Predicting Box Office Success: Do Critical Reviews Really Matter?

Lopamudra Pal
lpal@cs.umass.edu

Ly (Harriet) Bui
bui23L@mtholyoke.edu

Rishi Mody
rmody@cs.umass.edu

Abstract

A single movie can be the difference between millions of dollars of profits or losses for theaters, distributors and producers in a given year. The ability to predict a movie's box-office revenues can considerably reduce the financial risk. Hence, they are extremely interested in predicting revenues of movies. A lot of research work is carried out in movie revenue prediction by Economists, Statisticians, Machine Learning researchers, Film industry technical professionals. Most of the previous work in Machine Learning uses the metadata about movies like its genre, MPAA rating, and cast, with limited work on sentiment analysis from movie reviews. This stokes our interest to study the prediction of opening weekend movie revenues based on features extracted from sentiment analysis of movie reviews combined with the general metadata features. We shall consider a typical sentiment analysis task of calculating sentiment score for each review and use the results to train a Machine Learning model for revenue prediction task.

1 Introduction

The increased access to the Internet has allowed users to share opinions and sentiments about products online. Businesses rely on reviews to see how their products perform and make necessary adjustments. Consumers read online reviews to decide whether to buy the products or go to particular events. The film industry has also long joined this movement. Tons of reviews can be found online for a particular movie right after its first day of release. Majority of the movie going audience reads movie reviews online before deciding to go to the theater or purchase a movie. Therefore, movie review mining has drawn great attention.

Significant interest has been given to study economic values of reviews. Researchers examine relationships between sales performance of a particular product and their reviews. Public opinion has been proven to influence how well a product performs in the marketplace. It is natural to assume that online movie reviews have the power to increase or decrease the box office revenue. It was calculated that $32 billion dollars in revenue was generated by the film industry in the US in 2014 according to PwC (Statista, 2016a). How much of this revenue is impacted by online movie reviews? In this research, we focus on applying sentiment analysis on online movie reviews to study its impact on movie revenues. Prior study has shown that movie review sentiment does not play a significant role in movie revenue prediction. Applying machine learning techniques to generate sentiment scores and build predictor models, we come to a conclusion that is consistent with the previous study.

In summary, we make the following contributions:

- Create a dataset with movie reviews, revenues, and other metadata such as genre, rating, popularity, release date, budget, runtime, etc.

- Generate sentiment scores from Naive Bayes and Vader.

- Select feature for revenue prediction.

- Apply Random Forest Regressor, Ridge Regression, Elastic Net, Decision Tree Regressor, and Linear Regression for revenue prediction with and without sentiment score.

Following is the structure of the rest of the paper. Section 2 provides a literature review about related work that has been done on movie reviews, movie prediction, machine learning models we are experimenting with in this study. In section 3, we discuss how our dataset is obtained and gathered, advantages, and disadvantages of our dataset. We propose our methods to generate sentiment scores from Naive Bayes and Vader in section 4 as well as predictor models applied to predict revenue and evaluate impact of sentiment score on revenue. Section 5 is where we report our results generated from mentioned methods. In section 6, we facilitate conclusion for our study. We present an outline for our future work in section 7.

2 Related Work

2.1 Revenue prediction with sentiment classification

[1] introduces us to the main motivation behind performing a sentiment analysis task on movie reviews to predict movie revenues. The prediction tasks this paper has worked on are predicting the total revenue generated by a movie during its release weekend and the per screen revenue generated during the release weekend. The model used is linear regression. The features used are extracted from the movie metadata and the text of the reviews. The features extracted from text are specifically n-grams, POS n-grams and dependency relations between words. The paper shows promising results when using both metadata features and text features together.

[5] describes an aspect oriented scheme that analyses the textual reviews of a movie and assigns it a sentiment label on each aspect. The scores on each aspect from multiple reviews are then aggregated and a net sentiment profile of the movie is generated on all parameters. The paper shows an implementation of the SentiWordNet based algorithmic formulation for both document-level and aspect-level sentiment classification. This gives us an insight into the various kinds of sentiment analysis problems related to movie reviews.

2.2 Machine Learning Models

[2] proposes a multimodal deep neural network for movie box-office revenues prediction. Though this paper uses movie posters as dataset, it is of interest to us because of the use of Convolutional neural networks model (CNN) as part of training the prediction model. A CNN is first constructed as a feature extractor for movie posters, then the movie poster content is combined with other selected movie-related data as input, which is expected to improve the performance of movie box-office prediction.

[3] gives us an idea of applying different machine learning models on movie dataset involving non-text features like Genre, MPAA, Ratings, Movie Length.

It provides us with a reasonable simple template to compare our results by including Sentiment Analysis of movie reviews as added features.
[4] presents us with an elaborated study in the field of revenue prediction and hence is a good material for us to refer for our study.

