UNIVERSITI
MALAYA

Faculty of Computer Science
and Information Technology

WIA1006/WID3006 MACHINE LEARNING

FINAL REPORT
Group 16 - BruhCode

LECTURER
Dr. Aznul Qalid Bin Md Sabri
&
Dr. Muhammad Shahreeza Safiruz Bin Kassim

Name	Matric Number
Tan Kuang Jack (Leader)	U2005315
Tan Woon Cong	U2005351
Phang Chi Yang	U2005266
Frankie Lim Qi Quan	U2005263
Marcus Meow Zi En	U2005311

Table of Contents

Introduction to problem

Nowadays, depression is rising among teenagers. The teenagers are living in a stressful and competitive environment which leads to depression and anxiety. This may lead to another serious problem which is suicide. The suicide rate has been increasing over the year. Depression can creep up gradually on a person without them realizing it. It is dangerous as depressed thoughts and feelings will affect people's perspectives and daily lives. Some of the effects include loss of appetite, erratic sleep habits, etc. Recently, there are numerous cases of suicide at the Penang Bridge in Malaysia. Last month, a total of 8 suicide cases happened at the Penang Bridge. This shows an increase in the suicidal rate and we should take account into this issue by cutting the depression happening among the teenagers. Early identification of depressive symptoms is a crucial first step toward assessment, intervention, and relapse prevention. Regarding this issue, are there any ways to detect and identify the symptoms of depression at an early stage so the necessary treatment can be given to the patients before the depression gets worse?

Hypothesis made for the problem

Expressing our feelings could be done in many ways. One of the most common ways emotion could be expressed is through words. By analyzing captions, messages, diaries, chats and etc that are written in words by somebody, we could know the feelings or emotions of the person. Thus, we can also determine whether the person has depression or not through their expressions in words. It is expected that a person is having depression if his probability of having depression is greater or equal to 0.5 and vice versa.

Likert scale questions can be used to ask respondents to rate items in order of importance, preference, or agreement. Thus, a series of likert questions can be prepared to let people respond. We can use these responses in the form of a scale to determine the severity of depression of the user.

Project Objectives

This project aims to:

1) Develop a model to predict the probability of a person having depression.
    - Multinomial Naive Bayes is used to analyze the text entered by the user and predict whether the text is depressive or not along with its probability.

2) Classify the severity of depression of the user whether it is mild/severe if the probability of the user having depression is equal to or greater than 0.5.
    - Logistic regression model is used to classify whether the person is having a mild or severe depression based on their symptoms and behaviors. If the predicted

output is 0, then the person is classified as having mild depression. If the predicted output is 1, then the person is classified as having severe depression.

3) Suggest activities or actions that need to be taken in order to overcome depression based on the severity of the depression.
    - If the user is suspected to have mild depression, the system will suggest the user to have social bonding, exercise, etc. If the user is suspected to have severe depression, the system will suggest the user to seek help or have treatment from professional mental health organizations.

Methodology

1. Data Collection

For Part 1, we managed to find the Sentiment140 dataset on Kaggle which is used for sentiment analysis. This dataset contains 1,600,000 tweets extracted using the Twitter API. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment. It contains the following 6 fields:

No.	Field	Description	Example
1	target	the polarity of the tweet	0 = negative, 2 = neutral, 4 = positive
2	ids	The id of the tweet	2087
3	date	the date of the tweet	Sat May 16 23:58:44 UTC 2009
4	flag	The query	lyx. If there is no query, then this valu NO_QUERY.
5	user	the user that tweeted	robotickilldozr
6	text	the text of the tweet	Lyx is cool,

However, we realized that there are some limitations of the dataset. Since this dataset is mainly used for general-purpose sentiment analysis to tell if the sentiment of a given text is either positive or negative, it is hard for us to conclude those negative sentiment tweets are a good indication of depression. In other words, we are not sure if the negative tweets are depressive. Therefore, we decided to collect the data for depressive tweets ourselves. This is done by scraping the tweets on Twitter. Since the

Twitter API requires a developer account and imposes a limit on the number of requests that can be made within ascertain duration, we move on to the Twitter Intelligence Tools (TWINT) library found on GitHub as an alternative.

Tweets indicating depression were retrieved using the Twitter scraping tool TWINT using linguistic markers (keywords) indicative of depression. The markers include but are not limited to **"depression", "hopeless", "depressed", "lonely", "suicide",** and **"antidepressant"**. The scraped tweets may contain tweets (noise) that do not indicate that the user is having depression. For instance, the tweets that link to articles about depression. As a result, the scraped tweets will need to be manually checked for better testing results.

After collecting the data for our depressive tweets (positive class), we still need other random tweets for the negative class. This is achieved by randomly picking the positive-sentiment tweets from the Sentiment140 dataset. The random tweets are then combined with the depressive tweets scraped earlier to become the dataset that we will be using in training our machine learning model. The fields selected are id, target and text because other fields like the date of the tweet and the user that tweeted it are not very useful in our opinion. The proportion of positive and negative examples in our dataset is about 40% and 60% respectively so that we do not have a skewed distribution of data.

	id	text	target
0	1533440155213737985	Me as a therapist: Why are you depressed your ...	1
1	1533440054420439040	I used to feel depressed but now that I'm livi...	1
2	1533439990641876993	@UnitedStandMUFC no	1
3	1533439977740173312	I'm thinking about deactivating my account I'm...	1
4	1533439972690210816	@jnklovebot a depressed one, a pick me up one,...	1
...	...	...	...
101604	1827995994	@Sarah_Burgess Yay! Ok Mami. ill text u in a ...	0
101605	2177947516	@AnnaLucyHewitt why do you bother writing to f...	0
101606	2180474899	Woo new followers Woke up at nine then attem...	0
101607	1882408206	just made a payment on my etsy bill. Ouch. G...	0
101608	2190278935	i want season tickets to the dodgers & pad...	0

101609 rows × 3 columns

For part 2, We have created our own dataset to train our logistic regression model to classify a person whether he/she is having a mild or severe depression based on their symptoms and behaviors. The dataset contains 500 observations. It also contains the following 6 fields:

No.	Field	Description	Data Type	Example
1	Little interest or pleasure in doing things	Indicate the frequency of having little interest in carrying out daily activities.	int64	0 = Not at all 1 = Several days 2 = More than half the days 3 = Nearly every day
2	Feeling down, depressed, or hopeless	Indicate the frequency of feeling down, depressed, or hopeless.	int64	0 = Not at all 1 = Several days 2 = More than half the days 3 = Nearly every day
3	Feeling bad about yourself	Indicate the frequency of feeling bad about oneself.	int64	0 = Not at all 1 = Several days 2 = More than half the days 3 = Nearly everyday
4	Feeling tired or having little energy	Indicate the frequency of being exhausted.	int64	0 = Not at all 1 = Several days 2 = More than half the days 3 = Nearly every day
5	Thoughts that you would be better off dead, or of hurting yourself	Indicate the frequency of having suicidal or self-hurting thoughts.	int64	0 = Not at all 1 = Several days 2 = More than half the days 3 = Nearly everyday
6	Depression_level	Indicate the depression level (Mild/Severe)	Object (text)	“Severe”

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 6 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   Little interest or pleasure in doing things           500 non-null    int64
 1   Feeling down, depressed, or hopeless                  500 non-null    int64
 2   Feeling bad about yourself                            500 non-null    int64
 3   Feeling tired or having little energy                 500 non-null    int64
 4   Thoughts that you would be better off dead, or of hurting yourself  500 non-null    int64
 5   Depression_level                                      499 non-null    object
dtypes: int64(5), object(1)
memory usage: 23.6+ KB
```

	Little interest or pleasure in doing things	Feeling down, depressed, or hopeless	Feeling bad about yourself	Feeling tired or having little energy	Thoughts that you would be better off dead, or of hurting yourself	Depression_level
0	1	1	3	1	2	Severe
1	1	0	3	0	1	NaN
2	0	2	3	0	2	Mild
3	2	0	3	1	1	Mild
4	1	1	3	3	3	Severe
...	...	...	...	...	...	...
495	1	1	1	0	1	Mild
496	1	0	1	1	0	Mild
497	1	0	1	1	0	Mild
498	1	0	1	1	0	Mild
499	1	0	1	1	0	Mild

500 rows × 6 columns

From this dataset, we can observe that the first 5 columns will be the input features which representing the symptom and behaviors and the last column which is **"Depression_level"** will be the output that we want to predict.

2. Data Preprocessing

a) Text Cleaning

The text data that we collected by scraping the tweets on Twitter contains a lot of noise, this takes the form of special characters such as hashtags, punctuation, and numbers. All of which are difficult for computers to understand if they are present in the data. We need to, therefore, process the data to remove these elements.

Overall, we have done 10 steps in cleaning the text data:

1) Expand contraction
2) Remove mentions @
3) Remove hyperlinks/URL
4) Remove emojis
5) Remove punctuations
6) remove special characters
7) Remove numeric
8) Convert to lowercase
9) Remove stopwords
10) Remove gibberish words

```
# expand contraction
# I'll -> I will
tweets = tweets.apply(contractions.fix)

# remove mentions
tweets = tweets.str.replace("@[A-Za-z0-9_]+","", regex=True)

# Remove hyperlink
tweets = tweets.str.replace(r"http\S+", "", regex=True)

# remove emoji
tweets = tweets.str.replace('[^\w\s#@/:%.,_-]', '', flags=re.UNICODE, regex=True)

# remove punctuation
tweets = tweets.apply(lambda text: strip_punctuation(text))

# remove special characters (@, #, $, ...)
tweets = tweets.apply(lambda text: strip_non_alphanum(text))

# remove numeric
tweets = tweets.apply(lambda text: strip_numeric(text))

# strip multiple whitespaces
tweets = tweets.apply(lambda text: strip_multiple_whitespaces(text))

# convert all words to lowercase
tweets = tweets.str.lower()
```

```
[ ]  # remove stopwords
     tweets = tweets.apply(lambda text: remove_stopwords(text))
```

Remove gibberish words
e.g. ajdbofsbg

```
[ ]  path = '/content/drive/MyDrive/ML Workspace/ASSIGNMENT/big.model'
     Detector = detector.create_from_model(path)

     def remove_gibberish(text):
         words = word_tokenize(text)
         not_gibberish = [w for w in words if not Detector.is_gibberish(w)]
         return " ".join(not_gibberish)
```

```
[ ]  clean_tweets = tweets.apply(remove_gibberish)
```

b) Missing data

The dataset that we found may contain some null values which is denoted by NaN. The machine learning model will provide an error if we pass NaN values into it. Therefore, we need to clean the missing data before we feed the data to the machine learning model for training. First of all, we make use of the **.isnull().sum()** function to check whether

there are any missing values in the dataset. Then, we assign default values for each data type of the dataset. Lastly, we iterate through all of the columns of the dataset and fill in the null values(NaN) with the default values that we previously defined. The figures below illustrate how the cleaning process is being done.

Before cleaning the missing data:

```
data.isnull().sum().max() #just checking if there's any missing data missing
```

```
1
```

```
data.head(5)
```

	Little interest or pleasure in doing things	Feeling down, depressed, or hopeless	Feeling bad about yourself	Feeling tired or having little energy	Thoughts that you would be better off dead, or of hurting yourself	Depression_level
0	1	1	3	1	2	Severe
1	1	0	3	0	1	NaN
2	0	2	3	0	2	Mild
3	2	0	3	1	1	Mild
4	1	1	3	3	3	Severe

Code for cleaning the missing data:

```
# Clean the missing Data
# Assign default values for each data type
defaultInt = 0
defaultString = 'Mild'
defaultFloat = 0.0

intFeatures = ['Little_Interest','Down_Depressed',"Tired_NoEnergy",'Feel_Bad',"Suicidal_Thought"]
stringFeatures = ['Depression_level','Scoring']
floatFeatures = []

# Clean the NaN's
for feature in data:
    if feature in intFeatures:
        data[feature] = data[feature].fillna(defaultInt)
    elif feature in stringFeatures:
        data[feature] = data[feature].fillna(defaultString)
    elif feature in floatFeatures:
        data[feature] = data[feature].fillna(defaultFloat)
    else:
        print('Error: Feature %s not recognized.' % feature)
data.head(5)
```

After cleaning the missing data:

	Little_Interest	Down_Depressed	Feel_Bad	Tired_NoEnergy	Suicidial_Thought	Depression_level
0	1	1	3	1	2	Severe
1	1	0	3	0	1	Mild
2	0	2	3	0	2	Mild
3	2	0	3	1	1	Mild
4	1	1	3	3	3	Severe

c) Categorical Data

We have used label encoding to process all the categorical data because the data is nominal data, which cannot be scaled or ranked like ordinal data. Label Encoding refers to converting the labels into a numeric form so as to convert them into a machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

The encoding process is as shown below:

In our case, we have a column called ***Depression_level*** in our dataset.

Depression_level
Severe
Mild
Mild
Mild
Severe

This is the code for the implementation of label encoding.

