Linear Regression

- Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y).
- y can be calculated from a linear combination of the input variables (x).
- When there is a single input variable (x), the method is referred to as simple linear regression.
- When there are multiple input variables - multiple linear regression.
- With simple linear regression we want to model our data as follows:
- $y = B0 + B1 \ x$
- *y is a dependent variable. x is an independent variable*.

Simple Linear Regression

- This is a line where y is the output variable we want to predict, x is the input variable
- B0 and B1 are coefficients that we need to estimate that move the line around.
- Technically, B0 is called the intercept because it determines where the line intercepts the y-axis.
- In machine learning we can call this the bias, because it is added to offset all predictions that we make.
- The B1 term is called the slope because it defines the slope of the line or how x translates into a y value before we add our bias

Simple Linear Regression

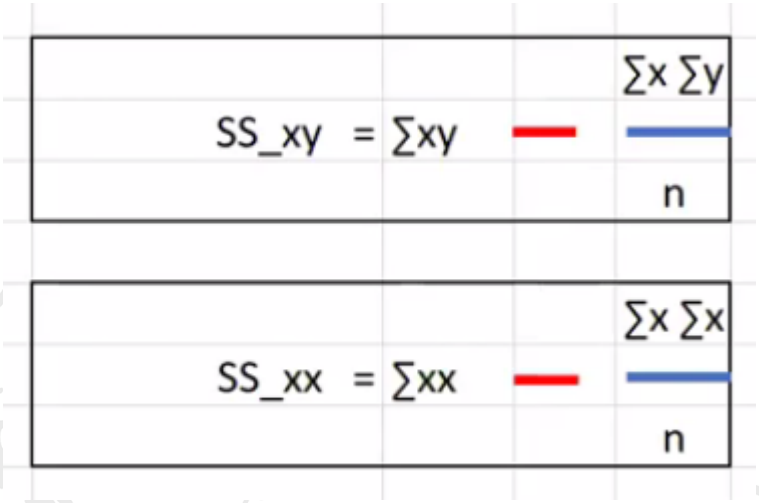
- The goal is to find the best estimates for the coefficients to minimize the errors in predicting y from x.
- B1 can be estimated as:

$$B1 = \frac{\sum_{i=1}^{n}(x_i - mean(x)) \times (y_i - mean(y))}{\sum_{i=1}^{n}(x_i - mean(x))^2}$$

- Where mean() is the average value for the variable in our dataset. The xi and yi refer to the fact that we need to repeat these calculations across all values in our dataset and i refers to the i'th value of x or y
- B0 can be calculated as

