Table of Contents

Introduction:

This project is a part of IBM Data Science Professional Certificate Course that I am pursuing on Coursera. This project aims to answer to the business problem: "Can a driver predict the severity of a future accident by observing a number of conditions such as road condition, weather condition, visibility condition etc.".

Car accident has become a very serious problem in Seattle. It has become essential to develop a model which can predict the severity of a car accident in terms of human fatality, physical injury, property damage, traffic delay etc and thereby alert the drivers about the accidents ahead of time.

Stakeholders: The reduction in severity of accidents can be beneficial to the Public Development Authority of Seattle which works towards improving those road factors and the car drivers themselves who may take precaution to reduce the severity of accidents.

Data Unserstanding

Data Collection

For this project, we don't need to go through web scraping as we already have a data set provided as a csv file which can be found here (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/). This csv file is based on car accidents which have taken place within the city of Seattle, Washington from the year 2004 to 2020. This data shows the severity of each car accidents along with the time and conditions under which each accident occurred. Theer are 194673 number of rows and 38 columns in this data set. Each row represents a particular accident whereas the columns represent various conditions under which the accidents take place.

Feature Selection

We will select "ROADCONDITION", "WEATHERCONDITION", "LIGHTCONDITION", "SPEEDING", "INATTENTIONIND" and "UNDERINFL" as the features/ attributes of our model.

Feature Variable	Description
INATTENTIONIND	Whether or ot the driver was inattentive
UNDERINFL	Whether or ot the driver was inattentive
ROADCONDITION	Road Condition during the collision
WEATHERCONDITION	Weather Condition during the collision
LIGHTCONDITION	Weather Condition during the collision
SPEEDING	Whether the car was above speed limit during collision

Target/ Label selection

We will choose the column "Severity Code" as the target/ label of our model.

The severity code is assigned as follows-

- 0 - No probability of accident
- 1 - Only Property Damage
- 2 - Physical Injury

Methodology

Exploratory Analysis

Lets take a look at the first five rows of the original data set.

```
C:\Users\Tania\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (33) have mixed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

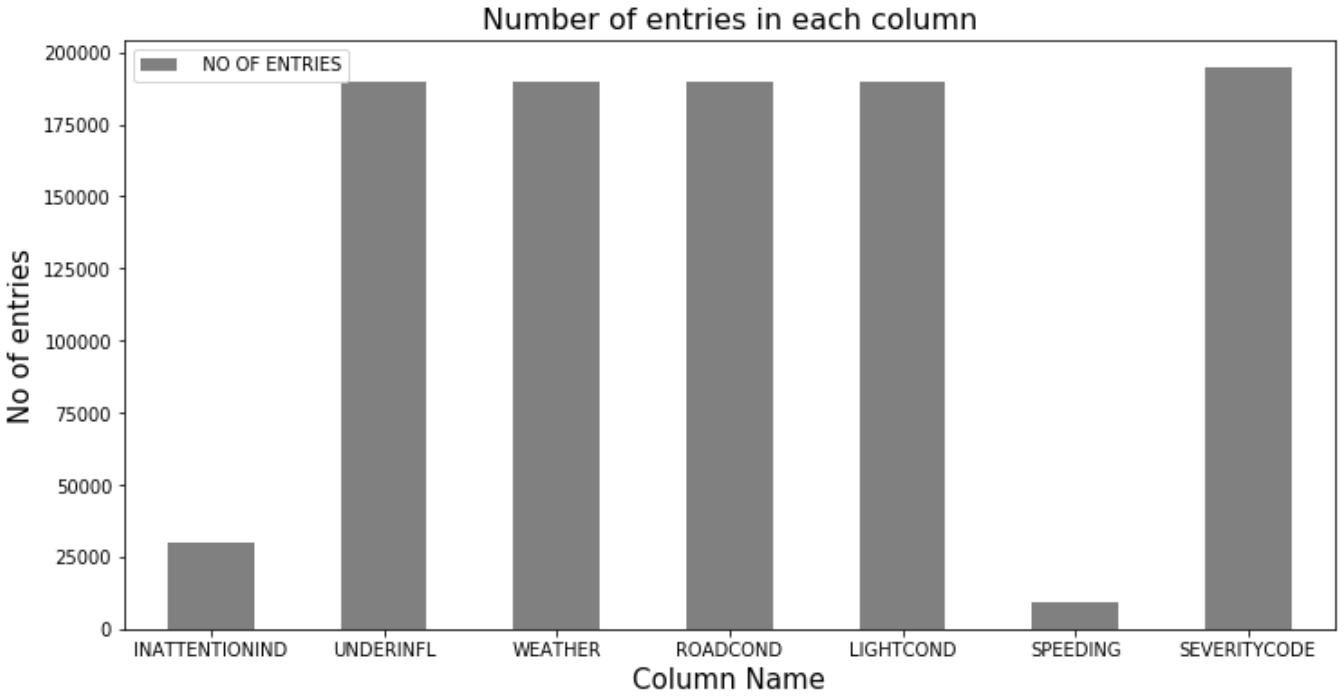
	ï»¿SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADD
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Inte
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Inte

5 rows × 38 columns

Lets remove the columns that are not needed for data analysis. We will use the following 7 columns in our analysis.

	INATTENTIONIND	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND	SPEEDING	SEVERITYCODE
0	NaN	N	Overcast	Wet	Daylight	NaN	2
1	NaN	0	Raining	Wet	Dark - Street Lights On	NaN	1
2	NaN	0	Overcast	Dry	Daylight	NaN	1
3	NaN	N	Clear	Dry	Daylight	NaN	1
4	NaN	0	Raining	Wet	Daylight	NaN	2

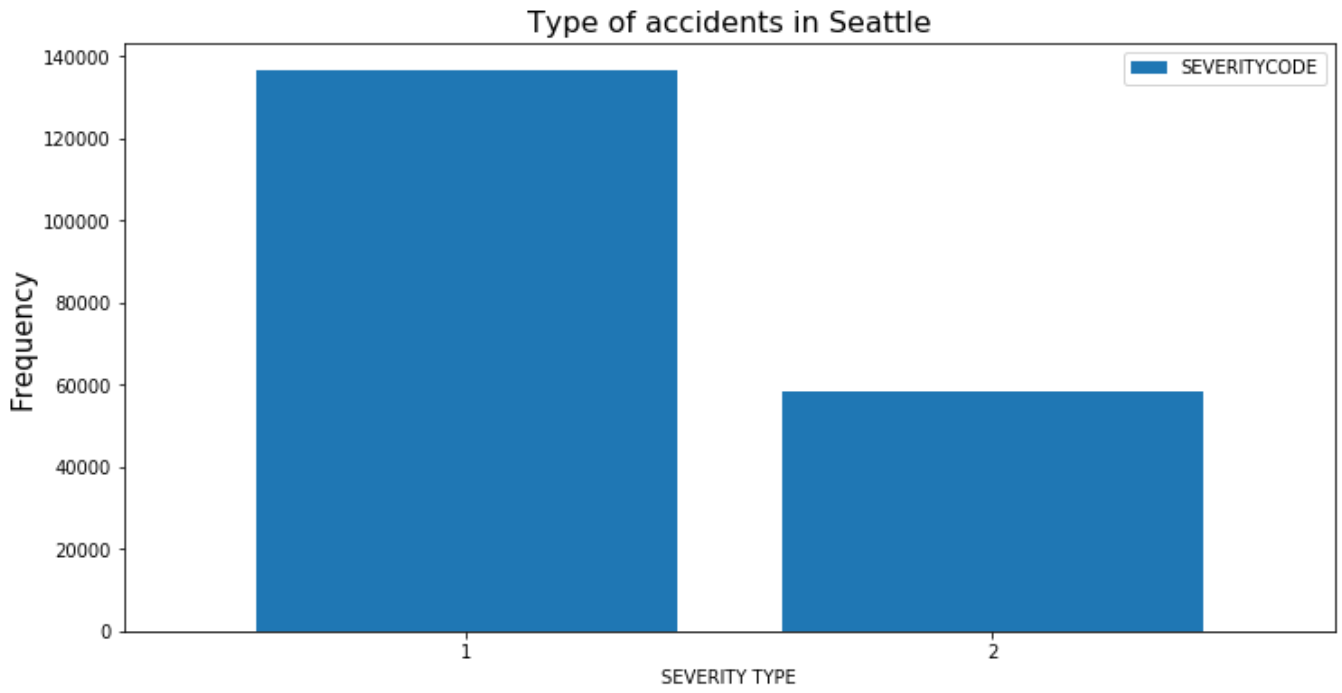
There are many rows where the values for all columns are not inserted. The following table and graph shows how many datas are inserted for each column. As there are significant number of rows with null entry, we cant just delete these rows. Later We will replace these null values by some sumber.



Balancing the Dataset

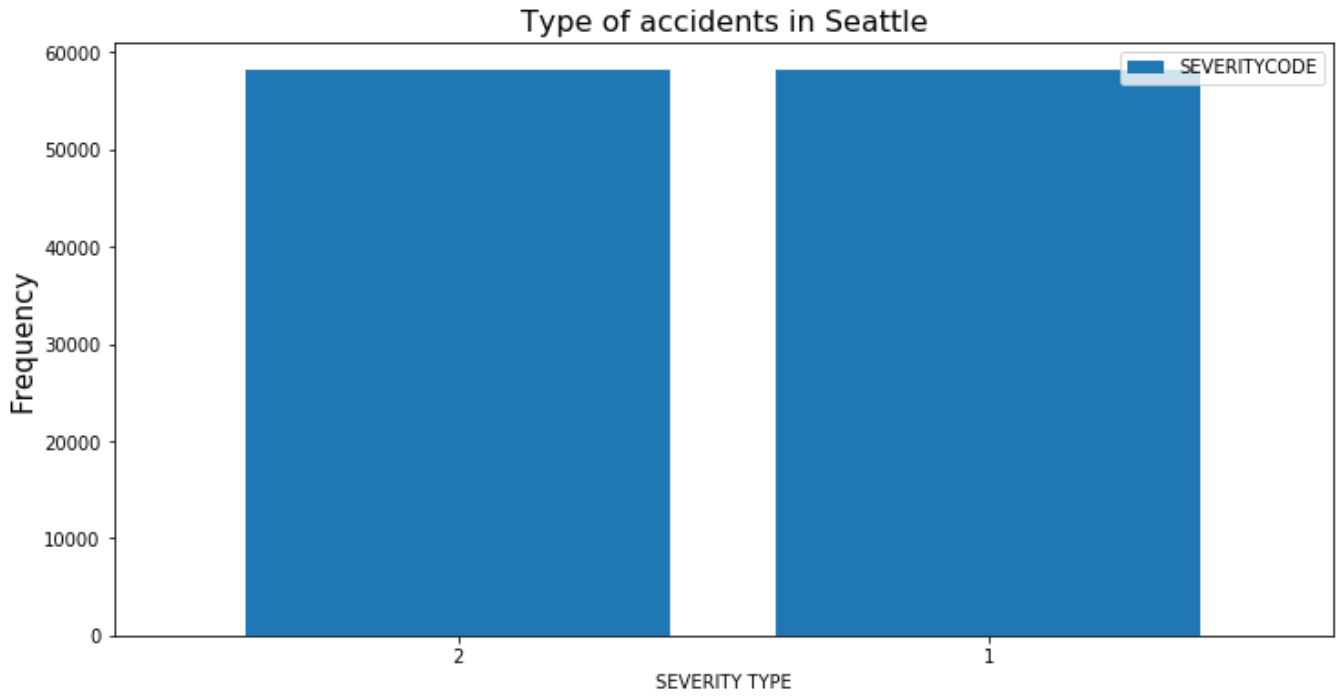
Number of entries in SEVERITYCODE is given as follows

	SEVERITYCODE
1	136485
2	58188



We can see that dataset is unbalanced where the distribution of the target variable is in almost 1:3 ratio in favor of property damage. It is very important to have a balanced dataset when using machine learning algorithms. Hence, we reasmpled the data in order to balance the target variable in equal proportions in order to have an unbiased classification model which is trained on equal instances of both the elements under severity of accidents.

	SEVERITYCODE
2	58188
1	58188



Converting Categorical features to Numerical values:

We will now investigate each column, we will replace the categorial value of each feature by numerical value depending on the accident severity.For this conversion, we cant just depend on the values inserted in the columns. For example, in weather column, the weather condition "Clear" appears the most, but it does not mean that the clear weather result in most severe accidents. Most of the days in the year are clear, thats why it results in highest number of entries in weather column. SO we will see how many accidents will result in physical injusy. We will categorize each column into different groups and assign a number to each group depending on the ration of physical injury to total number of accidents.

Column SEVERITYCODE

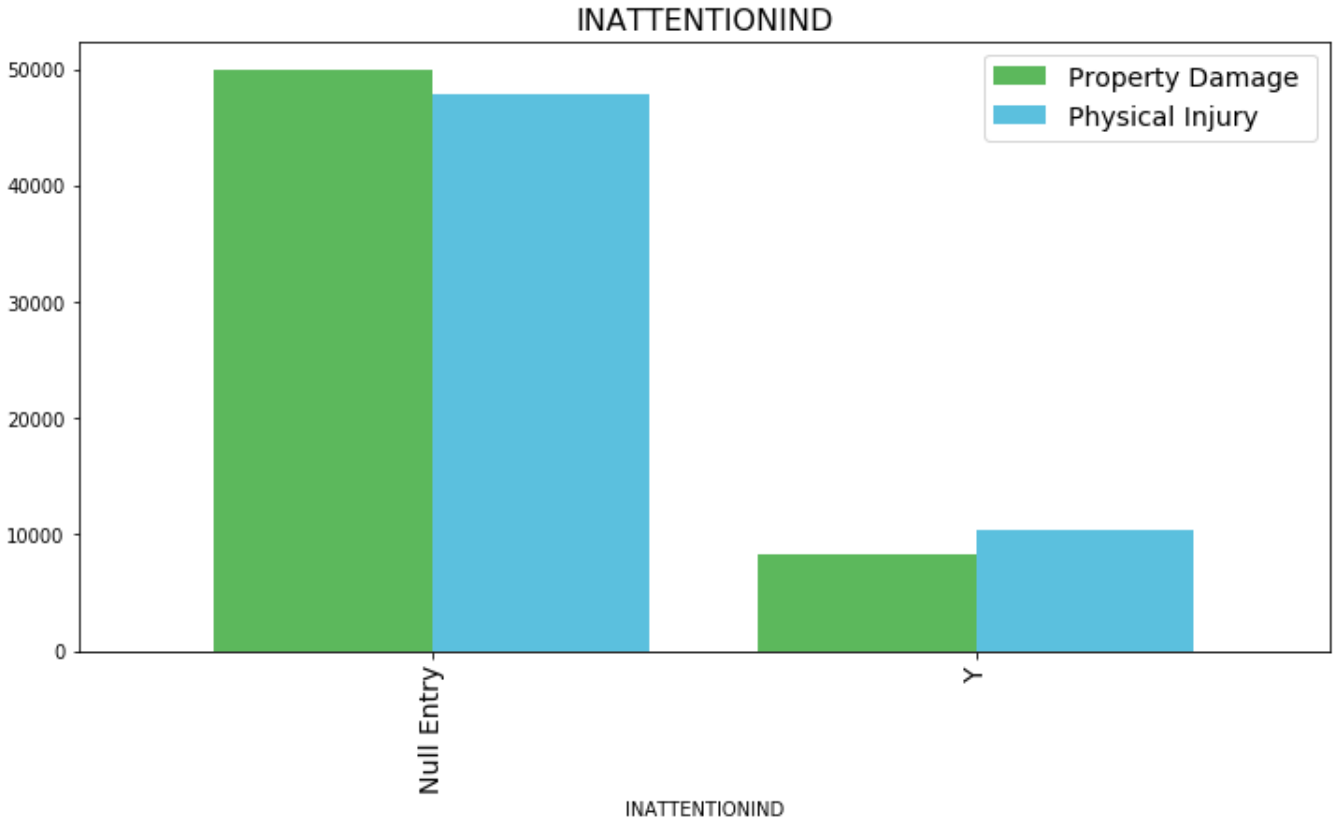
```
   SEVERITYCODE
2         58188
1         58188

