Car Accident Severity

By Abhishek Vasant Girkar

Introduction

Average number of car accidents in the **U.S. every year is 6 million. 3 million people in the U.S. are injured every year in car accidents**.

Analysing the conditions that contribute to these accidents would lead to the prevention of significant loss of life and financial resources.

The project is aimed at predicting the severity of a car accident given the location, weather, and road and visibility conditions in order to reduce the frequency of car collusions in a community based on dataset provided by Seattle PD.

Data

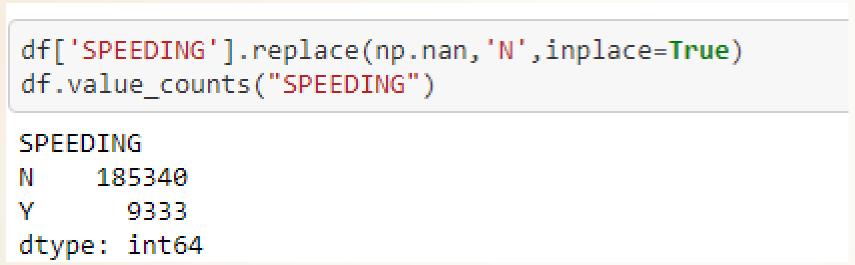
- Dataset provided by the Seattle Department of Transportation Traffic Management Division with 194673 rows (accidents) and 37 columns (features) where each accident is given a severity code.

- *Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model*

Here We considered Following attributes

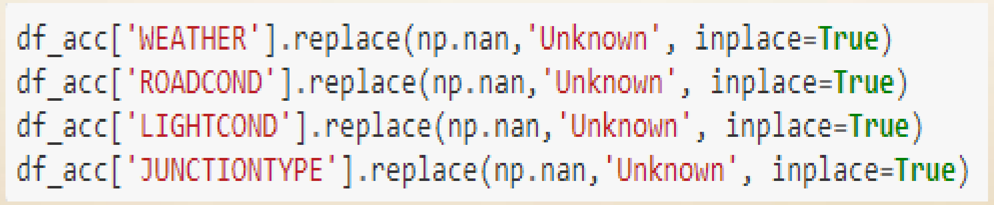
SEVERITYCODE	**ADDRTYPE**	**LOCATION**	**JUNCTIONTYPE**	**WEATHER**	**ROADCOND**	**LIGHTCOND**	**SPEEDING**

Replace **NAN** Value to achieve Good Accuracy

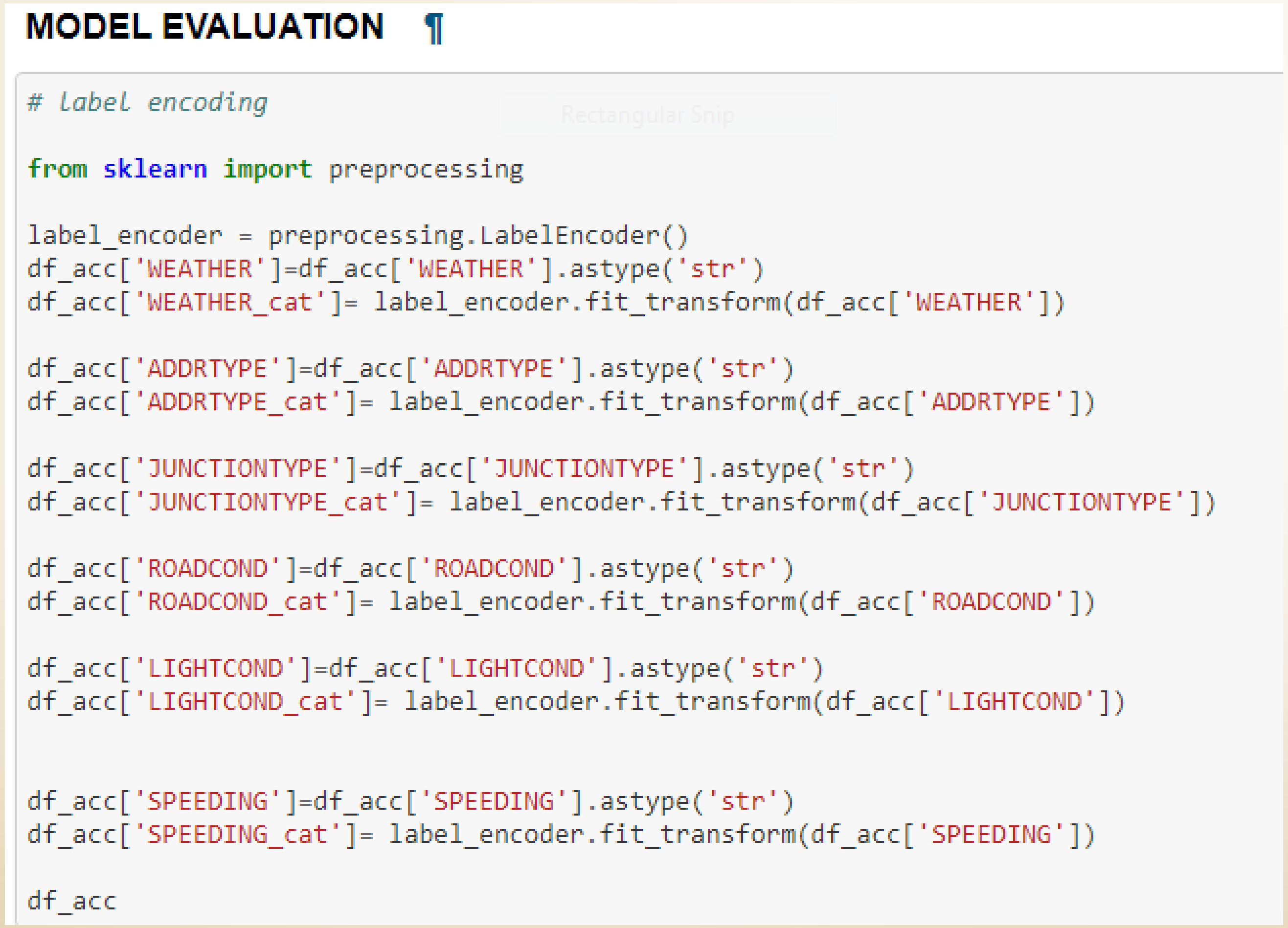


```
df['SPEEDING'].replace(np.nan,'N',inplace=True)
df.value_counts("SPEEDING")
```

```
SPEEDING
N    185340
Y      9333
dtype: int64
```



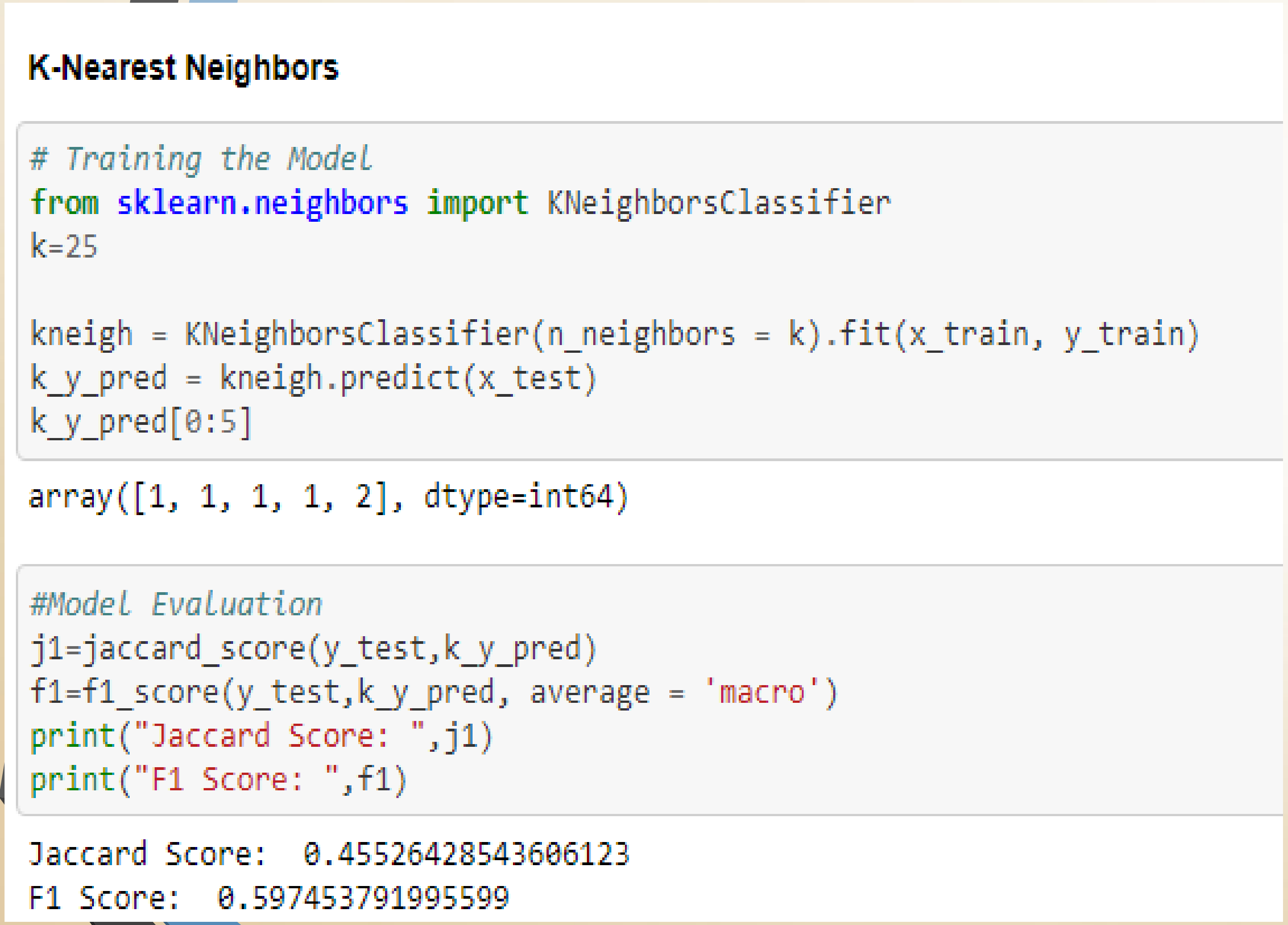
```
df_acc['WEATHER'].replace(np.nan,'Unknown', inplace=True)
df_acc['ROADCOND'].replace(np.nan,'Unknown', inplace=True)
df_acc['LIGHTCOND'].replace(np.nan,'Unknown', inplace=True)
df_acc['JUNCTIONTYPE'].replace(np.nan,'Unknown', inplace=True)
```



