MIS 546 : Disney Plus Popularity & Recommendation Project

Erwin Chege, Chenrui Niu, Rachel Carlson

Introduction

The Disney plus streaming service has over 1000 titles with over 30k actors and directors. The front page of Disney Plus is a competitive marketplace, with each show trying to grab the audience's attention. Disney's audience has an abundance of films to choose from, and Disney's producers have a wealth of directors and actors willing to work for them. Audience attention is limited however, and streaming services such as Paramount Plus and Netflix are also competing for the same audience. It is in Disney's best interest to find which directors, genres of movie, and production locations are the most popular among its audience to continue providing them with the most appealing content. Our goal is to search through the dataset of movies and shows provided by Kaggle to find which will be the most popular within the next 5 years.

Dataset Details

Our team chose to use the dataset "Disney+ Movies and TV Shows" from Kaggle.com, published by Diego Enrique. The Data contained 2 files, one is called Titles.csv which contains 1000 movies and 15 columns of data, another is called Credits.csv which contains 30,000 credits for actors and directors.

The information and dataset are from the website: www.justwatch.com/us/provider/disney-plus.

Variables

Raw Variables

The variables included in the raw data were TMDB score (1-10), TMDB popularity (based on user clicks, saves, watchlist adds, etc.), title, type (movie or show), description, release year, age certification, genre, production country, seasons, IMDB ID, IMDB score, IMDB popularity, runtime, name, character name, and director.

Custom Variables

A variable named "decade" was created based on release year. Dummy variables were then created from the new "decade" variable and allowed us to more easily incorporate release year into our various models.

For the text analytics two custom variables were created. The variable, "actors" was created from the credit.csv file. The actors in the file were put into a list and merged into the dataset based on movie id. The second variable created was titled "combined". It combined all of the text data used in the text analytics into one column – this included the description, genre, directors and actors.

Dummy variables were also created for various models for fields such as type, age certification, decades, country and genre.

Data Cleaning

The kaggle dataset had two files, credit.csv and title.csv. The title file contained the majority of the information about each movie (title, description, production country, etc.). The credit data file contained all of the actors and directors in the movies in the title dataset. For our base file we just wanted to keep the director data from the

credit file. To do this we split the credit data into two dataset – actors and directors – and then merged the director file with the movie file based on movie id. We used a left join on the title data to make sure all movie ids were kept even if the director data was blank.

Once the merged dataset was created it then needed to be cleaned. We started by identifying all fields with null or missing values:

```
merged.isna().sum()
```

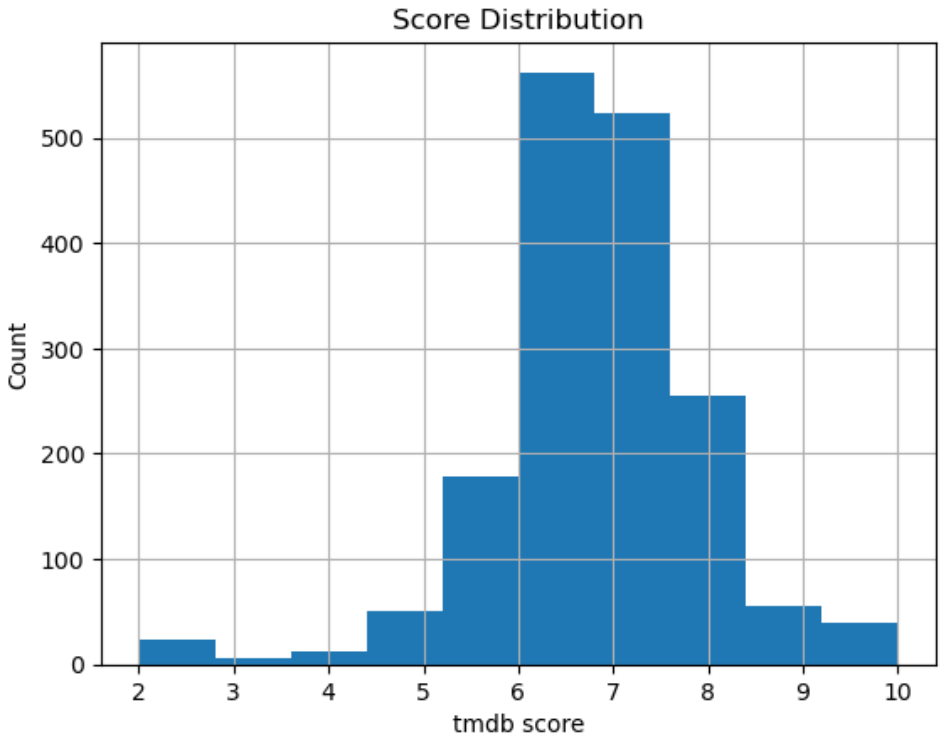
```
id                        0
title                     0
type                      0
description               9
release_year              0
age_certification       451
runtime                   0
genres                    0
production_countries      0
seasons                1314
imdb_id                 478
imdb_score              515
imdb_votes              526
tmdb_popularity          15
tmdb_score              146
director                  0
dtype: int64
```

All of the rows that were missing a value in the variable “season” were movies, so we replaced NA with zero in these rows. We replaced the missing age certification fields with “unrated” because removing these rows would eliminate almost 25% of our dataset. Based on the large amount of IMDB data that was missing we decided to use TMDB popularity and score in our models, and we removed the IMDB id, IMDB score and IMDB votes from our dataset. The rows with missing values for TMDB popularity and TMDB score were removed from the dataset and the missing descriptions were left blank for our analysis.

