**Ανάκτηση Πληροφορίας**

Εργαστηριακή Άσκηση 2021

**Όνομα:** Μηναδάκης Εμμανουήλ **ΑΜ:** 1041815 **email:** minadakis@ceid.upatras.gr

Η άσκηση υλοποιήθηκε στη γλώσσα προγραμματισμού python. Η εγκατάσταση της python στα Windows 10 γίνεται μέσω του Microsoft Store. Η έκδοση που χρησιμοποιήθηκε είναι η 3.10. Για το πρώτο και δεύτερο ερώτημα χρησιμοποιήθηκαν οι βιβλιοθήκες elasticsearch, csv και warnings. Για το τρίτο ερώτημα χρησιμοποιήθηκαν επίσης οι βιβλιοθήκες genism.models και nltk. Το τέταρτο ερώτημα δεν έχει υλοποιηθεί.

Μια σύντομη περιγραφή των βιβλιοθηκών που αναφέρονται παραπάνω είναι:

**Elasticsearch**: Παρέχει τη δυνατότητα σύνδεσης με την elasticsearch. Εγκαθίσταται με την εντολή

python -m pip install elasticsearch

**csv**: Βοηθάει στο import των dataset στην elasticsearch

python -m pip install csv

**warnings**: Μορφοποιεί την έξοδο του προγράμματος ώστε να μην φαίνονται μερικές άχρηστες πληροφορίες. Δεν χρειάζεται εγκατάσταση.

**gensim.models**: Κάνουμε import το Word2Vec από αυτήν τη βιβλιοθήκη για να χρησιμοποιήσουμε την τεχνική των word embeddings.

python -m pip install gensim

python -m pip install word2vec

**nltk**: Βοηθάει στο tokenize των περιλήψεων των βιβλίων από όπου και κατεβάζουμε τα αρχεία ‘stopwords’.

python -m pip install nltk

Σύντομη περιγραφή της διαδικασίας:

Αρχικά, εγκατέστησα την Elasticsearch και το γραφικό περιβάλλον Kibana. Δημιούργησα ένα .bat file για να ενεργοποιώ και τις δύο υπηρεσίες ταυτόχρονα με τον παρακάτω κώδικα:

START cmd /C E:\DOWNLOADS\elasticsearch-7.16.1-windows-x86\_64\elasticsearch-7.16.1\bin\elasticsearch.bat

START cmd /C E:\DOWNLOADS\kibana-7.16.1-windows-x86\_64\kibana-7.16.1-windows-x86\_64\bin\kibana.bat

Αρχικά δημιουργήσαμε το αρχείο parser.py το οποίο κάνει import τα BX-Books.csv και BX-Book-Ratings.csv στην elasticsearch. H Η έξοδος του προγράμματος (αφού περιμένουμε μερικά λεπτά) είναι όπως παρακάτω:

Text

Description automatically generated

Βεβαιωνόμαστε ότι τα αρχεία φορτώθηκαν εκτελώντας την παρακάτω εντολή στο console του Kibana:

Graphical user interface, text, application

Description automatically generated

Βλέπουμε ότι τα indexes των books και ratings υπάρχουν.

Για το δεύτερο ερώτημα υλοποίησα τη main.py η οποία εκτελεί την αναζήτηση φαίνεται παρακάτω:

Text

Description automatically generated

Έχουμε επιλέξει τον χρήση 8 με το keyword “clara”. Γνωρίζουμε ότι ο συγκεκριμένος χρήστης έχει βαθμολογήσει το βιβλίο Clara Callan με 5. Παρακάτω φαίνεται η αναζήτηση του ίδιου keyword από έναν άλλο χρήστη ο οποίος δεν έχει βαθμολογήσει το βιβλίο. Παρατηρήστε τη διαφορά:

Text

Description automatically generated

Βλέπουμε ότι το βιβλίο Clara Callan δεν βρίσκεται τόσο ψηλά στα αποτελέσματα διότι έχει χαμηλότερο rating. Αυτό σημαίνει ότι η αναζήτηση λειτουργεί με βάση το τι βιβλία έχει βαθμολογήσει ο χρήστης. Ένα βιβλίο το οποίο έχει βαθμολογηθεί με καλό βαθμό να φαίνεται πιο ψηλά.

Για το τρίτο ερώτημα υλοποίησα το Word2Vec.py το οποίο μαντεύει τις βαθμολογίες που θα έβαζε ο χρήστης σε παρόμοια βιβλία με αυτά που έχει βαθμολογήσει. Αυτό γίνεται αφού παίρνουμε όλα τα summaries από τα βιβλία που έχει βαθμολογήσει ένας χρήστης, τα ενοποιούμε σε ένα, κάνουμε tokenize και δημιουργούμε ένα model το οποίο με βάσει τις λέξεις που υπάρχουν μέσα σε αυτό βρίσκει ομοιότητες με τα summaries που προκύπτουν από τις αναζητήσεις. Υπάρχουν αναλυτικά σχόλια μέσα στο αρχείο του κώδικα. Μία έξοδος του προγράμματος είναι όπως φαίνεται παρακάτω:

Text

Description automatically generated

Αυτό που βλέπουμε είναι ότι στο μέρος πάνω από τις παύλες, έχει βρει ποια βιβλία από αυτά που υπάρχουν στα αποτελέσματα, παρουσιάζουν ομοιότητα με αυτά που έχει βαθμολογήσει και τους βάζει ένα rating το οποίο προκύπτει από τα summaries που αυτά έχουν.