Authors: Fateen Anam Rafid, Mark Raj, Daniel Zhang
Team Name: Big Data, Machine Learning, Internet of Things, and Artificial Intelligence
Team Number: 05

Spatial, Temporal, and Meteorological Analysis of Accidents in Greater Nashville

Overview

We were tasked with using data of accidents in Tennessee and Big Data tools and methodologies to discover trends in accident data. Specifically, this report analyzes incidents based on their spatial and temporal distribution and investigates the relationship between response time and location and time of the incident. The spatial analysis examines the incidents' distribution and identifies areas with a high incidence rate. The temporal analysis investigates patterns in the occurrence of incidents over time.

There were 3 main parts of our project. First, we looked at accidents through a spatial lens, identifying high frequency census tracts and correlating response time and demographic information using machine learning. The second was looking at response time through a temporal lens, where we tracked how long response times were depending on time of day, the month across several years, and investigated if there was a trend from year to year. Finally, we looked at response time through the lens of weather, where we created visualizations for response time based on weather conditions and used machine learning models to predict response time based on known weather conditions. Overall, we hope each of these insights could be provided to the Nashville Metropolitan government to improve response time for neighborhoods that have been underinvested in, to have smart suggestions for the time of day/year, and to adapt to the weather conditions of a given day.

Data

We were provided with 3 main sources of data. The primary source was the incident data itself, which was a log of accidents in Nashville from 2017 to 2021 (Pettet et al., 2017). The data contains several variables describing incidents, including a unique ID, the incident's latitude and longitude coordinates, the type of mission based on a priority dispatch code, the time the incident was reported in UTC and local time zones, the response time in seconds, the day of the week and whether it was a weekend day, the location of the incident, a generated incident ID, and the distance to the closest XDSegID. Of these, we focused on location for the spatial analysis, local time of day for the temporal analysis, and the response time for the weather analysis.

The second source of data was the Tennessee census tract geometries and demographics. This was provided in the form of a folder with shapefiles that contained information about the geometry of each census tract and other geographic information. There was also a demographic information .csv file for each census tract containing population numbers for different demographics and housing numbers.

The third data source was the weather information across Tennessee from 2010 onwards. This was stored into one large zipped Parquet to allow for easy download and extract. The directory structure was a folder for each year and a nested folder for each month, which contained a parquet with the weather data. This dataset provides daily weather history data for various locations, including latitude and longitude coordinates, the local timezone and city name, city and country codes, and the nearest station ID. The data includes several weather-related variables, such as average and maximum wind speed and direction, temperature, relative humidity, cloud coverage, precipitation and snowfall amounts, solar radiation, and UV index. The data also includes timestamps and time zones for each variable, as well as sources used in the response.

Solution Design

To tackle this problem, we started with the largest dataset, which was weather parquet. To store the size of the weather dataset, we placed it in the AWS S3 bucket that we each created for ourselves. This was an initial pain point as with the lab version of AWS as we aren't able to share resources or keys unless one person has their lab running. This meant we had to parallelize our personal tasks so each of us could work independently without blocking one another. Once this data and the incident data was included in S3, we then wrote a Spark job using Elastic Map Reduce to join each incident to all the corresponding weather data that it shared a timestamp with. Since our weather data couldn't be held in memory locally, we used EMR. The join produced another large dataframe because an incident would be joined with multiple weather rows from multiple TN weather stations that took readings at the same time. Using another EMR Spark job, we then filtered all the rows to only include the weather data for the weather station that was closest to the incident, ensuring we had accurate weather data for the time of incident. This result was then stored in a parquet file and shared.

Once this Big Data processing component was completed, we had a smaller dataset with the same number of rows as our original incidents dataset (~30k rows), but had weather data from the closest weather station for each incident. The first section was using plotly and seaborn to begin visualizing the data to both understand the relationship between the weather and response time and to provide clear stories for a potential audience. Finally, we used scikits-learn to see if we could predict response time based on weather factors. We used a random forest model and gradient boosted regression to compare different models and their performance on the dataset. We used an 80/20 training/test split and hyperparameter grid search to improve performance.

Once we had this large dataset processed, we moved into each of the smaller datasets that we could extract the necessary information from locally. We used Google Colab to load in the incidents dataset into a Pandas Dataset. We conducted the temporal analysis using pandas functions, starting with a year-by-year analysis (from the included datetime information), then a month-by-month analysis across the years, and, finally, a time of day analysis. We used plotly

and matplotlib to create visualizations, discovering correlations between each of these time metrics and response time and incident frequency.