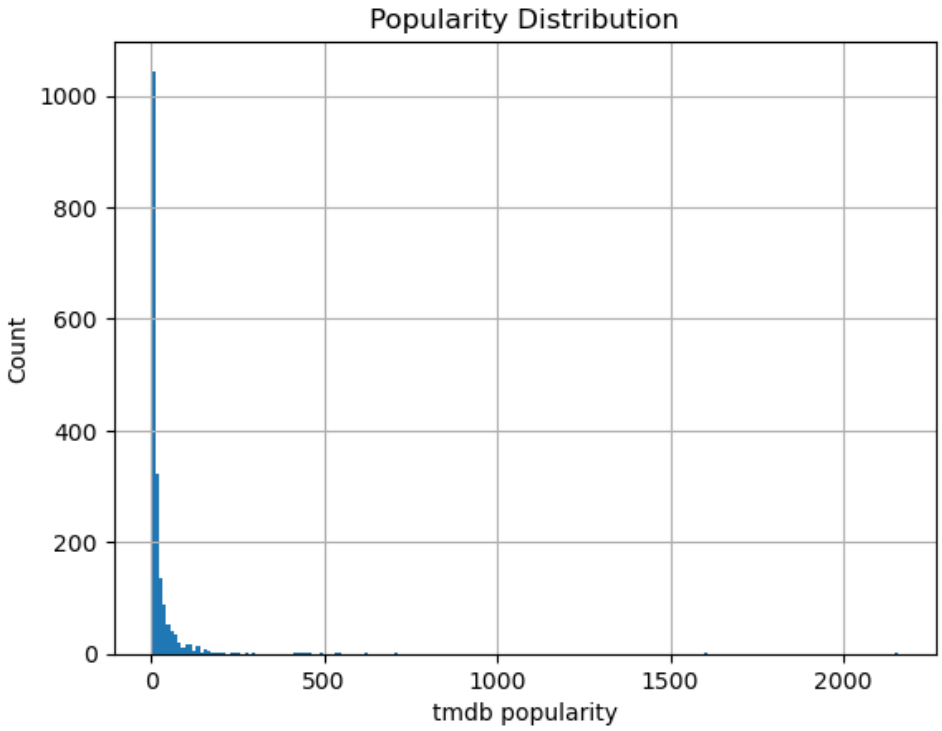
Additional data cleaning was done based on the needs of the various models built. Some examples of these are removing special characters, adding actors in a list format as a new variable, and the combining of text variables for text analytics.

Data Distribution

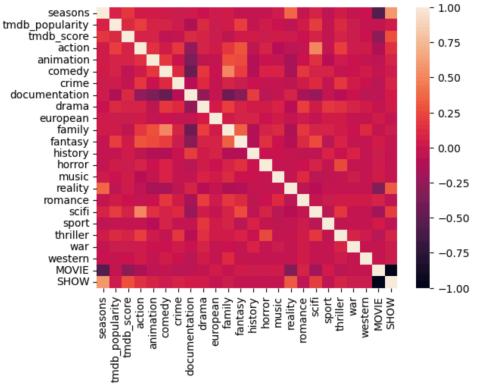
To get to know the data better before we started building our models, we looked at the distribution of some variables we were initially interested in. First, we wanted to look at both TMDB score and TMDB popularity, since these would be the primary variables were going to be trying to predict in our models. An initial look at the score distribution showed a distribution that was slightly skewed to the right. It showed that the majority of scores were between 5.5 and 8.5, but there were scores as low as two and as high as ten.



Next we looked at TMDB popularity distribution. TMDB popularity is based on number of votes, number of views, number of times marked “favorite”, number of times saved to a users’ watchlist, total votes, etc. The range of scores for TMDB popularity ranged from .6 to 2,159, with the majority of values falling between 0 to 250.

