

Experiment -18

IRIS Dataset (Sepal length, Sepal width, Petal length, Petal width, Class)

```
data = [  
    [5.1, 3.5, 1.4, 0.2, 1], # Setosa  
    [4.9, 3.0, 1.4, 0.2, 1],  
    [6.2, 3.4, 5.4, 2.3, 0], # Non-Setosa  
    [5.9, 3.0, 5.1, 1.8, 0],  
    [5.5, 2.3, 4.0, 1.3, 0],  
    [6.5, 2.8, 4.6, 1.5, 0]  
]
```

Initialize weights and bias

w = [0, 0, 0, 0]

b = 0

lr = 0.1

Training

for _ in range(20):

for row in data:

x = row[:-1]

y = row[-1]

pred = 1 if sum(w[i]*x[i] for i in range(4)) + b >= 0 else 0

error = y - pred

for i in range(4):

w[i] += lr * error * x[i]

b += lr * error

Prediction function

def predict(x):

return 1 if sum(w[i]*x[i] for i in range(4)) + b >= 0 else 0

Test sample

test = [5.0, 3.4, 1.5, 0.2]

result = predict(test)

print("Test Flower:", test)

print("Prediction:", "Setosa" if result == 1 else "Non-Setosa")

Output:

main.py

Share

Run

```
1 data = [  
2     [5.1, 3.5, 1.4, 0.2, 1],  
3     [4.9, 3.0, 1.4, 0.2, 1],  
4     [6.2, 3.4, 5.4, 2.3, 0],  
5     [5.9, 3.0, 5.1, 1.8, 0],  
6     [5.5, 2.3, 4.0, 1.3, 0],  
7     [6.5, 2.8, 4.6, 1.5, 0]  
8 ]  
9 w = [0, 0, 0, 0]  
10 b = 0  
11 lr = 0.1  
12 for _ in range(20):  
13     for row in data:  
14         x = row[:-1]  
15         y = row[-1]  
16         pred = 1 if sum(w[i]*x[i] for i in range(4)) + b >= 0 else 0  
17         error = y - pred  
18         for i in range(4):  
19             w[i] += lr * error * x[i]  
20         b += lr * error  
21 def predict(x):  
22     return 1 if sum(w[i]*x[i] for i in range(4)) + b >= 0 else 0  
23 test = [5.0, 3.4, 1.5, 0.2]  
24 result = predict(test)  
25 print("Test Flower:", test)  
26 print("Prediction:", "Setosa" if result == 1 else "Non-Setosa")  
27
```

Output

Clear

Test Flower: [5.0, 3.4, 1.5, 0.2]
Prediction: Setosa

=== Code Execution Successful ===

Activate Windows

Go to Settings to activate Windows.