BITS Pilani, Dubai Campus

Dubai International Academic City (DIAC)

Dubai, U.A.E

Assignment for Artificial Intelligence (CS F407)

On

Content Based Movies Recommendation System

Submitted to

Dr. Sujala D. Shetty

By

Aafia Iqbal	**2018A7PS0061U**
Imad Mehmood	**2019A7PS0182U**
Mardiyah Khadijah	**2018A7PS0257U**

TABLE OF CONTENTS

Abstract

Recommendation systems are the systems that are used to gain more user attraction by understanding the user's taste. These systems have now become popular because of their ability to provide personalized content to users that are of the user's interest. These days millions of products are listed on e-commerce websites that make it impossible to find a product of our desired choice. This is where these systems help us by quickly recommending us with the desired products. Also, Netflix suggests the same genre movies to us by understanding our interest/ choice of movies we like similarly Youtube recommends videos to us. There are many different recommendation engines that work backends to make it possible.

Chapter 1 : PROBLEM STATEMENT

For building a recommender system from scratch, we face several different problems. Currently there are a lot of recommender systems based on the user information, so what should we do if the website has not gotten enough users. After that, we will solve the representation of a movie, which is how a system can understand a movie. That is the precondition for comparing similarity between two movies. Movie features such as genre, actor and director is a way that can categorize movies. But for each feature of the movie, there should be different weight for them and each of them plays a different role for recommendation. So we get these questions:

- How to recommend movies when there is no user information.
- What kind of movie features can be used for the recommender system.
- How to calculate the similarity between two movies.
- Is it possible to set weight for each feature?

Chapter 2 : INTRODUCTION

A recommendation engine is a data filtering tool that uses machine learning algorithms to suggest the most relevant products to a certain user or client. It works on the premise of identifying patterns in customer behavior data, which may be acquired either implicitly or explicitly.

Content-based filtering operates on the assumption that if you like one item, you'll enjoy this one as well. Algorithms employ cosine and Euclidean distances to calculate the similarity of objects based on a profile of the customer's interests and a description of the item (genre, product category, color, word length) [1]. These are the systems that look for similarity before recommending something. We all have seen whenever we are looking for a movie or web series on Netflix, we get the same genre movie recommended by Netflix. The similarity of different movies is computed to the one you are currently watching and all the similar movies are recommended to us. In the case of e-commerce websites, similarity in terms of products is calculated. Considering I am looking for a MacBook then the website will look for all similar products that are similar to MacBook and straight away will recommend us.

Because the suggestions are personal to a single user, the content-based filtering technique does not require any data about other users. This makes scaling down to a large number of people more simple. Collaborative Filtering Methods cannot be compared in any way [2].

The similarity is the main key fundamental in the case of content-based recommendation systems. A most similar thing to what we are currently watching gets recommended to us. There are different techniques or similarity measures that are used to compute the similarity. The following are a few:

Euclidean Distance:- This distance metric is used when we have numeric data. For example, if I want to compute the similarity between One plus 6 and other one plus variants based on ram and camera. The values of ram and camera for each variant would be in numbers. In these cases, we calculate Euclidean distance if the results of this distance come out to be 0 then both the two are considered to be similar whereas if the distance is anything other than 0 then are not similar.

Cosine Similarity:- This type of metric is used to compute the similarity textual data. Consider an example where we have to find similar news or similar movies. We convert these textual data in the form of vectors and check for cosine angle between those two vectors if the angle between them is 0. It means they are similar or else they are not. Most used similarity measures when we talk about the similarity between any textual content. There are other different metrics as well like Jaccard Similarity that is used when we have categorical data.

