ACC-HPC June 2025

ENSEMBLE LEARNING & CLASSIFICATION

Machine Learning

Dr Kiran Waghmare
CDAC Mumbai

SLR Program Implementation

What is Linear Regression (LR)?

- Linear regression (LR) models the linear relationship between the one independent (x) variable with that of the dependent variable (y). If there are multiple independent variables in a model, it is called as multiple linear regression.
- For example, how the likelihood of blood pressure is influenced by a person's age and weight. This relationship can be explained using linear regression.
- In LR, the y variable should be continuous, whereas the x variable can be continuous or categorical. If both x and y are continuous, the linear relationship can be estimated using correlation coefficient (r) or the coefficient of determination (R-Squared)

- LR is useful if the relationships between the x and y variables are linear
- LR is helpful to predict the value of y based on the value of the x variable

Note: Dependent variable also called a response, outcome, regressand, criterion, or endogenous variable.

Independent variable also called explanatory, covariates, predictor, regressor, exogenous, manipulated, or feature (mostly in machine learning) variable.

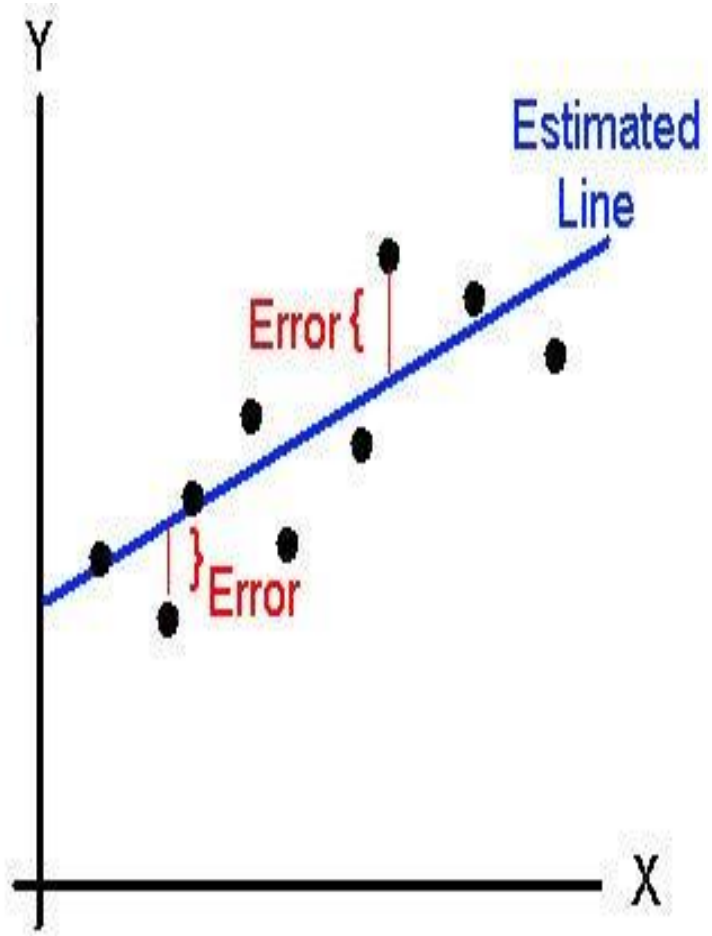
Estimated (or predicted) Y value for observation i

Estimate of the regression intercept

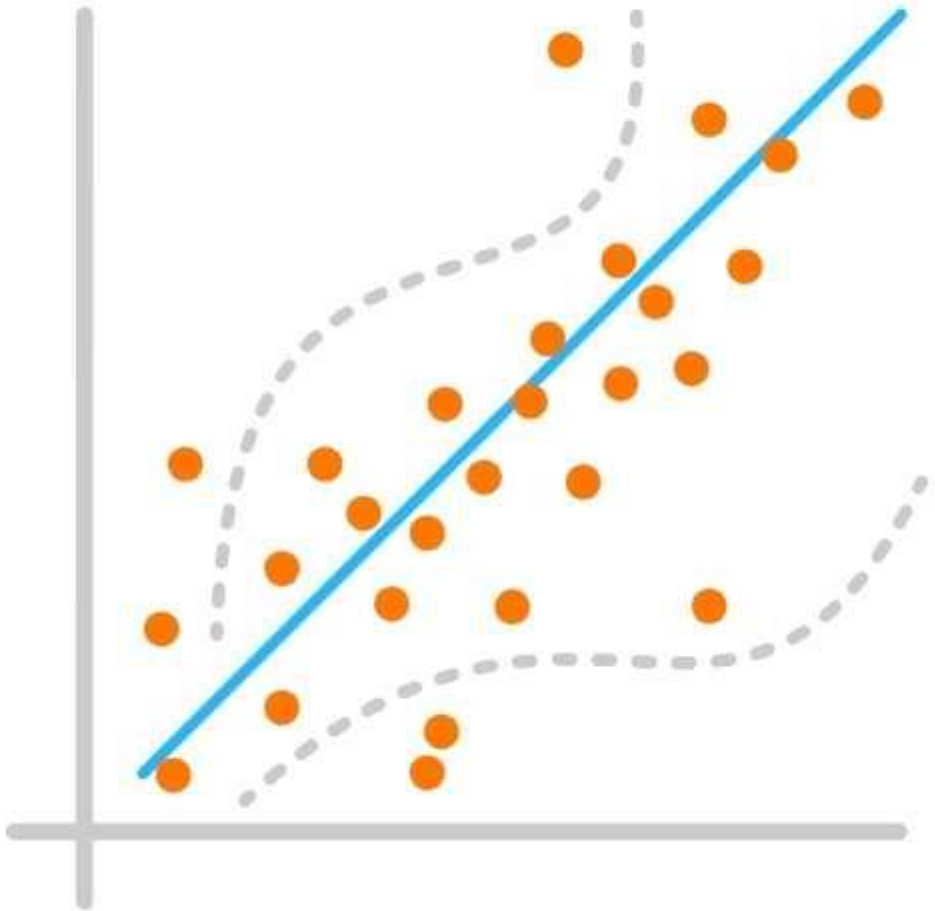
Estimate of the regression slope

Value of X for observation i

$$\hat{Y}_i = b_0 + b_1 X_i$$



4 Common Regression Metrics



1. Mean Squared Error (MSE)
2. Root Mean Squared Error (RMSE)
3. Mean Absolute Error (MAE)
4. R-squared (R^2)

Linear model

We are not interested in the intercept a but only in the coefficient b.

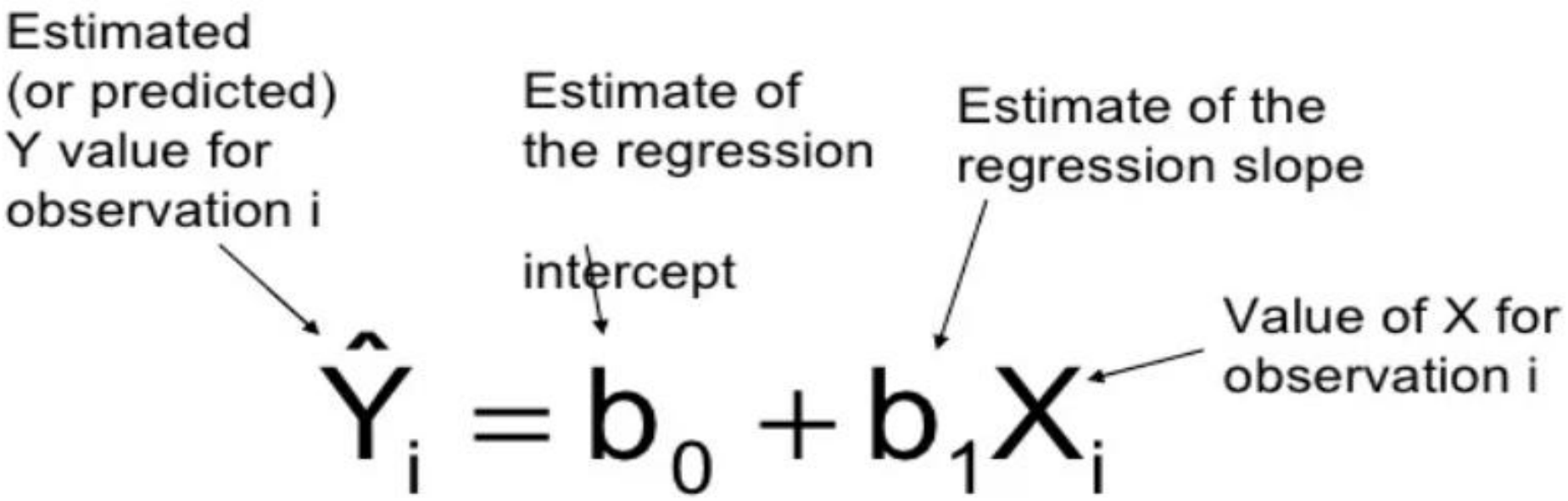
The coefficient b represents the relationship between X and Y.

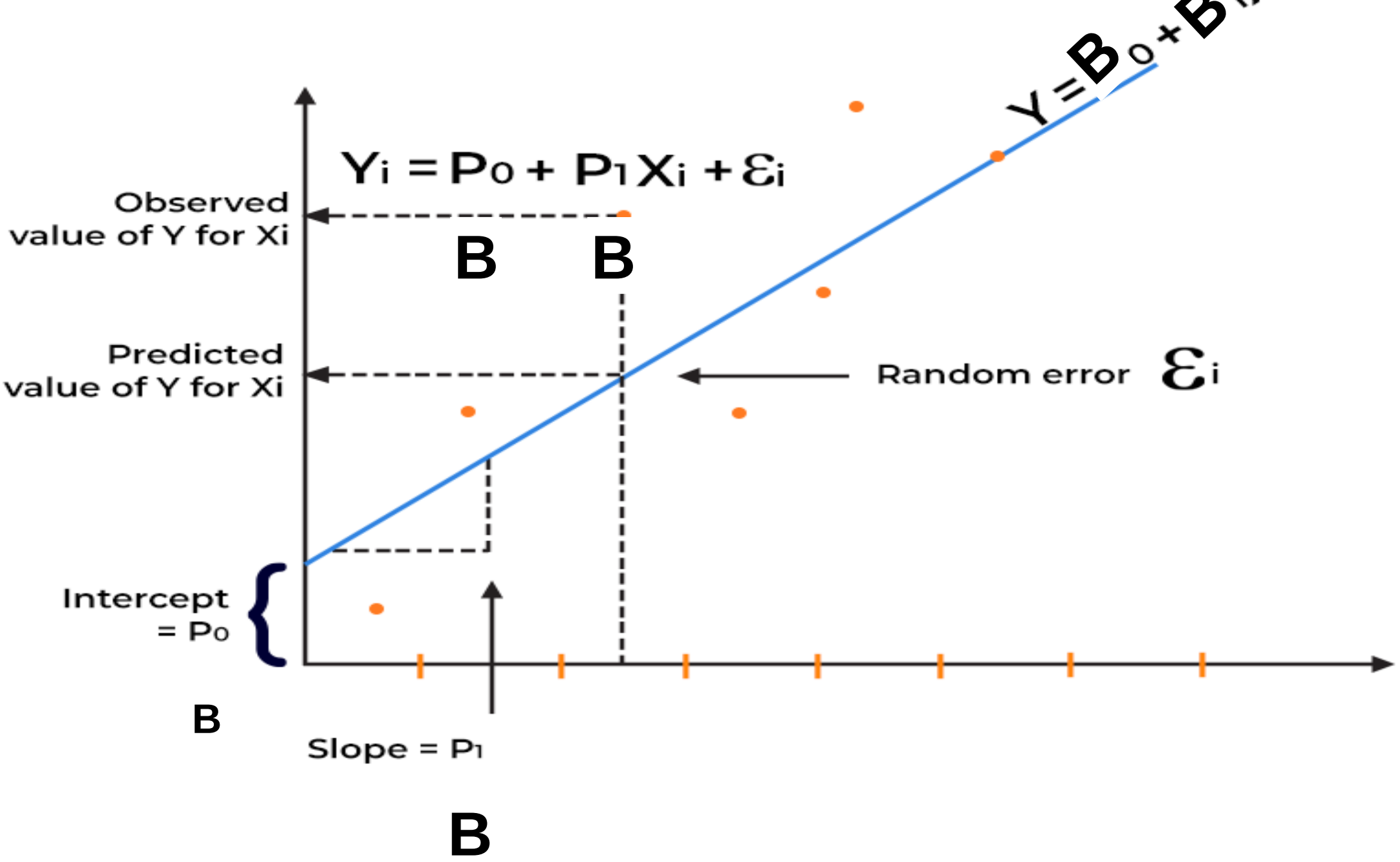
- If **b is positive,** X has a positive effect on Y (as X increases, Y increases);
- If **b is negative**, X has a negative effect on Y (as X increases, Y decreases).

