# “CAPSTONE PROJECT REPORT”

## TOPIC: CAR ACCIDENT SEVERITY

*AUTHOR: MURAD POPATTIA*

## Introduction (Business Problem):

In this project we will try to find potential location for a car accident given the weather and road conditions. Specifically, this report will be targeted to stakeholders interested in increasing public road safety in Seattle, Washington, United States.

Unfortunate incidents often occur on the road. The unpredictability of these accidents make them so dangerous. It is usually the impatient nature of the human which leads to an accident. But wouldn't it be a blessing if the number of accidents that occur on a daily basis be reduced? That is the problem which will be addressed in this project. As we know, accidents can happen anytime and anywhere. However, there are many factors which might influence the severity of the accident. For instance, these include the weather conditions, time of the day, the speed of the car and the area the car is being driven. These factors greatly contribute whether the accident will be severe or not. Having the information about the above-mentioned factors can be used to predict if the accident happens can be severe or not. The indication that there is a possibility of a severe accident if it happens might warn the car drivers to drive more carefully and hence prevent accidents.

This would greatly reduce the loss of life and also damage to the property. This would also help routing software to give a warning which would alert the drivers and their insurance companies which would help them in saving cost.

## Data:

The dataset used was provided by Coursera and it contains all collisions provided by SPD and recorded by Traffic Records, including all types of collision since 2004 to present.

It provides many columns with details on each type of colision. We excluded most of them and kept the columns that represents information data that would be available to some kind of traffic app, for example, in real time. The selected columns were:

- **SEVERITYCODE:** A code that corresponds to the severity of the collision:
  - **3**—fatality
  - **2b**—serious injury
  - **2**—injury
  - **1**—prop damage
  - **0**—unknown
- **SEVERITYDESC:** A detailed description of the severity of the collision
- **COLLISIONTYPE:** Collision Type
- **INJURIES:** The number of total injuries in the collision. This is entered by the state.
- **SERIOUSINJURIES:** The number of serious injuries in the collision. This is entered by the state.
- **FATALITIES:** The number of fatalities in the collision. This is entered by the state.
- **INCDATE:** Date of Accident
- **INCDTTM:** The Date and Time of Accident
- **JUNCTIONTYPE:** Category of junction at which collision took place
- **INATTENTIONIND:** Whether or not collision was due to inattention. **(Y/N)**
- **UNDERINFL:** Whether or not a driver involved was under the influence of drugs or alcohol.
- **WEATHER:** A description of the weather conditions during the time of the collision.
- **ROADCOND:** The condition of the road during the collision.
- **LIGHTCOND:** The light conditions during the collision.
- **SPEEDING:** Whether or not speeding was a factor in the collision. **(Y/N)**
- **HITPARKEDCAR:** Whether or not the collision involved hitting a parked car. **(Y/N)**

An overview of the data columns can be seen from the screenshot below:

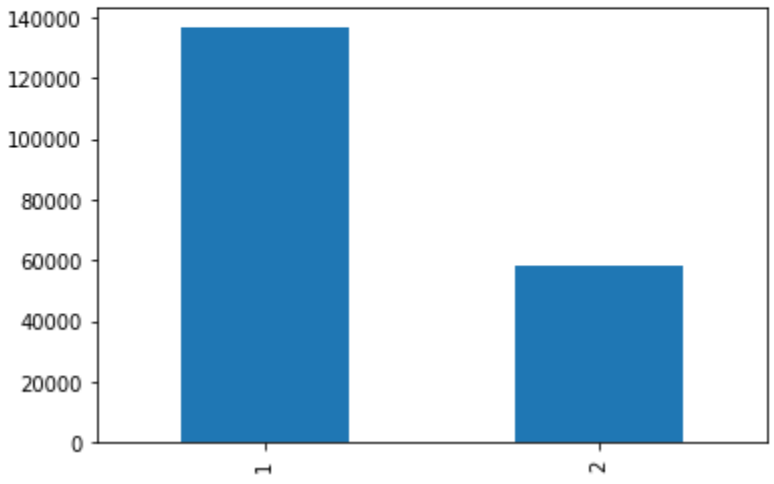
```
In [27]: df.columns

Out[27]: Index(['SEVERITYCODE', 'X', 'Y', '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')
```

The data labels only contain two classes of severity, hence this can be treated as a binary classification problem, for which we can use a logistic regression classifier in order to determine the severity of the accident.

```
In [26]: # Checking dataset balanced or not
         %matplotlib inline
         df['SEVERITYCODE'].value_counts().plot(kind='bar')

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0f66af5518>
```




The data is also imbalanced hence, we need to take appropriate measures in order to scale the data to get correct predictions.