# IMDb Movie Review Prediction using Text, and Emoji Ideogram Data on Twitter

Rohit Bapat
rbapat@iu.edu
Indiana University, Bloomington

Vishal Singh
singhvis@iu.edu
Indiana University, Bloomington

Tanvi Patil
tpatil@iu.edu
Indiana University, Bloomington

## ABSTRACT

**Twitter is an online microblogging platform where the users can voice their opinions using Tweets. Twitter data acts as a vast and real-time source of information. It contains a huge amount of sentiment data which can be used for analysis and prediction. Movie rating prediction is one such application of Twitter sentimental analysis.IMDb ratings of movies can be used as performance measures to test the predictions made using Twitter data. This analysis can be implemented using textual, emoji ideograms or emoticon based tweets. We predicted the rating of movies from IMDb Database independently based on the movie name related tweets.**

## KEYWORDS

Twitter, Sentiment Analysis, Tweet, Tweepy, Movie Reviews, Emoji Ideograms, Data Mining, Random Forest Classifier, Support Vector Machines, Light GBM, F1 Measure

## 1 INTRODUCTION

The number of people adapting to Social Networking platforms, blogging and micro-blogging websites to express their opinions in forms for subjective texts or representative images, audios or videos is increasing exponentially. The amount of data generated from these sources has a variety of applications. Various studies and research projects are being conducted using this data. One such field of study is to measure customer response and reaction to various products and services. IMDb movie reviews have been used as a metric to gauge a movie's critical response, we used these ratings as the target variables for our proposed machine learning analysis. These reviews can be directly attributed to the general sentiment of the masses towards a movie. Sentiments can be mined from the emojis and text in movie-related tweets. Previous research has only emphasized on judging the text of the tweet using measures like polarity and subjectivity. We feel that Emoji Ideograms help better describe the sentiment of the tweet. Emojis directly relate to the feeling expressed by the user in a tweet. We used the emoji ideograms and texts from the data to design a comparative approach for machine learning models.

- Emoji Ideograms
- Text in tweets

We utilized this data independently and in combinations to test the prediction accuracy of the machine learning models. We treated it as a multi-class classification problem to bin the ratings into 4 categories

- Bad (1-4.9)
- Average(5-6.4)
- Good(6.5-7.9)
- Excellent(8.0-10)

We used the emoji data in this study which was at par with performance of textual features in the classification model and gave us valuable insights in this domain

## 2 LITERATURE REVIEW

Alexander Pak et al.[2] discuss how to collect and preprocess twitter data for sentiment analysis and opinion mining purposes. They build a sentiment classifier using this corpus that is able to determine positive, negative and neutral sentiment for a tweet by training two Bayes classifiers, which use the presence of n-grams and part-of-speech distribution information. They exclude the emoticons from the tweet and only use the language to classify the document.

Akshay Amolik et. al[1] use similar techniques as used by Alexander Pak et al.,[2] to perform sentiment analysis and opinion mining on tweets pertaining to the latest movies and classify them as positive, negative and neutral. They perform similar preprocessing of tweets and then perform classification using Naïve Bayes and Linear SVM classifier on text unigrams.

Andrei Oghina et al.[3], explore the combination of Youtube likes and dislikes and tweets of Twitter. The article tries to develop a relationship between the number of tweets and the rating given by IMDb. The intial results stated that the rating majorly depended on the tweets and not on the likes on Youtube. However, the model got the best results when the tweets and the like/dislike ratio was combined together. We think, Tweet being the decision making factor, adding Emoji data will make the Twitter medium more powerful for rating predictions.

Akshi Kumar et al.[4] proposed a novel approach for sentiment analysis using Twitter data. They calculated sentiment

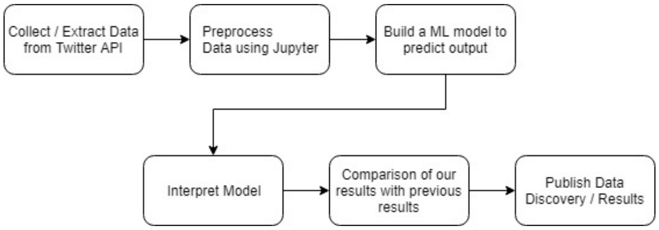
Collect / Extract Data from Twitter API
Preprocess Data using Jupyter
Build a ML model to predict output
Interpret Model
Comparison of our results with previous results
Publish Data Discovery / Results

**Figure 1: Design Methodology**

score to identify whether the comment is positive negative or neutral using opinion words (combination of adjectives along with adverbs and verbs) based on a linear equation.

Vaibhavi Patodkar et al.[6] proposed a system for sentiment analysis to understand the response of the public to different television shows. They have categorized the collected data into two types - positive and negative. After testing algorithms like Naïve Bayes, Decision Tree, Support Vector Machines and Random Forest, Random Forest gave the best performance.

Wernard Schmit et. al [7] talk about the use of Twitter data to match the ratings of public voting portals like IMDb. Regression and Classification were explored on the Twitter data to predict movie revenues or to classify sentiments respectively. The paper highlights the use of K Fold cross-validation, classifying conditions and Feature generation using Natural Language Processing approaches. The training data is solely dependent on Twitter data, which the authors describe as a bottleneck in comparison to other studies. The article gives reasonable explanations for implementing the Linear Regression and Support vector regression models for predicting sentimental and IMDb scores. Support Vector Classifier and Stochastic Gradient Descent Classification was used for classifying the groups of IMDb ratings.

Wieslaw Wolny [8] states that emoticons and emojis can be very useful for sentiment analysis especially when combined with analysis based on the Twitter names and hashtags. They combined the Natural Language Processing and Symbol Analysis approaches.

Wieslaw Wolny [9] describes the difference between a sentiment and an emotion which is essential for our study on emoticons and emoji ideograms. It builds a relationship between the sentiment polarity, emotion classes , and emoticons. It explains the steps to extract the Twitter data using Tweepy and searching for the emoticon strings in this data.

# 3 METHODOLOGY

In the first step, we collected the data related to various movies, movies were chosen to make the training dataset and test dataset. The second step is preprocessing. We cleaned the data and transformed it into the desired format using Python. Depending on the Twitter API restrictions, we processed the data for movies in a particular time frame. The clean data was used to train our Machine Learning models.

In our study we used Jupyter Notebooks and Google Colab based on Python 3.6. Libraries such as numpy, pandas, Matplotlib, ScikitLearn were used for data analysis. Tweepy was used for procuring data from Twitter.

# 4 DATASET DESCRIPTION

We extracted the data for our study from Twitter, this comprised of the tweets which we used for our analysis. The dependent variable i.e. movie ratings were obtained from IMDb, this data is readily available from various data sources across the net. We made use of the Twitter public API to retrieve tweets posted for the relevant movies. We used a 3 week frame to pull the data based on the movie names from $14^{th}$ March 2019 to $4^{th}$ April 2019. We used Twitter data object that provided various data points such as id, created at, tweet text, user and source information, location, retweet count, reply count, favorite count, language etc. Due to restriction with Twitter data concerning historical data, we were not able to get quality tweets for older and lesser known movies. The generic behavior of trending topics sustaining on the social media affected our over all corpus of tweets. Movies with at least 3000 tweets were considered. Only these 3000 tweets were considered in the corpus for each movie. Still we were able to get a corpus of 1175600 tweets across 392 movies.Some movies with very generic names affected our corpus by pulling non-movie irrelevant data. We used the hashtag movie names(For eg. #piratesofthecarribean ) for extracting the tweets. After collecting this corpus we split the data for training and testing in ratio 80:20. We further split the training data into training and validation sets.

# 5 FEATURE ENGINEERING

## 5.1 Emoji Based Features

As we wanted to study the correlation between the emoji data and Movie rating we decided to generate some features just based on emoji data. We filtered emojis from the tweet text and added them as different features for further processing. We used the Vader Sentiment Analyzer to get the sentiment score of the sentence and emoji list separately. We used the last emoji polarity as an independent feature. This feature was important as a way of describing an overall sentiment score of the whole tweet. We differentiated the emojis into positive and negative, to generate a number based feature counting the positive and negative emojis.

