Introduction/Business Problem:

A WHO report says that approximately 1.35 million people lose their life as a result of a road traffic crash. Between 20 and 50 million more people suffer non-fatal injuries, with many incurring a disability as a result of their injury. Road traffic injuries cause considerable economic losses to individuals, their families, and to nations as a whole. These losses arise from the cost of treatment as well as lost productivity for those killed or disabled by their injuries, and for family members who need to take time off work or school to care for the injured. Road traffic crashes cost most countries 3% of their gross domestic product.

The Center of Disease control and prevention says that road traffic crashes are a leading cause of death in the U.S.A. for people aged 1-54 and the leading cause of non-natural death for healthy U.S.A. citizens residing or travelling abroad.

It would be really helpful if there was a system which can tell a driver based on the weather condition, road condition that the driver will get into an accident and how severe would that accident be so that the driver will be more careful while driving and also the driver may change the route he was planning to take.

So, the audience here would be the people driving the vehicle in the highway. It would also be very helpful for the emergency/rescue team and polices as based on the weather and road condition they could get a prediction data about what severity level accidents are likely to happen on that day so that they can plan the emergency response accordingly. The data generated from this system could be very useful in a broader sense. Analysis of these data could lead to formulating of policies and strategies related to driving, accidents. It could be also helpful in designing the new highways. Also, it could be helpful for the insurance companies also.

Data Section:

The dataset contains the vehicle collision data of Seattle from 2004 to 2020. This data is provided by Seattle Police Department (SPD) and recorded by Traffic Records. There are total 38 features in this dataset. After some exploratory analysis it can be found that there are missing values for different features. Some features have too many missing values so it would be best to remove these features. There are also some features which are unnecessary for the analysis so those features should also be excluded. If we look at severitycode column which is what we need to predict, we can see that the data is not balanced. There are 136485 entries for severitycode 1 and 58188 entries for severitycode 2.

```
In [12]: data.SEVERITYCODE.value_counts()

Out[12]: 1    136485
         2     58188
         Name: SEVERITYCODE, dtype: int64
```

Fig 1: Total entries for severitycode 1 and severitycode 2.

There are certain features which has less than 50% values entered. These features were INTKEY, EXCEPTRSNCODE, EXCEPTRSNDESC, INATTENTIONIND, PEDROWNOTGRNT, SPEEDING. These features should be removed as there too many missing values.

```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 194673 entries, 0 to 194672
Data columns (total 38 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   SEVERITYCODE    194673 non-null  int64  
 1   X               189339 non-null  float64
 2   Y               189339 non-null  float64
 3   OBJECTID        194673 non-null  int64  
 4   INCKEY          194673 non-null  int64  
 5   COLDETKEY       194673 non-null  int64  
 6   REPORTNO        194673 non-null  object 
 7   STATUS          194673 non-null  object 
 8   ADDRTYPE        192747 non-null  object 
 9   INTKEY          65070 non-null   float64
 10  LOCATION        191996 non-null  object 
 11  EXCEPTRSNCODE   84811 non-null   object 
 12  EXCEPTRSNDESC   5638 non-null    object 
 13  SEVERITYCODE.1  194673 non-null  int64  
 14  SEVERITYDESC    194673 non-null  object 
 15  COLLISIONTYPE   189769 non-null  object 
 16  PERSONCOUNT     194673 non-null  int64  
 17  PEDCOUNT        194673 non-null  int64  
 18  PEDCYLCOUNT     194673 non-null  int64  
 19  VEHCOUNT        194673 non-null  int64  
 20  INCDATE         194673 non-null  object 
 21  INCDTTM         194673 non-null  object 
 22  JUNCTIONTYPE    188344 non-null  object 
 23  SDOT_COLCODE    194673 non-null  int64  
 24  SDOT_COLDESC    194673 non-null  object 
 25  INATTENTIONIND  29805 non-null   object 
 26  UNDERINFL       189789 non-null  object 
 27  WEATHER         189592 non-null  object 
 28  ROADCOND        189661 non-null  object 
 29  LIGHTCOND       189503 non-null  object 
 30  PEDROWNOTGRNT   4667 non-null    object 
 31  SDOTCOLNUM      114936 non-null  float64
 32  SPEEDING        9333 non-null    object 
 33  ST_COLCODE      194655 non-null  object 
 34  ST_COLDESC      189769 non-null  object 
 35  SEGLANEKEY      194673 non-null  int64  
 36  CROSSWALKKEY    194673 non-null  int64  
 37  HITPARKEDCAR    194673 non-null  object 
dtypes: float64(4), int64(12), object(22)
memory usage: 56.4+ MB
```

Fig: Various features of the dataset.

Methodology:

After removing the features that were mentioned in data section, there were still some features that were not required for the analysis. Removing these features, I ended up with 8 features excluding the SEVIRITYCODE which is the prediction label.

```
In [11]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 194673 entries, 1 to 219547
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   SEVERITYCODE   194673 non-null  int64 
 1   STATUS         194673 non-null  object
 2   ADDRTYPE       192747 non-null  object
 3   SEVERITYDESC   194673 non-null  object
 4   COLLISIONTYPE  189769 non-null  object
 5   WEATHER        189592 non-null  object
 6   ROADCOND       189661 non-null  object
 7   LIGHTCOND      189503 non-null  object
 8   HITPARKEDCAR   194673 non-null  object
dtypes: int64(1), object(8)
memory usage: 19.9+ MB
```

Fig: Remaining features after removing other features not required for the analysis.

