Delhi Technological University

Swarm Optimization & Evolutionary Computing
(CO-423)

Movie recommendation system

SUBMITTED BY:
Om nath kushwaha (2K18/CO/409)

SUBMITTED TO:
Ms. Pratima Sharma
Dept. of Computer Engineering

November, 2021

ACKNOWLEDEMENT

We would like to express our sincere gratitude to all those who provided us with the opportunity to complete this report.

First of all, we would like to thank our parents for giving us encouragement, enthusiasm, and invaluable assistance. Without their support, we might not have been able to complete this project properly. Secondly, we would like to thank Mr. Yogesh Singh, Vice-Chancellor of Delhi Technological University for managing the college and the curriculum and for his supervision and support.

Thirdly, we would like to thank Ms. Pratima Sharma, Professor in the Department of Computer Engineering, who provided us the opportunity for writing this report and for her constant guidance, comments, and suggestions throughout the development of the project.

Finally, we extend our gratitude to all other unnamed people who helped us in various ways to complete this report.

CERTIFICATE

This is to certify that the technical report entitled “Movie recommendation system” is a record of the bonafide work carried out by Mr. om nath kushwaha 2k18/co/409)
Delhi Technological University during the academic year 2021.

MS PRATIMA SHARMA

PROFESSOR

DEPT. OF COMPUTER ENGINEERING

Table of Contents

ABSTRACT:

A recommendation engine filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behavior of a customer and based on that, recommends products which the users might be likely to buy. If a completely new user visits an e-commerce site, that site will not have any past history of that user. So how does the site go about recommending products to the user in such a scenario? One possible solution could be to recommend the best selling products, i.e. the products which are high in demand. Another possible solution could be to recommend the products which would bring the maximum profit to the business. Three main approaches are used for our recommender systems. One is Demographic Filtering i.e They offer generalized recommendations to every user, based on movie popularity and/or genre. The System recommends the same movies to users with similar demographic features. Since each user is different, this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience. Second is content-based filtering, where we try to profile the users interests using information collected, and recommend items based on that profile. The other is collaborative filtering, where we try to group similar users together and use information about the group to make recommendations to the user.

INTRODUCTION:

A recommendation system is a type of information filtering system which challenges to assume the priorities of a user, and make recommendations on the basis of user's priorities. Huge range of applications of recommendation systems are provided to the user. The popularity of recommendations systems have gradually increased and are recently implemented in almost all online platforms that people use. The content of such system differs from films, podcasts, books and videos, to colleagues and stories on social media, to commodities on e-commerce websites, to people on commercial and dating websites. Often, these systems are able to retrieve and filter data about a user's preferences, and can use this intel to advance their suggestions in the upcoming period. For an instance, Twitter can analyze your collaboration with several stories on your wall so as to comprehend what types of stories please you. Many a times, these systems can be improvised on the basis of activities of a large number of people. For example, if Flipkart notices that a large number of users who buy the modern laptop also buy a laptop bag. They can commend the laptop bag to a new customer who has just added a laptop to his cart. Due to the advances in recommender systems, users continuously expect good results. They have a low edge for services that are not able to make suitable recommendations. If a music streaming application is not able to foresee and play song that the user prefers, then the user will just stop using it. This has led to a high importance by technical corporations on refining their recommendation structures. However, the problem is more complicated than it appears. Every user has different likes and dislikes. In addition, even the taste of a single customer can differ depending on a large number of aspects, such as mood, season, or type of activity the user is performing. For an instance, the type of music one would prefer to listen during exercising varies critically from the type of music he would listen to while preparing dinner. They must discover new areas to determine more about the customer, whilst still determining almost all of what is already known about of the customer. Two critically important methods are widely used for recommender systems. One is content-based filtering, where we attempt to shape the users preferences using data retrieved, and suggest items based on that profile. The other is collaborative filtering, where in we try to cluster alike users together and use data about the group to make recommendations to the customer.

