Coronavirus Tweets

Wynne Zhang

Sapiesi Tupou   
Jessica Kwon

Leonard Paul Kamara

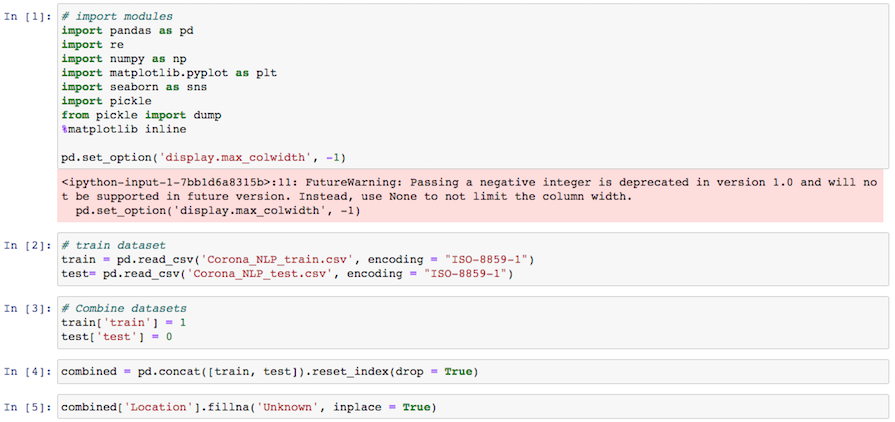
Dilafruz Yunusova

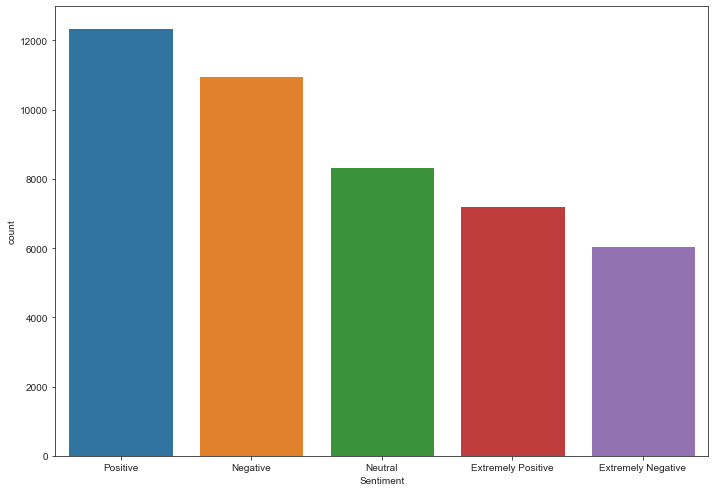
# **Introduction**

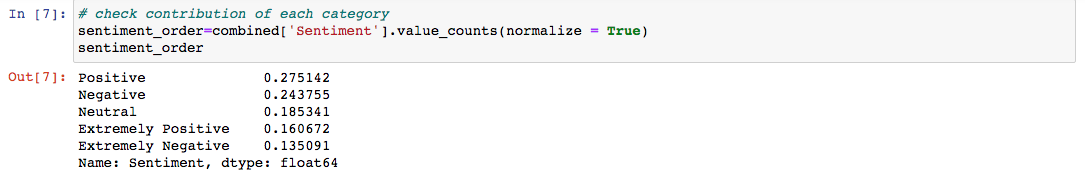
Upon learning machine learning and the awesome tools it can provide, our group set out to see if we can use machine learning to predict a person’s state of mind in terms of depression and suicide and create awareness amongst those close to them. We believe social media provides an incredible source of material for analysis. People use social media not only to share information but also to share their feelings.

