

Experiment -13

```
# Dataset: [Car Age (years), Mileage (km)], Price
```

```
X = [
```

```
    [1, 20000],
```

```
    [2, 30000],
```

```
    [3, 40000],
```

```
    [4, 50000],
```

```
    [5, 60000]
```

```
]
```

```
Y = [20000, 18000, 16000, 14000, 12000]
```

```
# Initialize weights
```

```
w1, w2, b = 0.1, 0.1, 0
```

```
lr = 0.00000001
```

```
# Training (Gradient Descent)
```

```
for _ in range(10000):
```

```
    for x, y in zip(X, Y):
```

```
        pred = w1*x[0] + w2*x[1] + b
```

```
        error = y - pred
```

```
        w1 += lr * error * x[0]
```

```
        w2 += lr * error * x[1]
```

```
        b += lr * error
```

```
# Prediction
```

```
def predict(age, mileage):
```

```
    return w1*age + w2*mileage + b
```

```
# Test
```

```
age = 3
```

```
mileage = 45000
```

```
price = predict(age, mileage)
```

```
print("Car Age:", age)
```

```
print("Mileage:", mileage)
```

```
print("Predicted Price:", round(price, 2))
```

Output:

main.py

Share

Run

```
1 X = [
2   [1, 20000],
3   [2, 30000],
4   [3, 40000],
5   [4, 50000],
6   [5, 60000]
7 ]
8 Y = [20000, 18000, 16000, 14000, 12000]
9 w1, w2, b = 0.1, 0.1, 0
10 lr = 0.00000001
11 for _ in range(10000):
12     for x, y in zip(X, Y):
13         pred = w1*x[0] + w2*x[1] + b
14         error = y - pred
15         w1 += lr * error * x[0]
16         w2 += lr * error * x[1]
17         b += lr * error
18 def predict(age, mileage):
19     return w1*age + w2*mileage + b
20 age = 3
21 mileage = 45000
22 price = predict(age, mileage)
23 print("Car Age:", age)
24 print("Mileage:", mileage)
25 print("Predicted Price:", round(price, 2))
```

Output

Clear

Car Age: 3
Mileage: 45000
Predicted Price: nan

=== Code Execution Successful ===

Activate Windows
Go to Settings to activate Windows.