

## Sentiment Analysis on Covid-19 Vaccination Tweets using Naïve Bayes and LSTM

Ranjan Raj Aryal\*, Ankit Bhattarai

Department of IT and Computer Engineering, Cosmos College of Management and Technology, Lalitpur, Nepal

### Abstract

Social media is one platform where people share their opinions and views on different topics, services, or behaviors that happen around them. Since the COVID19 pandemic that started at the end of 2019, it has been a topic on which people express their sentiments. Recently, the COVID19 vaccination programs have got a lot of responses. In this paper, we have proposed two models: one based on the machine learning approach: Naive Bayes & the other based on deep learning: LSTM, whose goal is to know the sentiment of Asian region tweets towards the vaccine through sentiment analysis. The data were extracted with the help of Twitter API from March 23, 2021, till April 2, 2021. The extraction approach contains keywords with geocoding of some of the Asian countries, especially Nepal, India and Singapore. After collecting data, some preprocessing such as removing numbers, non-English & stop words, removing special characters, and hyperlinks were done. The polarity of tweets was assigned using the Textblob library. The tweets were classified into one of the three: positive, negative, or neutral. Now the data were preprocessed with the splitting of tweets into training & testing sets. Both the models were trained & tested using 10767 unique tweets. This experiment shows that a number of people in these three countries (Nepal, India and Singapore) have positive sentiment towards the vaccine and are taking the first dose of Covid19 vaccine. At last, the accuracy of the LSTM model was found to be 7% greater than that of the Naive Bayes-based model.

*Keywords: Twitter, sentiment analysis, Naive Bayes, RNN-LSTM.*

## 1. Introduction

In this 21st century, expressing one's feelings and sentiments in public has been regarded as the best communication method. Different forms of communicating media have been introduced with the growing advancement in the field of technology. Over the past decades, expressing one's sentiment in social networking

\*Corresponding author

Email: ranjanaryal@cosmoscollege.edu.np

Received: April 05, 2021; Revised: July 02, 2021; Accepted: July 02, 2021

has become a crucial factor. Social networking is used mainly for communication purposes & for information sharing, and this is not limited to a single field. Not only are social networking sites used for information sharing, but they have also become user's easy tools to express their feelings in the form of their sentiments. Being an easy tool, social networking has succeeded in the collection of enormous data. Social network analysis (SNA) is generally defined as mapping and measuring the relationships and flows between people, groups, organizations, computers, or other information processing entities [1]. In the field of analysis between the individual as well as group behavior, SNA has played a vital role and its use is not limited to this only. Large data can be analyzed with the use of SNA.

Sentiment analysis is used basically to detect the text's contextual polarity and determines whether a particular text is positive, negative, or neutral. It is also called opinion mining, as it tries to understand individuals' attitudes [2]. Learning-based and lexicon-based are the commonly used techniques for sentiment analysis. In the case of learning-based techniques, testing, and training the data set samples are used. Machines are supervised by using these training and testing techniques. In the case of Supervised learning techniques, we have used a Naive Bayes classifier in our experiment. The classifier in Naive Bayes is based on the Bayes theorem. Deep learning is a subset of the machine learning approach that learns in an unsupervised way from the data set that is unlabeled or unseen. RNN (Recurrent Neural Network) is one the deep learning approaches which is good for sentiment and video classification.

During the time of the pandemic, due to COVID-19, Physical gathering in public areas has been restricted from almost all countries of the globe. The emergence of the COVID19 pandemic & its mitigating means of Lockdown led the whole world to sit at their home. The flow of information along with the way of expressing their sentiments on social media increased drastically. This led the field of research to boom too. Researches based on social media analysis were effective as it helps to analyze where a society or a country is heading towards. Social media had played a great role in expressing one's feeling in the public and constituted a higher level of belongingness between people by providing a platform for their information exchange. In this regard, one of the most popular social media, Twitter, became the basic source of information. Twitter messages are well suited for knowledge discovery and it provides a free Application Programming Interface (API) which allows them to gather and analyze large data sets of tweets [3]. Through it, any researcher or developer can easily practice the large number of data gathered from the public tweets. In our experiment, we have analyzed the public sentiment over the COVID19 Scenario in Asian countries & the public responses over the Vaccination programs practiced in those countries.

A huge amount of data gets generated in tweets, Facebook & blog posts, which leads to knowing the user's opinion towards particular projects or subjects. Twitter data has a limitation of a number of characters to generate a text of about 140 characters [4]. Social networking has been using connected data between individuals, groups, organizations and other related systems. This information from different sources is represented as nodes [1]. These nodes are interconnected in many aspects. This interconnected information gives birth to so-called Social Networking. One experiment done on sentiment analysis using Tkinter which is supported by the Naive Bayes Algorithm provided a good accuracy of 83% for Twitter data [1]. Different machine learning approaches are being used by data analysts, engineers, and researchers. Support Vector Machine, Naive Bayes and Entropy methods are the commonly used techniques in the field of Sentiment Analysis. Among them, Naive Bayes has given the highest accuracy [5].

Besides these, a comparative study between Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Conditional Random Field (CRF) has also done, Result shows that Naive Bayes gave a good performance [6] by showing 0.96 precision, 0.92 recall, and 0.94 f-measure value. Twitter data can also be used to analyze sentiment by classifying the polarity of Tweets. The classification of these polarities can also be done using the Naive Bayes algorithm. The approach, based on a discriminative multinomial naïve Bayes (DMNB) method with 4-grams tokenizer, stemming, and term frequency-inverse document frequency (TF-IDF) techniques, was also used to analyze the polarity [8].

The machine learning algorithm may be regarded as a computational model that handles input data to achieve

a goal and deliver a specific outcome Machine learning is also used to create an algorithm that aids in the improvement of a system's performance output based on knowledge or experience [9]. Similarly, deep neural network algorithms show meaningful improvements over existing machine learning approaches in various domains like speech recognition & computer vision [10]. Sentiment classification techniques were used to classify US airline tweets based on sentiment polarity due to flight services as positive, negative and neutral connotations on six different US airlines. The word embedding models (Word2Vec, Glove) detect sentiment polarity using deep learning methods and LSTM is a powerful classifier for sentiment analysis [11]. Deep learning-based sentiment analysis financial news predicts the change in price earlier based on the polarity of data. It uses LSTM/RNN and CNN methodologies for better improvement [12]. In paper [13], using the Twitter datasets of IMDB, Amazon & Airline the Naive Bayes & RNN- LSTM were performed. For the Naive Bayes and RNN-algorithm, the accuracies were 81% and 87% for the IMDB dataset, 69% and 88% for the IMDB dataset, 60% and 93% for the Airline dataset respectively. Also, the resulting experiment done in paper [15] showed the Naive Bayes result is 96.6%, which is higher than other classifiers, Nearest Neighbor (k-NN) and support vector machine (SVM). So, in our experiment, we have chosen these two methods i.e. Naive Bayes and LSTM as classifiers.

## 2. Materials And Methods

The proposed system was designed by applying major five steps: data gathering, pre-processing, feature extraction, feature selection and classifying methods. These steps are further elaborated and explained as shown in figure 1.

### 2.1 Data Gathering

We have collected twitter data using keywords and without using any keywords and also by geo location. Used hashtags like #CovidVaccine, #VaccinesWork, #vaccination #VaccinesSaveLives, #AstraZeneca and #COVAX and for geo location we have used longitude, latitude & radius for different countries like Nepal, India, Singapore. To collect these data, first we need to make a twitter account and using this account we need to register for the developer account. After that, we requested the twitter developer team along with our valid reason for our academic research, to provide access to the twitter API. Validation of our research work proposal by the twitter developer team leads to access over the twitter API. To access the twitter API, the twitter developer team provided us the API key, API Secret Key and Bearer token. Using this information, we can gather twitter data for our research.

### 2.2 Pre-processing

After collection of data from twitter, pre-processing is done. Removal of unnecessary data which are not suitable for our experiment is considered as Pre-processing.

- Tokenization

The tokenization process removes the hashtags, re-tweets marks and hyperlinks from the keywords that we have used in our experiment and converts the text into tokens. Removal of numbers (like 1, 2, 3....) and special characters (like @) are done under this process. A Natural Language Processing package has also been installed for the Tokenization process.

- Removal of non-English words

Our project is related to English keywords. So, we removed all non-English words from the data that we have collected from Twitter.

- Emotion Replacement

Users sentiments shared using emotional stickers also need to be replaced by similar synonyms. For it, polarity has been given for each emotional sticker.

- Removal of Stop words

Words that have no meaning even if they are English words on the experiment have to be removed. Such words may be articles (a, an, the..).

## 2.3 Feature Extracting

The selection of necessary and useful words from the different tweets of the user is defined as Feature Extraction. In the English language, parts of speech can be used to express one's sentiments in their tweets. They also indicate the polarity of the user's sentiment over the topic they are expressing their views. There are three types of features: unigram, bigram and n-gram.

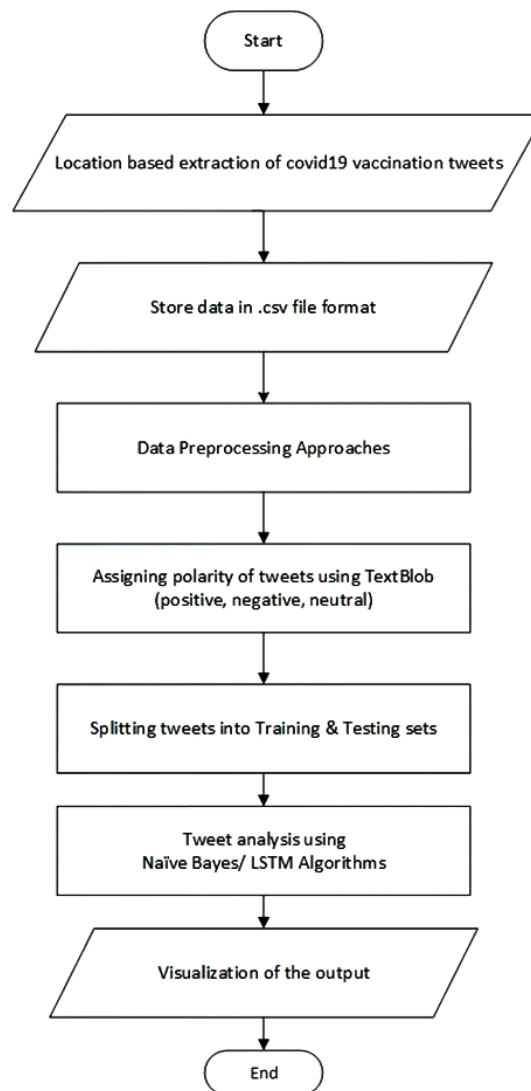


Figure 1: Proposed Twitter's sentiment analysis steps

## 2.4 Feature Selection

Feature selection is done to classify the attributes according to the exact meaning delivered by those attributes. A correct classification describes its accuracy. For it, different feature selection techniques (Natural Language Processing, statistical-based, Clustering-based, Hybrid based) can be used. In our experiment, we have used Natural Language Processing classification technique.

## 2.5 Classifying Methods

There are various classifying methods. The most commonly used machine learning classifying methods are:

### Naïve Bayes classifier

The text classification process can be done by using the Naive Bayes Classifier. The basic principle of the Naive Bayes Classifier technique is derived from the Bayes theorem. It is one of the simplest and most powerful classification algorithms which includes a dataset for high-dimensional training that helps to create fast machine learning models that can make predictions quickly [4]. It is mostly used for text analysis, sentiments analysis and classifying articles.

Bayes theorem can be explained by the following relation [4].

$$\text{Prob}(A/B) = (\text{Prob}(B/A) \times \text{Prob}(A)) / \text{Prob}(B)$$

Where,

- $\text{Prob}(A | B)$  is known as (posterior probability) of hypothesis A which occurs when some condition is already provided.
- $\text{Prob}(B | A)$  It is also called likelihood probability. It is the probability of evidence E when we presume that the given hypothesis is true.
- $\text{Prob}(A)$  is the prior known probability of A and does not include any condition.
- $\text{Prob}(B)$  is the prior known probability of A and does not include any condition.

### LSTM (Long Short Term Memory)

LSTM is a deep learning approach and a type of Recurrent Neural Network architecture which helps to remember previously read values over time. LSTMs contain three gates: input, forget & output gate [7]. As shown in Figure 2, the input gate controls the input of new information to the memory, followed by the forget gate which controls how long certain values are held in memory. At last, the output gate controls how much the value stored in memory affects the output activation of the block. [7]

Text analysis can be carried out using LSTMs as it helps to remember previously read values effectively. The LSTM will have a better understanding of the tweets analysis. The tweet may have changed sentiment such as "I hated reading books until I read Muna Madan", which may be easily dealt with by LSTM.

## 3. Results and Discussion

### 3.1 Experimental Setup & parameters

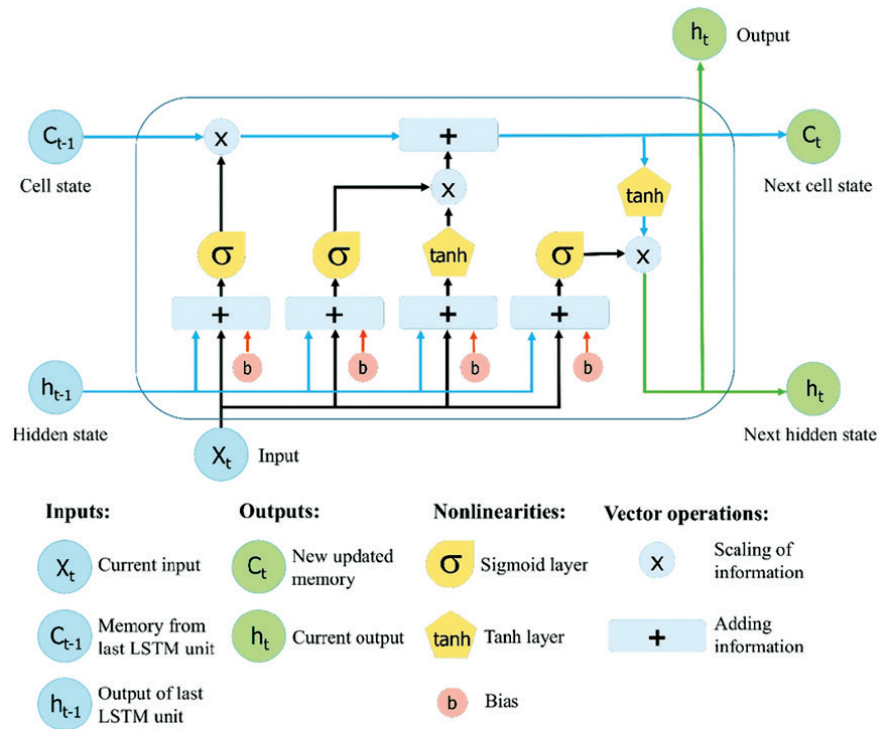


Figure 2: Basic Diagram of LSTM [10]

For this observation, we generated training sets of about 7537 tweets & a testing set of about 3230 tweets. The tweets were preprocessed using data preprocessing approaches. For tweets labeled with the help of the Textblob library we were able to assign each tweet a polarity as positive, negative or neutral as shown in figure 3.

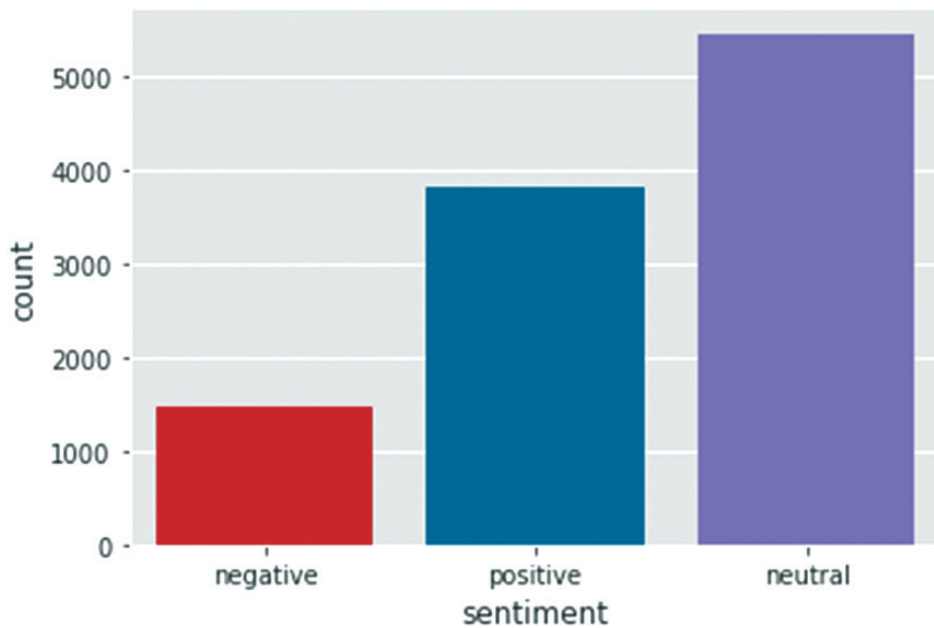


Figure 3: Graph of Labeled Tweets



Among 10767 tweets, 1478, 3822 & 5467 tweets were labelled as negative, positive and neutral tweets respectively. Some of the tweets along with its polarity or sentiment are shown in Table 1.

Table 1: Examples of Tweets with their polarities

Tweets	Sentiment
AstraZeneca COVID vaccine found 76% effective in updated US trial results	Positive
At least three members of polio vaccination team, all of them women, were shot dead by unknown gunmen	Negative
India to increase interval between doses of AstraZeneca shot	Neutral

In the case of the Naive Bayes approach the test size was about 30% of the total dataset and the random state was set to 0. On the other hand, the LSTM uses the following parameters as shown in Table 2.

Table 2: Parameter Selected for them LSTM model

Embedding Dimension	32
Epoch	20
Batch Size	64
Dropout	0.2

### 3.2 Results

The tests were carried out 5 times & the average accuracy was taken into consideration. The two approaches: Naive Bayes & LSTM model accuracy was observed as in Table 3.

Table 3: Average Accuracy Observation on Models

Models	Average Accuracy on 5 tests
Naive Bayes	77.25%
LSTM	84.13%

From Fig. 4, we can clearly see some of the words that are used in the tweets. Some of the words that are visualized apart from keywords are India, first dose, vaccine dose, corona virus among others. This also shows that the people are receiving their first shot of vaccine in some Asian countries, i.e. Nepal, India and Singapore during the time frame that we worked on.

The word clouds of tweets show that people are receiving their first dose of covid19 vaccination. Fig 5 shows that they are quite positive towards it & are also expressing their concern as the vaccine may have its side effects like blood clot and some of the words also suggest that the vaccine is in the trial phase, see Fig 6. The other concern seems to be about the vaccine center due to lockdowns at different places & inadequate ways to distribute vaccines, see Fig 7.







was made using the Naive Bayes algorithm and LSTM. This model can be used to analyze the sentiments related to the Covid19 vaccine in the future. Also, their accuracy has been calculated separately. In this experiment LSTM and Naive Bayes showed an average accuracy of 84.13% & 77.25% on five tests respectively. The LSTM approach shows about 7% more efficiency than the Naive Bayes Approach. This concludes that the deep learning algorithm is more efficient to classify any given tweet into positive, negative & neutral. The visualized most used words clearly show that a number of people in mainly these three countries (Nepal, India and Singapore) are taking the first dose of Covid19 vaccine. It shows that the public are aware about the Covid19 vaccine and their sentiments are positive towards it.

The data were extracted from a given time frame so the availability of good tweets on that timeline needs to be taken under consideration. Also, the Textblob library is sometimes slow and does not take integrated word vectors into account which may result in the inaccuracy of assigning polarity to tweets.

It is only applicable for the English language tweets. With the widespread use of twitter, native languages like Hindi, Nepali, etc. are also popular in expressing sentiments. So, other languages should also be taken into consideration in future works. Apart from that the deep learning algorithm works well with the large datasets, so in order to make the model more generalized dataset size can be increased. An alternative to assigning polarity to tweets may be taken into consideration. CNN- LSTM & Bidirectional LSTM network (Bi-LSTM) may be used for further enhancement.

## Appendix

### Extracting Tweets Using Twitter API

#### **#Importing Libraries**

```
import tweepy
from textblob import TextBlob
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from wordcloud import WordCloud
import json
from collections import Counter
#Getting authorization
consumer_key = "XXXXXXXXXXXXXXXXX"
consumer_key_secret = "XXXXXXXXXXXXXXXXX"
access_token = "XXXXXXXXXXXXXXXXX"
access_token_secret = "XXXXXXXXXXXXXXXXX"
auth = tweepy.OAuthHandler(consumer_key, consumer_key_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True)
query = 'vaccination OR VaccinesSaveLives'
max_tweets = 2000
date_since = "2021-03-28"
date_until = "2021-03-29"
searched_tweets = [status for status in tweepy.Cursor(api.search, q=query, geocode="28.3949, 84.1240,, 2000km",
since = date_since, until = date_until).items(max_tweets)]
```

```

my_list_of_dicts = []
for each_json_tweet in searched_tweets:
    my_list_of_dicts.append(each_json_tweet._json)
with open('tweet_json_Data.txt', 'w') as file:
    file.write(json.dumps(my_list_of_dicts, indent=4))
my_demo_list = []
with open('tweet_json_Data.txt', encoding='utf-8') as json_file:
    all_data = json.load(json_file)
    for each_dictionary in all_data:
        tweet_id = each_dictionary['id']
        text = each_dictionary['text']
        favorite_count = each_dictionary['favorite_count']
        retweet_count = each_dictionary['retweet_count']
        created_at = each_dictionary['created_at']
        tweets_location = each_dictionary['text']
        my_demo_list.append({'tweet_id': str(tweet_id),
                            'text': str(text),
                            'favorite_count': int(favorite_count),
                            'retweet_count': int(retweet_count),
                            'created_at': created_at,
                            'tweet_location': tweet.user.location,
                            })
    tweet_dataset = pd.DataFrame(my_demo_list, columns =
                                ['tweet_id', 'text',
                                'favorite_count', 'retweet_count',
                                'created_at', 'tweet_location'])

```

#### **#Writing tweet dataset to csv file for future reference**

```
tweet_dataset.to_csv('2_Loc_Singapore_2000_tweet_data_28_29.csv')
```

### **Finding Polarity of tweets**

#### **# Import the libraries**

```
import csv
from textblob import TextBlob
import sys
```

#### **# Do some version specific stuff**

```
if sys.version[0] == '3':
    from importlib import reload
    sntTweets = csv.writer(open("newPolarity.csv", "w", encoding='utf-8', newline=''))
```

```
if sys.version[0] == '2':
    reload(sys)
    sys.setdefaultencoding("utf-8")
    sntTweets = csv.writer(open("newPolarity.csv", "w"))
```

```
alltweets = csv.reader(open("outputFile.csv", 'r', encoding='utf-8'))
```

```
for row in alltweets:
```

```
    blob = TextBlob(row[2])
    print (blob.sentiment.polarity)
    if blob.sentiment.polarity > 0:
```

```

sntTweets.writerow([row[0], row[1], row[2], row[3], row[4], row[5],row[6], blob.sentiment.polarity,
"positive"])
elifblob.sentiment.polarity< 0:
sntTweets.writerow([row[0], row[1], row[2], row[3],row[4], row[5],row[6], blob.sentiment.polarity,
"negative"])
elifblob.sentiment.polarity == 0.0:
sntTweets.writerow([row[0], row[1], row[2], row[3],row[4], row[5],row[6], blob.sentiment.polarity, "neutral"])

```

## Training the model

### Naive Bayes

```
class TweetNBClassifier(object):
```

```

def __init__(self, df_train):
self.df_train = df_train
self.df_pos = df_train.copy()[df_train.sentiment == 'positive']
self.df_neg = df_train.copy()[df_train.sentiment == 'negative']
self.df_neu = df_train.copy()[df_train.sentiment == 'neutral']
def fit(self):
Pr_pos = df_pos.shape[0]/self.df_train.shape[0]
Pr_neg = df_neg.shape[0]/self.df_train.shape[0]
Pr_neu = df_neu.shape[0]/self.df_train.shape[0]
self.Prior = (Pr_pos, Pr_neg, Pr_neu)
self.pos_words = ' '.join(self.df_pos['text']).tolist().split()
self.neg_words = ' '.join(self.df_neg['text']).tolist().split()
self.neu_words = ' '.join(self.df_neu['text']).tolist().split()
all_words = ' '.join(self.df_train['text']).tolist().split()
self.vocab = len(Counter(all_words))
wc_pos = len(' '.join(self.df_pos['text']).tolist().split())
wc_neg = len(' '.join(self.df_neg['text']).tolist().split())
wc_neu = len(' '.join(self.df_neu['text']).tolist().split())
self.word_count = (wc_pos, wc_neg, wc_neu)
return self
def predict(self, df_test):
class_choice = ['positive', 'negative', 'neutral']
classification = []
for tweet in df_test['text']:
text = tweet.split()
val_pos = np.array([])
val_neg = np.array([])
val_neu = np.array([])
for word in text:
tmp_pos = np.log((self.pos_words.count(word)+1)/(self.word_count[0]+self.vocab))
tmp_neg = np.log((self.neg_words.count(word)+1)/(self.word_count[1]+self.vocab))
tmp_neu = np.log((self.neu_words.count(word)+1)/(self.word_count[2]+self.vocab))
val_pos = np.append(val_pos, tmp_pos)
val_neg = np.append(val_neg, tmp_neg)
val_neu = np.append(val_neu, tmp_neu)
val_pos = np.log(self.Prior[0]) + np.sum(val_pos)
val_neg = np.log(self.Prior[1]) + np.sum(val_neg)

```

```

    val_neu = np.log(self.Prior[2]) + np.sum(val_neu)
    probability = (val_pos, val_neg, val_neu)
    classification.append(class_choice[np.argmax(probability)])
    return classification
def score(self, feature, target):
    compare = []
    for i in range(0,len(feature)):
        if feature[i] == target[i]:
            tmp = 'correct'
            compare.append(tmp)
        else:
            tmp = 'incorrect'
            compare.append(tmp)
    r = Counter(compare)
    accuracy = r['correct']/(r['correct']+r['incorrect'])
    return accuracy
tnb = TweetNBClassifier(df_train)
tnb = tnb.fit()
predict = tnb.predict(df_test)
score = tnb.score(predict,df_test.sentiment.tolist())
print(score)

```

### **LSTM (Long Short Term Memory)**

```

EMBEDDING_DIM = 32
model = Sequential()
model.add(Embedding(30000, EMBEDDING_DIM, input_length=train_padding.shape[1]))
model.add(Dropout(0.2))
model.add(LSTM(128, dropout=0.2,return_sequences=True))
model.add(LSTM(64, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(32))
model.add(Dropout(0.2))
model.add(Dense(3, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())

epochs = 20
batch_size = 64
model.fit(train_padding, y,
          epochs=epochs,
          batch_size=batch_size,
          validation_split=0.3)

```

### **Conflict of Interest**

Not declared by the authors.



## References

1. T. Surya Gunawan, N. Aleah Jehan Abdullah, M. Kartiwi and E. Ihsanto, "Social Network Analysis using Python Data Mining," 2020 8th International Conference on Cyber and IT Service Management (CITSM), Pangkal, Indonesia, 2020, pp. 1-6, doi: 10.1109/CITSM50537.2020.9268866..
2. Devika, M.D. & C, Sunitha & Ganesh, Amal. (2016). Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*. 87. 44-49. 10.1016/j.procs.2016.05.124.
3. Wolny, Wieslaw. (2016). Knowledge Gained from Twitter Data.
4. M. Neethu and R. Rajasree, "Sentiment analysis in twitter using machinelearning techniques," in *Computing, Communications and NetworkingTechnologies (ICCCNT), 2013 Fourth International Conference on*. IEEE, 2013, pp. 1-5.
5. L. Mandloi and R. Patel, "Twitter Sentiments Analysis Using Machine Learning Methods," 2020 International Conference for Emerging Technology
6. R. I. Permatasari, M. A. Fauzi, P. P. Adikara and E. D. L. Sari, "Twitter Sentiment Analysis of Movie Reviews using Ensemble Features Based Naïve Bayes," 2018 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, Indonesia, 2018, pp. 92-95, doi: 10.1109/SIET.2018.8693195.
7. Sundermeyer, Martin / Schlüter, Ralf / Ney, Hermann (2012): "LSTM neural networks for language modeling", In *INTERSPEECH-2012*, 194-197.
8. H. AlSalman, "An Improved Approach for Sentiment Analysis of Arabic Tweets in Twitter Social Media," 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2020, pp. 1-4, doi: 10.1109/ICCAIS48893.2020.9096850.
9. T. Shaikh and D. Deshpande, "A review on opinion mining and sentimentanalysis," in *IJCA Proceedings on Nat onal Conference on Recent Trends in Computer Science and Information Technology*, no. 2, 2016, pp. 6-9.
10. M. D. Bah, A. Hafiane, and R. Canals, "Deep learning with unsupervised data labeling for weeds detection on uav images," arXiv preprint arXiv:1805.12395, 2018.
11. R. Monika, S. Deivalakshmi and B. Janet, "Sentiment Analysis of US Airlines Tweets Using LSTM/RNN," 2019 *IEEE 9th International Conference on Advanced Computing (IACC)*, Tiruchirappalli, India, 2019, pp. 92-95, doi: 10.1109/IACC48062.2019.8971592.
12. Souma, Wataru&Vodenska, Irena & Aoyama, Hideaki. (2019). Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science*. 2. 10.1007/s42001-019-00035-x.
13. Y. M. Wazery, H. S. Mohammed and E. H. Houssein, "Twitter Sentiment Analysis using Deep Neural Network," 2018 *14th International Computer Engineering Conference (ICENCO)*, Cairo, Egypt, 2018, pp. 177-182, doi: 10.1109/ICENCO.2018.8636119.
14. Le, Xuan Hien & Ho, Hung & Lee, Giha& Jung, Sungho. (2019). Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water*. 11. 1387. 10.3390/w11071387.
15. Abo, Mohamed & Shah, Nordiana & Balakrishnan, Vimala & Abdelaziz, Ahmed. (2018). Sentiment analysis algorithms: evaluation performance of the Arabic and English language. *IEEE Expert*. 10.1109/ICCCEEE.2018.8515844.