# Machine Learning for Traffic Crash Prediction

**Lisa Li**

9/20/2020

1. **Introduction**

Seattle is the largest city in the state of Washington, as well as the largest in the Pacific Northwest. The current metro area population of Seattle in 2020 is **3,433,000**, a 0.79% increase from 2019 [1]. Seattle residents get around mostly by car; and the car crash happened in Washington every 4.5 minutes, Seattle also is the 8th most dangerous city for car accidents in USA.

For this project, we want to use this car collision data provided by SPD of Seattle, to build a machine learning model; to predict the severity of an accident using given drive condition like weather, road condition, and light condition, to better understand the relationship between these factors and traffic crash outcomes.

1. **Data**

This collisions data was been provided by Seattle Police Department and weekly recorded by Traffic Records from 2004 to present. In this dataset, there are total 194673 observations with 38 attributes. Using DataFrame dtypes property, we noticed that main types stored in this dataset are object, which are the categorical data.

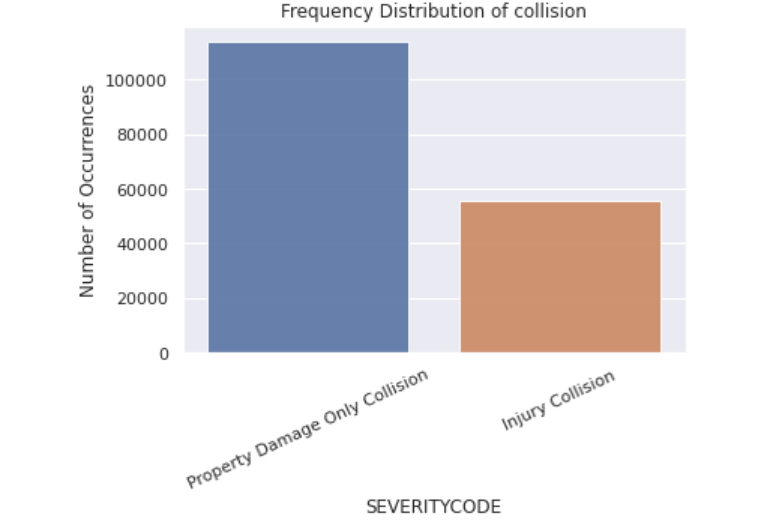
Attribute SEVERITYCOD and SEVERITYDESC describe the fatality of an accident, the remaining attributes are address type, location, collision type, weather, road condition, light condition, speeding, whether collision was due to inattention, whether or not a driver involved was under the influence of drugs or alcohol etc.

There are significant data missed or unknow on this dataset. For example, in total 194673 observations, INATTENTIONIND attribute has 164868 rows of null value; and SPEEDING has 185340 rows of null value. Data cleaning or data wrangling were performed, including drop missing data, binning of the data, to improve the predictability when built the machine model.

1. **Methodology**

Duo to the nature of this dataset being categorical, the classification machine learning algorithm Logistic regression was been used in this analysis. With this algorithm, it can not only predict the class of each car collision case, but also measure the probability of a car collision case belonging to a specific class.

SEVERITYCOD and SEVERITYDESC attributes describe the fatality of an accident. There are only two types of car accident severity code, it is a binary variable. It was been used for the target or dependent variable. With the [numpy.where](http://docs.scipy.org/doc/numpy-1.10.1/reference/generated/numpy.where.html) function, the value of SEVERITYCOD was been converted to 0 for “Injury Collision” and 1 for “Property Damage Only Collision”.

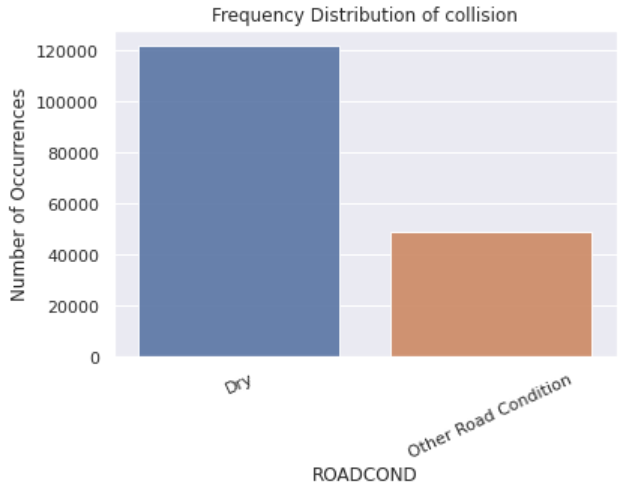
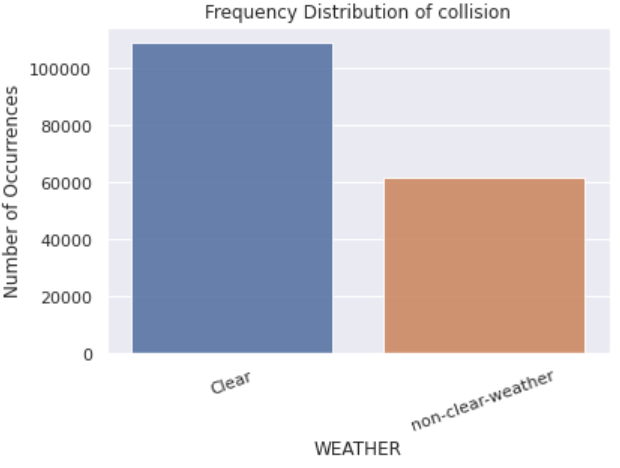


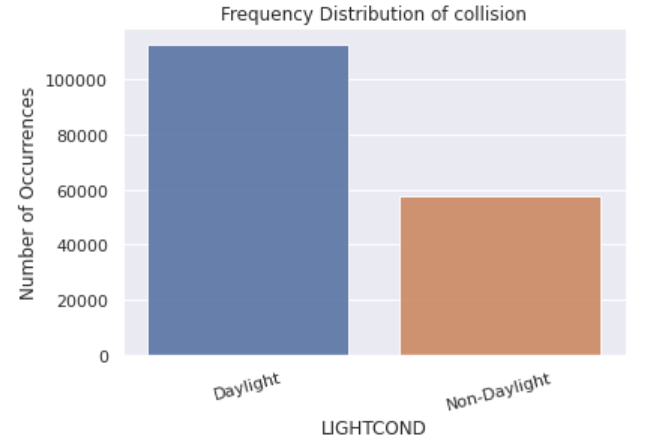
1 - Property Damage Only Collision - 67.2%

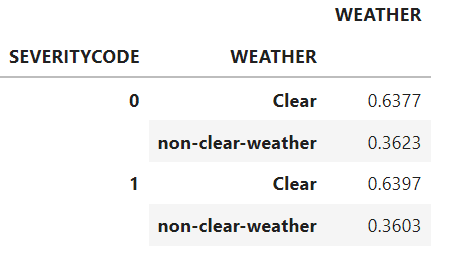
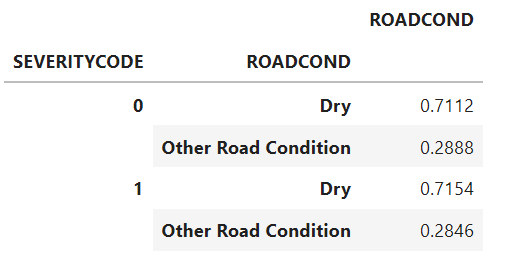
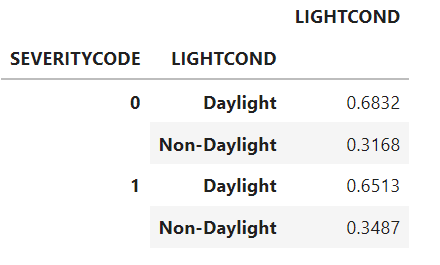
0 - Injury Collision – 32.8%

