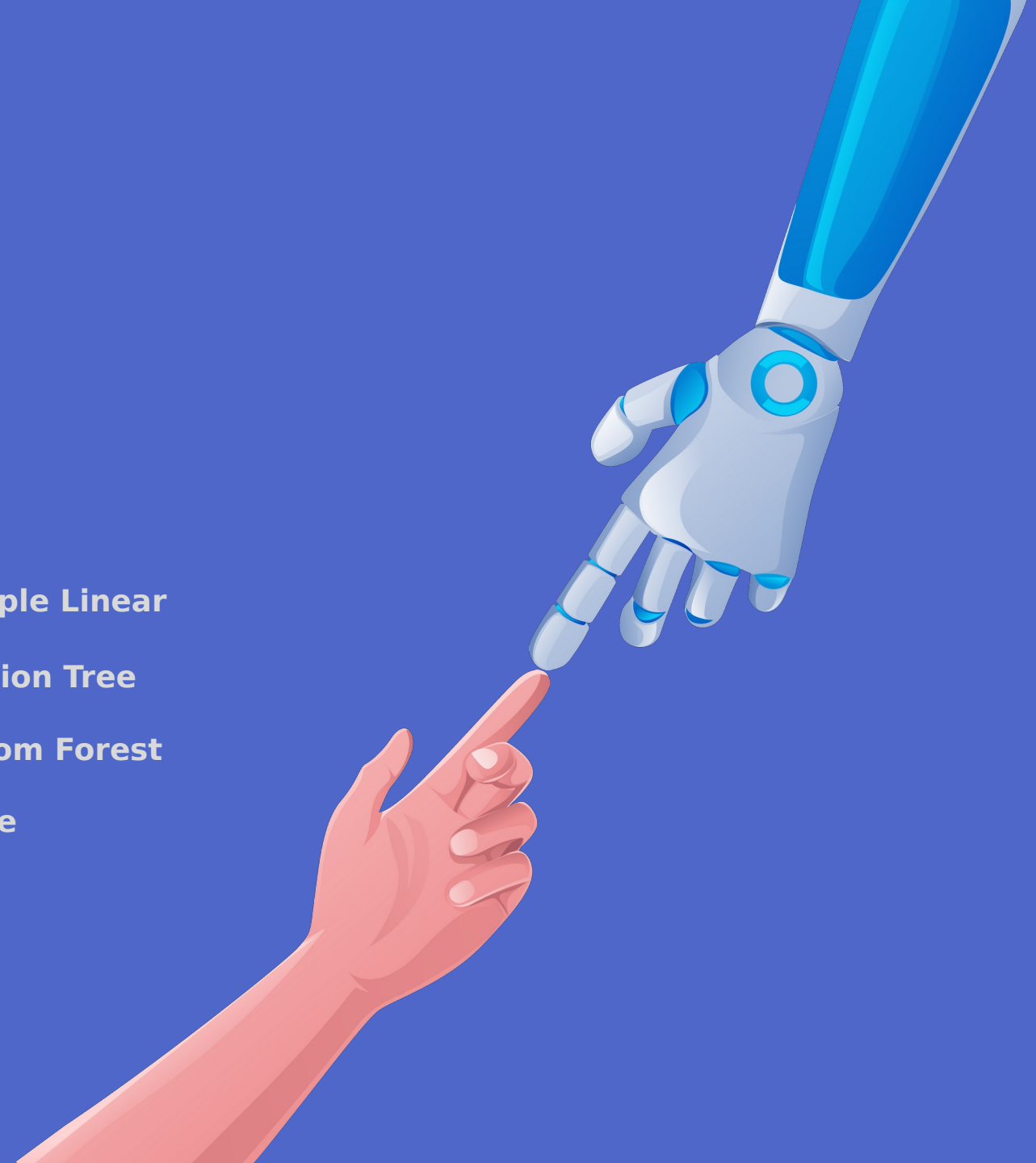
Machine Learning

By : Khaled Tarek

Agenda:

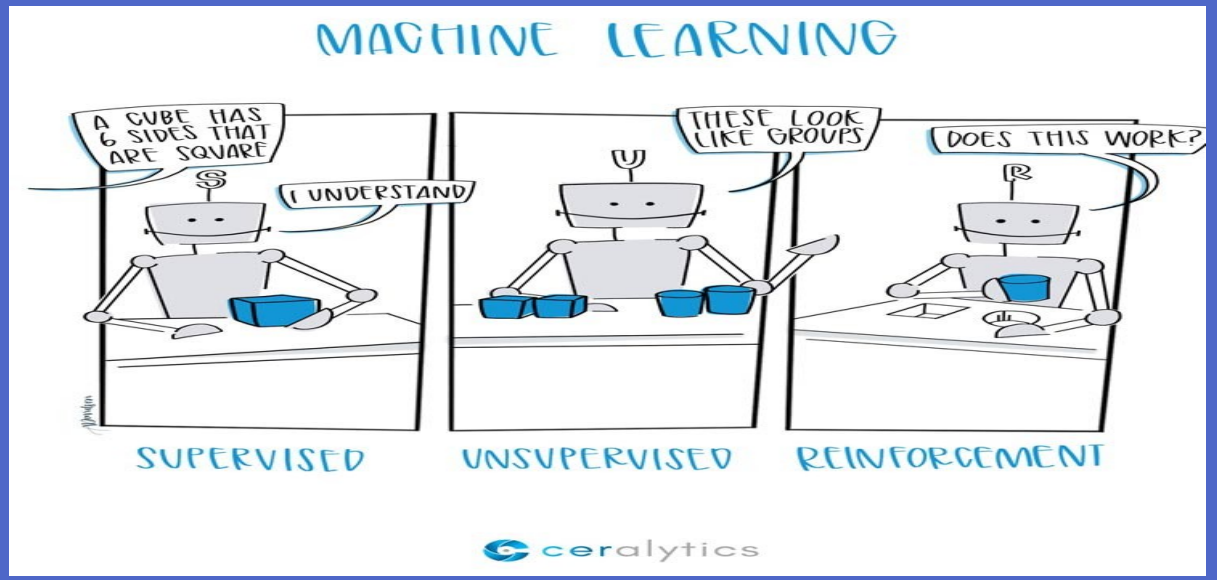
- What is Machine Learning?
- Why use Machine Learning?
- Data Preprocessing.
- Regression

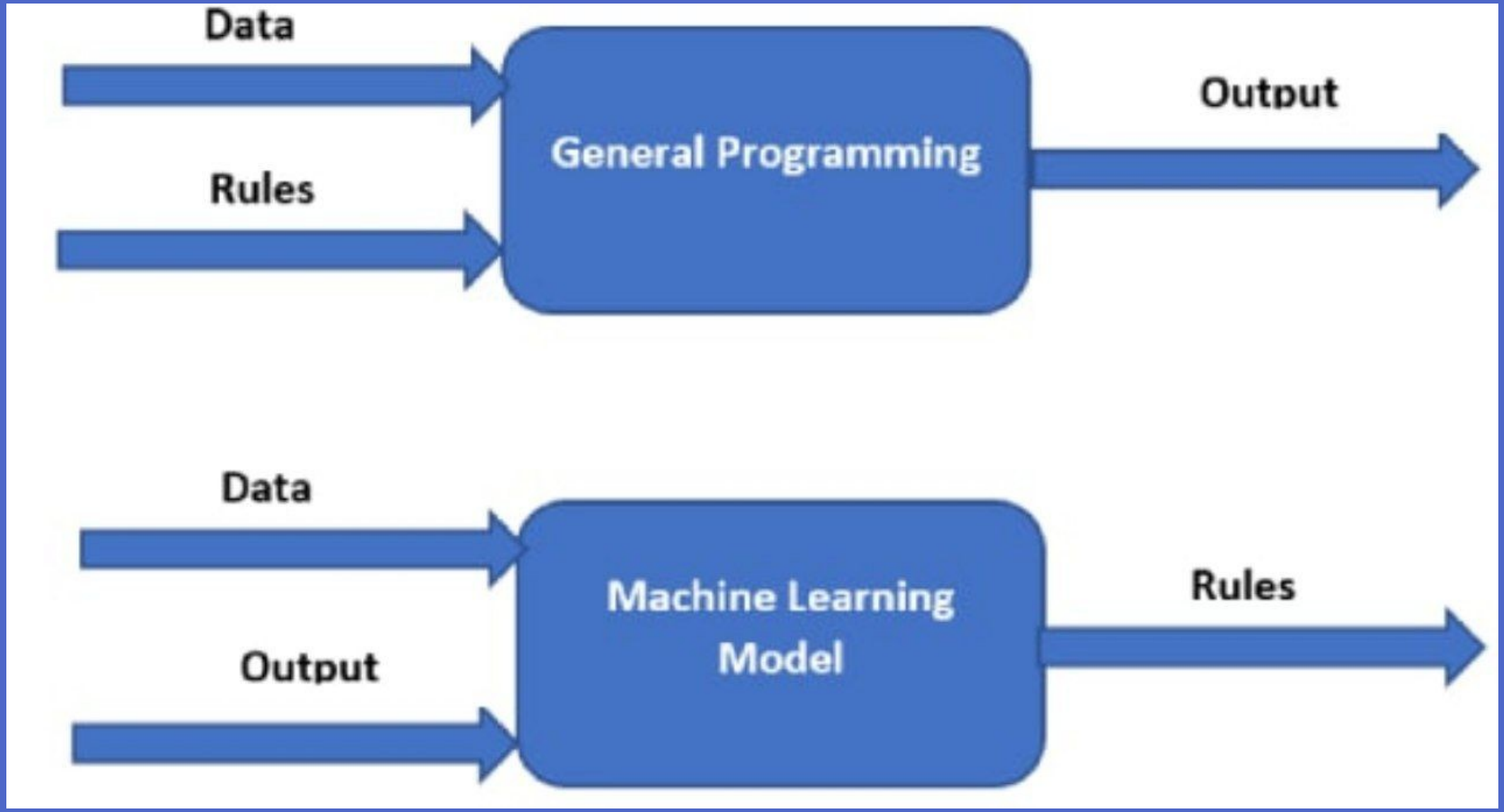
- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Decision Tree Regression
- Support Vector Regression (SVR)
- Random Forest Regression
- Evaluating Regression Models Performance
- Regression Model Selection



What is Machine Learning?

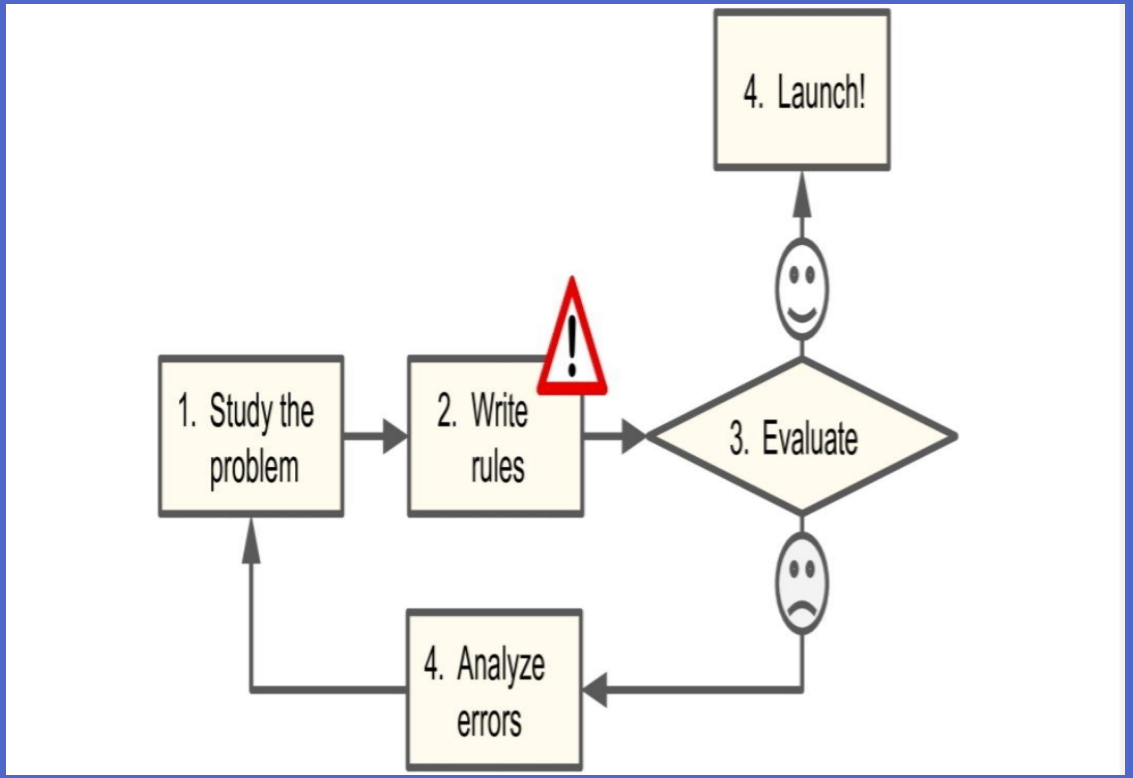
Machine Learning is the science (and art) of programming computers so they can learn from data and field of study that gives computers the ability to learn without being explicitly programmed.