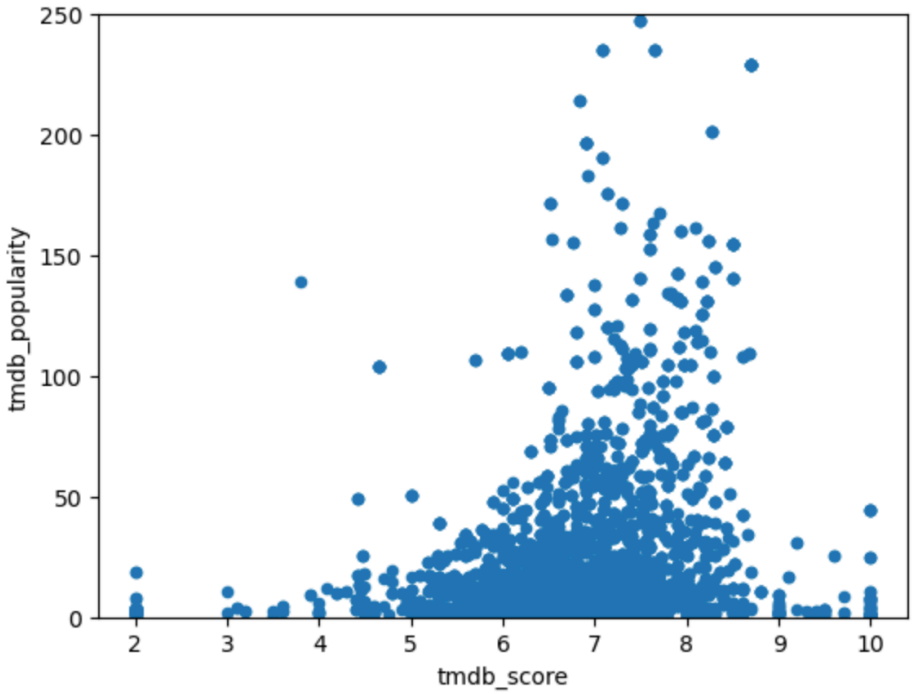


Next, we did a high-level correlation matrix that included the genres, type (movie or show), TMDB score TMDB popularity to see if anything stuck out as being a good variable to focus on in our analysis. Based on this correlation matrix we could see that there were no variables with a very high correlation to TMDB score or TMDB popularity. We also could see that TMDB score and TMDB popularity were not as correlated as we thought they would be, only have a correlation value of .10.

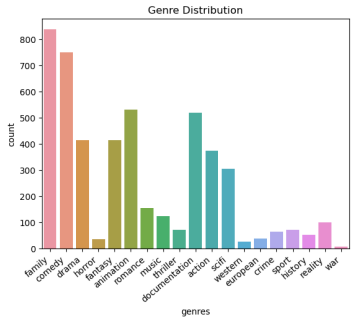


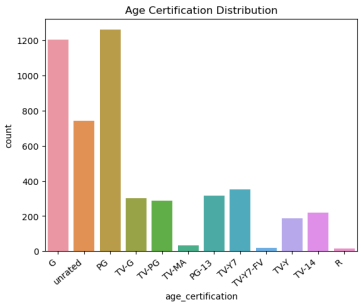
	seasons	tmdb_popularity	tmdb_score	action
seasons	1.000000	0.090294	0.169648	0.046381
tmdb_popularity	0.090294	1.000000	0.100453	0.205600
tmdb_score	0.169648	0.100453	1.000000	0.066006
action	0.046381	0.205600	0.066006	1.000000
animation	0.035548	0.064434	0.072353	0.136534
comedy	0.026527	0.055587	-0.049898	0.030986
crime	0.011999	0.021336	-0.014351	0.159279
documentation	0.017008	-0.135790	0.094266	-0.244631
drama	0.005130	0.099639	0.071093	0.077131
european	0.017774	0.012539	0.012232	0.042273
family	0.024991	0.024302	-0.039428	0.159323

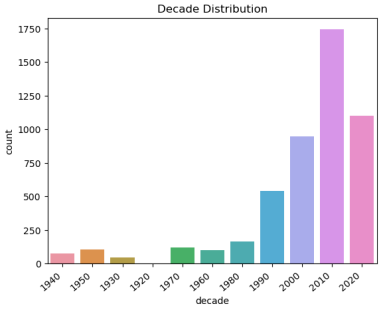
When comparing TMBD score just to TMDB popularity (graph focused on 0-250 where majority of popularity values are located) we could see that there were data points where movies had low scores and low popularity values, but also data points where movies had high scores but low popularity values. When looking further into these we discover that this was often related to more seasonal movies, or movies not always available on the platform, among other reasons. We also noticed that where the majority of the score data was (5-8) most popularity scores were below 100.



The final item we looked at was the distribution of some of the key variables we initially thought would be helpful in predicting the score or popularity of a movie. We looked at the distributions for each variable, and as expected, the majority of data was in family friendly genres and age certification categories, and movies released after 1990.







Feature Selection

Although we had some initial ideas of variables we wanted to focus on for our project to try to make models from we wanted to first do a feature selection model to see if any other variables stood out, and if we could make a good linear regression model based on these features. We wanted to do feature selection and a linear regression model for both TMDB popularity and TMDB score.

Data Preprocessing:

For our feature selection we wanted to use as many variables as possible. In order to do this ,we needed to create dummy variables for type (movie or show), genre and age certification. We also created a new variable called “decades” that was based on the release year. We did not use production country for this analysis because of the number of countries in the dataset, and many of the movies were produced in the United States. We also normalized the runtime and TMDB popularity values because of their wide ranges and higher values when compared to the rest of the dataset.

Feature Selection:

TMDB Popularity

We used both k-best using f-regression and p-values for our features selection. The k-best resulted in “seasons” being the top feature, followed by “R”, “PG-13” and several genres as seen below:

	Features	Popularity_Score
1	seasons	2846.548043
25	R	606.168085
24	PG-13	438.995592
7	documentation	374.569693
3	action	366.490438
11	fantasy	362.947374
17	scifi	342.962804
31	unrated	327.329318
19	thriller	313.057327
13	horror	306.297720
26	TV-14	267.438077
5	comedy	263.053322
16	romance	240.217231
20	war	230.850104
10	family	227.279954
4	animation	225.421014
8	drama	222.646615
6	crime	221.021012
42	2000	201.707265
30	TV-Y7	195.328903
44	2020	187.443623
9	european	185.307003
41	1990	173.803851
23	PG	171.808415

We then compared this to the results obtained from doing feature selection using p-values. When doing the feature selection using p-values we removed each variable one at a time until all variables had a p-value of less than .05. This resulted in the variables runtime, TMDB score, animation, documentation, family, sci-fi, thriller, PG-13 and 2020 being used in our model.

	feature	pvalue
0	runtime	0.0000
1	tmdb_score	0.0000
2	documentation	0.0000
3	fantasy	0.0000
4	2020	0.0000
5	const	0.0001
6	scifi	0.0001
7	PG-13	0.0015
8	thriller	0.0072
9	family	0.0125
10	animation	0.0474

TMDB Score

We used both k-best using f-regression and p-values for our features selection. The k-best resulted in “seasons” being the top feature, followed by “show”, “2020”, documentation and several age certifications.

