**MOVIE RECOMMENDATION SYSTEM**

**Project report in partial fulfillment of the requirement for the award of the degree of**

**Bachelor of Technology**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

Debanjan Bhattacharya University Roll No. 12018009019006

Soumyadeep Maji University Roll No. 12018009019307

Shouryadeep Mahato University Roll No. 12018009019576

Aniket Roy University Roll No. 12018009019188

Arkadyuti Dam University Roll No. 12018009019654

Soumyajit Roy University Roll No. 12018009019386

**Under the guidance of**

Prof. Varsha Poddar

&

Prof. Sukalyan Goswami

Department of Computer Science



UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

**CERTIFICATE**

This is to certify that the project titled **Movie Recommendation System** submitted by **Debanjan Bhattacharya (University Roll No.** 12018009019006**), Soumyadeep Maji (University Roll No.** 12018009019307**)**, **Shouryadeep Mahato (University Roll No.** 12018009019576**), Aniket Roy (University Roll No.** 12018009019188**)**, **Arkadyuti Dam (University Roll No.** 12018009019654**) and Soumyajit Roy(University Roll No.** 12018009019386 **)** students of **UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA**, in partial fulfillment of requirement for the degree of Bachelor of Computer Science Engineering, is a bonafide work carried out by them under the supervision and guidance of **Prof. Varsha Poddar & Prof. Sukalyan Goswami** during 7th Semester of academic session of 2018-2022. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Varsha Poddar Prof. Sukalyan Goswami

Guide Details: XXX (Name & Designation) Guide Details: XXX (Name & Designation)

Department of Computer Science Department of Computer Science

UEM, Kolkata UEM, Kolkata

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Sukalyan Goswami

HOD, Department of Computer Science

UEM, Kolkata

**ACKNOWLEDGEMENT**

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide Prof. Varsha Poddar and Prof. Sukalyan Goswami of the Department of Computer Science, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Sukalyan Goswami, HOD, Department of Computer Science, UEM, Kolkata and all other departmental faculties for their ever-present assistant and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

**Debanjan Bhattacharya**

**Soumyadeep Maji**

**Shouryadeep Mahato**

**Aniket Roy**

**Arkadyuti Dam**

**Soumyajit Roy**

***TABLE OF CONTENTS***

**ABSTRACT........................................................................01**

**CHAPTER – 1: INTRODUCTION...............................................02-03**

**CHAPTER – 3: PROBLEM STATEMENT**

***3.1. Project Title.................................04***

***3.2. Project Requirements.................................04***

**CHAPTER – 4: PROPOSED SOLUTION**

***4.1. Types Of Recommenders......................05***

***4.2. About The Dataset........................05***

***4.3. Simple Recommender........................05***

***4.4. Content Base Recommender.................06***

***4.5. Final Result....................07-08***

**CHAPTER – 5: CONCLUSION & FUTURE SCOPE**

***6.1. Conclusion...............................09***

***6.2. Future Scope .............................09***

**BIBLIOGRAPHY ........................................................................10**

***ABSTRACT***

*The purpose of this project is to build a* ***Movie Recommendation System*** *that suggests users various movies and similar contents depending on the type of recommendation system in play.*

*To define a recommendation system, it is a kind of system that provides suggestions to users for certain resources like books, movies, songs, etc., based on some particular data set. Movie recommendation systems usually predict what movies a user will like based on the attributes present in previously liked movies. Such recommendation systems are beneficial for organizations that collect data from large amounts of customers, and wish to effectively provide the best suggestions possible. A lot of factors can be considered while designing a movie recommendation system like the genre of the movie, actors present in it or even the director of the movie. The systems can recommend movies based on one or a combination of two or more attributes.*

*In this paper, the recommendation system has been built on two types of recommenders: (i) Simple Recommender and (ii) Content-based Recommender. Two different approaches are used to achieve the results for both types of recommendation system. The dataset used for the system is downloaded from a free-source database in .csv format and is named as* ***“tmdb\_5000\_movies.csv”*** *in our project.*

***INTRODUCTION***

1. **What is a recommendation system?**

*A recommender system, or a recommendation system, is a subclass of the information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.*