$$B0 = mean(y) - B1 \times mean(x)$$

Simple Linear Regression



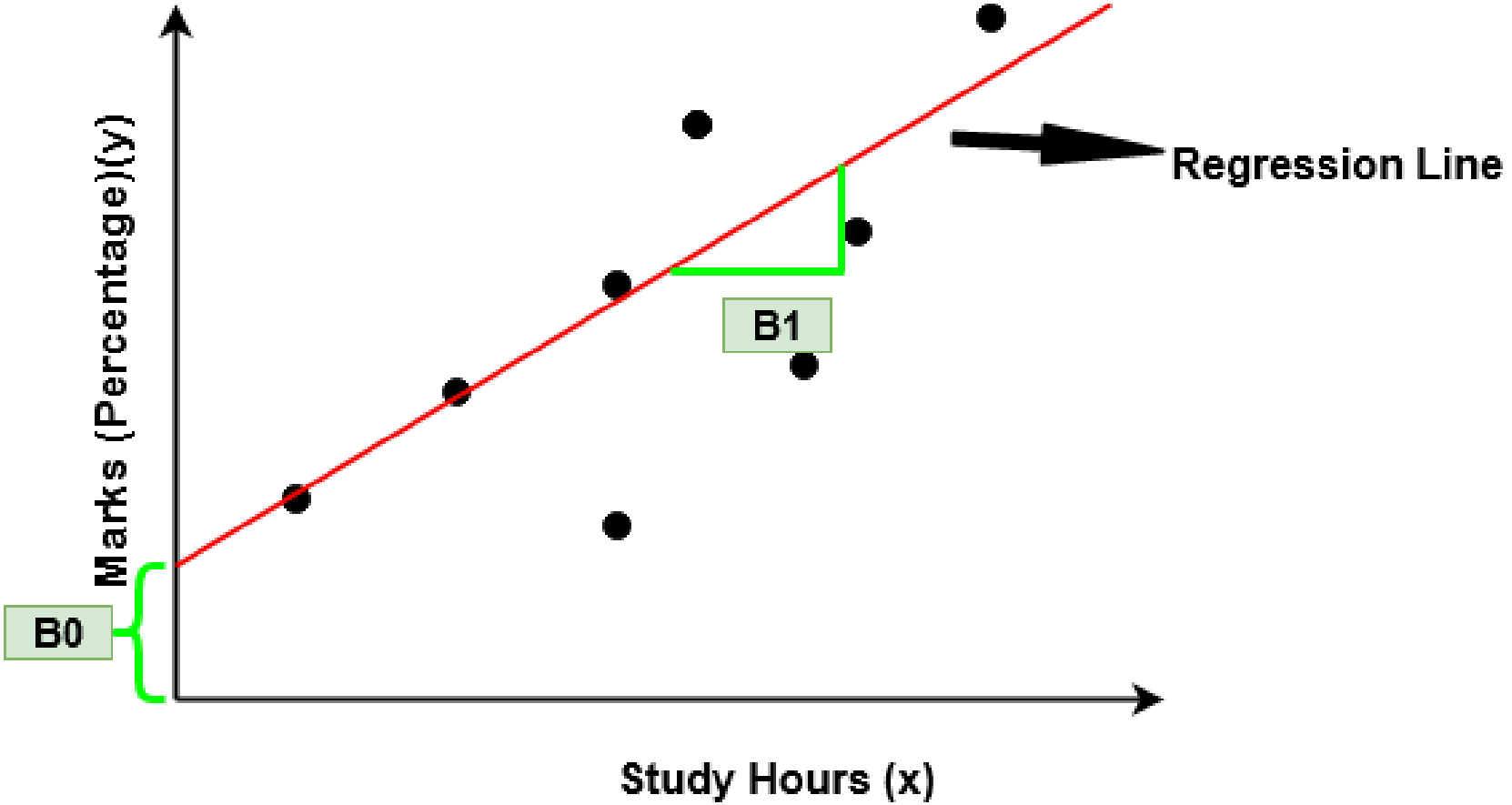
Regression Coefficients

B1 or M = SS_xy / SS_xx

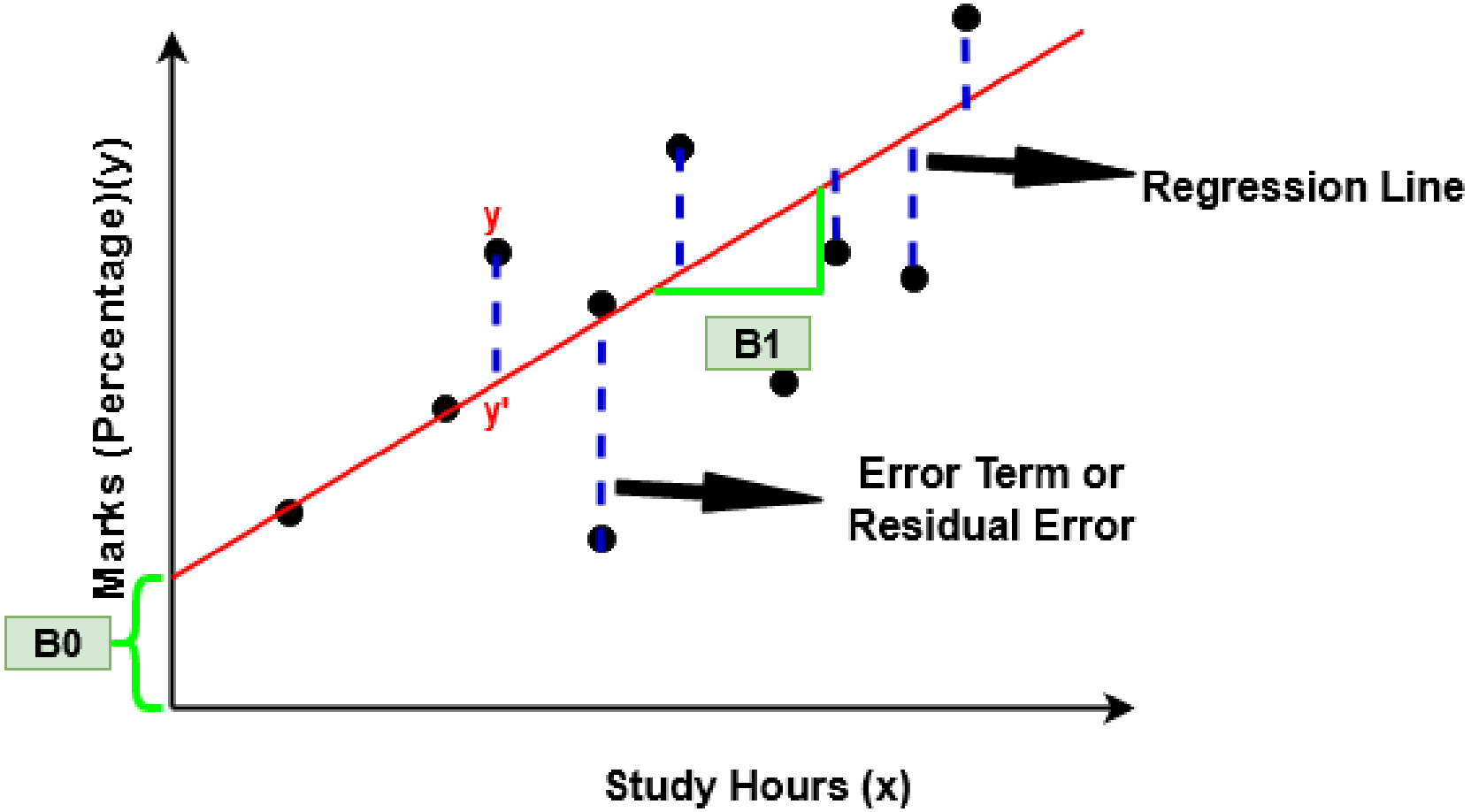
B0 or C = mean(y) - m * mean(x)

C=y-Mx

Simple Linear Regression



Simple Linear Regression

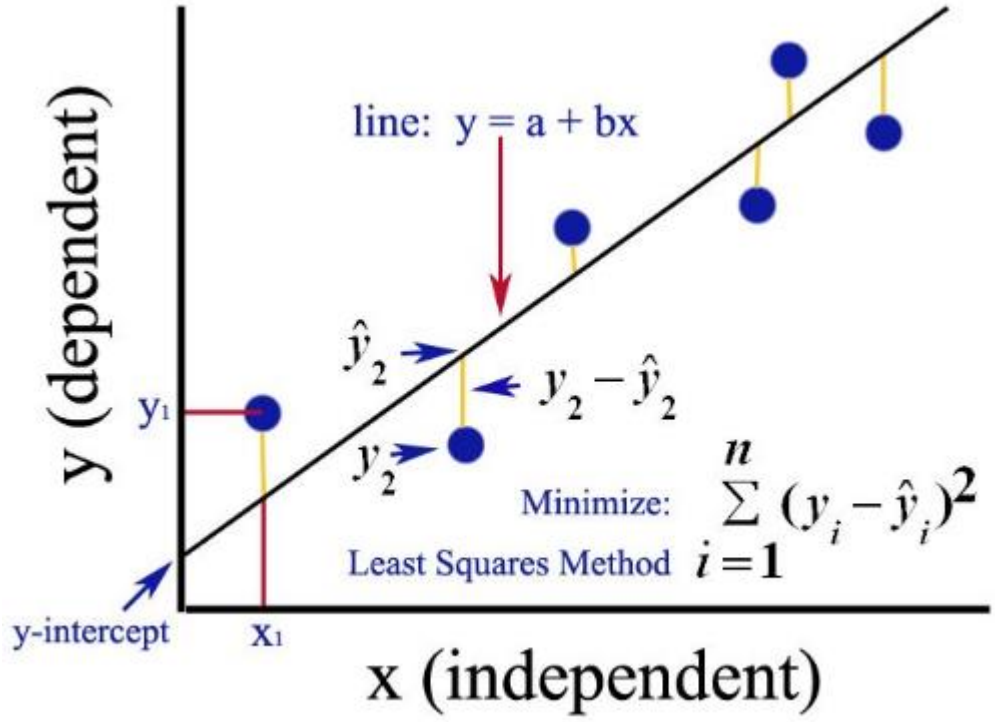


Find slope and intercept given measurements xi,yi, i=1..n

that minimizes the sum of the squares of the residuals

Simple Linear Regression

- Ordinary Least Squares (OLS)



