INST737 – Fall 2015

Digging into Data

<u>Project Report</u>

Twitter Sentiment Prediction Using Sentiment Analysis and Classifiers on #Windows10

Submitted by:
Gnanasekaran, Rajesh Kumar – rgnanase@umd.edu
Pujari, Krishnesh – kpujari@umd.edu
Sharma, Anuj – asharm24@umd.edu

Table of Contents

Objective 3
About the Data 3
Feature Engineering & Sentiment Analysis 3
Exploring the Data 4
Multinomial Regression 7
Description 7
Important variables 7
Interpretation 7
Results 8
Decision Tree Analysis 9
Description 9
Important variables 9
Interpretation 10
Results 10
Random Forest Analysis 10
Description 10
Important variables 11
Results & Interpretation 11
Recommendations 12
Limitations 12
Individual Contribution 12

Objective

Twitter, as we all are aware, has gained a lot of attention like other social main stream media like Facebook, Google+, etc., and has an immense user base that comprises of imminent people from various businesses with officially handling their own account. Since the last decade or so, tweets have been increasing day-by-day with people pouring in their thoughts in the form of 140 characters. The tweets are a good source of raw text data which inherently contains people's true emotions. This project is an attempt to analyze the sentiments of tweet responses, for a recent worldwide Operating System Windows Upgrade by software giant Microsoft on 07/29/2015 of their product 'Windows10', using text mining techniques and try to predict the emotional polarity of the tweets and categorize them in to 'positive', 'negative' or 'neutral'. With the results from these analyses, the inferences would be studied to see if any recommendations would be provided to Microsoft or any other businesses on how to improve or sustain their business image on social media platforms like Twitter.

About the Data

- 'NodeXL' Excel template software was used to import the Twitter feed using Twitter Search Network option on hashtag "#Windows10" for dates between 07/28/2015 and 08/05/2015. The data was downloaded in a .csv file format with a total 9000+ records.
- The dataset was cleaned up performing missing value analysis, removing duplicate entries and by manually removing the tweets from languages other than English. After the cleaning, the dataset had 4646 Observations with 28 original factors. The dataset is attached under the name 'data.csv' in the project submission zip file.
- The original data did not have the desired predictor factor conducive for us to perform emotional polarity prediction. Hence, we performed feature engineering using Sentiment Analysis explained below, to first arrive at the predictor factor 'Polarity' and other techniques to arrive at few other factors like logarithmic values, tweet length, etc.
- This is the reason we would like to call our approach as semi-supervised, where in, we created the predictor variable as a first step using the best available input variables. Then, use that predictor variable in addition to the other original and derived factors to predict a new tweet's polarity.
- We had factors of types - Categorical and Continuous.

Feature Engineering & Sentiment Analysis

Sentiment Analysis to determine the predictor factor, "Emotional Polarity" which takes the values +ve, -ve or neutral was performed using text mining techniques resulting out of customized 'R' code which followed the below steps:

- This code is attached in the .rmd file beginning with the comment, "#Sentiment Analysis to calculate polarity of tweets".
- A method 'score.sentiment' was created which will take the input values as the raw text tweet, a list of positive word vocabulary, a list of negative word vocabulary. These +ve and –ve word lists are individually created .txt files that have words which are classified based on

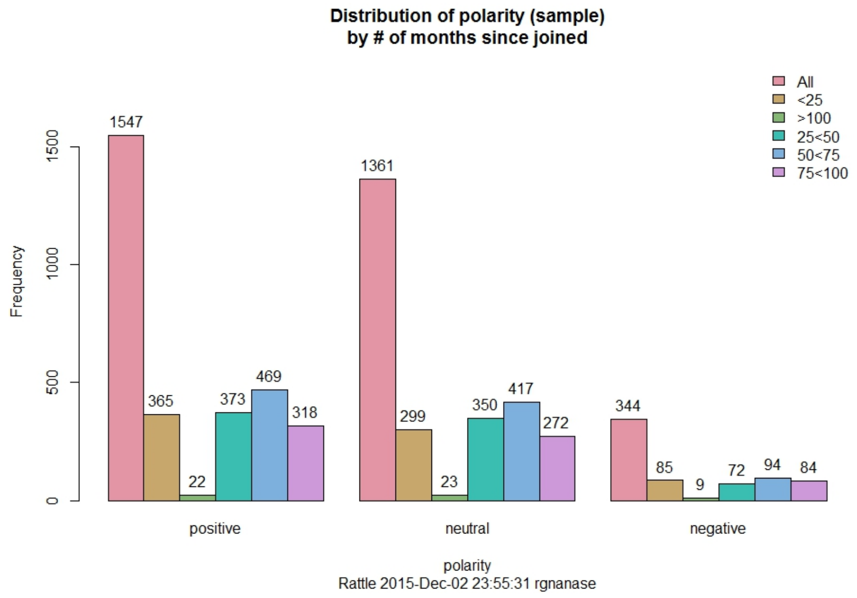
their sentiment they represent, for example: words like best, efficient, awesome depict +ve sentiment, whereas words like disappointed, negligence, pathetic imply –ve sentiment. These words are added without any delimiter in the .txt files, so the R code finds a matching set of letters forming a word without spaces from the original tweet in one of these files. These .txt files are attached along with .zip file. It is to be noted that these 2 files supplied in the .zip submission are to be placed in the working directory before running the .rmd file.