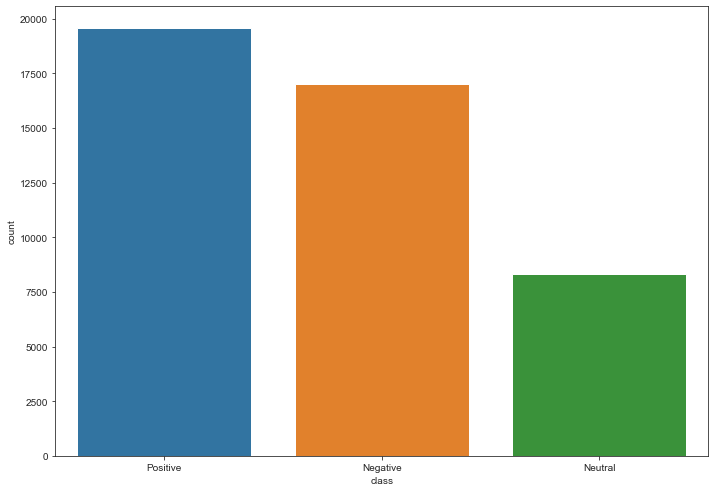
We know that over the last year, Coronavirus and the resulting quarantine has contributed greatly to triggering mental health conditions and exacerbating existing ones. As Coronavirus affected our lives, social media platforms, primarily Twitter, were loaded with posts about this topic and how people felt about it. News sources also utilized Twitter to update the general public with current events. We decided to use COVID-related tweets to run our model through and see what kind of results we can get.

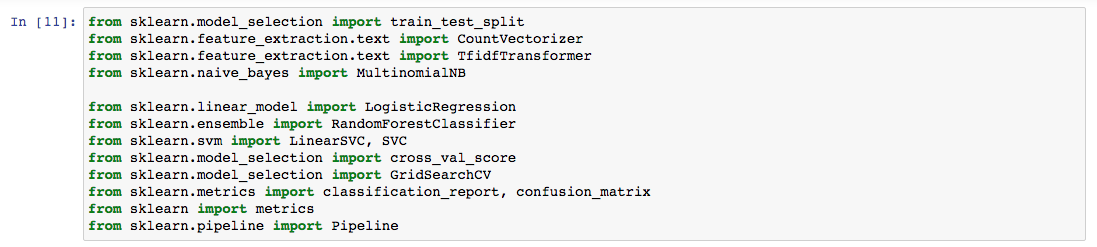
Through Kaggle, we were able to obtain an ample set of COVID-related tweets as a data source for our project. Jupyter Notebook was used for clean up and sorting purposes. Seabron and matplot were used to identify multiple categories of sentiments. Initially we wanted to use five categories: extremely positive, positive, extremely negative, negative, and neutral. Ultimately we decided to classify tweets into three categories instead: positive, negative, and neutral.







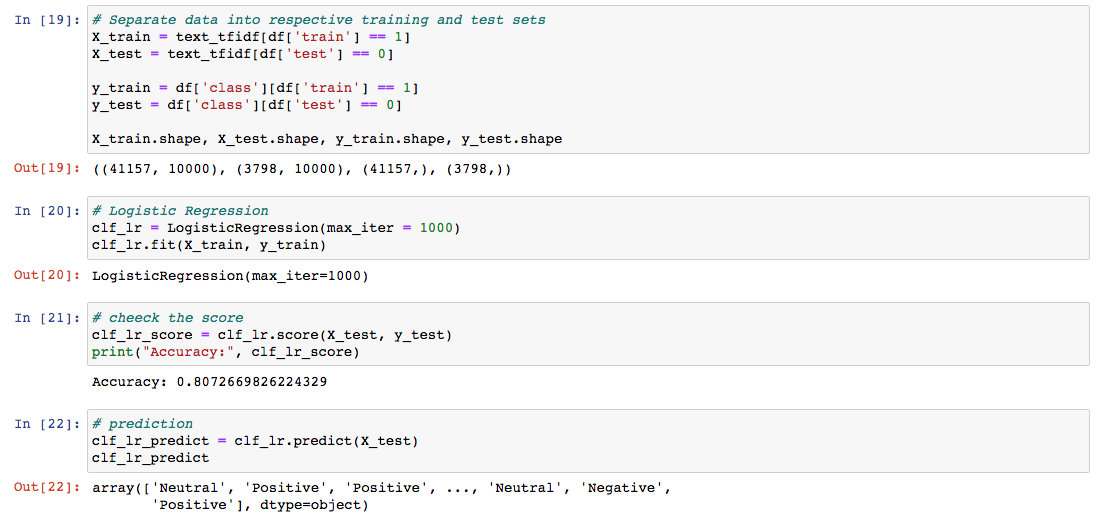




We selected Scikit-learn as our primary machine learning library due to its simple and effective nature. With Scikit, we began text processing. Next we tokenized the text. To transform the data, we used TFIDF Transformer.

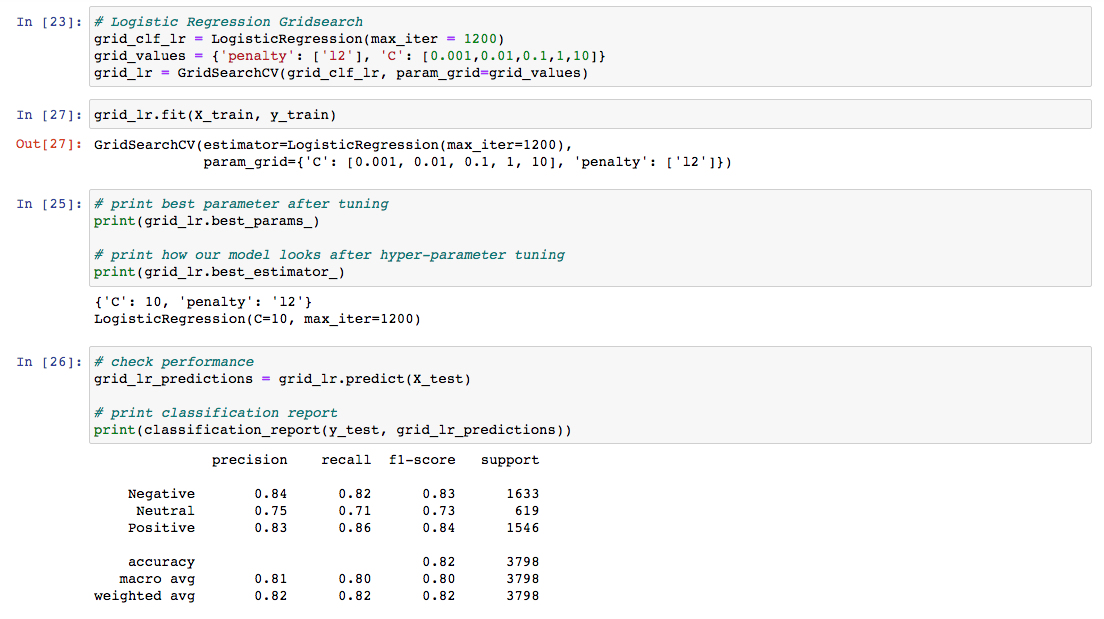
# We began by evaluating our machine learning model using the train-test split, and then using the following classification models to do predictions:

