Selecting Movie Features to Predict Monetary Success and Popularity

Kamarion Porter
Harvey Mudd College
kporter@g.hmc.edu

Abstract

In this paper, I seek to discover the similarities and differences between monetary success and critical acclaim in the film industry. To accomplish this, I first find two columns in the IMDB Movies Dataset to represent monetary success and popularity, taking these to be possible output variables. Next, I extract features from the remaining columns; the column that is not being treated as the output variable is included in this group. Then, I use the chi-square statistic in order to test how closely related the features are to the output variable. I also make logistic regression and Naive Bayes models using these features. Finally, I perform a comprehensive analysis on the features and models for each output.

1 Introduction

From their humble beginnings as reels of sewn-together film, motion pictures have grown to become a staple of entertainment in the modern day. Its crafting has become an art form that is cherished by many. However, these films have also cultivated a billion-dollar industry and the goals of the companies at its helm center around making a profit. Sometimes this goal pairs well with providing the audience with a memorable experience, and other times one is sacrificed for the sake of the other. Using the IMDB Movies Dataset, which boasts over 2.5 million movies and TV series, I seek to discover the similarities and differences between monetary success and critical acclaim in the film industry.

2 Related Work

Much work has got in predicting the attributes of motion pictures. Latif and Afzal (2016) [4] applied feature selection and various classifiers, including logistic regression, Naive Baynes, and PART, upon data that they extracted from IMDb in order to predict movie popularity. I also use IMDb data to perform feature selection and classification, however, I employ multiple classifiers to compare multiple outputs. Abidi et al. (2020) [2] extracted a small sample of IMDb data and performed feature selection to find the best inputs and outputs features within a movie for predicting the movie's popularity. I also attempt to predict a movie's popularity using through its features, however, I work with a large dataset featuring movies and TV series. Dhir and Raj (2018) [3] utilized correlation analysis and several machine learning classifiers ob the IMDb 5000 Movie Dataset in order to predict a movie's success, which they represent by the IMDb rating (or score as they reference it). Along with the different strategies that we employ and outputs that we focus on, we differ in that they rely on the IMDb rating as a reliable measure of success, which I disagree with.

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

3 Procedure

3.1 Preprocessing

3.1.1 Choosing Output Variables

My initial intention was to use two of the set's columns, 'gross_income' and 'rating', as my measures of monetary success and popularity respectively. IMDb records gross income as the profit (the total revenue - the total cost) earned by a motion picture. As any net loss results in no profit, net losses are recorded as zero. IMDb's rating system is also simple. Any individual who visits the site can rate a choice film on an integer scale of 1-10. The average of all of the individual ratings becomes the movie's overall rating. In this dataset, the number of people who rated that movie is stored in the column 'votes'. Because two films can have the same rating but a drastically different number of votes, a film's rating, on its own, cannot be used as a measure of popularity.

However, I was able to create such a statistic using the 'rating' and 'votes' columns called the adjusted rating. The adjusted rating is the product of the film's rating - 5.5 and the natural log of the number of votes. The rating is initially subtracted by 5.5 as 5.5 lies at the center of the rating scale, thus any motion pictures rated higher than 5.5 have an overall positive rating, those lower than 5.5 have an overall negative rating, and those at 0 have an overall neutral rating. Votes are a factor within this derivation to account for the popularity/infamy of the film. We then multiply this value by the natural log of the number of votes for the difference between 10 and 100 votes to be more notable than the difference between 10,000 and 10,090 as the former displays a much bigger difference in how well-known the corresponding films are.

