Multiple Linear Regression

Presented By
Dr.A.Bazila Banu,
Prof & Head/AIML

•

Multiple linear regression

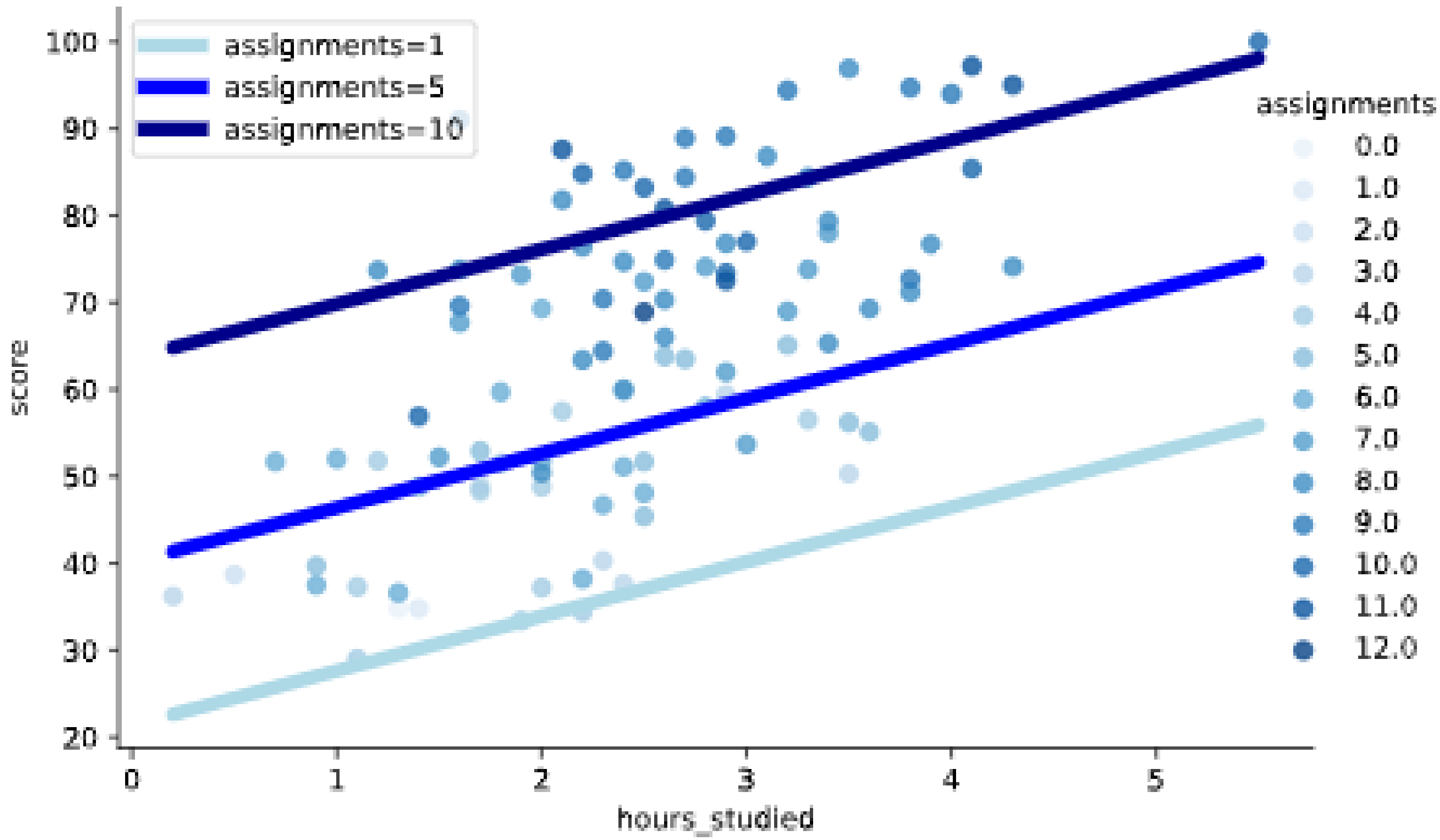
- Multiple linear regression is a method we can use to quantify the relationship between two or more independent variables(X1,X2…) and a dependent variable(Y).
- **Multiple Linear Regression Formula**

$$yi = \beta 0 + \beta 1xi1 + \beta 2xi2 + \ldots + \beta pxip + \epsilon$$

Where:

- **yi** is the dependent or predicted variable
- **β0** is the y-intercept, i.e., the value of y when both xi and x2 are 0.
- **β1** and **β2** are the regression coefficients representing the change in y relative to a one-unit change in **xi1** and **xi2**, respectively.
- **βp** is the slope coefficient for each independent variable
- **ϵ** is the model's random error (residual) term.

Visualizing a Multiple Regression Model



Steps to calculate

- **Step 1: Calculate X_1^2, X_2^2, X_1y, X_2y and X_1X_2 and Regression Sums.**
- **Step 3: Calculate b_0, b_1, and b_2.**
- **Step 5: Place b_0, b_1, and b_2 in the estimated linear regression equation.**

Sample Data Set

y	X_1	X_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11

	y	X_1	X_2
	140	60	22
	155	62	25
	159	67	24
	179	70	20
	192	71	15
	200	72	14
	212	75	14
	215	78	11
Mean	181.5	69.375	18.125
Sum	1452	555	145

	X_1^2	X_2^2	X_1y	X_2y	X_1X_2
	3600	484	8400	3080	1320
	3844	625	9610	3875	1550
	4489	576	10653	3816	1608
	4900	400	12530	3580	1400
	5041	225	13632	2880	1065
	5184	196	14400	2800	1008
	5625	196	15900	2968	1050
	6084	121	16770	2365	858
Sum	38767	2823	101895	25364	9859
Reg Sums	263.875	194.875	1162.5	-953.5	-200.375

Calculate b_0, b_1, and b_2.

- The formula to calculate

- $b_1 = [(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)] / [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$

b1 = [(194.875)(1162.5) – (-200.375)(-953.5)] /
[(263.875) (194.875) – (-200.375)2]
= 3.148

- The formula to calculate

b2 = [(Σx12)(Σx2y) – (Σx1x2)(Σx1y)] / [(Σx12) (Σx22) – (Σx1x2)2]

b2 = [(263.875)(-953.5) – (-200.375)(1152.5)] / [(263.875) (194.875) – (-200.375)2]
= -1.656

The formula to calculate

$b_0 = y - b1X1 - b2X2$

- Thus, b0 = 181.5 – 3.148(69.375) – (-1.656)(18.125) = **-6.867**

Estimated linear regression equation is

- $\hat{y} = -6.867 + 3.148x_1 – 1.656x_2$

How to Interpret a Multiple Linear Regression Equation

- Here is how to interpret this estimated linear regression equation: $\hat{y}$ = -6.867 + $3.148x_1 - 1.656x_2$ **b_0 = -6.867**. When both predictor variables are equal to zero, the mean value for y is -6.867.
- **b_1 = 3.148**. A one unit increase in x_1 is associated with a 3.148 unit increase in y, on average, assuming x_2 is held constant.
- **b_2 = -1.656**. A one unit increase in x_2 is associated with a 1.656 unit decrease in y, on average, assuming x_1 is held constant.

Multiple Linear Regression in Python

- ***Step 1: Load the Boston dataset***
-

```
import pandas as pd
import numpy as np
dataset = pd.read_csv('Boston1.csv')
dataset
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9
10	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15.0
11	0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.90	13.27	18.9
12	0.09378	12.5	7.87	0	0.524	5.889	39.0	5.4509	5	311	15.2	390.50	15.71	21.7
13	0.62976	0.0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21.0	396.90	8.26	20.4
14	0.63796	0.0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21.0	380.02	10.26	18.2
15	0.62739	0.0	8.14	0	0.538	5.834	56.5	4.4986	4	307	21.0	395.62	8.47	19.9

Medv as dependent variable (Y) and other columns as Independent Variables(X1,X2 …)

Split the data for X and Y

```
In [3]: X = pd.DataFrame(dataset.iloc[:,:-1])
        y = pd.DataFrame(dataset.iloc[:,-1])
```

Step 5: Divide the data into train and test sets:

```
In [6]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
```

Step 6: Train the algorithm:

```
In [8]: from sklearn.linear_model import LinearRegression
        regressor = LinearRegression()
        regressor.fit(X_train, y_train)
```

Step 7: Comparing the predicted value to the actual value:

```
In [13]: y_pred = regressor.predict(X_test)
         y_pred = pd.DataFrame(y_pred, columns=['Predicted'])
         y_pred
```

Step 10: Evaluate the algorithm

```
In [15]: 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: 3.2132704958423757
Mean Squared Error: 20.86929218377072
Root Mean Squared Error: 4.568292042303198
```