## 5.2 Text Based Features

We used the Vader Sentiment Analyzer to get a polarity of the whole tweet text without the emojis. We used a negative word list to get a count of negative words in the tweet text.

## 5.3 Tweet Text Preprocessing

Intially we tokenized the tweet text and removed the stop words using standard NLTK stopword list. We reduced the words using WordNet Lemmatizer to their root forms based on part of speech, such that we would have lesser unique words across the data, there by reducing the dimensions.We used stemmer to stem the words to further reduce the unique words. We explored Lancaster and Porter Stemmer to check its effect on the tweet text. However after using stemmer many important words got reduced to meaningless root forms. So we decided to drop stemming and just use lemmatization. We used the Tfidf vectorizer with the $max\_df$ and $min\_df$ to limit the very frequently ocurring words and drop very rare words which are like outliers.

After applying this text preprocessing we got N features based on text and M features based on emoji ideograms.

# 6 EXPLORATORY DATA ANALYSIS

After the process of feature engineering, we proceeded to derive insights from the data through the process of Exploratory Data Analysis. This is done to have a better understanding and summarize the data. First, we looked at number of movies per label, we observed two minority classes - Bad and Excellent and two majority classes - Good and Average (Fig.??). Number of movies per label is as follows-

(1) Excellent - 17
(2) Good - 188
(3) Average - 157
(4) Bad - 30

This result is similar to the ratio of movies per label if we consider the entire movie list i.e. number of movies in the Excellent and Bad category are less than the number of movies in the other two categories.

Next, we looked at the movies with the highest number of 'Favorite'(Fav) or 'Retweet'(RT) count (Fig. ??,??). We can infer two things from these figures-

(1) The number of movies with the highest number of RT or Fav are mainly in 'Good' or 'Average' category
(2) We also a drawback in our approach which is, some of the movie names in the list are also famous hastags, therefore, on observing the tweets, we also found some tweets which were not about the movies such as 'where many infamous rallies had taken place.No one yet owned televisions.'

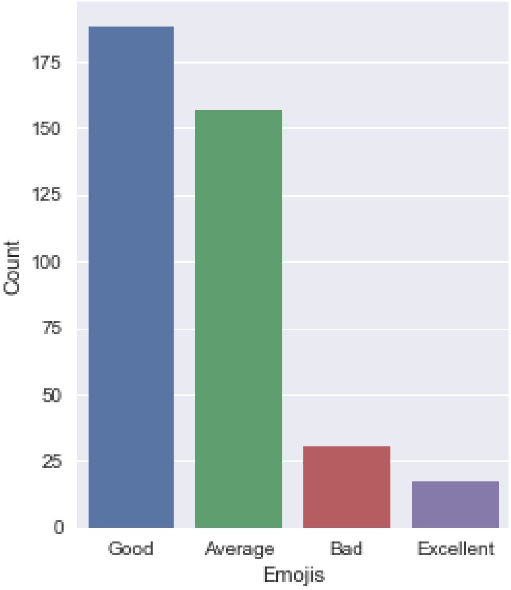

Figure 2: Number of movies in each label

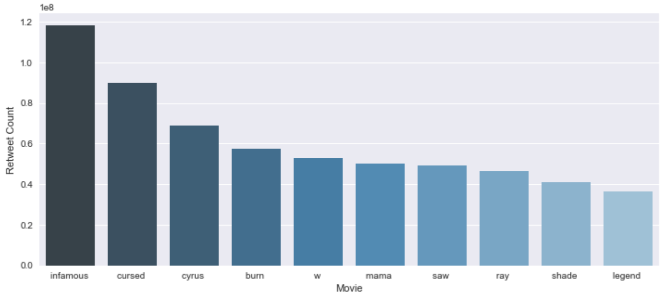

Figure 3: Movies with most RTs

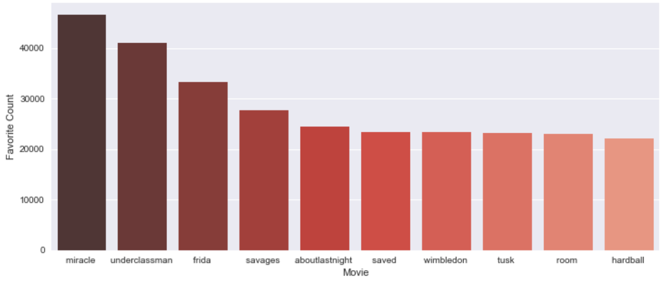

Figure 4: Movies with most Favs

We also looked at the emojis which were most frquently used, the top 10 most frequently used emojis are plotted in Fig ??. We can observed that most of the top-10 have a positive polarity indicating that peopl tend to used emojis more when they are posting a positive tweet about movies.

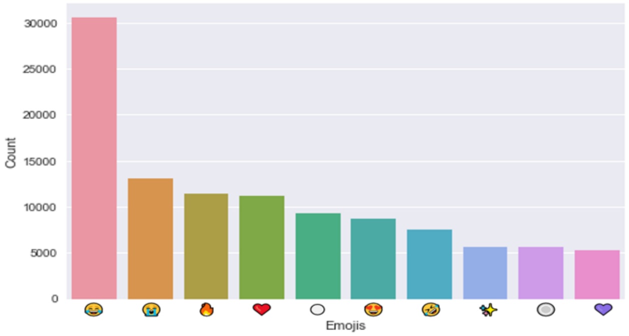

**Figure 5: Most frequent Emojis used**

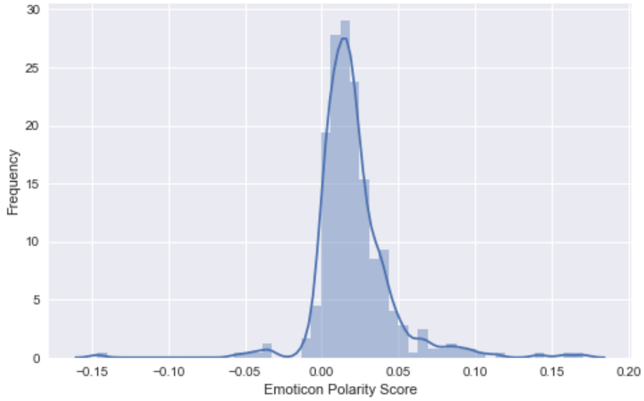

**Figure 6: Distribution of Emoji polarity scores**

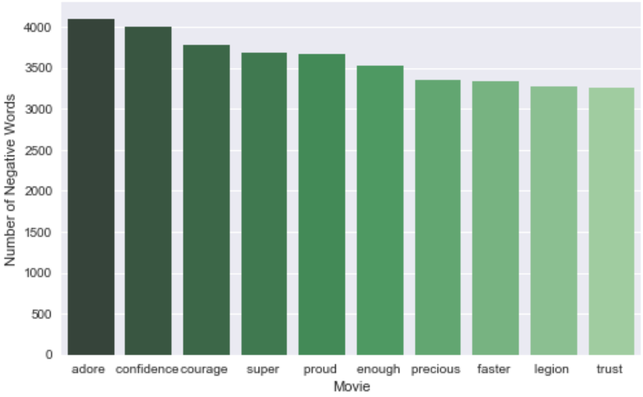

**Figure 7: Movies with most number of Negative Words**

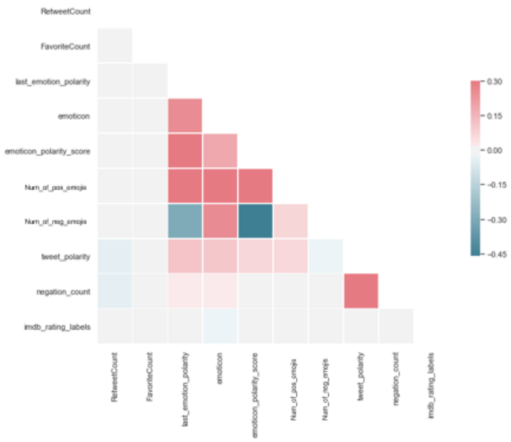

**Figure 8: Correlation Matrix**

This results is also observed if we plot the frequency of emoji polarity scores. It show a distribution with mean greater than 0 (Fig.**??**). We can also see similar results if we plot for the last emoji used in a tweet.

On looking at the column containing the count of negative words, we observed the movies with highest number of negative words in all the tweets of a particular movie. Here we saw a results which bolsters our approach which is the movies which were in the top-10 list were movies which were mainly in the 'Bad' category. This is because people tweeted negative comments about the movie. The list of movies along with their count of bad words can be seen in Fig.**??**.

Lastly, we plotted a correlation matrix (Fig.**??**) between the variables used for our analysis to check for redundancy and observe dependencies between variables. We can observe correlation in the range of -0.45 to 0.3, no pairs has a very high correlation which means that we do not have to exclude any variable from our analysis. We can see a moderately high negative correlation between number of negative emojis in a tweet and emoji polarity score which is obvious as negative emojis decrease the emoji polarity score. Also since majority of emojis used are polarity, we can observe a positive high correlation between number of emojis used and polarity scores.

## 7 PREDICTIVE MODELING

After performing the Exploratory Data Analysis, we proceeded to perform supervised predictive modeling using Machine Learning(ML) models. Our approach is apply the ML models on two data sets-

(1) Data that includes all the score based variables which were created during the features engineering phase, emoji based features and text based features
(2) Data that includes vectorized forms of all the words in the text, which is performed in the 'Tweet Text Preprocessing' step of feature engineering

We used three machine model on the given data-sets, Random Forest Ensemble

## 7.1 Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over-fitting to their training set. For both the datasets, we first divided the data into training and testing datasets using sklearn's 'train_test_split' function with the ratio of test set equal to 20% of the entire data. We then proceeded to apply and baseline Random Forest Classifier for the dataset of score based features. We obtained an accuracy of 56.95%.

*7.1.1 Parameter Tuning.* For hyperparameter tuning, we used sklearn's 'GridSearchCV function' to tune different values for 3 hyperparameters:

1. n_estimators: This hyperparameter is used to control the number of decision trees whose vote is counted to construct an ensemble random forest
2. max_depth: This variable is used to control the depth up-to which a decision tree can grow. This parameter is used to control for overfitting
3. criterion: This variable is used to specify the function which is used to measure the quality of split

The best parameters obtained after performing Grid Search are:

1. n_estimators = 100
2. max_depth = 15
3. criterion = "entropy" (Gini Entropy)

*7.1.2 Results:* After training the model on the best parameters obtained using Grid Search CV, we achieve a weighted F1 score of **0.64** on sentiment based features data and **0.63** on the vectorized text and sentiment based features data. We also find out the best features from the dataset which contains score based features which are obtained using sentiment scores of text and emoji using 'feature_importance_' function of RandomForestClassifier in sklearn. 'Retweet Count' and 'Tweet Polarity' are obtained to have the highest feature importance.

## 7.2 Light Gradient Boosting Machinery

Light GBM is a tree based model which grows at the leaf level based on the entropy calculations. It is a more DFS based approach than a level wise approach in the Random Forest. We chose Light GBM after reviewing readings which supported use of this model for large data and gave a better accuracy. Like Random Forest, this model also gave us a feature importance after training. We used this property to get important words in the corpus affecting the rating.

*7.2.1 Parameter Tuning.* Based on the requirement, we tuned the parameters to get a better performance on the test data. At the same time we ensured that the model does not overfit the training data as Light GBM may tend to do so based on tree complexity.

1. objective : Multiclass
   Since we have 4 classes to classify as Bad, Average, Good, Excellent.
2. metric : Multierror
   This error is used when the multiclass objective is needed. The aim of the model is to reduce this multierror.
3. $num\_class$ : 4
   Number of target classes.
4. $num\_boost\_round$ : 9000
   Boosting rounds were tried from 5000 to 9000. These rounds are to reduce the multierror parameter.
5. $early\_stopping_rounds$ : 100
   The boosting cycle stops if the multierror value does not change for 100 rounds.
6. $verbose\_eval$ : 25
   Step rate for the boosting rounds. We observed 25 gives us significant yet gradual change will approaching $num_boost_round$ value.

