Predicting Accident Severity in Seattle

- Kapil Warghane

1.1 Introduction

Seattle is a seaport city on the West Coast of the United States. According to U.S. Census data released in 2019, the Seattle metropolitan area's population stands at 3.98 million, making it the 15th-largest in the United States. In July 2013, Seattle was the fastest-growing major city in the United States and remained in the top five in May 2015 with an annual growth rate of 2.1%. In July 2016, Seattle was again the fastest-growing major U.S. city, with a 3.1% annual growth rate.

With this amazing growth of the Seattle, there is still a room for improvement in the Road Accidents where people are losing the life's or getting injured in traffic collisions every year. Realizing traffic accidents as a preventable problem. Different policies and measures are implemented to reduce this problem. These include enforcement, education, training and engineering improvements. To improve road safety and combat life-threatening crashes we will be working on a data set that will help us Predicting Car Accident Severity across the Roads making it safer for the travellers.

1.2 Business Problem

In the world there are around tens of thousands of people are killed or injured in traffic collisions each year. While Seattle has seen a 30 percent decline in traffic fatalities over the last decade, traffic collisions are still a leading cause of death for Seattle residents age 5-24. Older adults are also disproportionately affected, so this trend could grow as the population ages. With the roads accident's becoming a major issue for the city around various ages and there are environmental factors and conditions of road.

Our main question here is How we might support Seattle to apply data science to reduce traffic fatalities and injuries across the roads in city? And make your next Seattle road trip or journey more accident free.

1.3 Interest

Obviously, The Government, Administration and People will be really happy considering the roads will be safer for people to travel around the city. And other cities along the world can also learn from the results. Ultimately reducing the death rate along accidents around the World.

2.1 Data Selection

Data selection is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data. Data selection precedes the actual practice of data collection.

This definition distinguishes data selection from selective data reporting (selectively excluding data that is not supportive of a research hypothesis) and interactive/active data selection (using collected data for monitoring activities/events, or conducting secondary data analyses). The process of selecting suitable data for a research project can impact data integrity.

The primary objective of data selection is the determination of appropriate data type and sources that allow us predicting the severity of accident. This determination is often discipline-specific and is primarily driven by the nature of the investigation, existing literature, and accessibility to necessary data sources.

The data we have consists of 194673 rows × 38 columns which is collected from previous accidents and events occurred in past across roads for the study. The set has various attributes some them describing environmental factors (Whether, Road Condition and Light condition) which can highly influence the road accidents.

For example: Consider you are visiting your friends place on the rainy day who live in a country side and while travelling you find the road is wet and making hard for tyres to have a proper grip along the road.

So same things are also taken into consideration while selecting the independent variables for the model.

2.2 Data Cleaning

Since our data consists of 194673 rows × 38 columns it's very important that we calculate the missing values and check for columns with most null values. Considering we will be making our model for predicting the accident severity. And most of values associated with the description of accidents, and secondary information we can remove those columns. With few columns describing the departmental code's for accident those also been removed.

2.3 Selecting Feature Set

After Cleaning the data there are 12 Columns selected based on there importance for calculating accident severity and less null values along the columns. But still there are new values missing along the columns and rows, hence we first calculate the frequencies along the columns and then we have replaced those values in the columns making it more suitable for use and predications.

```
In [6]: data_collisions.isnull().sum()
```

```
Out[6]: SEVERITYCODE           0
        X                  5334
        Y                  5334
        OBJECTID              0
        INCKEY                0
        COLDETKEY             0
        REPORTNO              0
        STATUS                0
        ADDRTYPE           1926
        INTKEY           129603
        LOCATION           2677
        EXCEPTRSNCODE    109862
        EXCEPTRSNDESC    189035
        SEVERITYCODE.1        0
        SEVERITYDESC          0
        COLLISIONTYPE      4904
        PERSONCOUNT           0
        PEDCOUNT              0
        PEDCYLCOUNT           0
        VEHCOUNT              0
        INCDATE               0
        INCDTTM               0
        JUNCTIONTYPE       6329
        SDOT_COLCODE          0
        SDOT_COLDESC          0
        INATTENTIONIND    164868
        UNDERINFL          4884
        WEATHER            5081
        ROADCOND           5012
        LIGHTCOND          5170
        PEDROWNOTGRNT    190006
        SDOTCOLNUM        79737
        SPEEDING         185340
        ST_COLCODE           18
        ST_COLDESC         4904
        SEGLANEKEY            0
        CROSSWALKKEY          0
        HITPARKEDCAR          0
        dtype: int64
```

Figure 1 – Calculating Null values throughout Dataset

After analysing the null values on dataset, we can't replace the values with high missing data, as it may be over dependent of values.

