IBM Data science Capstone project

Predicting Seattle car accident severity

Renuka Devi Ulaganathan

October 22, 2020

Introduction:

With the prevailing traffic conditions around the globe, there is a need for an alert system that would predict the possibility of an accident based on given conditions. Using machine learning techniques, a model can be built and trained from the past collision data available. The model thus trained shall be used to predict the occurrence of a collision and its severity and alert the parties involved.

1. Business understanding:

The Seattle government, in an effort to reduce the number of car collisions, wants to reduce the car accidents in Seattle. To implement this, a model to be developed to predict the possibility of car accidents given factors like accident spot, driving speed,weather, road, light conditions etc. This model will help in warning the local Seattle government, traffic police and the drivers on the targeted roads that will help in reducing the frequency of accidents.

The target audience of this project is Seattle government, traffic police department,city traffic surveillance team, car drivers and the local residents in the neighbourhood.

2. Data understanding:

2.1 Source of the data:

The collision data used for this project is taken from Seattle Department of Transport (SDOT) provided in capstone project introduction. The link for the data is here (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv). The metadata about the dataset can be found here (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf).

The data is be downloaded from the portal and saved into a python dataframe for further analytics.

```
--2020-10-22 17:04:07--  https://s3.us.cloud-object-storage.appdomain.cloud/cf
-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv
Resolving s3.us.cloud-object-storage.appdomain.cloud (s3.us.cloud-object-stora
ge.appdomain.cloud)... 67.228.254.196
Connecting to s3.us.cloud-object-storage.appdomain.cloud (s3.us.cloud-object-s
torage.appdomain.cloud)|67.228.254.196|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 73917638 (70M) [text/csv]
Saving to: 'collision_data.csv'

collision_data.csv  100%[===================>]  70.49M  17.9MB/s    in 4.2s

2020-10-22 17:04:12 (16.9 MB/s) - 'collision_data.csv' saved [73917638/7391763
8]

Dataset downloaded successfully

/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/IPython/core/in
teractiveshell.py:3072: DtypeWarning: Columns (33) have mixed types.Specify dt
ype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STAT
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matc
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matc
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matc
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matc
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matc

5 rows × 38 columns

Taking a look at the collision dataset, we have **194,673 rows** and **38 columns.**

```
The number of columns is :  38
The number of (rows,columns) is :  (194673, 38)
The size of the dataset is : 7397574
The data types of different columns are below :


SEVERITYCODE        int64
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.1      int64
SEVERITYDESC       object
COLLISIONTYPE      object
PERSONCOUNT         int64
PEDCOUNT            int64
PEDCYLCOUNT         int64
VEHCOUNT            int64
INCDATE            object
INCDTTM            object
JUNCTIONTYPE       object
SDOT_COLCODE        int64
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
```

Taking a back up of the dataframe for any reference down the line, as we will now proceed with data cleaning

The target or dependent variable is **'SEVERITYCODE'** which determines the severity of the accident. Below are the possible values for this variable. However, the dataset provided in the capstone project has only **'1'** and **'2' severity codes**. so, our machine learning technique will apply on the provided data source.

Severity codes:

0: Little to no Probability (Clear Conditions)

1: Very Low Probability — Chance or Property Damage

2: Low Probability — Chance of Injury

3: Mild Probability — Chance of Serious Injury

4: High Probability — Chance of Fatality

Out of the 38 columns present in the dataset, it is necessary to determine the predictor or independent variables that would influence the severity of the accident. Observing the data, the variables **'ADDRTYPE','WEATHER','ROADCOND','LIGHTCOND','UNDERINFL', 'COLLISIONTYPE','JUNCTIONTYPE' and 'SPEEDING'** seem to more likely to impact the accident severity. Let us plot these variables against severity code and observe relationship between them.

Identify the unique values for each of these variables:

```
The frequency of address type is:

Block           126926
Intersection     65070
Alley              751
Name: ADDRTYPE, dtype: int64

The frequency of weather conditions is:

Clear                       111135
Raining                      33145
Overcast                     27714
Unknown                      15091
Other                         5913
Snowing                        907
Fog/Smog/Smoke                 569
Sleet/Hail/Freezing Rain       113
Blowing Sand/Dirt               56
Severe Crosswind                25
Partly Cloudy                    5
Name: WEATHER, dtype: int64

The frequency of road conditions is:


Dry               124510
Wet                47474
Unknown            15078
Other               5144
Ice                 1209
Snow/Slush          1004
Standing Water       115
Sand/Mud/Dirt         75
Oil                   64
Name: ROADCOND, dtype: int64
```

```
The frequency of light conditions is:


Daylight                    116137
Dark - Street Lights On      48507
Unknown                      13473
Dusk                          5902
Other                         5405
Dawn                          2502
Dark - No Street Lights       1537
Dark - Street Lights Off      1199
Dark - Unknown Lighting         11
Name: LIGHTCOND, dtype: int64

The distinct 'under influence' values and its respective counts are:

N    105158
0     80394
Y      5126
1      3995
Name: UNDERINFL, dtype: int64

The distinct speeding values and its respective counts are:

N    185340
Y      9333
Name: SPEEDING, dtype: int64

The frequency of collision types is:

Parked Car    47987
Angles        34674
Rear Ended    34090
Other         28607
Sideswipe     18609
Left Turn     13703
Pedestrian     6608
Cycles         5415
Right Turn     2956
Head On        2024
Name: COLLISIONTYPE, dtype: int64

The frequency of junction types is:

Mid-Block (not related to intersection)              89800
At Intersection (intersection related)               62810
Mid-Block (but intersection related)                 22790
Driveway Junction                                    10671
Other                                                 6329
At Intersection (but not related to intersection)     2098
Ramp Junction                                          166
Unknown                                                  9
Name: JUNCTIONTYPE, dtype: int64
```

2.2 Data Cleaning:

```
The total number of null/NaN values for each variable is as follows:

