Regression

Multiple linear regression

- Used to estimate the relationship between **two or more independent variables** and **one dependent variable**.
- Used when
  - How strong the relationship is between two or more independent variables and one dependent variable
  - The value of the dependent variable at a certain value of the independent variables

Multiple linear regression

- In multiple linear regression, it is possible that some of the independent variables are actually correlated with one another, so it is important to check these before developing the regression model.
- If two independent variables are too highly correlated, then only one of them should be used in the regression model.

Multiple linear regression

- The data follows a normal distribution.
- The formula for a multiple linear regression is:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_N x_N$$

y predicted value of the dependent variable

Multiple linear regression

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{N1} \\ 1 & x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & & & & \\ 1 & x_{1n} & x_{2n} & \cdots & x_{Nn} \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$$

- Regression Coefficient

$$B = (X^T X)^{-1} X^T Y$$

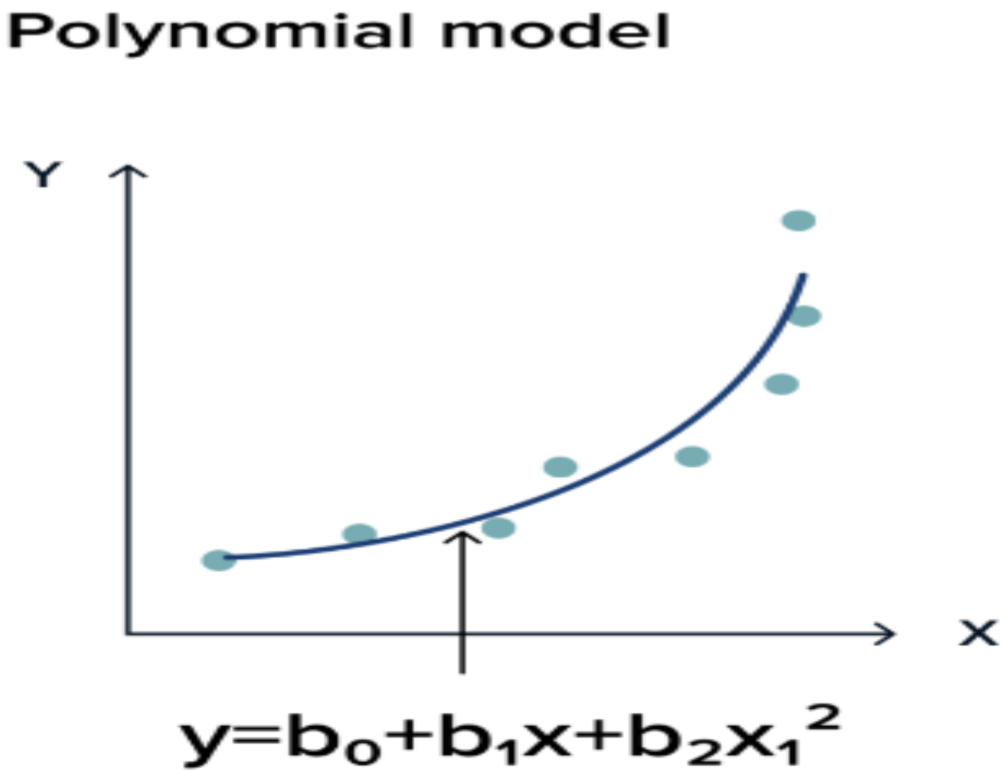
Polynomial regression

- The relationship between the independent variable x and the dependent variable y is described as an nth degree polynomial in x.
- the fitting of a nonlinear relationship between the value of x and the conditional mean of y.

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$

- Eg: to predict how many likes your new social media post will have at any given point after the publication.

Polynomial regression



- A polynomial regression model defines the relationship between x and y by an equation in the following form:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_k x^k.$$

- To determine the optimal values of the parameters α0, α1, …, αk the method of ordinary least squares is used. The values of the parameters are those values which minimizes the sum of squares:

$$E = \sum_{i=1}^{n} [y_i - (\alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \cdots + \alpha_k x_i^k)]^2.$$

- The optimal values of the parameters are obtained by solving the following system of equations:

$$\frac{\partial E}{\partial \alpha_i} = 0, \quad i = 0, 1, \ldots, k.$$

- Let the values of values of the parameters which minimizes E be

$$\alpha_i = a_i, \quad i = 0, 1, 2, \ldots, n.$$

- System of (k+1) linear equations

$$
\begin{aligned}
\sum y_i &= \alpha_0 n + \alpha_1(\sum x_i) + \cdots + \alpha_k(\sum x_i^k) \\
\sum y_i x_i &= \alpha_0(\sum x_i) + \alpha_1(\sum x_i^2) + \cdots + \alpha_k(\sum x_i^{k+1}) \\
\sum y_i x_i^2 &= \alpha_0(\sum x_i^2) + \alpha_1(\sum x_i^3) + \cdots + \alpha_k(\sum x_i^{k+2}) \\
&\vdots \\
\sum y_i x_i^k &= \alpha_0(\sum x_i^k) + \alpha_1(\sum x_i^{k+1}) + \cdots + \alpha_k(\sum x_i^{2k})
\end{aligned}
$$

Polynomial regression

$$D = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}$$

$$\vec{a} = (D^T D)^{-1} D^T \vec{y},$$

Polynomial regression

- Can model non-linear relationships between variables.
- A large range of different functions that you can use for fitting.

Gradient Descent

- Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms
- The model tries to get the best fit regression line to predict the value of y based on the given input value (x)
- Cost function: the Root Mean Squared error between the predicted value (pred) and true value (y)
- Model targets to minimize the cost function.

Gradient Descent

- Cost Function

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$\boxed{minimize \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2}$$

Cost Function

$$J(\Theta_0, \Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}[\underset{\substack{\uparrow \\ \text{Predicted Value}}}{h_\Theta(x_i)} - \underset{\substack{\uparrow \\ \text{True Value}}}{y_i}]^2$$

Gradient Descent

$$\boxed{\Theta_j = \Theta_j - \underset{\substack{\uparrow \\ \text{Learning Rate}}}{\alpha}\frac{\partial}{\partial \Theta_j} J(\Theta_0, \Theta_1)}$$

Now,

$$\begin{aligned}
\frac{\partial}{\partial \Theta} J_\Theta &= \frac{\partial}{\partial \Theta}\frac{1}{2m}\sum_{i=1}^{m}[h_\Theta(x_i) - y]^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}(h_\Theta(x_i) - y)\frac{\partial}{\partial \Theta_j}(\Theta x_i - y) \\
&= \frac{1}{m}(h_\Theta(x_i) - y)x_i
\end{aligned}$$

θ_j : Weights of the hypothesis.

$h_{\theta(xi)}$: predicted y value for i^{th} input.

j : Feature index number (can be 0, 1, 2, ...

α : Learning Rate of Gradient Descent.

Therefore,

$$\boxed{\Theta_j := \Theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}[(h_\Theta(x_i) - y)x_i]}$$

Gradient Descent

- Cost function can be plotted as a function of parameter estimates i.e. parameter range of hypothesis function and the cost resulting from selecting a particular set of parameters.
- Move downward towards pits in the graph, to find the minimum value.
- Gradient Descent step-downs the cost function in the direction of the steepest descent.
- The size of each step is determined by parameter $\boldsymbol{\alpha}$ known as **Learning Rate**.

Gradient Descent