Number of Null entries:  0
```

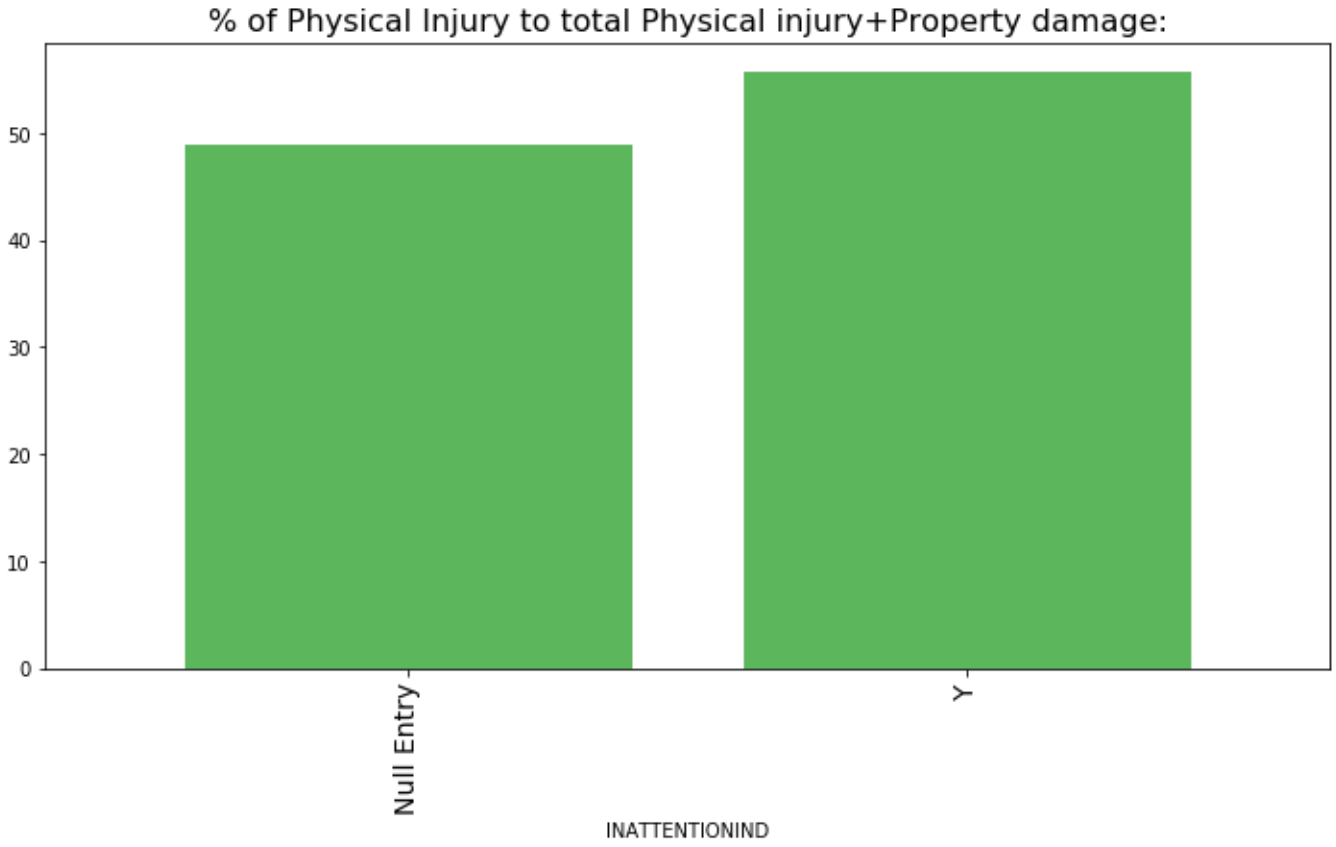
The column SEVERITYTYPE does not need to change, there is no null value. We will use this severity code as the label of out model.

Column INATTENTIONIND

Number of Null entries: 97717

INATTENTIONIND	Property Damage	Physical Injury
Null Entry	49926	47791
Y	8262	10397





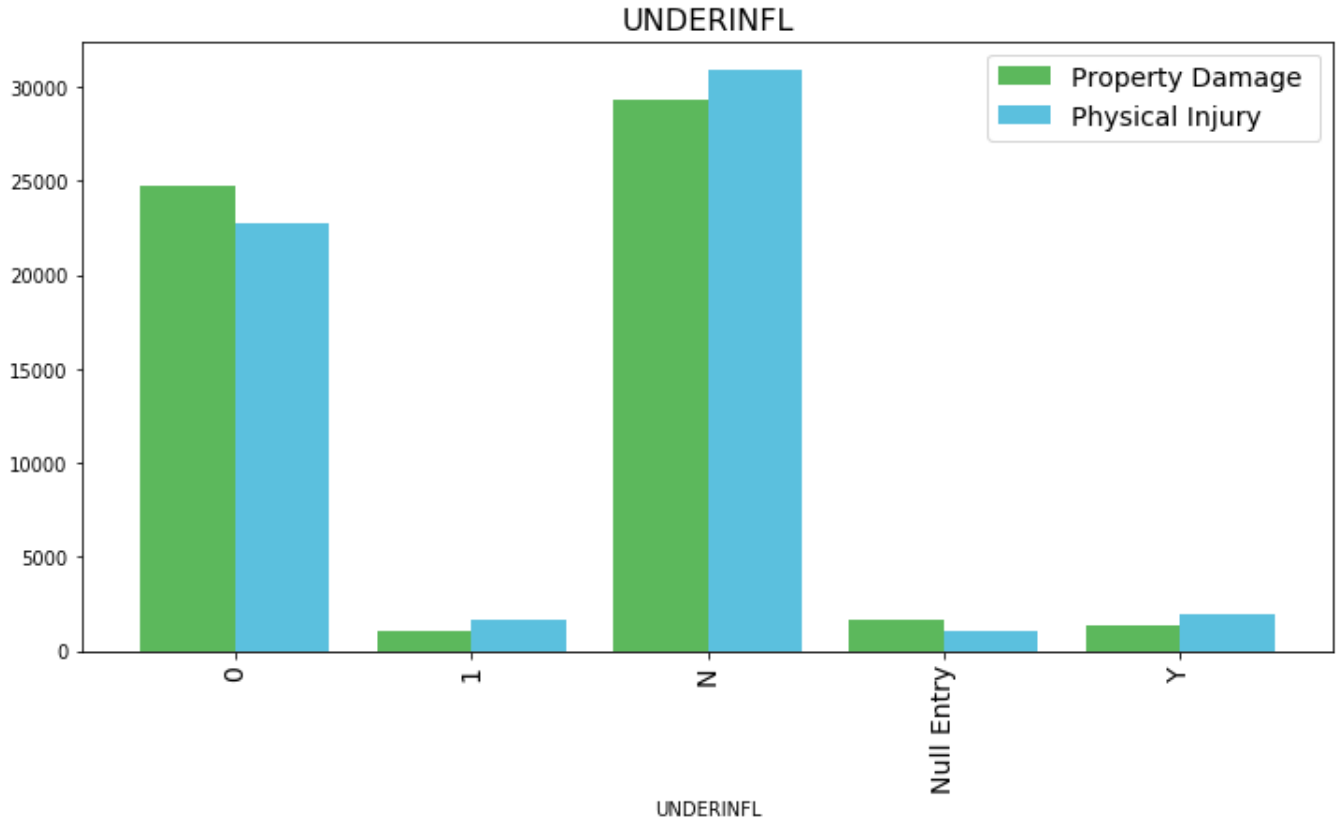
There are 97717 null entries in the column INATTENTIONIND. We can assume blank cell/ null entry means there was no such condition. Lets replace null values by 0 and 'Y' by 1.

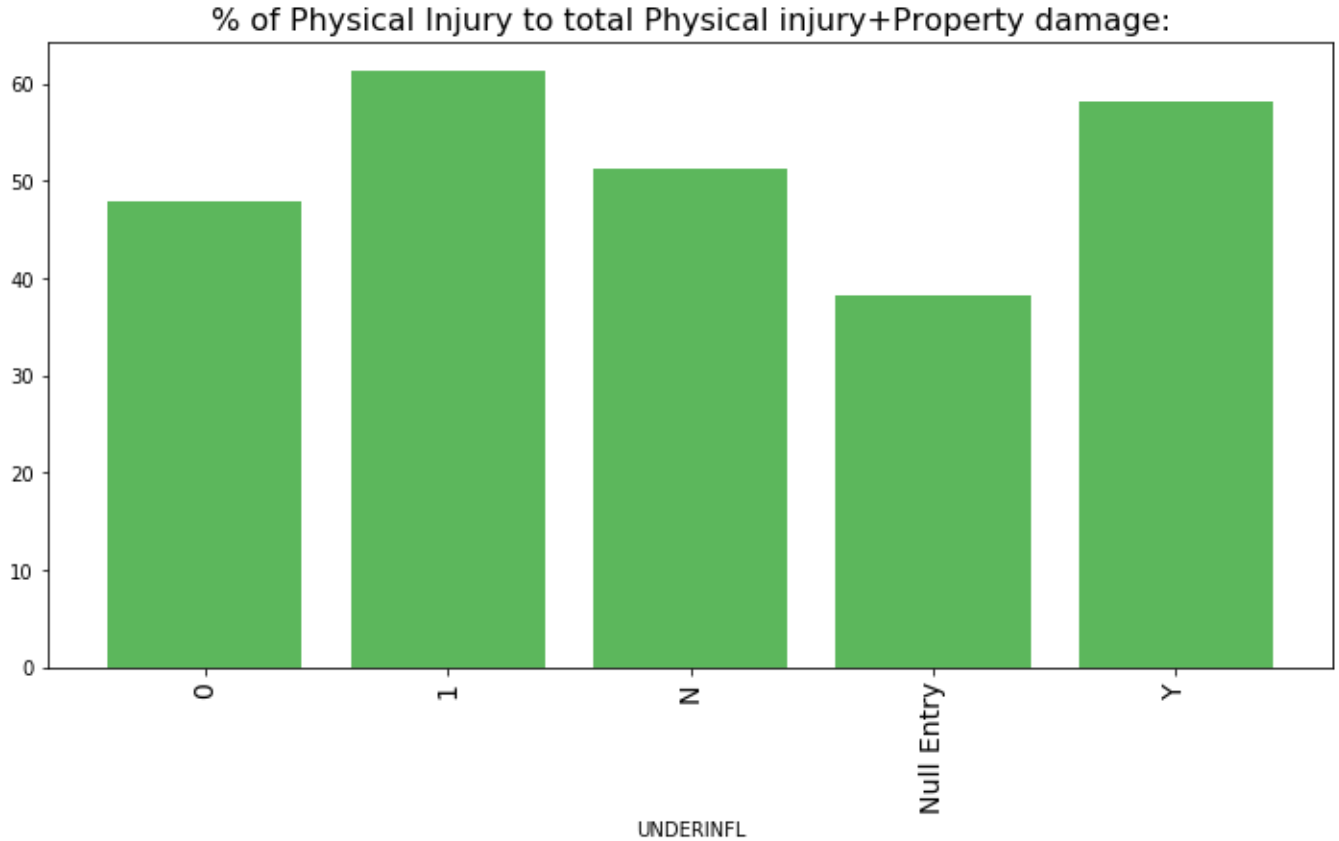
	INATTENTIONIND
0	97717
1	18659

Column UNDERINFL

Number of Null entries: 2693

UNDERINFL	Property Damage	Physical Injury
0	24762	22701
1	1025	1623
N	29336	30896
Null Entry	1664	1029
Y	1401	1939





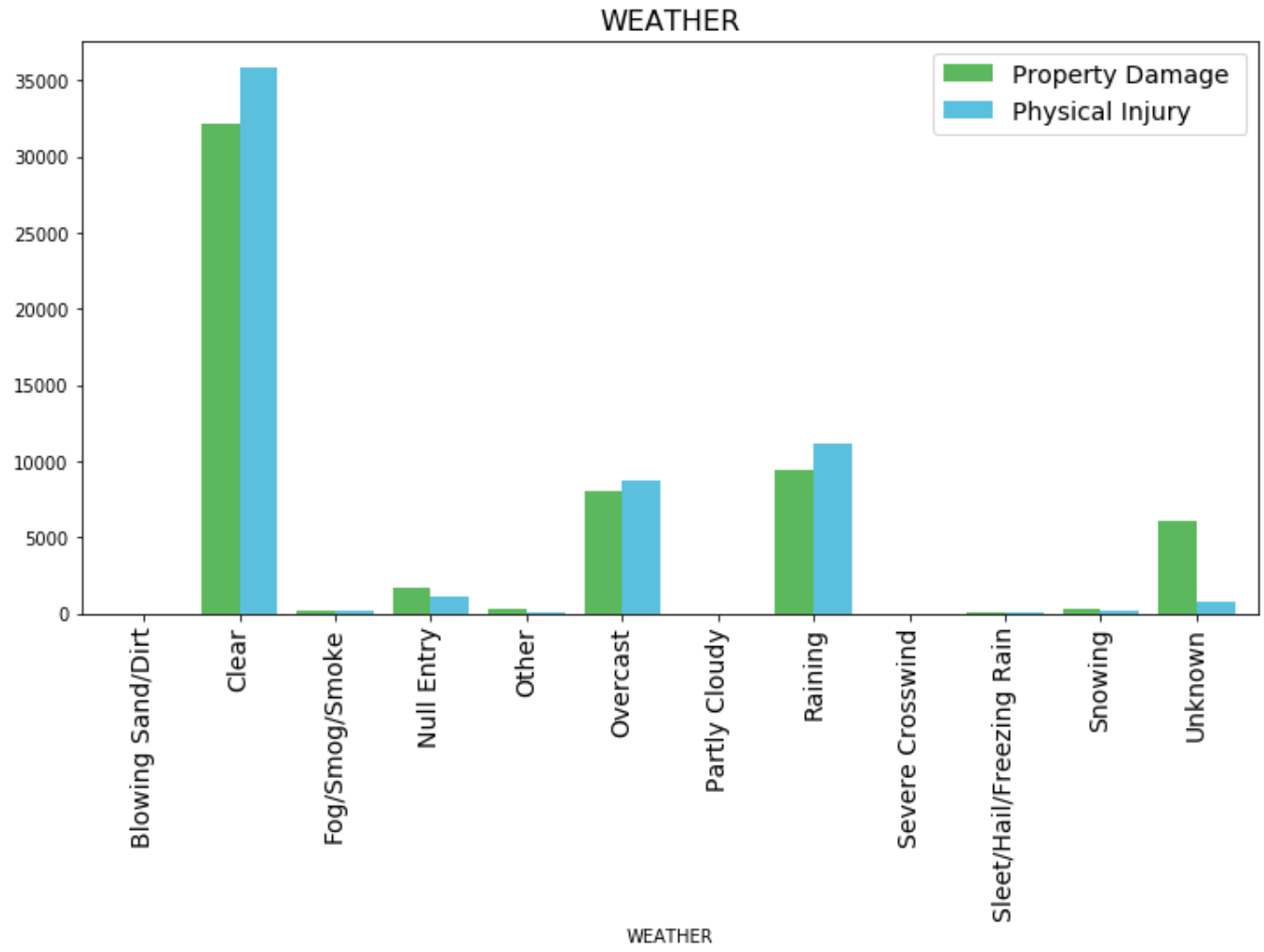
There are 2693 null entries in the column UNDERINFL, lets replace null values by 0, 'Y' by 1, 'N' by 0

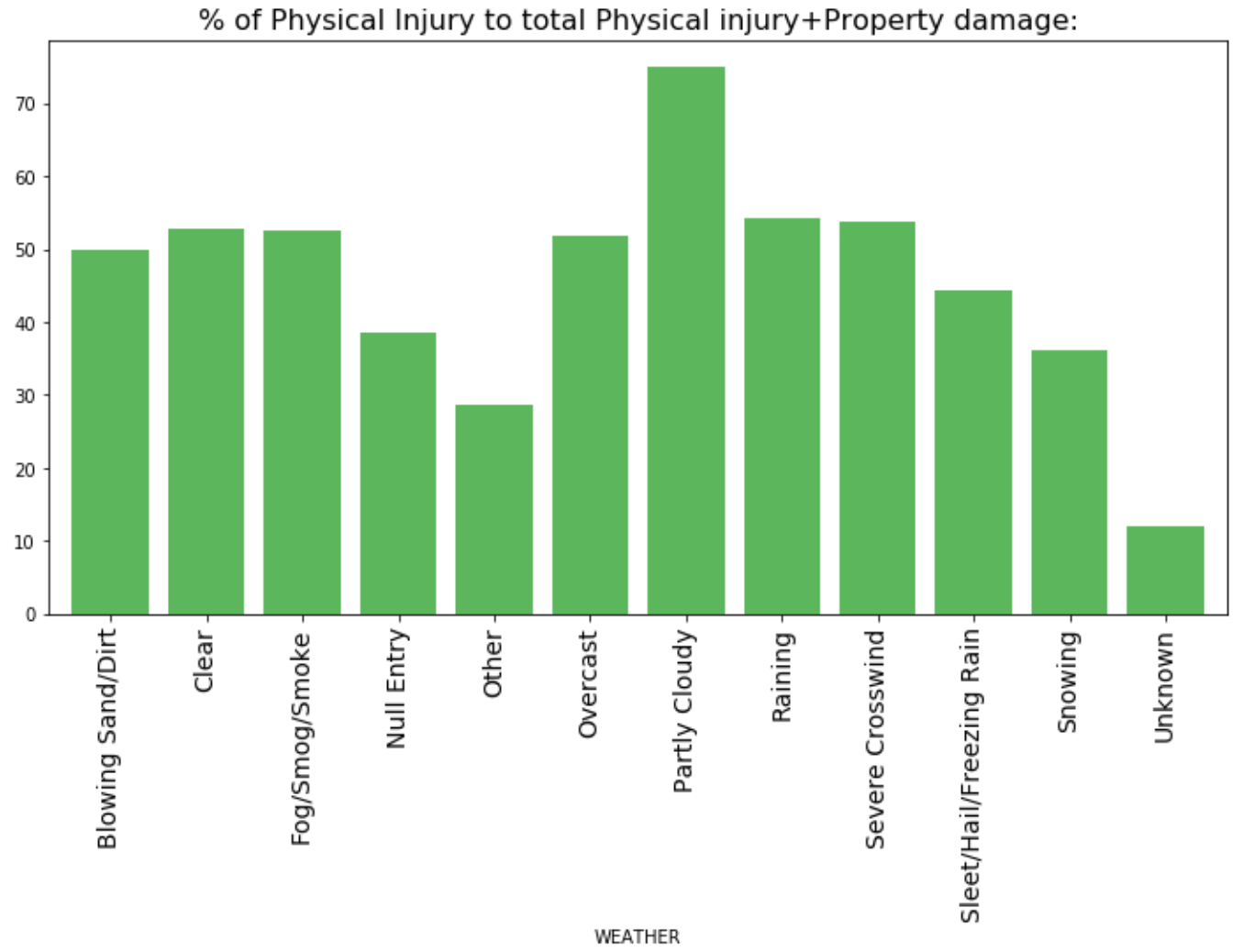
	UNDERINFL
0	110388
1	5988

Column WEATHER

Number of Null entries: 2816

