Movie Revenue Prediction

Rajan Garg
140101057
rajan.garg@iitg.ernet.in

Chinangshuk Roy
140101017
chinangshuk@iitg.ernet.in

Raj Shekhar
140101056
raj.shekhar@iitg.ernet.in

Kuldeep Parewa
140101034
k.parewa@iitg.ernet.in

***Abstract*—Prediction of the movie revenue is an important aspect of Machine Learning which have not been tackled a lot and needs to be addressed as it includes billions of money of investors on risk. In our study we will analyze the data of the previous movies from last 100 years having their features like title, actors, directors, ratings along with their gross revenue collection and perform training and testing on the data set using our model. We will present the results shown by our model on pre-processed data and compare with previous works, and conclude our study.**

***Keywords*—Movie, Revenue, Neural network, SVM, Regression, Ratings.**

I. Introduction

Movie revenue relies upon various factors, for example, cast, budget, movie reviews, ratings, on-screen characters, director, genre, release year. As a result of these numerous elements , there is no scientific technique for predicting how much income will be produced by the film. However, by analyzing the data of revenues created by the past films, we can fabricate a model which can enable us to anticipate the expected revenue of a movie. Such an expectation can be exceptionally valuable for the Movie studio which will deliver the film, so they can settle on the costs like artist's pay, advertisements, promotions appropriately. Additionally investors can also invest, seeing the expected profit for their venture.

The film industry in U.S. is expected to collect a revenue of more than 32 billion USD in the year 2018 and 35 billion USD in the year 2019 and increasing year by year. This show how big is the film industry in U.S. and how important it is for the country. Film producers spend millions of dollars to produce a high budget movie, and risk their money on the filmmaking project. So it becomes really important for them to analyse the worth of taking that risk and know the expected target revenue which can be achieved by that film, which will help to reduce the risk or help in managing the budget and expenses better to gain the maximum profit. We plan to develop a prediction model based on the data of the previous movie that has been released and analyse the data to make a prediction of the expected revenue by the movie.

In section II, we will give various studies and models which have been developed by previous works in the field of prediction of movie revenues, by explaining their techniques and the results they had presented in their study. In section III, we will describe our dataset which we will be using to train our model and describe the features, their selection and normalization used for them. In section IV, we will present the various models we have tried in our research on the features we have elected and also present the results. In section V, we will discuss the results and compare them to the previous works and we will conclude our study.

II. LITERATURE SURVEY

There were some previous works performed by some famous researchers in the past. We have tried to study some of their works and these are the result findings listed below:

In [1], the paper provides a new approach for sentiment analysis in documents. The initial bag of words containing single words and bigrams were associated with a feature value. This value however was not just raw word count or the TFIDF value. Instead, it was the difference between the TFIDF score in the positive corpus and negative corpus. With a balanced training set, the prominence of a word in the positive training corpus brought out a negative value and vice versa hence creating a clear boundary between the two sentiments. The Delta TFIDF showed an improved accuracy of 88.1% on Pang and Lee's movie review dataset and established significantly better results than raw word frequencies and normal TFIDF weights.

In [2], the model was built around a set of features which were prior to release of the movie such as genre, length, release date, actors and directors, locality etc. obtained from the OMDB API. Four models were used to predict revenue: Linear regression on the logarithm(base 10) on the revenue, Naive Bayes classification on partitioned data (log base 10 buckets) with Laplace smoothing, Multinomial support vector machine model and Multinomial logistic regression. Naive Bayes performed better (51.6%) than the more complex SVM model (49.4%) and logistic regression (50.9%). The sparsity of data in buckets with the densest bucket containing more than 80% of the data points was a major drawback and adversely affected the results.

In [3], The Rotten Tomatoes API was queried with movie titles acquired from IMDB database. The feature set was represented in the form of a sparse vector of 0s and 1s (denoting absence or presence of the feature respectively). The first model used was linear regression on the raw data without using logarithms which produced very poor results (30% accuracy even with a predefined margin of 100% error). Next logistic regression was used with the help of logarithmic classification buckets and additional k-means clustering on movie titles increased accuracy to 52%. The problems with the experiment was that the feature set which included every unique actor and director was too large to handle and slowed down the process significantly.