- The code was written to replace any punctuations marks like ‘,”, etc, to replace control characters and digits using the gsub() function with spaces.
- The tweet was then split into distinct words using the function strsplit() which disintegrates the original tweet into words delimited by spaces. This function’s output is a vector with a list of words which can be then matched with the +ve and –ve word lists mentioned above.
- If there is a match then the number of +ve matches and the number of –ve matches are identified. Then the final score calculation which is the difference of the two numbers, if score >=1 which means the +ve words are more in the tweet, hence the tweet would be classified as with polarity = positive, if the score <= -1, then it is classified with polarity = negative. If the score is between 1 and -1, it means that the tweet is possibly emoting a neutral polarity.
- We tested this model to verify if it worked by inputting an ad-hoc dataset with a random 100+ tweets by keying in words of both +ve and –ve polarity in each of them and ran it across. This particular model had a success ratio of 70% with 70 tweets out of 100 being classified under the right polarity.
- In addition to this field, we also created a few other derived factors from variables such as Tweets, Followed, Followers, and Favorites into logarithmic factors of the same so as to perform mathematical functions in an easier way. We used log (as.numeric()) function to arrive at these new factor values.
- We split the timestamp field into date and time fields. We took difference between the current date and the date of the tweet, also created the number of months and number of weeks since joining Twitter, calculated the length of the tweet by calculating the number of characters.

Exploring the Data

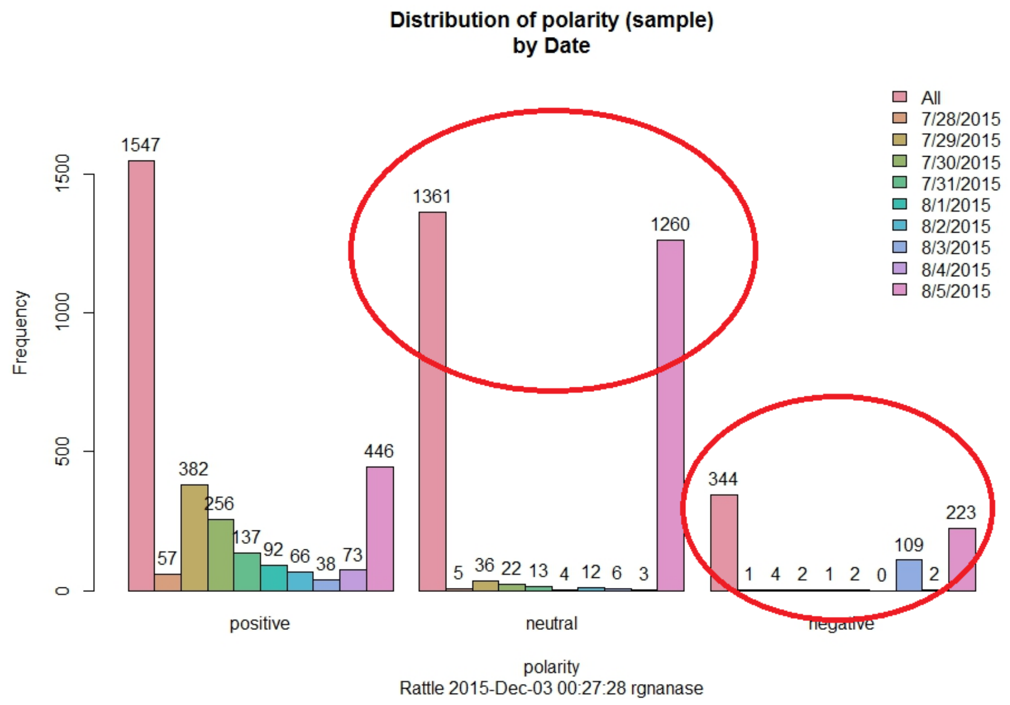
After creating the derived factors, we performed exploratory data analysis by creating histograms between the factors to identify any unique patterns. We found the below ones to be interesting:

- **Polarity vs # of months joined on Twitter:**



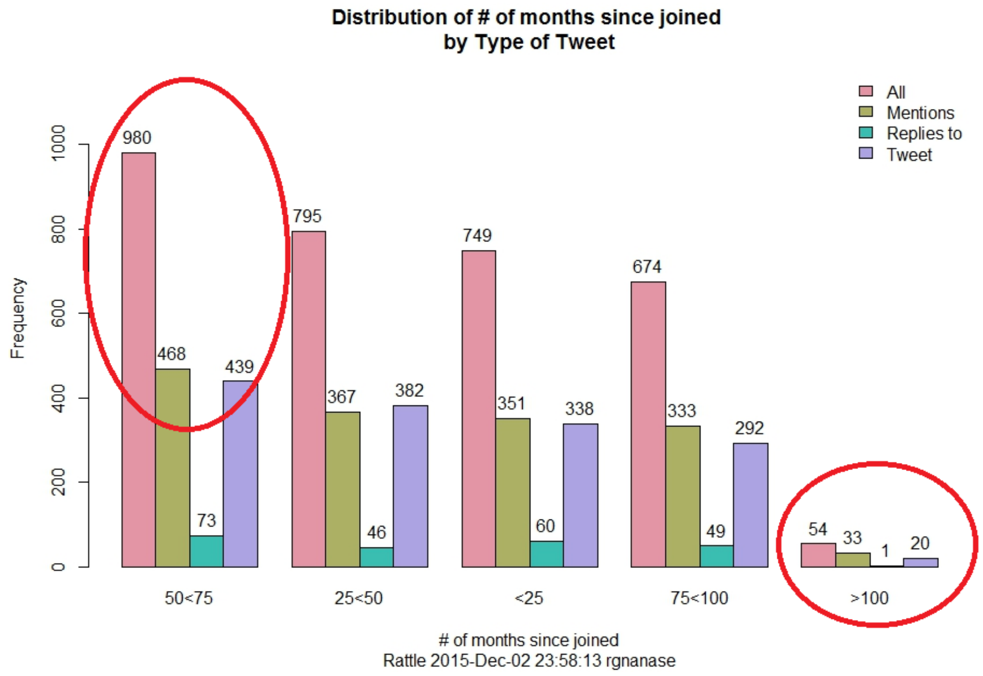
The above chart shows us that users from the dataset we used, those who joined twitter more than 100 months ago had very limited activity on tweeting, and the very few who did, had a positive or neutral emotion overall. Whereas those tweets from users who joined 50 to 75 months ago were the most active among the group.

- **Polarity vs Date of Tweets:**



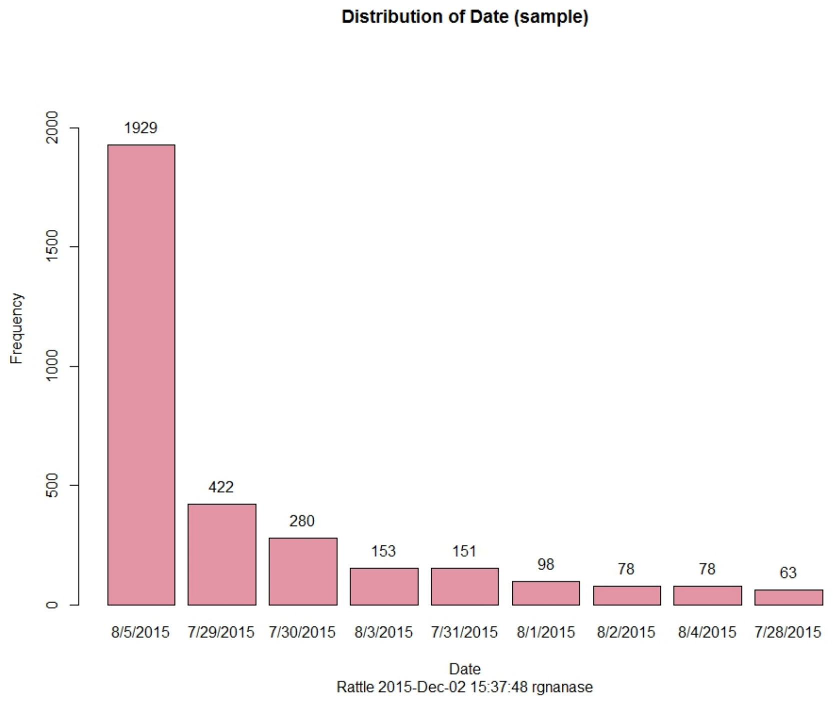
From the above graph, we can see that on or immediately after the upgrade date 07/29/2015, there was a positive vibe across the tweeting communities and almost nil negativity, but suddenly after a week of the upgrade, we see that the neutral and especially negative polarity starting to drastically increase, indicating that there may have been issues arising out of the upgraded software mal functioning on those users who tried it out.

- **Type of Tweet vs # of months since joining Twitter:**



Similar to the first pattern, we found that those users who joined twitter > 100 months ago did very less replies to the original and mostly were only mentioning the other twitter handles in their own tweet or creating a new tweet. Whereas the group with 50 to 75 months since joining tweeted more in total and more replies to other tweets.

- **Date distribution:**



By looking at the dataset entirely just on Date dimension, we find that there were increasing number of tweets around the upgrade release date which gradually decreases as the days pass, but there is a sudden spike on 08/05/2015 with –ve tweets on the rise, indicating that twitter communities emote out in the public more if to report problems and issues, and are mostly reserved about the +ve aspect of huge releases such as this event.

Now that we have the predictor variable, set of independent variables, exploratory data analysis ready, we created classification models using methods below, which will be used to predict the polarity of a new tweet which might have characteristics as one of the tweets from our training dataset. We split the original dataset into 70% of training data and 30% of testing data to work with the different models below:

Multinomial Regression

Description

We have used multinomial logistic regression model to predict the nominal dependent variable (in our case, polarity) given one or more independent variables. Multinomial logistic regression is an extension of binomial logistic regression to allow for the target variable with more than two categories (in our case, positive, neutral and negative). We are here trying to find out which variables are affecting the tweet polarity in any manner. The variables of importance which came out to be prominent after running this model are “Relationship”, “No. of followers”, “No. of tweets”, “No. of favorites” and “No. of weeks since joined twitter”.

Important variables

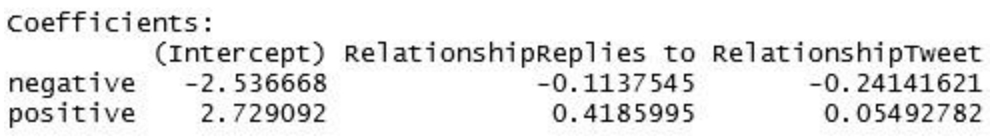
I. Relationship: It has three values,
   a) Replies to: If the user replies to other users putting #windows10 in the tweet
   b) Mentions: If the user mentions other users anywhere in his/her tweet.
   c) Tweet: If a user tweet using #windows10 in the tweet.
II. Followers: number of followers, that particular user has
III. Tweets: number of tweets, that particular has tweeted till date
IV. Favorites: the number of favorites that user has received on his/her tweets till date
V. number_weeks: No. of weeks since the user has joined twitter

Interpretation

The log of odds formula for multinomial logistic regression is:
Logit: $F(Y) = \log[Y/(1-Y)]$

The odds of happening of Y against the odds of it not happening i.e. $(Y/(1-Y))$ is nothing but exponential of the logistic function. We will try to explain it using one of our variables of importance,

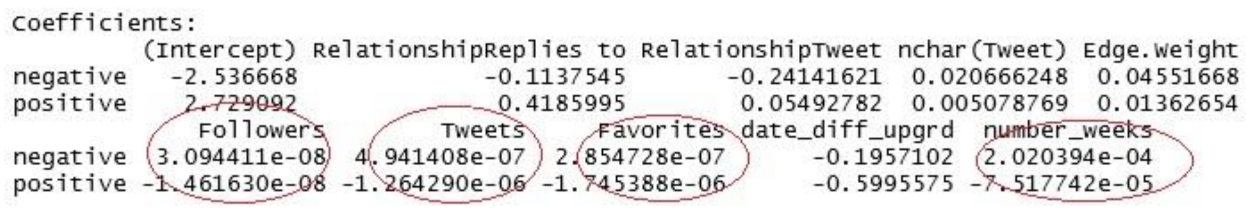
Relationship



```
Coefficients:
         (Intercept) RelationshipReplies to RelationshipTweet
negative   -2.536668             -0.1137545       -0.24141621
positive    2.729092              0.4185995        0.05492782
```

For a change in relationship from “Mentions” to “Replies to”, the change in the polarity of tweet from neutral to negative is exp(-0.1137545) = 0.89 whereas the change in the polarity of tweet from neutral to positive is exp(0.4185995) = 1.52. Hence the incoming new tweet is more likely to be a positive tweet if the relationship changes from “Mentions” to “Replies to”. Similarly, for a change in relationship from “Mentions” to “Tweet”, the change in the polarity of tweet from neutral to negative is exp(-0.24141621) = 0.78 whereas the change in the polarity of tweet from neutral to positive is exp(0.05492782) = 1.05. Hence again, the incoming new tweet is more likely to be a positive tweet if the relationship changes from “Mentions” to “Tweet”.

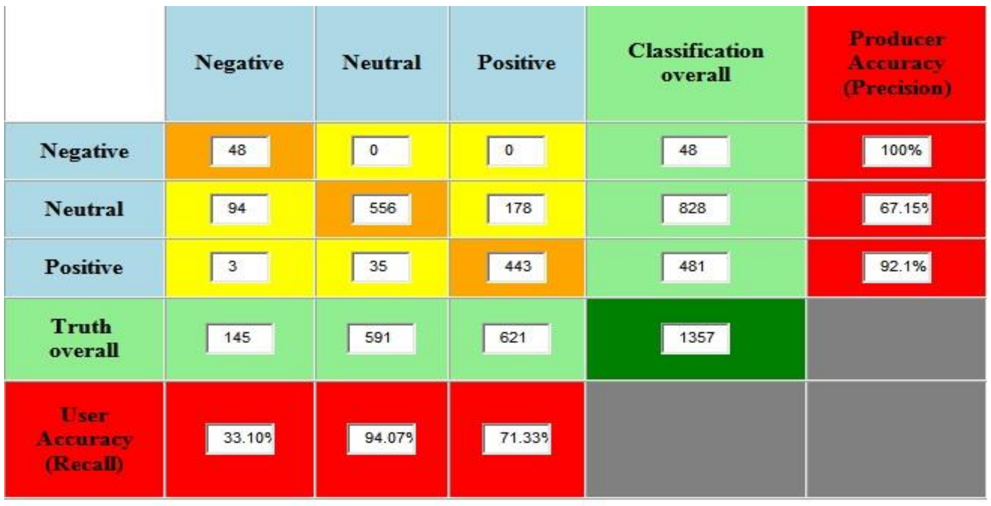
In the similar manner we can conclude from the below image that other variables of importance i.e. “No. of followers”, “No. of tweets”, “No. of favorites” and “No. of weeks since joined twitter” are also behaving in the same way. As, these predictor variables are bringing the maximum amount of variation in our target variable, the outcome model will be more accurate than the one having less important variables.



```
Coefficients:
         (Intercept) RelationshipReplies to RelationshipTweet nchar(Tweet) Edge.Weight
negative   -2.536668             -0.1137545       -0.24141621  0.020666248  0.04551668
positive    2.729092              0.4185995        0.05492782  0.005078769  0.01362654
             Followers        Tweets     Favorites date_diff_upgrd  number_weeks
negative  3.094411e-08  4.941408e-07  2.854728e-07      -0.1957102  2.020394e-04
positive -1.461630e-08 -1.264290e-06 -1.745388e-06      -0.5995575 -7.517742e-05
```

Results

From the below image we can see that, negative polarity has a better precision but a very poor recall. Hence, this model has helped us to categorize positive and neutral polarity better but performed badly for negative polarity. The overall accuracy from this model comes to be pretty good, around 77%.



	Negative	Neutral	Positive	Classification overall	Producer Accuracy (Precision)
Negative	48	0	0	48	100%
Neutral	94	556	178	828	67.15
Positive	3	35	443	481	92.1%
Truth overall	145	591	621	1357	
User Accuracy (Recall)	33.10	94.07	71.33		

Overall accuracy (OA): 77.15%

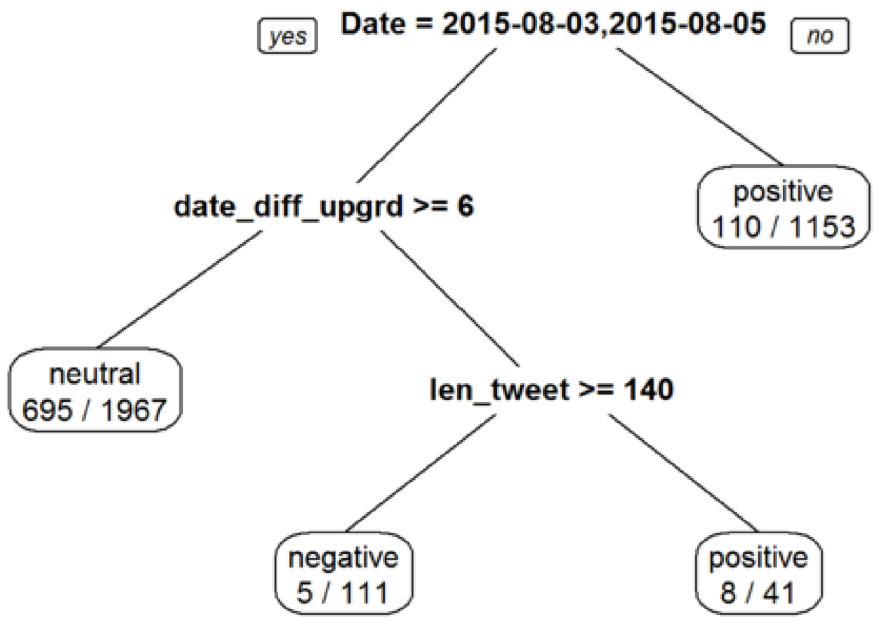
Decision Tree Analysis

Description

Apart from the Multinomial Regression model discussed above we also used Decision Tree Algorithm to predict the polarity of the tweet. Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g. Date) has two branches (e.g., positive, date_diff_upgrade). Leaf node (e.g., Positive or Negative or Neutral) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. In our project we used the rpart() function from the CART(Classification and Regression Tree) method to implement decision tree on the dataset. The rpart programs build classification or regression models of a very general structure using a two-stage procedure; the resulting models can be represented as binary trees. The tree is built by the following process: first the single variable is found which best splits the data into two groups. The data is separated, and then this process is applied separately to each sub-group, and so on recursively until the subgroups either reach a minimum size or until no improvement can be made. The resultant model is, with certainty, too complex, and the question arises as it does with all stepwise procedures of when to stop. The second stage of the procedure consists of using cross-validation to trim back the full tree.

Important variables

Please find below Decision tree which displays the important variables



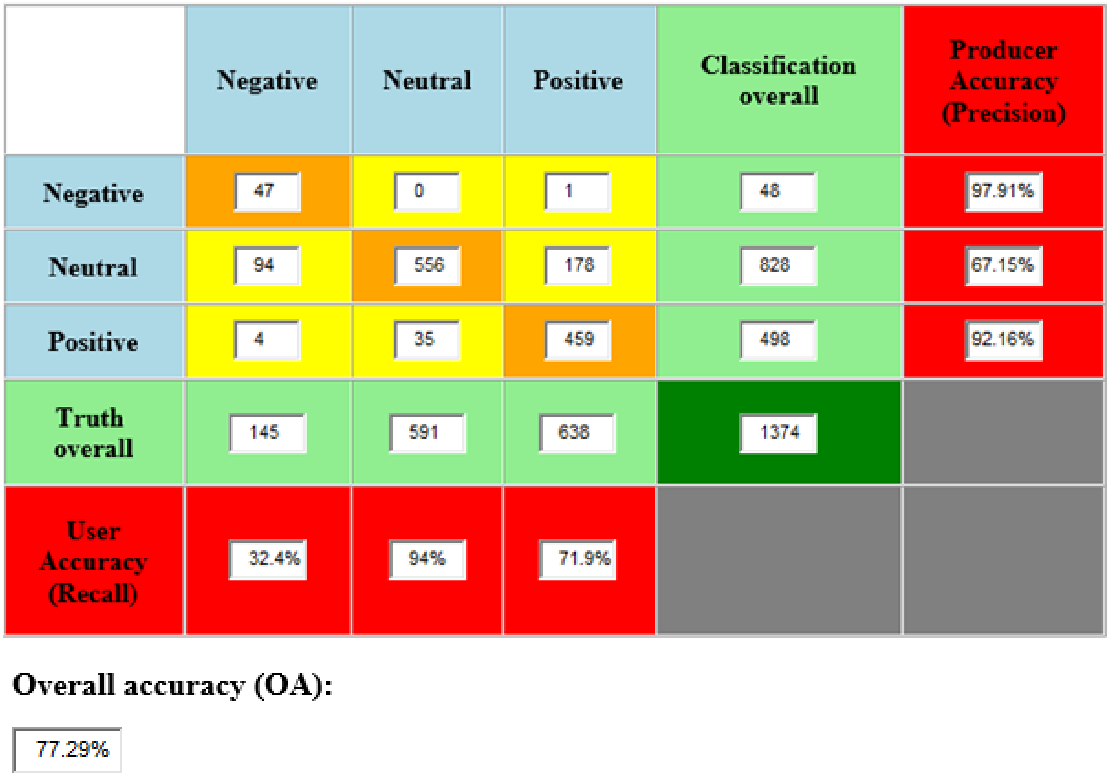
As you can see from the above decision tree important variables are – 'Date' of the tweet, 'number of days' since upgrade after which the tweet was made regarding the windows10 upgrade, and 'length of the tweet' is also an important factor to predict the polarity.

Interpretation

The decision tree generated to classify the tweet, and to identify the important variable helps us to decide whether the tweet is positive/negative/neutral. For instance, in the sequence of conditions (Date = 3rd August or 5th August) -> (date_diff_upgrade = '>=6') -> (len_tweet>=140) = tweet_sentiment (polarity) = negative, whereas in the sequence (Date != 3rd or 5th August) -> polarity = positive. This shows that a decision tree is a great tool for making decisions. Thus rpart() package performs a binary classification at each stage to classify the tweet into different polarities.

Results

From the below image we can see that negative polarity has a better precision but a very poor recall. Hence, this model has helped us to categorize positive and neutral polarity better but performed badly for negative polarity which is similar in nature when compared to Logistic regression. The overall accuracy is 77.29 % which is slightly better than multinomial regression.



	Negative	Neutral	Positive	Classification overall	Producer Accuracy (Precision)
Negative	47	0	1	48	97.91%
Neutral	94	556	178	828	67.15%
Positive	4	35	459	498	92.16%
Truth overall	145	591	638	1374	
User Accuracy (Recall)	32.4%	94%	71.9%		

Overall accuracy (OA):

77.29%

Random Forest Analysis

Description

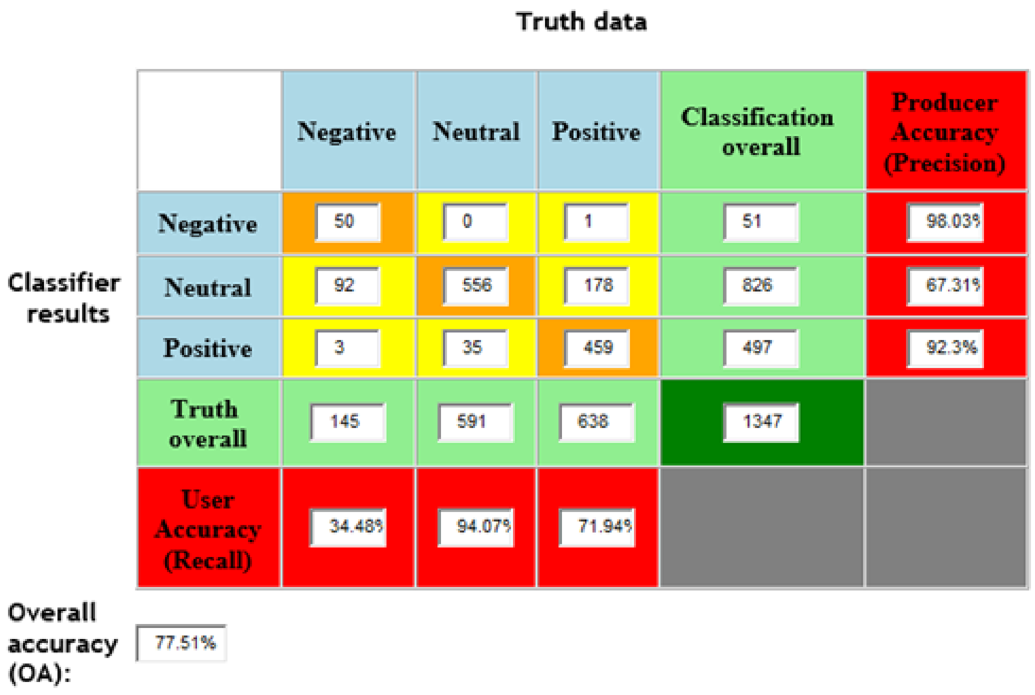
After observing the results from Multinomial Logistic and Decision trees model, we chose to implement Random Forest classification, which is an extension to the decision trees. The main advantage of this method is its high rate of accuracy since it is an ensemble method. Random Forest estimates the output through Bootstrap Aggregation method. It takes into account a number of iterations of the decision trees performed and gives the best-aggregated class as the output. We modeled our dataset to perform n=501 iterations and to identify the important variable and classify the target variable i.e. polarity with higher accuracy. Random forest corrects decision trees habit of overfitting the training dataset. Each node is split using the best split in a small random sample of available variables and every tree is constructed at random and is independent of other trees. Therefore, adding trees to the forest does not cause a problem of overfitting.

Important variables

As you can see from the graphs above, we see similar pattern in the variable of importance as seen in the decision tree model. Independent variables like Date, date_diff_upgrade, len_tweet, number of weeks since the user has joined twitter, etc. are the influential factors to classify polarity of tweets.

Results & Interpretation

Even for Random Forest, the positive and negative polarities ae classified better compared to negative polarity. However, the negative polarity is slightly better classified compared to previous models. Moreover, the overall accuracy of the model has slightly increased compared to Multinomial and Decision Trees which is around 77.51%.



Truth data

Classifier results		Negative	Neutral	Positive	Classification overall	Producer Accuracy (Precision)
	Negative	50	0	1	51	98.03%
	Neutral	92	556	178	826	67.31%
	Positive	3	35	459	497	92.3%
	Truth overall	145	591	638	1347	
	User Accuracy (Recall)	34.48%	94.07%	71.94%		

Overall accuracy (OA): 77.51%

Recommendations

- Overall from the analysis methods' results above, we find that the negative sentiment starts to arise after a week of the upgrade release date, so we suggest companies like Microsoft, during events like these to not stop their product branding and also to look for the negative sentiments to provide quick turnaround resolution as possible. They can use this platform as an efficient feedback mechanism to identify the problems with their product or service and provide appropriate bug fixes or enhancements at the earliest possible.
- One other finding was that Microsoft should try to come up with some algorithm which will bring in those twitter users who have joined well before the others to participate and provide their feedback on the product to get a uniform reply from the entire customer base.
- For users with more followers or followed, it is found that they tweet with a –ve emotion most of the time, companies should try to reach out to these users and address their issues which will in turn be an influencer to their followers in order to get a +ve feedback or non-negative feedback.

Limitations

There are a number of limitations we face with this project.

- The dataset is mostly not a considerable huge number of sample as NodeXL software restricts the amount of twitter data we can collect. If we would have had more data about this particular hashtag, the results may have been different, particularly we could have been establish even more accuracy regarding the –ve tweets sentiment.
- The whole project was driven on the assumption that only English language tweets are considered. As Twitter is available worldwide and Microsoft has customers worldwide, the effect of other language tweets may have had a telling effect on the results.
- The text mining sentiment analysis did not take into account the different types of emotion like cynicism, sarcasm, etc, where +ve emotion words are intentionally mentioned with an opposite underlying meaning. This will be considered for future scope of the project
- We were not able to take advantage of the geo-spatial coordinates of the tweet location where it got generated as most of the records had a value - n/a.

Individual Contribution

- Gnanasekaran, Rajesh Kumar – Helped with the exploratory data analysis
- Pujari, Krishnesh – Helped with cleaning of the data, Decision Tree & Random Forest methods
- Sharma, Anuj – Helped with the Multinomial Logistic Regression
- All – Worked on the sentiment analysis, ppt and report