Data Analytics

B.Tech III Year I Semester

by
S Kranthi Reddy, B.Tech, M.Tech
Assistant Professor
Department of Computer Science & Engineering
Blog: https://technoteskranthi.blogspot.com/
Email id: kranthi.vits@gmail.com

Introduction

Prerequisites:

- ✓ A course on “Database Management Systems”.
- ✓ Knowledge of probability and statistics.

Course Outcomes:

After completion of this course students will be able to

1. **Design Data Architecture u-1**
2. **Understand various Data Sources u-1**
3. Understand the impact of data analytics for business decisions and strategy **u-2,3,4**
4. Carry out data analysis/statistical analysis **u-2,3,4**
5. To carry out standard data visualization and formal inference procedures **u-5**

Introduction

Unit-1: Data Management

Unit-2: Data Analytics

Unit-3: Regression and Logistic Regression

Unit-4: Object Segmentation and Time Series Methods

Unit-5: Data Visualization

Text Books:

- Student's Handbook for Associate Analytics – II, III.
- Data Mining Concepts and Techniques, Han, Kamber, 3rd Edition, Morgan Kaufmann Publishers.

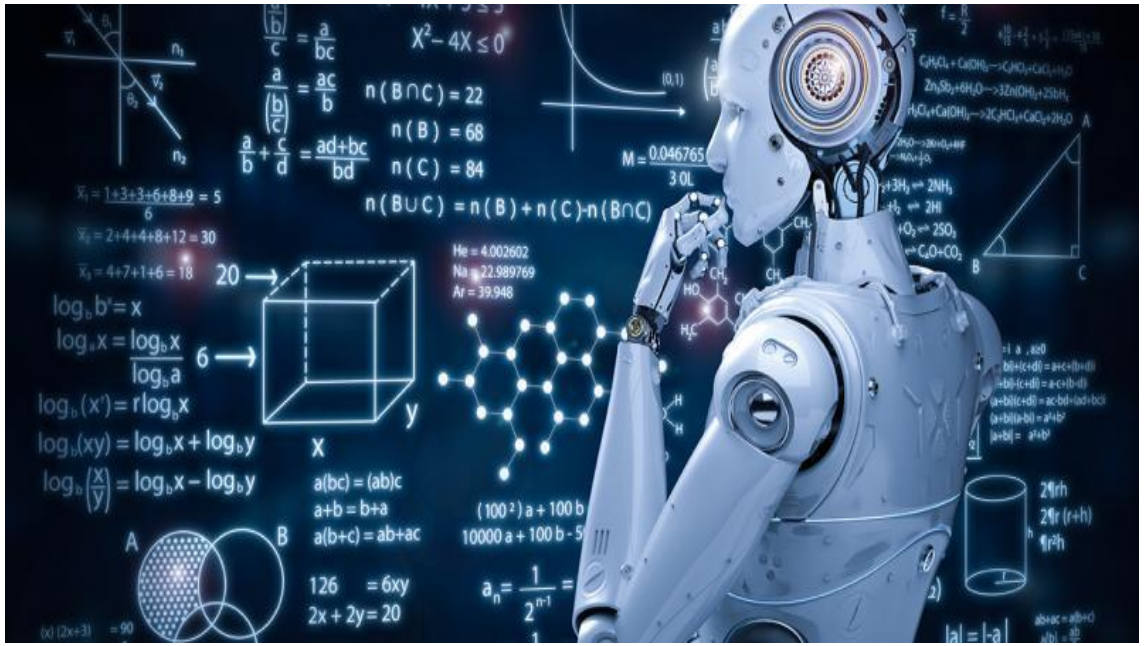
Unit-3

- ***Regression*** *– Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization and Model Building etc.*
- ***Logistic Regression:*** *Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.*

Unit-4

- ***Object Segmentation:*** *Regression Vs Segmentation –* ***Supervised and Unsupervised Learning****, Tree Building – Regression, Classification, Overfitting and Complexity, Multiple Decision Trees etc.*
- ***Time Series Methods:*** *Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height, Average Energy etc and Analyze for prediction*

What is Machine Learning?



- ***Machine Learning is a branch of Artificial Intelligence.***
- ***Machine Learning Provides Statistical tools to analyze the data and Predictive models.***
- ***Machine Learning concepts used to develop an application makes decision by itself.***

What is Machine Learning?

- Machine learning systems learns from historical data , builds the prediction models whenever it receives new data, predicts output for it.

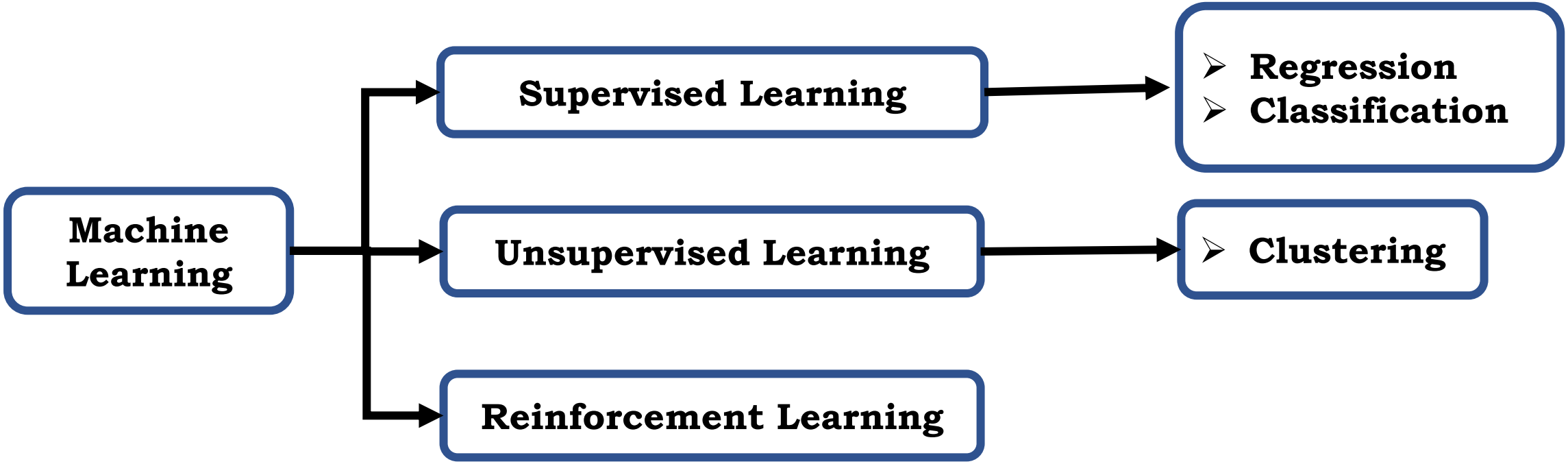


Figure: Classification of Machine Learning

Supervised & Unsupervised Learning

- **A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.** ***Mitchell's definition of ML***
- In simpler terms, a computer program is considered to be learning if it gets better at doing a certain type of task T as it gains more experience *E*, and this improvement is measured by a performance metric *P*. In other words, the program learns from its past experiences, and its performance in the tasks it's designed for improves over time.

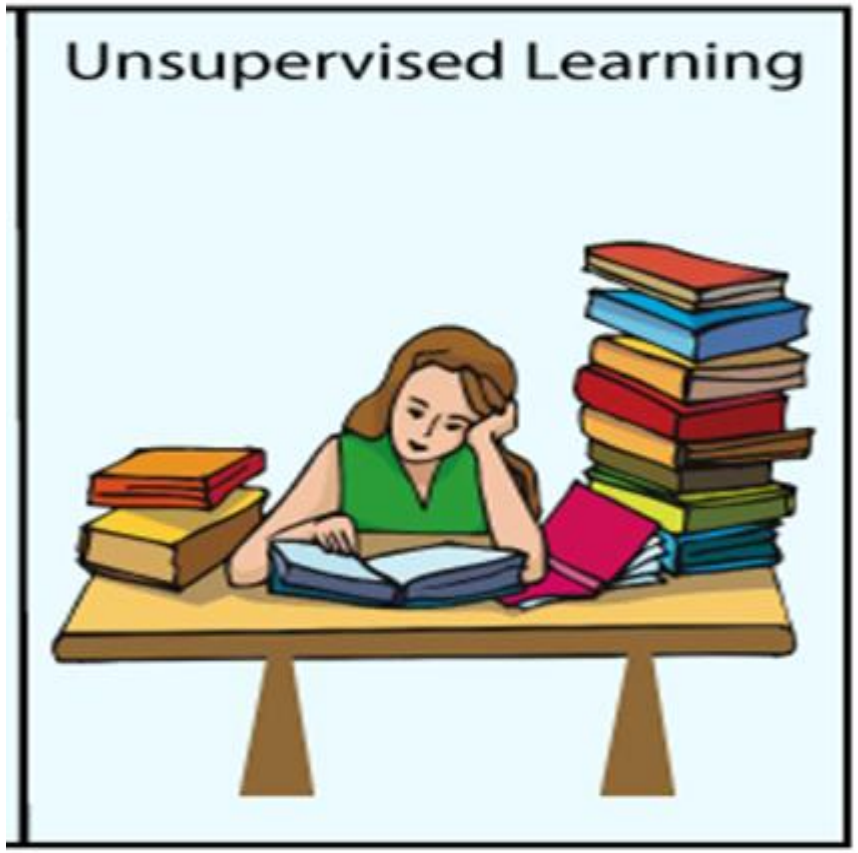
Supervised & Unsupervised Learning

- Regression and Classification algorithms are Supervised Learning algorithms. Both the algorithms are used for prediction in Machine learning and work with the labeled datasets. But the difference between both is how they are used for different machine learning problems.
- The main difference between Regression and Classification algorithms that Regression algorithms are used to **predict the continuous** values such as price, salary, age, etc. and Classification algorithms are used to **predict/Classify output** such as Male or Female, True or False, Spam or Not Spam, etc.
- Clustering is the task of dividing the unlabeled data or data points into different clusters such that similar data points fall in the same cluster than those which differ from the others.

