Predicting retweet count based on sentiment analysis of historical tweets

Alex Szilagy, Animesh Patni, Chandana Ravindra Prasad

ABSTRACT

Our project revolves around analyzing the sentiment of an incoming tweet and performing predictive analysis on the retweet range given by a specific user. The data that was utilized was gathered through the Twitter API. To ensure that the token vocabulary was large enough and covered varied sets of words, we chose six different categories, in which each of the categories includes profiles of both male and female so that the data bias is minimized. In total, we gathered 38,400 tweets from 12 unique profiles. 1,200 of these tweets were manually tagged to serve as a baseline result to compare the others results against. Our main goal of this project was to study, observe, and implement various machine learning algorithms to find the ones that best fit the needs of this problem. In order to achieve this goal, we decided to choose the best model among Logistic Regression, Support Vector Machines and Naive Bayes. On careful analysis of sentiment classifiers, we found that the accuracy of Naive Bayes was comparatively higher than the other two. In the predictive analysis, we compared the workings and results of Linear Regression and Random Forest Regressor. One of the important steps here was to recognize the different features that were affecting the retweet count. Sentiment polarity alone as a feature would not be sufficient to obtain promising results. Due to this, we have chosen a total of five features, which when used along with sentiment polarity, gave an interesting insight which we have explored in this project.

INTRODUCTION

The problem that we attempted to solve was to predict the range of retweets a certain tweet would get based on historic data gathered through twitter API. The reach of a tweet primarily depends on the category of the profile and the number of followers a certain person or profile has. By observing the dynamics of aggregate behaviour, having information about all the followers of a certain profile would enable us to understand the cascade of a tweet and effectively determine its consequences. If not for the API limitations, the dynamic follower count would have been very useful in analyzing the reach and growth of a specific tweet. The research papers that we referenced suggested that the sentiment of a tweet had an impact on the number of retweets a tweet gets. Of all the different approaches that were explored, we decided to examine any possible correlation of sentiments and retweets as this approach was the one we could effectively test and implement in the time frame we were allowed.

RELATED WORK

Prediction of retweet cascade size over time; Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, Andrey Kustarev, Yandex Leo Tolstoy st. 16, Moscow, Russia

The authors of this paper [1] provides an analysis of retweet cascades. The main idea behind this paper is to predict the size of the cascade of the tweet at a certain time in the future. This idea is similar to that of ours in the sense that we wish to examine the spread (or retweet) of a single user's tweet. In addition, this paper mentions some features of a tweet such as a user's social network (number of friends, followers, etc..), content features (the length of the tweet, its sentiment analysis, etc..) as well as a few others. This paper gave us an insight as to how the features impact the number of retweet a tweet can get. It was a very informative paper and helped us in a careful analysis of feature selection.

Analyzing User Retweet Behavior on Twitter; Zhiheng Xu, Qing Yang Institute of Automation, Chinese Academy of Sciences

In this paper [2], the authors provide an analysis of user retweet behavior on twitter. They do this by providing an analysis of retweet behavior and train prediction models on different classification frameworks. This is very similar to our first step, in which we wish to analyze different classification frameworks as mentioned above to ensure a high accuracy is obtained. This paper primarily focuses on reflecting a variety of features from the user and reflects the importance of a variety of features. Instead of comparing these features, we wish to perform analysis utilizing semantic values for tweets.

Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network;
Bongwon Suh, Lichan Hong, Peter Pirolli, Ed H. Chi
Palo Alto Res. Center, Inc., Palo Alto, CA, USA
This paper [3] provides an analysis of tweet features and their correlation to retweets. Some of these features include the number of URLs in a tweet, the number of followers, hashtags, days, status, and favorites. They have found that URLs and hashtags correlate highly with retweetability. Although this paper is fundamentally different than ours, it provides a frame of reference that must be recognized. In order to eliminate any bias in our data and ensure that the attention that we are trying to measure correlates a tweet's semantic analysis, we must be open to recognizing any additional features that play a large role in the correlation of retweets.

Understanding Email Writers: Personality Prediction from Email Messages Jianqiang
Shen, Oliver Brdiczka, and Juan Liu Palo
Alto Research Center, 3333 Coyote Hill Road,
Palo Alto, CA 94304, USA {jianqiang.shen,oliver.brdiczka,juan.liu}@parc.com
In this paper [4], the authors infer the personality of users based on the content of their emails. Such inference can be used for various applications such as better personalization, recommendation, and targeted advertising. They have a way to deal with the private and sensitive nature of email content without barging in on the privacy. Based on the Big Five personality model, they have trained predictors to work on extracted email features. The authors report prediction performance of 3 generative models with different assumptions. The results show that personality prediction is feasible, and the email feature set can predict personality with reasonable accuracies. The authors speak about two approaches for developing their SOCAL (Sentiment Orientation CALculator), one being lexicon-based approach and text classification approach being the other one. The dictionary for lexicon-based approach is created manually with the focus on adjectives being used as indicators of the semantic orientation.

Twitter Sentiment Analysis; Sarlan, A.; Nadam, C.; Basri, S.
The authors of this paper [5] examine sentiment analysis of web-based applications and the limitations that are faced. These limitations appear in the use of inappropriate English (slang terms) as well as the limitations placed upon a text-only message, which are void of all facial expressions and body language. This paper provides the groundwork for a lexicon-based and machine learning based approach to sentiment analysis. This groundwork supports our paper but does not carry out any technical implementations.

APPROACH

The entire flow of the project is as follows: We start by gathering data systematically, keeping in mind the variety of tokens, category specific words, and biases. It is then followed by sentiment analysis, where we explore three different classifiers and ultimately select the one with comparatively higher accuracy. The chosen classifier is further used to serve as one of the features in the predictive analysis wherein the workings of Linear Regression and Random Forest Regression are tested.

1. Data Retrieval

We have used Tweepy, a Python library through which we could access the Twitter API. On making an API call for "user_timeline". This returns a collection of the most recent tweets posted by the user. We have chosen to gather data from 12 different timelines in which two users, one male and one female, were selected to represent a respective field. The 12 users that were selected are as follows : ['BarackObama', 'HillaryClinton', 'KingJames', 'serenawilliams', 'katyperry', 'justinbieber', 'jk_rowling', 'johngreen', 'SpaceX', 'pepsi', 'elonmusk', 'mtbarra']. These users were selected based on activity to ensure that enough data could be gathered from their respective timelines. By selecting the users in this manner, it is possible to minimize any bias that may be put on the dataset. By understanding where the data is coming from, it is possible to gain a deeper understanding of the results without limiting the scope of our data.

The format of the extracted data from Twitter through Tweepy consisted of a variety of possible features. The Twitter API offers a variety of data that can be accessed. For our analysis, we utilized the Name (of the twitter account), the Id of the twitter account, the date the tweet was created, the text of the tweet, and the number of retweets and favorites that the tweet has received.

2. Sentiment Analysis

2.1 Support Vector Machines

SVM is a classifier with efficient memory functions that uses a subset of decision functions called support vectors. Linear SVC is capable of handling multi-class classification by using the option multi_class='crammer_singer'. This method is consistent as opposed to one-vs-rest classification. The confusion matrix lets us know how well the model can classify. Confusion matrix can be used to calculate attributes such as accuracy, precision, recall and f score. It is constructed using the labels predicted by the model against the true label of the data.

- True Negatives(TN:0,0) – Reviews predicted negative and contained negative
- False Positives(FP:1,0) – Reviews predicted positive but had negative sentiment
- True Positives(TP:1,1) – Reviews predicted positive and are positive review
- False Negatives(FN:0,1) – Reviews predicted as negative but was a positive review

Precision: When the prediction is positive, shows how often it is correct. *Recall:* When the review is positive, shows the number of positive reviews predicted correctly. *F Score:* Weighted average of *Recall* and *Precision* to combine the two attributes. *Accuracy:* The number of classifications done correctly for positive and negative reviews

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN} \qquad F\ Score = 2.\frac{Precision\ .Recall}{Precision+Recall} \qquad Accuracy = \frac{TP+TN}{Total}$$

Ideally, we look for a balanced score of *Accuracy* and *Precision* , to ensure the model to predicts positive, neutral,and negative sentiments correctly without having an overfitting issue.

2.2 Logistic Regression

Logistic Regression utilizes one or more predictor values to predict a categorical dependent variable. A Logistic Regression classification can be most commonly described as binomial, ordinal, or multinomial. Binomial logistic regression relies on the notion that the outcome is coded as a binary value. Ordinal Logistic Regression acts upon dependent variables that are ordered. Multinomial Logistic Regression deals with situations in which outcomes may have three or more possibilities. In this project, it was important to recognize that not all tweets contained only a positive or negative connotation and that such tweets would be recognized as neutral. For this reason, multinomial Logistic Regression was utilized to properly analyze each of these three fields (negative, neutral, and positive). The scikit-learn linear_model package was utilized to perform Logistic Regression and the scikit-learn model_selection package was used to perform cross validation in an attempt to recognize overfitting.

2.3 Naive Bayes

Naive Bayes is a simple yet powerful algorithm. The Naive Bayes classifier is based on the Bayes Theorem, and assumes that the value of a feature(x) on a given class is independent of the values of the other feautres. This assumption is called class conditional independence. The assumption is very strong but still it gives surprisingly well results. We used Naive Bayes from the Natural Language ToolKit package of Python.

Bayes Theorem: $P(label|features) = (P(features|label) * P(label))/P(features)$

The NLTK.classify package was used for Naive Bayes classifier. To get the accuracy of the model we used "nltk.classify.accuracy".

2.4 Text Blob

We used Text Blob from NLTK for preliminary sentiment analysis. When used on the gathered data, it classifies a tweet into positive, negative, and neutral. Although it was sufficient for the preliminary analysis, the accuracy obtained was not satisfactory. The classification feature in Text blob is implemented using Naive Bayes and Decision tree.

2.5 Vader

VADER [6][7] (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. We used this to get the compound polarity score which acts as a single measure of polarity (combines positive, negative, and neutral polarity scores).

2.6 Manual Tagging

Once the data was gathered, existing libraries were used to perform sentiment analysis, like text blob. It marked and returned the polarity based on their overall sentiment. Upon observing this automated process, we recognized that the outcome was unsuitable due to its lower accuracy. In order to obtain a higher accuracy of classifications, we manually tagged 100 tweets from each profile to get a baseline result to compare the others against other classifiers.

3. Predictive Analysis

In order to predict the retweet count of a new tweet for a particular user, we analysed varieties of tweets and different types of features from the user_timeline that would affect the retweet predictions. The features that were examined are listed in the next section (3.1).

3.1 Feature Selection

I. Followers Count:

This feature can be extracted from “tweet.user.followers_count” of the user_timeline. Tweet.user.followers_count returns the followers count at the time the call was made. In order to get historical followers count, the data had to be collected continuously for a long period of time (utilizing the live Twitter API). We are using followers count as one of the features because high follower count correlates highly to high retweet count (the reach of a tweet).

II. Friends Count:

This feature can be extracted from “tweet.user.friends_count” of the user_timeline. Friends count in Twitter represents the amount of other users that an individual is following. This feature gives us an insight into the users friends on twitter and can often display a deeper level of interconnectivity than follower count. This also returns the count at the time that the call is made. To get more insight (dynamic friend count), the data must been collected over a longer period of time.

III. Sentiment Analysis:

This is one of the features which we wished to analyze the impact on retweet count. By calculating the average positive, negative, and neutral retweet count we found that the average retweet value for a negative tag tweet is higher than those with other polarities. For this reason, we used the polarity tag of a tweet as a feature in our model.

IV. Average Retweet Count:

By obtaining historical retweet count for a user, we are able to gather an average measure of reach. Combining this with other features such as polarity and follower count will allow for an overall greater accuracy in our predictive model. We calculated the average retweet count for a specific user and used it as a feature column, which can be found as labeled as ‘avg_re’.

V. Average Favorite Count:

We calculated the average favorite count and used it as a feature column, which is labeled as ‘avg_fav’. This was done in an attempt to gather historical favorite count data and utilize it to predict a new tweet. Initially, we found that favorite count correlates with an accuracy of roughly 84% to retweet count (utilizing Random Forest and Linear Regression analysis). Since we do not have the favorite count of a new tweet at the time it is published, this measure was utilized to represent an overall measure of historical favorite data.

VI. Favorite Count:

In our first experiment we attempted to use favorite count as one of our feature columns. Since favorite count would be zero for a newly posted tweet, it is not possible to use in our prediction. For this reason, this feature was dropped as it would simply substitute as a retweet prediction count.

3.2 Linear Regression

Linear Regression predicts the score of one variable utilizing the score from the second one. The value we are predicting is taken as Y and the variable we are basing our prediction is called variable X (also known as predictor variable). Linear Regression attempts to find the best fitting linear line through the available data. This is noted as the regression line. In addition, the distance between each plotted point and the regression line represent the error.

3.3 Random Forest Regression

Random Forest Regressor operates by constructing a multitude of decision trees at training time and outputs the normalized mean prediction of each individual tree. We opted to use Random Forest since the model is considerably good at handling tabular data with numerical features and different categories. Random Forest also excels at capturing non-linear behavior between different features and the target. One setback is that it doesn't work very well with sparse features, so in future instances we can preprocess the sparse features to generate more numerical statistics.

EXPERIMENT

4. Sentiment Analysis Results

4.1 SVM

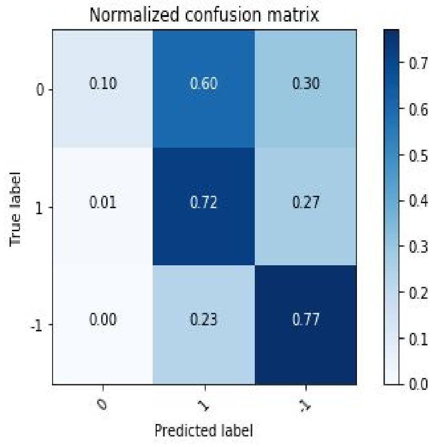
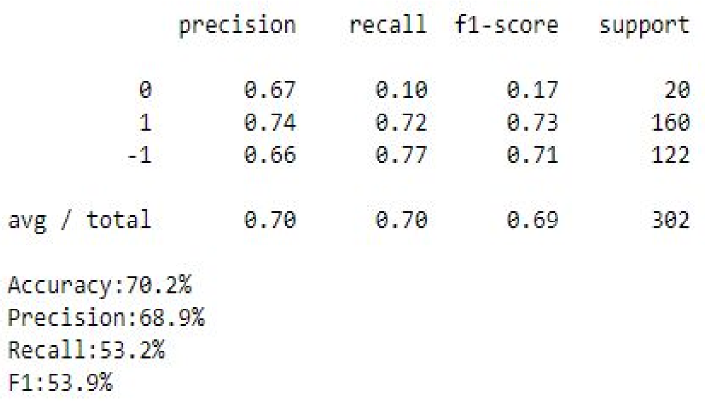


Fig.4.1.1: Normalized Confusion Matrix from SVM



```
             precision    recall  f1-score   support

          0       0.67      0.10      0.17        20
          1       0.74      0.72      0.73       160
         -1       0.66      0.77      0.71       122

avg / total       0.70      0.70      0.69       302

Accuracy:70.2%
Precision:68.9%
Recall:53.2%
F1:53.9%
```

Fig.4.1.2: Results observed in the SVM classification

The first figure above (Fig.4.1.1) depicts a confusion matrix, which is a table that describes the performance of the classification model on a set of data for which true values are already known.

The second figure above (Fig.4.1.2) displays a list of results obtained. We can observe the precision, recall, f1-score and support for positive, negative and neutral classifications.

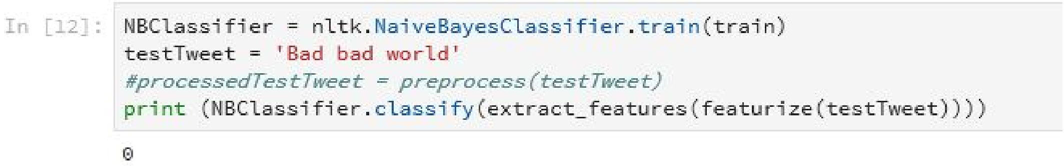
4.2 Naive Bayes

We opted to utilize Naive Bayes as one of the text classification model based on the notion that predictors are independent of one another, that is, the naive bayes classifier assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. It works well with the binary values.

The accuracy with the training set was 72.56%

Naive Bayes Accuracy with the Test Set: 71.00%

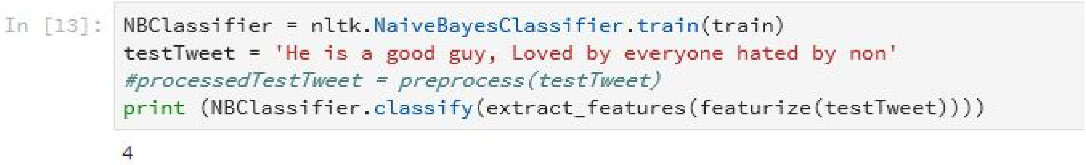
To improve the accuracy of this model, future work can include smoothing implementations. One of the cases where naive bayes performed badly is shown below: The model predicted a neutral polarity for a testTweet = “Bad bad world” which was due to there being a high amount of tweets with neutral polarity in our training model. This can be improved if the vocabulary contained more extensive negative words. One way to accomplish this would be to collect more real world data and closely examine each and every data point.



```
In [12]: NBClassifier = nltk.NaiveBayesClassifier.train(train)
         testTweet = 'Bad bad world'
         #processedTestTweet = preprocess(testTweet)
         print (NBClassifier.classify(extract_features(featurize(testTweet))))

         0
```

Another example where Naive Bayes predicted correctly is as follows: The model predicts a positive polarity which is acceptable.

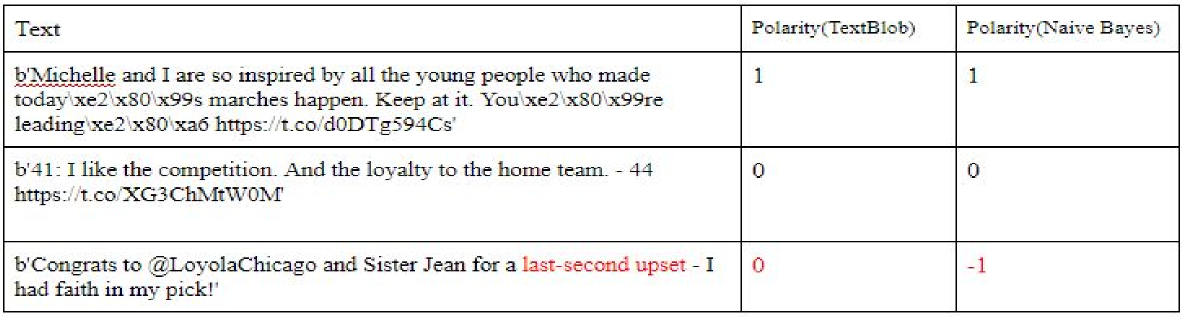


```
In [13]: NBClassifier = nltk.NaiveBayesClassifier.train(train)
         testTweet = 'He is a good guy, Loved by everyone hated by non'
         #processedTestTweet = preprocess(testTweet)
         print (NBClassifier.classify(extract_features(featurize(testTweet))))

         4
```

The below table shows the polarity result we ran on the data we scraped for Barack Obama.

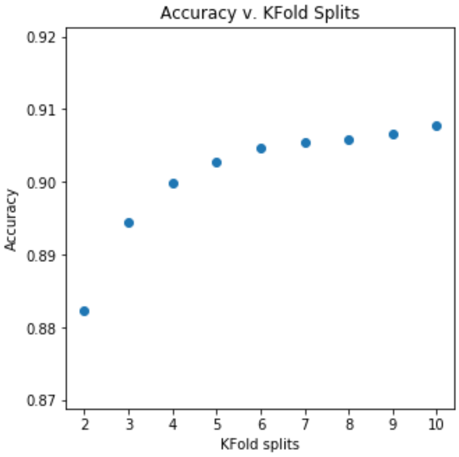
The disagreement that was found is shown in example in red. The tweet was tagged neutral by TextBlob but was tagged negative by Naive Bayes. This is due to negative terms existing that were not recognized by TextBlob.

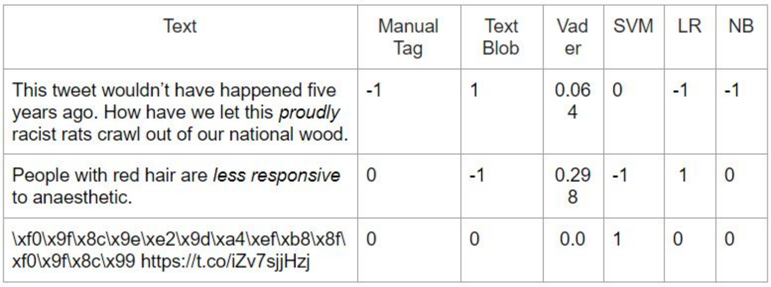


Text	Polarity(TextBlob)	Polarity(Naive Bayes)
b'Michelle and I are so inspired by all the young people who made today\xe2\x80\x99s marches happen. Keep at it. You\xe2\x80\x99re leading\xe2\x80\xa6 https://t.co/d0DTg594Cs'	1	1
b'41: I like the competition. And the loyalty to the home team. - 44 https://t.co/XG3ChMtW0M'	0	0
b'Congrats to @LoyolaChicago and Sister Jean for a last-second upset - I had faith in my pick!'	0	-1

4.3 Logistic Regression

In addition to Naive Bayes and SVM, Logistic Regression seemed to fit our problem statement well. This is based on the assumption that Logistic Regression performs well in scenarios in which the dependent variable is categorical. Initially, the Logistic Regression model was fit in an attempt to predict the Manual Tagging based on the Text. In this scenario, the accuracy of the model was 76%. By applying KFold to the model, a 10-Fold cross validation score was found to be 68.6%. This represents the overfitting potential for our model. In a second iteration of Logistic Regression, an LR model was fit to predict the Textblob polarity based on Textual analysis. This was done in an attempt to replicate a successful Logistic Regression on the entire dataset, given that only 1,200 of the tweets were manually tagged. In this scenario, Logistic Regression performed with an accuracy of 90%. Once again, a cross-validation score was created utilizing the scikit-learn KFold method. In order to be better suited to perform error analysis, the model was performed with KFold splits ranging from 2 to 11 (*Fig.4.3.1 - Accuracy v. KFold Splits*). An analysis was then performed on the top 10 most frequent terms as well as the top 10 highest weighted features. Upon analyzing each list, it was immediately recognized that links and emoticons played a large role in this Logistic Regression model.





Text	Manual Tag	Text Blob	Vader	SVM	LR	NB
This tweet wouldn't have happened five years ago. How have we let this *proudly* racist rats crawl out of our national wood.	-1	1	0.064	0	-1	-1
People with red hair are *less responsive* to anaesthetic.	0	-1	0.298	-1	1	0
\xf0\x9f\x8c\x9e\xe2\x9d\xa4\xef\xb8\x8f\xf0\x9f\x8c\x99 https://t.co/iZv7sjjHzj	0	0	0.0	1	0	0

Fig5.4.1 Highest Agreement observed among Naive Bayes and Manual Tag

5. Predictive Analysis Results

5.1 Linear Regression

One of the two models that were utilized in the predictive analysis was Linear Regression. This was initially utilized as described above, in order to find a correlation between favorite count and retweet count. As mentioned, using favorite count to predict a retweet gave us roughly 84% accuracy with Linear Regression. By utilizing this knowledge, it could be inferred that using historical favorite count would be a feature worth examining. While performing predictive analysis with the second set of features (as discussed in section 3.1), Linear Regression (*Fig. 5.2.1)* was found to give a much lower accuracy than Random Forest Regression (*Fig. 5.2.2)*. This is due to Random Forest Regression being capable of capturing non-linear behavior between the features and the target.

5.2 Random Forest Regression

The second model that was utilized in assisting with predictive analysis was Random Forest Regression. The features utilized in this model are those discussed in section 3.1 of the predictive approach. *Fig. 5.2.2* demonstrates the different ranges of predictions that were tested and the total percentage of tweets contained within that range. The graph reflects 3,200 tweets from Barack Obama, out of these 3200, around 2400 were used as training data and around 800 were used as testing data. Nearly identical percentages were calculated for other participants. From examining the graph, it can be determined that roughly 77% of all predictions were predicted within 1,500 retweet count of the actual value. Meanwhile, only 45% of the predicted values were within 1,000 retweet count of the actual value and only 19% of predicted values were within 500 retweet count of the actual values. It can be observed that our Random Forest predictor for Barack Obama, who has an average retweet count of around 5956, performed well for a range of 1000 (plus or minus the predicted value) retweet count. This can be improved by adding more feature columns since retweet count depends on a variety of features.

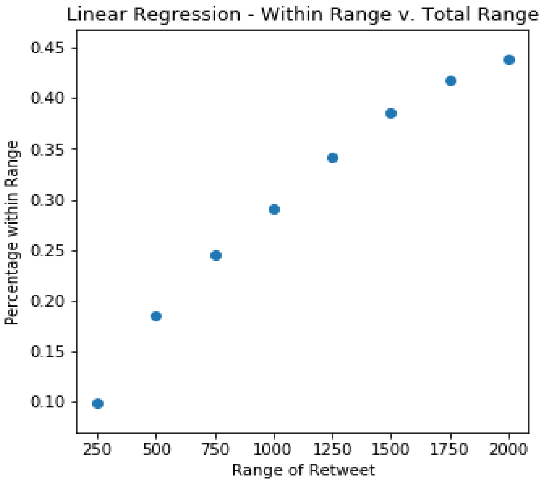


Fig.5.2.1: LR - Percentage of Tweets within Predicted Range

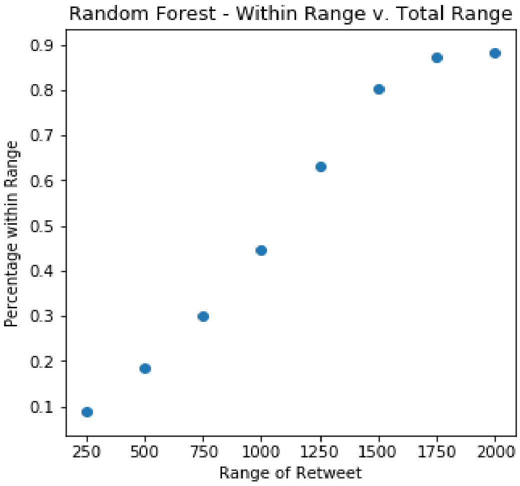


Fig.5.2.2: RF- Percentage of Tweets within Predicted Range

6. Error Analysis Results

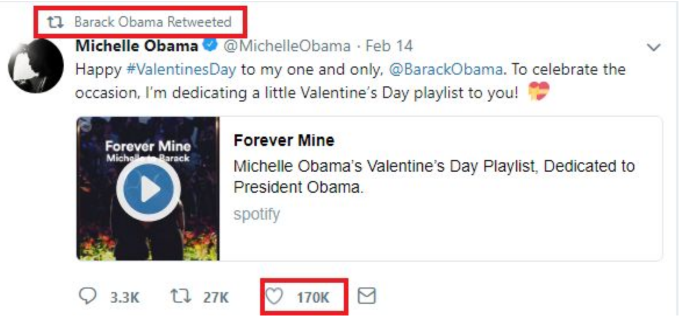
It is important to recognize some of the areas in our research in which error could occur. Some such areas are listed below in an attempt to be transparent in the analysis of any bias that exists in this project.

- The data that we gathered from Twitter was inherently biased due to there being comparatively more positive and neutral tweets than there were negative. This bias affects the way sentiments are classified. However, we cannot manually pick an equal number of positive, negative and neutral tweets and prepare a corpus, because, we cannot dismiss or add anything as every tweet is a crucial data point.
- This can be further observed in the case of a potentially viral tweet, which consisted of a link that directed to a video related to SpaceX. Since we regard every piece of information as a representative of a crucial data, we decided against removing emoticons or links from the tweets during pre-processing.

6.1 Outlier Analysis

It is worth noting that there was one major outlier in our data that was removed. This tweet was by Barack Obama and contained > 1,700,000 retweets and >4,600,000 favorites. In comparison, the average number of retweets in the entire dataset was ~37,000 and the average number of favorites was ~150,000. In order to ensure our model wasn't biased to this extreme outlier, it was removed.

Upon examining predictive analysis with Random Forest Regression, some data points existed that contained a large number of retweets but 0 favorite count. This is due to a function in the Twitter API in which the favorite count is not displayed for a tweet that the user retweets from another user; however, the tweet is still displayed on their timeline. One example of this is in the below figure. This causes the average favorite count to lower by a substantial amount while maintaining a high average retweet count. We could have removed all the retweeted tweets from the user profile but that would have greatly affected average retweet count and the insight about what kind of tweets a user in retweeting.



Barack Obama Retweeted

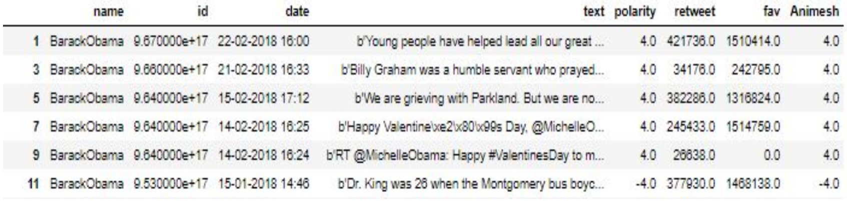
Michelle Obama @MichelleObama · Feb 14

Happy #ValentinesDay to my one and only, @BarackObama. To celebrate the occasion, I'm dedicating a little Valentine's Day playlist to you!

Forever Mine

Michelle Obama's Valentine's Day Playlist, Dedicated to President Obama.

spotify



	name	id	date	text	polarity	retweet	fav	Animesh
1	BarackObama	9.670000e+17	22-02-2018 16:00	b'Young people have helped lead all our great ...	4.0	421736.0	1510414.0	4.0
3	BarackObama	9.660000e+17	21-02-2018 16:33	b'Billy Graham was a humble servant who prayed...	4.0	34176.0	242795.0	4.0
5	BarackObama	9.640000e+17	15-02-2018 17:12	b'We are grieving with Parkland. But we are no...	4.0	382286.0	1316824.0	4.0
7	BarackObama	9.640000e+17	14-02-2018 16:25	b'Happy Valentine\xe2\x80\x99s Day, @MichelleO...	4.0	245433.0	1514759.0	4.0
9	BarackObama	9.640000e+17	14-02-2018 16:24	b'RT @MichelleObama: Happy #ValentinesDay to m...	4.0	26638.0	0.0	4.0
11	BarackObama	9.530000e+17	15-01-2018 14:46	b'Dr. King was 26 when the Montgomery bus boyc...	-4.0	377930.0	1468138.0	-4.0

CONCLUSION

Throughout the analysis of our project, we understood the workings of different machine learning techniques and their uses. The prediction analysis part provided an insightful analysis in which we learned the importance of feature selection and its effects. Additionally, it gave us an insight into how every feature plays an important role in the outcome and in the overall analysis.

One of the major points that was learned through this process was that the majority of time should be spent analyzing the data and looking at which feature affects the outcome the most. In order to obtain a model with high accuracy, it is important to understand which features play the greatest role. Additionally, collecting data for a long period of time and closely analyzing how each aspects interacts with each other could provide insightful analysis into the process of building a model.

FUTURE WORK

In the sentiment analysis part of our project, the models can be tweaked to include additional normalization method. In addition, the data in which was analyzed could be detailed to include categorization of tokens (such as emoticons). Below is a list of some future work that could improve the overall accuracy of the models in this report.

1. Collecting more data and manually tagging it will ensure that our vocabulary and training set will cover a larger variety of words. By utilizing this extensive tagging, we will have more features for each model.
2. In the predictive analysis part, a variety of different features can be added to the models, such as number of "hashtags", number of user mentions in a particular tweet, the context behind a tweet, whether the tweet posted has a hashtag that is top trending during the time the tweet was posted, or if the tweet is about some event that is generating great hype. Each of these may reveal some additional knowledge about the key features in predicting the retweet count.
3. As mentioned above, the data can be tokenized to include tagging of emoticon code to refer to the connotation of the original emoticons. This may provide some increased analysis of the sentiment analysis tagging.

REFERENCES

[1] https://dl.acm.org/citation.cfm?id=2398634

[2] https://dl.acm.org/citation.cfm?id=2457094

[3] http://ieeexplore.ieee.org/document/5590452/

[4] http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00049?class=ref+nowrap+pdf

[5] https://www.researchgate.net/publication/301408174_Twitter_sentiment_analysis

[6] https://github.com/cjhutto/vaderSentiment

[7] https://medium.com/@aneesha/quick-social-media-sentiment-analysis-with-vader-da44951e4116