For the spatial analysis, we used our local devices to perform a spatial join with GeoPandas. We read the census tract geometries into a geopandas dataframe and read our incidents data into a geopandas dataframe. We did a left join on "within", where each row of the incidents data was matched with the census tract whose polygon contained it. We then joined this resulting dataframe with the demographic information on the TRACTCE/tract columns, which identified the census tract. Finally, we used sklearn and plotly to visualize and analyze the data. We produced plots of demographic info, response time, and number of incidents. We also ran a simple linear regression to predict response time.

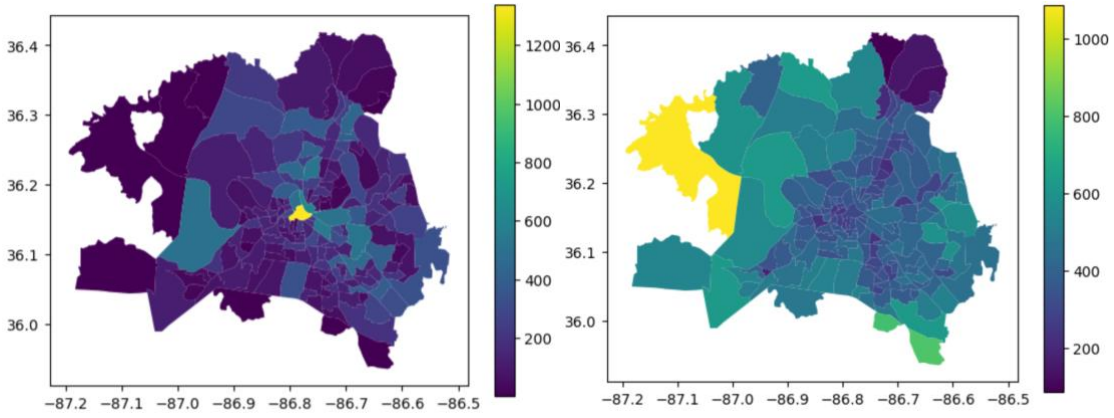
Technologies

From the lessons learned from class, we used pandas to work with data that could fit into memory on Google Colab while EMR and Spark we used to work with large datasets that couldn't. As an extension of what we discussed in class, we used GeoPandas to work with geospatial data and visualize the census tracts. Using EMR and Spark, we were able to join and filter the large weather data and add data to each incident in our primary incident dataset. With GeoPandas, we were able to do spatial joins on data stored as points (our incidents) and polygons (the census tracts). With pandas, plotly, and seaborn, we were able to analyze and visualize characteristics of the final joined table (which was small enough to fit into memory). Finally, we used scikits-learn to conduct machine learning to predict response time based on different factors. Overall, it was very interesting

