Applied Data Science Capstone - Predicting Car Accident Severity based on Environmental Conditions

Introduction:

All major cities face numerous traffic accidents each year, with severity ranging from minor vehicle damage to multiple fatalities. These accidents can happen in a variety of situations and involve a variable number of vehicles, pedestrians, cyclists and others. They also happen under a variety of environmental conditions, such as weather, road conditions and lighting. In this project, we look specifically at traffic accident data from the Seattle area, and are interested in predicting the likely severity of accidents based on environmental conditions.

This could provide multiple benefits, such as: providing the capability to inform drivers of conditions correlated with higher accident severity to raise awareness and encourage safer driving; and allowing emergency responders to plan ahead and be better prepared for severe accidents based on expected environmental conditions. Both could help reduce the severity of any accidents that happen and/or the damage resulting from said accidents, thus reducing the toll in both human life and on vital municipal resources.

Data:

The data set on historical accidents in the Seattle area consists of 37 variables and 194,673 entries covering: date and time, accident severity, collision type, number of people/pedestrians/bicycles/vehicles involved, number of injuries/serious injuries/fatalities, weather conditions, road conditions, light conditions, location, location details, whether speeding/inattention/drugs/alcohol were involved, and other descriptive characteristics of the accident.

Since we are interested in predicting the likely severity of accidents based on environmental conditions, we will limit our independent variables to environmental variables beyond the control of the driver and set our dependent variable to the severity of the accidents which occur under the given environment.

Thus we will use WEATHER, ROADCOND and LIGHTCOND as our independent variables to predict our independent variable, SEVERITYCODE.

The initial data set contains many variables that we do not need, and the variables we do need are mostly in object rather than numerical format making them unsuitable for our needs. To start, we first limit the dataset to our desired variables and check to see what categorical values are present and in what mix as seen in the next page::

Choosing relevant variables and identifying values present:

Narrow dataset to relevant variables/entries

In [5]:
```
#selecting variables of interest

seattle_select_df = seattle_df[['SEVERITYCODE', 'WEATHER', 'ROADCOND', 'LIGHTCOND']]
seattle_select_df.shape
```

Out[5]:
```
(194673, 4)
```

In [6]:
```
#check values of columns and count for each value

for column in seattle_select_df.columns.values.tolist():
    print(column)
    print (seattle_select_df[column].value_counts())
    print("")
```

```
SEVERITYCODE
1    136485
2     58188
Name: SEVERITYCODE, dtype: int64

WEATHER
Clear                       111135
Raining                      33145
Overcast                     27714
Unknown                      15091
Snowing                        907
Other                          832
Fog/Smog/Smoke                 569
Sleet/Hail/Freezing Rain       113
Blowing Sand/Dirt               56
Severe Crosswind                25
Partly Cloudy                    5
Name: WEATHER, dtype: int64

ROADCOND
Dry               124510
Wet                47474
Unknown            15078
Ice                 1209
Snow/Slush          1004
Other                132
Standing Water       115
Sand/Mud/Dirt         75
Oil                   64
Name: ROADCOND, dtype: int64

LIGHTCOND
Daylight                    116137
Dark - Street Lights On      48507
Unknown                      13473
Dusk                          5902
Dawn                          2502
Dark - No Street Lights       1537
Dark - Street Lights Off      1199
Other                          235
Dark - Unknown Lighting         11
Name: LIGHTCOND, dtype: int64
```

After removing entries with missing values, with values of extremely low count, and values that are unlikely to be seen in much of Seattle, we are left with 169,643 entries as seen below:

In [8]:
```
#cleaned dataset before encoding

seattle_cleaned_df
```

Out[8]:

	SEVERITYCODE	WEATHER	ROADCOND	LIGHTCOND
0	2	Overcast	Wet	Daylight
1	1	Raining	Wet	Dark - Street Lights On
2	1	Overcast	Dry	Daylight
3	1	Clear	Dry	Daylight
4	2	Raining	Wet	Daylight
...	...	...	...	...
169638	2	Clear	Dry	Daylight
169639	1	Raining	Wet	Daylight
169640	2	Clear	Dry	Daylight
169641	2	Clear	Dry	Dusk
169642	1	Clear	Wet	Daylight

169643 rows × 4 columns

We then proceed to encode the categorical data to a numerical format, nominally ranked by potential impact on driving ability and normalize the data for modeling purposes to avoid undesired impacts from varying scales to obtain the following dataset:

In [11]:
```
#normalize data

X = preprocessing.StandardScaler().fit(X).transform(X.astype(float))
X
```

Out[11]:
```
array([[ 0.4845433 ,  1.3738043 , -0.69034376],
       [ 1.63627901,  1.3738043 ,  1.43852641],
       [ 0.4845433 , -0.59346323, -0.69034376],
       ...,
       [-0.66719242, -0.59346323, -0.69034376],
       [-0.66719242, -0.59346323,  0.72890302],
       [-0.66719242,  1.3738043 , -0.69034376]])
```

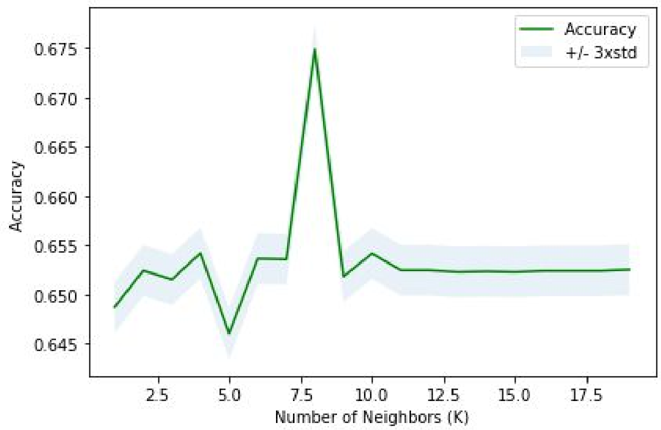
Next, we split the dataset 80:20 into training and testing sets for our models.

Methodology:

As we are predicting severity of accidents based on environmental factors and all data is labeled categorical data, we will be using clustering models for our prediction.

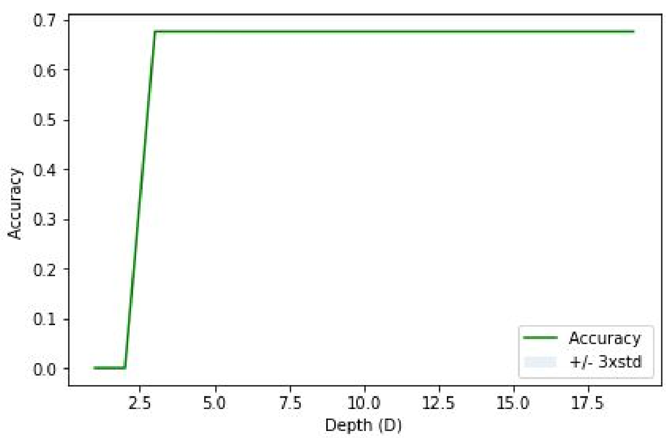
In this case we chose to utilize K Nearest Neighbor (KNN), Decision Tree and Logistic Regression as our potential clustering model candidates.

For KNN, we test for the ideal number of neighbors k (up to 20) and find that the k with the highest accuracy is 8 as seen below and choose that for our KNN model.




```
The best accuracy was 0.6748798962539421 with k= 8
```

For our Decision Tree model, we test different depths d (up to 20) and find that there is no increase in accuracy beyond 3, choosing that for our Decision Tree model:




```
The best accuracy was 0.6755577824280115 with d= 3
```

Lastly, for our Logistic Regression model, we found the XX method to have the highest accuracy:

Results:

In this section, we tested the different models to ascertain Jaccard Similarity Score, F1-Score and Log Loss (Logistic Regression only).

By doing so, we found that the Decision Tree and Logistic Regression models likely provide the best accuracy, though the Logistic Regression model also provides the benefit of a probability that an accident will be severe.

Though the accuracy could be higher we do show that our model is able to predict the severity of accidents based on environmental conditions with an accuracy of nearly 70% and reasonable Jaccard Similarity and F1-Scores as see below:

	Algorithm	Jaccard	F1-score	Log Loss
0	KNN	0.674544	0.546306	NA
1	Decision Tree	0.675558	0.544748	NA
2	Logistic Regression	0.675558	0.544748	0.62928

Discussion:

In our analysis and modeling, we only utilized three environmental conditions, namely weather, road conditions and lighting conditions, all encoded with a fairly unscientific ranking of potential impact on driver ability to avoid accidents.

This leaves much room for improvement, with potentially a more accurate, non-integer scale for the three environmental conditions included, as well as other factors that have not been taken into account such as time of day, time of year, area, and so on.

Conclusion:

We can see that it is possible to predict the potential severity of accidents occuring based on environmental conditions, thus opening the possibility of notifying drivers of more dangerous conditions to potentially encourage safer driving and allowing emergency services and other municipal resources to be better deployed in accordance with forecasted accident severity. Both have the potential to both reduce the number of severe accidents and potentially the toll in human lives and economic costs.