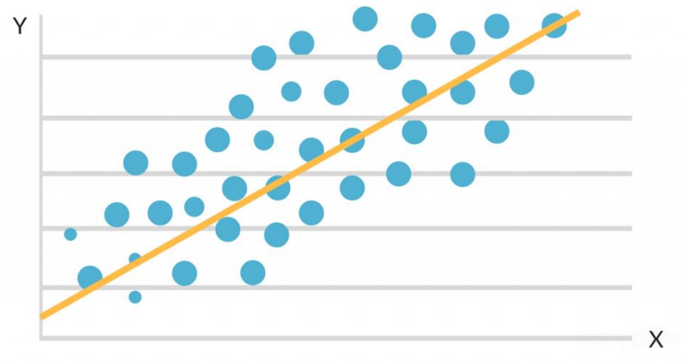
Regression Analysis



Dr. Adel Abusitta

What is Regression Analysis?

- **Regression analysis** is a statistical technique used to examine the relationship between a **dependent variable** (also known as the response variable) and **independent variable** (also known as the predictor variables).

- The objective of regression analysis is to develop a mathematical model that can be used **to predict the value** of the dependent variable based on the values of the independent variables.

Linear Regression

- Linear regression is a statistical technique used to model the relationship between a dependent variable and one or more independent variables.
- The **goal** of linear regression is to **find a linear relationship between the independent variables and the dependent variable**, such that the independent variables can be used to predict the dependent variable.
- The equation for simple linear regression is given by:

$$\mathbf{y = mx + b}$$

where **y** is the dependent variable, **x** is the independent variable, **m** is the slope of the line, and **b** is the **y-intercept** (which represents the value of y when x is equal to 0).

Linear Regression (Multiple linear regression)

- Multiple linear regression is a more complex form of linear regression that involves more than one independent variable. The equation for multiple linear regression is given by:

y = b0 + b1x1 + b2x2 + ... + bnxn

- where **y** is the dependent variable, **x1, x2, ..., xn** are the independent variables, and **b0, b1, b2, ..., bn** are the **coefficients** (which represent the contribution of each independent variable to the dependent variable).

Linear Regression (One-feature example)

We want to fit a linear regression model to predict **Y** based on **X**. The linear regression model has the form:

$$\mathbf{Y = \beta_0 + \beta_1 * X}$$

where $\boldsymbol{\beta_0}$ is the intercept, $\boldsymbol{\beta_1}$ is the slope.

X	Y
1	2
2	4
3	6
4	8
5	10

Linear Regression (One-feature example)

To fit this model, we need to **estimate the values of β_0 and β_1** that best fit the data.

We can do this using the **method of least squares. The idea is to minimize the sum of the squared errors between the predicted values of Y and the actual values of Y**:

minimize $\sum_1^n (Y_i - \hat{Y}_i)^2$

X	Y
1	2
2	4
3	6
4	8
5	10

Linear Regression (One-feature example)

minimize $\sum_{1}^{n} (Y_i - \widehat{Y}_i)^2$

where Y_i is the actual value of Y for observation i, and $\widehat{Y}_i$ is the predicted value of Y for observation i based on the linear regression model.

We can solve for the values of β_0 and β_1 that minimize this sum using the following formulas (**derived using the first derivative**):

$$\beta_1 = \sum_{1}^{n}(X_i - X)(Y_i - \overline{Y}) \,/\, \sum_{1}^{n} (X_i - \overline{X})^2$$

$$\beta_0 = \overline{Y} - \beta_1 * \overline{X}$$

where $\overline{X}$ is the mean of X, $\overline{Y}$ is the mean of Y.

X	Y
1	2
2	4
3	6
4	8
5	10

Linear Regression (One-feature example)

$\boldsymbol{\beta_1} = \sum_1^n (X_i - X)(Y_i - \bar{Y}) \ / \ \sum_1^n (X_i - \bar{X})^2$

$\boldsymbol{\beta_0} = \bar{Y} - \boldsymbol{\beta_1} * \bar{X}$

Using the data in the table above, we can calculate the values of β0 and β1 as follows:

$\bar{X} = (1+2+3+4+5)/5 = 3$ $\bar{Y} = (2+4+6+8+10)/5 = 6$

$\sum_1^n (X_i - X)(Y_i - \bar{Y}) = (1-3)(2-6) + (2-3)(4-6) + (3-3)(6-6) + (4-3)(8-6) + (5-3)(10-6) = 20$

$\sum_1^n (X_i - \bar{X})^2 = (1-3)^2 + (2-3)^2 + (3-3)^2 + (4-3)^2 + (5-3)^2 = 10$

$\boldsymbol{\beta_1} = 20/10 = 2$

$\boldsymbol{\beta_0} = 6 - 2*3 = 0$

X	Y
1	2
2	4
3	6
4	8
5	10

Linear Regression (One-feature example)

- Therefore, the linear regression model that best fits the data is:

Y = 0 + 2*X

- To make a prediction using this model, we can plug in a value of X and solve for Y. For example, if we want to predict Y for **X = 6**, we can use the formula:

- Y = 0 + 2*6 = 12

- So our prediction is that **Y = 12**. Since we don't have a random error term in this model, this prediction is exact and not subject to any variability.

X	Y
1	2
2	4
3	6
4	8
5	10

Fit a multiple linear regression model using more than one feature

Linear Regression (Multiple-feature example)

We want to fit a multiple linear regression model to predict $\boldsymbol{Y}$ based on both $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$. The multiple linear regression model has the form:

$\boldsymbol{Y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 X_1} + \boldsymbol{\beta_2}\ \boldsymbol{X_2}$

where $\boldsymbol{\beta_0}$ is the intercept, $\boldsymbol{\beta_1}$ is the coefficient for $\boldsymbol{X_1}$, and $\boldsymbol{\beta_2}$ is the coefficient for $\boldsymbol{X_2}$.

X1	X2	Y
1	2	10
2	4	15
3	6	20
4	8	25
5	10	30

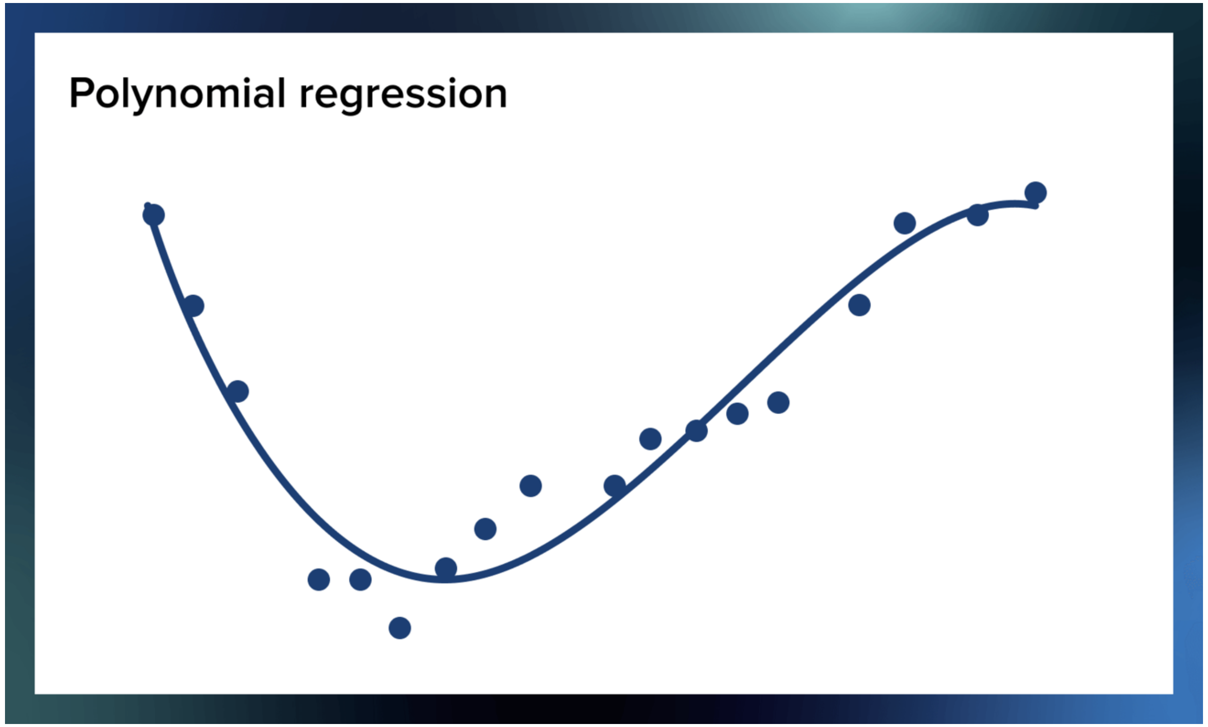
Non-Linear Regression

Polynomial Regression

The general form of a polynomial regression model with degree n is:

$$y = b_0 + b_1x + b_2x^2 + \ldots + b_nx^n$$

Here, **y** represents the dependent variable, **x** is the independent variable, and $\mathbf{b_0, b_1, b_2, \ldots, b_n}$ are the coefficients of the polynomial.

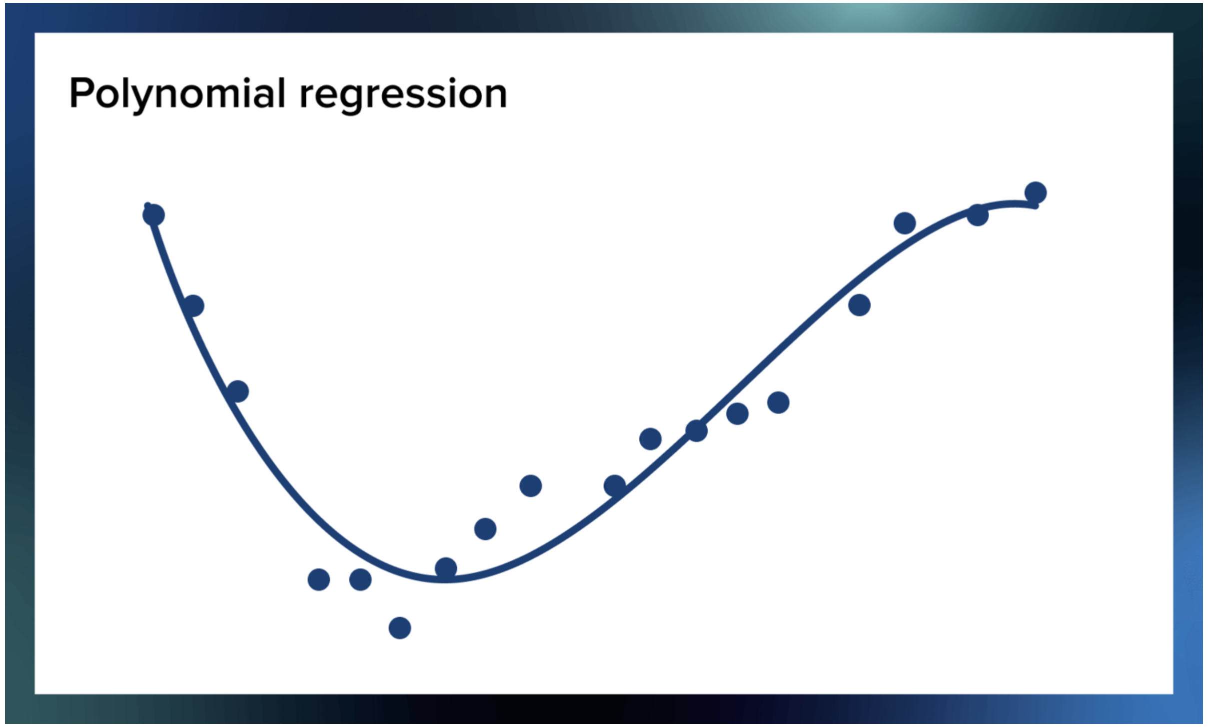


Polynomial Regression

The general form of a polynomial regression model with degree n is:

$$y = b_0 + b_1x + b_2x^2 + \dots + b_nx^n$$

We can solve for the coefficients using **linear transformations**



Polynomial Regression

Example: We want to find the coefficients c0, c1, and c2 for the polynomial model
y = c0 + c1*X + c2*X^2 using linear transformations.

X	y
1	2
2	5
3	10
4	17
5	26

Feature Transformation: Create a new matrix by applying a **linear transformation** to the original feature X. The transformed matrix will include the square term X^2. The transformed matrix will look like this:

X=X1	X^2=X2
1	1
2	4
3	9
4	16
5	25

Polynomial Regression

Another example:

y = c0 + c1*X + c2*X^2

Linear Regression Coefficients:

Coefficient	Value
c0: Intercept	0.8
c1	0.86666667
c2	0.23333333

The polynomial regression model equation is: **y = 0.2333 * X^2 + 0.8667 * X + 0.8**

Original Data:

X	y
1	2
2	4.7
3	6.8
4	8
5	10

Polynomial Features:

X=X1	X^2 = X2
1	1
2	4
3	9
4	16
5	25

K-Nearest Neighbors (KNN) Regression

Let's say we want to predict the output target for a test point with input feature value of 2.5 using KNN regression with **K=2.**

To do this, we first need to calculate the distances between the test point and each of the 5 training data points:

Input Feature (X)	Output Target (Y)
1	2
2	4
3	1
4	5
5	3

KNN Regression

Next, we select the **K=2** training data points with the smallest distances to the test point, which are (2,4) and (3,1).

To predict the output target for the test point, we take the average of the output target values of **the K nearest neighbors**, which in this case is **(4 + 1)/2 = 2.5.** This is our predicted output target value for the test point.

Input Feature (X)	Output Target (Y)	Distance to Test Point (X=2.5)
1	2	1.5
2	4	0.5
3	1	0.5
4	5	1.5
5	3	2.5

KNN Regression

To substitute this predicted value for the test point, we replace the input feature value with the test point's value (2.5) and the output target value with the predicted value (2.5). So our final result would be:

Input Feature (X)	Output Target (Y)	Distance to Test Point (X=2.5)
1	2	1.5
2	4	0.5
3	1	0.5
4	5	1.5
5	3	2.5

Input Feature (X)	Output Target (Y)
2.5	2.5