Predicting "Severity" of Car Accidents in Seattle City

Bhanu Bommidi

September 10, 2020

1. Introduction

1.1 Background

Car Accidents are one of the most common hazards we all face in daily life. Such accidents often result in injury, disability, death and property damage as well as financial costs and therefore change the lives of the person involved and the those closer to them forever.

Let's imagine, if we somehow predict the possibility of somebody getting into an accident and even gauge the severity, based on the weather and road conditions. Wouldn't it be great, that we can somehow predict and warn the car commuters to drive carefully or change their travel route or maybe even avoid it, thereby minimizing the possibility of them getting involved in serious car accidents.

1.2 Problem

Data that might contribute to determining accident severity might include the weather / road / light condition, collision type, accident location like address / junction type etc.; This project aims to predict the severity of car accidents based on the weather and road conditions.

1.3 Interest

Obviously, every car commuter would be very interested in accurate prediction of accident severity, to enable them the drive carefully and if needed change their travel route or maybe even avoid it.

2. Data Acquisition and Cleaning

2.1 Data Sources

Data has been sourced from one of the example datasets here. Along with the data, the detailed description of the metadata can be found here. The example dataset consists of various Incidents in Seattle City since 2004 and consists of information pertaining to various types of collisions, Types of Address location like Alley, Block, Intersection etc.;, the Types of Collision like Angles, Parked Car etc.;, Number of persons involved etc.;.

2.2 Data Cleaning

Data has been downloaded from the source and loaded into a table. Then, I've tried to determine the "Top 5" Weather conditions to see how much an impact this has on the various Incidents being reported and surprisingly majority of the accidents did happen when the weather was "Clear", followed by when it is "Raining" and "Overcast".

Then I've tried to cross-check on how balanced the data is in terms of various "Severity". What I've observed is that the data only consists of only Severity 1 & 2, of which the Severity 1 incidents are more than twice of the Severity 2 and if not fixed will result in a biased prediction model. To fix this problem & create a more unbiased model, I've tried to randomly remove "Severity 1" incidents to make it inline with those of "Severity 2".

After balancing the data, I deduced the Year, Month, Day, Hour & Day of Week from the "Incident Date / Time" field after normalizing the label to a standard date-time field. Then I've tried to obtain the count of number of incidents by Year to cross-check if there are any outliers. Except for 2020, for which the data only exists till mid of May, the number of Incidents are pretty much linear since 2010 except for a slight increase in 2015.

I observed that there is some incorrect encoding for the "UNDERINFL" i.e.; 0, 1, N & Y and therefore performed the translation to convert N, Y to 0, 1 respectively. Also, for the "ST_COLCODE" we've observed a few missing values for which we've tried to replace them with the best suitable value based on the metadata sheet i.e.; to 31, which means "Not Stated".

Next, we've tried to figure out the list of features for which majority of the values are missing or not captured. Technically speaking, such features does not really contribute to the prediction. Looking at the ratio of missing data, I've set a threshold to ignore all columns for which more than 50% of the rows have missing data for e.g. EXCEPTRSNCODE, INATTENTIONIND, SPEEDING etc.;

2.3 Feature Selection

Along with the ignoring features for which more than 50% of the rows have missing data, we've also ignored some irrelevant fields like Keys, Location, Description etc.;. From the subset of the various features, we've tried to translate the "Categorical" data to "Numeric" codes. In addition, according to the "Metadata" information there could be a possibility of a alpha-numeric "Severity Code" like "2b" i.e.; "Serious Injury" and to anticipate such data, we've tried to convert to 2.5.

After translating the "Categorical" features to "Numeric" codes and converting the adjusting the features for some Columns, we've tried to come-up with a sort of "Correlation Matrix" between the various features and the "Severity" Code. Using this "Correlation Matrix", we've tried to identify the various features for which the correlation coefficient is between outside the range of -0.05 and +0.05.

We've observed that the "Weather" condition and "Address Type" have a good correlation with the "Severity" Code / Description (see charts below).

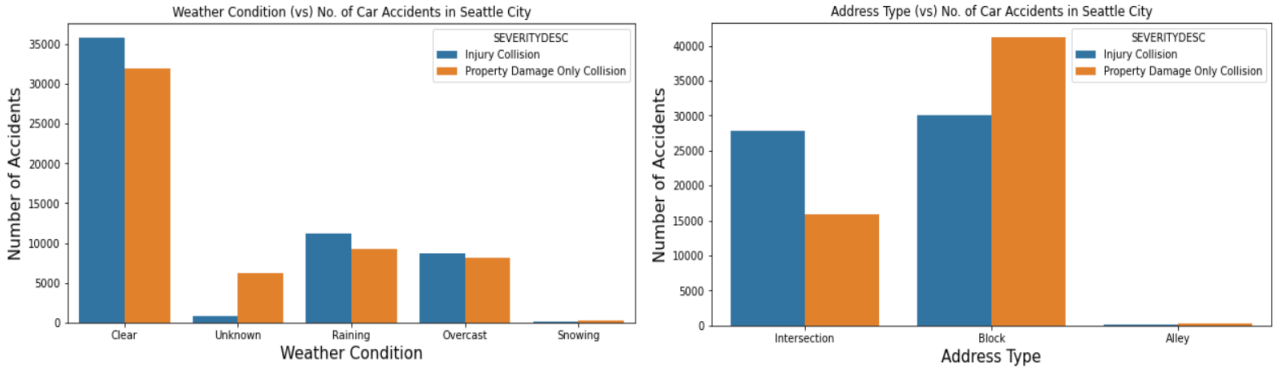

Figure (1) & (2) above represents the positive correlation between the "Weather Condition" & "Address Type" respectively with the "Severity" of the Car Accident.

Below, you will also find the correlation between the various features (correlation coefficient is between outside the range of -0.05 and +0.05) and "Severity" Code.

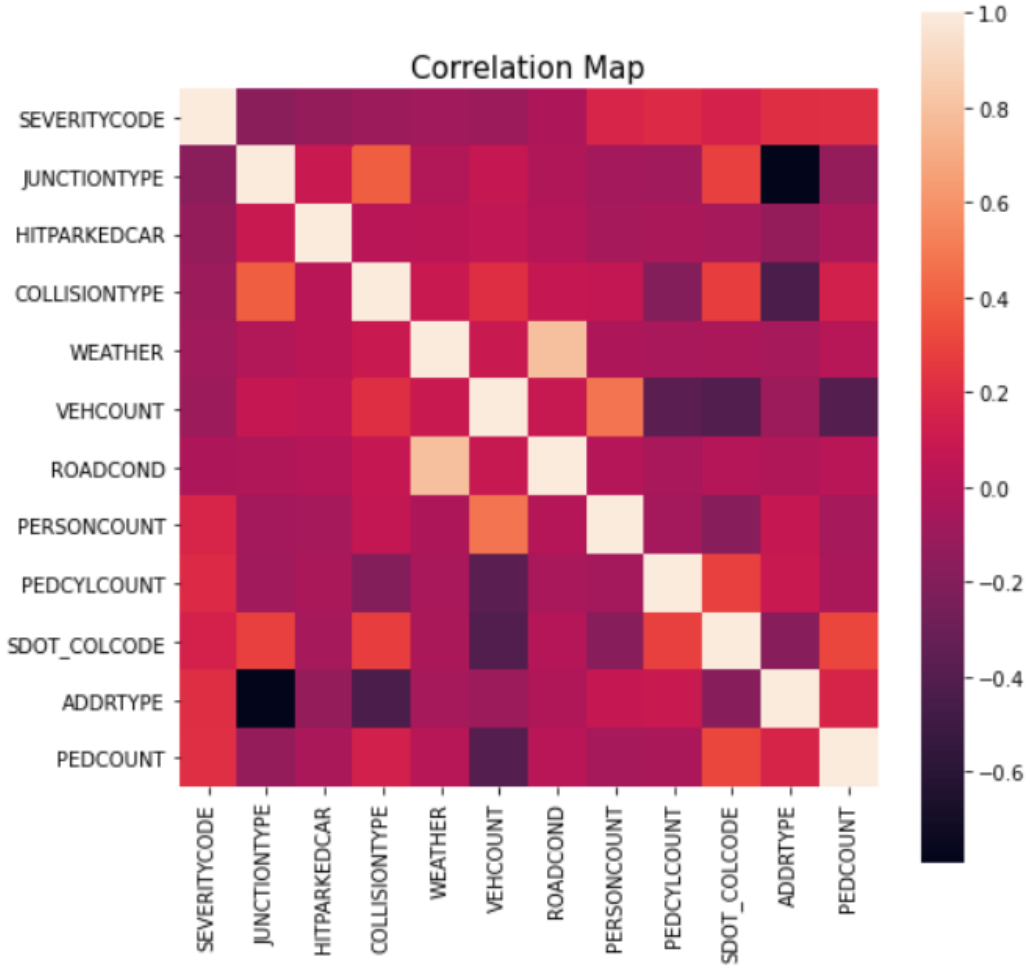

Figure (3) above represents the correlation between various features with the "Severity" of the Car Accident.

Table 1: Feature Selection after Data Cleaning

Features Retained	Features Dropped	Reason for dropping Features
ADDRTYPE, COLLISIONTYPE, PERSONCOUNT, PEDCYLCOUNT, VEHCOUNT, JUNCTIONTYPE, SDOT_COLCODE, WEATHER, ROADCOND, ST_COLCODE, HITPARKEDCAR	OBJECTID, INCKEY, COLDETKEY, REPORTNO, INTKEY, SDOTCOLNUM, SEGLANEKEY, CROSSWALKKEY	ID, Key or Number fields probably identifying specific fields.
	LOCATION, EXCEPTRSNDESC, SEVERITYDESC, SDOT_COLDESC, ST_COLDESC	Description or Text fields.
	X, Y, STATUS, EXCEPTRSNCODE, PEDCOUNT, INCDATE, INCDTTM, INATTENTIONIND, UNDERINFL, LIGHTCOND, PEDROWNOTGRNT, SPEEDING	Correlation with Severity is extremely low.

3. Predictive Modelling

Firstly, I've tried to split the feature dataset into "Training" & "Testing" datasets with a 70-30 split respectively. Then, I've used to "Training" set to train the various models and then used the "Testing" set to test the corresponding models to determine the accuracy of the models.

I've tried to use various "Classification" models like K Nearest Neighbor (KNN), Decision Tree, Logistic Regression, Random Forest & Gradient Boost and tried to measure their performance based on various metrics like Logarithmic Loss, Jaccard Similarity Score, F1-Score etc.;.

Of the various models, the "Gradient Boost" & "Decision Tree" performed the best followed by "Random Forest" & "KNN" and lastly "Logistic Regression" performed slightly less. Either way, the overall difference between the various models are comparatively small. Below you will find the various comparison metrics, with the best metrics highlighted in Green.

Table 2 : Performance of various "Classification" models, with best Performance highlighted in Green.

	KNN	Decision Tree	Logistic Regression	Random Forest	Gradient Boost
Logarithmic Loss	0.915	0.578	0.594	0.626	**0.535**
Jaccard Similarity	0.516	**0.524**	0.518	0.516	0.517
F1-score	0.693	**0.711**	0.665	0.707	0.709
No. of True Positives	12782	13851	10699	13935	**13999**
No. of False Positives	5885	6259	**4744**	6467	6478
No. of False Negatives	4833	3764	6916	3680	**3616**
No. of True Negatives	11413	11039	**12554**	10831	10820

Below you will also find the ROC Curve for the various Classification models. For this particular case, higher True Positives is more important than the higher False Positives, as it is important to have better probability of detection for the Accident "Severity" . As per the ROC Curves, the "Gradient Boost" model also has the highest "False Positive Rate" compared to the remaining models.

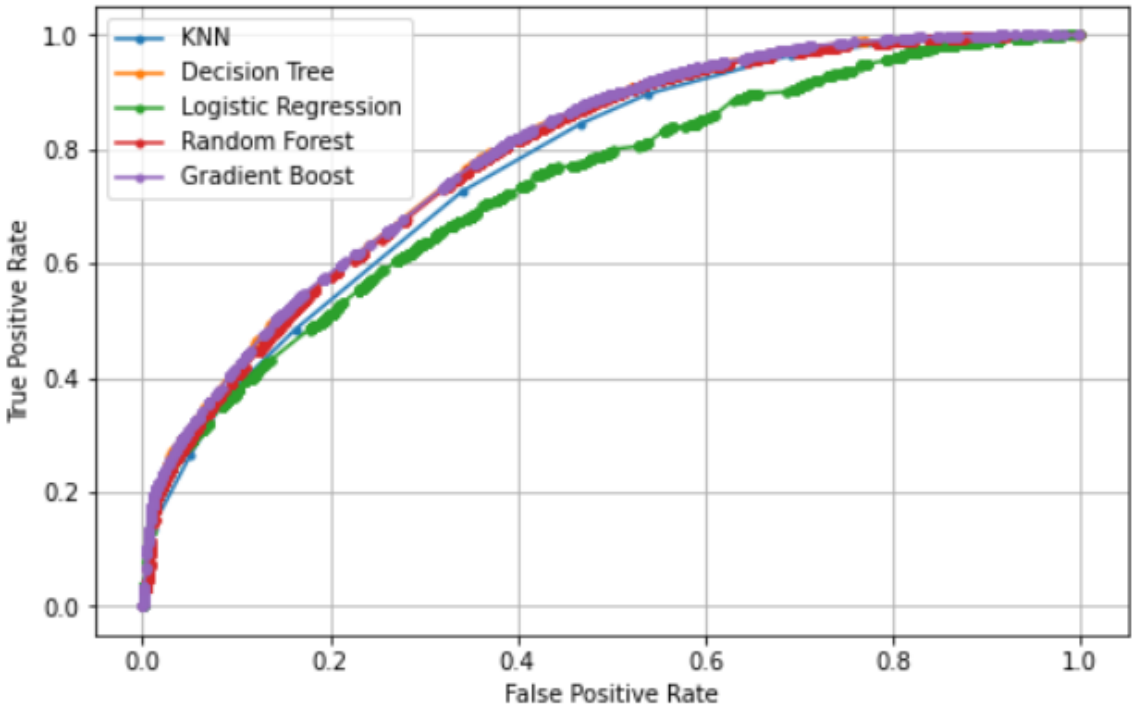

Figure (4) above shows the ROC Curves of different Classification models.

4. Conclusion

As part of this exercise, I have tried to analyze the relationship between various Car Accidents in the Seattle City for the past 15-16 years and tried to predict the Severity of an Accident based on various features like Weather, Road Condition, Address Type, Collision Type etc.;.

I have built various classification models to predict the severity of car accidents. These models can be extremely useful in keeping the various car commuters informed on the severity of accidents and advise them to either drive carefully and if needed change their travel route or maybe even avoid it.