Back ground

Objective of this project is to analyze the accident data from the year 2014. Data is collected from traffic department source which most authenticated data source. Data provided includes the severity of accident and other variables like Speed, Road condition, location, collision type, number of persons in the vehicle, weather condition etc. which could be dependent factors for the severity of the accident.

In the project considerable factor is severity of accident based on the level of fatality and property damage. We will analyze correlation between variable to understand the different factors impacting the severity of accident. Further will build a model to predict severity of accident based on the different variables like Weather, Road Condition, traffic condition etc.

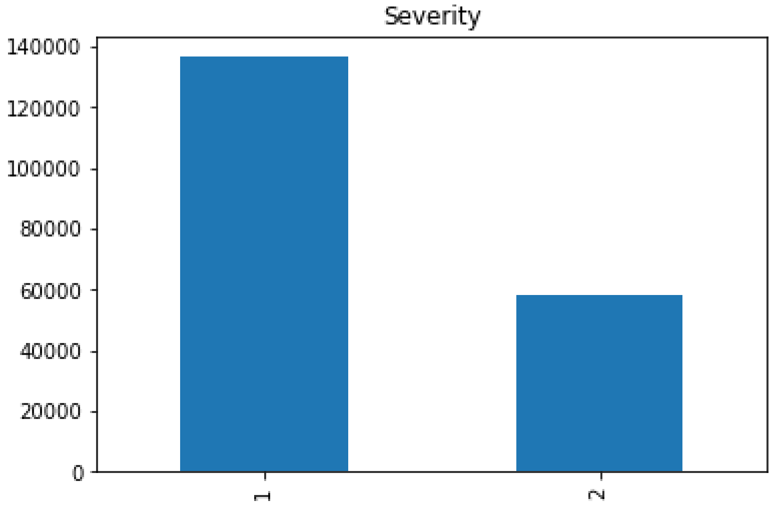
Data

There are 194,672 accident records available in the given data set. Our major consideration is SEVERITYCODE which given as 1-With property damage and 2 Injury of passenger. There are 136,485 Severity 1 accidents and 58,188 Severity 2 Accidents in the data set. We are considering Weather Condition, Road Condition, Light Condition and Number of people in the car and independent variable to analyze the dependent variable of accident Severity.

Methodologyd

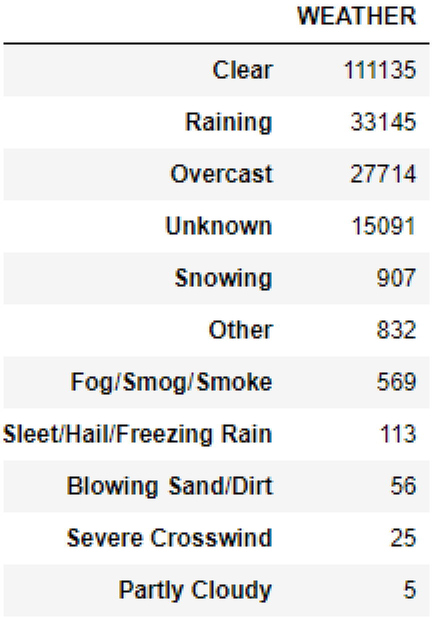
We used Jupiter note book Anaconda for data analysis. Data set has been moved to local computer. Imported various libraries including Pandas, Numpy, Matplotlib , Seaboarn and scikit-learn. Data narrowed down to required variables of Weather Condition, Road Condition, Light Condition and Number of passengers.

Most of the accident reported are severity 1– Property damage only.

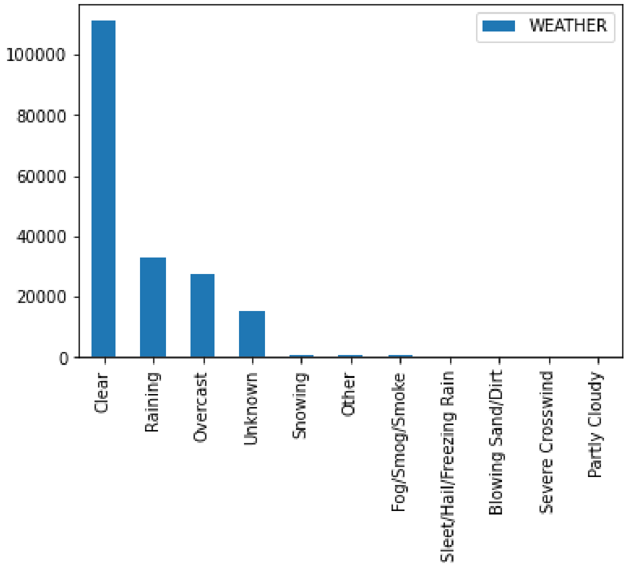


Exploratory data Analysis

First thing we checked is what is the Weather condition causes the majority of accidents. It is found most of the accident are in the Normal-Clear weather condition.

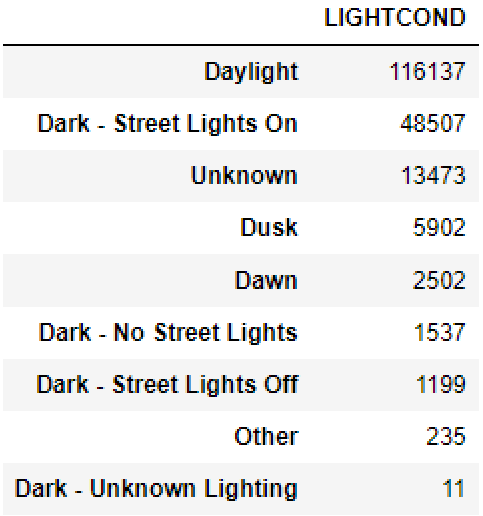


	WEATHER
Clear	111135
Raining	33145
Overcast	27714
Unknown	15091
Snowing	907
Other	832
Fog/Smog/Smoke	569
Sleet/Hail/Freezing Rain	113
Blowing Sand/Dirt	56
Severe Crosswind	25
Partly Cloudy	5

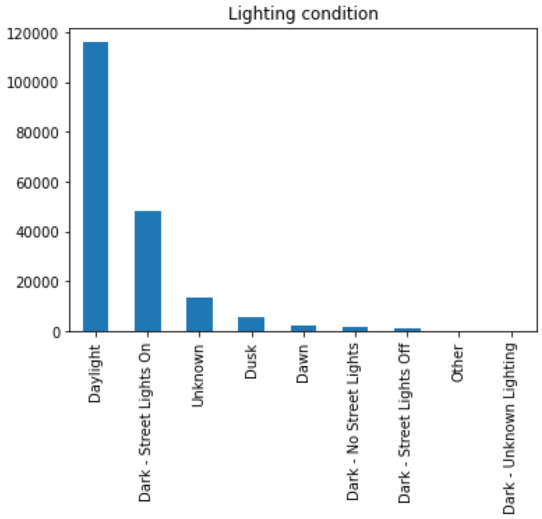


.

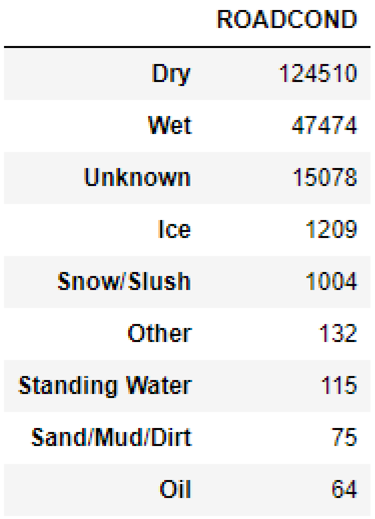
As a next step looked in to the data of light condition and found most of the accidents are happening in the day time and next is in the evening with the street light.



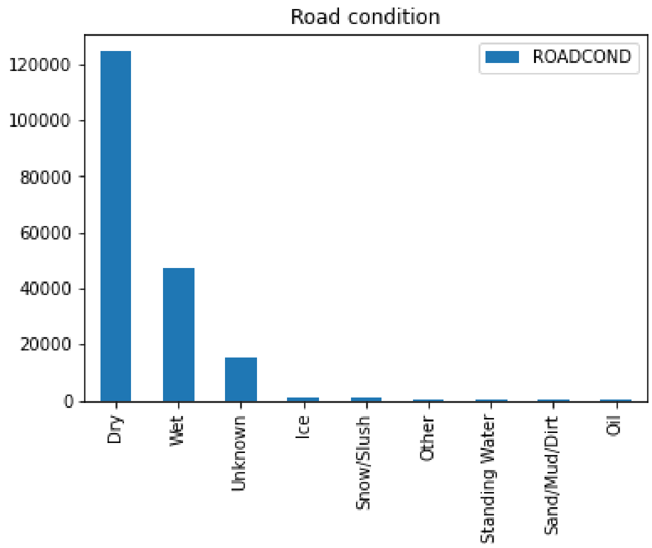
	LIGHTCOND
Daylight	116137
Dark - Street Lights On	48507
Unknown	13473
Dusk	5902
Dawn	2502
Dark - No Street Lights	1537
Dark - Street Lights Off	1199
Other	235
Dark - Unknown Lighting	11



Further analysed the road condition based on given data and found accidents are happening on normal dry roads and the other factors are not the major reason for the accidents.



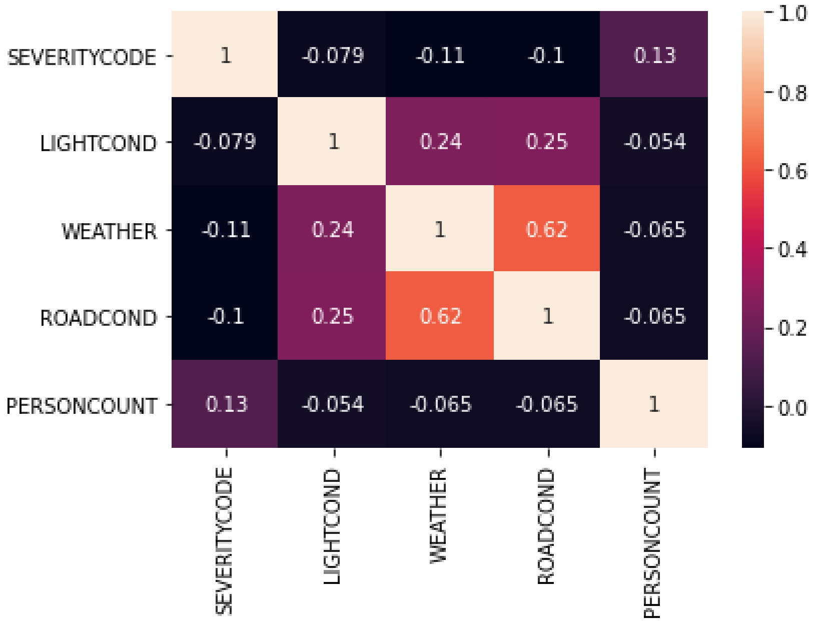
	ROADCOND
Dry	124510
Wet	47474
Unknown	15078
Ice	1209
Snow/Slush	1004
Other	132
Standing Water	115
Sand/Mud/Dirt	75
Oil	64



Based on the above information we have anlaysed is there any correlation between Weather, Road Condition, Light condition and Severity of accident.

PERSONCOUNT, LIGHTCOND, WEATHER, ROADCOND data are categorical varibales and hence it is convered to numberic values to check the corelation. Padas *replace* function is used to replace categorical values.

