HEC Montreal
Msc. Data Science and Business Analytics

Road Safety Pilot Project

Technical Report
MATH 60611A
Advanced Statistical Methods

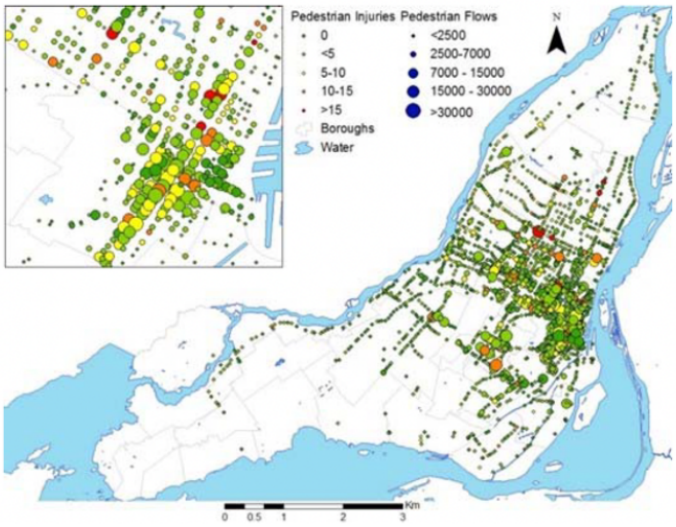

Figure 1: Map of Montreal with intersections of interest.

Hair Albeiro Parra Barrera, Charles Julien
March 9, 2024

Contents

1 Mandate

Faced with the constant increase in the number of accidents involving pedestrians on its road network, the city decided to carry out a study to assess the level of risk at road intersections. To do this, a sample of 1864 signalized intersections was selected (figure 1). This sample represents 80% of all signalized intersections on the island of Montreal. For each intersection, the city has the total number of accidents involving pedestrians and reported by the police (data from the SAAQ) in the last 10 years. In addition, Department of Transportation engineers collected the following data:

- Traffic data on the average daily flows of pedestrians and vehicles at each intersection.
- Geometry data (design) for each intersection including the number of lanes, types of signals, layout for left turns, parking restrictions, etc...
- Road safety performance measures calculated by engineers for each intersection, based on the number of potential daily interactions between pedestrians and vehicles, depending on the directions and movements of the vehicles.

Objective: Modeling of the accidents observed in the last 10 years to provide the city of Montreal with a ranking of the 1864 intersections in terms of safety (from the most dangerous to the least dangerous), so that it can prioritize the riskiest intersections with the aim of improving infrastructure.

1.1 Scope of Work

1. **Data Cleaning & Feature Engineering**: In order to correctly analyze our data, we make sure to perform a number of transformations and corrections to our data such as standarization, missing values imputation etc. We also extend our predictors through simple transformations of certain variables.
2. **Exploratory Data Analysis**: We study linear two way correlations between the data covariance and the target, and inspect seasonality of the accidents, and distributions of specific covariates of interest. We address this more in depth in the **Data** section.
3. **Variable Selection:** We employ a number of variable selection methods including correlation-based, Lasso, VIMP methods and regression boosting. These selected variable subsets are then used in combinations with different models for model selection.
4. **Model Selection:** In combination with the different sets in the variable selection section, we test a different of linear and non-linear methods and compare them together for best model selection. Our modelisation procedure is described in the **Methodology** section.
5. **Intersections Ranking:** Using our "best" model from the prevoius step, we proceed to rank the intersections, thus completing the mandate.

2 Data

2.1 Description

The original data contains a number of variables including meta data such as the intersection id, latitude, longitude and the street names, date of where the safety measures where recorded, and a number of safety measures which function of the intersections, flow of pedestrians and vehicles, number of lanes, exclusive turns, the Montreal borough where the intersection occurred, numerical scaling and simple transformations of the same, and more. These variables are often coded numerically, although several represent categorical measures.

