Which Movies Endure and Why?

Department of Computer Science,Stony Brook University

December 6, 2019

Abstract

This project is aimed at constructing a model which can predict if a movie endures over time or not based on the variables known at its time of release. We have used various features to design a scoring function ζ_{RP}, which gives us the popularity of a movie at the time of its release. By comparing it with the current popularity, ζ_{CP}, which was also obtained using a scoring function, we calculated the endurance metric ε, which gives us whether a movie endures over time or not. We then build our baseline regression model to predict the endurance metric ε, based on ζ. We then used a SVM classifier to say if a movie will endure or not based on the parameters at the time of its release.Accuracy of 82% was obtained using SVM.

Introduction

One of the biggest challenges for any production houses is to make movies which don't fade away with time, i.e. movies which endure. Also,whenever a new user registers himself on any movie website like *Netflix* or a book website like *goodreads*, the recommender systems on the website provides him with the most popular movies or the books, i.e. the ones which have stood the test of time.

A lot of research has been done on prediction of the success, IMDb the rating, the box office collections of a movie. Most of them include user ratings on different movies, whereas, some of them use social media (e.g. YouTube, Twitter etc) for prediction. However, less work has done the popularity of the movies over time. The movie popularity is an abstract entity in itself and therefore, it is paramount that we define popularity at various levels. Even in many of such works which define popularity, it is the number of ratings (or the number of user votes) provided by users for movies and consider it as an indicator of the popularity of movies. While this could be a powerful indicator of the popularity, it still doesn't answer which movies endure and why? Our work in this research project aims to analyze which movies endure with time and the reasons concerned with it. For this, we use the IMDb Dataset which is the most comprehensive Internet Movie DataBase of information related to films including crew,cast,vote count, average rating, genre. We merged this this with various other datasets such as the Social Media data, Numbers, the BoxOfficeMojo, the OMDb, all of which obtained by scraping. Below, we explain the need to scrape many of them and the role they played in our research. We have explained in further sections how we have used various features to design a scoring function ζ, which gives us the popularity of a movie at the time of its release. By comparing it with the current popularity, we calculated the endurance metric ε, which gives us whether a movie endures over time or not. We then build our baseline regression model to predict the endurance metric ε, based on

ζ. We then used a classifier like an SVM to say if a movie will endure or not based on the parameters at the time of its release.

Literature Survey

[**1**] examines the genre trends in the top 50 grossing films the US box office each year from 1991 to 2010.Focuses on the frequency and rank of different genres and the box office gross.The trend in the time series of the frequency with which a genre occurs in each year is described by linear ordinary least squares(OLS) model, y=at+b where the independent variable t is the time of release.Reduced major axis(RMA) regression was used to predict the relationship between the ranks of a film's opening and total gross.Concludes that the range of genres for highest grossing films has become narrower over the past twenty years and that different genres are characterized by different release patterns.
[**2**] uses hybrid visually-driven features which represent the movie content to predict movie popularity and average rating of the movie.The movie was split into shots and a frame is selected as representative of each shot.From this frame the features were extracted and then aggregated to build an individual feature vector that is descriptive of the movie.These features are then fed to a gradient booster model for training and prediction.It was identified that popularity and average rating were positively correlated and naturalness ,which is one of the visual features was highly correlated with average rating.
[**3**] built a Naive-Bayes model that uses character and plot based user-identified trope labels to predict audience rating for movies.After comparing the accuracy of various models they conclude that the narrative tropes used to create a movie are not very predictive of the final rating or quality of the movie.
[**4**] predicts movie box office performance using followers count on twitter and sentiment analysis of YouTube viewers' comments.The predictions were labeled as three classes,Hit,Neutral and Flop using K- Means clustering.The following patterns were identified: a new movie in non-popular genre and an actor with low popularity resulted in a flop ,popularity of leading actress is crucial to the success of the film.
[**5**] evaluates if the success of a film can be predicted well with data obtained through conventional methods such as web databases or with data obtained from social media such as Twitter,Youtube.The results demonstrate that the sentiments harnessed from social media can predict the success with more accuracy than that of using conventional features.A blend of both outperformed the existing approaches.

