Mastering Machine Learning with Python
(27th Aug 2024 - 18th Oct 2024)

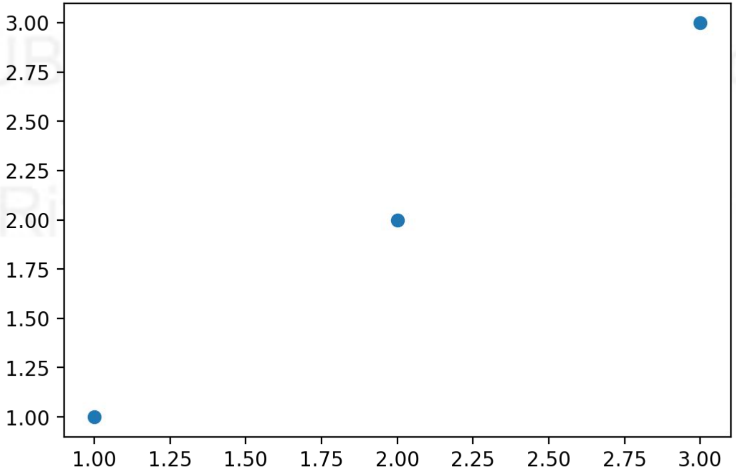
Linear Regression

iHUB DivyaSampark, IIT Roorkee
and
Ritvij Bharat Private Limited

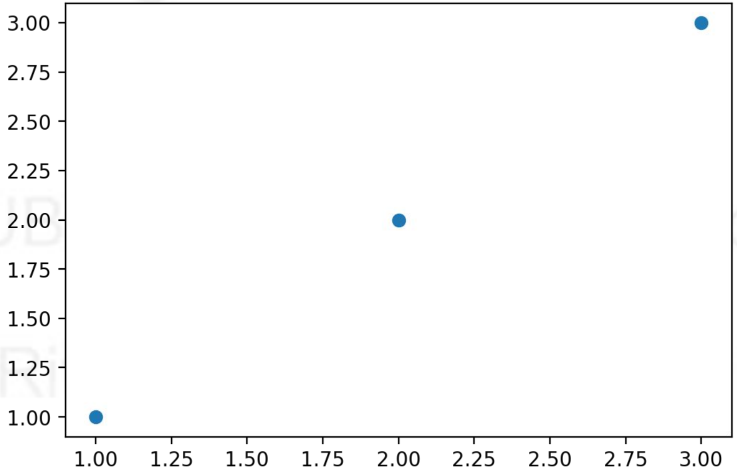
Led by : Shreyas Shukla

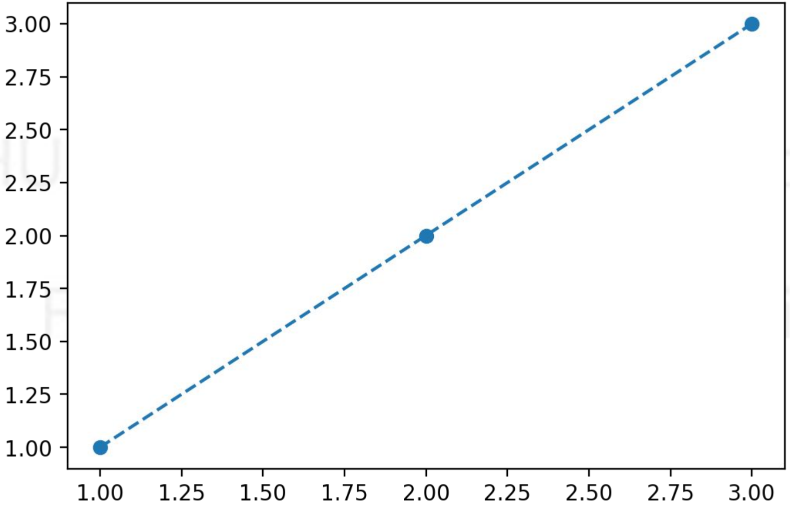
Mastering Machine Learning with Python

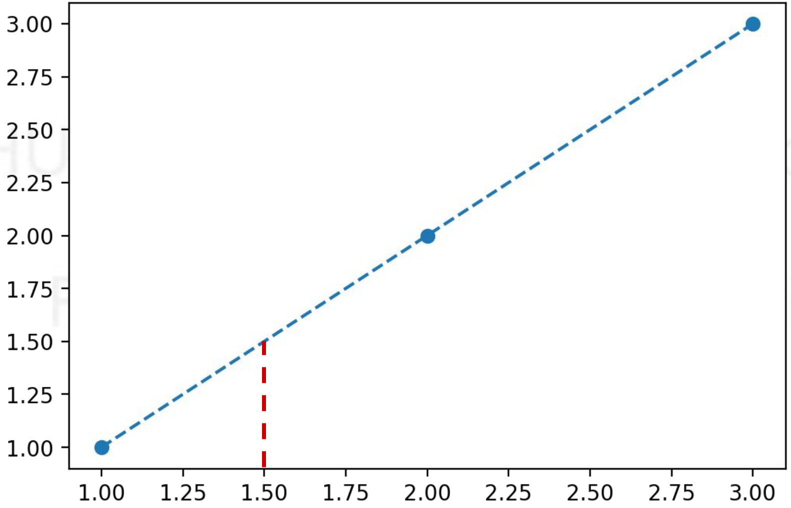
(27th Aug 2024 - 18th Oct 2024)

