ISyE 6740 – Spring 2021
Project Proposal (or Final Report)

Team Member Names: Jin Ren, Cong Hu

Project Title: Movie Recommendation System

Problem Statement

The COVID-19 pandemic and subsequent lockdowns increased movie consumption for those who were forced to stay at home. Many sought a distraction from the grim reality, giving companies like Netflix a huge boost in subscribers. Nevertheless Netflix was not the only company to benefit from this increase in demand. Thus started streaming wars amongst companies like Disney, Paramount, HBO, etc.

To stay competitive in this field, streaming companies must offer the best content and make it as easy as possible for viewers to find and watch the movies that they like, giving customers a reason to stay subscribed to the service. Companies like Spotify excel in song recommendation, which is why people continue to subscribe even if there are cheaper and more convenient options. Therefore having a great movie recommendation system is vital for any streaming company. Movie recommendation is an extraordinarily complex and interesting problem because there are a variety of genres and two people could like the same movie for varied reasons. The number of movies available to watch is also large and ever-growing which makes it a type of big data problem.

Data Source

The dataset for this project contains two csv files: 'tmdb_5000_movies.csv' which has approximately 4800 rows(movies) and 20 columns and 'tmdb_5000_credits.csv' which has approximately 4800 rows and 4 columns. Kaggle removed the original IMDB version of this dataset per a DMCA takedown request from IMDB and replaced it with a similar set of movies from The Movie Database (TMDb) in accordance with the terms of use. The data source is obtained from Kaggle: https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata

TMDB is a popular movie review website that can be edited by users. People from all over the world and of all ages submit their reviews of movies on TMDB. This data source was selected for its high usability score of 8.2 on Kaggle and the amount of information that can be mined from it. It is also not behind a paywall like some companies' data, where one must pay for the subscription service to submit a review (ex. Netflix). Psychologically people will rate things they paid for higher than they would have if they had gotten that item or service for free. Also, the viewership base for Netflix may be different compared to Disney+ or Paramount+, which have more children-friendly content.

Methodology

Recommendation systems can be classified into demographic filtering, content-based filtering, and collaboration-based filtering recommendation systems. In this project, we will use content-based filtering and collaboration-based filtering method to do the movie recommendations.

Our first step is to do data cleaning and exploration. The initial data includes two csv files, we will combine them into one table and convert the JSON format to strings. We will also deal with missing values by removing them. One-hot encoding was done for the 'genres' and 'words' variables. First a list of the unique genres was created: 'Action', 'Adventure', 'Fantasy', 'ScienceFiction', 'Crime', 'Drama', 'Thriller', 'Animation', 'Family', and 'Western'. For each genre we created a binary variable to indicate whether a movie was in that genre. Same approach was used for 'words' variable.

Next, we will do data analysis on those columns (variables). For content-based, a function was defined to calculate the pairwise similarity between one movie and another based on "overview" column. The pairwise similarity can be calculated based on cosine similarity. Based on similarity score, we can sort movies and recommend top n movies when the user inputs a movie name. For collaborative-based filtering, we focused on item-item collaborative filtering. Several columns are used to calculate the similarity between movies based on several variables: genres, score (rating), director, and (relevant) words. Two methods were used to calculate the similarity: Cosine and Jaccard [1]. Sklearn was used to calculate cosine similarity and Scipy was used to calculate Jaccard similarity.

If the vectors are close to being parallel (angle close to zero), they are similar. If they are orthogonal (angle close to 90), they are independent. The formula for cosine similarity and Jaccard similarity are given below:

$$cosine\ similarity(A,B) = \frac{A\ \cdot B}{\left||A|\right| \times\ ||B||} = \sum_{i=1}^{n} \frac{A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

$$Jaccard\ similarity(A,B) = \frac{|A\ \cap B|}{|A \cup B|} = \frac{|A\ \cap B|}{|A| + |B| - |A\ \cap B|}$$

The larger the distance, the less similar the two movies are. After we get the results of similarity, we can implement KNN algorithm to do the movie recommendation. After tuning, $k = 10$ was chosen. A lower k means that the results will be affected by noise more and a bigger k will be more computationally expensive. Typically, an odd number is chosen if the number of classes is two. Another common way of selecting k is by using $k = \sqrt{n}$.