*Recommender systems are used in a variety of areas, with commonly recognized examples taking the form of playlist generators for video and music services, product recommenders for online stores, or content recommenders for social media platforms and open web content recommenders. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. Recommender systems have also been developed to explore research articles and experts, collaborators, and financial services.*

1. **What is a movie recommendation system?**

*A movie recommendation system is a domain-specific recommendation program that seeks to predict the “ratings” or “preferences” of a user towards a particular movie or a set of movies. Therefore, the main focus of our recommendation system is to filter and predict only those movies which a user would prefer given some data about the user him or herself.*

1. **Types of movie recommender systems being used.**
2. ***Simple Recommender*** *– The Simple Recommender offers generalized recommendations to every user based on movie popularity and (sometimes) genre. The basic idea behind this recommender is that movies that are more popular and more critically acclaimed will have a higher probability of being liked by the average audience. We will sort our movies based on ratings and popularity and display the top movies of our list. As an added step, we can pass in a genre argument to get the top movies of a particular genre.*

**Methodology for Simple Recommender System: -**

*We will use the TMDB Ratings to come up with our Top Movies Chart. We will use IMDB's weighted rating formula to construct our chart. Mathematically, it is represented as follows:*

**Weighted Rating (WR) = (𝑣𝑣 + 𝑚. 𝑅) + (𝑚𝑣 + 𝑚. 𝐶)(vv + m.R) + (mv + m.C),**

*where,*

* **v***is the number of votes for the movie*
* ***m****is the minimum votes required to be listed in the chart*
* ***R****is the average rating of the movie*
* ***C****is the mean vote across the whole report*

*The next step is to determine an appropriate value for m, the minimum votes required to be listed in the chart. We will use****95th percentile****as our cutoff. In other words, for a movie to feature in the charts, it must have more votes than at least 95% of the movies in the list.*

1. ***Content Based Recommender*** *– The recommender we built in the previous section suffers some severe limitations. For one, it gives the same recommendation to everyone, regardless of the user's personal taste. If a person who loves romantic movies (and hates action) were to look at our Top 15 Chart, she/he wouldn't probably like most of the movies. If she/he were to go one step further and look at our charts by genre, she/he wouldn't still be getting the best recommendations.*

*For instance, consider a person who loves Dilwale Dulhania Le Jayenge, My Name is Khan and Kabhi Khushi Kabhi Gham. One inference we can obtain is that the person loves the actor Shahrukh Khan and the director Karan Johar. Even if she/he were to access the romance chart, she/he wouldn't find these as the top recommendations.*

*To personalize our recommendations more, we are going to build an engine that computes similarity between movies based on certain metrics and suggests movies that are most similar to a particular movie that a user liked. Since we will be using movie metadata (or content) to build this engine, this also known as Content Based Filtering.*

*We will build two types of Content Based Recommenders based on:*

* *Movie Overviews and Taglines*
* *Movie Cast, Crew, Keywords and Genre*

***Methodology for Content Based Recommender:***

***Cosine Similarity*** *– We will be using the Cosine Similarity to calculate a numeric quantity that denotes the similarity between two movies. Mathematically, it is defined as follows:*

𝑐𝑜𝑠𝑖𝑛𝑒(𝑥,𝑦) = 𝑥.𝑦⊺/|𝑥||.||𝑦||

*Since we have used the TF-IDF Vectorizer, calculating the Dot Product will directly give us the Cosine Similarity Score. Therefore, we will use sklearn's linear kernel instead of cosine similarities since it is much faster.*

***PROBLEM STATEMENT***

***Project Title -*** *To build a movie recommendation system that filters movies based on specific types of data taken from the user. In our project, we have priorities specific data fields like ratings, votes, movie descriptions, genre, overviews and taglines, movie cast, crew, and certain keywords to extract accurate information and then display the movies accordingly.*

***Project Requirements -*** *The main language to be used in this project is Python and the concept of data processing is being used to get the desired result in the form of movie recommendations.*

***PROPOSED SOLUTION***

1. ***Types of recommenders:***
2. ***Simple Recommender***
3. ***Content-based Recommender***
4. ***About the dataset:***

