CAPSTONE PRO JECT 2

MOVIE RECOMMENDATION SYSTEM

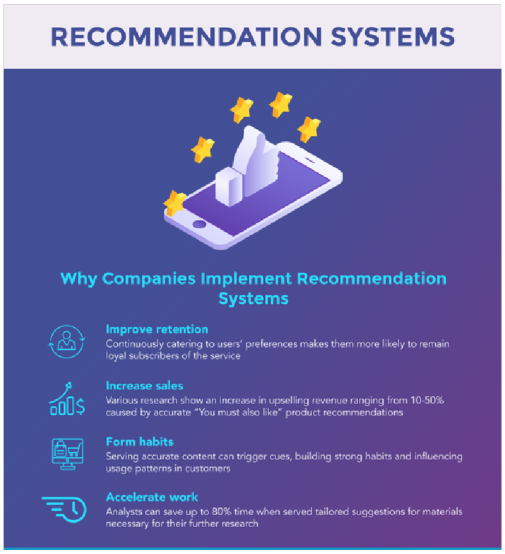
Zubair Hossain Mahamud

MDA 620

Table of Contents

Introduction:

A recommendation system basically searches for content that an individual might find interesting. In addition, it requires several factors to produce individualized lists of interesting and useful content for each user. Algorithms based on artificial intelligence are used in recommendation systems. They scan through all the options and create a personalized list of items that an individual might find interesting or useful. These outcomes are based on their profile, browsing history, what other people with similar traits and demographics are watching, and how likely they are to watch those movies. Using the available data, this is accomplished through predictive modeling and heuristics.



RECOMMENDATION SYSTEMS

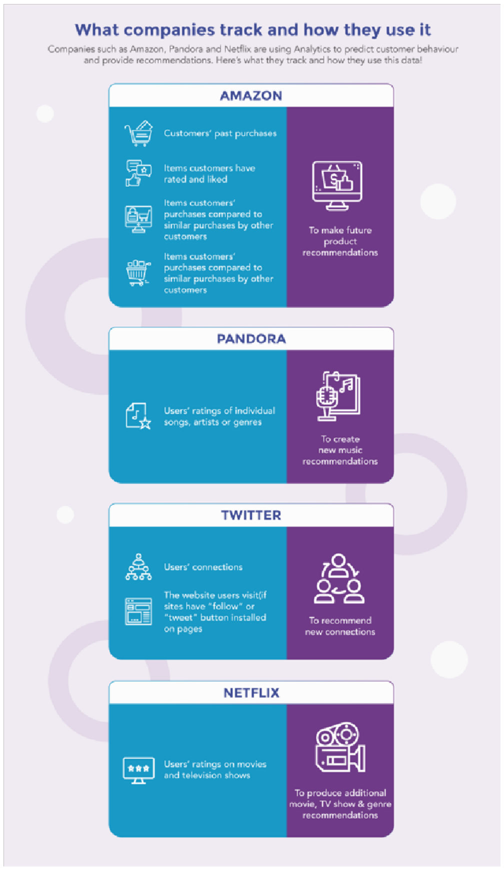
Why Companies Implement Recommendation Systems

Improve retention
Continuously catering to users' preferences makes them more likely to remain loyal subscribers of the service

Increase sales
Various research show an increase in upselling revenue ranging from 10-50% caused by accurate "You must also like" product recommendations

Form habits
Serving accurate content can trigger cues, building strong habits and influencing usage patterns in customers

Accelerate work
Analysts can save up to 80% time when served tailored suggestions for materials necessary for their further research



What companies track and how they use it

Companies such as Amazon, Pandora and Netflix are using Analytics to predict customer behaviour and provide recommendations. Here's what they track and how they use this data!

AMAZON

Customers' past purchases

Items customers have rated and liked

Items customers' purchases compared to similar purchases by other customers

Items customers' purchases compared to similar purchases by other customers

To make future product recommendations

PANDORA

Users' ratings of individual songs, artists or genres

To create new music recommendations

TWITTER

Users' connections

The website users visit(if sites have "follow" or "tweet" button installed on pages

To recommend new connections

NETFLIX

Users' ratings on movies and television shows

To produce additional movie, TV show & genre recommendations

Individuals are consistently watching out for items/benefits that are the most appropriate for them. As a result, recommendation systems are crucial because they assist them in making the right decisions without requiring them to use their cognitive resources.

Objective:

For this project I am using the Movie Database (TMDb) is a community-built database that contains a wealth of information about movies and television shows. I have utilized the TMDb 5000 dataset, a subset of this enormous dataset, for simplicity and ease of computation. It is divided into two CSV files and contains information about 5000 movies.

I will use the K-Nearest Neighbors algorithm and collaborative filtering to create a Movie Recommendation System for this project. In addition, I will compare my prediction of the movie's actual rating to that of its neighbors.