We plot the heat map based on LIGHTCOND, WEATHER, ROADCOND and SEVERITYCODE



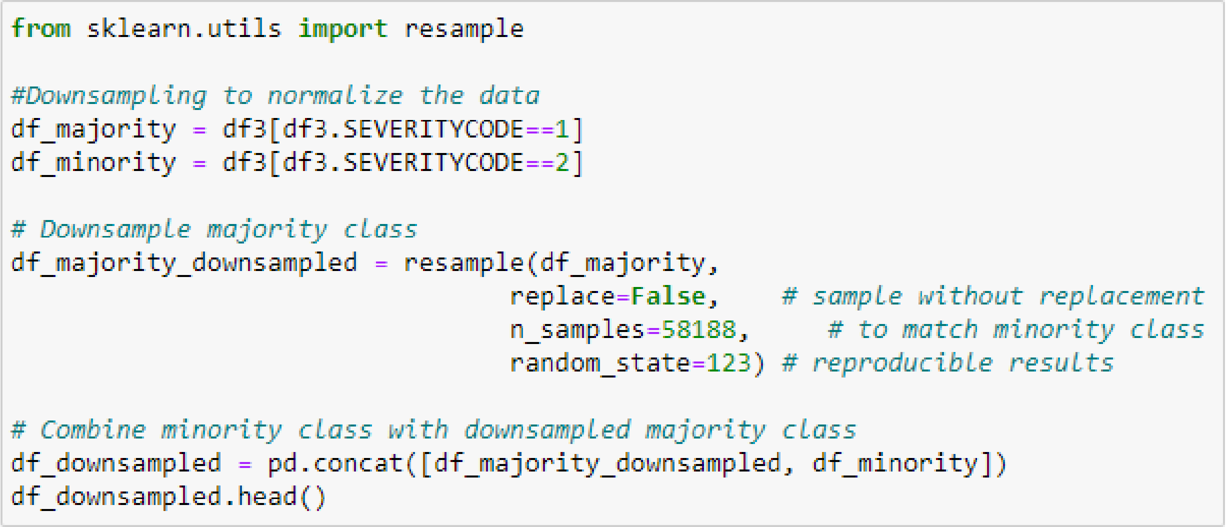
Looking in to the heat map, there is no significant corelation between LIGHTCOND , WEATHER, ROADCOND WITH SEVERITY. Only corilation is between weather and road condition that obivious.

Predictive Model.

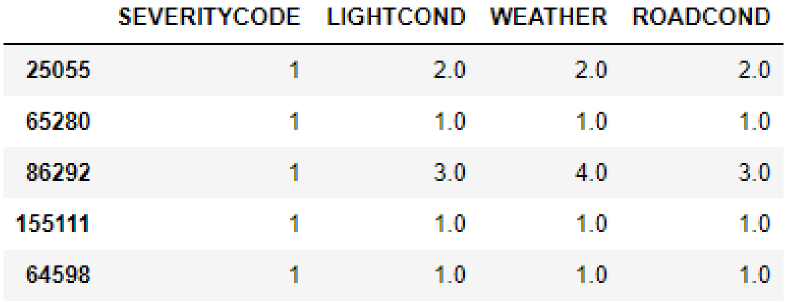
We have build the model to predict the SEVERITY based on the three indepedant variable LIGHTCOND, WEATHER and ROADCOND.

Normalizing the data set

It is found that the 29% of accidents are Severity 2 and 71% are Severity 1. Inorder to build the accurate model, it is to be normailized. We used downsampling for severity 1 cases to 58188, the count of Serverity 2.



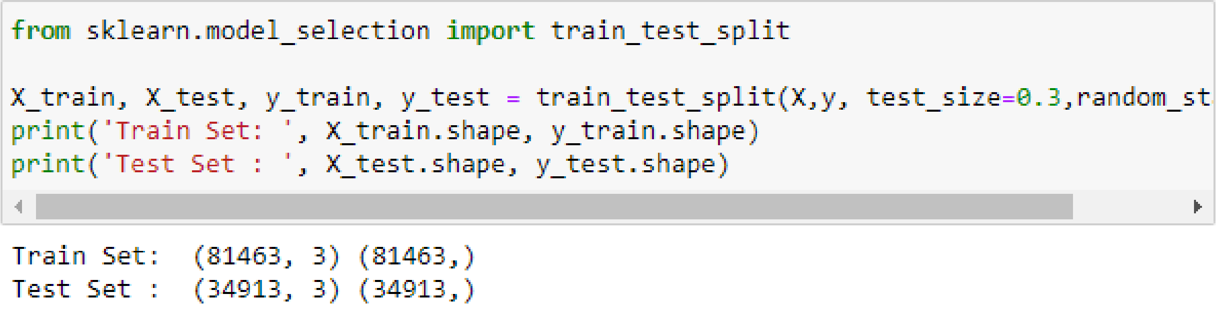
```
from sklearn.utils import resample

#Downsampling to normalize the data
df_majority = df3[df3.SEVERITYCODE==1]
df_minority = df3[df3.SEVERITYCODE==2]

# Downsample majority class
df_majority_downsampled = resample(df_majority,
                                 replace=False,    # sample without replacement
                                 n_samples=58188,     # to match minority class
                                 random_state=123) # reproducible results

# Combine minority class with downsampled majority class
df_downsampled = pd.concat([df_majority_downsampled, df_minority])
df_downsampled.head()
```



	SEVERITYCODE	LIGHTCOND	WEATHER	ROADCOND
25055	1	2.0	2.0	2.0
65280	1	1.0	1.0	1.0
86292	1	3.0	4.0	3.0
155111	1	1.0	1.0	1.0
64598	1	1.0	1.0	1.0

Train and test Split

The dataset has been split to 70% train set and 30% test.

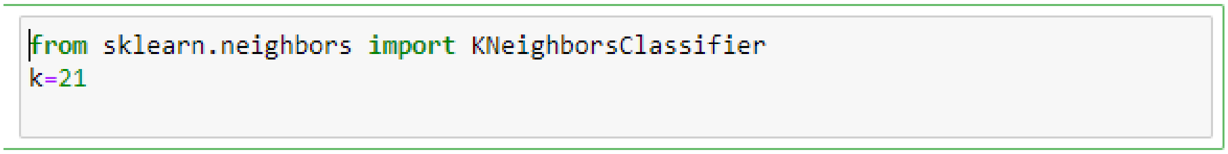


```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,random_st
print('Train Set: ', X_train.shape, y_train.shape)
print('Test Set : ', X_test.shape, y_test.shape)
```

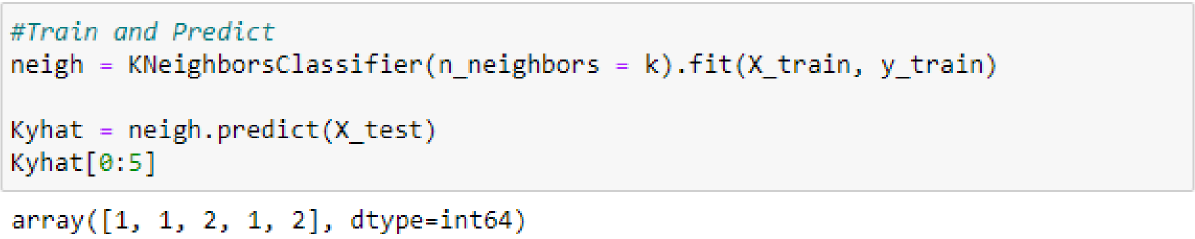
```
Train Set:  (81463, 3) (81463,)
Test Set :  (34913, 3) (34913,)
```

K-Nearest Neighbors (KNN)

Further model is build based on KNN alogorithm. Initally K has been set to 25 and tested the model . It was giving 54% accuracy for the test data. An itrated process excuted to find out the best K value and found 21 is the best K value with give 55%

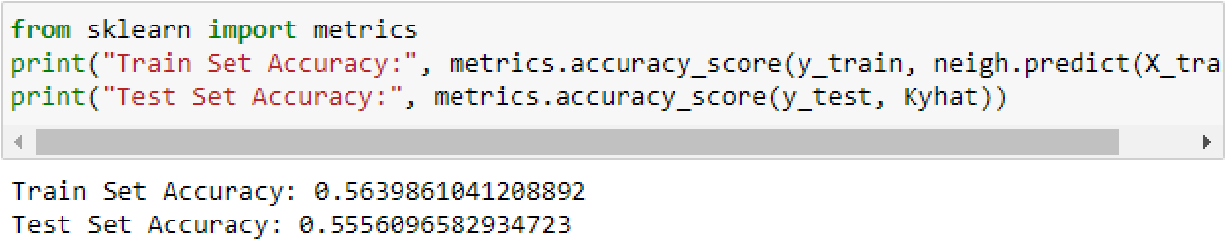


```
from sklearn.neighbors import KNeighborsClassifier
k=21
```



```
#Train and Predict
neigh = KNeighborsClassifier(n_neighbors = k).fit(X_train, y_train)

Kyhat = neigh.predict(X_test)
Kyhat[0:5]
```