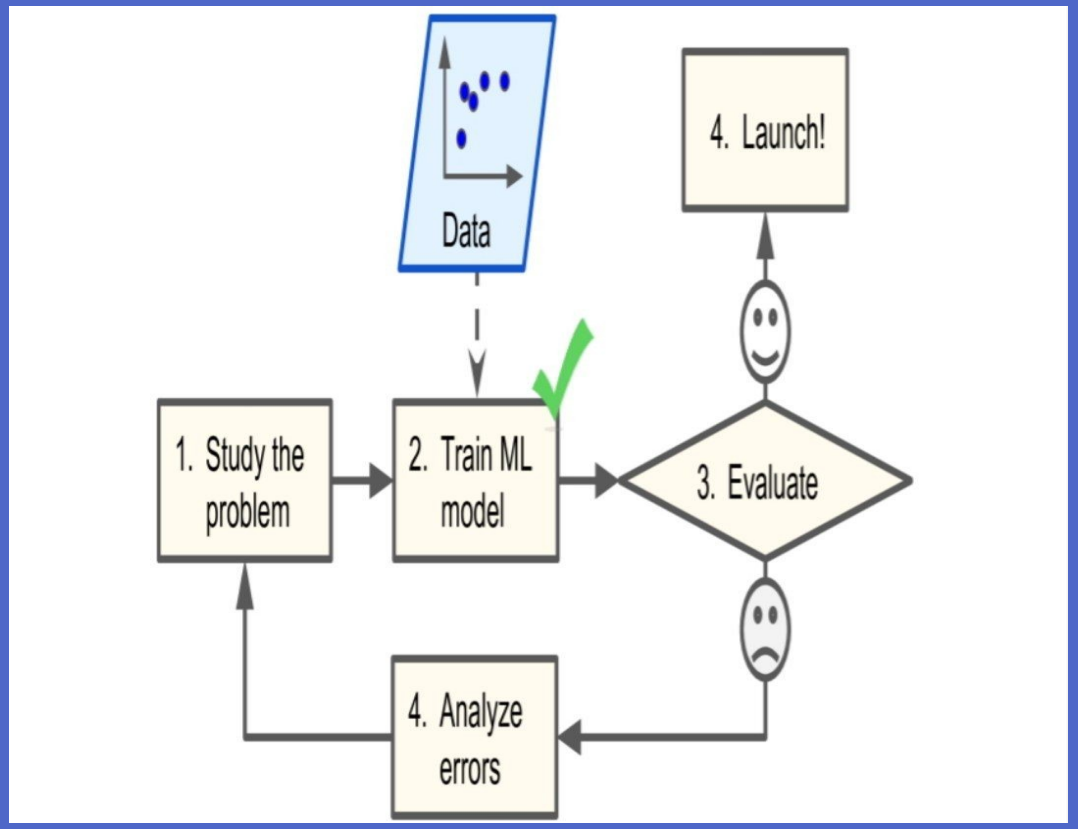

Data
Rules
General Programming
Output
Data
Output
Machine Learning Model
Rules

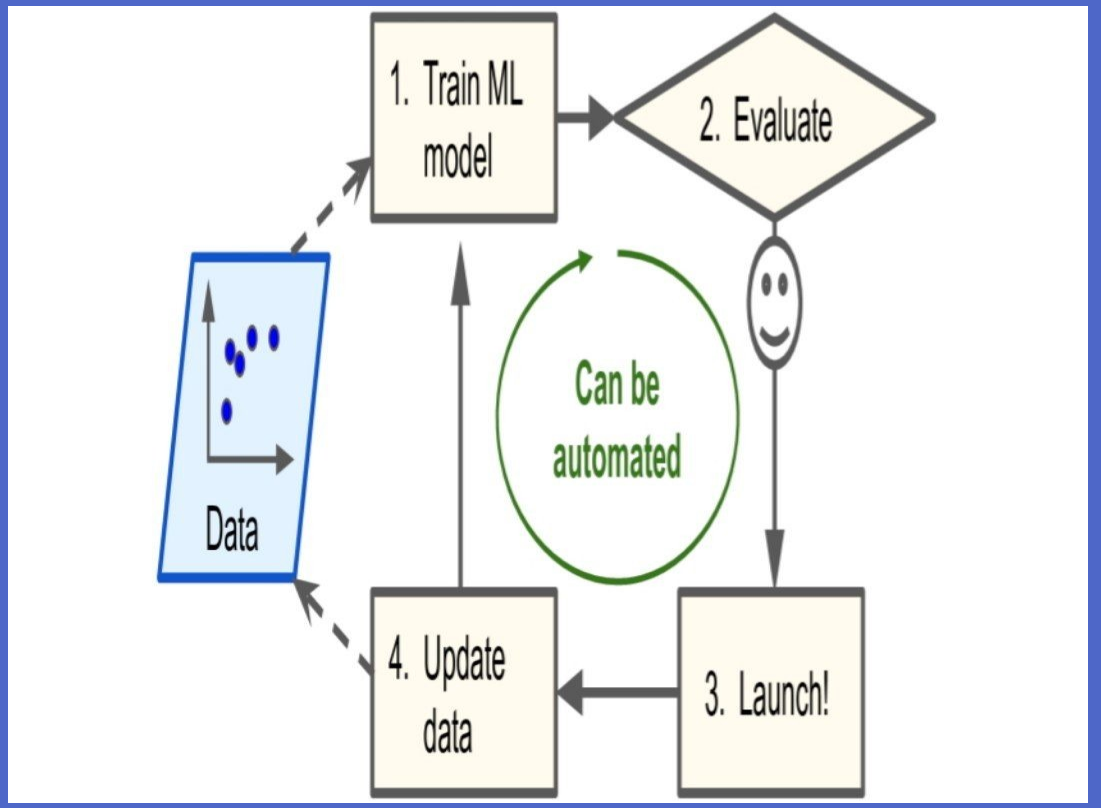
Why Use Machine Learning?

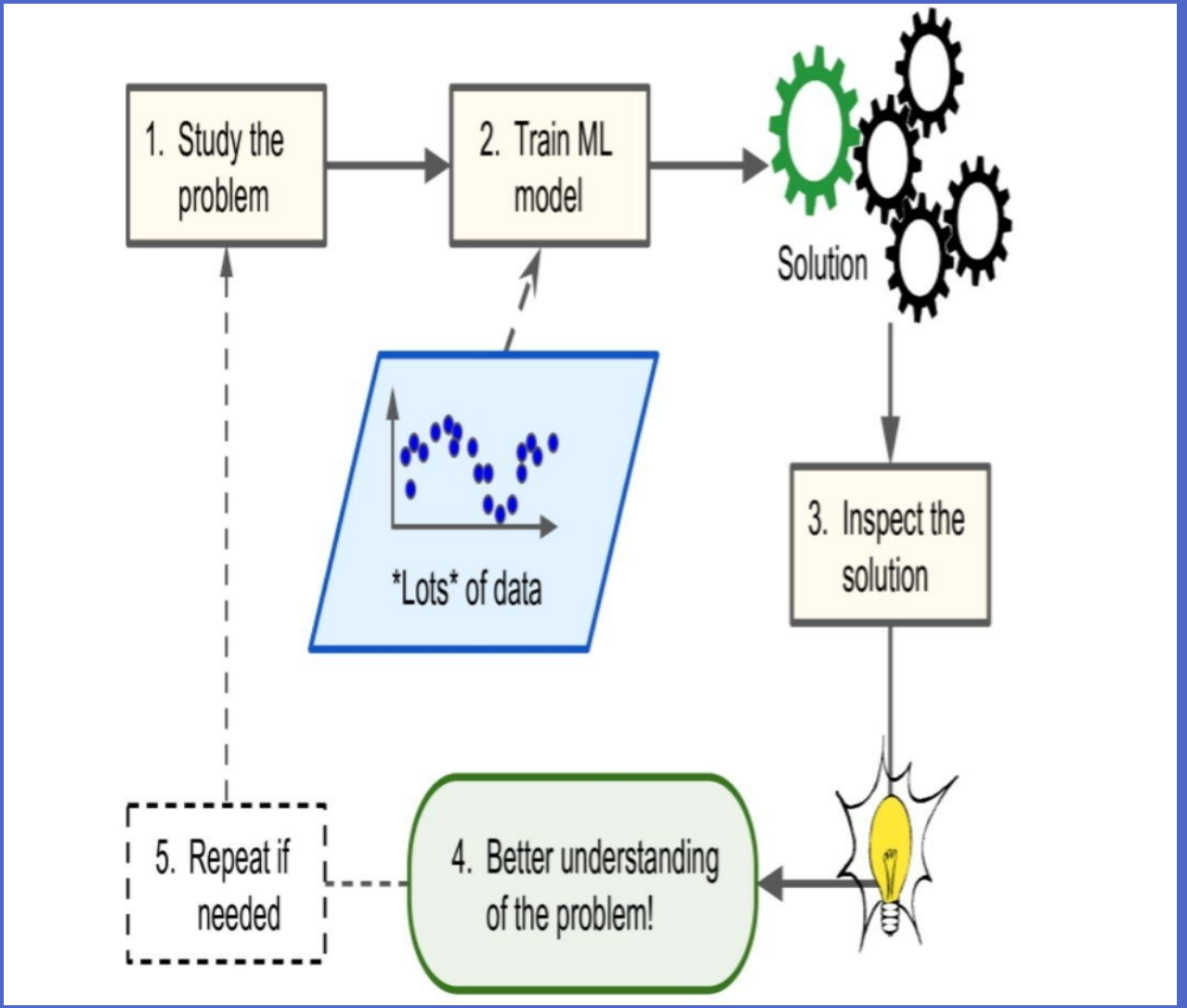
Consider how you would write a spam filter using traditional programming techniques



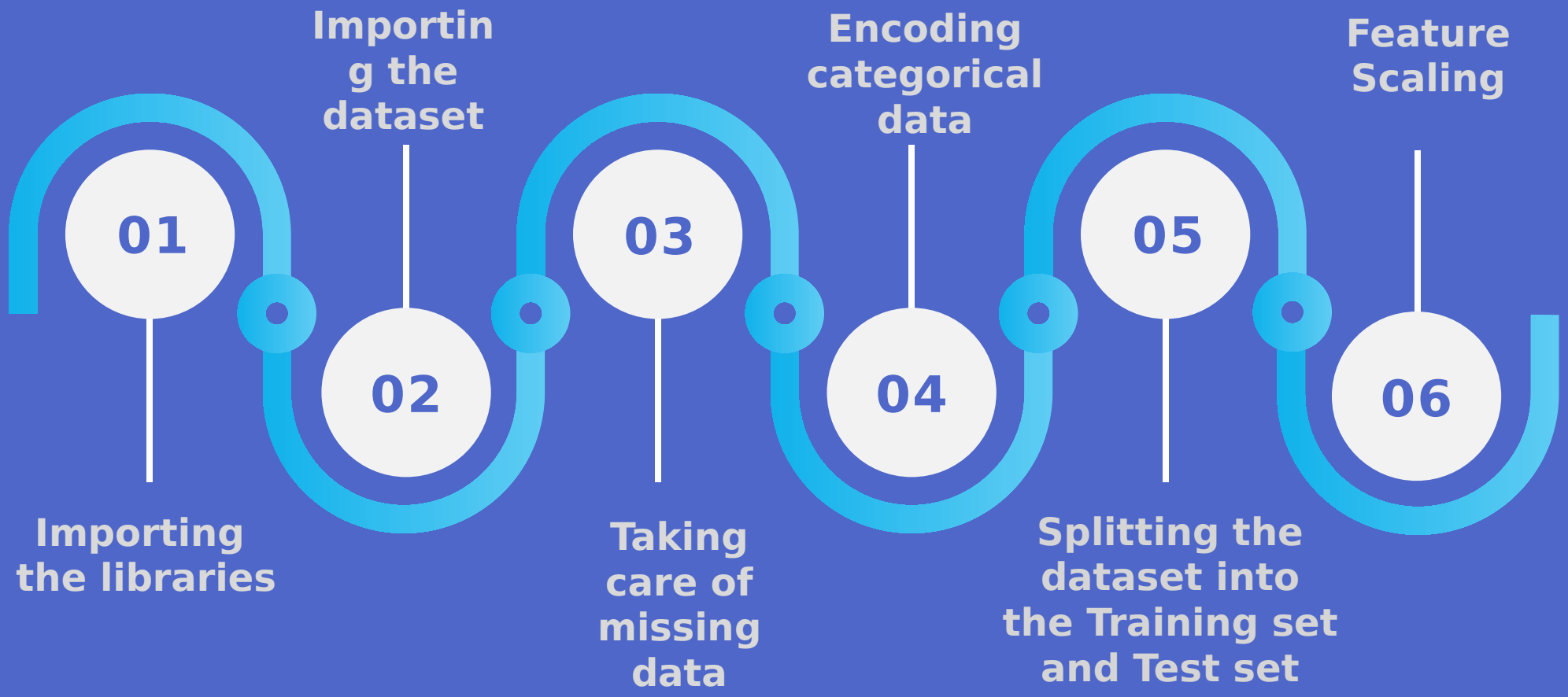
In contrast, a spam filter based on Machine Learning techniques automatically notices that “For U” has become unusually frequent in spam flagged by users, and it starts flagging them without your intervention






