In [1]:

```
# Importing libraries

import numpy as np
import pandas as pd
from pathlib import Path
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
```

Data Collection

Data is obtained from all road traffic accidents recorded in the Seattle municipal area between Jan 2004–Aug 2020 by the Seattle Department of Transport (SDOT).

Data is available in Seattle Open Data portal and saved as CSV.

URL: http://data-seattlecitygis.opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0 (http://data-seattlecitygis.opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0)

In [2]:

```
# Read the Data
df = pd.read_csv("C://Users/ManojKumar Chalamala/Downloads/Collisions.csv")
df.head()
```

Out[2]:

	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE
0	-122.356511	47.517361	1	327920	329420	3856094	Matched	Intersection
1	-122.361405	47.702064	2	46200	46200	1791736	Matched	Block
2	-122.317414	47.664028	3	1212	1212	3507861	Matched	Block
3	-122.318234	47.619927	4	327909	329409	EA03026	Matched	Intersection
4	-122.351724	47.560306	5	104900	104900	2671936	Matched	Block

5 rows × 40 columns

Exploratory Data Analysis

In [3]:

```
# Dimensions of the Dataframe

df_shape = df.shape
print("Dimensions of the data frame: "+str(df_shape))
```

```
Dimensions of the data frame: (221738, 40)
```

In [4]:
```
# Type of Data in Dataframe

df.dtypes
```

Out[4]:
```
X                   float64
Y                   float64
OBJECTID              int64
INCKEY                int64
COLDETKEY             int64
REPORTNO             object
STATUS               object
ADDRTYPE             object
INTKEY              float64
LOCATION             object
EXCEPTRSNCODE        object
EXCEPTRSNDESC        object
SEVERITYCODE         object
SEVERITYDESC         object
COLLISIONTYPE        object
PERSONCOUNT           int64
PEDCOUNT              int64
PEDCYLCOUNT           int64
VEHCOUNT              int64
INJURIES              int64
SERIOUSINJURIES       int64
FATALITIES            int64
INCDATE              object
INCDTTM              object
JUNCTIONTYPE         object
SDOT_COLCODE        float64
SDOT_COLDESC         object
INATTENTIONIND       object
UNDERINFL            object
WEATHER              object
ROADCOND             object
LIGHTCOND            object
PEDROWNOTGRNT        object
SDOTCOLNUM          float64
SPEEDING             object
ST_COLCODE           object
ST_COLDESC           object
SEGLANEKEY            int64
CROSSWALKKEY          int64
HITPARKEDCAR         object
dtype: object
```

In [5]:

```
# Information about the Dataframe

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221738 entries, 0 to 221737
Data columns (total 40 columns):
X                  214260 non-null float64
Y                  214260 non-null float64
OBJECTID           221738 non-null int64
INCKEY             221738 non-null int64
COLDETKEY          221738 non-null int64
REPORTNO           221738 non-null object
STATUS             221738 non-null object
ADDRTYPE           218024 non-null object
INTKEY             72027 non-null float64
LOCATION           217145 non-null object
EXCEPTRSNCODE      101335 non-null object
EXCEPTRSNDESC      11785 non-null object
SEVERITYCODE       221737 non-null object
SEVERITYDESC       221738 non-null object
COLLISIONTYPE      195287 non-null object
PERSONCOUNT        221738 non-null int64
PEDCOUNT           221738 non-null int64
PEDCYLCOUNT        221738 non-null int64
VEHCOUNT           221738 non-null int64
INJURIES           221738 non-null int64
SERIOUSINJURIES    221738 non-null int64
FATALITIES         221738 non-null int64
INCDATE            221738 non-null object
INCDTTM            221738 non-null object
JUNCTIONTYPE       209759 non-null object
SDOT_COLCODE       221737 non-null float64
SDOT_COLDESC       221737 non-null object
INATTENTIONIND     30188 non-null object
UNDERINFL          195307 non-null object
WEATHER            195097 non-null object
ROADCOND           195178 non-null object
LIGHTCOND          195008 non-null object
PEDROWNOTGRNT      5195 non-null object
SDOTCOLNUM         127205 non-null float64
SPEEDING           9936 non-null object
ST_COLCODE         212325 non-null object
ST_COLDESC         195287 non-null object
SEGLANEKEY         221738 non-null int64
CROSSWALKKEY       221738 non-null int64
HITPARKEDCAR       221738 non-null object
dtypes: float64(5), int64(12), object(23)
memory usage: 67.7+ MB
```

In [6]:

```
# Explore the Statistical features of data

df.describe().T.style.background_gradient(cmap='Set2',axis=0)
```

Out[6]:

	count	mean	std	min	25%	50%	
X	214260	-122.331	0.0300583	-122.419	-122.349	-122.33	
Y	214260	47.6202	0.056059	47.4956	47.5771	47.616	
OBJECTID	221738	110870	64010.4	1	55435.2	110870	
INCKEY	221738	145007	89372.4	1001	71721.2	127358	
COLDETKEY	221738	145237	89749.6	1001	71721.2	127358	
INTKEY	72027	37637	52000.8	23807	28653	29973	
PERSONCOUNT	221738	2.22674	1.4697	0	2	2	
PEDCOUNT	221738	0.0380945	0.201704	0	0	0	
PEDCYLCOUNT	221738	0.0273521	0.164512	0	0	0	
VEHCOUNT	221738	1.72944	0.830529	0	2	2	
INJURIES	221738	0.373964	0.73205	0	0	0	
SERIOUSINJURIES	221738	0.0152026	0.158004	0	0	0	
FATALITIES	221738	0.0017002	0.044967	0	0	0	
SDOT_COLCODE	221737	13.3833	7.29829	0	11	11	
SDOTCOLNUM	127205	7.97106e+06	2.61152e+06	1.00702e+06	6.00703e+06	8.03301e+06	1.
SEGLANEKEY	221738	262.625	3252.88	0	0	0	
CROSSWALKKEY	221738	9568.04	71427.8	0	0	0	

In [7]:
```
# Check for any null values in the Dataframe

df.isnull().sum(axis=0)
```

Out[7]:
```
X                    7478
Y                    7478
OBJECTID                0
INCKEY                  0
COLDETKEY               0
REPORTNO                0
STATUS                  0
ADDRTYPE             3714
INTKEY             149711
LOCATION             4593
EXCEPTRSNCODE      120403
EXCEPTRSNDESC      209953
SEVERITYCODE            1
SEVERITYDESC            0
COLLISIONTYPE       26451
PERSONCOUNT             0
PEDCOUNT                0
PEDCYLCOUNT             0
VEHCOUNT                0
INJURIES                0
SERIOUSINJURIES         0
FATALITIES              0
INCDATE                 0
INCDTTM                 0
JUNCTIONTYPE        11979
SDOT_COLCODE            1
SDOT_COLDESC            1
INATTENTIONIND     191550
UNDERINFL           26431
WEATHER             26641
ROADCOND            26560
LIGHTCOND           26730
PEDROWNOTGRNT      216543
SDOTCOLNUM          94533
SPEEDING           211802
ST_COLCODE           9413
ST_COLDESC          26451
SEGLANEKEY              0
CROSSWALKKEY            0
HITPARKEDCAR            0
dtype: int64
```

Data Cleaning

Remove the data with unknown information in the Target variable

The predefined target variable in the data deteremines the Car Accident Severity. However there are few rows in the dataframe with 'SEVERITYCODE = 0' which means an accident with "Unknown" Severity. We cannot use these accident data with unknown information to predict the Car Accident severity. So these rows should be dropped.

In [8]:

```
# Identify the rows with SeverityDESC = Unknown
df['SEVERITYDESC'].value_counts()
```

Out[8]:

```
Property Damage Only Collision    137776
Injury Collision                   58842
Unknown                            21657
Serious Injury Collision            3111
Fatality Collision                   352
Name: SEVERITYDESC, dtype: int64
```

In [9]:

```
# Remove the Unknown Accident Severity rows
Unknown = df['SEVERITYDESC'] == 'Unknown'
df.drop(df.index[Unknown], inplace=True)

# Reset index of the data frame
df.reset_index(inplace=True)
```

Relabel the Target Variable

The Target Variable "SEVERITYCODE" is having values (0, 1, 2, 2b, 3). It contains categorical values. So this has to be converted into numerical format. We have already dropped the rows with code value "0". So we are left with (1, 2, 2b, 3)

Relebel the codes from (1, 2, 2b, 3) to (1, 2, 3, 4).

In [10]:

```
# Values before Conversion
print(df["SEVERITYCODE"].value_counts())

#Clean...
count2b = 0
count3  = 0
for i in range(0,len(df["SEVERITYCODE"])):
    if df["SEVERITYDESC"][i] == 'Serious Injury Collision':
        df["SEVERITYCODE"][i] = 3
        count2b += 1
    if df["SEVERITYDESC"][i] == 'Fatality Collision':
        df["SEVERITYCODE"][i] = 4
        count3 += 1

#Make sure that SEVERITYCODE is cast as an integer, rather than an object
df = df.astype({'SEVERITYCODE':np.int})



# Converted values
df["SEVERITYCODE"].value_counts()
```

```
1     137776
2      58842
2b      3111
3        352
Name: SEVERITYCODE, dtype: int64

C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\ipykernel_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  if __name__ == '__main__':
C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\ipykernel_launcher.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  if sys.path[0] == '':
```

Out[10]:

```
1    137776
2     58842
3      3111
4       352
Name: SEVERITYCODE, dtype: int64
```

Remove Columns with unnecessary information

In [11]:

```
df = df.drop(['OBJECTID','INCKEY','LOCATION','COLDETKEY','REPORTNO','STATUS',
'INTKEY','EXCEPTRSNCODE',
              'EXCEPTRSNDESC','SEVERITYDESC','INCDATE','SDOT_COLCODE','SDOT_CO
LDESC','SDOTCOLNUM','ST_COLCODE',
              'ST_COLDESC','SEGLANEKEY','CROSSWALKKEY','INCDTTM'],axis=1)
df.columns
```

Out[11]:

```
Index(['index', 'X', 'Y', 'ADDRTYPE', 'SEVERITYCODE', 'COLLISIONTYPE',
       'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT', 'INJURIES',
       'SERIOUSINJURIES', 'FATALITIES', 'JUNCTIONTYPE', 'INATTENTIONIND',
       'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND', 'PEDROWNOTGRNT',
       'SPEEDING', 'HITPARKEDCAR'],
      dtype='object')
```

Finding missing values and handling them

In [12]:
```
def missing_function(df):
    missing_data = df.isnull()
    missing_data.head()

    for column in missing_data.columns.values.tolist():
        print(column)
        print(missing_data[column].value_counts())
        print(df[column].value_counts())
        print("")

missing_function(df)
```

```
index
False    200081
Name: index, dtype: int64
2047      1
208079    1
191735    1
189686    1
195829    1
         ..
188937    1
190984    1
170502    1
164357    1
0         1
Name: index, Length: 200081, dtype: int64

X
False    194672
True       5409
Name: X, dtype: int64
-122.332653    269
-122.344896    263
-122.328079    262
-122.344997    248
-122.299160    236
              ...
-122.298305      1
-122.357075      1
-122.389188      1
-122.360755      1
-122.403378      1
Name: X, Length: 24033, dtype: int64

Y
False    194672
True       5409
Name: Y, dtype: int64
47.708655    269
47.717173    263
47.604161    262
47.725036    248
47.579673    236
            ...
47.644114      1
47.693902      1
47.594138      1
47.680666      1
47.690589      1
Name: Y, Length: 24033, dtype: int64

ADDRTYPE
False    198148
True       1933
Name: ADDRTYPE, dtype: int64
Block           129852
Intersection     67532
Alley              764
```

```
Name: ADDRTYPE, dtype: int64

SEVERITYCODE
False    200081
Name: SEVERITYCODE, dtype: int64
1    137776
2     58842
3      3111
4       352
Name: SEVERITYCODE, dtype: int64

COLLISIONTYPE
False    195285
True       4796
Name: COLLISIONTYPE, dtype: int64
Parked Car    48558
Angles        35588
Rear Ended    34706
Other         24601
Sideswipe     18900
Left Turn     14121
Pedestrian     7668
Cycles         5936
Right Turn     3018
Head On        2189
Name: COLLISIONTYPE, dtype: int64

PERSONCOUNT
False    200081
Name: PERSONCOUNT, dtype: int64
2     117291
3      36564
4      15024
1      13692
5       6800
0       5575
6       2796
7       1180
8        547
9        227
10       133
11        59
12        35
13        22
14        22
15        11
17        11
16         8
44         6
20         6
25         6
18         6
19         6
22         5
29         4
26         4
23         3
```

```
32         3
47         3
27         3
28         3
37         3
34         3
21         2
36         2
31         2
30         2
24         2
35         1
81         1
39         1
41         1
43         1
48         1
53         1
54         1
57         1
93         1
Name: PERSONCOUNT, dtype: int64

PEDCOUNT
False    200081
Name: PEDCOUNT, dtype: int64
0    192006
1      7761
2       274
3        28
4         9
5         2
6         1
Name: PEDCOUNT, dtype: int64

PEDCYLCOUNT
False    200081
Name: PEDCYLCOUNT, dtype: int64
0    194068
1      5962
2        51
Name: PEDCYLCOUNT, dtype: int64

VEHCOUNT
False    200081
Name: VEHCOUNT, dtype: int64
2     150426
1      27920
3      13387
0       5020
4       2525
5        557
6        153
7         53
8         18
9         10
11         6
```

```
10         2
15         1
14         1
13         1
12         1
Name: VEHCOUNT, dtype: int64

INJURIES
False    200081
Name: INJURIES, dtype: int64
0     138011
1      47364
2      10703
3       2734
4        817
5        274
6        100
7         40
8         12
9         10
10         6
11         5
13         2
78         1
15         1
12         1
Name: INJURIES, dtype: int64

SERIOUSINJURIES
False    200081
Name: SERIOUSINJURIES, dtype: int64
0     196967
1       2946
2        133
3         23
4          6
5          5
41         1
Name: SERIOUSINJURIES, dtype: int64

FATALITIES
False    200081
Name: FATALITIES, dtype: int64
0    199729
1       334
2        14
3         2
5         1
4         1
Name: FATALITIES, dtype: int64

JUNCTIONTYPE
False    193698
True       6383
Name: JUNCTIONTYPE, dtype: int64
Mid-Block (not related to intersection)              92224
At Intersection (intersection related)               65233
```

```
Mid-Block (but intersection related)                     23079
Driveway Junction                                        10852
At Intersection (but not related to intersection)         2130
Ramp Junction                                              171
Unknown                                                      9
Name: JUNCTIONTYPE, dtype: int64

INATTENTIONIND
True     169893
False     30188
Name: INATTENTIONIND, dtype: int64
Y    30188
Name: INATTENTIONIND, dtype: int64

UNDERINFL
False    195305
True       4776
Name: UNDERINFL, dtype: int64
N    104000
0     81676
Y      5399
1      4230
Name: UNDERINFL, dtype: int64

WEATHER
False    195094
True       4987
Name: WEATHER, dtype: int64
Clear                       114806
Raining                      34037
Overcast                     28555
Unknown                      15131
Snowing                        919
Other                          860
Fog/Smog/Smoke                 577
Sleet/Hail/Freezing Rain       116
Blowing Sand/Dirt               56
Severe Crosswind                26
Partly Cloudy                   10
Blowing Snow                     1
Name: WEATHER, dtype: int64

ROADCOND
False    195175
True       4906
Name: ROADCOND, dtype: int64
Dry               128660
Wet                48734
Unknown            15139
Ice                 1232
Snow/Slush          1014
Other                136
Standing Water       119
Sand/Mud/Dirt         77
Oil                   64
Name: ROADCOND, dtype: int64
```

```
LIGHTCOND
False    195005
True       5076
Name: LIGHTCOND, dtype: int64
Daylight                    119552
Dark - Street Lights On      50139
Unknown                      13533
Dusk                          6085
Dawn                          2609
Dark - No Street Lights       1580
Dark - Street Lights Off      1239
Other                          244
Dark - Unknown Lighting         24
Name: LIGHTCOND, dtype: int64

PEDROWNOTGRNT
True     194887
False      5194
Name: PEDROWNOTGRNT, dtype: int64
Y    5194
Name: PEDROWNOTGRNT, dtype: int64

SPEEDING
True     190146
False      9935
Name: SPEEDING, dtype: int64
Y    9935
Name: SPEEDING, dtype: int64

HITPARKEDCAR
False    200081
Name: HITPARKEDCAR, dtype: int64
N    192479
Y      7602
Name: HITPARKEDCAR, dtype: int64
```

In [13]:

```
df.replace(r'^\s*$', np.nan, regex=True)
df.replace("Unknown", np.nan, inplace = True)
df.replace("Other", np.nan, inplace = True)

#removing columns with more than 20% values missing (INATTENTIONIND,PEDROWNOTG
RNT,SPEEDING)
df = df.drop(["INATTENTIONIND","PEDROWNOTGRNT","SPEEDING"],axis=1)

#removing rows for columns with less than 20% values missing (X, Y,COLLISIONTY
PE,JUNCTIONTYPE,
                                                            #UNDERINFL,WEATHE
R,ROADCOND,LIGHTCOND)
df.dropna(subset=["X","Y","COLLISIONTYPE","JUNCTIONTYPE","UNDERINFL","WEATHER"
,"ROADCOND","LIGHTCOND"],
          axis=0, inplace=True)

#making sure all missing values are handled with
print(df.info())
missing_function(df)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 148171 entries, 0 to 200080
Data columns (total 19 columns):
index              148171 non-null int64
X                  148171 non-null float64
Y                  148171 non-null float64
ADDRTYPE           148171 non-null object
SEVERITYCODE       148171 non-null int64
COLLISIONTYPE      148171 non-null object
PERSONCOUNT        148171 non-null int64
PEDCOUNT           148171 non-null int64
PEDCYLCOUNT        148171 non-null int64
VEHCOUNT           148171 non-null int64
INJURIES           148171 non-null int64
SERIOUSINJURIES    148171 non-null int64
FATALITIES         148171 non-null int64
JUNCTIONTYPE       148171 non-null object
UNDERINFL          148171 non-null object
WEATHER            148171 non-null object
ROADCOND           148171 non-null object
LIGHTCOND          148171 non-null object
HITPARKEDCAR       148171 non-null object
dtypes: float64(2), int64(9), object(8)
memory usage: 22.6+ MB
None
index
False    148171
Name: index, dtype: int64
2047      1
130325    1
183667    1
181618    1
185712    1
         ..
166148    1
176387    1
178434    1
174336    1
0         1
Name: index, Length: 148171, dtype: int64

X
False    148171
Name: X, dtype: int64
-122.328079    244
-122.332653    195
-122.344896    187
-122.344997    181
-122.299160    180
              ...
-122.326846      1
-122.350101      1
-122.363154      1
-122.405820      1
-122.366112      1
Name: X, Length: 20666, dtype: int64
```

```
Y
False    148171
Name: Y, dtype: int64
47.604161    244
47.708655    195
47.717173    187
47.725036    181
47.579673    180
            ...
47.666429      1
47.705400      1
47.532706      1
47.585025      1
47.690589      1
Name: Y, Length: 20666, dtype: int64

ADDRTYPE
False    148171
Name: ADDRTYPE, dtype: int64
Block           88522
Intersection    59649
Name: ADDRTYPE, dtype: int64

SEVERITYCODE
False    148171
Name: SEVERITYCODE, dtype: int64
1    95913
2    49569
3     2449
4      240
Name: SEVERITYCODE, dtype: int64

COLLISIONTYPE
False    148171
Name: COLLISIONTYPE, dtype: int64
Angles        34479
Parked Car    32772
Rear Ended    32132
Sideswipe     17303
Left Turn     13677
Pedestrian     7243
Cycles         5659
Right Turn     2833
Head On        2073
Name: COLLISIONTYPE, dtype: int64

PERSONCOUNT
False    148171
Name: PERSONCOUNT, dtype: int64
2     87190
3     31902
4     13251
5      6138
0      4607
6      2537
7      1057
8       488
```

```
1       466
9       200
10      120
11       46
12       30
14       20
13       18
15       11
17       11
16        7
44        6
18        6
19        5
22        4
29        4
26        4
32        3
20        3
23        3
37        3
25        3
27        3
28        3
47        3
34        3
36        2
31        2
21        2
24        2
53        1
41        1
43        1
54        1
39        1
35        1
30        1
93        1
Name: PERSONCOUNT, dtype: int64

PEDCOUNT
False    148171
Name: PEDCOUNT, dtype: int64
0    140604
1      7276
2       259
3        25
4         5
6         1
5         1
Name: PEDCOUNT, dtype: int64

PEDCYLCOUNT
False    148171
Name: PEDCYLCOUNT, dtype: int64
0    142462
1      5660
2        49
```

```
Name: PEDCYLCOUNT, dtype: int64

VEHCOUNT
False    148171
Name: VEHCOUNT, dtype: int64
2     120191
1      12557
3      12174
4       2311
5        505
0        223
6        138
7         41
8         14
9          9
11         3
15         1
14         1
13         1
12         1
10         1
Name: VEHCOUNT, dtype: int64

INJURIES
False    148171
Name: INJURIES, dtype: int64
0     96075
1     39245
2      9304
3      2447
4       714
5       239
6        81
7        35
8        10
9         9
10        6
11        2
78        1
15        1
13        1
12        1
Name: INJURIES, dtype: int64

SERIOUSINJURIES
False    148171
Name: SERIOUSINJURIES, dtype: int64
0     145690
1       2350
2        103
3         18
5          5
4          4
41         1
Name: SERIOUSINJURIES, dtype: int64

FATALITIES
```

```
False    148171
Name: FATALITIES, dtype: int64
0    147931
1       235
2         4
5         1
Name: FATALITIES, dtype: int64

JUNCTIONTYPE
False    148171
Name: JUNCTIONTYPE, dtype: int64
Mid-Block (not related to intersection)              63766
At Intersection (intersection related)               58060
Mid-Block (but intersection related)                 18189
Driveway Junction                                     6479
At Intersection (but not related to intersection)     1565
Ramp Junction                                          112
Name: JUNCTIONTYPE, dtype: int64

UNDERINFL
False    148171
Name: UNDERINFL, dtype: int64
N    80860
0    60669
Y     3763
1     2879
Name: UNDERINFL, dtype: int64

WEATHER
False    148171
Name: WEATHER, dtype: int64
Clear                       96360
Raining                     27389
Overcast                    23224
Snowing                       631
Fog/Smog/Smoke                424
Sleet/Hail/Freezing Rain       81
Blowing Sand/Dirt              39
Severe Crosswind               16
Partly Cloudy                   7
Name: WEATHER, dtype: int64

ROADCOND
False    148171
Name: ROADCOND, dtype: int64
Dry               107762
Wet                38977
Ice                  695
Snow/Slush           635
Standing Water        51
Sand/Mud/Dirt         30
Oil                   21
Name: ROADCOND, dtype: int64

LIGHTCOND
False    148171
Name: LIGHTCOND, dtype: int64
```

```
Daylight                    100665
Dark - Street Lights On      38503
Dusk                          5010
Dawn                          2007
Dark - No Street Lights       1059
Dark - Street Lights Off       913
Dark - Unknown Lighting         14
Name: LIGHTCOND, dtype: int64

HITPARKEDCAR
False    148171
Name: HITPARKEDCAR, dtype: int64
N    143537
Y      4634
Name: HITPARKEDCAR, dtype: int64
```

In [14]:

```
df['UNDERINFL'] = df['UNDERINFL'].replace(['0'],'N')
df['UNDERINFL'] = df['UNDERINFL'].replace(['1'],'Y')
```

Data Visualization

In [15]:

