TWITTER SENTIMENT ANALYSIS

CA1 PROJECT REPORT

By

Aditya Raj	**Pankaj Raina**
11912185	**11910326**
Section – KM056	**Section – KM056**
Roll No. - 42	**Roll No. – 37**

Department of Intelligent Students

School of Computer Science Engineering

Lovely Professional University, Jalandhar

DECLARATION

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we shall take full responsibility for it.

Aditya Raj

11912185

Pankaj Raina

11910326

LOVELY PROFESSIONAL UNIVERSITY
11/11/22

TABLE OF CONTENTS

ABSTRACT:

Twitter sentiment analysis is the method of Natural Language Processing (NLP). In this project named Twitter sentiment Analysis we analyze the sentiments behind the twitter's tweet. We have three type of sentiment: Positive, Neutral and Negative. Analyzing the sentiments behind every tweet is the biggest problem in the early days but now it can be solved with the help of Machine Learning. Twitter is an online micro-blogging and social-networking platform which allows users to write short status udates of maximum length 140 characters and through the Twitter Sentimental Analysis we can analysis the mood of the person who tweet which can helps in the industries to analyze the market and their product reviews or we can know the sentiments behind the opinion on any topic on which the group of people tweet and through this we can find the final result that the people point on view on the particular topic, product and any other tweets suggestions.

Keywords : Machine learning, Twitter Sentiment Analysis, Natural Language Process, Data Minning, Bag of Words(BoG), Embedded layer, Naïve Bayes Classifier, Keras, Natural Language Toolkit.

INTRODUCTION

Sentiment Analysis is the process of determining the sentiment behind the tweet. whether a piece of written text(tweet) is positive, neutral or negative.
It is also referring as opinion analysis, it is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive, neutral or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling object, or predict stock markets for a given company.
Let's suppose There is an election in our country so government starts their campaigning and they want to analyses the people's reaction on their campaigning's advertisements and the tweets of their leaders or party members so they have to know the mood of the public on their actions for this they can analyze the sentiments of replies by the public on social media platforms and calculate the average that their action has been liked by public or not. Other example is like a company launch any product so they have to know the opinions of their buyers who used it. that, are they liked that product or not but they can read each and every reviews so here sentiment analysis plays a very important role in calculating the sentiments through the reviews given by the users in few seconds.
Sentiment analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. This could be done by analyzing overall public sentiment towards that firm with respect to time and using economics tools for finding the correlation between public sentiment and the firm's stock market value. Firms can also estimate how well their product is responding in the market, which areas of the market is it having a favorable response and in which a negative response (since twitter allows us to download stream of geo-tagged tweets for particular locations.

These are the steps used in Machine Learning for analyzing the sentiment.

- Stemming
- Tokenization
- Part of speech tagging
- Parsing
- Lexicon analysis (depending on the relevant context)

Prior Knowledge:

In every field of education, we need prior knowledge to understand an analyze that field very well, prior knowledge become base for successful understanding and analyses of any study. So, before we start to study the actual content of any paper. We have to understand the basic concepts related to the paper that will help us to understand and comprehend the paper very well.

Natural language processing (NLP)

Natural language processing is a subfield linguistic; computer science, information engineering, and artificial intelligence describes the interaction between human language and computers, in particular to program computers to process and analyze large amount of natural language data.

Data Mining:

Data mining is the method to find the useful patterns in large datasets involving methods at the interaction of machine learning, statistics, and database system in order to generate the new useful information from that datasets.

Bag of Words(Bog)

Bag of words model is the NLP technique of text modeling. Whenever we apply any algorithm in NLP, it works on numbers.

We cannot directly feed out text into that algorithm. Hence Bag or Words model is Used to preprocessing the text by converting it into a bag of words, which keeps a count of the total occurrences of most frequent used words.

Embedded layer:

The Embedding layer is used to create word vectors for incoming words.

The Embedding layers is defined as the hidden layers of a network. It must specify 3 arguments:

• Input_dim: this is the size of the vocabulary in the text data. As we know the maximum length twitter allows is 140 words. So, if your data is integer encoded to values between 0-139, then the size of the vocabulary would be 140 words.

• output_dim: It is the size of the vector space in which words will be embedded. It defines the size of the output vectors from the layers for each word.

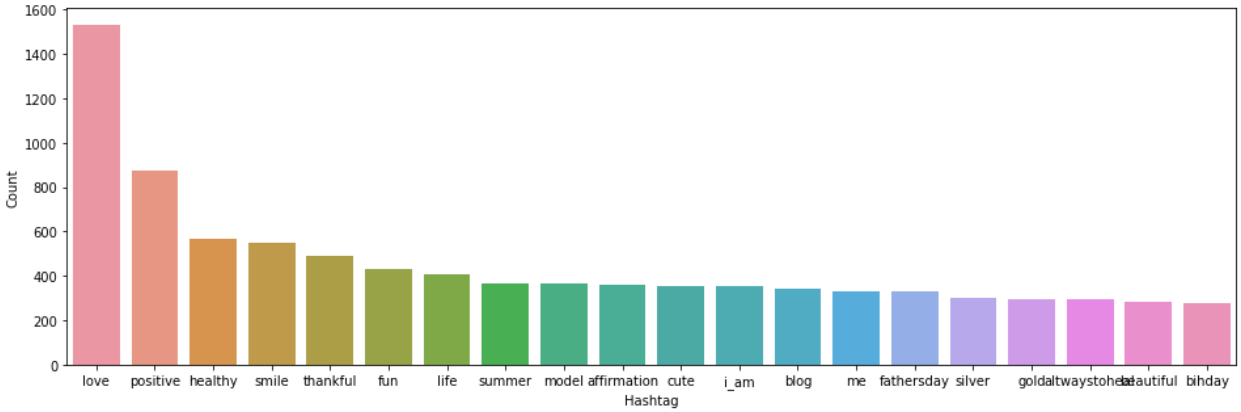
• Input_length: it is the length of input sequence, as you would define for any input layers of a keras mode.

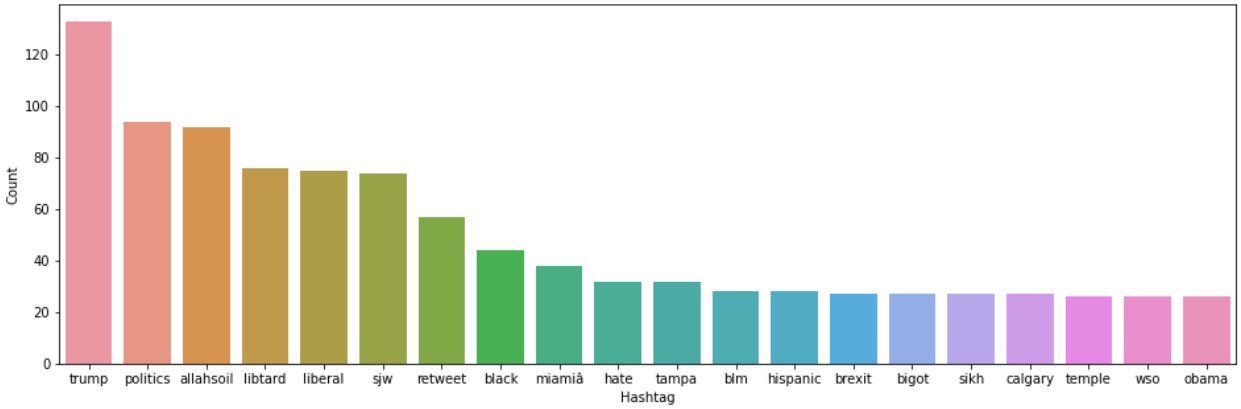
Stopswords:
Stopwords(these are the word we don't want to use in tweets after cleaning the text which are not relevant to predict the sentiments are Positive, Negative and Neutral.
These words may be like 'The', and etc. These are the word which did not give hint about the sentiments is Positive, Negative and Neutral.

Natural language Toolkit(NLTK):
It is a classic library in NLP which allow us to download the ensemble of Stopwords.
The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

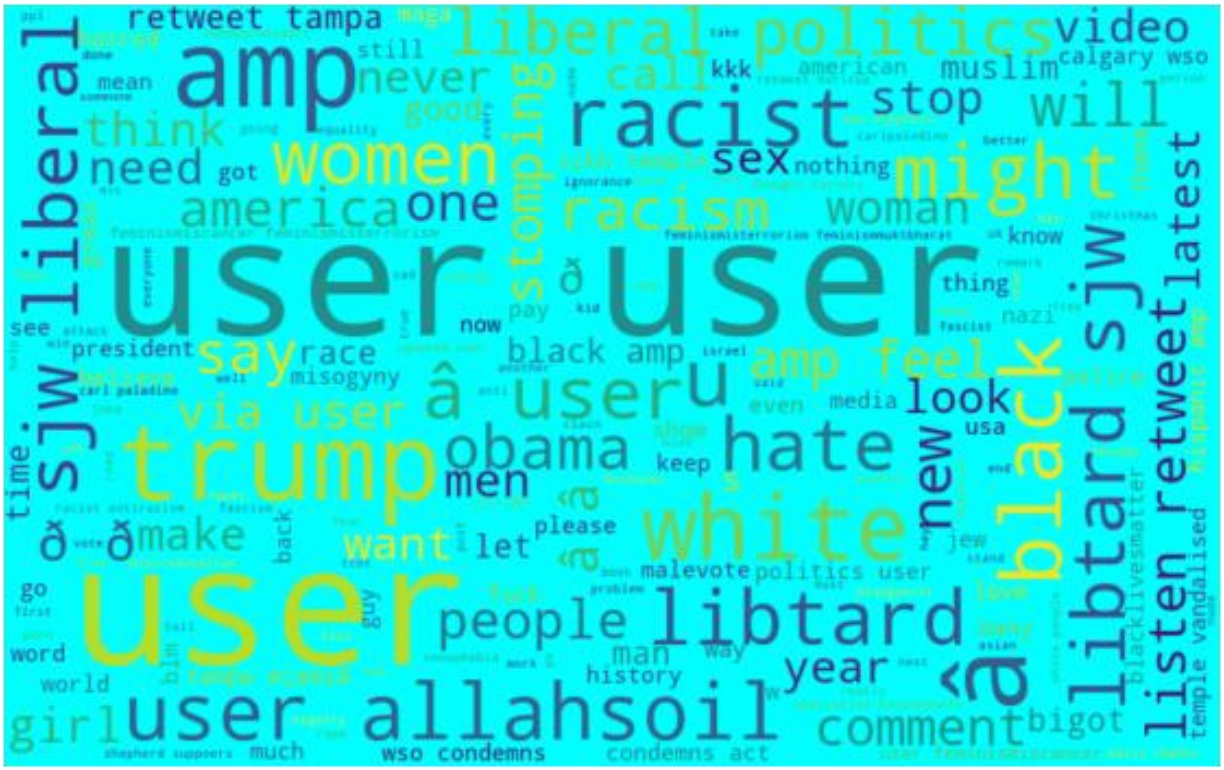
Stemming:
It is the method of NLP use in cleaning the text. It transforms all the conjugates of the words for exam it treated loved & love same just simplify the tweets because both means the positive sentiments.

Luv	0.5732780694961548
Loves	0.5732780694961548
Loved	0.5373271703720093
Amazing	0.5026600360870361
Adore	0.4942743480205536
Awesome	0.4598265290260315
Loveee	0.4531649351119995
Loooove	0.44260522723197937





The Negative Words



Machine Learning

It is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. It focuses on the development of computer programs that can access data and use it learn for themselves.

Machine learning are of three types:

1. Supervised leaning
2. Unsupervised learning
3. Reinforcement learning

In this project we use Naïve Bayes classifier methods which are the part of Supervised learning .

Naïve Bayes Classifier:

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive)independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum likelihood training can be done by evaluating a closed form expression (mathematical expression that can be evaluated in a finite number of operations), which takes linear time.

It is based on the application of the Baye's rule given by the following formula: Bayes theorem provides a way of calculating posterior P(c|x) from P(c) and P(x|c). Look at the equation below:

Above,

- P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

How Naive Bayes algorithm works?

Let's understand it using it an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggestion possibilities of playing). Now, we need to classify whether player will play or not based on weather condition Let's follow the steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast Probability = 0.29 and probability of playing is 0.64.

Step 3: Now, use Naive Bayes equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: player will play if weather is sunny. Is this statement is correct?

We can solve it using discussed method of posterior probability.

P(Yes|Sunny) = P(Sunny|Yes) * P(Yes)/P(Sunny)

Here we have P(Sunny|Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(yes) =9/14 =0.64

Now, P (Yes|Sunny) = 0.33 * 0.64/0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

Naive Bayes uses a similar method to predict probability of different class based on various attributes. This algorithm is we used in text classification through which we can analysis the sentiment behind the text(tweet).

Cross Validation:

Cross-validation is a technique to evaluate predictive models by dividing the original dataset into a training set and test set. Training set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original sample is randomly divided into k equal size subsets. Of the k subsets, a single subset is taken as the validation data for testing the model, and the remaining k-1 subsets are used for training the model. The cross-validation process is then repeated k times (the folds), with each of the k subsets used exactly once as the validation data and average accuracy of k-folds is taken as final accuracy. In most experiments 10-fold cross validation technique issued. In10-fold cross validation all the instances of the dataset are used and are divided into 10 disjoint groups, where nine groups are used for training and the remaining one is used for testing. The algorithm runs for10 times and average accuracy of all folds is calculated.

METHODOLOGY

Dataset has been divided into two sub categories:

train_tweet and test_tweet for the purpose of training and testing and validation.

train_tweet consists of 3 columns of name id, label, tweet and 31963 rows for evaluation purpose.

test_tweet consists of 2 columns of name id and tweet; rows in no. 17198

In below picture we are showing the positive comments from the train set

```
[ ] # checking out the postive comments from the train set

    train[train['label'] == 1].head(10)
```

	id	label	tweet
13	14	1	@user #cnn calls #michigan middle school 'buil...
14	15	1	no comment! in #australia #opkillingbay #se...
17	18	1	retweet if you agree!
23	24	1	@user @user lumpy says i am a . prove it lumpy.
34	35	1	it's unbelievable that in the 21st century we'...
56	57	1	@user lets fight against #love #peace
68	69	1	ð□□©the white establishment can't have blk fol...
77	78	1	@user hey, white people: you can call people '...
82	83	1	how the #altright uses & insecurity to lu...
111	112	1	@user i'm not interested in a #linguistics tha...

Adding a column to represent the length of the tweet:

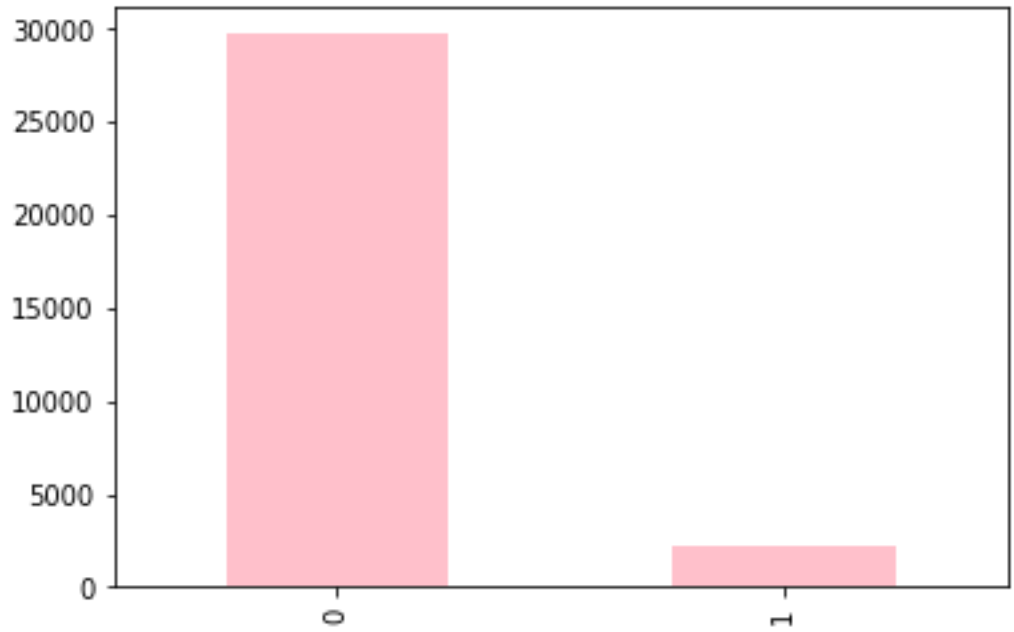
```
[ ] # adding a column to represent the length of the tweet

    train['len'] = train['tweet'].str.len()
    test['len'] = test['tweet'].str.len()

    train.head(10)
```

	id	label	tweet	len
0	1	0	@user when a father is dysfunctional and is s...	102
1	2	0	@user @user thanks for #lyft credit i can't us...	122
2	3	0	bihday your majesty	21
3	4	0	#model i love u take with u all the time in ...	86
4	5	0	factsguide: society now #motivation	39
5	6	0	[2/2] huge fan fare and big talking before the...	116
6	7	0	@user camping tomorrow @user @user @user @use...	74
7	8	0	the next school year is the year for exams.ð□□...	143
8	9	0	we won!!! love the land!!! #allin #cavs #champ...	87
9	10	0	@user @user welcome here ! i'm it's so #gr...	50

Graphical representation of Dataset:



Dataset after cleaning:

For cleaning the dataset, we apply following methods:

1-removing all the punctuations (letters expect [a-zA-Z])

2-Removing all the Stopwords(these are the word we don't want to use in tweets after cleaning the text which are not relevant to predict the sentiments are Positive, Negative and Neutral.

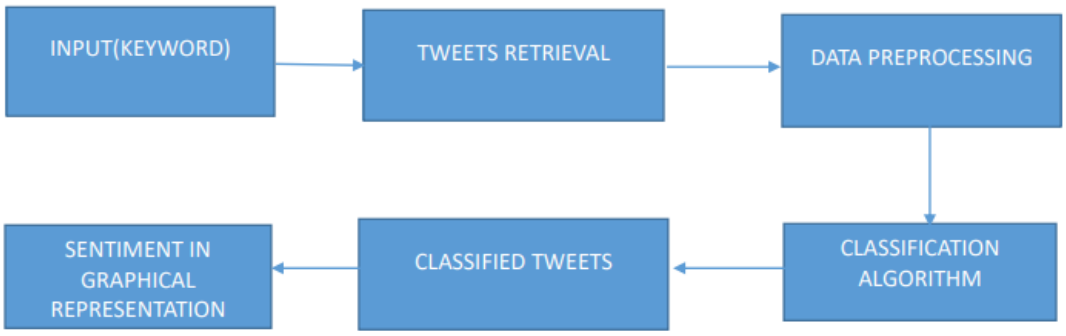
These words may be like 'The', 'and', 'is', 'are' etc. These are the word which did not give any hint about the sentiments is Positive, Negative and Neutral.

3-Make all the words of text in lower case.

4- Apply Stemming

Proposed Model:

Design:



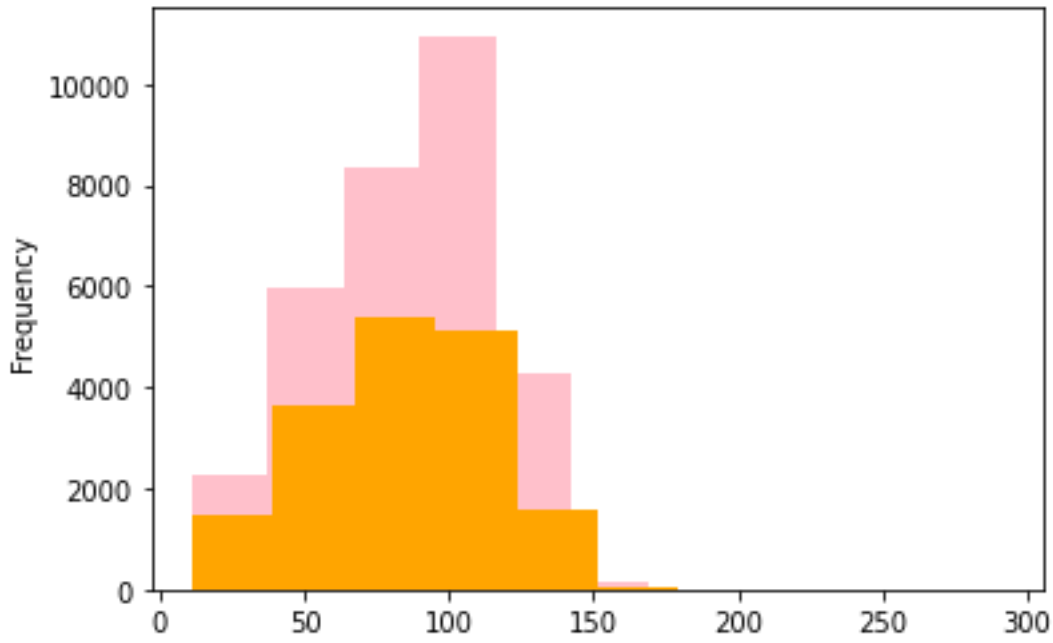
Observation:

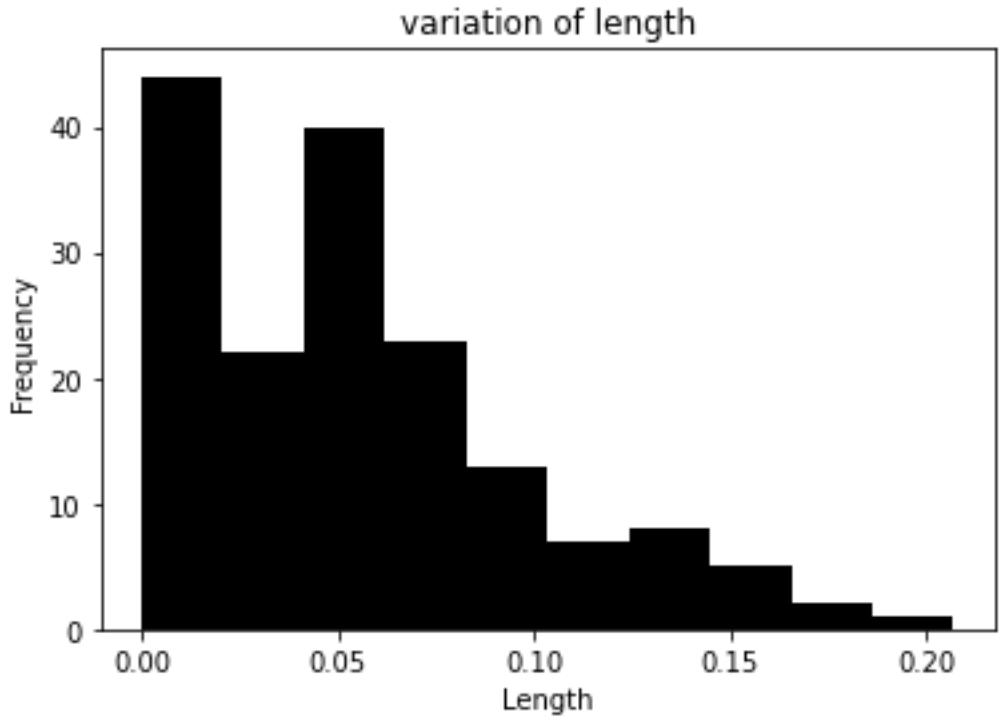
To make the validation set, there are two main options:

• Split the training set into two parts (60%, 20%) with a ratio 2:8 where each part contains an equal distribution of example types. We train the classifier with the largest part, and make prediction with the smaller one to validate the model. This technique works well but has the disadvantage of our classifier not getting trained and validated on all examples in the data set (without counting the test set).

• The K-fold cross validation. We split the data set into k parts, hold out one, combine the others and train on them, then validate against the held-out portion. We repeat that process k times (each fold), holding out a different portion each time. Then we average the score measured for each fold to get a more accurate estimation of our model's performance.

Graphical representation of accuracy, validation_loss, validation_accuracy:





We split the training data into 8 folds and cross validate on them using scikit learn as shown in the figures above. The number of K-folds is arbitrary and usually set to 8 it is not a rule. In fact, determine the best K is still an unsolved problem but

with lower K: computationally cheaper, less variance, more bias. With large K: computationally expensive, higher variance, lower bias.

Observation:

Confusion Matrix:

Confusion matrix it the table that is used to describe the performance of the model on the set of the test data. It is also known as error matrix. In this the row of the matrix represented the instances in a predicted class and column of the matrix represent the instances in an actual class (or vice versa).

	Actually positive(1)	Actually negative(0)
Predicted positive(1)	True Positives(TPs)	False Positives(FPs)
Predicted negative(0)	True Negatives(TNs)	False Negatives(FNs)

- True Positive (TP): Observation is Positive, and is predicted to be positive.
- False Negative (FN): Observation is positive, but predicted negative
- True Negative (TN): Observation is negative, and is predicted to be negative.
- False Positive (FP): Observation is negative, but is predicted positive.

Accuracy:

Accuracy is the ratio of total number of Correct predictions to the total number of input samples. It is also known as Classification Rate. This is the formula of accuracy/Classification Rate.

Accuracy = (TP+TN)/(TP+TN+FP+FN)

RESULTS:

After Implementing the Machine learning Algorithm or method, the table in below figure represent the Accuracy, Precision, F-1 Score, macro_average, Weighted_average.

Random Forest Classifier:
A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-sample of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score

model = RandomForestClassifier()
model.fit(x_train, y_train)

y_pred = model.predict(x_valid)

print("Training Accuracy :", model.score(x_train, y_train))
print("Validation Accuracy :", model.score(x_valid, y_valid))

# calculating the f1 score for the validation set
print("F1 score :", f1_score(y_valid, y_pred))

# confusion matrix
cm = confusion_matrix(y_valid, y_pred)
print(cm)
```

```
Training Accuracy : 0.999123941429227
Validation Accuracy : 0.9520710799649605
F1 score : 0.6103763987792472
[[7308  124]
 [ 259  300]]
```

Logistic Regression:

Logistic regression is commonly used for prediction and classification problems

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(x_train, y_train)

y_pred = model.predict(x_valid)

print("Training Accuracy :", model.score(x_train, y_train))
print("Validation Accuracy :", model.score(x_valid, y_valid))

# calculating the f1 score for the validation set
print("f1 score :", f1_score(y_valid, y_pred))

# confusion matrix
cm = confusion_matrix(y_valid, y_pred)
print(cm)
```

```
Training Accuracy : 0.9851487213716574
Validation Accuracy : 0.9416843949443123
f1 score : 0.5933682373472949
[[7185  247]
 [ 219  340]]
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

SVC:

```
from sklearn.svm import SVC

model = SVC()
model.fit(x_train, y_train)

y_pred = model.predict(x_valid)

print("Training Accuracy :", model.score(x_train, y_train))
print("Validation Accuracy :", model.score(x_valid, y_valid))

# calculating the f1 score for the validation set
print("f1 score :", f1_score(y_valid, y_pred))

# confusion matrix
cm = confusion_matrix(y_valid, y_pred)
print(cm)
```

```
Training Accuracy : 0.978181969880272
Validation Accuracy : 0.9521962207483419
f1 score : 0.4986876640419947
[[7419   13]
 [ 369  190]]
```

XGB regression:

Short form of extreme Gradient Boosting classifier

```
from xgboost import XGBClassifier

model = XGBClassifier()
model.fit(x_train, y_train)

y_pred = model.predict(x_valid)

print("Training Accuracy :", model.score(x_train, y_train))
print("Validation Accuracy :", model.score(x_valid, y_valid))

# calculating the f1 score for the validation set
print("f1 score :", f1_score(y_valid, y_pred))

# confusion matrix
cm = confusion_matrix(y_valid, y_pred)
print(cm)
```

CONCLUSION:

Machine Learning is a hot topic for the industries. In this project we are tried to analyze the sentiments of the tweets. But we are still far to detect the sentiments of corpus of texts very accurately of the complexity in the English language and even more if we consider the other countries languages like chines.
In this project we tried to show the basic way of classifying tweets into positive, neutral and negative category using Naïve Bayes as baseline and how language models are related of the Naïve Bayes and can produce better results.
We could further improve our class classifier by trying to extract more features from the tweets, trying different kinds of features, tuning the parameters of the Naive Bayes Classifier.

REFERENCES:

[1]. Alexander Pak, Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining.

[2]. Twitter sentiments analyze dataset from Kaggle

[3]. Jin Bai, JianYun Nie. Using Language Models for Text Classification.

[4]. AnalyticsVidya: For Naïve Bayes, (https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/)

[5]. Twitter sentiment analyses report on www.cse.ust.hk

[6]. Natural language processing from Wikipedia

GITHUB LINK

venom005/Twitter-Sentiment-Analysis (github.com)