Principles of Social Media and Data Mining

Analyzing COVID-19 Tweets using Distil BERT and Naive Bayes

Members: (Name – SUID)

Sai Karthik Kosuri - 804667002
Rohith Mekala - 957274137
Sai Sreeram Nachireddi - 611833596
Surya Chakra Mani Kuntha Sai Kapisetti - 214073519
Shyam Sudheer Nadella - 3649231521
Murali Venkata Ratna Sai Gunnam - 296770000
Anoushka Mergoju – 328542442

CONTENTS

1. INTRODUCTION:

The COVID-19 pandemic has had a profound impact on the world, affecting every aspect of daily life. As a result, social media platforms such as Twitter have become an essential source of information and communication for individuals worldwide. People share their thoughts, experiences, and opinions about the pandemic through Twitter, creating an enormous amount of unstructured data. Analyzing this data can provide valuable insights into public opinion, the effectiveness of public health measures, and the overall impact of the pandemic on society.

The project "Analyzing COVID-19 Tweets Using Distil BERT and Naive Bayes Models" seeks to leverage natural language processing (NLP) techniques to analyze and extract meaningful information from tweets about COVID-19. NLP is a branch of artificial intelligence that focuses on the interaction between computers and human language, enabling computers to understand, interpret, and generate human language.

The project uses two main models to analyze the tweet data: a Distil BERT model and a Naive Bayes model.

The Distil BERT model is a simplified version of the BERT (Bidirectional Encoder Representations from Transformers) model, which is a state-of-the-art NLP model. The Naive Bayes model is a simple but effective probabilistic algorithm used for text classification.

By combining these two models, the project aims to improve the accuracy and effectiveness of the analysis, providing more comprehensive insights into the COVID-19 pandemic. The project aims to uncover hidden patterns and trends in the data that can aid researchers and policymakers in better understanding public perceptions and responses to the pandemic. These insights can ultimately contribute to the creation of more effective mitigation efforts.

The project aims to contribute to the ongoing efforts to understand and mitigate the impact of the pandemic on society.

2. DATA:

2.1. Data Gathering:

To gather the data for this project, we utilized a Twitter developer account to access the Twitter API and collect all COVID-related tweets from the beginning of the pandemic in March 2020 up until the end of 2021. The Twitter API allows developers to retrieve data from the social media platform, including tweets, user profiles, and other information. Using specific search parameters, we were able to narrow down the tweets that we collected to those that were related to COVID-19, including keywords such as "COVID," "coronavirus," "pandemic," and others.

We collected a large volume of tweets over the course of the pandemic, and we used various natural language processing techniques to analyze the data and gain insights into how people were talking about and reacting to the pandemic on social media. We also collected additional metadata such as the tweet timestamp, user location, username, and user description, which were available through the Twitter API.

The data collection process was automated using a custom Python script that utilized the Twitter API's streaming capabilities to continuously retrieve new tweets that matched our search query. Overall, the data gathering process was a crucial step in building our COVID-19 sentiment analysis model, and the high volume and diversity of the collected tweets enabled us to generate robust and accurate insights.

2.2. Data Preprocessing –

In the data preprocessing stage, two essential steps were taken to prepare the data for analysis: data wrangling and data cleaning.

2.2.1. Data Wrangling –

Data wrangling is a critical process for interpreting large and complex datasets, whether for analysis or manual review. The goal of data wrangling is to transform raw data into a structured format that is more accessible and easier to understand.

There are several steps involved in data wrangling, which can vary depending on the nature of the data and the objectives of the analysis. One possible approach involves data extraction, where data is extracted from sources such as deep networks or web tables. Schema matching is another important step that involves identifying and resolving any inconsistencies or differences in the data schema, ensuring that the data is consistent and uniform.

Data visualization is another essential aspect of data wrangling, as it allows analysts to explore and understand the data more effectively. Data repair involves identifying and correcting any errors or inconsistencies in the data, while value format conversion is concerned with converting data values to a more standard or uniform format. Entity settlement and merger is another important step that involves consolidating data from multiple sources to create a more comprehensive dataset.

In the context of Twitter data processing, data wrangling is performed to change the format of the raw data obtained from Twitter into a format that is easy to understand. The process of changing the data involves a join query, which combines information from different tables into a single piece of information for processing. The Tweet table and User table are joined so that the information obtained is the Tweet data along with the information from the User who made the tweet. This process of data wrangling ensures that the Twitter data is structured and organized in a way that makes it easier to analyze and understand.

2.2.2. Data Cleaning -

Data cleaning is a crucial step in data preprocessing that involves removing characters or words that are not needed for analysis. In the context of Twitter data, this may include removing punctuation marks, white space, numbers, and capital letters, which can help to make the data more consistent and easier to analyze.

There are typically three stages involved in data cleaning: cleaning, stop-word removal, and stemming. The cleaning stage involves converting characters to lowercase, removing punctuation marks, white space, and numbers, which can help to make the data more consistent and easier to analyze.

Stop-words are words that have less significance in terms of meaning than other tokens, such as "on," "the," and "and." These words can be removed during the stop-word removal stage to improve filtering, indexing, crawling, and final tweeting efficiency. However, there is no universal stop-word list that is used by all natural language processing tools for individual languages, and the list of stop-words may vary depending on the specific context or analysis.

Stemming is another important step in data cleaning that involves reducing words to their common morphological definition called the stem. This process removes the root of derivational and inflectional affixes, resulting in a single form, the stem, for all words that share the same root. In some cases, only suffixes that have been added to the right-hand end of the root are removed. The goal of stemming is to reduce the number of distinct terms and increase the chances of matching the context concept, thereby improving the accuracy and effectiveness of the analysis.

In summary, data cleaning is a crucial step in data preprocessing that involves removing unnecessary characters or words to make the data more consistent and easier to analyze. This may involve cleaning, stop-word removal, and stemming, which can help to improve the accuracy and effectiveness of the analysis by reducing the number of distinct terms and increasing the chances of matching the context concept.

3. APPROACH –

3.1. Preprocessing of Tweets –

Preprocessing of data is important for cleaning actual tweets by users to ensure the success rate of the project. Here are the details of the preprocessing carried out in the project:

1. Cleaning the data: Before starting with the actual preprocessing of tweets, it is important to ensure that the data is clean. For this, the data is checked for null values and duplicates, if any are found, they are removed. This step ensures that the data is error-free and there are no unnecessary entries.

2. Using regular expressions: Regular expressions are used to clean the tweets by removing any unwanted elements such as emojis, retweets, usernames, URLs, websites, and special characters, numbers, and punctuations. This step helps in removing any irrelevant information from the tweet and makes the data more structured.

3. Keeping words from hashtags: Hashtags are used by Twitter users to categorize their tweets and make them more discoverable. When modeling sentiment analysis, it is important to keep the words from hashtags as they can be a major factor in calculating sentiment. Therefore, regular expressions are used to identify any hashtags in the tweet, and if they exist, the hashtag is removed, and the word is kept.

4. Converting everything to lowercase: This step ensures that all the text in the tweets is in the same case, which helps in better analysis and modeling. By converting everything to lowercase, we ensure that words with different cases are treated as the same.

5. Using the Tweet-Preprocessor Module: The Tweet-Preprocessor Module is a Python library specifically designed for cleaning and preprocessing tweets. It is used to clean any leftover junk that might have been missed in the previous steps.

6. Tokenizing all words: Tokenization is the process of breaking down a sentence into individual words. The NLTK Module is used to tokenize all words in the tweets, which helps in analyzing individual words and their respective sentiments.

7. Removing stop words: Stop words are commonly used words in a language that do not add any meaning to the text. Examples of stop words include "or," "from," "them," "does," etc. The NLTK Module is used to remove these stop words from the tweets, which helps in reducing noise in the data and improving the accuracy of the sentiment analysis.

8. Performing stemming: Stemming is the process of reducing a word to its root form. The NLTK Module is used to perform stemming on the tweets, which helps in reducing the number of words in the dataset and improving the accuracy of the sentiment analysis.

9. Dropping short words: Words with a length less than 2 are dropped as they do not add any significant meaning to the text. This step further helps in reducing the noise in the data and improving the accuracy of the sentiment analysis.

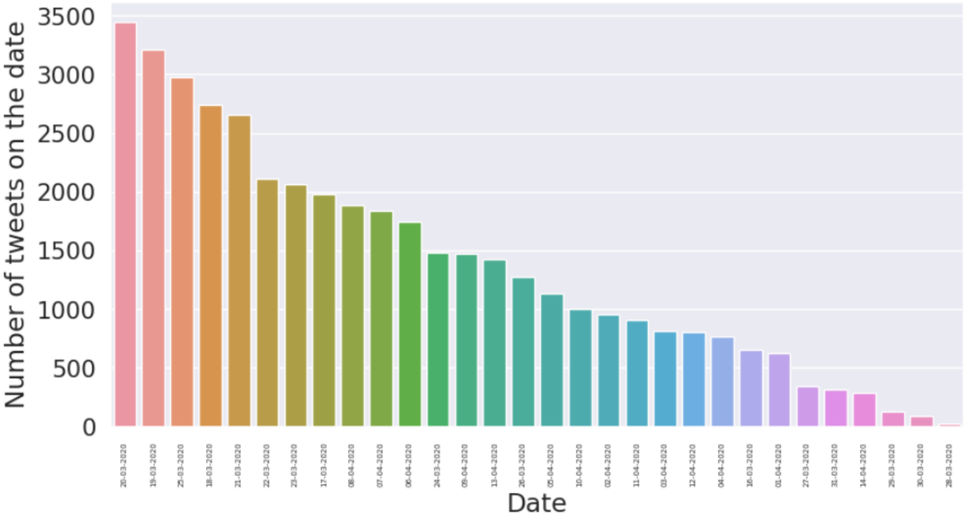
4. DATA ANALYSIS –

On the gathered data, an exploratory data analysis is also performed. Since we have two datasets, both datasets are used for this analysis.

4.1. Frequency of tweets collected per day

We create a graph that displays the daily average for tweets sent out. This demonstrates the level of user activity during certain days.

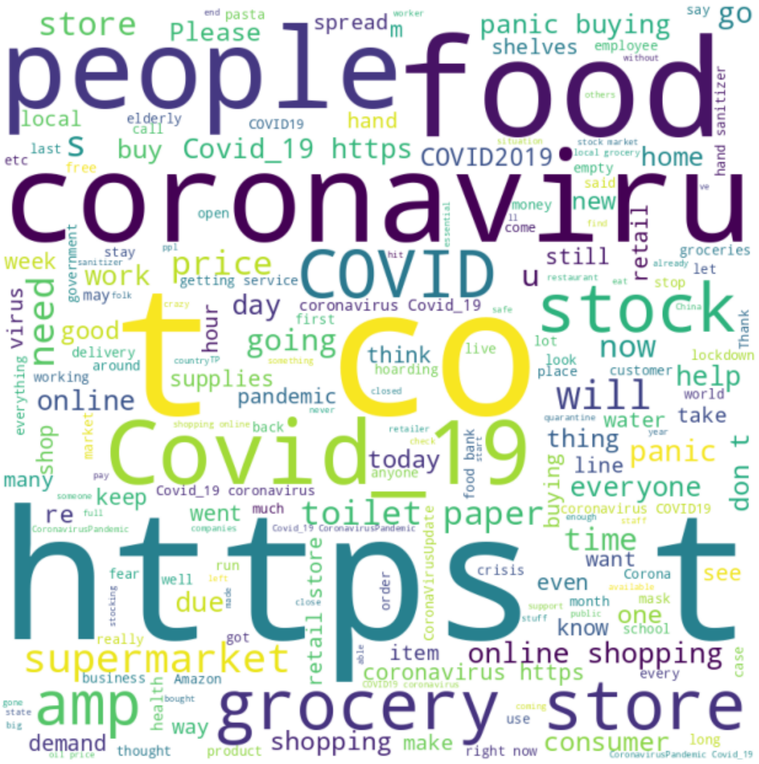
The plot for the same is as displayed below:



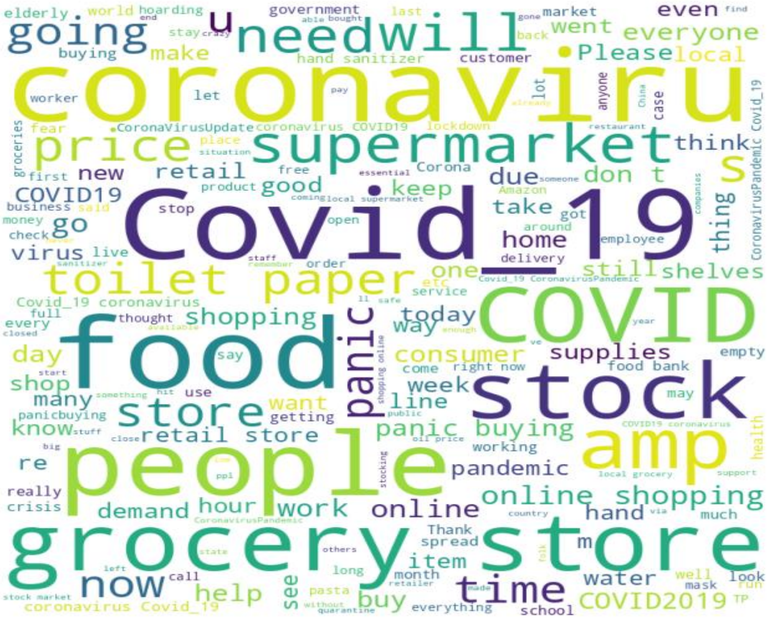
4.2. Word Cloud

The words are visually represented in a word cloud. The word cloud displays the presence of the most widely used and frequent words. Each word's size is determined by how many times it appears in the text used to generate the word cloud. The size of frequent words is larger than that of less frequently used terms.

The word cloud that was generated with URLs included is as follows:



The below word cloud is generated after the removal of URLs:



5. SENTIMENT ANALYSIS –

NLP (natural language processing) enables machines to decode and comprehend human language. Sentiment analysis and emotions analysis are two of the most used NLP approaches. Although both involve identifying emotions, there are significant differences between the two.

A computerized text-processing method called sentiment analysis is used to gauge people's opinions and attitudes. Although frequently utilized in social media data to gauge overall opinions of goods and companies, it can shed light on current affairs and problems, as it compares terms to a lexical dictionary with predetermined emotion scores in place of manually identifying.

Each word in a tweet is assessed separately, with a score, and a total score is returned for each tweet. favorable scores reflect attitudes that are favorable, negative scores reflect attitudes that are negative, and terms that are not found in the related lexicon receive a score of zero.

It is feasible to modify sentiment analysis models so that they go beyond polarity and identify emotions and even intentions, depending on the collection of tasks. The most pertinent types of sentiment analysis are listed below: Sentiment analysis with grades, sentiment analysis based on aspects, intent analysis, and emotion detection.

As we can see, one sort of sentiment analysis is emotion detection. Sentiment analysis can be used for several purposes, such as examining customer reviews, monitoring company reputations, or gauging public sentiment. Understanding the customer's true feelings, however, might not be sufficient in some circumstances. Irony or sarcasm, for instance, might be a significant problem for sentiment analysis programs. Emotional analysis is necessary at that point.

What is emotion detection?

Advanced machine learning algorithms are used in emotion analysis to examine more complex emotions including fear, rage, sadness, love, impatience, and many others.For instance, the negative phrases "I don't like the interface" and "I hate this product" can both be compared. But “hate” and "don't like" have very different emotional implications. This approach makes it easier to gauge the intricacies within an emotion and even examine motivations and inclinations of people.

The ability to recognize emotions has drawbacks and restrictions. Understanding someone's emotional condition can be difficult despite all the experience humans have. Additionally, humans assign labels to the datasets that computers used to detect emotions based on their own subjective observations that may not always be applicable.

However, Emotional recognition shows more complicated emotions, which can offer deeper, more insightful understandings.

6. MODEL SELECTION –

The goal of this project is to categorize tweets. The classification of tweets as favorable, neutral, or negative was required. The Naive Bayes Model, which is recognized to be straightforward but effective for classification, as well as the distilBERT model, a neural network proven to be efficient are used here for classification.

6.1. The Naive Bayes Model

The Naive Bayes model is a probabilistic algorithm that uses Bayes' theorem to make predictions or classifications. It is widely used in machine learning and natural language processing tasks, particularly in situations where the number of features is large and the computation complexity is a concern.

To understand how the Naive Bayes model works, let's break down its key components:

1. Bayes' Theorem: Bayes' theorem is a fundamental principle in probability theory. It calculates the probability of a hypothesis (H) given evidence (E) using the following formula:

$$P(H|E) = (P(E|H) * P(H)) / P(E)$$

- $P(H|E)$: The probability of hypothesis H given evidence E (posterior probability).
- $P(E|H)$: The probability of evidence E given hypothesis H (likelihood).
- $P(H)$: The probability of hypothesis H being true without considering any evidence (prior probability).
- $P(E)$: The probability of evidence E occurring.

Assumption of Conditional Independence: The Naive Bayes model assumes that all features are conditionally independent of each other given the class label. In other words, it assumes that the presence or absence of one feature does not affect the presence or absence of any other feature.

Now, let us see how the Naive Bayes model applies Bayes' theorem and the assumption of conditional independence to make predictions or classifications:

1. Training Phase:

- Gather a labeled training dataset, where each example consists of a set of features and a corresponding class label.
- Calculate the prior probability $P(H)$ for each class label in the training dataset.
- Calculate the likelihood $P(E|H)$ for each feature given each class label. This involves estimating the probability distribution of each feature for each class label.

2. Prediction/Classification Phase:

- Given a new example with a set of features, calculate the posterior probability $P(H|E)$ for each class label using Bayes' theorem.

- Use the assumption of conditional independence to calculate the overall probability of the features P(E) without considering the class label.
- Compare the posterior probabilities for each class label and choose the class label with the highest probability as the predicted class for the new example.

The Naive Bayes model's strength lies in its simplicity and computational efficiency. Despite its "naive" assumption of conditional independence, it often performs well in practice, especially in text classification tasks like spam detection, sentiment analysis, and document categorization. However, it may not be suitable for problems where the independence assumption is violated or when the relationships between features are essential for accurate predictions.

6.2. The DistilBERT Model

The DistilBERT model is a deep learning algorithm that utilizes a pre-trained transformer-based neural network. It is a variant of the popular BERT (Bidirectional Encoder Representations from Transformers) model that aims to reduce the model's size and computational requirements while maintaining its performance to some extent.

To understand how the DistilBERT model works, let's break down its key components and steps:

1. Transformer-Based Neural Network: The core architecture of the DistilBERT model is based on the transformer model. Transformers are deep learning models designed to handle sequential data, such as text, by capturing the relationships and dependencies between words or tokens in the input sequence.

2. Pre-Training: The DistilBERT model is pre-trained on a large corpus of text data using an unsupervised learning approach. During pre-training, the model learns to predict missing words in a sentence, known as masked language modeling (MLM). It also learns to understand the relationships between different sentences in a document, known as next sentence prediction (NSP).

- Masked Language Modeling (MLM): The model is given a sentence with some of its words randomly masked, and it learns to predict the original masked words based on the context provided by the surrounding words.

- Next Sentence Prediction (NSP): The model is trained to determine whether two sentences appear consecutively in the original document or if they are randomly paired sentences from different parts of the corpus.

By pre-training on a large amount of text data, the DistilBERT model learns to capture a rich representation of the language and develops a strong understanding of the contextual relationships between words.

3. Distillation: Once the pre-training is complete, the DistilBERT model undergoes a distillation process. Distillation involves transferring knowledge from a larger, more computationally expensive model (such as BERT) to a smaller model (DistilBERT) while attempting to retain a significant portion of the performance.

Distillation typically involves training the smaller model (DistilBERT) on the same or similar tasks as the larger model. However, during distillation, the model is trained to mimic the behavior of the larger model rather than independently learning from scratch. This allows the distilled model to benefit from the rich representations and understanding acquired by the larger model.

The distillation process helps reduce the size and computational requirements of the model, making it more practical and efficient for deployment in various applications.

4. Fine-Tuning: After distillation, the DistilBERT model can be further fine-tuned on specific downstream tasks. Fine-tuning involves training the model on a smaller task-specific dataset that is labeled for the target task, such as sentiment analysis, named entity recognition, question-answering, or text classification.

During fine-tuning, the model's parameters are adjusted to adapt to the specific task at hand. The model learns to generalize from the labeled examples in the training dataset and make accurate predictions on new, unseen data.

Overall, the DistilBERT model offers a computationally efficient alternative to the larger BERT model while retaining a significant portion of its performance. It has been widely adopted in natural language processing tasks and has proven to be effective in various domains, enabling faster inference and deployment of transformer-based models in real-world applications.

7. MODEL EVALUATION –

Model evaluation is an essential step in assessing the performance of a classification model. In our case, we are evaluating a model's performance on a binary classification problem, where we have defined one class as positive and the other as negative.

Here is a brief explanation of the evaluation metrics you mentioned:

1. **Confusion Matrix**: A confusion matrix is a table that helps you understand the types of errors made by the classifier. It presents a summary of the model's predictions and the actual class labels. It includes four components:
    - True Positive (TP): The model correctly predicted a positive class.
    - False Positive (FP): The model incorrectly predicted a positive class when the actual class was negative.
    - True Negative (TN): The model correctly predicted a negative class.
    - False Negative (FN): The model incorrectly predicted a negative class when the actual class was positive.

2. **Accuracy**: Accuracy is a metric that measures the percentage of correct predictions made by the model. It is calculated by dividing the number of correct predictions (TP + TN) by the total number of predictions. However, accuracy alone may not be sufficient if the dataset is imbalanced or the cost of false positives and false negatives differs.

3. **Recall**: Recall, also known as sensitivity or true positive rate, measures the ability of the classifier to find all the positive instances correctly. It is calculated by dividing the number of true positive predictions (TP) by the sum of true positives and false negatives (TP + FN).

4. **Precision**: Precision measures the proportion of correct positive predictions out of all positive predictions made by the model. It is calculated by dividing the number of true positive predictions (TP) by the sum of true positives and false positives (TP + FP).

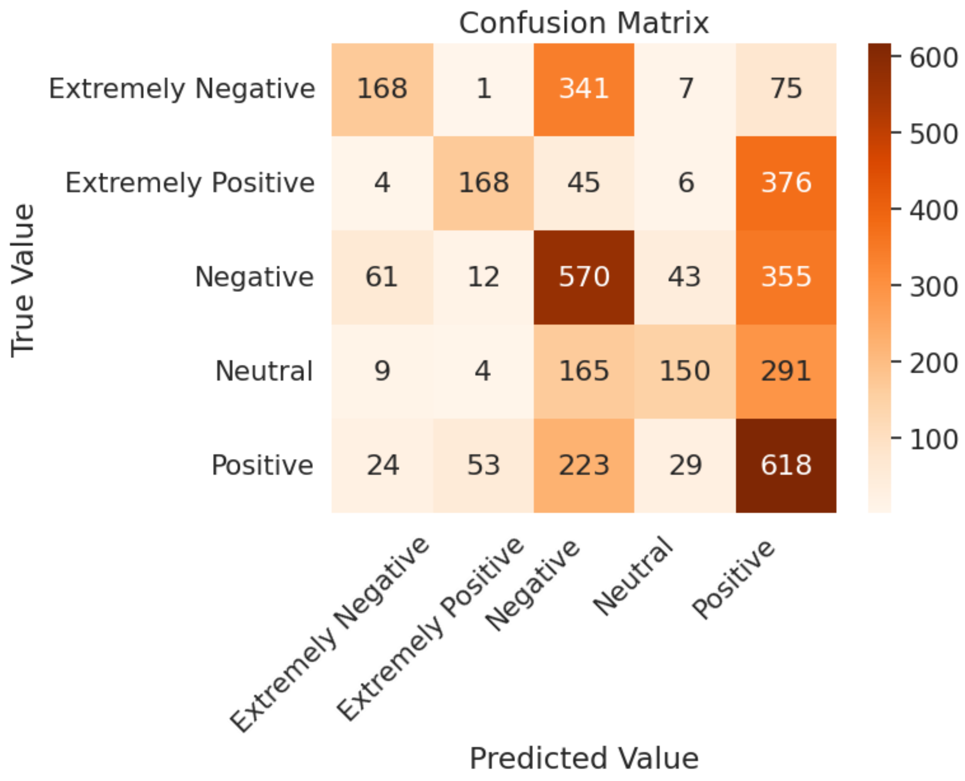
5. **F1 Score**: The F1 score is the harmonic mean of precision and recall. It provides a balanced measure that takes both false positives and false negatives into account. It is calculated using the formula: F1 = 2 * (Precision * Recall) / (Precision + Recall).

