Examining Weather, Traffic, and Incident data in Nashville, TN

Zakariyya Al-Quran
Vanderbilt University
Nashville, TN
zakariyya.s.al-quran@vanderbilt.edu

Rithik Reddy
Vanderbilt University
Nashville, TN
rithik.g.reddy@vanderbilt.edu

Mohamed Sufiyaan
Vanderbilt University
Nashville, TN
mohamed.sufiyaan@vanderbilt.edu

***Abstract*— There are a large number of variables that can contribute to an incident occurring. Ideally, we want to minimize the likelihood of an incident occurring, or at the very least respond fast and effectively. The best way to do this is to examine the data from the past on incidents and the many cofactors. In this project, four datasets (traffic, incidents, weather, and roadways) were examined to better understand what contributes to or correlates to incidents and response time in Davidson County. Using S3, Athena, and Spark EMR to store, transform, and query the datasets, we were able to find many temporal and spatial trends. The results were used to generate visuals. A machine learning component was also performed to estimate the number of incidents occurring. This data can be used to make better policy decisions and improve public safety depending on time, location, or weather.**

Keywords—Big Data, Traffic, Weather, Incident Response, Davidson County

I. INTRODUCTION

Incident response is a challenge faced by cities all over the world. Globally, about 3,200 people die from road accidents alone daily, leading to a total of 1.25 million deaths annually [1]. Furthermore, accidents can cause increasing traffic and congestion which may lead to more accidents. It would be ideal if we could minimize the rate of incidents occurring. In the case that an incident does occur, it is critical for the emergency response to be as efficient and possible. To do so, the emergency response management units (defined as "the set of procedures and tools that first responders use to deal with incidents" [7]) must be proactive. We can use data of past incidents to learn how to minimize and to best respond to incidents. The goal of this project is to analyze the reported incidents data in Nashville, TN and the relationship with response time, location, traffic, and weather.

There were four main datasets provided: Traffic, Weather, Incidents, and Roadways. The traffic dataset provides the average speed, congestion, and other properties of every road across Davidson County over the last five years in five-minute intervals. The roadway dataset provides details about each main road including its geometry, major connections, minor roads, length, etc. The traffic and roadway sets had to be used in union. The Weather data is simply a record of the weather at some time for stations in the state of Tennessee. The entries from stations in the Greater Nashville area were examined. Finally, the incidents dataset was most essential for this project. It provides information on all reported emergency response incidents in Davidson County including the type of incident, date & time, and nearest road.

Altogether, the datasets can be used to find trends and correlations. For example, it would be useful to know how much weather affects traffic and the probability of an incident occurring on some major roadway. In that case, the speed limit can be adjusted accordingly to prevent an accident from occurring in the future based on the weather forecast. Analyzing data can be used to make better decisions regarding policy and responding to emergencies. Data-driven models can help minimize loss and improve safety measures across the state of Tennessee and beyond [4]. Each dataset alone is valuable and can be examined on its own, but they are even more valuable when merged together. Ultimately, we can perform spatial and temporal analysis on the incident data along with the other data to find trends between these variables and response time that will help us better understand the problem.

II. EXPERIMENTAL DESIGN

The specific technologies used to manage the data in this project are described below. A GitHub repository created and used during the project to share the code used between group members. The final results and project submission are stored in the repository.

A. Downloading and Storing Data

The datasets were all provided through Microsoft Teams SharePoint. Due to their relatively large sizes, the data sets were downloaded to a local machine before being uploaded to an AWS S3 bucket. Amazon's Simple Storage Service (S3) is "an object storage service offering industry-leading scalability, data availability, security, and performance" [2]. It is the best option to store data for this project due to its low cost, extreme durability, and robust services available. This allows the data to be accessed anywhere with an internet connection and the AWS proper credentials. The process of creating the bucket and uploading the data is simple. A private bucket was created (named "vandy-bigdata-finalproject") with the default configurations and the data was uploaded into a directory using the AWS S3 console. The same S3 bucket holding the original datasets was also used to store the merged data and results of the queries run.

Data stored on S3 can be queried using AWS Athena [3] which supports standard SQL. The only extra steps were creating a table for each dataset on S3 using its given schema. Athena can be used to analyze many types of data including CSV, JSON, or Parquet. Since the datasets were a combination of these storage types, Athena is ideal for merging the sets together. Athena executes queries in parallel very quickly, so performance was more than sufficient. Athena is a serverless interactive query service, meaning we do not have to manage the infrastructure behind the querying engine, and we can start querying instantly. We only have to pay for the queries we run based on the amount of data scanned. This meant we had to be careful when defining our queries as to not be too general. Also, there exists pyAthena, a Python library that allows a user to create queries using Python, simplifying our project since we can put all our code in a Jupyter notebook on GitHub. Athena server access logs were stored as objects in the same S3 bucket. The results of the most important queries were downloaded locally and uploaded to a directory in the project repository.

B. *Tranforming the Data*

Prior to merging, we sought to perform transformations to our data for more nuanced primary keys for joins (i.e., 6-hour window columns) and more valuable feature extractions (i.e., rush hour, month, and day columns). To undergo this feat, we utilized AWS EMR (“Elastic Map-Reduce”), a managed cluster platform with Big Data framework compatibility used for large-volume processing. In particular, the frameworks and softwares we configured with EMR were Hadoop (for storing and processing the datasets), JupyterEnterpriseGateway (for Jupyter Notebooks running on our EMR cluster), and most notably, Spark (for the RDD transformations). We deployed our EMR cluster with a master node (which runs the YARN Resource Manager) and several core nodes (which coordinate data storage through the Hadoop Distributed File System) and imported various transformation utilities from the pySpark API (the Python driver Spark), such as SparkSession, SQLfunction, and SQLtypes.

C. *Merging and Quering the Data*

Given the relational nature of our data, SQL was the natural language of choice for our querying. After assessing the magnitude and complexity of our datasets, we decided to utilize AWS Athena, an interactive SQL querying service that integrates seamlessly with AWS S3 where our data is stored. Using the S3 locations, database schema, and partitioning format, we can load up our data into tables and directly query via the Athena console. For some queries, we also leveraged the pyAthena library through Google Colaboratory. However, the majority of our queries resided on the Athena console due to factors like size, complexity, and convenience. For intermediate queries we wanted to store (e.g., Top 20 roadways, Grid-Partitioned Subsets, etc.), Athena provided an interface for creating and naming subsequent tables/views from our queries. For data output, Athena provided an option to download results locally in csv format.

D. *Post Processing and Visualizations*

The resultant merged data CSV file was small enough to be loaded into memory using Pandas. So, the data was read into a Pandas DataFrame on a Colab notebook and went through simple post processing. Then, the processed merged data was uploaded onto S3 again for the rest of the queries as described above. Those queries generated CSV files which were loaded into memory in a local Jupyter notebook for generating visuals. The result CSV files were processed further if necessary — for example, adding a differentiating column when concatenating the results of multiple queries — and plotted using Plotly. The documentation for Plotly [6] was examined to generate different types of visuals . The type of plot used depended on the query. For example, the geospatial data was best visualized with a heatmap or scatter map. Line graphs were ideal for temporal data. Also, variables that were suspected of being related (such as speed and congestion) were plotted together. All of the plots were saved to the project repository. The most relevant plots are shown in the Results and Visuals section.

E. *Predicting Accident Occurance With Machine Learning*

For the machine learning aspect, we wanted to have some way to predict the number of accidents for the roadways across all of Davidson County. We partitioned the data using the same grid as the geo-partitioned queries (shown in Fig. 1). For each grid “square”, we decided to group by month, so we could predict the number of accidents for the next month. We then decided to use the features: average(speed difference) (defined as speed – reference speed), average precipitation, average temperature, average visibility, average wind speed, and then month and year as the features for our model. Three models were then trained, a linear regression model, a random forest regression model, and a gradient boosted tree regression model. The RMSE (Root Mean Square Errors) of the model were then calculated, and then the model with the lowest RMSE was chosen. Then the prediction for that model was then reported.

III. Queries

Queries were used to merge the data and find possible trends based on traffic, weather, location, and time. Related queries were grouped and run together.

A. *Queries to Merge the Datasets*

We were supplied with various datasets with differing relationships from one another. Therefore, we had to devise informative and qualitative queries to join all four datasets into one merged table. We leveraged Spark for RDD transformations to devise columns for window IDs (i.e., a string corresponding to the 6-hour window of a particular datum, containing the format “X.YYY.ZZZZ”, where X is the window number, Y is the Day number, and Z is the Year) for each incident, traffic, and weather data reported. We joined the traffic and incident datasets by the primary key XDSegID (to line up traffic information for the segment closest to a particular accident) and window ID. Prior to merging this with weather data, we ensured that we narrowed down the weather corpus to contain only information from the 5 weather stations within Nashville (e.g., KBNA, KJWN, etc.), and averaged all the weather statistics for each 6-hour window. Then, we joined our weather dataset with traffic and incidents by XDSegID and window ID. Finally, we merged this dataset with the Road data by primary key SegID in order to later classify the cluster of roads (road group) corresponding to the XDSegID of the incident in question.

B. *Queries to Generate Historical Trends for Nashville's 20 Major Roadways*

In identifying daily, monthly, and yearly historical trends for the incident response data, we first had to identify the 20 major roadways in the greater Nashville area. As such, we prioritized the road groups with the highest incident frequency, then by their congestion. After gathering these roadways, we performed queries on this subset to gather incident statistics (i.e., incident frequency, average response time, and average speed) grouped by various time intervals (i.e., month, hour, 6-hour window, day of week, day of year, and year). The intuition behind these queries was to gauge information on the idiosyncrasies of the incident statistics, identify time intervals of high and low incident densities, and provide an outlook on the historical trends in order to provide further insight towards future analysis. Furthermore, we also extracted the bottom 20 roadways, which we identified as the road groups with the lowest congestions (but still had some measure of incident frequencies) and used similar queries in order to provide a contrast of the different ranges of roadways in Nashville. The main query for this is shown below:

```
SELECT xdgroup, COUNT(DISTINCT id_original) as cnt,
AVG(congestion) as cong
FROM "incidents"."merge" WHERE xdgroup != 0
GROUP BY xdgroup
ORDER BY cnt DESC, cong DESC
LIMIT 20;
```

The query above gets the 20 major roadways using the merged dataset. Note that the top 20 roadways are defined as the main road groups with the most incidents and congestion.

C. *Queries to Generate Correlation between Incident Trends and Weather Patterns*

In generating correlations between incident response data and local weather patterns, we had to first average all relevant weather information from all 5 stations (across 6-hour windows) for each accident that took place. This way, we would be able to provide a more holistic weather outlook for that incident (with respect to all of Nashville within a particular segment of time). We chose temperature, precipitation, and visibility as our weather patterns of interest, and applied a query to allocate every incident into a specified range for each of these parameters (see figure 1 below).

Temperature Range

-1	-10 °C to 0 °C
0	0 °C to 10 °C
1	10 °C to 20 °C
2	20 °C to 30 °C
3	30 °C +

Precipitation Range (mm)

0	0 to 0.05
1	0.05 to 0.1
2	0.1 to 0.15
3	0.15 to 0.2
4	0.25 to 0.3
5+	0.3+

Visibility Range (km)

0	0 to 2	5	10 to 12
1	2 to 4	6	12 to 14
2	4 to 6	7	14 to 16
3	6 to 8	8	16 to 18
4	8 to 10		

Fig. 1 Temperature, precipitation, and visibility ranges.

We further performed queries to gather incident statistics (i.e., incident frequency, average response time, average congestion, and average speed) grouped by the aforementioned ranges (temperature, precipitation, and visibility). The intuition behind these queries was to assess the relevance and correlative impact of certain weather patterns to accidents in Nashville and use these insights towards further analysis. The sample of query for this group is shown below:

```
SELECT COUNT(incident_id) AS num_incident, temp_range,
AVG(avg_response) AS avg_response, AVG(avg_speed) AS
avg_speed, AVG(avg_congestion) AS avg_congestion FROM
"incidents"."weather_avg";
```

The query above gets the unique incident frequency, average response time, average speed, and average congestion for each temperature range.

D. *Queries to Generate Geo-Partitioned Incident Occurrences in the Greater Nashville Area*

In gathering a more robust cognizance of the incident response data across different regions of Nashville, we had to generate geo-partitioned clusters encompassing our entire dataset. As such, we partitioned the greater Nashville area into 16 equally sized partitions (see figure below), as well as some particular areas of interest (i.e., Vanderbilt Campus, Downtown Nashville, East Nashville, and BNA International Airport), and performed further queries and analysis on these subsets of data. In particular, we performed queries to gather information on the incident frequencies for these areas of interest grouped by various time intervals (i.e., month, hour, 6-hour window, and year). The partitions were used in further ML analysis.

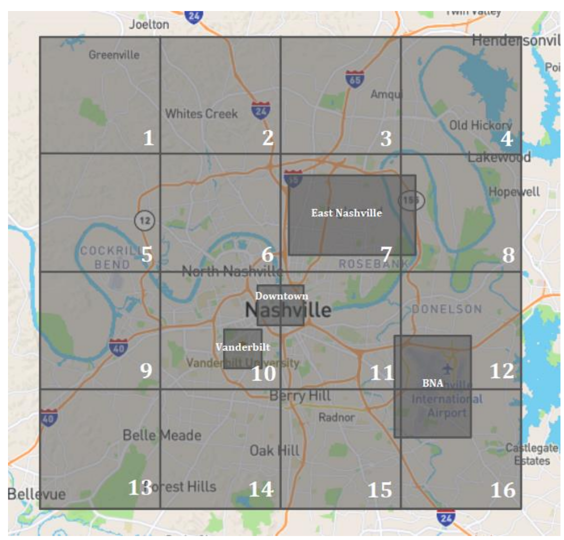

Fig. 2 Geo-partitioned map of Nashville.

IV. RESULTS & VISUALS

In all, over fifty queries were run across all the datasets. The queries were grouped as described above. The repository contains the complete set of the results and visuals. The most significant visuals from each group of queries are shown are described below. Note that the first few plots are shown on the next page and the remaining are in the appendix.

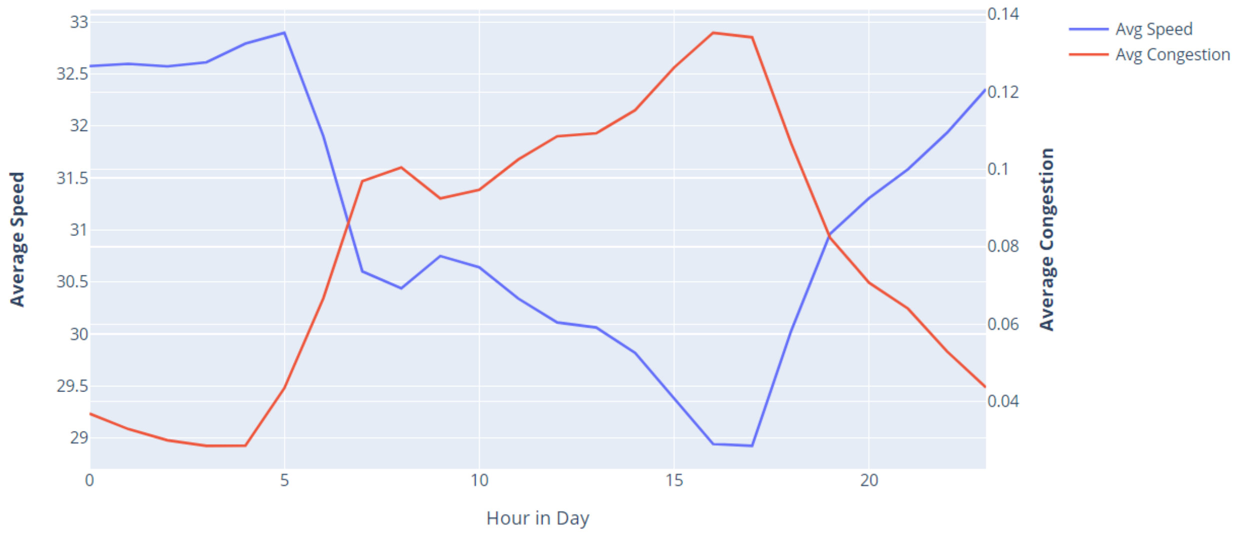

Fig. 3 Average Congestion and Speed per Hour in Davidson County

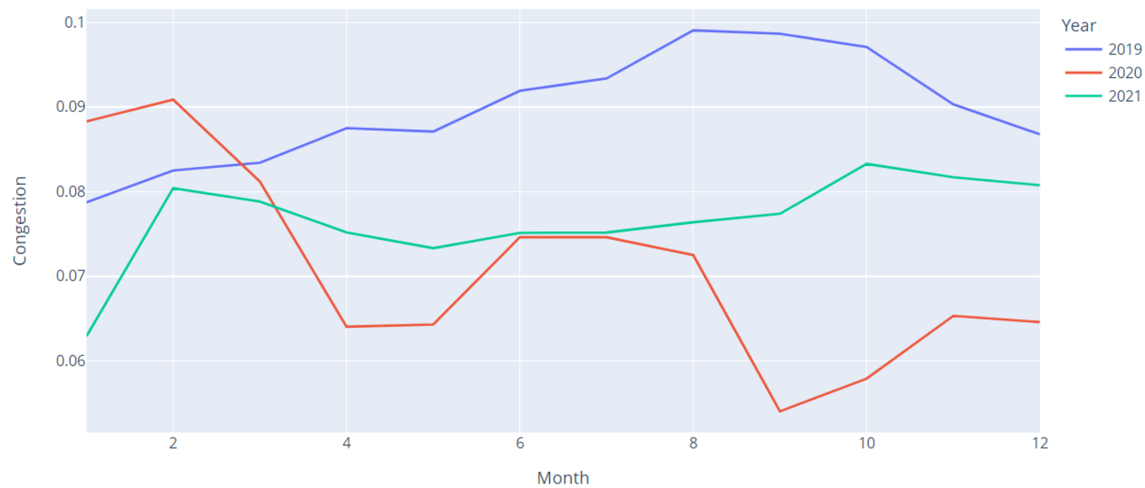

Fig. 4 Average Congestion and Speed per Hour in Davidson County

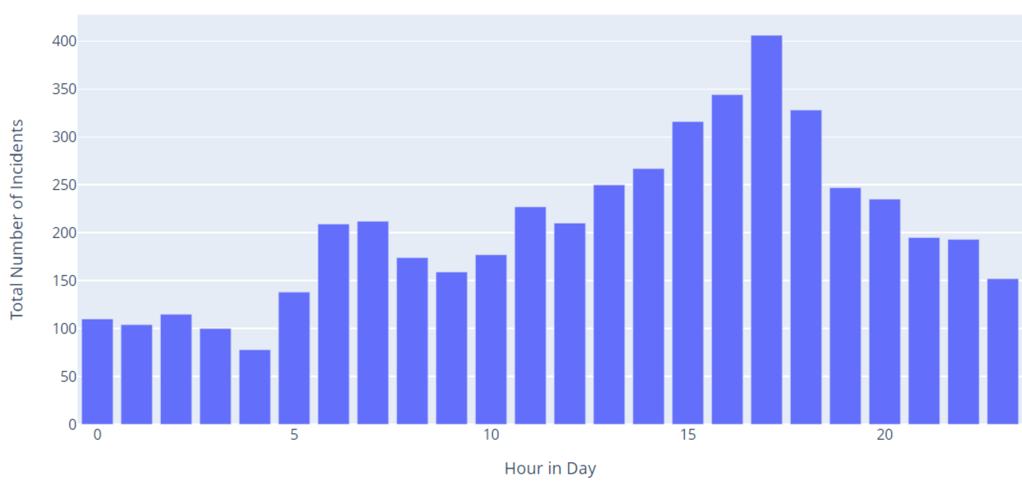

Fig. 5 Incidents per Hour in Top 20 Roads in Davidson County

A. Unmerged Traffic Plots

1) The average congestion and speed across all roads in Davidson County per hour is shown in Fig. 3 above.

2) The average congestion per month in Davidson County for the years 2019 through 2021 are shown in Fig. 4 above.

B. Merged T20 Trends

1) Fig. 5 above shows the distribution of all incidents by hour on the top 20 roadways.

2) The average speed and response time distribution per hour on the top 20 roadways is shown in Fig. 6.

3) Fig. 7 compares the total number of incidents per day on the top 20 roadways for 2019 vs 2020.

4) The number of incidents per window on the top 20 roadways in 2019 vs 2020 is shown in Fig. 8.

5) The average speed per hour on top 20 roads for 2019 vs 2020 is shown in Fig. 9.

C. Merged Weather trends

1) Fig. 10 shows the distribution of incidents by precipitation range.

2) Figures 11 and 12 show the response time and incidents by visibility range respectively.

3) Figures 13 and 14 show the response time and incidents by temp range respectively.

D. Merged Geolocation trends

1) Fig. 15 is a heatmap of reported incidents in Nashville.

2) Fig. 16 is a scatter map of the response time for each reported incident in Nashville, TN.

3) The distribution of incidents in the BNA area by month is shown in Fig. 17.

4) The distribution of incidents in the Vanderbilt area by month is shown in Fig. 18.

5) The distribution of incidents in the downtown Nashville area by hour is shown in Fig. 19.

E. Machine Learning Regressors

Tables 1 and 2 in the appendix show the predicted number of incidents for each grid area or area of interest based on the regression models. The bolded values signify the regressor with the lowest RMSE.

V. Discussion

A. Outcomes of Queries

After performing the queries and generating visualizations of the results, it became clear that there were trends across the datasets with many implications. The major takeaways are summarized below.

1) Time Analysis

It makes sense that congestion and speed across a roadway will depend on the number of cars traveling which will vary by the time of day, the day of the week, the month of the year, etc. From Fig. 4, it is clear that there is an inverse relationship between average congestion and speed; as the average congestion increases, speed decreases. The plot also shows what hour of the day roads are least and most congested.

Additionally, it is interesting to see the effects of the COVID-19 pandemic on traffic and incidents. According to Fig. 7, COVID did not greatly affect the total number of incidents on Nashville roadways as compared to the year before. Travel patterns also didn't appear to change as the trend for incidents by window in Fig. 8 is largely unchanged. The pandemic did appear to affect congestion though. 2019 had the most congestion to date, increasing month to month through 2020. However, in March of 2020, the average congestion in Davidson County significantly declined, presumably due to the first lockdowns at the emergence of the COVID-19 pandemic. The congestion recovered slightly in the summer but declined again in the fall, possibly due to lockdowns being reinstated when there was a rise in cases. As a result of the lower congestion, the average speed per hour across the top 20 roadways was higher in 2020 than in 2019.

Some other trends noted were that most accidents occur around 5 PM (considered "rush hour") when most people are driving home from work or activities. Conversely, the least number of incidents occur at 6 AM, likely due to the few cars on the roads. Weekends were evidently the busiest days to drive and most accident prone.

2) Weather Analysis

The weather analysis was done on temperature, visibility, and precipitation. Most accidents happened in the 20-30°C range which is the most common temperature in Nashville throughout the year. Fig. 13 hints at response time being higher in the lowest temperature range however, possibly due to inclement road conditions such as snow.

Counterintuitively, visibility did not appear to have a significant effect on incident occurrence or response time. More incidents did occur at higher visibility, but Nashville has good visibility year-round, so more incidents are bound to occur in higher ranges. Drivers may also drive more carefully when visibility is low which leads to higher congestion and lower average speed, lowering the probability of an accident occurring.

As for precipitation, response time was found to be inversely related. Also, precipitation was directly proportional to temperature and visibility (since it rains more in the summer when the temperature is higher and there is good visibility).

3) Location-Specific Analysis

Areas of interest were examined separately from the grid of Nashville. Based on the spatial analysis, certain areas in Nashville are more prone to incidents and at different times.

For example, Downtown Nashville is a quarter of the area of East Nashville but has twice as many incidents in the same time period. In the Vanderbilt area, the first months of the fall semester have the highest incident frequency while the summer months have the lowest. During summer winter vacation months, the BNA airport area suffers most of its accidents. As for the downtown Nashville area, the afternoon hours of 12 to 2 PM and the night after 7pm are when most accidents occur. This can be attributed to people going out to lunch in the afternoon and people going out at night.

B. Machine Learning

We were able to predict the number of accidents segmented into different grids across Davidson County. As we can see from the results, in the vast majority of the grids, the regression model which performed the best was Random Forest Regression. It makes sense why this produced a better approximation than the normal linear regression in most cases, as it can generate more rules with the data as opposed to just coming up with a linear model.

Furthermore, there are many avenues for future work and improvement. The clusters were very rudimentary, as we just split the city up into 16 grids and 4 areas of interest. We computed the number of accidents in those areas, but in the future, we could cluster by something more specific and useful, such as different roadways. Aggregating results by month can lead to a loss of a lot of granularities of the data being lost, we can aggregate with a smaller timescale, such as by week or by day. The features to be selected could have been chosen better, for example by looking at correlations between the features and the number of accidents or having a variance threshold for a given feature. Also, we did not perform any cross validation on the data, so the models may not be as robust, and that is something to look to do in the future. Further, in Random Forest Regression and Gradient Boosted Tree Regression, we can look at the relative importance of the variables to see how much a variable dominated in predicting the number of accidents. Finally, different approaches could also be taken. For example, we can use a neural network or even convert the problem into one where we would find the probability of an accident occurring in a given time.

C. Challenges faced

The initial datasets provided were too large to load into Pandas. Thus, we had to find another way to access the data. Once we got data onto S3 and set up the Athena tables, the rest of the project was just applying the skills we learned in class.

After determining what technologies to use, there were no major complications. Our project flow was very straightforward. Adding columns was a minor difficulty at first since we were experiencing severe data loss when writing our updates back to a parquet file. After realizing that the Spark progress meter was stalling at the same task, we immediately identified the issue to be our configuration for the EMR cluster we deployed. To remedy this, we used three core nodes, instead of our original two, which allowed us to manage a larger volume of data.

Another minor issue faced was grouping by time. It was not ideal with Athena since we could only extract and group by discrete date parts such as hour or month, contrasting with a time series database such as InfluxDB which would have made it easier to plot all the relevant values. On the other hand, Athena gave us the flexibility to perform many different types of analysis and queries through one service.

With machine learning, we could have gone into varying degrees of detail, so coming up with a machine learning plan was difficult. However, we were able to come up with a sufficient plan and generate viable results in the end.

VI. Conclusion

Overall, the project was completed successfully. The datasets were all merged, queried, visualized, and analyzed. The data was stored in S3, transformed with Spark EMR, queried with AWS Athena, and visualized with Plotly. Machine learning regressions were also run using Spark EMR to generate estimates of incidents in different areas of Nashville. The correlations between different variables were interesting to view. The trends found and model generated can be used to better understand the factors contributing to an incident occurring and the response time. Thus, it is evident that data is very valuable and can be useful in improving the world.

Acknowledgment

We thank Dr. Dubey for teaching us the skills we needed and providing us with the data to carry out this project. We would also like to thank Pravesh for help throughout the semester and Geoff for his guidance relating to the project specifically.

Attestation

All members of this team worked on this report and the entire project together. The work was split up as evenly as possible and all members contributed to the best of our ability.

References

[1] Center of Disease Control and Prevention. Road traffic injuries and deaths — a global problem. https://www.cdc.gov/injury/features/global-road-safety/index.html, 2019.

[2] "S3," *Amazon*. [Online]. Available: https://aws.amazon.com/s3/. [Accessed: 02-May-2022].

[3] B. A. Hoena and L. Bowman, "Athena," *Amazon* [Online]. Available: https://aws.amazon.com/athena/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc. [Accessed: 02-May-2022].

[4] A. Mukhopadhyay, G. Pettet, S. M. Vazirizade, D. Lu, A. Jaimes, S. E. Said, H. Baroud, Y. Vorobeychik, M. Kochenderfer, and A. Dubey, "A review of incident prediction, resource allocation, and dispatch models for emergency management," *Accident Analysis & Prevention*, vol. 165, p. 106501, Nov. 2021.

[5] "Home," *NHTSA*. [Online]. Available: https://www.nhtsa.gov/. [Accessed: 02-May-2022].

[6] "Plotly Python," *Plotly Python Graphing Library*. [Online]. Available: https://plotly.com/python/. [Accessed: 02-May-2022].

