Data

The car accident data (i.e. Collisions-data) has been given by the Traffic Records Group in the **SDOT Traffic Management Division from Seattle, WA**. This includes all types of collisions. The collisions will display at the intersection or mid-block of a segment. The data consists of **37** independent variables and **194673** rows. The dependent variable **SEVERITYCODE**, contains numbers that correspond to different levels of severity caused by an accident from 0 to 3.

Severity codes

- **0:** No Probability - Clear Condition
- **1:** Low Probability - Chance of Property Damage
- **2:** Mild Probability - Chance of Injury
- **2b:** Probability - Chance of Serious Injury
- **3:** High Probability - Chance of Fatality

By analysing the provided Seattle car accident data, we have to train a model and it should predict the severity of an accident.

Data Processing

The provided data is not ready for data analysis, right away. So, I have to prepare the data, before 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.

After close analyses of provided data, I have decided to focus on only four features i.e. severity, weather conditions, road conditions, and light conditions among others. Apart from that, I can see that SEVERITYCODE data is not balanced, so I will use a simple statistical technique to balance it.

```
In [9]: pre_accident_df["SEVERITYCODE"].value_counts()
```

```
Out[9]: 1    136485
        2     58188
        Name: SEVERITYCODE, dtype: int64
```

From the above observation, we can deduce that the number of rows in class 1 is almost three times bigger than the number of rows in class 2. It is possible to solve the issue by down sampling the class 1 and balance the data.

```
In [10]: pre_accident_df_maj = pre_accident_df[pre_accident_df.SEVERITYCODE == 1]
         pre_accident_df_min = pre_accident_df[pre_accident_df.SEVERITYCODE == 2]

         pre_accident_df_maj_dsample = resample(pre_accident_df_maj,
                                       replace=False,
                                       n_samples=58188,
                                       random_state=123)

         balanced_accident_df = pd.concat([pre_accident_df_maj_dsample, pre_accident_df_min])

         balanced_accident_df.SEVERITYCODE.value_counts()
```

```
Out[10]: 2    58188
         1    58188
         Name: SEVERITYCODE, dtype: int64
```

The data for SEVERITYCODE is now balanced as both the classes have similar rows.