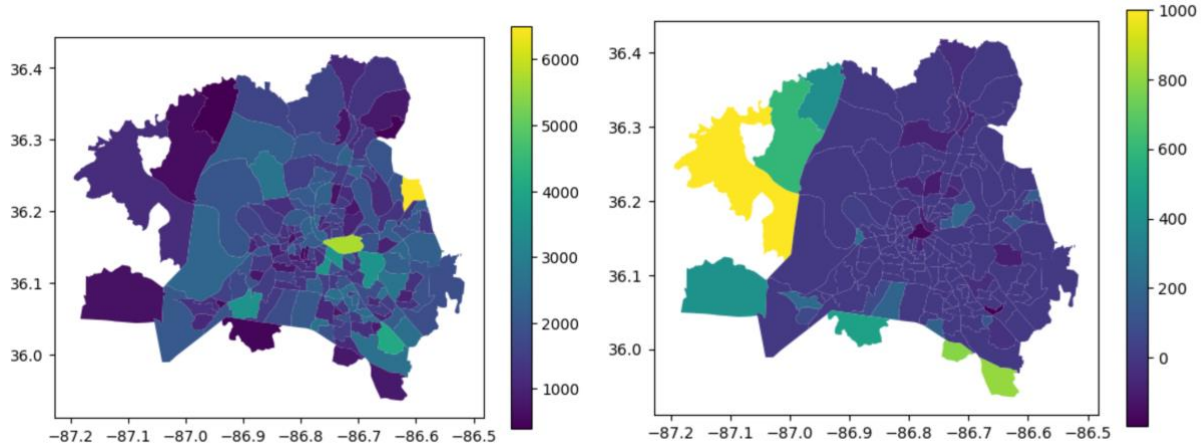
Insights and Visualizations

Spatial Analysis

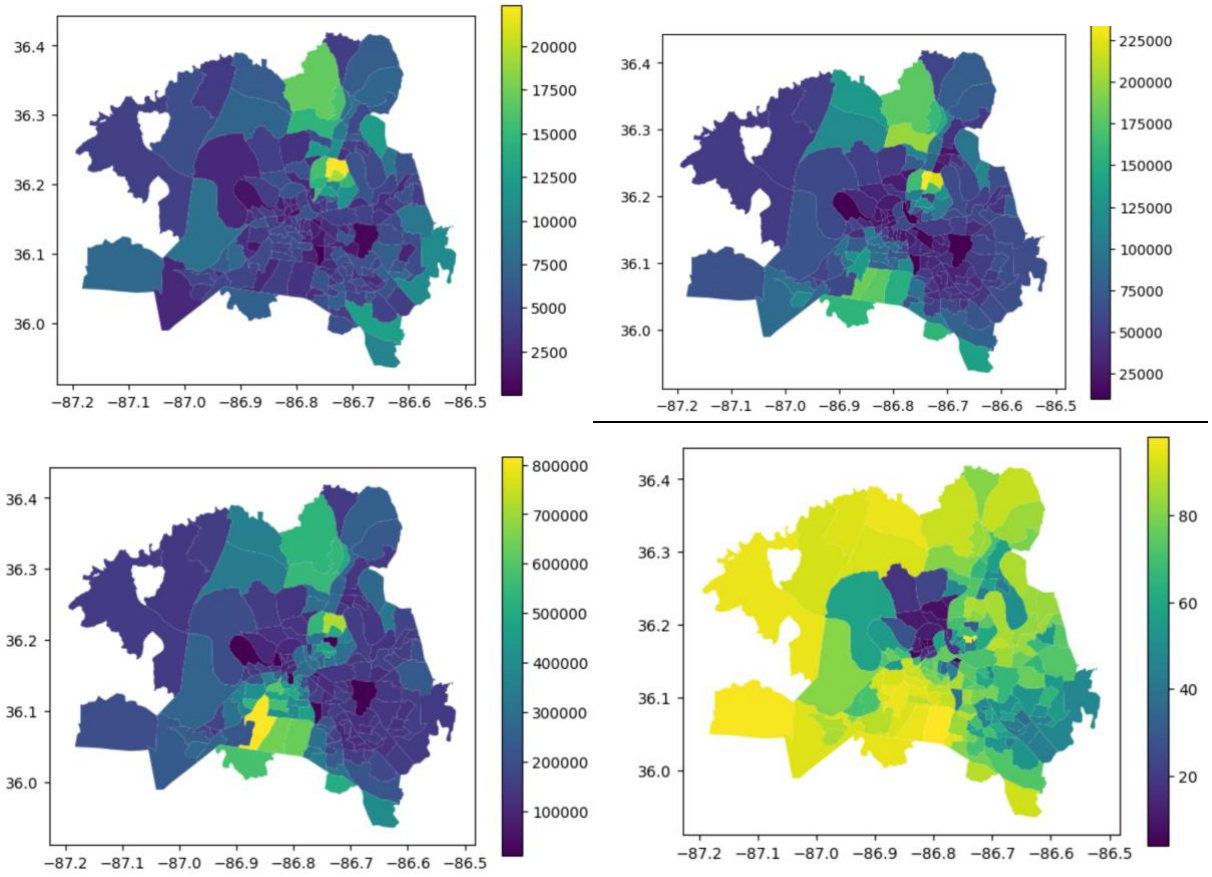
We generating the following visualizations of Nashville's census tracts:



Average response time across tracts (left) and number of incidents per tract (right).



Summary statistics for the maximum response time (left), minimum response time (right)



Census tracts by demographic information: population (top left), household income (top right), housing value (bottom left) and percentage population identified as white (bottom right)

To try to get further insights, we also did a simple linear regression on the 172 census tracts represented here, using the demographic data as features to predict the average response time. We split the 172 tracts with a 80/20 train test split, then fit to the training data. Here are the

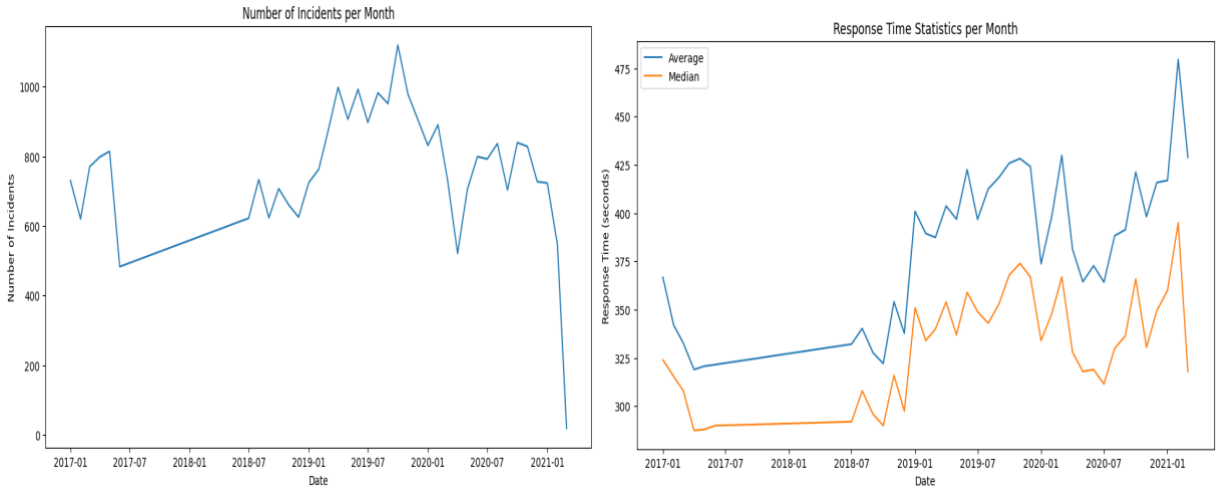
MSE of the fitted linear model on the test split and the coefficients of the linear model after being fit.

```
Mean Squared Error: 12117.671798986745
coefficients:
white_pct: 4.87436652984707
black_pct: 3.97840481083617
native_pct: -33.45522997972097
asian_pct: 6.469662191794404
hawaiin_pct: 6.802346635761282
other_race_pct: 6.162992574284308
two_or_more_pct: 5.167454849428027
hispanic_pct: -2.704187689415109
total population: 0.023690781814500527
male 25 to 29: 0.05490643190439744
female 25 to 29: 0.004868666862261324
median household income: -7.863166948803038e-08
median family income: -2.5983042702090285e-07
total housing units: -0.062447952415772695
occupied housing units: -0.04154458414277852
owner occupied housing units: 0.05713651643416034
housing units btw 500k and 750k: -0.11577965542443706
housing units btw 750k and 1m: -0.30426385245045806
housing units above 1m: 0.694254936261771
vacant housing units: -0.020903368017232925
median housing value: 3.9279671870673383e-07
median gross rent: -5.747521969468039e-08
```

Temporal Analysis

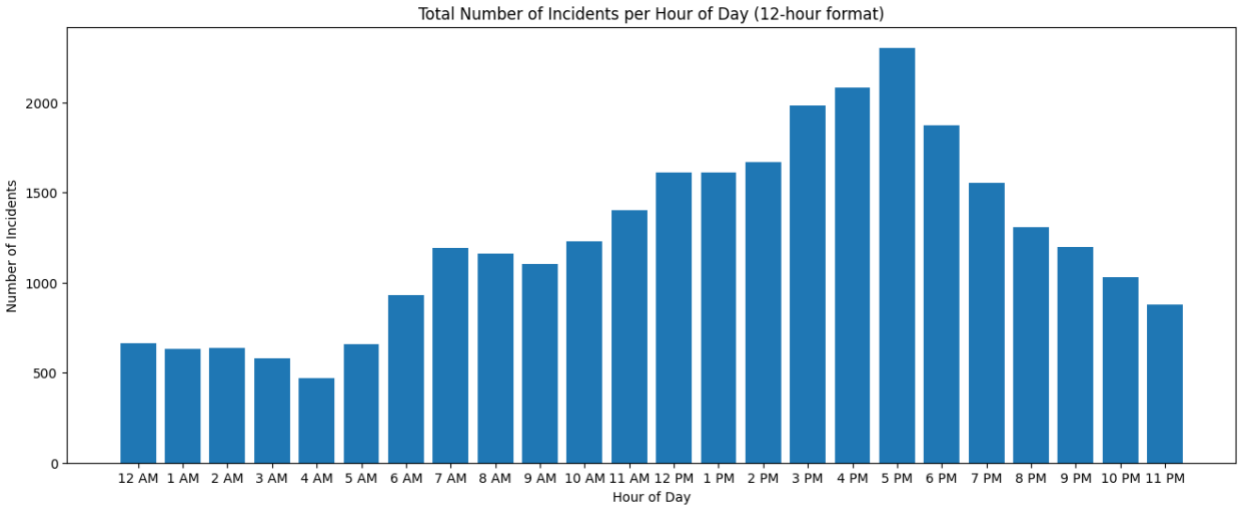
To do our temporal analysis, we grouped by hour and month and plotted how response time and total number of incidents vary by hour and month. Incidents happen more in January and February, which inspired us to think about weather, which we investigated later. They also happen more at certain times of the day, like 5PM (rush hour), though we did not have the scope to investigate the relationship between traffic and incidents. We overlaid some of these charts to demonstrate the relationships more obviously.

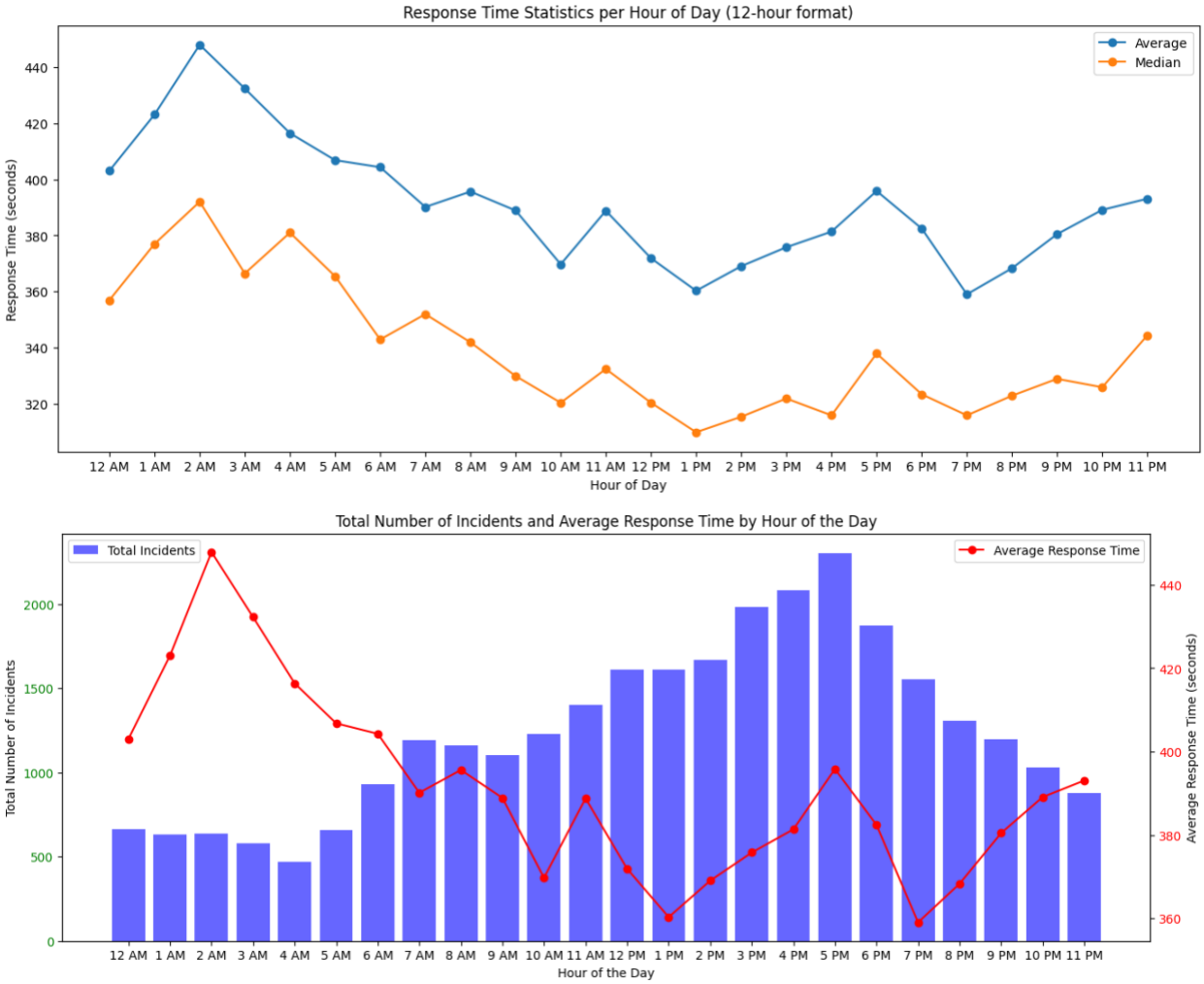
Here are some of the graphs we generated:



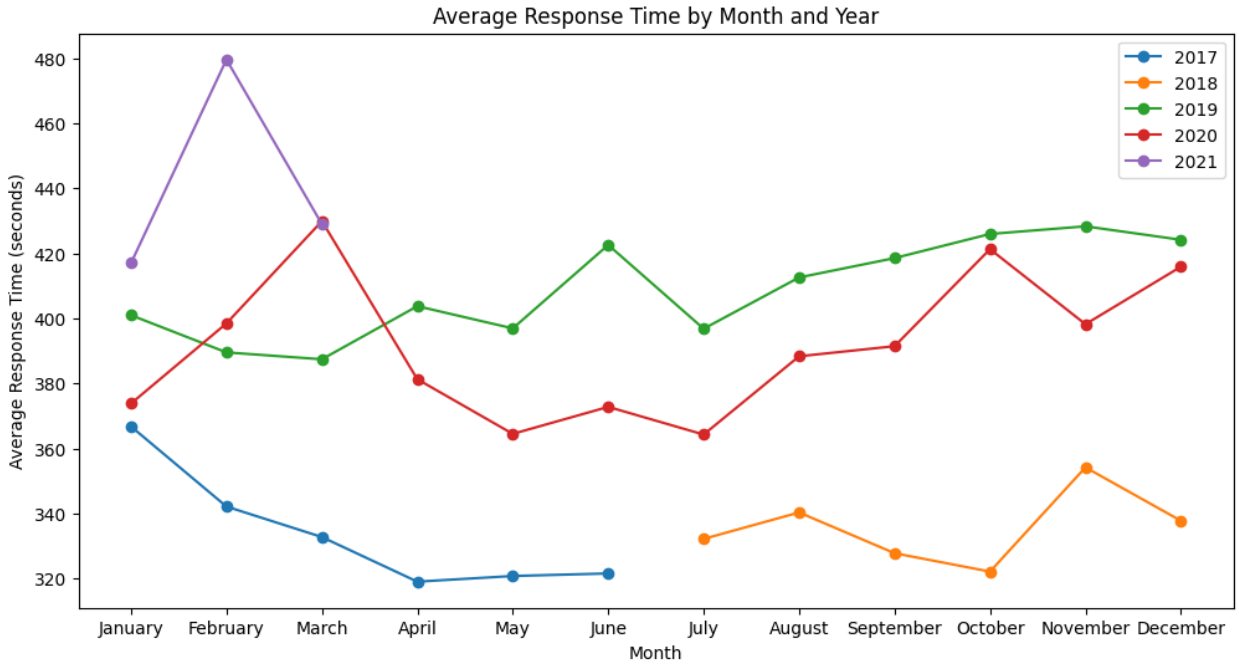
Total Incident and Response time year-by-year

Month of the year comparing incident frequency and average response time



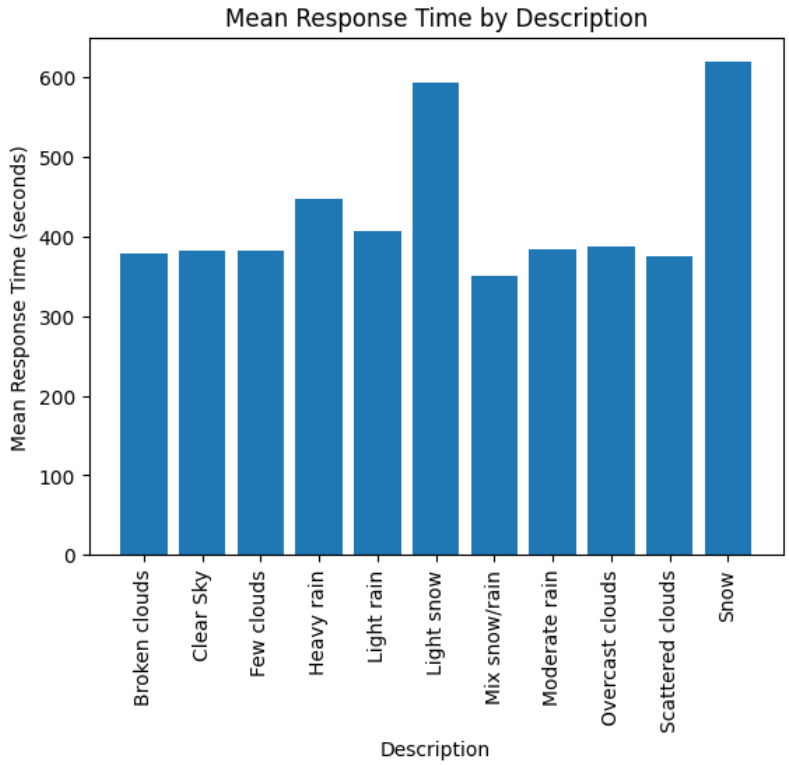


Time of Day and Average Response Time and Incident Frequency

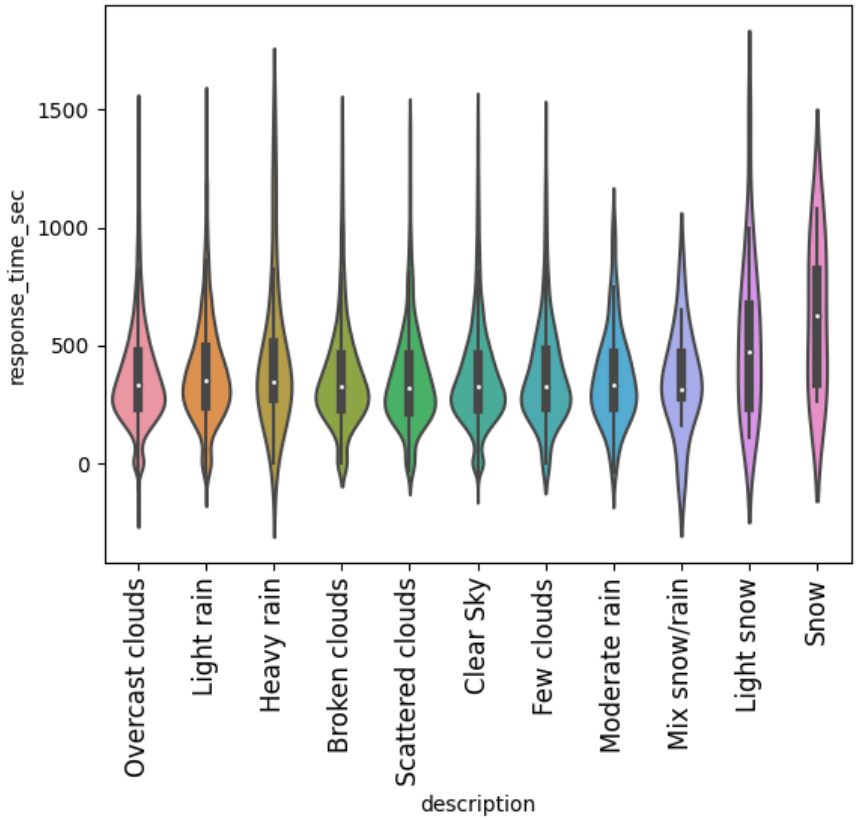


Weather and Response Time

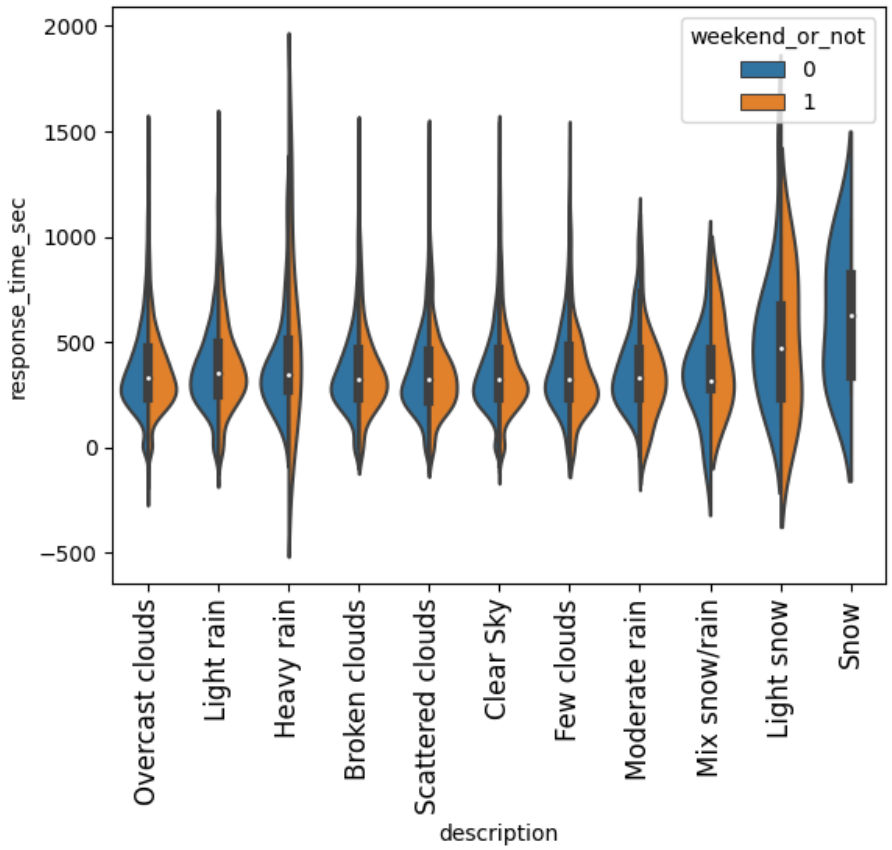
The weather data proved to be quite easy and interesting to visualize. It included a "description" string variable that categorizes the conditions into human-readable and well-known weather conditions.



In this figure, we see the categories of different weather conditions and their associated average response time. We see a global average for those with sufficient data at round 375 seconds. There's a significant slowdown in response time when there is heavy rain, light slow, or full snow in Nashville, all of which is in line with expected results. The mixed snow/rain category has a low average response time due to the low sample size here and we'd expect that to be an additional high response time condition.



Here we have what are known as violin plots, which allow one to show all the information of a boxplot along with the data distribution, which is especially advantageous for multi-modal data (Soni 2018). These were created after removing significant outliers for each category, providing a more zoomed-in view of the interesting data. They have the same categorical variable on the x-axis but now show the distribution of response times for each category. First, our suspicion of low data for mix snow and rain is confirmed along with for light and regular snow as these have a less normal distribution due to the lower amounts of data. Secondly, we see the high variability for response time in the more severe conditions like snow and heavy rain while most of the other categories will peter out after 1500 seconds. Finally, the distributions between scattered clouds, clear skies, and few clouds all seem to mirror one another, which shows little relationship between any of these categories and the response time for that day.



This is an additional violin plot for the same weather categories as above but they are split by whether the accident occurred on a weekend or weekday to compare if there's a particular intersection of weather and time of day that is particularly troublesome for authorities to handle. The weekend data tend to suffer from fewer incidents so there aren't too many meaningful conclusions to draw aside from a slightly worse weekend performance for moderate rain and a large variable for heavy rain weekend performance.

The final insights were drawn from using machine learning to predict response time from the numerical weather variables, an encoding of the string description, and the location of the incident. First, we used a Random Forest Regression for the time prediction, resulting in an RMSE of 157.60 seconds and R^2 value of 0.048. This means our model is doing a relatively decent job of predicting response times with the RMSE within 3 minutes but doesn't capture all the factors that determine response time. We then used a Gradient Boosting Regressor with the same splits, generating an RMSE of 54191 seconds and R^2 value of 0.096. This model did better at capturing the variation in the response time but significantly worse in predicting the actual response time. Finally, we looked at the variable importance for each model and found latitude/longitude of the accident was by far the most important factor (~0.3 combined for random forest, 0.7 combined for gradient boosted regression), followed by the time of day (0.07 for random forest, 0.13 for gradient boosted regression). This points to the time of day and location of the accident being more predictive of response time than weather is.

Conclusion

In conclusion, our project aimed to use big data tools and methodologies to analyze accidents in Tennessee and investigate the relationship between response time and the location, demographics of the location of the accident, time of the incident, and weather at the time of the accident. Our primary dataset was the incidents dataset. We joined it with various other datasets to investigate relationships with other data and plot our findings on a map of Nashville. We joined with census tract geometric info, demographic information and weather info.

We used AWS EMR and Spark to process the large weather dataset and join it with our incidents based on time. We also used EMR and Spark to filter the joined dataset to only contain weather data from the closest weather station to the accident. We then used this filtered table to investigate the relationship between weather and response time. We generated graphs and violin plots detailing these relationships.

We used Geopandas to join the census tract geometry data with our incident dataset spatially. Then, we joined that new dataframe with demographic data on the census tract column. Using the geometry data, we were able to plot information about response time and number of incidents per census tract on a map of nashville. We were also able to plot data about the demographic information as well. Finally, we did a simple linear regression to predict response time with demographic info.

For the temporal info, we grouped by time of day and by month to investigate how they might be associated with accident frequency/response time variance. We produced graphs to demonstrate these associations.

Bibliography

Pettet, G., Nannapaneni, S., Stadnick, B., Dubey, A., & Biswas, G. (2017, August). Incident analysis and prediction using clustering and bayesian network. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 1-8). IEEE.

Soni, A. (2018). Violin plots explained. Towards Data Science. https://towardsdatascience.com/violin-plots-explained-fb1d115e023d