Find slope and intercept given measurements xi,yi, i=1..n

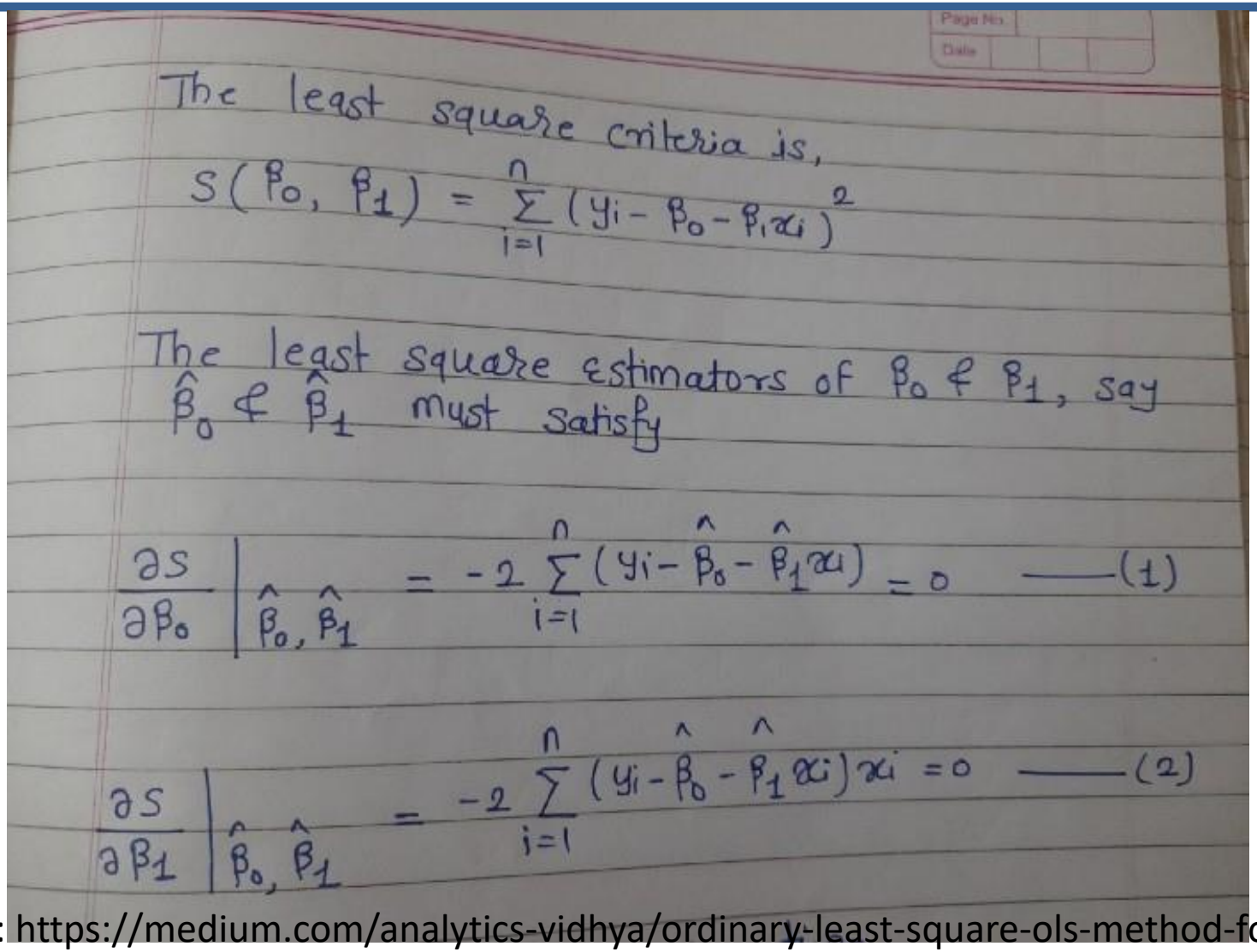
that minimizes the sum of the squares of the residuals

Source: https://medium.com/analytics-vidhya/ordinary-least-square-ols-method-for-linear-regression-ef8ca10aadfc

Ordinary Least Square (OLS)

- The OLS method is used to estimate **B**0 and B1. The OLS method seeks to minimize the sum of the squared residuals. This means from the given data we calculate the distance from each data point to the regression line, square it, and the sum of all of the squared errors together.

Ordinary Least Square (OLS)



The least square criteria is,

$$S(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

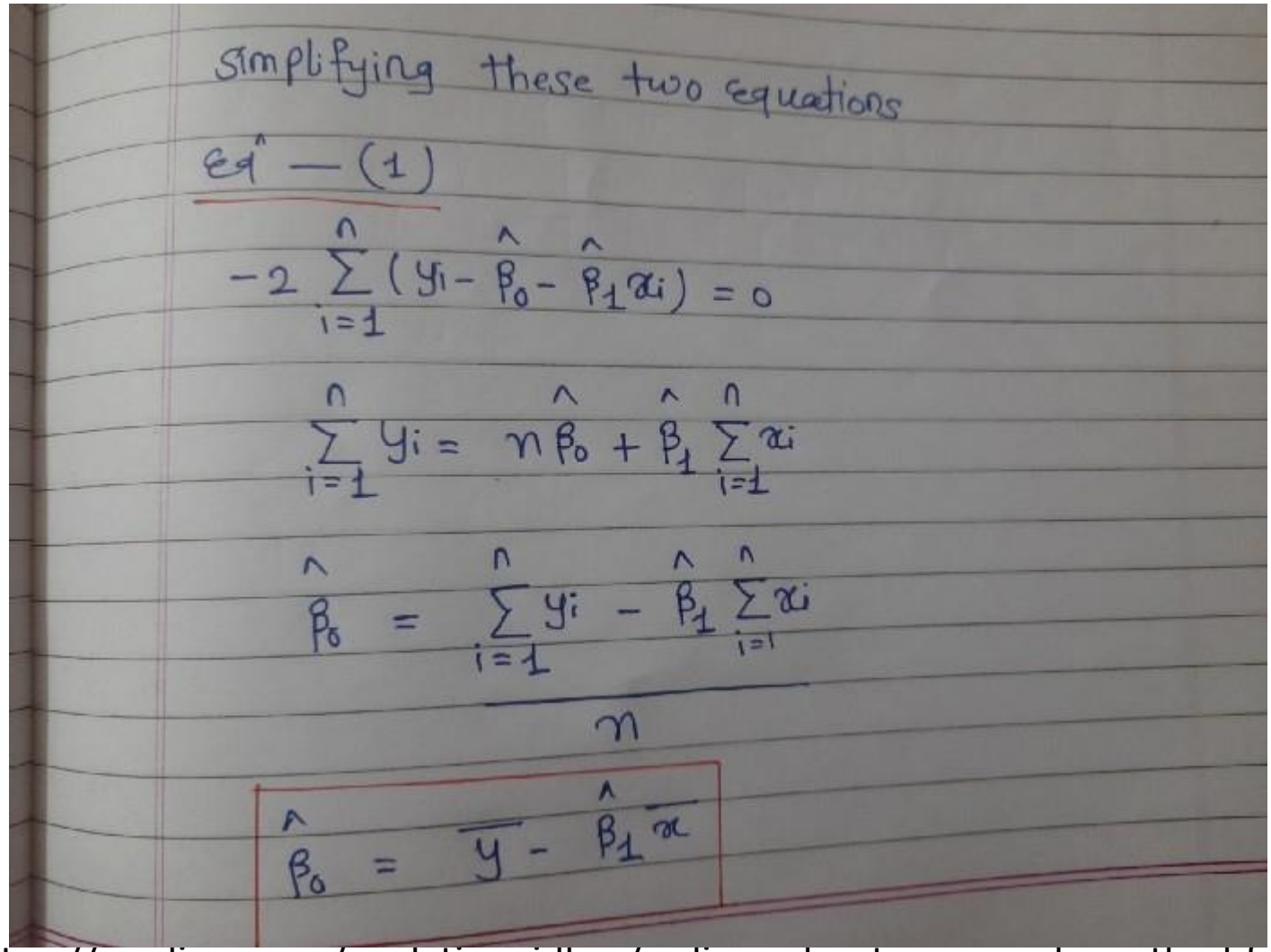
The least square estimators of β_0 & β_1, say $\hat{\beta}_0$ & $\hat{\beta}_1$ must satisfy

$$\left.\frac{\partial S}{\partial \beta_0}\right|_{\hat{\beta}_0, \hat{\beta}_1} = -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \quad \text{——(1)}$$

$$\left.\frac{\partial S}{\partial \beta_1}\right|_{\hat{\beta}_0, \hat{\beta}_1} = -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) x_i = 0 \quad \text{——(2)}$$