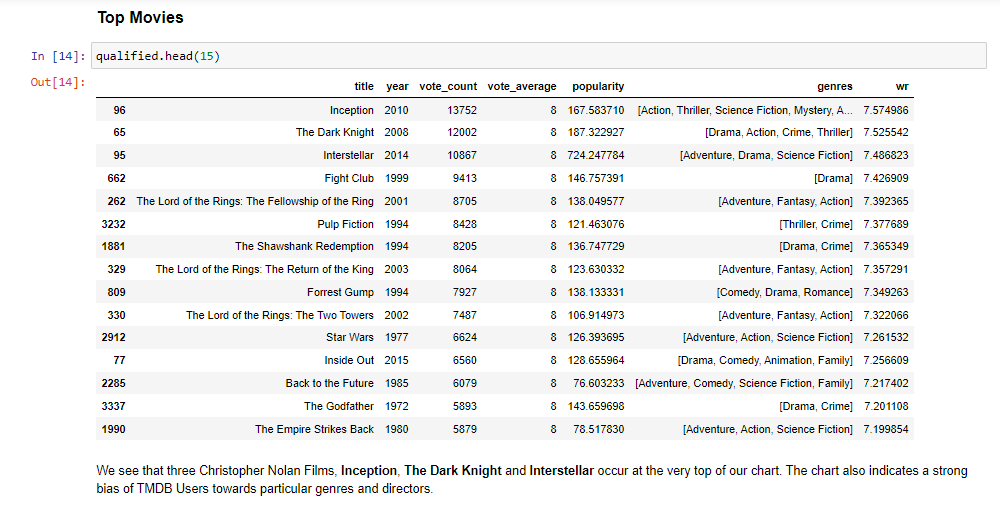
The dataset consists of movies released on or before July 2017. Data points include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts and vote averages.

This dataset also has files containing 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5 and have been obtained from the official GroupLens website.

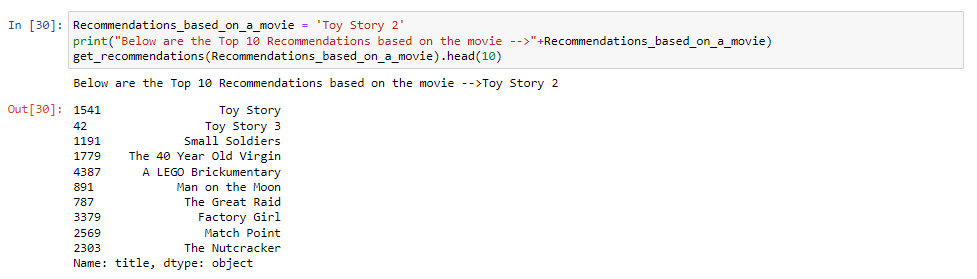
* **The Full Dataset**: Consists of 26,000,000 ratings and 750,000 tag applications applied to 45,000 movies by 270,000 users. Includes tag genome data with 12 million relevance scores across 1,100 tags.
* **The Small Dataset:** Comprises of 100,000 ratings and 1,300 tag applications applied to 9,000 movies by 700 users.

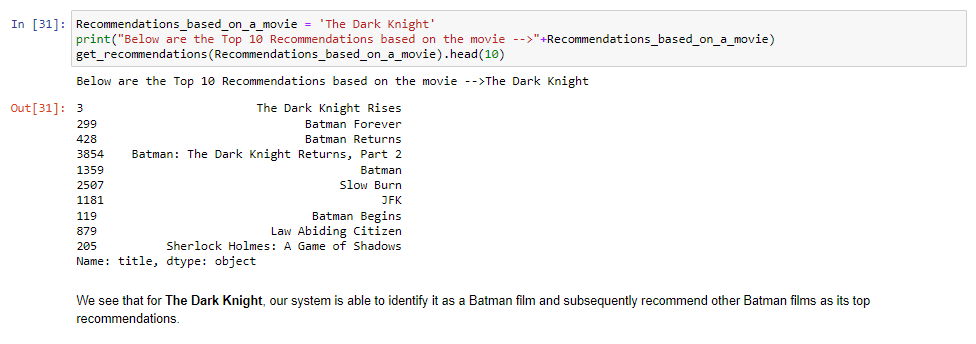
We will build our Simple Recommender using movies from the Full Dataset whereas all personalized recommender systems will make use of the small dataset.

