Assignment 1 Linear regession

June 30, 2023

1 Module 8: Linear Regression

Assignment Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python for Data Science Certification Course Problem Statement: You work in XYZ Company as a Python Data Scientist. The company officials have collected some data on salaries based on year of experience and wish for you to create a model from it. Dataset: data.csv Tasks To Be Performed: 1. Load the dataset using pandas 2. Extract data from years experience column is a variable named X 3. Extract data from the salary column is a variable named Y 4. Divide the dataset into two parts for training and testing in 66% and 33% proportion 5. Create and train Linear Regression Model on training set 6. Make predictions based on the testing set using the trained model 7. Check the performance by calculating the r2 score of the model

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

[2]:
```
## import your dataset
data=pd.read_csv(r'C:/Users/Vikas/Downloads/data.csv')
```

[4]:
```
## i want to see the first five rows
data.head()
```

[4]:

	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]:
```
## i want to see the all col
data.columns
```

[5]:
```
Index(['YearsExperience', 'Salary'], dtype='object')
```

[6]:
```
## i want to see the shape of data
data.shape
```

[6]:
```
(30, 2)
```

[7]:
```
## i want to see all information of my dataset.
data.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: 608.0 bytes
```

[8]:
```
## i wnt to see the count,min max,std,
data.describe().T
```

[8]:

	count	mean	std	min	25%	50%	75%	max
YearsExperience	30.0	5.313333	2.837888	1.1	3.20	4.7	7.70	10.5
Salary	30.0	76003.000000	27414.429785	37731.0	56720.75	65237.0	100544.75	122391.0

[9]:
```
##  i want to see the how much null values in my dataset
data.isnull().sum()
```

[9]:
```
YearsExperience    0
Salary             0
dtype: int64
```

[12]:
```
## extract the col YearsExperience  independent(x)
x=data.iloc[:,:-1]
```

[13]:
```
x
```

[13]:

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0
4	2.2
5	2.9
6	3.0
7	3.2
8	3.2

```
9               3.7
10              3.9
11              4.0
12              4.0
13              4.1
14              4.5
15              4.9
16              5.1
17              5.3
18              5.9
19              6.0
20              6.8
21              7.1
22              7.9
23              8.2
24              8.7
25              9.0
26              9.5
27              9.6
28             10.3
29             10.5
```

[14]:
```
## extract the col Salary  dependent(y)
y=data.iloc[:,1]
```

[15]:
```
y
```

[15]:
```
0      39343.0
1      46205.0
2      37731.0
3      43525.0
4      39891.0
5      56642.0
6      60150.0
7      54445.0
8      64445.0
9      57189.0
10     63218.0
11     55794.0
12     56957.0
13     57081.0
14     61111.0
15     67938.0
16     66029.0
17     83088.0
18     81363.0
19     93940.0
```

```
20     91738.0
21     98273.0
22    101302.0
23    113812.0
24    109431.0
25    105582.0
26    116969.0
27    112635.0
28    122391.0
29    121872.0
Name: Salary, dtype: float64
```

[16]:
```
##split the data into traning set and testing set,

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
x, y, test_size=0.33, random_state=0)
```

[17]:
```
 ## fitting the dataset into the training test and test set
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

[17]:
```
LinearRegression()
```

[19]:
```
 ## predicting the test set result
y_pred=lr.predict(x_test)
y_pred
```

[19]:
```
array([ 40835.10590871, 123079.39940819,  65134.55626083,  63265.36777221,
       115602.64545369, 108125.8914992 , 116537.23969801,  64199.96201652,
        76349.68719258, 100649.1375447 ])
```

[24]:
```
r2_score = r2_score(y_test,y_pred)


print(r2_score)
```

```
0.9749154407708353
```

[ ]:
```

```