**COVID Sentiment through Tweets**

****

**Emilio Ramos Monzalvo**

**IST 664 Natural Language Processing**

**Professor Wallace**

**Contents**

**1 Introduction……………………………………………….……….………………………………..………3**

**2 Research Question…………………………………….……….…………………………………...……3**

**3 Analysis and Visualization……………………………………………………………….……….……3**

**3.1 Per Sentiment………………………………………………………………….………….….4**

**3.2 Per Location……………………………………………………………………..…………….4**

**3.3 Month/Date…………………………………………………………………………………...5**

**4 Preprocess…………………………………………………………………………………...................6**

**4.1 Clean Up……………………………………………………..………….………..…………...6**

**4.1.1 Remove Stopwords and Punctuation…..……………………….….6**

**4.1.2 Remove Hashtags, Mentions, and Websites…..…………….….7**

**4.2 Tokenizer………………………………………………………………………..………...….7**

**4.3 Padding………………………………………………………………………………………...8**

**4.3. Encoding Sentiments……………………….………………….……………….…...…9**

**5 Modeling………………………………………………………………………………………………..…..9**

**5.1 SKLearn Random Forest………………………………………………………………..9**

**5.2 Deep Learning with Tensorflow………………………………………….…..…….9**

**6 Prediction Results……………………………………………………………………….……….…...10**

**7 Conclusion……………………………………………………………………………………….……….12**

**7 References..……………………………………………………………………………………………...13**

**1 Introduction**

The year 2020 has brought on new stressors into people’s everyday life. This includes the United States being on the verge of war with Iran, electing a new president, and living in the middle of the Covid-19 Pandemic. Covid, especially, has brought on the most challenges of all with the introduction of shelter-in-place mandates throughout the whole world. Individuals have had to adapt to different hardships like working from home, losing income, and communicating online.

Social media has therefore been a gateway for people to keep in touch with the outside world while staying home. Companies like Facebook, Twitter and Tik Tok have been an outlet to express one’s opinion on their experiences throughout the pandemic. With so much information about Covid circulating online since January from this year, people’s opinion can be derived from their Facebook posts and tweets. Unfortunately, with the massive amount of data stored online, there needs to be a different solution to acquiring people’s opinion rather than individually scrolling through user’s feeds.

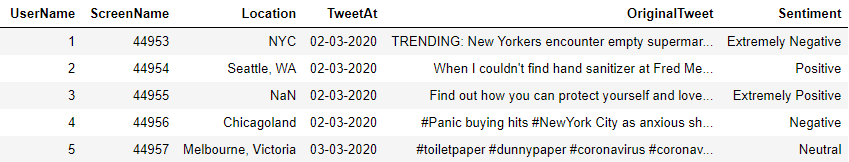
**2 Research Question**

In a perfect world, politician’s actions are driven by their constituents’ opinion. Especially in a pandemic, it is necessary for politicians to listen to how the quarantine and shelter-in-place are affecting the people they represent. With Natural Language Processing, the web can be made into a feeding-frenzy for people’s opinion on a much larger scale. There is no need to take a subset of data because a simple algorithm can categorize the whole data in a near immediate manner. Therefore, the question arises to see if it is possible to extract people’s opinion on the pandemic from the web, but more specifically Twitter.

**3 Analysis and Visualization**

The dataset we set to analyze comes from a Kaggle competition called “Coronavirus tweets NLP - Text Classification.” Each tweet is already pre-classified, i.e., it already has labels attached to it. In this case, the labels range from extremely negative to extremely positive with three other labels in between. The data has already been separated into a training and testing set containing 41,157 and 3,798 tweets correspondingly. Each tweet also comes with four other columns: User Name, Screen Name, Location, and Tweet At which can be seen below.

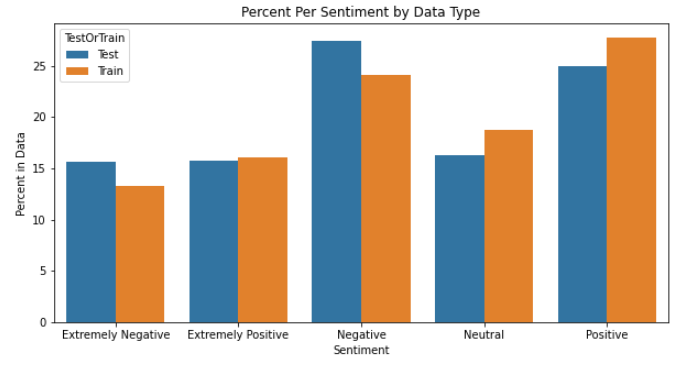
**Table1: Sample Data Table**



**3.1 Sentiment**

Out of the six columns, the Sentiment column contains the most valuable information. As explained previously, the column contains five different categories that fall in a positive to negative spectrum. In order to have an unbiased model, the dataset needs to be distributed evenly among the categories. Below we see that the Negative and Positive categories contain more than half of the data while the rest of the categories are distributed evenly among the remaining half. Since the categories with most of the data are not in the extremes of the spectrum, we should not expect the model to overly prefer a sentiment.

**Figure 1: Percent per Sentiment**



**3.2 Location**

The location of the user can be a significant variable in determining the sentiment towards the coronavirus. Countries like Australia that adapted more strict rules in the beginning of the pandemic were able to contain the number of cases and prevent economic shutdowns from happening in the long term. On the other hand, Italy had one of the worst outbreaks from the beginning of the pandemic which led to extreme caution being taken to prevent the spread. This geographical input can help improve the accuracy of the model. Unfortunately, the dataset contains more than 13,000 unique location names that range from cities like Los Angeles to countries like China. Decoding each one and generalizing them into countries or states will take too much effort. Additionally, since the dataset only contains 45,000 tweets, there are too many locations to be able to find a pattern from them. Therefore, the location variable will be taken out of the modeling.

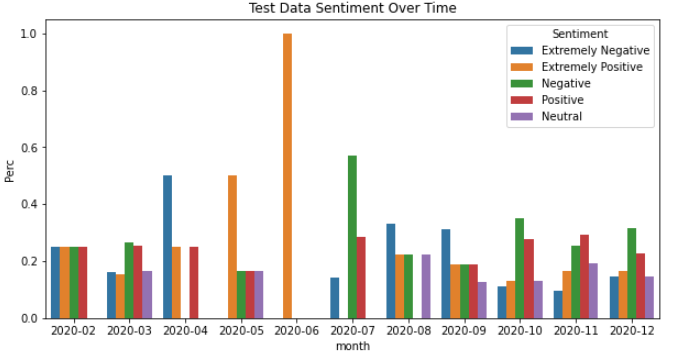
**3.3 Month/Date**

At the start of the pandemic around February or March, there was little to be known about the virus since it mostly resided in other countries from the perspective of the United States. Then, as the virus started to become more of a part of our lives, new opinions and facts arised like the economic hardships that the restaurant industry has experienced throughout the 2020 summer.

Therefore, in order to have an impartial model, the data used to train it has to also represent the data available in the test set. The charts in Figure 2 clearly show that the train dataset is evenly distributed among all the months while the test set does not. In the month of June, the test data only contains only tweets with a negative sentiment. Even if the training dataset creates an unswayed model, the test set might not be truly represented by it. Consequently, the date column will not have much effect on the accuracy of the model, so it will be excluded from the analysis.

**Figure 2: Train vs Test Data Over Time**





**4 Preprocess**

In order to place data into the model, the dataset needs to be translated into a language that an algorithm can interpret. To do this, the dataset needs to be cleaned up, tokenized, and encoded into a shape that can be inputted. This can be considered to be the most important part of a Natural Language Processing model as here we are converting a text into numbers that a machine can understand. Taking out unnecessary words that do not add meaning to a sentence can be crucial for the model to work more efficiently.