3.1.2 Data Filtering

According to Kaggle.com [1], the IMDB dataset, developed by Ashish Jangra, contains over 2.5 million movies and TV series as well as a usability rating of 8.2. However, as I delved further into the dataset, I discovered that the data did not exist as advertised. To start, the year column contained not just the motion picture's start and end year, but also the type of media it was. From this, I learned that in addition to motion pictures (movies and TV series), this dataset contained podcast series, video games, and music videos. Furthermore, within the remaining group of true motion pictures, the vast majority of them had both no gross income and no votes, meaning that the film made no money (potentially resulting in a net loss) and had no ratings on IMDB. After filtering the data once more, I was left with only around 550 thousand shows (many of which had no gross income), however, they all had at least 5 votes. This is reassuring as for a motion picture to have any sort of popularity or monetary success, it has to be known, so the fact that within our dataset, films only have a positive gross income (our measure of monetary success) if they have votes (a part of our measure of popularity) reaffirms the usability of the dataset for this project. The remaining dataset still contained some invalid entries in the form of invalid years (a placeholder of 2100 was used) and invalid ratings (a joke rating of 11.0 was applied to certain shows), however, these made up a very small portion of the data. With this, I concluded the data filtering process.

3.1.3 Normalization and Discretization

One final preprocessing step that I had to take was normalizing and discretizing the columns. A good amount of the columns had unique numerical formats (some with mixed types) and others had multiple categories ('genre', 'director_id', 'stars_id'). Because I had a mix of categorical and numerical data, I discretized the numerical columns by making ordered categories that corresponded to numerical ranges. For each column, the ranges were selected such that there was a proportionate amount of data in each category, and the categories were semantically meaningful. For example, the duration column contains the runtime for a movie, the number of minutes in an episode for an ongoing TV series, and the total number of minutes for a completed TV series. After discretization, the column's categories became hour-long windows, with anything 4 hours and above grouped together. Since the runtime for most TV episodes is under an hour, movies are typically 1-4 hours

long, and most TV series contain at least 4 hours of content, these categories fit nicely to these media types (although not one-to-one). This also leaves room for separation with regards to shorter and longer movies.

3.1.4 Issues with Multi-Labeled Data

For the multi-labeled data, I first tried to make a column that corresponded to each label. Unfortunately, this was only plausible for the 'genre' column as the other columns ('directors_id' and 'stars_id') had too many unique values for this to be feasible. I then tried to use the first label within each entry. Since the labels were sorted by the corresponding director or star's placement in the credits, this would've been an analytically useful feature. However, the caveat to this was that because these columns contain unordered categorical data, I had to encode these columns using a one hot encoding. A one-hot encoding makes a one-to-one mapping between a label and a vector that is the length of all of the unique labels in the set. This resulted in hundreds of thousands of columns being added to the encoded training set. As my laptop could not process this in a reasonable amount of time, I was forced to drop these columns.

