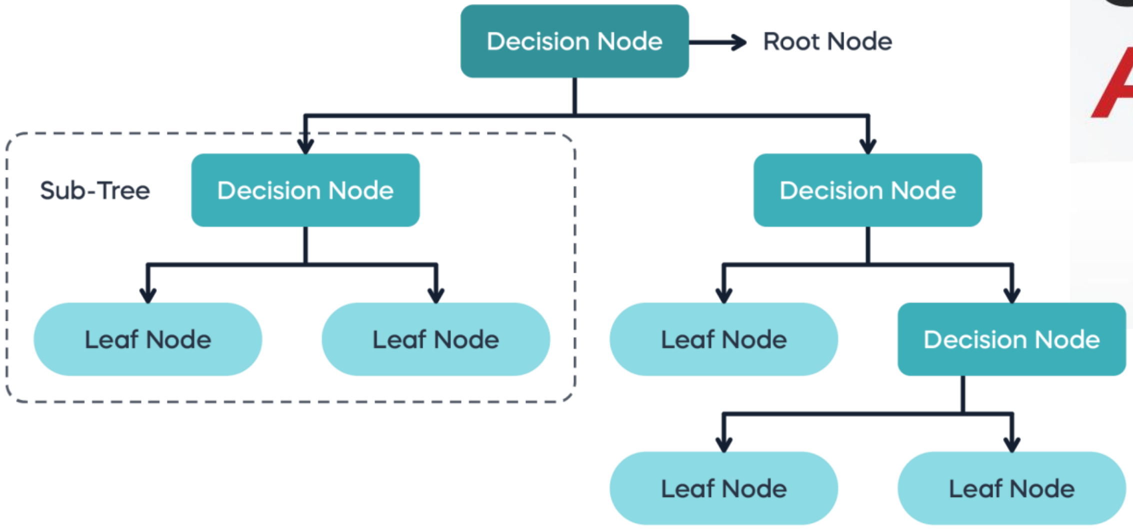
CM2604 Machine Learning

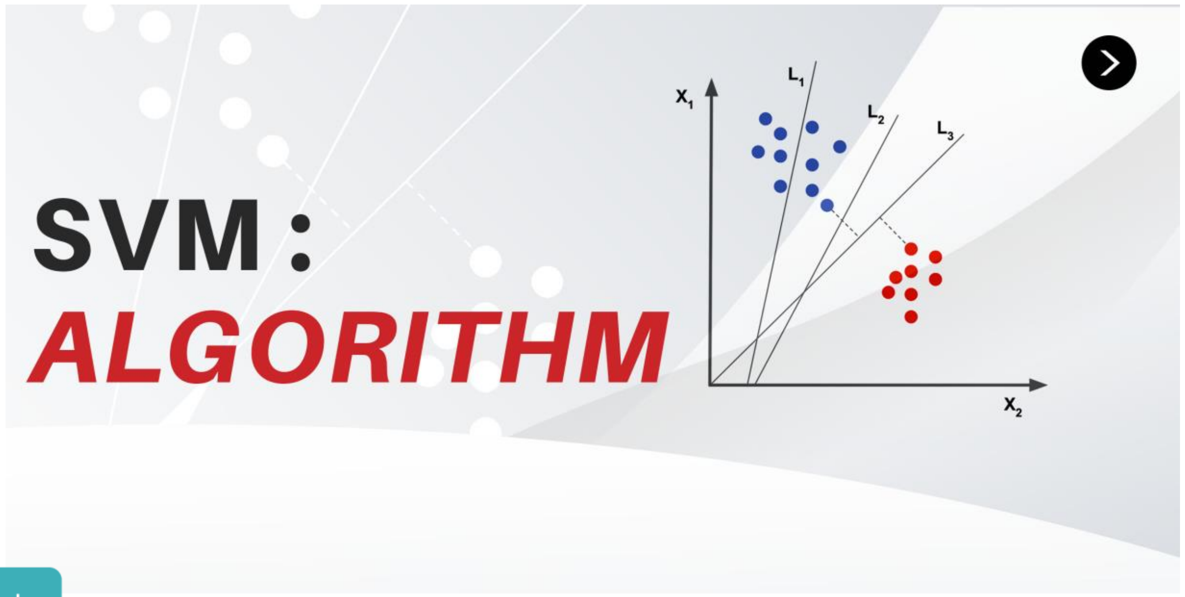
Supervised Machine Learning - Part 2

Week 04 | Prasan Yapa

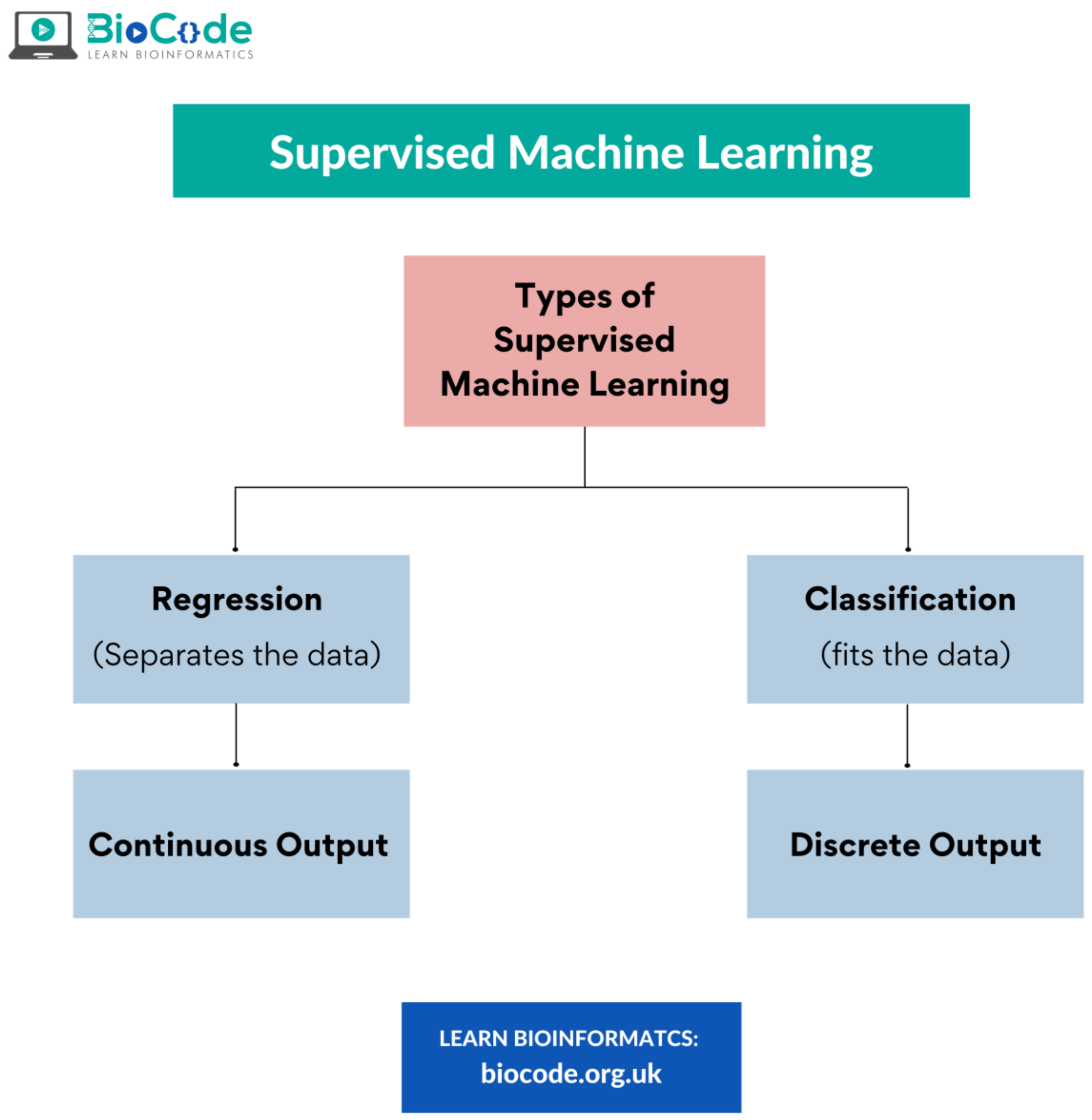
Overview

- General Framework of Supervised Learning
- Support Vector Machines
- Decision Trees





General Framework of Supervised Learning



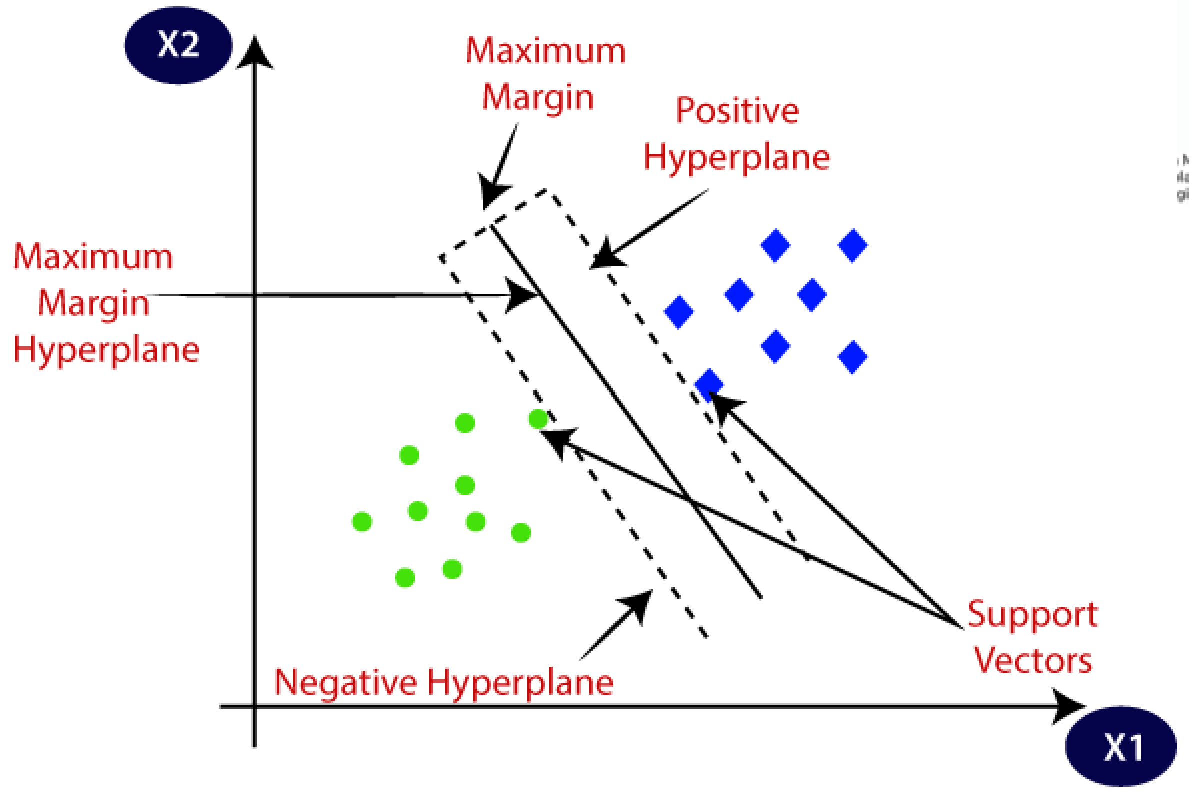
Support Vector Machines (SVM)

