

Ομαδική Έκθεση Εργασίας

Στοιχεία Έργου & Μελών Ομάδας	
Αναγνωριστικό Έργου	40
Τίτλος Έργου	Πρόβλεψη Ανταπόκρισης Καμπάνιας Μάρκετινγκ με χρήση αλγορίθμων μηχανικής μάθησης
Ακαδημαϊκό Έτος	2024-2025
Τμήμα	ΗΛΕ52
Ονοματεπώνυμο Καθηγητή Συμβούλου	Ευθυμίουπουλος Νικόλαος
Ονοματεπώνυμα Μελών Ομάδας	Ασλανίδης Ραφαήλ (Συντονιστής) Πιτσαρής Κωνσταντίνος Κρανίτσα Αντωνία
Ημερομηνία Υποβολής	11/06/2025

Περιεχόμενα

1. Εισαγωγή – Περιγραφή Προβλήματος & Στοχοθεσία.....	2
2. Μεθοδολογία και Υλοποίηση Εφαρμογής.....	4
2.1 Σχεδιασμός και Υλοποίηση Μοντέλου Πρόβλεψης (KNN).....	4
2.2 Σχεδιασμός και Υλοποίηση Γραφικού Περιβάλλοντος Χρήστη (GUI).....	5
3. Παρουσίαση Λειτουργικότητας Εφαρμογής (με στιγμιότυπα εκτέλεσης):.....	9
4. Κατανομή Εργασιών Ομάδας.....	17
4.1 Κατανομή Ρόλων και Εργασιών.....	17
4.2 Εργαλεία Συνεργασίας.....	19
5. Βιβλιογραφία.....	20
6. Παράρτημα.....	22

1. Εισαγωγή – Περιγραφή Προβλήματος & Στοχοθεσία

Αυτό το έργο, με τίτλο «Πρόβλεψη Καμπάνιας Marketing με χρήση αλγορίθμων μηχανικής μάθησης» (Project ID 40) αναπτύχθηκε στο πλαίσιο ενός ομαδικού προγραμματιστικού project για τη Θεματική Ενότητα ΠΛΗΠΡΟ του Ελληνικού Ανοικτού Πανεπιστημίου το ακαδημαϊκό έτος 2024-2025. Σκοπός του έργου είναι η ανάπτυξη λογισμικού Μηχανικής Μάθησης σε γλώσσα Python, το οποίο ενσωματώνει Γραφικό Περιβάλλον Χρήστη (GUI). Το λογισμικό αυτό στοχεύει στην πρόβλεψη της ανταπόκρισης πελατών σε προωθητικές ενέργειες marketing.

Το πρόβλημα αφορά μια εταιρεία marketing που σχεδιάζει μια νέα καμπάνια για μία αλυσίδα καταστημάτων ηλεκτρικών ειδών. Για να στοχεύσει άτομα (πελάτες), η εταιρεία χρησιμοποιεί ιστορικά δεδομένα από μια προηγούμενη διαφημιστική καμπάνια, αποθηκευμένα στο αρχείο 'PastCampaignData.xlsx'. Αυτά τα δεδομένα περιλαμβάνουν δημογραφικές πληροφορίες (όπως ηλικία, φύλο, περιοχή διαμονής), στοιχεία χρήσης μέσων επικοινωνίας (email, κινητό τηλέφωνο), πληροφορίες χρήσης του ηλεκτρονικού καταστήματος (logins, αγορές στο πρόσφατο παρελθόν, σύνολο αγορών) και το κρίσιμο στοιχείο της **ανταπόκρισης**, εάν δηλαδή το άτομο συμμετείχε ή όχι στη τελευταία διαφημιστική καμπάνια, για κάθε πελάτη.

Ο στόχος είναι η πρόβλεψη της ανταπόκρισης για τα άτομα που αποτελούν στόχο της νέας καμπάνιας, τα στοιχεία των οποίων βρίσκονται στο αρχείο 'NewCampaignData.xlsx'. Τα δεδομένα αυτά είναι παρόμοια με τα ιστορικά, αλλά **χωρίς την κρίσιμη πληροφορία της ανταπόκρισης**. Η πρόβλεψη αυτή πραγματοποιείται μέσω του αλγορίθμου k-NN (k-Nearest Neighbors), μίας δημοφιλούς μεθόδου μηχανικής μάθησης. Ο υπολογιστής «μαθαίνει» από την συμπεριφορά των πελατών στα ιστορικά δεδομένα για να προβλέψει τη συμπεριφορά των μελλοντικών πελατών.

Τα ζητούμενα της εργασίας περιλαμβάνουν την ανάπτυξη κώδικα σε Python:

- Την **ανάγνωση των αρχείων** 'PastCampaignData.xlsx' και 'NewCampaignData.xlsx'.
- Την **εκπαίδευση ενός μοντέλου k-NN** χρησιμοποιώντας τα ιστορικά δεδομένα, προσδιορίζοντας το πεδίο της ανταπόκρισης ως την κλάση-στόχο.
- Την **πρόβλεψη της ανταπόκρισης** των νέων πελατών με βάση το εκπαιδευμένο μοντέλο.
- Τον **υπολογισμό μετρικών ακρίβειας** για τις προβλέψεις, όπως η συνολική ακρίβεια (accuracy) και η ακρίβεια ανά κλάση (class-specific precision/accuracy).
- Τη δημιουργία ενός **κατάλληλο περιβάλλοντος διεπαφής (GUI)** για τη διαχείριση των πληροφοριών και λειτουργιών της εφαρμογής από

τον χρήστη. Το GUI επιτρέπει τη φόρτωση δεδομένων, εκκίνηση εκπαίδευσης/πρόβλεψης, εμφάνιση αποτελεσμάτων/μετρικών και αποθήκευση αποτελεσμάτων (προβλέψεων).

Η υλοποίηση της εφαρμογής χρησιμοποιεί τη γλώσσα προγραμματισμού Python και συγκεκριμένες βιβλιοθήκες, όπως η scikit-learn για τον αλγόριθμο k-NN, pandas για τη διαχείριση δεδομένων, tkinter και sv_ttk για το GUI και matplotlib για την οπτικοποίηση των αποτελεσμάτων και των μετρικών. Περιλαμβάνει λειτουργίες όπως την εύρεση του βέλτιστου αριθμού γειτόνων (k) μέσω διασταυρούμενης επικύρωσης (cross-validation) και την εμφάνιση γραφημάτων, όπως διάγραμμα πίτας για την κατανομή των προβλέψεων ανταπόκρισης ανά φύλο.

2. Μεθοδολογία και Υλοποίηση Εφαρμογής

2.1 Σχεδιασμός και Υλοποίηση Μοντέλου Πρόβλεψης (KNN)

1. Συλλογή και Προ επεξεργασία Δεδομένων Εργασίας

- a. Φόρτωση και Επισκόπηση Δεδομένων των αρχείων δεδομένων (PastCampaignData.xlsx, NewCampaignData.xlsx)
- b. Ανάλυση χαρακτηριστικών και προσδιορισμός στοχευόμενη μεταβλητής (η στήλη Ανταπόκρισης στο αρχείο PastCampaignData.xlsx)
- c. Διαχωρισμός των χαρακτηριστικών σε κατηγορικά και αριθμητικά
- d. Μετασχηματισμός χαρακτηριστικών (π.χ. κωδικοποίηση κατηγορικών μεταβλητών σε δυαδικές τιμές και κανονικοποίηση / κλιμάκωση των αριθμητικών.)

2. Ανάπτυξη και Εκπαίδευση Μοντέλου Πρόβλεψης (k-NN)

- a. Θεωρητικό Υπόβαθρο Αλγορίθμου (k-NN):
 - i. Ο αλγόριθμος KNN λειτουργεί βασιζόμενος στην υπόθεση ότι δεδομένα με παρόμοια χαρακτηριστικά είναι γεωμετρικά κοντά σε έναν πολυδιάστατο χώρο. Για να καθορίσει την κατηγορία ενός νέου δείγματος, υπολογίζει την απόσταση του από όλα τα δεδομένα εκπαίδευσης και επιλέγει τους **K** πιο κοντινούς γείτονες. Έπειτα, επιλέγει την κατάλληλη κατηγορία αναλόγως την κατηγορία της πλειοψηφίας των γειτόνων.
 - ii. Στα πλαίσια της εργασίας αναπτύχθηκε ένας αλγόριθμος όπου αναζητά τον βέλτιστο αριθμό **K** με βάση κάποιο metric, (π.χ. **precision**) εκτελώντας το **GridSearchCV** με διαφορετικό αριθμό folds (π.χ. 3, 5, 10) μέσω του **StratifiedFold**, ώστε να παρατηρηθεί ποια τιμή του **K** εμφανίζεται συχνότερα ως βέλτιστη.
 - iii. Μόλις βρεθεί το βέλτιστο **K** προχωράει στην εκπαίδευση του τελικού μοντέλου, αυτήν την φορά σε ολόκληρο το dataset. Αφού τελειώσει η εκπαίδευση, χρησιμοποιείτε το μοντέλο για να προβλεφθεί η ανταπόκριση των πελατών.

3. Υπολογισμός μετρικών

- a. Τέλος, για να γίνει μια πλήρη εκτίμηση της απόδοσης χρησιμοποιούνται τα functions **classification_report** και **confusion_matrix**, ξανά εκπαιδεύεται το μοντέλο, αυτήν την φορά σε ένα κομμάτι του dataset και στην συνέχεια γίνεται μια πρόβλεψη στο υπόλοιπο, έπειτα, αυτά τα functions συγκρίνουν τις προβλέψεις με τις υπάρχουσες τιμές (από το υπόλοιπο

dataset αφού είναι γνωστές) και εκφράζουν τα αποτελέσματα ως **precision**, **accuracy**, **class_specific precision** και **class_specific accuracy**, τέλος τα αποθηκεύονται σε ένα string ώστε να μπορεί εύκολα να γίνει η εμφάνιση τους στο GUI της εφαρμογής.

4. Έννοιες:

a.

- i. **cross-validation**: χωρίζει το dataset σε k-folds και χρησιμοποιεί εκπαιδεύει το μοντέλο σε k-1 τμήματα και το δοκιμάζει σε στο υπόλοιπο 1. Επαναλαμβάνει αυτήν την διαδικασία k φορές και υπολογίζει την μέση απόδοση του μοντέλου.
- ii. **GridSearchCV**: αναζητά τις βέλτιστες τιμές υπερ-παραμέτρων μέσω **cross-validation**.
- iii. **StratifiedFold**: Παράμετρος του **GridSearch**, εξασφαλίζει ότι η κατανομή των κλάσεων σε κάθε fold είναι παρόμοια με αυτή στο αρχικό dataset για να αποφευχθούν οι ανισορροπίες.
- iv. **classification_report**: Μέρος της βιβλιοθήκης sklearn, επιστρέφει ένα report της απόδοσης του μοντέλου.
- v. **confusion_matrix**: Μέρος της βιβλιοθήκης sklearn, επιστρέφει έναν matrix που περιγράφει την αντιστοίχιση μεταξύ των πραγματικών και των προβλεπόμενων κλάσεων.
- vi.
 - TP** (Αληθώς θετικά)
 - TN** (Αληθώς αρνητικά)
 - FP** (Ψευδώς θετικά)
 - FN** (Ψευδώς αρνητικά)

$$\text{Precision} = \frac{TP}{TP + FP}$$

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

$$\text{Class-Specific Precision} = \frac{TP_i}{TP_i + FN_i}$$

$$\text{Class-Specific Accuracy} = \frac{TP_i}{TP_i + FN_i}$$

2.2 Σχεδιασμός και Υλοποίηση Γραφικού Περιβάλλοντος Χρήστη (GUI)

Το GUI υλοποιείται ως κλάση `CampaignPredictionApp`, βασισμένη στη βιβλιοθήκη `tkinter`. Κατά την εκκίνηση, δημιουργείται το κύριο παράθυρο (`root`) με συγκεκριμένο τίτλο, προεπιλεγμένες διαστάσεις και θέμα, χρησιμοποιώντας το έτοιμο θέμα `Sun Valley ttk` σε `dark mode`). Μέσα σε αυτό το παράθυρο, διαμορφώνονται δύο βασικές περιοχές:

1. Πλαίσιο Κουμπιών

- Τοποθετείται στο πάνω μέρος της εφαρμογής (`frame` με χρήση `ttk.Frame`).
- Περιλαμβάνει έξι κουμπιά, καθένα για ένα στάδιο της διαδικασίας της εφαρμογής:
 - i.«Φόρτωση Δεδομένων Προηγούμενης Καμπάνιας»
 - ii.«Εκπαίδευση Μοντέλου Πρόβλεψης με χρήση βέλτιστου K»
 - iii.«Εκπαίδευση Μοντέλου Πρόβλεψης με εισαγωγή K»
 - iv.«Φόρτωση Δεδομένων Νέας Καμπάνιας»
 - v.«Πρόβλεψη Ανταπόκρισης Νέων Πελατών»
 - vi.«Αποθήκευση Πρόβλεψης»
- Κάθε κουμπί συνδέεται με σχετική μέθοδο-χειριστή, π.χ. `self.load_past_campaign_data`, `self.on_train` κλπ.
- Η μέθοδος `_update_button_states()` ρυθμίζει ποιο κουμπί είναι ενεργό ή απενεργοποιημένο, ανάλογα με τα `flags` της εφαρμογής (`training_data_loaded`, `model_trained`, `predictions_data_loaded`, `predictions_made`). Με αυτόν τον τρόπο ο χρήστης δεν μπορεί να προχωρήσει σε επόμενο βήμα αν δεν έχει ολοκληρώσει το προηγούμενο (π.χ. δεν επιτρέπεται να τρέξει την πρόβλεψη αν δεν έχει φορτώσει πρώτα νέα δεδομένα προς χρήση για την πρόβλεψη).

2. Notebook με δύο καρτέλες (`tabs`)

- Tab 1: «Αρχείο Καταγραφής»
Περιέχει ένα `ScrolledText` widget στο οποίο εμφανίζονται μηνύματα καταγραφής (`logs`). Κάθε ενέργεια (φόρτωση αρχείου, εκπαίδευση, πρόβλεψη κλπ.) καταγράφεται με περιγραφικό κείμενο, ώστε ο χρήστης να ενημερώνεται για την εξέλιξη της εφαρμογής και τυχόν σφάλματα.
- Tab 2: «Γράφημα Πρόβλεψης Ανταπόκρισης»
Εδώ χρησιμοποιείται ένα αντικείμενο `matplotlib.Figure` ενσωματωμένο σε ένα `FigureCanvasTkAgg` ώστε να εμφανίζονται διαδραστικά γραφήματα μέσα στο παράθυρο. Πιο συγκεκριμένα, μετά την ολοκλήρωση της πρόβλεψης, καλείται η μέθοδος `responses_by_gender_pie()`, η οποία δημιουργεί δύο πίτες (`pie charts`) με την κατανομή «Yes/No» ανταποκρίσεων χωριστά ανά

φύλο. Το αποτέλεσμα σχεδιάζεται στα υποπλάισια (subplots) της figure και εμφανίζεται στο δεύτερο tab.

Λειτουργίες που προσφέρει το GUI

- Φόρτωση Αρχείων (Past & New Campaign Data):
 - i. Μέσω του διαλόγου (filedialog.askopenfilename) επιλέγεται αρχείο Excel (.xlsx) με συγκεκριμένες υποχρεωτικές στήλες.
 - ii. Το DataFrame που φορτώνεται ελέγχεται ως προς τις απαιτούμενες στήλες και, αν το index είναι RangeIndex, μετονομάζεται σε «Πελάτης N».
 - iii. Εκτελείται αμυντικός προγραμματισμός για τον έλεγχο τυχόν σφαλμάτων (π.χ. ελλιπείς στήλες, κενά αρχεία), αναφέρονται με messagebox.showerror σε pop-up παράθυρο και καταγράφονται στο log.
- Εκπαίδευση Μοντέλου k-NN:
 - i. Αυτόματη Εκπαίδευση ([on_train](#)):
 1. Αρχικοποίηση αντικειμένου KNN(neighbors=None, test_size=0.2, random_state=42).
 2. Κλήση feed_data() με τα φορτωμένα ιστορικά δεδομένα.
 3. Εύρεση βέλτιστου k μέσω του αλγόριθμου που ανέπτυξε ο Ραφαήλ (μέθοδος [find_best_neighbors](#) για εύρος k=(2,16) και folds=(2,8).
 4. Εκπαίδευση του τελικού μοντέλου με το βέλτιστο k (fit()).
 5. Γεννήτρια μετρικών επικύρωσης ([gen_metrics\(\)](#)), οι οποίες καταγράφονται στο log (accuracy, precision, recall κλπ.).
 6. Εμφάνιση μηνύματος επιτυχούς εκπαίδευσης (π.χ. «Βέλτιστο k=...»), ενημέρωση flags και πληροφορία για το επόμενο βήμα της διαδικασίας.
 - ii. Χειροκίνητη Εκπαίδευση ([manual_train](#)):
 1. Ζητείται από τον χρήστη μέσω simpledialog.askinteger να εισάγει τον αριθμό γειτόνων k.
 2. Δημιουργία KNN(neighbors=k, test_size=0.2, random_state=42).
 3. Κλήση [feed_data\(\)](#) και [fit\(\)](#).
 4. Καταγραφή και εμφάνιση των μετρικών επικύρωσης.

5. Ενημέρωση flags και καταγραφής εξέλιξης.

- Πρόβλεψη για Νέα Δεδομένα (on predict):
 - i. Έλεγχος ότι υπάρχει εκπαιδευμένο μοντέλο και φορτωμένα νέα δεδομένα πρόβλεψης.
 - ii. Κλήση `self.knn_model.predict(self.new_campaign_data, output_path=None)`, που επιστρέφει εάν DataFrame με τις προβλέψεις ανταπόκρισης για τα νέα δεδομένα πελατών.
 - iii. Ενημέρωση flag `predictions_made`, ενεργοποίηση κουμπιού «Αποθήκευση» και εμφάνιση πληροφορικού μηνύματος.
 - iv. Κλήση μεθόδου `responses_by_gender_pie()`, η οποία δημιουργεί διάγραμμα πίτας με τις προβλεπόμενες ανταποκρίσεις ανά φύλο.
- Αποθήκευση Αποτελεσμάτων (save predictions wrapper):
 - i. Εμφάνιση διαλόγου `asksaveasfilename` για αποθήκευση των προβλέψεων σε αρχείο Excel.
 - ii. Ενημέρωση του χρήστη για επιτυχία/αποτυχία/ακύρωση/σφάλμα.
 - iii. Επαναφορά όλων των flags και δεδομένων σε αρχική κατάσταση ώστε η εφαρμογή να ξεκινήσει με νέα ροή.

Εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του GUI

Για την υλοποίηση του GUI επιλέχθηκαν βιβλιοθήκες που προσφέρουν απλότητα, ευελιξία και σύγχρονη αισθητική, εξασφαλίζοντας ταυτόχρονα αξιόπιστη διαχείριση δεδομένων και εύκολη ενσωμάτωση οπτικοποιήσεων:

- **Tkinter:**

Αποτελεί βασική βιβλιοθήκη της Python για ανάπτυξη γραφικών διεπαφών και περιλαμβάνεται ήδη στο standard library, γεγονός που απαλείφει επιπλέον εξαρτήσεις. Με τη χρήση της tkinter δημιουργείται το κύριο παράθυρο της εφαρμογής, ορίζονται τα βασικά widgets (όπως Frame, Button, Notebook) και διαχειρίζονται τα συμβάντα (π.χ. κλικ σε κουμπιά ή πάτημα του πλήκτρου «Esc» για έξοδο). Επιπλέον, μέσω των έτοιμων διαλόγων (`messagebox`, `filedialog`, `simpledialog`) διευκολύνεται η επικοινωνία με τον χρήστη. Η επιλογή της tkinter βασίζεται στο ότι είναι «ελαφριά», ευέλικτη για βασικές ανάγκες GUI και συνεργάζεται άψογα με βιβλιοθήκες όπως `pandas` και `matplotlib`.

- **Sv_ttk (Sun Valley Theme)**

Το επίσημο `ttk` (το styling module της tkinter) περιορίζεται σε κλασικές θεματικές επιλογές. Η εξωτερική βιβλιοθήκη `sv_ttk` δίνει σύγχρονη αισθητική στα `ttk widgets`, επιτρέποντας να εφαρμοστεί ένα ομοιόμορφο dark theme σε όλη την εφαρμογή. Εξασφαλίζεται έτσι μοντέρνα αισθητική

χωρίς επιπλέον πολυπλοκότητα, ώστε το περιβάλλον της εφαρμογής να είναι ευχάριστο και «up-to-date» με τρέχουσες τάσεις σχεδιασμού.

- **Pandas**

Η pandas χρησιμοποιείται για την ανάγνωση αρχείων Excel και τη μεταγενέστερη επεξεργασία των δεδομένων (indexing, μετονομασίες στηλών, ομαδοποιήσεις). Η δυνατότητα εύκολης φόρτωσης και διαχείρισης πινάκων δεδομένων είναι κρίσιμη για τη ροή φόρτωσης δεδομένων των πελατών και την εκτέλεση προβλέψεων. Επιπλέον, η pandas προσφέρει ευανάγνωστη σύνταξη και αποδοτικές λειτουργίες επεξεργασίας, πράγμα που επιτρέπει γρήγορο καθαρισμό και ομαδοποίηση των δεδομένων, διευκολύνοντας την ανάπτυξη και συντήρηση του κώδικα.

- **Matplotlib**

Για την απεικόνιση γραφημάτων επιλέχθηκε η matplotlib, η πιο διαδεδομένη βιβλιοθήκη σχεδίασης γραφημάτων στην Python. Συγκεκριμένα, δημιουργούνται διαγράμματα πίτας (pie charts) που παρουσιάζουν την κατανομή «Yes/No» απαντήσεων ανά φύλο στους νέους πελάτες. Η ενσωμάτωση στη tkinter επιτυγχάνεται μέσω του FigureCanvasTkAgg, το οποίο επιτρέπει την εμφάνιση του matplotlib figure απευθείας στο GUI χωρίς να απαιτείται ξεχωριστό παράθυρο.

- **Pathlib**

Για τη διαχείριση διαδρομών αρχείων χρησιμοποιείται η Pathlib, καθώς προσφέρει άνετο χειρισμό μονοπατιών χωρίς να χρειάζεται «χειροκίνητος» συνδυασμός συμβολοσειρών. Στην εφαρμογή, εντοπίζει τον υποφάκελο data που βρίσκεται παράλληλα με τον φάκελο του κώδικα και τον ορίζει ως αρχική διαδρομή του file dialog. Αν ο φάκελος data δεν υπάρχει, τότε προσανατολίζεται στην τρέχουσα διαδρομή εργασίας (cwd). Η επιλογή της Pathlib έγινε για να διατηρηθεί η εύκολη αναγνωσιμότητα του κώδικα.

- **Typing (Optional)**

Για την τυποποίηση των συναρτήσεων χρησιμοποιούνται hints από το module typing, όπως Optional[pd.DataFrame], προκειμένου να διευκρινίζεται ότι κάποιες μεταβλητές μπορεί να έχουν τιμή None. Αυτό αυξάνει την αναγνωσιμότητα του κώδικα και επιτρέπει σε IDEs να ελέγχουν τη ροή του προγράμματος, διευκολύνοντας τόσο τον έλεγχο σφαλμάτων όσο και το debugging. Με αυτόν τον τρόπο είναι σαφές ποιες συναρτήσεις επιστρέφουν DataFrame και ποιες συναρτήσεις μπορεί να μην επιστρέψουν τίποτα, βελτιώνοντας την ποιότητα του κώδικα.

Συνοψίζοντας, η συνδυαστική χρήση των παραπάνω βιβλιοθηκών εξασφαλίζει ένα ελαφρύ, αξιόπιστο και ευανάγνωστο GUI:

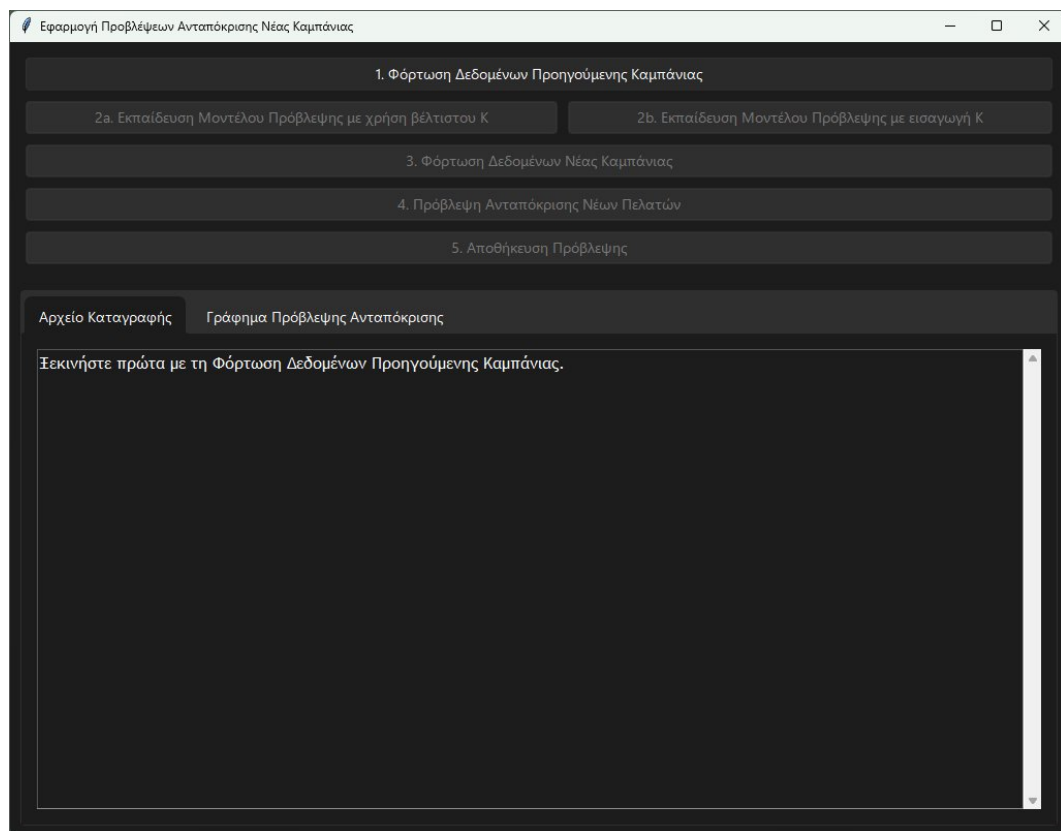
- **Tkinter**: καλύπτει τη βασική λειτουργικότητα,
- **Sv_ttk**: προσθέτει σύγχρονη εμφάνιση,

- **Pandas:** επιτρέπει άμεση και αξιόπιστη διαχείριση δεδομένων,
- **Matplotlib:** παρέχει οπτικοποίηση,
- **Pathlib:** παρέχει ευανάγνωστη διαχείριση αρχείων,
- **Typing:** υποστηρίζει την ποιότητα και συντηρησιμότητα του κώδικα.

3. Παρουσίαση Λειτουργικότητας Εφαρμογής (με στιγμιότυπα εκτέλεσης):

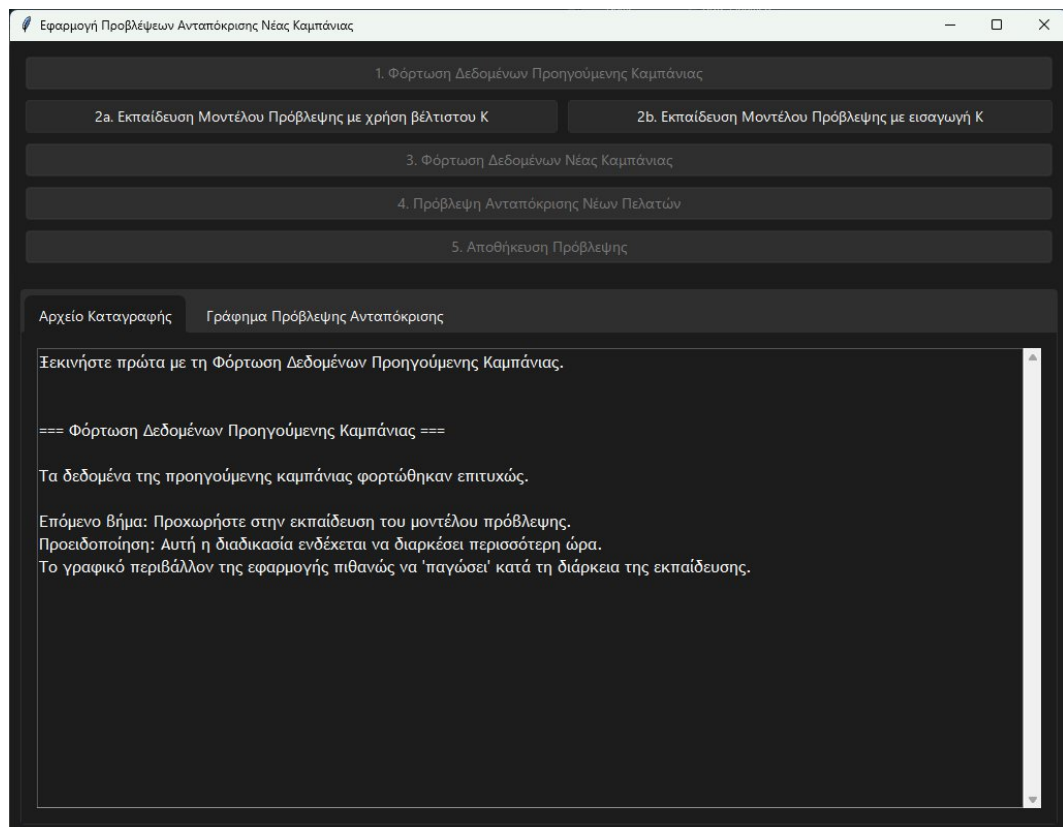
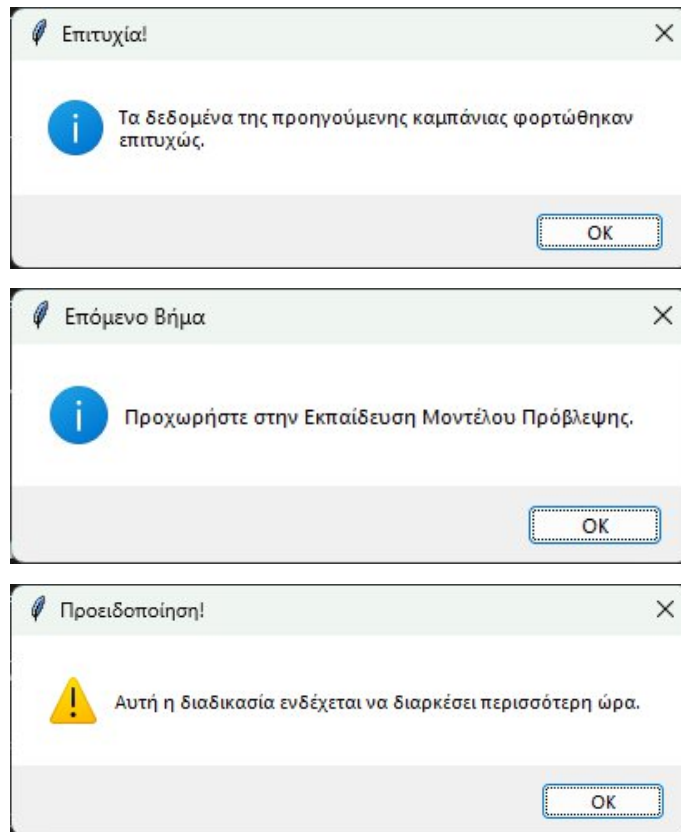
***Σημειώνουμε** πως ο χρήστης καθοδηγείται με pop-up παράθυρα καθ' όλη τη διάρκεια της εκτέλεσης της εφαρμογής. Κάθε ενέργεια του χρήστη ελέγχεται για την ορθότητα της εκτέλεσης της και το γραφικό περιβάλλον εμφανίζει σχετικό παράθυρο είτε για την επιτυχία της ή την αποτυχία της (π.χ. σε σφάλματα φόρτωσης δεδομένων κλπ).

Εκκίνηση Εφαρμογής:



Στάδιο 1: Φόρτωση Δεδομένων Προηγούμενης Καμπάνιας (

Ο χρήστης φορτώνει από αρχείο Excel 'PastCampaignData.xlsx, μέσω διαλόγου Επιλογής Αρχείου, τα δεδομένα προηγούμενης καμπάνιας που χρησιμοποιούνται για την εκπαίδευση του μοντέλου πρόβλεψης.



Στάδιο 2: Εκπαίδευση Μοντέλου.

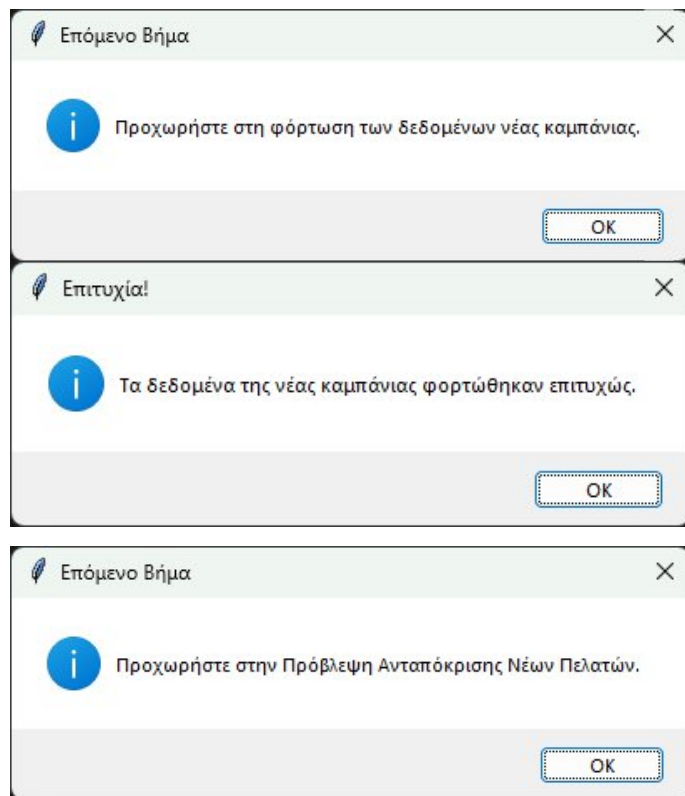
Σε αυτό το στάδιο ο χρήστης έχει την επιλογή για αυτόματη εκπαίδευση μοντέλου με χρήση βέλτιστου αριθμού γειτόνων (k) ή να επιλέξει ο ίδιος, μέσω παραθύρου διαλόγου, τη τιμή k.

Για χάρη παραδείγματος, χρησιμοποιείται η αυτόματη εκπαίδευση. Μετά την επιτυχή εκπαίδευση, εμφανίζονται τα μετρικά απόδοσης στο tab «Αρχείο Καταγραφής».



Στάδιο 3: Φόρτωση Δεδομένων Νέας Καμπάνιας

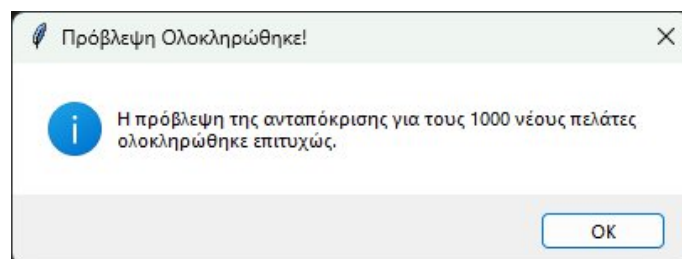
Ο χρήστης φορτώνει από αρχείο Excel 'NewCampaignData.xlsx' μέσω διαλόγου Επιλογής Αρχείου τα δεδομένα νέας καμπάνιας που χρησιμοποιούνται για την πρόβλεψη ανταπόκρισης. Μετά από επιτυχή φόρτωση των νέων δεδομένων ο χρήστης ενημερώνεται καταλλήλως και καθοδηγείται στο επόμενο βήμα.

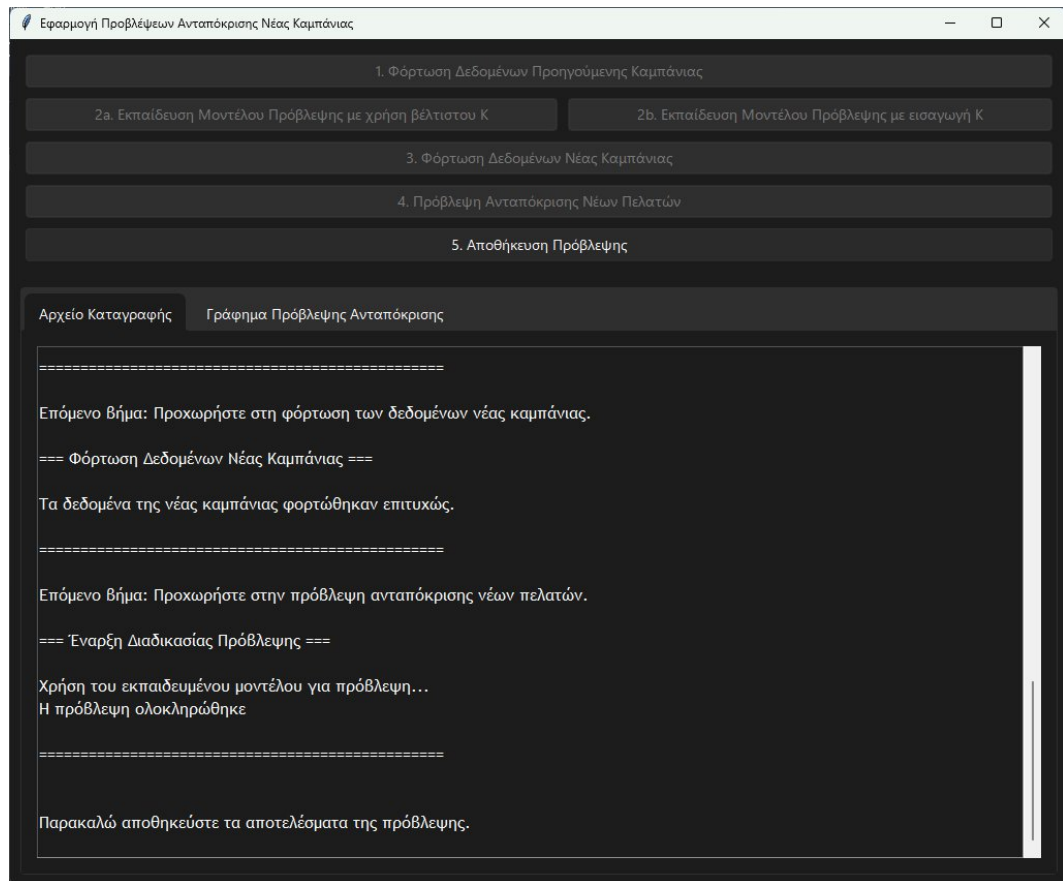
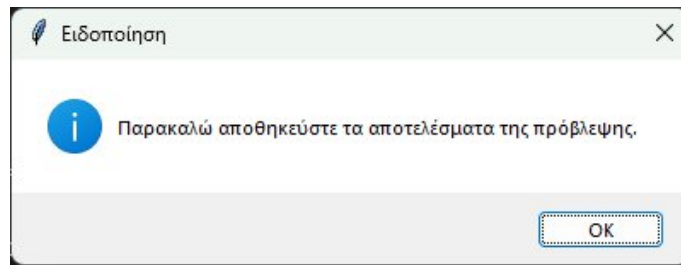




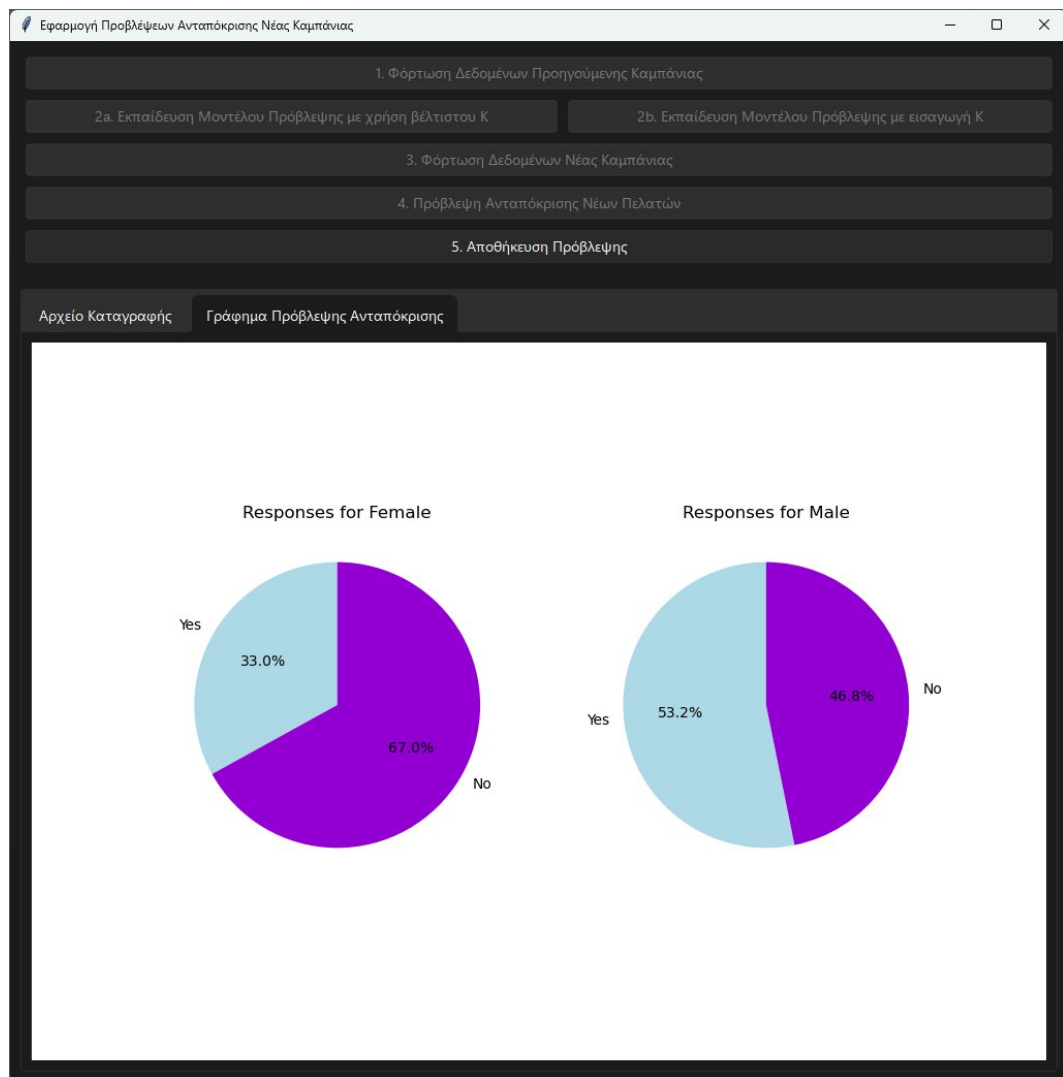
Στάδιο 4: Πρόβλεψη Ανταπόκρισης Νέων Πελατών:

Μόλις ολοκληρωθεί η πρόβλεψη, ο χρήστης καθοδηγείται στο να αποθηκεύσει τα αποτελέσματα της πρόβλεψης σε νέο αρχείο Excel.



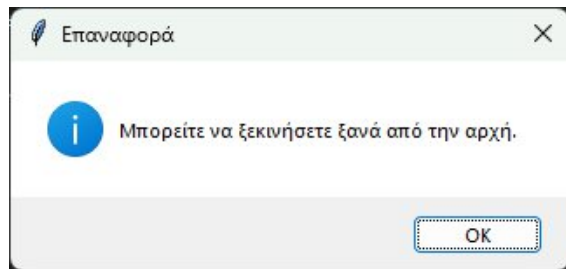
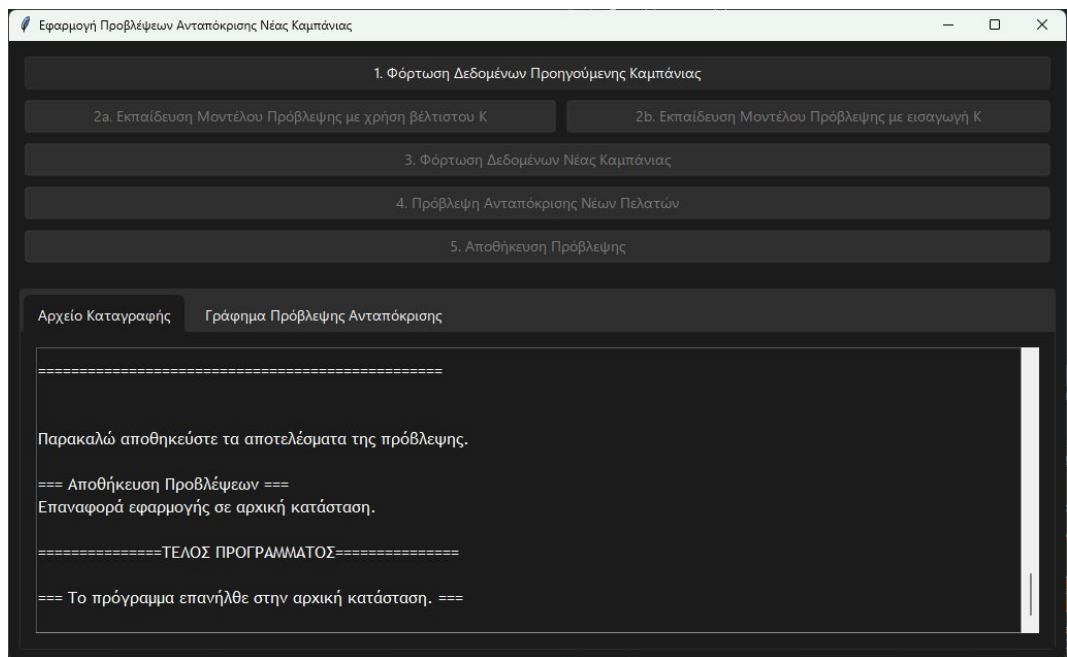


Σε αυτό το στάδιο ο χρήστης επίσης έχει τη δυνατότητα να δει οπτικοποιημένα τα αποτελέσματα της πρόβλεψης που έχει εκτελέσει το μοντέλο. Έχουμε επιλέξει να εμφανίζονται τα αποτελέσματα ταξινομημένα ανάλογα με το φύλο των πελατών τα οποία στοχεύει η νέα καμπάνια.



Στάδιο 5: Αποθήκευση Αποτελεσμάτων Πρόβλεψης και Επαναφορά στην αρχική κατάσταση

Τελικά, ο χρήστης αποθηκεύει τα αποτελέσματα της πρόβλεψης, επιλέγοντας μέσω διαλόγου τοποθεσία και ονομασία του νέου αρχείου Excel. Έπειτα, εάν η αποθήκευση ολοκληρωθεί επιτυχώς, η εφαρμογή επανέρχεται στην αρχική της κατάσταση και ο χρήστης έχει τη δυνατότητα να εκκινήσει τη διαδικασία πρόβλεψης από την αρχή (π.χ. για άλλη τιμή k, άλλο αρχείο δεδομένων πελατών κλπ.).



4. Κατανομή Εργασιών Ομάδας

Για την επιτυχή ολοκλήρωση της ομαδικής εργασίας υπήρξε σαφής κατανομή αρμοδιοτήτων και αξιοποίηση σύγχρονων εργαλείων συνεργασίας.

4.1 Κατανομή Ρόλων και Εργασιών

Η ομάδα, αποτελούμενη από τους φοιτητές Ασλανίδη Ραφαήλ, Πιτσαρή Κωνσταντίνο και Κρανίτσα Αντωνία προχώρησε στην ακόλουθη κατανομή εργασιών:

- **Ασλανίδης Ραφαήλ:** ανέλαβε τον συντονισμό της ομάδας, την οργάνωση των εργασιών και τη δημιουργία ενός GitHub repository για την αποτελεσματική συνεργασία των μελών της ομάδας. Στο τεχνικό σκέλος, ανέλαβε πλήρως την υλοποίηση της «καρδιάς» της εφαρμογής, της κλάσης KNN (k-Nearest Neighbors) με χρήση της βιβλιοθήκης `scikit-learn` (στο αρχείο `model.py`), συμπεριλαμβανομένων των μεθόδων:
 - `__init__()`: αρχικοποίησης των attributes της κλάσης KNN
 - `feed_data()`: φόρτωσης δεδομένων,
 - `fit()`: εκπαίδευσης μοντέλου,
 - `predict()`: πρόβλεψης αποτελεσμάτων,
 - `gen_metrics()`: υπολογισμού μετρικών απόδοσης,
 - `find_best_neighbors()`: αυτόματης εύρεσης του βέλτιστου αριθμού γειτόνων (k)

Επιπλέον, δημιούργησε την κλάση `Plotter` (στο αρχείο `plotter.py`) για την οπτικοποίηση των σχετικών μετρικών απόδοσης του μοντέλου. Συνέδραμε επίσης στη δημιουργία `keybind` για την έξοδο από το γραφικό περιβάλλον της εφαρμογής και στην τεχνική υποστήριξη της Κρανίτσα Αντωνίας.

- **Πιτσαρής Κωνσταντίνος:** ανέλαβε την κύρια ευθύνη για τον σχεδιασμό και την ανάπτυξη της κλάσης `CampaignPredictionApp` (στο αρχείο `gui.py`) που διαχειρίζεται το γραφικό περιβάλλον χρήστη (GUI) με χρήση της βιβλιοθήκης `tkinter`. Υλοποίησε το μεγαλύτερο μέρος των μεθόδων της διεπαφής χρήστη, συμπεριλαμβανομένων:
 - `__init__()`: της αρχικοποίησης του βασικού παραθύρου της εφαρμογής,
 - `_create_notebook()`: της δημιουργίας καρτελών σε γραφικό στοιχείο (widget) μορφής notebook,
 - `_update_button_states()`: Της διαχείρισης της κατάστασης των κουμπιών που διαχειρίζονται τη ροή της εφαρμογής,
 - `_log()`: της καταγραφής μηνυμάτων,

- `_load_data()`, `load_past_campaign_data()` και `load_new_campaign_data()`: της φόρτωσης δεδομένων από αρχεία Excel με κατάλληλο έλεγχο σφαλμάτων,
- `on_train()`: της εκκίνησης της διαδικασίας εκπαίδευσης με βέλτιστο k
- `on_predict()`: της εκκίνησης της διαδικασίας πρόβλεψης,
- `_save_predictions()` και `save_predictions_wrapper()`: της αποθήκευσης των προβλέψεων καθώς και της ορθής επαναφοράς της εφαρμογής στην αρχική της κατάσταση για επανεκκίνηση της διαδικασίας πρόβλεψης.

Συνεργάστηκε με την Κρανίτσα Αντωνία στην υλοποίηση των μεθόδων δημιουργίας και τοποθέτησης των κουμπιών της εφαρμογής (`_create_buttons()`).

Επιπλέον, ανέλαβε πλήρως τη συγγραφή αναλυτικής τεκμηρίωσης (doc-strings) του module `gui(gui.py)`.

- **Κρανίτσα Αντωνία:** Υλοποίησε πλήρως τις μεθόδους της κλάσης `CampaignPredictionApp`:

- `manual_train()`: χειροκίνητη εκπαίδευση του μοντέλου KNN μέσω εισόδου τιμής k από το χρήστη με χρήση διαλόγου,
- `responses_by_gender_pie()`: δημιουργία γραφήματος πίτας (pie chart) για την απεικόνιση των αποτελεσμάτων των προβλέψεων έπειτα από κατάλληλη προ επεξεργασία των σχετικών δεδομένων που προκύπτουν από τη πρόβλεψη ανταπόκρισης.

Συνέβαλε, επίσης, στην ανάπτυξη του γραφικού περιβάλλοντος χρήστη (GUI), συνεργαζόμενη στη δημιουργία του βασικού παραθύρου της εφαρμογής και της μεθόδου δημιουργίας και τοποθέτησης των κουμπιών της εφαρμογής του γραφικού περιβάλλοντος (`_create_buttons`). Επιπροσθέτως, ανέλαβε εξ ολοκλήρου τη συγγραφή της παρουσίασης της ομαδικής εργασίας σε μορφή διαφανειών (slides).

Όλα τα μέλη της ομάδας συνέβαλαν ενεργά στην ολοκλήρωση του έργου. Οι αρμοδιότητες περιλάμβαναν την οργάνωση της δομής του κώδικα, τη διαχείριση των εκδόσεων του μέσω της πλατφόρμας GitHub, τη διεξοδική δοκιμή της εφαρμογής για τη διασφάλιση της σταθερότητας και λειτουργικότητας της, τον εντοπισμό και την επίλυση σφαλμάτων, καθώς και την τελική διαμόρφωση της εμφάνισης και της χρηστικότητας του λογισμικού. Επιπροσθέτως, υπήρξε συλλογική συνεισφορά στη συγγραφή της ομαδικής έκθεσης και των οδηγιών χρήσης της εφαρμογής.

4.2 Εργαλεία Συνεργασίας

Η πλατφόρμα **GitHub** αποτέλεσε κεντρικό εργαλείο για την ομάδα, διευκολύνοντας την αποτελεσματική συνεργασία, την παρακολούθηση της προόδου και την διαχείριση του κώδικα. Μέσω του **GitHub**, επιτεύχθηκε η συστηματική καταγραφή των τροποποιήσεων στον κώδικα, η ευέλικτη διαχείριση των διαφορετικών εκδόσεων του λογισμικού και η υποστήριξη της ταυτόχρονης εργασίας των μελών, παράγοντες που συνέβαλαν καθοριστικά στην ομαλή ροή και επιτυχή ολοκλήρωση του έργου.

5. Βιβλιογραφία

Ακαδημαϊκές πηγές:

- **Swaroop C H (2009)**, [A Byte Of Python](#)
- **Νικόλαος Α. Αγγελιδάκης (2015)**, [Εισαγωγή στον προγραμματισμό με την Python, Ά Έκδοση](#)

Τεκμηρίωση βιβλιοθηκών Python:

- Scikit-learn:
 - [API](#)
 - [Pipeline](#)
 - [ColumnTransformer](#)
 - [KNeighborsClassifier](#)
 - [Preprocessing](#)
 - [OneHotEncoder](#)
 - [StandardScaler](#)
 - [TrainTestSplit](#)
 - [GridSearchCV](#)
 - [StratifiedKFold](#)
 - [AccuracyScore](#)
 - [PrecisionScore](#)
 - [ConfusionMatrix](#)
 - [ClassificationReport](#)
- Pandas
 - [DataFrame](#)
 - [Excel](#)
- Tkinter
 - [Tk](#)
 - o [Grid](#)
 - o [Pack](#)
 - o [Frame](#)
 - o [Button](#)
 - o [Notebook](#)
 - [FileDialog](#)
 - [Scrolledtext](#)
 - [MessageBox](#)
 - [simplifiedialog](#)
- Pathlib
 - [API](#)
- Typing
 - [API](#)
- Matplotlib
 - [API](#)
 - [PyPlot](#)
 - [Backends.backend_tkagg](#)
- Seaborn

- [Lineplot](#)
- [Barplot](#)

Tutorials:

- [Python Tkinter Tutorial](#)
- [Pathlib Module in Python](#)
- [Type Hints in Python](#)
- [Matplotlib PyPlot](#)
- [Creating Tabbed Widget with Python-Tkinter](#)
- [How to embed Matplotlib charts in Tkinter GUI?](#)
- [Python Pie Chart: Build and Style with Pandas and Matplotlib](#)
- [Plot a Pie Chart in Python using Matplotlib](#)
- [An Essential Guide to the Tk Elements By Practical Examples](#)

Extra:

- [Typst Documentation](#)
- [Sun Valley ttk theme](#)

Videos:

- [\[YouTube\] K Nearest Neighbors | Intuitive explained | Machine Learning Basics](#)
- [\[YouTube\] StatQuest: K-nearest neighbors, Clearly Explained](#)
- [\[YouTube\] Machine Learning Fundamentals: Cross Validation](#)
- [\[YouTube\] Hyperparameters Tuning: Grid Search vs Random Search](#)
- [\[YouTube\] Tkinter Beginner Course - Python GUI Development](#)
- [\[YouTube\] Create tabs in your GUI interface using Notebook - Python Tkinter GUI Tutorial #64](#)
- [\[YouTube\] Python Tutorial: Pathlib - The Modern Way to Handle File Paths](#)
- [\[YouTube\] Python Typing - Type Hints & Annotations](#)
- [\[YouTube\] Add PyPlot to TKinter interface | Essential Engineering](#)
- [\[YouTube\] ttk Style Tutorial - Customizing your Tkinter Widgets in Python](#)
- [\[YouTube\] Python Tkinter GUI Design Using ttkbootstrap - Complete Course](#)
- [\[YouTube\] Matplotlib Tutorial 6 - Pie Chart](#)
- [\[YouTube\] Matplotlib Pie Charts - How to Plot and Customize!](#)
- [\[YouTube\] Matplotlib Tutorial \(Part 3\): Pie Charts](#)

6. Παράρτημα

- Σύνδεσμος για το repository [[GitHub](#)]
- Σύνδεσμος για το εγχειρίδιο χρήσης της εφαρμογής [[GitHub](#)]