



Artificial Intelligence With TensorFlow

ML with Scikit-learn and TensorFlow

Supervised Learning - Regression & Classification

Agenda

- Introduction to Regression & Classification
- Linear Regression
- Multi Linear Regression
- Polynomial Regression
- Logistic Regression
- Decision Tree Classification
- K-nearest Neighbors
- Naïve-Bayes
- Support Vector Machine



Introduction to Regression & Classification

Regression is a type of supervised learning algorithm that is used when the target variable is continuous. The goal of regression is to establish a relationship between the input features and the target variable.

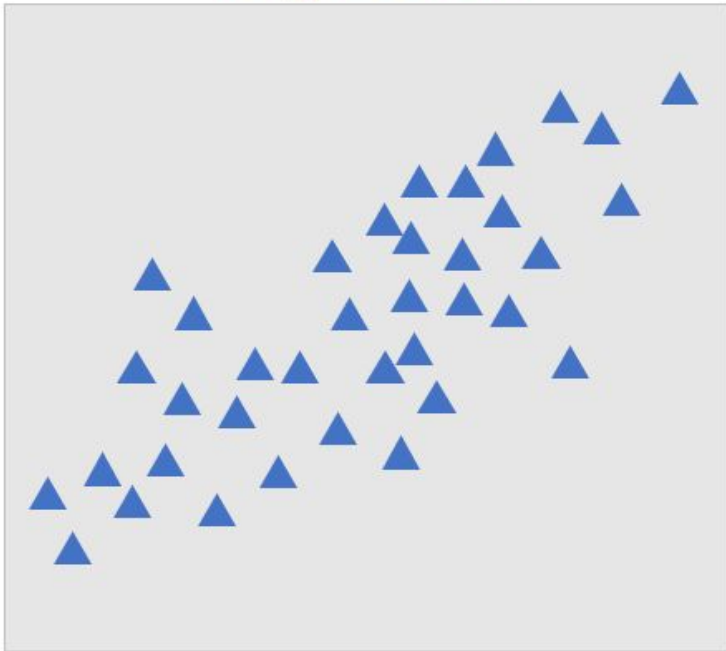
- Eg 1: Predicting house prices based on features such as square footage, number of bedrooms, ...
- Eg 2: Estimating the temperature based on factors like humidity, wind speed, ...

Classification is another type of supervised learning algorithm used when the target variable is categorical. The goal is to learn a mapping from input features to a discrete class label.

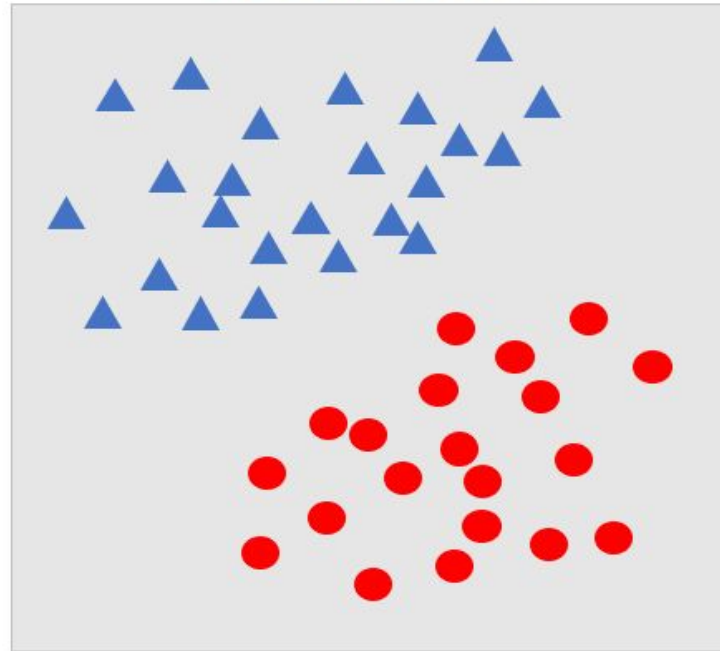
- Eg 1: Identifying spam emails based on content and metadata.
- Eg 2: Predicting whether a customer will churn or not based on their behavior.

Regression vs Classification

Regression



Classification



Simple Linear Regression

Linear Regression (Simple LR)

Simple linear regression is a statistical method used to model the relationship between a single independent variable (X) and a dependent variable (Y).

The relationship is represented by a linear equation:

- $Y = mX + b$
- Y is the dependent variable (the one you are trying to predict),
- X is the independent variable (the one used to make predictions),
- m is the slope of the line, and
- b is the y-intercept.

Linear Regression (Simple LR)

$X = [2, 4, 6, 8, 10]$ (Hours studied)

$Y = [50, 60, 65, 80, 85]$ (Test scores)

Step 1: Calculate Mean

Calculate the mean of X and Y :

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \quad \text{and} \quad \bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$$

$$\bar{X} = \frac{2+4+6+8+10}{5} = 6$$

$$\bar{Y} = \frac{50+60+65+80+85}{5} = 68$$

Step 2: Calculate Slope (m) and Y-Intercept (b)

The slope (m) is given by:

$$m = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

The y-intercept (b) is given by:

$$b = \bar{Y} - m \cdot \bar{X}$$

Calculate m and b :

$$m = \frac{(2-6)(50-68) + (4-6)(60-68) + \dots + (10-6)(85-68)}{(2-6)^2 + (4-6)^2 + \dots + (10-6)^2}$$

$$m \approx \frac{-16+16+6+48+34}{16+4+4+4+16} \approx \frac{88}{44} = 2$$

$$b = \bar{Y} - m \cdot \bar{X} = 68 - 2 \cdot 6 = 56$$

Step 3: Build the Linear Regression Model

The linear regression model is given by $Y = mX + b$. So, our model is $Y = 2X + 56$.

Step 4: Make Predictions

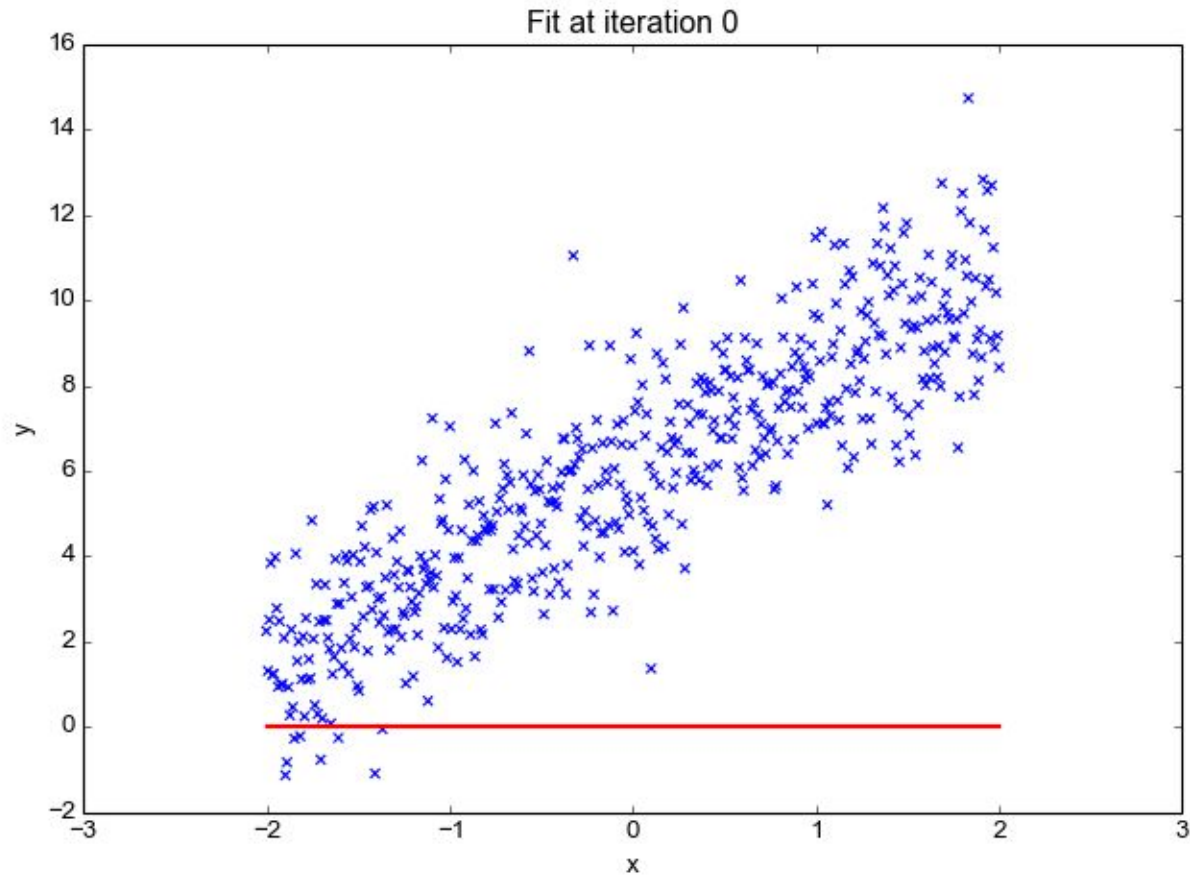
Let's predict the test score for a student who studied for 5 hours ($X = 5$):

$$Y_{\text{predicted}} = 2 \cdot 5 + 56 = 66$$

Summary:

The linear regression model for our data is $Y=2X+56$. If a student studies for 5 hours, the predicted test score is 66.

Linear Regression (Simple LR)



Multiple Linear Regression

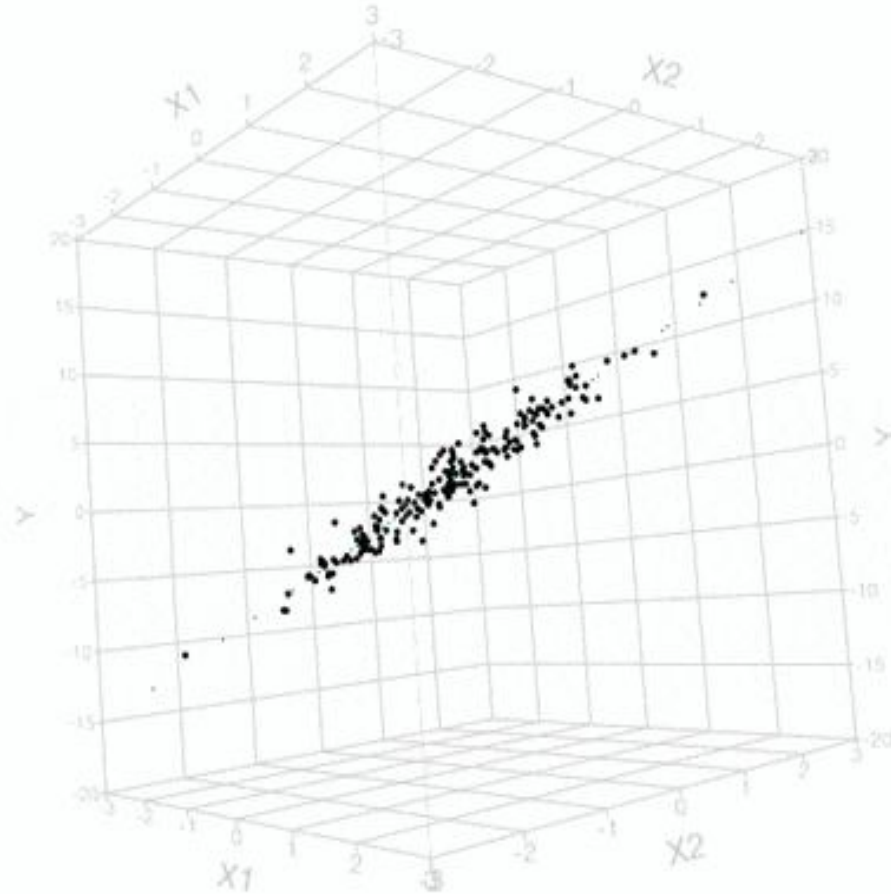
Linear Regression (Multilinear regression)

Multilinear regression, also known as multiple linear regression, is a statistical method used to model the relationship between two or more independent variables and a dependent variable. The general form of the multilinear regression model is:

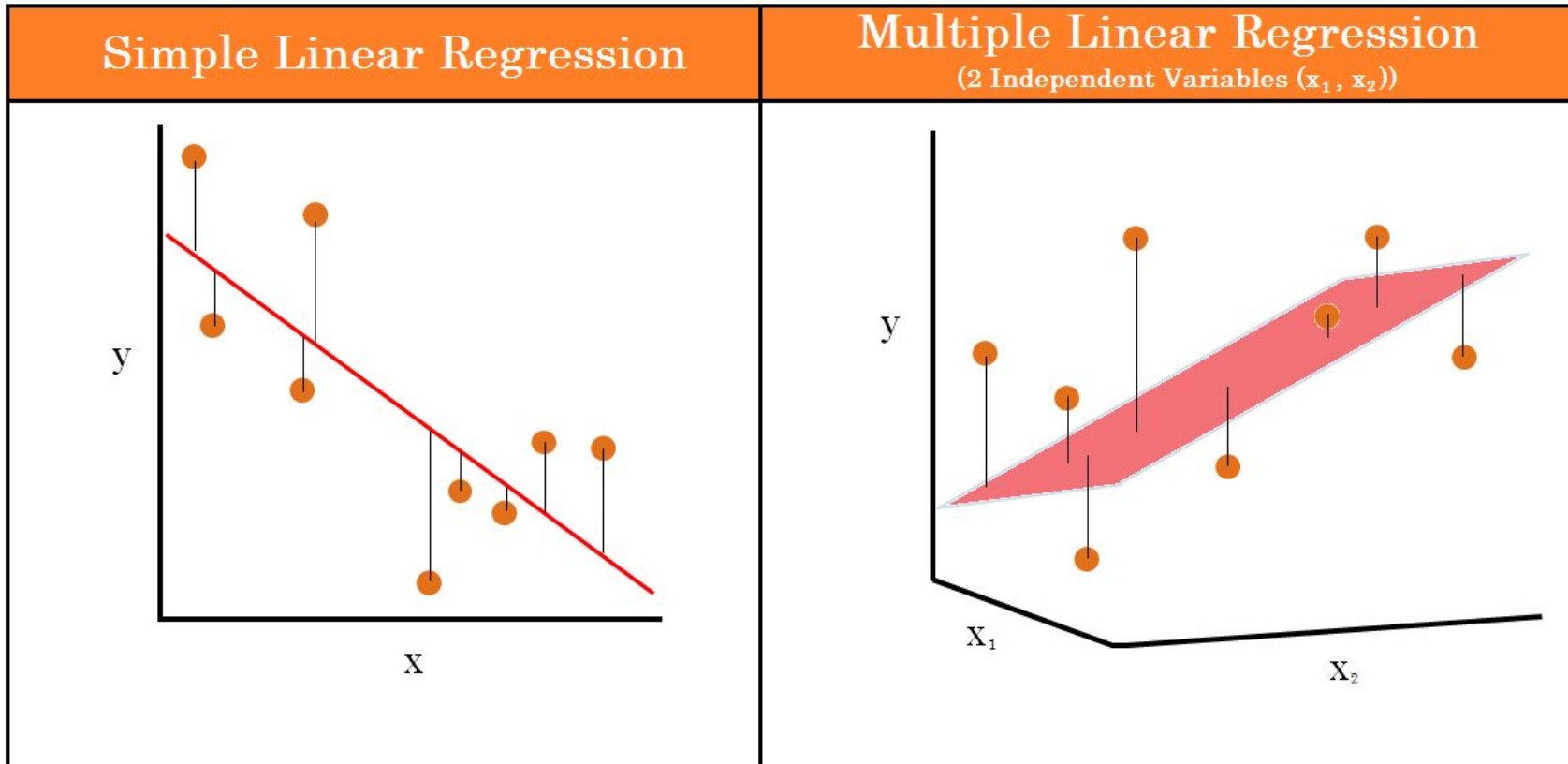
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon$$

- Y is the dependent variable.
- X_1, X_2, \dots, X_n are the independent variables.
- β_0 is the intercept (the value of Y when all X variables are zero).
- $\beta_1, \beta_2, \dots, \beta_n$ are the coefficients (representing the change in Y for a one-unit change in the corresponding X variable).
- ε is the error term, representing the unobserved factors that affect Y but are not accounted for in the model.

