Theory :

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc... Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one.

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy as sp
```

In [2]:

```
dataset=pd.read_csv('~/Desktop/binary.csv')
```

In [10]:

```
dataset['admit']=dataset['admit'].astype('category')
dataset['rank']=dataset['rank'].astype('category')
```

In [3]:

```
X = dataset.iloc[:,1:4].values
Y = dataset.iloc[:,0].values
```

In [5]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=0)
```

In [6]:

```
from sklearn.linear_model import LogisticRegression
lregressor = LogisticRegression()
```

In [7]:

```
lregressor.fit(X_train,Y_train)
```

```
c:\users\pranjalpc\appdata\local\programs\python\python35\lib\site-package
s\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will
be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[7]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=Tru
e,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=
0,
                   warm_start=False)
```

In [8]:

```
pred = lregressor.predict(X_test)
```

In [9]:

```
pred
```

Out[9]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0], dtype=int64)
```

In [11]:

```
dataset.dtypes
```

Out[11]:

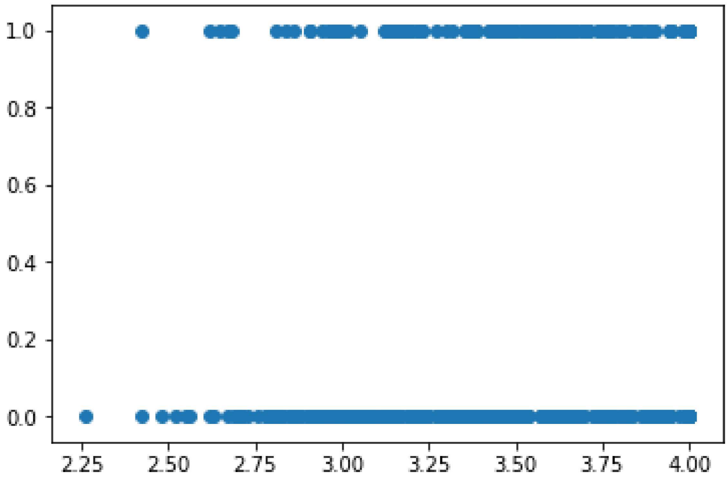
```
admit    category
gre         int64
gpa       float64
rank     category
dtype: object
```

In [12]:

```
plt.scatter(dataset.gpa,dataset.admit)
```

Out[12]:

```
<matplotlib.collections.PathCollection at 0x23d484855c0>
```

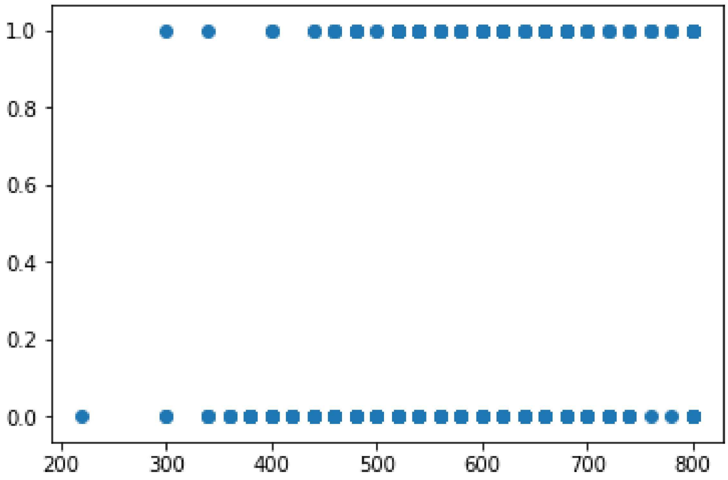


In [13]:

```
plt.scatter(dataset.gre,dataset.admit)
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x23d484e6da0>
```



In [14]:

```
from sklearn.metrics import confusion_matrix
```

In [15]:

```
conc = confusion_matrix(Y_test,pred,)
```

In [16]:

```
conc
```

Out[16]:

```
array([[50,  5],
       [20,  5]], dtype=int64)
```

In [17]:

```
from sklearn import metrics
```

In [18]:

```
roc = metrics.roc_curve(Y_test,pred)
```

In [19]:

```
roc
```

Out[19]:

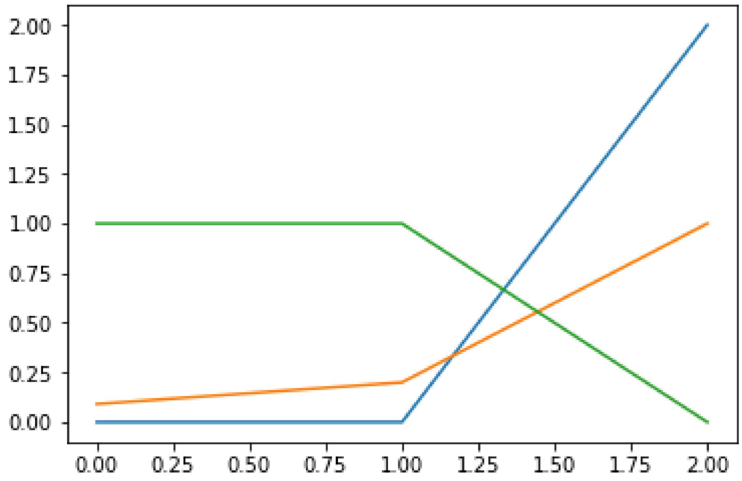
```
(array([0.        , 0.09090909, 1.        ]),
 array([0. , 0.2, 1. ]),
 array([2, 1, 0], dtype=int64))
```

In [20]:

```
plt.plot(roc)
```

Out[20]:

```
[<matplotlib.lines.Line2D at 0x23d484fbac8>,
 <matplotlib.lines.Line2D at 0x23d484fba20>,
 <matplotlib.lines.Line2D at 0x23d484fb3c8>]
```



Conclusion : In this experiment we learnt how to build a logistic regression model using machine learning libraries such as sklearn. Also we used the fit and predict functions to fit our data in the model and to predict the output results of our model.We also used the split data function to split our data into training and testing set. Also we used the matplotlib functions to plot the logistic regression graph. Also we plotted the roc curve.