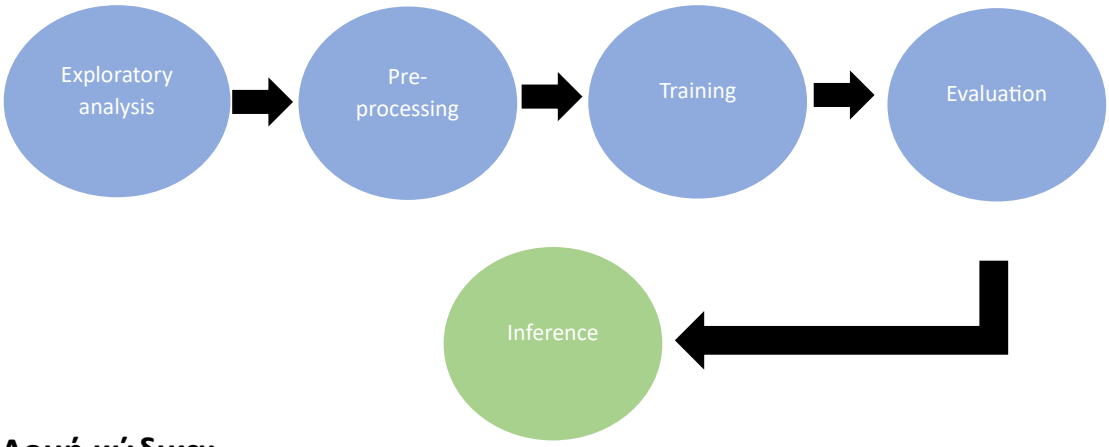
Assignment 1: Multilabel Classification

Στο πρώτο assignment έχουμε ένα dataset από 50k publications και χρειάζεται να τα ταξινομήσουμε σε 15 labels (multi-label classification). Ξεκινάμε κάνοντας exploratory analysis ώστε να κατανοήσουμε τα δεδομένα, στην συνέχεια preprocessing ώστε να επεξεργαστούμε το κείμενο και να το φέρουμε στη μορφή που προϋποθέτουν τα μοντέλα, έπειτα εκπαιδεύουμε τα μοντέλα και το αξιολογούμε και τέλος τα κάνουμε deploy ταξινομώντας νέα κείμενα. Στο κομμάτι της ταξινόμησης επιλέχθηκαν τρεισ προσεγγίσεις, η μια είναι One-Vs-Rest όπου για κάθε κλάση εκπαιδεύτηκε ένας logistic regressor, στη δεύτερη προσέγγιση έγινε fine-tune του BERT και στην τρίτη εκπαιδεύτηκε ένα LSTM. Το BERT περιορισμού στα resources δεν μπόρεσε να τελειώσει την εκπαίδευση ενώ το lstm είχε χαμηλή ακρίβεια, και άρα επιλέχθηκε η πρώτη προσέγγιση.



Δομή κώδικα:

-data : περιέχονται όλα τα δεδομένα που χρησιμοποιήθηκαν

-notebooks: ΄το jupyter notebook που χρησιμοποιήθηκε για το exploratory analysis

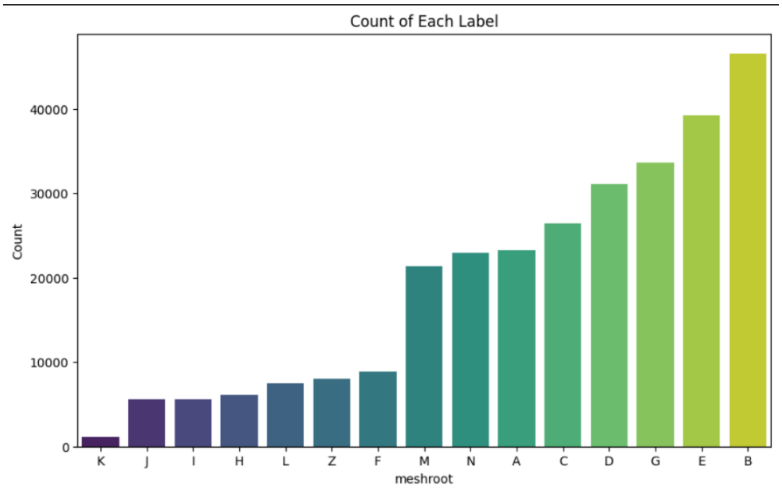
-src: ο βασικός κώδικας του project. Το αρχείο app.py εμπεριέχει το exposed api για inference ενώ τα αρχειο train_linear.py, train_bert, train_lstm χρησιμοποιήθηκαν για την εκπαίδευση των μοντέλων. Στον φάκελο utils βρίσκονται καποια εργαλεία (preprocessing και evaluation) ενώ στον φάκελο classifiers, όλοι οι classifiers που αποθηκεύτηκαν.

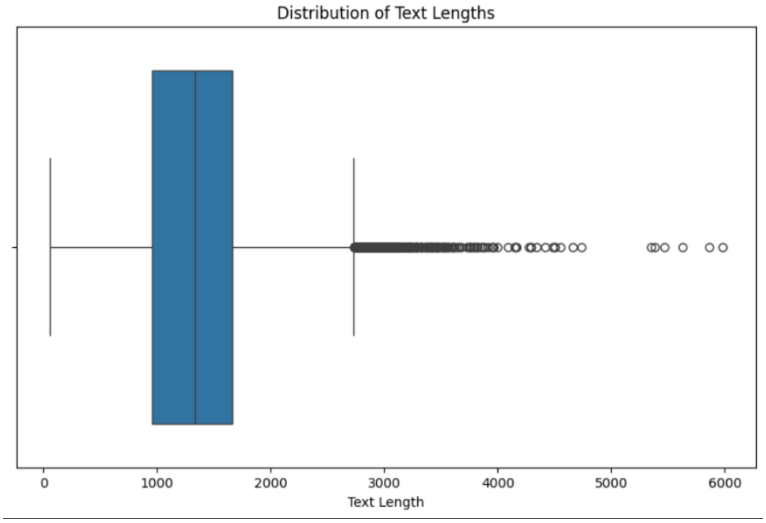
-READM.md

-requirements.txt

Exploratory Analysis

Αρχικά έψαξα αν υπάρχουν missing values, ενώ έπειτα είδα την κατανομή των labels. Στη συνέχεια έλεγξα το μέγεθος των κειμένων ώστε να μην συμπεριλάβω ακραίες τιμές καθώς επίσης και τη γλώσσα που ήταν γραμμένα. Τέλος, εφόσον το dataset ήταν imbalanced (εφόσον οι κλάσεις δεν ήταν ισο-κατανεμημένες) δημιουργήθηκε και ένα δεύτερο (sampled_dataset) όπου όλες οι κλάσεις είχαν ίδιο αριθμό δεδομένων.





Training - Inference

Όπως προαναφέρθηκε η προσέγγιση που ακολουθήθηκε ήταν one vs rest. Δηλαδή για καθεμία από τις 15 κλάσεις εκπαιδεύτηκε ένα logistic regressor ο οποίος έπαιρνε σαν είσοδο τα embeddings των κειμένων. Ως embedder χρησιμοποιήθηκε ο transformer “neuml/pubmedbert-base-embeddings” ο οποίος έχει εκπαιδευτεί σε medical research papers και κρίθηκε ως ο καταλληλότερος.

Προκειμένου να τρέξουμε την εφαρμογή

$python main.py

και από ένα άλλο terminal προκειμένου να κάνουμε ένα request:

$python client.py

Αξιολόγηση

Για την αξιολόγηση του μοντέλου χρησιμοποιήθηκε το classification report, δηλαδή υπολογίστηκαν το accuracy, precision, recall και f1score. Επίσης εφόσον το πρόβλημα αφορούσε multi-label classification χρησιμοποιήθηκε το Hamming Loss.