 SEVERITYCODE           0
X                  5334
Y                  5334
OBJECTID              0
INCKEY                0
COLDETKEY             0
REPORTNO              0
STATUS                0
ADDRTYPE           1926
INTKEY           129603
LOCATION           2677
EXCEPTRSNCODE    109862
EXCEPTRSNDESC    189035
SEVERITYCODE.1        0
SEVERITYDESC          0
COLLISIONTYPE      4904
PERSONCOUNT           0
PEDCOUNT              0
PEDCYLCOUNT           0
VEHCOUNT              0
INCDATE               0
INCDTTM               0
JUNCTIONTYPE       6329
SDOT_COLCODE          0
SDOT_COLDESC          0
INATTENTIONIND   164868
UNDERINFL          4884
WEATHER            5081
ROADCOND           5012
LIGHTCOND          5170
PEDROWNOTGRNT    190006
SDOTCOLNUM        79737
SPEEDING         185340
ST_COLCODE           18
ST_COLDESC         4904
SEGLANEKEY            0
CROSSWALKKEY          0
HITPARKEDCAR          0
dtype: int64

Let us replace the null or NaN values of SPEEDING to N, null, NaN and 0 values
of UNDERINFL to N and all other null values of predictor variables to 'other'
```

Now, Let us check if there are still any null values for these independent variables.

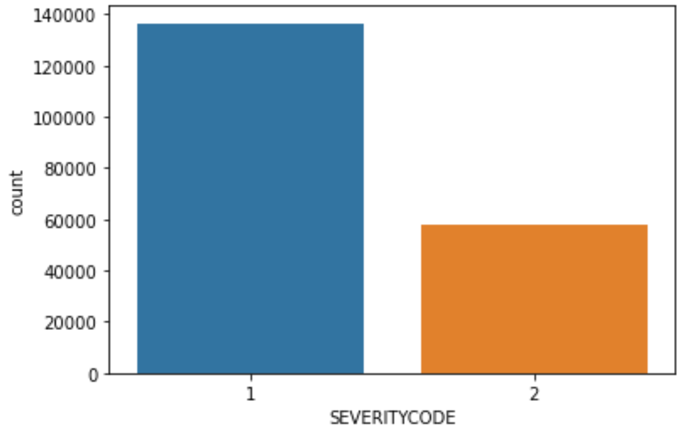
```
SEVERITYCODE           0
X                   5334
Y                   5334
OBJECTID               0
INCKEY                 0
COLDETKEY              0
REPORTNO               0
STATUS                 0
ADDRTYPE               0
INTKEY            129603
LOCATION            2677
EXCEPTRSNCODE     109862
EXCEPTRSNDESC     189035
SEVERITYCODE.1         0
SEVERITYDESC           0
COLLISIONTYPE          0
PERSONCOUNT            0
PEDCOUNT               0
PEDCYLCOUNT            0
VEHCOUNT               0
INCDATE                0
INCDTTM                0
JUNCTIONTYPE           0
SDOT_COLCODE           0
SDOT_COLDESC           0
INATTENTIONIND    164868
UNDERINFL              0
WEATHER                0
ROADCOND               0
LIGHTCOND              0
PEDROWNOTGRNT     190006
SDOTCOLNUM         79737
SPEEDING               0
ST_COLCODE            18
ST_COLDESC          4904
SEGLANEKEY             0
CROSSWALKKEY           0
HITPARKEDCAR           0
dtype: int64
```

Good! We have got rid off Nan and NULL values for the independent variables.

2.3 Understanding of Target variable:

```
The severity codes by count is as follows:
 1    136485
2     58188
Name: SEVERITYCODE, dtype: int64
```

```
<AxesSubplot:xlabel='SEVERITYCODE', ylabel='count'>
```



2.3.1 Downsampling and Balancing the dataset:

As we see, the severity code '1'(property damage) is more than doubles the time of severity code '2'(Injury). Hence the dataset is highly unbalanced. It will be difficult to predict or apply machine learning algorithms on unbalanced dataset. Hence, I shall downsample the dataset to balance it.

Let us take these variables to understand what kind of impact it has on SEVERITYCODE.

```
The dataset is now balanced and can be used for machine learning. Severity cod
e frequency is below.
```

```
2    58188
1    58188
Name: SEVERITYCODE, dtype: int64
```

2.3.2 Feature selection:

As part of feature selection, let us drop the rest of the fields that are not part of independent variable list.

```
The variables part of dataframe are listed below:
```

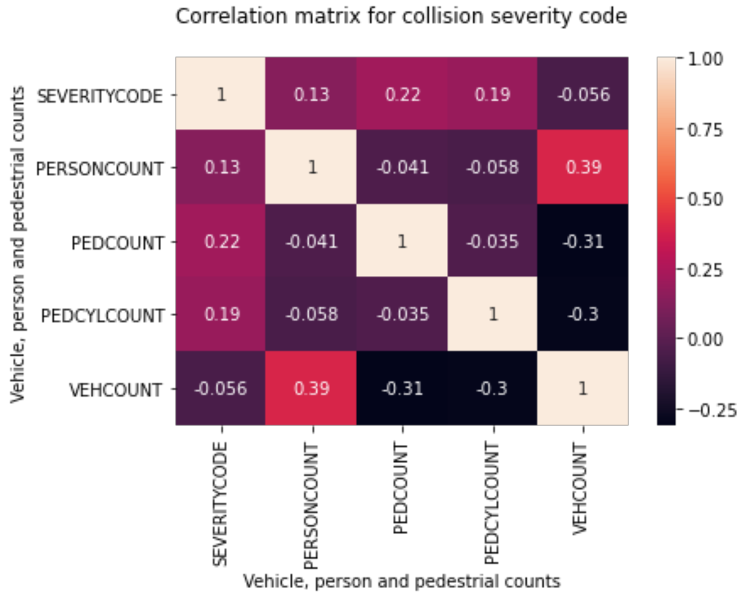
	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	PEDCOUNT	PEDCYLC
25055	1	Intersection	Angles	2	0	0
65280	1	Intersection	Angles	2	0	0
86292	1	Intersection	Angles	2	0	0
155111	1	Block	Sideswipe	2	0	0
64598	1	Block	Head On	3	0	0

```
SEVERITYCODE       int64
ADDRTYPE          object
COLLISIONTYPE     object
PERSONCOUNT        int64
PEDCOUNT           int64
PEDCYLCOUNT        int64
VEHCOUNT           int64
JUNCTIONTYPE      object
UNDERINFL         object
WEATHER           object
ROADCOND          object
LIGHTCOND         object
SPEEDING          object
dtype: object
```

As a next step, want to understand if the variables **PERSONCOUNT,PEDCOUNT,PEDCYLCOUNT and VEHCOUNT** has impact on the severity code before doing **dimension reduction** . Let us draw the **correlation matrix** with these numerical data to analyze further.

2.3.3 Correlation Matrix of numeric variables:

```
Text(32.99999999999999, 0.5, 'Vehicle, person and pedestrial counts')
```



From the correlation matrix above, we can see that the correlation coefficient is less than 0.5 for all the variables mapped and hence none of them have stronger relation between them. Comapring the variables against SEVERITYCODE, the variables **PERSONCOUNT,PEDCOUNT,PEDCYLCOUNT and VEHCOUNT** have **weaker relationship** with **severity code.** Hence these can be dropped from the dataframe.

