Department of Computer Science

Missouri State University

CSC735 - Data Analytics

Traffic Accident Duration Prediction Using Regression Models

Project Report

Submitted by

Sujung Choi

Submitted on December 7, 2023

Abstract

Prediction models for traffic accident clearance duration can be helpful in practice because by finding the underlying factors, those factors can be considered for transportation policies to improve highway safety [1]. The purpose of this project is to identify relevant features that can potentially explain the duration of traffic accident clearance and build models for predicting traffic accident duration. Three types of regression models -DTR, RF, and GBT- were utilized to train on historical data and to predict the duration of traffic accident clearance on unseen data. To compare the performance of models, evaluations were made based on RMSE and MAE. From the experiment, we found that the GBT model performs the best among the three models by achieving an MAE of 20.86 minutes and a RMSE of 31.94 minutes, but no models were sufficient for predicting the duration. Such results shed light on possible future improvements to build better-performing models by adding more features.

1 Introduction

As society has developed and roads have gotten more complex, many cities, especially metropolitan and urban cities have seen an increase in severe traffic congestion [12]. Maintaining efficient traffic management and reducing congestion has become very difficult but crucial. In particular, traffic accidents are one of the factors that impact delays on the road. Therefore, accurate prediction of the accident duration can help travelers and traffic management systems [3]. In general, the accident duration is defined as the time interval between an accident's occurrence and its clearing from the scene (i.e., the time it takes for the impact of an accident to be dismissed from the road) [4]. The challenge of predicting it is that the traffic accident duration is influenced by a number of variables, some of which are not observable [4]. As a result, the traffic accident duration is inherently heterogeneous, which makes it highly unpredictable [4].

Identifying what attributes are related to the duration of a traffic accident is important to implementing prediction models and will directly affect the usability of prediction models. The underlying patterns could be linked to weather conditions, road conditions, location, certain days, or certain times of day. Thus, the objective of this project is to 1) identify relevant features that can offer insights into predicting the duration of traffic accidents and 2) build prediction models by integrating these features.

For this project, we used the 'US-Accidents' dataset from Kaggle. Using a number of APIs that offer streaming traffic incident data, the data were gathered between February 2016 and March 2023 for the Contiguous United States [11]. These APIs publish traffic data that has been collected by a number of organizations, such as law enforcement organizations, state and federal transportation departments, traffic sensors, and traffic cameras installed in road networks [11]. It contains about 7.7 million accident entries (7,728,394) and 47 features for each entry. The features include but are not limited to 1) the severity of the accident, 2) timestamps of an accident when it occurred and when it ended, 3) geographic and locational information such as latitude, longitude, state, county, city, street, zip code, and the 4) weather information such as temperature, humidity, and precipitation (for the detailed features, see *Table 1* in *Appendix A*). Within the given data, selecting and preprocessing appropriate features were very critical.

2 Methodology

To be able to work with big data efficiently, this project was intended to leverage Spark, a distributed data processing framework [8]. Specifically, PySpark, which is the Python API for Spark, was employed.

2.1 Regression Modeling Approaches

Three different tree-based models are used as they are good at capturing complex non-linear relationships. These models are deployed from the 'pyspark.ml' package in PySpark: *DecisionTreeRegressor*, *RandomForestRegressor*, and *GBTRegressor*.

- Decision Tree (DT) [7]: A DT is a predictive model that recursively splits a dataset into subsets based on input features. A decision is made at each internal node of a decision tree. The tree structure consists of internal nodes representing decision rules and leaf nodes that provide predictions. The disadvantage of the DT model is its susceptibility to overfitting.
- Random Forest (RF) [2]: The RF model is an ensemble of decision trees. It means multiple trees are trained to produce regression in the RF as opposed to a single one in a DT model. Many de-correlated trees are trained and then aggregated to average the predictions to get a single result. Thus, it reduces the risk of overfitting.

- Gradient Boosted Tree (GBT) [2]: GBT is based on the same concept as RF, but the difference is in the GBT model, each tree makes a weighted prediction. This results in some trees being more predictive than others for certain classes.

2.2 Evaluation Metrics

The mean absolute error (MAE) and the root mean squared error (RMSE) are used to evaluate the performance of our models. They are commonly used to measure how well the line fits the data in regression analysis [2]. For both metrics, it is ideal to have a value close to 0. The metrics are utilized from the *RegressionEvaluator* class in PySpark's ML package (pyspark.ml).