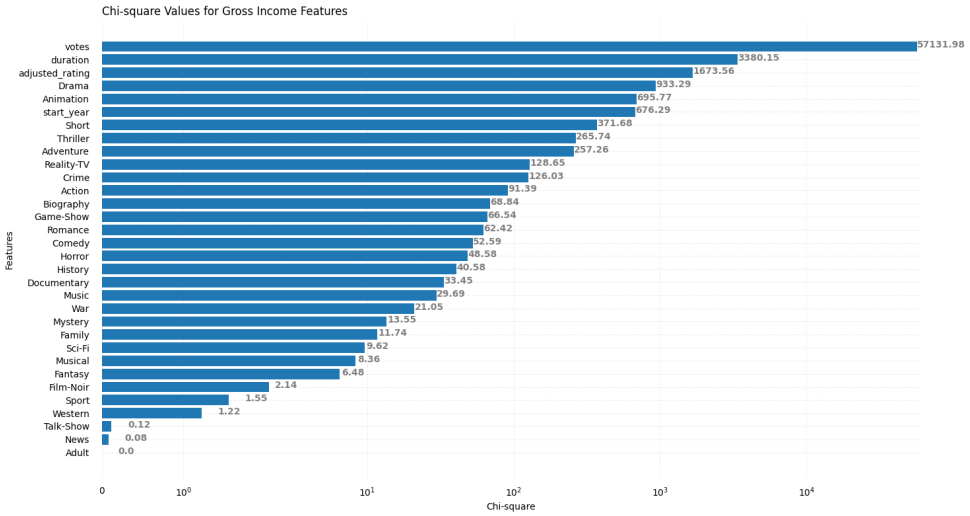
4 Creating Metrics and Models

All in all, the features that I was able to extract from the dataset were 'gross_income', 'adjusted_rating', 'duration', 'votes', 'start_year' (the year in which the show debuted), and all of the genre categories, which (besides the genres) are all ordered variables. Encoding the ordered variables was much easier and less time-consuming than the unordered variables. Using two-thirds of the data as the training set and one-third as the test set, I was able to encode both sets using an ordinal encoding, which mapped each label to a single integer. With the features prepared, I measured the chi-square statistic for each feature. For the 3 features with the best chi-square values, I made a multi-bar chart of the percent frequency of the output variable categories (gross income or adjusted rating) over the feature variable categories. For example, the percentage of entries in the dataset in which the gross income is less than $1,000 given that the duration is less than an hour would be a single bar in a plot if duration was a top feature. And finally, using either all of the encoded features or the three with the best chi-square scores, we fit either a logistic regression or Naive Bayes classifier to the training data, predicted the output values using the testing data, and then evaluated the strength of this prediction by calculating the model's accuracy, balanced accuracy, recall, precision, and F1-score. Besides accuracy and balanced accuracy, these scores are calculated using a macro-average, which means that the given statistic is calculated for each output category, and the average of their percentages is taken as the value. This provides more accurate data regarding each model's effectiveness on imbalanced datasets, such as this one. The balanced accuracy is the average of the micro-averaged recall (the more standard definition of recall) and the true negative rate (the proportion of true negative values over all of the negative values).

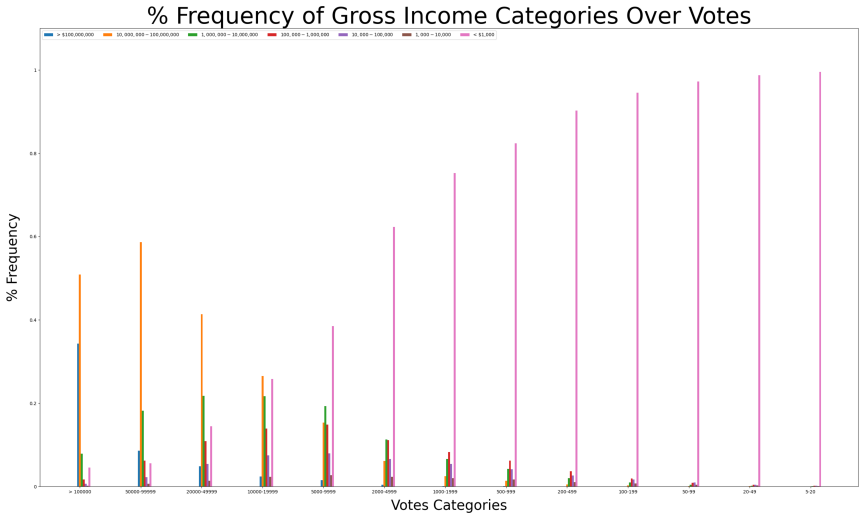
5 Results

5.1 Chi-Square-selected Features

As we can see from horizontal bar plot below, the top 5 features for gross income are votes, duration, adjusted rating, the Drama genre, and the Animation genre, all with large chi-square scores, although votes has the largest by a wide margin. The worst 5 features for gross income are the Adult genre, the News genre, the Talk-show genre, the Western genre, and the Sports genre, with a large jump between the worst 3 features and the rest. Underneath the chi-square plot lies a multi-bar plot of the percent frequency of the gross income categories over the categories of its top feature, which in this case is votes. For example, the chance that a movie in this dataset has a gross income under $1,000 given that it has 100-199 votes is the value of a bar in this graph.