WEATHER	Property Damage	Physical Injury
Blowing Sand/Dirt	15	15
Clear	32106	35840
Fog/Smog/Smoke	168	187
Null Entry	1732	1084
Other	290	116
Overcast	8089	8745
Partly Cloudy	1	3
Raining	9408	11176
Severe Crosswind	6	7
Sleet/Hail/Freezing Rain	35	28
Snowing	303	171
Unknown	6035	816





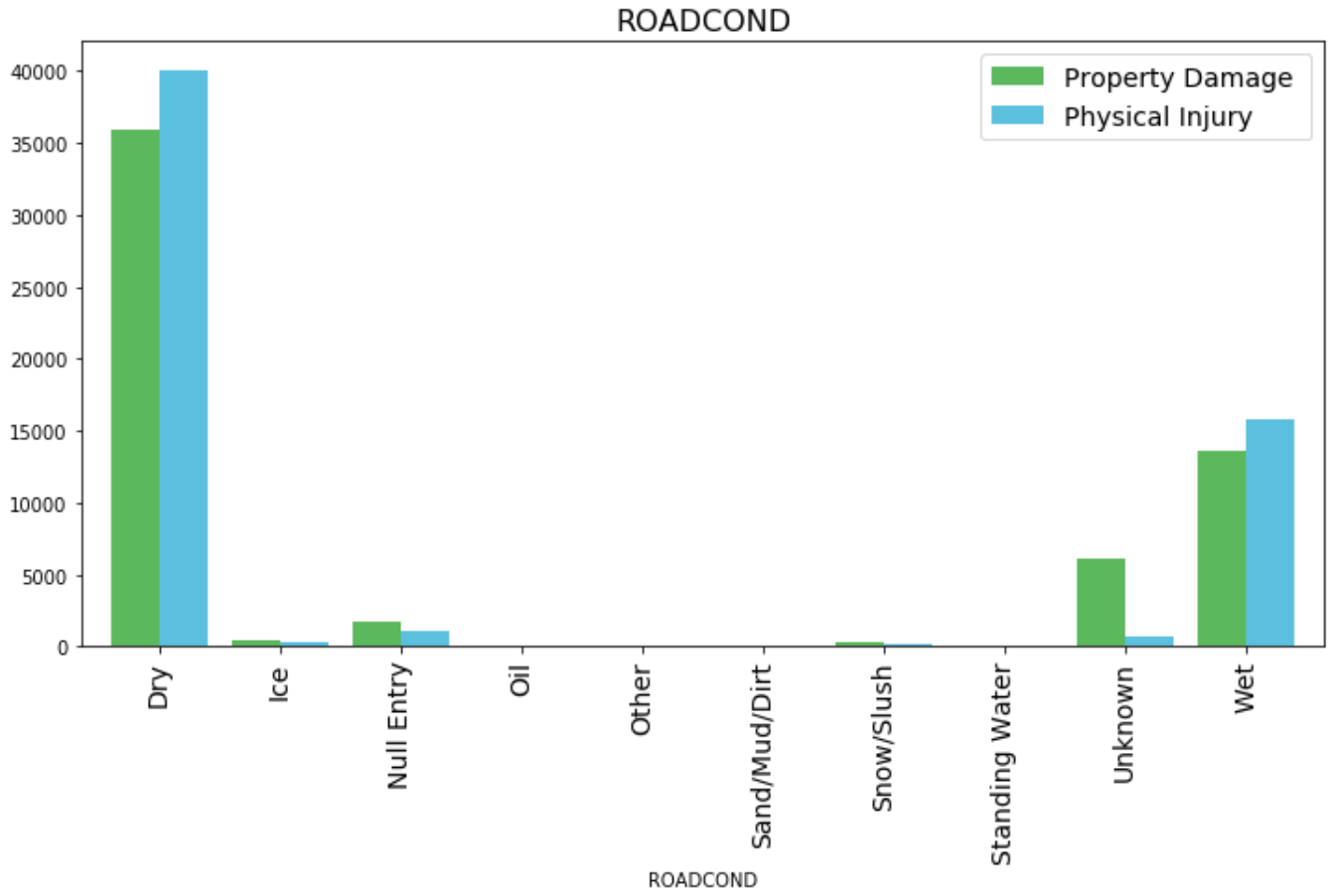
There are 2816 null entries in the column WEATHER. From the above graph we can categorize the weather condition into 9 different groups and assign a point in the range 0-8.

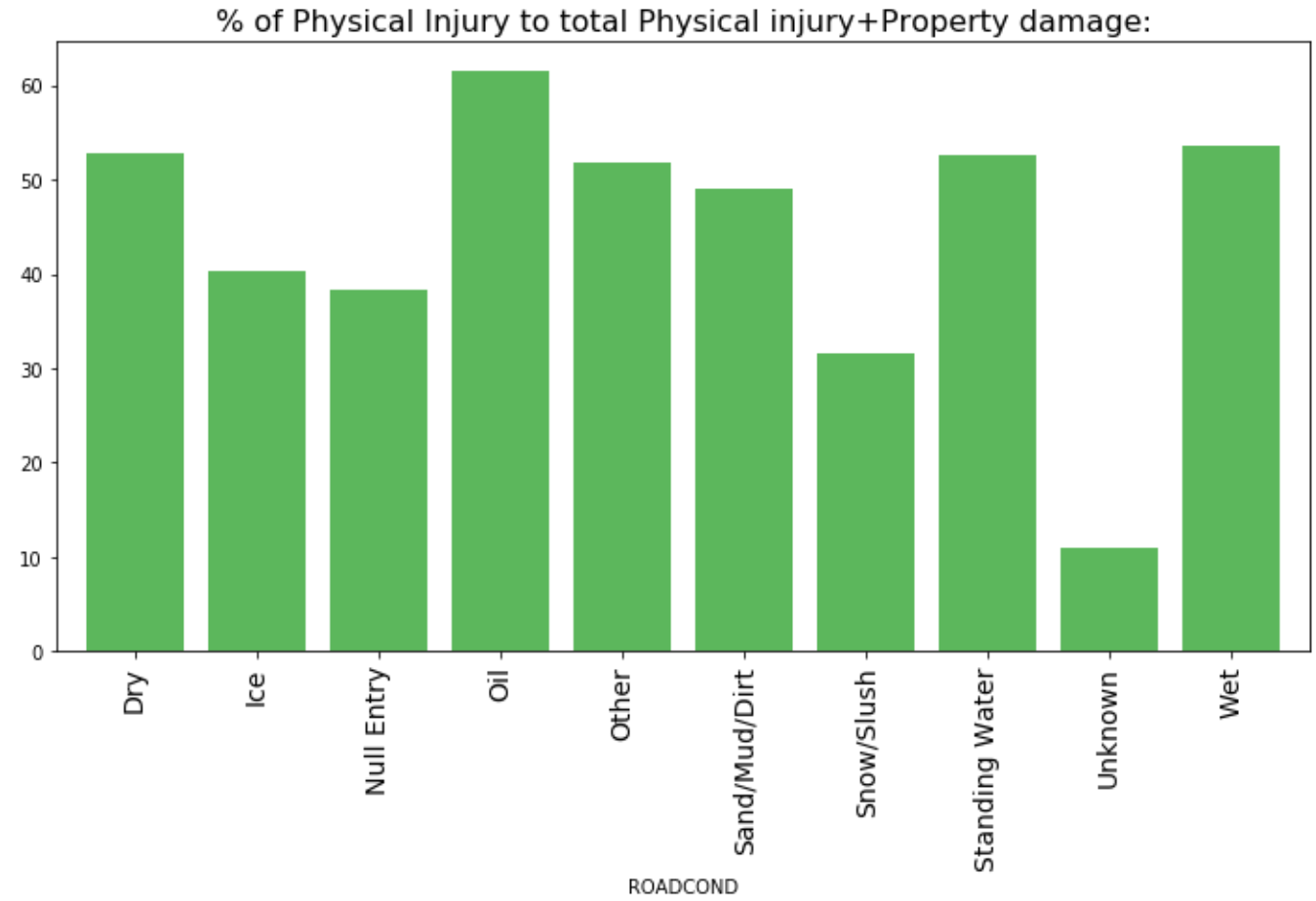
	WEATHER
7	67946
6	20939
5	16847
0	6851
2	2816
3	474
1	406
4	93
8	4

Column ROADCOND

Number of Null entries: 2764

ROADCOND	Property Damage	Physical Injury
Dry	35936	40064
Ice	405	273
Null Entry	1704	1060
Oil	15	24
Other	40	43
Sand/Mud/Dirt	24	23
Snow/Slush	361	167
Standing Water	27	30
Unknown	6057	749
Wet	13619	15755





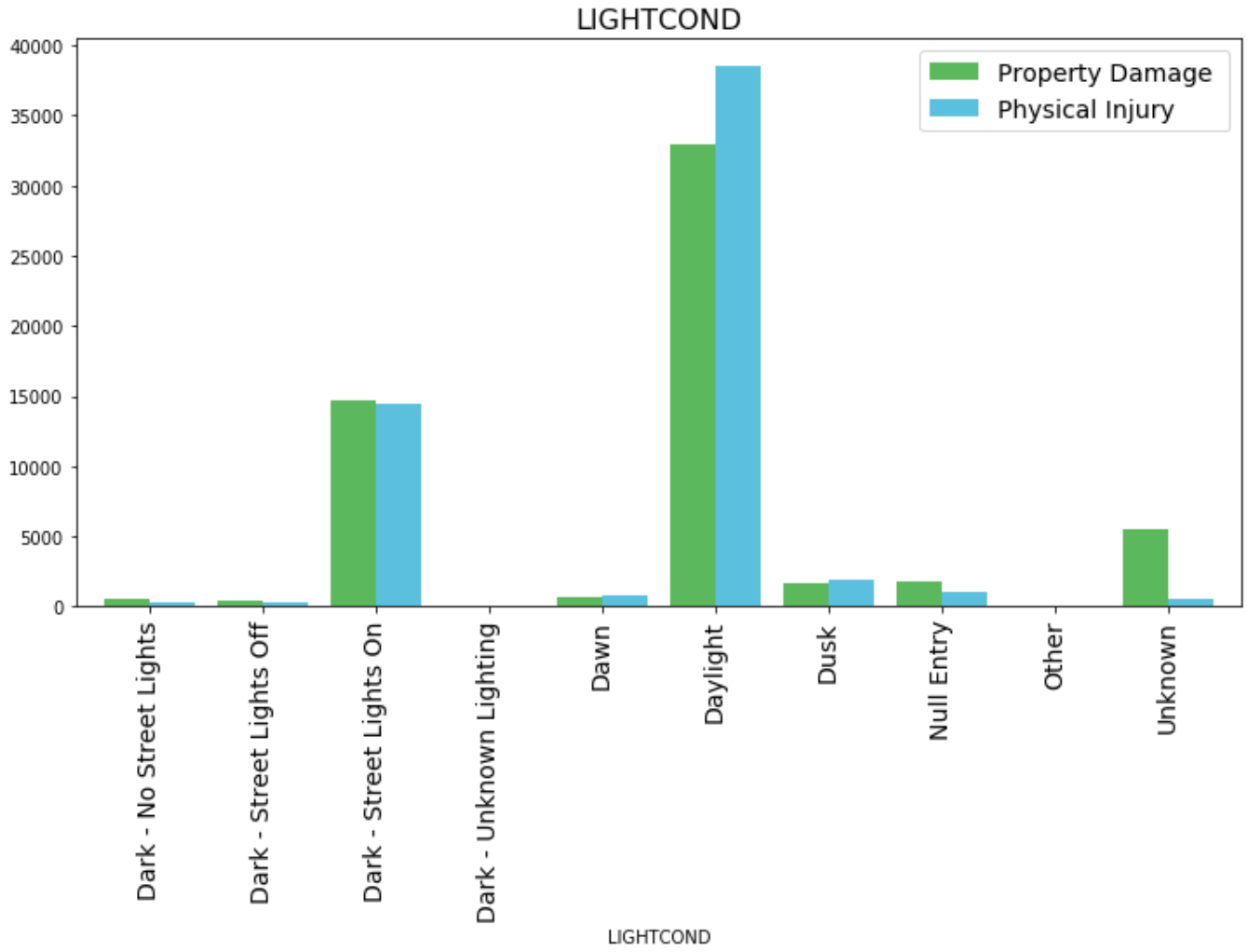
There are 2764 null entries in the column ROADCOND. From the above graph we can categorize the weather condition into 9 different groups and assign a point in the range 0-8.

