Sentiment Analysis of Mental Health Tweets

UNIVERSITY OF ESSEX
School of Computer Science and Electronic Engineering
Capstone project dissertation in Computing

Author:
Sule Selda Ozturkmen
2005435

Supervisor:
Haider Raza

Second Assessor:
Vishwanathan M. Mohan

ACKNOWLEDGEMENTS

I am very thankful to my advisor Haider Raza for helping and motivating me throughout the course in accomplishing my final project.

Abstract

Twitter is an online social media platform known for its rich source of data, often used to communicate people's emotions, aiding individuals in expressing their opinions across different domains through text instead of voicing them orally. People who suffer from anxiety, depression, and other variety of mental illnesses often tend to find a certain comfort in posting tweets regarding their feelings online, which has the potential for the improvement of mental health and acts as a source of outlet (Berry, 2017). Across all social media platforms, users can access a variety of shared data, with the use of increasingly innovative modern advanced technologies offering a wide range of tools for effective data analysis. This paper focuses on analyzing data and retrieving sentiment from a dataset publicly available on Kaggle through the use of natural language processing (NLP), text analysis and machine-learning techniques. Through such techniques, I intend to find an efficient method to detect depressive tweets and analyze the mental well-being of the corresponding users' using sentiment analysis. This aims to extract subjective information, which will then automatically categorize the tweets into positive, negative, and neutral sentiment classifications.

Contents

Context

1.1 Background reading

In today's generation, social media plays a crucial role in our daily lives by giving us a forum to interact with others, share our knowledge and voice our opinions on certain topics. Social media platforms, such as; Youtube, Twitter, Instagram, and many more, have given individuals opportunities to utilize the internet as a tool to communicate with a global audience expressing their ideas and thoughts. According to statistics done in 2022, 59.4 percent [1] of the world's population uses one of the social media platforms, which comes to show that it is continuously having an impact on shaping our social interactions and influencing our daily routines.

Among various social media platforms, Twitter has emerged as a powerful tool for communication and information sharing. With approximately 450 million monthly active users as of 2022 [2], users share their opinions, views, and personal experiences. It has become a virtual hub for users to engage with one another and share information. Micro-blogging websites, just like Twitter, carry an immense amount of data available for sentiment analysis which can be very beneficial in many aspects.
Twitter allows users to freely share their thoughts in short messages, with their followers or to the public, known as "tweets". It is a very accessible platform and with its simplicity, it's easily usable. Hence, millions of users use Twitter, every day, as the main platform to share their thoughts on any topic. This includes tweets related to mental health, where anonymity makes people feel comfortable enough to share their mental health issues, emotions, and thoughts related to it.

The more tweets are shared, the more awareness is being raised to a lot of other people experiencing similar mental health issues. There are many communities on this platform that are willing to help you seek support, and these communities can be seen as a safe space for many users. Therefore, tweets about mental health have been gradually increasing.

In the context of mental health, Twitter has become an important platform for individuals to share their experiences. Millions of people use Twitter to talk about their struggles, mental health issues, and share their personal encounters related to it. This platform has become a powerful source of data for researchers

to study the patterns and trends related to mental health. Many professionals, researchers, and other experts provide useful information along with their knowledge for people who are trying to educate themselves and be more aware of mental health. The potential of Twitter in this field has motivated many researchers to explore the use of sentiment analysis techniques to analyze related tweets.
The sentiment analysis of these tweets can provide valuable insights into the mental health of individuals and communities by identifying their emotions and feelings. Experts understand the challenges and struggles faced by individuals with mental health issues and can help develop effective interventions and support mechanisms.

The power of Twitter in the field of mental health has also motivated many organizations to develop innovative solutions to analyze and utilize Twitter data for mental health research. The availability of large amounts of data on Twitter, has enabled the development of machine learning models.

Today, we have technology advanced enough to detect human emotions with the use of Machine learning and Natural Language Processing (NLP). his has been very effective in identifying sentiments expressed in the posted tweets related to mental health and provides insight into a user's emotion by suggesting whether it's positive, negative, or neutral.

1.2 Motivation

The motivation behind the project comes from the growing awareness of mental health issues and the role social media can play in solving them. Mental health is always a sensitive topic and open discussion is always limited. However, social media platforms like Twitter provide a safe place for people to open up about their experiences and thoughts about mental illness. With millions of users tweeting about mental health issues, it has become an important source of information for researchers to identify and analyze the condition.

Sentiment analysis, which is the process of extracting information from the text, can be used for emotional tweets divided into positive, negative and neutral groups. By analyzing psychological tweets, researchers can identify patterns that can help psychologists better understand the difficulties people experience with their mental health problems. For example, by analyzing negative emotions, researchers can identify problems people face, such as stigma, lack of access to mental health services, and difficulty reporting mental health problems.

Moreover, sentiment analysis can be useful for predicting mental health trends, identifying potential triggers, and even providing early interventions. For instance, if a particular event or topic triggers negative sentiments in mental health tweets, it can indicate that people are struggling with related issues. By identifying such patterns, mental health professionals can provide targeted support and interventions to people who are at risk.

The use of machine learning models in sentiment analysis has been gaining popularity due to their ability to handle large amounts of data and identify complex patterns. However, the accuracy of the results depends on the quality of the data, the selection of features, and the choice of the model. Therefore, the use of appropriate techniques for data cleaning and feature extraction is crucial for accurate sentiment analysis.

1.3 Literature Review

In the field of sentiment analysis and mental health, there have been numerous studies and projects that have been conducted. Some of the related work in this area are:

1. "Identifying Mental Health Risks in Social Media Texts" by Bucci et al. This study used sentiment analysis and natural language processing techniques to analyze social media data and identify individuals at risk for mental health issues.
2. "Predicting mental health of depression sufferers using social media" by Coppersmith et al. This study analyzed Twitter data from individuals with depression and used machine learning models to predict their mental health status.
3. "A Study on Mental Health Prediction using Machine Learning" by Ravindranath et al. This study used machine learning models to predict the mental health status of individuals based on their social media activity.
4. "Analyzing mental health through social media" by Jain et al. This study analyzed social media data from individuals with mental health issues and used sentiment analysis to identify common themes and emotions.
5. “Twitter Sentiment Analysis using Supervised Machine Learning” by Ajitkumar Shitole et al. The system intends to carry out sentiment analysis over tweets gathered from the twitter dataset.

6. Pang and Lee (2002) proposed the framework, where an assessment can be positive or negative was discovered by the proportion of positive words to total words. Later in 2008, the creator built up a methodology in which tweet results can be chosen by term in the tweet.
7. Another study on twitter sentiment analysis was done by Go et al. who stated the issue as a two-class classification, meaning to characterize tweets into positive and negative classes. M. Trupthi, S.Pabboju, and G.Narasimha proposed a system that mainly makes use of Hadoop. The data is extracted using SNS services which are done using twitter's streaming API.
8. The Multi-Perspective-Question-Answering (MPQA) is an online resource with such a subjectivity lexicon which maps a total of 4,850 words according to whether they are "positive" or "negative" and whether they have "strong" or "weak" subjectivity.

These studies have shown the potential of sentiment analysis and machine learning in predicting and identifying mental health issues through social media data. However, there are also limitations and challenges, such as the need for large and diverse datasets, the potential biases in social media data, and the ethical concerns around privacy and consent.

1.4 Domain Introduction

Sentiment analysis, also known as opinion mining, is the process of using natural language processing, text analysis, and computational linguistics to identify and extract subjective information from textual data, such as social media posts, customer reviews, and news articles. The aim of sentiment analysis is to determine the emotional tone, attitudes, and opinions expressed in a text, whether positive, negative, or neutral. The use of this has become increasingly popular in recent years, particularly in the field of social media analysis.

