Natalie Humber, nhumber@usc.edu

For this project I decided to do a regression task because it felt more intuitive for classifying tweets than a classification task. Classification tasks work to assign labels to examples in a problem space, where regression tasks aim to predict continuous values within a problem. In the case of classifying tweets as “positive”, “negative”, or “neutral” a regression task (using logistic regression) made the most sense to me. Additionally, due to the fact I am still new to a lot of the material we have been learning in the CAIS++ curriculum, regression also felt more comfortable to me.

I tried and implemented a lot of different data preprocessing techniques. First I used some data visualization techniques to understand the dataset which was provided to us. I found that as described there were 3 rows (valence, author, and tweet) and 1,600,000 columns. As I began to work on the code I noticed that 1,600,000 was a little hard on my computer, so I created a copy of the dataset which only included every 25th tweet. After making the data more manageable, I cleaned the tweets of usernames and links. Then I made sure all of the tweets were in lowercase, and that any non alphanumeric values were replaced with spaces. Next I created a token for the clean tweets and filtered out stop words. After that I stemmed and lemmatized the tweets.

After that I used a logistic regression model to predict the valence of the tweets based on prior observations from the dataset. I thought that this model would be the best for this situation, because tweets are continuously updated, so the model would get better over time. I looked into a few other methods, but I thought logistic regression was the most straight forward. The data was compiled with “traintfidf_lemm” which was created with the help of TfidfVectorizor, a feature of the sklearn library and “datas[‘valence’]” the valence column of datas. These sources were then combined and divided into test and training data for the logistic regression model. The goal of the model is to correctly classify the valence of tweets as positive, negative, or neutral. However, because there were no neutral valence values given in the original dataset, my model was not able to predict neutral tweets.

Feature Names	Domain	Type
Valence	{0,4}	Int
Author	String	String
Tweet	String	String

To evaluate my data I generated a classification report. It turned out that my model was not very great with an overall accuracy of 75.19%.