Simple Linear Regression

July 15, 2023

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
# Load The Datasets
df = pd.read_csv("D:\Machine-learning-algorithm\Linear Model\data\Salary_Data.
 ↪csv")
```

[3]:
```
df
```

[3]:

	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
```

[7]:
```
df.shape
```

[7]:
```
(30, 2)
```

[8]:
```
# Dividing the data two part X and Y
```

[9]:
```
X = df.drop(['Salary'],axis= True)
y = df['Salary']
```

[10]:
```
 # Spliting the data Traing And testing
```

[11]:
```
from sklearn.model_selection import train_test_split
X_train, X_test,y_train,y_test = train_test_split(X,y,test_size=(0.
 ↪2),random_state =42)
```

[12]:
```
X_train
```

[12]:
```
    YearsExperience
28             10.3
24              8.7
12              4.0
0               1.1
4               2.2
16              5.1
5               2.9
13              4.1
11              4.0
22              7.9
1               1.3
2               1.5
25              9.0
3               2.0
21              7.1
26              9.5
18              5.9
29             10.5
20              6.8
7               3.2
10              3.9
14              4.5
19              6.0
6               3.0
```

[ ]:
```
#Checking The Shape of Spliting data
```

[16]:
```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

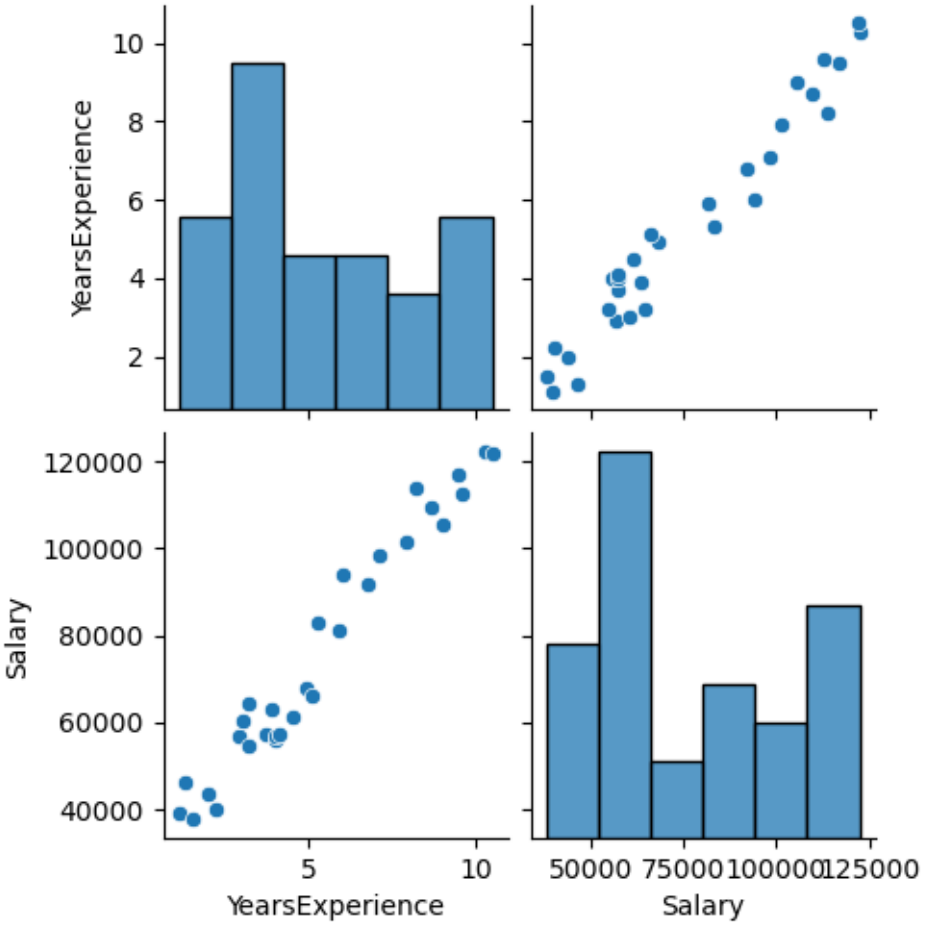
[16]:
```
((24, 1), (6, 1), (24,), (6,))
```

1 Visualize The Data

[17]:
```
# Visualize whole dataset
```

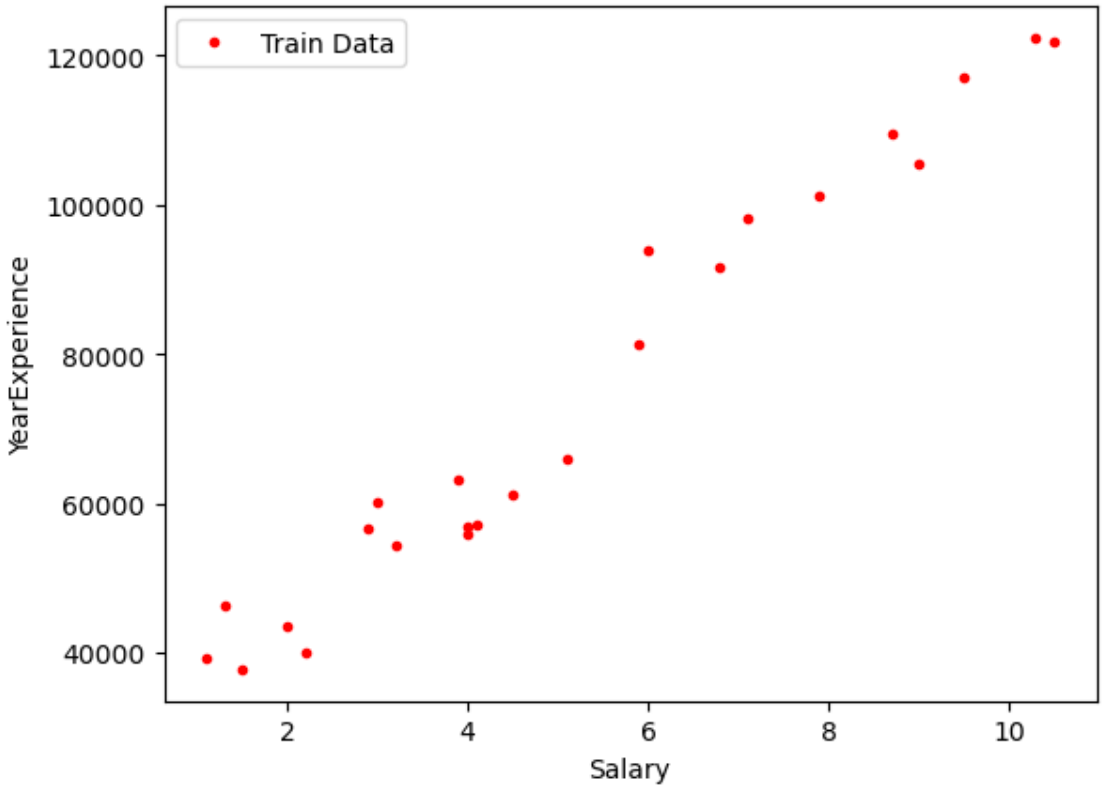
[19]:
```
sns.pairplot(df)
```

[19]:
```
<seaborn.axisgrid.PairGrid at 0x20e93111d90>
```



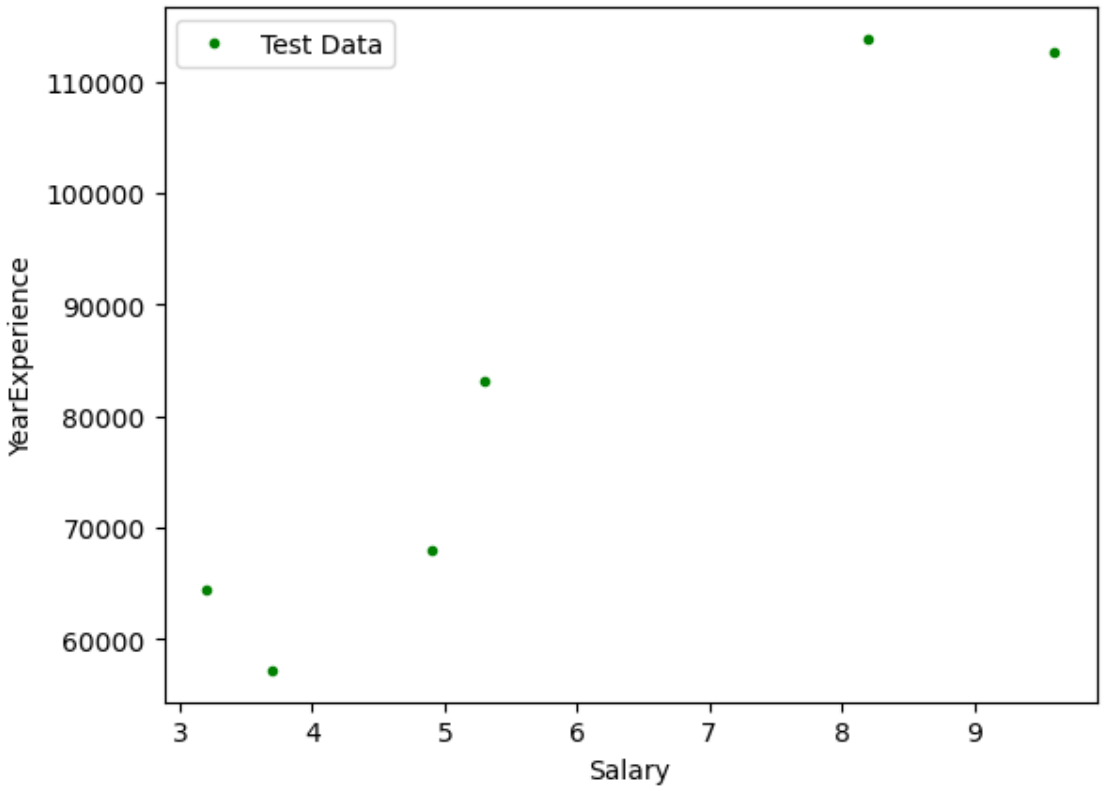
[20]:
```
# Visulize the Train Data
```

[25]:
```
plt.plot(X_train,y_train,'r.',label ="Train Data")
plt.xlabel('Salary')
plt.ylabel('YearExperience')
plt.legend()
plt.show()
```



[26]:
```
# Visualize the test data
```

[27]:
```
plt.plot(X_test,y_test,'g.',label='Test Data')
plt.xlabel('Salary')
plt.ylabel('YearExperience')
plt.legend()
plt.show()
```



2 Training the Simple Linear Regression

[32]:
```
from sklearn.linear_model import LinearRegression
lr =LinearRegression()
lr.fit(X_train,y_train)
```

[32]:
```
LinearRegression()
```

3 Prediction of Test Data

[33]:
```
Prediction = lr.predict(X_test).round(1)
```

[34]:
```
Prediction
```

[34]:
```
array([115790.2,  71498.3, 102596.9,  75267.8,  55477.8,  60189.7])
```

[36]:
```
y_test
```

[36]:
```
27    112635
15     67938
```

```
23    113812
17     83088
8      64445
9      57189
Name: Salary, dtype: int64
```

[38]:
```
calculation = pd.DataFrame(np.c_[y_test,Prediction], columns
↪=['Original_salary','Prdiction_salary'])
```

[39]:
```
calculation
```

[39]:

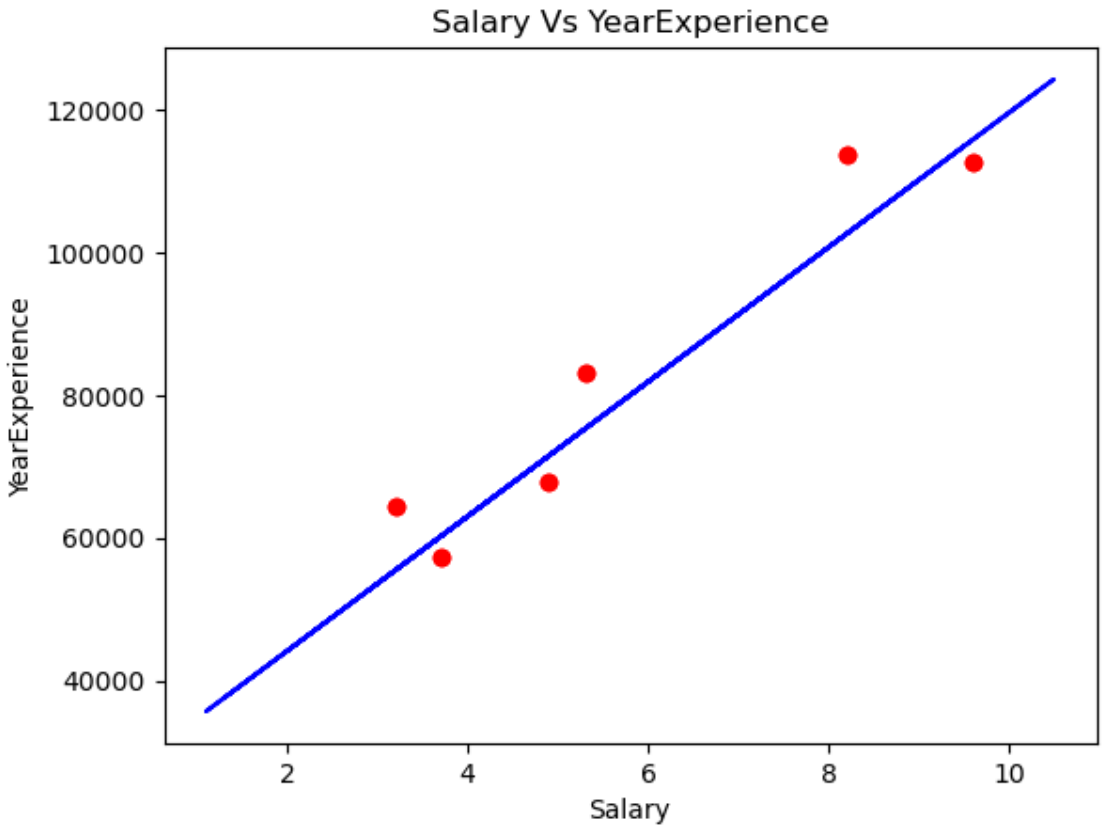
	Original_salary	Prdiction_salary
0	112635.0	115790.2
1	67938.0	71498.3
2	113812.0	102596.9
3	83088.0	75267.8
4	64445.0	55477.8
5	57189.0	60189.7

4 Visualising The training set Result

[42]:
```
plt.scatter(X_train,y_train, color = 'red')
plt.plot(X_train,lr.predict(X_train),color ="blue")
plt.title('salary vs yearExperience (Train Data)')
plt.xlabel('Salary')
plt.ylabel('YearExperience')
plt.show()
```

5 Visualising The Testing set Result

[44]:
```
plt.scatter(X_test,y_test, color = 'red')
plt.plot(X_train,lr.predict(X_train),color = 'blue')
plt.title("Salary Vs YearExperience")
plt.xlabel('Salary')
plt.ylabel('YearExperience')
plt.show()
```



[46]:
```
#Print Accuracy Score Traing And Test
```

[47]:
```
print("Training Accuracy :",lr.score(X_train,y_train))
print("Testing Accuracy :",lr.score(X_test,y_test))
```

```
Training Accuracy : 0.9645401573418146
Testing Accuracy : 0.9024461774180497
```

[ ]: