

vc63d50kk

October 17, 2024

```
[1]: #Aim : To perform Simple Linear Regression and find out the coefficients of it.
```

```
[3]: # Name : Ritika Rajesh Junekar  
# Roll no : 30  
# Sec: C  
# Subject : ET1
```

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

```
[6]: import os
```

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

```
[9]: 'C:\\Users\\USER'
```

```
[11]: os.chdir("C:\\Users\\USER\\Desktop\\")
```

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

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

```
[15]:
```

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

```
[17]: df.head(10)
```

```
[17]:
```

	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

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   YearsExperience  35 non-null    float64
1   Salary          35 non-null    int64
dtypes: float64(1), int64(1)
memory usage: 692.0 bytes
```

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

```
[21]:   YearsExperience  Salary
30          11.2  127345
31          11.5  126756
32          12.3  128765
33          12.9  135675
34          13.5  139465
```

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

```
[23]:   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
```

```
[25]: df.shape
```

```
[25]: (35, 2)
```

```
[27]: df.size
```

```
[27]: 70
```

```
[29]: df.ndim
```

```
[29]: 2
```

```
[31]: df.isnull()
```

```
[31]:
```

	YearsExperience	Salary
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False
20	False	False
21	False	False
22	False	False
23	False	False
24	False	False
25	False	False
26	False	False
27	False	False
28	False	False
29	False	False
30	False	False
31	False	False
32	False	False
33	False	False
34	False	False

```
[33]: df.isnull()
```

```
[33]:      YearsExperience  Salary
      0             False   False
      1             False   False
      2             False   False
      3             False   False
      4             False   False
      5             False   False
      6             False   False
      7             False   False
      8             False   False
      9             False   False
     10             False   False
     11             False   False
     12             False   False
     13             False   False
     14             False   False
     15             False   False
     16             False   False
     17             False   False
     18             False   False
     19             False   False
     20             False   False
     21             False   False
     22             False   False
     23             False   False
     24             False   False
     25             False   False
     26             False   False
     27             False   False
     28             False   False
     29             False   False
     30             False   False
     31             False   False
     32             False   False
     33             False   False
     34             False   False
```

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

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

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

```
[37]:      YearsExperience  Salary
      0             1.1   39343
```

1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

```
[39]: df.columns
```

```
[39]: Index(['YearsExperience', 'Salary'], dtype='object')
```

```
[41]: df.loc[4, "Salary"]
```

```
[41]: 39891
```

```
[43]: df.head(15)
```

```
[43]:
```

	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

```
[45]: df.loc[2, "YearsExperience"]
```

```
[45]: 1.5
```

```
[47]: df.loc[12]
```

```
[47]: YearsExperience    4.0
Salary                56957.0
Name: 12, dtype: float64
```

```
[49]: df.loc[4]
```

```
[49]: YearsExperience    2.2
Salary                39891.0
Name: 4, dtype: float64
```

```
[51]: a=(1,2,3,4,5,6,7,8,9,10)
```

```
[53]: a[1:4]
```

```
[53]: (2, 3, 4)
```

```
[55]: df.loc[0:3,'YearsExperience':'Salary']
```

```
[55]:
```

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

```
[57]: df.iloc[1,0]
```

```
[57]: 1.3
```

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

```
[59]:
```

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

```
[61]: df.loc[1,"Salary"]
```

```
[61]: 46205
```

```
[63]: #Assigning values in X & Y  
x=df.iloc[:, :-1].values  
y=df.iloc[:, :-1].values
```

```
[65]: a[:2]
```

```
[65]: (1, 2)
```

```
[67]: a[2:]
```

```
[67]: (3, 4, 5, 6, 7, 8, 9, 10)
```

```
[69]: a[1:6:2]
```

```
[69]: (2, 4, 6)
```

```
[71]: 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]]
```

```
[73]: 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]
[11.2]
[11.5]
[12.3]
[12.9]
[13.5]]
```

```
[75]: #splitting testdata into x_train,y_train'
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)
```

```
[77]: 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]]
```

```
[79]: print(x_test)
```

```
[[ 9.5]
 [ 4.1]
 [ 8.7]
 [ 7.1]
 [ 4.9]
[10.5]
 [ 6. ]
 [ 4. ]
 [ 3.2]
 [ 5.1]
 [ 3.7]]
```

```
[81]: print(y_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]]
```

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

```
[[ 9.5]
 [ 4.1]
 [ 8.7]
 [ 7.1]
 [ 4.9]
[10.5]
 [ 6. ]
 [ 4. ]
 [ 3.2]
 [ 5.1]
 [ 3.7]]
```

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

```
[85]: LinearRegression()
```

```
[87]: #Assigning coefficient (slope) to m
m = lr.coef_
```

```
[89]: print("Coefficient :",a)
```

```
Coefficient : (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

```
[ ]:
```