# Table of Contents

***[Abstract 3](#_gjdgxs)***

1. ***[Introduction 3](#_30j0zll)***
2. ***[Methods 4](#_1fob9te)***
   1. **[Study Background and Framework 4](#_3znysh7)**
   2. **[Specify the Research Design 4](#_2et92p0)**
   3. **The version [of the Libraries 4](#_tyjcwt)**
   4. **[Dataset: 5](#_3dy6vkm)**
   5. **[Data Wrangling: 5](#_1t3h5sf)**
      1. [Handling Missing Value 5](#_4d34og8)
   6. **[Text pre-processing: 6](#_2s8eyo1)**
      1. [Stop Words Removal](#_17dp8vu) [6](#_3rdcrjn)
   7. **[Implementing Spacy 6](#_26in1rg)**
   8. **[Vectorization 10](#_35nkun2)**
      1. [Bag of words 10](#_1ksv4uv)
      2. [TF-IDF 10](#_44sinio)
3. ***[Modelling 10](#_2jxsxqh)***
   1. **[Model 1: Classification Model using the Random Forest 10](#_z337ya)**
   2. **[Model 2: Classification Model using the Naive Bayes 10](#_3j2qqm3)**
   3. **Modeling [Summary](#_1y810tw)**
   4. **[11](#_1y810tw)**
4. ***[Results 11](#_4i7ojhp)***
5. ***[Metrics of the Models 14](#_2xcytpi)***
6. ***[Discussion 18](#_1ci93xb)***
7. ***[Conclusions and Future Work 19](#_3whwml4)***
8. ***[Reference 20](#_2bn6wsx)***
9. ***[Acknowledgment 20](#_qsh70q)***

**SENTIMENT ANALYSIS**

# Abstract

This Sentiment Analysis project is based on one of the most popular social media platforms, X, which was called Twitter before. This project aims to develop a sentiment analysis chatbot using machine learning techniques using the "Sentiment140" dataset, which contains 1599999 observations and six columns. We will preprocess the data, conduct exploratory data analysis (EDA), extract features, train machine learning models, and deploy the chatbot. The project will enhance understanding sentiment analysis, natural language processing (NLP), and chatbot development. We compared our model accuracy using different Machine Learning algorithms, where Logistic Regression stood as the best with an accuracy of 78.69%.

# Introduction

Nowadays, social media platforms have large amounts of data posted about their views and opinions in their applications. X, formerly known as Twitter, is one of the world's most-used social media platforms. The X users can share their short messages, images, and videos, called tweets. These tweets comprise different types of information, which will finally imply positive, negative, and neutral. Users even share their confidential information in their tweets; therefore, analyzing these tweets may help determine whether they are positive or negative. In the project, sentiment analysis is conducted on users' tweets, which will help find the user's situation and feelings.

Sentiment Analysis is a crucial application of NLP, enabling automated classification of texts based on the sentiments. It is also called opinion mining, which involves analyzing the virtual text to predict whether its tone is positive or negative. We used word cloud to understand the common words and to learn the common sentiments behind each word. These words are subject to different machine learning algorithms to train the model. The whole data set is split into 80-20 proportions. The project included classification algorithms Naive Bayes, Random Forest, Logistic Regression, and Gradient Boosting, in which Logistic Regressed noted high accuracy among the others.

# Methods

## Study Background and Framework:

The fundamental goal of this report is to develop a machine-learning model that analyses our sentiment analysis dataset. It mainly focuses on identifying whether the text in the tweet is good(positive) or bad (negative). The data set also involves unigram and bigram analysis using a Count Vectorizer. The main objective is to apply different machine-learning classification techniques and find the most accurate model for our project.

