Cineflix-Movie Recommendation System

Project Report Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Engineering
in
Computer Science & Engineering

Submitted by

Anjali Chauhan: (Roll No. 19UCSE4022)

Priyanshi Garg: (Roll No. 19UCSE4034)

Under the Mentorship of Mr. Anil Gupta, Professor **&** ***Under the Guidance of*** Dr. Shrawan Ram, Assistant Professor

Department of Computer Science & Engineering
MBM University, Jodhpur

July, 2022

This page was intentionally left blank.

Department of Computer Science & Engineering

M.B.M. University,
Ratanada, Jodhpur, Rajasthan, India –342011

CERTIFICATE

This is to certify that the work contained in this report entitled **"Cineflix-Movie Recommendation System**" is submitted by the group members Ms. Anjali Chauhan (Roll. No: 19UCSE4022) and Ms. Priyanshi Garg, (Roll. No: 19UCSE4034) to the Department of Computer Science & Engineering, M.B.M. University, Jodhpur, for the partial fulfilment of the requirements for the degree of **Bachelor of Engineering** in **Computer Science & Engineering**.

They have carried out their work under my guidance. This work has not been submitted elsewhere for the award of any other degree or diploma.

The project work in our opinion, has reached the standard fulfilling the requirements for the degree of Bachelor of Engineering in Computer Science in accordance with the regulations of the Institute.

Dr. Shrawan Ram
Associate professor
(Guide)
Dept. of Computer Science & Engg.
M.B.M. University, Jodhpur

Mr. Anil Gupta
Professor
(Mentor)
Dept. of Computer Science & Engg.
M.B.M. University, Jodhpur

This page was intentionally left blank.

DECLARATION

We, ***Anjali Chauhan and Priyanshi Garg,*** hereby declare that this project titled **"*Cineflix-Movie Recommendation System*"** is a record of original work done by us under the supervision and guidance of ***Dr. Shrawan Ram***.

We further certify that this work has not formed the basis for the award of the Degree/Diploma/Associateship/Fellowship or similar recognition to any candidate of any university and no part of this report is reproduced as it is from any other source without appropriate reference and permission.

SIGNATURE OF STUDENT	SIGNATURE OF STUDENT
(Anjali Chauhan)	**(Priyanshi Garg)**
8^{th} Semester, CSE	**8^{th} Semester, CSE**
Enroll. - 18R/04662	Enroll. - 18R/06201
Roll No. - 19UCSE4022	Roll No. - 19UCSE4034

This page was intentionally left blank.

ACKNOWLEDGEMENT

We take the opportunity to express our gratitude to all who have provided their immense support and their valuable time in guiding us. It is due to their support, Guidance, Supervision and Encouragement that We have successfully completed our Project Report on "***Cineflix-Movie Recommendation System***". We are highly indebted to our guide "***Dr. Shrawan Ram, Associate Professor"*** and mentor "***Mr. Anil Gupta, Professor"***, for their guidance and constant supervision as well as providing necessary information regarding this Project.

This page was intentionally left blank.

ABSTRACT

This Cineflix webapp is implemented on Flask framework. The main aim of this project is to create a movie recommendation system based on the searched movies by users. This application provides all the details of the requested movie such as overview, genre, release date, rating, runtime, top cast, reviews, recommended movies, etc. This project uses cosine similarity for recommending similar movies to searched movies. The details of the movies(title, genre, runtime, rating, poster, etc) are fetched using an API by TMDB and using the IMDB id of the movie in the API, we did web scraping to get the reviews given by the user in the IMDB site using beautifulsoup4 and performed sentiment analysis on those reviews.

This page was intentionally left blank.

Table of Contents

List of Figures

This page was intentionally left blank.

Chapter 1: Introduction

1.1 Relevance of project

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggests based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences and behaviour, and then use this information to improve their suggestions in the future.

Movies are a part and parcel of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Other way to distinguish among movies can be either by releasing year, language, director etc. Watching movies online, there are a number of movies to search in our most liked movies . Movie Recommendation Systems helps us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching our favourable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies which are exactly same or most matched with our preferences.

A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues.

1.2 Problem Statement

The goal of the project is to recommend a movie to the user. Providing related content out of relevant and irrelevant collection of items to users of online service providers.

1.2.1 Functionalities

Functionalities provided by the Movie Recommendation System are as follows:

- Provides details of the searched movies
- Recommend similar movies based on content based filtering

1.3 Motivation and Scope of the project

The objective of this project is to provide accurate movie recommendations to users. The goal of the project is to improve the quality of movie recommendation system, such as accuracy, quality and scalability of system than the pure approaches. This is done by using content based filtering approach ,To eradicate the overload of the data, recommendation system is used as information filtering tool in social networking sites .Hence, there is a huge scope of exploration in this field for improving scalability, accuracy and quality of movie recommendation systems Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues.

1.3.1 Aim

Our project aims at recommending movies based on searched movie.

- To recommend similar movies
- It satisfies the user requirement.
- Be easy to operate.
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget.