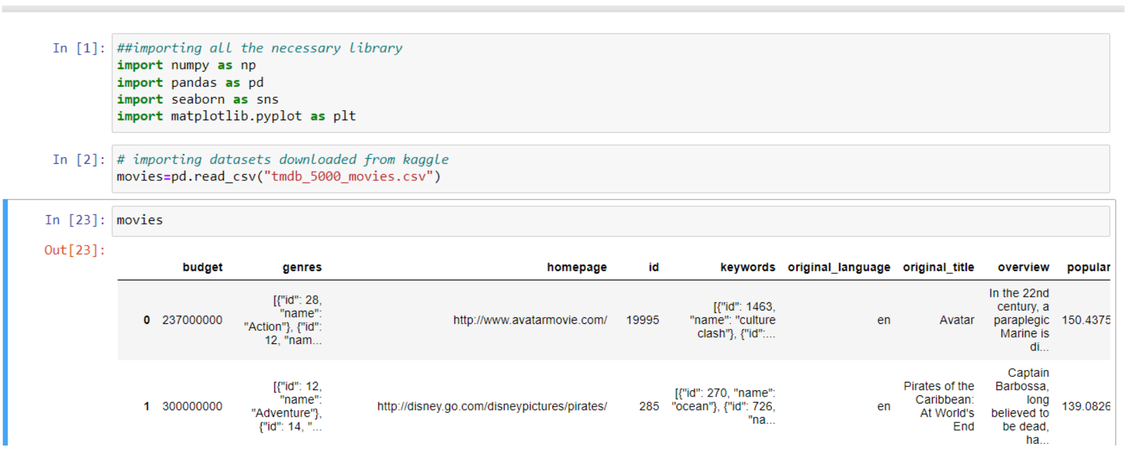
TYPES OF MOVIE RECOMMENDATION SYSTEM:

Collaborative Filtering: The basic methodology of collaborative filtering systems is that these undetermined ratings can be credited since the noticed ratings are often highly linked across several users and items. For an instance, assume two users named Ramu and Shamu, who have very comparable tastes. If the ratings, which both have stated, are very similar, then their resemblance can be determined by the fundamental algorithm. In such cases, there is a high probability that the ratings where in just one of them has definite value, are also likely to be similar. This similarity can be used to make interpretations about partly stated values. Almost all the projects for collaborative filtering emphasis on leveraging either item associations or user associations for the calculation procedure. Many of the models implement both kinds of correlations. Additionally, some mock-ups use judiciously designed optimization procedures to generate a training model in much the similar way a classifier generates a training model from the mentioned or specified information. This model is later used for assigning the absent values in the matrix, in the similar way that a classifier assigns the absent test tags. There are two types of methods which are frequently implemented in collaborative filtering that are denoted to as memory-dependent procedures and model-dependent procedures.

<u>**Content Based Filtering**:</u> Content Based Recommendation procedure checks for the adores and aversions of the user and creates a User-based Profile. For producing a user profile, we check for the item profiles and their equivalent user rating. The user profile is the combination of sum of the item profiles where combination being the ratings customer or user has evaluated. After profile of the user has been generated, we estimate the resemblance of the user profile with all the items in the database, which is considered using cosine resemblance between the user generated profile and item profile. Benefits of Content oriented procedure is that other user's information or data is not essential, and the recommender system can commend new commodities or anything which are not evaluated presently, nevertheless the recommender system will not recommend the items outside the type of items the user has given ratings of.

DATASETS USED:

- We have used Movie Dataset downloaded from kaggle
  This dataset consists of: 5000 movies list with different attributes like, their, name, popularity, budget, etc.
- This datasets only consists of movie upto 2014.



```
In [1]: ##importing all the necessary library
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt

In [2]: # importing datasets downloaded from kaggle
        movies=pd.read_csv("tmdb_5000_movies.csv")

In [23]: movies

Out[23]:
```

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popular
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.4375
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.0826

DATA PREPROCESSING:

A real-world data Movie data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

- Since the movie datasets still consists of unwanted columns which are no longer needed so droping unwanted columns.
- We have dropped the columns like budgets, homepage, id, keywords, original_languages, production etc.
- Dropping duplicates movies if there are any.

