MOVIE RECOMMENDER SYSTEM

Navya Sai Reddy Gorre
Chetana Alekhya Reddy Gorre
Maheswar Reddy Peram

INTRODUCTION

Earlier, when there are no movie recommendation systems, people use to rely on the suggestions and recommendations done by other people based on their personal interests. In order to overcome this, now a days, many OTT platforms such as Netflix, Amazon Prime Video, Hulu and so on are implementing a special feature, recommendation system, in their application on considering the one's watch history and most liked movies. It is possible for business organizations to exploit the usage of these Recommendation Systems by suggesting items that are lovable to the users. Ever wondered how these applications suggest or recommend the movies which are appealing to us using such recommendation systems? This system is used to learn the watching patterns of the user and recommends the movies accordingly. We all come across these systems once in a while. These systems are not only used in entertainment applications but they are also used in many E-Commerce websites such as Amazon, Ebay and Best Buy.

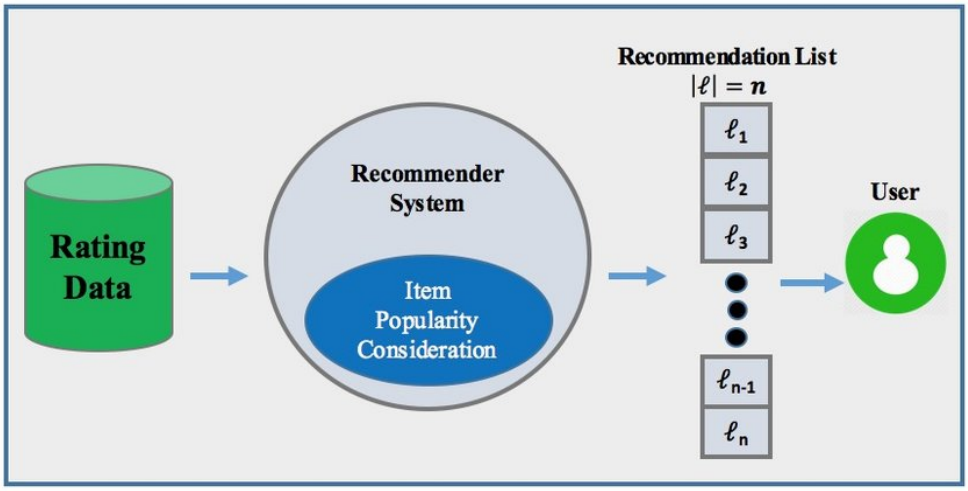
We have come up with the similar idea of designing our own Movie recommendation system which helps the user by recommending the top movies based on his/her preferences and ratings history.

There are three algorithms which helps us to build Recommender Systems

1. Popularity based Filtering
2. Content based Filtering
3. User based Collaborative Filtering

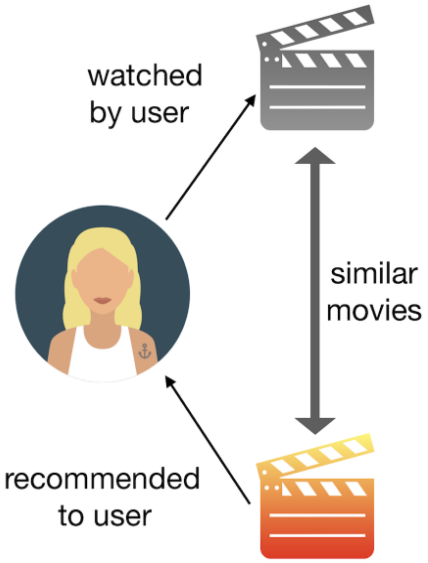
Popularity based Filtering:

This is a simple Recommender System which recommends the movies to all users in a generalized way based on popularity. The basic idea behind this is that the user likes the movies which are highly popular. So, we cannot expect user preference based recommendations in this model.



