Υλοποίηση Μοντέλου KNN

Ραφαήλ Ασλανίδης

Τελικό Project ΠΛΗΠΡΟ – Ελληνικό Ανοιχτό Πανεπιστήμιο – 2025

Εισαγωγή

Ο αλγόριθμος KNN (K-Nearest Neighbors) είναι ένας αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται τόσο για ταξινόμηση (classification) όσο και για παλινδρόμηση (regression). Βασίζεται στην υπόθεση ότι δεδομένα με παρόμοια χαρακτηριστικά βρίσκονται κοντά το ένα στο άλλο σε έναν πολυδιάστατο χώρο. Η απόδοση του KNN εξαρτάται σε μεγάλο βαθμό από την επιλογή του κατάλληλου πλήθους γειτόνων K.

Στην παρούσα εργασία υλοποιείται ένας αλγόριθμος KNN με χρήση της τη βιβλιοθήκη scikit-learn και χρησιμοποιεί την μέθοδο k-fold_cross-validation με διαφορετικούς συνδυασμούς τιμών του K και πλήθους fold. Στόχος είναι να εντοπιστεί η τιμή του K που εμφανίζεται συχνότερα ως η βέλτιστη κατά τη διάρκεια των επαναληπτικών πειραμάτων, ώστε να επιλεγεί ως η πλέον κατάλληλη για το τελικό μοντέλο. Θα υπολογίζονται επίσης και οι μετρικές απόδοσης του μοντέλου, όπως accuracy, precision, class-specific accuracy, και class-specific precision.

Ο παραπάνω αλγόριθμος είναι το βασικό μέρος μιας GUI εφαρμογής που αναπτύσσεται στο πλαίσιο του ομαδικού project του μαθήματος ΠΛΗΠΡΟ στο Ελληνικό Ανοιχτό Πανεπιστήμιο για την πρόβλεψη της ανταπόκρισης πελατών σε καμπάνιες marketing με χρήση μηχανικής μάθησης.

Μεθοδολογία και Συνεισφορά

Ο αλγόριθμος KNN βασίζεται στον υπολογισμό αποστάσεων μεταξύ σημείων δεδομένων. Συνήθως (όπως και σε αυτήν την εργασία) χρησιμοποιείται η απόσταση Euclidean όπου είναι και το default μέτρο απόστασης στη βιβλιοθήκη scikit-learn.

Η σωστή επιλογή του παραμέτρου K είναι κρίσιμη για την απόδοση του μοντέλου. Μικρές τιμές K οδηγούν overfitting, ενώ πολύ μεγάλες τιμές μπορεί να οδηγήσουν σε underfitting, το optimal είναι κάπου ενδιάμεσα.

Το k-fold cross-validation, είναι μια μέθοδος που επιτρέπει την αποτίμηση της απόδοσης ενός μοντέλου με μεγαλύτερη σταθερότητα. Το dataset χωρίζεται σε k ίσα μέρη (folds), και το μοντέλο εκπαιδεύεται και αξιολογείται k φορές, κάθε φορά με διαφορετικό fold ως σύνολο δοκιμής.

Η διαδικασία αυτή εφαρμόζεται για κάθε τιμή του K (γειτόνων), και η απόδοση του μοντέλου καταγράφεται ξεχωριστά για κάθε fold. Σκοπός είναι να αναγνωριστεί η τιμή του K που εμφανίζεται συχνότερα ως η βέλτιστη.

Μέθοδοι της Κλάσης KNN

Συνοπτικά, υλοποίησα τον ακόλουθο αλγόριθμο:

- `find_best_neighbors()` Για κάθε διαφορετικό πλήθος fold (π.χ. 5, 10) τρέχουμε τον αλγόριθμο KNN με διαφορετικές τιμές του K και καταγράφουμε την απόδοση του μοντέλου. Στο τέλος, αναλύουμε τα αποτελέσματα και επιλέγουμε την τιμή του K που εμφανίζεται συχνότερα ως η βέλτιστη με τη βοήθεια της μεθόδου mode() της βιβλιοθήκης pandas.

Πέρα από την υλοποίηση του παραπάνω αλγορίθμου, υλοποίησα και τις ακόλουθες μεθόδους:

- `__init__()` Αρχικοποιεί τα attributes της κλάσης KNN.
- `feed_data()` Φορτώνει τα δεδομένα εκπαίδευσης και τα μετατρέπει σε μορφή κατάλληλη (numerical & categorical) για να χρησιμοποιηθούν από τον preprocessor του KNN.
- `fit()` Εκπαιδεύει το μοντέλο KNN με τα δεδομένα εκπαίδευσης.
- `predict()` Κάνει πρόβλεψη για νέα δεδομένα και αποθηκεύει τα αποτελέσματα σε αρχείο εάν δοθεί το `output_path`.
- `gen_metrics()` Υπολογίζει και αρχικοποιεί τις μετρικές απόδοσης του μοντέλου, όπως accuracy, precision, class-specific accuracy, και class-specific precision σε dicts αλλά και σε ένα string για εύκολη εμφάνιση.

Επιπλέον, δημιούργησα και μια κλάση `Plotter` για την οπτικοποίηση της απόδοσης του μοντέλου ανά συνδυασμό τιμών K και folds σε περίπτωση που ο χρήστης επιλέξει να χρησιμοποιήσει την μέθοδο `find_best_neighbors()`.

Μέθοδοι της Κλάσης Plotter

Η παρακάτω κλάση δημιουργήθηκε για να διευκολύνει την οπτικοποίηση των αποτελεσμάτων στο ομαδικό paper και περιλαμβάνει τις ακολούθες μεθόδους:

- `__init__()` Αρχικοποιεί τα attributes της κλάσης Plotter.
- `plot_neighbors_vs_metric_per_fold()` Δημιουργεί ένα graph που απεικονίζει την απόδοση του μοντέλου ανά συνδυασμό τιμών K και folds για μια συγκεκριμένη μετρική (π.χ. accuracy).
- `plot_trisurf_neighbor_vs_metric_per_fold()` Δημιουργεί ένα 3D graph που απεικονίζει την απόδοση του μοντέλου ανά συνδυασμό τιμών K και folds για μια συγκεκριμένη μετρική (π.χ. accuracy).
- `plot_mean_metric_per_fold()` Δημιουργεί ένα graph που απεικονίζει την μέση απόδοση του μοντέλου ανά fold για μια συγκεκριμένη μετρική (π.χ. accuracy).

Επίσης, βοήθησα στη δημιουργία keybind για την έξοδο του GUI της εφαρμογής στον κώδικα του Κώνσταντινου Πιτσαρή, καθώς και με κάποια προβλήματα που προέκυψαν κατά το setup του project στον υπολογιστή της Αντωνίας Κρανίτσας. Ακόμη, ως συντονιστής της ομάδας, βοήθησα στην οργάνωση των εργασιών και στην δημιουργία ενός GitHub repository για την αποθήκευση του κώδικα και των αρχείων της εργασίας.

Ώρες Εργασίας

- Εισαγωγική μελέτη του αλγορίθμου KNN και των τεχνικών cross-validation και grid search. (≈ **1** ώρα)
- Υλοποίηση της κλάσης `KNN` με χρήση της βιβλιοθήκης scikit-learn. (≈ **6** ώρες)
- Δημιουργία της κλάσης `Plotter` για την οπτικοποίηση των αποτελεσμάτων. (≈ **2** ώρες)
- Τεκμηρίωση του κώδικα και των μεθόδων με σχόλια και docstrings. (≈ **1** ώρα)
- Συγγραφή του ατομικού paper. (≈ **2** ώρες)
- Συνεισφορά στη συγγραφή του ομαδικού paper. (≈ **5** ώρες)

Βιβλιογραφία

- Scikit-learn
  - API
  - Pipeline
  - ColumnTransformer
  - KNeighborsClassifier
  - Preprocessing
  - OneHotEncoder
  - StandardScaler
  - TrainTestSplit
  - GridSearchCV
  - StratifiedKFold
  - AccuracyScore
  - PrecisionScore
  - ConfusionMatrix
  - ClassificationReport
- **Pandas**
  - DataFrame
- **Matplotlib**
  - Pyplot
- **Seaborn**
  - Lineplot
  - Barplot
- **LLMs**
  - Χρησιμοποιήθηκε το ChatGPT της OpenAI ως βοηθητικό εργαλείο μάθησης για την κατανόηση εννοιών που δεν μου ήταν ξεκάθαρες κατά την μελέτη του documentation των βιβλιοθηκών. Δεν υπήρξε αντιγραφή κώδικα.
- **Βίντεο**
  - [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
- **Extra**
  - Typst Documentation
- **Bonus**
  - Terry A. Davis