

## ΜΥΥΕ003: Ανάκτηση Πληροφορίας

### Εαρινό Εξάμηνο 2021-2022

Εργασία: Μηχανή αναζήτησης ταινιών

Σωτήρης Παναγιώτου, AM: 4456

Νικολέττα Μπιντζηλαίου, AM: 4435

- Link στο GitHub repository της εργασίας:

<https://github.com/sotblad/anaktisi>

- Link στο youtube για το βίντεο με το demo:

<https://www.youtube.com/watch?v=5OFRFAJl47M>

- Περιληπτικά για το Scraper:

Τα δεδομένα συλλέχθηκαν από την σελίδα imdb με scraper.

Συγκεκριμένα, συλλέξαμε τους τίτλους που εμφανίζονται με αύξοντα

The screenshot shows the IMDb website interface. At the top, there's a navigation bar with the IMDb logo, a menu icon, a search bar, and links for IMDbPro, Watchlist, Sign In, and EN. Below the navigation bar, the main content area displays a list of movies under the heading "Feature Film/TV Movie/Short Film (Sorted by Popularity Ascending)". The list includes the following details for the top three movies:

- 1. CODA (2021)**  
PG-13 | 111 min | Comedy, Drama, Music  
★ 8,1 ☆ Rate this 74 Metascore  
As a CODA (Child of Deaf Adults) Ruby is the only hearing person in her deaf family. When the family's fishing business is threatened, Ruby finds herself torn between pursuing her passion at Berklee College of Music and her fear of abandoning her parents.  
Director: Sian Heder | Stars: Emilia Jones, Marlee Matlin, Troy Kotsur, Daniel Durant  
Votes: 92,168
- 2. Έγκλημα στον Νείλο (2022)**  
K-12 | 127 min | Crime, Drama, Mystery  
★ 6,4 ☆ Rate this 52 Metascore  
While on vacation on the Nile, Hercule Poirot must investigate the murder of a young heiress.  
Director: Kenneth Branagh | Stars: Tom Bateman, Annette Bening, Kenneth Branagh, Russell Brand  
Votes: 68,072
- 3. Morbius (2022)**  
PG-13 | 104 min | Action, Adventure, Horror  
★ 5,2 ☆ Rate this 35 Metascore  
Biochemist Michael Morbius tries to cure himself of a rare blood disease, but he inadvertently infects himself with a form of vampirism instead.  
Director: Daniel Espinosa | Stars: Jared Leto, Matt Smith, Adria Arjona, Jared Harris  
Votes: 28,357

αριθμό δημοφιλίας- αριθμό ψήφων.

([https://www.imdb.com/search/title/?title\\_type=feature,tv\\_movie,short](https://www.imdb.com/search/title/?title_type=feature,tv_movie,short))

Συγκεκριμένα, από τη κάθε featured ταινία, συλλέγουμε μέσω του scraper τον url της εικόνας του poster της, τον τίτλο, την ημερομηνία που βγήκε, το είδος, την βαθμολογία της στα 100, την διάρκεια της σε λεπτά, την περιγραφή/περίληψη, τον/τους σκηνοθέτες και τους ηθοποιούς με πρωταγωνιστικούς ρόλους. Αυτές τις πληροφορίες τις αποθηκεύουμε σε ένα csv αρχείο, όπου οι παραπάνω κατηγορίες χωρίζονται σε στήλες με βάση το χαρακτήρα «|».



1 **CODA** (2021)  
PG-13 | 111 min | Comedy, Drama, Music  
★ 8,1 ★ Rate this 74 Metascore  
As a CODA (Child of Deaf Adults) Ruby is the only hearing person in her deaf family. When the family's fishing business is threatened, Ruby finds herself torn between pursuing her passion at Berklee College of Music and her fear of abandoning her parents.  
Director: [Sian Heder](#) | Stars: [Emilia Jones](#), [Marlee Matlin](#), [Troy Kotsur](#), [Daniel Durant](#)  
Votes: 92.168

Π.χ.

