Introduction to Data Science

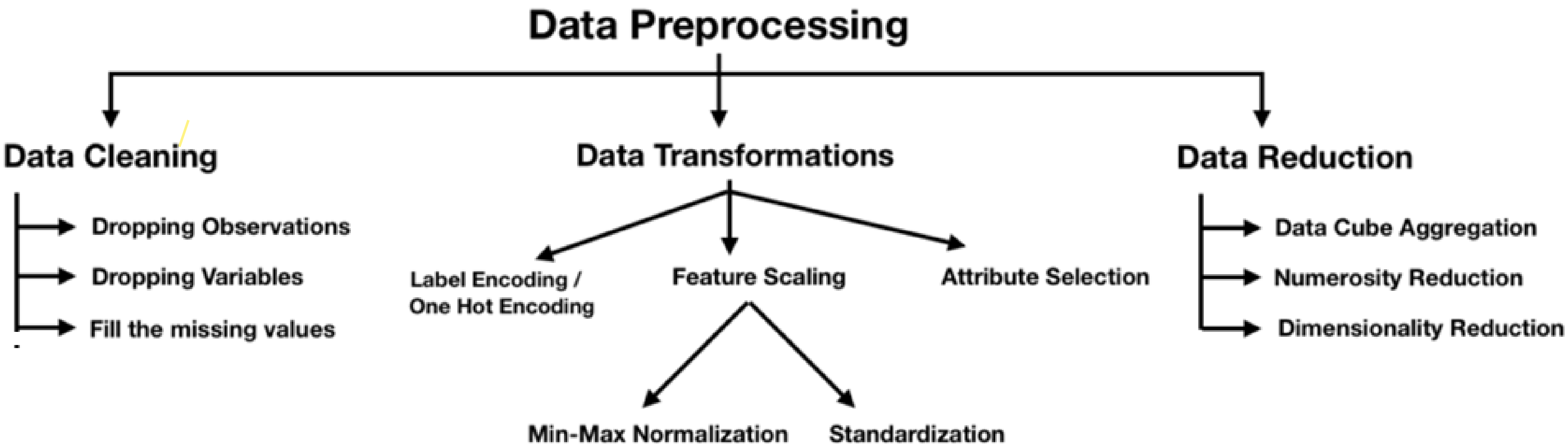
Dr. Irfan Yousuf
Department of Computer Science (New Campus)
UET, Lahore

(Week 13; April 15 - 19, 2024)

Outline

- Data Transformation (Attribute Selection)
  - Feature Generation and Selection
  - Decision Tree

Data Preprocessing



Feature Generation and Selection

- A **feature** represents a measurable piece of data that can be used for analysis.
- **Feature generation** is the process of creating new features from one or multiple existing features, potentially for using in statistical analysis. This process **adds new information** to be accessible during the model construction and therefore hopefully result in more accurate model.
- **Feature selection** is the process of **reducing (or selecting a subset of features) the number** of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

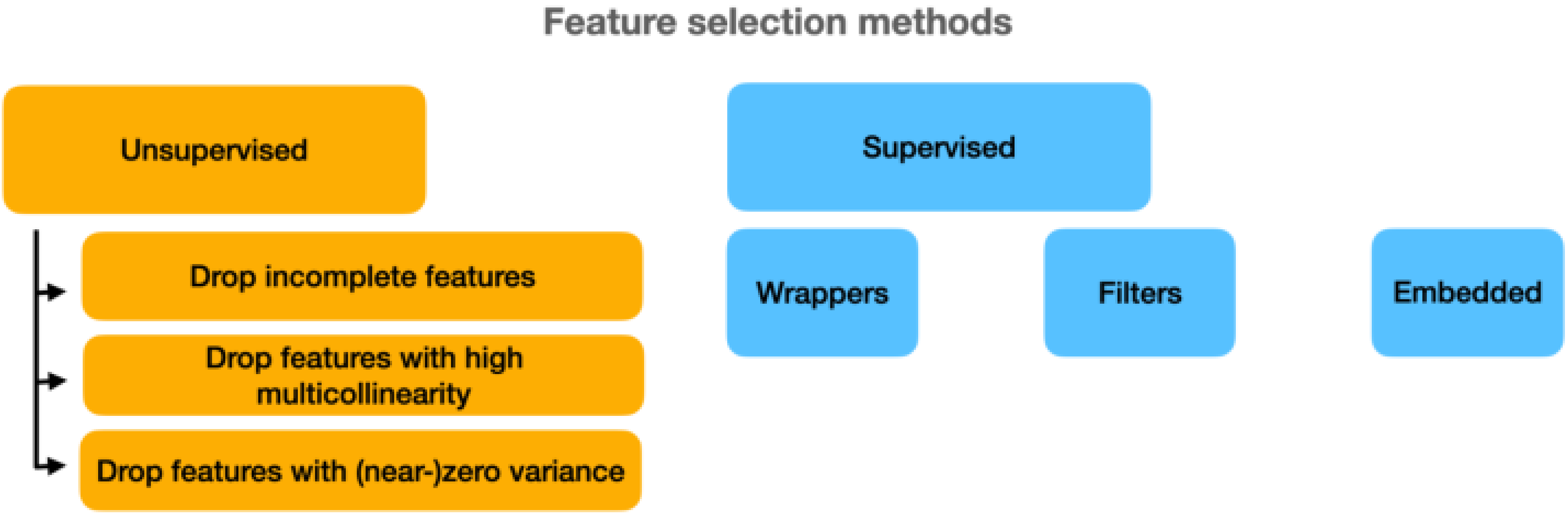
Feature Generation and Selection

	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85

Instance

Feature

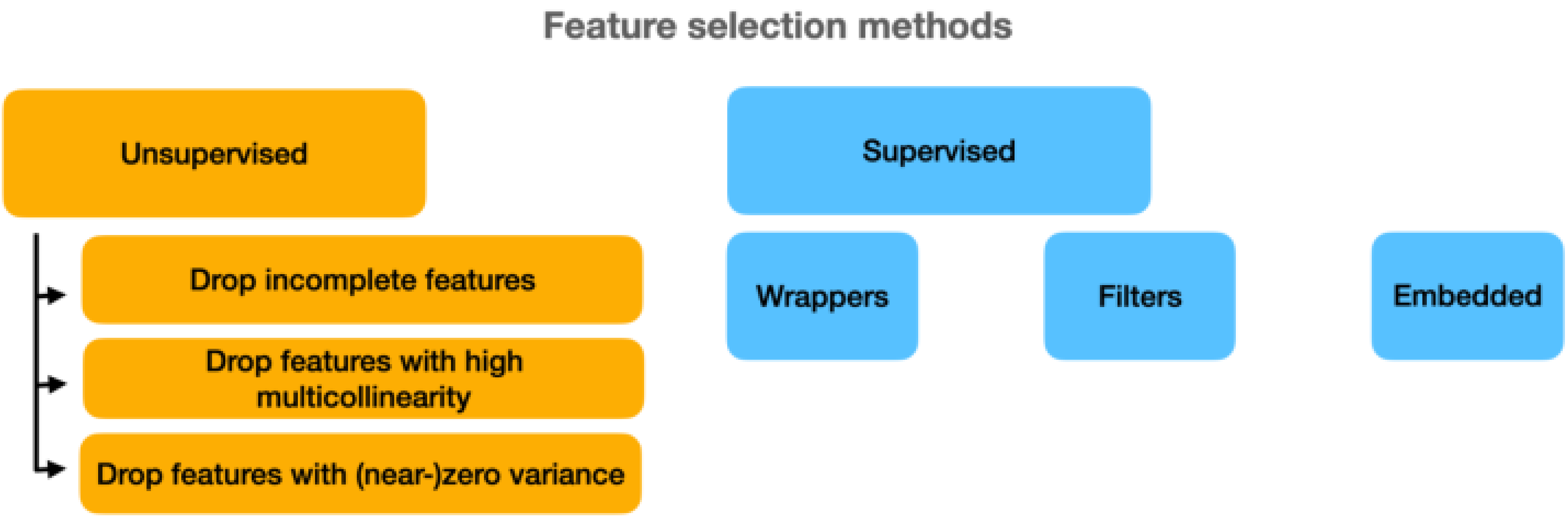
Feature Selection Methods



Unsupervised Feature Selection Methods

- **Zero or near-zero variance:** Features that are (almost) constant provide little information to learn from and thus are irrelevant.
- **Many missing values:** While dropping incomplete features is not the preferred way to handle missing data, it is often a good start, and if too many entries are missing, it might be the only sensible thing to do since such features are likely inconsequential.
- **High multicollinearity:** multicollinearity means a strong correlation between different features, which might signal redundancy issues.

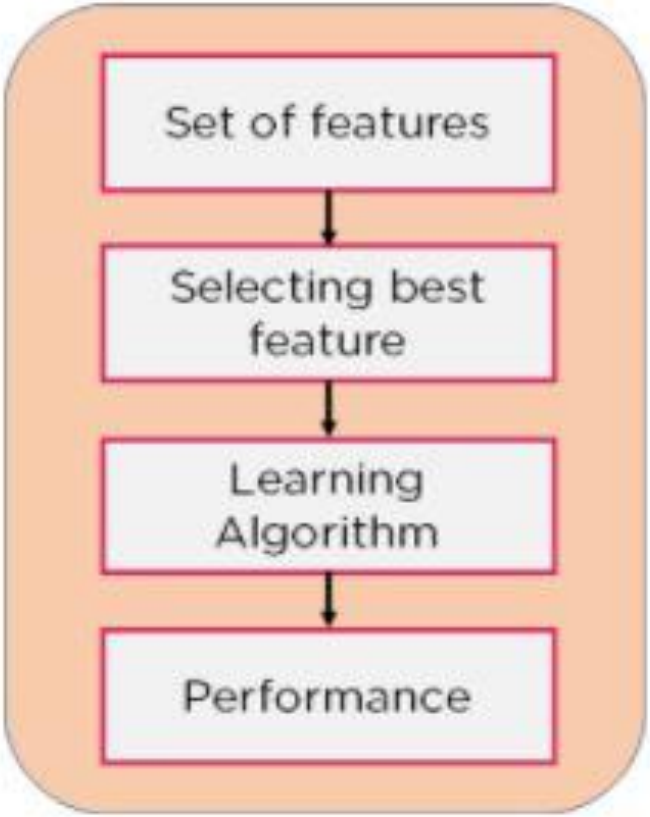
Feature Selection Methods



Filters Methods

- Filter methods are also called as Single Factor Analysis. Using this method, the predictive power of each individual variable (feature) is evaluated.
- Various statistical methods can be used to determine predictive power. One way is by **correlating the feature with the target** (what we are predicting).
- In this method, features are dropped based on their relation to the output, or how they are correlating to the output.
- We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly.

Filters Methods



Filters Methods: Correlation

Numerical Input, Numerical Output

This is a regression predictive modeling problem with numerical input variables.

The most common techniques are to use a correlation coefficient, such as Pearson's for a linear correlation, or rank-based methods for a nonlinear correlation.

- Pearson's correlation coefficient (linear).
- Spearman's rank coefficient (nonlinear)

Filters Methods: Correlation

Correlation Coefficient Formula

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Filters Methods: Information Gain / Mutual Information

Numerical Input, Categorical Output

This is a classification predictive modeling problem with numerical input variables.

This might be the most common example of a classification problem,

Again, the most common techniques are correlation based, although in this case, they must take the categorical target into account.

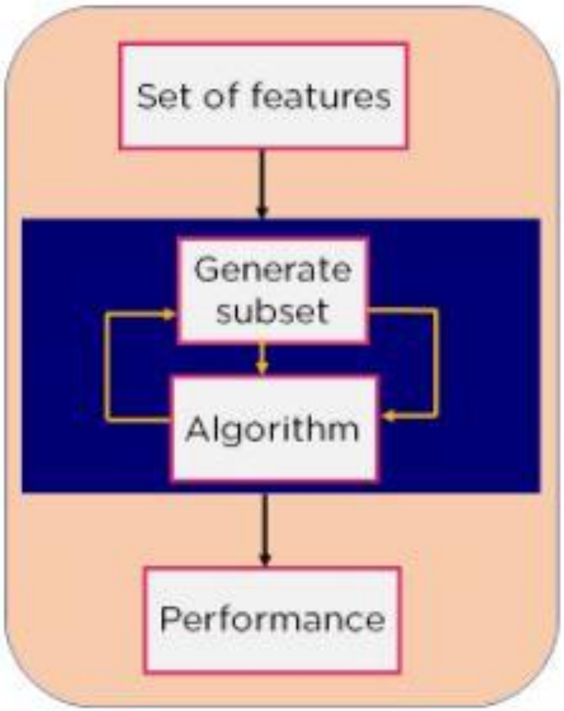
Wrappers Methods

- In wrapper methods, the **feature selection process is based on a specific machine learning algorithm** that we are trying to fit on a given dataset.

- It follows **a greedy search approach by evaluating all the possible combinations of features** against the evaluation criterion.

Wrappers Methods

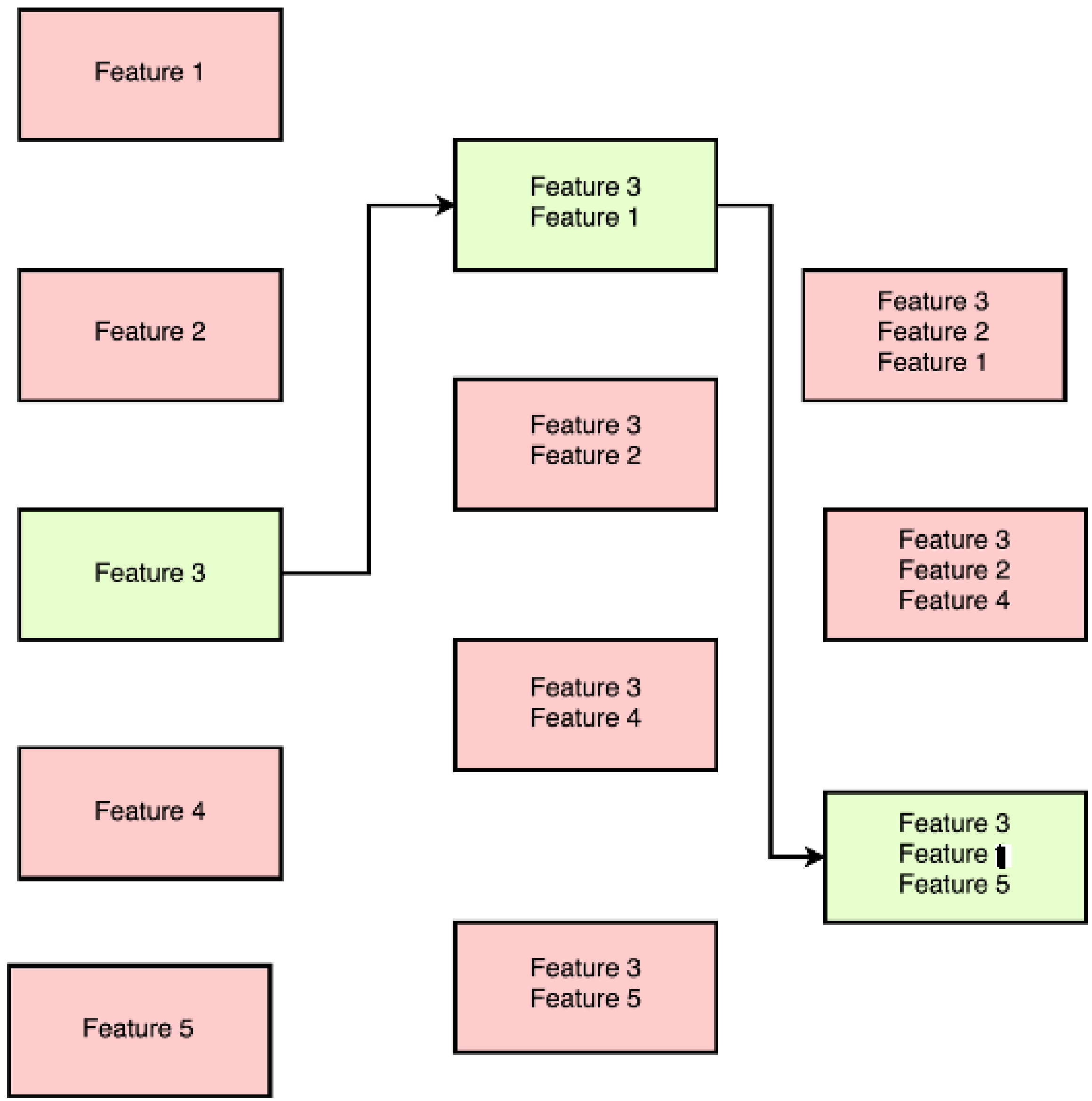
We split our data into subsets and train a model using this. Based on the output of the model, we add and subtract features and train the model again. It forms the subsets using a greedy approach and evaluates the accuracy of all the possible combinations of features



Wrappers Methods: Forward selection

- In **forward selection**, we start with a null model and then start fitting the model with each individual feature one at a time and select the feature with the **best result**.
- Now fit a model with two features by trying combinations of the earlier selected feature with all other remaining features. Again select the feature with the best results.
- Now fit a model with three features by trying combinations of two previously selected features with other remaining features. Repeat this process until we have a set of selected features with **the best result**.

