Seattle Car Crash Severity Prediction

Yohanes A Crux Gosal

4 October 2020

1 Introduction

Car crashes are one of the most common causes of death and injuries. Car accidents may be caused by several factors such as drunk drivers, poor lighting, bad weather, speeding, etc. By predicting the severity of a car accident, the local government (Seattle government in this case) may be able to take preventive actions like installing more street lights, closing potholes, build some speed bumps, etc. Therefore, the number of car accidents could be reduced and more lives could be saved.

2 Data

The data used in this project can be acquired here. This data is downloaded from Seattle GeoData website in late September 2020. This data set contains various information regarding collisions that happened in Seattle from 2004 to the present (September 2020). Where each collision data includes information about the severity, location, road condition, light condition, whether the driver speeding or not, etc. This dataset has several attributes describing each collision case, for example, ROADCOND describes the road condition (dry, wet, ice, etc.), ADDRTYPE describes the address type, and SEVERITYCODE describes the severity of the collision. The complete description of each attribute is available here.

3 Methodology

3.1 Feature Selection

For this project, I used the road condition (ROADCOND), light condition (LIGHTCOND), weather (WEATHER), and address type (ADDRTYPE) feature to predict the severity of a collision, while the SEVERITYCODE used as target values.

3.2 Data Preparation

After exploring each feature, I found several missing values. For ADDRTYPE and SEVERITY CODE column, the rows containing missing values are dropped, because there are not many missing values in these columns. For the other columns, each of them has an “Unknown” category. Hence, all the missing values in those columns are replaced with “Unknown” to avoid losing a lot of data. Next, since all of the attributes are categorical, we transformed each of them into a binary-valued feature using one-hot encoding.

The target feature, SEVERITYCODE, contains several classes. They are class 0 for unknown severity, 1 for collisions that caused property damages, 2 for collisions with injuries, 2b for collisions with serious injuries, and 3 for collisions with fatalities. Because this project’s goal is to prevent injuries or fatalities, I decided to re-group the data into two classes. The first class, class 0, is for collisions with property damage (considered as not severe collisions). The second class, class 1, is for collisions with any injuries or even fatalities (severe collisions). I ignored the data with unknown severity code since those data will not help in predicting the severity of a collision. After re-grouping the data, I found out that the data distribution is imbalanced. As can be seen in Figure 1, most of the data are in class 0 (not severe collisions).

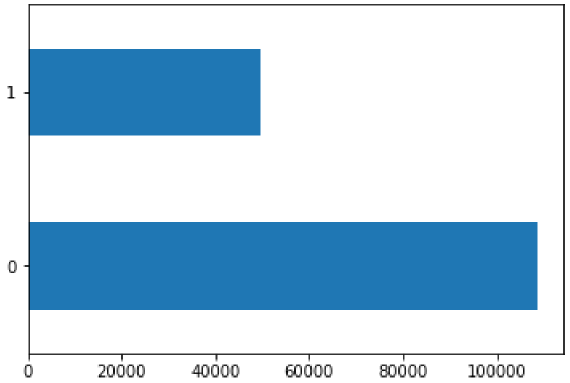

Figure 1: Number of data in each class.

This will make the model biased towards the majority class. It is less important to accurately predict collisions with low severity, but it is more important to predict collisions with high severity. So, I apply the SMOTE (Synthetic Minority Oversampling Technique) to balance the data distribution. The distribution of the data after re-sampling is given in Figure 2.

3.3 Models

Several common machine learning techniques are employed to predict the severity of the collision. In this project, we used Decision Tree, Random Forest, AdaBoost, K Nearest Neighbor, and Support Vector Machine. The dataset was split into three parts: train, validation, and test set. All models were trained using the train set and evaluated on the validation set to find the best model. Each model were evaluated using various metrics such as accuracy, precision, recall, and f1 score. The performance of each model on the validation set is given in Table 1.

Based on results in Table 1, I decided to do hyperparameter tuning on the Random Forest model to get better prediction. First, I tried to change the number of estimators (number of trees) used in Random Forest. Grid search method with 5-fold cross validation is employed to find best Random

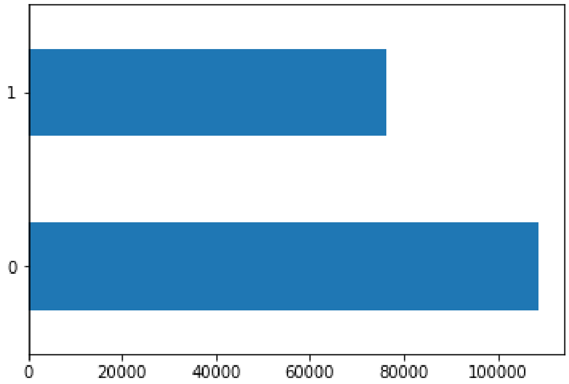

Figure 2: Number of data in each class after using SMOTE.

Table 1: Various models performance on validation set.

Model Name	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.6292	0.62	0.63	0.62
Random Forest	0.6293	0.62	0.63	0.62
AdaBoost	0.6287	0.62	0.63	0.62
K Nearest Neighbor	0.5478	0.53	0.55	0.53
SVM	0.6289	0.62	0.63	0.62

Forest model where the number of estimators ranging from 40 to 60. The ROC-AUC metric was used to evaluate performance of each model. Then, we found out that the optimal number of estimator is 55. Since 55 is in between 40 and 60, finding best model in larger range of parameter is not needed.

4 Results and Discussion

After tuning the model, it is trained on the full train set (including the validation set) and evaluated on the test set, to see how well the model works on unseen data. Random Forest model with 55 estimators is able to classify the severity of collisions with $\tilde{6}$5% accuracy. This is indeed not a very good model and still have to be improved. This model may have higher accuracy, if the data is more complete (not much missing values) or more features used as predictors.

5 Conclusion

In conclusion, more features and more valid data are needed in order to make better prediction of the severity of a crash.