```
Dropping PERSONCOUNT,PEDCOUNT,PEDCYLCOUNT,VEHCOUNT from the Data Frame
```

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	JUNCTIONTYPE	UNDERINFL	WEATHEF
25055	1	Intersection	Angles	At Intersection (intersection related)	0	Raining
65280	1	Intersection	Angles	At Intersection (intersection related)	0	Clear
86292	1	Intersection	Angles	At Intersection (intersection related)	N	Unknown
155111	1	Block	Sideswipe	Mid-Block (not related to intersection)	N	Clear
64598	1	Block	Head On	Mid-Block (not related to intersection)	0	Clear

```
SEVERITYCODE       int64
ADDRTYPE          object
COLLISIONTYPE     object
JUNCTIONTYPE      object
UNDERINFL         object
WEATHER           object
ROADCOND          object
LIGHTCOND         object
SPEEDING          object
dtype: object
```

3. Exploratory data analysis:

The variable **UNDERINFL** currently has values **'0','N','Y' and '1'**. To keep it unform for machine learning, let us map 'N' to '0'and Y to '1'.

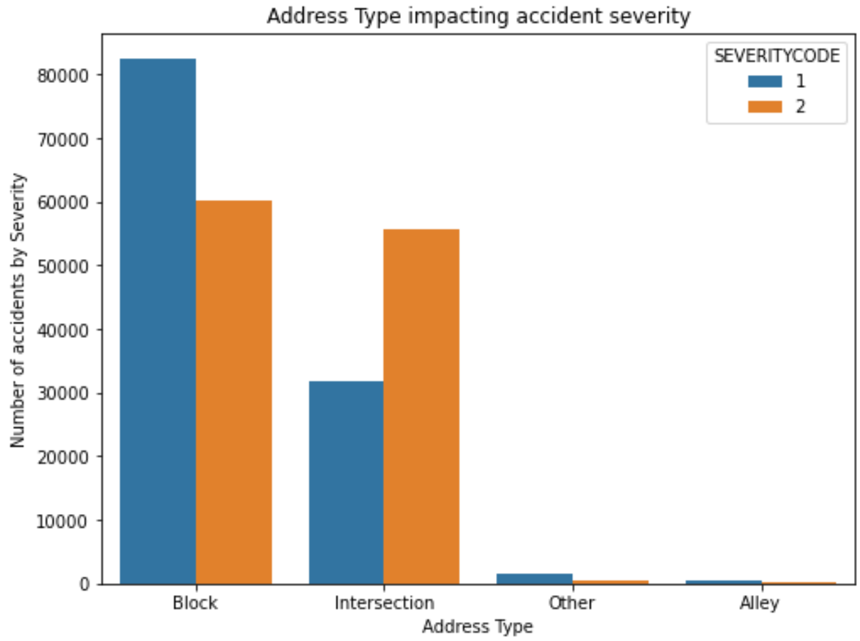
```
The values variable UNDERINFL is mapped to values 0 and 1
```

3.1 Relationship between ADDRTYPE and SEVERITYCODE:

	ADDRTYPE	SEVERITYCODE	ADDRCOUNT
2	Block	1	82450
3	Block	2	60192
5	Intersection	2	55638
4	Intersection	1	31862
6	Other	1	1462
0	Alley	1	602
7	Other	2	382
1	Alley	2	164

3.1.1 ADDRCOUNT vs SEVERITYCODE - Data Visualization