MODEL EVALUATION ¶

```
# label encoding

from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()
df_acc['WEATHER']=df_acc['WEATHER'].astype('str')
df_acc['WEATHER_cat']= label_encoder.fit_transform(df_acc['WEATHER'])

df_acc['ADDRTYPE']=df_acc['ADDRTYPE'].astype('str')
df_acc['ADDRTYPE_cat']= label_encoder.fit_transform(df_acc['ADDRTYPE'])

df_acc['JUNCTIONTYPE']=df_acc['JUNCTIONTYPE'].astype('str')
df_acc['JUNCTIONTYPE_cat']= label_encoder.fit_transform(df_acc['JUNCTIONTYPE'])

df_acc['ROADCOND']=df_acc['ROADCOND'].astype('str')
df_acc['ROADCOND_cat']= label_encoder.fit_transform(df_acc['ROADCOND'])

df_acc['LIGHTCOND']=df_acc['LIGHTCOND'].astype('str')
df_acc['LIGHTCOND_cat']= label_encoder.fit_transform(df_acc['LIGHTCOND'])


df_acc['SPEEDING']=df_acc['SPEEDING'].astype('str')
df_acc['SPEEDING_cat']= label_encoder.fit_transform(df_acc['SPEEDING'])

df_acc
```

KNN Algorithms



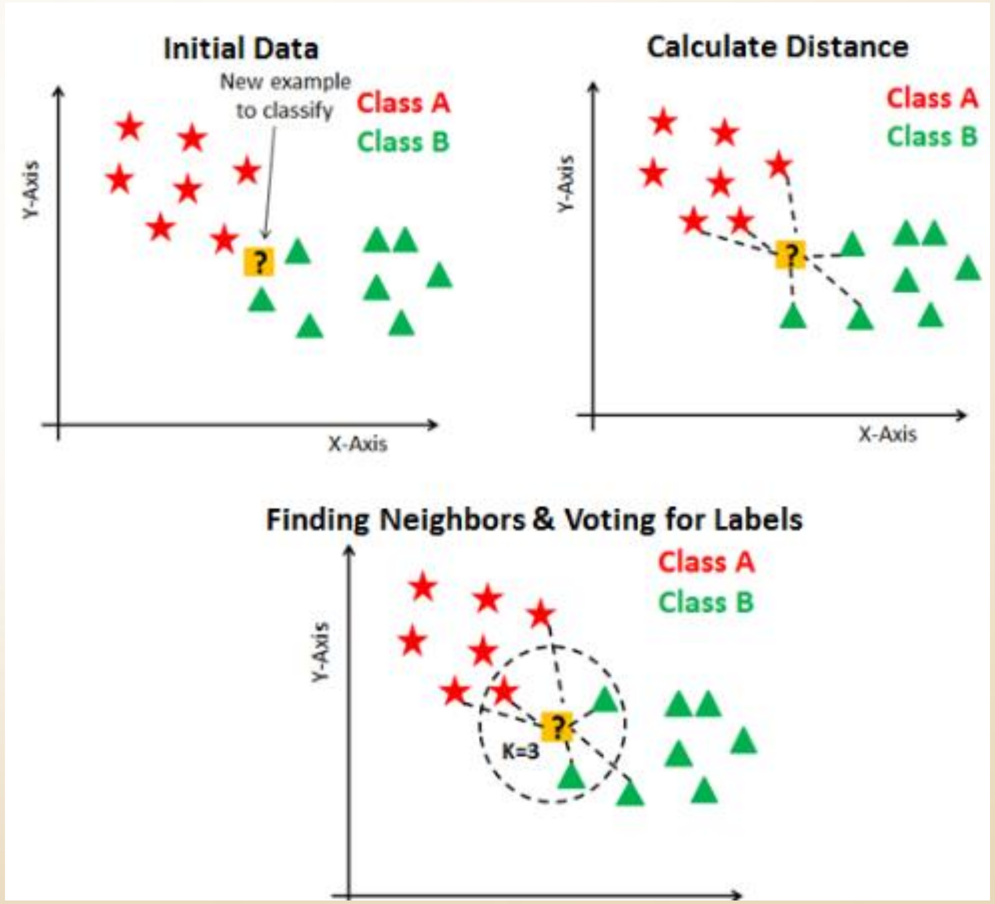
K-Nearest Neighbors

```
# Training the Model
from sklearn.neighbors import KNeighborsClassifier
k=25

kneigh = KNeighborsClassifier(n_neighbors = k).fit(x_train, y_train)
k_y_pred = kneigh.predict(x_test)
k_y_pred[0:5]
```

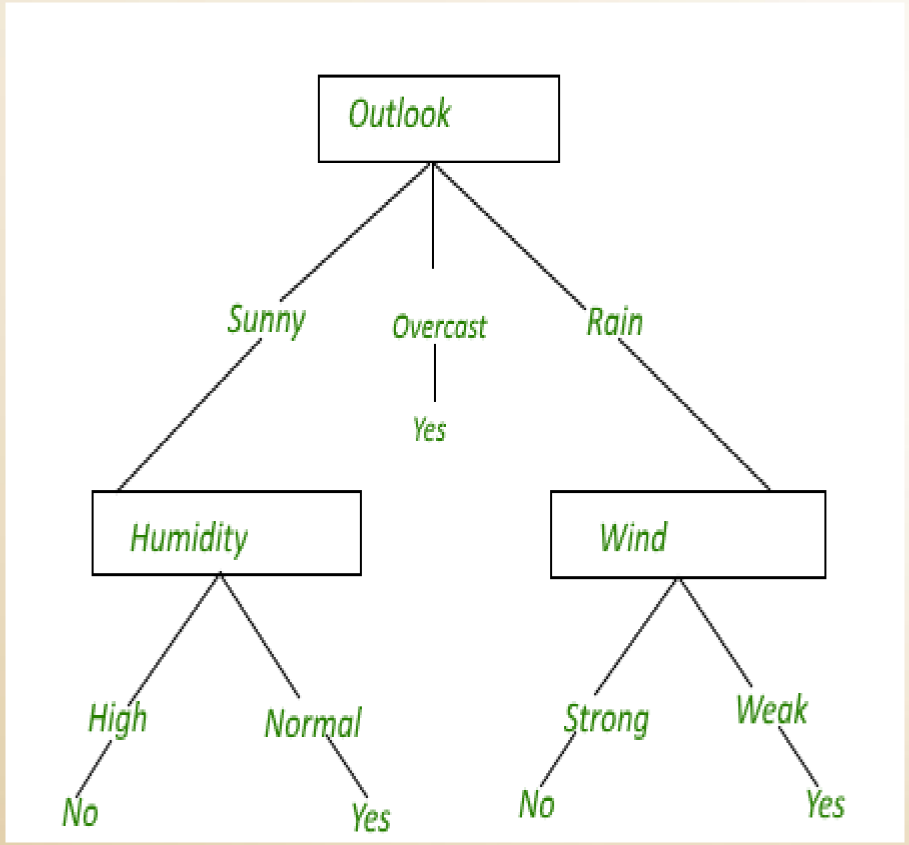
```
array([1, 1, 1, 1, 2], dtype=int64)
```

```
#Model Evaluation
j1=jaccard_score(y_test,k_y_pred)
f1=f1_score(y_test,k_y_pred, average = 'macro')
print("Jaccard Score: ",j1)
print("F1 Score: ",f1)
```

```
Jaccard Score:  0.45526428543606123
F1 Score:  0.597453791995599
```