* Logistic regression
* Linear SVC
* Naive Bayes
* Random Forest Classifier



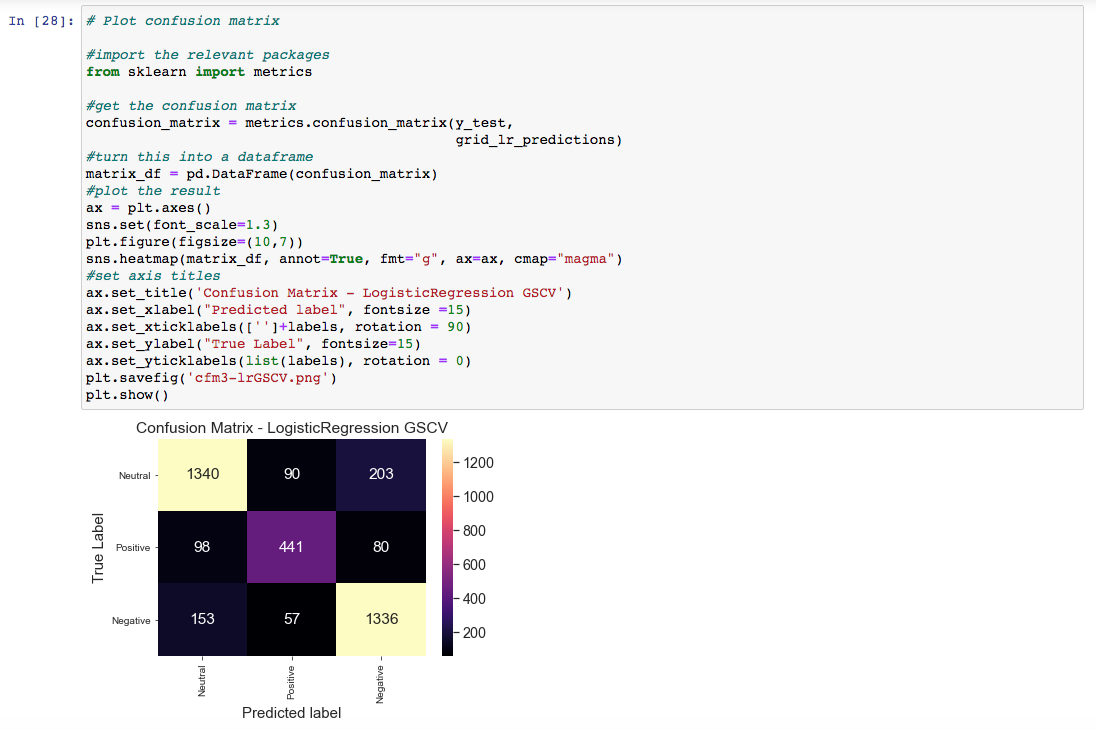
## Logistic Regression

The logistic regression model provided an accuracy of 81%. We did GridsearchCv to see if we can improve the accuracy. Accuracy increased to 82%, and best parameter settings that gave the best results on the hold out data were ‘C’:10, ‘penalty’ :12.



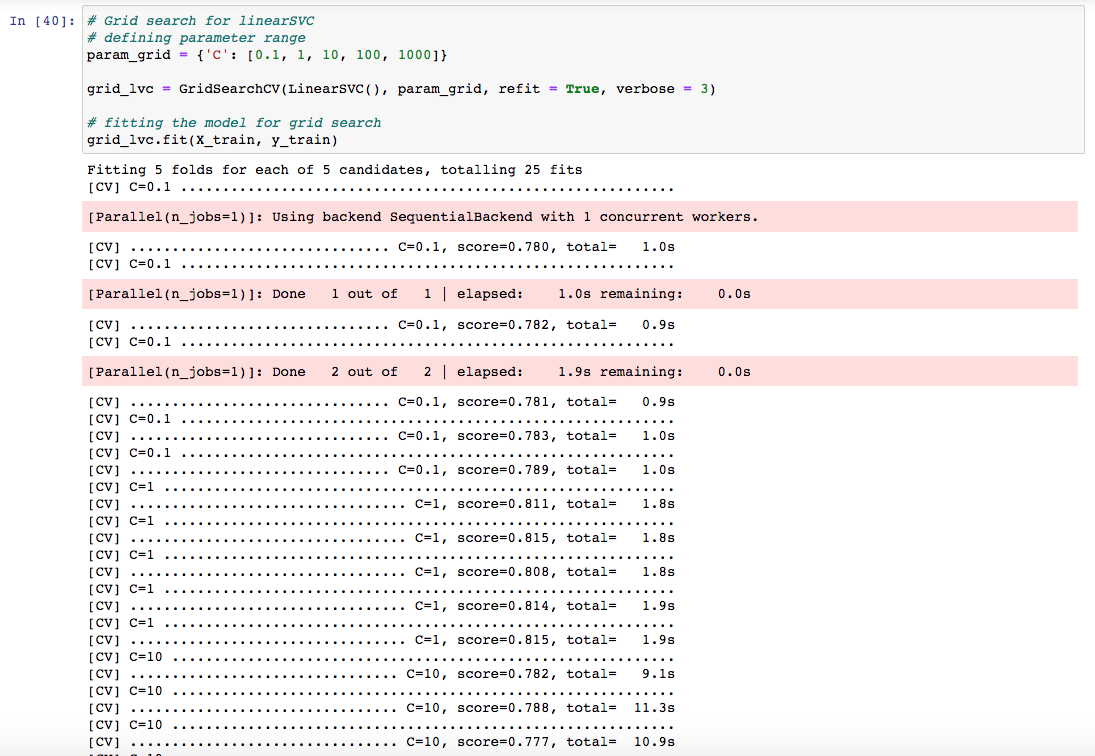
We also plotted confusion matrices that performed logistic regression with GridSearchCV and below shown results were observed.