1.3.2 System Requirements

The proposed system has the following requirements

- System needs a search area.
- Display the details of the movie
- Also display the recommended movie
- It also needs a security system to prevent data.

1.4 Agile Methodology:

1. Collecting the data sets: Collecting all the required data set from Kaggle web site.in this project we require movie.csv,ratings.csv,users.csv.
2. Data Analysis: make sure that the collected data sets are correct and analysing the data in the csv files. i.e. checking whether all the column fields are present in the data sets.
3. Algorithms: in our project we have only one algorithms which is cosine similarity to build the machine learning recommendation model.
4. Training and Testing the model: once the implementation of algorithm is completed . We have to train the model to get the result.
5. Improvements in the project: In the later stage we can implement different algorithms and methods for better recommendation

Chapter 2: Related Work

Recommender systems have been a very hot research topic in recent years. Many researchers raised a lot of different recommendation approaches. The most famous category of these approaches is:

• Content-based Recommendation.
• Collaborative-filtering Recommendation.
• Hybrid Recommendation.

2.1 Content Based Recommendation

Content-based recommendation is an important approach in recommender systems. The basic idea is to recommend items that are similar with what user liked before. The core mission of content-based recommender system is to calculate the similarity between items. There are a lot of methods to model item and the most famous one is Vector Space Model. The model extracts keywords of the item and calculate the weight by TF-IDF. For example, set ki as the ith keyword of item dj , wij is the weight of ki for dj , then the content of dj can be defined as:

$$\text{Content}(dj\ ) = \{w1j\ , w2j\ , ...\}$$

As we talked before, content-based recommender system recommends items that are similar with what user liked before. So the tastes of a user can be modeled according to the history of what the user liked.

N(u) is what the user u liked before. After calculating content vector Content(.) and content preference vector ContentBasedProfile(.) of all users, given any user u and an item d, how the user like the item is defined as the similarity between

Using keywords to model item is an important step for many recommender systems. But extracting keywords of an item is also a difficult problem, especially in media field, because it is very hard to extract text keywords from a video. For solving this kind of problem, there are two main ways. One is letting experts tag the items and another one is letting users tag them. The representative of expert tagged systems are Pandora for music and Jinni for movies. Let's take Jinni as an example, the researchers of Jinni defined more than 900 tags as movie gene, and they let movie experts to make tags for them. These tags belong to different categories, including movie genre, plot, time, location and cast.

2.2 Collaborative Filtering Recommendation

Collaborative-filtering recommendation is the most famous algorithm in recommender systems. This algorithm models user's taste according to the history of user behavior. GroupLens published the first paper about collaborative filtering and the paper raised user-based collaborative filtering. In 2000, Amazon came up with item-based collaborative filtering in their paper. These two algorithms are very famous in business recommender systems.

2.2.1 User-based collaborative-filtering

In user-based collaborative filtering, it is considered that a user will like the items that are liked by users with whom have similar taste. So the first step of user-based collaborative-filtering is to find users with similar taste. In collaborative filtering, the users are considered similar when they like similar items. Simply speaking, given user u and v, N(u) and N(v) are items set liked by u and v respectively. So the similarity of u and v can be simply defined as:

$$suv = |N(u) \cap N(v)| / |N(u) \cup N(v)| \quad (2.4)$$

There are a lot of similarity algorithm, Equation 2.4 is one of them. User u's likeability for item i can be calculated by:

$$pui = X \quad v \in S(u,k) \cap N(i) \quad suvpvi \quad (2.5)$$

2.2.2 Item based collaborative filtering

Item-based collaborative-filtering is different, it assumes users will like items that are similar with items that the user liked before. So the first step of item-based collaborative-filtering is to find out items that are similar with what the user liked before. The core point of item-based collaborative-filtering is to calculate the similarity of two items. Item CF considers that items that are liked by more same users, the more similar they are.

User-based and Item-based collaborative-filtering algorithms are all neighborhood based algorithm, there are also a lot of other collaborative-filtering algorithms. Hoffman raised Latent Class Model in this paper, the model connects user and item by latent class, which considers that a user will not become interested in items directly. Instead, a user is interested in several categories that contain items, so the model will learn to create the categories according to user's behavior. On top of Latent Class Model, researchers came up with Matrix Decomposition Model, which is called Latent Factor Model as well.

There are a lot of models based on matrix decomposition and they mostly came from Netflix Prize Competition, such as RSVD, SVD++ and so on. Besides Matrix Decomposition Model, Graph Model is widely applied in collaborative filtering. Baluja introduced graph model of co-view behind the recommender algorithm of YouTube in and also raised a broadcast algorithm on graph to measure how much a user like an item. This literature research how to increase serendipity of recommendation result by means of the analysis of the path between nodes in the graph. Mirza systematically studied recommendation problems based on graph model and point out the essence of the recommendation is to connect user and item. The graph is the natural method for that studies similarity algorithms between the nodes of the graph and compares the recommendation precision of different algorithms.

