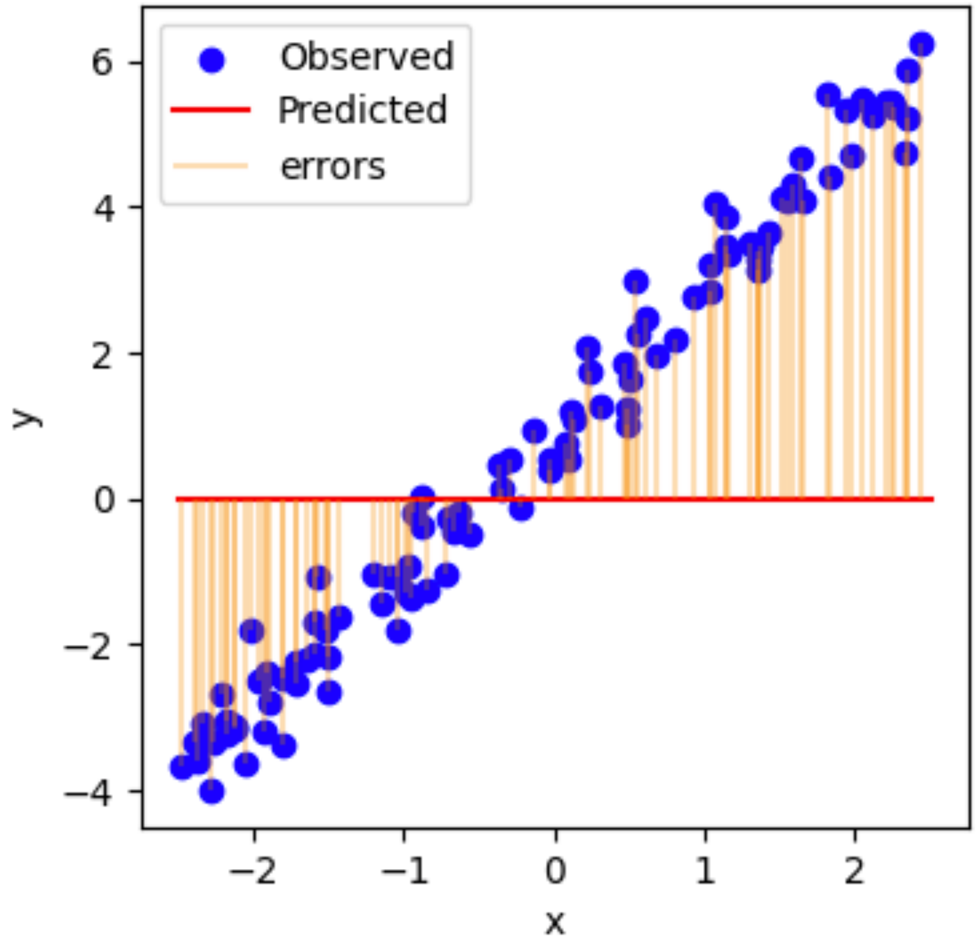
Brain Code Camp

Simple Linear Models

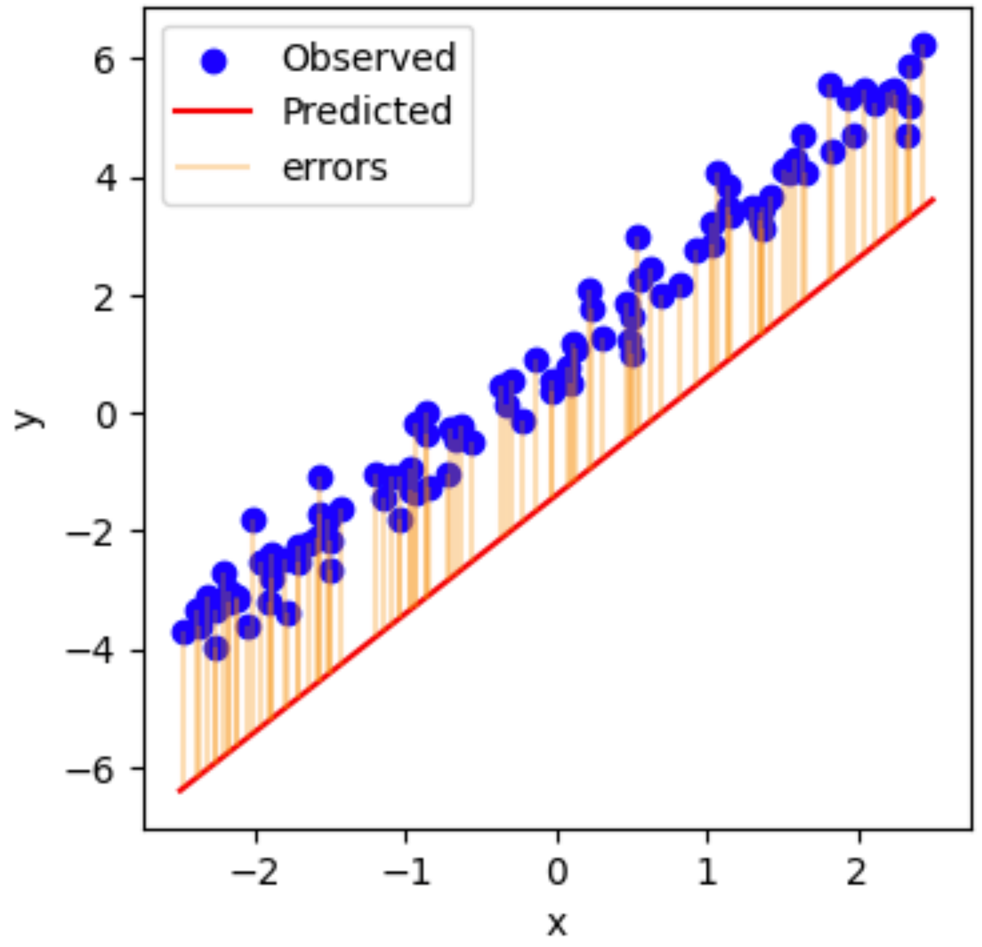
Itthi Chatnuntawech

Manual Parameter Tuning

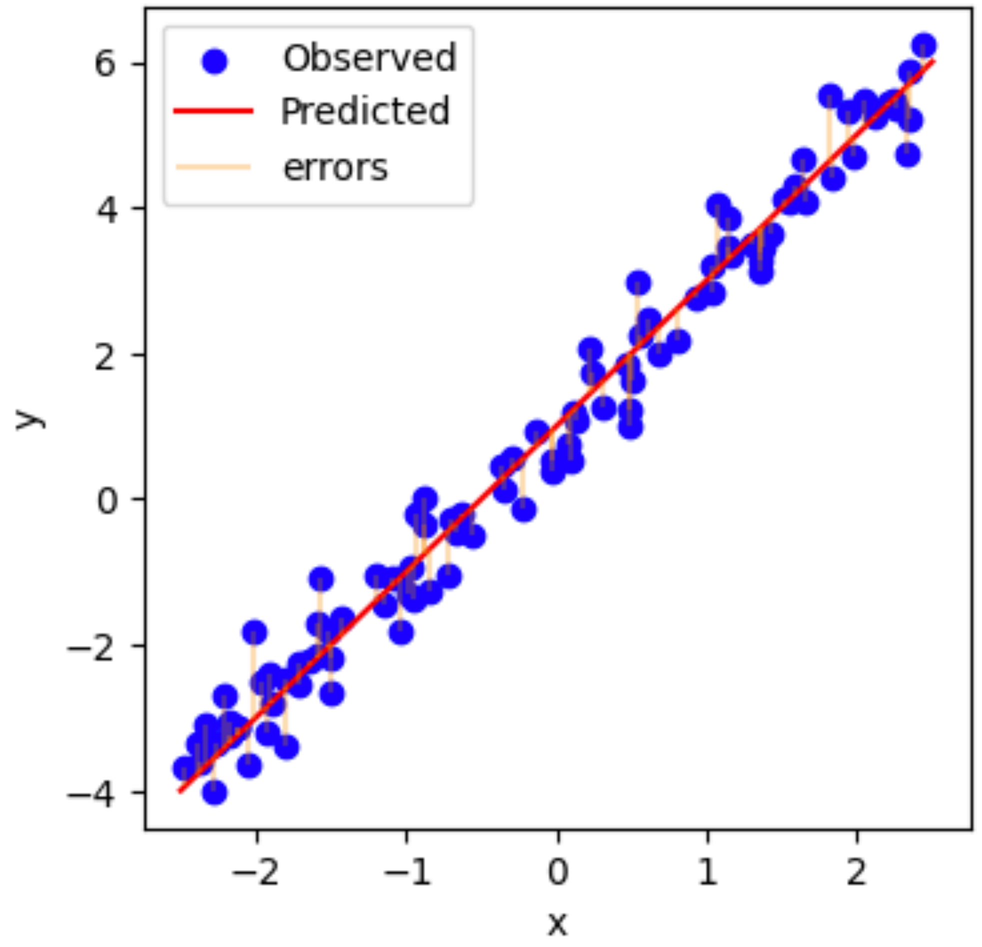
$$\hat{y} = \hat{w}_0 + \hat{w}_1 x$$



$$\hat{y} = 0$$



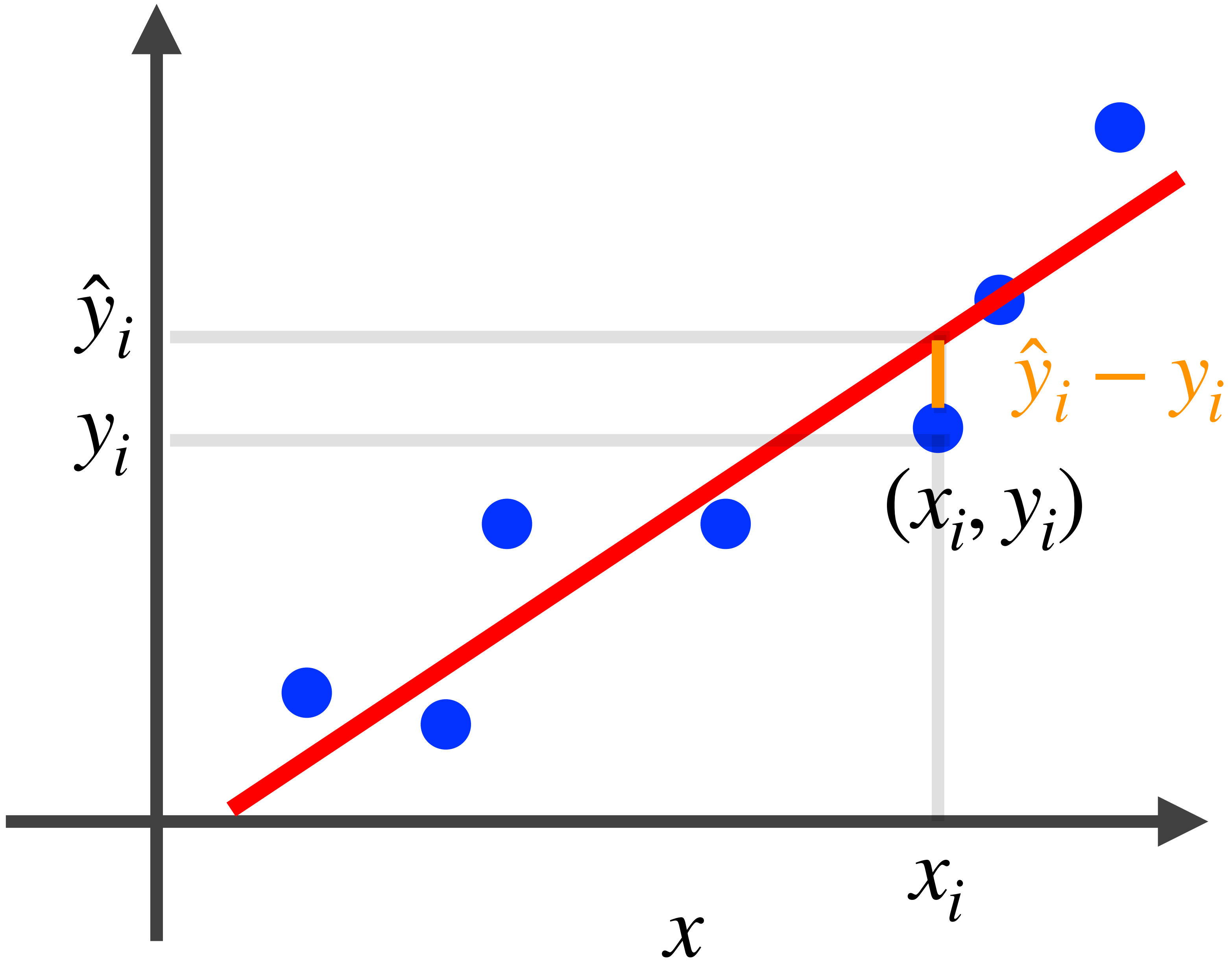