Supervised & Unsupervised Learning

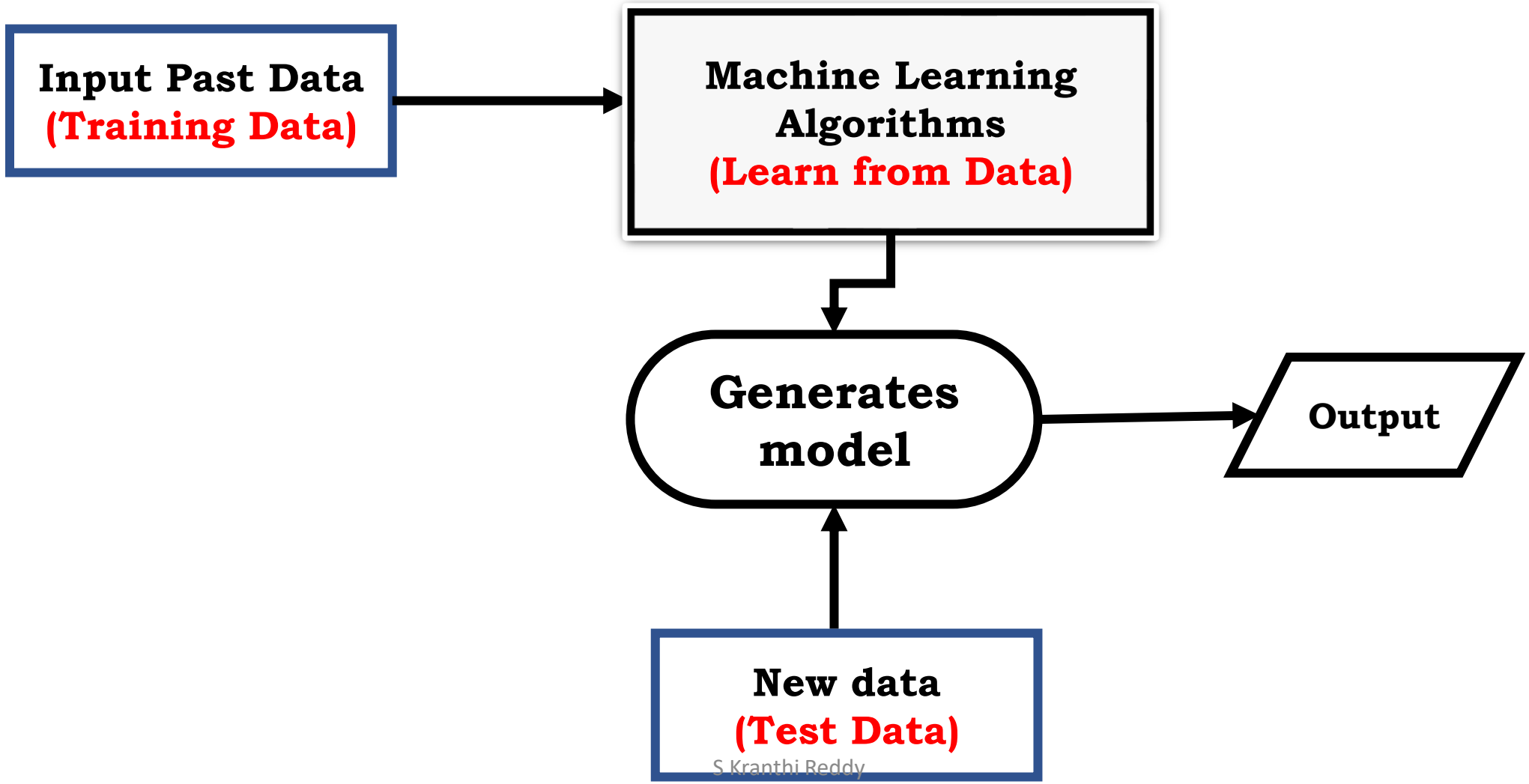
Supervised learning is a machine learning method in which models are trained using labeled data.

Supervised & Unsupervised Learning

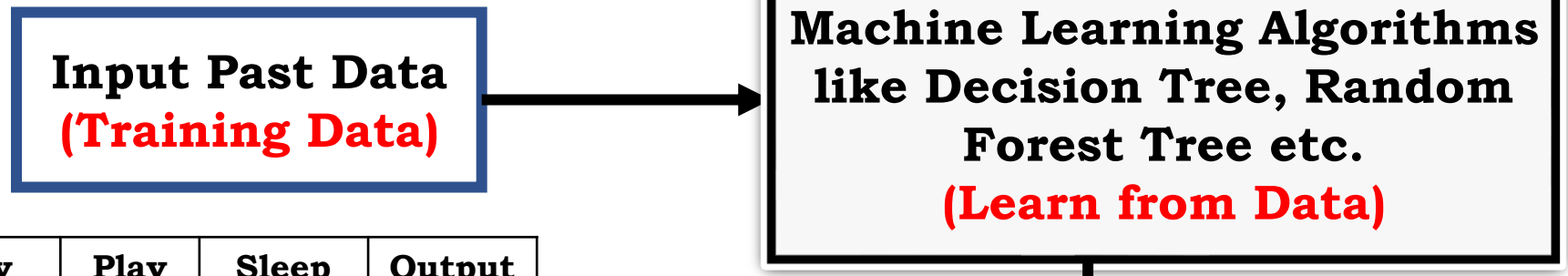


Unsupervised learning is another machine learning method in which patterns inferred from the unlabeled input data.

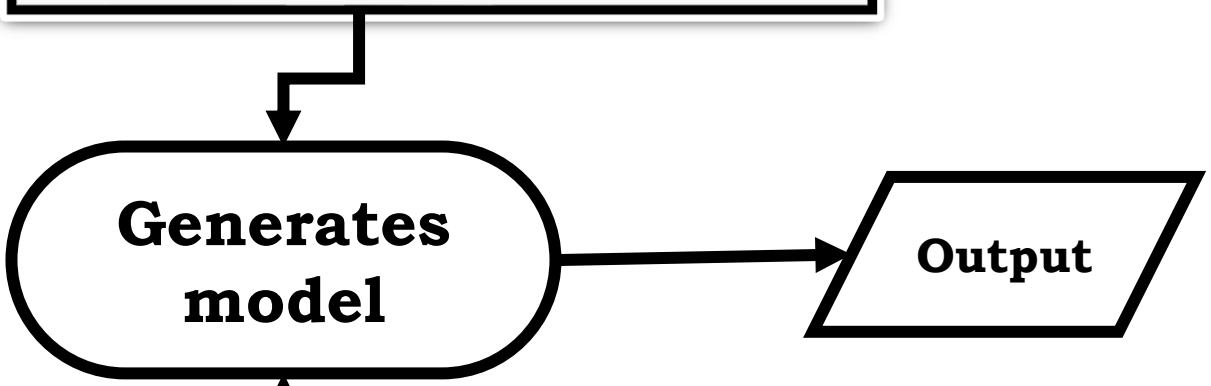
What is Machine Learning?