2.2 Data Cleaning & Feature Engineering

We perform a number of data cleaning steps described below:

1. **Separate meta data**: We keep a separate table with the streets names corresponding to each intersection id. We do not consider these variables as predictors in our analysis.
2. **Irrelevant variable removal:** We remove corrupted, duplicate and redudant variables such as `street_1` (same as `rue_1`), `traffic_10000` (scaled version of `traffic`), etc.
3. **Borough names cleaning:** In this step we correct all the borough names to correct UTF format for convenience and clarity.
4. **Correct datatypes:** We ensure that both numerical and categorical variables are correctly coded as such.
5. **NA removal:** Since after the previous steps the resulting data does not contain more than a very small percentage of NAs for the variable `ln_distdt`, we fill them with zeroes. As these observations are actually located downtown ($\ln(0) = \text{NA}$).
6. **Date variables and date imputation:** We treat the `date_` variable as a potential covariate in our analysis. In particular, we consider the `month` and **day of the week** (`dow`) with the hypothesis that seasonality might be a good predictor of accidents. As a small portion of these dates are missing, we employ **KNN imputation** [1] based on the rest of the datapoints to obtain the missing dates as a proxy. Our hypothesis is that similar datapoints are reflective or similar characteristics for most measurements. For details on the procedure, see **Appendix**.
7. **Borough grouping:** Through exploration, we noticed that several levels of the categorical variable 'borough' do not contain enough observations. Since our final aim is a more flexible model, we group the levels to form a more informative covariate in terms of predictive power.
8. **Standarization:** Once all preprocessing and cleaning steps have been applied, we standarize all covariates with the formula:
$$x_{ij} := \frac{x_{ij} - \overline{x_j}}{s_j}$$
, where $\overline{x_j}$ and s_j are the corresponding covaraiate's training sample mean and standard derviation.

[1] We note that, as expected, we also find a strong correlation between the distribution of the dates before and after imputation. See the **Exploratory Data Analysis** section.

9. **Feature expansion:** For a number of selected features, we also include (mostly) 2n'd degree polynomial expansions:

   - **Traffic flow variables:** Squared `pi` and `fi`.
   - **Distance from Downtown:** Squared and cubed `distdt`.
   - **Crosswalk and Road Widths:** Squared `tot_crossw`, `avg_crossw` and `tot_road_w`.
   - **Turning Flows:** Squared `fli` [2], `fri`[3] and `fti`[4]

2.3 Exploratory Data Analysis

In this section, we present visualizations of a number of selected variable distributions that we consider of high importance for our analysis.

