SENTIMENT CLASSIFICATION AND ANALYSIS OF TWEETS RELATED TO ONLINE EDUCATION DURING COVID-19

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

MASTER OF SCIENCE

IN

MATHEMATICS

Submitted by:

LAKSHAY SAINI

(2K21/MSCMAT/28)

PRACHI VERMA

(2K21/MSCMAT/34)

Under the supervision of

Ms. SUMEDHA SENIARAY

DEPARTMENT OF APPLIED MATHEMATICS

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 202

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi- 110042

CANDIDATE'S DECLARATION

We, Lakshay Saini, 2K21/MSCMAT/228 & Prachi Verma, 2K21/MSCMAT/34 students of M.Sc. Mathematics, hereby declare that the project Dissertation titled "Sentiment Classification and Analysis on Distance Learning " which is submitted by us to the Department of Applied Mathematics Delhi Technological University, Delhi in partial ful fillment of the requirement for the award of the degree of Master of Science, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: New Delhi — Lakshay Saini

Date: 22 May, 2023

Prachi Verma

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi- 110042

CERTIFICATE

I hereby certify that the Project Dissertation titled " SENTIMENT CLASSIFICATION AND ANALYSIS ON DISTANCE LEARNING " which is submitted by Lakshay Saini, Roll No. 2K21/MSCMAT/28 & Prachi Verma, Roll No. 2K21/MSCMAT/34, Department of Applied Mathematics, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Science, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: New Delhi

Date:

Ms. Sumedha Seniaray
SUPERVISOR
Assistant Professor
Deptt. Of Applied Mathematics
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi- 110042

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi- 110042

ACKNOWLEDGEMENT

We would like to express our profound gratitude to Dr. S. Sivaprasad Kumar HOD, of Department of Applied Mathematics. We would like to express our special thanks to our mentor Ms. Sumedha Seniaray for her time and efforts she provided throughout the year. Your useful advice and suggestions were really helpful to us during the project's completion. In this aspect, we are eternally grateful to you. We would like to acknowledge that this project was completed entirely by us and not by someone else.

Lakshay Saini

Prachi Verma

ABSTRACT

The COVID-19 outbreak impacted drastically to education and most of the educational institutions started preferring online education for students. However, after the settlement of the pandemic there is uncertainty among people about whether they should prefer online education for furthermore or start in offline mode to make it more interactive, so this paper is about an analysis of people's sentiments and emotions through Tweets about COVID-19 Education. This paper aims to study the reaction of people around the world toward online education during COVID-19. This study is conducted on the basis of the responses of students, teachers, parents, college professors etc. We started with labeling the data into three sentiments namely positive, neutral, and negative and for validation then we used Machine learning (ML) classifiers namely, Logistic regression, Decision tree, Random forest, Multilayer Perceptron (MLP), Naïve Bayes, Support vector machine (SVM), K-nearest neighbors (KNN), and XG-Boost. Then we performed emotion detection by considering 5 emotions namely happy, surprise, sad, fear, and angry and for validation we used ML classifiers. After applying all these ML approaches, the XG Boost ML classifier achieved highest accuracy of 94% in classifying the tweets as positive, neutral, or negative, and 96% accuracy in classifying the tweets as happy, surprised, sad, fearful, or angry.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML (Machine Learning)

LR (Logistic Regression)

DT (Decision Tree)

RF (Random Forest)

NB (Naïve Bayes)

KNN (K-nearest Neighbors)

MLP (Multilayer Perceptron)

SVM (Support Vector Machine)

XG Boost (Extreme Gradient Boosting

CHAPTER 1

INTRODUCTION

1.1 ABOUT TWITTER

Users can post status updates (referred to as "tweets") using the well-known micro-blogging site Twitter. These tweets are filled with several examples of how people express their likes and dislikes as well as their contributions to various issues. Systems that attempt to recognize and extract opinions from the text are created using sentiment analysis.

1.2 ONLINE EDUCATION

Humans have feelings that correspond to internal statuses and behavioral patterns. Sentiment analysis is the method of determining the different sentiments or emotions in the data, such as a social media post, review, or comment. It involves using techniques such as NLP that stands for Natural Language Processing to categorize the text as positive, negative, or neutral, basically into sentiments.

Emotion detection, on the other hand, goes a step further and aims to identify specific emotions like happiness, sad, fear, anger, and surprise presented in data or speech. Emotion detection methods can be utilized in different areas like twitter or any other social network monitoring, and market research.

As for online education, there are several reasons why it is needed:

1. Accessibility: Online education breaks down geographical barriers and allows people from different locations to access educational resources and opportunities.
2. Flexibility: Online education offers flexible learning options. The availability of various books of their course and lectures at any time enables students to complete their studies at their own speed. Working professionals or people with other responsibilities who might not be able to attend regular classes will especially benefit from this flexibility.
3. A large selection of course options: Online education offers a variety of courses and programmes, some of which might not be accessible locally. Students have the opportunity to choose from various disciplines and pursue specialized knowledge in their areas of interest.
4. Cost-effective: As compared to traditional in-person education online education is cost effective. It eliminates the need for commuting, relocation, and sometimes even textbooks, making education more accessible to a broader range of learners.
5. Technological advancements: With the advancement of technology, online education platforms offer learning experiences in which many students get engaged. These resources upgrade the learning process and provide a more immersive educational experience.

6. Lifelong learning: Online education promotes lifelong learning by offering continuing education and professional development opportunities. Individuals can update their skills and knowledge throughout their careers, keeping up with the evolving demands of the job market.

Nowadays, most of the institutes are giving predilection to online education and some of the people are accepting it contentedly and others used to give some different reactions, this motivated us to do sentiment and emotional study to get to know the conception of different communities. Therefore, we used social media handles namely *Twitter* because connectivity through social media is flourishing nowadays. People used to share their ideas and suggestions on such platforms. In this study, we will focus on how events in such forums generate collective responses. So, for this purpose, we will be extracting the data related to reactions towards online education, especially during the outbreak of the pandemic i.e., COVID-19 from Twitter using API.

1.3 OBJECTIVE

In this paper, we have mainly focused on online education tweets, not just to examine the performance of various ML classifiers but also to infer genuineness so that different institutions can see what people wants whether they are satisfied with online education or want to continue with offline education. This will assist them to take an unerring pronouncement whether they should continue with online education or not.

The Objective of our work is, to check the effect of the combination of TF-IDF and n-gram technique for feature extraction from Twitter online education data on different ML classifiers and analyzation of how different communities reacted towards online education during COVID-19 that help future generations to conclude whether they prefer online education or offline.

CHAPTER 2

RELATED WORK

Baboo, S. S., et al. [1] collected the Twitter dataset that contains 45,000 tweets regarding corona virus and performed sentiment and emotion of tweets by Logistic regression, Random forest, Support vector machine, Naïve bayes, and stochastic gradient boosting (SGD) machine learning algorithms and observed that SVM achieved the highest accuracy of 95%. Jalil, Z., Abbasi, et al. in [2] classified COVID-19 tweets into sentiments (positive, negative, and neutral) and for feature extraction authors used methods that learn features automatically without any human interference and proposed DistilBERT model which achieved 96.66% of accuracy. [3] represents the classification of the Amazon reviews dataset by convolutional neural network into sentiments and achieved 90% of accuracy.

In [4], study used twitter data set of lockdown and performed sentiment study by using multinomial logistic regression and has extracted features using the TF-ID technique for model training then for validation purpose author used a decision tree, naïve bayes, and KNN and observed multinomial logistic regression gives better accuracy of 94%. In [5], the authors had taken short text data from Twitter and performed emotion detection, and used LSTM, XG Boost, SVM-SGD, naïve bayes, random forest, and decision tree and observed that LSTM achieved higher accuracy of 91.9%. In [6], the author has used the location of a tweet in sentimental analysis, the author has used CNN, BiLSTM, and Vader library and observed BiLSTM achieved 87.69% of accuracy and after combining location of tweet to the tweet made, author observed the fall in overall accuracy of classifiers. Chowanda, et al. in [7] used Naive Bayes, Random Forest, Artificial Neural Network, Generalised Linear Model (GLM), Fast-Large Margin, Decision Tree, and Support Vector Machine (SVM) machine learning techniques for text-based emotion recognition and observed GLM provides the best accuracy score (90.2%).

In [8], the author has done emotion recognition from text stories by collecting 144,701 tweets and gave emotional hash tag to each tweet and used CNN model for emotion classification. [9] represents sentiment and emotion study of tweets by DNN, LSTM, GRU, and BiLSTM and observed that For sentiment analysis LSTM with Fast Text as pre-defined word embedding and for emotion analysis LSTM with GloVeTwitter as pre-defined word embedding achieved higher accuracy. Balabantaray, et al. in [10], done labeling of data regarding posts on social media manually followed by analysis of the data linguistically. In [11], [12] a technique to classify tweets into six emotions is presented. In [13], the authors labeled the data manually and data was from children's stories. In [14], the discrete emotion theory is presented regarding emotion categorization, which was expressed by Charles Darwin [15] for the first time.

CHAPTER 3

METHODOLOGY

We first collected the tweets through the Twitter API, a total of approximately 100k tweets were collected, followed by the pre-processing of data by cleaning to remove special characters, URLs, unwanted information, and duplicate tweets. The data was then labeled into three different sentiments that are positive, neutral, and negative with the help of the Text Blob and Vader algorithm followed by training the ML classifiers and predicting the performance of all classifiers. Then, labeled the data set with 5 different emotions namely happy, angry, surprise, fear, and sad with the help of the text2emotion python library, followed by training the ML classifiers and predicting the performance of all classifiers in the terms of f1 score, recall, precision, and accuracy. Figure 3.1.1 represents the methodology of this paper.