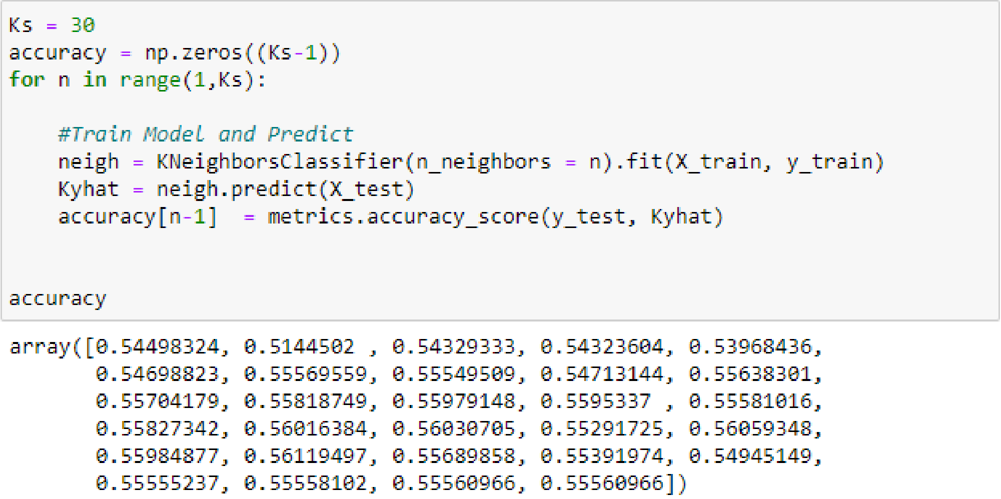
```
array([1, 1, 2, 1, 2], dtype=int64)
```



```
from sklearn import metrics
print("Train Set Accuracy:", metrics.accuracy_score(y_train, neigh.predict(X_tra
print("Test Set Accuracy:", metrics.accuracy_score(y_test, Kyhat))
```

```
Train Set Accuracy: 0.5639861041208892
Test Set Accuracy: 0.5556096582934723
```

KNN Accuracy :

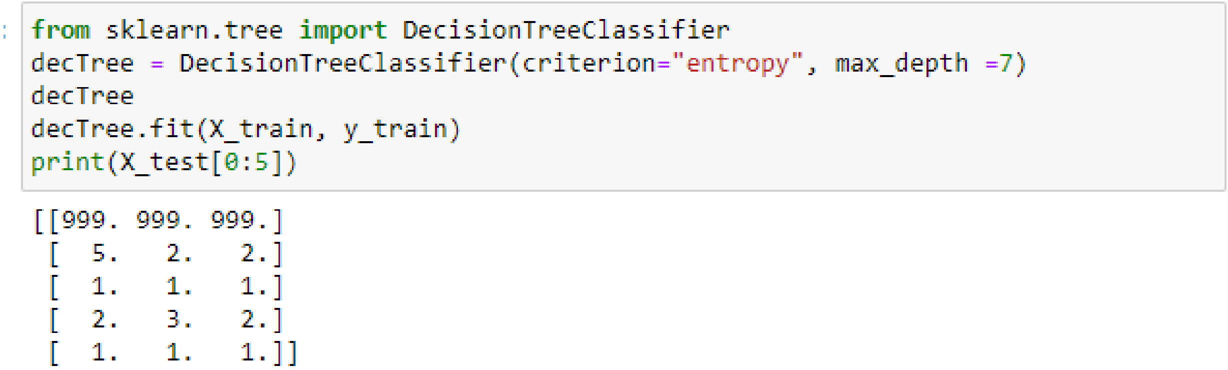


```
Ks = 30
accuracy = np.zeros((Ks-1))
for n in range(1,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train, y_train)
    Kyhat = neigh.predict(X_test)
    accuracy[n-1]  = metrics.accuracy_score(y_test, Kyhat)


accuracy
```

```
array([0.54498324, 0.5144502 , 0.54329333, 0.54323604, 0.53968436,
       0.54698823, 0.55569559, 0.55549509, 0.54713144, 0.55638301,
       0.55704179, 0.55818749, 0.55979148, 0.5595337 , 0.55581016,
       0.55827342, 0.56016384, 0.56030705, 0.55291725, 0.56059348,
       0.55984877, 0.56119497, 0.55689858, 0.55391974, 0.54945149,
       0.55555237, 0.55558102, 0.55560966, 0.55560966])
```

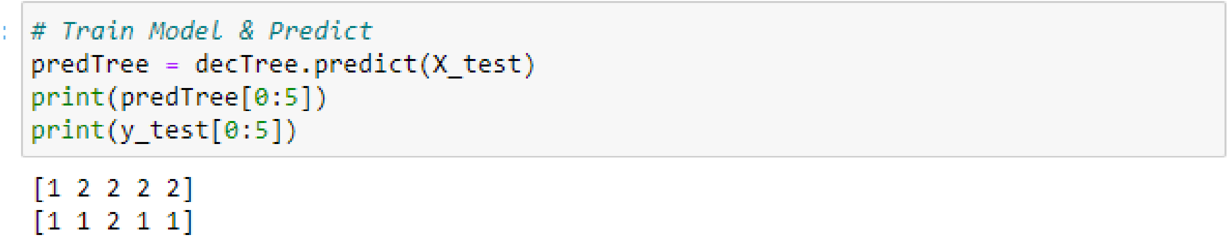
Decision Tree

Another model is build based on Decision tree algorithm. A decision tree model gives us a layout of all possible outcomes so we can fully analyze the the output. It context, the decision tree observes all possible outcomes of different weather conditions, light and road condition.



```
from sklearn.tree import DecisionTreeClassifier
decTree = DecisionTreeClassifier(criterion="entropy", max_depth =7)
decTree
decTree.fit(X_train, y_train)
print(X_test[0:5])
```

```
[[999. 999. 999.]
 [  5.   2.   2.]
 [  1.   1.   1.]
 [  2.   3.   2.]
 [  1.   1.   1.]]
```

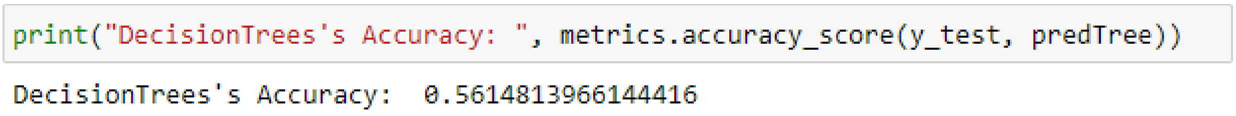


```
# Train Model & Predict
predTree = decTree.predict(X_test)
print(predTree[0:5])
print(y_test[0:5])
```

```
[1 2 2 2 2]
[1 1 2 1 1]
```

Decision Tree Accuracy

Found the decision tree is 56% accurate.

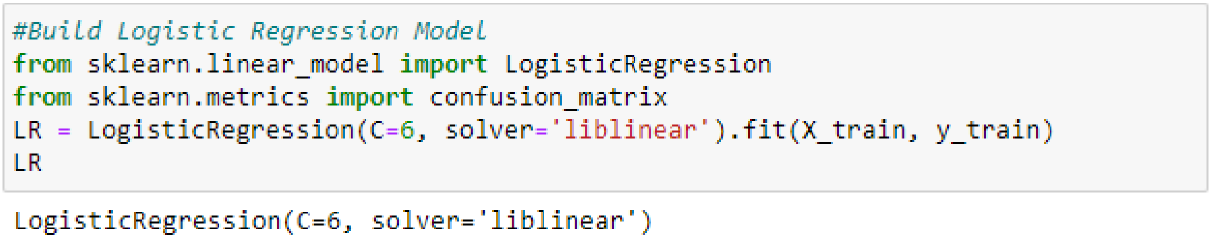


```
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test, predTree))
```

```
DecisionTrees's Accuracy:  0.5614813966144416
```

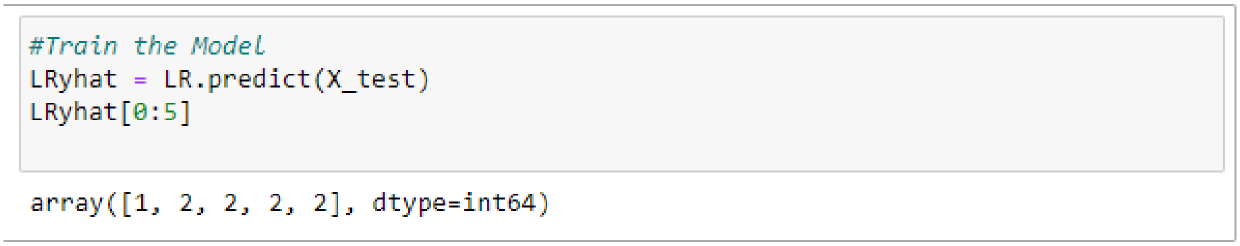
Logistic Regression

Since our dataset only provides 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.



