**Predicting the Severity of Car Accidents in Seattle**

Bingqing Lu

September 22, 2020

# **1. Introduction/Business Problem**

Nowadays, with more and more vehicles on the road and people’s increasing needs for traveling. The traffic problems are becoming one of the biggest headaches in our daily life, such as traffic jam, accidents... Especially, in several weather conditions, people who need to travel would really like a predict app that can predict/warn the possibility of car accident and its severity to improve their safety and avoid unnecessary troubles.. The predict app can use the weather / road / light information from the users to predict the accidents. With this app, drives could plan their trips ahead, and make the best decisions about the date/route... for their trips.

# **2. Data**

Data were provided by SPD and recorded by Traffic Records in Seattle from 2004 to present. The dataset has 194673 accident records and 37 attributes, providing comprehensive data to train and test the machine learning models. The dataset has a “severitycode” column, labeling the accidents’ severity, which can be used in supervised machine learning models.

## 2.1 Data preparation

There are 37 attributes in the dataset and some of them are irrelevant to our prediction models, such as ESRI unique identifier, unique key for the incident…To save computational memory and make the model clear, after carefully inspecting the metadata of the dataset, I choose 8 out of the 37 attributes as the major predictors for the probability and severity of the car accidents. They are location of the accidents, road condition, weather condition, junction type, car speeding, number of people involved, light condition and number of vehicles involved in the accidents. The “speeding” attribute has a lot of “NAN”, indicating the vehicles did not speed. I fill the “NAN” with “N”, meaning “not speeding”, and the rest records have “Y”, meaning “speeding”. Then, I remove the rows with missing data. The resultant dataset contains 58188 type 1 severity cases (property damage) and 136485 type 2 severity cases (injury), so the data needs to be balanced before training and testing. Since the dataset has abundant records. I choose the “undersampling” method to balance the dataset. I randomly remove 69638 type 1 records from the dataset, resulting in equal number of type 1 and 2 records, and they both have 56638 records.

## 2.2 Exploratory Data Analysis

To better understand the dataset and the relationships between different attributes, I conduct the analyses in the cleaned dataset. For the total number of people involved in each collision, the mean is 2.5, indicating that there are 2 to 3 people involved in one accident in average. The mean of number of vehicles involved in a collision is 1.95, which means that there are around 2 cars in one accident in average. There are 3 unique values for the collision address type: alley, block, intersection, and block is the most frequent value. There are 7 unique values for the junction type where collision took place, and most frequent one is mid-block (not related to intersection). There are 11 unique values for the weather, and most frequent one is clear. There are 9 unique values for the road condition, and most frequent one is dry. There are 9 unique values for the light condition, and most frequent one is daylight. There are 2 unique values for the speeding, and most frequent one is without-speeding. The above findings are a little bit shocking because we would assume that most accidents happened at wet, rainy, night with speeding cars.

I performed data visualization to have a deeper insight into the data, and relationships between different attributes.

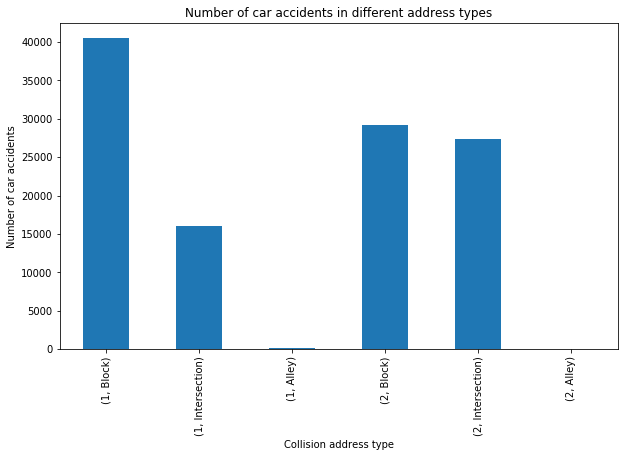


Figure 1. Bar chart for different collision address types in two severity levels 1 and 2.

