

5CS037-Concepts and Technologies of AI
Lecture-10

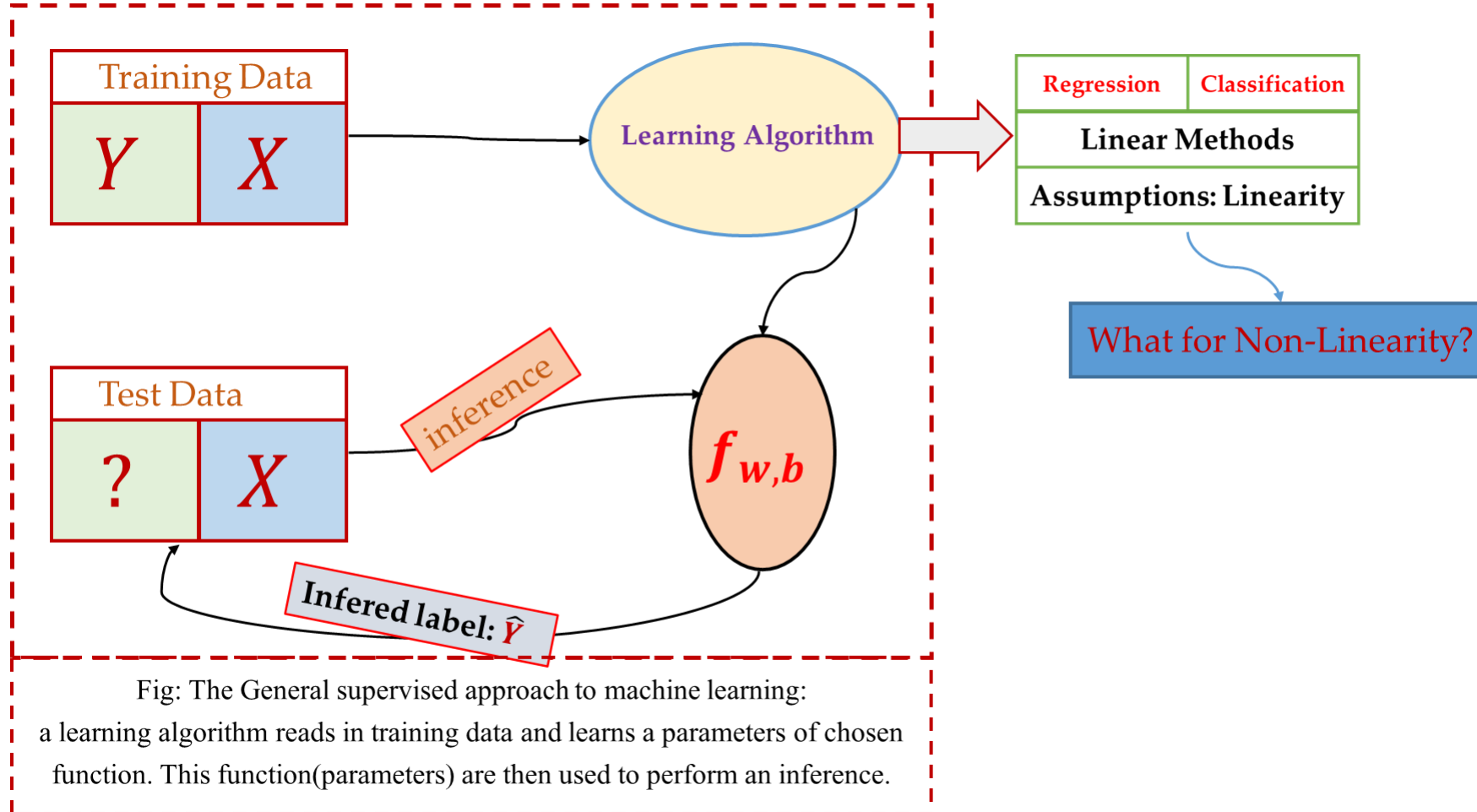
Supervised Machine Learning

Decision Tree and Ensemble Methods.

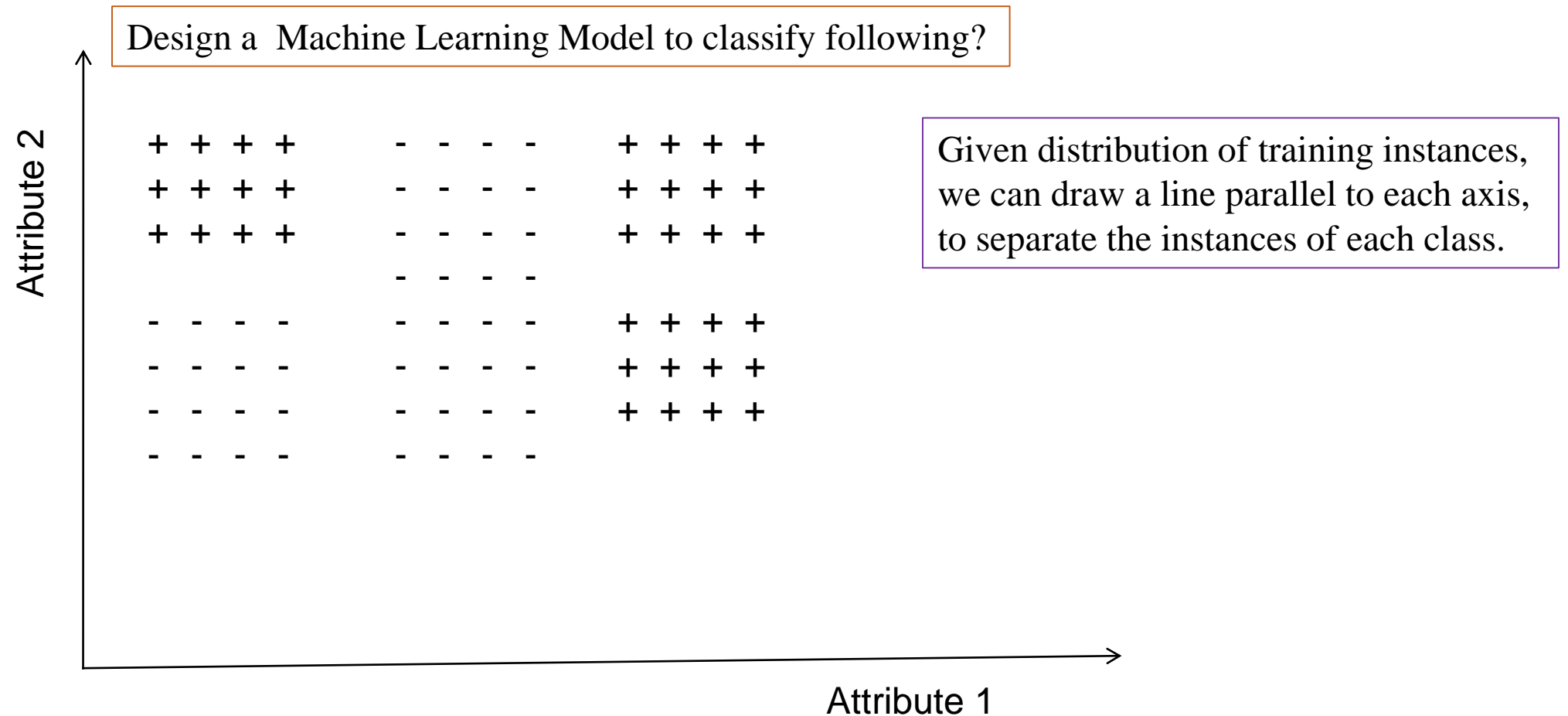
Siman Giri



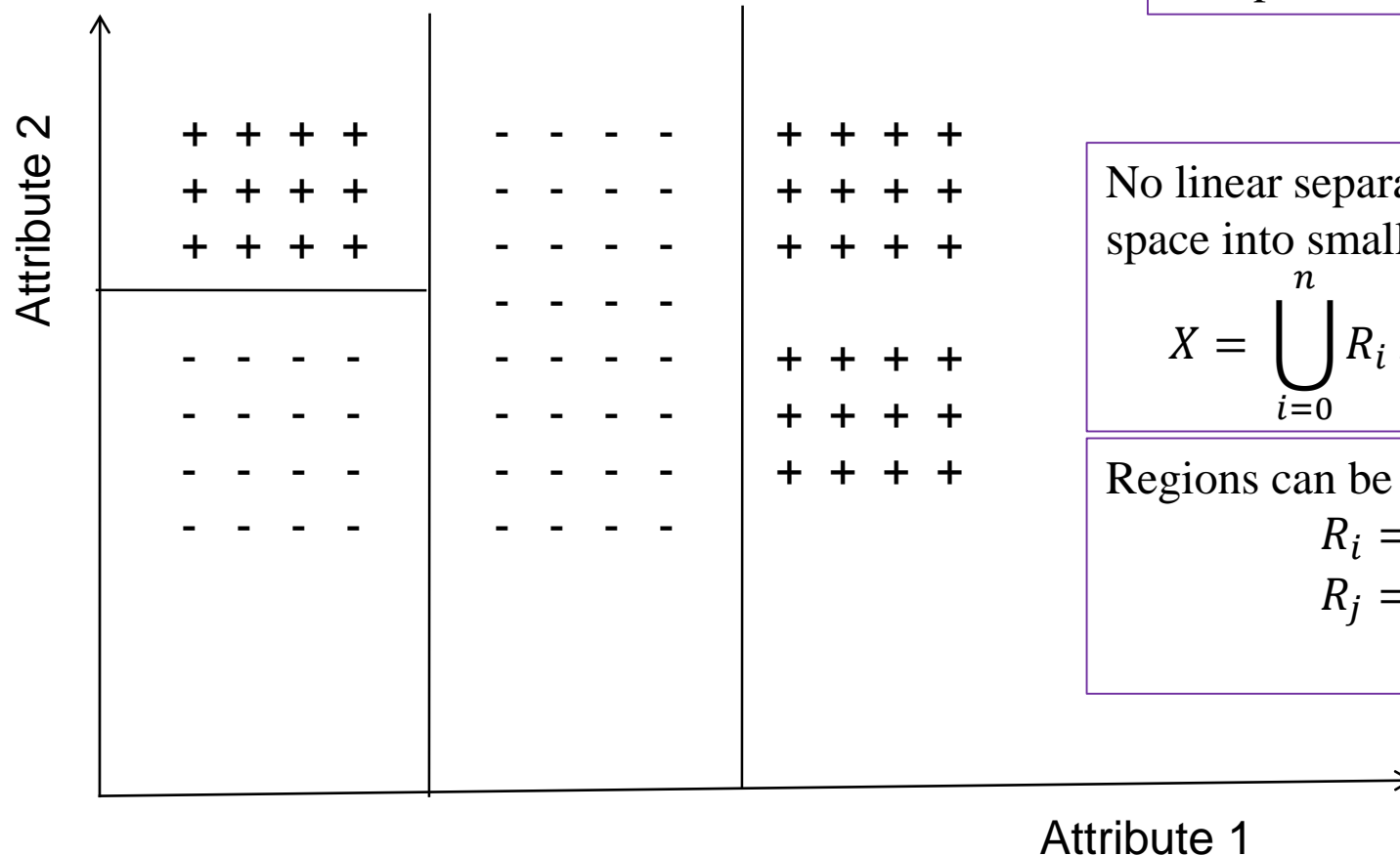
Story So Far.....



Background and Motivations!!!



Background and Motivations!!!



Given distribution of training instances, we can draw a line parallel to each axis, to separate the instances of each class.

No linear separation line exist, so divide input space into smaller disjoint regions i.e.

$$X = \bigcup_{i=0}^n R_i \text{ s.t. } R_i \cap R_j = \emptyset \text{ for } i \neq j$$

Regions can be created by splitting on features:

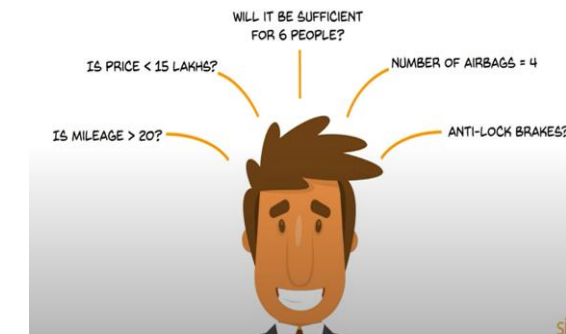
$$R_i = \{X | X_j < t, X \in R_p\}$$

$$R_j = \{X | X_j \geq t, X \in R_p\}$$

1. Decision Tree.

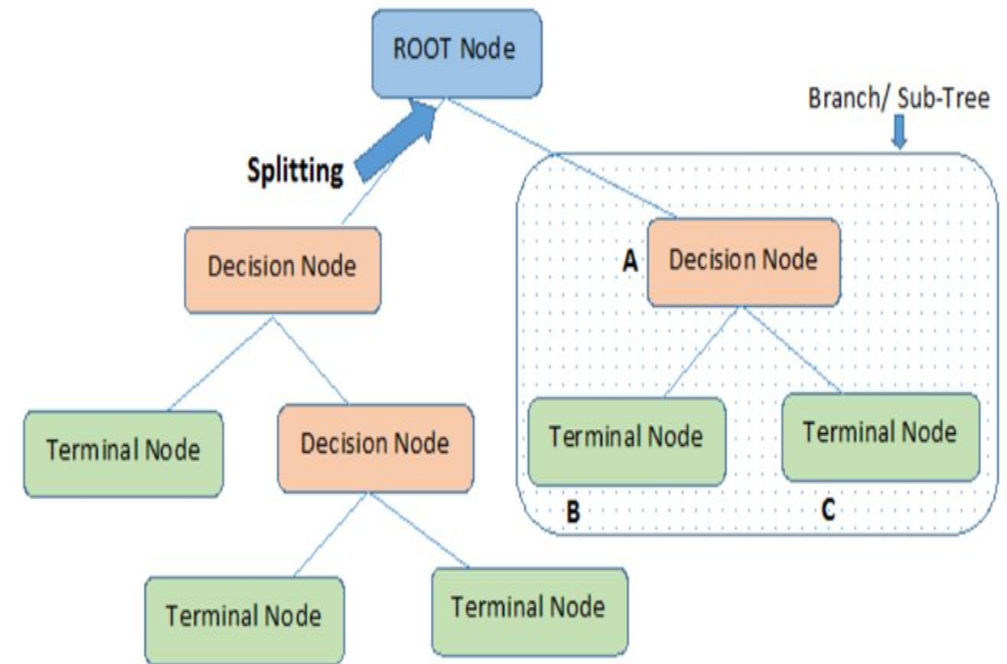
1.1 What is Decision Tree?

- It's a supervised machine learning algorithm, **which represents flowchart like inverted tree structure**,
- Where each **nodes represents a feature**, the link between the nodes represents a decision and **leaf node represents an outcome**
- Starts with **Roots** and branches off to **number of decisions** which keeps on **growing** as number of **decisions** keeps on **growing**.
- This approach is used in statistics, data mining and machine learning to assist us in making decisions.



Terminology Alert!!!

- Some important terminology related to Decision Tree:
 - **Root Node**: Very first node, from where we start dividing to various features
 - **Decision Nodes**: Nodes from splitting the root nodes are called decision nodes
 - **Leaf nodes**: nodes from where further splitting is not possible
 - **Sub-tree**: sub-section of decision tree



Where are we?

1. Decision Tree.

1.1 What is Decision Tree?

1.2 Decision Tree Algorithm.

- Let's Try to Build a Algorithm...

Example Dataset.

Rain	Wind	Temperature	Match
Yes	Yes	24	No
Yes	No	28	No
No	Yes	20	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	44	No
No	No	34	No

Where are we?

1.Decision Tree.

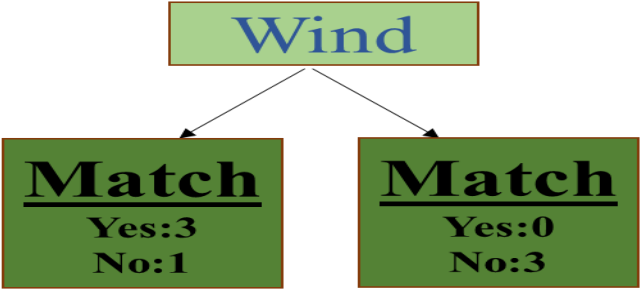
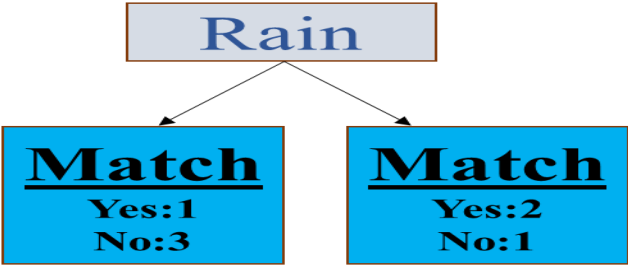
1.1 What is Decision Tree?

1.2 Decision Tree Algorithm.

- Let's Try to Build a Algorithm...

Example Dataset.

Rain	Wind	Temperature	Match
Yes	Yes	24	No
Yes	No	28	No
No	Yes	20	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	44	No
No	No	34	No



Question?
How to select best attributes?

1.3 Decision Tree-Function Approximation.

- **Given:**

- Set of possible input, output pairs:
- $\{\langle x_i, y_i \rangle\}_{i=1}^n = \{\langle x_1, y_1 \rangle, \dots, \langle x_n, y_n \rangle\}$

- **To Find:**

- Unknown mapping function:
- *i.e.* $h : X \rightarrow Y$

- **How:**

- From a **set of hypothesis** that best approximates the underlying mapping function:
- $h_{best} \in H, \text{ where } H = \{h | h: X \rightarrow Y\}$

1.4 Choosing Best Attributes.

- Ideally a **good attribute** would split the datasets into subsets that are all **positive** or all **negative** i.e. into **pure leaf nodes** and also keep the **number of splits or depth of the tree to minimum**.
 - **Key Problem: which attribute to best splits a given set of examples?**
 - Decision of making strategic splits heavily affects a **tree's accuracy**.
- Some possibilities:
 - **Random Approach(Brute-Force/Greedy):**
 - **Select any attribute at random.**
 - **Is it a good idea?**
 - **Entropy and Information Gain(ID3 Algorithm or C4.5)**
 - **Gini Impurity (CART Algorithm)**

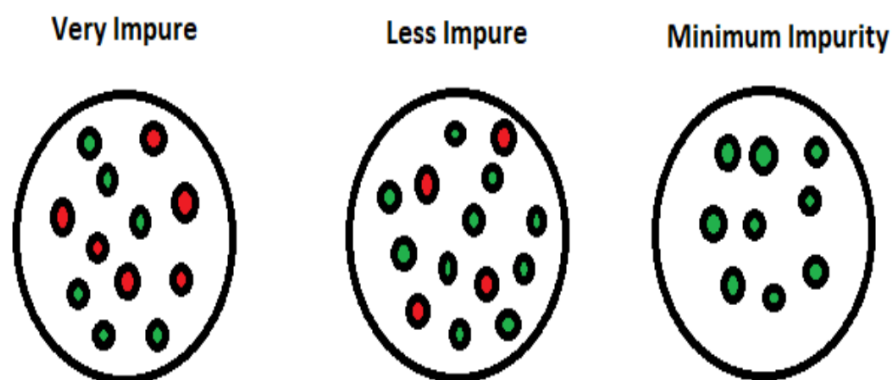
1.4 Choosing Best Attributes.

- Ideally a **good attribute** would split the datasets into subsets that are all **positive** or all **negative** i.e. into **pure leaf nodes** and also keep the **number of splits or depth of the tree to minimum**.
 - **Key Problem: which attribute to best splits a given set of examples?**
 - Decision of making strategic splits heavily affects a **tree's accuracy**.
- Some possibilities:
 - **Random Approach(Brute-Force/Greedy):**
 - **Select any attribute at random.**
 - **Is it a good idea?**
 - **Or there exist any new mathematical methods?**

1.5 Attributes Selection Measures.

- Can we calculate **some values or score** related to each attributes which measures **the purity (quality)** of the leaves it created?
 - The values can be measure of impurity:
 - What do we mean by **Impurity**?

- This can help:
 - **Entropy,**
 - **Information gain,**
 - **Gini index,**
 - **Gain Ratio,**
- Let's See what are these?



Where are we?

1. Decision Tree.

1.1 What is Decision Tree?

1.2 Decision Tree Algorithm.

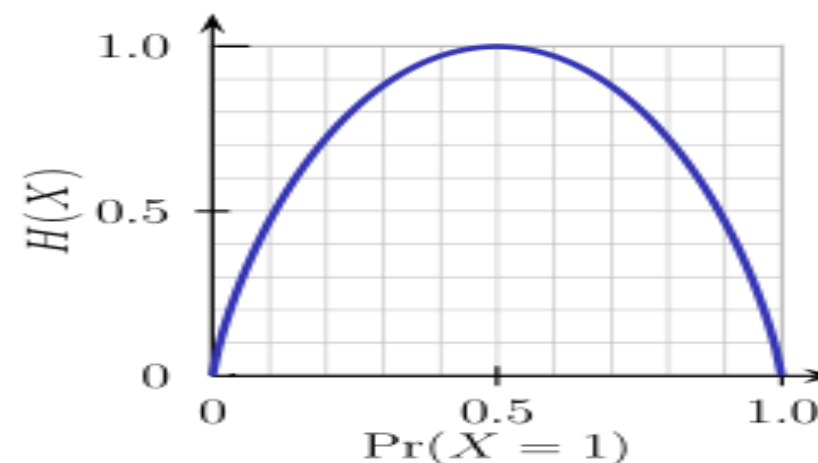
1.3 Decision Tree-Function Approximation.

1.4 Choosing Best Attributes.

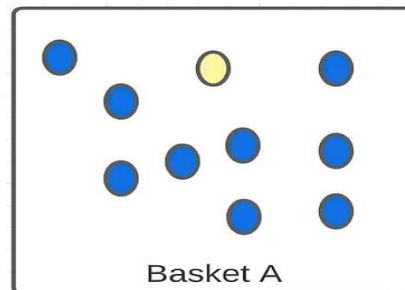
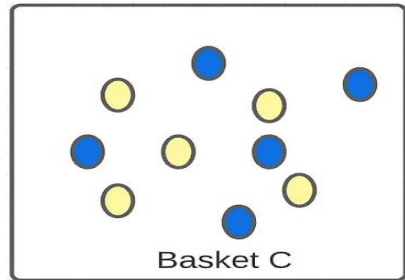
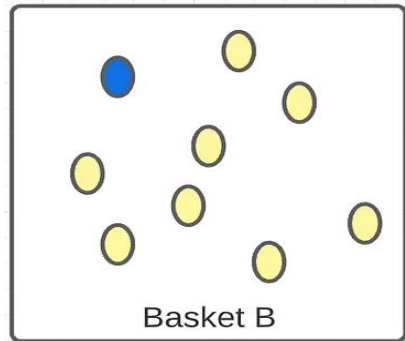
1.5 Attributes Selection Measures.

1.6 Entropy.

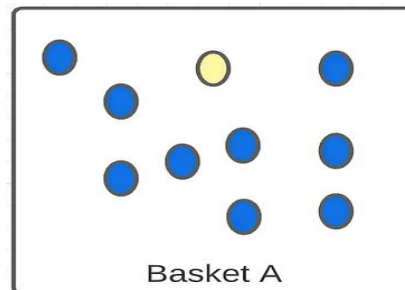
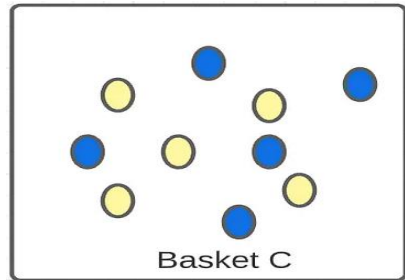
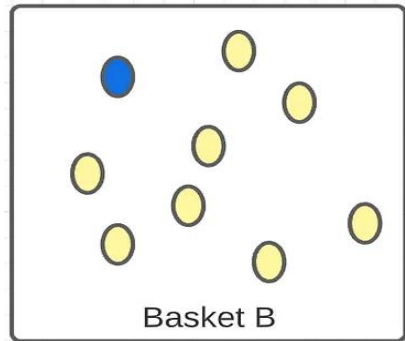
- Entropy is a **Measure of Uncertainty**
- Entropy is a **Measure of information**
- Represented by **E or H**
 - $E = -\sum p_i \log_2 p_i$



Measure of Surprise!!!

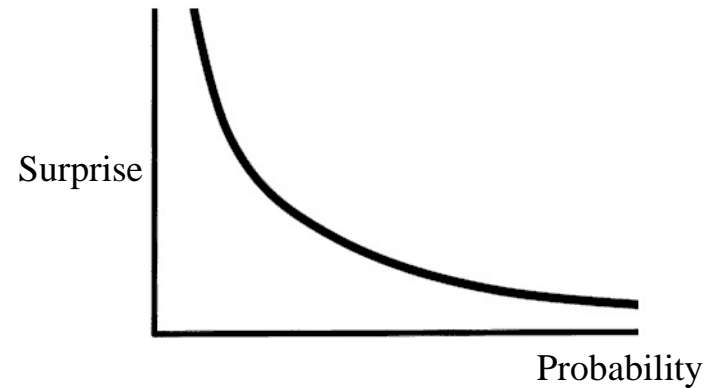


Measure of Surprise!!!

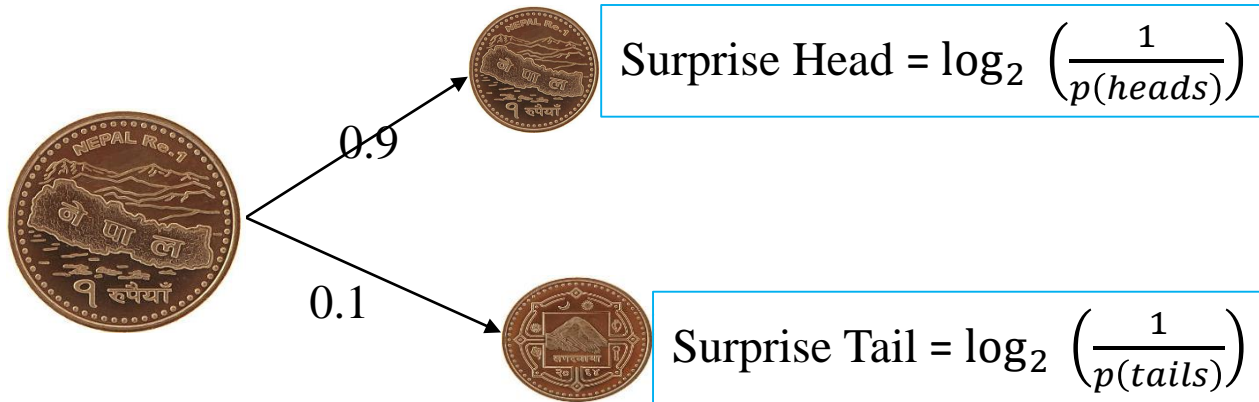


Surprise can be somewhat inversely related to probability!!!!

$$\text{Surprise} = \log\left(\frac{1}{\text{probability}}\right)$$



Coin Toss Example:



	Heads	Tails
P(x)	0.9	0.1
$\log_2 1/p(x)$	0.15	3.32

$$\text{Entropy} = \sum \log \left(\frac{1}{p(x)} \right) p(x)$$

$$\begin{aligned} \text{Entropy} &= \sum p(x) \log \left(\frac{1}{p(x)} \right) \\ &= \sum p(x) [\log(1) - \log(p(x))] \\ &= \sum p(x) [0 - \log(p(x))] \\ &= -\sum p(x) \log(p(x)) \end{aligned}$$

Where are we?

1. Decision Tree.

1.1 What is Decision Tree?

1.2 Decision Tree Algorithm.

1.3 Decision Tree-Function Approximation.

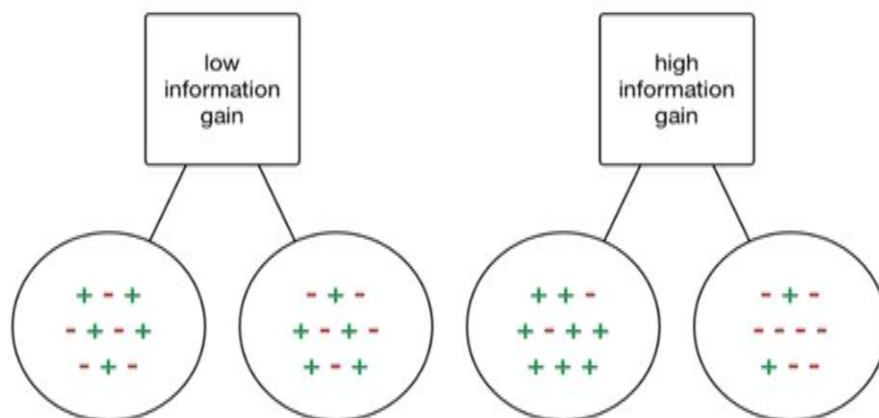
1.4 Choosing Best Attributes.

1.5 Attributes Selection Measures.

1.6 Entropy.

1.7 Information Gain.

- **IG** is a statistical property that measures how well a given attribute separates the training examples according to their target classification.
- Measures the impurity in the leaf nodes



- Information gain is calculated by comparing the entropy of the dataset before and after a transformation.
 - $\text{Information Gain}(T, X) = \text{Entropy}(T) - \text{Entropy}(T, X);$
 - *Here: $T \rightarrow \text{Current state}; X \rightarrow \text{Current Attributes}$*
- **For Now: $\text{IG} = E(\text{parent}) - [\text{Avg. } E(\text{children})]$**

1.8 Gini Index/impurity

- Gini Impurity also known as Gini Index
 - determined by deducting the sum of squared of probabilities of each class from one, mathematically, Gini Index can be expressed as:
 - $Gini\ Index = 1 - \sum_{i=1}^c P_i^2$
- calculates the amount of probability of a specific attributes/features that is classified incorrectly when selected randomly.
- Gini impurity is **lower bounded by 0**, with 0 occurring if the data set contains only one class or if we reached pure leaves
- Does the similar task as Information Gain.

1.9 Gini Index Vs. Information Gain.

- Gini impurity and entropy/Information Gain are often interchanged in the construction of decision trees.
- The method of the Gini Index is **used by CART algorithms**, in contrast to it, Information Gain is **used in ID3, C4.5 algorithms**.
- **How do you pick one?**
 - Depends on the algorithm you are implementing.
 - **Accuracy:**
 - **Neither metric results in a more accurate tree than the other.**
 - **Computational Complexity:**
 - preference might go to Gini since it doesn't involve a more computationally intensive log to calculate.

2. How to Learn Decision Tree.

2. Learning Decision Tree.

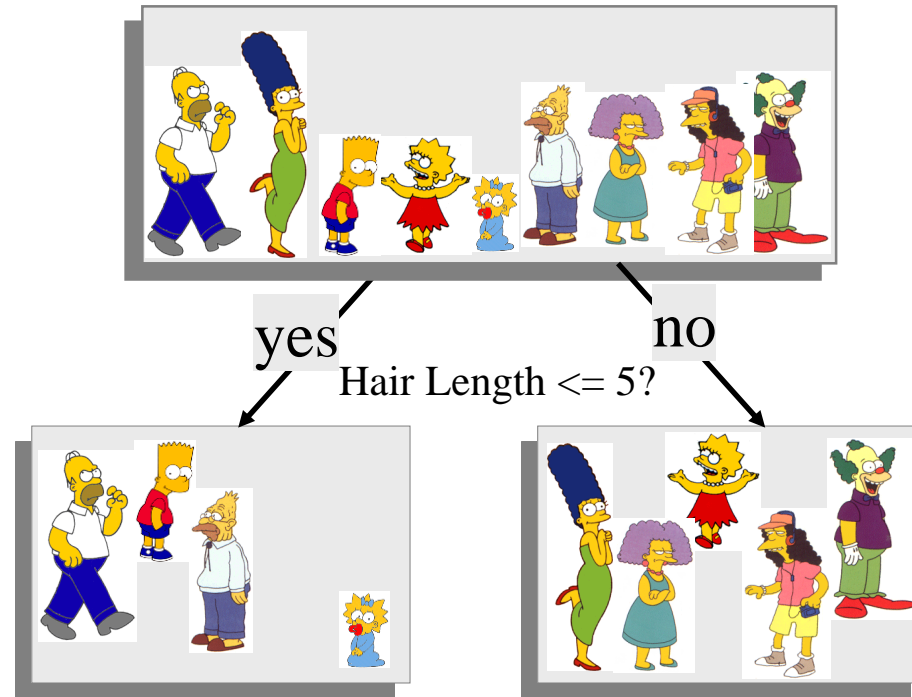
Example-The Simpsons.

Person	Hair Length	Weight	Age	Class
 Homer	0"	250	36	M
 Marge	10"	150	34	F
 Bart	2"	90	10	M
 Lisa	6"	78	8	F
 Maggie	4"	20	1	F
 Abe	1"	170	70	M
 Selma	8"	160	41	F
 Otto	10"	180	38	M
 Krusty	6"	200	45	M

Let us try splitting on *Hair length*!!!

$$Entropy(S) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right)$$

$$Entropy(4\text{F}, 5\text{M}) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9) = 0.9911$$



$$Entropy(1\text{F}, 3\text{M}) = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4) = 0.8113$$

$$Entropy(3\text{F}, 2\text{M}) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.9710$$

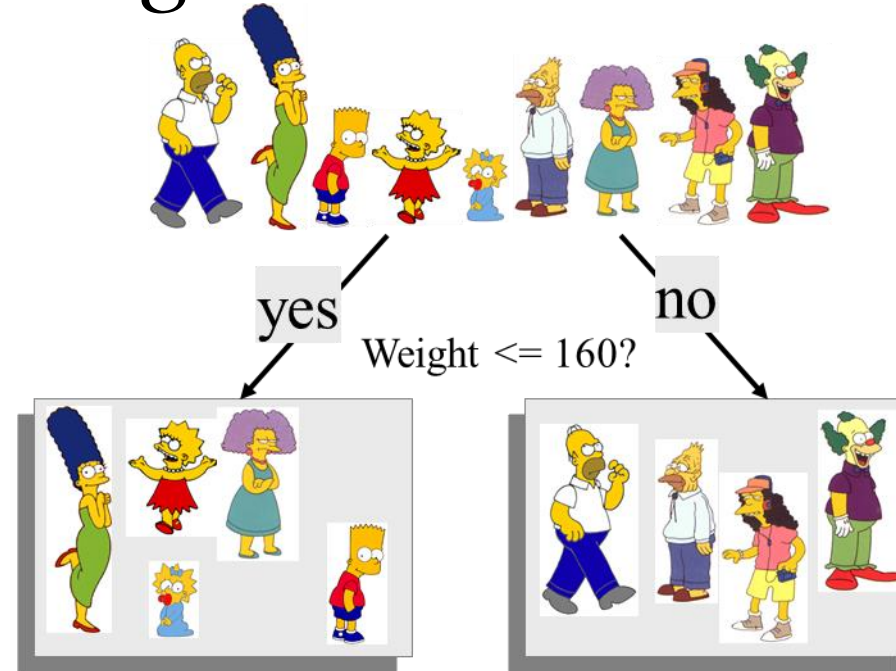
$$Gain(\text{Hair Length} \leq 5) = 0.9911 - (4/9 * 0.8113 + 5/9 * 0.9710) = 0.0911$$

$$Gain(A) = E(\text{Current set}) - \sum E(\text{all child sets})$$

Let us try splitting on *Weight!!!*

$$Entropy(S) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right)$$

$$Entropy(4\text{F}, 5\text{M}) = -(4/9) \log_2(4/9) - (5/9) \log_2(5/9) \\ = \mathbf{0.9911}$$



$$Entropy(4\text{F}, 1\text{M}) = -(4/5) \log_2(4/5) - (1/5) \log_2(1/5) \\ = \mathbf{0.7219}$$

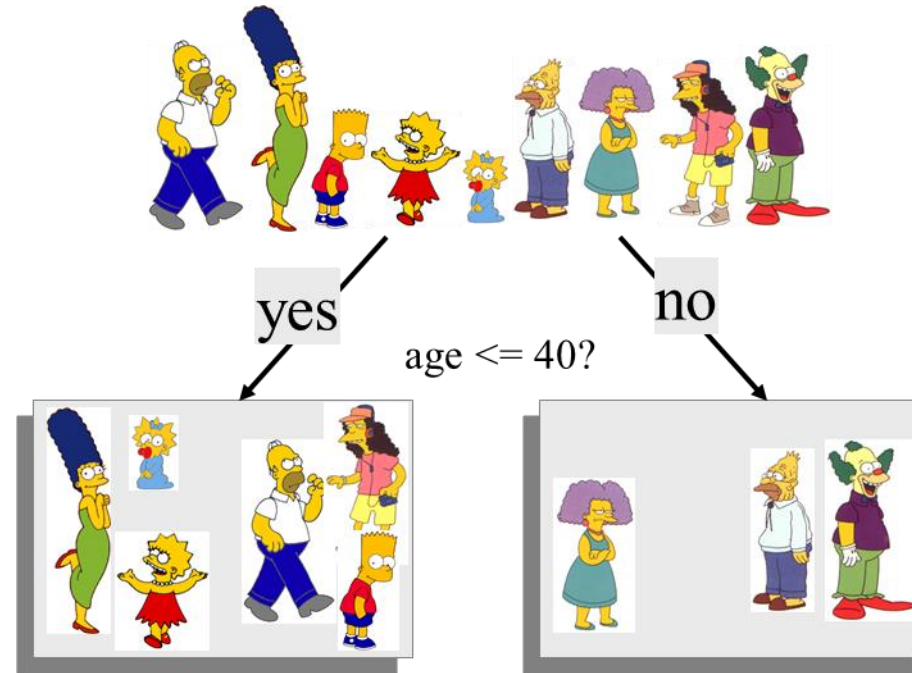
$$Entropy(0\text{F}, 4\text{M}) = -(0/4) \log_2(0/4) - (4/4) \log_2(4/4) \\ = \mathbf{0}$$

$$Gain(\text{Weight} \leq 160) = \mathbf{0.9911} - (5/9 * \mathbf{0.7219} + 4/9 * \mathbf{0}) = \mathbf{0.5900}$$

