# MACHINE LEARNING

Trinh Thanh Nam - Phan Tan Thinh


# ALGORITHM

01 Linear Regression

02 Gradient Boosting

03 Decision Tree Regression

04 ANN-Artificial Neural Networks

05 Random Forest Regression

06 LSTM-Long Short Term Memory

07 SVM-Support Vector Machine

# GOALS & OBJECTIVES

## ABOUT OUR DATA

We use the Melbourne housing dataset. This is a data set taken from real estate websites. The data set includes Address, Property type, Suburb, Sale method, Room, Price, Real estate agent, Date of sale and range distance from C.B.D(Central Business District).

## ABOUT OUR TARGET

With this dataset, we hope to build models that accurately predict housing prices in Melbourne based on the attributes in the dataset.

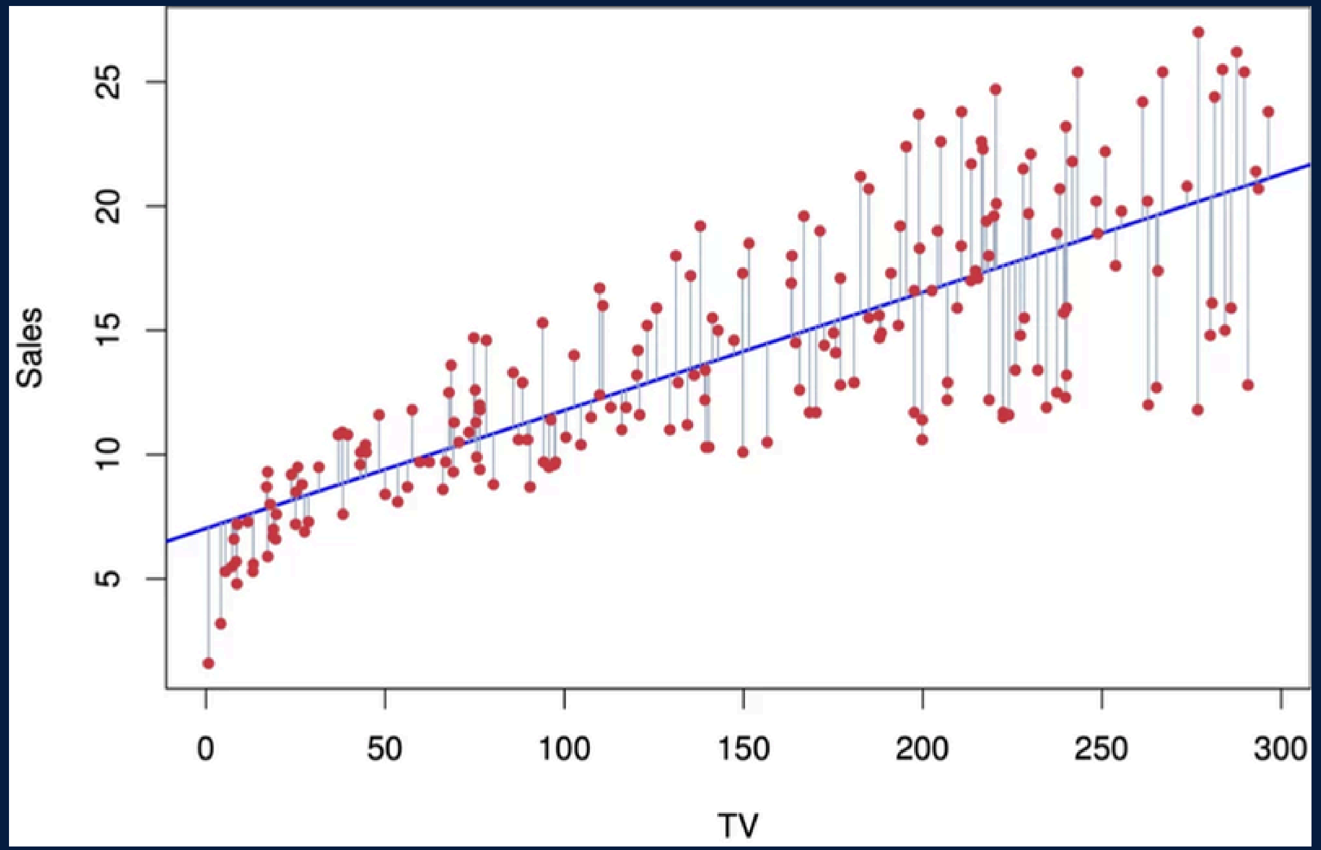
# 01 LINEAR REGRESSION

- WHAT IS LINEAR REGRESSION ?
- TYPES OF LINEAR REGRESSION
- THE GOAL OF LINEAR REGRESSION
- APPLICATION
- LINEAR REGRESSION IN PREDICTION PROBLEM

# 01 LINEAR REGRESSION

Linear Regression is a supervised learning algorithm in Machine Learning, it is a statistical method used to estimate the relationship between independent variables (input features) and dependent variables (output target).

Linear Regression assumes that the correlation between variables is linear, thereby finding the best linear function to represent this relationship. This algorithm predicts the value of the output variable from the values of the input variables.



# TYPES OF LINEAR REGRESSION

01

## SIMPLE LINEAR REGRESSION

This model has only one independent variable that describes the linear relationship between the dependent variable and the independent variable. The equation of Simple Linear Regression has the form as shown below

$$y = a + bx + \varepsilon$$

02

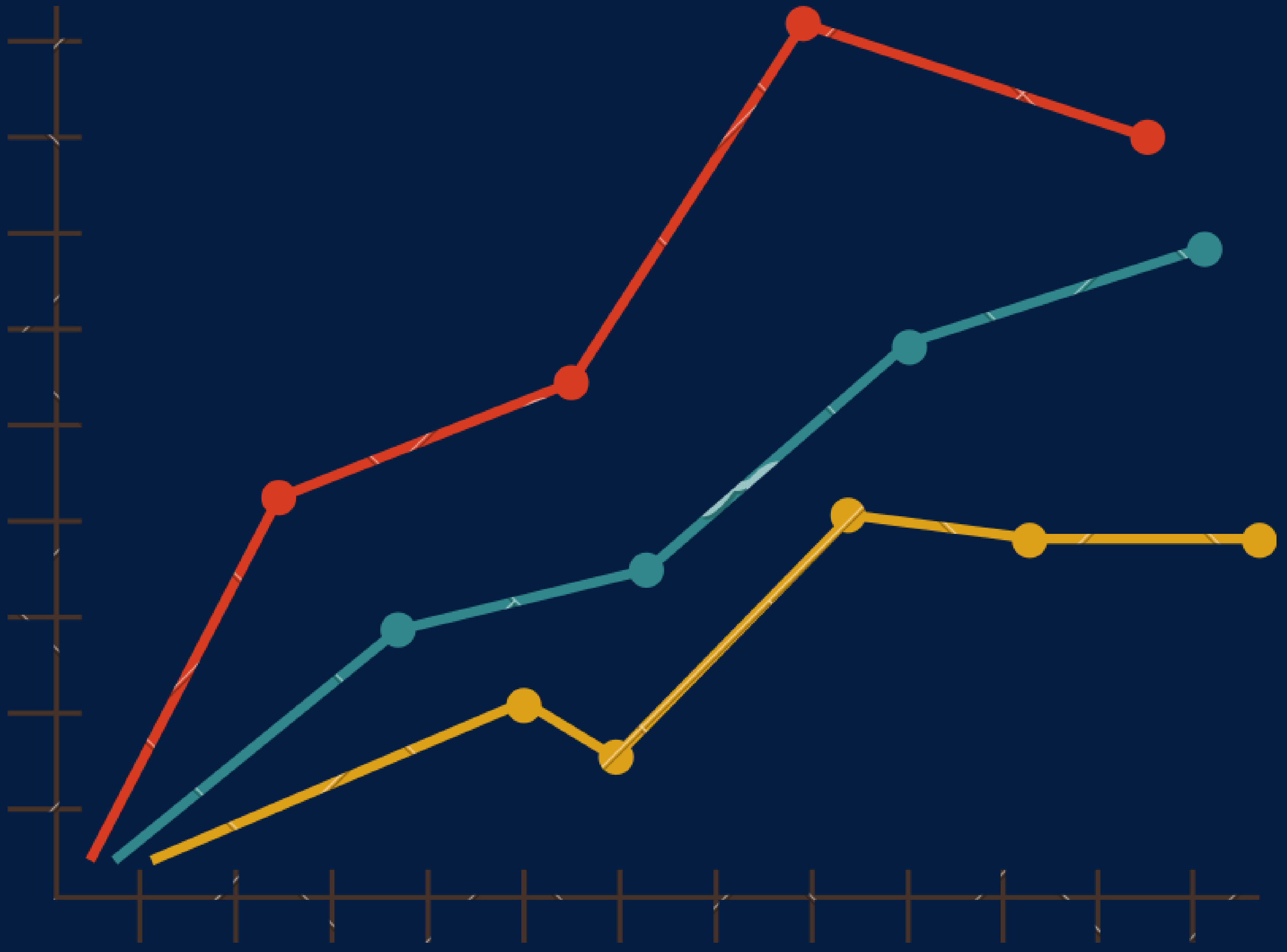
## MULTIPLE LINEAR REGRESSION

This model has more than one independent variable, demonstrating the linear relationship between independent variables and dependent variables. The Multiple Linear Regression equation has the form as shown below

$$y = a + b1x2 + b2x2 + \ldots + bnxn + \varepsilon$$

# THE GOAL OF LINEAR REGRESSION

The goal of Linear Regression is to find the slope and the intercept so that the linear prediction function achieves the smallest error. One common approach to estimate the coefficients is to use the Ordinary Least Squares method, in which we need to minimize the sum of squared errors.

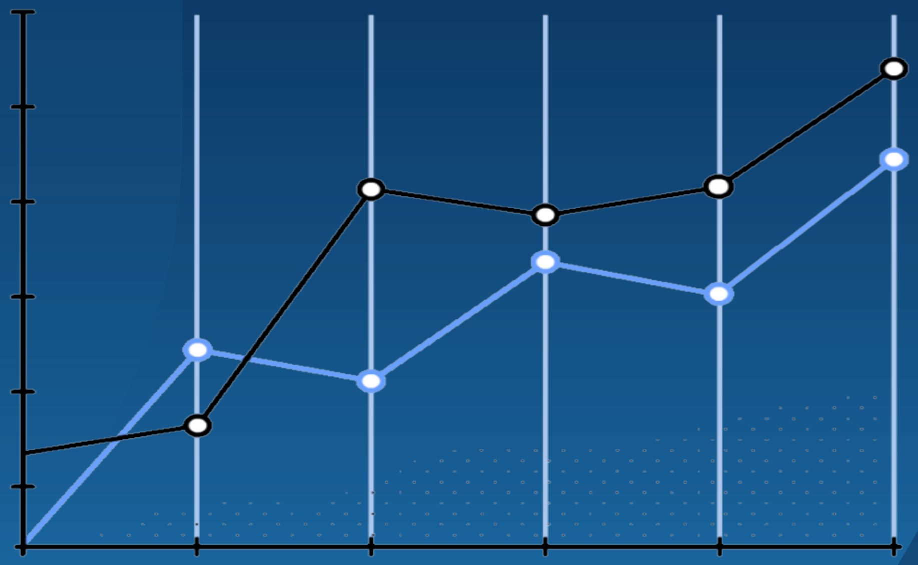


$$y=\beta 0+\beta 1x1+\beta 2x2+\ldots+\beta pxp$$

**OLS (Ordinary Least Squares)** : a popular method to estimate coefficients in linear regression models. The goal of OLS is to find the values of the coefficients β0,β1, ..,βp

$$\mathrm{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

# APPLICATION



- Price forecast: predict house prices, stock prices, fuel prices based on factors such as location, size, quality, supply and demand, ...
- Score prediction: predict student scores based on study time, effort, skills, teacher qualifications, ...
- Product forecasting: predict production output based on time, capacity, raw materials, labor, ...
- Time series analysis: predict trends and cycles of data series, such as real estate, weather, production trends,...

# LINEAR REGRESSION IN PREDICTION PROBLEM

01 Data Preprocessing

02 Feature Selection

03 Model Training use Ridge Regression

04 Model Evaluation

05 Visualizations

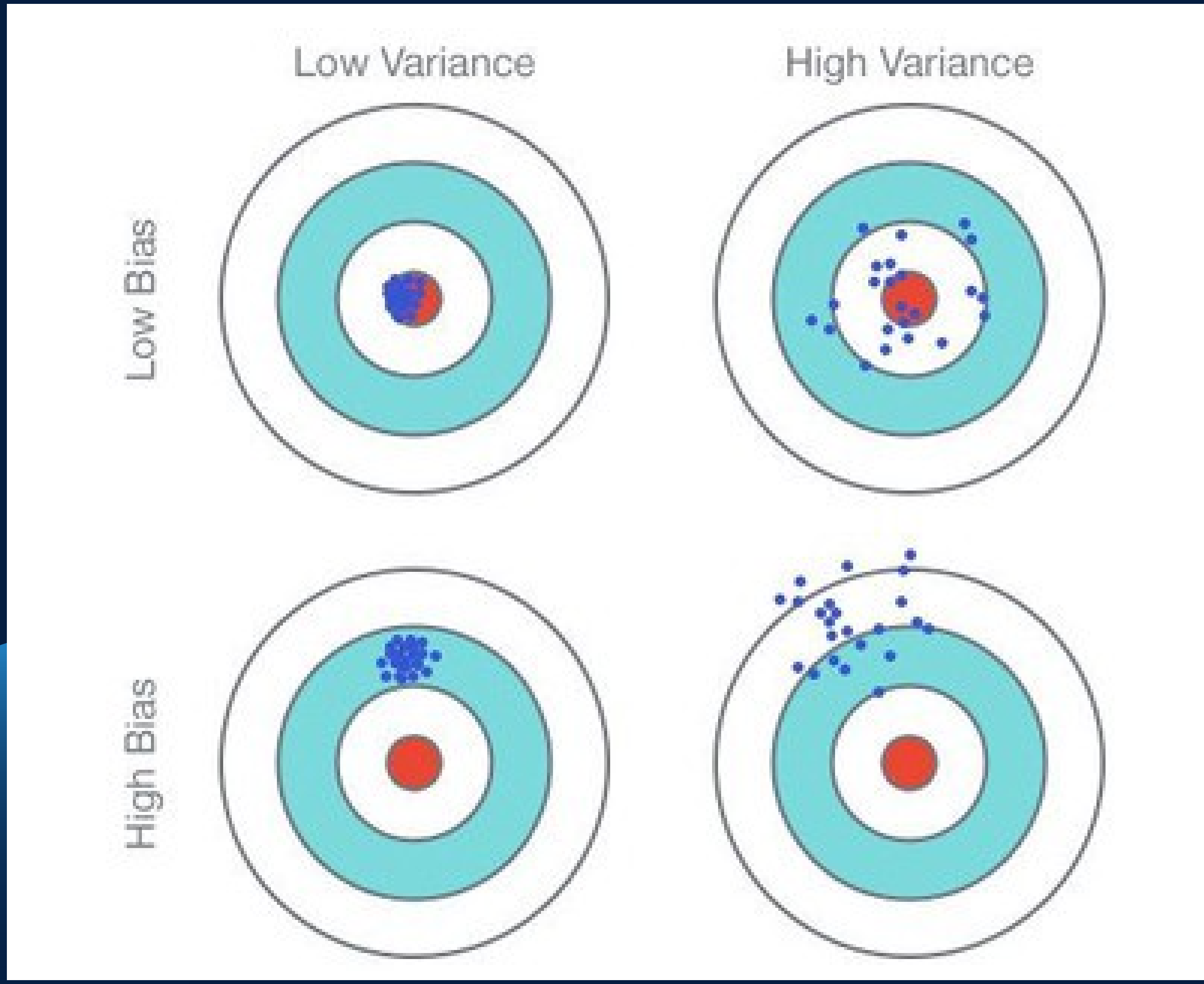
# 02 GRADIENT BOOSTING

- What is Boosting?
- Type of Boosting?
- Application
- Boosting in predict problem

# WHAT IS BOOSTING?

*"If one model cannot solve it by itself, let multiple models solve it together"*

To solve the problem of bias-variance trade off, the general solution will be to build many small models and then combine these models to create a superior model

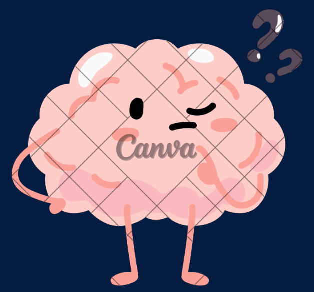



**Bagging**

- Build a large number of models on different subsamples from the training dataset.
- These models will be trained independently and in parallel, but their output will be averaged to produce the final result.

# WHAT IS BOOSTING?

*"If one model cannot solve it by itself, let multiple models solve it together"*

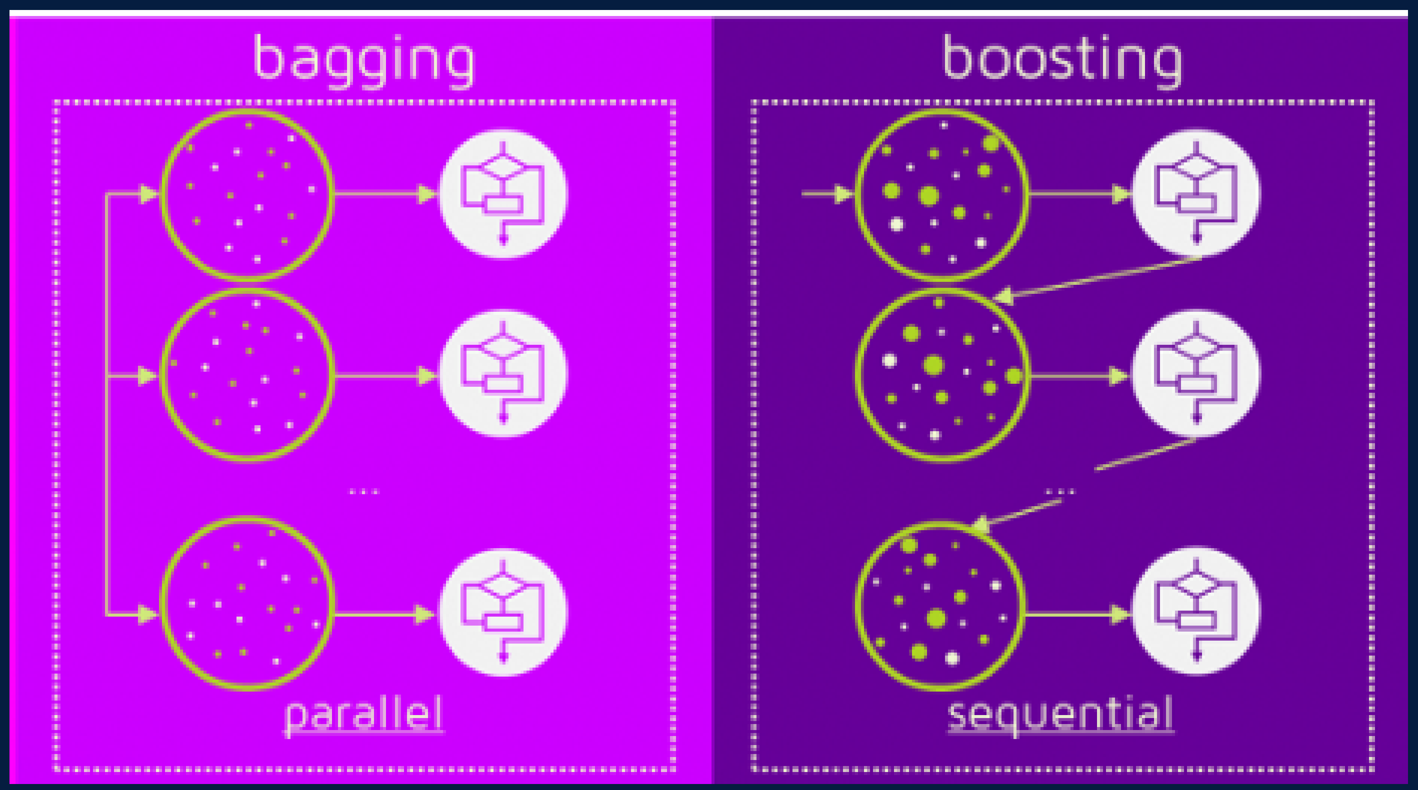



## Is Bagging the optimal solution?

- The models in Bagging all learn separately, without any relation or influence to each other, which in some cases can lead to bad results when models can learn the same result.
- Bagging cannot build weak models that can support each other and learn from each other to avoid making the mistakes of previous models.

## Boosting was born to solve these limitations

*The basic idea is that Boosting will create a series of weak models that learn to complement each other. In other words, in Boosting, later models will try to learn to limit the mistakes of previous models.*



# TYPE OF BOOSTING?

*Boosting limits mistakes from previous models by weighting newly added models based on different optimizations. Depending on the weighting method (how the models are fitted sequentially) and the way the models are aggregated, two types of Boosting are formed:*

01

### ADAPTIVE BOOSTING (ADABOOST)

AdaBoost trains new models based on reweighting current data points, to help new models focus more on data samples that are being mislearned, thereby reducing value. loss of the model

02

### GRADIENT BOOSTING

Gradient Boosting is a generalization of AdaBoost, with the ability to use more flexible loss functions, optimize with gradient descent, and be more customizable, helping it solve more complex problems.

# TYPE OF BOOSTING?

*Both AdaBoost and Gradient Boosting build algorithms to solve the following optimization problem:*

$$\min_{c_n=1:N, w_n=1:N} L(y, \sum_{n=1}^{N} c_n w_n))$$

L : loss function value

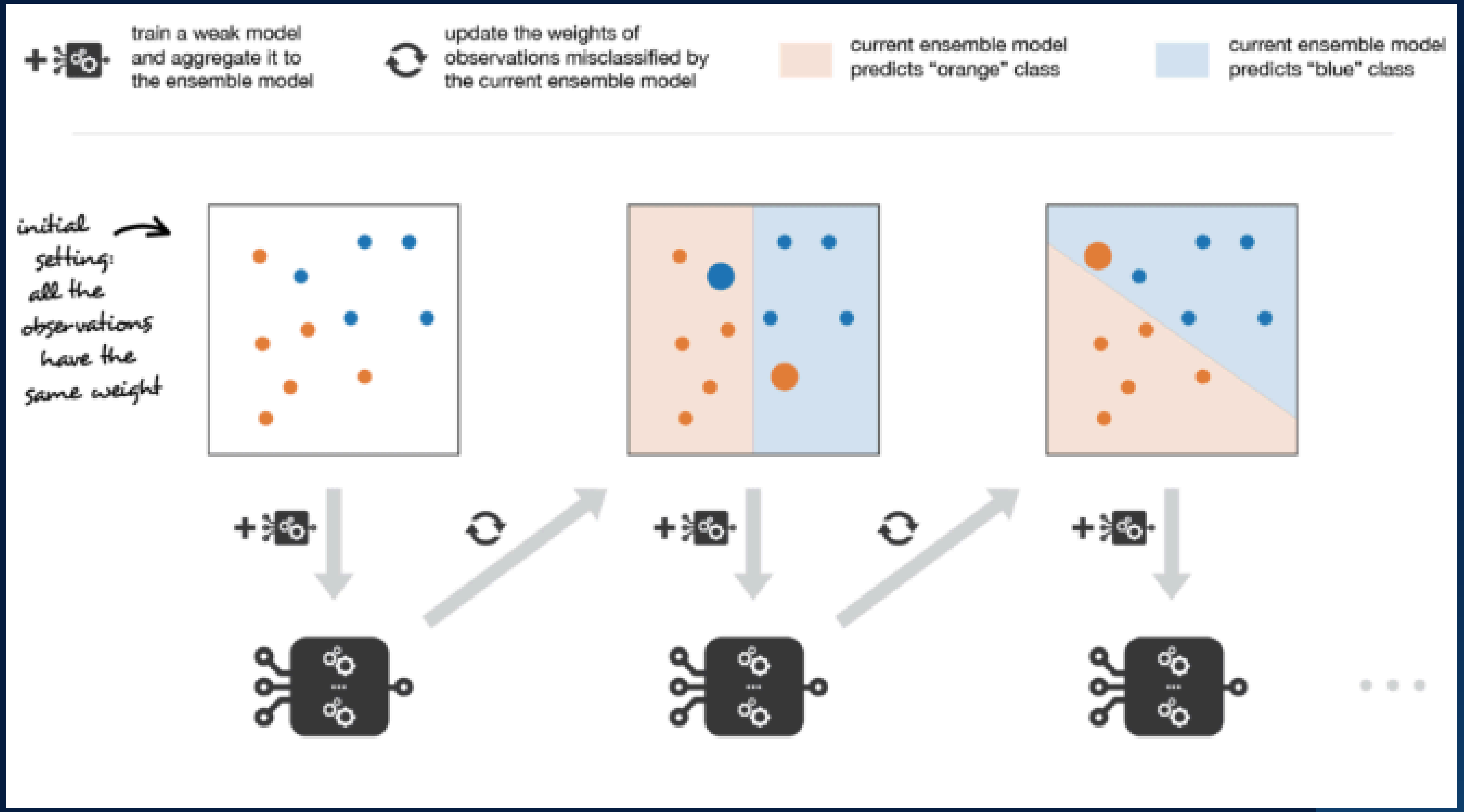
y : label

Cn : confidence score of the nth weak learner (also known as weight)

Wn: nth weak learner

# 01 ADAPTIVE BOOSTING (ADABOOST)

- Initialize the initial weight to be equal (equal to 1/N) for each data point
- At ith loop:
  - Train model wi (weak learner) has just been added
  - Calculate the loss (error) value, from there calculate the confidence score value ci of the model just trained
  - Update main model W=W+ci * wi
  - Finally, re-weight the data points (Incorrectly guessed data points -> increase weight, correctly guessed data points -> decrease weight).
- Then repeat with the loop adding the next model i + 1.



# 02 GRADIENT BOOSTING

Gradient Boosting is a generalized form of AdaBoost

The problem needs optimization

$$\min_{c_n, w_n} L(y, W_{n-1} + c_n w_n))$$

**Gradient Descent**

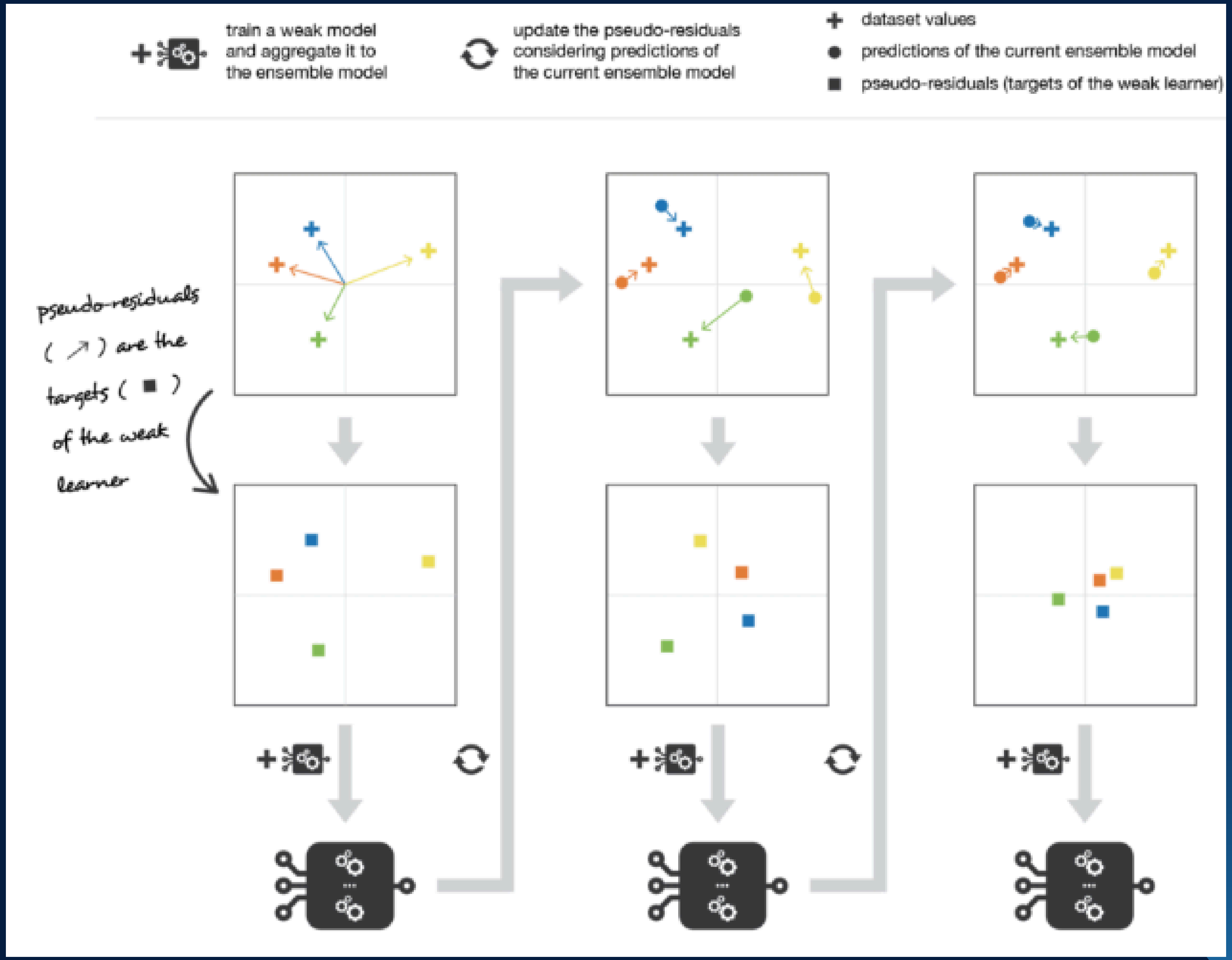
$$W_n = W_{n-1} - \eta \frac{\partial}{\partial w} L(W_{n-1})$$

We can see the following related relationships

$$c_n w_n \approx -\eta \frac{\partial}{\partial w} L(W_{n-1})$$

where wn is the model added next.At that time, the new model needs to learn to fit to enter the value:

$$-\eta \frac{\partial}{\partial w} L(W_{n-1})$$ **(pseudo-residuals)**



# COMPARE

| Characteristic | AdaBoost | Gradient Boosting |
|---|---|---|
| Principle | Update misclassified data sample weights | Learn from residuals and improve models via gradient descent |
| Loss function | Exponential loss | Many different loss functions can be used |
| Model update method | Add weight to misclassified samples | Add a model to learn residuals (errors) from the previous model |
| Optimization method | Weighted voting | Gradient descent |

# APPLICATION

## Finance

Credit classification, stock price forecast, credit risk assessment.

## Medical

Diagnose diseases (cancer, cardiovascular disease), classify medical tests, predict health problems

## Analyze customer data

Predict customer churn (churn prediction), suggest products based on shopping behavior.

## Transport and traffic

Predict traffic congestion, forecast delivery time in the logistics industry.

## Natural Language Processing (NLP)

Classify text (spam, topic), predict answers in chatbot system.

# BOOSTING IN PREDICT PROBLEM

01 Data Preprocessing

02 Feature Selection

03 Boosting Model

04 Hyperparameter Tuning

05 Model Evaluation

06 Visualizations

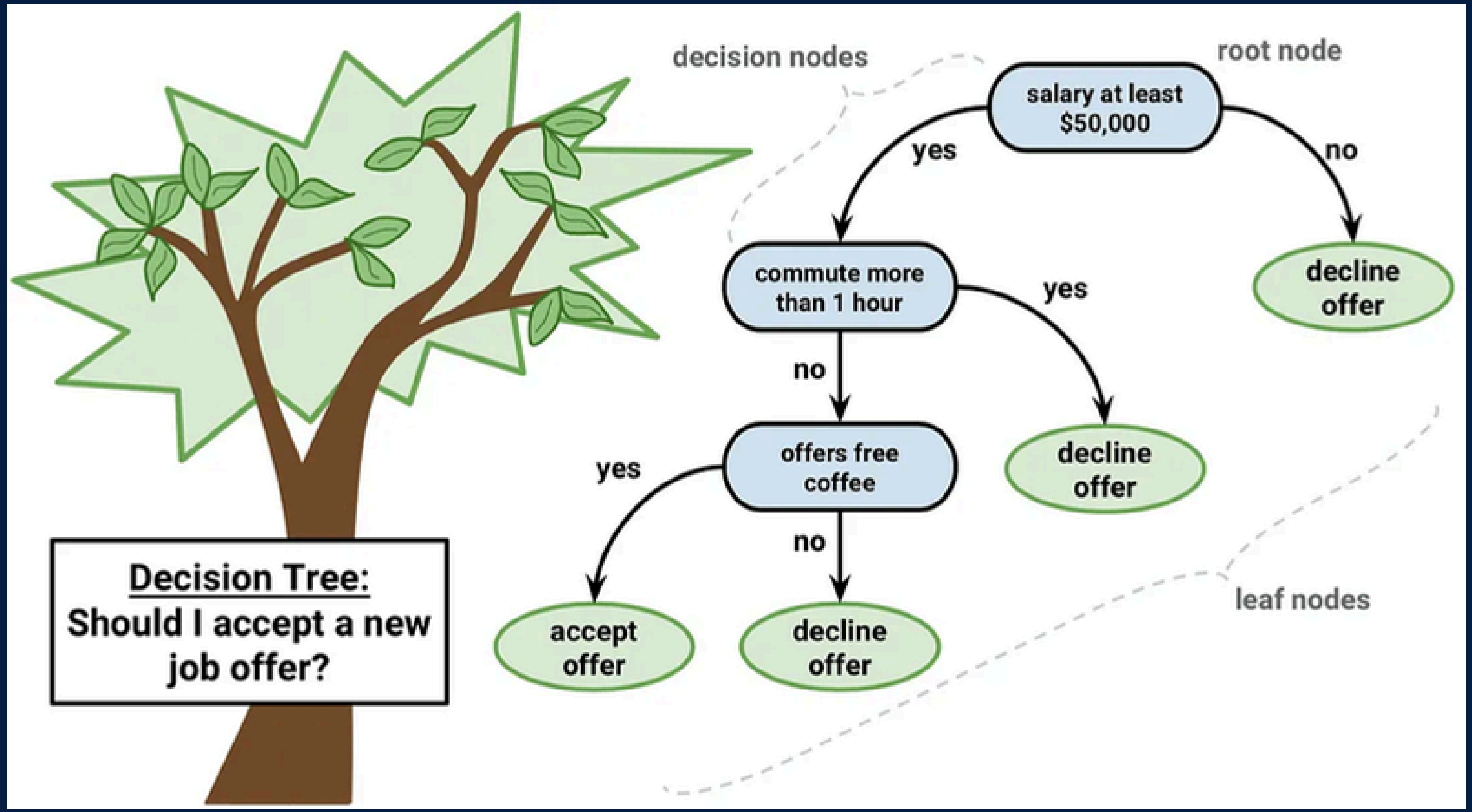
# 03 DECISION TREE REGRESSION

- What is a Decision Tree?
- Types of Decision Trees
- How do Decision Trees work?
- Application
- Decision Tree Regression in predict problem

# WHAT IS A DECISION TREE?

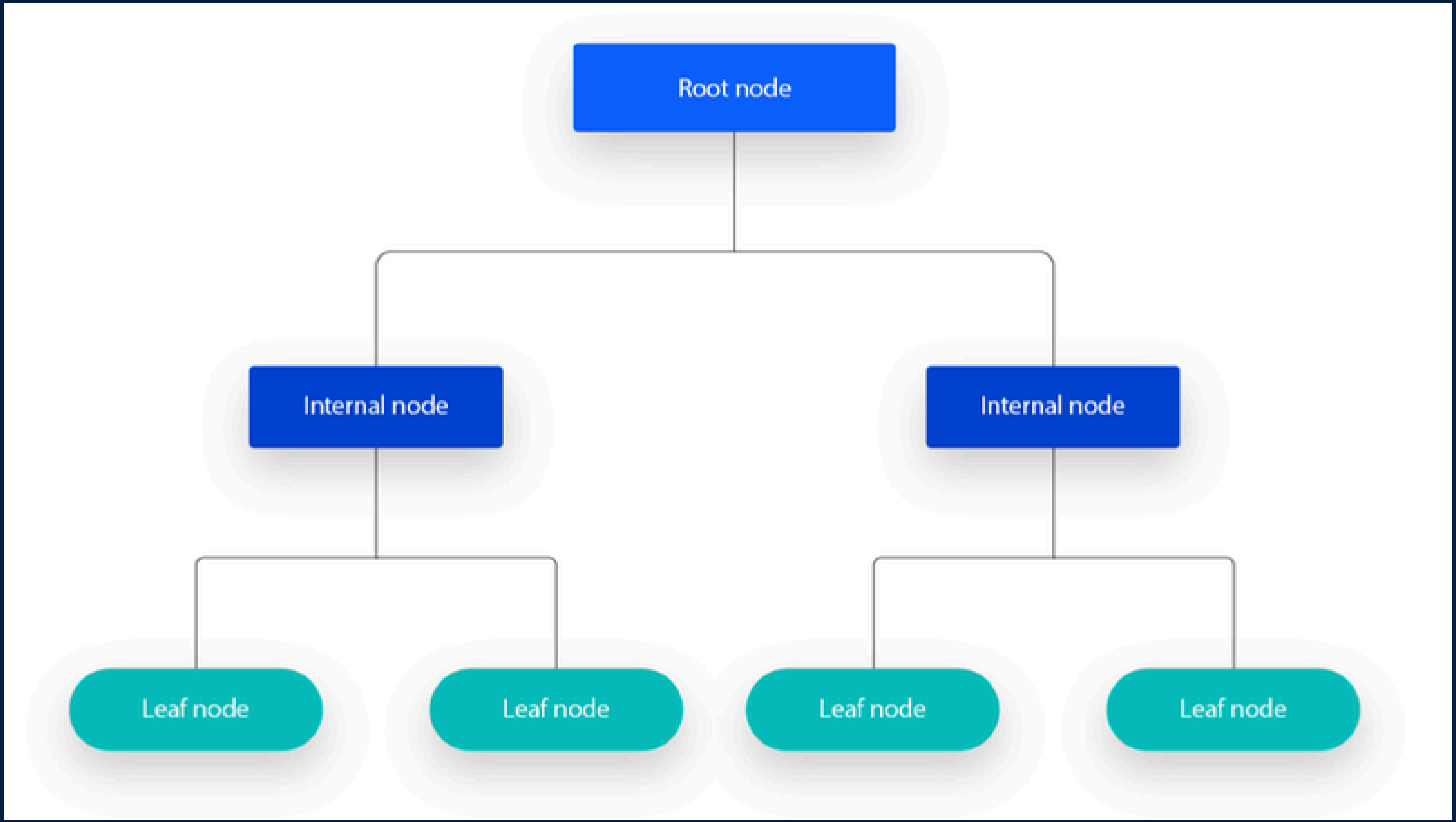
- A decision tree is a supervised machine learning algorithm. It is used in both classification and regression algorithms.

- The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data.



# STRUCTURE OF A DECISION TREE

1. **Root Node**: Represents the entire dataset and the initial decision to be made.
2. **Internal Nodes**: Represent decisions or tests on attributes. Each internal node has one or more branches.
3. **Branches**: Represent the outcome of a decision or test, leading to another node.
4. **Leaf Nodes**: Represent the final decision or prediction. No further splits occur at these nodes.



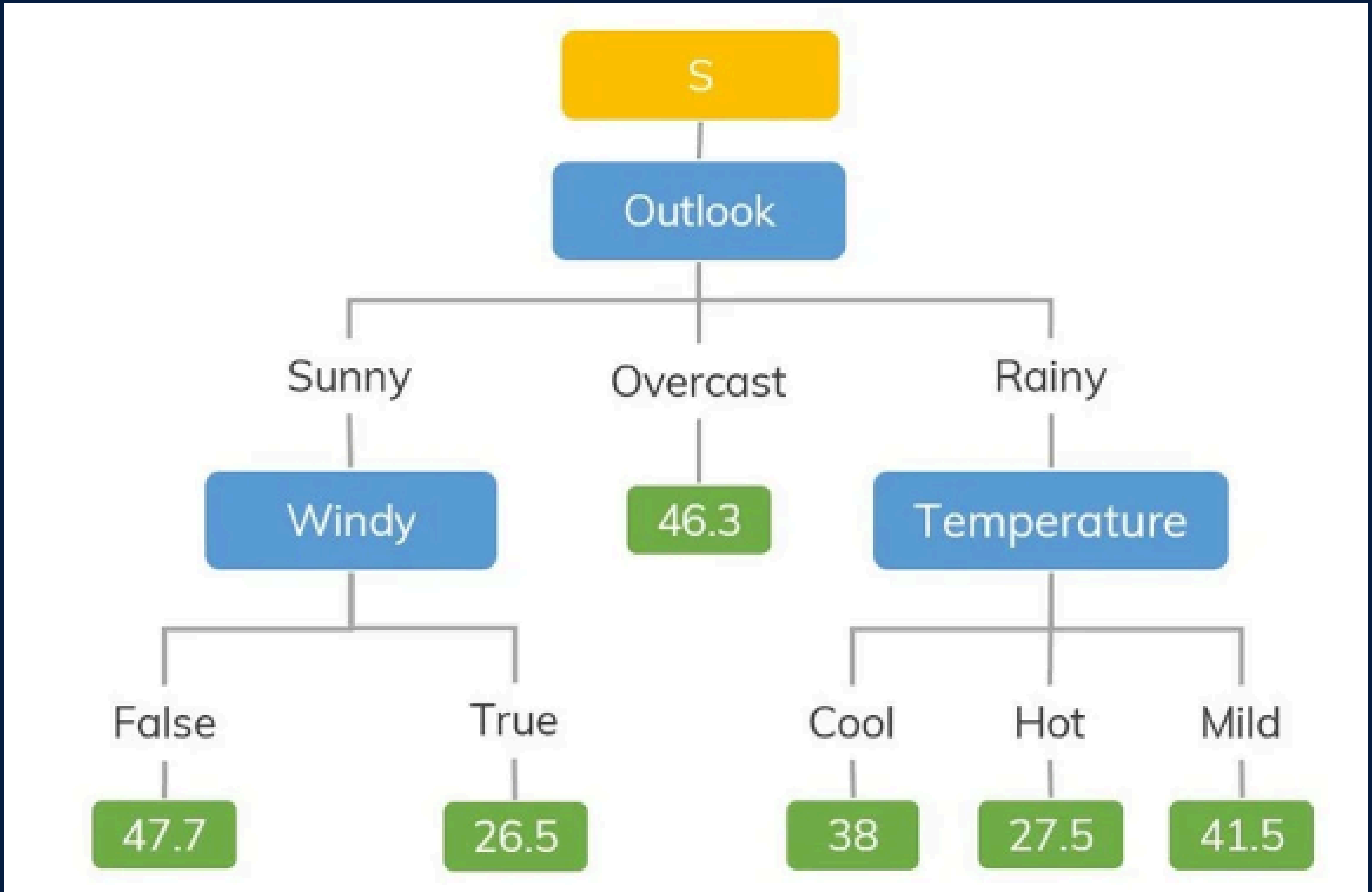

*Structure of a Decision Tree*

# TYPES OF DECISION TREES

The types of decision trees are based on the type of target variable we have. It can be of two types:

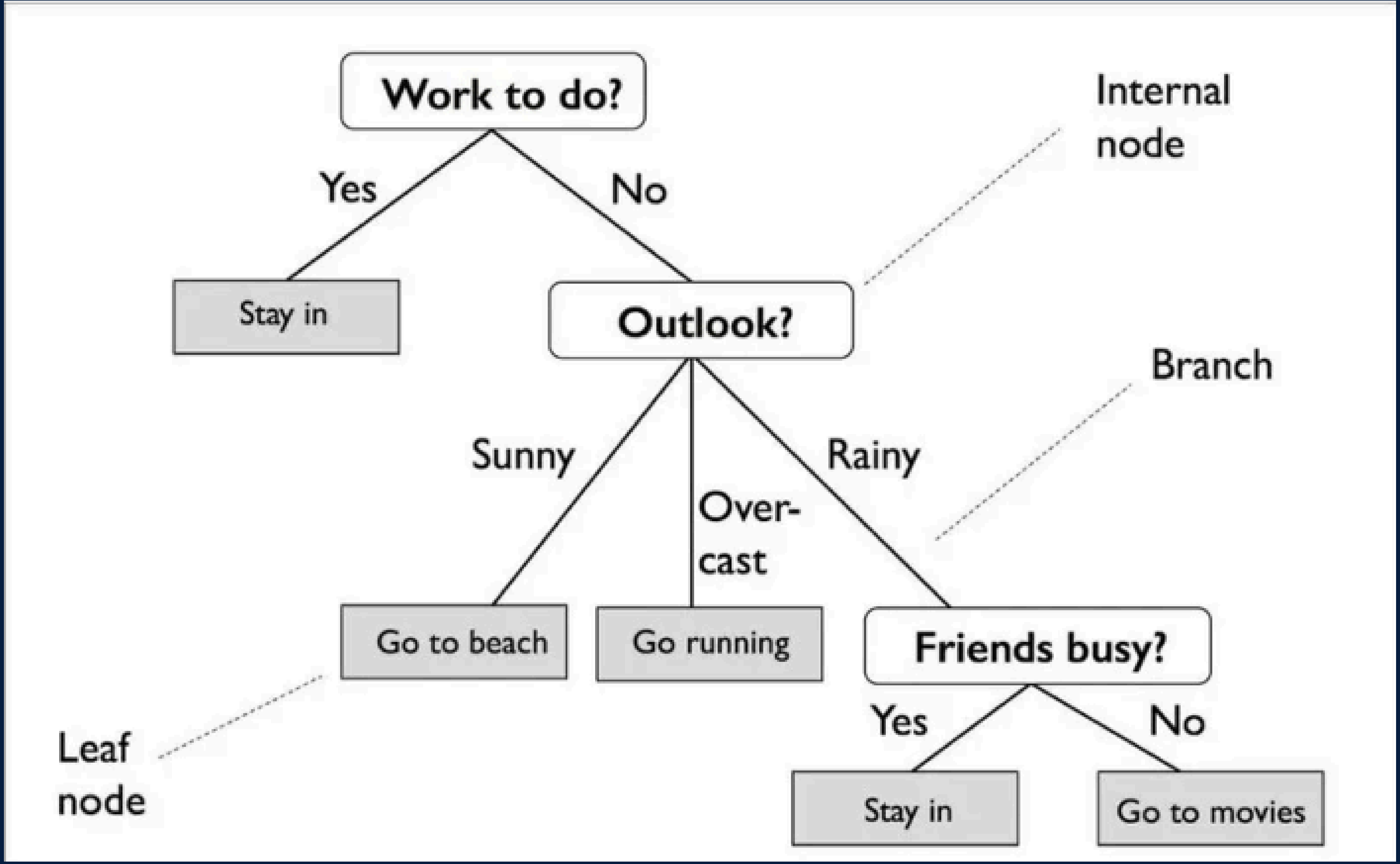
- Decision Tree which has a categorical target variable then it called a Classification Tree.
- Decision Tree has a continuous target variable then it is called a Regression Tree

# TYPES OF DECISION TREES




Regession Tree

# TYPES OF DECISION TREES




Classification Tree

# HOW DO DECISION TREES WORK?

- Step 1: Start the tree with the root node (S), which contains the complete data set.
- Step 2: Find the best attribute in the data set using Attribute Selection Measurement (ASM).
- Step 3: Divide S into subsets containing possible values for the best attributes.
- Step 4: Create a decision tree node containing the best attribute.
- Step 5: Recursively create a new decision tree using subsets of the dataset created in step -3. Continue this process until you reach a stage where you cannot classify more nodes and call the last node a leaf node.

# HOW DO DECISION TREES WORK?

*Some classic algorithms used in Decision Trees:*

01

**CART ( CLASSIFICATION & REGRESSION TREE )**

CART (Classification and Regression Trees) is a machine learning algorithm for building decision trees, which can be applied to both classification and regression problems.

02

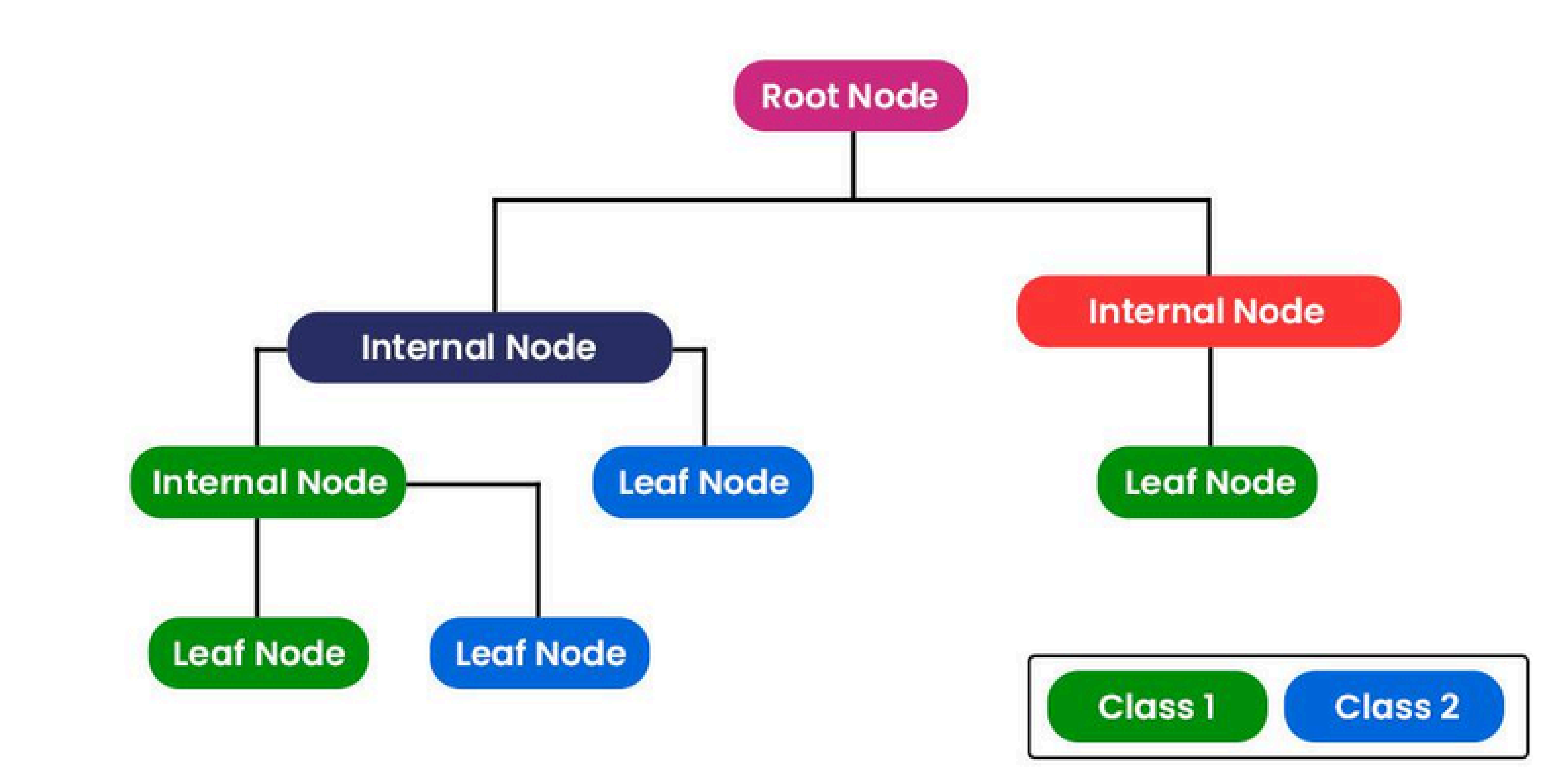
**C4.5**

Algorithm C4.5 is an extension of the ID3 algorithm. C4.5 is capable of handling both continuous quantitative data and qualitative data, a typical Decision Tree algorithm.

# 01 CART ( CLASSIFICATION & REGRESSION TREE )

The CART algorithm works via the following process:

- The best-split point of each input is obtained.
- Based on the best-split points of each input in Step 1, the new "best" split point is identified.
- Split the chosen input according to the "best" split point.
- Continue splitting until a stopping rule is satisfied or no further desirable splitting is available.



# 01 CART ( CLASSIFICATION & REGRESSION TREE )

For the classification task, the CART algorithm uses Gini Impurity to divide the data into a decision tree. It does this by finding the best coherence for the child nodes, with the help of the Gini index criterion.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Gini Index

pi is the probability of an object being classified to a particular class.

# 01 CART ( CLASSIFICATION & REGRESSION TREE )

How does CART for Classification Work?

CART for classification works by recursively splitting the training data into smaller and smaller subsets based on certain criteria.For classification tasks, CART uses Gini impurity

- Gini Impurity- Gini impurity measures the probability of misclassifying a random instance from a subset labeled according to the majority class. Lower Gini impurity means more purity of the subset.

- Splitting Criteria- The CART algorithm evaluates all potential splits at every node and chooses the one that best decreases the Gini impurity of the resultant subsets. This process continues until a stopping criterion is reached, like a maximum tree depth or a minimum number of instances in a leaf node.

# 01 CART ( CLASSIFICATION & REGRESSION TREE )

How does CART works for Regression?

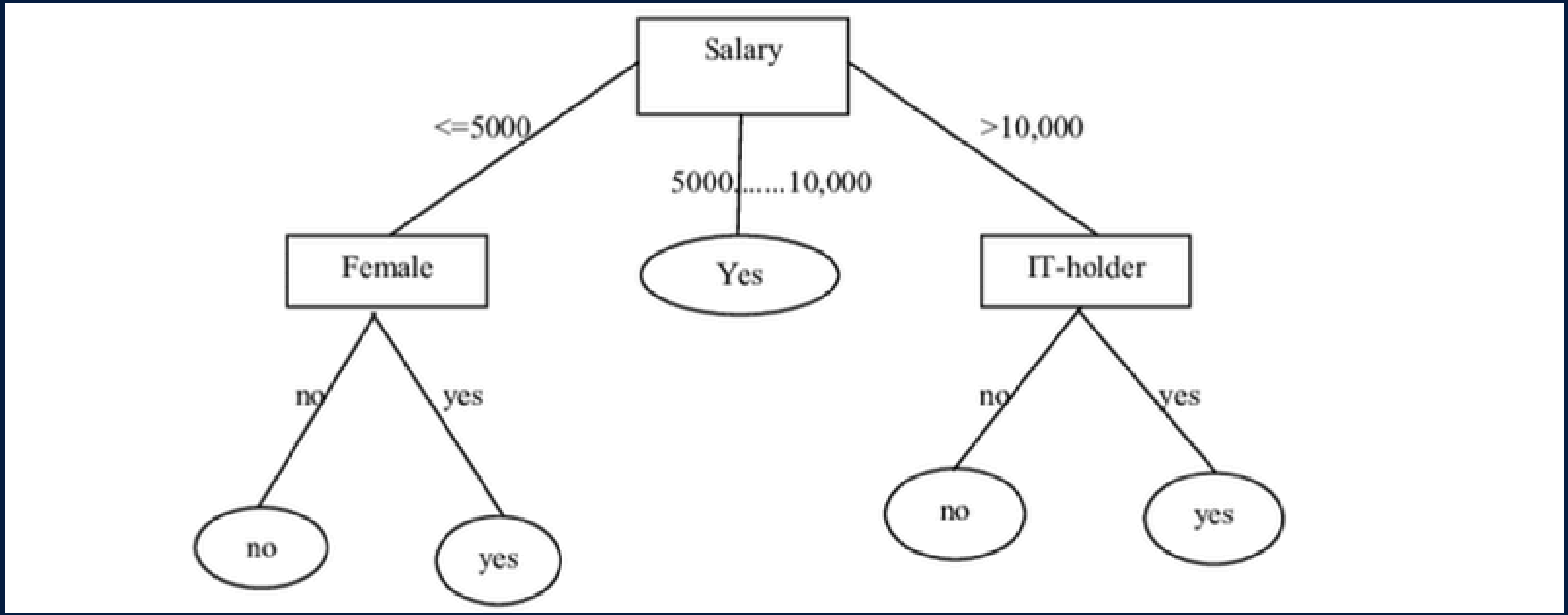
Regression CART works by splitting the training data recursively into smaller subsets based on specific criteria. The objective is to split the data in a way that minimizes the residual reduction in each subset.

- Residual Reduction- Residual reduction is a measure of how much the average squared difference between the predicted values and the actual values for the target variable is reduced by splitting the subset. The lower the residual reduction, the better the model fits the data.

- Splitting Criteria- CART evaluates every possible split at each node and selects the one that results in the greatest reduction of residual error in the resulting subsets. This process is repeated until a stopping criterion is met, such as reaching the maximum tree depth or having too few instances in a leaf node.

# 01 C4.5 ALGORITHM

- The C4.5 algorithm is one of the machine learning algorithms used to build Decision Trees, and is an improved version of the ID3 algorithm, with a number of outstanding features to improve performance and reduce minimize overfitting problem. C4.5 is mainly used in classification problems, where the goal is to group objects into different classes.



# 01 C4.5 ALGORITHM

## Entropy

If we are given a probability distribution P = (p1,p2,..., pn) and a sample S then the Information carried by this distribution, also called the entropy of P is giving by:

$$Entropy = \sum_{i=1}^{n} p_i . log_2\left(\frac{1}{p_i}\right) = \sum_{i=1}^{n} -p_i . log_2(p_i)$$

Entropy is a measure of the disorder of a system. If a set has high entropy, it means that the set is very "chaotic" or uncertain, meaning that the elements in the set have an uneven distribution among classes or values. Conversely, if a set has low entropy, it means that the elements in the set are almost identical in class or value.

# 01 C4.5 ALGORITHM

## Information Gain

Information Gain (IG) is a metric used in decision tree algorithms, such as ID3, C4.5, and CART, to measure the information gain when dividing data by a specific feature. . It helps determine which features will provide the best division of the data, i.e. help minimize the uncertainty (entropy) the most after division.

*It defines the gain for a test T and a position p*

$$Gain(p,T) = Entropie(p) - \sum_{j=1}^{n} (p_j \times Entropie(p_j))$$

(pj) is the set of all possible values for attribute T

# 01 C4.5 ALGORITHM

## Gain ratio

Gain Ratio (GR) is an improvement of Information Gain (IG), used in the C4.5 algorithm to overcome the disadvantage of Information Gain in favoring features with many discriminant values. This helps reduce overfitting and makes the decision tree more general and more effective in classifying data.

$$GainRatio(p,T) = \frac{Gain(p,T)}{SplitInfo(p,T)}$$

where SplitInfor measures the degree of "separation" that a feature creates when dividing data into subgroups

$$SplitInfo(p,test) = -\sum_{j=1}^{n} P'\left(\frac{j}{p}\right) \times log\left(P'\left(\frac{j}{p}\right)\right)$$

# HOW TO AVOID OVERFITTING IN DECISION TREES?

## PRE-PRUNING

Adjust parameters before building the tree (like max_depth, min_samples_split, etc.)

## POST-PRUNING

Once the decision tree is built, unnecessary branches (those with low accuracy or that do not contribute significantly) are removed.

# APPLICATION

## Banking

Decision trees can be used to analyze credit risk, evaluate customers' ability to repay debt, thereby helping financial institutions decide to grant credit.

## Medical

Decision trees can assist doctors in diagnosing diseases based on symptoms and information about the patient.

## Logistics

Decision trees can help in supply chain optimization, product demand prediction, and inventory management.

## Finance

Decision trees can help analyze and predict financial factors such as stock prices, company profits, etc.

# DECISION TREE IN PREDICT PROBLEM

01 Data Preprocessing

02 Data Splitting

03 Build a Decision Tree Regressor model

04 Optimize the model with Grid Search and Cross-Validation

05 Model Evaluation

06 Visualizations

04

# ANN-ARTIFICIAL NEURAL NETWORKS

- WHAT IS ANN ?
- TYPES OF ARTIFICIAL NEURAL NETWORKS
- HOW DO ARTIFICIAL NEURAL NETWORKS LEARN?
- APPLICATION
- ANN IN PREDICT PROBLEM

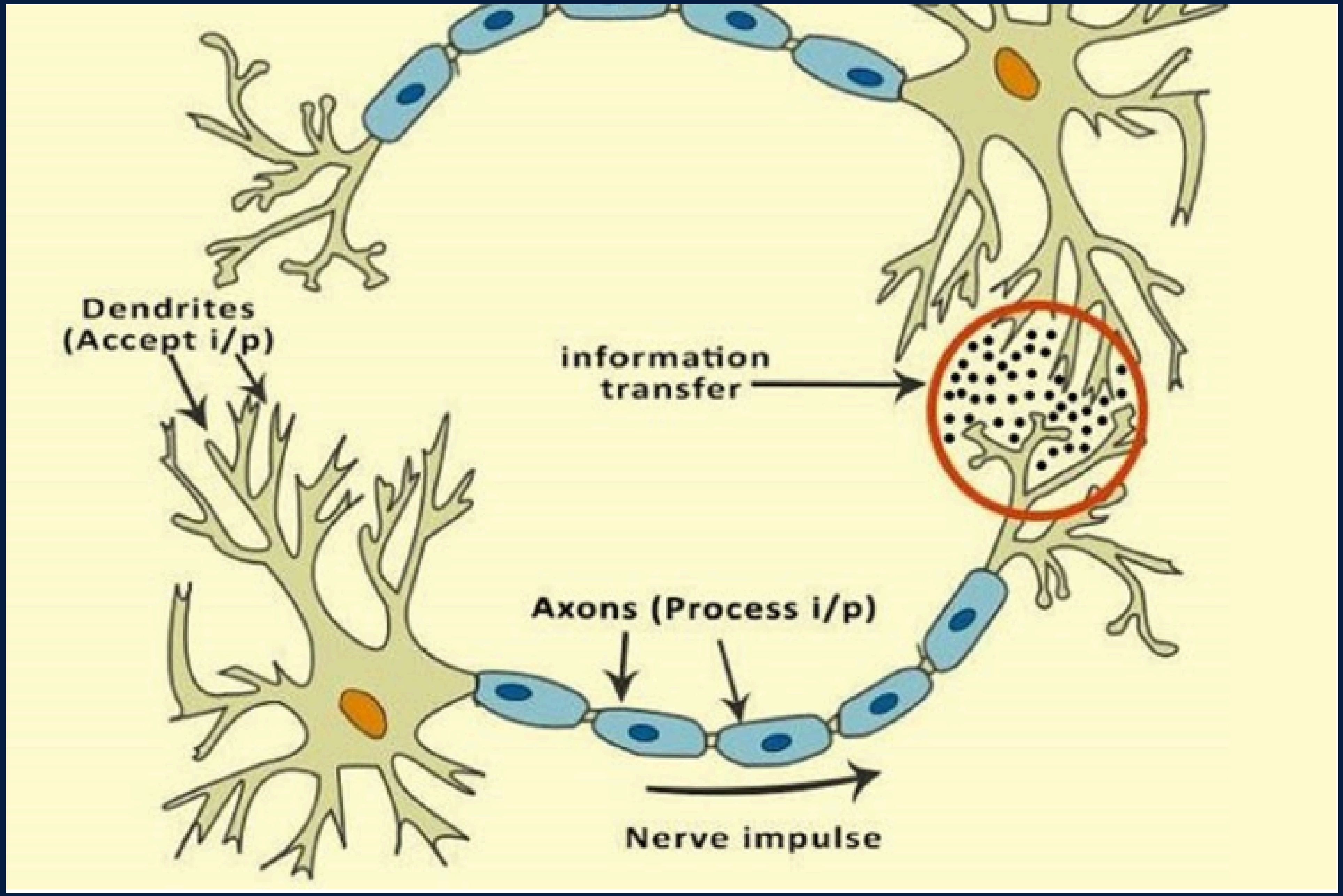
# WHAT IS ANN ?

- A neural network is a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning (ML) process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain. It creates an adaptive system that computers use to learn from their mistakes and improve continuously.

- Like human brains, artificial neural networks are made up of neurons that are connected like brain cells. These neurons process and receive information from nearby neurons before sending it to other neurons.

# STRUCTURE

ANN gets its idea from how the human brain works - creating the right connections.
Therefore, ANN used silicon and wires to make living neurons and dendrites for themselves.

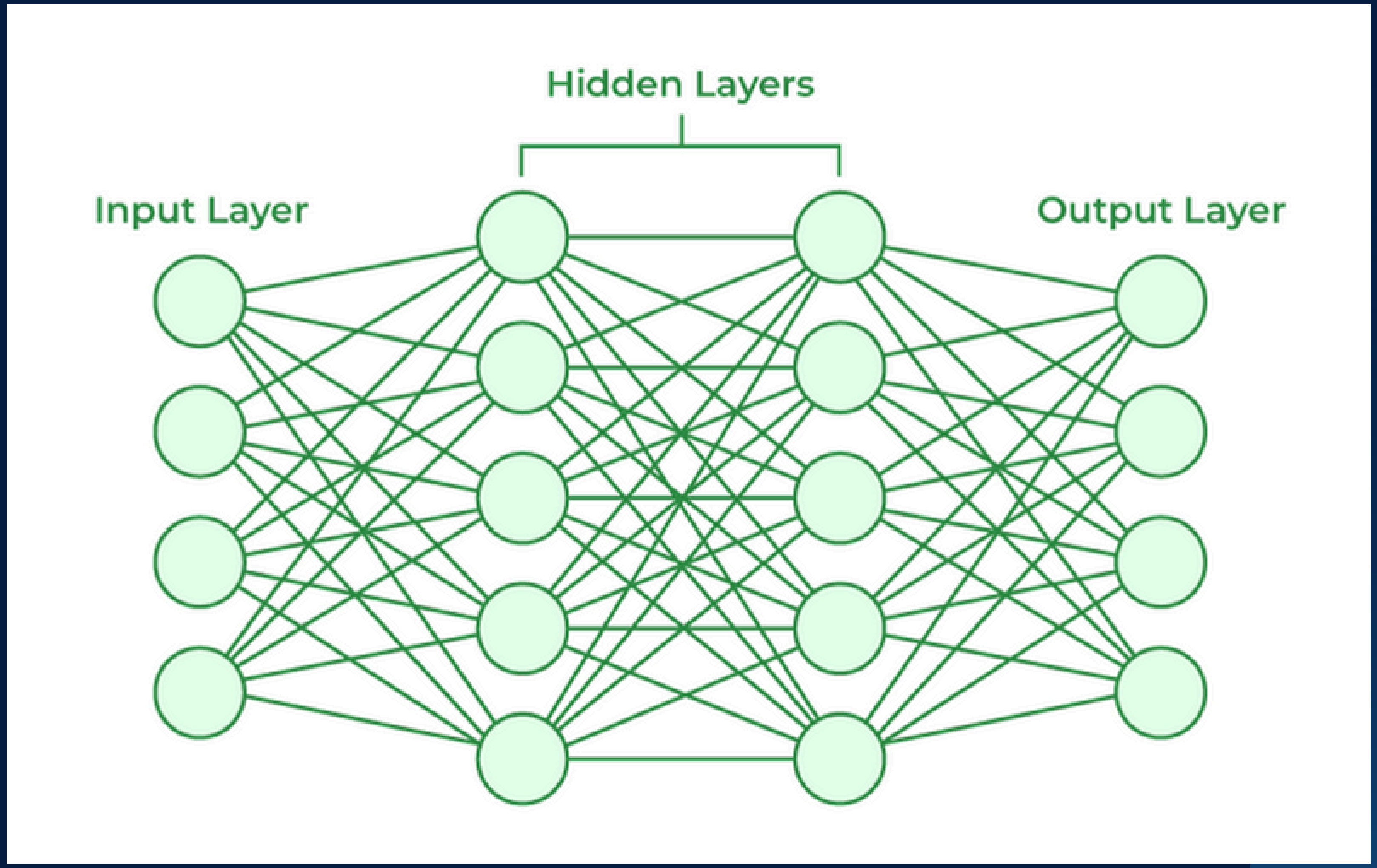


# STRUCTURE

There are three layers in the network architecture:

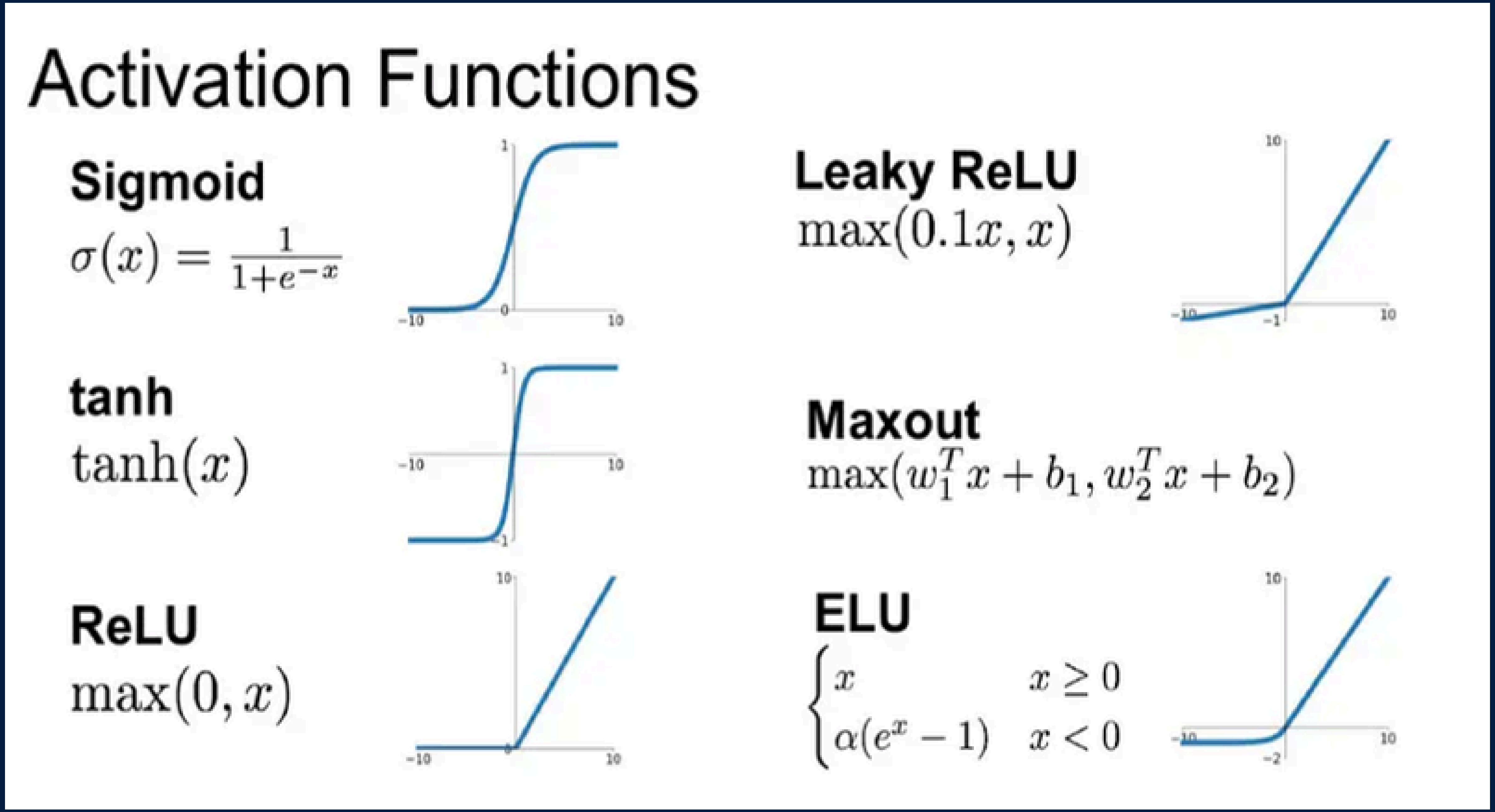
- **Input layer**
- **Output layer**: includes only one layer
- **Hidden layer**: can have one or more layers depending on the specific problem.

Hidden layer which extracts some of the most relevant patterns from the inputs and sends them on to the next layer for further analysis. It accelerates and improves the efficiency of the network by recognizing just the most important information from the inputs and discarding the redundant information.



# ACTIVATION FUNCTION

The Activation Function in an artificial neural network (ANN) is an important component that helps the model learn non-linear relationships between input and output. It determines the output of each neuron in the network, and plays an important role in helping the network learn and solve complex problems.



# SIGMOID

**Sigmoid**

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

- The output of the Sigmoid function is in the range (0, 1), making the model suitable for binary classification problems.
- **Advantage**: Minimizes output to a value between 0 and 1, suitable for classification problems.
- **Disadvantage**: Easy to encounter the vanishing gradient problem when the input value is too large or too small.

# TANH

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- The output of the Tanh function is in the range (-1, 1), making it possible for the network to handle both negative and positive values.
- **Advantage**: Better than Sigmoid because the output value has a wider range (-1, 1), which improves learning and reduces vanishing gradients.
- **Disadvantage**: There is also the problem of vanishing gradients when the input value is too large or too small.

# RELU

**ReLU**

$$\max(0, x)$$

- If the input value is negative, the output will be 0, if the input value is positive, the output will be equal to the input value.
- **Advantages**: Effective in reducing the vanishing gradient problem and helps train the network faster. ReLU also reduces computation because it only requires the computation of comparisons (no complex functions like exponential).
- **Disadvantage**: There may be a dying ReLU problem (when all input values are negative, the neuron will never fire, reducing the network's ability to learn).

# LEAKY RELU

**Leaky ReLU**

$$\max(0.1x, x)$$

- **Advantage**: Solves the ReLU dying problem by allowing a small amount of signal to pass when $x$ is negative.
- **Disadvantage**: Need to adjust parameter $\alpha$ to achieve optimal performance.

# THE IMPORTANCE OF ACTIVATION FUNCTION

- **Create nonlinearity**: Without an activation function, a neural network will only learn linear relationships, which limits its ability to solve complex problems. The activation function helps create nonlinearity, allowing the network to learn complex patterns and structures in the data.

- **Determining neuron activation**: The activation function determines whether a neuron is "active" (i.e., whether it transmits a further signal in the network).

# TYPES OF ARTIFICIAL NEURAL NETWORKS

- **Feedforward Neural Networks (FNNs)**
- **Convolutional Neural Networks (CNNs)**
- **Recurrent Neural Networks (RNNs)**
- **Long Short-Term Memory Networks (LSTMs)**
- **Generative Adversarial Networks (GANs)**

# HOW DO ARTIFICIAL NEURAL NETWORKS LEARN?

01 **Starting Point**

02 **Seeing Data**

03 **Guessing and Checking**

04 **Getting Feedback**

05 **Adjusting Strengths**

06 **Practice Makes Perfect**

07 **Testing Skills**

# APPLICATION

- Finance
- Medical
- Computer Vision
- Forecasting
- E-commerce and Marketing
- Logistics

# ANN IN PREDICT PROBLEM

01 Data Preprocessing

02 Characteristic preprocessing

03 Split the data into training and testing sets

04 Build and optimize the ANN model

05 Model training

06 Evaluate the model & Visualizations

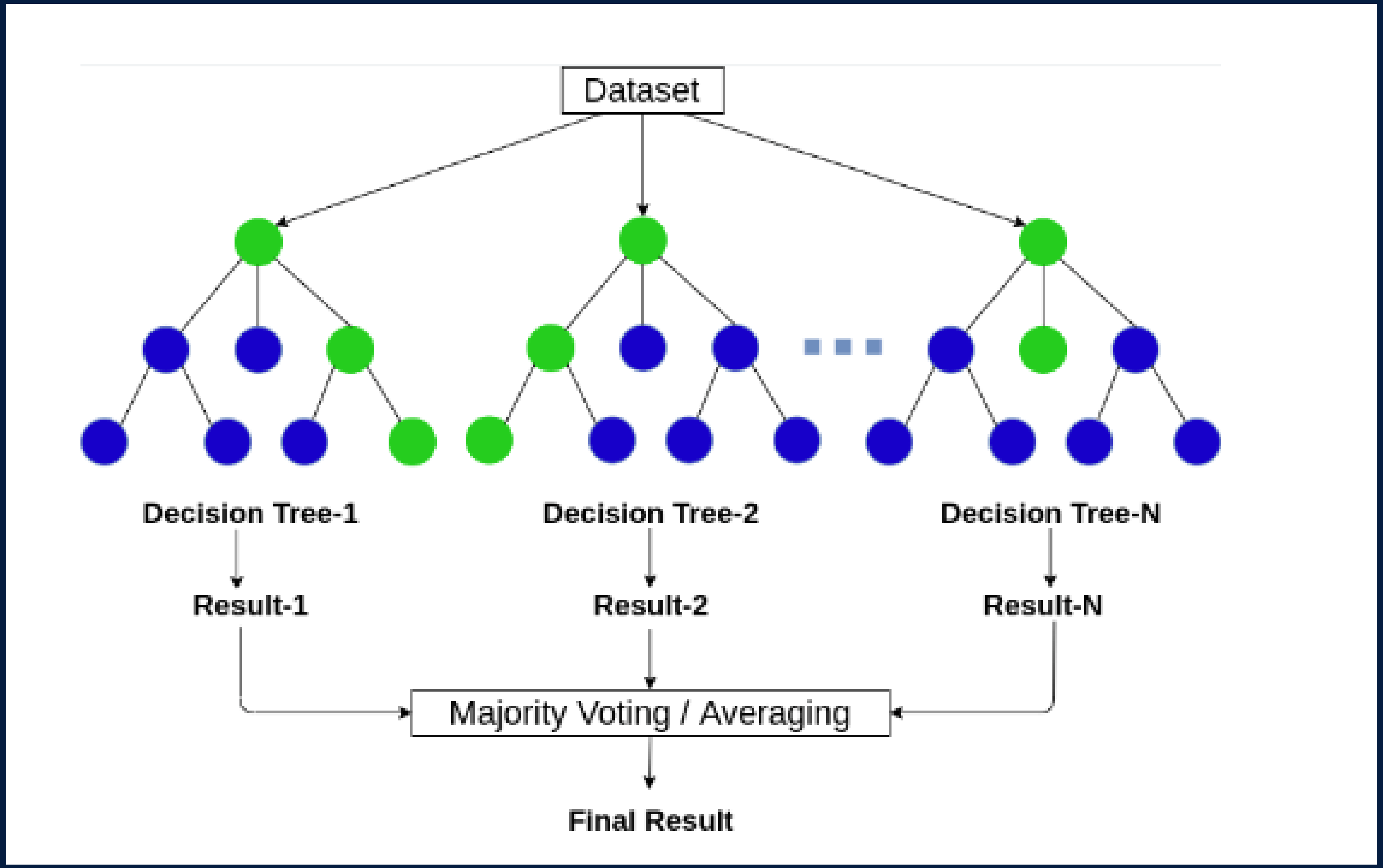
05

# RANDOM FOREST REGRESSION

- What is Random Forest Regression?
- Type of Random Forest Regression?
- How do Random Forest Regression Learn?
- Application
- Random Forest Regression in predict problem

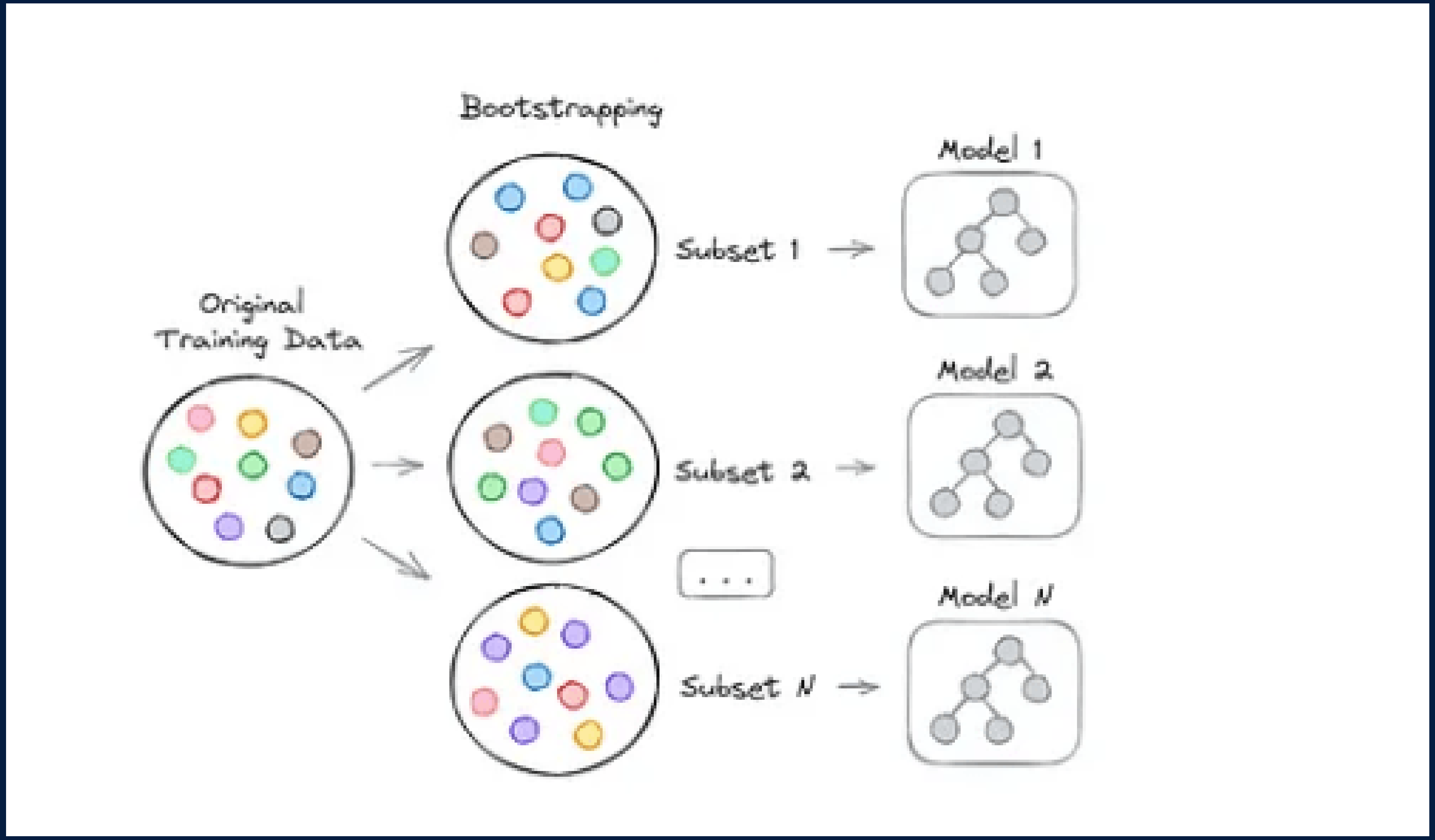
# WHAT IS RANDOM FOREST REGRESSION?

