LinkedIn: linkedin.com/in/stanley-sayianka-8a6450170
GitHub: github.com/stanleyrazor

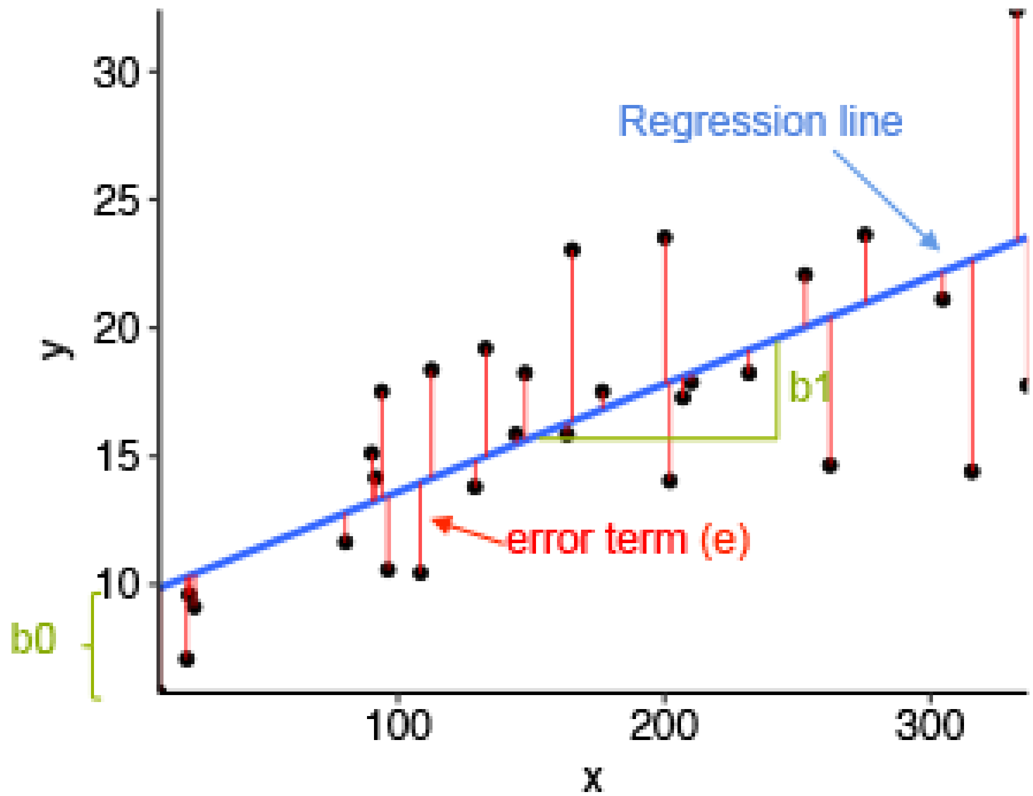
# ADDING NOISE TO LINEAR REGRESSION PREDICTIONS USING THE NEAREST NEIGHBOUR ALGORITHM.

Author: Stanley Sayianka

Date: 10/02/2021

LinkedIn: linkedin.com/in/stanley-sayianka-8a6450170
GitHub: github.com/stanleyrazor

I assume that the reader has working knowledge on the basic simple linear regression model, deriving the coefficients and interpreting them, the reader should also have some knowledge on Expectations and variances and their properties as they relate to random variables.



LinkedIn: linkedin.com/in/stanley-sayianka-8a6450170
GitHub: github.com/stanleyrazor

The simple linear regression model is given as:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \text{ where } i = 1, \ldots, n$$

Where

$$y_i - \text{is the response variable}$$

$$\beta_0 - \text{is the intercept}$$

$$\beta_1 - \text{is the slope of the regression line}$$

$$x_i - \text{is an independent variable}$$

$$\varepsilon_i - \text{is the random error term}$$

We also know that:

$$E(\varepsilon_i) = 0$$

$$Var(\varepsilon_i) = \sigma^2$$

$$\text{The error terms are uncorrelated, and independent, and follow a}$$

$$\text{normal distribution with mean } 0 \text{ and variance } \sigma^2$$

Based on my idea of the noise-added linear regression model, (I will stick to the x-y notation, where x is the independent variable and y is the dependent variable)

For a new test input x, we use the linear regression model to predict its y value, and then for every y value predicted, I will add an error term which is obtained by averaging the error terms of the nearest neighbor of our new x instance. Remember the nearest neighbors of the test instance are all obtained from the training set that was used to fit the linear regression model.

Therefore:

$$\text{Let } \hat{y}_i \text{ be the predicted value of the new test instance } x_{i.},$$

$$\text{and let } \theta_i \text{be the error component that we desire to add to } \hat{y}_i$$

Assuming we obtain K nearest neighbors to the new test instance and we obtain their residuals from the linear regression model as shown below:

$$\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \ldots, \varepsilon_k$$

Then:

$$\theta_i = \frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \cdots + \varepsilon_k}{k}$$

Thus for the new test instance, its predicted value will be given by:

LinkedIn: linkedin.com/in/stanley-sayianka-8a6450170
GitHub: github.com/stanleyrazor

$$\hat{y}_i = \beta_0 + \beta_1 x_i + \frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \cdots + \varepsilon_k}{k}$$

The desired error term has the following properties:

$$E(\theta_i) = E\left(\frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \cdots + \varepsilon_k}{k}\right)$$

$$= \frac{1}{k} E(\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \cdots + \varepsilon_k)$$

$$= \frac{1}{k}\{E(\varepsilon_1) + E(\varepsilon_2) + E(\varepsilon_3) + \cdots + E(\varepsilon_k)\}$$

$$But\ E(\varepsilon_i) = 0, therefore$$

$$E(\theta_i) = \frac{1}{k} * 0 = 0$$

$$Var(\theta_i) = Var\left(\frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \cdots + \varepsilon_k}{k}\right)$$

$$= \frac{1}{k^2} * Var(\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \cdots + \varepsilon_k)$$

$$= \frac{1}{k^2}\{Var(\varepsilon_1) + Var(\varepsilon_2) + Var(\varepsilon_3) + \cdots + Var(\varepsilon_k) + 2 * Covariance(\varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots, \varepsilon_k)\}$$

$$Since\ the\ error\ terms\ are\ uncorrelated\ and\ independent, their\ covariance\ vanishes$$

$$But\ Var(\varepsilon_i) = \sigma^2$$

$$= \frac{1}{k^2} * k\sigma^2$$

$$Thus: Var(\theta_i) = \frac{\sigma^2}{k}$$

We see that then the desired noise follows a Normal distribution with mean 0 and variance equal to $\frac{\sigma^2}{k}$

$$\theta_i \sim N(\mu = 0, \sigma^2 = \frac{\sigma^2}{k})$$