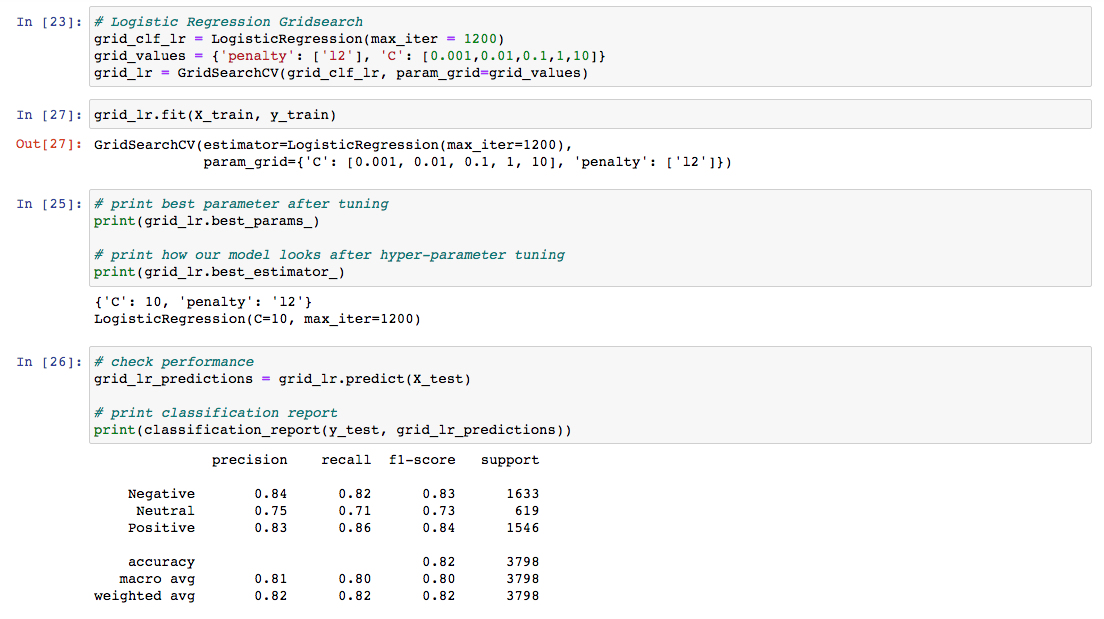
Most of the class data points(1340 Neutral/441 Positive/1346 Negative) were correctly classified by the model.



## Linear SVC

In our initial run, we obtained an accuracy percentage of 83%. After applying GridsearchCV, results did not improve noticeably, but accuracy is still higher than the logistic regression model.

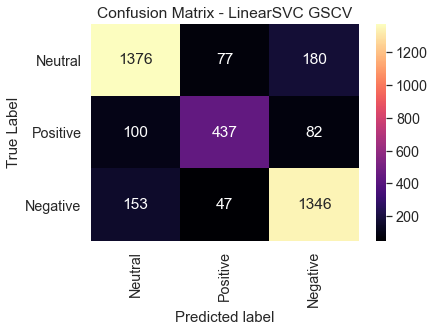




Next, we plotted confusion matrices:

We also plotted confusion matrices that performed linear SVC with GridSearchCV and below shown results were observed.