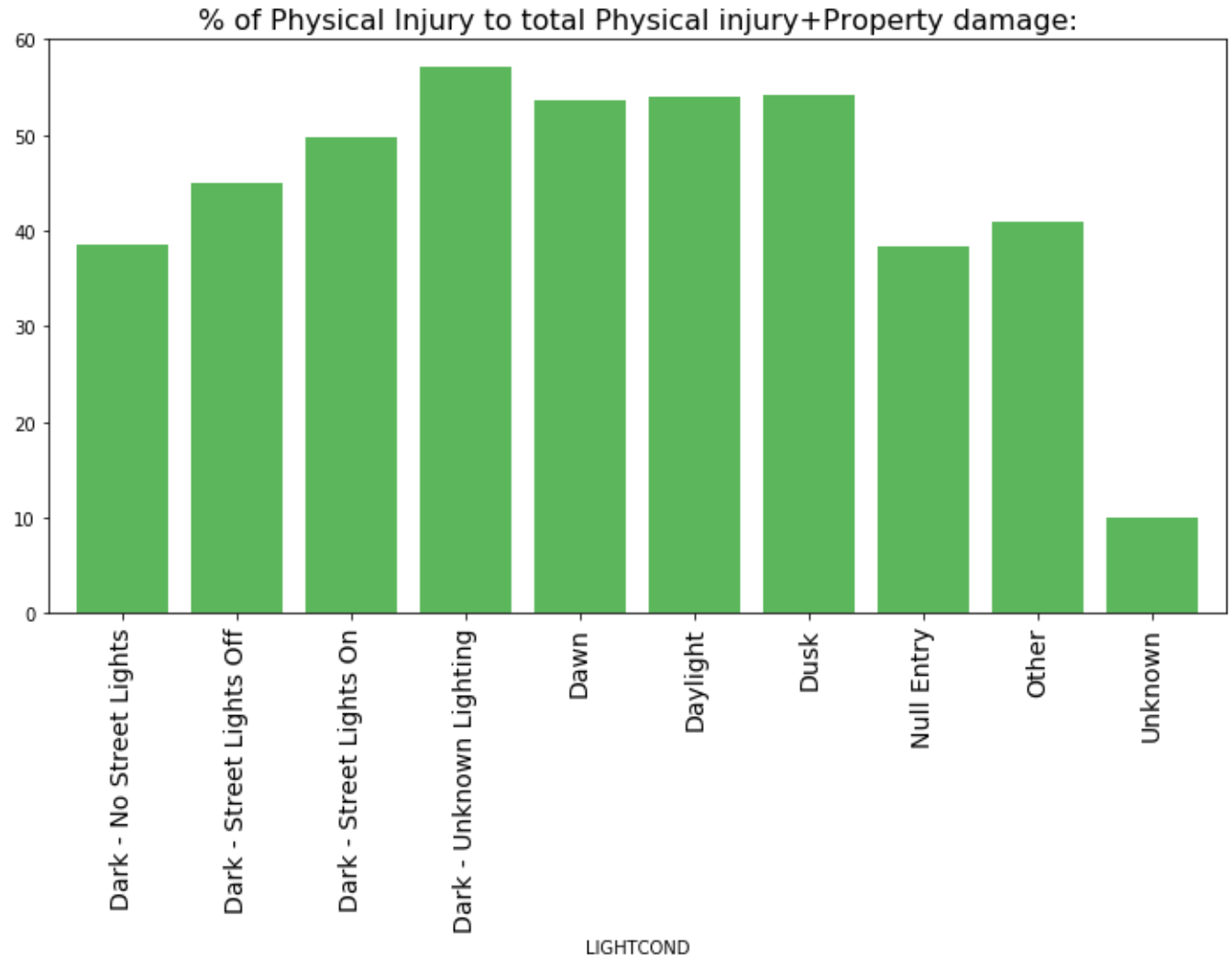
	ROADCOND
6	76000
7	29431
0	6806
1	2764
4	678
3	528
2	83
5	47
8	39

Column LIGHTCOND

Number of Null entries: 2848

LIGHTCOND	Property Damage	Physical Injury
Dark - No Street Lights	532	334
Dark - Street Lights Off	387	316
Dark - Street Lights On	14658	14475
Dark - Unknown Lighting	3	4
Dawn	711	824
Daylight	32959	38544
Dusk	1648	1944
Null Entry	1758	1090
Other	75	52
Unknown	5457	605





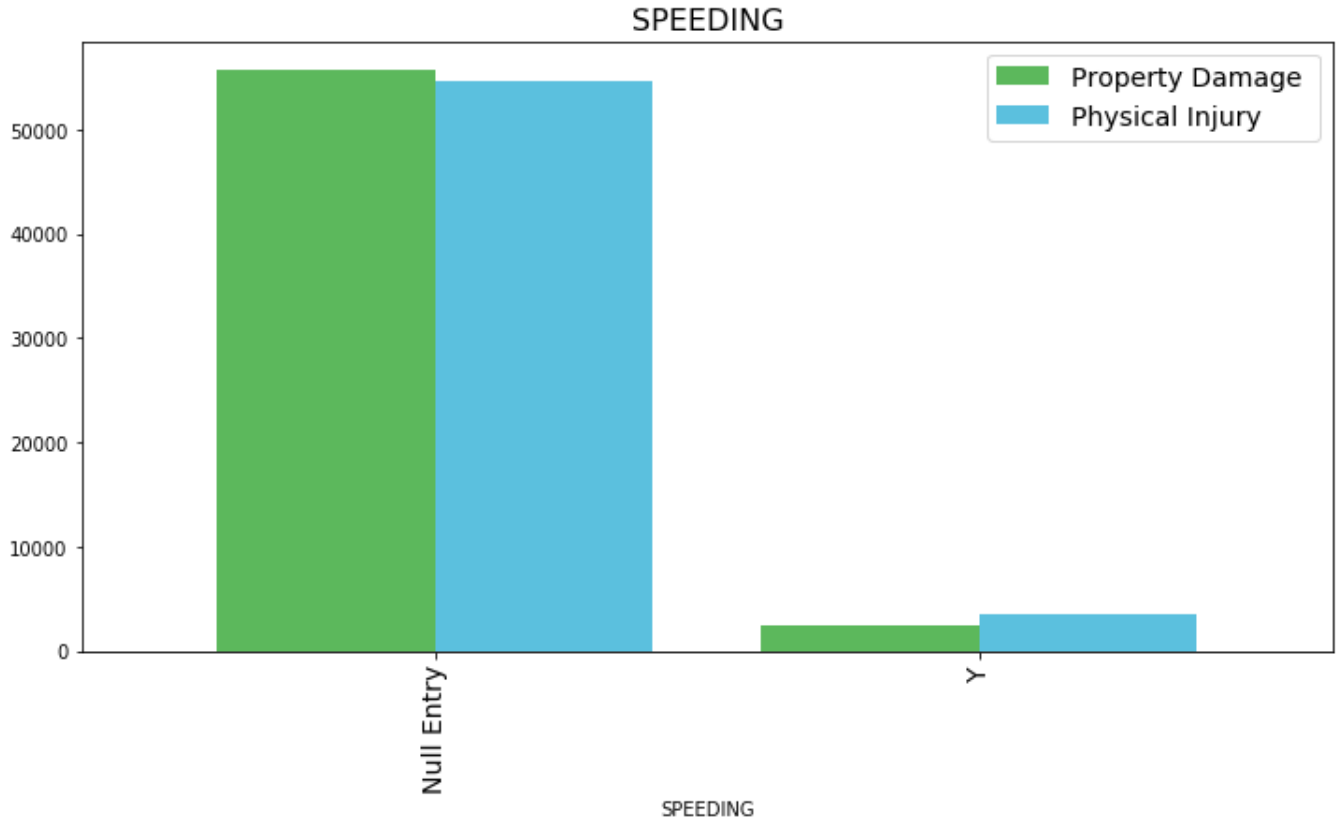
There are 2848 null entries in the column LIGHTCOND. From the above graph we can categorize the light condition into 7 different groups and assign a point in the range 0-6.

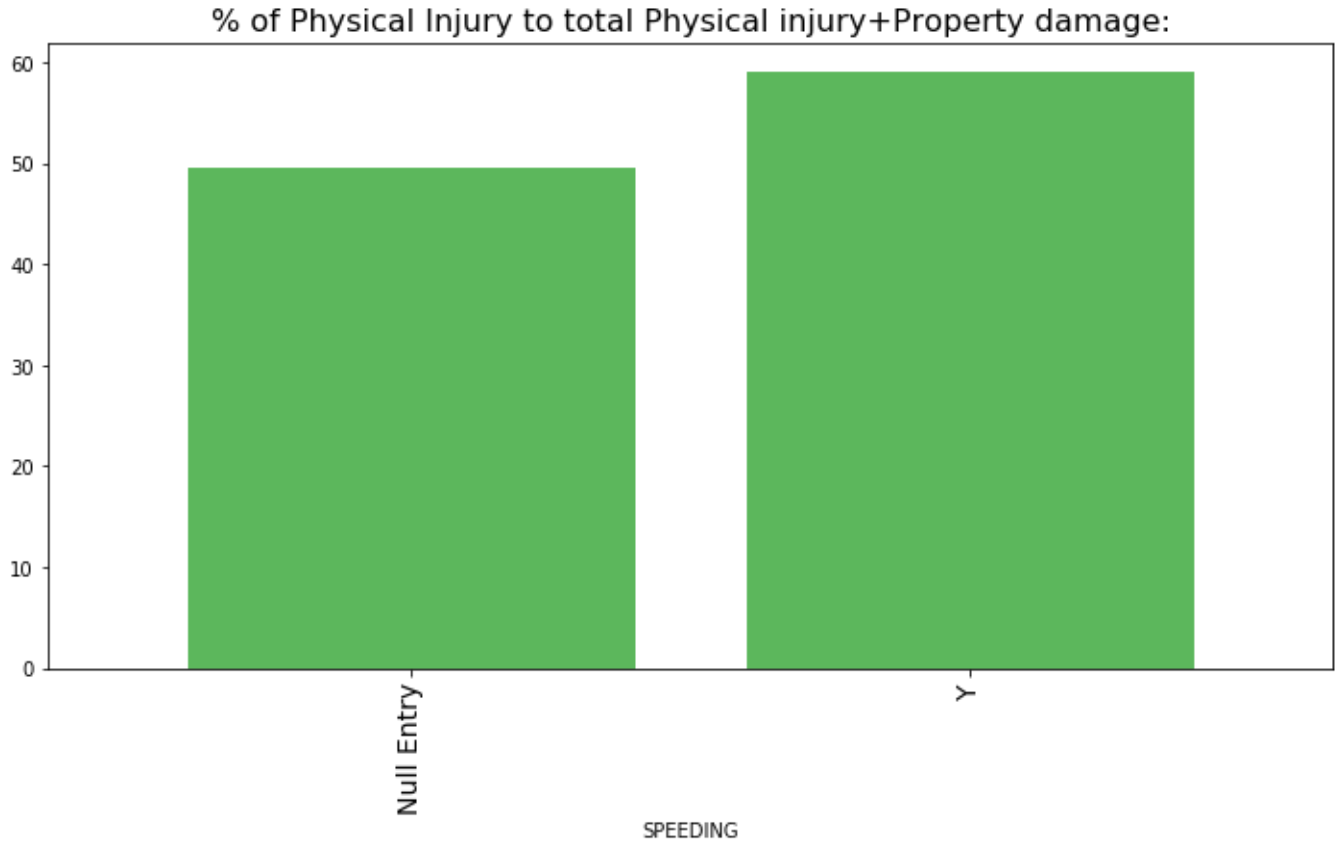
	LIGHTCOND
5	76630
4	29133
0	6062
1	2975
2	866
3	703
6	7

Column SPEEDING

Number of Null entries: 110396

SPEEDING	Property Damage	Physical Injury
Null Entry	55739	54657
Y	2449	3531





There are 110396 null entries in the column SPEEDING. We can assume blank cell/ null entry means there was no such condition. Lets replace null values by 0 and 'Y' by 1.