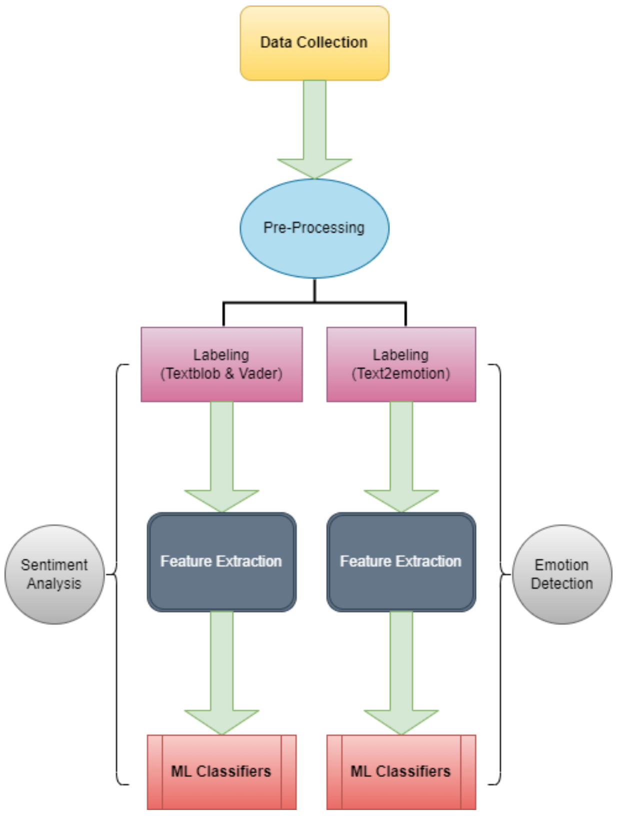


Figure 3.1.1: Proposed Framework

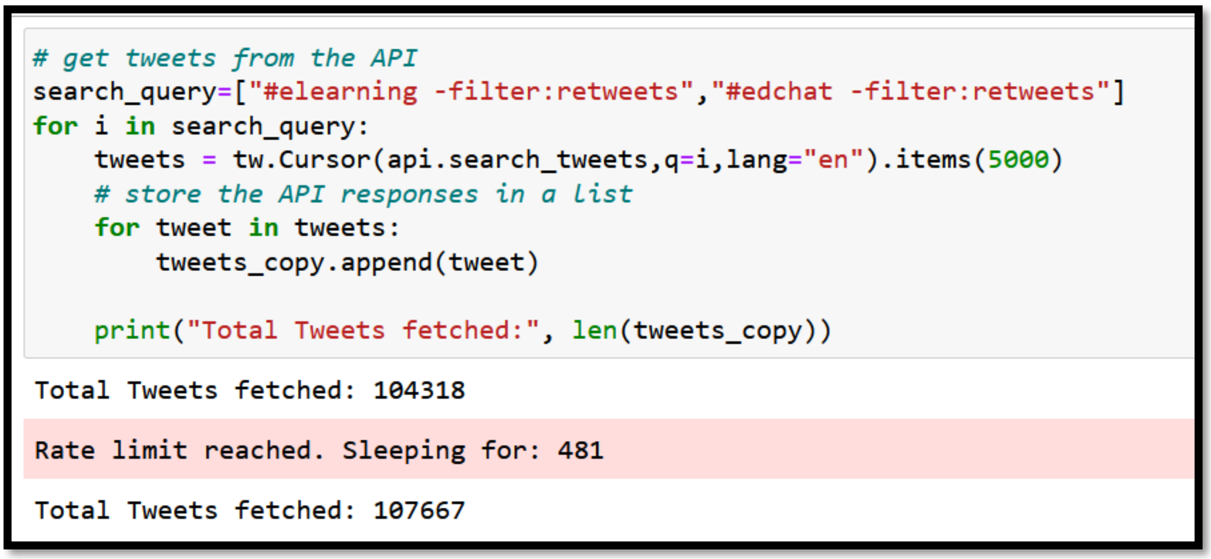
3.1 DATA COLLECTION AND PREPROCESSING

We used Twitter API and the tweepy library in python to collect the data. A total of approximately 100k tweets were collected to perform the analysis. A sample of the population was taken from tweets that were made available to the public on Twitter in order to examine the opinions of students and educators. Following data gathering and filtering, the gathered tweets were examined for emotional expressions.

The tweets have been collected in two phases

A) By hashtag search
B) By keyword search

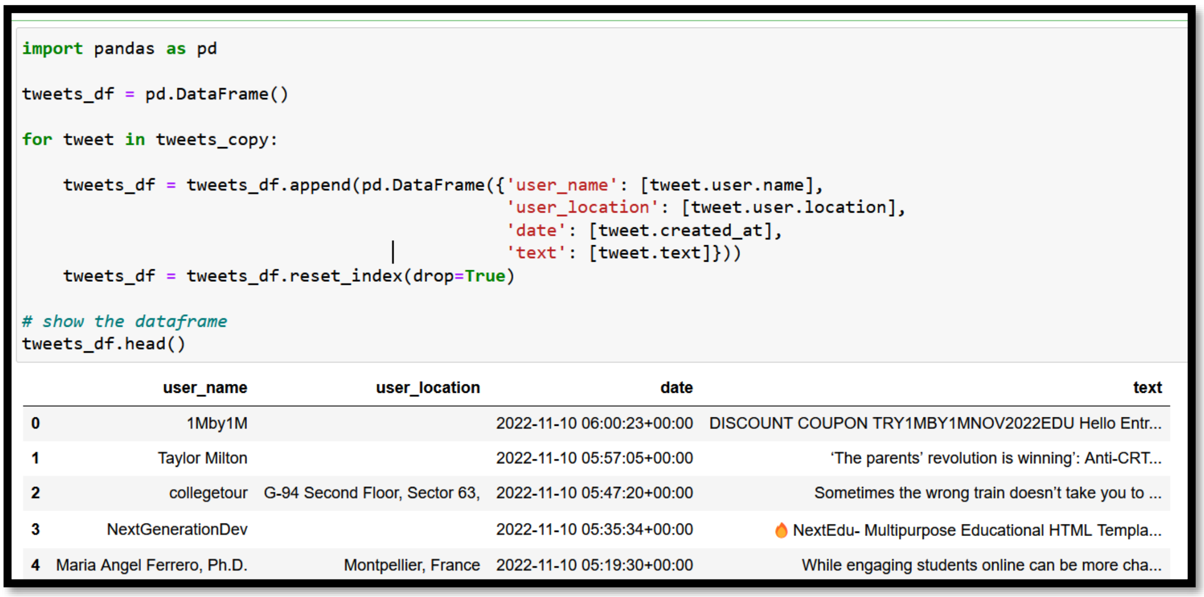
A twitter developer account was created to access the tweets using the hashtags & keywords. Figure 3.1.2 is the snap of python code in jupyter notebook for collecting the tweets in the form of list.



```
# get tweets from the API
search_query=["#elearning -filter:retweets","#edchat -filter:retweets"]
for i in search_query:
    tweets = tw.Cursor(api.search_tweets,q=i,lang="en").items(5000)
    # store the API responses in a list
    for tweet in tweets:
        tweets_copy.append(tweet)

    print("Total Tweets fetched:", len(tweets_copy))
```

```
Total Tweets fetched: 104318

Rate limit reached. Sleeping for: 481

Total Tweets fetched: 107667
```

Figure 3.1.2: Code for collecting tweets

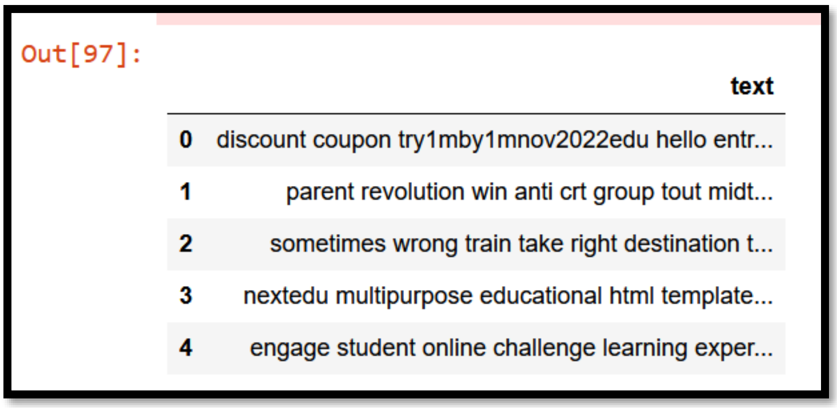


```
import pandas as pd

tweets_df = pd.DataFrame()

for tweet in tweets_copy:

    tweets_df = tweets_df.append(pd.DataFrame({'user_name': [tweet.user.name],
                                               'user_location': [tweet.user.location],
                                               'date': [tweet.created_at],
                                               'text': [tweet.text]}))
    tweets_df = tweets_df.reset_index(drop=True)

# show the dataframe
tweets_df.head()
```

	user_name	user_location	date	text
0	1Mby1M		2022-11-10 06:00:23+00:00	DISCOUNT COUPON TRY1MBY1MNOV2022EDU Hello Entr...
1	Taylor Milton		2022-11-10 05:57:05+00:00	'The parents' revolution is winning': Anti-CRT...
2	collegetour	G-94 Second Floor, Sector 63,	2022-11-10 05:47:20+00:00	Sometimes the wrong train doesn't take you to ...
3	NextGenerationDev		2022-11-10 05:35:34+00:00	🔥 NextEdu- Multipurpose Educational HTML Templa...
4	Maria Angel Ferrero, Ph.D.	Montpellier, France	2022-11-10 05:19:30+00:00	While engaging students online can be more cha...

Figure 3.1.3: Code for saving the data into data frame

After collecting the tweets, next step is to covert the list into the data frame for which we used panda's library and we have converted it into the CSV file which have three columns namely 'User name', 'User location', 'Created at', 'text' as shown in figure 3.1.3 which contains the name of the person who tweeted, location of the person who tweeted, time of tweeting and the tweet itself respectively. As the tweets contain a lot of unwanted information which may cause problems while performing sentiment analysis. In the next step the tweets were cleaned so as to remove special characters, URLs, unwanted information, duplicate tweets, etc. and then we converted the processed data to a CSV file and further into a data frame to perform sentiment analysis. The processed data has been shown in figure 3.1.4.



```
Out[97]:
                                               text
   0  discount coupon try1mby1mnov2022edu hello entr...
   1         parent revolution win anti crt group tout midt...
   2        sometimes wrong train take right destination t...
   3    nextedu multipurpose educational html template...
   4     engage student online challenge learning exper...
```

Figure 3.1.4: Processed data

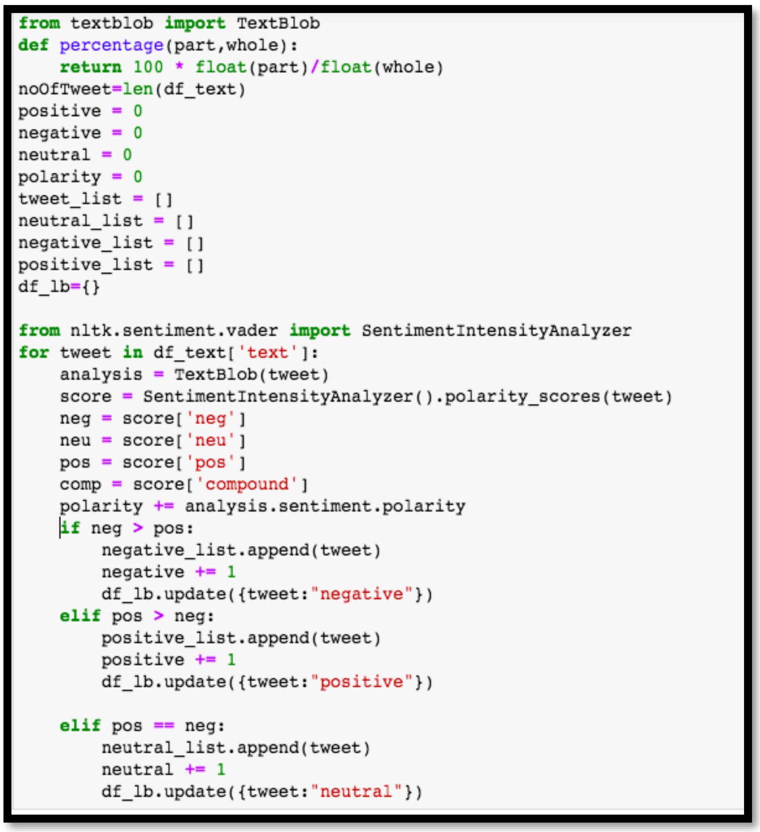
3.2 SENTIMENT ANALYSIS

3.2.1 Labeling of data

Sentiment analysis can be used to analyze students' feedback and sentiments regarding online courses, modules, or specific assignments. Teachers can pinpoint areas for improvement, address student complaints, and make required adjustments to improve the learning experience by gathering and analysing this input. Sentiment analysis combined with machine learning algorithms can create adaptive learning systems that tailor educational content and activities to individual students' needs and emotional states. By continuously analyzing sentiments, the system can adapt and provide personalized learning paths, resources, and support.

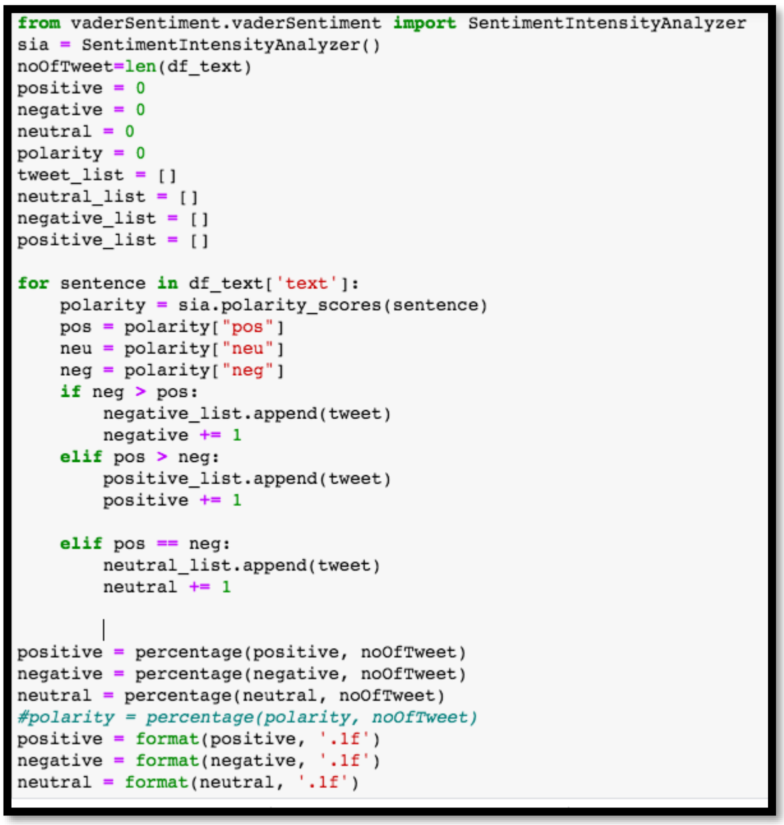
If we give large amount of raw data to classifiers, it will provide abrupt result, so it is necessary to do labeling of data. Data labeling process work in the following order: Data collection, Data tagging, and Quality assurance. Data collection includes collecting raw data followed by tagging the data by various data labeling approaches such as Text Blob and Vader (discussed in this section) and at last quality assurance is done which means how correctly the tags are tagged for the particular data point where one can use Consensus algorithm.

For labeling our data set into different sentiments, we used Text blob and Vader for labeling the data. Text blob and Vader are data labeling approaches for Natural Language Preprocessing (NLP) data. Text blob is a library in python for textual data processing (NLP) and when any text is given to it as an input, it returns its polarity score lies in [-1,1] (where 1 tells data is positive, 0 tells data is neutral, and -1 tells data is neutral) and subjectivity which measures the amount of factual information and personal opinion presented in the text. Valence aware dictionary and sentiment reasoning in short known as Vader is also a python library that is specifically used for social media sentiments that computes the positive, negative, and compound scores to find out the polarity of the textual data.



```python
from textblob import TextBlob
def percentage(part,whole):
    return 100 * float(part)/float(whole)
noOfTweet=len(df_text)
positive = 0
negative = 0
neutral = 0
polarity = 0
tweet_list = []
neutral_list = []
negative_list = []
positive_list = []
df_lb={}

from nltk.sentiment.vader import SentimentIntensityAnalyzer
for tweet in df_text['text']:
    analysis = TextBlob(tweet)
    score = SentimentIntensityAnalyzer().polarity_scores(tweet)
    neg = score['neg']
    neu = score['neu']
    pos = score['pos']
    comp = score['compound']
    polarity += analysis.sentiment.polarity
    if neg > pos:
        negative_list.append(tweet)
        negative += 1
        df_lb.update({tweet:"negative"})
    elif pos > neg:
        positive_list.append(tweet)
        positive += 1
        df_lb.update({tweet:"positive"})

    elif pos == neg:
        neutral_list.append(tweet)
        neutral += 1
        df_lb.update({tweet:"neutral"})
```

Figure 3.2.1: Code for Text Blob Algorithm



```python
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()
noOfTweet=len(df_text)
positive = 0
negative = 0
neutral = 0
polarity = 0
tweet_list = []
neutral_list = []
negative_list = []
positive_list = []

for sentence in df_text['text']:
    polarity = sia.polarity_scores(sentence)
    pos = polarity["pos"]
    neu = polarity["neu"]
    neg = polarity["neg"]
    if neg > pos:
        negative_list.append(tweet)
        negative += 1
    elif pos > neg:
        positive_list.append(tweet)
        positive += 1

    elif pos == neg:
        neutral_list.append(tweet)
        neutral += 1


positive = percentage(positive, noOfTweet)
negative = percentage(negative, noOfTweet)
neutral = percentage(neutral, noOfTweet)
#polarity = percentage(polarity, noOfTweet)
positive = format(positive, '.1f')
negative = format(negative, '.1f')
neutral = format(neutral, '.1f')
```

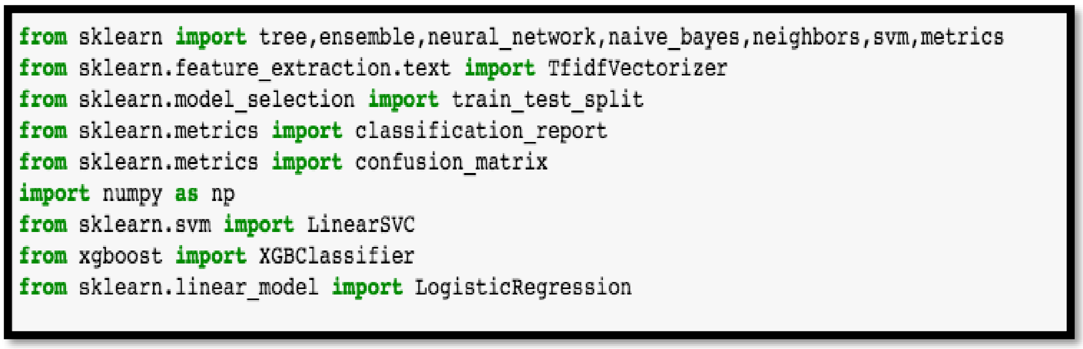
Figure 3.2.2: Code for Vader Algorithm

Figures 3.2.1 and 3.2.2 show a snapshot of the code where we first used the text blob library and the Vader library, respectively, after which we calculated the negative score and positive score of a tweet and further classified it as positive, negative, or neutral depending on whether the positive score outweighed the negative score or vice versa. The fraction of good, negative, and neutral tweets was then calculated and used to create the pie chart.

3.2.2 Features Extractions

The results that we have obtained by Vader and Text Blob can be further studied and more concretely confirmed by the use of machine learning classifiers. There are 6 algorithms we have taken into consideration (Logistic Regression, Decision Tree, Random Forest, Neural Network, Naive Bayes, K-Nearest Neighbours (KNN), Support Vector Machine (SVM), XG Boost).

As seen in figure 3.2.2.1, we first imported the essential Python libraries. Then, as indicated in figure 3.2.2.2, data had been divided into two groups: 80% for training and 20% for testing. The textual version of the training set is shown in Figure 3.2.2.3.

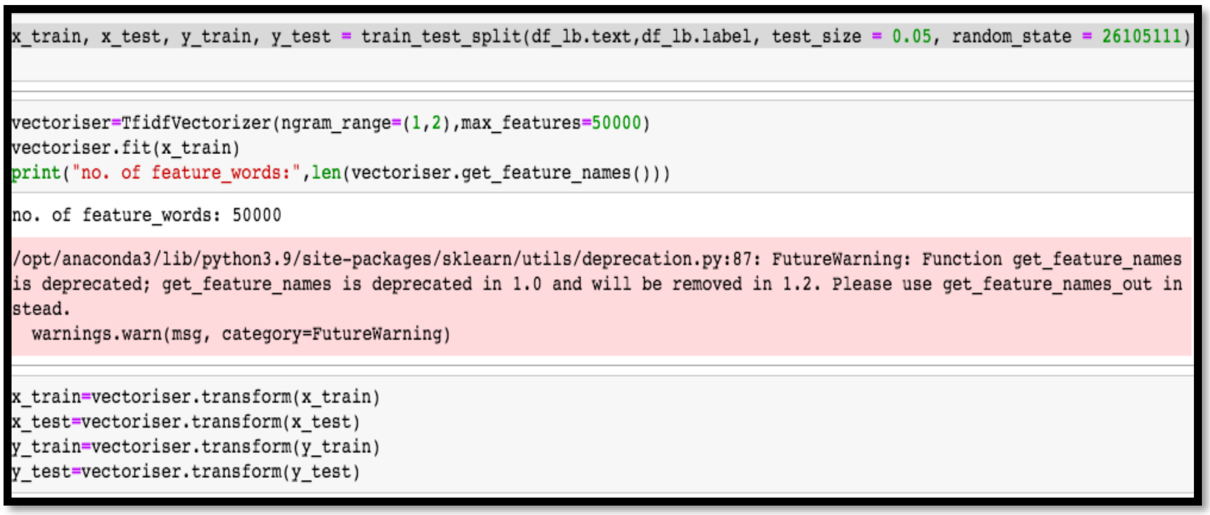


```
from sklearn import tree,ensemble,neural_network,naive_bayes,neighbors,svm,metrics
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import numpy as np
from sklearn.svm import LinearSVC
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression
```

Fig 3.2.2.1: Code for importing python libraries

For feature extractions, we used a combination of Term Frequency Inverse Document Frequency (TF-IDF) and N-gram techniques. Term frequency tells the frequency of the term in a document and Inverse document frequency tells about the relative rarity of a term in the collection of documents. To get the final TF-IDF value, multiply TF and IDF.

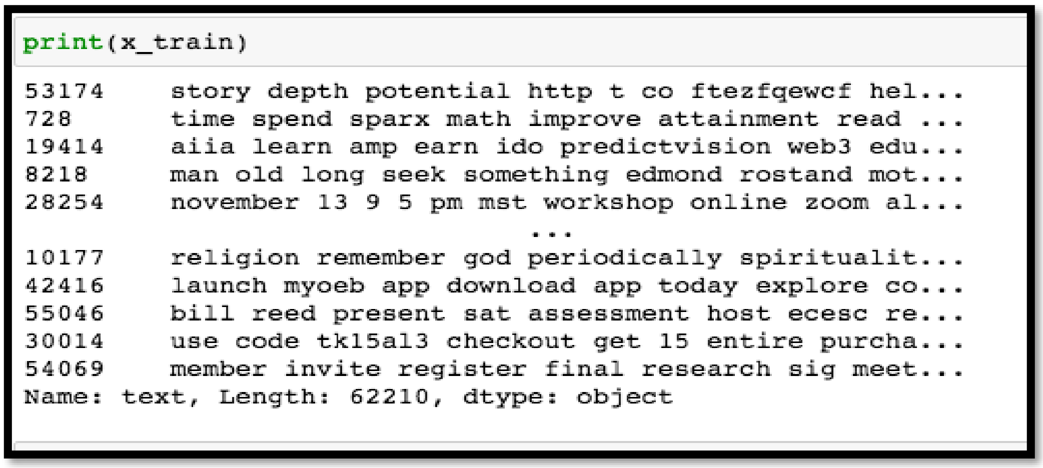
N-gram is a probabilistic model which counts how many times word sequences occur in corpus text and estimates the probability it is important in capturing the meaning of word sequences in text, where N represents the number of tokens. For example, “I” is a one-gram, “I am” is a two-gram, “I am a student” is a three-gram, and so on. Without N-grams “I”, “am”, and “student” would be considered separately, thus not giving the required output.



```
x_train, x_test, y_train, y_test = train_test_split(df_lb.text,df_lb.label, test_size = 0.05, random_state = 26105111)

vectoriser=TfidfVectorizer(ngram_range=(1,2),max_features=50000)
vectoriser.fit(x_train)
print("no. of feature_words:",len(vectoriser.get_feature_names()))

no. of feature_words: 50000

/opt/anaconda3/lib/python3.9/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names
is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out in
stead.
  warnings.warn(msg, category=FutureWarning)

x_train=vectoriser.transform(x_train)
x_test=vectoriser.transform(x_test)
y_train=vectoriser.transform(y_train)
y_test=vectoriser.transform(y_test)
```

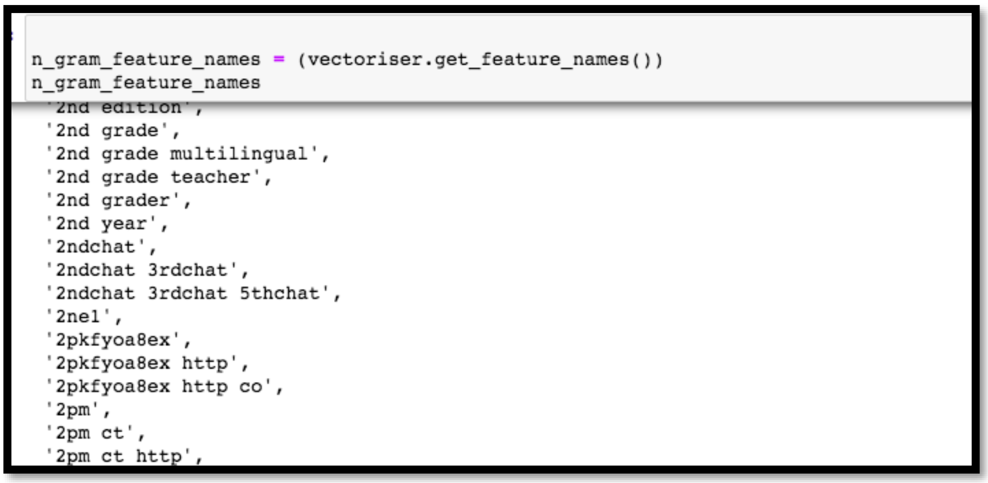
Figure 3.2.2.2: Code for splitting data set and converting data into numerical values

The data set in text form then we transformed it into a numerical representation with the help of feature vectors and we generated 50,000 feature words as shown in figure 3.2.2.4. In numerical dataset in output like "(0, 44667) 0.199853008 " , 0 represents 0 represents the index of the feature extracted and 44677 is a numerical notation of that particular feature and the last the TF-IDF value has defined.



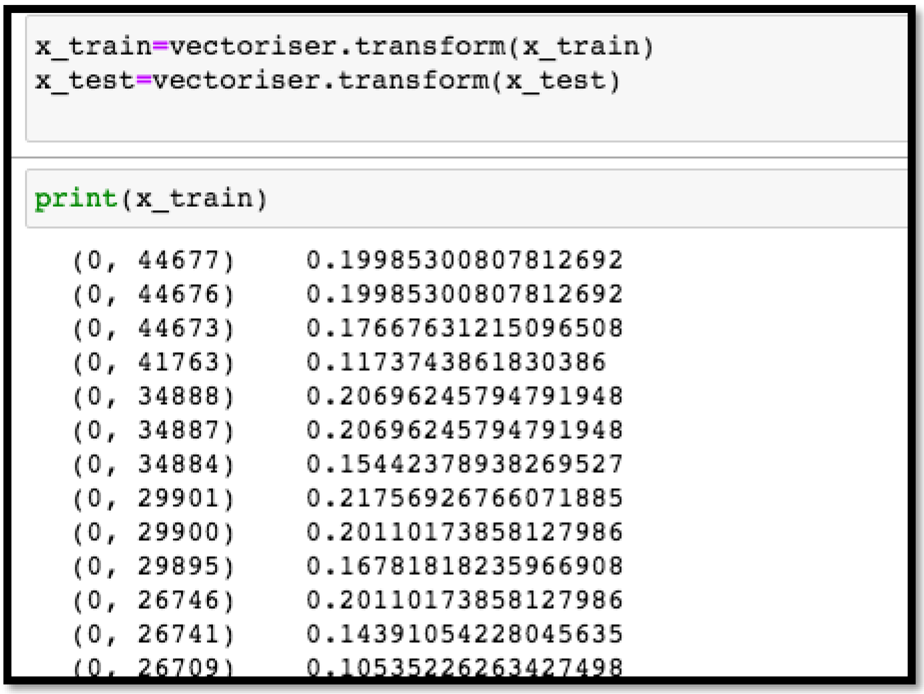
```
print(x_train)

53174    story depth potential http t co ftezfqewcf hel...
728      time spend sparx math improve attainment read ...
19414    aiia learn amp earn ido predictvision web3 edu...
8218     man old long seek something edmond rostand mot...
28254    november 13 9 5 pm mst workshop online zoom al...
                               ...
10177    religion remember god periodically spiritualit...
42416    launch myoeb app download app today explore co...
55046    bill reed present sat assessment host ecesc re...
30014    use code tk15al3 checkout get 15 entire purcha...
54069    member invite register final research sig meet...
Name: text, Length: 62210, dtype: object
```

Figure 3.2.2.3: Training set in Text form



```
n_gram_feature_names = (vectoriser.get_feature_names())
n_gram_feature_names
 '2nd edition',
 '2nd grade',
 '2nd grade multilingual',
 '2nd grade teacher',
 '2nd grader',
 '2nd year',
 '2ndchat',
 '2ndchat 3rdchat',
 '2ndchat 3rdchat 5thchat',
 '2nel',
 '2pkfyoa8ex',
 '2pkfyoa8ex http',
 '2pkfyoa8ex http co',
 '2pm',
 '2pm ct',
 '2pm ct http',
```

Figure 3.2.2.4: Feature words



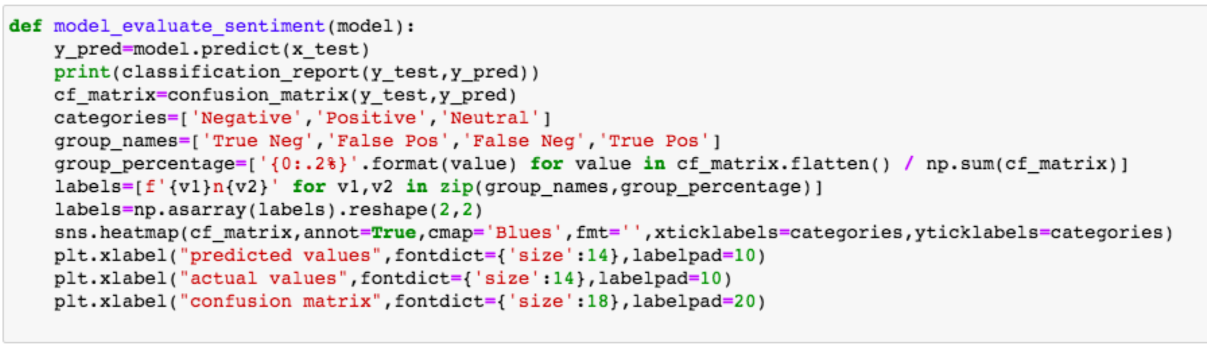
```
x_train=vectoriser.transform(x_train)
x_test=vectoriser.transform(x_test)
```

```
print(x_train)
```

```
  (0, 44677)	0.19985300807812692
  (0, 44676)	0.19985300807812692
  (0, 44673)	0.17667631215096508
  (0, 41763)	0.1173743861830386
  (0, 34888)	0.20696245794791948
  (0, 34887)	0.20696245794791948
  (0, 34884)	0.15442378938269527
  (0, 29901)	0.21756926766071885
  (0, 29900)	0.20110173858127986
  (0, 29895)	0.16781818235966908
  (0, 26746)	0.20110173858127986
  (0, 26741)	0.14391054228045635
  (0, 26709)	0.10535226263427498
```

Figure 3.2.2.5: Training set in numerical values

In figure 3.2.2.5 the entries in first column, say (0,44677),0 represents the index of the feature extracted and 44677 is numerical notation of that particular feature. And the last column is TF-IDF value. For calculating precision, recall, f1-score, and accuracy we made general function as shown in in Figure 3.2.3.1.



```python
def model_evaluate_sentiment(model):
    y_pred=model.predict(x_test)
    print(classification_report(y_test,y_pred))
    cf_matrix=confusion_matrix(y_test,y_pred)
    categories=['Negative','Positive','Neutral']
    group_names=['True Neg','False Pos','False Neg','True Pos']
    group_percentage=['{0:.2%}'.format(value) for value in cf_matrix.flatten() / np.sum(cf_matrix)]
    labels=[f'{v1}n{v2}' for v1,v2 in zip(group_names,group_percentage)]
    labels=np.asarray(labels).reshape(2,2)
    sns.heatmap(cf_matrix,annot=True,cmap='Blues',fmt='',xticklabels=categories,yticklabels=categories)
    plt.xlabel("predicted values",fontdict={'size':14},labelpad=10)
    plt.xlabel("actual values",fontdict={'size':14},labelpad=10)
    plt.xlabel("confusion matrix",fontdict={'size':18},labelpad=20)
```

Figure 3.2.2.6: Code for evaluating classification report

3.2.3 ML Classifiers

ML classifiers were used for validation. ML classifiers uses either supervised or unsupervised learning. Supervised learning trains the classifier on labeled input while unsupervised trains the classifier on unlabeled input, as we have labeled the data therefore, we used 8 supervised learning ML classifiers namely, Logistic regression, Decision tree, Random forest, Multilayer Perceptron(MLP), Naïve Bayes, Support vector machine(SVM), K-nearest neighbors(KNN), and XG-Boost. Decision Tree classify the data in tree structured form which contains decision node which have branches and make decision by using the features of the given data set and leaf node which have output. Random forest classifier combines the result of several decision tree. MLP classifier is based on neural network for classification. Naïve bayes classifier depends on Bayes theorem of probability. SVM classifier finds best decision line that divides the space into different classes. In Logistic regression we don't go for regression line but we use logistic function which converts the data into the range of 0 and 1. KNN classifier is a lazy classifer which learns from the training data only when some test data is given. XG Boost uses ensemble approach to classify the data which is based on Gradient boosting algorithm.

Then we trained the ML classifiers and calculated precision, recall, and f1-score . Precision is the metric which measures the number of positive predictions predicted by classifier were correct while Recall measures the number of positive class data in original dataset were correctly identified by the classifier. There is a trade-off between the precision and recall metrics. The F1-score is a metric which is the combination of two metrics that are precision and recall and f1 score use their harmonic mean to combine them. Thus, increasing the f1 score increases precision and recall simultaneously. Thus, scientists use F1 score with accuracy for evaluating the performances of their models. Accuracy is a metric which measures the number of correct classifications classified by the trained classifier.

Logistic Regression

Logistic regression is a ML classifier that uses a linear function (known as logit function) that is $g(y) = a_0 + a_1y_1 + \cdots + a_ry_r$. The coefficients a_0, a_1, …, a_r are the predicted weights and they are taken in such a way that accuracy of result is near to 1.

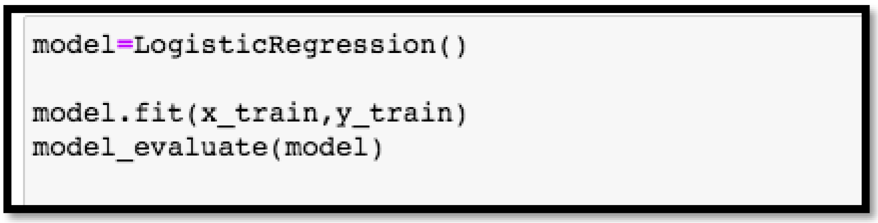
The sigmoid function of $f(y)$ that is $q(y) = 1 / (1 + \exp(-f(y))$ is a logistic regression function. The $q(y)$ is the probability that is predicted.

For example, if for a given value of y, the value of $q(y)$ that we get is 1. Then $1 - q(y)$ is the probability whose value is 0.

Calculated the probability of belonging to a particular class.

- 11111111111If p>50%-->1(positive)
- If p=50%-->0(neutral)
- If p<50%-->2(negative)

Figure 3.2.3.1 represents the code for Logistic regression used for training the same model and evaluating the classification report and confusion matrix.



```
model=LogisticRegression()

model.fit(x_train,y_train)
model_evaluate(model)
```

Figure 3.2.3.1: Code for Logistic Regression

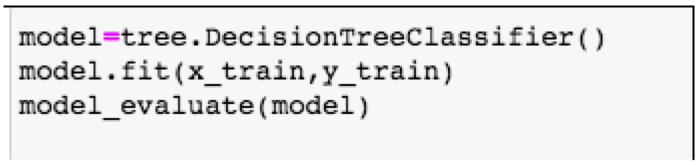
Decision Tree

Decision Tree is a classifier that classify the data with the tree like structure that contains branches and nodes. Branches are used for the decision making and nodes have features of dataset. When nodes contain the result of any other nodes that don't have any sub-branch that nodes are called leaf nodes and when nodes have sub-branches those nodes are called decision nodes.

The most widely used selection techniques are:

- Entropy
- Information Gain
- Gain Ratio
- Gini Index

Figure 3.2.3.2 represents the code for Decision tree used for the training the same model and evaluating the classification report and confusion matrix.

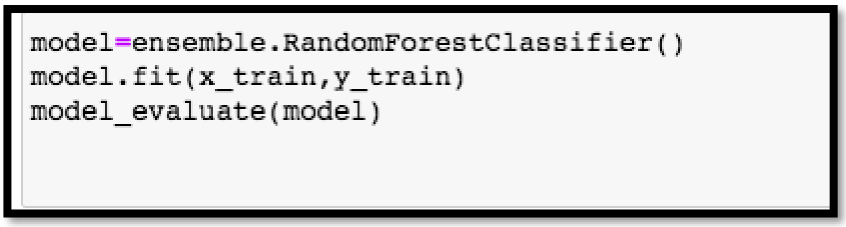


```
model=tree.DecisionTreeClassifier()
model.fit(x_train,y_train)
model_evaluate(model)
```

Figure 3.2.3.2: Code for Decision Tree

Random Forest
To increase the accuracy, one can use several decision tress and combine the output to produce best output that is what one of the classifier namely Random Forest do.

Figure 3.2.3.3 represents the code for Random forest used for the training the same model and evaluating the classification report and confusion matrix.



```
model=ensemble.RandomForestClassifier()
model.fit(x_train,y_train)
model_evaluate(model)
```

Figure 3.2.3.3: Code for Random Forest

Multi-layer Perceptron (Neural Network)
MLP which stands for Multi-layer Perceptron is a supervised machine learning technique as shown in figure 3.2.3.4 we have n input attributes that is x1,x2,…,xn, this collection of attributes is called neuron that made the input layer of MLP, after that by using weights the input values are biased and transformed to a hidden layer of MLP and last it has activation function basically non-linear in nature that produces the output.

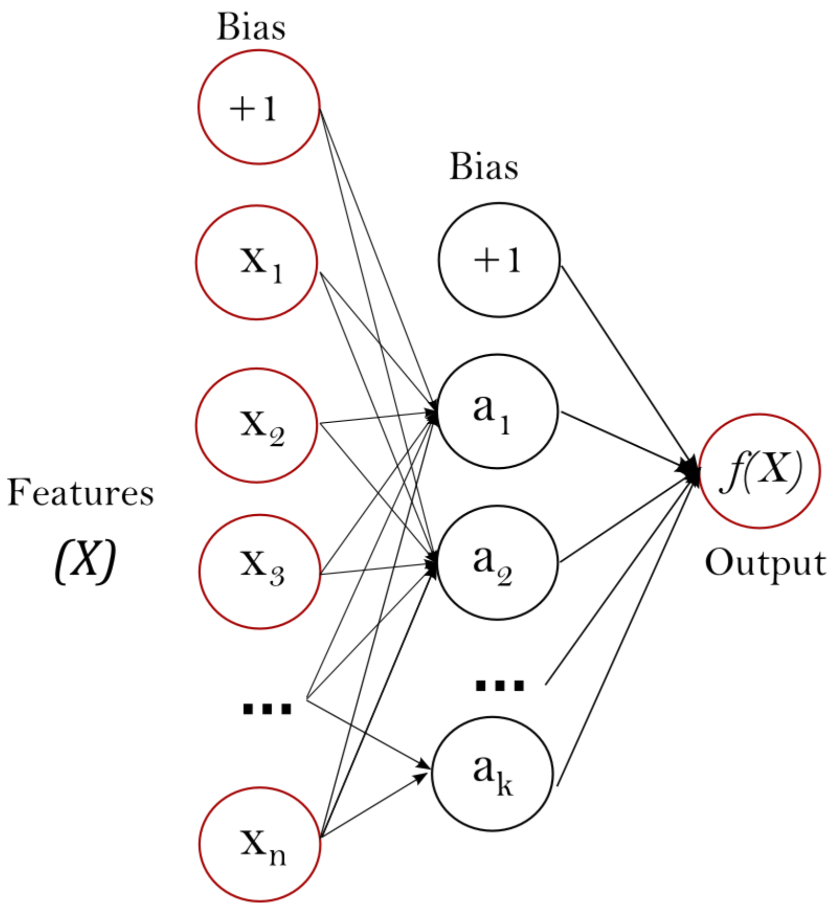
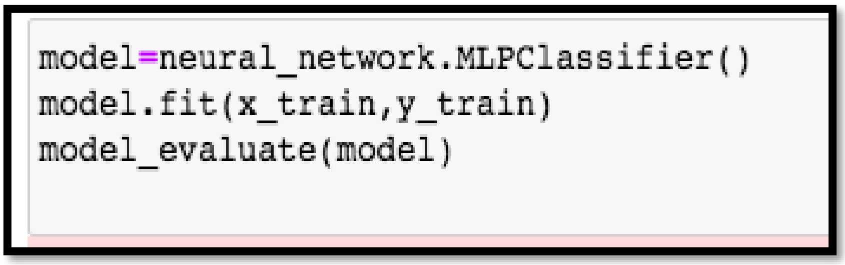


Figure 3.2.3.4: MLP layers

Figure 3.2.3.5 represents the code for MLP used for the training the same model and evaluating the classification report and confusion matrix.



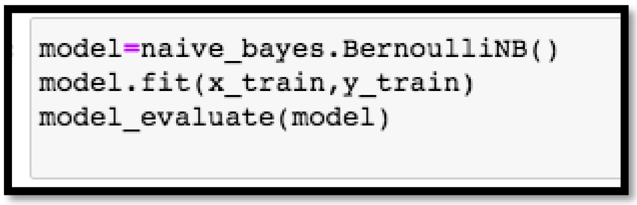
```
model=neural_network.MLPClassifier()
model.fit(x_train,y_train)
model_evaluate(model)
```

Figure 3.2.3.5: Code for MLP

Naive Bayes

This ML classifier is based on the famous probability theorem namely, Naïve Bayes. This classifier converts the input data set into frequency form and then calculate the corresponding probability by Bayes theorem and generates likelihood probability to finally classify the test data. It can’t be used to find the relation between different features and it is on of the main disadvantage.

Figure 3.2.3.6 represents the code for Naïve Bayes used for the training the same model and evaluating the classification report and confusion matrix.



```
model=naive_bayes.BernoulliNB()
model.fit(x_train,y_train)
model_evaluate(model)
```

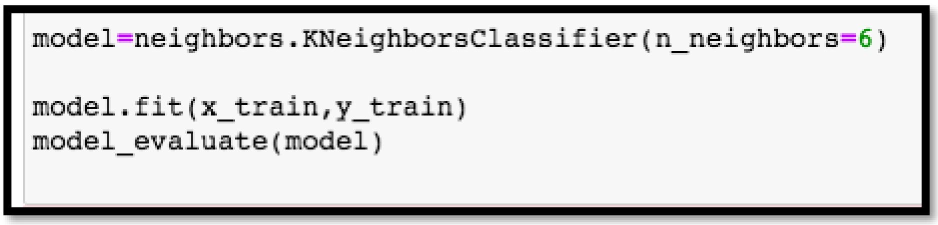
Figure 3.2.3.6: Code for Naïve Bayes

K-Nearest Neighbours (KNN)

KNN is a lazy learner which means that it don’t train the model when data set is given but it stores the given dataset and when some test data is given then it gets activated and do the classification. It basically finds the Euclidean distance as shown below of the test data and the input data. Now take K data point which is nearest to the input data. From these K data points the category which is occurring maximum times is assigned to the test data set.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Figure 3.2.3.7 represents the code for KNN used for the training the same model and evaluating the classification report and confusion matrix.



```
model=neighbors.KNeighborsClassifier(n_neighbors=6)

model.fit(x_train,y_train)
model_evaluate(model)
```

Figure 3.2.3.7: Code for KNN

Support Vector Machine (SVM)

SVM is a ML classifier that's main goal is to select support vectors which helps to create hyperplane or decision boundary. This decision boundary helps to classify the new test data as shown in Figure 3.2.3.8.

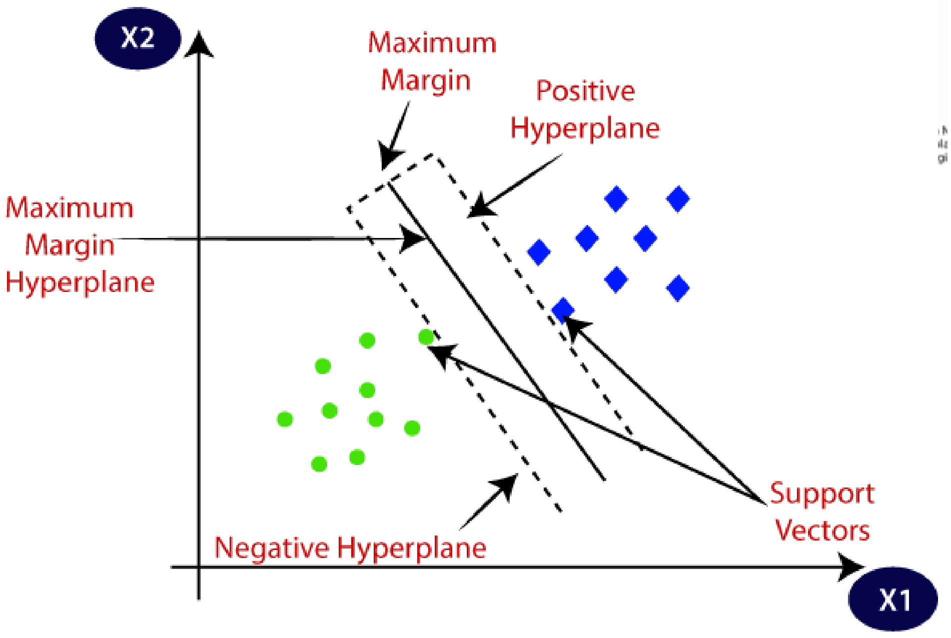
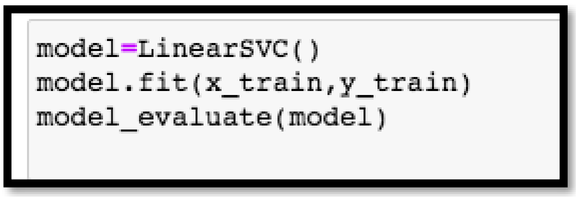


Figure 3.2.3.8: SVM method

Figure 3.2.3.9 represents the code for SVM used for the training the same model and evaluating the classification report and confusion matrix.



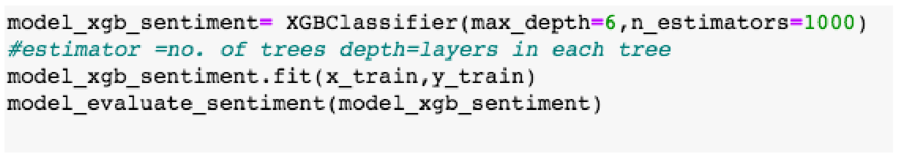
```
model=LinearSVC()
model.fit(x_train,y_train)
model_evaluate(model)
```

Figure 3.2.3.9: Code for SVM

XG Boost

Extreme Gradient Boosting increases the training accuracy of normal gradient boosting. It contains the various decision trees in a sequence and first, it uses one decision tree with some initial weights, after that the wrong prediction's weights are updated and sent to the second decision tree, and so on.

Figure 3.2.3.10 represents the code for XG Boost used for training the same model and evaluating the classification report and confusion matrix.



```
model_xgb_sentiment= XGBClassifier(max_depth=6,n_estimators=1000)
#estimator =no. of trees depth=layers in each tree
model_xgb_sentiment.fit(x_train,y_train)
model_evaluate_sentiment(model_xgb_sentiment)
```

Figure 3.2.3.10: Code for XG Boost

3.3 EMOTION DETECTION

3.3.1 Labeling of data

Emotion detection techniques can identify students' emotional states, such as stress, boredom, or engagement, during online learning activities. Educators can receive real-time notifications about students experiencing negative emotions, enabling them to intervene and provide timely support. Online education can be isolating for some students, leading to feelings of loneliness or stress. Emotion detection can identify students' emotional states related to mental health, such as signs of anxiety or distress. Educators can use this information to reach out, offer support resources, or connect students with mental health services, promoting their well-being and academic success.

In emotion detection process, we considered the previously used pre-processed data and performed the labeling with the help of text2emotion python library into 5 emotions namely 'happy', 'surprise', 'sad', 'fear', and 'angry'. Text2emotion is the package in python for extracting the emotions from any textual message and it gives emotion scores ranging from 0 to 1 for five emotions namely happy, surprise, sad, angry, and fear. Text2emotion gives the output in the form of a dictionary. Let us understand how this algorithm work, consider if you are a teacher and your students are doing well in studies then you will say 'Very good my child! Keep it up.' , such type of comment show that you are happy with student performance. Text2emotion also extracts the emotion in similar manner.

3.3.2 Features Extractions
The fundamental step in the emotion detection process is feature extraction. In emotion detection as well we used combination of TF-IDF and n gram technique for extraction of features. We used 1 to 3 n gram range and give it as an input to TF-IDF vectorizer to get feature vectors. TF-IDF vectorizer will kept only one to three words sentences as we used n gram range from 1 to 3. Therefore, feature vectors will be of length of 1 to 3 words only. For emotion detection purposes we generated 42,000 feature vectors. Then we converted the training data set into numerical form.

3.3.3 ML Classifiers
For training the ML classifiers namely, Logistic regression, Decision tree, Random forest, Multilayer Perceptron (MLP), Naïve Bayes, Support vector machine(SVM), K-nearest neighbors(KNN), and XG-Boost we used 80% numerical training data set and used 20% data set to evaluate the f1 score, recall, precision, and accuracy of eight different ML classifiers and the accuracy of the classification depends on these features trained ML classifiers. We used classification report and confusion matrix to evaluates the performance of classifiers in f1 score, precision, recall, and accuracy form and in matrix form respectively.

CHAPTER 4

RESULT AND DISCUSSION

4.1 SENTIMENT ANALYSIS RESULTS

Sentiment analysis and emotion detection techniques offer valuable insights and opportunities for personalized interventions in online education. They can help improve student satisfaction, engagement, collaboration, and emotional well-being. By harnessing the power of these techniques, online education can become more effective, inclusive, and tailored to meet the diverse needs of students in the digital learning landscape.

In this paper, we evaluated the accuracy of eight machine learning algorithms for both sentiment analysis and emotion detection of tweets related to online education. Figure 4.1.1 and figure 4.1.2 represent the pie chat for text blob and Vader result respectively. Table 4.1.1 shows Text Blob results and Vader algorithm results. According to Text Blob result, data set consisting of 59.8%, 13.7%, and 26.5% of positive, negative, and neutral tweets respectively and according to Vader result, data set consisting of 60.1%, 13.6%, and 26.3% of positive, negative, and neutral tweets respectively.

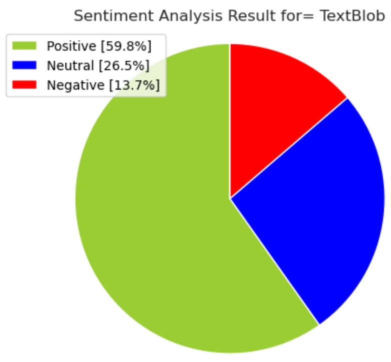


Figure 4.1.1. Pie chart for Text blob result.

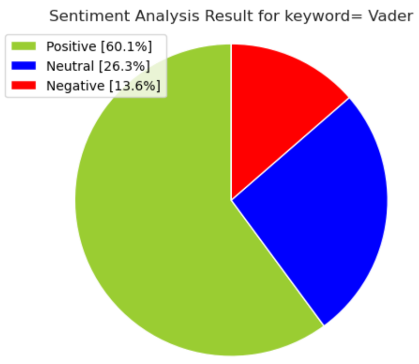


Figure 4.1.2. Pie chart for Vader result.

Table 4.1.1. Result of Text Blob and Vader algorithm.

No. Of Tweets	Text Blob	Vader
Positive	59.8%	60.1%
Negative	13.7%	13.6%
Neutral	26.5%	26.3%

Figure 4.1.3 and figure 4.1.4 represent the frequency of positive, negative, and neutral tweets for Text Blob and Vader results, respectively. Figure 4.1.5 and figure 4.1.6 represent the word cloud for Text Blob and Vader result respectively which displays the most prevalent words that are evident in the tweets.

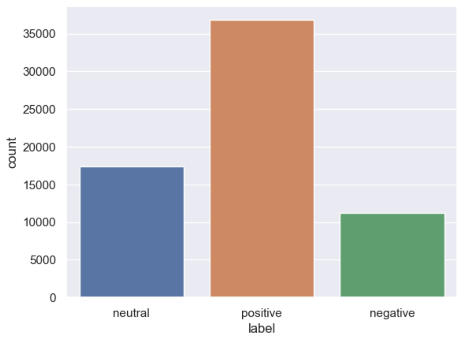


Figure 1.1.3: Frequency plot for Text Blob result.

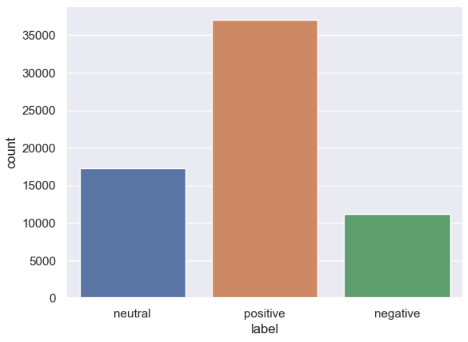


Figure 4.1.4:. Frequency plot for Vader result.

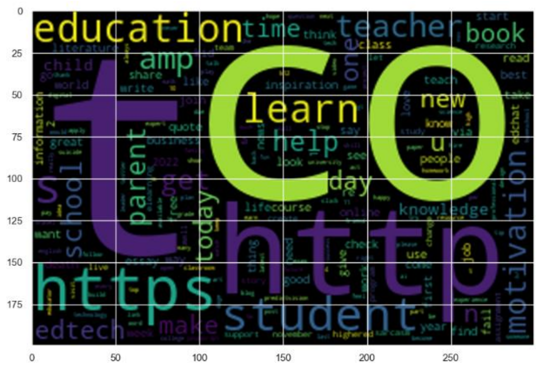


Figure 4.1.5: Word Chloud for Text Blob result.

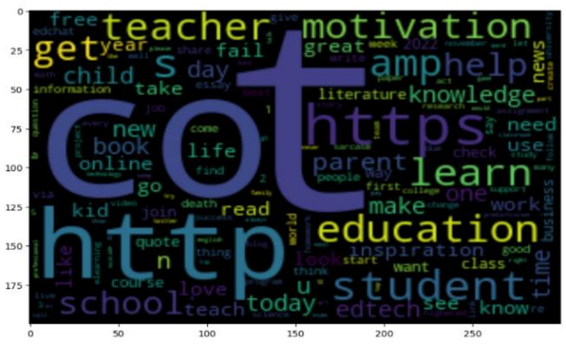


Figure 4.1.6: Word Cloud for Vader result.

Table 4.1.2 shows the precision, recall, f1-score, and accuracy of 8 different ML classifiers for the sentiment. Figure 4.1.7 represents the result of Logistic Regression which achieved 92% accuracy and the confusion matrix tells that 1,763, 826, and 423 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified. Figure 4.1.8 shows the result of the Decision Tree which achieved 91% accuracy and the confusion matrix tells that 1,740, 830, and 410 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified. Figure 4.1.9 shows the result of Random Forest which achieved 89% accuracy and the confusion matrix tells that 1,701, 842, and 383 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified. Figure 4.1.10 shows the result of the MLP classifier which achieved 92% accuracy and the confusion matrix tells that 1,778, 811, and 440 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified. Figure 4.1.11 shows the result of Naive Bayes which achieved 81% accuracy and the confusion matrix tells that 1,665, 565, and 421 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified. Figure 4.1.12 shows the result of KNN which achieved 70% accuracy and the confusion matrix tells that 1,548, 497, and 262 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified. Figure 4.1.13 shows the result of SVM which achieved 94% accuracy and the confusion matrix tells that 1,784, 843, and 449 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified. Figure 4.1.14 shows the result of XG boost which achieved 94% accuracy and the confusion matrix tells that 1,786, 864, and 443 from 1,866 negative tweets, 872 neutral tweets, and 537 positive tweets respectively were correctly classified.

Table 4.1.2: Result of ML Classifiers for Sentiment Analysis.

Classifier	Precision	Recall	F1-score	Accuracy
Logistic regression	0.95	0.94	0.95	92%
DT	0.94	0.93	0.94	91%
RF	0.95	0.91	0.93	89%
MLP	0.95	0.95	0.95	92%
NB	0.83	0.89	0.86	81%
K-NN	0.75	0.83	0.79	70%
SVM	0.97	0.96	0.96	94%
XG Boost	0.97	0.96	0.96	94%

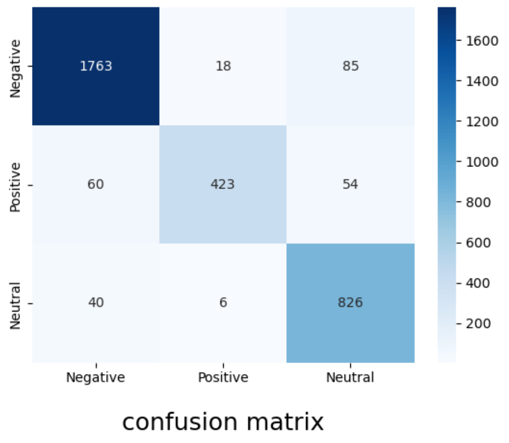


Figure 4.1.7: Sentiment analysis result for Logistic Regression.

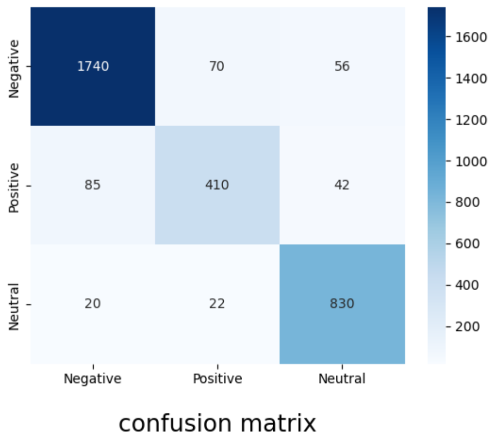


Figure 4.1.8: Sentiment analysis result for Decision Tree.

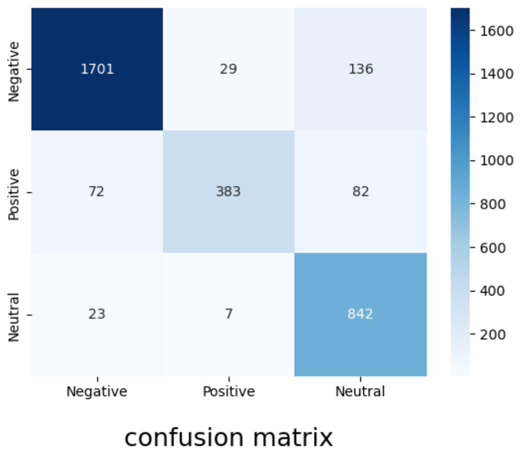


Figure 4.1.9: Sentiment analysis result for Random Forest

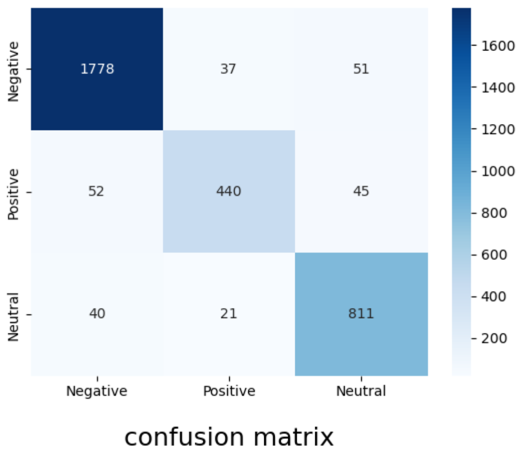


Figure 4.1.10:. Sentiment analysis result for MLP

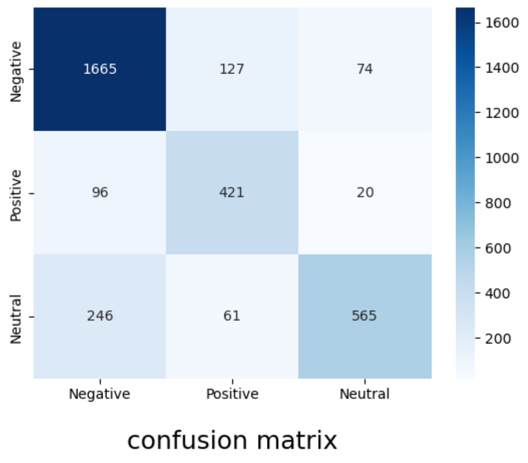


Figure 4.1.11:. Sentiment analysis result for Naïve Bayes

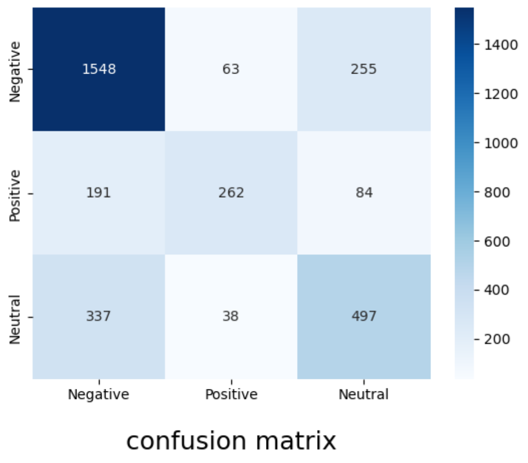


Figure 4.1.12:. Sentiment analysis result for KNN

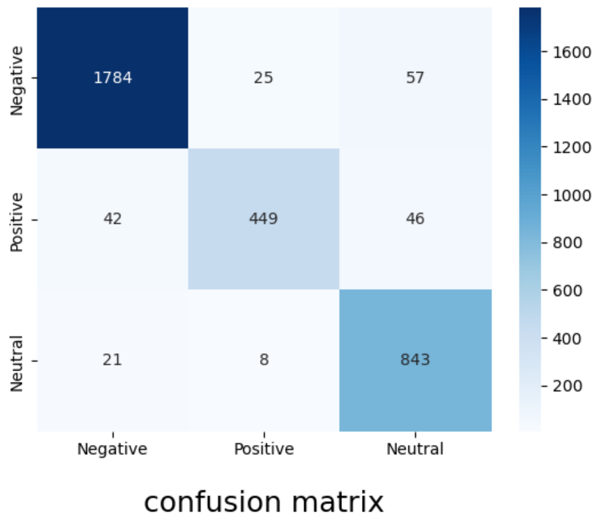


Figure 4.1.13:. Sentiment Analysis Results for SVM.

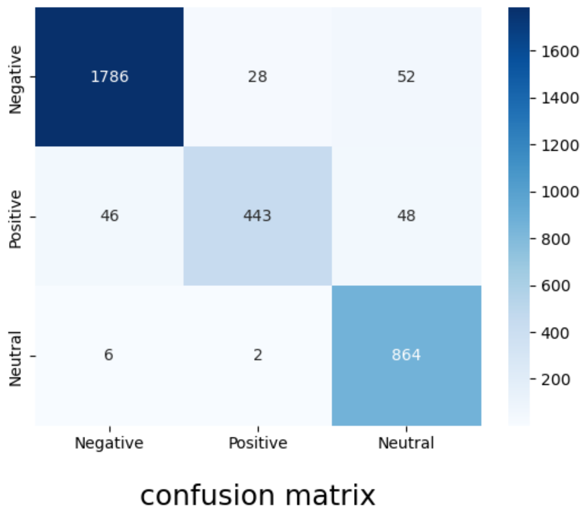


Figure 4.1.14:. Sentiment Analysis Results for XG Boost.

4.2 EMOTION DETECTION RESULTS

Figure 4.2.1 represents a pie chart that tells that 15.6% of people were happy, 35.8% of people were surprised, 18.6% of people were sad, 28% of people were fearful and 2% of people were angry with online education. Figure 4.2.2 represents a bar graph where we plotted the frequency count of five different emotions.

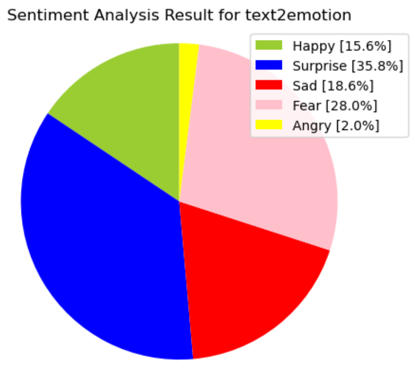


Figure 4.2.1: Pie Chart for 5 different emotions

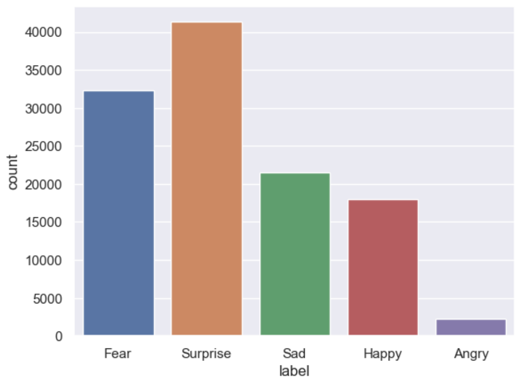


Figure 4.2.2. Frequency graph for 5 different emotions.

Table 4.2.1 shows the result of text2emotion library and according to it data set consisting of 15.6% of happy emotion tweets, 35.8% of surprised emotion tweets, 18.6% of sad emotion tweets, 28% of fearful emotion tweets, and 2% of angry emotion tweets. Therefore, from these results we conclude that most of the people reacted with surprised emotion toward online education during COVID-19.

Table 4.2.1: Result of Text2emotion library

Emotions	Number of tweets in percentage
Happy	15.6%
Surprise	35.8%
Sad	18.6%
Fear	28.0%
Angry	2.0%

Table 4.2.2 shows the precision, recall, f1-score, and accuracy of 8 different ML classifiers for the emotion detection of tweets. The highest accuracy is achieved by MLP, SVM, and XG-Boost and the lowest accuracy is achieved by KNN that is 71% and the accuracy of Logistic regression, Decision Tree, Random forest, Naïve Bayes are 93%, 92%, 93%, and 79% respectively. The f1 score of MLP, and SVM are 0.97 which is highest among all other classifiers and f1 score of Logistic regression, Decision Tree, Random Forest, Naïve Bayes, KNN, and XG Boost are 0.93, 0.91, 0.93, 0.80, 0.66, and 0.96 respectively. The precision score of MLP is highest that is of 0.98 and of KNN is lowest that is of 0.66 among all classifiers. The precision score of other classifiers that are Logistic regression, Decision tree, Random forest,

Naïve bayes, SVM, and XG Boost are 0.95, 0.91, 0.94, 0.82, 0.97, and 0.96 respectively. The recall score of SVM and XG Boost are highest that is of 0.96 and of KNN is lowest that is of 0.67 among all classifiers. The recall score of other classifiers that are Logistic regression, Decision tree, Random forest, Naïve bayes, and MLP are 0.92, 0.92, 0.93, 0.77, and 0.95 respectively. Therefore, we conclude that the best model is XG boost, SVM and MLP.

Table 4.2.2: Result of ML Classifiers for Emotion Detection.

Classifier	Precision	Recall	F1-score	Accuracy
Logistic regression	0.95	0.92	0.93	93%
DT	0.91	0.92	0.91	92%
RF	0.94	0.93	0.93	93%
MLP	0.98	0.95	0.97	96%
NB	0.82	0.77	0.80	79%
K-NN	0.66	0.67	0.66	71%
SVM	0.97	0.96	0.97	96%
XG Boost	0.96	0.96	0.96	96%

Figure 4.2.3 shows the result of Logistic Regression which achieved 93% accuracy and the corresponding confusion matrix tells that 871 sad emotion tweets, 943 happy emotion tweets, 1,947 surprise emotion tweets, 1,532 angry emotion tweets, and 68 fear emotion tweets from 951 sad emotion tweets, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were correctly classified. Figure 4.2.4 shows the result of the Decision Tree which achieved 92% accuracy and corresponding the confusion matrix tells that 872 sad emotion tweets, 953 happy emotion tweets, 1,863 surprise emotion tweets, 1,513 angry emotion tweets and 89 fear emotion tweets from 951 sad emotion tweets, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were correctly classified. Figure 4.2.5 shows the result of Random Forest which achieved 93% accuracy and the corresponding confusion matrix tells that 886 sad emotion tweets, 949 happy emotion tweets, 1,914 surprise emotion tweets, 1,539 angry emotion tweets and 85 fear emotion tweets from 951 sad, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were correctly classified. Figure 4.2.6 shows the result of the MLP classifier which achieved 96% accuracy and corresponding the confusion matrix tells that 907 sad emotion tweets, 1,011 happy emotion tweets, 1,974 surprise emotion tweets, 1,588 angry emotion tweets and 87 fear emotion tweets from 951 sad emotion tweets, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were correctly classified.

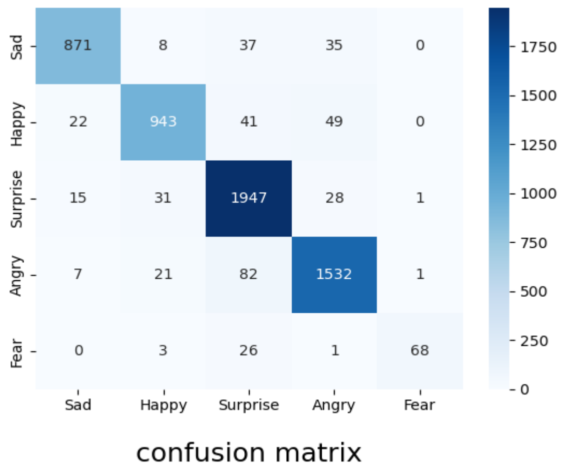


Figure 4.2.3: Emotion detection results for Logistic Regression

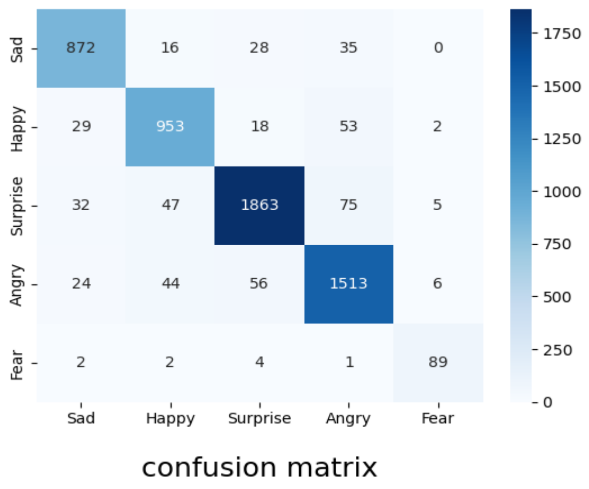


Figure 4.2.4: Emotion detection results for Decision tree

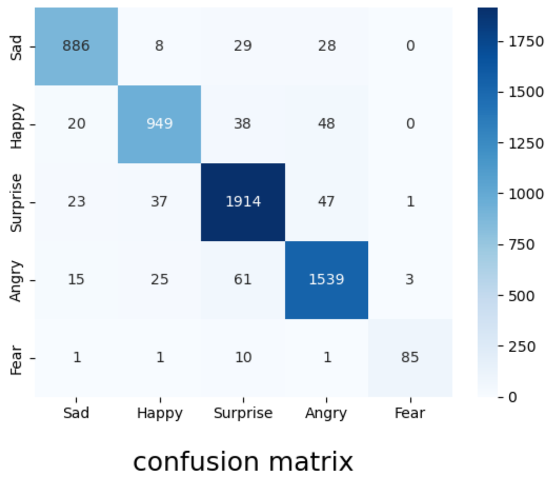


Figure 4.2.5: Emotion detection results for Random forest

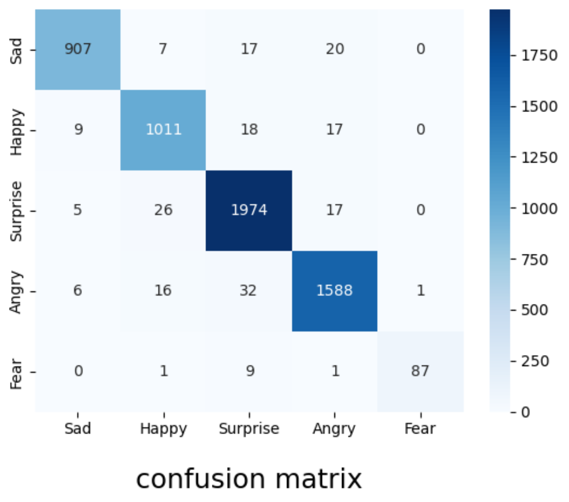


Figure 4.2.6: Emotion detection results for MLP.

Figure 4.2.7 shows the result of Naive Bayes which achieved 79% accuracy and the confusion matrix tells that 736, 822, 1,551, 1,407 and 38 from 951 sad, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were correctly classified. Figure 4.2.8 shows the result of KNN which achieved 71% accuracy and the confusion matrix tells that 635, 675, 1,627, 1,095 and 56 from 951 sad emotion tweets, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were .correctly classified.

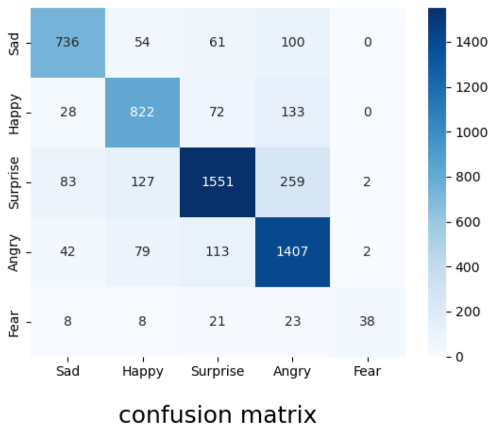


Figure 4.2.7: Emotion detection results for Naive Bayes.

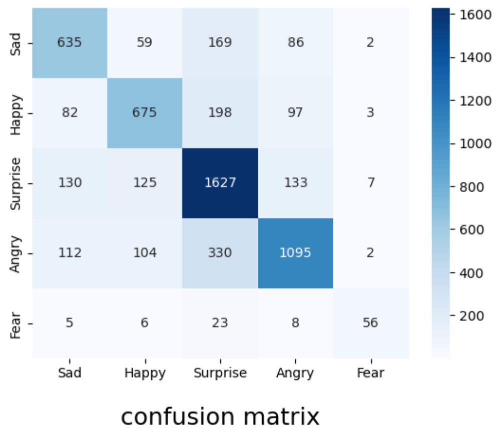


Figure 4.2.8:. Emotion detection results for KNN.

Figure 4.2.9 shows the result of SVM which achieved 96% accuracy and the confusion matrix tells that 917, 1008, 1,970, 1,586 and 85 from 951 sad emotion tweets, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were correctly classified. Figure 4.2.10 shows the result of XG boost which achieved 96% accuracy and the confusion matrix tells that 916, 1008, 1,977, 1,556 and 88 from 951 sad emotion tweets, 1,055 happy emotion tweets, 2,022 surprise emotion tweets, 1,643 angry emotion tweets, and 98 fear emotion tweets respectively were correctly classified.

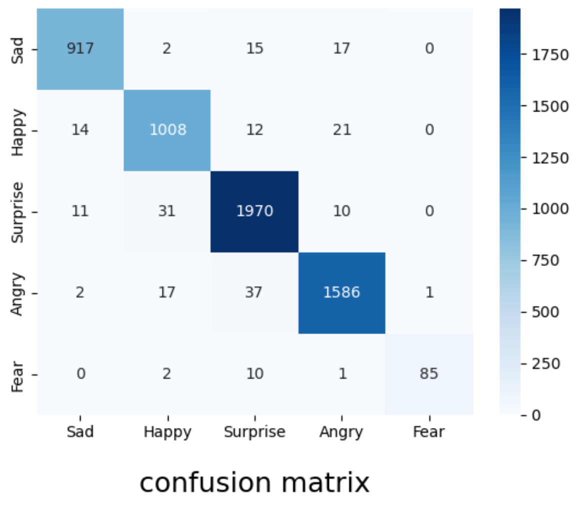


Figure 4.2.9: Emotion detection results for SVM.

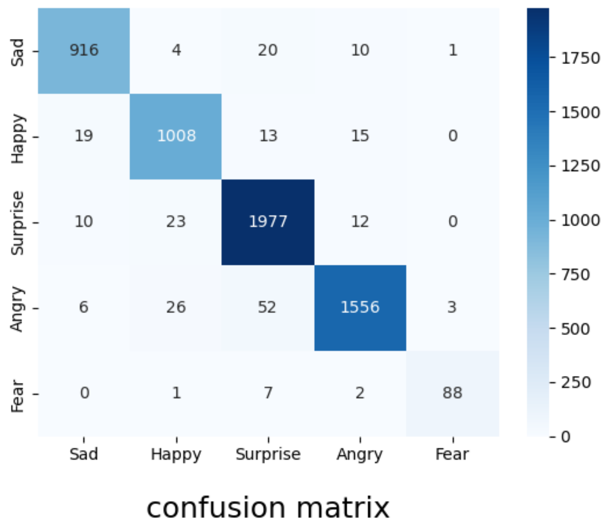


Figure 4.2.10: Emotion detection results for XG Boost.

CHAPTER 5

CONCLUSION

The significance of sentiment analysis and emotion detection in online education lies in their ability to enhance the learning experience, support student well-being, and improve educational outcomes. In this work, we analyzed the sentiments and emotions of tweets about online education during COVID-19. For this purpose, twitter dataset of 100K tweets was collected through the twitter API and after preprocessing, labeling of data by Text Blob, Vader, and text2emotion was done. For validation, we used eight Machine Learning techniques namely, Logistic regression, Decision tree, Random forest, Multilayer Perceptron(MLP), Naïve Bayes, Support vector machine(SVM), K-nearest neighbors(KNN), and XG-Boost. Combination of TF-IDF and n-gram technique was used for feature extraction. After performing the experiment it was observed that the XG boost ML classifier is more accurate in classifying tweets into positive, negative, or neutral with 94% accuracy where as it can classify the same tweet into happy, surprised, sad, fearful, or angry with 96% accuracy. SVM classifier also has the same accuracy as XG boost i.e., 94% in sentiment analysis and 96% in emotion detection. On the basis of this we observed that most people reacted positively with surprised emotion towards online education.

In the Future, we plan to further classify the positive and negative sentiments into five emotions namely happiness, surprise, sadness, fear, and anger by utilizing the output of sentiment classifier to the input of emotion classifiers.

REFERENCES

1. Baboo, S. S., & Amirthapriya, M.,Sentiment Analysis And Automatic Emotion Detection Analysis Of Twitter Using Machine Learning Classifiers. International Journal of Mechanical Engineering, Vol. 7, No. 2 (2022,February).
2. Jalil, Z., Abbasi, A., Javed, A. R., Badruddin Khan, M., Abul Hasanat, M. H., Malik, K. M., & Saudagar, A. K. J., Covid-19 related sentiment analysis using state-of-the-art machine learning and deep learning techniques. Frontiers in Public Health, 9, 2276 (2022).
3. Qorich, M., & El Ouazzani, R., Text sentiment classification of Amazon reviews using word embeddings and convolutional neural networks. The Journal of Supercomputing, 1-26 (2023).
4. Raheja, S., & Asthana, A., Sentiment Analysis of Tweets During the COVID-19 Pandemic Using Multinomial Logistic Regression. International Journal of Software Innovation (IJSI), 11(1), 1-16 (2023).
5. Krommyda, M., Rigos, A., Bouklas, K., & Amditis, A., An experimental analysis of data annotation methodologies for emotion detection in short text posted on social media. In: Informatics, Vol. 8, No. 1, p. 19, MDPI (2021, March).
6. Lim, W. L., Ho, C. C., & Ting, C. Y., Sentiment analysis by fusing text and location features of geo-tagged tweets. IEEE Access, 8, 181014-181027 (2020).
7. Chowanda, A., Sutoyo, R., & Tanachutiwat, S., Exploring text-based emotions recognition machine learning techniques on social media conversation. Procedia Computer Science, 179, 821-828 (2021).
8. Park, S. H., Bae, B. C., & Cheong, Y. G., Emotion recognition from text stories using an emotion embedding model. In: 2020 IEEE international conference on big data and smart computing (BigComp) (pp. 579-583). IEEE (2021, February).
9. Imran, A. S., Daudpota, S. M., Kastrati, Z., & Batra, R., Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets. Ieee Access, 8, 181074-181090 (2020).
10. Balabantaray, R. C., Mohammad, M., & Sharma, N., Multi-class twitter emotion classification: A new approach. International Journal of Applied Information Systems, 4(1), 48-53 (2012).
11. Bollen, J., Mao, H., & Pepe, A.: Modeling public mood and emotion, Twitter sentiment and socio-economic phenomena. In: Proceedings of the international AAAI conference on web and social media, Vol. 5, No. 1, pp. 450-453 (2011).
12. Larsen, M. E., Boonstra, T. W., Batterham, P. J., O'Dea, B., Paris, C., & Christensen, H., We feel: mapping emotion on Twitter. IEEE journal of biomedical and health informatics, 19(4), 1246-1252 (2015).
13. Alm, C. O., Roth, D., & Sproat, R., Emotions from text: machine learning for text-based emotion prediction. In Proceedings of human language technology conference and conference on empirical methods in natural language processing, pp. 579-586 (2005, October).
14. Roseman, I. J.: Cognitive determinants of emotion, A structural theory. Review of personality & social psychology (1984).
15. Darwin, C., & Prodger, P., The expression of the emotions in man and animals. Oxford University Press, USA (1998).