IBM Data Science Capstone Project - Seattle, Washington Car accident Severity

By

Chukwuemeka Okwuonu

Table of Contents

1 Introduction/Business Problem

Road accident has been a major human issue for ages, with the government enforcing more safety rules and standards, vehicle manufactures adding more safety feature to their vehicles, the fatality rate of car accident has not really reduced.

A study of Seattle Department of Transportation traffic accident cases from 2004 to 2020, almost 16 years, provides a good insight to some of the reasons behind these accident/collisions. It goes from driving at high speed, alcohol and drug abuse to lack of attention/focus while during driving – use of cell phone.

Other feature observed from that data includes weather, visibility and road conditions. This project will analyze the collision dataset, find patterns and determinate key factors using various analytical techniques and machine learning classification algorithms such as logistic regression, decision tree analysis, k-nearest-neighbors, support vector machine, etc. and predict the different accidents severity.

The target audience for this project is as follows: -

1. The Seattle local government: - it will help improve traffic policies, update public facilities such as streetlight, traffic signs and alert etc.
2. Police: - Enforcing the law and make sure traffic laws are following and distracted drivers are taken off the road.
3. Car insurance institutes: - Check driver’s license and records before issuing.
4. ALL DRIVERS: - To follow/obey all traffic signs and laws, knowing that safety is very important and what their family or love ones will pass through if there was an accident.

The Project's objective is to create a way to warn people based on factors such as weather and road conditions, the possibility of getting into a car accident and how severe it would be. So that the end user can drive more carefully or change travel plans if necessary.

2 Data

The dataset for this project is from Seattle, Washington Police Department and Accident Traffic Records Department from 2004 to present, provided by Coursera IBM capstone project for downloading through a link. It consists of 37 independent variables and 194,673 rows, also a dependent variable, SEVERITYCODE, that contains numbers which corresponds to different levels of severity caused by an accident/collision.
Severity codes are as follows:
0: Little to no Probability (Clear Conditions)
1: Very Low Probability — Chance or Property Damage
2: Low Probability — Chance of Injury
3: Mild Probability — Chance of Serious Injury - 4: High Probability — Chance of Fatality

The indicator SEVERITYCODE is chosen as the dependent variable while other attributes like nature and human factors will enable us to build a model that will predict the chance of an accident and how severe it would be based on them.
Human factors are as follows INATTENTIONIND, UNDERINFL, and SPEEDING which shows the concentration of the driver's mind, drug or alcohol influence and overspeed. Nature factors are made up by WEATHER, ROADCOND and LIGHTCOND which represents weather, road and view circumstance respectively. The dataset needs to be process before use, drop non-relevant columns and for the features, convert object data types into numerical data types. The target feature "SEVERITYCODE" is imbalance, simple statistical technique is used to balance it.

To fully understand the data, I run a value of the following features/attributes to those having most influence.

- ROADCOND: - The condition of the road during the collision.
- WEATHER: - A description of the weather conditions during the time of the collision
- LIGHTCOND: -The light conditions during the collision.
- UNDERINFL: - Whether a driver involved was under the influence of drugs or alcohol.

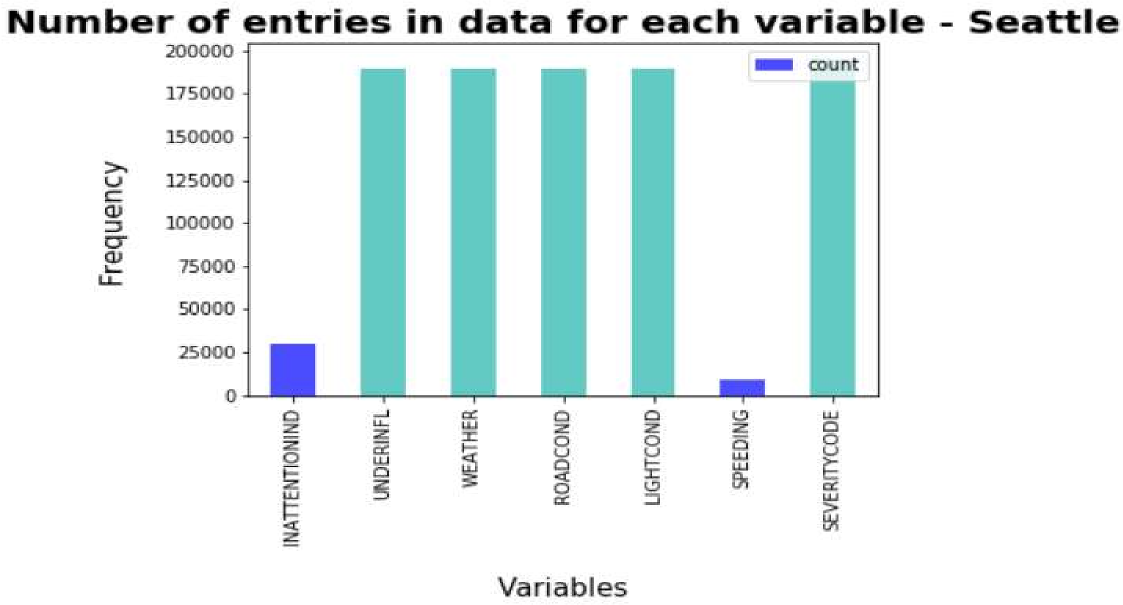
The results can be seen below:

Notebook from the IBM WATSON Studio is used to process the dataset and will equally be used to build Machine Learning models. The Github repository enables the link sharing.
Packages and libraries: We will use libraries and packages for both data manipulation and data visualization. PANDA, NUMPY, SCIPY, Matplotlib, Seaborn

3 Methodology

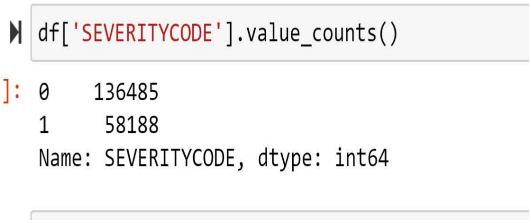
3.1 Data Processing

Notebook from the IBM WATSON Studio is used to process the dataset. The data was imported, data types, shape and information of data was carefully studied and descriptive statistics bar plots of some variables to well understand their counts or frequency of data entry. This operation was useful in detecting missing values or possibly errant data



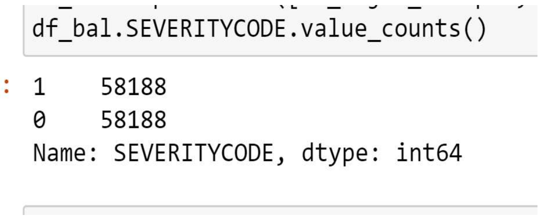
Looking at the above plot and after analyzing the dataset, the focus for an accurate result will be on only four features, WEATHER, ROADCOND, LIGHTCOND, UNDERINFL, among others with SEVERITYCODE been dependent variable.

Severity Code was converted from (1/2) to (0/1) and a value counts shows huge imbalance between class 0 which is Property Damage and class 1 Physical Injury. The is an unbalanced dataset where the distribution of the target variable is in almost 1:2 ratio in favor of property damage. It is very important to have a balanced dataset when using machine learning algorithms. Statistical technique is used to balance class 0 and class 1



```
df['SEVERITYCODE'].value_counts()
```

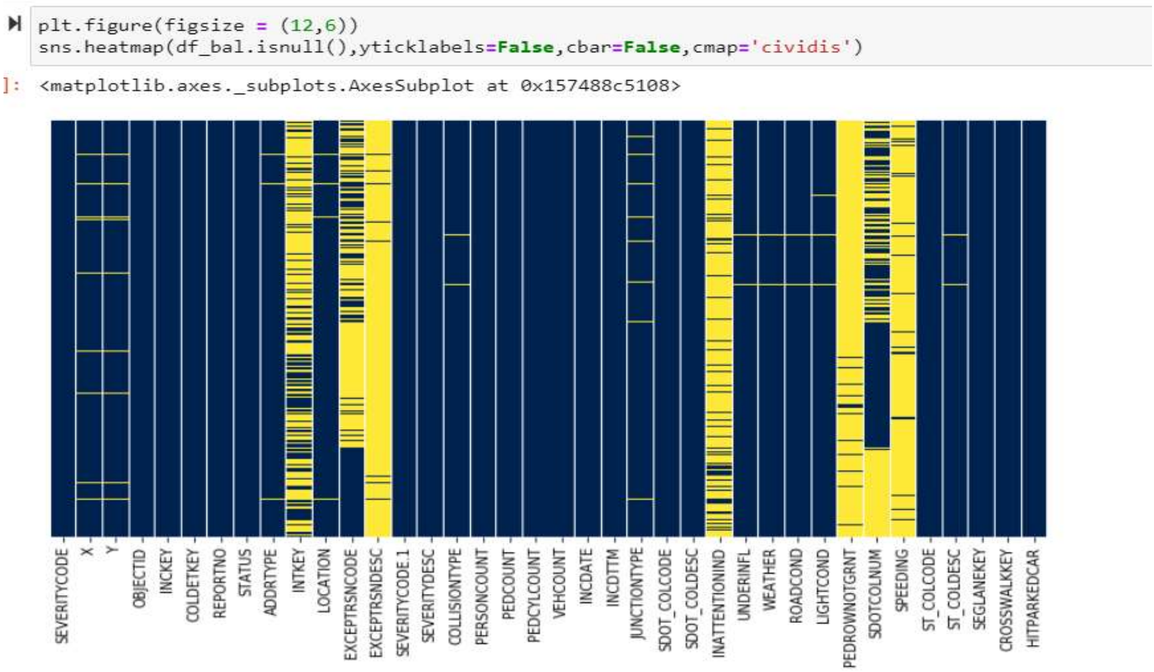
```
]: 0    136485
   1     58188
   Name: SEVERITYCODE, dtype: int64
```



```
df_bal.SEVERITYCODE.value_counts()
```

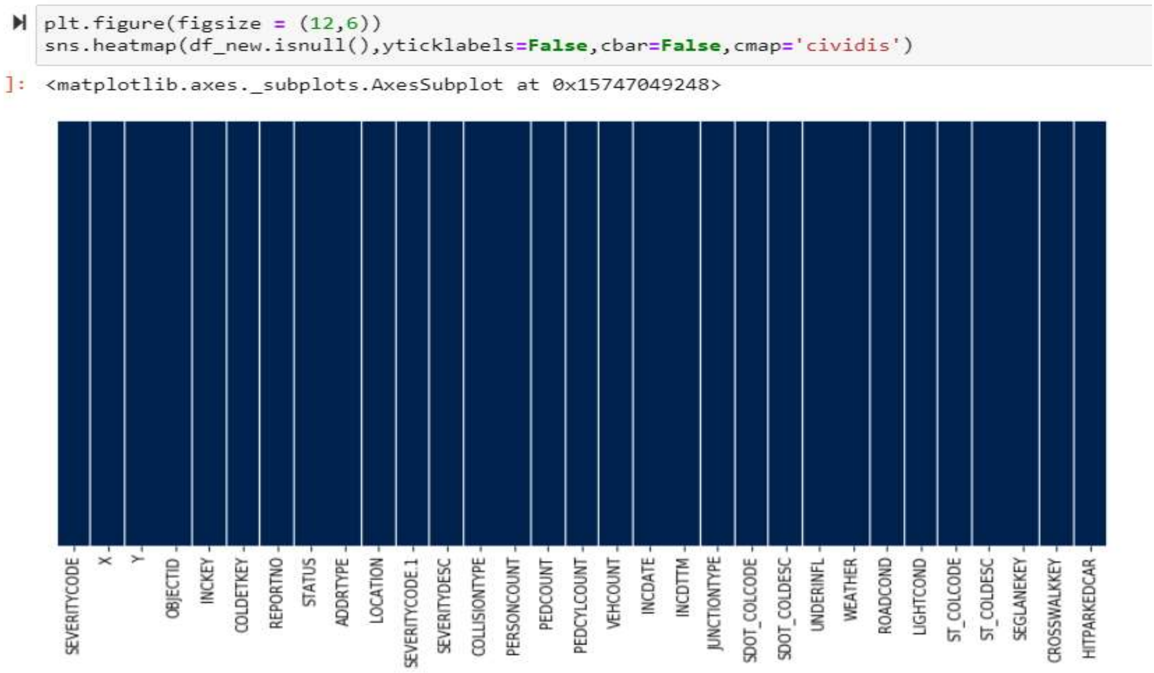
```
: 1    58188
  0    58188
  Name: SEVERITYCODE, dtype: int64
```

Missing Data Heatmap was the technique used in the process of detecting and correcting (or removing) corrupt or inaccurate records from the dataset. With the removal of all unnecessary columns /null data from dataset, we now have a good quality data to work with.



```
plt.figure(figsize = (12,6))
sns.heatmap(df_bal.isnull(),yticklabels=False,cbar=False,cmap='cividis')
```

```
]: <matplotlib.axes._subplots.AxesSubplot at 0x157488c5108>
```



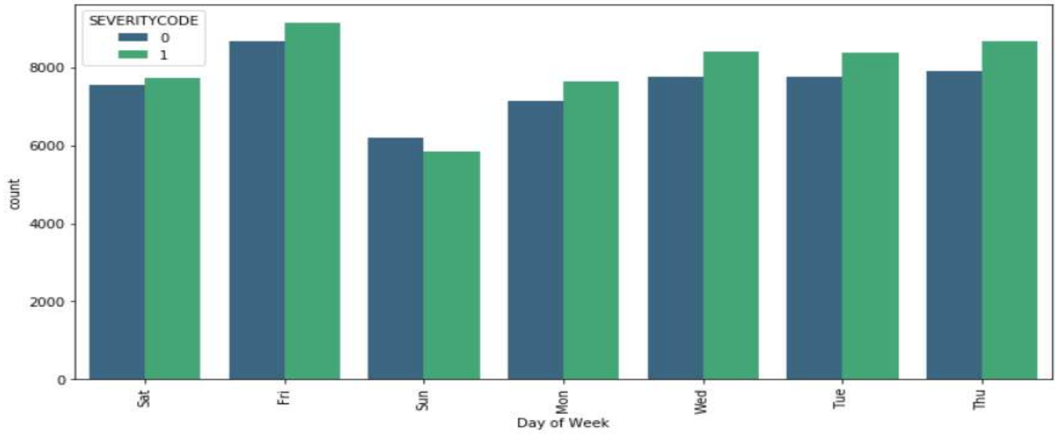
```
plt.figure(figsize = (12,6))
sns.heatmap(df_new.isnull(),yticklabels=False,cbar=False,cmap='cividis')
```

```
]: <matplotlib.axes._subplots.AxesSubplot at 0x15747049248>
```

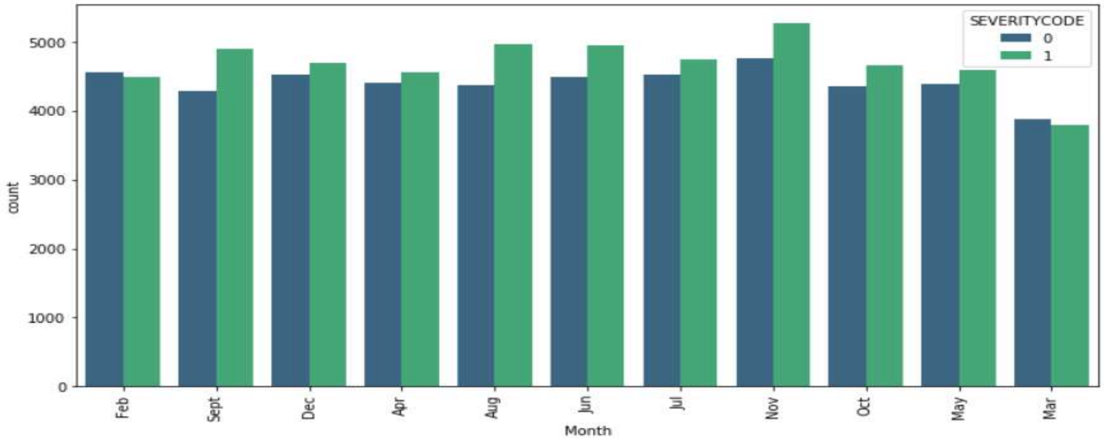
3.2 Analyzing Individual Feature Patterns using Visualization

To further understand and analysis the data, different plots were made to see the relationship between individual features and severity code - Day of the week, Month and the four main features/attributes to severity code and number of accidents.

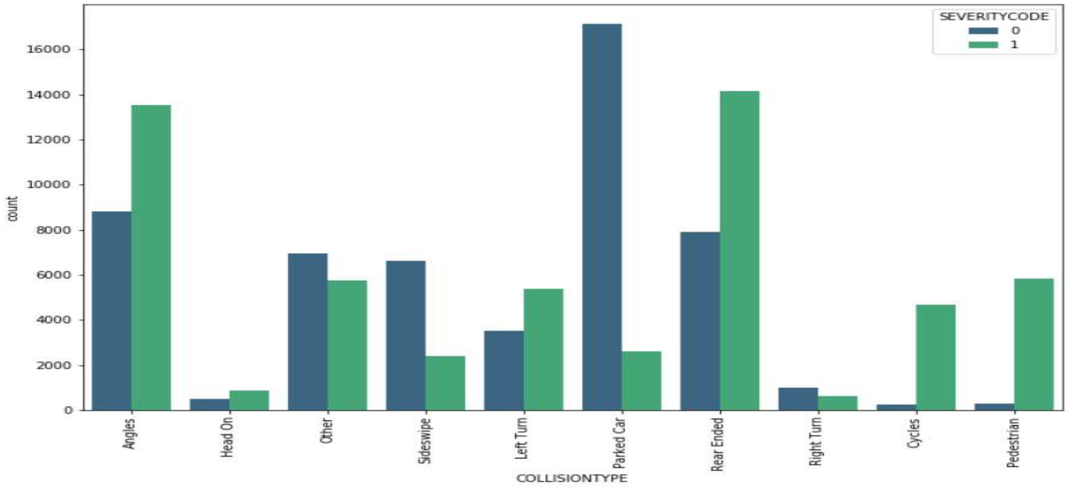
3.2.1 Day of The Week



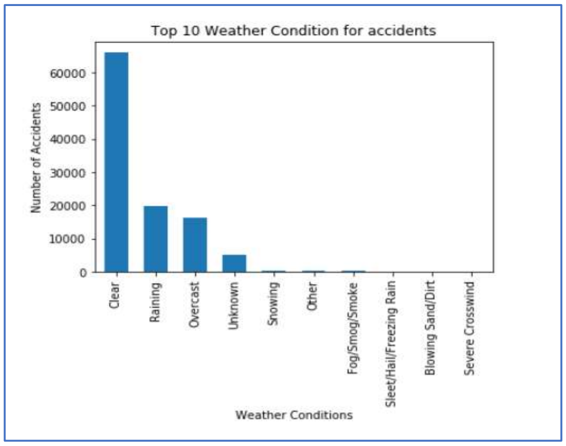
3.2.2 Month

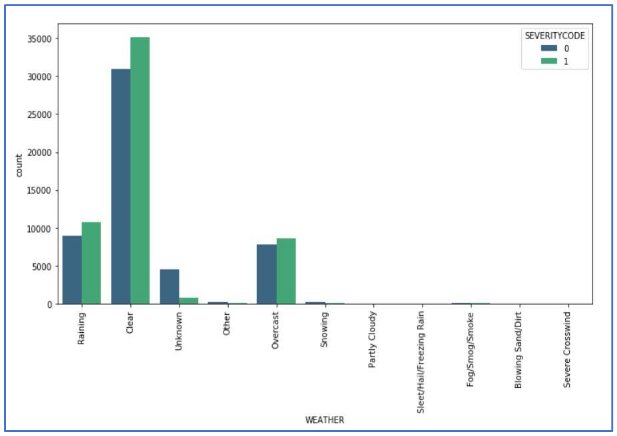


3.2.3 Collision type Vs Severity Code

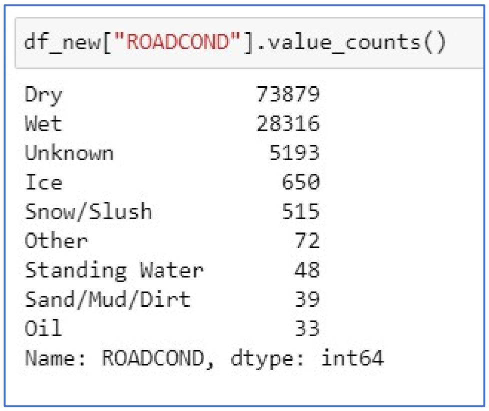


3.2.4 Weather

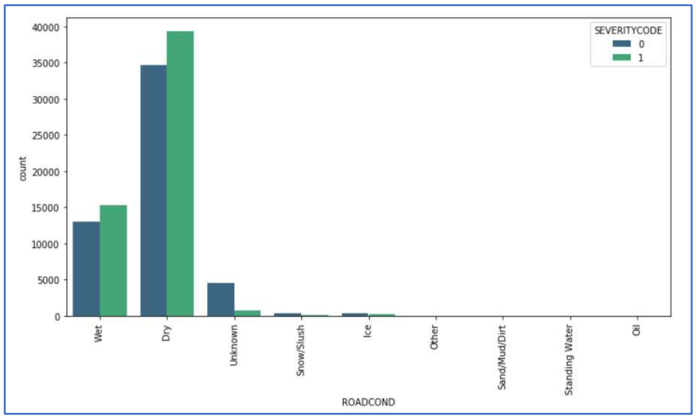




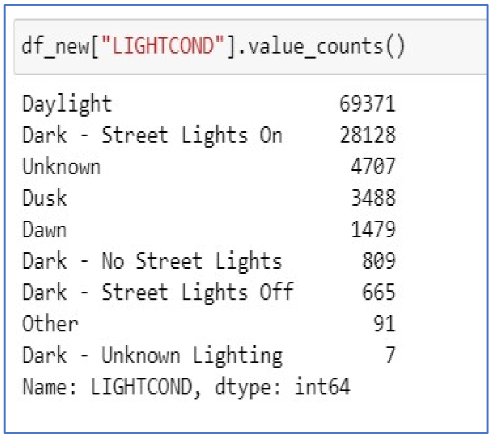
3.2.5 Road Condition



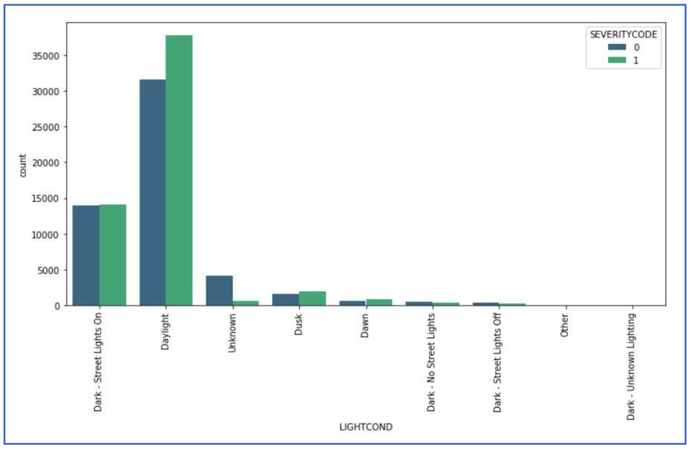
```
df_new["ROADCOND"].value_counts()

Dry               73879
Wet               28316
Unknown            5193
Ice                 650
Snow/Slush          515
Other                72
Standing Water       48
Sand/Mud/Dirt        39
Oil                  33
Name: ROADCOND, dtype: int64
```



3.2.6 Light Condition

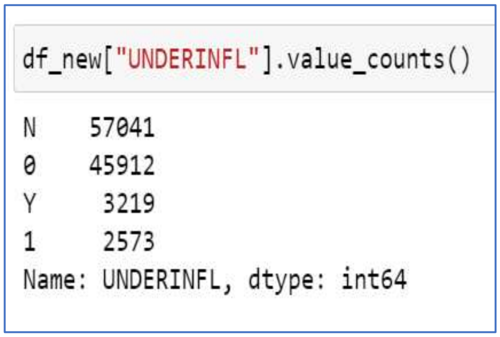


```
df_new["LIGHTCOND"].value_counts()

Daylight                    69371
Dark - Street Lights On     28128
Unknown                      4707
Dusk                         3488
Dawn                         1479
Dark - No Street Lights       809
Dark - Street Lights Off      665
Other                          91
Dark - Unknown Lighting         7
Name: LIGHTCOND, dtype: int64
```



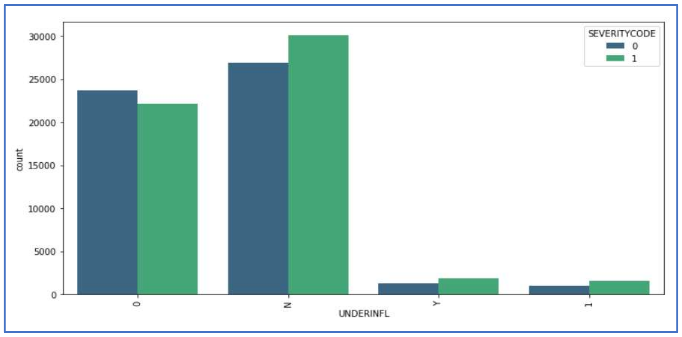
3.2.7 Underinfl :

Whether or not a driver involved was under the influence of drugs or alcohol.



```
df_new["UNDERINFL"].value_counts()
```

```
N    57041
0    45912
Y     3219
1     2573
Name: UNDERINFL, dtype: int64
```

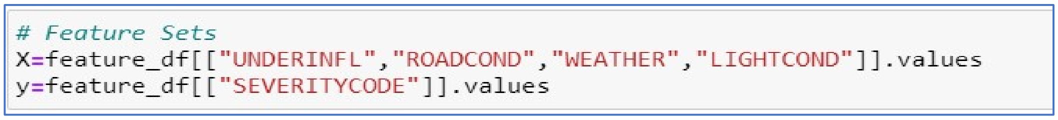


After balancing SEVERITYCODE feature, and standardizing the input features, the data is now ready for machine learning models building.

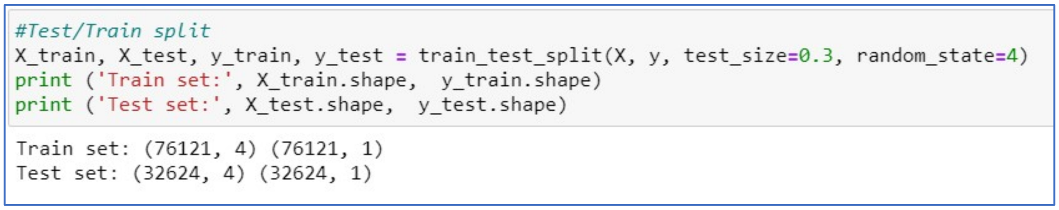
The following algorithms will be used in this project:

- K Nearest Neighbour - Grouping data points into categories/ groups based on similarity measures (or distance in between)
- Decision Tree - Breaking down the prediction into smaller subsets and generating a tree-like logic flow to model the prediction
- Logistic Regression - Using logistic functions to model binary output (dependent variable)

X and Y is defined for our dataset, we split our dataset into train and test set after normalization of dataset.



```
# Feature Sets
X=feature_df[["UNDERINFL","ROADCOND","WEATHER","LIGHTCOND"]].values
y=feature_df[["SEVERITYCODE"]].values
```



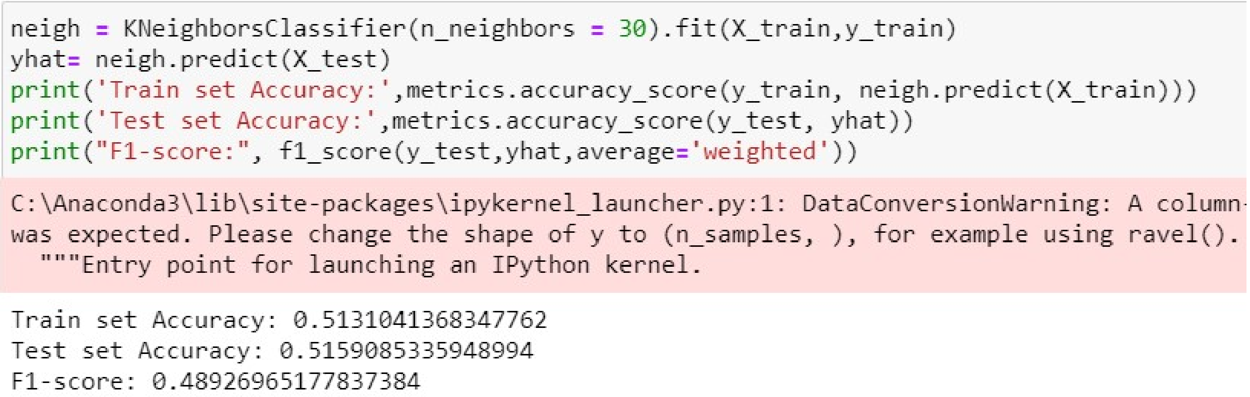
```
#Test/Train split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=4)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (76121, 4) (76121, 1)
Test set: (32624, 4) (32624, 1)
```

4 Results

4.1 K Nearest Neighbour

A simple algorithm that stores all available cases and classifies new cases based on a similarity measure i.e. distance. Obtain the best k at 30 to build the model with the best accuracy.

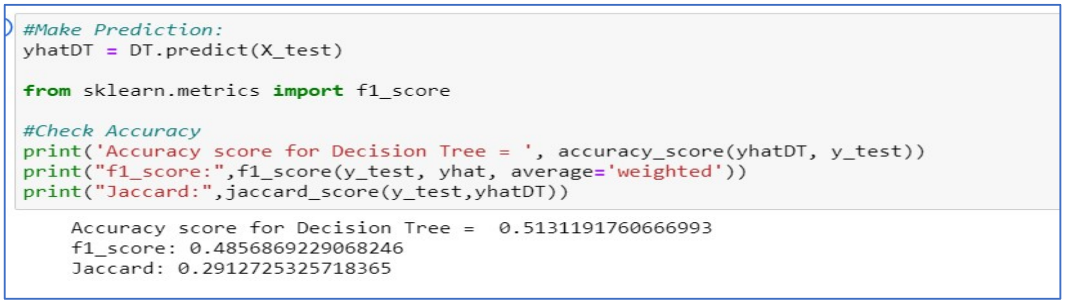


```
neigh = KNeighborsClassifier(n_neighbors = 30).fit(X_train,y_train)
yhat= neigh.predict(X_test)
print('Train set Accuracy:',metrics.accuracy_score(y_train, neigh.predict(X_train)))
print('Test set Accuracy:',metrics.accuracy_score(y_test, yhat))
print("F1-score:", f1_score(y_test,yhat,average='weighted'))
```

```
C:\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DataConversionWarning: A column-
was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  """Entry point for launching an IPython kernel.

Train set Accuracy: 0.5131041368347762
Test set Accuracy: 0.5159085335948994
F1-score: 0.48926965177837384
```

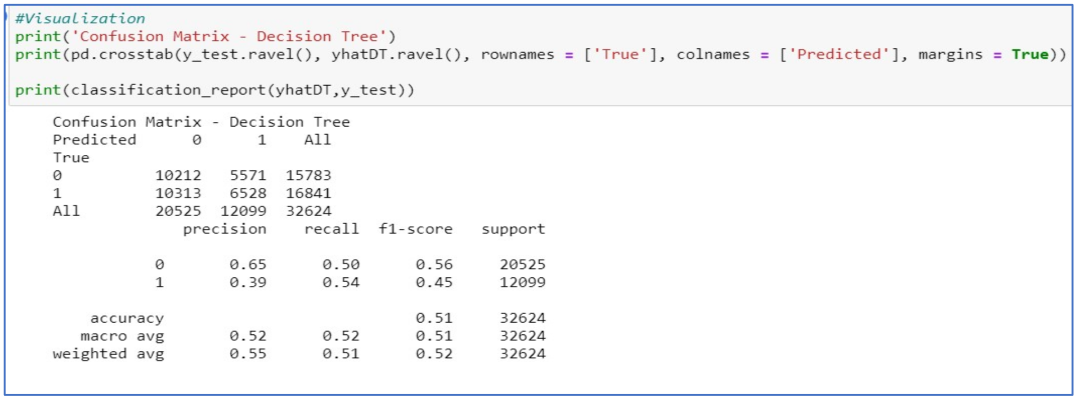
4.2 Decision Tree

A flowchart-like structure in which each internal node represents a test on a feature, thus create a model that predicts the value of a target variable. The criterion chosen for the classifier was entropy and the max depth was 6.



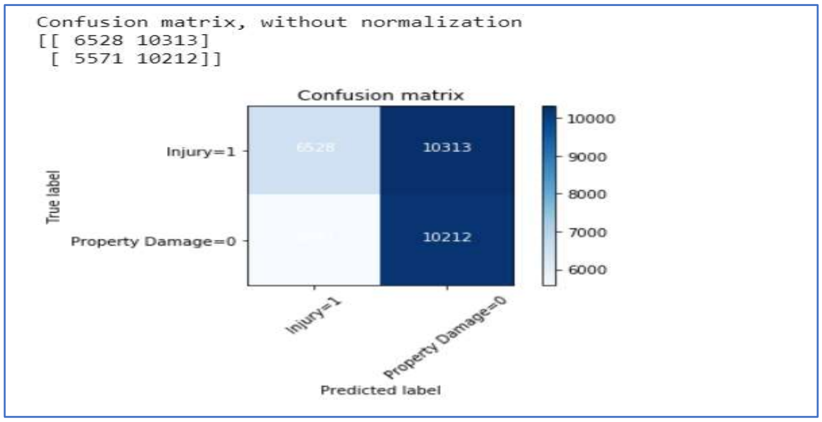
```
#Make Prediction:
yhatDT = DT.predict(X_test)

from sklearn.metrics import f1_score

#Check Accuracy
print('Accuracy score for Decision Tree = ', accuracy_score(yhatDT, y_test))
print("f1_score:",f1_score(y_test, yhat, average='weighted'))
print("Jaccard:",jaccard_score(y_test,yhatDT))
```

```
    Accuracy score for Decision Tree =  0.5131191760666993
    f1_score: 0.4856869229068246
    Jaccard: 0.2912725325718365
```



```
#Visualization
print('Confusion Matrix - Decision Tree')
print(pd.crosstab(y_test.ravel(), yhatDT.ravel(), rownames = ['True'], colnames = ['Predicted'], margins = True))

print(classification_report(yhatDT,y_test))
```

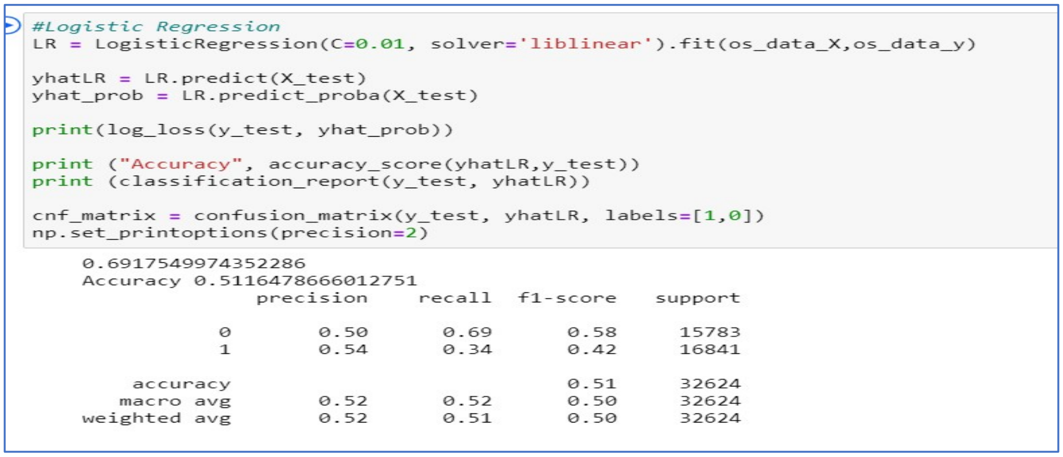
```
    Confusion Matrix - Decision Tree
    Predicted      0      1    All
    True
    0          10212   5571  15783
    1          10313   6528  16841
    All        20525  12099  32624
                  precision    recall  f1-score   support

               0       0.65      0.50      0.56     20525
               1       0.39      0.54      0.45     12099

        accuracy                           0.51     32624
       macro avg       0.52      0.52      0.51     32624
    weighted avg       0.55      0.51      0.52     32624
```



```
Confusion matrix, without normalization
[[ 6528 10313]
 [ 5571 10212]]
```

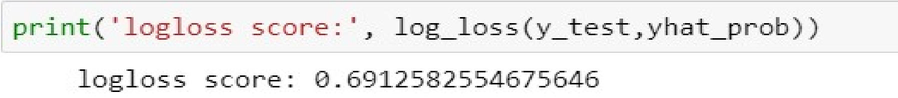
4.3 Logistic Regression

A supervised learning classification algorithm used to predict the probability of a target variable. The C used for regularization strength was 0.01 whereas the solver used was liblinear.



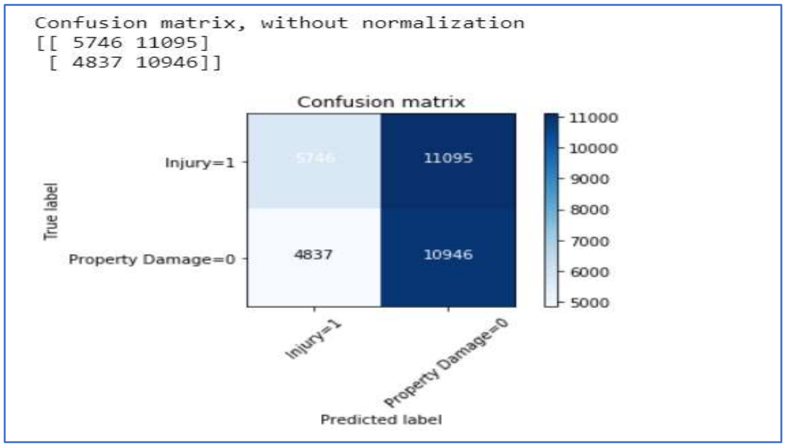
```
#Logistic Regression
LR = LogisticRegression(C=0.01, solver='liblinear').fit(os_data_X,os_data_y)

yhatLR = LR.predict(X_test)
yhat_prob = LR.predict_proba(X_test)

print(log_loss(y_test, yhat_prob))

print ("Accuracy", accuracy_score(yhatLR,y_test))
print (classification_report(y_test, yhatLR))

cnf_matrix = confusion_matrix(y_test, yhatLR, labels=[1,0])
np.set_printoptions(precision=2)
```

```
0.6917549974352286
Accuracy 0.5116478666012751
              precision    recall  f1-score   support

           0       0.50      0.69      0.58     15783
           1       0.54      0.34      0.42     16841

    accuracy                           0.51     32624
   macro avg       0.52      0.52      0.50     32624
weighted avg       0.52      0.51      0.50     32624
```



```
print('logloss score:', log_loss(y_test,yhat_prob))
```

```
logloss score: 0.6912582554675646
```

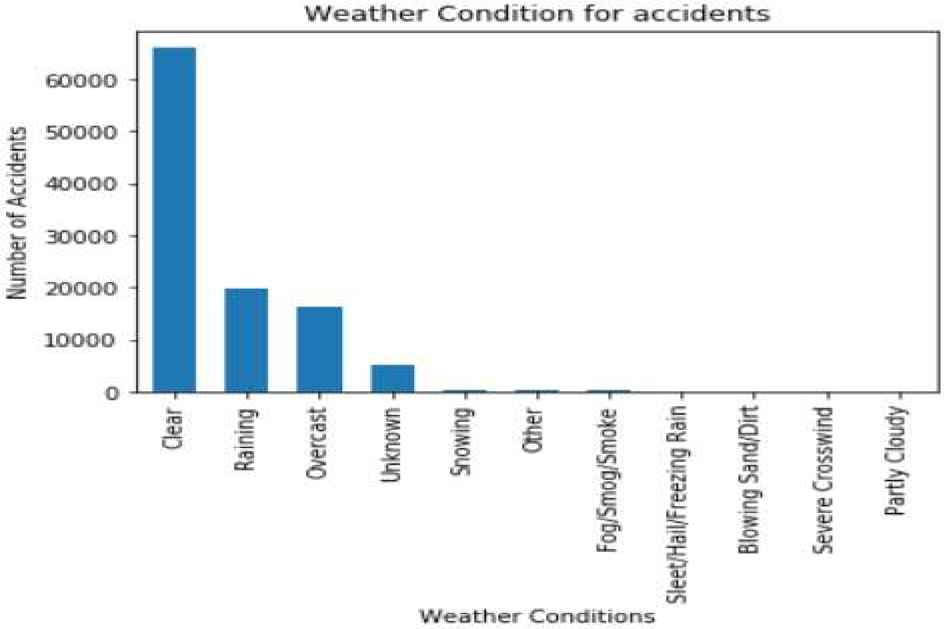


```
Confusion matrix, without normalization
[[ 5746 11095]
 [ 4837 10946]]
```

5 Discussion

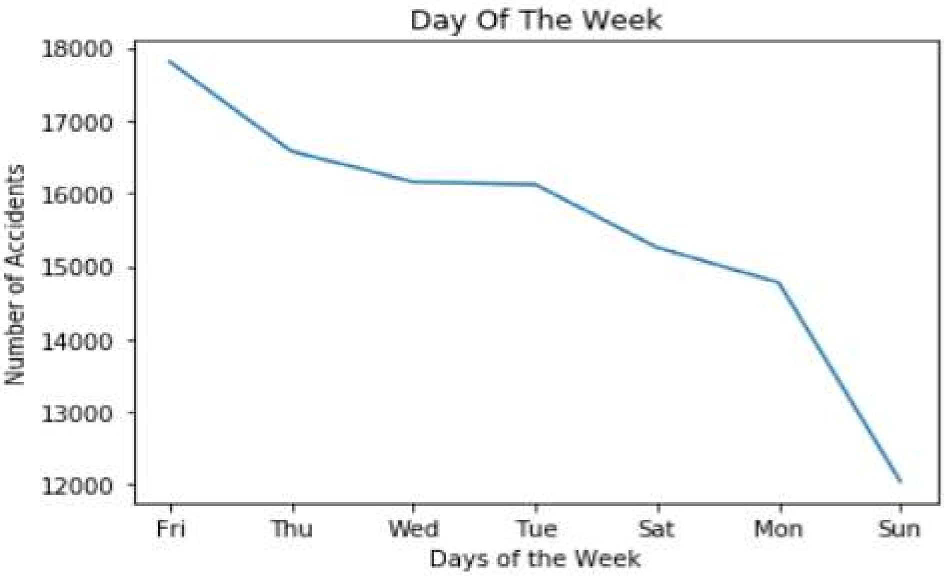
Based on the available data on collision from Seattle, provided for this capstone, it was observed that weather, road and light conditions have great impact on the accident rate. Plots shows the number of accidents that occurred when the weather was clear, more light on the road, etc.

5.1.1 Weather Conditions



It was equally observed that more accident occur on Fridays and the trend slowly decreases till Sunday.

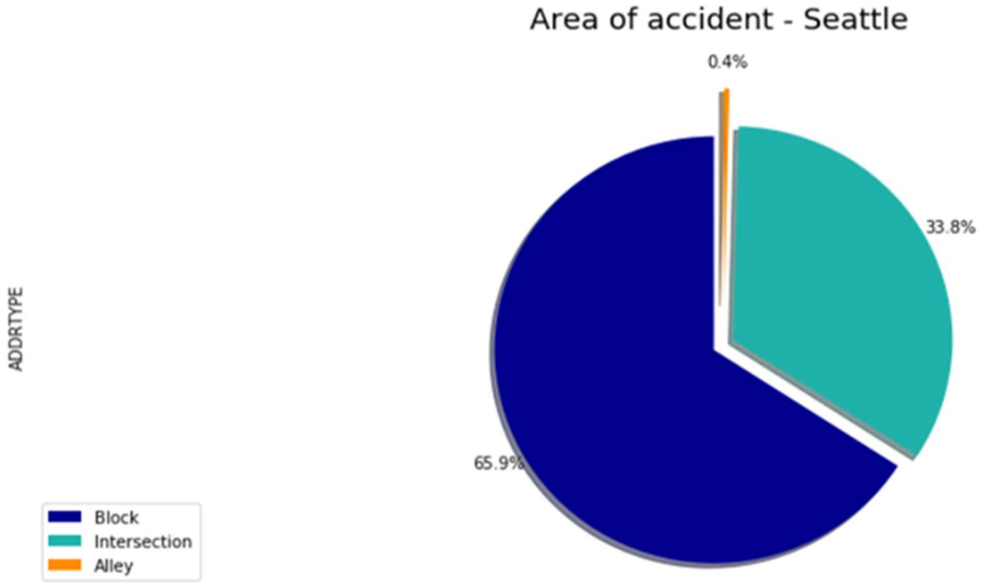
5.1.2 Days of the Week



This could be related to people's action/behavior, speeding or alcohol intake etc, which shows that people were overindulging more on Fridays – the beginning of weekend.

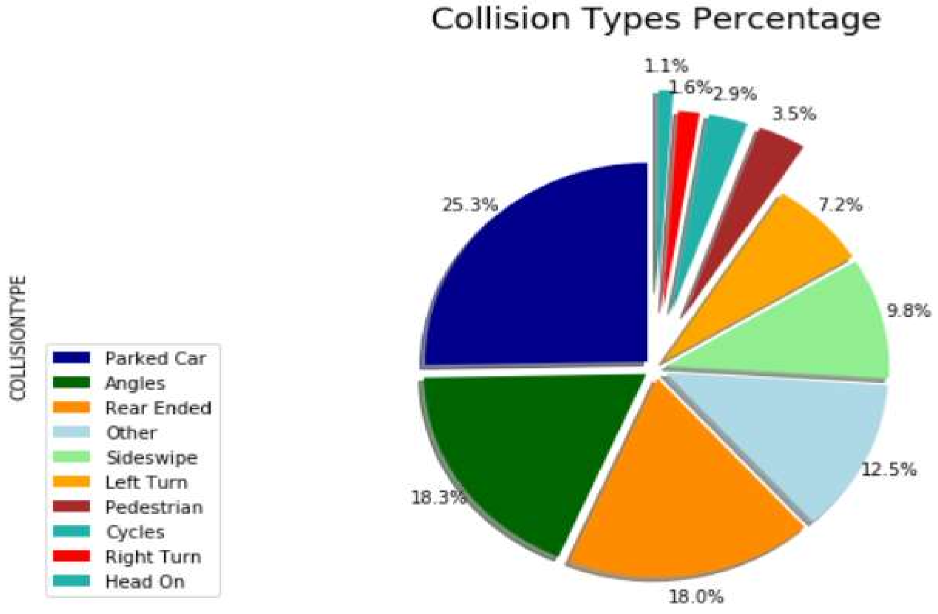
The area of accident – Seattle plot indicates that the percentage of accident was high at the block followed by at intersection and then lower at the alley.

5.1.3 Area of accident



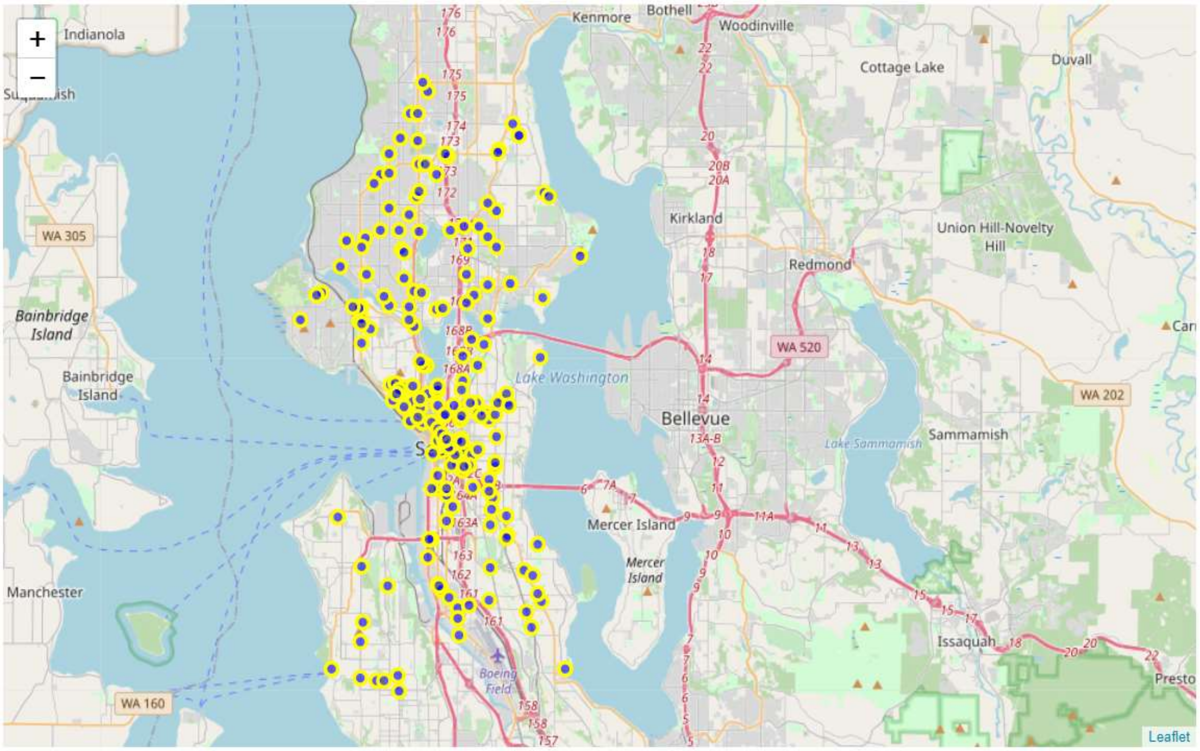
5.1.4 Collision Type Percentage

More accident occurred on parked cars with low physical injury and high property damage. Rear ended and Angles have more physical injury.



5.1.5 Seattle_map.add_child(1000 incidents)

A map of Seattle plotted with 1000 accidents incident picked.



6 Recommendations

The Seattle local government:

- Improve traffic policies, update public facilities such as streetlight, traffic signs and alert, etc.
- Warning and alert signs should include speed limits, road conditions and even weather when necessary.
- Barricades bad roads with potholes

Police:

- Enforcing the law and make sure traffic laws are followed
- Caution distracted drivers, remove ones under the influence of alcohol off the road.

Car insurance institutes:

- Check driver's license and records before issuing.

ALL DRIVERS:

- To follow/obey all traffic signs and laws, knowing that safety is very important and what their family or love ones will pass through if there was an accident.
- Assess to this information will enable them take extra precautions on the road under the given updates on light condition, road condition and weather, in order to avoid a severe accident.

7 Conclusion

Based on the available data on collision from Seattle provided for this capstone, we can conclude that weather, road and light conditions have a great impact on the accident rate that results in one of the two classes, property damage (class 1) or injury (class 2)

Other factors like speeding equally have an impact to the accident rate but not as the four used in the modeling and should not be overlooked.

When comparing all three models by their f1-scores, Precision and Recall, in terms of accuracy. k-Nearest Neighbor f1-score is low at 0.48 while Decision Tree and Logistic Regression with the average f1-score of 0.52 and 0.50 respectively are very close. The same goes for their Precision and Recall. It can be concluded that Decision Tree and Logistic Regression models can be used side by side for the best performance.