A Machine Learning

approach for detection of depression in Tweets using

Sentiment Analysis

By

Arshia Sultana

May 13th 2019

sultanaa1@montclair.edu

Instructor: Jing Peng

Montclair State University, NJ, USA

Table of Contents

[Project Overview 2](#_Toc8914894)

[Depression on social media 2](#_Toc8914895)

[Creating a model to detect Depression in Tweets 2](#_Toc8914896)

[Dataset 2](#_Toc8914897)

[Exploratory Data Analysis 3](#_Toc8914898)

[Steps in Part 1 4](#_Toc8914899)

[Word tokenizing 4](#_Toc8914900)

[Splitting the dataset 5](#_Toc8914901)

[Classifier Used 5](#_Toc8914902)

[Decision Tree Classifier 5](#_Toc8914903)

[Results 5](#_Toc8914904)

[Classification report and confusion matrix 6](#_Toc8914905)

[Predictions on the tweets 6](#_Toc8914906)

[Steps in part 2 7](#_Toc8914907)

[Term Frequency Inverse Document Frequency 7](#_Toc8914908)

[Exploratory Data Analysis 8](#_Toc8914909)

[Splitting the dataset 9](#_Toc8914910)

[Classifier Used 9](#_Toc8914911)

[Support Vector Machines 9](#_Toc8914912)

[Naïve Bayes Classifier 10](#_Toc8914913)

[Results 10](#_Toc8914914)

[Applying Decision Tree classifier and finding the accuracy of the 10](#_Toc8914915)

[Applying SVM for classification and finding the accuracy of the model 10](#_Toc8914916)

[Applying Naïve Bayes Classifier for classification and finding the accuracy of the model 10](#_Toc8914917)

[Classification report and confusion matrix for Decision Tree Classifier 10](#_Toc8914918)

[Predictions on the tweets 11](#_Toc8914919)

[(Decision Tree Classifier) 11](#_Toc8914920)

[Classification report and confusion matrix for SVM 11](#_Toc8914921)

[Predictions on the tweets(SVM) 13](#_Toc8914922)

[classification report and confusion matrix for Naïve Bayes Classifier 13](#_Toc8914923)

[Predictions on the tweets(Naïve Bayes Classifier) 15](#_Toc8914924)

[Conclusions and Future Work 15](#_Toc8914925)

# 

[Figure 1 Info about the dataset 4](#_Toc8915599)

[Figure 2 Word Cloud for Depressed Tweets 5](#_Toc8915600)

[Figure 3 Word Cloud for Positive Tweets 6](#_Toc8915601)

[Figure 4 Input Text 7](#_Toc8915602)

[Figure 5 Matrix from Count Vectorizer 7](#_Toc8915603)

[Figure 6 Accuracy Of Decision Tree 8](#_Toc8915604)

[Figure 7 Classification Report for Decision Tree 8](#_Toc8915605)

[Figure 8 Confusion Matrix for Decision Tree 8](#_Toc8915606)

[Figure 9 Predictions for tweets in Part 1 9](#_Toc8915607)

[Figure 10 Word Cloud For Depressed Tweets 11](#_Toc8915608)

[Figure 11 Word Cloud for Positive tweets 11](#_Toc8915609)

[Figure 12 Accuracy score for Decision Tree 12](#_Toc8915610)

[Figure 13 Accuracy Score for SVM 13](#_Toc8915611)

[Figure 14 Accuracy Score for Naive Bayes Classifier 13](#_Toc8915612)

[Figure 15 Classification report for Decision Tree 13](#_Toc8915613)

[Figure 16 Confusion Matrix for Decision Tree 14](#_Toc8915614)

[Figure 17 Predictions for Decision Tree 14](#_Toc8915615)

[Figure 18 Classification report for SVM 15](#_Toc8915616)

[Figure 19 Confusion Matrix for SVM 16](#_Toc8915617)

[Figure 20 Prediction using SVM 17](#_Toc8915618)

[Figure 21 Classification Report for Naive Bayes Classifier 17](#_Toc8915619)

[Figure 22 Confusion Matrix for Naive Bayes Classifier 18](#_Toc8915620)

[Figure 23 Predictions for Naive Bayes Classifier 18](#_Toc8915621)

# Project Overview

Depression is a mental illness that is often ignored. It is a mood disorder characterized by persistently low mood and a feeling of sadness and loss of interest. According to WHO, depression is a common disorder and globally more than 300 million people of all ages suffer from depression. According to American foundation for suicide prevention, over 50 percent of the people who die by suicide suffer from major depression. Whereas, depression is among the most treatable of psychiatric illness (between 80 percent and 90 percent of people with depression respond positively to the treatment). Hence, we can say that many suicides can be avoided by treatment.

# Depression on social media

Social media platforms are becoming an integral part of people’s life. They reflect the user’s personal life. People like to share day to day activities, happiness, joy, and sadness on social media. These platforms are being used by the researchers to collect a lot of data and do some meaningful research on the data

# Creating a model to detect Depression in Tweets

Creating a model to detect depression in tweets:

In Machine Learning, there are many ways for sentiment analysis such: decision-based systems, Bayesian classifiers, support vector machine, neural networks and sample-based methods.

I had used Decision Tree Classifier, SVM, Naïve Bayes Classifier. Hence, I decided to use these three for running the model. The model will be written in python and it will tell whether a given tweet is depressive or not

The model is done in two parts. In Part 1, I wanted to get an idea of how the model performs initially and I had used Count Vectorizer and Decision Tree Classifier.

# Dataset

The sentiment\_tweets3 dataset is taken from the following Github repository

<https://github.com/viritaromero/Detecting-Depression-in-Tweets>

It contains 10,314 tweets. The tweets are labeled 0 for Positive(not depressed tweets) and labelled 1 for Depressed tweets

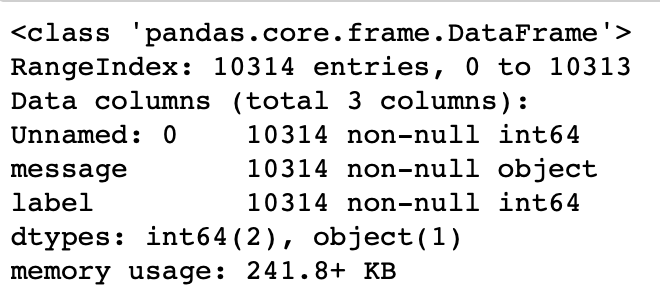


Figure Info about the dataset

# Exploratory Data Analysis

The data in the CVS file is analyzed using Word Cloud. We have seen a cloud filled with lots of words in different sizes, which represent the frequency or the importance of each word. This is called [Tag Cloud](https://en.wikipedia.org/wiki/Tag_cloud) or Word Cloud. Two different word clouds are made. One is made from positive tweets that is displayed in “Red” and the other one is made out of depressed tweets that is displayed in “Blue”. We can make conclusions about what kinds of words are used more in positive tweets.



Figure Word Cloud for Depressed Tweets



Figure Word Cloud for Positive Tweets

# Steps in Part 1

* Word tokenization: Count vectorizer
* Splitting the data
* Classification: Decision tree classifier
* Accuracy
* Confusion matrix
* Analyzing the data
* Classification of text

## Word tokenizing

For this model, I have used a count vectorizer for tokenization. Count vectorizer uses bag of words approach where each message is separated into tokens and the number of times each token occurs in a message is counted. A count vectorizer has built in functions for lowercase, tokenizer, filtering stop words etc. Text is presented in the form of a matrix

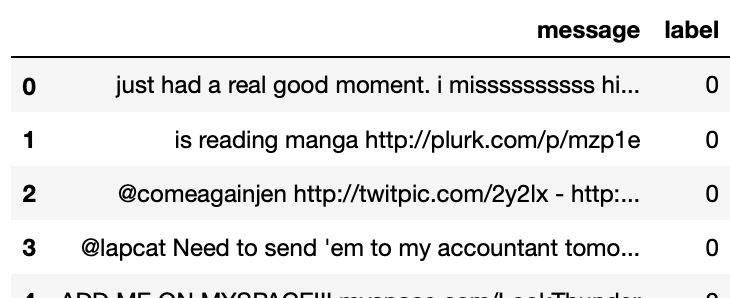


Figure Input Text



Figure Matrix from Count Vectorizer

## Splitting the dataset

**Training and Testing Data**

I split the dataset as follows: 80 % for the Training Data and 20% for Testing Data, where polarity “0” means positive and polarity “1” means depressed. I feed my model with these depressive and positive tweets to learn about them and make predictions.

## Classifier Used

### Decision Tree Classifier

The classifier that I have used is “Decision Tree Classifier”. A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

# Results

After training the model, the results are as follows:



Figure Accuracy of Decision Tree

We can see that the accuracy is really impressive i.e 99.6%

## Classification report and confusion matrix

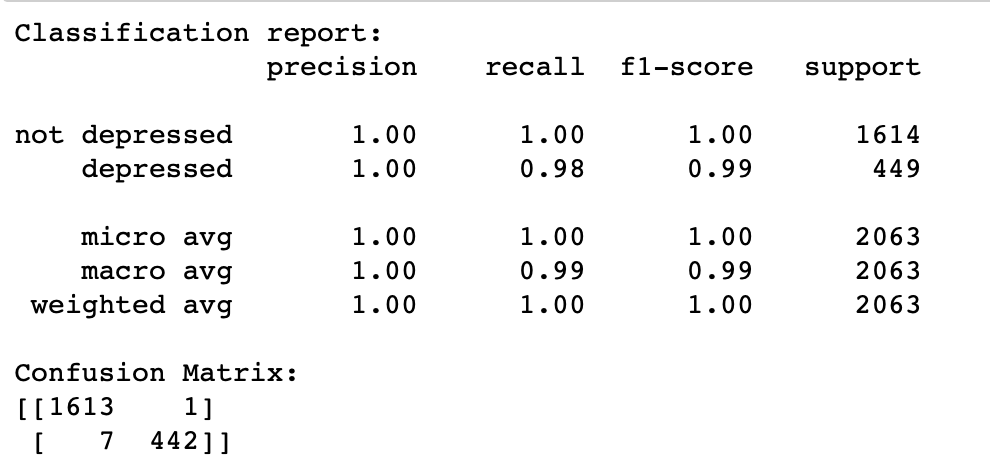


Figure Classification Report for Decision Tree

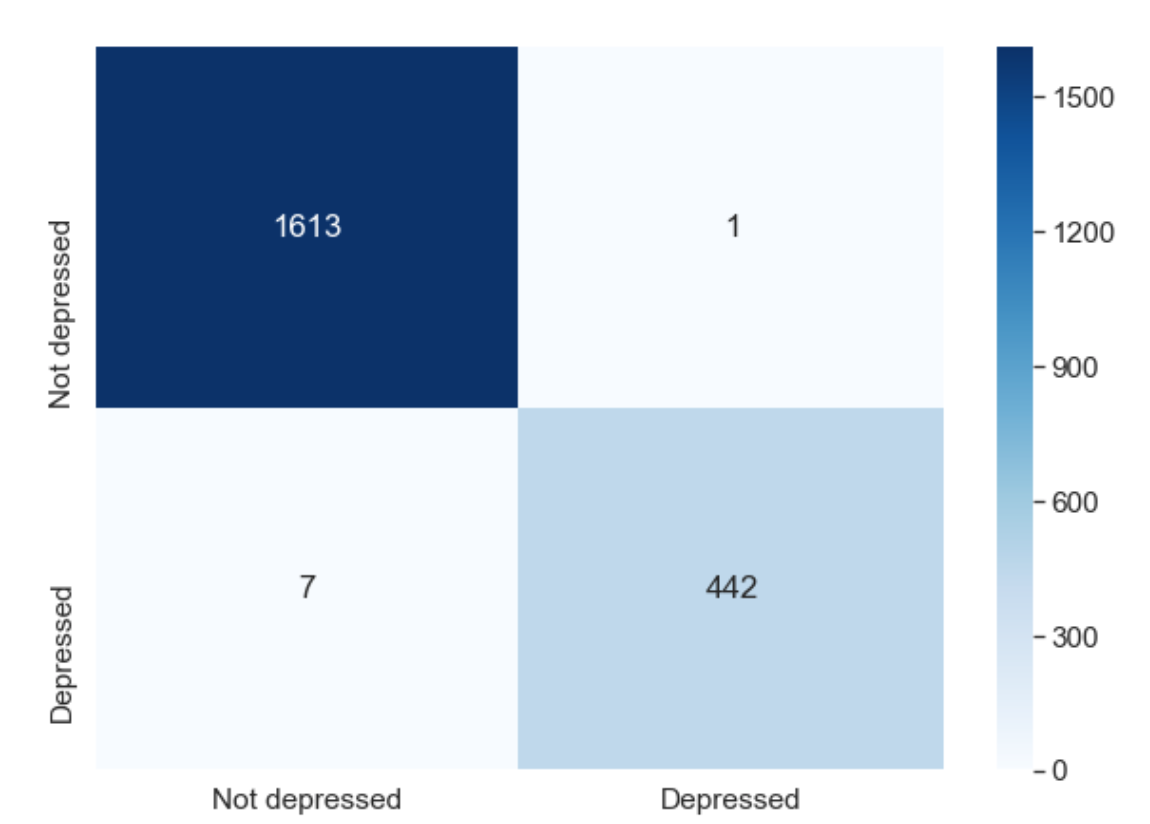


Figure Confusion Matrix for Decision Tree

# Predictions on the tweets

After the model is implemented, I decided to test the model by some tweets.



Figure Predictions for tweets in Part 1

We can notice that even after the accuracy being 99.6% the model was not able to predict the second tweet properly. In the part two, I have used Tf- Idf for tokenization and different classification models in order to improve the model

# Steps in part 2

* Word Tokenization, lower case
* Identifying the Stop words
* Stemming using Porter Stemmer
* Tf-idf vectorizer
* Splitting the dataset
* Decision tree classifier
* SVM
* Naïve Bayes classifier
* Accuracy
* Confusion matrix
* Analyzing the data
* classification

For part 2, Tokenization, converting the text to lowercase, identification of stop words, stemming is done manually in order to increase the efficiency.

## Term Frequency Inverse Document Frequency

Tf-idf stands for*term frequency-inverse document frequency*, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

Typically, the tf-idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

* **TF: Term Frequency**, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:   
    
  TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).
* **IDF: Inverse Document Frequency**, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:   
    
  IDF(t) = log\_e(Total number of documents / Number of documents with term t in it).

# Exploratory Data Analysis

I have created the word cloud once again for the Part 2. We can notice that because we have done the tokenization, stemming ,

converting the text to lowercase, identification of stop words manually, it is evident from the word cloud that the word “depression” is now take as “depress” and it is the same with many other words



Figure Word Cloud For Depressed Tweets

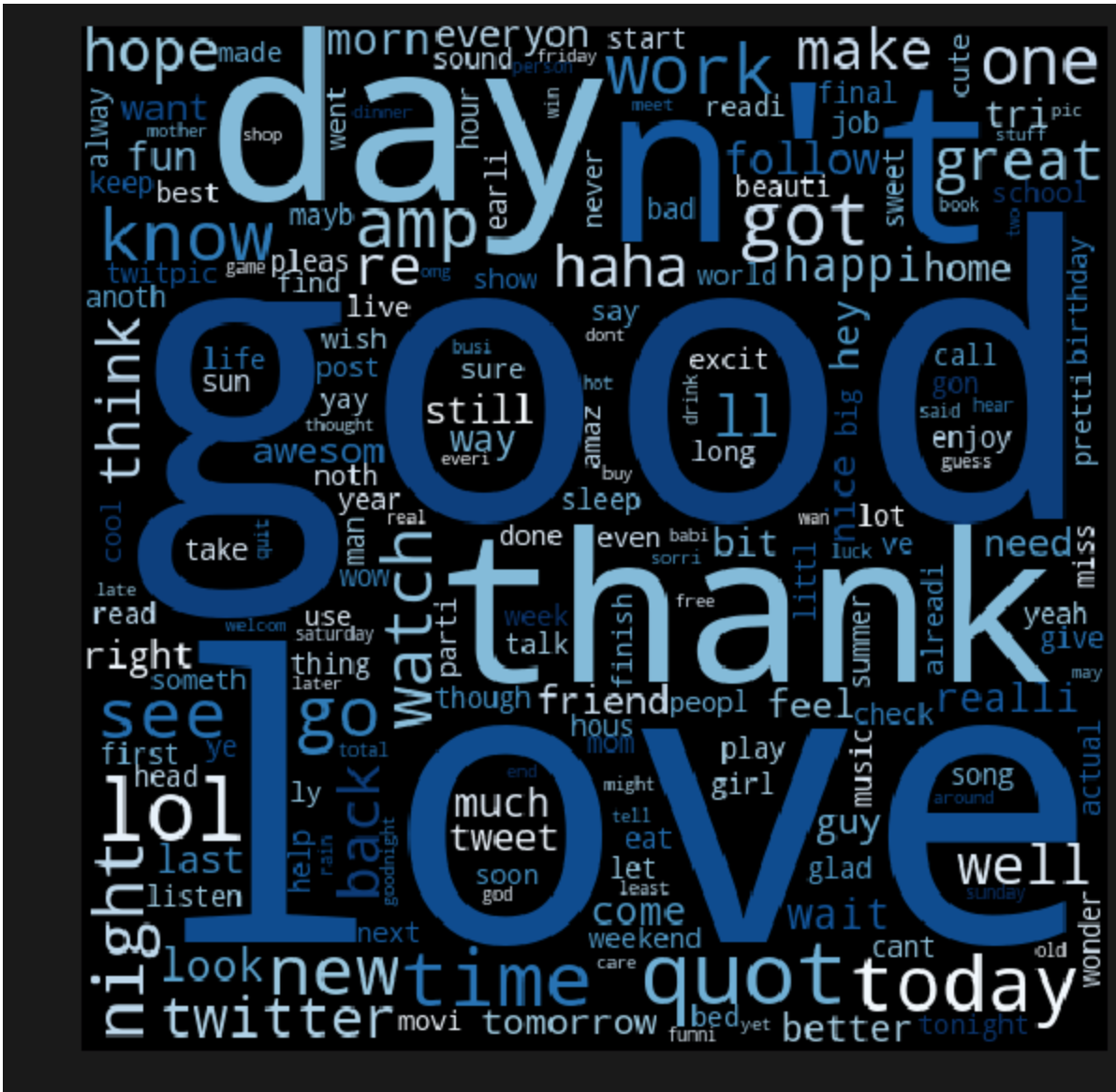


Figure Word Cloud for Positive tweets

## Splitting the dataset

**Training and Testing Data**

I split the dataset as follows: 80 % for the Training Data and 20% for Testing Data, where polarity “0” means positive and polarity “1” means depressed. I feed my model with these depressive and positive tweets to learn about them and make predictions.

## Classifier Used

As in part 1, Decision Tree Classifier is used in part 2 as well. In addition to this two more classifiers are used which are SVM and Naïve Bayes Classifier.

### Support Vector Machines

“Support Vector Machine” (SVM) is a supervised [machine learning algorithm](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle) which can be used for both classification or regression challenges. However,  it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well

### Naïve Bayes Classifier

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes’ Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

# Results

I have used three classifiers and the results are as follows

## Applying Decision Tree classifier and finding the accuracy of the

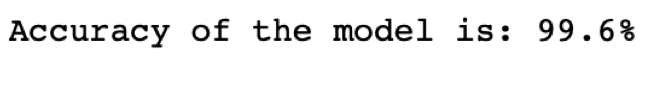


Figure Accuracy score for Decision Tree

## Applying SVM for classification and finding the accuracy of the model

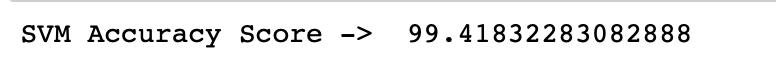


Figure Accuracy Score for SVM

## Applying Naïve Bayes Classifier for classification and finding the accuracy of the model

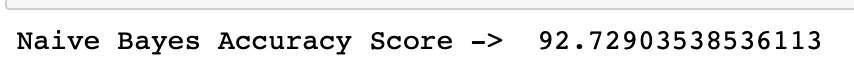


Figure Accuracy Score for Naive Bayes Classifier

We can notice that the accuracy score is the least in Naïve Bayes Classifier.

# Classification report and confusion matrix for Decision Tree Classifier

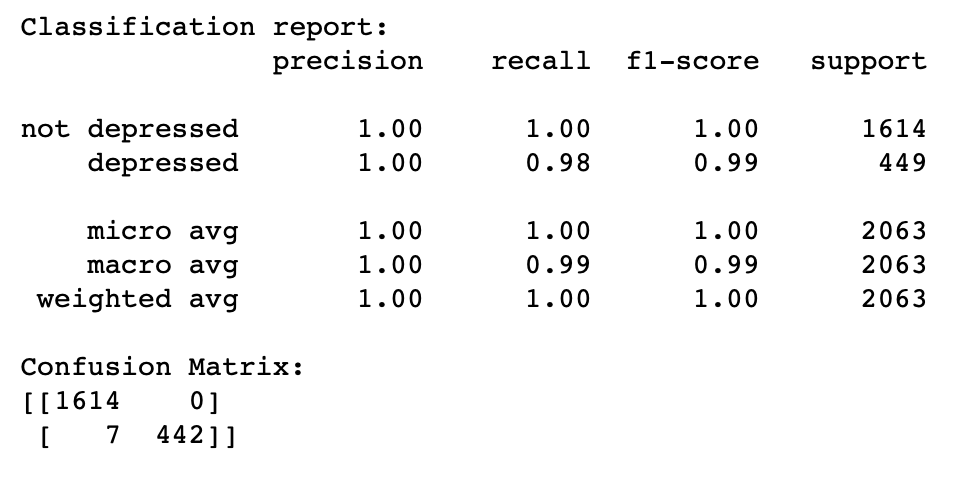


Figure Classification report for Decision Tree

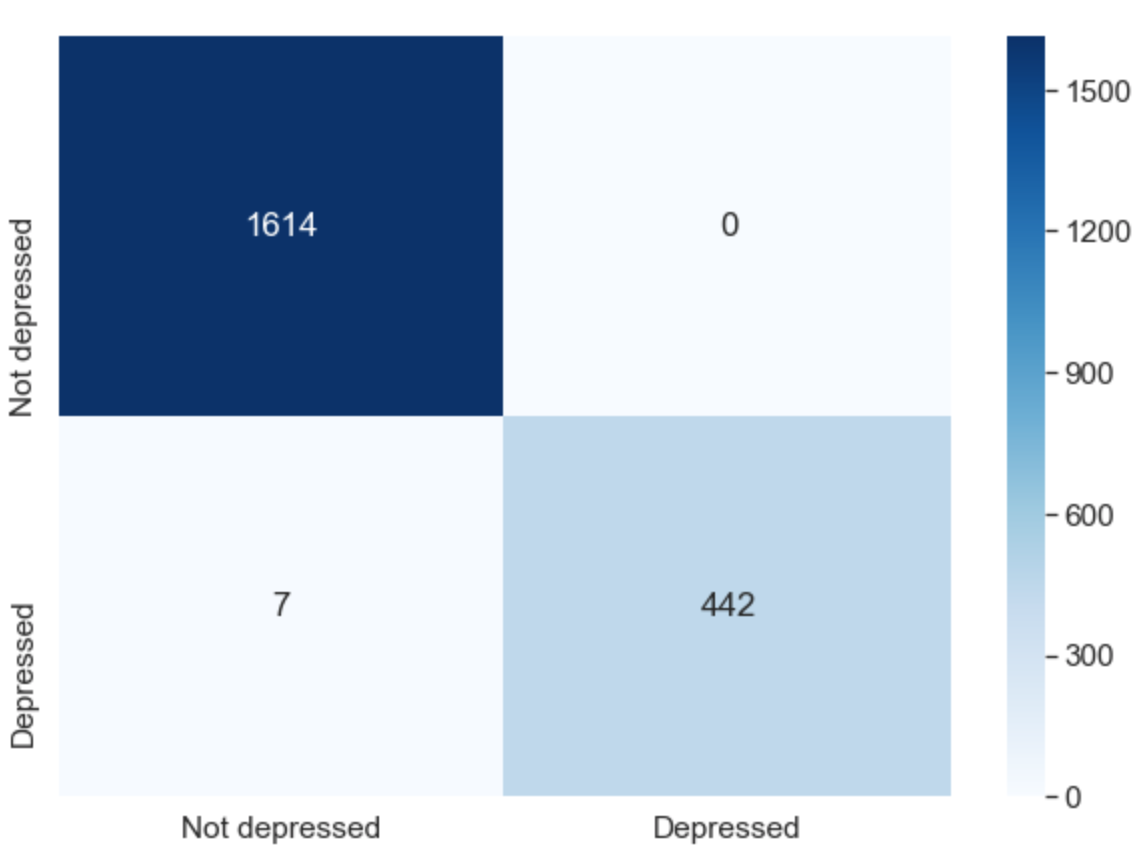


Figure Confusion Matrix for Decision Tree

# Predictions on the tweets

# (Decision Tree Classifier)



Figure Predictions for Decision Tree

# Classification report and confusion matrix for SVM

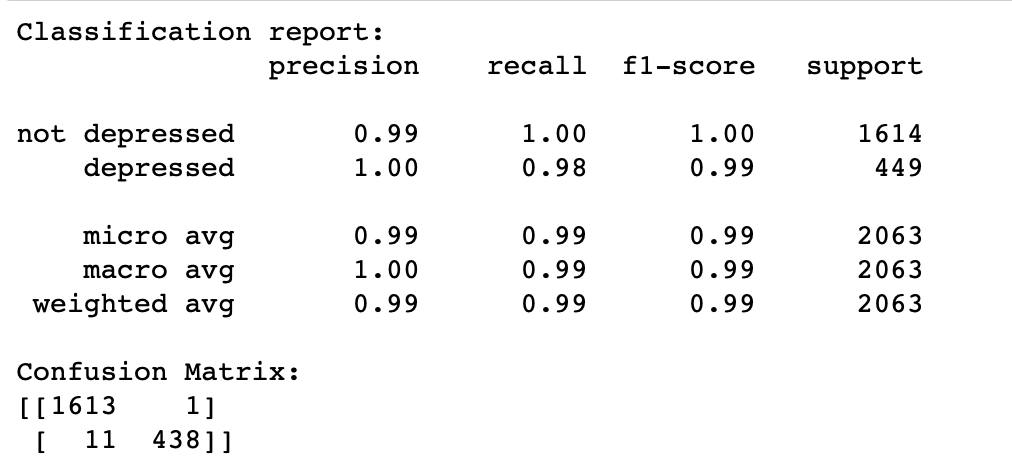


Figure Classification report for SVM

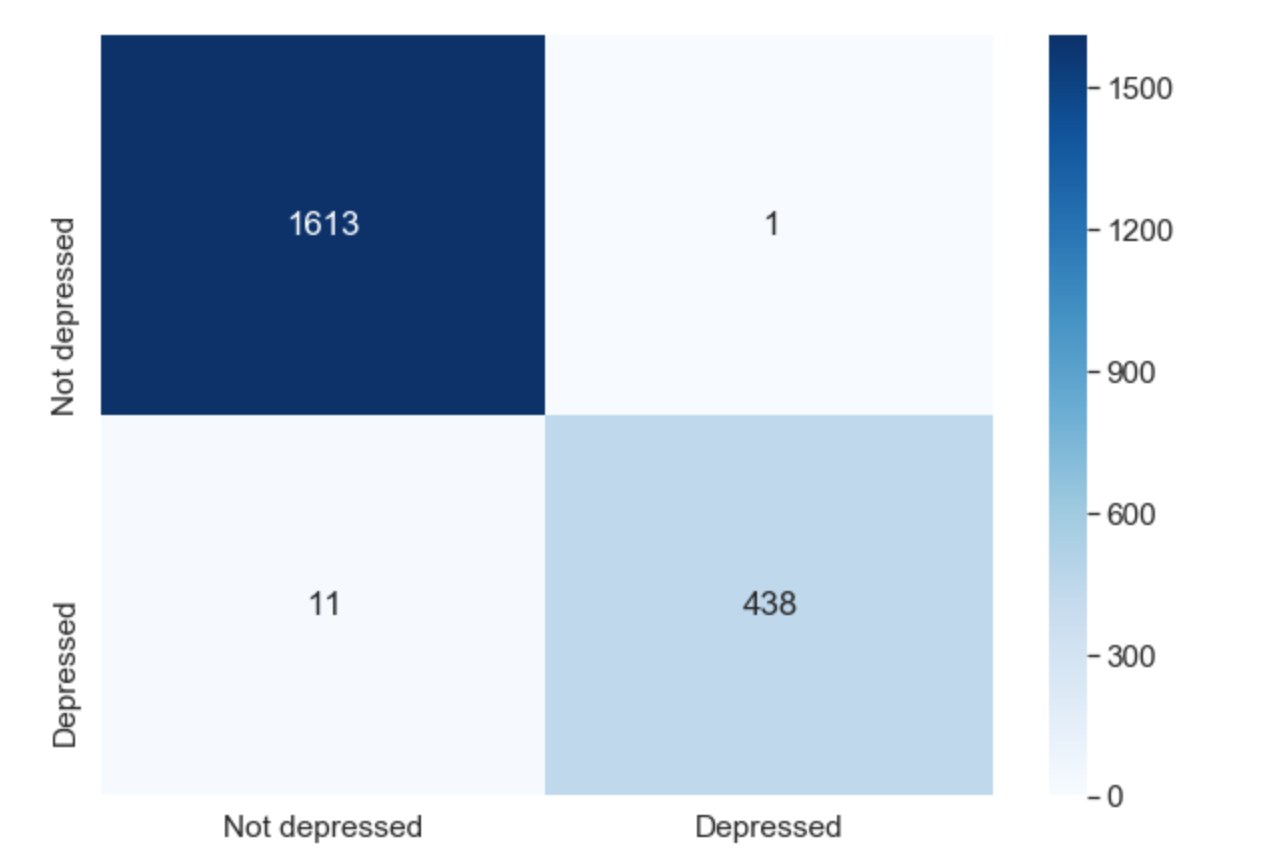


Figure Confusion Matrix for SVM

# Predictions on the Tweets (SVM)

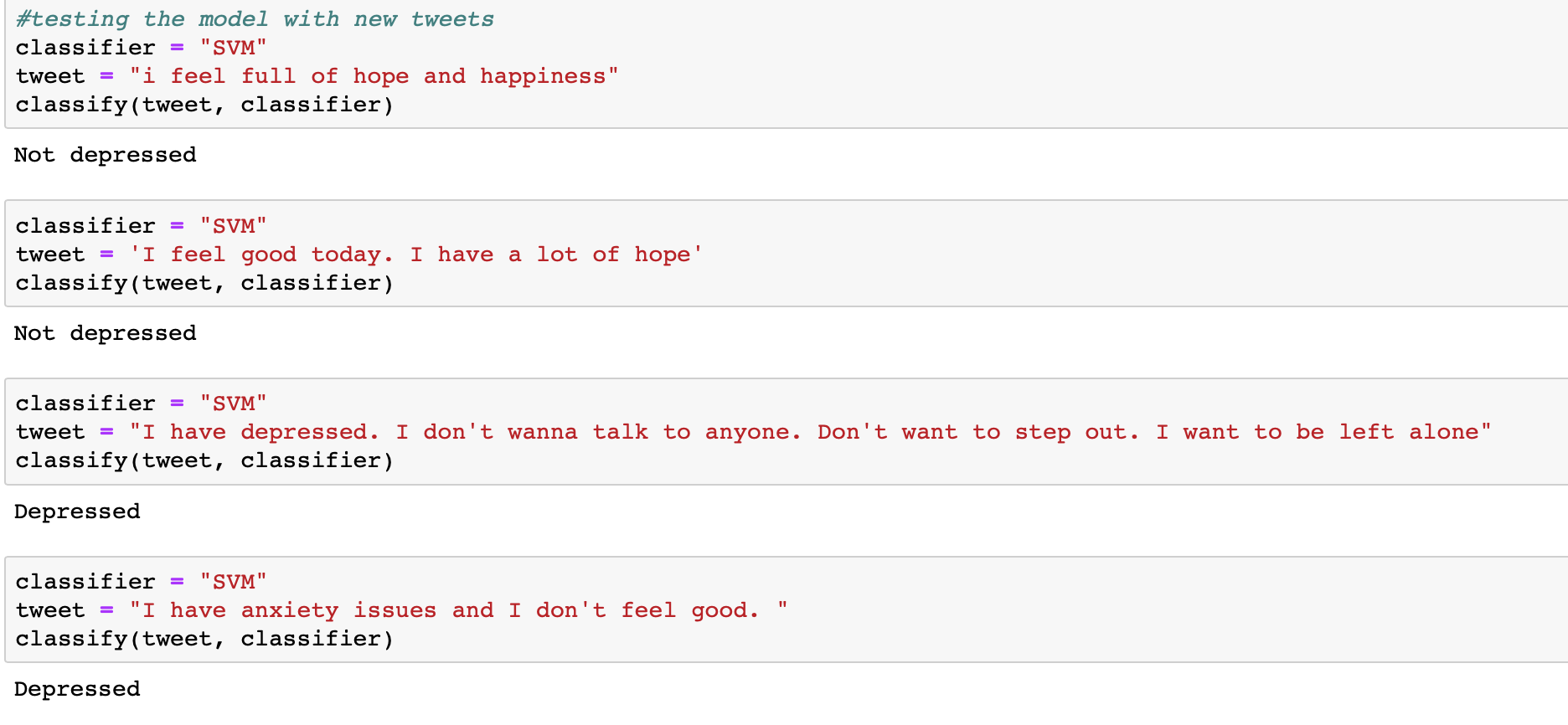


Figure Prediction using SVM

# classification report and confusion matrix for Naïve Bayes Classifier

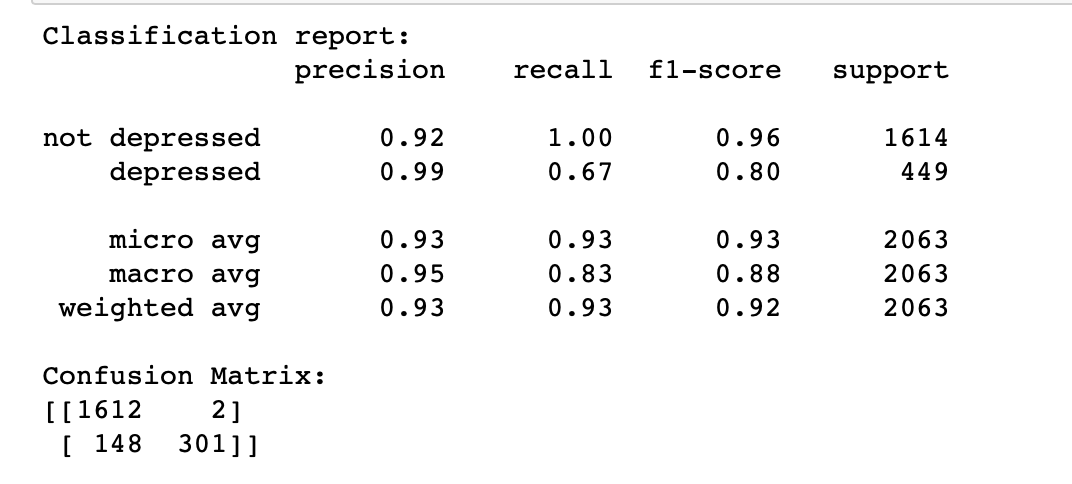


Figure Classification Report for Naive Bayes Classifier

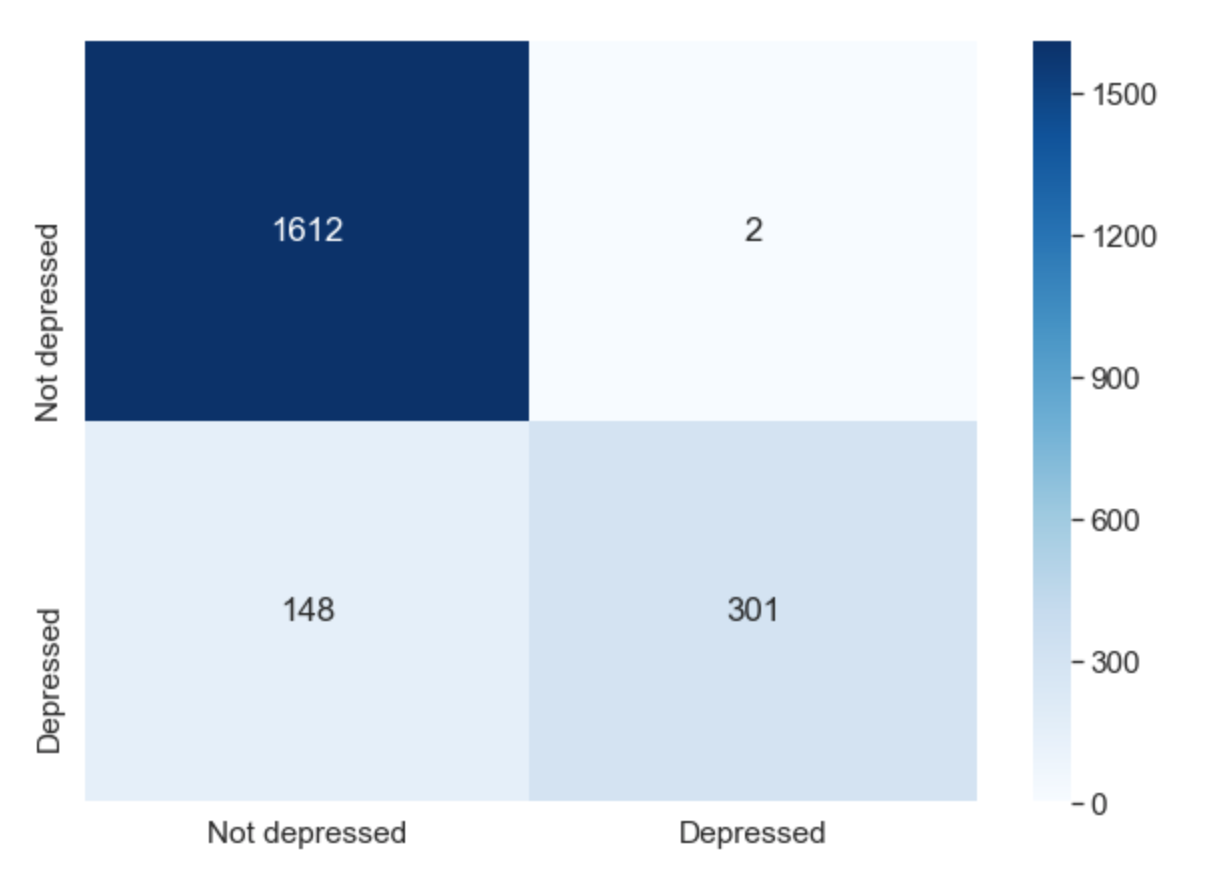


Figure Confusion Matrix for Naive Bayes Classifier

# Predictions on the tweets(Naïve Bayes Classifier)

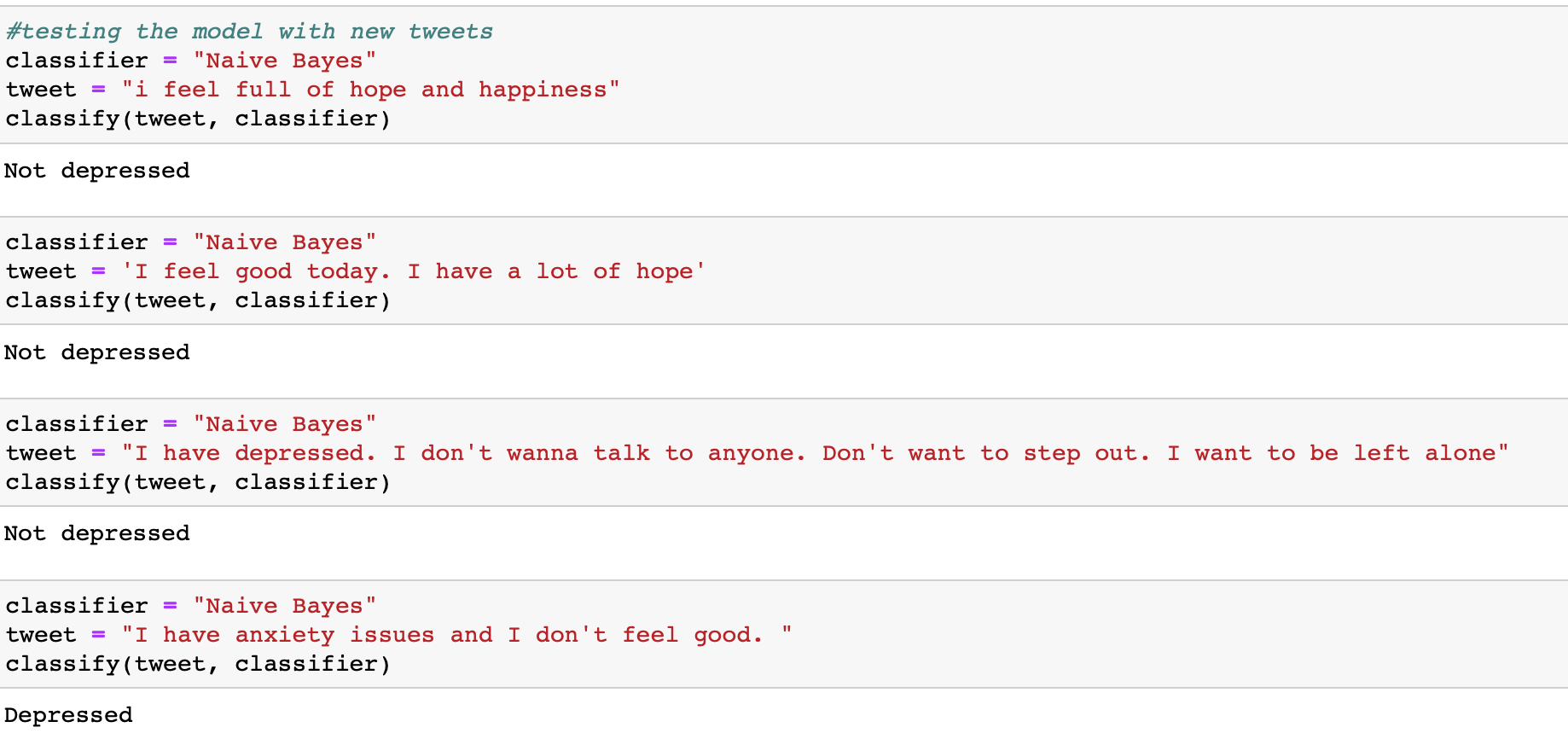


Figure Predictions for Naive Bayes Classifier

# Conclusions and Future Work

As we can see that the Decision Tree Classifier and SVM were able to predict the tweets correctly in part 2 whereas Naïve Bayes Classifier was not able to predict the tweets correctly.

Hence, we can say that Decision Tree Classifier and SVM were best suited for this model

For future work, I will increase the number of depressed tweets in the dataset.