2.3 Sentiment Analysis Methods

[6] captures semantic and sentiment similarities among words. This paper is of relevance to our study because it uses movie reviews dataset for the experiments. To capture semantic similarities among words, they derive a probabilistic model of documents which learns word representations. This component does not require labeled data, and shares its foundation with probabilistic topic models such as LDA. Using Logistic Regression as a predictor function, the paper maps a word vector to a predicted sentiment label. This in turn helps in improving the word vector to better predict the sentiment labels of contexts in which that word occurs.
Paper [8] The paper discusses in details two supervised machine learning algorithms: K-Nearest Neighbour(K-NN) and Naive Bayes. Their overall accuracy, precision and recall values are evaluated and compared. For movie review, Naive Bayes performs much better than K-NN with more than 80% accuracy. However, for hotel review both classifiers give out similar lower accuracies. The papers conclusion confirms Naive Bayes as a good baseline model for our sentiment analysis of movie review.
Paper [9] compares Vaders effectiveness to eleven typical models including LIWC, ANEW, the General Inquirer, SentiWordNet, Naive Bayes, Maximum Entropy, and Support Vector Machine algorithms. Applying parsimonious rule-based model to evaluate sentiment of tweets, the paper concludes that Vader outperforms individual human raters and is able to generalize across contexts better than other models. For our paper, we have used Vader to generate sentiment score form our movies along with Naive Bayes. We will see how these scores generated from two different benchmarks affect the revenue prediction results.
Paper [10] provides insight into an interesting sub-discipline known as ”opinion mining” which is considered to be at the cross roads of information retrieval and computational linguistics. Recent research has been more focused on computing the PN-polarity of subjective terms. Thus in a sentence it must identify whether a term or multiple terms that can be used to provide an opinion has a positive or a negative sentiment attached to it. On the other hand, research on determin-

ing the SO-polarity of terms, i.e. whether a term indeed indicates the presence of an opinion (a subjective term) or not (an objective, or neutral term) has been instead much scarcer. This paper describes SentiWordNet which is a lexical resource which gives each term a triplet of scores which denote positive, negative and objective. The value of these scores describe how strongly the term enjoys each of the three properties.

3 Dataset

The problem we are trying to analyze and tackle can be divided into two parts. The first part is based on analyzing movie reviews. We use multiple natural language processing models to analyze these reviews. We not only divide them into two categories and label them as positive or negative as per the overall sentiment that they bear but we also aim to quantify the positiveness/ negativeness of the review.

So for this first part of our project we had shortlisted multiple datasets which could have been of use to us. They include:
1) Movie Data Corpus: It contains movie metadata, financial information and movie critic reviews. http://www.cs.cmu.edu/ ark/movie$-data/
2) Large Movie Review Dataset: This is a dataset of movie reviews for binary sentiment classification. http://ai.stanford.edu/ amaas/data/sentiment/
3) Web Data: Amazon Movie Reviews: This data contains movie review from amazon, which have been collected over a duration of almost 15 years till Oct 2012. https://snap.stanford.edu/data/web-Movies.html

To conduct sentiment analysis experiments we have used the reviews from the ”Large Movie Review Dataset”. There were multiple reasons why we chose to use this dataset. Firstly, the number of reviews for testing and training are 25,000 each. They are equally divided into positive and negative reviews. Thus giving us a very varied kind of data of movies released in the 1950s to movies released in 2010. Secondly, the other options that were available to us did not really fit the bill here. The ”Movie Data Corpus” is small as it contains only 10,521 reviews and it contains no labels with which we could compare our predictions. Similarly, the ”Amazon Movie Reviews” data though very big, had we used it we would not have been able to cross-check our sentiment analysis. Thus considering the time frame we had to successfully complete this project we decided to choose the 2nd dataset.

A lot of the movies in the dataset had multiple reviews. Some movies even

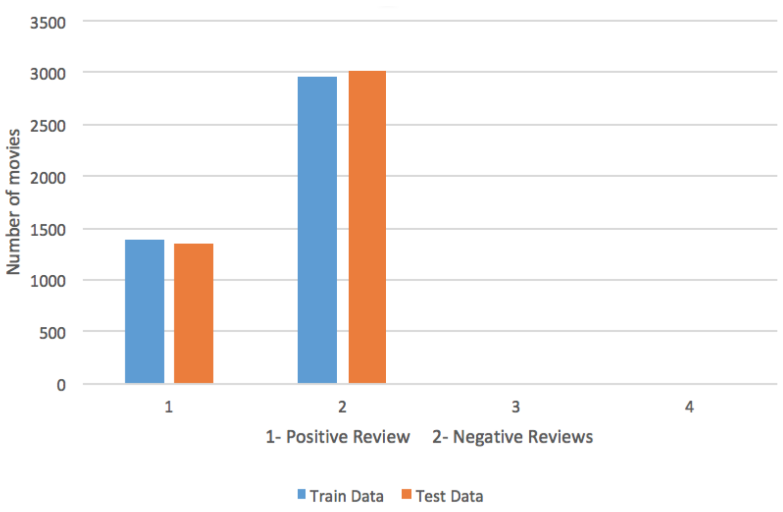


Figure 1: Distribution of movies with the review sentiments

had positive as well as negative reviews. So we have taken the average of these scores and assigned the mean sentiment score to the movie.

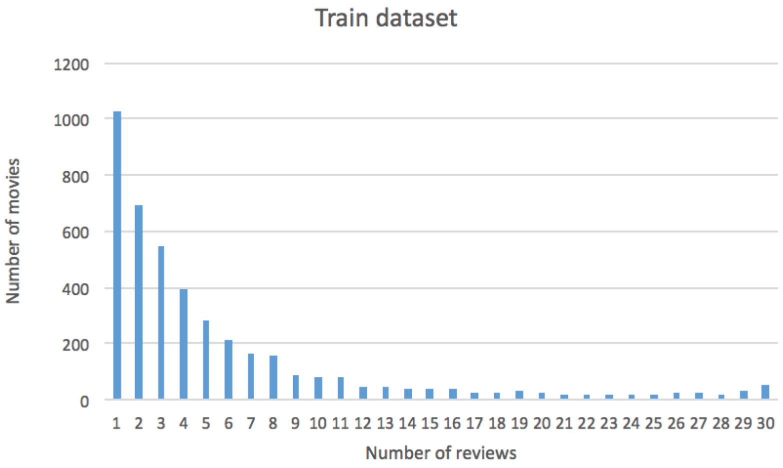


Figure 2: Review Distribution in Train Dataset

After running sentiment analysis on the dataset, we would get, as output the sentiment tags of each of the reviews using which we have calculated the overall sentiment score. These scores would be used as a feature for the data set we have created for the predicting the movie revenue using machine learning models.