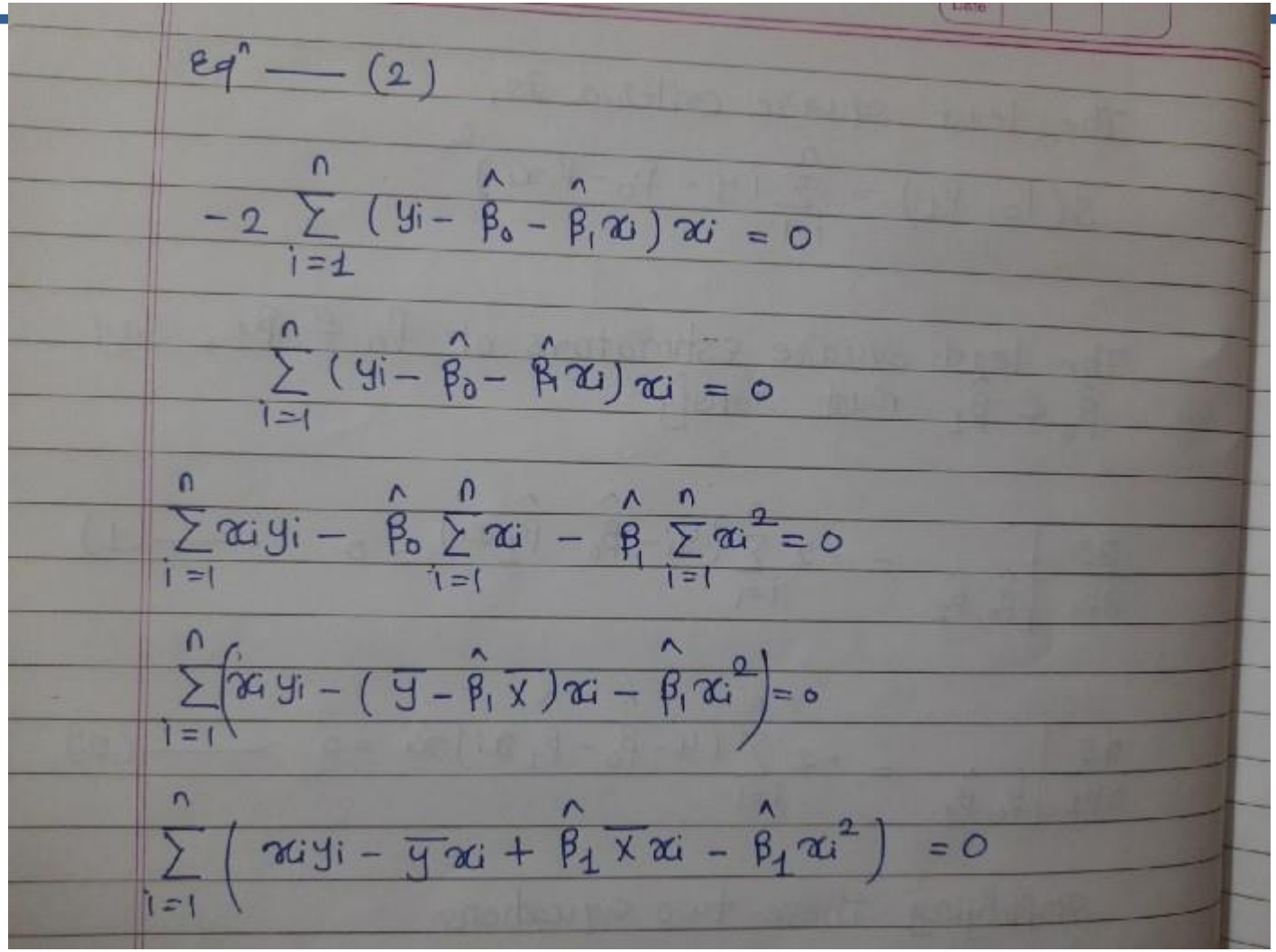
Source: https://medium.com/analytics-vidhya/ordinary-least-square-ols-method-for-linear-regression-ef8ca10aadfc

Ordinary Least Square (OLS)



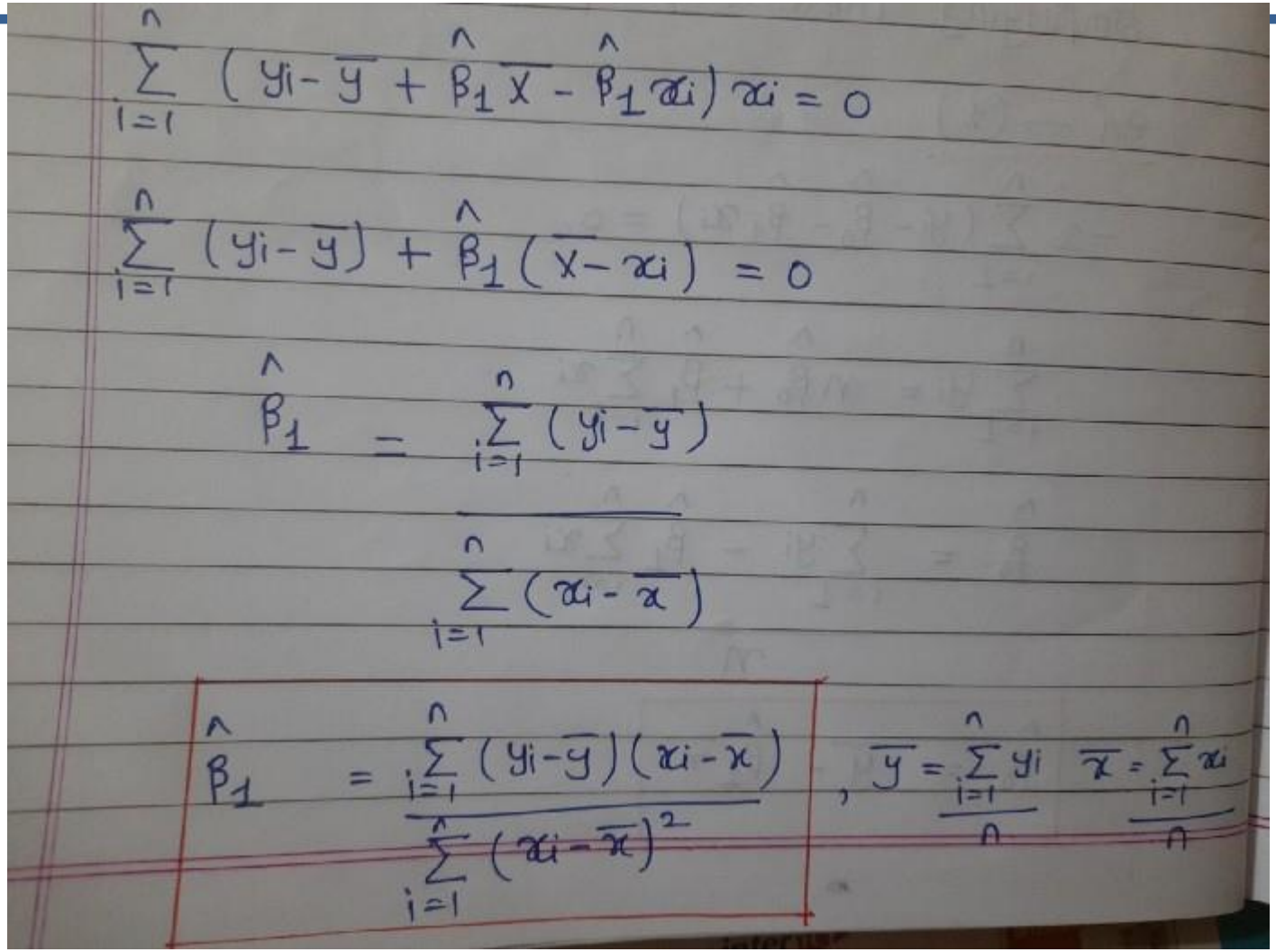
Source: https://medium.com/analytics-vidhya/ordinary-least-square-ols-method-for-linear-regression-ef8ca10aadfc

Ordinary Least Square (OLS)



Source: https://medium.com/analytics-vidhya/ordinary-least-square-ols-method-for-linear-regression-ef8ca10aadfc

Ordinary Least Square (OLS)



Source: https://medium.com/analytics-vidhya/ordinary-least-square-ols-method-for-linear-regression-ef8ca10aadfc

Root Mean Square Error

- Known as RMSE

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - y_i)^2}{n}}$$

- p is the predicted value and y is the actual value, i is the index for a specific instance, because we must calculate the error across all predicted values.

Preparing Data For Linear Regression

- **Linear Assumption**. Linear regression assumes that the relationship between your input and output is linear. It does not support anything else. This may be obvious, but it is good to remember when you have a lot of attributes. You may need to transform data to make the relationship linear (e.g. log transform for an exponential relationship).
- **Remove Noise**. Linear regression assumes that your input and output variables are not noisy. Consider using data cleaning operations that let you better expose and clarify the signal in your data. This is most important for the output variable and you want to remove outliers in the output variable (y) if possible.

Preparing Data For Linear Regression

- **Remove Collinearity.** Linear regression will overfit your data when you have highly correlated input variables. Consider calculating pairwise correlations for your input data and removing the most correlated.
- **Gaussian Distributions.** Linear regression will make more reliable predictions if your input and output variables have a Gaussian distribution. You may get some benefit using transforms (e.g. log or BoxCox) on you variables to make their distribution more Gaussian looking.
- **Rescale Inputs:** Linear regression will often make more reliable predictions if you rescale input variables using standardization or normalization.

Multiple Linear Regression

- When there are multiple input variables - multiple linear regression.
- With multiple linear regression we want to model our data as follows:
- y = B0 + B1 * x1 + B2 * x2 + B3 * x3……..Bn *xn
- *y is a dependent variable. xi are independent variable*.
- B0 is a constant and B1 …Bn are coefficients that we need to estimate
- Example: House Price -> Area, location, no of bedrooms etc

Multiple Linear Regression –Data Set

R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	**192261.8**
162597.7	151377.6	443898.5	California	**191792.1**
153441.5	101145.6	407934.5	Florida	**191050.4**
144372.4	118671.9	383199.6	New York	**182902**
142107.3	91391.77	366168.4	Florida	**166187.9**
131876.9	99814.71	362861.4	New York	**156991.1**
134615.5	147198.9	127716.8	California	**156122.5**
130298.1	145530.1	323876.7	Florida	**155752.6**
120542.5	148719	311613.3	New York	**152211.8**
123334.9	108679.2	304981.6	California	**149760**
101913.1	110594.1	229161	Florida	**146122**

Multiple Linear Regression – Dummy Variables

R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	**192261.8**
162597.7	151377.6	443898.5	California	**191792.1**
153441.5	101145.6	407934.5	Florida	**191050.4**
144372.4	118671.9	383199.6	New York	**182902**
142107.3	91391.77	366168.4	Florida	**166187.9**
131876.9	99814.71	362861.4	New York	**156991.1**
134615.5	147198.9	127716.8	California	**156122.5**
130298.1	145530.1	323876.7	Florida	**155752.6**
120542.5	148719	311613.3	New York	**152211.8**
123334.9	108679.2	304981.6	California	**149760**
101913.1	110594.1	229161	Florida	**146122**

Multiple Linear Regression – Dummy Variables

R&D Spend	Administration	Marketing Spend	New York	California	Florida
165349.2	136897.8	471784.1	1	**0**	**0**
162597.7	151377.6	443898.5	0	**1**	**0**
153441.5	101145.6	407934.5	0	**0**	**1**
144372.4	118671.9	383199.6	1	**0**	**0**
142107.3	91391.77	366168.4	0	**0**	**1**
131876.9	99814.71	362861.4	1	**0**	**0**
134615.5	147198.9	127716.8	0	**1**	**0**
130298.1	145530.1	323876.7	0	**0**	**1**
120542.5	148719	311613.3	1	**0**	**0**
123334.9	108679.2	304981.6	0	**1**	**0**
101913.1	110594.1	229161	0	**0**	**1**