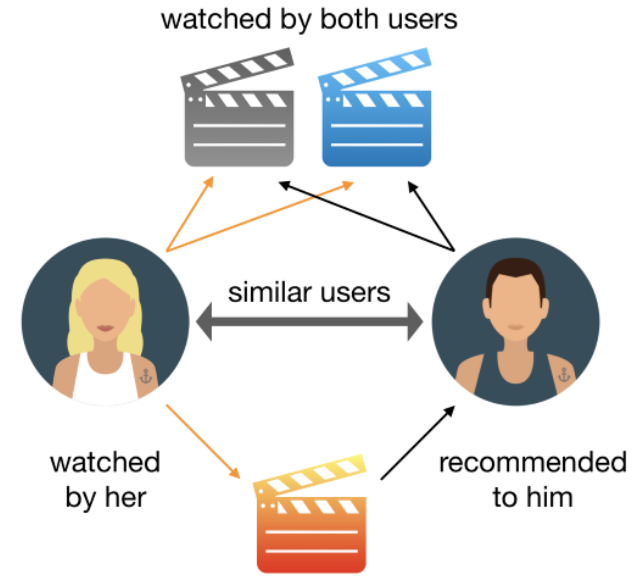
Content based Filtering:

This is a Recommender System which recommends the movies by his/her preference and their data. This system does not take any data from other users. The basic idea behind this model is if the user likes one movie he may like the movies which are similar to it. Attributes such as genre, director, description and actors will be used by this model.



Collaborative based Filtering:

This is a Recommender System which recommends the movies by his/her preference and it also considers data of other similar users. The watch history of every user will be used in this algorithm. The basic idea behind this model is that movies watched by one user are recommended to similar users. On using Collaborative based filtering, the outcome of the recommendation is obtained from multiple users' data and it doesn't rely on single user data.



The main motivation behind doing this project is the Netflix application. It suggests movies according to my interests. So, it raised a bean of curiosity to learn more about these recommendation systems and how it works.

We built a recommender system using the three algorithms mentioned above and also observed how best these algorithms worked.

RELATED WORK

We went through various academic papers to complete this work. We got to know how to build a data mining pipeline.

There is an exponential growth in data nowadays. So, it is so important to extract usable data. Now, the data mining pipeline comes into picture[1].

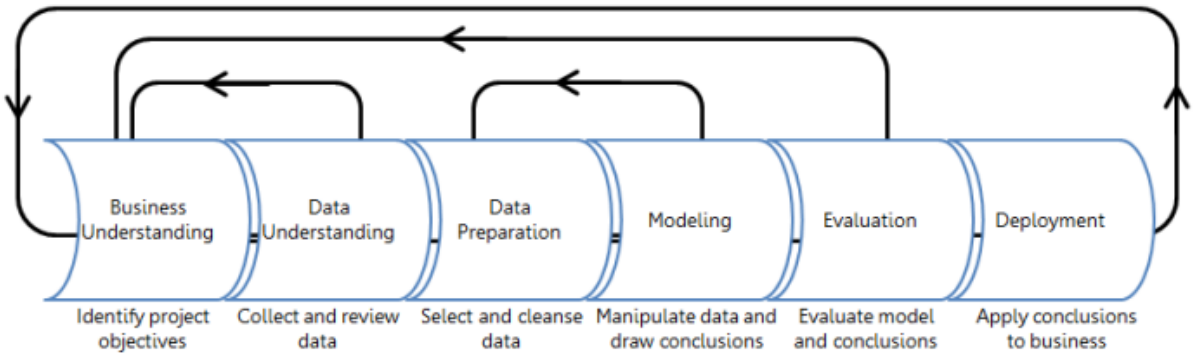
We learnt the business purposes of recommendation systems.

We learnt how the movie recommendation systems are being implemented. We learnt the details of evaluation of the model from this paper[5].

These references helped us to finish our work efficiently without any hardships. There is a possibility of building Hybrid Recommendation models[2]. We came across these Hybrid Recommendation Systems in our research. We are going to save this Hybrid model as the future progress of our work.

LITERATURE SURVEY

[1] illustrates that an individual needs to build the data mining pipeline in such a way that the inputs have to go through a number of steps or functions connected together to create an output. They state that each step in the pipeline should follow the next step providing some type of outcome.



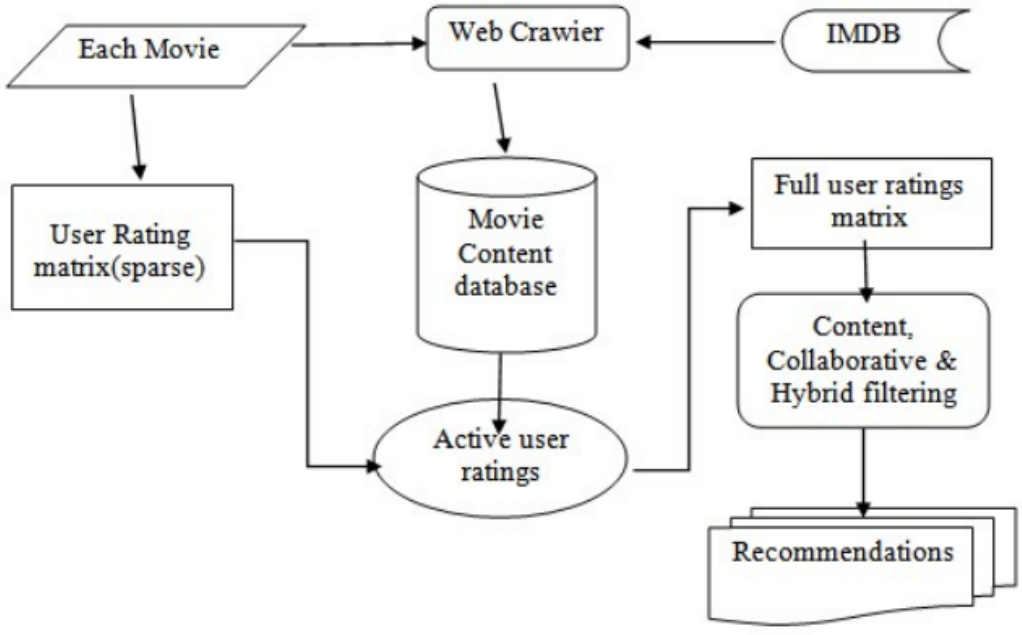
[5] states that Recommender systems improves businesses by assisting their customers in selecting the best choice out of uncountable choices. They implemented five of the popular movie recommendation approaches to predict unknown ratings and recommend to users accordingly. After implementation, their performances were compared using RMSE and MSE metrics. We also used our metrics as RMSE to evaluate our models. They also used two different approaches using neural networks: one neural network for all users, and one neural network for each user. In both cases, 'tanh' activation function performed better than the others.

[3] proposes that Recommender systems are so popular these days but still have few shortcomings, such as scalability and sparsity. They addressed these problems in their paper.

This gave us the understanding of advantages and disadvantages of each model we constructed.

In [2] they have implemented a hybrid movie recommendation system. This is implemented in Python Programming Language. They used the metrics as RMSE to observe the model.

They proposed structure of the Hybrid model as this:



METHODS

For this Recommender Systems to work we need a dataset of movies with ratings and popularity. And we also need metadata of the movie to build a Content based Filtering model.

There's a dataset called "The Movies Dataset" on the Kaggle website which consists of data which is suitable for our project. The metadata of 45000 movies are available in this data. It is taken from MovieLens Dataset. These files contain movies released before 2017. Attributes such as cast, crew, keywords, overview, budget, revenue, release dates, languages, countries, vote counts and vote averages are contained in this dataset. 26 million ratings of all the movies are also present in this file. These ratings are obtained from a website called GroupLens.

Data Preparation:

We collected the data from the Kaggle website and MovieLens website to build Recommender Systems using three different approaches.

Data Cleaning:

We removed the rows which contain Null values. We removed all the duplicate values present in the data. We changed data-types of particular columns to do the computation according to our requirement.

