In [1]:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
```

In [2]:

```
#Load the data set
claimants = pd.read_csv("C:/Users/Ashraf/Documents/Py_files/claimants.csv")
claimants.head()
```

Out[2]:

	CASENUM	ATTORNEY	CLMSEX	CLMINSUR	SEATBELT	CLMAGE	LOSS
0	5	0	0.0	1.0	0.0	50.0	34.940
1	3	1	1.0	0.0	0.0	18.0	0.891
2	66	1	0.0	1.0	0.0	5.0	0.330
3	70	0	0.0	1.0	1.0	31.0	0.037
4	96	1	0.0	1.0	0.0	30.0	0.038

In [3]:

```
# dropping the case number columns as it is not required
claimants.drop(["CASENUM"],inplace=True,axis = 1)
```

In [4]:

```
#Shape of the data set
claimants.shape
```

Out[4]:

```
(1340, 6)
```

In [5]:

```
# Removing NA values in data set
claimants = claimants.dropna()
claimants.shape
```

Out[5]:

```
(1096, 6)
```

In [6]:

```
# Dividing our data into input and output variables
X = claimants.iloc[:,1:]
Y = claimants.iloc[:,0]
```

In [7]:

```
X
```

Out[7]:

	CLMSEX	CLMINSUR	SEATBELT	CLMAGE	LOSS
0	0.0	1.0	0.0	50.0	34.940
1	1.0	0.0	0.0	18.0	0.891
2	0.0	1.0	0.0	5.0	0.330
3	0.0	1.0	1.0	31.0	0.037
4	0.0	1.0	0.0	30.0	0.038
...	...	...	...	...	...
1334	1.0	1.0	0.0	16.0	0.060
1336	1.0	1.0	0.0	46.0	3.705
1337	1.0	1.0	0.0	39.0	0.099
1338	1.0	0.0	0.0	8.0	3.177
1339	1.0	1.0	0.0	30.0	0.688

1096 rows × 5 columns

In [8]:

```
Y
```

Out[8]:

```
0       0
1       1
2       1
3       0
4       1
       ..
1334    1
1336    0
1337    1
1338    0
1339    1
Name: ATTORNEY, Length: 1096, dtype: int64
```

In [9]:

```
#Logistic regression and fit the model
classifier = LogisticRegression()
classifier.fit(X,Y)
```

Out[9]:

```
LogisticRegression()
```

In [10]:

```
#Predict for X dataset
y_pred = classifier.predict(X)
```

In [11]:

```
y_pred_df= pd.DataFrame({'actual': Y,
                         'predicted_prob': classifier.predict(X)})
```

In [12]:

```
y_pred_df
```

Out[12]:

	actual	predicted_prob
0	0	0
1	1	1
2	1	1
3	0	0
4	1	1
...	...	...
1334	1	1
1336	0	0
1337	1	1
1338	0	0
1339	1	1

1096 rows × 2 columns

In [13]:

```
# Confusion Matrix for the model accuracy
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(Y,y_pred)
print (confusion_matrix)
```

```
[[381 197]
 [123 395]]
```

In [14]:

```
((381+395)/(381+197+123+395))*100
```

Out[14]:

```
70.8029197080292
```

In [15]:

```
#Classification report
from sklearn.metrics import classification_report
print(classification_report(Y,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.76      0.66      0.70       578
           1       0.67      0.76      0.71       518

    accuracy                           0.71      1096
   macro avg       0.71      0.71      0.71      1096
weighted avg       0.71      0.71      0.71      1096
```

In [16]:

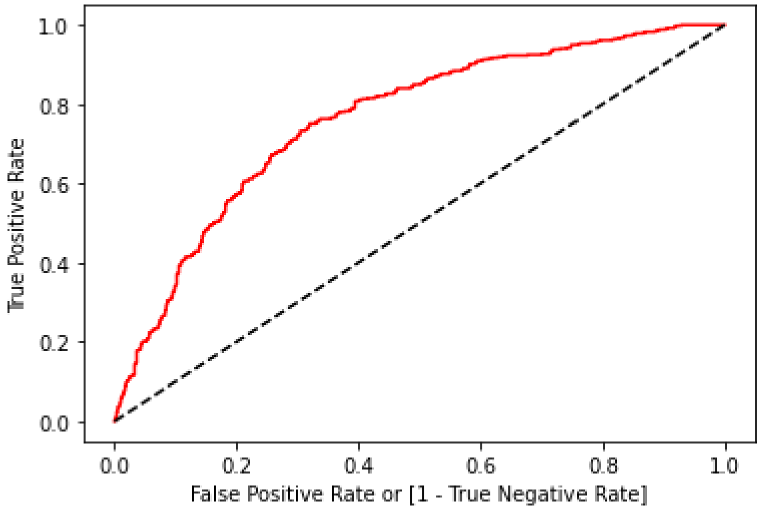
```
# ROC Curve
```

In [17]:

```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score

fpr, tpr, thresholds = roc_curve(Y, classifier.predict_proba (X)[:,1])

auc = roc_auc_score(Y, y_pred)

import matplotlib.pyplot as plt
plt.plot(fpr, tpr, color='red', label='logit model ( area  = %0.2f)'%auc)
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
plt.ylabel('True Positive Rate')
```

Out[17]:

```
Text(0, 0.5, 'True Positive Rate')
```



In [18]:

```
auc
```

Out[18]:

```
0.7108589063606365
```

In [ ]:

```

```