```
Text(0, 0.5, 'Number of accidents by Severity')
```



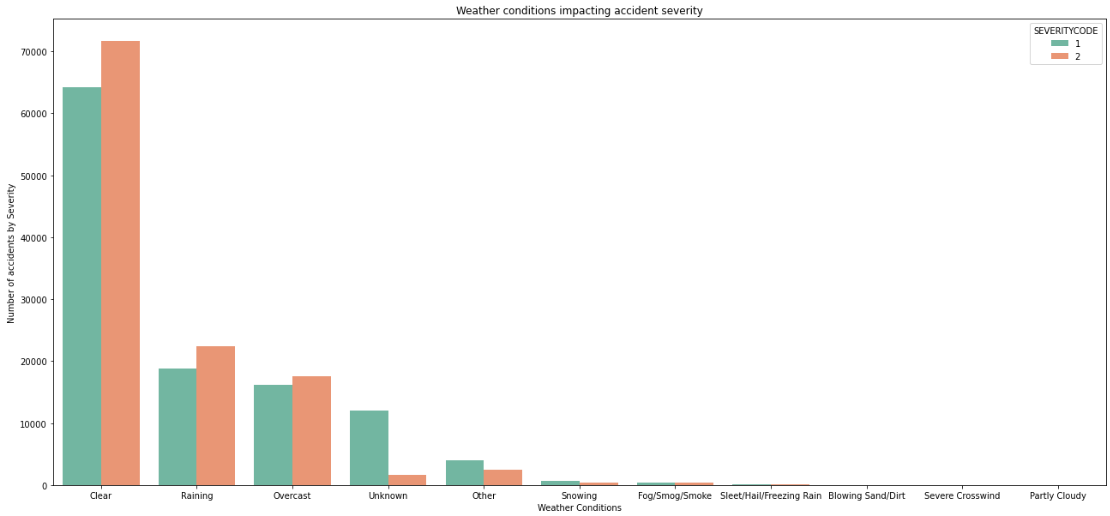
From the above, we can see that **ADDRCOUNT** has strong relationship with SEVERITYCODE influencing the accident severity. Hence, it should be considered as part of the independent variables to predict SEVERITYCODE.

3.2 Relationship between WEATHER and SEVERITYCODE:

	WEATHER	SEVERITYCODE	WEATHCOUNT
3	Clear	2	71680
2	Clear	1	64212
13	Raining	2	22352
12	Raining	1	18816
9	Overcast	2	17490
8	Overcast	1	16178
20	Unknown	1	12070
21	Unknown	2	1632
18	Snowing	1	606
6	Other	1	580
5	Fog/Smog/Smoke	2	374
19	Snowing	2	342
4	Fog/Smog/Smoke	1	336
7	Other	2	232
16	Sleet/Hail/Freezing Rain	1	70
17	Sleet/Hail/Freezing Rain	2	56
0	Blowing Sand/Dirt	1	30
1	Blowing Sand/Dirt	2	30
15	Severe Crosswind	2	14
14	Severe Crosswind	1	12
11	Partly Cloudy	2	6
10	Partly Cloudy	1	2

3.2.1 WEATHER vs SEVERITYCODE - Data Visualization

Text(0, 0.5, 'Number of accidents by Severity')



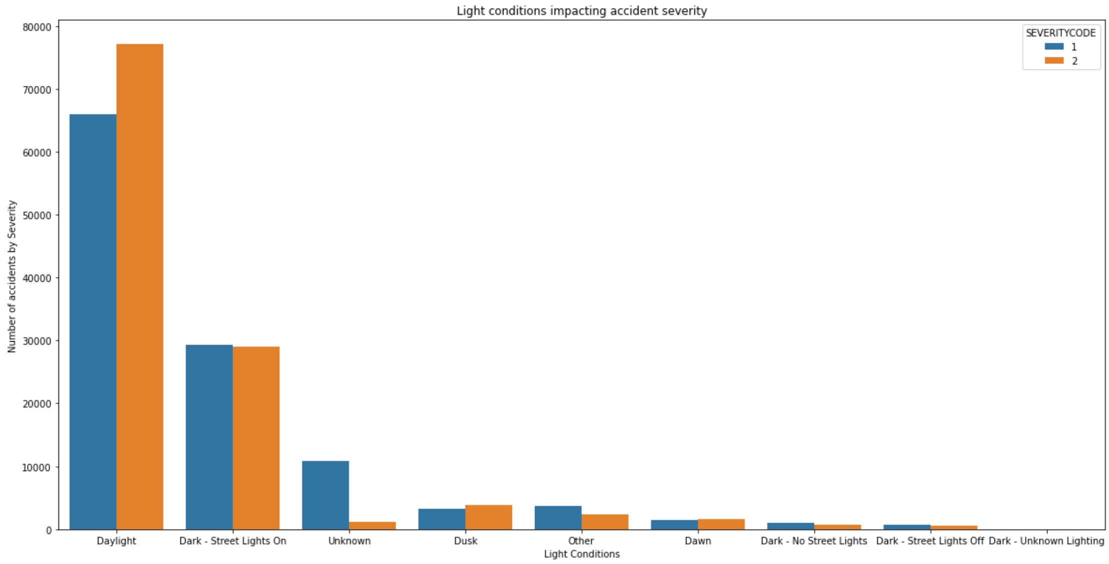
From the above, we can see that **WEATHER** has strong relationship with SEVERITYCODE influencing the accident severity. Hence, it should be considered as part of the independent variables to predict accident severity.

3.3 Relationship between LIGHTCOND and SEVERITYCODE:

	LIGHTCOND	SEVERITYCODE	LIGHTCOUNT
11	Daylight	2	77088
10	Daylight	1	65918
4	Dark - Street Lights On	1	29316
5	Dark - Street Lights On	2	28950
16	Unknown	1	10914
13	Dusk	2	3888
12	Dusk	1	3296
9	Dawn	2	1648
8	Dawn	1	1422
17	Unknown	2	1210
0	Dark - No Street Lights	1	1064
2	Dark - Street Lights Off	1	774
1	Dark - No Street Lights	2	668
3	Dark - Street Lights Off	2	632
14	Other	1	150
15	Other	2	104
7	Dark - Unknown Lighting	2	8
6	Dark - Unknown Lighting	1	6

3.3.1 LIGHTCOND vs SEVERITYCODE - Data Visualization

```
Text(0.5, 1.0, 'Light conditions impacting accident severity')
```



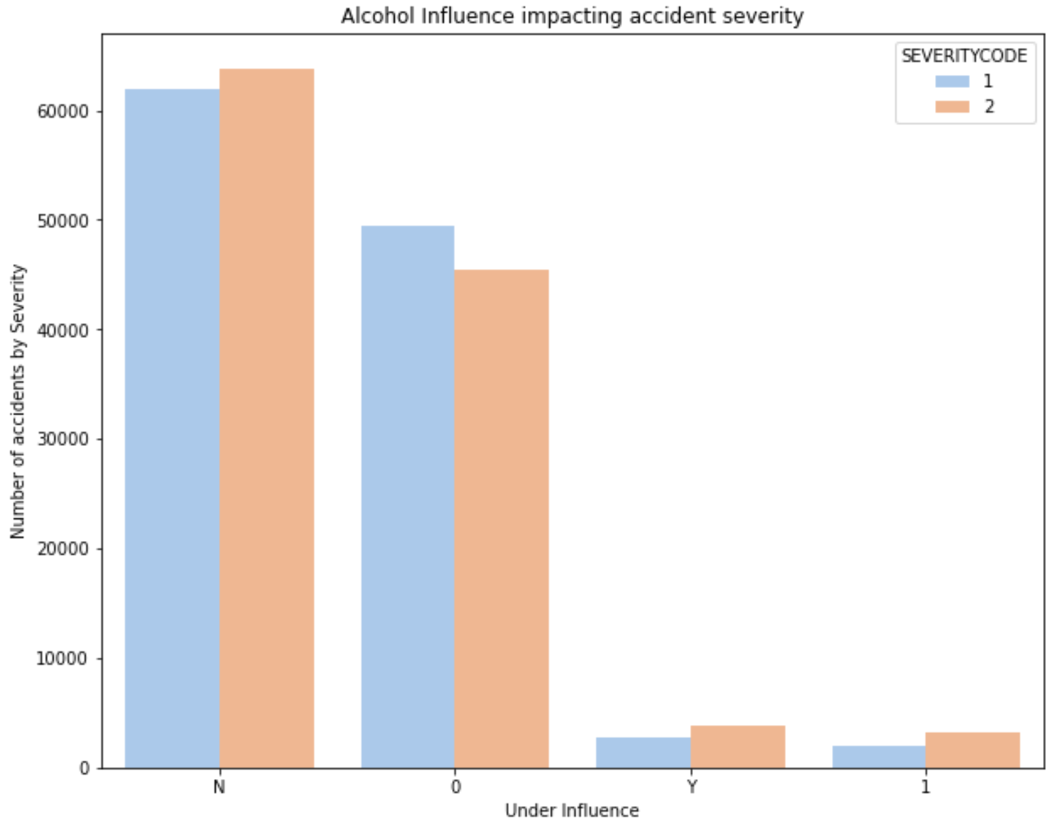
From the above, we can see that **LIGHTCOND** has strong relationship with SEVERITYCODE influencing the accident severity. The accident has resulted in injury mainly during daylight and dark when street lights are on. Hence, it should be considered as part of the independent variables to predict accident severity.

3.4 Relationship between UNDERINFL and SEVERITYCODE:

	UNDERINFL	SEVERITYCODE	INFLCOUNT
0	0	1	111524
1	0	2	109252
3	1	2	7124
2	1	1	4852

3.4.1 UNDRINFL vs SEVERITYCODE - Data Visualization

