PROJECT REPORT

GT / Michelin: Driving Route Risk Analysis

Group 2: Michael Crist, Vishal Jada, Hersh Gupta

Applied Analytics Practicum

11/23/2023

Contents

Abstract

In this report we detail the exploratory data analysis, model building / evaluation, and feature analysis contributing to the ranking of riskiest delivery routes in the greater Atlanta area. The goal of this report is to result in improved driver safety and efficiency for Michelin Connected Fleet.

We chose to use a supervised learning approach by creating a response variable representing the risk score of a road segment. The models evaluated in the report include Linear Regression, Ridge Regression, Lasso Regression, Elastic Net Regression, KNN Regression, Random Forest Regression, and Neural Net Regression. Random Forest Regression was determined to be the best performing model for predicting risk score. The most important features for predicting risk were number of road lanes, road priority, and speed limit.

Introduction / Problem Statement

Based upon National Highway Traffic Safety Administration data, there were approximately 5.2 million reported accidents in the US in 2020, resulting in over 35,000 fatalities (Bieber, 2023). In addition to the disturbing fatalities figure, road incidents cost Americans approximately $340 billion per year in medical expenses, property damage, lost productivity, and traffic congestion (Gopin, 2023).

Michelin Connected Fleet provides the tools and solutions necessary for fleet operators / managers to safely and efficiently manage their fleets, working as a partner to provide recommendations and data insights to reduce cost, improve productivity, and ensure safety of drivers to manage sustainable fleets. Our aim is to define a data-driven approach to calculate a route safety score to inform fleet managers which delivery routes have the highest risk. We look to accomplish the following goals:

1. Explore available data and evaluate metrics for assessing road risk.
2. Build a variety of models to predict risk along a road segment and determine the features that have the greatest impact on road risk.
3. Rank the highest-risk delivery routes in the greater Atlanta area.

The motivation for this project is to provide the necessary analytics to implement intelligent routing for Michelin Connected Fleet, resulting in improved driver safety, reduced liability cost, and improved efficiency.

Exploratory Data Analysis

Data Sources

Multiple data sources were used for this project. We utilized the following datasets:

- TRIPS dataset: 173.5GB dataset (1.3B rows, 54 columns) composed of telematics data from Michelin fleets

- EVENTS dataset: 52.7GB dataset (337M rows, 59 columns) composed of Michelin driver telematics data in which a potential safety event occurred (ex: cell phone usage, excessive speeding, etc)

- CRASHES dataset: 6.6MB dataset (70.8K rows, 51 columns) from GA Department of Transportation containing GA public accident data from 2019 to 2023

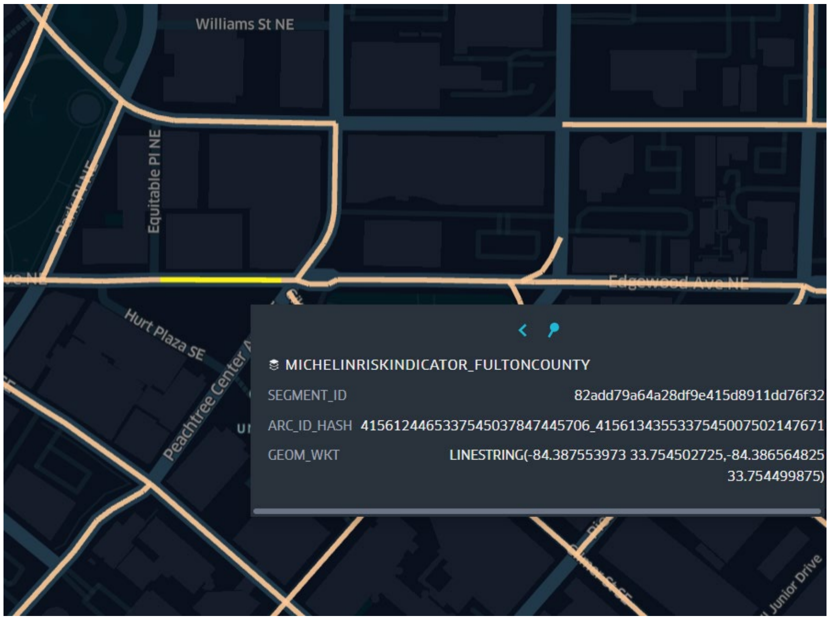
The Michelin datasets were collected using multiple methods: infrastructures via road scanning assets collection, and fleet drivers via in-app usage. The TRIPS and EVENTS datasets are composed of a wide variety of categorical and numeric features including geographic information, terrain information, and road information such as speed limit, number of road lanes, etc.

Tools / Software

- **Kepler**: open-source geospatial analysis tool for large-scale data sets
- **Snowflake**: cloud-based data warehouse for data storage and analysis
- **Databricks**: analytics platform for building, deploying, sharing, and maintaining data, analytics, and modeling solutions at scale.

ETL (Extract, Transform, Load)

As a first step in preparing the data for EDA, we performed ETL to transform the data into a useful state. The datasets are broken down into two primary geographical identifiers: Arc ID and Segment ID. The arc ID is a unique identifier representing the smallest linear "stretch" of road corresponding to a lat/long coordinate. The segment ID is a unique identifier representing a segment, which is a stretch of road between two intersections. A segment is composed of one or more arcs.



MICHELINRISKINDICATOR_FULTONCOUNTY

SEGMENT_ID	82add79a64a28df9e415d8911dd76f32
ARC_ID_HASH	4156124465337545037847445706_4156134355337545007502147671
GEOM_WKT	LINESTRING(-84.387553973 33.754502725,-84.386564825 33.754499875)

Kepler.gl map displaying an example of arc/segment representation

We chose to perform our EDA and modeling using road segments rather than arcs. As such, we aggregated the data on segment ID in preparation for EDA and modeling. To summarize:

- Each entry in the raw datasets represents a lat/long coordinate which is mapped to a unique "Arc ID", which represents the shortest linear stretch of road containing that lat/long.
- "Segment ID" is a unique identifier for each stretch of road between two intersections. A segment is composed of one or more arcs.

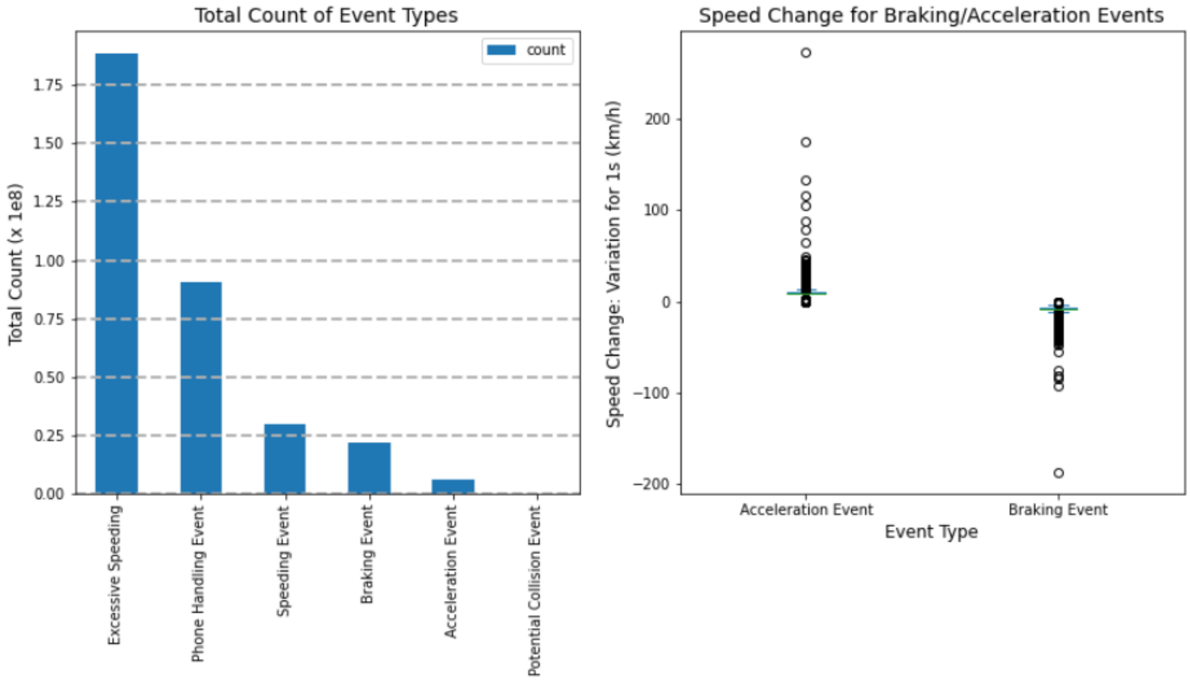
- We transformed the data such that each entry represents a segment, rather than an arc. We achieved this by aggregating on Segment ID (eg, “Terrain Roughness Index” was transformed from the terrain roughness index for a specific arc to the average terrain roughness index along all arcs that compose a segment)

Feature Definition

Feature	Type	Label
Unique identifier representing the road segment	Categorical	SEGMENT_ID
Number of entries in TRIPS dataset for corresponding segment	Numeric	TRIP_COUNT
Number of events that occurred on segment	Numeric	EVENT_COUNT
Number of crashes that occurred on segment	Numeric	CRASH_COUNT
Number of excessive speeding events that occurred on segment	Numeric	EXC_SPEED_COUNT
Number of speeding events that occurred on segment	Numeric	SPEED_COUNT
Number of phone handling events that occurred on segment	Numeric	PHONE_COUNT
Number of potential collisions that occurred on segment	Numeric	POT_COLL_COUNT
Number of acceleration events that occurred on segment	Numeric	ACC_COUNT
Number of braking events that occurred on segment	Numeric	BRAKE_COUNT
Max slope of road on segment	Numeric	MAX_ARCSLOPE
Min slope of road on segment	Numeric	MIN_ARCSLOPE
Avg slope of road along segment	Numeric	AVG_ARCSLOPE
Max number of road lanes along segment	Numeric	MAX_ROAD_LANES
Avg number of road lanes along segment	Numeric	AVG_ROAD_LANES
Max road curve angle along segment	Numeric	CURVE_MAXANGLE
Avg speed limit along segment	Numeric	SPEEDLIMIT_AVG
Max speed limit along segment	Numeric	SPEEDLIMIT_MAX
Presence of a tunnel along segment	Boolean	TUNNEL
Presence of a bridge along segment	Boolean	BRIDGE
Max terrain roughness index along segment	Numeric	TRI_MAX
Avg terrain roughness index along segment	Numeric	TRI_AVG
Road priority of segment (1-15)	Numeric	ROAD_PRIORITY
Average annual daily traffic on segment in 2022	Numeric	AADT_AVG_2022
Response variable (see following section for details)	Numeric	RISK_NORM

Establishing a Response Variable

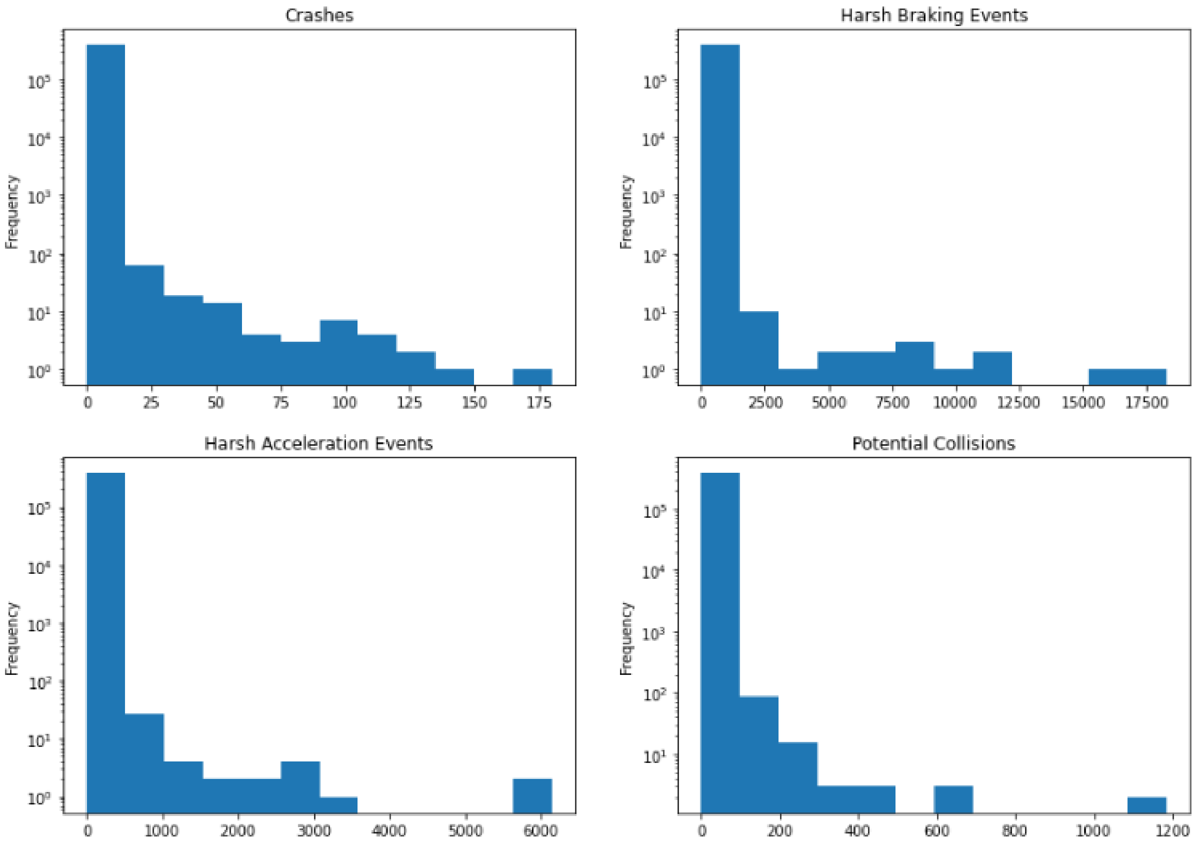
A major challenge of this project was the selection of an appropriate response variable to use for modeling road risk. The ideal response variable would be the normalized rate of crashes along a segment (total number of crashes divided by daily traffic on segment). However, crash data is very sparse due to the relative infrequency and underreporting of crashes. As such, we leveraged both crash data and Michelin-provided events data to create an appropriate response variable to represent road risk. To achieve this, we first stratified the events data to utilize only the "harshest" events.



Stratification of events data

As can be seen in the bar plot above, there were many excessive speeding and phone handling events. Due to their high frequency, we chose not to incorporate those event types into our response variable because they would bias risk score significantly. We focused on braking, acceleration, and potential collision events. Since most braking and acceleration events were very "mild", as can be seen by the midpoints in the boxplot above, we only used braking / acceleration events in the top 10^{th} percentile of speed change during the event. These we considered "harsh" braking / acceleration events.

The response variable we created, called RISK_NORM, is a risk score ranked in the range 0-100. It is the weighted summation of number of crashes, harsh braking events, harsh acceleration events, and potential collision events, divided by the average daily traffic on the segment and normalized to a range of 0-100. To weight the different types of events we looked at the histogram distribution for each event type over all road segments:



Clearly, there are far fewer crashes and potential collision events than there are harsh braking and acceleration events. To address this, we need to apply a weighting factor to these event types to increase their relative importance in the risk score metric. As such, in our risk score response variable we weighted crashes by a factor of 100 and potential collisions by a factor of 10.

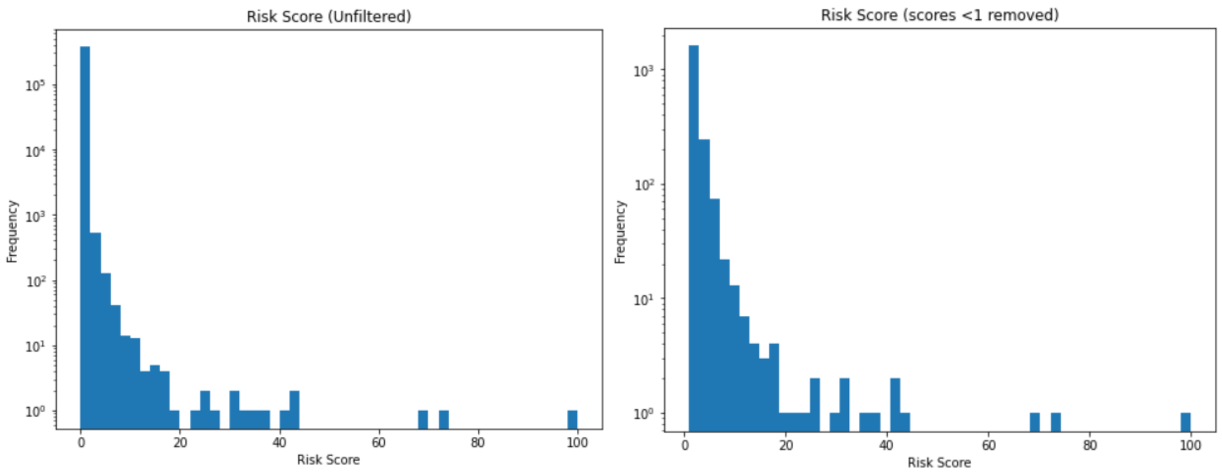
The calculation to obtain the response variable is shown in the pseudo-code below:

<u>Pseudo-code:</u>

```
# calculate risk metric by taking weighted summation of crashes + adverse events, divided by AADT
RISK = [ 100*CRASH_COUNT + BRAKE_COUNT(top 10% only) + ACC_COUNT(top 10% only) + 10*POT_COLL_COUNT ] / AADT_AVG_2022

# normalize to get metric in range 0-100
RISK_NORM = [ RISK / MAX(RISK) ] * 100
```

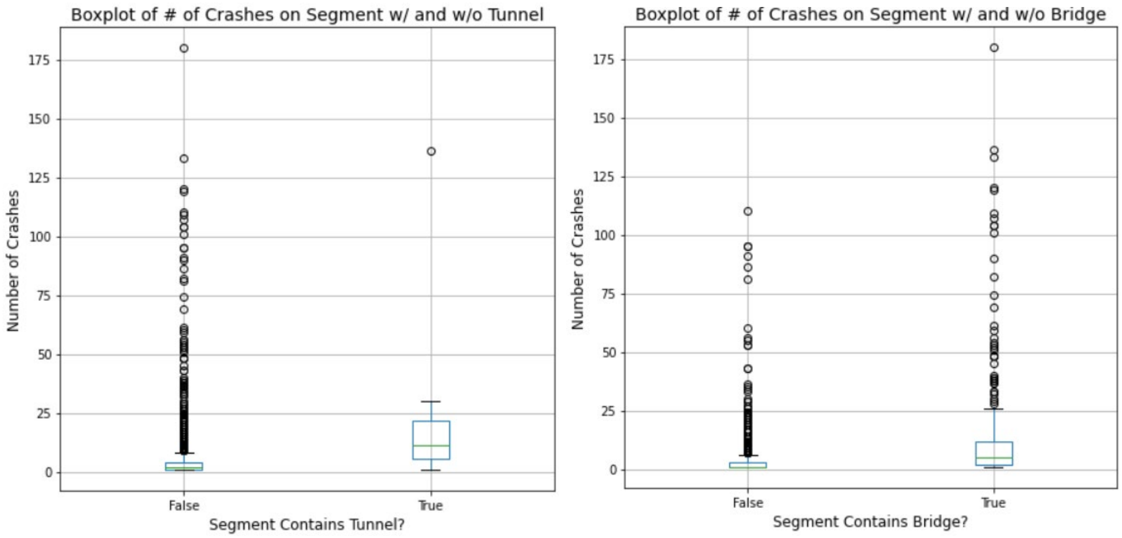
Once we established the risk score variable, we plotted a histogram to visualize the risk score distribution for all segments. The distribution was heavily right-tailed and biased toward lower scores, which is expected because crashes and other harsh road events are sparse, relative to the total population of road segments. To address this and reduce bias in our modeling, we removed segments with risk score < 1 from the dataset, and assumed those segments to have a risk score of zero.



Risk Score distribution before and after filtering out low scores

Boolean Features

There are two Boolean features contained in the data that we chose to explore: tunnel (indicating the presence of a tunnel along a segment) and bridge (indicating the presence of a bridge along a segment). To explore these features, we generated boxplots to compare the number of crashes occurring on road segments with and without tunnels and bridges.

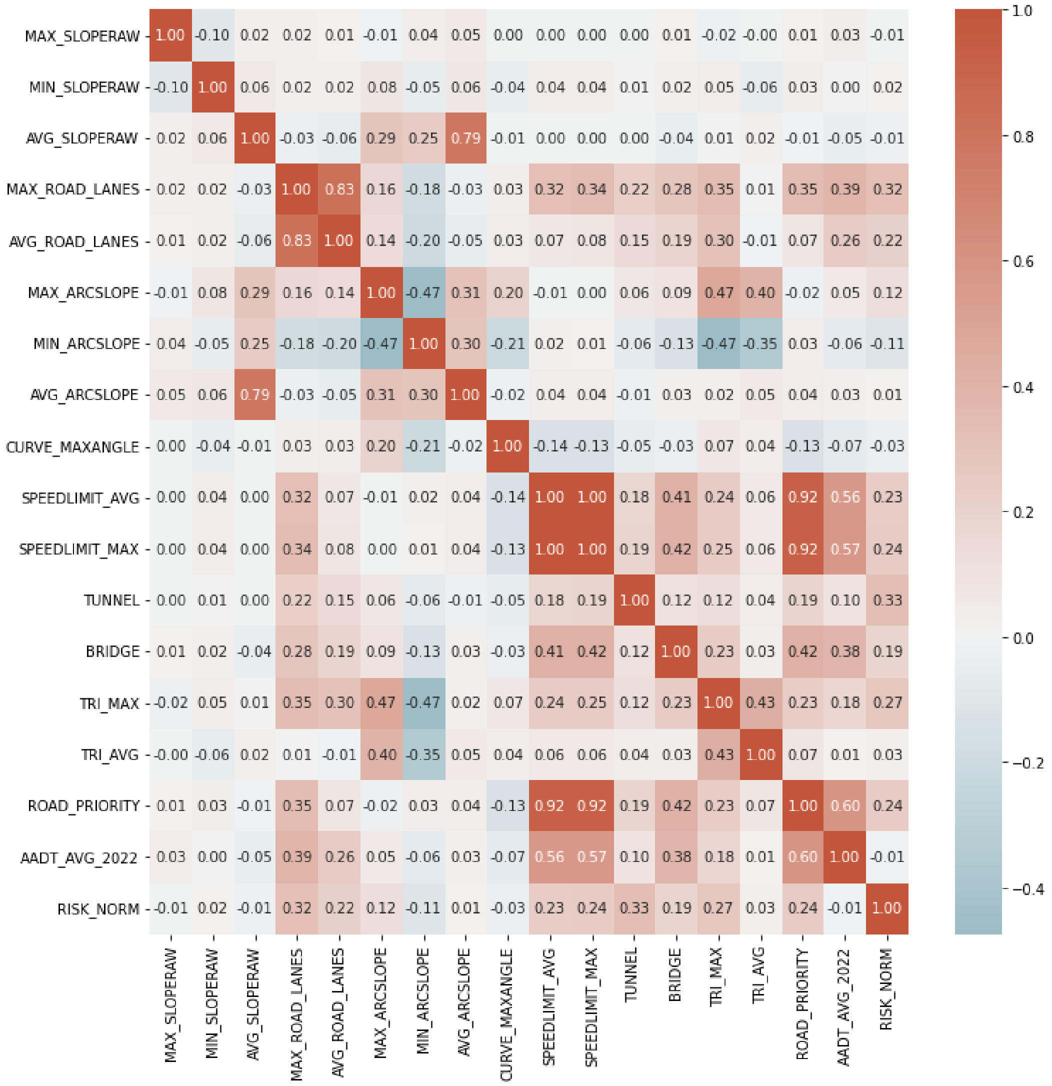


Boxplots of Boolean Variables

Looking at the boxplots, there is a clear correlation between the presence of a tunnel or bridge and the likelihood of a crash. The median number of crashes occurring on a segment is shifted substantially higher when a tunnel or bridge is present; this effect is most pronounced for the presence of a tunnel.

Numeric Features

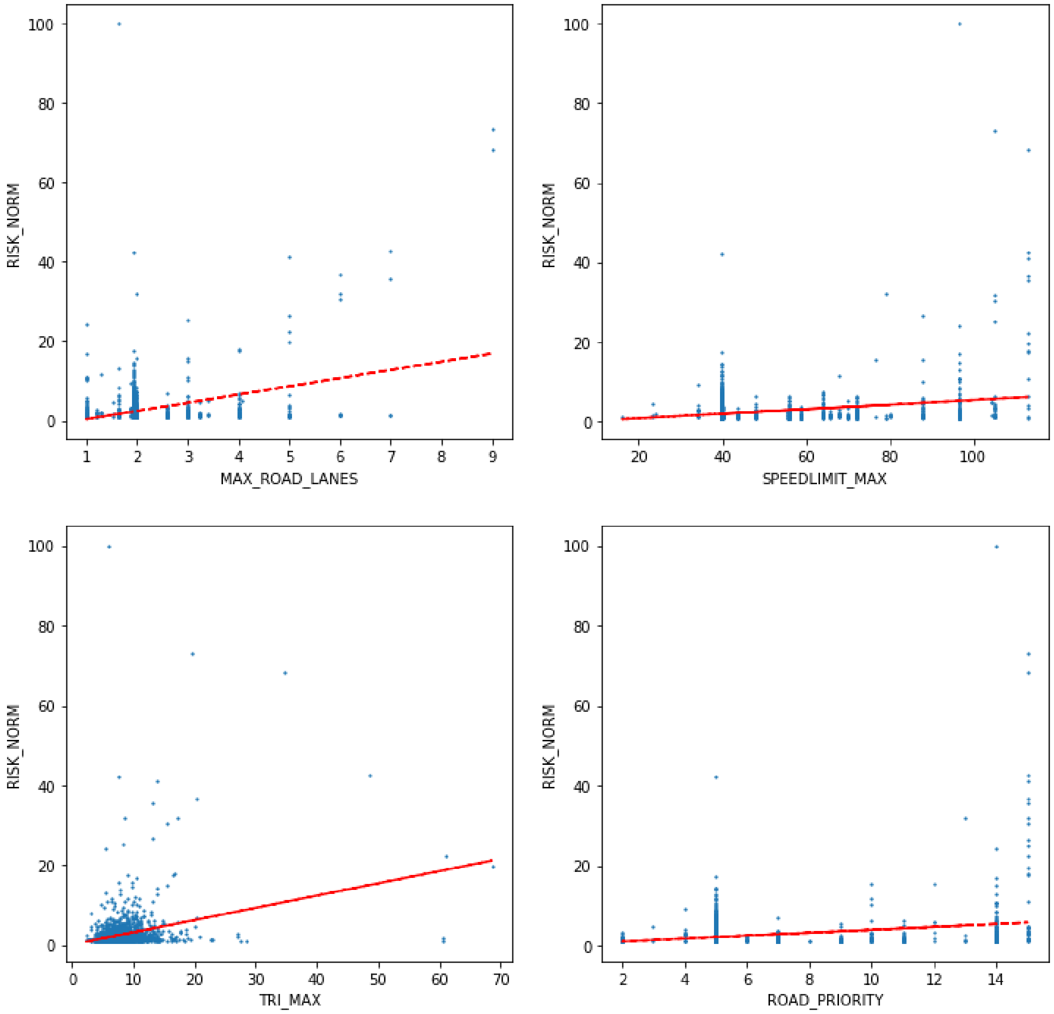
As a first step in exploring the numeric predicting features in our dataset we plotted a correlation matrix of all numeric features, which can be found in the figure below.



	MAX_SLOPERAW	MIN_SLOPERAW	AVG_SLOPERAW	MAX_ROAD_LANES	AVG_ROAD_LANES	MAX_ARCSLOPE	MIN_ARCSLOPE	AVG_ARCSLOPE	CURVE_MAXANGLE	SPEEDLIMIT_AVG	SPEEDLIMIT_MAX	TUNNEL	BRIDGE	TRI_MAX	TRI_AVG	ROAD_PRIORITY	AADT_AVG_2022	RISK_NORM
MAX_SLOPERAW	1.00	-0.10	0.02	0.02	0.01	-0.01	0.04	0.05	0.00	0.00	0.00	0.00	0.01	-0.02	-0.00	0.01	0.03	-0.01
MIN_SLOPERAW	-0.10	1.00	0.06	0.02	0.02	0.08	-0.05	0.06	-0.04	0.04	0.04	0.01	0.02	0.05	-0.06	0.03	0.00	0.02
AVG_SLOPERAW	0.02	0.06	1.00	-0.03	-0.06	0.29	0.25	0.79	-0.01	0.00	0.00	0.00	-0.04	0.01	0.02	-0.01	-0.05	-0.01
MAX_ROAD_LANES	0.02	0.02	-0.03	1.00	0.83	0.16	-0.18	-0.03	0.03	0.32	0.34	0.22	0.28	0.35	0.01	0.35	0.39	0.32
AVG_ROAD_LANES	0.01	0.02	-0.06	0.83	1.00	0.14	-0.20	-0.05	0.03	0.07	0.08	0.15	0.19	0.30	-0.01	0.07	0.26	0.22
MAX_ARCSLOPE	-0.01	0.08	0.29	0.16	0.14	1.00	-0.47	0.31	0.20	-0.01	0.00	0.06	0.09	0.47	0.40	-0.02	0.05	0.12
MIN_ARCSLOPE	0.04	-0.05	0.25	-0.18	-0.20	-0.47	1.00	0.30	-0.21	0.02	0.01	-0.06	-0.13	-0.47	-0.35	0.03	-0.06	-0.11
AVG_ARCSLOPE	0.05	0.06	0.79	-0.03	-0.05	0.31	0.30	1.00	-0.02	0.04	0.04	-0.01	0.03	0.02	0.05	0.04	0.03	0.01
CURVE_MAXANGLE	0.00	-0.04	-0.01	0.03	0.03	0.20	-0.21	-0.02	1.00	-0.14	-0.13	-0.05	-0.03	0.07	0.04	-0.13	-0.07	-0.03
SPEEDLIMIT_AVG	0.00	0.04	0.00	0.32	0.07	-0.01	0.02	0.04	-0.14	1.00	1.00	0.18	0.41	0.24	0.06	0.92	0.56	0.23
SPEEDLIMIT_MAX	0.00	0.04	0.00	0.34	0.08	0.00	0.01	0.04	-0.13	1.00	1.00	0.19	0.42	0.25	0.06	0.92	0.57	0.24
TUNNEL	0.00	0.01	0.00	0.22	0.15	0.06	-0.06	-0.01	-0.05	0.18	0.19	1.00	0.12	0.12	0.04	0.19	0.10	0.33
BRIDGE	0.01	0.02	-0.04	0.28	0.19	0.09	-0.13	0.03	-0.03	0.41	0.42	0.12	1.00	0.23	0.03	0.42	0.38	0.19
TRI_MAX	-0.02	0.05	0.01	0.35	0.30	0.47	-0.47	0.02	0.07	0.24	0.25	0.12	0.23	1.00	0.43	0.23	0.18	0.27
TRI_AVG	-0.00	-0.06	0.02	0.01	-0.01	0.40	-0.35	0.05	0.04	0.06	0.06	0.04	0.03	0.43	1.00	0.07	0.01	0.03
ROAD_PRIORITY	0.01	0.03	-0.01	0.35	0.07	-0.02	0.03	0.04	-0.13	0.92	0.92	0.19	0.42	0.23	0.07	1.00	0.60	0.24
AADT_AVG_2022	0.03	0.00	-0.05	0.39	0.26	0.05	-0.06	0.03	-0.07	0.56	0.57	0.10	0.38	0.18	0.01	0.60	1.00	-0.01
RISK_NORM	-0.01	0.02	-0.01	0.32	0.22	0.12	-0.11	0.01	-0.03	0.23	0.24	0.33	0.19	0.27	0.03	0.24	-0.01	1.00

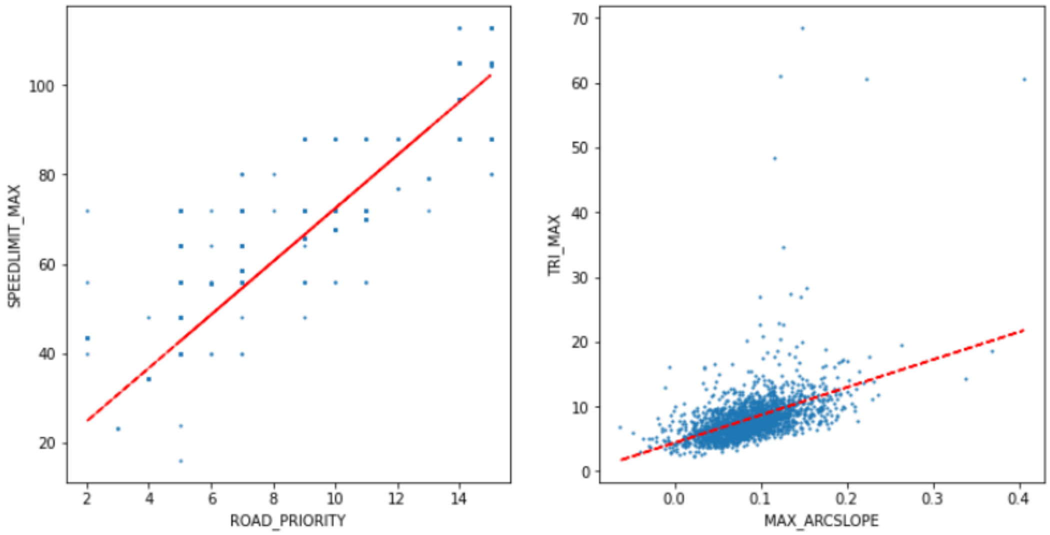
Correlation Matrix of all features

From the correlation matrix, some features have a much stronger correlation with risk score (RISK_NORM) than others. The number of road lanes and the presence of a tunnel have the strongest positive correlation with risk score, with correlation > 0.30. Additionally, speed limit, terrain roughness index, and road priority all have positive correlations with risk score that are > 0.20. These marginal relationships are visualized in the scatterplots below.



Risk score plotted against high-correlation predicting features

In addition to correlation with the response variable (RISK_NORM), there also appear to be strong correlations between some of the predictor variables, which could indicate the presence of multicollinearity in the data. For example, the correlation is very strong between road priority and speed limit, as well as max slope (MAX_ARCSLOPE) and terrain roughness index (TRI_MAX). This is expected, because higher priority roads (like highways) logically have a higher speed limit, and roads with steep gradients logically have a more severe terrain. To visualize this relationship, we plotted the visuals below.



Scatterplots displaying correlation between important predictors

Methodology

For this project, we built and evaluated the following regression models for predicting risk score:

- Linear Regression
- Ridge, Lasso, and Elastic Net Regression
- K-Nearest Neighbors (KNN) Regression
- Random Forest Regression
- Neural Net Regression

In addition to model building, we also performed feature analysis to determine and quantify the features that most impact the safety of a road segment. Both model building and feature analysis are discussed in the following sections.

Splitting the Data / Addressing Missing Data

Prior to building models, we split the data into a 70% training set and 30% testing set. The train_test_split() function from the sklearn library was utilized to randomly divide the data into training and testing sets. The training set was used for model building, and testing set was used for model evaluation.

Some of the variables in the dataset, such as speed limit and AADT, had a substantial proportion of road segments containing missing data. This is because these variables were obtained from the crashes dataset, which is far more sparse than the Michelin-provided trips and events datasets. To address this, we imputed missing values by aggregating on the road priority variable. For example, for road segments with missing speed limit value, we imputed the value using the median speed limit value for all other road segments with the same road priority as the segment with the missing value.

Models

The following models were built, trained, and evaluated:

Linear Regression

Linear regression is the most common predictive model. Linear regression models relationships using linear combinations of predicting variables, whose model coefficients are estimated from the training data. This linear model is then used to predict the response variable of a "new" datapoint, given the values of all predicting variables for the datapoint. In our modeling, we used the sklearn LinearRegression() function.

Ridge, Lasso, Elastic Net Regression

Ridge, lasso, and elastic net regression are all forms of linear regression, adjusted to reduce the effect of multicollinearity (strong correlations between predicting variables). Multicollinearity can cause large variance in the model, resulting in poor performance. Ridge regression addresses this by introducing an "L2 regularization" penalty term in the model that penalizes large regression coefficients, essentially "shrinking" all the coefficients except for the most important predictors. Similarly, Lasso Regression introduces an "L1 regularization" penalty term to the model that shrinks the coefficients for "unimportant" predictors all the way to zero, such that they are removed from the model entirely. Elastic Net Regression uses a weighted combination of the penalty terms from both lasso and ridge regression. In our modeling, we used the sklearn Ridge(), Lasso(), and ElasticNet() functions.

K-Nearest Neighbors (KNN) Regression

The KNN regression algorithm is the simplest algorithm explored in this project. The algorithm predicts risk score based upon the average of its k "nearest neighbors" in the p-dimensional space. As such, we can only use the numeric predictor variables in KNN. Our KNN model was tuned by evaluating the optimal k value, as well as the optimal weighting function used in the prediction. In our modeling, we used the sklearn KNeighborsRegressor() function.

Random Forest

Random Forest is an "ensemble method", meaning that the model prediction is obtained from an average of many predictions. In Random Forest, we utilize the bagging algorithm to draw a bootstrap sample from the training data and fit a base tree model to each bootstrap sample. Additionally, we randomly select only a portion of the predictors to use in each bootstrap sample. Our Random Forest model was optimized by tuning the number of trees to use in the model, the maximum depth of each tree, the minimum number of samples required to be at each leaf node, and the minimum number of samples required to branch an internal node. In our modeling, we used the sklearn RandomForestRegressor() function.

Neural Net

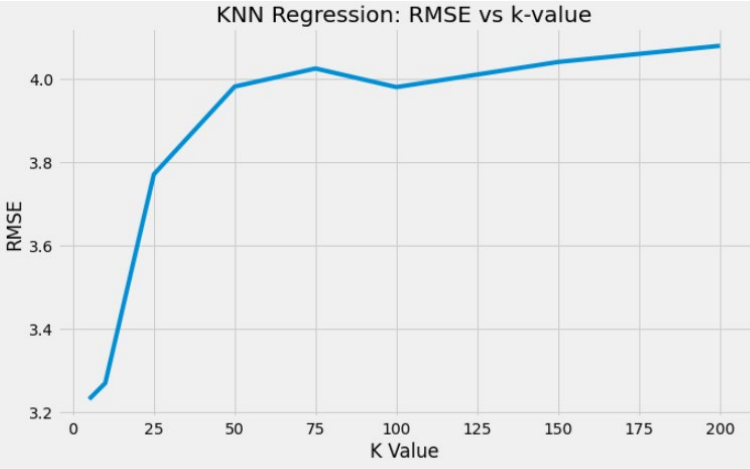
Neural networks have gained substantial popularity recently in the analytics world. They are complex models composed of many layers of "neurons" that process inputs to produce an output, modeled similarly to the human brain. Neural networks are most used for classification problems, but they can be applied to regression problems as well. Our neural net model was tuned by optimizing the L2 regularization parameter. In our modeling, we used the sklearn MLPRegressor() function.

Model Tuning

Initially, we built all models using their default parameters. Once that was complete, we further tuned the KNN, Random Forest, and Neural Net models to optimize performance.

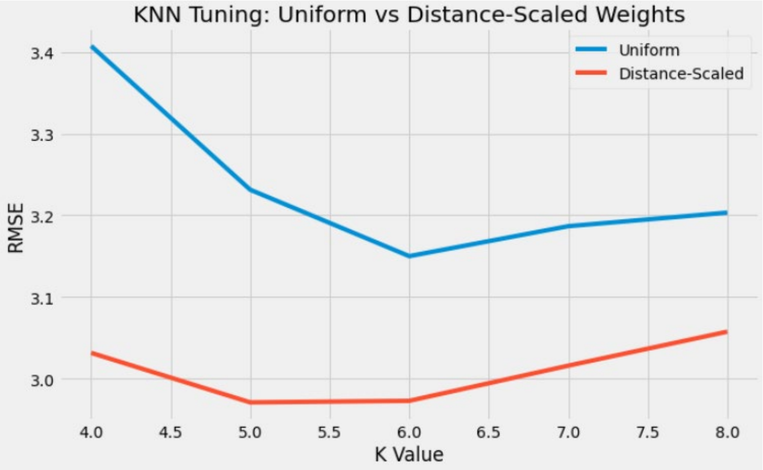
KNN Model Tuning

To tune the KNN model, we compared the cross-validated RMSE value of the model for a variety of different k-values, in a wide range [5, 200] to gain a rough approximation of the optimal k value.



Approximate k-value tuning

It is clear from the plot that the optimal k-value is somewhere in the low end of the range. Next, we further tuned the k-value by evaluating several values in the range [4, 8] as well as comparing uniform vs distance-scaled weighting metrics. Uniform weighting uses uniform weights for all neighboring values, whereas the distance-scaled weighting weights neighbors by the inverse of their distance to the datapoint, so that closer neighbors have a greater influence.



Final KNN model tuning

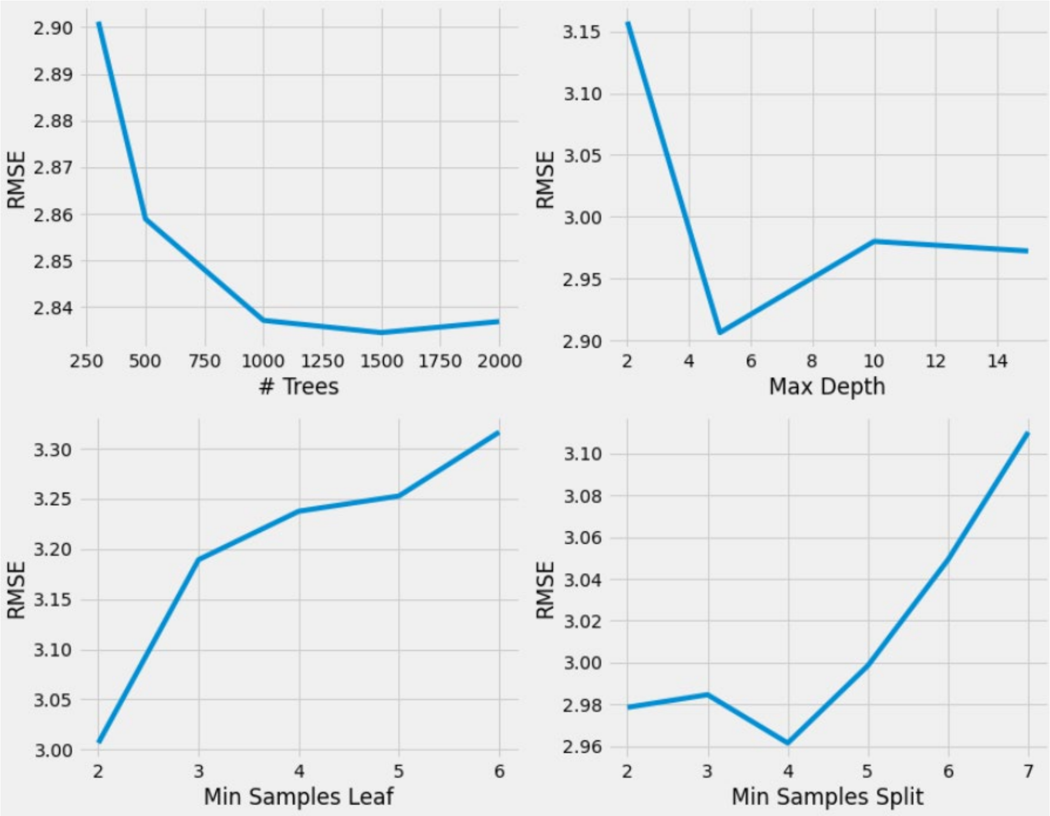
From the result above, we determined that the optimal KNN model should use a k-value of 5, and should use distance-scaled weighting.

Random Forest Model Tuning

The random forest model was tuned by optimizing the hyperparameters below:

- # Trees: the number of trees to use in the model
- Max Depth: the maximum depth of each tree

- Min Samples Leaf: the minimum number of samples required to be at each leaf node
- Min Samples Split: the minimum number of samples required to branch an internal node



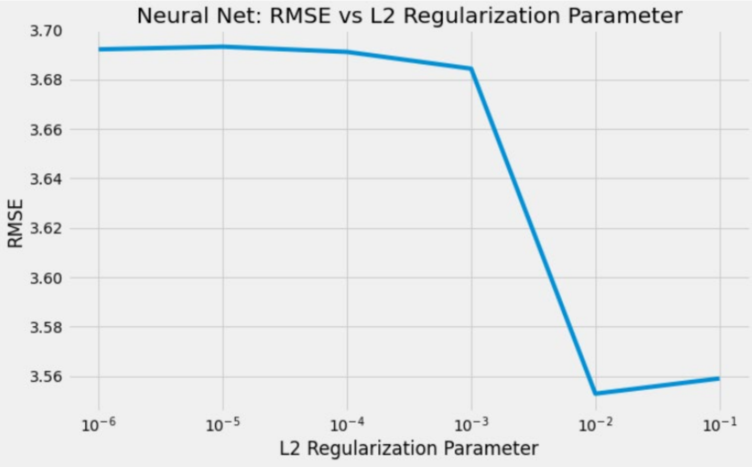
Random Forest model tuning

As is clear from the plots above, the random forest model is optimized for the hyperparameter values below:

- # Trees: 1500
- Max Depth: 5
- Min Samples Leaf: 2
- Min Samples Split: 4

Neural Net Model Tuning

To tune the neural net model, we considered only one hyperparameter to optimize: L2 regularization parameter (alpha). Like in ridge regression, this term is used to reduce variance in the model.



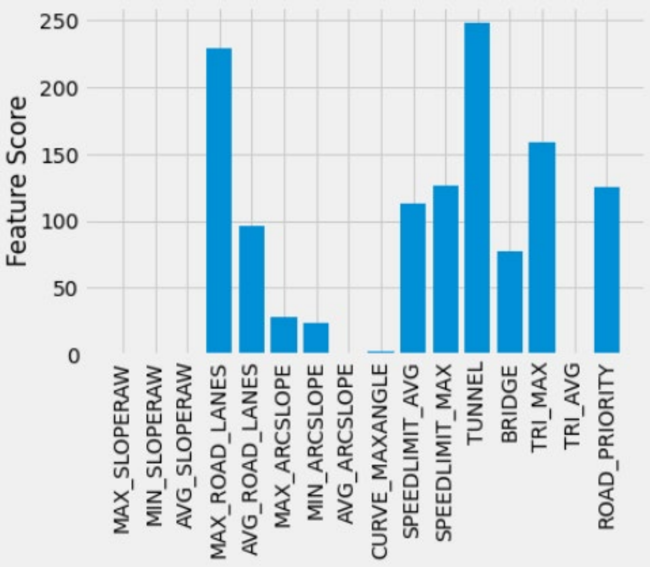
Neural Net model tuning

As can be seen in the figure above, the neural net model is optimized with an L2 regularization parameter of 0.01.

Feature Importance

Using unimportant features in our models can lead to unnecessary variance in the model, overfitting, and overall poor model performance. The first step in determining important features to use in the models, as discussed in the EDA section of this report, is to visualize the correlation between each predicting variable and the response variable. Our intuition tells us that the most important predictors will have the strongest correlation to the response variable. The predictors with the strongest correlation to risk score are: MAX_ROAD_LANES, AVG_ROAD_LANES, MAX_ARCSLOPE, MIN_ARCSLOPE, SPEEDLIMIT_AVG, SPEEDLIMIT_MAX, TUNNEL, BRIDGE, TRI_MAX, and ROAD_PRIORITY.

To validate the "correlation intuition", we next used the SelectKBest() function from the sklearn library. This function selects the "k" best features from set of predicting variables by determining a score based on a regression-based scoring function. The result can be found below.



Feature scores using SelectKBest() function

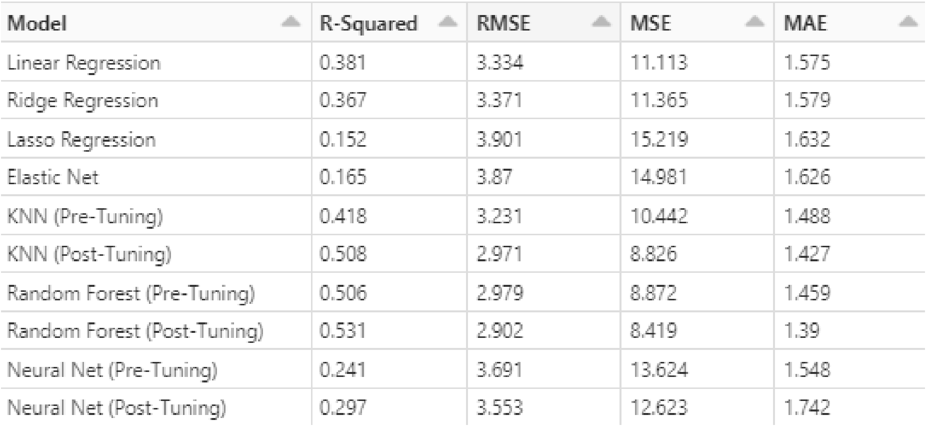
As can be seen above, our intuition regarding feature correlation holds true for feature importance. The most important features are MAX_ROAD_LANES, AVG_ROAD_LANES, MAX_ARCSLOPE, MIN_ARCSLOPE, SPEEDLIMIT_AVG, SPEEDLIMIT_MAX, TUNNEL, BRIDGE, TRI_MAX, and ROAD_PRIORITY. These are the features we used to build our models. Further evaluation of feature importance and contribution to risk score is discussed in the following "Analysis and Results" section.

Analysis and Results

Model Performance

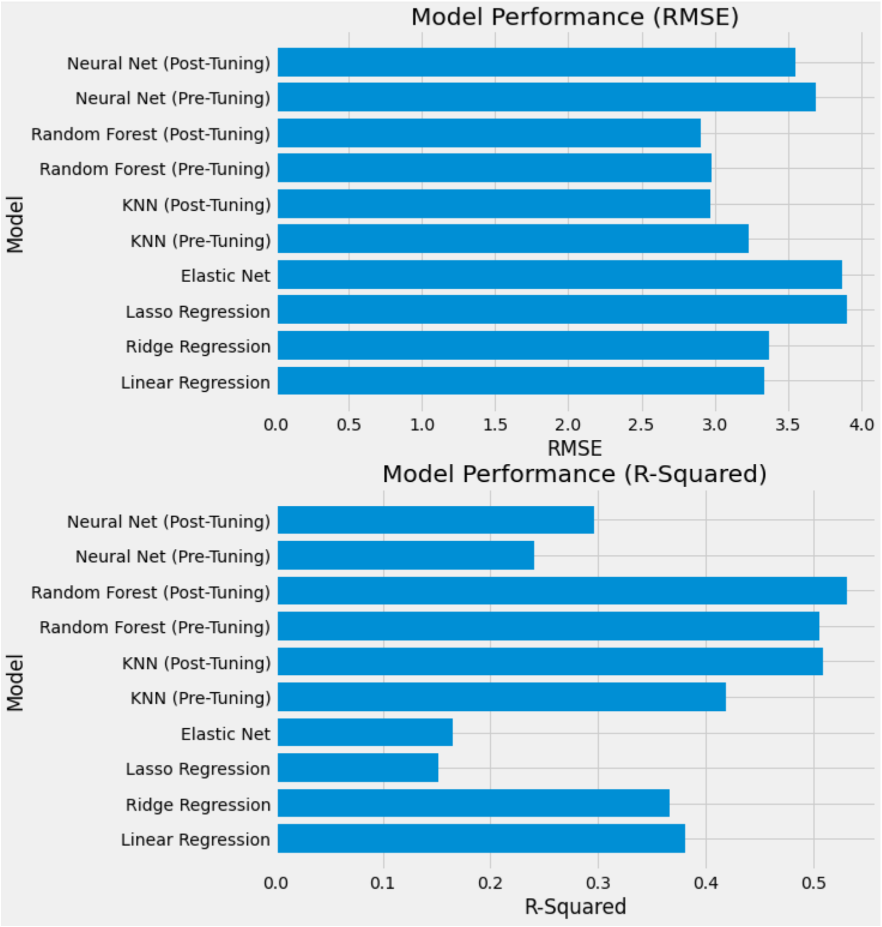
The comparative performance among all models is summarized below, using the following metrics:

- R-squared: the proportion of variance in the response that is explained by the model
- RMSE (root mean square error): square-root of the average of the squared error terms
- MSE (mean squared error): average of the squared error terms
- MAE (mean absolute error): average of the absolute values of error terms



Model	R-Squared	RMSE	MSE	MAE
Linear Regression	0.381	3.334	11.113	1.575
Ridge Regression	0.367	3.371	11.365	1.579
Lasso Regression	0.152	3.901	15.219	1.632
Elastic Net	0.165	3.87	14.981	1.626
KNN (Pre-Tuning)	0.418	3.231	10.442	1.488
KNN (Post-Tuning)	0.508	2.971	8.826	1.427
Random Forest (Pre-Tuning)	0.506	2.979	8.872	1.459
Random Forest (Post-Tuning)	0.531	2.902	8.419	1.39
Neural Net (Pre-Tuning)	0.241	3.691	13.624	1.548
Neural Net (Post-Tuning)	0.297	3.553	12.623	1.742

Model performance summary



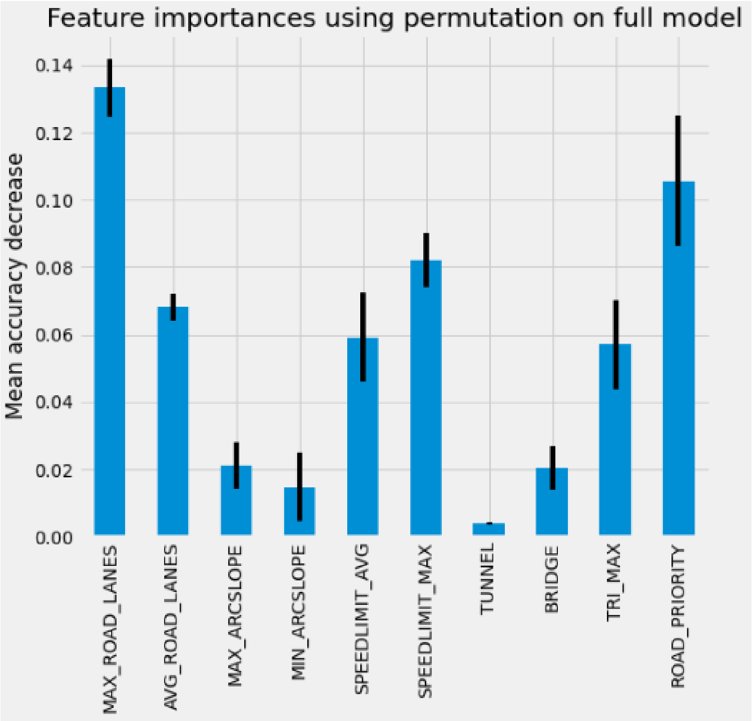
Model performance summary visualized

Best Performing Model: Random Forest (after hyperparameter tuning)

As can be seen in the summary above, the best performing model for predicting risk score was the Random Forest model (after tuning for optimization of hyperparameters). This model had both the lowest RMSE (2.902) and the highest R-squared (0.531). In addition to the Random Forest model, the KNN model performed very well after tuning to optimize hyperparameters, with an RMSE value of 2.971 and R-squared of 0.508.