The project follows a standard machine-learning workflow:

**Data Collection**: A pre-existing dataset from kaggle of 1.6 million tweets.

**Data Preprocessing**: Cleaning and preparing the text data for analysis.

**Feature Extraction**: Transforming text data into numerical data.

**Modeling**: Training various machine learning models.

**Evaluation**: Assessing model performance using accuracy and other metrics.

## Specify the Research Design:

This study uses an experimental and comparative research approach involving pre-processing text, exploratory data analysis, finding a correlation, preparation of data, feature extraction, model training, and finally, model evaluation. The evaluated results of these models are further compared based on their precision, recall, and f1 score, therefore identifying the advantages and disadvantages of the respective model.

## The Libraries We Used

We have used the following mentioned libraries to build our project:

The Pandas

The Scikit-learn

The Numpy

The NLTK

Matplotlib

WordCloud

## Dataset:

The dataset consists of 1.6 million tweets with six features. The features in the dataset are as follows:

sentiment: The sentiment label of the tweet (0 for negative, 4 for positive).

id: A unique identifier for each tweet.

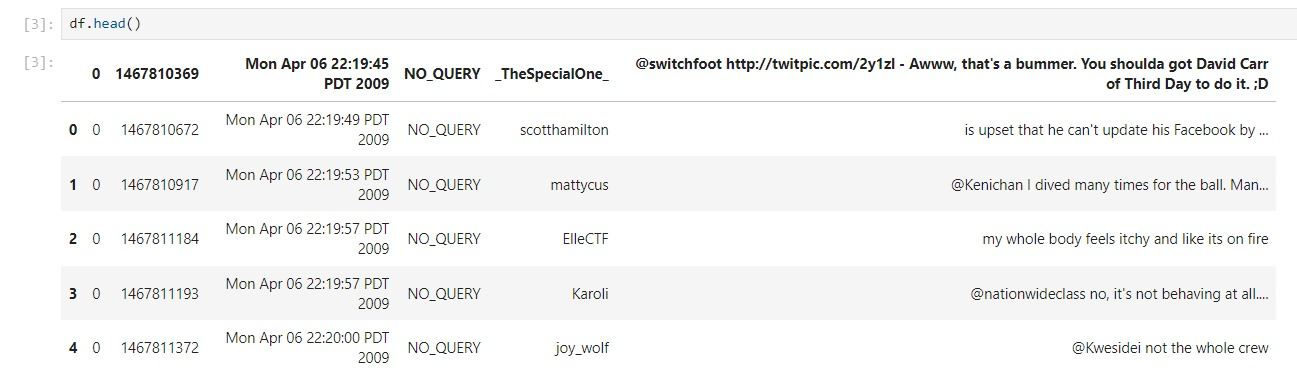
date: The date and time the tweet was posted.

query: The query used to retrieve the tweet (always "NO\_QUERY" in this dataset).

username: The username of the person who posted the tweet.

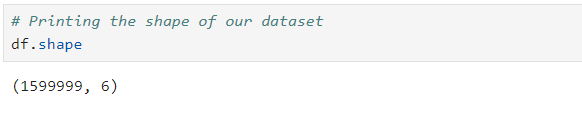
text: The actual content of the tweet.

Displaying the first five observations from the dataset:

****

**Shape of the Dataset:**

* Observations: 1,599,999
* Features: 6

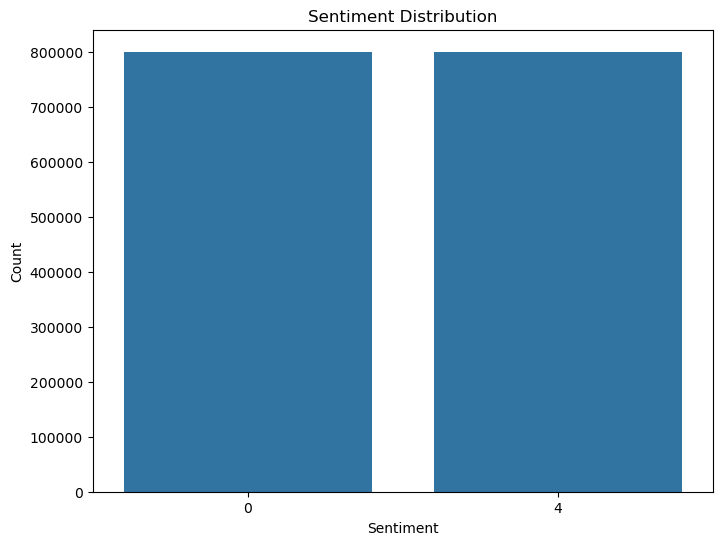


**Correlation Heatmap**

## 

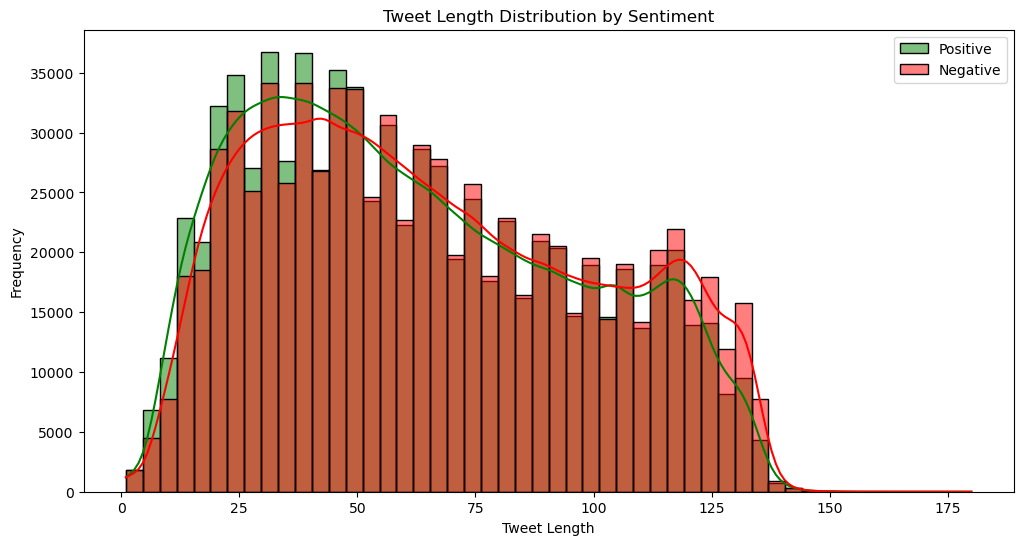
## 

**Bar Graph for evenly Distributed tweets**



*Figure 1 Sentiment Distribution*

Tweets are evenly distributed towards postive sentiment(4) and negative sentiment(0)



*Figure 2 Distribution of tweet length*

#### The tweet distribution shows that shorter tweets (around 20-50 characters) are more common for positive and negative sentiments. However, negative tweets tend to have a slightly higher frequency at most lengths, particularly in the 50-100 character range.

*.*

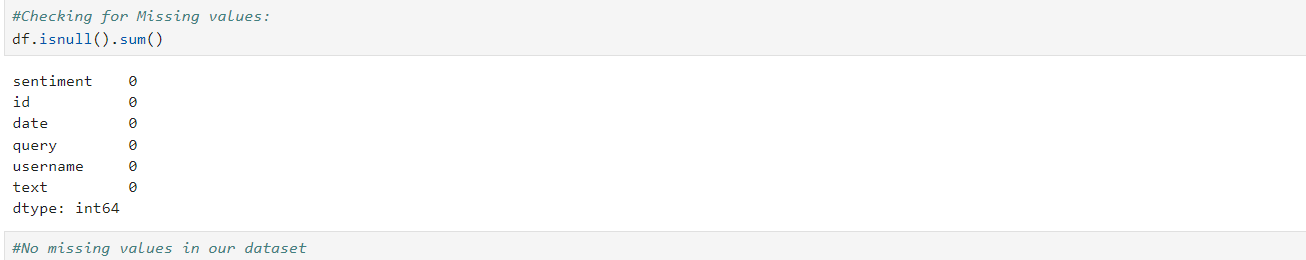
## Data Wrangling:

The Dataset was initially imported from Kaggle and wasn’t pre-handled.

Initially, the dataset had six columns: sentiment, id, date, query, username, and text. Therefore, the sentiment, id, date, query, username, and text columns were irrelevant to the sentiment analysis task and were removed.

### Handling Missing Value

There are no missing values in our dataset.



*Figure 3 Missing Values*

## Text pre-processing:

### Unwanted symbols and punctuations Removal (#Change)

In the text preprocessing, we have removed all the unwanted symbols, URLs, mentions, hashtags, and punctuations and prepared only the textual data.

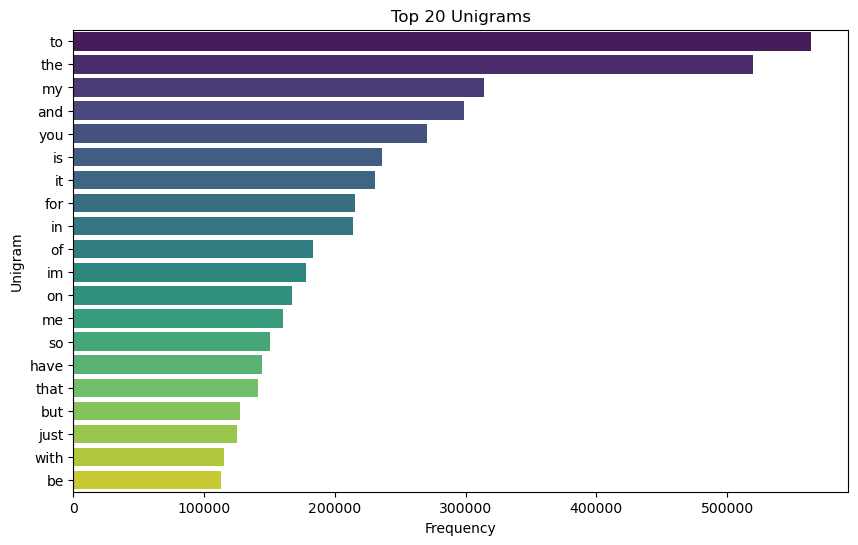
**Column Renaming**: The columns were renamed for clarity:

* sentiment, id, date, query, username, text

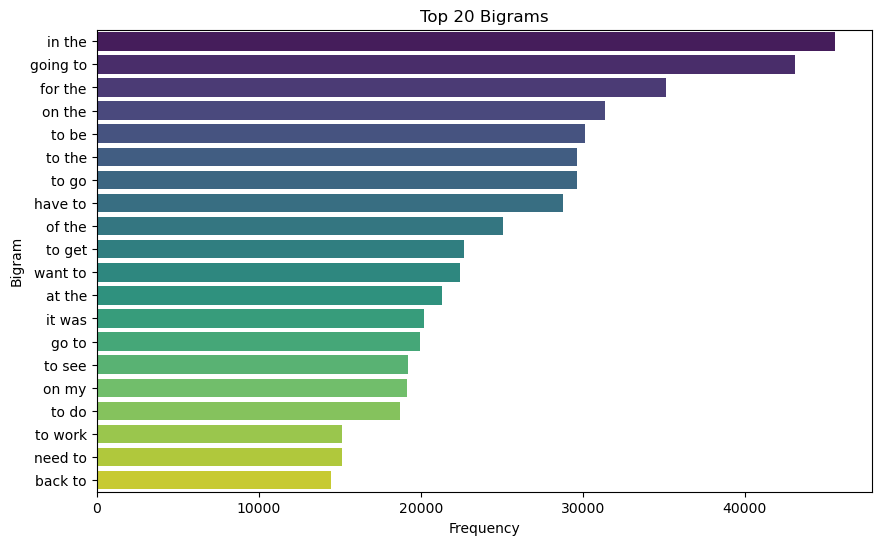
We also use a split, lower, sub, and join techniques to eliminate the unique characters and punctuation. Then, we passed the cleaned text to a corpus for further analysis.