If **b = 0,** there is **no effect** of X on Y.

Simple Linear Regression Equation (Prediction Line)

The simple linear regression equation provides an estimate of the population regression line



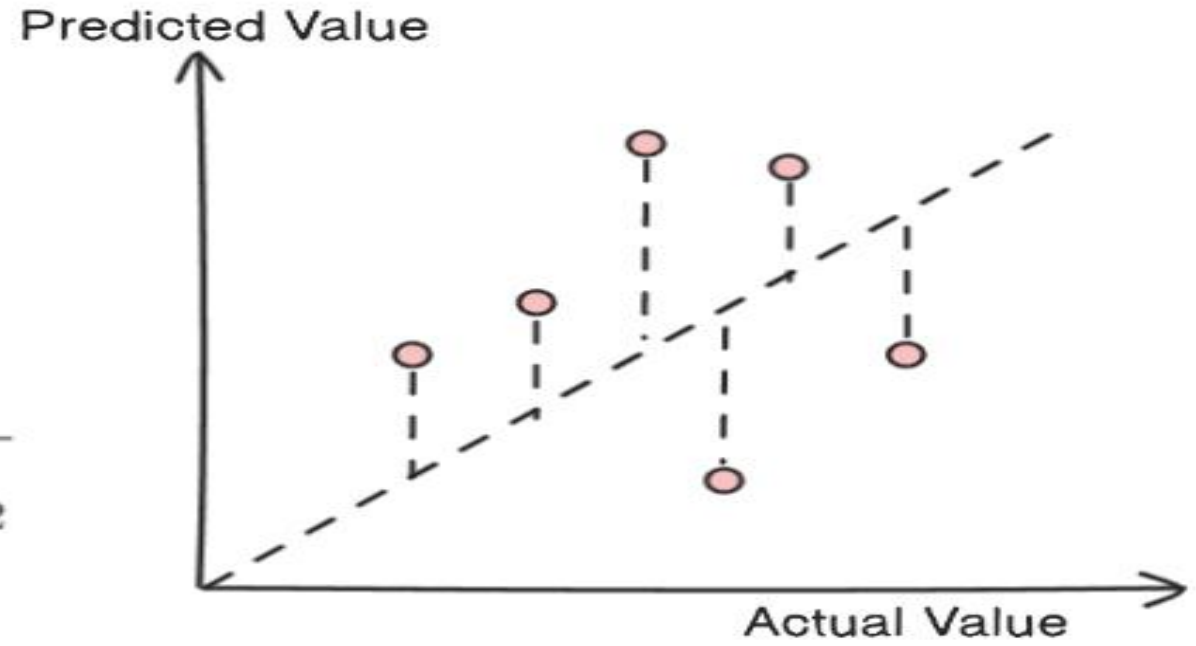

Y = B0 + B
Yi = P0 + P1 Xi + εi
Observed value of Y for Xi
B
B
Predicted value of Y for Xi
Random error εi
Intercept = P0
B
Slope = P1
B

Regression Metrics

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

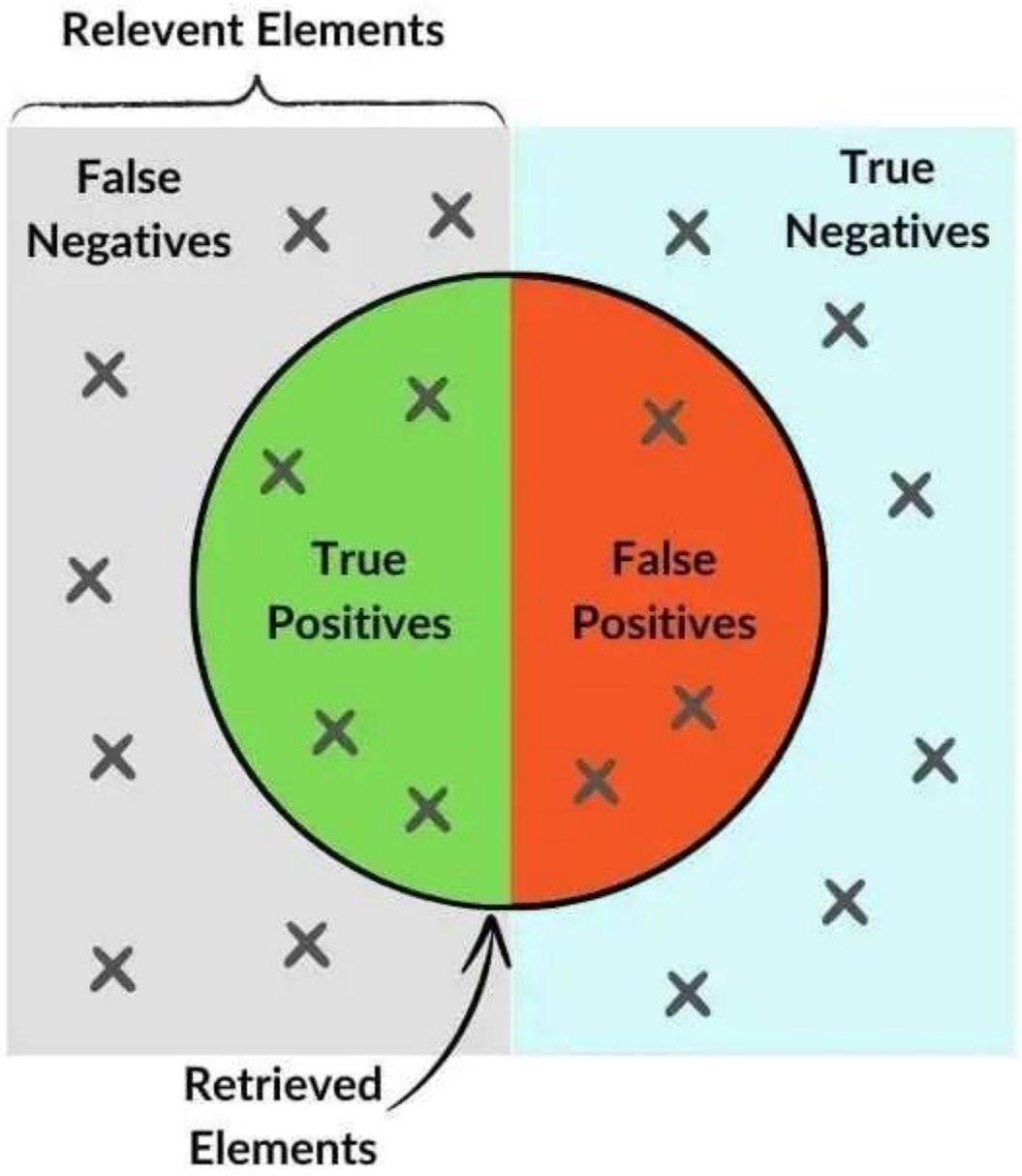
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$



$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

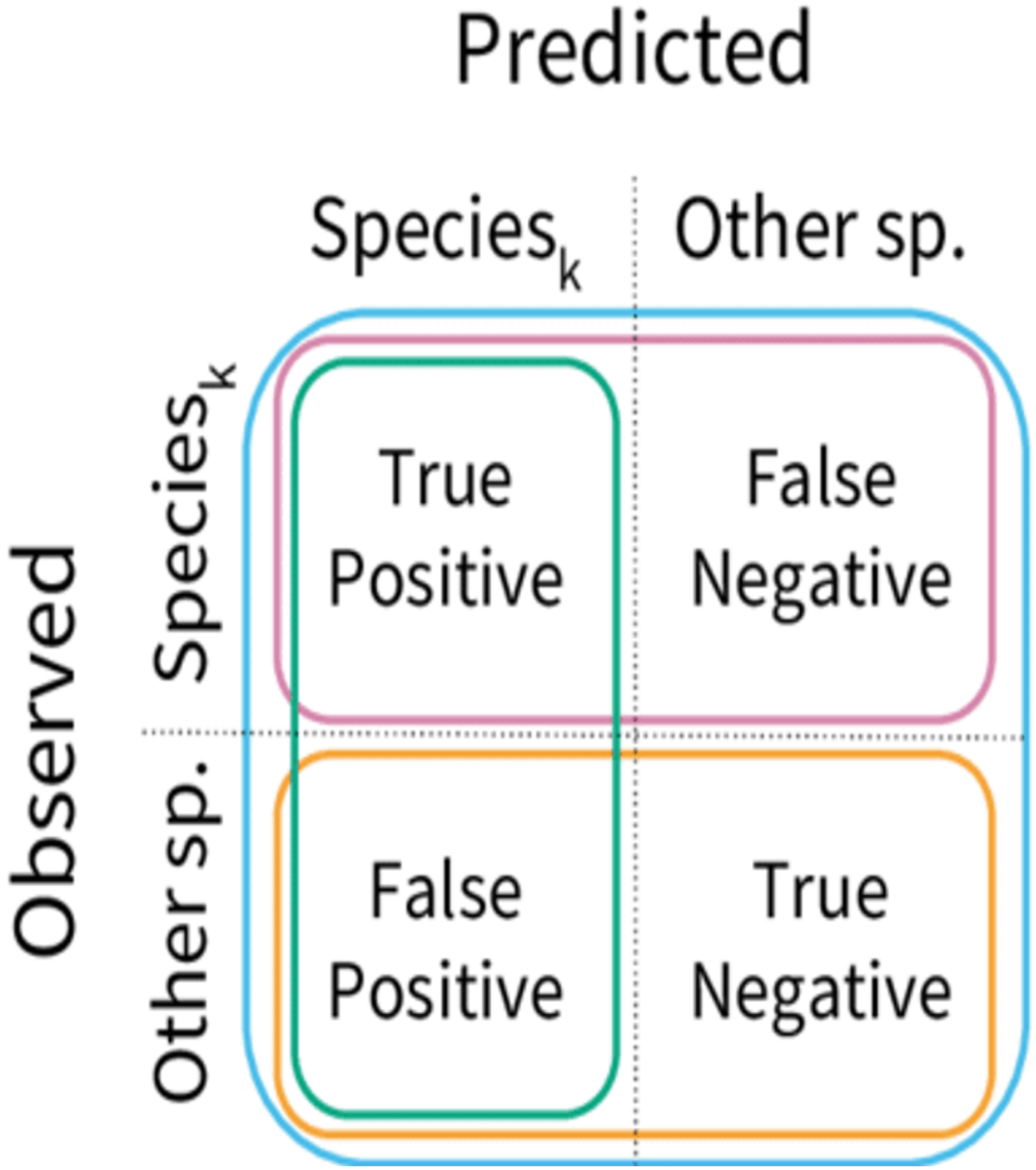


How many retrieved elements are relevant?