Chapter 3 : DATASET

The ‘TMDB 5000 Movie Dataset’ is taken into consideration for movie recommendation purposes for this assignment work. This dataset is available on kaggle.com. The dataset is composed of 2 CSV files - ‘tmdb_5000_movies.csv’ and ‘tmdb_5000_credits.csv’

The ‘tmdb_5000_movies.csv’ dataset consists of the following attributes:

- ‘budget’: It indicates the budget of the movie.
- ‘genres’: It indicates the genres of the movie like Action, Documentary, etc.
- A movie can have multiple genres.
- ‘homepage’: It indicates the homepage of the movie. It is basically a website link.
- ‘id’: It indicates movie ID.
- ‘keywords’: It indicates the keywords of the movie. Apart from the title of the movie, keywords give a quick information about the movie.
- ‘original_language’: It indicates whether the movie is originally created in English or other language.
- ‘original_title’: It is nothing but the movie title.
- ‘overview’: It is a short description of the movie.
- ‘popularity’: It is a metric which indicates popularity.
- ‘production_companies’: It consists of the names of companies which has produced the movie.
- ‘production_countries’: It consists of the names of the countries in which the movie production took place.
- ‘release_date’: It consists of the release date of the movie. The format used is yyyy-mm-dd where ‘yyyy’ indicates year of release, ‘mm’ indicates the month of release, and ‘dd’ indicates the day of release.
- ‘revenue’: It indicates the revenue earned by the movie.
- ‘runtime’: It indicates the runtime of a movie. Runtime basically means the length of the movie.
- ‘spoken_languages’: It consists of the languades spoken in the movie.
- ‘status’: It indicates the status of the movie. For example, a movie can be released or not released which basically indicates the status of that movie.
- ‘tagline’: It consists of the tagline of the movie.
- ‘title’: It consists of the title of the movie.
- ‘vote_average’: It indicates the average of the votes.
- ‘vote_count’: It indicates the vote count.

The ‘tmdb_5000_credits.csv’ dataset consists of the following attributes:

- ‘movie_id’: It indicates the movie ID.

- ‘title’: It indicates the title of the movie.
- ‘cast’: It consists of the cast of the movie. Cast implies the actors and actresses who appear in the movie.
- ‘crew’: It consists of those people who are concerned with the production of the movie.

Chapter 4 : EXPERIMENTAL PLATFORM

The analyses were carried out using Jupyter notebook and Python 3.8. We perform the Collaboratory tests, which make it possible to create and execute Python code in the browser, making it ideal for machine learning and data analysis. Colab requires no installation and is a shared Jupyter notebook platform that provides free accessibility to computer resources such as GPUs.

Chapter 5 : CONTENT- BASED FILTERING

A content-based algorithm is proposed for the recommendation system. The objective is to identify clearly recognized qualities of the products, which will subsequently be used to create a "profile." In this sort of system, determining the similarity between objects is equivalent to determining the similarity among their profiles. User profiles are another vital component of a content-based system. Vectors describing the user interests in the same dimensions as the item profiles must be generated. A utility matrix depicts the relationship among users and objects when they are depicted in the same dimensions. The products that users enjoy are a combination of those items' profiles. Both with user and item profile vectors, an estimate of the level to which a user favors an item may be established, which is usually done by calculating the cosine distance between both the user's and item's vectors. Singular value decomposition and the cosine similarity technique were used during this method. It suggests the top n movies list that the active user would watch.

The Vector Space Model is the technique used to model this technique (VSM). It presents the notion of TF-IDF (Term Frequency-Inverse Document Frequency) [20], which determines the item's similarity from its description.

$$\mathrm{Tf}(t)=\frac{\text{frequency occurrence of term t in document}}{\text{totalnumberoftermsindocument}}$$

$$\mathrm{If}(t)= log10\frac{\text{totalnumberofdocument}}{\text{numberofdocumentscontainingtermt}}$$

Figure 2: Term Frequency-Inverse Document Frequency

Three approaches could be used to calculate the similarity among item vectors:

1. Cosine similarity

2. Euclidean distance

3. Pearson’s correlation

The angle of cosine between two things is measured by cosine similarity between two objects. It uses a standardized scale to compare the two texts. It is possible to do so by calculating the dot product of the two identities. The angle between v1 and v2 is shown in the picture below. The greater the similarity in between two vectors, the smaller the angle among them. It indicates that if the angle between the two vectors is small, they are identical, and if the angle is high, the vectors are quite dissimilar.

