Movie Recommendation System

Julianne Lin and Ian McSpiritt

Introduction

As society's lives become more and more digital, algorithms heavily influence much of the decisions that are made on a daily basis. From in-person shopping experiences to binging your favorite show on a streaming service, the decisions that users make are shaped by a plethora of machine learning algorithms. These algorithms leverage vast amounts of user data to effectively train for their intended purpose, which help industries gain insights, influence decisions, drive business, and retain customers. For any industry, effective implementation of machine learning models can drive business goals.

The video streaming industry is an industry that has heavily leveraged the effectiveness of machine learning in their business models. Content streaming platforms utilize powerful algorithms to influence user engagement, retention, and recommendation. These platforms have been entrenched in the streaming market harbouring vast user bases. With the utilization of these powerful tools, it has streamlined the user experience on the platform allowing for more time consuming the end product and less time browsing. Machine learning tools are driving this industry allowing expansion of content, features, and interactivity.

"In 2020, more than 500 hours of video were uploaded every minute." (Ceci, 2021) This statistic exemplifies the sheer volume of content to choose from and in turn creates a challenge of how this content will be delivered to end users. With this volume of content to choose from, it is paramount that these platforms are providing pipelines of entertainment that users will enjoy. The utilization of strong recommendation systems allow platforms such as YouTube to suggest content that is catered to their users' habits, based on a variety of attributes.

In addition to consumable videos such as YouTube, we have seen entire industries shift to digital platforms, such as the feature length movie industry. Leaders in the space are Netflix, Amazon Prime Video, and Hulu. We are also seeing the emergence of industry giants such as HBO through their HBOMax platform and NBC through their Peacock streaming platform. With a similar challenge to face as Youtube, these movie platforms are utilizing highly developed

recommender algorithms to gain market advantage in the space. Users are ever increasing time spent consuming movie content on these platforms, especially over the past two years during the COVID-19 pandemic. With users at home spending more time on platforms, the velocity of this data has increased tremendously. This velocity has allowed leaders in the space to iterate on their recommendation designs and tune their algorithms to consumer preferences. In the end, we are seeing huge shifts in how consumers are watching movies - moving from the box office to the comfort of their own couch.

In order to explore these technologically advanced recommender systems, we decided to develop our own, utilizing industry standard content data from the leading aggregators of movie information. With volume being paramount in design, we obtained content data from IMDb and user generated data from TMDb. In addition to the content-based approach, we were able to obtain sample user activity from MovieLens, which is a gold-standard dataset to test the effectiveness of our recommender systems. By cleaning and merging all three of these datasets in Python, it allowed us to implement a k-nearest neighbor clustering algorithm that classifies movie attributes using cosine similarity and feed our classifications based on that user's rating activity. Ultimately, our goal was to explore how content attributes and user activity influence machine learning recommendation systems in suggesting movies to existing users. Our analysis of the effectiveness utilized RMSE and analyzed F1 scores of the predictions compared to the user's top rated movies. This model provided predictions and allowed us to dip our feet into what is the massive ocean of recommendation systems.

Motivation

Consumer research suggests that a typical streaming platform user will lose interest after 60 to 90 seconds of deciding before selecting a movie title or exiting the platform altogether (Gomez-Uribe & Hunt, 2016). With this attrition potentially being a matter of seconds, it is crucial for recommendation algorithms to provide succinct suggestions of similar titles to retain that user. Nearly every popular streaming platform of movies uses unique algorithms to keep their users engaged and subscribed. With so many options of platforms available these streaming services are competing for market share. To provide a competitive advantage and differentiate in the market, recommendation systems are of the utmost importance.

Platforms that host movies are in a constant state of competition, vying for user base supremacy. Many of these platforms have even heavily invested in their own production studios, such as Netflix and Amazon Prime Video. In addition to producing their own original content, the cost for streaming rights of existing licensing is enormous. Netflix planned to spend over $17 billion in 2021 on content alone, based on their first quarter earnings report (Low, 2021). With platforms heavily investing in the content that is hosted, it is crucial that they have underlying data on what content users are watching and what attributes surrounding this content are important.

We are currently in the age of influence, and consumers are bombarded with so many choices of content that curation of consumption can set apart a platform's utility to the market. Across social media platforms, content is being suggested by influencers, algorithms, and critics. With this change in user behavior, there is an expectation that a platform provides a curated experience. Curating a user's experience not only retains them on the platform but allows them to advocate new content to their own social network. An effective recommendation system not only influences the user's decision but exposes them to new content that can potentially be shared by that user, expanding the platform's user-base umbrella.

With user retention impact, costly content budgets, and consumer choice behavior shifting, this leaves platforms with the underlying challenge of creating a product that combats these challenges. While strong content can be the differentiator, it is important not to overlook how that content is delivered to the end user. An effective recommender system not only caters to users preferences but creates an ecosystem that the user can't live without.

Literature Review

In preparing for this study, literature review was conducted. Some of the articles we found included "Comparative Analysis of Clustering Techniques for Movie Recommendation" (Aditya et al., 2018), "Feature Selection for Movie Recommendation" (ÇATALTEPE et al., 2016), "Movie Recommendation System Using Clustering Mining with Python" (Sadhasivam et al., 2021), "Precomputed Clustering for Movie Recommendation System in Real Time" (Li et al., 2014), "Movies Recommendation System Using Collaborative Filtering and k-means"

(Phorasim & Yu, 2017), and "Prediction of Movies Popularity Using Machine Learning Techniques" (Latif & Afzal, 2016). Of these articles, the first three were reviewed in depth.

The first paper to be reviewed was "Comparative Analysis of Clustering Techniques for Movie Recommendation" (Aditya et al., 2018). They planned to feed TMDb movie datasets from Kaggle into Euclidean distance based algorithms to cluster them based on specific features. They first preprocessed the data by loading and removing irrelevant data by using the sci-kit learn, matplotlib, and pandas python libraries. This study used a variety of different clustering techniques, including agglomerative, BIRCH, k-means, and mean shift. In their results, they found that k-means clustering worked well, but the output clusters differ each time the algorithm was run since it begins with a random choice of cluster, whereas hierarchical clustering has reproducible results in each run. K-means clustering also requires a pre-hand knowledge of the number of clusters to divide the data into. Overall, k-means clustering performed well, but some of the other algorithms are able to outperform in specific areas.

The second paper was "Feature Selection for Movie Recommendation" (ÇATALTEPE et al., 2016). In the system they were using, users do not explicitly rate movies with a like or dislike or a discrete numerical rating, so they had to calculate implicit ratings by using the viewing durations. They used collaborative recommendation to predict the rating for instances where the user has not seen a certain movie by basing it on the user's neighbors' opinions. During evaluation, the training set was used to produce the implicit ratings. Each recommender would run for a set number of days, then the average would be calculated to compute the performance of the recommender and also evaluated with precision, recall, and f-measure metrics. From their findings, content-based recommendations based only on actor, director, and keyword features outperformed the collaborative method. The content-based recommender based only on actor features gave the best results, and genre and year-based recommendations performed significantly worse. In the end, the combination of director and feature selected actor and keyword resulted in the best performance in terms of precision.

The last paper we reviewed in depth was "Precomputed Clustering for Movie Recommendation System in Real Time" (Sadhasivam et al., 2021). This study aimed to create a simple recommendation analysis which would use the notion that a popular and critically acclaimed movie would be more likely chosen by the audience. For this, they used the IMDb

rating algorithm. They used content based filtering for clustering similar movies, like sequels and franchises, and then used collaborative filtering to calculate the error rate. Since they found that latency is a key issue in these systems, they used a combination of category based and user-based approaches for scalability. For evaluation metrics, they used the mean square error root. In their conclusion, they found that the recommendation methods are robust on their own but combining them helped reduce disadvantages.