	Features	Score
1	seasons	550.373425
33	SHOW	201.144904
44	2020	110.774219
7	documentation	77.996480
32	MOVIE	74.581818
27	TV-G	60.011758
26	TV-14	59.104083
28	TV-PG	58.169035
15	reality	57.344107
42	2000	57.175452
22	G	51.313170
5	comedy	35.789309
43	2010	35.227955
4	animation	33.345755
36	1940	32.646956
10	family	31.279642
30	TV-Y7	30.209197
29	TV-Y	30.208224
23	PG	30.133198
41	1990	29.271422
11	fantasy	25.522492
8	drama	24.662462
17	scifi	23.928667

We then compared this to the results obtained from doing feature selection using p-values. When doing the feature selection using p-values we removed each variable one at a time until all variables had a p-value of less than .05. This resulted in the variables of animation, documentation, show, drama, 2010, tmdb popularity, 1950, family, music and 1960 being used in our model.

	feature	pvalue
0	const	0.0000
1	animation	0.0000
2	documentation	0.0000
3	SHOW	0.0000
4	drama	0.0001
5	2010	0.0001
6	tmdb_popularity	0.0003
7	1950	0.0003
8	family	0.0207
9	music	0.0279
10	1960	0.0457

Model Selection:

For this section we just wanted to run an initial linear regression model to see the results before we focused more on our areas of interest.

Model Results and Evaluation:

For all models we used a 70/30 training testing dataset split.

TMDB Popularity

The linear regression model using the variables selected from f-regression (seasons, show, 2020, TV-PG, and documentation) produced and R2 value of .03 in the training model and .07 in the testing model with RMSE values of .04 in the training model and .03 in the testing model. Both the training and testing models produced very poor results.

The linear regression model ran with the variables selected using p-values resulted in an R2 of around .10 for both the training and testing dataset and RMSE values of 65 for the training dataset and 101 for the testing dataset. The linear regression model ran with all variables produced and R2 of .147. Although the model with the reduced variables had a comparable R2 score with the full variable model – neither model produced good results and we would not recommend using a linear regression model with this dataset.

TMDB Score

The linear regression model ran with the variables selected using p-values resulted in an R2 of .15 in the training dataset and .02 in the testing dataset and RMSE values of 71 for the training dataset and 90 for the testing dataset. The linear regression model ran with all variables produced and R2 of .157, similar to the training dataset. Although the model with the reduced variables had a comparable R2 score with the full variable model – neither model produced good results and we would not recommend using a linear regression model with this dataset.

Discussion:

The linear regression models using the top features as a result of our analysis did not produce good results. If this had been the focus of our research we could have continued to refine, use more model types, and possibly include some text analysis to produce a better model. A larger data dataset would also benefit this analysis.

Regression Models – Genre

We would like to know if we can build a predictive model to predict the TMDB score and TMDB popularity of movies or TV shows by their genre, or if additional variables would be needed to predict whether a movie or show would have a high score or be popular.

Feature Selection:

Two models were created, one using genre and TMDB popularity and one using genre and TMDB score.

Data Preprocessing:

The genre variable in the initial dataset was categorical and depending on the movie it often contained more than one genre type. The first step was to regular expression to remove all brackets. This resulted in the following nineteen genres: action, animation, comedy, crime, documentation, drama, European, family, fantasy, history, horror, music, reality, romance, sci-fi, sport, thriller, war, and western. We also normalized the TMDB popularity variable for this model.

Model Selection:

For both the TMDB popularity and the TMDB score predictor models we decided to create both a linear regression model and a random forest model. We knew that the results from the linear regression model probably would not be good based on the feature selection analysis that we had done but were hoping that random forest would produce better results based on our dataset size.

Model Results and Evaluation:

For both the popularity and score prediction models a 70/30 split was used to create a training and testing dataset. RMSE and R2 were used to evaluate the model results and cross validation was performed on both models.

IMDB Popularity

In the training dataset the random forest model had significantly better R2 results when compared to the linear regression model. In the random forest model almost 61% of all variance was explained by the model, compared

to only 10% in the linear regression model. Both models had good RMSE scores as well in the testing datasets, with the random forest model producing an RMSE of .03 and the linear regression model producing and RMSE of .04.

While the testing dataset for the linear regression model produced similar results as the training dataset, the random forest model produced significantly different results indicating there may be some overfitting. The linear regression testing model produced an R2 of .02, similar to the testing dataset, but the random forest model produced an R2 of –2.8, significantly worse than the training dataset. Both the linear regression and random forest models RMSE were similoar to their training datasets. These results were then cross validated, and the results are shown below:

	Linear Regression	Random Forest
Training RMSE	0.039352	0.026064
Testing RMSE	0.028014	0.055190
Cross Validation - RMSE	0.033633	0.042131
Training R^2	0.100055	0.605206
Testing R^2	0.022018	-2.795803
Cross Validation - R2	0.015293	-1.092631

