Predicting the Occurrence of High Severity Level Road Accident

Gary Poon

September 25, 2020

1. Introduction and Background

Traffic congestion is a big problem for all countries, especially in big cities. The problem become serious because of high volume of vehicles and the development of population. Traffic congestion always happens when there is a road accident. The higher severity level the accident is, the longer time the congestion lasted for. It normally takes hours to clean the road and release the traffic. Drivers need to wait, and it is not easy to change their travel path when they are already suffering from it. The most effective solution is aware the road condition, route to other path before you were suffering from a traffic jam. In this project, the collision data from Seattle will be used to study and predict the happens of a high severity level road accident. It provides a warning to driver to take actions before suffering from a traffic congestion.

1.1. Problem

One of the major causes of traffic congestion is a road accident. The higher severity level the accident is, the longer time the traffic jam is. To avoid suffering, this project would like to predict the occurrence or even the probability of a high severity level road collision.

1.2. Interest

Drivers would be interested to the likelihood of the occurrence of a high severity level road collision. It helps them to arrange a better plan before they start their journey.

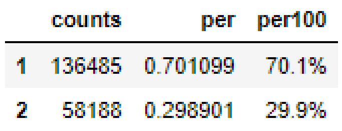
2. Data acquisition and cleaning

2.1. Data sources

Coursera shares the data which are downloaded from Seattle government's open data platform. The data includes all types of collisions from 2004 to 2020-May. There are total 195K collision records with 37 variables. The data contains many useful information including the severity level, the accident datetime, collision type, weather condition, road condition, the number of vehicles involved, the number of pedestrians involved, etc. To predict the occurrence of a high severity level accident. The variable "SEVERITYCODE" is selected as our target variable. There are two available values "1 – prop damage" and "2 – injury". In this project will define "2 – injury" as a high-level severity road accident and predict its occurrence.

2.2. Data cleaning

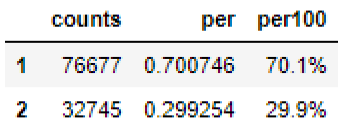
Firstly, the distribution of the target variable "SEVERITYCODE" is studied. The "2 – injury" only occupies 30% of the total records which shows it is an unbalanced data. Data balancing may need to be performed before building a predictive model.



	counts	per	per100
1	136485	0.701099	70.1%
2	58188	0.298901	29.9%

Secondly, the distributions of the rest variables are observed. There are some variables with >=80% records contain NaN or missing value. Some of them are indicator variables and the rest of them are not. We can identify them based on the distribution of the available values and their business meaning. For the indicator variables, we will keep the inductor with its original value "Y" and impute the NaN values to "N". For the rest of NaN variables, we will drop them from our dataset because they are not informative and should not be selected as the predictors in our model.

Finally, after dropping the columns with lots of NaN values, we check the dataset and remove the records will NaN in the remaining variables. There are 109K clean records and 35 variables are left. The distribution of the target variables "SERVERITYCODE" is checked again. The "2 – injury" still occupies 30% of the total clean records which indicates the clean dataset is still an unbalanced dataset. Therefore, we need to perform data balance before fitting a model.



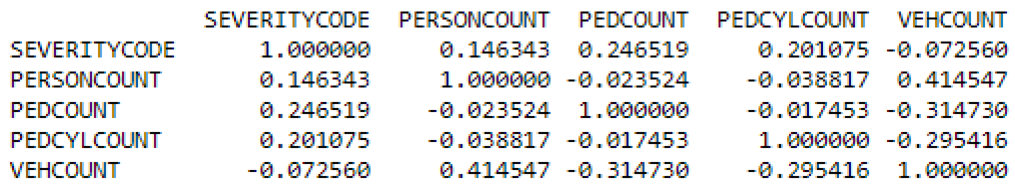
	counts	per	per100
1	76677	0.700746	70.1%
2	32745	0.299254	29.9%

3. Exploratory Data Analysis

The cleaned dataset contains 35 variables, only 4 variables are numeric and the rest of them are categorical. For the numeric variables, we will calculate the correlation matrix and select the variables which are highly correlated to the target variable and independent among each other to avoid the collinearity problem. For the categorical variables, we will perform the chi-square test between each categorical variable and the target variable and select the ones which are significant. The selected numeric and categorical variables will be the predictor for model fitting.