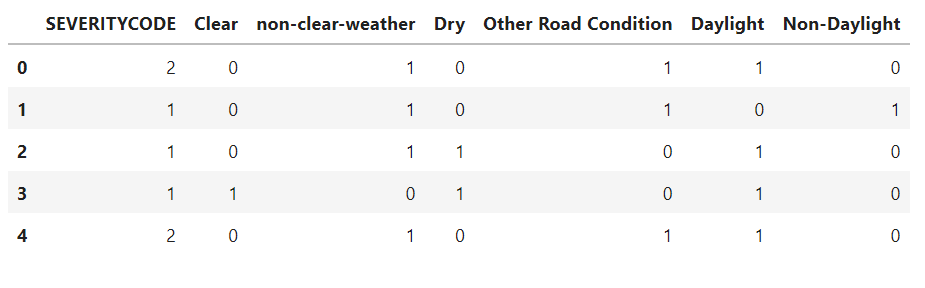
Drive condition attributes WEATHER, ROADCOND, LIGHTCOND are been selected as the independent variables for the modeling. After cleaning the null and unknow value from these columns, they were been categorized and data distribution are shown as following groups:

|  |  |
| --- | --- |
| WEATHER | 1) Clear  2) non-clear-weather: Snowing,Raining,Overcast,Fog/Smog/Smoke,Sleet/Hail/Freezing Rain,Blowing Sand/Dirt,Severe Crosswind, Partly Cloudy, Other |
| ROADCOND | 1) Dry  2) Other Road Condition:  Snow/Slush, Wet, Other,Standing Water,Ice, Oil, Sand/Mud/Dirt |
| LIGHTCOND | 1) Light  2) Non-Daylight:  Dark - Street Lights On, Dusk, Dawn, Dark - No Street Lights, Dark - Street Lights Off, Other, Dark - Unknown Lighting |



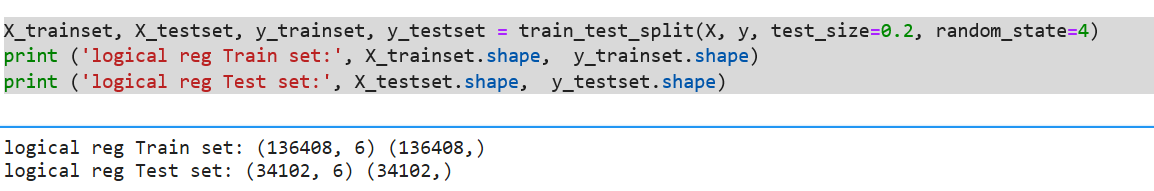


Since independent variables are all categorical, they were been dummy or indicator coded as following. 

As you can see at this point, every field in this data set is now numeric, it is ready for a logistic regression machine learning algorithm.

Using built-in functionality from scikit-learn, the dataset was been split into train and test set, fit the model with train set, and then predict using the test set. The test size consisted of 20% of the original data set, which has 34,102 samples, while reminder is used for train the model, which has 136,408 samples.

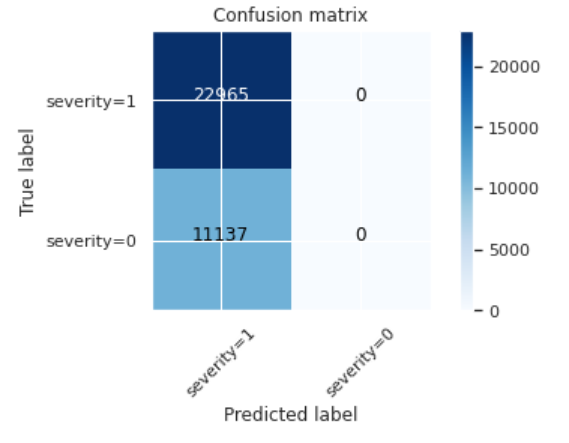


During built the model development, Jupyter Notebooks was been used to conduct this analysis; and some basic Python libraries Pandas, Numpy, Matplotlib, and Seaborn, Scikit-learn were also been imported.

1. **Result**

In the Logistic Regression model, the accuracy evaluation was been done using jaccard index, the result showed the test dataset and train dataset has 67.3% alike.

Another method **confusion matrix** was been used to looking at accuracy of a classifier.

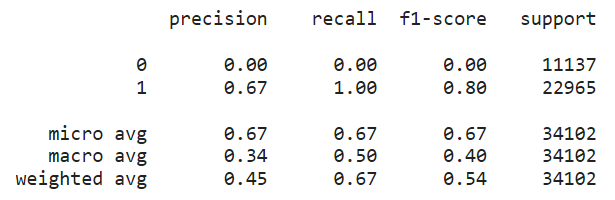


There are total 34102 samples used for this testset, 22965 are Property Damage Only Collision and 11137 are Injury Collision.

Look at first row. The first row is for collision class 1 whose actual value in test set is 1. Out of 34102 collision, the value of 22965 of them is 1. And out of these 22965, the classifier correctly predicted 22965 of them as 1, and none of them as 0. It means, for 22965 collision the actual value was 1 in test set, and classifier also correctly predicted those as class 1, the error of the model is 0.

From the second row with collision class 0, there were 11137 collision cases which value was 0. The classifier didn’t correctly predict any of them as 0, and 11137 of them wrongly as 1, the type II – false-negative error is extremely high. And it didn’t do a good job in predicting the collision with value class 0.

A Classification report is used to measure the quality of predictions from this Logistic Regression algorithm. The report showed that accuracy of positive predictions for Property Damage Only Collision is 67% but for Injury Collision is 0; weighted avg precision is 45%, misclassification is 55%; and weight avg f1-score is that 54% of positive predictions were correct; Recall showed the 67% of the positive cases this model catched.



Finally, the score **of the log loss** evaluation is 0.63, which measures the performance of a classifier where the predicted accuracy percentage is 37%.

1. **Discussion**

A good candidate of dataset using Logistic Regression model is when the target field is categorical or specifically is binary; when you need the probability of your prediction and you need to understand the impact of a feature. Seattle’s car accident dataset seems meet these criteria.

Overall, Logistic Regression model using drive condition as predictor variables in this dataset is not adequate to predict likelihood of a car accident on the severity, especially for Injury Collision class.

We possible can try to use other features in this dataset, for example, the date and time of the incident, geographic location using Logistic Regression model to better predict the likelihood of car accidents.

Also looking more details, Speeding and INATTENTIONIND (Whether or not collision was due to inattention) missed over 85% of data and these two features are expected to be the top causes of car collision. If these data can be carefully collected, it may help to use machine learning Logistic Regression model to predict the severity of an accident using these given features,

1. **Conclusion**

In this analysis, we have completed the data wrangling, exploratory data analysis; built the Logistic Regression model from Scikit-learn package with numerical optimizers saga solvers and **C** parameter sets to 0.01; predicted the collision accuracy percentage is 37%, which indicted the drive condition failed to predict the likelihood of collision severity. This method may use to predict the “Property Damage Only Collision” in the future.

Only weather, road condition and light condition features have been selected for this study. Some were excluded from this analysis due to missing significant amount of data, but it does not mean that they have no influent on the severity of a car crash. For example, speeding and inattention attributes can be selected for car crash prediction and prevention studies in further analysis.

**6. References**

[1]

<https://www.macrotrends.net/cities/23140/seattle/population#:~:text=The%20current%20metro%20area%20population%20of%20Seattle%20in,was%203%2C379%2C000%20%2C%20a%201.2%25%20increase%20from%202017>

[2] <https://www.currentresults.com/Weather-Extremes/US/cloudiest-cities.php>

[3] <https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e>

[4] <https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/>