Overall, we noticed that many of these studies used k-means clustering with Euclidean distance, precision and mean square error root for evaluation metrics, and a combination of multiple filtering techniques, which we plan to incorporate into our study.

Approach

Our data was sourced from a multitude of industry standard sources. The first one being IMDb in which we were able to obtain a comprehensive table including attributes such as ID, Title, Title type, isAdult, Year, run-time, and Genres. We also sourced data from TMDb, which is a crowd sourced version of IMDb that aggregates user tags to different movies. The data from TMDb was very similar to IMDb, but we ended up using it as a source for our keywords tags and cast, as the IMDb dataset did not have these. Our last data source was from MovieLens, which gave us our user activity. This source listed particular users and the movies that they have watched, as well as their ratings. With all three of these datasets, we had a plethora of attributes to utilize for testing our recommender system.

The first challenge in using all three of these datasets was to be able to merge and link them up so that we could leverage the attributes we needed. In order to do this, we ended up utilizing a links file that was given in the MovieLens data package. This file contained IMDb, TMDb, and MovieLens identifiers. However, this file had some issues, as the IMDb ID was missing an importing string prior to the numerical ID. We created a function to add the predecessor “tt” and any missing 0s, and tested it against the IMDb data. This would ultimately allow us to pull attributes from any source data and read it into the MovieLens user data.

In addition to merging the data based on the links file, we needed to clean IMDb data, as the set included tv shows, adult movies, and other unnecessary data. The data in the IMDb genre

column were also strings and not lists containing each genre type. We proceeded to create functions to fix these for the attributes that we were planning to test. At this point, we also performed some exploratory analysis to visualize the genre attribute, which we planned on using in our recommender.

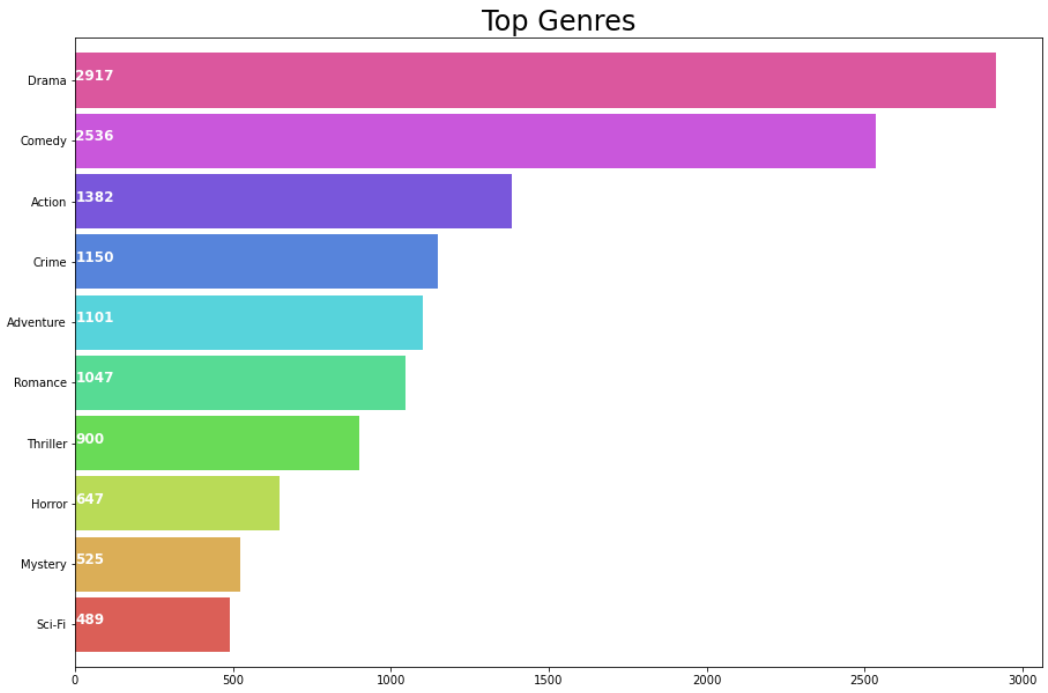


Figure 1: Top Genres

This visualization helped give us an idea of the distribution of genre types and how movies were assigned.

We also had to perform cleaning and exploratory analysis on the TMDb dataset. This included transforming similar string to list approaches for the attributes we wanted to test. These attributes included cast, keywords, and director. Intuitively, we found these attributes to be important, as users tend to watch movies with their favorite actors and from their favorite directors. We also explored keywords, due to TMDb being user generated, and were interested in the results. We were able to visualize our exploration of the cast and director attributes by showing the top ten actors with the highest appearances and the top ten directors with the highest appearances.

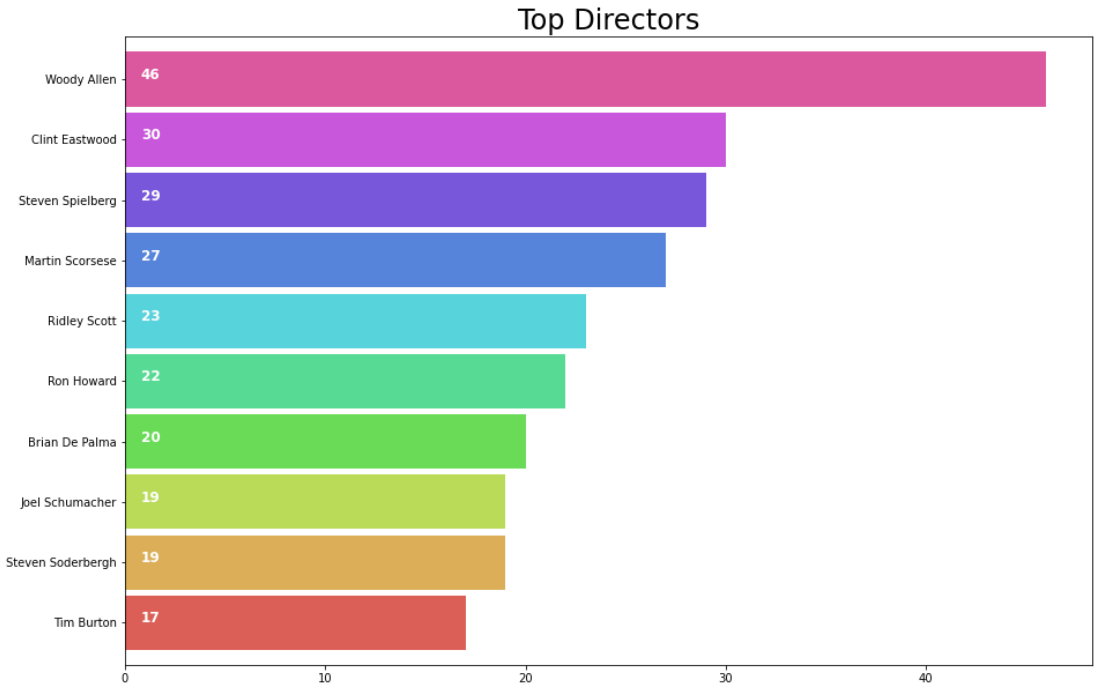


Figure 2: Top Directors

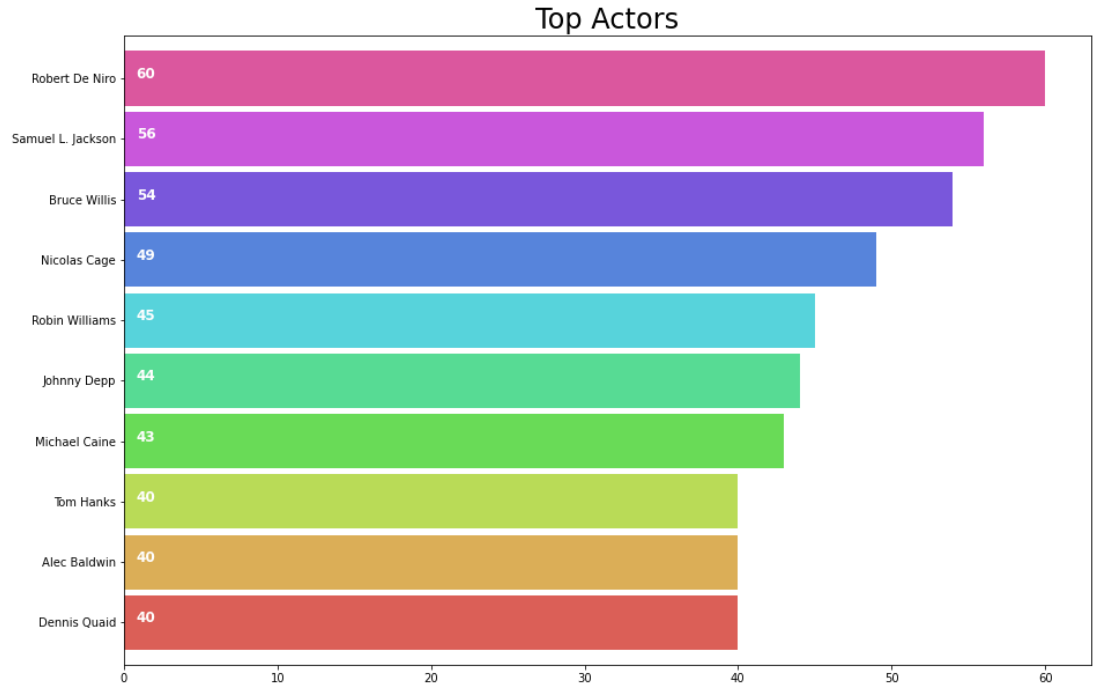


Figure 3: Top Actors

One attribute that was unique to the TMDb dataset were keywords and, as mentioned before, were user generated from their open source webpage, which allows users to rate and tag movies. We found the data to be interesting and visualized it in a word cloud.

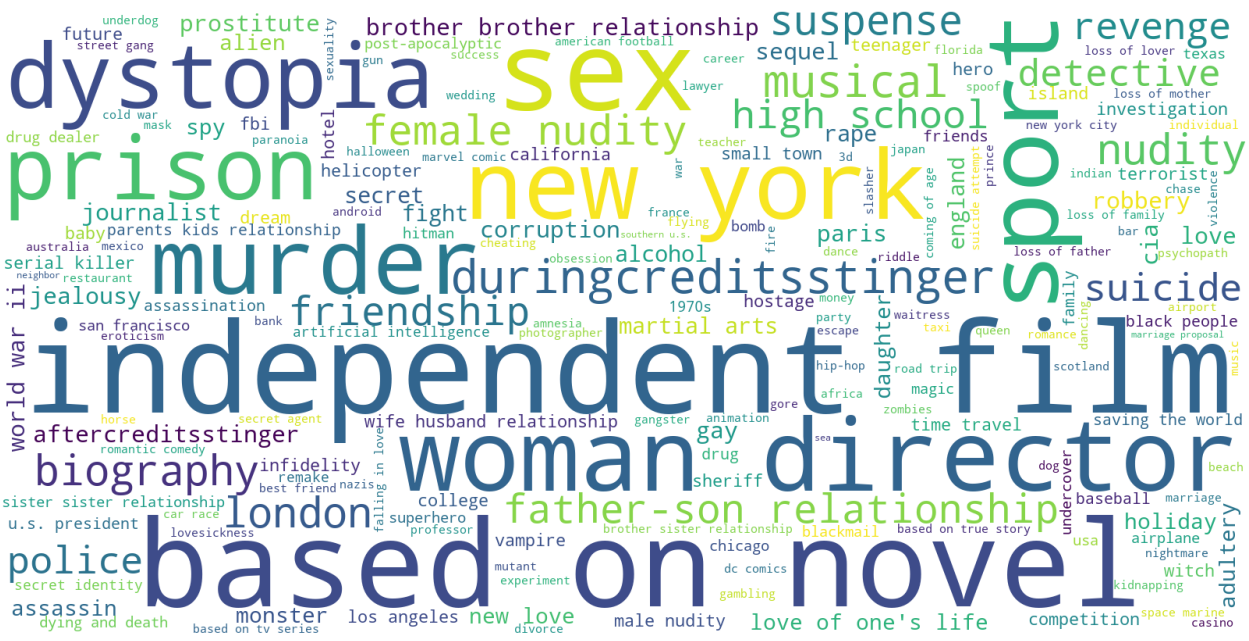


Figure 4: Keywords word cloud

By visualizing this into a word cloud, we were able to get an idea of what types of movies were included in the TMDb dataset

The MovieLens data contained user activity with corresponding IDs containing each user's watched movies as well as their ratings of each. This dataset contained over 10,000 instances of user ratings, with each user having at least 20 ratings. We would end up leveraging this data to train our model on top rated movies by each individual user. This set also had movie titles that contained years that we eventually dropped to merge in IMDb titles in order to compare accurate titles by a subject matter expert organization.

With datasets pulled from three different sources, it was important to merge them carefully. We first merged the IMDb and TMDb data together, ensuring that only movies they had in common were in the merged set. Then, we merged the merged set with the MovieLens links dataset, ensuring that only movies they had in common were in this second merged set. Lastly, we filtered the ratings dataset to ensure that the ratings we would test on would only contain movies that had all of the IMDb and TMDb attributes we needed.

Now with three datasets cleaned properly to feed into a model, we moved on to our attribute selection phase. We decided to take an intuitive approach by selecting features that would be considered in a user's decision making process to watch a movie, as well as features that would help find the most accurate similarity between movies. The attributes selected were genres, cast, director, and keywords. We felt these attributes would provide us with accurate results in our approach. After merging the data and dropping the unnecessary columns, we were left with a final and cleaned dataset.

Using this final dataset, we decided to take a content-based approach by exploring each attribute and making recommendations based on each individual movie's features. In order to do this approach, we had to transform our categorical features to be utilized in a classification algorithm. We ended up transforming these features binarily, using a multilabel binarizer function. This allowed us to numerically convert the classes in each feature to feed into the model. By having the attributes in this binary format, we were able to implement a k-nearest neighbors approach to classification. By feeding in a user's top-rated movies, we classified our multiple attributes on these top-rated movies to recommend similar movies based on each individual movie's attributes. We did this classification manually by calculating the cosine similarity between each attribute selected effectively comparing them to the top rated movies. From here, we output the 10 nearest neighbors from the entire merged dataset with the lowest cosine similarity to the users' top rated movies. This approach was simple in nature and utilized an instance based learning approach by using the entire dataset of movie attributes, outputting movies within the same neighborhood. The attributes we selected were vectorized to get this done. We also calculated the RMSE between each to give us another metric of comparison with cosine similarity. This approach was heuristic in nature by selecting how many top rated movies to compare and the number of recommended movies to output.

Results

Our final results consisted of 10 recommendations based on 5 of the user's top rated movies. These 10 were selected by sorting the aggregated cosine similarity of the content features of the recommended movies. By training on a small sample of the top rated movies, it gave us predictions that had fairly average similarity scores. We had to dial back our input of

training data since the computation of classifying and iterating through our instances of movies took a long time. We can see in the table below that while RMSE was accurate the cosine similarity was lackluster.