1. Study the problem
2. Train ML model
Solution
Lots of data
3. Inspect the solution
4. Better understanding of the problem!
5. Repeat if needed

Data Preprocessing

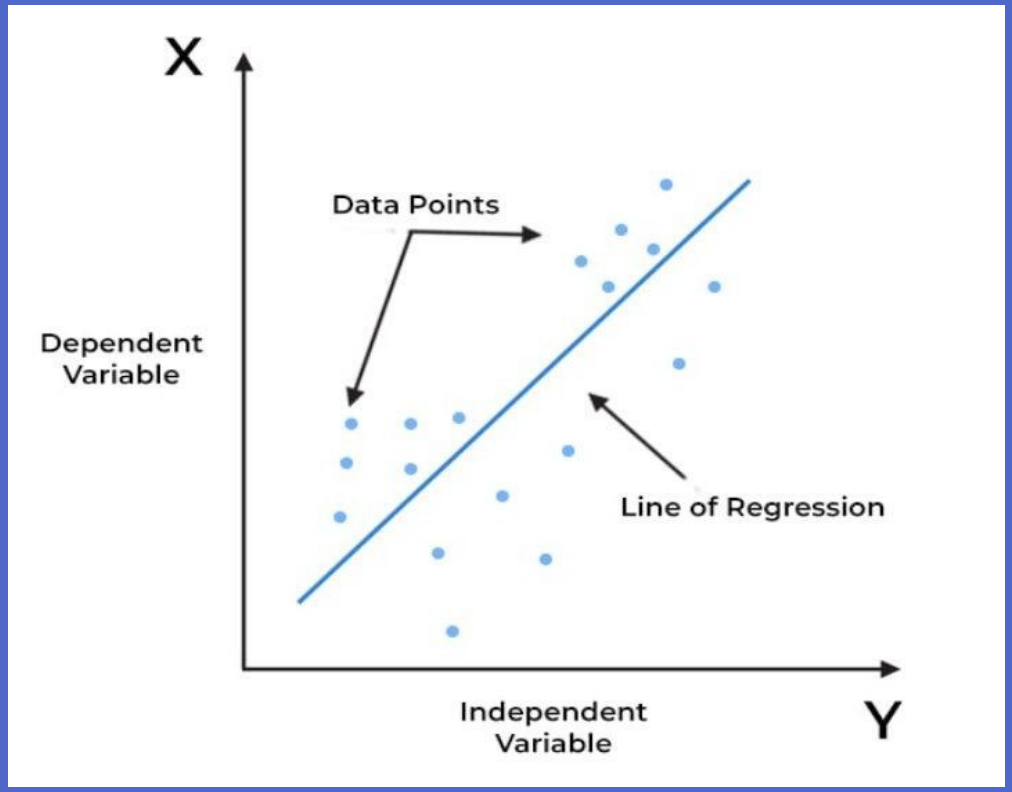


Regression

- Regression is defined as a statistical method that helps us to analyze and understand the relationship between two or more variables of interest.
  regression analysis helps to understand which factors are important, which factors can be ignored, and how they are influencing each other.

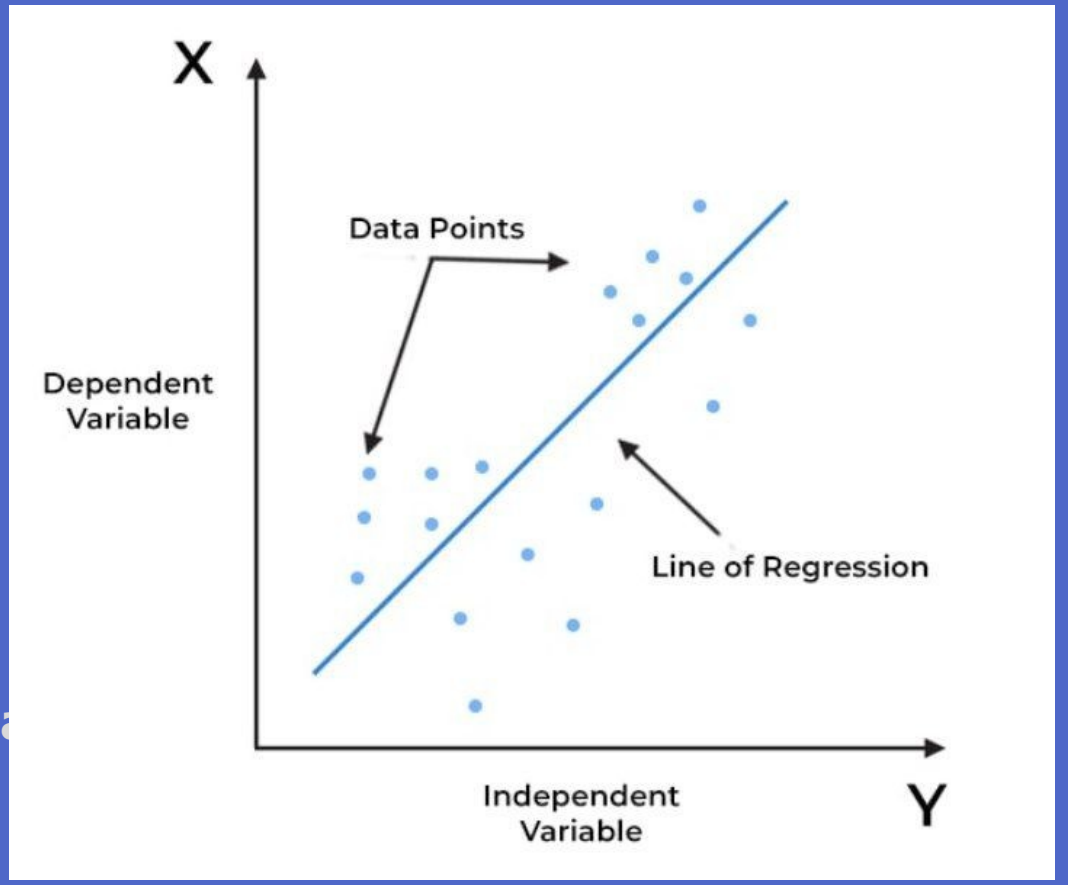
For example,
it can be used to quantify the relative impacts of age, gender, and diet (the predictors) on weight (the output or dependent).

- House prices based on size, locations,...
- Salary prediction based on experience
- Stock Market prediction
- Weather prediction
- ...



Regress
ion

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Support Vector Machine (SVR)
- Decision Tree Regression
- Random Forest Regression
- Evaluating Regression Models Performa
- Regression Model Selection



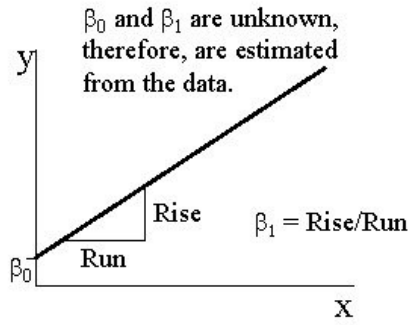
1- Simple Linear Regression (Equation of a straight line)

It also called ordinary least squares (OLS)

Simple Linear Regression Model

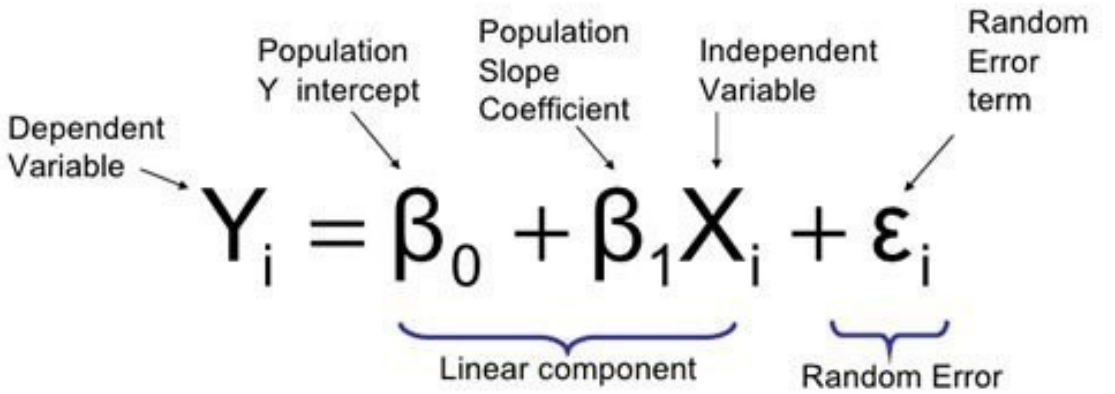
$$y = \beta_0 + \beta_1 x + \varepsilon$$

y = dependent variable
x = independent variable
β_0 = intercept
β_1 = slope of the line
ε = error variable

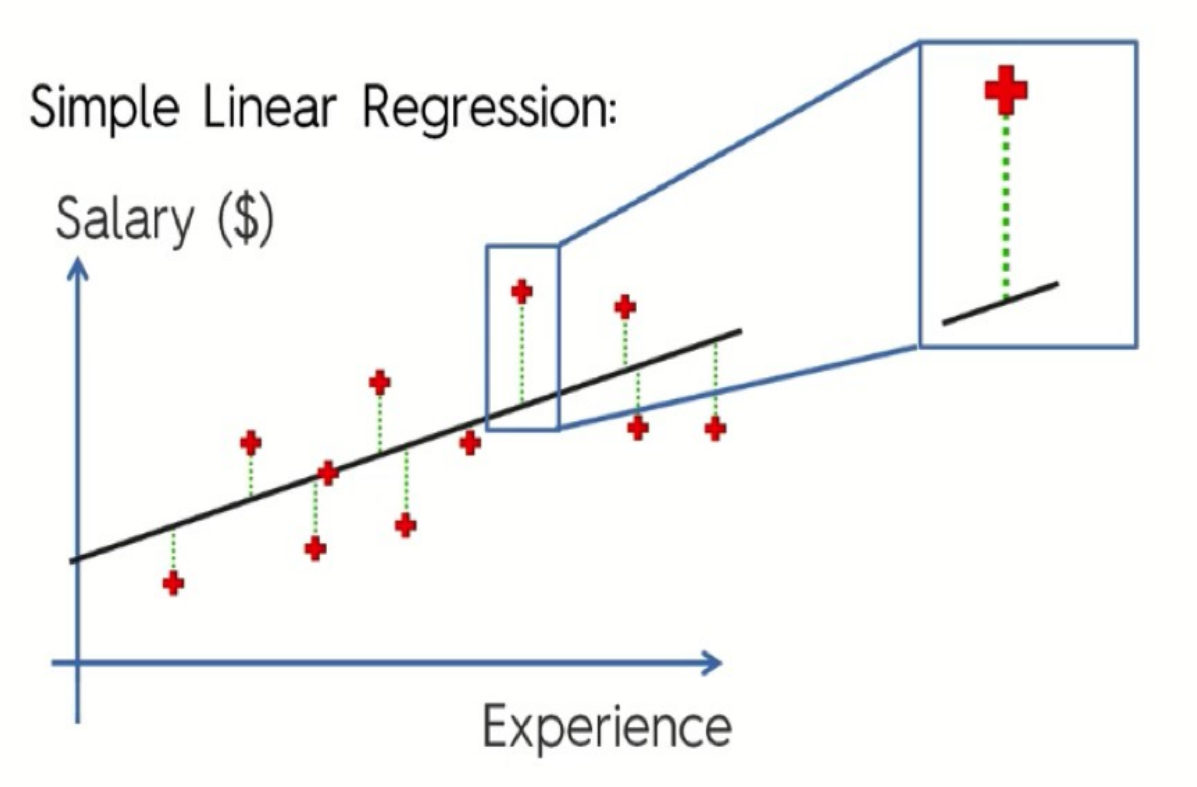


6

Simple Linear Regression Model



1- Simple Linear Regression (Calculate Cost Function)



The goal is to minimize the sum of squares of residuals (y-y') is as small as possible

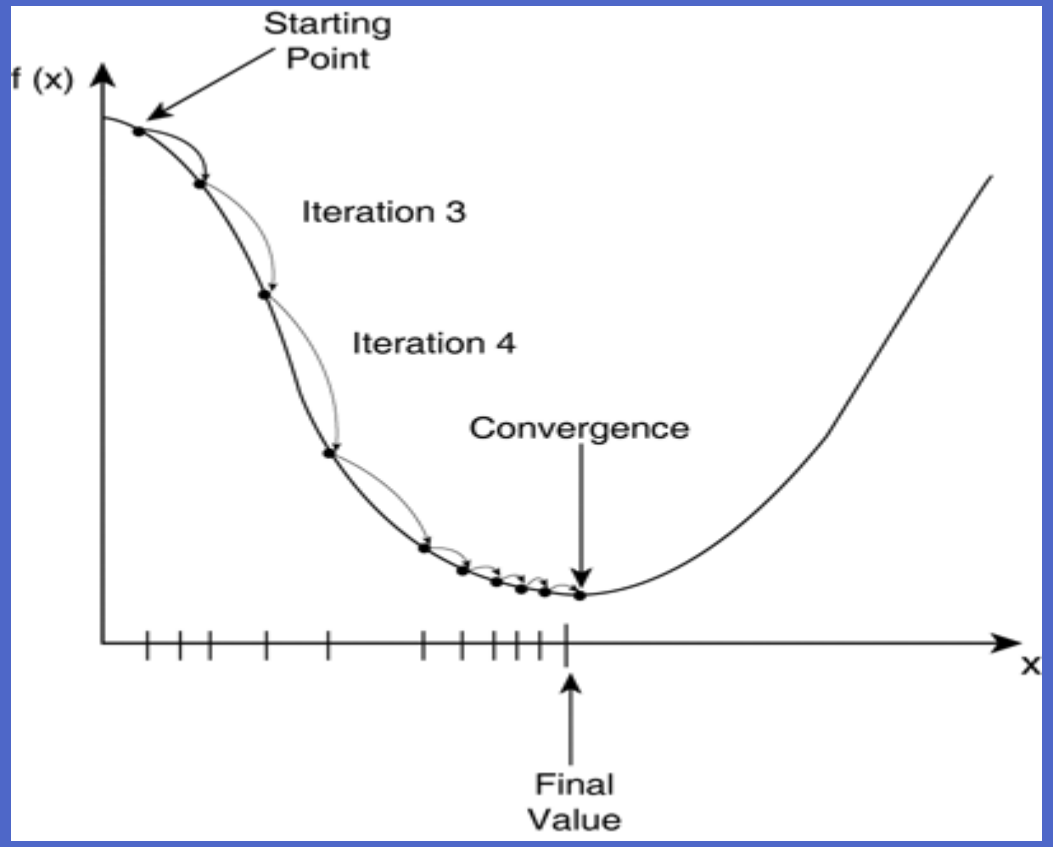
Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

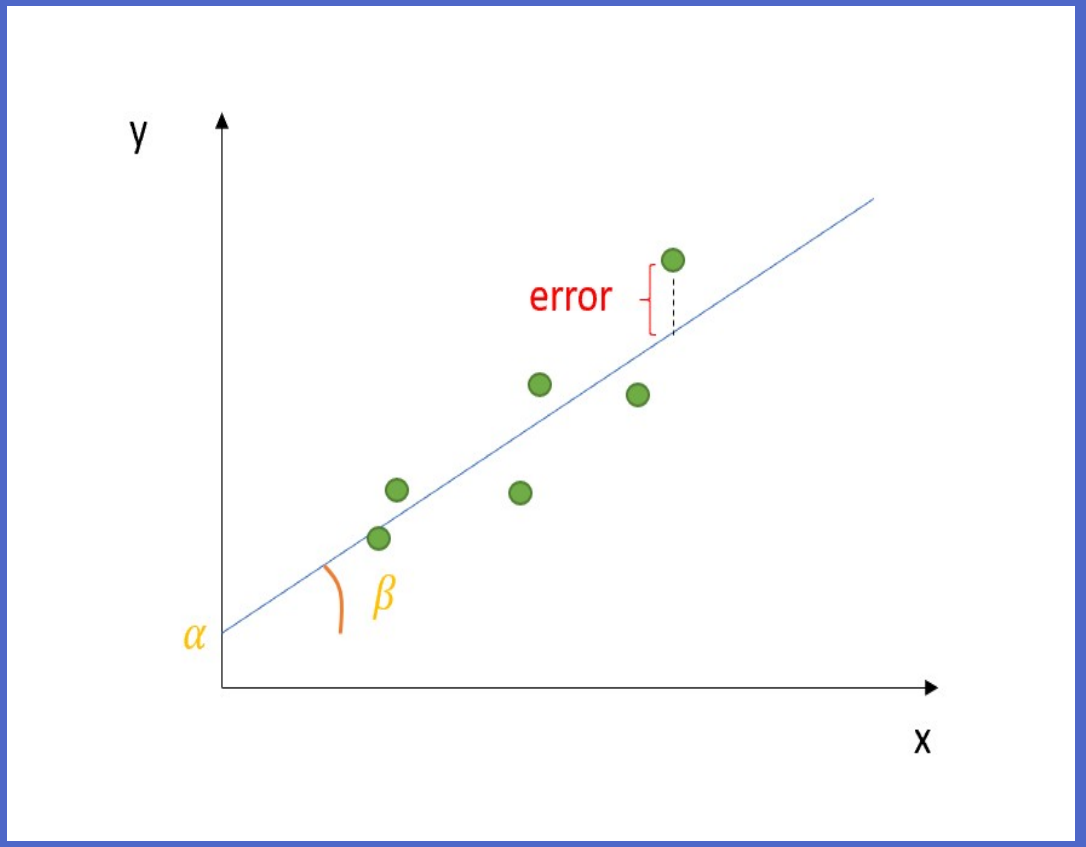
Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

1- Simple Linear Regression (Minimize cost using Gradient Descent)





repeat until convergence:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i)$$

1- Simple Linear Regression (Minimize cost using Gradient Descent)

Cost Function

$$J(\Theta_0, \Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}[h_\Theta(x_i) - y_i]^2$$

$h_\Theta(x_i)$: Predicted Value; y_i: True Value

Gradient Descent

$$\Theta_j = \Theta_j - \alpha\frac{\partial}{\partial\Theta_j}J(\Theta_0, \Theta_1)$$

α: Learning Rate

Now,

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

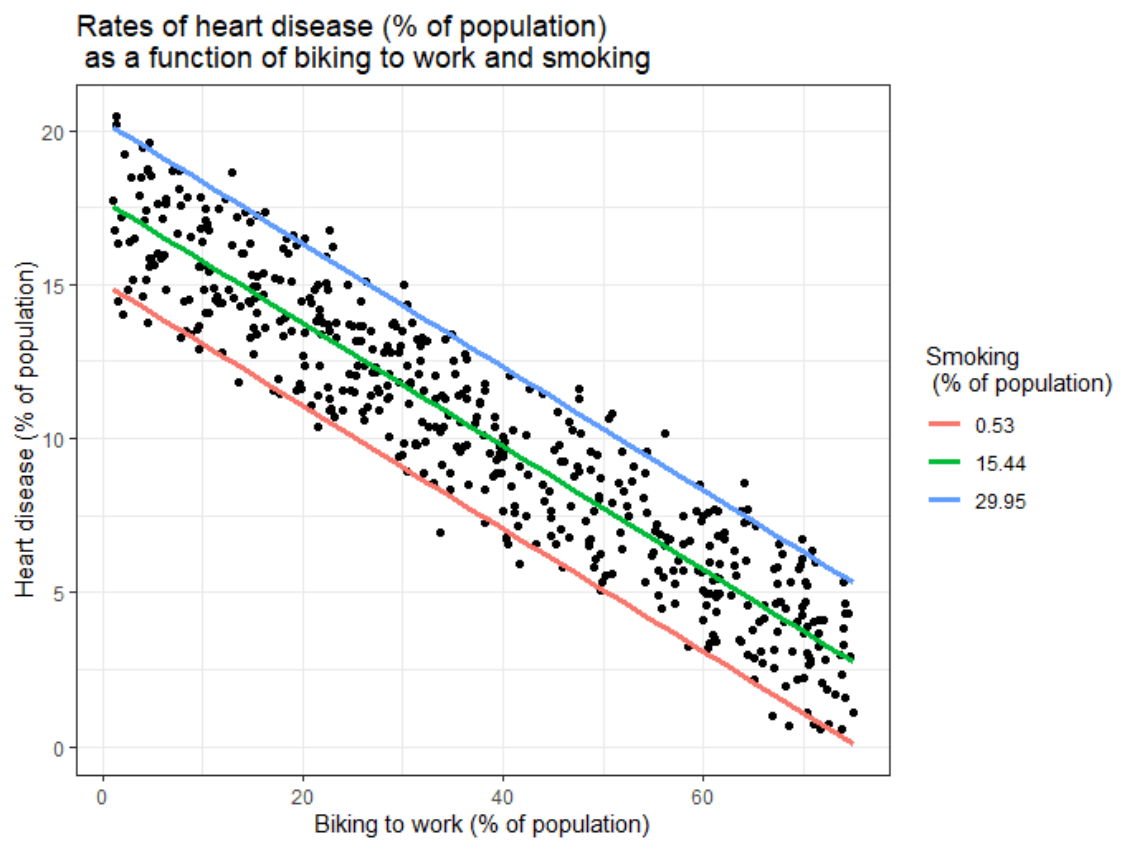
$$= \frac{1}{m}\sum_{i=1}^{m}(h_\Theta(x_i) - y)\frac{\partial}{\partial\Theta_j}(\Theta x_i - y)$$

$$= \frac{1}{m}(h_\Theta(x_i) - y)x_i$$

Therefore,

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

2-Multiple Linear Regression (Equation)



Simple Linear Regression

$$y = b_0 + b_1 * x_1$$

Multiple Linear Regression

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n$$

2- Multiple Linear Regression (Update theta values)

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}$$

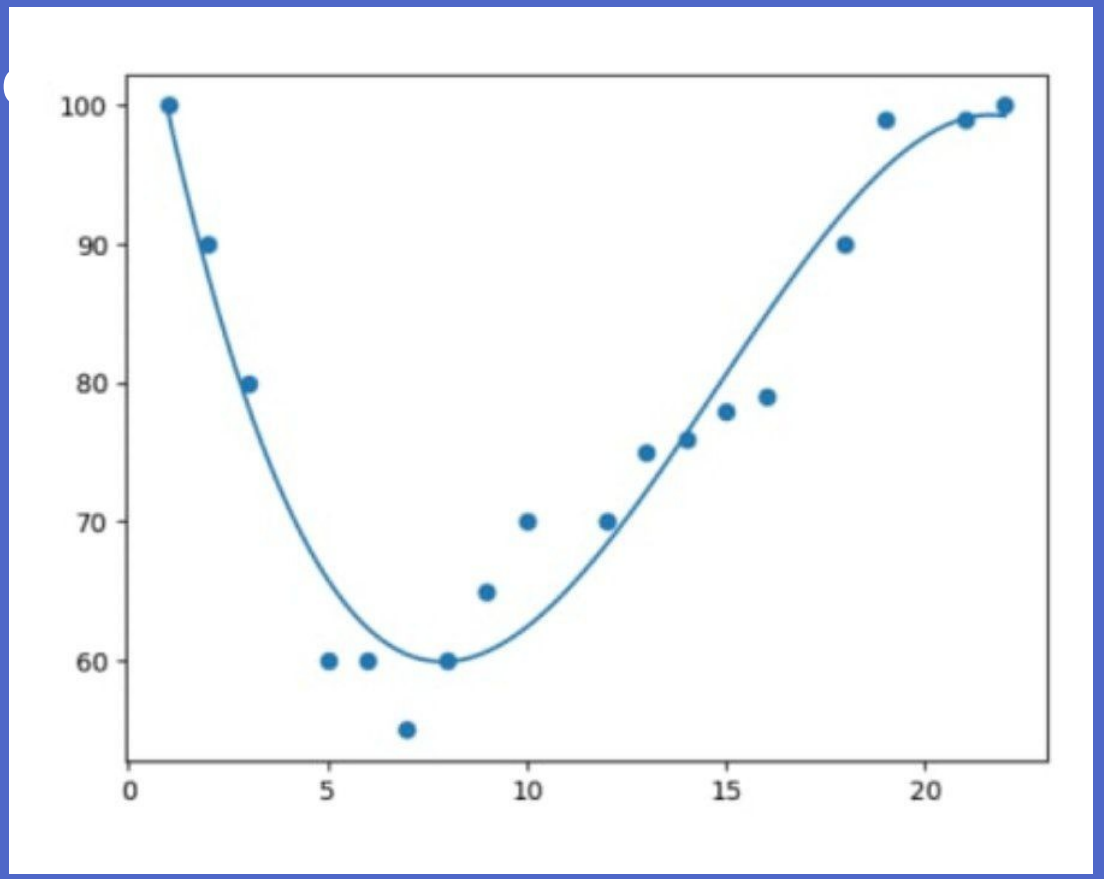
$\ldots$

}

3-Polynomial Regression (Poly equati

If your data points clearly will not fit a linear regression (a straight line through all data points), it might be ideal for polynomial regression.

Polynomial regression, like linear regression, uses the relationship between the variables x and y to find the best way to draw a line through the data points



Simple Linear Regression	$y = b_0 + b_1 x_1$
Multiple Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$
Polynomial Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$

Types of Polynomials

Linear —— $ax + b = 0$

Quadratic —— $ax^2 + bx + c = 0$

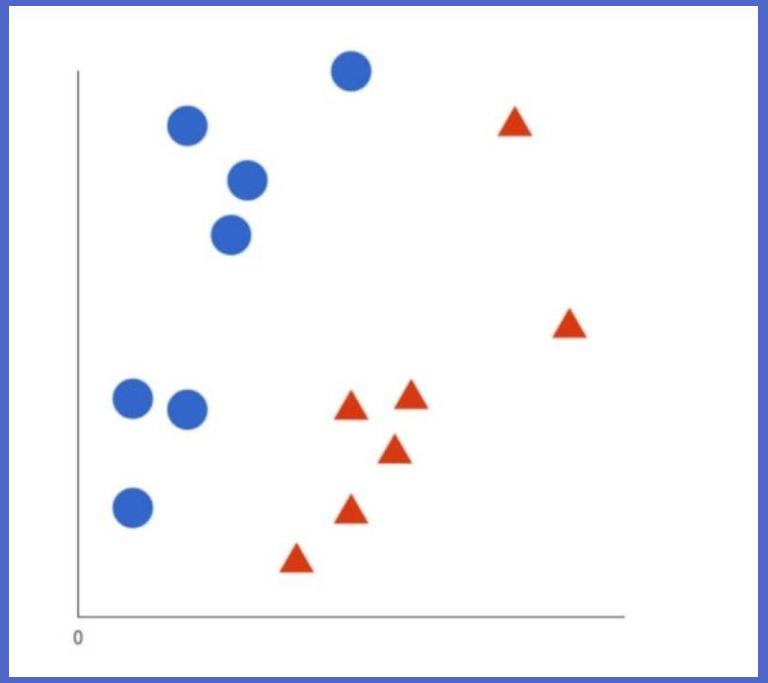
Cubic —— $ax^3 + bx^2 + cx + d = 0$

4- Support Vector Machine (SVR)

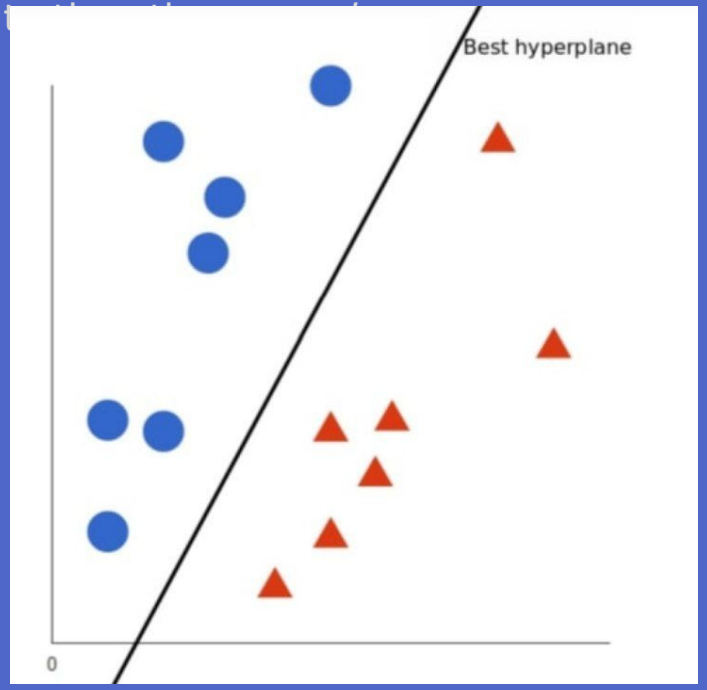
- A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

- they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands)

- Let's imagine we have two tags: *red* and *blue*, and our data has two features: *x* and *y*. We want a classifier that, given a pair of *(x,y)* coordinates, outputs if it's either *red* or *blue*.

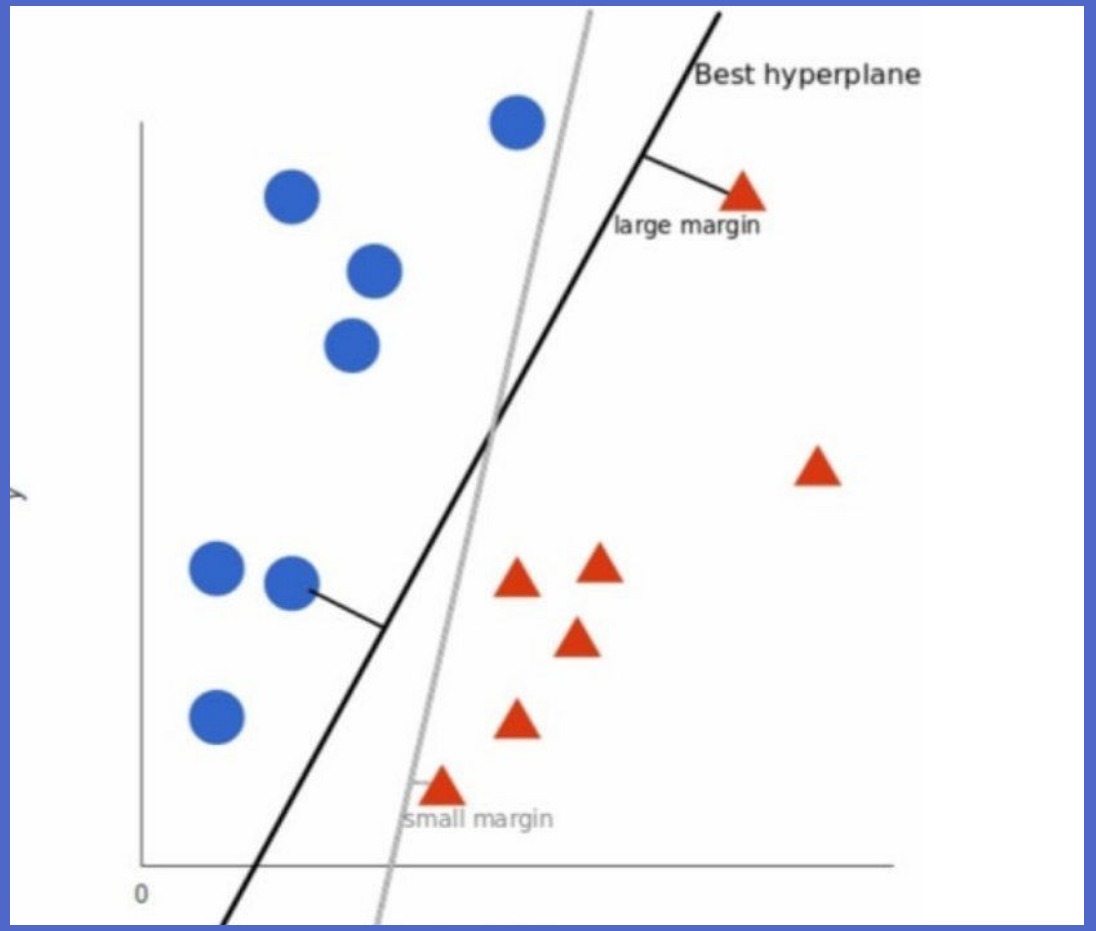


- A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the **decision boundary**: anything that falls to one side of it we will classify as *blue*, and anything that falls

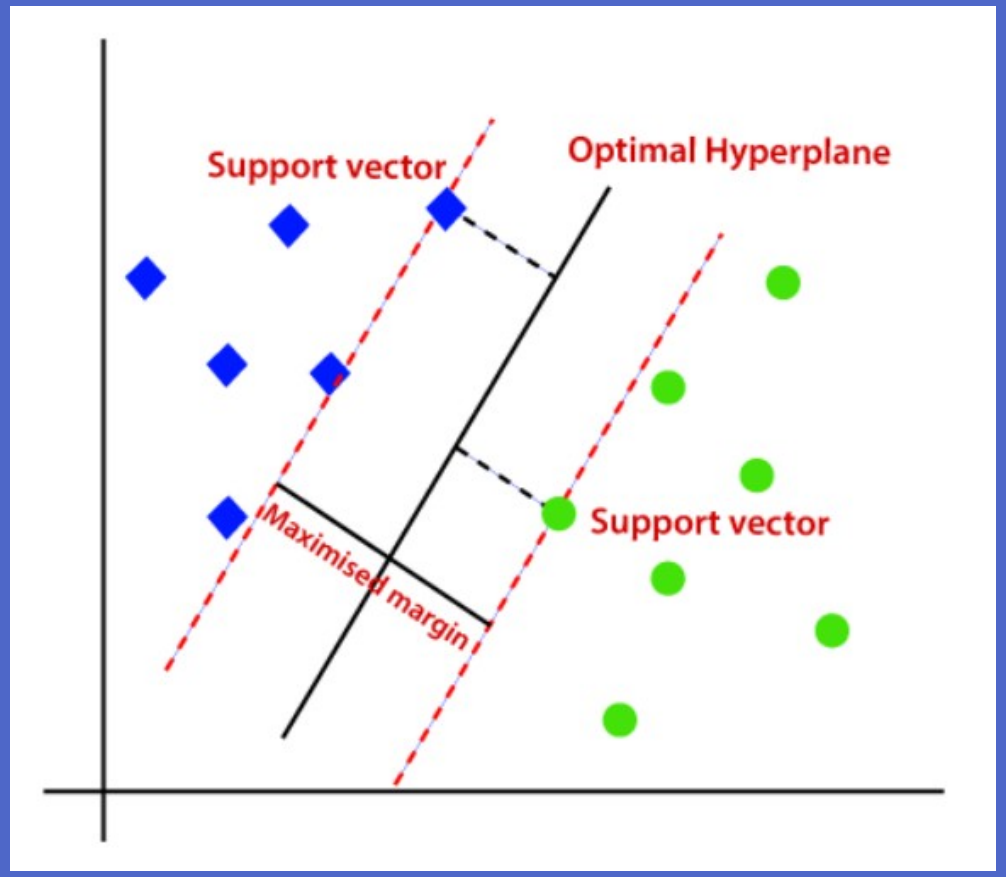


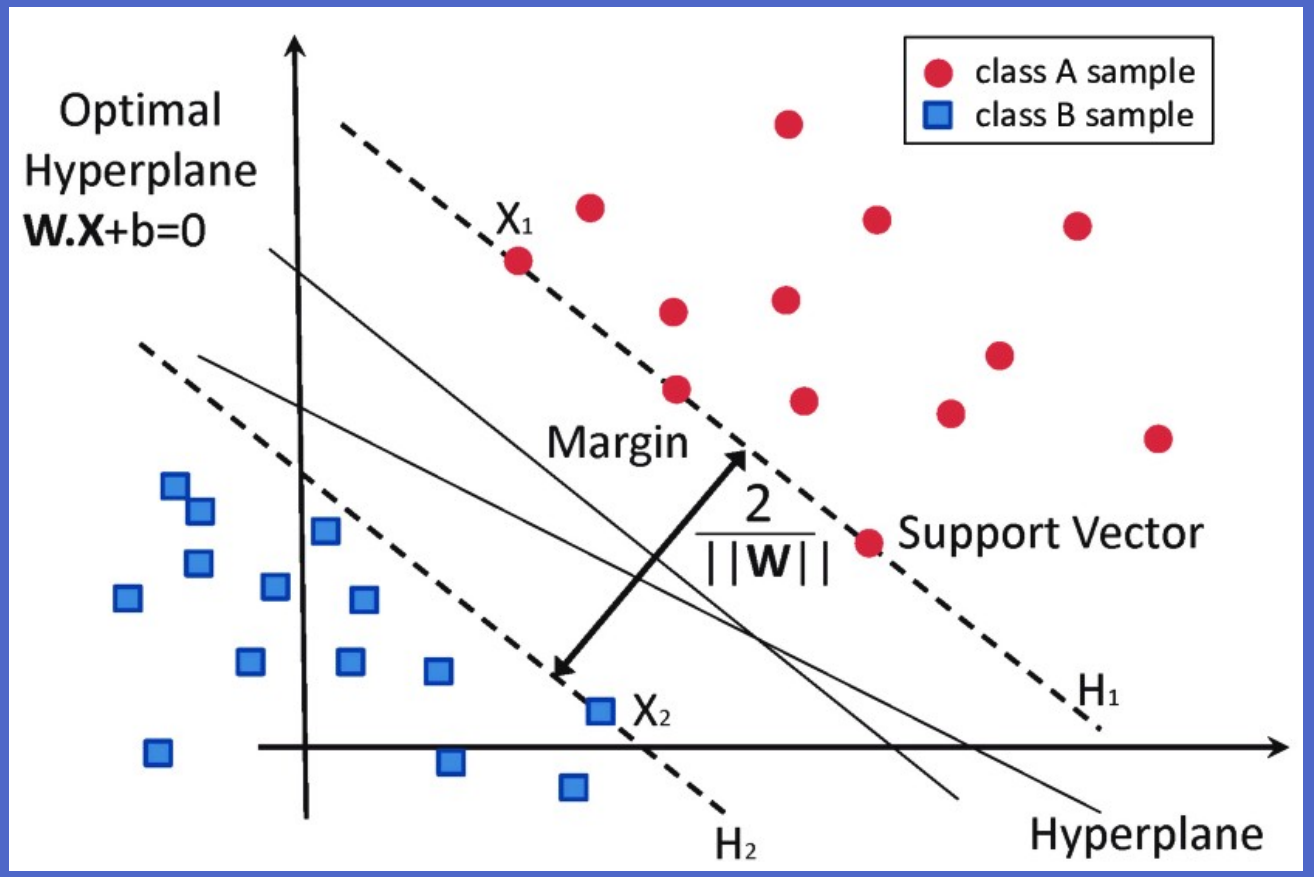
4- Support Vector Machine (SVR)

- But, what exactly is *the best* hyperplane? For SVM, it’s the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.

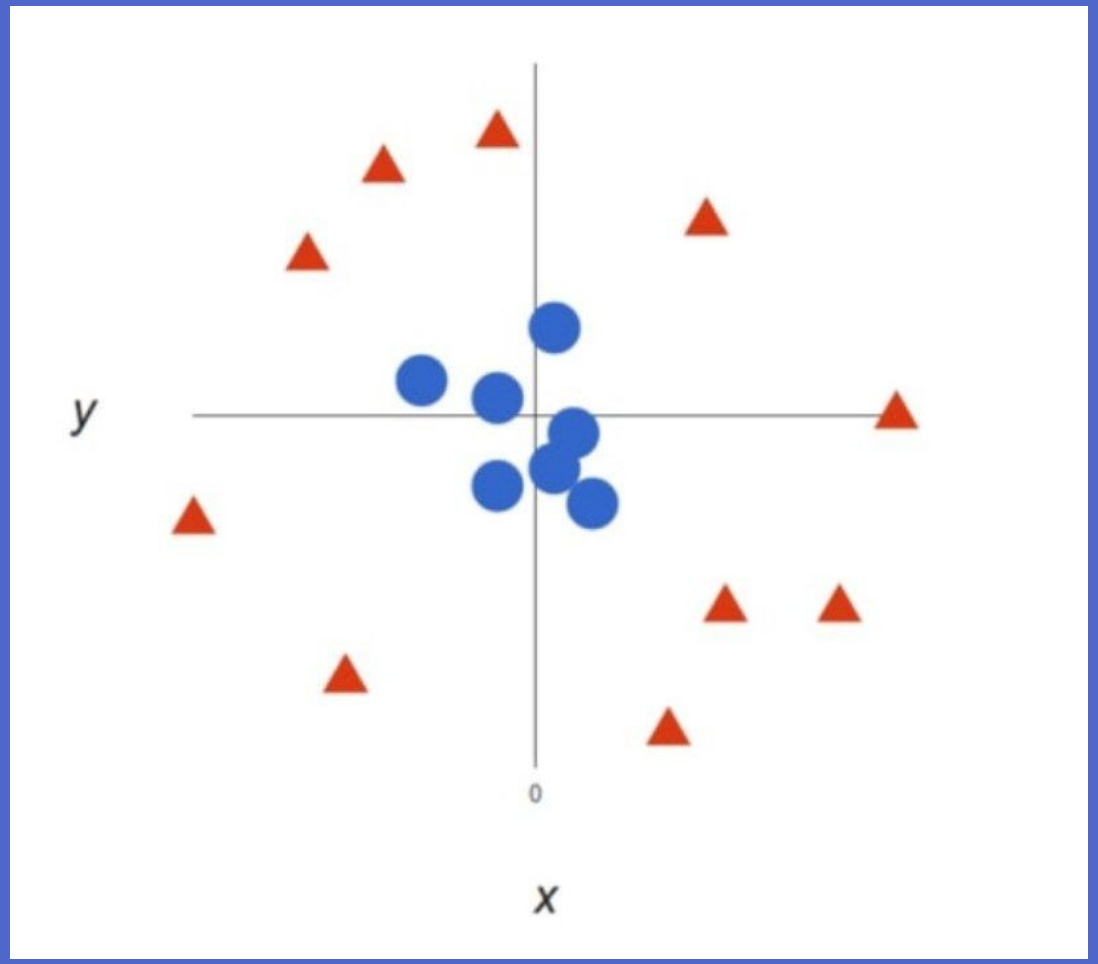


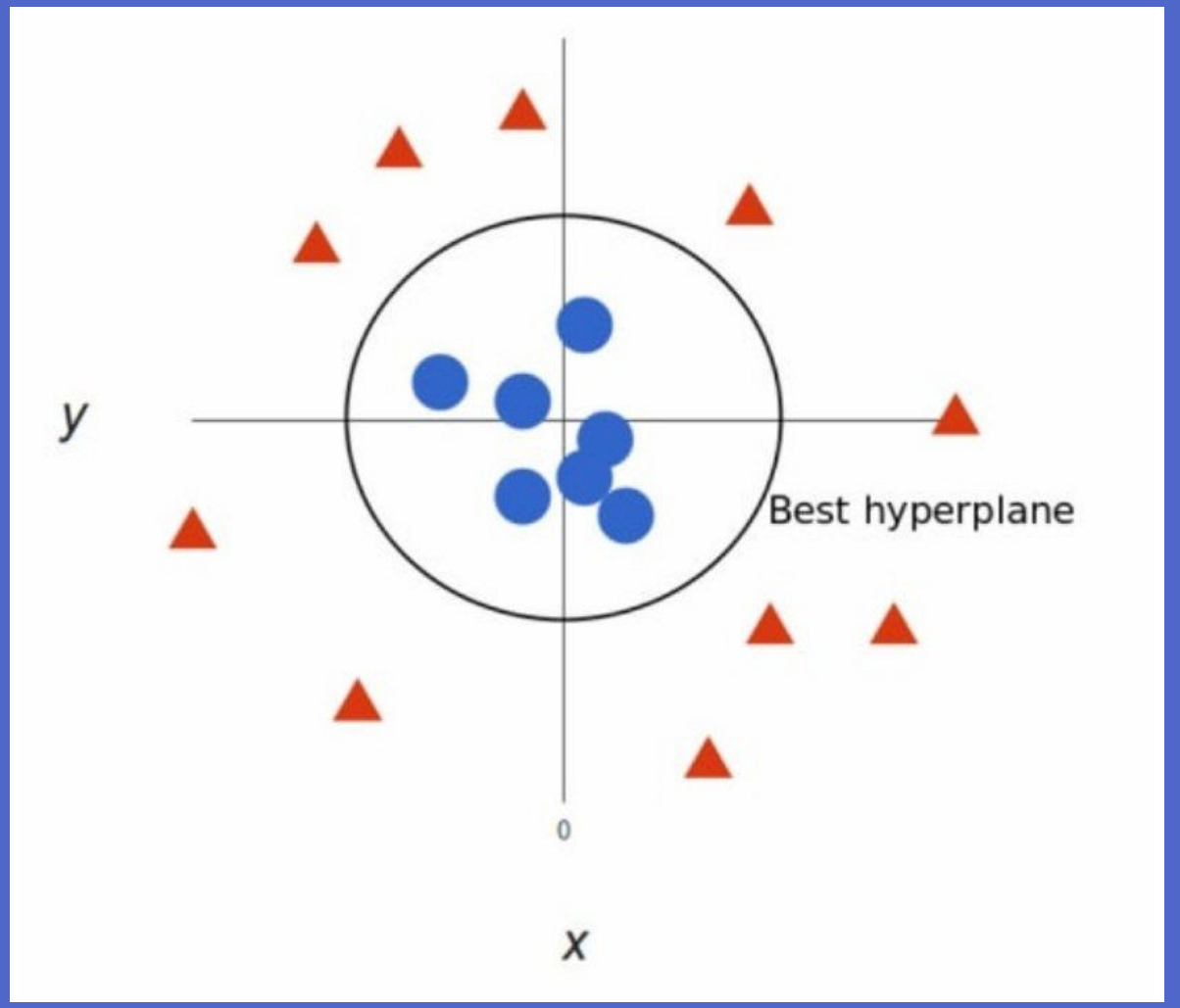
4- Support Vector Machine (SVR)



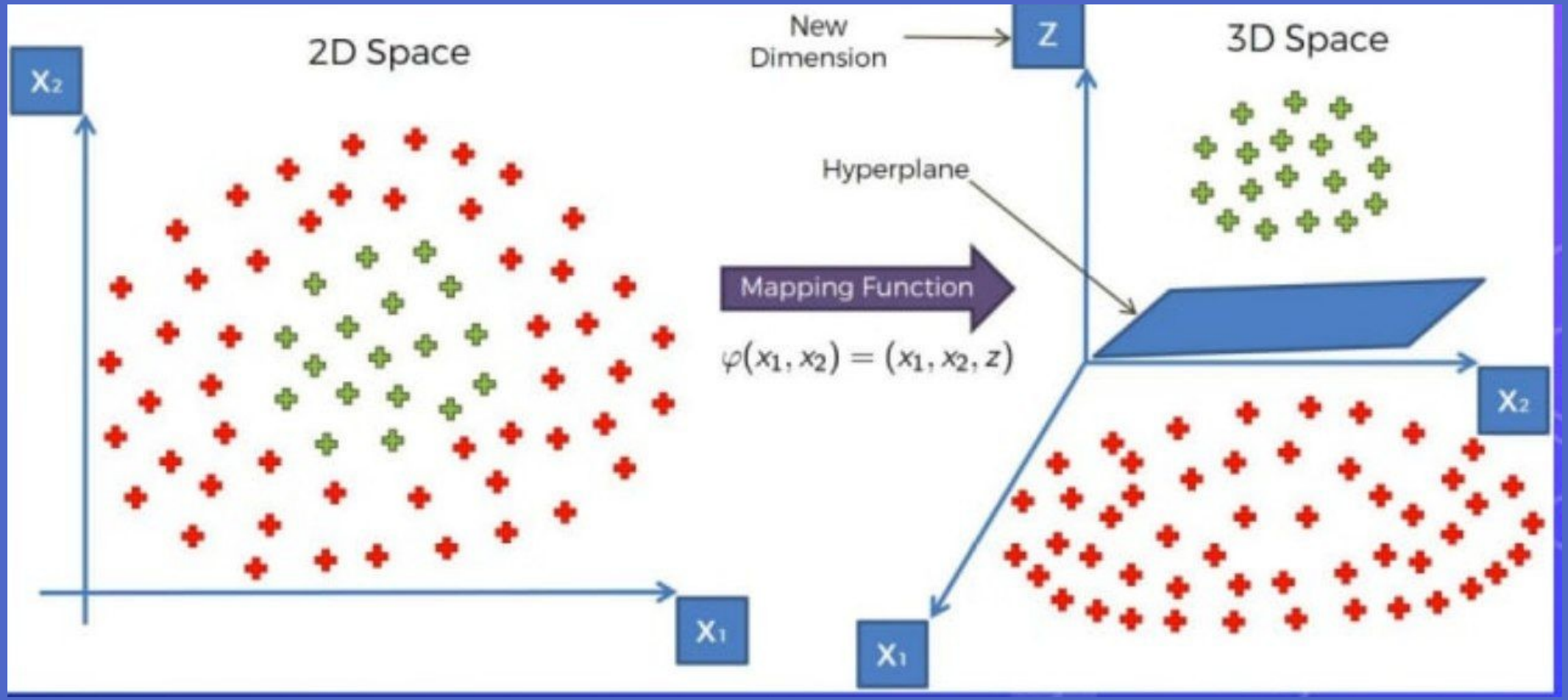


4- Support Vector Machine (SVR) “Not Linear”



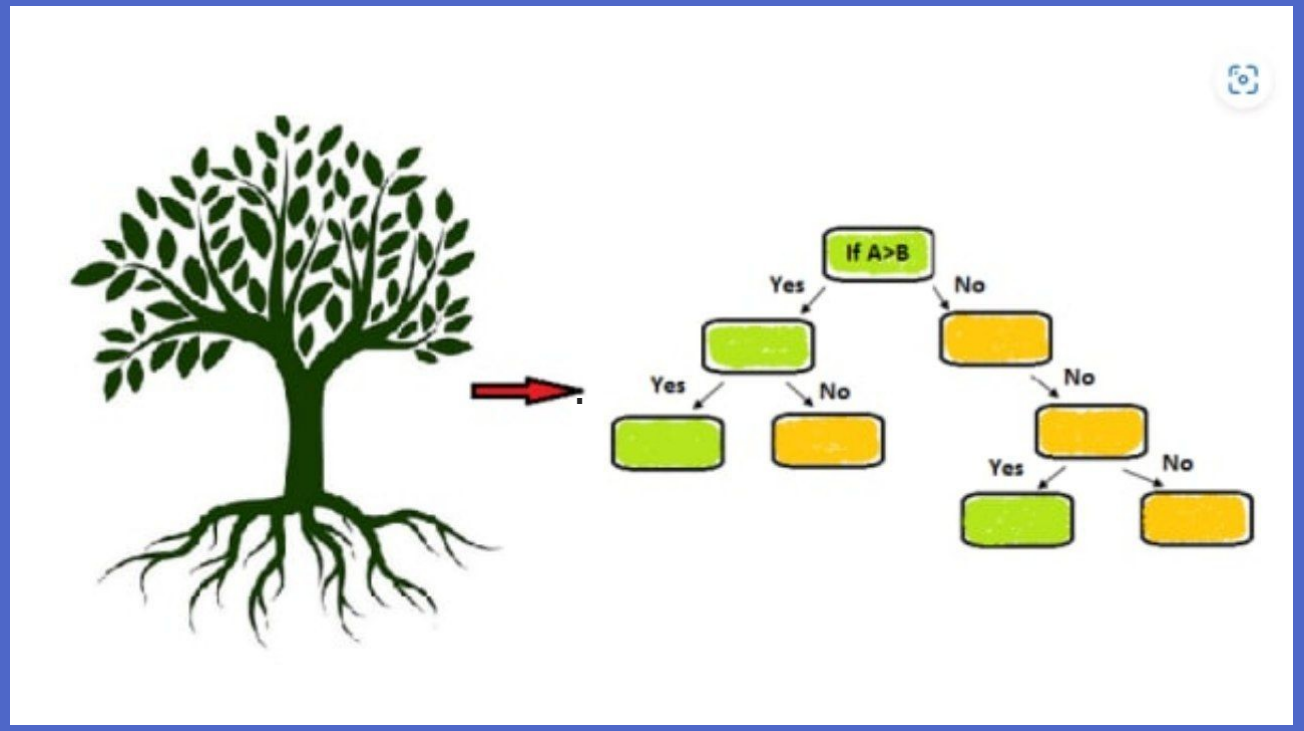


4- Support Vector Machine (SVR)

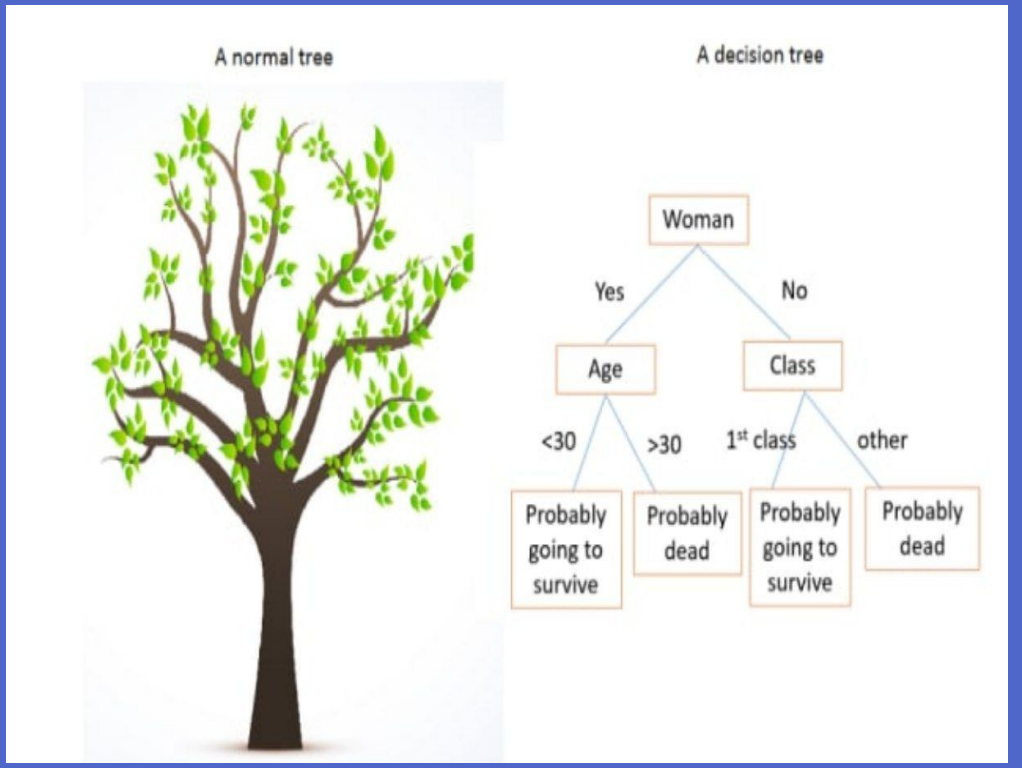


5-Decision Tree Regression

The Decision Tree Algorithm: Discover This Powerful Tool



- A decision tree is a graphical representation of possible solutions to a problem, where each node in the tree represents a possible decision, and the branches represent the possible consequences of each decision



5-Decision Tree Regression

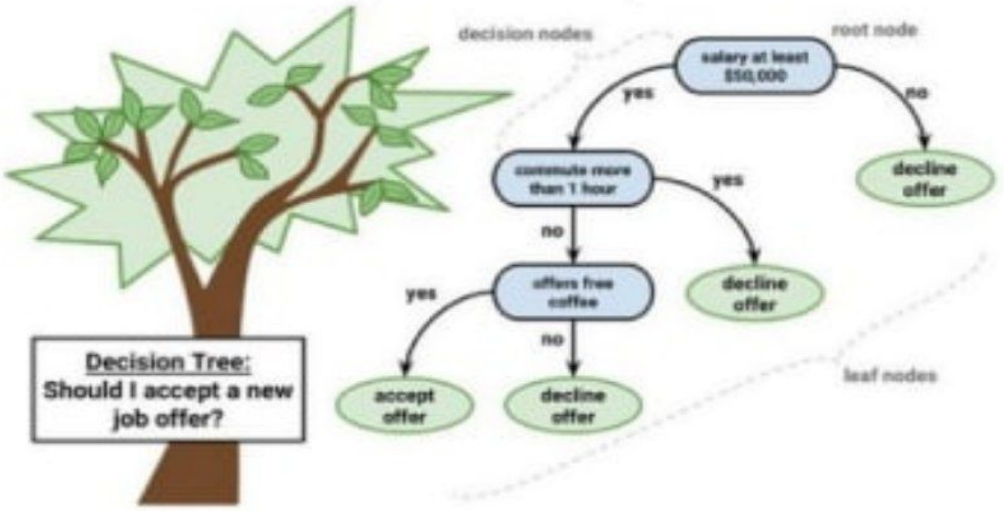
- **What is a Decision Tree Algorithm?**
  A decision tree algorithm is a tool that can be used to help make decisions by creating a tree-like structure of possible solutions. The algorithm starts with a root node, which represents the initial decision to be made, and then expands the tree by adding nodes for each possible decision until a final decision is reached.

- **How does a Decision Tree Algorithm work?**

Understanding decision trees

- **tree structure** to model the relationships among the features and the potential outcomes
  - branching decisions, which channel examples into a final predicted class value

E.g.,
- Root node
- Decision nodes – split the data across branches
- Leaf nodes (terminal nodes – the action to be taken or expected results

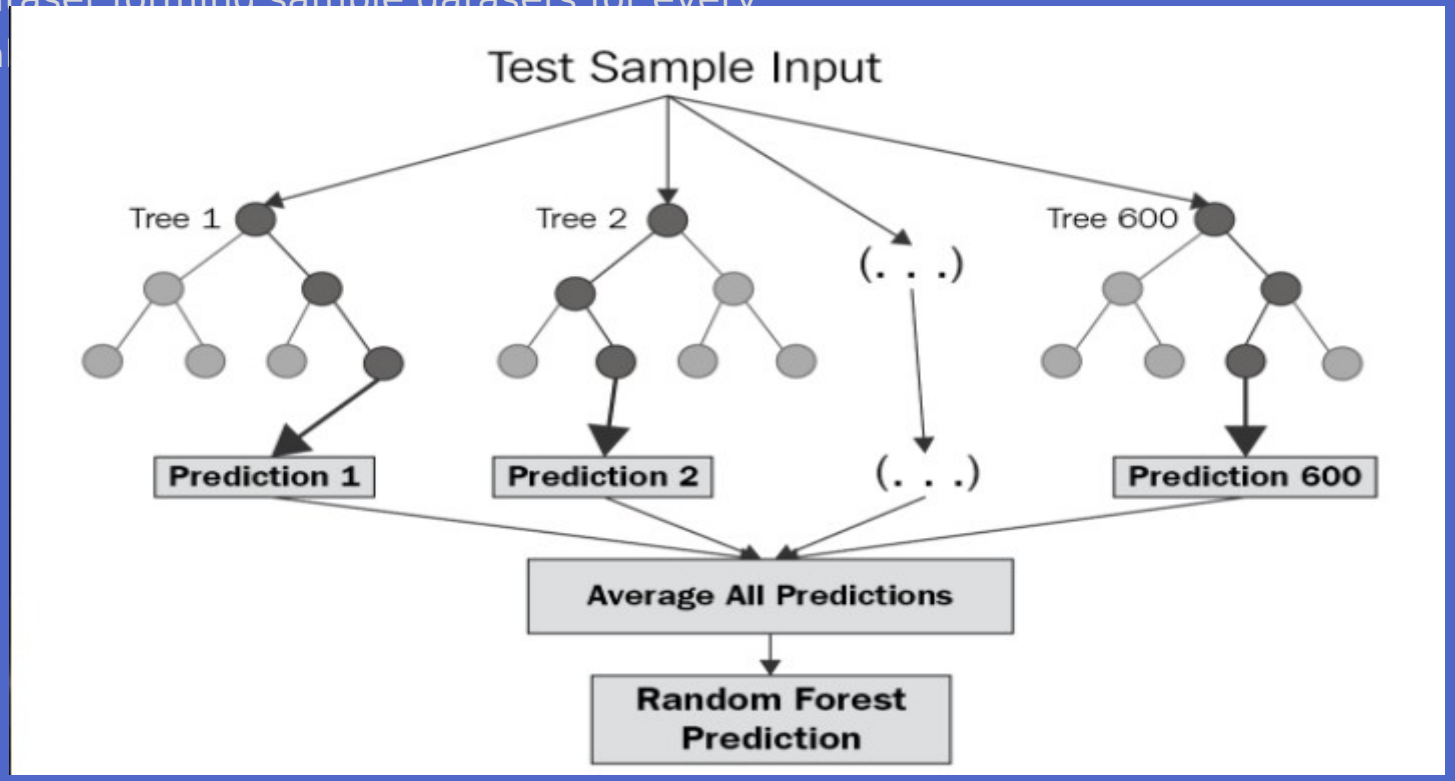


- The decision tree algorithm works by starting with a root node, which represents the initial decision to be made.
- The algorithm then expands the tree by adding nodes for each possible decision, until a final decision is reached.
- The algorithm is designed to help make decisions by creating a tree-like structure of possible solutions.

5- Random Forests

- Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

- Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is ca



6- Evaluating Regression Models Performance

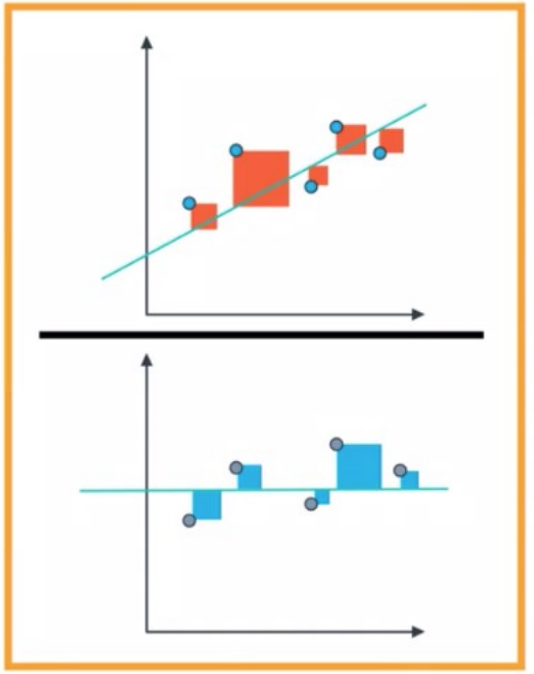
- **BAD MODEL**

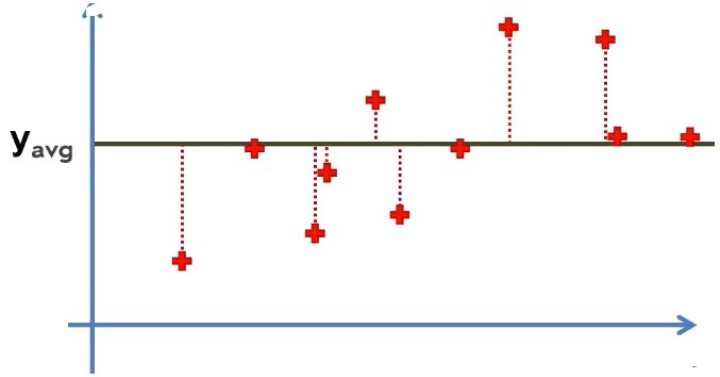
  The errors should be similar.
  R2 score should be close to 0.

- **GOOD MODEL**

  The mean squared error for the linear regression model should be a lot smaller than the mean squared error for the simple model.

$$R2 = 1 -$$





$$SS_{res} = SUM\ (y_i - \hat{y_i})^2$$

$$SS_{tot} = SUM\ (y_i - y_{avg})^2$$

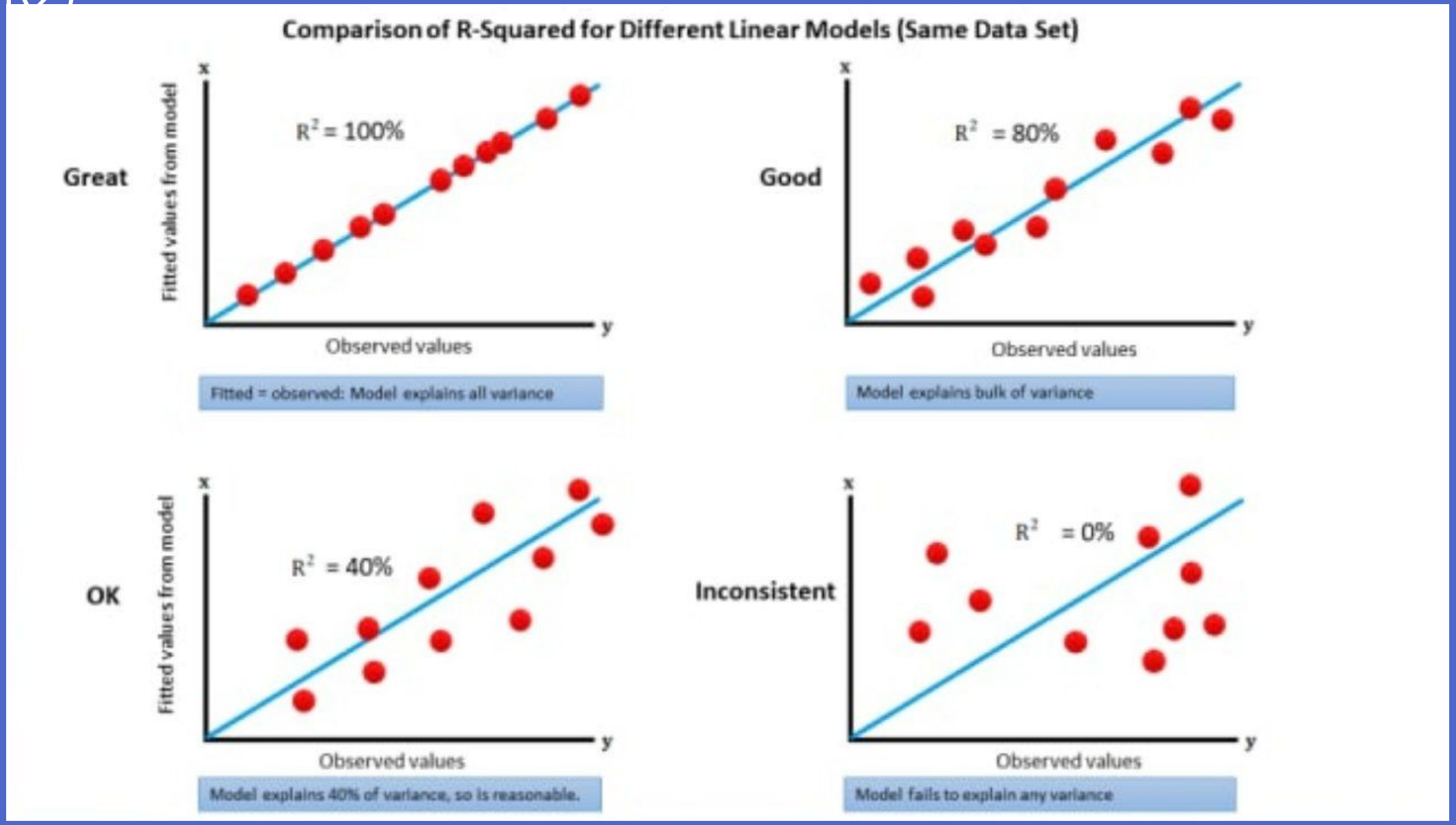
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

R2 Squared = $1 - \frac{SSr}{SSm}$

SSr = Squared sum error of regression line

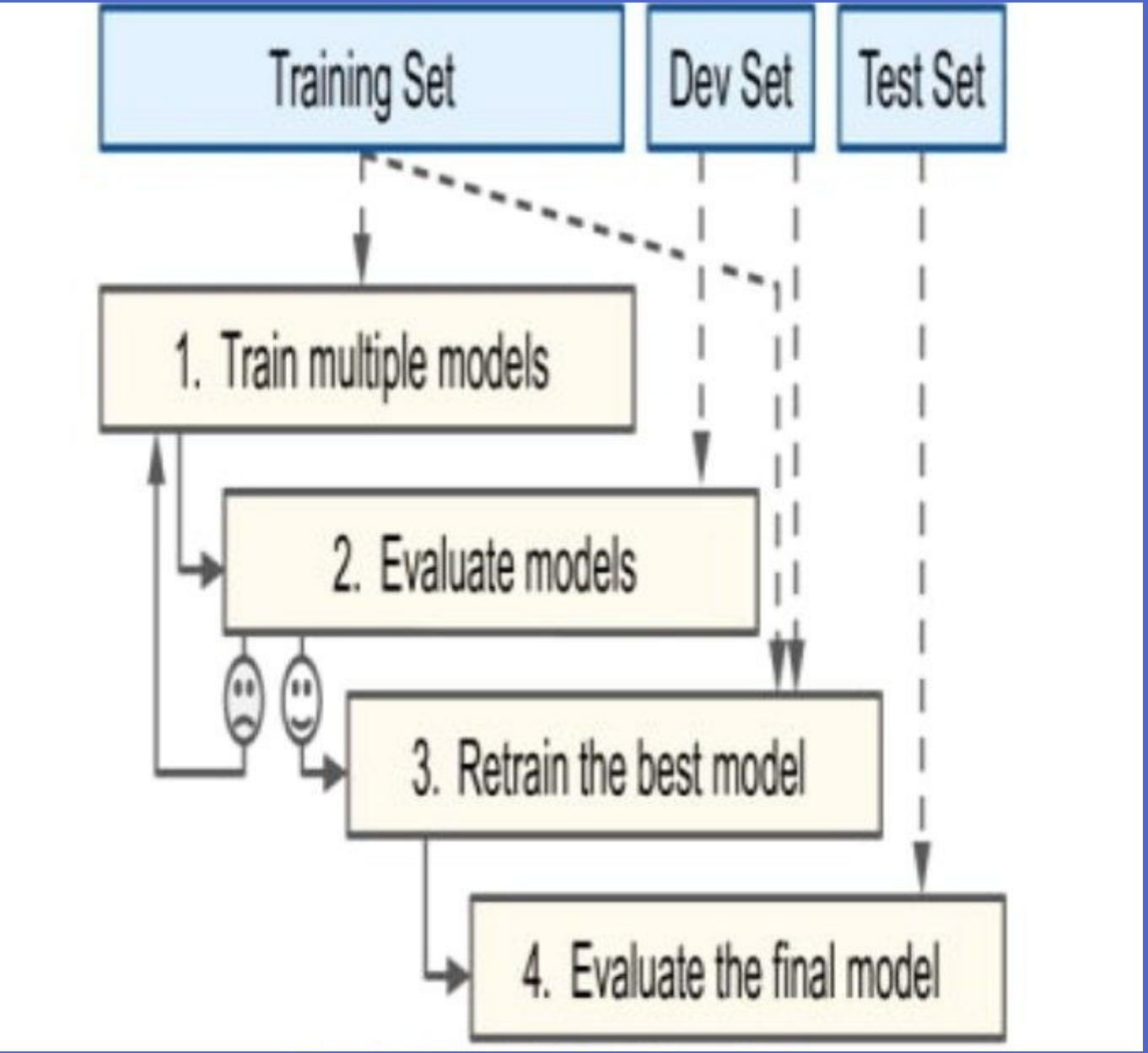
SSm = Squared sum error of mean line

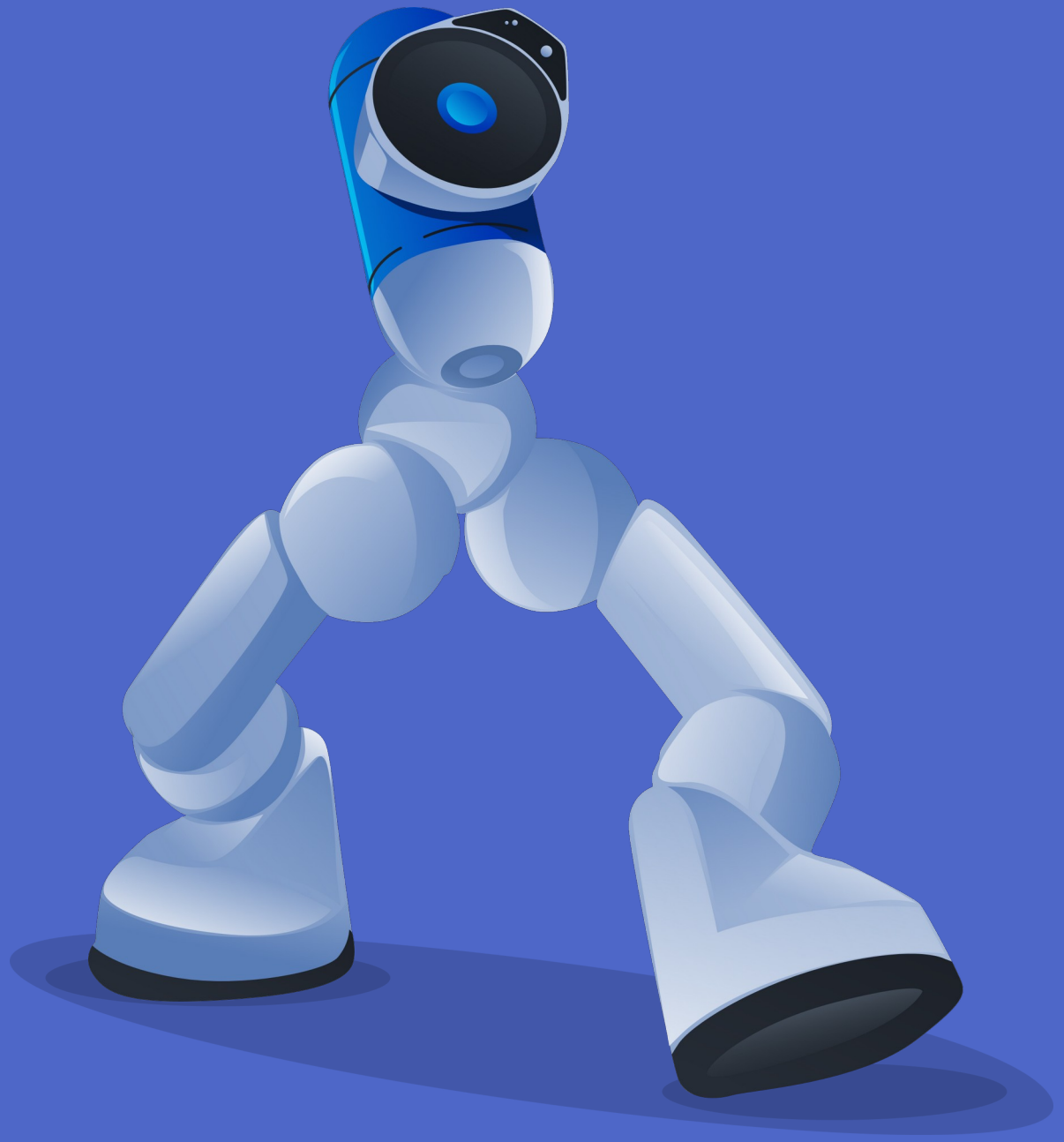
6- Evaluating Regression Models Performance (R2)



7- Regression Model Selection

- suppose you are hesitating between two types of models (say, a linear model and a polynomial model): how can you decide between them?
- One option is to train both and compare how well they generalize using the test set.
- A common solution to this problem is called holdout validation you simply hold out part of the training set to evaluate several candidate models and select the best one. The new held-out set is called the validation set (or the development set, or dev set). More specifically, you train multiple models with various hyperparameters on the reduced training set (i.e., the full training set minus the validation set), and you select the model that performs best on the validation set. After this holdout validation process, you train the best model on the full training set (including the validation set), and this gives you the final model. Lastly, you evaluate this final model on the test set to get an estimate of the generalization error





THANK YOU

Khaled Tarek