Precision = $\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$

How many relevant elements are retrieved?

Recall = $\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$



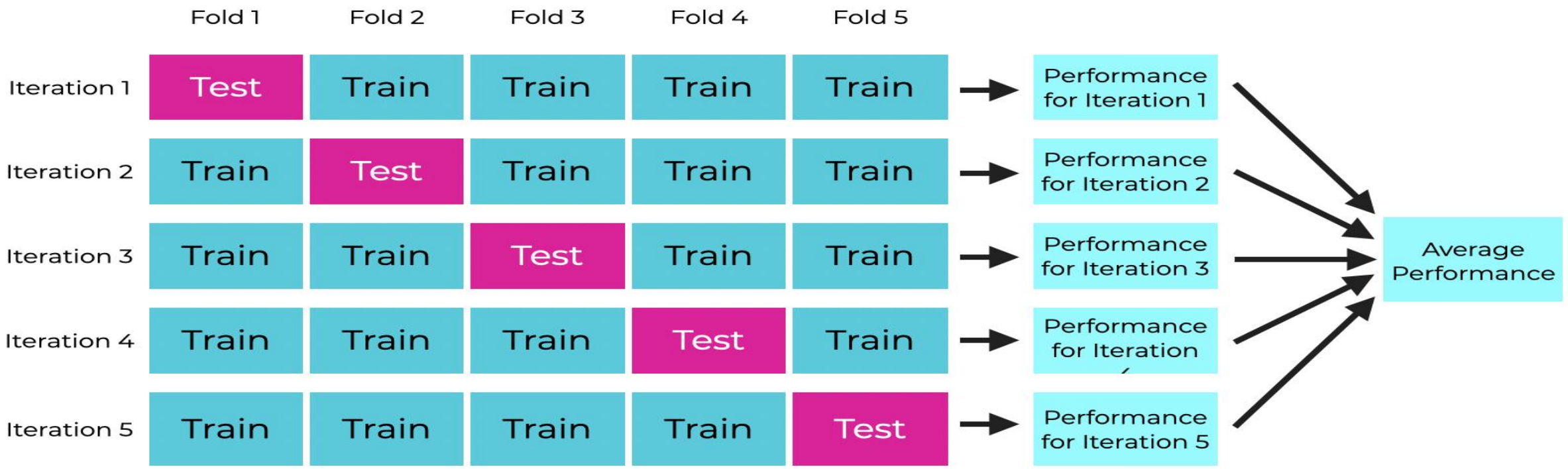
Accuracy $= \frac{TP + TN}{TP + TN + FP + FN}$

Specificity $= \frac{TN}{TN + FP}$

Precision $= \frac{TP}{TP + FP}$

Recall $= \frac{TP}{TP + FN}$

CROSS VALIDATION, EXPLAINED



Fitting curves and regression analysis

- **Model Fit Quality:**
  - A model fits the data well when differences between observed and predicted values are small and unbiased.
- **Scatterplots & Line of Best Fit:**
  - Trend lines in scatterplots help visualize the direction of data.
  - The line of best fit should:
    - Have an equal number of data points on each side (median position).
    - Not simply connect the first and last data points (to avoid bias from outliers).

Iris dataset

- **Many exploratory data techniques are nicely illustrated with the iris dataset.**
  - Dataset created by famous statistician Ronald Fisher
  - 150 samples of three species in genus *Iris* (50 each)
    - *Iris setosa*
    - *Iris versicolor*
    - *Iris virginica*
  - Four attributes
    - sepal width
    - sepal length
    - petal width
    - petal length
  - Species is class label

Iris virginica. Robert H. Mohlenbrock. USDA NRCS. 1995. Northeast wetland flora: Field office guide to plant species. Northeast National Technical Center, Chester, PA. Courtesy of USDA NRCS Wetland Science Institute.

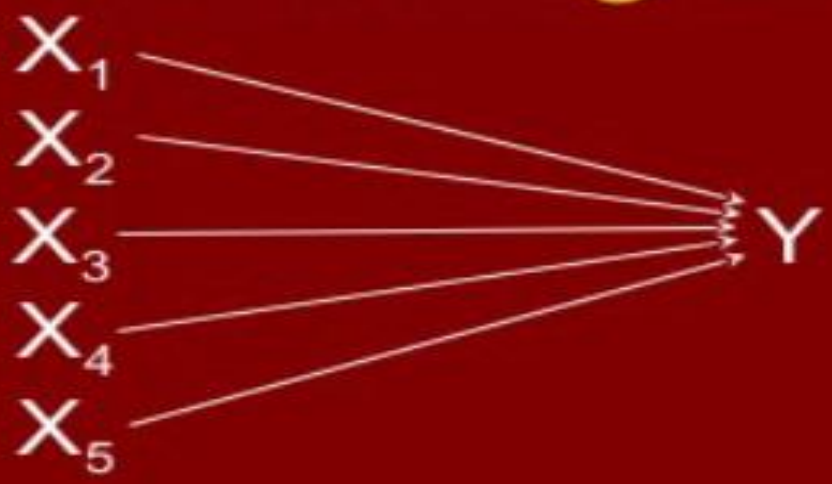
What is multiple linear regression (MLR)?
Visual model