	IMDbID	similarity	RMSE
4932	tt0424345	0.216667	0.008196
3829	tt0261392	0.300000	0.010529
11514	tt0104694	0.483333	0.078746
12212	tt0117372	0.500000	0.014638
6040	tt1007028	0.523223	0.059516
11872	tt0110971	0.545876	0.061062
2900	tt0118842	0.555662	0.078746
2610	tt0113749	0.573223	0.060121
12953	tt0200027	0.583333	0.080291
38327	tt0077766	0.595876	0.062843

Table 1: 10 recommended movies with similarity and RMSE

Here, shown in Table 1, we have an output that contains the 10 recommended movies with the corresponding aggregated cosine similarities of each attribute. We also included RMSE to give another metric of comparison between the predicted 10 to the 5 trained movies.

	IMDbID	title	genres	cast	director	keywords
0	tt0109445	Clerks	[Comedy]	[Brian O'Halloran, Jeff Anderson, Jason Mewes,...	[Kevin Smith]	[salesclerk, loser, aftercreditsstinger]
1	tt0094737	Big	[Comedy, Drama, Fantasy]	[Tom Hanks, Elizabeth Perkins, Robert Loggia, ...	[Penny Marshall]	[baseball, co-worker, bronx, pinball machine, ...
2	tt0080761	Friday the 13th	[Horror, Mystery, Thriller]	[Betsy Palmer, Adrienne King, Harry Crosby, La...	[Sean S. Cunningham]	[drowning, lake, cabin, cult, revenge]
3	tt0113118	Friday	[Comedy, Drama]	[Ice Cube, Chris Tucker, Nia Long, Tommy 'Tiny...	[F. Gary Gray]	[rap, parents kids relationship, rapper, job]
4	tt0073195	Jaws	[Adventure, Thriller]	[Roy Scheider, Robert Shaw, Richard Dreyfuss, ...	[Steven Spielberg]	[fishing, atlantic ocean, bathing, shipwreck, ...

Table 2: User 225's top 5 rated movies

When evaluating the training set more qualitatively, we can see similarities in the movies that this user rated. Table 2 shows the user's top 5 rated movies, which were the movies that were used for creating recommendations. Most of their top rated movies consisted of comedy and drama genres.

	IMDbID	title	genres	cast	director	keywords
0	tt0424345	Clerks II	[Comedy]	[Brian O'Halloran, Jeff Anderson, Jason Mewes,...	[Kevin Smith]	[independent film, aftercreditsstinger, during...
1	tt0261392	Jay and Silent Bob Strike Back	[Comedy]	[Kevin Smith, Jason Mewes, Ben Affleck, Jeff A...	[Kevin Smith]	[film making, jay and silent bob, self mocking...
2	tt0104694	A League of Their Own	[Comedy, Drama, Sport]	[Tom Hanks, Geena Davis, Madonna, Lori Petty, ...	[Penny Marshall]	[baseball, world war ii, sport, baseball playe...
3	tt0117372	The Preacher's Wife	[Comedy, Drama, Fantasy]	[Denzel Washington, Whitney Houston, Courtney ...	[Penny Marshall]	[angel, church choir, gospel, reverend, crisis...
4	tt1007028	Zack and Miri Make a Porno	[Comedy, Romance]	[Seth Rogen, Elizabeth Banks, Jennifer Schwalb...	[Kevin Smith]	[pornography, love of one's life, platonic lov...
5	tt0110971	Renaissance Man	[Comedy, Drama]	[Danny DeVito, Mark Wahlberg, Gregory Hines, J...	[Penny Marshall]	[vietnam veteran, commercial, advertising expe...
6	tt0118842	Chasing Amy	[Comedy, Drama, Romance]	[Ben Affleck, Joey Lauren Adams, Jason Lee, Dw...	[Kevin Smith]	[new jersey, coming out, love of one's life, b...
7	tt0113749	Mallrats	[Comedy, Romance]	[Jason Lee, Jeremy London, Shannen Doherty, Cl...	[Kevin Smith]	[sex, game show, slacker, comic, shopping]
8	tt0200027	Riding in Cars with Boys	[Biography, Comedy, Drama]	[Drew Barrymore, Steve Zahn, Adam Garcia, Brit...	[Penny Marshall]	[baby, becoming an adult, puberty, dream, drug...
9	tt0077766	Jaws 2	[Adventure, Horror, Thriller]	[Roy Scheider, Lorraine Gary, Murray Hamilton,...	[Jeannot Szwarc]	[mayor, island, police chief, sailing, boat ac...

Table 3: 10 recommended movies

From our output in Table 3, we can see that comedy and drama are two of the most prevalent genres of our 10 recommendations. We can also see that most of the movies were directed by Kevin Smith or Penny Marshall, the directors of two of the user's top rated movies. Additionally, we can see that the user rated *Clerks* highly, and the recommender recommended *Clerks 2*.

To evaluate our recommendations, we calculated the weighted f1, accuracy, precision, and recall score. From these metrics, shown in figure 5, we can see that our recommendation system works very well, with an F1 score of 0.9752, an accuracy score of 0.9752, a precision score of 0.9763, and a recall score of 0.9752.

```
F1 score: 0.9751964835896518
Accuracy: 0.9752010635291732
Precision: 0.9763319495131824
Recall: 0.9752010635291732
```

Figure 5: Evaluation metrics of the recommendations for user 225

To further test our recommender, we randomly selected 5 users, recommended each user 10 movies based on their top 5 movies, and calculated the weighted f1, accuracy, precision, and recall score. From our results, shown in Table 4, we can again see that our recommendation system is consistent across different users' test sets, with an average f1 score of 0.9742, an average accuracy of 0.9751, an average precision score of 0.9738, and an average recall score of 0.9751.

	userID	f1	accuracy	precision	recall
0	586	0.970253	0.971860	0.969159	0.971860
1	521	0.983144	0.982433	0.984415	0.982433
2	267	0.969903	0.972029	0.968182	0.972029
3	37	0.974888	0.975893	0.974436	0.975893
4	265	0.972951	0.973466	0.972665	0.973466

Table 4: Evaluation metrics of the recommendations for 5 randomly selected users

To help visualize these results, we created cosine similarity matrices. These matrices helped us visualize how similar a movie is to another movie. The darker the cell, the more similar the movies were.

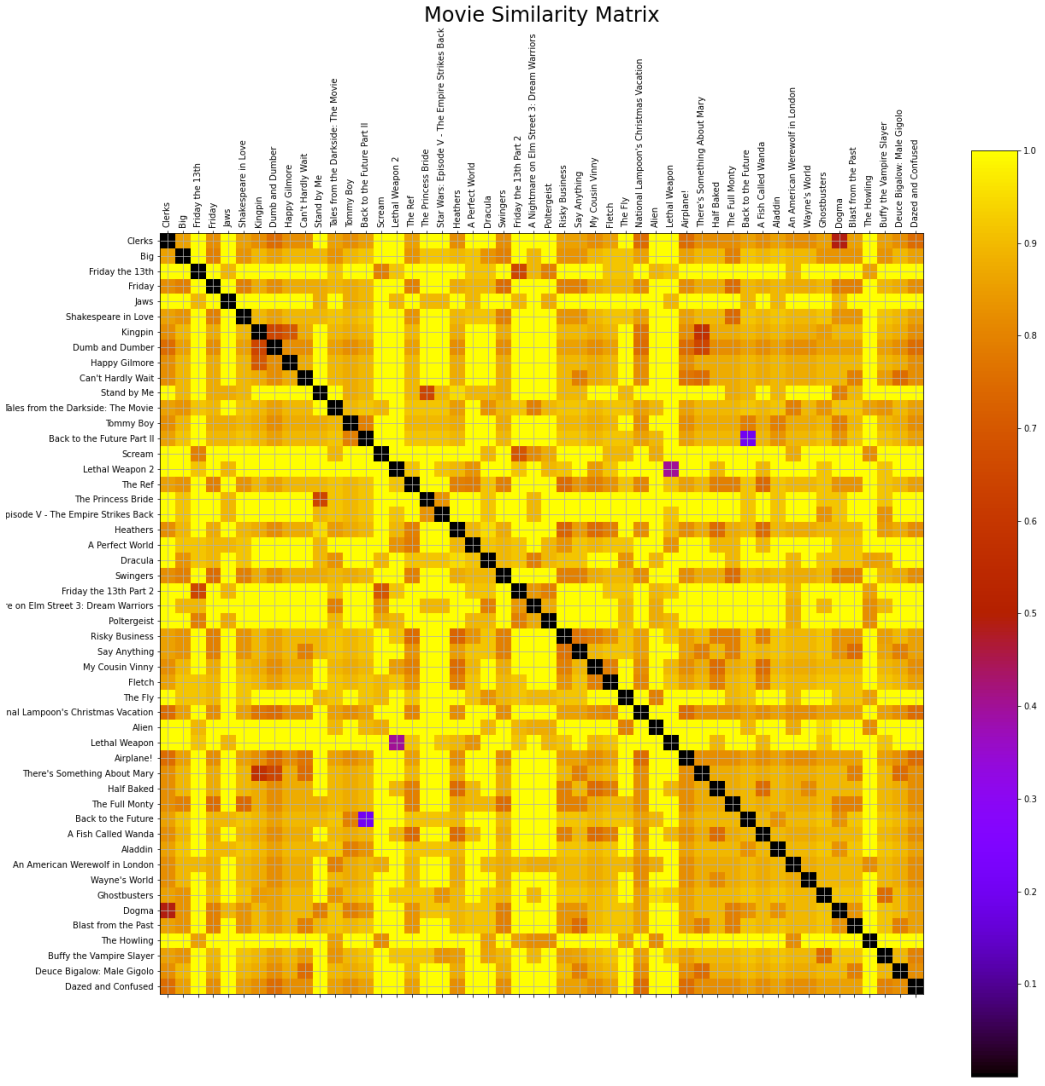


Figure 6: Cosine similarity matrix of user 225's top 50 movies

Figure 6 shows the cosine similarity matrix of a user's top 50 movies. This helps us visualize the similarities between the different movies a user has watched, specifically, the top 50 movies the user rated. We can see that the user watched relatively different movies, with a few movies that were very similar to each other.

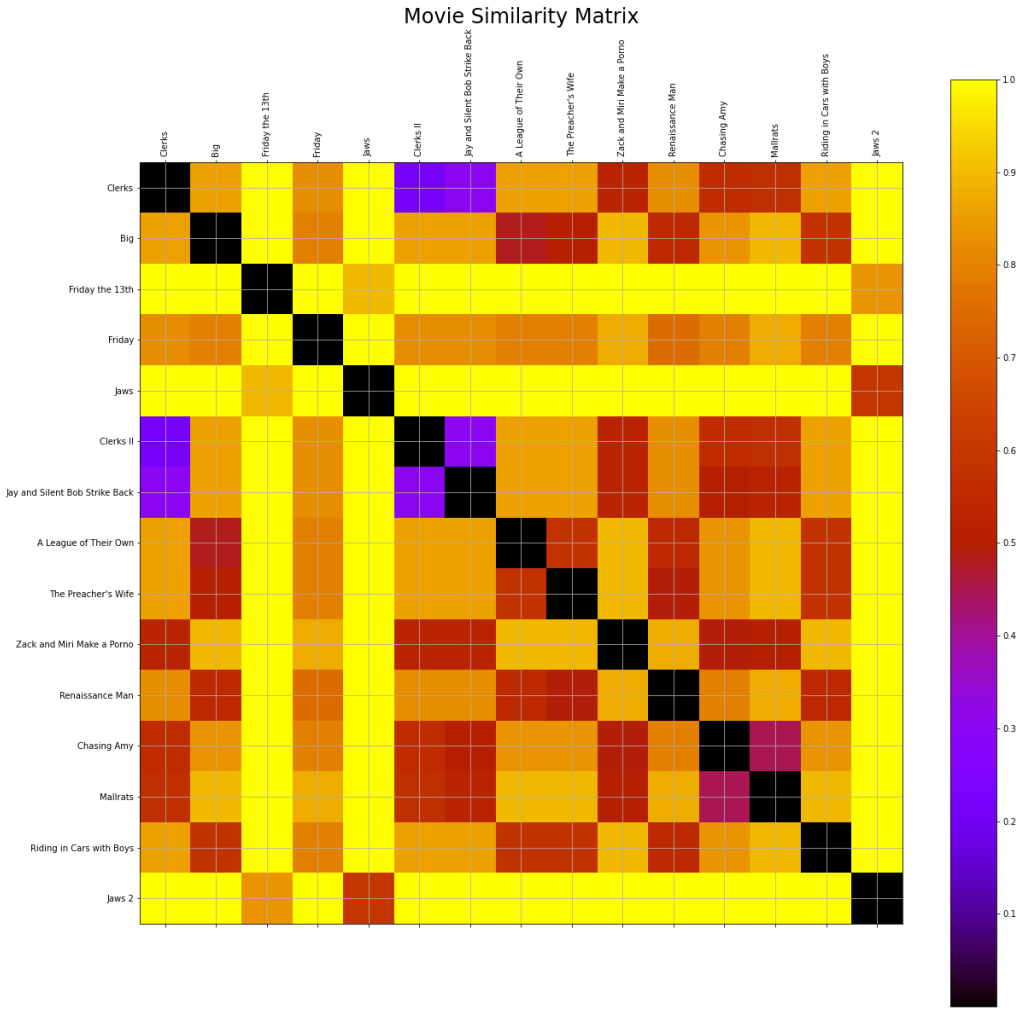


Figure 7: Cosine similarity matrix of user 225's top movies and recommended movies

Figure 7 shows the cosine similarity matrix between a user's top 5 rate movies and the 10 movies that our algorithm recommended. For this matrix, we combined the user's top 5 movies and the 10 recommended movies to create a dataframe with 15 movies. Then, we computed the similarity between each movie, so the area that the first five rows and columns cover contains the similarities between each of the user's top rated movies. From analyzing this matrix, we can see

that the user's top 5 movies weren't very similar to each other, but the algorithm recommended movies that were similar to each other and/or the user's top 5 movies. For example, *Jaws 2*, the last movie recommended, is not similar to any of the other recommended movies, but it is similar to two of the user's top movies, whereas *Clerks 2* is not only very similar to three of the user's top movies but it's also similar to eight of the other recommended movies.

From these results and visualizations, it's clear that our recommender works very well in calculating the cosine similarities between movies and recommending similar movies to what the user has watched and enjoyed.

Discussion

From analyzing the RMSE of the recommended movies in Table 1, we noticed that the RMSE of each movie is extremely low. While this may seem like a good indication of our algorithm working very well, it is also highly likely that our model is overfitted. This could be because our model is training on such a small set of data - only 5 of the user's top rated movies. To combat this issue, we think that training the model on a larger set of data would allow us to have more variations in the predictions, which could increase our cosine similarity in the vector space.

When analyzing the results of the recommender system, we had strong F1, accuracy, precision, and recall. These metrics all point towards the algorithm being able to recommend movies with accuracy, as seen in Table 4. Similarly, we can see our recommendations are very relevant to the user's top rated movies in Figure 5. Our content-based approach successfully classified our attributes and allowed us to provide solid movie recommendations. One attribute that seemed to distinctly drive recommendations was the director attribute. In the majority of our user testing, we noticed that our recommendations were heavily similar based on the director attributes.

When exploring cosine similarity between the attributes, we noticed middle of the range performance between the top rated and recommendations. Similarly in our matrix, we did not see any upper range values here. We think the driver of this could be the keywords section. Many of these tags can be extremely unique to a particular movie. With further exploration of the

keywords attribute, we could narrow down whether this would be good for using in a k-nearest neighbor classifier. Also, on a purely anecdotal inference, we did notice that genres are a very vague descriptor for movie classification. We think expansion to a data set that may have a more nuanced set of classes could have served the model better. In Figure 7, we can see the cosine similarity matrix between the top rated movies compared to the recommended ones. While certain movies stand out with high similarity, it would have given us more confidence if we saw more similarity between the choices.

Our F1 metrics were a strong signifier that the predictions between our attributes in the top 5 rated movies reflected similarly to our predicted movies attributes. This score being close to 1 signifies the strength in instances yielding strong precision and recall. Our model had very strong accuracy on our particular dataset as we either recommended the movies that had true positives and negatives when compared to the total test set. This was due to the small size of our training and test splits in each particular user test.

Conclusion

In conclusion, utilizing a content-based approach using k-nearest neighbors for recommending similar movies based on said users watched movies yielded significant results. RMSE and cosine similarity sorting allowed us to provide the most similar movies from our attribute selection. In addition, we were able to output strong accuracy and qualitatively similar attributes in our predictions. However, utilizing only a content-based approach is just scratching the surface of recommender systems. When putting these attributes in a vector space and calculating cosine similarity, it definitely caused some computation timing issues. Testing our model on 5 random users took over 20 minutes to run. In addition to the computational run-time issues, we could have focused more on attribute selection in a less intuitive approach, diving more into the details of our data's classes. Ultimately, our results were a good jumping off point in the recommendation space.

All throughout our literature review, we saw mention of collaborative filtering that focused on classifying users and recommending based on similar users that shared attributes in the movies they watched. We did not implement this approach due to the timeline and scope of the project, but a hybrid method that utilized not only the content of the movies but also

recommending on a user-user basis could have potentially yielded higher accuracy. In addition to this approach, finding data about each user could have assisted here. Attributes such as each user's age, gender, time spent watching, etc. would allow us to define profiles to classify. To further expand on this system, utilizing collaborative filtering would strengthen the accuracy and use cases of our model.

Attribute selection was an area where we took an intuitive approach. We selected based on our own notions of what were effective attributes to classify. Spending more on time on attribute selection could have potentially yielded more consistently similar movies from our feature space. In addition to performance, the attributes distribution in our data could potentially be weighted more heavily in certain categories such as the most occurring classes in genre and director. A further exploration could potentially be calculating individual metrics for each attribute, such as precision. By doing this, we could weigh which attributes to utilize in the recommendation to provide more accurate results.

Overall, this project was geared towards exploring the recommendation system space and the power that movie streaming recommender systems implement with their users. As mentioned previously, our implementation only scratched the surface to the methods that could be utilized. Our intuitive approach of attribute selection and content-based approach could potentially serve as a baseline model to highly developed ones. With the usage of an easy to understand approach this can provide a high-level understanding of the development of complex recommendation systems while also providing accurate predictions from the data.

References

Aditya, T. S., Rajaraman, K., & Monica Subashini, M. (2018). Comparative analysis of Clustering Techniques for movie recommendation. *MATEC Web of Conferences*, *225*, 02004. https://doi.org/10.1051/matecconf/201822502004

Ceci, L. (2021, July 12). YouTube - Statistics & Facts. Statista. Retrieved December 20, 2021, from https://www.statista.com/topics/2019/youtube/#dossierKeyfigures

Gomez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System. *ACM Transactions on Management Information Systems*, *6*(4), 1–19. https://doi.org/10.1145/2843948

Latif, M. H., & Afzal, H. (2016, August). Prediction of Movies Popularity Using Machine Learning Techniques. Retrieved from https://www.researchgate.net/publication/311913687_Prediction_of_Movies_popularity_Using_Machine_Learning_Techniques.

Li, B., Liao, Y., & Qin, Z. (2014). Precomputed clustering for movie recommendation system in Real time. *Journal of Applied Mathematics*, *2014*, 1–9. https://doi.org/10.1155/2014/742341

Low, E. (2021, April 20). Netflix reveals $17 billion in content spending in fiscal 2021. Variety. Retrieved December 20, 2021, from https://variety.com/2021/tv/news/netflix-2021-content-spend-17-billion-1234955953/

Phorasim, P., & Yu, L. (2017). Movies recommendation system using collaborative filtering and K-means. *International Journal of Advanced Computer Research*, *7*(29), 52–59. https://doi.org/10.19101/ijacr.2017.729004

Sadhasivam, J., Cera, J. M., Deepa, R., Satheshkumar, K., Muthukumaran, V., Satheesh Kumar, S., & Angeline Kavitha, M. (2021). Movie recommendation system using clustering mining with python. *Journal of Physics: Conference Series*, *1964*(4), 042073. https://doi.org/10.1088/1742-6596/1964/4/042073

ÇATALTEPE, Z., ULUYAĞMUR, M., & TAYFUR, E. (2016). Feature selection for movie recommendation. *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, *24*, 833–848. https://doi.org/10.3906/elk-1303-189

Presentation

Movie Recommendation System

Ian McSpiritt
Julianne Lin

Introduction

- Consumer research suggests that a typical streaming platform user will lose interest after 60 to 90 seconds of deciding before selecting a movie title or exiting the platform altogether (Gomez-Uribe & Hunt, 2016)
- Age of influence and curation
- Decision paralysis amongst users
- A great recommender retains users and differentiates from other movie streaming services

Background

Goal

Recommend Movies based on users history and movie attributes

Find Data

Find data that has user history as well as attributes on each movie

Lit Review

Research algorithms that are commonly used in recommender systems

Cleaning

Researching Methods to clean and merge data

Implement Algorithm

Explore classification algorithms and other supervised such as kNN methods

Evaluate

Figure out method to evaluate results using loss functions (Cosine Similarity, RMSE)

Literature Review

Comparative Analysis of Clustering Techniques for Movie Recommendation

They fed datasets into Euclidean distance based algorithms like k-means to cluster them based on specific feature (Aditya et al., 2018)

Feature Selection for Movie Recommendation

They used collaborative recommendation to predict the rating for instances where the user has not seen a certain movie by basing it on the user's neighbors' opinions (ÇATALTEPE et al., 2016)

Data

Movie Lens: Ratings

	userID	movieLensID	rating
0	1	1	4.0
1	1	3	4.0
2	1	6	4.0
3	1	47	5.0
4	1	50	5.0

Movie Lens: Links

	movieLensID	IMDbID	TMDbID
0	1	tt0114709	862
1	2	tt0113497	8844
2	3	tt0113228	15602
3	4	tt0114885	31357
4	5	tt0113041	11862

TMDB

TMDbID	IMDbID	popularity	budget	revenue	title	cast	director	keywords
135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	monster|dna|tyrannosaurus rex|velociraptor|island
76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	future|chase|post-apocalyptic|dystopia|australia
262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	based on novel|revolution|dystopia|sequel|dyst...
140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	android|spaceship|jedi|space opera|3d
168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	car race|speed|revenge|suspense|car

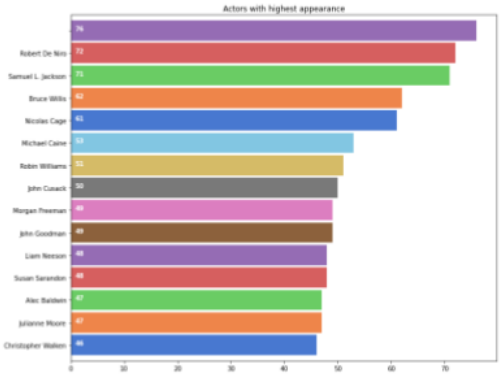
IMDB

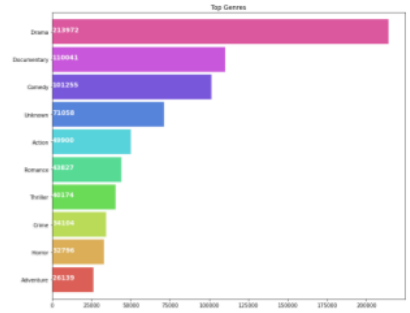
	IMDbID	type	title	originalTitle	adult	year	endYear	runtime	genres
0	tt0000001	short	Carmencita	Carmencita	0	1894	\N	1	Documentary,Short
1	tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	0	1892	\N	5	Animation,Short
2	tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892	\N	4	Animation,Comedy,Romance
3	tt0000004	short	Un bon bock	Un bon bock	0	1892	\N	12	Animation,Short
4	tt0000005	short	Blacksmith Scene	Blacksmith Scene	0	1893	\N	1	Comedy,Short

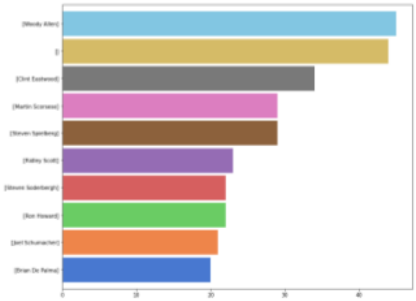
Data

movieLensID	IMDbID	title	genres	cast	director	keywords
1	tt0114709	Toy Story	[0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
2	tt0113497	Jumanji	[0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3	tt0113228	Grumpier Old Men	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
4	tt0114885	Waiting to Exhale	[0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
5	tt0113041	Father of the Bride Part II	[0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

Data Explored







Approach

Explore each data set and attributes in each

Use KNN by comparing users top rated movies recommending similar movies with strong cosine similarity

Step 1

Step 2

Clean Data to merge and encode attributes that we intuitively selected

Step 3

Step4

Evaluate Cosine similarity and RMSE of top rated movies compared to recommendations

Results

```
Hello! What is your user id? (num between 1 and 610): 246
You've rated 161 movies. How many of your top rated movies would you like us to compare?: 5
How many movies would you like us to recommend?: 10
```

	IMDbID	similarity	RMSE
22432	tt0167261	0.050000	0.003983
22113	tt0120737	0.100000	0.005633
25389	tt1170358	0.447938	0.091168
24884	tt0903624	0.447938	0.091168
27412	tt2310332	0.447938	0.091168
46192	tt2194499	0.468784	0.077678
4751	tt0388795	0.500000	0.014638
23424	tt0360717	0.500000	0.014638
3022	tt0119349	0.523223	0.060121
34514	tt1014759	0.533333	0.079686

	IMDbID	title	genres	cast	director	keywords
3023	tt0119349	The Ice Storm	[Drama]	[Kevin Kline, Joan Allen, Sigourney Weaver, He...	[Ang Lee]	[based on novel, 1970s, thanksgiving, dysfunct...
3183	tt0120737	The Lord of the Rings: The Fellowship of the Ring	[Action, Adventure, Drama]	[Elijah Wood, Ian McKellen, Viggo Mortensen, L...	[Peter Jackson]	[elves, dwarves, orcs, middle-earth (tolkien),...
3503	tt0167261	The Lord of the Rings: The Two Towers	[Action, Adventure, Drama]	[Elijah Wood, Ian McKellen, Viggo Mortensen, L...	[Peter Jackson]	[elves, orcs, middle-earth (tolkien), hobbits,...
4495	tt0360717	King Kong	[Action, Adventure, Drama]	[Naomi Watts, Jack Black, Adrien Brody, Thomas...	[Peter Jackson]	[film business, screenplay, show business, fil...
4752	tt0388795	Brokeback Mountain	[Drama, Romance]	[Heath Ledger, Jake Gyllenhaal, Randy Quaid, M...	[Ang Lee]	[gay, countryside, homophobia, loss of lover, ...
5955	tt0903624	The Hobbit: An Unexpected Journey	[Adventure, Fantasy]	[Ian McKellen, Martin Freeman, Richard Armitag...	[Peter Jackson]	[riddle, elves, dwarves, orcs, middle-earth (t...
6120	tt1014759	Alice in Wonderland	[Adventure, Family, Fantasy]	[Mia Wasikowska, Johnny Depp, Anne Hathaway, H...	[Tim Burton]	[based on novel, fictional place, queen, alice...
6460	tt1170358	The Hobbit: The Desolation of Smaug	[Adventure, Fantasy]	[Martin Freeman, Ian McKellen, Richard Armitag...	[Peter Jackson]	[elves, dwarves, orcs, middle-earth (tolkien),...
8333	tt2194499	About Time	[Comedy, Drama, Fantasy]	[Rachel McAdams, Bill Nighy, Domhnall Gleeson,...	[Richard Curtis]	[london, father-son relationship, time travel]
8483	tt2310332	The Hobbit: The Battle of the Five Armies	[Adventure, Fantasy]	[Martin Freeman, Ian McKellen, Richard Armitag...	[Peter Jackson]	[corruption, elves, dwarves, orcs, middle-eart...

Conclusion

- Our predictions had very 'good' cosine similarities and RMSE and we believe it could potentially be overfitted to the top rated movies
- Potential to cluster user's and recommend based on similar users opposed to focusing more on content of instances
- Expansion of our training set to include not just highly rated movies by the particular users
- Recommendations were strong

--

References

Aditya, T. S., Rajaraman, K., & Monica Subashini, M. (2018). Comparative analysis of Clustering Techniques for movie recommendation. MATEC Web of Conferences, 225, 02004. https://doi.org/10.1051/matecconf/201822502004

Gomez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System. ACM Transactions on Management Information Systems, 6(4), 1–19. https://doi.org/10.1145/2843948

ÇATALTEPE, Z., ULUYAĞMUR, M., & TAYFUR, E. (2016). Feature selection for movie recommendation. TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES, 24, 833–848. https://doi.org/10.3906/elk-1303-189

Thanks!

Do you have any questions?

ian.mcspiritt@rutgers.edu
jjl302@scarletmail.rutgers.edu

Data Sources

The following are the first five instances and the features of each of the cleaned datasets that we used.

MovieLens Ratings

	userID	movieLensID	rating
0	1	1	4.0
1	1	3	4.0
2	1	6	4.0
3	1	47	5.0
4	1	50	5.0

MovieLens Links

	movieLensID	IMDbID	TMDbID
0	1	tt0114709	862
1	2	tt0113497	8844
2	3	tt0113228	15602
3	4	tt0114885	31357
4	5	tt0113041	11862

MovieLens Ratings & Links

	userID	movieLensID	rating	IMDbID	TMDbID
0	1	1	4.0	tt0114709	862
1	5	1	4.0	tt0114709	862
2	7	1	4.5	tt0114709	862
3	15	1	2.5	tt0114709	862
4	17	1	4.5	tt0114709	862

IMDb

	IMDbID	title	genres	genres_binary
498	tt0000502	Bohemios	[\N]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
570	tt0000574	The Story of the Kelly Gang	[Action, Adventure, Biography]	[1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
587	tt0000591	The Prodigal Son	[Drama]	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...
610	tt0000615	Robbery Under Arms	[Drama]	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...
625	tt0000630	Hamlet	[Drama]	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...

IMDb with ratings

	IMDbID	title	genres	genres_binary	IMDb_rating	IMDb_votes
0	tt0000502	Bohemios	[\N]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	4.5	14
1	tt0000574	The Story of the Kelly Gang	[Action, Adventure, Biography]	[1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	6.1	739
2	tt0000591	The Prodigal Son	[Drama]	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...	5.2	16
3	tt0000615	Robbery Under Arms	[Drama]	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...	4.5	23
4	tt0000630	Hamlet	[Drama]	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...	3.8	23

TMDb

	TMDbID	IMDbID	title	cast	director	keywords	TMDb_votes	TMDb_rating	cast_binary	keywords_binary	director_binary
0	135397	tt0369610	Jurassic World	[Chris Pratt, Bryce Dallas Howard, Irrfan Khan...	[Colin Trevorrow]	[monster, dna, tyrannosaurus rex, velociraptor...	5562	6.5	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1	76341	tt1392190	Mad Max: Fury Road	[Tom Hardy, Charlize Theron, Hugh Keays-Byrne,...	[George Miller]	[future, chase, post-apocalyptic, dystopia, au...	6185	7.1	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
2	262500	tt2908446	Insurgent	[Shailene Woodley, Theo James, Kate Winslet, A...	[Robert Schwentke]	[based on novel, revolution, dystopia, sequel,...	2480	6.3	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3	140607	tt2488496	Star Wars: The Force Awakens	[Harrison Ford, Mark Hamill, Carrie Fisher, Ad...	[J.J. Abrams]	[android, spaceship, jedi, space opera, 3d]	5292	7.5	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
4	168259	tt2820852	Furious 7	[Vin Diesel, Paul Walker, Jason Statham, Miche...	[James Wan]	[car race, speed, revenge, suspense, car]	2947	7.3	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

IMDb and TMDb

	IMDbID	title	genres	genres_binary	IMDb_rating	IMDb_votes	TMDbID	cast	director	keywords	TMDb_votes	TMDb_rating	cast_binary	keywords_binary	director_binary
0	tt0035423	Kate & Leopold	[Comedy, Fantasy, Romance]	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...	6.4	82109	11232	[Meg Ryan, Hugh Jackman, Liev Schreiber, Breck...	[James Mangold]	[lover [female], love of one's life, time trav...	248	5.8	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1	tt0052646	The Brain That Wouldn't Die	[Horror, Sci-Fi]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ...	4.5	6801	33468	[Jason Evers, Virginia Leith, Doris Brent, Aud...	[Joseph Green]	[transplantation, experiment, mutant, brain, f...	12	4.9	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
2	tt0053559	13 Ghosts	[Horror, Mystery]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ...	6.1	6059	29756	[Charles Herbert, Jo Morrow, Martin Milner, Ro...	[William Castle]	[haunted house]	12	5.5	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3	tt0053580	The Alamo	[Adventure, Drama, History]	[0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, ...	6.8	15401	11209	[John Wayne, Richard Widmark, Laurence Harvey,...	[John Wayne]	[texas, assault, alamo, mexican, mexican army]	27	6.2	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
4	tt0053604	The Apartment	[Comedy, Drama, Romance]	[0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...	8.3	173892	284	[Jack Lemmon, Shirley MacLaine, Fred MacMurray,...	[Billy Wilder]	[new york, new year's eve, lovesickness, age d...	235	7.9	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

IMDb and TMDb and MovieLens

	movieLensID	IMDbID	TMDbID	title	genres	genres_binary	IMDb_rating	IMDb_votes	cast	director	keywords	TMDb_votes	TMDb_rating	cast_binary	keywords_binary	director_binary
0	1	tt0114709	862	Toy Story	[Adventure, Animation, Comedy]	[0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	8.3	936979	[Tom Hanks, Tim Allen, Don Rickles, Jim Varney,...	[John Lasseter]	[jealousy, toy, boy, friendship, friends]	3141	7.5	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1	2	tt0113497	8844	Jumanji	[Adventure, Comedy, Family]	[0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...	7.0	327696	[Robin Williams, Jonathan Hyde, Kirsten Dunst,...	[Joe Johnston]	[board game, disappearance, based on a childre...	1105	6.6	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
2	3	tt0113228	15602	Grumpier Old Men	[Comedy, Romance]	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	6.7	26208	[Walter Matthau, Jack Lemmon, Ann-Margret, Sop...	[Howard Deutch]	[fishing, best friend, duringcreditsstinger, o...	45	6.7	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
3	4	tt0114885	31357	Waiting to Exhale	[Comedy, Drama, Romance]	[0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...	6.0	10547	[Whitney Houston, Angela Bassett, Loretta Devi...	[Forest Whitaker]	[based on novel, interracial relationship, sin...	18	6.1	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
4	5	tt0113041	11862	Father of the Bride Part II	[Comedy, Family, Romance]	[0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...	6.1	36802	[Steve Martin, Diane Keaton, Martin Short, Kim...	[Charles Shyer]	[baby, midlife crisis, confidence, aging, daug...	82	5.7	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

Time Logs

Julianne Lin

date	time	hours	task
10/13	10-11	1	Data Search
10/16	9-10	1	Literature Search
10/17	5-7	2	Literature Review
10/20	6-7	1	Meet with teaching team
10/26	5-6	1	Project Proposal Planning
10/27	12-2	2	Literature Search
10/29	6-8	2	Literature Review
10/30	1-3	2	Literature Review
10/31	4-6	2	Literature Review
10/31	5-6	1	Project Proposal Writing
10/31	9-12	3	Project Proposal & Presentation Prep
11/2	5-6	1	Project Proposal Final Details & Presentation Prep
11/9	5-6	1	Project Timeline Planning
11/14	12-2	2	Data Accessing
11/16	5-6	1	Data Accessing & Meeting
11/30	7-9	2	NLP Research & Meeting
12/7	7-10	3	Data Accessing & Cleaning & Meeting
12/8	5-7	2	Data Cleaning
12/9	7-10	3	Data Cleaning
12/10	1-4	3	Data Cleaning & Meeting
12/11	10-12	2	Data Cleaning & Algorithm Implementation
12/12	2-4	2	Data Cleaning & Algorithm Implementation
12/12	6-7	1	Data Cleaning & Algorithm Implementation
12/12	10-11	1	Algorithm Implementation & Meeting
12/13	12-2	2	Data Cleaning & Transformation
12/13	4-6	2	Data Cleaning & Algorithm Implementation
12/14	2-4	2	Algorithm Implementation
12/14	10-12	2	Algorithm Implementation & Meeting
12/15	2-3	1	Data Visualization
computer decided to break :')			

12/18	5-7	2	Data Visualization
12/18	9-12	3	Data Evaluation
12/19	12-2	2	Data Evaluation
12/19	1-7	6	Data and Output Visualization and Final Report Write Up
12/19	8-12	4	Data and Output Visualization and Final Report Write Up

Ian McSpiritt

date	**time**	**hours**	**task**
10/13	10-11	1	Data Search/Meeting
10/17	8-10	2	Api Research on TMDB
10/20	4:30-6:30	1	Meeting and Data exploration
10/29	12-2	2	Working on Proposal
10/30	12-4	4	Working on Proposal
10/31	12-5	5	Working on Proposal
10/31	9-11	2	Working on Proposal
11/22	5-9	4	Data Cleaning and Meeting
11/29	5-9	4	Data Cleaning and Meeting
12/5	11-5	6	Data Cleaning
12/7	7-10	3	Data Accessing & Cleaning & Meeting
12/10	1-4	3	Data Cleaning & Meeting
12/11	12-5	5	Data Cleaning & Algorithm Implementation
12/12	6-11	5	Data Cleaning & Algorithm Implementation & Meeting
12/13	10-1	3	Algorithm Implementation/ Feature testing
12/14	8-12	4	Evaluation and presentation
12/15	3:30-6:30	3	Presentation and Prep
12/18	5-12	7	Data Visualization & Evaluation
12/19	12-2	2	Data Visualization & Evaluation
12/19	1-12	11	Final Report Write-Up