Detecting Depression in Social Media Posts using Machine Learning Models

January 29, 2023

Varun Pininty
California High School, 9870 Broadmoor Dr, San Ramon, CA 94583

1 Abstract:

A big issue that isn't stressed enough for adolescents and emerging adults is depression. Depression can affect you in various ways, whether it be physically, mentally, or socially. Professional and academic performance, personal relationships, and other factors that are important to long-term success are negatively impacted. In trying to help those that need help, we come across a major issue; identifying depression. Social media has become increasingly popular over the years, with many younger generations spending vast amounts of time on it. Detecting depression from a platform that adolescents and emerging adults frequently visit would contribute significantly to helping those suffering from it. In this study, the datasets of tweets were utilized to create multiple machine-learning models that would accurately detect signs of depression. Out of 4 different machine learning models, this research used the Random Forest Classifier model to achieve an accuracy of 96.52%.

Keywords: depression, social media, detecting, machine learning

2 Introduction:

The key to any society is the youth that grow and contribute to expanding and building upon the previous generation's effort. With 90% of emerging adults (18-26 years old) using social media and around 24% of adolescents (13-17 years old) using it "almost constantly", using social media to track signs of depression becomes crucial [1]. WIth nearly of all cases of depression being undiagnosed, the first step to help our youth to receive help is to identify who is suffering (William et al., 2017). I wanted to develop a machine learning model that can accurately detect signs of depression through social media posts to identify those suffering from depression and help them to receive help.

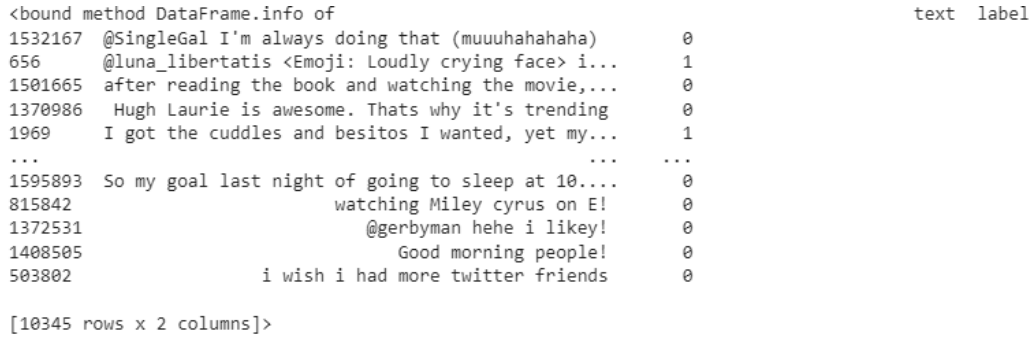
3 Methodology:

3.1 Data Acquisition:

In order to train our models to detect depression, we used two datasets of tweets. The first dataset is called Sentiment140 which contains 1.6 million tweets. The Sentiment140 dataset has columns of info about the date published, the id of the post, the user who posted, and the context or text of the tweet. The second dataset we used is a dataset that uses twint, a tool for scraping tweets, in which we have tweets that have depressive content. It only contains the id of the post and the text of the tweet.

3.2 Data Cleaning

While the Sentiment140 dataset has various pieces of information regarding its tweets, we only need the actual content or the text of the tweet. This means we can simply drop the other information and only retain the text. We will also repeat this process with our dataset from twint. However, due to the large sample of tweets from Sentiment140 and the substantially less sample size from our second dataset, we will take a small random sample from Sentiment140 of 8000 tweets. We will add the label 0 to all the tweets from the small random sample from Sentiment140 to represent that the tweet is not depressive in content and add the label 1 to the tweets from the other dataset to represent that the tweet is depressive in content. We then combine these two datasets into a singular dataset and shuffle them which is represented in Figure 1. As we can see, we have a dataset of over 10,000 tweets where you have the text of the tweet and the label.



```
<bound method DataFrame.info of                                                  text  label
1532167  @SingleGal I'm always doing that (muuuhahahaha)       0
656      @luna_libertatis <Emoji: Loudly crying face> i...     1
1501665  after reading the book and watching the movie,...     0
1370986   Hugh Laurie is awesome. Thats why it's trending      0
1969     I got the cuddles and besitos I wanted, yet my...     1
...                                                  ...    ...
1595893  So my goal last night of going to sleep at 10....     0
815842                        watching Miley cyrus on E!       0
1372531                          @gerbyman hehe i likey!       0
1408505                             Good morning people!       0
503802                 i wish i had more twitter friends       0

[10345 rows x 2 columns]>
```