Datasets

The dataset used was generated from :

- IMDb dataset
- IMDb metadata
- OMDb API
- TMDb data
- Popularity of actors, directors from IMDb.
- BoxOfficeMojo
- News API data

The data was retrieved from IMDb , OMDb , TMDb, Wikidata and Kaggle awards Dataset.These were merged using the imbd id column. The final data set had columns such as movie title, actor, genre, awards, keywords, vote count ,vote average, movie type, runtime. We had issues dealing with the unavailability of the revenue and the budget data. We had to scrape the gross revenue data from the BoxOfficeMojo and the budget data from the Numbers. Inspite of this, many of its values were missing. Also, we had exported the most popular directors and actors for each decade and replaced the directors and actors with the popularity metric. Further more, we had also scraped the NewsAPI data to get the number of news articles published, the number of photos available online and the number of posters published, all of which help in identifying the popularity at the time of its release. The IMDb metadata gives us parameters like the likes on various social media websites which helps us in finding the current popularity. This can be validated with the popularity metric on the TMDb to adjust our scoring function accordingly.

Data Pre-processing

Apart from the data pre-processing which we had performed during the proposal and the progress reports like the name mismatches, redundant features(while in case of merging two different data sets).

1) There were multiple genres associated with the movie in almost every data set. We had split each genre and considered the top 2 genres and had label encoded accordingly. Also, we had given a metric to each of this genres considering the time period and the popularity of the genre in that particular time period.
2) The director and actor popularity charts had given us the values only for the top 500-600 directors. When we tried to merge it, a lot of directors had a NaN values. Therefore, we gave it a value equal to the half of the minimum value.
3) The CPI index has been used to consider inflation and the budgets and revenues had been accordingly adjusted.
4) The old movies were rated 1-4 while the current movies are rated between 1-10 and there is a need to scale it. The inflated profit was taken into consideration while scaling the ratings.
5) In the TMDb credits dataset , the columns cast and crew were in json format.Same was the case with genres, keywords, production companies and production countries columns that were present in the TMDb movies dataset.To extract this data and store it as a string with comma being used as the separator we had to write down a function. $json.loads()$ method was used to load this data.

Features Extracted

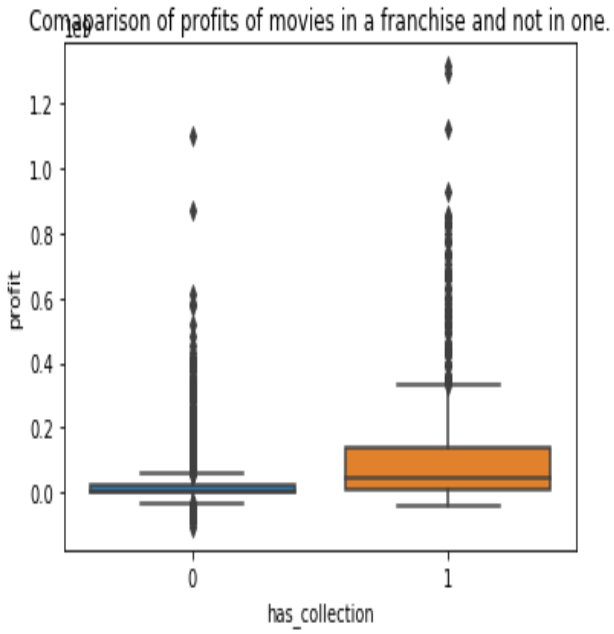
- IMDbId
- Budget
- Revenue
- Genre(For movies with multiple genres, top two genres were used)
- IMDb Rating
- IMDb Views
- Actor current popularity

- Director current popularity
- Movie current popularity
- Number of critic reviews
- Number of user reviews
- Number of awards
- Number of nominations
- Number of articles published
- Number of photos available

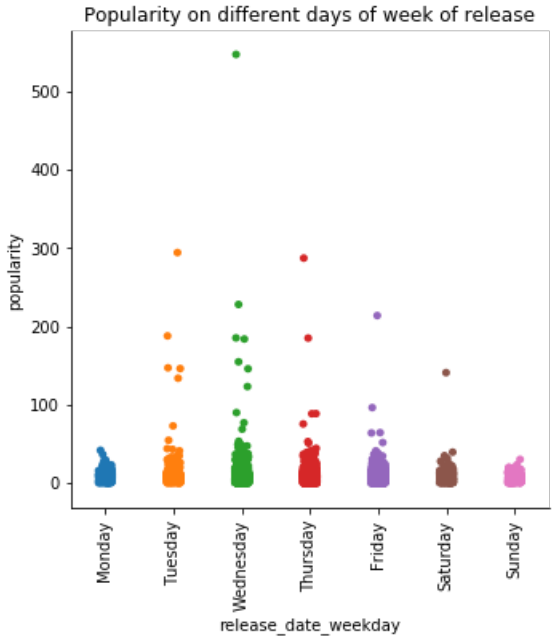
Exploratory data analysis and Feature Engineering

We have created four new features during feature engineering. The four features are:

- $len\_tagline$
- num_cast
- $'release\_date\_weekday'$
- $'has\_collection'$



((a)) Figure 1

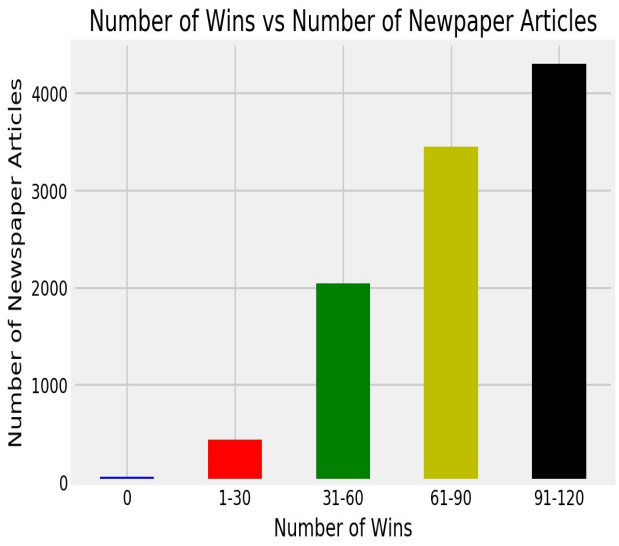


((b)) Figure 2

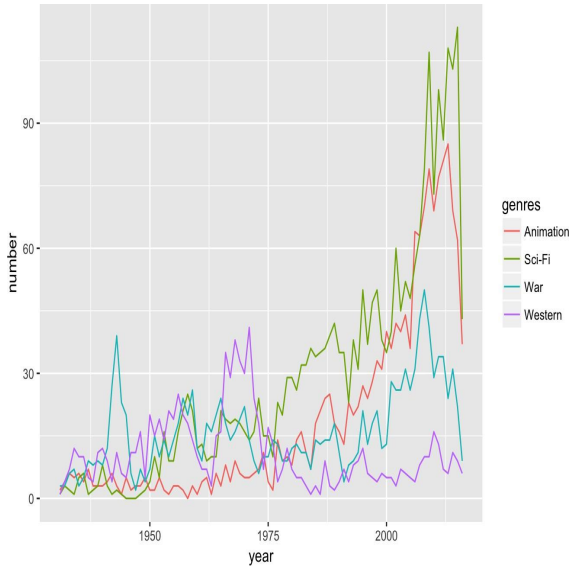
Having obtained the tagline and the cast from the data, to make use of it, we calculated the length of the tagline ,$len\_tagline$, ie the number of words and the number of cast members, num_cast. While building

the model to predict endurance, we calculated the feature importances and we saw that both the new features, $len\_tagline$ and num_cast, were in the list of top 10 features with scaled values around 53.87 and 72.33 while the most important variable had a value of 183.7 The feature '$has\_collection'$ groups various movies based on if it's part of some franchise or if there exists a sequel or a prequel and every such movie is a part of that collection. The plot as shown in figure 1 is a boxplot where it clearly indicates that the movies with a sequel or the ones which exist as a part of franchise gain more profits and therefore, stay more popular than the ones which aren't a part of a collection.

We have created a new feature '$release\_date\_weekday'$ based on the date of its release. We plotted the popularity of a movie given in the TMDb based on the day of release as shown in Figure 2. Surprisingly films releases on the weekdays have more popularity than the ones released on the weekend.



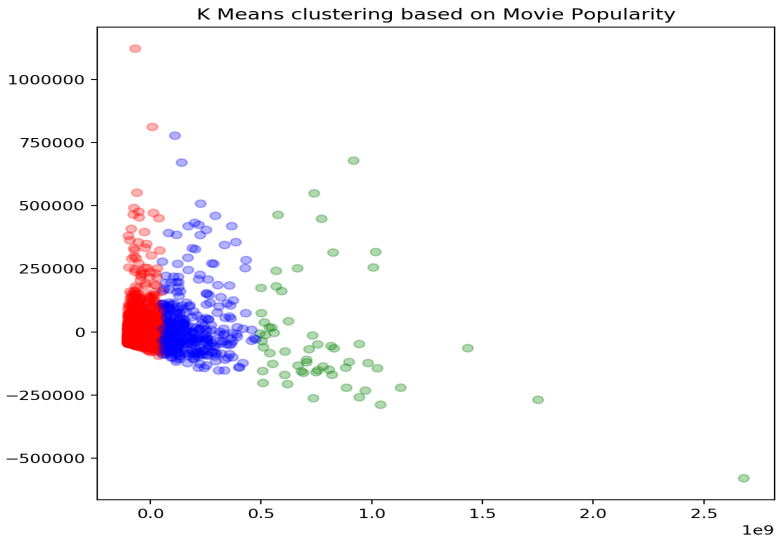
((c)) Figure 3



((d)) Figure 4

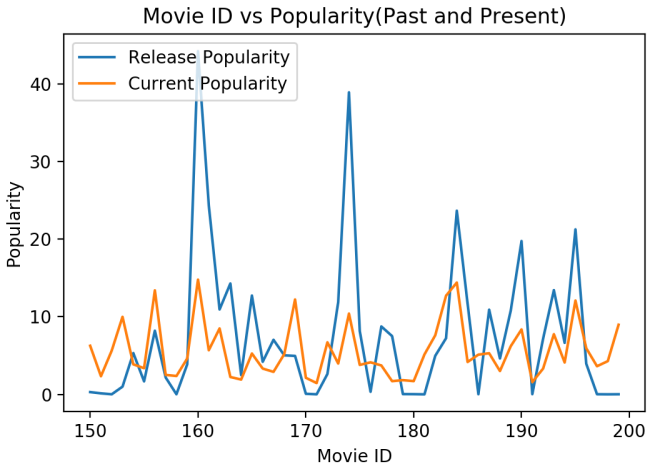
A user defined function was written to segregate the number of wins into few bins.Once,this was done we calculated the median of number of newspaper articles for these specific bins.These were plotted to see that as the number of wins increased the number of newspaper articles published were also more as seen in Figure 3.

Plot 4 shows us how the popularity of various genres has changed over time with a rise in the Sci-fi and a fall in 'western'.



We have used K means clustering depicted in Figure 4 to show movies that are close to each other in terms of properties that determine the popularity of a movie. The properties taken into account are awards won by movie, critic reviews, number of mentions in newspaper articles, director popularity, Imdb views and score, user reviews, budget, and revenue. As feature set is larger than two, we use PCA, taking five principle components and reduce the dimensions of the feature space to be able to use K means clustering.

Scoring Functions



After the feature engineering where we extracted the important features as well, we wanted to analyse the endurance metric. To calculate the endurance metric, we have written a couple of scoring functions. ζ_{RP} : This is the popularity of a movie based on the release parameters such as the Budget, Revenue, Genre, Number of awards, Number of nominations, Number of critic reviews,Number of articles published and photos available.

$$\zeta_{RP} = \Sigma_{i=0}^{n} \quad \widehat{C_i * \hat{F}_i}$$

ζ_{CP}. : This is the popularity of a movie based on the parameters such as the rating, IMDb views, current popularity of the actor, director, number of user reviews.

$$\zeta_{RP} = \Sigma_{j=0}^{n} \quad \widehat{K_j * \hat{Y}_j}$$

Where, C_i, K_j and F_i, Y_j are coefficients and the features respectively. Each feature F_i, Y_j is normalised to $\hat{F}_i, \hat{Y}_j$ respectively as the parameters like budget, revenue, critic reviews, can significantly impact our function. Later, the value obtained is normalised using a min-max scaler.
Now, we define endurance as,

$$\varepsilon = \zeta_{RP} - \zeta_{CP}$$

If the endurance ε value is negative, i.e if we have the normalised current popularity greater than than the normalised release time popularity and if the $\varepsilon <$ -20 , then the movie has endured over time. On the other side, if the endurance is positive and the $\varepsilon >$ 20, then the movie has dipped significantly with time. For all the other values, we cannot say for sure if the movie endured.
The above plot compares the normalised release time popularity of the movie computed before with the normalised current popularity. We can see that the release time popularity is generally higher and the current popularity in 2019 is significantly lower. However, for most of the movies, the difference is not so high which indicates the movies that has endured over time. For some of them, the difference is significant which can suggest that the endurance is low.

Model Buidling

Linear Regression

We model our data using linear regression. We take a set of twelve features which are based on the production of the movie and the kind of reception the movie received(critic and user reviews/hits) to predict whether a movie will endure or not. We also use the correlation matrix to find the features that correlate well with popularity. We convert the categorical values to numerical values for genre, cast and director. We use L2 regularization(Lasso Regression) to penalize the features with large coefficients. It shrinks the coefficients of the less important features to 0, and in our model we notice that some production features such as production house and runtime do not affect the prediction on endurance. Through k cross validation we notice that the lambda with value 1 gives least root mean square error. **The root mean square error we get on predicting endurance is 7.299**

Random Forrest Regressor

We also use random forest regression to model the data. In this approach we make use of bagging, which trains each decision tree on a different data sample, where sampling is done without replacement. We combine multiple decision trees, to predict a single popularity. We choose appropriate values for the hyper-parameters max depth of tree and number of trees in the forest. We use K cross validation again to tune the hyperparameters and we obtain max depth of tree as 5 and number of trees as 100, and use mean absolute error as the error metric to measure how well the model performs. **The root mean square error we get on predicting endurance is 6.828**

Support Vector Machine

We build a model using Support Vector Machine(SVM) to predict whether a movie has endured or not. We convert the task into a classification task by converting the endurance value into three separate classes based on whether the value is positive or negative and the change in popularity between the two time periods is over a given threshold. Using SVM and setting the values of γ and C appropriately, we get a non linear decision boundary separating the three classes. SVM maximizes the margin between the decision boundaries of the three classes by finding the maximum margin separating hyper plane. We set the value of γ to 1 and value of C to 100. Increasing the value of γ tries to fit the training data better and can lead to over fitting, and the term C is the penalty parameter for the error term. **Using SVM we get an accuracy of 82% on the validation set.**

Sniff test

Having obtained the endurance values, we sorted it in the ascending order with more negative implying more endurance. By considering the top 100 movies and the bottom 100 movies, we performed a sniff test by considering the popularity index obtained from the TMDb API. After sorting the movies based on TMDb popularity, we compared how they fared with one another. We saw that the movie 'Shawshank Redemption' with a high popularity metric was not found in the top 100 but had a high popularity index in the TMDb dataset. The same was the case with 'Fight Club' . From this we could conclude that our model wasn't able to calculate endurance accurately for movies that did not do well at the time of it's release but went on to do well a few years later.

For each decade , we considered the top 10 movies and compared the genre of these movies with the most popular genre of that decade.We could see that for all decades except for three ,maximum movies had the most popular genre as their first genre or second genre.

References

1. Using Crowd-source based features from social media and Conventional features to predict the movies popularity by M.Ahmed,M.Jahangir,Dr.Hammad Afzal,Dr. Awais Majeed,Dr.Imran Siddiqi
2.Predicting movie success from tweets by Rose Catherine, Sneha Chaudhari
3. SOCIAL MEDIA,TRADITIONAL MEDIA AND MUSIC SALES by Sanjeev Dewan , Jui Ramaprasad,

K. Apala, M. Jose, S. Motnam, C.-C. Chan, K. Liszka, and F. de Gregorio.
4. Prediction of movies box office performance using social media In Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, pages 1209–1214, Aug 2013
5. Predicting Movie Popularity and Ratings with Visual Features by Farshad B. Moghaddam, Mehdi Elahi, Reza Hosseini, Christoph Trattner ,Marko Tkalčič.
6. Predicting the movie popularity using user-identified troops by Amy Xu,Dennis Jeong