Decision Tree

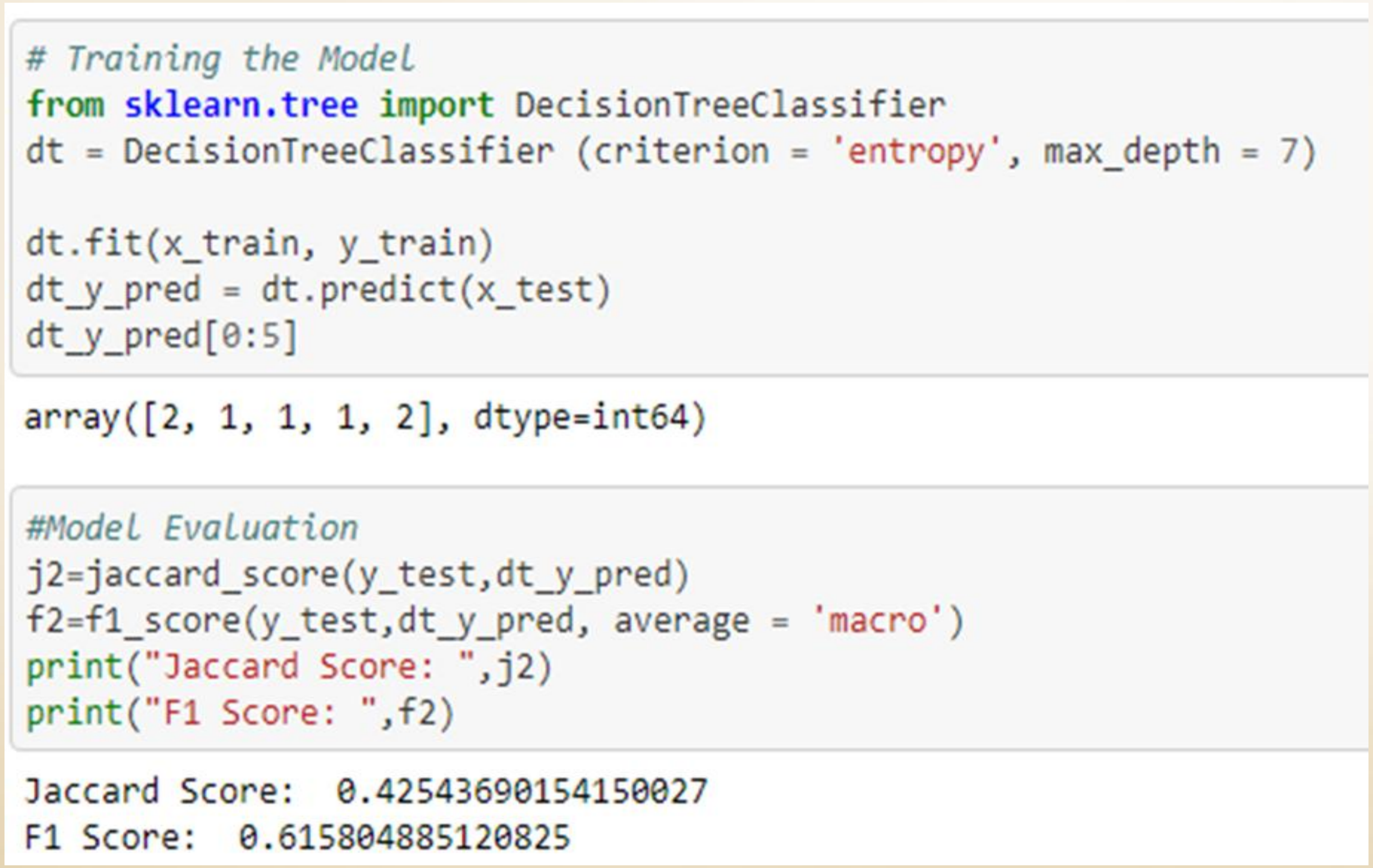


$$\mathrm{Entropy(PlayGolf)} = \mathrm{Entropy}\ (5,9)$$
$$= \mathrm{Entropy}\ (0.36,\ 0.64)$$
$$= -\ (0.36 \log_2 0.36) - (0.64 \log_2 0.64)$$
$$= 0.94$$

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247



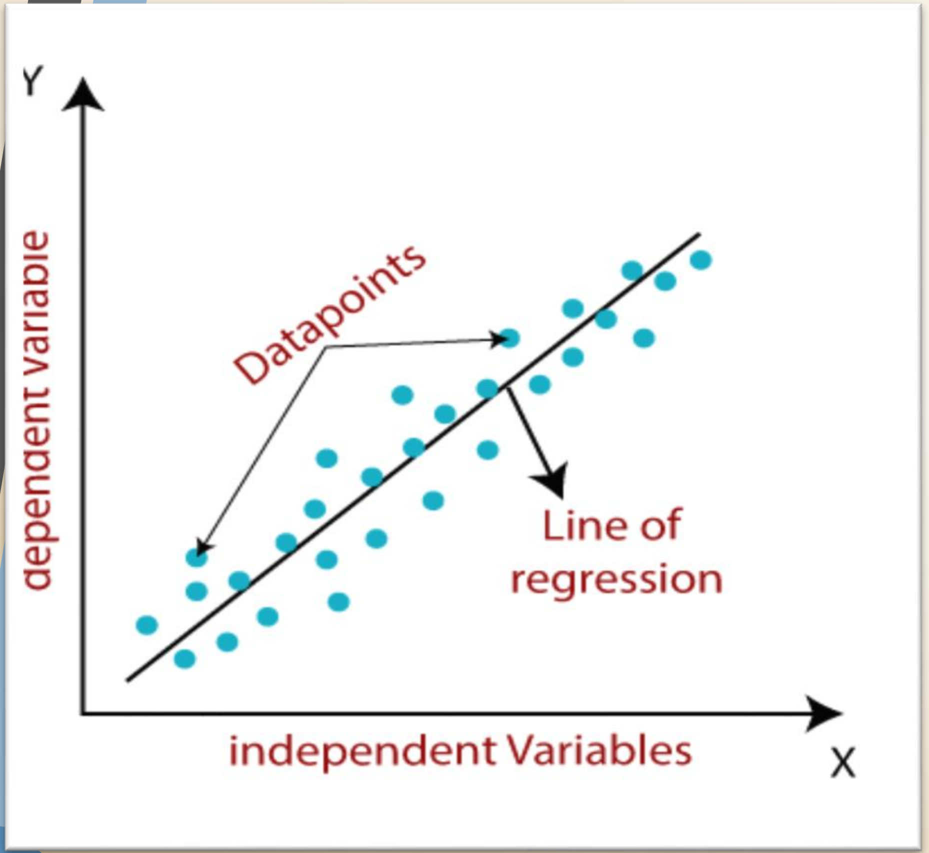
```
# Training the Model
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier (criterion = 'entropy', max_depth = 7)

dt.fit(x_train, y_train)
dt_y_pred = dt.predict(x_test)
dt_y_pred[0:5]
```

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

```
#Model Evaluation
j2=jaccard_score(y_test,dt_y_pred)
f2=f1_score(y_test,dt_y_pred, average = 'macro')
print("Jaccard Score: ",j2)
print("F1 Score: ",f2)
```