IMDB Score

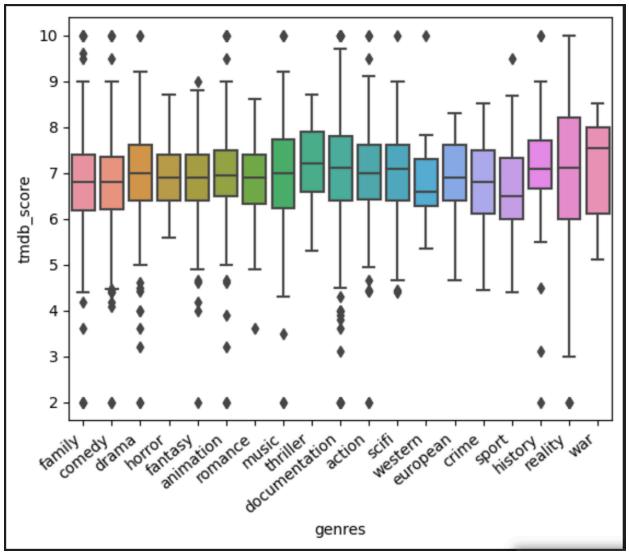
In the training dataset the random forest model produced a better R2 of .26 compared to an R2 of .07 in the linear regression model. Both models had fairly similar RMSE values, with the random forest model having andRMSE of 1.0 and the linear regression model having and RMSE of 1.12. Neither of these values indicate that either model will have good results.

In the testing dataset the random forest model and the linear regression model had negative R2 values, with the random forest model have a signficant change in its R2 value, indicating overfitting. The RMSE values were similar the the testing model for both the linear regression model and the random forest model. All results were then cross validated, and the results are shown below:

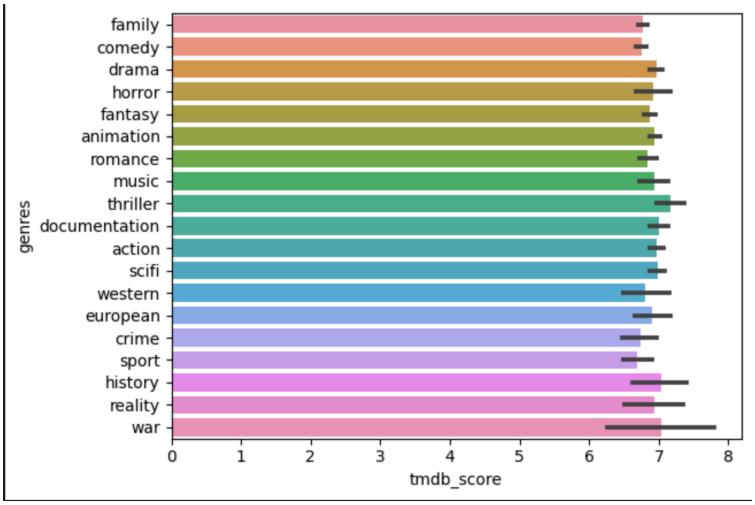
	Linear Regression	Random Forest
Training RMSE	1.126565	1.004356
Testing RMSE	1.171568	1.220018
Cross Validation - RMSE	1.138586	1.191989
Training R^2	0.066305	0.257891
Testing R^2	-0.011535	-0.096927
Cross Validation - R2	0.037819	-0.056558

Discussion:

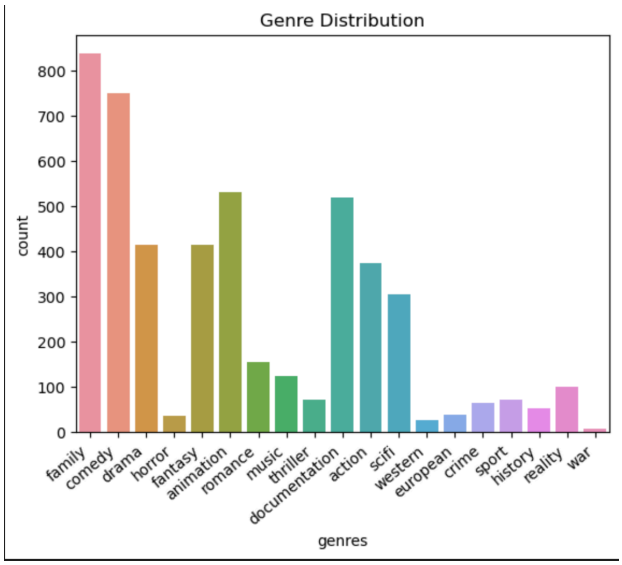
All models performed very poorly, and based on our initial review of the data we did not expect this to improve much with any other type of model created. When looking further into the distribution and spread of both popularity it was evident that it would be incredibly difficult to predict a movie or show's popularity, or score based on genre alone. In the graph below showing a zoomed in view of popularity by genre you can see that there is a wide range, with many outliers for each genre, but in general a lot of the median values were fairly close across genres.



Similarly for TMDB score, there was not much variance across genres in the average score as shown below:



In addition to this, the distribution of data across genres is not equal, create some areas of thin data that are much harder to predict:



Even with a larger dataset it would probably be difficult to predict a movie or show's popularity or score with much accuracy based on genre alone. You would probably need to include multiple features, try multiple more types of models, and include more data in the dataset to get a model that had higher accuracy and could be used.

Recommendation Model

Being able to predict to users new shows and movies to watch on a streaming platform is important. It can help drive use of the platform, keep users engaged and prevent users from using other platforms.

Feature Selection:

For this model we wanted to use text analytics on the description, directors, actors and genre for each movie or show. In addition to this we wanted to be able to display each movies' respective TMDB score and TMDB popularity value.