Linear Regression

Single predictor

Multiple Linear Regression

Multiple predictors



Simple Linear Regression

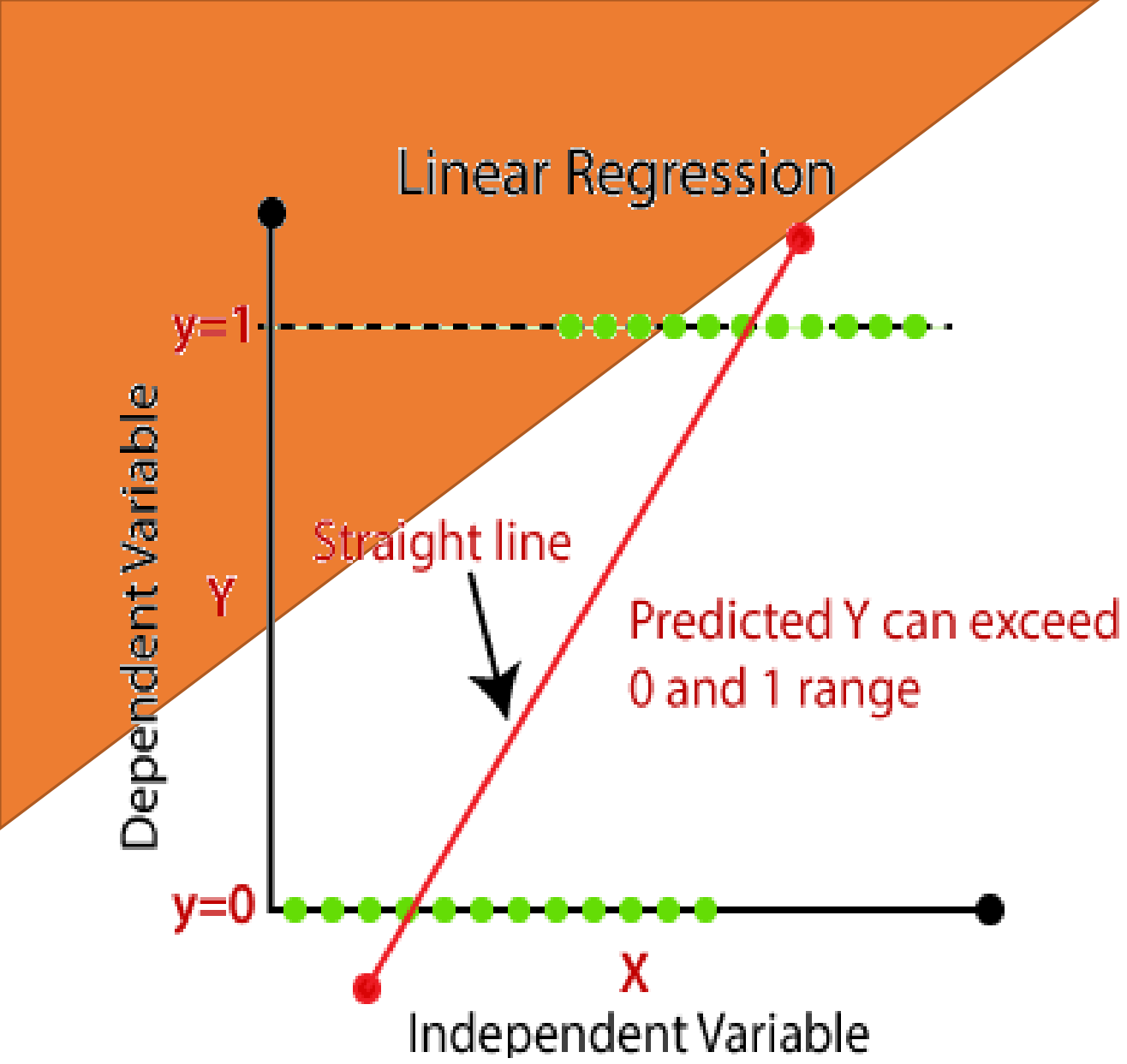
$$y = b_0 + b_1 x_1$$

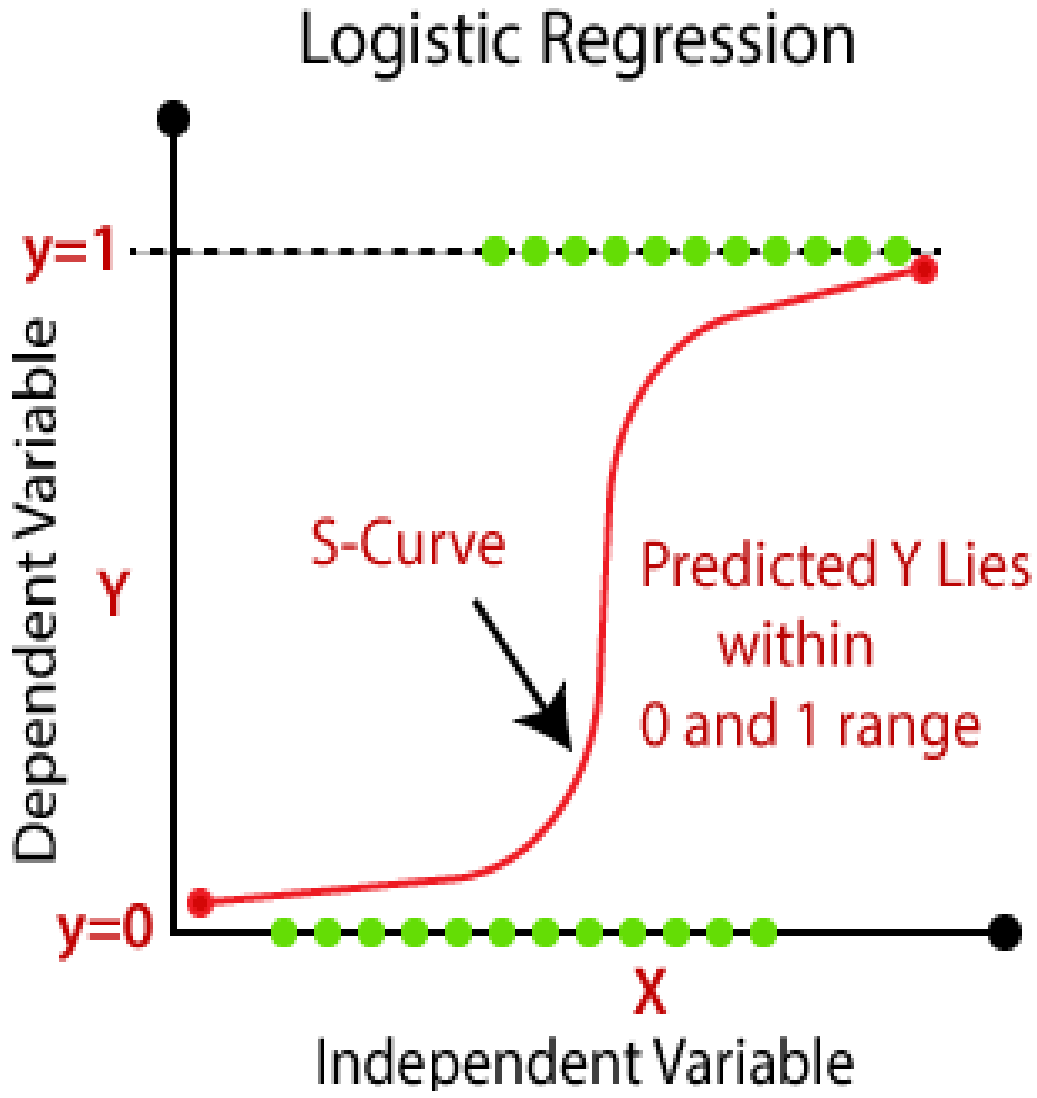
Multiple Linear Regression

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

Polynomial Linear Regression

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$


Linear Regression
y=1
Dependent Variable
Y
Straight line
Predicted Y can exceed
0 and 1 range
y=0
X
Independent Variable


Logistic Regression
y=1
Dependent Variable
Y
S-Curve
Predicted Y Lies
within
0 and 1 range
y=0
X
Independent Variable

Logistic Regression:

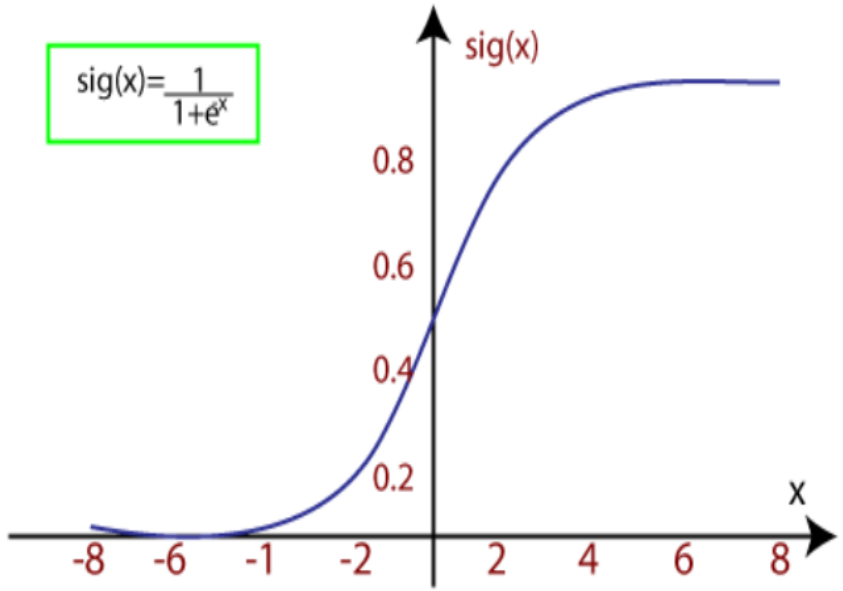
- Logistic regression is another supervised learning algorithm which is used to solve the classification problems. In classification problems, we have dependent variables in a binary or discrete format such as 0 or 1.
- Logistic regression algorithm works with the categorical variable such as 0 or 1, Yes or No, True or False, Spam or not spam, etc.
- It is a predictive analysis algorithm which works on the concept of probability.
- Logistic regression is a type of regression, but it is different from the linear regression algorithm in the term how they are used.
- Logistic regression uses sigmoid function or logistic function which is a complex cost function. This sigmoid function is used to model the data in logistic regression. The function can be represented as:

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

- f(x)= Output between the 0 and 1 value.
- x= input to the function
- e= base of natural logarithm.

When we provide the input values (data) to the function, it gives the S-curve as follows

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



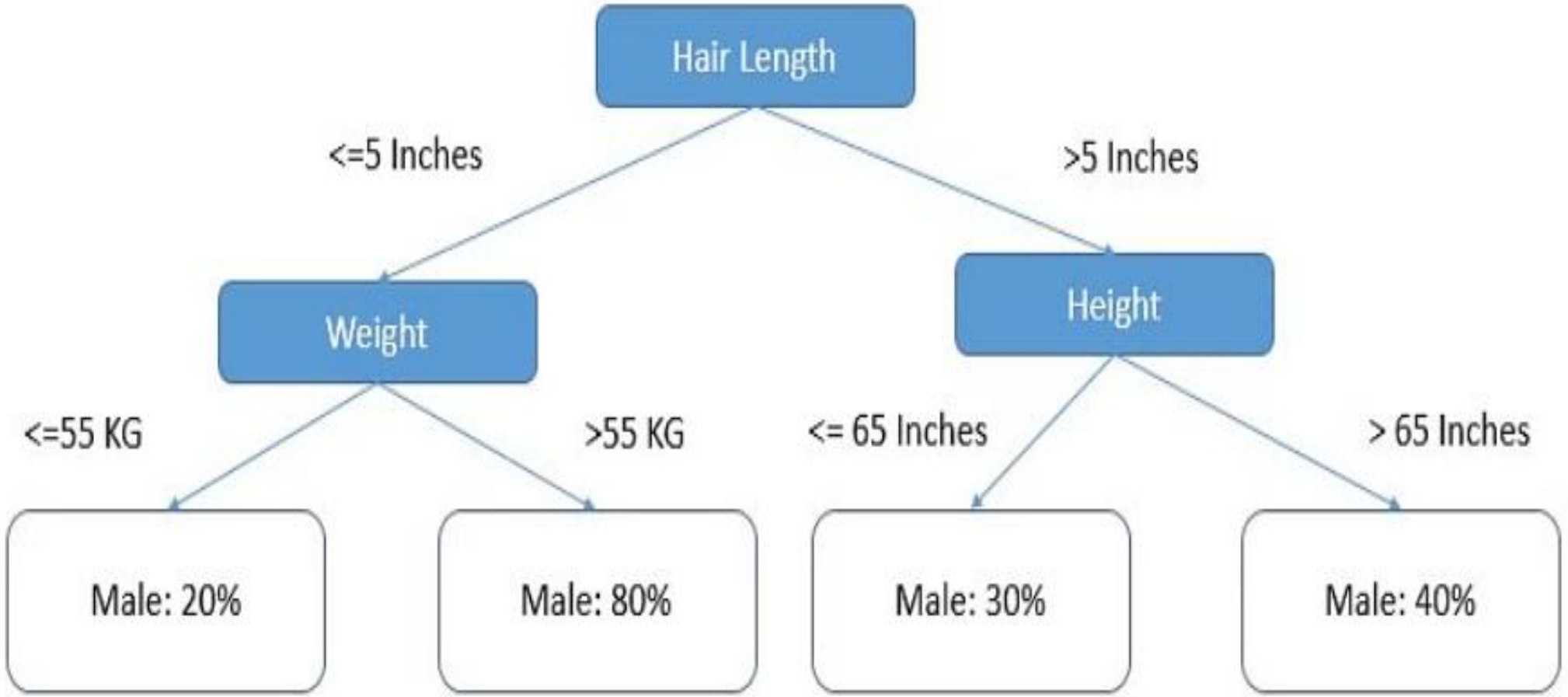
- It uses the concept of threshold levels, values above the threshold level are rounded up to 1, and values below the threshold level are rounded up to 0.

There are three types of logistic regression:

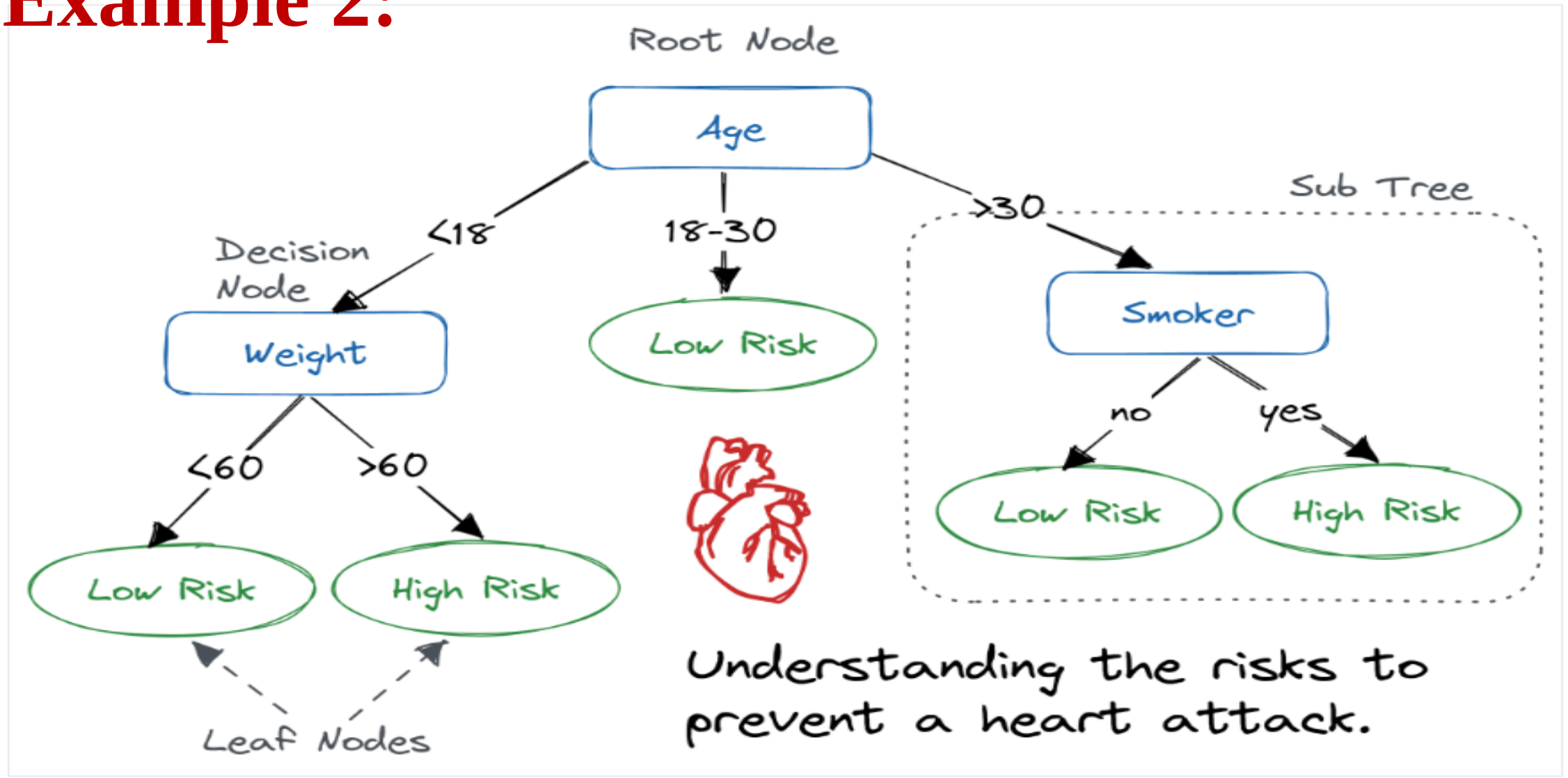
- **Binary(0/1, pass/fail)**
- **Multi(cats, dogs, lions)**
- **Ordinal(low, medium, high)**

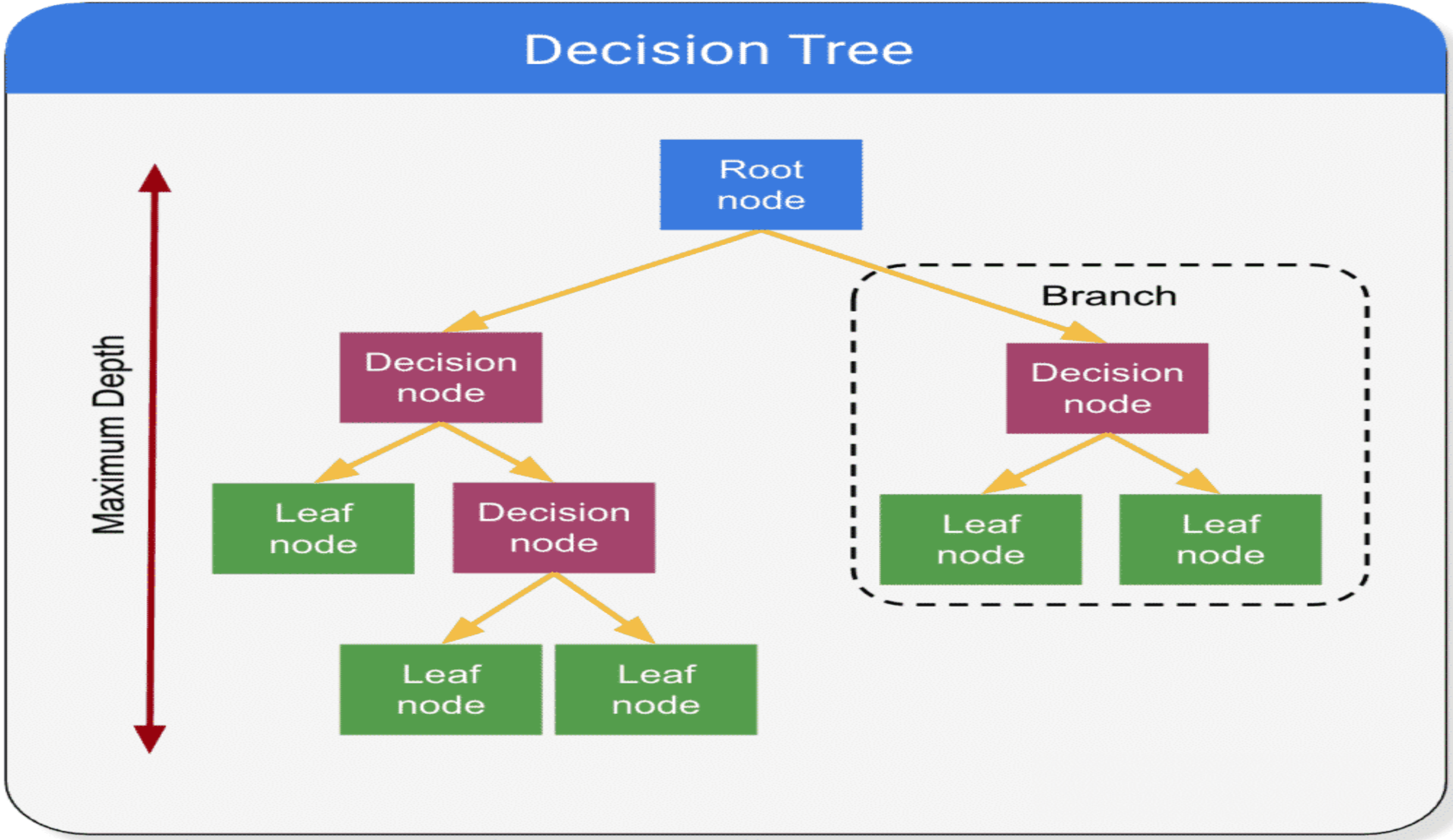
Case Study : Titanic dataset

Example 1:


Hair Length
<=5 Inches
>5 Inches
Weight
Height
<=55 KG
>55 KG
<= 65 Inches
> 65 Inches
Male: 20%
Male: 80%
Male: 30%
Male: 40%

Example 2:


Root Node
Age
<18
18-30
>30
Decision Node
Weight
Low Risk
Sub Tree
Smoker
<60
>60
no
yes
Low Risk
High Risk
Low Risk
High Risk
Leaf Nodes
Understanding the risks to prevent a heart attack.


Decision Tree
Root node
Decision node
Leaf node
Decision node
Leaf node
Leaf node
Branch
Decision node
Leaf node
Leaf node
Maximum Depth

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.

Decision Tree

- A decision tree is one of the **most powerful tools** of supervised learning algorithms used for **both classification and regression tasks.**

- It builds a **flowchart-like tree structure** where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

- It is **constructed by recursively splitting the training data into subsets** based on the values of the attributes until a stopping criterion is met, such as the maximum depth of the tree or the minimum number of samples required to split a node.

Attribute Selection Measures

- While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM.
- By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:
  - Information Gain
  - Gini Index

$$Entropy\,(P) = -\sum_{i=1}^{n} p_i \log_2 (p_i)$$

Information Gain and Gini Index in Decision Tree

$$Gini\,(P) = 1 - \sum_{i=1}^{n} (p_i)^2$$

1. Information Gain:

- **Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.**

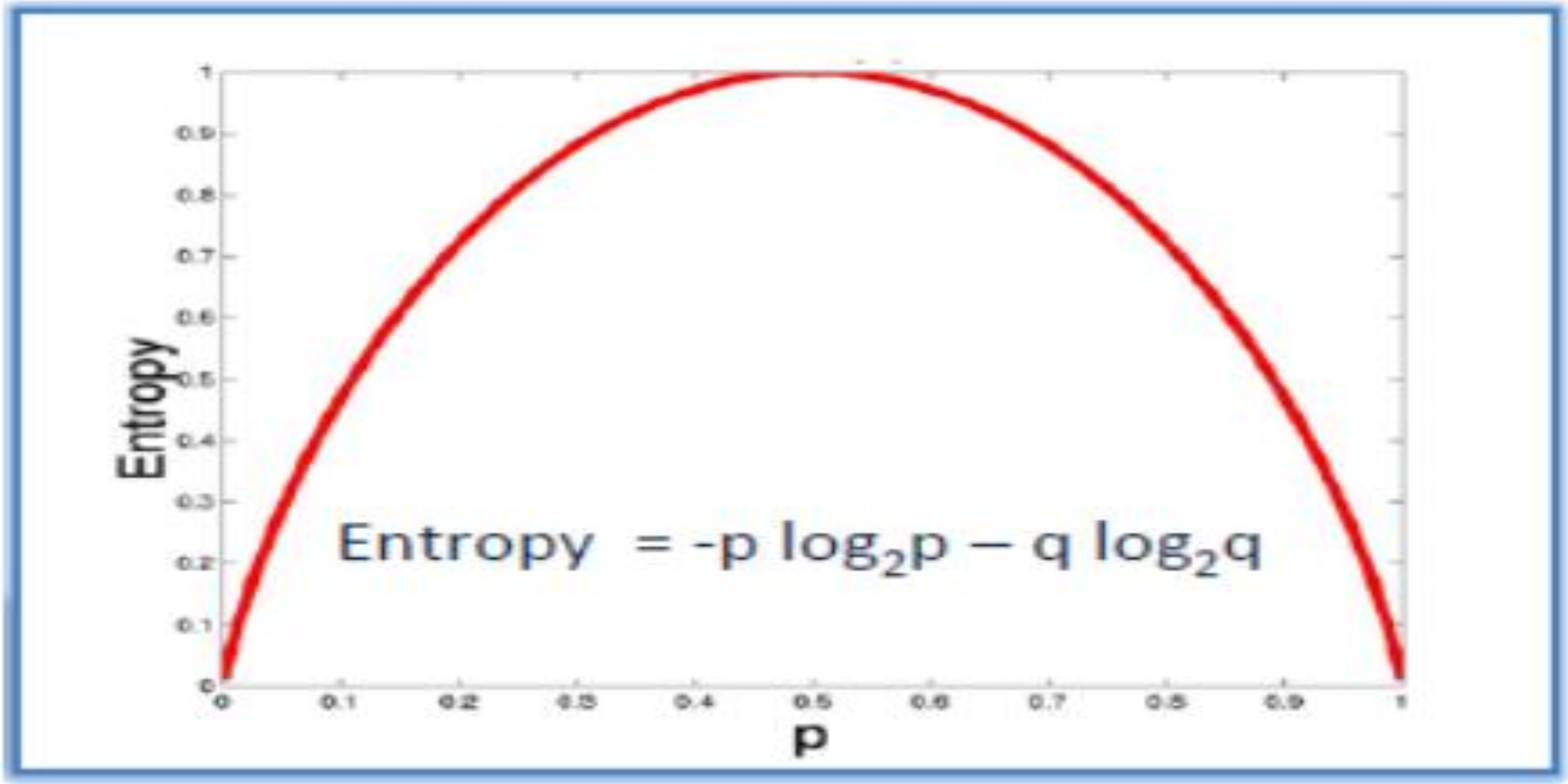
- **It calculates how much information a feature provides us about a class.**

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

$$\text{Entropy}(s) = -P(\text{yes})\log_2 P(\text{yes}) - P(\text{no})\log_2 P(\text{no})$$

Where,

- **S= Total number of samples**
- **P(yes)= probability of yes**
- **P(no)= probability of no**



Entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A

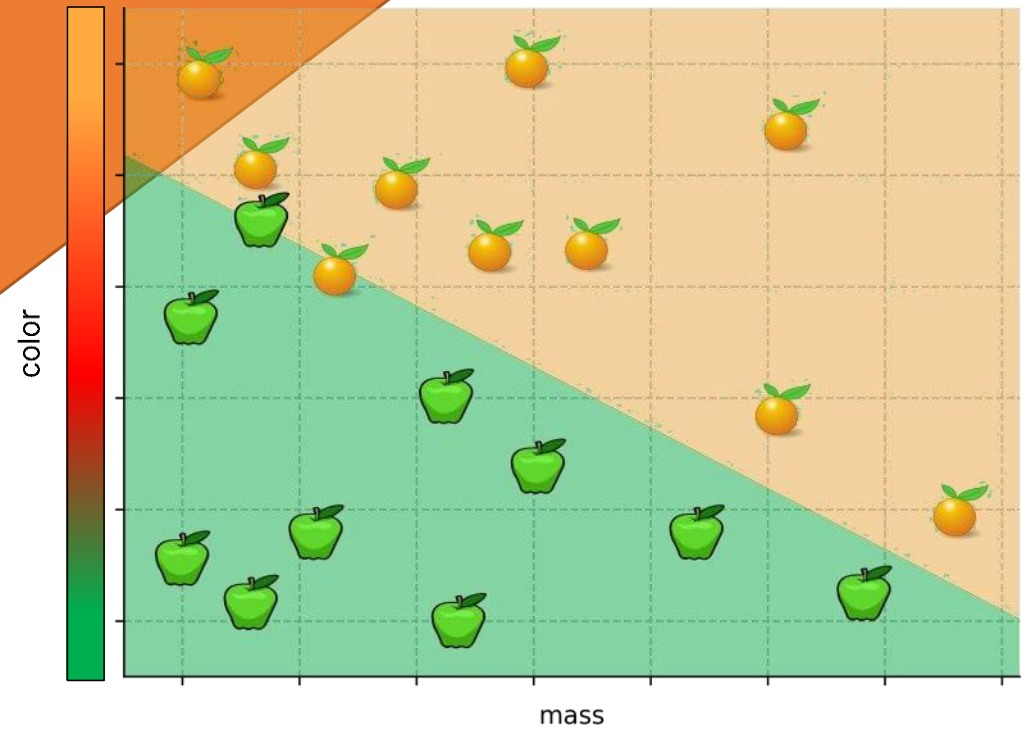
$$Gain(A) = Info(D) - Info_A(D)$$

Radom Forest

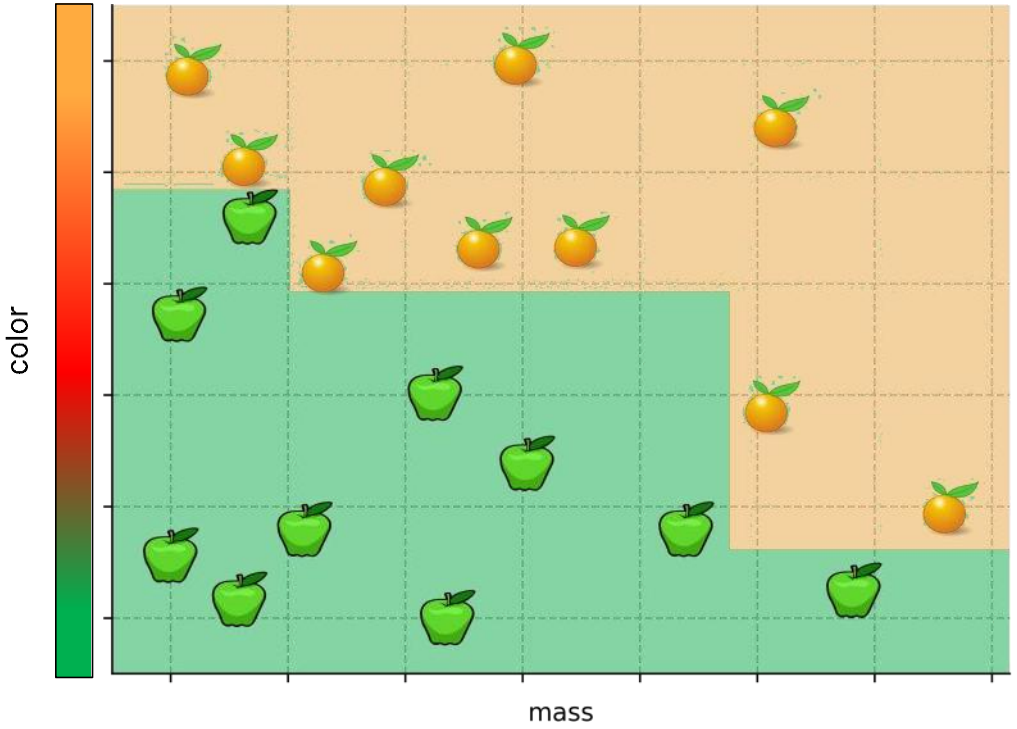
Decision Boundaries

- Decision trees produce non-linear decision boundaries

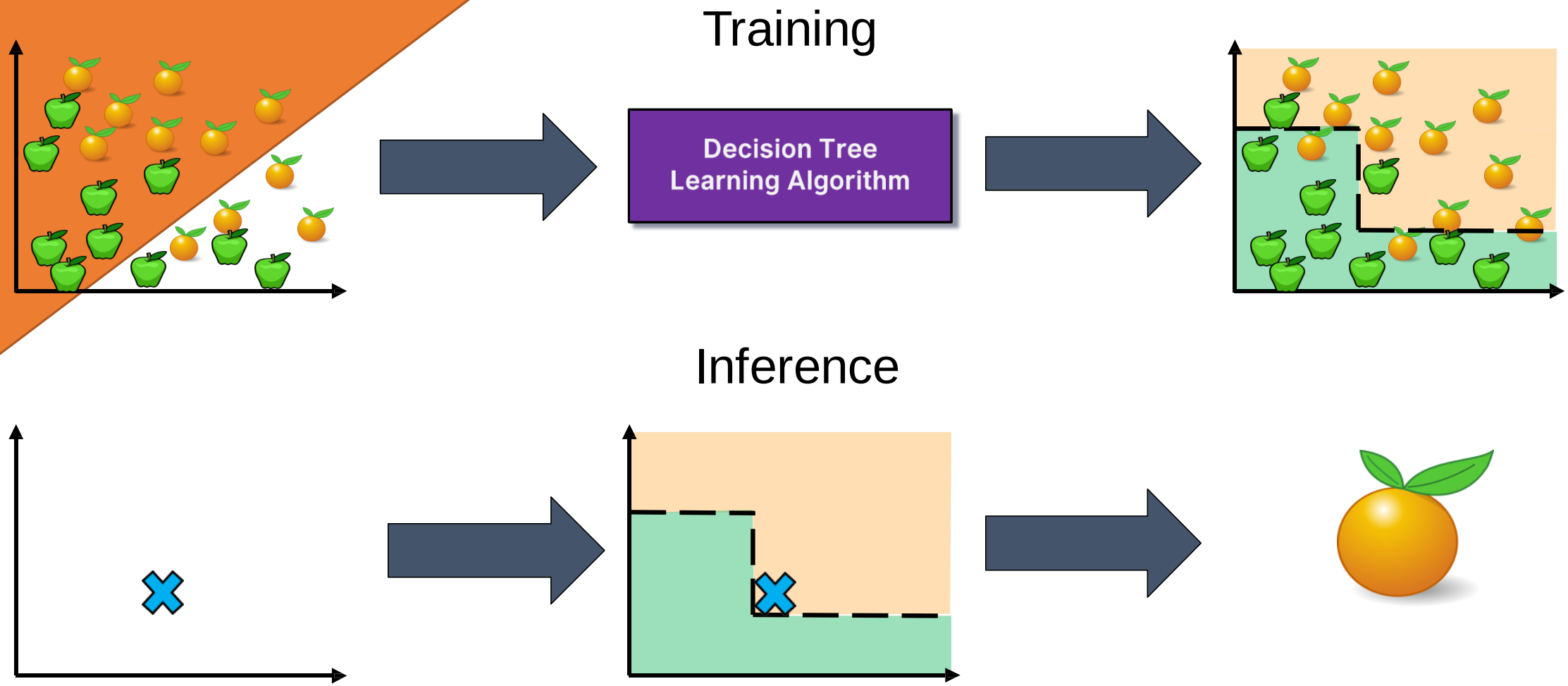
Support Vector Machines



Decision Tree



Decision Trees: Training and Inference



Random Forests

(Ensemble learning with decision trees)

Random Forests

- Random Forests:
  - Instead of building a single decision tree and use it to make predictions, build many slightly different trees and combine their predictions
- We have a single data set, so how do we obtain slightly different trees?

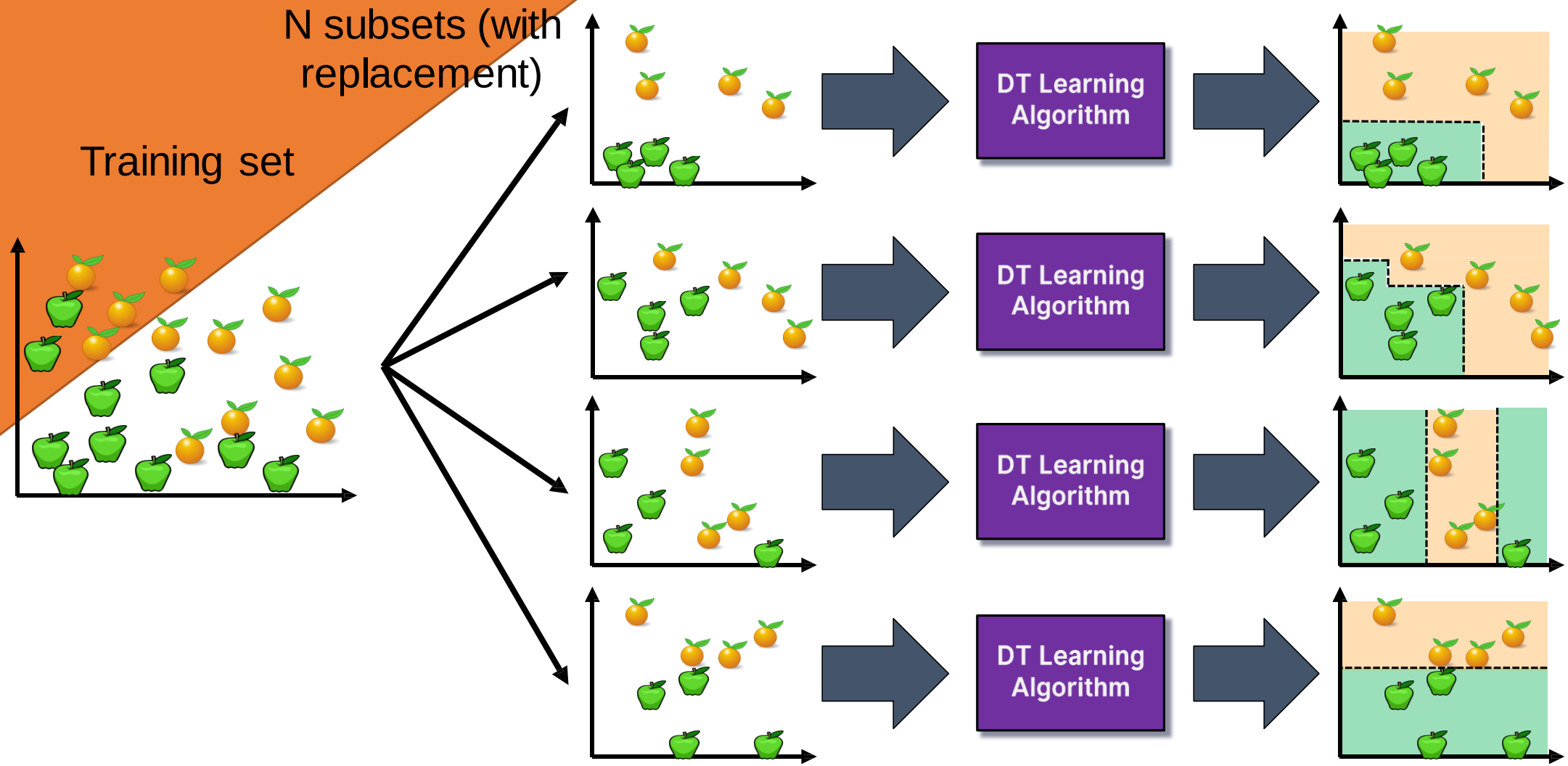
1. Bagging (**B**ootstrap **Agg**regat**ing**):
   - Take random subsets of data points from the training set to create N smaller data sets
   - Fit a decision tree on each subset

2. Random Subspace Method (also known as Feature Bagging):
   - Fit N different decision trees by constraining each one to operate on a random subset of features

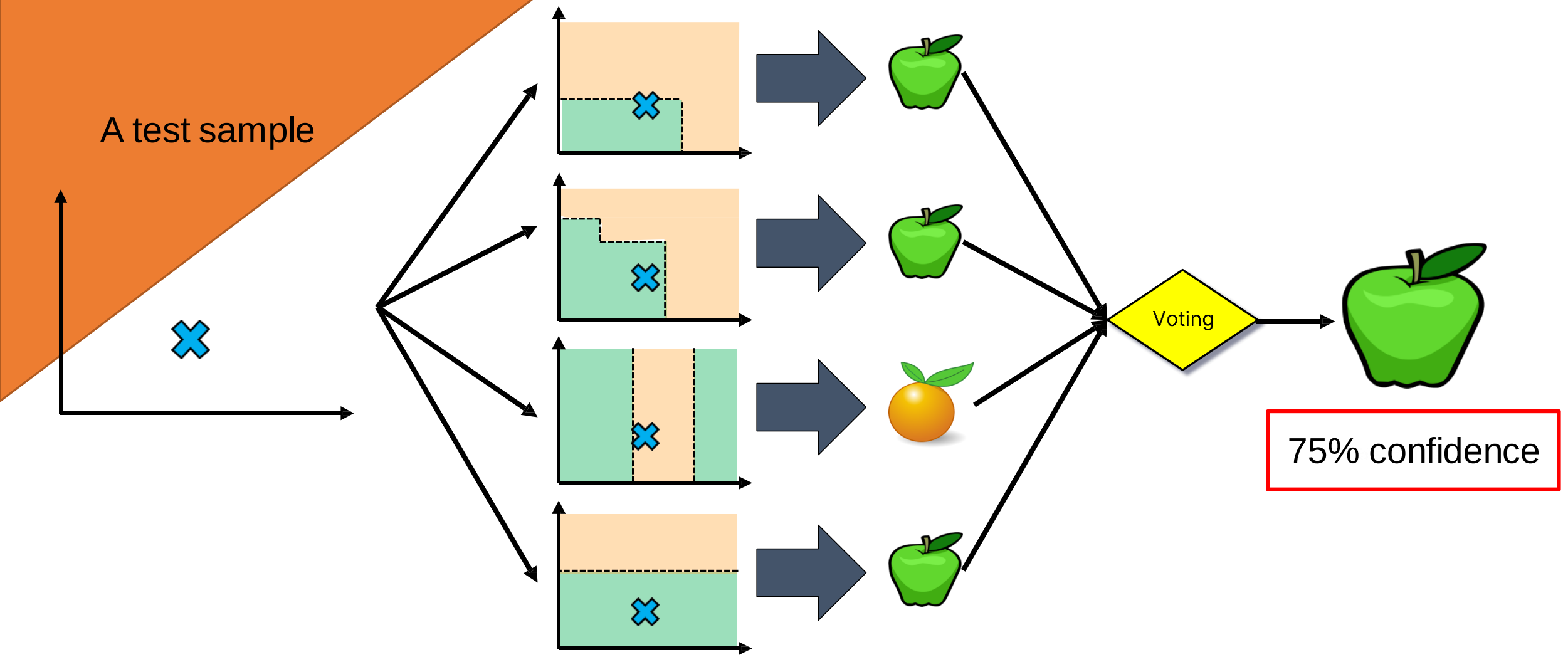
Ensemble Learning

- Ensemble Learning:
  - Method that combines multiple learning algorithms to obtain performance improvements over its components
- **Random Forests** are one of the most common examples of ensemble learning
- Other commonly-used ensemble methods:
  - Bagging: multiple models on random subsets of data samples
  - Random Subspace Method: multiple models on random subsets of features
  - Boosting: train models iteratively, while making the current model focus on the mistakes of the previous ones by increasing the weight of misclassified samples

Bagging at training time



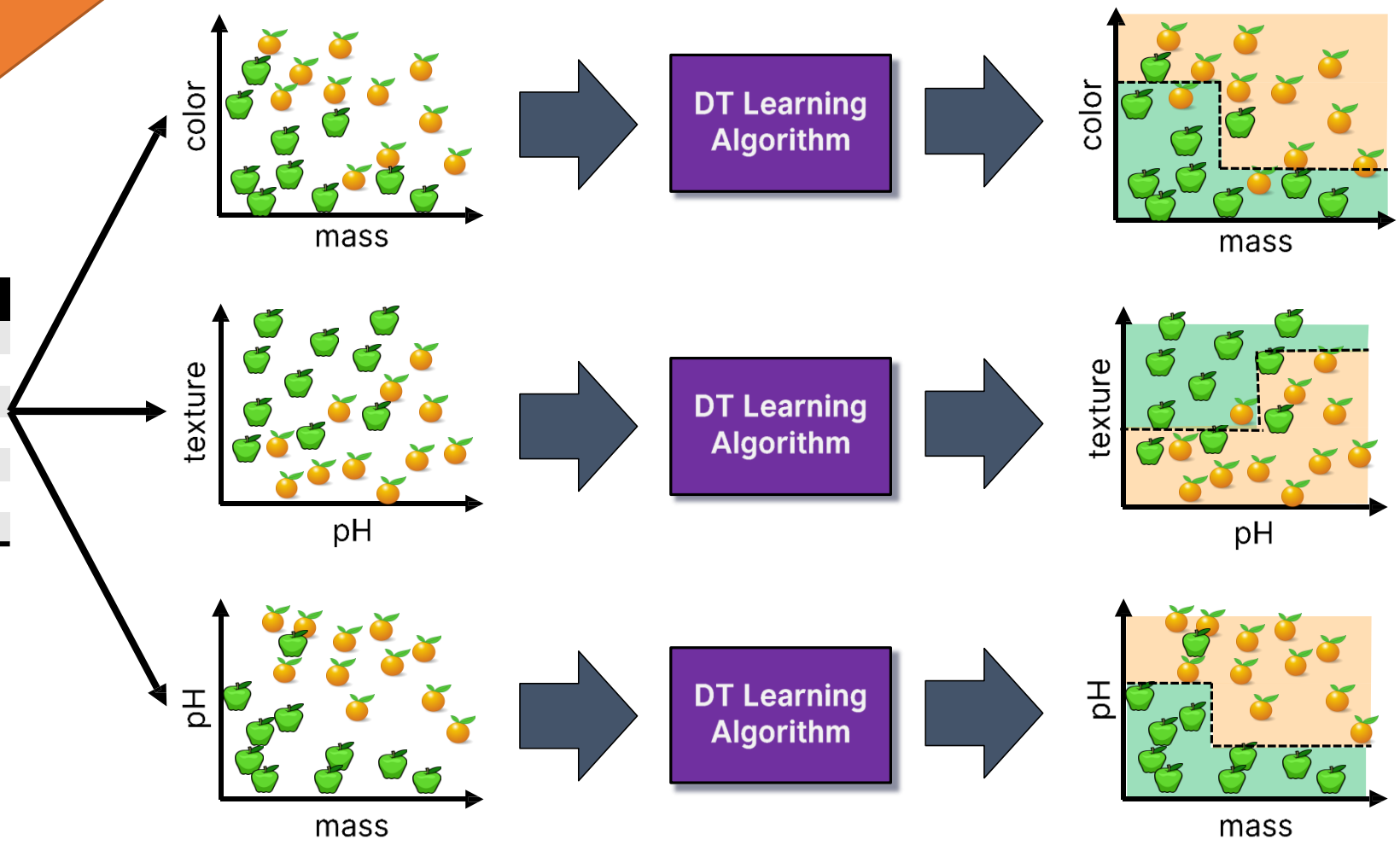
Bagging at inference time



Random Subspace Method at training time

Training data

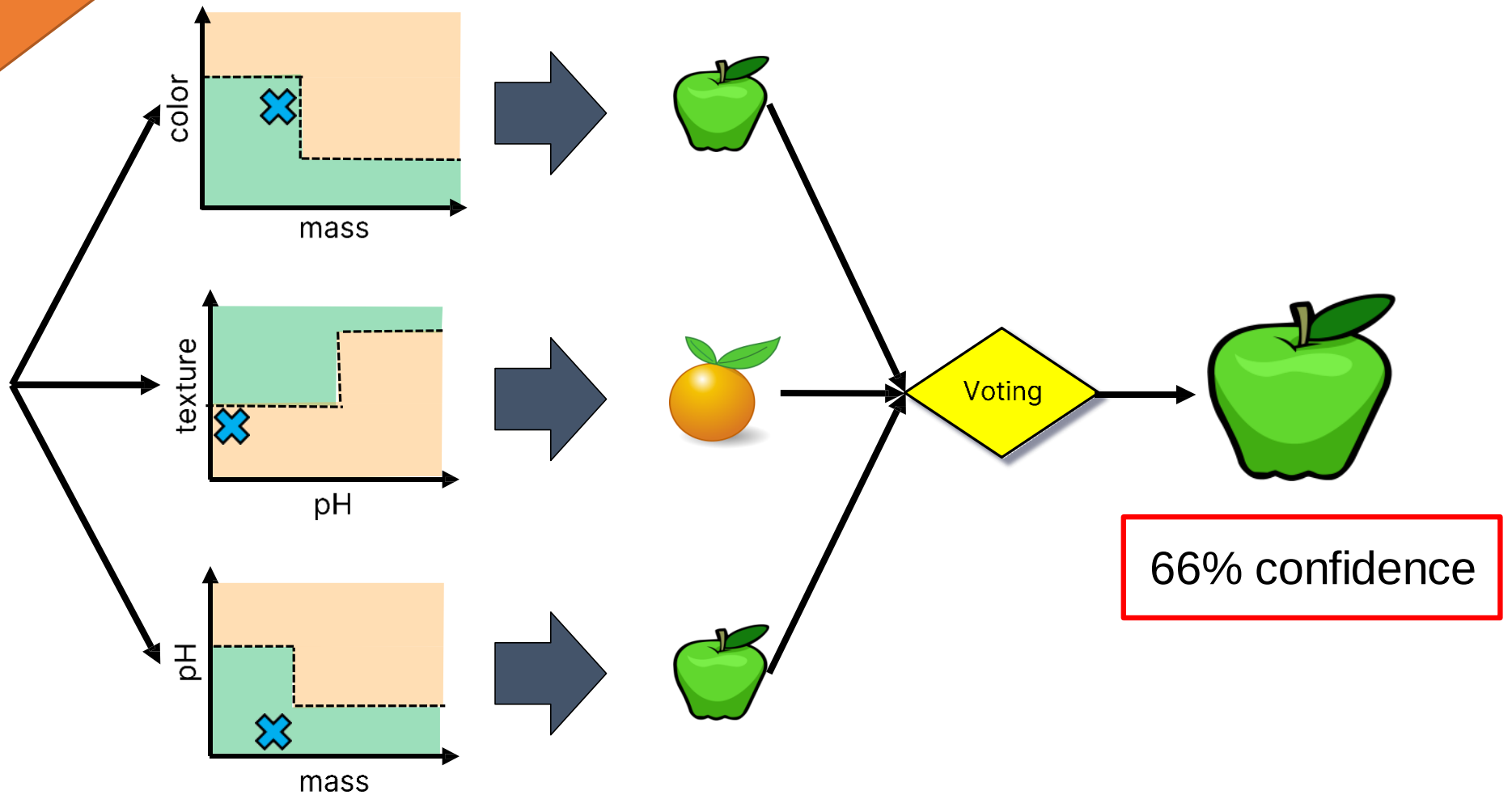
Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	**Apple**
121	Orange	Rough	3.9	**Orange**
85	Red	Smooth	3.3	**Apple**
101	Orange	Smooth	3.7	**Orange**
111	Green	Rough	3.5	**Apple**
...				
117	Red	Rough	3.4	**Orange**



Random Subspace Method at inference time

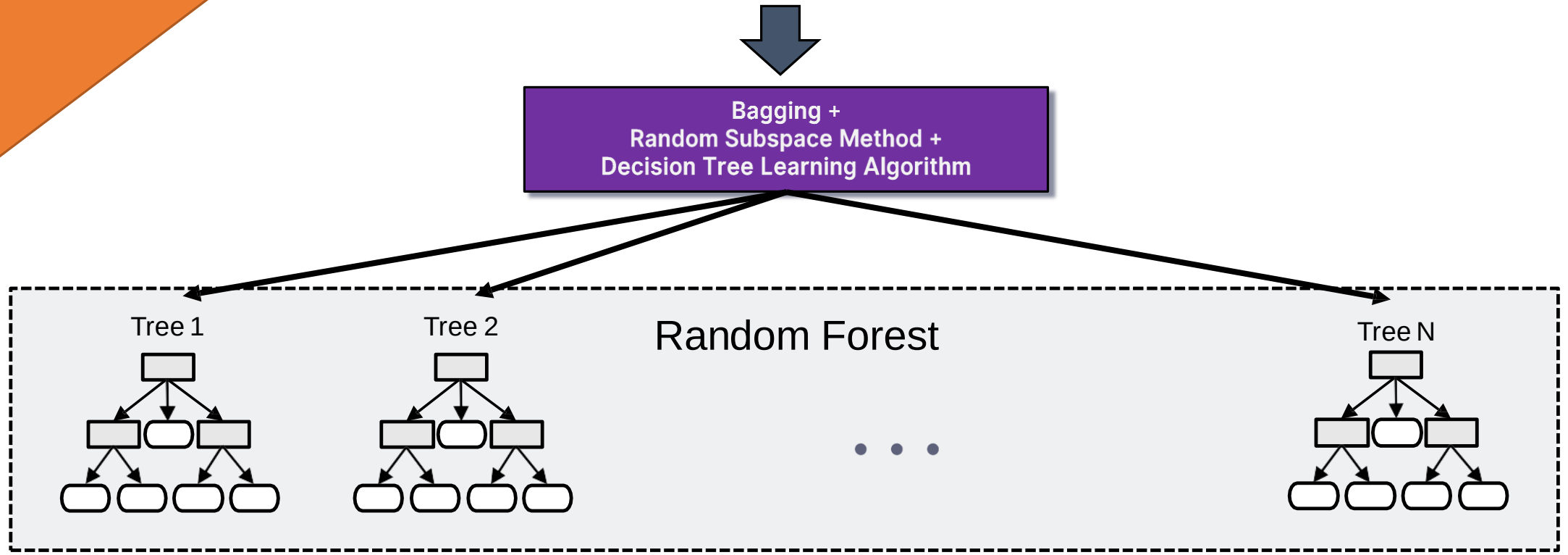
A test sample

87	Red	Smooth	3.1



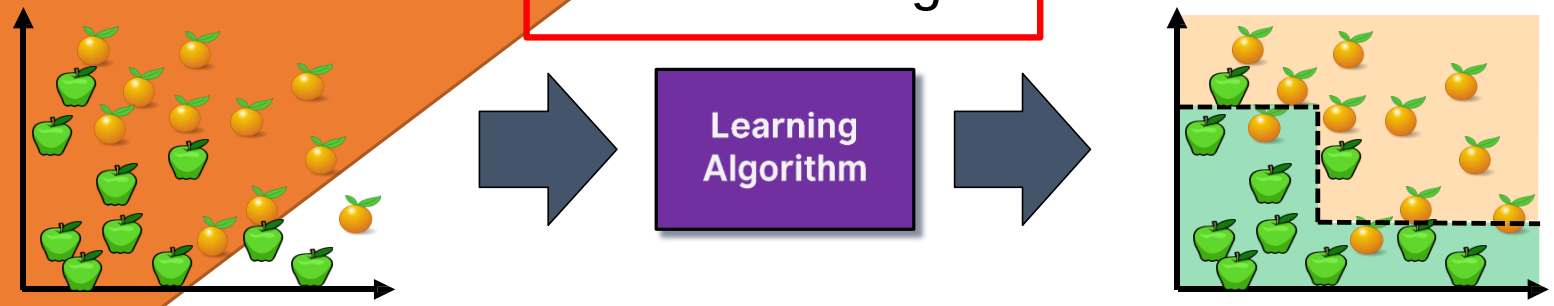
Random Forests

Mass (g)	Color	Texture	pH	Label
84	Green	Smooth	3.5	**Apple**
121	Orange	Rough	3.9	**Orange**
85	Red	Smooth	3.3	**Apple**
101	Orange	Smooth	3.7	**Orange**
111	Green	Rough	3.5	**Apple**
...				
117	Red	Rough	3.4	**Orange**

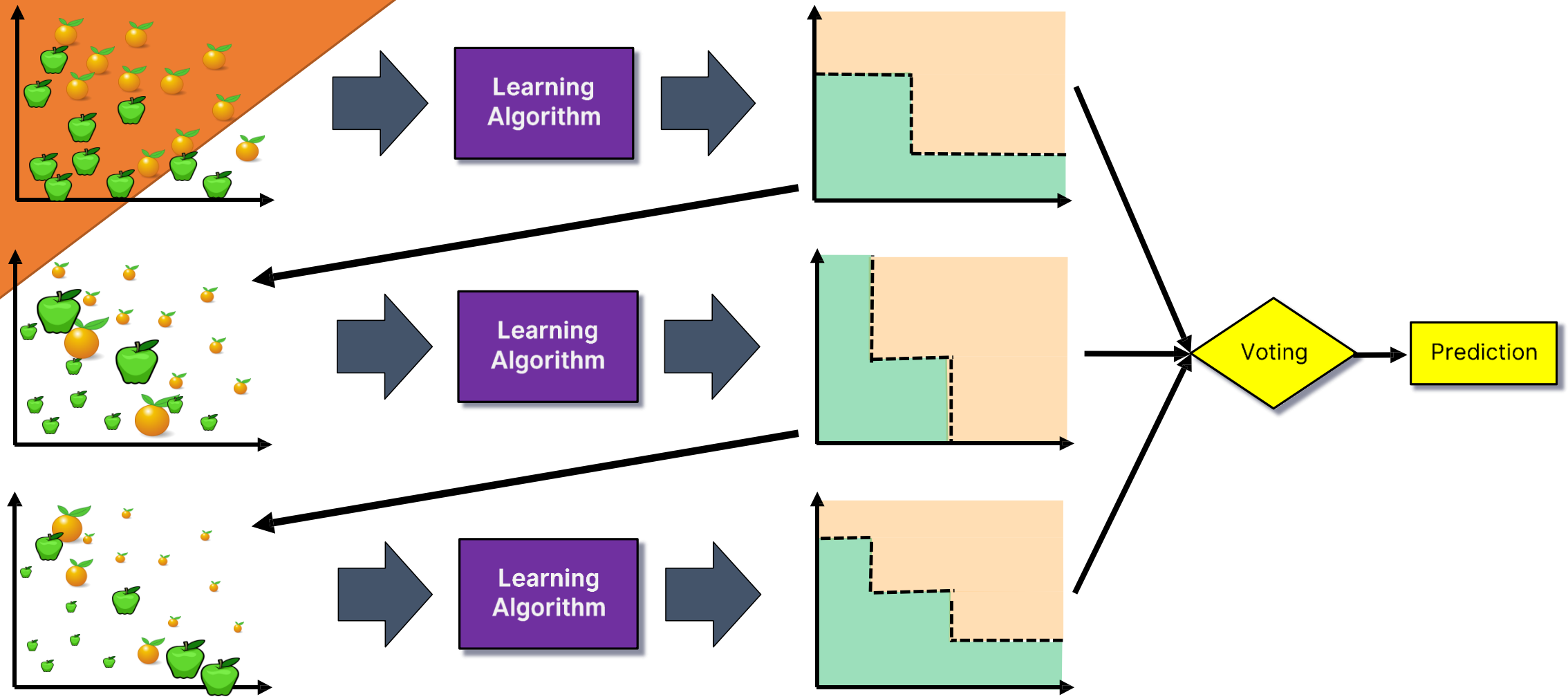


Boosting

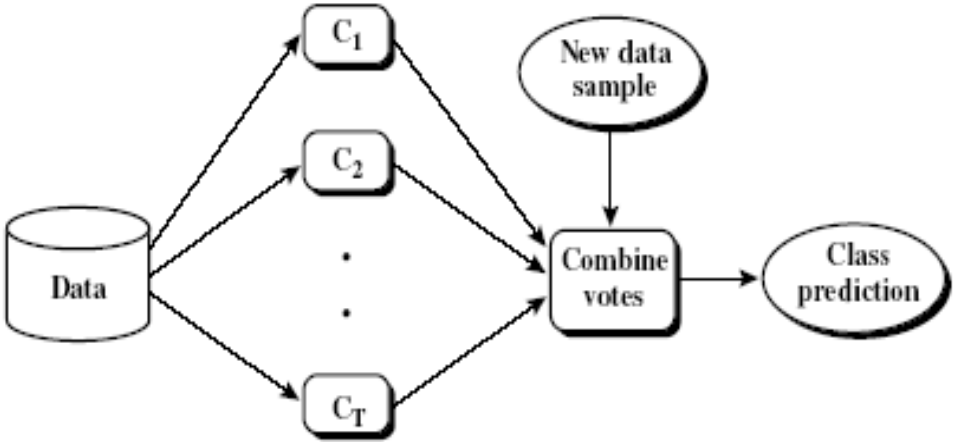
All samples have the same weight



Boosting


Learning Algorithm
Learning Algorithm
Learning Algorithm
Voting
Prediction

Ensemble Methods: Increasing the Accuracy



- **Ensemble methods**
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, …, M_k, with the aim of creating an improved model M*
- **Popular ensemble methods**
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers
  - Ensemble: combining a set of heterogeneous classifiers

Summary

- Ensemble Learning methods combine multiple learning algorithms to obtain performance improvements over its components
- Commonly-used ensemble methods:
  - Bagging (multiple models on random subsets of data samples)
  - Random Subspace Method (multiple models on random subsets of features)
  - Boosting (train models iteratively, while making the current model focus on the mistakes of the previous ones by increasing the weight of misclassified samples)
- **Random Forests** are an ensemble learning method that employ decision tree learning to build multiple trees through **bagging** and **random subspace method**.
  - They rectify the overfitting problem of decision trees!