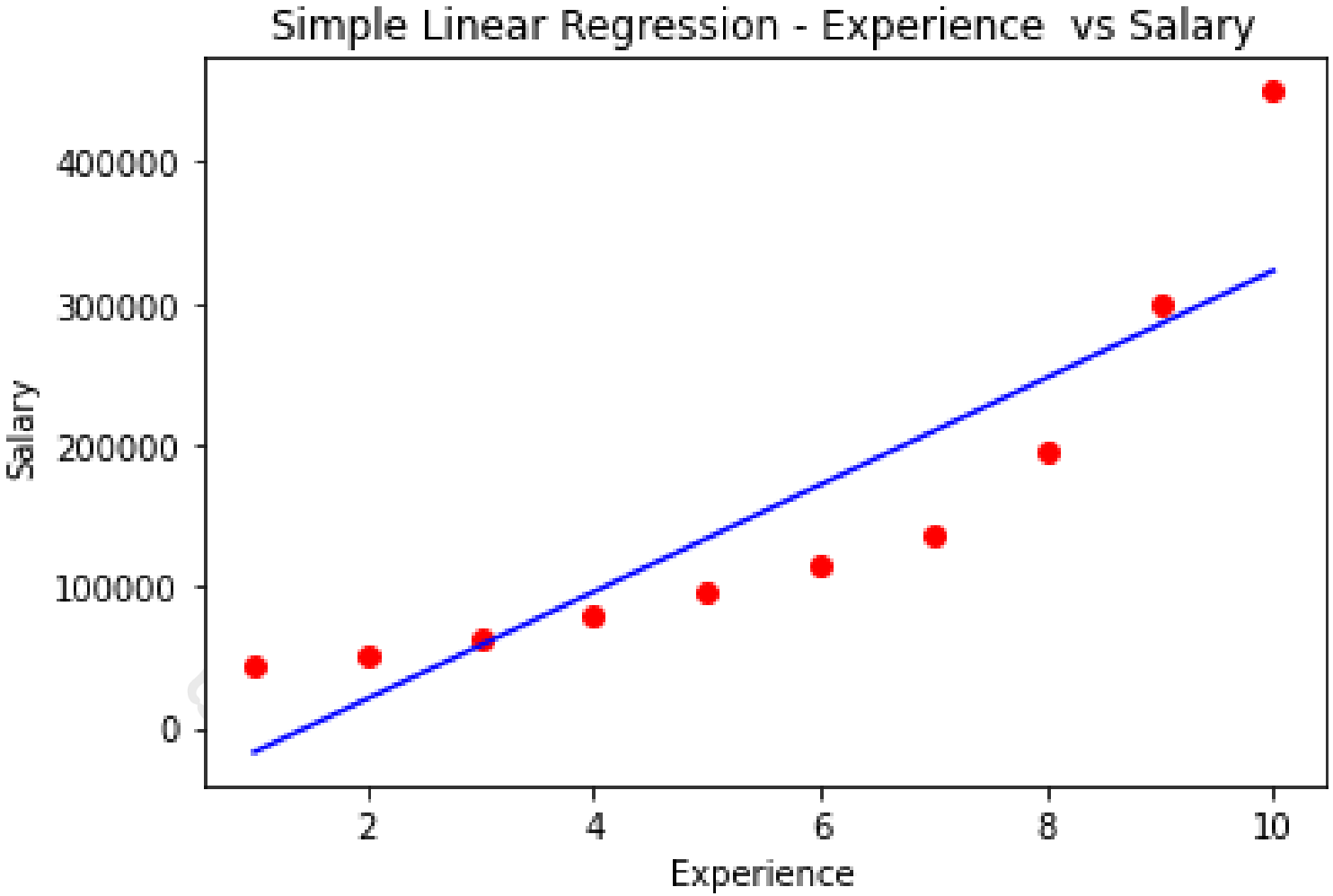
Polynomial Regression

- polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial in x.
- Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y
- With Polynomal regression we want to model our data as follows:
- $y = B0 + B1 * x + B2 * x^2 + B3 * x^3 ……..Bn * x^n$
- *y is a dependent variable. xi are independent variable*.
- B0 is a constant and B1 …Bn are coefficients that we need to estimate
- The number of higher-order terms increases with the increasing value of ***n***, and hence the equation becomes more complicated.

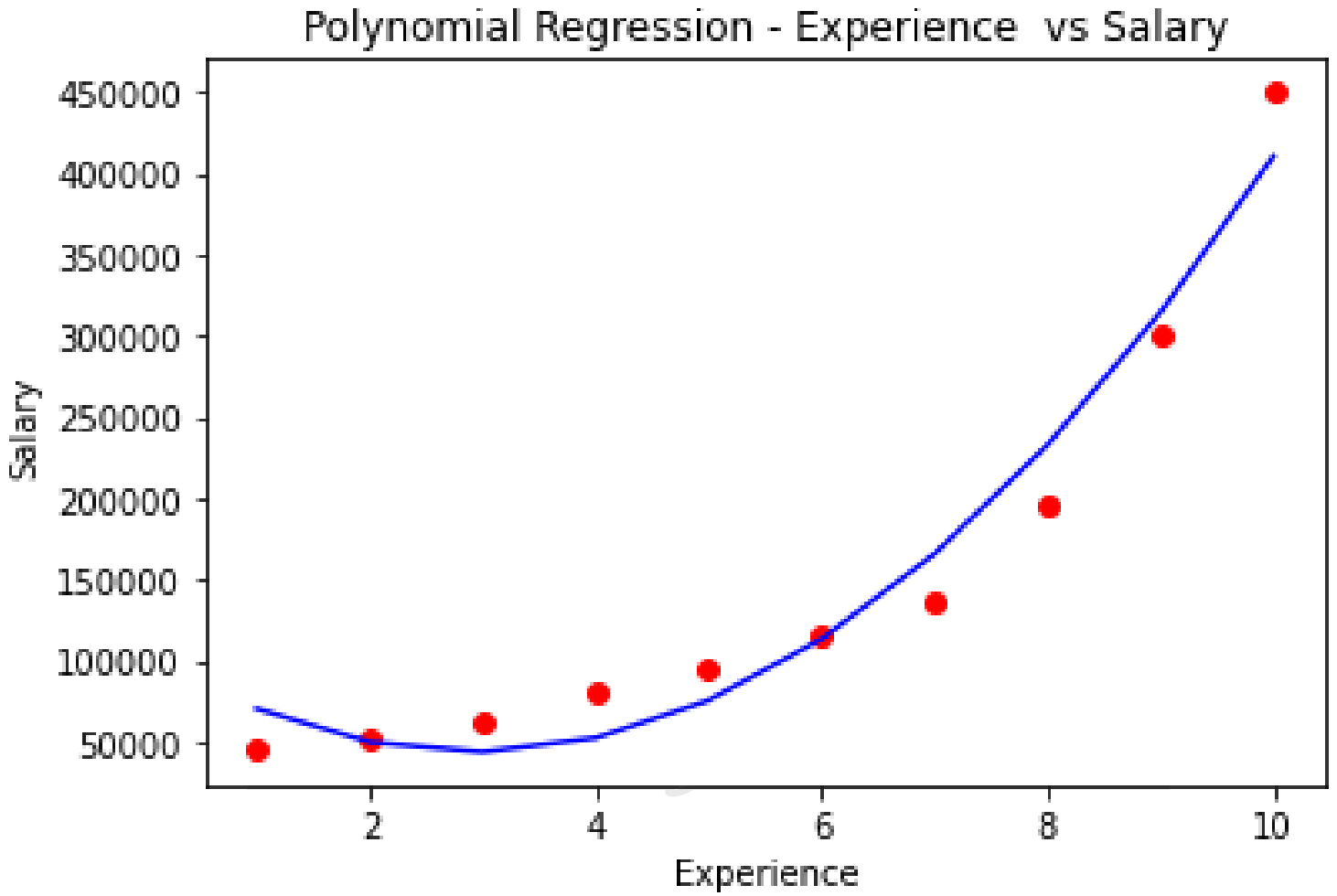
Polynomial Regression

- **Advantages of using Polynomial Regression:**
- Polynomial provides the best approximation of the relationship between the dependent and independent variable.
- A Broad range of function can be fit under it.
- Polynomial basically fits a wide range of curvature.
- **Disadvantages of using Polynomial Regression**
- The presence of one or two outliers in the data can seriously affect the results of the nonlinear analysis.
- These are too sensitive to the outliers.
- In addition, there are unfortunately fewer model validation tools for the detection of outliers in nonlinear regression than there are for linear regression.

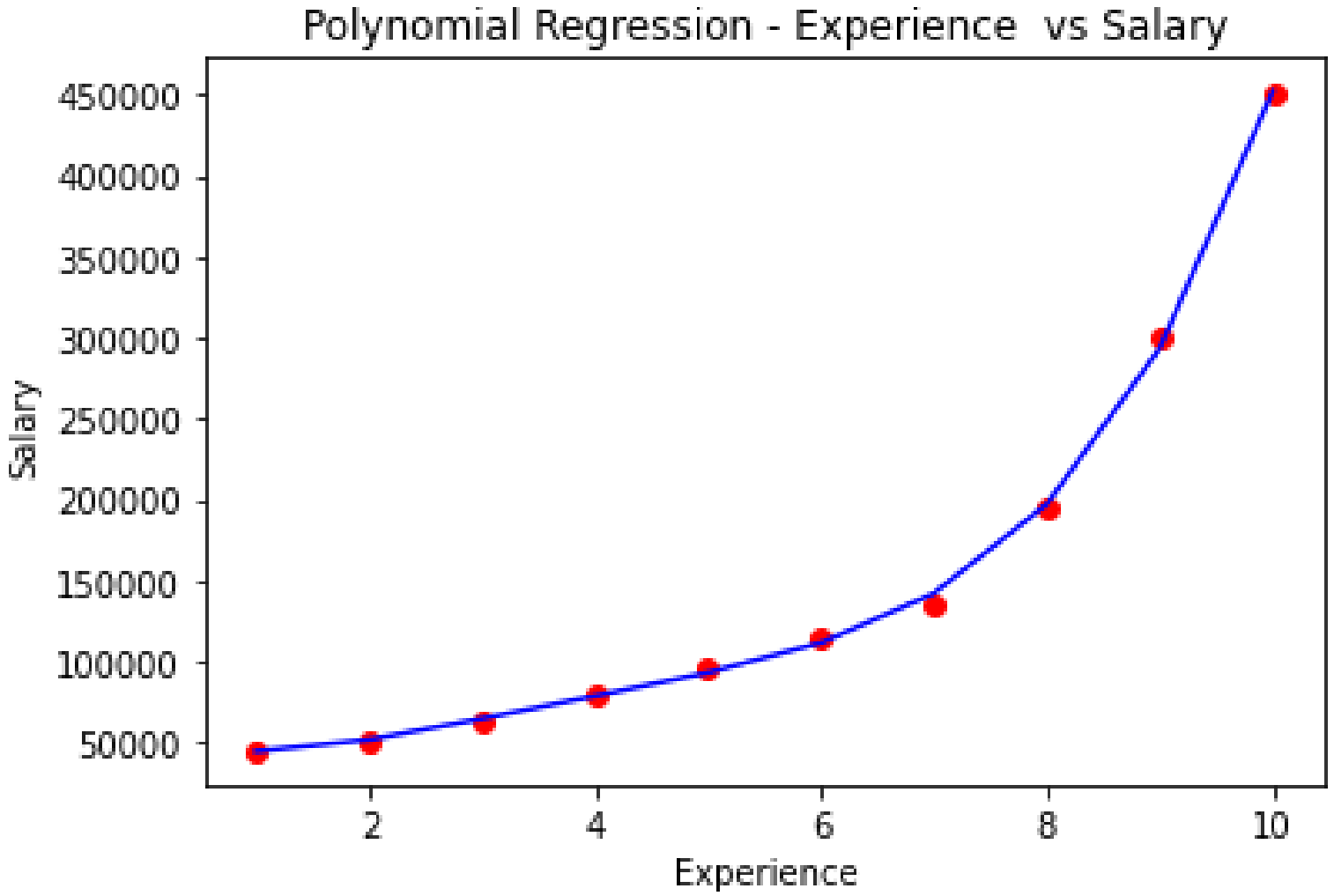
Polynomial Regression



Polynomial Regression



Polynomial Regression

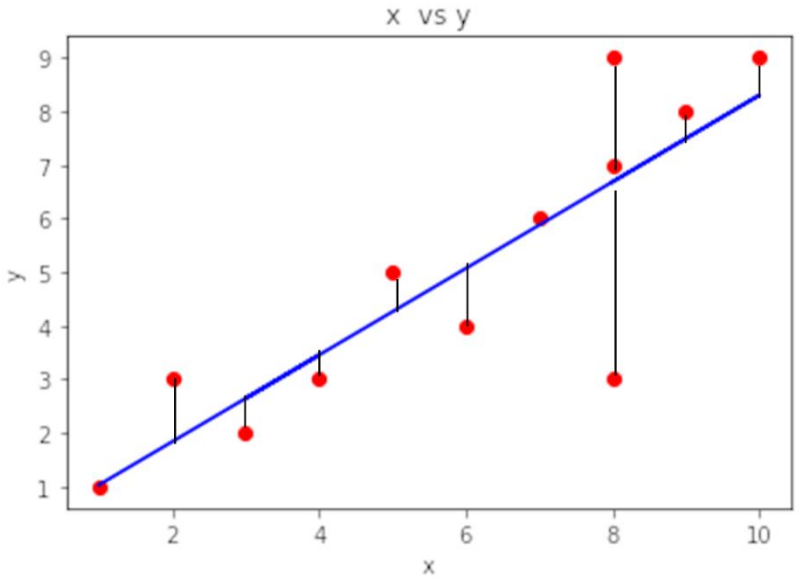


Support Vector Regression (SVR)

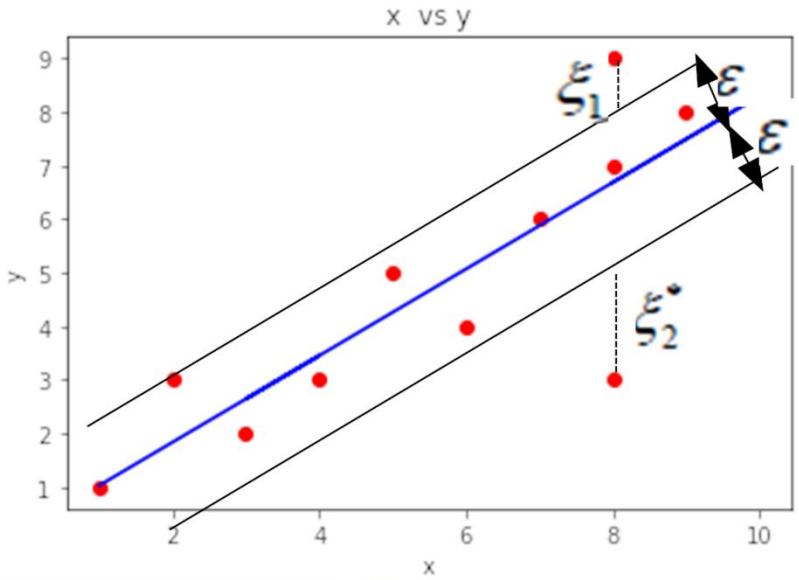
- No ML method is superior to any other..
- Each method is different
  - Type of problem
  - Prior distribution
- SVR gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data
- In contrast to OLS, the objective function of SVR is to minimize the coefficients — more specifically, the *l*2-norm of the coefficient vector — not the squared error.
- The error term is instead handled in the constraints, where we set the absolute error less than or equal to a specified margin, called the maximum error, ϵ (epsilon). We can tune epsilon to gain the desired accuracy of our model.

Support Vector Regression (SVR)

Linear Regression



Support Vector Regression



Ordinary Least Squares

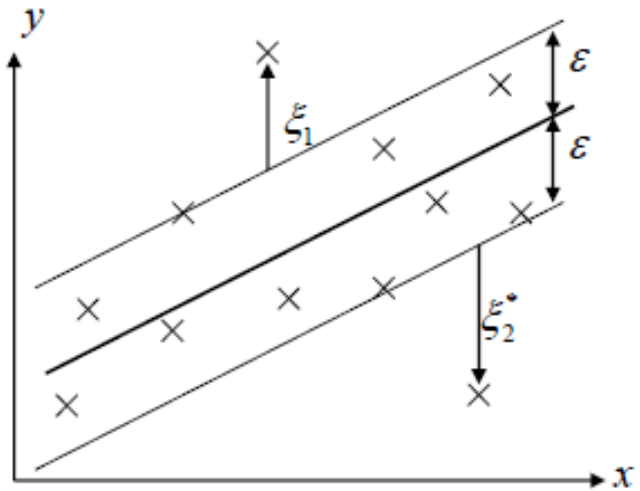
minmize<-sum(y-y')^2

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \rightarrow min$$

Support Vector Regression (SVR)

Given training data

$$(\mathbf{x}_i, y_i) \quad i = 1, \ldots, m$$



Minimize

$$\frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)$$

Under constraints

$$\begin{cases} y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b \le \varepsilon + \xi_i \\ (\mathbf{w} \cdot \mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, i = 1, \ldots, m \end{cases}$$