Supervised Machine Learning : Classification



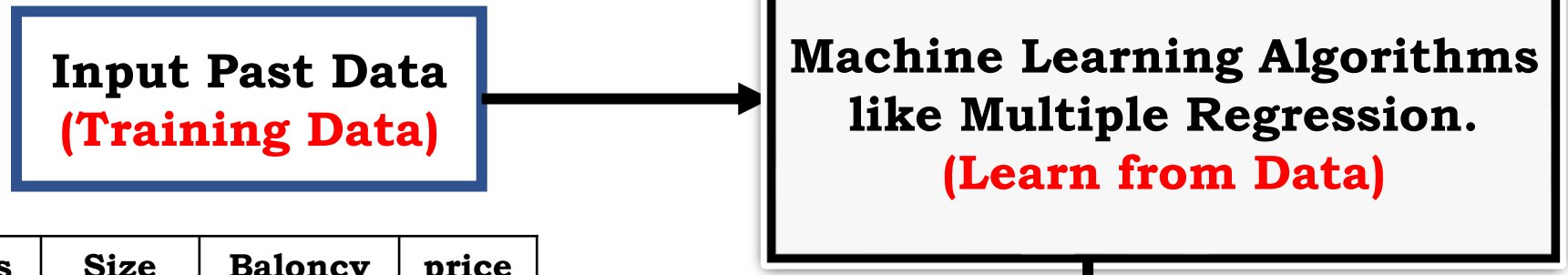
Study Time	Play Time	Sleep Time	Output
2	7	6	F
7	2	6	P
3	3	7	MayBe
-	-	-	-
-	-	-	-



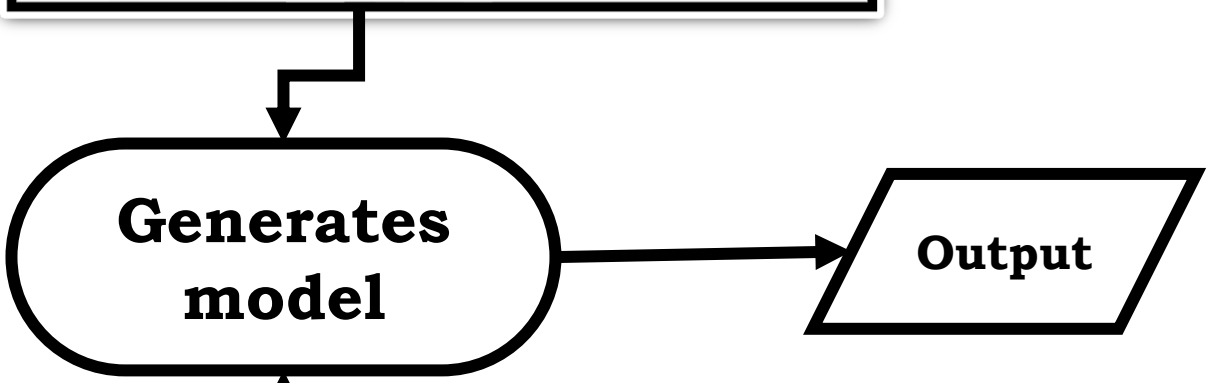
Output
P

Study Time	Play Time	Sleep Time
6	3	6

Supervised Machine Learning : Regression



Rooms	Size	Baloncy	price
4	2000	3	40
3	1500	2	35
2	800	2	20
-	-	-	-
-	-	-	-



Price
25

Rooms	Size	Baloncy
2	850	2

Unsupervised Machine Learning : Clustering

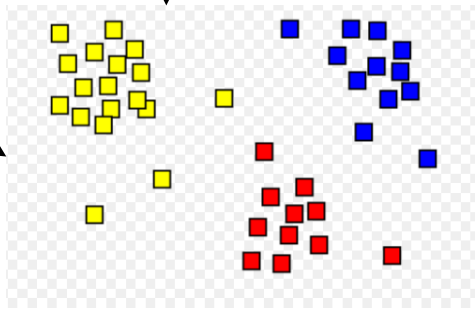
Input Past Data
(Training Data)

Study Time	Play Time	Sleep Time	Output
2	7	6	No Output
7	2	6	
3	3	7	
-	-	-	
-	-	-	

Machine Learning Algorithms like K-Means, Hierarchical etc
(Learn from Data)

Generates model

Groups or forms set of clusters



Study Time	Play Time	Sleep Time
6	3	6

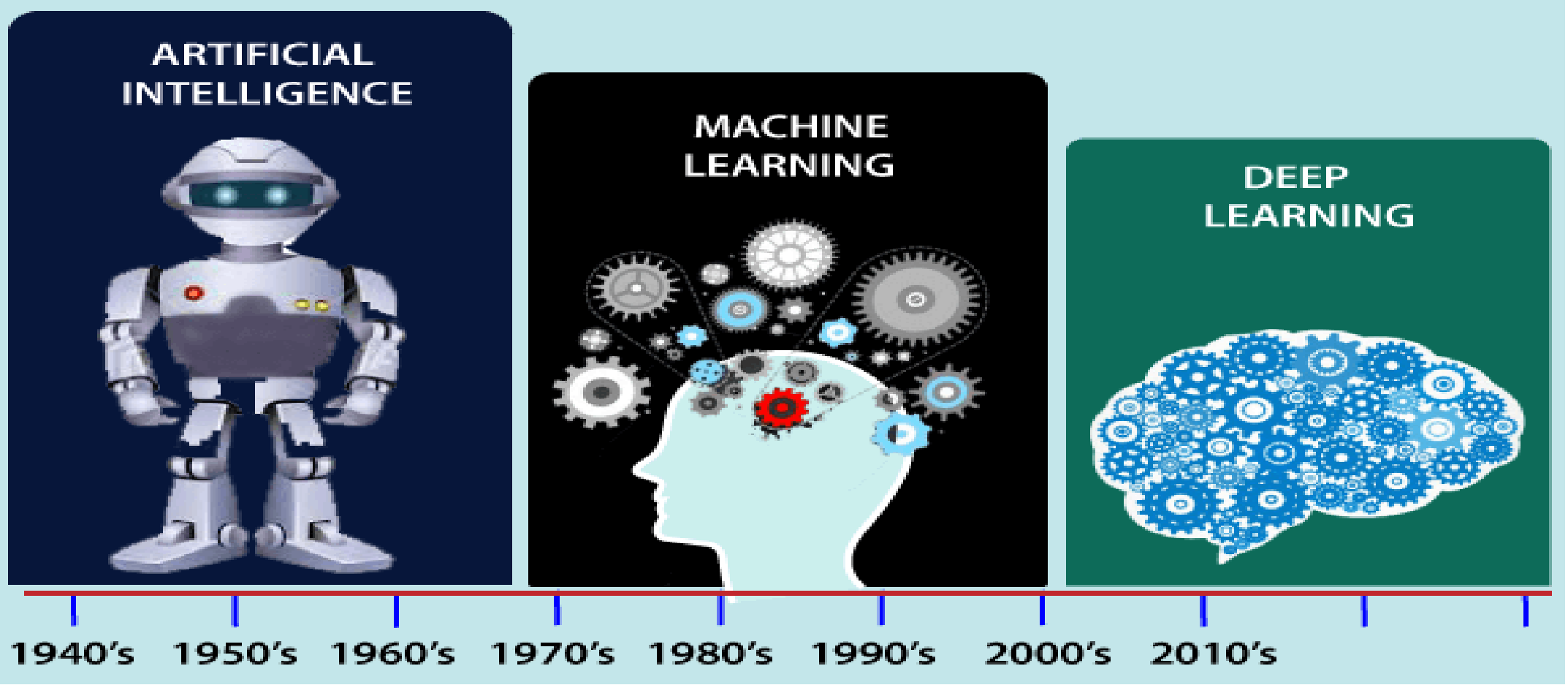
New data
(Test Data)

New input data will placed in one of cluster

Regression

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning can be categorized in Classification and Regression problems.	Unsupervised Learning can be classified in Clustering and Associations problems.
Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and Apriori algorithm.

