Analysis of Collision Type in Seattle with Tree-based Methods and Multinomial Regression

Aishwarya Saibewar, Karthika Selvaraj, Xuanzhi Li

**Abstract:**

Predicting the collision type helps in finding the most common type of collision happening in the city. This study helps in the identification of the factors contributing to the collision and to take necessary precautions to avoid collisions using Machine learning techniques. The Decision tree model is built using Seattle’s Collision dataset to analyze the various factors contributing to the city’s collision rate. Further, exploring other methods of bagging boosting, random forest, and logistic regression to improve the model performance in predicting the collision type. The random forest model with the sample size of predictors as the square root of the total number of predictors was found to perform best at predicting the collision type.

**Introduction:**

All over the world, “drive safe” is a common saying we hear when our loved ones or friends are traveling. This in itself explains the gravity of accident patterns in general. Commonly while we are driving, we see a speed limit sign that is designed for that specific type of road and weather conditions. Irrespective of all the safety measures taken by the governments and the drivers themselves, we still see accidents on a daily basis. This highlights the need for developing an algorithm to predict the type of collision by narrowing down the best factors that can influence the collision type. In this paper, we will be using Seattle’s Collision dataset to train the model to make predictions.

**Theoretical Background:**

In this paper, tree-based models are used to predict the type of collision. Tree-based algorithms are supervised learning methods that could be applied for both regression and classification problems. These include fragmenting the predictor space into several sections. To produce a forecast for a given observation, the mean or mode response values for the training observations in the region to which it belongs are utilized. These types of approaches are known as decision tree methods, as the set of division rules used to segment the predictor space can be summarized in a tree. Two entities, internal nodes and leaves (or terminal nodes) can be used to explain the tree. The decisions or final outcomes are represented by the leaves, and the data is separated at the internal nodes. To prevent overfitting of the training data, a pruning method is applied. It reduces the decision tree's size and creates a subtree to balance variation and bias. However, there are a few drawbacks with decision trees, i.e., they are prone to overfitting and, therefore, ultimately lead to wrong predictions. While pruning is a good way to improve a decision tree model's predictive ability, a single decision tree model will not yield strong predictions on its own. Ensemble methods are used to improve the predictive capability of the model by creating several trees and integrating the predictions.

Ensemble methods are a type of machine learning technique that integrates numerous base models to create a single best-fit predictive model. Bootstrap aggregation, often known as bagging, is an ensemble algorithm for improving the robustness and performance of decision trees. In bagging, all the data is bootstrapped into various samples with replacement. The decision tree is formed with all the predictors on each of these samples, with the results averaged out to reach a final prediction. As the predictions from bagged trees are strongly correlated, Random Forest (RF) could be used to solve the problem as it just considers a subset of the predictors and de-correlates the tree. Usually, in RF, the number of predictors considered at each split (denoted as m in this paper) is equal to the square root of the total number of predictors(p). If m=p, then it is bagging. When using bagging, we look into the Out-Of-Bag (OOB) error. We can use the unused observations as a “test set” for each tree and predict them using the corresponding tree. The overall MSE (for regression problems) or the overall classification error rate (for classification problems) is the OOB error.

Boosting is a sequential ensemble strategy that improves the model by leveraging data from previously developed weaker models. This process is repeated for several rounds until a final model is created that accurately predicts the outcome. The algorithms that are commonly used for boosting are gradient boosting (GBM) and AdaBoost. In this paper, the GBM model was developed as a boosting technique. GBM learns from its mistakes and improves forecasts by using information and assessing the previous errors. As the response variable is categorical the distribution is multinomial. Three basic tuning parameters used in boosting are 1) number of trees (B); 2) interaction depth (d): number of splits to be performed on a tree, which controls the complexity of the model; and 3) shrinkage parameter or learning rate (λ). To address residual errors in the predictions from the existing tree sequence, new trees are constructed. As a result, the model might overfit the training data. Applying a weighting factor to the corrections made by new trees when they are added to the model is one way to slow down learning in the GBM model. The weighting factor added is the shrinkage parameter or learning rate(λ).

Another model developed in this paper for predicting the type of collision is Logistic Regression. Logistic regression is used to predict a dependent variable based on a set of independent variables, where the dependent variable is categorical. The categorical suggests that the outcome of an event is binary. The binary outcomes are "Yes" or "No", "True" or "False" and so on. To predict the target variable, a sigmoid curve is used where the outcome is determined by the threshold value. Binomial Logistic Regression is used to fit models that have two classes in the dependent variable, whereas multinomial Logistic Regression is used to fit models that have more than two classes in the dependent variable. Logistic regression is mathematically represented as, Text

Description automatically generatedwhere, P(X) is the dependent variable, which is to be predicted, X1, X2... Xp are the independent variables which determine the occurrence of an event and β0, β1..., βp is calculated using maximum likelihood method. While logistic regression is easier to implement and can be used for multinomial regression, the limitation of Logistic regression is that the non-linear problems can’t be solved using this algorithm.

**Methodology:**

The dataset is recorded by the Seattle Police Department and retrieved from the Seattle Open Data website. It is updated weekly on the website. The dataset used in this study was downloaded in the last week of May. Thus, traffic collision records from 2004 to the present are analyzed. For data pre-processing, the variables of the document key indexes and variables with enormous categories (for example, the address of where the collision happened) are excluded from the analysis. The remaining variables were investigated individually, then casted into appropriate object types. Some categorical variables were encoded into dummy variables as shown in Table 1. And observations with NA values were removed. Then the dataset is splitted into training and testing data sets. Sixty percent of the data were randomly sampled into a training dataset, while the rest formed the testing set.

|  |  |
| --- | --- |
| Variable | Description |
| SEVERITY | A code that corresponds to the severity of the collision:  1— Property damage  2— Injury  3— Fatality  4— Serious injury  0— Unknown |
| ADDRESS\_TYPE | Collision address type:  Alley  Block  Intersection |
| LIGHT\_CONDITIONS | The light conditions during the collision.  0 = Light,  1 = Medium,  2 = Dark |
| COLLISION\_TYPE | Collision type  Collision of vehicles at angles=1 ,  Collision with Cycles=2,  Head On Collision =3,  Collision while turning left or right =4,  Collision with Parked Car =5,  Collision with Pedestrian =6,  Rear end collision = 7,  Sideswipe collision = 8,  Unknown. |

*Table 1: Descriptions of dummy variables and variables with string categories.*

A decision tree model is developed and pruned to find the collision type. The decision tree and pruned tree models are used to find the most influential factors in collision type. Several random forests and gradient boosting models are built with different parameter values. The model with the best performance within each of these two methods is selected and later compared with models using other methods. When selecting the best model within the random forest method, OOB errors, correction rate as well as performance in reducing variance and dimensionality are considered. As for gradient boosting, both training errors and testing correction rates are considered. A logistic regression model was also fitted. Then we found the best model predicting collision type for the test dataset by the correction rate.

**Computational Results:**

1. **Decision Tree:**

Chart, line chart

Description automatically generatedAfter developing a decision tree model, we pruned the tree with optimal number of terminal nodes with the information provided by Figure 1.

Figure 1:

**Diagram

Description automatically generated**The pruned tree is shown as in Figure 2. It classifies the vehicle's collision type primarily based on the number of vehicles involved in the collision. If one or no vehicle is involved in the collision, it means that the collision only involved bicyclists, pedestrians, and at most one vehicle. This portion is further split by the number of pedestrians. If the collision involved pedestrians, then it is classified as a collision with pedestrians (encoded as 6). If no pedestrians are involved in the collision, the tree splits further down to the number of bicycles. If there are bicycles involved, then it is a cycle collision (encoded as 2). If not, the collision type is unknown. So far, the relationships between the splits and the corresponding collision types are easy to understand. But they don't provide much insightful information. The other half of the tree does.

Figure 2: Pruned Tree with Seven Terminal Nodes

The descriptions of variables can be found in Table1.

In the other half of the tree, where there are more than one vehicles involved, the tree splits down by collision address type. If the collision address type is 'Alley' or 'Intersection,' it is highly likely to result in a collision of vehicles at an angle. This could be related to drivers not paying attention when making a turn at the end of an alley or a corner of an intersection or when going straight without paying attention to turning cars. It could also be relevant to the carelessness of the cars that were trying to change a lane in the near lanes.

When the address type is 'block,' it is mostly related to Rear-end collisions and collisions with the parked car. The tree further splits by the severity of the collision. If the collision is not fatal, then it is classified as a rear-end collision (encoded as 7). This could be related to cars in the same lane having very different speeds from each other. It could also result in carelessness in the inter-car distance.

If a car accident happens in a block area and is fatal, it is likely to be involved in parked cars (encoded as 5). This is a very shocking finding. People normally think fatal car collisions are mostly related to locations at an intersection with high speed and one car hitting others at an angle. But the analysis tells a different story. That could be because when cars are moving, drivers can change the direction to avoid further damage. But when collisions involve parked cars, only the driving car has the ability to change direction, but the meanwhile roads in block areas don't provide enough room to move dramatically. The great number of parking cars along the roads in blocks further limits moving space. Another possible reason could be that drivers are more careless when driving in blocks, which results in very little response time when the accident happens.

1. **Bagging and Random Forest:**

The bagging model with the full set of predictors (m=p) and the RF models with a subset of predictors are plotted as shown in Figure 3. Three boosting models were developed with the Chart

Description automatically generatednumber of predictors considered at each split (m) equal to “p/2”, ”p/3”, and “sqrt(p).” To compare the errors for each model with the increasing number of trees, the number of trees was plotted against the OOB error. It can be observed from the plot that the OOB errors decrease with the increase in the number of trees and convergence. From the plot, it can be noted that all the RF models have approximately similar error rates, but looking through actual numbers, the RF model with half of the predictors has better performance in the context of OOB error.

Figure 3

**Chart

Description automatically generated**The correction rate for model on test dataset with split equals to p, p/2, p/3 and sqrt(p) is 65.42%,  65.43%, 65.36%, 65.45% respectively. That means the correct predicting rate for test data is the highest when the sample size of predictors is the square root of the total number of predictors. This conflicts with the OOB error values. But lower OOB errors could also indicate overfitting. Thus the correct prediction rate for the test set is weighted more than the OOB error. Also, in the context of reducing the variance and dimensionality, the RF model with sqrt(p) many predictors performs the best. Thus we concluded that the model with the sample size of predictors equals to the square root of the total number of predictors is the model with the best performance among all RF models tested.

Figure 4

1. **Gradient Boosting:**

Then gradient boosting is applied with different values of depths. And as shown in Figure 4, the gradient boosting model with an in-depth of 3 and a learning rate of 0.01 has the smallest training error among all gradient boosting models. The gradient boosting models with an in-depth of 3 and a learning rate of 0.01 also performs best in predicting the test set. It has a correction rate of 65.43% compared to 65.21% and 65.38% for in-depth 1 and 2, respectively. Thus, we conclude that gradient boosting with an in-depth of 3 and a learning rate of 0.01 perform best in all tested gradient boosting models.

1. **Logistic Regression & Final:**

The Multinomial Logistic Regression model was implemented as the collision type which was to be predicted is categorical with nine classes. On looking at the confusion matrix below, it could be observed that 11698, 2800, 14675, 3529, and 4444 values were correctly classified as class 1, 2, 5, 6, 7, respectively. Also, it can be noted that classes 3, 4, and 8 classes were not properly classified. This finding is consistent with the pruned tree where these classes were not included. It might be because there was not enough data for these classes. The correction rate for multinomial logistic regression is 65.30%.

Figure 5: Confusion matrix of the count for each categories in data and by prediction

Table

Description automatically generated

1. **Final Result**

As shown in Figure 3, the developed models were plotted against their accuracies. The models, decision tree, best amongst random forest model with ‘sqrt(p)’ number of predictors, best amongst boosting model with depth as 3 and shrinkage as 0.01 and Multinomial Logistic regression have predicted with an accuracy of  65.16%, 65.45%, 65.43%, 65.30% respectively. When comparing all the tested models, the random forest model with the square root of p **Chart

Description automatically generated**number of predictors appears to be the best model for predicting collision type for the test set.

Figure 6: Test prediction correct rate for the best model for each method.

**Summary:**

In summary, among all models investigated, the random forest model with a predictor size of sqrt(p) performs best at predicting collision type for the test set. With approaches of decision trees, vehicle count is found to be the most influential factor for traffic collision type in Seattle. It is followed by address type, severity, pedestrians count, bicycle count, and light conditions. We also see how these variables are related to different types of collisions. And that gives both drivers and authorities some ideas for preventing each kind of collision in the future.

Collisions with a car hitting with an angle usually happen in alleys and intersections. Drivers should especially pay attention to turning cars, cars in the left or right lanes, and before making a turn when driving in an alley or at an intersection to avoid a car accident.

Rear-end collisions and collisions with the parked car mostly happen in blocks. We can prevent these kinds of collisions by restricting car parkers on the block and driving more carefully within the speed limit when driving in a block. Car accidents that happen in the block with parked cars are also related to the severity of fatality. This counterintuitive finding warns drivers should be even more careful when driving in blocks and especially pay attention to driving speed compared with adjusting cars and inter-car distance.

These pieces of advice can be accordingly adopted by drivers and authorities in other States and even countries with similar city designs. With the bits of advice above, hopefully, we will have fewer car accidents and fewer lives lost or harmed by car collisions.

**Appendix:**