**Sentiment Analysis FOR FITBIT USING TWITTER DATA**

*Vineet Barshikar – Priyanka Jha – Sivakumar Palanivel*

**Indiana University**

Bloomington

[vbarshik@indiana.edu](mailto:vbarshik@indiana.edu) - [prijha@umail.iu.edu](mailto:prijha@umail.iu.edu) - [sivannam@umail.iu.edu](mailto:sivannam@umail.iu.edu)

***Abstract:*** *We present you the sentiment analysis for the Fitbit product using the twitter data. This study uses three modern classifiers (Logistic Regression, Support Vector Machines and Naïve Bayes) to predict the sentiment of the tweets targeted on Fitbit products. We faced a stiff challenge by the data imbalance wherein the negative tweets were highly scarce. In spite of this, we were able to achieve state of the art performance, not only for the positive tweets but even for the negative ones. Although this study focusses only on the Fitbit related tweets but it could be easily implemented on any other product.*

***Keywords:*** *BeautifulSoup, CountVectorizer, StratifiedKFold, WordNetLemmatizer*

**1. Introduction**

Fitbit is currently one of the pioneers in fitness activity trackers. Fitbit launched their first activity tracker in 2009 and since then it has become a phenomenon in fitness world with more and more people buying and wearing it daily to maintain their health. Since then Fitbit has launched multiple products like Alta, Charge, Blaze, Surge, etc.

We wanted to use this high popularity of Fitbit products and understand their users’ opinions and perform a sentiment analysis on these. Social media postings are used by a lot of users to express opinions and reviews about various products. This publicly available data can be used to extract valuable insights using Data Analytics.

We used twitter API to collect the data source. This is publicly available data, which is obtained from the popular microblogging site called www.twitter.com. It holds microblog as 140 character tweets, which makes it a great platform to share thoughts and opinions. Therefore, Twitter data can efficiently be used to draw several insights including sentiment analysis.

Our system used the tweets collected from Twitter API (<https://apps.twitter.com)> and analyzed the sentiments of Fitbit users by using several keywords, time-stamps, and locations queries on these tweets. The design of the project is such that the functionality is easily scalable to sentiment analysis of other products.

1. **Related Work**

There has been significant advancement achieved in the sentiment analysis specially for the product reviews and customer feedback. Go et al. performed the sentiment analysis of twitter messages using distant supervised learning using machine learning algorithms like Naïve Bayes, Maximum Entropy and Support Vector Machines. However, the data studied upon was almost perfectly balanced with 50-50 positive and negative tweet words. Kolchanya et al. performed sentiment analysis of the twitter using lexicon based method and machine learning algorithms like Naïve Bayes and SVM. Wakade et al. presented the use of Weka data mining tools to perform sentiment analysis of tweets related to iPhone and Microsoft. Their experimental result showed that decision tree classifiers outperformed Naïve Bayes classifier.

As we see most of this work leveraged a balanced dataset with respectful proportion of positive and negative datasets. However, our dataset was highly imbalanced and due to the 7-day limitation of Twitter API, the dataset quantity was also less. But we overcame this by oversampling of the negative set from training data and hyper-tuning of our classifiers which resulted in good accuracy. We tried 3 classifiers and performed manual labelling of the dataset.

* 1. **Data Collection**

We collected about 2 months of tweets from Twitter API (<https://apps.twitter.com)>. Twitter API allows only last 7 days of tweet data hence we had to accumulate data per week for last 2 months. We filtered the tweets based on the tweet origination from USA only. Only those tweets were retrieved which had the below hashtags.

* #fitbit
* #fitbitfriends
* #charge2
* #fitbitsurge
* #fitbitblaze

While understanding the data, the following observations were made on the Twitter users’ geographical and gender information. With the exception of NY state, most of the tweets were originated from the southern states, maybe because of the warmer climate over there, many people were outdoors, and hence more active. The Fitbit users in other areas were experiencing cold weather in the last few months when the data for this project was collected.

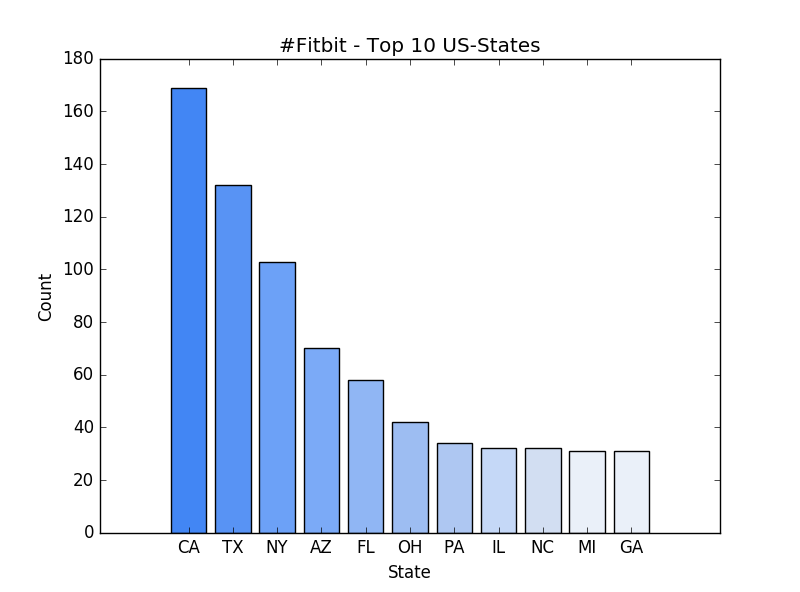


Figure 1: Tweets – Statewise distribution



Figure 2: Tweets – Heatmap

Twitter API does not maintain the gender data by itself, so we used *gender\_guesser* API to get the gender of users based on their names. Obviously, there are many ambiguous names which are unisex but this API considers higher probability gender.

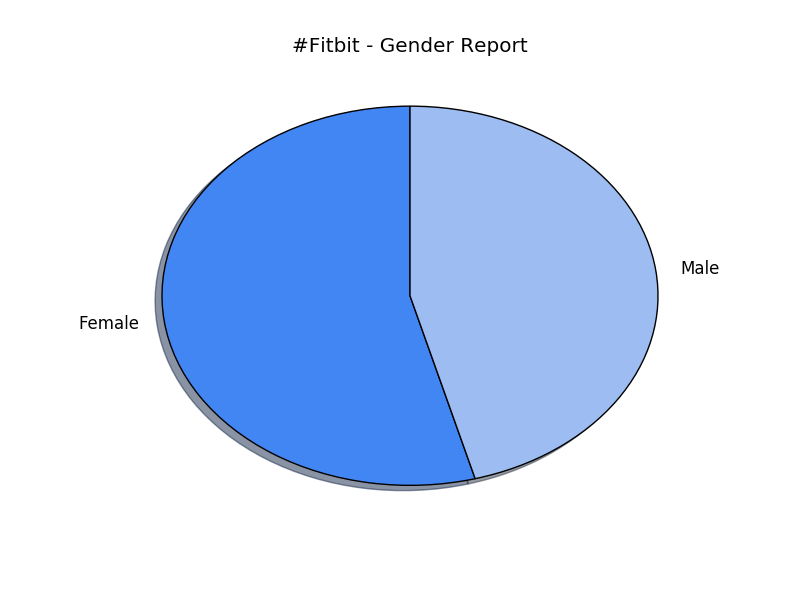


Figure 3: Tweets – Gender distribution

**1.2 Data Preprocessing**

We removed all the attributes of the tweet objects which were not required for this study. Also, the tweet object is in json format and hence required to be flattened. As part of the study we collected around 1365 tweets. Our next task was to label these tweets as positive or negative depending on their sentiment. Many times, the tweets’ sentiments are difficult to categorize as positive or negative and could be differently interpreted by different persons. Hence the labelling was done manually with all 3 members involved. After that we calculated the inter-rater reliability to gain confidence on the labelling accuracy. This rating was calculated per 2 member pairs and we selected that labelled set which had best rating. We observed almost 91% inter-rater reliability for this pair and was sufficient enough for us to proceed with the study.

The dataset was split into training and test set with 70-30 ratio. One major hurdle for us was the highly skewed nature of this dataset, with only as less as 6-7% of tweets qualified as negative sentimental. To overcome such kind of challenge we did the oversampling of the negative tweets from the training set. This reduced the data imbalance to some extent since now the ratio of positive to negative increased to approximately 3:2.

Both training and test set had to undergo a cleansing process and as part of it, following steps were performed.

* Remove Html – With the help of BeautifulSoup all the html characters were removed
* Remove Urls – All the words in the tweet text which were starting with “http” were removed since every url starts with “http”
* Remove non-alphabets – All the non-alphabets were removed the tweets using regular expression “[^a-zA-Z]”
* Lemmatize – As part of the lemmatization process, most of the inflectional forms of the words are reduced and brought to their most basic form using NLTK’s WordNetLemmatizer module
* Remove stopwords – A list of stopwords provided by NLTK was removed from the tweet text

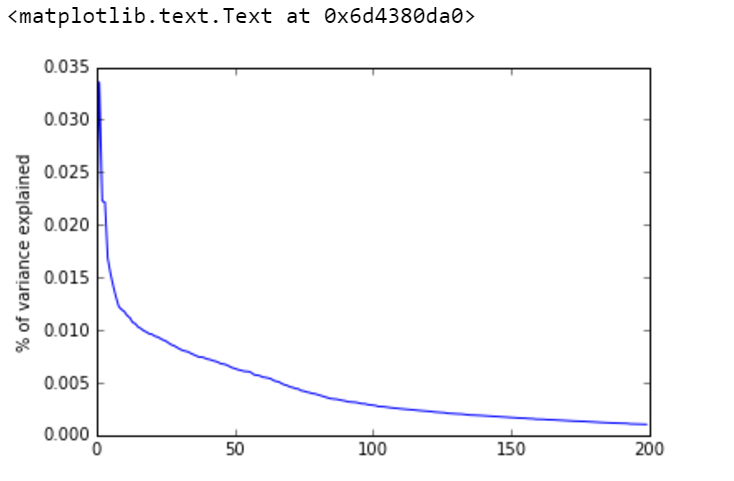
The html, Urls, and regular expression part of the above process was removed in the final version of the code as it was degrading the performance of the overall classifier.

* 1. **Classification**

We started out with several classifiers (the poster submitted earlier described 5 of those), and decided to tune three promising ones for our data. For this project, we focused on Logistic Regression, Naïve Bayes, and Support Vector Machines.

The text data needed to be vectorized, and we tried both CountVectorizer and TFI-DF, for our training set we found TFI-DF to give slightly better and mostly consistent results. For those reasons, we decided to use TfidfVectorizer. We also tried both unigram and bigram with these vectorizers, and observed that unigram performed better on this dataset.

As the dataset in hand was a sparse vector, we chose LSA over PCA (we did not want to densify the vector as that would be an overhead for Text Analytics). We plotted the explained variance ratio for 200 components and decided 85 components was ideal to use in this case.



We used a stratified kfold cross-validation with 5 folds for our training data. We then did hyper-parameter tuning for the classifiers. We observed that non-kernelized (Linear) SVM works better than Gaussian or Poly, which is not a surprise for text analytics on limited dataset. We further tested for parameter C and settled for C=0.3 as that worked best for the data.

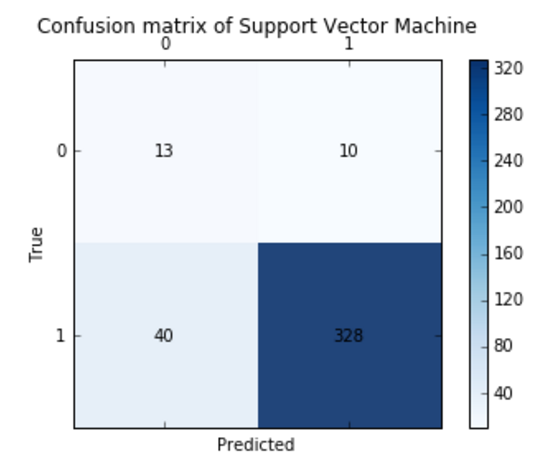
We then tuned Naïve Bayes and Logistic regression classifiers on the data and performed hyper-parameter tuning. Please refer to the code for the chosen parameters for these classifiers.

**Results**

The classification results from the three classifiers are as below:

\*\*\*\*\*\*\*\*Support Vector Machines \*\*\*\*\*

Accuracy = 0.872122762148

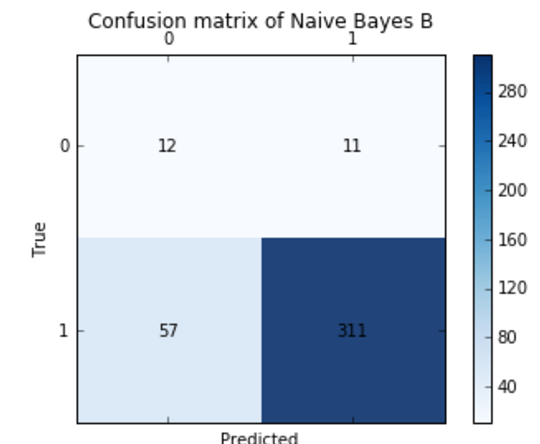


True Positive % match = 89.1304347826

True Negative % match = 56.5217391304

\*\*\*\*\*\*\*\*\*\*\*\* Naive Bayes B \*\*\*\*\*\*\*\*\*\*

Accuracy = 0.826086956522

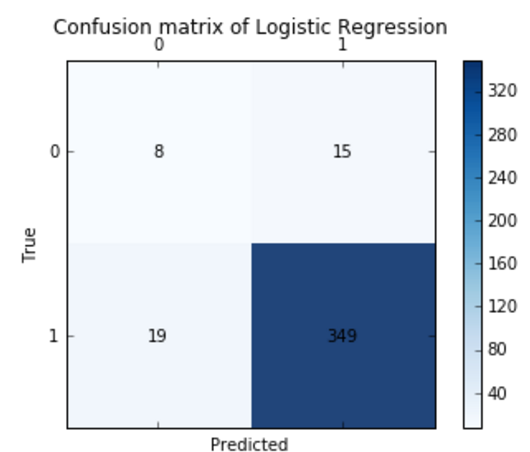


True Positive % match = 84.5108695652

True Negative % match = 52.1739130435

\*\*\*\*\*\*\*\*\*Logistic Regression \*\*\*\*\*\*\*\*

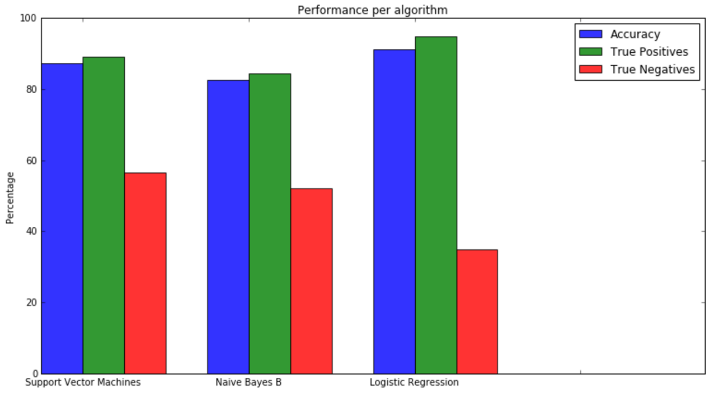
Accuracy = 0.913043478261



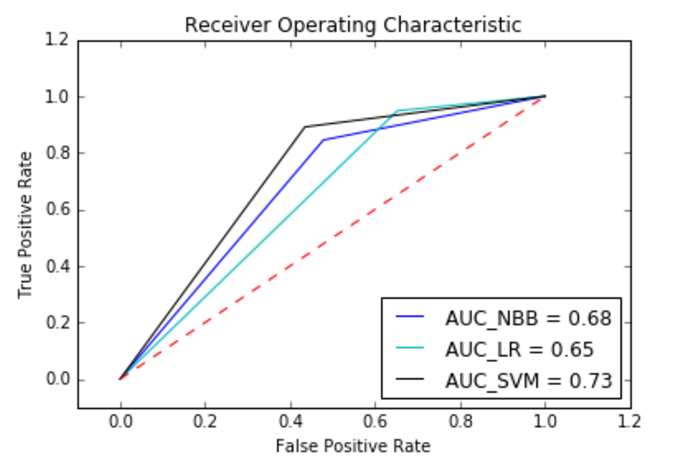
True Positive % match = 94.8369565217

True Negative % match = 34.7826086957

A graphial view of the results can be seen in form of histograms below:



Since the data is skewed, the accuracy alone is not good enough metric for performance. We calculated the area under the curve and plotted the ROC curve as a prformance measure –



**Discussion**

One thing to notice here is that we did not have a lot of data to work with due to limited time on the project and also the Twitter API restriction. As a team, we feel this classifier will do a lot better if it gets to train on more input data. Another enhancement can be to experiment with higher ngram. We did not go beyond the bigram analysis, so it would be interesting to see if higher ngram will help the classifier perform better.

This study is for those who want to extend and add on top of this. Whenever in future Fitbit launches any new product, we can use our classifiers to do a comparison of the sentiment analysis of the older product and this newly launched product. Also, a comparison can be done with other brands activity trackers. With small modifications and tweaks, we’re confident that this study can very well serve the same purpose for other products too. We just have to collect data with appropriate twitter hashtags.

**Conclusions**

In this paper, we presented you a comparison of 3 machine learning algorithms to perform sentiment classification of Fitbit tweets. Performance observed by Support Vector Machines trumped the Naïve Bayes and Logistic Regression algorithms. We performed stratified k-fold cross validation to avoid problems like over-fitting. From the ROC curve above, we can see that the Support Vector Machines gives the maximum area under the curve, and therefore is our chosen classifier for Fitbit sentiment analysis. Overall, we observed a good accuracy and AUC even though the data set was highly imbalanced.

**Team Member Contributions**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Vineet** | **Priyanka** | **Siva** |
| **Data Collection** | Y |  |  |
| **Data Validation** |  | Y | Y |
| **Data Cleanup** | Y |  | Y |
| **Data Annotation** | Y | Y | Y |
| **Inter-reliability code** |  | Y |  |
| **Data Preprocessing code** |  | Y | Y |
| **State-wise distribution bar-graph** | Y |  | Y |
| **Gender distribution pie-chart** |  | Y | Y |
| **Location Heat-map** | Y |  |  |
| **SVM classifier code** |  | Y | Y |
| **Naive Bayes code** | Y |  | Y |
| **Logistic Regression code** | Y | Y |  |
| **Confusion Matrices plot** | Y | Y |  |
| **Results histogram** | Y |  | Y |
| **ROC diagram** |  | Y |  |
| **Report writing** | Y | Y | Y |

**Acknowledgements**

We thank Prof. Vincent Malic – Indiana University Bloomington for his valuable feedback right from the proposal to the execution.

**References**

[1] https://pdfs.semanticscholar.org/ad8a/7f620a57478ff70045f97abc7aec9687ccbd.pdf

[2] https://pdfs.semanticscholar.org/ad8a/7f620a57478ff70045f97abc7aec9687ccbd.pdf

[3] https://dev.twitter.com/rest/public

[4] <https://github.com/vgm64/gmplot>

[5] <https://docs.python.org/2/library/index.html>

[6] <https://www.tutorialspoint.com/python/>

[7] <http://www.nltk.org/api/nltk.stem.html>

[8] <http://scikit-learn.org/stable/index.html>

[9] <http://www.numpy.org/>

[10] <https://matplotlib.org/2.0.0/index.html>

[11] http://www.fitbit.com/