Project no :- 3

Movie recommendation system

(Content based)

By

Ankit Gurudas Dhanore

Content

- Problem statement
- Objective
- Introduction
- Importing libraries
- Data summary
- EDA and Preprocessing
- Feature engineering
- Modelling
- Challenges faced
- Conclusion

Problem Statement

- Develop a movie recommendation system that can suggest movies to user based on thier preferences and similarity between the movies.

Objective

- With the increasing number of movies available on various platforms, it has become difficult for users to find the right movie to watch. This has resulted in a growing need for a

recommendation system that can assist users in discovering new and relevant movies based on their personal preferences.

- The objective of the movie recommendation system is to analyze the preferences and provide recommendations that are most relevant and appealing to them. This requires the system to have a deep understanding of movie genre, director, cast, and other important attributes to make accurate recommendations.

Introduction

❖ Popularity-Based Recommendation System

- It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the product or movie which are in trend or are most popular among the users and directly recommend those.
- For example, if a product is often purchased by most people then the system will get to know that that product is most popular so for every new user who just signed it, the system will recommend that product to that user also and chances becomes high that the new user will also purchase that.

Merits of popularity based recommendation system

- It does not suffer from cold start problems which means on day 1 of the business also it can recommend products on various different filters.
- There is no need for the user's historical data.

Demerits of popularity based recommendation system

- Not personalized
- The system would recommend the same sort of products/movies which are solely based upon popularity to every other user.

Example

- Google News: News filtered by trending and most popular news.
- YouTube: Trending videos.

❖ Content-Based Recommendation System

- It is another type of recommendation system which works on the principle of similar content. If a user is watching a movie, then the system will check about other movies of similar content or the same genre of the movie the user is watching. There are various fundamentals attributes that are used to compute the similarity while checking about similar content.

- Figure 1 image shows the different models of one plus phone. If a person is looking for one plus 7 mobile then, one plus 7T and one plus 7 Pro is recommended to the user.

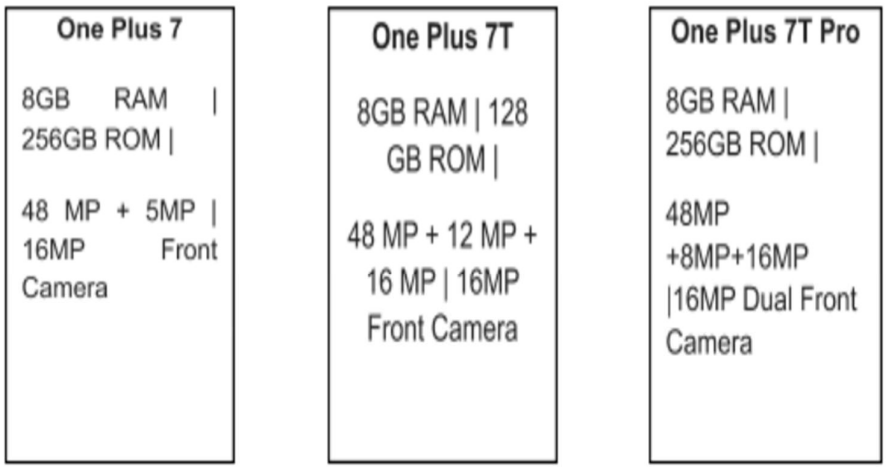


Figure1: Different models of one plus.

- There are different scenarios where we need to check about the similarities, so there are different metrics to be used. For computing the similarity between numeric data, Euclidean distance is used, for textual data, cosine similarity is calculated and for categorical data, Jaccard similarity is computed.

- Cosine Similarity: Cosine of the angle between the two vectors of the item, vectors of A and B is calculated for imputing similarity. If the vectors are closer, then small will be the angle and large will be the cosine.

$$\text{Similarity}(X,Y) = \frac{X.Y}{|X| \times |Y|}$$

Cosine Similarity

Merits

- There is no requirement for much of the user's data.
- We just need item data that enable us to start giving recommendations to users.
- A content-based recommender engine does not depend on the user's data, so even if a new user comes in, we can recommend the user as long as we have the user data to build his profile.
- It does not suffer from a cold start.

Demerits

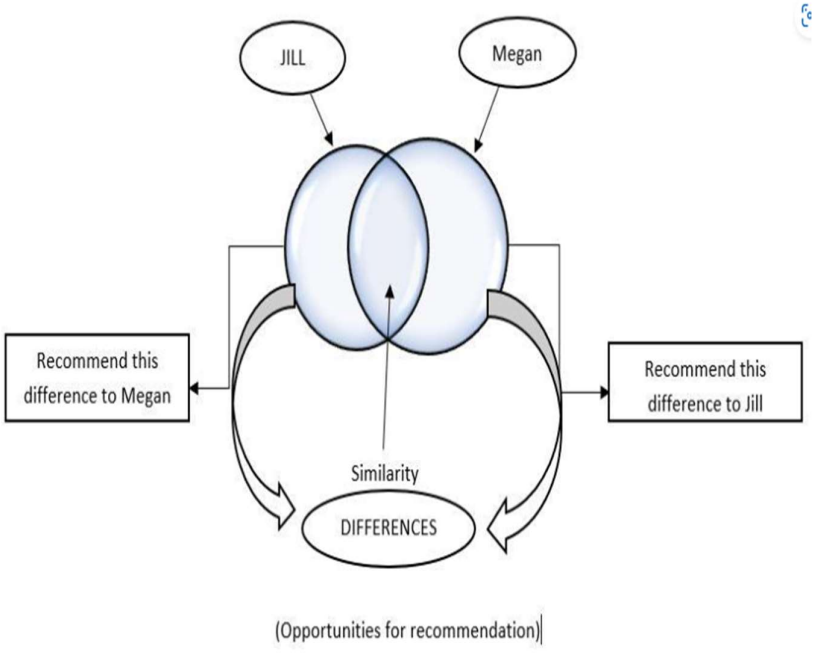
- Items data should be in good volume.
- Features should be available to compute the similarity.

❖ Collaborative Filtering

- It is considered to be one of the very smart recommender systems that work on the similarity between different users and also items that are widely used as an e-commerce website and also online movie websites. It checks about the taste of similar users and does recommendations.
- The similarity is not restricted to the taste of the user moreover there can be consideration of similarity between different items also. The system will give more efficient

recommendations if we have a large volume of information about users and items.

- Figure 2 shows the two different users and their interests along with the similarity between the taste of both the users. It is found that both Jil and Megan have similar tastes so Jill's interest is recommended to Megan and vice versa.



Concept of collaborative filtering.

❖ Hybrid Recommendation systems:

- Hybrid Recommendation systems are combining collaborative and content-based recommendation can be more effective. Hybrid approaches can be implemented by making content-based and collaborative-based predictions separately and then combining them.

Importing Libraries

- Here we import all the libraries which are required for the project like sklearn for modelling and preprocessing , matplotlib and seaborn for data visualization etc.
- Then import the data set and see some basic information regarding the dataset like no of rows and columns and their type.

