Sairashmika Jammi

DATA 606 Spring 2020

Deliverable 3

**Twitter Sentiment Analysis on Climate Change**

**Objective Changes**

The more I worked with my data and my code, the more I had to rethink my objectives and goals for my capstone project. My original plan was to analyze tweets relating to climate change dating back 5 years, but once I started working with my code more, I realized how much data that entails. I simply do not have the time nor the equipment to analyze all that data on such a huge level. I decided to create my dataset going back only one month and limit my tweets to climate change only. This has greatly improved the performance of my models since it is not has expansive as before and it is more specific. I also excluded any retweets to reduce redundancy, which also greatly improved performance.

**Dataset**

Once I had my cleaned dataset, I began the process of creating my classifier. Natural language processing is a common way to analyze the sentiment of any text, therefore I thought it would be of great use for my project. I used stop words provided by NLTK and I removed them from my dataset because they will affect the polarity rating of the tweets. Once I did that, I used a package called Textblob to assign them polarity values. This package is used to calculate the sentiment of text, and I used this to calculate the sentiment of the tweets. This package assigned a numerical value to the tweets – the polarity value. If the polarity value is positive, it’s a positive tweet and if it is negative it’s a negative tweet. Once I iterated through my entire dataset and assigned them values, I labeled the tweets as positive or negative accordingly.

**Model Construction**

The majority of this deliverable was spent on constructing my model – and I used a different approach than expected. I started out by creating a matrix to store my tweets, and to do so I used a method called Bag of Words. **Bag of words** is a natural language processing method that converts text into numerical representation and can be created using the CountVectorizer function. CountVectorization “generates a sparse matrix representing all the words in the document” and in this case we are using this function to create a matrix to represent tweets.

Once I created my matrix, I need to create my various algorithms. Using logistic regression model, I created my model. I created training and test sets from my dataset and split it randomly in the middle.

I am in the process of testing my model on my test set. I have run into some hurdles regarding this, and will hope to resolve them in my next deliverable.

**Unfinished**

I am still in the process of implementing another method called Tf-Idf, which is also another method in which to create matrix with weights. This is different to the Bag of Words method, where it creates a numerical value that measures relevancy of the word “by multiplying two metrics: how many times a word appears in a document and the inverse document frequency of the word across a set of documents” [2].

Once I created my TF-IDF model. I will build my predictive model around that as well and compare them to the Bag of Words model and see which ones performs better.

Additionally, I hope to implement the Naïve Bayes classifier as well. I have been doing some research into the scikit-learn machine learning pipeline technique that helps construct this classifier.

**Deliverable 4 Plans**

In my next deliverable, I hope to validate both my models and determine which one is more successful. I will do so by utilizing the build in scikit matrix report that gives the accuracy score for our model. Once I identify which model is more accurate, either the Naïve Bayes or the logistic regression, I will apply that model on the test data. I hope to complete this model, so I can feed tweets real time, and determine the sentiment of those tweets.

**References:**

**[1]:** <https://towardsdatascience.com/twitter-sentiment-analysis-classification-using-nltk-python-fa912578614c>

**[2]:** <https://monkeylearn.com/blog/what-is-tf-idf/>