Predicting the opening weekend revenue of a movie based on pre-release data

Abhay Mittal
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
abhaymittal@umass.edu

Ankur Aggarwal
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
aaggarwal@umass.edu

Abstract

The movie industry is a billion dollar industry and has a lot of risks associated with it. A lot of time and budget is spent on creating a movie and while some turn out to be profitable by huge amounts, there are many which either go in loss or have meagre profits. In this project, we look at some of the factors that are available before release to see the impact they have on the opening weekend revenue of the movie. We also incorporate page view statistics from Wikipedia for 8 weeks before the release of the movie as a measure of the popularity of the movie. Using the data from 2007 to 2013 we are trying to predict the opening weekend revenue for the movies from 2014 to 2016.

1 Introduction

We all love movies. Watching movies is a preferable way to spend time for a lot of people. Movie industry is a huge industry with an expected global revenue of about 38 billion U.S. dollars in 2016. More than 1.2 billion movie tickets were sold in the U.S. in 2015. There are about 5,800 cinema sites in the U.S. as of 2015 [3]. Creating movies is a time intensive process and the rewards are not always fruitful.Apart from the money spent in making the movie, there is a huge risk involved in investing money in advertisement and booking of cinema sites.For example production houses have spent around $723.5 million [5] in advertising alone in 2015.A good prediction of opening weekend revenue helps production houses make rational decisions with regards to booking of movie cinema sites and spending on advertisement and save millions of dollars.This problem is difficult as there is a lack of structured data for its analysis.Hence, it is a very practical and challenging problem.

In order to solve this problem we collected data movie details from The Internet Movie Database (IMDb) using The Open Movie Database (OMDb) API [4], movie revenue data from BoxOfficeMojo [2] and page view data [7] of 8 weeks prior to the release from Wikipedia. Revenue of Hollywood movies have inherent clusters in between them and each cluster defines a range of revenue which is very different from that of other clusters. We used K-Means clustering to identify those inherent clusters and then used Random Forest Classifier to predict the revenue range of future movies with an accuracy of around 39%.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

2 Related Work

Opening weekend revenue is highly dependent on the popularity of the movie amongst the viewers for its release. Previous work by Mestyán et al. [15] shows Wikipedia Page Views to be an important metric for measuring the popularity of the movie. Mestyán et al. [15] built a multivariate linear regression model for predicting the box office revenue using features like number of users,collaborative rigor and number of edits.His analysis for a single year(2010) met with less constraints on the availability of data in comparison to ours.Moreover, using number of theatres as a predictor does not suit our problem as we want our revenue predictions as an indicator of the number of theaters to book.

Movie genre can also be a good metric for consideration as stated by Sharda and Delen [17].They trained a neural network to process pre-release data, such as quality and popularity variables, and classify movies into nine categories according to their anticipated income, from “flop” to “blockbuster”. For test samples, the neural network classified only 36.9% of the movies correctly, while 75.2% of the movies were at most one category away from correct.Sharda and Delen [17] used fixed buckets of movie revenue in their classification. This assumption on the revenue range introduces rigidity in the classification as these ranges are imposed on the data. On the contrary, unsupervised clustering on the revenue on the logarithmic scale gives more dynamic clusters.Sharda and Delen [17] have come up with two interesting features : competition and sequel, which we will explore in our future work.

Asur and Huberman [8] in their work have demonstrated the demonstrated how sentiments extracted from Twitter can be further utilized to improve the forecasting of movie revenue.While the power of twitter demonstrating true popularity of a movie is undeniable, in our analysis we are constrained by the accessibility of the data as we are analysing 10 years of movies.

We strongly believe that cast members and their recent successes or failures bear a strong impact on the coming movie. For example if a director has a history of successes, some judgement can be made about the revenue of its upcoming movie. The importance of using actors,directors and writers in predicting movie has also been explored by Flora,Lampo and Yang of Stanford University in their CS 229 project [6].