3.1. Numeric Variables – Correlation Matrix

As mentioned above, a correlation matrix will be created to study the relationship among each numeric variables and target variable.



```
              SEVERITYCODE  PERSONCOUNT  PEDCOUNT  PEDCYLCOUNT  VEHCOUNT
SEVERITYCODE      1.000000     0.146343  0.246519     0.201075 -0.072560
PERSONCOUNT       0.146343     1.000000 -0.023524    -0.038817  0.414547
PEDCOUNT          0.246519    -0.023524  1.000000    -0.017453 -0.314730
PEDCYLCOUNT       0.201075    -0.038817 -0.017453     1.000000 -0.295416
VEHCOUNT         -0.072560     0.414547 -0.314730    -0.295416  1.000000
```

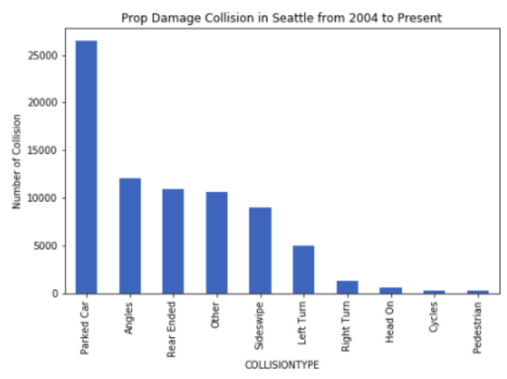
We can find that “PERSONCOUNT”, “PEDCOUNT” and “PEDCYLCOUNT” have positive relationship with the “SEVERITYCODE”. Only “VEHCOUNT” has negative relationship.

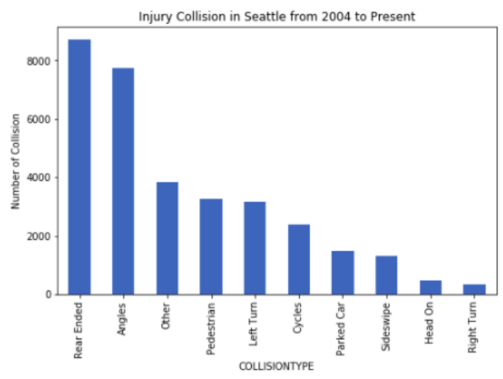
By comparing the magnitude, “PEDCOUNT” and “PEDCYLCOUNT” have a larger correlation with the “SEVERITYCODE”, but they are also correlation to each other. To avoid the collinearity problem, we should keep the predictors to be independent to each other and prevent select highly correlated variables in the same model. In this case, we would only select “PEDCOUNT” which has the highest correlation value as the predictor.

3.2. Categorical Variables – Chi-Square Test

To find out significant categorical variables, we will perform chi-square test of independence between each categorical variable and the target variable. Since there are too many categorical variables in the dataset, we would only cover the details of the significant ones in the report.

<u>Distribution Plot and Chi-Square Test of "COLLISIONTYPE"</u>





```
SEVERITYCODE        1     2
COLLISIONTYPE
Angles          12045  7739
Cycles            345  2393
Head On           631   485
Left Turn        4958  3152
Other           10679  3843
Parked Car      26479  1490
Pedestrian        308  3276
Rear Ended      10909  8722
Right Turn       1350   334
Sideswipe        8973  1311
```

```
significance=0.050, p=0.000
Dependent (reject H0)
```

From the distribution,

- For "1 – prop damage", the highest collision type is "Parked Car"
- For "2 – injury", the highest collision type is "Rear Ended"

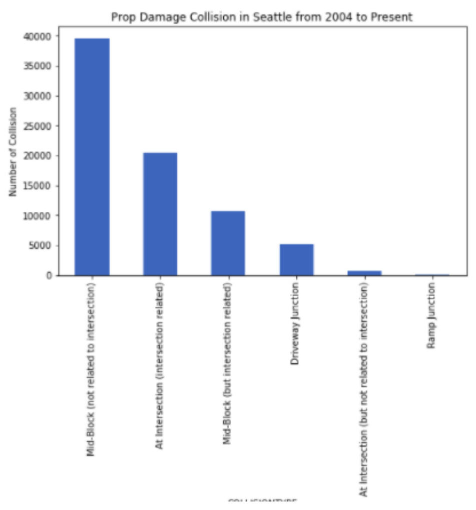
The differences of the distribution imply that "COLLISIONTYPE" may be able to classify a high severity level accident. Then, a Chi-Square test is performed to proof our hypothesis.

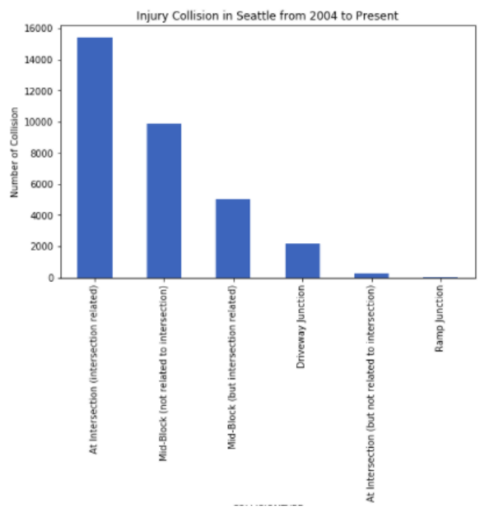
Chi-Square Test:

- H0: There is no relationship between "COLLISIONTYPE" and "SEVERITYCODE".
- H1: H0 is not true.

Chi-Square test of independence is performed. The result p-value is 0.00 which are less 0.05 significant level. We can reject the H0 and conclude that there is a relationship between "COLLISIONTYPE" and "SEVERITYCODE". "COLLISIONTYPE" will be selected as one of our predictors for model fitting.

<u>Distribution Plot and Chi-Square Test of “JUNCTIONTYPE”</u>





```
SEVERITYCODE                                          1      2
JUNCTIONTYPE
At Intersection (but not related to intersection)   717    268
At Intersection (intersection related)            20482  15418
Driveway Junction                                  5121   2170
Mid-Block (but intersection related)              10689   5036
Mid-Block (not related to intersection)           39638   9847
Ramp Junction                                        30      6
```

```
significance=0.050, p=0.000
Dependent (reject H0)
```

From the distribution, the first and the second-high occurrence “JUNCTIONTYPE” of two “SEVERSITYCODE” are switched.

- For “1 – prop damage”, the first-highest junction type is “Mid-Block (not related to intersection)” and the second-high junction type is “At Intersection (intersection related)”.
- For “2 – injury”, the first-highest junction type is “At Intersection (intersection related)” and the second-high junction type is “Mid-Block (not related to intersection)”.

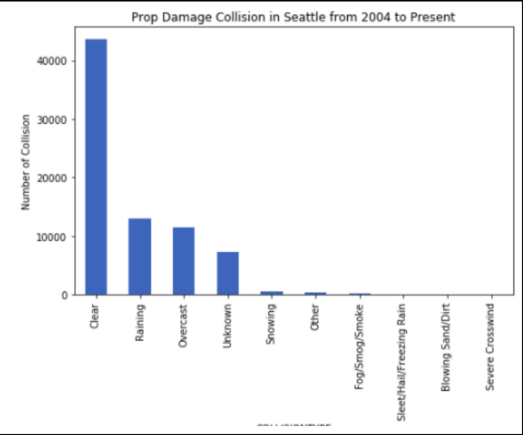
The differences of the distribution imply that “JUNCTIONTYPE” may also be able to classify a high severity level accident. Then, we can carry out a Chi-Square test to proof it.

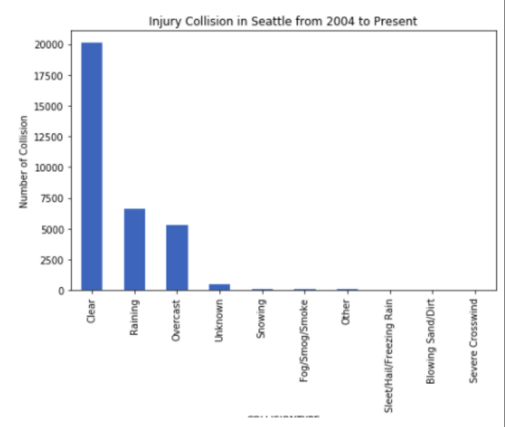
Chi-Square Test:

- H0: There is no relationship between “JUNCTIONTYPE” and “SEVERITYCODE”.
- H1: H0 is not true.

Chi-Square test of independence is performed. The result p-value is 0.00 which are less 0.05 significant level. We can reject the H0 and conclude that there is a relationship between “SEVERITYCODE” and “SEVERITYCODE”.
“JUNCTIONTYPE” will be selected as one of our predictors for model fitting.

Distribution Plot and Chi-Square Test of “WEATHER”





```
SEVERITYCODE                  1      2
WEATHER
Blowing Sand/Dirt            17      7
Clear                     43589  20092
Fog/Smog/Smoke              196     87
Other                       447     65
Overcast                  11530   5271
Raining                   13022   6583
Severe Crosswind              8      3
Sleet/Hail/Freezing Rain     67     21
Snowing                     488    114
Unknown                    7313    502
```

```
significance=0.050, p=0.000
Dependent (reject H0)
```

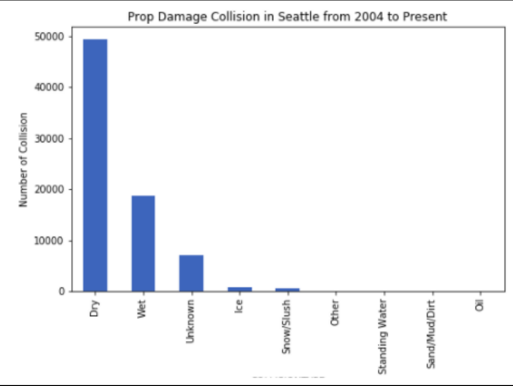
The distribution of “WEATHER” between two “SEVERSITYCODE” are similar. It is difficult to determine whether there is a relationship between “WEATHER” and “SEVERSITYCODE”. For this case, we directly carry out a Chi-Square test to test the independence.

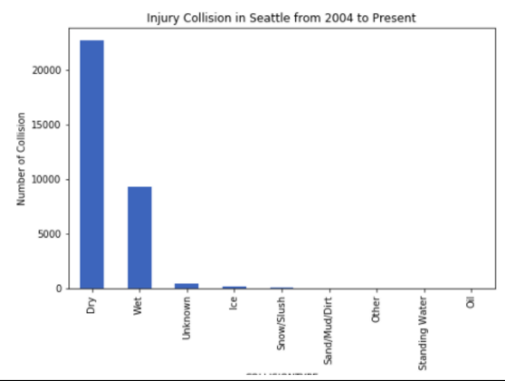
Chi-Square Test:

- H0: There is no relationship between “WEATHER” and “SEVERITYCODE”.
- H1: H0 is not true.

Chi-Square test of independence is performed. The result p-value is 0.00 which are less 0.05 significant level. We can reject the H0 and conclude that there is a relationship between “WEATHER” and “SEVERITYCODE”.
“WEATHER” will be selected as predictor for model fitting.

<u>Distribution Plot and Chi-Square Test of “ROADCOND”</u>





```
SEVERITYCODE         1      2
ROADCOND
Dry              49411  22677
Ice                713    187
Oil                 17     15
Other               50     21
Sand/Mud/Dirt       25     21
Snow/Slush         588    111
Standing Water      48     19
Unknown           7114    409
Wet              18711   9285
```

```
significance=0.050, p=0.000
Dependent (reject H0)
```

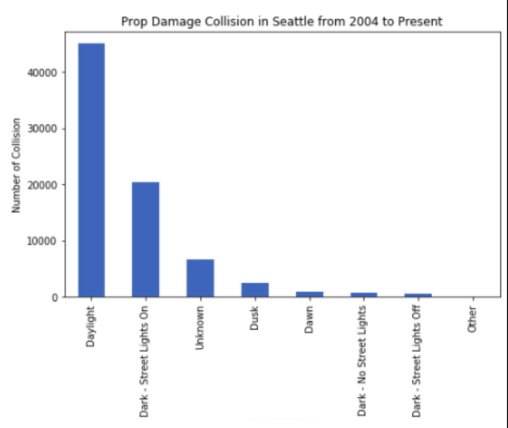
The distribution of “ROADCOND” between two “SEVERSITYCODE” are similar. It is not clear to determine whether there is a significant relationship between “ROADCOND” and “SEVERSITYCODE”. We would also carry out a Chi-Square test to test the independence.

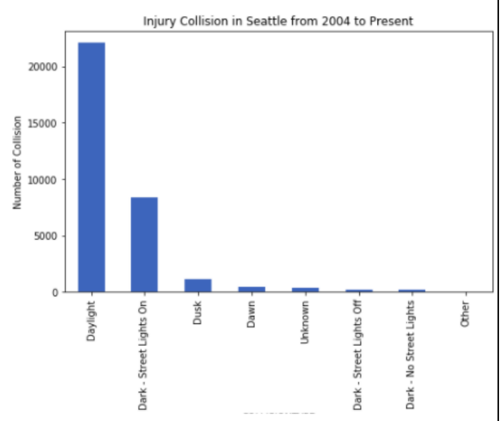
Chi-Square Test:

- H0: There is no relationship between “ROADCOND” and “SEVERITYCODE”.
- H1: H0 is not true.

Chi-Square test of independence is performed. The result p-value is 0.00 which are less 0.05 significant level. We can reject the H0 and conclude that there is a relationship between “ROADCOND” and “SEVERITYCODE”.
“ROADCOND” will be selected as predictor for model fitting.

Distribution Plot and Chi-Square Test of “LIGHTCOND”





```
SEVERITYCODE                  1      2
LIGHTCOND
Dark - No Street Lights     686    167
Dark - Street Lights Off    539    177
Dark - Street Lights On   20400   8398
Dawn                        920    431
Daylight                  44953  22052
Dusk                       2432   1135
Other                        79     17
Unknown                    6668    368
```

```
significance=0.050, p=0.000
Dependent (reject H0)
```

The distribution of “LIGHTCOND” between two “SEVERSITYCODE” are similar. Both occurrences are occupied by “Daylight” and “Dark – Street Lights On”. However, the distribution of the rest of light conditions are slightly different. We perform a Chi-Square test to test the independence.

Chi-Square Test:

- H0: There is no relationship between “LIGHTCOND” and “SEVERITYCODE”.
- H1: H0 is not true.

Chi-Square test of independence is performed. The result p-value is 0.00 which are less 0.05 significant level. We can reject the H0 and conclude that there is a relationship between “LIGHTCOND” and “SEVERITYCODE”.
“LIGHTCOND” will be selected as predictor for model fitting.

3.3. Convert target variable into flag

For simplicity, the target variable is converted into a flag. There are two available values in “SEVERITYCODE”, “1 – prop damage” and “2 – injury”. In this project, we target to predict the high severity level accident. Therefore, a new flag variable “SEVERITY_TARGET” is created, if the “SEVERITYCODE” equals to “2 – injury”, the “SEVERITY_TARGET” will be set as 1. For the rest records “SEVERITY_TARGET” will be set as 0.

3.4. Feature Selection

For the above sections, we have selected the following variables as the predictors. We would pass them into the machine learning algorithm and build a predictive model predict the occurrence of a high severity level accident.

<u>Select Variables:</u>

- COLLISIONTYPE - Collision type
- JUNCTIONTYPE - Category of junction at which collision took place
- WEATHER - A description of the weather conditions during the collision
- ROADCOND - The condition of the road during the collision
- LIGHTCOND - The light conditions during the collision
- PEDCOUNT - The number of pedestrians involved in the collision

<u>Target Variable:</u>

- SEVERITY_TARGET – if the “SEVERITYCODE” equals to “2 – injury”, the value will be set to 1; else set to 0.

4. Predictive Modeling

It is a supervised learning problem to predict the occurrence of the high severity level accident. We would try to build logistic regression to predict the probability of event occurrence.

4.1. Data Split – Training 80%, Testing 20%

We would split the dataset into Train set (80%) and Test set (20%).
The Train set will be passed into the machines learning algorithm for model training. The test set will be kept for model performance evaluation. After data split, the Train set contains 87537 records and Tet set have 21885 records

```
Train set: (87537, 44) (87537,)
Test set: (21885, 44) (21885,)
```

4.2. Data Balancing – Oversampling

As mention in the section 2.2 the dataset is an unbalanced dateset. The response records only occupied 30% of the total records. To balance the data for model building, the oversampling technique will be applied on the Train set. Oversampling will re-sample the minority response records with replacement to up-sample the number of response records to meet the number of majority records. This technique can emphasize the characteristics of significant predictors for building model.

Train set (Before Oversampling)	Train set (After Oversampling)
| | counts | per | per100 | | 0 | 61304 | 0.700321 | 70.0% | | 1 | 26233 | 0.299679 | 30.0% |	1 61304 0 61304 Name: SEVERITY_TARGET, dtype: int64

Oversampling only will be performed on Train set for building a model. For the Test set, we will keep its original distribution for accurate model performance evaluation.

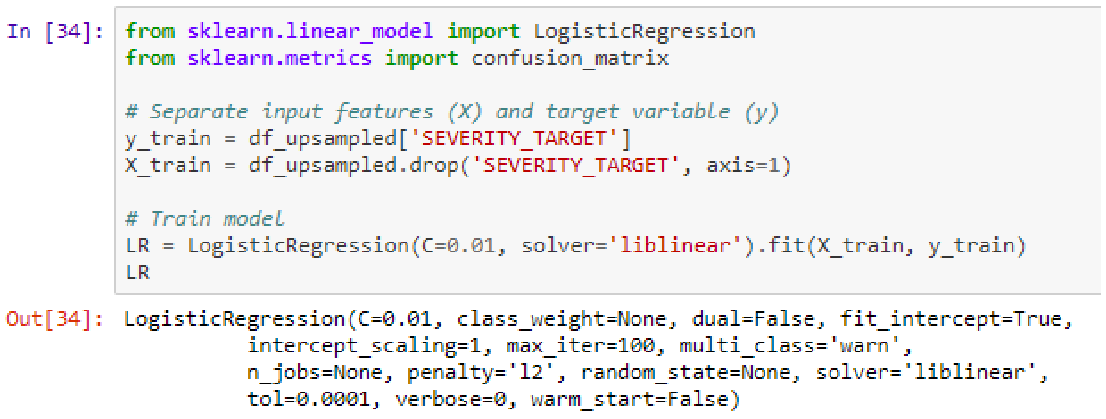
4.3. Classification Model – Logistic Regression

Logistic regression is a machine learning algorithm to model the probability of the respond records. The predicted probability is between 0 and 1. It can be used in classification problem. If it is larger than 0.5, we would classify the prediction would be “Will be happened”. If the predicted value is smaller then 0.5, the prediction would be “Will not be happened”. The threshold value 0.5 can be adjusted based on the business problem’s objectives. For some special problem, like fault detection, the business users are used to prefer a smaller threshold to reduce the false negative and capture fault case as more as they can. In this project, for simplicity, we will use 0.5 as the threshold value.

<u>Logistic Regression:</u>

$$\log \frac{p(x)}{1-p(x)} = \beta_0 + x \cdot \beta$$

$$p(x; b, w) = \frac{e^{\beta_0 + x \cdot \beta}}{1 + e^{\beta_0 + x \cdot \beta}} = \frac{1}{1 + e^{-(\beta_0 + x \cdot \beta)}}$$



```
In [34]: from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import confusion_matrix

         # Separate input features (X) and target variable (y)
         y_train = df_upsampled['SEVERITY_TARGET']
         X_train = df_upsampled.drop('SEVERITY_TARGET', axis=1)

         # Train model
         LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train, y_train)
         LR

Out[34]: LogisticRegression(C=0.01, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='warn',
                   n_jobs=None, penalty='l2', random_state=None, solver='liblinear',
                   tol=0.0001, verbose=0, warm_start=False)
```

4.4. Model Performance Evaluation

Three metrics are calculated using Test set to evaluate the model performance.

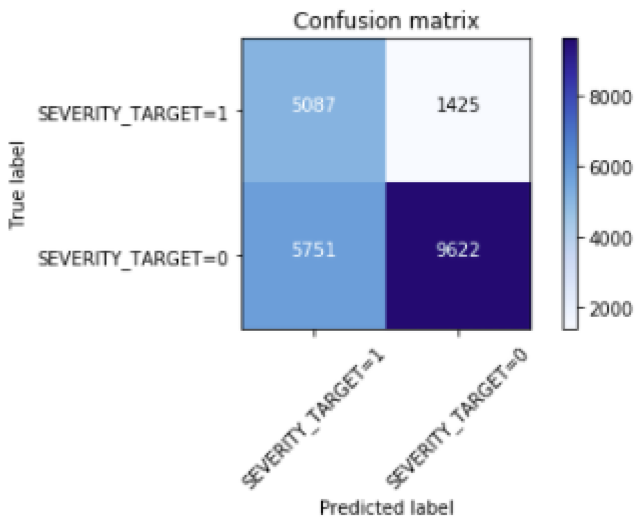
Accuracy Score

```
In [35]: yhat = LR.predict(X_test)
         yhat_prob = LR.predict_proba(X_test)

         # Accuracy
         print( 'Accuracy Score = ' + str(accuracy_score(y_test, yhat)) )

         Accuracy Score = 0.6721041809458533
```

The accuracy score is 0.6721 which means that the 67% of the response records are correctly predicted.

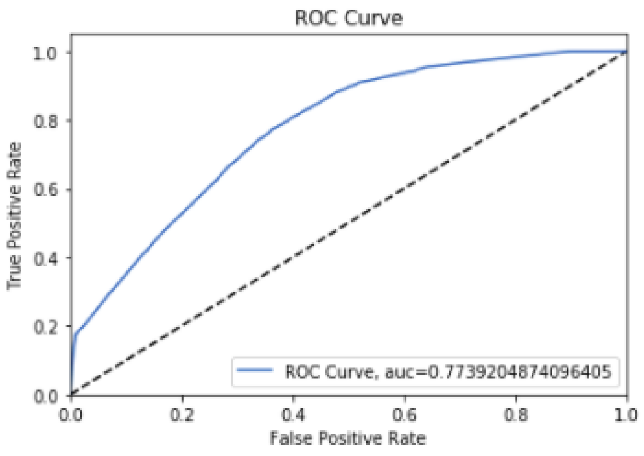
Confusion Matrix

```
Confusion matrix, without normalization
[[5087 1425]
 [5751 9622]]
```



From the confusion matrix, we can know the proportion of false positive and false negative records. In this case 0.5 threshold is used, there are 5751 records are false positive and 1425 records are false negative.
Since our project aims is to predict the occurrence of high severity level accident and prevent users suffer from traffic congestion, the false negative should be reduced. The false negative records occupy 6.51% (1425) of the Test set which are acceptable.

Area under ROC Curve

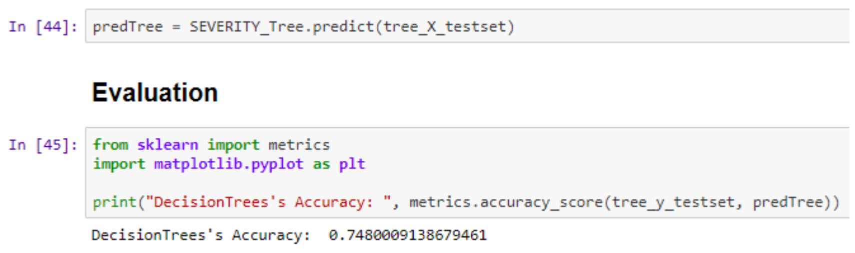


The area under ROC curve (AUC) measure how well the model can distinguish between the response. The higher the AUC, the better the model predictive power. The AUC=0.7739 means 77.39% that the model will be able to distinguish response.

5. Discussion

In the project, we use oversampling to balance the data which can avoid the information loss of the response records. However, there are some disadvantage of this method. Oversampling will up-sample the minority response records which would increase the likelihood of overfitting.
Beside oversampling, there are other technique to handle the unbalanced data. For example, undersample the majority class, use penalize algorithms to increase the cost of classification mistakes on the minority class, use Tree-based algorithm etc. They are also good methods to tackle the unbalanced data.
In this section, we will try decision tree to explore whether the model performance can be improved or not. Decision trees normally will have better performance on unbalanced data because the hierarchical structure of tree-based algorithms can learn from both minority and majority classes.

Decision Tree



```
In [44]: predTree = SEVERITY_Tree.predict(tree_X_testset)
```

Evaluation

```
In [45]: from sklearn import metrics
         import matplotlib.pyplot as plt

         print("DecisionTrees's Accuracy: ", metrics.accuracy_score(tree_y_testset, predTree))

         DecisionTrees's Accuracy:  0.7480009138679461
```

The accuracy is 0.7480 which means 74% of the response records can be correctly identified. The result is better than the logistic regression with 0.5 as the threshold value.

Since the project aims to predict the probability of the occurrence high severity level accident. Logistic regression can provide the predicted probability and users can adjust the threshold to decrease the false negative percentage until an acceptable level. Logistic regression is more preferred under this situation.

6. Conclusions

In this study, we have explored the collision data from Seattle and successfully built a logistic regression model to predict the occurrence of the high severity level accident. In the process, we use correlation matrix and chi-square test of independent to select significant numeric and categorical variable. Also, we use oversample technique to balance the dataset before model training. The result model can achieve 67% accuracy with only 6.51% false negative rate. The performance indicates that the model is sufficient to help drivers to avoid traffic congestion caused by high severity level road accident.