

Hayden Lingle

CSCI 1310, 2pm Recitation on Tuesday w/Camilla

4 December 2016

Purpose:

My purpose for this final project was to interpret the instructions given and to apply my knowledge of python towards the files and to get a working project.

Procedure:

First, I had to interpret the meaning of all the existing data, existing code, and which parts of those needed to be used by me to complete my project. Interpreting the interaction between the existing python files was the hardest part for me. I did not realize an instance of the state class was called for each state. I also did not initially understand how the state class could reference itself and have 50 instances with data. When I started working on this project, I began to create my own version of what the state.py file already accomplished: I made a list of state lists that had pertinent information from tweets, and then I loaded data from the JSON files into that list. A friend of mine in CSCI 1310 explained to me how state.py worked, and since I already had a working implementation to import and sort important parts of each tweet, I just repurposed that code to send the information to the state.py class.

My program begins by reading a list of JSON file names and looping through them, opening the files and appending the files into a variable. Then I write that variable to a new, blank Json file so that all 4 Json files with timestamps are now inside of one master Json file. Then I have a for loop that reads the data line by line, and if there is no errors in reading, then the line is examined in a nested for loop: if any part of the user's query is in the text element of the tweet, the line is split up using markers for 'text' and 'coordinates'[1] & 'coordinates'[0]. These three important parts of the tweet are sent to the tweet.py class to create a tweet object. This new tweet object is appended to an empty list of tweets that contain words the user gave in the query. Now I have a list of tweet objects that contain tweet text, latitude, and longitude data while all including query terms. I created a for loop to go element by element through this new list. Inside of this for loop, I have a loop that goes state by state and uses existing functions to calculate distance from the tweet's latitude and longitude to the centroid of every state. While this loop

runs, it progressively saves any new, shortest distance and the state associated with that distance. When the loop finishes, my program saves the distance and the state abbreviation of the closest state centroid. After the tweet has been associated with a state, I use the python `.split()` function to split the tweet text word by word. Then I use another loop to go through each word of the new list and check if it is in the huge sentiment reference file; if it is, I append the score of the word to a list of sentiment scores that I have added in the `state.py` file by passing the score to a function I added to the `state.py` file called `addToListOfSentiments()`.

After every tweet including query elements has been associated with a state and all words have been scored for sentiment number and appended to the state's list of sentiment scores, I run a function inside of the `state.py` file for each state that I added called `calculateAverageSentiment()`. This function loops through the list of sentiment scores for each state if there were any values that existed, and adds them. Then it divides that sum by the length of the list to find the average sentiment for each state.

Now that every state that has an average sentiment score, I show the state map then loop through each state and send the `setFillColor` function the state abbreviation and the color associated with the average sentiment. Each time the loop runs, it will update the color of an individual state that had sentiment data.

Results:

When query words are passed and the program runs, a map successfully appears and displays colors associated with each state's average sentiment score. If there was no sentiment scores associated with tweets from a state, or if there were no tweets from that state, my program will not color those states.