- Random Forest Regression is an ensemble learning algorithm designed for regression tasks. It combines multiple decision trees to make more accurate predictions by averaging the outputs of all trees.



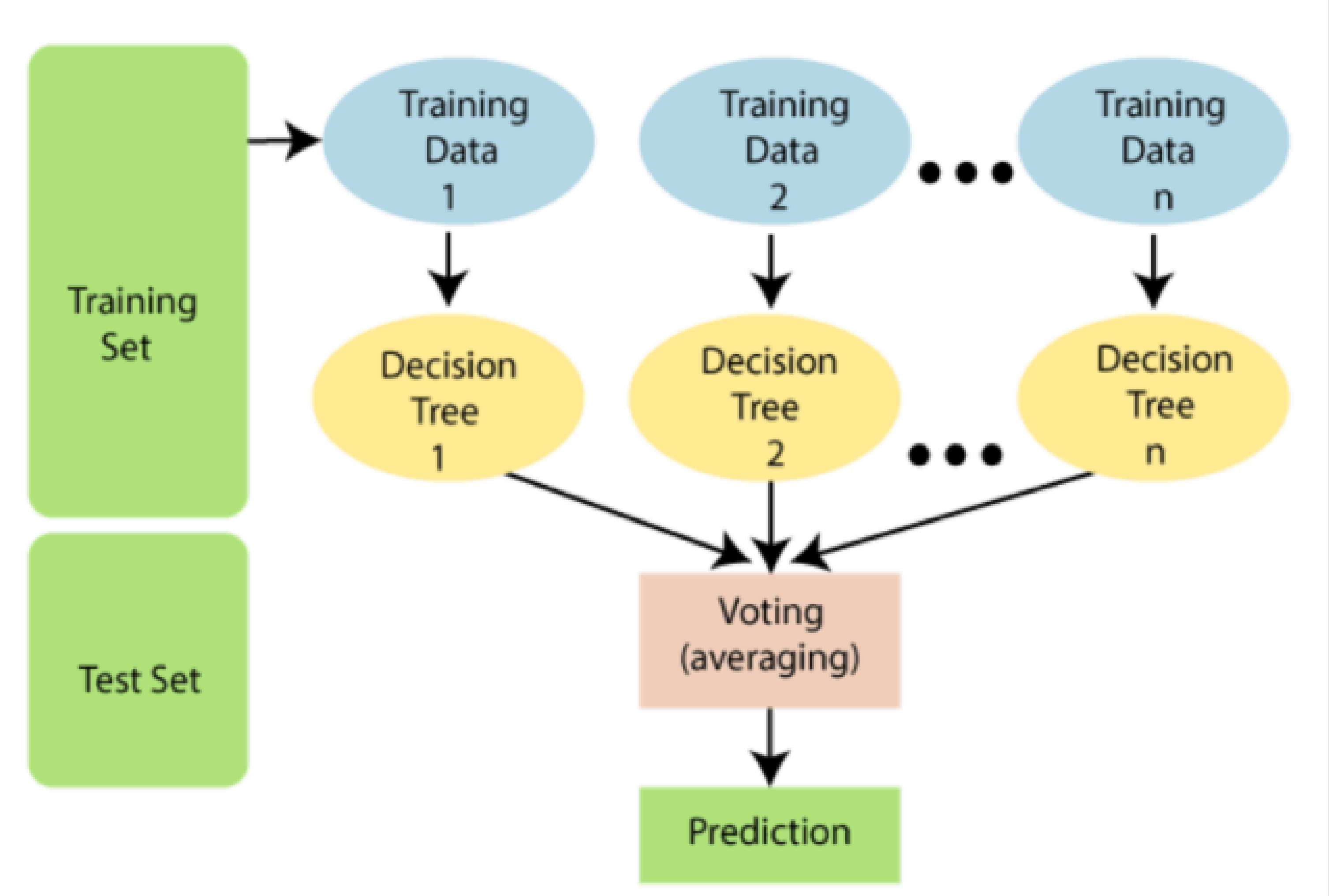
# KEY FEATURES

- Ensemble Learning: Combines multiple models to reduce errors.
- Bagging: Randomly samples data multiple times for training.
- Final Prediction: Averages predictions from all decision trees.



# STRUCTURE

- Bootstrap Sampling: Splits data into random subsets.
- Decision Trees: Builds independent decision trees for each subset.
- Aggregation: The final result is the average of all tree outputs.




Ứng dụng của thuật toán rừng ngẫu nhiên – Random Forests

# TYPES OF RANDOM FOREST REGRESSION

1. By Prediction Method:
   - Standard Random Forest: Predicts the mean value.
   - Quantile Random Forest: Predicts value ranges (quantiles).
2. By Application:
   - Time Series Analysis: Predict trends, seasonality.
   - High-Dimensional Data: Handles datasets with many features.

# HOW DO RANDOM FOREST REGRESSION LEARN?

Steps in the Learning Process:

1. Bootstrap Sampling:
   - Randomly selects subsets from the original dataset.
   - Some data points may appear multiple times in subsets.
2. Build Decision Trees:
   - Constructs independent trees for each subset.
   - Randomly selects a subset of features for splitting at each node.
3. Combine Results:
   - Averages the predictions from all trees to produce the final result.

Advantages:

- Reduces overfitting through result averaging.
- Improves accuracy due to diverse trees.

# APPLICATIONS

1. Finance: Predicting stock prices and investment returns.
2. Healthcare: Estimating treatment costs, predicting disease outcomes.
3. Real Estate: Forecasting property prices based on factors like area, location.
4. E-commerce: Revenue prediction, product pricing optimization.

# RANDOM FOREST REGRESSION IN PREDICT PROBLEMS

01 Data Preprocessing

02 Characteristic preprocessing

03 Split the data into training and testing sets

04 Build and optimize the Random Forest Regression model

05 Evaluate the model

06 Deploy the Model

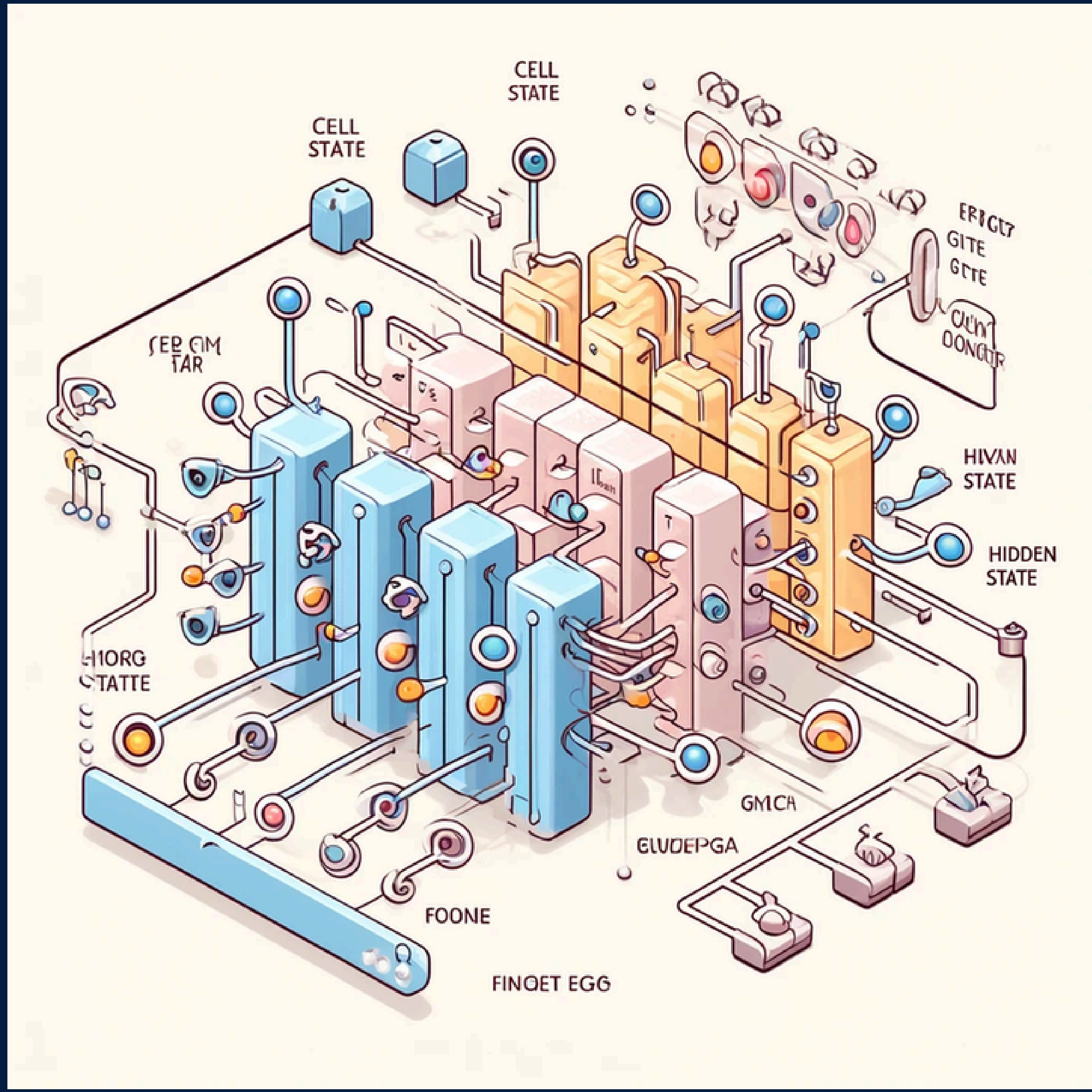
06

# LSTM-LONG SHORT TERM MEMORY

- WHAT IS LSTM ?
- TYPES OF LSTM
- HOW DO LSTM LEARN?
- APPLICATION
- LSTM IN PREDICT PROBLEM

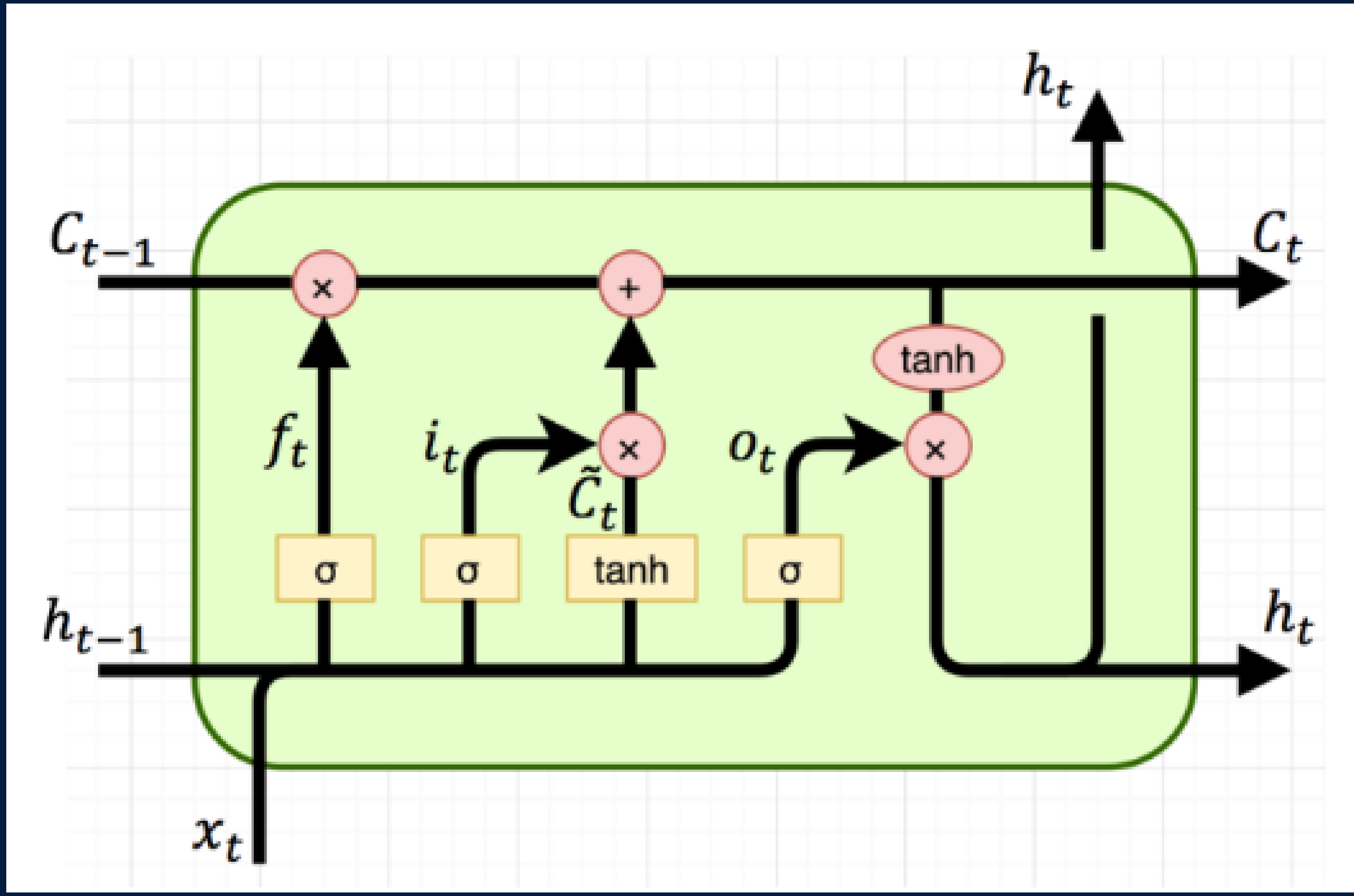
# WHAT IS LSTM?

- LSTM is a type of Recurrent Neural Network (RNN) designed to effectively learn long-term dependencies in sequential data. It is widely used in time series forecasting, natural language processing (NLP), and speech recognition
- LSTM introduces memory cells and gates to retain or forget information over long sequences, addressing the vanishing gradient problem of traditional RNNs.



# STRUCTURE

- **Cell State (Ct)**: The cell state acts as a memory that carries relevant information through the sequence. It allows information to flow unchanged across the cell, providing a direct path for gradients during backpropagation.
- **Hidden State (ht)**: The hidden state is the output of the LSTM cell at a given time step, contributing to the final output and being passed to the next cell in the sequence.
- **Gates**: LSTMs use three types of gates to regulate information flow:
  - **Forget Gate (ft)**: Decides what portion of the cell state to discard.
  - **Input Gate (it)**: Determines which new information to add to the cell state.
  - **Output Gate (ot)**: Controls the output and the updated hidden state.



# TYPES OF LSTM

- Vanilla LSTM: Standard LSTM with a single layer of LSTM units.
- Stacked LSTM: Multiple LSTM layers stacked to model more complex data patterns.
- Bidirectional LSTM: Processes input sequences in both forward and backward directions, capturing contextual information from both sides.
- Multivariate LSTM: Designed to handle datasets with multiple input variables.
- ConvLSTM (Convolutional LSTM): Combines LSTMs with convolutional layers for spatiotemporal data, such as video analysis.

# HOW DO LSTM LEARN?

1. Data Preparation:
    - Preprocess sequential data, such as normalizing values or encoding categorical data.
    - Convert data into sequences (e.g., using a sliding window for time series).
2. Memory Cell Mechanism:

- At each time step t, the LSTM updates its memory using:
    - Forget Gate: $ft = \sigma\left(W_f.\left[h_{t-1}, x_t\right] + b_f\right)$
    - Input Gate: $i_t = \sigma\left(W_i.\left[h_{t-1}, x_t\right] + b_i\right)$

      $\widetilde{C}_t = tanh(W_C \cdot \left[h_{t-1}, x_t\right] + b_C)$
    - Output Gate: $o_t = \sigma(W_o \cdot \left[h_{t-1}, x_t\right] + b_o)$

1. Forward Propagation:
    - Compute outputs at each time step by passing information through memory cells.
2. Backpropagation Through Time (BPTT):
    - Adjust weights by propagating errors backward through the time steps.

# APPLICATIONS

1. Time Series Forecasting: Predicting stock prices, energy consumption, or weather patterns.
2. Natural Language Processing (NLP): Sentiment analysis, language translation, or text generation.
3. Speech Recognition: Converting spoken language into text.
4. Anomaly Detection: Identifying irregularities in sensor data or system logs.
5. Video Analysis: Activity recognition or video captioning.

# LSTM IN PREDICT PROBLEMS

01 Data Preprocessing

02 Characteristic preprocessing

03 Split the data into training and testing sets

04 Build and optimize the LSTM model

05 Evaluate the model

06 Deploy the Model

# 07 SVM - SUPPORT VECTOR MACHINE

What is a SVM?

Types of SVM

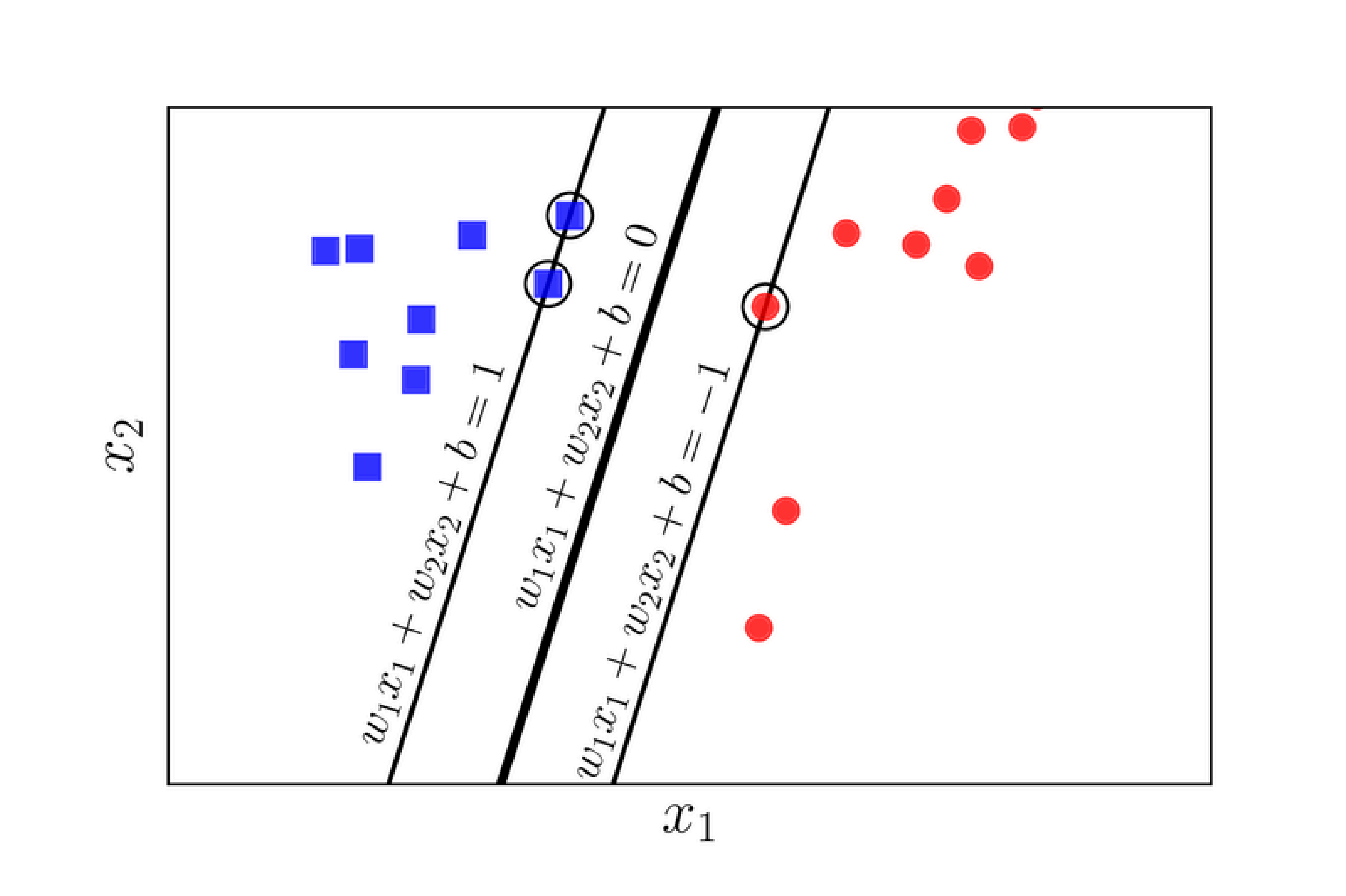
How do SVM work?

Application

SVM in predict problem

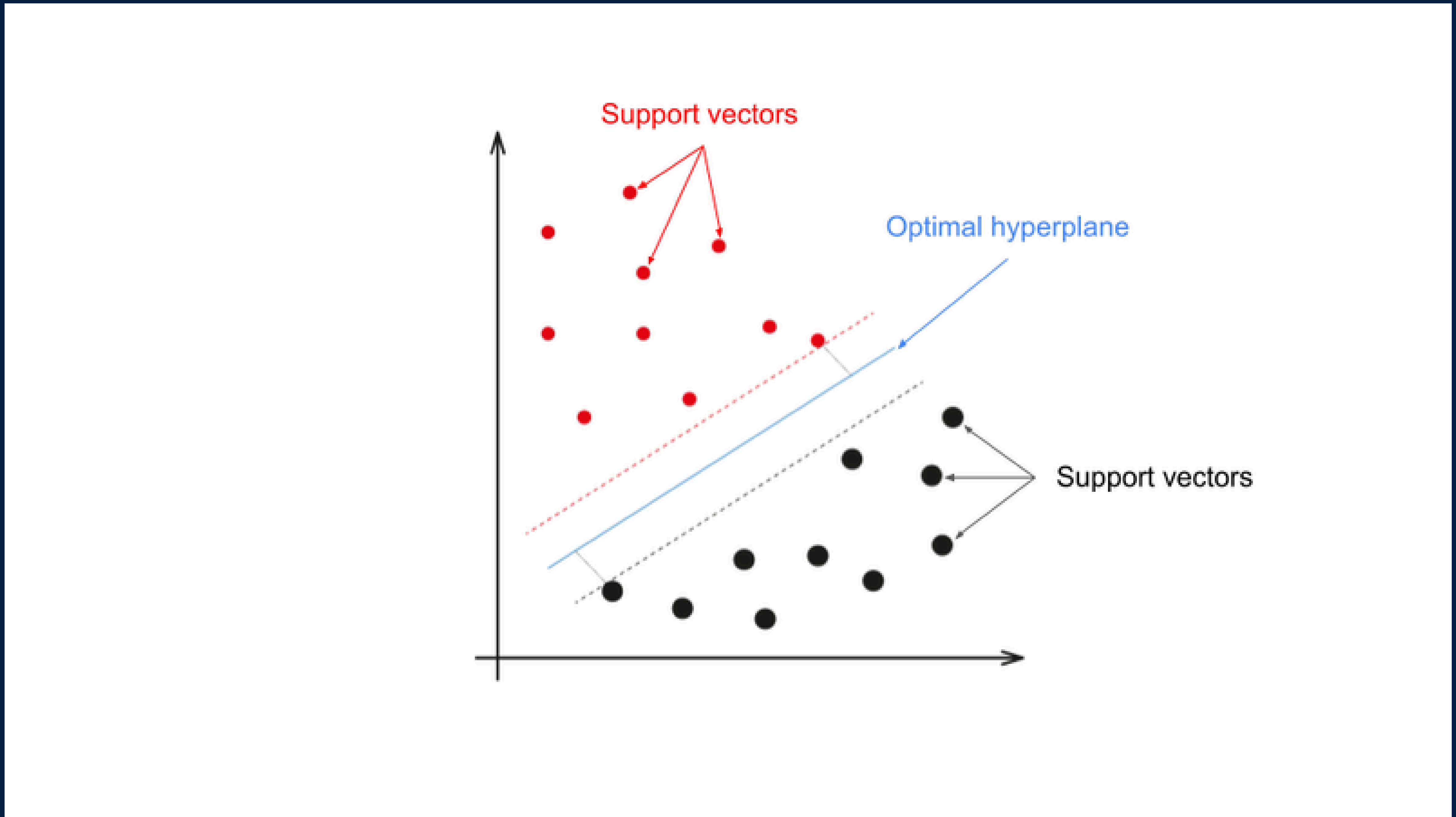
# WHAT IS SVM?

- SVM is a supervised learning algorithm used for classification and regression tasks. It finds the hyperplane that best separates data points from different classes in a high-dimensional space.
- SVM focuses on finding the maximum margin between the hyperplane and the nearest data points (support vectors), ensuring the model is robust to new data.



# STRUCTURE

- Hyperplane: Decision boundary separating different classes.
- Support Vectors: Closest data points to the hyperplane, crucial for defining it.
- Kernel Trick: Maps input data into higher dimensions to handle non-linear relationships.



# TYPES OF SVM

1. Linear SVM:Used when data is linearly separable.
2. Non-Linear SVM:
   - Handles non-linear data by using kernel functions.
   - Common kernels include: Polynomial Kernel, Radial Basis Function (RBF) Kernel.
3. Soft-Margin SVM: Allows misclassification of some data points to avoid overfitting.
4. Hard-Margin SVM: Requires all data points to be correctly classified, used for perfectly separable data.
5. SVM for Regression (Support Vector Regression - SVR): Predicts continuous outputs by finding a margin of tolerance around the true data points.

# HOW DO LSTM LEARN?

1. Data Preparation:
    - Normalize or standardize features to ensure all variables contribute equally.
    - Split data into training and testing sets.
2. Identify Optimal Hyperplane:
    - SVM searches for the hyperplane w.x+b=0 that maximizes the margin:

      $Maximize \frac{2}{\|w\|}$ ,subject to constraints: $y_i(w.x_i + b) \geq 1$
3. Kernel Trick for Non-Linear Data:
    - Maps input data into a higher-dimensional space: $K(x_i, x_j) = \phi(x_i).\phi(x_j)$

# HOW DO LSTM LEARN?

4. Optimization Problem:
   - Solved using quadratic programming to minimize : $\frac{1}{2}\|w\|^2 + C\sum \xi_i$
   - C: Regularization parameter controlling trade-off between maximizing the margin and minimizing misclassification.
   - $\xi_i$: Slack variables for soft margin.

5. Model Training:
   - Learn w and b based on training data.

6. Make Predictions:
   - Maps input data into a higher-dimensional space:

$$y = sign(w \cdot x + b)$$

   - For regression (SVR): Predict continuous values within a defined margin of tolerance.

# APPLICATIONS

1. Classification Tasks:
    - Image recognition (e.g., handwritten digit recognition).
    - Email spam detection.
2. Regression Tasks (SVR):
    - Forecasting stock prices or house prices.
3. Anomaly Detection:
    - Detecting fraud or irregular patterns in datasets.
4. Text Mining:
    - Sentiment analysis, topic categorization.
5. Bioinformatics:
    - Classification of genes or proteins.

# SVM IN PREDICT PROBLEMS

01 Data Preprocessing

02 Characteristic preprocessing

03 Split the data into training and testing sets

04 Build and optimize the SVM model

05 Evaluate the model

06 Deploy the Model

# THANK YOU

CONNECT WITH US.