# COURSERA DATA SCIENCE CAPSTONE

Anshuman Harshwardhan

# 1. Introduction/Business Problem

As the cities become larger with growing urbanization, so does the road traffic. Road traffic comprises of different participants, from pedestrians and cyclists to large vehicles. And it becomes very important to not only ensure smooth traffic flow but also road safety. Improper traffic conditions can impede the economic development in city and make a city less competent and attractive than their peers for business. Road Safety requires in-depth understand and insights from the data to ensure that we learn from the historical errors. A data model based on the historical data, would be able to provide the insights model the severity of accidents based on multiple factors. This in turn would be able to provide help in putting place measures, signs, improvements etc. which would help to reduce accidents and improve road safety conditions. In the current situation, based on the Data available for Seattle (from SDOT Traffic Management Division) the goal is to develop a model which predicts Severity of the accident based on multiple factors. The model would help in testing and provide guidance before implementation of measures.

# 2. Data

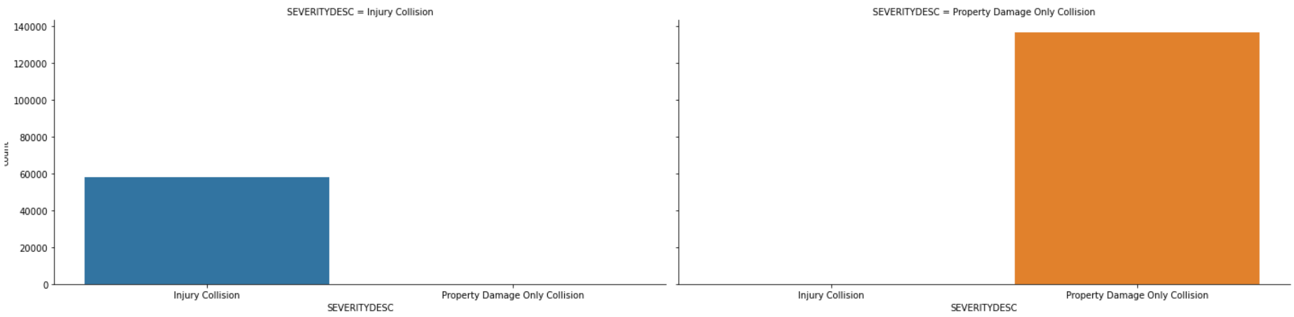
The data file used is "Collision - All Years" which has been sourced from SDOT Traffic Management Division. The data has 38 attributes and covers the time period from 2004 to present date.

Each of the data point (rows) has a Severity Code which specific the severity of the accident and has details about various other attributes (like Junction Type, Weather condition, Light condition, Road conditions etc.). The aim of the model would be to model Severity based on the multiple attributed (below table has correlations).

```
SEVERITYCODE         1.000000
PERSONCOUNT          0.130949
PEDCOUNT             0.246338
PEDCYLCOUNT          0.214218
VEHCOUNT            -0.054686
SDOT_COLCODE         0.188905
OBJECTID             0.020131
INTKEY               0.104973
LIGHTCOND_N         -0.038968
ROADCOND_N          -0.043377
WEATHER_N           -0.099871
INATTENTIONIND_N     0.046378
UNDERINFL_N          0.041599
HITPARKEDCAR_N      -0.101498
SPEEDING_N           0.038938
COLLISIONTYPE_N     -0.112318
JUNCTIONTYPE_N      -0.136318
PEDROWNOTGRNT_N      0.206283
```

For Reference Type of Data Attributed
Further Metadata Description from Source : [https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf](https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf)

The data is split into 2 kinds of severity ”Injury collision” and “Property Damage Only Collision”. As evident from the below plot the data is lopsided towards “Property Damage Only Collision” type




# 3. Methodology

The data has multiple attributed which can be used for prediction of Severity. Post intimal analysis a predictor matrix was arrived at. Predictor matrix was further refined using Recursive Feature Elimination and then implemented to further refine the attributed.

As the output of the model is Binomial, Logistic Regression should be used to develop a model. Categorical variables were updated to numbers and the SEVERITYCODE was updated to represent 0 or 1 as the values.

