**Name: Rebecca Karunakaran**

**NLP Final Project -** SemEval shared task Twitter data

**Experiment 1 Baseline: I decided to check the baseline accuracy with no preprocessing and limiting the number of tweets to 5000.**

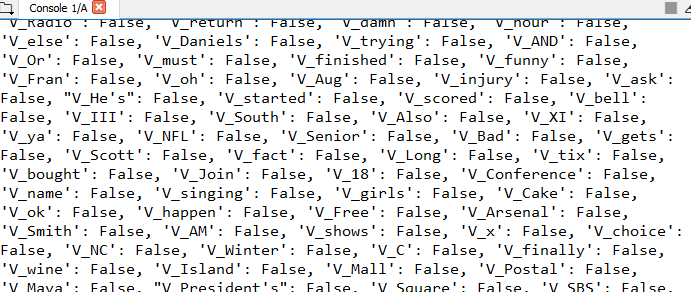
Screen shot shows a sample of the first 10 tweets -



The program tokenizes the tweets and condenses the tweets to 3 – 'pos', 'neg', 'neu' The number of words after condensing the label dropped from 5000 to 4282.

The screen shot shows the most common keyword was “the”, followed by some punctuations.

The first experiment was to use a feature that takes the 1000 most frequently occurring words and check to see the presence or absence of the word in document of the tweets –

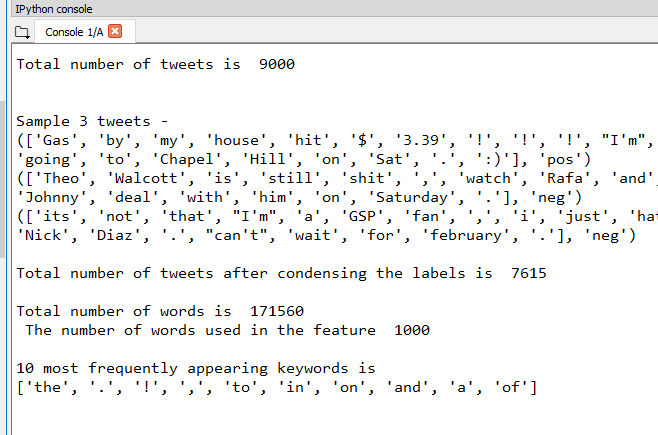


Naïve Bayes algorithm was trained using holdout method of 90% data for training and 10% for testing.

Base Line Accuracy with No Preprocessing was 57%

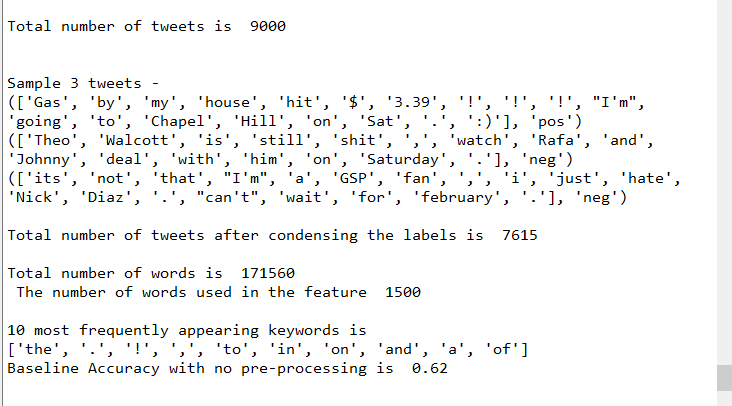
**Experiment 2 Vocabulary Size : The next experiment was to see if increasing the number of number of tweets from 5000 to 9000 would improve the Accuracy**

Screen shot below shows that there is a definite improvement in the Baseline Accuracy from 57% to 63% which shows that **the larger the data** the more **data points which help** Machine Learning Algorithms in predicting the accuracy!!

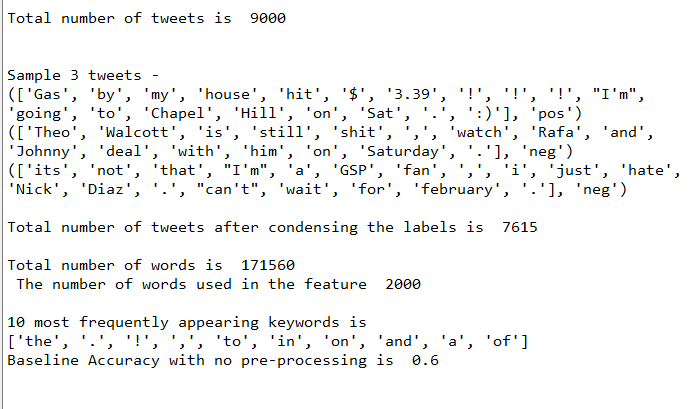


**Experiment 3 Vocabulary Size: The next experiment was to keep the number of tweets at 9000 and increase the number of words used in the feature generation from 1000 to 1500 words.**

The Accuracy decreased slightly from 63 to 62%



With 2000 words for feature generation the accuracy has dropped further to 60%

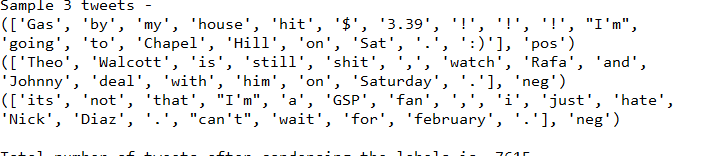


This experiment indicates that increasing the number of words in the feature does not necessarily improve the accuracy.

**Experiment 4 Pre-Processing : The next experiment was to keep the number of tweets at 9000 and the number of words used in the feature generation at 1000 and use some Pre-processing to see if that improves accuracy.**

The twitter data has many punctuations, so the first step was to remove the punctuations and remove the stop words from the provided stop word text - stopwords\_twitter.txt

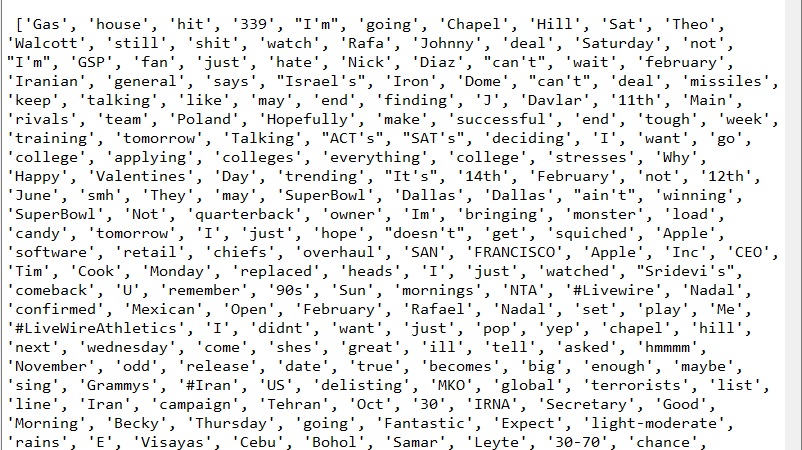
Screenshot shows the punctuations –



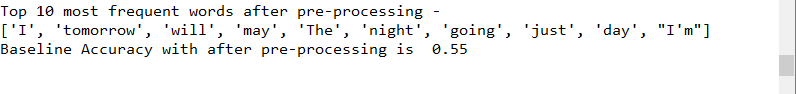
After preprocessing from punctuation and stop-words the text looks much better but there are twitter handles (@text) , URLs and Emojis (😊) that need to be removed -

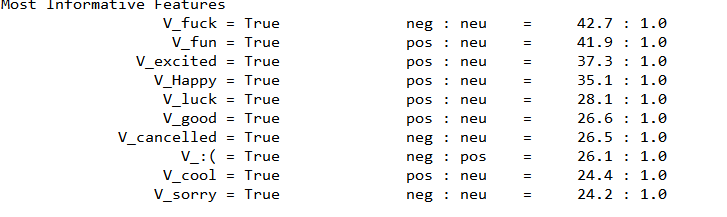


Screen shot after removing twitter handles (@text) , URLs and Emojis (😊) -



Notice that the top 10 most frequent words now don’t have punctuations or stop-words.

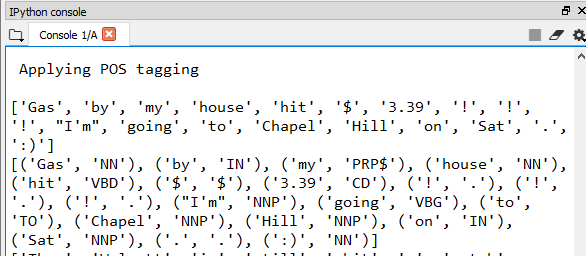




The Accuracy after Preprocessing however has dropped to 55% which shows that the algorithm was actually using some of the words removed during pre-processing may actually have be useful to train the algorithm!!

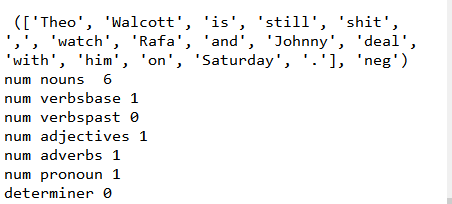
**Experiment 5 POS Tagging: The next experiment is to see if we use can use POS tagging to build features.**

The first step is to apply the POS tagging using the Stanford POS tagger which is the default in NLTK

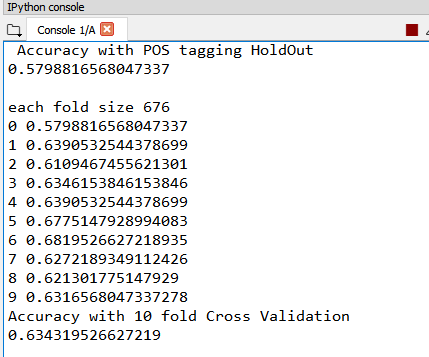


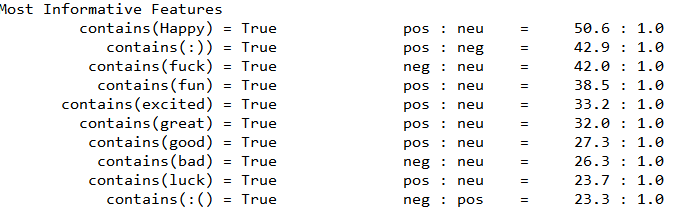
The POS features function builds feature based on the count of the POS. The POS features used where – Nouns, VerbsBase, VerbsPast, VerbsPastPresent, VerbsPresentParticiple, Verb Non 3rd Persons, Verb 3rd Person, Adjectives, Adverbs, Preposition, Determiner, To, Interjection, Cardinal, Coordinating Conjunction and Pronouns.

The following screen shot shows the features on one of the sentences in the twitter text -



Accuracy using POS tagging as feature with HOLD-OUT method with 90% of data for training and 10% for testing was 57.89%. Cross Validation is however a better way to calculate the Accuracy because it splits the data in k folds and uses one-fold for testing and the rest for training and repeats k times to calc the average accuracy. The 10-fold Accuracy with POS was 63.43%.

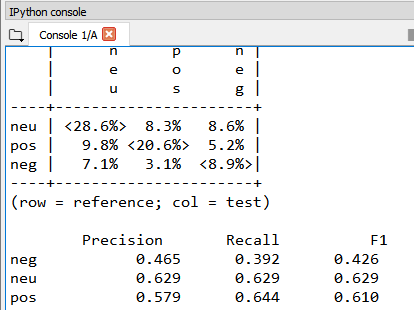




Screen shot below shows the Confusion Matrix which shows that the “neutral” tweets seems have been predicted correctly (28%) compared to the “positive” (20%) or “negative” (8.9%) tweets.

Precision tell us among all the positive/negative/neutral predictions, how may were correct. We see that “neutral” has the highest precision which means the algorithm was able to predict the “neutral” tweets the best.

Recall tells us among all the positive/negative/neutral examples, how many were correctly predicted. Again, we see that the “neutral” tweets have the highest Recall.



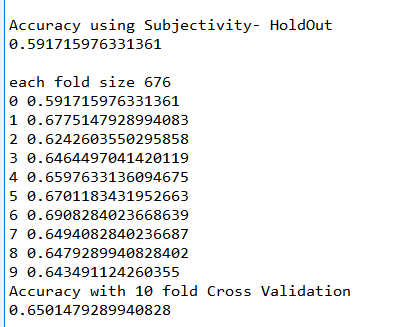
**Experiment 6 Subjectivity: The next experiment is to see if we use can use Sentiment Lexicon -Subjectivity to build features.**

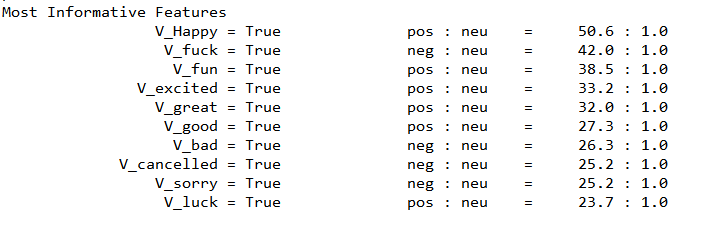
The Lexicon used to build the feature is the **MPQA subjectivity lexicon** which is a lexicon containing a list of words that are categorized as - weakly or strongly positive, negative or neutral in subjectivity.

A feature extraction functions was used to calculate the word features and has three features ‘positive count’ , ‘negative count’ and “neutral count” which is the count of all the positive , negative and neutral subjectivity words

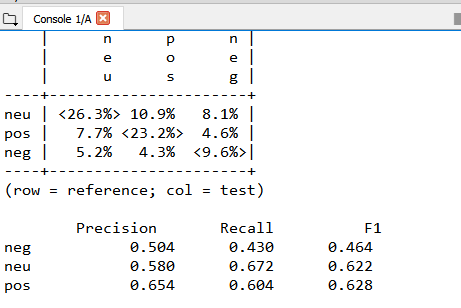
Accuracy using Subjectivity Lexicon as feature with HOLD-OUT method with 90% of data for training and 10% for testing was 59.17%.

The 10-fold Accuracy with Subjectivity Lexicon was 65.01 % which is a slight improvement from the POS 10 fold Accuracy of 63.43%. which indicates that subjectivity features are more useful compared to just POS features for twitter data.





Screen shot below shows the Confusion Matrix which shows that the “neutral” tweets seems have been predicted correctly (26%) compared to the “positive” (23%) or “negative” (9.6%) tweets.



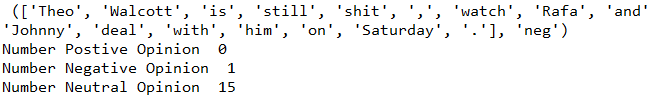
We see that “positive” has the highest precision which means the algorithm was able to predict the “positive” tweets the best.

Recall tells us among all the positive/negative/neutral examples, how many were correctly predicted. Again, we see that the “neutral” tweets have the highest Recall.

**Experiment 9 Opinion: Using Bing Liu Opinion Lexicon for feature extraction**

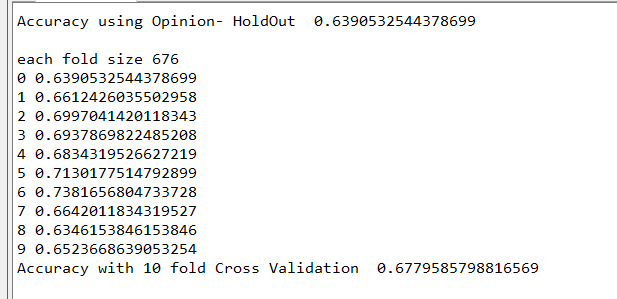
The Opinion Lexicon used has about 2000 positive and 4800 negative opinion words.

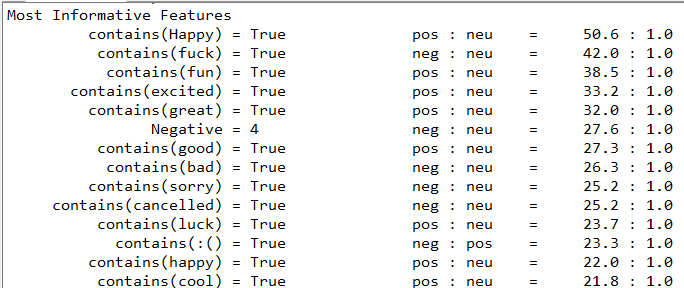
A feature extraction functions was used to calculate the word features and a feature count of ‘positive’ , ‘negative’ and “neutral” word was calculated for each tweet.



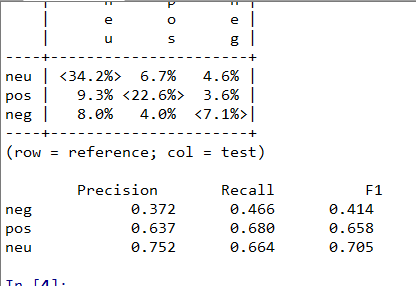
Accuracy using Opinion Lexicon as feature with HOLD-OUT method with 90% of data for training and 10% for testing was 63.9%.

The 10-fold Accuracy with Opinion Lexicon was 67.79 % which is a big improvement from the Subjectivity 10-fold Accuracy of 65%. which indicates that opinion features are useful for twitter data.





Screen shot below shows the Confusion Matrix which shows that the “neutral” tweets seems have been predicted correctly (34%) compared to the “positive” (22%) or “negative” (7%) tweets.



We see that “neutral” has the highest precision which means the algorithm was able to predict the “neutral” tweets the best.

Recall tells us among all the positive/negative/neutral examples, how many were correctly predicted. Again, we see that the “positive” tweets have the highest Recall.

**Experiment 10 : Combining Opinion sentiment with Bigrams**

The final experiment was to combine Opinion sentiments with Bigrams to see if the combination would improve the accuracy. The Bigram feature list was generated using the **PMI measure** (top 500 words)

Surprisingly the Accuracy with HOLD-OUT method with 90% of data for training and 10% for testing was only 63.90 % .

The 10-fold Accuracy with Opinion Lexicon was 67.95% which is very close to the accuracy of 67.79% with just opinion sentiments which indicated that the addition of **bigram has not helped the algorithm improve the prediction of twitter data.**

