Linear Regression

November 15, 2022

1 Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning.It performs a regression task. Regression models a target prediction value based on independent variables

[12]:
```
import pandas as pd
import numpy as np
df=pd.read_csv('Quiz Marks.csv')
print(df)
```

```
    Regd. Num. Quiz 1  Quiz 2
0   20761A0565    5.5     6.5
1   20761A0566    5.5     7.0
2   20761A0567    6.5     5.5
3   20761A0568    6.5     6.0
4   20761A0569      7     6.5
..         …    …      …
66  21765A0508      4     4.0
67  21765A0509    4.5     4.0
68  21765A0510      4     4.0
69  21765A0511    6.5     2.5
70  21765A0512      5     5.0

[71 rows x 3 columns]
```

[13]:
```
df['Quiz 1']=pd.to_numeric(df['Quiz 1'],errors='coerce')
```

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Regd. Num.  71 non-null     object
 1   Quiz 1      70 non-null     float64
 2   Quiz 2      71 non-null     float64
dtypes: float64(2), object(1)
```

```
memory usage: 1.8+ KB
```

[15]:
```
df.isna().sum()
```

[15]:
```
Regd. Num.    0
Quiz 1        1
Quiz 2        0
dtype: int64
```

[16]:
```
df['Quiz 1']=df['Quiz 1'].fillna(df['Quiz 1'].mean())
x=np.array(df['Quiz 1']).reshape([-1,1])
y=np.array(df['Quiz 2'])
```

[17]:
```
df.isna().sum()
```

[17]:
```
Regd. Num.    0
Quiz 1        0
Quiz 2        0
dtype: int64
```

[18]:
```
from sklearn.linear_model import LinearRegression
```

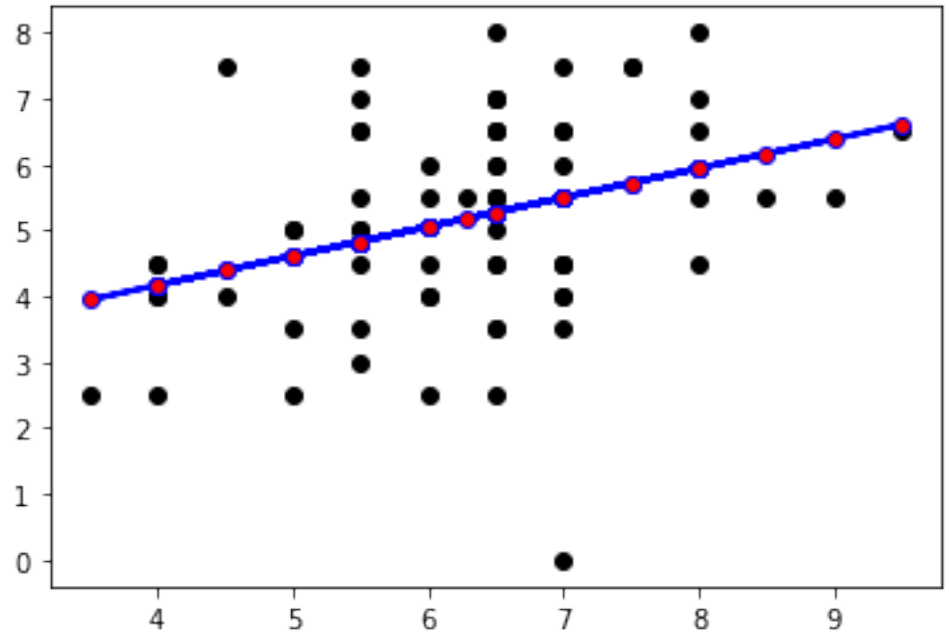
[19]:
```
model=LinearRegression()
model.fit(x,y)
result=model.score(x,y)
print("score =",result,"\n")
print("intercept=",model.intercept_,"\n")
print("slope=",model.coef_,"\n")
print("actual values of y \n",y,"\n")
ypred=model.predict(x)
print("predicted values of y\n ",ypred)
```

```
score = 0.11410179774327167

intercept= 2.394396937276719

slope= [0.4431535]

actual values of y
 [6.5 7.  5.5 6.  6.5 4.5 7.5 6.  5.  7.  6.5 7.5 7.5 4.  4.5 6.5 6.5 5.5
 8.  4.  3.  2.5 7.  5.5 3.5 6.  4.5 5.5 6.5 5.5 3.5 4.5 8.  5.  4.  3.5
 3.5 5.5 4.5 4.5 6.5 5.5 6.  7.  6.5 4.5 4.  2.5 4.5 0.  5.5 5.5 7.5 5.5
 5.5 5.  3.5 5.  4.5 4.5 7.  7.5 6.5 2.5 5.5 2.5 4.  4.  4.  2.5 5. ]

predicted values of y
  [4.83174118 4.83174118 5.27489467 5.27489467 5.49647142 4.16701093
 5.49647142 5.49647142 4.61016443 5.27489467 5.49647142 5.71804817
 4.83174118 5.05331792 4.83174118 5.27489467 5.93962492 4.83174118
```

```
5.27489467 5.49647142 4.83174118 3.94543418 5.27489467 5.49647142
4.61016443 5.05331792 5.49647142 6.38277842 5.27489467 5.05331792
4.83174118 5.05331792 5.93962492 4.83174118 5.49647142 5.27489467
5.27489467 5.27489467 5.93962492 5.27489467 4.83174118 5.49647142
5.27489467 5.93962492 6.60435517 5.49647142 5.05331792 5.05331792
5.49647142 5.49647142 5.18309859 6.16120167 5.71804817 5.27489467
5.93962492 4.83174118 5.49647142 5.27489467 5.27489467 4.16701093
5.27489467 4.38858768 5.27489467 4.61016443 5.49647142 4.16701093
4.16701093 4.38858768 4.16701093 5.27489467 4.61016443]
```

[20]:
```
import matplotlib.pyplot as plt
plt.scatter(x,y,color='black')
plt.plot(x,ypred,color='blue',linewidth=2,marker='o',markerfacecolor='red')
```

[20]:
```
[<matplotlib.lines.Line2D at 0x7f25124dba60>]
```



[22]:
```
from sklearn.metrics import r2_score
r2_score(y,ypred)
```

[22]:
```
0.11410179774327167
```