Car Accident Severity

Introduction to the Problem

- Weather acts through visibility impairments, precipitation, high winds, and temperature extremes to affect driver capabilities, vehicle performance.
- This study tries to determine the correlation between weather conditions and traffic accident occurrences by analyzing collected data. Data analysis is one of the activities of data science focused on obtaining important information from collected data
- To reduce the frequency of car collisions in a community, an algorithm must be developed to predict the severity of an accident given the current weather, road and visibility conditions. When conditions are bad, this model will alert drivers to remind them to be more careful or possible change travel date, mode or time, so that it would be safer.

Target Audience

- Daily commuters, who have travel on a regular basis for work. This project would warn them of possible dangers while driving based on the weather condition, road conditions and the time of the day
- It would also help the first responders like ambulance and firefighter services. The system could help them stay prepared in advance by indicating them of high chances of accidents that could take place.
- Thus, ensuring anyone in need could receive their help without much delay

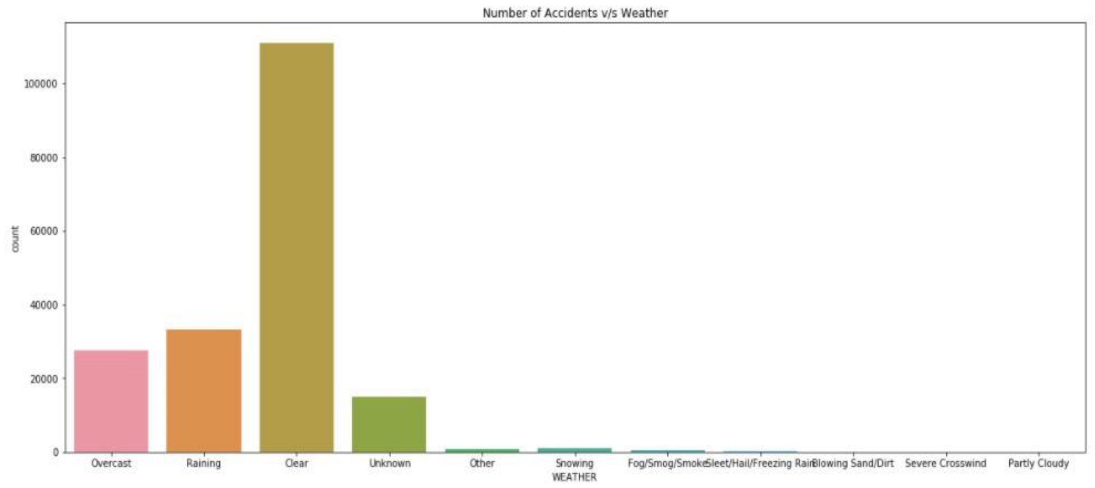
Data Understanding

- The data has been retrieved and processed through various sources, and database.
- The main source being the data-collisions csv file
- Our predictor or target variable will be 'SEVERITYCODE' because it is used measure the severity of an accident from 0 to 5 within the dataset. Attributes used to weigh the severity of an accident are 'WEATHER', 'ROADCOND' and 'LIGHTCOND'

Number of Accidents v/s Weather:

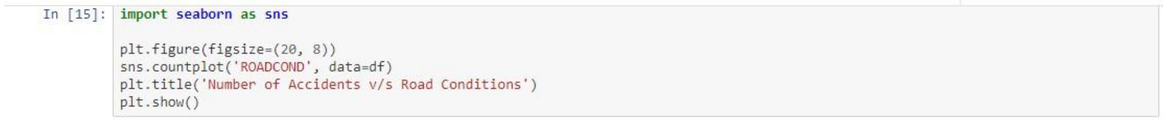
- The graph determines the most of accidents take place in clear weather, and other weather conditions which also cause some of the accidents are Overcast and Rain.
- So we can ignore the other weather conditions
- Overcast: Overcast sky conditions occur when clouds cover all or most of the sky and cause low visibility conditions
- Rains: Heavy or moderate rainfall, which causes roads to be slippery
- Clear: Clear weather conditions

```
sns.countplot('WEATHER', data=df)
plt.title('Number of Accidents v/s Weather')
plt.show()
```

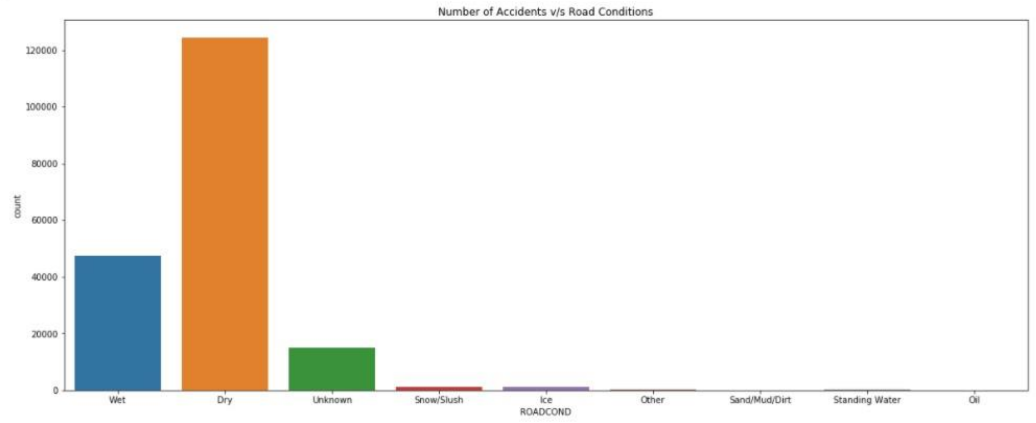


Number of Accidents v/s Road Conditions:

- Most of the accidents took place on dry road conditions, and the other condition which also caused a few accidents was wet road conditions, other conditions can be ignored
- This attribute refers to the road condition for a day
- Wet: Usually refers to wet and snowy conditions on a day
- Dry: Normal road conditions

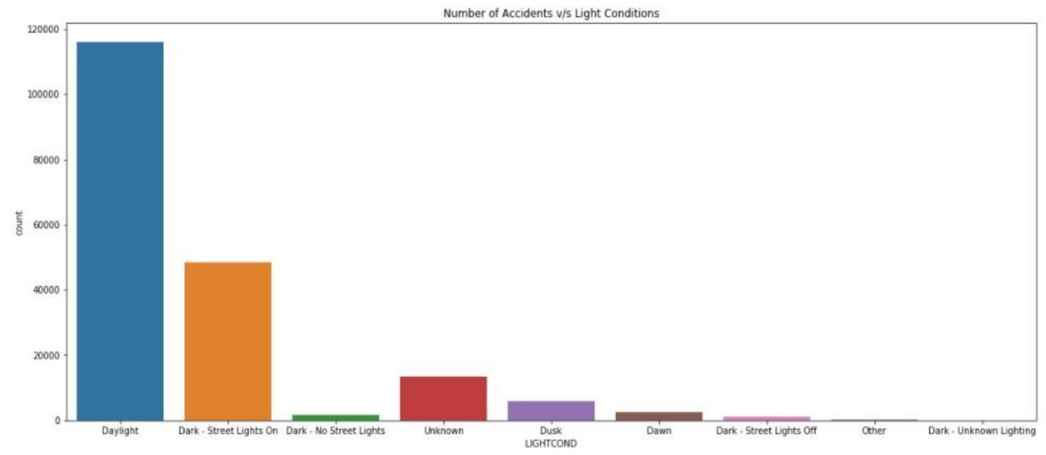


```
In [15]: import seaborn as sns

         plt.figure(figsize=(20, 8))
         sns.countplot('ROADCOND', data=df)
         plt.title('Number of Accidents v/s Road Conditions')
         plt.show()
```



Number of Accidents v/s Light Conditions:

- Most of the accidents were caused during daylight, the other major cause of accidents was Dark with streetlights on, the other features can be ignored
- This attribute gives information of light conditions when the accident took place and will be useful in predicting in what conditions an accident is probable.
- Daylight: Indicates daylight conditions
- Dark with Street Lights on: Dark conditions but streetlight source was present
- Dark without Street Lights: Pitch dark conditions, only head lights to guide the way
- Dawn: Early Morning, day starts getting more light
- Dusk: late evening, light from the day reduces



Data Balancing and Cleaning

Cleaning

Data is unbalanced and cannot be directly used for analysis

In its original form, this data is not fit for analysis. For one, there are many columns that we will not use for this model. Also, most of the features are of type object when they should be numerical type.