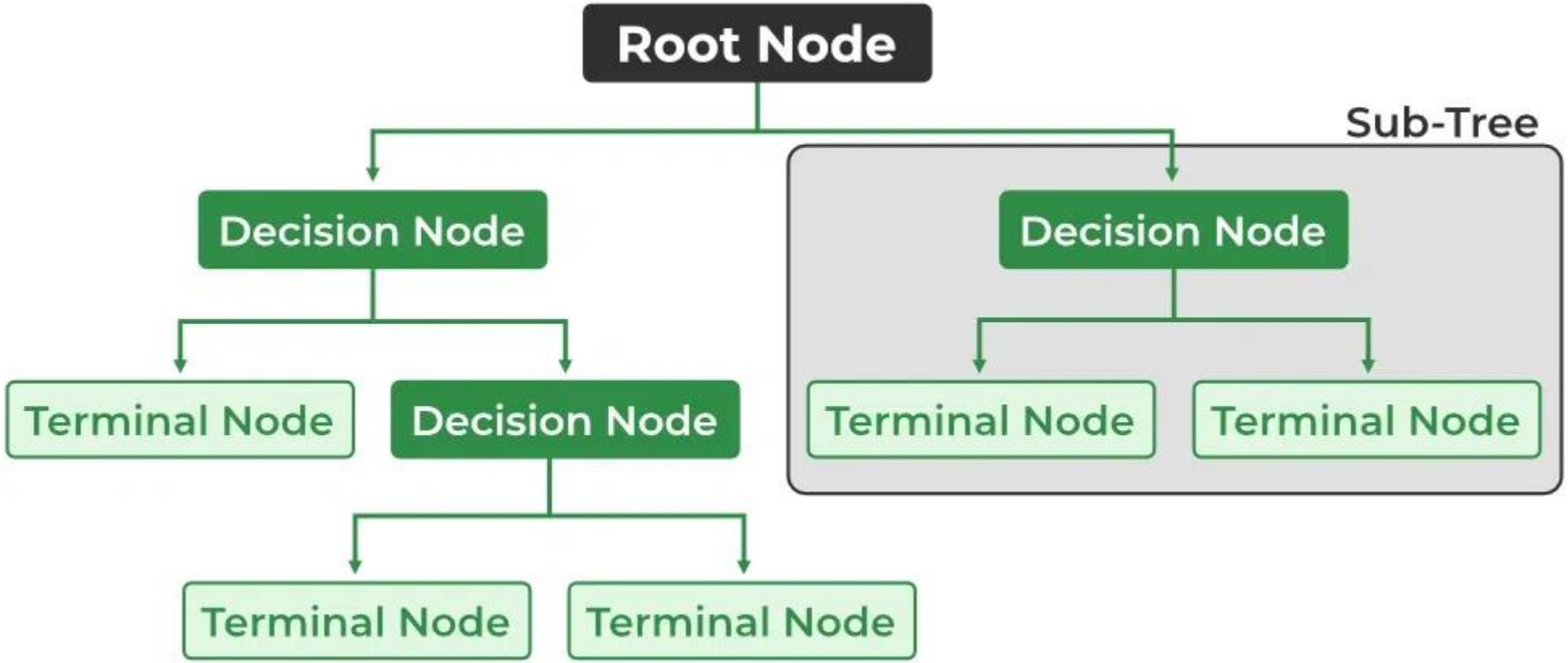
Machine Learning



Decision Tree

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- It is a tree-structured classifier, where **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 and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

Decision Tree



Decision Tree

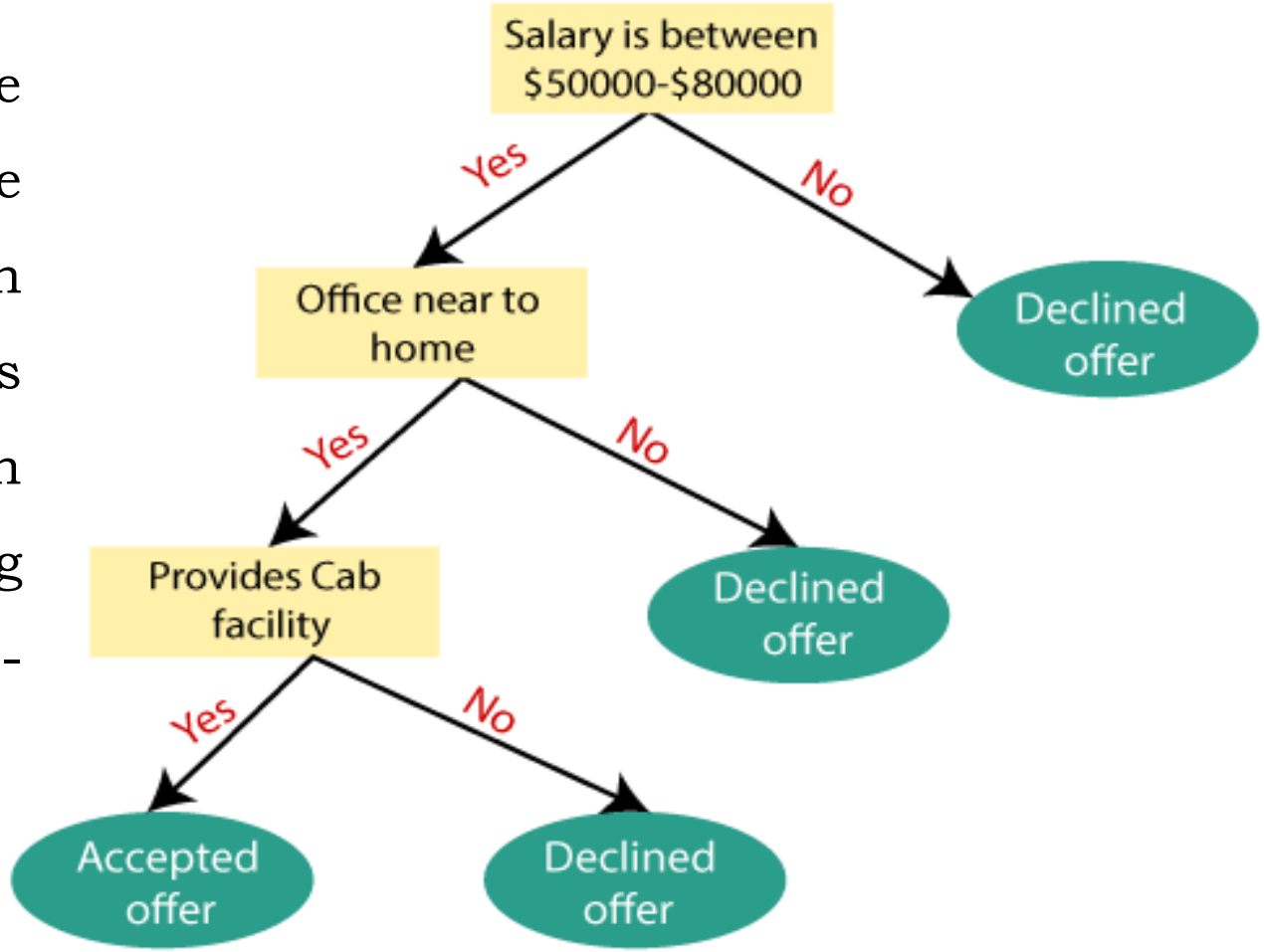
- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Decision/Internal Node:** A node that symbolizes a choice regarding an input feature. Branching off of internal nodes connects them to leaf nodes or other internal nodes.
- **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 subsection of the decision tree starts at an internal node and ends at the leaf nodes.

Decision 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

➢In decision tree prediction, start at the root, compare dataset attributes with node values, follow branches based on comparisons, and repeat until a leaf node is reached. The associated leaf class is then the predicted outcome, efficiently capturing data patterns through hierarchical decision-making.



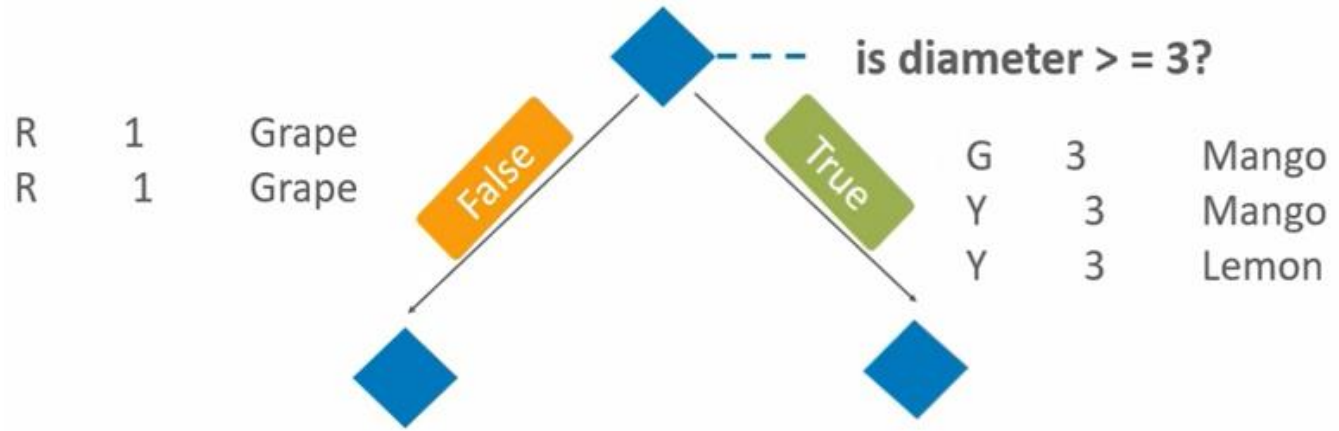
How Decision Tree Alogorithm works

- ✓ 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.

Decision Tree

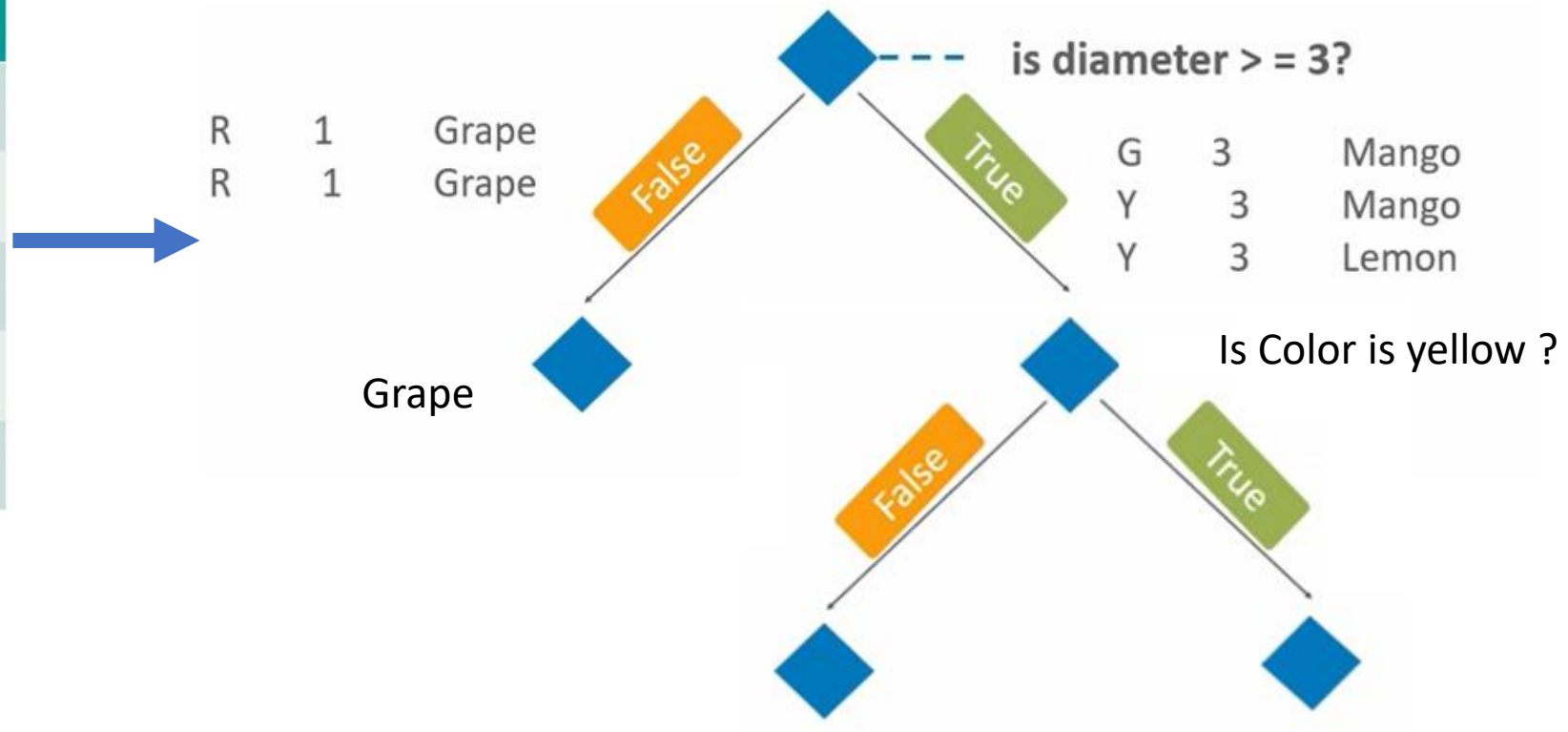
Colour	Diameter	Label
Green	3	Mango
Yellow	3	Mango
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Decision is constructed by asking a question about colour and diameter

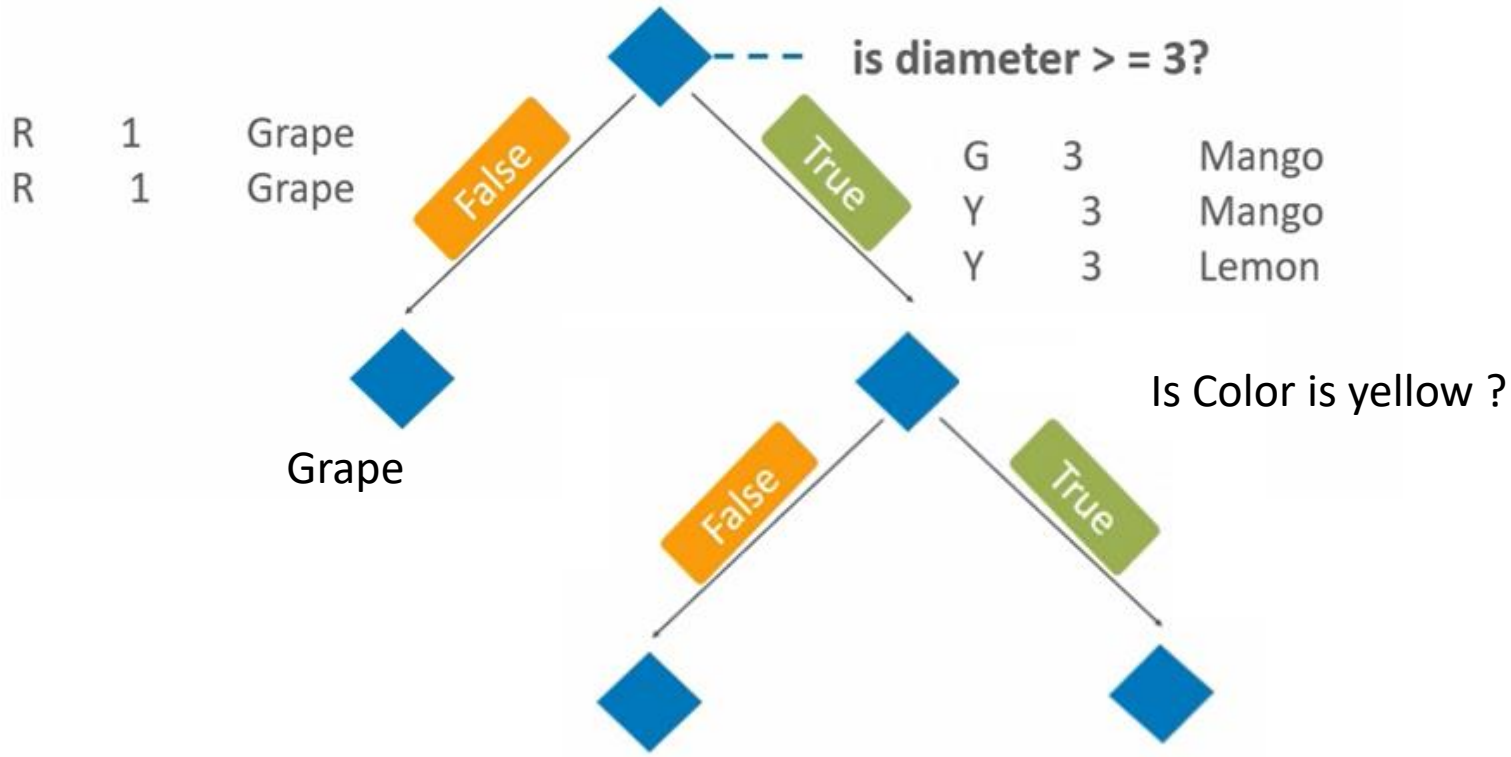


Decision Tree

Colour	Diameter	Label
Green	3	Mango
Yellow	3	Mango
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon



Decision Tree



R	1	Grape
R	1	Grape
G	3	Mango
Y	3	Mango
Y	3	Lemon

Decision Tree

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 and Entropy
- ✓ Gini Index

Decision Tree

- Entropy is a measure of impurity or disorder in a dataset. In the context of decision trees and classification, entropy is used to quantify the uncertainty associated with a set of labels.

$$Entropy(Class) = -\frac{P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

- Example: Consider a binary classification problem with a dataset of emails labeled as either spam (S) or not spam (NS). If all emails are spam, or if all emails are not spam, the entropy is low (0). If the dataset is evenly split between spam and not spam, the entropy is at its maximum.

Decision Tree

- Information Gain measures the effectiveness of an attribute in reducing the uncertainty (entropy) about the classification of the data. It helps the decision tree algorithm decide which attribute to split on at each node.
- Example: Continuing with the email classification example, if you have additional attributes like "Contains Link" or "Subject Length," Information Gain is used to evaluate which attribute provides the most useful information for classifying emails as spam or not spam.

Decision Tree

For any Attribute,

$$InformationGain(Attribute)$$

$$I(p_i, n_i) = -\frac{p_i}{p_i + n_i}\log_2\left(\frac{p_i}{p_i + n_i}\right) - \frac{n_i}{p_i + n_i}\log_2\left(\frac{n_i}{p_i + n_i}\right)$$

Entropy of an Attribute is:

$$Entropy(Attribute) = \frac{\sum(p_i + n_i)}{P + N} I(p_i + n_i)$$

$$Gain = Entropy(Class) - Entropy(Attribute)$$

Construction of Decision Tree

1. Calculate Entropy impurity or disorder of Class attribute using following formula:

$$Entropy(Class) = -\frac{P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

2. Calculate Gain for every attribute in dataset, the attribute with highest is considered as root node. Now exclude that root repeat step 1 & 2 continuously till tree is formed.

- To calculate Gain, Entropy of attribute is required, To find Entropy of attribute, information gain w.r.t categorical values in attribute is required.

For any Attribute,

$$InformationGain(Attribute)$$

$$I(p_i, n_i) = -\frac{p_i}{p_i+n_i}\log_2\left(\frac{p_i}{p_i+n_i}\right) - \frac{n_i}{p_i+n_i}\log_2\left(\frac{n_i}{p_i+n_i}\right)$$

Entropy of an Attribute is:

$$Entropy(Attribute) = \frac{\sum(p_i+n_i)}{P+N} I(p_i+n_i)$$

$$Gain = Entropy(Class) - Entropy(Attribute)$$

Construction of Decision Tree

- Consider the dataset which contains four columns/attributes, three are independent and one is dependent i.e., Profit. Which is call class attribute or outcome attribute. First step in construction of decision is calculate Entropy of class.

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

- Class attributes contain two categories i.e, Down and Up. let consider P is positive class i.e, Up and N is negative class i.e, Down.

P	5	No. of Up
N	5	No. of Down

1. Calculate Entropy of Class attribute using following formula:

$$-P/(P+N) * \log(P/(P+N)) - N/(P+N)*\log(N/(P+N))$$

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

- Class attributes contain two categories i.e, Down and Up. let consider P is positive class i.e, Up and N is negative class i.e, Down.

P	5	No. of Up
N	5	No. of Down

1. Calculate Entropy of Class attribute using following formula:

$$-P/(P+N) * \log(P/(P+N)) - N/(P+N)*\log(N/(P+N))$$

Entropy of Class $= -5/10 * \log2(5/10) -5/10*\log2(5/10)$

$= 1$

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

2. For each attribute we have to calculate information gain by using following formula:

$$I(Pi, Ni) = -Pi/(Pi+Ni) * \log2(Pi/(Pi+Ni)) \; Ni/(Pi+Ni)*\log2(Ni/(Pi+Ni))$$

3. Entropy of each attribute is calculated using following formula:

$$\frac{\sum Pi + Ni}{P+N} * (I(Pi, Ni))$$

Gain = Entropy of Class – Entropy of Attribute.

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

The age column contains three categorical values i.e, old, mid and new. Tabulate the count of categorical values w.r.t to outcome attribute as follwos.

Age	Pi(Up)	Ni(Down)	I(Pi, Ni)
Old	0	3	
Mid	2	2	
New	0	3	

The information gain for categorical value can calculated using formula or else if any one value is zero either (pi or ni = 0) then gain of that categorical value is 0. if both are same values (Pi==x and Ni==X) then information gain 1.

Age	Pi(Up)	Ni(Down)	I(Pi, Ni)
Old	0	3	0
Mid	2	2	1
New	0	3	0

Entropy(age)= ((0+3)/(5+5))*0 + ((2+2)/(5+5))*8 + ((0+3)/(5+5))*0
Entropy(age)= 4/10 = 0.4
Gain of age = Entropy of class – Entropy of age = 1- 0.4 = 0.6

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

The competition column contains two categorical values i.e, yes and no. Tabulate the count of categorical values w.r.t to outcome attribute as follwos.

Competition	Pi(Up)	Ni(Down)	I(Pi, Ni)
Yes	1	3	
No	4	2	

The table contains different values, in this case information gained is calculated. Now Substitute Pi and Ni in information gain, find the value IG for Pi=1, Ni=3 and Ni=4,Ni=2

I(Pi, Ni) = -Pi/(Pi+Ni) * log2(Pi/(Pi+Ni)) - Ni/(Pi+Ni)*log2(Ni/(Pi+Ni))

I(1,3) = (-1/4)*log2(-1/4) – (3/4)*log2((3/4) = 0.81127

I(4,2) = (-4/6)*log2(4/6) –(2/6)*log2(2/6) = 0.9182

Competition	Pi(Up)	Ni(Down)	I(Pi, Ni)
Yes	1	3	0.81127
No	4	2	0.9182

E(Comp)= ((1+3)/(5+5))*0.81127 + ((4+2)/(5+5))*0.92182

E(Comp)= 0.875

Gain of competition = Entropy of class – Entropy of attribute

Gain = 1- 0.875 = 0.12415

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

The competition column contains two categorical values i.e, yes and no. Tabulate the count of categorical values w.r.t to outcome attribute as follwos.

Competition	Pi(Up)	Ni(Down)	I(Pi, Ni)
Yes	1	3	
No	4	2	

The table contains different values, in this case information gained is calculated. Now Substitute Pi and Ni in information gain, find the value IG for Pi=1, Ni=3 and Ni=4,Ni=2

$I(Pi, Ni) = -Pi/(Pi+Ni) * \log2(Pi/(Pi+Ni)) - Ni/(Pi+Ni)*\log2(Ni/(Pi+Ni))$

$I(1,3) = (-1/4)*\log2(-1/4) – (3/4)*\log2((3/4) = 0.81127$

$I(4,2) = (-4/6)*\log2(4/6) –(2/6)*\log2(2/6) = 0.9182$

Competition	Pi(Up)	Ni(Down)	I(Pi, Ni)
Yes	1	3	0.81127
No	4	2	0.9182

$E(Comp)= ((1+3)/(5+5))*0.81127 + ((4+2)/(5+5))*0.92182$

$E(Comp)= 0.875$

Gain = Entropy of class – Entropy of attribute = 1- 0.875 = 0.12415

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

The type column contains two categorical values i.e,software and hardware. Tabulate the count of categorical values w.r.t to outcome attribute as follows.

Type	Pi(Up)	Ni(Down)	I(Pi, Ni)
Software	3	3	
Hardware	2	2	

Competition	Pi(Up)	Ni(Down)	I(Pi, Ni)
Yes	3	3	1
No	2	2	1

E(Type)= ((3+3)/(5+5))*1 + ((2+2)/(5+5))*1

E(Type)= 1

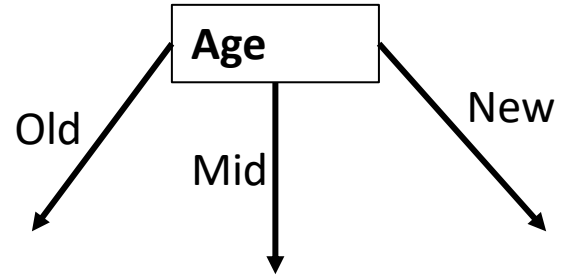
Gain of Type = Entropy of class – Entropy of attribute = 1- 1 = 0

Type	Gain
Age	0.6
Competition	0.124
Type	0

Type	Gain
Age	**0.6**
Competition	0.124
Type	0

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	**Software**	**Down**
Old	No	**Software**	**Down**
Old	No	Hardware	Down
Mid	Yes	**Software**	**Down**
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	**Software**	**Up**
New	Yes	**Software**	**Up**
New	No	Hardware	Up
New	No	**Software**	**Up**



Age	Competition	Type	Profit
Old	Yes	**Software**	**Down**
Old	No	**Software**	**Down**
Old	No	Hardware	Down

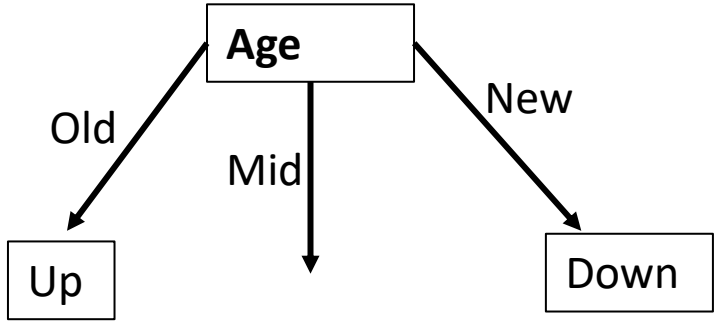
Age	Competition	Type	Profit
New	Yes	**Software**	**Up**
New	No	Hardware	Up
New	No	**Software**	**Up**

Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

Type	Gain
Age	0.6
Competition	0.124
Type	0

Type	Gain
Age	**0.6**
Competition	0.124
Type	0

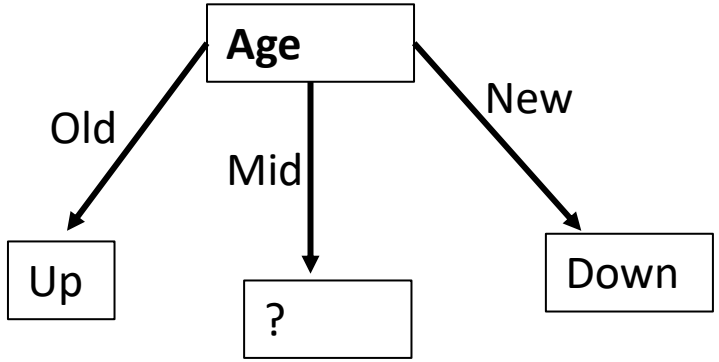


Construction of Decision Tree

Age	Competition	Type	Profit
Old	Yes	Software	Down
Old	No	Software	Down
Old	No	Hardware	Down
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up
New	Yes	Software	Up
New	No	Hardware	Up
New	No	Software	Up

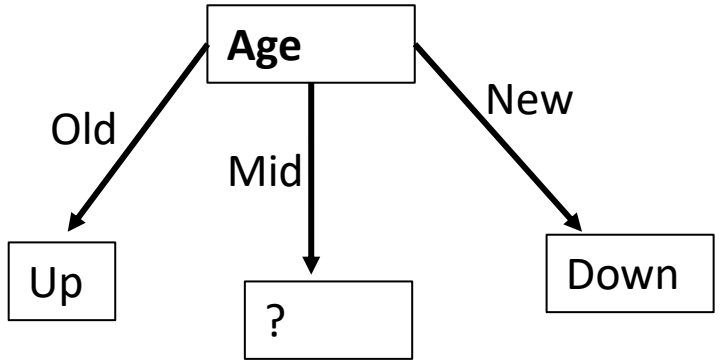
Type	Gain
Age	0.6
Competition	0.124
Type	0

Type	Gain
Age	**0.6**
Competition	0.124
Type	0



Construction of Decision Tree

Age	Competition	Type	Profit
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up



1. Calculate Entropy of Class attribute using following formula:

$$-P/(P+N) * \log2(P/(P+N)) - N/(P+N)*\log2(N/(P+N))$$

Entropy of Class $= -2/4 * \log2(2/4) -2/4*\log2(2/4)$

$= 1$

Competition	Pi	Ni	I(Pi, Ni)
Yes	0	2	
No	2	0	

Competition	Pi	Ni	I(Pi, Ni)
Yes	0	2	0
No	2	0	0

E(Comp)= ((0+2)/(4))*0 + ((0+2)/(4))*0

E(comp)= 0

Gain = Entropy of class – Entropy of attribute

Gain = 1- 0 =1

Construction of Decision Tree

Age	Competition	Type	Profit
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up

Competition	Pi	Ni	I(Pi, Ni)
Yes	0	2	
No	2	0	

Competition	Pi	Ni	I(Pi, Ni)
Yes	0	2	0
No	2	0	0

E(Comp)= ((0+2)/(4))*0 + ((0+2)/(4))*0
E(comp)= 0
Gain = Entropy of class – Entropy of attribute
Gain = 1- 0 =1

Type	Pi	Ni	I(Pi, Ni)
Software	1	1	
Hardware	1	1	

Type	Pi	Ni	I(Pi, Ni)
Software	1	1	1
Hardware	1	1	1

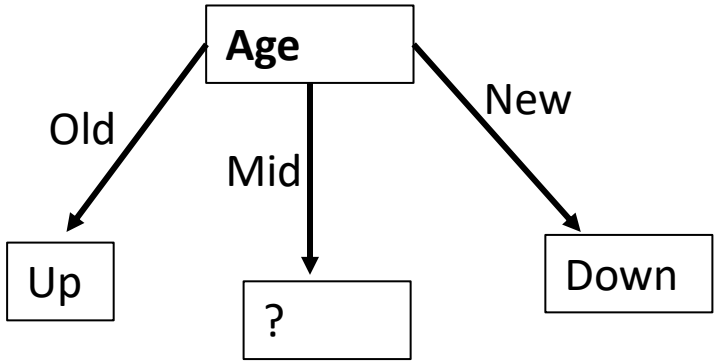
E(Type)= ((1+1)/(4))*1 + ((1+1)/(4))*1
E(comp)= 1
Gain = Entropy of class – Entropy of attribute
Gain = 1- 1 = 0

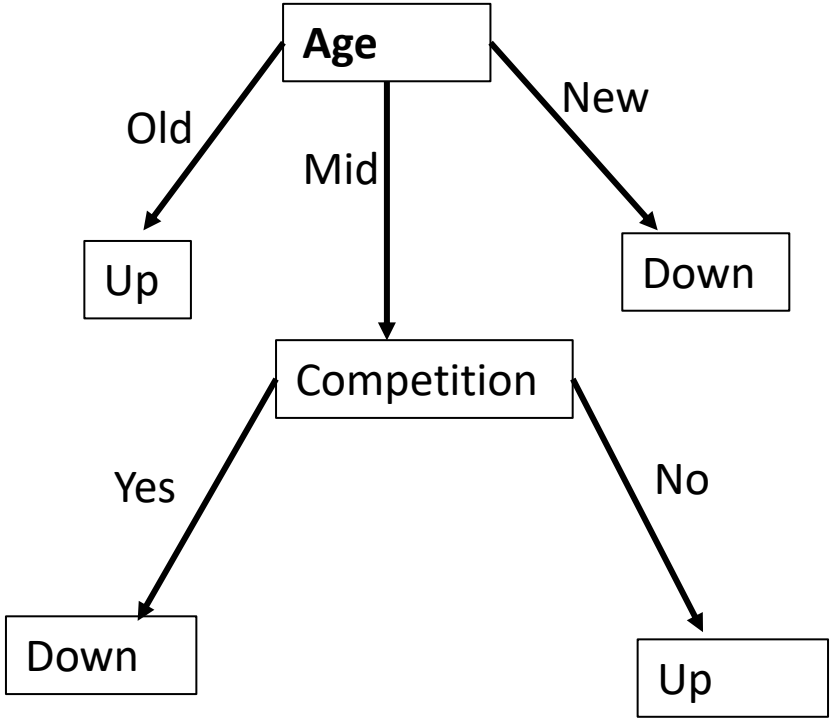
Type	Gain
Competition	1
Type	0

Construction of Decision Tree

Age	Competition	Type	Profit
Mid	Yes	Software	Down
Mid	Yes	Hardware	Down
Mid	No	Hardware	Up
Mid	No	Software	Up

Type	Gain
Competition	**1**
Type	0





Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$

$S_{Sunny} \leftarrow [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$

$S_{Overcast} \leftarrow [4+, 0-]$ $\quad Entropy(S_{Overcast}) = -\frac{4}{4}log_2\frac{4}{4} - \frac{0}{4}log_2\frac{0}{4} = 0$

$S_{Rain} \leftarrow [3+, 2-]$ $\quad Entropy(S_{Rain}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$

$$Gain\,(S, Outlook) = Entropy(S) - \sum_{v \in \{Sunny, Overcast, Rain\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Outlook)$$
$$= Entropy(S) - \frac{5}{14}Entropy(S_{Sunny}) - \frac{4}{14}Entropy(S_{Overcast})$$
$$- \frac{5}{14}Entropy(S_{Rain})$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14}0.971 - \frac{4}{14}\,0 - \frac{5}{14}0.971 = 0.2464$$

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes

No of no in Play Tennis: 4 (-ve class)

No of yes in Play Tennis: 9 (+ve class)

First calculate Entropy of class variable.

-P/(P+N) * log2(P/(P+N)) – N/(P+N)*log2(N/(P+N))

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$

For each attribute Gain has to be calculated for

Information gain is required and Entropy is required.

I(Pi, Ni) = -Pi/(Pi+Ni) * log2(Pi/(Pi+Ni)) - Ni/(Pi+Ni)*log2(Ni/(Pi+Ni))

Entropy of each attribute is calculated using following formula:

$$\frac{\sum Pi + Ni}{P+N} * (I(Pi, Ni))$$

Gain of attribute = Entropy of Class – Entropy of Attribute.

Comparative Study of three Classification Algorithms

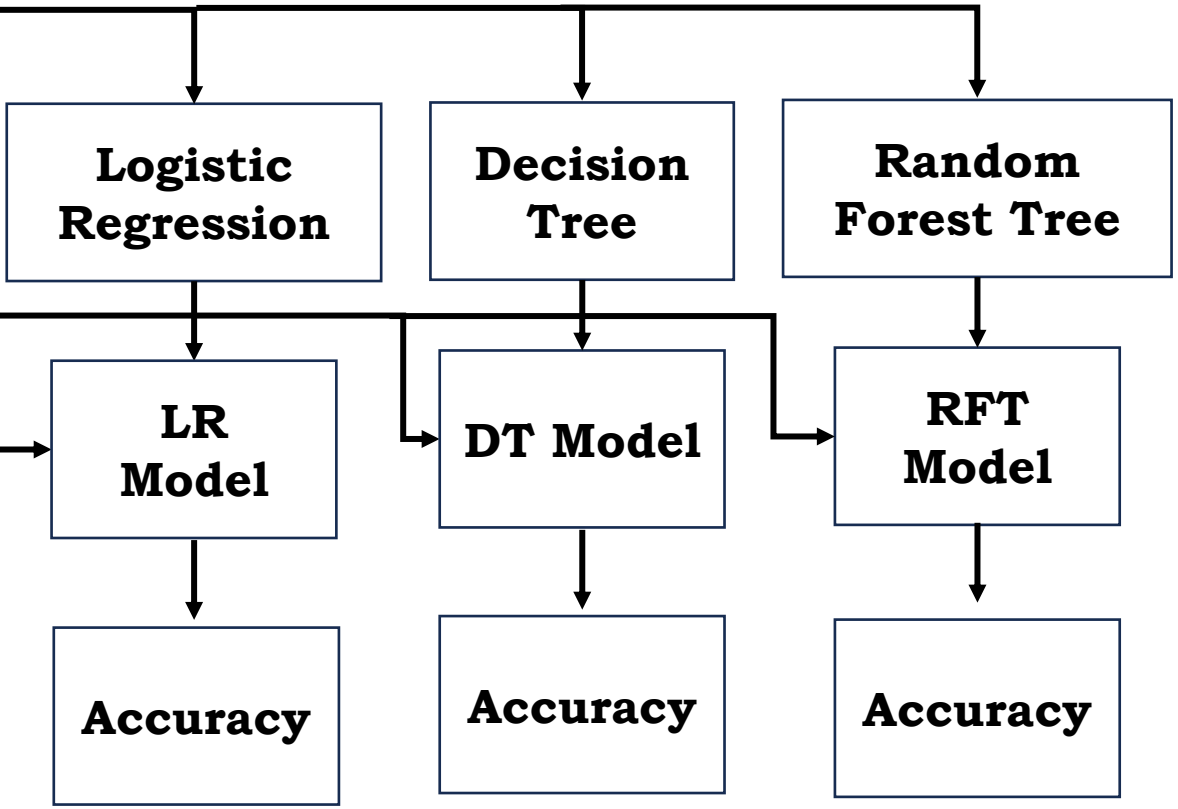
d3

	admit	gre	gpa	rank
1	0	380	3.61	3
2	1	660	3.67	3
3	1	800	4.00	1
4	1	640	3.19	4
5	0	520	2.93	4
6	1	760	3.00	2
7	1	560	2.98	1
9	1	540	3.39	3
10	0	700	3.92	2
11	0	800	4.00	4
12	0	440	3.22	1
13	1	760	4.00	1

```
> nrow(d3)
[1] 400
```

```
> nrow(train)
[1] 330
```

```
> nrow(test)
[1] 70
```



Finding Optimal Model based on Accuracy which can be used for prediction

Overfitting

- When a model ***performs very well for training data but has poor performance with test data (new data), it is known as overfitting.***
- In this case, the machine learning model learns the details and noise in the training data such that it negatively affects the performance of the model on test data.
- Overfitting can happen due to low bias and high variance.

Underfitting

- When a model ***has not learned the patterns in the training data well and is unable to generalize well on the new data,*** it is known as underfitting.
- An underfit model has poor performance on the training data and will result in unreliable predictions.
- Underfitting occurs due to high bias and low variance.

Overfitting

- **Reasons for Overfitting:**
  - ✓ Data used for training is not cleaned and contains noise (garbage values) in it.
  - ✓ The model has a high variance.
  - ✓ The size of the training dataset used is not enough.
  - ✓ The model is too complex.

- **Ways to Tackle Overfitting**
  - ✓ Using K-fold cross-validation
  - ✓ Training model with sufficient data
  - ✓ Adopting ensembling techniques

Underfitting

- **Reasons for Underfitting:**
  - ✓ Data used for training is not cleaned and contains noise (garbage values) in it
  - ✓ The model has a high bias
  - ✓ The size of the training dataset used is not enough
  - ✓ The model is too simple

- **Ways to Tackle Underfitting**
  - ✓ Increase the number of features in the dataset
  - ✓ Increase model complexity
  - ✓ Reduce noise in the data
  - ✓ Increase the duration of training the data

Overfitting & Underfitting

A

Not interested in learning

Class test ~50%
Test ~47%

Under-fit/ biased learning

B

Memorizing the lessons

Class test ~98%
Test ~69%

Over-fit/ Memorizing

C

Conceptual Learning

Class test ~92%
Test ~89%

Best-fit

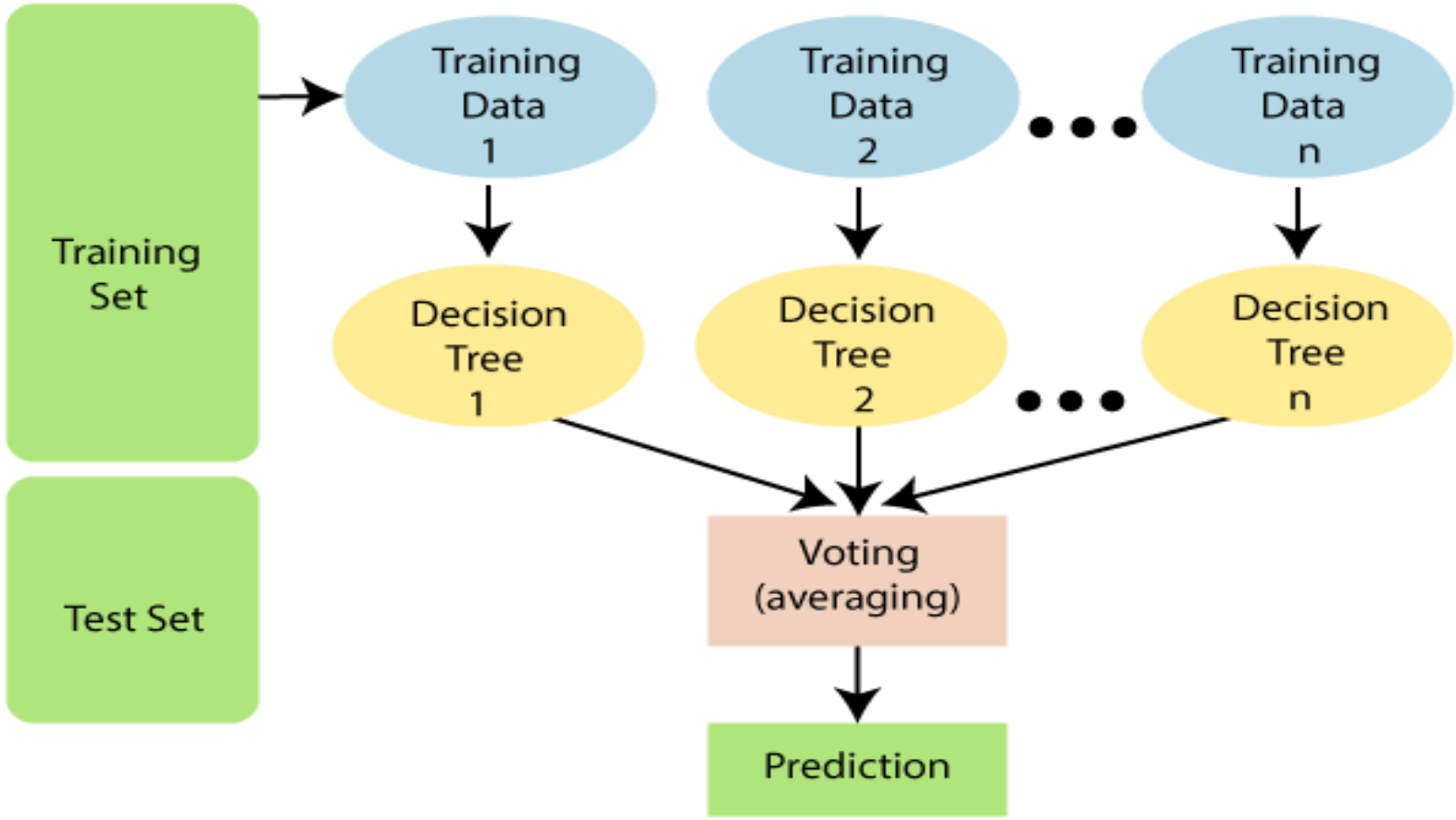
Multiple Decision Trees

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.
- It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

Multiple Decision Trees

- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
- Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Multiple Decision Trees



working of the Random Forest algorithm

Multiple Decision Trees

- **Step-1:** Select random K data points from the training set.
- **Step-2:** Build the decision trees associated with the selected data points (Subsets).
- **Step-3:** Choose the number N for decision trees that you want to build.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Segmentation

- Segmentation and regression are both important techniques in various fields, but they have distinct purposes and applications. The key differences:
- Segmentation: Categorizes data points into groups or clusters based on their shared characteristics. The output is a discrete label assigned to each data point, indicating its group membership. Think of it like sorting fruits into different baskets based on their color (red, green, yellow).
- Regression: Predicts continuous numerical values for a given input. The output is a real number representing the predicted value for a specific data point. Imagine estimating the price of a house based on its size, location, and other features.