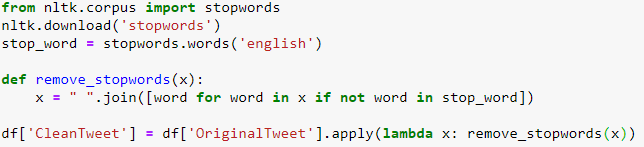
**4.1 Clean Up**

Cleaning up the tweets can be tricky as tweets do not have to be grammatically correct to be understood. The standards created by the English Language are usually not followed, so the preprocessing of the data has to be creative for it to make sense. Initially, stopwords and punctuation can be removed so we can focus on words that contain more connotation and can lead to extracting the correct opinion. Moreover, websites, tags and mentions will not effectively bring useful information as more information is needed on them to be able to understand their implications.

**4.1.1 Remove Stopwords and Punctuation**

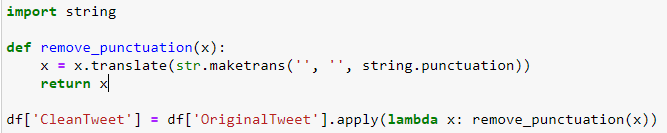
A library full of English stop words can be downloaded from the NLTK python package. Then, we a simple function, all the stop words can be removed by iteration through every tweet.

**Figure 3: Removing Stopwords**

****

Next, the punctuation is removed similarly using the string package available in python.

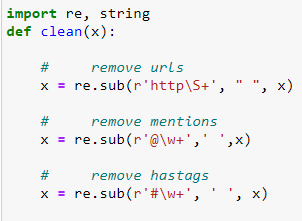
**Figure 4: Removing Punctuation**

****

**4.1.2 Remove Hashtags, Mentions, and Websites**

After removing the stop words and punctuation, the next obvious choice to remove is the unnecessary tokens like the websites and mentions that users add to their tweets. To perform this operation, regular expressions are used which were found in a stackoverflow question. To do this, the library re was used.

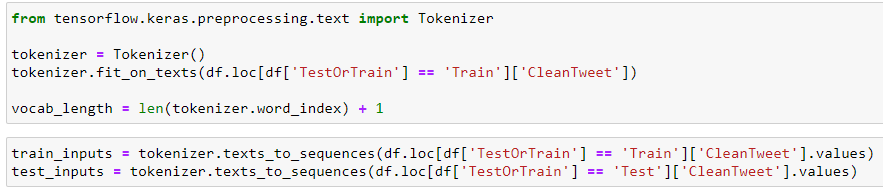
**Figure 5: Removing unrecognisable objects with Regex**

****

**4.2 Tokenizer**

After removing the most of the unusable words in each tweet, the text is still made up of words which a model cannot interpret. With a tokenizer, the words can be translated into a tensorflow friendly input. Traditionally, tweets are tokenized using the Tweet Tokenizer function available in the NLTK package, but for more complicated machine learning models in Natural Language Processing, the Tokenizer in Tensorflow is preferred. To use it, we first import it from Tensorflow and we fit it using the tweets from the training dataset.

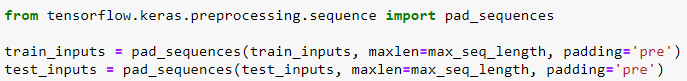
**Figure 6: Tokenizer with Tensorflow**

****

**4.3 Padding**

Finally, the inputted x variable in a model needs to be standardized in length, so a padding function is used to make the tweets of similar length. The padding function comes from Tensorflow where a max length is imputed after calculating it from the longest sentence out of the whole dataset. This is a straightforward calculation since each tweet can only have a certain amount of characters.

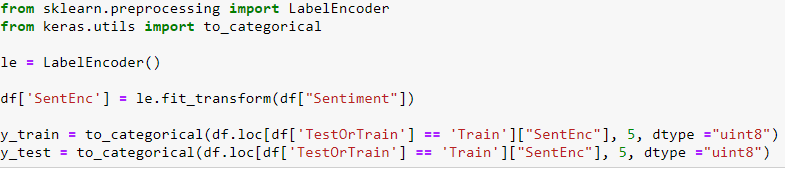
**Figure 7: Padding with Tensorflow**

****

**4.4. Encoding Sentiments**

Finally, since we have five categories for each tweet, the input y variable needs to be encoded for the model to make sense of the five categories. Simply encoding the category name into a number won’t help us, so we use a hot encoder to create five columns with each corresponding to one of the five categories.

**Figure 8: Encoding Y-Variable**

****

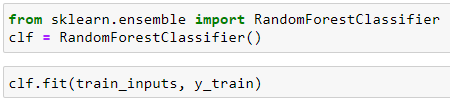
**5 Modeling**

After successfully converting the dataset into a computer-friendly form, the tweets are finally able to be plugged into an algorithm to train and test a model. Before using a deep learning model using Tensorflow, it is conventional to use a simpler model to establish a baseline by using a simple model. In this case a Random Forest model was used.

**5.1 SKLearn Random Forest**

The package SKLearn contains a vast choice of classification models, but the Random Forest model was the choice for this analysis. SKLearn makes it easy to plug-in the data without many transformations necessary. Below Figure 9 shows the code used to train the Random Forest model.

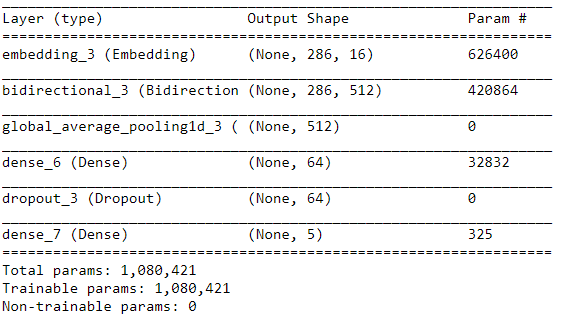
**Figure 9: SKLearn Random Forest Training**

****

**5.2 Deep Learning with Tensorflow**

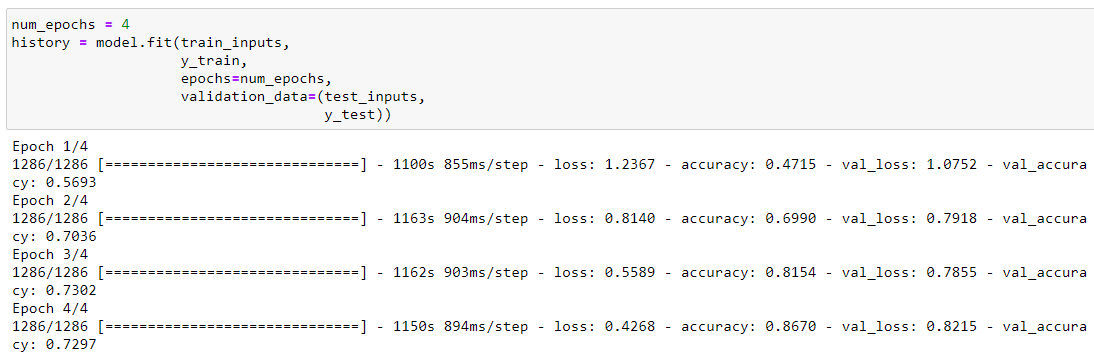
The article “*Natural Language Processing with Tensorflow”* was used to create a Tensorflow deep learning model for Natural Language Processing. The article shows a way to configure the different layers in the Neural Network to effectively train the model. In Figure 10, the Neural Network configuration can be seen. The use of a bidirectional layer is preferred in NLP to understand the forward and backward interaction between words.

**Figure 10: Neural Network Dimensions**

****

The model was trained with only four epochs since there is a limited amount of resources to use locally. Each epoch lasted about 15min which added up to one hour to train the model.

**Figure 11: Tensorflow Model Training**

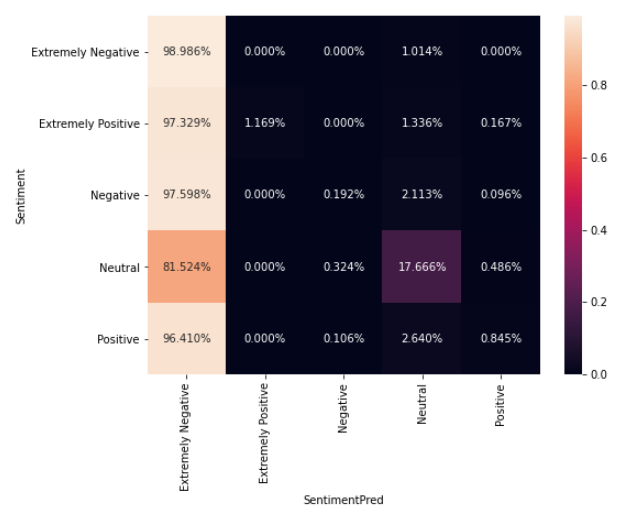
****

**6 Prediction Results**

Once the two models finished training, the testing dataset was plugged into the two models. Both models got a satisfactory training error as the Random Forest model got almost a 100% accuracy while the Neural Network got a 86% accuracy. The Random Forest’s results are troubling as having such a high training accuracy can be a sign of overtraining.

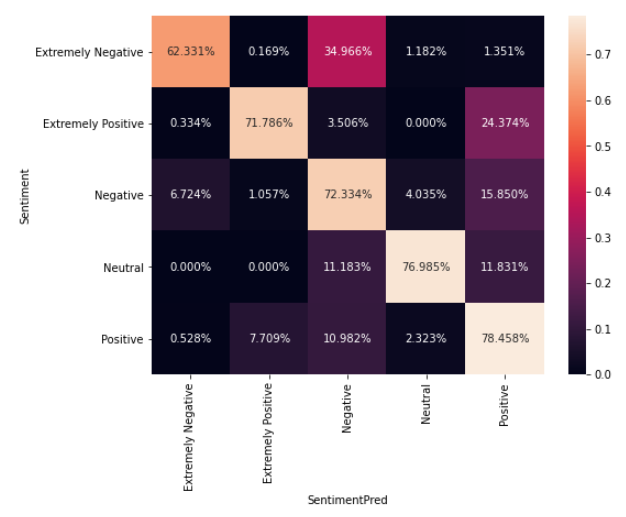
On the other hand, the test data model accuracy shows a different story. The Random Forest model converged into the Extremely Negative sentiment as illustrated in Figure 12. Also every tweet in the test set was classified as an Extremely Negative tweet in regards to Covid. Therefore, it is understood that the model was overtrained and will not give insight into people’s opinion of Covid.

**Figure 12: Test Accuracy for Random Forest Model**



The Tensorflow Neural Network did perform similarly to its training accuracy as it scored a 72% accuracy on the testset tweets. While the accuracy is not perfect, this a positive sign that the opinions can be learned and deduced using tweets. Furthermore, the results show how the model confused about 35% extremely negative tweets as just negative tweets which a human can also confuse depending on the diction used. Similarly, 25% of extremely positive tweets were confused as positive tweets. The model nonetheless seems to understand the difference between a negative and positive tweet. One thing to try later on is to relabel the dataset into positive and negative while excluding extremes to improve the accuracy.

**Figure 13: Test Accuracy for Neural Network Model**



**7 Conclusion**

Understanding people’s opinion on the arguably most jarring event of this century, can prove extremely advantageous. Knowing which industry is suffering the most can help distribute resources more effectively. Then, if gathering the opinions was in question, using tweets is proved to work. With 72% test accuracy and limited resources, the Tensorflow Neural Network shows the potential to get a generalized opinion of people on the subject of the coronavirus.

**8 References**

Covid Twitter Image:

<https://indianexpress.com/article/technology/techook/how-to-use-twitter-effectively-amid-covid19-coronavirus-pandemic-6345438/>

Covid Dataset:

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

TensorFlow Language Modeling:

<https://towardsdatascience.com/natural-language-processing-with-tensorflow-e0a701ef5cef>

Removing website, hashtag and mentions:

<https://stackoverflow.com/questions/13896056/how-to-remove-user-mentions-and-urls-in-a-tweet-string-using-python>