1. ***Simple Recommender:***



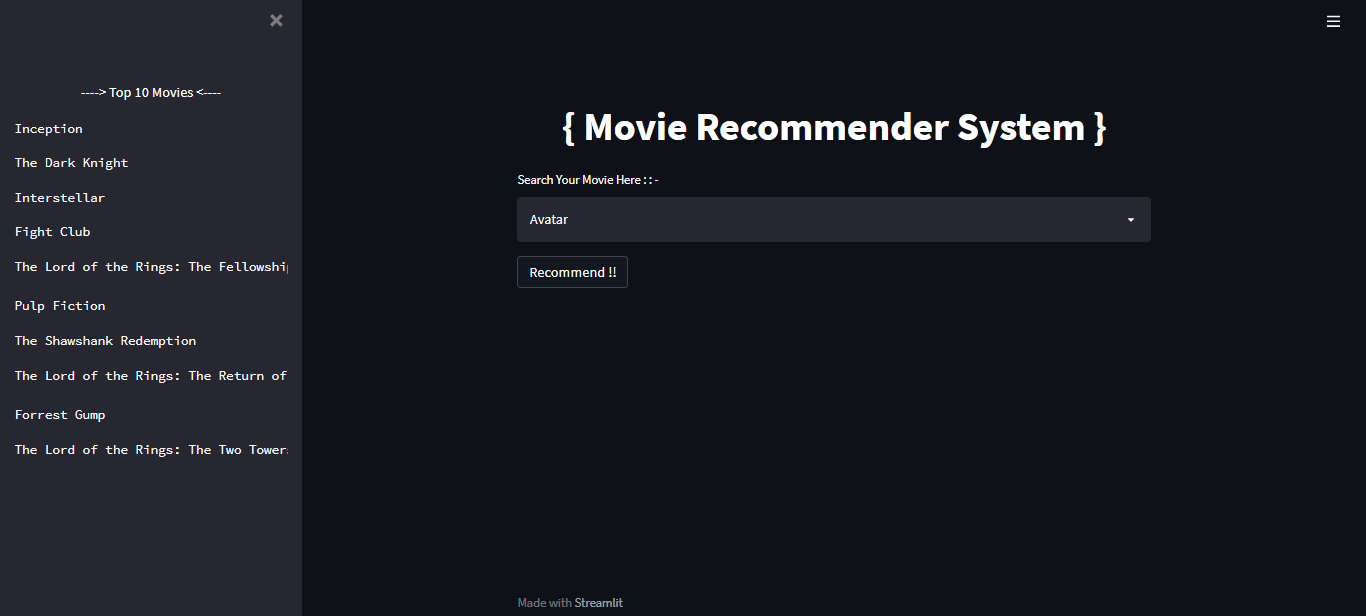
1. ***Content-based Recommender:***



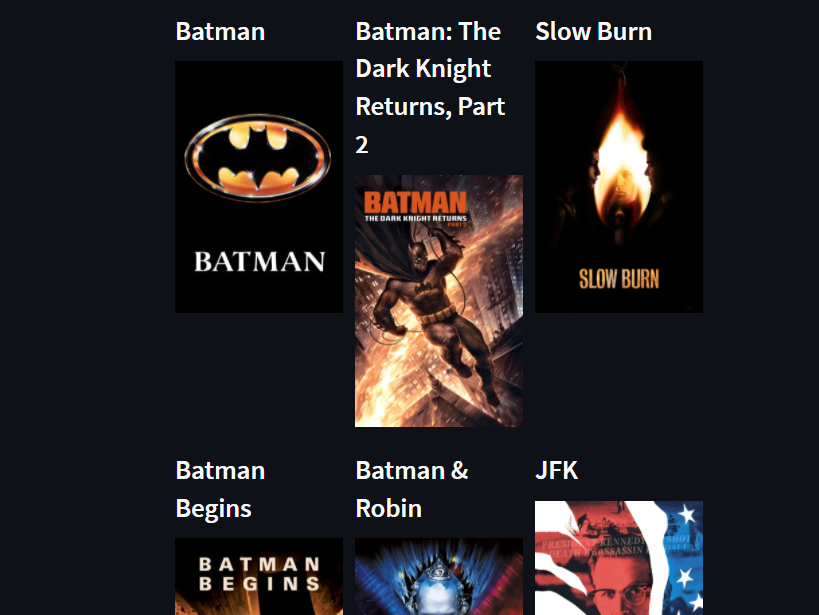
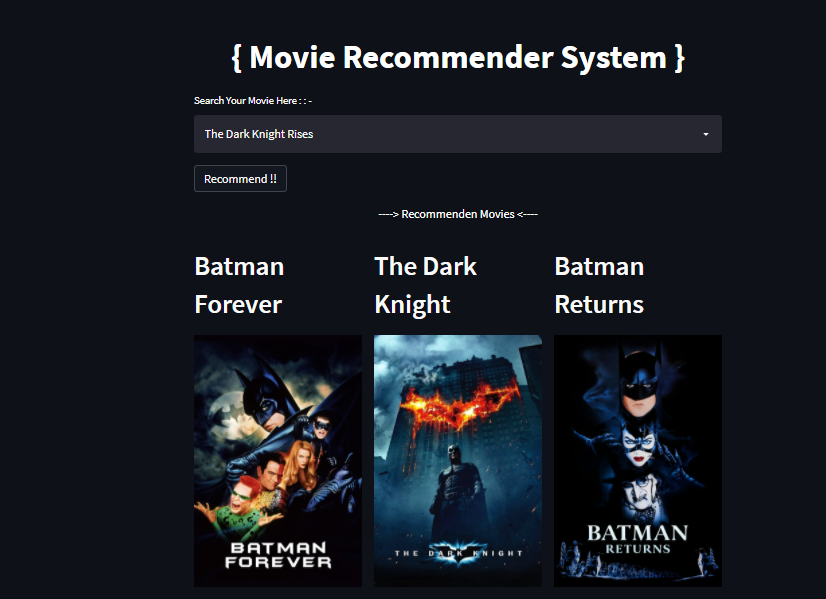


***FINAL RESULT***

1. ***Simple Recommender displaying the top 15 movies based on overall user ratings.***



1. ***Content-based Recommender showing output of the movies similar to the required one based on the similarity score used in the engine.***



***CONCLUSION & FUTURE SCOPE***

***Conclusion:***

In this project, we have built 2 different types of recommendation engines based on different ideas and algorithms. They can be summarized as follows:

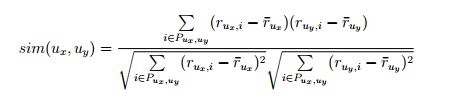
1. **Simple Recommender**: This system used overall TMDB Vote Count and Vote Averages to build Top Movies Charts, in general and for a specific genre. The IMDB Weighted Rating System was used to calculate ratings on which the sorting was finally performed.
2. **Content Based Recommender**: We built a content-based engines; one that took movie overview and taglines as input and the other which took metadata such as cast, crew, genre and keywords to come up with predictions. We also devised a simple filter to give greater preference to movies with more votes and higher ratings.

***Future Scope:***

Cosine similarity calculation do not work well when we don't have enough rating for movie or when user's rating for some movie is exceptionally either high or low. As an improvement on this project some other methods such as adjusted cosine similarity can be used to compute similarity.

Adjusted cosine similarity, which is similar to cosine similarity, is measured by normalizing the user vectors Ux and Uy and computing the cosine of the angle between them. However, unlike cosine similarity, when computing the dot product of the two user vectors, adjusted cosine similarity uses the deviation between each of the user’s item ratings, denoted Ru, and their average item rating, denoted ¯Ru, in place of the user’s raw item rating.

 In equation form, the adjusted cosine similarity computation is expressed as:

[](https://sites.google.com/site/chatrasadagopansharma/future-scope/Capture07.JPG?attredirects=0)

where Pux,uy represents the subset of items i ∈ I for which both users have rated, Rux,i is the rating of user Ux on item i and Ruy,i in the rating of user Uy on item i. The main advantage of this approach is that in item-based collaborative filtering, the item vectors consist of ratings from different users who often have varying rating scales.

***BIBLIOGRAPHY***

1. ***Kaggle.com –*** *Obtained the dataset for your project*
2. ***Streamlit.io –*** *Learning and implementation of Streamlit concept in Python and using the syntaxes to get the desired results.*
3. ***Stackoverflow.com –*** *Concept on how to use HTML codes in a Python program for easier website modification.*
4. ***Ivyproschool.com –*** *Initial idea about the movie recommendation system.*
5. ***Themoviedb.com –*** *Obtaining the API key and display images for the content-based recommender.*