```
   SPEEDING
0    110396
1      5980
```

Correlation between diffeternt columns

	INATTENTIONIND	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND	SPEEDING	SEVERIT
INATTENTIONIND	1.000000	-0.033188	0.096425	0.085185	0.107399	-0.051636	0
UNDERINFL	-0.033188	1.000000	0.059519	0.065540	-0.032940	0.100824	0
WEATHER	0.096425	0.059519	1.000000	0.756772	0.657270	0.027109	0
ROADCOND	0.085185	0.065540	0.756772	1.000000	0.666291	0.070574	0
LIGHTCOND	0.107399	-0.032940	0.657270	0.666291	1.000000	0.021356	0
SPEEDING	-0.051636	0.100824	0.027109	0.070574	0.021356	1.000000	0
SEVERITYCODE	0.050000	0.044185	0.179374	0.189786	0.187863	0.042111	1

From the correlation table, we can see that, there is a strong relation between Weather, Road Condition and Light Condition

Feature Selection

The following columns will be used as features in our model.

	INATTENTIONIND	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND	SPEEDING
25055	0	0	6	7	4	0
65280	0	0	7	6	5	0
86292	0	0	0	0	0	0
155111	0	0	7	6	5	0
64598	0	0	7	6	5	0

Label Selection

The column SEVERITYCODE will be used as our label. SO there will be 2 labels: label 1 and label 2.

Normalize Data

We will normalize the features before developing the model. The first 5 rows of normalized featured data is as follows-