There were missing values present in these features. So first these missing values needed to be filled.

```
In [12]: data.isnull().sum()

Out[12]: SEVERITYCODE       0
         STATUS             0
         ADDRTYPE        1926
         SEVERITYDESC       0
         COLLISIONTYPE   4904
         WEATHER         5081
         ROADCOND        5012
         LIGHTCOND       5170
         HITPARKEDCAR       0
         dtype: int64
```

Fig: Total null values in each of the remaining features.

These missing values were filled using the most common values for each feature. For this I created a custom transformer called fill that performs this action.

```
In [18]: class fill(BaseEstimator,TransformerMixin):
             def __init__(self):
                 pass
             def fit(self,X,y=None):
                 return self
             def transform(self,X):
                 X['ADDRTYPE']=X['ADDRTYPE'].fillna(X['ADDRTYPE'].mode()[0])
                 X['COLLISIONTYPE']=X['COLLISIONTYPE'].fillna(X['COLLISIONTYPE'].mode()[0])
                 X['WEATHER']=X['WEATHER'].fillna(X['WEATHER'].mode()[0])
                 X['ROADCOND']=X['ROADCOND'].fillna(X['ROADCOND'].mode()[0])
                 X['LIGHTCOND']=X['LIGHTCOND'].fillna(X['LIGHTCOND'].mode()[0])
                 return X
```

Fig: A custom transformer that fills the missing values for each feature with most common values of the feature.

Taking a look at SEVERITYCODE we can see that it is unbalanced. SEVERITYCODE 1 has 136485 value counts whereas SEVERITYCODE 2 has only 58188 value counts. So, this unbalanced dataset needs to be balanced first because this could lead in creating a biased ML model.

```
In [7]: data.SEVERITYCODE.value_counts()

Out[7]: 1    136485
        2     58188
        Name: SEVERITYCODE, dtype: int64

In [8]: sns.countplot(x="SEVERITYCODE",data=data)

Out[8]: <AxesSubplot:xlabel='SEVERITYCODE', ylabel='count'>
```

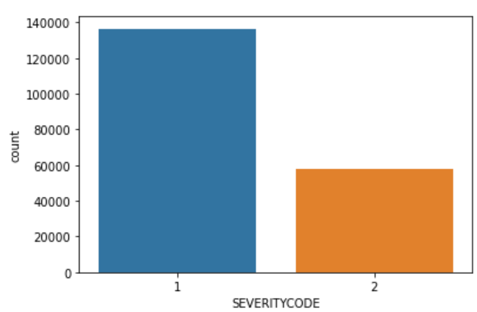

Fig: Value counts for different SEVERITYCODE values and countplot showing the same.

For this I applied the downsizing strategy. I reduced the number of SEVERITYCODE 1 values to 58188 which is that same as the number of SEVERITYCODE 2 values.

```
In [16]: sns.countplot(x="SEVERITYCODE",data=normdata)

Out[16]: <AxesSubplot:xlabel='SEVERITYCODE', ylabel='count'>
```

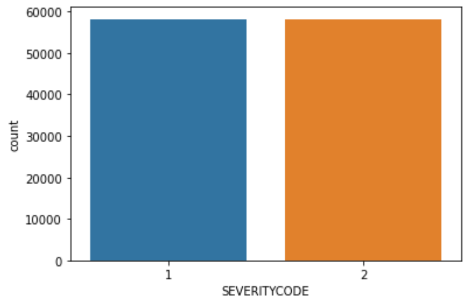

Fig: Balanced values for SEVERITYCODE 1 and 2 after downsizing.

As we can see above that there are 8 features with non-numeric categorical values, so these needed to be converted. So, I created a custom transformer factorize to convert these non-numeric categorical values. After that I created a pipeline to incorporate the two custom transformers. I used an ensemble model, RandomForestClassifier, for analysis. The reason I used this particular model is because of its high accuracy through cross validation, maintains the accuracy of large proportion of data and if there are more trees it won't allow over-fitting trees in the model. The values were first split into training and testing set using sklearn's train_test_split.

Result:

The final pipeline was trained and tested which had a training and testing score of 1. In the testing set it predicted all the values accurately. This can be seen in the confusion matrix below:

```
In [81]: confusion_matrix(y_test,ypred)

Out[81]: array([[11687,     0],
                [    0, 11589]], dtype=int64)
```

Fig: Confusion matrix showing the predicted and the original values.

Discussion:

The score of 1 in the training set is highly improbable. This score could be due to overfitting but if we look at the testing set it predicted very accurately there also. The data shows that the accident with severity code 1 occurs most in daylight condition followed by dark with street lights on condition. Similarly, severity code 2 accidents also occur most in the daylight condition followed by dark with street lights on condition. If we look at road condition, severity code 1 accidents happen mostly in dry road and severity code 2 accidents happens mostly in dry road followed by wet road. Similarly, severity code 1 accident happen in clear weather and severity code 2 accident happen in clear weather followed by raining weather.

Conclusion:

Hence, for this project the dataset was first pre-processed. The unnecessary features were removed, features with many missing values were removed. The remaining feature were further pre-processed by replacing all the missing values with the most common values for each feature. Since the data was unbalanced, so downsizing strategy was used to balance the data. The data was split into training and testing data. Random forest classifier was used for this analysis and using this model we got a training and testing score of 1.