- MAE [10]: MAE is defined as the average of absolute errors. The error here means the difference between the predicted value and the actual value in the data. MAE has the same unit as the target value, which is intuitive for understanding. Moreover, MAE increases linearly because it is always a positive value.
- RMSE [10]: RMSE is defined as a square root of mean squared error (MSE). Like MAE, RMSE has the same unit as the target value, which makes it easier to interpret than MSE. However, in RMSE, as it squared the errors, the larger errors will be more penalized. Therefore, if the data has many outliers, the RMSE will be much larger.

3 Data Preprocessing and Exploratory Data Analysis

In order to select relevant features, preliminary data exploration was necessary. In the preprocessing steps, we conducted feature creation, feature extraction, handling missing data, removing outliers, categorical variables encoding, and standardization of numerical features. By utilizing several preprocessing and feature engineering techniques, the data was cleaned and prepared in a usable form to be used by the regression models.

3.1 Feature Creation

As the duration feature is not explicitly given in the dataset, a new column named 'Duration' was created by deducting the timestamp in the 'End_Time' column from the 'Start_Time' column using the unix_timestamp function. The result was then divided by 60 to convert the unit from seconds to minutes for easier interpretation.

3.2 Handling Outliers

The outliers were removed from the columns of ‘Duration’, ‘Distance(mi)’, and ‘Visibility(mi)’ as these columns were highly concentrated on the left side. The interquartile method was used for handling the outliers in ‘Duration’ and ‘Distance’. For ‘Visibility’, using the interquartile method drops too many values, so it was reasonably adjusted to include only values that are less than 20 since 75% of the values contain less than 10 (see *Figures 1,2,3* in *Appendix A*).

3.3 Handling Missing Data

There are several columns containing null values. Different methods were used for different columns. As the ‘Weather_Timestamp’ column contains the time of weather observations, the 'Temperature(F)' column was treated as time-series data. A window was set to be ordered based on the continuous ‘Weather_Timestamp’. The imputation strategy included: 1) using the previous value, 2) using the next value, or 3) using the average value if both were unavailable. This approach aimed to maintain temporal continuity, assuming that adjacent time periods have similar temperatures.

For the ‘Sunrise_Sunset’ column, as it contains categorical data (Day/Night), the forward-filling method was used to fill in missing values by replacing them with previous data. Moreover, the 'Street' and 'City' columns contained 8,406 and 144 null values, respectively, which accounted for less than 0.15% of the data. As it is a very small proportion of the dataset, the decision was made to drop null-containing rows from those columns.

3.4 Feature Extraction

The dataset contains timestamps of the accident, but it does not provide meaningful insights directly. Thus, new features including ‘hour’, ‘day_of_week’, ‘month’, ‘year’, and ‘day_of_month’ were created by extracting timestamp from the ‘Start_Time’ column. This allowed for a further analysis of their relationships with the accident duration.

During the exploratory data analysis, observations were made on day of month (1-31), day of week (Monday through Sunday), hour of day (1-24), month of year (January through December), and year (2016-2023). Some of the timing attributes revealed certain patterns associated with traffic accidents.

The following are the key points (see *Figures 4-9* in *Appendix A*):

- There are more traffic accidents during rush hour (from 7 to 8 a.m. and from 4 to 5 p.m.) in a day.
- Most accidents occur on weekdays rather than weekends.
- Generally, the number of accidents has been increasing every year since 2016.
- The day of the month does not appear to affect the number of accidents (i.e., it was consistent from the first day to the last day of the month).
- There are more accidents occurring in winter than in summer.

After the observation, it was decided to drop the 'day_of_month' column as it would not provide much meaningful information for the model to learn. Also, the data for the year 2023 was dropped because the data contains only the first three months of that year (January through March), so it does not provide complete information for 2023.

3.5 Handling Categorical Values

Categorical features including 'Street', 'City', 'County', 'State', and 'Sunrise_Sunset' was first handled by *StringIndexer* which maps the string data with different indices. For the 'Sunrise_Sunset' containing Day or Night, Day was converted into 1, and Night was converted into 0, so it did not need one-hot encoding afterward.

Although 'hour,' 'day_of_week,' 'month,' and 'year' were already in integer format, it was deemed appropriate to treat them as categorical data, given the absence of meaningful orders in their values. As such, *OneHotEncoder* was used to encode them into a binary vector format.

3.6 Feature Selection

For weather-related data, initial consideration included 'Temperature(F)', 'Wind_Chill(F)', 'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Speed(mph)', 'Precipitation(in)', and 'Sunrise_Sunset'. Due to more than 27% of null values in 'Precipitation(in)', it was excluded from further analysis. Additionally, the correlation matrix was analyzed to check correlations among some weather-related variables (see *Figure 10* in *Appendix A*). While a correlation matrix can capture only linear relationships between variables, observing a correlation matrix was still informative to reduce redundancy by selecting representatives from amongst

similar features. Finally, 'Temperature(F)' and 'Visibility(mi)' were selected to represent weather conditions, reducing redundancy.

For geographic and locational features, including latitude, longitude, street, city, county, and state, initial consideration revealed some of them are correlated (e.g., County and City), as shown in the correlation heatmap (see *Figure 10* in *Appendix A*). Additionally, due to the high dimensionality of using categorical features like street, city, county, and state that need to be encoded, it was decided to use latitude and longitude to represent locational information.

After comprehensive analysis, the following 11 features were selected to be independent variables to represent each attribute that can potentially affect the prediction of duration:

- Accident attributes: Severity, Distance(mi)
- Location attributes: Start_Lat, Start_Lng
- Weather attributes: Temperature(F), Visibility(mi)
- Time attributes: Sunrise_Sunset, hour, day_of_week, month, year

3.7 Standardization of Numerical Features

For the numerical features including 'Start_Lat', 'Start_Lng', '[1]Distance(mi)', 'Temperature(F)', and 'Visibility(mi)', they were first concatenated using *VectorAssembler* and standardized to have a mean of 0 and a standard deviation of 1 using *StandardScaler*. Then all the feature columns were assembled into a vector using *VectorAssembler*. As a result of preprocessing, about 5.6 million (5,682,003) rows remained, and the data frame for the regression models was prepared with the 'label' column (renamed from 'Duration') and the 'features' column containing a vector of 11 features.

4 Experiments and Results

Due to the significant computational demand corresponding to the data size, it was decided to work with a sample dataset, which is 10% of the entire dataset (approximately 560,000 rows). This small subset was extracted from the data frame after shuffling. The subset was

[1] Distance(mi): the length of the road impacted by the accident, measured in miles.

further split into training and testing sets using the *randomSplit* method, maintaining an 80:20 ratio. Initially, baseline models of DTR, RF, and GBT with default parameters were trained and tested on a new dataset. These baseline models will be compared with the tuned models.

Hyperparameter tuning was carried out using *TrainValidationSplit* and *ParamGridBuilder*. Optimized parameters were determined based on each model's performance assessed by RMSE. The previously prepared training set was then again randomly divided into a training set (80%) and a validation set (20%) during the tuning process. After tuning, the models were evaluated on a test set containing previously unseen data.

For DTR and RF models, the tried values included [10, 15] for maximum depth and [40, 60] for maximum bins. The default parameters for baseline models of DTR and RF were 5 for maximum depth and 32 for maximum bins. However, through the hyperparameter tuning, the optimal hyperparameters for both models were found to be 15 for maximum depth and 60 for maximum bins (refer to *Figures 12 and 13* in *Appendix A*). DTR model and RF model performance comparisons are shown in the following *Figure 1* and *Figure 2*, respectively.

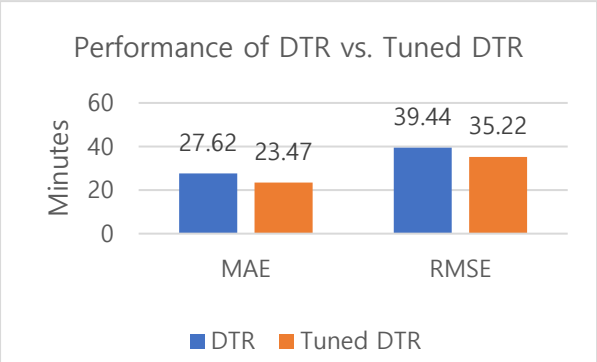


Figure 1. Performance of DTR vs. Tuned DTR

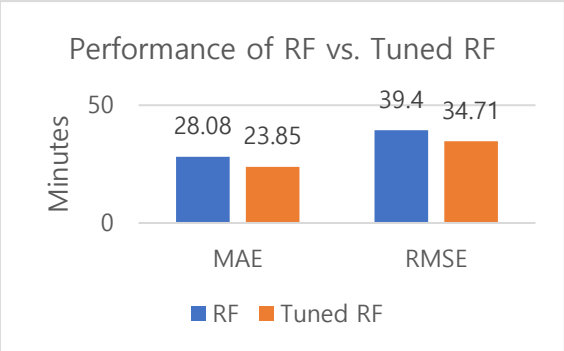


Figure 2. Performance of RF vs. Tuned RF

In the case of the GBT model, the tried values for tuning were [10, 15] for maximum depth and [20, 30] for maximum iteration. GBT baseline model's default parameter for maximum depth was 5 and maximum iteration was 20. After tuning, the optimal hyperparameters were identified as 15 for maximum depth and 30 for maximum iteration. The performance comparison for the GBT model's before and after tuning is shown in *Figure 3*.

The hyperparameter tuning resulted in a reduction in MAE and RMSE for all three models. On average, MAE decreased by 4.07 minutes and RMSE decreased by 4.13 minutes. It is worth noting that initially different sets of hyperparameters were utilized, but expanding the hyperparameter grid significantly increases the runtime. As a result, the choice of hyperparameter was made within the given limitations. However, it was observed that all the models worked better with the largest number among the given range of hyperparameter values, so it indicates that it could be possible to tune the models even more with different sets of hyperparameters.

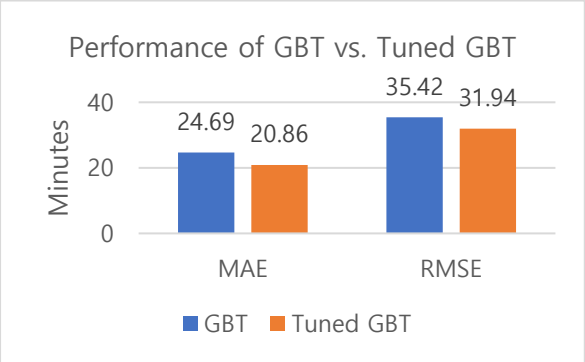


Figure 3. Performance of GBT vs. Tuned GBT

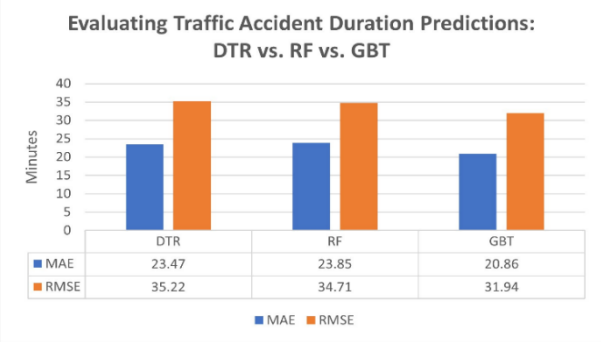


Figure 4. Performance Comparison between tuned DTR, RF, and GBT models

As illustrated in F*igure 4*, when assessed by MAE (in minutes), the tuned models' performance is ranked in the following order: the GBT model achieved the lowest error at 20.86, followed by the DTR model at 23.47, and the RF model at 23.85. In terms of RMSE (in minutes), the GBT model remains the top performer with an RMSE of 31.94. However, in this instance, the RF model performs slightly better than the DTR model with RMSE values of 34.71 and 35.22, respectively, which indicates the DTR model might have a greater number of large errors than the RF model.

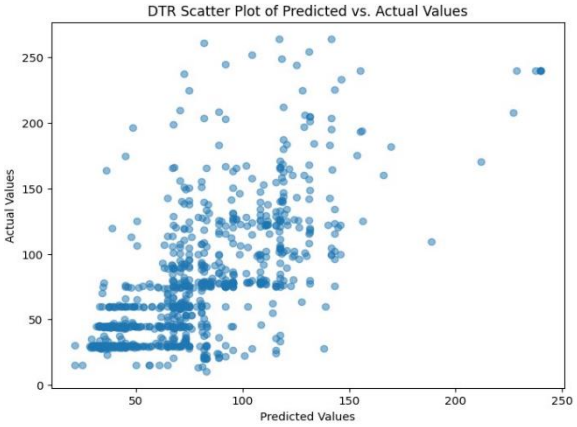


Figure 5. DTR Predicted vs. Actual Plot

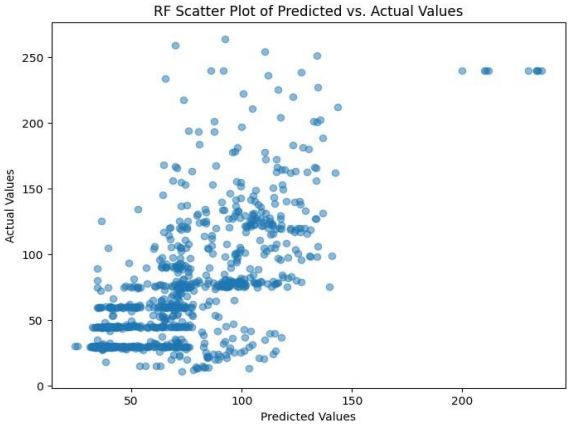


Figure 6. RF Predicted vs. Actual Plot

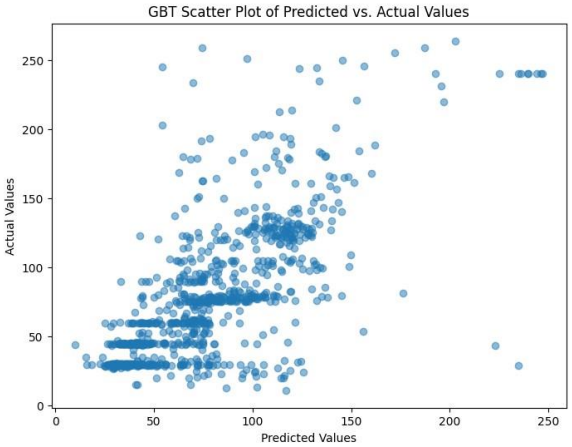


Figure 7. GBT Predicted vs. Actual Plot

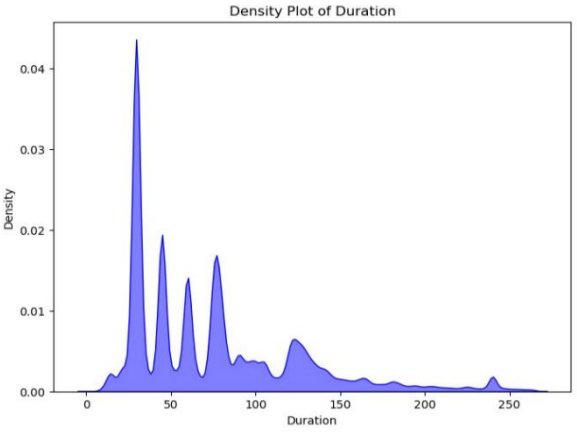


Figure 8. Density Plot of Duration

To see the whole picture of the difference between predicted values and actual values, the scatter plots in *Figures 5,6*, and *7* were generated with a random subsample of 1000 points for each graph. Generating this way was to help visualization be more understandable because including every point for a large dataset did not yield a clear representation and rather looked too cluttered. However, this approach should be considered cautiously because 1000 points out of 560,000 points might not accurately generate the representative points. To help avoid any misinterpretation, it would be ideal to generate plots with different sample numbers and compare them to see if the same patterns could be observed. Moreover, getting summary statistics for the predicted values can also give more information on the results.

In the graphs above, the x-axis indicates predicted values, and the y-axis indicates actual values. It is ideal to have values closer to a diagonal line from the bottom left to the

upper right because that means the predicted values are similar to actual values. Figure 8 shows the density of the 'Duration' column. It includes the range from 1.22 minutes to 265.78 minutes. The mean is 76.17 and the median is 60.17 minutes (refer to *Figure 11* in *Appendix A*). Even though outliers were removed, it is still highly skewed to the right. Therefore, it is reasonable to see all the scatter plots are cluttered in the bottom left of the graphs.

From the scatter plots, some observations on each model's performance can be made. The DTR model exhibits a higher degree of dispersion, indicating greater variability in its predictions. In comparison, the RF model demonstrates improved accuracy, yet it struggles to capture values beyond 150 minutes accurately, frequently predicting values within the 150-minute range when actual values span from 150 to 250 minutes. The GBT model emerges as slightly more effective in handling values over 150 minutes but performs similarly in smaller durations. Therefore, it is observable that while the GBT model exhibits some improvement over the other models, it falls short of achieving a significant advantage.

5 Conclusions and Future Work

The complexity and inherent unpredictability of traffic accidents presents challenges in capturing the multitude of influencing factors accurately. In this study, we explored the traffic accident dataset and experimented with DTR, RF, and GBT models for the prediction of traffic accident durations. Despite extensive efforts to enhance model performance through feature engineering and the iterative process for refining the models, none of the models were sufficient for predicting the duration accurately. When the range of 'Duration' is between 1.22 minutes and 265.78 minutes, the top-performing model, which was GBT, had an MAE of 20.86 minutes and a RMSE of 31.94 minutes, which are still large errors for use in traffic accident duration prediction. While the obtained RMSE and MAE values suggest suboptimal performance, the results still provide valuable insights which were that the chosen attributes contributed to predicting the duration.

However, we faced limitations regarding computing resources that restricted the increase in the number of features, the expansion of the dataset size, and exhaustive hyperparameter tuning. If given sufficient computing resources, we can enhance the models' capacity, thus predicting accident duration better. Moreover, future work should focus on incorporating more sophisticated features to enhance the model's performance.

For example, one could use text mining techniques to get detailed information from the ‘Description’ column which contains a human-provided description of the accident. In conclusion, while the models currently face limitations, the insights gained suggest avenues for improvement, and following these steps will increase the applicability of traffic accident duration prediction models.

References

[1] A. Abdulhafedh, “Road Crash Prediction Models: Different Statistical Modeling Approaches,” *Journal of Transportation Technologies*, vol. 07, no. 02, pp. 190–205, 2017, doi: https://doi.org/10.4236/jtts.2017.72014.

[2] B. Chambers and M. Zaharia, Spark: The Definitive Guide. *O’Reilly Media, Inc*., 2018.

[3] J. Chen and W. Tao, “Traffic accident duration prediction using text mining and ensemble learning on expressways,” *Scientific Reports*, vol. 12, no. 1, Dec. 2022, doi: https://doi.org/10.1038/s41598-022-25988-4.

[4] R. Li, F. C. Pereira, and M. E. Ben-Akiva, “Overview of traffic incident duration analysis and prediction,” *European Transport Research Review*, vol. 10, no. 2, May 2018, doi: https://doi.org/10.1186/s12544-018-0300-1.

[5] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath, “Accident Risk Prediction based on Heterogeneous Sparse Data,” Proceedings of the 27th ACM SIGSPATIAL *International Conference on Advances in Geographic Information Systems* - SIGSPATIAL ’19, 2019, doi: https://doi.org/10.1145/3347146.3359078.

[6] S. Moosavi, M. Samavatian, S. Parthasarathy, and R. Ramnath, “A Countrywide Traffic Accident Dataset,” Jun. 2019, doi: https://doi.org/doi.org/10.48550/arXiv.1906.05409.

[7] J. Ali, Rehan Ullah Khan, N. Ahmad, and I. Maqsood, “Random Forests and Decision Trees,” *International Journal of Computer Science Issues*, vol. 9, no. 5, pp. 272–278, Sep. 2012.

[8] “PySpark Documentation — PySpark 3.3.1 documentation,” spark.apache.org. https://spark.apache.org/docs/3.3.1/api/python/index.html (accessed Dec. 06, 2023).

[9] D. S. Tripathy and B. R. Prusty, “Forecasting of renewable generation for applications in smart grid power systems,” *Advances in Smart Grid Power System*, pp. 265–298, 2021, doi: https://doi.org/10.1016/b978-0-12-824337-4.00010-2.

[10] P. Schneider and F. Xhafa, “Anomaly detection,” Anomaly Detection and Complex Event Processing over *IoT Data Streams*, pp. 49–66, 2022, doi: https://doi.org/10.1016/b978-0-12-823818-9.00013-4.

[11] S. Moosavi, “US Accidents (2016 - 2023),” kaggle. https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents

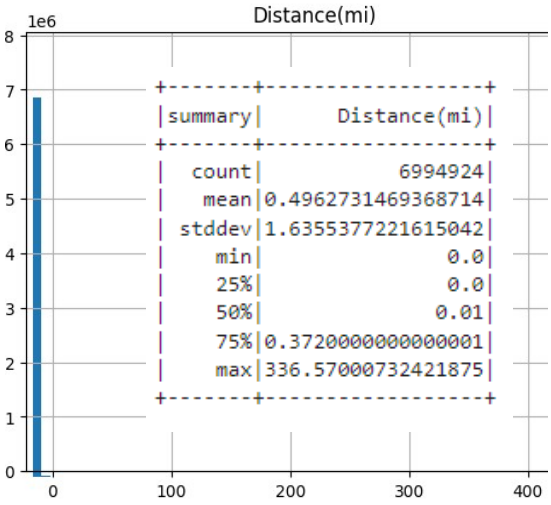
[12] Y. Liu, C. Liu, and Z. Zheng, “Traffic Congestion and Duration Prediction Model Based on Regression Analysis and Survival Analysis,” *Open Journal of Business and Management*, vol. 08, no. 02, pp. 943–959, 2020, doi: https://doi.org/10.4236/ojbm.2020.82059.

Appendices

A Supplementary Figures

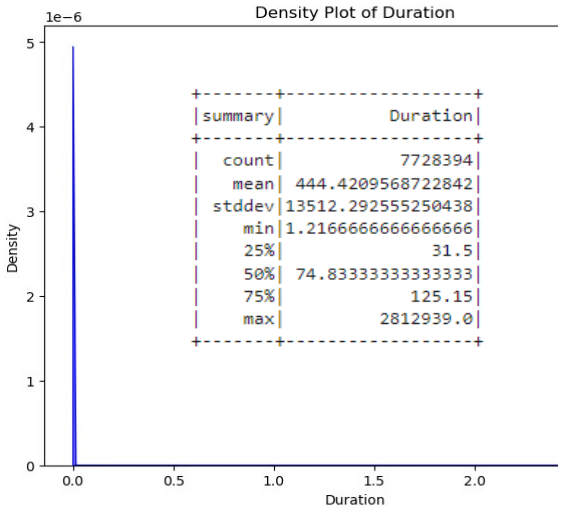
Table 1. The US-Accidents Dataset Attributes [6]

Attributes	47
Traffic (5)	ID, Source, Severity, Distance(mi), Description
Time (3)	Start_Time, End_Time, Timezone
Location (10)	Start_Lat, Start_Lng, End_Lat, End_Lng, Street, City, County, State, Zipcode, Country
Weather (11)	Airport_Code, Weather_Timstamp, Temperature(F), Wind_Chill(F), Humidity(%), Pressure(in), Visibility(mi), Wind_Direction, Wind_Speed(mph), Precipitation(in), Weather_Condition
Point of interest (13)	Amenity, Bump, Crossing, Give_Way, Junction, No_Exit, Railway, Roundabout, Station, Stop, Traffic_Calming, Traffic_Signal, Turning_Loop
Period of day (4)	Sunrise_Sunset, Civil_Twilight, Nautical_Twilight, Astronomical_Twilight



summary	Distance(mi)
count	6994924
mean	0.4962731469368714
stddev	1.6355377221615042
min	0.0
25%	0.0
50%	0.01
75%	0.3720000000000001
max	336.57000732421875

Figure 1. Density Plot and Summary Statistics of "Distance(mi)" Before Removing Outliers



summary	Duration
count	7728394
mean	444.4209568722842
stddev	13512.292555250438
min	1.2166666666666666
25%	31.5
50%	74.83333333333333
75%	125.15
max	2812939.0

Figure 2. Density Plot and Summary Statistics of "Duration" Before Removing Outliers

summary	Visibility(mi)
count	5862493
mean	9.119237012308606
stddev	2.642281516788764
min	0.0
25%	10.0
50%	10.0
75%	10.0
max	140.0

Figure 3. Summary Statistics of "Visibility(mi)" Before Removing Outliers

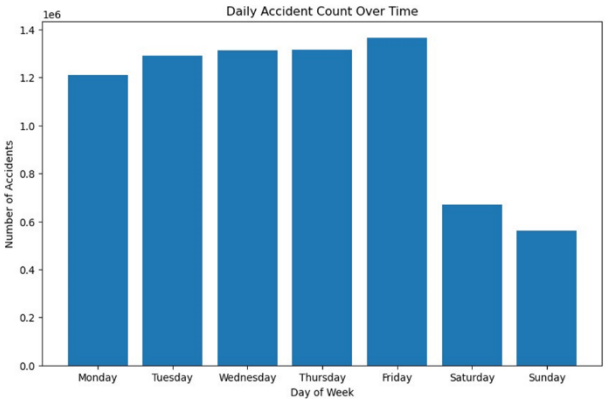


Figure 4. Number of Accidents per Day of Week

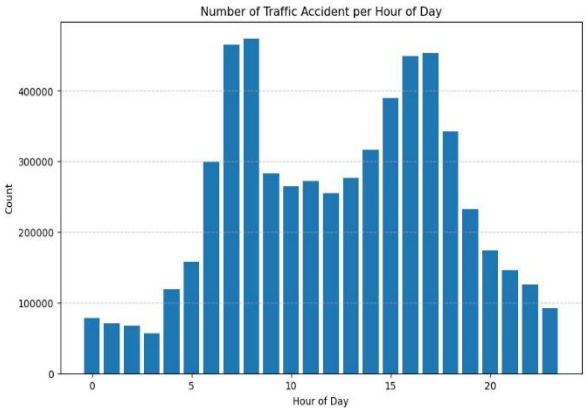


Figure 5. Number of Accidents per Hour of Day

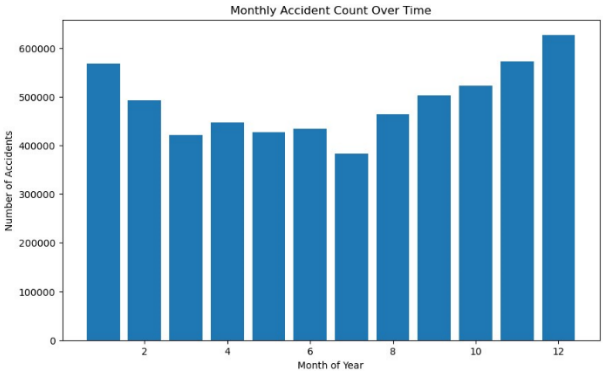


Figure 6. Number of Accidents per Month of Year

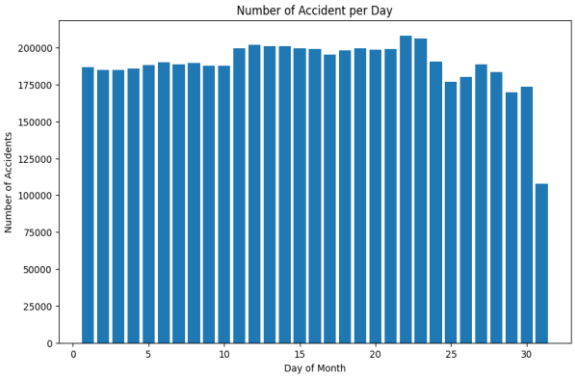


Figure 7. Number of Accidents per Day of Month

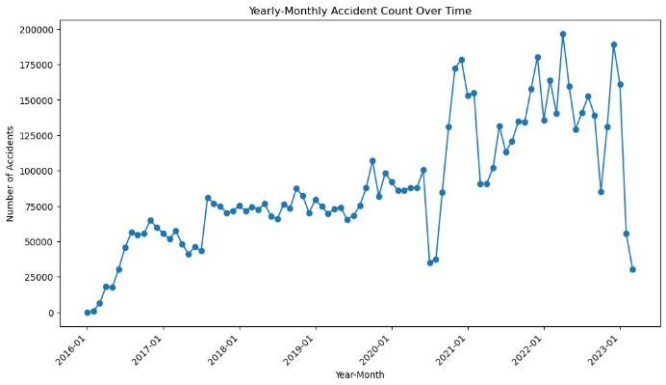


Figure 8. Accidents across Months in a Yearly Time Series

year	count
2016	272550
2017	515530
2018	746478
2019	862030
2020	938218
2021	1119208
2022	1225752
2023	171637

Figure 9. Number of Accidents per Year

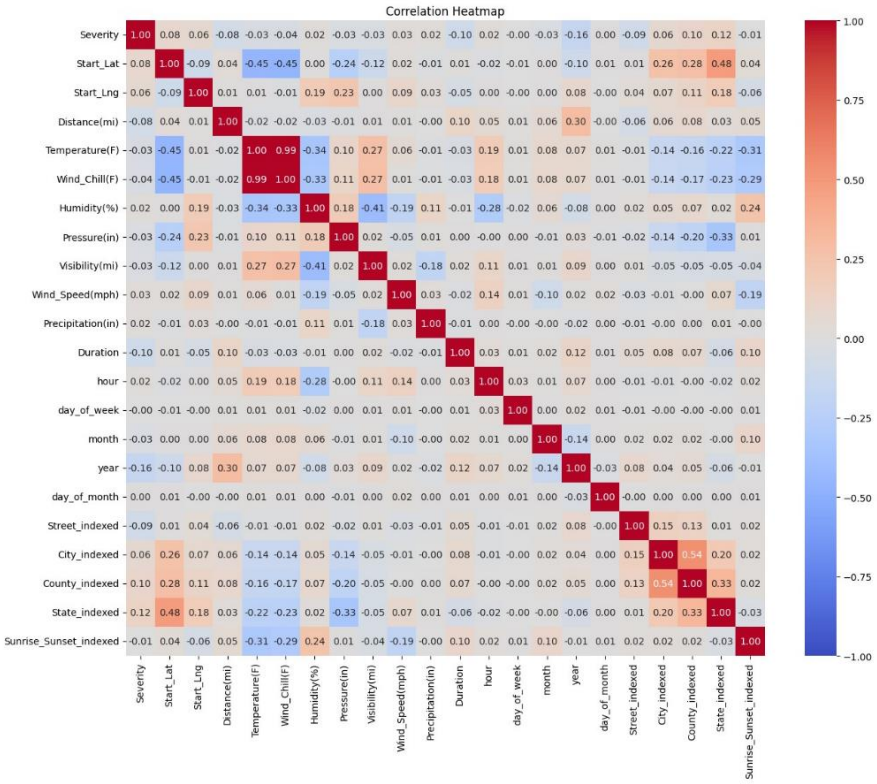


Figure 10. Correlation Heatmap

```
In [17]: removed_duration_df.select("Duration").summary().show()
```

```
+-------+------------------+
|summary|          Duration|
+-------+------------------+
|  count|           6996693|
|   mean| 76.16869711638464|
| stddev| 51.01711450327963|
|    min|1.2166666666666666|
|    25%|              29.9|
|    50%|60.166666666666664|
|    75%|103.58333333333333|
|    max|265.78333333333336|
+-------+------------------+
```

Figure 11. Summary statistic of Duration after removing outliers