

Ανάπτυξη Εφαρμογής για Ανάλυση Δεδομένων

Σπυρίδων Κιτσπς Π2018119, Σιμεών Θεοφανίδης inf2021063, Στέλιος Βούρλος inf2021033

30 Μαΐου 2024

Περίληψη

Η παρούσα εργασία περιγράφει την ανάπτυξη μιας διαδικτυακής εφαρμογής για εξόρυξη και ανάλυση δεδομένων χρησιμοποιώντας το Streamlit. Η εφαρμογή ενσωματώνει δεδομένα σε μορφή πίνακα και υποστηρίζει διάφορες λειτουργίες οπτικοποίησης και μηχανικής μάθησης.

1 Εισαγωγή

Η παρούσα εργασία περιγράφει την ανάπτυξη μιας διαδικτυακής εφαρμογής για εξόρυξη και ανάλυση δεδομένων.

Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας το Streamlit και προσφέρει διάφορες λειτουργίες οπτικοποίησης και μηχανικής μάθησης για την ανάλυση των δεδομένων. Ο κύριος στόχος αυτής της εργασίας είναι να παρέχει μια πλατφόρμα για την ευκολότερη ανάλυση δεδομένων, χρησιμοποιώντας τεχνικές μείωσης διάστασης, κατηγοριοποίησης και ομαδοποίησης.

2 Σχεδιασμός Εφαρμογής

Η εφαρμογή αναπτύχθηκε με τη χρήση του Streamlit για την ανάπτυξη της διεπαφής χρήστη και των εργαλείων της Python για την ανάλυση δεδομένων και την οπτικοποίηση.

2.1 Φόρτωση Δεδομένων

Τα δεδομένα φορτώνονται από αρχεία CSV χρησιμοποιώντας τη βιβλιοθήκη pandas.

```
import pandas as pd
```

```
def load_data(file):  
    if file is not None:
```

```

        data = pd.read_csv(file)
    return data
return None

```

2.2 Οπτικοποίηση Δεδομένων

Οι τεχνικές PCA and t-SNE χρησιμοποιούνται για τη μείωση των διαστάσεων των δεδομένων.

```

from sklearn.decomposition import PCA
from sklearn.manifold import TSNE

def plot_pca(data):
    pca = PCA(n_components=2)
    components = pca.fit_transform(data)
    return components

def plot_tsne(data):
    tsne = TSNE(n_components=2)
    components = tsne.fit_transform(data)
    return components

```

3 Αποτελέσματα Αναλύσεων

Τα αποτελέσματα των αλγορίθμων κατηγοριοποίησης και ομαδοποίησης παρουσιάζονται παρακάτω.

3.1 Αποτελέσματα Κατηγοριοποίησης

```

from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.neighbors import KNeighborsClassifier

X_train, X_test, y_train, y_test = train_test_split(
    knn = KNeighborsClassifier(n_neighbors=5)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    print(classification_report(y_test, y_pred))

```

3.2 Αποτελέσματα Ομαδοποίησης

```

from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3)

```

```
clusters = kmeans.fit_predict(
#                                     clusters)
```

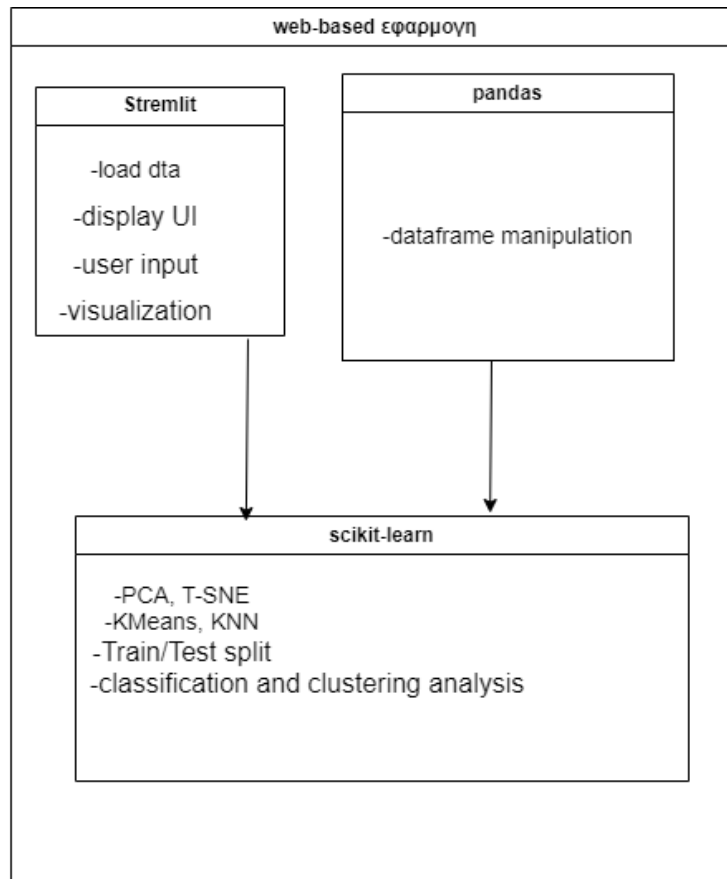
4 Συμπεράσματα

Η εφαρμογή προσφέρει μια ευέλικτη πλατφόρμα για την ανάλυση δεδομένων, επιτρέποντας στους χρήστες να φορτώσουν δεδομένα, να εκτελέσουν οπτικοποιήσεις και να εφαρμόσουν αλγορίθμους μηχανικής μάθησης για κατηγοριοποίηση και ομαδοποίηση. Μελλοντικές βελτιώσεις περιλαμβάνουν την προσθήκη επιπλέον αλγορίθμων και την επέκταση των δυνατοτήτων οπτικοποίησης.

5 Συνεισφορά Ομάδας

- Σιμεών Θεοφανίδης inf2021063 : Ανάπτυξη Κώδικα, Δοκιμές και ανάλυση δεδομένων
- Σπυρίδων Κίτσος π2018119 : Σχεδίαση Διεπαφής Χρήστη, Τεκμηρίωση και αξιολόγηση αλγορίθμων
- Στέλιος Βούρλος inf 2021033: Δημιουργία της αναφοράτου UML διαγράμματος και Latex.

6 UML Διάγραμμα



Σχήμα 1: UML Διάγραμμα της εφαρμογής

7 Κύκλος Ζωής Έκδοσης Λογισμικού

Η ανάπτυξη της εφαρμογής ακολουθεί το μοντέλο Αγιλε, το οποίο επιτρέπει την ευέλικτη και σταδιακή ανάπτυξη του λογισμικού. Οι κύριες φάσεις του κύκλου ζωής είναι οι εξής:

7.1 Σχεδιασμός

Σε αυτή τη φάση καθορίζονται οι απαιτήσεις του έργου και προγραμματίζονται οι εργασίες. Χρησιμοποιούμε εργαλεία όπως το Jira ή το Trello για την παρακολούθηση των εργασιών.

7.2 Ανάπτυξη

Η υλοποίηση των λειτουργιών γίνεται σε μικρούς, επαναλαμβανόμενους κύκλους. Χρησιμοποιούμε το Git για την παρακολούθηση των αλλαγών στον κώδικα και τη συνεχή ενσωμάτωση των αλλαγών μέσω GitHub.

7.3 Δοκιμή

Σε κάθε κύκλο ανάπτυξης, η εφαρμογή ελέγχεται για να διασφαλιστεί η ποιότητα και η λειτουργικότητα. Δημιουργούμε αυτοματοποιημένες δοκιμές για τη διασφάλιση της λειτουργικότητας.

7.4 Ανασκόπηση

Κατά τη διάρκεια της ανασκόπησης, η ομάδα αξιολογεί την πρόοδο και εντοπίζει ευκαιρίες για βελτιώσεις. Οι κριτικές κώδικα και η ανατροφοδότηση από τα μέλη της ομάδας συμβάλλουν στη συνεχή βελτίωση της εφαρμογής.

7.5 Διανομή

Η διανομή της τελικής έκδοσης της εφαρμογής γίνεται με τη χρήση Δοσχερ για την πακετοποίηση και διανομή της εφαρμογής.

7.6 Συνεχής ενσωμάτωση και παράδοση

Διαρκής ενσωμάτωση νέων λειτουργιών και βελτιώσεων, με συνεχή ενημέρωση και παράδοση της εφαρμογής στους χρήστες.

7.7 Ανατροφοδότηση από χρήστες

Η συλλογή ανατροφοδοτήσεων από χρήστες είναι κρίσιμη για τη βελτίωση της εφαρμογής και τη διόρθωση σφαλμάτων.

7.8 Ενημερώσεις

Τακτικές ενημερώσεις για την προσθήκη νέων λειτουργιών και βελτιώσεων.

8 Github

<https://github.com/eftasteros13/Ergasia>