METHODOLOGIES:

In this projects we have used Content Based recommendation systems for filtering movies because of some advantages.

- The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users.
- The model can capture the specific interests of a user, and can recommend nice items that very few other users are interested in.

- ➢ It uses only the item data maintaining a profile for each item. Each user is assumed to operate independently. No need for data on other users.
- ➢ Considering the attributes or feature of the items, it finds the similarity between items, and recommends the most similar item for an item
- ➢ If we consider the content of a movie as director, writer, cast etc., then each
  of these attributes can be considered as a feature.

EXPERIMENTATION:

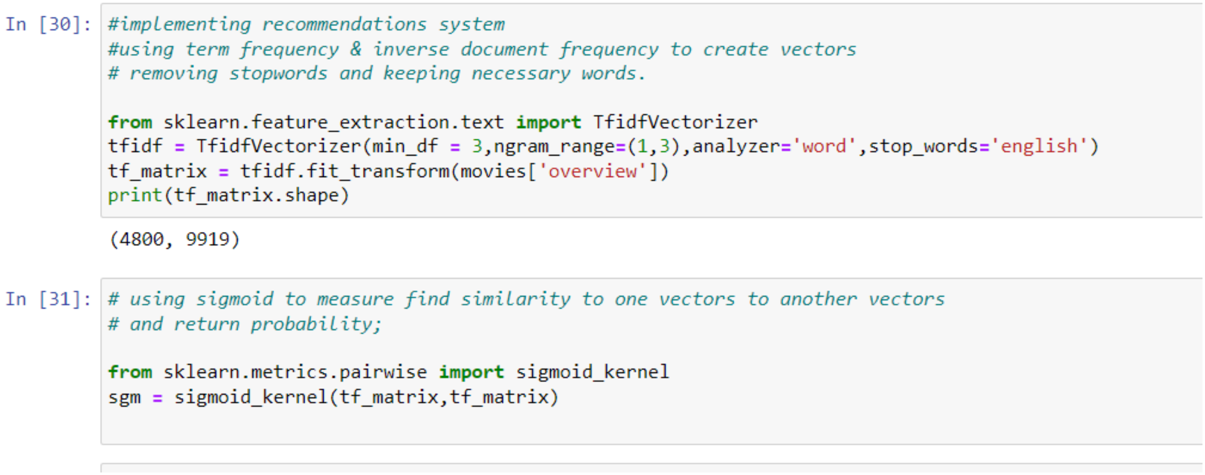
- We recommend the items to the users which are very much similar to the rated item by the user.
- we have used Term frequency and inverse document frequency techniques to create vectors.

TF=No. of repetition of words/No of words in sentence

IDF=log(No of sentences/ No of sentences containing required word).