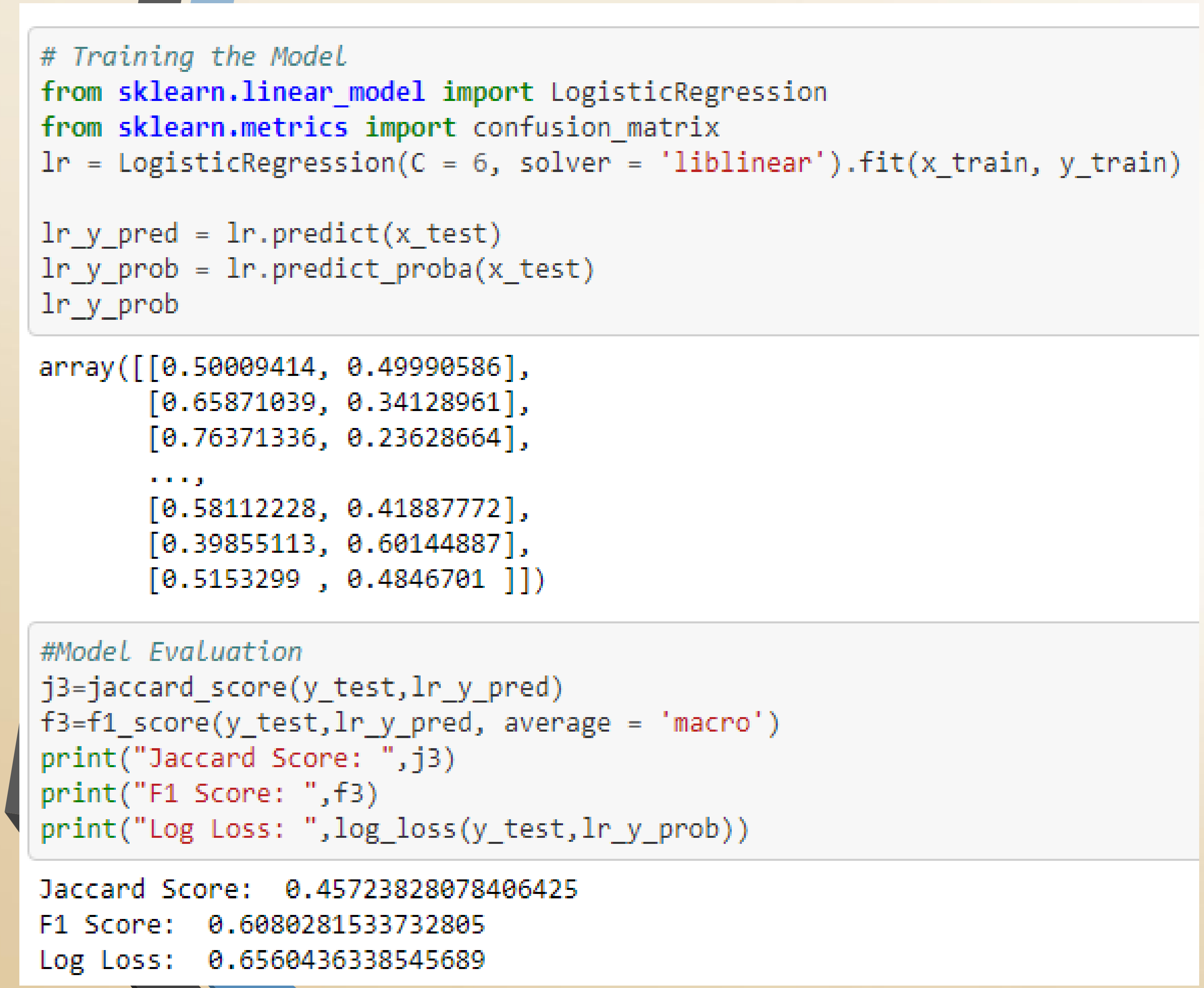
```
Jaccard Score:  0.42543690154150027
F1 Score:  0.615804885120825
```

