Executive Summary

The objective of this study is to identify dangerous sections of roads in Texas by performing a clustering analysis using the locations of all crashes in Texas. The number of crashes on a particular section of a roadway doesn't necessarily indicate how dangerous the road is, as busier roads tend to have more crashes compared to less busy ones. It might be that many crashes on an interstate result in far less injuries and deaths than a few crashes that take place at a dangerous intersection that receives far less traffic volume than an interstate. As a result, dangerous sections of roads in this study are identified through the number of fatalities per crash in a roadway section. If dangerous sections of roads are identified through this crash data analysis, then the information can be shared with civil engineering firms who can present the information to local municipalities or with Texas Department of Transportation (TXDOT). The engineering firms can petition to reconstruct the roadways to improve the overall safety of the community. A K-Means Clustering algorithm was used to perform the clustering due to its efficient computations and interpretable results. An EMR Cluster was utilized to perform the clustering analysis on the entire dataset. The top five most dangerous clusters were then found by calculating the fatality rate of each cluster. The most dangerous roadway segment is located on TX 54 approximately eighteen miles north of Van Horn, TX with a fatality rate of 0.57. This may be attributed to the increased traffic volume generated by the oil and gas industry, as well as the rocket development industry in the area.

Background

Data for this study was obtained from the Crash Record Information System Database (CRIS) Database. It is a requirement to report all motor vehicle crashes that result in injury, death or property damage to the authorities and to TXDOT. Each crash is then meticulously logged into CRIS. Currently, CRIS does not have a public API and all crashes must be requested through email and the data will be sent via CSV file chunks. CRIS has been actively collecting crash data since 2014.

The K-Means Clustering algorithm was utilized as part of the Big Data Solution for this project although Gaussian Mixture Model (GMM), DBSCAN and Hierarchical Clustering algorithms were all considered and tested on smaller subsets of the data.

K-means was chosen for its fast and efficient algorithm and easily interpretable results, however estimating the initial number of clusters for accurate clustering results was difficult. Additionally, K-means struggles to cluster irregular shapes which led to poor clustering results in downtown sections or other high density areas where it clusters parallel roads in the same groups.

GMM was considered because it excels at clustering data with various sizes, shapes and densities, which is optimal when trying to cluster crashes which can form shapes and densities of various sizes. However, GMM is computationally intensive for a Big Data Solution and

estimating the number of clusters is difficult. Nonetheless, GMM has a built-in method in spark.ml.clustering and will be the next clustering algorithm implemented in future iterations of this project despite its drawbacks.

DBSCAN and Hierarchical clustering were also explored on small subsets of the data using scikit-learn and pandas, however a Big Data solution was not implemented using either of these algorithms. There is a lot of potential to improve clustering methods using DBSCAN because it relies on density based metrics and can cluster irregular groups very well, however it was not used because there was not a built-in implementation in spark.ml.clustering. Clustering results of DBSCAN and Hierarchical Clustering can be in the **Results** sections below.

Two equations were created to obtain an accurate metric on the danger of a roadway section:

$$Fatality\ Rate\ = \frac{Fatality\ Count\ of\ the\ Cluster}{Count\ of\ Crashes\ in\ the\ Cluster}$$

$$Injury\ Rate\ = \frac{Injury\ Count\ of\ the\ Cluster}{Count\ of\ Crashes\ in\ the\ Cluster}$$

In the current iteration of the project, the Fatality Rate is the only metric utilized to determine the most dangerous cluster. In future iterations of this project the Injury Rate will be incorporated into the analysis to more accurately determine dangerous clusters.

Implementation

Dataset

The dataset utilized is 1.3 GB and is stored as a CSV file in an S3 bucket on AWS. Each row in the dataset corresponds to a single crash. The crash can contain multiple vehicles and people. A few key features were utilized from the dataset:

1. **Latitude (float)**: Latitude of crash
2. **Longitude (float)**: Longitude of crash
3. **Sus_Serious_Injry_Cnt (int)**: The number of serious injuries
4. **Tot_Injry_Cnt (int)**: The total number of injuries in a crash
5. **Death_Cnt (int)**: The total number of deaths in a crash

Methods

The data was first loaded into a spark batch py file called `src/create_regions.py` that broke the dataset into four small regions for simple development and testing on a local computer. The four small regions included crashes in Boerne, TX, Downtown Austin, TX, Sugarland, TX, and Downtown San Antonio, TX. Exploratory Data Analysis (EDA) was performed on the regions using Pandas. Maps were created plotting the crashes and the preliminary results of K-Means, Hierarchical Clustering and DBSCAN clustering algorithms using Scipy and Scikit-learn. The results of the EDA using Pandas on the four small regions can be found in the `notebooks/test_pandas_clustering.ipynb` file. Classes and functions defined for the Pandas EDA can be found `src/pandas_clustering.py`.

A module containing Spark classes and functions was then created to perform the clustering analysis that would be capable of scaling to a large dataset. The spark module was then tested on the four small regions. This module can be found in `src/spark_clustering.py`. Results of the Spark Clustering Analysis can be found in `notebooks/test_spark_clustering.ipynb`. The implementation steps follow this outline:

1. **Load CSV file as a spark.sql.dataframe**. A spark.sql.dataframe was used because this class object is compatible with spark.ml.clustering.kmeans machine learning algorithm.
2. **Clean spark.dataframe.** All crashes containing NaNs of the five key features outlined above were removed.
3. **Implement spark.ml.clustering.KMeans.** Clusters for the K-Means algorithm were chosen by dividing the number of total crashes by 250. This number was chosen because the four small regions analyzed in the EDA indicated that there were roughly 250 crashes per intersection in the last 5 years of data. Therefore, determining the number of clusters that results in 250 crashes per cluster resulted in moderately distinct clusters per intersection. Distinct clusters for intersections is crucial when determining risk because specific intersections can oftentimes be much more dangerous than others even though they are in close proximity to each other.
4. **Run EMR Cluster on the entire dataset.** A spark batch file was then created using the classes and functions defined in the `spark_clustering.py` and the computations were run on an EMR cluster. Results were saved as a CSV file in an s3 bucket.
5. **Plot results.** The top five most dangerous clusters and the surrounding clusters were then plotted on a map based on the results of the Big Data Analysis.

Results

The following section compares the results of DBSCAN, Hierarchical Clustering, and K-Means clustering algorithm on a small subset of the data. **Figures 1, 2** and **3** show DBSCAN, KMeans and Hierarchical clustering results in Boerne, TX (~4,000 crashes). Boerne, TX was chosen because there are three types of crash clusters in this area: dense urban intersections, a major highway, and low traffic rural roads.

The clusters created from the DBSCAN algorithm using hyperparameters of epsilon=0.003 and min_samples=100 resulted in clusters containing distinct intersections and roadway segments along the major highways. However it performed poorly distinguishing different clusters in the low volume rural roads. DBSCAN clustering results can be seen **Figure 1**.
The clusters from the KMeans algorithm in **Figure 2** resulted in clusters of semi-distinct intersections and roadway segments. However, some of the intersections in the dense urban section were combined, and clusters along the major highway were not separated very well from intersections that align with the major highway.

Hierarchical Clustering in **Figure 3** performs similar to KMeans with moderately distinct clusters in the dense urban environments and along the major highway. It does seem like it produced more distinct clusters in the low traffic rural areas than the KMeans algorithm.

DBSCAN is certainly preferable to KMeans or Hierarchical Clustering for the crash clustering task given the visual analysis of plots below. However the KMeans algorithm should be sufficient for determining high danger traffic areas, and it is much more computationally efficient when applied to big data than DBSCAN. Additionally there is already a built-in method for using KMeans with spark.sql.dataframes simplifying the implementation.

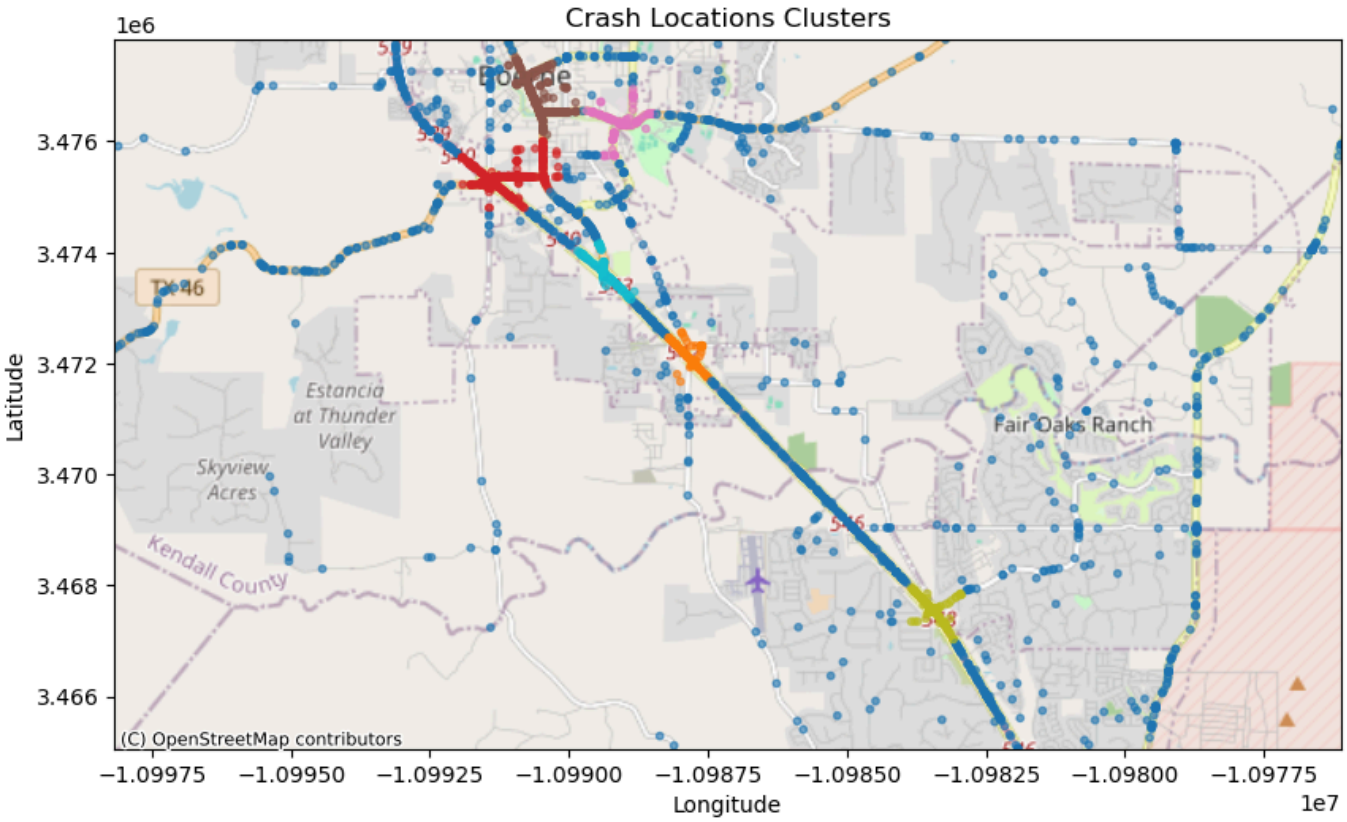

Figure 1: Boerne, TX Clusters produced by DBSCAN Algorithm.

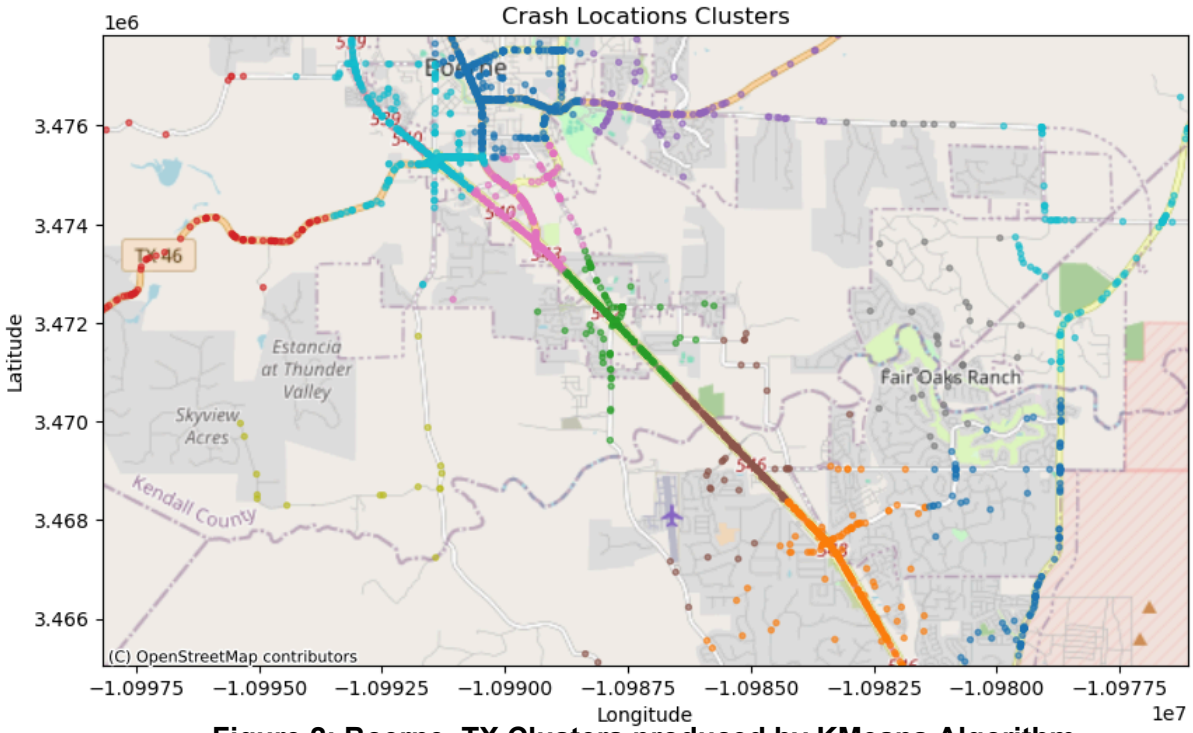

Figure 2: Boerne, TX Clusters produced by KMeans Algorithm.

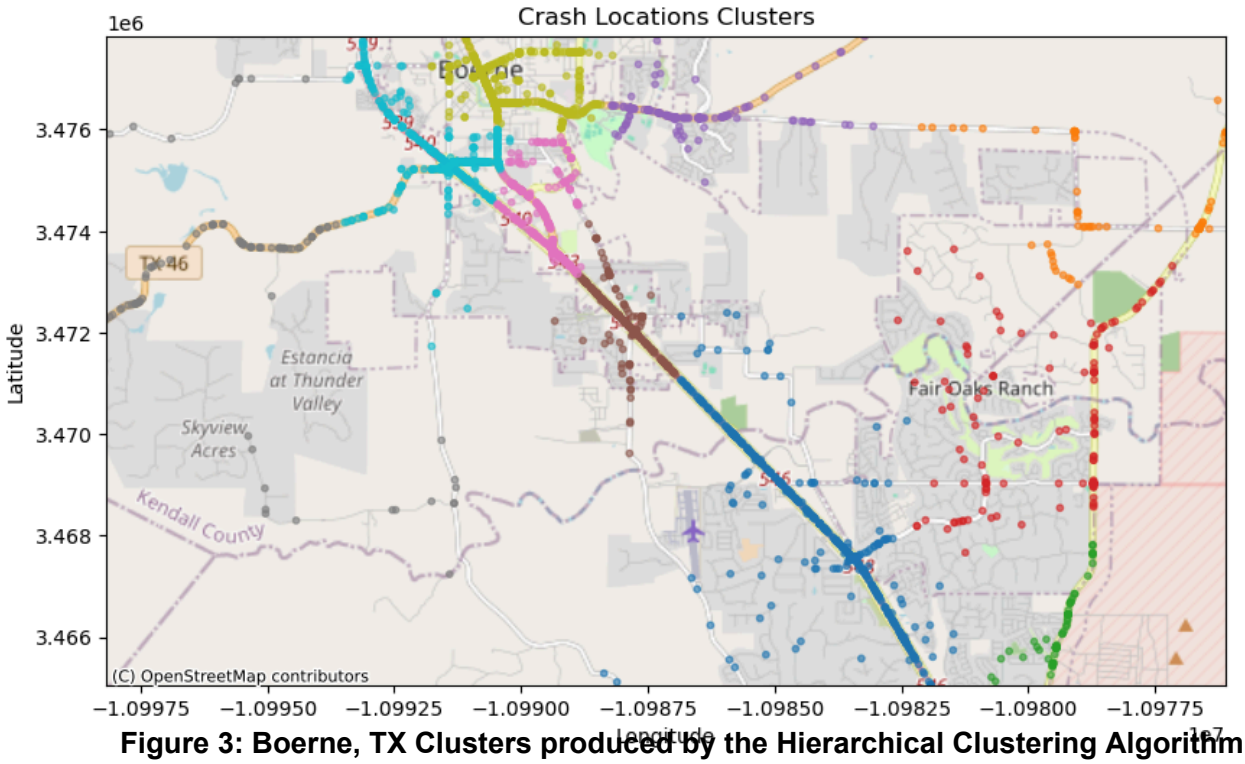

Figure 3: Boerne, TX Clusters produced by the Hierarchical Clustering Algorithm.

The following table shows the top five most dangerous clusters in the state given the results of the Big Data analysis using the K-Means clustering algorithm. Column 1 describes the approximate location of the cluster centroid and column 2 lists the fatality rate of each cluster. Plots of the top five most dangerous clusters are saved as .html files in the `plots/fatality_plots/` subdirectory in the supporting files.

Table 1: The Location of The Top Five Most Dangerous Clusters in Texas.

Location	Fatality Rate
TX 54 eighteen miles north of Van Horn, TX	0.57
US 277 sixty south of Sonora, TX	0.53
TX 349 on the county line of Pecos/Terrel County	0.50
US 380 thirteen miles east of Tahoka, TX	0.45
US 90 ninety west of Del Rio, TX	0.44

Figure 4 below shows the most dangerous cluster centroid on a map and the surrounding area. The green dots along the lower half of **Figure 4** are clusters with low fatality rates along I-10 near Van Horn. The red dot in the upper half of **Figure 4** is the location of the roadway segment with the highest fatality rate on TX 54 about 18 miles north of Van Horn, TX.

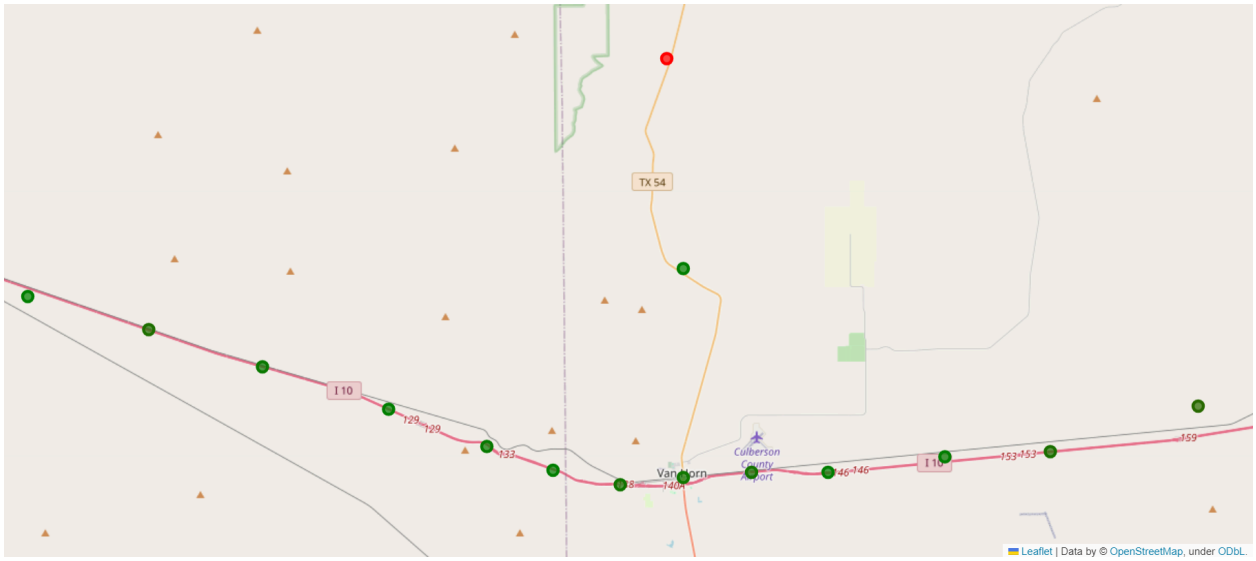

Figure 4: Picture of the area with the highest fatality rate in Texas. Dots on the map represent cluster centroids and their relative fatality rates. The full plot can be found in the `plots/fatality_plots/` subdirectory in the supporting files.

Discussion

The number one most dangerous cluster in Texas is TX 54 which is a rural road in West Texas approximately eighteen miles north of Van Horn. This may be attributed to the increased traffic volume generated by the oil and gas industry, as well as the rocket development industry in the area.

The K-Means algorithm does not perform well when clustering distinct intersections in urban environments. This can be seen in **Figure 5** of US 277 in Del Rio, TX. Cluster centroids very clearly represent multiple intersections (For a more detailed view of plots, open .html files found in `plots/fatality_plots` subdirectory of the supporting files). Intersections should contain distinct clusters to determine which might need design improvements to increase safety.

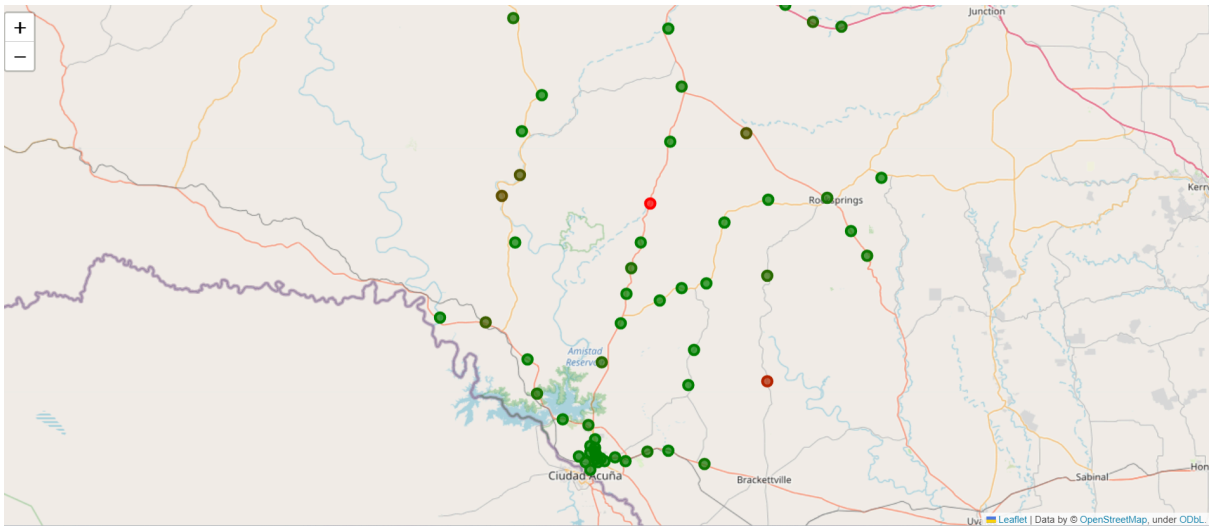

Figure 5: Second most dangerous cluster in Texas along US 277 sixty miles south of Sonora, TX. The full plot can be found in the `plots/fatality_plots` subdirectory in the supporting files.

Some clusters represent side streets and major highways and it is confusing whether the dangerous roadway segment is on the highway or a side street. This can be seen in **Figure 6** on US 380 thirteen miles east of Tahoka, TX.

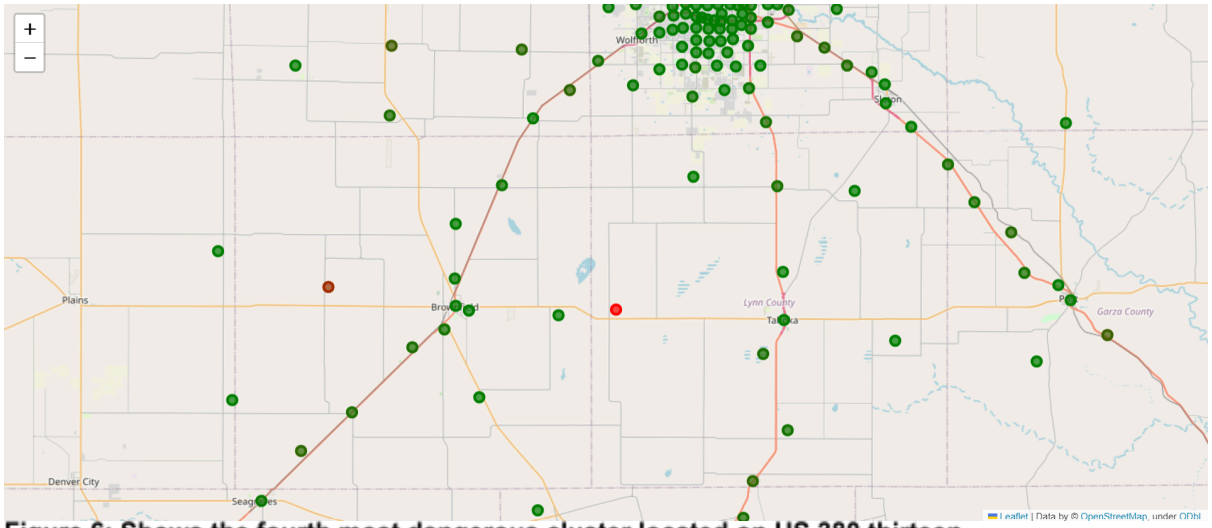

Figure 6: Shows the fourth most dangerous cluster located on US 380 thirteen miles east of Tahoka, TX. The full plot can be found in the `plots/fatality_plots` subdirectory in the supporting files.

Four of the five most dangerous clusters on this list are on rural roads in West Texas. This could be due to low crash counts in these areas skewing the fatality rates. The analysis could be improved by separating crashes in urban and rural environments before clustering. Urban crash clusters might be more accurate if a different clustering algorithm was used such as DBSCAN. Overall the K-Means algorithm seems sufficient for clustering.

Future Work

For future iterations of the project I want to perform a better analysis to evaluate the results of the K-Means Clustering algorithm on the entire dataset. This can be done by creating more plots to visualize the clusters in the most dangerous sections of the state with both the crashes and the fatality rate of the clusters. This analysis will help determine whether a cluster is associated with a hazardous intersection or curve in a roadway. It can also assist in identifying if the cluster is poorly defined (e.g., multiple intersections grouped into one cluster), potentially skewing the fatality data as a result.

Determining the size and number clusters is one of the most crucial steps when performing this analysis and I think there could be lots of improvement in that area of the analysis. There are three different methods that I want to experiment with during future work in this study: Use DBSCAN to determine the clusters, use shapefiles and roadway segments determined by TXDOT to determine clusters, and cluster using more dimensions to determine dangerous clusters.

1. **DBSCAN:** This algorithm would be advantageous for determining clusters in crash data because it is density based and can model irregular shapes very well. Crash clusters are almost always in tightly grouped irregular shapes either along a roadway segment or at an intersection which is a perfect use case for DBSCAN. I implemented DBSCAN using scipy and scikit-learn on a small scale using pandas dataframes with good clustering results after I tuned the hyperparameters. I did not initially use DBSCAN during the spark implementation of the project because there was no built-in implementation that supported the DBSCAN algorithm using a spark.sql.dataframe.
2. **Merging on Roadway shapefile:** TXDOT has shapefiles mapping out all major roads in Texas and has broken the roads into segments based on various characteristics. It could be very advantageous to merge the road segments and the crash points based on their locations. An unsupervised ML algorithm would not be required in this scenario because the clusters would be the roadway segments determined by the TXDOT shapefiles. There are built-in methods for merging geographic points and regions in geopandas which would make this kind of merge very simple on a small scale. I do not know how easy this kind of merge would be on a larger scale.
3. **Clustering using more Dimensions:** Instead of using a fatality rate equation to determine dangerous clusters, it would be interesting to cluster using more dimensions than just latitude and longitude. The analysis could yield better results by applying GMM or DBSCAN on the data with latitude, longitude, fatality count and injury count. The

results of this model alone might provide an accurate summary of the most dangerous intersections and roadways segments in Texas.

References

1. https://www.nhtsa.gov/research-data/crash-injury-research
2. https://highways.dot.gov/public-roads/septemberoctober-2016/big-data
3. Source Code for the project can be seen in the attached zip file.