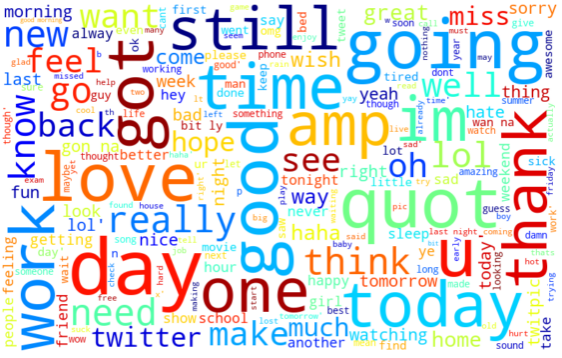
Figure 1: Info of the dataset that we created from manipulating our original two datasets

In Figure 1, we can see that while we do have the text from the tweet, they are not in proper English, with various symbols, bad grammar, and incorrect spelling. In order to train these models, we need to have proper, legible sentences. In order to do this, I defined a function that cleaned each tweet by having all letters to be lowercase, expanding english contractions (Ex: it's = it is), and removing urls, hashtags, emojis and more through the help of a package called preprocessor. By

cleaning our dataset we can ensure proper testing and coding by using proper and known english words compared to the jargon previously used in the tweets.

3.3 Analyzing words through Word Cloud

Using Word Clouds, we are able to see the keywords or the words that show up the most using the given input. In Figure 2, we can see the keywords from the sample we obtained from the Sentiment140 dataset which represents random tweets. We can see words like day, love, work, and thank being prominent. In Figure 3, we can see the keywords from the dataset that uses tweets scraped by twint. We can see that words like depression, anxiety, help, and mental health, a big difference from the words in Figure 2.



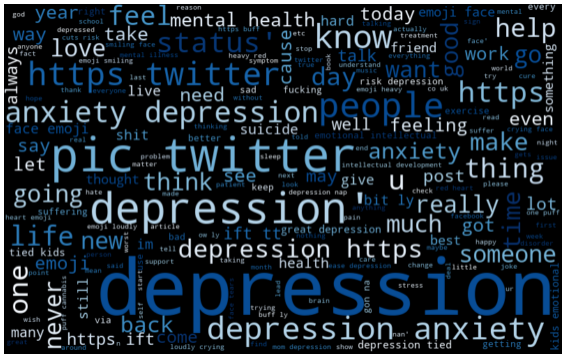


Figure 2 (On the left): Word Cloud of keywords from the sample from the Sentiment140 dataset.
Figure 3 (On the right): Word Cloud of keywords from the tweets scraped by twint.

3.4 Tokenization

In order to help train and test our models, it's important for our dataset to go through the process of tokenization. Tokenization is a form of segmentation that identifies boundaries separating semantic units, for example words, dates, numbers and symbols, within a text [2]). It's hard for our models to simply take large pieces of letters and words and be able to identify which words or letters are important. By breaking our text into groups with assigned indexes called tokens, models are able to take in data in a much more acceptable data type and more easily compute what is important in detecting signs of depression.

3.5 Creating Machine Learning Models

The first thing to do to create a machine learning model is to split the dataset into two parts, a training set and a testing set. We will use the train-test-split from Scikit-Learn to train the model

on 80% of the data and to test the models on 20% of the data. After doing this, we will create 4 different machine learning models to choose the best algorithm for finding the accuracy. We will use Logistic Regression, SVM, Decision Tree Classifier, and Random Forest Classifier.

3.6 Results:

```
[[1593   20]
 [ 334  122]]
              precision    recall  f1-score   support

           0       0.83      0.99      0.90      1613
           1       0.86      0.27      0.41       456

    accuracy                           0.83      2069
   macro avg       0.84      0.63      0.65      2069
weighted avg       0.83      0.83      0.79      2069

Accuracy of logistic regression classifier on test set: 0.8289
```

```
[[1585   28]
 [ 320  136]]
              precision    recall  f1-score   support

           0       0.83      0.98      0.90      1613
           1       0.83      0.30      0.44       456

    accuracy                           0.83      2069
   macro avg       0.83      0.64      0.67      2069
weighted avg       0.83      0.83      0.80      2069

Accuracy of SVM classifier on test set: 0.8318
```

```
[[1571   42]
 [  46  410]]
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      1613
           1       0.91      0.90      0.90       456

    accuracy                           0.96      2069
   macro avg       0.94      0.94      0.94      2069
weighted avg       0.96      0.96      0.96      2069

Accuracy of decision tree classifier on test set: 0.9575
```

```
[[1586   27]
 [  45  411]]
              precision    recall  f1-score   support

           0       0.97      0.98      0.98      1613
           1       0.94      0.90      0.92       456

    accuracy                           0.97      2069
   macro avg       0.96      0.94      0.95      2069
weighted avg       0.96      0.97      0.96      2069

Accuracy of random forest classifier on test set: 0.9652
```

Figure 4: Accuracy of four different machine learning models

	Predicted Negative	Predicted Positive
Actual Negative	a	b
Actual Positive	c	d

Figure 5: The confusion matrix of a two-class classification problem

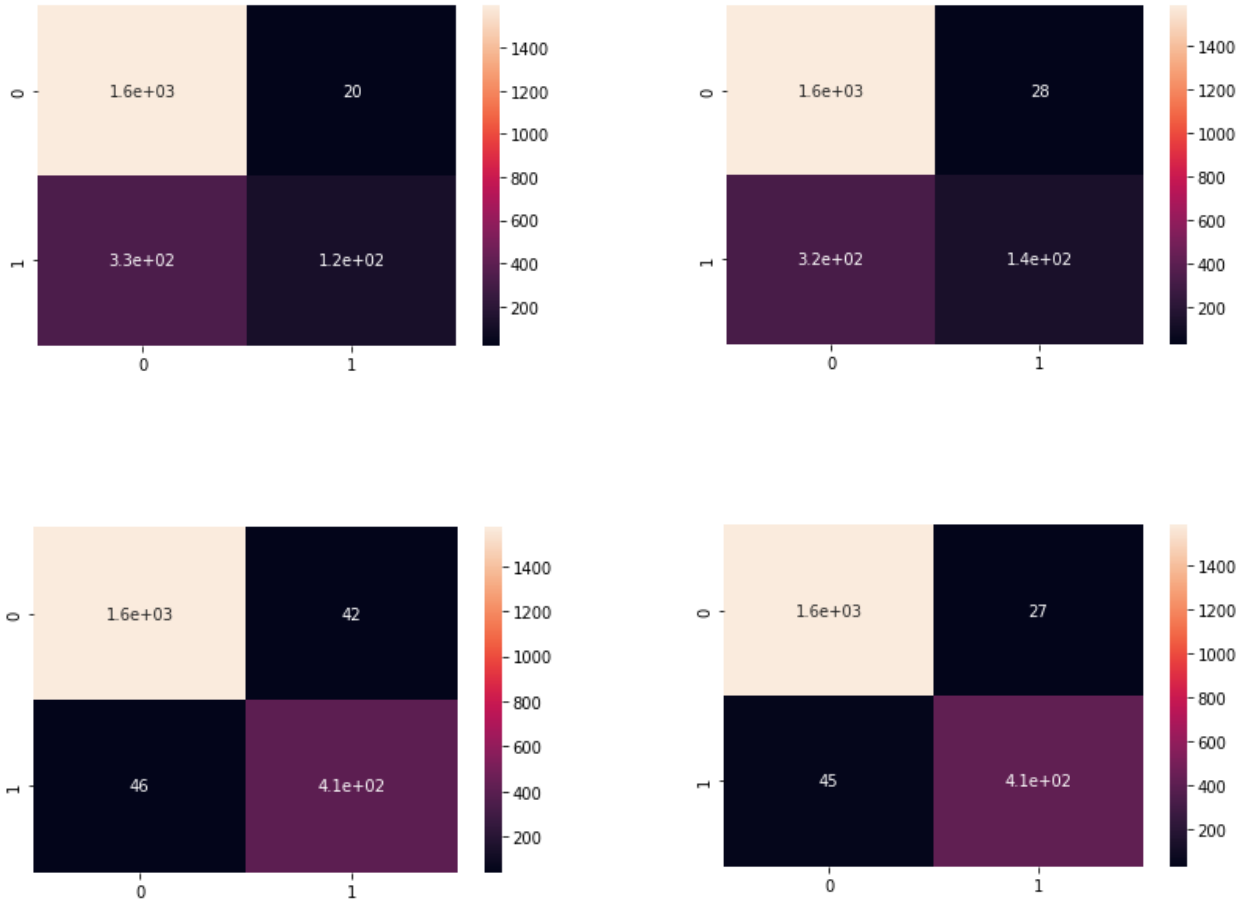


Figure 6: Confusion Matrix of each Machine Learning model.

As seen with Figure 4; All of the models had over a 80% accuracy with Random Forest Classifier being the most accurate with 96.52% accuracy and Logistic Regression Classifier being the least accurate at 82.89%. The SVM Classifier had an 83.18% accuracy and the Decision Tree Classifier had an 95.75% accuracy. To find out the specifics of what our machine models were predicting, we can use confusion matrices. With Figure 5; we can see how to read a confusion matrix with a being the number of correct negative predictions, b is the number of incorrect positive predictions, c being the number of incorrect negative predictions, and d being the number of correct positive predictions [3]. In our case, positive predictions means that our model detected signs of depression and negative predictions means that the models did not detect signs of depression. With our various models, we can see that our Decision Tree and Random Forest Classifiers overall did a good job while our Logistic Regression and SVM models had issues with predicting that a tweet did not have signs of depression when in actuality it did.

4 Conclusion:

The machine learning models, especially the Random Forest and Decision Tree Classifiers, were able to achieve relatively high accuracy rates. However, more improvements to these models are necessary if these models are to truly be useful in detecting signs of depressions in social media posts. There are many routes of improvements such as using tweets from other languages. We could also try other machine learning models such as Naive-Bayes or using the LTSM model. The biggest need for improvement of these models however is simply more data. If this model wants to be feasible for the general population, the model needs to be able to train on more data and have more variety of data in order to avoid overfitting the model. With over 3 billion people using social media, the more data the model trains on, the more likely it is to go through billions of unique tweets from various of different people and find those suffering from depression [4]. Using the basis of the models created, with enough improvements, we can hopefully be able to successfully detect signs of depression on social media and help those that need help.

References

[1] Scott C et al. In: (2017).

[2] Wrenn J, Stetson P, and Johnson S. "An unsupervised machine learning approach to segmentation of clinician-entered free text". In: *AMIA ... Annual Symposium proceedings. AMIA Symposium* (2007), pp. 811–815.

[3] Visa Sofia et al. "Confusion Matrix-based Feature Selection". In: *CEUR Workshop Proceedings. 710. 120-127* (2011).

[4] Karim F et al. "Social Media Use and Its Connection to Mental Health: A Systematic Review". In: *Cureus* 12 (6 2020). DOI: 10.7759/cureus.8627.