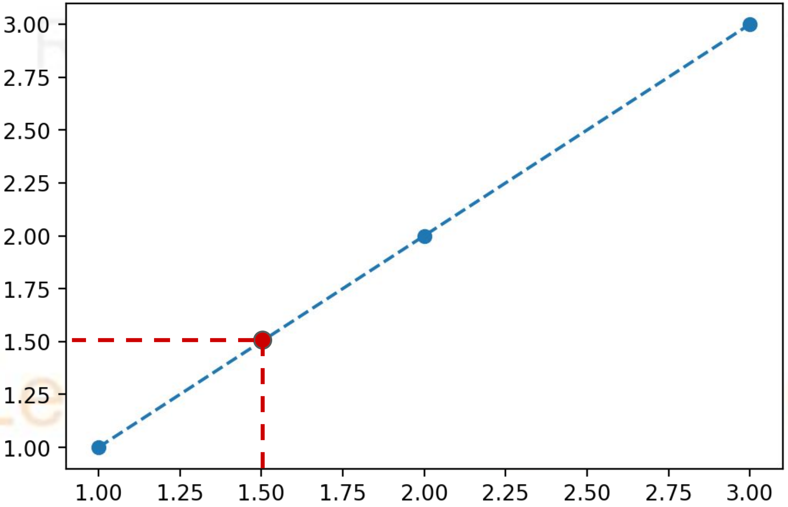


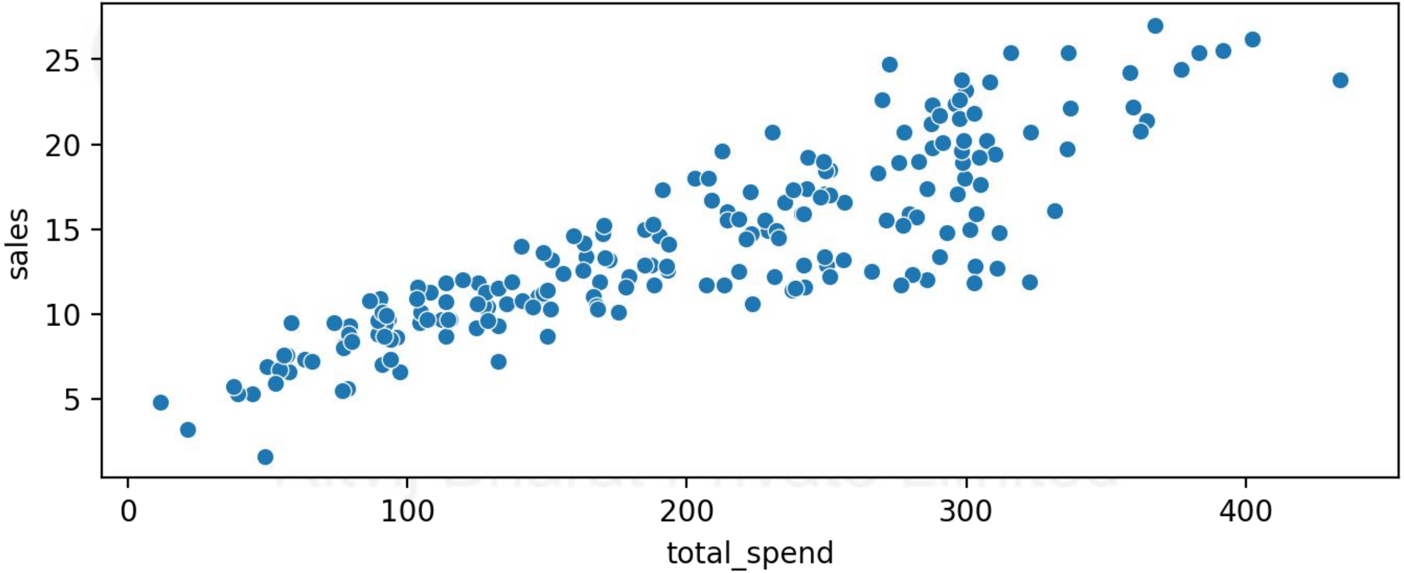
Led by : Shreyas Shukla

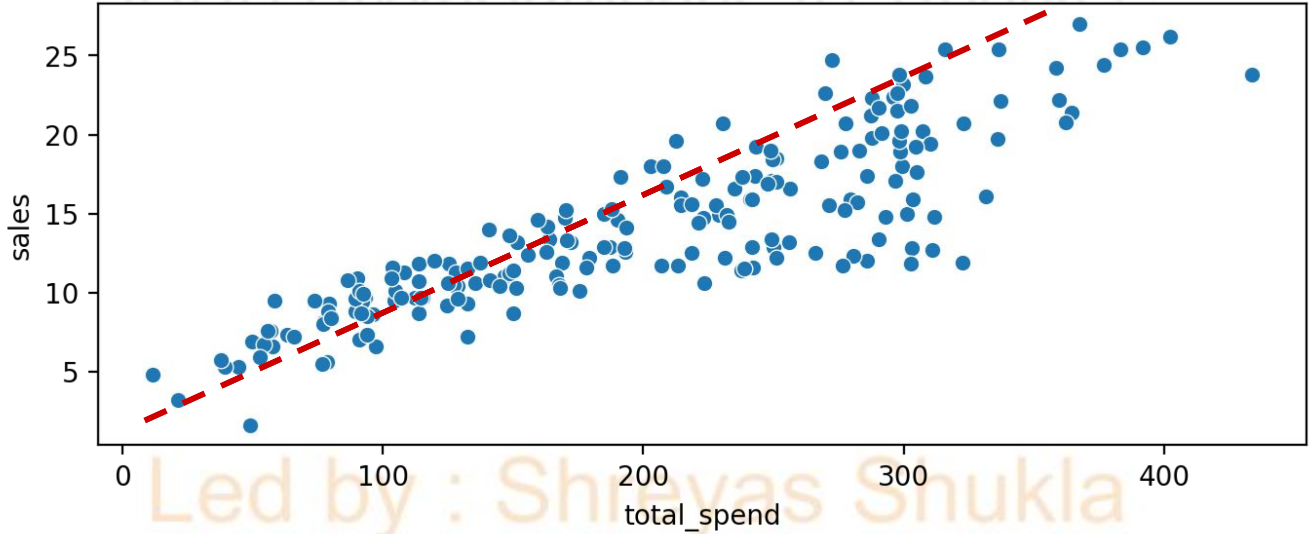

3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00

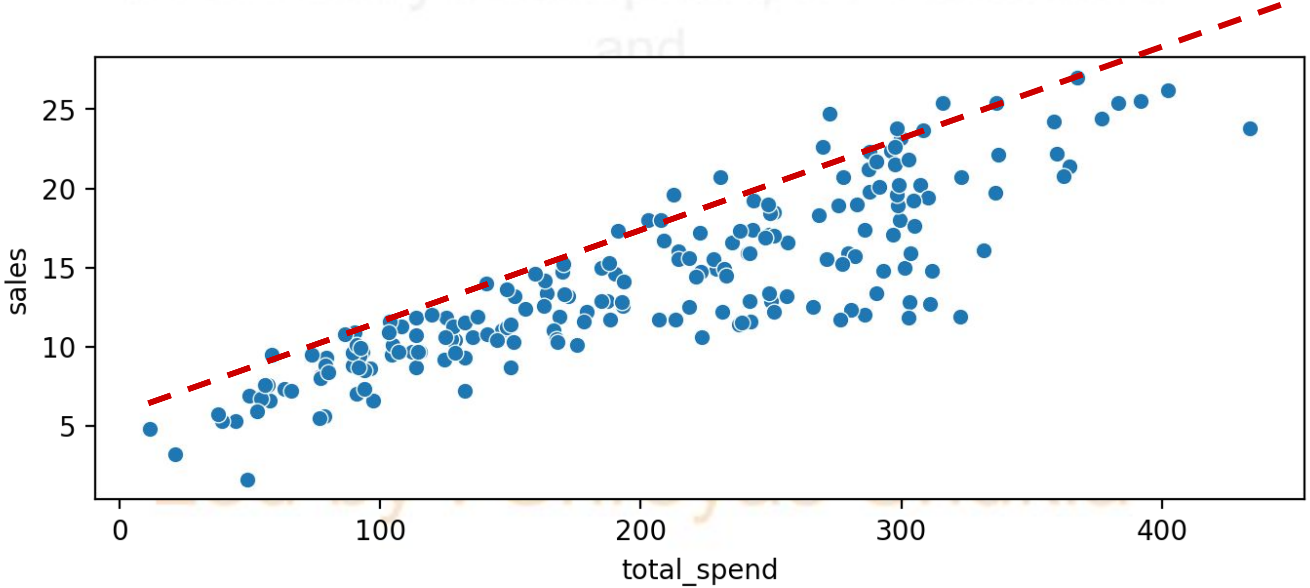

3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00


3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00


3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00


25
20
15
10
5
sales
0
100
200
300
400
total_spend


sales
25
20
15
10
5
0
100
200
300
400
total_spend


sales
total_spend
25
20
15
10
5
0
100
200
300
400

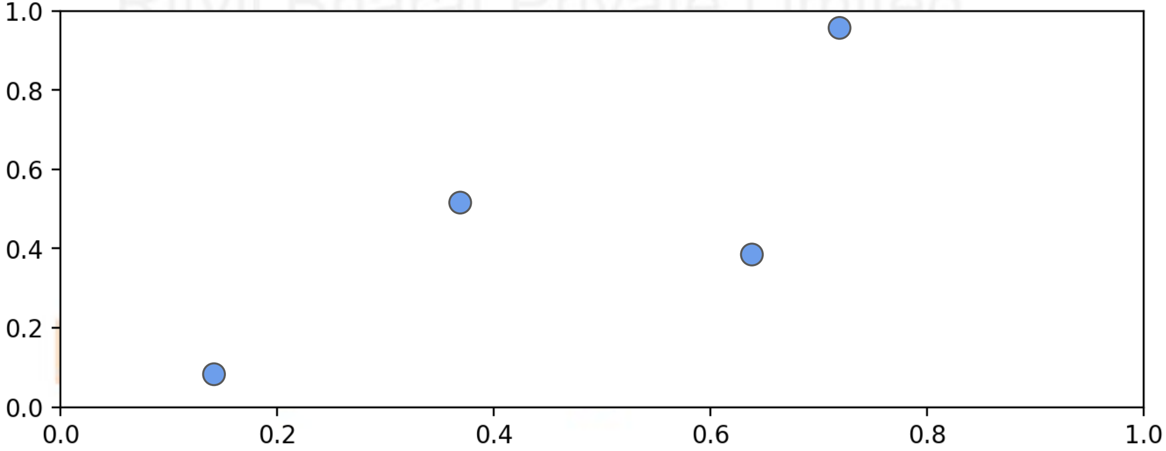
Mastering Machine Learning with Python

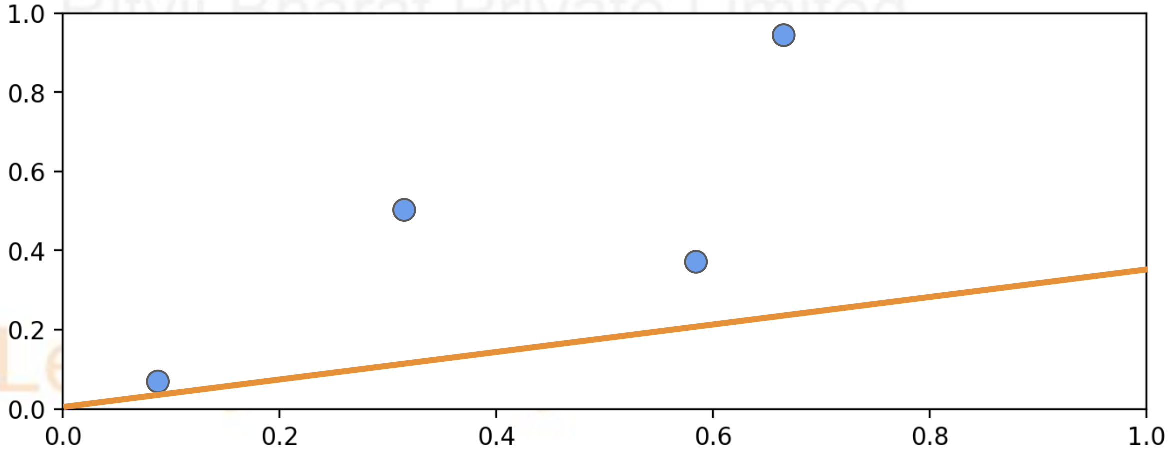
(27th Aug 2024 - 18th Oct 2024)

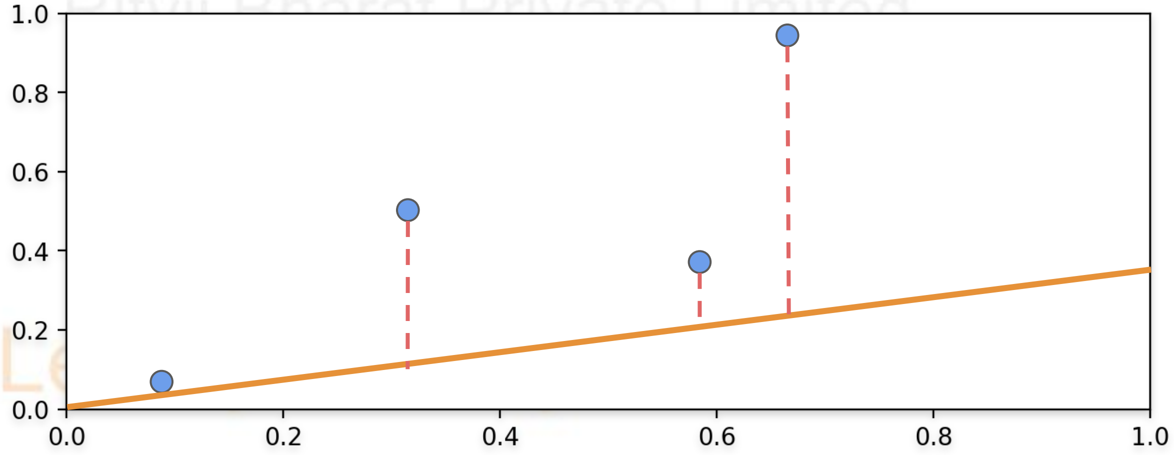
iHUB DivyaSampark, IIT Roorkee

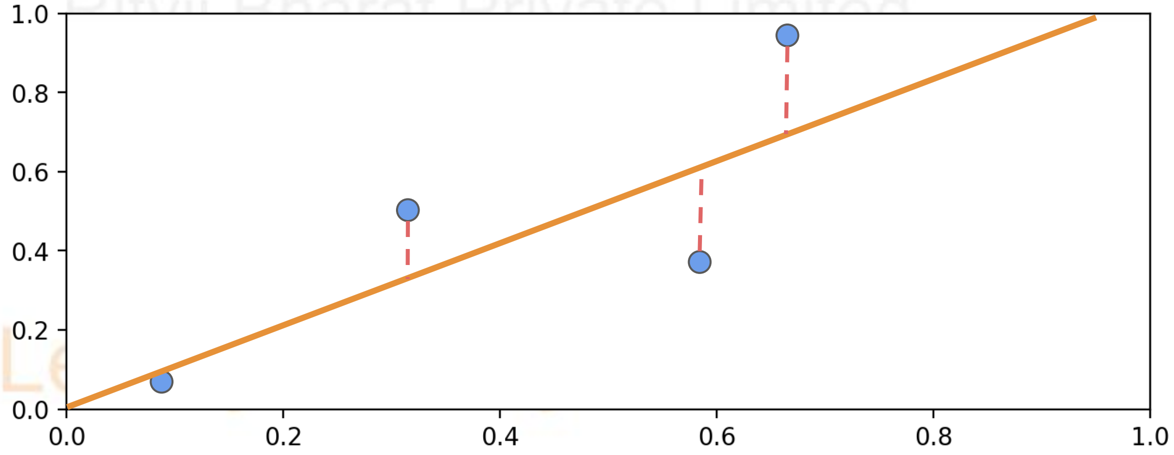
and

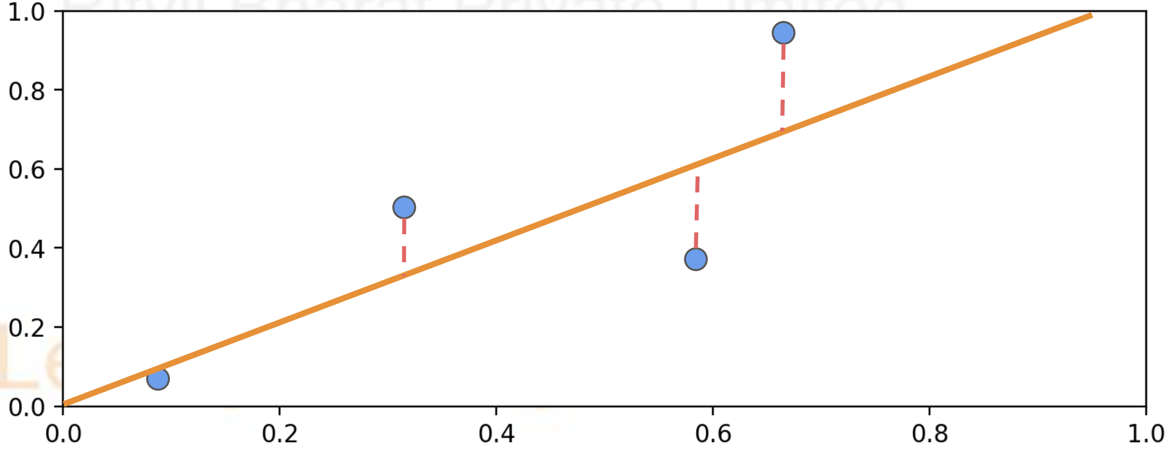
Ritvij Bharat Private Limited




1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

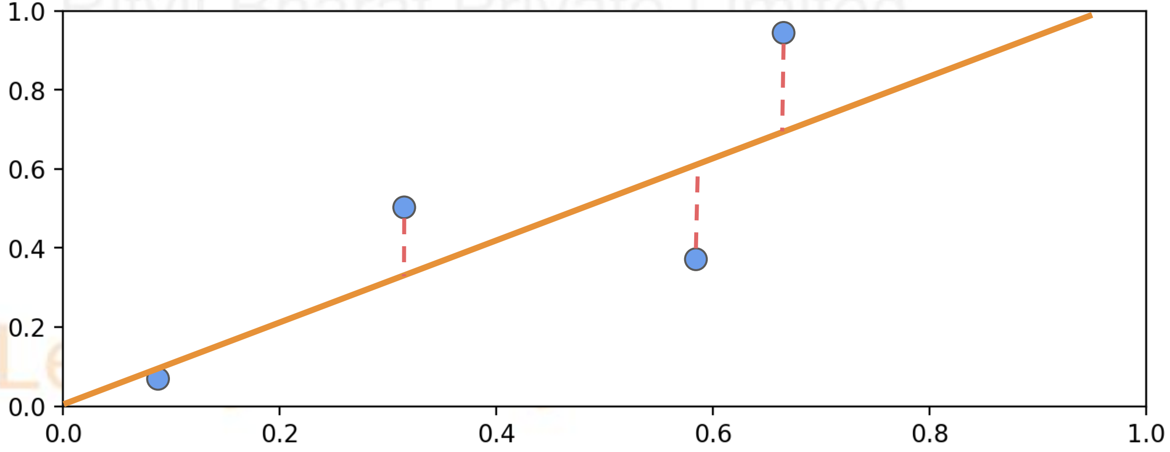

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

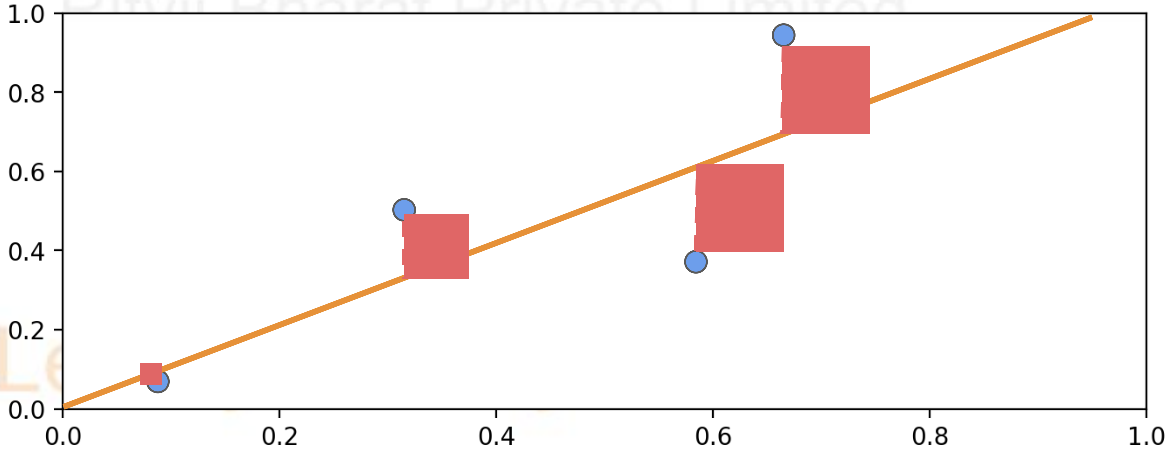

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Ordinary Least Squares (OLS)

Ordinary Least Squares (OLS) operates by minimizing the total sum of the squared deviations between the observed values of the dependent variable in the provided dataset and the values predicted by the linear function.


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

We can visualize squared error to minimize:



Let’s continue exploring OLS by converting a real data set into mathematical notation, then working to solve a linear relationship between features and a variable!

Algorithm Theory - Part Two
OLS Equations

- **$y = mx + b$**
  - m is slope
  - b is intercept with y-axis

We can see for **y=mx+b**
there is only room for one possible feature x.

OLS will allow us to directly solve for **m** and **b.**

Linear Regression enables us to establish a connection between multiple features in order to estimate a desired output.

Area m^2	Bedrooms	Bathrooms	Price
200	2	2	Rs.50,00,000
190	1	1	Rs. 40, 50,000
230	3	2	Rs. 60, 50,000
180	2	1	Rs. 40, 00,000
210	3	1	Rs. 50, 50,000

Let’s translate this data into generalized mathematical notation.

X			y
Area m^2	**Bedrooms**	**Bathrooms**	**Price**
200	2	2	Rs.50,00,000
190	1	1	Rs. 40, 50,000
230	3	2	Rs. 60, 50,000
180	2	1	Rs. 40, 00,000
210	3	1	Rs. 50, 50,000

Let's translate this data into generalized mathematical notation.

X			**y**
x_1	x_2	x_3	y
200	3	2	Rs.50,00,000
190	2	1	Rs.40,50,000
230	3	3	Rs.60,50,000
180	1	1	Rs.40,00,000
210	2	2	Rs.50,50,000

X			y
x_1	x_2	x_3	y

Reformat for **y = x** equation

y	**X**		
y	x_1	x_2	x_3

Every feature should possess an associated Beta coefficient.

y		**X**	
y	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

Same as the common notation for a simple line: **y=mx+b**

y	**X**		
y	x_1	x_2	x_3

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

Fully generalized for any number of features.

y	X		
y	x_1	x_2	x_3

$$\hat{y} = \beta_0 x_0 + \cdots + \beta_n x_n$$

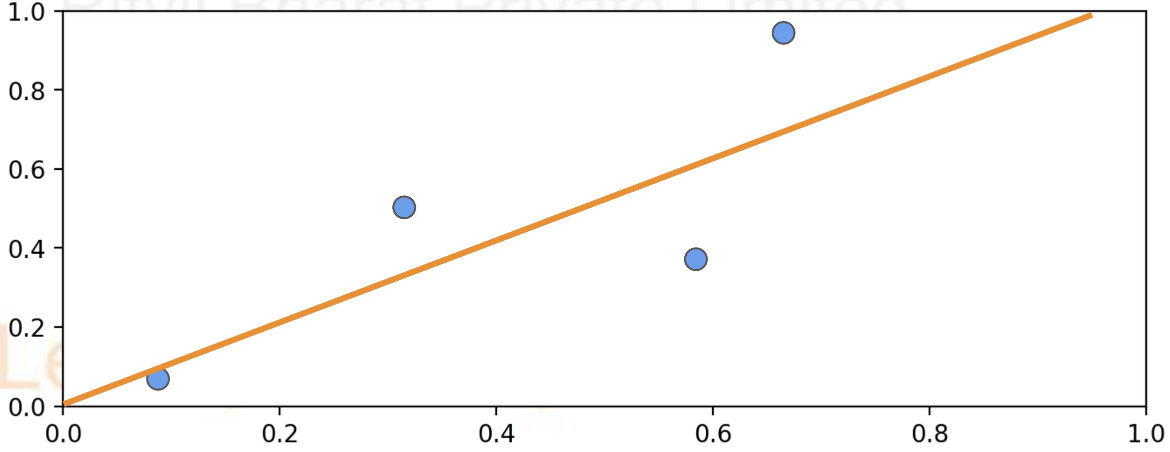
$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

y hat because there is usually no set of Betas to create a perfect fit to y!

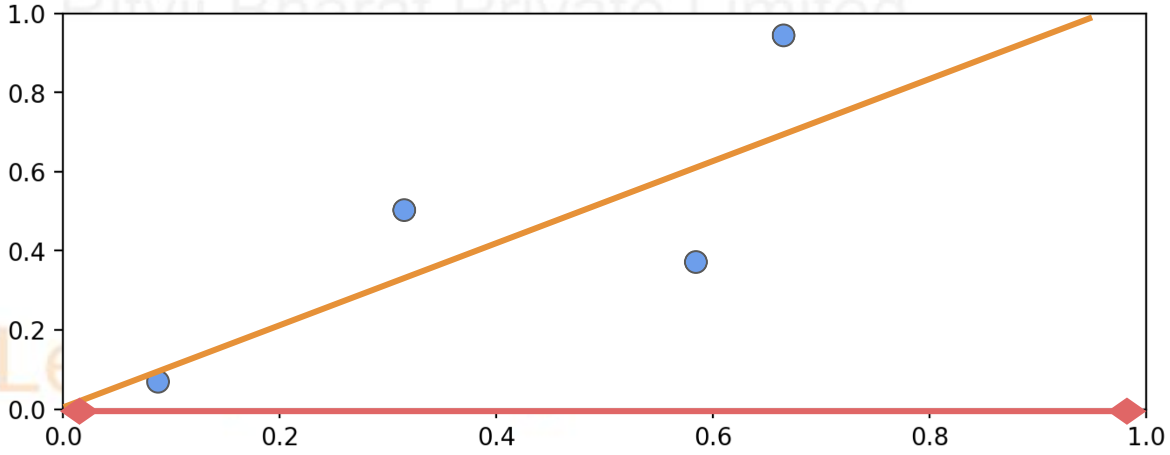
$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$

Line equation:

$$\hat{y} = \sum_{i=0}^{n} \beta_i x_i$$



$$\hat{y} = \sum_{i=0}^{n} \beta_i \boxed{x_i}$$



For simple problems with one X feature we can easily solve for Betas values with an analytical solution.