# CAR ACCIDENT SEVERITY in SEATTLE

## 1. INTRODUCTION

Car accidents in Seattle happen at all times, but if the main causes of accidents are determined, advance warning or mitigating methods can be performed. For example, certain intersections may be more susceptible to accidents due to heavy usage or the way they are constructed. As a result, better street lights can be added (only protected left and right turns) or traffic personnel can be used to direct the cars. If it is determined that accidents occur the majority of a time a driver is speeding, has a high blood alcohol level, or was not paying attention, the data can be used as evidence for enacting harsher laws and regulations. In addition, the data can be advertised to the public to show them the consequences of driving under these conditions. There are also uncontrollable factors such as weather and road conditions. If certain patterns are discovered to cause many accidents, local government can know when to send alerts to the public to drive more cautiously or even avoid the roads entirely.

## 2. TARGET AUDIENCE

2.1. The Seattle administration,

By targeting areas prone to areas to speeding accidents, interventions such as speed bumps, street light, traffic personnel etc. can be put in place to reduce accidents.

2.2. Rescue groups and emergency services in Seattle,

By having enough data on the crash one can predict the severity and therefore take action more quickly potentially saving lives.

2.3. Car Insurance Companies,

Areas where parked cars are prone to getting damaged. Owners in those localities may be asked to pay more premium on their car insurance.

## 3. DATA

The data comes from collision and accident reports in Seattle during the years 2004-present. It was collected by the Seattle Police Department and Traffic Records department. The data has 38 independent variables and 194,673 records and will be used to identify the key variables that cause accidents. We will use SEVERITYCODE as our dependent variable Y, and try different combinations of independent variables X to get the result. Since the observations are quite large, we may need to filter out the missing value and delete the unrelated columns first.

After analyzing the data set, I have decided to focus on only four features, severity, weather conditions, road conditions, and light conditions, among others.
Then we can select the factor which may have more impact on the accidents, such as address type, weather, road condition, and light condition.

## 4. DATA PREPARATION

### 4.1. Data Preprocessing

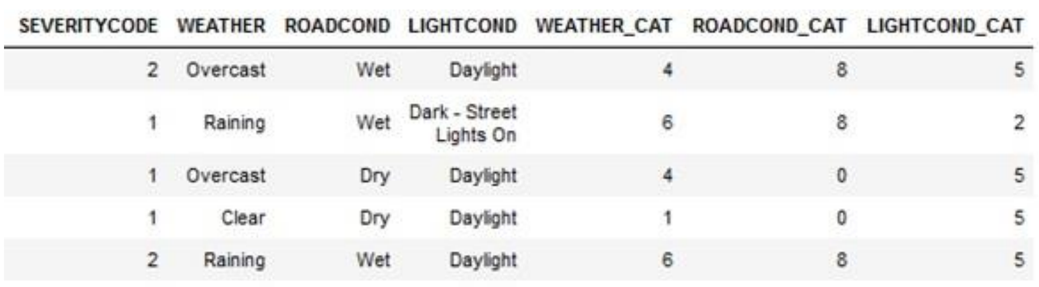
The dataset in the original form is not ready for data analysis. In order to prepare the data, first, we need to drop the non-relevant columns. In addition, most of the features are of object data types that need to be converted into numerical data types.

Our target variable will be “SEVERITYCODE” within data set accident from “0’ to “5”. Attributes used to weigh the severity of an accident are “WEATHER’,”ROADCOND” and “LIGHTCON”.

