

Data Mining Assignment

NOTE:

- You only have to complete **ONE** of these suggested problems.
- You need to complete these in Python and it has to run.
- Feel free to use any Python library you wish.
- My email address is sreejata@leadsift.com, in case you have any questions.
- Due date: Sunday - March 16th, 2023 (midnight Atlantic Time).

Assignment 1:

Use any data either by downloading from Twitter or one given by govt of Canada (or any other dataset you're interested in playing with) and figure out a data point that will surprise and impress you, your friends or me. Link to Canadian datasets: <http://open.canada.ca/data/en/dataset>

Upload the files/folder and clearly name the file that you want me to run as: your
firstname_lastname_any_other_description.py

Notes: This can be as simple as reading a file and finding the frequency or counts, averages etc. Or as complex as running clustering on interesting datasets to find patterns. The goal is to start understanding how to deal with large datasets and make sense of them in the real world, with undefined goals - what data scientists deal with on a daily basis.

Simple example: Which car gives the best mileage or has the least depreciation etc.

Interesting example: Read earning calls or nasdaq reports to predict which stock we should buy

Assignment 2:

Build a simplistic search engine. Please document the code well and write down what it does and doesn't do, what kind of ranking algorithm it uses, what data it searches on and what are the assumptions.

Upload the files/folder and clearly name the file that you want me to run as: your
firstname_lastname_any_other_description.py

Note: The goal here is to understand the workings of a search engine - please design the whole search engine and feel free to mock out the first or the last parts of it. For example, just download some webpages (using curl or wget) and search on them instead of trying to crawl the whole web or crawl in real time. You can also do a search on a single website (like SMU or Amazon) but please don't submit code that does a simple string match on it. No need for fancy display of search results. [Here is a good resource on building a Python search engine](#); but yours can be even simpler. Please include your design (a photo of

your scratchpad is acceptable).

Assignment 3:

Download the sentiment labeled dataset of tweets:

<https://archive.ics.uci.edu/ml/machine-learning-databases/00331/>

1. Write a Naive Bayes Classifier for sentiment detection by training using above data (or any other labeled dataset you find)
2. Use at least two other classifier algorithms and report the difference in accuracy, tweak them so you get the best possible results.
3. Save the classifiers so you (and I) don't have to retrain every time. Please send me the code to read from the saved classifiers so I don't have to run the classification again. (You can do this by either writing the classifying code in a separate file or by checking if classifier pickle exists)

When submitting the code, the file that you want me to run should be called `firstname_lastname_any_other_description.py` and it should assume the training files, saved classifiers and any other required files are all in the same folder. It should read from the saved classifiers and not retrain it when I am running the code. Please include the results (accuracy) of classifiers as a text file.

Notes: Take a look at Lecture 2, the classification code is provided. Feel free to use it as a starting point and add the other classifiers. [Here's another helpful tutorial on NLTK/Python.](#)

Feel free to email me with any questions you may have.

All the best!