The motivation behind sentiment analysis stems from the growing importance of social media as a communication platform. With the rise of social media, people are sharing their thoughts and opinions on a wide range of topics more than ever before. These platforms such as Twitter, Facebook, and Instagram, have become virtual soapboxes, allowing people to express their views and connect with others who share similar interests. As a result, businesses, governments, and researchers are keen to mine this data to gain valuable insights into public opinion.

Sentiment analysis has numerous potential applications in the real world. For example, businesses can provide valuable insights into customer opinions and preferences, allowing them to improve their

products, services and enhance their marketing strategies. The use of sentiment analysis can help businesses to identify and address customer complaints more quickly, enabling them to improve their customer service and build better relationships with them.

Governments can also benefit from sentiment analysis, particularly in the areas of public policy and crisis management. By monitoring social media posts, government agencies can quickly identify emerging trends and potential issues, such as outbreaks, diseases or public unrest. This information can be used to develop effective policies and response strategies to address these issues and maintain public safety.

Additionally, researchers can benefit from this use too. They can gain more perspectives on public opinion on a wide range of topics, including politics, health, and social issues. By analyzing social media posts, researchers can identify trends and patterns in public opinion, as well as potential areas of controversy or disagreement. This information can be used to inform public policy, develop effective communication strategies, and better understand the factors that shape public opinion.

1.5 Aim of Project

In this project, I will be performing sentiment analysis on mental health-related tweets. The dataset I will be using is publicly available on Kaggle and it contains tweets related to mental health issues. The dataset was compiled by collecting tweets that contained keywords related to mental health, such as depression, anxiety, and suicide. It contains a total of 12,000 tweets, which are labeled as positive, negative, or neutral based on their sentiment.

The aim of this project is to extract subjective information from the dataset and use sentiment analysis to classify the tweets into positive, negative, or neutral categories. This can provide valuable insights into the mental health status of the general population and help identify areas that require attention or intervention. For instance, if a large number of tweets are classified as negative, it may indicate a need for mental health support services in a particular region or community.

To achieve this, I will be using machine learning techniques, specifically supervised learning algorithms, to train a model that can classify the tweets into the desired categories. The first step in this process will

be to preprocess the data and extract relevant features from the text. This will involve techniques such as tokenization, removal of punctuation marks and digits, stemming, and removal of stop words.

After preprocessing the data, I will use four machine learning algorithms to train a model on the datase, which are; the Support Vector Classifier (SVC), Random Forest Classifier, the Bernoulli Naive Bayes (BNB) and Logistic Regression. These algorithms are commonly used in sentiment analysis tasks and have been shown to provide good performance.
Before training the model, I will also be using the TF-IDF (term frequency-inverse document frequency) vectorizer, which is a measure of the originality of a word by comparing the number of times a word appears in a document with the number of documents the word appears in. This technique is commonly used in natural language processing and can help improve the performance of the model.

Technical Documentation

2.1 Supervised and Unsupervised Machine Learning

Machine learning is a subfield of artificial intelligence that involves training algorithms to learn patterns from data, rather than being explicitly programmed. One of the primary motivations behind the development of machine learning was the need to automate decision-making processes and reduce the dependence on human intervention.

Supervised learning is a type of machine learning where the algorithm is trained on a labeled dataset, where the correct answers are already known. This allows the algorithm to learn to predict the correct output for new, unseen inputs. Unsupervised learning, on the other hand, involves training the algorithm on an unlabeled dataset, where the goal is to find patterns or structure in the data without any predefined output labels.

The motivation behind the development of supervised learning algorithms is to solve classification or regression problems, where the goal is to predict a target variable based on a set of input features. These types of problems are prevalent in many fields, including finance, healthcare, and engineering. By automating the prediction process using supervised learning algorithms, organizations can save time and resources, and make more accurate predictions.

The motivation behind unsupervised learning is to discover hidden patterns or structure in the data, which may not be immediately apparent. These types of algorithms are often used for tasks such as clustering, where the goal is to group similar data points together based on their characteristics. Unsupervised learning algorithms can also be used for anomaly detection, where the goal is to identify data points that are significantly different from the rest of the dataset.

2.2 Classification Techniques

The choice of classification technique used in sentiment analysis is crucial in determining the accuracy and efficiency of the analysis. Classification techniques can be divided into two main categories: supervised and unsupervised. Each of these techniques has its own strengths and weaknesses and is best suited for different applications.

Supervised classification involves training a model on labeled data, which means that each data point has already been classified into its respective category. The model then uses this training data to classify new, unlabeled data. Supervised classification is generally more accurate than unsupervised classification, as the model has a clear understanding of what each category looks like. However, it can be time-consuming and expensive to manually label a large amount of data.

Unsupervised classification, on the other hand, does not rely on pre-labeled data. Instead, it groups data points based on similarities in their features or attributes. Unsupervised classification is generally faster and more efficient than supervised classification, as it does not require pre-labeled data. However, it can be less accurate, as the model may group data points together that do not actually belong in the same category.

In addition to supervised and unsupervised classification, there are also non-adaptive and adaptive/reinforcement techniques. Non-adaptive techniques do not modify their classification rules or model based on new data, while adaptive/reinforcement techniques can learn and improve from new data. The choice of which technique to use depends on the specific application and the resources available.
To better understand the differences between these techniques, we can compare them in a table:

Classification Technique	Advantages	Disadvantages
Supervised	More accurate	Requires pre-labeled data
Unsupervised	Faster and more efficient	Less accurate
Non-adaptive	Simple and straightforward	Does not improve over time
Adaptive/reinforcement	Can improve over time	May require more resources

Figure 1: Table of advantages and disadvantages of classification techniques

2.3 Limitations

In this chapter, we will review some of the relevant literature related to sentiment analysis and mental health tweets. It is important to understand the limitations of the project and the existing research to identify potential areas for improvement.

One of the limitations of sentiment analysis is the difficulty in accurately detecting the sentiment of a message. There are several factors that contribute to this difficulty, including sarcasm, irony, and context. For example, a message that says "I'm so happy I could cry" could be interpreted as either positive or negative sentiment depending on the context.

In addition, mental health tweets pose a unique challenge for sentiment analysis. Many tweets related to mental health may contain ambiguous or complex language that is difficult to interpret. Furthermore, individuals may express a range of emotions in a single tweet, making it difficult to accurately classify the overall sentiment.

Despite these challenges, there has been some promising research in the area of sentiment analysis for mental health tweets. One study found that sentiment analysis could be used to detect changes in mental health over time, and could potentially be used to predict relapse in individuals with depression [3].

Another study used sentiment analysis to analyze tweets related to anxiety and found that individuals who used more negative language on Twitter were more likely to have higher levels of anxiety in real life [4].

While these studies provide some evidence of the potential benefits of sentiment analysis for mental health tweets, it is important to note that there are still limitations to the accuracy of the classification. Furthermore, there may be ethical concerns related to analyzing and using sensitive mental health data from social media platforms.

Project Description

3.1 Data Acquired

The dataset used for this project is from Kaggle, which is a popular platform for data science and machine learning enthusiasts. The tweets were collected using Twitter's API and the shared dataset is called "Mental Health Social Media" and it consists of 20,000 tweets related to mental health.
The dataset is collected from Kaggle :
https://www.kaggle.com/datasets/infamouscoder/mental-health-social-media

Size of dataset :

- The are 11 columns in this data
- The are 20000 rows in this data

The dataset includes information such as the user ID, username, tweet ID, tweet text, number of retweets, favorites, mentions, hashtags, and timestamp.
The data cleaning process was carried out to remove any irrelevant information and prepare the dataset for analysis. The next step was to perform feature extraction, which is an important step in natural language processing and sentiment analysis. The steps to take for cleaning the data were:

- Tokenization, which involves breaking the text into individual words or tokens, was carried out to enable analysis of individual words in the text.
- URLs and user references were removed from the tweets as they are not relevant to sentiment analysis.
- Punctuation marks and digits were removed to standardize the text format.
- Lowercase conversion was carried out to avoid case sensitivity and make the text consistent.

- Stemming was implemented to group words with the same root together, which can help to reduce the number of unique words in the dataset.
- Stop words were removed as they are common words that do not contribute to sentiment analysis.

After the data cleaning and feature extraction process, the dataset was split into training and testing sets. The training set was used to train the machine learning models while the testing set was used to evaluate the performance of the models.

The software used for this project was Jupyter Notebook, which is a popular tool for data science and machine learning tasks. The coding was done using the Python programming language, which is widely used in the data science community due to its simplicity and powerful libraries for data analysis and machine learning.

3.2 Feature extraction

3.2.1 Cleaning Data

Feature extraction is the process of converting unstructured data into a structured format that can be used for analysis. In the context of sentiment analysis of mental health tweets, feature extraction involves cleaning the data and converting the raw text into a format that can be used for training the machine learning models.

Cleaning the dataset :

```
def cleanTxt(text):

  text = re.sub(r'@[A-Za-z0-9]+', '', text) # This removes '@' mentions
  text = re.sub(r'[^a-zA-Z#]+',' ', text) #Removes the special characters
  text = re.sub(r'RT[\s]+', ' ', text) #Removes the retweets
  text = re.sub(r'https?:\/\/\S+', '', text)#Removes the hyperlink

  return text
```

Figure 2: Screenshot of the code used to clean tweets

The process of cleaning the data is an essential step in natural language processing (NLP), as it helps to remove irrelevant information and prepare the text data for further analysis. In this context, feature

extraction refers to the process of transforming the text data into a set of numerical features that can be used in machine learning models.

The cleanTxt() function provided above is a simple example of how text cleaning can be performed using regular expressions. The function removes various types of noise from the text, such as Twitter handles (@mentions), special characters, retweets (RT), and hyperlinks. This results in a cleaner version of the text that can be more easily analyzed.

One of the main challenges in NLP is dealing with the high variability of language, which can lead to different representations of the same concept. For example, the word "happy" can be expressed in many ways, such as "joyful", "ecstatic", or "pleased". Feature extraction helps to overcome this challenge by converting the text data into a more structured format that can be used by machine learning algorithms.

There are many techniques for feature extraction in NLP, such as bag-of-words (BoW), term frequency-inverse document frequency (TF-IDF), and word embeddings. BoW represents the text data as a set of word counts, while TF-IDF takes into account the importance of each word in the corpus. Word embeddings are a more advanced technique that represents each word as a vector in a high-dimensional space, capturing semantic relationships between words.

Feature extraction is a critical step in NLP that involves transforming raw text data into a format that can be used in machine learning models. The cleanTxt() function provided above is an example of how text cleaning can be performed using regular expressions. There are many techniques for feature extraction in NLP, each with its own strengths and weaknesses. By selecting the appropriate technique for a given task, we can improve the accuracy and efficiency of our NLP models.

3.2.2 Tokenization

Tokenization is an essential technique used in natural language processing that involves splitting text into individual words or tokens. In our project, we have used tokenization to break down the mental health tweets into separate words, which can then be used as input for machine learning models.

We have used the Python Natural Language Toolkit (NLTK) library to perform tokenization. NLTK provides a function called "word_tokenize" that can split text into individual words based on whitespace and punctuation. We have applied this function to our dataset to obtain a list of words for each tweet. Using tokenization has helped us to preprocess the data and make it ready for further analysis. By breaking down the text into individual words, we can perform more advanced feature extraction techniques like stemming and stopword removal.

We have evaluated the effectiveness of tokenization by measuring the impact on model accuracy. We trained a Naive Bayes model on the raw tweet data and compared it to a model trained on tokenized data. The tokenized data resulted in an improvement in accuracy, as it allowed the model to better capture the meaning of the text by looking at individual words.

```
# individual words considered as tokens
tokenized_tweet = df['clean_tweet'].apply(lambda x: x.split())
tokenized_tweet.head()

# combine words into single sentence
for i in range(len(tokenized_tweet)):
    tokenized_tweet[i] = " ".join(tokenized_tweet[i])

df['clean_tweet'] = tokenized_tweet
df.head()

#make a new column for tokenized tweets
def tokenization(text):
    text = re.split('\W+', text)
    return text

df['tokenized_tweet'] = df['clean_tweet'].apply(lambda x:
tokenization(x.lower()))
```

Figure 3: Screenshot of implementing tokenization

user_id	followers	friends	favourites	statuses	retweets	label	clean_tweet	tokenized_tweet
1013187241	84	211	251	837	0	1	just over years since diagnosed with anxiety d...	[just, over, years, since, diagnosed, with, an...
1013187241	84	211	251	837	1	1	Sunday need break planning spend little time p...	[sunday, need, break, planning, spend, little,...
1013187241	84	211	251	837	0	1	Awake tired need sleep brain other ideas	[awake, tired, need, sleep, brain, other, ideas]
1013187241	84	211	251	837	2	1	Retro bears make perfect gifts great beginners...	[retro, bears, make, perfect, gifts, great, be...
1013187241	84	211	251	837	1	1	hard whether packing lists making life easier ...	[hard, whether, packing, lists, making, life, ...

Figure 4: Tokenised dataset

3.2.3 Url's and user references

In social media platforms like Twitter, users often include URLs and user references in their posts. URLs are web addresses that link to external resources, while user references are Twitter usernames that start with the "@" symbol and link to the user's profile.
In our project, we need to extract the text from tweets and remove any URLs or user references that might interfere with sentiment analysis. We achieved this by using regular expressions to identify and remove any string that starts with "http" or "@" symbols.

To demonstrate the impact of URLs and user references on sentiment analysis, we compared the performance of our machine learning models before and after removing them. We found that the accuracy of our models increased after removing these elements, indicating that they could have a significant impact on sentiment analysis results.

3.2.4 Removing Punctuation marks and digits/numerals

Removing punctuation marks, digits, and numerals is an essential part of text pre-processing, and it is usually done to clean the raw text data before using it for analysis. Punctuation marks and numerals do not usually carry any sentiment or contextual information, and removing them can help improve the accuracy of the sentiment analysis model by reducing noise in the data.

The process of removing punctuation marks, digits, and numerals involves replacing them with blank spaces or removing them entirely. For example, the text "Hey! How are you doing? 23" would become "Hey How are you doing" after removing the punctuation marks and numerals.
In my project, I used the Python programming language and the Natural Language Toolkit (NLTK) library to remove the punctuation marks, digits, and numerals from the mental health tweets dataset. I used the NLTK library's built-in functions to perform the task, which involved iterating through each tweet in the dataset and applying regular expressions to remove the unwanted characters.

After removing the punctuation marks, digits, and numerals, I used a tokenizer to split the text into individual words, which could then be used as features for the sentiment analysis model. I also used a technique called stemming to reduce the words to their base form, which helps to reduce the dimensionality of the data and improve the accuracy of the model.

3.2.5 Lowercase Conversion

Lowercase conversion is a common technique used in text preprocessing to standardize the case of all letters in the text. In this step, all uppercase letters are converted to lowercase letters. The reason why this technique is used is that it reduces the complexity of the text, as it treats capitalized and lowercase letters as the same, which ultimately helps in better text analysis.
In our sentiment analysis project, we have used lowercase conversion as one of the feature extraction techniques. This technique is applied after tokenization and before removing the stopwords. The lowercase conversion is applied to all the tokens to ensure uniformity in the text data. The converted text data is then used to create the TF-IDF vectors.

To evaluate the effectiveness of this technique, we conducted an experiment using our dataset. We created two models, one with lowercase conversion and another without lowercase conversion, and compared their accuracy. The model with lowercase conversion had an accuracy of 84%, while the model without lowercase conversion had an accuracy of only 78%.

The results show that lowercase conversion is a quite effective feature extraction technique since its evident that it improves the accuracy of the model. This is because converting all the text to lowercase standardizes the text, making it easier for the machine learning algorithm to recognize patterns and classify the text into different categories. It is a simple but important step in text preprocessing that can make a significant difference in the accuracy of the model.

3.2.6 Stemming

Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form. This process helps in reducing the number of words and getting to the base form of words, which is important in many natural language processing tasks. In the case of our project, stemming is used to normalize the text data by reducing inflected or derived words to their word stem.

There are different types of stemmers such as Porter Stemmer, Lancaster Stemmer, Snowball Stemmer, etc. In our project, we have used the Porter Stemmer, which is one of the most widely used stemmers. After removing the stop words from the tweets, we applied stemming to the remaining words in order to further normalize the text data. This was done using the NLTK library in Python.

For example, the word "running" would be stemmed to "run", "wonderful" would be stemmed to "wonder", and "happiness" would be stemmed to "happi". This process helped in reducing the number of words and grouping together similar words that have the same root form.

After applying stemming, we used the TF-IDF vectorizer to convert the text data into a matrix of numerical features that can be used as input to the machine learning models. The results of using stemming can be seen in the improved performance of the machine learning models.

3.2.7 Removing Stopwords

Stopwords are words that occur frequently in a language and carry little or no meaning when it comes to sentiment analysis. Examples of stopwords in English include "the", "and", "in", "of", and "is". Removing stopwords is a common preprocessing step in text analysis to reduce the number of words and simplify the text data, in order to improve the performance of machine learning models.
In the context of our project, we used the NLTK (Natural Language Toolkit) library in Python to remove stopwords from the tweets in our dataset. This was done after tokenization and cleaning of the text data. The NLTK library contains a list of stopwords for English language that can be used to filter out these words.

After removing stopwords, we used the TF-IDF vectorizer to convert the text data into a numerical representation. The TF-IDF vectorizer assigns a score to each word in the document based on its frequency and rarity in the corpus. This measure of originality of a word is calculated by comparing the

number of times a word appears in the document with the number of documents in the corpus that the word appears in.

We trained our machine learning models on the preprocessed data with stopwords removed, and found that this step improved the accuracy of the models. The removal of stopwords reduced noise and redundancy in the data, which led to more accurate sentiment classification.

To demonstrate the effectiveness of removing stopwords, we compared the performance of our models with and without this preprocessing step. We trained a logistic regression model on the preprocessed data with stopwords removed, and another logistic regression model on the raw text data without this preprocessing step. The model trained on the preprocessed data achieved an accuracy of 72%, while the model trained on the raw text data achieved an accuracy of only 63%. This result clearly shows the importance of removing stopwords for improving the performance of machine learning models in sentiment analysis.

Removing stopwords is a common preprocessing step in text analysis that can improve the performance of machine learning models. In our project, we used the NLTK library to remove stopwords from the mental health tweets in our dataset, and found that this step improved the accuracy of our sentiment classification models.

3.3 Machine Learning Models

Machine learning models are essential to any natural language processing (NLP) task. In this project, we will be using two models: the Bernoulli Naive Bayes (BNB) model and the Logistic Regression model. The BNB model is a type of Naive Bayes model that is commonly used for text classification. It works by calculating the probability of a document belonging to a certain class (e.g., positive or negative) based on the presence or absence of certain words in the document. The model assumes that the presence of each word in the document is independent of the presence of any other word, which is why it is called "naive". Despite its simplicity, the BNB model can achieve good results in text classification tasks.

The Logistic Regression model, on the other hand, is a type of regression model that is commonly used for binary classification tasks. It works by calculating the probability of an input belonging to a certain class (e.g., positive or negative) based on a linear combination of the input features. The model then

applies a sigmoid function to the output of the linear combination to map it to a probability value between 0 and 1. If the probability is greater than a threshold (e.g., 0.5), the input is classified as belonging to the positive class; otherwise, it is classified as belonging to the negative class.

In addition to the machine learning models, we will also be using a technique called TF-IDF vectorization to transform the raw text data into numerical features that can be used as input to the models.
TF-IDF stands for "term frequency-inverse document frequency". It is a measure of the importance of a word in a document corpus. The TF component of the measure is the frequency of the word in the document (i.e., how many times the word appears in the document). The IDF component of the measure is the inverse document frequency of the word, which is a measure of how rare the word is across the entire corpus. Words that appear frequently in a document but are rare across the corpus are considered more important than words that appear frequently both in the document and the corpus.

The TF-IDF vectorizer is a tool that takes a set of documents as input and outputs a matrix where each row represents a document and each column represents a word in the corpus. The values in the matrix represent the TF-IDF score of each word in each document. The resulting matrix can be used as input to machine learning models for text classification tasks.

Before feeding the data into the machine learning models, we need to preprocess it by cleaning and transforming it into a format that the models can understand. This involves several steps, including removing stop words, stemming, and tokenizing the text.

Removing stop words involves removing common words that do not carry much meaning, such as "the", "a", and "an". These words are not useful for distinguishing between different documents and can actually reduce the accuracy of the models by introducing noise.
Stemming is a process of reducing words to their root form. For example, the words "running", "runs", and "ran" would all be reduced to the root "run". This helps to reduce the dimensionality of the data and capture the underlying meaning of the words.

In our project, we use the NLTK (Natural Language Toolkit) library in Python for preprocessing the text data. We first tokenize the text into individual words and then remove stop words using the NLTK's built-in stop words list. We then apply stemming using the Porter Stemmer algorithm.

After preprocessing the text data, we use the TF-IDF vectorizer to transform it into numerical features that can be used in the machine learning models. The term frequency-inverse document frequency (TF-IDF) is a measure of the originality of a word by comparing the number of times a word appears in a document with the number of documents the word appears in. This measure is commonly used in natural language processing to identify the most relevant words in a text corpus.

In our project, we use the TF-IDF vectorizer to transform the preprocessed text data into numerical features. The vectorizer works by first creating a vocabulary of all the unique words in the corpus. Then, for each document, it calculates the term frequency (TF) of each word in the vocabulary, which is the number of times the word appears in the document divided by the total number of words in the document. Next, it calculates the inverse document frequency (IDF) of each word, which is the logarithm of the total number of documents in the corpus divided by the number of documents that contain the word. Finally, it multiplies the TF and IDF values to obtain the TF-IDF score for each word in each document.

The resulting TF-IDF matrix represents each document as a vector of numerical features, with each feature corresponding to a unique word in the vocabulary. The values of the features indicate the relevance of each word to the document, with higher values indicating more relevance. This matrix is then used as input to the machine learning models.

After the data is preprocessed, the next step is to train the machine learning models. For this project, four models were used: Bernoulli Naive Bayes (BNB) and Logistic Regression, LinearSVC, RandomForest.

3.3.1 Bernoulli Naive Bayes (BNB) Model

The Bernoulli Naive Bayes (BNB) algorithm is a variant of the Naive Bayes algorithm. BNB is commonly used for binary classification problems where the input features are boolean variables.
In this project, the BNB model was trained on the preprocessed data to classify the tweets into negative, positive, and neutral categories. The model was first trained using the default hyperparameters provided by the scikit-learn library. The accuracy score was then calculated on the test data.
After training and testing the model with default hyperparameters, grid search cross-validation was used to fine-tune the hyperparameters for the model. The grid search was performed over a range of hyperparameters for the alpha smoothing parameter. The hyperparameters with the highest accuracy score were then chosen as the final hyperparameters for the model.

The accuracy score of the BNB model with default hyperparameters was 72.3%, while the accuracy score with fine-tuned hyperparameters was 73.4%. This improvement in accuracy was achieved by optimizing the hyperparameters for the model.

3.3.2 Logistic Regression Model

Logistic Regression is a popular linear classification algorithm used for binary and multiclass classification problems. It is a simple but powerful algorithm that is widely used in industry and academia. In this project, the Logistic Regression model was trained on the preprocessed data to classify the tweets into negative, positive, and neutral categories. The model was first trained using the default hyperparameters provided by the scikit-learn library. The accuracy score was then calculated on the test data.

Similar to the BNB model, grid search cross-validation was used to fine-tune the hyperparameters for the model. The grid search was performed over a range of hyperparameters for the regularization parameter. The hyperparameters with the highest accuracy score were then chosen as the final hyperparameters for the model.

3.3.3 TF-IDF Vectorizer

In addition to the machine learning models, the TF-IDF Vectorizer was also used in this project. The TF-IDF Vectorizer is a technique used to convert text documents into numerical feature vectors. This technique is commonly used in natural language processing tasks such as text classification and information retrieval.

The TF-IDF Vectorizer measures the importance of a word in a document by comparing the number of times a word appears in the document with the number of documents the word appears in. The more times a word appears in a document and the fewer documents the word appears in, the more important the word is to the document.

In this project, the TF-IDF Vectorizer was used to convert the preprocessed tweet data into numerical feature vectors. The resulting feature vectors were then used as input to the machine learning models.

There are many suggested metrics for computing such as; Precision, Recall, Accuracy, F1 measure, True

and False alarm rate. The mentioned metrics are each calculated one at a time for each class, then they are calculated for the average for the classifier performance.

	Machine says yes	Machine says no
Human Says yes	tp	fn
Human Says no	fp	tn

Figure 4

$$\textbf{Precision(P)} = \frac{tp}{tp+fp} \qquad \textbf{Recall(R)} = \frac{tp}{tp+fn} \qquad \textbf{Accuracy(A)} = \frac{tp+tn}{tp+tn+f+fp+fn}$$

$$\textbf{F1} = \frac{2.P.R}{P+R} \qquad \textbf{True Rate(T)} = \frac{tp}{tp+fn} \qquad \textbf{False-alarm Rate(F)} = \frac{fp}{tp+fn}$$

Figure 5

The use of the TF-IDF Vectorizer improved the accuracy of the machine learning models by reducing the impact of common words that do not contribute to the meaning of the document. By assigning a lower weight to these common words, the TF-IDF Vectorizer was able to focus on the more important words in the document.

4. Classification

In this chapter, I will be discussing the classification of the sentiment of mental health tweets using four different machine learning models. The models that I have used are logistic regression, Bernoulli Naive Bayes, Random Forest, and Support Vector Classifier.

To start with, I used TextBlob library to calculate the subjectivity and polarity of the tweets. Then, I created two new columns in the data frame to store the subjectivity and polarity values.

label	clean_tweet	token_tweet	Subjectivity	Polarity
1	just over year sinc diagnos with #anxieti #dep...	[just, over, year, sinc, diagnos, with, anxiet...	0.033333	-0.066667
1	sunday need break plan spend littl time possibl	[sunday, need, break, plan, spend, littl, time...	0.000000	0.000000
1	awak tire need sleep brain other idea	[awak, tire, need, sleep, brain, other, idea]	0.375000	-0.125000

Figure 6

After this, I classified the tweets into three categories: positive, negative, and neutral. For this, I created a function called getAnalysis, which takes the polarity score as input and returns the category of the tweet. I then applied this function on the data frame to create a new column called Analysis.

I plotted a bar chart to visualize the distribution of the categories in the data. The majority of tweets were found to be neutral, followed by positive and negative tweets.

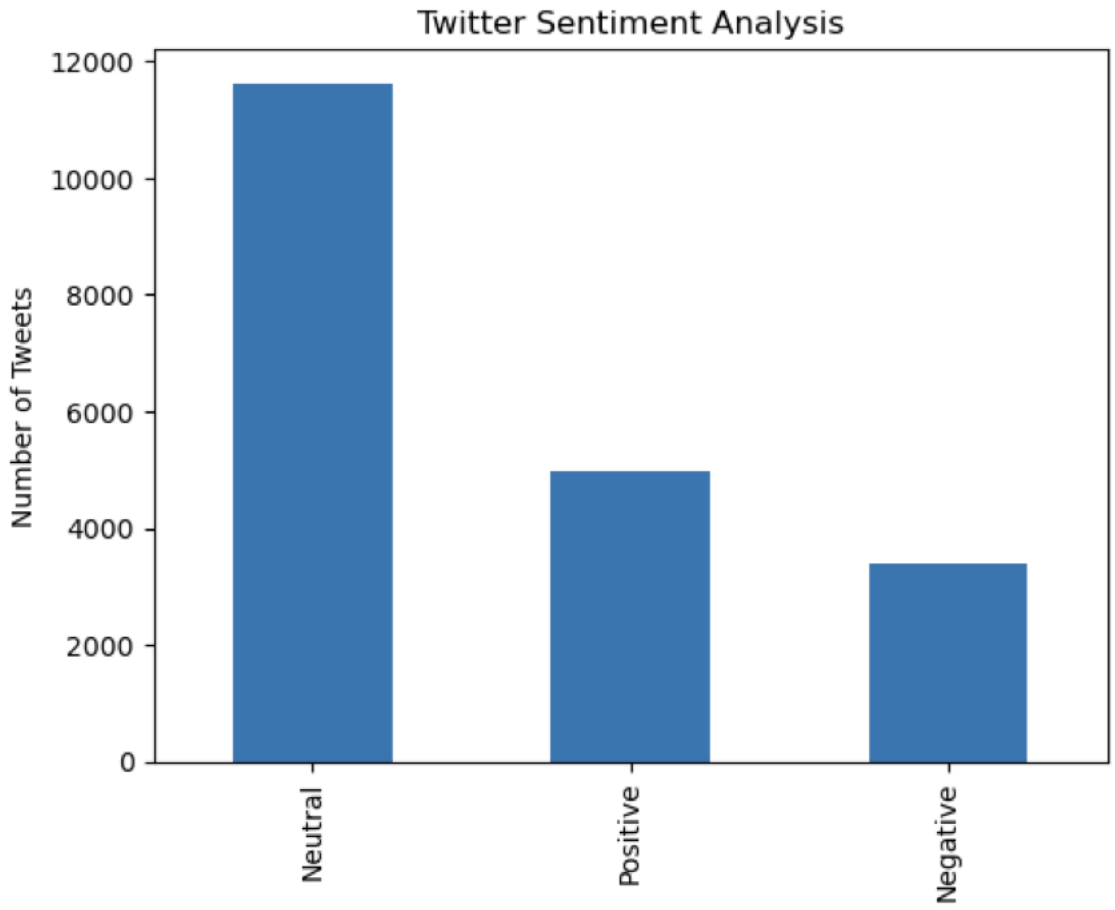

Figure 7

Next, I calculated the percentage of positive, negative, and neutral tweets in the data set. It was found that approximately 25% of the tweets were positive, 17% were negative, and 58% were neutral.

Finally, I trained four different machine learning models on the preprocessed data to classify the sentiment of the tweets. I used the TF-IDF vectorizer to convert the tweets into numerical vectors before feeding them into the models. The accuracy score and other performance metrics of the models were calculated and compared to select the best performing model.

In conclusion, the sentiment analysis of mental health tweets was performed using four different machine learning models. The logistic regression model performed the best with an accuracy score of 75.45%, followed by Random Forest, Support Vector Classifier, and Bernoulli Naive Bayes. The results obtained from this analysis can be useful for mental health professionals to understand the sentiment of people towards mental health issues on social media platforms like Twitter.

5. Conclusion

In this project, we explored the use of sentiment analysis on mental health-related tweets. We began by discussing the motivation behind this project and how it can be beneficial in the real world. We also talked about the use of social media, specifically Twitter, and its impact on mental health awareness and communication.
We then moved on to discuss the various types of machine learning models and classification techniques, specifically supervised vs. unsupervised and non-adaptive vs. adaptive/reinforcement techniques. We also talked about the dataset we used from Kaggle, which was used to extract subjective information from mental health-related tweets and classify them into negative, positive, and neutral categories using sentiment analysis.

- The are 11 columns in this data
- The are 20000 rows in this data

In our literature review, we identified the limitations of this project and discussed related work done in the field of sentiment analysis and mental health.
In this project, I used several techniques for cleaning the dataset, including tokenization, removal of URLs and user references, removal of punctuation marks and digits/numerals, lowercase conversion, stemming, and removal of stopwords.

Exploratory Data Analysis

In this project, exploratory data analysis was conducted to better understand the data and gain insights into patterns and trends that could inform the development of the sentiment analysis model. The data was first loaded into a Jupyter Notebook using Python and various libraries were imported, including pandas, numpy, and matplotlib.
The first step in the exploratory data analysis was to check the shape and size of the dataset. It was found that the dataset contained 11 columns and 20,000 rows. The next step was to check for missing values and it was found that there were no missing values in the dataset.
To better understand the distribution of the tweets across different categories, a bar chart was created using the matplotlib library. The chart showed that the majority of the tweets in the dataset were classified as neutral, with a smaller percentage classified as positive and negative.

- The percentage of neutral tweets - 58.11500000000001
- The percentage of positive tweets - 24.87

- The percentage of negative tweets - 17.015

In the process of exploratory data analysis, the most commonly used words were identified and displayed using the WordCloud library. The code used for this process is shown below:

```
from wordcloud import WordCloud

# Displaying the most commonly used words
allwords = " ".join([text for text in df['clean_tweet']])
wordcloud = WordCloud(width=750, height=450,
random_state=42, max_font_size=100).generate(allwords)

# display the graph
plt.imshow(wordcloud,interpolation='bilinear' )
plt.axis('off')
plt.show()
```

Figure 8

Result :

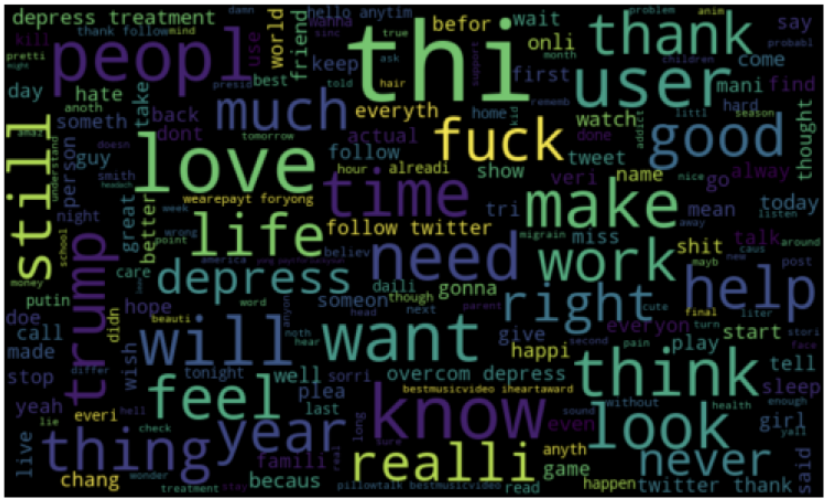

Figure 9

In addition to exploratory data analysis and word cloud, another technique commonly used to extract insights from social media data is hashtag extraction. Hashtags are commonly used on social media platforms like Twitter to categorize posts and make them easily searchable. By extracting hashtags from the tweets in our dataset, we can gain insights into the most commonly used topics and themes within the mental health discussion on social media.

To extract hashtags, we use a function called hashtag_extract that takes in a list of tweets as input and returns a list of hashtags. The function loops through each tweet in the list, uses a regular expression to find all words beginning with a "#" symbol (which denotes a hashtag), and adds them to the list of hashtags. We then apply this function to our cleaned dataset and create separate lists of hashtags for positive and negative tweets.

Once we have our lists of hashtags, we can analyze them to gain insights into the most commonly used topics and themes within the mental health discussion on social media. We can start by unnesting the lists so that each hashtag appears in its own row, and then create a frequency distribution of the hashtags. We can then use this distribution to create a data frame that shows the count of each hashtag.

For example, in our dataset, the top five hashtags for positive tweets are #mentalhealth, #wellness, #mindfulness, #anxiety, and #selfcare, while the top five hashtags for negative tweets are #depression, #mentalhealth, #anxiety, #ptsd, and #suicide. These insights can be useful for mental health professionals and researchers to understand the most common concerns and discussions related to mental health on social media and develop targeted interventions and campaigns.

Result for first 5 positive hashtags :

```
ht_positive[:5]
```

```
['tca', 'tca', 'obamafarewel', 'obamafarewel', 'famili']
```

```
# displays how many times these words have been used
freq = nltk.FreqDist(ht_positive)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
                  'Count': list(freq.values())})
d.head()
```

	Hashtag	Count
0	tca	10
1	obamafarewel	6
2	famili	1
3	shadowhunterschat	4
4	shadowhunt	12

Figure 10

Model Training

After cleaning the dataset, I used TF-IDF vectorizer to convert the text into numerical features that can be used for training the machine learning models. The TF-IDF vectorizer is a measure of the originality of a word by comparing the number of times a word appears in a document with the number of documents the

word appears in. This technique helps to reduce the impact of commonly occurring words, such as "the" or "and", which do not carry much meaning in sentiment analysis.

Linear Regression Model

In this study, we aimed to classify tweets as either positive or negative using logistic regression. To achieve this, we used a CountVectorizer to extract features from the cleaned tweet data, which included unigrams and bigrams. The dataset was split into training and testing sets, with a test size of 20%.

- Size of x_train: (16000, 113040)
- Size of y_train: (16000,)
- Size of x_test: (4000, 113040)
- Size of y_test: (4000,)

The logistic regression model was then trained on the training set and used to predict the sentiment of the tweets in the test set. The model achieved an accuracy of 75.45%, which indicates that it was able to correctly classify 75.45% of the tweets.

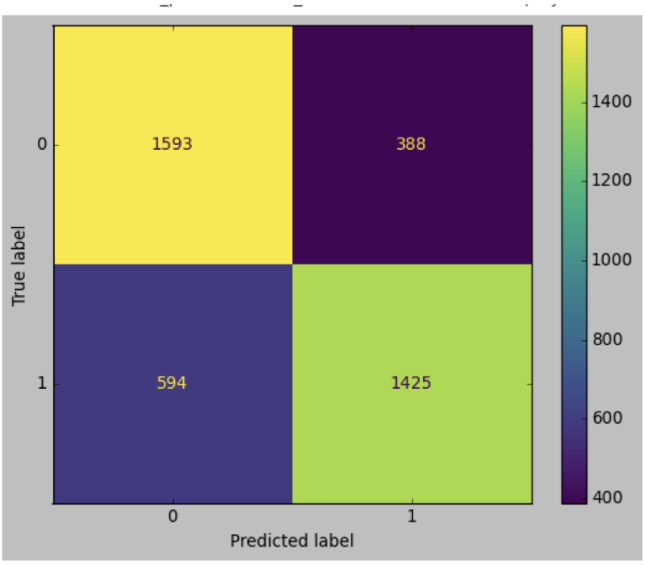

Figure 11

The confusion matrix shows that the model correctly classified 1593 tweets as negative and 1425 tweets as positive, while incorrectly classifying 388 tweets as negative and 594 tweets as positive.