2.3.1 Accidents distribution by borough

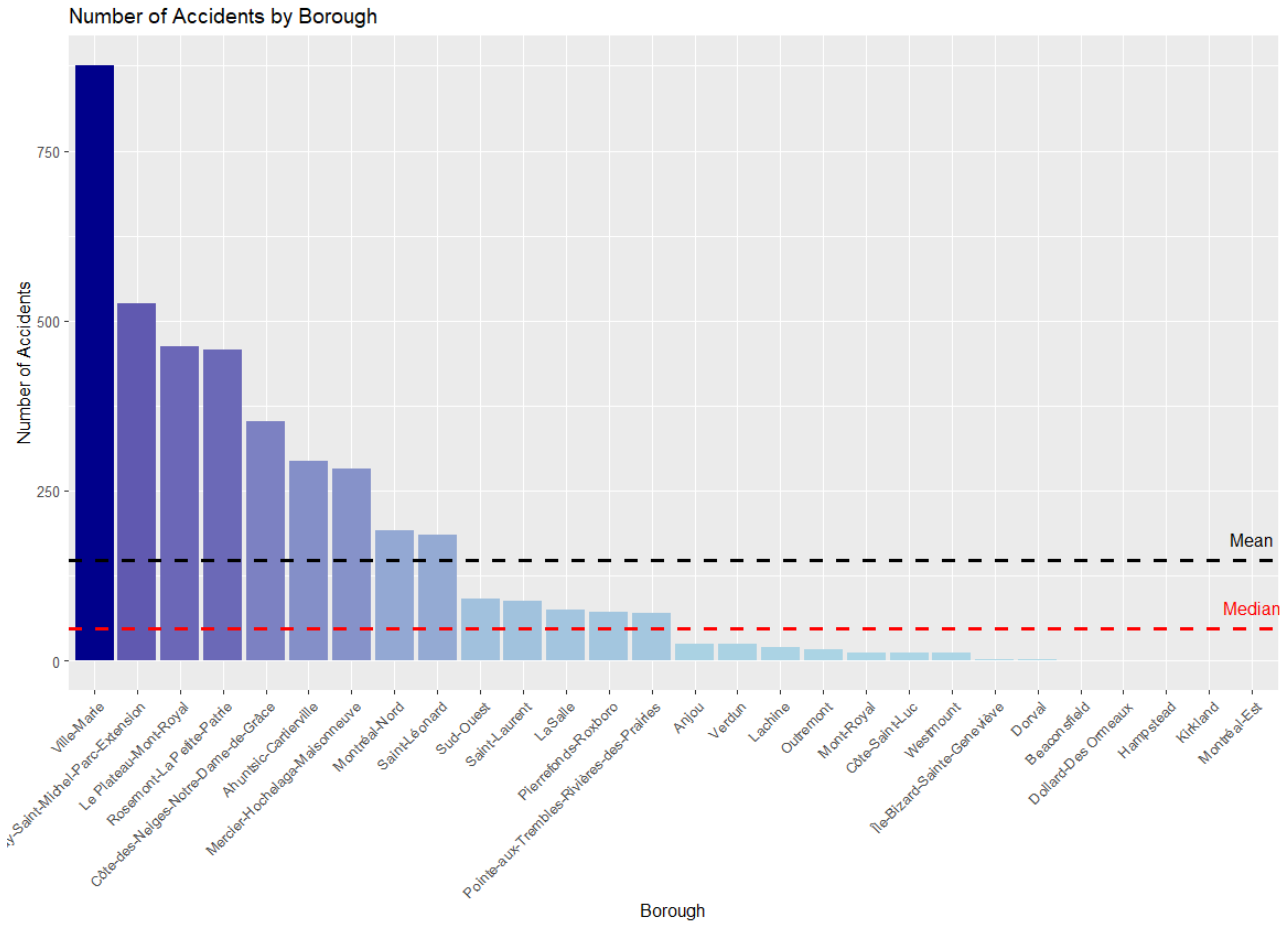

Figure 2: Distribution of Accidents per borough without level grouping. Note that we use the grouped version of the `Borough` covariate since as it can be noted, several boroughs contain a relative insignificant amount of accidents in the dataset.

[2]Average annual daily flow for vehicules turning left at the intersection.
[3]average annual daily flow for vehicules turning right at the intersection.
[4]average annual daily flow for vehicules going through (straight) at the intersection.

2.4 Seasonality of Accidents

It's important to highlight that missing values in the date variable were not missing at random. Their presence or absence demonstrated statistical significance in a simple linear regression predicting the accident variable, indicating a potential relationship between the timing of safety measure recordings and the observed number of accidents. In the subsequent plots, we investigate this potential relationship.

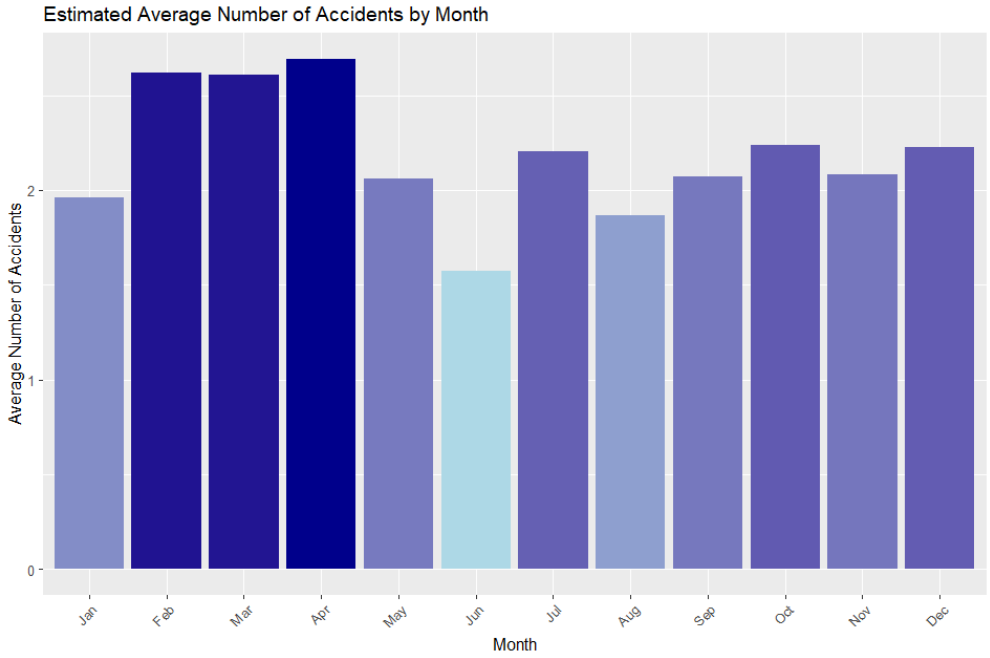

Figure 3: Distribution of recorded accidents by month where safety measure were taken. We note that in this distribution we have already used the KNN Date Imputation method mentioned in a previous section.

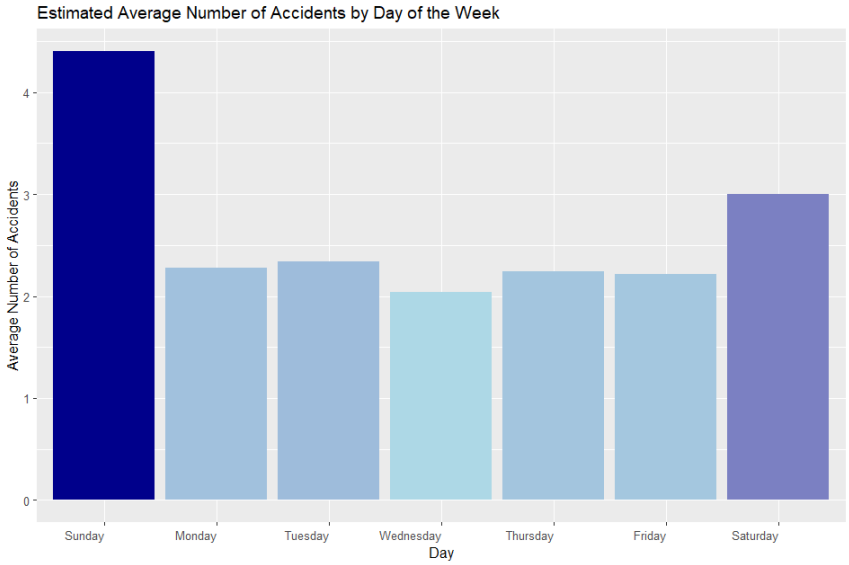

Figure 4: Estimated distribution of average number of accidents per day of week where safety measurements were recorded.

3 Methodology

The primary aim is to rank intersections based on safety, prioritizing improvements for high-risk intersections to enhance overall infrastructure safety.

To achieve this, we defined intersection safety as a metric quantifying the total risk taken by all individuals crossing the intersection over an almost infinite timeframe. Notably, this definition differs from assessing an individual's risk when crossing a specific intersection. The former focuses on risk at a population level, aligning with Montreal city's authorities concerns.

Our methodology rests on the assumption that the frequency of accidents at an intersection over a ten-year period provides a somewhat noisy yet informative indicator of its overall safety.

Formally, let Y_{acc} be the number of accidents at any given time. Given a true generating process f and a number of true factors $X_1, \ldots, X_q$, we model the number of accidents by

$$Y_{acc} = f(X_1, \ldots, X_q) + \epsilon$$

, where ϵ is random noise with $\mathbb{E}[\epsilon] = 0$ and $\mathbb{V}ar[\epsilon] = \sigma^2$. We approximate this function by

$$\hat{Y}_{acc} = \hat{f}_\theta(X_1, \ldots, X_p)$$

, where $\hat{f}_\theta$ is some model (linear, regularized linear, glm, random forest, etc), and $\{X_1, \ldots, X_p\}(p << q)$ is a subset of the available variables in the data. Estimating the function f can be done using statistical learning methods and will allow to learn a function $\hat{f}_\theta$ to rank intersection based on their safety. Indeed, the best estimate of the safety function is a function that minimizes the expected prediction error, hence a function that minimizes the error due to bias and variance, that is:

$$\hat{f}_\theta = \text{MSE}(\tilde{f}_\theta) = \arg\min_{\tilde{f}_\theta} \mathbb{E}[(Y - \tilde{f}_\theta(\mathbf{X}))^2] = \arg\min_{\tilde{f}_\theta} \text{Bias}^2(\tilde{f}_\theta) + \mathbb{V}ar(\tilde{f}_\theta) + \sigma^2$$

Alternatively, this can be seen as removing the noise from the observed count of accidents to truly asses which intersection are safe based on their features.

It is to notice that in line with the mandate, our objective is solely to obtain the most effective predictive model for ranking intersections. As a result, we do not seek value in interpreting the covariates or ensuring the inclusion of specific covariates in the model.

To obtain the best safety function estimate, our methodology consists of trying different models on different subsets of selected variables using a variety of methods to predict the variable accident and identify the one that generalizes the best. Once the function is properly estimated, it can be used to estimate each intersection's safety and gives us a basis on which to rank them.

3.1 Framework

Our framework for this analysis was described in section 1.1, but we include a visual flowchart for clarity in figure 5. The details of the the **Train-Validation Split**, **Variable Selection** and **Model Selection** are given in the following sections.

