Project 1: Milestone 3 – White Paper

DSC680

Rohan Valder

Topic

Movie Recommendation – Recommend ten movies to the user based on the reviews received.

Business Problem

As in Netflix, we receive recommendations of movies to be watched based on the list of movies watched in the past; I plan to analyze the reviews and recommend movies for future watch. The movies are ranked considering the average of all the reviews for a movie, and the top 10 of them, are suggested to be watched. Recommend ten movies to the user based on the reviews received.

Background

Nowadays, there are so many movies on Netflix and other OTT platforms. It is not easy to decide what movie to watch when you have plenty. Based on the history watched and the reviews, it will be easy to list the top movies, and then you can pick the movie of your choice from the top ten. The ten movies differ from user to user as the watching history would be different, which indirectly would rely on the categories, and the highest rating from that category will be shortlisted.

Data Explanation

These files contain the metadata for 45,000 movies listed in the full Movie Lens dataset. The dataset consists of movies released on or before July 2017. Data points include cast, crew, production companies, budget, revenue, languages, countries, TMDB votes, vote averages, plot keywords, posters, and release dates. The dataset contains a file with 26 million ratings from 270,000 users for the 45,000 movies. Ratings are on a scale from 1 to 5 and are obtained from the official Group Lens Website.

The dataset consists of the following files:

Movies_metadata.csv – The main movies dataset file. Contains information about 45,000 movies in the Full MovieLens dataset. Features include posters, budgets, companies, backdrops, production countries, release dates, revenue, and languages.

Keywords.csv – Contains the movie plot keywords for MovieLens movies. Available in the form of a stringified JSON object.

Credits.csv – Contains the cast and crew information for all our movies.

Links.csv – Contains TMDB and IMDB ID of the movies in the MovieLens movies.

Link_small.csv – Contains TMDB and IMDB ID of the small set of 9,000 movies in the dataset.

Ratings_small.csv – The subset of 100,000 ratings by 700 users on 9,000 movies.

The Full MovieLens dataset consists of 26 million ratings from 270,000 users for 45,000 movies.

The data can be downloaded here - https://grouplens.org/datasets/movielens/latest/

Methods

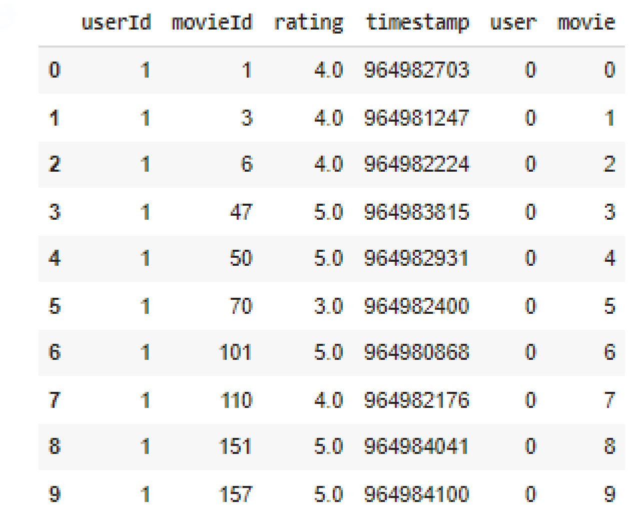
This project will be using the collaborative filtering to build the movie recommender system. Collaborative filtering is a unique technique of the recommender system. If you want to build the recommender system, you will likely use this technique for the lack of large, detailed data. Collaborative filtering has two senses: A narrow sense and a general sense. For the narrow sense, the collaborative filter, will automatically predict based on the interest of the user. This happens because the preferences and the details are taken from other users, hence the term collaborating in the name.

Content based filtering takes a different approach when compared to collaborative filtering. It relies on the description of certain items and user's preference rather than the similarly of the items of the same group of users. This raises a good case to use content-based filtering. When the data about the item not the user is known such as the name, location and feature, content-based filtering is the best choice. It treats this problem as classification for specific users. It can learn a classifier on what the user likes and dislikes, based on the feature of specific items. This approach is perhaps more accurate when more data is given. However, for the sake of building a basic movie recommender system, collaborative filtering is a clear superior choice.

Analysis

As part of the development, collect the data, preprocess it, analyze it before building and training the machine learning model. This project will use Matplotlib library for simplicity because it does not need extensive exploratory data analysis for its depth. The dataset can be loaded in a dataframe in a few minutes before they can be cleaned up. Most real-world data need to be preprocessed before it can be explored, analyzed, and fed into the machine learning algorithm. This step allows the researchers to design and determine what are the relevant features that helps in prediction. In addition, the researchers can tweak the model with different parameters to achieve the highest possible accuracy score.

Rows with missing data will be removed. After examining the data, the userID and the movieID can be converted into additional columns as User and Movie. The reason for conversion is we need to encode the users and movies to indices. Later, the users and movies column will be fed into the machine learning algorithm. Below are the first ten rows of the dataframe after preprocessing:



	userId	movieId	rating	timestamp	user	movie
0	1	1	4.0	964982703	0	0
1	1	3	4.0	964981247	0	1
2	1	6	4.0	964982224	0	2
3	1	47	5.0	964983815	0	3
4	1	50	5.0	964982931	0	4
5	1	70	3.0	964982400	0	5
6	1	101	5.0	964980868	0	6
7	1	110	4.0	964982176	0	7
8	1	151	5.0	964984041	0	8
9	1	157	5.0	964984100	0	9

Figure 1. The first ten rows of the data after being randomized.

Conclusion

The focus of this project is to come up with the top ten recommendations for the movies. The reason is that we have a lot of movies to watch these days because of the pandemic. More time is available, and more movies are listed. If I were to watch a movie, I would randomly pick a movie that I see in front of me, and think is interesting. I would give a glance at the trailer and finalize the movie to watch. But when I have recommendations, I would pick one of them from the recommendations since they would match my taste, as well as the viewers, have liked them. With that in mind, movie recommendations will be of great help to me especially when there are ten of them. The ten recommendations have been different as per the user. Hence recommendations are expected to be liked by all.

Assumptions

The data science application is not fancy and complicated as other search engine data science projects. Its main goal is to understand and implement the basic movie recommendation system. There are many movies recommender system. All the movies have been rated. Movies not categorized are more likely not recommended. The researchers go ahead with the assumptions that if a user A and user B have a preference on a certain topic, they will have same preference on some other topic. On the other hand, the general sense will allow collaborative filtering to use different data sources, viewpoints in collaboration to obtain patterns.

Limitations

The cleaning up of the data is the most time-consuming processes.

Challenges/Issues

With the techniques used in the past, I may have to refresh it to be familiar with it again. I'm thinking if I have a tie based on the average rating, how do I handle it.

Future Uses/Additional Applications

In the future, application can be developed for TV shows.

Recommendations

Although it is not implemented in the current system, low movie ratings can be filtered, and only high-rated movies can be recommended. This will improve the user experience with the system. However, it can be biased, as low rating movies would be recommended, and some people want to watch such movies.

Implementation Plan

Many programming languages can be used to build the movie recommender system, python is the easiest to implement. Python provides the most user-friendly environment to build the data and machine learning algorithms using its rich module libraries such as TensorFlow, keras, and pytorch; Pandas is also an added advantage to prepare the data for the machine learning algorithms compared to other programming languages. In addition, python has a very rich and diverse data visualization libraries such as matplotlib, plotly and seaborn. These data visualization libraries are very convenient to learn more about the data.

The data for this project was collected and organized by group lens research which was a computer science research lab in 1997. The file of the dataset can be directly downloaded from the Group Lens Website. The rating csv one of the downloaded files will be loaded into the dataframe using pandas. The data will be explored and examined to determine the relevant information. The data preprocessing steps involve checking the number of users and how many ratings the users have made. It is important to have all users with the same number of ratings. From the dataset the minimum rating is

0.5 and the maximum rating is 5.

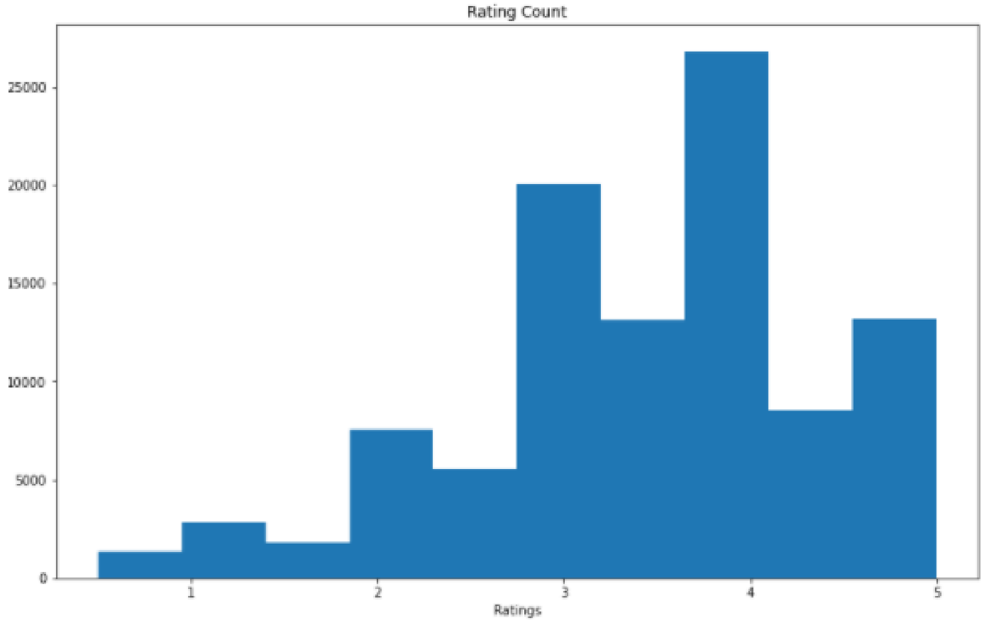
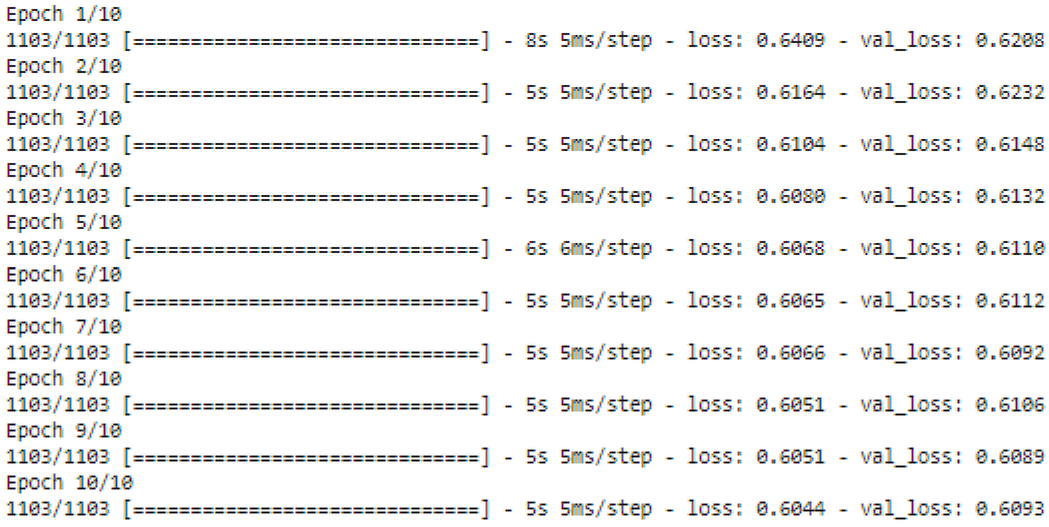


Figure 2. Rating counts of movie reviews

Figure 2 shows it is left-skewed, and there are more high rating counts than low rating counts. This suggests that most people feel neutral about the movie or good about the movie. This is a positive signal for building a movie recommender system as the recommended movies would be enjoyed by the people.

70% of the data is used for training and 30% is used for validation. The features of the model are the values of users and movies. To build the model, the model function of the library Keras is used. The movie recommender function is created which has three inputs – the number of users, the number of movies, and the number of vector dimensions. Once every parameter is set in place, the training process can take place. Below is the image of the training process:



```
Epoch 1/10
1103/1103 [==============================] - 8s 5ms/step - loss: 0.6409 - val_loss: 0.6208
Epoch 2/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6164 - val_loss: 0.6232
Epoch 3/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6104 - val_loss: 0.6148
Epoch 4/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6080 - val_loss: 0.6132
Epoch 5/10
1103/1103 [==============================] - 6s 6ms/step - loss: 0.6068 - val_loss: 0.6110
Epoch 6/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6065 - val_loss: 0.6112
Epoch 7/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6066 - val_loss: 0.6092
Epoch 8/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6051 - val_loss: 0.6106
Epoch 9/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6051 - val_loss: 0.6089
Epoch 10/10
1103/1103 [==============================] - 5s 5ms/step - loss: 0.6044 - val_loss: 0.6093
```

Figure 3. The training process of the building the machine learning model.

Figure 3 shows there are 10 epochs for the training. This means there 10 passes of the entire dataset the machine learning has performed.

The image below shows after the model has been trained.

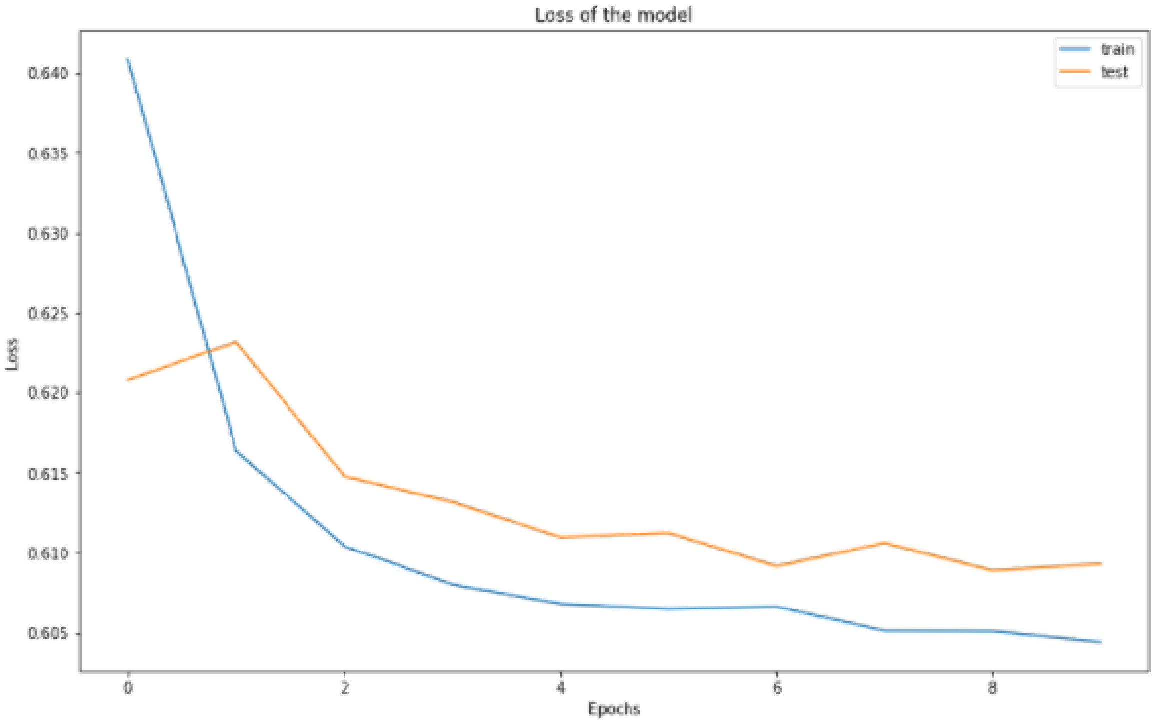
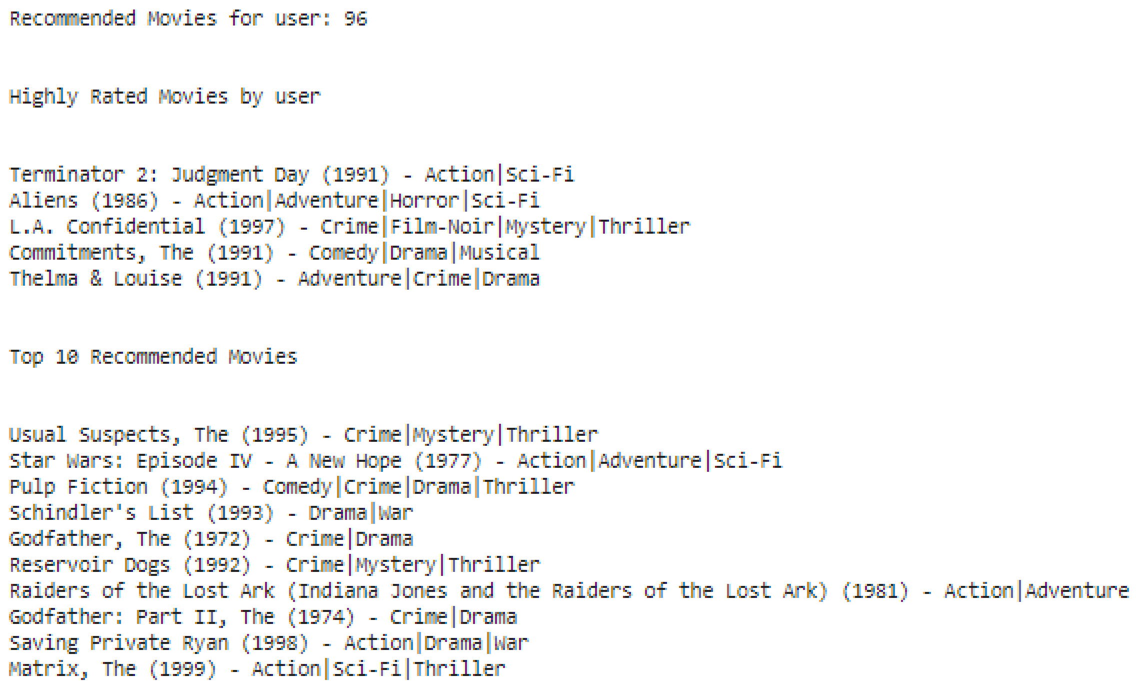


Figure 4. The graphs of the training and test data after fitting.

It is typical to see a drop in the training data because that is the place the model is used to learn. The loss is a bigger concern for the test data as the lower the loss better is the learning process. While the training loss continues to drop after 10 epochs the test loss forms a plateau.

Once the model function has been put, the predict function can be used to perform recommendations of the movies. Ten movies will be recommended for any random user in the database. Below is an example of a random user receiving recommendations based on their preferences.



```
Recommended Movies for user: 96


Highly Rated Movies by user


Terminator 2: Judgment Day (1991) - Action|Sci-Fi
Aliens (1986) - Action|Adventure|Horror|Sci-Fi
L.A. Confidential (1997) - Crime|Film-Noir|Mystery|Thriller
Commitments, The (1991) - Comedy|Drama|Musical
Thelma & Louise (1991) - Adventure|Crime|Drama


Top 10 Recommended Movies


Usual Suspects, The (1995) - Crime|Mystery|Thriller
Star Wars: Episode IV - A New Hope (1977) - Action|Adventure|Sci-Fi
Pulp Fiction (1994) - Comedy|Crime|Drama|Thriller
Schindler's List (1993) - Drama|War
Godfather, The (1972) - Crime|Drama
Reservoir Dogs (1992) - Crime|Mystery|Thriller
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981) - Action|Adventure
Godfather: Part II, The (1974) - Crime|Drama
Saving Private Ryan (1998) - Action|Drama|War
Matrix, The (1999) - Action|Sci-Fi|Thriller
```

Figure 5. Ten recommended movies for a random user.

The user's index is 96. Based on the 5 highly rated movies by the user, 10 movies are recommended.

Recommender system is turned out to be successful to the business with a competitive edge.

Ethical Assessment

Nowadays, my wife and I are watching the top 10 movies/shows on Netflix. They have been considered, top 10 based on the views and ratings I suppose. We like it. We do not spend much time deciding what we will be watching next as it is easy now with the top list available. I want to come up with such a list for the users.

References

Nurdialit, Dwi Gustin (2020, September). Netflix Movies and TV Shows — Exploratory Data Analysis (EDA) and Visualization Using Python. - https://medium.com/analytics-vidhya/netflix-movies-and-tvshows-exploratory-data-analysis-eda-and-visualization-using-python-80753fcfcf7

Pradhan, Karan (2021, July). Visualizing Netflix Data Using Python! - https://www.analyticsvidhya.com/blog/2021/07/visualizing-netflix-data-using-python/

Ramos, Leodanis Pozo (2022, Feb). Python's zipfile: Manipulate Your ZIP Files Efficiently - https://realpython.com/python-zipfile/

Ebrahim, Mokhtar (2022, Feb). Python time module (Simple Examples) - https://likegeeks.com/python-time-module/

10 Questions

1) What was the reason for selecting this topic?
   - I have been using the Netflix recommendation for a long time. I liked the feature and thought it might be a good topic for the project.

2) Why didn’t you choose tv shows instead of movies?
   - The dataset had only movies. If the tv shows were part of the dataset, I might have included the tv shows in the recommendations.

3) Why did you focus on the number 10 for movie recommendations?
   - As mentioned earlier, my influence is Netflix, and Netflix has the top ten recommendations. Ten is ideal; it is not too much nor too less to select a movie.

4) What was the reason for the selected data set?
   - I usually prefer data from Kaggle. The dataset had a good number of movies and the desired ratings.

5) Did you have to make any adjustments to the dataset for the insights?
   - I had to join multiple dataframes to get the data together to have a better insight into the movies and the ratings.

6) Do all the movies have the same number of ratings?
   - No, every movie has a different number of ratings.

7) What are you going to do with the insights you found?
   - I can make use of it in real-world scenarios. The movies could be replaced, with anything else too.

8) Are there any areas of the project you would like to improve?
   - More visuals and tags can be made used for the recommended movies.

9) How are the moviemakers going to benefit from this project?
   - Moviemakers will be able to see the success of their movies. Also, the top ten movies, will be given the moviemakers an idea about the kind of movies the audience like the most.

10) How do we evaluate if the project has succeeded?
    - If the project can execute the business problem without many issues and the expected output is as desired.

Appendix

Figure 1. The first ten rows of the data after being randomized.

Figure 2. Rating counts of movie reviews.

Figure 3. The training process of the building the machine learning model.

Figure 4. The graphs of the training and test data after fitting.

Figure 5. Ten recommended movies for a random user.