**Polynomial Regression**

Polynomial regression is a type of regression analysis used to model relationships between a dependent variable (target) and one or more independent variables (predictors) by fitting a polynomial equation to the data. While simple linear regression fits a straight line to the data, polynomial regression uses higher-degree polynomial functions to better capture more complex relationships.

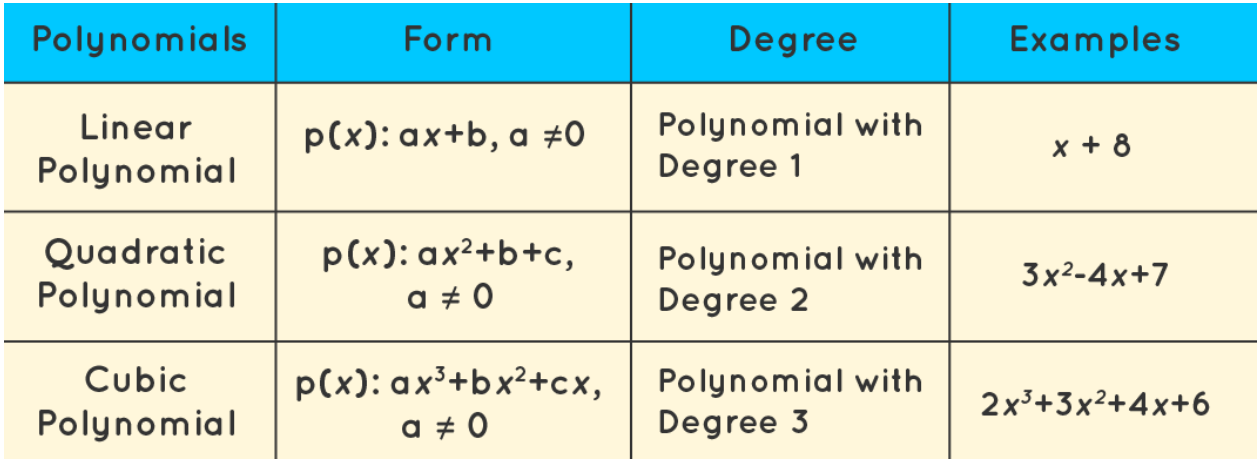
The general equation for polynomial regression with one independent variable can be written as:



The goal of polynomial regression is to find the coefficients (the *β* values) that minimize the difference between the predicted values and the actual values of the dependent variable. This is often done using techniques like the method of least squares.

A blue line graph with numbers

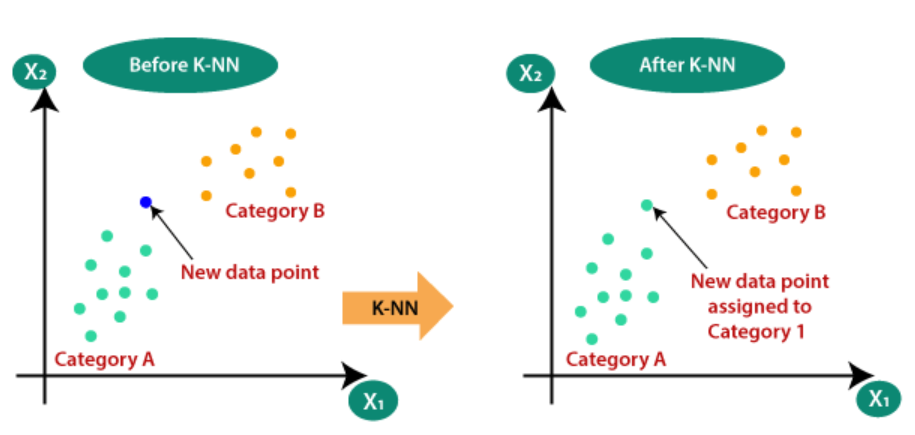
Description automatically generated



**K-Nearest Neighbour**

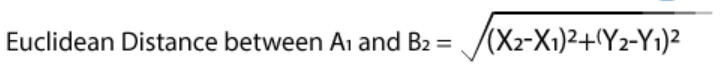
The K-Nearest Neighbour (KNN) algorithm is a popular machine learning technique used for classification and regression tasks. It relies on the idea that similar data points tend to have similar labels or values.

During the training phase, the KNN algorithm stores the entire training dataset as a reference. When making predictions, it calculates the distance between the input data point and all the training examples, using a chosen distance metric such as Euclidean distance. The algorithm identifies the K nearest neighbors to the input data point based on their distances. In the case of classification, the algorithm assigns the most common class label among the K neighbors as the predicted label for the input data point. For regression, it calculates the average or weighted average of the target values of the K neighbors to predict the value for the input data point.

****

**Working**

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

****

**Advantages**

* It is simple to implement.
* It is robust to the noisy training data.
* It can be more effective if the training data is large.

**Disadvantages**

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.