```
array([[-0.43697754, -0.23290562,  0.02297305,  0.72225251, -0.27587799,
        -0.23274155],
       [-0.43697754, -0.23290562,  0.56847646,  0.14194529,  0.50649789,
        -0.23274155],
       [-0.43697754, -0.23290562, -3.25004737, -3.33989801, -3.40538151,
        -0.23274155],
       [-0.43697754, -0.23290562,  0.56847646,  0.14194529,  0.50649789,
        -0.23274155],
       [-0.43697754, -0.23290562,  0.56847646,  0.14194529,  0.50649789,
        -0.23274155]])
```

Dividing data into training set and test set

In order to develop a model for predicting accident severity, the resampled, cleaned dataset was split in to testing and training sub-samples (containing 30% and 70% of the samples, respectively) using the scikit learn “train_test_split” method.

Machine Learning Models: Classification

We will use the training set to build an accurate model. Then use the test set to report the accuracy of the model.

We will use the following algorithm:

- K Nearest Neighbor(KNN)
- Decision Tree
- Logistic Regression

Support Vector Machine will not be used as this algorithm is ussed for small dataset.

K Nearest Neighbor(KNN)

KNN will help us predict the severity code of an outcome by finding the most similar to data point within k distance.

We will first find the best k to build the model with the best accuracy.