Linear Regression (Multilinear regression)



SLR vs MLR



Polynomial Regression

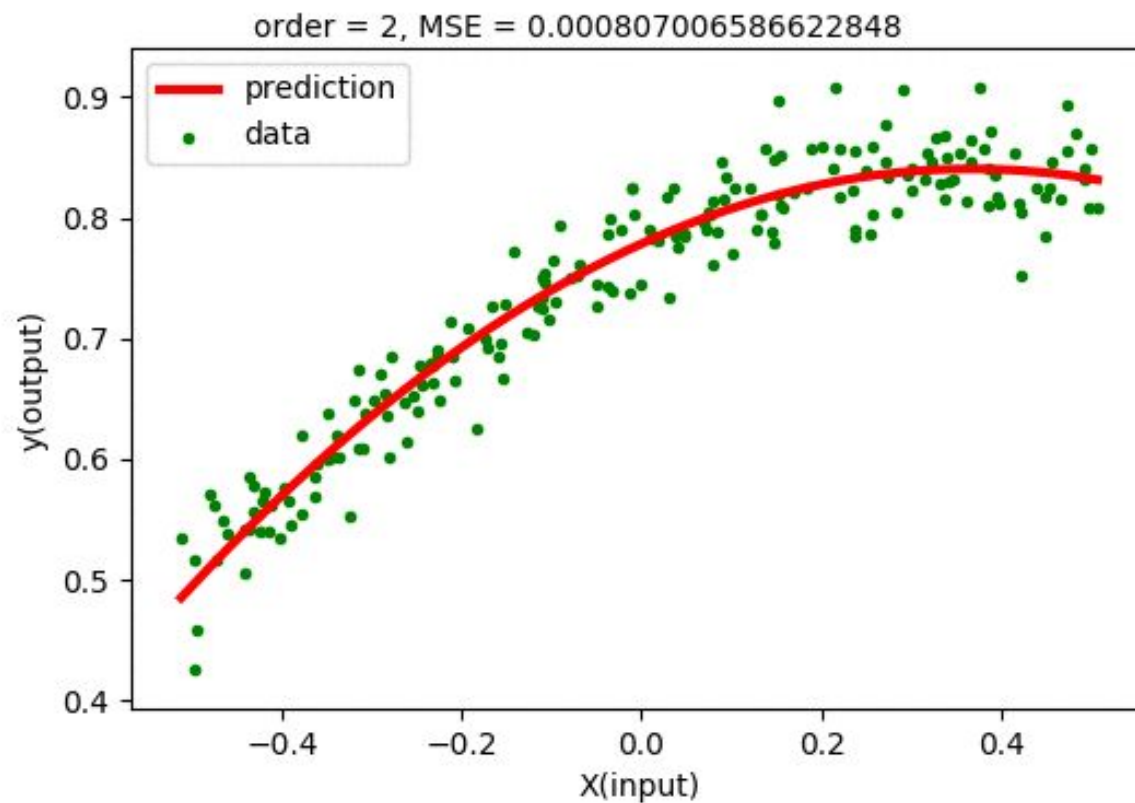
Polynomial regression

- Polynomial regression is a type of regression analysis that models the relationship between the independent variable (X) and the dependent variable (Y) as an n -th degree polynomial. It extends the idea of linear regression, allowing for more complex relationships between the variables.
- The polynomial regression equation has the form:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_n X^n + \varepsilon$$

- Y is the dependent variable (the variable being predicted).
- X is the independent variable.
- $\beta_0, \beta_1, \dots, \beta_n$ are the coefficients.
- n is the degree of the polynomial, determining the complexity of the model.
- ε is the error term.

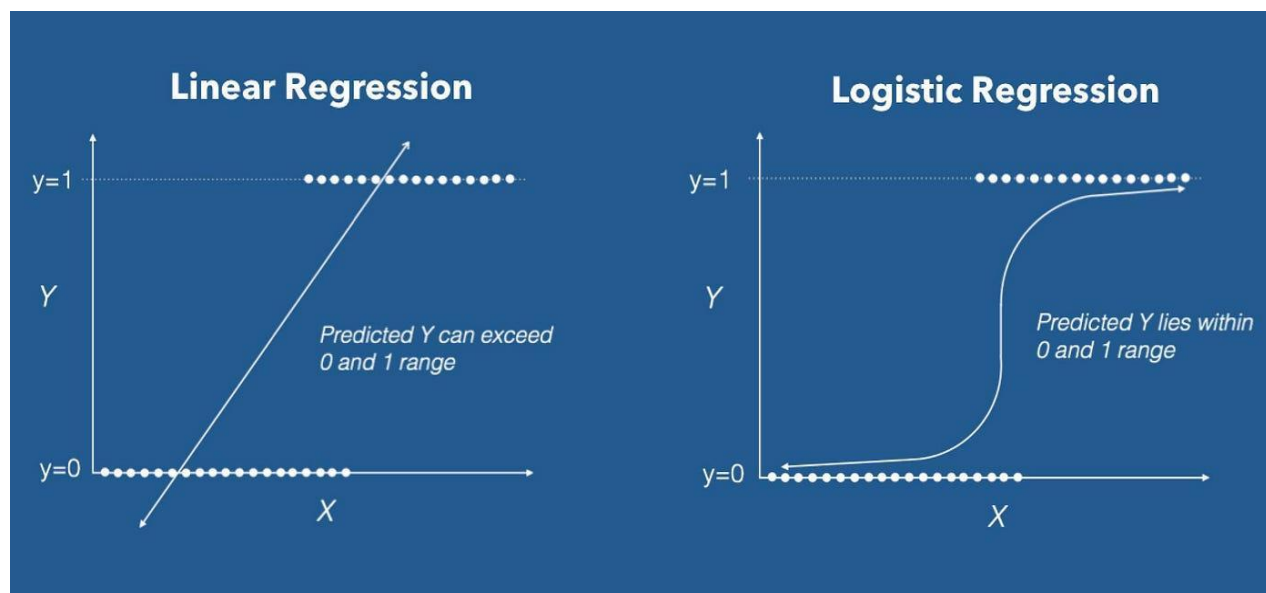
Polynomial regression



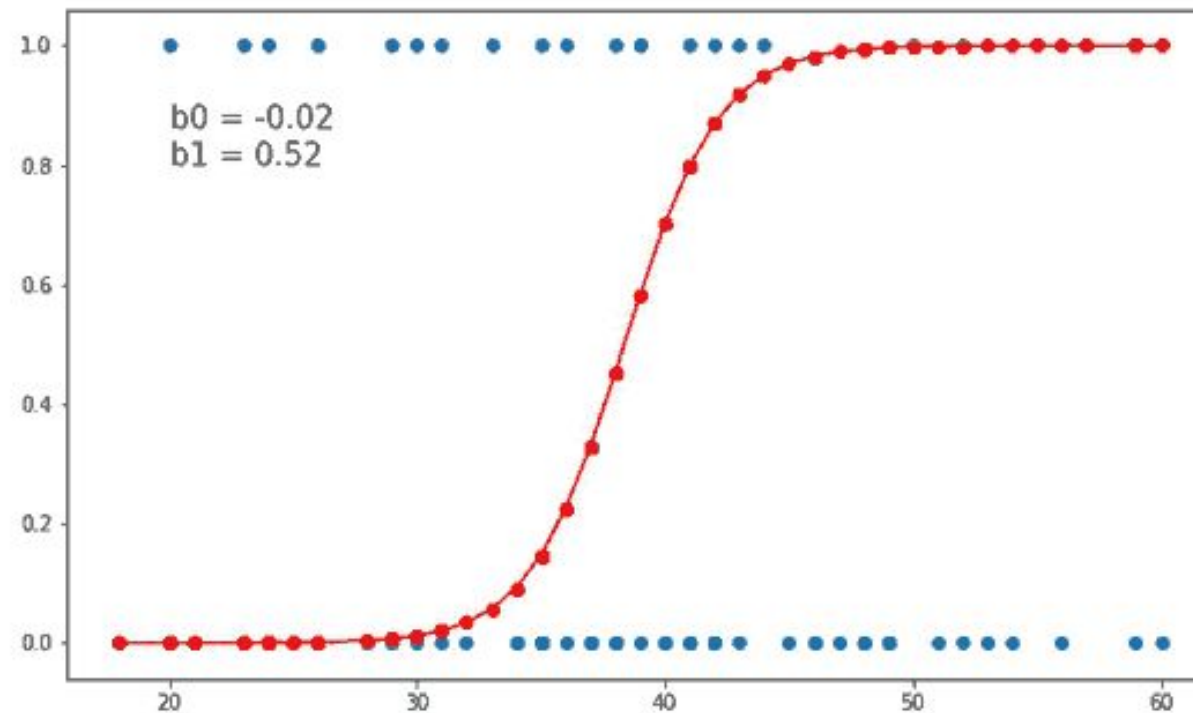
Logistic Regression

Logistic regression

Logistic regression is a statistical method used for modeling the probability of a binary outcome.



Logistic regression



Decision Tree Classification

Decision Tree

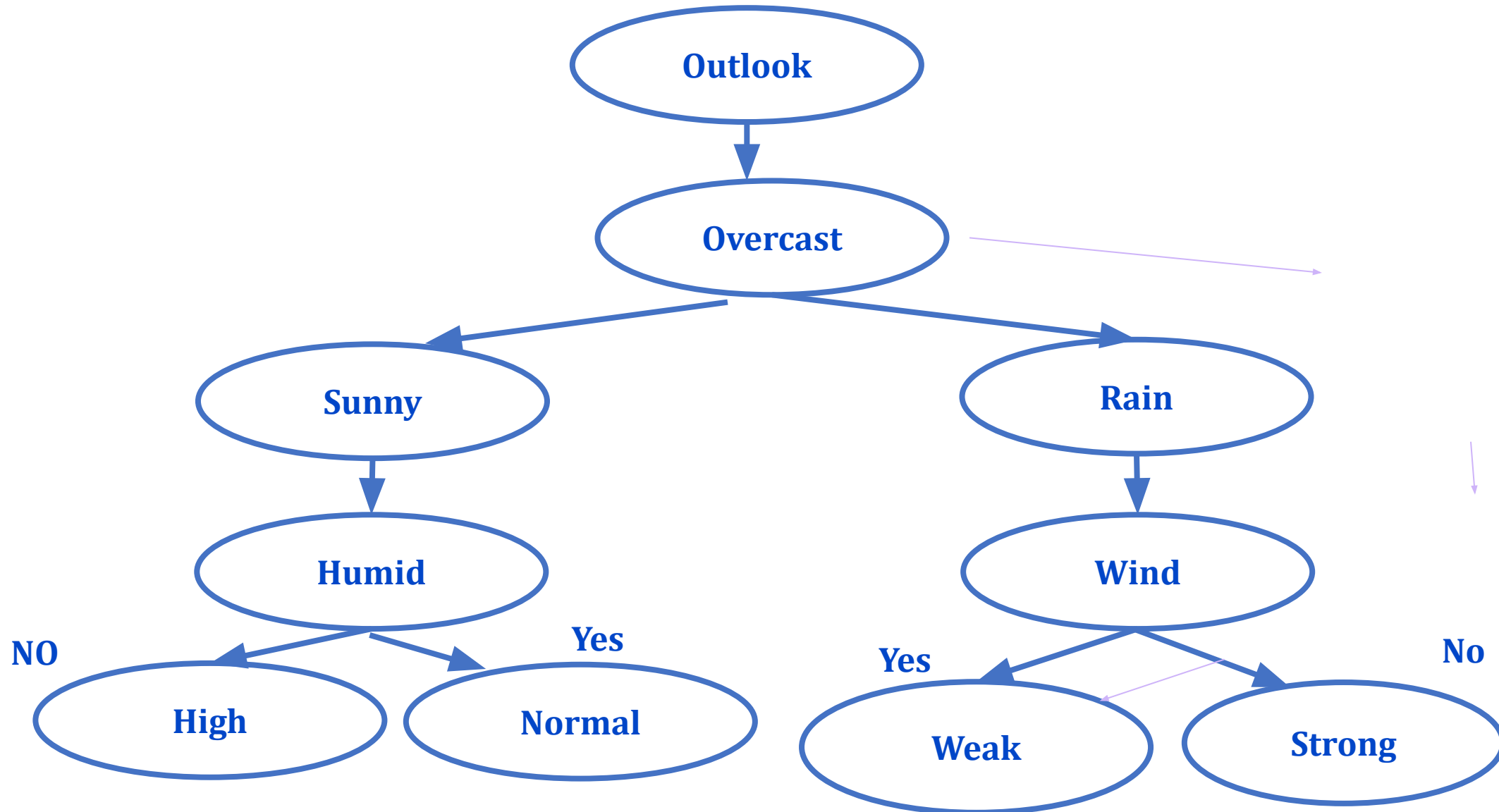


Decision Tree

A Decision Tree is a supervised machine learning algorithm that partitions the feature space into segments, or regions, based on a series of binary decisions. It is represented as a tree-like structure, where each internal node represents a decision based on a feature, each branch represents the outcome of that decision, and each leaf node represents the final classification or regression output.

- Inspire by human Decision Making
- Decision tree Classification and Regression.
- Advantages of Decision Tree
- Disadvantages of Decision Tree
- Mathematics

Decision Tree (Human Decision Making)



Decision Tree Regression And Classification

spect	Regression Trees	Classification Trees
Output	Predicts continuous numerical values	Predicts categorical class labels
Target variable type	Continuous	Discrete
Objective	Minimize mean squared error (MSE)	Maximize purity (e.g., Gini impurity, entropy)
Splitting criterion	Based on reduction in variance (e.g., MSE, RMSE)	Based on purity gain (e.g., Gini impurity, entropy)
Evaluation metric	Typically MSE, RMSE	Gini impurity, entropy
Prediction mechanism	Average value of target variable in leaf nodes	Majority class in leaf nodes
Visualization	Continuously changing splits with numerical values	Discrete splits based on feature categories
Example	Predicting house prices	Classifying spam vs. non-spam emails
Example algorithm	CART (Classification and Regression Trees)	CART (Classification and Regression Trees)

Decision Tree Advantages and Disadvantages

- **Advantages**

- 1. Interpretability:

- 1. Decision trees are easily interpretable, making them useful for understanding the decision-making process and communicating results to non-experts.

- 2. Handling Non-linear Relationships:

- 1. Decision trees can capture non-linear relationships between features and the target variable without requiring complex transformations.

- 3. Handling Mixed Data Types:

- 1. Decision trees can handle both numerical and categorical data without the need for feature scaling or one-hot encoding.

- **Disadvantages**

- 1. Overfitting: (Low Bias and High Variance)

- 1. Decision trees are prone to overfitting, especially when the tree depth is not properly controlled or when the dataset is small.

- 2. Instability:

- 1. Small changes in the data can lead to a significantly different decision tree, making them unstable models.

- 3. Bias towards Features with Many Levels:

- 1. Features with many levels may be favored by decision trees, leading to biased splits and potentially inaccurate predictions.



diameter size

colour

Mathematics: (How to choose Best Feature for Split)

Day	Outlook	Humid	Wind	Play
1	Sunny	High	weak	No
2	Sunny	High	Strong	No
3	Overcast	High	Weak	Yes
4	Rain	High	Weak	Yes
5	Rain	Normal	Weak	Yes
6	Rain	Normal	Strong	No
7	Overcast	Normal	Strong	Yes
8	Sunny	High	Weak	No
9	Sunny	Normal	Weak	Yes
10	Rain	Normal	Weak	Yes
11	Sunny	Normal	Strong	Yes
12	Overcast	High	Strong	Yes
13	Overcast	Normal	Weak	Yes
14	Rain	High	Strong	No

Entropy Calculation:

Entropy is calculated using the formula:

$$E(S) = - \sum_{i=1}^c p_i \log_2(p_i)$$

where c is the number of classes and p_i is the probability of class i in the dataset S .

1. Calculate Entropy of the Target Variable (Play Tennis):

The formula for entropy is:

$$Entropy(S) = - \sum_{i=1}^c p_i \log_2(p_i)$$

Where S is the set of all samples, c is the number of classes, and p_i is the proportion of samples in class i .

Calculate the entropy of the entire dataset:

- Number of samples (total): 14
- Number of 'Yes' (Play Tennis) samples: 9
- Number of 'No' (Don't Play Tennis) samples: 5

$$Entropy(S) = - \left(\frac{9}{14} \log_2 \left(\frac{9}{14} \right) + \frac{5}{14} \log_2 \left(\frac{5}{14} \right) \right)$$

$$Entropy(S) \approx - (0.94 \times -0.76 + 0.36 \times -2.32)$$

$$Entropy(S) \approx -(-0.72 + -0.84)$$

$$Entropy(S) \approx 1.56$$

2. Calculate Information Gain for Outlook:

The formula for information gain is:

$$IG(S, A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} \times Entropy(S_v)$$

Where A is the set of all values for feature A , S_v is the subset of samples for which feature A has value v .

Calculate the information gain for the Outlook feature:

- Split the dataset based on Outlook:
 - Sunny: 5 samples (2 'No', 3 'Yes')
 - Overcast: 4 samples (0 'No', 4 'Yes')
 - Rainy: 5 samples (3 'No', 2 'Yes')

$$E(\text{Sunny}) = - \left[\frac{2}{5} \log_2 \left(\frac{2}{5} \right) + \frac{3}{5} \log_2 \left(\frac{3}{5} \right) \right]$$

$$IG(S, \text{Outlook}) = 1.56 - \left(\frac{5}{14} \times 0.97 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.97 \right)$$

$$IG(S, \text{Outlook}) = 1.56 - (0.35 + 0 + 0.35)$$

$$IG(S, \text{Outlook}) \approx 1.56 - 0.70$$

$$IG(S, \text{Outlook}) \approx 0.86$$



Information Gain for Humidity:

1. **Split the dataset based on Humidity:**

- High: 7 samples (3 'No', 4 'Yes')
- Normal: 7 samples (2 'No', 5 'Yes')

2. **Calculate the entropy for each subset:**

- Entropy(High) = $-\left(\frac{3}{7} \log_2 \left(\frac{3}{7}\right) + \frac{4}{7} \log_2 \left(\frac{4}{7}\right)\right)$
- Entropy(Normal) = $-\left(\frac{2}{7} \log_2 \left(\frac{2}{7}\right) + \frac{5}{7} \log_2 \left(\frac{5}{7}\right)\right)$

3. **Calculate Information Gain:**

$$IG(S, \text{Humidity}) = 1.56 - \left(\frac{7}{14} \times 0.99 + \frac{7}{14} \times 0.86\right)$$

$$IG(S, \text{Humidity}) = 1.56 - (0.50 + 0.49)$$

$$IG(S, \text{Humidity}) \approx 1.56 - 0.99$$

$$IG(S, \text{Humidity}) \approx 0.57$$

Information Gain for Wind:

1. **Split the dataset based on Wind:**

- Strong: 6 samples (2 'No', 4 'Yes')
- Weak: 8 samples (3 'No', 5 'Yes')

2. **Calculate the entropy for each subset:**

- Entropy(Strong) = $-\left(\frac{2}{6} \log_2 \left(\frac{2}{6}\right) + \frac{4}{6} \log_2 \left(\frac{4}{6}\right)\right)$
- Entropy(Weak) = $-\left(\frac{3}{8} \log_2 \left(\frac{3}{8}\right) + \frac{5}{8} \log_2 \left(\frac{5}{8}\right)\right)$

3. **Calculate Information Gain:**

$$IG(S, \text{Wind}) = 1.56 - \left(\frac{6}{14} \times 0.92 + \frac{8}{14} \times 0.81\right)$$

$$IG(S, \text{Wind}) = 1.56 - (0.39 + 0.48)$$

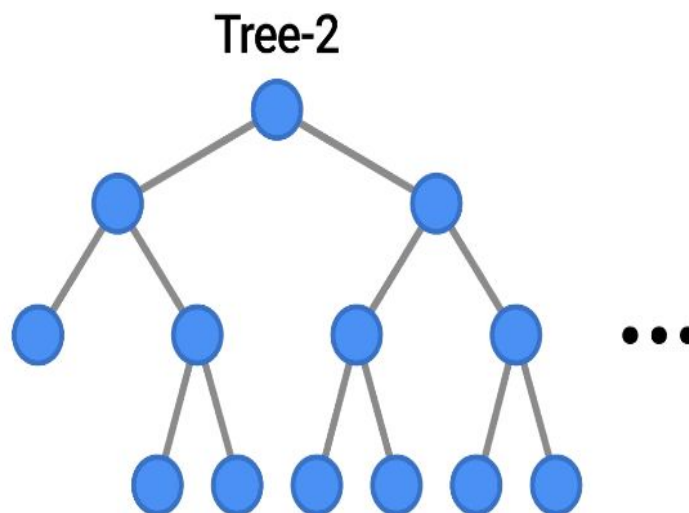
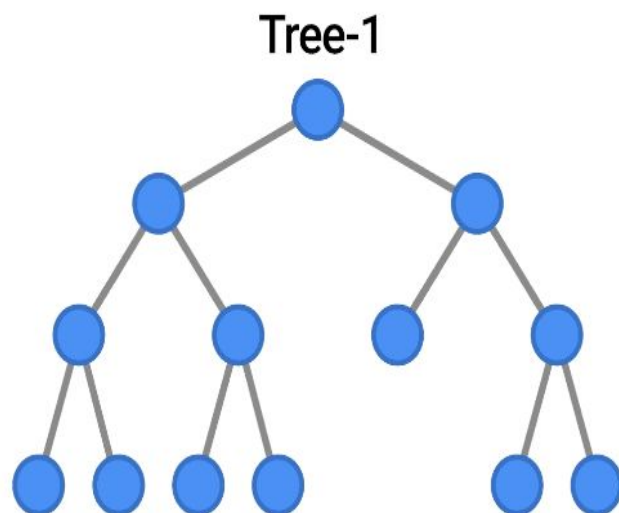
$$IG(S, \text{Wind}) \approx 1.56 - 0.87$$

$$IG(S, \text{Wind}) \approx 0.69$$

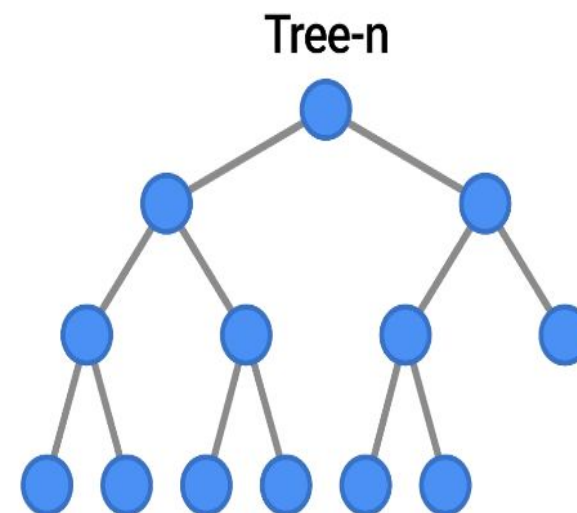


Ranodom Forest

EXAMPLES



...

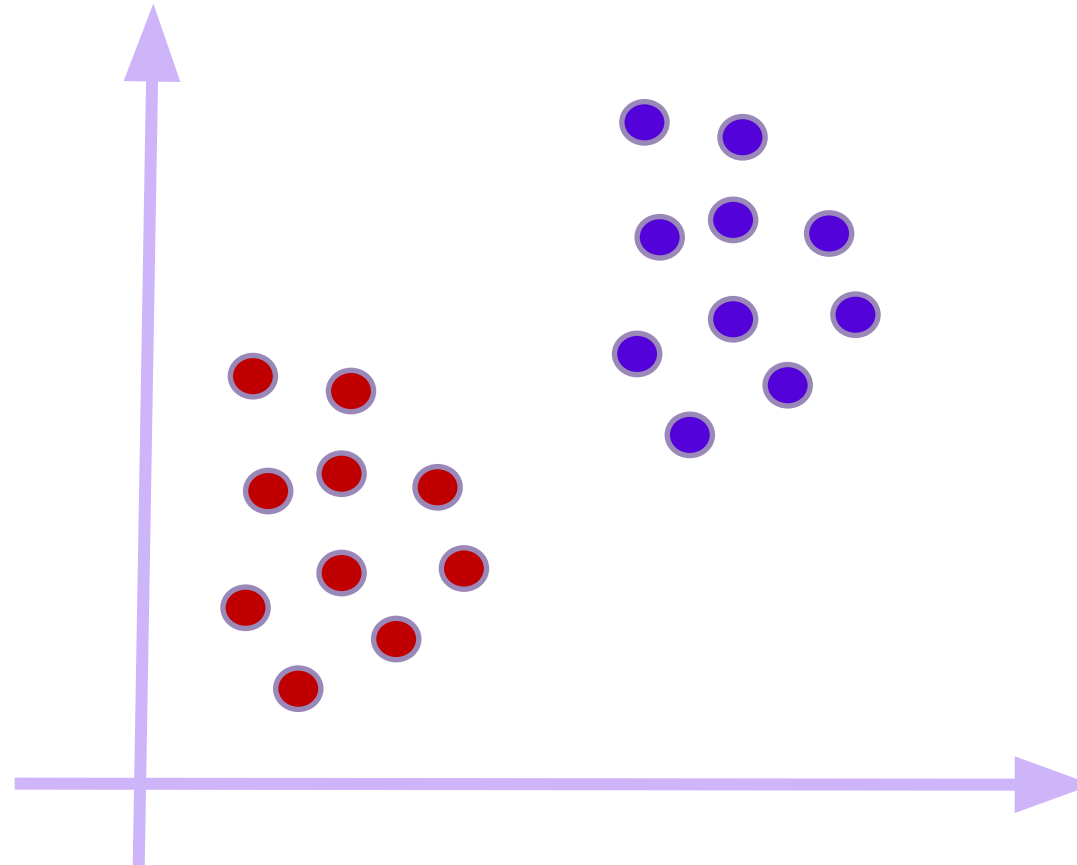


- 1. Decision Trees:** Random Forest is composed of a collection of decision trees. Decision trees are hierarchical structures where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the outcome or class label. Random Forest utilizes decision trees as its base learners.
- 2. Bootstrapping:** Random Forest employs a technique called bootstrapping to create multiple subsets of the original dataset. Bootstrapping involves randomly sampling data points from the dataset with replacement to create new datasets of the same size as the original. Each decision tree in the Random Forest is trained on one of these bootstrapped datasets, introducing diversity among the trees.
- 3. Random Feature Selection:** In addition to bootstrapping, Random Forest also randomly selects a subset of features to consider when splitting nodes in the decision trees. This random feature selection helps to decorrelate the trees and reduces the risk of overfitting. Typically, the square root of the total number of features is used as the number of features to consider at each split.

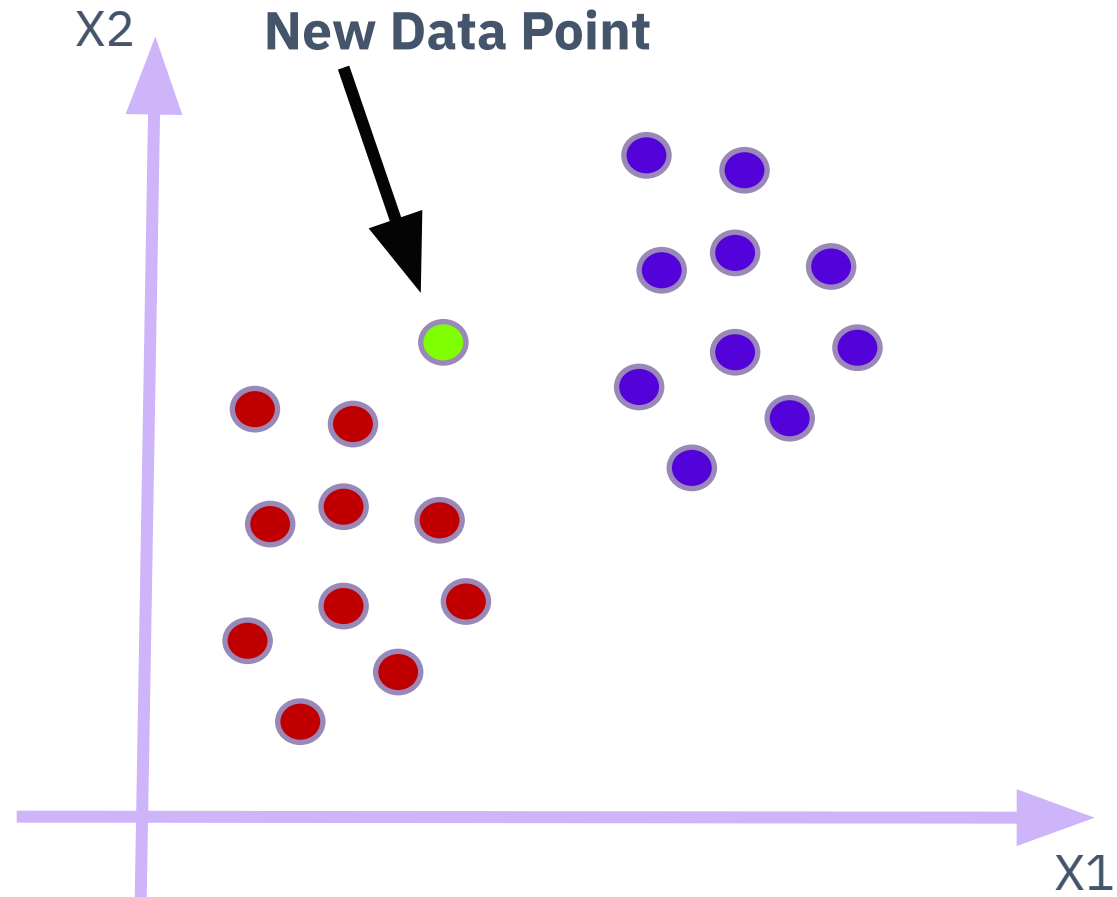
- 1.Voting or Averaging:** During prediction, Random Forest combines the predictions of all individual trees to make the final prediction. For classification tasks, it uses a majority voting mechanism where each tree "votes" for the most common class label among its predictions. For regression tasks, it averages the predictions of all trees to obtain the final output.
- 2.Ensemble Learning:** Random Forest belongs to the ensemble learning family of algorithms. Ensemble learning combines multiple models to improve predictive performance compared to any individual model. Random Forest leverages the wisdom of crowds by aggregating the predictions of multiple decision trees, resulting in better generalization and robustness compared to a single decision tree.

K-Nearest Neighbour (KNN)

K -Nearest Neighbors (KNN)

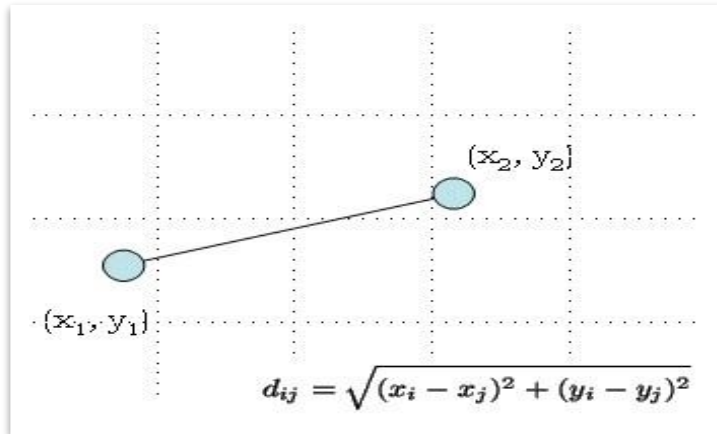


KNN-K Nearest Neighbors



KNN-K Nearest Neighbors

Euclidean Distance



Minkowski Distance

Manhattan Distance

Manhattan

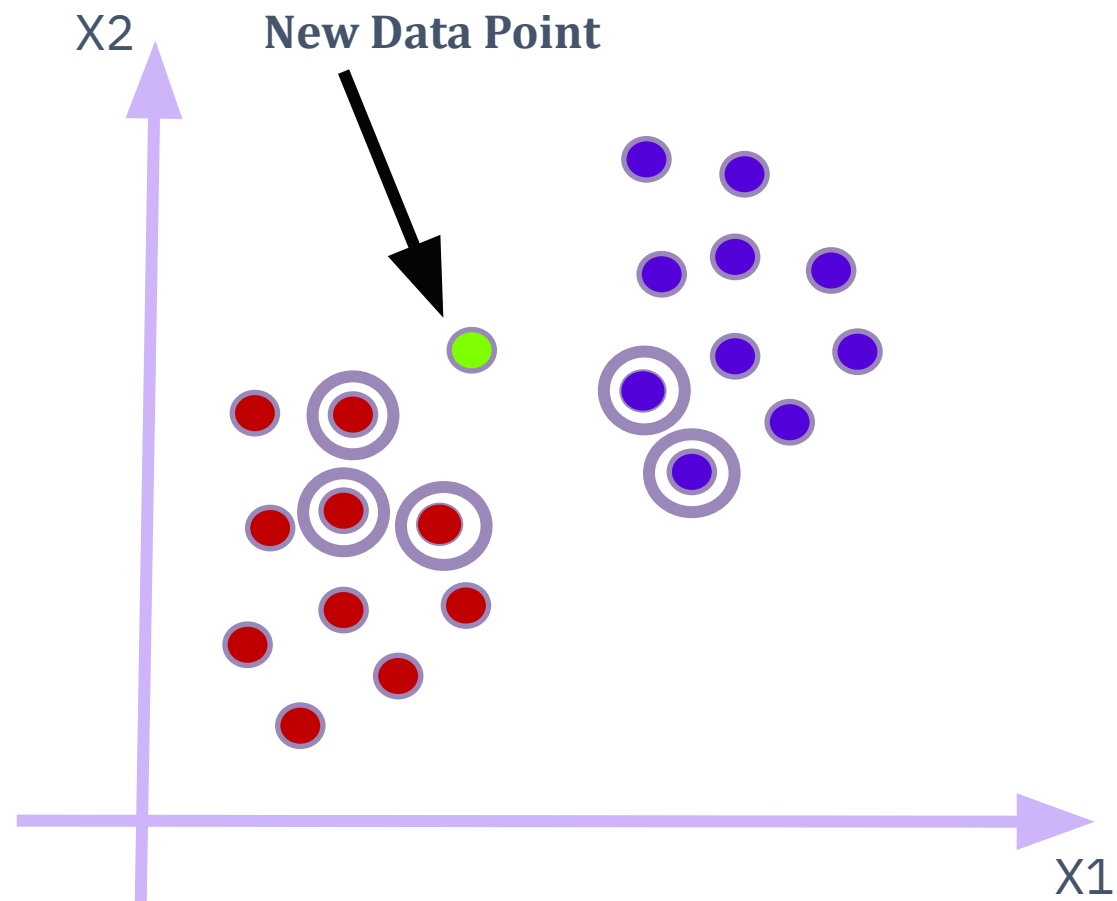
$$\sum_{i=1}^k |x_i - y_i|$$

Minkowski

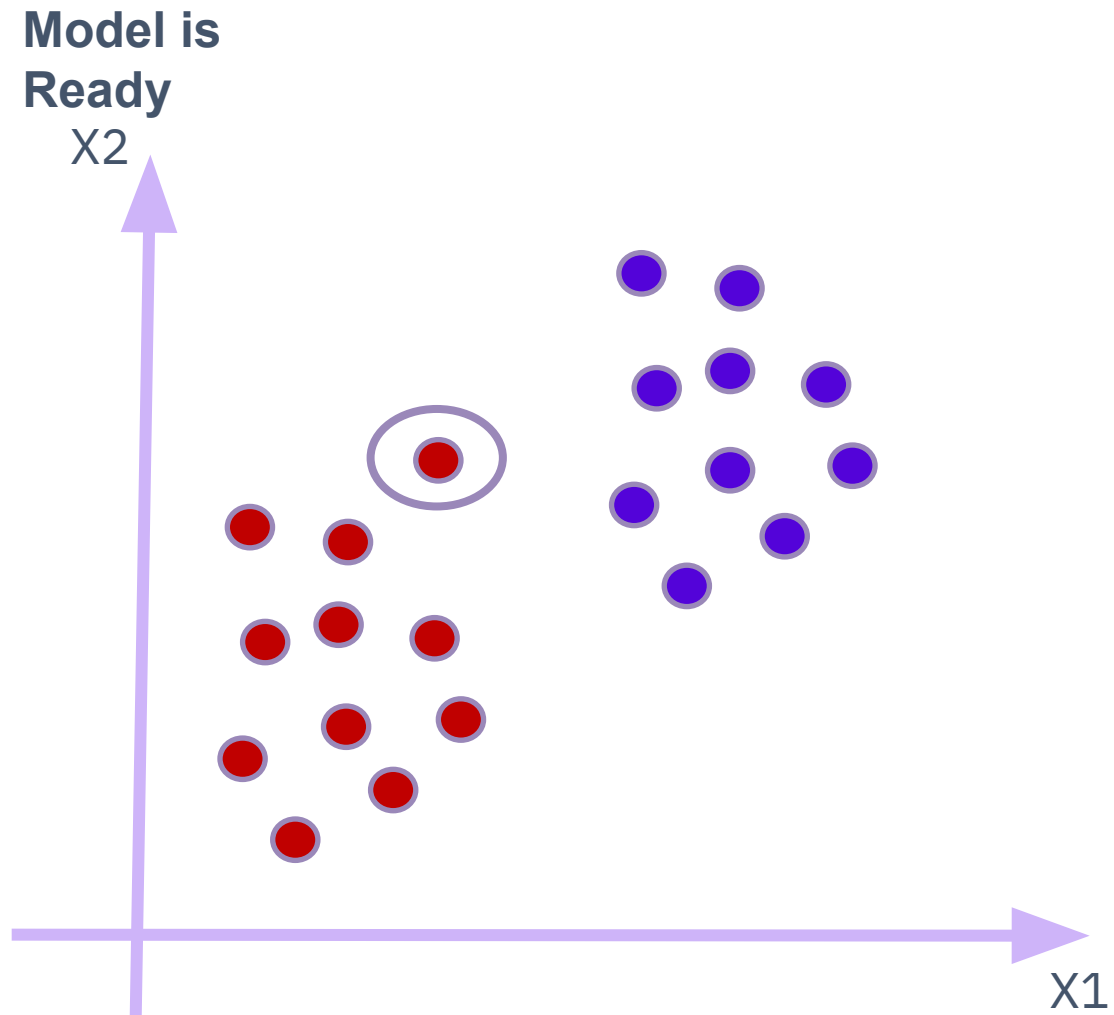
$$\left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{1/q}$$

KNN-K Nearest Neighbors

Let consider $K = 5$



KNN-K Nearest Neighbors



KNN-K Nearest Neighbors

Step -1 : Choose the number K of neighbors



Step -2 : Take the K nearest neighbors of the new datapoint, according to the Euclidean distance



Step -3 : Among the K Neighbors, count the number of data points in each category



Step -4 : Assign the new data point to the category where you counted the most neighbors

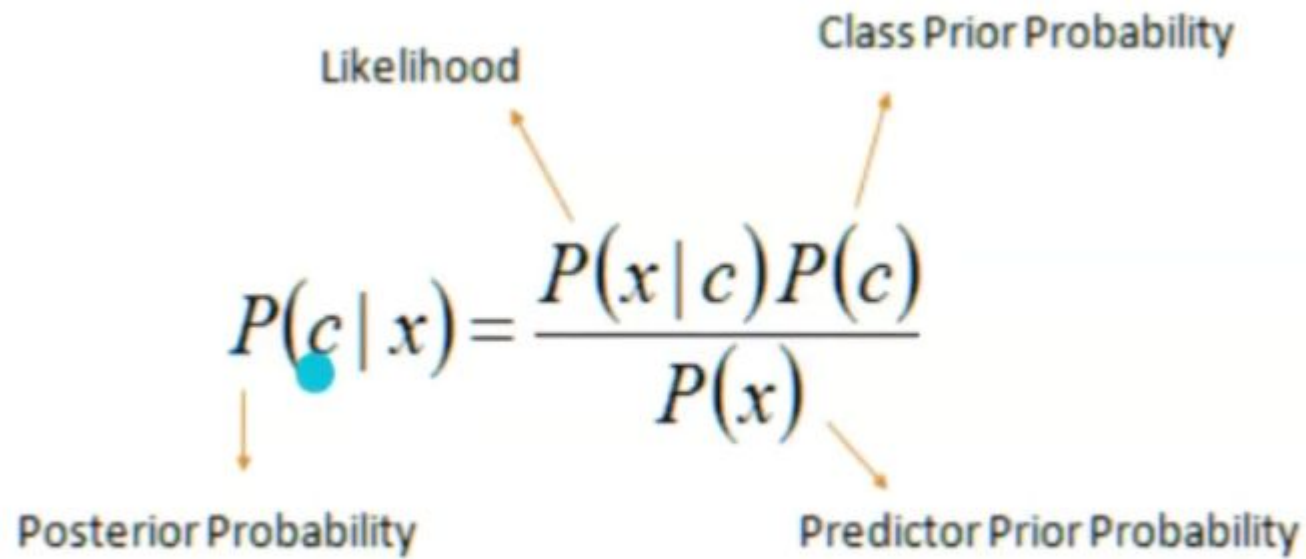


Model is Ready

Naïve Bayes

Bayes theorem

- The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors.
- The essence of the Bayes theorem is conditional probability where conditional probability is the probability that something will happen, given that something else has already occurred.
- By using conditional probability, we can find out the probability of an event will occur given the knowledge of the previous event.



The diagram shows the equation for Bayes' Theorem: $P(c | x) = \frac{P(x | c) P(c)}{P(x)}$. The variable c in the numerator is highlighted with a blue dot. Four orange arrows point from text labels to parts of the equation: 'Likelihood' points to $P(x | c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c | x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c | x) = \frac{P(x | c) P(c)}{P(x)}$$

Labels and their corresponding terms in the equation:

- Likelihood: $P(x | c)$
- Class Prior Probability: $P(c)$
- Posterior Probability: $P(c | x)$
- Predictor Prior Probability: $P(x)$

- $P(c|x)$ is the posterior probability of class (target) given predictor (attribute).
- $P(x|c)$ is the likelihood which is the probability of predictor given class.
- $P(c)$ is the prior probability of class.

- $P(x)$ is the prior probability of predictor.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Diagram illustrating the components of the equation:

- $P(c|x)$ is labeled as Posterior Probability.
- $P(x|c)$ is labeled as Likelihood.
- $P(c)$ is labeled as Class Prior Probability.
- $P(x)$ is labeled as Predictor Prior Probability.

- The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target.
- Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class.
- The class with the highest posterior probability is the outcome of prediction.

- The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target.
- Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class.
- The class with the highest posterior probability is the outcome of prediction.

Day	Outlook	Humid	Wind	Play
1	Sunny	High	weak	No
2	Sunny	High	Strong	No
3	Overcast	High	Weak	Yes
4	Rain	High	Weak	Yes
5	Rain	Normal	Weak	Yes
6	Rain	Normal	Strong	No
7	Overcast	Normal	Strong	Yes
8	Sunny	High	Weak	No
9	Sunny	Normal	Weak	Yes
10	Rain	Normal	Weak	Yes
11	Sunny	Normal	Strong	Yes
12	Overcast	High	Strong	Yes
13	Overcast	Normal	Weak	Yes
14	Rain	High	Strong	No

- The posterior probability can be calculated by first, constructing a **frequency table** for each attribute against the target.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Then, transforming the frequency tables to **likelihood tables**

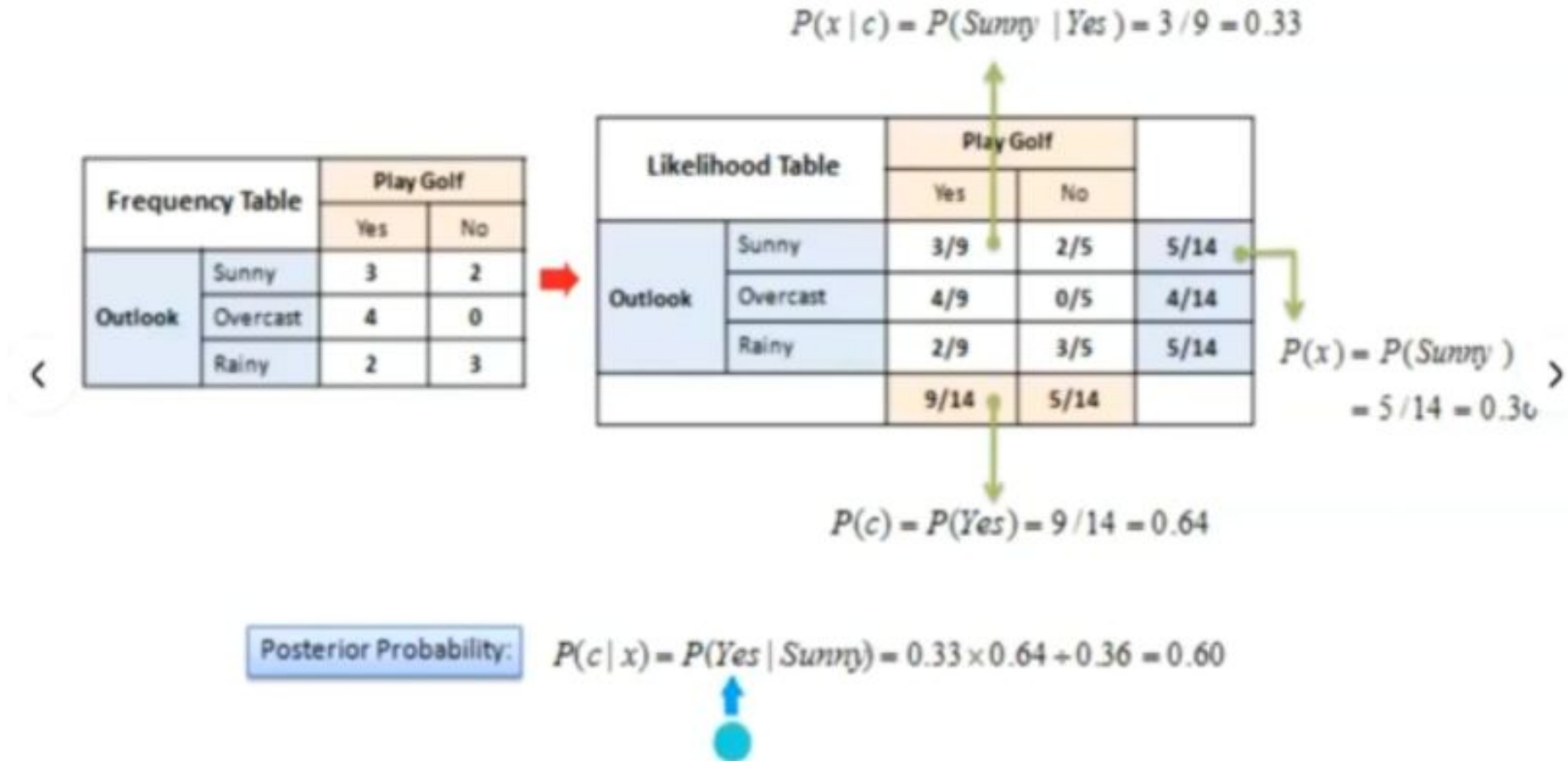
		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

$$P(x|c)$$

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

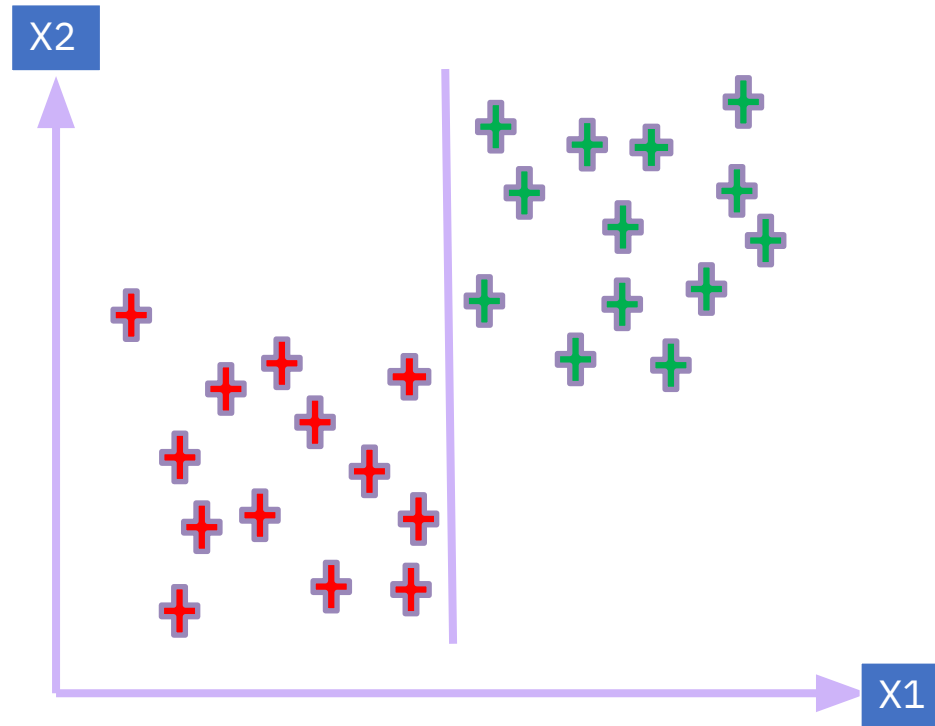
		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

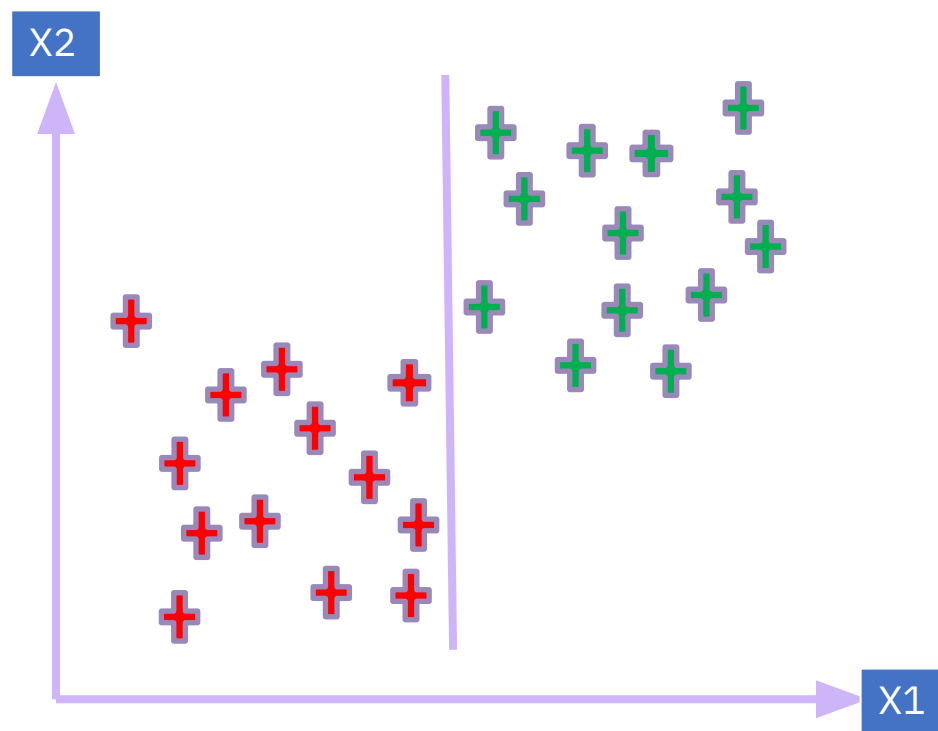


Support Vector Machine(SVM)

