**User Generated Content Analytics (Fall 2022)**

**Assignment #2**

**New due date: 10/06/22 by 11:59 p.m. on Canvas**

Note: While this assignment is about politics and the upcoming midterm elections, upon doing this assignment you will obviously see how the overall approach could be applied to a business setting where we analyze what attributes people associate with two competing products or brands, how they feel about such attributes (sentiments), and how the sentiments vary by geography.

1. Collect 4-6k tweets on the any senate race in a battleground state (may I suggest John Fetterman and Dr. Mehmet Oz in Pennsylvania) using **getTweets2022.py**. It is preferable to use general hashtags and search keywords (like PA Senate race 2022) rather than “Fetterman” or “Oz” because of the probability of Fetterman or Oz being close to 1 in the latter cases. Let the occurrences of Fetterman or Oz come naturally in a subset of the tweets you collect.
2. Find **FOUR** key issues mentioned by the public in the tweets – e.g., character, personality, healthcare, crime, jobs, etc. (these are examples only, use wordfrequency.py to find out). Replace words if necessary (Using **find\_and\_replace.py** script).
3. Perform ***lift*** and ***sentiment analysis*** on each candidate and issue:

|  |  |  |
| --- | --- | --- |
|  | **Lift** | **Sentiment Score (if lift > 1)** |
| Issue 1 & Fetterman | ? | ? |
| Issue 2 & John Fetterman | ? | ? |
| Issue 3 & John Fetterman | ? | ? |
| Issue 4 & John Fetterman | ? | ? |
| Issue 1 & Mehmet Oz | ? | ? |
| Issue 2 & Mehmet Oz | ? | ? |
| Issue 3 & Mehmet Oz | ? | ? |
| Issue 4 & Mehmet Oz | ? | ? |

1. Show the attributes and candidates on an MDS plot (Using **mds.py** script).
2. Use the location data collected in step A to find out **lifts** and **sentiments** regarding the candidates in large versus small cities/towns in Pennsylvania.
3. What advice would you give to each of the two candidates based on your analyses in C, D and E above?

**Guidelines for parts A, B, C, and D – Please read carefully before running the scripts.**

1. Before you collect tweets, find suitable hashtags and keywords (go to Twitter and also search on Google to find out which hashtags and keywords are popular in describing the PA senate race). Keep in mind that there is some trial and error involved in finding suitable keywords for the Twitter search. As a last resort, you can do two searches, one with Fetterman and one with Oz, but first you should try to find more general keywords about the PA senate race.
2. Twitter doesn’t like to give more than 3000 tweets in a single search. In fact if you ask for more than 3000, you may get an error. My suggestion is to do multiple searches, each for 3000 tweets, spaced at least by 12 hours (otherwise Twitter will simply send more duplicate tweets). Make sure the output file name for GetTweets2022.py is changed before running the script each time. Then combine the multiple csv files into a single data file with copy and paste in Excel.
3. Once you have the required tweets, check for duplicate tweets and eliminate them (in Excel you can search for duplicate rows and eliminate them – Google it if you are not familiar with it. If Twitter gives too many duplicates, try searching again after a day or so. Remember, retweets (RT) are NOT duplicates. A row of data is a duplicate if it completely matches another row.
4. Make sure the text of the tweets is in the **third column** (most of the scripts were written for the Edmunds data, and hence expect the text to be in column 3). Also ensure columns 1 and 2 are not blank. Just delete the first few columns so that the tweets are in column 3.
5. You may have to do replacements – e.g., replace Mehmet or other variations by **Oz**, etc. (Using find\_and\_replace.py script and the models.csv file with appropriate changes)
6. Run the lifts between the 4 issues (like immigration, crime, etc.) and 2 candidates as in the Edmunds assignment (using **lift.py** and Edmunds\_key\_pair file with suitable changes). So you Edmunds\_key\_pair.txt file should contain: Fetterman,Oz,issue1,issue2,issue3,issue4
7. For sentiment analysis with issues and candidates (need to do this for candidates & attributes where the lift > 1), use the **parserforsentiment.py** script to extract mentions of a candidate and an issue. The script uses the tweets file as input, and asks you for (i) brand (Fetterman or Oz in this case), (ii) attribute (e.g., inflation in this case) and the number of words to retain on the right and left hand sides of the issue (the assumption being that people express their emotions within a close proximity of the attribute). I would recommend trying 7 or 8 words. **Note that both inputs (brand and attribute) must be typed in lowercase.** The output file should be labeled carefully, e.g., Oz\_immigration.csv. **Make sure the output file is renamed each time before running the script.** Now make this output file the input to **senitment.py**. **For each candidate and each issue, you will run sentiment.py once. So for Fetterman and Oz and 4 attributes, you will be running parserforsentiment.py and sentiment.py 8 times.**

**Guidelines for E**

1. For location analysis, get a list of PA city and town names from Google. Use the **models.csv file** and **find\_and\_replace.py** script to replace the 5 biggest cities by population (e.g., Pittsburgh, Philadelphia, etc.) with **big\_city** and the rest with **small\_town**. In the models.csv file, add big\_city and small\_town in the left column and the real city/town names of PA in the right column, and run **find\_and\_replace.py**. **To do this replacement, temporarily make the location field (in your input file) the 3rd column, since the find\_and\_replace.py script works on the third column**.
2. Once the replacements are done, **merge the location and the text columns** into one with the concatenate command in Excel (Google it if not familiar). Make sure this is the third column in the data file.
3. Perform a lift analysis as follows:

|  |  |  |
| --- | --- | --- |
|  | John Fetterman | Mehmet Oz |
| Big\_city PA | Lift? | Lift? |
| Small\_town PA | Lift? | Lift? |

Obviously the Edmunds\_pair\_keys.txt file should contain: Fetterman,Oz,big\_city,small\_town

Philadelphia 1,584,981 bigcity

Pittsburgh 302,407 bigcity

Allentown 123,828 bigcity

Erie 97,369 bigcity

Reading 89,872 bigcity

1. Use parserforsentiment.py again on this dataset, but now brand=candidate Fetterman or Oz, attribute=big\_city or small\_town and # words = 20. The output of this script should be a file which has tweets on a candidate from big\_city or small\_town. Now pass this output file through sentiment.py to get the overall sentiment of big\_city or small\_town residents about a candidate. **You will run the parserforsentiment.py and sentiment.py 4 times.**

Create a matrix as follows from the sentiment scores:

|  |  |  |
| --- | --- | --- |
|  | John Fetterman | Mehmet Oz |
| Big\_city PA | Sentiment score? | Sentiment score? |
| Small\_town PA | Sentiment score? | Sentiment score? |

You may find this article relevant to this assignment: <https://medium.com/swlh/the-pulse-of-the-nation-a-twitter-analysis-of-the-2020-u-s-presidential-election-15f9afa86ec7>