```
K=  12  ,Accuracy Score=  0.5191762380775069 , F1 Score=  0.4667196901947999
K=  13  ,Accuracy Score=  0.5359035316357804 , F1 Score=  0.5333447086160248
K=  14  ,Accuracy Score=  0.5166843296193395 , F1 Score=  0.4676649660358987
K=  15  ,Accuracy Score=  0.522470140062441 , F1 Score=  0.49584148775223524
K=  16  ,Accuracy Score=  0.5220691432990576 , F1 Score=  0.4765245571776488
K=  17  ,Accuracy Score=  0.515853693466617 , F1 Score=  0.5088676541335826
K=  18  ,Accuracy Score=  0.5249906911465643 , F1 Score=  0.4962442609245468
```

We can see that for *k=13*, KNeighborsClassifier is giving the best F1 score. Lets build the KNN model with k=13

```
For K= 13 , Accuracy Score= 0.5359035316357804 , Jaccard Score= 0.5359035316357804 , F1
Score=  0.5333447086160248

C:\Users\Tania\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:664: Futu
reWarning: jaccard_similarity_score has been deprecated and replaced with jaccard_scor
e. It will be removed in version 0.23. This implementation has surprising behavior for
binary and multiclass classification tasks.
  FutureWarning)
```

Decision Tree

A decision tree model gives us a layout of all possible outcomes so we can fully analyze the concequences of a decision. It context, the decision tree observes all possible outcomes of different weather conditions.

```
Max Depth=  1 , Accuracy Score=  0.5451550998195515 , F1 Score=  0.4359419322700957
Max Depth=  2 , Accuracy Score=  0.5587030618967147 , F1 Score=  0.4754656341244437
Max Depth=  3 , Accuracy Score=  0.5586744192707588 , F1 Score=  0.47538895933312886
Max Depth=  4 , Accuracy Score=  0.5670380660498955 , F1 Score=  0.5520076852738023
Max Depth=  5 , Accuracy Score=  0.5703606106607854 , F1 Score=  0.5494107069975682
Max Depth=  6 , Accuracy Score=  0.5705038237905651 , F1 Score=  0.5511519243168462
Max Depth=  7 , Accuracy Score=  0.5714776730730673 , F1 Score=  0.534849820934144
Max Depth=  8 , Accuracy Score=  0.5725088076074815 , F1 Score=  0.5377219696765829
Max Depth=  9 , Accuracy Score=  0.5726520207372612 , F1 Score=  0.5408474598884503
Max Depth=  10 , Accuracy Score=  0.5729098043708647 , F1 Score=  0.5406095352632063
```

We can see that for *Max Depth=6*, DecisionTreeClassifier is giving the best result. Lets build the Tree Classifier model with max depth=6

```
For Maximum Depth= 6 , Accuracy Score= 0.5705038237905651 , Jaccard Score= 0.5705038237
905651 , F1 Score=  0.5511519243168462

C:\Users\Tania\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:664: Futu
reWarning: jaccard_similarity_score has been deprecated and replaced with jaccard_scor
e. It will be removed in version 0.23. This implementation has surprising behavior for
binary and multiclass classification tasks.
  FutureWarning)
```

Logistic Regression

Because our dataset only provides us with two severity code outcomes, our model will only predict one of those two classes. This makes our data binary, which is perfect to use with logistic regression.

```
For solver: lbfgs , Log loss=  0.6664525059380347 , F1 Score=  0.5492367655690457
For solver: saga , Log loss=  0.6664524760862334 , F1 Score=  0.5492367655690457
For solver: liblinear , Log loss=  0.6664528829319477 , F1 Score=  0.5492367655690457
For solver: newton-cg , Log loss=  0.6664524759351989 , F1 Score=  0.5492367655690457
For solver: sag , Log loss=  0.6664526056433632 , F1 Score=  0.5492367655690457
```

We can see that for all solvers, logistic regression is giving the same result. We will choose *liblinear* method for logistic regression. Lets see the effect of regularization on this model.