We must use label encoding to covert the features to our desired data type.

Data Understanding

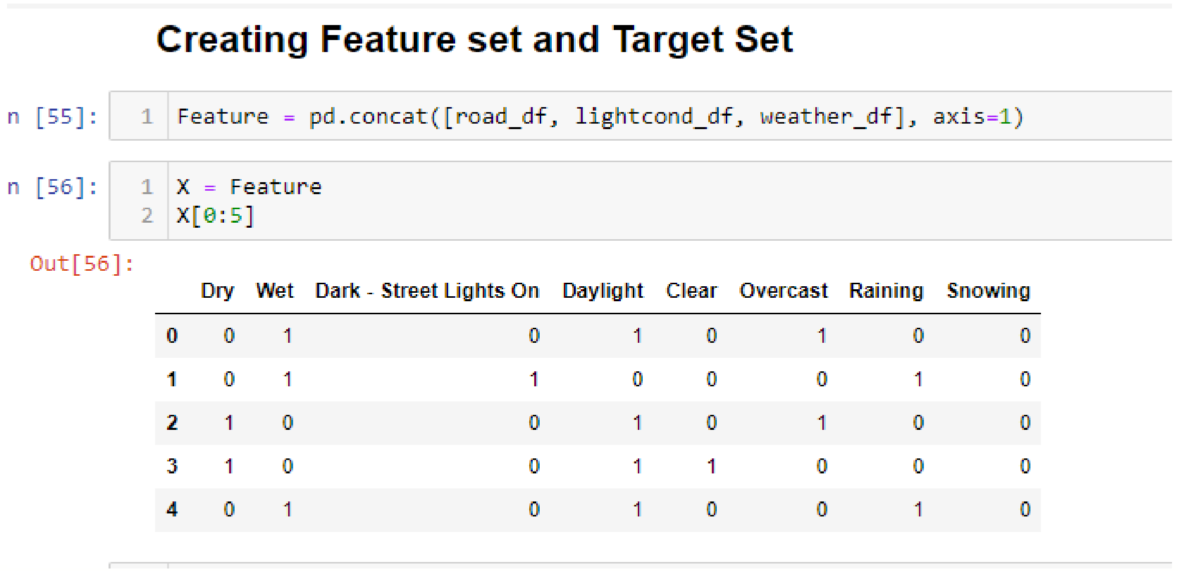
Data Cleaning is an essential step because we must identify which data features are important for our analysis and which attributes are not.

So, it is essential that we go through each attribute carefully and determine how important it is and whether it can be used

Data Preprocessing

Since most of the data like 'WEATHER', 'ROADCOND', 'LIGHTCOND' are all categoric, since most of the data like 'WEATHER', 'ROADCOND', 'LIGHTCOND' are all categoric

They must be converted to numeric type so they can be fed as feature set for Classifier Models:



Creating Feature set and Target Set

```
n [55]:  Feature = pd.concat([road_df, lightcond_df, weather_df], axis=1)
```

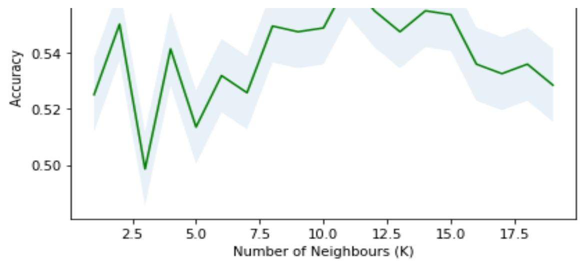
```
n [56]:  X = Feature
         X[0:5]
```

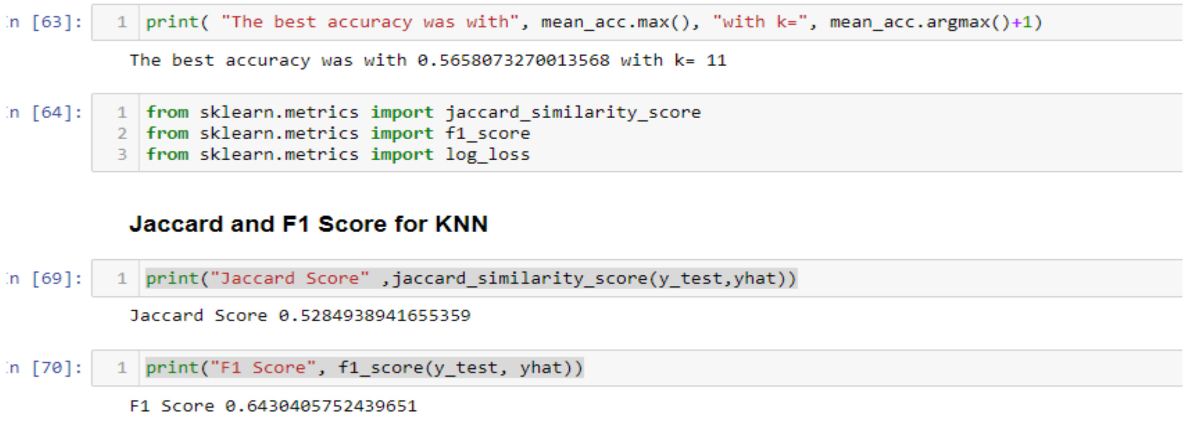
Out[56]:

	Dry	Wet	Dark - Street Lights On	Daylight	Clear	Overcast	Raining	Snowing
0	0	1	0	1	0	1	0	0
1	0	1	1	0	0	0	1	0
2	1	0	0	1	0	1	0	0
3	1	0	0	1	1	0	0	0
4	0	1	0	1	0	0	1	0

KNN Model

- KNN yielded:
- Jaccard Score: 0.528
- F1 Score: 0.64
- K for best Accuracy: 11





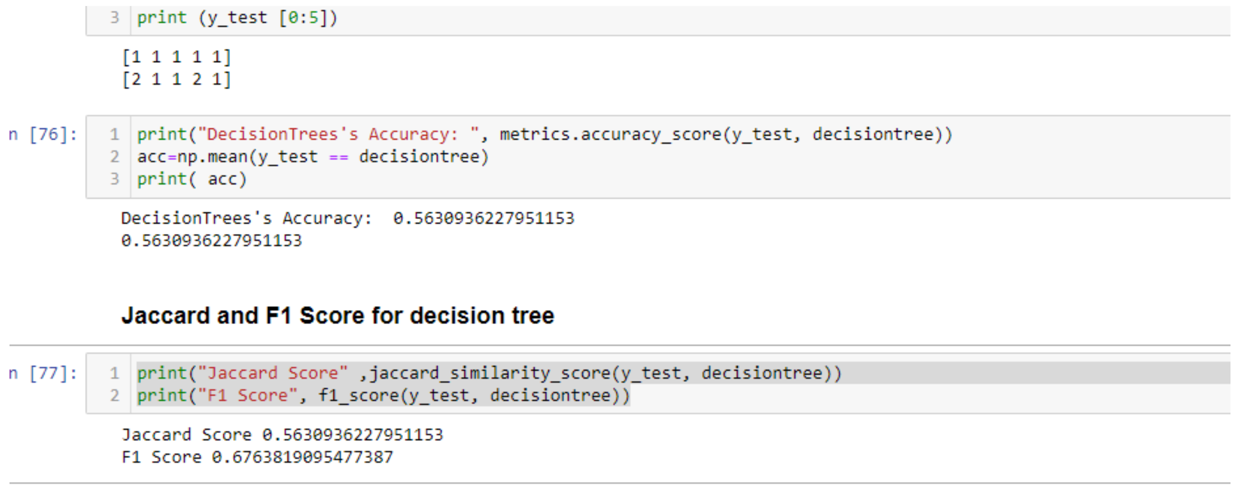
```
In [63]: print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)

The best accuracy was with 0.5658073270013568 with k= 11

In [64]: from sklearn.metrics import jaccard_similarity_score
         from sklearn.metrics import f1_score
         from sklearn.metrics import log_loss
```

Jaccard and F1 Score for KNN

```
In [69]: print("Jaccard Score" ,jaccard_similarity_score(y_test,yhat))

Jaccard Score 0.5284938941655359

In [70]: print("F1 Score", f1_score(y_test, yhat))

F1 Score 0.6430405752439651
```

Decision Tree

- Decision Tree yielded:
- Jaccard Score: 0.563
- F1 Score: 0.676



```
print (y_test [0:5])
```

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

n [76]:
```
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test, decisiontree))
acc=np.mean(y_test == decisiontree)
print( acc)
```

```
DecisionTrees's Accuracy:  0.5630936227951153
0.5630936227951153
```

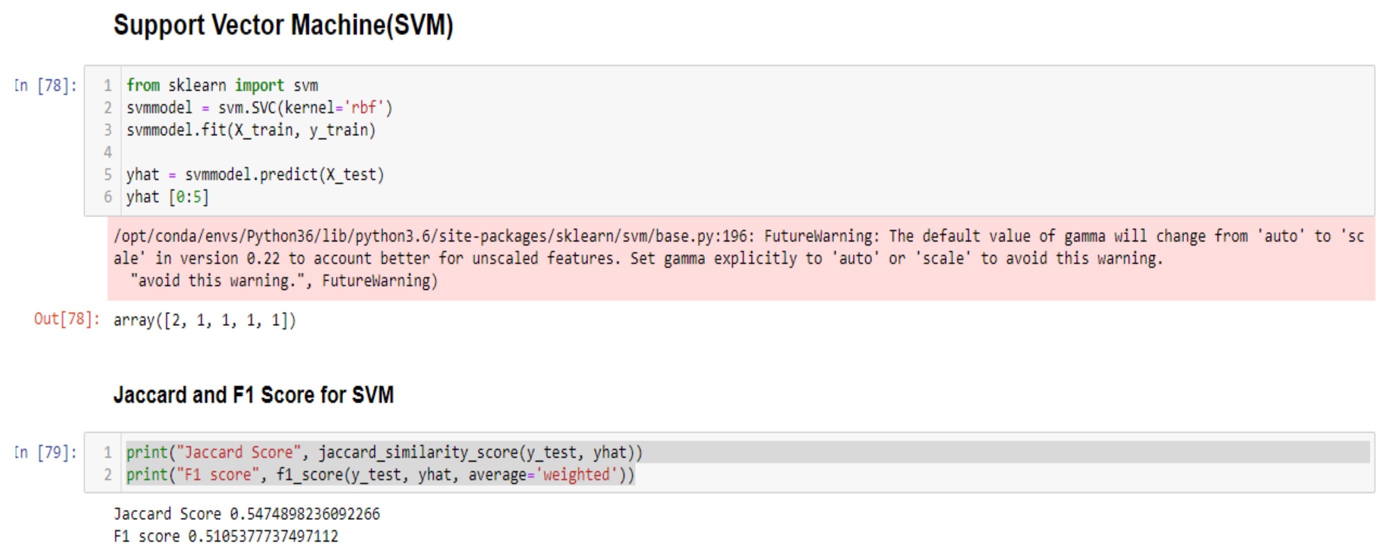
Jaccard and F1 Score for decision tree

n [77]:
```
print("Jaccard Score" ,jaccard_similarity_score(y_test, decisiontree))
print("F1 Score", f1_score(y_test, decisiontree))
```

```
Jaccard Score 0.5630936227951153
F1 Score 0.6763819095477387
```

Support Vector Machine(SVM):

- SVM yielded:
- Jaccard Score: 0.547
- F1 Score: 0.501



Support Vector Machine(SVM)

In [78]:
```
from sklearn import svm
svmmodel = svm.SVC(kernel='rbf')
svmmodel.fit(X_train, y_train)

yhat = svmmodel.predict(X_test)
yhat [0:5]
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/svm/base.py:196: FutureWarning: The default value of gamma will change from 'auto' to 'sc
ale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[78]:
```
array([2, 1, 1, 1, 1])
```

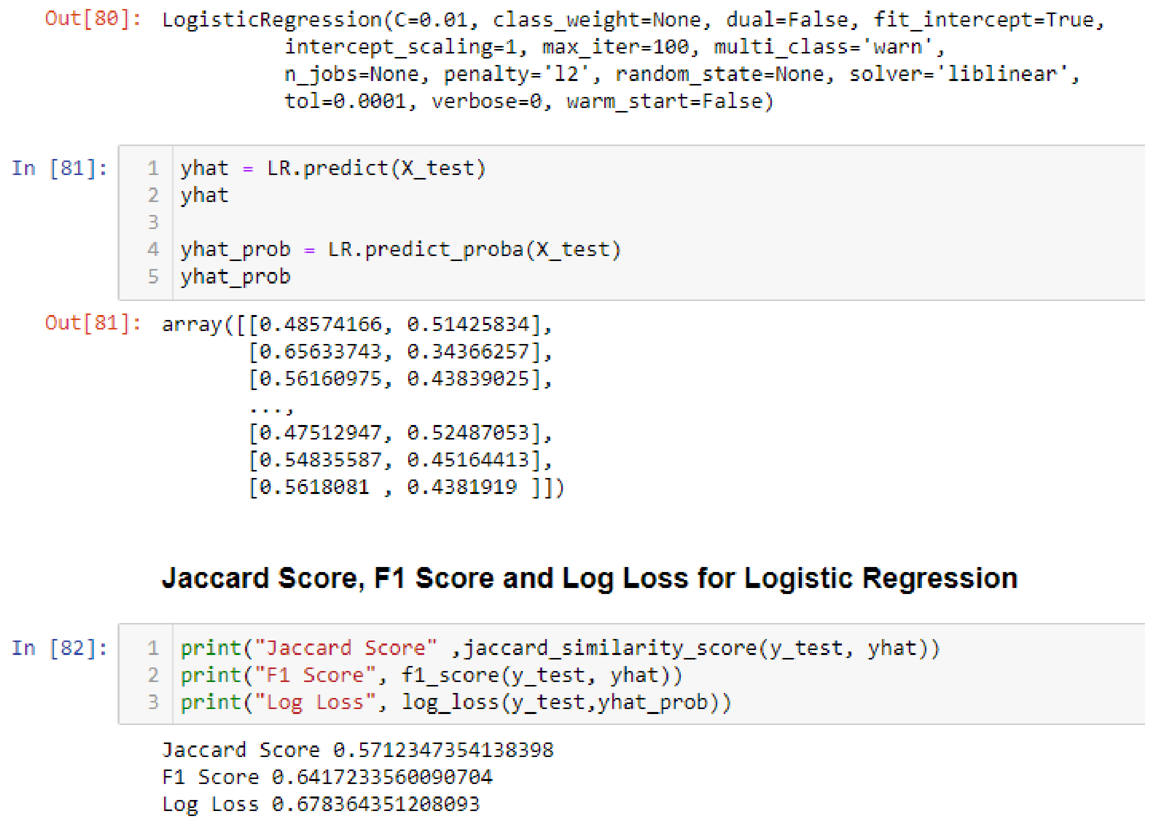
Jaccard and F1 Score for SVM

In [79]:
```
print("Jaccard Score", jaccard_similarity_score(y_test, yhat))
print("F1 score", f1_score(y_test, yhat, average='weighted'))
```

```
Jaccard Score 0.5474898236092266
F1 score 0.5105377737497112
```

Logistic Regression:

- Logistic Regression Yielded:
- Jaccard Score: 0.5712
- F1 Score: 0.6417
- Log Loss: 0.678



```
Out[80]: 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)
```

```
In [81]: yhat = LR.predict(X_test)
         yhat

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

```
Out[81]: array([[0.48574166, 0.51425834],
                [0.65633743, 0.34366257],
                [0.56160975, 0.43839025],
                ...,
                [0.47512947, 0.52487053],
                [0.54835587, 0.45164413],
                [0.5618081 , 0.4381919 ]])
```

Jaccard Score, F1 Score and Log Loss for Logistic Regression

```
In [82]: print("Jaccard Score" ,jaccard_similarity_score(y_test, yhat))
         print("F1 Score", f1_score(y_test, yhat))
         print("Log Loss", log_loss(y_test,yhat_prob))
```

```
Jaccard Score 0.5712347354138398
F1 Score 0.6417233560090704
Log Loss 0.678364351208093
```

CONCLUSION

USING THE EXISTING DATASET OF THE COURSE, SOME REMARKABLE INSIGHTS HAVE BEEN OBTAINED. AT THE BEGINNING IT WAS GUESSED, THAT THE SEVERITY OF AN ACCIDENT COULD BE PREDICTED BY THE WEATHER OR SPEEDING CONDITIONS. USING DIFFERENT METHODS FOR ESTIMATION OF THE SEVERITY BASED ON THE EXISTING DATASET IT COULD BE OBSERVED, THAT ONLY SMALL AMOUNT OF INFORMATION CAN BE GAINED BY PREDICTIONS.