## 📘 TextLab - Εργαλείο Ανάλυσης Κειμένου και Φυσικής Γλώσσας με Streamlit

### 🔍 Περιγραφή Εφαρμογής

Η εφαρμογή **TextLab** είναι ένα διαδραστικό εργαλείο γραμμένο σε **Python** με χρήση της βιβλιοθήκης **Streamlit**, το οποίο επιτρέπει την επεξεργασία και ανάλυση κειμένου με τεχνικές **επεξεργασίας φυσικής γλώσσας (NLP)**.

Απευθύνεται σε φοιτητές, εκπαιδευόμενους ή επαγγελματίες που θέλουν:

- Να κατανοήσουν το συναίσθημα ενός κειμένου.
- Να συγκρίνουν την ομοιότητα δύο κειμένων.
- Να ομαδοποιήσουν (cluster) πολλαπλά κείμενα με βάση το νόημά τους.
- Να εξάγουν λέξεις-κλειδιά.
- Να επεξεργαστούν CSV αρχείο με κείμενα.
- Να κάνουν ανάλυση αναρτήσεων από το Reddit.

---

### 🧱 Αρχιτεκτονική και Λειτουργικά Μέρη

#### 🗀 1. app.py - Το περιβάλλον διεπαφής χρήστη

Αυτό το αρχείο περιέχει τη **Streamlit εφαρμογή** και χειρίζεται όλη την επικοινωνία με τον χρήστη.

#### 🧠 Γενική Ροή

- Ο τίτλος της εφαρμογής είναι **"TextLab - Ανάλυση Κειμένου & Επεξεργασία Φυσικής Γλώσσας"**.
- Ο χρήστης επιλέγει τη **γλώσσα του κειμένου**: Ελληνικά ή Αγγλικά.
- Υπάρχουν **6 καρτέλες λειτουργιών (tabs)**:

---

### 🧪 Λειτουργικές Καρτέλες (Tabs)

#### 📄 1. Ανάλυση Κειμένου

- Ο χρήστης εισάγει ένα κείμενο και πατά το κουμπί **"Ανάλυση Κειμένου"**.
- Η συνάρτηση analyze_text() κάνει:
    - **Ανάλυση συναισθήματος** (π.χ., θετικό, αρνητικό, ουδέτερο).
    - **Μετατροπή σε διανύσματα (embeddings)** για μελλοντική επεξεργασία.

- Παρουσίαση συναισθήματος με **emoji** και **πίνακα με στοιχεία**.

### 🆚 2. Σύγκριση Κειμένων

- Δύο κείμενα συγκρίνονται μέσω της συνάρτησης compare_texts() που:
  - Χρησιμοποιεί διανύσματα νοήματος (embeddings).
  - Υπολογίζει **ομοιότητα cosine** (0 = διαφορετικά, 1 = παρόμοια).

### 🧠 3. Clustering Κειμένων

- Ο χρήστης εισάγει πολλαπλά κείμενα, ένα ανά γραμμή.
- Με χρήση της cluster_texts(), γίνεται ομαδοποίηση (KMeans clustering).
- Τα κείμενα παρουσιάζονται οργανωμένα ανά ομάδα (cluster).

### 🔑 4. Εξαγωγή Λέξεων-Κλειδιών

- Το κείμενο αναλύεται με **TF-IDF** για τις πιο σημαντικές λέξεις.
- Η συνάρτηση extract_keywords() επιστρέφει τις κορυφαίες λέξεις.

### 📂 5. CSV Ανάλυση

- Ο χρήστης ανεβάζει ένα αρχείο CSV που περιέχει κείμενα.
- Επιλέγει στήλη και γίνεται μαζική ανάλυση συναισθήματος για κάθε γραμμή.

### 🔍 6. Ανάλυση Αναρτήσεων Reddit

- Ο χρήστης ορίζει ένα subreddit και αριθμό αναρτήσεων.
- Η εφαρμογή χρησιμοποιεί τη fetch_reddit_posts() για άντληση τίτλων αναρτήσεων και εφαρμόζει analyze_text() σε κάθε post.

---

### 🔧 Αναλυτικά για το αρχείο utils.py

Το αρχείο **utils.py** περιέχει τις βασικές συναρτήσεις επεξεργασίας:

**1. analyze_text(text, lang)**

- Χρησιμοποιεί το **BERT sentiment model** από HuggingFace.
- Μετατρέπει το κείμενο σε **διανυσματική μορφή** με το SentenceTransformer.
- Επιστρέφει:
  - Ετικέτα συναισθήματος (π.χ., 5 stars).
  - Emoji + score.
  - Πρώτα 5 στοιχεία του embedding.

**2. compare_texts(t1, t2)**

- Μετατρέπει και τα δύο κείμενα σε embeddings.
- Υπολογίζει και επιστρέφει τη **συνάφεια cosine similarity**.

**3. cluster_texts(texts, k=2)**

- Μετατρέπει όλα τα κείμενα σε διανύσματα.
- Χρησιμοποιεί **KMeans clustering** για να τα ομαδοποιήσει σε k κατηγορίες.

**4. extract_keywords(text, lang)**

- Εφαρμόζει **TF-IDF vectorization** για την εξαγωγή σημαντικών όρων.
- Υποστηρίζει αγγλικά και ελληνικά κείμενα (όχι explicit stopwords για ελληνικά).

**5. fetch_reddit_posts(subreddit, limit)**

- Κάνει **HTTP αίτημα** στο Reddit API.
- Λαμβάνει τους τίτλους των hot posts ενός subreddit.
- Χρησιμοποιεί requests και αναλύει μόνο τον τίτλο κάθε post.

---

**□ Τεχνολογίες & Βιβλιοθήκες**

- **Streamlit**: Διαδραστικό UI.
- **Transformers (HuggingFace)**: Ανάλυση συναισθήματος με BERT.
- **Sentence-Transformers**: Embeddings για semantic similarity και clustering.
- **Scikit-learn**: TF-IDF και KMeans.
- **Pandas**: Διαχείριση CSV δεδομένων.
- **Requests**: Αιτήματα προς Reddit API.

---

**🎓 Εκπαιδευτική Χρησιμότητα**

Η εφαρμογή είναι ιδανική για:

- Μαθήματα **Φυσικής Γλώσσας (NLP)**.
- **Εργαστήρια Data Science** ή **Μηχανικής Μάθησης**.
- Παρουσίαση πρακτικών εφαρμογών AI σε **γλωσσικά δεδομένα**.
- Φοιτητές που θέλουν να δουν πώς συνδέεται θεωρία (π.χ. embeddings, similarity) με πρακτική.

---

## 🏁 Προτάσεις για Φοιτητές

- Μπορούν να επεκτείνουν το utils.py με πρόσθετες λειτουργίες όπως:
    - **Ανάλυση θεμάτων (topic modeling)**.
    - **NER (Named Entity Recognition)**.
    - **Μετάφραση ή περίληψη κειμένου**.
- Μπορούν να φτιάξουν projects για:
    - **Σύγκριση γλωσσικού ύφους** σε συγγραφείς.
    - **Συναισθηματική ανάλυση σχολίων social media**.
    - **Εργαλεία για φοιτητικές έρευνες και συνεντεύξεις**.