Data Preprocessing:

The first step in getting our dataset ready was to include the actors from the original credit file. The actors were brought in as a list and merged with the move dataset based on movie id using a left join. We then removed all special characters from the actors field. In order to make the model easier to run we combined all text into one new variable called "combined". This contained the description, genre, actors and directors associated with the movie. We then removed unnecessary columns from the dataset.

Model Selection:

We used cosine similarity and count vectorizer to help us build a user recommendation model that could be used on the Disney plus platform as well as a model that could be used by Disney executives to help determine what new movies could be added to their platform. The user recommendation model produced ten recommendations for the user based on a move the user liked and the executive recommendation model produced ten similar movies or shows currently on the platform to the new movie being evaluated.

Model Results and Evaluation:

We testing this model using “Toy Story” as a move that the user liked and received the following recommendations:

Ten recommended movies/shows based on what user likes with popularity and score:

	id	movie	tmdb_popularity	tmdb_score
0	tm14765	Toy Story	118.084	7.966
1	tm40070	Toy Story 2	95.388	7.583
2	tm101226	Toy Story 3	75.159	7.800
3	tm84279	The Pixar Story	8.330	7.600
4	tm62982	A Bug's Life	69.991	6.964
5	tm11948	Toy Story 4	72.285	7.500
6	tm1216953	Beyond Infinity: Buzz and the Journey to Light...	29.080	7.100
7	tm171776	Cars 2	109.501	6.047
8	tm37560	Monsters, Inc.	133.758	7.827
9	tm31341	Cars	70.655	6.907

The first row is the movie the user liked (Toy Story) followed by ten recommended movies for that user. Based on our model this user would also like the remaining Toy Story movies in the franchise as well as other Pixar movies.

Next, we wanted to take the recommender a step further and make it useful for Disney executives to help them determine if they should add movies to their platform by seeing the popularity, score and how many similar movies they already have on their platform. To do this we added a new movie (The Land Before Time) to the dataset, using data obtained from tmdb.com. We input the movie title, description and genre to create our model prediction.

	id	movie	tmdb_popularity	tmdb_score
0	new	The Land Before Time		
1	tm164519	La luna	5.772	7.866
2	tm91170	Four Days in October	3.182	6.5
3	ts34060	Secret Life of Predators	0.874	7.5
4	ts35950	Wicked Tuna	13.942	8.2
5	tm449027	42 to 1	1.704	6.7
6	tm29336	Liz & Dick	4.678	3.6
7	ts54096	Ancient X-Files	3.872	6.9
8	ts271431	Into the Unknown: Making Frozen II	11.692	6.895
9	tm97275	Pony Excess	2.543	7.3

Based on our inputs the model provided ten similar movies on the Disney Plus platform already (that are in our dataset). They consisted of animated short files, animal or predator movies, but also some movies some sports movies. The model also showed that overall these movies did not have high popularity scores, but most of them

did have average scores. Based on our knowledge of “The Land Before Time” some of these movies would be fitting, but others are probably not as accurate.

Discussion:

The movie recommender model worked well for movies that were already on the platform (and in our dataset). In order to improve the executive recommendation model (and the user recommendation model) increasing the dataset size would be useful to make sure we have enough similar movies to pull from. Another way to improve the model would be to include the cosine similarity score in the outputs and to possibly include age certification and other variables in the text data.

Rising Star Director Prediction

From the perspective of a producer for Disney, the streaming service industry is incredibly competitive. New streaming services pop up every day, each producing hundreds of new shows each year. Multiple features could have an impact on a consumer’s intent to watch a film, from runtime, to rating, to online review score. The director of a film or television show is the one person that can have the greatest impact on the quality of the work being produced. To stay competitive, Disney needs to invest in creative and popular directors that can produce movies that are well received by the public. It would benefit Disney’s producer to be able to see which directors are produce films and tv show that are the most well received and be able to tell which directors will continue to be well received in the future. For this project we developed the “Rising Star Prediction Script”.

Data Preprocessing:

The goal of this analysis is to create a prediction model that can accurately predict the directors who will have the highest online ratings (TMDB Score) over the next 5 consecutive years. Our first step was to select the features which would have a meaningful impact on the TMDB score. Our choices were: release year, run time, seasons, tmdb popularity, and age rating as these were the variables that were either already numerical or could be transformed into numerical data. For age rating, we assigned a number to each category; R rating would be 4, and Unrated would be 0. We also performed sentiment analysis on the Description section of each movie in the dataframe, creating separate columns for Sentiment & Polarity.

Model Selection:

With our features selected our next task was to select the regression model that would most accurately predict the TMDB score for each Director’s Data. From our testing the Random Forest Regression Model was the perfect model for us due to its low mean squared error score and its adequate r^2 score.

Prediction:

After selecting the model, the next step is to prepare a new set of data to feed into it. In the original Kaggle dataset, multiple directors have worked on multiple films over the years, each with their own unique age rating, runtime, and popularity score. To better process this data, we filtered the entire dataset to display the films and shows that each director has worked on and each year that the film was released.

```
year_pred(12,'Wilfred Jackson')
```

✓ 0.0s

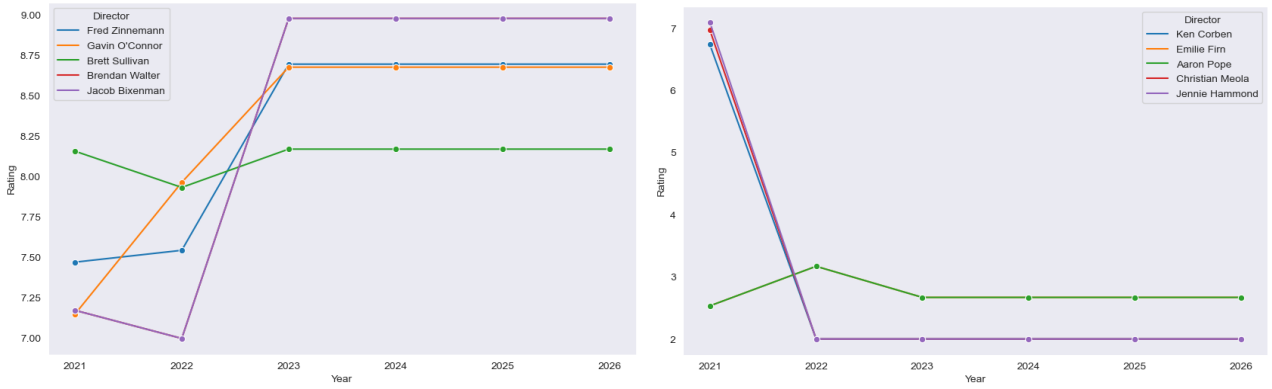
	release_year	runtime	seasons	tmdb_popularity	age_rating	subjectivity	polarity
0	2021	111.056571	0.0	9.526870	0.834078	0.235216	0.041561
1	2022	33.757951	0.0	7.709032	0.949187	0.500221	-0.439724
2	2023	72.872893	0.0	53.950376	0.426347	0.973748	0.303212

After selecting the model, the next step is to prepare a new set of data to feed into it. In the original Kaggle dataset, multiple directors have worked on multiple films over the years, each with their own unique age rating, runtime, and popularity score. To better process this data, we filtered the entire dataset to display the films and shows that each director has worked on and each year that the film was released. This process was repeated on every director in the Data frame, then we applied the Random Forest Regressor to each row of data.

	Year	Rating	Director
0	2021	6.8682	George Seaton
1	2022	6.5280	George Seaton
2	2023	6.5290	George Seaton
3	2024	6.5290	George Seaton
4	2025	6.5290	George Seaton
0	2021	6.9749	James Algar
1	2022	6.9772	James Algar
2	2023	6.6562	James Algar
3	2024	5.9800	James Algar
4	2025	6.7001	James Algar

Results:

This provided us with a Data frame column of predicted ratings. Our last step was to plot the predicted ratings assigned to each director on a line-plot using seaborn libraries and to find the average rating for each director's set of years.



Through this prediction model, we found that the directors that are predicted to receive the highest scores on their content in the next 5 years are: Gavin O'Connor, Jacob Bixenman, Brendan Walter, Fred Zinnemann, and Fred Zinnemann. Opposed to that, the directors that are predicted to create the lowest scoring shows and movies are: Aaron Pope, Emilie Firn, Ken Corben, Christian Meola, and Jennie Hammond.

The top 5 directors are the most likely to continue producing well received films and are directors that Disney Plus executives would benefit from keeping on their platform and providing them with more opportunities to make films for their company. However, the bottom 5 directors are predicted to continue receiving low scores

for their content each year. Executives looking to replace low performing content with newer ones might benefit from removing work done by the bottom 5 directors.

Production Country Prediction

Build a predictive model to predict the TMDb score of movies or TV shows by their production country would be useful to provide valuable insight of film industry and help make better decisions of movie production, distribution and marketing for Disney plus.

Feature Selection:

The first step is to select the relevant features from the dataset. In this case, the features are release_year, runtime, production_countries, tmdb_popularity, seasons, and age_rating, and the target variable is tmdb_score.

Data Preprocessing:

The production_countries feature is categorical, so it is converted to dummy variables using pandas get_dummies function to make it suitable for machine learning algorithms. The selected dataset is split into 80/20 for the training and test sets using the train_test_split function from scikit-learn.

Model Selection:

By comparing the R-squared and root mean squared error scores of the multiple linear regression model, decision tree regression model, random forest regression model and k-neighbor regression model, the Random Forest Regressor model was chosen as the machine learning algorithm for this problem. It is a powerful ensemble learning method that combines multiple decision trees to improve the accuracy of prediction.

Model Evaluation:

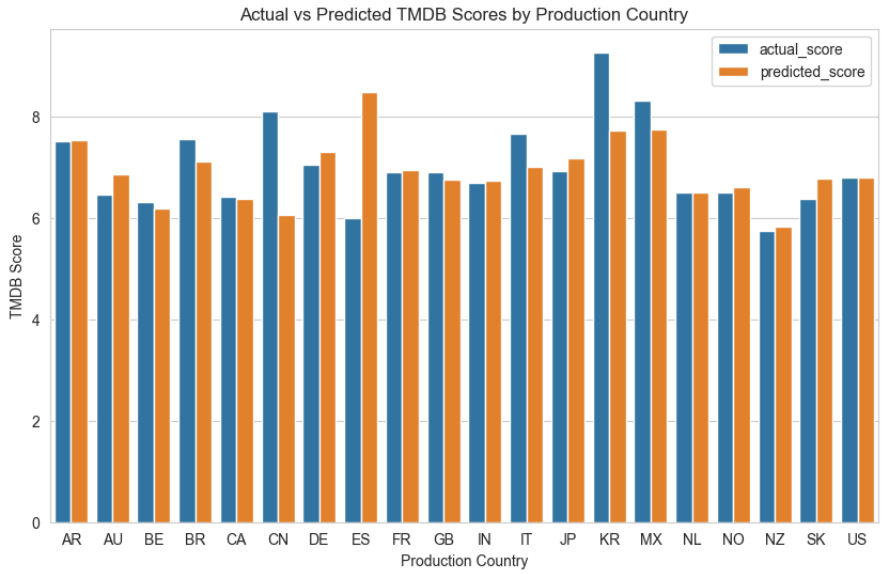
The model is trained on the training dataset, then the trained model is used to make predictions on the test dataset, and the performance is evaluated using two metrics: Root Mean Squared Error (RMSE) and R-squared. The RMSE measures the difference between the predicted and actual values of the target variable, which is 0.95 for this model. while the R-squared measures the proportion of variance in the target variable that is explained by the model, which is 0.30 for this model. The R-squared value is low, which indicates there’re lot’s of unexplained variation in the data.

Result:

Make a table to display the predicted score and actual score of the instances in testing dataset.

	production_countries	actual_score	predicted_score
1253	US	8.478	8.01903
546	US	5.900	6.28156
869	US	6.800	7.21221
1233	IT	8.000	6.85700
525	US	5.900	6.38804
...	...	...	...
552	US	7.638	7.23720
138	US	5.967	6.20360
900	US	7.200	7.33869
602	US	6.800	6.73429
1704	US	9.500	6.54487

Visualizing the testing data result of the actual and predicted TMDb scores for different production countries by bar plot, visually compare the bars, If the predicted scores are close to the actual scores, the bars will be similar in height. If the predicted scores are different from the actual scores, the bars will be different in height:

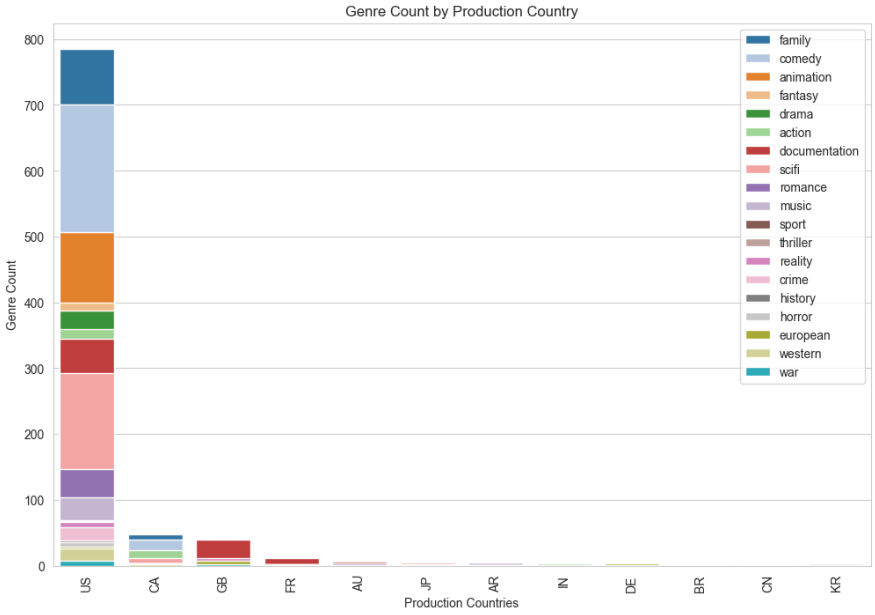


The plot shows the result is highly variable for different countries, so the model's predictive performance may be affected by the production countries of the movies or TV shows.

Discussion:

One possible reason for this highly variable for different countries is that the model may be biased towards certain production countries due to imbalanced or insufficient training data. If the model is trained on a dataset that is dominated by movies or TV shows from a few countries, it may not generalize well to other countries.

Check the total count of movies of each production country by plotting a bar graph, the height of each bar represents the total count of movies or TV shows for that country, and the colors of the bars represent different genres:



It is obviously shown that the feature production country is uneven in this plot. Thus, the model is biased and as a result, the model's predictions are less accurate for other underrepresented countries. To address these issues, collecting more data from diverse production countries may be necessary to balance the representation of different countries.

Conclusion

In conclusion, the dataset we are using in Disney Plus analysis is relatively small, which can limit the accuracy and generalizability of the predictive models. Despite the difficulties, the random forest regression was found to be the most accurate model for all three predictions.

The results show that the random forest regression model can accurately select the best rated and worst rated directors with works on Disney Plus, which could assist the decision making for Disney Plus. Additionally, the recommender model for users and executives produced some good recommendations, but further refinement is needed to improve the result.

In using the Random Forest Prediction, the features that were effective at determining the rating of a director's films were "Release Year, Runtime, Seasons, TMDB Popularity, Age Rating, Subjectivity, Polarity". Currently the best rated director is Jacob Bixenman and the worst rated director is Jennie Hammond

Furthermore, the distribution of the production countries variable is uneven. As a result, the score prediction by country model tends to perform better for the United States than other production countries. Collecting more data from diverse production countries is needed to improve the model.

Overall, despite the Disney Plus analysis emphasizing some potential applications of predictive models, like the director selection recommender system, some further improvements and refinements of the dataset are needed to develop additional prediction models and achieve more accurate and reliable results.