img	title	year	genre	rating	duration	description
<a href="https://m.media-amazon.com/ima/Εγκλημα_στον_Νείλο">https://m.media-amazon.com/ima/Εγκλημα_στον_Νείλο</a>	2022	['Crime', 'Drama', 'Mystery']	64	127	While on vacation on the Nile, Hercule Poirot must investigate	
<a href="https://m.media-amazon.com/ima/Morbius">https://m.media-amazon.com/ima/Morbius</a>	2022	['Action', 'Adventure', 'Horror']	52	104	Biochemist Michael Morbius tries to cure himself of a rare blood	
<a href="https://m.media-amazon.com/ima/The_Bubble">https://m.media-amazon.com/ima/The_Bubble</a>	2022	['Comedy']	47	126	A group of actors and actresses stuck inside a pandemic bubble	
<a href="https://m.media-amazon.com/ima/The_Batman">https://m.media-amazon.com/ima/The_Batman</a>	2022	['Action', 'Crime', 'Drama']	83	176	When the Riddler, a sadistic serial killer, begins murdering ke	
<a href="https://m.media-amazon.com/ima/Φανταστικά_Ζώα:_Τα_Μυστικά_του_Ντάμπλντορ">https://m.media-amazon.com/ima/Φανταστικά_Ζώα:_Τα_Μυστικά_του_Ντάμπλντορ</a>	2022	['Adventure', 'Family', 'Fantasy']	66	142	Albus Dumbledore assigns Newt and his allies with a mission	
<a href="https://m.media-amazon.com/ima/Sonic:_Η_Ταινία_2">https://m.media-amazon.com/ima/Sonic:_Η_Ταινία_2</a>	2022	['Action', 'Adventure', 'Comedy']	70	122	When the manic Dr Robotnik returns to Earth with a new ally,	
<a href="https://m.media-amazon.com/ima/Τα_Πάντα_Όλα">https://m.media-amazon.com/ima/Τα_Πάντα_Όλα</a>	2022	['Action', 'Adventure', 'Comedy']	89	139	An aging Chinese immigrant is swept up in an insane adventure	
<a href="https://m.media-amazon.com/ima/CODA">https://m.media-amazon.com/ima/CODA</a>	2021	['Comedy', 'Drama', 'Music']	80	111	As a CODA (Child of Deaf Adults) Ruby is the only hearing pers	

Συλλογή εγγράφων: Τρέχουμε το αρχείο imdbScraper.py δίνοντας ως όρισμα τον αριθμό των ταινιών που θέλουμε να αποθηκεύσουμε. Ο αριθμός αυτός χρησιμοποιείται για να ξέρουμε πόσες σελίδες από τη imdb ιστοσελίδα θα χρειαστεί να «ψάξουμε». Φτιάχνουμε το αρχείο scraped\_movies.csv στο οποίο γράφουμε ώστε να έχει τις στήλες-πεδία: img|title|year|genre|rating|duration|description|directors|starsList. Επίσης, φτιάχνουμε μία άδεια τελική λίστα με ταινίες.

Όσο δεν έχουμε φτάσει των αριθμό σελίδων που χρειαζόμαστε:

- ✓ Το url των σελίδων που θα χρειαστούμε μέχρι τις πρώτες 10.000 ταινίες είναι το  
`"https://www.imdb.com/search/title/?title_type=feature,tv_movie,short&count=250&start=" + str(1+(i*250))`, όπου `i=0..αριθμός των συνολικών σελίδων που θα χρειαστούμε`. Μετά απο αυτές, το url αλλάζει  
(Π.χ.`https://www.imdb.com/search/title/?title_type=feature,tv_movie,short&count=250&after=WzE0MTI0LCJ0dDE3MzIxMjMwliw5NzUxXQ%3D%3D` ) και έτσι λαμβάνουμε το url της επόμενης σελίδας κατευθείαν απο τον υπερσύνδεσμο στην html.
  
- ✓ Αν ο αριθμός των ταινιών που εισάγαμε στο terminal είναι μικρότερος των 250, τότε ζητάμε αυτό τον αριθμό featured, διαφορετικά, αφαιρούμε από τον συνολικό αριθμό τις πρώτες 250 υπάρχουσες και ζητάμε ακριβώς 250.
  
- ✓ Ανοίγουμε το url και κόβουμε τον html κώδικα από το `<div class="lister-list">` μέχρι το `</div class="desc">`, που περιέχει τις πληροφορίες που μας ενδιαφέρουν για όλες τις ταινίες. Ανάλογα με τη σελίδα στην οποία είμαστε βρίσκουμε και κρατάμε το κατάλληλο url για την επόμενη. Καλούμε την την μέθοδο `twofifty` με ορίσματα τις πληροφορίες που μας ενδιαφέρουν στην html, την λίστα με τις ταινίες και των αριθμών των ζητούμενων featured, για να ξεχωρίσουμε και να αποθηκεύσουμε τις ταινίες με τα πεδία τους.
  
- ✓ `def twofifty(movies, moviesList, want):`  
Χρησιμοποιούμε έναν μετρητή για να μετράμε τις ταινίες που είναι "έτοιμες", δηλαδή για τις οποίες έχουμε αποθηκεύσει όλα τα απαραίτητα πεδία. Όσο αυτός ο μετρητής είναι μικρότερος από των αριθμών των ζητούμενων featured, «σπάμε» την πληροφορία (movies) σε ξεχωριστές ταινίες, σε κάθε μία από τις 250 featured. Με τη σειρά:
  1. Κρατάμε url της φωτογραφίας που αντιστοιχεί σε κάθε poster.
  2. Κρατάμε το header και από αυτόν τον τίτλο.

3. Κρατάμε το κατάλληλο κομμάτι του header για να βρούμε τη χρονολογία. Αν ανοίγει παρένθεση και ο τίτλος συνεχίζει, την προσθέτουμε στο πεδίο title, διαφορετικά, η χρονολογία έχει πάντα 4 ψηφία.

Στη συνέχεια:

1. Ψάχνουμε το επόμενο κομμάτι που μας ενδιαφέρει που έχει να κάνει με το info για τη διάρκεια, το είδος και την βαθμολόγηση. Μία ταινία μπορεί να ανήκει σε πάνω από ένα είδος και για αυτό είναι λίστα.
2. Βρίσκουμε το κομμάτι για την περίληψη-περιγραφή. Μπορεί να «λείπει» ή να μην είναι ολοκληρωμένη λόγω αναλυτικής περιγραφής.
3. Προχωράμε στο κομμάτι για το σκηνοθέτη και τους ηθοποιούς. Οι σκηνοθέτες μπορεί να είναι πάνω από ένας και οι ηθοποιοί είναι πολλαπλοί συνηθώς, οπότε πρέπει να «χωριστούν» αντίστοιχα και να μπουν μέσα σε μία λίστα για την κάθε ταινία.

Μπορούμε πλέον να προχωρήσουμε στην επόμενη featured. Αυξάνουμε τον μετρητή και προσθέτουμε τα «πεδία» μας σε μία προσωρινή λίστα. Ελέγχουμε ότι η λίστα αυτή δεν υπάρχει ήδη στη συνολική λίστα με τις ταινίες και την προσθέτουμε, γράφοντας τις αντίστοιχες πληροφορίες στο csv αρχείο μας.

- Λειτουργία του gkougkl:

Στόχος του συστήματος μας είναι ο χρήστης να μπορεί να ψάξει κάποια πληροφορία για μία ταινία και το σύστημα να του επιστρέφει τον τίτλο, και μια σύντομη περιγραφή της πιο σχετικής (με το μεγαλύτερο score) και τους τίτλους από τις επόμενες 9 σχετικές αντίστοιχα.

#### Ανάλυση κειμένου και κατασκευή ευρετηρίου:

Για την προεπεξεργασία των άρθρων για τη δημιουργία του εγγράφου, χρησιμοποιούμε τον StandardAnalyzer, που είναι ο default της Lucene.

Δεν έχουμε πληροφορία για urls ή emails, αλλά θέλουμε να αφαιρούνται τα stop words και να γίνονται lowercase τα tokens.

Η μονάδα εγγράφου που θα προσθέτουμε στον IndexWriter, θα είναι μια ταινία σαν 'σύνολο', σαν αντικείμενο, δηλαδή μια ταινία με όλα τα χαρακτηριστικά της, μια γραμμή από το αρχείο csv. Ο CSVParser που διαβάζει το csv ανάλογα με το path, διαβάζει κάθε γραμμή, παίρνει τα περιεχόμενα κάθε στήλης ξεχωριστά με βάση το «|» και τα αποθηκεύει σε έναν πίνακα από string columns.

Φτιάχνουμε ένα entity Movie, με attributes imageUrl, title, year, genre, rating, duration, description, directors, stars. Τα πεδία genre, directors και stars αρχικά είναι λίστες από String, μιας που μπορούν να περιέχουν πάνω από μία 'τιμή'. Μέσα στο Movie, αφαιρούμε τους χαρακτήρες για τις λίστες, αυτάκια κτλπ. Έτσι, κάθε μονάδα εγγράφου θα αποτελείται από τα 9 πεδία (fields) αντίστοιχα, τα imageUrl, title, year, genre, rating, duration, description, directors, stars όπως αναφέρθηκαν παραπάνω, για να μπορεί να τα διαχειριστεί η Lucene. Το imageUrl είναι StringField, καθώς θέλουμε να το αποθηκεύσουμε αλλά όχι να το κάνουμε analyse και index. Τα υπόλοιπα 8 πεδία που προστίθενται στο document, είναι TextField, καθώς θέλουμε να γίνουν tokenized.

### Αναζήτηση:

Για την αναζήτηση, έχουμε ένα αντικείμενο QueryParser που θα χρησιμοποιεί τον StandardAnalyzer, για την επεξεργασία του input του χρήστη. Ανάλογα με το αν ο χρήστης θέλει να ψάξει σε όλο το λεξικό ή μόνο σε ένα συγκεκριμένο πεδίο (custom search), θα πρέπει να διαχειριζόμαστε το query διαφορετικά.

Στην πρώτη περίπτωση, δημιουργούμε ένα αντικείμενο Query κάνοντας parse το input και το search sensitivity από το slider, μέσω του QueryParser. Όσο μικρότερο είναι το sensitivity (αριστερά στο slider) τόσο αυστηρότερη είναι η αναζήτηση, ενώ όσο μεγαλύτερο είναι (δεξιά στο slider), τόσο πιο ελαστική-χαλαρή γίνεται.

Στην δεύτερη περίπτωση, ανάλογα με τα ποια κουμπιά έχει πατήσει ο χρήστης, τα προσθέτουμε στα fields και κάνουμε parse πλέον μέσω του

MultiFieldQueryParser. Χρησιμοποιούμε αντίστοιχα το input και το search sensitivity από το slider.

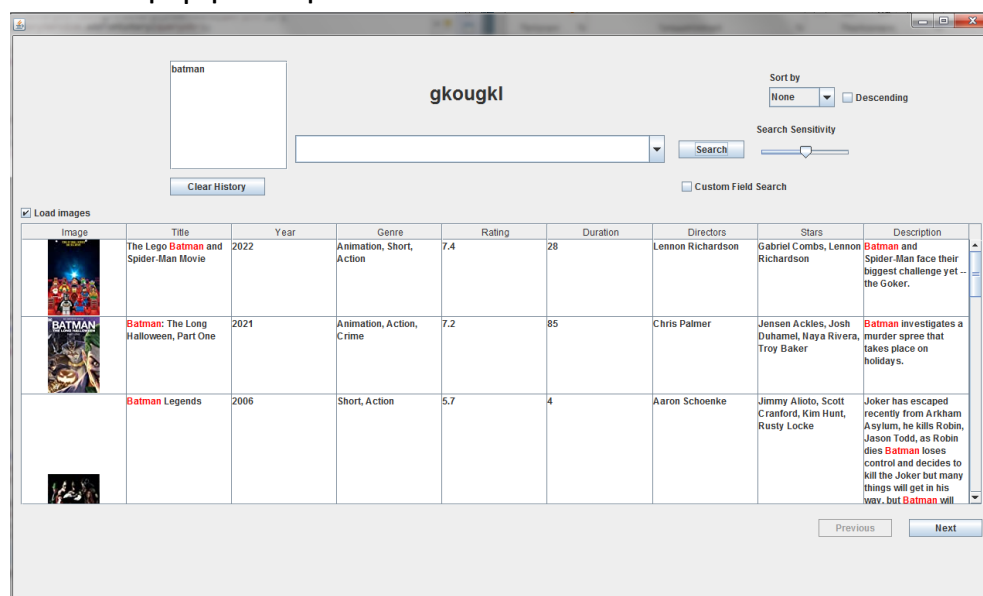
Συνεχίζοντας, θα δημιουργήσουμε ένα αντικείμενο IndexReader και ένα αντικείμενο IndexSearcher. Το αντικείμενο IndexSearcher θα χρησιμοποιεί τον reader και θα δείχνει μέσω του TopDocs στα έγγραφα που ταιριάζουν στα κριτήρια αναζήτησης (query). Αν ο χρήστης επιλέξει τα αποτελέσματα να φαίνονται ταξινομημένα με κάποιο από τα 3 διαθέσιμα πεδία year, duration και rating, θα κάνουμε sort μέσω του αντικειμένου searcher και τα docs θα είναι σε διάταξη. Διαφορετικά, τα docs μένουν ως έχουν. Έχουμε επίσης ένα αντικείμενο SpellChecker, με accuracy 0.3, το οποίο χρησιμοποιούμε για να κάνουμε index το πεδίο

Τέλος, για να προτείνουμε στον χρήστη διαφορετικά ερωτήματα παίρνουμε όλο το ιστορικό σαν ένα String και χρησιμοποιώντας τη suggestSimilar του SpellChecker, φτιάχνουμε ένα πίνακα από String. Για κάθε ένα String στον πίνακα, το προσθέτουμε στο searchField.

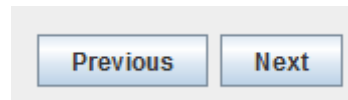
Παρουσίαση Αποτελεσμάτων: Το σύστημα μας παρουσιάζει τα αποτελέσματα σε διάταξη με βάση τη συνάφεια τους με το ερώτημα. Συγκεκριμένα, βάζουμε τα hits από τα TopDocs, στον πίνακα με τις πληροφορίες-πεδία που θέλουμε και την λέξη κλειδί υπογραμμισμένη.

- GUI

Συνολική εμφάνιση:



Προαιρετικά, ο χρήστης μπορεί να περάσει στην επομένη (και προηγούμενη) σελίδα των αποτελεσμάτων [σειρά 289-302, 323-337, 359-374], όπως και να δει τα poster των ταινιών. [σειρά 305-317, 338-355, 375-391]



<input checked="" type="checkbox"/> Load images	
Image	Title
	Life Flight: The Movie

Επιπλέον, ο χρήστης μπορεί να δει το ιστορικό των αναζητήσεών του και να το διαγράψει. Αυτό επιτυγχάνεται κρατώντας τη λέξη κλειδί από το searchFiled και αποθηκεύοντας/διαγράφοντας το μέσω του HistoryService. [σειρά 231-246, 403-410]

batman  
life

Clear History

Σύμφωνα με αυτό το ιστορικό, γίνονται 5 προτάσεις στον χρήστη για εναλλακτικά ερωτήματα, κάτω από το search bar. [σειρά 274-286]

life  
beautiful  
dark

Clear History

gkougkl

A Beautiful Curse  
A beautiful woman  
A Beautiful Life  
A Beautiful Now  
All the Beautiful Things

Title

Year

Συνεχίζοντας, η εφαρμογή υποστηρίζει την αναζήτηση πεδίου. Πατώντας το custom field search, ο χρήστης μπορεί να επιλέξει σε ποιο πεδίο θα γίνει η αναζήτηση της λέξης κλειδιού.

☐ Title ☐ Genre ☐ Duration ☐ Directors ☒ Custom Field Search

☐ Year ☐ Rating ☐ Description ☐ Stars

Παράλληλα, μπορεί να επιλέξει να αναδιατάξει τα αποτελέσματα της αναζήτησης του με βάση κάποιο από τα 3 πεδία year, duration και

rating. Η default διάταξη είναι σε αύξουσα σειρά, αλλά ο χρήστης έχει την επιλογή να το αλλάξει αυτό, πατώντας το Descending.