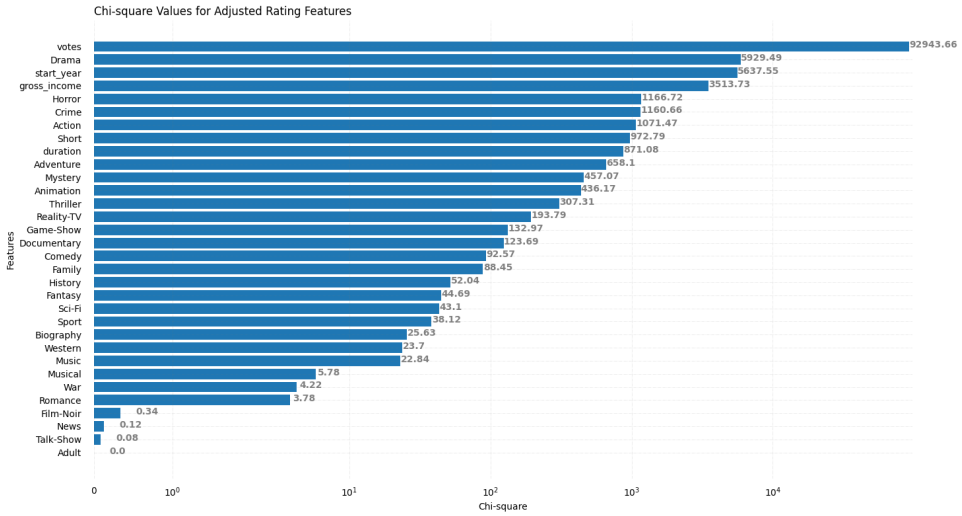


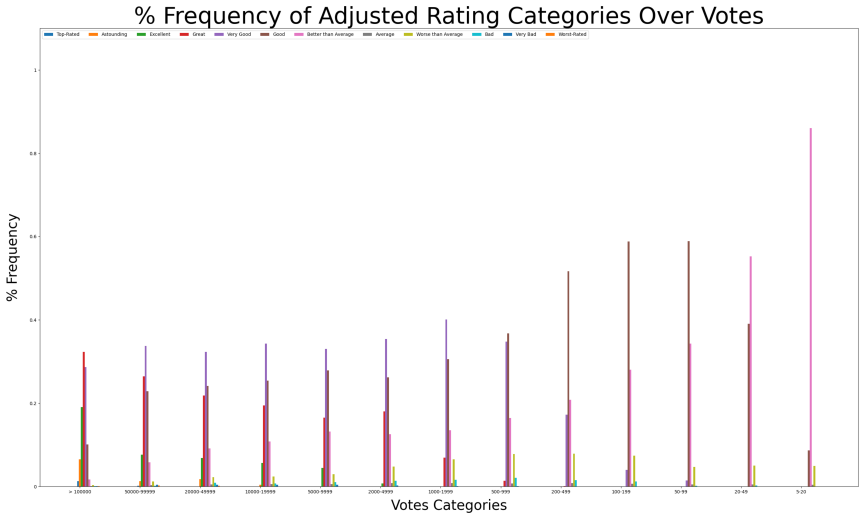
125



126

127 Shifting to the next plot, we can see that the top 5 features for the adjusted rating are votes, the
128 Drama genre, the start year, gross income, and the Horror genre, all with large chi-square scores,
129 although votes, once again, has the largest by a wide margin. The worst 5 features for the adjusted
130 rating are the Adult genre, the Talk-show genre, the News genre, the Film Noir genre, and the
131 Romance genre, with a large jump between the Romance genre and the rest. Underneath the chi-
132 square plot lies a multi-bar plot of the percent frequency of the adjusted income categories over the
133 categories of its top feature, which in this case is also votes.





5.2 Logistic Regression Models

Now, we'll turn our attention to the logistic regression models. The table below refers to the statistics of models used to predict either gross income or the adjusted rating using all of the available features or the top 3 features based on the chi-square statistic.

Statistic (%)	Gross Income (All)	Gross Income (Top 3)
Accuracy	93.82	93.75
Balanced Accuracy	25.72	22.25
Precision	37.55	32.37
Recall	25.72	22.25
F1	27.35	21.52

Statistic (%)	Adj. Rating (All)	Adj. Rating (Top 3)
Accuracy	66.61	62.44
Balanced Accuracy	18.32	13.81
Precision	28.56	15.65
Recall	18.32	13.81
F1	20.02	14.02

5.3 Naive Bayes Models

Now, we'll turn our attention to the Naive Bayes models. The table below uses the same format as the previous one.

Statistic (%)	Gross Income (All)	Gross Income (Top 3)
Accuracy	92.50	93.24
Balanced Accuracy	32.63	27.41
Precision	32.30	28.61
Recall	32.63	27.41
F1	30.84	26.63

Statistic (%)	Adj. Rating (All)	Adj. Rating (Top 3)
Accuracy	64.64	65.26
Balanced Accuracy	21.13	16.28
Precision	22.01	16.96
Recall	21.13	16.28
F1	21.18	15.69

6 Discussion

The most apparent discovery from this dataset is that the number of votes is the most powerful feature for both gross income and the adjusted rating. Now, with regards to the adjusted rating, while votes is one of factors used to calculate it, the adjusted rating can be positive or negative based on how high or low the rating is. As such, it is only guaranteed that the magnitude of the adjusted rating is proportional to that of the number of votes. Furthermore, from the percent frequency plot for adjusted income and votes, as the number of votes increase, the percentage of larger adjusted rating scores increases. This implies that there is a positive correlation between the two.

It is also interesting that gross income and adjusted rating both had high chi-square values with respect to the other variable, earning third and fourth on the chart respectively, which implies a reflexive relationship between the two. Many shows become profitable because they're good, and many shows become good because they're profitable. While the former idea is pretty easy to comprehend, the latter could use more of an example. Take the somewhat recent Sonic the Hedgehog movie. An early trailer used to promote the movie became infamous because of how bad the live-action Sonic looked. Because of the public backlash, the movie's release was delayed in order to improve the film, and it became a well-received movie, which for a video-game adaption was a rare

feat. These improvements only occurred because of the profitability of the Sonic the Hedgehog IP. The movie's high earning-potential made it better.

One surprising difference between the two outputs is that while the chi-square values for the adjusted rating are generally higher, the logistic regression and Naive Bayes models for gross income performed much better in every statistic. This implies that while the features cover more of the variation with regard to the adjusted income, the features are better suited for predicting gross income (at least within these models), which is strange. Despite these differences, the models for both outputs also share quite a few similarities. With the exception of accuracy, their statistics are all very low with precision being the next highest and balanced accuracy approximately equal to recall. Note that the difference between precision and recall is much larger across the logistic regression models compared to the Naive Bayes models. Another similarity across all of the models is that those that only used the top 3 features had lower statistics than their counterparts that used all of the features but not by a wide margin (aside from the precision metric for the adjusted rating). Another thing to note is the accuracy being so much larger than any other statistic for all of the models reveals how imbalanced this dataset is. While the performance of these models is lack-luster at best, they do have a saving grace in that precision is its highest, accurate metric. When attempting to find motion pictures of a certain quality (high income, low performance, etc.), it is more important for the results to be accurate rather than comprehensive. Therefore, I conclude that the logistic regression model performed best in all cases due to its higher precision. Also, the lack of a significant difference (with one exception) between the full models and the 3-feature models displays how strong the relationship between the top 3 features and their output is.

7 Conclusion

My goal for this project was to discover what features go into making a monetarily successful motion picture and a popular one then compare their similarities and differences, and I believe that, in a way, I succeeded. I discovered that these two pursuits had a lot more in common than I initially thought, and that surface-level information about a show is ultimately not enough to determine whether it will be good or profitable.

8 References

[1] https://www.kaggle.com/datasets/ashishjangra27/imdb-movies-dataset

[2] Abidi, Syed Muhammad Raza, et al. "Popularity Prediction of Movies: From Statistical Modeling to Machine Learning Techniques - Multimedia Tools and Applications." SpringerLink, Springer US, 6 Jan. 2020, https://link.springer.com/article/10.1007/s11042-019-08546-5.

[3] Dhir, Rijul, and Anand Raj. Movie Success Prediction Using Machine Learning ... - IEEE Xplore. 2018, https://ieeexplore.ieee.org/abstract/document/8703320.

[4] Latif, Muhammad Hassan, and Hammad Afzal. Prediction of Movies Popularity Using Machine Learning Techniques. Aug. 2016, https://www.researchgate.net/profile/Hammad-Afzal/publication/311913687_Prediction_of_Movies_popularity_Using_Machine_Learning_Techniques/links/586253ce08ae6eb8
of-Movies-popularity-Using-Machine-Learning-Techniques.pdf.