In [4], The data worked on was obtained from the IMDB manually and was further pruned to satisfy certain conditions (2001-2010, English and gross revenue of at least $500,000). Conventional features such as genre, user rating, budget, run-time, MPAA ratings were included. Features had different range of values and they were required to be normalised. The model they applied was Linear regression using gradient descent. Linear buckets were used for logistic regression of variable size which gave the resultant high error of 57.1%. Their model was not successful in predicting the amount of revenue but rather predicted the profitability with an accuracy of 72.4%. The accuracy even in assigning a movie to a bucket was 25.3%. The major drawbacks were that the accuracy was poor, some features were unavailable pre-release (user review) and success of actors and directors were not taken into account which would have affected results significantly.

In [5], Data consisted of 834 movies in the time period of 1998-2002 from ShowBiz Data, Inc. The objective was to classify movies into 9 predefined classes starting from flop to blockbusters based on revenue generation. The independent features included MPAA rating, Competition, Star value, Genre, Technical effects, Sequel and Number of screens. A multi-layer perceptron neural network architecture was implemented using 2 hidden layers which employed sigmoid functions. For experimentation, a 10-fold cross-validation system was used and achieved the correct classification with an accuracy of 36.9% and classification with an error margin of 1 neighboring class with an accuracy of 75.2%. The absence of key features such as movie budget and also the linear categorization instead of logarithmic classes had a negative impact on the efficiency of the method employed.

III. DATASET

In this paper, we are using the famous IMDB Movie database which contains data of more than 5000 movies. The dataset is available on kaggle and has 5043 instances with 28 features. It contains data of all kind of movies across last 100 years with thousands of directors and actors information.

The variables used for the prediction are - Title/Name of Movie, Color represents if movie is Colored or Blank and white, Number of Critics for Reviews for the Movie, Movie’s Facebook Page Likes, Duration of the Movie, Director’s Name, Director’s Facebook Page Likes, Actor 3’s Name, Actor 3’s Facebook Page Likes, Actor 2’s Name, Actor 2’s Facebook Page Likes, Actor 1’s Name, Actor 1's Facebook Likes, Gross Collection of the Movie, Genre of Movie, Number of Voted Users, Cast’s total Facebook Likes, Number of Faces on Poster, Plot Keywords, Movie’s IMDB Link, Number of user for Reviews, Language of Movie, Country in which Movie Released, Content Rating, Budget of the Movie, Title Year, IMDB Score, Aspect Ratio.

Among these 28 features, we will not use many of these to train and predict our models. We assume that those features will have no significant role in the prediction as all the movies have different value for that variable. The feature we are using to train our models are -

Number of Critics for Reviews for the Movie, Movie’s Facebook Page Likes, Duration of the Movie, Director’s Facebook Page Likes, Actor 3’s Facebook Page Likes, Actor 2’s Facebook Page Likes, Actor 1's Facebook Likes, Genre of Movie, Number of Voted Users, Cast’s total Facebook Likes, Number of user for Reviews, Content Rating, Budget of the Movie, Title Year, IMDB Score, Language.

The genre of the movies also have different categories, which we will consider each of them as a feature representing 0 or 1. The movie has these genre - Action, Adventure, Fantasy, Sci-fi, Thriller, Comedy, Family, Horror, War, Animation, Western, Romance, Musical, Documentary, Drama, History,Biography, Mystery, Crime.So if the movie has genre War and History, then these 2 features will have value 1 and all other as 0.

The Content rating of the movies also have different categories, which we will consider each of them as a feature representing 0 or 1. The movie has these content ratings - General Audiences, Parental Guidance Suggested, Parents Strongly Cautioned, Restricted, Adults Only. So if the movie has rating Parental Guidance suggested, then Parental Guidance Suggested, Parents Strongly Cautioned features will have value 1 and all remaining as 0. As if the movie can watched with parental guidance, then it is available for Parents Strongly Cautioned category also which is a subset of the former category.

To represent the feature Language and Country, we will use 1 if Movie has Language English and 0 for others and 1 if Country is USA and 0 for others.

Gross is the output value that we will predict by our model. We will take the logarithm to the base 10 of the gross values and take the integer floor, to convert it into classes labelled from 0 to 8. As gross collection is less than 10 e-9.