Moreover, once the predictor variables are finalized, the performance of the below models would be compared

- Logistic Regression
- Gaussian Naïve Bayes
- Random Forest Classifier

The below attributed were selected as predictor variables for the model

```
Predictor_Matrix=
     ['PERSONCOUNT',
     'PEDCOUNT',
     'PEDCYLCOUNT',
     'VEHCOUNT',
     ‘SDOT_COLCODE',
     ‘OBJECTID’,
     'INTKEY',
     'LIGHTCOND_N',
     'ROADCOND_N',
     'WEATHER_N',
     ’INATTENTIONIND_N',
     'UNDERINFL_N',
     'HITPARKEDCAR_N',
     'SPEEDING_N',
     'COLLISIONTYPE_N',
     'JUNCTIONTYPE_N',
     'PEDROWNOTGRNT_N']
```

## 3.1. Recursive Feature Elimination

Recursive Feature Elimination has been used to further refine the selected features. The Logistic Regression with solver ‘liblinear’ has been used.

```
Output :
[ True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

## 3.2. Model Implementation using statsmodel.api

Based on the output all the predictor variable will be retained and we will further implement the model using Python package statsmodel.api to further refine the predictor.

```
Optimization terminated successfully.
         Current function value: 0.521967
         Iterations 7
                          Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: 0.144
Dependent Variable: y                AIC:              162771.8083
Date:               2020-09-08 01:53 BIC:              162941.0755
No. Observations:   155889           Log-Likelihood:   -81369.
Df Model:           16               LL-Null:          -95085.
Df Residuals:       155872           LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     7.0000
-------------------------------------------------------------------
           Coef.    Std.Err.      z       P>|z|     [0.025   0.975]
-------------------------------------------------------------------
x1         0.2173    0.0054    40.6165   0.0000    0.2068   0.2278
x2         2.8881    0.0492    58.6689   0.0000    2.7916   2.9846
x3         2.3197    0.0513    45.2048   0.0000    2.2191   2.4203
x4         0.2894    0.0124    23.3720   0.0000    0.2651   0.3137
x5         0.0378    0.0012    31.2503   0.0000    0.0354   0.0401
x6         0.0196    0.0314     0.6230   0.5333   -0.0420   0.0811
x7         0.9176    0.1327     6.9171   0.0000    0.6576   1.1775
x8        -0.0190    0.0037    -5.1619   0.0000   -0.0262  -0.0118
x9         0.0348    0.0027    13.0637   0.0000    0.0296   0.0400
x10       -0.1088    0.0038   -28.5075   0.0000   -0.1163  -0.1014
x11        0.3446    0.0162    21.2451   0.0000    0.3128   0.3764
x12        0.0560    0.0086     6.5487   0.0000    0.0393   0.0728
x13       -1.4588    0.0557   -26.2067   0.0000   -1.5679  -1.3497
x14        0.6276    0.0252    24.8800   0.0000    0.5782   0.6771
x15       -0.0714    0.0024   -29.3687   0.0000   -0.0762  -0.0667
x16       -0.1473    0.0052   -28.2989   0.0000   -0.1575  -0.1371
x17        0.8650    0.0428    20.2136   0.0000    0.7811   0.9488
=================================================================
```

Based on the output summary below the variable X6 (OBJECTID) has p-value greater than 0.05 so we will drop the attribute and update the predictor variable to the below and re-run summary.

```
Predictor_Matrix_N=
      ['PERSONCOUNT',
      'PEDCOUNT',
      'PEDCYLCOUNT',
      'VEHCOUNT',
      'SDOT_COLCODE',
      'INTKEY',
      'LIGHTCOND_N',
      'ROADCOND_N',
      'WEATHER_N',
```

```
'INATTENTIONIND_N',
'UNDERINFL_N',
'HITPARKEDCAR_N',
'SPEEDING_N',
'COLLISIONTYPE_N',
'JUNCTIONTYPE_N',
'PEDROWNOTGRNT_N']
```

```
Optimization terminated successfully.
         Current function value: 0.521968
         Iterations 7
                         Results: Logit
==================================================================
Model:              Logit            Pseudo R-squared: 0.144
Dependent Variable: y                AIC:              162770.1964
Date:               2020-09-08 02:00 BIC:              162929.5068
No. Observations:   155889           Log-Likelihood:   -81369.
Df Model:           15               LL-Null:          -95085.
Df Residuals:       155873           LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     7.0000
--------------------------------------------------------------------
           Coef.    Std.Err.      z       P>|z|     [0.025    0.975]
--------------------------------------------------------------------
x1         0.2174    0.0053    40.6612   0.0000    0.2069    0.2279
x2         2.8899    0.0492    58.7902   0.0000    2.7935    2.9862
x3         2.3209    0.0513    45.2567   0.0000    2.2204    2.4214
x4         0.2878    0.0121    23.7450   0.0000    0.2641    0.3116
x5         0.0377    0.0012    31.3228   0.0000    0.0353    0.0400
x6         0.9204    0.1326     6.9430   0.0000    0.6606    1.1803
x7        -0.0193    0.0037    -5.2554   0.0000   -0.0264   -0.0121
x8         0.0348    0.0027    13.0676   0.0000    0.0296    0.0400
x9        -0.1090    0.0038   -28.5837   0.0000   -0.1165   -0.1015
x10        0.3447    0.0162    21.2548   0.0000    0.3129    0.3765
x11        0.0599    0.0059    10.0772   0.0000    0.0482    0.0715
x12       -1.4549    0.0553   -26.2996   0.0000   -1.5634   -1.3465
x13        0.6258    0.0251    24.9712   0.0000    0.5767    0.6750
x14       -0.0715    0.0024   -29.5194   0.0000   -0.0763   -0.0668
x15       -0.1471    0.0052   -28.3158   0.0000   -0.1572   -0.1369
x16        0.8563    0.0405    21.1651   0.0000    0.7770    0.9356
==================================================================
```

# 4. Results

## 4.1. Logistic Regression Model Fitting , Confusion Matrix and Results

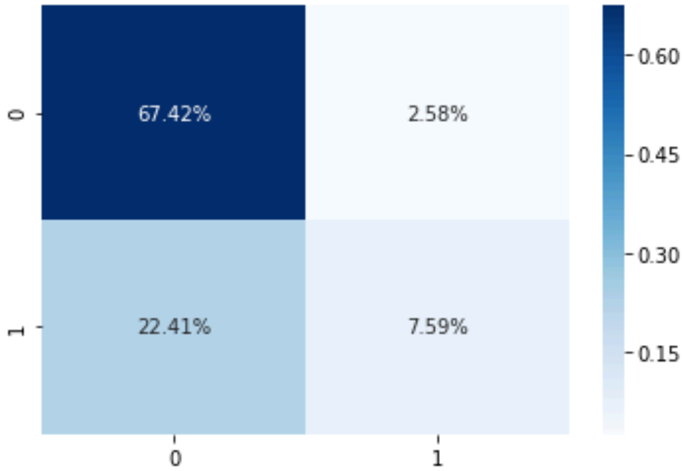
Below are the final results for 3 models which were used for modelling.

- Logistic Regression
- Gaussian Naïve Bayes
- Random Forest Classifier

In terms of the confusion matrix, the best performance is the model which uses Random forest Classifier.

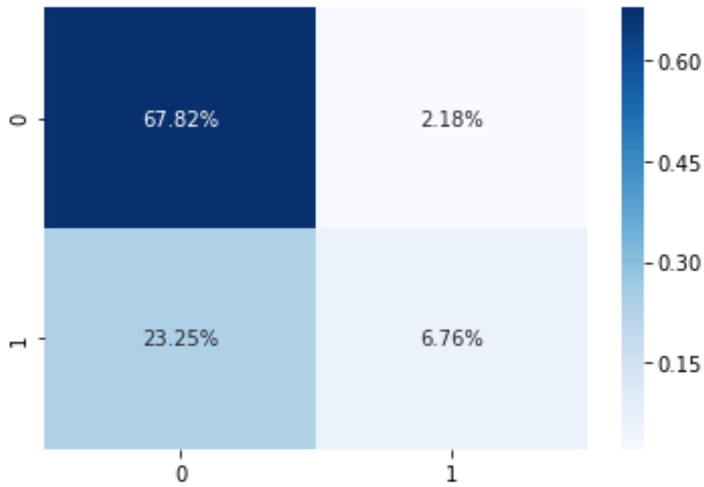
```
LOGISTIC REGRESSION
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.96      0.84     27221
           1       0.75      0.25      0.38     11669

   micro avg       0.75      0.75      0.75     38890
   macro avg       0.75      0.61      0.61     38890
weighted avg       0.75      0.75      0.70     38890

Confusion Matrix
```

<AxesSubplot:>

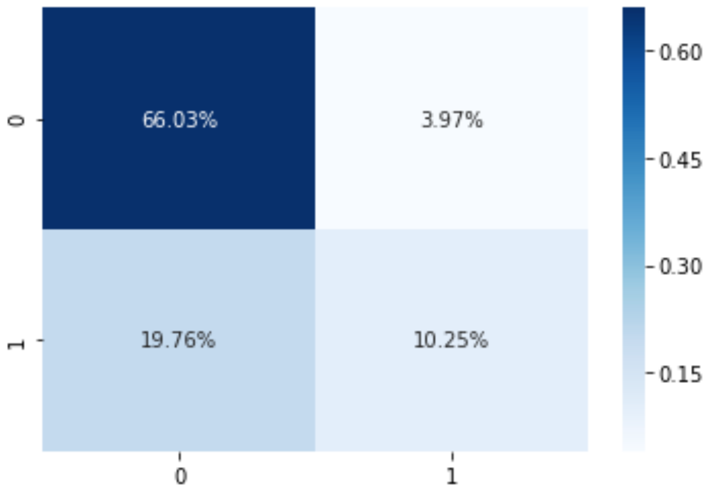



```
GAUSSIAN NAIVE BAYES
Classification Report
              precision    recall  f1-score   support