3 Datasets

We considered the following datasets:

3.1 BoxOfficeMojo

We scraped BoxOfficeMojo [2] to get revenue information of the movies. We wrote a scraper using Beautiful Soup [1] in python and first grabbed the complete alphabetical list of movies in BeautifulSoup. The list had 16,539 movies. Since we had Wikipedia data regarding page views post 2007 only, we removed the movies which were before 2007 leading to a set of 7,020 movies. The next step was obtaining revenue information. We had initially planned to get the opening day revenue for our project but very few number of movies had that data. So we opted for opening weekend revenue and scraped that. Our final data from BoxOfficeMojo consisted of 6,166 movies which we stored as a csv file using pandas [14] library for python. We used MinMaxScaler by scikit-learn to scale the page views to [0,1]

3.2 The Open Movie Database

We started with a list of movies provided by IMDb and filtered out the movies between 2007 and 2016. We then scraped each of the movie title in OMDb API [4] and got details for 7660 movies.The details of the movies included date of release ,director(s), cast, writer(s), language, country and type. We filtered this data for movies released in USA and in English language, and for which we had revenue information; giving us 3831 movies.

3.3 Wikipedia

For the above 3831 movies detail we scraped the Wikipedia dataset of the page views [7] to get page views for each movie for last 8 weeks before its release.We had to scrape the Wikipedia page using

Table 1: Feature summary

Name	Description	Type
Month	The month of the release of movie	One Hot Encoded boolean
Mean Page Views for Weeks 1-8	Average page views on the Wikipedia page of the movie for 1-8 weeks before release. (one separate column for each week)	Float
Director	The director(s) of the movie	One Hot encoded boolean
Actor	The actor(s) involved in the movie	One Hot encoded boolean
Writer	The writer(s) of the movie	One Hot encoded boolean
Genre	The movie Genre	One Hot encoded boolean

BeautifulSoup [1] to extract the relevant information. We were able to retrieve 2,155 movies out of the 3831 movies we got from OMDb.

3.4 Complete Dataset

The final step in data collection was to merge the data obtained from the above sources and define the features. After merging, the final number of data samples was 2,155. We took the movies released in 2007-2013 as our training dataset and the movies released later as test. The features that we considered are in table 1. Our data set contained 26 genres, 1842 directors, 5246 actors and 3337 writers. The total number of features in our dataset after doing one hot encoding turned out to be 10,481. This was a lot of information considering the number of samples we had. Hence we had to do feature selection. (See 4)

4 Proposed Solution

4.1 Clustering to convert the problem to classification

We experimented with some regression techniques initially but they generated poor regression scores. Thus, we converted our problem of regression into classification which has been found to give better accuracy and faster learning algorithms [17] [12]. This concept of ’discretization’ also increases the interpretability as we can consider the movie in the least revenue class as flop and the movies in the highest revenue class as blockbuster [17]. Thus, our pipeline consisted of a K Means clustering algorithm which took revenue as input (only for the training data) and then clustered them into classes. Hierarchical Agglomerative Clustering did not work as its scikit-learn implementation requires the input data to have atleast two features. KMeans algorithm divides the data into K mutually disjoint clusters C. The centroids of the clusters are selected in such a way that the within cluster sum of square distances is minimized [16], i.e. we find:

$$\sum_{i=1}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

where μ_j represents the center of cluster j. This clustering method has the advantage of being very fast and easy to implement. However, it is sensitive to initial cluster assignment and can converge to a local minimum. To tackle that, we run the algorithm five times and take the best result.

4.1.1 Determining the number of clusters

To decide the number of clusters, we used within cluster variance as our cluster quality measuring method along with the elbow criterion. Thus, we plot the within cluster variance for varying number of clusters and see the elbow point. We further corroborate our results using the Davies Bouldin Index ([11]). The details are mentioned in section 5.

4.2 Feature Selection

The next step involved feature selection. Since we had a large amount of features, this step was crucial. We decided to experiment with Recursive feature elimination and SelectKBest methods of scikit-learn [16]. But due to a huge number of features, recursive feature elimination was taking too long to run and we had to drop it. SelectKBest is a univariate feature selection method which scores the input on the basis of univariate statistical tests and selects the K Highest scoring features. We used the chi-squared statistic for scoring the features [16].

4.2.1 Determining the number of features

To select the number of features, we searched over a range of features and trained classifiers using 5-fold cross validation. Our results and plots are mentioned in Experiments (See section 5)

4.3 Classification

For the final classification step, we experimented over a large number of classifiers (See section 5) and found random forest classifiers [9] to give the best accuracy during cross-validation. A random forest classifier is an ensemble of decision trees. A decision tree classifier makes predictions for a point by assigning to it the most common class in its region [13][10]. Gini index or Cross entropy are the general criterion used to decide where to split the data. If we consider K to be the total number of classes, N_m to be the number of samples in region m and p_{mk} to be the proportion of observations of class k that are in region m, i.e.

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

The Gini index is calculated as:

$$G = \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$$

and measures the node purity. The smaller the Gini index, the more is the purity. The cross-entropy is calculated as:

$$D = -\sum_{k=1}^{K} p_{mk} \log p_{mk}$$

Generally the values of cross-entropy and Gini index are similar [13]. It fits a large number of decision trees on sub samples of the data and makes predictions by averaging out the results of independent classifiers. Random forests have better performance than Bagged trees as they make splits by considering only a subset of the predictors and thereby decorrelating the trees [13].

4.3.1 Determining the optimal Hyperparameters

We chose to optimize the maximum number of features , minimum samples that must exist in a node to consider it for a split and the minimum number of leaf nodes as our hyperparameters. The number of estimators for the random forest was fixed to 100 as it seemed good enough comparison between training time and accuracy. We used grid search, i.e. we selected every possible combination of the above hyperparameters and did 5 fold cross validation to find the optimal values. The experiments are detailed in section 5

5 Experiments and Results

5.1 Reasons for selecting log-revenue for clustering

While clustering directly on the basis of revenue we found that more than 65% of our data lied in a single cluster only (\$100 to \$3,000,000 revenue) only and thus the trained classifier mostly predicted this cluster and achieved a test accuracy of 60-63% (test data also followed the same distribution). On clustering on the basis of log-revenue we achieved a more even distribution of data but it also lead to a reduction of the accuracy. The comparison is done in figure 1. Also, the minimum and maximum log revenue assigned to each cluster can be seen in the right half of figure 2.

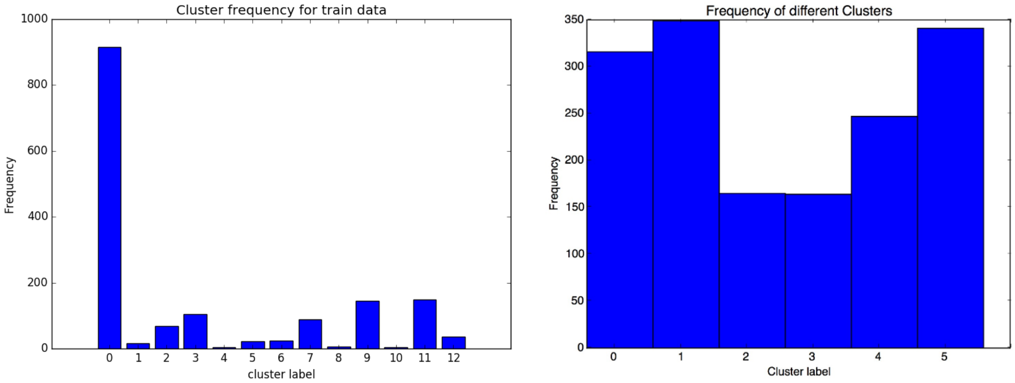

Figure 1: Histogram showing: Left- the data distribution among the clusters when clustering on basis of revenue; Right - the data distribution among the clusters when clustering on basis of log-revenue

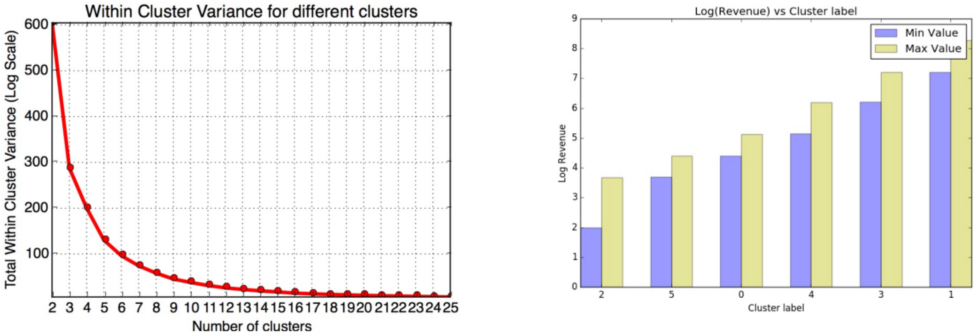

Figure 2: Plot showing: Left- the within cluster variance with varying number of clusters, Right- The minimum and maximum values of log-revenue in each cluster

5.2 Determining the number of clusters

As discussed in section 4 we used the elbow method to determine the best number of clusters. The graph generated can be seen in figure 2. As can be seen, the elbow of the graph is somewhere around 6 which we further corroborated using the Davies Bouldin Index [11] which was calculated to be around 0.58.

5.3 Determining the number of features to select

As discussed in section 4, we used SelectKBest with chi-squared statistic from scikit-learn [16] to determine which features to use. We observed that after around 2500 features, the accuracy used to stabilize and there were very small changes which varied in both directions (See figure 3)

5.4 Selecting a classifier

We did 5-fold cross validation to select a classifier among the following: Random Forests (RF), Logistic Regression (logit), K Nearest Neighbor (knn), Multinomial Naive Bayes (mnb), Linear Discriminant Analysis (lda), Quadratic Discriminant Analysis (qda), Ada Boost (ada). The cross validation accuracy for these classifiers is show in figure 4 As per our results, Random Forest Classifiers were found to be the best performers.

