The City College of New York

Department of Computer Science

DSE I2450: Big Data and Scalable Computation

SPRING 2019

Final Challenge

Problem Statement: As the problem of drug abuse intensifies in the U.S., many studies that primarily utilize social media data, such as postings on Twitter, to study drug abuse-related activities use machine learning as a powerful tool for text classification and filtering. However, given the wide range of topics of Twitter users, tweets related to drug abuse are rare in most of the datasets. This imbalanced data remains a major issue in building effective tweet classifiers, and is especially obvious for studies that include abuse-related slang terms. In this final challenge, we would like to explore two methods to facilitate the capturing process of drug abuse activities more effectively (using Twitter data):

1. Generate a visualization showing the distribution of drug-abuse-related tweets throughout the country
2. Discover keywords in drug-abuse-related tweets using term-frequency/invert document frequency

Note that, in this initial study, we only want to apply the methods over major areas in the US, in particular, **within the top 500 largest cities in the US**. Regardless of the method, we also need to **filter** a collection of geo-tagged tweets to keep only those that are related to drug-abused **based on a set of pre-defined keywords**.

You can select to implement **one of the two methods** as your final challenge. They are listed below:

CHALLENGE 1: Visualizing the distribution of drug-abuse-related tweets

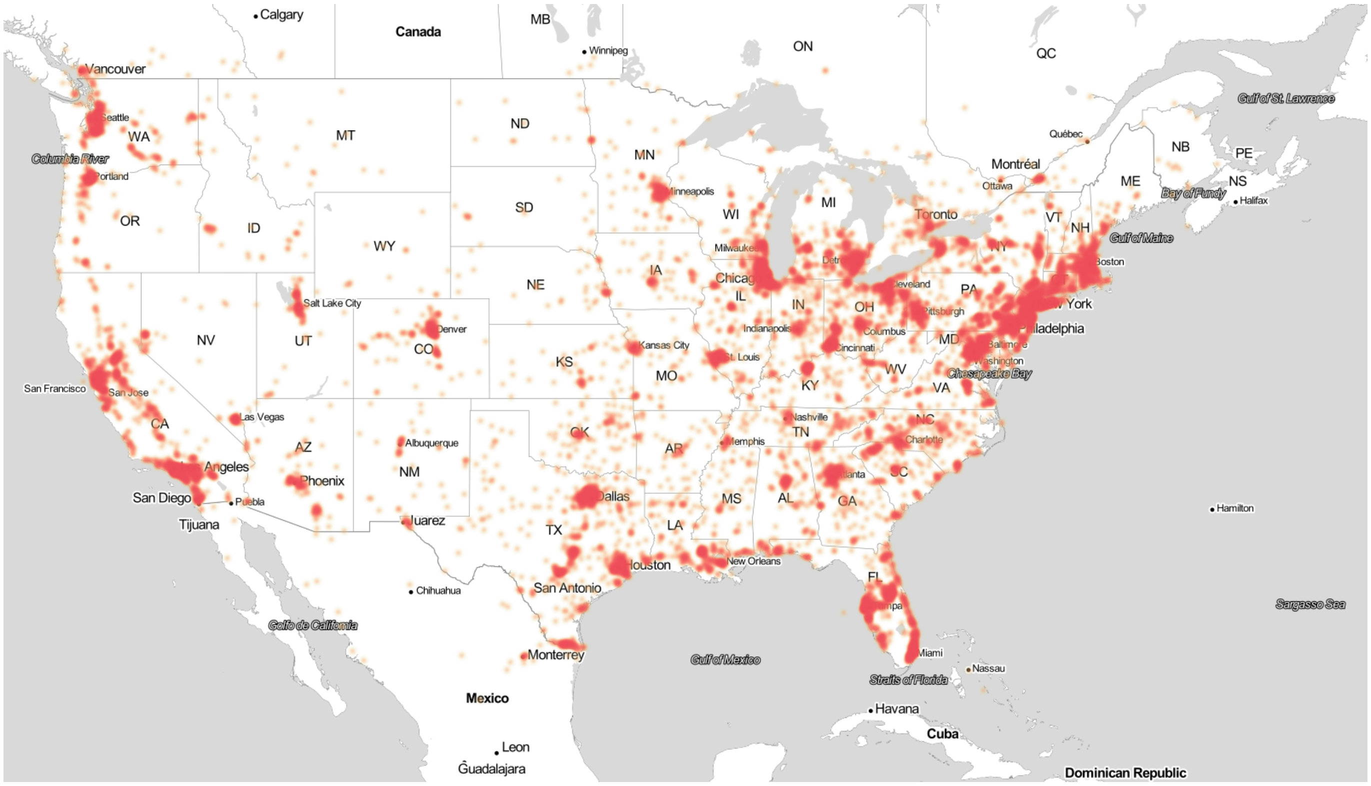
The geo-location information tagged in drug-related tweets is very useful to capture the distribution of drug abuse-risk behaviors. An example of the tweet distribution across the US is shown in the figure below. According to this visualization, the potential greatest drug threat regions could have been in the Florida region, the Great Lakes region, the Mid-Atlantic region, the New York/New Jersey region, the New England region, the Pacific region, the Southeast region, the Southwest region, and the West Central region. However, this information might be biased since the geo-location distribution should be normalized by population density. Your task is to produce the data needed for a less-biased visualization at the census tract level, where each represents the normalized number of drug-related tweets per population density. In other words, for each census tract, you need to compute the ratio of the number drug-related tweets over the population in the tract.

Your Objective: spatially join tweets that contains drug-related terms with the census tracts, and compute the normalized number of tweets using the provided boundary and population data. You must implement this using Spark, and demonstrate that you can do this task in a scalable way such that if there are additional tweets, or census tract data available, your code can still run efficiently (perhaps at the expense of using more cores).

Your submission: you can turn in one or more files including your application's main (Python) file and any dependencies that it may need. However, all of the submitted file(s) must be able to fit into a single **spark-submit** command running on NYU Dumbo cluster. Please provide the command that you run when submitting your code. For sanity check, please include the output of your job, in a single CSV/TXT file, with two columns consisting of the ***plctract10*** ID and its normalized number. The output must be sorted lexicographically by the ***plctract10*** field. The sample output is shown below:

```
0107000-01073000100,0.5
0107000-01073000300,1.5
```

…



** The distribution of drug-related tweets throughout the US.*

CHALLENGE 2: Identifying keywords for drug-abuse-related tweets

We would like to build a classifier of drug-abuse-related tweets to automatically determine whether a tweet is related to drug-abused. We already have labeled data for this classifier, but in order to efficiently extract tweet features, we would like to use a form of the Term Frequency-Inverse Document Frequency (TF-IDF). This metric can help us determine the importance of a word in a tweet, and whether we should use that as a feature in our model. In this challenge, to simplify the problem, you are asked to compute a simpler model of TF-IDF.

Your Objective: for each tweet that contains drug-related terms, compute the top 3 words with the smallest document frequency. From these words, please provide the top 100 words, and their respective tweet counts as your output. For example, assuming the following tweet message has passed your filter of drug-related terms and is within 500 largest cities in the US:

```
this is drug-related message
```

For each word in the message, you need to compute their document frequency defined as the number of tweets (including those that do not pass your filter) that contains the word. For example:

- the word "is" is included in 1 million tweets
- the word "this" is included in 50 thousands tweets
- the word "drug-related" is included in a thousand tweets
- the word "message" is included in 10 thousands tweets

In this case, the top 3 words are "drug-related", "message", and "this", in the exact order. Given the top 3 words for each tweets, you then need to compute which are the top 100 words that appeared the most in the top 3 words of the tweets.

<u>Your submission:</u> you can turn in one or more files including your application's main (Python) file and any dependencies that it may need. However, all of the submitted file(s) must be able to fit into a single **spark-submit** command running on NYU Dumbo cluster. Please provide the command that you run when submitting your code. For sanity check, please include the output of your job, in a single CSV/TXT file, with two columns consisting of the ***word*** and the number of tweets they appear as the "top 3" words with the smallest document frequency. The sample output format is shown below:

```
oxycontin,1000
weed,989
…
```

<u>DATA SETS:</u>

500 Cities: Census Tract Boundaries (including population data)

Source: https://catalog.data.gov/dataset/500-cities-census-tract-boundaries-b4acc

A GeoJSON version of file (with only the ***plctract10*** and ***plctrpop10*** fields) is also available on HDFS at:

hdfs:///data/share/bdm/500cities_tracts.geojson

100 million geo-tagged tweets in the US

Source: collected through the Twitter Open API

The data is in CSV format, however, the delimiter is the pipe character ("|"). The file location on HDFS at:

hdfs:///data/share/tweets-100m.csv

Drug-related keywords

Source: manually generated from schedule 2 drug names, and illegal drug names

The file contains a list of keywords, one on each line. If a tweet contains any of these words, they are considered drug-related. The file location on HDFS is at:

hdfs:///data/share/drug_illegal.txt
hdfs:///data/share/drug_sched2.txt

<u>EVALUATION:</u>

your code will be tested to run with exactly 10 executors, and 5 cores per each executor. As mentioned above, you must submit the command that you ran with your code, for example:

```
spark-submit --conf spark.executorEnv.LD_LIBRARY_PATH=$LD_LIBRARY_PATH \
--executor-cores 5 --num-executors 10 --py-files … \
--files  hdfs:///tmp/500cities_tracts.geojson,hdfs:///tmp/drug_sched2.txt,hdfs:///tmp/drug_illegal.txt \
final_challenge.py hdfs:///tmp/tweets-100m.csv
```

<u>JUSTIFICATION:</u>

You must provide a justification for your solution. Note that you are allowed to use approximation techniques, e.g. CountMinSketch, for these challenges, but you must an expected error for your solution. However, sampling technique is not allowed.