2.3 Hybrid Recommender Systems

Hybrid Recommender System is more and more popular currently. Combining collaborative filtering and content-based filtering can be more effective by recently research. There are many ways to implement hybrid recommender systems: simply combine the result of CF and CB recommendations, add CF capability to a CB method.

There are seven hybridization methods:

• Weighted: Add scores from different recommender components.

• Switching: Choose methods by switching in different recommender components.

• Mixed: Show recommendation result from different systems.

• Features Combination: Extract features from different sources and combine them as a single input.

• Feature Augmentation: Calculate features by one recommender and put the result to the next step.

• Cascade: Generate a rough result by a recommender technique and recommend on the top of the previous result.

• Meta-level: Use the model generated by one recommender as the input of another recommender technique.

2.4 Comparison

Each approach has its advantage and disadvantage, and the effects are different as well for different dataset. The approach may not suitable for all kinds of problems because of the algorithm itself. For example, it is hard to apply automate feature extraction to media data by content-based filtering method. And the recommendation result only limits to items the user ever chose, which means the diversity is not so good. It is very hard to recommend for users who never choose anything. Collaborative filtering method overcomes the disadvantage of mentioned before somehow. But CF based on big amount of history data, so there are problems of sparsity and cold start. In terms of cold start, as collaborative filtering is based on the similarity between the items chosen by users, there are not only new user problem, but also new item problem, which means it is hard to be recommended if the new item has never been recommended before.

Chapter 3: Tools and Technologies

In this Movie Recommendation System, we have used the Flask framework to develop this project.

3.1 Flask

Flask (source code) is a Python web framework built with a small core and easy-to-extend philosophy. Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

Flask is pretty impressive too with its:

- built-in development server and fast debugger
- integrated support for unit testing
- RESTful request dispatching
- Jinja2 templating
- support for secure cookies (client side sessions)
- WSGI 1.0 compliant
- Unicode based

3.2 TMDB API

The API service is for those of you interested in using our movie, TV show or actor images and/or data in your application. Our API is a system we provide for you and your team to programmatically fetch and use our data and/or images.

3.3 Cosine Similarity

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

Example :

Consider an example to find the similarity between two vectors – ‘x’ and ‘y’, using Cosine Similarity.

The ‘x’ vector has values, x = { 3, 2, 0, 5 }

The ‘y’ vector has values, y = { 1, 0, 0, 0 }

The formula for calculating the cosine similarity is : Cos(x, y) = x . y / ||x|| * ||y||

x.y = 3*1 + 2*0 + 0*0 + 5*0 = 3

||x|| = sqrt(3^2 + 2^2 + 0^2 + 5^2) = 6.16

||y|| = sqrt (1^2 + 0^2 + 0^2 + 0^2) = 1

cos(x,y) = 3/6.16 = 0.49

The cosine similarity between two vectors is measured in ‘θ’.

- If θ = 0°, the ‘x’ and ‘y’ vectors overlap, thus proving they are similar.
- If θ = 90°, the ‘x’ and ‘y’ vectors are dissimilar.

Advantages :

- The cosine similarity is beneficial because even if the two similar data objects are far apart by the Euclidean distance because of the size, they could still have a smaller angle between them. Smaller the angle, higher the similarity.
- When plotted on a multi-dimensional space, the cosine similarity captures the orientation (the angle) of the data objects and not the magnitude.

3.4 Project Architecture

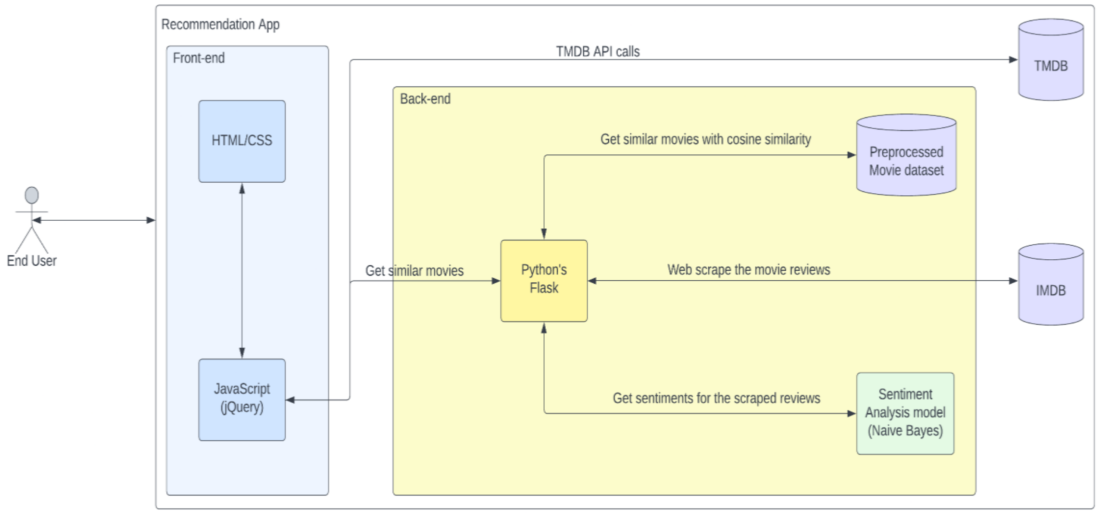

Fig 3.1 Project Architecture

3.5 Data processing

The first step to build a movie recommendation system is getting the appropriate data. This would be a file titled “movie_dataset.csv”.

Our CSV file contains a many movies and columns: index, budget, genres, homepage, id, keywords, original_language, original_title, overview, popularity, production_companies, production_countries, release_date, revenue, runtime, spoken_languages, status, tagline, title, vote_average, vote_count, cast, crew and director. Among all these different features, the ones we are interested in to find the similarity for making the next recommendation are keywords, cast, genres & director.

A user who likes a horror movie will most probably like another horror movie. Some users may like seeing their favorite actors in the cast of the movie. Others may love movies directed by a particular person. Combining all of these aspects, our shortlisted 4 features are sufficient to train our recommendation algorithm.

- First things first, let’s import the libraries we need, as well as the CSV file of the movies’ dataset.

```
import pandas as pd

import numpy as np

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.metrics.pairwise import cosine_similarity

df = pd.read_csv(r"...\movie_dataset.csv")
```

- Next, we will define a function called combined_features. The function will combine all our useful features (keywords, cast, genres & director) from their respective rows, and return a row with all the combined features in a single string.

```
movie['comb'] = movie['actor_1_name'] + ' ' + movie['actor_2_name'] + ' '+ movie['actor_3_name'] + ' '+ movie['director_name'] +' ' + movie['genres']
```

Python

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb
0	John Lasseter	Tom Hanks	Tim Allen	Don Rickles	Animation Comedy Family	toy story	Tom Hanks Tim Allen Don Rickles John Lasseter Animation Comedy Family
1	Joe Johnston	Robin Williams	Jonathan Hyde	Kirsten Dunst	Adventure Fantasy Family	jumanji	Robin Williams Jonathan Hyde Kirsten Dunst Joe Johnston Adventure Fanta...
2	Howard Deutch	Walter Matthau	Jack Lemmon	Ann-Margret	Romance Comedy	grumpier old men	Walter Matthau Jack Lemmon Ann-Margret Howard Deutch Romance Comedy
3	Forest Whitaker	Whitney Houston	Angela Bassett	Loretta Devine	Comedy Drama Romance	waiting to exhale	Whitney Houston Angela Bassett Loretta Devine Forest Whitaker Comedy Dr...
4	Charles Shyer	Steve Martin	Diane Keaton	Martin Short	Comedy	father of the bride part ii	Steve Martin Diane Keaton Martin Short Charles Shyer Comedy
...	...	...	...	...	...	...	...
45438	Ben Rock	Monty Bane	Lucy Butler	David Grammer	Horror	the burkittsville 7	Monty Bane Lucy Butler David Grammer Ben Rock Horror
45439	Aaron Osborne	Lisa Boyle	Kena Land	Zaneta Polard	Sci-Fi	caged heat 3000	Lisa Boyle Kena Land Zaneta Polard Aaron Osborne Sci-Fi
45440	John Irvin	Patrick Bergin	Uma Thurman	David Morrissey	Drama Action Romance	robin hood	Patrick Bergin Uma Thurman David Morrissey John Irvin Drama Action Romance
45441	Lav Diaz	Angel Aquino	Perry Dizon	Hazel Orencio	Drama	century of birthing	Angel Aquino Perry Dizon Hazel Orencio Lav Diaz Drama
45442	Mark L. Lester	Erika Eleniak	Adam Baldwin	Julie du Page	Action Drama Thriller	betrayal	Erika Eleniak Adam Baldwin Julie du Page Mark L. Lester Action Drama Th...

39201 rows × 7 columns

Fig 3.2 Combined dataset

- The sklearn.feature_extraction module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image. We will use CountVectorizer's fit_transform to count the number of texts and we will print the transformed matrix count_matrix into an array for better understanding

```
cv = CountVectorizer()

count_matrix = cv.fit_transform(data['comb'])
```

- Then we will use the Cosine Similarity from Sklearn, as the metric to compute the similarity between two movies.

```
similarity = cosine_similarity(count_matrix)
```

Chapter 4: Layout

4.1 Home Page

First Page of the Movie Recommendation System.

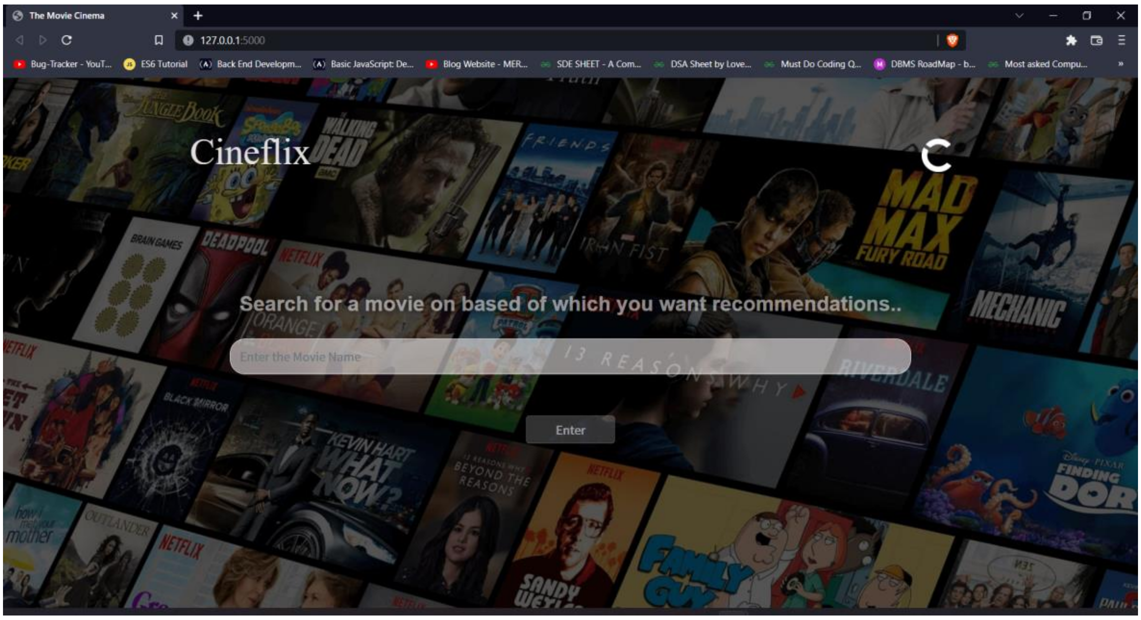

Fig 4.1.1 Home Page

You can search for a movie. Also here we implemented a functionality in which it will autocomplete the movies name/ give suggestions based on the initial words searched.

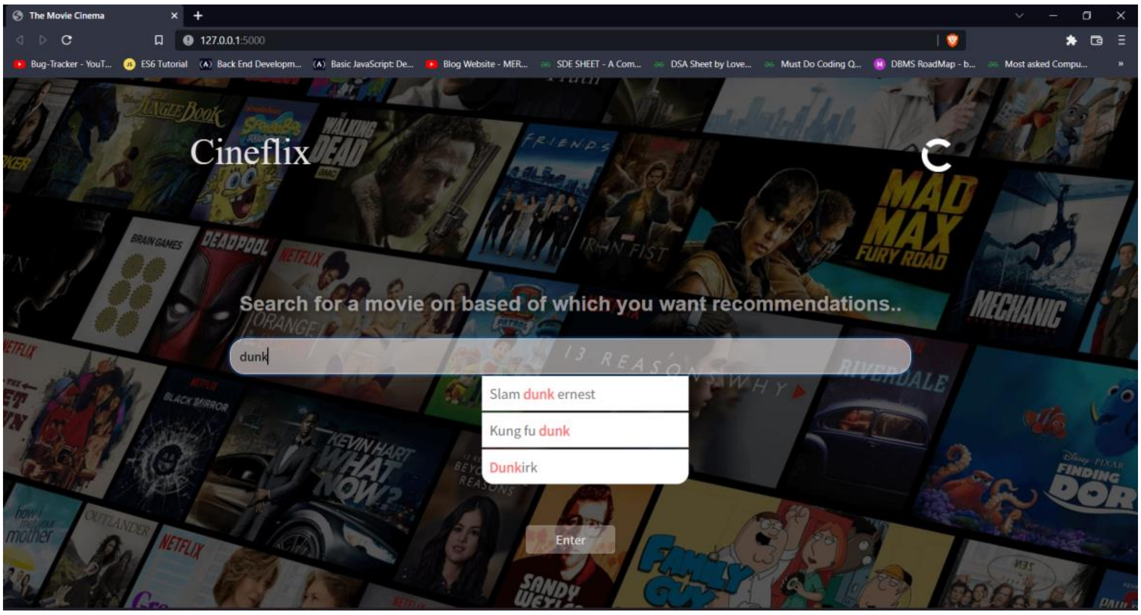

Fig 4.1.2 Autocomplete

4.2 Search Result

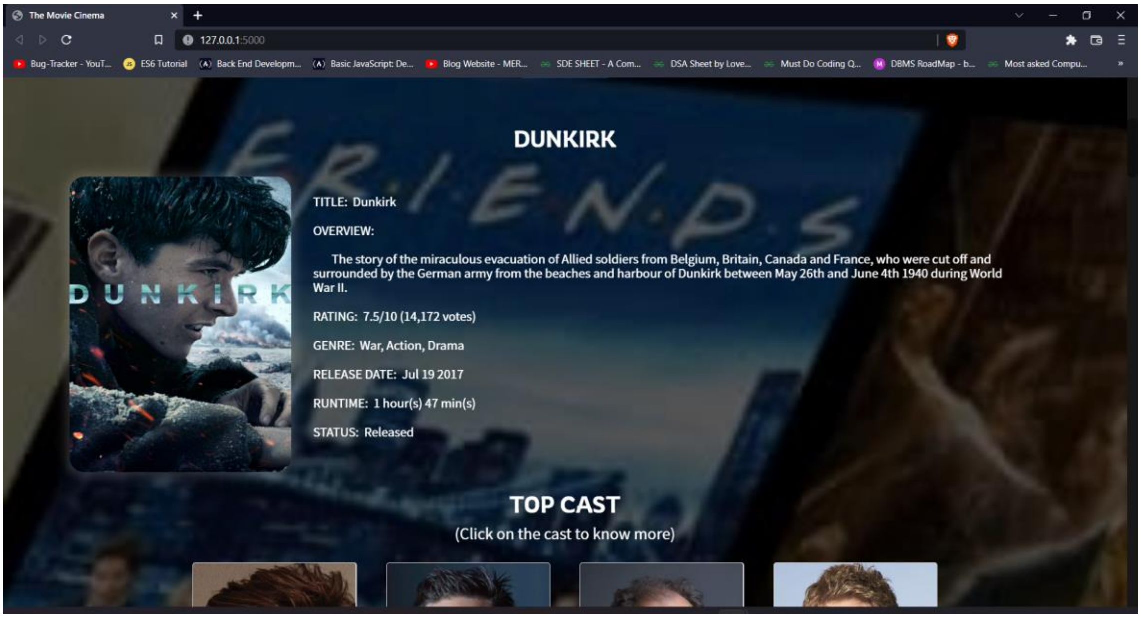

Fig 4.2.1: Details of Movie

So here the details about the searched movie is displayed. Like its casts, plot, rating, director, genre etc.

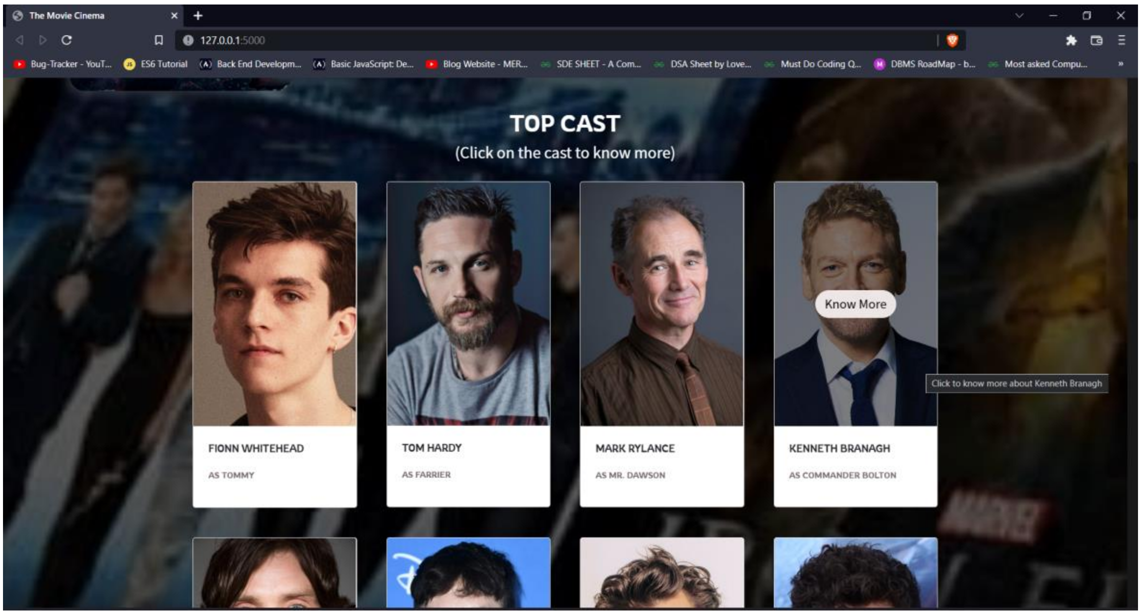

Fig 4.2.2: Top Casts

If you click on any cast card, details of that actor/actress is displayed.

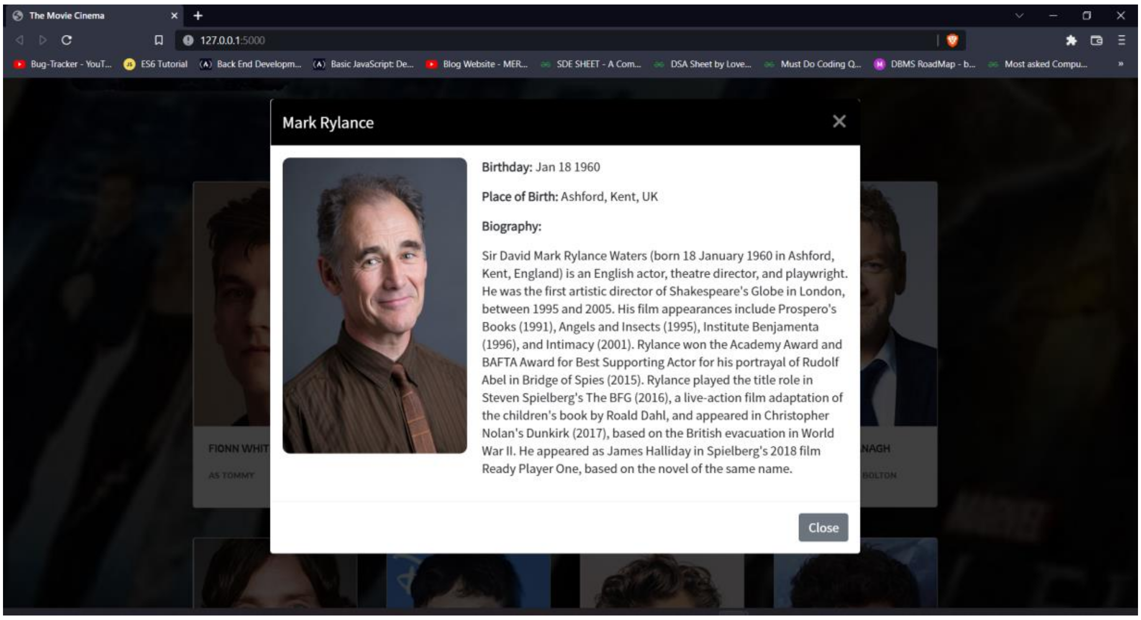

Fig 4.2.3: Details of Casts

Also fetched User reviews from IMDB site using web scraping.

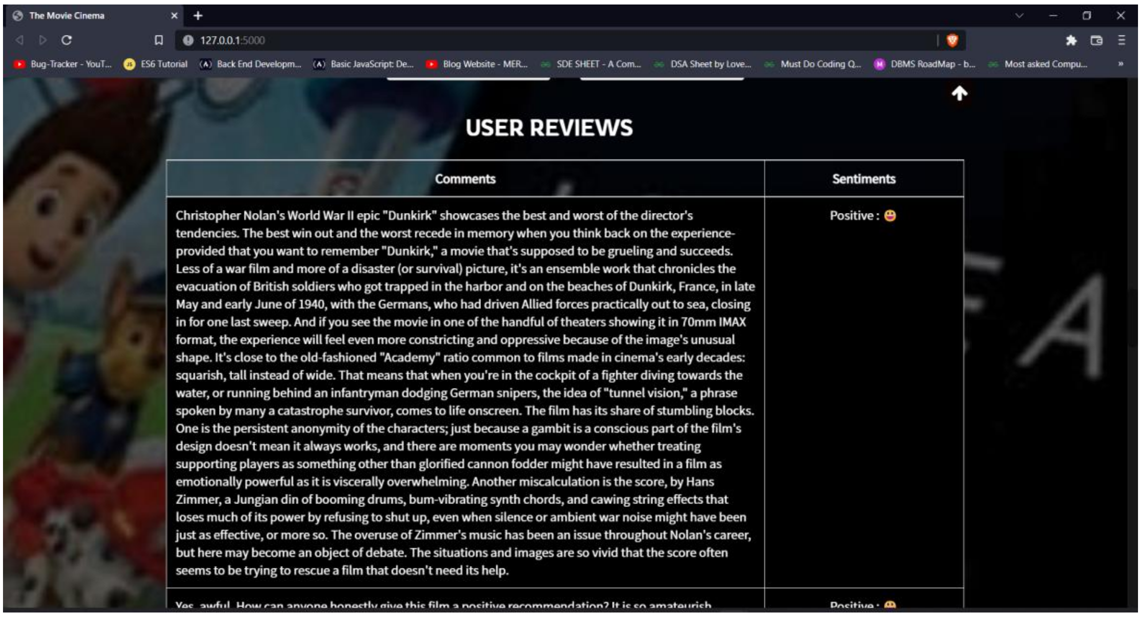

Fig 4.2.4: User Reviews

And finally based on the searched movie this application will recommend similar movies as that of searched movie.

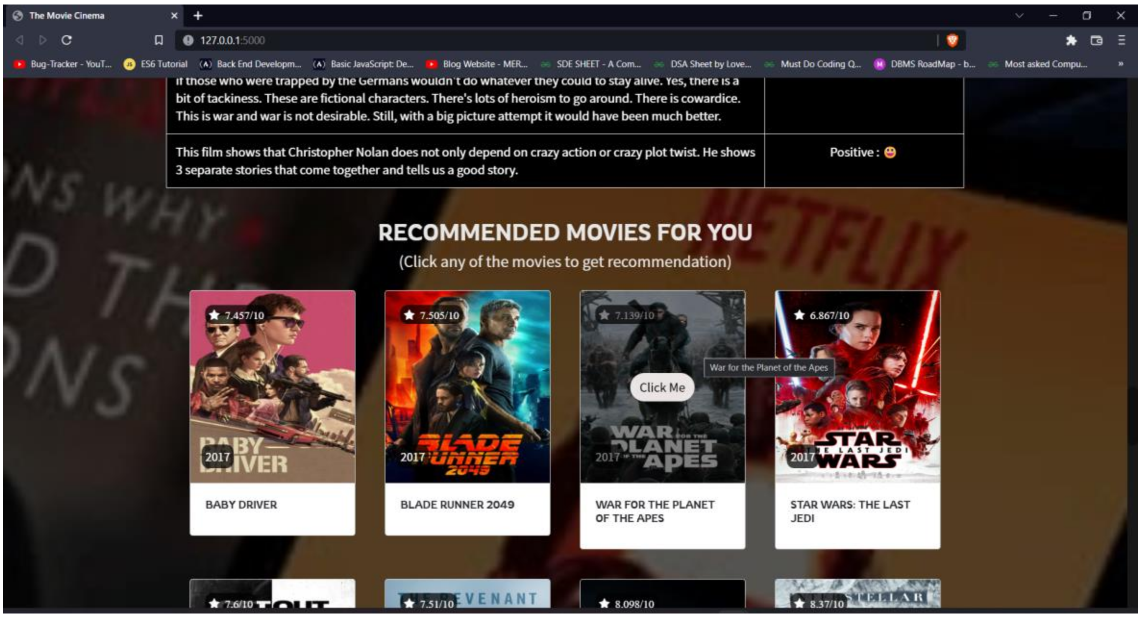

Fig 4.2.5: Recommendations

Chapter 5: Conclusion and Future Work

5.1 Conclusion

Recommender system has become more and more important because of the information overload. For content-based recommender system specifically, we attempt to find a new way to improve the accuracy of the representative of the movie.

For the problems we mentioned at beginning, firstly, we use content-based recommender algorithm which means there is no cold start problem. Some of them are from other research team in the company, so the features are diversity and more accurate than others. Then we introduced the cosine similarity which is commonly used in industry. For the weight of features, we introduced TF-IIDF-DC which improve the representative of the movie.

On searching for any movie name, the site displays the details of the movie, the details of cast, and viewer's review and similar movie recommendations. For viewer's reviews, web scraping using beautifulsoup4 and prediction is also done if the review is good or bad.

5.2 Future Work

The following functionalities can be added in future in this project:

- The movie dataset in this project can be changed into more diverse ones.
- Login and sign up functionalities can be added for the users
- Personalised recommendations for every user
- Analysis for a user's taste like genres they like, favourite actors etc.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. Modern information retrieval, volume 463. ACM press New York, 1999.

[3] Shumeet Baluja, Rohan Seth, D Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In Proceedings of the 17th international conference on World Wide Web, pages 895–904. ACM, 2008.

[4] Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 95–104. ACM, 2007.

[5] Suvir Bhargav. Efficient features for movie recommendation systems. 2014.

[6] Robin Burke. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4):331–370, 2002.

[7] Robin Burke. Hybrid web recommender systems. In The adaptive web, pages 377–408. Springer, 2007.

[8] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems, pages 293–296. ACM, 2010.

[9] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. JAsIs, 41(6):391–407, 1990.

[10] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with ap39

BIBLIOGRAPHY plication to collaborative recommendation. Knowledge anddata engineering, ieee transactions on, 19(3):355–369, 2007.

[11] Xu Hailing, Wu xiao, Li Xiaodong, and Yan Baoping. Comparison study of internet recommendation system. Journal of Software, 20(2):350–362, 2009.

[12] Anne Håkansson. Portal of research methods and methodologies for research projects and degree projects. In The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP 2013; Las Vegas, Nevada, USA, 22-25 July, pages 67–73. CSREA Press USA, 2013.

[13] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In IJCAI, volume 99, pages 688–693, 1999.

[14] Anna Huang. Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, pages 49–56, 2008.

[15] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. Recommender systems: an introduction. Cambridge University Press, 2010.

[16] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In Knowledge Discovery in Databases: PKDD 2007, pages 506–514. Springer, 2007.

[17] Joseph A Konstan. Introduction to recommender systems: Algorithms and evaluation. ACM Transactions on Information Systems (TOIS), 22(1):1–4, 2004.

[18] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434. ACM, 2008.

[19] Daniel Z Lieberman, Suena H Massey, Vilmaris Quiñones Cardona, and Kenneth P Williams. Predicting content preference: Applying lessons learned from the commercial web to therapeutic software. Cyberpsychology, 2(2), 2008.

[20]Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations:Item-to-item collaborative filtering. Internet Computing, IEEE, 7(1):76–80, 2003.

[21] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Recommender systems handbook, pages 73–105. Springer, 2011.

[22] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. Introduction to information retrieval, volume 1. Cambridge university press Cambridge,2008.40

[23] Christopher D Manning and Hinrich Schütze. Foundations of statistical natural language processing. MIT press, 1999.

[24] Bradley N Miller, Istvan Albert, Shyong K Lam, Joseph A Konstan, and John Riedl. Movielens unplugged: experiences with an occasionally connected recommender system. In Proceedings of the 8th international conference on Intelligent user interfaces, pages 263–266. ACM, 2003.