Capstone Project - Predicting the Occurrence of High Severity Level Road Accident

GARY POON

SEPTEMBER 25, 2020

Background & Problem

Traffic congestion is a big problem for all countries, especially in big cities. Traffic congestion always happens when there is a road accident. The higher severity level the accident is, the longer time the congestion lasted for.

The most effective solution is aware the road condition, route to other path before you were suffering from a traffic jam.

To avoid suffering, this project would like to predict the happens or even the probability of a high severity level road collision.

Data Source

Coursera shared data is used in this study. The data 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.

The variable "SEVERITYCODE" is selected as our target variable. There are two available values

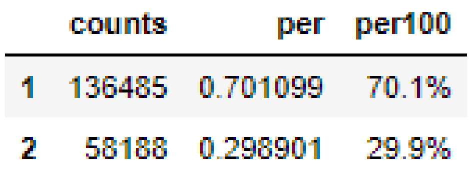
- "1 – prop damage"
- "2 – injury"

In this project will define **"2 – injury"** as a **high-level severity road accident** and predict its occurrence.

Data Cleaning

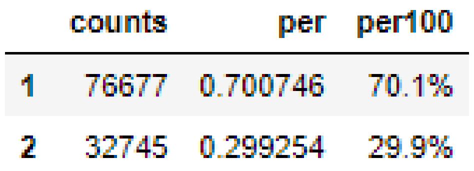
The distribution of the target variable “SEVERITYCODE” is studied.

- 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.



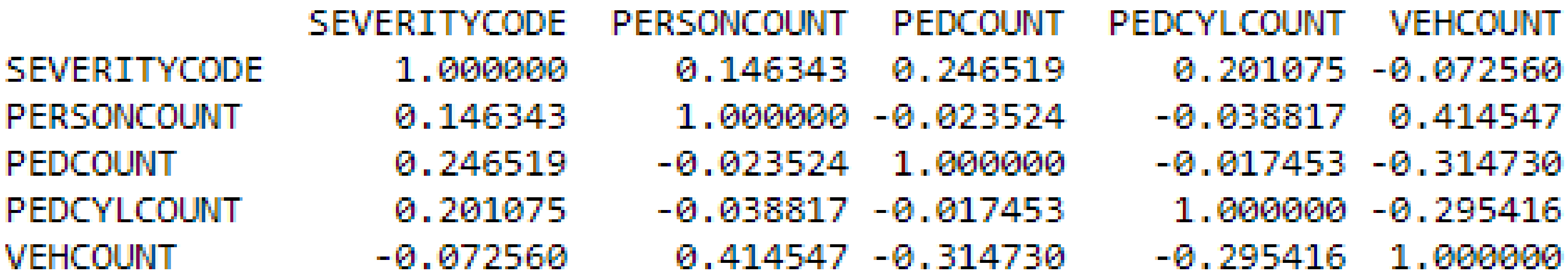
	counts	per	per100
1	136485	0.701099	70.1%
2	58188	0.298901	29.9%

- Drop the variables with >=80% records contain NaN or missing value
- 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.



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

Numeric Variables – Correlation Matrix



```
               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
```

- “PERSONCOUNT”, “PEDCOUNT” and “PEDCYLCOUNT” have positive relationship with the “SEVERITYCODE”
- Only “VEHCOUNT” has negative relationship

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.

- “PEDCOUNT” is selected as predictor

Categorical Variables – Chi-Square Test

Chi-Square Test:

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

Variable	P-value	Test Result
COLLISIONTYPE	< 0.05	Reject H0, there is a relationship with “SEVERITYCODE”
JUNCTIONTYPE	< 0.05	Reject H0, there is a relationship with “SEVERITYCODE”
WEATHER	< 0.05	Reject H0, there is a relationship with “SEVERITYCODE”
ROADCOND	< 0.05	Reject H0, there is a relationship with “SEVERITYCODE”
LIGHTCOND	< 0.05	Reject H0, there is a relationship with “SEVERITYCODE”