```
Text(0.5, 1.0, 'Alcohol Influence impacting accident severity')
```



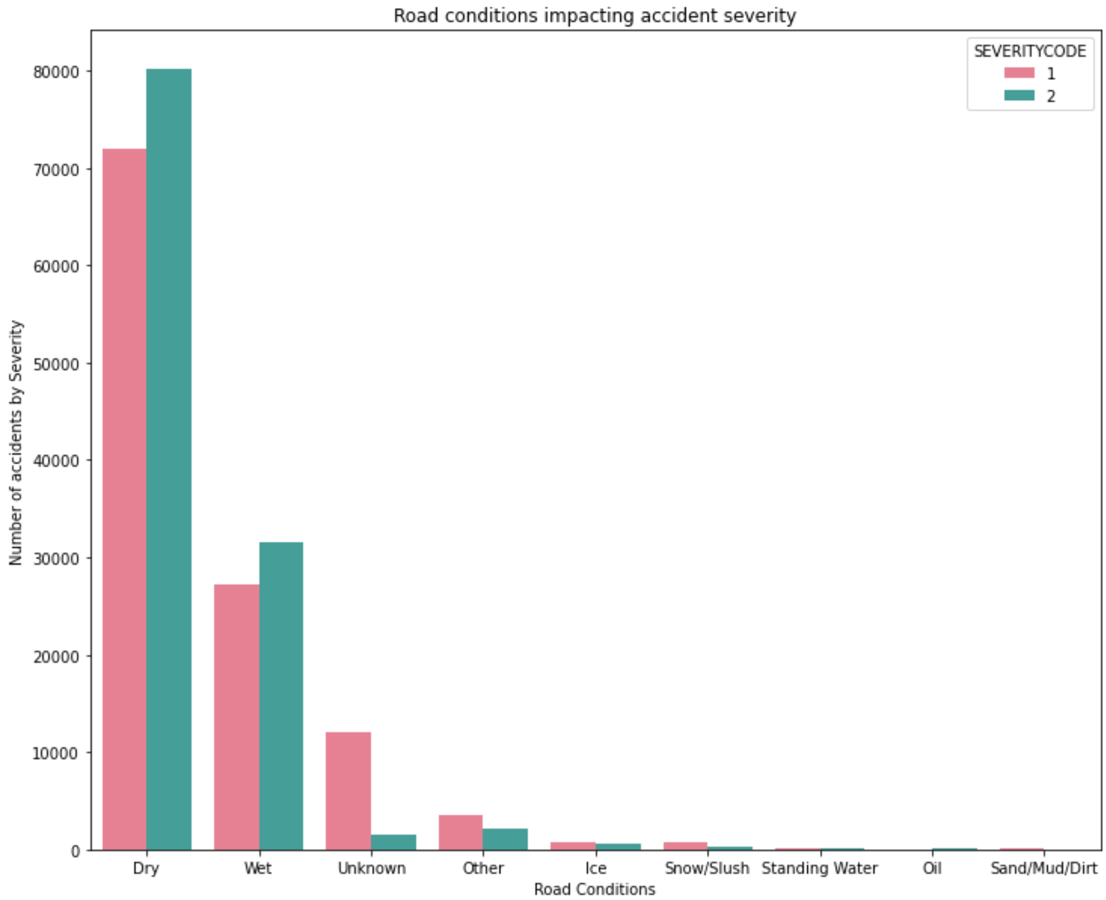
From the above, we can see that **UNDERINFL** has weaker relationship with SEVERITYCODE in influencing the accident severity. We can see that the accident has resulted in property damage or injury when the driver is not under alcohol influence. Hence, it should **not be considered** as part of the independent variables to predict accident severity.

3.5 Relationship between ROADCOND and SEVERITYCODE:

	ROADCOND	SEVERITYCODE	ROADCOUNT
1	Dry	2	80128
0	Dry	1	71872
17	Wet	2	31510
16	Wet	1	27238
14	Unknown	1	12114
15	Unknown	2	1498
2	Ice	1	810
10	Snow/Slush	1	722
3	Ice	2	546
11	Snow/Slush	2	334
7	Other	2	86
6	Other	1	80
13	Standing Water	2	60
12	Standing Water	1	54
5	Oil	2	48
8	Sand/Mud/Dirt	1	48
9	Sand/Mud/Dirt	2	46
4	Oil	1	30

3.5.1 ROADCOND vs SEVERITYCODE - Data Visualization

