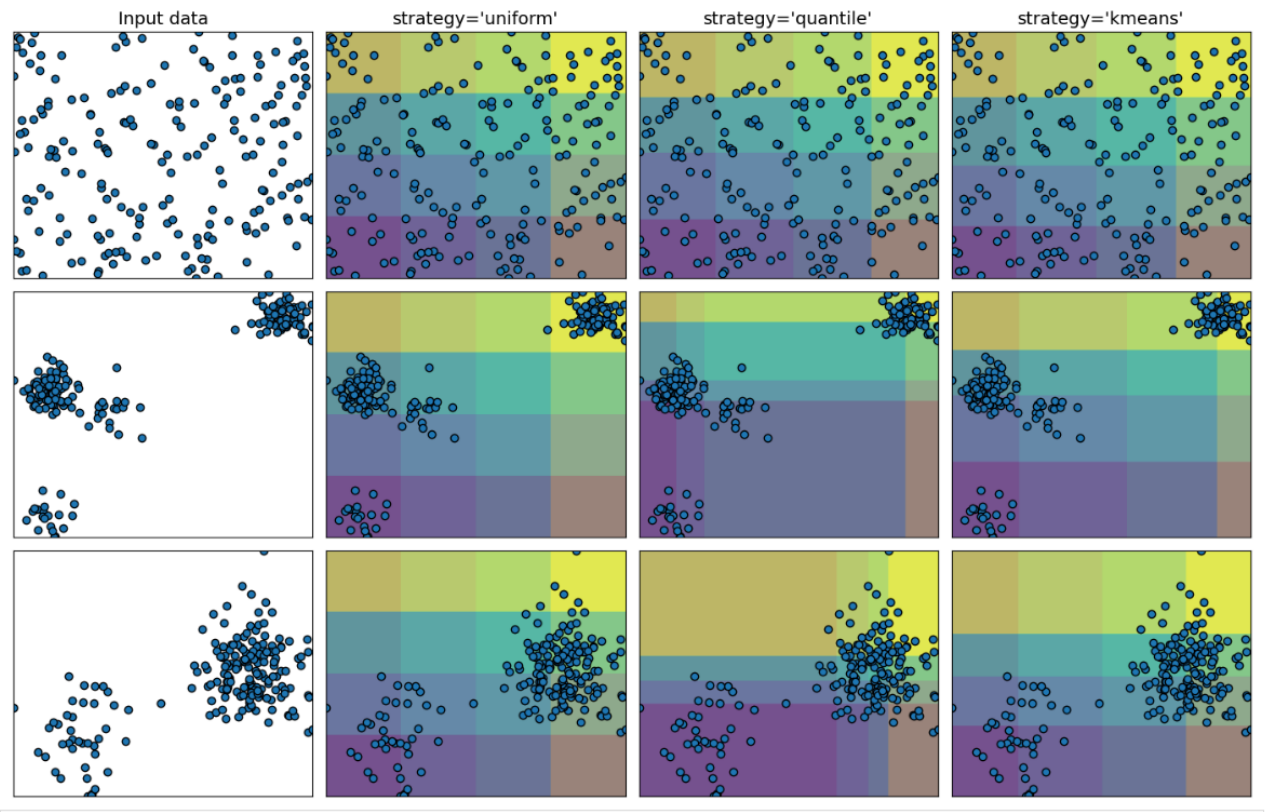
Μετά απο όλες τις επαναλήψεις του k-means, το σετ δεδομένων θα έχει διακριτοποιηθεί σε 3 συστάδες. Κάθε συστάδα θα περιλαμβάνει τις τιμές που είναι πιο κοντά στο κέντρο της. Ας υποθέσουμε ότι τα τελικά κέντρα είναι κοντά στις τιμές 5, 25, και 35 αντίστοιχα, τα δεδομένα μετά τη διακριτοποίηση θα κατανεμηθούν ως εξής:

Α/Α	Τιμή	Συστάδα
1	2	1
2	3	1
3	10	1
4	12	1
5	20	2
6	25	2
7	30	2
8	31	3
9	35	3
10	40	3

Σχήμα 2.8: K-means Παράδειγμα Αποτελέσματα

Σε αυτό το παράδειγμα:
Οι τιμές 2, 3, 10, και 12 έχουν ανατεθεί στη συστάδα 1 με κέντρο κοντά στο 5.
Οι τιμές 20, 25, και 30 έχουν ανατεθεί στη συστάδα 2 με κέντρο κοντά στο 25.
Οι τιμές 31, 35, και 40 έχουν ανατεθεί στη συστάδα 3 με κέντρο κοντά στο 35.

Η διακριτοποίηση k-means βοηθά στην οργάνωση των δεδομένων σε ομάδες με βάση την ομοιότητα των τιμών, κάνοντας τα δεδομένα πιο διαχειρίσιμα για ανάλυση.



Σχήμα 2.9: Στρατηγικές διακριτοποίησης

Κεφάλαιο 3

Αυτόματος προσδιορισμός παραμέτρων διακριτοποίηση

3.1 Κατηγοριοποίηση Naive Bayes

Ο αλγόριθμος Naive Bayes είναι μια τεχνική ταξινόμησης βασισμένη στη θεωρία πιθανοτήτων και στο θεώρημα του Bayes. Ανήκει στην κατηγορία των εποπτικών αλγορίθμων μάθησης και χρησιμοποιείται ευρέως στον τομέα της μηχανικής μάθησης για την κατηγοριοποίηση δεδομένων. Ο αλγόριθμος βασίζεται στην απλοϊκή ("naive") υπόθεση ότι κάθε χαρακτηριστικό του σετ δεδομένων είναι ανεξάρτητο από τα υπόλοιπα [33]. Αυτή η υπόθεση της ανεξαρτησίας καθιστά τον Naive Bayes εξαιρετικά αποτελεσματικό και εύκολο στην υλοποίηση, ιδιαίτερα για μεγάλα σετ δεδομένων. Παρά την απλότητά του, ο Naive Bayes έχει αποδειχθεί αποτελεσματικός σε πολλές εφαρμογές. Χρησιμοποιείται συχνά σε συστήματα ανίχνευσης απάτης, αναγνώριση spam σε emails, κατηγοριοποίηση κειμένων, ακόμα και στην ιατρική για την πρόβλεψη ορισμένων παθήσεων [33].

Η διακριτοποίηση είναι ένα σημαντικό κομμάτι για την αποδοτική λειτουργία του Naive Bayes. Καθώς αυτός ο αλγόριθμος είναι βασισμένος σε πιθανότητες, η διακριτοποίηση των δεδομένων σε σαφώς ορισμένες κατηγορίες βοηθά στην ακριβέστερη εκτίμηση των πιθανοτήτων [24]. Για παράδειγμα, στην κατηγοριοποίηση κειμένων, η διακριτοποίηση των λέξεων και φράσεων σε ξεχωριστές κατηγορίες βοηθά τον Naive Bayes να υπολογίζει πιο ακριβής τις πιθανότητες που αφορούν την ταξινόμηση ενός κειμένου σε ένα συγκεκριμένο θέμα ή κατηγορία. Επιπλέον, η διακριτοποίηση απλοποιεί την επεξεργασία και ανάλυση των δεδομένων, καθώς μετατρέπει τα συνεχή χαρακτηριστικά σε διακριτές κατηγορίες. Αυτό καθιστά τον Naive Bayes ιδανικό για περιπτώσεις όπου τα δεδομένα είναι ετερογενή ή όταν οι σχέσεις μεταξύ των χαρακτηριστικών και των κλάσεων δεν είναι προφανείς.

Η βάση του αλγορίθμου Naive Bayes είναι η θεωρία πιθανοτήτων, η οποία παρέχει το μαθηματικό πλαίσιο για την πρόβλεψη και την κατηγοριοποίηση δεδομένων. Στον πυρήνα του Naive Bayes βρίσκεται το Θεώρημα του Bayes, μια θεμελιώδης αρχή στη στατιστική που περιγράφει πώς μπορεί να ενημερωθεί η πιθανότητα ενός γεγονότος με βάση προηγούμενες γνώσεις ή στοιχεία [33].

Ο βασικός τύπος του Naive Bayes είναι:

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)} \tag{3.1}$$

$P(C_k|x)$: Η μετα-πιθανότητα της κλάσης C_k δεδομένου του δείγματος x.
$P(C_k)$: Η προηγούμενη πιθανότητα της κλάσης C_k.
$P(x|C_k)$: Η πιθανοφάνεια του δείγματος x δεδομένου της κλάσης C_k.
$P(x)$: Η συνολική πιθανότητα του δείγματος x.

Ας δούμε ενα παράδειγμα για να κατανοήσουμε καλύτερα τον Naive Bayes. Έστω ότι έχουμε ένα σετ δεδομένων που περιλαμβάνει καιρικές συνθήκες και την απόφαση αν θα παίξει ή όχι ένα παιχνίδι τένις. Τα δεδομένα μας είναι τα εξής:

Καιρός	Παίζει Τένις
Ηλιόλουστος	Ναι
Ηλιόλουστος	Όχι
Συννεφιασμένος	Ναι
Βροχερός	Ναι
Βροχερός	Όχι

Σχήμα 3.1: Naive Bayes Παράδειγμα

Θέλουμε να χρησιμοποιήσουμε τον Naive Bayes για να προβλέψουμε αν θα παίξουμε τένις βάσει της καιρικής συνθήκης.

Πρώτα, υπολογίζουμε τις πιθανότητες για κάθε κατηγορία καιρού (ηλιόλουστο, συννεφιασμένο, βροχερό). Στο δείγμα μας, έχουμε 2/5 ηλιόλουστο, 1/5 συννεφιασμένο, και 2/5 βροχερό. Στη συνέχεια, υπολογίζουμε τις πιθανότητες να παίξουμε τένις δεδομένης κάθε καιρικής συνθήκης. Για παράδειγμα, η πιθανότητα να παίξουμε τένις όταν είναι ηλιόλουστος είναι 1/2 και όταν είναι βροχερός είναι επίσης 1/2. ας υποθέσουμε ότι θέλουμε να υπολογίσουμε την πιθανότητα να παίξουμε τένις δεδομένου ότι ο καιρός είναι ηλιόλουστος.

$$P(\text{Τένις}|\text{Ηλιόλουστο}) = \frac{P(\text{Ηλιόλουστο}|\text{Τένις}) \times P(\text{Τένις})}{P(\text{Ηλιόλουστο})} \tag{3.2}$$

Υπολογίζουμε τις απαραίτητες πιθανότητες:

P(Τένις) = Συνολικές φορές που παίχτηκε τένις. Συνολικές παρατηρήσεις = 3/5.
P(Ηλιόλουστο) = Συνολικές φορές με ηλιόλουστο καιρό. Συνολικές παρατηρήσεις = 2/5.
P(Ηλιόλουστο|Τένις) = Συνολικές φορές με ηλιόλουστο καιρό όταν παίχτηκε τένις. Συνολικές φορές που παίχτηκε τένις = 1/3.

Υπολογισμός της πιθανότητας να παίξουμε τένις δεδομένου ότι είναι ηλιόλουστος:

$$P(\text{Τένις}|\text{Ηλιόλουστο}) = \frac{1/3 \times 3/5}{2/5} = 1/2 \tag{3.3}$$

Έτσι, βάσει των δεδομένων που έχουμε, η πιθανότητα να παίξουμε τένις όταν ο καιρός είναι ηλιόλουστος είναι 50%.

Στη συνέχεια, θα αναλύσουμε τον υπολογισμό της ακρίβειας του μοντέλου μας, με στόχο να αξιολογήσουμε την ικανότητά του να πραγματοποιεί αξιόπιστες προβλέψεις και να κατηγοριοποιεί αποτελεσματικά τα δεδομένα. Στον Naive Bayes, όπως και σε άλλους αλγορίθμους μηχανικής μάθησης, η ακρίβεια (accuracy) αναφέρεται στο ποσοστό των περιπτώσεων που το μοντέλο προβλέπει σωστά την κατηγορία ή την ετικέτα ενός δεδομένου σε σύγκριση με τις πραγματικές ετικέτες ή κατηγορίες που είναι γνωστές από το σετ δεδομένων. Πιο συγκεκριμένα, στον Naive Bayes, το accuracy υπολογίζεται ως το ποσοστό των σωστών προβλέψεων από το σύνολο των προβλέψεων που έκανε το μοντέλο. Αυτό συνήθως γίνεται μέσω της σύγκρισης των προβλεπόμενων ετικετών του μοντέλου με τις πραγματικές ετικέτες σε ένα δοκιμαστικό σετ δεδομένων (test set) [34].

Η εξίσωση για τον υπολογισμό της ακρίβειας είναι:

$$\text{Ακρίβεια} = \frac{\text{Αριθμός Σωστών Προβλέψεων}}{\text{Συνολικός Αριθμός Προβλέψεων}} \tag{3.4}$$

Ας εξετάσουμε ένα παράδειγμα για τον υπολογισμό της ακρίβειας (accuracy) του αλγορίθμου Naive Bayes για να κατανοήσουμε με μεγαλύτερη σαφήνεια τη σχετική διαδικασία:

Έστω ότι έχουμε τα εξής αποτελέσματα προβλέψεων Naive Bayes: (σχήμα 3.2)

Καιρός	Παίζει Τένις (Πραγματικό)	Παίζει Τένις (Προβλέψεις Naive Bayes)
Ηλιόλουστος	Ναι	Όχι
Συννεφιασμένος	Ναι	Ναι
Βροχερός	Όχι	Όχι
Ηλιόλουστος	Όχι	Ναι
Βροχερός	Ναι	Ναι

Σχήμα 3.2: Naive Bayes Ακρίβεια Παράδειγμα

Για να υπολογίσουμε την ακρίβεια, μετράμε πόσες φορές οι προβλέψεις του μοντέλου συμφωνούν με τις πραγματικές τιμές και διαιρούμε αυτόν τον αριθμό με το συνολικό αριθμό των παρατηρήσεων. Στο παράδειγμά μας, οι σωστές προβλέψεις είναι 3 (συννεφιασμένος-ναι, βροχερός-όχι, βροχερός-ναι), ενώ ο συνολικός αριθμός των παρατηρήσεων είναι 5. Έτσι, το accuracy είναι:

$$\text{Ακρίβεια} = \frac{\text{Αριθμός Σωστών Προβλέψεων}}{\text{Συνολικός Αριθμός Προβλέψεων}} = \frac{3}{5} = 0.6 \text{ ή } 60\% \tag{3.5}$$

Αυτό σημαίνει ότι το μοντέλο Naive Bayes στο δεδομένο παράδειγμα έχει ακρίβεια 60%.

Προχωρώντας, θα εξετάσουμε πώς ο αλγόριθμος Naive Bayes μπορεί να λειτουργήσει ως αποδοτική μέθοδος για την εύρεση του ιδανικού αριθμού διακριτών τιμών, αλλά και για την επιλογή της πιο κατάλληλης στρατηγικής κατά τη διαδικασία διακριτοποίησης των δεδομένων μας. Βασιζόμενοι στη μετρική της ακρίβειας (accuracy) του Naive Bayes, μπορούμε να αξιολογήσουμε και να βελτιστοποιήσουμε την απόδοση του μοντέλου στην κατηγοριοποίηση των δεδομένων.

3.2 Αυτόματη επιλογή στρατηγικής μέσω κατηγοριοποίησης Naive Bayes

Στη διακριτοποίηση δεδομένων, η επιλογή μιας αποδοτικής στρατηγικής αποτελεί σημαντικό βήμα για την αποτελεσματική επεξεργασία τους. Η βέλτιστη στρατηγική εξαρτάται σημαντικά από τη φύση των δεδομένων. Στην προκειμένη περίπτωση, θα αναλύσουμε πώς μπορούμε να καθορίσουμε την ιδανική στρατηγική για τα δεδομένα μας, εφαρμόζοντας την τεχνική κατηγοριοποίησης μέσω του αλγορίθμου Naive Bayes:

Η διαδικασία ξεκινά με την εφαρμογή της ομοιόμορφης (uniform) διακριτοποίησης στα δεδομένα. Αυτή η στρατηγική αναλύει τα δεδομένα με έναν τρόπο που διασφαλίζει ομοιόμορφη κατανομή των τιμών σε διάφορα εύρη. Στη συνέχεια, εφαρμόζουμε τον αλγόριθμο Naive Bayes για την κατηγοριοποίηση των διακριτοποιημένων δεδομένων, με σκοπό την εκτίμηση της ακρίβειας του.

Μετά, επαναλαμβάνουμε την διαδικασία αλλά αυτή τη φορά χρησιμοποιούμε ποσοστιαία (quantile) διακριτοποίηση. Αυτή η μέθοδος χωρίζει τα δεδομένα σε ομάδες με βάση τις τιμές τους, εξασφαλίζοντας ότι κάθε ομάδα περιέχει περίπου ίσο αριθμό παρατηρήσεων. Και πάλι, εφαρμόζουμε τον αλγόριθμος Naive Bayes και βρίσκουμε την ακρίβεια.

Τέλος, πραγματοποιείται η διακριτοποίηση μέσω της μεθόδου k-means, η οποία ομαδοποιεί τα δεδομένα σε k συστάδες βάσει των χαρακτηριστικών τους. Ακολουθεί η εφαρμογή του Naive Bayes και η αντίστοιχη αξιολόγηση της ακρίβειας.

Σε κάθε στάδιο, η ακρίβεια του μοντέλου Naive Bayes λειτουργεί ως κριτήριο για την αξιολόγηση της επιτυχίας της στρατηγικής διακριτοποίησης. Η στρατηγική που παράγει την υψηλότερη ακρίβεια κρίνεται ως η πλέον αποτελεσματική. Η προσέγγιση αυτή επιτρέπει την αντικειμενική και αυτοματοποιημένη επιλογή της καλύτερης μεθόδου διακριτοποίησης. Παρακάτω είναι ο αλγόριθμος για την αυτόματη επιλογή της κατάλληλης στρατηγικής διακριτοποίησης:

Algorithm 1 Αυτόματη επιλογή στρατηγικής μέσω κατηγοριοποίησης Naive Bayes

```
Δεδομένα: Σύνολο δεδομένων
Μεταβλητές: καλύτερη_στρατηγική, καλύτερη_ακρίβεια
Μέθοδοι_Διακριτοποίησης: [Ομοιόμορφη_Διακριτοποίηση, Ποσοστιαία_Διακριτοποίηση, Διακριτοποίηση_KMeans]
Ονόματα_Μεθόδων: ["Ομοιόμορφη", "Ποσοστιαία", "KMeans"]
καλύτερη_ακρίβεια ← 0
καλύτερη_στρατηγική ← " "
function naive_bayes(δεδομένα)
    Κώδικας για την εκτίμηση της ακρίβειας μέσω Naive Bayes
    return accuracy
end function
for i ← 0 έως μήκος(Μέθοδοι_Διακριτοποίησης) − 1 do
    Μέθοδος ← Μέθοδοι_Διακριτοποίησης[i]
    διακριτοποιημένα_δεδομένα ← Μέθοδος(Δεδομένα)
    τρέχουσα_ακρίβεια ← naive_bayes(διακριτοποιημένα_δεδομένα)
    if τρέχουσα_ακρίβεια > καλύτερη_ακρίβεια then
        καλύτερη_ακρίβεια ← τρέχουσα_ακρίβεια
        καλύτερη_στρατηγική ← Ονόματα_Μεθόδων[i]
    end if
end for
Εκτύπωση "Η καλύτερη στρατηγική είναι", καλύτερη_στρατηγική, "με ακρίβεια", καλύτερη_ακρίβεια
```

3.3 Αυτόματος προσδιορισμός αριθμού bins μέσω κατηγοριοποίησης Naive Bayes

Ένας σημαντικός παράγοντας στην επεξεργασία δεδομενων και στην διακριτοποίηση είναι ο προσδιορισμός του αριθμού των bins, δηλαδή των διακριτών εύρων τιμών, στα οποία μπορούν να κατανεμηθούν οι συνεχείς μεταβλητές. Για τον εντοπισμό του ιδανικού αριθμού των bins στη διαδικασία διακριτοποίησης, χρησιμοποιώντας τη μέθοδο κατηγοριοποίησης Naive Bayes, ακολουθούμε τα παρακάτω βήματα:

Η διαδικασία ξεκινά με τη διακριτοποίηση των δεδομένων με χρήση 2 bins. Στη συνέχεια, εφαρμόζουμε τον αλγόριθμο Naive Bayes για την κατηγοριοποίηση αυτών των δεδομένων και καταγράφουμε την ακρίβεια της πρόβλεψης. Επαναλαμβάνουμε την διαδικασία, αυξάνοντας κάθε φορά τον αριθμό των bins κατά ένα. Μετά από κάθε αύξηση, τα δεδομένα διακριτοποιούνται ξανά βάσει του νέου αριθμού των bins και ακολουθεί η εφαρμογή του Naive Bayes, με την αντίστοιχη καταγραφή της ακρίβειας. Αυτή η διαδικασία συνεχίζεται μέχρι να φτάσει σε έναν αριθμό των bins όπου η ακρίβεια δεν βελτιώνεται περαιτέρω.

Στο τέλος αυτής της διαδικασίας, ο αριθμός των bins που συνδέεται με την υψηλότερη ακρίβεια στο μοντέλο Naive Bayes θεωρείται ο βέλτιστος. Αυτός ο τρόπος προσδιορισμού των bins επιτρέπει την προσαρμογή της διακριτοποίησης των δεδομένων στις ειδικές απαιτήσεις κάθε συγκεκριμένου σετ δεδομένων και προβλήματος, βελτιστοποιώντας την απόδοση του μοντέλου πρόβλεψης. Παρακάτω είναι ο αλγόριθμος για τον αυτόματο προσδιορισμό αριθμού bins μέσω κατηγοριοποίησης Naive Bayes:

Algorithm 2 Αυτόματος Προσδιορισμός Αριθμού Bins

```
Δεδομένα: Σύνολο δεδομένων
Μεταβλητές: καλύτερος_αριθμός_bins, καλύτερη_ακρίβεια
καλύτερη_ακρίβεια ← 0
καλύτερος_αριθμός_bins ← 0
function Naive_Bayes(διακριτοποιημένα_δεδομένα)
    Κώδικας για την εκτίμηση της ακρίβειας μέσω Naive Bayes
    return accuracy
end function
for num_bins ← 2 to N do
    διακριτοποιημένα_δεδομένα ← Διακριτοποίηση(Δεδομένα, num_bins)
    τρέχουσα_ακρίβεια ← Naive_Bayes(διακριτοποιημένα_δεδομένα)
    if τρέχουσα_ακρίβεια > καλύτερη_ακρίβεια then
        καλύτερη_ακρίβεια ← τρέχουσα_ακρίβεια
        καλύτερος_αριθμός_bins ← num_bins
    end if
end for
Εκτύπωση "Ο καλύτερος αριθμός bins είναι", καλύτερος_αριθμός_bins, "με ακρίβεια", καλύτερη_ακρίβεια
```

Κεφάλαιο 4

Γλώσσες και Τεχνολογίες

4.1 Server Side

Η πλευρά του διακομιστή, γνωστή ως server side, αφορά όλες τις διεργασίες και τις λειτουργίες που εκτελούνται στο διακομιστή, σε αντίθεση με τον πελάτη ή τον χρήστη (client side). Στο πλαίσιο αυτό, ο διακομιστής είναι ένας υπολογιστής ή ένα σύστημα που παρέχει πόρους, δεδομένα, υπηρεσίες ή προγράμματα σε άλλους υπολογιστές, γνωστούς ως πελάτες, σε ένα δίκτυο. Στην πλευρά του διακομιστή, οι βασικές λειτουργίες περιλαμβάνουν την επεξεργασία αιτημάτων από πελάτες, συνήθως μέσω του διαδικτύου, τη διαχείριση βάσεων δεδομένων για αποθήκευση και ανάκτηση δεδομένων, και την παροχή αυξημένων μέτρων ασφαλείας για προστασία από ανεπιθύμητες ή επιβλαβείς προσβάσεις. Επιπλέον, εκτελείται η λογική της εφαρμογής, όπως υπολογισμοί, επεξεργασία δεδομένων και λήψη αποφάσεων. Για την ανάπτυξη του server-side, χρησιμοποιούνται διάφορες γλώσσες προγραμματισμού και τεχνολογίες. Παρακάτω θα αναλύσουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής AutoDiscretizer στο server side:

4.1.1 PHP

Η PHP, συντομογραφία των λέξεων Hypertext Preprocessor, είναι μια δημοφιλής γλώσσα ανοικτού κώδικα, ειδικά σχεδιασμένη για την ανάπτυξη ιστοσελίδων και είναι ευρέως χρησιμοποιούμενη για server-side προγραμματισμό. Η PHP επιτρέπει στους προγραμματιστές να δημιουργήσουν δυναμικές ιστοσελίδες που μπορούν να αλληλεπιδράσουν με βάσεις δεδομένων και να παρέχουν προσαρμοσμένο περιεχόμενο στους χρήστες [35].

Επιπλέον, η PHP είναι μια γλώσσα σεναρίων (scripting) επειδή προσφέρει ειδικές λειτουργίες για συγκεκριμένες δράσεις ή εργασίες. Αυτό σημαίνει ότι η PHP μπορεί να ερμηνεύσει κώδικα που είναι ενσωματωμένος σε διάφορα λογισμικά περιβάλλοντα. Επιπλέον, ο κώδικας της PHP είναι συχνά πιο απλός και άμεσος. Η κύρια διαφορά μεταξύ μιας γλώσσας προγραμματισμού και μιας γλώσσας σεναρίων (scripting), όπως η PHP, βρίσκεται στον τρόπο εκτέλεσής τους. Οι γλώσσες προγραμματισμού λειτουργούν ανεξάρτητα και μετατρέπονται σε κώδικα που μπορεί να διαβάσει και να εκτελέσει ένας υπολογιστής. Αντιθέτως, οι γλώσσες σεναρίων(scripting) λειτουργούν εντός ενός υπάρχοντος προγράμματος και χρησιμοποιούν ερμηνευτή για την εκτέ-

λεση του κώδικά τους.

Η PHP διακρίνεται ως γλώσσα που λειτουργεί κυρίως στην πλευρά του διακομιστή (server-side). Αυτό σημαίνει ότι ένας διακομιστής εκτελεί τις οδηγίες που περιέχονται σε ένα script. Με αυτόν τον τρόπο, ο διακομιστής δέχεται αιτήματα και παρέχει τα απαραίτητα δεδομένα. Όταν λάβει ένα script, ο διακομιστής το επεξεργάζεται και στέλνει την αποτέλεσματική έξοδο προς τον περιηγητή ιστού, επιτρέποντας έτσι την εμφάνιση περιεχομένου στον χρήστη.

Ένα από τα βασικά της χαρακτηριστικά της PHP είναι η εύκολη ενσωμάτωσή της με HTML, πράγμα που επιτρέπει τη δημιουργία δυναμικού περιεχομένου μέσα σε ιστοσελίδες. Ακόμη, η PHP υποστηρίζει μια ευρεία γκάμα βάσεων δεδομένων, όπως MySQL, PostgreSQL και Oracle. Αυτό την καθιστά ιδανική για την ανάπτυξη ιστοτόπων που βασίζονται σε βάσεις δεδομένων, προσφέροντας στους προγραμματιστές τη δυνατότητα να διαχειρίζονται μεγάλους όγκους δεδομένων με αποτελεσματικότητα. Επιπρόσθετα, η PHP διαθέτει ενσωματωμένες λειτουργίες για τη διαχείριση συνεδριών (sessions) και cookies, προσφέροντας μεγάλη ευελιξία στη δημιουργία προσαρμοσμένων εμπειριών χρήστη. Αυτό επιτρέπει στους διαχειριστές ιστοσελίδων να προσφέρουν πιο προσωποποιημένο περιεχόμενο και να βελτιώσουν την αλληλεπίδραση με τους χρήστες. Η PHP είναι επίσης γνωστή για την ευελιξία και την επεκτασιμότητά της. Μπορεί να ενσωματωθεί με διάφορες άλλες τεχνολογίες και εργαλεία, καθιστώντας την κατάλληλη για μια ευρεία γκάμα εφαρμογών, από απλές ιστοσελίδες έως πολύπλοκα συστήματα διαχείρισης περιεχομένου και εφαρμογές web. Τέλος, η PHP διαθέτει μια μεγάλη και ενεργή κοινότητα προγραμματιστών. Λόγω της μακράς της ιστορίας και της ευρείας χρήσης της, οι προγραμματιστές της PHP έχουν συγκεντρώσει μια πλούσια πηγή γνώσης και προσφέρουν εκτεταμένη υποστήριξη, κάτι που την καθιστά εξαιρετική επιλογή για αρχάριους και έμπειρους προγραμματιστές [35]. Παρακάτω είναι ένα απλό παράδειγμα κώδικα PHP:

```php
<?php
echo "Hello World!";

function add($a, $b) {
    return $a + $b;
}

$result = add(6, 4);
echo "The sum of 6 and 4 is " . $result;
?>
```

Σχήμα 4.1: Παράδειγμα PHP

4.1.2 Python

Η Python είναι μια εξαιρετικά δημοφιλής, υψηλού επιπέδου γλώσσα προγραμματισμού. Είναι γνωστή για την έμφαση που δίνει στην αναγνωσιμότητα του κώδικα, με τη χρήση σημαντικών κενών επιτρέπει στους προγραμματιστές να εκφράζουν ιδέες σε λιγότερες γραμμές κώδικα

από ό,τι θα χρειαζόταν σε γλώσσες όπως η C++ ή Java. Η Python υποστηρίζει πολλαπλά προγραμματιστικά στοιχεία, όπως τον δομημένο αντικειμενοστραφή και λειτουργικό προγραμματισμό. Είναι ευρέως χρησιμοποιημένη σε διάφορους τομείς, όπως η ανάπτυξη ιστοσελίδων, η αυτοματοποίηση, η ανάλυση δεδομένων και τεχνητής νοημοσύνης [36] [37].

Μία από τις βασικές αρχές της Python είναι η απλότητα και η σαφήνεια. Ο κώδικας της Python συχνά χαρακτηρίζεται ως σχεδόν αναγνώσιμος, κάτι που καθιστά εύκολη την εκμάθηση της γλώσσας για νέους προγραμματιστές και βοηθά στην αποφυγή λαθών που συμβαίνουν λόγω περίπλοκου ή ασαφούς κώδικα. Η δημοτικότητα της Python έχει αυξηθεί σημαντικά τα τελευταία χρόνια, χάρη στην ευελιξία της και στην ευρεία χρήση της σε τομείς όπως η ανάλυση δεδομένων και η μηχανική μάθηση [37]. Η κοινότητα της Python είναι ενεργή και υποστηρικτική, με πολλές διαθέσιμες πηγές για μάθηση και προβλήματα, πράγμα που την καθιστά ιδανική για αρχάριους και εμπειρογνώμονες προγραμματιστές.

Ένα από τα πιο ισχυρά χαρακτηριστικά της Python είναι η μεγάλη της βιβλιοθήκη, το Standard Library, που περιλαμβάνει μονάδες για σχεδόν κάθε ανάγκη, από τον χειρισμό αρχείων μέχρι τις δικτυακές επικοινωνίες. Επιπλέον, υπάρχει μια τεράστια κοινότητα που αναπτύσσει και συντηρεί πληθώρα εξωτερικών βιβλιοθηκών, προσφέροντας λύσεις για συγκεκριμένα προβλήματα ή τομείς. Ας δούμε μερικές σημαντικές βιβλιοθήκες της Python:

NumPy:
Αυτή η βιβλιοθήκη είναι η βάση για την επιστημονική υπολογιστική εργασία στην Python. Προσφέρει μια ευρεία γκάμα από αριθμητικές λειτουργίες, επιτρέποντας την αποτελεσματική εργασία με πολυδιάστατους πίνακες και μαθηματικές λειτουργίες [38].

Pandas:
Ειδικεύεται στην ανάλυση και την κατανόηση δεδομένων. Με τις δομές δεδομένων όπως τα DataFrame και τα Series, η Pandas διευκολύνει την επεξεργασία και την ανάλυση των συγκεντρωτικών δεδομένων [39].

Matplotlib:
Είναι μια βασική βιβλιοθήκη για τη δημιουργία στατικών, διαδραστικών και κινούμενων οπτικοποιήσεων στην Python. Χρησιμοποιείται ευρέως για τη δημιουργία γραφημάτων και διαγραμμάτων [40].

SciPy:
Στηρίζεται στην NumPy και είναι αφιερωμένη στην επιστημονική και τεχνική υπολογιστική. Περιλαμβάνει μονάδες για την βελτιστοποίηση, την αλγεβρική επίλυση, την ολοκλήρωση, και πολλά άλλα [41].

Flask και Django:
Για την ανάπτυξη ιστοσελίδων, οι Flask και Django είναι δύο από τις πιο δημοφιλείς επιλογές. Η Flask είναι ένα μικροπλαίσιο για απλές ιστοσελίδες [42], ενώ η Django προσφέρει μια πιο ολοκληρωμένη λύση για μεγάλα και περίπλοκα έργα [43].

TensorFlow και PyTorch:
Αυτές οι δύο βιβλιοθήκες είναι κυρίαρχες στον τομέα της μηχανικής μάθησης και της τεχνητής νοημοσύνης. Προσφέρουν εργαλεία για την δημιουργία και την εκπαίδευση νευρωνικών δικτύων, πραγματοποιώντας πολύπλοκες υπολογιστικές εργασίες [44] [45].

Scikit-learn:
Η Scikit-learn είναι μια ισχυρή και δημοφιλής βιβλιοθήκη στην Python, αφιερωμένη στην μηχανική μάθηση. Αναπτύχθηκε στηριζόμενη στις βιβλιοθήκες NumPy, SciPy και Matplotlib, και προσφέρει ένα εύκολο στη χρήση και προσβάσιμο περιβάλλον για την εκπαίδευση και την εφαρμογή διαφόρων μοντέλων μηχανικής μάθησης. Οι κύριες δυνατότητες της Scikit-learn περιλαμβάνουν κατηγοριοποίηση, παλινδρόμηση (regression), συσταδοποίηση καθώς και προεπεξεργασία δεδομένων [46].

4.1.3 XAMPP

Το XAMPP είναι ένα δημοφιλής και εύχρηστο λογισμικό που περιλαμβάνει τον Apache HTTP Server, την MariaDB και τους ερμηνευτές για τις γλώσσες προγραμματισμού PHP και Perl. Το όνομα ”XAMPP” προέρχεται από τον συνδυασμό των αρχικών των λογισμικών που περιέχει (X, Apache, MariaDB, PHP, Perl). Η ”X” στην αρχή συμβολίζει ότι είναι διαθέσιμο για διάφορες πλατφόρμες (cross-platform). Ο κύριος στόχος του XAMPP είναι να προσφέρει στους προγραμματιστές μια εύκολη και γρήγορη λύση για τη δημιουργία ενός πλήρους εξυπηρετητή web στον τοπικό υπολογιστή τους. Αυτό επιτρέπει την ανάπτυξη και τη δοκιμή ιστοσελίδων ή εφαρμογών πριν την κατάθεση σε έναν πραγματικό διακομιστή, προσφέροντας ένα ιδανική περιβάλλον για την εκμάθηση και την πειραματική ανάπτυξη [47].

Το XAMPP διατίθεται για Windows, Linux και macOS και περιλαμβάνει διάφορα εργαλεία όπως το phpMyAdmin για τη διαχείριση της βάσης δεδομένων MariaDB. Είναι επίσης δημοφιλές στην κοινότητα των προγραμματιστών λόγω της ευκολίας εγκατάστασης και ρύθμισης. Η εγκατάσταση του XAMPP είναι απλή και απαιτεί μόνο το κατέβασμα και την εκτέλεση του. Μόλις εγκατασταθεί, οι χρήστες μπορούν να ξεκινήσουν τον Apache και την MariaDB μέσω του πίνακα ελέγχου του XAMPP και να ξεκινήσουν την ανάπτυξη των ιστοσελίδων τους.

4.1.4 API

Το API (Application Programming Interface) είναι ένα σύνολο κανόνων και προδιαγραφών που επιτρέπουν σε διαφορετικά λογισμικά ή εφαρμογές να επικοινωνούν μεταξύ τους. Μέσω ενός API, διαφορετικά συστήματα μπορούν να επικοινωνούν και να ανταλλάσσουν δεδομένα ή λειτουργίες με τρόπο τυποποιημένο και αυτοματοποιημένο [48]. Τα APIs είναι θεμελιώδη στοιχεία στη σύγχρονη ανάπτυξη λογισμικού, καθώς επιτρέπουν την ευελιξία και την επεκτασιμότητα των εφαρμογών. Με τη χρήση τους, οι προγραμματιστές μπορούν να ενσωματώσουν λειτουργίες από άλλες εφαρμογές ή υπηρεσίες, όπως κοινωνικά δίκτυα, πληρωμές, χάρτες, κτλ., χωρίς να χρειάζεται να αναπτύξουν από την αρχή όλη την υποδομή και τις λειτουργίες αυτές.

Υπάρχουν διάφοροι τύποι APIs, όπως τα Web APIs, που επιτρέπουν την επικοινωνία μέσω

διαδικτύου, και τα Library APIs, που παρέχουν λειτουργίες και εργαλεία ενσωματωμένα σε συγκεκριμένες βιβλιοθήκες λογισμικού. Τα APIs διατίθενται συνήθως με σαφή τεκμηρίωση που περιγράφει τον τρόπο χρήσης τους, τις διαθέσιμες λειτουργίες και τα προαπαιτούμενα για την επικοινωνία με αυτά. Υπάρχουν τέσσερις διαφορετικοί τρόποι με τους οποίους μπορούν να λειτουργήσουν τα API:

SOAP APIs:
Το SOAP (Simple Object Access Protocol) είναι ένα πρότυπο πρωτόκολλο που χρησιμοποιείται για την ανταλλαγή δομημένων πληροφοριών σε ένα δίκτυο υπολογιστών. Τα SOAP APIs είναι αυστηρά στη δομή τους και βασίζονται σε XML για την ανταλλαγή μηνυμάτων [48]. Είναι ένα λιγότερο ευέλικτο API που ήταν πιο δημοφιλές στο παρελθόν.

RPC APIs:
Το RPC (Remote Procedure Call) είναι μια τεχνική που επιτρέπει σε έναν προγραμματιστή να εκτελέσει κώδικα σε έναν απομακρυσμένο διακομιστή. Είναι απλά στην υλοποίηση και επιτρέπουν την άμεση εκτέλεση λειτουργιών σε έναν απομακρυσμένο διακομιστή [48].

Websocket APIs:
Τα Websocket APIs χρησιμοποιούν το πρωτόκολλο WebSocket για να επιτρέψουν διπλής κατεύθυνσης επικοινωνία μεταξύ πελάτη και διακομιστή σε πραγματικό χρόνο [48]. Αυτό τα καθιστά ιδανικά για εφαρμογές που απαιτούν συνεχή ανταλλαγή δεδομένων, όπως παιχνίδια ή chat εφαρμογές.

REST APIs:
Το REST (Representational State Transfer) είναι μια αρχιτεκτονική για την ανάπτυξη δικτυακών εφαρμογών. Το κύριο χαρακτηριστικό του REST API είναι ότι λειτουργεί χωρίς να διατηρεί κατάσταση (stateless). Stateless σημαίνει ότι οι διακομιστές δεν αποθηκεύουν δεδομένα του πελάτη μεταξύ των αιτημάτων [49]. Τα REST APIs χρησιμοποιούν HTTP αιτήματα για την εκτέλεση λειτουργιών, όπως GET για τη λήψη δεδομένων, POST για τη δημιουργία νέων στοιχείων, PUT για την ενημέρωση υπαρχόντων και DELETE για την αφαίρεσή τους. Τα REST APIs είναι ευέλικτα και ευρέως διαδεδομένα λόγω της απλότητάς τους.

4.1.5 Postman

Το Postman είναι ένα λογισμικό για τη δοκιμή και ανάπτυξη API (Application Programming Interface). Χρησιμοποιείται από προγραμματιστές για να στείλουν αιτήματα HTTP σε web servers και να δουν τις απαντήσεις τους. Το Postman παρέχει μια εύχρηστη γραφική διεπαφή χρήστη, καθιστώντας το εύκολο στην εγκατάσταση και τη χρήση, ακόμα και για αυτούς που δεν είναι πολύ εξοικειωμένοι με τον προγραμματισμό.

Ένα από τα βασικά πλεονεκτήματα του Postman είναι η δυνατότητα δημιουργίας και αποθήκευσης συλλογών από αιτήματα, γεγονός που βοηθά τους προγραμματιστές να οργανώσουν και να ανακαλύψουν εύκολα τα API τους. Επιπλέον, υποστηρίζει διάφορες μεθόδους HTTP όπως GET, POST, DELETE και PUT.

Η ικανότητα του Postman να ενσωματώνει τεστ περιβάλλον και να διαχειρίζεται μεταβλητές το καθιστά πολύτιμο για την εξομοίωση διαφορετικών δοκιμών. Η δυνατότητα αυτή βοηθά στην επιβεβαίωση της συμπεριφοράς των API σε διάφορα περιβάλλοντα και υποσυστήματα. Συνολικά, το Postman είναι ένα ισχυρό εργαλείο που βοηθά τους προγραμματιστές να αναπτύσσουν, δοκιμάζουν και να ενσωματώνουν APIs με αποδοτικό και απλό τρόπο, διευκολύνοντας τη διαδικασία ανάπτυξης λογισμικού [50].

4.2 Client Side

Το client side, αναφέρεται σε δραστηριότητες που λαμβάνουν χώρα στην πλευρά του πελάτη (client) σε ένα επίπεδο πελάτη-διακομιστή (client-server) στον υπολογιστικό κόσμο. Σε αυτό το πλαίσιο, ο ”πελάτης” συνήθως αναφέρεται στο λογισμικό (όπως ένας web browser) που τρέχει στη συσκευή του χρήστη, σε αντίθεση με τον ”διακομιστή” (server), που είναι ένας απομακρυσμένος υπολογιστής που παρέχει δεδομένα και υπηρεσίες. Στο πλαίσιο του web development, ο όρος ”client-side” συνήθως αναφέρεται στην εκτέλεση κώδικα στον web browser του χρήστη. Αυτό σημαίνει ότι οποιαδήποτε επεξεργασία, αλλαγή, ή διαχείριση δεδομένων που γίνεται στην πλευρά του πελάτη, λαμβάνει χώρα εντός του browser του χρήστη και όχι στον διακομιστή. Παρακάτω θα αναλύσουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής AutoDiscretizer στο client side:

4.2.1 HTML

Η HTML, ή HyperText Markup Language, αποτελεί ένα σύστημα κανόνων και οδηγιών για τη διαμόρφωση και την παρουσίαση του περιεχομένου σε μια ιστοσελίδα. Δεν θεωρείται γλώσσα προγραμματισμού στην παραδοσιακή έννοια, αλλά μια γλώσσα σήμανσης που καθορίζει τις ιδιότητες και τη δομή των διάφορων στοιχείων ενός ιστοτόπου. Ξεκίνησε ως μια απλή ιδέα για να διευκολύνει την ανταλλαγή εγγράφων και πληροφοριών μεταξύ ερευνητών και σήμερα έχει εξελιχθεί σε μια πλήρη γλώσσα που ορίζει τον τρόπο που εμφανίζονται κείμενα, εικόνες και άλλα στοιχεία στο διαδίκτυο.

Τα στοιχεία της HTML αποκαλούνται tags, τα οποία υποδεικνύουν στον περιηγητή πώς να εμφανίσει το κείμενο, τις εικόνες και άλλα στοιχεία. Κάθε ιστοσελίδα ξεκινά με το tag <html> και κλείνει με το αντίστοιχο </html>, δημιουργώντας ένα πλαίσιο μέσα στο οποίο αναπτύσσεται όλο το περιεχόμενο. Οι περισσότερες ιστοσελίδες περιλαμβάνουν αρκετές διαφορετικές σελίδες HTML. Για παράδειγμα, μια αρχική σελίδα (home page), μια σελίδα σχετικά με τον ιστότοπο (about), και μια σελίδα επικοινωνίας (contact) θα είχαν όλες ξεχωριστά αρχεία HTML. Τα έγγραφα HTML είναι αρχεία που τελειώνουν με την επέκταση .html ή .htm. Ένας web browser διαβάζει το αρχείο HTML και αποδίδει το περιεχόμενό του, ώστε οι χρήστες του διαδικτύου να μπορούν να το δουν [51]. Παρακάτω είναι ένα απλό παράδειγμα κώδικα HTML:

```html
<html>
<head>
    <title>Απλή Σελίδα HTML</title>
</head>
<body>
    <h1 class="important-text">Καλωσήρθατε στην ιστοσελίδα.</h1>
</body>
</html>
```

Σχήμα 4.2: Παράδειγμα HTML

4.2.2 CSS

Το Cascading Style Sheets, που είναι πιο γνωστό ως CSS, αποτελεί μία γλώσσα ορισμού στυλ, η οποία χρησιμοποιείται για να καθορίσει τον τρόπο εμφάνισης ενός εγγράφου που έχει δημιουργηθεί με μια γλώσσα διακριτικών στοιχείων (tags), όπως η HTML. Το CSS είναι το κλειδί για τη δημιουργία ελκυστικών, συνεπών και εύχρηστων ιστοσελίδων. Το CSS λειτουργεί με την αρχή της επιλογής στοιχείων από το HTML έγγραφο και την εφαρμογή στυλ σε αυτά. Μια επιλογή μπορεί να είναι το όνομα ενός HTML tag, μια ταυτότητα ID, μια κλάση, ή ακόμη και μια πιο περίπλοκη έκφραση που περιγράφει σχέσεις μεταξύ των στοιχείων. Η δύναμη του CSS έγκειται στην ικανότητά του να εφαρμόζει διάφορα στυλ ανάλογα με το πλαίσιο, όπως στην απόκριση σε διάφορες διαστάσεις οθόνης ή συσκευών.

Η χρήση του CSS δεν περιορίζεται μόνο στην αλλαγή των χρωμάτων και του μεγέθους του κειμένου. Με το CSS, μπορείτε να δημιουργήσετε περίπλοκες διατάξεις, να εφαρμόσετε animated εφέ και μεταβάσεις, να ελέγξετε το πλαίσιο και τα περιθώρια των στοιχείων, και πολλά άλλα. Επιπλέον, το CSS προσφέρει δυνατότητες για την προσαρμογή των ιστοσελίδων για διαφορετικές συσκευές και μεγέθη οθόνης, γνωστό ως responsive design [51]. Παρακάτω είναι ένα απλό παράδειγμα κώδικα CSS:

```css
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}

.important-text {
    font-size: 1.5em;
    font-weight: bold;
}
```

Σχήμα 4.3: Παράδειγμα CSS

Στο συγκεκριμένο παράδειγμα, έχουμε τρία κομμάτια CSS κώδικα. Στην αρχή ορίζουμε το χρώμα φόντου (background-color) για το σώμα (body) του εγγράφου HTML. Στην περίπτωση

αυτή, το φόντο έχει ένα ανοιχτό μπλε χρώμα (lightblue). Στην συνέχεια, εφαρμόζουμε στυλ στα στοιχεία h1 (κεφαλίδες πρώτου επιπέδου). Ορίζει το χρώμα του κειμένου σε μπλε (navy) και προσθέτει ένα αριστερό περιθώριο (margin-left) των 20 pixel και τέλος ορίζουμε ένα στυλ για οποιοδήποτε HTML στοιχείο έχει ανατεθεί η κλάση important-text. Το στυλ αυτό αυξάνει το μέγεθος της γραμματοσειράς (font-size) σε 1.5 φορές το μέγεθος των γραμμάτων του γονικού στοιχείου και ορίζει το βάρος της γραμματοσειράς (font-weight) σε έντονο (bold).

4.2.3 Bootstrap

Το Bootstrap είναι ένα πλαίσιο ανάπτυξης (framework) για το web design και την ανάπτυξη ιστοσελίδων. Αναπτύχθηκε αρχικά από τους Mark Otto και Jacob Thornton του Twitter, και κυκλοφόρησε το 2011. Στόχος του ήταν να διευκολύνει τη δημιουργία συνεπών και διαισθητικών διεπαφών χρηστών σε διάφορες συσκευές και μεγέθη οθόνης, χρησιμοποιώντας responsive design τεχνικές [52]. Το Bootstrap είναι ένα δημοφιλές πλαίσιο ανάπτυξης για το web design και την ανάπτυξη ιστοσελίδων, το οποίο χαρακτηρίζεται από την παροχή ενός ευέλικτου συστήματος grid, το οποίο βοηθά στη δημιουργία διεπαφών που προσαρμόζονται σε διάφορες αναλύσεις και μεγέθη. Επιπλέον, περιλαμβάνει ένα πλήθος έτοιμων στοιχείων διεπαφής χρήστη, όπως κουμπιά, φόρμες, καρτέλες, και nav bars, τα οποία μπορούν να ενσωματωθούν και να προσαρμοστούν εύκολα σε ιστοσελίδες.

Ένα σημαντικό χαρακτηριστικό του Bootstrap είναι η υποστήριξη του responsive design, που επιτρέπει στις ιστοσελίδες να προσαρμόζονται αυτόματα στο μέγεθος της οθόνης της συσκευής που τις προβάλλει, είτε πρόκειται για κινητό, tablet ή desktop. Επιπρόσθετα, το Bootstrap παρέχει πολλά JavaScript plugins, τα οποία προσθέτουν δυναμικά στοιχεία στις ιστοσελίδες, όπως μονταλ παράθυρα, tabs, και tooltips. Το Bootstrap είναι επίσης γνωστό για την ευελιξία και την επεκτασιμότητά του, καθώς μπορεί να προσαρμοστεί με custom CSS. Η μεγάλη κοινότητα του Bootstrap συνεχίζει να προσφέρει νέες ενημερώσεις, plugins και λύσεις σε προβλήματα, κάτι που το καθιστά ιδιαίτερα δημοφιλές για web developers σε όλο τον κόσμο. Συνοψίζοντας, το Bootstrap είναι ένα εργαλείο που διευκολύνει τη γρήγορη και αποτελεσματική ανάπτυξη διαδικτυακών εφαρμογών και ιστοσελίδων, ενώ ταυτόχρονα διασφαλίζει την ομοιομορφία και την ποιότητα του τελικού προϊόντος.

4.2.4 JavaScript

Η JavaScript είναι μια ελαφριά γλώσσα προγραμματισμού που χρησιμοποιείται συχνά από web developers για να προσθέσουν δυναμικές αλληλεπιδράσεις σε ιστοσελίδες και εφαρμογές. Λειτουργεί αρμονικά μαζί με HTML και CSS, συμπληρώνοντας το CSS στη μορφοποίηση των στοιχείων HTML ενώ παράλληλα παρέχει δυνατότητες αλληλεπίδρασης με τον χρήστη, μια δυνατότητα που το CSS μόνο του δεν διαθέτει.

Η JavaScript είναι γνωστή για την ικανότητά της να επιτρέπει τη δημιουργία δυναμικών στοιχείων UI, όπως αναδυόμενα παράθυρα και διαδραστικές φόρμες. Οι βασικές λειτουργίες της JavaScript περιλαμβάνουν τη διαχείριση στοιχείων HTML, τη δημιουργία διαδραστικών γραφικών και την επεξεργασία δεδομένων πριν από την αποστολή σε έναν server. Η JavaScript επίσης έχει πολλά frameworks και βιβλιοθήκες, όπως η React, Vue.js και jQuery τα οποία βοη-

θούν στην ανάπτυξη εξελιγμένων και καλά οργανωμένων εφαρμογών. Επιπλέον, μία από τις πιο σημαντικές πτυχές της JavaScript είναι η ασύγχρονη επεξεργασία, κάτι που επιτρέπει στις εφαρμογές να εκτελούνται αποδοτικά χωρίς να διακόπτεται η εμπειρία του χρήστη. Το AJAX (Asynchronous JavaScript and XML) είναι μια τεχνική που χρησιμοποιείται ευρέως για τη δημιουργία διαδραστικών web εφαρμογών, επιτρέποντας την ενημέρωση τμημάτων μιας ιστοσελίδας χωρίς να χρειάζεται να φορτώσει εκ νέου ολόκληρη η σελίδα [53].

Η JavaScript είτε ενσωματώνεται απευθείας σε μια ιστοσελίδα είτε αναφέρεται μέσω ενός ξεχωριστού αρχείου με κατάληξη .js. Όταν ένας χρήστης επισκέπτεται αυτή την ιστοσελίδα, ο περιηγητής του θα εκτελέσει το script μαζί με τον κώδικα HTML και CSS δημιουργώντας μια λειτουργική σελίδα που εμφανίζεται μέσω της καρτέλας του περιηγητή. Ο javascript κώδικας κατεβαίνει στις συσκευές των επισκεπτών και επεξεργάζεται εκεί. Αυτό διαφέρει από μια γλώσσα που λειτουργεί στην πλευρά του διακομιστή (server side), όπου ο διακομιστής επεξεργάζεται το σενάριο πριν το στείλει στον περιηγητή. Όταν συναντάει ένα μπλοκ κώδικα JavaScript, ένας περιηγητής ιστού θα το επεξεργαστεί από την κορυφή προς τα κάτω, δεδομένου ότι είναι ευαίσθητο στη σειρά εκτέλεσης [53]. Η JavaScript, αποτελεί μια από τις πιο δημοφιλείς και δυναμικές γλώσσες προγραμματισμού. Η ικανότητά της να προσθέτει διαδραστικότητα και δυναμική λειτουργικότητα σε ιστοσελίδες σε συνεργασία με HTML και CSS, τη καθιστά αναντικατάστατη στον σύγχρονο ιστό. Ακολουθεί ένα απλό παράδειγμα κώδικα HTML, το οποίο ενσωματώνει JavaScript μέσα σε <script> tag.

```
<html>
<head>
    <title>JavaScript Example</title>
</head>
<body>

<h2>A Simple JavaScript Example</h2>

<p id="test">This is a paragraph.</p>

<button onclick="changeText()">Click Me!</button>

<script>
    function changeText() {
        document.getElementById("test").innerHTML = "Text has been changed!";
    }
</script>

</body>
</html>
```

Σχήμα 4.4: Παράδειγμα JavaScript

4.2.5 jQuery

Η jQuery είναι μια πλούσια σε δυνατότητες βιβλιοθήκη JavaScript. Είναι ευρέως γνωστή για την ευκολία χρήσης της σε διάφορα web projects, καθώς κάνει πιο απλές πολλές εργασίες που απαιτούν πολλές γραμμές κώδικα σε JavaScript. Τα κύρια χαρακτηριστικά της jQuery περιλαμβάνουν τη διευκόλυνση σε διάφορες πτυχές του web development. Αυτά περιλαμβάνουν

τον απλοποιημένο χειρισμό της διαχείρισης HTML (HTML manipulation), την επεξεργασία του Document Object Model (DOM manipulation), τη διαχείριση των Cascading Style Sheets (CSS manipulation), καθώς και την παροχή εργαλείων για τη δημιουργία εφέ και animations. Επιπλέον, η jQuery διευκολύνει την εκτέλεση ασύγχρονων αιτήσεων μέσω AJAX και προσφέρει μεθόδους για HTML events, βελτιώνοντας έτσι τη δυναμικότητα και την αλληλεπίδραση σε web εφαρμογές [54].

Τέλος, η jQuery προσφέρει μια πλούσια συλλογή προσθέτων (plugins), η οποία επιτρέπει στους developers να επεκτείνουν τις δυνατότητες της βιβλιοθήκης για να καλύψουν συγκεκριμένες ανάγκες. Αυτή η ευελιξία τη καθιστά ένα πολύτιμο εργαλείο για την ανάπτυξη τόσο απλών όσο και περίπλοκων web εφαρμογών. Συνοψίζοντας, η jQuery αποτελεί μια εξαιρετικά χρήσιμη βιβλιοθήκη JavaScript που προσφέρει απλότητα και ευελιξία στην ανάπτυξη web εφαρμογών. Μέσω της αποδοτικής σύνταξης, η jQuery βοηθά τους developers να επιτύχουν περίπλοκες λειτουργίες με λιγότερο και πιο καθαρό κώδικα. Ακολουθεί ένα παράδειγμα που συγκρίνει τον κώδικα JavaScript με αντίστοιχο κώδικα jQuery:

Javascript:

```
1 document.getElementById("test").innerHTML = "Hello World!";
```

jQuery:

```
1 $("#test").html("Hello World!");
```

Αυτό το σύντομο παράδειγμα δείχνει πώς η jQuery μπορεί να πετύχει το ίδιο αποτέλεσμα με την JavaScript με μια πιο συνοπτική μέθοδο.

Κεφάλαιο 5

Υλοποίηση του AutoDiscretizer

5.1 Προδιαγραφές User Stories

Τα ”User Stories” ή ”Ιστορίες Χρηστών” είναι ένας τρόπος προσέγγισης στην ανάπτυξη εφαρμογών. Πρόκειται για σύντομες, περιεκτικές περιγραφές λειτουργικότητας από την οπτική γωνία του τελικού χρήστη. Στόχος τους είναι να παρουσιάσουν τις ανάγκες και τις επιθυμίες του χρήστη με έναν απλό και κατανοητό τρόπο.

Ένα User Story περιλαμβάνει τρία βασικά στοιχεία:

1. **Ποιος**: Ο χρήστης ή η ομάδα χρηστών που θα επωφεληθεί από τη λειτουργία.
2. **Τι**: Η λειτουργικότητα που αναφέρεται στην ιστορία.
3. **Γιατί**: Ο λόγος που αυτή η λειτουργία είναι σημαντική ή χρήσιμη για τον χρήστη.

Τα User Stories βοηθούν τις ομάδες ανάπτυξης εφαρμογών να εστιάσουν στις ανάγκες του χρήστη και να δημιουργήσουν λειτουργίες που πραγματικά προσθέτουν αξία στο τελικό προϊόν. Επιπλέον, διευκολύνουν την επικοινωνία μεταξύ των μελών της ομάδας και των πελατών ή των χρηστών, ενώ παράλληλα βοηθούν στην οργάνωση και τον προγραμματισμό της δουλειάς [55]. Σε αυτήν την ενότητα θα δουμε τα User Stories για την εφαρμογή AutoDiscretizer:

Τίτλος: Διακριτοποίηση αριθμητικών δεδομένων σε ενα σύνολο δεδομένων με την εφαρμογή AutoDiscretizer.

Ως χρήστης,

Θέλω μια εφαρμογή που να μου επιτρέπει να ανεβάζω ένα σύνολο δεδομένων με τουλάχιστον μια αριθμητική στήλη, να επιλέγω συγκεκριμένες στήλες για διακριτοποίηση, να επιλέγω μέθοδο διακριτοποίησης (Ομοιόμορφη, Ποσοστιαία ή K-means) και να καθορίζω τον αριθμό των διακριτών ομάδων. Επιπλέον, θα έχω την δυνατότητα να επιλέγω αυτόματη επιλογή μεθόδου ή αυτόματη επιλογή αριθμού διακριτών ομάδων.

Ωστε να μπορώ να διακριτοποιώ τα δεδομένα μου αποδοτικά, να συγκρίνω την απόδοση διαφορετικών μεθόδων διακριτοποίησης και αριθμών διακριτών ομάδων βάσει κατηγοριοποίησης Naive Bayes, και να λαμβάνω ένα βελτιστοποιημένο σύνολο διακριτοποιημένων δεδομένων για τις αναλυτικές μου ανάγκες.

Πιο αναλυτικά οι λειτουργικές απαιτήσεις της διαδικτυακής εφαρμογής AutoDiscretizer:

1. **Αρχική Σελίδα**
   Είναι μια αρχική σελίδα η οποία θα λειτουργεί ως εισαγωγή για την εφαρμογή, θα προσφέρει μια πλήρη περιγραφή της λειτουργίας της, των υπηρεσιών που παρέχει και των βασικών χαρακτηριστικών της, καθώς και μια ανάλυση του τρόπου λειτουργίας της.

2. **Σελίδα KBins**
   Αυτή η σελίδα θα αποτελεί τη διεπαφή χρήστη για την εφαρμογή AutoDiscretizer, προσφέροντας έναν διαισθητικό και αποτελεσματικό τρόπο διαχείρισης των λειτουργιών της.

3. **Ανέβασμα αρχείου δεδομένων**
   Ο χρήστης θα μπορεί να ανεβάσει ένα αρχείο δεδομένων, εφόσον αυτό έχει επέκταση csv.

4. **Εμφάνιση αρχείου**
   Ο χρήστης θα μπορεί να δεί ένα στιγμιότυπο των πρώτων 20 γραμμών του αρχείου δεδομένων που έχει ανεβάσει σε μορφή πίνακα.

5. **Επιλογή γνωρισμάτων**
   Ο χρήστης θα έχει τη δυνατότητα να επιλέξει τις στήλες με τα γνωρίσματα που επιθυμεί να μετατραπούν σε κατηγορικά μέσω διακριτοποίησης.

6. **Μέθοδος διακριτοποίησης**
   Ο χρήστης θα μπορεί να επιλέξει τη μέθοδο διακριτοποίησης που επιθυμεί (Ομοιόμορφη, Ποσοστιαία ή K-means).

7. **Επιλογή Bins**
   Ο χρήστης θα μπορεί να ορίσει τον επιθυμητό αριθμό διακριτικών ομάδων (κατηγοριών) για την διακριτοποίηση.

8. **Επιλογή Auto**
   Ο χρήστης θα μπορεί να επιλέξει είτε την αυτόματη επιλογή μεθόδου, είτε αυτόματη επιλογή των bins, ή και τα δύο. Σε περίπτωση που ο χρήστης επιλέξει την αυτόματη επιλογή, τότε θα εφαρμόζεται κατηγοριοποίηση Naive Bayes στα δεδομένα και θα επιλέγετε η μέθοδος και ο αρθμός των bins με την υψηλότερη ακρίβεια.

9. **Διακριτοποίηση Δεδομένων**
   Ο χρήστης θα μπορεί να δεί σε μορφή πίνακα τις πρώτες 20 γραμμές του αρχείου δεδομένων που έχουν υποστεί διακριτοποίηση και θα έχει την επιλογή να το κατεβάσει ολόκληρο σε μορφή CSV.

10. **Αξιολόγηση**
    Αν επιλεγεί η αυτόματη λειτουργία, θα παρουσιάζεται επιπρόσθετα ένας πίνακας με την ακρίβεια, την επιλεγμένη μέθοδο και των αριθμό των bins που καθορίστηκαν αυτόματα.

11. **Ελεύθερο API**
    Θα διαθέτει ελεύθερο web API όπου προγραμματιστές θα μπορούν να χρησιμοποιούν τις λειτουργίες της εφαρμογής μέσω http requests.

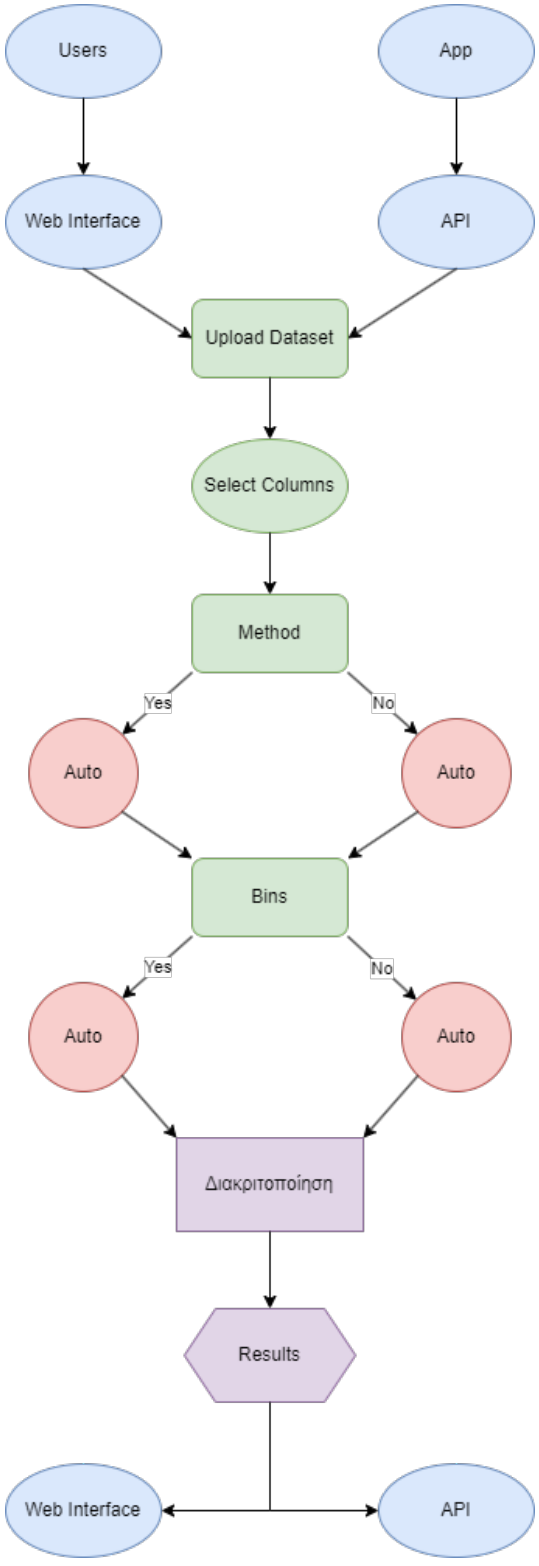
5.2 Η αρχιτεκτονική του AutoDiscretizer

Η διαδικτυακή εφαρμογή AutoDiscretizer έχει σκοπό τη διακριτοποίηση συνόλων δεδομένων με τη χρήση των βιβλιοθηκών scikit-learn της Python. Επιλέον, προσφέρει τη δυνατότητα αυτόματης επιλογής της μεθόδου διακριτοποίησης και του αριθμού των bins (κατηγοριών), βασιζόμενη στην εφαρμογή του αλγορίθμου Naive Bayes στα διακριτοποιημένα δεδομένα και στα υπόλοιπα κατηγορικά δεδομένα του συνόλου. Η επιλογή της καλύτερης μεθόδου ή του αριθμού των bins γίνεται με βάση το μέγιστο accuracy που επιτυγχάνεται από τον Naive Bayes. Ο συγκεκριμένος Naive Bayes που χρησιμοποιείται είναι ο Categorical Naive Bayes της βιβλιοθήκης scikit-learn.

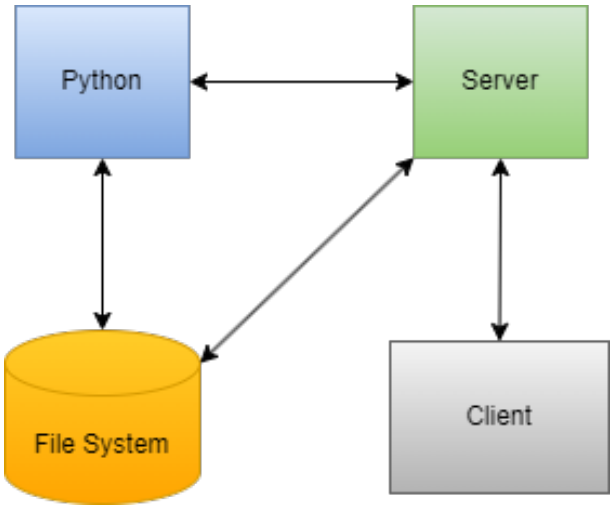
Επιπλέον, η εκτέλεση όλων των παραπάνω λειτουργιών μπορεί να πραγματοποιηθεί μέσω του κατάλληλου διαμορφωμένου API της εφαρμογής. Η ανάπτυξη του API έχει γίνει χρησιμοποιώντας PHP, η οποία ενσωματώνεται με Python scripts για την εκτέλεση της διακριτοποίησης και των λειτουργιών αυτόματης επιλογής. Μέσω της κλήσης των κατάλληλων endpoints του API, συνοδευόμενη από τις απαιτούμενες παραμέτρους, οι χρήστες μπορούν να αξιοποιήσουν τις λειτουργίες της εφαρμογής AutoDiscretizer. Τα αποτελέσματα της διακριτοποίησης και της αυτόματης επιλογής παρέχονται σε μορφή JSON, προσφέροντας έτσι την δυνατότητα ενσωμάτωσης του API σε διάφορες εφαρμογές και συστήματα.

Το front-end της εφαρμογής AutoDiscretizer και η ανάπτυξη του περιβάλλοντος χρήστη (UI) έχουν υλοποιηθεί με τη χρήση τεχνολογιών όπως JavaScript, πιο συγκεκριμένα τη βιβλιοθήκη jQuery για την απλοποίηση της δυναμικής διαδραστικότητας. Επιπλέον, έχουν χρησιμοποιηθεί HTML και CSS για τη δημιουργία και στυλιστική διαμόρφωση της διεπαφής, ενώ η χρήση του Bootstrap προσδίδει μια συνεπή και ανταποκρίσιμη σχεδίαση, που εγγυάται την ομαλή λειτουργία της διεπαφής σε διάφορες συσκευές και μεγέθη οθόνης. Το UI είναι εύχρηστο και προσαρμοστικό, προσφέροντας μια αποτελεσματική εμπειρία για τον τελικό χρήστη, με έμφαση στην απλότητα και την άμεση πρόσβαση στις λειτουργίες της εφαρμογής.

Τέλος, η εφαρμογή AutoDiscretizer είναι ανοιχτή και προσφέρεται δωρεάν σε όλους τους χρήστες που ενδιαφέρονται να εξερευνήσουν και να χρησιμοποιήσουν τις δυνατότητές της. Επιπλέον, το repository της εφαρμογής είναι διαθέσιμο στο GitHub, επιτρέποντας στους χρήστες να μελετήσουν τη δομή και τη λειτουργία της εφαρμογής, προσφέροντας έτσι μια εκπαιδευτική ευκαιρία για εκείνους που ενδιαφέρονται να μάθουν περισσότερα για τις τεχνικές και τις τεχνολογίες που χρησιμοποιήθηκαν.



Σχήμα 5.1: Διάγραμμα ροής AutoDiscretizer



Σχήμα 5.2: Διάγραμμα αρχιτεκτονικής AutoDiscretizer

5.3 Υλοποίηση του Back-End

Σε αυτή την ενότητα, θα αναλύσουμε πιο λεπτομερώς το πρακτικό τμήμα του back-end που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής AutoDiscretizer.

API

Το API αποτελεί ένα σημαντικό στοιχείο της εφαρμογής. Χρησιμοποιώντας το API, οι χρήστες έχουν τη δυνατότητα να αξιοποιήσουν τις λειτουργίες της εφαρμογής είτε μέσω του διαδραστικού περιβάλλοντος (front-end) είτε μέσω της δικής τους εφαρμογής, καλώντας τα σχετικά API endpoints (πίνακα 5.1). Ένα API endpoint ουσιαστικά είναι ένα URI (Uniform Resource Identifier), το οποίο, κατά την κλήση του, ενεργοποιεί ένα script στον διακομιστή (server), προκαλώντας έτσι την εκτέλεση ειδικών λειτουργιών της εφαρμογής.

Το API της εφαρμογής αναπτύχθηκε χρησιμοποιώντας τη γλώσσα PHP, επιλογή που προσδίδει ευελιξία και αποδοτικότητα στη διαχείριση των δεδομένων και των επικοινωνιών μεταξύ της εφαρμογής και του διακομιστή. Κάθε endpoint απαιτεί τις αντίστοιχες παραμέτρους για να λειτουργήσει σωστά. Ο τρόπος διαβίβασης των παραμέτρων ποικίλλει ανάλογα με τη μέθοδο που εφαρμόζει το συγκεκριμένο endpoint.

Για παράδειγμα, το endpoint /api/read_dataset.php λειτουργεί χρησιμοποιώντας τη μέθοδο GET. Γι' αυτόν τον λόγο, είναι απαραίτητο να διαβιβάσουμε τις απαιτούμενες παραμέτρους μέσω του URI. Το συγκεκριμένο endpoint δέχεται δύο παραμέτρους: το όνομα του dataset και μια boolean παράμετρο, ονομαζόμενη 'binned'. Η παράμτερος binned μπορεί να λάβει τις τιμές 'true' ή 'false', καθορίζοντας εάν θέλουμε να ανακτήσουμε τα δεδομένα του διακριτοποιημένου dataset ή του αρχικού. Για παράδειγμα, αν θέλουμε να ανακτήσουμε το dataset με την ονομασία 'iris.csv', το οποίο έχει διακριτοποιημένα χαρακτηριστικά, τότε θα πρέπει να καλέσουμε το URI: /api/read_dataset.php?dataset=iris.csv&binned=true.

Method	API Endpoint
GET	`/api/read_dataset.php` - Ανάκτηση Συνόλου Δεδομένων (Dataset)
GET	`/api/download_dataset.php` - Λήψη Διακριτοποιημένου Συνόλου Δεδομένων (Dataset)
POST	`/api/upload_dataset.php` - Μεταφόρτωση Συνόλου Δεδομένων (Dataset) στον Server
POST	`/api/KBinsDiscretizer.php` - Εφαρμογή Διακριτοποίησης σε ένα Σύνολο Δεδομένων (Dataset)
POST	`/api/auto_methods.php` - Εφαρμογή Διακριτοποίησης με αυτόματη επιλογή (μεθόδου, bins) σε ένα Σύνολο Δεδομένων (Dataset)

Πίνακας 5.1: API Endpoints

Σε περίπτωση που θέλουμε να ανεβάσουμε ένα dataset στον διακομιστή, χρησιμοποιούμε το endpoint /api/upload_dataset.php, το οποίο λειτουργεί μέσω της μεθόδου POST. Σε αντίθεση με τη μέθοδο GET, όπου οι παράμετροι περνούν μέσω του URI, στη μέθοδο POST οι παράμετροι ενσωματώνονται στο σώμα (body) του αιτήματος προς το συγκεκριμένο endpoint.

Παρακάτω, θα δούμε μια πιο λεπτομερή ανάλυση όλων των API Endpoints, καθώς και των παραμέτρων τους, παρέχοντας παράλληλα παραδείγματα για καλύτερη κατανόηση:

GET /api/read_dataset.php
Δέχεται δύο παραμέτρους για την ανάκτηση ενός συγκεκριμένου συνόλου δεδομένων.

1. `dataset`: Το όνομα του dataset που ο χρήστης επιθυμεί να ανακτήσει.
2. `binned`: Μια boolean παράμετρος που δέχεται τις τιμές `true` ή `false`. Αν η τιμή είναι `true`, επιστρέφει το διακριτοποιημένο dataset. Αν η τιμή είναι `false`, επιστρέφεται το αρχικό dataset.

Επιστρέφει τα δεδομένα του επιλεγμένου dataset μαζί με τις στήλες που περιέχουν αριθμητικά δεδομένα, οι οποίες είναι επιλέξιμες για διακριτοποίηση. Επίσης, επιστρέφει τις στήλες με ακέραιες αριθμητικές τιμές ή κατηγορικά δεδομένα, που μπορούν να χρησιμοποιηθούν για την κλάση του Categorical Naive Bayes, εφόσον ο χρήστης επιλέξει κάποια αυτόματη επιλογή.

Παράδειγμα

GET api/read_dataset.php?dataset=iris.csv&binned=false

Response:

```
{
    "numericColumns": [
        "sepal.length",
        "sepal.width",
        "petal.length",
        "petal.width"
    ],
    "categoricalIntegerColumns": [
        "variety"
    ],
    "dataset": [
        [
            "sepal.length",
            "sepal.width",
            "petal.length",
            "petal.width",
            "variety"
        ],
        [
            "5.1",
            "3.5",
            "1.4",
            "0.2",
            "Setosa"
        ],
        [
            "4.9",
            "3",
            "1.4",
            "0.2",
            "Setosa"
        ],
        // Επιπλέον δεδομένα παραλείπονται για συντομία
    ]
}
```

Αρχικά, γίνεται έλεγχος για την επιλογή της μεθόδου GET ως την κατάλληλη για το συγκεκριμένο endpoint. Στη συνέχεια, πραγματοποιείται έλεγχος για το αν έχουν οριστεί οι παράμετροι 'dataset' και 'binned' (σχήμα 5.3). Σε περίπτωση που αυτές οι παράμετροι δεν έχουν οριστεί ή εάν η μέθοδος δεν είναι η σωστή για αυτό το endpoint, τότε εμφανίζεται το ανάλογο μήνυμα σφάλματος (σχήμα 5.4).

```php
if ($_SERVER['REQUEST_METHOD'] === 'GET') {

    if (isset($_GET['dataset']) && isset($_GET['binned'])) {

        $originalFilename = $_GET['dataset'];
        $binned = $_GET['binned'];

        if ($binned === 'false') {
            $dataset_path = '../datasets/';
        } elseif ($binned === 'true') {
            $dataset_path = '../binned_datasets/';
        } else {
            header('HTTP/1.1 400 Bad Request');
            echo 'Error: Unsupported binned value';
            exit;
        }

```

Σχήμα 5.3: Κώδικας επαλήθευσης παραμέτρων

```php
    } else {
        header('HTTP/1.1 400 Bad Request');
        echo 'Error: Required parameters have not been specified.';
    }
} else {
    header("HTTP/1.1 403 Forbidden");
    echo 'Invalid request method';
}
```

Σχήμα 5.4: Κωδικός επαλήθευσης σφάλματος

Όπως αναφέραμε προηγουμένως, το endpoint /api/read_dataset.php επιστρέφει επίσης τις στήλες με αριθμητικές τιμές (κώδικας βλ. σχήμα 5.5) οι οποίες είναι διαθέσιμες για διακριτοποίηση και τις στήλες που περιέχουν ακέραιες αριθμητικές τιμές ή κατηγορικά δεδομένα (κώδικας βλ. σχήμα 5.6), οι οποίες είναι διαθέσιμες για επιλογή κλάσης του categorical naive bayes.

```php
$numericColumns = array();
foreach ($data[0] as $index => $columnName) {
    $isNumeric = true;
    for ($i = 1; $i < count($data); $i++) {
        if (!is_numeric($data[$i][$index])) {
            $isNumeric = false;
            break;
        }
    }
    if ($isNumeric) {
        $numericColumns[] = $columnName;
    }
}
```

Σχήμα 5.5: Κώδικας για αριθμητικές τιμές

```php
$categoricalIntegerColumns = array();
foreach ($data[0] as $index => $columnName) {
    $isInteger = true;
    $hasCategorical = false;
    for ($i = 1; $i < count($data); $i++) {
        $value = $data[$i][$index];
        if (!is_numeric($value) || strpos($value, '.') !== false) {
            $isInteger = false;
        }
        if (!is_numeric($value)) {
            $hasCategorical = true;
        }
        if (!$isInteger || $hasCategorical) {
            break;
        }
    }

    if ($isInteger || $hasCategorical) {
        $categoricalIntegerColumns[] = $columnName;
    }
}
```

Σχήμα 5.6: Κώδικας για ακέραιες αριθμητικές τιμές ή κατηγορικές

POST /api/upload_dataset.php

Δέχετε μια παράμετρο για την μεταφόρτωση ενός συνόλου δεδομένων στον διακομιστή. Κατά τη χρήση της μεθόδου POST, ο χρήστης πρέπει να περιλάβει την απαιτούμενη παράμετρο στο σώμα του αιτήματος (request body) και οχι στο URI όπως στη μέθοδος GET.

1. `file`: Το αρχείο του dataset που ο χρήστης επιθυμεί να μεταφόρτωσει στον server.

Επιστρέφει το όνομα του dataset που έχει επιτυχώς μεταφορτωθεί στον διακομιστή.

Παράδειγμα

Request:

```
{
 form-data: file:{iris.csv}
}
```

Response:

```
"iris.csv"
```

Όπως και στο προηγούμενο endpoint, αρχικά πραγματοποιείται έλεγχος για την εγκυρότητα της μεθόδου POST. Στη συνέχεια, εξετάζεται αν έχει οριστεί σωστά το αρχείο που θελουμε να ανεβάσουμε με κατάληξη csv. Επιπλέον, ελέγχει αν υπάρχει τουλάχιστον μια στήλη με αριθμητικά δεδομένα στο συγκεκριμένο αρχείο δεδομένων. Εφόσον όλοι οι έλεγχοι είναι θετικοί, τότε επιστρέφει το όνομα του αρχείου που έχει μεταφορτωθεί στον διακομιστή επιτυχώς (σχήμα 5.7).

```php
if (isset($_FILES['file']) && $_SERVER['REQUEST_METHOD'] === 'POST') {
    $uploadDir = '../datasets/';

    if (!file_exists($uploadDir)) {
        mkdir($uploadDir, 0777, true);
    }

    $originalFilename = basename($_FILES['file']['name']);
    $uploadedFile = $uploadDir . basename($_FILES['file']['name']);
    $fileType = strtolower(pathinfo($uploadedFile, PATHINFO_EXTENSION));

    $allowedExtensions = ['csv'];

    if (!in_array($fileType, $allowedExtensions)) {
        echo 'Invalid file format. Please upload a valid CSV file.';
        exit;
    }

    if (move_uploaded_file($_FILES['file']['tmp_name'], $uploadedFile)) {
        $hasNumericColumn = checkForNumericColumn($uploadedFile);

        if ($hasNumericColumn) {
            header('Content-Type: application/json');
            echo json_encode($originalFilename, JSON_PRETTY_PRINT);
        } else {
            unlink($uploadedFile);
            echo 'no numeric';
```

Σχήμα 5.7: Κώδικας για μεταφόρτωση αρχείου δεδομένων

```php
function checkForNumericColumn($filePath) {

    if (($handle = fopen($filePath, "r")) !== FALSE) {

        $firstLine = fgets($handle);
        $commaCount = substr_count($firstLine, ',');
        $semicolonCount = substr_count($firstLine, ';');
        $delimiter = $commaCount >= $semicolonCount ? ',' : ';';

        rewind($handle);

        $numericColumns = [];
        $headerRead = false;

        while (($data = fgetcsv($handle, 1000, $delimiter)) !== FALSE) {
            if (!$headerRead) {
                foreach ($data as $index => $column) {
                    $numericColumns[$index] = true;
                }
                $headerRead = true;
                continue;
            }

            foreach ($data as $index => $value) {
                if (!is_numeric(trim($value))) {
                    $numericColumns[$index] = false;
                }
            }
        }

        fclose($handle);
        foreach ($numericColumns as $isNumeric) {
            if ($isNumeric) {
                return true;
            }
        }
    }
    return false;
}
```

Σχήμα 5.8: Κώδικας για την επιβεβαίωση ύπαρξης τουλάχιστον μίας αριθμητικής στήλης

Η μέθοδος checkForNumericColumn εξετάζει το αρχείο δεδομένων για να ελέγξει την ύπαρξη τουλάχιστον μίας αριθμητικής στήλης (σχήμα 5.8), κρίσιμη προϋπόθεσης για την εφαρμογή διακριτοποίησης. Για κάθε στήλη στη γραμμή επικεφαλίδων, αρχικοποιεί έναν πίνακα ($numericColumns) που δηλώνει ότι όλες οι στήλες θα ελεγχθούν αν είναι αριθμητικές. Στη συνέχεια, για κάθε μεταγενέστερη γραμμή, ελέγχει κάθε τιμή στην αντίστοιχη στήλη: εάν μια τιμή δεν είναι αριθμητική, θέτει την αντίστοιχη θέση στον πίνακα $numericColumns σε false. Εάν βρει τουλάχιστον έναν δείκτη με τιμή true (δηλαδή, μια στήλη ήταν αριθμητική σε όλες τις γραμμές), επιστρέφει true. Διαφορετικά, επιστρέφει false.

POST /api/KBinsDiscretizer.php

Δέχεται τέσσερις παραμέτρους και εκτελεί διακριτοποίηση στα επιλεγμένα αριθμητικά γνωρίσματα του συνόλου δεδομένων. Παράμετροι:

1. `dataset`: Το όνομα του αρχείου δεδομένων στο οποίο θα εφαρμοστεί διακριτοποίηση.
2. `checkedCheckboxes`: Πίνακας που περιλαμβάνει τις στήλες με αριθμητικά δεδομένα που ο χρήστης έχει επιλέξει για διακριτοποίηση.
3. `strategy`: Η επιλεγμένη μέθοδος διακριτοποίησης από τον χρήστη, η οποία μπορεί να είναι Uniform, Quantile ή Kmeans.
4. `bins`: Ο αριθμός των κατηγοριών (bins) που έχει ορίσει ο χρήστης.

Επιστρέφει μήνυμα που επιβεβαιώνει την επιτυχή ολοκλήρωση της διακριτοποίησης.

Παράδειγμα

Request:

```
{
   "dataset": "iris.csv",
   "checkedCheckboxes": ["sepal.width"],
   "strategy": "Kmeans",
   "bins": "6"
}
```

Response:

```
{"message":["Binned dataset processing was successful"]}
```

Αρχικά, ο κώδικας ελέγχει αν έχει οριστεί η σωστή μέθοδος για το συγκεκριμένο endpoint και αν έχουν οριστεί οι τέσσερις απαιτούμενες παράμετροι (σχήμα 5.9). Σε περίπτωση που δεν έχουν οριστεί σωστά, εμφανίζεται το αντίστοιχο μήνυμα σφάλματος. Στη συνέχεια, εκτελείται το Python script, λαμβάνοντας ως ορίσματα αυτές τις τέσσερις παραμέτρους (σχήμα 5.10).

```php
if (!isset($data['dataset'], $data['checkedCheckboxes'], $data['strategy'], $data['bins'])) {
    log_error('Invalid JSON structure');
    header('HTTP/1.1 400 Bad Request');
    echo json_encode(['error' => 'Invalid JSON structure']);
    exit;
}

    $dataset_name = $data['dataset'];
    $columns = $data['checkedCheckboxes'];
    $strategy = strtolower($data['strategy']);
    $bins = $data['bins'];

    $target_dir = "../datasets/";
    $target_file = $target_dir . basename($dataset_name);
```

Σχήμα 5.9: Κώδικας ελέγχου παραμέτρων

```php

    $pythonScript = "../python/KBinsDiscretizer.py";

    $command = "python $pythonScript " . escapeshellarg($target_file) . " " .
           escapeshellarg($strategy) . " " . escapeshellarg($bins) . " " .
           implode(' ', array_map('escapeshellarg', $columns)) . " 2>&1";

    exec($command, $output, $return_var);

```

Σχήμα 5.10: Κώδικας εκτέλεσης python script μέσω php

Ο κώδικας Python αρχικά αναθέτει στη μεταβλητή csv_file το δεύτερο όρισμα από τη γραμμή εντολών, το οποίο περιέχει το όνομα του αρχείου δεδομένων. Στην Python, η λίστα sys.argv περιέχει τα ορίσματα της γραμμής εντολών. Το sys.argv[0] είναι το όνομα του εκτελούμενου Python script, το οποίο σε αυτή την περίπτωση είναι το KBinsDiscretizer.py. Στη συνέχεια, αναθέτει στη μεταβλητή strategy το τρίτο όρισμα από τη γραμμή εντολών, το οποίο περιέχει τη μέθοδο διακριτοποίησης. Μετά από αυτό, ορίζει τη μεταβλητή bins στην ακέραια τιμή του τέταρτου ορίσματος από τη γραμμή εντολών και τέλος, δημιουργεί τη λίστα selected_columns, η οποία περιέχει τις στήλες που θα υποβληθούν σε διακριτοποίηση (σχήμα 5.11).

```python
csv_file = sys.argv[1]
strategy = sys.argv[2]
bins = int(sys.argv[3])
selected_columns = sys.argv[4:]
```

Σχήμα 5.11: Κώδικας ορισμάτων KBinsDiscretizer.py

Έπειτα, χρησιμοποιεί τη βιβλιοθήκη pandas για να φορτώσει δεδομένα από ένα αρχείο CSV με όνομα csv_file σε ένα DataFrame data. Ελέγχει αν το data περιέχει μόνο μία στήλη. Αν ναι, τότε οι στήλες δεν διαχωρίστηκαν σωστά κατά την αρχική φόρτωση και εκτελεί εκ νέου τη

φόρτωση του αρχείου CSV, αυτή τη φορά χρησιμοποιώντας τον διαχωριστή (;). Με αυτόν τον τρόπο, ο κώδικας είναι σε θέση να διαβάσει σωστά τα δεδομένα είτε τα διαχωριστικά των στηλών είναι κόμματα (,) είτε ερωτηματικά (;). Στην επόμενη φάση, ο κώδικας δημιουργεί ένα νέο DataFrame με την ονομασία X, το οποίο περιλαμβάνει αποκλειστικά εκείνες τις στήλες από το αρχείο δεδομένων που έχουν επιλεγεί για διακριτοποίηση (σχήμα 5.12).

```
14  data = pd.read_csv(csv_file, sep=",", quotechar='"')
15
16  if len(data.columns) == 1:
17      data = pd.read_csv(csv_file, sep=";", quotechar='"')
18
19  selected_columns = list(map(str.strip, selected_columns))
20
21  X = data.loc[:, selected_columns]
```

Σχήμα 5.12: Κώδικας ανάγνωσης αρχείου δεδομένων

Στη συνέχεια, χρησιμοποιούμε τη βιβλιοθήκη scikit-learn για να εκτελέσουμε διακριτοποίηση στις επιλεγμένες στήλες του DataFrame. Ορίζουμε τον αριθμό των κατηγοριών μέσω της παραμέτρου n_bins=bins και επιλέγουμε τη μέθοδο διακριτοποίησης (Uniform, Quantile, Kmeans) μέσω της παραμέτρου strategy. Τέλος, αντικαθιστούμε τις αρχικές συνεχείς τιμές στο DataFrame data με τις νέες διακριτές τιμές από το X_binned (σχήμα 5.13).

```
23  kbins = KBinsDiscretizer(n_bins=bins, encode='ordinal', strategy=strategy)
24  X_binned = kbins.fit_transform(X)
25
26  X_binned = X_binned.astype(int).astype(str)
27
28  data[selected_columns] = X_binned
29
```

Σχήμα 5.13: Κώδικας διακριτοποίησης δεδομένων

POST /api/auto_methods.php

Δέχεται έξι παραμέτρους και εκτελεί διακριτοποίηση, χρησιμοποιώντας είτε αυτόματη επιλογή της μεθόδου διακριτοποίησης, είτε αυτόματο καθορισμό του αριθμού των διακριτών κατηγοριών (bins), είτε και τα δύο. Παράμετροι:

1. `dataset`: Το όνομα του αρχείου δεδομένων στο οποίο θα εφαρμοστεί διακριτοποίηση.
2. `checkedCheckboxes`: Πίνακας που περιλαμβάνει τις στήλες με αριθμητικά δεδομένα που ο χρήστης έχει επιλέξει για διακριτοποίηση.
3. `strategy`: Η επιλεγμένη μέθοδος διακριτοποίησης από τον χρήστη, η οποία μπορεί να είναι Uniform, Quantile, Kmeans ή Auto.
4. `bins`: Ο αριθμός των κατηγοριών (bins) που έχει ορίσει ο χρήστης για τη διαδικασία της διακριτοποίησης.

5. `target_class`: Η κλάση στην οποία θα εφαρμοστεί ο αλγόριθμος Naive Bayes για την αυτόματη επιλογή και αξιολόγηση των παραμέτρων.

6. `autoCheck`: Μεταβλητή που λαμβάνει την τιμή '1' όταν ο χρήστης επιλέγει την αυτόματη επιλογή των bins και '0' όταν δεν την επιλέγει. Εάν ο χρήστης ορίσει την τιμή της μεταβλητής autoCheck σε '1', τότε η παράμετρος bins δεν θα συμμετέχει στο script.

Επιστρέφει τη μέγιστη ακρίβεια (accuracy) που επιτεύχθηκε μέσω της κατηγοριοποίησης με τη χρήση του αλγορίθμου Naive Bayes (στα διακριτοποιημένα δεδομένα και στα κατηγορικά), καθώς και τη μέθοδο και τον αριθμό των bins που συνέβαλαν στην επίτευξη αυτής της ακρίβειας.

Παράδειγμα auto method και auto bins

Request:

```
{
   "dataset": "iris.csv",
   "checkedCheckboxes": ["sepal.length","sepal.width"],
   "strategy": "auto",
   "bins": "7",
   "target_class": "variety",
   "autoCheck": 1
}
```

Response:

```
{
    "output": [
        "{"best_accuracy": 0.8,
        "best_strategy": "quantile",
        "best_bin_number": 3,
        "script": "auto_all"}"
    ]
}
```

Παράδειγμα auto method μόνο

Request:

```
{
   "dataset": "iris.csv",
   "checkedCheckboxes": ["sepal.length","sepal.width"],
   "strategy": "auto",
   "bins": "9",
   "target_class": "variety",
   "autoCheck": 0
}
```

Response:

```
{
    "output": [
        "{"best_accuracy": 0.72,
        "best_strategy": "quantile",
        "script": "auto_strategy"}"
    ]
}
```

Το /api/auto_methods.php είναι υπεύθυνο για την εκτέλεση ενός εκ των τριών Python scripts - auto_strategy.py, auto_bins.py ή auto_all.py - επιλέγοντας το κατάλληλο script βάσει των δοθέντων παραμέτρων. Το script auto_strategy είναι υπεύθυνο για την αυτόματη επιλογή μεθόδου, ενώ το auto_bins χρησιμοποιείται για την αυτόματη επιλογή του πλήθους των κατηγοριών bins. Το auto_all συνδυάζει και τις δύο λειτουργίες, εφαρμόζοντας αυτόματα και την επιλογή μεθόδου και την επιλογή των bins.

Το auto_all.py script (σχήμα 5.14) χρησιμοποιεί τη συνάρτηση train_test_split από τη βιβλιοθήκη scikit-learn για τη διαχωριστική διαίρεση των δεδομένων σε εκπαιδευτικό (training) και δοκιμαστικό (test) σετ. Η μεταβλητή y περιέχει τη στήλη που αντιστοιχεί στην κλάση για τον Naive Bayes. Στη συνέχεια, ο κώδικας με δύο for loops διασχίζει όλους τους δυνατούς αριθμούς των κατηγοριών (bins) από 2 έως 20 και για κάθε μία από τις τρεις στρατηγικές διακριτοποίησης (Uniform, Quantile, Kmeans). Για κάθε συνδυασμό του αριθμού των bins και της στρατηγικής, δημιουργείται ένα αντικείμενο KBinsDiscretizer. Αυτό το αντικείμενο χρησιμοποιείται για να διακριτοποιήσει τα δεδομένα εκπαίδευσης X_train και δοκιμής X_test. Η μέθοδος fit_transform εκπαιδεύει τον KBinsDiscretizer στο X_train και μετατρέπει τα δεδομένα σε διακριτοποιημένη μορφή. Η μέθοδος transform χρησιμοποιείται για να διακριτοποιήσει τα δεδομένα δοκιμής X_test βάσει της εκπαίδευσης που έγινε στο X_train. Στη συνέχεια, δημιουργείται ο κατηγοριοποιητής Naive Bayes για κατηγορηματικά δεδομένα (CategoricalNB). Ο κατηγοριοποιητής χρησιμοποιείται για να προβλέψει τις τιμές για το δοκιμαστικό σετ X_test_binned, και η ακρίβεια του μοντέλου υπολογίζεται συγκρίνοντας τις προβλέψεις (y_pred) με τις πραγματικές τιμές της κλάσης (y_test). Αν η ακρίβεια του συγκεκριμένου συνδυασμού bins και στρατηγικής είναι υψηλότερη από κάθε προηγούμενη που έχει υπολογιστεί (best_accuracy), τότε αυτή η ακρίβεια, μαζί με τη στρατηγική και τον αριθμό των bins, αποθηκεύονται ως οι καλύτερες τιμές και χρησιμοποιούνται για τη διακριτοποίηση των επιλεγμένων στηλών του συνόλου δεδομένων (σχήμα 5.15). Τα scripts auto_strategy.py και auto_bins.py ακολουθούν παρόμοια λογική, με την κύρια διαφορά τους να εντοπίζεται στη δομή των for loops.

```
X_train, X_test, y_train, y_test = train_test_split(combined_data, y, test_size=0.33, random_state=125)

for bins in range(2, 21):
    for strategy in ['uniform', 'quantile', 'kmeans']:
        kbins = KBinsDiscretizer(n_bins=bins, encode='ordinal', strategy=strategy)

        X_train_binned = kbins.fit_transform(X_train)
        X_test_binned = kbins.transform(X_test)

        nb_classifier = CategoricalNB()
        nb_classifier.fit(X_train_binned, y_train)

        y_pred = nb_classifier.predict(X_test_binned)
        accuracy = accuracy_score(y_test, y_pred)

        if accuracy > best_accuracy:
            best_accuracy = accuracy
            best_strategy = strategy
            best_bin_number = bins
```

Σχήμα 5.14: Κώδικας python auto_all.py

```
kbins_best = KBinsDiscretizer(n_bins=best_bin_number, encode='ordinal', strategy=best_strategy)
kbins_best.fit(data[selected_columns])

data[non_numeric_columns] = data_copy

data_binned = kbins_best.transform(data[selected_columns])

for i, col in enumerate(selected_columns):
    data[col] = data_binned[:, i].astype(int).astype(str)

best_accuracy = round(best_accuracy, 4)

print(json.dumps({"best_accuracy": best_accuracy, "best_strategy": best_strategy, \
"best_bin_number": best_bin_number, "script": "auto_all"}))
```

Σχήμα 5.15: Κώδικας python auto_all.py end

5.4 Υλοποίηση του Front-end

Οι front-end τεχνολογίες είναι απαραίτητες για την ανάπτυξη σύγχρονων ιστοσελίδων. Για την υλοποίηση της εφαρμογής AutoDiscretizer, επιλέξαμε τεχνολογίες front-end όπως HTML, CSS, Bootstrap, JavaScript και jQuery. Η HTML αποτελεί τον πυρήνα για τη δημιουργία των βασικών στοιχείων της ιστοσελίδας. Χρησιμοποιήσαμε την HTML για να σχεδιάσουμε την navigation bar, η οποία περιλαμβάνει τις ακόλουθες επιλογές: 'Αρχική Σελίδα', όπου παρέχονται σημαντικές πληροφορίες για την εφαρμογή, 'KBins', όπου αποτελεί την πρόσβαση στην κύρια λειτουργία της εφαρμογής AutoDiscretizer, 'Σχετικά', όπου ενημερώνουμε τους χρήστες σχετικά με την ανάπτυξη της ιστοσελίδας, και τέλος την επιλογή 'Αξιολόγηση', που προσκαλεί τους χρήστες να αφήσουν τη γνώμη τους για την λειτουργία της εφαρμογής.

Επιπλέον, με τη χρήση της HTML, αναπτύξαμε τα διαδραστικά στοιχεία της ιστοσελίδας μας, όπως τα κουμπιά πλοήγησης και τα αναδυόμενα μενού (dropdown menus) για την επιλογή της μεθόδου διακριτοποίησης και της κλάσης. Προσθέσαμε επίσης πλαίσια επιλογής (checkboxes)

για τον καθορισμό των στηλών που θέλει ο χρήστης να διακριτοποιήσει, καθώς και ένα πεδίο εισαγωγής αριθμών (number input), όπου ο χρήστης μπορεί να εισάγει τον αριθμό για τις διακριτικές κατηγορίες (bins) που επιθυμεί. Ακόμη, η χρήση των Bootstrap κλάσεων, μας βοήθησε να πετύχουμε responsive design και να εφαρμόσουμε μερικά προκαθορισμένα στυλ στην διαδικτυακή εφαρμογή μας. Με τη Bootstrap, δημιουργήσαμε επίσης τους πίνακες που παρουσιάζουν το αρχικό dataset και το dataset μετά τη διακριτοποίηση. Τέλος, είναι σημαντικό να τονίσουμε ότι κάναμε εκτενή χρήση του CSS για να δημιουργήσουμε το δικό μας μοναδικό σχεδιασμό, προσφέροντας πλήρη ελευθερία στην προσαρμογή της εμφάνισης πέρα από τα προκαθορισμένα στυλ που προσφέρει η Bootstrap. Συνοψίζοντας, η συνδυασμένη χρήση της HTML και CSS συνέβαλε στη δημιουργία μιας λειτουργικής και αισθητικά προσελκυστικής ιστοσελίδας, δίνοντας προσοχή τόσο στη δομή όσο και στο στυλ.

Εκτός από τις βασικές τεχνολογίες HTML, CSS και Bootstrap, η JavaScript και η βιβλιοθήκη jQuery υπήρξαν καθοριστικές στη διαδικασία ανάπτυξης της εφαρμογής AutoDiscretizer, προσδίδοντας σημαντική διαδραστικότητα και δυναμικότητα στην ιστοσελίδα. Η jQuery, ειδικότερα, αποδείχθηκε κρίσιμη στην παροχή μιας πιο απλοποιημένης και αποδοτικής προσέγγισης στη γραφή JavaScript. Αυτή η βιβλιοθήκη επέτρεψε την ευκολότερη διαχείριση και τροποποίηση του Document Object Model (DOM), κάτι που κατέστησε την διαδικασία ανάπτυξης πιο ομαλή και ευέλικτη. Μέσω της jQuery, ήταν δυνατή η εφαρμογή πολύπλοκων λειτουργιών JavaScript με λιγότερες γραμμές κώδικα και μεγαλύτερη αποδοτικότητα. Η jQuery συνέβαλε επίσης στην αυξημένη διαλειτουργικότητα μεταξύ διαφορετικών web browsers, μειώνοντας έτσι τις προκλήσεις που συνδέονται με τις διαφορές στην υποστήριξη της JavaScript.

Ένας σημαντικός ρόλος της JavaScript στην ανάπτυξη της ιστοσελίδας είναι η επικοινωνία με τον διακομιστή. Χρησιμοποιώντας την τεχνολογία AJAX (Asynchronous JavaScript and XML), η jQuery μας επέτρεψε να πραγματοποιούμε ασύγχρονες κλήσεις στο API, φορτώνοντας δεδομένα από τον διακομιστή και ενσωματώνοντας αυτά τα δεδομένα στην ιστοσελίδα μας. Το πιο σημαντικό πλεονέκτημα αυτής της προσέγγισης είναι ότι μπορούμε να ενημερώσουμε και να προσθέσουμε περιεχόμενο στην ιστοσελίδα χωρίς την ανάγκη για πλήρη ανανέωση της σελίδας, προσφέροντας έτσι μια πιο ομαλή και διαδραστική εμπειρία στον χρήστη. Παρακάτω, θα δούμε τον κώδικα που χρησιμοποιεί την τεχνολογία AJAX για να πραγματοποιήσει μια αίτηση GET προς το API endpoint /api/read_dataset.php (σχήμα 5.16).

```
160         $.ajax({
161             type: "GET",
162             url: "./api/read_dataset.php?dataset=" + dataset_name + "&binned=" + flag,
163             dataType: "json",
164             success: function (response) {
```

Σχήμα 5.16: Κλήση API endpoint με τεχνολογία AJAX

Ο κώδικας που παρουσιάζεται στο σχήμα 5.15 εκτελεί μια κλήση προς το API. Αυτή η κλήση API έχει ως στόχο την ανάκτηση δεδομένων από ένα συγκεκριμένο dataset. Μέσω της αίτησης, τα δεδομένα αυτά ανακτώνται από τον διακομιστή και επιστρέφονται στην ιστοσελίδα για περαιτέρω επεξεργασία ή εμφάνιση. Η συνάρτηση success καλείται εάν το αίτημα AJAX ολοκληρωθεί επιτυχώς. Το response περιέχει τα δεδομένα που επέστρεψε ο διακομιστής. Σε

περίπτωση επιτυχούς ολοκλήρωσης του αιτήματος, η εφαρμογή δημιουργεί δυναμικά έναν πίνακα μέσω της jQuery, ο οποίος περιλαμβάνει τα δεδομένα του dataset που ανακτήθηκαν από τον διακομιστή. Παράλληλα, αναπτύσσονται checkboxes για κάθε στήλη του dataset που περιέχει αριθμητικά δεδομένα. Επιπρόσθετα, το dropdown menu, το οποίο σχετίζεται με την επιλογή της κλάσης για τον Naive Bayes, ενημερώνεται για να περιλαμβάνει τις στήλες με κατηγορικά και ακέραια αριθμητικά δεδομένα (σχήμα 5.17).

```
164            success: function (response) {
165
166                if (flag !== undefined) {
167                    displayTable(response.dataset, flag);
168
169                    if (!flag) {
170                        displayCheckboxes(response.numericColumns);
171                        updateDropdown(response.categoricalIntegerColumns);
172                    }
173                } else {
174                    openModal('Error Dataset', 'Please choose binned or unbinned!');
175                }
176
177                $('.spinner-cst1').hide();
178            },
```

Σχήμα 5.17: Κώδικας API endpoint success function

Σε περίπτωση που υπάρξει κάποιο σφάλμα κατά την εκτέλεση του αιτήματος, η συνάρτηση success δεν καλείται. Αντ' αυτού, ενεργοποιείται η συνάρτηση error. Αυτή η συνάρτηση είναι υπεύθυνη για την καταγραφή ενός κατάλληλου μηνύματος σφάλματος, παρέχοντας λεπτομέρειες για τη φύση του προβλήματος που προέκυψε κατά την προσπάθεια επικοινωνίας με τον διακομιστή (σχήμα 5.18).

```
177            error: function (error) {
178                $('.spinner-cst1').hide();
179                console.error("Error getting dataset content:", error);
180                openModal('Error Dataset', 'Unable to load the dataset');
181            }
182        });
```

Σχήμα 5.18: Κώδικας API endpoint error function

Για την παρουσίαση των μηνυμάτων σφάλματος, χρησιμοποιούμε την openModal function (σχήμα 5.19) για να ενεργοποιήσουμε τα modals, τα οποία είναι επίσης γνωστά ως modal windows ή modal dialogs. Αποτελούν παράθυρα διαλόγου που εμφανίζονται πάνω από την κύρια οθόνη της εφαρμογής, παρέχοντας σημαντικές ειδοποιήσεις. Τα modals απαιτούν από τον χρήστη να αλληλεπιδράσει με αυτά επιβεβαιώνοντας το μήνυμα σφάλματος, προτού μπορέσει να συνεχίσει με τη χρήση της εφαρμογής.

```
80
81      function openModal(title, message) {
82          $('#modal-title').html(title);
83          $('#modal-message').html(message);
84          $('#modal-toggle').prop('checked', true);
85          $('.spinner-cst1').hide();
86          $('.spinner-cst2').hide();
87      }
```

Σχήμα 5.19: Κώδικας συνάρτησης modal

5.5 Github repository

Το GitHub είναι μια πλατφόρμα ανάπτυξης λογισμικού που χρησιμοποιείται από εκατομμύρια προγραμματιστές και εταιρείες παγκοσμίως για να φιλοξενούν και να διαχειρίζονται τον κώδικά τους, να συνεργάζονται σε έργα και να δημιουργούν καινοτόμο λογισμικό. Είναι η μεγαλύτερη και πιο δημοφιλής διαδικτυακή υπηρεσία φιλοξενίας αποθετηρίων κώδικα στον κόσμο και χρησιμοποιεί το σύστημα ελέγχου εκδόσεων Git.

Το GitHub έχει επιδράσει σημαντικά στην ανάπτυξη λογισμικού. Έχει γίνει ένα ουσιαστικό εργαλείο για τους προγραμματιστές, επιτρέποντάς τους να συνεργάζονται εύκολα σε έργα ανεξάρτητα από τη γεωγραφική τους θέση. Η πλατφόρμα ενισχύει την ανοιχτή πηγή και την κοινότητα ανάπτυξης λογισμικού, προσφέροντας έναν χώρο όπου μπορούν να διαμοιράζονται, να επεκτείνονται και να βελτιώνονται έργα λογισμικού. Επιπλέον, πολλές εταιρείες και οργανισμοί χρησιμοποιούν το GitHub για τη διαχείριση των εσωτερικών τους έργων λογισμικού.

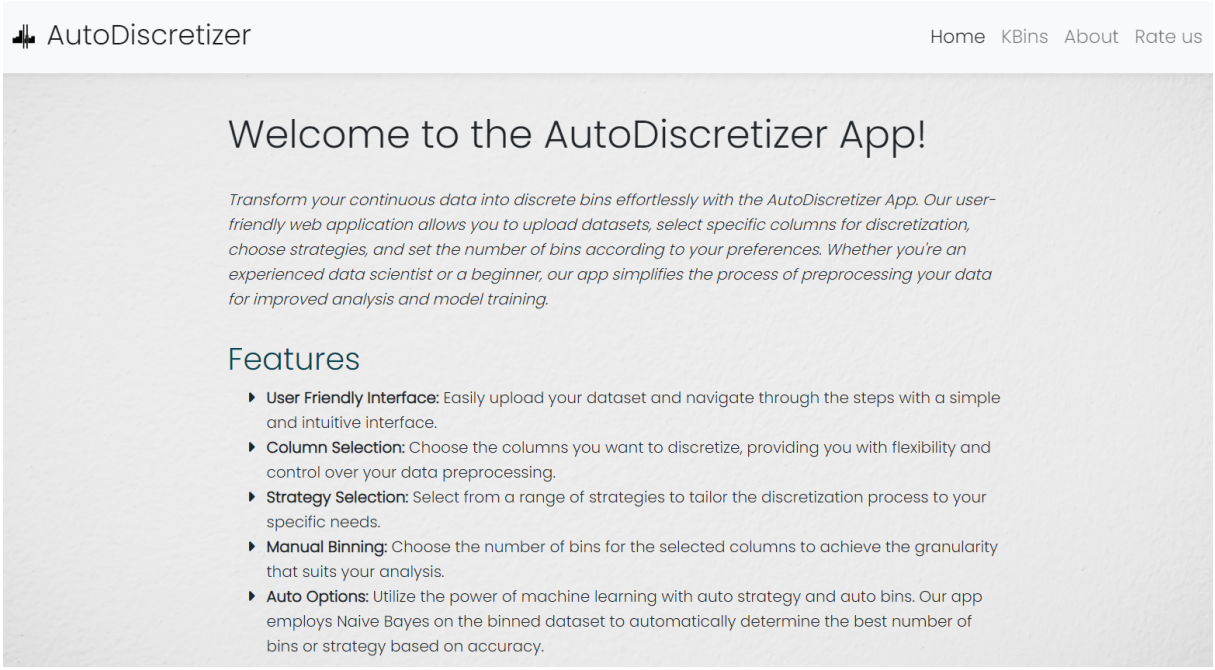
Το Github Repository είναι μία βασική δομική μονάδα του GitHub. Αποτελεί έναν διαδικτυακό χώρο όπου αποθηκεύεται και διαχειρίζεται ο πηγαίος κώδικας για ένα έργο λογισμικού. Τα Github repositories είναι σημαντικά για την οργάνωση, τον έλεγχο εκδόσεων, και τη συνεργασία σε έργα λογισμικού. Ολόκληρος ο πηγαίος κώδικας της εφαρμογής AutoDiscretizer, τόσο για το back-end όσο και για το front-end, είναι διαθέσιμος προς προβολή και λήψη στο ακόλουθο GitHub repository: `https://github.com/dpavlidis/AutoDiscretizer`

Κεφάλαιο 6

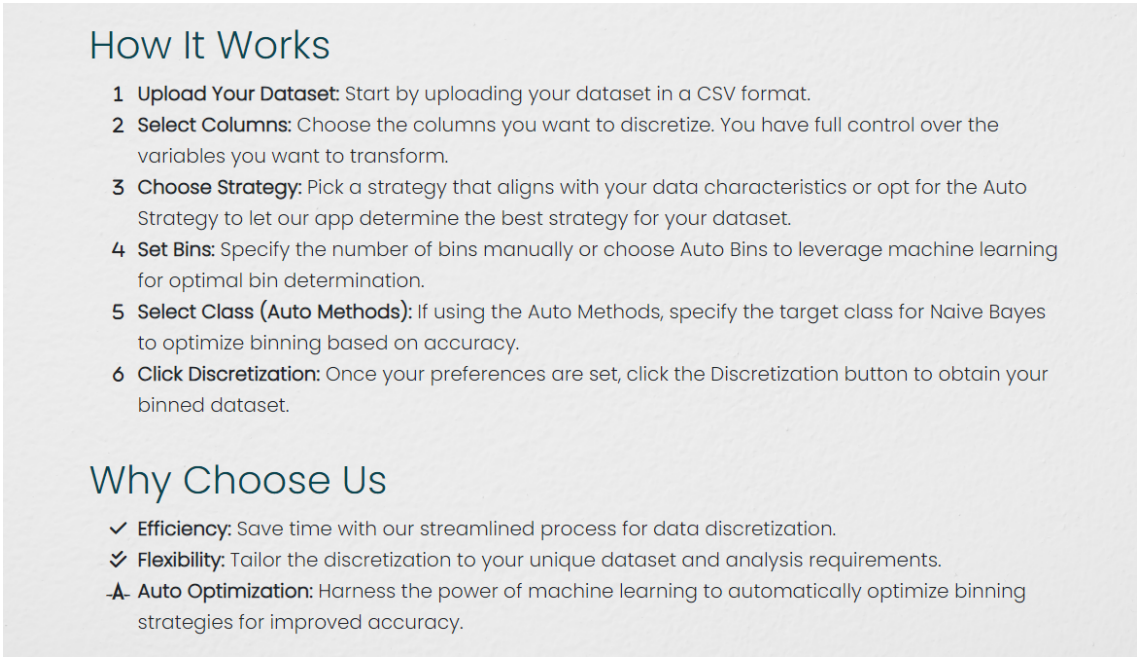
Παρουσίαση του AutoDiscretizer

6.1 Αρχική σελίδα

Η αρχική σελίδα της εφαρμογής AutoDiscretizer είναι σχεδιασμένη για να παρέχει στους χρήστες σημαντικές πληροφορίες σχετικά με τη λειτουργία της. Αρχικά, παρουσιάζεται μια σύνοψη της εφαρμογής, η οποία δίνει μια γενική εικόνα των δυνατοτήτων και του σκοπού της. Ακολουθούν λεπτομερείς πληροφορίες για τα χαρακτηριστικά της εφαρμογής, επισημαίνοντας τις μοναδικές της λειτουργίες (σχήμα 6.1). Στη συνέχεια, παρέχονται οδηγίες χρήσης, βοηθώντας τους χρήστες να κατανοήσουν πώς να αξιοποιήσουν αποτελεσματικά την εφαρμογή. Τέλος, τονίζονται οι λόγοι για τους οποίους η AutoDiscretizer αποτελεί μια ιδανική επιλογή για τους πιθανούς χρήστες, επισημαίνοντας τα μοναδικά της πλεονεκτήματα (σχήμα 6.2).



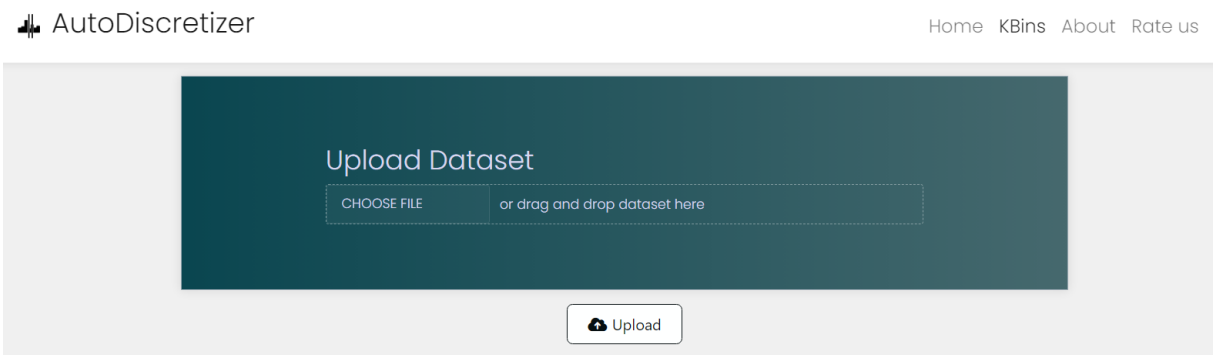
Σχήμα 6.1: Αρχική σελίδα πάνω



Σχήμα 6.2: Αρχική σελίδα κάτω

6.2 Ανέβασμα αρχείου συνόλου δεδομένων

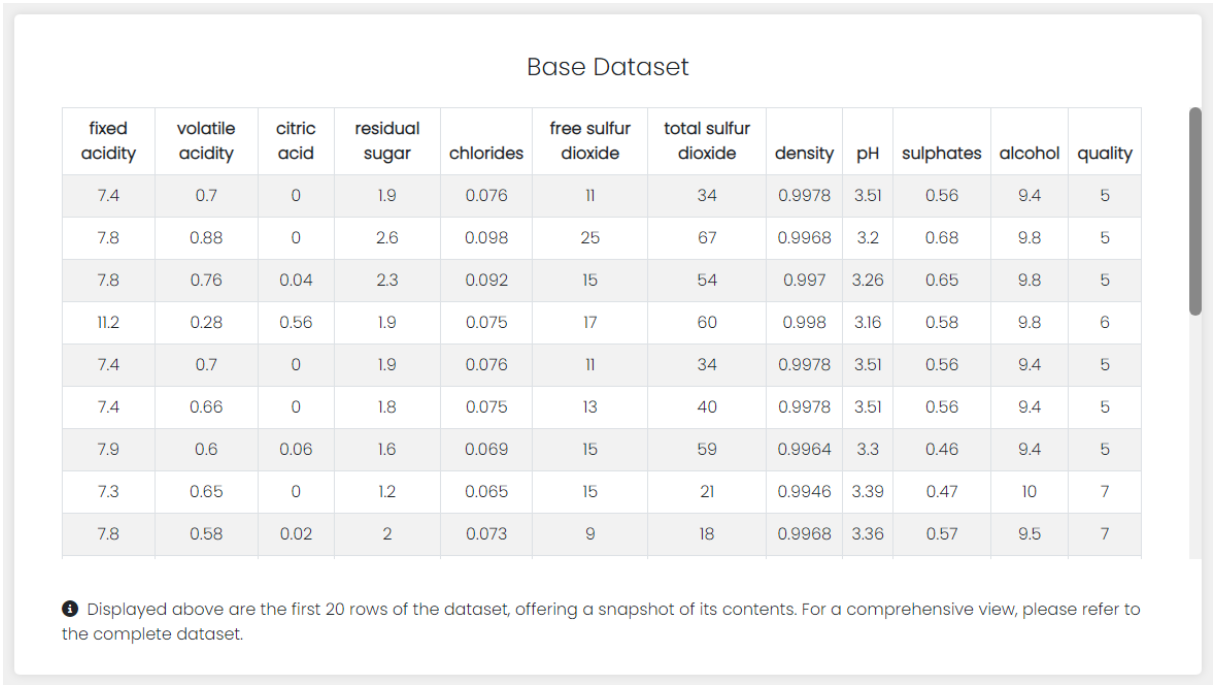
Όπως παρατηρήσαμε στην αρχική σελίδα, υπάρχει επίσης μια γραμμή πλοήγησης (navigation bar), η οποία περιλαμβάνει διάφορες επιλογές. Ανάμεσα σε αυτές, είναι η ενότητα KBins, η οποία οδηγεί τον χρήστη απευθείας στην εφαρμογή. Αρχικά, η εφαρμογή περιέχει ένα πλαίσιο διεπαφής, όπου ο χρηστης έχει τη δυνατότητα να επιλέξει και να φορτώσοει το dataset που επιθυμεί (σχήμα 6.3). Για την επιτυχή εφαρμογή της διακριτοποίησης, το dataset που φορτώνεται στην εφαρμογή πρέπει να περιέχει τουλάχιστον μία στήλη με αριθμητικά δεδομένα. Σε περίπτωση που το dataset δεν πληροί αυτή την προϋπόθεση, η εφαρμογή αυτόματα εμφανίζει κατάλληλο μήνυμα σφάλματος, καθοδηγώντας τον χρήστη για την απαιτούμενη διόρθωση.



Σχήμα 6.3: Ανέβασμα αρχείου συνόλου δεδομένων

Ο χρήστης μπορεί να ανεβάσει ένα dataset είτε επιλέγοντας την επιλογή CHOOSE FILE για να εντοπίσει και να επιλέξει το αρχείο από τον υπολογιστή του, είτε σύροντας το αρχείο (drag and drop) απευθείας στο πλαίσιο διεπαφής. Μόλις γίνει η επιλογή ή η μεταφορά του αρχείου, το κείμενο drag and drop στο πλαίσιο διεπαφής θα αντικατασταθεί αυτόματα από το όνομα του αρχείου που έχει επιλέξει ο χρήστης.

Αφού ο χρήστης επιλέξει το επιθυμητό αρχείο και πατήσει το κουμπί Upload, το σύστημα θα δημιουργήσει και θα εμφανίσει ένα στιγμιότυπο με τα δεδομένα που περιέχονται στο αρχείο σε μορφή πίνακα (σχήμα 6.4). Αυτό το στιγμιότυπο παρέχει μια άμεση οπτική απεικόνιση των δεδομένων, επιτρέποντας στον χρήστη να επαληθεύσει την επιτυχή φόρτωση και να προβεί σε προκαταρκτική εξέταση των δεδομένων πριν προχωρήσει σε διακριτοποίηση.


Base Dataset

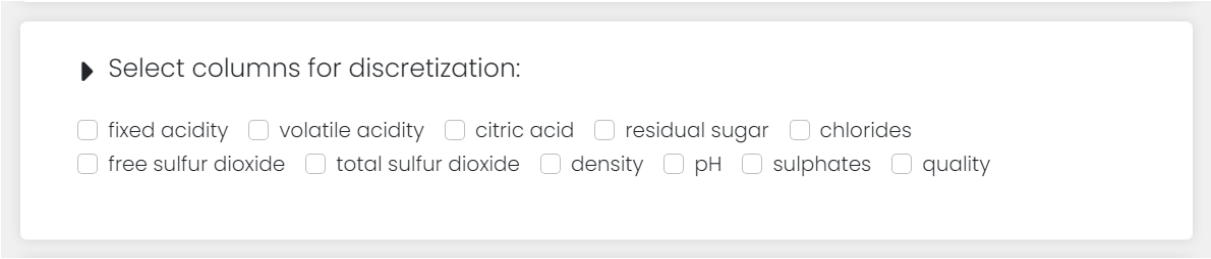
fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5
7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	7
7.8	0.58	0.02	2	0.073	9	18	0.9968	3.36	0.57	9.5	7

Displayed above are the first 20 rows of the dataset, offering a snapshot of its contents. For a comprehensive view, please refer to the complete dataset.

Σχήμα 6.4: Προβολή του αρχείου δεδομένων που έχει ανέβει

6.3 Επιλογή χαρακτηριστικών προς διακριτοποίηση

Στο επόμενο βήμα, παρέχεται ένα πλαίσιο επιλογών όπου ο χρήστης μπορεί να επιλέξει τις στήλες που θέλει να εφαρμόσει διακριτοποίηση (σχήμα 6.5). Σε αυτό το πλαίσιο εμφανίζονται μόνο τα ονόματα των στηλών που περιέχουν αριθμητικά δεδομένα. Σε περίπτωση που κάποιες στήλες του συνόλου δεδομένων περιέχουν κενές τιμές ή σύμβολα, η διακριτοποίηση δεν μπορεί να εφαρμοστεί σε αυτές τις στήλες. Ως αποτέλεσμα, οι συγκεκριμένες στήλες δεν θα εμφανίζονται ως διαθέσιμες επιλογές στο πλαίσιο επιλογής στηλών. Η εφαρμογή διενεργεί όλους τους παραπάνω ελέγχους για να διασφαλίσει ότι η διαδικασία της διακριτοποίησης θα εφαρμοστεί σωστά.

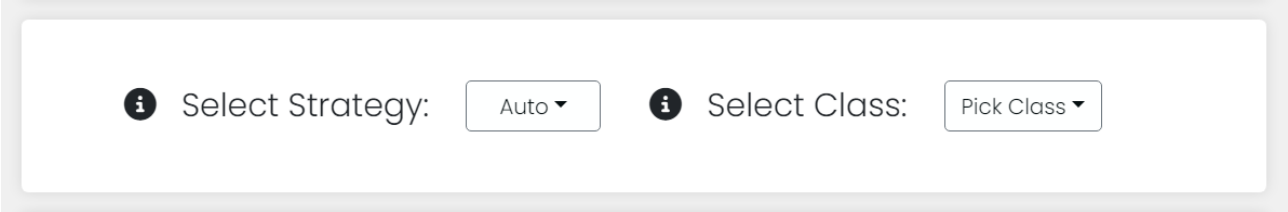


Σχήμα 6.5: Στήλες του συνόλου δεδομένων για διακριτοποίηση

6.4 Επιλογή στρατηγικής

Στη συνέχεια, ο χρήστης καλείται να επιλέξει την επιθυμητή στρατηγική διακριτοποίησης. Οι διαθέσιμες επιλογές περιλαμβάνουν τις μεθόδους Uniform, Quantile, Kmeans και Auto. Η Uniform διαιρεί το εύρος των δεδομένων σε διαστήματα ίσου μεγέθους. Τα δεδομένα κατανέμονται σε αυτά τα διαστήματα με τρόπο που κάθε διάστημα έχει το ίδιο πλάτος, ανεξάρτητα από το πόσες παρατηρήσεις περιέχει. Στην Quantile τα δεδομένα χωρίζονται σε διαστήματα με τρόπο ώστε κάθε διάστημα να περιέχει περίπου τον ίδιο αριθμό παρατηρήσεων και η Kmeans χρησιμοποιεί τον αλγόριθμο Kmeans για να διακριτοποιήσει τα δεδομένα. Οι τιμές των δεδομένων ομαδοποιούνται σε k συστάδες με βάση την ομοιότητά τους. Κάθε συστάδα αντιπροσωπεύει ένα διάστημα στη διακριτοποίηση, με την κάθε τιμή να αντιστοιχίζεται στη συστάδα που είναι πιο κοντά σε αυτήν.

Τέλος, η επιλογή Auto αυτοματοποιεί τη διαδικασία, εφαρμόζοντας τον αλγόριθμο Naive Bayes στα δεδομένα για να αξιολογήσει και να επιλέξει την πιο αποτελεσματική μέθοδο διακριτοποίησης. Η επιλογή γίνεται βάσει της μεγαλύτερης ακρίβειας που επιτυγχάνει κάθε μέθοδος. Σε περίπτωση που ο χρήστης επιλέξει την επιλογή Auto, τότε εμφανίζεται ένα επιπλέον dropdown μενού, στο οποίο ο χρήστης οδηγείται να επιλέξει τη στήλη που θα λειτουργήσει ως κλάση για την εφαρμογή του κατηγοριοποιητή Naive Bayes (σχήμα 6.6).

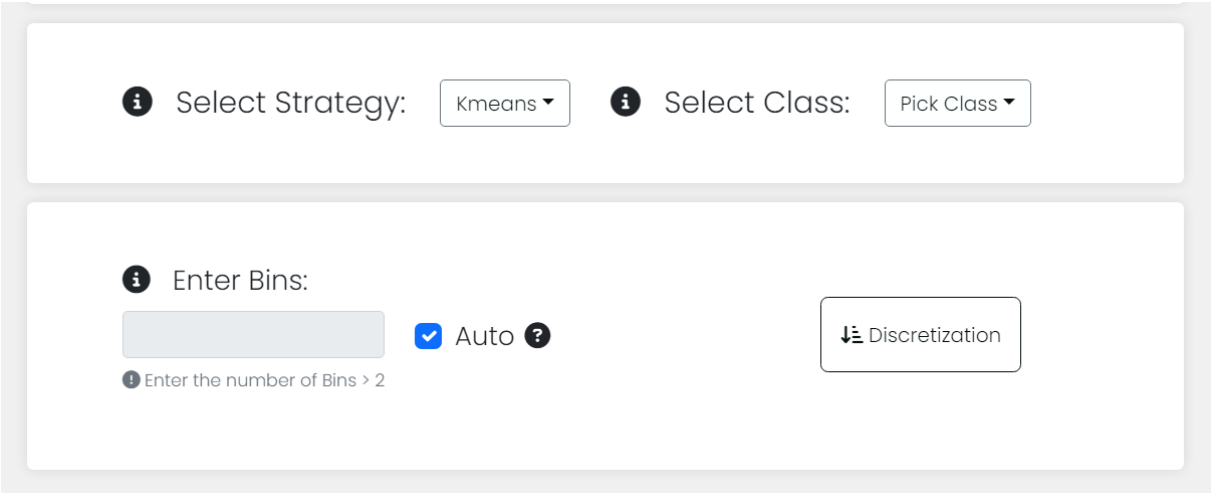


Σχήμα 6.6: Πλαίσιο επιλογής στρατηγικής και κλάσης

6.5 Επιλογή πλήθους bins

Αφού ο χρήστης επιλέξει τις στήλες για διακριτοποίηση και την αντίστοιχη μέθοδο, τότε πρέπει να καθορίσει τον αριθμό των διακριτών κατηγοριών, δηλαδή τον αριθμό των διαστημάτων στα οποία θα χωριστούν τα επιλεγμένα δεδομένα. Οι τιμές που μπορεί να εισάγει ο χρήστης

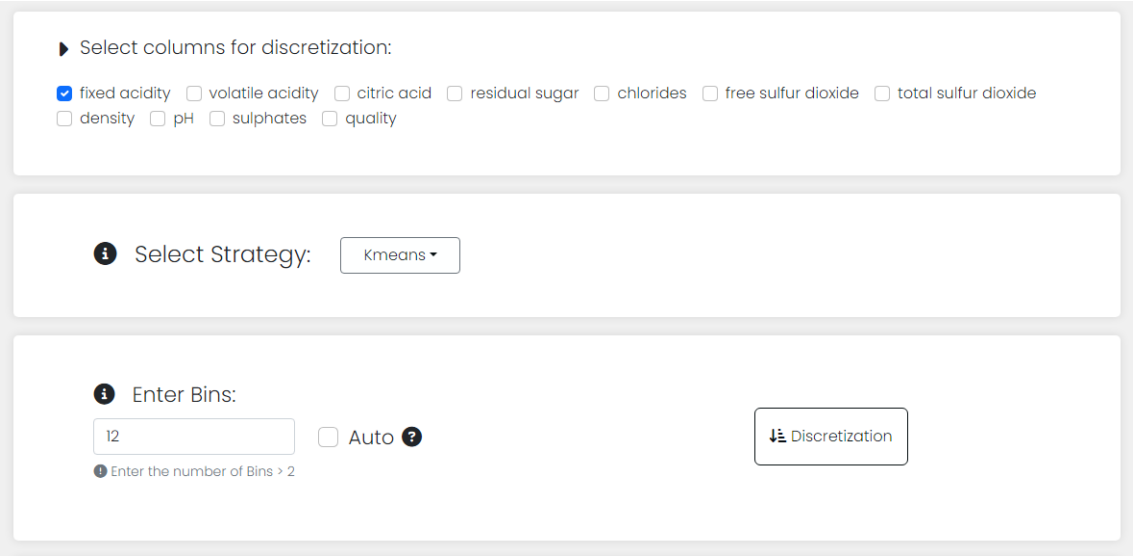
για τον αριθμό των κατηγοριών πρέπει να είναι δύο ή περισσότερες. Εάν το επιθυμεί, μπορεί να επιλέξει την επιλογή Auto, η οποία ενεργοποιεί την αυτόματη εύρεση του βέλτιστου αριθμού κατηγοριών μέσω της κατηγοριοποίησης Naive Bayes (σχήμα 6.7). Αυτή η διαδικασία επιλέγει τον αριθμό των bins που προσφέρει την υψηλότερη ακρίβεια. Όπως και στην επιλογή της στρατηγικής, έτσι και εδώ, αν ο χρήστης επιλέξει την αυτόματη επιλογή, θα εμφανιστεί ένα dropdown μενού για να την επιλογή της κλάσης που θα χρησιμοποιηθεί για τον κατηγοριοποιητή Naive Bayes.



Σχήμα 6.7: Επιλογή Auto πλήθους bins

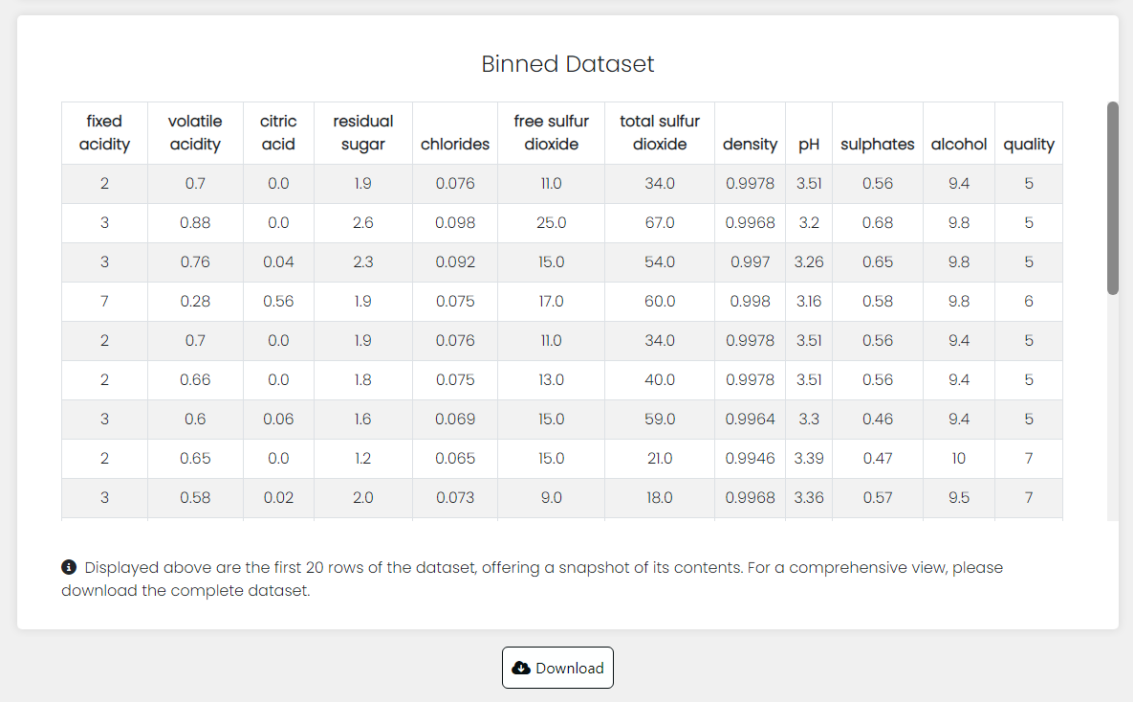
6.6 Ανάκτηση του συνόλου δεδομένων μετά τη διακριτοποίηση

Ο χρήστης, επιλέγοντας τη μέθοδο διακριτοποίησης και ορίζοντας το πλήθος των κατηγοριών (bins), μπορεί να προχωρήσει στην εφαρμογή της διακριτοποίησης πατώντας το κουμπί Discretization. Με την επιλογή και πάτημα του κουμπιού, εμφανίζεται ένας πίνακας που απεικονίζει τα δεδομένα σε διακριτοποιημένη μορφή, βάσει των επιλεγμένων παραμέτρων. Για παράδειγμα, όπως φαίνεται στο σχήμα 6.8, διαλέξαμε για διακριτοποίηση τη στήλη fixed acidity από το dataset των κρασιών. Ως μέθοδο διακριτοποίησης επιλέξαμε τον αλγόριθμο Kmeans και ορίσαμε τον αριθμό των κατηγοριών (bins) σε 12. Αυτές οι επιλογές θα μας επιτρέψουν να ομαδοποιήσουμε τις τιμές της στήλης fixed acidity σε 12 διαφορετικές κατηγορίες, βάσει των ομοιοτήτων τους.



Σχήμα 6.8: Παράδειγμα επιλογής παραμέτρων

Μετά το πάτημα του κουμπιού Discretization, παράγεται ένας πίνακας όπου η στήλη fixed acidity εμφανίζεται πλέον με τις τιμές της διακριτοποιημένες (σχήμα 6.9). Από την ανάλυση του αρχικού dataset, διαπιστώνουμε ότι η πρώτη τιμή της στήλης fixed acidity, η οποία είναι 7.4, έχει τοποθετηθεί στην κατηγορία 2. Επίσης, οι δεύτερη και τρίτη τιμή, που είναι 7.8, ανήκουν στην κατηγορία 3, ενώ η τέταρτη τιμή, που ανέρχεται στο 11.2, βρίσκεται στην κατηγορία 7.

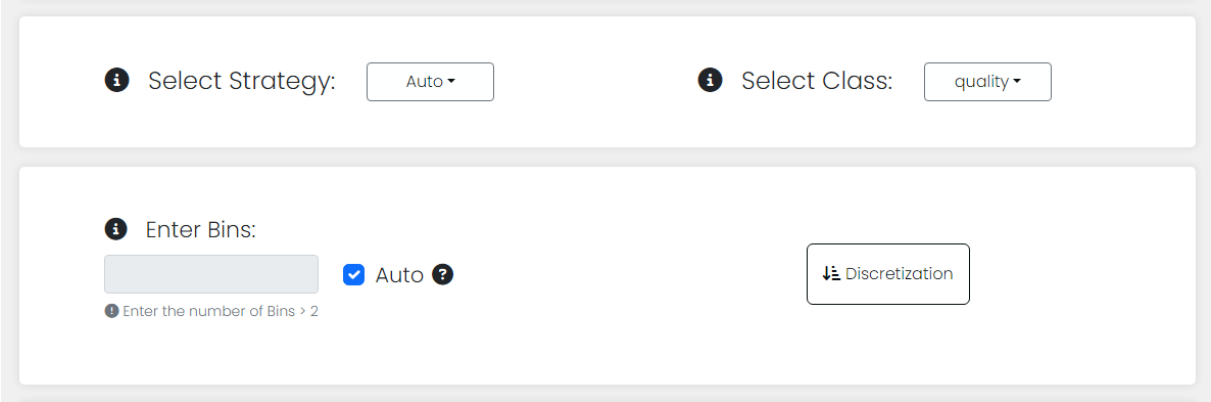


Σχήμα 6.9: Προβολή των διακριτοποιημένων δεδομένων

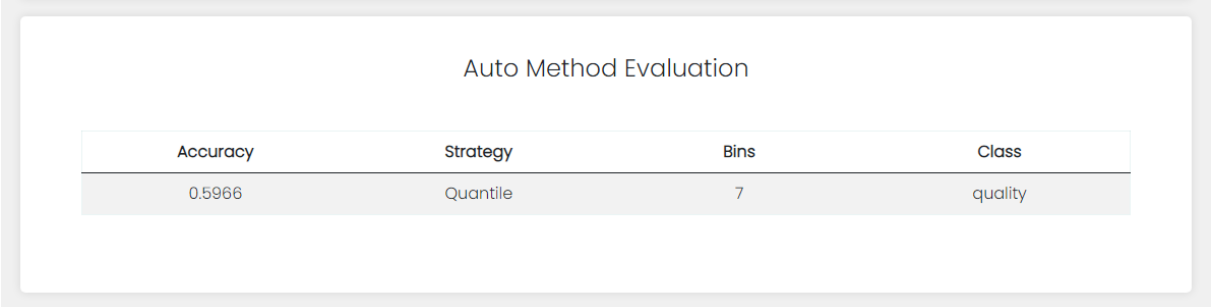
Επιπλέον, πατώντας το κουμπί Download, ο χρήστης έχει τη δυνατότητα να κατεβάσει το πλήρες σετ των διακριτοποιημένων δεδομένων σε μορφή csv για να το χρησιμοποιήσει στις αναλυτικές του ανάγκες.

6.7 Αξιολόγηση αυτόματης επιλογής

Στην περίπτωση που ο χρήστης επιλέξει τις αυτόματες επιλογές, θα κληθεί να ορίσει επίσης και την κλάση (όπως φαίνεται στο σχήμα 6.10). Εκτός από την εμφάνιση του πίνακα με τα διακριτοποιημένα δεδομένα, θα παρουσιαστεί και ένας επιπρόσθετος πίνακας που παρέχει πληροφορίες σχετικά με τις αυτόματες επιλογές. Αυτός ο πίνακας θα περιλαμβάνει την επιλεγμένη μέθοδο διακριτοποίησης και τον αριθμό των κατηγοριών (bins) που ορίστηκαν αυτόματα από την εφαρμογή, καθώς και τη μέγιστη ακρίβεια (accuracy) που επιτεύχθηκε με τη χρήση του αλγορίθμου Naive Bayes για τον συγκεκριμένο συνδυασμό μεθόδου και κατηγοριών. Όπως φαίνεται στο σχήμα 6.11, η μέθοδος που επιλέχθηκε είναι η Quantile και ο αριθμός των bins είναι 7. Επιπλέον, αυτός ο συνδυασμός πέτυχε ακρίβεια 59% στο συγκεκριμένο dataset.



Σχήμα 6.10: Παράδειγμα επιλογής αυτόματων παραμέτρων



Σχήμα 6.11: Πίνακας Αξιολόγησης

Μπορείτε να χρησιμοποιήσετε την διαδικτυακή εφαρμογή **AutoDiscretizer** στον ακόλουθο σύνδεσμο: `https://kclusterhub.iee.ihu.gr/autodiscretizer`

Κεφάλαιο 7

Συμπεράσματα και Μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Η διακριτοποίηση αποτελεί μια κρίσιμη τεχνική στην επεξεργασία και την ανάλυση δεδομένων. Η βασική της ιδέα είναι η μετατροπή συνεχών δεδομένων, όπως οι αριθμητικές τιμές, σε μια περιορισμένη ομάδα κατηγοριών ή διακριτών διαστημάτων. Αυτή η μέθοδος είναι ιδιαίτερα χρήσιμη σε περιπτώσεις όπου οι αλγόριθμοι μηχανικής μάθησης ή άλλες μέθοδοι ανάλυσης δεδομένων λειτουργούν καλύτερα με διακριτές εισόδους αντί για συνεχείς. Η διακριτοποίηση μπορεί να βοηθήσει επίσης στην απλοποίηση και την κατανόηση των δεδομένων, καθώς οι κατηγοριοποιημένες τιμές είναι συχνά πιο εύκολο να ερμηνευτούν από τα ανθρώπινα μυαλά. Στη διαδικασία διακριτοποίησης, τα δεδομένα χωρίζονται συνήθως σε “δοχεία“ ή “καλάθια“, κάθε ένα από τα οποία περιέχει ένα εύρος τιμών. Αυτό μπορεί να γίνει με βάση διάφορα κριτήρια, όπως η ομοιομορφία του εύρους, η συχνότητα εμφάνισης τιμών, ή με συσταδοποίηση. Η διακριτοποίηση είναι επίσης σημαντική στην προεπεξεργασία δεδομένων για την ανάλυση και την εξόρυξη δεδομένων. Μέσω αυτής της μεθόδου, μπορεί να ενισχυθεί η απόδοση των προβλεπτικών μοντέλων και να απλοποιηθούν τα δεδομένα για καλύτερη ερμηνεία και ευκολότερη οπτικοποίηση.

Μετά απο έρευνα, διαπιστώσαμε ένα κενό στην αγορά όσον αφορά την διαθεσιμότητα εφαρμογών που προσφέρουν υπηρεσίες διακριτοποίησης δεδομένων με εύχρηστη διεπαφή χρήστη. Αυτή η ανάγκη, σε συνδυασμό με τις απαιτήσεις που αναφέραμε παραπάνω, οδήγησε στη δημιουργία της διαδικτυακής εφαρμογής AutoDiscretizer. Αυτή η εφαρμογή επιτρέπει στους χρήστες να ανεβάζουν τα δεδομένα τους μέσω μιας φιλικής προς τον χρήστη διεπαφής, παρέχοντας τους τη δυνατότητα να επιλέξουν ποια χαρακτηριστικά των δεδομένων τους θέλουν να υποβάλλουν σε διακριτοποίηση. Οι χρήστες μπορούν επίσης να επιλέξουν την επιθυμητή μέθοδο διακριτοποίησης και τον αριθμό των διακριτών διαστημάτων (bins) που επιθυμούν. Τα επεξεργασμένα δεδομένα παρέχονται στη συνέχεια σε μορφή csv, επιτρέποντας εύκολη ανάκτηση και χρήση. Επιπρόσθετα, η εφαρμογή προσφέρει τη δυνατότητα αυτόματης επιλογής μεθόδου και bins. Αυτό επιτυγχάνεται μέσω της εφαρμογής του αλγορίθμου Naive Bayes, ο οποίος αναλύει τα δεδομένα και προσδιορίζει την πιο κατάλληλη μέθοδο διακριτοποίησης και

τον αριθμό των bins, βασιζόμενος στην μέγιστη δυνατή ακρίβεια (accuracy) που επιτυγχάνεται μέσω της εφαρμογής του. Με αυτόν τον τρόπο, η AutoDiscretizer απλοποιεί την διαδικασία διακριτοποίησης δεδομένων, καθιστώντας την πιο αποτελεσματική και προσιτή στους χρήστες.

7.2 Μελλοντικές επεκτάσεις

Οπτικοποίηση Δεδομένων:
Η οπτικοποίηση δεδομένων περιλαμβάνει τη δυνατότητα παρουσίασης των δεδομένων μέσω ευέλικτων και διαισθητικών γραφημάτων, όπως ιστογράμματα, scatter plots, pie charts και heat maps. Κάθε μορφή οπτικοποίησης θα είναι ιδανικά προσαρμοσμένη για διαφορετικούς τύπους δεδομένων και αναλυτικές ανάγκες, επιτρέποντας στους χρήστες να εντοπίζουν τάσεις, πρότυπα και ανωμαλίες με μεγαλύτερη ευκολία και ακρίβεια. Επιπλέον, η αναβαθμισμένη λειτουργία παρουσίασης των επεξεργασμένων δεδομένων στην εφαρμογή θα επιτρέπει την πιο λεπτομερή και ξεκάθαρη αναγνώριση των διακριτοποιημένων κατηγοριών που έχουν δημιουργηθεί. Αυτή η βελτίωση θα διευκολύνει τους χρήστες να διαχωρίζουν και να κατανοούν τις διακριτές κατηγορίες με μεγαλύτερη ακρίβεια, αυξάνοντας την κατανόηση και την εποπτεία των δεδομένων. Πέραν αυτού, η εφαρμογή θα προσφέρει επιλογές προσαρμογής των γραφημάτων, όπως η επιλογή χρωμάτων, τύπων γραφημάτων, και μεγέθους, επιτρέποντας στους χρήστες να τα προσαρμόσουν σύμφωνα με τα δεδομένα που διαθέτουν και τις προτιμήσεις τους. Αυτή η δυναμική αλληλεπίδραση με τα γραφήματα θα καταστήσει την εμπειρία ανάλυσης πιο διαισθητική και διαδραστική.

Ενσωμάτωση επιπλέον αλγορίθμων μηχανικής μάθησης:
Πέρα από την εφαρμογή του αλγορίθμου Naive Bayes για την αυτόματη καθοδήγηση στην επιλογή της καταλληλότερης μεθόδου διακριτοποίησης και του αριθμού των bins βάσει της ακρίβειας, η εφαρμογή θα μπορούσε επίσης να προσφέρει την επιλογή χρήσης άλλων αλγορίθμων μηχανικής μάθησης, όπως Decision Trees, K-Nearest Neighbors (KNN) και Support Vector Machines (SVM), προσφέροντας έτσι μια ευρύτερη γκάμα αναλυτικών εργαλείων. Κάθε ένας από αυτούς τους αλγόριθμους μπορεί να προσφέρει διαφορετικές προοπτικές και βελτιστοποιήσεις στην ανάλυση, επιτρέποντας την ακριβέστερη προσαρμογή των μεθόδων διακριτοποίησης και του πλήθους των bins στις ανάγκες των δεδομένων.

Απόδοση και αποθήκευση αποτελεσμάτων:
Όταν το σύνολο δεδομένων που επιθυμεί ο χρήστης να υποστεί διακριτοποίηση είναι αρκετά μεγάλο, δηλαδή περιλαμβάνει 500.000 γραμμές ή περισσότερες, είναι λογικό να υπάρχει καθυστέρηση κατά την επεξεργασία των δεδομένων. Σε αυτήν την περίπτωση, θα υπάρχει ένα πλαίσιο όπου ο χρήστης μπορεί να καταχωρίσει το email του, ώστε να λάβει τα αποτελέσματα όταν ολοκληρωθεί η διαδικασία. Επιπλέον, εκτός από την εξαγωγή των δεδομένων σε μορφή csv, θα παρέχεται η δυνατότητα αποθήκευσής τους σε πρόσθετες μορφές, όπως PDF, Word, ή ακόμα και PowerPoint, μέσω διαθέσιμων επιλογών.

Βιβλιογραφία

[1] Tableau, "Data management: What it is, importance, and challenges." https://www.tableau.com, 2023. Accessed: 2023-12-23.

[2] P. L. Online, "5 key reasons why data analytics is important to business." https://lpsonline.sas.upenn.edu, 2022. Accessed: 2023-12-23.

[3] Y. Yang, *Discretization*, pp. 287–288. Boston, MA: Springer US, 2010.

[4] F. A., S. Qian, and C. Stow, "Comparative analysis of discretization methods in bayesian networks," *Environmental Modelling Software*, vol. 87, pp. 64–71, 01 2017.

[5] A. Rosenfeld, R. Illuz, D. Gottesman, and M. Last, "Using discretization for extending the set of predictive features," *EURASIP Journal on Advances in Signal Processing*, vol. 2018, no. 1, p. 7, 2018.

[6] "Ml | binning or discretization." https://www.geeksforgeeks.org/ml-binning-or-discretization/. Accessed: Dec 23, 2023.

[7] M. Feurer and F. Hutter, *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[8] Q. He and J. Liu, "The evolution of automated machine learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1480–1495, 2019.

[9] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[10] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proceedings of the International Conference on Machine Learning*, pp. 1996–2004, 2015.

[11] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[12] R. Caruana and A. Niculescu-Mizil, "Model selection for accurate and interpretable high-dimensional predictive modeling," *Proceedings of the 22nd International Conference on Machine Learning*, pp. 161–168, 2006.

[13] T. G. Dietterich, "Ensemble learning," *The Handbook of Brain Theory and Neural Networks*, vol. 2, pp. 110–125, 2002.

[14] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "AutoWEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2702–2708, 2017.

[15] R. S. Olson, R. Bart, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 485–492, 2016.

[16] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r," *Journal of Statistical Software*, vol. 27, no. 3, pp. 1–22, 2008.

[17] E. Brown, "Domain-specific automl solutions: Applications and challenges," *Data Science Review*, vol. 12, no. 4, pp. 345–362, 2021.

[18] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," *Advances in neural information processing systems*, vol. 28, pp. 2962–2970, 2015.

[19] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 485–492, 2016.

[20] H2O.ai, "H2o.ai: An open source leader in ai and machine learning," *arXiv preprint arXiv:2020*, 2020.

[21] G. Cloud, "Google automl: Machine learning made easy," *https://cloud.google.com/automl*, 2022.

[22] "Data types." `https://www.mygreatlearning.com`. Accessed: Dec 25, 2023.

[23] "Amount of data created daily." `https://explodingtopics.com/blog/data-generated-per-day`. Accessed: Dec 25, 2023.

[24] "Data discretization in machine learning." `https://www.blog.trainindata.com/data-discretization-in-machine-learning/`. Accessed: Dec 25, 2023.

[25] "Discretization and when to use it." `https://medium.com`. Accessed: Dec 26, 2023.

[26] "Data discretization." `https://medium.com`. Accessed: Dec 27, 2023.

[27] G. Mitra, S. Sundareisan, and B. Sarkar, "A simple data discretizer," 10 2017.

[28] "Demonstrating the different strategies of kbinsdiscretizer." `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html`. Accessed: Dec 28, 2023.

[29] P. Putri, S. Prasetiyowati, and Y. Sibaroni, "The performance of the equal-width and equal-frequency discretization methods on data features in classification process," *sinkron*, vol. 8, pp. 2082–2098, 10 2023.

[30] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: An enabling technique," *Data Min. Knowl. Discov.*, vol. 6, pp. 393–423, 10 2002.

[31] H. Elhilbawi, S. Eldawlatly, and H. Mahdi, "The importance of discretization methods in machine learning applications: A case study of predicting icu mortality," in *Advanced Machine Learning Technologies and Applications* (A.-E. Hassanien, K.-C. Chang, and T. Mincong, eds.), (Cham), pp. 214–224, Springer International Publishing, 2021.

[32] "Machine learning mastery." `https://machinelearningmastery.com/`. Accessed: Dec 27, 2023.

[33] M. Parsian, *Data Algorithms*. O'Reilly Media, 2015.

[34] "Naïve bayes classification in python." `https://medium.com/@shuv.sdr/na%C3%AFve-bayes-classification-in-python-f869c2e0dbf1`. Accessed: Dec 30, 2023.

[35] K. Tatroe and P. MacIntyre, *Programming PHP*. O'Reilly Media, Inc., 4th ed., March 2020.

[36] "General python." `https://docs.python.org/3/faq/general.html`. Accessed: Jan 06, 2024.

[37] M. Lutz, *Learning Python*. O'Reilly Media, 5th ed., July 2013.

[38] "Numpy documentation." `https://numpy.org/`. Accessed: Jan 06, 2024.

[39] "Pandas." `https://pandas.pydata.org/`. Accessed: Jan 06, 2024.

[40] "Matplotlib: Visualization with python." `https://matplotlib.org/`. Accessed: Jan 06, 2024.

[41] "Scipy." `https://scipy.org/`. Accessed: Jan 06, 2024.

[42] "Welcome to flask." `https://flask.palletsprojects.com/en/3.0.x/`. Accessed: Jan 06, 2024.

[43] "Meet django." `https://www.djangoproject.com/`. Accessed: Jan 06, 2024.

[44] "Introduction to tensorflow." `https://www.tensorflow.org/learn`. Accessed: Jan 06, 2024.

[45] "Pytorch." `https://pytorch.org/`. Accessed: Jan 06, 2024.

[46] "Scikit-learn machine learning in python." `https://scikit-learn.org/stable/`. Accessed: Jan 06, 2024.

[47] "About xampp." `https://www.apachefriends.org/about.html`. Accessed: Jan 06, 2024.

[48] "What is an api?." `https://www.ibm.com/topics/api`. Accessed: Jan 06, 2024.

[49] "Stateless-ness in restful apis." https://kalemaedgar.medium.com/stateless-ness-in-restful-apis-65db25dc96a1. Accessed: Jan 06, 2024.

[50] "Build and test apis together." https://www.postman.com/. Accessed: Jan 06, 2024.

[51] B. Frain, *Responsive Web Design with HTML5 and CSS*. Packt Publishing, 4th ed., 2022.

[52] "About bootstrap." https://getbootstrap.com/docs/4.0/about/overview/. Accessed: Jan 08, 2024.

[53] D. Flanagan, *JavaScript: The Definitive Guide*. O'Reilly Media, Inc., 7th ed., May 2020.

[54] M. Delamater and Z. Ruvalcaba, *Murach's JavaScript and jQuery*. Murach Books, 3rd ed., 2017.

[55] "Atlassian | user stories with examples." What are agile user stories? Accessed: Jan 10, 2024.