- “COLLISIONTYPE”, “JUNCTIONTYPE”, “WEATHER”, “ROADCOND” and “LIGHTCOND” are selected as predictors

Selected Features

Select Variables:

- 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

Target Variable:

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

Data Split – Training 80%, Testing 20%

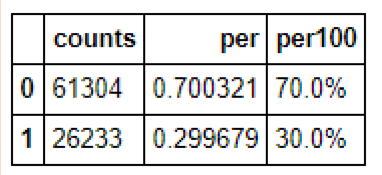
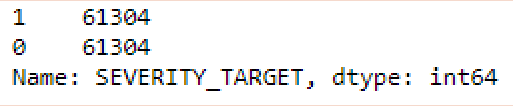
The Train set will be passed into the machines learning algorithm for model training.

The Test set will be kept for model performance evaluation.

- Train set contains 87537 records
- Tet set have 21885 records

Data Balancing – Oversampling

- 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.

<table>
<tr><th>Train set (Before Oversampling)</th><th>Train set (After Oversampling)</th></tr>
<tr>
<td>

<table>
<tr><th></th><th>counts</th><th>per</th><th>per100</th></tr>
<tr><td>0</td><td>61304</td><td>0.700321</td><td>70.0%</td></tr>
<tr><td>1</td><td>26233</td><td>0.299679</td><td>30.0%</td></tr>
</table>
</td>
<td>

<pre>
1   61304
0   61304
Name: SEVERITY_TARGET, dtype: int64
</pre>
</td>
</tr>
</table>

- 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

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.

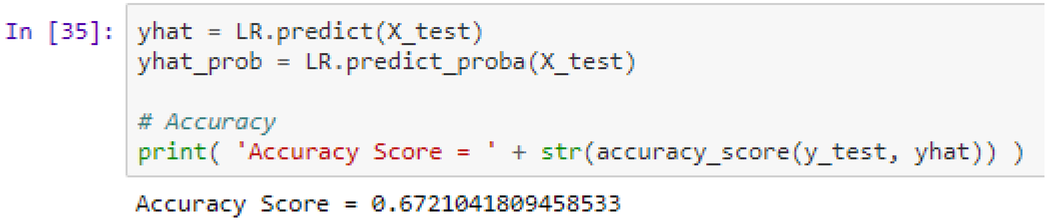
- If it is larger than 0.5, we would classify the prediction would be “Will be happened”.
- If the predicted value is smaller then 0.5, the prediction would be “Will not be happened”.

Logistic Regression

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

Model Performance Evaluation

Accuracy Score



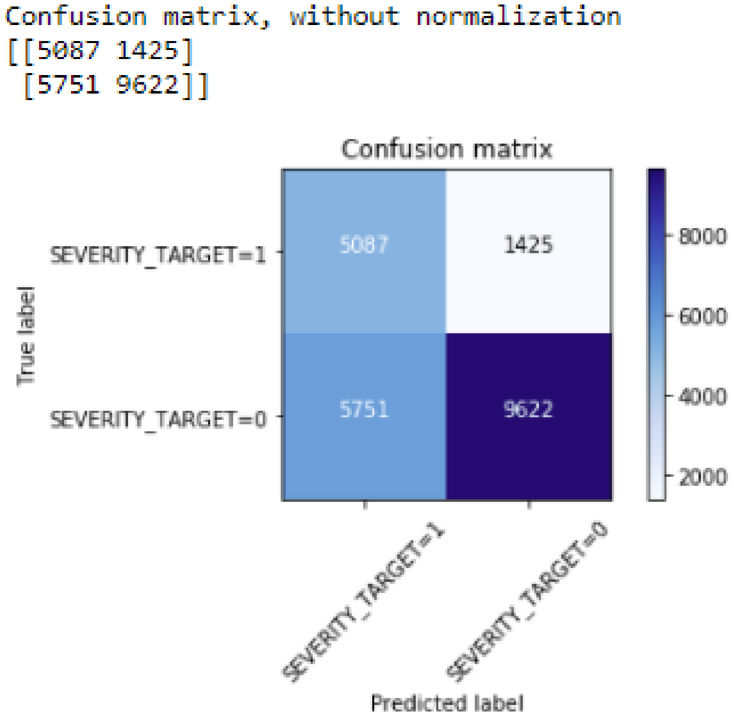
In [35]:
```
yhat = LR.predict(X_test)
yhat_prob = LR.predict_proba(X_test)

# 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.

Model Performance Evaluation (Cont)

Confusion Matrix



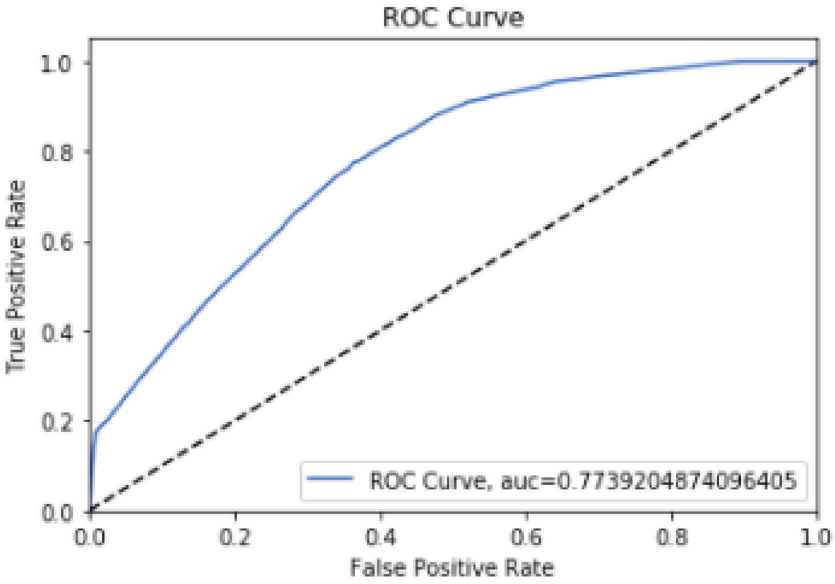
```
Confusion matrix, without normalization
[[5087 1425]
 [5751 9622]]
```

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.

Model Performance Evaluation

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.

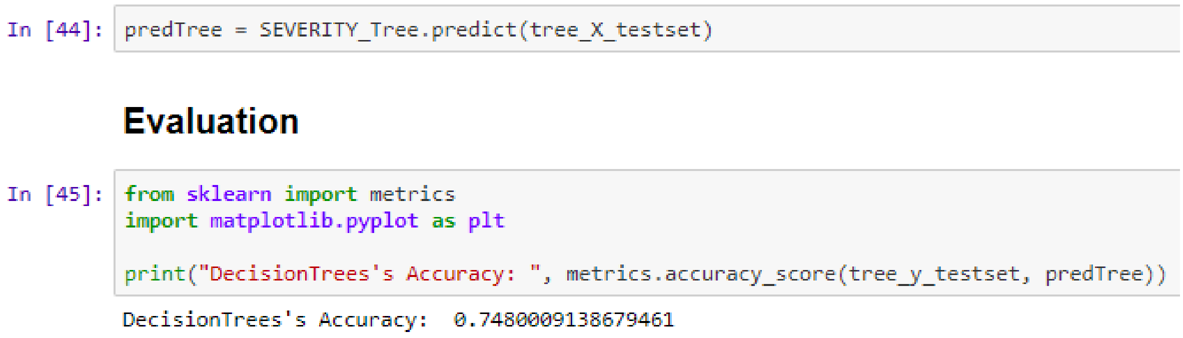
Discussion

Disadvantage:

➢Oversampling will up-sample the minority response records which would increase the likelihood of overfitting.

Tree-based algorithm:

➢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.



```
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 result is better than the logistic regression with 0.5 as the threshold value.
- 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.**

Conclusion

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.

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.