Salary By Years Of Experience

May 27, 2024

1 Ali Nawaz 2k22/CSE/20

2 Assignment : Salary And Years Of Experience

2.0.1 https://www.kaggle.com/datasets/harsh45/random-salary-data-of-employes-age-wise

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from pandas.core.common import random_state
from sklearn.linear_model import LinearRegression
```

[2]:
```
df=pd.read_csv('Salary_Data.xls')
df.head()
```

[2]:

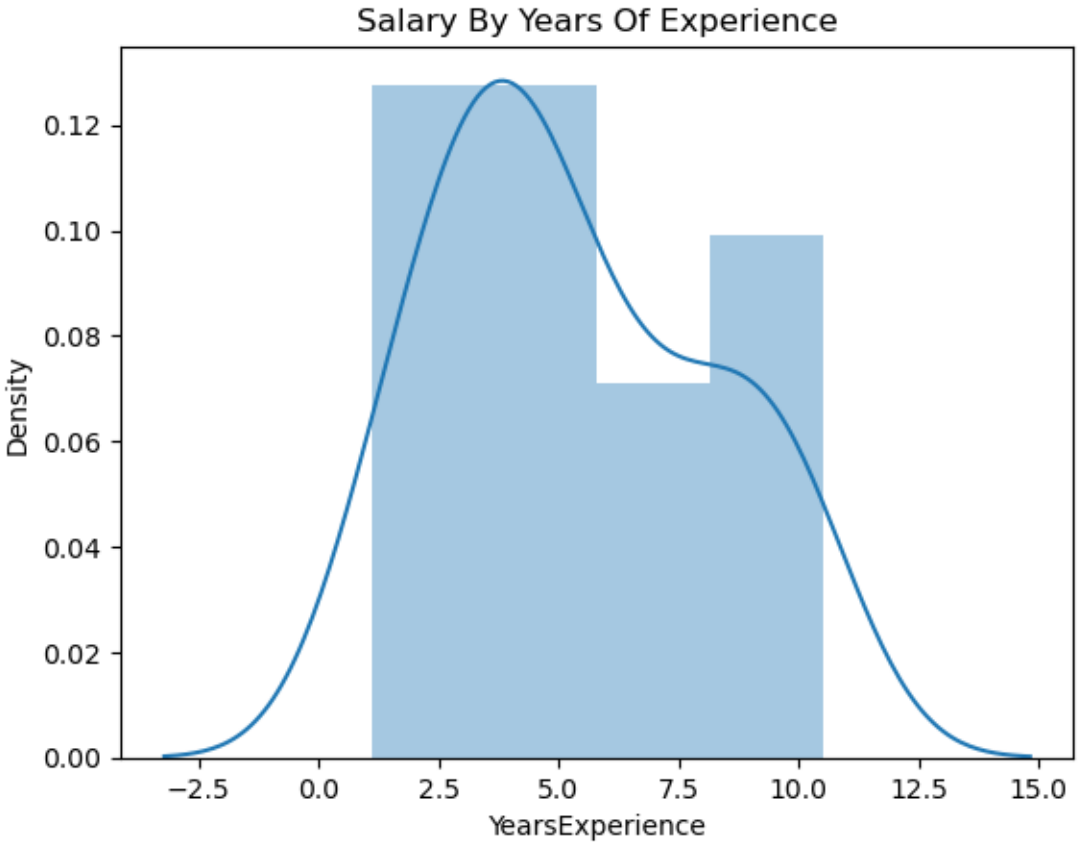
	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

[3]:
```
df.info()
```

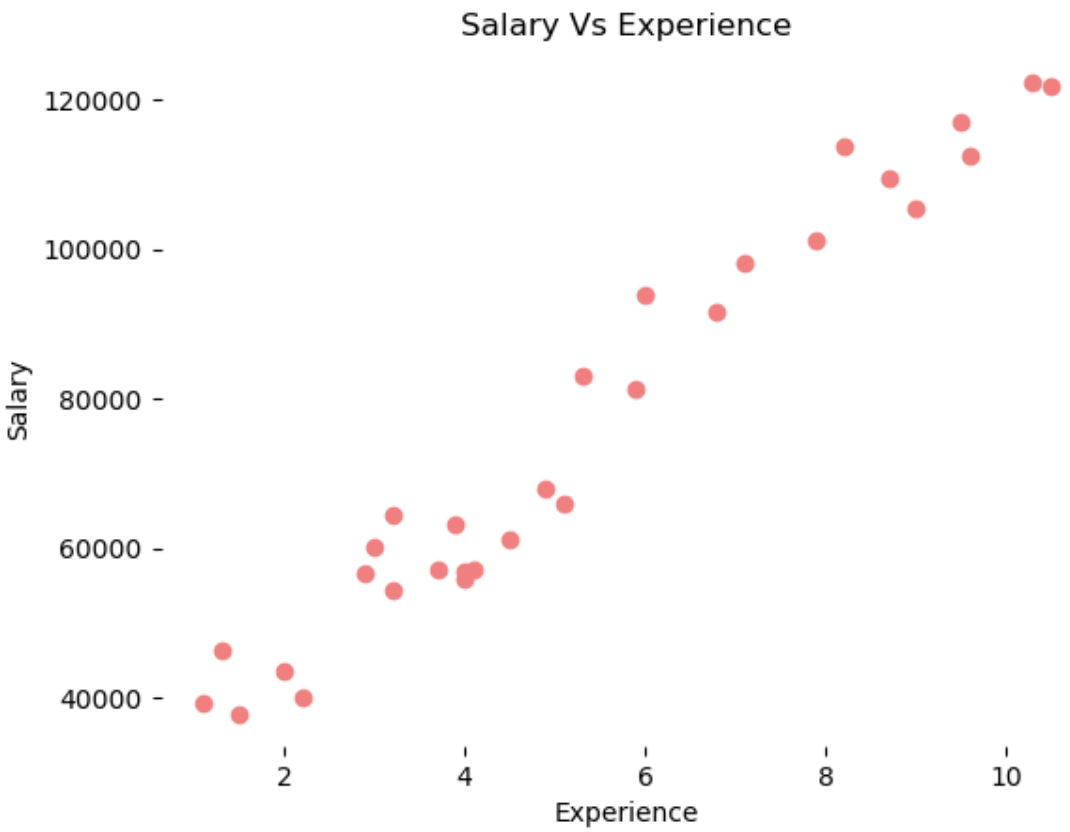
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 612.0 bytes
```

[5]:
```
plt.title('Salary By Years Of Experience')
sns.distplot(df['YearsExperience'])
plt.show()
```

```
C:\Users\beydaah\AppData\Local\Temp\ipykernel_13396\3975537158.py:2:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['YearsExperience'])
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



[6]:
```
plt.scatter(df['YearsExperience'], df['Salary'], color = 'lightcoral')
plt.title('Salary Vs Experience')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.box(False)
plt.show()
```



[7]:
```
X = df[['YearsExperience']]
y = df['Salary']
```

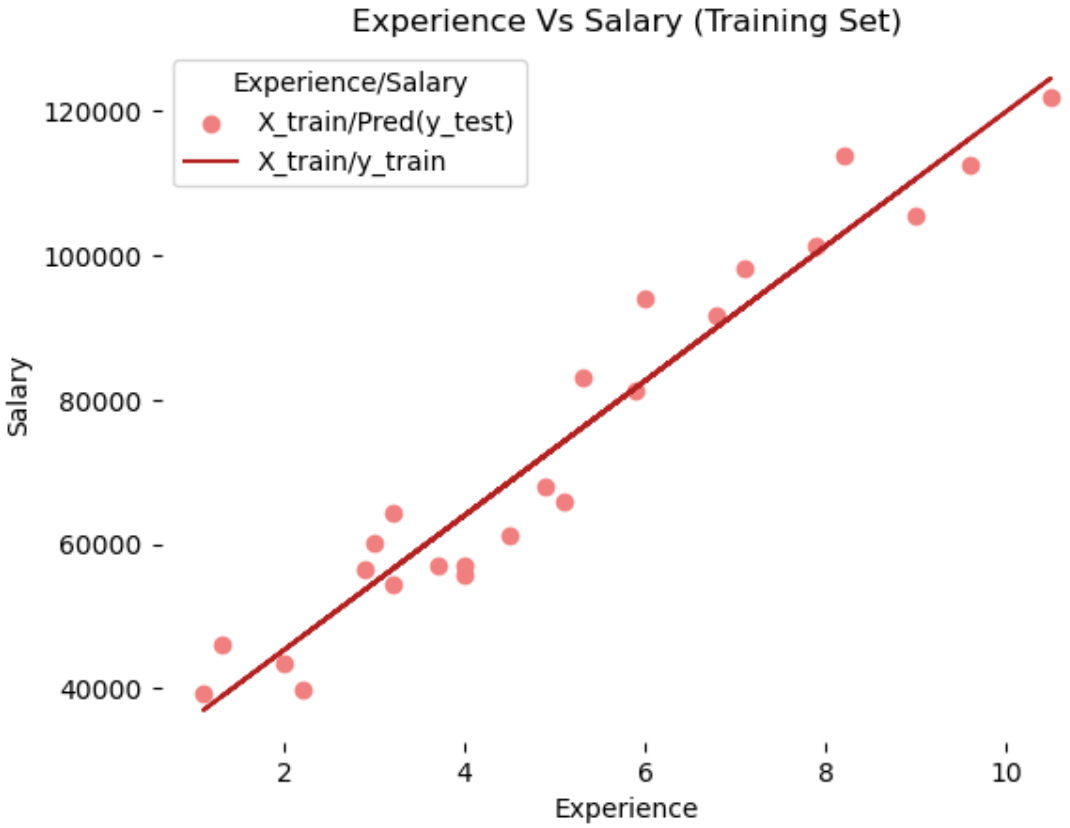
[8]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=0)
```