Feature Importance

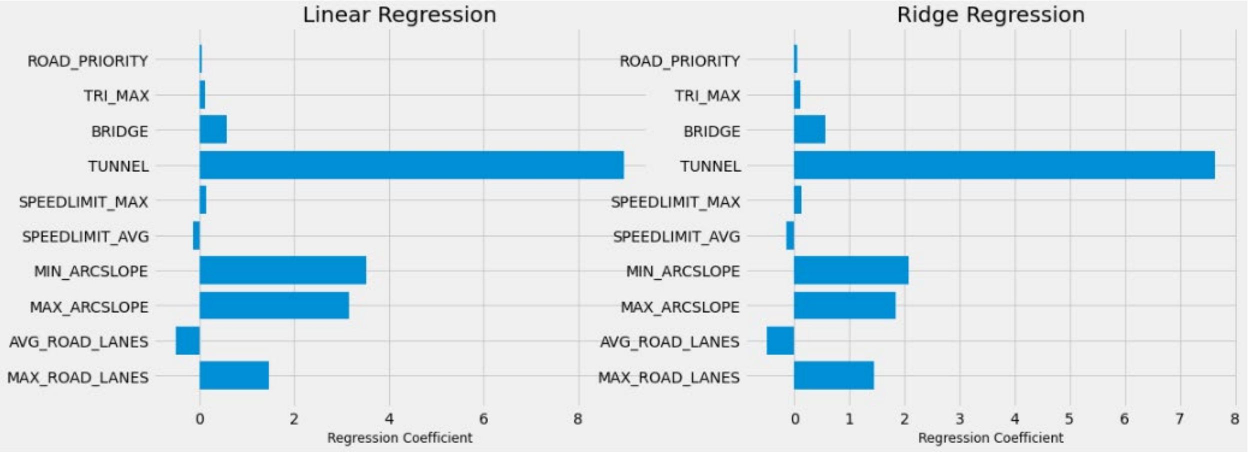
Because Random Forest was the best-performing model, we will first use the random forest model to evaluate feature importance. Feature importance in a random forest model is murky, as it is not possible to derive a coefficient for each predicting feature (like in linear regression). However, we can look at relative feature importance using the permutation feature importance method. Permutation feature importance is a method of determining feature contribution by determining the decrease in model score when a single feature value is randomly shuffled.



Feature importance in random forest model

As can be seen above, the three most important features to the model are: max road lanes, road priority, and speed limit. If these features are left out of the model, we expect a decrease in model score of 0.13, 0.11, and 0.08 for MAX_ROAD_LANES, ROAD_PRIORITY, and SPEEDLIMIT_MAX, respectively. Also important to the model are average road lanes, average speed limit, and max TRI. It appears that max/min arc slope, tunnel, and bridge may not be very important to the random forest model.

In addition to evaluating the feature importance of the random forest model, we also evaluated feature importance using the linear and ridge regression models. Even though these models did not perform quite as well as the random forest model, they can provide some more meaningful interpretation of the feature contribution to risk score by analyzing the regression coefficients for each feature.



Feature coefficients for linear/ridge/lasso/elastic net regression models

The ridge regression model has roughly the same relative model coefficients as linear regression (except all coefficients are "shrunk"), so we will focus on coefficients of the linear regression model for ease of explanation. Interestingly, the feature importance in these models seems to defer from the random forest model:

- The feature with the largest regression coefficient is TUNNEL, with a regression coefficient of ~9.0 for linear regression. This means that the presence of a tunnel is expected to increase risk score by 9.0 vs no tunnel present, holding all other features constant. However, this is a binary feature, so the maximum amount it can contribute to risk score is 9.0. This leads us to the conclusion that the presence of a tunnel significantly increases the risk of a road.
- Min and max arc slope were also significant contributors to the model, with coefficients of ~3.3 and ~3.1, respectively. This means that for a unit increase in min/max arc slope, the predicted risk score will increase by 3.3 and 3.1, respectively, holding all other features constant. We conclude that the slope of a road can significantly impact road safety.
- Max road lanes is also a significant contributor to the model. For each unit increase in number of road lanes, we expect risk score to increase by 1.5, holding all other features constant. As such, we conclude that roads with many lanes (such as large highways) can be substantially more dangerous than single-lane roads.

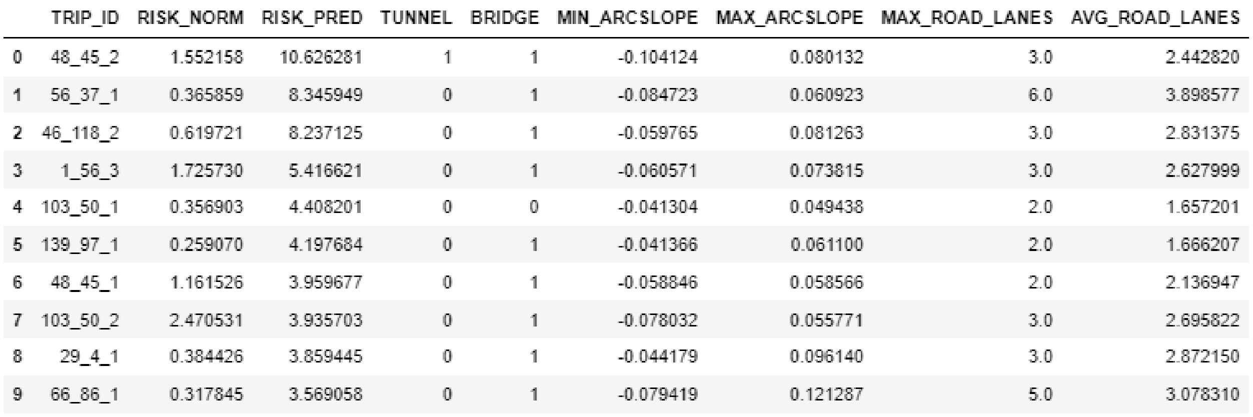
While the random forest model performed best, the problem with random forest is that it is difficult to surmise any clear interpretation of feature contribution to risk score. Although the linear regression model did not perform as well, it allows us to gain more meaningful insights to the contribution of each feature to risk score.

Route Ranking

The goal for route ranking is to rank the 10 least safe itineraries contained within a list of Atlanta-area routes provided by Michelin. Prior to our discussion of route ranking, let's define some terms:

- TRIP_ID: identifier for a specific itinerary between a starting and ending point. A trip is composed of a list of road segments that make up the complete itinerary.
- ROUTE: each ROUTE represents a starting and ending point. Each ROUTE can have multiple possible itineraries (ie, multiple possible TRIP_IDs).

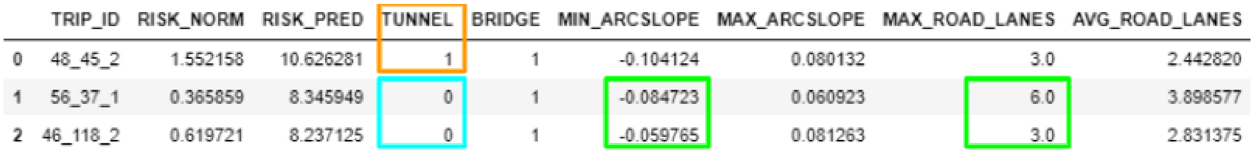
The route list provided by Michelin contains a total of 70 routes, with 139 total trips. To calculate the risk score for each TRIP ID, we aggregated the segment-level risk score (which we defined in the previous sections of this report) along all segments on each TRIP ID, and took the average. This is called "RISK_NORM" in the table below. Note that there are many segments found in the routes table with no record of trips data; for such segments a null risk score is assigned and as a result to compute trip/route level risk score, we applied a custom average function that ignores null values. In addition to the calculated risk score, we also predicted a risk score using our model, called "RISK_PRED" in the table below. The 10 least safe trips are the 10 TRIP IDs with the highest average predicted risk score for all segments on the trip:



	TRIP_ID	RISK_NORM	RISK_PRED	TUNNEL	BRIDGE	MIN_ARCSLOPE	MAX_ARCSLOPE	MAX_ROAD_LANES	AVG_ROAD_LANES
0	48_45_2	1.552158	10.626281	1	1	-0.104124	0.080132	3.0	2.442820
1	56_37_1	0.365859	8.345949	0	1	-0.084723	0.060923	6.0	3.898577
2	46_118_2	0.619721	8.237125	0	1	-0.059765	0.081263	3.0	2.831375
3	1_56_3	1.725730	5.416621	0	1	-0.060571	0.073815	3.0	2.627999
4	103_50_1	0.356903	4.408201	0	0	-0.041304	0.049438	2.0	1.657201
5	139_97_1	0.259070	4.197684	0	1	-0.041366	0.061100	2.0	1.666207
6	48_45_1	1.161526	3.959677	0	1	-0.058846	0.058566	2.0	2.136947
7	103_50_2	2.470531	3.935703	0	1	-0.078032	0.055771	3.0	2.695822
8	29_4_1	0.384426	3.859445	0	1	-0.044179	0.096140	3.0	2.872150
9	66_86_1	0.317845	3.569058	0	1	-0.079419	0.121287	5.0	3.078310

Top 10 least safe trips

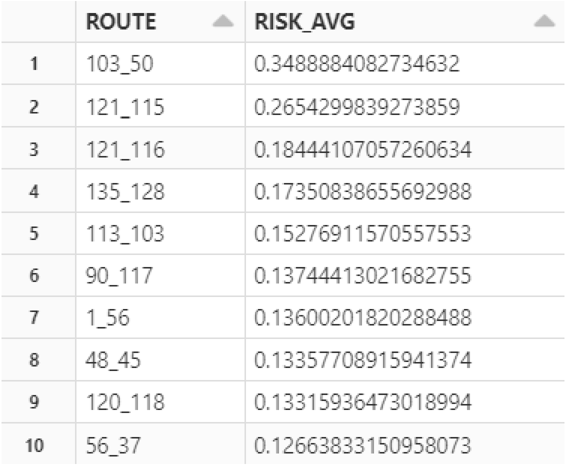
Comparison of the most important features provides insight into the model's predicted risk score for each trip. For example, TRIP ID 48_45_2 is the riskiest route because the route has tunnels, which was the most influential feature for predicting risk score. In contrast, TRIP ID 56_37_1 is riskier than 46_118_2 because even though neither trips have tunnels, 56_37_1 has a higher MIN_ARCSLOPE and MAX_ROAD_LANES than 46_118_2.



	TRIP_ID	RISK_NORM	RISK_PRED	TUNNEL	BRIDGE	MIN_ARCSLOPE	MAX_ARCSLOPE	MAX_ROAD_LANES	AVG_ROAD_LANES
0	48_45_2	1.552158	10.626281	1	1	-0.104124	0.080132	3.0	2.442820
1	56_37_1	0.365859	8.345949	0	1	-0.084723	0.060923	6.0	3.898577
2	46_118_2	0.619721	8.237125	0	1	-0.059765	0.081263	3.0	2.831375

Critical feature comparison of top 3 riskiest trips

Next, we identified the top 10 least safe ROUTES. To do so, we aggregated the trip-level average risk score for all trips on each ROUTE, and took the average. The 10 least safe routes are the 10 routes with the highest average aggregated risk score for all trips on the route:



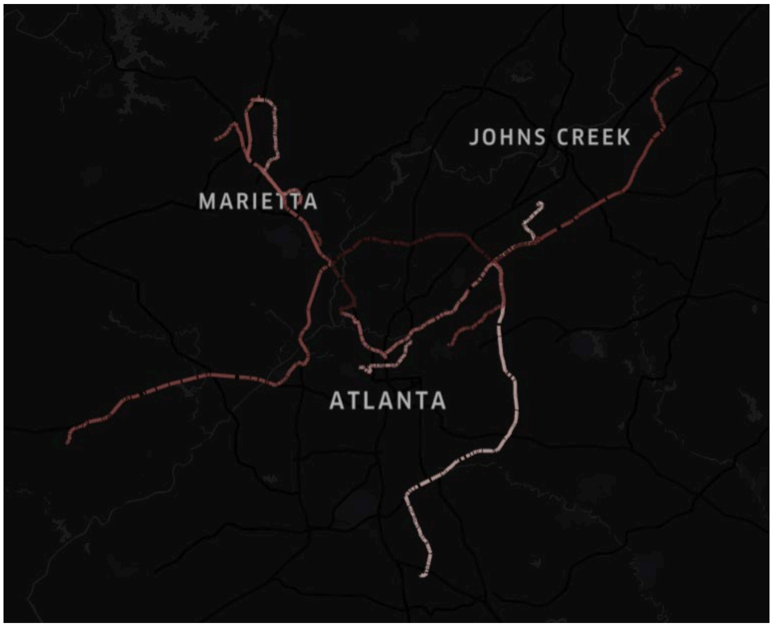
	ROUTE	RISK_AVG
1	103_50	0.3488884082734632
2	121_115	0.2654299839273859
3	121_116	0.18444107057260634
4	135_128	0.17350838655692988
5	113_103	0.15276911570557553
6	90_117	0.13744413021682755
7	1_56	0.13600201820288488
8	48_45	0.13357708915941374
9	120_118	0.13315936473018994
10	56_37	0.12663833150958073

Top 10 least safe routes

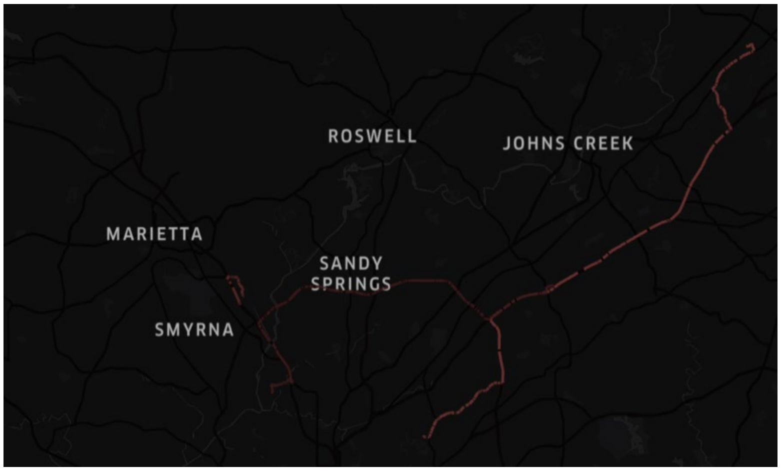
Closing Statements

Proposed Implementation

For visualization, we mapped the 10 least safe itineraries on a map. These itineraries are color-coded with darker color representing higher risk.



Top 10 least safe itineraries on a map (increasing risk goes from light to dark brown)



Top 3 least safe itineraries on a map

The mapping above is representative of how our risk score prediction could be incorporated into a modern GPS routing application to help drivers evaluate route safety. Given multiple itinerary options between two locations, a risk score for each itinerary would be provided to the user. This allows drivers to weigh both trip time and safety as parameters for deciding on a route.

A Michelin fleet could use this risk score metric to implement policy preventing drivers from taking any itineraries that exceed a certain risk score threshold. This would reduce cost, improve productivity, and ensure safety of drivers.

Lessons Learned

The key analytics concepts utilized throughout this project were:

- Methods for exploring data to identify patterns / correlations and other useful information for identifying an appropriate analytics solution.
- Understanding the theory behind several regression models including Linear Regression, Ridge / Lasso / Elastic Net Regression, KNN, Random Forest, and Neural Net.
- Methods for training, optimizing, and evaluating model performance.
- Evaluating / interpreting features to determine their contribution to a regression prediction outcome.
- Large-scale data analysis using scalable tools deployed in a professional environment.
- Geo-spatial data analysis.
- Incorporating data analysis / modeling into a feasible, actionable solution while considering business objectives.
- Data visualization to provide a business-friendly explanation of our solution.

Citations

Bieber, Christy. "Car Accident Statistics for 2023." *Forbes*, Forbes Magazine, 18 July 2023, www.forbes.com/advisor/legal/car-accident-statistics/.

Gopin, Michael J. "How Much Do Motor Vehicle Crashes Cost Americans Every Year?" *Law Offices of Michael J. Gopin, PLLC*, 23 May 2023, www.michaelgopin.com/blog/how-much-do-vehicle-crashes-cost-americans-every-year/#:~:text=Cost%20of%20Vehicle%20Collisions%20Each,for%20a%20family%20of%20four.

Appendix

Workload Distribution:

Task	Description	Owner
ETL + Data Aggregation	Approach 1 - aggregate data on segment ID	Michael Crist
	Approach 2 - aggregate data with risk related features at road id level.	Vishal Jada, Hersh Gupta
EDA	Visualize correlations, relationships w/ outcome variable	Michael Crist
Midterm Report	Prepare midterm report	Michael Crist, Hersh Gupta, Vishal Jada
Model Building	Perform cross validation, hyper parameter tuning	Michael Crist
	Outlier detection & clustering	Hersh Gupta

Route Ranking	Apply risk metric to rank least safe routes	Michael Crist, Vishal Jada
Conclusions / Final Report	Perform model evaluation, interpretation, draw conclusions, and prepare final report	Michael Crist, Hersh Gupta, Vishal Jada