```
Text(0.5, 1.0, 'Road conditions impacting accident severity')
```



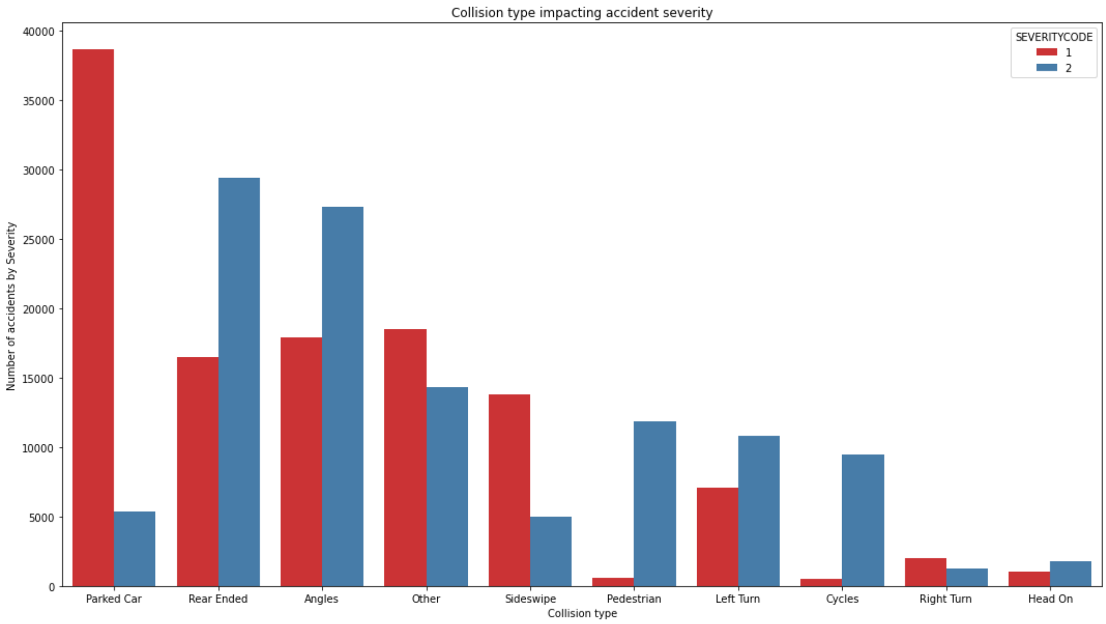
From the above, we can see that **ROADCOND** has strong relationship with SEVERITYCODE influencing the accident severity. The accident has resulted in injury mainly when the road was dry and wet. Hence, it should be considered as part of the independent variables to predict accident severity.

3.6 Relationship between COLLISIONTYPE and SEVERITYCODE:

	COLLISIONTYPE	SEVERITYCODE	COLLNCOUNT
10	Parked Car	1	38614
15	Rear Ended	2	29342
1	Angles	2	27248
8	Other	1	18450
0	Angles	1	17868
14	Rear Ended	1	16466
9	Other	2	14306
18	Sideswipe	1	13794
13	Pedestrian	2	11872
7	Left Turn	2	10822
3	Cycles	2	9488
6	Left Turn	1	7052
11	Parked Car	2	5324
19	Sideswipe	2	5012
16	Right Turn	1	1998
5	Head On	2	1744
17	Right Turn	2	1218
4	Head On	1	1018
12	Pedestrian	1	586
2	Cycles	1	530

3.6.1 COLLISIONTYPE vs SEVERITYCODE - Data Visualization

```
Text(0.5, 1.0, 'Collision type impacting accident severity')
```



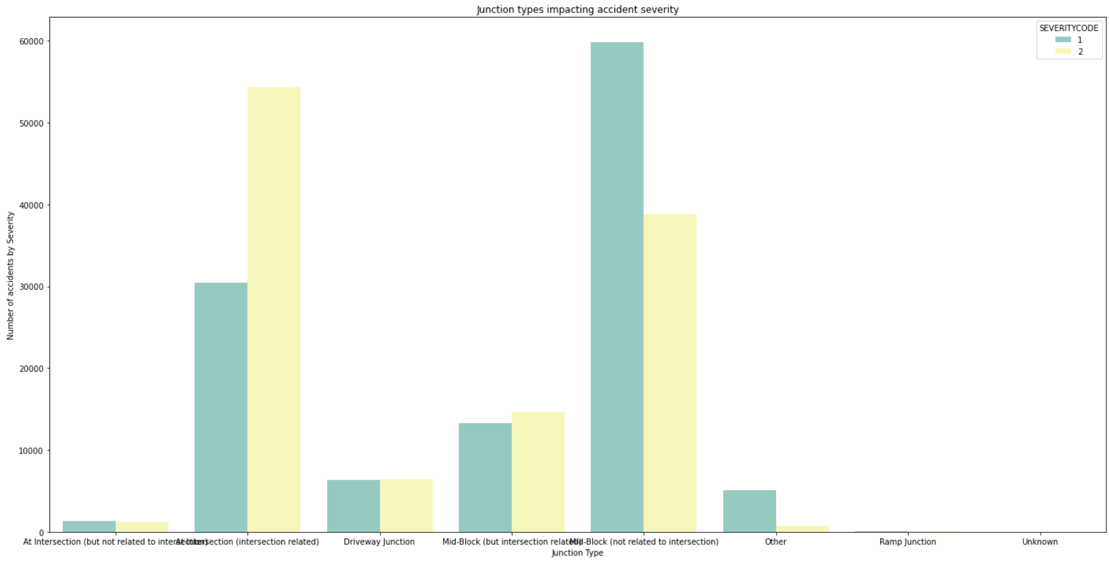
From the barplot, we can see that **COLLISIONTYPE** has strong relationship with SEVERITYCODE influencing the accident severity. Hence, it should be considered as part of the independent variables to predict accident severity.

3.7 Relationship between JUNCTIONTYPE and SEVERITYCODE:

	JUNCTIONTYPE	SEVERITYCODE	JUNCNCOUNT
0	At Intersection (but not related to intersection)	1	1290
1	At Intersection (but not related to intersection)	2	1246
2	At Intersection (intersection related)	1	30446
3	At Intersection (intersection related)	2	54348
4	Driveway Junction	1	6336
5	Driveway Junction	2	6468
6	Mid-Block (but intersection related)	1	13258
7	Mid-Block (but intersection related)	2	14594
8	Mid-Block (not related to intersection)	1	59890
9	Mid-Block (not related to intersection)	2	38808
10	Ramp Junction	1	80
11	Ramp Junction	2	108
12	Unknown	1	2
13	Unknown	2	4

3.7.1 JUNCTIONTYPE vs SEVERITYCODE - Data Visualization

```
Text(0.5, 1.0, 'Junction types impacting accident severity')
```



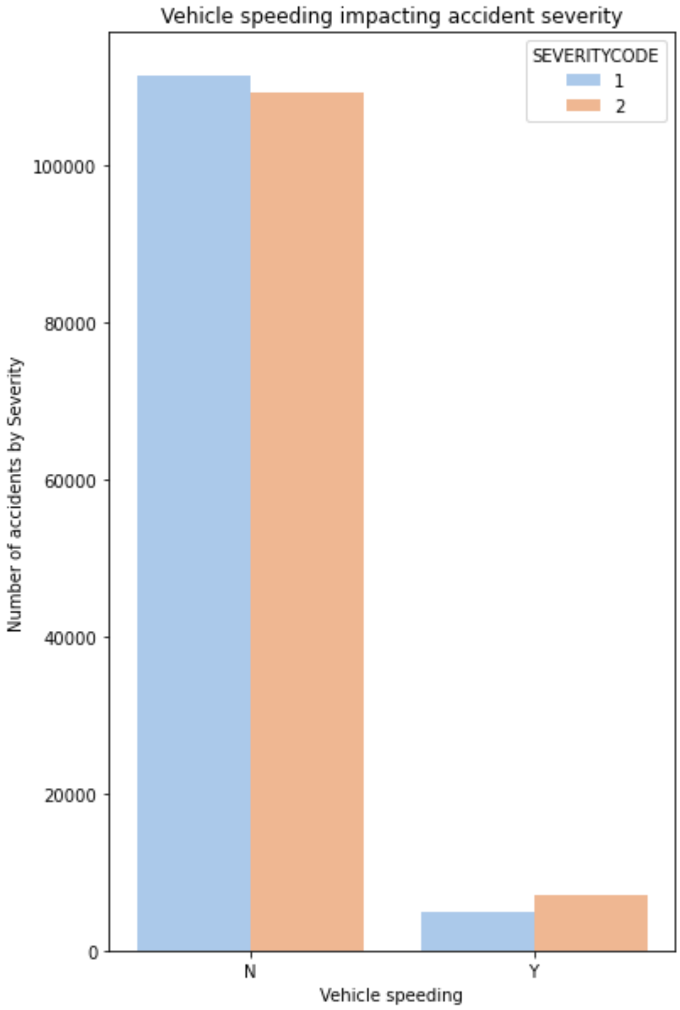
From the barplot, we can see that **JUNCTIONTYPE** has strong relationship with SEVERITYCODE influencing the accident severity. The accidents are more at midblock and intersection points. Hence, it should be considered as part of the independent variables to predict accident severity.

3.8 Relationship between SPEEDING and SEVERITYCODE:

	SPEEDING	SEVERITYCODE	SPEEDCOUNT
0	N	1	111478
1	N	2	109314
3	Y	2	7062
2	Y	1	4898

3.8.1 SPEEEDING vs SEVERITYCODE - Data Visualization

```
Text(0.5, 1.0, 'Vehicle speeding impacting accident severity')
```



From the barplot above, we can see that **SPEEEDING** has less impact on SEVERITYCODE influencing the accident severity. The accident has been reported more from vehicles which were not speeding. Hence, it should **not be considered** as part of the independent variables to predict accident severity.

At the end of Data visualization, we undersgtand that **UNDERINFL and SPEEDING** cannot be relied upon to predict SEVERITYCODE. Let us remove them from dataframe.

```
UNDERINFL and SPEEDING has been dropped from the Data Frame.
```

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	JUNCTIONTYPE	WEATHER	ROADCONI
25055	1	Intersection	Angles	At Intersection (intersection related)	Raining	Wet
65280	1	Intersection	Angles	At Intersection (intersection related)	Clear	Dry
86292	1	Intersection	Angles	At Intersection (intersection related)	Unknown	Unknown
155111	1	Block	Sideswipe	Mid-Block (not related to intersection)	Clear	Dry
64598	1	Block	Head On	Mid-Block (not related to intersection)	Clear	Dry

```
SEVERITYCODE      int64
ADDRTYPE         object
COLLISIONTYPE    object
JUNCTIONTYPE     object
WEATHER          object
ROADCOND         object
LIGHTCOND        object
dtype: object
```

4. Data Preprocessing:

The above indpendent variables are now categorical variables. To apply machine learning algorithms, we have to convert the categorical values to a dummy numeric values.

Let us do **Label encoding** to assign a unique numeric value to each catagory of variables.

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	JUNCTIONTYPE	WEATHER	ROADCOND	LIGI
1	1	2	0	1	6	8	2
2	1	2	0	1	1	0	5
3	1	2	0	1	10	7	8
4	1	1	9	4	1	0	5
5	1	1	2	4	1	0	5

Now, we have the target variable balanced and the input feature standardized. Now ,the data is ready to be fed to build data models

5. Data modeling:

5.1 Normalize the dataset:

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/sklearn/preproc
essing/data.py:625: DataConversionWarning: Data with input dtype int64 were al
l converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/ipykernel_launc
her.py:3: DataConversionWarning: Data with input dtype int64 were all converte
d to float64 by StandardScaler.
  This is separate from the ipykernel package so we can avoid doing imports un
til
```

```
array([[ 1.19928446, -1.54308736, -1.16620991,  1.14540715,  1.51817092,
        -1.44024903],
       [ 1.19928446, -1.54308736, -1.16620991, -0.72831759, -0.70452717,
         0.35067745],
       [ 1.19928446, -1.54308736, -1.16620991,  2.64438695,  1.24033366,
         2.14160392],
       ...,
       [-0.76190332, -0.83207189,  0.94047317, -0.72831759, -0.70452717,
         0.35067745],
       [ 1.19928446, -0.47656415, -1.16620991, -0.72831759, -0.70452717,
         0.35067745],
       [ 1.19928446, -1.18757963, -1.16620991, -0.72831759, -0.70452717,
         0.94765294]])
```

5.2 Splitting the dataset into test and train data:

```
Train set: (93100, 6) (93100,)
Test set: (23276, 6) (23276,)
```

