Twitter Sentimental Analysis

Sami Ali Choudhry
Masters of Computer Science
University of Windsor
Windsor, Canada
Email: choud116@uwindsor.ca

Balsaharan Singh Bedi
Masters of Computer Science
University of Windsor
Windsor, Canada
Email: bedib@uwindsor.ca

***Abstract*—Social media is one of the biggest platform through which every individual can express his/her thought, thus this paper majorly focuses upon the reviews provided by the people on twitter within a limit of 140 characters for a particular airlines and after the processing of each and every review it basically classifies them into positive, negative or neutral form of it and thus we use different algorithms like SVM, Logistic Regression, Multinomial Naive Bayes, Random Forest Classifier and thus obtain the accuracy by adjusting and tuning of different parameters like gamma, alpha and etc and select the best algorithm after comparison of each.**

***Index Terms*—NLP(Natural Language Processing), dataset, Multinomial Naive Bayes, Classification.**

I. Introduction

A. Twitter

Twitter is a popular real time social media platform that allows users to share short information known as tweets based on their point of view which are limited to 140 characters. Thus, everyday active users can portrait their own perception for ongoing issues and their thoughts. Twitter is an ideal platform for the extraction of general public opinion on specific issues and to review anything and know choice of majority of population on any particular type of product or any upcoming idea. A collection of tweets is used as the primary corpus for sentiment analysis, which refers to the use of opinion mining or natural language processing[1].

Twitter, with 500 million users and million messages per day, has quickly became a valuable asset for organizations to invigilate their reputation and brands by extracting and analysing the sentiment of the tweets by the public about their products, services market and even about competitors [2]. A variety of opinion texts in the form of tweets, reviews, blogs or any discussion groups and forums are available to study and understand each of them as per the demand of every product, thus making the world wide web the fastest, most accessible and convenient medium for sentiment analysis.

B. Twitter Sentiment Analysis

The reviews made by users can be found in the comments or tweet which show peoples review and thereby can be used as data to provide useful indicators for many different purposes and for categorizing them into different categories by using classification. Also, for twitter sentiment analysis a sentiment can be categorized into three groups, which are negative, positive and neutral words. Sentiment analysis is a natural language processing technique to quantify an expressed opinion or sentiment within a selection of tweets [3]. Sentiment analysis refers to the general method to extract polarity and subjectivity from semantic orientation which refers to the strength of words and polarity text or phrases [4]. Using these methods we thus perform classification of text after data cleaning and generate a desirable result.

C. Objective

The project will use Natural Language Processing (NLP) to solve the above problem statement as the major domain. Natural language Processing is concerned with the interaction between the computers and human.Furthermore, it deals with writing a computer program to process and analyse huge amount of natural language data. The natural language processing started in 1950's by Alan Turing as a criterion of intelligence. In 1980's the natural language processing was done using sets of handwritten rules. But there was a revolution in 1980's when machine learning algorithms were used for natural language processing. At this time machine learning algorithms such as decision trees were used. But there was limitation in this approach as it was based on if else rules similar to the handwritten rules. Many researches took place in this field to use the unsupervised and semi supervised learning to solve natural language problem. These algorithms can learn from data that can lead to desired results. Now machine learning algorithms are used to do natural language processing. There are advantages of using machine learning algorithms.

- Machine learning algorithms focus on the most common cases in the language.
- When input is erroneous it is hard to detect such errors by handwritten rules but learning algorithms can figure out such errors easily.
- The performance of machine learning algorithms can be improved by training it on more input data. This cannot be done while working with handwritten rules.

Thus we will make use of Natural Language Processing for sentimental analysis by various stages of processes such as stop word removal, punctuation removal, lemmatization, building feature set, text classification and various other methods which are discussed in the methodologies.

II. Problem Statement

We have a dataset that contains more than 10000 twitter reviews about an airline company, our task is to predict the sentiment of each tweet. This means that we will predict whether each twitter review about this airline company is:

- Positive
- Negative
- Neutral

The dataset will contain more than 10 columns and more than 10000 rows. But in this dataset not all columns are required to achieve considerable result hence only important values are to be taken into account. The job involves dropping of the columns that are not required and cleaning each review using Natural Language Processing and correct use of the algorithm in order to achieve better result so that the text classification algorithm can give a good accuracy when classification is done.

III. Project Overview

Natural Language Processing used for text classification firstly all the stop words as well as punctuations are removed from the given text document which itself are not useful for classification purpose. Then, the meaningful words are extracted from the given text, this is one of the most important initializing process in order to clean the data. This task is performed by the word tokenizer. Now there are some words which can be written differently but they mean the same. Stemmer is used to convert such words to their root form. After all the text is cleaned the next task is to build the feature set which is later used for classifying them into different types that is positive, negative, neutral. This task is done by the count vectorizer which converts a collection of text document to a matrix of token counts it produces sparse representation of counts which is the basic output. The figure below gives outline of the basic functioning of the strategic implementation of sentimental analysis

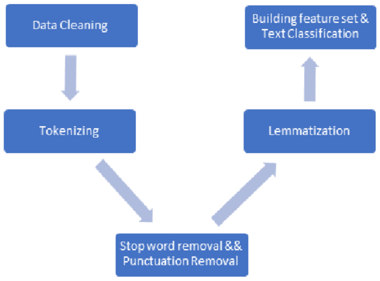
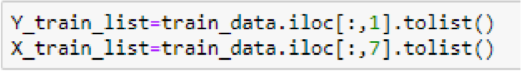

Data Cleaning
Building feature set & Text Classification
Tokenizing
Lemmatization
Stop word removal && Punctuation Removal

Fig. 1. Project Overview.

A. Data Cleaning

The process of dataset cleaning thereby involves removal of unwanted NaN values and also the columns which serve of no good contribution to the dataset in order to produce a useful output and help in improving the performance by improving the accuracy of classifier and also the classification process. Also the raw data present is high dimensional and totally unstructured so it has to be cleaned and moulded as per the requirements of processing in order to obtain an operable dataset. After this the portion of the data is divided into training and testing set based upon the percentage of the data needed for better results we divide it into two parts and obtain this by trial and error method.

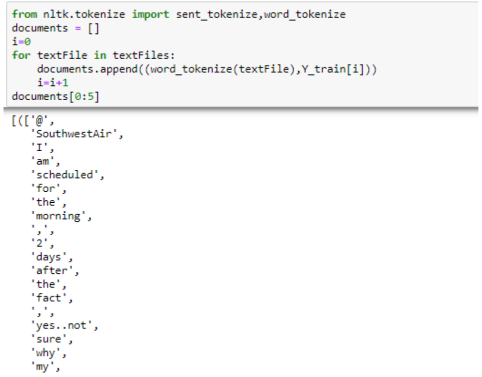


```
Y_train_list=train_data.iloc[:,1].tolist()
X_train_list=train_data.iloc[:,7].tolist()
```

Fig. 2. Train and Test.

B. Tokenizing

This is a process which is used to extract only the important words which are required for further processing, also in tokenizing we split the string or texts into an initial list of tokens, which is like a sub-part of a sentence. It is one of the most important methods which help in simplifying the content prior to the steps which will help in classification and give results.



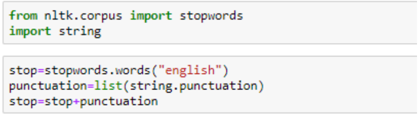
```
from nltk.tokenize import sent_tokenize,word_tokenize
documents = []
i=0
for textFile in textFiles:
    documents.append((word_tokenize(textFile),Y_train[i]))
    i=i+1
documents[0:5]
```

```
[(['@',
   'SouthwestAir',
   'I',
   'am',
   'scheduled',
   'for',
   'the',
   'morning',
   ',',
   '2',
   'days',
   'after',
   'the',
   'fact',
   ',',
   'yes..not',
   'sure',
   'why',
   'my',
```

Fig. 3. Tokenizing.

C. Stop word and punctuation removal

This part of the workflow removes stop words like a,an,the,this, also a part of it is used to remove the punctuation's which does not help in determining whether a particular tweet specifically belongs to positive, negative or neutral.



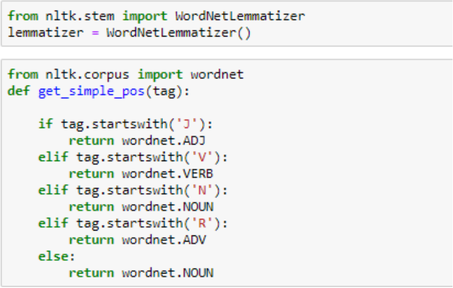
```
from nltk.corpus import stopwords
import string
```

```
stop=stopwords.words("english")
punctuation=list(string.punctuation)
stop=stop+punctuation
```

Fig. 4. Stop words removal.

D. Lemmatization

Lemmatization is used to convert the words which are having the similar meaning into the same type of words, for example if we consider ’good’ and ’better’ they have similar meaning and serve the similar purpose in order to describe the meaning or intention of a tweet in the given data and thus both the words ’good’ and ’better’ are converted to ’good’ which means converting a particular word in their root form.



```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

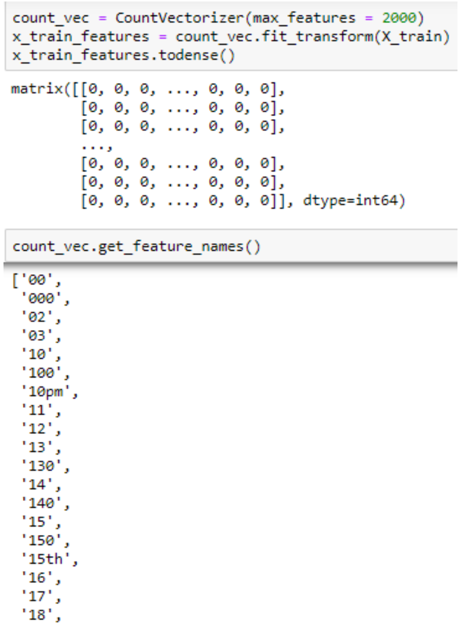
```
from nltk.corpus import wordnet
def get_simple_pos(tag):

    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('V'):
        return wordnet.VERB
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN
```

Fig. 5. Lemmatization.

E. Count Vectorizer

It is used to convert the text data into a format that can be understood by the text classification algorithm. The data is converted to a matrix where columns will represent the important words and each row is the count of that word in the document.



```
count_vec = CountVectorizer(max_features = 2000)
x_train_features = count_vec.fit_transform(X_train)
x_train_features.todense()
```

```
matrix([[0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        ...,
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
count_vec.get_feature_names()
```

```
['00',
 '000',
 '02',
 '03',
 '10',
 '100',
 '10pm',
 '11',
 '12',
 '13',
 '130',
 '14',
 '140',
 '15',
 '150',
 '15th',
 '16',
 '17',
 '18',
```

Fig. 6. Count Vectorizer.

Also the table below summarizes the information of methodology used in the software

TABLE I
SUMMARY TABLE

Parameters	Functioning
Data Cleaning	Removal of NaN values
Tokenizing	Extract only important words
Stop words and punctuation removal	Optimization
Lemmatization	Convert words to root form
Count vectorizer	Convert text data into machine language

IV. Classification

A. Naive Bayes Classifier

In machine learning, Naive Bayes Classifier uses Bayes’ theorem with strong (naive) independence assumptions between the features which were word frequencies. Naive Bayes classifiers are highly accessible, requires number of parameters which are linear in the number of variables (features/predictors) in the learning problem. Training of Maximum-likelihood can be used for evaluation of a closed form expression which considers linear time, rather than expensive approximate iteration that is used for different types of classifiers.[5] Naive Bayes is a classifier technique used for building classifiers: Models assigns class labels to instances, represented feature values as vectors, in which class labels are extracted from some finite set. For example, ”the fruit which is round may be considered as an orange if its colour is orange, round, and it is about 4” in radius. Naive Bayes classifier independently considers each of these features to find the probability whether this fruit is an orange, regardless of any possible relationships between the features like roundness, colour and diameter[5].

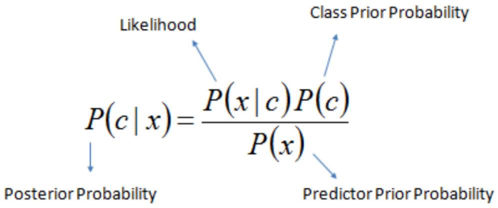

Likelihood
Class Prior Probability
$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability

Fig. 7. Bayes Theorem.

- This is the probability of event c given that event x has already occured.
- P(c) is the priori of c or the probability of event c.
- P(x—c) is the posteriori probability of x.

This algorithm makes very strong assumptions such as :

1) The first assumption is that all the features are considered to be independent of each other. For instance, if temperature and pressure are two features in dataset and the aim is to determine the temperature of that day. Then, these two features have no relation with each other.
2) The second assumption is that, every feature is given equal importance. Considering the above example to

determine the temperature of that day, temperature and pressure are important and none of them can be dropped.

There are 3 distributions of Naive Bayes namely:

- **Gaussian**
- **Multinomial**
- **Bernoulli**

The Multinomial Naive Bayes is a variant of the Naive Bayes classifier that is used for text classification. This classification algorithm works upon the probability of a given word in a document. The formulae can be given as **P(word=w/document=d1) = Count(w in d1)/Count(total words in d1).**

Here we are trying to find out the probability of a word w in a given document d1. This can be calculated by counting the word w in the document d1 divided by the total number of words in the document d1.

Multinomial Naive Bayes gives the best accuracy when there are a lot of features and the data of these features is in the form of frequencies. This means, that we must use **Multinomial Naive Bayes** in case of text classification.

B. Support Vector Machine

A support vector machine works by mapping the datapoints onto the N-Dimensions and creating a hyperplane that distinctly classifies the datapoints. The objective is to figure out a plane that maximizes the margin, i.e. the maximum distance between data points of both classes. This ensures that the future datapoints will be classified correctly. The points closest to the hyperplane are known as support vectors. Figure below shows how support vectors influence the orientation of the hyperplane. Additionally, selecting the optimal kernel function and tuning the parameters produces results that have high accuracy.

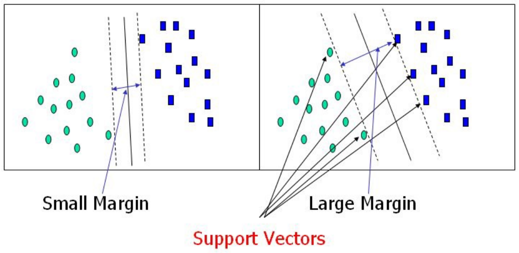


Fig. 8. Support Vector Machine.

The RBF Kernel uses the formula to map the data onto infinite dimensions and produces a hyperplane that separates the data. The RBF kernel is the most powerful and is utilized when the dataset is complex and overlapping. The formula for the radial basis function is given below:

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2} \right)$$

Fig. 9. Radial Basis Function.

C. Random Forest Classifier

A decision tree is used for classification in which at each step a division takes place by using certain parameter.

- Node: a point at which decision is taken
- Edge: a connection between nodes
- Leaf Nodes: the final node that gives the decision

It consists of a Node, where a value is tested for splitting, edges, used to connect nodes with each other and leaf nodes that are the terminating nodes of a decision tree. We use classification trees for the purpose of classification. This tree is built using a technique called recursive partitioning. This is a process in which a node takes a decision of splitting into further branches based upon the feature value.

Decision trees are generally of two types, classification and regression trees. Here, our aim is to classify the reviews so we use the classification decision trees. In this type of decision tree, we firstly begin with the root node. This root node is split on the basis of feature values. Information gain, a decision function, is calculated for each feature. The feature that shows the maximum information gain for that node is then selected. This iterative process is repeated for every node that is split till we reach the node that is pure. This pure node is called the leaf node. This provides us with the final decision taken by the decision tree. But the major problem with this approach is that the depth of the tree can become too much. This can lead to overfitting and our test results will be highly poor. So, in order to avoid this problem, we fix the depth.

This process is also divide and conquer because the data is spit into smaller parts in each step. When we use the decision tree as a classifier, we pick up a feature from the dataset showing the maximum information gain to represent the root node. We keep on picking the features from the dataset iteratively based upon the information gain till a pure node is created in the tree. A combination of such trees is called a random forest. The major advantage of this technique is that it is very fast in classification.

A Random Forest classifier consists of many decision tree. Whenever a prediction needs to be made, this input is given to all the decision trees that are a part of that random forest. Each decision tree given out a prediction. The majority output class becomes our final output.

The major idea behind this classifier is correlation. According to this classifier, low correlation is the major concern when it comes to giving great results. As we can see, that in a random forest, each tree is highly uncorrelated to each other. Each tree protects itself from errors and this ends up in giving

good results.
But, overfitting can take place due to which the testing accuracy will be affected. **Moreover, when the number of features are very large, decision trees can become very complex. They are difficult to understand and the results are inaccurate due to increased complexity**. This affects the classification accuracy. Due to this reason, Random Forest classifier does not perform well on our twitter sentiment analysis dataset

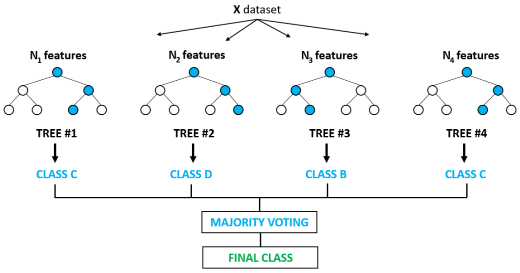


Fig. 10. Random Forest.[6]

D. Logistic Regression

Logistic Regression uses the logistic function for classification. This function is also called as the sigmoid function which is in the form of an S shape. This curve can take any real value and provides the result between 0 and 1. The sigmoid function can be given by :

f(n)=1 / (1 + power(e,-i))

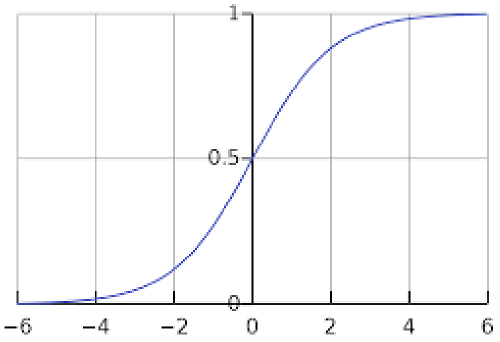


Fig. 11. Sigmoidal Function.

Here, e represents the exponential function and i represents the input for which we want the output. This machine learning algorithm uses a equation just as used in linear regression. To find out the output say, y, a combination of input values and weighs is used. In case of linear regression, we get a continuous output value but in case of logistic regression, we get an output class. The equation for logistic regression is given by,

y = power(e,(a0 + a1*x)) / (1 + power(e(a0 + a1*x)))

In the above equation, y is the output class, a0 is the intercept on axis, and a1 is the weight for the input value x. The logistic regression algorithm learns the intercept and weight during the training phase. These values are determined by the algorithm using the MLE(Maximum Likelihood Estimation). Even though very strong assumptions are made by MLE, but still it is the one of the most used methods for finding out the parameters. The idea behind Maximum Likelihood is that the value of the weights is determined such that the error in probability is minimized. In other words, it can be said that this algorithms is a numerical optimization algorithm. But in order for the logistic regression algorithm to perform well, some key points need to considered. Mostly, this algorithm is used for binary classification problem, but in our case for twitter sentiment analysis, we have three output classes, 'positive', 'negative' and 'neutral'. So, for multiclass classification, logistic regression uses one vs rest approach. In actual implementation we can set the multiclass parameter to 'ovr'. Also, logistic regression is a linear algorithm that assumes linear relationship between the input and the output classes. So, from the entire training data only the data that best shows this linear relationship must be chosen. Furthermore, overfitting can take place if there is a high correlation between the features. So, feature selection must be done carefully to improve the testing accuracy. Moreover, there is still one more problem associated with such feature selection. The weights that are determined by the maximum likelihood estimation might fail to converge.

V. Related Work

General method for sentiment analysis is done on three levels:

- Document Level: To analyse the whole document and then classify whether the document positive or negative sentiment[12].
- Sentence level: It is related to find sentiment polarity(positivity or negativity) from short sentences. Sentence level is merely close to subjectivity classification based upon the input provided.
- Entity /Aspect Level sentiment analysis performs augmented analysis. The aim is to find sentiment on entities or aspects. eg: consider a statement "My Iphone x phone's picture quality is good but its phone storage capacity is low". Iphone camera and the quality of display has positive sentiment but phone's storage memory sentiment is negative[12].

Pak et al.[7]made a twitter corpus by which automatically collects number of tweets with the help of Twitter API and annotating them automatically using the emotions.Using this corpus, they created a sentiment classifier functioning up-on the multinomial Naive Bayes classifier using N-gram and POS-tags as features. In this method there are chances of error as these emotions of tweets are labelled solely upon the positivity and negativity of emotions. Also the training set will be limited as it will consist only those tweets which has certain types of emotions.

Wu et al.[8]developed an influence probability model for twitter sentimental analysis. If @username is found in the particular tweet, it is changes the action and it contributes to changing probability for processing of the given tweet. Thus, any tweet that starts with @username is a considered as a retweet that shows an influenced action and it contributes to influenced probability. Thus it is observed that there is a strong correlation between these probabilities.

Few of softwares are made by some researches to identify the public opinion about movies,TV series, news etc from the twitter posts. V.M. Kiran et al.[9]thus made use of the information from other public databases like IMDB and Blippr, which are considered as one of the important channels which provide ratings for the a particular show after proper modifications to aid twitter sentiment analysis in movie.

Xia et al.[10]makes use of ensemble framework as a sentiment classifier. many classification techniques and feature sets are combined together to form the ensemble framework. In this, there are two types of feature sets and three base classifiers to form the ensemble framework. Two types of feature sets include using Part-of-Speech and Word-relations which are important information for classification. Naive Bayes, Maximum Entropy and Support Vector Machines are selected as base classifiers. They applied different ensemble methods like Fixed combination, Weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy.

Barbosa et al.[11]created a double step analysis method to classify tweets. A noisy training set was used in order to reduce the additional work of labelling the classifiers. Initially tweets were diveded into subjective or objective types of tweets. Then, subjective tweets are classified as positive and negative tweets. In pronunciation based word clustering, words having similar pronunciation are clustered and assigned common tokens. They also used for text processing techniques like assigning similar tokens for numbers, html links, user identifiers, and target organization names for normalization. After the process of normalization, they used probabilistic models to identify positive and negative lexicons. They performed classification using the BoosTexter classifier with these polarity lexicons as features and obtained a reduced error rate[11].

VI. Challenges

A. Grammatical Errors

Grammatical errors may affect the processing of a particular type of a tweet, however filtering and understanding the closest possible word to a specified may be a big task. There are different types of natural language processing softwares which may carry out filtering and better understanding of the errors but still with the developed technology we are not yet able to carry out this process with great accuracy.

B. Sarcasm

Sarcasm can be used in order to indirectly portrait a particular object as good, bad or neutral however it may also be used to upgrade or degrade a product in order to showcase them as false-positive, true-positive, false-negative, true-negative. Thus, the goal to understand the following sentences and classify them as positive, negative and neutral can be achieved in the future software development due to limited availability of technology now.

C. Spam Identifiers

There are number of companies which use spamming of comments in order to degrade the ratings of their competitive companies based upon their domain of products, in order to develop a better overall review from their customers. Thus, understanding the pattern of these spammers and classifying them as original or fake can be considered as one of the task to increase the authentication in sentimental analysis.

D. Denoising

The quantity of data present on twitter is by far way too much and also every human has a different method of framing and constructing their methods of speech. Thus the unstructured tweets along with different formatting and unnecessary words tend to impact upon the classification and techniques used for data cleaning. Thus a better performing denoising software and some unique methods can be used in order to improve the results.

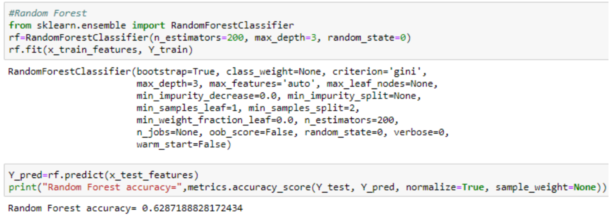
VII. Results

On implementing the number of algorithms such as Multinomial Naive Bayes, Random Forest Classifier, Logistic Regression and SVM using the linear SVC, we obtain the following results shown below.

A. Random Forest Classifier

Implementing Random Forest Classifier on the given dataset was simple, as it gives correct answers for relatively small number of samples but with the given dataset the accuracy comes up to approximately **62%**, which is very less due to large number of samples and has a major limitation which cannot be improved even after parameter tuning. It is also like a black box approach in certain sentimental analysis cases, as there is not much of a control on how the model performs on the particular dataset and will give sufficient results for small samples, but will not be up to the mark for large dataset.
Also in our case we make use of make use of parameters set such as

- **n-estimators** which is set to 200
- **max-depth** which is set to 3
- **random-state** which is set to 0



```
#Random Forest
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=200, max_depth=3, random_state=0)
rf.fit(x_train_features, Y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=3, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)
```

```
Y_pred=rf.predict(x_test_features)
print("Random Forest accuracy=",metrics.accuracy_score(Y_test, Y_pred, normalize=True, sample_weight=None))
```

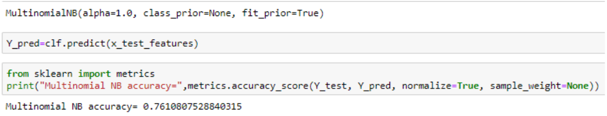
```
Random Forest accuracy= 0.6287188828172434
```

Fig. 12. Random Forest Classifier.

B. Multinomial Naive Bayes

Implementing Multinomial Naive Bayes on the given dataset which is classified as multi class problem for text classification it gives a better result with respect to random forest classifier because of the independence rule involved in the formula, and also the high scalability with the given dataset helps us to improve the accuracy of the software upto **76%** also it is well suited for continuous or discrete kind of data which can be used for sentimental analysis and can handle the missing values well, however it still does not give the satisfactory results even after tuning the values and parameters in the code. Also in our case we make use of the parameters as follows:

- **alpha** which is set to 1.
- **class-prior** which is set to None.
- **fit-prior** which is set to True



```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

```
Y_pred=clf.predict(x_test_features)
```

```
from sklearn import metrics
print("Multinomial NB accuracy=",metrics.accuracy_score(Y_test, Y_pred, normalize=True, sample_weight=None))
```

```
Multinomial NB accuracy= 0.7610807528840315
```

Fig. 13. Multinomial Naive Bayes.

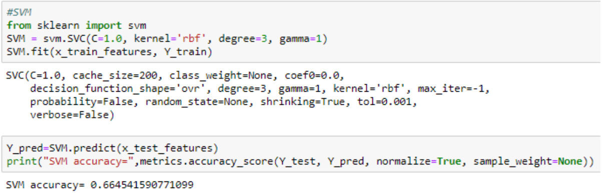
C. Support Vector Machine

Implementing Support Vector Machine on the given dataset gives us a result of about **66%** when we use a RBF kernel which is considered as a part of parameter tuning, the RBF kernel is shown in the above figure and by using it the accuracy doesnt comes up to the required results however if we use linear kernel we get **75%** as accuracy.
Also the parameters which are used for Support Vector Machines for RBF kernel are as follows:

- **C** which is set to 1.
- **Kernel** which is set to RBF.
- **degree** which is set to 3.
- **gamma** which is set to 1.

Also the parameters which are used for Support Vector Machines for Linear kernel are as follows:

- **C** which is set to 1.
- **Kernel** which is set to RBF.
- **random-state** which is set to None.
- **loss** which is set to square-hinge.



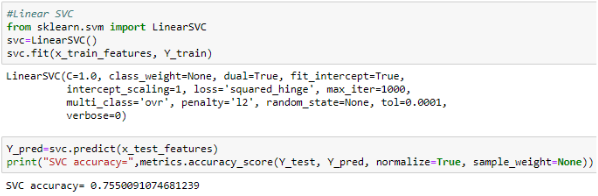
```
#SVM
from sklearn import svm
SVM = svm.SVC(C=1.0, kernel='rbf', degree=3, gamma=1)
SVM.fit(x_train_features, Y_train)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf', max_iter=-1,
    probability=False, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

```
Y_pred=SVM.predict(x_test_features)
print("SVM accuracy=",metrics.accuracy_score(Y_test, Y_pred, normalize=True, sample_weight=None))
```

```
SVM accuracy= 0.664541590771099
```

Fig. 14. SVM with RBF Kernel.



```
#Linear SVC
from sklearn.svm import LinearSVC
svc=LinearSVC()
svc.fit(x_train_features, Y_train)
```

```
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,
          multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
          verbose=0)
```

```
Y_pred=svc.predict(x_test_features)
print("SVC accuracy=",metrics.accuracy_score(Y_test, Y_pred, normalize=True, sample_weight=None))
```

```
SVC accuracy= 0.7550091074681239
```

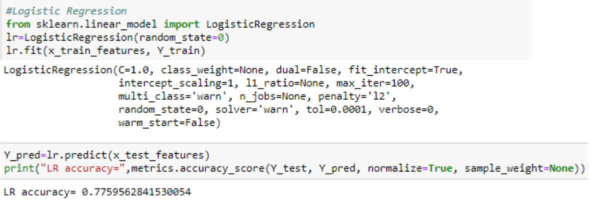
Fig. 15. SVM with Linear Kernel.

D. Logistic Regression

Implementing Logistic Regression upon the given dataset was simpler and easier as it doesn't require much of computational resources and covers up the limitations of other algorithms also the input features doesn't require to be scaled as twitter has large amount of raw and unstructured data and also it does not require much of the samples to train the data. Hence by implementing logistic regression upon the mentioned dataset gives up an accuracy of about **77.5%** which is by far the best result achieved from all the other algorithms implemented.

Also the parameters which are used for Logistic Regression are given below:

- **random state** which is set to 0.
- **fit-intercept** which is set to True.
- **C** which is set to 1.
- **class-weight** which is set to None.



```
#Logistic Regression
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(random_state=0)
lr.fit(x_train_features, Y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=0, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
Y_pred=lr.predict(x_test_features)
print("LR accuracy=",metrics.accuracy_score(Y_test, Y_pred, normalize=True, sample_weight=None))
```

```
LR accuracy= 0.7759562841530054
```

Fig. 16. Logistic Regression.

TABLE II
COMPARISON TABLE

Algorithms	**Accuracy**
Random Forest Classifier	62%
Multinomial Naive Bayes	76%
Support Vector Machine(RBF)	66%
Support Vector Machine(Linear)	75%
Logistic Regression	77.5%

Conclusion

On combining the results of all the 4 algorithms we successfully managed to classify the tweets into positive,negative and neutral tweets for a particular airline review, however we select the best algorithm which according to our implementation and dataset comes through the **Logistic Regression** and also due to limitation of available libraries and api's there is still scope for further improvement in the project which includes covering all the challenges such as **sarcasm, grammatical errors** and etc with 100% accuracy and also in future we try to overcome **different language barriers**, along with use of emoticons also **detect spam reviews** and research upon improving accuracy with new libraries.

Appendix

The software works on different algorithms and makes use of the best algorithm which gives highest accuracy based upon different parameters of tuning which is obtained by trial and error, there are altogether 4 algorithms implemented out of which the coding was divided equally between the two members and also the paper writing was equally contributed between both members, however the initial introductions and project overview along with comparison of results was written by 1st author, where as classification along with part of references was completed by the 2nd author.

References

[1] M.Rambocas, and J. Gama, "Marketing Research:The Role of Sentiment Analysis". The 5th SNA-KDD Workshop'11. University of Porto, 2013.

[2] H. Saif, Y.He, and H. Alani, "Semantic Sentiment Analysis of Twitter," Proceeding of the Workshop on Information Extraction and Entity Analytics on Social Media Data. United Kingdom: Knowledge Media Institute, 2011.

[3] T. Carpenter, and T. Way, "Tracking Sentiment Analysis through Twitter,". ACM computer survey. Villanova:VillanovaUniversity, 2010

[4] M.Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, " Lexicon Based Methods for Sentiment Analysis," Association for Computational Linguistics, 2011.

[5] Trupthi, M., Pabboju, S., Narasimha, G. (2017). Sentiment Analysis on Twitter Using Streaming API. 2017 IEEE 7th International Advance Computing Conference (IACC).

[6] "Random Forest Classifier - Machine Learning," Global Software Support, 09-Oct-2019. [Online]. Available: https://www.globalsoftwaresupport.com/random-forest-classifier

[7] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in Proceedings of LREC, vol. 2010, 2010.

[8] Y. Wu and F. Ren, "Learning sentimental influence in twitter," in Future Computer Sciences and Application (ICFCSA), 2011 International Conference on, pp. 119–122, IEEE, 2011.

[9] V. M. K. Peddinti and P. Chintalapoodi, "Domain adaptation in sentiment analysis of twitter," in Analyzing Microtext Workshop, AAAI, 2011.

[10] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152, 2011.

[11] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 36–44, Association for Computational Linguistics, 2010.

[12] Po-Wei Liang, Bi-Ru Dai, "Opinion Mining on Social Media-Data",IEEE 14th International Conference on Mobile Data Management,Milan, Italy, June 3- 6, 2013, pp 91-96, ISBN: 978-1-494673-6068-5,http://doi.ieeecomputersociety.org/10.1109/MDM.2013.