## Unigram and Bigram Analysis

The top 20 most frequent unigrams and bigrams were extracted and visualized using bar plots. This analysis highlighted the importance of specific phrases in sentiment classification.



*Figure 4 Top 20 unigrams*



*Figure 5 Top 20 Biigrams*

## 

## 

## 

## 

## 

## 

## 

## 

## **Word Cloud**

Word cloud, also known as tag cloud, is common. A word cloud is a pictorial representation of the frequency distribution of words or tags in the text data. Word clouds were generated to visualize the most common words in positive and negative tweets. Common words like "thanks" and "thank you" were frequently associated with positive sentiments, while words expressing dissatisfaction or frustration appeared more in negative sentiments. As per the project requirement, the word cloud for the dataset developers are working on has been provided below for both Bag of Words and TF-IDF vectorization techniques. This word diagram helps find much more insight in visualized form. In which the size of each word in the word cloud depicts the frequency in the input or the dataset we provided. This method of visualizing the word in the cloud eases the qualitative understanding of the words extracted by the vectorization method from the dataset. This word cloud was formed after text processing was done for the original dataset. Further processing can be done further processing after looking at the word cloud, which is out of scope for this project.

 *Figure 6 Word Cloud after the Stemming and Lemmatization (NLTK)*

## Vectorization:

Once preprocessing was done, the vectorization technique was applied to each word in full\_review, corpus, and for only one review to get the weight of each word. The strategies used for this task are bags of words (BOW) and TF-IDF. For each method, word cloud representation was worked on to visualize the recurrence of each word. The point is to fabricate classification models on the data by utilizing different vectorization techniques and thoroughly analyze and complete the best strategy.

### Bag of Words

The Bag of Words model was initially considered for vectorizing the text data but was later replaced by TF-IDF vectorization for better performance.

### TF-IDF

The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer was used to convert the cleaned text data into numerical vectors. The vectorizer was set to consider a maximum of 50,000 features and used unigrams (single words) for feature extraction.

# Modelling

After all the preprocessing steps are completed in the whole dataset, the dataset is split into 80% and 20%. 80% of the data will be utilized for the training aspect of the project, while the remaining 20% will be used for the testing purpose. Several machine learning models were trained on the dataset to classify the sentiment of tweets. The models were evaluated using accuracy and other performance metrics like precision, recall, and F1-score.

*Precision* is a measure of how many of the optimistic predictions made are correct (true positives).

*Recall* is a measure of how many of the positive cases the classifier correctly predicted over all the positive instances in the data. It is also called Sensitivity.

*F1-Score* is a measure combining both precision and recall. It is generally described as the harmonic mean of the two.

## Model 1: Classification Model using the Random Forest

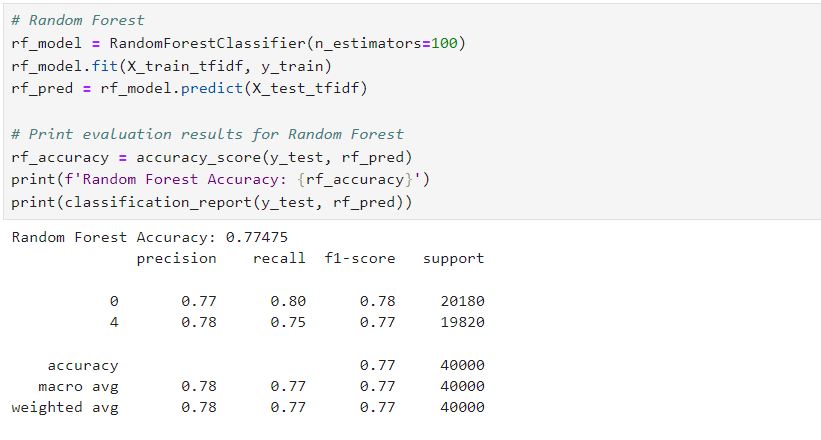
It is a supervised machine learning algorithm that handles complex datasets based on the decision trees that can be used for regression and classification. It is more famous for minimizing overfitting and high predictive accuracy. The parameter n\_estimators defines the number of trees in the forest.

**Accuracy**: 77.47%

**Precision, Recall, F1-Score**:

Negative Sentiment (0): Precision: 77%, Recall: 80%, F1-Score: 78%

Positive Sentiment (4): Precision: 78%, Recall: 75%, F1-Score: 77%



## Model 2: Classification Model using the Naive Bayes

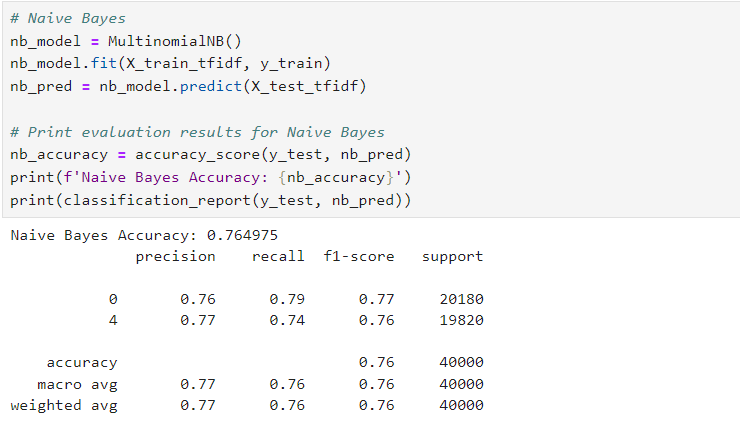
It is also a classification algorithm based on the Bayes theorem and is highly used for text-based classifications. It is a collection of algorithms that work on a single common principle.

**Accuracy**: 76.50%

**Precision, Recall, F1-Score**:

Negative Sentiment (0): Precision: 76%, Recall: 79%, F1-Score: 77%

Positive Sentiment (4): Precision: 77%, Recall: 74%, F1-Score: 76%



## Model 3: Classification Model using Logistic Regression

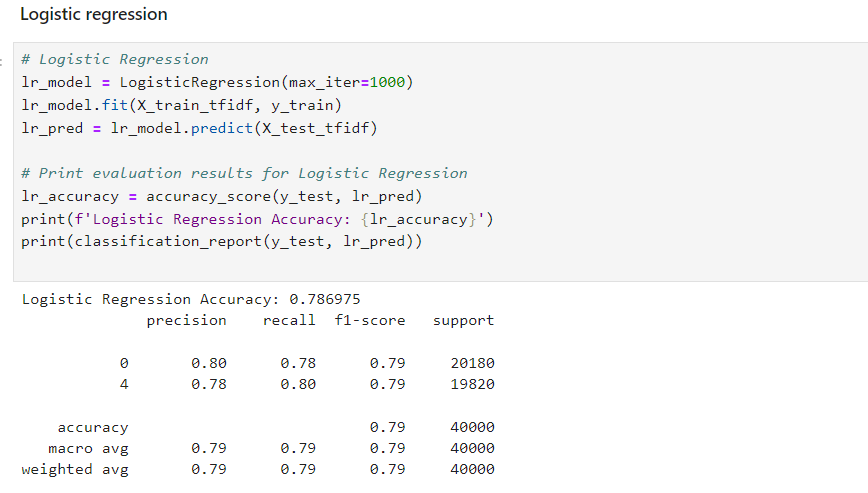
Logistic Regression is also a supervised machine learning algorithm for classifying the target variable. It is well suited when the target variable has only possibilities of two classes, i.e., binary values. It gives the probability that the specific instance belongs to a particular class.

**Accuracy**: 78.70%

**Precision, Recall, F1-Score**:

Negative Sentiment (0): Precision: 80%, Recall: 78%, F1-Score: 79%

Positive Sentiment (4): Precision: 78%, Recall: 80%, F1-Score: 79%



## Model 4: Classification Model using Gradient Boosting

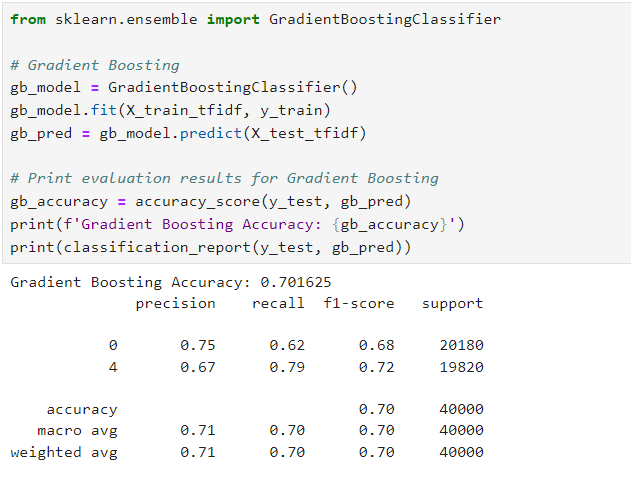
Boosting is an ensemble learning method that trains the model sequentially. Gradient Boosting is a robust boosting algorithm that works for classification and regression tasks.

**Accuracy:** 0.70

**Precision, Recall, F1 Score:**

Negative Sentiment (0): Precision: 75%, Recall: 62%, F1-Score: 68%

Positive Sentiment (4): Precision: 67%, Recall: 79%, F1-Score: 72%



## Modeling Summary

These models were combined with both vectorization methods, such as Bag of Word and TF- IDF, which the developer explored in the initial stage of the project. Technically, four models have been performed in this project.

# Results

**Sentiment Distribution**: A bar graph was plotted to visualize the distribution of sentiments in the dataset, showing an even distribution between positive and negative feelings.

**Tweet Length Distribution**: The length of tweets was analyzed, revealing that shorter tweets (20-50 characters) were more common across both sentiments. However, negative tweets had a slightly higher frequency across most lengths.

**Correlation Heatmap**: A heatmap was generated to explore the correlation between sentiment and tweet length. The results showed no significant linear correlation between the two.

# Metrics of the Models

Metrics is an evaluation of the measures of the model performance in a quantitative manner. As the project focuses on classification, this section will focus on the metrics related to the classification model. The project developers analyze all the metrics available for models formed with the different Bag of Word and TF-IDF vectorization.

Metrics used in this project are Accuracy, Precision, Recall, and F1 Score. Although ultimately, these metrics depend on the values True Positive(TP), True Negative(TN), False Positive(FP), and False Negative(FN). The insight we can gain from the different metrics is different.

All the metrics results will be summarized in the report below. However, a metrics type can be visually represented for classification-related analysis. So, the next section will look into the confusion matrix for each type of vectorization with the two models developed.

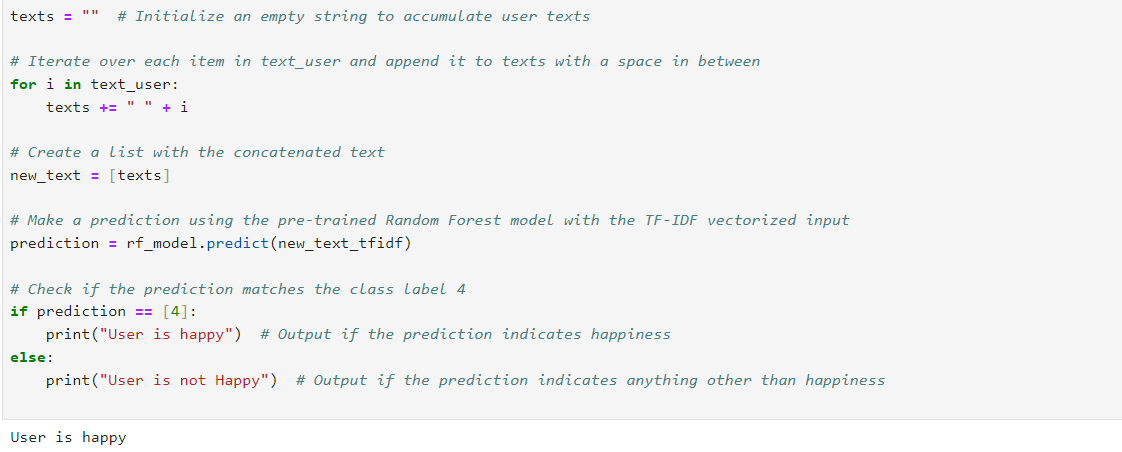
|  |  |  |  |
| --- | --- | --- | --- |
| Classification Algorithms | Accuracy | F1 score | Precision |
| Logistic Regression | 78.70 | 0.79 | 0.80 |
| Naive Bayes | 76.50 | 0.77 | 0.76 |
| Random Forest | 77.47 | 0.78 | 0.77 |
| Gradient Boosting | 70.16 | 0.68 | 0.75 |

# Chatbot Implementation

To start a chat session, the start\_chat method is called, where the user interacts with the bot, and the session continues until the user types "exit." Afterward, the entire conversation is displayed.

This chatbot can be expanded with additional features, such as more complex response handling or integration with sentiment analysis tools.

Predicting messages from the Chatbot:



# Discussion

# In this sentiment analysis task, the logistic regression model outperformed the Naive Bayes and Random Forest models. This result suggests that a linear model like logistic regression, combined with TF-IDF vectorization, is well-suited for binary text classification tasks. The Random Forest model, while powerful, may have been affected by the high dimensionality of the data, leading to slightly lower performance.

The exploratory data analysis (EDA) provided insights into the distribution of sentiments and tweet lengths, revealing that negative tweets tend to be slightly longer on average.

Chatbot Integration:

The developed chatbot is a simple, rule-based system that interacts with users based on predefined keywords. While it is effective for fundamental interactions, integrating sentiment analysis capabilities could significantly enhance the chatbot. For instance, by analyzing the sentiment of user messages in real-time, the chatbot could tailor its responses more appropriately, offering more empathetic or encouraging replies when negative sentiment is detected.

Additionally, the chatbot could benefit from learning from past interactions to improve its response quality. By combining the sentiment analysis model with the chatbot, we could develop a more intelligent system capable of understanding and responding to user emotions more effectively, providing a more personalized user experience.

# Conclusions and Future Work

This project provides hands-on training with sentiment analysis, machine learning, and chatbot development. It aims to use the “Sentiment140” dataset to investigate different methods of building and deploying a sentiment analysis chatbot, thus enhancing our knowledge about NLP applications and model deployment in practical situations.

**Model Performance:** Among the models tested, Logistic Regression performed the best with an accuracy of **78.70%**. It also had the best balance between precision and recall for positive and negative sentiments.

**Feature Importance:** The analysis shows that certain words and phrases are strong sentiment indicators. However, tweet length alone is not a reliable predictor.

**Future Work:** This project can be further expanded with the YouTube comments sections to find whether the video receives positive feedback using our chatbot.

# References:

[1] <https://www.kaggle.com/datasets/kazanova/sentiment140>

[2] <https://scikit-learn.org/stable/supervised_learning.html>

[3] <https://www.deeplearning.ai/resources/natural-language-processing/>

[4] <https://pypi.org/project/wordcloud/>

# Acknowledgment

We thank Mr. Pouria Modaresi for his valuable lectures, which helped us gain Python programming knowledge and complete this project successfully. We would like to thank our team members for their efforts.