Below are the columns selected.

[[SEVERITYCODE, ADDRTYPE, COLLISIONTYPE, PERSONCOUNT, PEDCOUNT, PEDCYLCOUNT, VEHCOUNT, JUNCTIONTYPE, UNDERINFL, WEATHER, ROADCOND, LIGHTCOND, ADDRTYPE1, COLLISIONTYPE1, JUNCTIONTYPE1, UNDERINFL1, WEATHER1, ROADCOND1, LIGHTCOND1]]

3. Exploratory Data Analysis

From Figure 1 we can find the null values and make a decision regarding selection of columns that can be used for predicting accident severity. Once the columns selected for making our final dataset, the next step will be replacing the null values.

Here we can replace the values by taking the most frequent values along the same columns respectively. Hence, we use. idxmax() function to calculate the maximum number of occurrences of value under a specific column. To replace the null values we can simply use the replace function of pandas to replace the values along the dataset.

Replace the missing of columns values by the most frequent

```
In [25]: data_collisions['ADDRTYPE'].value_counts().idxmax()
         data_collisions["ADDRTYPE"].replace(np.nan, "Block", inplace=True)

In [26]: data_collisions['COLLISIONTYPE'].value_counts().idxmax()
         data_collisions["COLLISIONTYPE"].replace(np.nan, "Parked Car", inplace=True)

In [27]: data_collisions['JUNCTIONTYPE'].value_counts().idxmax()
         data_collisions["JUNCTIONTYPE"].replace(np.nan, "Mid-Block (not related to inter

In [28]: data_collisions['UNDERINFL'].value_counts().idxmax()
         data_collisions["UNDERINFL"].replace(np.nan, "N", inplace=True)

In [29]: data_collisions['WEATHER'].value_counts().idxmax()
         data_collisions["WEATHER"].replace(np.nan, "Clear", inplace=True)

In [30]: data_collisions['ROADCOND'].value_counts().idxmax()
         data_collisions['ROADCOND'].replace(np.nan,'Dry',inplace=True)

In [31]: data_collisions['LIGHTCOND'].value_counts().idxmax()
         data_collisions['LIGHTCOND'].replace(np.nan, 'Daylight', inplace=True)
```

Figure 2 Replacing values along the dataset

4.1 Methodology section

Once we get our final data set, we can check for the dtypes and change our categorical values to int64 since maximum values are of int64. For this we use the factorize function of pandas. Note here we will get the values under new columns so we represented the new rows with adding a suffix 1 on the name. for example, ADDRTYPE becomes ADDRTYPE1, similarly other columns are converted.

Converting the dtypes from object to int64

```
In [32]: data_collisions['ADDRTYPE1'] = pd.factorize(data_collisions.ADDRTYPE)[0]
         data_collisions['COLLISIONTYPE1'] = pd.factorize(data_collisions.COLLISIONTYPE)[
         data_collisions['JUNCTIONTYPE1'] = pd.factorize(data_collisions.JUNCTIONTYPE)[0]
         data_collisions['UNDERINFL1'] = pd.factorize(data_collisions.UNDERINFL)[0]
         data_collisions['WEATHER1'] = pd.factorize(data_collisions.WEATHER)[0]
         data_collisions['ROADCOND1'] = pd.factorize(data_collisions.ROADCOND)[0]
         data_collisions['LIGHTCOND1'] = pd.factorize(data_collisions.LIGHTCOND)[0]
```

Figure 3 Converting data types from object to int64

Now we still have values in categorical state so we will drop those values using drop function.