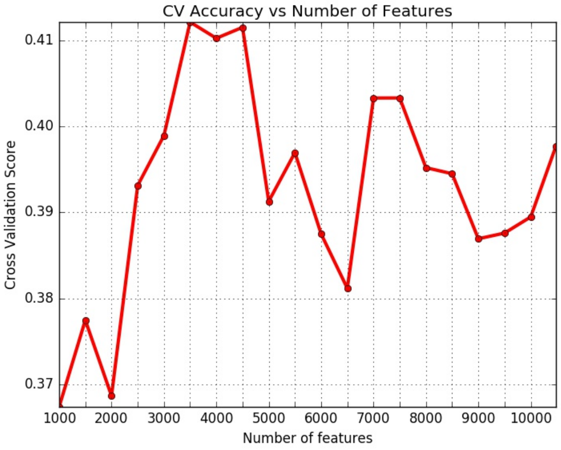

Figure 3: Plot showing the cross validation accuracy vs number of features

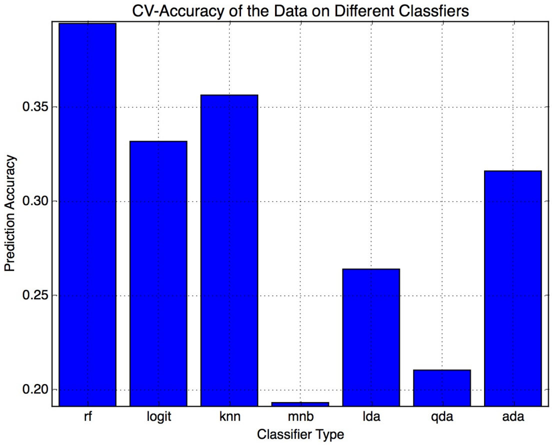

Figure 4: Plot showing the cross validation accuracy for Random Forests (RF), Logistic Regression (logit), K Nearest Neighbor (knn), Multinomial Naive Bayes (mnb), Linear Discriminant Analysis (lda), Quadratic Discriminant Analysis (qda), Ada Boost (ada)

5.5 Hyperparameter Optimization

We considered the following hyperparameters:Max number of features to consider when making a split, Minimum samples that are required to split an internal node and Minimum samples for leaf nodes. To optimize the model, we first considered a coarse range of hyperparameter values to search for and then fine tuned them. In our optimized model, the maximum number of features to consider was found to be the square root of the total number of features, minimum samples to split were found to be 38 and minimum number of samples per leaf node was found to be 3. This model achieved a cross-validation score of 48% and a test accuracy of 39%.

6 Discussion and Conclusions

The top five features from our analysis are : Page Views for previous week 3, Page Views for previous week 5,Page Views for previous week 4,Page Views for previous week 1, Month of November and Month of December.This shows that wikipedia page views are a good indicator of the movie revenue range and movies released in the month of November and December have similarity in their revenue buckets.Using the features discussed above and automatic discovery of buckets, we can predict the opening weekend movie revenue range of the upcoming movie with 39% accuracy. Using the method of fixed bucketing as stated in [17] we achieved the maximum cross validation accuracy of around 30%. Our approach of automatic clustering, yields cross-validation accuracy of 48% on the training data. Using the optimum number of clusters from the 'elbow criterion' we were able to obtain an even distribution of data with good classification accuracy[Figure:2]. Our initial proposal was to find out the first day revenue, but due to unavailability of relevant data we had to take into consideration the first weekend revenue for analysis. Since data is collected from three different sources, aggregated data for only 2155 movies was obtained. With such small data our ability to train complex features reduced significantly. Due to time constraint we could not explore some of the more convoluted features like competition during release,sequel ([17]) and revenue of the cast membersṕrevious movies.

References

[1] Beautiful soup: We called him tortoise because he taught us. https://www.crummy.com/software/BeautifulSoup/. (Accessed on 12/12/2016).

[2] Box office mojo. http://www.boxofficemojo.com/. (Accessed on 12/11/2016).

[3] Film and movie industry - statistics & facts | statista. https://www.statista.com/topics/964/film/. (Accessed on 12/11/2016).

[4] Omdb api - the open movie database. https://www.omdbapi.com/. (Accessed on 12/11/2016).

[5] U.s. advertising industry - statistics & facts. https://www.statista.com/topics/979/advertising-in-the-us/. (Accessed on 12/11/2016).

[6] cs229.stanford.edu/proj2015/203_report.pdf. http://cs229.stanford.edu/proj2015/203_report.pdf. (Accessed on 12/11/2016).

[7] stats.grok.se/en. http://stats.grok.se/en. (Accessed on 12/11/2016).

[8] Sitaram Asur and Bernardo A Huberman. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 492–499. IEEE, 2010.

[9] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[10] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[11] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[12] James Dougherty, Ron Kohavi, Mehran Sahami, et al. Supervised and unsupervised discretization of continuous features. In *Machine learning: proceedings of the twelfth international conference*, volume 12, pages 194–202, 1995.

[13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 6. Springer, 2013.

[14] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[15] Márton Mestyán, Taha Yasseri, and János Kertész. Early prediction of movie box office success based on wikipedia activity big data. *PloS one*, 8(8):e71226, 2013.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Ramesh Sharda and Dursun Delen. Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, 30(2):243–254, 2006.