Most of the class data points(1376 Neutral/437 Positive/1346 Negative) were correctly classified by the model.

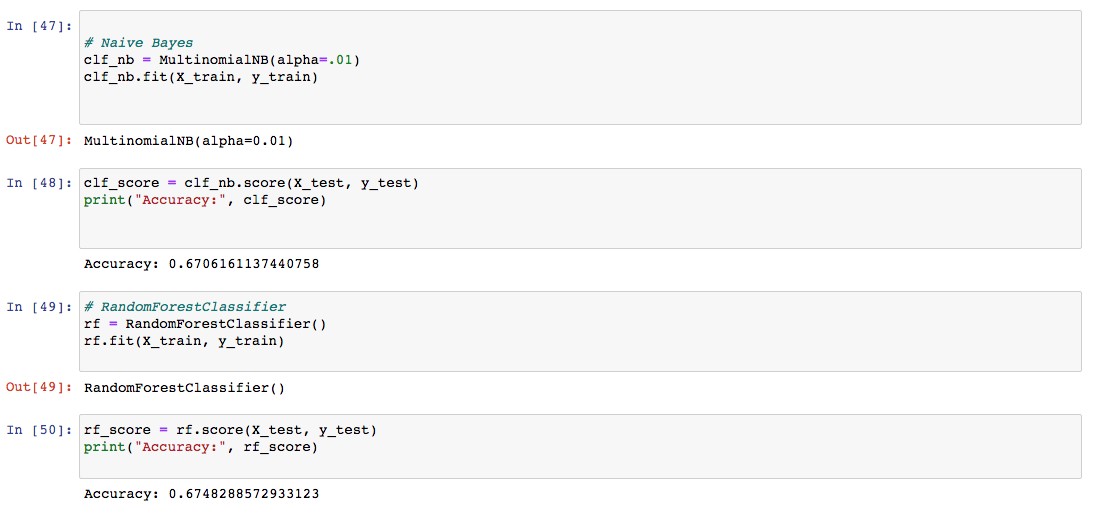


## Naive Bayes

This model resulted in an accuracy of 67%. Since the result is so low, we deemed it unnecessary to go further with the model and we did not use GridSearchCV.

## Random Forest Classifier

This model also resulted in an accuracy of 67%, so we did not choose to do further examination and prediction with this model.



# Choosing Our Model

We selected our second model LinearSVC as it yielded the highest accuracy. We then saved CountVectorizer and TFID Transformer objects into a pickle file. Also, saved a LinearSVC model with best parameters from the GridSearchCV.

The model is deployed on the web for immediate use. We used Flask to render the result of submitted tweets. This will enable us to make real time predictions.

Additional Dataset - interactive chart

We extracted more data to test our model. Due to the large size of datasets, we decided to use about 0.5% of the new data which still had about 360000 rows. Thisdata together with train and test data were loaded in a postgresql database. The results will be presented in an interactive chart found in the web app.

# 

# 

# 

# Challenges

Undeniably, an accuracy of 83% is a great achievement. However, 17% of errors should also be scrutinized. Our selected model can still give us false positive or false negative results. While testing the model, we came across some foreseen errors in our prediction.

For instance, the tweet says: “Positive test result of coronavirus increasing by 300 daily.” In reality this is a tweet with negative sentiment, however because of the word “*positive*” in the written context, the model predicts it as a positive tweet. In the future, the model can be improved and errors minimized. The surrounding context must be better incorporated in order for the model to be effective and improvement to be made for the classification process.

# Conclusion

We successfully predicted the sentiments of COVID-19 related tweets with 83% accuracy. Our trained model was able to ascertain whether or not a randomly selected tweet had a positive, negative, or neutral sentiment. While we are unable to directly determine whether or not an individual is distressed based solely on our analysis of their tweets, we believe we are taking a step towards more complex predictions.

Use case: analyzing tweets to detect mental health issues / depression.

Data Source:

<https://www.kaggle.com/datatattle/covid-19-nlp-text-classification>

<https://www.kaggle.com/smid80/coronavirus-covid19-tweets-late-april>