1. Introduction

As we all know that thousands of people lost their lives or become disable every year due to some accidents. According to USA National high authority safety administration, every year car accident costs up to $850 billion. It's a huge amount and the rate the death in car accidents is also very high i.e. 3800 people per year. In addition to this 4.4 million people are seriously injured. The project aims to predict how the severity of accidents can be based on various factors like the condition of the road, condition of light of that area at the time of the accident, and many other factors.

2. Data

• Understanding the data:

There are lots of problems in the dataset which we are going to use in this problem. There are total 194672 rows/observations and 37 columns/features namely Severity Code, X(X co-ordinate of location), Y(Y co-ordinate of location),and many other you can check it here.

```
In [ ]: df.columns

Out[ ]: Index(['SEVERITYCODE', 'X', 'Y', 'OBJECTID', 'INCKEY', 'COLDETKEY', 'REPORTNO',
               'STATUS', 'ADDRTYPE', 'INTKEY', 'LOCATION', 'EXCEPTRSNCODE',
               'EXCEPTRSNDESC', 'SEVERITYCODE.1', 'SEVERITYDESC', 'COLLISIONTYPE',
               'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT', 'INCDATE',
               'INCDTTM', 'JUNCTIONTYPE', 'SDOT_COLCODE', 'SDOT_COLDESC',
               'INATTENTIONIND', 'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND',
               'PEDROWNOTGRNT', 'SDOTCOLNUM', 'SPEEDING', 'ST_COLCODE', 'ST_COLDESC',
               'SEGLANEKEY', 'CROSSWALKKEY', 'HITPARKEDCAR'],
              dtype='object')

In [ ]: df.shape

Out[ ]: (194673, 38)
```

This dataset has very high variation for our machine learning problem which to categories the severity of accidents for which we are going to use machine learning algorithms like "Logistic Regression", "Support Vector Machine", "Decision Tree" or any other. As in all of these algorithms use numerical

values as an intake to predict the result.So we have to clean the data.

- **Cleaning the data:**

First we try to find how any values are empty in all columns.We can do that like in I do below:

```
missing_value =  df.isnull()
for column in missing_value.columns:
    print(missing_value[column].value_counts())
```

```
False    194673
Name: SEVERITYCODE, dtype: int64
False    189339
True       5334
Name: X, dtype: int64
False    189339
True       5334
Name: Y, dtype: int64
False    194673
Name: OBJECTID, dtype: int64
False    194673
Name: INCKEY, dtype: int64
False    194673
Name: COLDETKEY, dtype: int64
False    194673
Name: REPORTNO, dtype: int64
False    194673
Name: STATUS, dtype: int64
False    192747
True       1926
Name: ADDRTYPE, dtype: int64
True     129603
False     65070
Name: INTKEY, dtype: int64
False    191996
True       2677
Name: LOCATION, dtype: int64
True     109862
False     84811
Name: EXCEPTRSNCODE, dtype: int64
True     189035
False      5638
```

There are many values which are empty so we try get the values for that or try to replace the empty values with average of the all values in that column and we also try to fill all the empty with the most occurring value for the columns like ”WEATHER”, “LIGHTCOND” and for columns like “UNERINFL” we convert all “No” to 0 and “ YES” to 1.

All the above change are given below in codes:

```
df1['ADDRTYPE'].fillna(value =df1['ADDRTYPE'].value_counts().idxmax(),inplace  = True)
df1['JUNCTIONTYPE'].fillna(value =df1['JUNCTIONTYPE'].value_counts().idxmax(),inplace  = True)
df1['UNDERINFL'].replace({'N':0,'Y':1}, inplace = True)
df1['WEATHER'].fillna(value =df1['WEATHER'].value_counts().idxmax(),inplace  = True)
df1['ROADCOND'].fillna(value =df1['ROADCOND'].value_counts().idxmax(),inplace  = True)
df1['UNDERINFL'].fillna(value = 0, inplace = True)
```

Next we drop some columns like “Reportno”, “Lcation” because these columns are not for our use.

```
df1 = df.drop(columns = ['LIGHTCOND','LOCATION','INCDATE','INCDTTM','SDOT_COLCODE','SDOT_COLDESC',
                         'SEVERITYDESC',
                         'SDOTCOLNUM','ST_COLCODE'
                        ,'ST_COLDESC','EXCEPTRSNCODE','EXCEPTRSNDESC','STATUS','CROSSWALKKEY',
                         'COLLISIONTYPE','COLDETKEY','REPORTNO'])
```

Now we just want to convert the columns like “ADDRTYPE” because column like these are categorical data with not numerical values. So we have convert all the columns like this one to get numerical values.
We can do that like given below:

```
ADDRTYPE = pd.get_dummies(df1['ADDRTYPE'])
```

```
WEATHER = pd.get_dummies(df1['WEATHER'])
```

```
JUNCTIONTYPE = pd.get_dummies(df1['JUNCTIONTYPE'])
```

```
ROADCOND = pd.get_dummies(df1['ROADCOND'])
```

```
ROADCOND = pd.get_dummies(df1['ROADCOND'])
```

```
HITPARKEDCAR  = pd.get_dummies(df1['HITPARKEDCAR'])
```

Now, we get our final dataframe which we are going to use in this problem
Which look this:

```
final_df.columns
```

```
Index(['SEVERITYCODE', 'X', 'Y', 'OBJECTID', 'INCKEY', 'ADDRTYPE', 'INTKEY',
       'SEVERITYCODE.1', 'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT',
       'JUNCTIONTYPE', 'UNDERINFL', 'WEATHER', 'ROADCOND', 'PEDROWNOTGRNT',
       'SEGLANEKEY', 'HITPARKEDCAR', 'Alley', 'Block', 'Intersection',
       'Blowing Sand/Dirt', 'Clear', 'Fog/Smog/Smoke', 'Other', 'Overcast',
       'Partly Cloudy', 'Raining', 'Severe Crosswind',
       'Sleet/Hail/Freezing Rain', 'Snowing', 'Unknown',
       'At Intersection (but not related to intersection)',
       'At Intersection (intersection related)', 'Driveway Junction',
       'Mid-Block (but intersection related)',
       'Mid-Block (not related to intersection)', 'Ramp Junction', 'Unknown',
       'Dry', 'Ice', 'Oil', 'Other', 'Sand/Mud/Dirt', 'Snow/Slush',
       'Standing Water', 'Unknown', 'Wet', 'N', 'Y'],
      dtype='object')
```

```
final_df = final_df.drop(columns = ['PEDROWNOTGRNT'])
```

```
final_df.shape
```

```
(194673, 50)
```

So this is our final dataset.