

# Readme

## Μέλη Ομάδας

Δανάη Σκαρλάτου / **2908**

Θοδωρής Δήμας / **2682**

**GitHub** repository: <https://github.com/ThodorisDimas/InformationRetrievalProject.git>

## [ A ] Συλλογή των εγγράφων

Η συλλογή μας αποτελείται από τις 1000 καλύτερες ταινίες στο διάστημα 2006 έως 2016, σύμφωνα με το [www.imdb.com](http://www.imdb.com).

Η **πληροφορία** που έχουν (αυτή που θα διατηρηθεί) είναι:

**Title** : ο τίτλος της ταινίας

**Description** : η σύντομη περιγραφή της ταινίας

**Directors** : οι σκηνοθέτες της ταινίας

**Actors** : οι ηθοποιοί της ταινίας

**Year** : η χρονιά που κυκλοφόρησε η ταινία

Το **format** των δεδομένων βρίσκεται σε μορφή csv και η πηγή από την οποία αντλήσαμε τα δεδομένα δίνεται παρακάτω.

**Πηγή** δεδομένων: <https://www.kaggle.com/datasets/PromptCloudHQ/imdb-data>

## [ B ] Σύντομη περιγραφή του σχεδιασμού του συστήματος

### [ B1 ] Στόχος και λειτουργικότητας του συστήματος

Στόχος του συστήματος είναι να παρέχει σε έναν χρήστη έναν εύκολο και απλό τρόπο αναζήτησης ταινιών. Μέσα από την εφαρμογή κάποιος θα έχει την δυνατότητα να θέσει αφηρημένα ερωτήματα με απλές λέξεις-κλειδιά είτε να κάνει μία πιο εξειδικευμένη αναζήτηση χρησιμοποιώντας πληροφορίες που μπορεί να αναφέρονται σε συγκεκριμένα στοιχεία των ταινιών. Με την χρήση ενός κατάλληλου γραφικού περιβάλλοντος θα είναι σε θέση να αλληλοεπιδρά με το σύστημα δίνοντας το περιεχόμενο εισόδου στο σύστημα και λαμβάνοντας κατάλληλη έξοδο με τα ζητούμενα αποτελέσματα. Επίσης θα είναι σε θέση να αναδιατάσσει τα αποτελέσματα με διάφορους τρόπους, καθώς και να διατηρεί το ιστορικό των αναζητήσεων του. Τέλος θα παρέχονται και δυνατότητες καθώς και προτάσεις για να βελτιώσει τις αναζητήσεις του και να λάβει καλύτερα αποτελέσματα.

### [ B2 ] Ανάλυση κειμένου και κατασκευή ευρετηρίου

Για την **προ επεξεργασία** των εγγράφων έχουμε αφαιρέσει ορισμένα πεδία από το dataset που θεωρήσαμε ότι δεν είχαν κάποιο ιδιαίτερο ενδιαφέρον. Έπειτα γίνεται parse του εγγράφου, το οποίο είναι δομημένο κατάλληλα ώστε να μπορούμε κάθε φορά να δημιουργήσουμε και να εισάγουμε τις πληροφορίες στα **document** που θα δημιουργήσουμε.

Η **μονάδα εγγράφου** ορίζεται από μία γραμμή στο αρχείο csv και τα αντίστοιχα **πεδία** είναι η πληροφορία όπως την ορίσαμε παραπάνω για το έγγραφο.

Τα έγγραφα θα εισαχθούν στο ευρετήριο του **IndexWriter** και με τον **IndexSearcher** θα γίνετε η αναζήτηση σε αυτό το ευρετήριο.

Για κάθε ένα από τα πεδία θα δημιουργήσουμε και ένα ευρετήριο για αναζήτηση με χρήση συγκεκριμένης πληροφορίας. Στα **Indexed fields** που αφορούν τα description και actors θα υπάρχει **analyzer** για να είναι **tokenized** και θα είναι **stored**.

## [ B3 ] Αναζήτηση

Η αναζήτηση των ταινιών θα γίνετε μέσα από ένα πλαίσιο αναζήτησης που θα γράφει ο χρήστης λέξεις κλειδιά. Δίπλα από το πλαίσιο αυτό θα υπάρχουν εξειδικευμένες επιλογές αναζήτησης και ανάλογα με την επιλογή του χρήστη θα εκτελείται η αντίστοιχη ερώτηση σε συγκεκριμένο πεδίο.

Ο **Query Parser** θα είναι σε θέση ανάλογα με το ερώτημα να κάνει κατάλληλο parse και να υποστηρίζει τα εξής ερωτήματα:

- Περιλαμβάνουν κάποιον όρο
- Περιλαμβάνουν παραπάνω από έναν όρους
- Εμφάνιση όρου σε συγκεκριμένο πεδίο
- Εμφάνιση όρων σε διαφορετικά πεδία
- Ολόκληρες φράσεις με ακρίβεια αναγραφής (Phrase Matches)
- Wild Cards
- Range Matches

Για την δόμηση ερωτημάτων θα υποστηρίζονται και λέξεις κλειδιά με παρόμοια συνάφεια, ορθογραφική διόρθωση για να διορθώνει ο χρήστης τυπογραφικά λάθη του καθώς αυτόματη συμπλήρωση λέξεων καθώς θα πληκτρολογεί μία φράση.

Θα χρησιμοποιήσουμε **embeddings** για την βελτίωση της αναζήτησης. Ακόμα θα υπάρχει κατάλληλη αποθήκευση των αναζητήσεων του χρήστη σε ένα **ιστορικό** αναζητήσεων και μέσα από αυτό θα προκύπτουν προτάσεις για τους όρους αναζήτησης του.

## [ B4 ] Παρουσίαση αποτελεσμάτων

Τα αποτελέσματα θα παρουσιάζονται ανά 10 (κατά μέγιστο αριθμό) και σε σελίδες. Οι λέξεις κλειδιά που χρησιμοποιήθηκαν στο ερώτημα θα εμφανίζονται έντονα τονισμένες. Το γραφικό περιβάλλον θα παρέχει την δυνατότητα ταξινόμησης των αποτελεσμάτων με διάφορους

τρόπους. Πιο συγκεκριμένα θα επιτρέπεται ταξινόμηση κατά αύξουσα και φθίνουσα σειρά με βάση την τιμή του πεδίου, τόσο σε αλφαριθμητικές όσο και σε αριθμητικές τιμές. Επίσης θα υπάρχει ταξινόμηση με βάση την συνάφεια που απέδωσε το σύστημα στα αποτελέσματα μας. Έπειτα από το parse που θα πραγματοποιήσει ο **Query Parser** θα λάβουμε ως έξοδο κάποια **ScoreDocs** και έπειτα τα αποτελέσματα που θα διατηρηθούν θα τα διατηρεί ο **TopDocs** θα τα αναδιατάσσουμε κατάλληλα σύμφωνα με τις επιλογές που θα μας έχει ορίσει ο χρήστης.

Τέλος θα μπορεί γίνεται **ομαδοποίηση των αποτελεσμάτων** με βάση πεδία που θα ορίζει ο χρήστης μέσα από κατάλληλη επιλογή στο γραφικό περιβάλλον.