Let us try splitting on *Age*!!!

$$Entropy(S) = -\frac{p}{p+n} \log_2 \left(\frac{p}{p+n} \right) - \frac{n}{p+n} \log_2 \left(\frac{n}{p+n} \right)$$

$$Entropy(4\text{F}, 5\text{M}) = -(4/9) \log_2(4/9) - (5/9) \log_2(5/9) \\ = \mathbf{0.9911}$$



$$Entropy(3\text{F}, 3\text{M}) = -(3/6) \log_2(3/6) - (3/6) \log_2(3/6) = \mathbf{1}$$

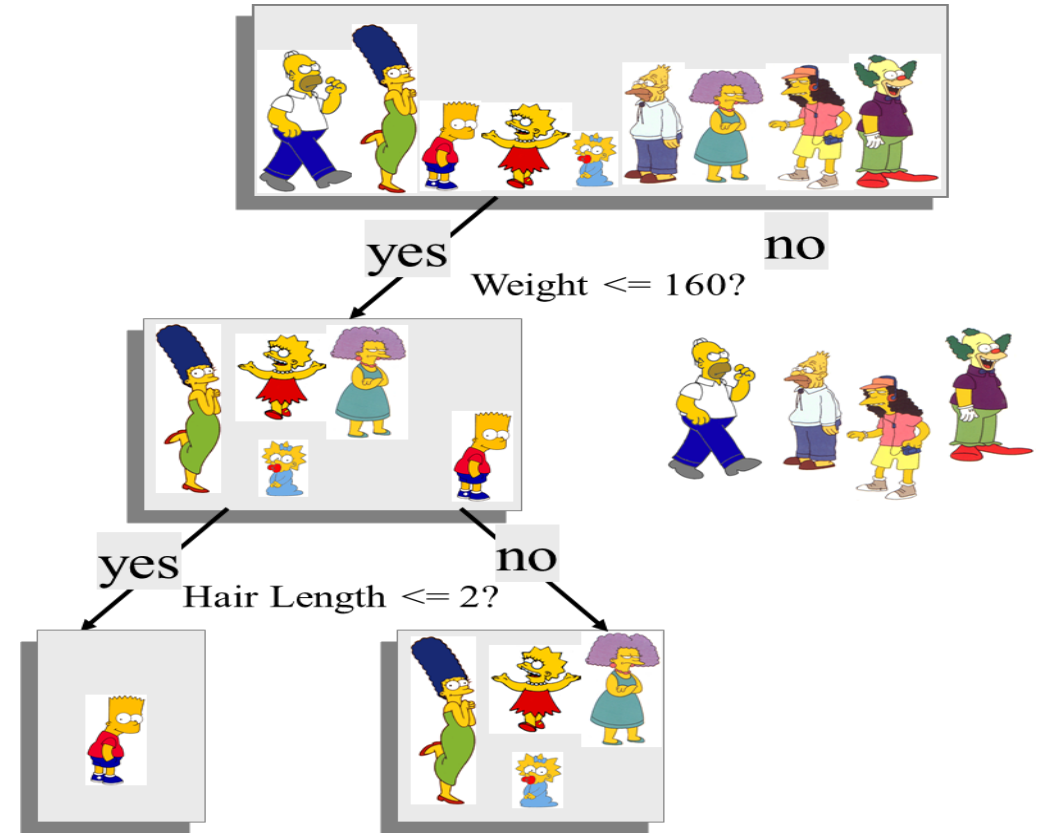
$$Entropy(1\text{F}, 2\text{M}) = -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) = \mathbf{0.9183}$$

$$Gain(\text{Age} \leq 40) = \mathbf{0.9911} - (6/9 * \mathbf{1} + 3/9 * \mathbf{0.9183}) = \mathbf{0.0183}$$

Putting it together!!!

Of the 3 features we had, *Weight* was best. But while people who weigh over 160 are perfectly classified (as males), the under 160 people are not perfectly classified... So we simply recurse!

This time we find that we can split on *Hair length*, and we are done!

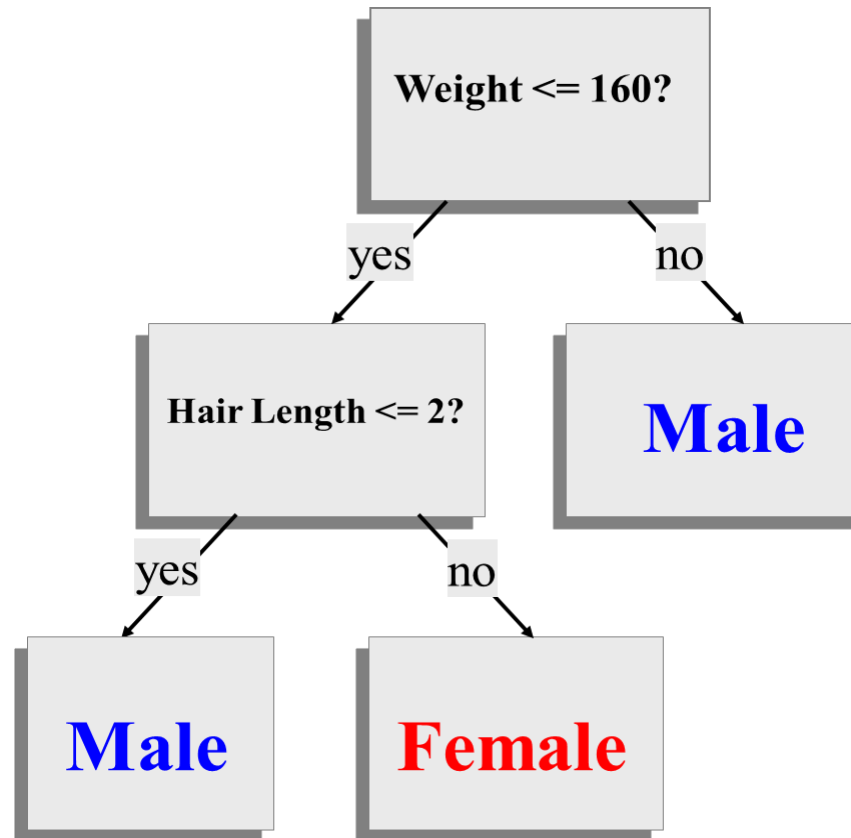


Inference!!!!

How will they will be classified?



Decision Tree->Rules



Where are we?

1.Decision Tree.

2.Learning Decision Tree.