[7] S. M. Vazirizade, A. Mukhopadhyay, G. Pettet, S. El Said, H. Baroud, and A. Dubey, "Learning incident prediction models over large geographical areas for emergency response," *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, Jun. 2021.

APPENDIX

TABLE I. THE ESTIMATED NUMBER OF ACCIDENTS FOR EACH SUBGRID

Area of grid	**Linear Regression**	**Random Forest Regression**	**Gradient Boosted Tree Regression (maxIter = 10)**	**Lowest RMSE Regressor's Prediction**
Q1	0.5369	0.8814	**0.5**	1
Q2	**1.6988**	1.83868	1.7388	5.33
Q3	10.914	**5.7268**	11.187	24.2
Q4	**2.499**	2.848	2.965	6.49
Q5	1.929	**1.456**	1.732	2.5
Q6	13.559	**5.552**	8.503	20.6
Q7	19.96	**7.439**	10.583	40.95
Q8	6.333	**2.327**	2.979	7.69
Q9	2.554	2.2238	**2.0288**	9.979
Q10	**15.542**	18.637	16.219	57.18
Q11	28.667	**27.069**	33.4068	69.0
Q12	12.92	7.535	**3.928**	26.423
Q13	2.814	**2.798**	3.559	9.26
Q14	4.523	**2.853**	4.391	7.261
Q15	10.50	**9.419**	15.38	41.63
Q16	14.14	**7.619**	15.049	34.34

TABLE II. THE ESTIMATED NUMBER OF ACCIDENTS FOR EACH AREA OF INTERST

Area	**Linear Regression**	**Random Forest Regression**	**Gradient Boosted Tree Regression (maxIter = 10)**	**Lowest RMSE Regressor's Prediction (number of accidents)**
Vanderbilt	5.933	**2.114**	5.740	5.365
BNA Airport	7.809	**5.949**	9.389	14.57
East Nashville	11.49	**5.482**	8.290	15.021
Downtown	17.05	13.838	**11.634**	29.72

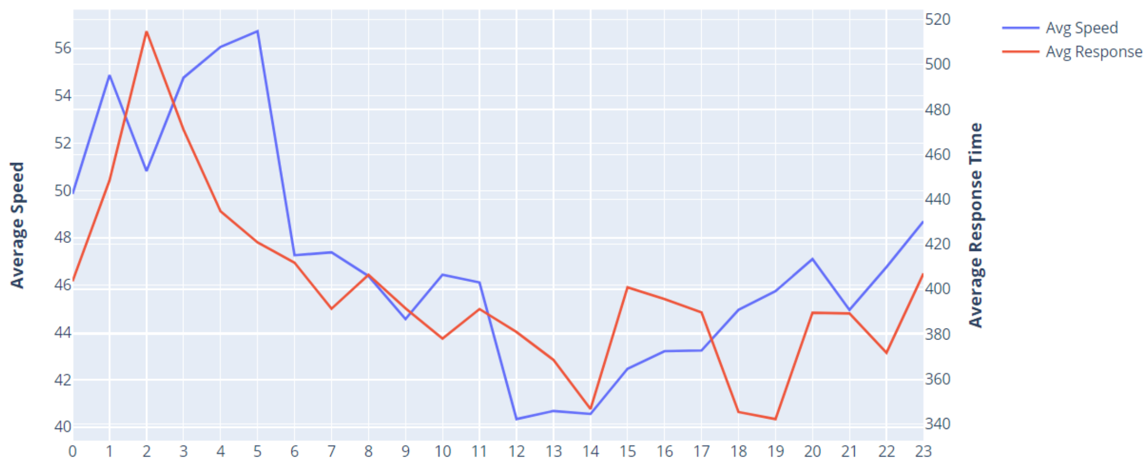

Fig. 6 Average Speed and Response Time per Hours in Top 20 Roadway in Davidson County

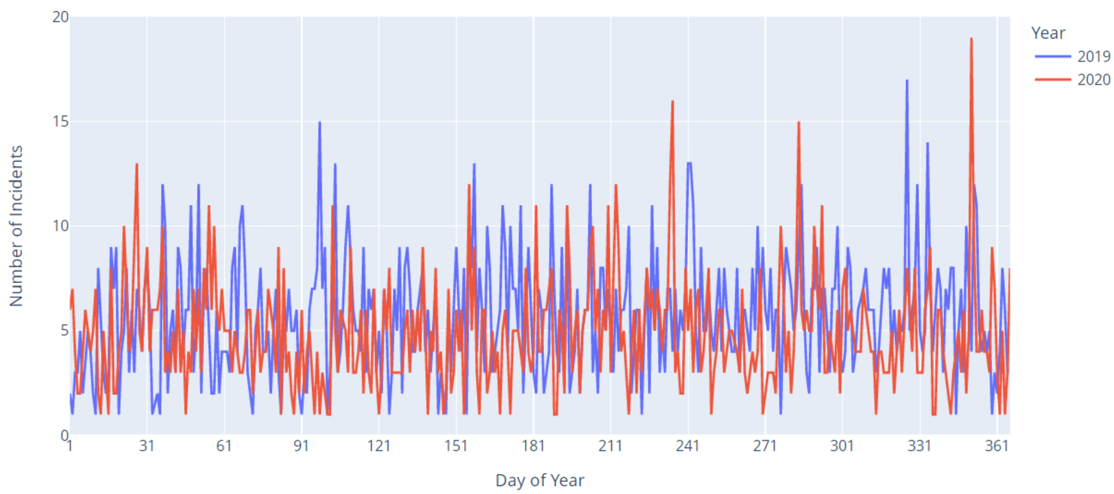

Fig. 7 Number of Incidents per day on Top 20 Roads in Davidson County in 2019 vs 2020

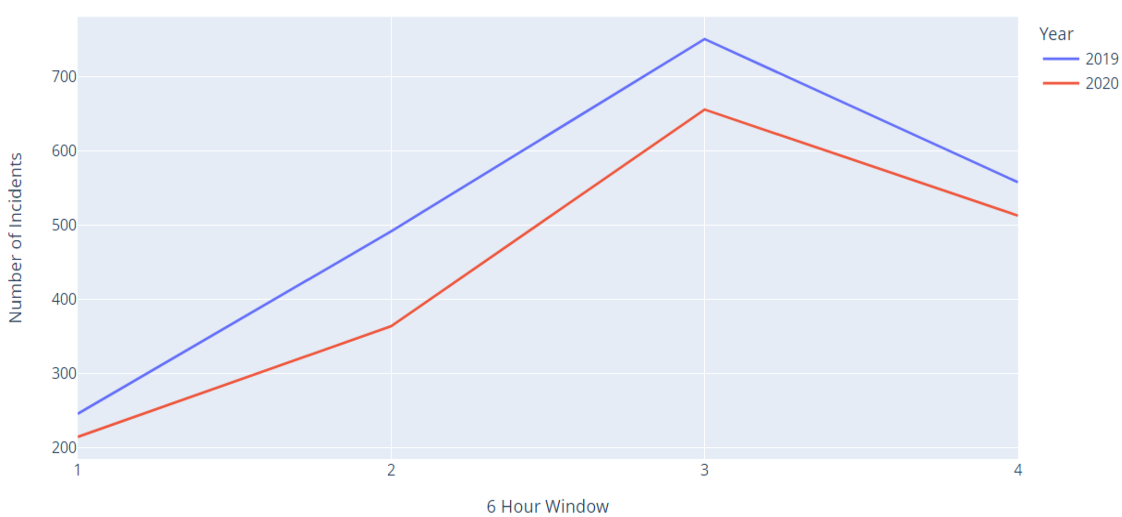

Fig. 8 Number of Incidents per 6 Hour Window on Top 20 Roads in 2019 vs 2020

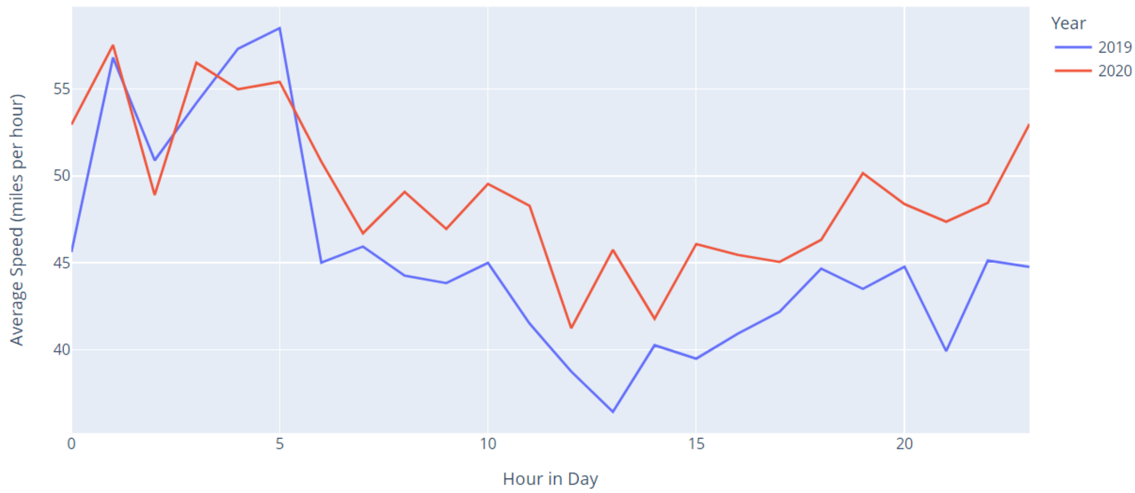

Fig. 9 Average Speed per Hour on Top 20 Roads in Davidson County in 2019 vs 2020

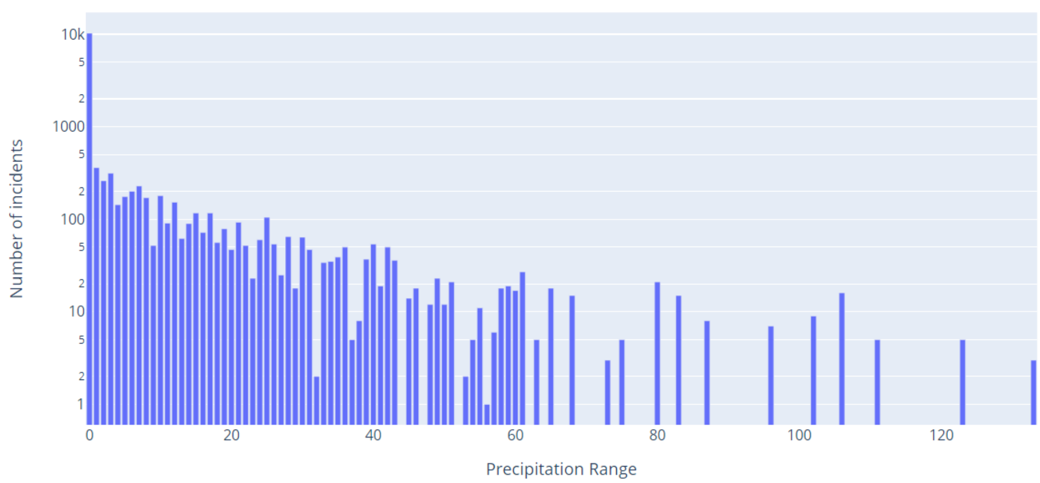

Fig. 10 Total Incidents by Precipitation Range in Davidson County

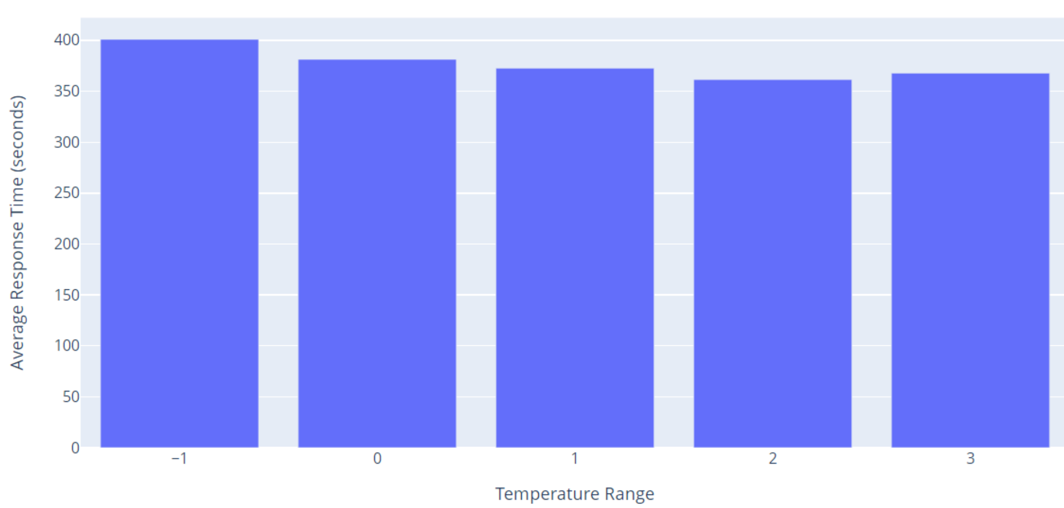

Fig. 11 Response Time by Visibility Range in Davidson County

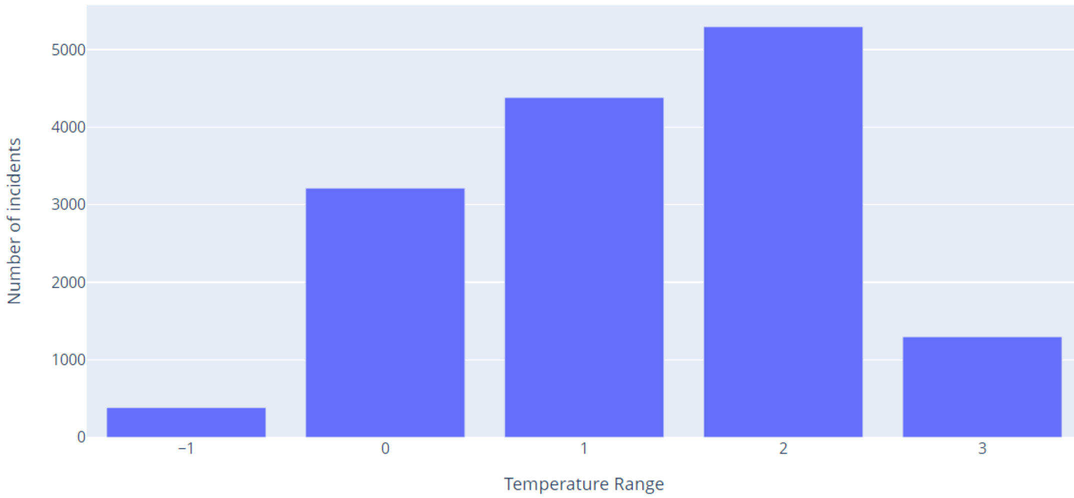

Fig. 12 Total Incidents by Temperature Range in Davidson County

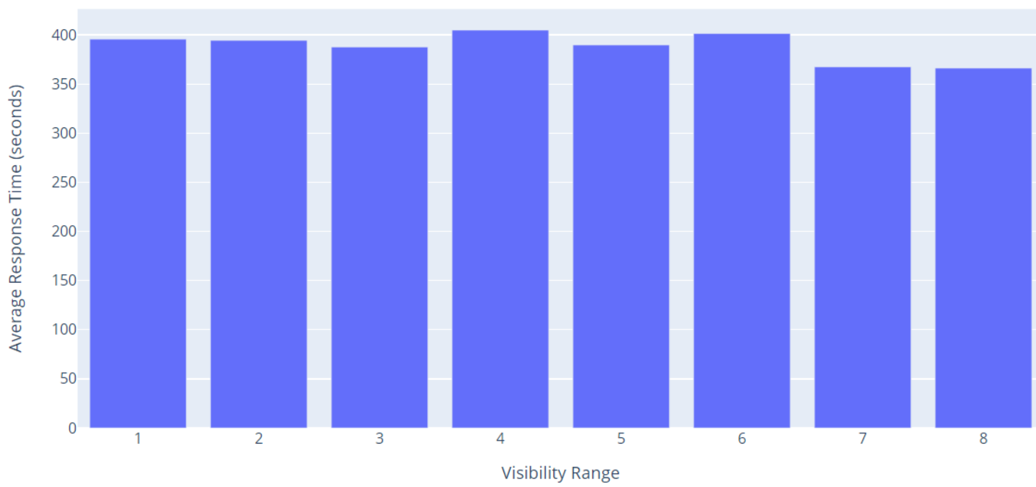

Fig. 13 Response Time by Visibility Range in Davidson County

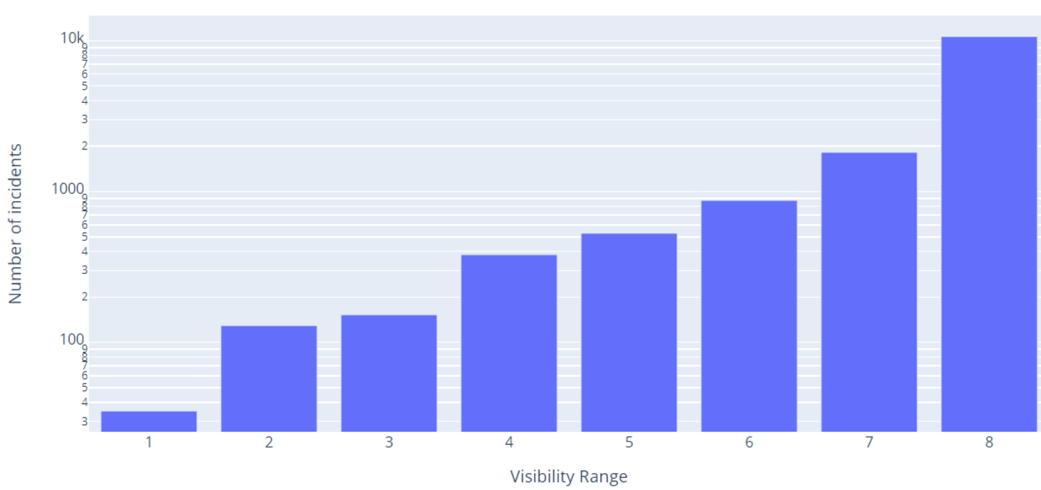

Fig. 14 Total Incidents by Visibility Range in Davidson County

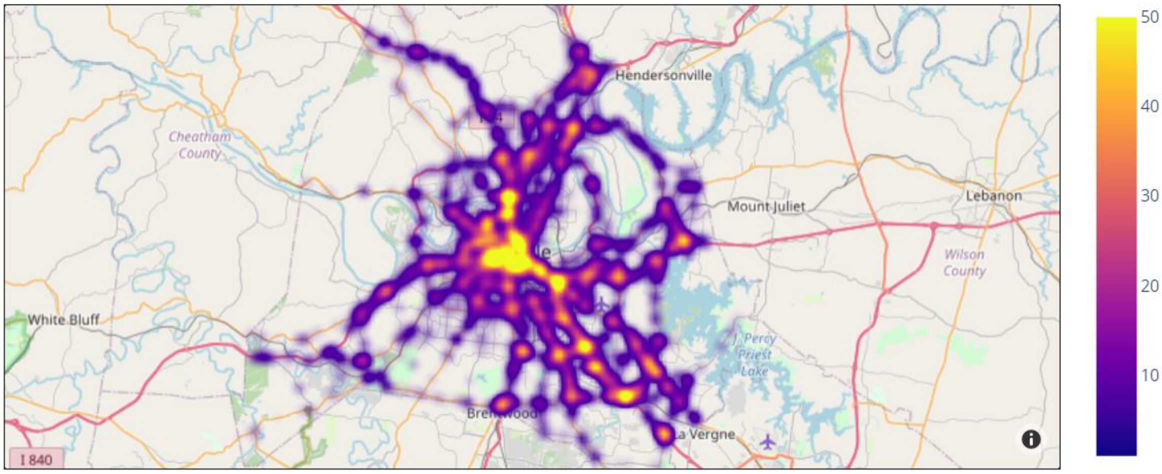

Fig. 15 Heatmap of Reported Incidents in Davidson County

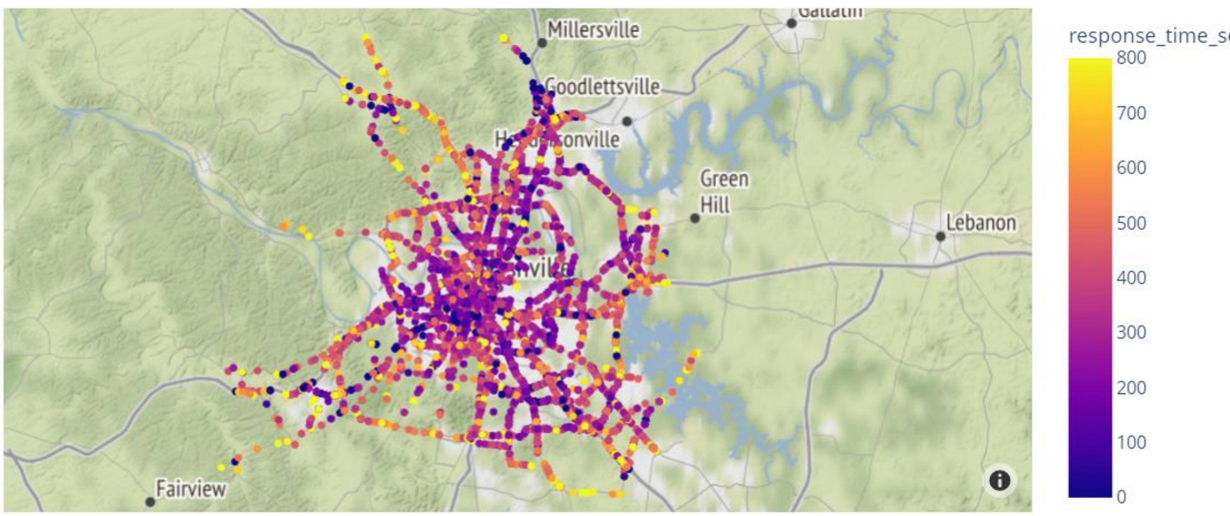

Fig. 16 Scatter Map of Incident Response Time in Davidson County

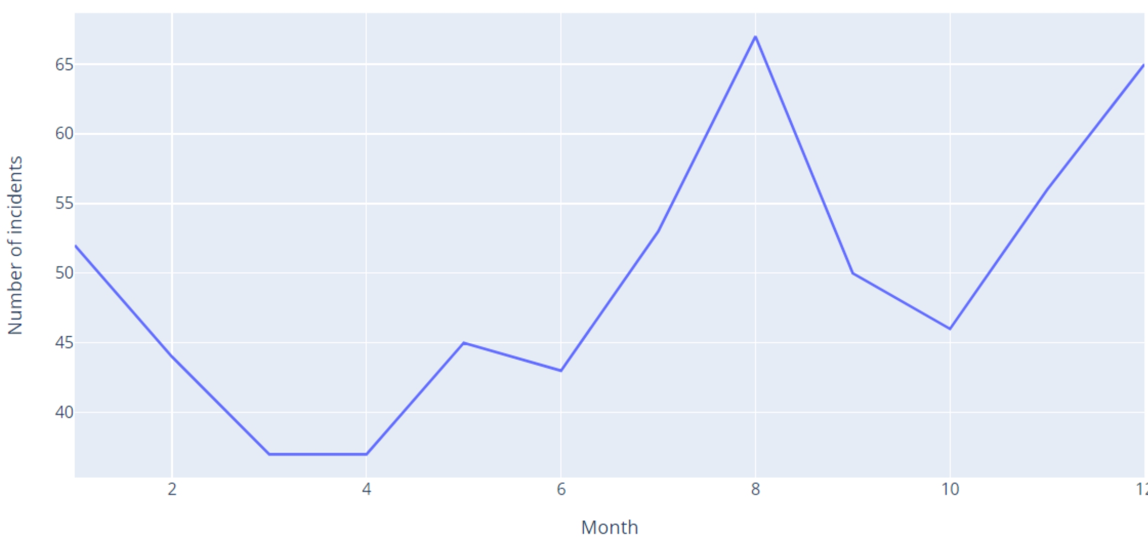

Fig. 17 Number of Incidents per Month in BNA Airport Area

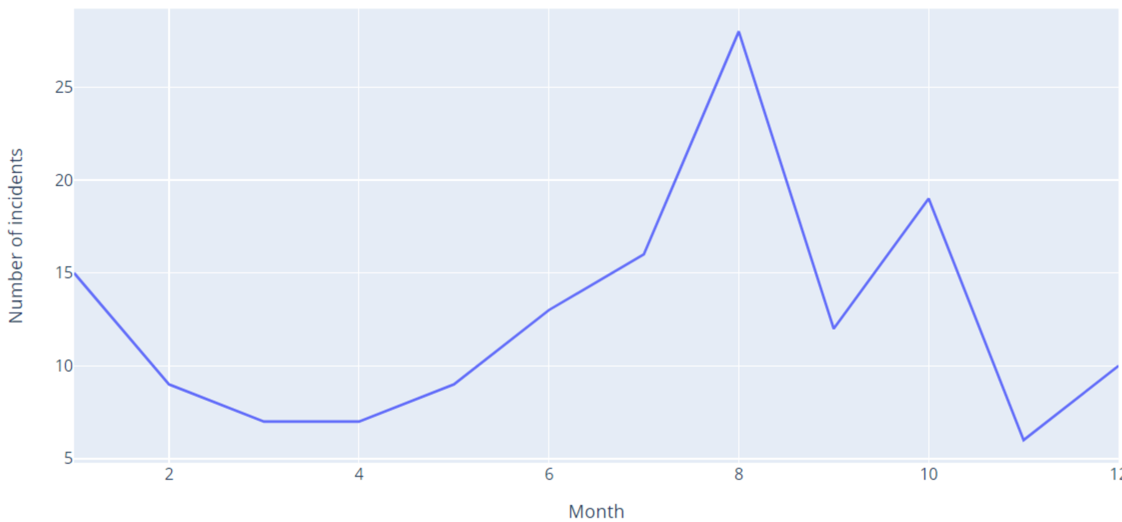

Fig. 18 Total Number of Incidents per Month in the Vanderbilt Area

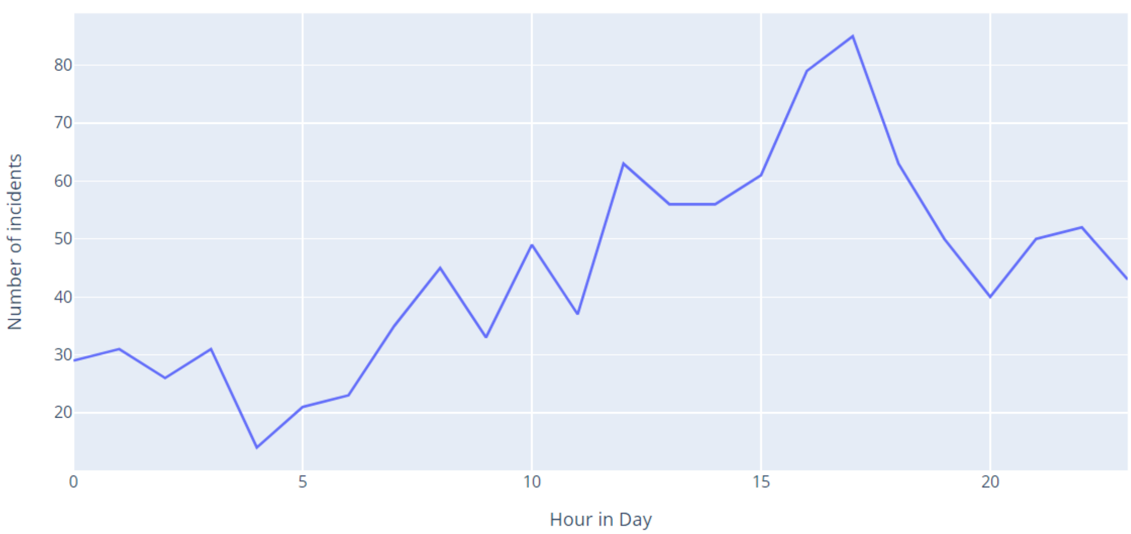

Fig. 19 Total Number of Incidents per Hour in the Downtown Nashville Area