Support Vector Machine

- One of the most popular Supervised Learning algorithms for Classification as well as Regression problems.
- The goal is to create the best line or decision boundary that can segregate n-dimensional space into classes.
- This best decision boundary is called a **hyperplane**.
- SVM chooses the extreme points/vectors that help in creating the hyperplane.
- These extreme cases are called as support vectors.

Support Vector Machine

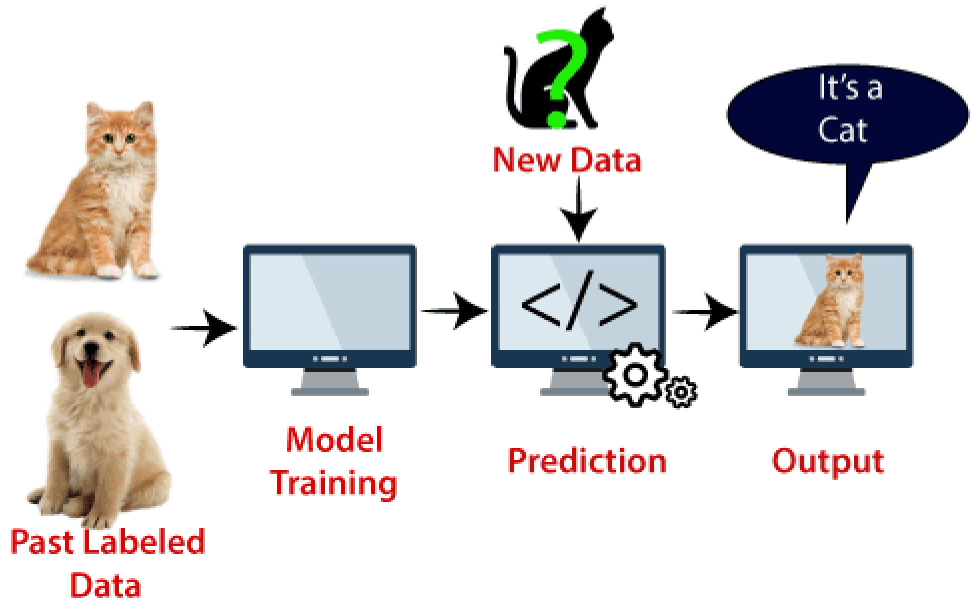
- Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Support Vector Machine

- Support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. Based on the support vectors, it will classify it as a cat.

- SVM algorithm can be used for Face detection, image classification, text categorization, etc.

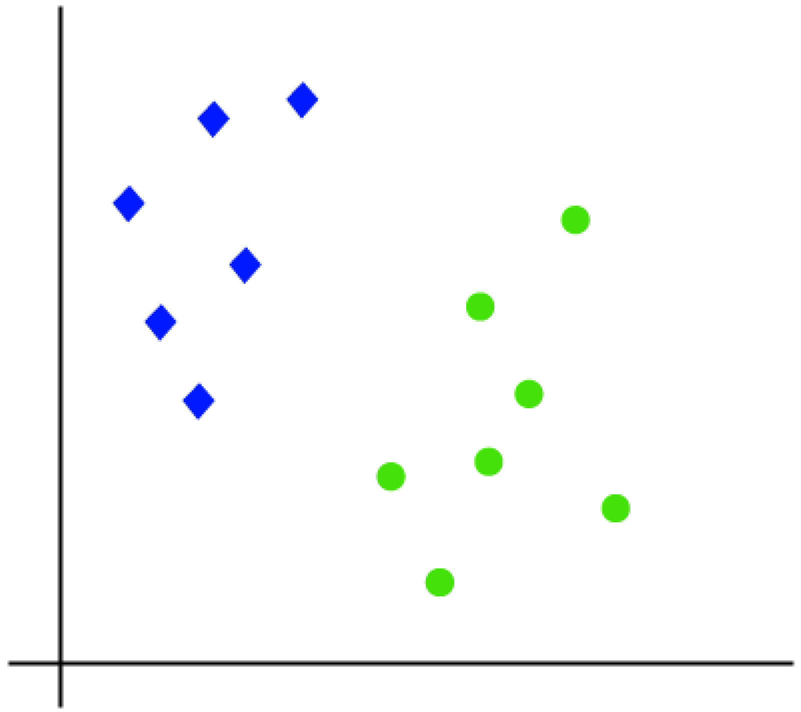


Types of SVM

- Linear SVM
  - Linear SVM is used for linearly separable data.
  - If a dataset can be classified into two classes by using a single straight line.
- Non-linear SVM
  - Non-Linear SVM is used for non-linearly separated data.
  - If a dataset cannot be classified by using a straight line, then such data is termed as non-linear data.

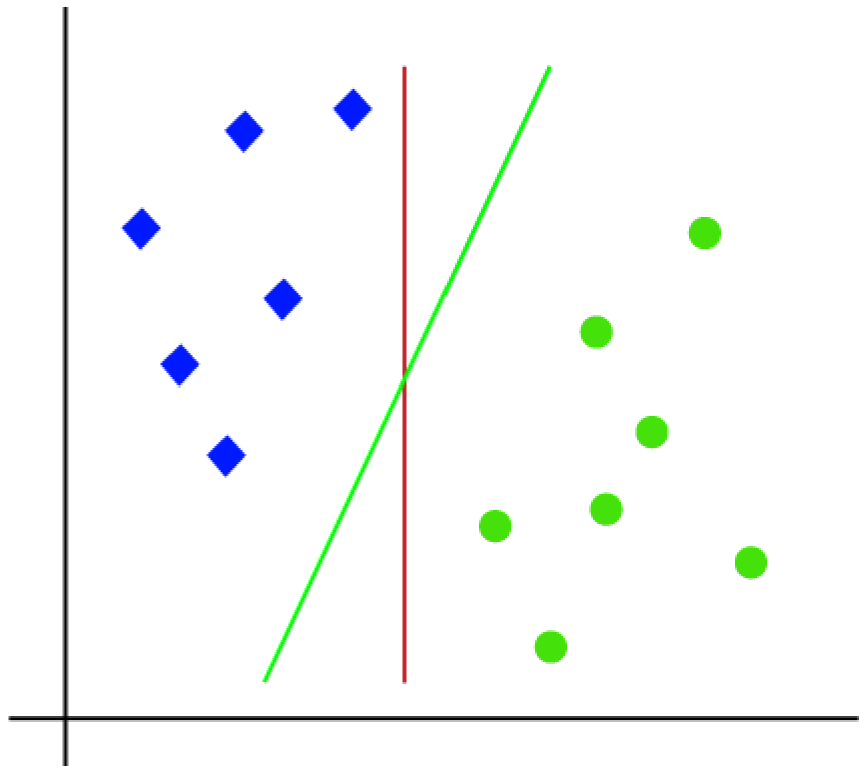
Linear SVM

- The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue.



Linear SVM

- So, as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes.

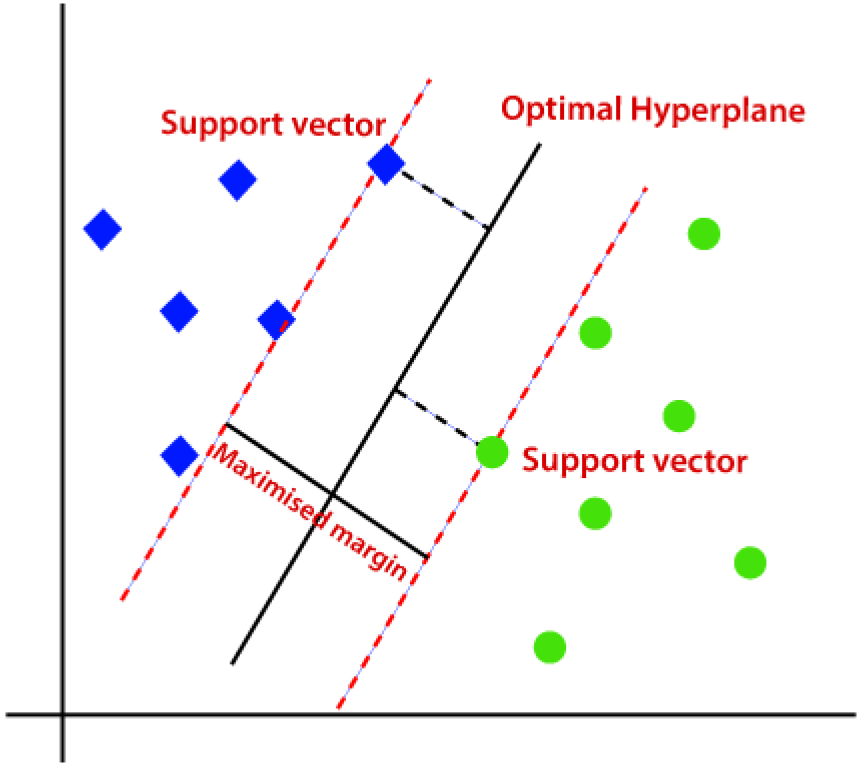


Linear SVM

- SVM algorithm helps to find the best line or decision boundary which is the hyperplane.
- SVM algorithm finds the closest point of the lines from both the classes.
- These points are called **support vectors**.
- The distance between the vectors and the hyperplane is called as **margin**.

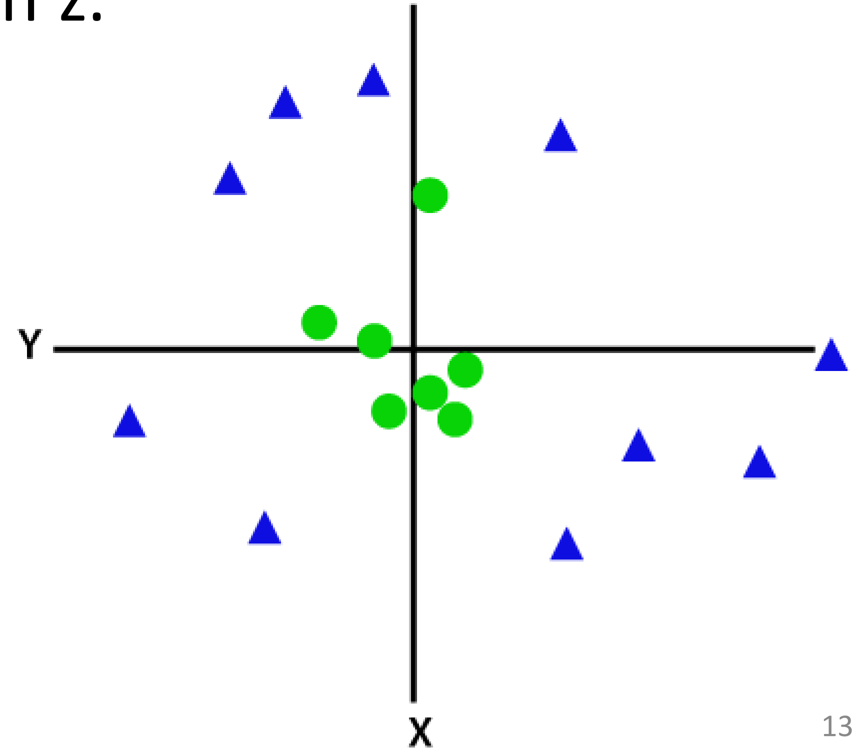
Linear SVM

- The goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the **optimal hyperplane**.



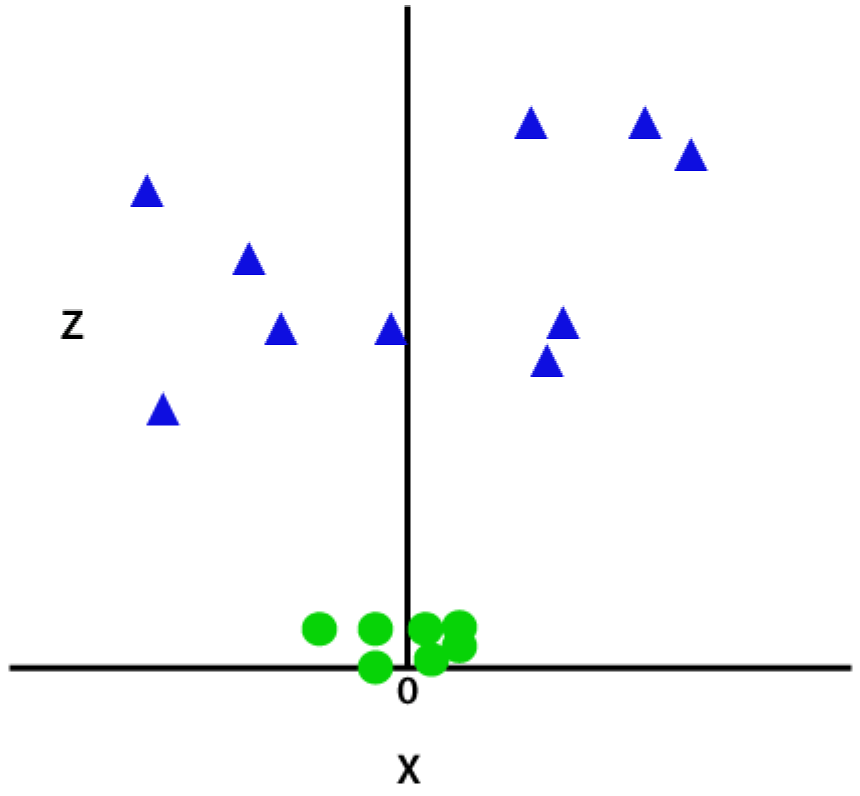
Non-Linear SVM

- For non-linear data, we cannot draw a single straight line.
- For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third-dimension z.
- z can be calculated as $z = x^2 + y^2$.



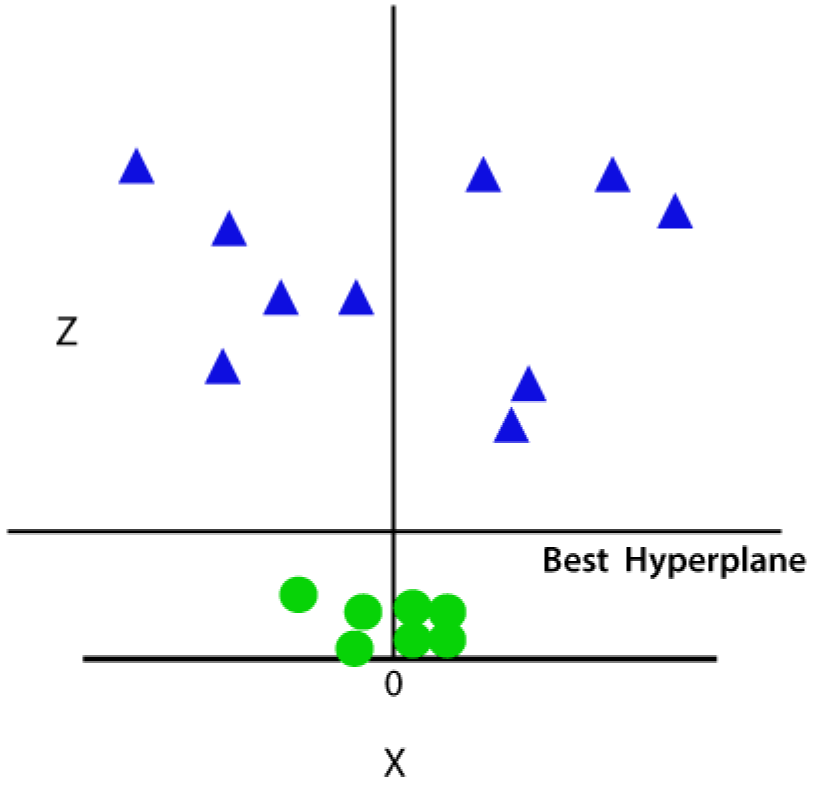
Non-Linear SVM

- By adding the third dimension, the sample space will become as below image:



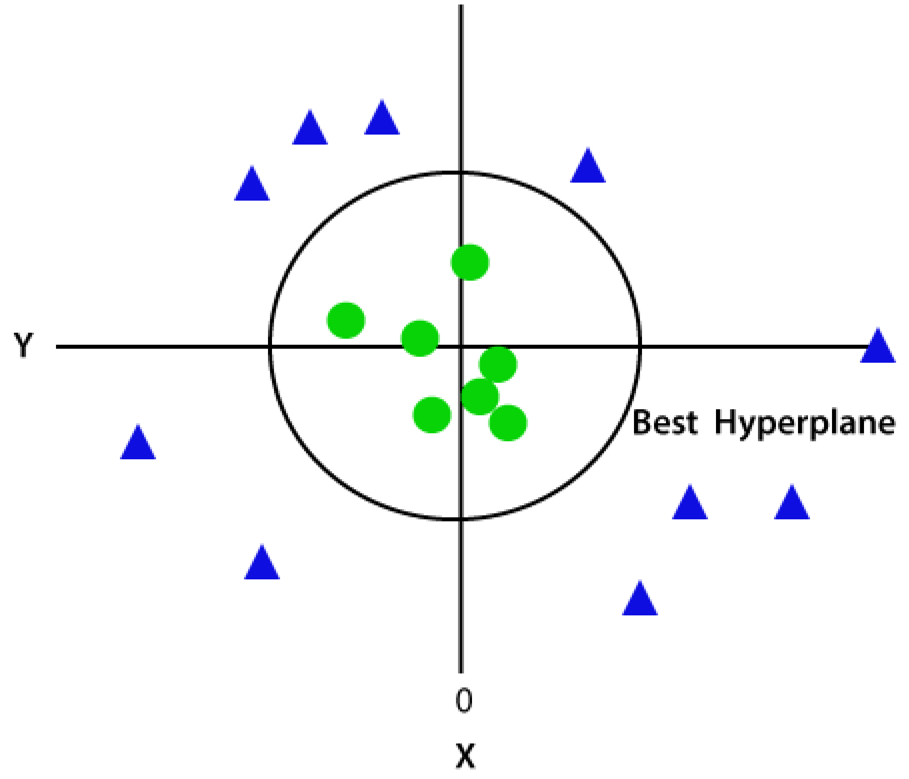
Non-Linear SVM

- So now, SVM will divide the datasets into classes in the following way.



Non-Linear SVM

- Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



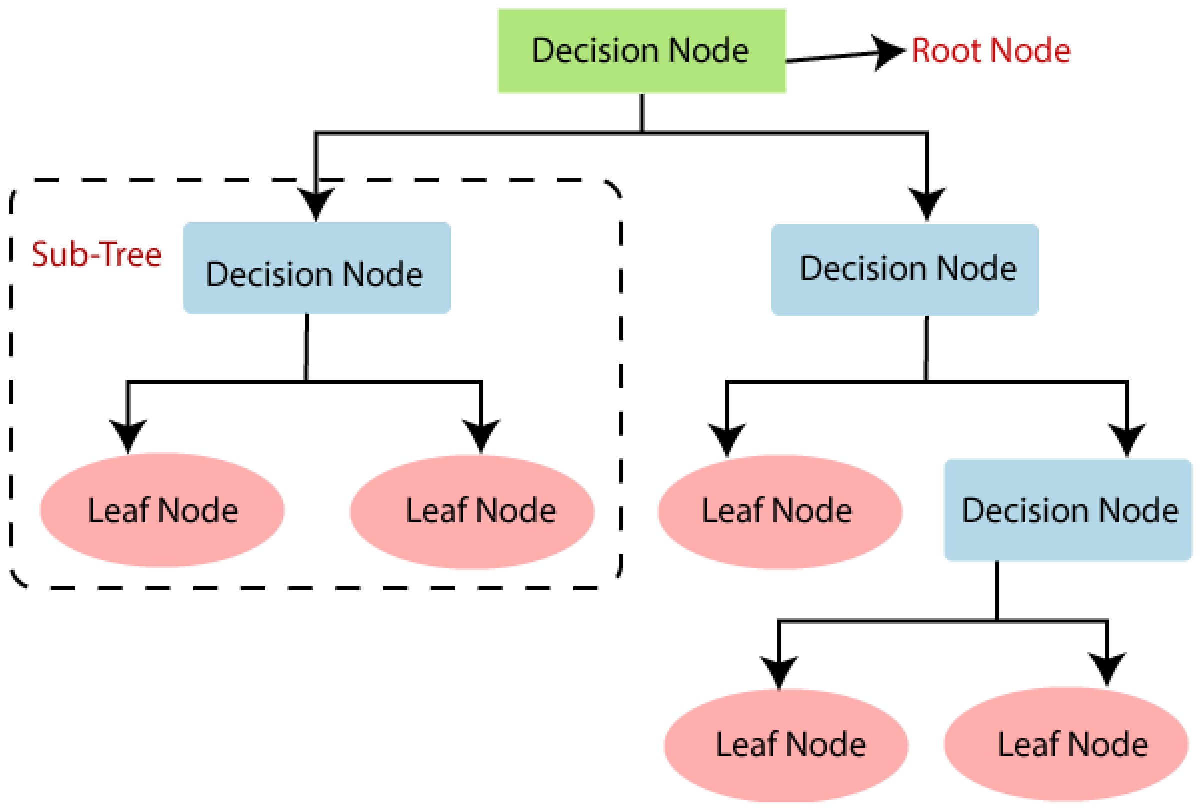
Decision Trees

Decision Trees

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems.
- It is a tree-structured classifier.
- Internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node**.
- Decision nodes are used to make any decision whereas Leaf nodes are the output of those decisions.

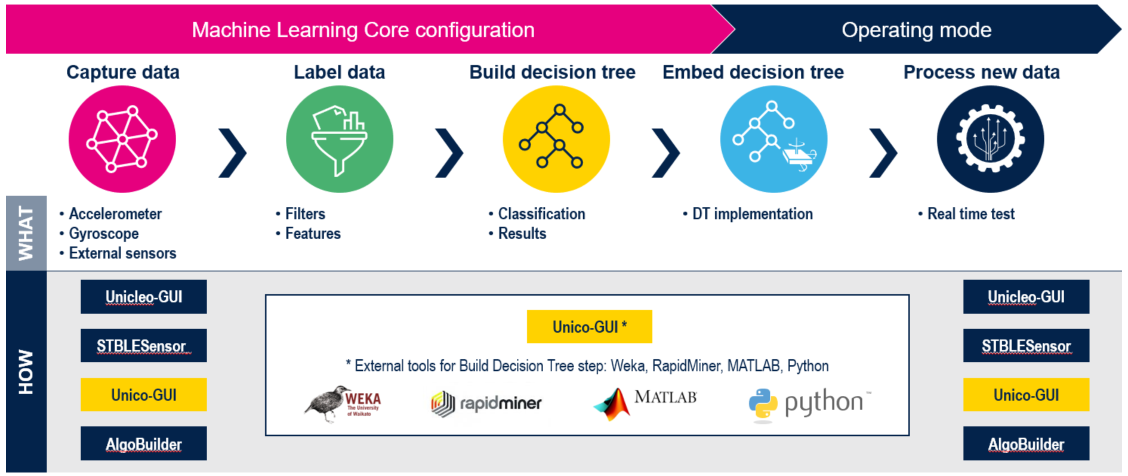
Decision Trees

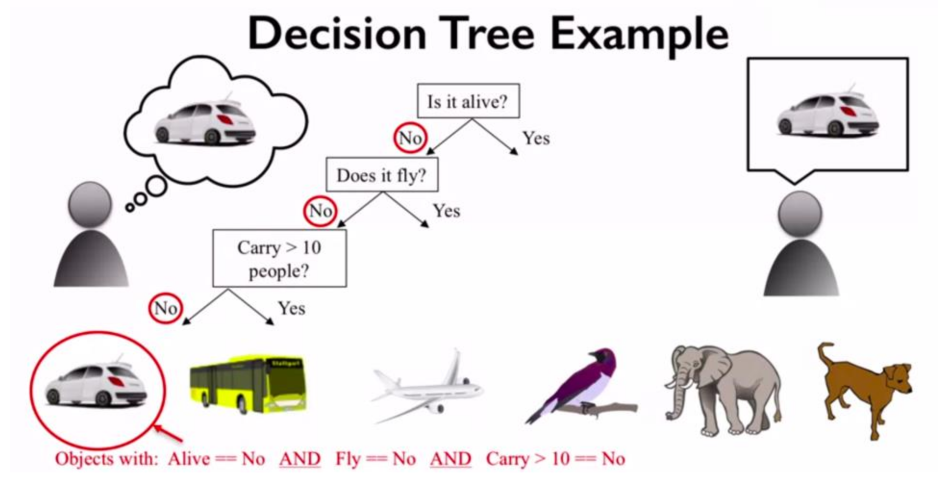
- It is called a decision tree because, similar to a tree, it starts with the root node.
- In order to build a tree, the Classification and Regression Tree (CART) algorithm is used.
- A decision tree simply asks a question and based on the answer (Yes/No), it further split the tree into subtrees.



Why Decision Trees?

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.





Decision Tree Terminologies

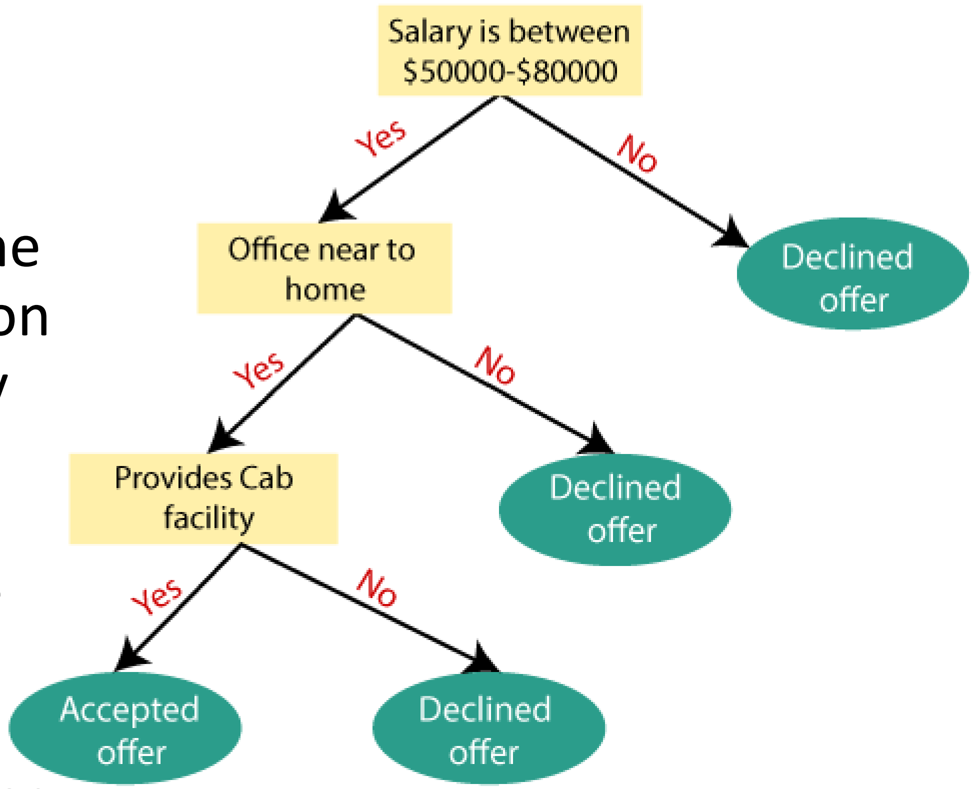
- Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- Branch/Sub Tree: A tree formed by splitting the tree.
- Pruning: Pruning is the process of removing the unwanted branches from the tree.
- Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

How Decision Tree algorithm Works?

- Step 1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step 2: Find the best attribute in the dataset using **Attribute Selection Measure (ASM)**.
- Step 3: Divide the S into subsets that contains possible values for the best attributes.
- Step 4: Generate the decision tree node, which contains the best attribute.
- Step 5: Recursively make new decision trees using the subsets of the dataset created in step 3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

How Decision Tree algorithm Works?

- Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer).



Pros & Cons

- Pros
  - It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
  - It can be very useful for solving decision-related problems.
  - It helps to think about all the possible outcomes for a problem.
  - There is less requirement of data cleaning compared to other algorithms.
- Cons
  - The decision tree contains lots of layers, which makes it complex.
  - It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
  - For more class labels, the computational complexity of the decision tree may increase.

Questions