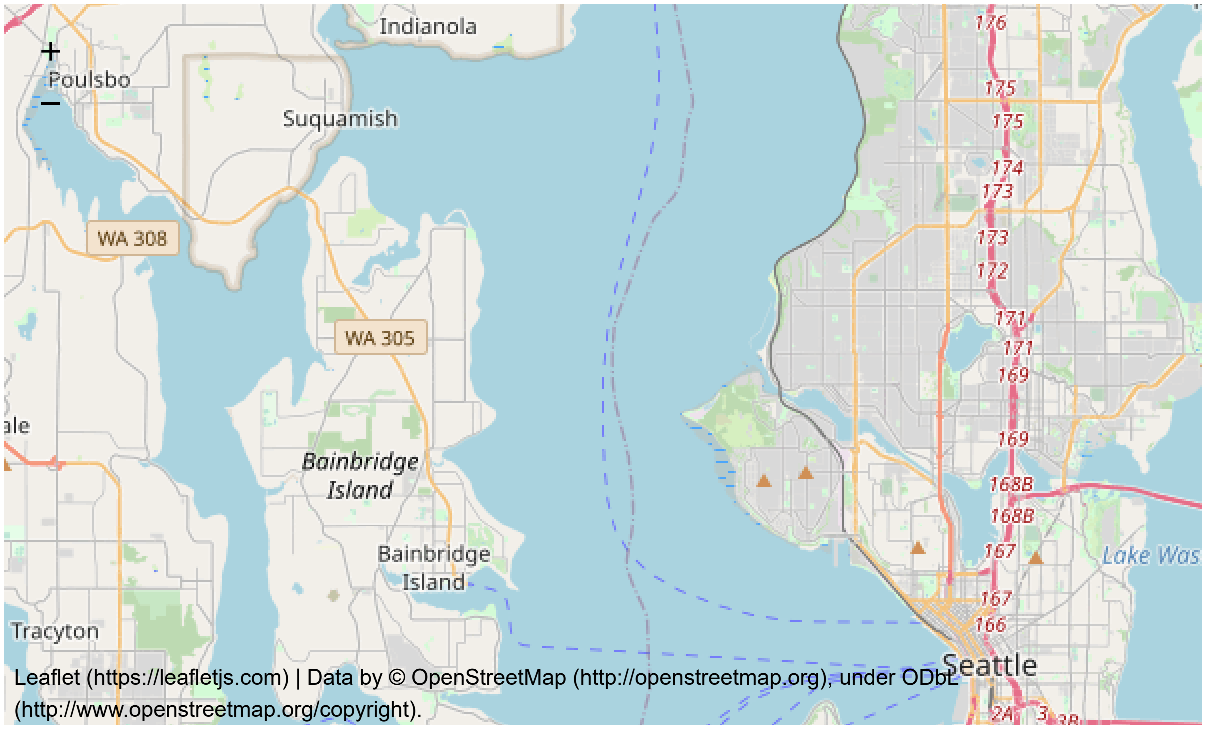
```
import pip
!pip install folium
```

```
Requirement already satisfied: folium in c:\users\manojkumar chalamala\appdat
a\local\continuum\anaconda3\lib\site-packages (0.11.0)
Requirement already satisfied: numpy in c:\users\manojkumar chalamala\appdata
\local\continuum\anaconda3\lib\site-packages (from folium) (1.16.5)
Requirement already satisfied: jinja2>=2.9 in c:\users\manojkumar chalamala\a
ppdata\local\continuum\anaconda3\lib\site-packages (from folium) (2.10.3)
Requirement already satisfied: requests in c:\users\manojkumar chalamala\appd
ata\local\continuum\anaconda3\lib\site-packages (from folium) (2.22.0)
Requirement already satisfied: branca>=0.3.0 in c:\users\manojkumar chalamala
\appdata\local\continuum\anaconda3\lib\site-packages (from folium) (0.4.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\manojkumar chalam
ala\appdata\local\continuum\anaconda3\lib\site-packages (from jinja2>=2.9->fo
lium) (1.1.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\manojkumar chalamala\appdata\local\continuum\anaconda3\lib\site-pack
ages (from requests->folium) (1.24.2)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\manojkumar c
halamala\appdata\local\continuum\anaconda3\lib\site-packages (from requests->
folium) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\manojkumar chal
amala\appdata\local\continuum\anaconda3\lib\site-packages (from requests->fol
ium) (2019.9.11)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\manojkumar chalamal
a\appdata\local\continuum\anaconda3\lib\site-packages (from requests->folium)
(2.8)
```

In [16]:

```
import folium

longitude = df["X"].mean()
latitude = df["Y"].mean()

folium.Map(location=[latitude, longitude], zoom_start=11)
```

Out[16]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [17]:

```
df.isnull().sum(axis=0)
```

Out[17]:

```
index             0
X                 0
Y                 0
ADDRTYPE          0
SEVERITYCODE      0
COLLISIONTYPE     0
PERSONCOUNT       0
PEDCOUNT          0
PEDCYLCOUNT       0
VEHCOUNT          0
INJURIES          0
SERIOUSINJURIES   0
FATALITIES        0
JUNCTIONTYPE      0
UNDERINFL         0
WEATHER           0
ROADCOND          0
LIGHTCOND         0
HITPARKEDCAR      0
dtype: int64
```

In [18]:
```
MyData = df
MyData.head()
```

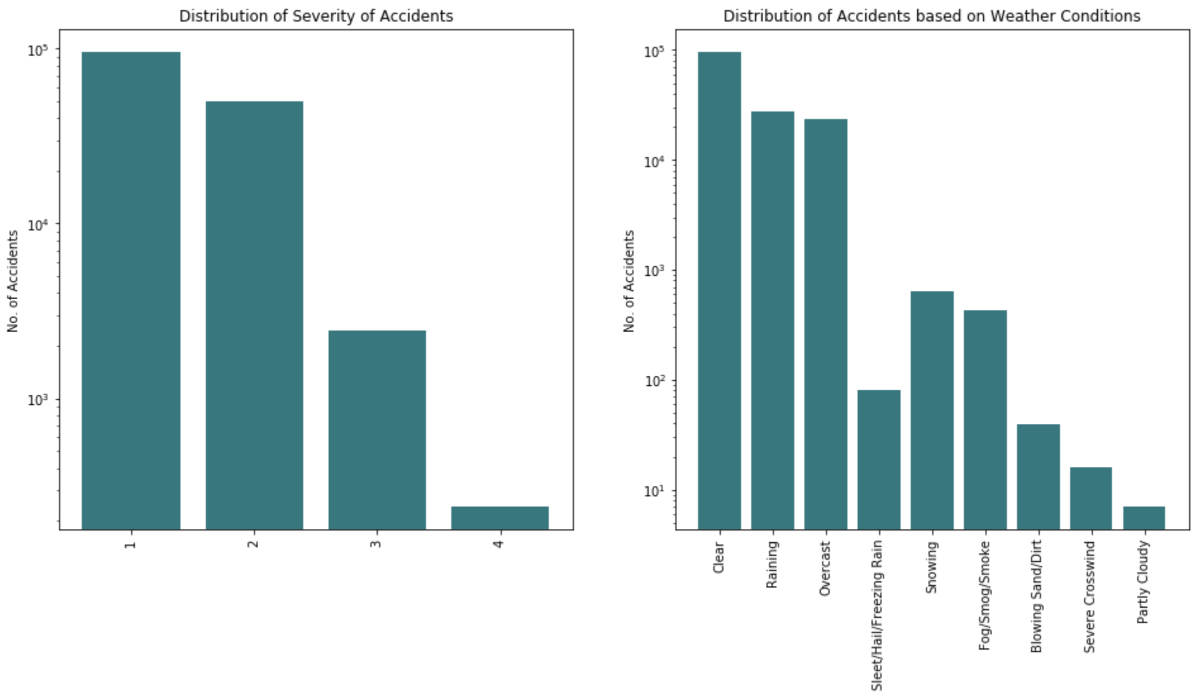
Out[18]:

	index	X	Y	ADDRTYPE	SEVERITYCODE	COLLISIONTYPE	PERSONCOUNT
0	0	-122.356511	47.517361	Intersection	1	Angles	2
1	1	-122.361405	47.702064	Block	1	Rear Ended	2
2	2	-122.317414	47.664028	Block	2	Head On	2
3	3	-122.318234	47.619927	Intersection	2	Pedestrian	3
5	5	-122.333067	47.544302	Block	1	Rear Ended	2

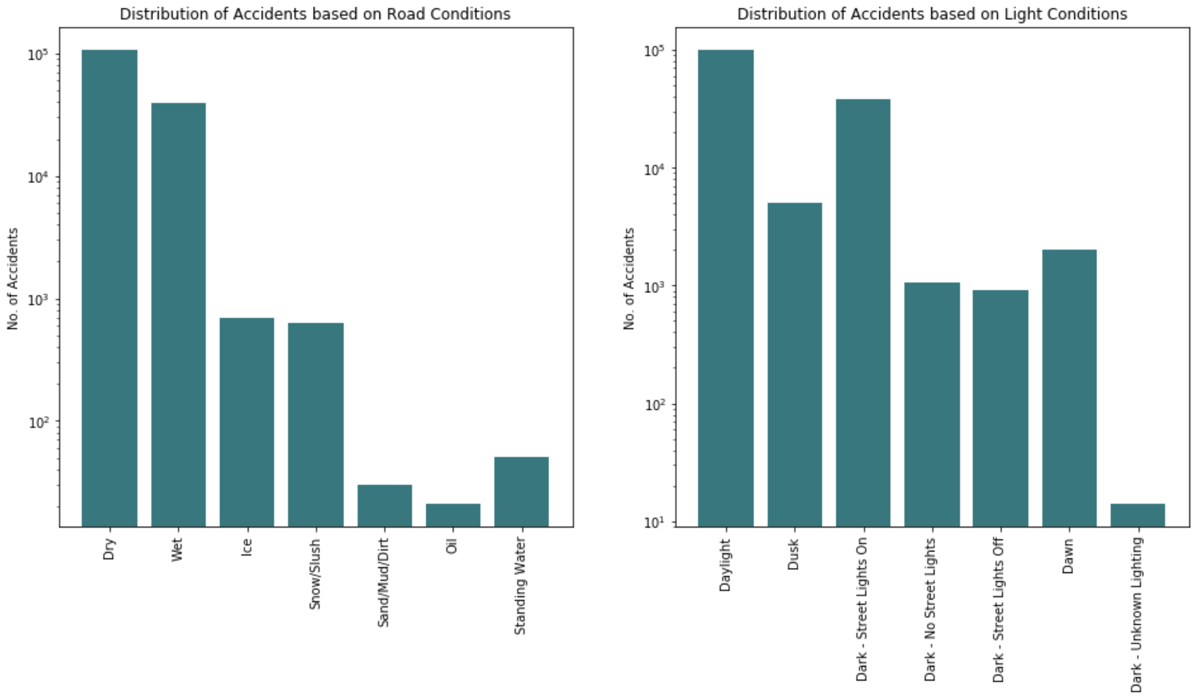
Histograms

In [19]:
```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
from collections import Counter
```

In [20]:
```
plt.rcParams["figure.figsize"] = (16,16)
plt.subplot(2,2,1)
freqs = Counter(MyData["SEVERITYCODE"])
xvals = range(len(freqs.values()))
plt.title("Distribution of Severity of Accidents")
plt.ylabel("No. of Accidents")
#plt.xlabel("Accident Severity")
plt.bar(xvals, freqs.values(), color='#37777D')
plt.xticks(xvals, freqs.keys(), rotation='vertical')
plt.yscale('log')



plt.subplot(2,2,2)
freqs = Counter(MyData["WEATHER"])
xvals = range(len(freqs.values()))
plt.title("Distribution of Accidents based on Weather Conditions")
plt.ylabel("No. of Accidents")
#plt.xlabel("Weather Conditions")
plt.bar(xvals, freqs.values(), color='#37777D')
plt.xticks(xvals, freqs.keys(), rotation='vertical')
plt.yscale('log')
```

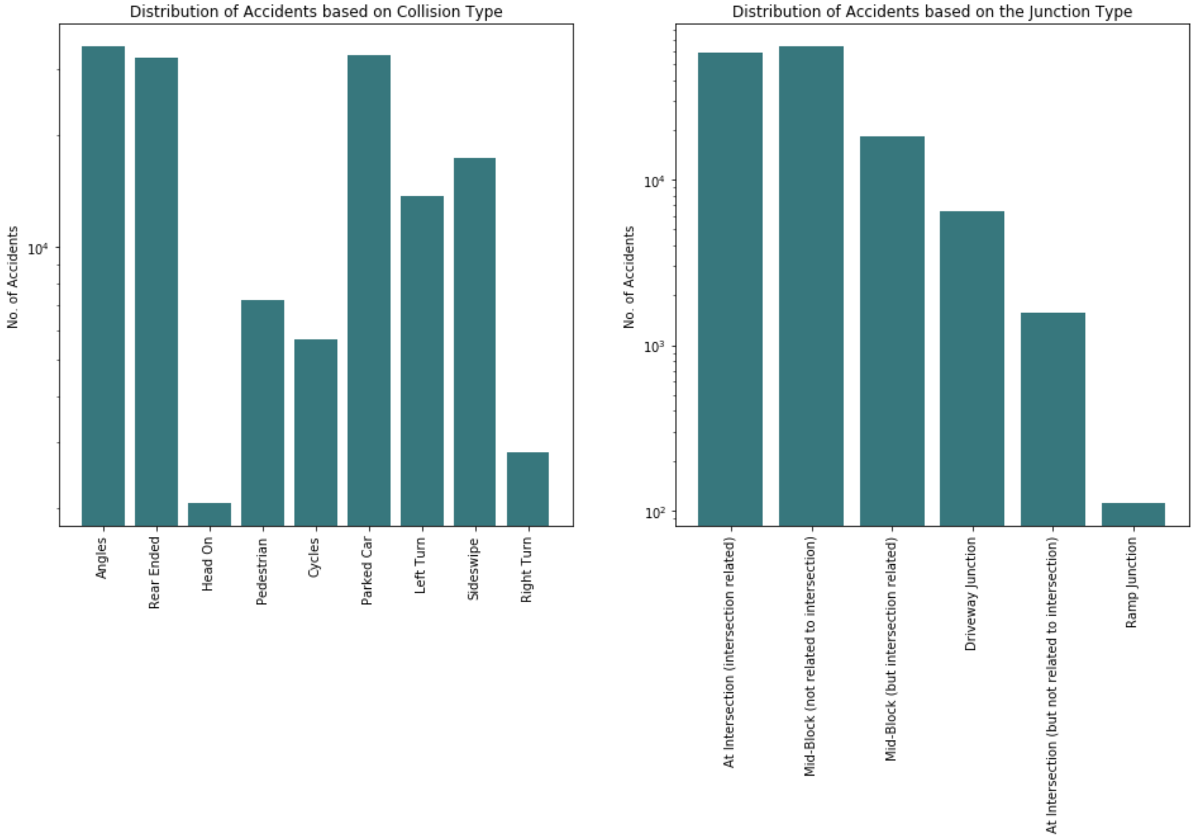


In [21]:
```
plt.subplot(2,2,1)
freqs = Counter(MyData["ROADCOND"])
xvals = range(len(freqs.values()))
plt.title("Distribution of Accidents based on Road Conditions")
plt.ylabel("No. of Accidents")
#plt.xlabel("Road Conditions")
plt.bar(xvals, freqs.values(), color='#37777D')
plt.xticks(xvals, freqs.keys(), rotation='vertical')
plt.yscale('log')



plt.subplot(2,2,2)
freqs = Counter(MyData["LIGHTCOND"])
xvals = range(len(freqs.values()))
plt.title("Distribution of Accidents based on Light Conditions")
plt.ylabel("No. of Accidents")
#plt.xlabel("Light Conditions")
plt.bar(xvals, freqs.values(), color='#37777D')
plt.xticks(xvals, freqs.keys(), rotation='vertical')
plt.yscale('log')
```



In [22]:

```
plt.subplot(2,2,1)
freqs = Counter(MyData["COLLISIONTYPE"])
xvals = range(len(freqs.values()))
plt.title("Distribution of Accidents based on Collision Type")
plt.ylabel("No. of Accidents")
plt.bar(xvals, freqs.values(), color='#37777D')
plt.xticks(xvals, freqs.keys(), rotation='vertical')
plt.yscale('log')



plt.subplot(2,2,2)
freqs = Counter(MyData["JUNCTIONTYPE"])
xvals = range(len(freqs.values()))
plt.title("Distribution of Accidents based on the Junction Type")
plt.ylabel("No. of Accidents")
plt.bar(xvals, freqs.values(), color='#37777D')
plt.xticks(xvals, freqs.keys(), rotation='vertical')
plt.yscale('log')
```



Balancing the DataSet

```
In [23]: MyData['SEVERITYCODE'].value_counts()
```

```
Out[23]: 1    95913
         2    49569
         3     2449
         4      240
         Name: SEVERITYCODE, dtype: int64
```

From the above list, most of the accidents involve either property damage or minor injuries. Very few accidents involve serious injuries and Fatalities.

If we train this model, the model will be biased. We need to balance the data by resampling.

Downsampling SEVERITYCODE 1, 2 and 3 to match the number of samples in SEVERITYCODE 4

```
In [24]: from sklearn.utils import resample

         #Re-sample the dataset
         shuffled_data = MyData.sample(frac=1, random_state=4)

         #Create separate dataframes for each of the values of SEVERITYCODE
         code_1 = shuffled_data.loc[shuffled_data["SEVERITYCODE"] == 1]
         code_2 = shuffled_data.loc[shuffled_data["SEVERITYCODE"] == 2]
         code_3 = shuffled_data.loc[shuffled_data["SEVERITYCODE"] == 3]
         code_4 = shuffled_data.loc[shuffled_data["SEVERITYCODE"] == 4]

         code_1_resample = shuffled_data.loc[shuffled_data["SEVERITYCODE"] == 1].sample
         (n=len(code_4), random_state=42)
         code_2_resample = shuffled_data.loc[shuffled_data["SEVERITYCODE"] == 2].sample
         (n=len(code_4), random_state=42)
         code_3_resample = shuffled_data.loc[shuffled_data["SEVERITYCODE"] == 3].sample
         (n=len(code_4), random_state=42)
         code_4_resample = code_4

         resampled_df = pd.concat([code_1_resample, code_2_resample, code_3_resample, c
         ode_4_resample])

         print(resampled_df.shape)
```

```
(960, 19)
```

Encoding Categorical columns and creating dummies

In [25]:

```
Feature = resampled_df.iloc[:,1:]

#Encoding Categorical Features - Training Dataset
Feature = pd.get_dummies(data=Feature, columns=['ADDRTYPE','COLLISIONTYPE','JUNCTIONTYPE','WEATHER',
                                                            'ROADCOND','LIGHTCOND','UNDERINFL','HITPARKEDCAR'])



Feature = Feature.drop(["SEVERITYCODE","INJURIES","SERIOUSINJURIES","FATALITIES","PERSONCOUNT"], axis=1)
```

In [26]:

```
Feature.isnull().sum(axis=0)
```

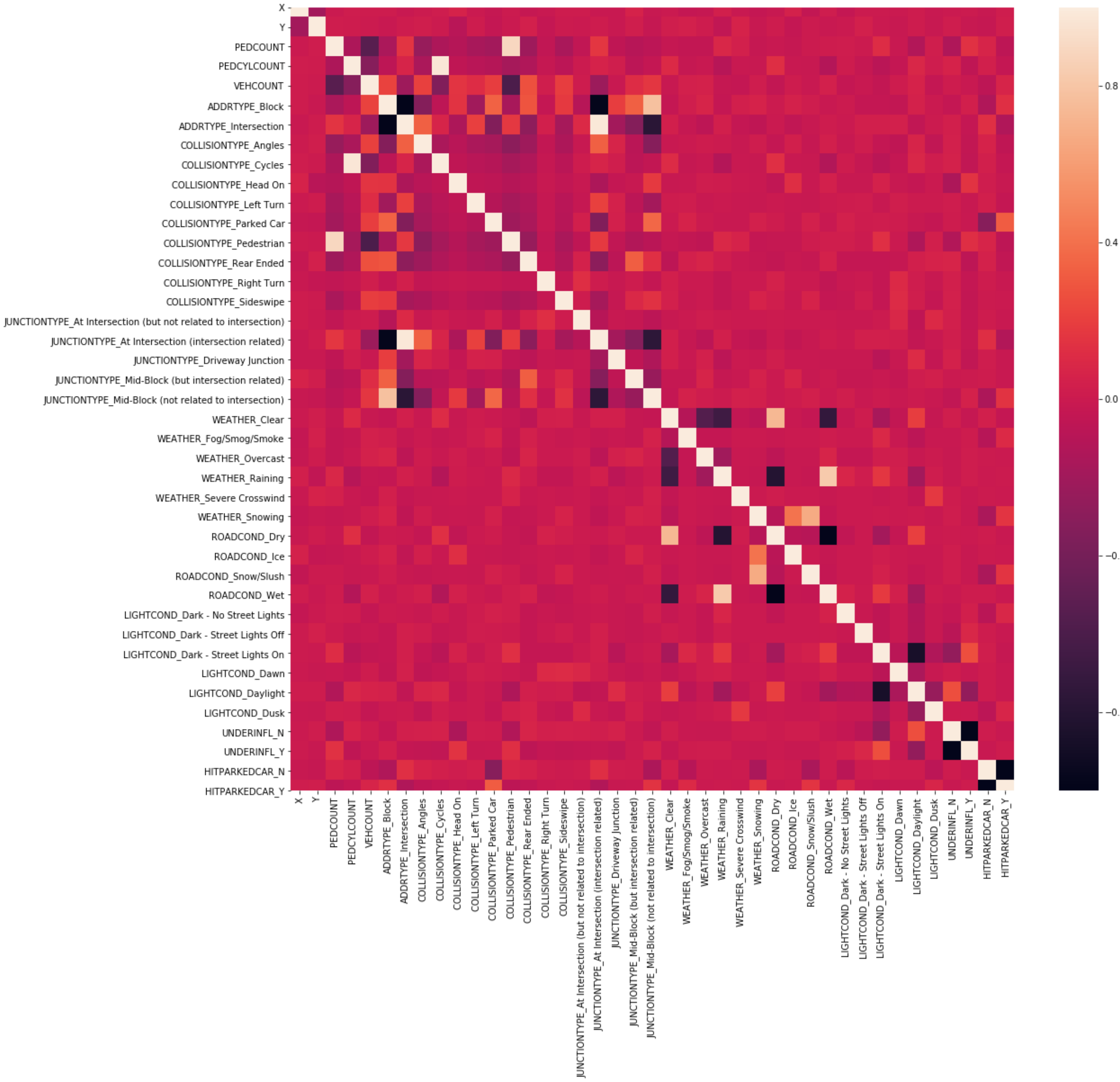
Out[26]:

```
X                                                                 0
Y                                                                 0
PEDCOUNT                                                          0
PEDCYLCOUNT                                                       0
VEHCOUNT                                                          0
ADDRTYPE_Block                                                    0
ADDRTYPE_Intersection                                             0
COLLISIONTYPE_Angles                                              0
COLLISIONTYPE_Cycles                                              0
COLLISIONTYPE_Head On                                             0
COLLISIONTYPE_Left Turn                                           0
COLLISIONTYPE_Parked Car                                          0
COLLISIONTYPE_Pedestrian                                          0
COLLISIONTYPE_Rear Ended                                          0
COLLISIONTYPE_Right Turn                                          0
COLLISIONTYPE_Sideswipe                                           0
JUNCTIONTYPE_At Intersection (but not related to intersection)    0
JUNCTIONTYPE_At Intersection (intersection related)               0
JUNCTIONTYPE_Driveway Junction                                    0
JUNCTIONTYPE_Mid-Block (but intersection related)                 0
JUNCTIONTYPE_Mid-Block (not related to intersection)              0
WEATHER_Clear                                                     0
WEATHER_Fog/Smog/Smoke                                            0
WEATHER_Overcast                                                  0
WEATHER_Raining                                                   0
WEATHER_Severe Crosswind                                          0
WEATHER_Snowing                                                   0
ROADCOND_Dry                                                      0
ROADCOND_Ice                                                      0
ROADCOND_Snow/Slush                                               0
ROADCOND_Wet                                                      0
LIGHTCOND_Dark - No Street Lights                                 0
LIGHTCOND_Dark - Street Lights Off                                0
LIGHTCOND_Dark - Street Lights On                                 0
LIGHTCOND_Dawn                                                    0
LIGHTCOND_Daylight                                                0
LIGHTCOND_Dusk                                                    0
UNDERINFL_N                                                       0
UNDERINFL_Y                                                       0
HITPARKEDCAR_N                                                    0
HITPARKEDCAR_Y                                                    0
dtype: int64
```

Correlation Matrix

```
In [27]: plt.rcParams["figure.figsize"] = (18,16)
         corr = Feature.corr()
         plt.rc('xtick',labelsize=10)
         plt.rc('ytick',labelsize=10)
         sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns)
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1bd99563f88>
```



In [28]:

```
Feature.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 960 entries, 16061 to 121419
Data columns (total 41 columns):
X                                                                 960 non-nul
l float64
Y                                                                 960 non-nul
l float64
PEDCOUNT                                                          960 non-nul
l int64
PEDCYLCOUNT                                                       960 non-nul
l int64
VEHCOUNT                                                          960 non-nul
l int64
ADDRTYPE_Block                                                    960 non-nul
l uint8
ADDRTYPE_Intersection                                             960 non-nul
l uint8
COLLISIONTYPE_Angles                                              960 non-nul
l uint8
COLLISIONTYPE_Cycles                                              960 non-nul
l uint8
COLLISIONTYPE_Head On                                             960 non-nul
l uint8
COLLISIONTYPE_Left Turn                                           960 non-nul
l uint8
COLLISIONTYPE_Parked Car                                          960 non-nul
l uint8
COLLISIONTYPE_Pedestrian                                          960 non-nul
l uint8
COLLISIONTYPE_Rear Ended                                          960 non-nul
l uint8
COLLISIONTYPE_Right Turn                                          960 non-nul
l uint8
COLLISIONTYPE_Sideswipe                                           960 non-nul
l uint8
JUNCTIONTYPE_At Intersection (but not related to intersection)    960 non-nul
l uint8
JUNCTIONTYPE_At Intersection (intersection related)               960 non-nul
l uint8
JUNCTIONTYPE_Driveway Junction                                    960 non-nul
l uint8
JUNCTIONTYPE_Mid-Block (but intersection related)                 960 non-nul
l uint8
JUNCTIONTYPE_Mid-Block (not related to intersection)              960 non-nul
l uint8
WEATHER_Clear                                                     960 non-nul
l uint8
WEATHER_Fog/Smog/Smoke                                            960 non-nul
l uint8
WEATHER_Overcast                                                  960 non-nul
l uint8
WEATHER_Raining                                                   960 non-nul
l uint8
WEATHER_Severe Crosswind                                          960 non-nul
l uint8
WEATHER_Snowing                                                   960 non-nul
l uint8
```

```
ROADCOND_Dry                                                      960 non-nul
l uint8
ROADCOND_Ice                                                      960 non-nul
l uint8
ROADCOND_Snow/Slush                                               960 non-nul
l uint8
ROADCOND_Wet                                                      960 non-nul
l uint8
LIGHTCOND_Dark - No Street Lights                                 960 non-nul
l uint8
LIGHTCOND_Dark - Street Lights Off                                960 non-nul
l uint8
LIGHTCOND_Dark - Street Lights On                                 960 non-nul
l uint8
LIGHTCOND_Dawn                                                    960 non-nul
l uint8
LIGHTCOND_Daylight                                                960 non-nul
l uint8
LIGHTCOND_Dusk                                                    960 non-nul
l uint8
UNDERINFL_N                                                       960 non-nul
l uint8
UNDERINFL_Y                                                       960 non-nul
l uint8
HITPARKEDCAR_N                                                    960 non-nul
l uint8
HITPARKEDCAR_Y                                                    960 non-nul
l uint8
dtypes: float64(2), int64(3), uint8(36)
memory usage: 78.8 KB
```

Normalizing and Feature Scaling

In [29]:

```
from sklearn import preprocessing
X = preprocessing.StandardScaler().fit(Feature).transform(Feature)

#Binarise SEVERITY code
Y = resampled_df["SEVERITYCODE"].apply(lambda x: 1 if (x>2)  else 0)
```

Split Train and Test Set

In [30]:

```
# We split X and Y into train and test subsets
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, rando
m_state=42)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

```
Train set: (672, 41) (672,)
Test set: (288, 41) (288,)
```

Classification:

Decision Tree

In [31]:

```
from sklearn.tree import DecisionTreeClassifier
DT_model = DecisionTreeClassifier(criterion="entropy", max_depth = 10)
DT_model.fit(X_train,Y_train)


#Prediction
DT_yhat = DT_model.predict(X_test)

#Model evaluation
from sklearn import metrics
from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import f1_score
from sklearn.metrics import log_loss
from sklearn.metrics import r2_score
from sklearn.metrics import classification_report, confusion_matrix

print("Accuracy of Decision Tree model:")
print("Train set Accuracy: ", metrics.accuracy_score(Y_train, DT_model.predict
(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(Y_test, DT_yhat))
print("Jaccard index: %.2f" % jaccard_similarity_score(Y_test, DT_yhat))
print("F1-score: %.2f" % f1_score(Y_test, DT_yhat, average='weighted') )
print("R2-score: %.2f" % r2_score(DT_yhat , Y_test) )
print(classification_report(Y_test, DT_yhat))
```

```
Accuracy of Decision Tree model:
Train set Accuracy:  0.8705357142857143
Test set Accuracy:  0.7048611111111112
Jaccard index: 0.70
F1-score: 0.70
R2-score: -0.23
              precision    recall  f1-score   support

           0       0.66      0.82      0.73       139
           1       0.78      0.60      0.68       149

    accuracy                           0.70       288
   macro avg       0.72      0.71      0.70       288
weighted avg       0.72      0.70      0.70       288


C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\sklearn\metrics\classification.py:635: DeprecationWarning: jaccard_simil
arity_score has been deprecated and replaced with jaccard_score. It will be r
emoved in version 0.23. This implementation has surprising behavior for binar
y and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

Visualize Decision Tree

In [32]:

```
from sklearn import tree
text_representation = tree.export_text(DT_model)
print(text_representation)
```

```
|--- feature_4 <= -0.40
|   |--- feature_3 <= 1.40
|   |   |--- feature_20 <= 0.30
|   |   |   |--- feature_1 <= -0.29
|   |   |   |   |--- feature_0 <= 1.52
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_0 >  1.52
|   |   |   |   |   |--- feature_0 <= 1.71
|   |   |   |   |   |   |--- feature_23 <= 1.00
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_23 >  1.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_0 >  1.71
|   |   |   |   |   |   |--- class: 1
|   |   |   |--- feature_1 >  -0.29
|   |   |   |   |--- feature_1 <= -0.26
|   |   |   |   |   |--- feature_35 <= -0.24
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_35 >  -0.24
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_1 >  -0.26
|   |   |   |   |   |--- feature_1 <= -0.22
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_1 >  -0.22
|   |   |   |   |   |   |--- feature_1 <= 1.39
|   |   |   |   |   |   |   |--- feature_29 <= 8.90
|   |   |   |   |   |   |   |   |--- feature_1 <= -0.21
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  -0.21
|   |   |   |   |   |   |   |   |   |--- feature_38 <= 1.17
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_38 >  1.17
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_29 >  8.90
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_1 >  1.39
|   |   |   |   |   |   |   |--- class: 1
|   |   |--- feature_20 >  0.30
|   |   |   |--- class: 1
|   |--- feature_3 >  1.40
|   |   |--- feature_0 <= -1.13
|   |   |   |--- class: 1
|   |   |--- feature_0 >  -1.13
|   |   |   |--- feature_0 <= -0.89
|   |   |   |   |--- class: 0
|   |   |   |--- feature_0 >  -0.89
|   |   |   |   |--- feature_16 <= 5.09
|   |   |   |   |   |--- feature_31 <= 5.41
|   |   |   |   |   |   |--- feature_0 <= 0.45
|   |   |   |   |   |   |   |--- feature_1 <= 0.84
|   |   |   |   |   |   |   |   |--- feature_2 <= 0.43
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_2 >  0.43
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 0.07
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_0 >  0.07
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_1 >  0.84
|   |   |   |   |   |   |   |   |--- feature_1 <= 1.19
|   |   |   |   |   |   |   |   |   |--- feature_23 <= 1.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_23 >  1.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  1.19
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_0 >  0.45
|   |   |   |   |   |   |   |--- feature_1 <= -0.33
|   |   |   |   |   |   |   |   |--- feature_1 <= -0.44
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 1.51
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_0 >  1.51
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  -0.44
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  -0.33
|   |   |   |   |   |   |   |   |--- feature_0 <= 1.08
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_0 >  1.08
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 0.92
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  0.92
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_31 >  5.41
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_16 >  5.09
|   |   |   |   |   |--- class: 0
|--- feature_4 >  -0.40
|   |--- feature_9 <= 2.60
|   |   |--- feature_2 <= 0.43
|   |   |   |--- feature_37 <= -1.17
|   |   |   |   |--- feature_0 <= 0.76
|   |   |   |   |   |--- feature_15 <= 1.63
|   |   |   |   |   |   |--- feature_20 <= 0.30
|   |   |   |   |   |   |   |--- feature_1 <= 0.01
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  0.01
|   |   |   |   |   |   |   |   |--- feature_1 <= 1.55
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  1.55
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_20 >  0.30
|   |   |   |   |   |   |   |--- feature_1 <= -1.13
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_1 >  -1.13
|   |   |   |   |   |   |   |   |--- feature_4 <= 2.33
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_4 >  2.33
|   |   |   |   |   |   |   |   |   |--- feature_33 <= 0.39
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_33 >  0.39
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_15 >  1.63
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_0 >  0.76
|   |   |   |   |   |--- feature_0 <= 1.24
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_0 >  1.24
|   |   |   |   |   |   |--- feature_20 <= 0.30
|   |   |   |   |   |   |   |--- feature_23 <= 1.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_23 >  1.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_20 >  0.30
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- feature_37 >  -1.17
|   |   |   |   |--- feature_17 <= -0.02
|   |   |   |   |   |--- feature_1 <= 1.99
|   |   |   |   |   |   |--- feature_0 <= -1.41
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_0 >  -1.41
|   |   |   |   |   |   |   |--- feature_21 <= -0.32
|   |   |   |   |   |   |   |   |--- feature_0 <= -0.44
|   |   |   |   |   |   |   |   |   |--- feature_13 <= 0.97
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_13 >  0.97
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_0 >  -0.44
|   |   |   |   |   |   |   |   |   |--- feature_1 <= -0.19
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  -0.19
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_21 >  -0.32
|   |   |   |   |   |   |   |   |--- feature_0 <= 1.87
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 1.09
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_0 >  1.09
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_0 >  1.87
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_1 >  1.99
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_17 >  -0.02
|   |   |   |   |   |--- feature_15 <= 1.63
|   |   |   |   |   |   |--- feature_0 <= 1.95
|   |   |   |   |   |   |   |--- feature_0 <= -0.09
|   |   |   |   |   |   |   |   |--- feature_0 <= -0.11
|   |   |   |   |   |   |   |   |   |--- feature_27 <= -0.56
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_27 >  -0.56
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_0 >  -0.11
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_0 >  -0.09
|   |   |   |   |   |   |   |   |--- feature_4 <= 0.96
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 1.29
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_0 >  1.29
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_4 >  0.96
|   |   |   |   |   |   |   |   |   |--- feature_30 <= 0.58
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_30 >  0.58
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_0 >  1.95
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_15 >  1.63
|   |   |   |   |   |   |--- class: 0
|   |   |--- feature_2 >  0.43
|   |   |   |--- class: 1
|   |--- feature_9 >  2.60
|   |   |--- feature_20 <= 0.30
|   |   |   |--- feature_27 <= -0.56
|   |   |   |   |--- class: 1
|   |   |   |--- feature_27 >  -0.56
|   |   |   |   |--- class: 0
|   |   |--- feature_20 >  0.30
|   |   |   |--- feature_4 <= 2.33
|   |   |   |   |--- class: 1
|   |   |   |--- feature_4 >  2.33
|   |   |   |   |--- feature_37 <= -1.17
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_37 >  -1.17
|   |   |   |   |   |--- class: 1
```

Random Forest

In [33]:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, r2_score, classi
fication_report
from sklearn.model_selection import train_test_split, GridSearchCV

rf = RandomForestClassifier()
params = {'n_estimators':[50,75,100],
          'criterion':['gini', 'entropy'],
          'random_state':[0]}
rf1 = GridSearchCV(rf, param_grid=params)
rf1.fit(X_train,Y_train)
rf_predictions = rf1.predict(X_test)
print('Best Hyperparameter RFT : ',rf1.best_params_)

#Confusion Matrix
rf_cm=confusion_matrix(Y_test,rf_predictions)
print(rf_cm,'\n')

#Classification Report
rf_cr = classification_report(Y_test,rf_predictions)
print(rf_cr,'\n')

#Accuracy
acc = accuracy_score(Y_test,rf_predictions)
print(acc,'\n')
```

```
C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\sklearn\model_selection\_split.py:1978: FutureWarning: The default value
of cv will change from 3 to 5 in version 0.22. Specify it explicitly to silen
ce this warning.
  warnings.warn(CV_WARNING, FutureWarning)

Best Hyperparameter RFT :  {'criterion': 'gini', 'n_estimators': 100, 'random
_state': 0}
[[102  37]
 [ 48 101]]

              precision    recall  f1-score   support

           0       0.68      0.73      0.71       139
           1       0.73      0.68      0.70       149

    accuracy                           0.70       288
   macro avg       0.71      0.71      0.70       288
weighted avg       0.71      0.70      0.70       288


0.7048611111111112
```

K Nearest Neighbour

In [34]:

```python
from sklearn.neighbors import KNeighborsClassifier
#Fitting and Predictions
knn = KNeighborsClassifier()
params = {'n_neighbors':[3,4,5,6,7],
          'p':[1,2]}
knn1 = GridSearchCV(knn, param_grid=params)
knn1.fit(X_train,Y_train.values.ravel())
knn_predictions = knn1.predict(X_test)

print('Best Hyperparameter KNN : ',knn1.best_params_)

#Confusion Matrix
knn_cm = confusion_matrix(Y_test,knn_predictions)
print(knn_cm,'\n')

#Classification Report
knn_cr = classification_report(Y_test,knn_predictions)
print(knn_cr,'\n')

#Accuracy
acc = accuracy_score(Y_test,knn_predictions)
print(acc,'\n')
```

```
C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\sklearn\model_selection\_split.py:1978: FutureWarning: The default value
of cv will change from 3 to 5 in version 0.22. Specify it explicitly to silen
ce this warning.
  warnings.warn(CV_WARNING, FutureWarning)

Best Hyperparameter KNN :  {'n_neighbors': 5, 'p': 2}
[[106  33]
 [ 46 103]]

              precision    recall  f1-score   support

           0       0.70      0.76      0.73       139
           1       0.76      0.69      0.72       149

    accuracy                           0.73       288
   macro avg       0.73      0.73      0.73       288
weighted avg       0.73      0.73      0.73       288


0.7256944444444444
```

Logistic Regression

In [35]:

```
from sklearn.linear_model import LogisticRegression
LR_model = LogisticRegression(C=0.01).fit(X_train,Y_train)


LR_yhat = LR_model.predict(X_test)

#Model evaluation
print("Accuracy of Logistic Regression model:")
print("Train set Accuracy: ", metrics.accuracy_score(Y_train, LR_model.predict
(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(Y_test, LR_yhat))
print("Jaccard index: %.2f" % jaccard_similarity_score(Y_test, LR_yhat))
print("F1-score: %.2f" % f1_score(Y_test, LR_yhat, average='weighted') )
print("R2-score: %.2f" % r2_score(LR_yhat , Y_test) )
print(classification_report(Y_test, LR_yhat))
```

```
Accuracy of Logistic Regression model:
Train set Accuracy:  0.7916666666666666
Test set Accuracy:  0.7465277777777778
Jaccard index: 0.75
F1-score: 0.74
R2-score: -0.05
              precision    recall  f1-score   support

           0       0.69      0.86      0.77       139
           1       0.83      0.64      0.72       149

    accuracy                           0.75       288
   macro avg       0.76      0.75      0.74       288
weighted avg       0.76      0.75      0.74       288


C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will
be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\sklearn\metrics\classification.py:635: DeprecationWarning: jaccard_simil
arity_score has been deprecated and replaced with jaccard_score. It will be r
emoved in version 0.23. This implementation has surprising behavior for binar
y and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

Support Vector Machine

In [36]:

```
from sklearn import svm
SVM_model = svm.SVC(kernel='linear')
SVM_model.fit(X_train, Y_train)

SVM_yhat = SVM_model.predict(X_test)

#Model evaluation
print("Accuracy of SVM model:")
print("Train set Accuracy: ", metrics.accuracy_score(Y_train, SVM_model.predict(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(Y_test, SVM_yhat))
print("Jaccard index: %.2f" % jaccard_similarity_score(Y_test, SVM_yhat))
print("F1-score: %.2f" % f1_score(Y_test, SVM_yhat, average='weighted') )
print("R2-score: %.2f" % r2_score(SVM_yhat , Y_test) )
print(classification_report(Y_test, SVM_yhat))
```

```
Accuracy of SVM model:

C:\Users\ManojKumar Chalamala\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\sklearn\metrics\classification.py:635: DeprecationWarning: jaccard_simil
arity_score has been deprecated and replaced with jaccard_score. It will be r
emoved in version 0.23. This implementation has surprising behavior for binar
y and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)

Train set Accuracy:  0.7991071428571429
Test set Accuracy:  0.7465277777777778
Jaccard index: 0.75
F1-score: 0.75
R2-score: -0.03
              precision    recall  f1-score   support

           0       0.70      0.83      0.76       139
           1       0.81      0.67      0.73       149

    accuracy                           0.75       288
   macro avg       0.75      0.75      0.75       288
weighted avg       0.76      0.75      0.75       288
```

Conclusion:

Although both has similar accuracy we prefer SVM, as it has key advantage of being able to return a ranked list of the most significant features in terms of their influence on the accident severity code (provided a linear mapping kernel is used).

The SVM model highlights that accidents involving pedestrians and multiple vehicles often have severe consequences, as do those in which excess speed is a factor. By identifying the ranking of the major causes of accident severity in this manner, it is hoped that town/city planners will be able to design new road infrastructure and target the introduction of traffic calming measures where they are most needed.