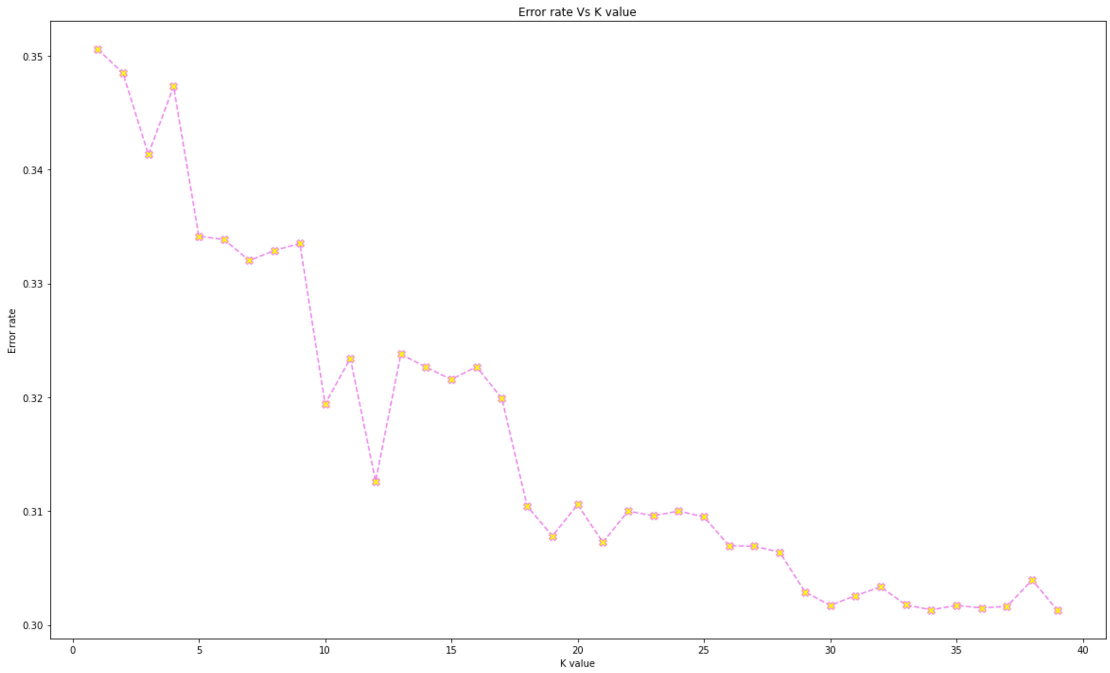
The four models which will be built, tested and compared are:

1. K-Nearest Neighbours (kNN)
2. Support Vector Machine (SVM)
3. Decision tree
4. Logistic Regression

1. K-Nearest Neighbours:

1.1.Determining the best K value:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=39, p=2,
           weights='uniform')
```



From the above plot, we can see that when the K value is 30, the error is low. So, let us choose the **K value as 30** to build the model.

```
Knn model with 30 is : KNeighborsClassifier(algorithm='auto', leaf_size=30, me
tric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=30, p=2,
           weights='uniform')
Confusion Matrix:

[[7405 2816]
 [4207 8848]]

Classification report:

              precision    recall  f1-score   support

           1       0.64      0.72      0.68     10221
           2       0.76      0.68      0.72     13055

   micro avg       0.70      0.70      0.70     23276
   macro avg       0.70      0.70      0.70     23276
weighted avg       0.71      0.70      0.70     23276


Model Accuracy

0.6982728991235607
```

2. Decision tree:

```
Decision tree classifier prediction is :
 [2 1 1 ... 2 2 2]
```

```
Confusion Matrix is below :
 [[7329 2599]
 [4283 9065]]

Classification report is below:
               precision    recall  f1-score   support

           1       0.63      0.74      0.68      9928
           2       0.78      0.68      0.72     13348

   micro avg       0.70      0.70      0.70     23276
   macro avg       0.70      0.71      0.70     23276
weighted avg       0.71      0.70      0.71     23276


Accuracy of Decision Tree Classifier is :
 0.7043306410036089
```

3. Support Vector Machine:

```
Support Vector Machine Prediction is :
 [2 1 1 ... 2 2 2]

Confusion Matrix is below :
 [[7034 2664]
 [4578 9000]]

Classification report is below:
               precision    recall  f1-score   support

           1       0.61      0.73      0.66      9698
           2       0.77      0.66      0.71     13578

   micro avg       0.69      0.69      0.69     23276
   macro avg       0.69      0.69      0.69     23276
weighted avg       0.70      0.69      0.69     23276


Accuracy of Support Vector Machine is :
 0.68886406599072

Logistic Regression Fit is :
 LogisticRegression(C=0.01, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='liblinear',
          tol=0.0001, verbose=0, warm_start=False)

Logistic Regression Prediction is :
 [1 1 1 ... 1 2 2]
```

```
Confusion Matrix is below :
 [[7534 5123]
 [4078 6541]]

Classification report is below:
               precision    recall  f1-score   support

           1       0.65      0.60      0.62     12657
           2       0.56      0.62      0.59     10619

   micro avg       0.60      0.60      0.60     23276
   macro avg       0.60      0.61      0.60     23276
weighted avg       0.61      0.60      0.61     23276


Accuracy of Logistic Regression is :
 0.6047001202955834
```

6. Data Evaluation:

6.1 Jaccard index:

```
The Jaccard score for knn is :0.698
The Jaccard score for Decision tree is :0.704
The Jaccard score for SVM is :0.689
The Jaccard score for Logistic regression is :0.605
```

6.2 F1 score:

```
The F1 score for Knn is :0.697
The F1 score for Decision Tree is :0.703
The F1 score for SVM is :0.687
The F1 score for Logistic Regression is :0.604

['NA', 'NA', 'NA', 0.6569156752383143]
```

6.3. Model scores report:

Algorithm	Jaccard	F1-score	LogLoss
KNN	0.698	0.697	NA
Decision Tree	0.704	0.703	NA
SVM	0.689	0.687	NA
Logistic Regression	0.605	0.604	0.656916

From the above scores, We can infer that **Decision tree classifier model** is the best model to predict the severity of accidents due to car collisions at Seattle.

7. Discussion:

As part of dimention reduction, the information after analysis is reduced drastically from 6 independent variables and 1 predictor variable and the number of records used had to be reduced by down sampling technique in order to arrive at better accurate prediction results.

Once I analyzed and cleaned the data, the data was then fed through four Machine learning models; K-Nearest Neighbor, Decision Tree, Support Vector Machine and Logistic Regression. Although the first three are ideal for this project, logistic regression made most sense because of its binary nature.

Evaluation metrics used to test the accuracy of our models were jaccard index, f-1 score and logloss for logistic regression. Choosing K value as 30 (max depth) and hyparameter C values helped to improve our accuracy to be the best possible.

8. Conclusion:

The information provided by Seattle Police Department is a first step to prove that a model can be generated to predict future accidents on the road and identify the type of information (independent variables) that can be used.

A condition specifying the reason behind car accidents during ideal driving conditions will increase the effectiveness of the model exponentially.