Simple Linear Regression_Salary_Experience

June 5, 2026

1 SIMPLE LINEAR REGRESSION WITH

1.1 prediction, x coefficient and slope values

[19]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[20]:
```
df = pd.read_csv("Salary_data.csv")
df
```

[20]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891
5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.2	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111
15	4.9	67938
16	5.1	66029
17	5.3	83088
18	5.9	81363
19	6.0	93940
20	6.8	91738
21	7.1	98273
22	7.9	101302
23	8.2	113812
24	8.7	109431
25	9.0	105582

```
26              9.5  116969
27              9.6  112635
28             10.3  122391
29             10.5  121872
```

[21]:
```
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
print(X)
print(y)
```

```
[[ 1.1]
 [ 1.3]
 [ 1.5]
 [ 2. ]
 [ 2.2]
 [ 2.9]
 [ 3. ]
 [ 3.2]
 [ 3.2]
 [ 3.7]
 [ 3.9]
 [ 4. ]
 [ 4. ]
 [ 4.1]
 [ 4.5]
 [ 4.9]
 [ 5.1]
 [ 5.3]
 [ 5.9]
 [ 6. ]
 [ 6.8]
 [ 7.1]
 [ 7.9]
 [ 8.2]
 [ 8.7]
 [ 9. ]
 [ 9.5]
 [ 9.6]
 [10.3]
 [10.5]]
[ 39343  46205  37731  43525  39891  56642  60150  54445  64445  57189
  63218  55794  56957  57081  61111  67938  66029  83088  81363  93940
  91738  98273 101302 113812 109431 105582 116969 112635 122391 121872]
```

[22]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3,␣
 ↪random_state = 0)
```

[23]:
```
print(f"Total rows in Dataframe are : {len(df)}")
print(f"Number of rows selected for X_train : {len(X_train)}")
print(f"\t Number of rows selected for X_test : {len(X_test)}")
print(f"Number of rows selected for y_train : {len(y_train)}")
print(f"\t Number of rows selected for y_test : {len(y_test)}")
```

```
Total rows in Dataframe are : 30
Number of rows selected for X_train : 20
         Number of rows selected for X_test : 10
Number of rows selected for y_train : 20
         Number of rows selected for y_test : 10
```

[24]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

[24]:
```
LinearRegression()
```

[25]:
```
y_pred = regressor.predict(X_test)
```

[26]:
```
plt.scatter(X_train, y_train, color='red')
plt.plot(X_train, regressor.predict(X_train), color = 'green')
plt.title('Salary vs Experience (Training set)')
plt.xlabel ("Years of Experiencec")
plt.ylabel ("Salary on Exeriencec")
plt.show()
```

[27]:
```
plt.scatter(X_test, y_test, color='blue')
plt.scatter(X_train, y_train, color='red')
plt.title('Salary vs Experience (red = Train, blue = predicted)')
plt.xlabel ("Years of Experiencec")
plt.ylabel ("Salary on Exeriencec")
plt.plot(X_train, regressor.predict(X_train), color = 'green')
```

[27]:
```
[<matplotlib.lines.Line2D at 0x243ff8348d0>]
```

1.2 Predicting probable salary with 12 years of experience

[28]:
```
print(regressor.predict([[12]]))
```

```
[138967.5015615]
```

[29]:
```
slope = f"{regressor.coef_[0]: .3f}"
yint = f"{regressor.intercept_: .3f}"

print(f"The slope of line is {slope} and its y-intercept is at {yint}")
```

```
The slope of line is  9345.942 and its y-intercept is at  26816.192
```

[31]:
```
# y = mx + c where c is intercept_, m is coef_, y = predicted value
# Salary = 9345.94 x experience + 26816.1922
#          YearsExperience        Salary
# 1st row in the table is:         1.1yrs,        39,343SAR
# 9345.94 * 1.1 + 26816.1922
```

2 how Y = MX + C is applied in practice

[32]:
```
df['pred sal'] = float(slope) * df['YearsExperience'] + float(yint)
```

[33]:
```
df
```

[33]:

	YearsExperience	Salary	pred sal
0	1.1	39343	37096.7282
1	1.3	46205	38965.9166
2	1.5	37731	40835.1050
3	2.0	43525	45508.0760
4	2.2	39891	47377.2644
5	2.9	56642	53919.4238
6	3.0	60150	54854.0180
7	3.2	54445	56723.2064
8	3.2	64445	56723.2064
9	3.7	57189	61396.1774
10	3.9	63218	63265.3658
11	4.0	55794	64199.9600
12	4.0	56957	64199.9600
13	4.1	57081	65134.5542
14	4.5	61111	68872.9310
15	4.9	67938	72611.3078
16	5.1	66029	74480.4962
17	5.3	83088	76349.6846
18	5.9	81363	81957.2498
19	6.0	93940	82891.8440
20	6.8	91738	90368.5976
21	7.1	98273	93172.3802
22	7.9	101302	100649.1338
23	8.2	113812	103452.9164
24	8.7	109431	108125.8874
25	9.0	105582	110929.6700
26	9.5	116969	115602.6410
27	9.6	112635	116537.2352
28	10.3	122391	123079.3946
29	10.5	121872	124948.5830

[34]:
```
df['pred sal'] = df['pred sal'].astype(float)
df['pred - sal'] = df['pred sal'] - df['Salary']
df
```

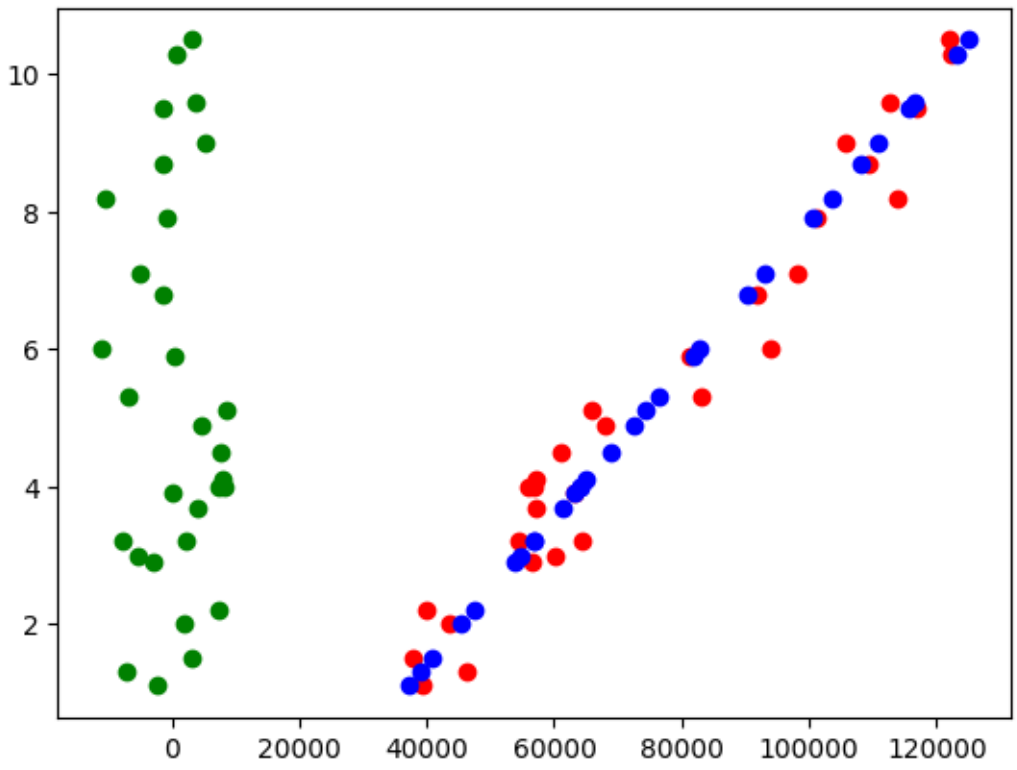
[34]:

	YearsExperience	Salary	pred sal	pred - sal
0	1.1	39343	37096.7282	-2246.2718
1	1.3	46205	38965.9166	-7239.0834
2	1.5	37731	40835.1050	3104.1050
3	2.0	43525	45508.0760	1983.0760

4	2.2	39891	47377.2644	7486.2644
5	2.9	56642	53919.4238	-2722.5762
6	3.0	60150	54854.0180	-5295.9820
7	3.2	54445	56723.2064	2278.2064
8	3.2	64445	56723.2064	-7721.7936
9	3.7	57189	61396.1774	4207.1774
10	3.9	63218	63265.3658	47.3658
11	4.0	55794	64199.9600	8405.9600
12	4.0	56957	64199.9600	7242.9600
13	4.1	57081	65134.5542	8053.5542
14	4.5	61111	68872.9310	7761.9310
15	4.9	67938	72611.3078	4673.3078
16	5.1	66029	74480.4962	8451.4962
17	5.3	83088	76349.6846	-6738.3154
18	5.9	81363	81957.2498	594.2498
19	6.0	93940	82891.8440	-11048.1560
20	6.8	91738	90368.5976	-1369.4024
21	7.1	98273	93172.3802	-5100.6198
22	7.9	101302	100649.1338	-652.8662
23	8.2	113812	103452.9164	-10359.0836
24	8.7	109431	108125.8874	-1305.1126
25	9.0	105582	110929.6700	5347.6700
26	9.5	116969	115602.6410	-1366.3590
27	9.6	112635	116537.2352	3902.2352
28	10.3	122391	123079.3946	688.3946
29	10.5	121872	124948.5830	3076.5830

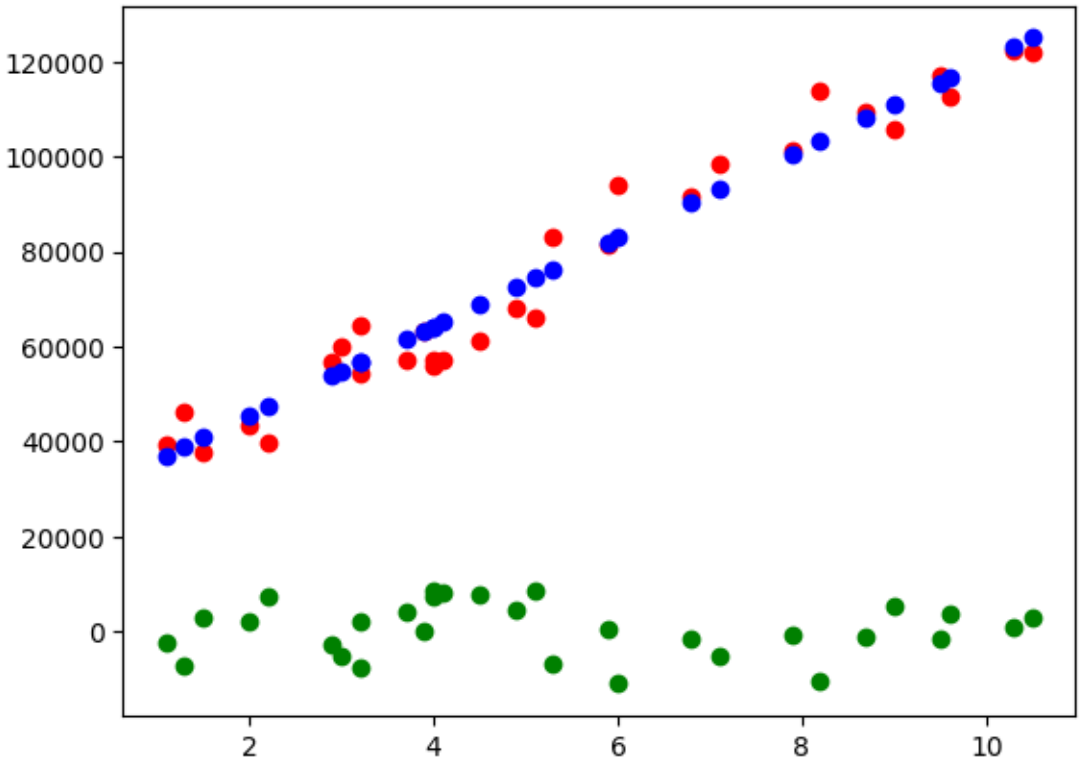
[35]:
```
plt.scatter(df['Salary'], df['YearsExperience'], color='red')
plt.scatter(df['pred sal'], df['YearsExperience'], color = 'blue')
plt.scatter(df['pred - sal'], df['YearsExperience'], color = 'green')
```

[35]: <matplotlib.collections.PathCollection at 0x243ff7c7950>



[36]:
```
plt.scatter(df['YearsExperience'], df['Salary'],  color='red')
plt.scatter(df['YearsExperience'], df['pred sal'],  color = 'blue')
plt.scatter(df['YearsExperience'], df['pred - sal'],  color = 'green')
```

[36]: <matplotlib.collections.PathCollection at 0x243ff6eb5d0>


120000
100000
80000
60000
40000
20000
0
2
4
6
8
10