The second part of our project is based on predicting movie revenues. We use multiple machine learning models to predict the revenues of movies and we

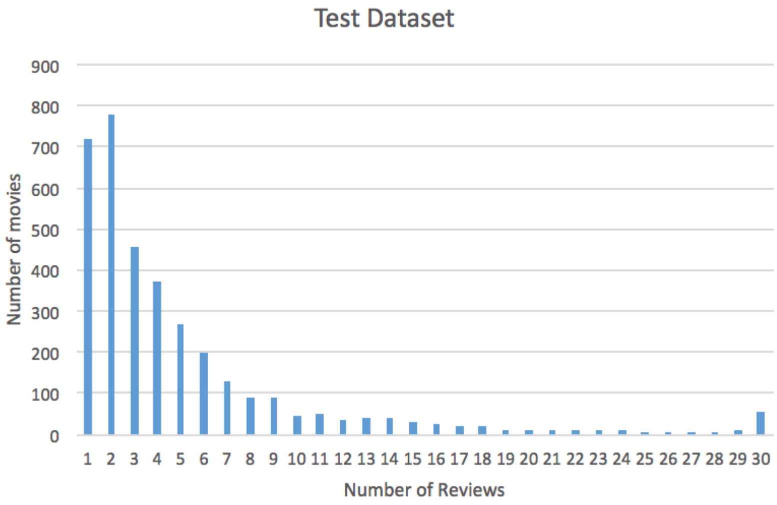


Figure 3: Review Distribution in Test Dataset

analyze the impact movie reviews have on the box office collections.

We checked various datasets that were available online as open source. We discussed if the features provided by these datasets were the ones we wanted to use or if we wanted to be a more innovative and decided to collect our own data. This dataset was created using multiple features which were extracted from internet websites such as rotten tomatoes, IMDB and the movie database(TMDB).

The movies that we needed data was indirectly shortlisted for us due to the dataset we used for the reviews. So we aimed at just extracting extra data for these movies. To obtain this data we accessed TMDB through the API that they provide on their website, a lot of their data has been taken from IMDB. Whatever extra data we needed, we scraped from the internet. We used Python as the sole scripting language. Further we had multiple mini datasets containing different features which we had to combine together. For eg. the movie review sentiment scores were mapped to the file number in the database. Thus we had to develop codes to match the exact sentiment score to the movie name as well as with the rest of the movie data.

We currently have features such as genres, popularity scores, votes, average vote, run time, budget, original language, release time and the revenue.

We had to clean this database a bit as well. Some of the movies could be classified into multiple genres, but what we noticed was that the first genre in this list was the more prominent one hence we used only that genre as a feature.

There were quite a few entries where at least one of the feature values were

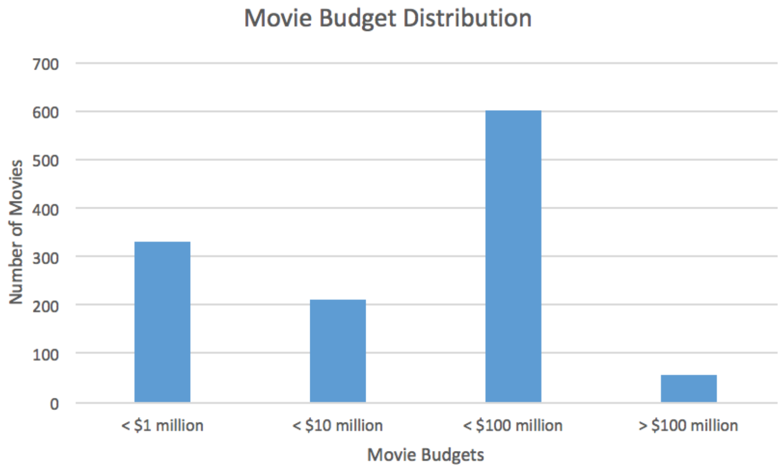


Figure 4: Movie Budget Distribution

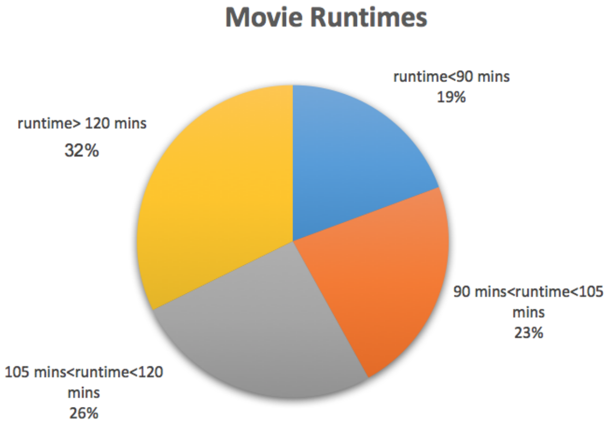


Figure 5: Movie Runtime Distribution

missing. To combat this we eliminated those movies from the database. We also dropped movies that had revenue less than $100.