Support Vector Regression (SVR)

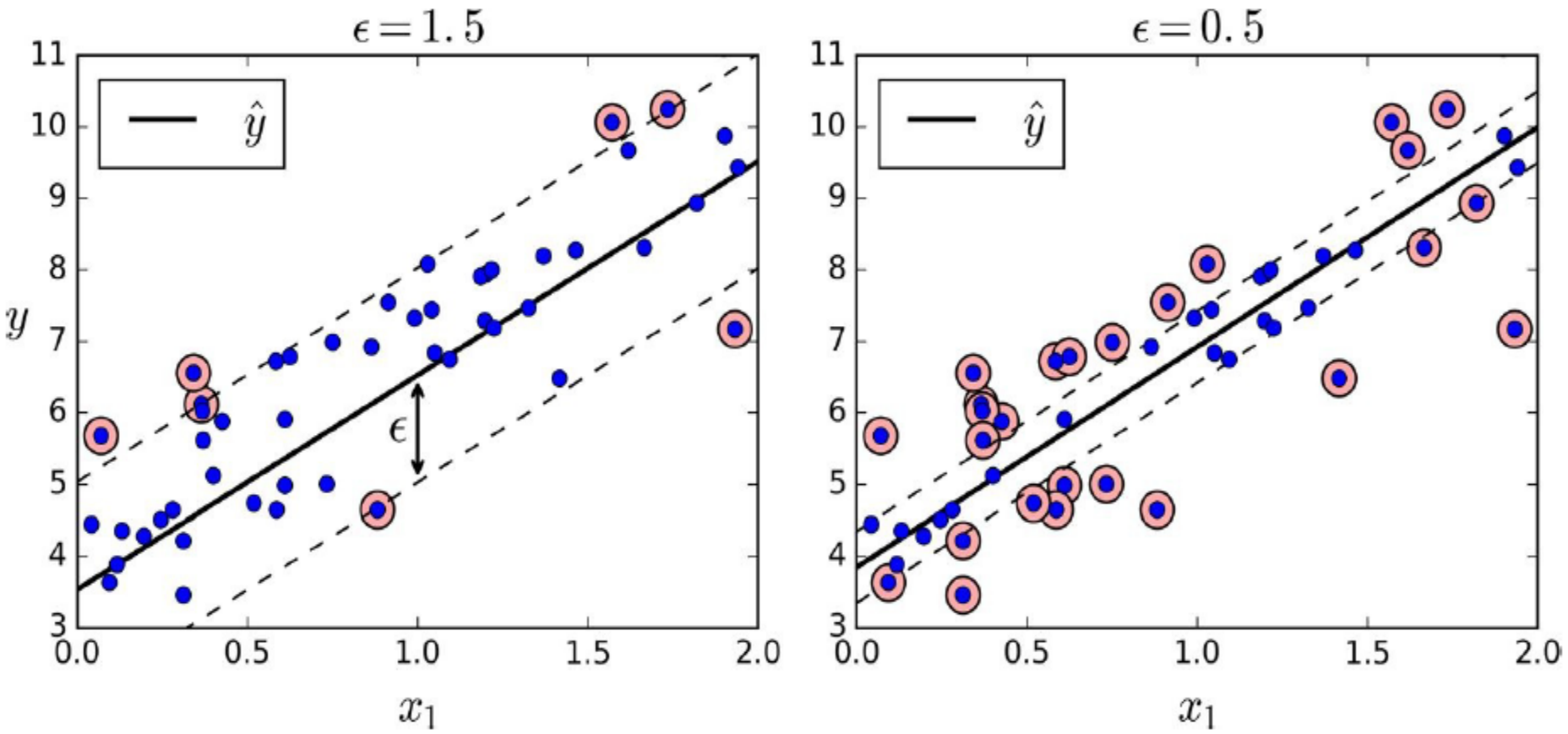


Image Source: Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron

Support Vector Regression (SVR)

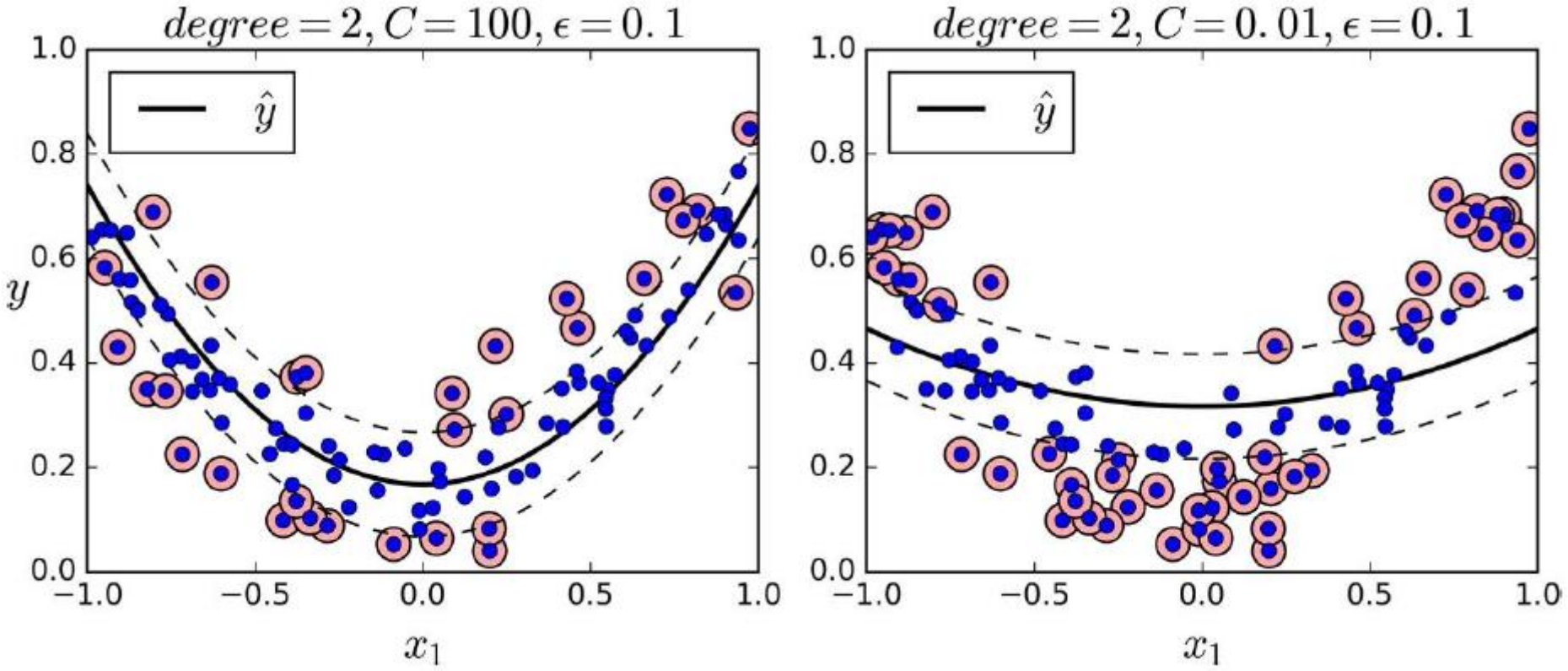


Image Source: Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron