Movie Data Analysis and Recommendation System

1. Introduction

The advent of the digital age has meant that every customer faces multiple choices at more than one time every day. For example, customers might be hard pressed to discover a new book to read or a new movie to watch, taking into account a plethora of choices available to them. People usually select or purchase a new product based on some friend's recommendations, comparison of similar products or feedbacks from other users. Providing a useful suggestion of products to online users to increase their consumption on websites is the goal of many companies nowadays. A recommendation system is a tool that automatically provides suggestions that best suit the user's needs and that offers a personalized content based on the user's past behaviour.

This project is aimed at creating a Movie Data Analysis and Recommendation System. The system's task is to recommend movies to the user by finding people with similar interests, analyzing their behaviour, and recommending our user the same items. The system can also find movies like ones which the user selected/watched earlier and recommend movies which are like them. Recommendation systems enable quick filtering of movies to suit the user's liking.

The final project prototype is a web-based recommender system that uses myriad machine learning techniques for genre classification, community backed movie recommendation system visualized using a User Interface developed using modern technologies.

2. Problem Definition and Algorithm

2.1 Task Definition

We created a recommendation engine using the following approaches to recommend movies to the users:

1. Content-based filtering
2. Collaborative filtering

2.1.1 Data Collection

The foundational dataset used for this project will be the Movie lens dataset https://grouplens.org/datasets/movielens/ , which is a web based recommender system created by a research lab in the University of Minnesota. This dataset contains more than 1 million ratings for about 60,000 movies. This rich set of data would help us recommend movies for users to watch, based on their film preferences using collaborative filtering of members' movie ratings and movie reviews.

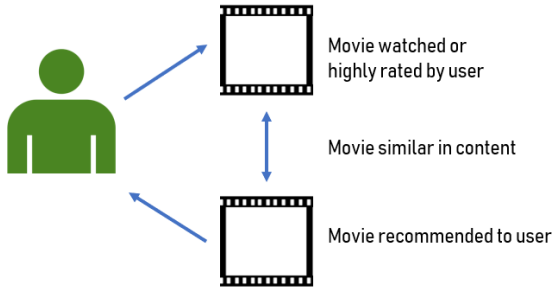
We have the following datasets:

- Movies Dataset: Each movie is uniquely defined by its Movie Id and consists of a set of associated genres.
- Ratings Dataset : It consists of user ids that have given a particular rating for a certain movie (given by Movie Id) and a timestamp associated with that rating.
- Users Dataset: The users are uniquely defined by a user id and have attributes such as gender, age, age-category, occupation and zip-code.

2.3 Algorithm

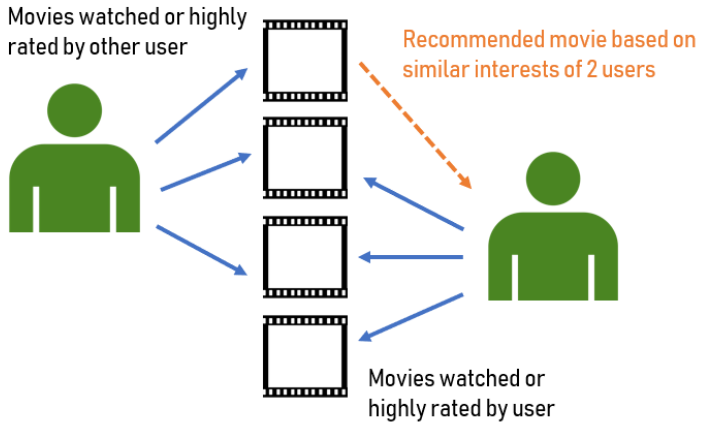
2.3.1. Content-Based Filtering

Content-based filtering is based on the similarity in attributes of the movies and a profile of the user's preferences. The user provides these preferences in the form of movie ratings. The system creates a user profile based on past and present ratings. Based on this, it recommends movies that similar to the movies rated highly or watched previously by the user. In particular, various movies are compared with movies previously or presently rated by the user and the best-matching movies are recommended. The idea behind this is that if you like an item, then you will also like a "similar" item. The user provides more ratings on the recommended movies making the system more accurate.



2.3.1. Collaborative Filtering

The Collaborative Filtering Recommender is uses the user's past behaviour (given ratings and movies watched before) and not on the content (attributes) of the movies. It is based on the similarity in preferences and choices of two users. This approach is based on the idea that other users similar to a certain user can be used to predict how much that user will like a particular movie by comparing the movies they have watched and the user has watched. This approach helps the system recommend movies that are much closer to the user's personal interest.



3. Experimental Evaluation

3.1 Methodology

3.1.1 Data Exploration:

We begin by exploring and preparing the data we have. We first load the datasets (.csv) – movies, rating and users - into our python program using pandas. We retrieve a description of the datasets to understand its structure.

When a user visits the recommendation system for the first time, he/she is shown the highest rated movies in the most popular genres. Hence, we find the most popular genres and movies with the highest ratings from the dataset first. We then implement the above-mentioned algorithms to build our recommendation engine.

3.1.2 Implementation:

In the first approach (Content-based), we will compute the similarity between movies based on movie genres. Comparing genres of different movies, the system will suggest movies that are most similar to a certain movie based on its genre.

For this we use the concept of Vector Space Model where each movie is stored as a vector of its attributes (also vectors) in an n-dimensional space and which computes the proximity (of movies) based on the angle between these vectors. To determine the similarity between movies/vectors, we calculate the angles between the vectors. We used the **TfidfVectorizer** function from scikit-learn, to transform text to feature vectors that can be used as input to estimator. We then calculated the similarity between 2 movies denoted by a numeric quantity that is calculated using the **Cosine Similarity.** This gives us a similarity matrix consisting of cosine similarity scores between different pairs of movies.

Cosine similarity is used because the value of cosine increases with decreasing value of the angle between. This signifies more similarity. Using the Similarity Matrix we can recommend the most closest movies to a certain movie.

In the second approach (Collaborative-based), we use a **Pearson Correlation matrix**. To determine the similarity of the users based on their ratings for a particular movie, we need User Ids, Ratings and Movie Ids. We can obtain these attributes from the ratings dataset. From here we created a training and test dataset. We then created two matrices consisting of user id, ratings and movie id for each dataset. To create the Pearson Correlation matrix (similarity matrix), we used pairwise_distances function of sklearn. We created a User similarity matrix and an Item similarity matrix.

Using this matrix, we are able to predict the ratings that were not included with the data. On predicting the ratings, we can verify them with the test data we created check the quality of the recommender engine.

3.1.3 Dependencies

- Pandas
- Sklearn
- Numpy
- Matplotlib
- Wordcloud
- Seaborn
- Flask-restful

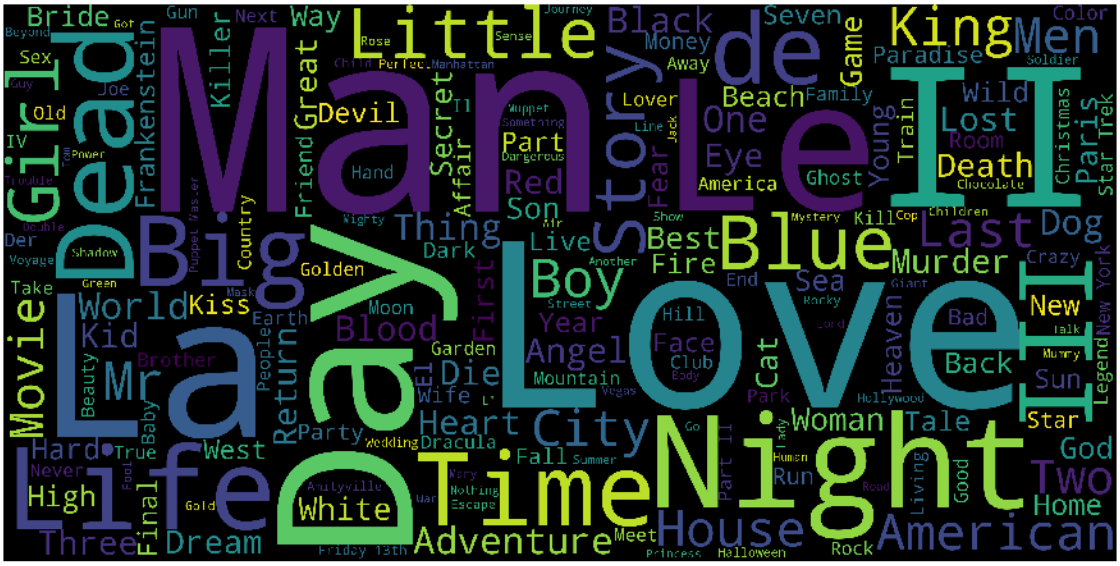
- NodeJS
- Angular CLI/Angular version 6

3.1.4 Results

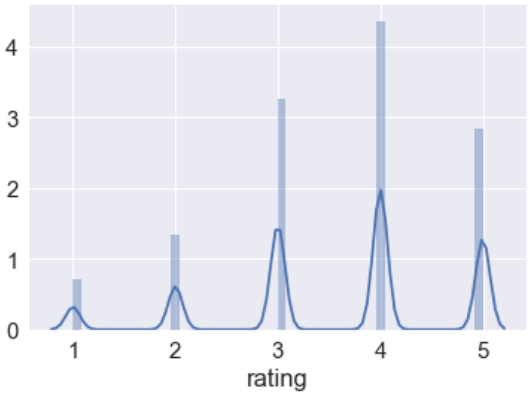
Root Mean Squared Error (RMSE) is one way to evaluate the accuracy of the predicted ratings. We therefore used the mean_square_error (MSE) function from sklearn (RMSE = square root of MSE).

From the outputs of the user-based CF RMSE and item-based CF RMSE, we found that user-based collaborative filtering gives a better result.

The following wordclouds tries to visualize both genres and the movie titles based on the frequency of their respective appearance in the data:



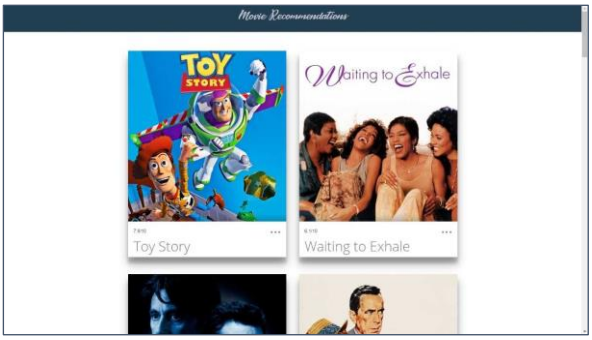
The following plot displays the distribution of the ratings:

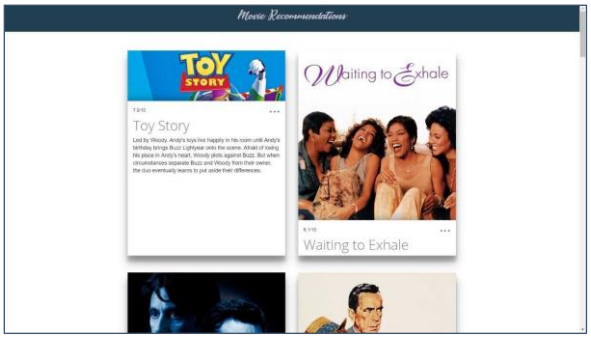


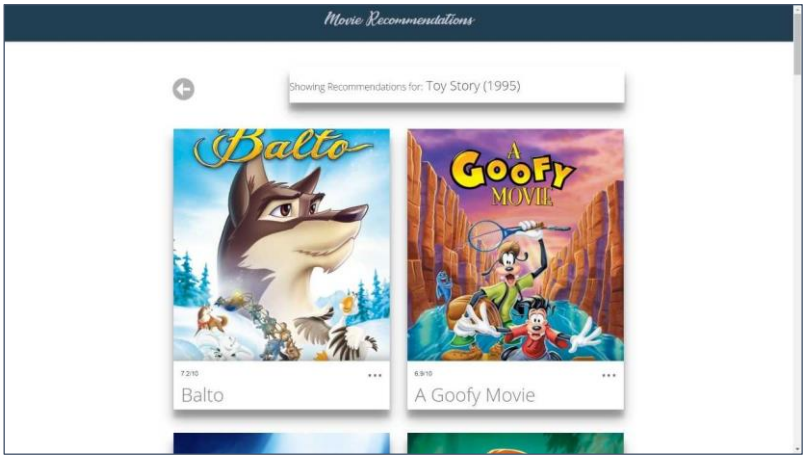
To visualize the recommendations engine's results, we developed a Rest API in python using flask to serve movie recommendations using basic HTTP verbs. We also developed a User Interface using Angular6, HTML and CSS to see the recommendation engine in action.

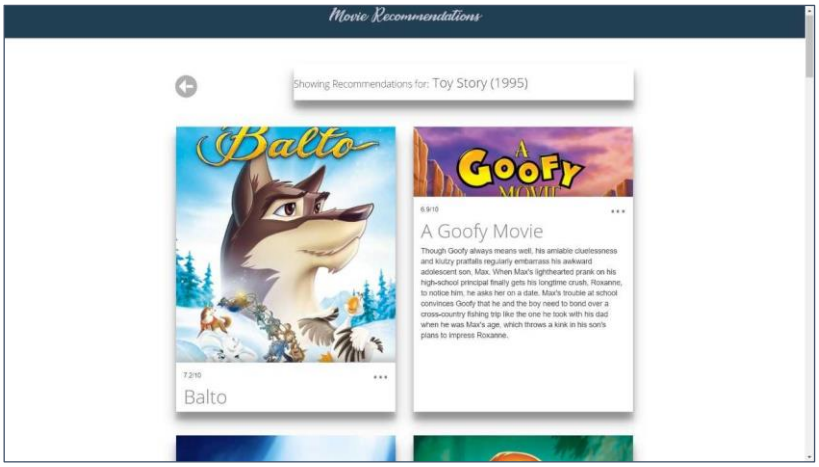
The UI features movies served from the python backend and combined it with "themoviedb" API to access movie poster, ratings and description, in the form of cue cards. Clicking on one of the movie cue cards sends the movie to the backend, runs it through the recommendation engine and serves new movie cue cards which are like the movie that was clicked.

The recommendation engine that uses Content-based filtering and Collaborative filtering to recommend movies to a user is as shown below:









4. Conclusion

We developed our recommendation engine by:

- Loading and reviewing the datasets.
- Using the content-based filtering concept based on movie genres.
- Using a memory-based collaborative filtering model based on user ratings.

We see that using Content Based Filtering method, we can recommend movies to users with unique tastes, and can recommend new & unpopular items.

5. References

1. Wikipedia contributors. (2018, November 30). Recommender system. In *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Recommender_system&oldid=871404126
2. Daniil Korbut (2017, July 6). Recommendation System Algorithms. https://blog.statsbot.co/recommendation-system-algorithms-ba67f39ac9a3

Team Members

Sushant Singh Dahiya, Meghna Mathur, Rishika Parashar