As can be seen from the movie revenue distribution graph, the revenues in our database are highly varied and well spread out. We have movies with revenues as less as $12,207 and as high as $1 billion thus we decided to normalize the revenues. After dividing all of them by a factor of 1000000 we calculated the mean of all movie revenues. We then took the absolute value of the revenue subtracted from the mean.

$$revenue_{normalized} = |revenue - revenue_{mean}| \tag{1}$$

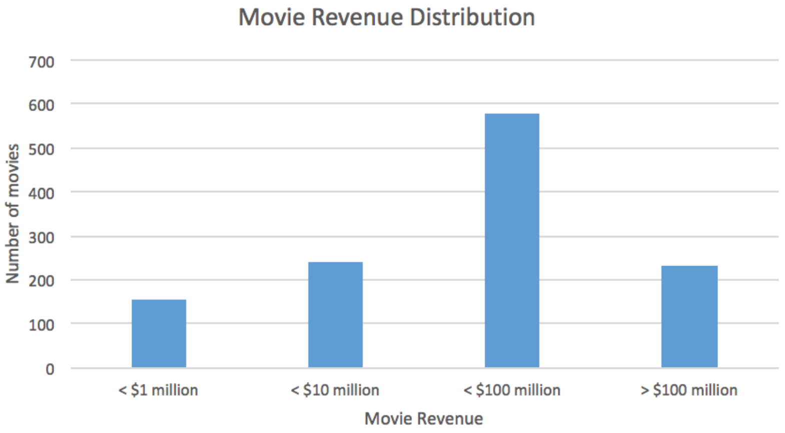


Figure 6: Movie Revenue Distribution

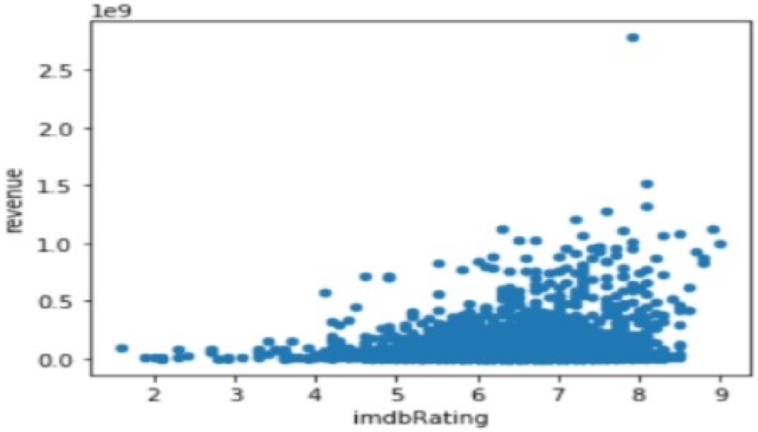


Figure 7: Revenue Distribution wrt IMDB Ratings Dataset

In the end, we have a total of 1202 movies with sentiment score, revenue data and other metadata. The database is stored in a .csv file format. We had planned to further add features such as if the movie was a summer release, released around public holidays, sequel or part of a franchise as well but since this required a lot of human intervention and annotation, our timeline did not allow us to do so.

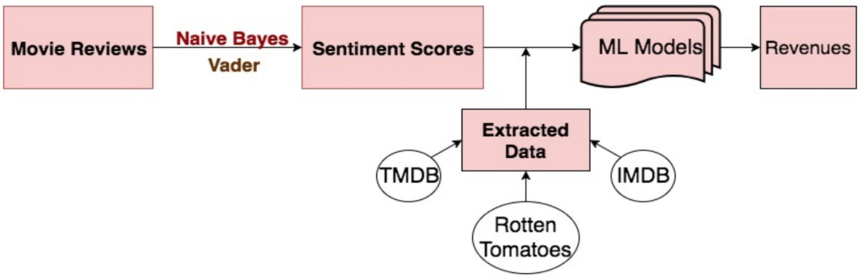


Figure 8: Process Flow

4 Methodology

4.1 Sentiment Classification Models

4.1.1 Naive Bayes

After training the Linear Regression model on the dataset after excluding sentiment scores, we incorporate our own sentiment score extracted from the movie review as a feature in the model besides those that have already been chosen. We examine if adding sentiment score from movie reviews will improve the revenue prediction. Our first baseline model used to extract sentiment score is Naive Bayes. For each movie review, we compute the probability for the movie review to be negative or positive with log normalizer. Then we experiment with two different method to computer the sentiment score. The first method is Subtraction and the second method is Comparison and Assignment.

To compute the probability of a movie review belonging to positive or negative class. First, we compute unnormalized log posterior for both labels (positive and negative). The unnormalized log posterior is the sum of log prior and log likelihood. Then we divide the unnormalized log posterior by the log normalizer.

$$P(y_{positive}|W_d) = \frac{P(y_{positive}|W_d)P(W_d|y_d)}{P(W_d)} \tag{2}$$

$$P(y_{negative}|W_d) = \frac{P(y_{negative}|W_d)P(W_d|y_d)}{P(W_d)} \tag{3}$$

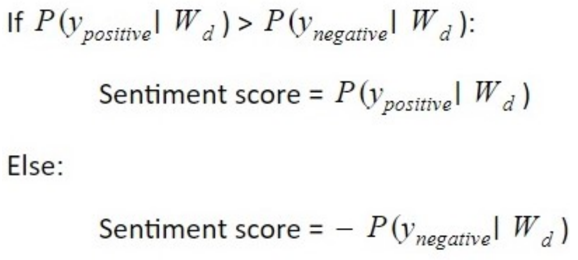
1. Subtraction Method:
   We take the difference between the two probabilities (positive - negative) to obtain the overall sentiment score for a movie review. If the score is greater

than 0, the review is positive. Otherwise, it is negative. If the score is 0, the review is neutral.