This bar chart shows that in type 1 and 2 accidents, block is the place where majority accidents happened. Compared to property damage accidents (type 1), more injury accidents (type 2) happened at the intersection.

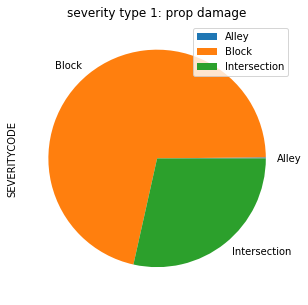


Figure 2. Pie chart for property damage accidents with different address types.

From this pie chart, I found that almost 3/4 of the property damage accidents happened at the block, and the rest are almost all intersection.

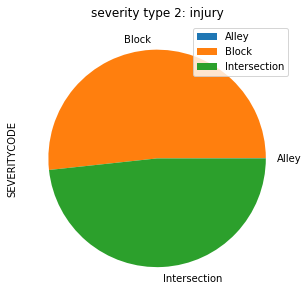


Figure 3. Pie chart for injury accidents with different address types.

The above pie chart tells us that over 50% of the injury accidents happened at the block, and the rest are almost all at intersection. Alley is in a minority for both types of accidents. More injury happened at the intersection than the block.



Figure 4. Histogram of number of people involved in the collision.

Figure 4 shows that majority accidents involve less then 10 people. In order to better view the distributions of the number of people and number of the vehicles, I computed the nature log of the (person count/vehicle count +1).

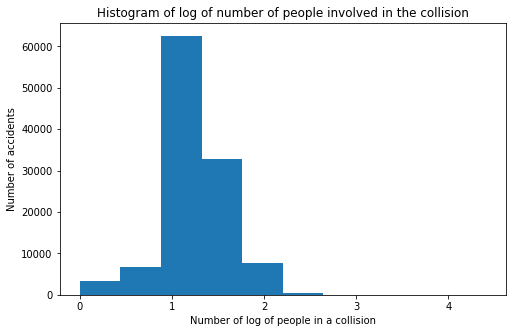


Figure 5. Histogram of log of number of people involved in the collision.

Figure 5 shows that the number of people follows the log-normal distribution. The same pattern is found for the number of vehicles involved in the accidents.

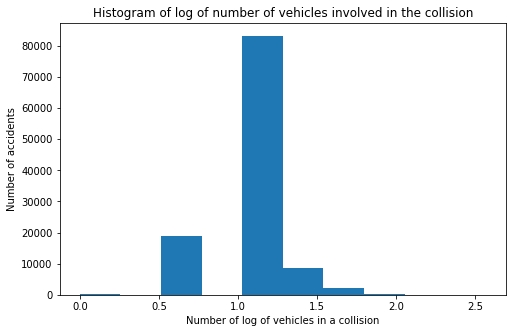


Figure 6. Histogram of log of number of vehicles involved in the collision.

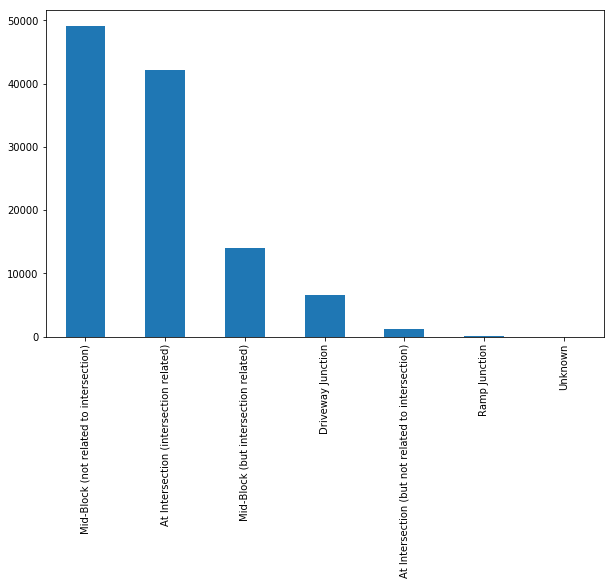


Figure 7. Bar chart for the junction types.

Most accidents happened at the mid-block (not related to intersection), which also demonstrates that block is the placed that most accidents happened.

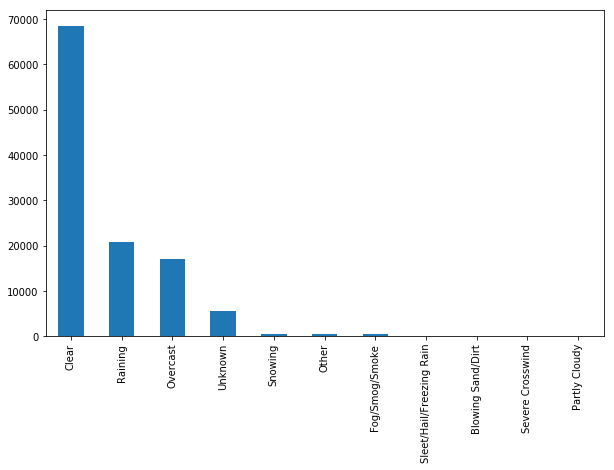


Figure 8. Bar chart for weather types.

Figure 8 shows that most accidents happened in clear weather.

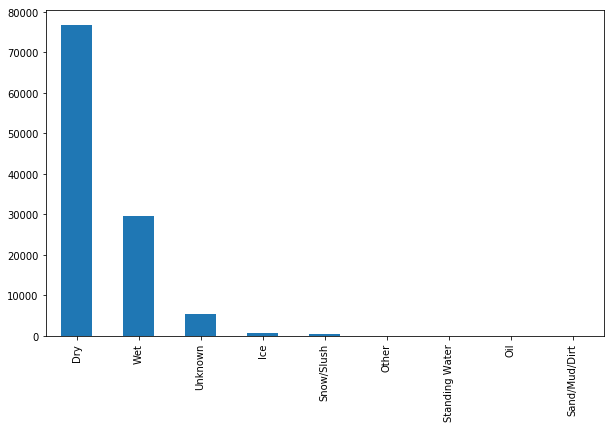


Figure 9. Bar chart for road conditions.

From figure 9, I found that most accidents happened at dry roads.

In order to investigate if the road condition will have an impact on different types of accidents. I plotted the same bar chart for the type 1 and 2 accidents.

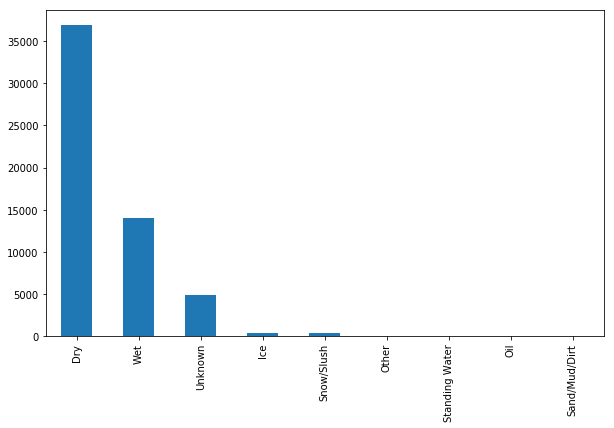


Figure 10. Bar chart for road conditions for type 1 accidents.

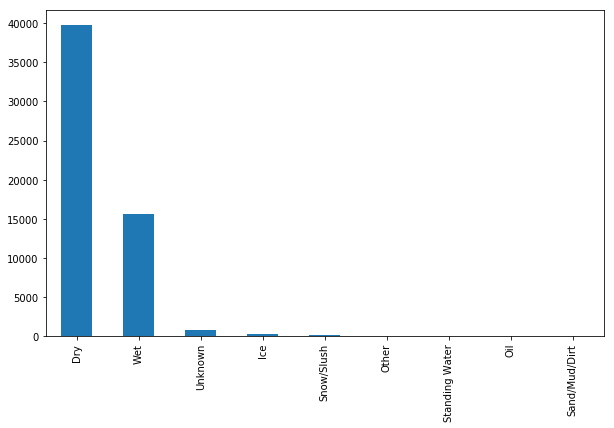


Figure 11. Bar chart for road conditions for type 2 accidents.

Comparing figure 10 and 11, I found that they are quite similar. The distribution for this attribute does not change much for different types of car accidents.

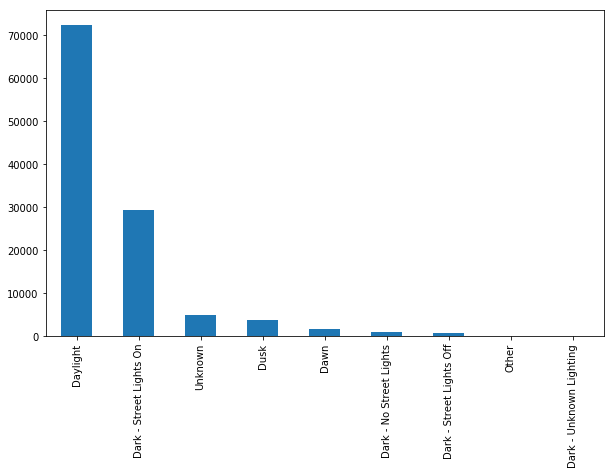


Figure 12. Bar chart for light conditions.

From figure 12, I found that most accidents happened at daylight.

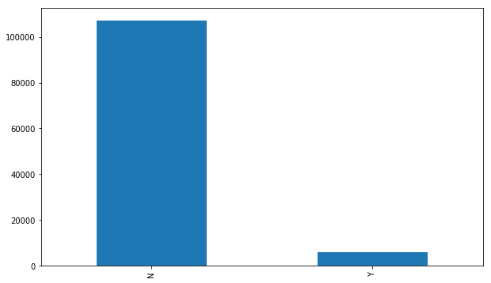


Figure 13. Bar chart for speeding conditions.

From figure 13, I found that most accidents happened without speeding.

I choose 8 attributes as predictors, and the severity level as the target to predict. There are 6 attributes are categorical data, instead of numerical data. They need to be converted to numerical data before fed into the models. After converting all data into float, I calculated the correlations between every two attributes, and plotted the heatmap.



Figure 14. Heat map for the attributes.

From figure 14, I found that address type and junction type are strongly correlated, because they both describe the place where the accidents happened. Weather and road condition are strongly correlated. Weather will largely determine the road condition. And the severity level relates most with address type and junction type, followed by the number of people positively. In general, all attributes do not correlate with each other very much, so they can be independent variables.

Finally, to eliminate the impact of the scale of different predictors. I normalized the predictors by subtracting the mean and dividing the variance of each attribute, so that all attributes have zero mean and one unit variance. The dataset is ready to be split into train and test set. I set the test size to 0.2. There are 90620 train records and 22656 test records.

# **3. Methodology**

I trained four machine learning models and evaluated and compared their performances.

## 3.1 Logistic Regression (LR) model

I first trained the logistic regression model, because it can not only predict the categorical data and also estimate the probability for each category. Logistic Regression (LR) passes the input through the logistic/sigmoid and then treats the result as a probability. I built the model using Logistic Regression from Scikit-learn package. This function implements logistic regression and can use different numerical optimizers to find parameters, including ‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’ solvers. I chose the “liblinear” solver, which is recommended for solving large-scale classification problems. C parameter indicates inverse of regularization strength which must be a positive float. Smaller values specify stronger regularization. I set C=0.01 to fit the train set, and calculated the predicted values / probabilities of the test set using the predict function.