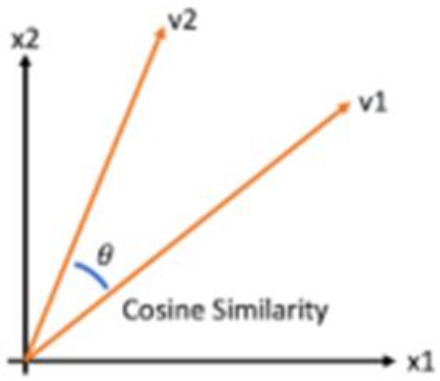


Figure 3: Angle between v1 and v2 to depict similarity

Advantages of content-based filtering are:

- They are capable of recommending unrated items. We can easily explain the working of the recommender system by listing the Content feature of an item.
- Content-based recommender system use needs only the rating of the concerned user, and not any other user of the system.

Disadvantages of content-based filtering are:

- It does not work for a new user who has not rated any item yet as enough ratings are required content-based recommender evaluates the user preferences and provides accurate recommendations.
- No recommendation of serendipitous items.
- Limited Content Analysis- The recommendation does not work if the system fails to distinguish the items that a user likes from the items that he does not like.

Chapter 6 : RESULTS

These are the results of the top five movies as per the attribute ‘popularity’ from the data set:

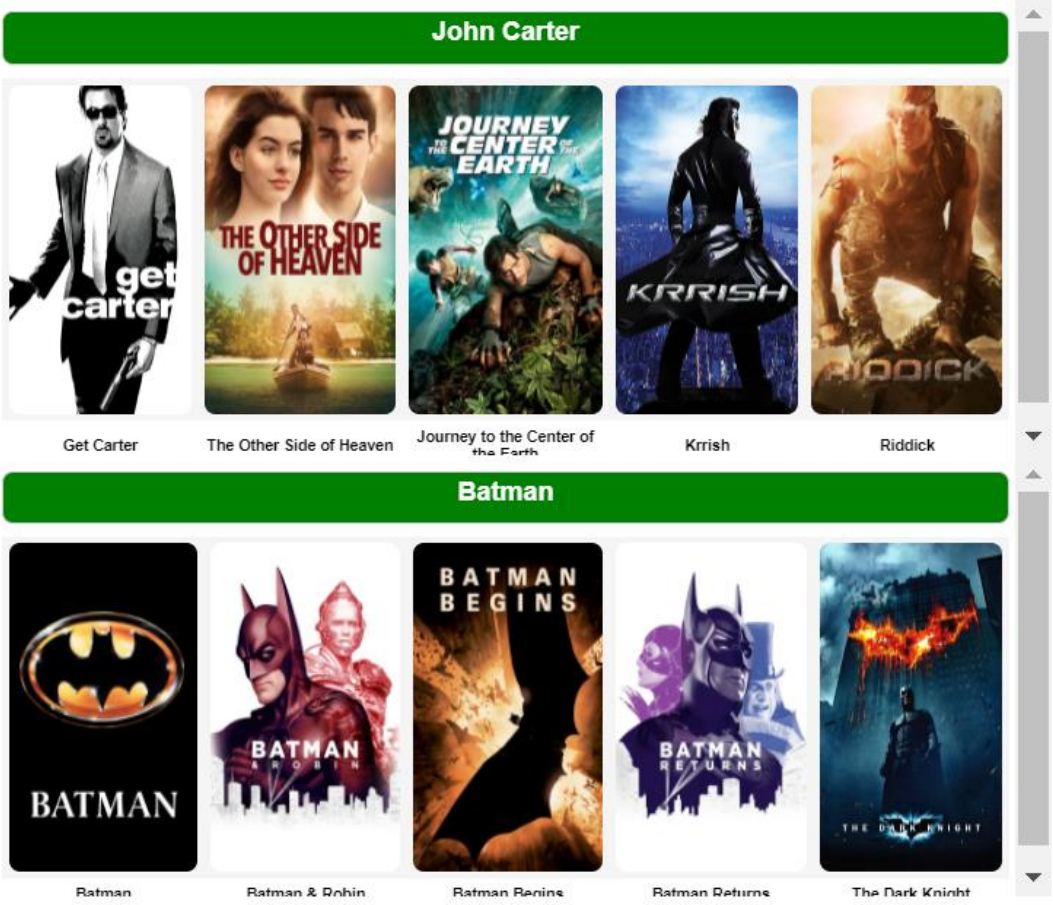
In [9]:
```
#Top 5-Movies as per Popularity Calculation
```

