```
import pandas as pd
```

```
df=pd.read_csv('/content/Salary_dataset.csv')
```

Add text cell

```
df.head()
```

	Unnamed: 0	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0

```
df.tail()
```

	Unnamed: 0	YearsExperience	Salary
25	25	9.1	105583.0
26	26	9.6	116970.0
27	27	9.7	112636.0
28	28	10.4	122392.0
29	29	10.6	121873.0

```
x=df['YearsExperience'].values
```

```
y=df['Salary'].values
```

```
print(x)
```

```
[ 1.2  1.4  1.6  2.1  2.3  3.   3.1  3.3  3.3  3.8  4.   4.1  4.1  4.2
  4.6  5.   5.2  5.4  6.   6.1  6.9  7.2  8.   8.3  8.8  9.1  9.6  9.7
 10.4 10.6]
```

```
x = df.iloc[:, -1].values
x
```

```
array([ 39344.,  46206.,  37732.,  43526.,  39892.,  56643.,  60151.,
        54446.,  64446.,  57190.,  63219.,  55795.,  56958.,  57082.,
        61112.,  67939.,  66030.,  83089.,  81364.,  93941.,  91739.,
        98274., 101303., 113813., 109432., 105583., 116970., 112636.,
       122392., 121873.])
```

```
y = df.iloc[:, -1].values
y
```

```
array([ 39344.,  46206.,  37732.,  43526.,  39892.,  56643.,  60151.,
        54446.,  64446.,  57190.,  63219.,  55795.,  56958.,  57082.,
        61112.,  67939.,  66030.,  83089.,  81364.,  93941.,  91739.,
        98274., 101303., 113813., 109432., 105583., 116970., 112636.,
       122392., 121873.])
```

```
import numpy as np
```

```
X = np.array(x)
Y = np.array(y)
```

```
X = X.reshape(-1, 1)
X
```

```
array([[ 39344.],
       [ 46206.],
       [ 37732.],
       [ 43526.],
```

```
       [ 39892.],
       [ 56643.],
       [ 60151.],
       [ 54446.],
       [ 64446.],
       [ 57190.],
       [ 63219.
```

Add text cell

```
       [ 55795.],
       [ 56958.],
       [ 57082.],
       [ 61112.],
       [ 67939.],
       [ 66030.],
       [ 83089.],
       [ 81364.],
       [ 93941.],
       [ 91739.],
       [ 98274.],
       [101303.],
       [113813.],
       [109432.],
       [105583.],
       [116970.],
       [112636.],
       [122392.],
       [121873.]])
```

```
X_min = X.min()
X_max = X.max()

X_norm = (X - X_min) / (X_max - X_min)
X_norm
```

```
array([[0.01904087],
       [0.1000945 ],
       [0.        ],
       [0.06843846],
       [0.02551382],
       [0.22337586],
       [0.26481219],
       [0.19742499],
       [0.31554453],
       [0.229837  ],
       [0.30105126],
       [0.21335932],
       [0.22709662],
       [0.2285613 ],
       [0.27616348],
       [0.35680369],
       [0.33425467],
       [0.53575478],
       [0.51537916],
       [0.66393811],
       [0.63792818],
       [0.7151193 ],
       [0.75089771],
       [0.89866525],
       [0.84691708],
       [0.80145287],
       [0.93595559],
       [0.88476258],
       [1.        ],
       [0.9938696 ]])
```

```
m = np.random.randn()
c = np.random.randn()

print("Randomly initialised slope:",m)
print("Randomly initialised intercept:",c)
```

```
Randomly initialised slope: -0.01975248985339005
Randomly initialised intercept: 0.014376397674075655
```

```
def predict(X, m, c):
  y_pred = m * X + c
  return y_pred
```

```
y_pred = predict(X, m, c)
y_pred
```

```
array([[ -777.12758439],
       [ -912.66916977],
       [ -745.28657075],
       [ -859.73249696],
       [ -787.95194883],
       [-1118.82590637],
       [-1188.1 ],
       [-1075.42968616],
       [-1272.95458469],
       [-1129.63051832],
       [-1248.71827964],
       [-1102.07579497],
       [-1125.04794067],
       [-1127.49724941],
       [-1207.09978352],
       [-1341.95003175],
       [-1304.24252862],
       [-1641.20025303],
       [-1607.12720803],
       [-1855.55427292],
       [-1812.05929026],
       [-1941.14181145],
       [-2000.97210322],
       [-2248.07575129],
       [-2161.54009324],
       [-2085.51275979],
       [-2310.43436175],
       [-2224.82707073],
       [-2417.53236174],
       [-2407.2808195 ]])
```

Add text cell

```
print("First 5 predicted values:")
print(y_pred[:5])
```

```
First 5 predicted values:
[[-777.12758439]
 [-912.66916977]
 [-745.28657075]
 [-859.73249696]
 [-787.95194883]]
```

```
def compute_cost(y, y_pred):
  n = len(y)
  cost = (1 / (2 * n)) * np.sum((y_pred - y) ** 2)
  return cost
```

```
cost = compute_cost(Y, y_pred)
print("Initial salary:",cost)
```

```
Initial salary: 3381273714.0882936
```

```
Start coding or generate with AI.
```