```
data.Depression_level = [1 if each == "Severe" else 0 for each in data.Depression_level]
data.head(5)
```

After applying label encoding, the ***Depression_level*** column is converted into:

Depression_level
1
0
0
0
1

where **0** is the label for **Mild**, **1** is the label for **Severe**.

3. Data Exploration

Data exploration is the first step in data analysis involving the use of data visualization tools and statistical techniques to uncover data set characteristics and initial patterns. We have utilized some python libraries such as matplotlib, plotly, and seaborn to visualize some of the data to look at the distribution of data. For instance, we plotted a bar graph to visualize the distribution of the random tweets and depressive tweets in the dataset as shown in Figure 1.

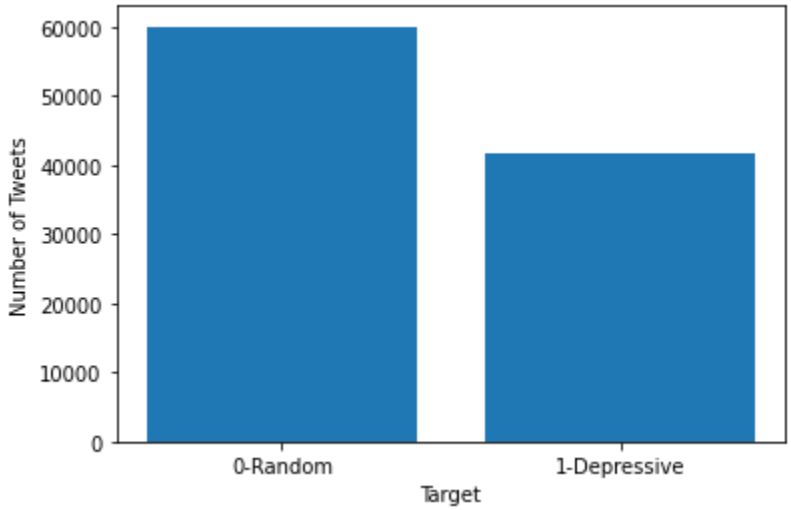
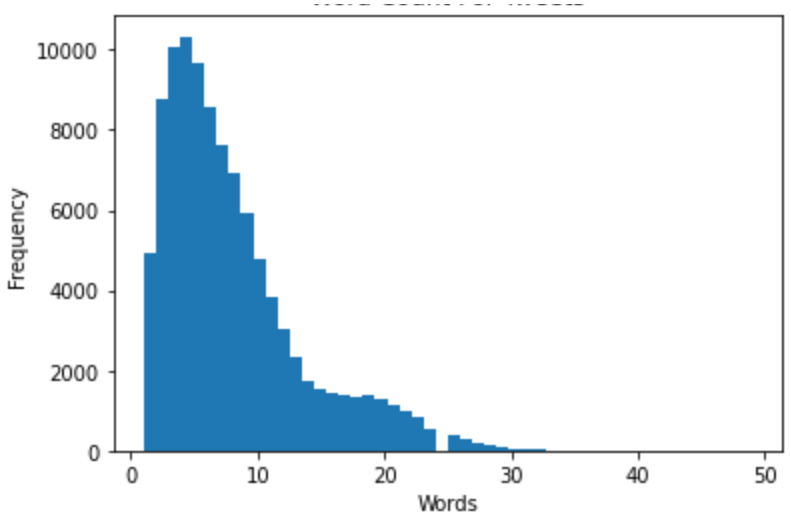


Figure 1: *Bar plot for tweets target label*

From the figure above, we can see that the number of random tweets in the dataset is more than the number of depressive tweets. The proportion of random tweets and depressive tweets are around 0.6 and 0.4 respectively. The data distribution is not very skewed.

We also plotted a histogram to analyze the word count for each tweet as shown in Figure 2 below.



	word_count
count	101609.000000
mean	7.809613
std	5.673539
min	1.000000
25%	4.000000
50%	6.000000
75%	10.000000
max	49.000000

Figure 2: *Histogram of word count For each tweet*

From figure 2, we can see that most of the tweets have a word count of less than 10.

Next, we also plotted a horizontal bar chart to display the top 15 most common words in tweets without stopwords.

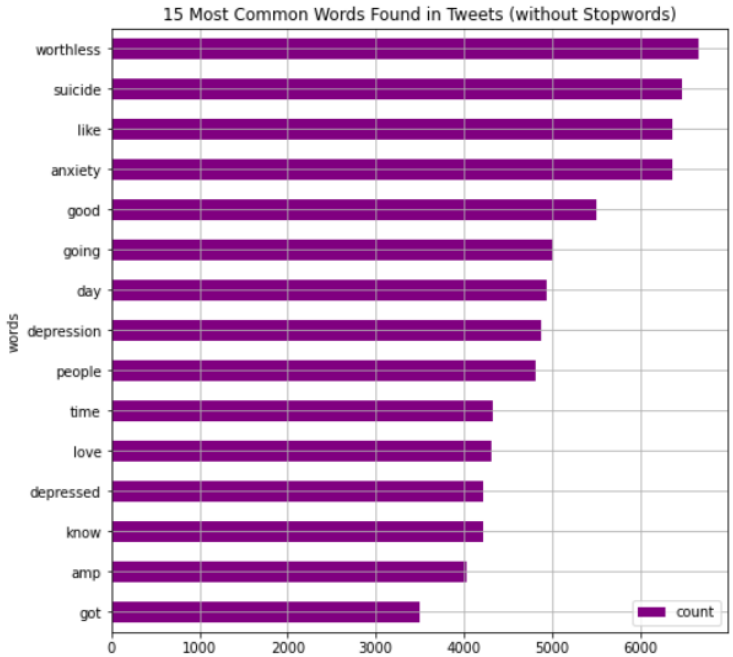


Figure 3: *Horizontal bar chart for top 15 most common words in tweets without stopwords*

From the figure above, we can observe that **“worthless”**, **“suicide”**, and **“anxiety”** are among the most common words found in the tweets without stopwords.

Next, we also created word clouds to display the popular keywords and phrases of the random tweets and depressive tweets from the dataset as shown in **Figure 4** and **Figure 5**.

Figure 4: *Word Cloud of the random tweets from the dataset*

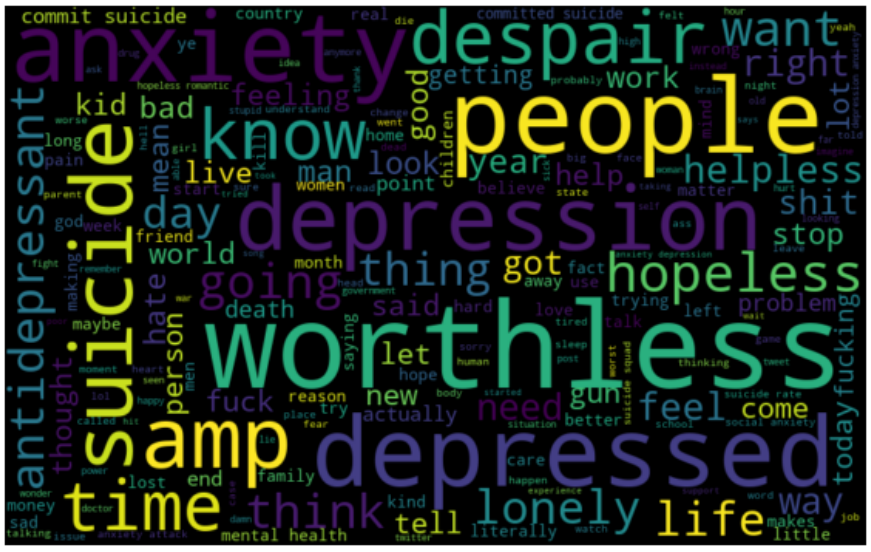


Figure 5: *Word Cloud of the depressive tweets from the dataset*

As shown in the figure above, we can clearly see that **“worthless”**, **“depressed”**, **“suicide”**, **“depression”**, **“despair”** and **“anxiety”** are among the most popular words found in depressive tweets.

We have also plotted the correlation matrix to find out the correlation between the input features as shown in **figure 6** and **figure 7**.

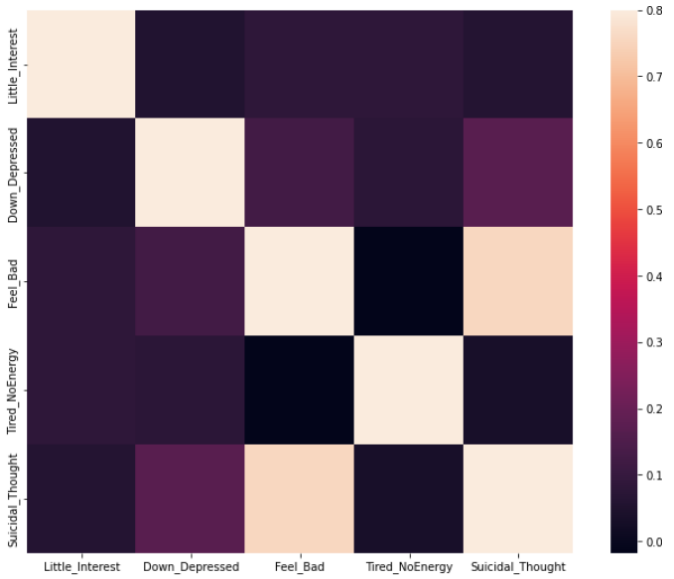


Figure 6: *Correlation matrix of input features*

From the figure above, we can see that there is a relatively strong correlation between the input feature **"Feel_Bad"** and the input feature **"Suicidal_Thought"**.

Besides that, we also plotted the correlation matrix to visualize the correlation between the input features and the output **- "Depression_level"**.

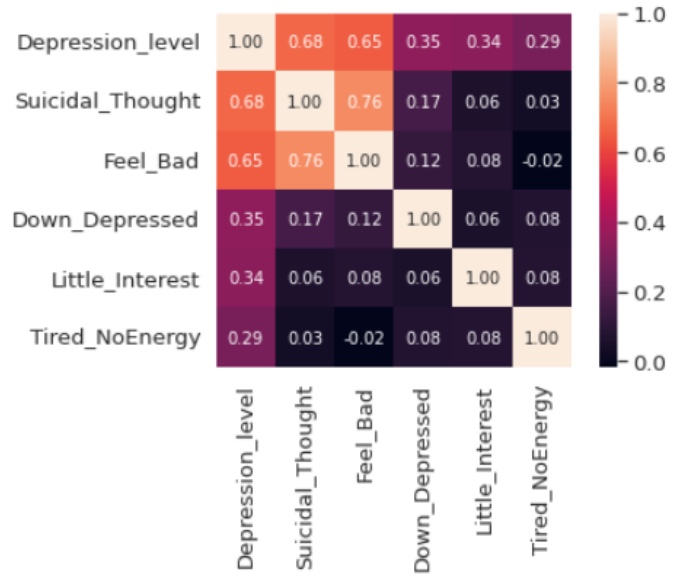


Figure 7: *Correlation matrix of Depression_level and the input features*