1. Movie Recommendation System using collaborative filtering by implementing the KNearest Neighbors algorithm.
2. predict the rating of the given movie based on its neighbors and compare it with the actual rating.
3. Few visualizations about directors, actors, and Genres.
4. Using the KNN model.
5. The recommendation model will be based on genres, cast, movie ratings, and directors.

Data Exploration:

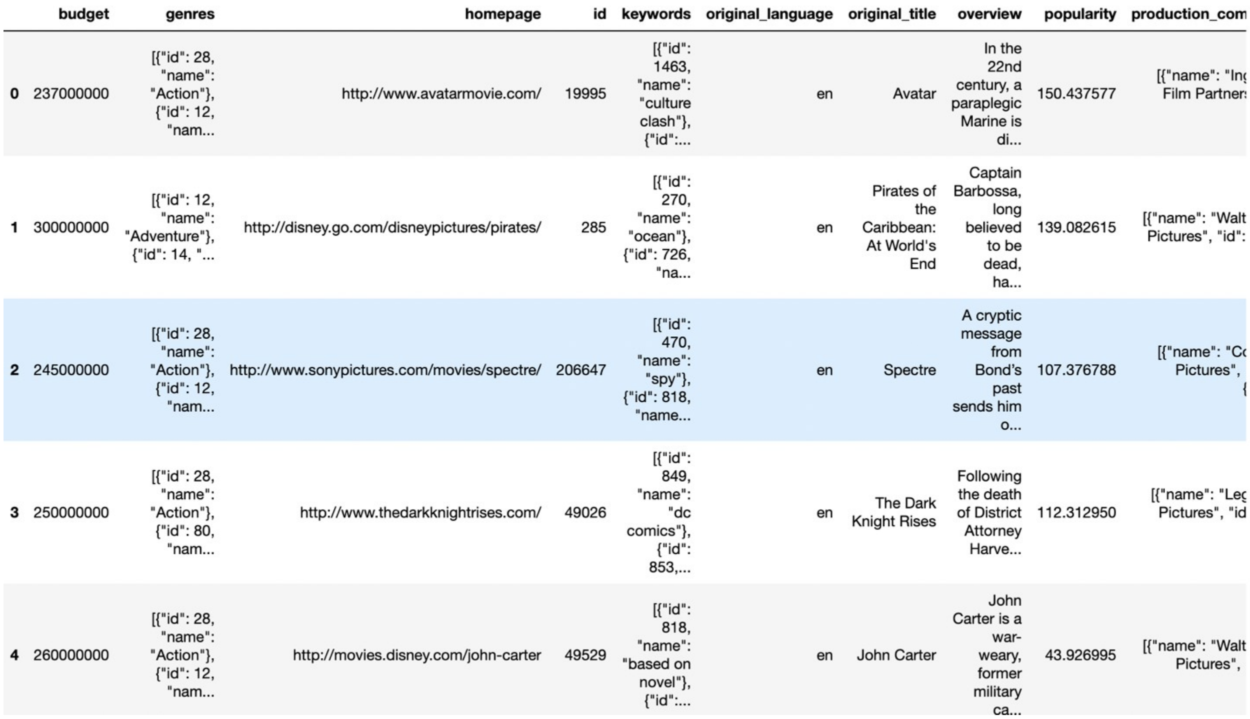
First, Import the required Python libraries like Pandas, Numpy, Seaborn and Matplotlib. Then import the CSV files using read_csv() function predefined in Pandas.. Next, we executed the head, tail, and describe commands to get all the primary information from the datasets. This includes the first and last couple of lines of the datasets, the generic material, the datatypes of each variable, and the counts of each variable. After viewing the outputs from all these snippets, we get a general idea of what the dataset includes and the type of information you’re looking at. From the exploration, you can make several inferences about the variables and how they connect to our focus on movie recommendations.

Data Manipulation:

Genres, keywords, production companies, production countries, and spoken languages are all in the JSON format when we look at the dataset. Cast and crew are also in JSON format in the other CSV file. Let's now format these columns in a way that makes them easy to read and understand. For easier interpretation, then will convert them into strings and lists later. Similar to a dictionary, the JSON format (key: value) pair within a string.

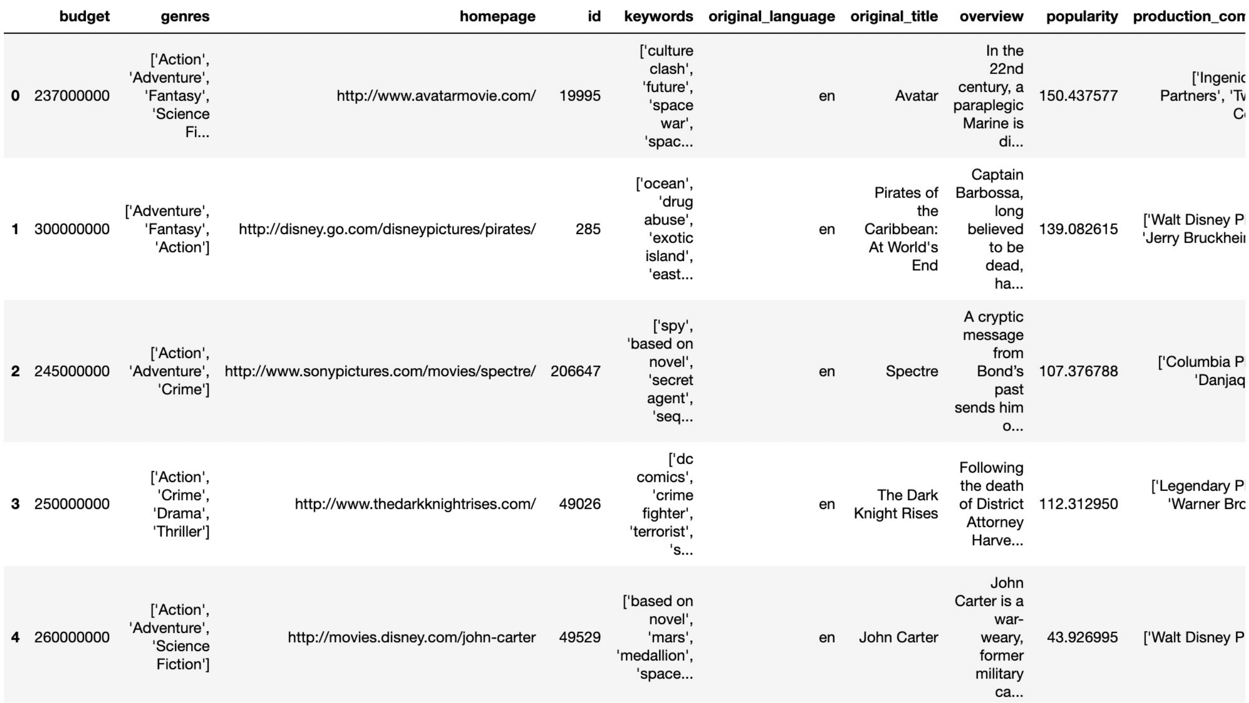
Yet, this can't be straightforwardly parsed this JSON as it must be decoded first. I have used the json.loads() method to decode it into a list for this purpose. After that, we can sort through this list to find the values we want. Then will transform the JSON into a list of column-specific strings: keywords, the cast and crew, production companies, and Using movies.iloc[index], we'll see if all of the necessary JSON columns have been converted to strings.

Working with one data frame will be an easier and more convenient way so I will merge two data files into one data frame. After combining the movies and credits data frames and selecting the necessary columns, will have a single movie data frame on which to work.



	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_com
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...	150.437577	[{"name": "Ing Film Partner
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...	139.082615	[{"name": "Walt Pictures", "id":
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[{"name": "Co Pictures", {
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	[{"name": "Leg Pictures", "id
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	[{"name": "Walt Pictures",

The data Table before changing Json format into a list



	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_com
0	237000000	['Action', 'Adventure', 'Fantasy', 'Science Fi...	http://www.avatarmovie.com/	19995	['culture clash', 'future', 'space war', 'spac...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	['Ingenic Partners', 'Tv C
1	300000000	['Adventure', 'Fantasy', 'Action']	http://disney.go.com/disneypictures/pirates/	285	['ocean', 'drug abuse', 'exotic island', 'east...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	['Walt Disney P 'Jerry Bruckheir
2	245000000	['Action', 'Adventure', 'Crime']	http://www.sonypictures.com/movies/spectre/	206647	['spy', 'based on novel', 'secret agent', 'seq...	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	['Columbia P 'Danjaq
3	250000000	['Action', 'Crime', 'Drama', 'Thriller']	http://www.thedarkknightrises.com/	49026	['dc comics', 'crime fighter', 'terrorist', 's...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	['Legendary P 'Warner Bro
4	260000000	['Action', 'Adventure', 'Science Fiction']	http://movies.disney.com/john-carter	49529	['based on novel', 'mars', 'medallion', 'space...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	['Walt Disney P

The data Table before changing Json format into a list

One hot encoding:

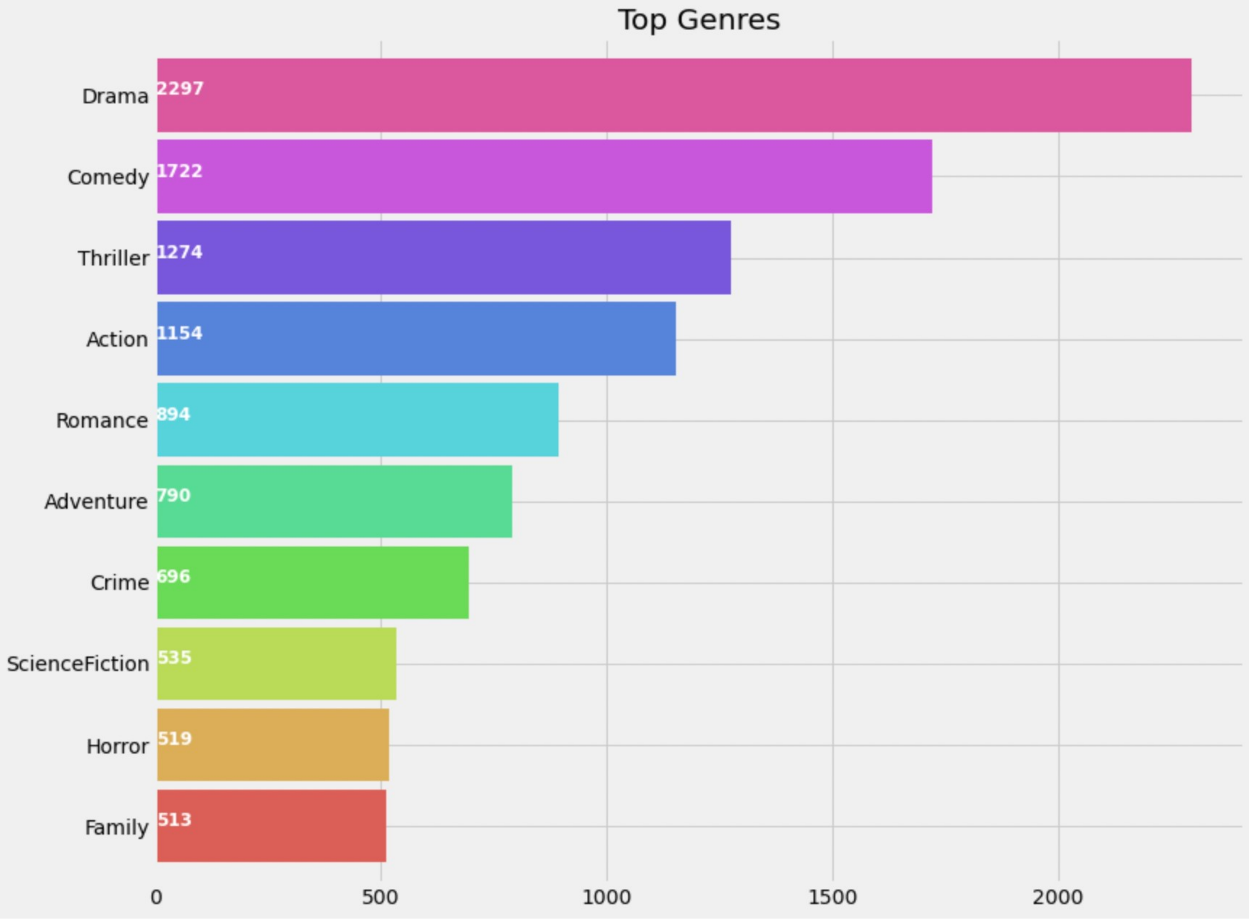
All genres will now be contained in "genreList." But how do we learn about the different movie genres? Now, some films will be called "Action," others will be called "Action, Adventure," etc. The films must be categorized according to their genres.

In the data frame, let's create a new column that will store the binary values that indicate whether a genre is present or not. First, let's come up with a method that will return a list of binary values for each movie's genre. The 'genreList' will be helpful now to look at against the qualities. Let's say, for instance, that the list includes 20 distinct genres. As a result, the function below will produce a list of 20 elements, each of which will be either 0 or 1. Now, for instance, if a movie has the genre set to "Action," the new column will contain [1,0,0,0,0,0,0,0,0,0,0,0].

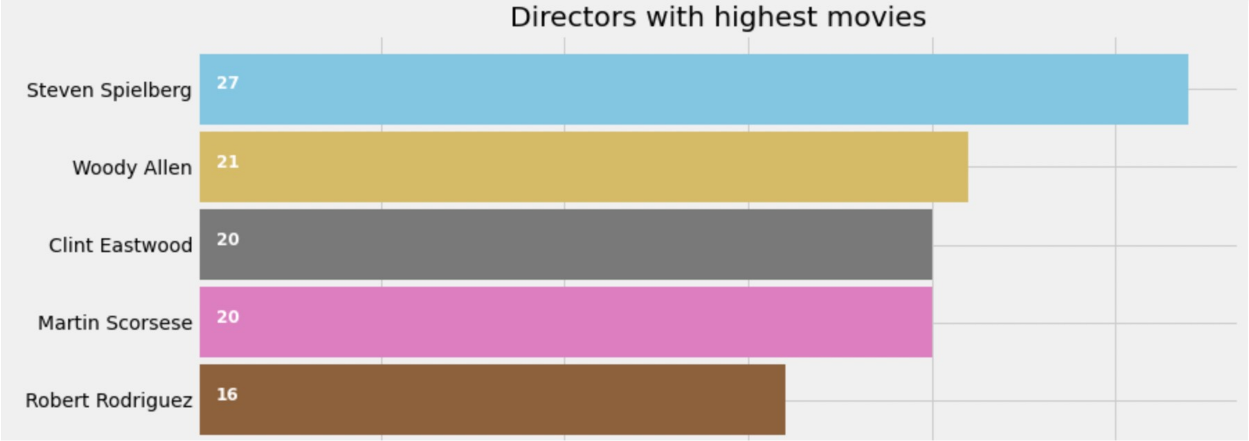
In a similar vein, we will have [1,1,0,0,0,0,0,0,0] for "Action, Adventure." It will be easier to classify movies according to their genres by converting the genres into such a list of binary values.

Applying the binary() function to the ‘genres’ column to get ‘genre_list’ then will follow the same notations for other features like the cast, director, and keywords.

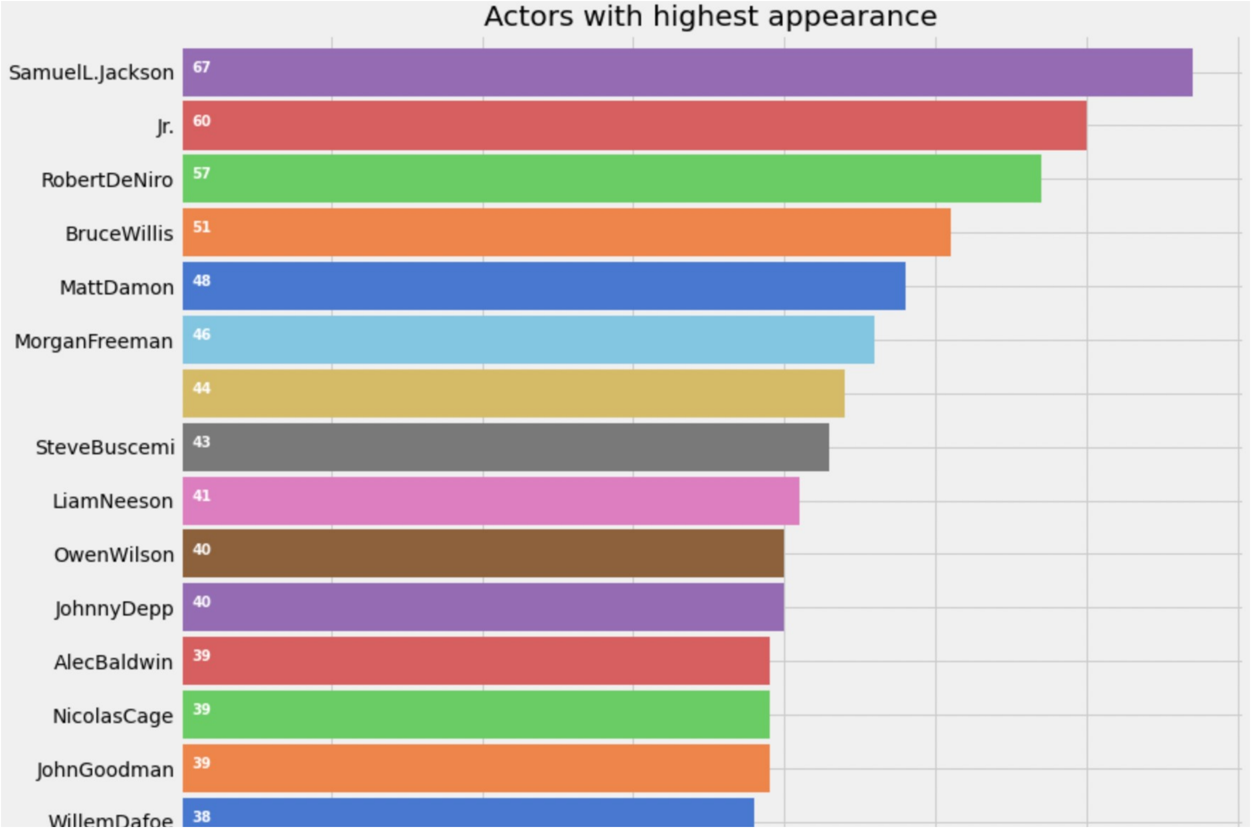
Data Visualization:



Above we can see the histogram plot for the Genre column. We can easily say that the Drama genre is the most popular one followed by comedy after that thriller and so on. The family type genre is the least popular one.



Steven Spielberg did 27 movies so far based on the data set I have used here and he holds the highest number of movies directed by any director. Woody Allen holds the second spot by directing 21 movies.

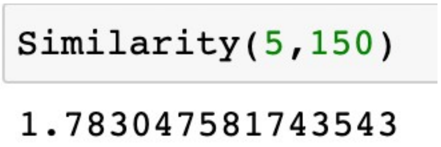


Samuel Jackson also known as Scratch Fierceness from Vindicators has showed up in greatest motion pictures. Data triumphs over presumptions! At first, I thought Morgan Freeman might be the actor with the most movies.

Due to the fact that numerous films have entries for anywhere from 15 to 20 actors, the initial list of all the cast members had approximately 50,000 unique values. The actors with the most to contribute to the film are all we need. For eg: The films in the Dark Knight series feature a large number of actors. However, we will only choose the major actors, such as Heath Ledger, Christian Bale, and Michael Caine. I've chosen the four main characters from each movie.

Similarity between movies:

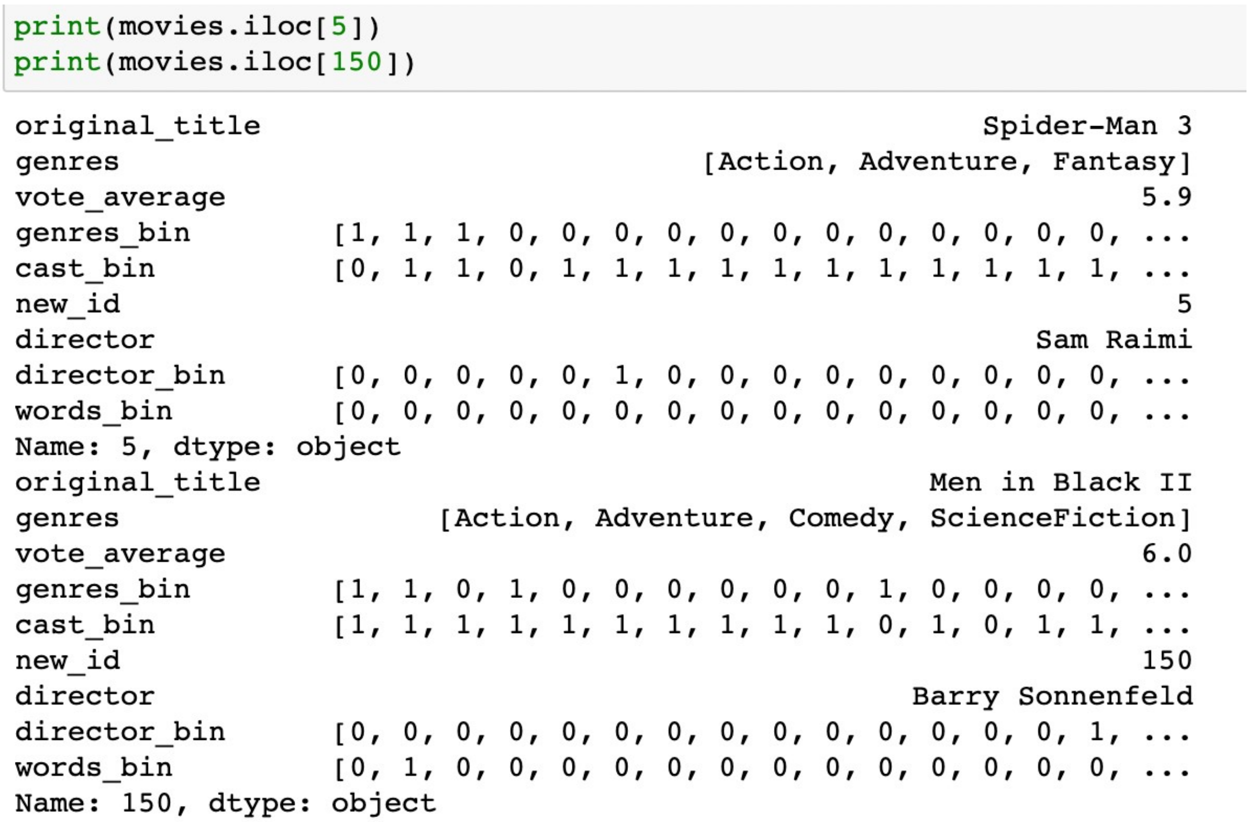
We will be using Cosine Similarity for finding the similarity between 2 movies. Lets check for two random movies from the list:



```
Similarity(5,150)
```

```
1.783047581743543
```

We can see that the distance is moderate, around 1.783. The films are less alike the further apart they are. Let's find out what these random films were about.



```
print(movies.iloc[5])
print(movies.iloc[150])
```

```
original_title                                           Spider-Man 3
genres                                  [Action, Adventure, Fantasy]
vote_average                                                      5.9
genres_bin        [1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
cast_bin          [0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
new_id                                                              5
director                                                    Sam Raimi
director_bin      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
words_bin         [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
Name: 5, dtype: object
original_title                                        Men in Black II
genres                     [Action, Adventure, Comedy, ScienceFiction]
vote_average                                                      6.0
genres_bin        [1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...
cast_bin          [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, ...
new_id                                                            150
director                                             Barry Sonnenfeld
director_bin      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ...
words_bin         [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
Name: 150, dtype: object
```

It is evident that Spider-Man 3 and Men in Black 2 your Dragon 2 are different movies. Thus, the movies do have similarities when it comes to genres that’s why the distance was moderate.

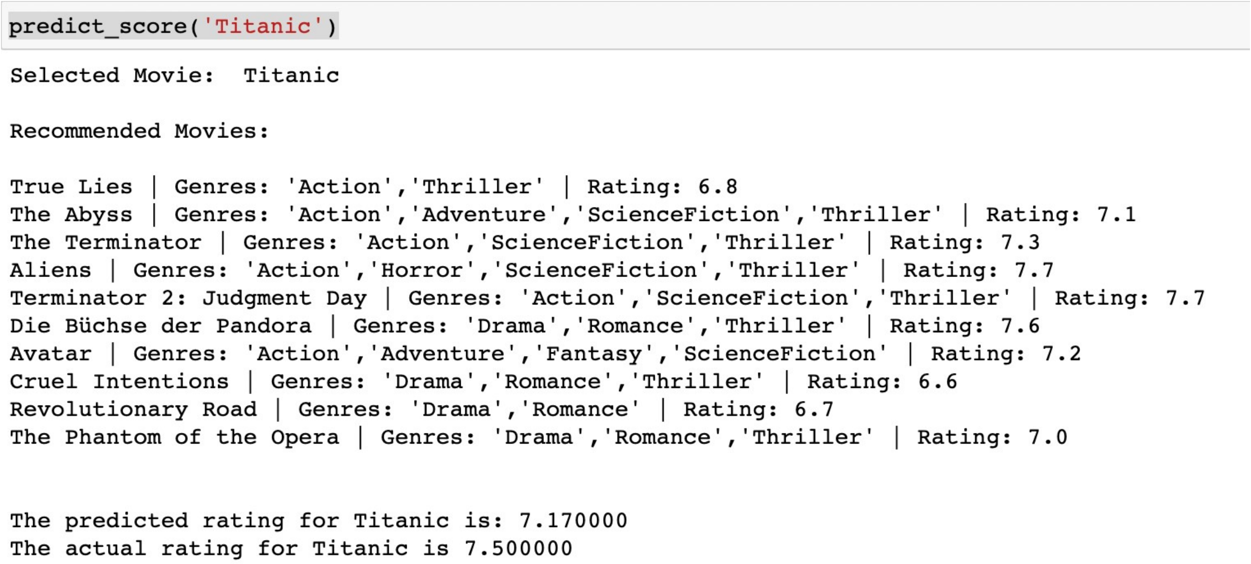
Movie Recommendation:

The classification and regression supervised machine learning algorithm are K-Nearest Neighbors. It modifies the training data and uses distance metrics to classify the new test data. It predicts the k closest neighbors. In this project, I have arbitrarily chosen the value K=10.

The Similarity() function, which determines the 10 movies that are most similar to each other and calculates their similarity, will be the main under-the-hood function. These ten films will assist us in predicting the movie's score. To determine the desired movie's score, we will average the scores of similar films.

Presently the closeness between the motion pictures will rely upon our recently made segments containing paired records. We are aware that aspects such as the cast and director will play a significant role in the movie's success. We always assume that films directed by Chris Nolan or David Fincher will do well. Additionally, their chances of success are even higher if they collaborate with their favorite actors, who consistently bring them success, and if they work in their preferred genres.

Now simply just run the function predict_score and enter the movie of choice to find 10 similar movies and it’s predicted ratings

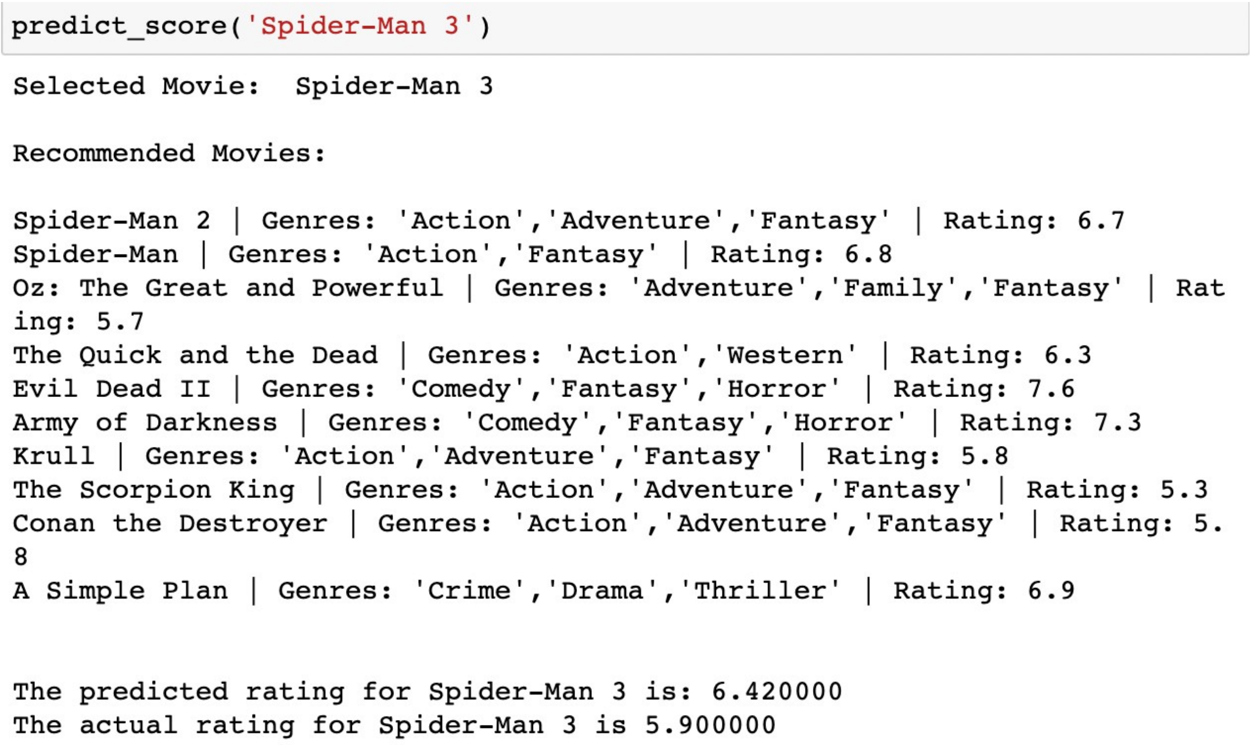


```
predict_score('Titanic')
```

```
Selected Movie:  Titanic

Recommended Movies:

True Lies | Genres: 'Action','Thriller' | Rating: 6.8
The Abyss | Genres: 'Action','Adventure','ScienceFiction','Thriller' | Rating: 7.1
The Terminator | Genres: 'Action','ScienceFiction','Thriller' | Rating: 7.3
Aliens | Genres: 'Action','Horror','ScienceFiction','Thriller' | Rating: 7.7
Terminator 2: Judgment Day | Genres: 'Action','ScienceFiction','Thriller' | Rating: 7.7
Die Büchse der Pandora | Genres: 'Drama','Romance','Thriller' | Rating: 7.6
Avatar | Genres: 'Action','Adventure','Fantasy','ScienceFiction' | Rating: 7.2
Cruel Intentions | Genres: 'Drama','Romance','Thriller' | Rating: 6.6
Revolutionary Road | Genres: 'Drama','Romance' | Rating: 6.7
The Phantom of the Opera | Genres: 'Drama','Romance','Thriller' | Rating: 7.0


The predicted rating for Titanic is: 7.170000
The actual rating for Titanic is 7.500000
```

After running the predict score function with the movie Titanic we can see the following recommended movies. These recommendations are based on the genre, rating, director, and cast. The predicted score is 7.17 and the actual rating was 7.5. The recommended rating is fairly close.

This one is another example of the recommendation model:



```
predict_score('Spider-Man 3')
```

```
Selected Movie:  Spider-Man 3

Recommended Movies:

Spider-Man 2 | Genres: 'Action','Adventure','Fantasy' | Rating: 6.7
Spider-Man | Genres: 'Action','Fantasy' | Rating: 6.8
Oz: The Great and Powerful | Genres: 'Adventure','Family','Fantasy' | Rat
ing: 5.7
The Quick and the Dead | Genres: 'Action','Western' | Rating: 6.3
Evil Dead II | Genres: 'Comedy','Fantasy','Horror' | Rating: 7.6
Army of Darkness | Genres: 'Comedy','Fantasy','Horror' | Rating: 7.3
Krull | Genres: 'Action','Adventure','Fantasy' | Rating: 5.8
The Scorpion King | Genres: 'Action','Adventure','Fantasy' | Rating: 5.3
Conan the Destroyer | Genres: 'Action','Adventure','Fantasy' | Rating: 5.
8
A Simple Plan | Genres: 'Crime','Drama','Thriller' | Rating: 6.9


The predicted rating for Spider-Man 3 is: 6.420000
The actual rating for Spider-Man 3 is 5.900000
```

Conclusion:

Recommender systems are quickly becoming an essential component of online e-commerce. The enormous volume of user data stored in existing corporate databases is putting a strain on recommender systems, and the growing volume of user data on the Internet will put even more strain on them. Recommender systems' scalability can be significantly improved by new technologies.

The Movie Recommendation System uses the Cosine Similarity algorithm to recommend the best movies that are related to the movie entered by the user based on a variety of factors, including the movie's genre, summary, cast, and ratings Thus the model was completed by the implementation of using the K Nearest Neighbors algorithm.

Bibliography:

https://www.mygreatlearning.com/blog/masterclass-on-movie-recommendation-system/#a3
https://www.kaggle.com/code/deepak525/investigate-tmdb-movie-dataset