           0       0.74      0.97      0.84     27221
           1       0.76      0.23      0.35     11669

   micro avg       0.75      0.75      0.75     38890
   macro avg       0.75      0.60      0.59     38890
weighted avg       0.75      0.75      0.69     38890

Confusion Matrix
```

<AxesSubplot:>



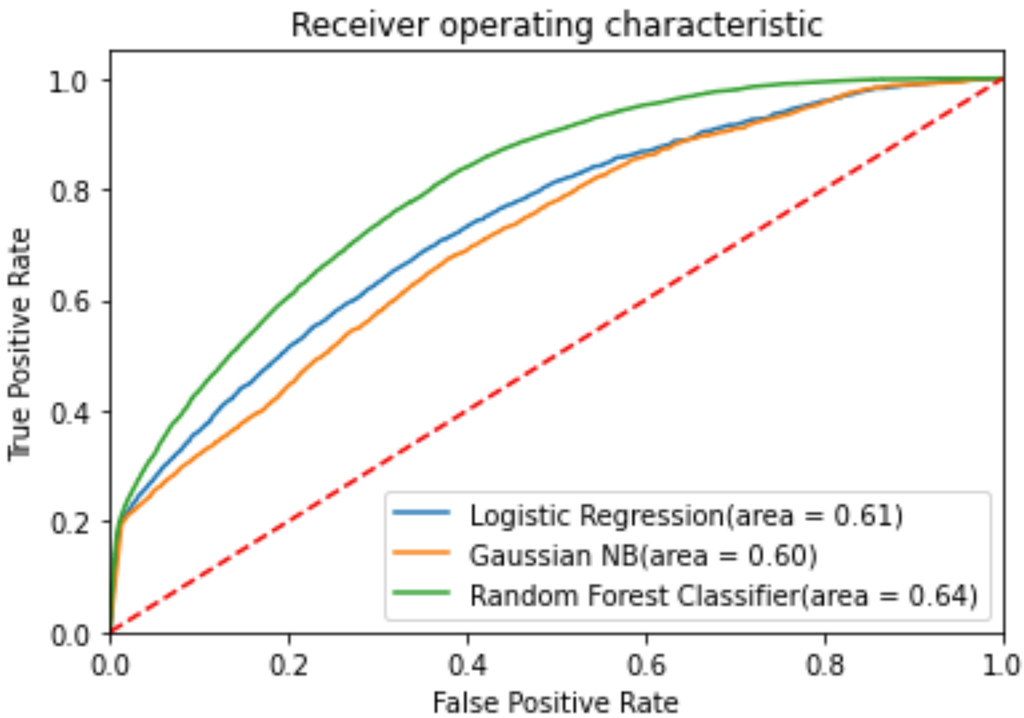
```
RANDOM FOREST CLASSIFIER
Classification Report
              precision    recall  f1-score   support

           0       0.77      0.94      0.85     27221
           1       0.72      0.34      0.46     11669

   micro avg       0.76      0.76      0.76     38890
   macro avg       0.75      0.64      0.66     38890
weighted avg       0.76      0.76      0.73     38890

Confusion Matrix
<AxesSubplot:>
```




## 4.2. Precision and ROC Curve

ROC (Receiver operating characteristic) curve illustrates diagnostic ability. Good classifier stays away from the red line. The graph is basically plot pf true positive rate and false positive rate at different thresholds.

Based on the below graph, the Random Forest Classifier has the best performance



## 5. Conclusion

Based on the dataset a model to predict severity was initially developed using LogisticRegression and then further modelled using the below 3 models.

- Logistic Regression
- Gaussian Naïve Bayes
- Random Forest Classifier

Random Forest Classifier model was able to achieve 75% success rate based on the test data.

In the given model, the model is predicting severity as two outcomes only (Injury Collision and Property Damage Collision). As a next step the larger data set with multiple severity can be used to develop an advanced model. Moreover, the data can be used to also gain insights to ensure that adequate measures are taken and implemented to ensure Road Safety.