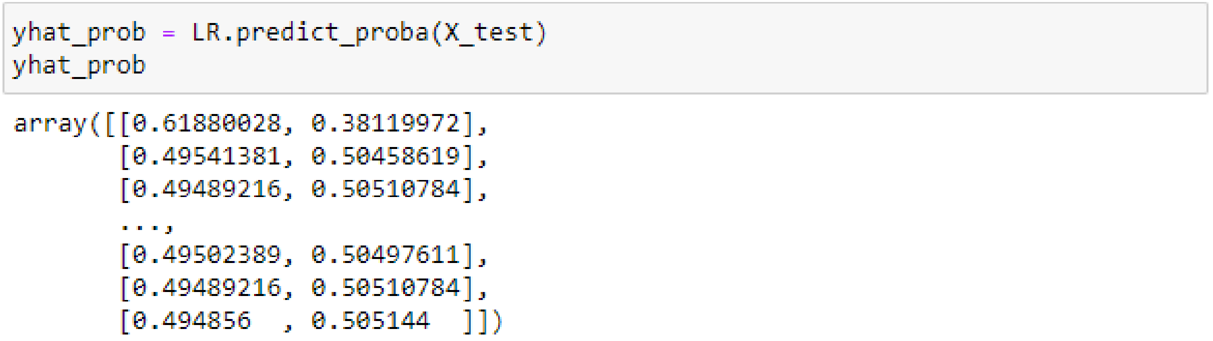
```
#Build Logistic Regression Model
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
LR = LogisticRegression(C=6, solver='liblinear').fit(X_train, y_train)
LR
```

```
LogisticRegression(C=6, solver='liblinear')
```



```
#Train the Model
LRyhat = LR.predict(X_test)
LRyhat[0:5]
```

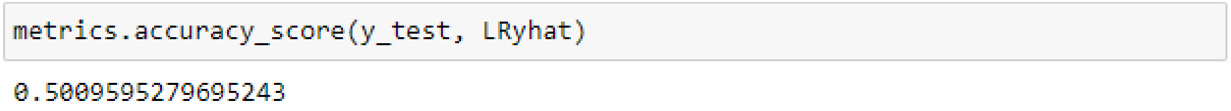
```
array([1, 2, 2, 2, 2], dtype=int64)
```



```
yhat_prob = LR.predict_proba(X_test)
yhat_prob
```

```
array([[0.61880028, 0.38119972],
       [0.49541381, 0.50458619],
       [0.49489216, 0.50510784],
       ...,
       [0.49502389, 0.50497611],
       [0.49489216, 0.50510784],
       [0.494856  , 0.505144  ]])
```

Logistic Regression Accuracy



```
metrics.accuracy_score(y_test, LRyhat)
```

```
0.5009595279695243
```

Discussion

Looking in to the data, initial understanding is Weather condition could be the major reason for the accidents. After analyzing by plotting heatmap, it is found there is no major correlation between weather, light and road condition and is not the major cause of accidents. Traffic team should consider looking into other data cause the accidents.

After downsampling and balancing the data, it was 70% of balanced data feeded to the training set. Three alogorith KNN, Decision Tree and Logistic regression being used to train the model. Since the target variable is binary in nature logistic regression gives the best out prediction output.

Conclusion

There is no information available about the traffic - *Number of vehicles passing through each street during different condition of weather, light and road*. If we have this information, we could been analyzed the number of accidents happening in different *Weather, Lighting* condition in comparing to the traffics in the road.

Dark with street light on is the second highest accidents, Light condition is one of the major reason for the accidents and to be analyzed based on the number of vehicles passed and number of accidents in comparison with the daylight.

We can conclude weather condition and light condition is having an impact on the accidents. Adding more light in the dark locations could reduce the number of accidents. More data to be collected about the traffic during different weather and light condition and to be anazlyzed.