## 3.2 Support Vector Machines (SVM) model

To compare the results of the Logistic Regression mode, I trained a second model, which is Support Vector Machines (SVM). SVM maps data into a high-dimensional feature space to categorize the data points, such as drawing a hyperplane. I used 'rbf' kernel.

## 3.3 Decision Tree (DT) model

I then trained another popular machine learning model, decision tree, to compare with the previous two models. I set the criterion as "entropy" and maximum depth as 4.

## 3.4 K-Nearest Neighbors (KNN) model

Finally, I trained the efficient K-Nearest Neighbors (KNN) model to compare the results. First, the optimal k value needs to be decided. I calculated the accuracy scores for 10 numbers form 1 to 10.

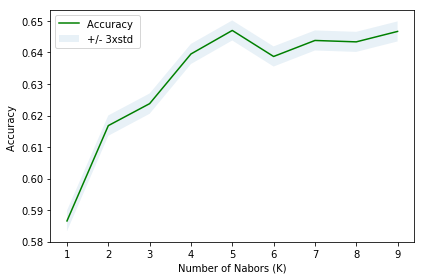


Figure 15. Relationship between KNN’s accuracy scores with different number of neighbors.

From figure 15, I found that the best accuracy was 0.65 with k= 5. And I trained and tested the model with k=5.

# **4. Results**

## 4.1 Logistic Regression (LR) model

The jaccard index is 0.64. Confusion matrix was plotted as follows:

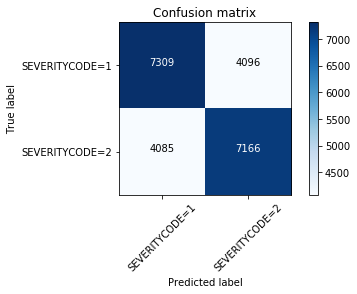


Figure 16. Confusion matrix for Logistic Regression model.

The precision, recall and f1-score are all 0.64. Log loss is 0.65.

## 4.2 Support Vector Machines (SVM) model

The precision, recall and f1-score are all 0.66. The jaccard index for SVM model is 0.66.

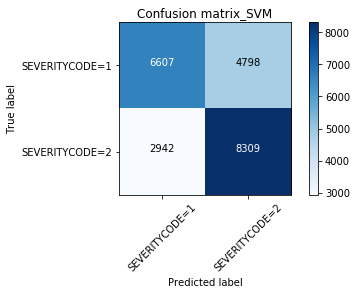


Figure 17. Confusion matrix for SVM model.

## 4.3 Decision Tree (DT) model

Decision Tree's accuracy/jaccard score is 0.67.

## 4.4 K-Nearest Neighbors (KNN) model

With optimal k value equals 5, the train set accuracy is 0.66, and test set accuracy is 0.65.

# **5. Discussion**

|  | Jaccard | Precision | Recall | F1-score | Log loss |
| --- | --- | --- | --- | --- | --- |
| LR | 0.64 | 0.64 | 0.64 | 0.64 | 0.65 |
| SVM | 0.66 | 0.66 | 0.66 | 0.66 | NAN |
| DT | 0.67 | NAN | NAN | NAN | NAN |
| KNN | 0.65 | NAN | NAN | NAN | NAN |

Table 1. Evaluation metrics for the four models. LR is Logistic Regression. SVM is Support Vector Machines. DT is Decision Tree. KNN is K-Nearest Neighbors.

From table 1, I found that DT (decision tree) has slightly better performance, followed by the SVM, then KNN and finally LR.

# **6. Conclusion**

Though the analyses of the car accident dataset in Seattle. I found that:

1. The most relevant attributes to the severity of the accidents is the address type. Block has the highest probability for car accidents, followed by the intersection and the alley is the least possible place for car accidents.
2. Four modes, namely Logistic Regression (LR), Support Vector Machines (SVM), Decision Tree (DT) and K-Nearest Neighbors (KNN), are applied to predict the severity of the accidents. Their performances are quite similar, and decision tree is slightly better than the rest of the models.