Movie Rating Prediction Using TwitterFeed

Partho Mandal (ASU ID: 1209311008 , Contribution : 16.66 %)
Nikhil Vementala (ASU ID: 1209370509 , Contribution : 16.66 %)
Sameeran Rao (ASU ID: 1209126785 , Contribution : 16.66 %)
Sai Kumar Chelkapally (ASU ID: 1209243018 , Contribution : 16.66 %)
Manogna Vennamaneni (ASU ID: 1209373643 , Contribution : 16.66 %)
Madhulahari Illuri (ASU ID: 1209311047 , Contribution : 16.66 %)

Abstract:

Movie ratings are influenced by a lot of factors. One of the main factors is the public opinion which is generated for a movie on social networking sites. Successful analysis of user tweets can lead to successful prediction of movie rating in IMDB. Prior research done in the area of movie rating prediction is done by using google trends and that research was explained in the subsequent sections. The results of that study was not satisfactory, the features used by the machine learning algorithms are also limited. The drawbacks of that was limited features and restricted data available. These limitations have be overcome in our study by using textual features and using TF-IDF vectors. The dataset for the study is being prepared by using the twitter feed against to the dataset which was created from IMDB in prior studies.

The main idea of the project is to extract features from twitter feed and use TF-IDF vectors to generate weight vectors and then use the machine learning algorithms to predict the rating for the movie. F1-score is being used as the quality measure.

The prior work, literature survey i.e, all the algorithms which have been used in the course of the project, results of the current project, Conclusion and future work, this is the order in which the following document is being arranged.

In what follows we have given a detailed explanation of how the project is being implemented and their results have been detailed and explained.

Introduction:

In recent years, public opinion has increased a lot in each and every matter. People discuss a lot on various topics and give their views and feedback on them. Social networking sites such as Twitter, Facebook give a good platform for such public opinion sharing. One such discussion that frequently happens on such sites are movie discussions, such as how good a movie will perform or what are the expectations of how much the movie will collect, how good a movie is, will it retrieve its investments and the discussion goes on. Because of the large number of people using these platforms a movie success is largely dependent on it. A success of a movie is measured by a rating associated to it, the rating is given out of 10. IMDB(internet Movie DataBase) is a website which gives rating to a movie based on user ratings on their site. The goal of the project is to find out rating for a new movie solely based on the public opinion generated on twitter platform. The basis for doing this is the textual data which can be acquired from twitterfeed. As stated before, twitter as millions of users and users like to share their opinion on various matters publicly and the best feature of twitter is the use of a hashtag(#somedata). Using hashtags the textual data from twitter can be categorized. For example all the discussions surrounding a movie will be tagged with a hashtag of the movie name. So, when people express their views on twitter and tag the tweet for a particular movie, it can be captured[Pak & Paroubek, 2010] and also the reason for choosing twitter data rather than any website which provides movie reviews, because we see that this twitter data is in more consistency with public opinion so it can be used to provide results with much better accuracy.

A tweet can be read as whether positive or negative by doing some sentiment analysis over the tweet. A major portion of rating prediction i.e telling whether the movie has positive response or negative response can be determined by doing sentiment analysis, for finding the appreciation of a movie or an event some tweets might serve the purpose as they give the general directive of what the public attitude is like. Sentiment analysis is a classification task, as we need to classify the tweet as a positive or a negative one. The aim of the project is to predict a rating for the movie, to do this we need to have numerical data. To get the numerical data from textual data we perform some machine learning and data mining techniques. This type of learning is called predictive text mining. Predictive text mining is generally used in stock prediction, that is based on press releases by companies, the actions taken by companies, the stocks prices of the following day are being predicted. This is somewhat similar to predicting a movie rating from a database of tweets which is similar to press releases.

For the prediction model to work, machine learning algorithms need to be implemented on the corpus of tweets and features from those need to be extracted. Algorithms such as Naïve bayes model, multinomial logistic regression and support vector regression, these are the algorithms which have been used to extrapolate the textual from the tweets and used to predict the movie ratings.

Previous work:

In 2012, Deniz et.al of Stanford university tried to develop a movie rating prediction algorithm based on the google trends. The idea behind this approach is to characterize each movie by using a set of features and then searching for those features on google. The idea is that, by searching for those features, if a movie is popular then the search for a specific feature leads directly to that movie. Machine learning algorithms such as multi layer perceptron, Support vector machines, logistic regression are being used to extract the features and perform the experiments.

The dataset was prepared as follows, a list of movies is prepared and the required data about the movie is being extracted from the IMDB database, for each movie the director, actor actress, movie title and the release date are being stored. This is done for about a list of 400 movies. Search is done on google trends. Now, google trends gives us a search popularity index for a specific query for a specific time in a specific place. The index is determined as follows, total searches for a specific query divided by the total searches done in that specific region. For each movie the google trends data is being captured from one month before the movie release to the current period. And for the second experiment, data is collected 4 months prior to movie release to 4 months after movie release.

Results of the First Experiment:The movie rating is being predicted as a two class classification problem in supervised learning. The problem is determining if a movie rating is high or low. All the movies with rating less than 6 are assigned to low class and all the movies with rating greater than 6 are assigned to high class. The classes were now compared in logistic classifier, l1 regularized SVM and a multi layer perceptron and the following results were generated

Correctly classified	Overall:	High rated movies:	Low rated movies:
Logistic	49.07%	44%	54%
ℓ_1-SVM	54.63%	11.26%	98%
Multilayer perceptron	51.85%	69%	35%

Table 1: Distribution of movie ratings in the first experiment

It was evident from the results that all the algorithms perform incorrectly. The result of the logistic regression is somewhat like a 50 percent game. The movies are classified with 50 percent accuracy.

Result of second Experiment: In this experiment the second dataset which was taken from one months prior to movie release till four months after movie release are being used. This dataset will contain most valuble information about the movie. Now the results are as follows

Correctly classified	Overall:	High rated movies:	Low rated movies:
Logistic	64.13%	63.53%	64.64%
ℓ_1-SVM	72.25%	72.22%	72.27%
MLP (3 hidden layers)	68.32%	67.37%	69.23%

Table 2: Distribution of movie ratings in the second experiment

The same algorithms perform with better accuracy rate when the information used to predict the movie rating has been increased.

Though this work has tried to predict the rating for a movie, but the data required for completing the calculation is huge and our proposed project tries to predict the rating of a movie just after the movie release. Here the features are the movie title, actress and actors and they have been already defined as features, but in the proposed project even though nothing is known of the movie cast, we can predict movie rating solely depending on the public reactions.

Problem Description:

Public opinion matters a lot when it comes to determining the value of a product. This is also the case for movie success and its rating. Conversion of textual features into numerical values to predict a movies value is the in hand problem. The problem being addressed in this project is, prediction of a movie's IMDB rating based on twitter reactions got for that specific movie.

Softwares and Tools Used:

1)Python

2)Scikit learn package

3)Tweepy package

Methodology

1. Data Creation

We used Twitter REST API(tweepy) to extract tweets of a movie using its respective hashtag. We have collected tweets for about 20-30 movies. One of the main challenge of using tweepy api was a restriction on amount of tweets that can be extracted. Specified api only provides tweets of last 7 days. So we mainly focused on newly released movies.

2. Data Preprocessing

Data preprocessing is the next step in dataset creation. From the data obtained in the above process, we remove the unwanted text.

1. First we store unique tweets in a list i.e. remove retweets
2. Remove stop words, non-ASCII characters, punctuations.
3. Replace movie titles, numbers, usernames, hyperlinks

After processing, the data will be stored in a csv file. This dataset will be used in all the three algorithms we have used namely **Support Vector Regression, Multinomial Logistic Regression, Multinomial Naive Bayes.**

3. Feature Extraction

The features we've used for training and testing our data is Term Frequency - Inverse Document Frequency vectors which is the TF-IDF vectors. TF- IDF is a numerical statistic that is intended to reflect on how important a word is to a document in a collection of corpus.

We have used Scikit (Python library) feature_extraction package and computed the TF-IDF vectors for the dataset. And these vectors are further transformed into a TF-IDF matrix.

4. Dimensionality Reduction using Matrix Factorization

Dimensionality Reduction is the process of reducing the number of random variables under consideration, via obtaining a set "uncorrelated" principal variables.

Non-negative Matrix factorization is a feature extraction algorithm that decomposes text data by creating a user-defined number of features. NMF gives a reduced representation of the original text data. It decomposes a text data matrix Amn where columns are text documents and rows are attributes or keywords, into the product of two lower rank matrices Wmk and Hkn, such that Amn is approximately equal to Wmk times Hkn. In NMF, in order to avoid cancellation effects, the factors Amn and Hkn should have non-negative entries. NMF uses an iterative procedure to modify the initial values of Wmk and Hkn so that the product approaches Amn. The procedure terminates when the approximation error converges or the specified number of iterations is reached.

In this project, as we deal with TF-IDF vectors to measure the importance of the word from a bag of words and as we are considering huge input data, we get a bigger TF-IDF matrix with tweets as rows and word importance as columns. It is difficult to deal with such huge matrices in real life, as it may take a lot of time to perform any operations on it. So we have reduced the rank/number of components to 130 using non-negative matrix factorization.

5. Scaling of data to Gaussian distribution

Standardization of datasets is a common requirement for many machine learning estimators implemented in the scikit; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with zero mean and unit variance.

In practice we often ignore the shape of the distribution and just transform the data to center it by removing the mean value of each feature, then scale it by dividing non-constant features by their standard deviation.

For instance, many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the l1 and l2 regularizers of linear models) assume that all features are centered around zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

6. Grid Search and Cross Validation

Parameters that are not directly learnt within estimators can be set by searching a parameter space for the best Cross-validation: evaluating estimator performance score. Typical examples include C, kernel and gamma for Support Vector Classifier, alpha for MNB etc.

Exhaustive search over specified parameter values for an estimator. Important members are fit, predict. GridSearchCV implements a “fit” and a “score” method. It also implements “predict”, “predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used. The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

7. Final Parameters

SVR

Kernel: rbf; Gamma: 1e-1, 1e-2, 1e-3, 1e-4;

Kernel: Linear; C: 1,10,100,1000

MNLR

C: 1,10,100,1000

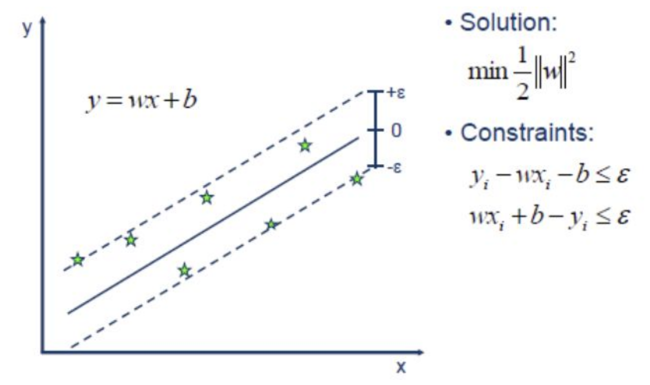
MNB

Alpha: 01, 0.01, 0.001, 0.0001

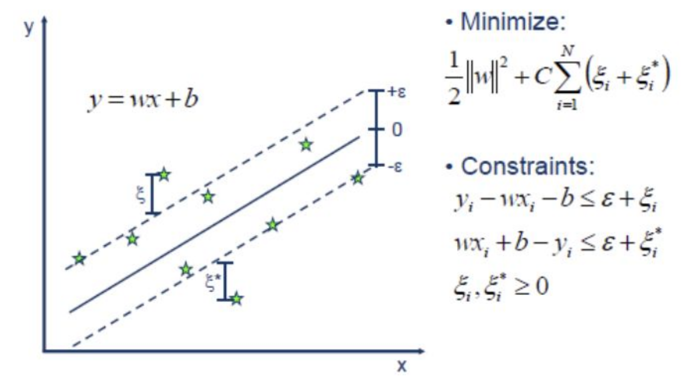
The following are the machine learning algorithms which have been used during the course of the project and F1-Score is the quality measure used. For the regression tasks Support vector regression is used and for classification tasks multinomial logistic regression and multinomial Naïve Bayes. F1 score is used as quality measure for classification tasks and MSE is used for regression tasks.

Support Vector Regression:

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.



y
$y = wx + b$
+ε
0
-ε
x
• Solution:
$\min \frac{1}{2}\|w\|^2$
• Constraints:
$y_i - wx_i - b \le \varepsilon$
$wx_i + b - y_i \le \varepsilon$



y
$y = wx + b$
+ε
0
-ε
ξ
ξ*
x
• Minimize:
$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$
• Constraints:
$y_i - wx_i - b \le \varepsilon + \xi_i$
$wx_i + b - y_i \le \varepsilon + \xi_i^*$
$\xi_i, \xi_i^* \ge 0$

Support vector regression is of 2 types, linear SVR, non-linear SVR.

One of the advantages of using SVR, is that it can be used to avoid difficulties of using linear functions for high dimensional features space and the optimization problem is transformed into quadratic programs. In the case of regression, the loss function needs to optimize the errors that are greater than threshold. Loss functions with value greater than threshold often lead to sparse representation of the decision rule, it gives a significant algorithmic and representational advantages.

Multinomial Naïve Bayes:

Naive Bayes method is a supervised learning algorithm, where the assumption that all the features are independent of each other is made. Given a class variable y and the dependent vector x1 to xn the following relationship is made:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

the denominator, is not to be taken seriously as the it is always constant, it doesn't have any effect on the overall probability. It finally sums up to

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

The naïve bayes classifier is a easy to understand and fast compared to more sophisticated methods. But it is prone to errors. Multinomial Naïve Bayes is a deviation from naïve bayes algorithm. In this algorithm the data is distributed multinomially. MNNB is a variant of naïve bayes algorithm in text classification The distribution is parametrized by vectors $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θ_{yi} is the probability $P(x_i \mid y)$ of feature i appearing in a sample belonging to class y.

The parameters θ_y is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

where $N_{yi} = \sum_{x \in T} x_i$ is the number of times feature i appears in a sample of class y in the training set T, and $N_y = \sum_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y.

The smoothing priors $\alpha \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha = 1$ is called Laplace smoothing, while alpha<1 is called Lidstone smoothing.

Multinomial logistic regression

Multinomial regression is the linear regression analysis conducted when the dependent variable is nominal with more than 2 levels. This is an extension of logistic regression which analyzes binary dependents. The multinomial regression does predictive analysis and what it does is that, it is used to describe and explain the data and the relationship between one dependent variable to independent variables. Linear regression requires the dependent variable to be continuous while logistic regression will work when the dependent variable is discrete.

Results:

While both regression and classification experiments used the same features, performances were different between regression and classification tasks. This section has a detailed explanation of how the implemented project was performing under various conditions. This section shows the results for the best performing configurations for both regression and classification tasks.

On visual as well as experimental inspection it can be noticed that as the total amount of data used for training increases, the quality of the results also increases. The best parameters for SVR found by automatic grid searching are:

Kernel: Radial Basis, C: 10, Gamma : 0.01

The datasets were divided into tweets of 20K, 40K,60K and 80K and the regression and classification algorithms were performed on them. The SVR-MSE learning curve and the MLNR-F1Score and MNB-F1Score learning curve have been represented in the following paragraphs.

Figure 1 shows the learning curve for the best performing support vector regression configuration, performing 10 - fold cross validation for each experiment. This curve shows that it is likely that performance will improve with more data than was used in these experiments.

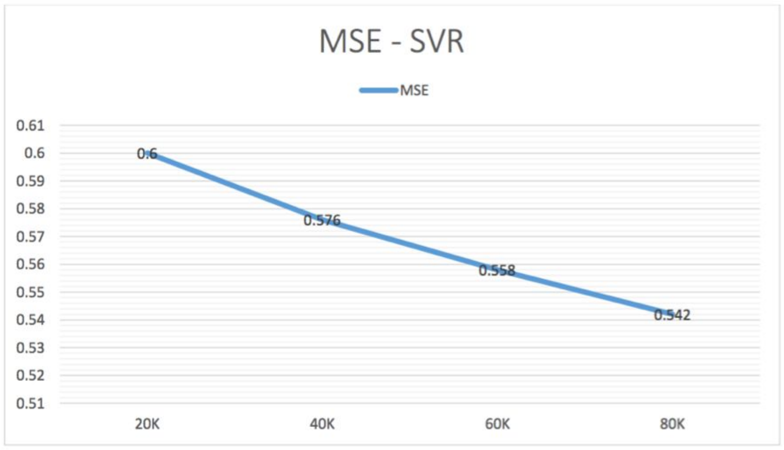


Fig 1: MSE score for SVR

Experiments with different amounts of training data for the best performing classification configuration again show that more data led to better results. Figure 2 and 3 shows the learning curve for the best performing classification using Multinomial Naïve Bayes and Multinomial Logistic Regression respectively. The best is Alpha value value for MNB is 0.01 and the best C value for MNLR came out to be 1. These experiments again used 10-fold cross validation for each experiment. The learning curve for the best performing classification configuration shows that it is likely that the optimal amount of training data has not yet been reached. It is quite interesting to see that for our training data, Multinomial Naïve Bayes with best F-score of 0.77 seemed to work better than MNLR F-score of 0.688.

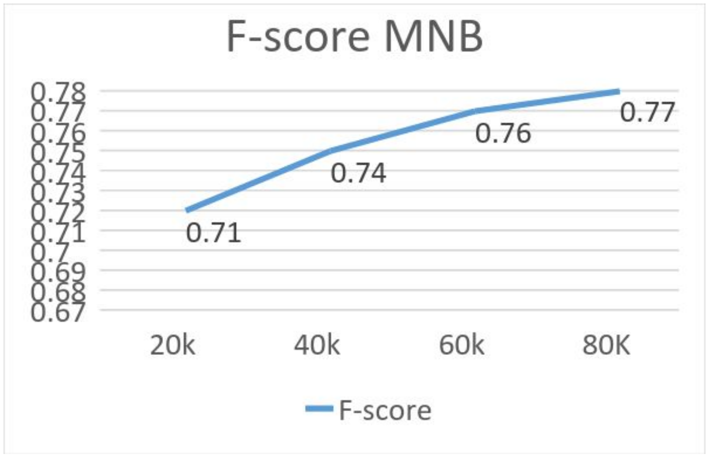


Fig 2: F-score for MNB

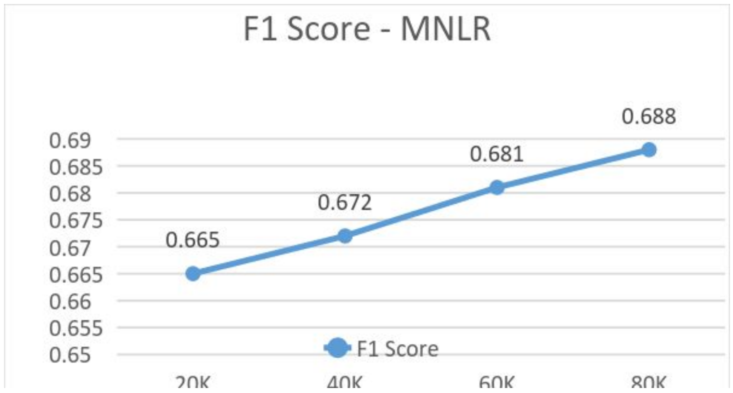


Fig 3: F-score for MNLR

Conclusion and Future Work:

In the project, successful implementation of a movie rating prediction system has been done. Machine learning algorithm such as support vector machines, multinomial logistic regression and Naïve bayes algorithms have been successfully implemented over the dataset. Both the prediction of exact numerical rating scores and the prediction of classes corresponding to a range of numerical scores, achieved a certain degree of success compared to their respective baselines.

The best performing regression configuration achieved an MSE of .542. This was achieved by using stemmed combinations of unigrams, bigrams and trigrams. There is still room for improvement. The best performing configuration of [1] achieved a RMSE of .523 for the prediction of IMDb rating scores, which translates to an MSE of .273. This model clearly outperforms our best performing configuration. However, our experiments focus solely on textual features derived from Twitter, as opposed to also including numerical features from other social media. Furthermore, in their model, more than 1 million tweets were used, whereas this study used a dataset consisting of roughly 80K tweets. It can be concluded that our best performing model is not the optimal prediction model for IMDB scores, but it does show that textual features can be useful for prediction of this kind.

Classification results also showed that predicting IMDB rating scores using tweets as training data can have a certain degree of success. The best performing configuration had an F- score of .77 in MNB.

Our classification results can be compared to other studies that performed classification tasks. The study of [2] explored 3-way classification for sentiment analysis. Their best performing model attained an F-score of .605. This is higher than our best performing score, but note that our experiments dealt with one more target variable. It should also be noted that this study deals with more general sentiment analysis, while our study is specifically aimed at predicting classes corresponding to IMDB scores

So, we see that the regression/classification system implemented has given comparable results to other studies which perform tasks. Our study deals with a dataset which is more generic in nature and has more than one target variable and has successfully proved that classification approach can be used for predicting movies.

In the future the project can be further extended by applying new weights to new features, finding prominent features and giving more weights to those features. One such correction that needs to be done is that, when a celebrity re tweets a tweet, then that tweet needs to be given more weightage. By doing so we increase the chances to getting more accuracy while predicting results.

Also, there are a lot of possibilities as to how this project can be taken forward, different algorithms, new feature set can be tried. One thing that is decisive in this study is the use of stemming, this reduces a lot of overhead work while assigning weights to words. In the current study the focus was only on textual features, but the future scope can include using more than textual features, such as number of retweets, previous box office history of the participants in the movie. While the focus was more on predictive text mining, we can consider the following options for future research. Performing sentiment analysis on the database would give us a fairly good idea on what side the movie will fall, and then if we use movie rating prediction we might get better result.

Preprocessing is also one stage we can use a lot of work, In the course of this study we have use TF-IDF vectors to store the database, but this database comes to a very large size and implementation of machine learning algorithms on this database is very hard, as that would require a lot of computation power, so working towards storing the database in a better manner would surely help in increasing the accuracy of the system.

References:

[1] Oghina, A., Breuss, M., Tsagkias, M., & De Rijke, M. (2012). Predicting IMDB Movie Ratings Using Social Media. Amsterdam: ISLA.

[2] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of Twitter data. LSM '11 Proceedings of the Workshop on Languages in Social Media, Pages 30-38.

[3] http://www.fastcompany.com/1604125/twitter-predicts-box-office-sales-better-prediction-market-updated

[4] http://www.lfd.uci.edu/~gohlke/pythonlibs/

[5] http://scikit-learn.org/stable/modules/cross_validation.html