**Lab Assignment No. 4**

**Code:**

from sklearn.datasets import make\_classification  
import numpy as np  
import matplotlib.pyplot as plt

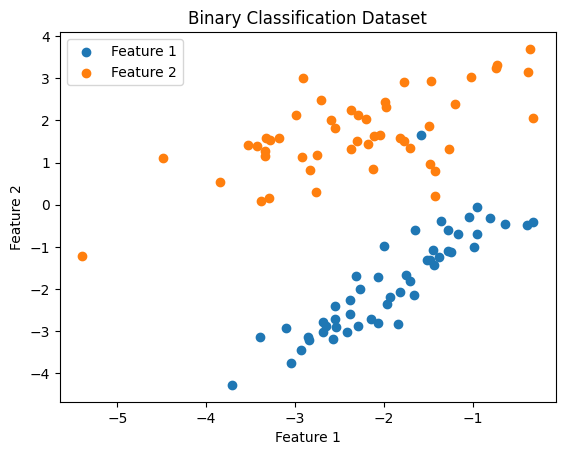
class Perceptron:  
 def \_\_init\_\_(self, input\_size, learning\_rate=0.01):  
 self.weights = np.zeros(input\_size)  
 self.bias = 0  
 self.learning\_rate = learning\_rate  
  
 def activation(self, weighted\_sum):  
 return np.where(weighted\_sum <= 0, 0, 1)  
   
 def predict(self, inputs):  
 weighted\_sum = np.dot(inputs, self.weights) + self.bias  
 return self.activation(weighted\_sum)  
  
 def train(self, inputs, labels, num\_epochs):  
 for \_ in range(num\_epochs):  
 for x, y in zip(inputs, labels):  
 predicted = self.predict(x)  
 error = y - predicted  
 self.weights += self.learning\_rate \* error \* x  
 self.bias += self.learning\_rate \* error

X, y = make\_classification(n\_samples=100, n\_features=2, n\_informative=2,  
 n\_redundant=0, n\_clusters\_per\_class=1,  
 class\_sep=2)  
inputs = X  
labels = y

labels

array([0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0,  
 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0,  
 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1,  
 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0,  
 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0])

plt.scatter(X[labels == 0][:,0], X[labels == 0][:, 1], label="Feature 1")  
plt.scatter(X[labels == 1][:,0], X[labels == 1][:, 1], label="Feature 2")  
plt.xlabel('Feature 1')  
plt.ylabel('Feature 2')  
plt.title('Binary Classification Dataset')  
plt.legend()  
plt.show()



perceptron = Perceptron(input\_size=2)  
perceptron.train(inputs, labels, num\_epochs=10)  
x\_min, x\_max = inputs[:, 0].min() - 1, inputs[:, 0].max() + 1  
y\_min, y\_max = inputs[:, 1].min() - 1, inputs[:, 1].max() + 1  
xx, yy = np.meshgrid(np.arange(x\_min, x\_max, 0.1),  
 np.arange(y\_min, y\_max, 0.1))  
Z = perceptron.predict(np.c\_[xx.ravel(), yy.ravel()])  
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, alpha=0.3)  
plt.scatter(inputs[labels == 0][:, 0], inputs[labels == 0][:, 1], color='blue', label='Feature 1')  
plt.scatter(inputs[labels == 1][:, 0], inputs[labels == 1][:, 1], color='red', label='Featuer 2')  
plt.xlabel('Feature 1')  
plt.ylabel('Feature 2')  
plt.title('Perceptron Decision Regions')  
plt.legend()  
plt.show()

**Output:**

