Part 1 Proposal

Austin Lipe

I want to do a traffic crash analysis in Texas using crash count data and shapefiles obtained from the TXDOT website. My backup project idea is to do some type of TF-IDF implementation to efficiently search the Texas Administrative Code (TAC) or the Code of Federal Regulations (CFR). I detail both project ideas in more depth below.

1. **Traffic Crash Analysis (Brief Description and Learning Objectives)**
    a. I want to work with crash data collected by TXDOT. An objective I think would be fun to try is to perform a clustering analysis using the location of crashes. This could indicate more dangerous sections of roads. However, clusters in cities will always have higher crash counts than clusters in rural areas. So, I also want to calculate a fatality rate in each cluster. This could help normalize each cluster to a standard metric. Then I could find the clusters with the highest fatality per crash which would indicate the most dangerous sections of roads in Texas. Civil Engineers could then use this data to locate road segments with design flaws that might require maintenance or need to be re-designed.
    b. Data
        i. TXDOT has a lot of data (~3 GBs) on crash counts over the last 10 years. Given the size of the data, distributed computing would be necessary to efficiently determine high density crash zones throughout the state. I have not researched crash data or traffic volume counts in other states. That would be a good topic to research further for my project.
    c. Learning Objectives
        i. Use s3 to store large csv files containing crash data.
        ii. Use Pyspark and a clustering algorithm to identify clusters of crashes in Texas.
        iii. Determine segments of roads in Texas that are particularly dangerous using a fatality rate metric that could indicate which areas need maintenance.
    d. Specific Big Data Task
        i. Use a clustering algorithm to find clusters of crashes using their location and then calculate a fatality metric to find the most dangerous clusters
    e. Reference List
        i. TXDOT Crash Records and Information System (CRIS) Website
            1. This source contains specific information about the dataset I am planning to use for my project.
        ii. https://www.nhtsa.gov/research-data/crash-injury-research
            1. This source contains multiple research papers about the causes of crashes in the US.
        iii. https://highways.dot.gov/public-roads/septemberoctober-2016/big-data
            1. This source contains information about how USDOT is using big data to improve traffic safety and flow in the US.
    f. Data Source
        i. TXDOT database for crash data (~3 GB)

ii. TXDOT database for major road segments (unknown size).

g. Further Research

i. Are some clustering algorithms easier to work with in pyspark than others?

ii. What are other metrics that are used to normalize raw crash counts in areas? Crash count per volume of traffic is another obvious metric, but I'm sure there are more metrics to determine dangerous sections of roads.

iii. Should I try to distinguish between behavioral and non-behavioral crashes? For example, a person who crashed their car because they were drunk would be a behavioral crash, and this data point wouldn't be a good indication of a section of road that needed maintenance.

iv. What kind of crash data is there in other states?

2. EPA and TCEQ Environmental Regulatory Documents (Brief Description)

a. There are so many rules and regulations from both the EPA and The Texas Commission of Environmental Quality (TCEQ) that I think it would be very helpful to collect a lot of these documents into files and implement some type of advanced search feature over them with TF-IDF. This would help environmental consultants and companies in general efficiently find relevant regulations to their industry.

b. **Learning Objectives**

i. Utilize common NLP techniques such as TF-IDF to search through document chunks for relevant sections

ii. Utilize pyspark to implement TF-IDF algorithm for all documents

iii. Create a basic search engine for TAC and CFR documents to aid individuals and companies who work with these documents on a regular basis.

c. Data

i. Texas Administrative code found on Texas.gov which is a compilation of all state agency rules in Texas.

ii. EPA regulations are contained in the Code of Federal Regulations (CFR).

1. They seem to have a much more advanced search engine than the TAC so implementing TF-IDF search on this document is not as useful.

d. **Specific Big Data Task**

i. Implement TF-IDF algorithm on the TAC and CFR to improve search accuracy.

e. Reference List

i. https://www.ecfr.gov/

1. This is the online location of the CFR

ii. https://texreg.sos.state.tx.us/public/readtac$ext.viewtac

1. This is the online location of the TAC

iii. https://www.researchgate.net/publication/301708935_Multi-Document_Summarization_Using_TF-IDF_Algorithm

1. This is a in depth analysis of using TF-IDF algorithm for search

f. Further Research

i. I don't know how I would extract the TAC or the CFR from their website yet so that is definitely something I need to search further.