```
import pandas as pd
df=pd.read_csv('/content/Salary_dataset.csv')
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
y=df.Salary
x
y
```

	Salary
0	39344.0
1	46206.0
2	37732.0
6	60151.0
7	54446.0
10	63219.0
11	55795.0
12	56958.0
14	61112.0
16	66030.0
17	83089.0
18	81364.0
21	98274.0
22	101303.0
23	113813.0
24	109432.0
28	122392.0
29	121873.0

dtype: float64

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

```
array([ 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])
```

```
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.])
```

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

```
x
```

```
array([ 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])
```

```
y=df.iloc[:,2].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
```

```
array([[ 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]])
```

```
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.])
```

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

```
X
```

```
array([[ 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.min()
```

```
np.float64(1.2000000000000002)
```

```
X.max()
```

```
np.float64(10.6)
```

```
Y.min()
```

```
np.float64(37732.0)
```

```
Y.max()
```

```
np.float64(122392.0)
```

```
X_mean=X.mean()
X_std=X.std()
X_standardized = (X - X_mean) / X_std
X_standardized
```

```
array([[-1.51005294, -1.43837321, -1.36669348],
       [-1.18749416, -1.11581443, -0.86493538],
       [-0.82909552, -0.75741579, -0.75741579],
       [-0.57821647, -0.50653674, -0.47069688],
       [-0.47069688, -0.43485702, -0.29149756],
       [-0.1481381 , -0.07645838, -0.00477865],
       [ 0.21026054,  0.2461004 ,  0.53281931],
       [ 0.6403389 ,  0.92705781,  1.03457741],
       [ 1.21377673,  1.32129632,  1.50049564],
       [ 1.5363355 ,  1.78721455,  1.85889428]])
```

```
X_min=X.min()
X_max=X.max()
X_normalized=(X-X_min)/(X_max-X_min)
X_normalized
```

```
array([[0.        ],
       [0.0212766 ],
       [0.04255319],
       [0.09574468],
       [0.11702128],
       [0.19148936],
       [0.20212766],
       [0.22340426],
       [0.22340426],
       [0.27659574],
       [0.29787234],
       [0.30851064],
       [0.30851064],
       [0.31914894],
       [0.36170213],
       [0.40425532],
       [0.42553191],
       [0.44680851],
       [0.5106383 ],
       [0.5212766 ],
       [0.60638298],
       [0.63829787],
       [0.72340426],
       [0.75531915],
       [0.80851064],
       [0.84042553],
       [0.89361702],
       [0.90425532],
```

```
        [0.9787234 ],
        [1.        ]])
```

```
m=np.random.randn()
c=np.random.randn()
print("randomly intialized slope:",m)
print("randomly intialized intercept:",c)
```

```
randomly intialized slope: -0.22922079203229448
randomly intialized intercept: 0.5124675187370522
```

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

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

```
print("First 5 predictions:")
print(y_pred[:5])
```

```
First 5 predictions:
[[0.51246752 0.50759048 0.50271344]
 [0.49052085 0.48564381 0.46857418]
 [0.46613566 0.46125862 0.46125862]
 [0.44906602 0.44418898 0.44175047]
 [0.44175047 0.43931195 0.42955787]]
```

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

```
cost = compute_cost(Y,y_pred)
print("initial cost (MSE):",cost)
```

```
initial cost (MSE): 97545674947.37328
```

```
learning_rate = 0.01
epochs=1000
print("l_r",learning_rate)
print("epochs",epochs)
```

```
l_r 0.01
epochs 1000
```

```
n=len(y)
```

```
for epoch in range(epochs):
  y_pred=predict(X_normalized,m,c)
  dm=(1/n)*np.sum((y_pred-Y)*X_normalized)
  dc=(1/n)*np.sum((y_pred-Y))
  m=m-learning_rate*dm
  c=c-learning_rate*dc
  cost=compute_cost(Y,y_pred)
  if epoch%100==0:
    print("epoch",epoch,"cost",cost)
```

```
epoch 0 cost 10897488926.0
epoch 100 cost 10897488926.0
epoch 200 cost 10897488926.0
epoch 300 cost 10897488926.0
epoch 400 cost 10897488926.0
epoch 500 cost 10897488926.0
epoch 600 cost 10897488926.0
epoch 700 cost 10897488926.0
epoch 800 cost 10897488926.0
epoch 900 cost 10897488926.0
```

```
final_slope=m
final_intercept=c
print("final slope:",final_slope)
print("final intercept:",final_intercept)
```

```
final slope: 1.1163278310114965e-10
final intercept: 76003.99999999993
```

```
def predict_salary(years_experience, m, c,X_min,X_max):
  x=np.array(years_experience).reshape(-1,1)
  X_normalized=(x-X_min)/(X_max-X_min)
  salary_pred=m*X_normalized+c
  return salary_pred
```

```
years = 5
predicted_salary = predict_salary(
    years,
    final_slope,
    final_intercept,
    X_min,
    X_max
)
print(f"predicted salary for{years}years of experience:")
print(predicted_salary)
```

```
predicted salary for5years of experience:
[[76004.]]
```

```
y_pred_final = predict(X_normalized,final_slope,final_intercept)
mse=np.mean((y-y_pred)**2)
print(mse)
```

```
726499261.7333333
```