By considering the confusion matrix, accuracy, recall, precision, and F1 score, you can gain insights into different aspects of the model's performance. These metrics provide a comprehensive evaluation of how well the model is performing in terms of correct predictions, identifying positive instances, and balancing precision and recall.

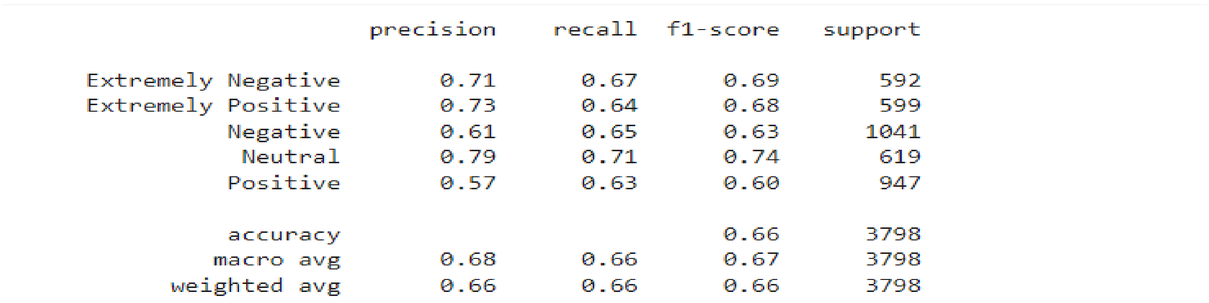
8. RESULTS –

Naive Bayes model

The confusion matrix generated for 5 label sentiment analysis with Naive Bayes model
Accuracy=0.44



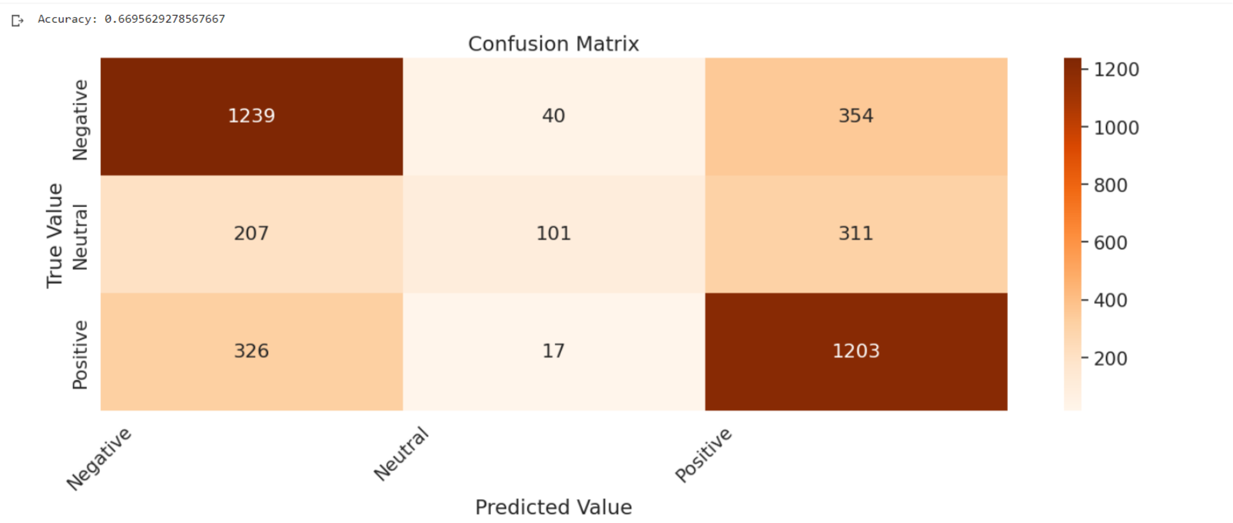
The Recall, Precision and f1-score of the 5-label sentiment analysis using Naive Bayes model



```
                    precision    recall  f1-score   support

Extremely Negative       0.71      0.67      0.69       592
Extremely Positive       0.73      0.64      0.68       599
          Negative       0.61      0.65      0.63      1041
           Neutral       0.79      0.71      0.74       619
          Positive       0.57      0.63      0.60       947

          accuracy                           0.66      3798
         macro avg       0.68      0.66      0.67      3798
      weighted avg       0.66      0.66      0.66      3798
```

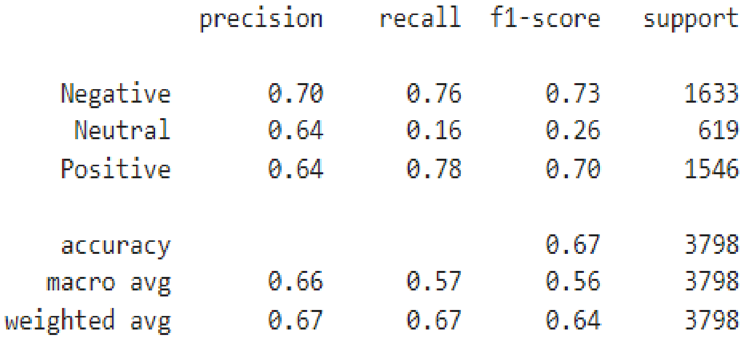
The confusion matrix generated for 3 label sentiment analysis with Naive Bayes model

Accuracy=0.67



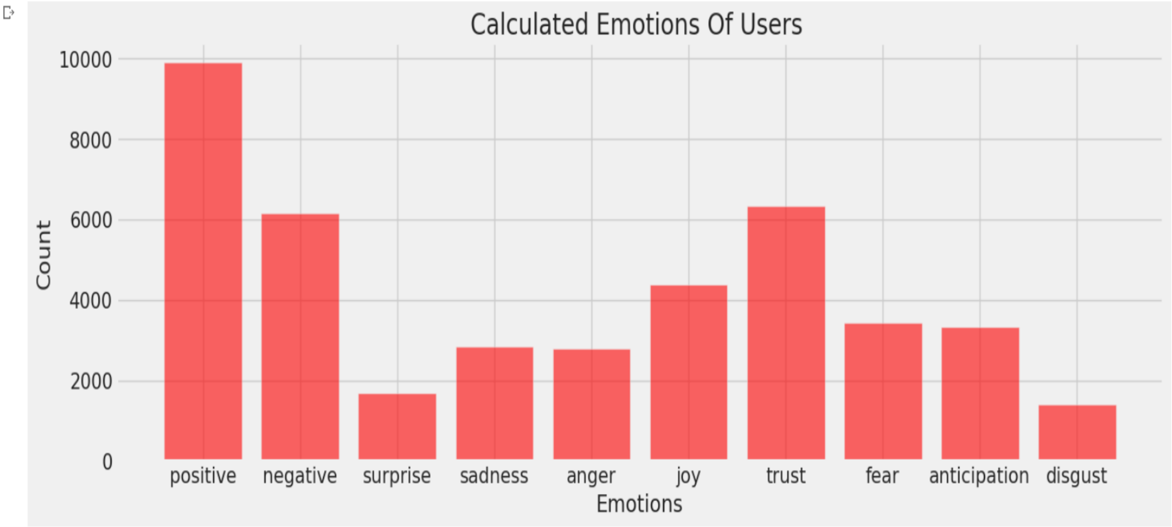
```
Accuracy: 0.6695629278567667
```

The Recall, Precision and f1-score of the 3-label sentiment analysis using Naive Bayes model



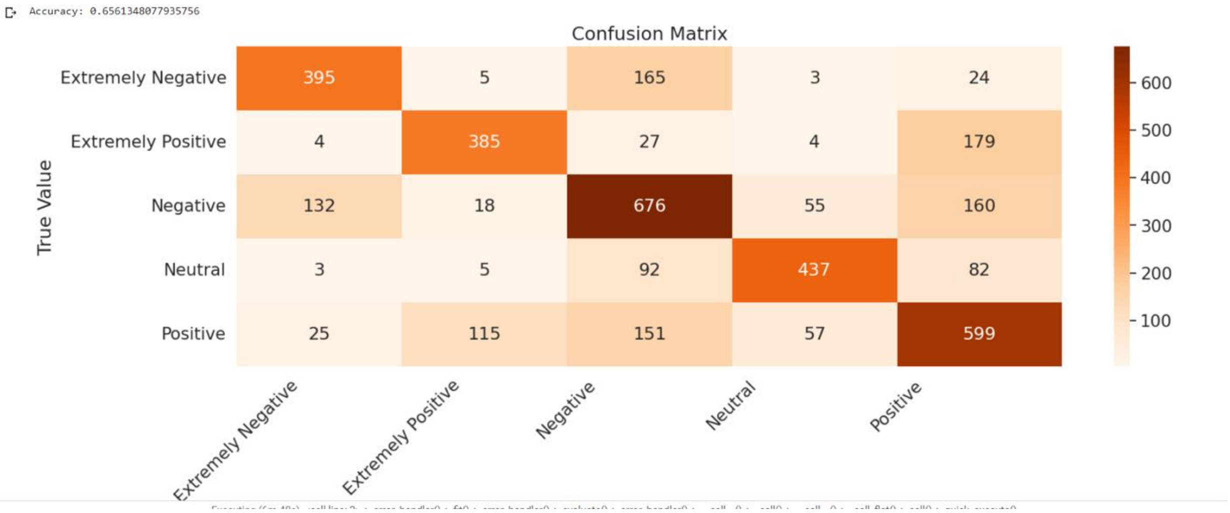
```
              precision    recall  f1-score   support

    Negative       0.70      0.76      0.73      1633
     Neutral       0.64      0.16      0.26       619
    Positive       0.64      0.78      0.70      1546

    accuracy                           0.67      3798
   macro avg       0.66      0.57      0.56      3798
weighted avg       0.67      0.67      0.64      3798
```

Histogram visualizes the calculated emotions of people


Calculated Emotions Of Users
Count
10000
8000
6000
4000
2000
0
positive
negative
surprise
sadness
anger
joy
trust
fear
anticipation
disgust
Emotions

The DistilBERT Model

The confusion matrix generated for 5 label sentiment analysis with **the** DistilBERT Model
Accuracy=0.656

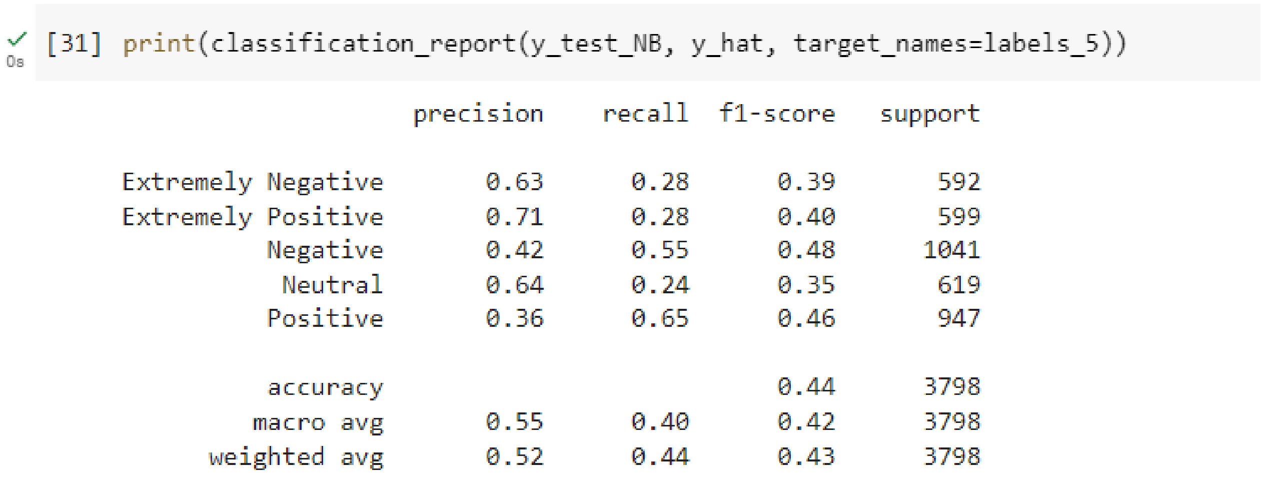


Accuracy: 0.6561348077935756

Confusion Matrix

True Value	Extremely Negative	Extremely Positive	Negative	Neutral	Positive
Extremely Negative	395	5	165	3	24
Extremely Positive	4	385	27	4	179
Negative	132	18	676	55	160
Neutral	3	5	92	437	82
Positive	25	115	151	57	599

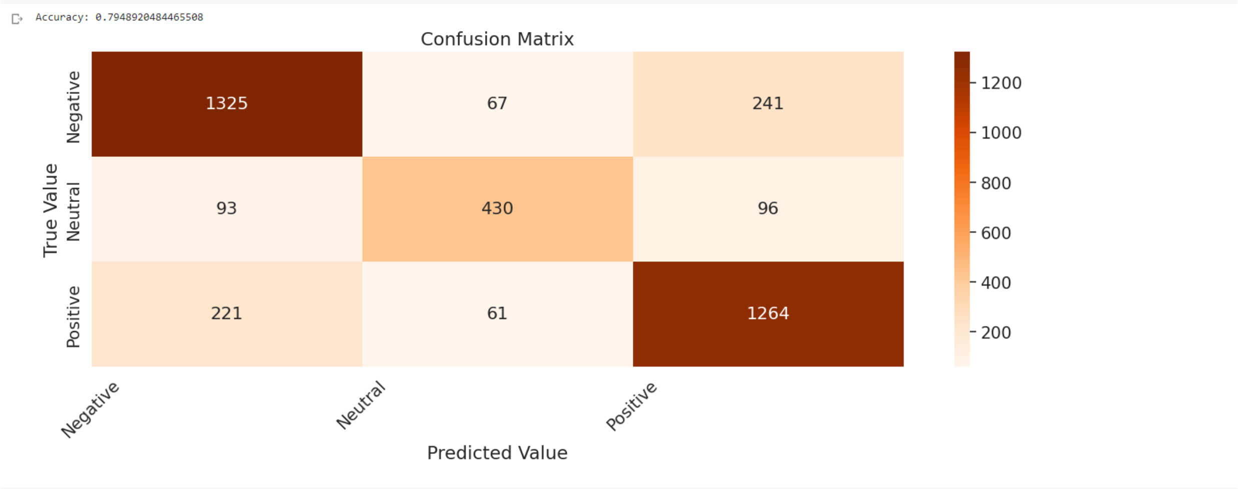
The Recall, Precision and f1-score of the 5-label sentiment analysis using the DistilBERT Model



```
[31] print(classification_report(y_test_NB, y_hat, target_names=labels_5))

                    precision    recall  f1-score   support

Extremely Negative       0.63      0.28      0.39       592
Extremely Positive       0.71      0.28      0.40       599
          Negative       0.42      0.55      0.48      1041
           Neutral       0.64      0.24      0.35       619
          Positive       0.36      0.65      0.46       947

          accuracy                           0.44      3798
         macro avg       0.55      0.40      0.42      3798
      weighted avg       0.52      0.44      0.43      3798
```

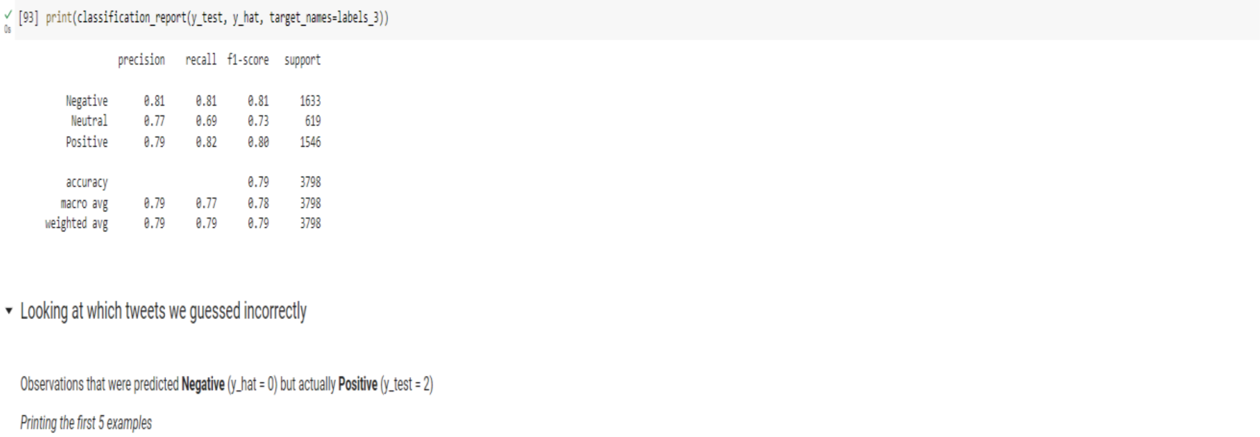
The confusion matrix generated for 3 label sentiment analysis with **the** DistilBERT Model

Accuracy=0.794



```
Accuracy: 0.7948920484465508
```

The Recall, Precision and f1-score of the 3-label sentiment analysis using the DistilBERT Model



```
[93] print(classification_report(y_test, y_hat, target_names=labels_3))
```

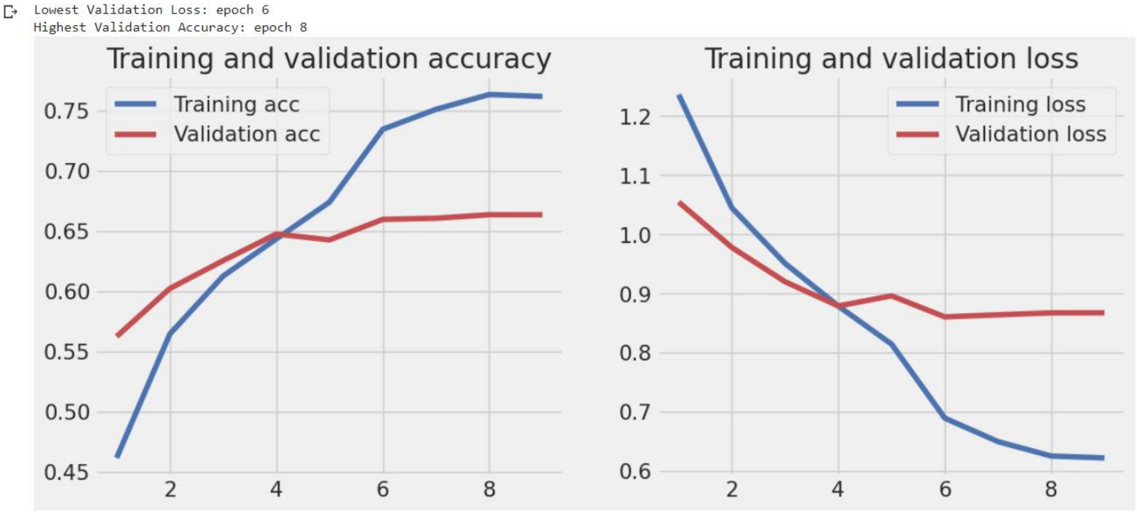
```
              precision    recall  f1-score   support

    Negative       0.81      0.81      0.81      1633
     Neutral       0.77      0.69      0.73       619
    Positive       0.79      0.82      0.80      1546

    accuracy                           0.79      3798
   macro avg       0.79      0.77      0.78      3798
weighted avg       0.79      0.79      0.79      3798
```

▾ Looking at which tweets we guessed incorrectly

Observations that were predicted **Negative** (y_hat = 0) but actually **Positive** (y_test = 2)

Printing the first 5 examples

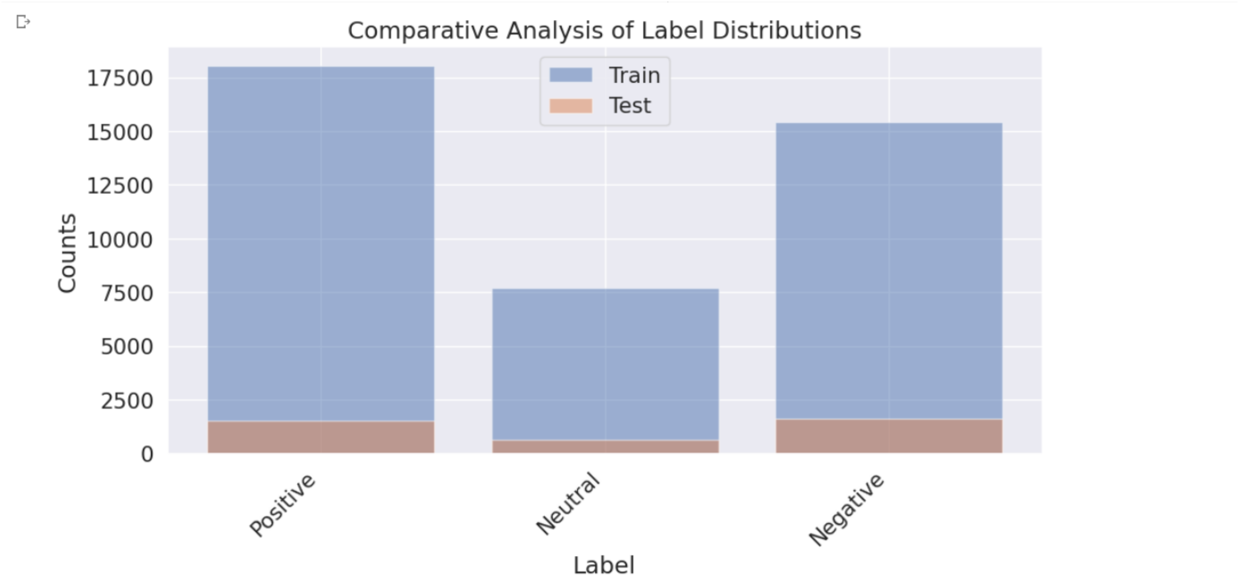
This graph shows trends of accuracy and loss for training and validation



```
Lowest Validation Loss: epoch 6
Highest Validation Accuracy: epoch 8
```

Loading the model with the best validation accuracy

Comparative analysis of label distribution across test and train data sets



9. CONCLUSION –

- The project aimed to understand people's sentiments during the Covid-19 pandemic using tweets.
- Two models were used for sentiment analysis - Naive Bayes and DistilBERT-based neural network model.
- The baseline NB model achieved an accuracy of 63.1%, while the DistilBERT model achieved an accuracy of 79.3%. This shows that the neural network-based model performed better than the traditional machine learning approach.
- Callback functions were used to improve the training process of the neural network-based model. This included early stopping, retrieving the best model weights, and reducing the learning rate on plateau.
- The confusion matrix was used to evaluate the performance of the models. The DistilBERT model had a better accuracy and precision compared to the NB model.
- Preprocessing of tweets was an important step in this project to remove noise and unwanted information. Regular expressions were used to remove emojis, retweets, URLs, special characters, and punctuation. NLTK module was used to tokenize words, remove stop words, perform stemming, and drop words with a length less than 2.
- The analysis of the tweets provided insights into people's sentiments during the pandemic. It helped in understanding the emotions, concerns, and attitudes of people towards the pandemic and related issues.

10. FUTURE WORK –

This work demonstrates that social media analytics is a field with a bright future. There are a lot of different approaches that can be applied on similar experiments. In this section we discuss different approaches, solutions and tools that can be applied in social media analytics. Firstly, sentiment analysis can be addressed using a different perspective. We used Distil BERT model in this work, but There are cases that SVM might perform better than Distil BERT if we choose to base our analysis on other parameters.

We can also try to address this problem using rule-based approaches. Such approaches are not effective for unstructured data, but in our case the data are structured, so this method might be applied with success. We must further experiment to determine which the best way to approach the problem is. In addition, the factor of importance for every tweet is not calculated. There are people that are considered influential, and their tweets reach more people. Also, there are some tools that can help to perform sentiment analysis more easily.

One of the tools we could use is Lexicons. A very simple example of a Lexicon is a dictionary. In our case, we needed a lexicon that defines words as positive or negative. The intensity of a word is also important. The most used Lexicon is Sent WordNet. The analysis of the tweets provided insights into people's sentiments during the pandemic. It helped in understanding the emotions, concerns, and attitudes of people towards the pandemic and related issues. Our method was proven to produce accurate results for sentiment analysis during the pandemic. There are many features and approaches that can improve our method and we plan to enhance our work in the future.

11. REFERENCES -

1. "Multilingual Sentiment Analysis on Social Media: An Empirical Study Using Twitter Data" by A. Ghosal et al. (2020)

2. "Twitter Sentiment Analysis for Predicting Stock Market Trends" by S. Roy et al. (2021)
3. "Sentiment Analysis of Online Political Discourse Using Twitter Data" by D. Das et al.

(2020)

4. "Gender-Based Sentiment Analysis of Tweets Using Machine Learning Techniques" by S. S. Akhtar et al. (2021)