5odru7v6p

December 13, 2024

#Simple Lineare Regression

[3]:
```
#Exp no. : 9
```

[5]:
```
#Aim : Simple linear Regression
```

[7]:
```
#Name :  Devesh J Arbat
#Roll no. : 06
#Section : A
```

[40]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

[41]:
```
import os
```

[42]:
```
os.getcwd()
```

[42]:
```
'C:\\Users\\salik\\DSS Practical'
```

[43]:
```
os.chdir('C:\\Users\\salik\\DSS Practical')
```

[44]:
```
df=pd.read_csv("Salary.csv")
```

[52]:
```
df.head()
```

[52]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

[54]:
```
df.tail()
```

[54]:

	YearsExperience	Salary
30	11.2	127345

```
31              11.5  126756
32              12.3  128765
33              12.9  135675
34              13.5  139465
```

[56]:
```
df.head(30)
```

[56]:
```
    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
```

[58]:
```
df.describe()
```

[58]:
```
       YearsExperience         Salary
count        35.000000      35.000000
mean          6.308571   83945.600000
std           3.618610   32162.673003
min           1.100000   37731.000000
25%           3.450000   57019.000000
```

```
50%            5.300000   81363.000000
75%            9.250000  113223.500000
max           13.500000  139465.000000
```

[60]:
```
df.shape
```

[60]:
```
(35, 2)
```

[62]:
```
df.size
```

[62]:
```
70
```

[64]:
```
df.ndim
```

[64]:
```
2
```

[66]:
```
df.isnull().sum()
```

[66]:
```
YearsExperience    0
Salary             0
dtype: int64
```

[68]:
```
#Assiging values in X & Y
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
```

[70]:
```
print(X)
```

```
[[ 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]
 [11.2]
 [11.5]
 [12.3]
 [12.9]
 [13.5]]
```

[72]:
```
print(y)
```

```
[ 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
 127345 126756 128765 135675 139465]
```

[74]:
```
#Splitting testdata into X_train,X_test,y_train,y_test
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.
 ↪3,random_state=42)
```

[76]:
```
print(X_train)
```

```
[[12.9]
 [ 1.1]
 [ 2.2]
 [ 5.3]
 [ 9.6]
 [ 2.9]
 [ 4. ]
 [ 1.3]
 [ 1.5]
 [12.3]
 [ 2. ]
 [11.2]
 [ 8.2]
 [11.5]
 [ 3.9]
 [ 7.9]
 [ 5.9]
```

```
 [ 9. ]
 [ 3. ]
 [ 6.8]
 [13.5]
 [ 3.2]
 [ 4.5]
 [10.3]]
```

[78]:
```
print(y_train)
```

```
[135675  39343  39891  83088 112635  56642  55794  46205  37731 128765
  43525 127345 113812 126756  63218 101302  81363 105582  60150  91738
 139465  54445  61111 122391]
```

[80]:
```
print (y_test)
```

```
[116969  57081 109431  98273  67938 121872  93940  56957  64445  66029
  57189]
```

[88]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

[88]:
```
LinearRegression()
```

[90]:
```
#Assigning Coefficient (slope) to m
m = lr.coef_
```

[92]:
```
print("Coefficient :" , m)
```

```
Coefficient : [8555.33918938]
```

[94]:
```
#Assigning Y-intercept to a
c = lr.intercept_
```

[96]:
```
print("Intercept : ", c)
```

```
Intercept :  29602.07353482095
```

[98]:
```
lr.score(X_test,y_test) * 100
```

[98]:
```
91.71426108885098
```

[ ]:
```

```