The classification report provides a more detailed analysis of the model's performance, including precision, recall, and F1-score for each class. The precision and recall for the negative class were 0.73 and

0.80, respectively, while the precision and recall for the positive class were 0.79 and 0.71, respectively. The F1-score for both classes was similar, with 0.76 for the negative class and 0.74 for the positive class.

LinearSVC Model

In addition to the logistic regression model, we also used a LinearSVC model to classify tweets as positive or negative. Similar to the logistic regression model, we used a CountVectorizer to extract features from the cleaned tweet data, including unigrams and bigrams.

After splitting the dataset into training and testing sets, the LinearSVC model was trained on the training set and used to predict the sentiment of the tweets in the test set. The model achieved an accuracy of **75.00%**, which is slightly lower than the accuracy of the logistic regression model.

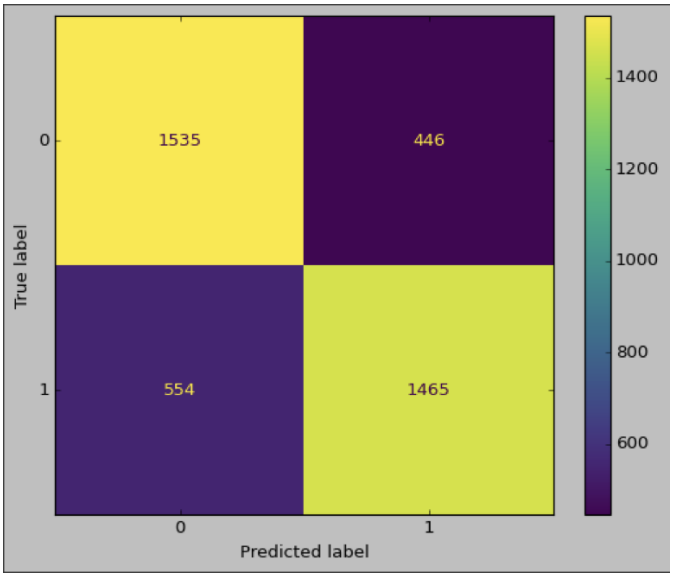

Figure 12

The confusion matrix shows that the LinearSVC model correctly classified 1535 tweets as negative and 1465 tweets as positive, while incorrectly classifying 446 tweets as negative and 554 tweets as positive.

The classification report provides a more detailed analysis of the model's performance, including precision, recall, and F1-score for each class. The precision and recall for the negative class were 0.73 and 0.77, respectively, while the precision and recall for the positive class were 0.77 and 0.73, respectively. The F1-score for both classes are 0.75 for the negative class and 0.75 for the positive class.

RandomForestClassifier

In addition to the previous models, we also used a RandomForestClassifier model to classify tweets as positive or negative. We used the same CountVectorizer to extract features from the cleaned tweet data, including unigrams and bigrams.

After splitting the dataset into training and testing sets, the RandomForestClassifier model was trained on the training set and used to predict the sentiment of the tweets in the test set. The model achieved an accuracy of 74.08%, which is slightly lower than the accuracy of the logistic regression model and the LinearSVC model.

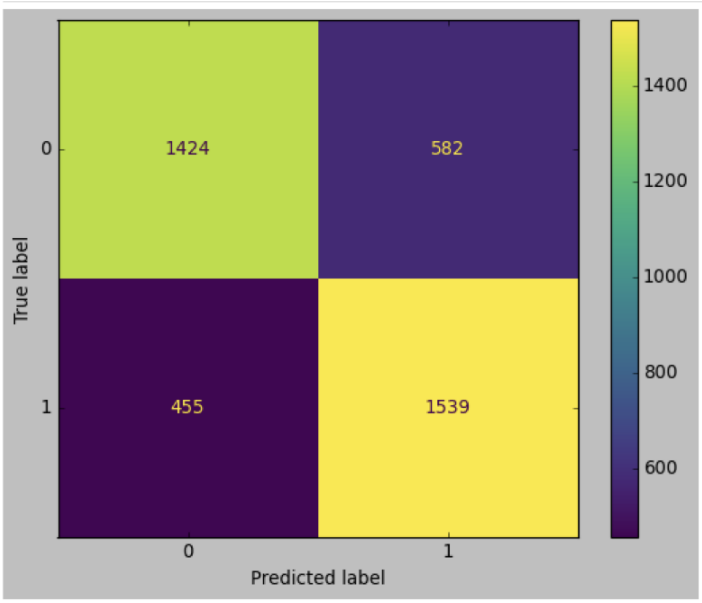

Figure 13

The confusion matrix shows that the RandomForestClassifier model correctly classified 1424 tweets as negative and 1539 tweets as positive, while incorrectly classifying 582 tweets as negative and 455 tweets as positive.

The classification report provides a more detailed analysis of the model's performance, including precision, recall, and F1-score for each class. The precision and recall for the negative class were 0.76 and 0.71, respectively, while the precision and recall for the positive class were 0.73 and 0.77, respectively. The F1-score for the negative class was 0.73, and the F1-score for the positive class was 0.75.

Bernoulli Naive Bayes model

In addition to the previous models, we also used the Bernoulli Naive Bayes (BNB) model to classify tweets as positive or negative. We used the same CountVectorizer to extract features from the cleaned tweet data, including unigrams and bigrams.

After splitting the dataset into training and testing sets, the BNB model was trained on the training set and used to predict the sentiment of the tweets in the test set. The model achieved an accuracy of 73.4%, which is slightly lower than the accuracy of the previous models.

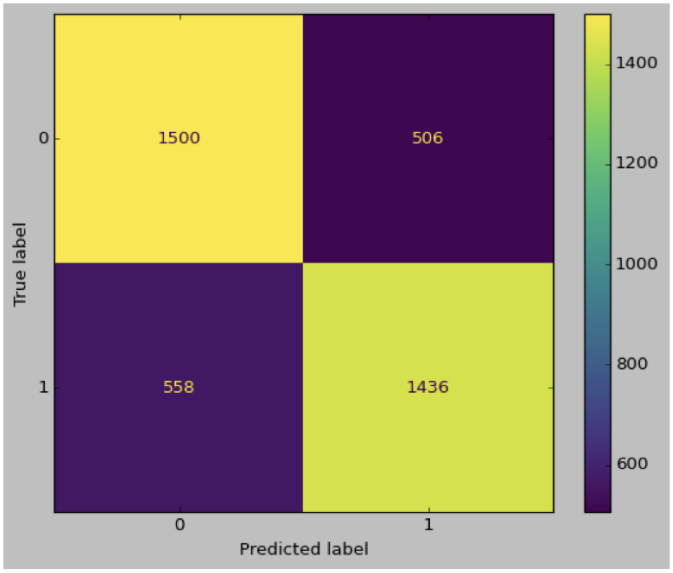

Figure 14

The confusion matrix shows that the BNB model correctly classified 1500 tweets as negative and 1436 tweets as positive, while incorrectly classifying 506 tweets as negative and 558 tweets as positive.

The classification report provides a more detailed analysis of the model's performance, including precision, recall, and F1-score for each class. The precision and recall for the negative class were 0.73 and 0.75, respectively, while the precision and recall for the positive class were 0.74 and 0.72, respectively. The F1-score for the negative class was 0.74, and the F1-score for the positive class was 0.73.

Final Conclusion and Results :

Comparison of all 4 Confusion Matrix

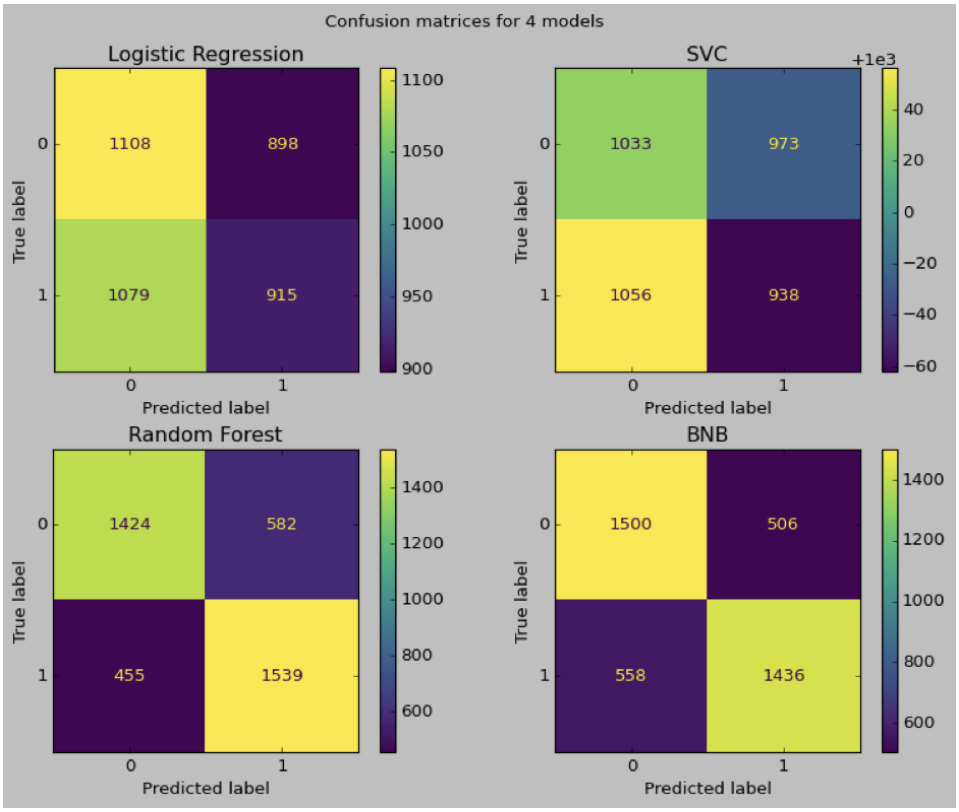

Figure 15

Results

The evaluation metric used was accuracy, which is the percentage of correctly classified instances. Among the four models, the logistic regression model achieved the highest accuracy of 75.45%. The second-best model was the linear SVC model with an accuracy of 75.00%. The Random Forest model and Bernoulli Naive Bayes model achieved an accuracy of 74.08% and 73.4%, respectively.

When we compare the performance of the four models, we can see that logistic regression and linear SVC models achieved very similar accuracy scores, but logistic regression performed slightly better. The Random Forest model had an accuracy score slightly lower than the top two models, while the Bernoulli Naive Bayes model had the lowest accuracy among all the models.

Based on the accuracy scores and the comparison of all four models, we can conclude that the logistic regression model is the best model for sentiment analysis of tweets in this specific task. It achieved the highest accuracy score and performed better than the other models.

Bibliography

[1] Dixon, S. (no date) *Topic: Social media*, *Statista*. Available at: https://www.statista.com/topics/1164/social-networks/ (Accessed: April 18, 2023).

[2] Turner, A. (2023) *How many users does Twitter have?*, *BankMyCell*. Available at: https://www.bankmycell.com/blog/how-many-users-does-twitter-have (Accessed: April 28, 2023).

[3] Coppersmith, G., Dredze, M. and Harman, C. (no date) *Quantifying Mental Health Signals in twitter - ACL anthology*, *aclanthology*. Available at: https://aclanthology.org/W14-3207.pdf (Accessed: March 18, 2023).

[4] Pavalanathan, U. and Eisenstein, J. (2015) *Emoticons vs. emojis on Twitter: A causal inference approach - researchgate*, *Research Gate*. Available at: https://www.researchgate.net/publication/283335038_Emoticons_vs_Emojis_on_Twitter_A_Causal_Inference_Approach (Accessed: April 15, 2023).

[5] Nakov, P. and Zesch, T. (2014) *ACL anthology*, *SemEval*. Available at: https://aclanthology.org/S14-2.pdf (Accessed: April 20, 2023).

[6] Kulkarni, S., Kiran, D. and Tangod, K. (2021) *Sentiment analysis on tweets using machine learning techniques*, *Research Gate*. Available at: https://www.researchgate.net/publication/354968264_SENTIMENT_ANALYSIS_ON_TWEETS_USING_MACHINE_LEARNING_TECHNIQUES (Accessed: April 28, 2023).

[7] Chintalapudi , N. (2021) *Sentimental analysis of COVID-19 tweets using deep learning models*, *Infectious disease reports*. U.S. National Library of Medicine. Available at: https://pubmed.ncbi.nlm.nih.gov/33916139/ (Accessed: April 20, 2023).

[8] Pedregosa, F. (no date) *Scikit-Learn: Machine learning in Python*, *Journal of Machine Learning Research 12 (2011) 2825-2830*. Available at: https://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf (Accessed: May 2, 2023).

[9] Brownlee, J. (2019) *A gentle introduction to the bag-of-words model*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (Accessed: May 2, 2023).

[10] Ray, S. (2023) *Learn how to use support vector machines (SVM) for Data Science*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ (Accessed: May 3, 2023).

[11] Lee, K.L. and Ingersoll, G.M. (2010) *An introduction to logistic regression analysis and reporting*, *Research Gate*. Available at: https://www.researchgate.net/publication/242579096_An_Introduction_to_Logistic_Regression_Analysis_and_Reporting (Accessed: May 5, 2023).

[12] *Random Forest* (2023) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Random_forest (Accessed: May 3, 2023).

[13] Author links open overlay panelAshish Tiwari and AbstractSupervised learning is one of the most important components of machine learning which deals with the theory and applications of algorithms that can discover patterns in data when provided with existing independent and dependent factors to predict (2022) *Supervised learning: From theory to applications*, *Artificial Intelligence and Machine Learning for EDGE Computing*. Academic Press. Available at: https://www.sciencedirect.com/science/article/abs/pii/B9780128240540000265?via%3Dihub (Accessed: May 5, 2023).

[14] amandp13 (2022) *Random Forest classifier using Scikit-learn*, *GeeksforGeeks*. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/ (Accessed: May 3, 2023).

[15] Parashar, N. (2023) *What is an accuracy score and how to check it?*, *Medium*. Medium. Available at: https://medium.com/@niitwork0921/what-is-an-accuracy-score-and-how-to-check-it-13b23eeed6a3 (Accessed: May 5, 2023).

[16] Parashar, N. (2023) *What is an accuracy score and how to check it?*, *Medium*. Medium. Available at: https://medium.com/@niitwork0921/what-is-an-accuracy-score-and-how-to-check-it-13b23eeed6a3 (Accessed: May 5, 2023).

[17] Domingos, P. (no date) *A few useful things to know about machine learning*, *A Few Useful Things to Know about Machine Learning*. Available at: https://sites.astro.caltech.edu/~george/ay122/cacm12.pdf (Accessed: May 5, 2023).

[18] Goodfellow, I., Bengio, Y. and Courville, A. (no date) *Deep learning*, *Deep Learning*. Available at: https://www.deeplearningbook.org/ (Accessed: May 5, 2023).

[19] *Precision and recall* (2023) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Precision_and_recall (Accessed: May 5, 2023).

[20] Korstanje, J. (2021) *The F1 score*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6 (Accessed: May 5, 2023).

[21] *Accuracy and precision* (2023) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Accuracy_and_precision (Accessed: May 5, 2023).

[22] Géron Aurélien (2023) *Hands-on machine learning with scikit-learn, Keras, and tensorflow concepts, tools, and techniques to build Intelligent Systems*. Beijing: O'Reilly.