Final result=TF* IDF

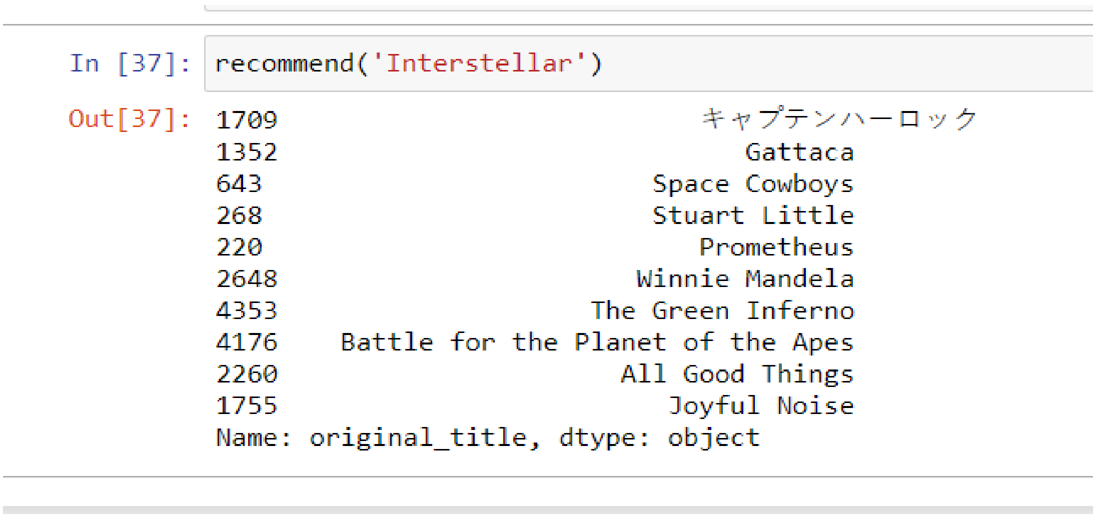
- We have used sigmoid kernel to find similarities between vectors.
- Sigmoid kernel will compare the one movies with indexes of other movies in matrix and return those movies which have some similar characters.



```
In [30]: #implementing recommendations system
         #using term frequency & inverse document frequency to create vectors
         # removing stopwords and keeping necessary words.

         from sklearn.feature_extraction.text import TfidfVectorizer
         tfidf = TfidfVectorizer(min_df = 3,ngram_range=(1,3),analyzer='word',stop_words='english')
         tf_matrix = tfidf.fit_transform(movies['overview'])
         print(tf_matrix.shape)

         (4800, 9919)

In [31]: # using sigmoid to measure find similarity to one vectors to another vectors
         # and return probability;

         from sklearn.metrics.pairwise import sigmoid_kernel
         sgm = sigmoid_kernel(tf_matrix,tf_matrix)
```

RESULTS:

- we can see that the Term frequency and inverse document frequency techniques will recommend the similar kinds of movies as shown in fig:



```
In [37]: recommend('Interstellar')

Out[37]: 1709                  キャプテンハーロック
         1352                             Gattaca
         643                        Space Cowboys
         268                        Stuart Little
         220                           Prometheus
         2648                      Winnie Mandela
         4353                    The Green Inferno
         4176     Battle for the Planet of the Apes
         2260                     All Good Things
         1755                         Joyful Noise
         Name: original_title, dtype: object
```

Fig: 1.1

- when we watch the movies named interstellar the model will recommend the similar types of space related movies to which its parameter is correlated as we can see it in above fig: 1

CONCLUSION:

We have learnt a lot from this project. In real world the recommendation system can play very vital role in changing role people. The model we have made can be used many videos streaming platform for recommending movies to users based on movies they have watched previously. Apart from this model can also be used on many other real world application like ecommerce, music etc In ecommerces it can be used to recommend products based on data and in music they can be used to recommend similar types of music they are listening too. This is what we have learnt from this projects

.

FUTURE WORK:

- In this project we have built the model for recommending the similar types of movies.
- Our future works would be to focued on building the user interface and to visualize the movies recommended with its pictures and artists details as well.

- we will be using react.js for building frontend user interface and databases we are using mongodb. We will feed the recommended movies to fronted to view the movies right in web browsers.

This type of technology are used by different video streaming platforms like Netflix, prime video’s etc to recommend movies to its users.

REFERENCES:

1. Peng, Xiao, Shao Liangshan, and Li Xiuran. "Improved Collaborative Filtering Algorithm in the Research and Application of Personalized Movie Recommendations", 2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications, 2013.

2. Munoz-Organero, Mario, Gustavo A. Ramíez-González, Pedro J. Munoz-Merino, and Carlos Delgado Kloos. "A Collaborative Recommender System Based on SpaceTime Similarities", IEEE Pervasive Computing, 2010.

3. Al-Shamri, M.Y.H.. "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model", Expert Systems With Applications, 200810 4. Hu Jinming. "Application and research of collaborative filtering in e-commerce recommendation system", 2010 3rd International Conference on Computer Science and Information Technology, 07/2010