Lab 13 26-10-2022

October 26, 2022

1 Linear Regression

[5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

1.1 Load Dataset

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

[3]:
```
print(df)
```

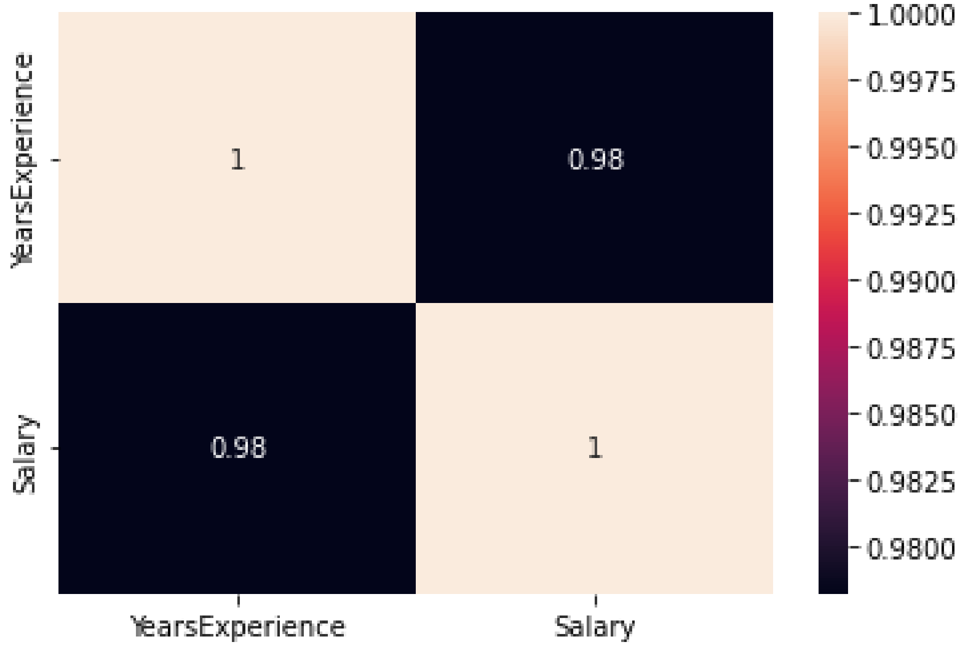
```
    YearsExperience    Salary
0               1.1   39343.0
1               1.3   46205.0
2               1.5   37731.0
3               2.0   43525.0
4               2.2   39891.0
5               2.9   56642.0
6               3.0   60150.0
7               3.2   54445.0
8               3.2   64445.0
9               3.7   57189.0
10              3.9   63218.0
11              4.0   55794.0
12              4.0   56957.0
13              4.1   57081.0
14              4.5   61111.0
15              4.9   67938.0
16              5.1   66029.0
17              5.3   83088.0
18              5.9   81363.0
19              6.0   93940.0
20              6.8   91738.0
21              7.1   98273.0
22              7.9  101302.0
```

```
23              8.2   113812.0
24              8.7   109431.0
25              9.0   105582.0
26              9.5   116969.0
27              9.6   112635.0
28             10.3   122391.0
29             10.5   121872.0
```

1.2 Heat Map

[7]:
```python
sns.heatmap(df.corr(),annot=True)
```

[7]: <AxesSubplot:>



1.3 Extraction of Variables

[8]:
```python
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.]
```

1.4 Dataset Splitting

[10]:
```
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)
```

[11]:
```
print(x_train)
```

```
[[ 2.9]
 [ 5.1]
 [ 3.2]
 [ 4.5]
 [ 8.2]
 [ 6.8]
 [ 1.3]
 [10.5]
 [ 3. ]
 [ 2.2]
```

```
 [ 5.9]
 [ 6. ]
 [ 3.7]
 [ 3.2]
 [ 9. ]
 [ 2. ]
 [ 1.1]
 [ 7.1]
 [ 4.9]
 [ 4. ]]
```

1.5 Fitting Dataset to Simple linear regression

[12]:
```python
from sklearn.linear_model import LinearRegression
regression = LinearRegression()
regression.fit(x_train, y_train)
```

[12]:
```
LinearRegression()
```

1.6 Prediction

[13]:
```python
y_pred= regression.predict(x_test)
x_pred= regression.predict(x_train)
```

1.7 Visualisation

1.7.1 Training Set

[30]:
```python
plt.scatter(x_train, y_train)
plt.plot(x_train, x_pred, color='g')
plt.title('"Salary vs Experience (Training Dataset)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary(In Rupees)')
```

[30]:
```
Text(0, 0.5, 'Salary(In Rupees)')
```

1.7.2 Testing Set

[20]:
```
plt.scatter(x_test, y_test)
plt.plot(x_test, y_pred, color='black')
plt.title('"Salary vs Experience (Testing Dataset)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary(In Rupees)')
```

[20]: Text(0, 0.5, 'Salary(In Rupees)')

1.8 Test Evaluation

1.8.1 Test Accuracy

[22]:
```
print(regression.score(x_test, y_test))
```

```
0.9749154407708353
```

1.8.2 Train Accuracy

[23]:
```
print(regression.score(x_train, y_train))
```

```
0.9381900012894278
```

[25]:
```
print(regression.intercept_)
print(regression.coef_)
```

```
26816.192244031183
[9345.94244312]
```

1.8.3 Error Values

[26]:
```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test,␣
↪y_pred)))
```

```
Mean Absolute Error: 3426.4269374307078
Mean Squared Error: 21026037.329511296
Root Mean Squared Error: 4585.4157204675885
```