Data summary

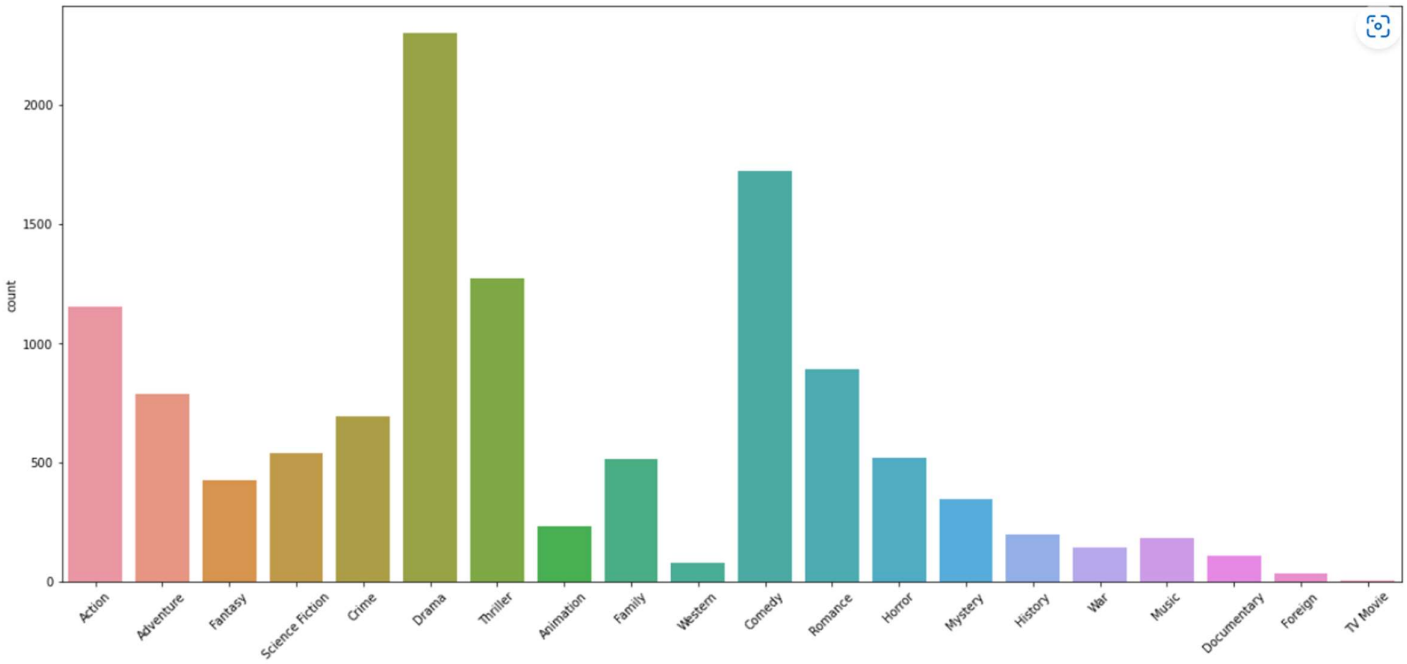
- The data set is from online website which shows the differents movies.
- This data contains two different dataset .after merging the two datasets we are getting many unnecessary data from that we select our required columns and at the end we are left with 4809 rows and 7 columns.
- **Genre :-** A genre is a way of categorising types or classes of literature. In popular usage, genres help us to group or organise literary works into recognisable styles, shared conventions, settings, and themes.
- **ID :-** This is a specific no allotted to the movie. this can be used in deployment part so we are keeping it for now.
- **Keywords :-** This are nothing but short phrase or tags that describe the movie in some way , this allow consumer to easily search and discover titles.
- **Title :-** It is an identifying name given to a film. This is again Important for deployment of model as well.
- **Overview :-** A overview is typically a one page document that summarizes a film.
- **Cast** :- A cast is the group of actors who make up a film
- **Crew**:- A film crew is a group of people hired by a production company, for the purpose of producting a film or motion picture.

EDA and Preprocessing

- First we checked for missing values but there are 0 null values present in the dataset.
- Then we checked for duplicate values but there also 0 duplicate values are present.
- Since there are no null or duplicate values are present in the dataset we jump to next step which is EDA and preprocessing.

❖ For Genres

- Here the data was in the form of the a string which contains list of dictionary so we cannot use that direct we need to do some preprocessing.



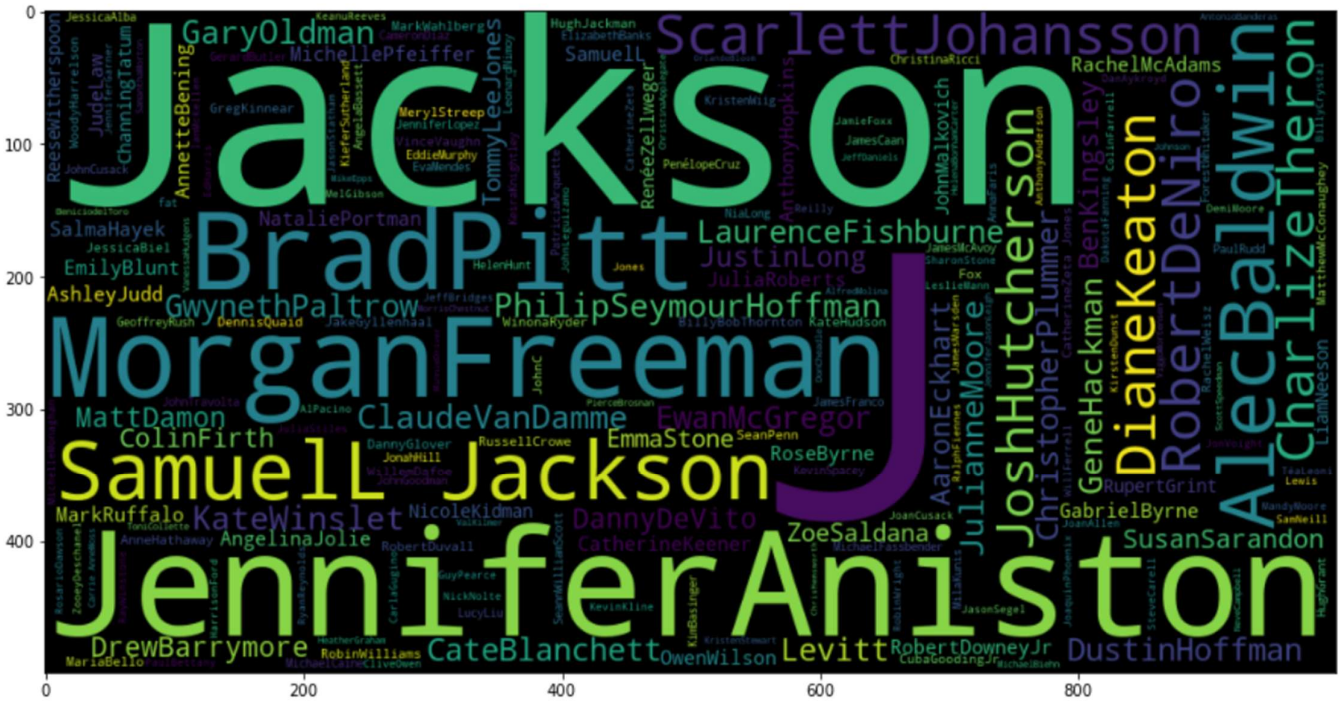
- And by doing countplot we can see that maximum no of genres are from the drama, comedy, action and Thriller.

❖ For Keywords

- Here again the data was in the form of the a string which contains list of dictionary so we cannot use that direct we need to do some preprocessing .
- We take out important tags form the dictionary and convert it into wordcloud so that we can see which tags are frequently repeating.

❖ For Cast

- We don’t want each and ever cast member so we selected only first three cast memers which are most important.

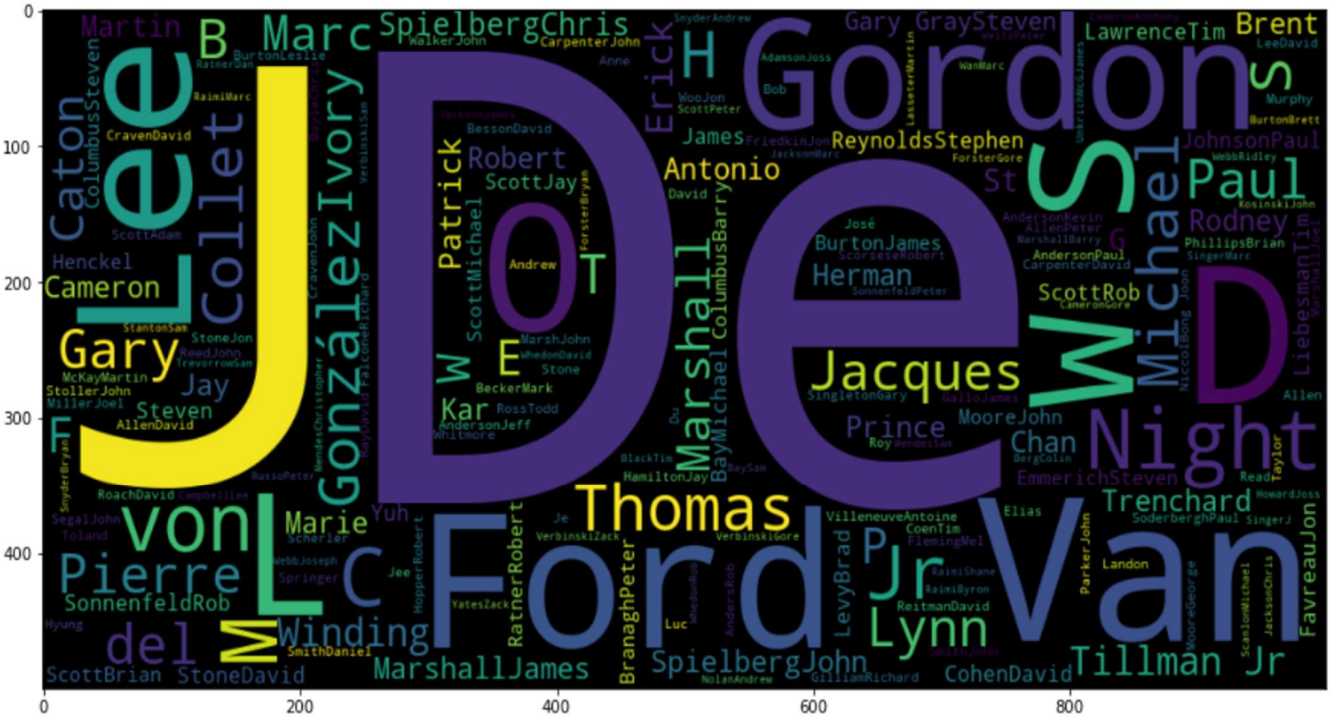


- We made wordcloud out of that so that we can see which actors and actresses are casted in most of the films .

- Jackson, Jennifer aniston, morgan freeman etc are some of the celebrities which work in most of the films.