From the figure above, we can see the correlation between the input feature **"Suicidal_Thought"** and the output variable which is **"Depression_level"** is the strongest which has a correlation coefficient of 0.68. We can also observe the correlation

between the input feature **"Tired_NoEnergy"** and the output variable **"Depression_level"** is the weakest which has a correlation coefficient of 0.29 only**.**

4. Model Training

a) Model used

As mentioned above, the models implemented are the **Multinomial Naive Bayes model** and **Logistic Regression Model**. The Multinomial Naive Bayes model is used to compute the probability of one having depression while the logistic regression model is used to classify whether he/she is having a mild or severe depression based on their symptoms and behavior.

b) Train set and test set

Before feeding the preprocessed data to the model for training, we split the data into 2 sets, which are the **training set** and **test set** in the portion of **0.6:0.4**. The training dataset is used to build up the model, while a test set is used to validate the model built and inform us about the final accuracy of the model after completing the training phase.

5. Performance Analysis method

a) Model score

We have used the **accuracy_score** method from sklearn to evaluate the accuracy of our Naive Bayes model and the **jaccard_score** method to evaluate the accuracy of our logistic regression model. However, the model's accuracy score alone does not reveal all of a model's shortcomings, therefore we utilized other performance analysis methods such as the confusion matrix and classification reports.

b) Confusion Matrix

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This can give us a better idea of what our classification model is getting right and what types of errors it is making.

c) Classification Report

Classification Report is one of the performance evaluation metrics of a classification-based machine learning model. It displays our model's precision, recall, F1 score, and support. Precision is defined as the ratio of true positives to the sum of true and false positives. Recall is defined as the ratio of true positives

to the sum of true positives and false negatives. The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0, the better the expected performance of the model is. Lastly, support is the number of actual occurrences of the class in the dataset. It doesn't vary between models, it just diagnoses the performance evaluation process. All these metrics can give us a better understanding of the overall performance of our trained model.

After obtaining the result of the model score, confusion matrix, and classification report, if the performance of the model was not satisfactory, we will return to the preprocessing part to make necessary amendments until it shows satisfaction.

Elaboration on Data & Features Used

Data

For Part 1, the dataset consisting of the random and depressive tweets has 101,609 entries. There are 3 columns namely id, text, and target. There is no null value in each of the columns.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101609 entries, 0 to 101608
Data columns (total 3 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   id      101609 non-null  int64
 1   text    101002 non-null  object
 2   target  101609 non-null  int64
dtypes: int64(2), object(1)
memory usage: 2.3+ MB
```

(a) id

The "id" field is the unique id associated with each tweet on Twitter. Its data type is integer. We used the "id" field to detect and remove duplicate tweets from our dataset.

(b) text

The "text" field stores the content of the tweet itself. Its data type is string. We only work on the text in the English language. We use **whatthelang** library to detect the language of the tweets and keep only those written in English. Detecting depression from a text other than English is beyond the scope of our project.

(c) target

The “target” field is the actual label for the tweets, where 0 indicates random tweet and 1 means depressive tweet. Its data type is integer. Since we are using a supervised learning model, it is important for us to provide the model with a labeled dataset.

For part 2, the dataset consisting of data related to the symptoms, behaviors, and level of depression has 500 entries. There are 6 columns in total which are Little_Interest, Down_Depressed, Tired_NoEnergy, Feel_Bad, Suicidal_Thought, and Depression_level.

	Little_Interest	Down_Depressed	Feel_Bad	Tired_NoEnergy	Suicidal_Thought	Depression_level
0	1	1	3	1	2	Severe
1	1	0	3	0	1	Mild
2	0	2	3	0	2	Mild
3	2	0	3	1	1	Mild
4	1	1	3	3	3	Severe

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 6 columns):
 #   Column                                                    Non-Null Count  Dtype
---  ------                                                    --------------  -----
 0   Little interest or pleasure in doing things               500 non-null    int64
 1   Feeling down, depressed, or hopeless                      500 non-null    int64
 2   Feeling bad about yourself                                500 non-null    int64
 3   Feeling tired or having little energy                     500 non-null    int64
 4   Thoughts that you would be better off dead, or of hurting yourself  500 non-null    int64
 5   Depression_level                                          499 non-null    object
dtypes: int64(5), object(1)
memory usage: 23.6+ KB
```

(a) Little_Interest

The “Little_Interest” field indicates the frequency of a person having little interest or pleasure in carrying out daily activities. Its data type is integer. It holds a value ranging from 0 - 3 in which 0 indicates “Not at all”, 1 indicates “Several days”, 2 indicates “More than half the days” and 3 indicates “Nearly every day”. We will use this field as one of the input features in training the logistic regression model to classify the severity of depression whether it is mild or severe.

(b) Down_Depressed

The “Down_Depressed” field indicates the frequency of a person feeling down, depressed, or hopeless. Its data type is integer. It holds a value ranging from 0 - 3 in

which 0 indicates “Not at all”, 1 indicates “Several days”, 2 indicates “More than half the days” and 3 indicates “Nearly every day”. We will use this field as one of the input features in training the logistic regression model to classify the severity of depression whether it is mild or severe.

(c) Feel_Bad

The “Feel_Bad” field indicates the frequency of a person having little interest or pleasure in carrying out daily activities. Its data type is integer. It holds a value ranging from 0 - 3 in which 0 indicates “Not at all”, 1 indicates “Several days”, 2 indicates “More than half the days” and 3 indicates “Nearly every day”. We will use this field as one of the input features in training the logistic regression model to classify the severity of depression whether it is mild or severe.

(d) Tired_NoEnergy

The “Tired_NoEnergy” field indicates the frequency of a person feeling tired or having little energy. Its data type is integer. It holds a value ranging from 0 - 3 in which 0 indicates “Not at all”, 1 indicates “Several days”, 2 indicates “More than half the days” and 3 indicates “Nearly every day”. We will use this field as one of the input features in training the logistic regression model to classify the severity of depression whether it is mild or severe.

(e) Suicidal_Thought

The “Suicidal_Thought” field indicates the frequency of a person having suicidal and self-hurting thoughts. Its data type is integer. It holds a value ranging from 0 - 3 in which 0 indicates “Not at all”, 1 indicates “Several days”, 2 indicates “More than half the days” and 3 indicates “Nearly every day”. We will use this field as one of the input features in training the logistic regression model to classify the severity of depression whether it is mild or severe.

(f) Depression_level

The “Depression_level” field indicates the severity of the depression of a particular person. Its data type is string. “Severe” indicates that the person is having severe depression and “Mild” indicated that the person is having mild depression. Later on, we will encode this field and convert the values into 0 or 1 which indicates “Severe” and “Mild” respectively. This field will act as the dependent variable in training the logistic regression model.

Features

TF-IDF

Every unique word in the tweets (excluding URL, emojis, punctuation, English stopwords, gibberish words and mentions) that made up the text corpus can be considered a feature for our model to learn. A TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer is used to extract the important features in the tweets. Below is the formula for calculating TF-IDF:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Consider an example where we have 3 reviews (documents) for a movie in our corpus:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

Bag of Words

Our model is not able to understand natural language in this case English. Therefore, we need to represent the text in numbers by constructing the Bag of Words. First, we identify all the unique words in the movie reviews to build a vocabulary. Next, we count the occurrence of each of these words in the reviews.

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

By referring to the table above, Review 1 can be represented as a vector [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]. The same rule can be applied to Review 2 and 3.

(a) Term Frequency

$$tf_{t,d} = \frac{n_{t,d}}{Number\ of\ terms\ in\ the\ document}$$

"Term Frequency (TF) is a measure of how frequently a term t appears in a document d". In other words, how many times a particular word appears in the sentence. For example, the word "This" appears one time in Review 1 with total words of 7. Therefore, the TF for "This" in Review 1 is 1/7. We continue to compute the TF for the rest of the words in the vocabulary for each movie review.

Term	Review 1	Review 2	Review 3	TF (Review 1)	TF (Review 2)	TF (Review 3)
This	1	1	1	1/7	1/8	1/6
movie	1	1	1	1/7	1/8	1/6
is	1	2	1	1/7	1/4	1/6
very	1	0	0	1/7	0	0
scary	1	1	0	1/7	1/8	0
and	1	1	1	1/7	1/8	1/6
long	1	0	0	1/7	0	0
not	0	1	0	0	1/8	0
slow	0	1	0	0	1/8	0
spooky	0	0	1	0	0	1/6
good	0	0	1	0	0	1/6

Table 1: *Term Frequency (TF) for each unique word in the movie reviews*

(b) Inverse Document Frequency

$$idf_t = \log \frac{number\ of\ documents}{number\ of\ documents\ with\ term\ 't'}$$

"Inverse Document Frequency (IDF) is a measure of how important a term is". If a word can be found in many documents, then it is not very important. In contrast, if a word can only be found in a particular document, then it is very important for that document. A word with a low IDF value indicates that it is not that important while a word with a high IDF value indicates that it is more important. For instance, the word "This" appears in all the 3 reviews (documents) and

its IDF is calculated as log(3/3) = log(1) = 0. Therefore, the word “This” has little importance in the movie reviews.

Term	Review 1	Review 2	Review 3	IDF
This	1	1	1	0.00
movie	1	1	1	0.00
is	1	2	1	0.00
very	1	0	0	0.48
scary	1	1	0	0.18
and	1	1	1	0.00
long	1	0	0	0.48
not	0	1	0	0.48
slow	0	1	0	0.48
spooky	0	0	1	0.48
good	0	0	1	0.48

Finally, we can get the TF-IDF of each word by multiplying its TF with its IDF. Words with a higher TF-IDF value are more important and vice versa.

Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.000	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	0	0.48	0.068	0.000	0.000
not	0	1	0	0.48	0.000	0.060	0.000
slow	0	1	0	0.48	0.000	0.060	0.000
spooky	0	0	1	0.48	0.000	0.000	0.080
good	0	0	1	0.48	0.000	0.000	0.080

By using the TF-IDF approach, we are able to capture the significant words as features that indicate whether the tweet is depressive or not.

	Word	TF-IDF
0	good	1308.987869
1	anxiety	1291.838050
2	worthless	1186.446345
3	like	1097.280717
4	depressed	1072.857668
5	day	1040.962662
6	love	1037.946019
7	suicide	1028.186611
8	thanks	1018.777575
9	going	1015.134750

Figure 8: *Top 10 most significant words in the tweets dataset*

N-grams

A N-gram is basically a sequence of N words in a document. For example, “Natural” is 1-gram or unigram because it contains only one word, “Natural Language” is a 2-gram or bigram, “Natural Language Processing” is a 3-gram or trigram, so on and so forth.

English is amazing because when the specific words are combined in a correct sequence to form a phrase, it gives us a new meaning. For instance, “Machine” and “Learning” when joined together becomes “Machine Learning” which is a field of computer science that make the computer learn from studying data and statistics.

Besides making every single word in our corpus as the features to train our model, we can also obtain more useful features by finding the n-grams (where $n > 1$) from the tweets. Since the English phrases with more than 3 words is uncommon and rarely used in the real world, we only generate the n-grams in the range of 1 up to 3 inclusively.

	bigram	count
0	(good, morning)	595
1	(commit, suicide)	533
2	(mental, health)	517
3	(feel, like)	447
4	(committed, suicide)	361
5	(social, anxiety)	340
6	(suicide, squad)	300
7	(good, luck)	299
8	(good, night)	289
9	(hopeless, romantic)	273

Table 2: *Top 10 most common bigrams found in our tweets dataset.*

As shown in the table above, we can see some bigrams such as “good morning”, “commit suicide” and “mental health” which can be the important features to detect whether a given text is depressive or not. Therefore, we can say that it is fruitful for us to generate and include the n-grams in our training data in order to build a more robust model for analyzing depressive text.

Results and discussions

After fitting the dataset into our Naive Bayes model, we evaluated the accuracy of our model using the accuracy_score method from sklearn.

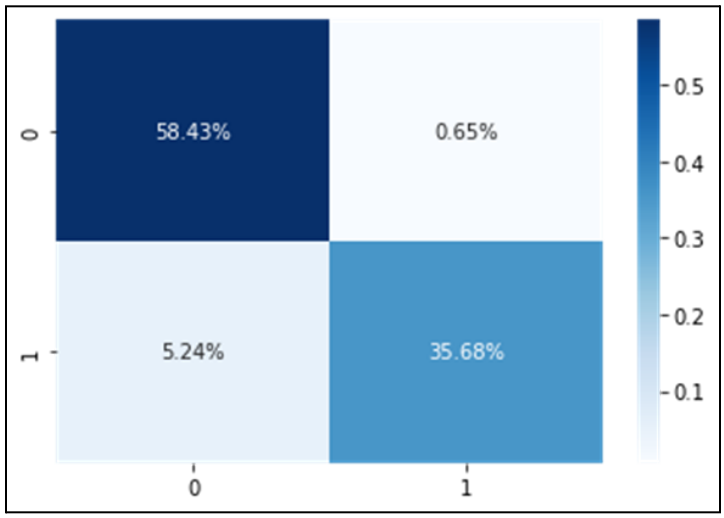
Accuracy of our model:

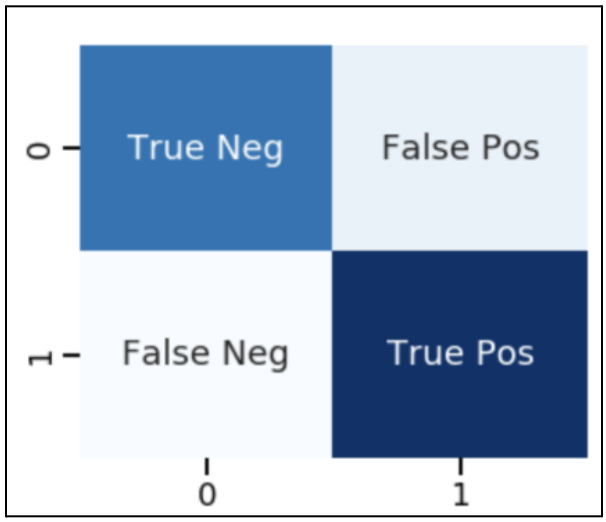
```
accuracy_score(y_test, validation)

0.9411533162763236
```

To see the effectiveness of our model, we use a confusion matrix to measure the performance of the model. 1 (positive) means depression and 0 (negative) means not depressed.

Y-axis = actual label

X-axis = predicted label





From the confusion matrix, we can compute **precision** and **recall** easily.

Precision: From all the classes we have predicted as positive, how many are actually positive, therefore the higher the better. Precision tells us how much the proportion of positives was correctly identified.

$$Precision = \frac{TP}{TP + FP}$$

Recall: From all the positive classes, how many we predicted correctly, therefore the higher the better. Recall tells us how much the proportion of actual positives was correctly identified.

$$Recall = \frac{TP}{TP + FN}$$

F1-score: Measures recall and precision at the same time. It is useful when the distribution of classes is not balanced.

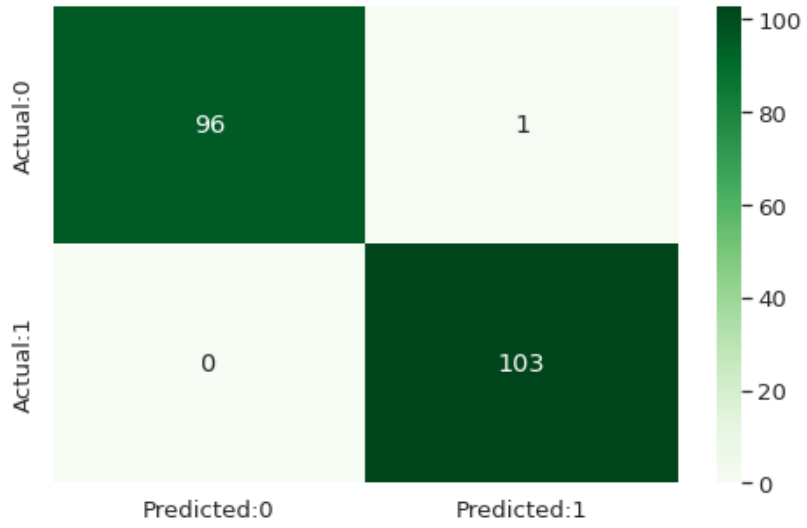
$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision}$$

	precision	recall	f1-score	support
0	0.92	0.99	0.95	3002
1	0.98	0.87	0.92	2079
accuracy			0.94	5081
macro avg	0.95	0.93	0.94	5081
weighted avg	0.94	0.94	0.94	5081

The picture above shows the classification report of the model, which includes accuracy, precision, recall, and f1-score. Recall is one of the important performance metrics in this model because we know that the cost of false negatives, in this case, is very high. This is due to the fact that when we predict a depressed user (true positive) as not depressed (false negative), the consequences can be severe. The recall (sensitivity) as shown in the table above is 87%, while specificity is 99%. The macro average recall and weighted average recalls are 93% and 94%, respectively. The accuracy, macro average f1-score, and weighted average f1-score are 94% as well. The difference between a macro average and a weighted average is that the macro average treats all classes equally regardless of support values, as it is computed by taking the arithmetic mean of all per-class values, while the weighted average is calculated by taking the mean of all per-class values while considering each class's support. Therefore, a macro average would be a good choice for a dataset with all classes equally important, while a weighted average is preferred when we want to assign greater contributions to classes with more examples in the dataset.

Besides determining whether or not the user is depressed, we also trained another logistic regression model to classify whether the user is having mild or severe depression. 1 means severe depression and 0 means mild depression.

Confusion matrix:



Classification report:

The details for confusion matrix is =

	precision	recall	f1-score	support
0	1.00	0.99	0.99	97
1	0.99	1.00	1.00	103
accuracy			0.99	200
macro avg	1.00	0.99	0.99	200
weighted avg	1.00	0.99	0.99	200

Since the datasets are pretty balanced in this model, accuracy will be a good performance metric, which is 99%. Accuracy gives a good overall performance regardless of the class.

The trained **Multinomial Naive Bayes model** will be used to predict whether the user is depressed or not in our program's **chatbot** and compute the depression probability for the user. After determining whether or not the user is depressed, our program will utilize the trained **logistic model** to determine if the user is suffering from mild or severe depression based on the user's symptoms and behaviors. When a user enters the text, our program predicts whether or not the user is depressed and computes the probability of depression. Throughout the chat session, our program will calculate the user's overall probability of depression. When the overall probability is more than 0.5, it will prompt the user to take a depression test, and the program will classify whether the user is having mild or severe depression based on the user's symptoms

and behaviors. The program will then recommend several suitable activities to overcome depression based on the type of depression.

Chatbot:

```
depression_indicator.chat()
```

```
Hi! I am Bingo. Nice to meet you!
Enter Here or Q to quit: Hi Bingo, I am anonymous
Not Depressed
Probability: 0.26294479937330173
Overall Probability: 0.26294479937330173

Enter Here or Q to quit: I feel lonely. Nobody likes me.
Depressed
Probability: 0.8973544873229944
Overall Probability: 0.5801496433481481

Enter Here or Q to quit: I hate myself and this world
Depressed
Probability: 0.5372185860651965
Overall Probability: 0.5658392909204976

Enter Here or Q to quit: Suicide is my only option
Depressed
Probability: 0.7733975483376498
Overall Probability: 0.6177288552747857

Enter Here or Q to quit: Q
Bingo has left the chat...
```

Mild depression result:

```
We detected that you might have depression.

We strongly suggest you to take the depression test below.

Over the last 2 weeks, how often have you been bothered by any of the following problems?

Rate from 0 - 3
0 represents NOT AT ALL
1 represents SEVERAL DAYS
2 represents MORE THAN HALF THE DAYS
3 represents NEARLY EVERY DAY

Little interest or pleasure in doing things: 0
Feeling down, depressed, or hopeless: 1
Feeling bad about yourself: 2
Feeling tired or having little energy: 2
Thoughts that you would be better off dead, or of hurting yourself: 0

Result:2

You have a Mild Depression

Suggestion 1: Exercise
-There is strong evidence that any kind of regular exercise is one of the best antidepressants.
-Exercise helps to lower symptoms of anxiety, improve sleep quality, and boost energy levels.

Suggestion 2: Gratitude
-Expressing gratitude has been shown to have a positive emotional effect on people with depression
-What you appreciate in your life can increase activity in the medial prefrontal cortex, the brain region often associated with depression.

Suggestion 3: Social connection
-Join a group devoted to something for which you have a strong passion.
-For instance, volunteering for a favorite cause can keep you connected with others on a regular basis, plus you have the extra motivation
 to engage because of your personal interest.
```

Severe depression result:

```
We detected that you might have depression.

We strongly suggest you to take the depression test below.

Over the last 2 weeks, how often have you been bothered by any of the following problems?

Rate from 0 - 3
0 represents NOT AT ALL
1 represents SEVERAL DAYS
2 represents MORE THAN HALF THE DAYS
3 represents NEARLY EVERY DAY

Little interest or pleasure in doing things: 2
Feeling down, depressed, or hopeless: 2
Feeling bad about yourself: 2
Feeling tired or having little energy: 2
Thoughts that you would be better off dead, or of hurting yourself: 3

Result:

You have a Severe Depression

Suggestion 1: Music Therapy Music has powerful effects on the mind
-Different styles of music can have a significant effect on a person’s mood very quickly.
-It can help them experience and process a wide range of emotions, from happiness to excitement, as well as sadness, calmness, and thoughtfulness.

Suggestion 2: Seek help from mental health helplines
First Organisation: MALAYSIAN MENTAL HEALTH ASSOCIATION (MMHA)
Contact Number: 03-2780 6803
E-Mail: admin@mmha.org.my
Website: https://mmha.org.my/
Second Organisation: SOLS HEALTH
Contact Number: 6018-900-7247
E-Mail: center@thethrive.center / info@thethrive.center
Website: https://www.thethrive.center/
```

Suggestions for future works

Currently, we are developing a chatbot to help those who think they are having depression and need an indicator to understand their current situation. This is able to help those who understand themselves. However, in most situations, people do not know they are depressed and do not ask for help from others due to different personal reasons. In this era, people will be more likely to share their feelings and thoughts on social media. By knowing this habit, we should indeed integrate the depression indicator with all of the social media like Facebook, Instagram, Twitter, WeChat, etc. With this, we would be able to detect the depression of someone earlier from their post and get to help them when they are still having light symptoms. How do we help them? When the depression indicator detected someone might have the possibility to have depression, we can use the recommender system of the respective social media to pop up posts that may heal them on the users' homepage. For example, memes, cute animals, warm-hearted sentences, healing music and etc.

Besides that, currently, how we determine whether a person is depressed or not is totally based on our own opinions which might be unreliable. We do not get any advice from those who are professionals in this field. In order to get a more accurate prediction, we should get opinions from experts in Psychology. Same to the suggestions to help those who are having depression, the suggestions should be improvised by those who are having knowledge in this field so that it is really effective to cure someone.

Appendix

Link to the presentation video, PowerPoint slides, and the source codes:
https://drive.google.com/drive/folders/1-7Us2h_3L2eRrZnvN9yOPOnCcTnD_EPP?usp=sharing

id	text	target
1.533E+18	therapist depressed ass fat	1
1.533E+18	feel depressed living little obsessed	1
1.533E+18		1
1.533E+18	thinking deactivating account depressed	1
1.533E+18	depressed pick mood swing flirty list goes normal people	1
1.533E+18	dsp wants like characters depressed shit	1
1.533E+18	today got flowers ring welp happy unfortunately depressed	1
1.533E+18	article says depressed	1
1.533E+18	kids depressed	1
1.533E+18	depressed person earth	1
1.533E+18	believe zoloft week month caused manic episode keeping manic episod	1
1.533E+18	week miss depressed coming	1
1.533E+18	embarrassing losing home shit barça	1
1.533E+18	gabbie hanna great person god dammit honestly encore gets stuck heac	1
1.533E+18	family asking want pride parade today nice feeling wildly depressed	1
1.533E+18	kids depressed dress	1
1.533E+18	everytime depressed selfs macro buffer depression lowered	1
1.533E+18	evangelion fans depressed little boy	1
1.533E+18	balling week keeps sane days depressed missing fun	1
1.533E+18	angry stab body depressed stab body happy stab body tired tired saw x	1
1.533E+18	depressed today time putting urself religious filth talk education health	1
1.533E+18	got mad said hated mappa	1

Figure 9: *Cleaned Dataset for Part 1*

Little interest	Feeling down	Feeling bad abou	Feeling tired or	Thoughts that	Depression_level
1	1	3	1	2	Severe
1	0	3	0	1	
0	2	3	0	2	Mild
2	0	3	1	1	Mild
1	1	3	3	3	Severe
2	2	3	2	0	Severe
2	3	3	0	2	Severe
2	3	3	1	2	Severe
1	3	3	1	3	Severe
1	1	3	1	2	Severe
0	2	3	0	1	Mild
1	3	3	1	1	Severe
3	2	3	0	0	Severe
3	0	3	1	3	Severe

Figure 10: *Raw Dataset for Part 2*

References

Bhatti, S. A. (2020, March 20). *Tackling depression with machine learning via Chatbot*. Medium. Retrieved June 16, 2022, from https://medium.com/analytics-vidhya/tackling-depression-with-machine-learning-via-chatbot-1e5f5546f36a

Chugh, A. (2022, May 18). *ML: Label encoding of datasets in Python*. GeeksforGeeks. Retrieved June 16, 2022, from https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/

Huilgol , P. (2020, December 23). *Bow model and TF-IDF for creating feature from text*. Analytics Vidhya. Retrieved June 16, 2022, from https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/

Kairos, A. (2018, June 26). *Machine Learning for Mental Health*. Kaggle. Retrieved June 16, 2022, from https://www.kaggle.com/code/kairosart/machine-learning-for-mental-health-1/notebook

Kazanova, M. M. (2017, September 13). *Sentiment140 dataset with 1.6 million tweets*. Kaggle. Retrieved June 16, 2022, from https://www.kaggle.com/datasets/kazanova/sentiment140

Malik, U. (2020, March 5). *Removing stop words from strings in Python*. Stack Abuse. Retrieved June 16, 2022, from https://stackabuse.com/removing-stop-words-from-strings-in-python/

Mamun, I. (2020, January 14). *Creating a TF-IDF in python*. Medium. Retrieved June 16, 2022, from https://medium.com/@imamun/creating-a-tf-idf-in-python-e43f05e4d424

Martinez, V. R. (2019, February 1). *A machine learning approach for detection of depression and mental illness in Twitter*. Medium. Retrieved June 16, 2022, from https://medium.datadriveninvestor.com/a-machine-learning-approach-for-detection-of-depression-and-mental-illness-in-twitter-3f3a32a4df60

Nithyashree, V. (2021, September 13). *What are N-grams and how to implement them in python?* Analytics Vidhya. Retrieved June 16, 2022, from https://www.analyticsvidhya.com/blog/2021/09/what-are-n-grams-and-how-to-implement-them-in-python/

Pathak, K. (2017, November 30). *Twitter_sentiment*. Kaggle. Retrieved June 16, 2022, from https://www.kaggle.com/datasets/ywang311/twitter-sentiment

Ponaganti, T. (2021, March 3). *Detecting depression in social media via Twitter usage*. Medium. Retrieved June 16, 2022, from https://medium.com/swlh/detecting-depression-in-social-media-via-twitter-usage-2d8f3df9b313

Suresh, A. (2021, June 22). *What is a confusion matrix?* Medium. Retrieved June 16, 2022, from https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Vickery, R. (2020, February 28). *Text cleaning methods for natural language processing*. Medium. Retrieved June 16, 2022, from https://towardsdatascience.com/text-cleaning-methods-for-natural-language-processing-f2fc1796e8c7

Zach. (2022, May 6). *A simple explanation of the jaccard similarity index*. Statology. Retrieved June 16, 2022, from https://www.statology.org/jaccard-similarity/