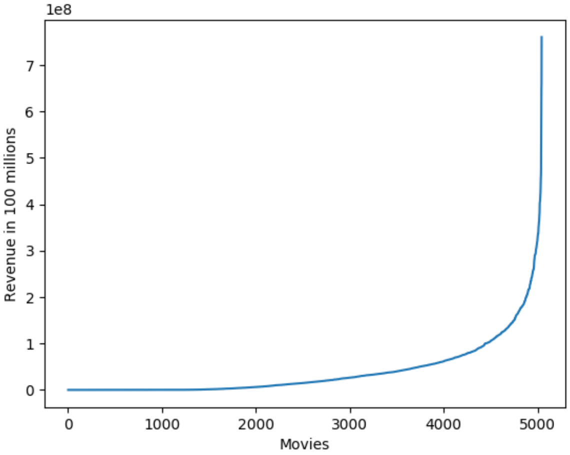
Then we normalize the data to convert all the values from range 0 to 1. We divided all the fields we are using to train the model with the maximum value of that field.

IV. EXPERIMENTS & RESULTS

We have done the pre-processing of the data which we have taken as a CSV input. In next step we have applied various ML models and predict the output. We will discuss the implementation of both stepwise below:

A. Pre-processing of dataset

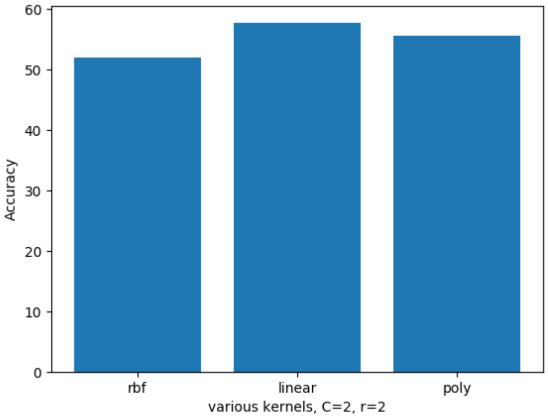
Some features/variables like Genre of the movie, Content rating and Language had values ranging in binary and could be subject to binary classification. However, the other features implemented by us such as Budget, Number of facebook likes, etc. had a larger range of values and are not binary classifications which could be highly variable. To tackle these shortcomings, we normalized the data such that the values had a mean of 0 and a variance of 1.



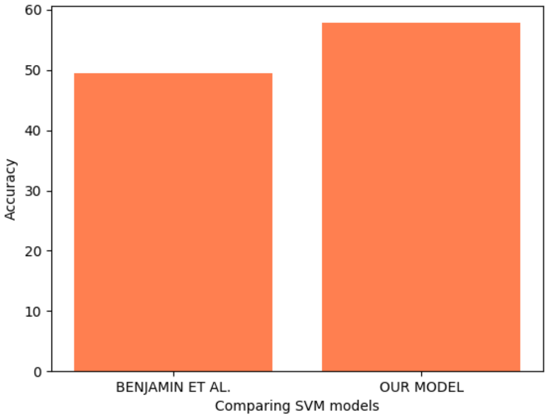
Also, the predicted revenue was classified into logarithmic buckets. We did this by converting the value of the revenue to its logarithm base 10 and rounding down to the nearest integer. Thus the range of values which our dependent variable (Movie revenue) could take was 0 to 8.

B. SVM

We implemented an SVM model using the python scikit library. Support Vector Machine is an effective machine learning model which is used for mainly classification and regression. The one reason we used SVM is because it is very effective in high dimensional spaces and as our number of parameters taken into account was significantly high, SVM was the clear choice.



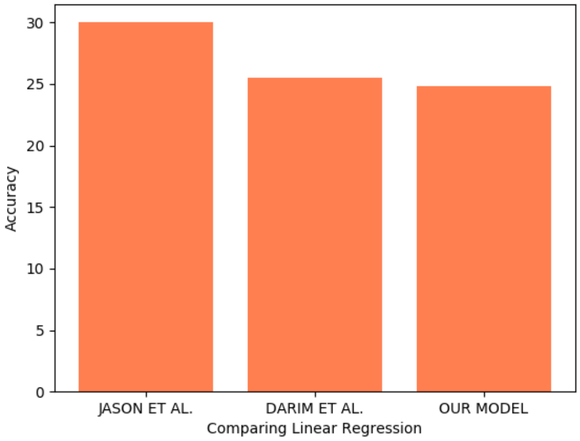
We implemented three separate SVM kernels (linear, polynomial and rbf) and compared the results. We found out that linear kernel gave the highest accuracy in our case.



We compared our result with one previous work implemented by Benjamin Flora and others and found that our model performed better than theirs. This was because the dataset used by them was insufficient to fill up the logarithmic buckets significantly. Our linear model gave an accuracy of 57.8% compared to their 49.4%.

C. Linear Regression

We implemented the linear regression model using python scikit library. It fit a linear model to minimize the sum of squares by observed response and predicted response by linear approximation.

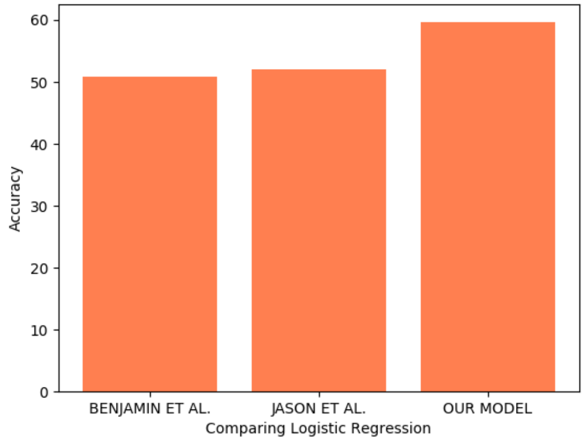


As the feature space was so large in our case, the linear model performed poorly with just 24.9% accuracy. Compared to other previous works, our minimalistic linear model did not perform as good and we decided not to experiment much with this model and move on to better models.

D. Logistic Regression

Logistic regression, contrary to its name, is a form of linear classification rather than regression. Our logistic regression model was multinomial which classified the revenues into the

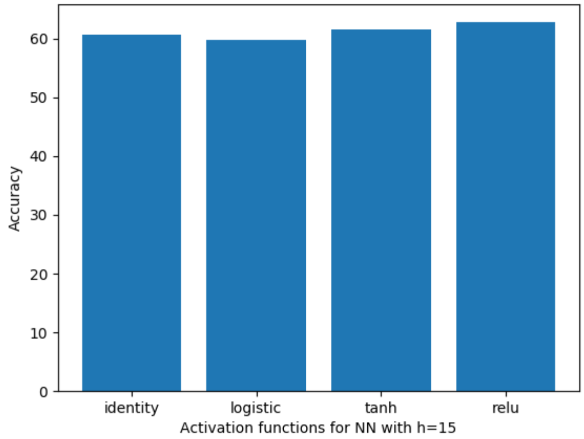
9 predefined classes. Adding or subtracting features under consideration did not affect the performance of the model.



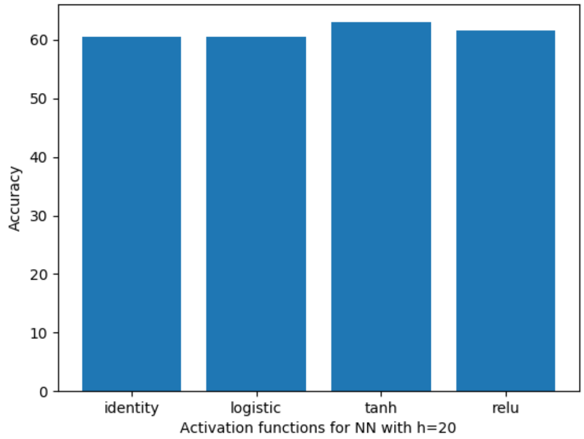
Logistic regression as expected performed significantly better than linear regression with an accuracy of 59.6%. Our model also outperformed the other previous works as our feature selection was optimal and dataset was abundant.