$$\hat{y} = -1.40 + 2x$$



$$\hat{y} = 1 + 2x$$

Pick the parameters that make the orange lines as short as possible

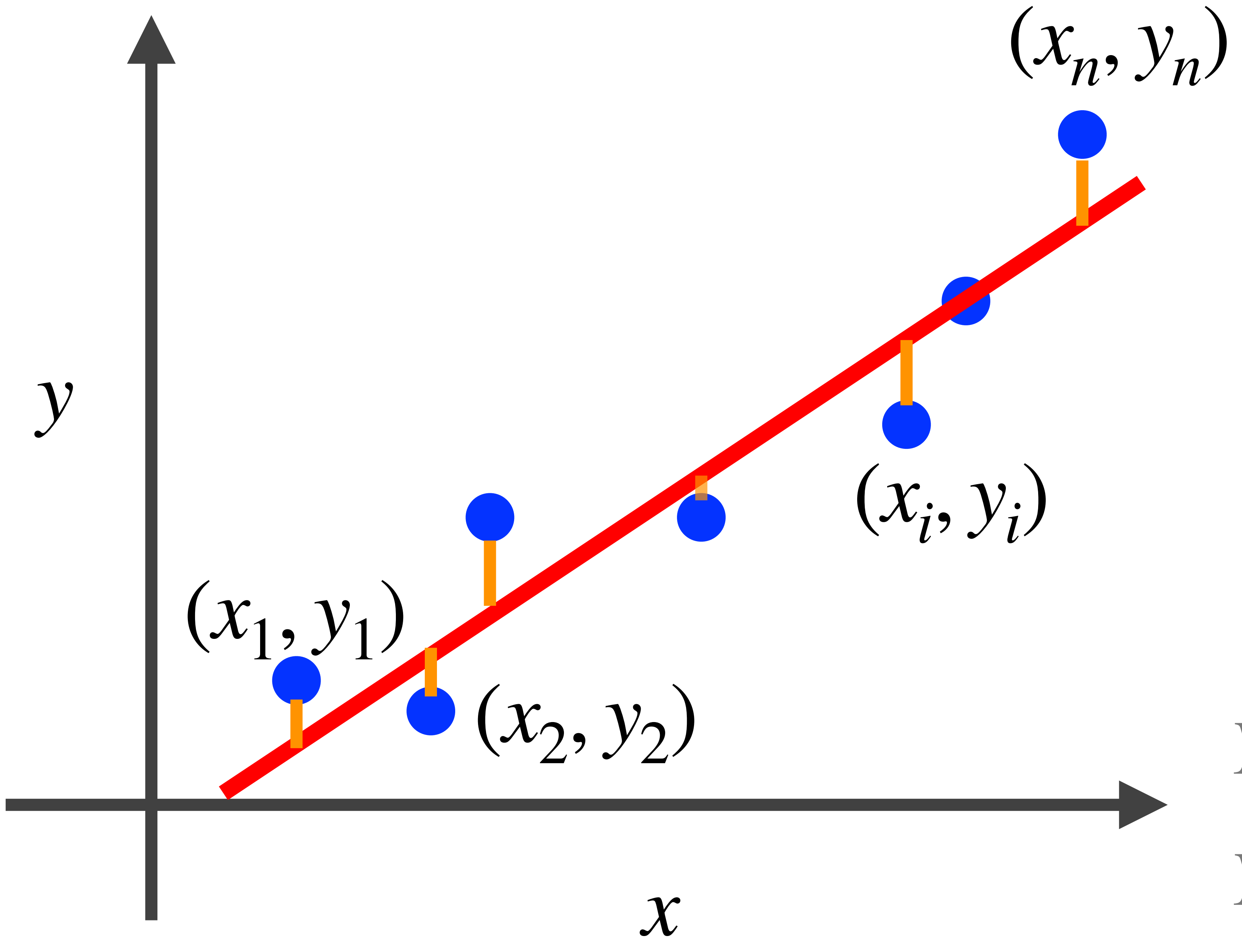
Mean-Squared Error (MSE)



Compute the error from sample i using the following loss function

$$L(y_i, \hat{y}_i) = \left(y_i - \hat{y}_i\right)^2$$

Mean-Squared Error (MSE)



Compute the error from sample i using the following loss function

$$L(y_i, \hat{y}_i) = \left(y_i - \hat{y}_i\right)^2$$

Combine the errors from all samples

$$MSE(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

$$Y : y_1, y_2, \ldots, y_i, \ldots, y_n$$

$$\hat{Y} : \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_i, \ldots, \hat{y}_n$$

$$= \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2$$

Linear Regression

$$MSE(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\hat{w}_0 + \hat{w}_1 x_i)\right)^2$$

Find $\hat{w}_0$ and $\hat{w}_1$ that minimize $MSE(Y, \hat{Y})$

$$\min_{\hat{w}_0, \hat{w}_1} MSE(Y, \hat{Y}) = \min_{\hat{w}_0, \hat{w}_1} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\hat{w}_0 + \hat{w}_1 x_i)\right)^2$$

Optional: Linear Regression

$$\min_{\hat{w}_0,\hat{w}_1} MSE(Y,\hat{Y}) = \min_{\hat{w}_0,\hat{w}_1} \boxed{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\hat{w}_0 + \hat{w}_1 x_i)\right)^2}$$

Compute the gradients of the loss function with respect to $\hat{w}_0$ and $\hat{w}_1$, and set them to 0

"Taking the partial derivatives and set them to 0"

$$\frac{\partial}{\partial \hat{w}_0}\boxed{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\hat{w}_0 + \hat{w}_1 x_i)\right)^2} = 0$$

$$\frac{\partial}{\partial \hat{w}_1}\boxed{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\hat{w}_0 + \hat{w}_1 x_i)\right)^2} = 0$$

Brain Code Camp

Optional: Linear Regression

Optimal solutions

$$\hat{w}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

and

$$\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

and

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

numpy.mean

for loop numpy.sum

Linear Regression

sklearn.linear_model.LinearRegression

class `sklearn.linear_model.LinearRegression`(**, fit_intercept=True, copy_X=True, n_jobs=None, positive=False*)

[source]

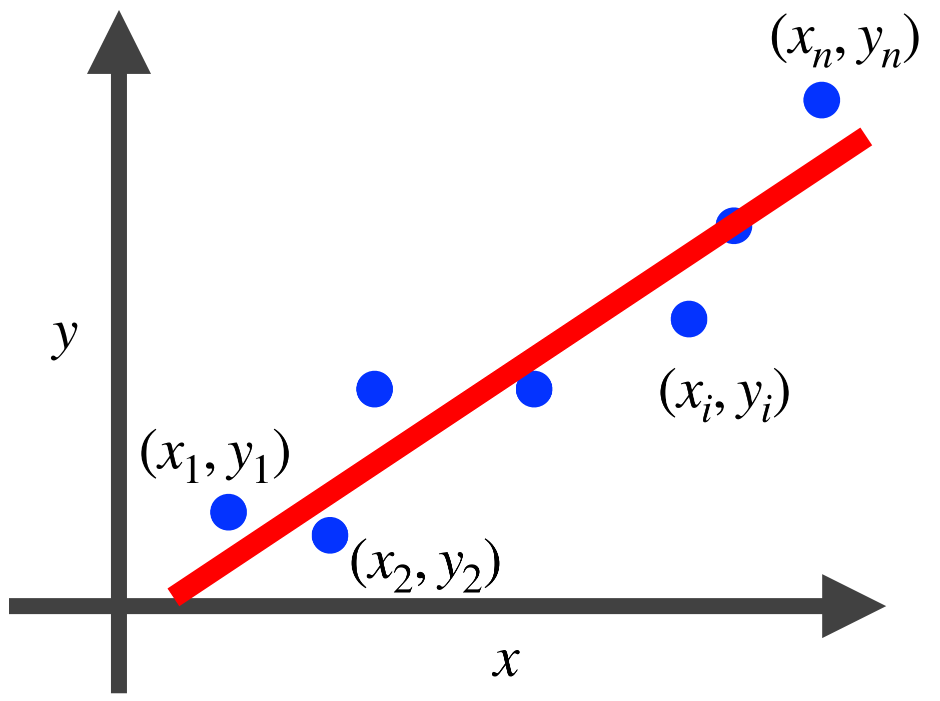
Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Linear Regression

Data

$$(x_1, y_1), (x_2, y_2), \ldots, (x_{100}, y_{100})$$



x $\quad$ **y**

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{100} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{100} \end{bmatrix}$$

shape = (100, 1) $\quad$ shape = (100, 1)

```
# Import a necessary module
from sklearn.linear_model import LinearRegression

# Create the model
model_linear = LinearRegression()

# Train the model
model_linear.fit(x, y)

# Make prediction
y_hat = model_linear.predict(x)
```