Evaluation and Final Results

I. Content-based filtering

In this recommendation system, the column 'overview' is used to find pairwise similarity score between one movie and the other. In calculating the similarity score, TF-IDF and cosine similarity are used in this system. Finally, the recommendation system will do the recommendation based on the similarity score.

The first row and second row in column ‘overview’ is given as below:

'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization.'
'Captain Barbossa, long believed to be dead, has come back to life and is headed to the edge of the Earth with Will Turner and Elizabeth Swann. But nothing is quite as it seems.'

The data in this column are sentences, we need to find out the most important words in these sentences and then to calculate the pairwise similarity. Our model is based on TF-IDF, which is a numerical statistics method to calculate the importance of a word to a document in a collection [2]. And we used Python built-in TfIdfVectorizer from scikit-learn package, which can help us convert the raw data to a matrix of TF-IDF features. The formula for TF-IDF is given below:

$$TF(term\ frequency) = \frac{term\ instances}{total\ instances}$$

$$IDF(inverse\ document\ frequency) = \ log\frac{number\ of\ documents}{documents\ with\ term}$$

$$overall\ importance = \ TF * IDF$$

Given TF-IDF vectors of over 2000 words, we generate the cosine similarity matrix for vectors by using Python built-in linear_kernel from sklearn-learn package.

After we get the pairwise similarity matrix, our recommendation system can give top five movies based on your inputs. Some results are shown below:

Input:	‘Harry Potter and the Half-Blood Prince’
Outputs:	Harry Potter and the Goblet of Fire Harry Potter and the Order of the Phoenix Harry Potter and the Prisoner of Azkaban Harry Potter and the Chamber of Secrets The Little Prince
Input:	‘Iron Man
Outputs:	Iron Man 2 Iron Man 3 Cradle 2 the Grave Avengers: Age of Ultron Hostage
Input:	‘Kung Fu Panda’
Outputs:	Kung Fu Panda 2 Legend of a Rabbit Kung Fu Hustle Bulletproof Monk Kung Pow: Enter the Fist

Input:	‘Titanic’
Outputs:	Raise the Titanic Ghost Ship I Can Do Bad All By Myself Event Horizon Niagara

On the one hand, the results seem very good, the recommendation system gives us very similar movies; On the other hand, we can see from ‘Titanic’ results, people may just want to see the actor Leonardo DiCaprio’s movie rather than disaster films. Let’s see if several variables are used to do the recommendation on collaborative-based filtering.

II. Collaborative-based filtering

Before the recommendation algorithm was implemented there were some basic visualizations done to get a general understanding of the data. Below are three bar graphs:

Fig.1: Table of Most Frequent Directors

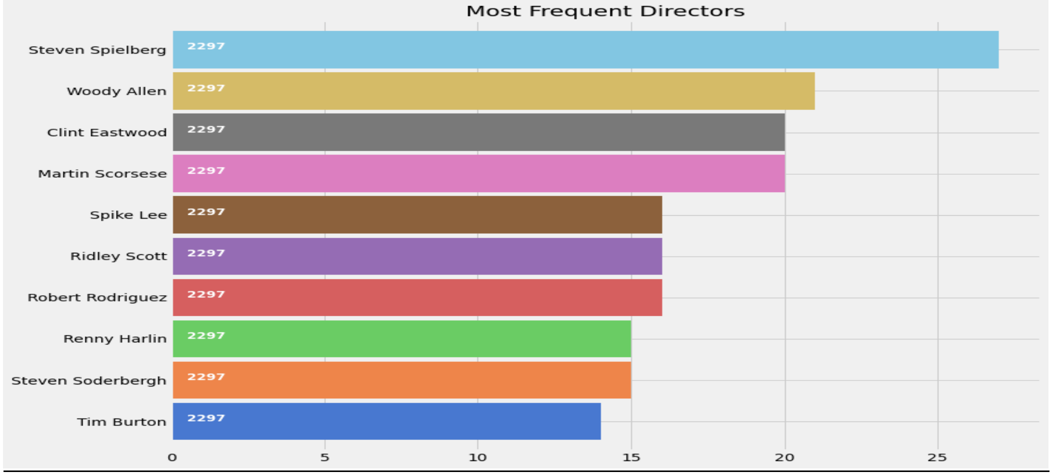


Fig.2: Table of Most Frequent Actors

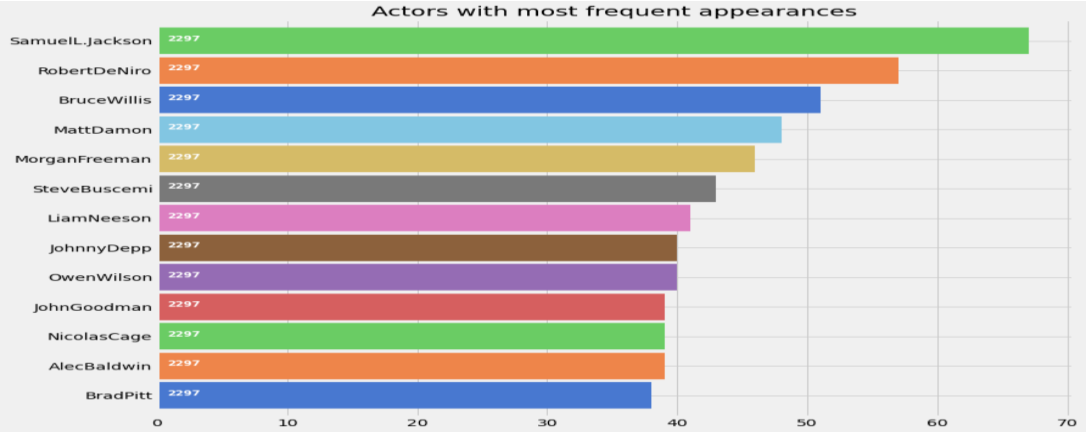
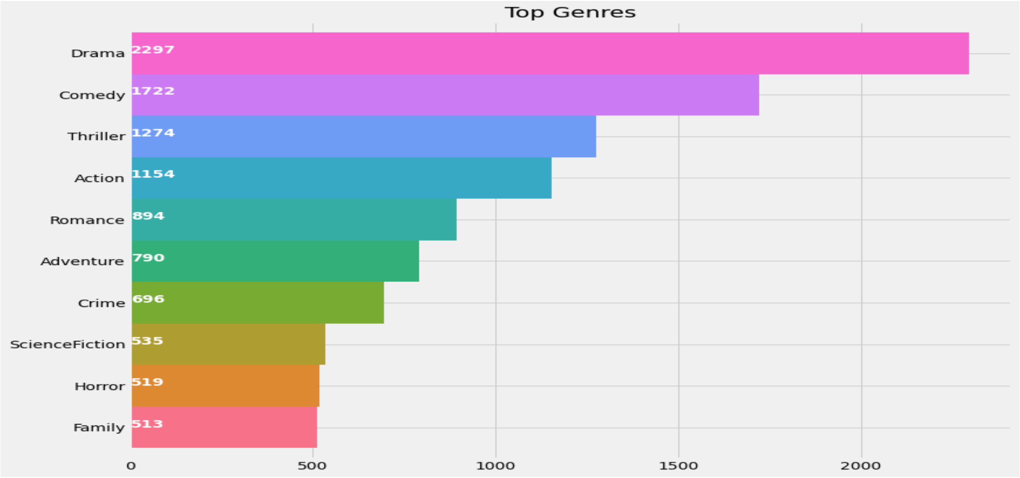


Fig.3: Table of Most Frequent Genres



The top three most prolific directors are Steven Spielberg, Woody Allen, and Clint Eastwood. The top three most frequent actors are Samuel L. Jackson, Robert DeNiro, and Bruce Willis. The top three most popular genres are dramas, comedies, and thrillers.

The results from cosine and Jaccard similarity are also compared in the table below:

Table 1. Predicted and Actual Ratings using Cosine & Jaccard Similarities for "The Godfather: Part III"

Godfather Part III	**Cosine**			**Jaccard**		
Rating 7.1	**Recommended Movie**	**Predicted Rating**	**Actual Rating**	**Recommended Movie**	**Predicted Rating**	**Actual Rating**
	The Rainmaker	6.7	6.7	The Rainmaker	6.7	6.7
	The Godfather: Part II	8.97	8.3	The Godfather: Part II	8.97	8.3
	The Godfather	9.3	8.4	The Godfather	9.3	8.4
	The Outsiders	7.83	6.9	The Outsiders	7.83	6.9
	The Cotton Club	7.38	6.6	Dead Man's Shoes	7.98	7.2
	End of Watch	7.94	7.2	The Conversation	8.3	7.5
	Only God Forgives	6.39	5.6	The Cotton Club	7.43	6.6
	Hannibal Rising	6.64	6	End of Watch	7.94	7.2
	Savages	6.86	6.2	Only God Forgives	6.39	5.6
	Donnie Brasco	8.09	7.4	Hannibal Rising	6.64	6

Table 2. Predicted and Actual Ratings using Cosine & Jaccard Similarities for “500 Days of Summer”

(500) Days of Summer	**Cosine**			**Jaccard**		
Rating 7.2	**Recommended Movie**	**Predicted Rating**	**Actual Rating**	**Recommended Movie**	**Predicted Rating**	**Actual Rating**
	Don Jon	5.9	5.9	Don Jon	5.9	5.9
	The Good Girl	6.69	6.1	The Good Girl	6.69	6.1
	Chocolat	7.47	6.8	Chocolat	7.47	6.8
	Shall We Dance?	6.65	5.9	Shall We Dance?	6.65	5.9
	Youth in Revolt	6.56	5.9	Youth in Revolt	6.56	5.9
	Enough Said	7.26	6.6	Enough Said	7.26	6.6
	Trust the Man	6.23	5.5	What to Expect When You're Expecting	6.53	5.8
	What to Expect When You're Expecting	6.42	5.8	Trust the Man	6.15	5.5
	Georgia Rule	6.24	5.6	My Beautiful Laundrette	7.22	6.6
	Secretary	7.32	6.7	Georgia Rule	6.32	5.6

Table 3. Predicted and Actual Ratings using Cosine & Jaccard Similarities for “Harry Potter and the Half-Blood Prince”

Harry Potter and the Half-Blood Prince	**Cosine**			**Jaccard**		
Rating 7.4	**Recommended Movie**	**Predicted Rating**	**Actual Rating**	**Recommended Movie**	**Predicted Rating**	**Actual Rating**
	Harry Potter and the Order of the Phoenix	7.4	7.4	Harry Potter and the Order of the Phoenix	7.4	7.4
	Harry Potter and the Goblet of Fire	8.24	7.5	Harry Potter and the Goblet of Fire	8.24	7.5
	Harry Potter and the Prisoner of Azkaban	8.52	7.7	Harry Potter and the Prisoner of Azkaban	8.52	7.7
	Harry Potter and the Philosopher's Stone	8.35	7.5	Harry Potter and the Philosopher's Stone	8.35	7.5
	Harry Potter and the Chamber of Secrets	8.24	7.4	Harry Potter and the Chamber of Secrets	8.24	7.4
	Oz: The Great and Powerful	6.52	5.7	Oz: The Great and Powerful	6.52	5.7
	Inkheart	6.65	6	Inkheart	6.65	6

	Percy Jackson: Sea of Monsters	6.57	5.9	Percy Jackson: Sea of Monsters	6.57	5.9
	The Indian in the Cupboard	6.56	5.9	The Indian in the Cupboard	6.56	5.9
	Pan	6.56	5.9	Pan	6.56	5.9

Table 4. Predicted and Actual Ratings using Cosine & Jaccard Similarities for "Memoirs of a Geisha"

Memoirs of a Geisha	**Cosine**			**Jaccard**		
Rating 7.3	**Recommended Movie**	**Predicted Rating**	**Actual Rating**	**Recommended Movie**	**Predicted Rating**	**Actual Rating**
	Nine	5.1	5.1	Nine	5.1	5.1
	Anna and the King	6.91	6.4	Anna and the King	6.91	6.4
	Cinderella Man	7.99	7.3	Kama Sutra	6.39	5.7
	Quo Vadis	7.8	7	Cinderella Man	7.94	7.3
	三城记	7.08	6.3	Quo Vadis	7.79	7
	Elizabeth: The Golden Age	7.31	6.6	The Best Years of Our Lives	8.38	7.6
	The Duchess	7.43	6.7	三城记	7.14	6.3
	The New World	7.14	6.4	Elizabeth: The Golden Age	7.31	6.6
	Aimee & Jaguar	7.01	6.3	The Duchess	7.43	6.7
	The Scarlet Letter	6.2	5.5	The New World	7.14	6.4

Table 5. Predicted and Actual Ratings using Cosine & Jaccard Similarities for "Despicable Me 2"

Despicable Me 2	**Cosine**			**Jaccard**		
Rating 7	**Recommended Movie**	**Predicted Rating**	**Actual Rating**	**Recommended Movie**	**Predicted Rating**	**Actual Rating**
	Despicable Me	7.1	7	Despicable Me	7.1	7
	Monsters, Inc.	8.21	7.5	Monsters, Inc.	8.21	7.5
	Cloudy with a Chance of Meatballs 2	7.22	6.4	Cloudy with a Chance of Meatballs 2	7.22	6.4
	Over the Hedge	7.02	6.3	Over the Hedge	7.02	6.3
	Hotel Transylvania 2	7.4	6.7	Hotel Transylvania 2	7.4	6.7
	Looney Tunes: Back in Action	6.34	5.6	Looney Tunes: Back in Action	6.34	5.6
	The Simpsons Movie	7.53	6.9	Barnyard	5.93	5.3
	Barnyard	6.05	5.3	The Simpsons Movie	7.49	6.9
	Chicken Little	6.21	5.6	Chicken Little	6.35	5.6
	Hop	6.12	5.5	Doug's 1st Movie	6.03	5.4

It appears that the predicted ratings are within ±1 of the actual ratings for each recommended movie for both cosine and Jaccard similarities. Also, for all five movies selected, the recommendations suggested by both cosine & Jaccard similarity are mostly the same, but towards the bottom of the table there might be one or two movies that are different. Overall, both similarity measures give decent movie recommendations. For example, Despicable Me 2 is a family-friendly movie that is funny and caters towards children. Monsters, Inc. and Cloudy with a Chance of Meatballs are also family-friendly movies that cater towards children. If the movie is part of a series, the other movies in that series will be recommended by both similarity measures, as seen for the Harry Potter, Despicable Me, and The Godfather series. For foreign movies like Memoirs of a Geisha, other foreign and historical-fiction films were recommended, which seem like good matches. This recommendation system seems robust enough to handle a variety of movies, which is important because as stated earlier, movies are very diverse. The algorithm also considers multiple factors, which is important because two people can like the same movies for different reasons.

Conclusions & Future Research

The recommendation algorithms are decent, but they can still be improved. Latest research shows that a hybrid method using both content and collaborative-based approaches can be more effective than either of the two alone. To improve on the results, the two recommendation systems could be combined to form a hybrid system.

A company that demonstrates the power of hybrid systems is Netflix, which currently has one of the best recommendation systems. They compare watching and searching habits of alike users (collaborative filtering) in addition to suggesting movies that share similarities with movies that a user has liked (content-based filtering).

However, if all factors are equal, the simplest model will be the one selected for usage. At most companies, management will usually prefer the simplest model if there is not a significant difference in the results. Simpler models take less time to set-up and implement, which also saves the company money. It appears that even though Netflix's hybrid system is more complex, the improvement in recommendation quality is enough to justify the added implementation costs. Recommendation algorithms will continue to be an exciting and popular field that evolves and improves upon itself.

References

[1] https://medium.com/analytics-vidhya/introduction-to-similarity-metrics-a882361c9be4

[2] https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/