❖ For Crew

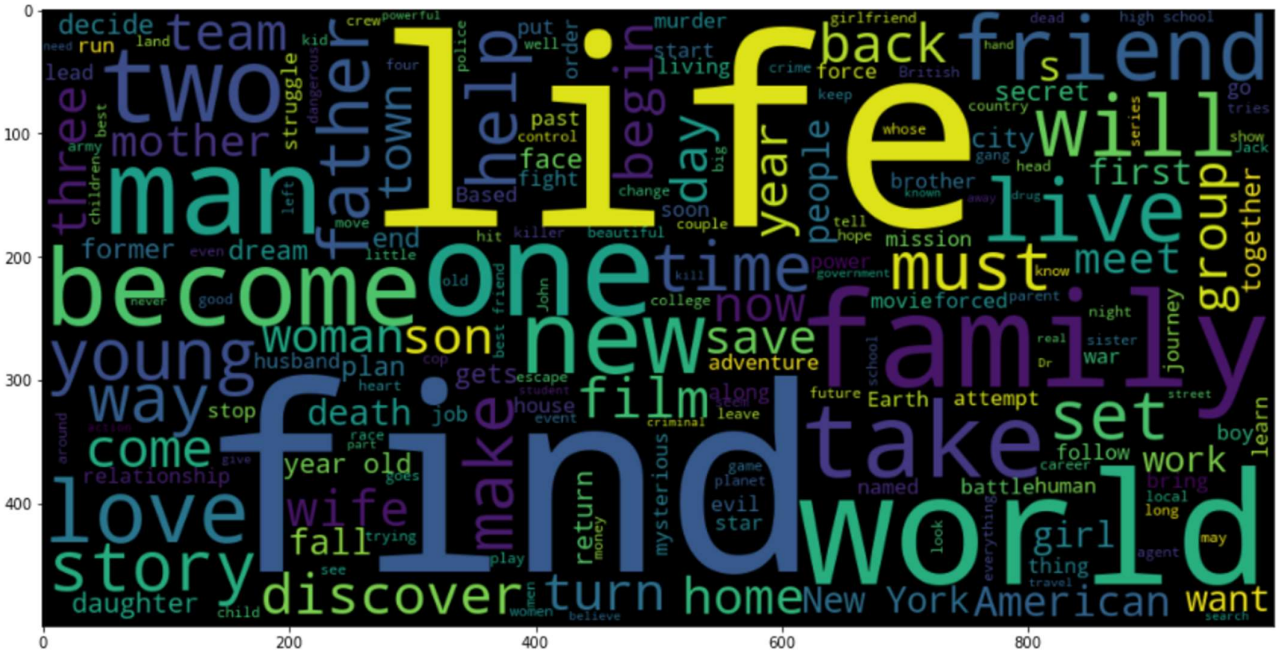
- In this column we only want to consider the director of the movie so that the data does not get so complicated.



- It is clear that the j, de, gordan, ford etc has made most of the movies.

❖ For Overview

- This are the some words which are most repeated in the overview .



- By this we can see that most common words in the overview is life, find, world etc.

Feature engineering

- Now we have to combine all list of columns and make one list but overview is in sring so we convert that to list.
- After this we will remove spaces from the elements of the column genre, keywords, cast, crew to avoid confusion for the model.
- Now we will combine all columns and make one column name tags which will include overview, genres, keywords, cast and crew.
- After this we will select for final columns which are id, title and tags.
- Earlier we convert all column into list and combine it now we will again convert it into string and convert it into lowercase.
- At last we will apply stemming to the tags column and again save it into the tags column.

Modelling

- For modelling we will use countvectorizer.
- After that we will fit_transform our tags column.
- We will get sparse matrix then we will convert that to array
- Now we will recommend movies on the basis of cosine similarity.
- Cosine similarity will provide distance of one movie with rest of the movies and provide similarity matrix.

```
]: similarity = cosine_similarity(count_vec)
   print(similarity)
```

```
[[1.         0.08346223 0.0860309  ... 0.04499213 0.         0.        ]
 [0.08346223 1.         0.06063391 ... 0.02378257 0.         0.02615329]
 [0.0860309  0.06063391 1.         ... 0.02451452 0.         0.        ]
 ...
 [0.04499213 0.02378257 0.02451452 ... 1.         0.03962144 0.04229549]
 [0.         0.         0.         ... 0.03962144 1.         0.08714204]
 [0.         0.02615329 0.         ... 0.04229549 0.08714204 1.        ]]
```

- This distance is according to the index.
- After sorting the distance we will get the index of that movie and from that index we will recommend the title of the movie to the user.

Challenges faced

- Most difficult task was to do preprocessing as data was not in proper format .

Conclusion

- The objective of the movie recommendation system is achieved by analyzing the preferences and providing recommendations that are most relevant and appealing to them.