

19CS10071

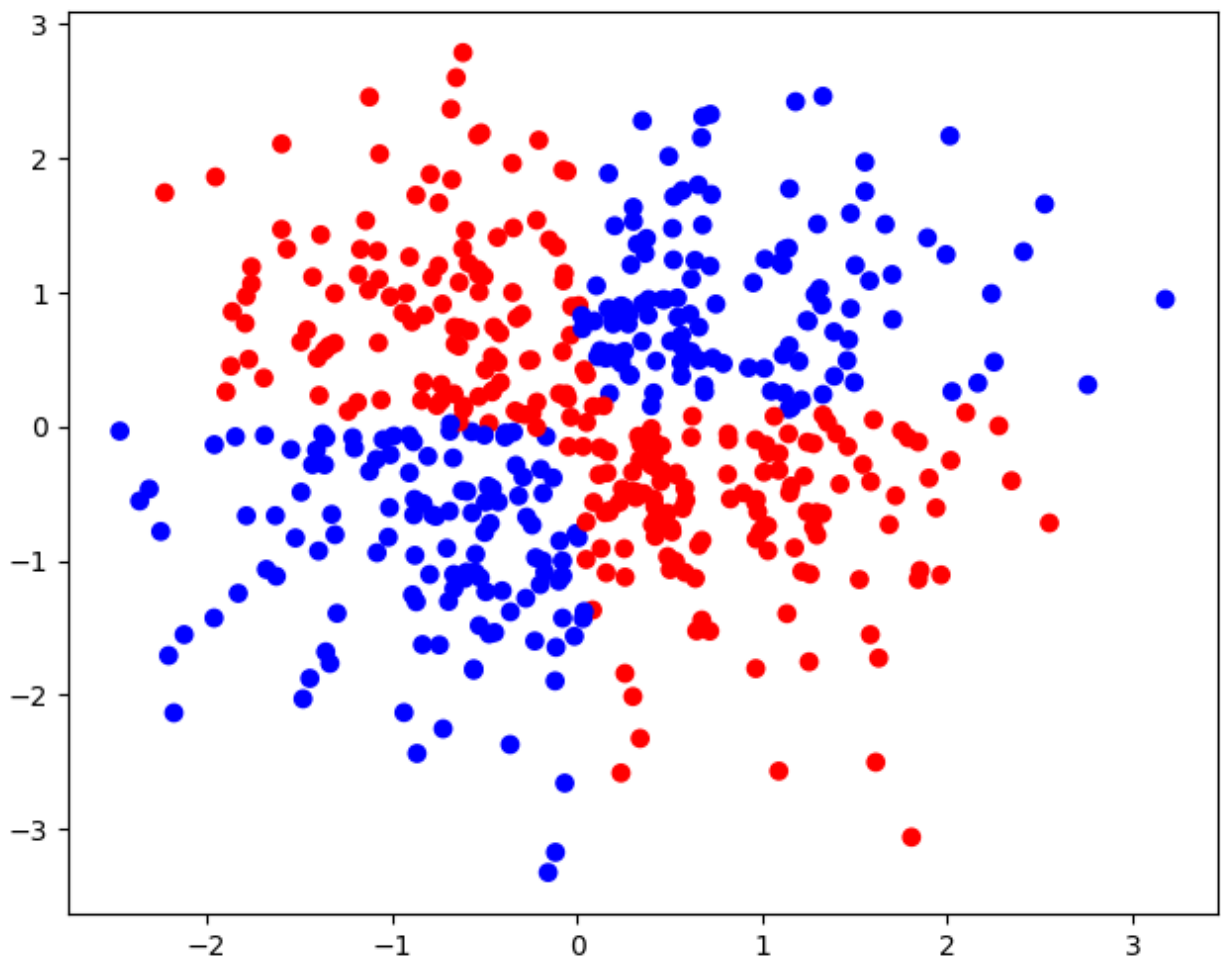
Q9)

Confusion Matrix

P Predicted	Ref	
	1	-1
1	243	4
-1	15	238

Accuracy: 96.2% , Error rate = 3.8%.

b)



Red: 1 - Blue: -1

c)

Q = [-0.01323906 -0.04324553 -0.01434454 0.598131 -0.01314315 0.02477809]

for $g(x) = x_1 x_2$ it will perfectly
fit $\because y = 1 \Leftrightarrow g(x) \geq 0 \Leftrightarrow x_1 x_2 \geq 0$
& $y = -1 \Leftrightarrow g(x) < 0 \Leftrightarrow x_1 x_2 < 0$

So condition holds in both cases

In our L.S computed theta also
we see that x_1, x_2 has the highest
coefficient ≈ 0.598131

Python Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
from pandas.core import api
X = np.random.randn(500,2)
A = np.array([[1,x[0],x[1],x[0]*x[1],x[0]*x[0],x[1]*x[1]] for x in X])
print(A.shape)
Y = np.array([1 if x[0]*x[1]>=0 else -1 for x in X])
theta = np.linalg.lstsq(A,Y)[0]
confusion_matrix = {}
confusion_matrix[1] = {1:0,-1:0}
confusion_matrix[-1] = {1:0,-1:0}

def predict(x):
    if(np.dot(theta,np.array([1,x[0],x[1],x[0]*x[1],x[0]*x[0],x[1]*x[1]]))>=0):
        return 1
    else:
        return -1
print(X.shape,Y.shape)
err = 0
label = []
colors = ['blue','red']
for i in range(500):
    x = X[i]
    y = Y[i]
    y_pred = predict(x)
    if(y_pred==1):
        label.append(0)
    else:
        label.append(1)
    if(y!=y_pred) :
        err+=1

    confusion_matrix[y][y_pred]+=1
confusion_matrix = pd.DataFrame(confusion_matrix)
print(confusion_matrix)
print('Accuracy :',(1.0-err/500))
print(theta)
X_ = X.T
plt.scatter(X_[0],X_[1],c=label,cmap=matplotlib.colors.ListedColormap(colors))
plt.show()
```