E. Neural Network

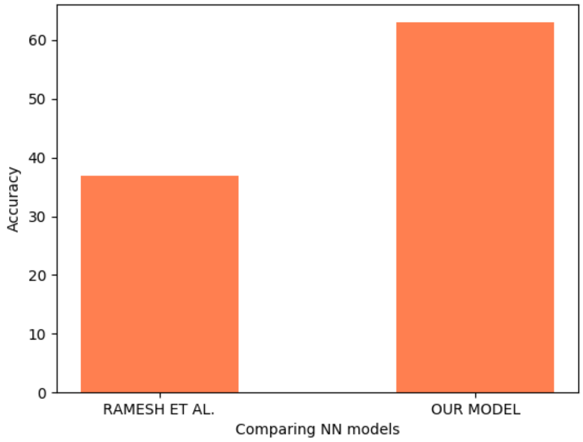
We implemented the neural network model using the python scikit library. We used the Multi-layer Perceptron (MLP) that learns a function by training on a dataset. We used it for performing classification on our target labeled from 0-8. We used MLP as it is capable of learning complex non linear models. An MLP is fitted with hidden layers followed by activation function.



Initially, we ran our MLP model with 15 hidden layers for 200 iteartions by varying the activation functions and found 'relu' performing better with an accuracy of 62.8%. While others were comparable with 'tanh' peformig at 61.6%.



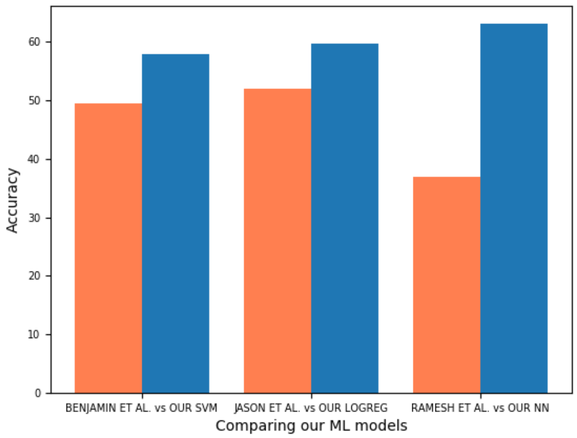
we ran our MLP model with 15 hidden layers for 200 iteartions by varying the activation functions and found 'tanh' performing better with an accuracy of 63%. While others were comparable with 'relu' peformig at 61.6%.



We compared our result with one previous work implemented by Ramesh Et Al. and found that our model performed much better with an accuracy of 63% compared to their 36.9%. This was because the dataset used by them was insufficient to fill up the logarithmic buckets significantly. As we had much more features to learn from and also our target buckets were logarithmic rather than linear buckets used by them.

V. CONCLUSION

The results show that our model performs much better compared the previous studies we discussed above. Our SVM model gave better results with accuracy of 57.8% compared to one in [2] with 49.4% as we have larger feature set and SVM is effective in high dimensional spaces. Our Logistic regression performed better with an accuracy of 59.6% compared to [2],[3] having accuracy around 50% our feature selection was optimal and dataset was abundant. We got best results using MLP model with an accuracy of 63% much better than in [5], because of using more features and using logarithmic buckets to classify.



In future, we can try to include the textual ratings in the features as the reviews given by critics and audience plays an important role in the success of the movie as they influence the larger audience to watch the movie. Also we could try to give variable weightage to the genre rather than binary, because sometimes the larger audience prefers a particular genre much more than others.

Acknowledgment

We would like to thank our professor Rashmi Dutta Barua, to give us this opportunity to do a useful research under her guidance. We would also like to thank the TA's for continous guidance and all our batchmates for clearing our doubts during the study.

References

[1] Martineau, J. and Finin, T., 2009. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. Icwsm, 9, p.106.

[2] Flora, Benjamin, Thomas Lampo, and Lili Yang. "Predicting Movie Revenue from Pre-Release Data." (2015).

[3] van der Merwe, J. and Eimon, B., 2013. Predicting Movie Box Office Gross. *Stanford University*.

[4] Im, D. and Nguyen, M.T., PREDICTING BOX-OFFICE SUCCESS OF MOVIES IN THE US MARKET.

[5] Sharda, R. and Delen, D., 2006. Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, *30*(2), pp.243-254.