4.2 Data Visualization

Since there are values defined of dependent variable i.e. SEVERITYCODE = [1,2], So if we try to plot scatter plot with independent variable PERSONCOUNT we will get.

```
In [43]: plt.scatter(data_collisions.PERSONCOUNT, data_collisions.SEVERITYCODE,  color='
         plt.xlabel("PERSONCOUNT")
         plt.ylabel("SEVERITYCODE")
         plt.show()
```

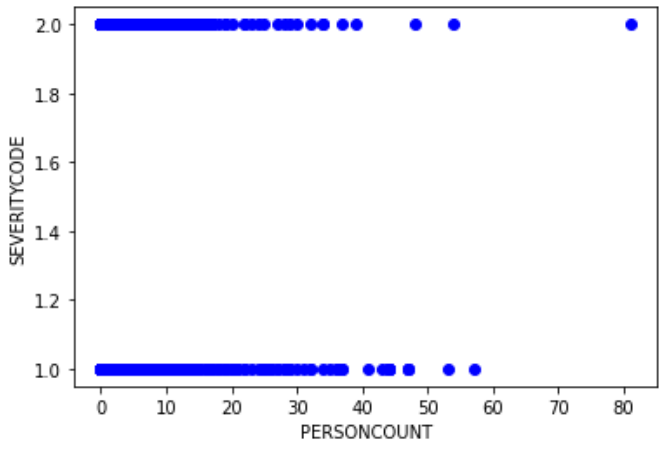


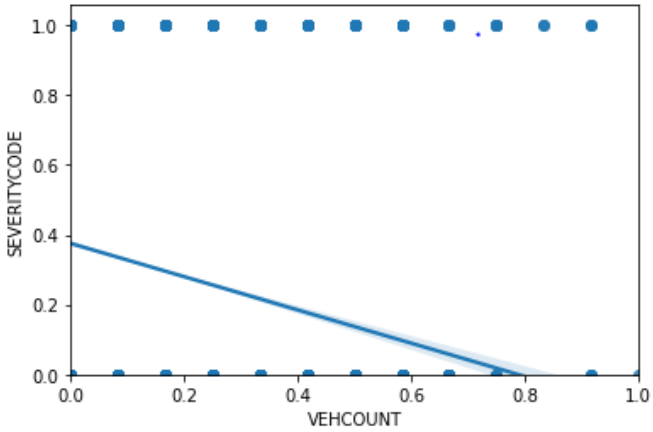
Figure – Data Visualization

To predict the correct RMSE we need to normalize our date set hence we use Min Max Scaler Function of pandas to normalize the dataset and store it new data frame name i.e. Seattle

Plotting Linear Regression Plots with different Independent Variables.

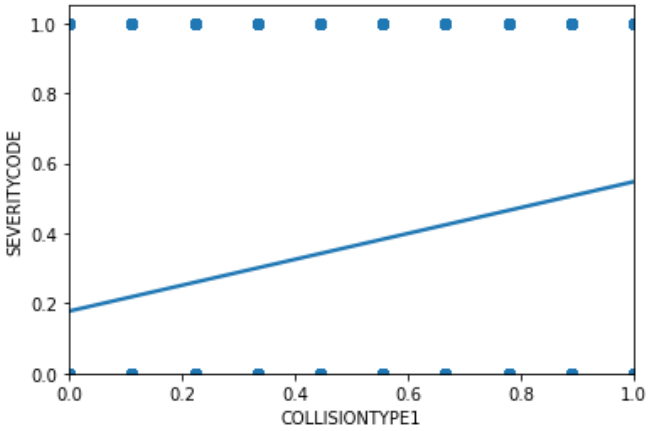
In [55]:
```
sns.regplot(x='VEHCOUNT', y='SEVERITYCODE', data=Collision)
plt.ylim(0,)
```

Out[55]: (0.0, 1.0568659907944749)



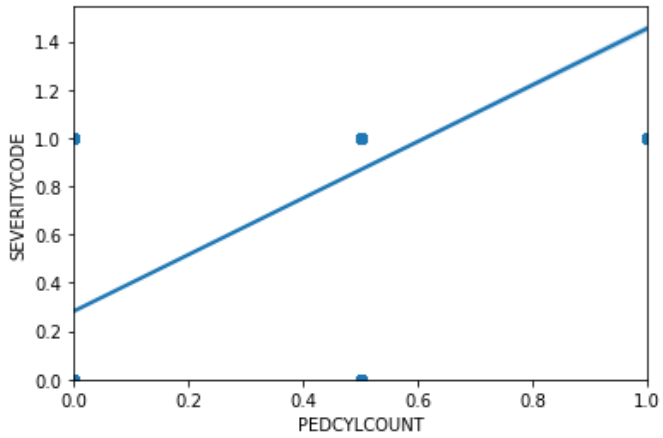
In [56]:
```
sns.regplot(x='COLLISIONTYPE1', y='SEVERITYCODE', data=Collision)
plt.ylim(0,)
```

Out[56]: (0.0, 1.05)



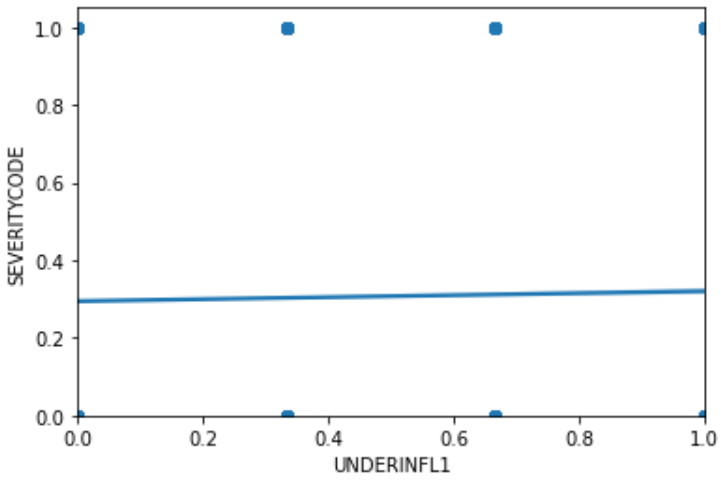
```
In [59]: sns.regplot(x='PEDCYLCOUNT', y='SEVERITYCODE', data=Collision)
         plt.ylim(0,)

Out[59]: (0.0, 1.5439162308849965)
```




```
In [61]: sns.regplot(x='UNDERINFL1', y='SEVERITYCODE', data=Collision)
         plt.ylim(0,)

Out[61]: (0.0, 1.05)
```



5 Results

Since we have more than one independent variable, we can check this problem by using 3 methods

1) Multiple Linear Regression Model
2) Polynomial Linear Regression Model
3) K Nearest Neighbour (KNN) Classification Model

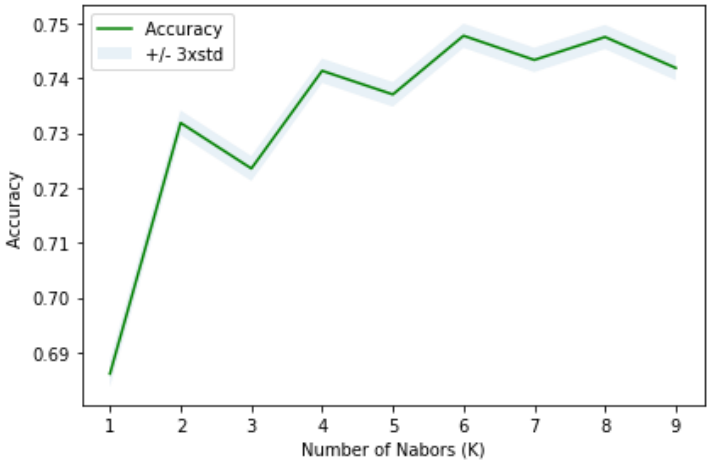
For Multiple Linear Regression we get the RMSE of 0.4179

For Polynomial Linear Regression we get RMSE of 0.4094

For KNN classification model we get Train set Accuracy: 0.7541 and Test set Accuracy: 0.7413

The best accuracy was with 0.7477 with k= 6

In [91]:
```
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alph
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Nabors (K)')
plt.tight_layout()
plt.show()
```



In [92]:
```
( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)
```

```
The best accuracy was with 0.7477590856555798 with k= 6
```

Hence the highest accuracy is given by the KNN algorithm at k=6

6 Conclusion

Based on a rule of thumb, it can be said that RMSE values between 0.2 and 0.5 shows that the model can relatively predict the data accurately, hence we can say that our model will give results with a good accuracy.

From the KNN method we can say that our model performs best with k=6 and the accuracy for the Training set is also 74.77 %