

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:

The screenshot shows a Jupyter Notebook interface with the following details:

- Code Cell:** The code cell contains a Python script named `main.py` which defines a list of X values (ages) and Y values (mileage), initializes parameters `w1`, `w2`, and `b`, and implements a gradient descent loop to find the best fit line. It also includes a `predict` function and prints the results for a car with age 3 and mileage 45000.
- Run Button:** A blue "Run" button is located above the output area.
- Output Area:** The output shows the execution results:
 - Car Age: 3
 - Mileage: 45000
 - Predicted Price: nan

==== Code Execution Successful ===
- Activation Message:** A message at the bottom right says "Activate Windows Go to Settings to activate Windows."