Severity codes are as follows :

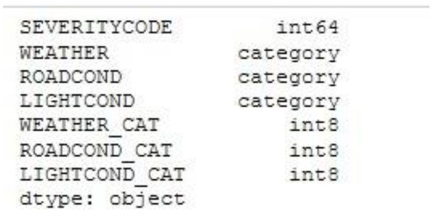
| | |
|---|---|
| 1 | Little to No Probability – Clear Condition |
| 2 | Very Low Probability – Chance or Property Damage |
| 3 | Low Probability – Chance of Injury |
| 4 | Mild Probability – Change of Serious Injury |
| 5 | High Probability – Chance of Fatality |

We must use label encoding to covert the features to our desired data type.



| SEVERITYCODE | WEATHER | ROADCOND | LIGHTCOND | WEATHER_CAT | ROADCOND_CAT | LIGHTCOND_CAT |
|---|---|---|---|---|---|---|
| 2 | Overcast | Wet | Daylight | 4 | 8 | 5 |
| 1 | Raining | Wet | Dark - Street Lights On | 6 | 8 | 2 |
| 1 | Overcast | Dry | Daylight | 4 | 0 | 5 |
| 1 | Clear | Dry | Daylight | 1 | 0 | 5 |
| 2 | Raining | Wet | Daylight | 6 | 8 | 5 |

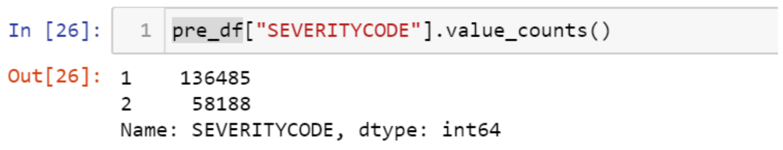
Now let's check the data types of the new columns in our dataframe. Moving forward, we will only use the new columns for our analysis.



```
SEVERITYCODE        int64
WEATHER          category
ROADCOND         category
LIGHTCOND        category
WEATHER_CAT          int8
ROADCOND_CAT         int8
LIGHTCOND_CAT        int8
dtype: object
```

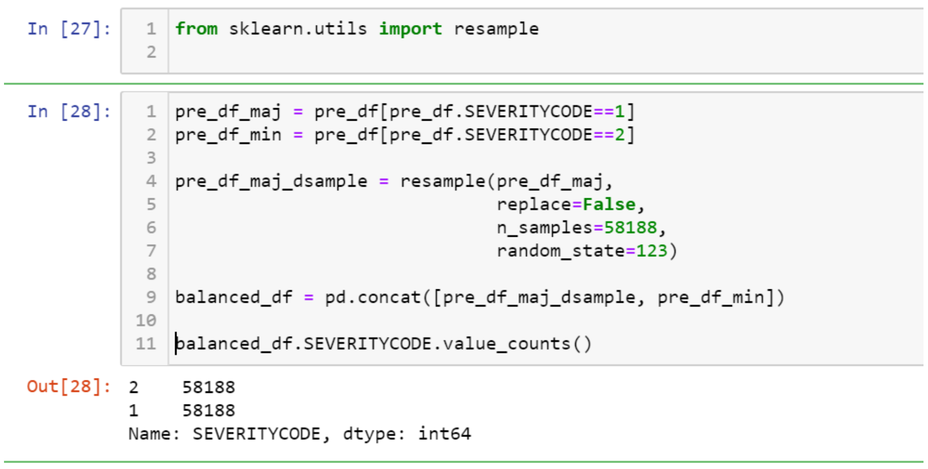
### 4.2. Balancing the Dataset

To get a good understanding of the dataset, I have checked different values in the features. The results show, the target feature is imbalance, so we use a simple statistical technique to balance it.



```
In [26]: pre_df["SEVERITYCODE"].value_counts()

Out[26]: 1    136485
         2     58188
         Name: SEVERITYCODE, dtype: int64
```

As you can see, the number of rows in class 1 is almost three times bigger than the number of rows in class 2. After down sampling the majority class



```
In [27]: from sklearn.utils import resample

In [28]: pre_df_maj = pre_df[pre_df.SEVERITYCODE==1]
         pre_df_min = pre_df[pre_df.SEVERITYCODE==2]

         pre_df_maj_dsample = resample(pre_df_maj,
                                       replace=False,
                                       n_samples=58188,
                                       random_state=123)

         balanced_df = pd.concat([pre_df_maj_dsample, pre_df_min])

         balanced_df.SEVERITYCODE.value_counts()

Out[28]: 2    58188
         1    58188
         Name: SEVERITYCODE, dtype: int64
```

## 5. METHODOLOGY

### 5.1. EXPLORATORY ANALYSIS

Considering that the feature set and the target variable are categorical variables with the likes of weather, road condition and light condition being an above level 2 categorical variables whose values are limited and usually based on a particular finite group whose correlation might depict a different image then what it actually is. Generally, considering the effect of these variables in car accidents are important hence these variables were selected.

### 5.2. MACHINE LEARNING MODEL SELECTION

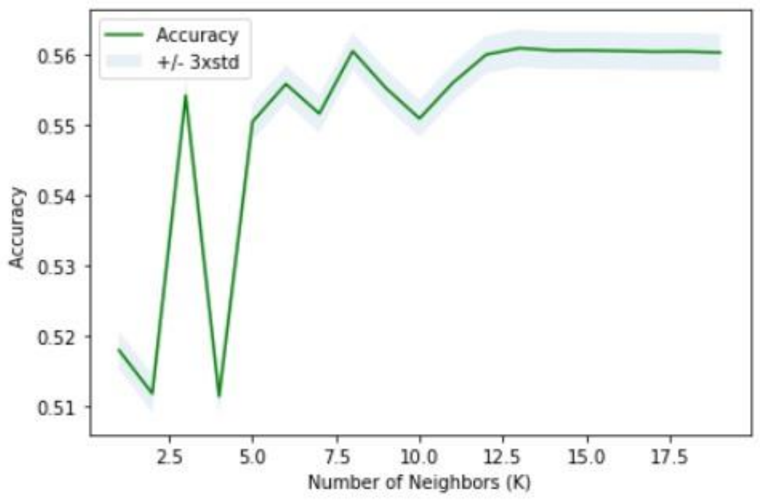
The machine learning models used are k-Nearest Neighbor, Decision Tree Analysis and Logistic Regression. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (based on distance). The Decision Tree Analysis breaks down a data set into smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. The reason why Decision Tree Analysis, Logistic Regression and k-Nearest Neighbor classification methods were chosen is because the Support Vector Machine (SVM) model is inaccurate for large data sets, while this data set has more than 180,000 rows filled with data.

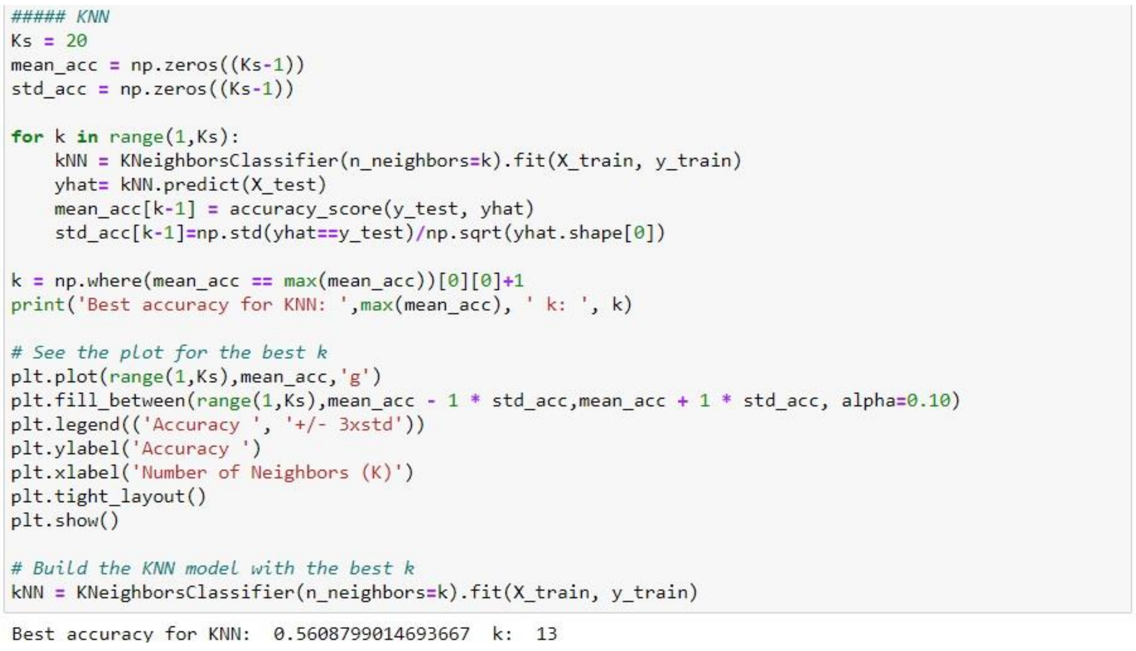
## 6. RESULT

### 6.1. K-Nearest Neighbor (KNN)

K-NN will help us predict the severity code of an outcome by finding the most similar to data point within k distance. k-Nearest Neighbor classifier was used from the scikit-learn library to run the k-Nearest Neighbor machine learning classifier on the Car Accident Severity data. The best K, as shown below, for the model where the highest elbow bend exists is at 4.

```
Best accuracy for KNN:  0.5608799014693667  k:  13
```



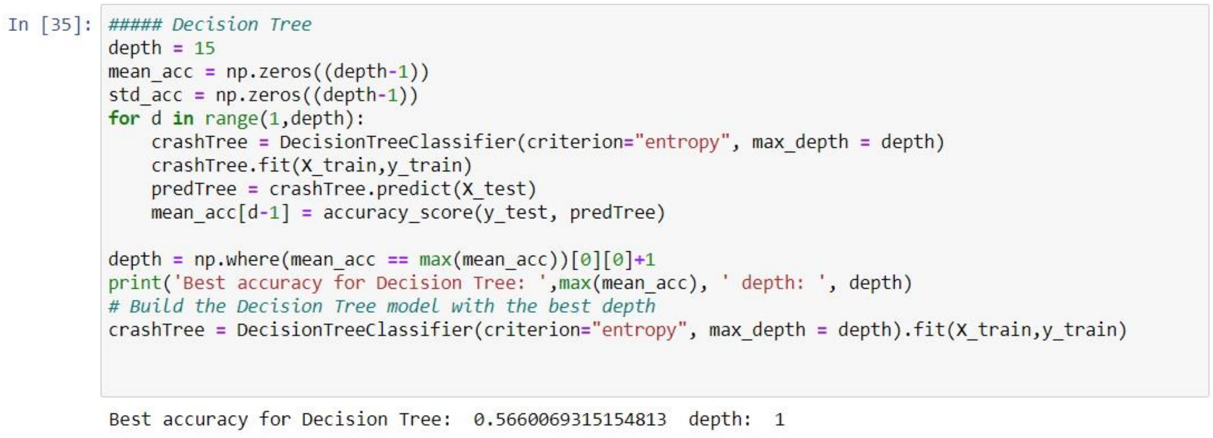


```
##### KNN
Ks = 20
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))

for k in range(1,Ks):
    kNN = KNeighborsClassifier(n_neighbors=k).fit(X_train, y_train)
    yhat= kNN.predict(X_test)
    mean_acc[k-1] = accuracy_score(y_test, yhat)
    std_acc[k-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

k = np.where(mean_acc == max(mean_acc))[0][0]+1
print('Best accuracy for KNN: ',max(mean_acc), ' k: ', k)

# See the plot for the best k
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Neighbors (K)')
plt.tight_layout()
plt.show()

# Build the KNN model with the best k
kNN = KNeighborsClassifier(n_neighbors=k).fit(X_train, y_train)
```

```
Best accuracy for KNN:  0.5608799014693667  k:  13
```

### 6.2. Decision Tree

A decision tree model gives us a layout of all possible outcomes so we can fully analyze the concequences of a decision. It context, the decision tree observes all possible outcomes of different weather conditions. Decision Tree Classifier from the scikit-learn library was used to run the Decision Tree Classification model on the Car Accident Severity data. The criterion chosen for the classifier was 'entropy' and the max depth was '6'. The post-SMOTE balanced data was used to predict and fit the Decision Tree Classifier.

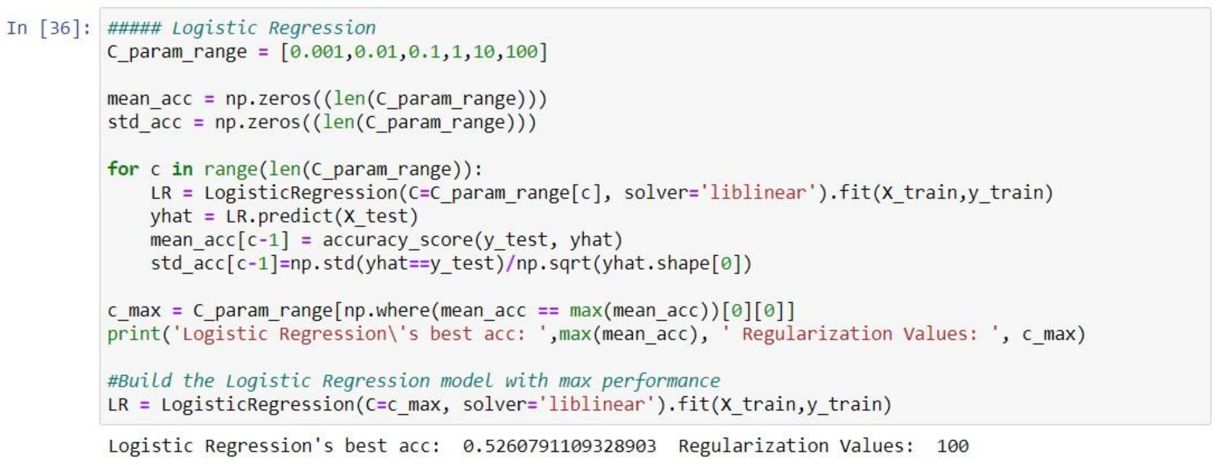


```
In [35]: ##### Decision Tree
depth = 15
mean_acc = np.zeros((depth-1))
std_acc = np.zeros((depth-1))
for d in range(1,depth):
    crashTree = DecisionTreeClassifier(criterion="entropy", max_depth = depth)
    crashTree.fit(X_train,y_train)
    predTree = crashTree.predict(X_test)
    mean_acc[d-1] = accuracy_score(y_test, predTree)

depth = np.where(mean_acc == max(mean_acc))[0][0]+1
print('Best accuracy for Decision Tree: ',max(mean_acc), ' depth: ', depth)
# Build the Decision Tree model with the best depth
crashTree = DecisionTreeClassifier(criterion="entropy", max_depth = depth).fit(X_train,y_train)
```

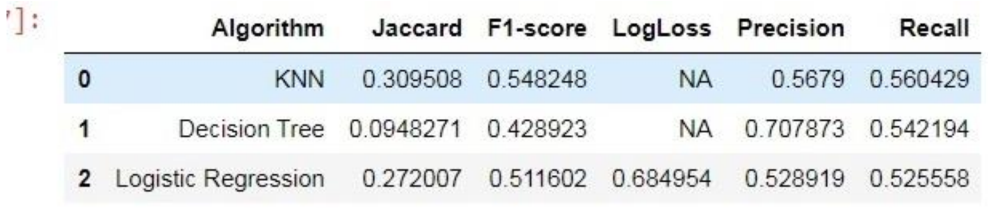
```
Best accuracy for Decision Tree:  0.5660069315154813  depth:  1
```

### 6.3. Logistic Regression

Logistic Regression from the scikit-learn library was used to run the Logistic Regression Classification model on the Car Accident Severity data. The C used for regularization strength was ‘0.01’ whereas the solver used was ‘liblinear’. Because our dataset only provides us with two severity code outcomes, our model will only predict one of those two classes. This makes our data binary, which is perfect to use with logistic regression.



```
In [36]: ##### Logistic Regression
         C_param_range = [0.001,0.01,0.1,1,10,100]

         mean_acc = np.zeros((len(C_param_range)))
         std_acc = np.zeros((len(C_param_range)))

         for c in range(len(C_param_range)):
             LR = LogisticRegression(C=C_param_range[c], solver='liblinear').fit(X_train,y_train)
             yhat = LR.predict(X_test)
             mean_acc[c-1] = accuracy_score(y_test, yhat)
             std_acc[c-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

         c_max = C_param_range[np.where(mean_acc == max(mean_acc))[0][0]]
         print('Logistic Regression\'s best acc: ',max(mean_acc), ' Regularization Values: ', c_max)

         #Build the Logistic Regression model with max performance
         LR = LogisticRegression(C=c_max, solver='liblinear').fit(X_train,y_train)

         Logistic Regression's best acc:  0.5260791109328903  Regularization Values:  100
```



']:

| | Algorithm | Jaccard | F1-score | LogLoss | Precision | Recall |
|---|---|---|---|---|---|---|
| 0 | KNN | 0.309508 | 0.548248 | NA | 0.5679 | 0.560429 |
| 1 | Decision Tree | 0.0948271 | 0.428923 | NA | 0.707873 | 0.542194 |
| 2 | Logistic Regression | 0.272007 | 0.511602 | 0.684954 | 0.528919 | 0.525558 |

## 7. DISCUSSION

We had categorical data that was of type 'object'. This is not a data type that we could have fed through an algoritim, so label encoding was used to created new classes that were of type int8; a numerical data type. After solving that issue we were presented with another - imbalanced data. As mentioned earlier, class 1 was nearly three times larger than class 2. The solution to this was downsampling the majority class with sklearn's resample tool. We downsampled to match the minority class exactly with 58188 values each.

Once we analyzed and cleaned the data, it was then fed through three ML models; K-Nearest Neighbor, Decision Tree and Logistic Regression. Although the first two are ideal for this project, logistic regression made most sense because of its binary nature.
Evaluation metrics used to test the accuracy of our models were jaccard index, f-1 score and logloss for logistic regression. Choosing different k, max depth and hyparameter C values helped to improve our accuracy to be the best possible.

## 8. CONCLUSION

When comparing these scores, we can see that the f1-score is highest for k-Nearest Neighbor at 0.75. However, later when we compare the precision and recall for each of the model, we can see that the k-Nearest Neighbor model performs poorly in the precision of 1 at 0.08. The variance is too high for the model to be selected as a viable option. When looking at the other two models, we can see that the Decision Tree has a more balanced precision for 0 and 1. Whereas, the Logistic Regression is more balanced when it comes to recall of 0 and 1. Furthermore, the average f1-score of the two models are very close but for the Logistic Regression it is higher by 0.04. It can be concluded that the both the models can be used side by side for the best performance.

In retrospect, when comparing these scores to the benchmarks within the industry, it can be seen that they perform well but not as good as the benchmarks. These models could have performed better if a few more things were present and possible.

- A balanced dataset for the target variable
- More instances recorded of all the accidents taken place in Seattle, Washington
- Less missing values within the dataset for variables such as Speeding and Under the influence
- More factors, such as precautionary measures taken when driving, etc.

Based on historical data from weather conditions pointing to certain classes, we can conclude that particular weather conditions have a somewhat impact on whether or not travel could result in property damage (class 1) or injury (class 2).

## 9. RECOMMENDATION

After assessing the data and the output of the Machine Learning models, a few recommendations can be made for the stakeholders. The developmental body for Seattle city can assess how much of these accidents have occurred in a place where road or light conditions were not ideal for that specific area and could launch development projects for those areas where most severe accidents take place in order to minimize the effects of these two factors. Whereas, the car drivers could also use this data to assess when to take extra precautions on the road under the given circumstances of light condition, road condition and weather, in order to avoid a severe accident, if any.