```
For regularization: 0.001 , Log loss=  0.666559065215783 , F1 Score=  0.549103480550091
5
For regularization: 0.01 , Log loss=  0.6664528829319477 , F1 Score=  0.549236765569045
7
For regularization: 0.1 , Log loss=  0.6664443445515155 , F1 Score=  0.5492367655690457
```

For logistic regression, we will choose liblinear solver with *C=0.001*.

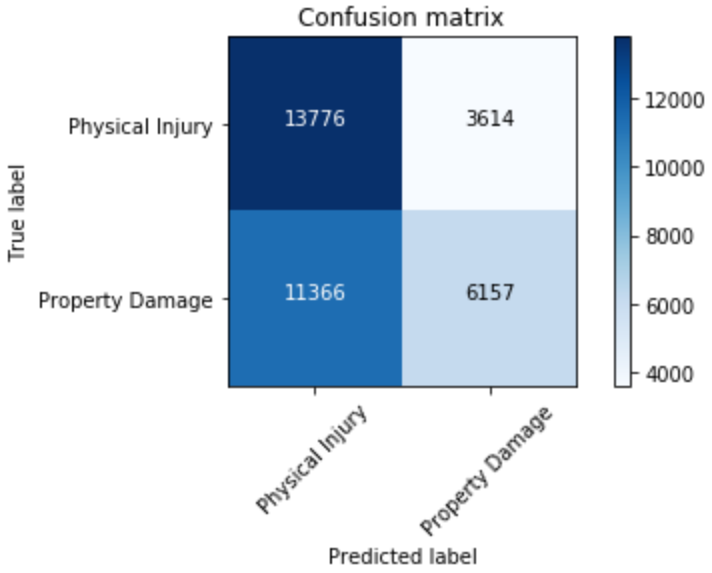
```
For liblinear: Accuracy Score=  0.5709334631799043 , Jaccard Score= 0.5709334631799043
, F1 Score=  0.5491034805500915

C:\Users\Tania\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:664: Futu
reWarning: jaccard_similarity_score has been deprecated and replaced with jaccard_scor
e. It will be removed in version 0.23. This implementation has surprising behavior for
binary and multiclass classification tasks.
  FutureWarning)


C:\Users\Tania\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:664: Futu
reWarning: jaccard_similarity_score has been deprecated and replaced with jaccard_scor
e. It will be removed in version 0.23. This implementation has surprising behavior for
binary and multiclass classification tasks.
  FutureWarning)
C:\Users\Tania\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:664: Futu
reWarning: jaccard_similarity_score has been deprecated and replaced with jaccard_scor
e. It will be removed in version 0.23. This implementation has surprising behavior for
binary and multiclass classification tasks.
  FutureWarning)
C:\Users\Tania\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:664: Futu
reWarning: jaccard_similarity_score has been deprecated and replaced with jaccard_scor
e. It will be removed in version 0.23. This implementation has surprising behavior for
binary and multiclass classification tasks.
  FutureWarning)
```

Results

Algorithm	Jaccard	F1-score	Logloss
KNN	0.54	0.53	NA
Decision Tree	0.57	0.55	NA
Logistic Regression	0.57	0.55	0.67

Based on Jaccard score, accuracy score and F1 score. Both the models Decision tree and Logistic regression are similar. But as the logloss is much better for logistic regression and this model is best for binary classification, we can conclude that logistic regression is the best classification model for the given problem.

```
(34913,)
(34913,)
Confusion matrix, without normalization
```



The above classification report shows that total number of physical injury is 17390 where the model identified 13776 such cases. Total number of property damage is 17523 where the model identified only 6157 cases correctly.

Classification report

	precision	recall	f1-score	support
1	0.63	0.35	0.45	17523
2	0.55	0.79	0.65	17390
accuracy			0.57	34913
macro avg	0.59	0.57	0.55	34913
weighted avg	0.59	0.57	0.55	34913

The precision, recall and f1 score for label 2 or Actual Phycal injury is good enough but both recall score and f1 score for the label 1 or Physical damage is very poor. That is why the average f1 score is not as high as expected even for a the logistic regression. We can conclude that, more datas are required to improve this model.

Discussions

In the beginning of this notebook, we had categorical data that was of type 'object'. This is not a data type that we could have fed through an algoritim, so label encoding was used to created new classes that were of type int8; a numerical data type.

After solving that issue we were presented with another - imbalanced data. As mentioned earlier, class 1 was nearly three times larger than class 2. The solution to this was downsampling the majority class with sklearn's resample tool. We downsampled to match the minority class exactly with 58188 values each.

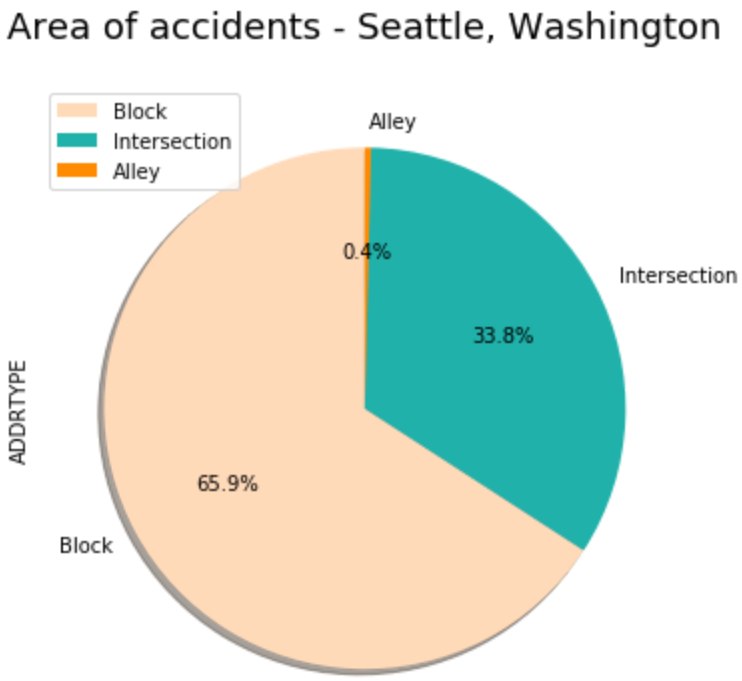
Once we analyzed and cleaned the data, it was then fed through three ML models; K-Nearest Neighbor, Decision Tree and Logistic Regression. Although the first two are ideal for this project, logistic regression made most sense because of its binary nature.

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

Recommendations

After assessing the data and the output of the Machine Learning models, a few recommendations can be made for the stakeholders. The developmental body for Seattle city can assess how much of these accidents have occurred in a place where road or light conditions were not ideal for that specific area and could launch development projects for those areas where most severe accidents take place in order to minimize the effects of these two factors. Whereas, the car drivers could also use this data to assess when to take extra precautions on the road under the given circumstances of light condition, road condition and weather, in order to avoid a severe accident, if any.

Public Development Authority of Seattle (PDAS)

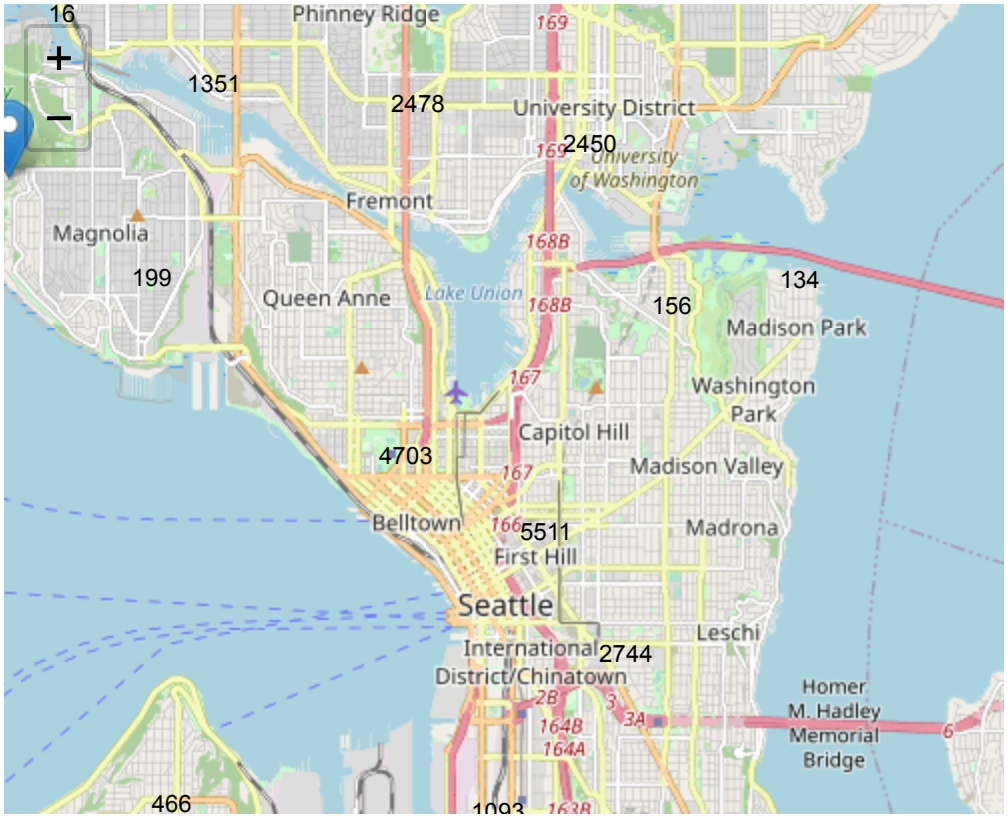


Almost all of the accidents recorded have occurred on either a block or an intersection, the PDAS can take the following measures in response car accidents:

1. Launch development projects for those areas where most severe accidents take place in order to minimize the effects of these two factors
2. Increased investment towards improving lighting and road conditions of the area which have high instances recorded
3. Install safety signs on the roads and ensure that all precautions are being taken by people within the area

Car Drivers

```
(37868, 38)
```



A higher concentration of accidents can be mostly seen on the main roads of the city, specifically near the highway in the city center. The following steps can be taken by car drivers to avoid severe accidents:

1. Be extra careful around the I-5 highway which goes through the city center since it has the highest proportion of accidents recorded of total seattle
2. Most incidents occur under adverse weather, road and light conditions. Precautions should be taken under such circumstances, for e.g. driving slow on a wet road which may lead to loss of control