In [10]:
```
popular_movies = dfmovies.nlargest(n=5, columns=['popularity'])[['id', 'title']]
getList_name = {}
for x, xRows in popular_movies.iterrows():
    #print(xRows['id'])
    getResponse = requests.get('https://api.themoviedb.org/3/movie/{}?api_key=c0bda0be71f7815fd6ba2eb5f5c86fd8'.format(xRows[
    getData = getResponse.json()
    getPath = "http://image.tmdb.org/t/p/w500" + getData['poster_path']
    getList_name[xRows['title']] = getPath
display( HTML(f"""<table>
                    <tr>
                        <td><img src={list(getList_name.values())[0]} style='border-radius:10px; height:300px; width:575px; b
                        <td><img src={list(getList_name.values())[1]} style='border-radius:10px; height:300px; width:575px; b
                        <td><img src={list(getList_name.values())[2]} style='border-radius:10px; height:300px; width:575px; b
                        <td><img src={list(getList_name.values())[3]} style='border-radius:10px; height:300px; width:575px; b
                        <td><img src={list(getList_name.values())[4]} style='border-radius:10px; height:300px; width:575px; b
                    </tr>
                    <tr>
                        <td><div style="height:40px; padding-top:15px; text-align:center; font-size:14px; font-weight:bold; b
                        <td><div style="height:40px; padding-top:15px; text-align:center; font-size:14px; font-weight:bold; b
                        <td><div style="height:40px; padding-top:15px; text-align:center; font-size:14px; font-weight:bold; b
                        <td><div style="height:40px; padding-top:15px; text-align:center; font-size:14px; font-weight:bold; b
                        <td><div style="height:40px; padding-top:15px; text-align:center; font-size:14px; font-weight:bold; b
                    </tr>
                </table>"""))
```

Minions | Interstellar | Deadpool | Guardians of the Galaxy | Mad Max: Fury Road

The following are the movies recommended for “John Carter” and “Batman” by our content based recommendation system:

```
In [91]:  # Show the Recommended Name and Poster of Movies in List
          movies_list = {"John Carter", "Batman"}
          for x in movies_list:
              names, poster = getRecommended_movies_name(x)
              show_poster(x, names, poster)
```



Chapter 7 : CONCLUSION

Recommended systems expand up unique possibilities to achieve personal data. It also aids in the reduction of information explosion, which is a real concern with information retrieval systems, and enables users to acquire products and services that are not easily accessible. We devise a technique that emphasizes on the user's specific preferences, and movies are recommended to them dependent on their prior evaluations. This method aids in the improvement of recommendation accuracy. Every user has their own unique profile, with accessibility to his or her own history, likes, ratings, comments, and password modification methods. It also aids in the collection of accurate and authentic data, as well as making the system more effective.

Our suggested algorithm suggests the top n movies to the active user base of the User's interests. As a result, our method may be applied in real-world situations. The KNN method, together with the notion of cosine similarity, is used in this model because it is more accurate than other distance metrics and has a lower computational complexity.

As previously said, recommendation systems cover a wide range of topics. Various strategies for making recommendations are also covered. As a result, the goal of any recommender system is to create a model that provides appropriate recommendations while maintaining the system's efficiency. We can work on a hybrid recommender in the future that uses clustering and similarity to boost efficiency. Our technique may be used to propose music, videos, venues, news, books, tourism, and e-commerce sites, among other things.

References

[1] Data Science & AI, Appier. “What Is a Recommendation Engine and How Does It Work?” Appier, 2020, https://www.appier.com/blog/what-is-a-recommendation-engine-and-how-does-it-work.

[2] Dey, Victor. “Collaborative Filtering vs Content-Based Filtering for Recommender Systems.” Analytics India Magazine, 24 Aug. 2021, https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/#:~:text=A%20Content%2DBased%20filtering%20model,done%20for%20Collaborative%20Filtering%20Methods.