Sentiment Score =

$$P(y_{positive}|W_d) - P(y_{negative}|W_d) \tag{4}$$

2. Comparison and Assignment method:
   Instead of taking the difference between positive and negative probability, we compare two probabilities.



If $P(y_{positive}|\ W_d) > P(y_{negative}|\ W_d)$:

Sentiment score = $P(y_{positive}|\ W_d)$

Else:

Sentiment score = $-\ P(y_{negative}|\ W_d)$

Figure 9: Naive Bayes score calculation using Comparison method

After running both algorithms for 4 sets of dataset: positive training, negative training, positive testing, and negative testing dataset, we calculate the accuracy rate for each algorithm. For each movie review, if the algorithm generates the right score that is greater than 0 for positive review and less than 0 for negative review, the algorithm will get a correct point. Therefore, the

$$accuracy\ rate = \frac{correct\ points}{total\ reviews} * 100 \tag{5}$$

Subtraction method has much higher accuracy rates compared to the Comparison & Assignment one. Therefore, we keep the scores generated with the subtraction method for our final dataset.

For each movie, we have several movie reviews. Thus we calculate the average of the sentiment scores from the reviews for the final sentiment score. The higher the sentiment score is the more positive the movie review becomes.
As the sentiment score generated with the Subtraction method is quite small, we normalize the result by multiplying the score with 100,000.

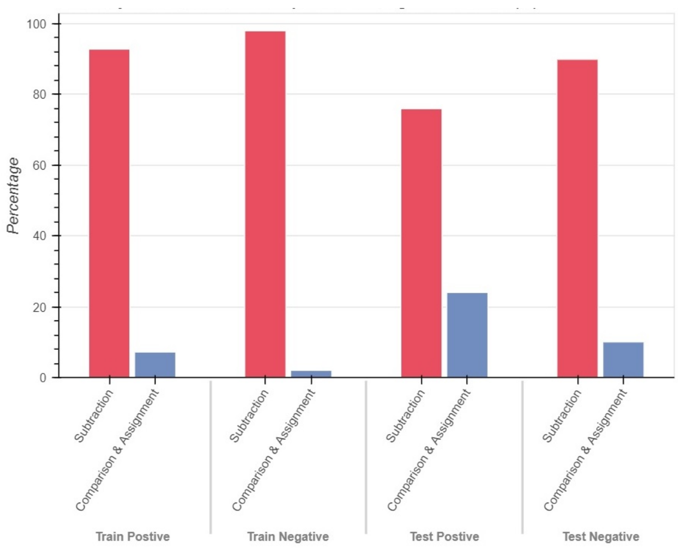


Figure 10: Naive Bayes Accuracy (Subtraction vs Comparison and Assignment)

Dataset	Subtraction Acc	Comparison Acc
PosTest	75.96	24.04
NegTest	89.92	0.08
PosTrain	92.792	7.208
NegTrain	97.96	2.04

Table 1: Comparing Accuracy Rates(in percentage) generated from 'Subtraction' method and 'Comparison and Assignment' method

This method of computing sentiment score allows us to work with continuous variable and improves the linear regression model to find correlation between movies sentiment score and its revenue.

4.1.2 Valence Aware Dictionary for Sentiment Reasoning (VADER)

VADER is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. VADER text sentiment analysis uses a human-centric approach, combining qualitative analysis and empirical validation by using human raters and the wisdom of the crowd.
Text sentiment analysis is a big field and can be narrowed down to two basic approaches. *Machine learning* approaches, on the other hand, look at previously labeled data in order to determine the sentiment of never-before-seen sentences.

The machine learning approach involves training a model using previously seen text to predict/classify the sentiment of some new input text. The nice thing about machine learning approaches is that, with a greater volume of data, we generally get better prediction or classification results. However, unlike lexical approaches, we need previously labeled data in order to actually use machine learning models. $Lexical$ approaches aim to map words to sentiment by building a lexicon or a dictionary of sentiment. We can use this dictionary to assess the sentiment of phrases and sentences, without the need of looking at anything else. Sentiment can be categorical such as negative, neutral, positive or it can be numerical like a range of intensities or scores. Lexical approaches look at the sentiment category or score of each word in the sentence and decide what the sentiment category or score of the whole sentence is. The power of lexical approaches lies in the fact that we do not need to train a model using labeled data, since we have everything we need to assess the sentiment of sentences in the dictionary of emotions. VADER is an example of a lexical method. In this method, lexical features other than text like ':-)', acronyms 'LOL', and slang 'meh' also get mapped to intensity values. Emotion intensity or sentiment score is measured on a scale from -4 to +4, where -4 is the most negative and +4 is the most positive. The midpoint 0 represents a neutral sentiment. Sample entries in the dictionary are 'horrible' and 'okay,' which get mapped to -2.5 and 0.9, respectively. In addition, the emoticons '/-:' and '0:-3' get mapped to -1.3 and 1.5.

VADER sentiment analysis (well, in the Python implementation anyway) returns a sentiment score in the range -1 to 1, from most negative to most positive. The sentiment score of a sentence is calculated by summing up the sentiment scores of each VADER-dictionary-listed word in the sentence and then normalizing according to the normalization used by *Hutto*

$$\frac{x}{\sqrt{x^2 + \alpha}} \tag{6}$$

where x is the sum of the sentiment scores of the constituent words of the sentence and α is a normalization parameter that we set to 15. The normalization is graphed in Figure 11

We see here that as x grows larger, it gets more and more close to -1 or 1. To similar effect, if there are a lot of words in the document youre applying VADER sentiment analysis to, you get a score close to -1 or 1. Thus, VADER sentiment analysis works best on short documents, like tweets and sentences, not on large documents.

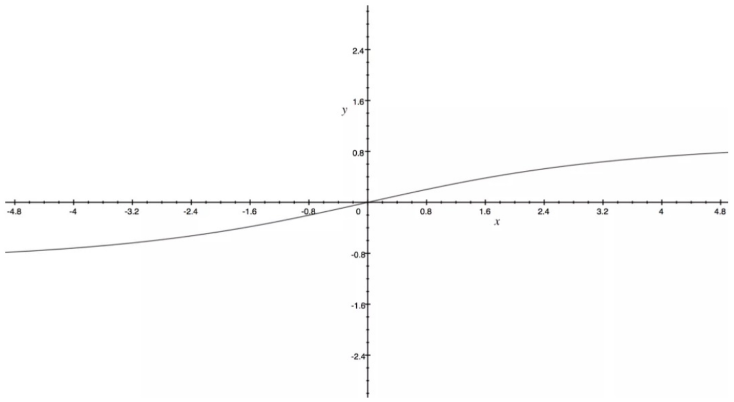


Figure 11: Normalization of sentiment scores in VADER

4.2 Machine Learning Models

We trained multiple machine learning models to be used for predicting the revenue, some of the models which we tried are detailed below. The results of these models are described in the Results section.

Linear Regression : Linear Regression is a parametric regression method that assumes the relationship between y and x is a linear function with parameters w = $[w_1, ...w_D]^T$ and b.The regression function is given as

$$f_{Lin}(X) = \Sigma_{d=1}^{D} w_d x_d + b \tag{7}$$

Decision Tree Regressor : Decision Tree Regressor is a parametric regressor that outputs data cases using a conjunction of rules organized into a binary tree structure. Each node in the tree consists of a rule of the form (xd < t) or (xd = t)t. The simplicity of this model is the primary reason we picked this model.

Ridge Regression: In regression methods such as least squares linear regression, the parameters are susceptible to very high variance.To control variance, we might need to regularize the coefficients. Ridge Regression is a form of regularized least squares when the weights are penalized using the l2 norm,

$||w||_2^2 = w^T w = \Sigma_{d=1}^{D} w_d^2$.

The regularization of weight parameters during learning is achieved by setting w^* as

$$w^* = argmin_w \frac{1}{N} \Sigma_{i=1}^{N} (y_i - x_i w)^2 + \lambda ||w||_2^2 \tag{8}$$

$$= argmin_w \frac{1}{N} \Sigma_{i=1}^{N} (y_i - x_i w)^2..st||w||_2^2 <= c \tag{9}$$

The optimized regularized weights which is the ridge regression estimator is $w^* = (X^T X + \lambda I)^{-1} X^T Y$. The regularization of Linear Regression may help Ridge to model the non-linearity of the revenues. Hence, we chose this model.

Random Forest Regressor : Random forests regressors are ensemble learning methods for regression that operate by constructing a multitude of decision trees at training time and outputting the mean prediction of the individual trees. Random decision forests correct for decision trees habit of over-fitting to their training set. This ensemble method may give better results than decision trees.

Elastic Nets : The elastic net method overcomes the limitations of the Ridge and Lasso method which uses a penalty function based on

$$\beta_1 = \Sigma_{j=1}^{p} |\beta_j| . \beta_1 = \Sigma_{j=1}^{p} |\beta_j| \tag{10}$$

The estimates from the elastic net method are defined by

$$\hat{\beta} = argmin_\beta(||y - X\beta||^2 + \lambda_2 ||\beta||^2 + \lambda_1 ||\beta||_1) . \hat{\beta} \tag{11}$$

$$= argmin_\beta(||y - X\beta||^2 + \lambda_2 ||\beta||^2 + \lambda_1 ||\beta||_1) \tag{12}$$

In this, both l1 and l2 norm are regularized. This variation may help Elastic Nets model the revenues accurately. Hence, we chose this model.

4.3 Modeling the Regression Algorithms as Classification Problems

The various regression models described above output a label which denotes the predicted revenues of the given movies. One of the most common ways of evaluating how well a machine learning model fits is to use **Root-mean-squared-error** as an evaluation metric. The RMSE represents the sample standard deviation of the differences between predicted values and expected values. When the calculations are performed over the entire data that was used for the prediction process, the individual differences are called residuals while if we consider some of these results individually out-of-sample it is called prediction errors. The RMSE returns a cumulative measure of the predictive capabilities of the model by summing over all the predictions. RMSE is the square root of the average of squared errors. The effect of each error on RMSE is proportional to the size of the squared error;

thus larger errors have a disproportionately large effect on RMSE. Consequently, RMSE is sensitive to outliers.

$$RMSE = \sqrt{\frac{\Sigma_{n=1}^{N}(\hat{y}_i - y_i)^2}{n}} \tag{13}$$

We obtained an RMSE score for our dataset using each of our models and were able to see the impact of sentiment analysis on our results as well. But we believe that RMSE, though it is an amazing metric to evaluate regression problems, it is not completely fitting for predicting movie revenues. Considering most movies have revenues of the order of tens of millions of dollars, not being able to get the prediction to the nearest penny is not necessary. For instance Alice in Wonderland had a revenue of almost \$ 1 billion. If our model was able to predict a revenue of around \$ 925 million it should still be considered as an "accurate" prediction as there is a plethora of factors that affect the revenue of the movie. Outliers such as a blackout on opening weekend in a moderately size city or more general events such as a discount of 0.5 cents in all screens in a large city would easily make the movie "lose" \$50 million in revenues. Thus we have tried to convert our regression problem to a classification one. With the above analogy we have allowed our model to have a leeway of 15% in the predictions values and if the prediction lies in this range we have counted it as an "accurate" prediction and to get the accuracy of the problem we have divided the count of accurate results with the total results.

$$accuracy = \frac{count\ of\ accurate\ results}{total\ no.\ of\ results} * 100 \tag{14}$$

5 Experiments and Results

"*I actually found this movie 'interesting'; finally one worth my time to watch and rent. It is true... some scenes were over the top on emotionalism, shouting, etc., but what movie doesn't stress its agenda, genre or 'ax to grind'? Almost None! What surprised me is that I read a review elsewhere done by a S.Fran reviewer on another review site, but found his negative review instead a more accurate description of his "own" review of the movie; not of the movie at all. Anyone that watches this movie will realize that it is great to recommend to family and friends; no car chases, Yea!! Being "in" an Italian family myself, I can fully relate to the environment portrayed on the screen. The movie has its tear jerking parts as well.*

It is what real life can be in such an environment. Nice movie. ”

VADER has calculated a sentiment score of **0.9597** for the above movie review. Clearly, the review is a positive because of words like ***Nice***, ***true***, ***interesting*** one and VADER has been able to calculate the sentiment score accurately. Following is an example of a negative review and the score associated with it. Naive Bayes also generates a positive score of **1.1737** for this review

”*No wonder this was released straight to DVD here in Australia, no redeeming features what so ever. The dialog was hokey, the acting, awful and the script sucked!! Whoever thought it would be a good idea to do a sequel or follow up to the far superior John Badham film, Wargames from the 80s, well they must of been on something cause it was a bad idea!! Amanda Walsh was good in it as the eye candy/love interest, while Matt Lanter was good as the other main lead- that is about it. I would not recommend Wargames: The Dead Code to anyone, check out Hackers or the original Wargames film- both are better than this piece of crap!!*”

For this movie review the sentiment score generated by VADER is **-0.7382** which is again very clear from the review that it is a negative one. The negative words in it like ***crap***, ***awful*** and ***bad*** are very strong enough to describe the entire review as negative. Naive Bayes generates a negative score of **-3.9131** for this review.

We modeled our problem as both regression and classification. Figure 12 shows a comparison of the error observed when predicting the revenues using the sentiment scores generated from NaiveBayes and VADER and feeding our data into various Regression algorithms. We observe almost similar results when comparing different models. Using sentiment scores generated by Naive Bayes or VADER did not affect the results much. This is probably because the values of the sentiment scores are quite less as compared to the revenue values. The best results obtained were with Decision Tree Regression.

Figure 13 shows that when RandomForestRegressor results were modeled as classification problem, they gave the best accuracy. This is most probably because we have different types of features. We have also used feature selection technique called ’SelectKBest’ which helps select the k most relevant features from the set of all features. Using feature selection technique has improved the performance

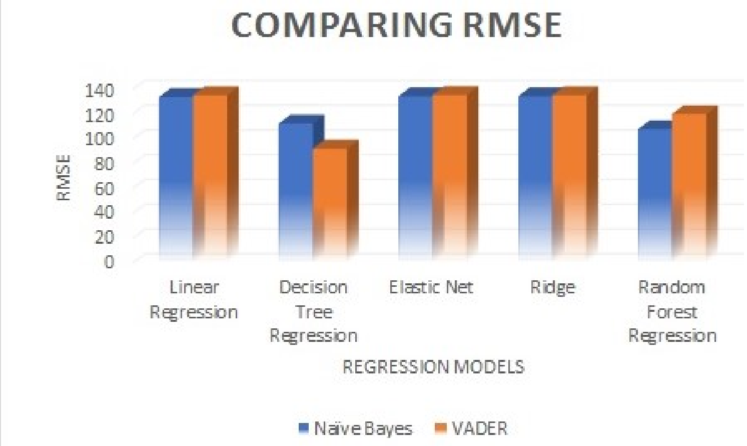


Figure 12: Comparison of Root Mean Squared error between the actual revenues and the predicted revenue using Naive Bayes Sentiment scores and VADER sentiment scores

of the model by around 10%.

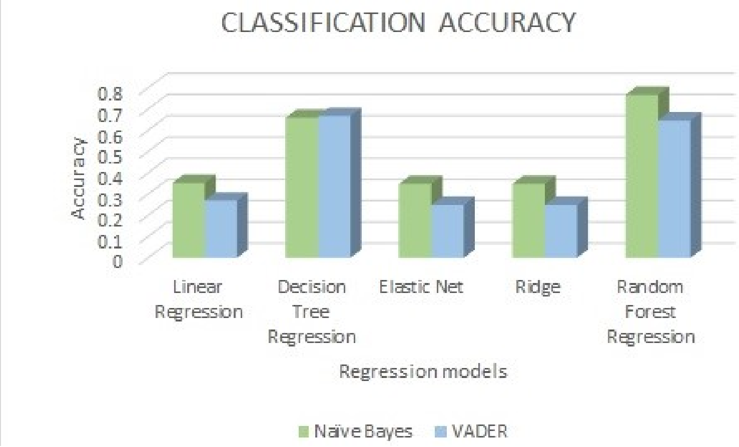
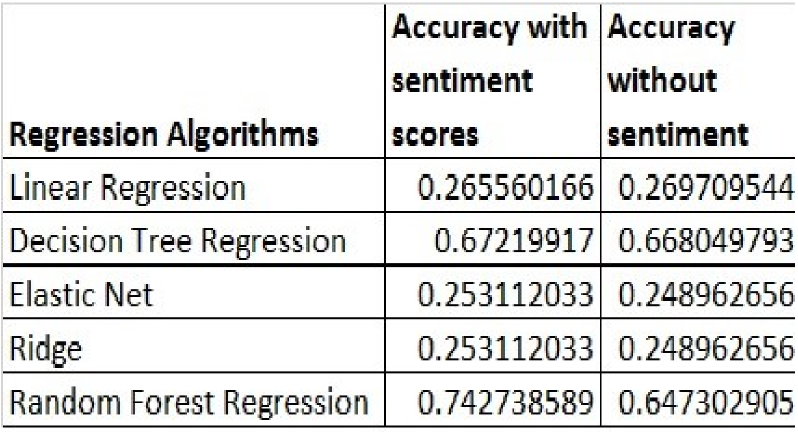


Figure 13: Comparison of classification accuracy using Naive Bayes Sentiment scores and VADER sentiment scores

It can be observed from Figure 14 that the sentiment scores do not differ the prediction accuracy of the models much.



Regression Algorithms	Accuracy with sentiment scores	Accuracy without sentiment
Linear Regression	0.265560166	0.269709544
Decision Tree Regression	0.67219917	0.668049793
Elastic Net	0.253112033	0.248962656
Ridge	0.253112033	0.248962656
Random Forest Regression	0.742738589	0.647302905

Figure 14: Comparing the Revenue Prediction Accuracy of models using sentiment scores and not using sentiment scores

6 Conclusion

From our experiments we conclude that our correlation between the sentiment scores and movie revenues is not very strong. When incorporating the sentiment score into the model, the RMSE error rates decreases from 111.33 to 91.92. However, the difference is not significant. We train the predictor models with two different set of sentiment scores generated from Naive Bayes and Vader and received similar results. This result confirms O'Driscoll (2016)[13] statement that research relying on information available only after the movie is released such as reviews and award nomination for prediction models has weakness regarding feature selection (O'Driscoll, 2016). Only 25% of total movie revenue is generated during the opening week (Simonoff and Sparrow, 2000)[4]. A large portion of the movie revenue has been generated before the movie is released based on factors like marketing strategies, who the directors and casts are, and genre. Thus, factors collected after a movie is released such as movie review is not significant in the prediction model.

However, we notice that even though the revenue can not be strongly correlated to movie reviews, movies with higher revenues, in general have more positive reviews.

7 Future Work

The evaluation method for sentiment classification and predictor model used in this paper is accuracy. Accuracy method asks what percentage of all the observations the classifier labels correctly. Even though accuracy is a natural metric to use for text classification, it does not work well when the classes are unbalanced. Precision, recall, and F-measure can be examined in future work to further evaluate our classification models.

Research by Yu et al.[14] selects sales performance and review quality as an additional features along with public sentiments for revenue prediction. Our model does not include review quality and past sale performance as features. The paper centers around the proposal of Sentiment Probabilistic Latent Semantic Analysis (S-PLSA) as an effective generative model for sentiment analysis. The model considers a review as a document composed of hidden sentiment factors in order to account for the complexity in nature of sentiments. It only focuses on the words that capture sentiments. The paper also shows that these features strongly correlates to revenue prediction. In the future, we would like to explore with the Sentiment PLSA and take into consideration past sale performance and review quality as features.

Our movie dataset with sentiment score and revenue is limited in size at the moment. We would love to expand our dataset of revenue in the future to see if our confidence in the results still hold.

References

[1] Mahesh Joshi, Dipanjan Das, Kevin Gimpel and Noah A. Smith. *Movie Reviews and Revenues: An Experiment in Text Regression. (2010)* Los Angeles, California ISBN:1-932432-65-5

[2] Yao Zhou, Lei Zhang, Zhang Yi. *Predicting movie box-office revenues using deep neural networks (2017)* DOI: 10.1007/s00521-017-3162-x

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

[4] J. S. Simonoff and I. R. Sparrow.*Predicting movie grosses: Winners and losers, blockbusters and sleepers. (2000)*,

[5] V. K. Singh, R. Piryani, A. Uddin, P. Waila *Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification (2013)* DOI: 10.1109/iMac4s.2013.6526500

[6] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts,*Learning word vectors for sentiment analysis, (2011)* ISBN: 978-1-932432-87-9

[7] Ccero Nogueira dos Santos, Mara Gatti *Deep Learning approach for sentiment analysis of short texts (2017)* DOI: 10.1109/ICCAR.2017.7942788

[8] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, Sweta Tiwari, *"Sentiment Analysis of Review Datasets using Naive Bayes and K-NN Classifier (2016)* arXiv:1610.09982

[9] C.J.Hutto, Eric Gilbert *Vader: A parsimonious Rule-based Model for Sentiment Analysis of Social Media Text (2015)*

[10] Andrea Esuli and Fabrizio Sebastiani *SentiWordNet: A High-Coverage Lexical Resource for Opinion Mining (2007)*

[11] Ohana, B. Tierney, B. *Sentiment classification of reviews using SentiWordNet. 9th. ITT Conference, Dublin Institute of Technology, Dublin, Ireland, 22-23 October.* doi:10.21427/D77S56

[12] Daniel Jurafsky James H. Martin. *Speech and Language Processing (2016)*

[13] Sean ODriscoll. *Early Prediction of a Film Box Office Success Using Natural Language Processing Technique and Machine Learning (2016)*

[14] Xiaohui Yu, Yang Liu, Jummy Xiangji Huang, Aijun An *Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain (2012)*

[15] Statista *Global filmed entertainment revenue 2015-2020. (2016a)* https://www.statista.com/statistics/259985/global-filmed-entertainment-revenue/