Linear Regression



y= a0+a1x+ ε

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

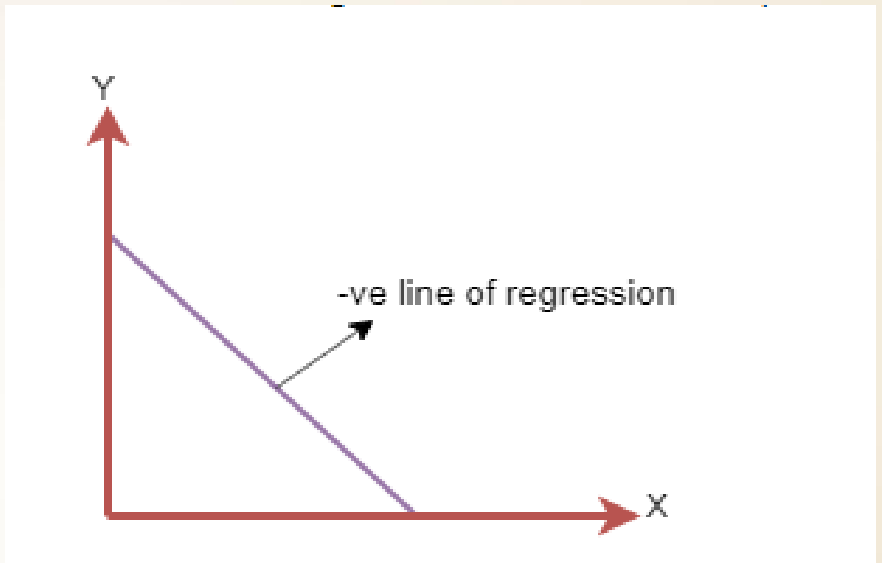


```
# Training the Model
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
lr = LogisticRegression(C = 6, solver = 'liblinear').fit(x_train, y_train)

lr_y_pred = lr.predict(x_test)
lr_y_prob = lr.predict_proba(x_test)
lr_y_prob
```

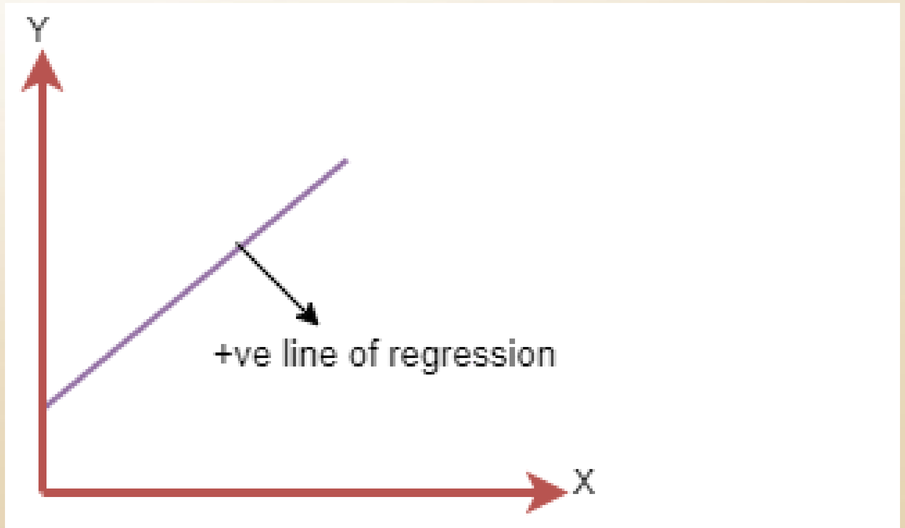
```
array([[0.50009414, 0.49990586],
       [0.65871039, 0.34128961],
       [0.76371336, 0.23628664],
       ...,
       [0.58112228, 0.41887772],
       [0.39855113, 0.60144887],
       [0.5153299 , 0.4846701 ]])
```

```
#Model Evaluation
j3=jaccard_score(y_test,lr_y_pred)
f3=f1_score(y_test,lr_y_pred, average = 'macro')
print("Jaccard Score: ",j3)
print("F1 Score: ",f3)
print("Log Loss: ",log_loss(y_test,lr_y_prob))
```

```
Jaccard Score:  0.4572382807840642 5
F1 Score:  0.6080281533732805
Log Loss:  0.6560436338545689
```



The line of equation will be: $\mathbf{Y= -a_0+a_1x}$



The line equation will be: $\mathbf{Y= a_0+a_1x}$

Algorithms And Their Camparison

Algorithms	Jaccard Score	F1 Score
KNN	0.455264	0.597454
Decision Tree	0.425437	0.615805
Linear Regression	0.457238	0.608028

Evaluation metrics used to test the accuracy of our models were jaccard index and f-1 score. Choosing different k, max depth and hyper parameter C values helped to improve the accuracy of our models.

From this exercise, we notice that we can train our model to determine the severity of an accident to a certain extent using the given dataset

Conclusion

- Based on evaluation of above models, it can be concluded that **Linear Regression** is the most ideal model for this case

- On conclude, this project aimed at exploring the data to provide insight in severity levels for road collisions at junctions. The predictive model would be useful to help local authorities decide on whether to implement new safety measures in certain areas