CSCE 5300 Introduction to Big Data and Data science

Twitter Sentiment Analysis using TextBlob and RoBERTa

Jaideep Janapati

11607202

jaideepjanapati@my.unt.edu

Professor :Yunhe Feng

Problem Statement:

Classification of tweets as positive/negative/neutral or determining the sentiment score for each tweet.

Software Requirements:

Python, Jupyter Notebook.

Sentiment Analysis is a process of computationally identifying and classifying the emotions expressed in a text regarding a product or topic. It is also called opinion mining and is an approach to **Natural Language Processing** [NLP]. Sentiment Analysis is useful for determining the opinions of people regarding a product which is helpful for product's improvement. In this task, for determining the sentiment of tweets, I have used TextBlob and Roberta. There are three steps in this process

- Reading the CSV file
- Data Cleaning
- Classification

Before reading the Data, we need to import libraries for our code

```
import pandas as pd

import numpy as np

from textblob import TextBlob

import re

import csv

import sys

from nltk.stem.porter import PorterStemmer

from transformers import AutoModelForSequenceClassification

from transformers import TFAutoModelForSequenceClassification
from transformers import AutoTokenizer

from scipy.special import softmax
```

Reading the Data:

We need to read to csv but here the raw data does not have ‘,’ as delimiter so we used read_table to read the data.

Code

```
data = pd.read_table("Task-1 tweets_1000.csv",header=None)

words=[]

for i in data[0]:

    words.append(i)
```

Data Cleaning:

Properly cleaned data will help us to do good text analysis. In this case, a tweet can contain URLs, emoticons, emojis, Punctuations etc. which are not necessary for the text analysis. We need to remove these unwanted texts in the tweets. Also, stemming is important aspect of data cleaning. Porter Stemmer is an important stemmer library which reduces a word to its stem word. From Data cleaing step we get processed tweets which we use for classification.

Code

```
use_stemmer =True

def preprocess_word(word):

    # Remove punctuation

    word = word.strip('\'"?!,.():;')

    # Convert more than 2 letter repetitions to 2 letter

    # funnnnny --> funny

    word = re.sub(r'(.)\1+', r'\1\1', word)

    # Remove - & '

    word = re.sub(r'(-|\')', '', word)

    return word



def is_valid_word(word):
    # Check if word begins with an alphabet
    return (re.search(r'^[a-zA-Z][a-z0-9A-Z\._]*$', word) is not None)


def handle_emojis(tweet):
    # Smile -- :), : ), :-), (:, ( :, (-:, :')
    tweet = re.sub(r'(:\s?\)|:-\)|\(\s?:|\(-:|:\'\))', ' EMO_POS ', tweet)
    # Laugh -- :D, : D, :-D, xD, x-D, XD, X-D
    tweet = re.sub(r'(:\s?D|:-D|x-?D|X-?D)', ' EMO_POS ', tweet)
    # Love -- <3, :*
    tweet = re.sub(r'(<3|:\*)', ' EMO_POS ', tweet)
    # Wink -- ;-), ;), ;-D, ;D, (;,  (-;
    tweet = re.sub(r'(;-?\)|;-?D|\(-?;)', ' EMO_POS ', tweet)
    # Sad -- :-(, : (, :(, ):, )-:
    tweet = re.sub(r'(:\s?\(|:-\(|\)\s?:|\)-:)', ' EMO_NEG ', tweet)
    # Cry -- :,(, :'(, :"(
    tweet = re.sub(r'(:,\(|:\'\(|:"\()', ' EMO_NEG ', tweet)
    return tweet


def preprocess_tweet(tweet):
    processed_tweet = []
    # Convert to lower case
    tweet = tweet.lower()
    # Replaces URLs with the word URL
    tweet = re.sub(r'((www\.[\S]+)|(https?://[\S]+))', ' URL ', tweet)
    # Replace @handle with the word USER_MENTION
    tweet = re.sub(r'@[\S]+', 'USER_MENTION', tweet)
    # Replaces #hashtag with hashtag
    tweet = re.sub(r'#(\S+)', r' \1 ', tweet)
    # Remove RT (retweet)
    tweet = re.sub(r'\brt\b', '', tweet)
    # Replace 2+ dots with space
    tweet = re.sub(r'\.{2,}', ' ', tweet)
    # Strip space, " and ' from tweet
    tweet = tweet.strip(' "\'')
    # Replace emojis with either EMO_POS or EMO_NEG
    tweet = handle_emojis(tweet)
    # Replace multiple spaces with a single space
    tweet = re.sub(r'\s+', ' ', tweet)
    words = tweet.split()

    for word in words:
        word = preprocess_word(word)
        if is_valid_word(word):
            word = str(porter_stemmer.stem(word))
            processed_tweet.append(word)

    return ' '.join(processed_tweet)

porter_stemmer = PorterStemmer()
```

<u>Classification:</u>

A.TextBlob:

TextBlob is a Python (2 and 3) library for handling textual data. It gives a basic Programming interface to look into natural language processing (NLP) tasks, for example, part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Code:

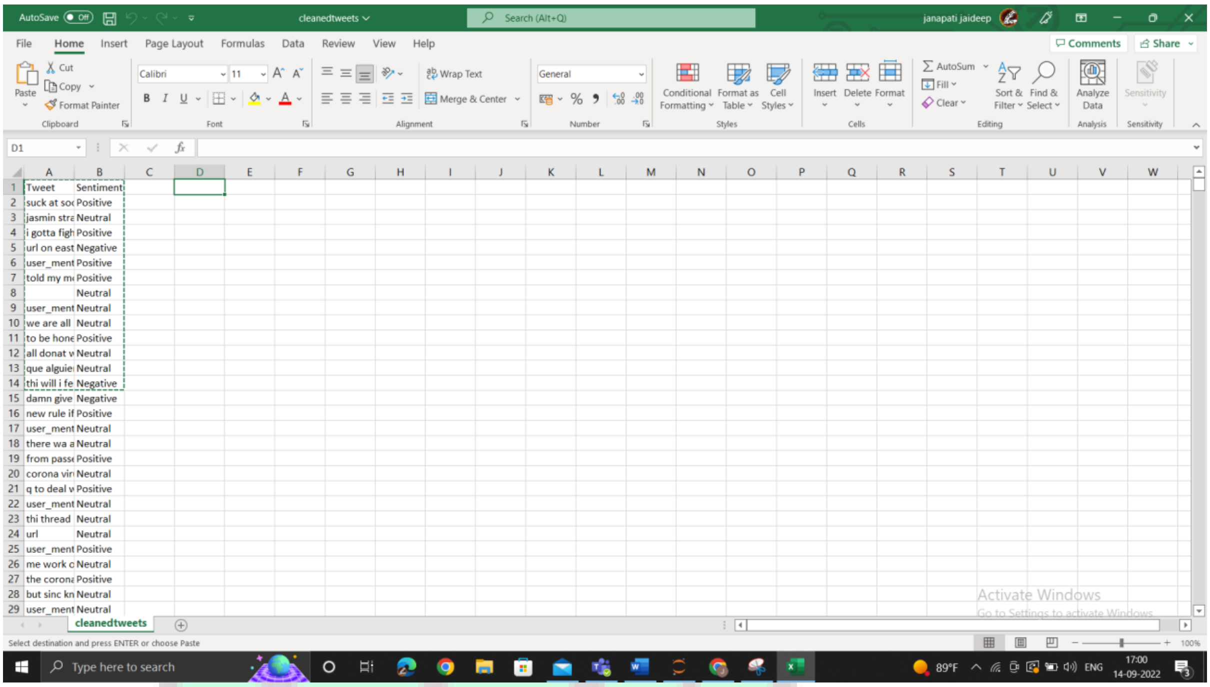
```
Tweetdf = pd.DataFrame(columns=['Tweet','Sentiment'])
```

```
for tweet in processed_tweets:
    print(tweet)
    textanalysis = TextBlob(tweet)
    if textanalysis.sentiment[0]>0:
        status = "Positive"
        print ('Positive')
    elif textanalysis.sentiment[0]<0:
        print ('Negative')
        status = "Negative"
    else:
        print ('Neutral')
        status = "Neutral"
    Tweetdf = Tweetdf.append({'Tweet': tweet, 'Sentiment': status}, ignore_index=True)
Tweetdf.reset_index(drop= True,inplace= True)
Tweetdf.to_csv('/Users/jaide/Desktop/cleanedtweets.csv',index=None)
```

Output:

```
Tweet                                                                                          Sentiment
suck at social distanc                                                                         Positive
jasmin strang share a messag of hope dure thi life of covid19 musicvideo product by moodolog if
you like it dm me whi if you coronaviru stayathom oaklandfilmmak url                           Neutral
i gotta fight these allergi in public to make sure peopl think i got corona                    Positive
url on easter pleas rememb the poor and desol covid19 infrastructur lalov natgeo compass      Negative
user_ment i have a cute one made from recycl sari silk my friend got me in nepal               Positive
told my mom we should start to work from home due to corona no one els work in our offic but
is but ya know work from home sound nice                                                       Positive
                                                                                               Neutral
user_ment user_ment user_ment user_ment user_ment user_ment so use your logic of not pay
ani attent to fact and detail trump call covid a and seven week later infect and kill american Neutral
we are all in deep doo doo                                                                     Neutral
to be honest everyon wa scare of coronaviru and govt had to somehow convinc the public to
compli inflat death number were use to keep us at home regardless the end result is what matter
unfortun dem want thi to be biblic so they could blame trump                                   Positive
all donat will be distribut by halo to lowincom famili elderli homeless commun member and those
recent affect by covid19 who need help feed their pet noth is                                  Neutral
que alguien expliqu                                                                            Neutral
```

thi will i fear continu as it seem like it wa a bad flu season last winter but mayb it wa circul quit a bit earlier than origin thought which ha far reach implic Negative



	A	B
1	Tweet	Sentiment
2	suck at soc	Positive
3	jasmin stra	Neutral
4	i gotta figh	Positive
5	url on east	Negative
6	user_ment	Positive
7	told my m	Positive
8		Neutral
9	user_ment	Neutral
10	we are all	Neutral
11	to be hone	Positive
12	all donat w	Neutral
13	que alguie	Neutral
14	thi will i fe	Negative
15	damn give	Negative
16	new rule if	Positive
17	user_ment	Neutral
18	there wa a	Neutral
19	from passe	Positive
20	corona vir	Neutral
21	q to deal w	Positive
22	user_ment	Neutral
23	thi thread	Neutral
24	url	Neutral
25	user_ment	Positive
26	me work c	Neutral
27	the corona	Positive
28	but sinc kn	Neutral
29	user_ment	Neutral

B. RoBERTa

RoBERTa builds on BERT’s language masking strategy, wherein the system learns to predict intentionally hidden sections of text within otherwise unannotated language examples. RoBERTa, which was implemented in PyTorch, modifies key hyperparameters in BERT, including removing BERT’s next-sentence pretraining objective, and training with much larger mini-batches and learning rates. This allows RoBERTa to improve on the masked language modeling objective compared with BERT and leads to better downstream task performance.

BERT is a robustly optimized method for pretraining natural language processing (NLP) systems that improves on Bidirectional Encoder Representations from Transformers, or BERT, the self-supervised method released by Google in 2018.

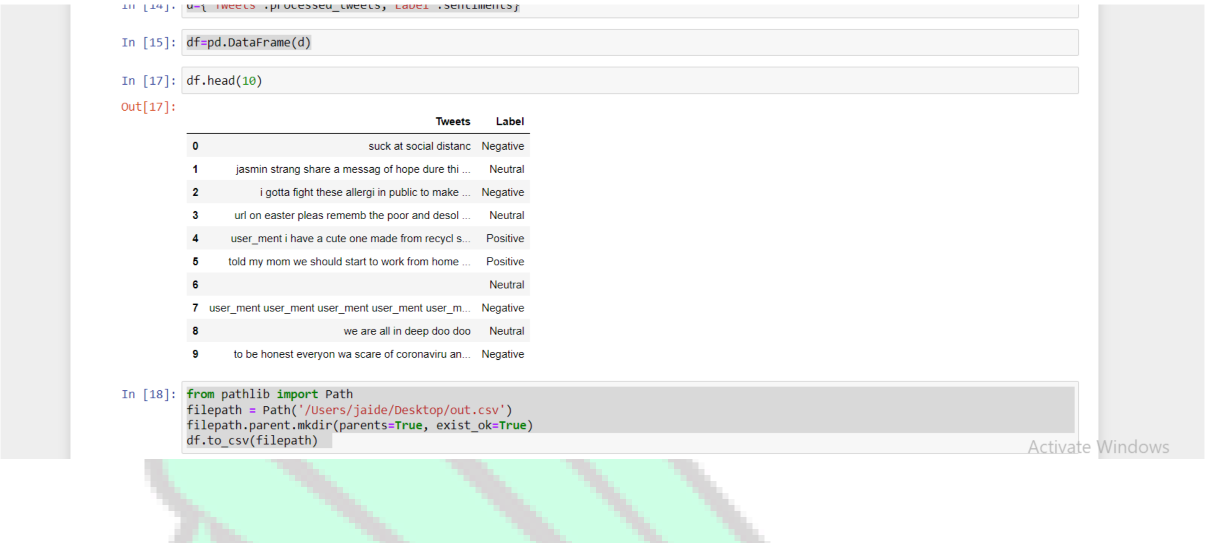
RoBERTa falls under Unsupervised Machine learning Algorithms.

Code

```
roberta ="cardiffnlp/twitter-roberta-base-sentiment"

model=AutoModelForSequenceClassification.from_pretrained(roberta)

tokenizer = AutoTokenizer.from_pretrained(roberta)
```

```
labels = ['Negative', 'Neutral', 'Positive']
sentiments=[]

# sentiment analysis
def sentiment(tweet):
        error=processed_tweets[942]
        if(tweet!=error):
            encoded_tweet = tokenizer(tweet, return_tensors='pt')
# output = model(encoded_tweet['input_ids'], encoded_tweet['attention_mask'])
            output = model(**encoded_tweet)
            scores = output[0][0].detach().numpy()
            scores = softmax(scores)
            max_score=max(scores)
            for i in range(len(scores)):
                if(scores[i]==max_score):
                    return  labels[i]
        else:
            return 'Negative'

for word in processed_tweets:
    sentiments.append(sentiment(word))

d={'Tweets':processed_tweets,'Label':sentiments}

df=pd.DataFrame(d)
from pathlib import Path
filepath = Path('/Users/jaide/Desktop/out.csv')
filepath.parent.mkdir(parents=True, exist_ok=True)
df.to_csv(filepath)
```

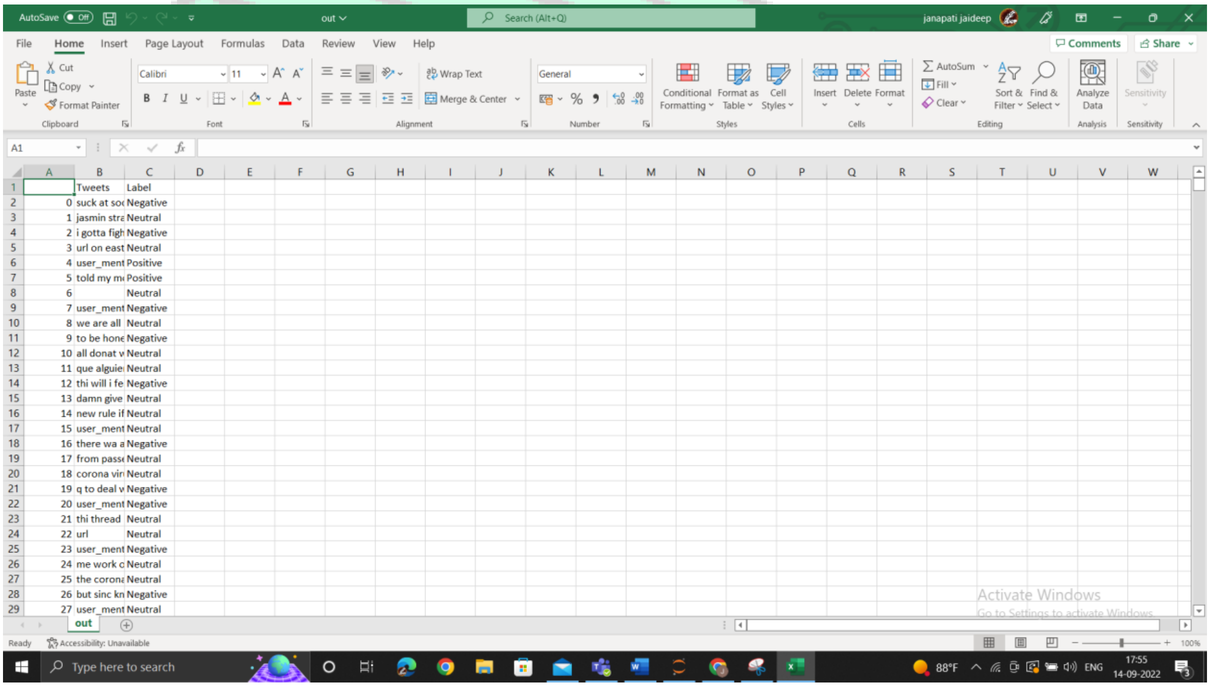
Output:



```
In [14]: d={'Tweets':processed_tweets,'Label':sentiments}

In [15]: df=pd.DataFrame(d)

In [17]: df.head(10)
Out[17]:
```

	Tweets	Label
0	suck at social distanc	Negative
1	jasmin strang share a messag of hope dure thi ...	Neutral
2	i gotta fight these allergi in public to make ...	Negative
3	url on easter pleas rememb the poor and desol ...	Neutral
4	user_ment i have a cute one made from recycl s...	Positive
5	told my mom we should start to work from home ...	Positive
6		Neutral
7	user_ment user_ment user_ment user_ment user_m...	Negative
8	we are all in deep doo doo	Neutral
9	to be honest everyon wa scare of coronaviru an...	Negative

```
In [18]: from pathlib import Path
         filepath = Path('/Users/jaide/Desktop/out.csv')
         filepath.parent.mkdir(parents=True, exist_ok=True)
         df.to_csv(filepath)
```



	A	B	C
1		Tweets	Label
2	0	suck at soc	Negative
3	1	jasmin stra	Neutral
4	2	i gotta figh	Negative
5	3	url on east	Neutral
6	4	user_ment	Positive
7	5	told my m	Positive
8	6		Neutral
9	7	user_ment	Negative
10	8	we are all	Neutral
11	9	to be hone	Negative
12	10	all donat w	Neutral
13	11	que alguie	Neutral
14	12	thi will i fe	Negative
15	13	damn give	Neutral
16	14	new rule if	Neutral
17	15	user_ment	Neutral
18	16	there wa a	Negative
19	17	from pass	Neutral
20	18	corona vir	Neutral
21	19	q to deal w	Negative
22	20	user_ment	Negative
23	21	thi thread	Neutral
24	22	url	Neutral
25	23	user_ment	Negative
26	24	me work c	Neutral
27	25	the corona	Neutral
28	26	but sinc kn	Negative
29	27	user_ment	Neutral

CONCLUSION:

The Models used for this task [Textblob & RoBERTa] are very powerful and efficient giving good classification results but there is some inaccuracy. The preprocessing techniques used for the models might be the one of the reasons for the inaccurate results. Overall, the results are satisfactory.