[9]:
```
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

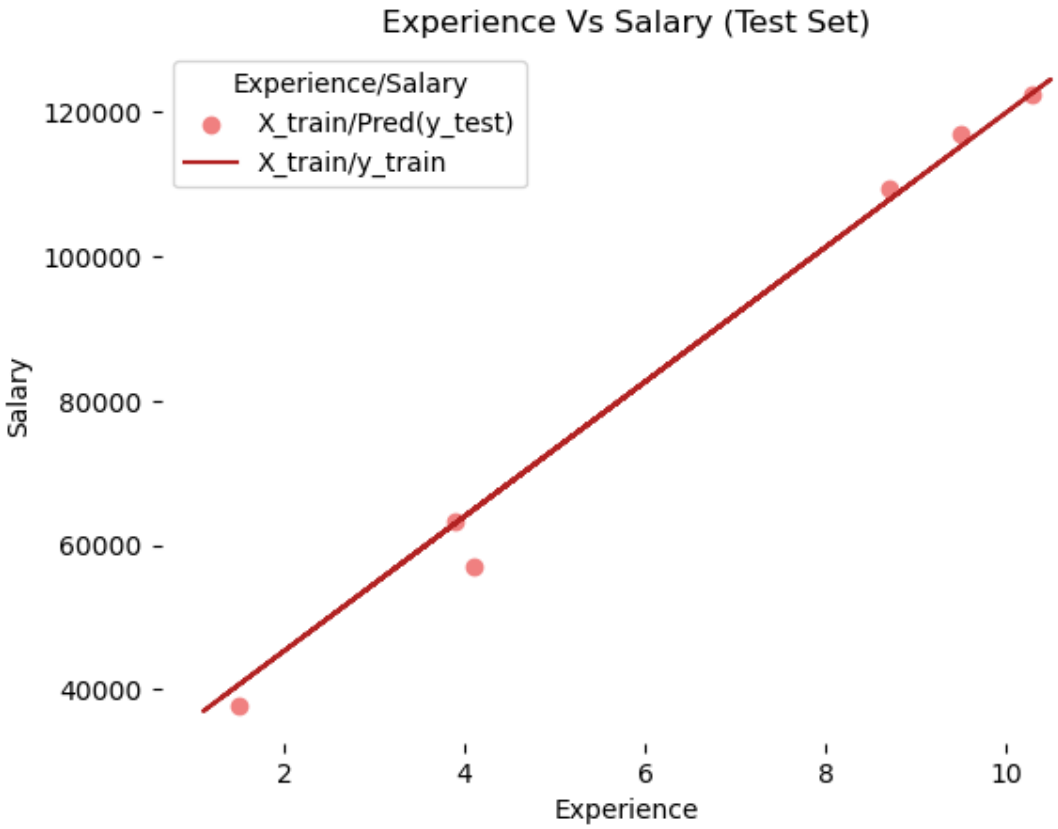
[9]:
```
LinearRegression()
```

[10]:
```
y_pred_test = regressor.predict(X_test)
y_pred_train = regressor.predict(X_train)
```

[11]:
```
plt.scatter(X_train, y_train, color = 'lightcoral')
plt.plot(X_train, y_pred_train, color = 'firebrick')
plt.title('Experience Vs Salary (Training Set)')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.legend(['X_train/Pred(y_test)', 'X_train/y_train'], title = 'Experience/
 ↪Salary', loc='best', facecolor='white')
plt.box(False)
plt.show()
```



[12]:
```
plt.scatter(X_test, y_test, color = 'lightcoral')
plt.plot(X_train, y_pred_train, color = 'firebrick')
plt.title('Experience Vs Salary (Test Set)')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.legend(['X_train/Pred(y_test)', 'X_train/y_train'], title = 'Experience/
 ↪Salary', loc='best', facecolor='white')
plt.box(False)
plt.show()
```


Experience Vs Salary (Test Set)
Experience/Salary
X_train/Pred(y_test)
X_train/y_train
120000
100000
80000
60000
40000
Salary
2
4
6
8
10
Experience

[ ]: