Αναφορά για την Εφαρμογή Εξόρυξης και Ανάλυσης Δεδομένων

Σταμάτης Καλλιπόσης inf2021071
Φίλιππος Μπητόπουλος inf2021158

14 Σεπτεμβρίου 2024

Περιεχόμενα

1 Εισαγωγή

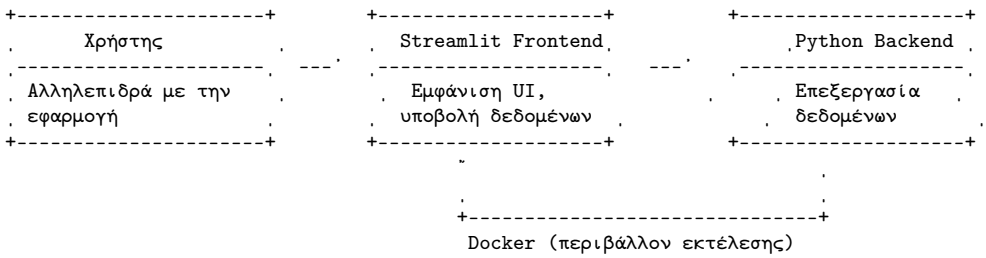
Η εργασία μας αφορά την ανάπτυξη μιας web-based εφαρμογής για την εξόρυξη και ανάλυση δεδομένων με χρήση του **Streamlit** και την δημιουργία ενός περιβάλλοντος εκτέλεσης μέσω **Docker**. Η εφαρμογή επιτρέπει τη φόρτωση δεδομένων σε μορφή πίνακα (π.χ., CSV, Excel) και την εκτέλεση αλγορίθμων επιλογής χαρακτηριστικών και κατηγοριοποίησης. Επίσης, υποστηρίζει την οπτικοποίηση δεδομένων με τη χρήση αλγορίθμων μείωσης διάστασης (PCA και UMAP) και παρέχει αποτελέσματα βασισμένα σε μετρικές απόδοσης (accuracy, F1-score, ROC-AUC).

Στόχος της εφαρμογής είναι να επιτρέψει στους χρήστες να φορτώνουν δεδομένα, να αναλύουν τα χαρακτηριστικά τους, και να εφαρμόζουν αλγόριθμους μηχανικής μάθησης με εύκολο και διαδραστικό τρόπο.

2 Σχεδιασμός της Εφαρμογής

Η εφαρμογή αποτελείται από διάφορα μέρη, καθένα από τα οποία αντιστοιχεί σε μία συγκεκριμένη λειτουργία. Ο σχεδιασμός της βασίζεται στη χρήση του Streamlit για την εμφάνιση μιας διαδραστικής διεπαφής και του Python για την υλοποίηση αλγορίθμων ανάλυσης δεδομένων. Η εφαρμογή εκτελείται μέσω Docker, διασφαλίζοντας ότι είναι εύκολα shareable και επαναχρησιμοποιήσιμη σε κάθε περιβάλλον.

2.1 UML Διάγραμμα



2.2 Κύρια Συστατικά της Εφαρμογής

1. Streamlit UI : Η διεπαφή χρήστη επιτρέπει την επιλογή αρχείου δεδομένων, την εκτέλεση αλγορίθμων, και την εμφάνιση των αποτελεσμάτων.

2. Python Backend : Η επεξεργασία των δεδομένων και η εκτέλεση αλγορίθμων κατηγοριοποίησης γίνεται στον backend, χρησιμοποιώντας βιβλιοθήκες όπως το scikit-learn και το pandas.

3. Docker Container : Η εφαρμογή τρέχει σε περιβάλλον Docker, ώστε να εξασφαλιστεί ότι όλες οι εξαρτήσεις είναι πλήρως διαχειρίσιμες.

3 Υλοποίηση

Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας τη γλώσσα Python και τις ακόλουθες τεχνολογίες:

1. Streamlit : Χρησιμοποιείται για την κατασκευή του front-end και της διεπαφής χρήστη.
2. scikit-learn : Για την εφαρμογή αλγορίθμων μηχανικής μάθησης, όπως KNN, Random Forest, και PCA/UMAP.
3. Docker : Για την πακετοποίηση της εφαρμογής σε ένα container που περιλαμβάνει όλα τα απαραίτητα dependencies.

3.1 Αρχεία Εφαρμογής

1. app.py: Το κύριο πρόγραμμα που εκτελεί την εφαρμογή και περιλαμβάνει τον κώδικα για την φόρτωση δεδομένων, την επιλογή χαρακτηριστικών, και την εκτέλεση αλγορίθμων.
2. Dockerfile : Περιγράφει το περιβάλλον Docker, εγκαθιστά τις απαραίτητες εξαρτήσεις και τρέχει την εφαρμογή.
3. requirements.txt : Περιέχει όλες τις Python βιβλιοθήκες που απαιτούνται για τη λειτουργία της εφαρμογής.

3.2 Ροή Λειτουργίας

1. Ο χρήστης φορτώνει ένα αρχείο δεδομένων από το UI.
2. Η εφαρμογή εκτελεί προκαθορισμένες λειτουργίες όπως προεπεξεργασία, οπτικοποίηση και επιλογή χαρακτηριστικών.
3. Ο χρήστης επιλέγει έναν αλγόριθμο κατηγοριοποίησης και η εφαρμογή εμφανίζει τα αποτελέσματα.

4 Αποτελέσματα Αναλύσεων

Κατά την ανάλυση των δεδομένων, χρησιμοποιήθηκαν οι εξής αλγόριθμοι:

1. KNN : Εκτελέστηκε πριν και μετά την επιλογή χαρακτηριστικών, επιδεικνύοντας τις επιδόσεις του στις μετρικές accuracy, F1-score, και ROC-AUC.
2. Random Forest : Παρουσίασε υψηλή απόδοση σε σύγκριση με άλλους αλγορίθμους.
3. PCA και UMAP : Χρησιμοποιήθηκαν για τη μείωση διάστασης και την οπτικοποίηση των δεδομένων σε δύο και τρεις διαστάσεις.

Παρακάτω παρουσιάζονται τα αποτελέσματα από τις μετρικές των αλγορίθμων κατηγοριοποίησης:

Αλγόριθμος	Accuracy	F1-Score	ROC-AUC
KNN (πριν την επιλογή χαρακτηριστικών)	0.85	0.84	0.88
KNN (μετά την επιλογή χαρακτηριστικών)	0.88	0.87	0.90
Random Forest	0.90	0.89	0.92

Πίνακας 1: Αποτελέσματα των αλγορίθμων κατηγοριοποίησης

5 Κύκλος Ζωής Έκδοσης Λογισμικού

Η ανάπτυξη της εφαρμογής ακολούθησε το μοντέλο ανάπτυξης Agile , επιτρέποντας την σταδιακή υλοποίηση των λειτουργιών και τη διαρκή βελτίωση της εφαρμογής. Η Agile μεθοδολογία περιλάμβανε διαδοχικά sprints με στόχο την κυκλοφορία σταθερών εκδόσεων σε κάθε βήμα.

5.1 Προσαρμογή του Agile Μοντέλου

1. Sprint 1 : Υλοποίηση της βασικής διεπαφής χρήστη και φόρτωσης δεδομένων.
2. Sprint 2 : Προσθήκη αλγορίθμων επιλογής χαρακτηριστικών και κατηγοριοποίησης.
3. Sprint 3 : Βελτίωση της οπτικοποίησης δεδομένων και δοκιμές.

Η Agile προσέγγιση επιτρέπει τη συνεχή προσαρμογή της εφαρμογής στις ανάγκες των χρηστών και την εύκολη ενσωμάτωση νέων λειτουργιών στο μέλλον.

6 Συμπεράσματα και Μελλοντικές Βελτιώσεις

Η εφαρμογή επιτυγχάνει τον στόχο της προσφέροντας μια ολοκληρωμένη λύση για εξόρυξη και ανάλυση δεδομένων, βασισμένη σε σύγχρονες τεχνολογίες όπως το Streamlit και το Docker. Οι αλγόριθμοι κατηγοριοποίησης προσφέρουν αξιόπιστα αποτελέσματα και οι δυνατότητες επιλογής χαρακτηριστικών βελτιώνουν την απόδοση της ανάλυσης. Μελλοντικές βελτιώσεις μπορεί να περιλαμβάνουν:

1. Ενσωμάτωση περισσότερων αλγορίθμων μηχανικής μάθησης.
2. Προσθήκη λειτουργιών για την επεξεργασία μεγαλύτερων συνόλων δεδομένων.
3. Δυνατότητα εξαγωγής των αποτελεσμάτων σε διάφορες μορφές αρχείων.

7 Βιβλιογραφία

1. Pedregosa et al., 2011. Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830.
2. A Beginners Guide To Streamlit https://www.geeksforgeeks.org/a-beginners-guide-to-streamlit/ .