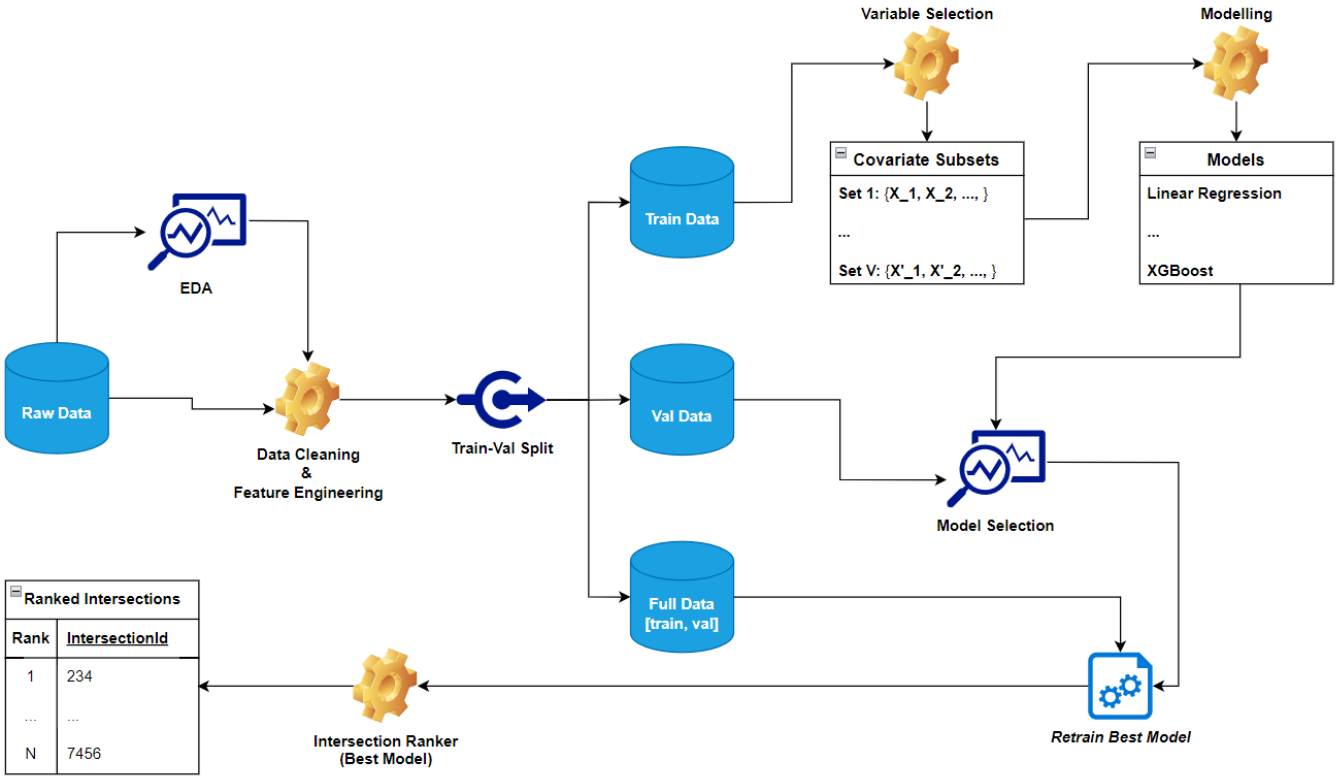

Figure 5: Workflow architecture of the analysis.

3.2 Train-Validation Split

In order to correctly perform variable and model selection, we based ourselves on the assumption that the distribution of the accidents given by the previous 10 years also reflects the distribution of accidents in the few upcoming years. Thus, since we do not necessarily consider temporal order in our data $\mathcal{D}$, we follow the approach below

1. Randomly shuffle the given data $\mathcal{D}$
2. Perform an 80:20 train-validation split, obtaining $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$
3. Use $\mathcal{D}_{train}$ for feature selection, and also to train models at the model selection step. The best models are chosen based on their MSE performance on the $\mathcal{D}_{val}$ set.
4. Hyperparameter tuning is also done through cross-validation on the $\mathcal{D}_{train}$ set only, and the models are retrained on the same set.

A test set was deemed not necessary as the specific measure of performance of the model is of secondary interest, as we maintain the assumption that the data sample available represents closely measure in the future, and the ranking is not meant to be a predictive model of new accidents in the future, but rather the general "dangerousness" of the intersections at hand.

3.3 Variable Selection

As shown in the workflow portrayed in figure 5, using the set $\mathcal{D}_{train}$ only we proceed to produce multiple feature subsets using a variety of methods. These methods are described below.

1. **No Selection:** All features were kept.
2. **Stepwise AIC/BIC:** Features are added and removed iteratively if they improve the AIC or BIC of the model until convergence.
3. **Lasso Selection:** L1 penalty is applied to the coefficients estimates in a given model. This shrinks some coefficient to zero and thus selects features. This was performed on gaussian and poisson distribution.
4. **Random Forest VIMP:** By permuting features one by one in a random forest and measuring the impact on the performance, it is possible to derive features importance. Then, by selecting the top n most important predictors, we get a subset of variables.
5. **Correlation Importance:** Similarly to sure independence screening, we measure the correlation between the target and all covariates. Then we select the top n most highly correlated covariates as the subset.
6. **Boosting with simple regression models:** This consists of iteratively adding models with one covariates for a specified number of time. At each step, the chosen covariate is the one that best explains the residuals. When the algorithm converges, some variables are selected and others are not.

3.4 Model Selection

Once the feature subsets have been determined, we pair them with a number of models, thus creating the final model which will be compared to all the other models on the $\mathcal{D}_{val}$ set for best model selection. As mentioned before, any hyper-parameter tunning is done through cross-validation on the train set, often handling internally by R's packages' functions. Below, we provide a brief description of the different models employed, independent of the feature sets:

1. **Baseline:** This model simply predicts $\hat{y} = \widehat{\mathbb{E}}[Y] = \frac{1}{n}\sum_i y_i$ for all i (taining sample mean).
2. **OLS:** Ordinary least square with all covariates in the subset.

$$\hat{y} = x\hat{\beta} \qquad \text{where } \hat{\beta} = (X^T X)^{-1} X^T Y$$

3. **Lasso:** Regularized linear regression with $L1$-penalty (note: the data is standardized):

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - (\beta_0 + \beta^T x_i))^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

where λ is estimated by cross validation and predictions are made like in ordinary least squares.

4. **Regularized GLM Poisson:** Since the target can be seen as a count frequency, we employ the Poisson regression model, which models $\mathbb{E}[Y|X] = \lambda = \exp\big(\beta_0 + \beta^T X\big)$. In Elasticnet-penalty form, the fitted coefficients are expressed as:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} w_i \left[e^{\beta_0 + \beta^T x_i} - y_i(\beta_0 + \beta^T x_i) \right] + \lambda \left[(1-\alpha)\, \frac{1}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right] \right\}$$

5. **Random Forest:** This model is an ensemble of trees where the final prediction is the mean of all the trees' predictions. Each tree is fitted using CART algorithm with a subset of the variables at each split and a bootstrap sample. A single tree's prediction can be written as:

$$\hat{y}_i = \sum_{j=1}^{m} d_j \mathbb{I}\{X_i \in R_j\}$$

   Where $\mathbb{I}\{X_i \in R_j\}$ returns 1 if a given observation belongs to region j of the partition, 0 otherwise and d_j is the average value of the target in region j.

6. **Gradient boosting:** Different methods were used for boosting including trees, linear regression and poisson regression. Boosting consists of combining weak learners to create a stable model. Each learner tries to predicts the negative gradient of the loss with respect to the current model. It is then added to the model with a shrinkage factor for a given number of iterations.

7. **XGBoost:** Extreme Gradient Boosting (XGBoost) is a powerful and efficient implementation of the gradient boosting framework. It builds upon the concept of gradient boosting by adding regularization and parallelization, making it extremely fast and scalable. XGBoost minimizes the following objective function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{K} \Omega(f_i)$$

   where:

   - l is the loss function that measures the difference between the prediction $\hat{y}_i$ and the true label y_i.
   - $\Omega(f_i)$ is the regularization term that penalizes complex models. It is a sum of penalties over all the leaves in all the trees (K is the number of trees).

3.5 Results

After fitting several combinations of selected feature subsets and statistical models, we measure their performance in terms of MSE on the validation set we set aside earlier, and rank them in decreasing order. These results are show in in table 1. We find that the tree based methods are performing the best **f**ollowed by boosting of poisson model and regularized poisson in GLM.

Model Performance			Model Performance		
Model Name	MSE	RMSE	Model Name	MSE	RMSE
Tuned XGBoost	**5.077557**	**2.253343**	GLM_Poisson_L1 + Refit	6.211458	2.492280
RandomForest	5.208176	2.282143	Corr + OLS	6.402864	2.530388
Boosted Poisson	5.371303	2.317607	Lasso + Lasso (1-SE rule)	6.523921	2.554197
GLM Poisson L1	5.854670	2.419643	Stepwise BIC + Lasso	6.743017	2.596732
Lasso + Lasso	5.934364	2.436055	Stepwise AIC + Lasso	6.873076	2.621655
OLS	5.950870	2.439441	Stepwise-BIC + OLS	6.893511	2.625550
Lasso + OLS	5.971233	2.443611	Stepwise-AIC + OLS	7.816796	2.795853
RF VIMP + Ridge	6.004438	2.450395	**Baseline**	**9.523514**	**3.086019**
GLM_Poisson_L1+L2	6.056684	2.461033			

Table 1: Model Performance Comparison on the Validation Set. The + symbolizes the variable selection method followed by the fitting method employed. Its absence means all variables were used.

While no testing sample was employed to gauge the unbiased performance of the top-performing model, the experiment was replicated using a different seed state to assess the consistency in the ranking of models. The overall order demonstrated notable similarity, underscoring the stability of the results.

Given that the target variable "accident" is a count variable, the optimal models exhibit a logical inclination toward tree-based methods. These models excel in navigating non-linearities and intricate relationships within the data. Additionally, Poisson models emerge as strong contenders, leveraging their inherent capability to effectively model non-negative integer values, aligning seamlessly with the nature of the count variable under consideration.

3.6 Intersection Ranking

Once the best model as been determined, it is retrained on the whole dataset. Then, it is used to predict the safety of the intersections on the same dataset. Those with higher accidents numbers will be deemed more dangerous. To compare the obtained ranking with the ranking from the observed label, we use Spearman rank correlation, given that we care that the predicted values for the number of accidents by our model are monotonically increasingly correlated, but not too close or too far to the recorded ones (true values). The Spearman correlation is calculated via

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

, where d_i is the difference between the ranks of corresponding observations, and n is the number of observations. Our rankings demonstrate a Spearman correlation of approximately 0.682, indicating a suitable modeling of the accidents. It's important to note that models with higher variance, signifying greater capacity to fit the data, will naturally result in higher correlations in the in-sample. Consequently, this measure is employed more as a sanity check rather than a definitive evaluation of performance."

int_no	ranking	acc	predicted_acc	int_no	ranking	acc	predicted_acc
931	1	15	13.98221	778	6	13	11.54115
481	2	20	13.03810	938	7	23	11.45697
928	3	17	12.81041	863	8	12	11.03188
601	4	22	12.68137	1092	9	12	11.02845
633	5	13	11.79596	906	10	29	11.00377

Table 2: Top 10 ranked intersections with the XGboost model. The variable `acc` is the true number of accidents, and `int_no` corresponds to the intersection id.

4 Conclusion

Our study aimed to assess the level of risk at road intersections in Montreal by modeling accidents observed over the last 10 years. Through various statistical learning methods and feature engineering techniques, we ranked 1864 intersections in terms of safety, providing the city with a prioritized list for infrastructure improvement. The Extreme Gradient Boosting model performed the best in terms of mean squared error on the validation set. Retrained on the entire dataset, this model was used to rank intersections based on predicted safety levels, showing a Spearman correlation of approximately 0.682 with observed accident counts. This indicates the model's reliability in ranking intersections and providing insights for urban planners and policymakers to enhance road safety measures.

5 Appendix

Algorithm 1 KNN Date Imputation

Require: Data frame D, Numerical variables N
Ensure: Data frame D with imputed date column
Preprocessing:
Convert valid date strings to Date objects and assign NA to invalid dates in D
Convert dates to numeric format (e.g., days since a reference date)
Standardize numerical variables in D using columns N
KNN Imputation:
Prepare a data frame I for imputation, including the numeric date column and other relevant numeric columns from N
Set the number of neighbors k
for each instance i with missing date in I **do**
    Find the k nearest neighbors of i in I based on the Euclidean distance in the feature space of N
    Impute the missing date of i by averaging the dates of the k nearest neighbors
end for
Postprocessing:
Update the date column in the original data frame D based on the imputed values in I
Optionally, remove any temporary columns added during preprocessing
return D