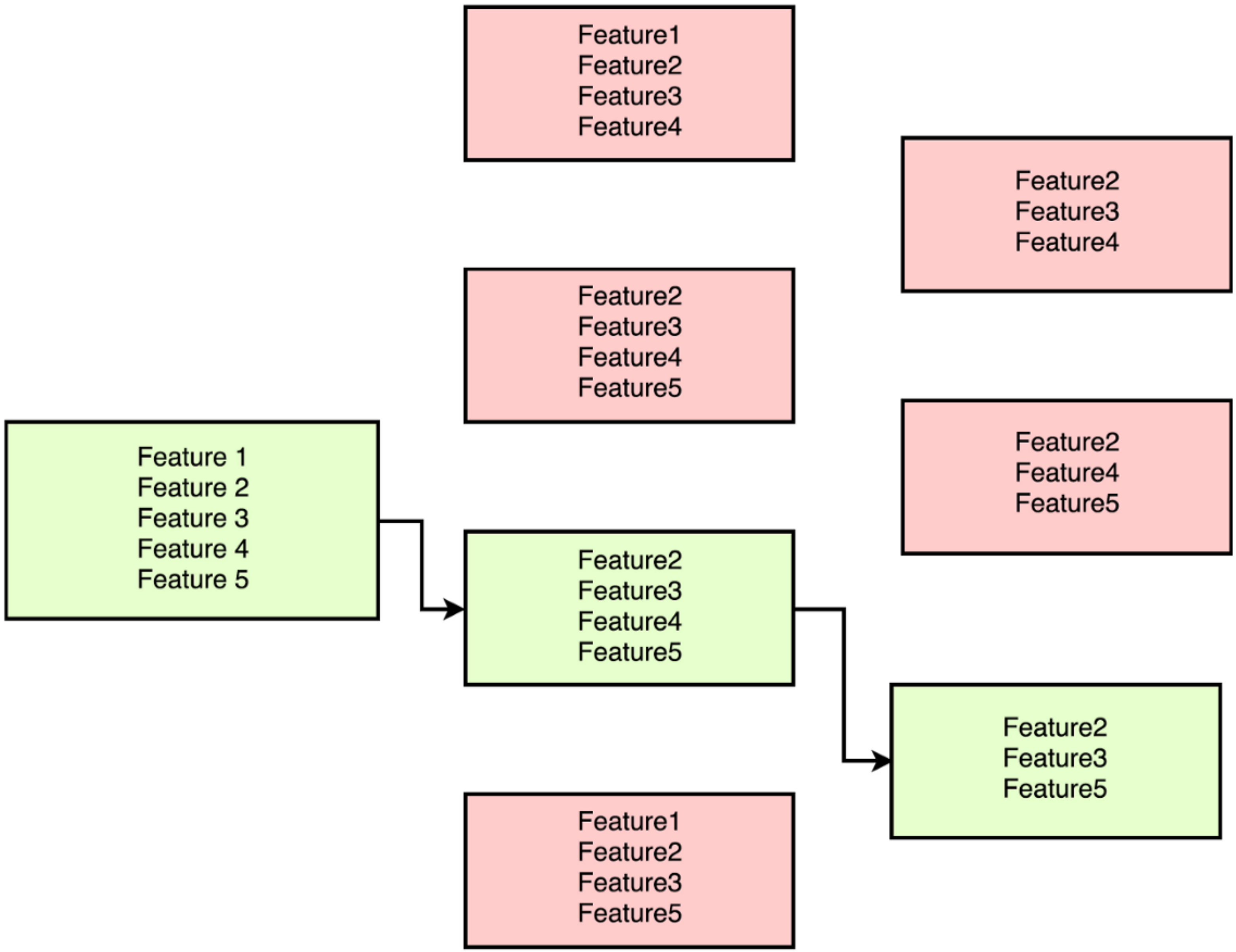
Wrappers Methods: Forward selection



Wrappers Methods: Backward elimination

- In **backward elimination**, we start with the full model (including all the independent variables) and then remove the insignificant feature.

- This process repeats again and again until we have the final set of significant features.


Feature1
Feature2
Feature3
Feature4
Feature2
Feature3
Feature4
Feature2
Feature3
Feature4
Feature5
Feature 1
Feature 2
Feature 3
Feature 4
Feature 5
Feature2
Feature4
Feature5
Feature2
Feature3
Feature4
Feature5
Feature2
Feature3
Feature5
Feature1
Feature2
Feature3
Feature5

Wrappers Methods

- **Bi-directional elimination** is similar to forward selection but the difference is while adding a new feature it also checks the significance of already added features and if it finds any of the already selected features insignificant then it simply removes that particular feature through backward elimination.

Benefits of Feature Selection

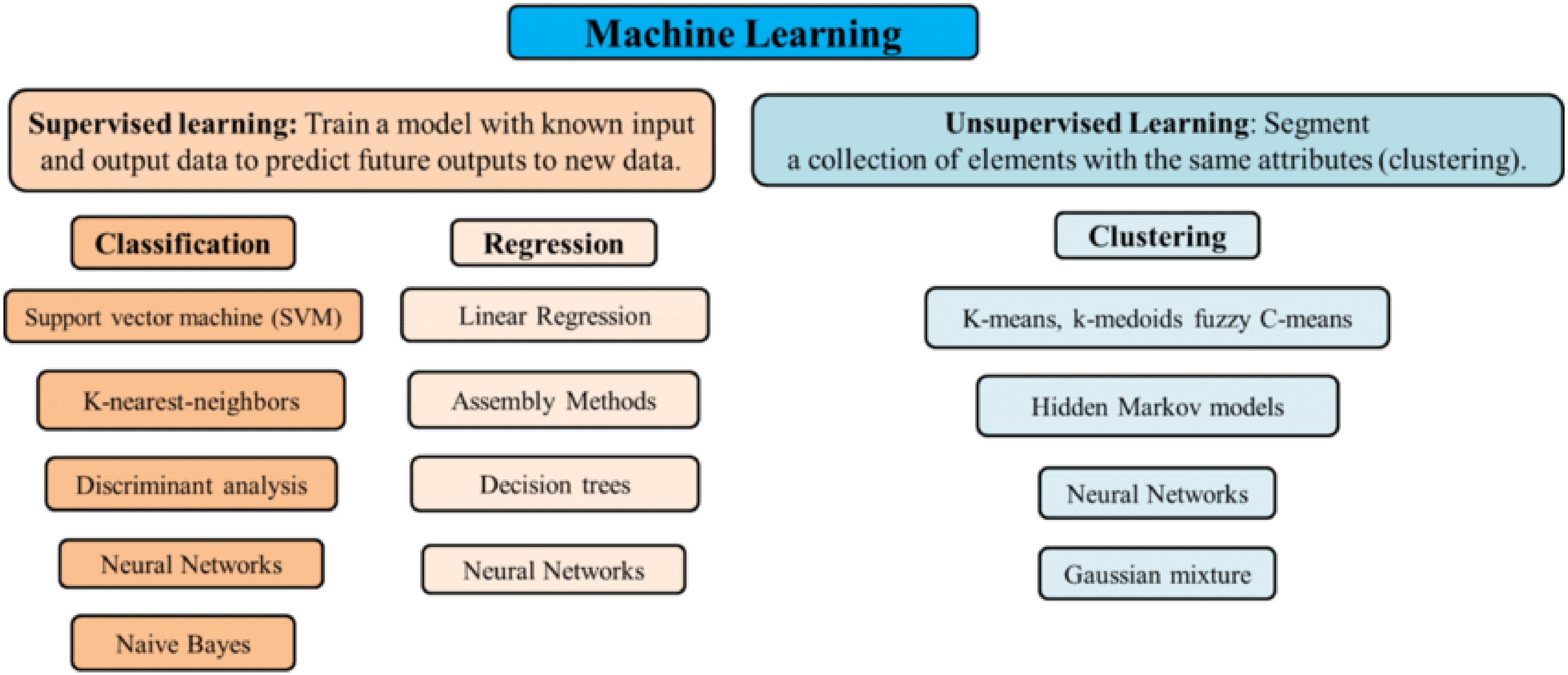
- **Reduction in Model Overfitting:** Less redundant data implies less opportunity to make noise based decisions.
- **Improvement in Accuracy:** Less misleading and misguiding data implies improvement in modeling accuracy.
- **Reduction in Training Time:** Fewer data implies that algorithms train at a faster rate.

Embedded Methods

- In embedded techniques, the feature selection algorithm is integrated as part of the learning algorithm.

- The most typical embedded technique is **decision tree** algorithm. Decision tree algorithms select a feature in each recursive step of the tree growth process and divide the sample set into smaller subsets. The more child nodes in a subset are in the same class, the more informative the features are.

Machine Learning Algorithms



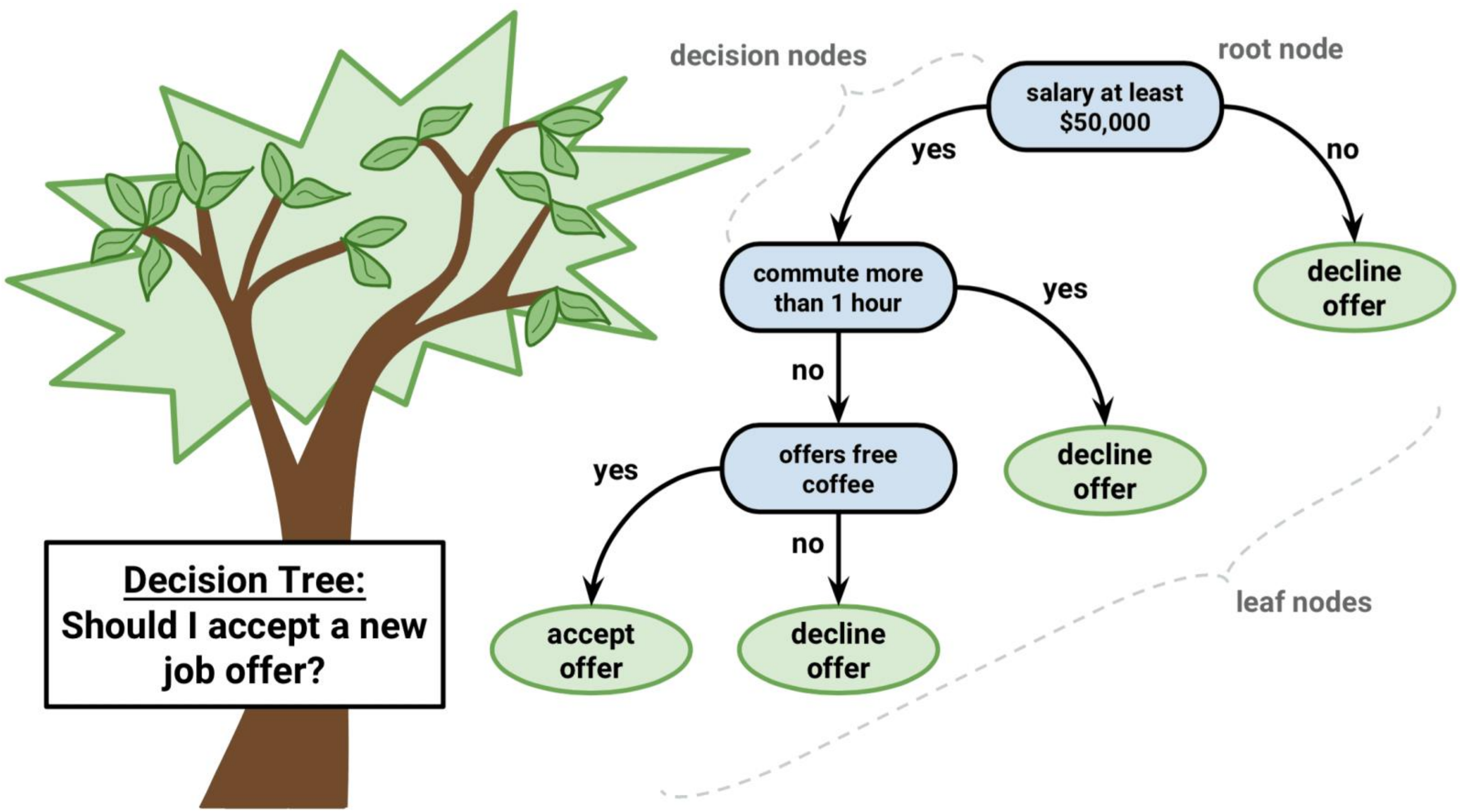
Decision Trees

- It covers both classification and regression.
- It is also used for selecting important features.

- In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.

- A Decision tree is 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.

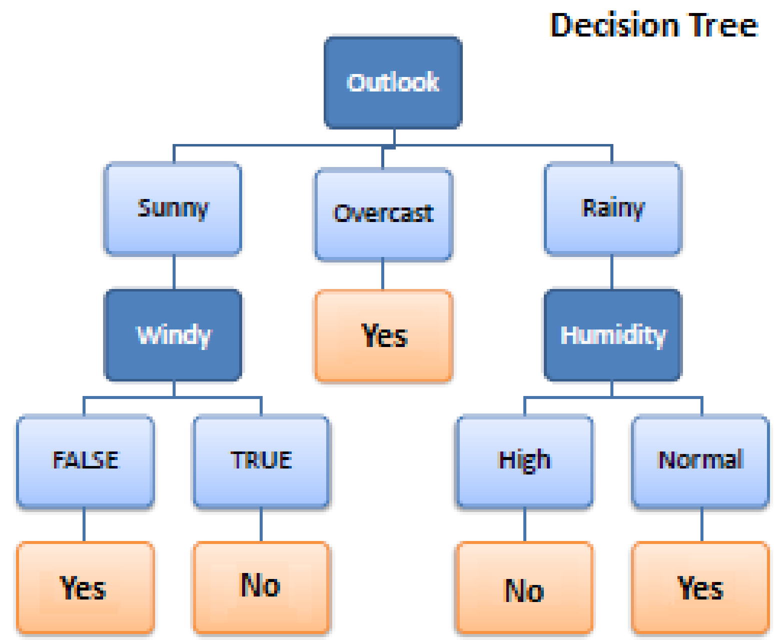
Decision Trees



Decision Trees

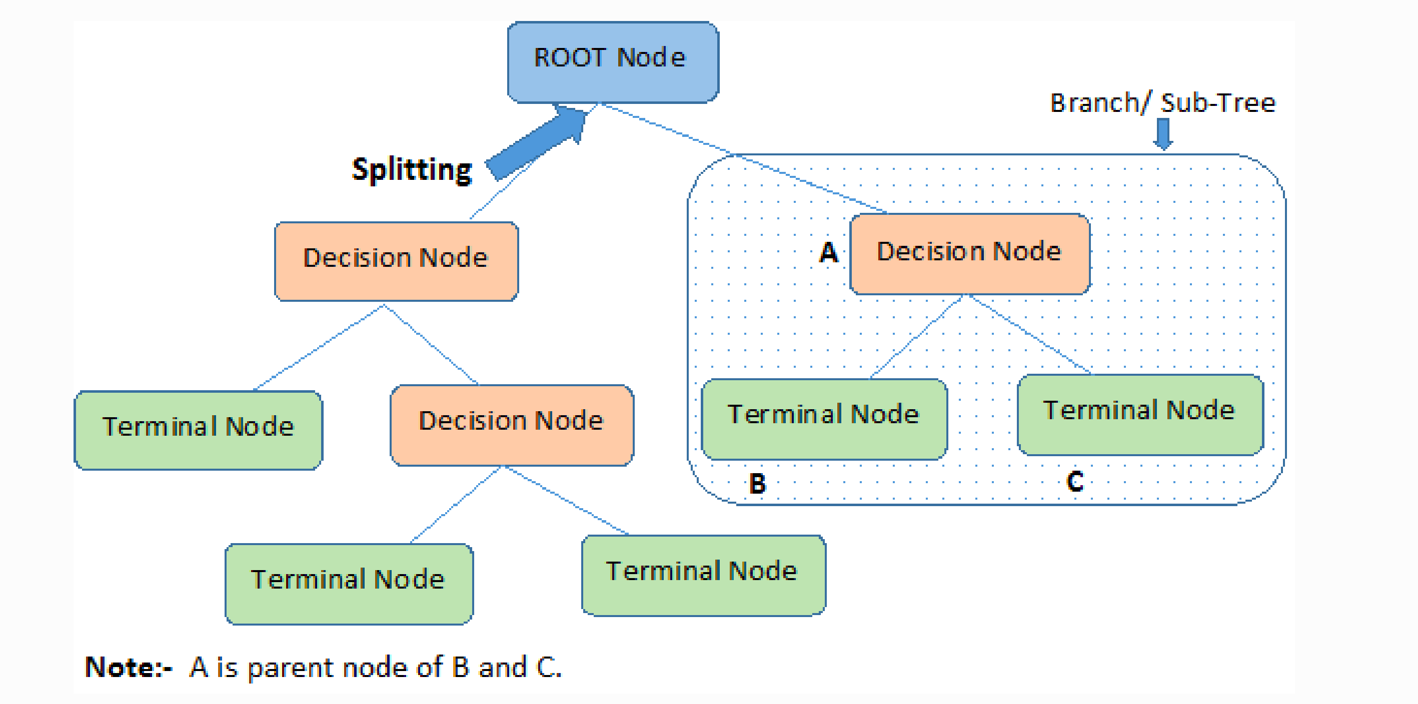
Predictors | Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



Decision Trees

- **Nodes :** Test for the value of a certain attribute.
- **Edges/ Branch :** Correspond to the outcome of a test and connect to the next node or leaf.
- **Leaf / Terminal nodes :** Terminal nodes that predict the outcome (represent class labels or class distribution).



Decision Tree Terminology

- **Root Node:** It represents the entire population or sample, and this further gets divided into two or more homogeneous sets.

- **Splitting:** It is a process of dividing a node into two or more sub-nodes.

- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.

- **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.

Decision Trees

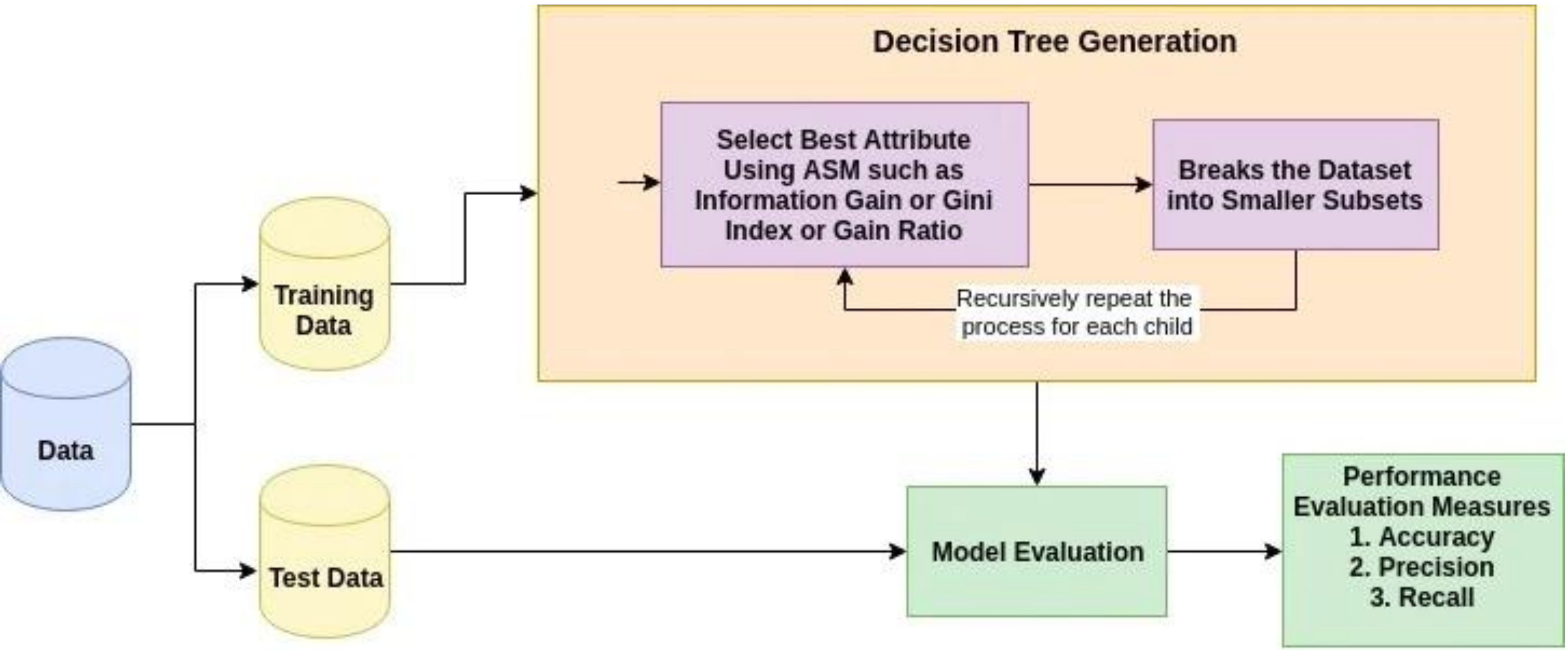
- **Classification Trees** (Yes/No types):
  - decision variable is Categorical/ discrete.
- **Regression Trees**:
  - Where the target variable can take continuous values (typically real numbers) are called regression trees.

How to Create a Decision Tree?

The basic idea behind any decision tree algorithm is as follows:

1. Select the best attribute using Attribute Selection Measures (ASM) to split the records.
2. Make that attribute a decision node and breaks the dataset into smaller subsets.
3. Start tree building by repeating this process recursively for each child until one of the conditions will match:
    - All the tuples belong to the same attribute value.
    - There are no more remaining attributes.
    - There are no more instances.

How to Create a Decision Tree?

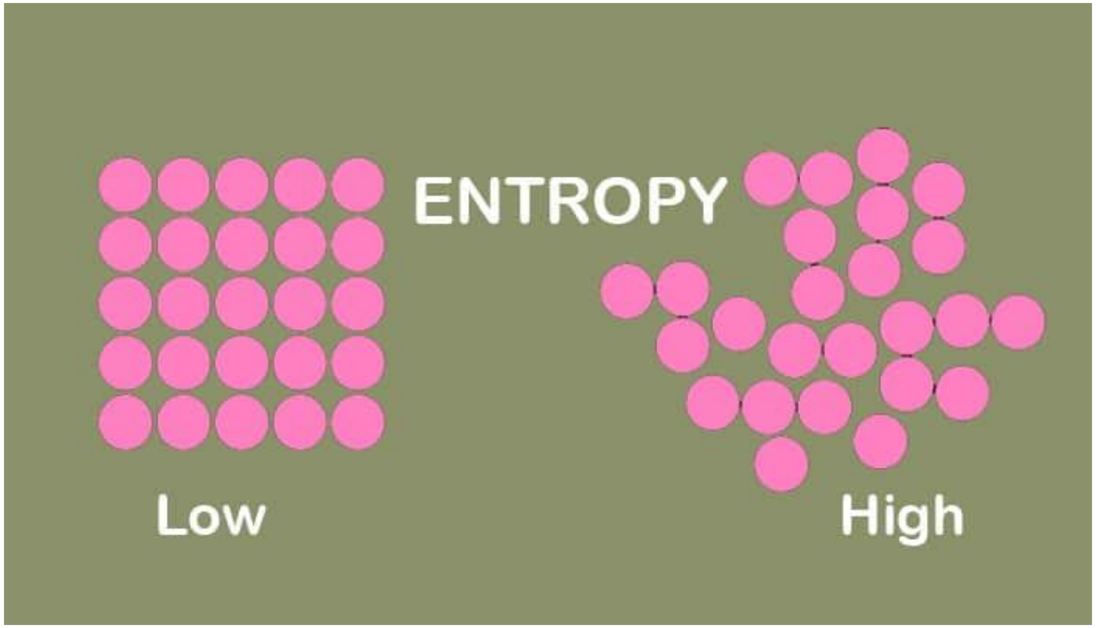


How to Create a Decision Tree?

- The primary challenge in the decision tree construction is to **identify which features** do we need to consider as the root node and each level.
- Handling this is known as the feature selection or Attribute Selection Measures (ASM). We have different attributes selection measures to identify the attribute which can be considered as the root note at each level.
  - Entropy
  - Information Gain
  - Reduction in variance
  - Chi-square
  - Gini Index
  - Gain Ratio

Entropy

- Entropy is a measure of the randomness in the information being processed.
- The higher the entropy, the harder it is to draw any conclusions from that information.
  - Example: Flipping a coin is an example of an action that provides information that is random.



Entropy

- Mathematically Entropy for 1 attribute is represented as:

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

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Where S → Current state, and Pi → Probability of an event *i* of state S or Percentage of class *i* in a node of state S.

Entropy

- Mathematically Entropy for multiple attributes is represented as:

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

where T→ Current state and X → Selected attribute

Information Gain

- Information gain or IG is a statistical property that **measures how well a given attribute separates the training examples according to their target classification.**

- Constructing a decision tree is all about finding an attribute that returns the **highest information gain** and the **smallest entropy**.

- Information gain is a decrease in entropy. It computes the **difference between entropy before split** and **average entropy after split** of the dataset based on given attribute values.

Information Gain

$$\text{Information Gain}(T,X) = \text{Entropy}(T) - \text{Entropy}(T, X)$$

$$\begin{aligned} IG(\text{PlayGolf, Outlook}) &= E(\text{PlayGolf}) - E(\text{PlayGolf, Outlook}) \\ &= 0.940 - 0.693 \\ &= 0.247 \end{aligned}$$

$$\text{Information Gain} = \text{Entropy}(before) - \sum_{j=1}^{K} \text{Entropy}(j,\ after)$$

Where "before" is the dataset before the split, K is the number of subsets generated by the split, and (j, after) is subset j after the split.

Decision Tree: Numerical Example

Predictors				Target
Outlook	**Temp.**	**Humidity**	**Windy**	**Play Golf**
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Decision Tree: Numerical Example

$$Entropy(PlayGolf) = \mathbf{E}(5,9)$$

$$\boldsymbol{E(PlayGolf)} = E(5,9)$$

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

$$= -(0.357\ \log_2 0.357) - (0.643\ \log_2 0.643)$$

$$= \mathbf{0.94}$$

Decision Tree: Numerical Example

Calculate Entropy for Other Attributes After Split.
For the other four attributes, we need to calculate the entropy after each of the split.

- E(PlayGolf, Outloook)
- E(PlayGolf, Temperature)
- E(PlayGolf, Humidity)
- E(PlayGolf,Windy)

Decision Tree: Numerical Example

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

$$\boldsymbol{E(Sunny)} = E(3,2)$$

$$= -\left(\frac{3}{5}\log_2\frac{3}{5}\right) - \left(\frac{2}{5}\log_2\frac{2}{5}\right)$$

$$= -(0.60\ \log_2 0.60) - (0.40 \log_2 0.40)$$

$$= -(0.60 * 0.737) - (0.40 * 0.529)$$

$$\boldsymbol{= 0.971}$$

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

$$\boldsymbol{E}(PlayGolf, Outlook) = \boldsymbol{P}(Sunny)\boldsymbol{E}(Sunny) + \boldsymbol{P}(Overcast)\boldsymbol{E}(Overcast) + \boldsymbol{P}(Rainy)\boldsymbol{E}(Rainy)$$

E(S, outlook) = (5/14)*E(3,2) + (4/14)*E(4,0) + (5/14)*E(2,3) = (5/14)(-(3/5)log(3/5)-(2/5)log(2/5))+ (4/14)(0) + (5/14)((2/5)log(2/5)-(3/5)log(3/5)) = 0.693

Decision Tree: Numerical Example

The next step is to find the information gain. It is the difference between **parent entropy** and **average weighted entropy** (after split) we found.

Information Gain(T,X) = Entropy(T) - Entropy(T, X)

$$\begin{aligned} IG(PlayGolf, Outlook) &= E(PlayGolf) - E(PlayGolf, Outlook) \\ &= 0.940 - 0.693 \\ &= 0.247 \end{aligned}$$

Decision Tree: Numerical Example

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

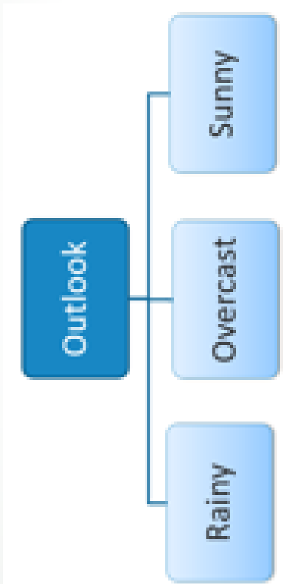
		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

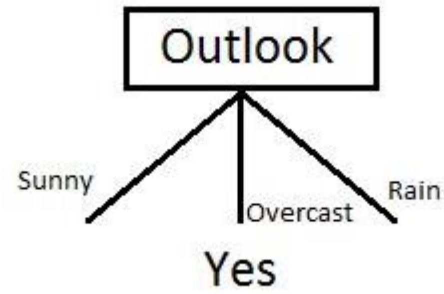
$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

Decision Tree: Numerical Example

Now select the feature **having the largest Information gain**. Here it is Outlook. So, it forms the first node(root node) of our decision tree.



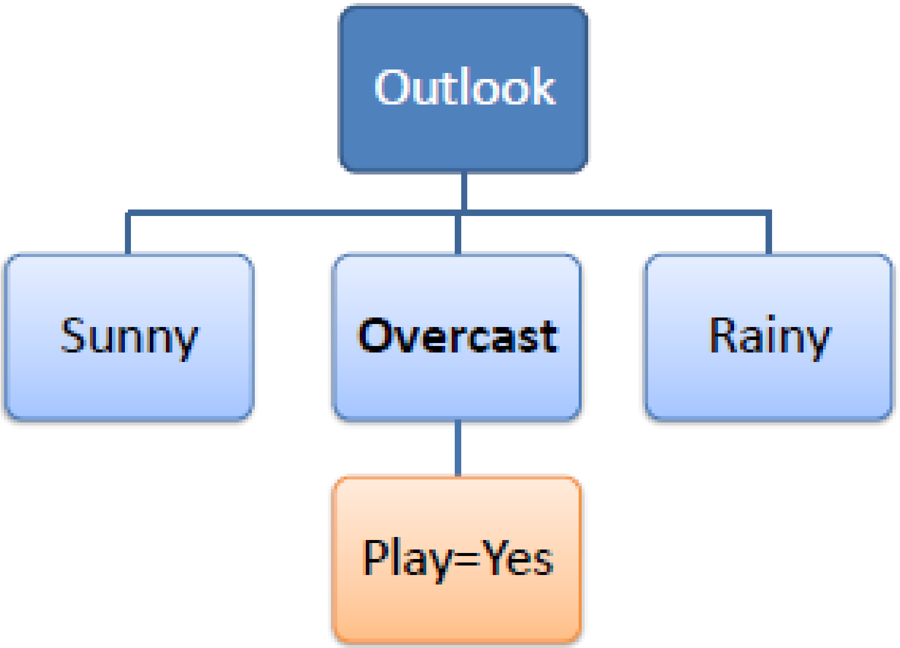
Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes



Decision Tree: Numerical Example

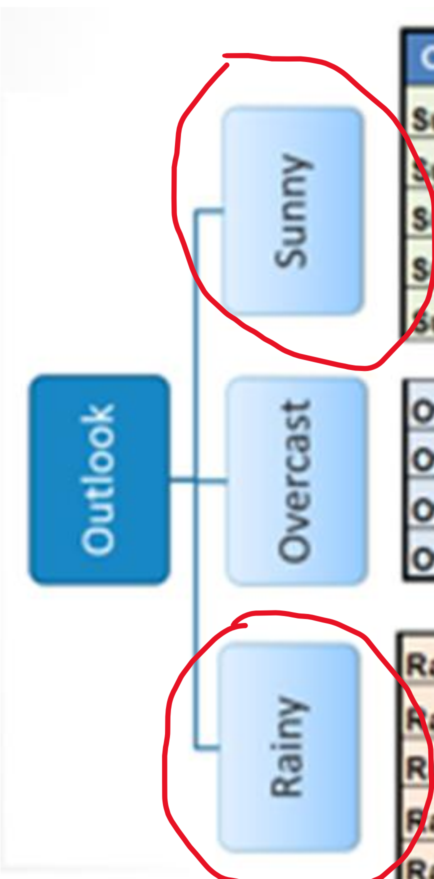
A branch with entropy of 0 is a leaf node.

Temp.	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Decision Tree: Numerical Example

A branch with entropy more than 0 needs further splitting.



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Decision Tree: Numerical Example

The next step is to find the next node in our decision tree. Now we will find **one under sunny**. We have to determine which of the following Temperature, Humidity or Wind has higher information gain.

Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Calculate parent entropy E(sunny)

$$E(sunny) = (-(3/5)\log(3/5)-(2/5)\log(2/5)) = 0.971.$$

Decision Tree: Numerical Example

$E(sunny) = (-(3/5)\log(3/5)-(2/5)\log(2/5)) = 0.971.$

E(Sunny, Temperature) = ?

IG(Sunny, Temperature) = 0.971 -

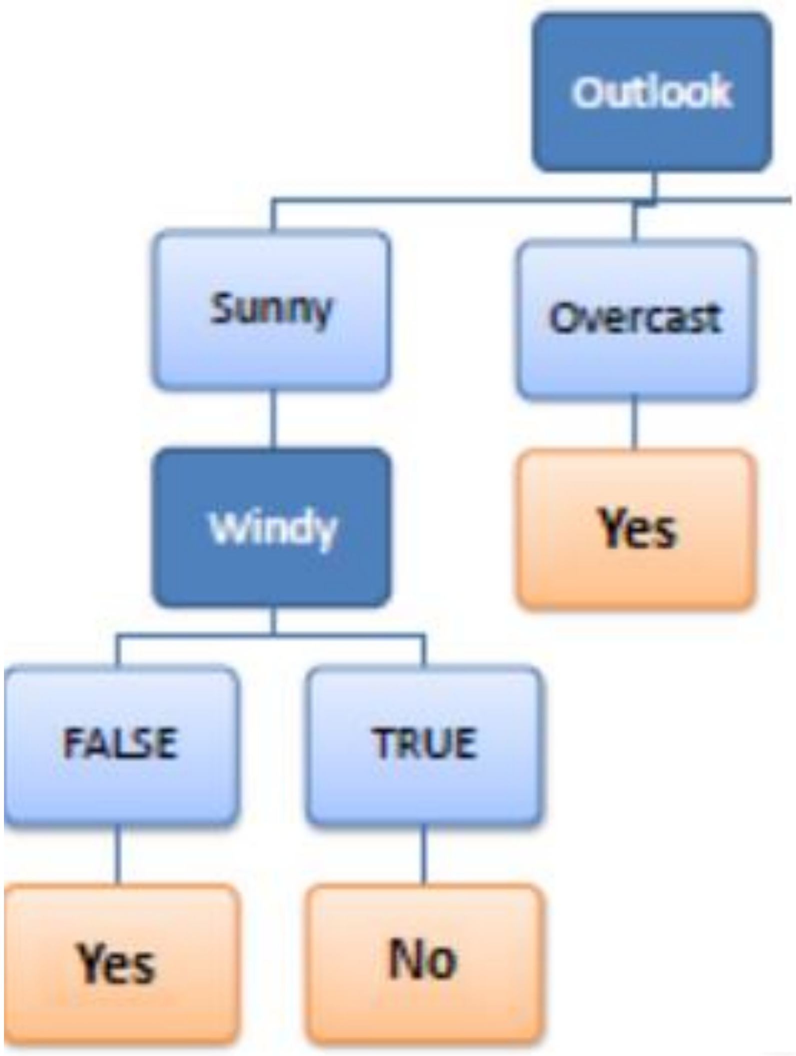
E(Sunny, Humidity) = ?

IG(Sunny, Humidity) = 0.971 -

E(Sunny, Windy) = ?

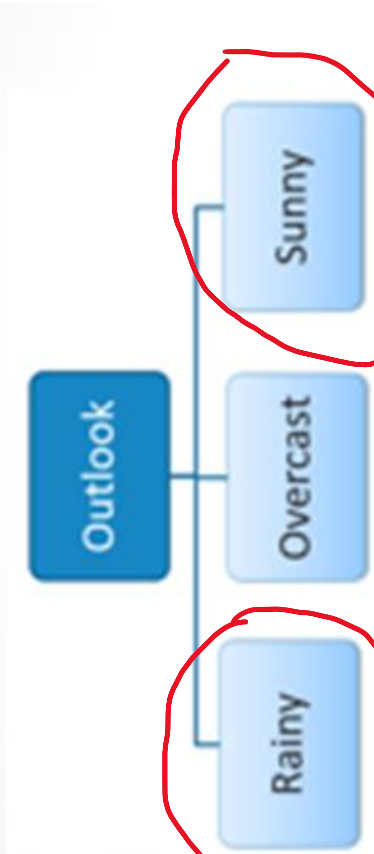
IG(Sunny, Windy) = 0.971 -

Decision Tree: Numerical Example



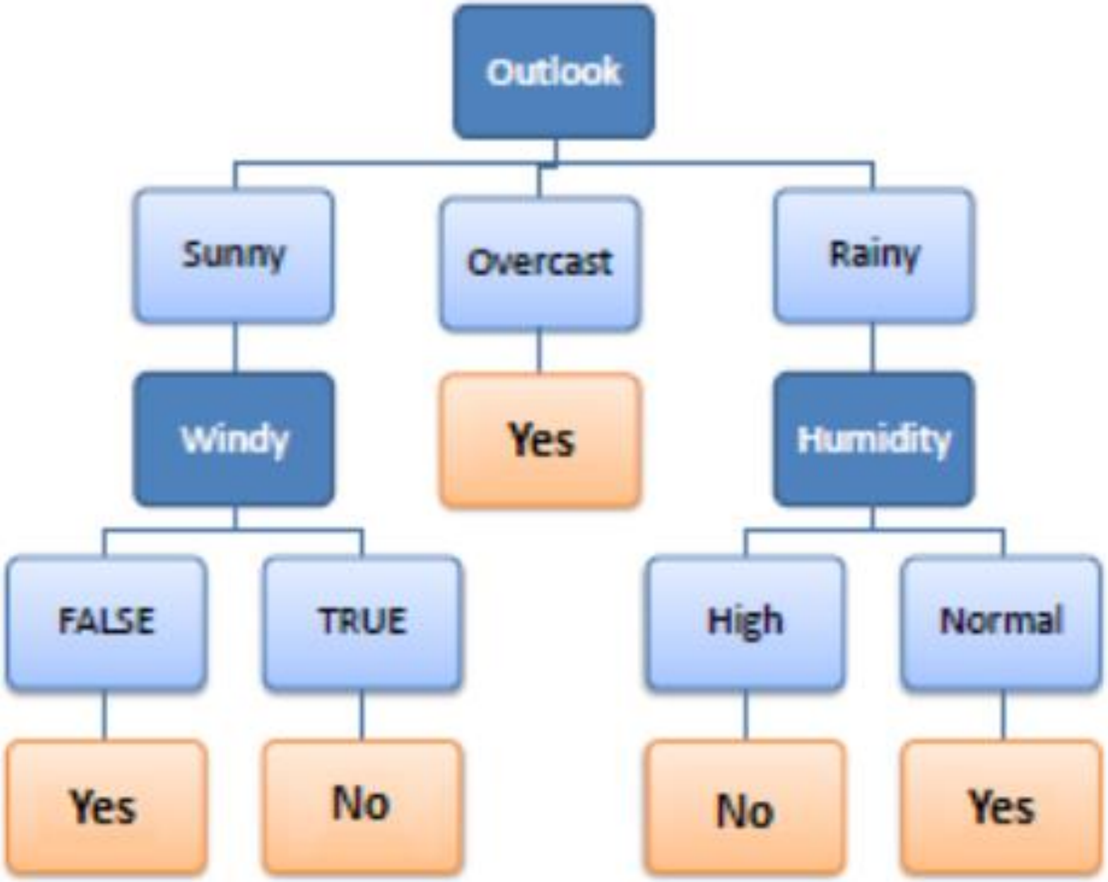
Decision Tree: Numerical Example

A branch with entropy more than 0 needs further splitting.



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Decision Tree: Numerical Example



Decision Tree

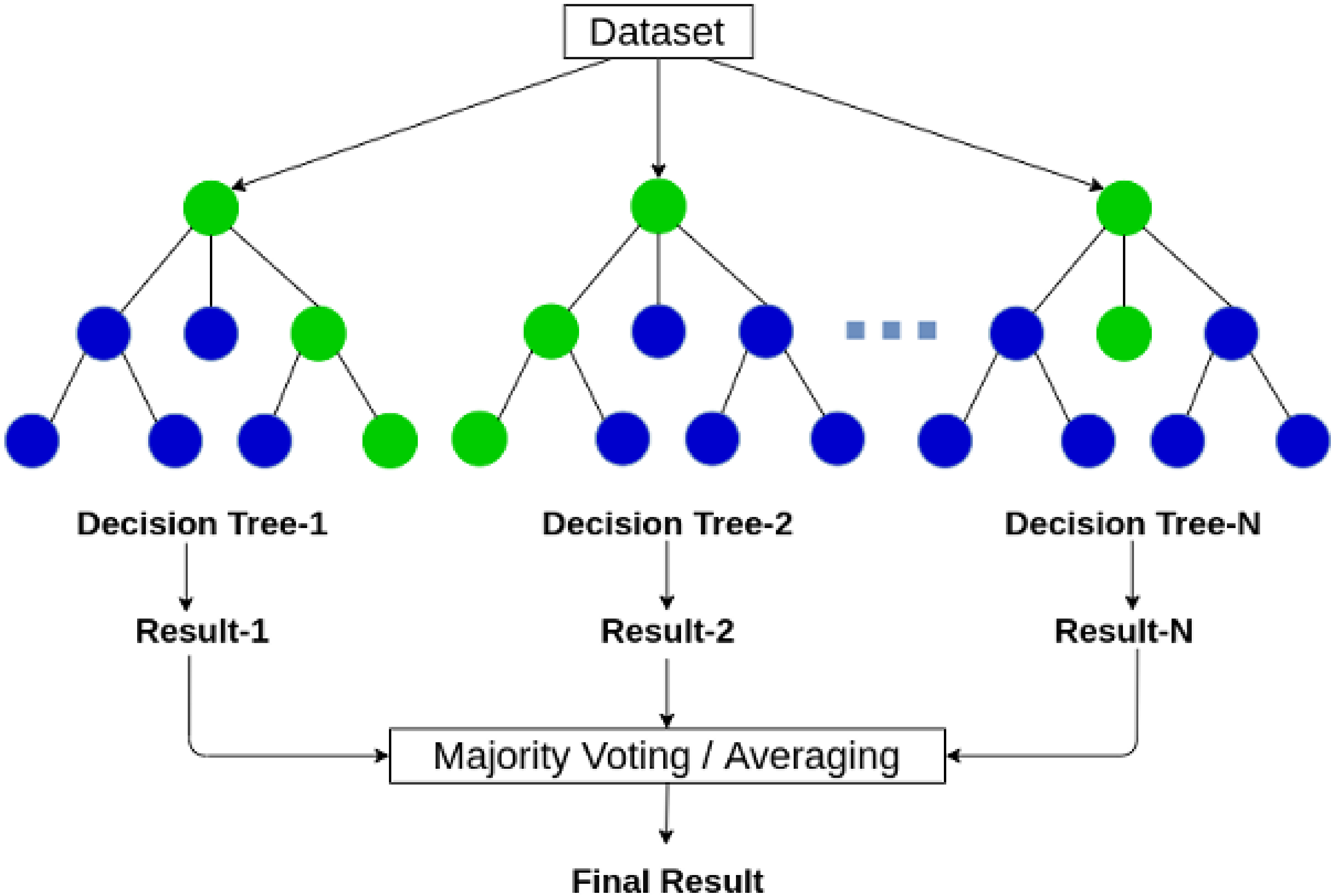
Implement Decision Tree

Random Forest

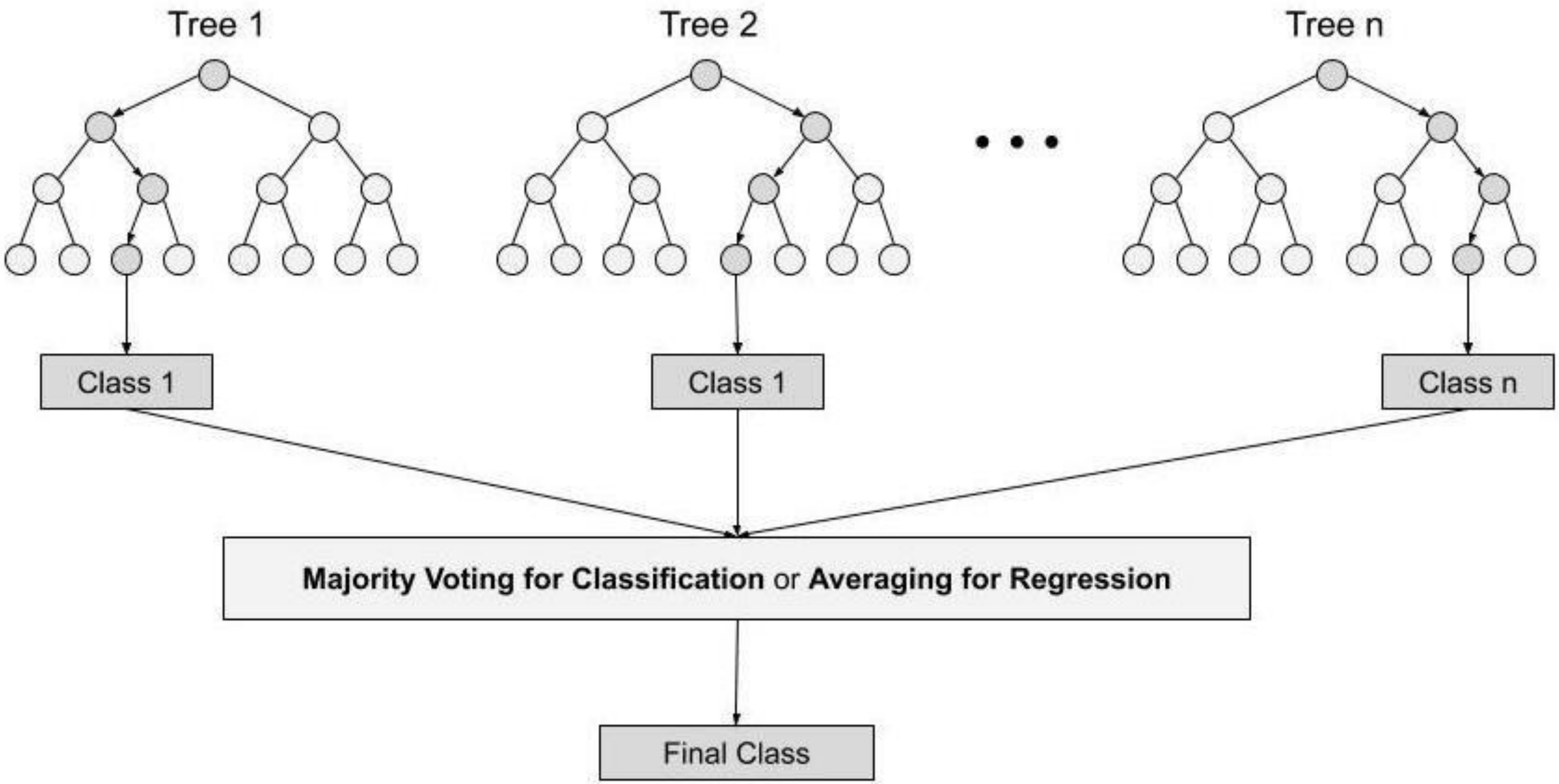
- Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems.

- It builds **decision trees on different samples** and takes their majority vote for classification and average in case of regression.

Random Forest



Random Forest



Decision Tree Exercise

Construct the decision tree of the data given below. The data represent different features of a file to check if it is infected with a virus or not. Use Entropy and Information Gain for attribute selection. You must go through all the steps to build the tree.

Permissions	Type	Size	Class
Read	Executable	Small	Infected
Write	Non-Executable	Large	Clean
Read	Executable	Medium	Infected
Read	Executable	Medium	Infected
Write	Executable	Medium	Clean
Read	Non-Executable	Large	Clean
Write	Executable	Small	Infected

Summary

- Feature Selection and Generation