Additionally the target variable had to be changed to numeric values for Light GBM implementation. For example, Bad as 0, Average as 1, Good as 2 and Excellent as 3.

*7.2.2 Results.* After applying the above parameters we achieved an overall **AUC score** of **0.78** for the data.
The **Weighted F1 score** for emoji and sentiment based data were **0.57**.
The **Weighted F1 score** for emoji and textual based data were **0.61**.
As you can see in figure 9 the multierror dropped gradually to a value below 0.35. Adding the text data made this multierror drop steep but the error still stabilized at 0.35 like shown in figure 10.

## 7.3 Support Vector Machines

A Support Vector Machine is a supervised discriminative classifier formally defined by a separating hyperplane [5]. In other words, the algorithm forms a separating hyperplane, given labeled training data, which categorizes new data points. SGDClassifier from sklearn linear model for implementing SVM.

*7.3.1 Parameter Tuning.* Grid Search was used for hyperparameter tuning.

1. Loss function : Hinge Loss
   Loss functions are used to evaluate designed model.

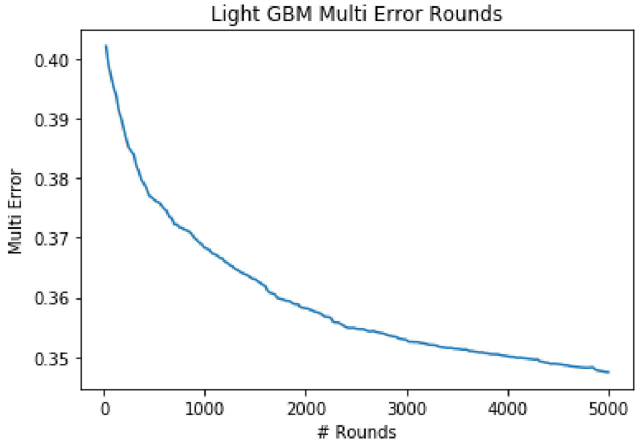

**Figure 9: Boosting Rounds for Emoticon based Data**

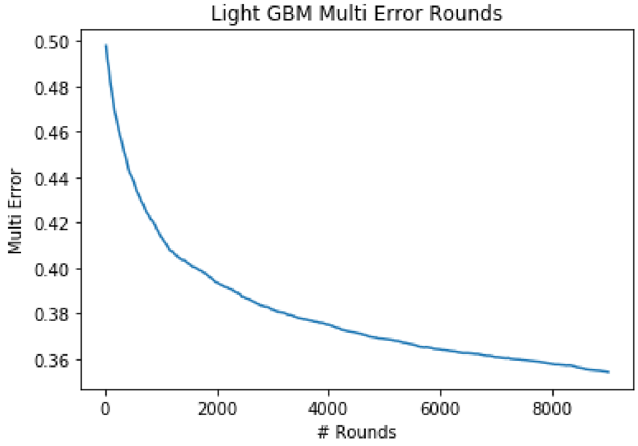

**Figure 10: Boosting Rounds for whole Data with 9000 Rounds**

SVM uses Hinge Loss which penalizes incorrect classification.

(2) penalty : L1
Penalty is used to avoid overfitting and to obtain relevant features. L1 (Lasso Regression) keeps only the important features, whereas L2 (Ridge Regression) reduces the weightage of less relevant features.

(3) alpha : $10^{-6}$
Alpha is the constant that multiplies the regularization term.

(4) maximum iterations : 75
The hyperparameter *max_iter* denotes passes over the training data.

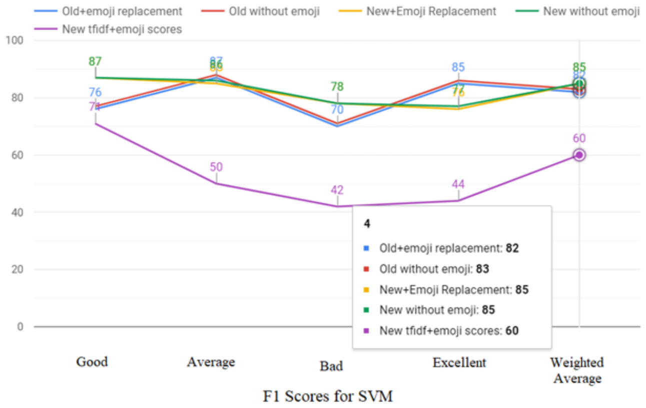

**Figure 11: Average and individual class F1 scores for SVM**

*7.3.2 Results.* After applying the above parameters we achieved an overall **Weighted F1 score** of **0.85** after replacing emojis with corresponding words and applying Count Vectorizer and Tfidf.
Method of using sentiment scores with emoji replacement resulted into **Weighted F1 score** of **0.60** .

Figure 11 shows the F1 scores obtained for different datasets used. Old dataset used was highly biased, so the F1 scores obtained for individual classes had higher variance. Whereas new dataset used had balanced data which resulted into better weighted F1 scores. Also, usage of emoji replaced data gave better performance as compared to usage of sentiment scores calculated.

## 8 RESULTS

The following are the weighted F1 scores for 3 models with and without the vectorized tweet text data :

| | Sentiment Analysis Scores | Vectorized Text and Scores |
|---|---|---|
| Random Forest | 0.64 | 0.63 |
| SVM | 0.60 | 0.85* |
| Light GBM | 0.57 | 0.61 |

From the above table we can see similar scores for both the datasets (with and without vectorized texts). The Support Vector Machine model gives the best scores when we add the emoji equivalent texts in the tweet texts feature and then vectorize them. All the other models gave comparable and similar scores for both the data sets in the range of 0.55 to 0.65. We can infer the following from the classification reports of the machine learning models:

(1) For Light GBM, we can see a good recall for the majority classes and a good precision value for the minority classes.

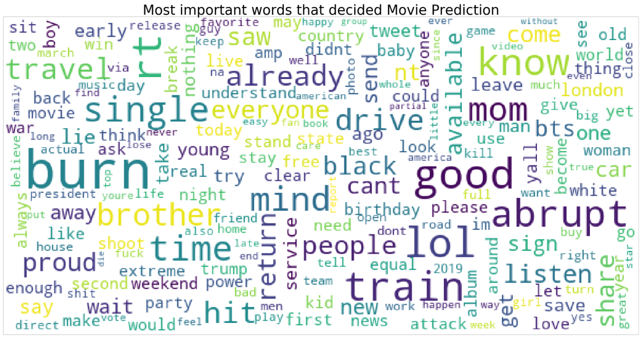

**Figure 12: Random Forest Word Cloud**

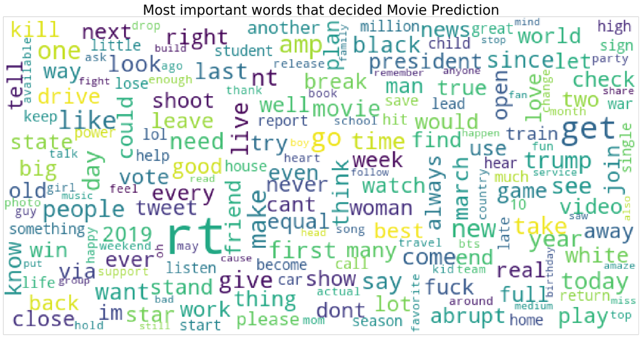

**Figure 13: LightGBM Word Cloud**

(2) For Random Forest classifier, we can observe a good recall value for classes ’Bad’, ’Average’, and ’Excellent’ and a good precision value for ’Good’ class.

(3) For SVM, we observed a good recall value for the ’Good’ class and a good precision value for all the other classes.

We checked feature importance for vectorized text features and plotted a word cloud after mapping these features to their corresponding words. Word Cloud generates a visual representation of the important words in a corpus. We observed words like ’goods’, ’get’, ’watch’, ’time’, ’lol’, ’look’, ’come’ etc. are important for the Random Forest and light GBM results. The word ’RT’ was seen for both the word clouds as a significant feature. Non-movie tweets generated from generic movie names tended to produce words like ’Trump’, ’Vote’, ’President’ etc. with many of them being politically associated words. Obtaining similar results of word clouds from two of the Machine Learning models reinforces the robustness of our analysis.

# 9 DISCUSSION

## 9.1 Conclusion

Previous work dealt with classifying a movie based tweet as positive or negative (binary classification). We extended this scope to further bin the movies into 4 classes. We still got comparable scores in terms of precision and recall. Previous studies on movie rating predictions were based on textual data only. Emoji based analysis was mainly used for sentiment identification. However, we combined these 2 approaches and achieved good scores for emoji data independently. Comparable scores for both data sets tell that emoji features are concise and reflective of textual data. We think the character limit in the tweet made the emoji data important.

Support Vector Machine model performed the best as a whole and for minority classes giving an Weighted F1 score of 0.85.

## 9.2 Future Work

As mentioned in the ’Exploratory Data Analysis’ section, we extracted a lot of tweets with movie names which were famous hashtags. These tweets were not relevant to the movies and produced skewed results. It would be ideal if we can come up with a framework which would detect these irrelevant tweets and remove them from the analysis. One resolution can be extracting tweets only close to the movie release dates which can reduce the amount of irrelevant tweets. We could also perhaps query for terms related to the movies such as actor names, genre along with the movie names and associate them with the movie names.

Removing the unbalance in the data by analyzing same number of movies in each class can produce better results in terms of accuracy. Redefining classes such that each class has similar number of records can also improve the results. We look forward to develop insightful trends and extend this study to other entertainment endpoints like Netflix buzz, Youtube trends etc.

# REFERENCES

[1] Mahavir Bhandari Dr.M.Venkatesan Akshay Amolik, Niketan Jivane. 2015. Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques. Retrieved February 4 , 2019 from https://www.researchgate.net/publication/291837156_Twitter_Sentiment_Analysis_of_Movie_Reviews_using_Machine_Learning_Techniques

[2] Patrick Paroubek Alexander Pak. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Retrieved February 4 , 2019 from https://www.semanticscholar.org/paper/Twitter-as-a-Corpus-for-Sentiment-Analysis-and-Pak-Paroubek/ad8a7f620a57478ff70045f97abc7aec9687ccbd

[3] Manos Tsagkias Andrei Oghina, Mathias Breuss and Maarten de Rijke. 2015. Predicting IMDB Movie Ratings Using Social Media. Retrieved February 4 , 2019 from https://staff.fnwi.uva.nl/m.derijke/wp-content/papercite-data/pdf/oghina-predicting-2012.pdf

[4] Akshi Kumar and Teeja Mary Sebastian. 2012. Sentiment Analysis on Twitter. Retrieved February 4 , 2019 from https://www.ijcsi.org/papers/IJCSI-9-4-3-372-378.pdf

[5] Savan Patel. 2017. SVM(Support Vector Machine)-Theory. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

[6] Sheikh I.R. Vaibhavi N Patodkar. 2016. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Retrieved February 4 , 2019 from https://www.researchgate.net/publication/317649188_Twitter_as_a_Corpus_for_Sentiment_Analysis_and_Opinion_Mining

[7] Sander Wubben Wernard Schmit. 2015. Predicting Ratings for New Movie Releases from Twitter Content. Retrieved February 4 , 2019 from https://www.researchgate.net/publication/301448987_Predicting_Ratings_for_New_Movie_Releases_from_Twitter_Content

[8] Wieslaw Wolny. 2014. Sentiment Analysis of Twitter Data Using Emoticons and Emoji Ideograms. Retrieved February 4 , 2019 from https://www.researchgate.net/publication/308413240_TWITTER_SENTIMENT_ANALYSIS_USING_EMOTICONS_AND_EMOJI_IDEOGRAMS

[9] Wieslaw Wolny. 2016. Emotion Analysis of Twitter Data That Use Emoticons and Emoji Ideograms. Retrieved February 4 , 2019 from https://www.researchgate.net/publication/308413240_TWITTER_SENTIMENT_ANALYSIS_USING_EMOTICONS_AND_EMOJI_IDEOGRAMS