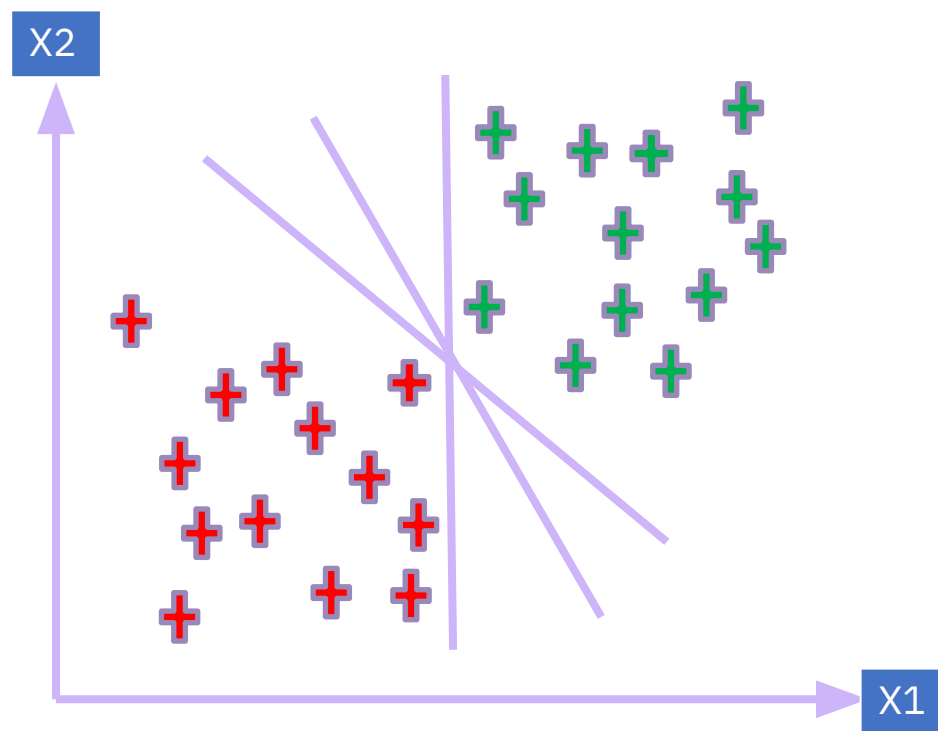
SVM



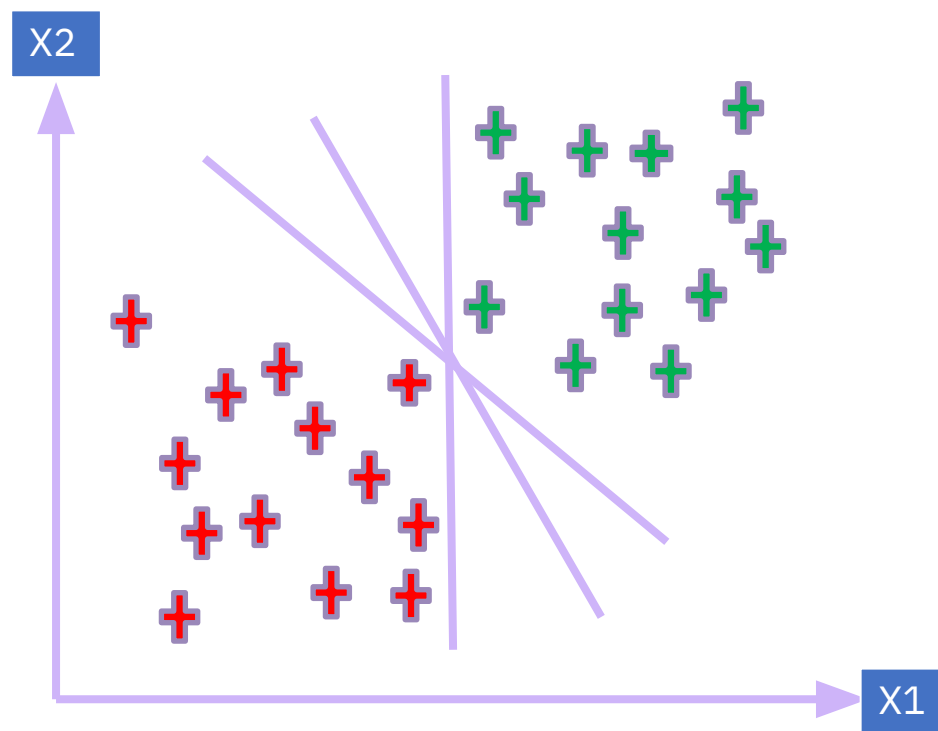
SVM



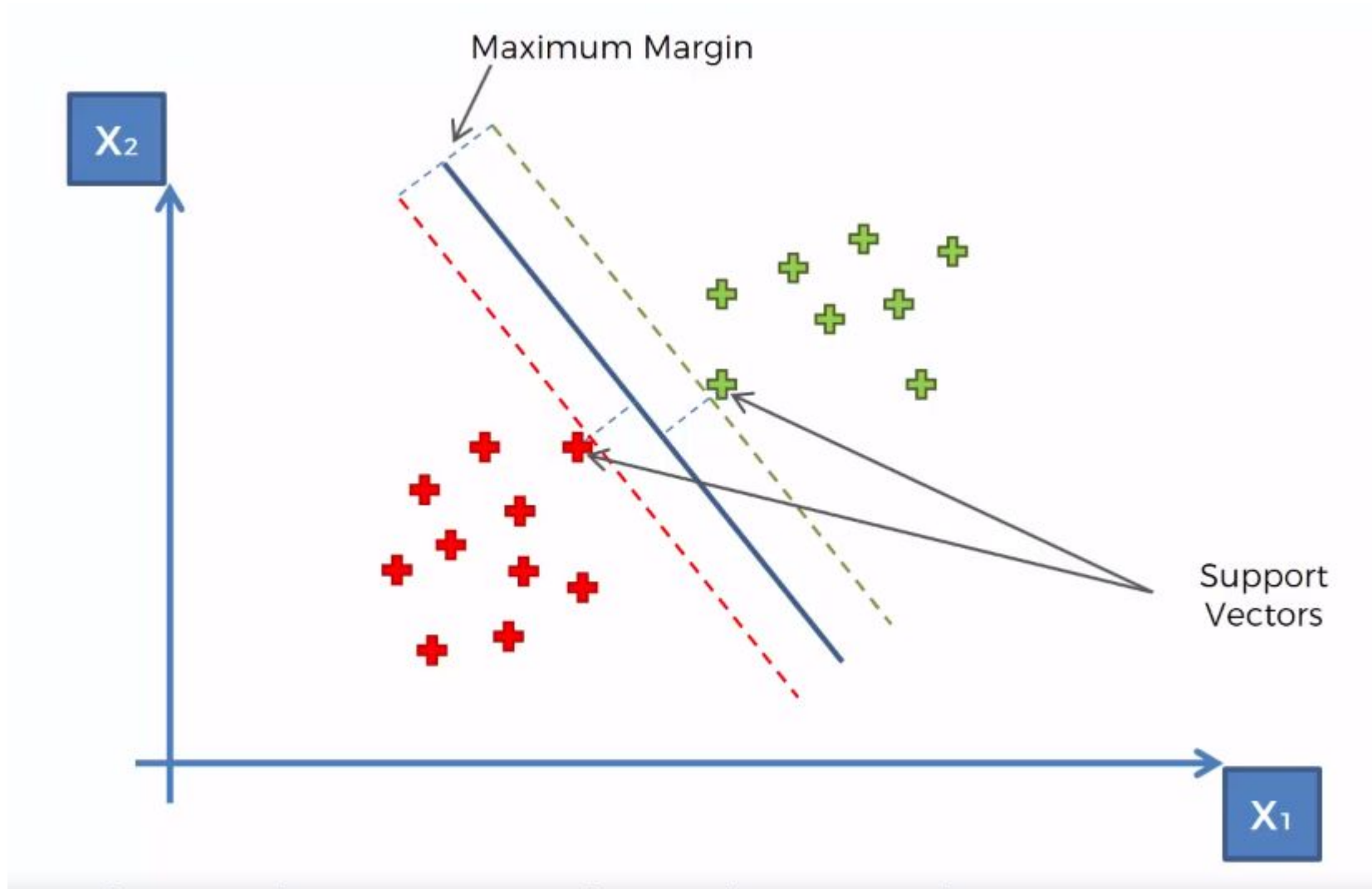
SVM



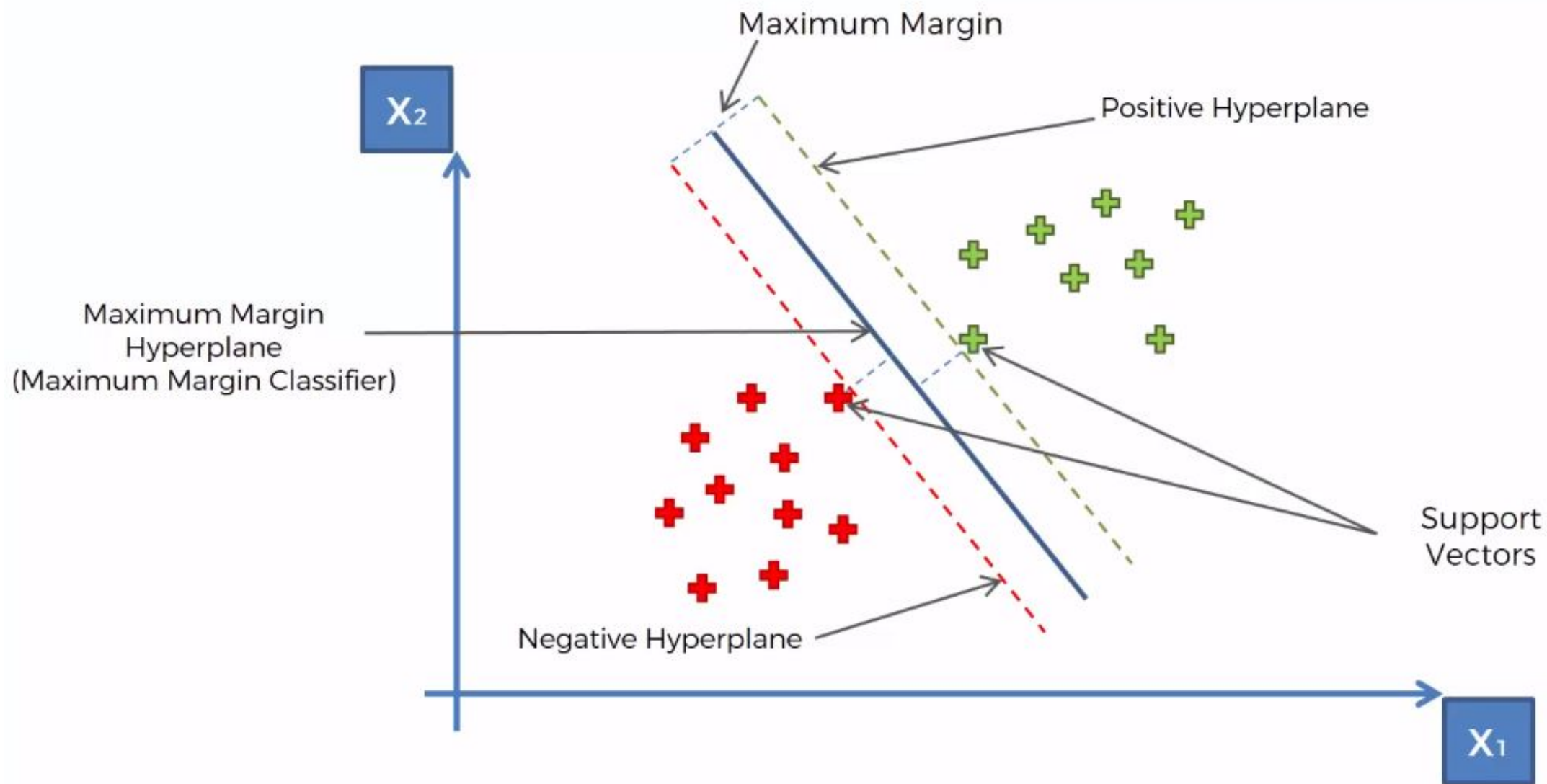
SVM



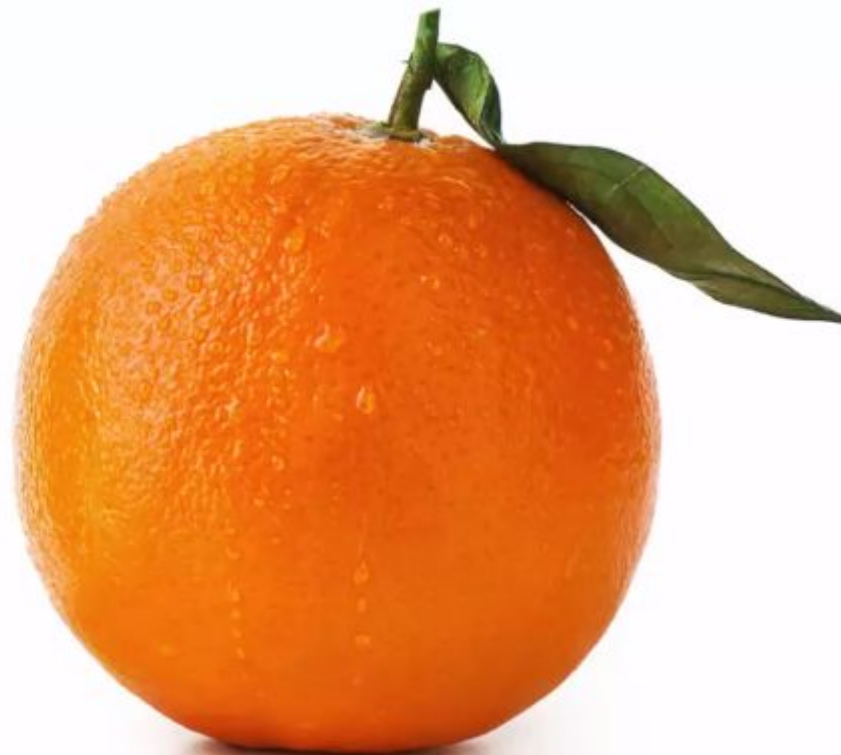
SVM



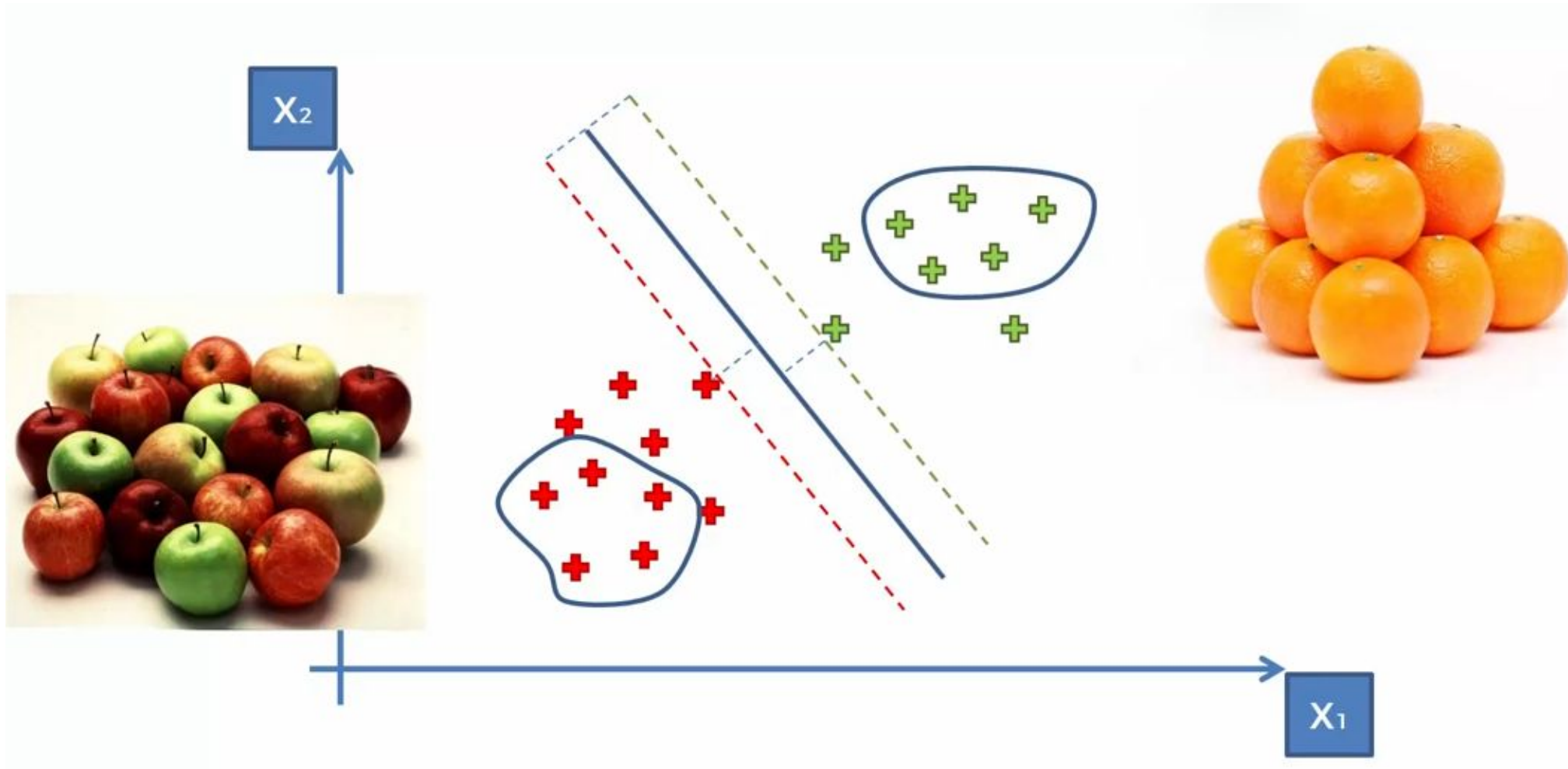
SVM



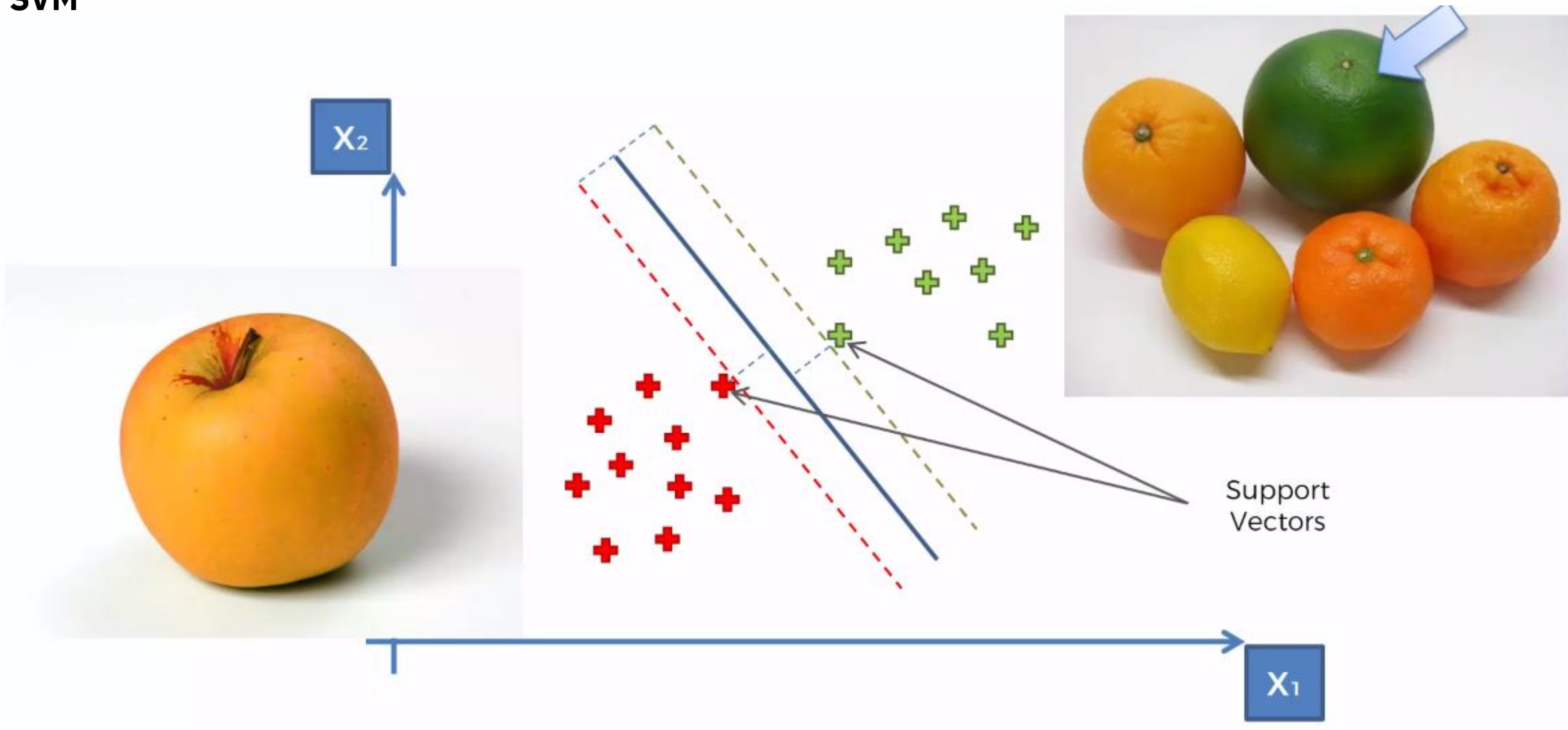
SVM



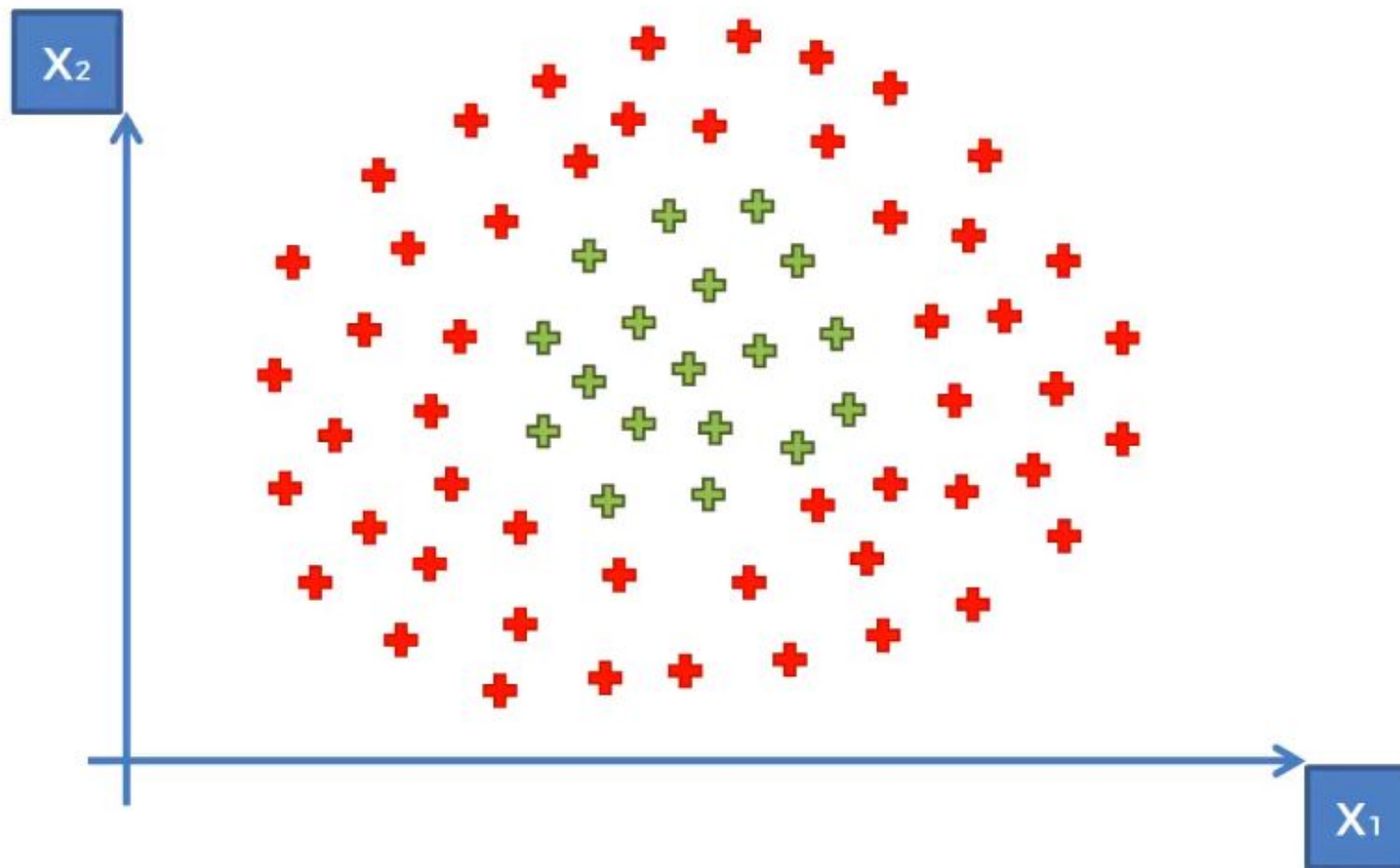
SVM



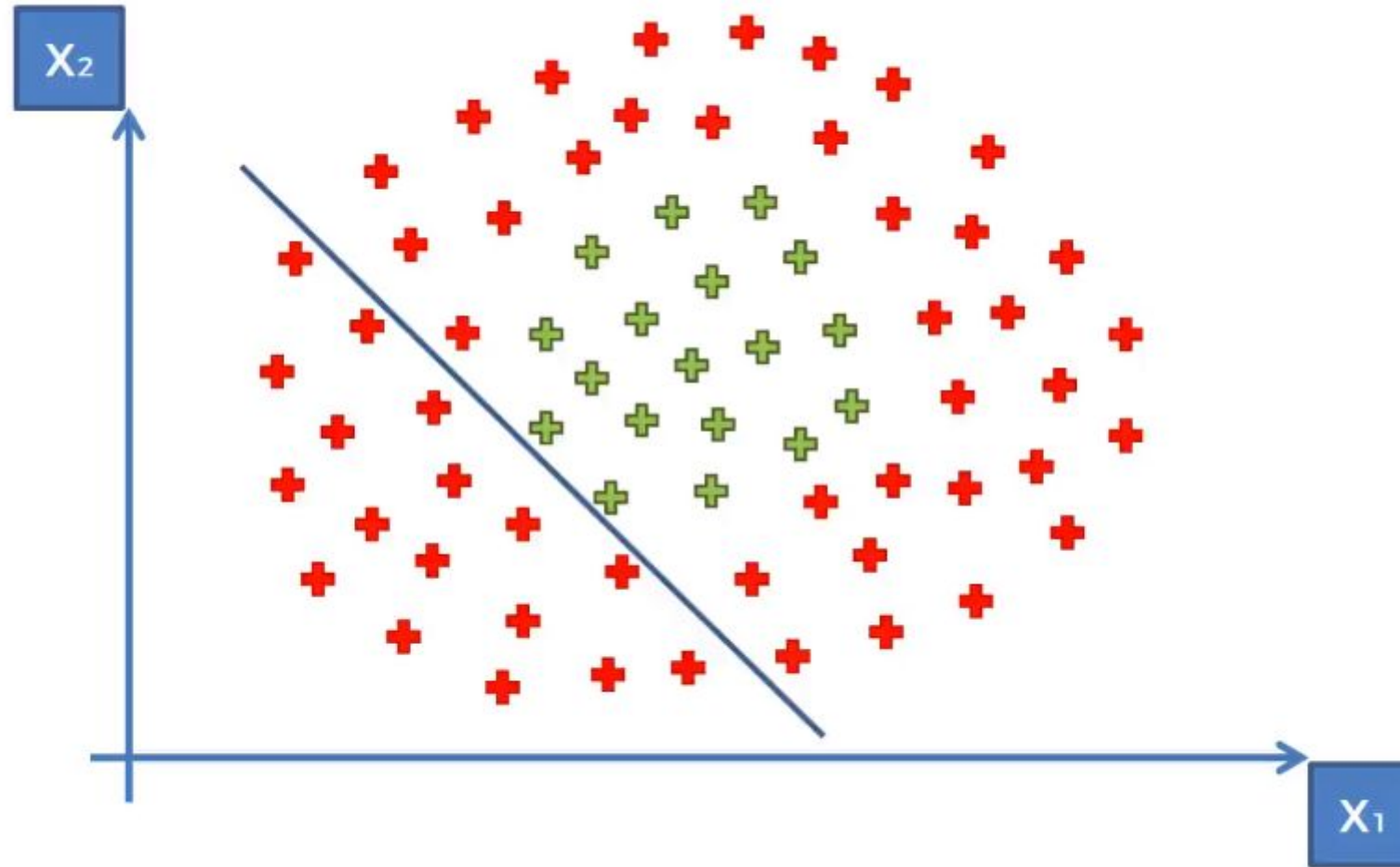
SVM



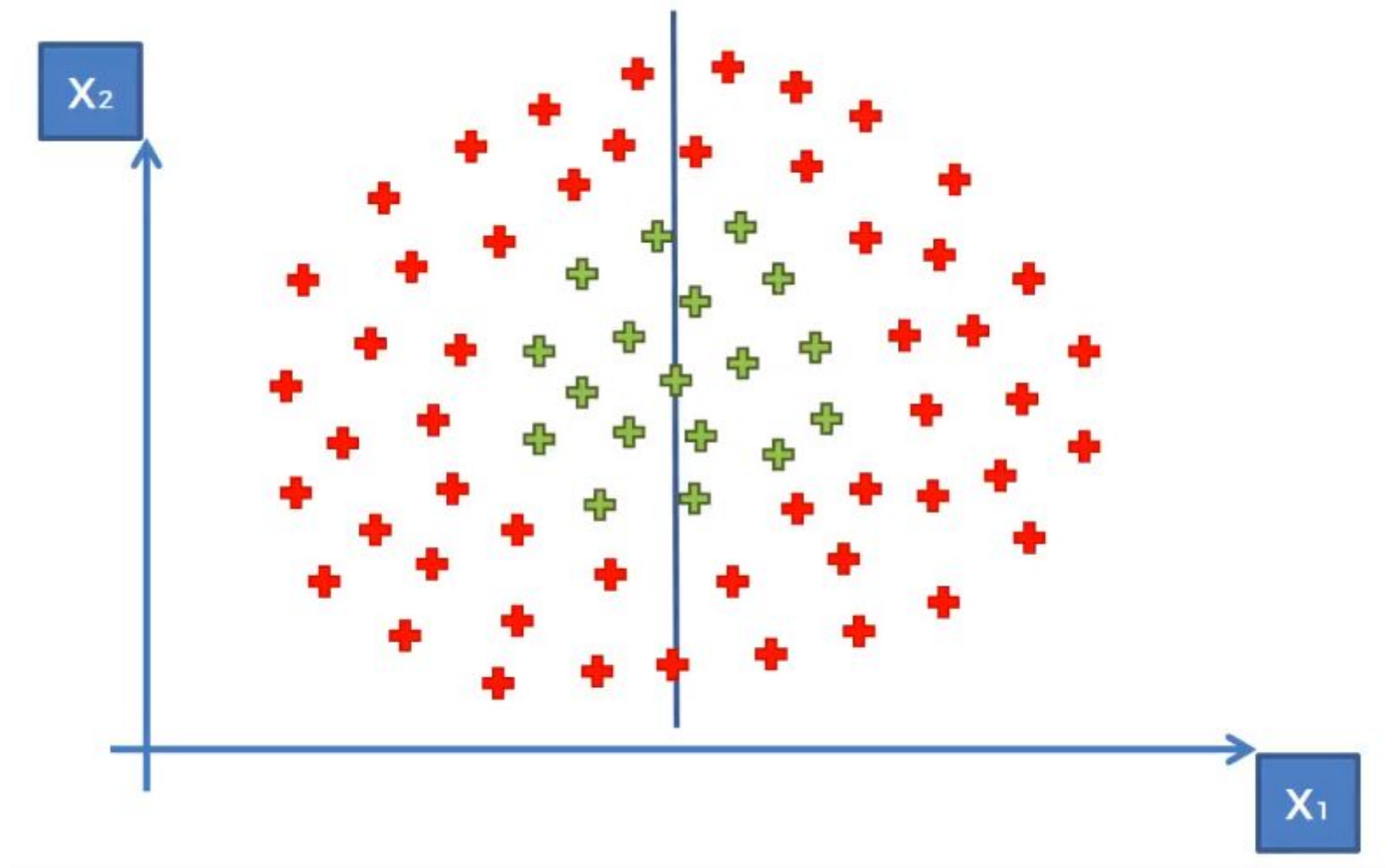
SVM



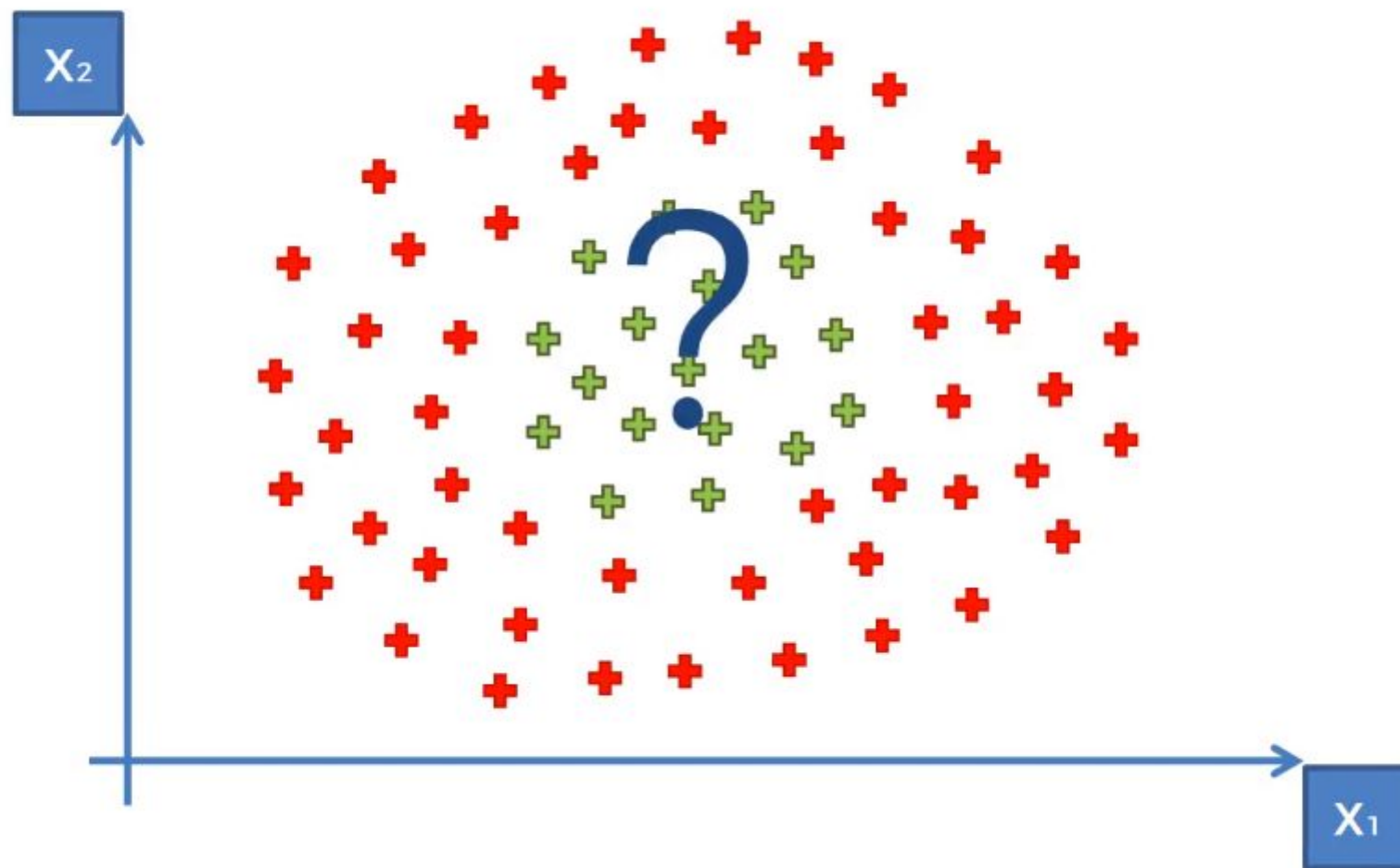
SVM



SVM



SVM

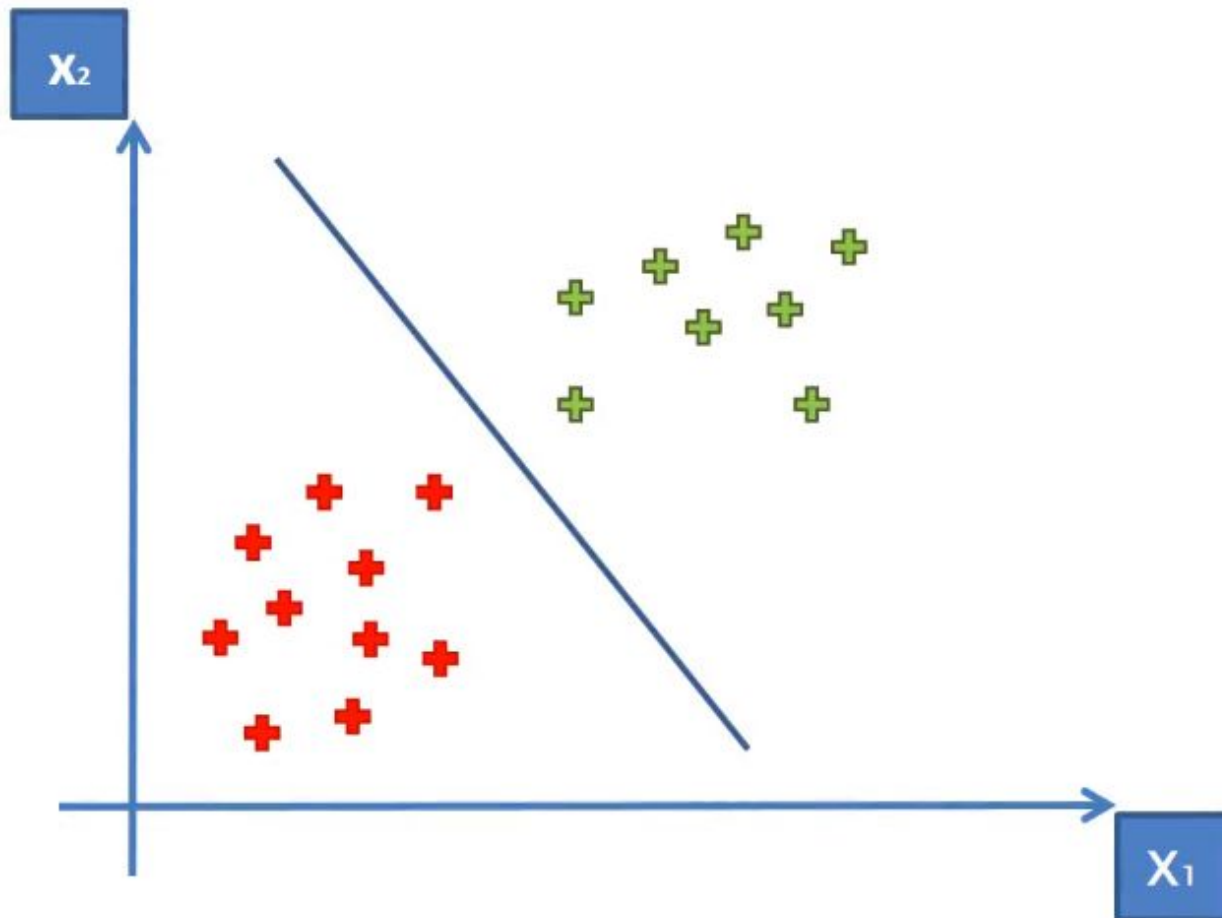


SVM

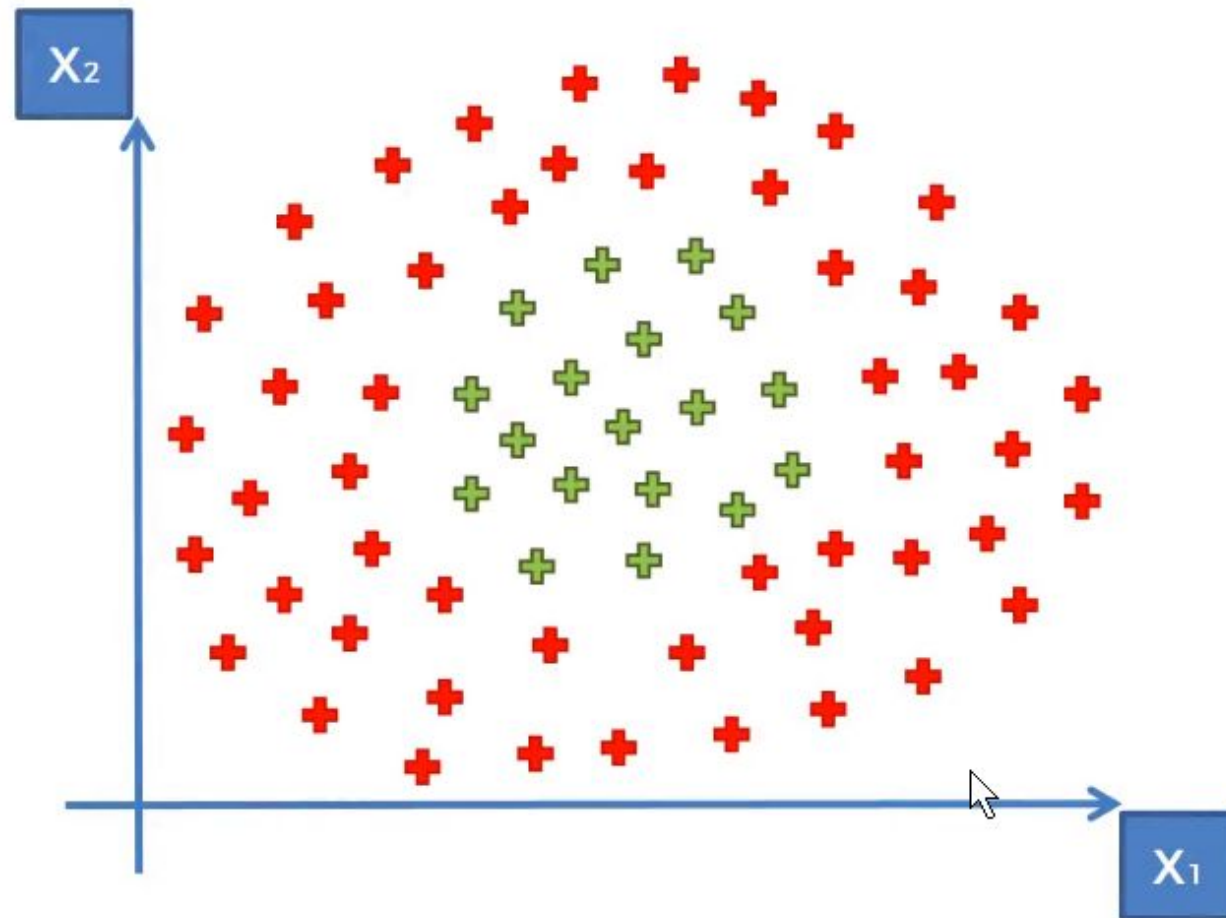
Because the data points are
not LINEARLY SEPARABLE

SVM

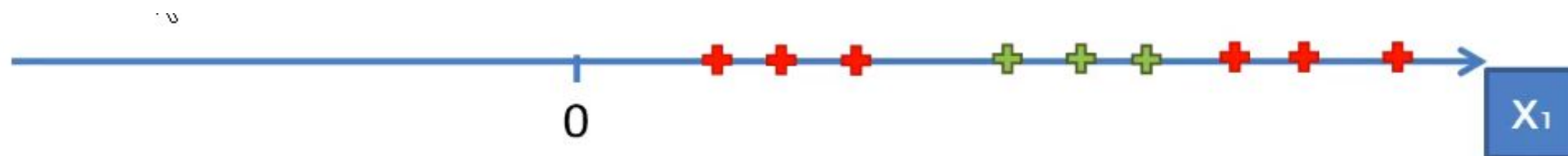
Linearly Separable



Not Linearly Separable

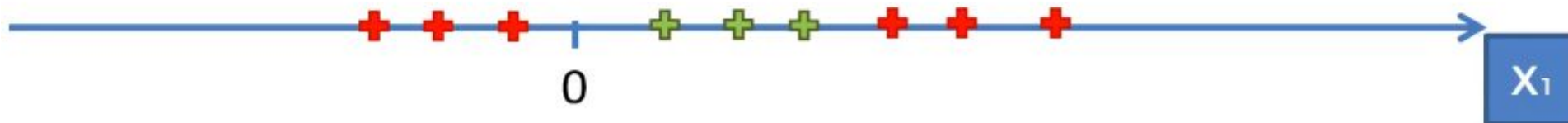


SVM

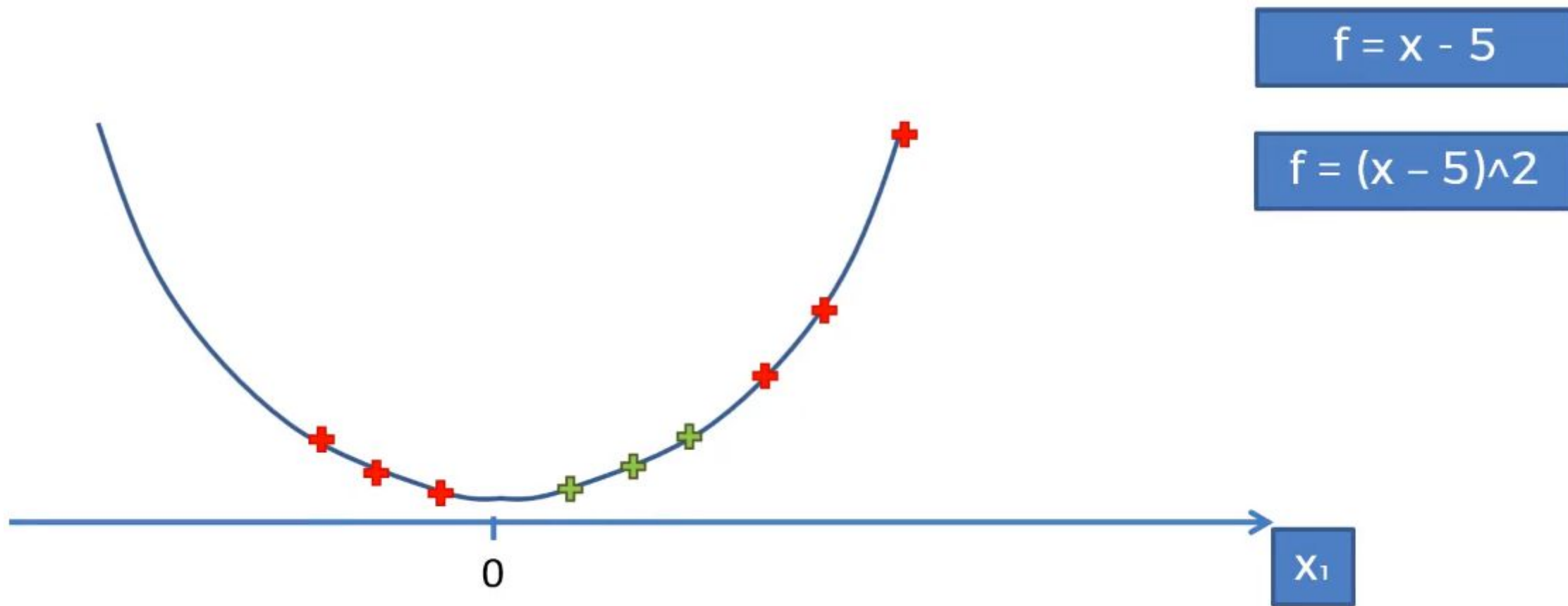


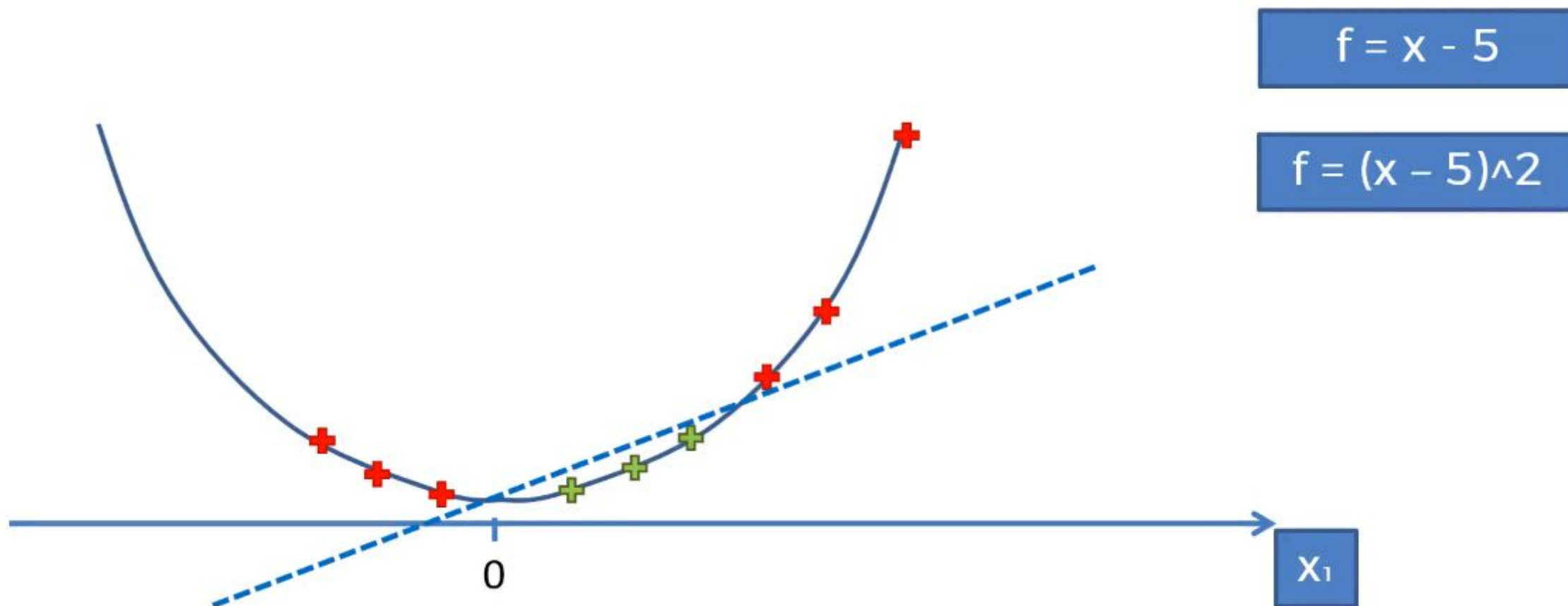
SVM

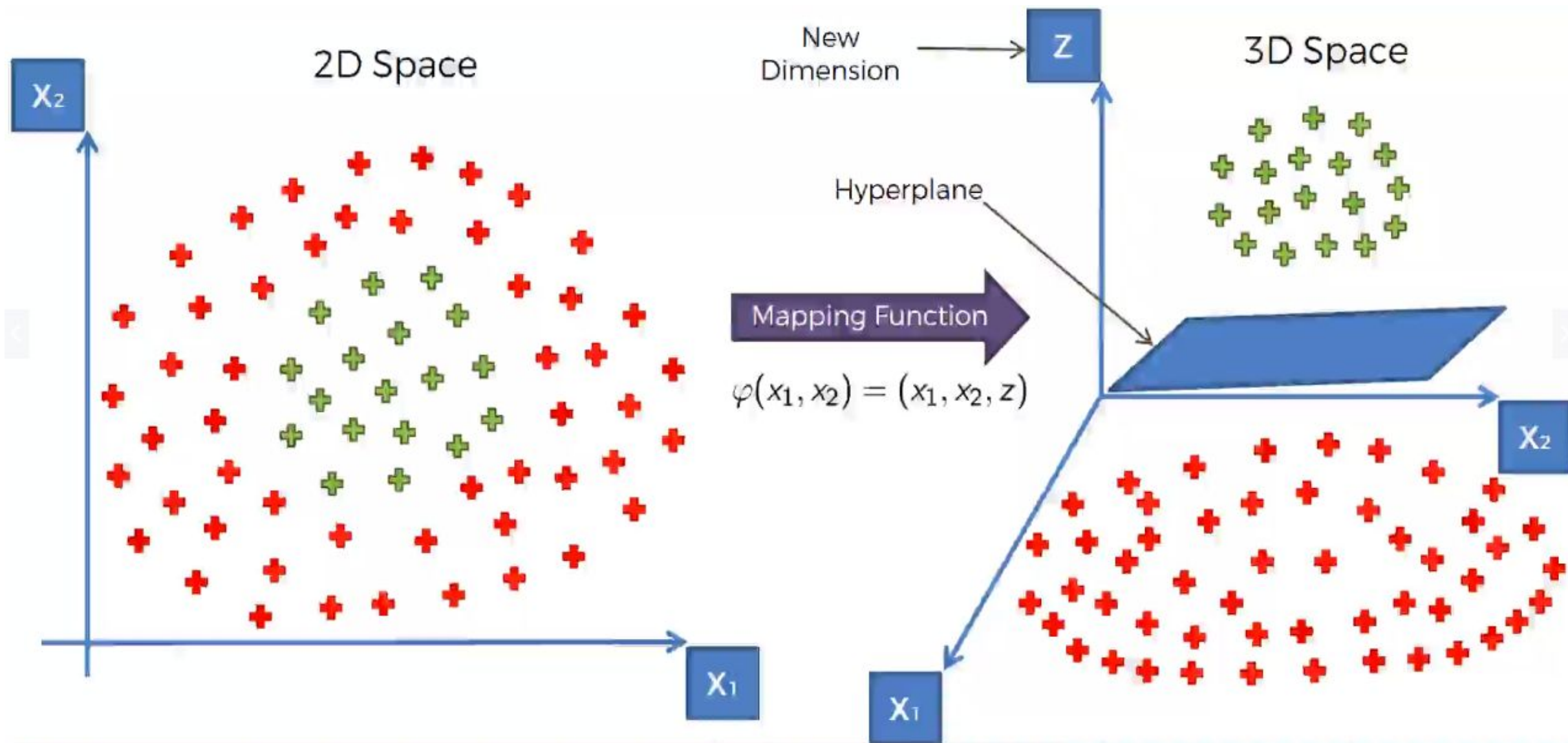
$$f = x - 5$$



SVM

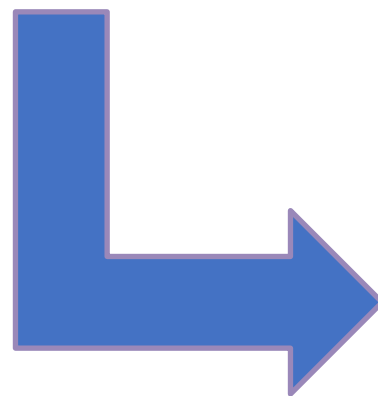




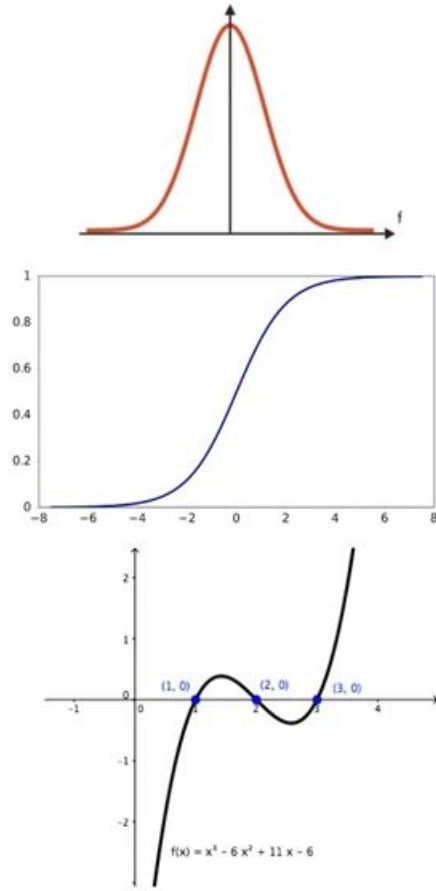


SVM

Mapping to a Higher Dimensional Space
can be highly compute-intensive



Solution : Kernel Trick



Gaussian RBF Kernel

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x} - \vec{l}^i\|^2}{2\sigma^2}}$$

Sigmoid Kernel

$$K(X, Y) = \tanh(\gamma \cdot X^T Y + r)$$

Polynomial Kernel

$$K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

Thank You