

# LP1 Assignment DA 4

Date: 19<sup>th</sup> October, 2020

Title: Twitter Data Analysis

## Problem Definition:

Use Twitter data for sentiment analysis. The dataset is 3MB in size and has 31,962 tweets. Identify the tweets which are hate tweets and which are not

## Learning Objectives:

- Learn Classification Algorithms.
- Learn to summarize the properties in the training dataset.
- Learn to split the dataset into training and test datasets.
- Learn to develop a predictive classification model.

## Learning Outcomes:

I will be able to develop a classification model for sentiment analysis of tweets on twitter.

## S/W & H/W Packages:

1. Operating System: 64-bit Open source Linux or its derivative
2. Programming Language: Python3
3. Google Colaboratory(uses Tesla K80 GPU)
4. Python Libraries : Sklearn, Pandas, Matplotlib, NLTK, Keras

## Related Mathematics:

Let  $S$  be the system set:

$$S = \{s; e; X; Y; Fme; DD; NDD; Fc; Sc\}$$

where Dataset is loaded into the dataframe

$s$ =start state

$e$ =end state predicted state of tweets (0 - Not a Hate Tweet, 1 - Hate Tweet)

$X$ =set of inputs

$X = \{X1\}$  where  $X1$  = Twitter Dataset (31962 records, 2 columns)

$Y$ =set of outputs  $Y = \{Y1, Y2, Y3\}$

1.  $Y1$  = Predicted Values
2.  $Y2$  = Confusion Matrix
3.  $Y3$  = Accuracy Score

$Fme$  is the set of main functions

$Fme = \{f0\}$

- $f0$  = Main Display Function

$Ff$  is the set of friend functions

$Ff = \{f1, f2, f3, f4, f5\}$

where

1.  $f1$  = function to load dataset into pandas dataframe
2.  $f2$  = function to clean data
3.  $f3$  = function to tokenize
4.  $f4$  = function to split dataset into test and train data
5.  $f5$  = function to train the model

$DD$  = Deterministic Data Twitter Dataset

$NDD$  = Non-deterministic data (Eg - Null Values in Dataset)

No null value detected

$Fc$  = failure case

Low Value of Accuracy Score

# Concepts Related to Theory:

## **Sentiment Analysis**

Sentiment analysis (also known as opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

## **Classification**

In statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. Examples are assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e., learning where a training set of correctly identified observations is available.

## **Support Vector Machines**

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. Computing the (soft-margin) SVM classifier amounts to minimizing an expression of the form

$$\left[ \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i(w \cdot x_i - b)) \right] + \lambda \|w\|^2.$$

## **LSTM**

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The compact forms of the equations for the forward pass of an LSTM unit with a forget gate are:

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}$$

## Natural Language Processing

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural-language generation.

### Tokenization

Tokenization is the process of demarcating and possibly classifying sections of a string of input characters. The resulting tokens are then passed on to some other form of processing. The process can be considered a sub-task of parsing input.

### Stemming

In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word

### Lemmatization

In computational linguistics, lemmatization is the algorithmic process of determining the lemma of a word based on its intended meaning. Unlike stemming, lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. As a result, developing efficient lemmatization algorithms is an open area of research.

### Stopwords

Stopwords are the words in any language which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on.

## TF-IDF

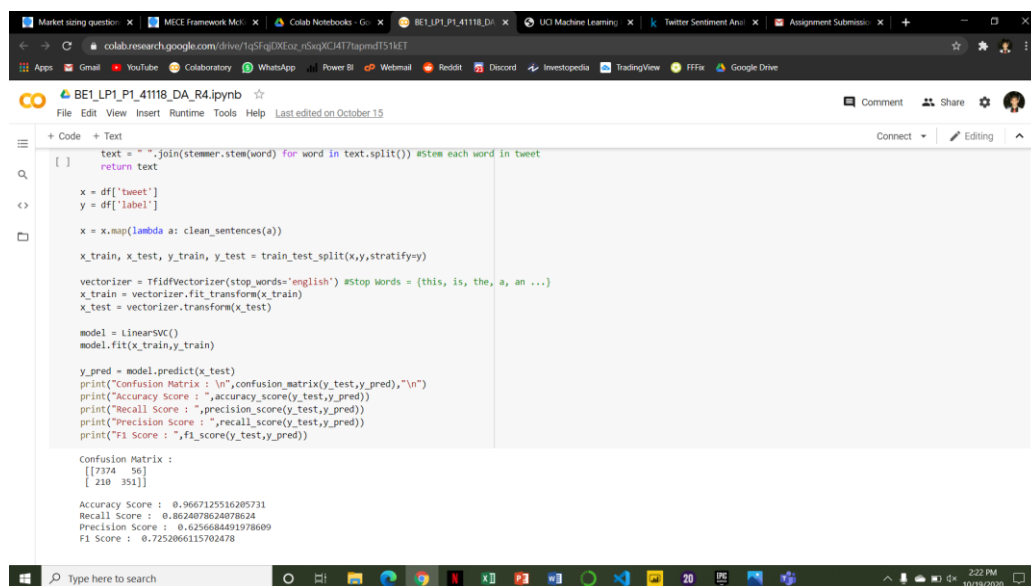
In information retrieval, tf-idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf-idf is one of the most popular term-weighting schemes today. A survey conducted in 2015 showed that 83% of text-based recommender systems in digital libraries use tf-idf.

## NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language.

## Output:

### 1. Model Metrics



```
text = " ".join(stemmer.stem(word) for word in text.split()) #stem each word in tweet
return text

x = df['tweet']
y = df['label']

x = x.map(lambda a: clean_sentences(a))

x_train, x_test, y_train, y_test = train_test_split(x, y, stratify=y)

vectorizer = TfidfVectorizer(stop_words='english') #stop words = {this, is, the, a, an ...}
x_train = vectorizer.fit_transform(x_train)
x_test = vectorizer.transform(x_test)

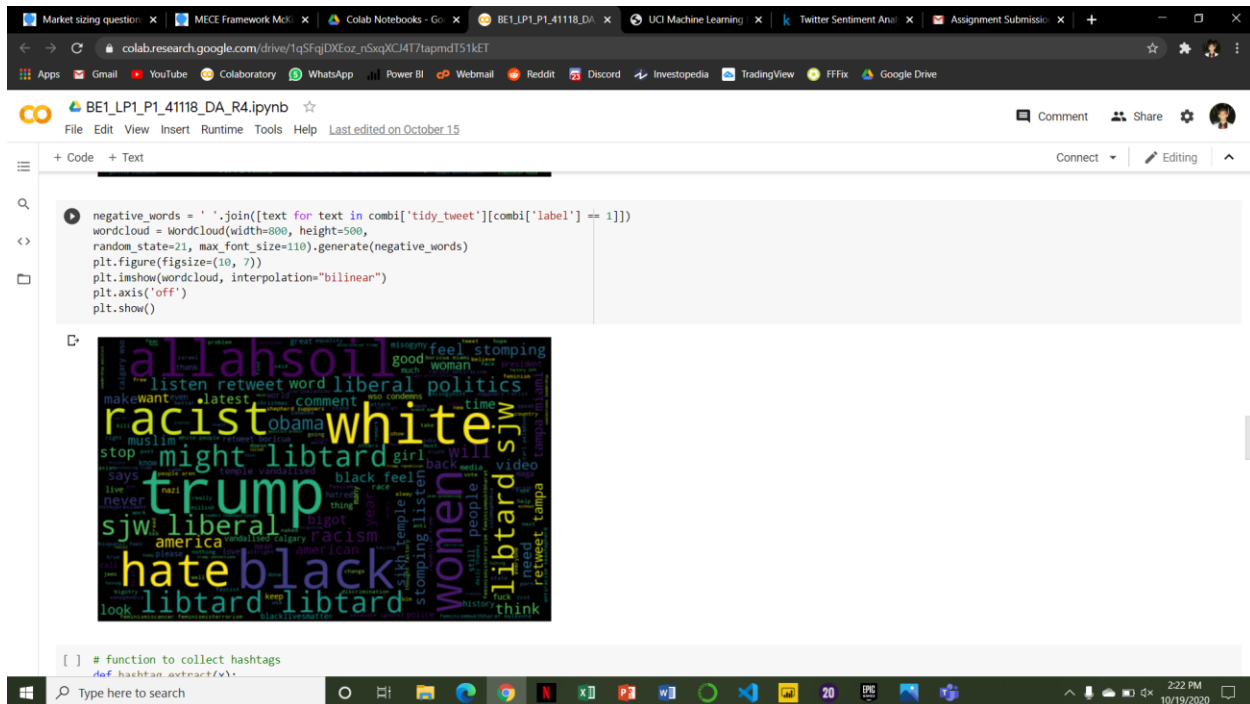
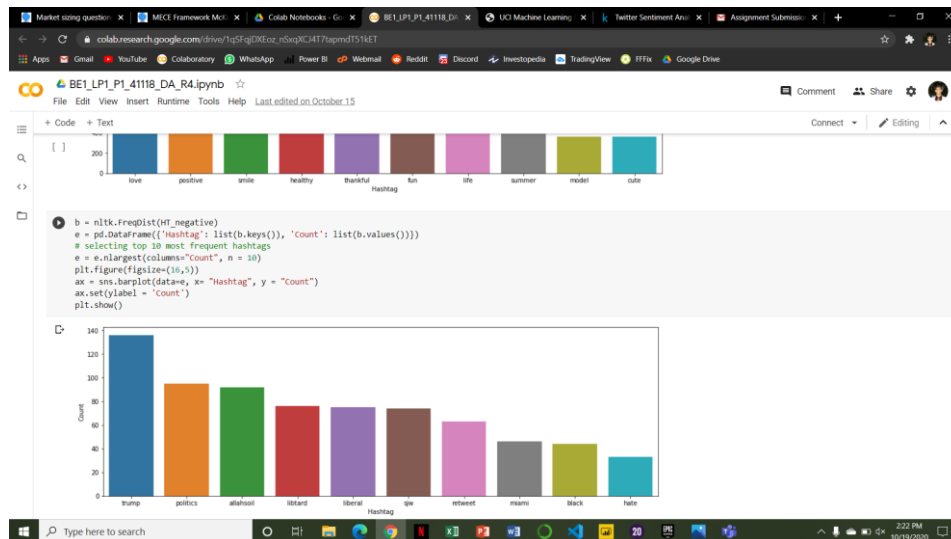
model = LinearSVC()
model.fit(x_train, y_train)

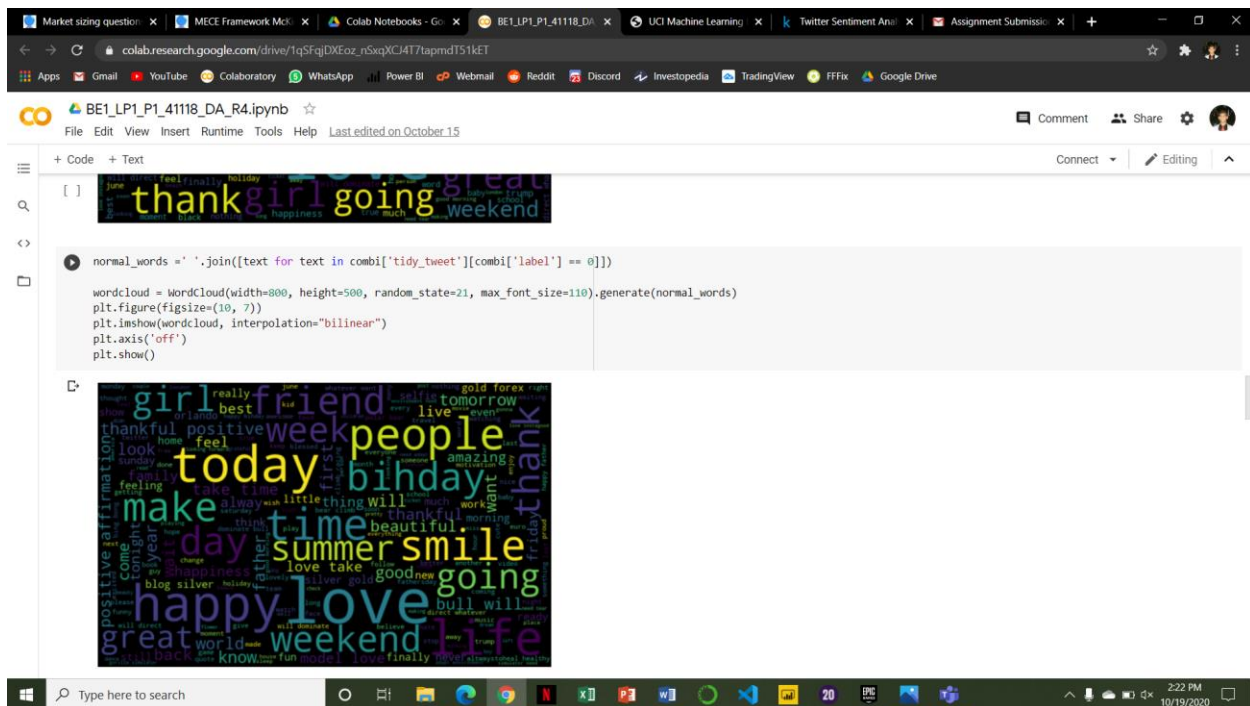
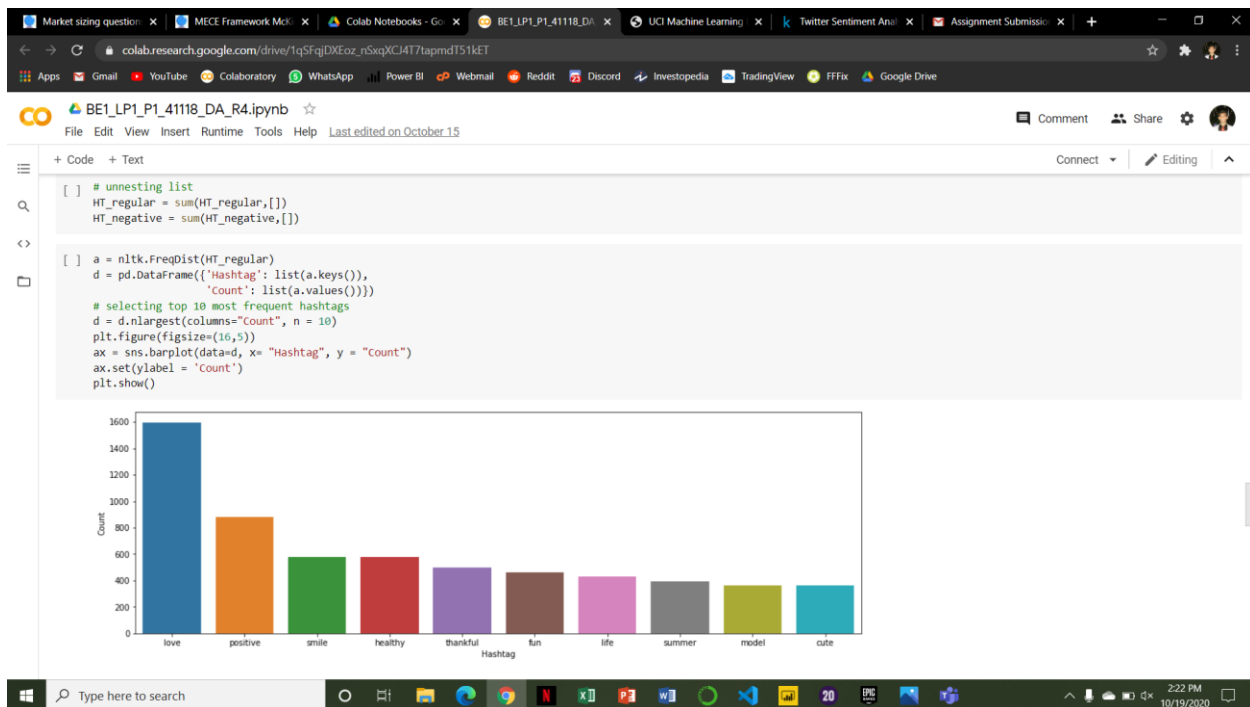
y_pred = model.predict(x_test)
print("Confusion Matrix : \n", confusion_matrix(y_test, y_pred), "\n")
print("Accuracy Score : ", accuracy_score(y_test, y_pred))
print("Recall Score : ", recall_score(y_test, y_pred))
print("Precision Score : ", precision_score(y_test, y_pred))
print("F1 Score : ", f1_score(y_test, y_pred))

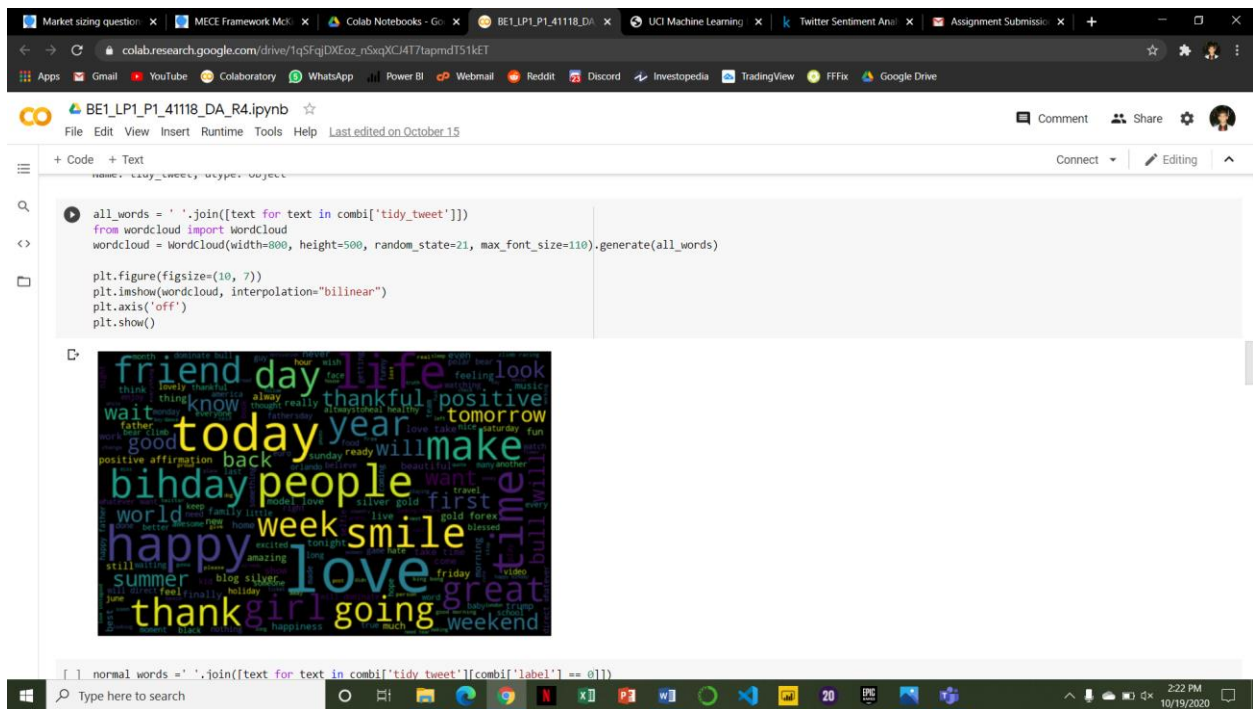
Confusion Matrix :
[[7374  56]
 [ 210 351]]

Accuracy Score : 0.9667125516285731
Recall Score : 0.8624078624078624
Precision Score : 0.6256684491378609
F1 Score : 0.7252866115702478
```

## 2. Visualization of Data







## Notebook Link:

[https://colab.research.google.com/drive/1qSFqjDXEoz\\_nSxqXCJ4T7tapmdT51kET?usp=sharing](https://colab.research.google.com/drive/1qSFqjDXEoz_nSxqXCJ4T7tapmdT51kET?usp=sharing)

## Conclusion:

I have successfully developed a classification model for sentiment analysis of Twitter dataset.