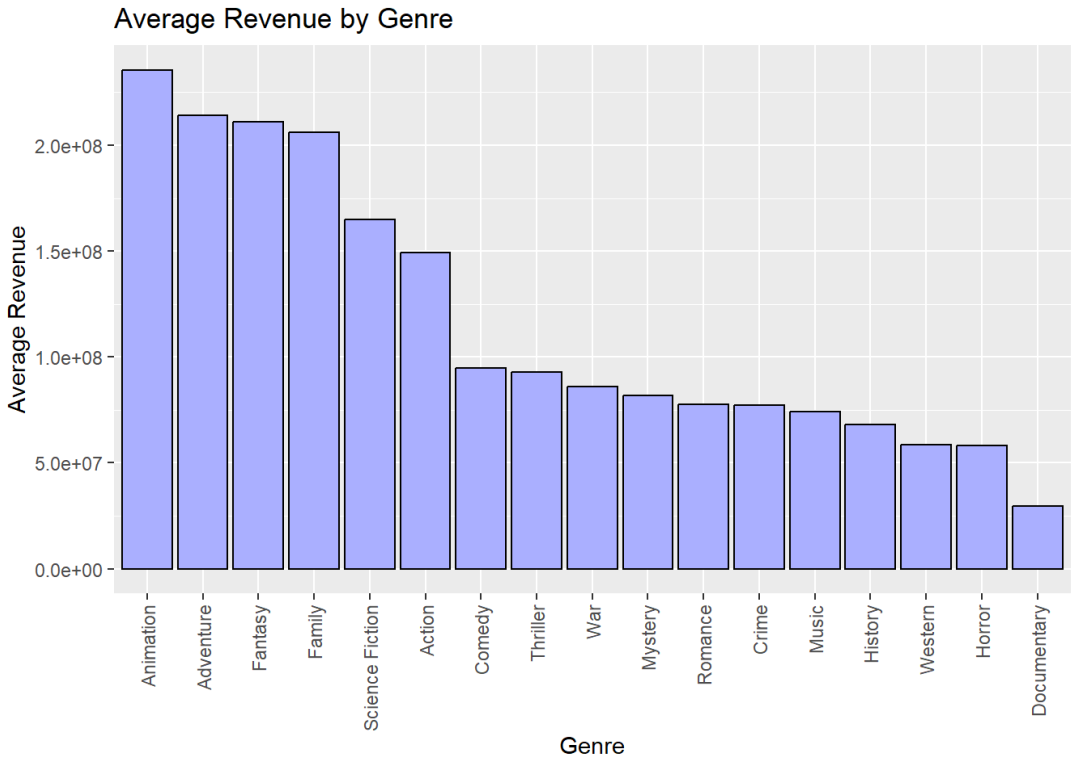
Data Filtering:

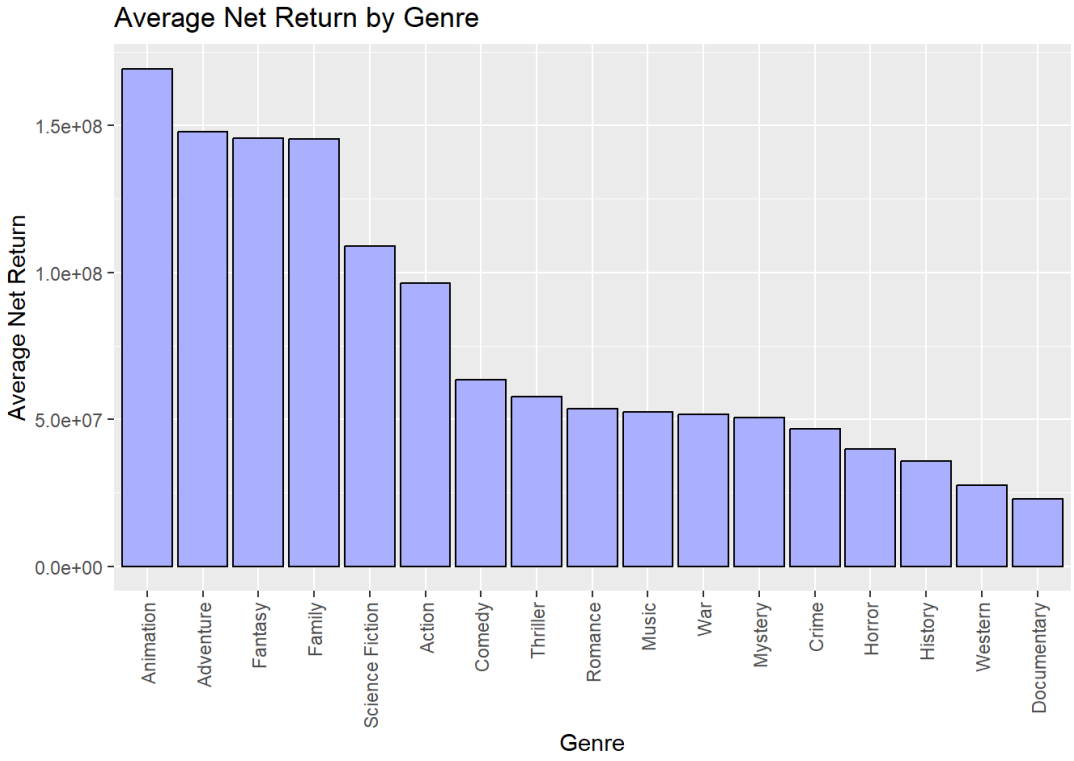
We removed the unwanted columns in our dataset. This will also reduce the burden to the system. We added a few extra columns from existing columns which helped us to do Analysis.

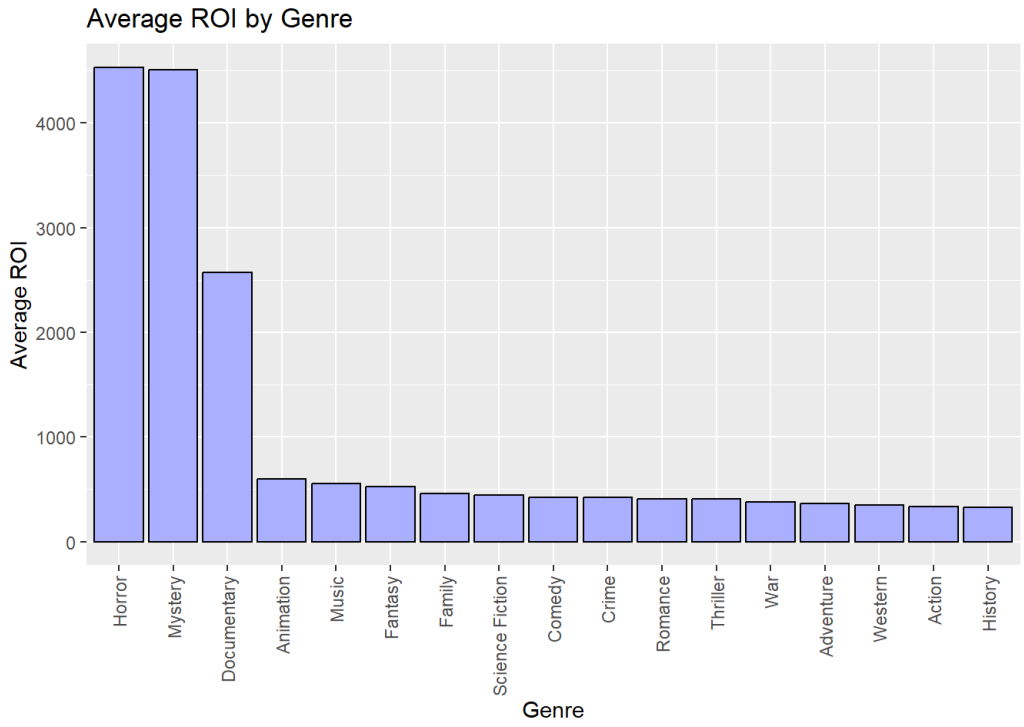
Outliers Removal:

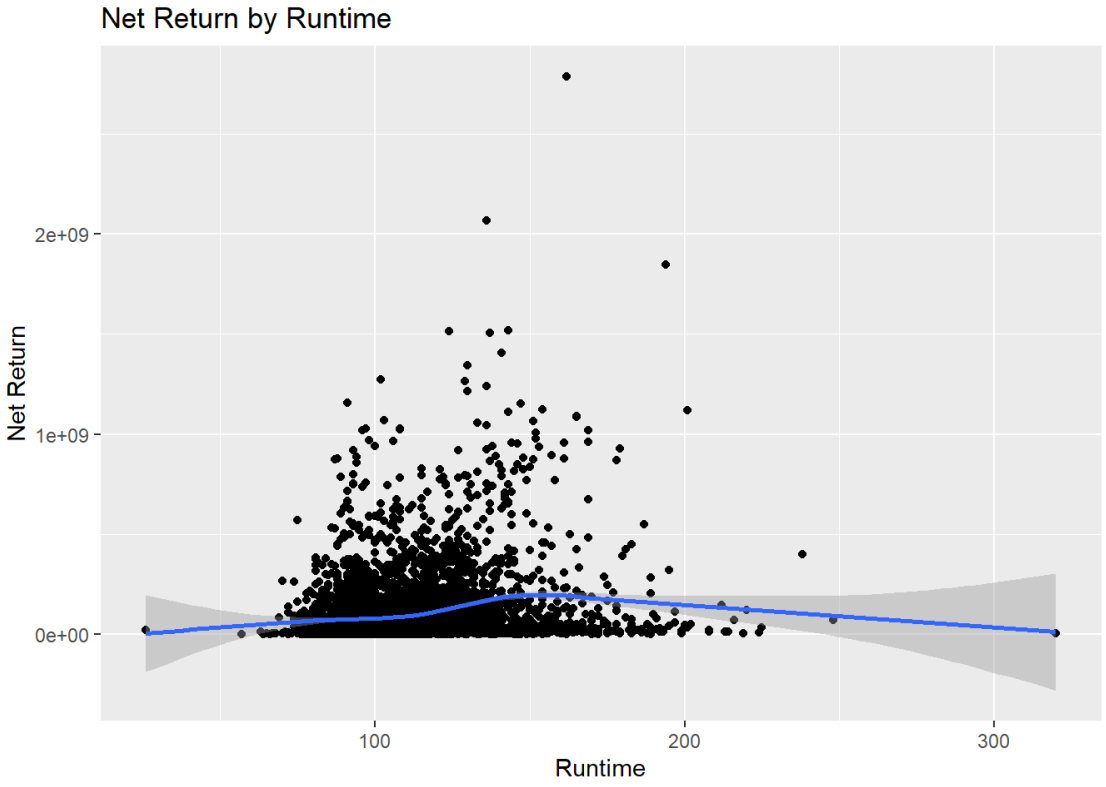
We removed only the one outlier present in our data.

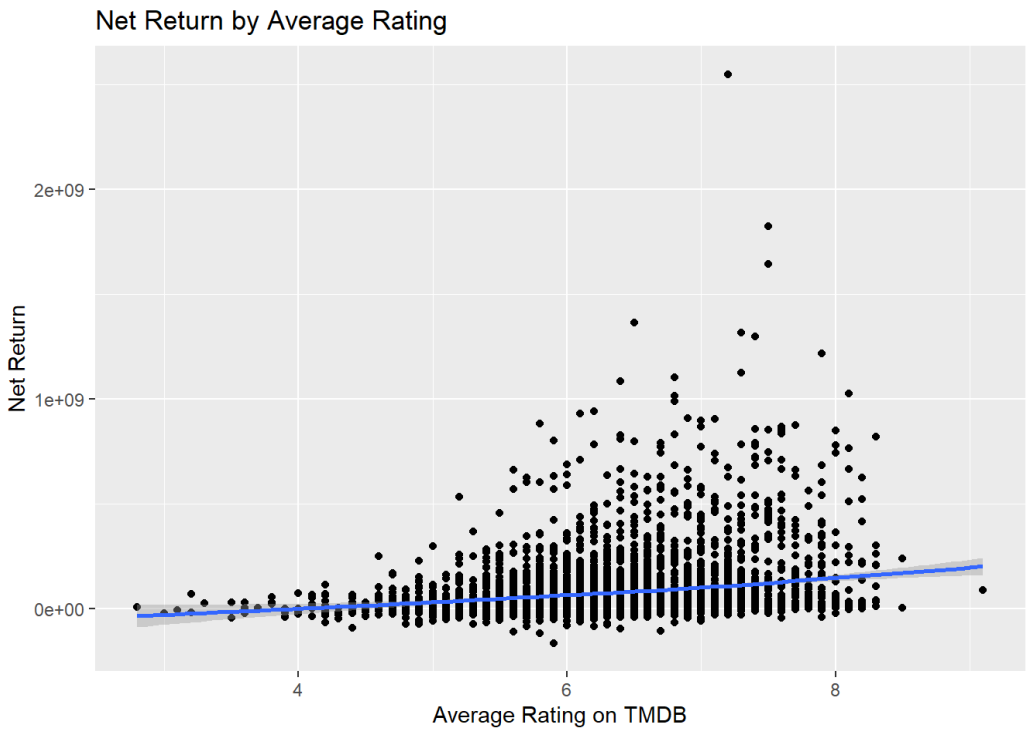
Analysis of the Data:






Average ROI by Genre
Average ROI
4000
3000
2000
1000
0
Horror
Mystery
Documentary
Animation
Music
Fantasy
Family
Science Fiction
Comedy
Crime
Romance
Thriller
War
Adventure
Western
Action
History
Genre


Net Return by Runtime
Net Return
2e+09
1e+09
0e+00
100
200
300
Runtime



POPULARITY BASED MODEL

This is a simple model we created using the basic idea that users are more likely to watch the movies which collected more revenue. These are the results that we obtained. We can see that those movies are highly appraised movies. But there is a lack of user preference in this model. But we can use this model to attract a large mass of population rather than a single user.

```
##                          top_10.original_title
## 1                                       Avatar
## 2                 Star Wars: The Force Awakens
## 3                                      Titanic
## 4                                The The Avengers
## 5                               Jurassic World
## 6                                    Furious 7
## 7                     Avengers: Age of Ultron
## 8  Harry Potter and the Deathly Hallows: Part 2
## 9                                       Frozen
## 10                        Beauty and the Beast
```

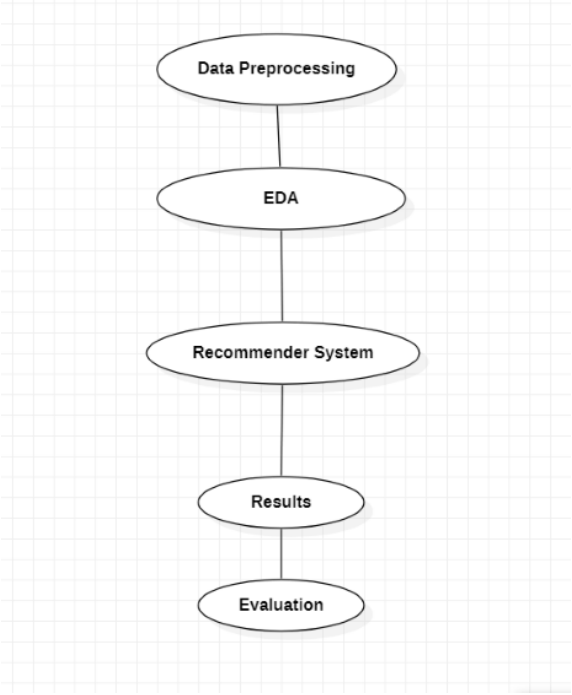
CONTENT BASED FILTERING APPROACH

```
##      movieId                      title                                genres
## 144      145           Bad Boys (1995) Action|Comedy|Crime|Drama|Thriller
## 1004    1026 So Dear to My Heart (1949)                      Children|Drama
## 1286    1319    Kids of Survival (1996)                         Documentary
## 2910    3001      Suburbans, The (1999)                               Drama
## 3231    3323      Chain of Fools (2000)                       Comedy|Crime
```

USER BASED COLLABORATIVE FILTERING

```
##       [,1]
##  [1,] "Heidi Fleiss: Hollywood Madam (1995)"
##  [2,] "Love & Human Remains (1993)"
##  [3,] "My Crazy Life (Mi vida loca) (1993)"
##  [4,] "Diabolique (1996)"
##  [5,] "Truth About Cats & Dogs, The (1996)"
##  [6,] "Visitors, The (Visiteurs, Les) (1993)"
##  [7,] "Cimarron (1931)"
##  [8,] "Marty (1955)"
##  [9,] "Rushmore (1998)"
## [10,] "Last Days, The (1998)"
```

METHODOLOGY



RESULTS AND DISCUSSIONS

Popularity based model

We used movie metadata to build this model. The ideology behind this model is that users like movies that are highly successful in profits. So, we just used data preparation techniques to build this model. So, it'll display the top 10 movies that gained higher profits at the box-office.

Advantages

The model can suggest movies without any information about the user.

Disadvantages

Every user will have the same recommendation list of movies.

Content based model

We used ratings.csv file to build this model. Here, we predicted the ratings based on the features of the movies. We construct a binary matrix with rows consisting of movies and columns consisting of features based on whether a feature is available in that movie or not. We calculated the cosine similarities between pairs of movies. The ratings of the movies can be predicted using the weighted averaging technique for recommendation purposes.

Advantages

Learns user's preferences and Highly personalized for the user.

Disadvantages

Doesn't take into account what others think of the item, so low quality item recommendations might happen.

Collaborative based model

All users' ratings for films were extracted using the ratings.csv file, which we got from the GroupLens website. We used-user based CF model in the recommenderlab package. Using centered cosine similarity between two users, you may determine which two people have similar tastes in movies. When two users are similar, their ratings' similarity value is close to 1 (or positive compared to other users), which indicates that these two individuals are probably watching similar movies. Therefore, the system suggests a movie to the first user who hasn't seen it if the other person has seen it and gave it a high rating.

Advantages

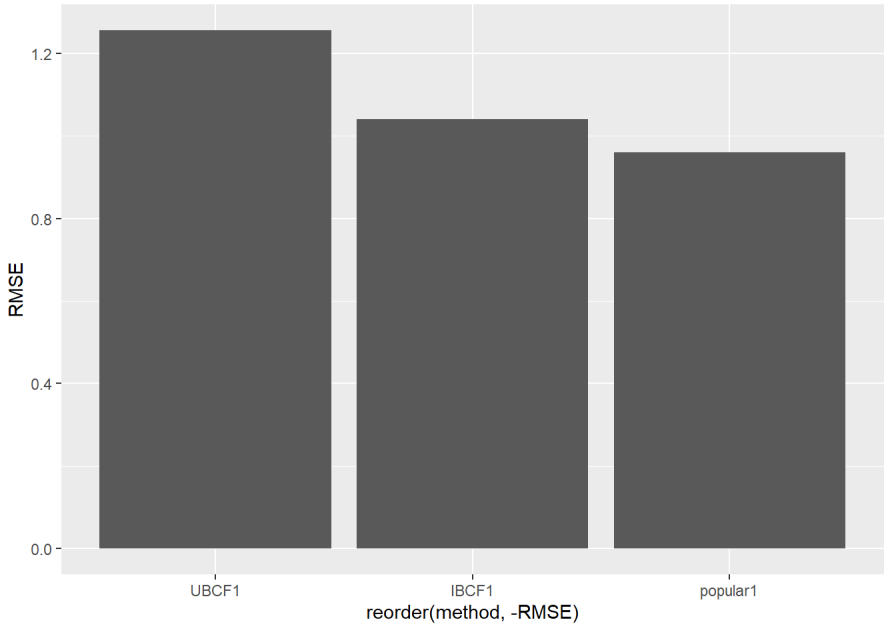
Takes other user's ratings into consideration and Adapts to the user's interests which might change over time.

Disadvantages

Privacy issues when trying to learn the user's preferences and There might be a low amount of users to recommend.

We evaluated our models by using the recommenderlab package present in R language. The results are as below.

EVALUATION



We evaluated our model with the parameter RMSE. RMSE for every model is low.

Every model has its own advantages and disadvantages.

CONCLUSION

We built three types of recommendation systems. But there are other models out there. We have data of only 671 users and their reviews which might not be sufficient data for a proper Recommendation System. Our work is better as far as possible.

There are advantages and disadvantages of every model. So, There is a Hybrid Recommendation System which overcomes the disadvantages of these models.

Disadvantages

Popularity based model - There is no user preference.

Content based model - The proper data distribution is mandatory for the model to work efficiently

User based Collaborative model - This model takes more memory as it should calculate similarity among every user. If the number of users are in millions. It will take more time for computation.

So, the future scope of our project will be building a Hybrid Recommendation System.

DATA AND SOFTWARE AVAILABILITY

You can find our work here :

https://github.com/maheswarreddy01/movie_recommendation_system

REFERENCES

[1]. Marija Juodyte, “Overview: Data Mining Pipeline”,
URL:
https://www5.in.tum.de/lehre/seminare/datamining/ss17/paper_pres/01_pipeline/Data_Mining_Pipeline.pdf

[2]. Rajeev Kumar |Guru Basava | Felicita Furtado “An Efficient Content, Collaborative – Based and Hybrid Approach for Movie Recommendation Engine” Published in International Journal of Trend in Scientific Research and Development(ijtsrd), ISSN: 2456-6470, Volume-4 |Issue-3,April 2020, pp.894-904,
URL: www.ijtsrd.com/papers/ijtsrd30737.pdf

[3]. MaríaN.Moreno, SaddysSegreraVivian, F.López,María Dolores Muñoz, Ángel LuisSánchez, “Web mining based framework for solving usual problems in recommender systems. A case study for movies′ recommendation”, Neurocomputing, volume 176, February 2016. https://doi.org/10.1016/j.neucom.2014.10.097

[4]. SRS Reddy, Sravani Nalluri, Subramanyam Kunisetti, S. Ashok & B. Venkatesh, Content-Based Movie Recommendation System Using Genre Correlation,
URL: https://link.springer.com/chapter/10.1007/978-981-13-1927-3_42#Sec3

[5]. Mojdeh Saadati, Syed Shihab, Mohammed Shaiqur Rahman, Movie Recommender Systems: Implementation and Performance Evaluation
URL: https://arxiv.org/ftp/arxiv/papers/1909/1909.12749.pdf