2.1 Example-The Simpsons.

2.2 ID3 Algorithm.

- Steps in ID3 Algorithm:
 - It begins with the original set S as the root node.
 - On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates **Entropy(H)** and **Information gain(IG)** of this attribute.
 - It then selects the attribute which has the smallest Entropy or Largest Information gain.
 - The set S is then split by the selected attribute to produce a subset of the data.
 - The algorithm continues to recur on each subset, considering only attributes never selected before.

Where are we?

1.Decision Tree.

2.Learning Decision Tree.

2.1 Example-The Simpsons

2.2 ID3 Algorithm.

2.3 Advantages and Disadvantages

- Advantages:
 - Simple white box model easy to interpret, which can handle both numeric and categorical variable.
 - Performs well with large datasets and also requires no data pre-processing
 - Robust against co-linearity
- Disadvantages:
 - A small change in the training data may result in a large change in the tree and consequently the final predictions.
 - Finding a optimal decision tree is NP-Hard problem, even for simple concepts.
 - The average depth of the tree that is defined by the number of nodes or tests till classification is not guaranteed to be minimal or small under various splitting criteria.
 - Can create over-complex trees that do not generalize well from the training data leading to **overfitting**.

3. Challenges of Decision Tree.

Where are we?

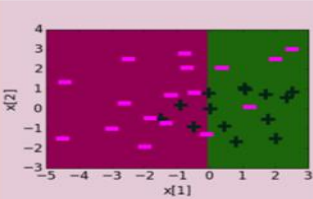
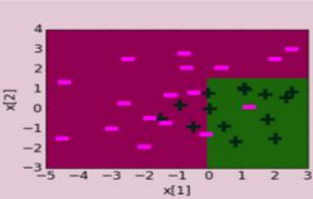
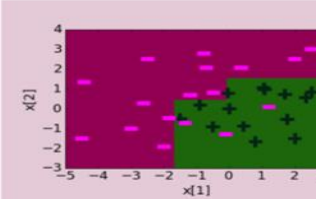
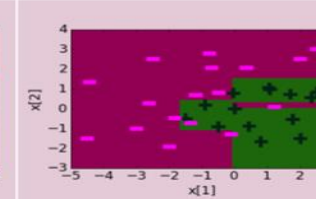
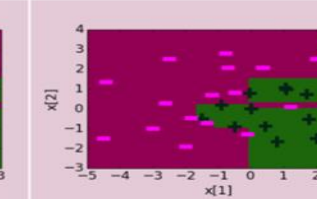
1. Decision Tree.
2. Learning of Decision Tree.
3. Challenges of Decision Tree.

3.1 Decision Tree and Overfitting.

- What happens when we increase depth?

Training error reduces with depth



Tree depth	depth = 1	depth = 2	depth = 3	depth = 5	depth = 10
Training error	0.22	0.13	0.10	0.03	0.00
Decision boundary					

Where are we?

1.Decision Tree.

2.Learning of Decision Tree.

3.Challenges of Decision Tree.

3.1 Decision Tree and Overfitting.

3.2 Pick Simpler Trees.

- Early Stopping:
 - i.e. stops the learning algorithm before tree becomes more complex
 - Limit the tree depth
 - Do not consider splits that do not cause sufficient decrease in classification error
 - Do not split an intermediate node which contains too few data points
- Pruning:
- Ensemble Methods:

Where are we?

1. Decision Tree.

2. Learning of Decision Tree.

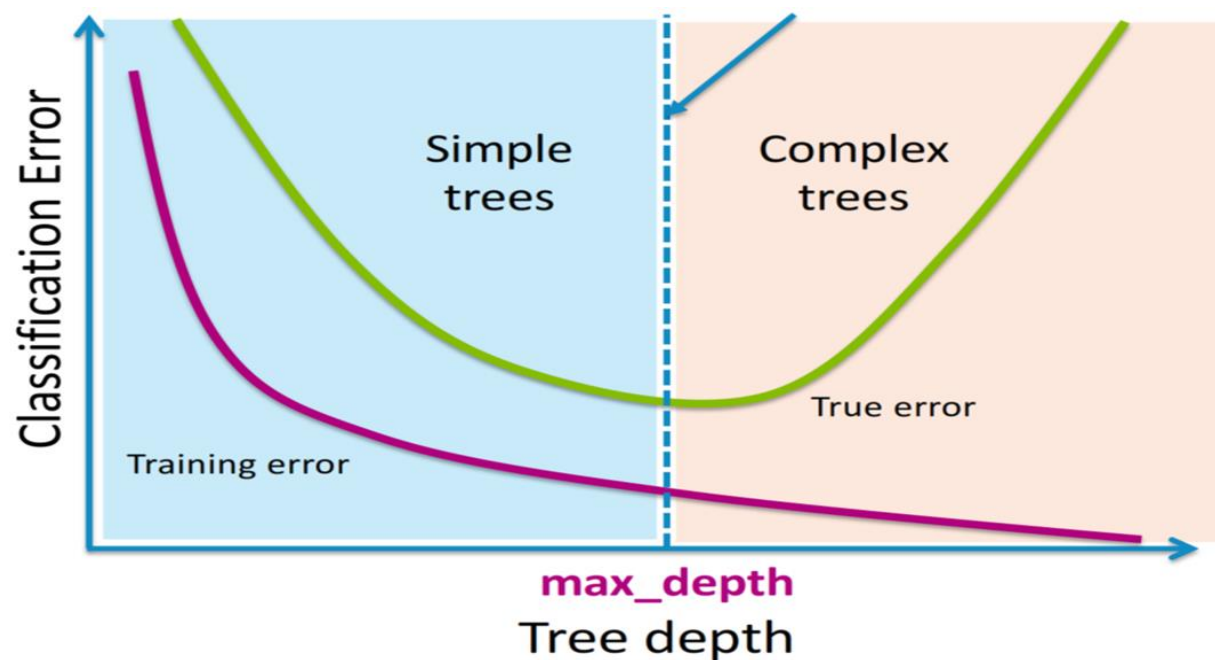
3. Challenges of Decision Tree.

3.1 Decision Tree and Overfitting.

3.2 Pick Simpler Trees.

3.3 Challenges with Early stopping.

- Hard to know exactly when to stop.
- Also we may want some branches to go deeper while other remain shallow.



- Pros of Early Stopping:
 - A reasonable heuristic approach to avoid useless splits
- Cons of Early Stopping:
 - We may miss good splits

Where are we?

1. Decision Tree.

2. Learning of Decision Tree.

3. Challenges of Decision Tree.

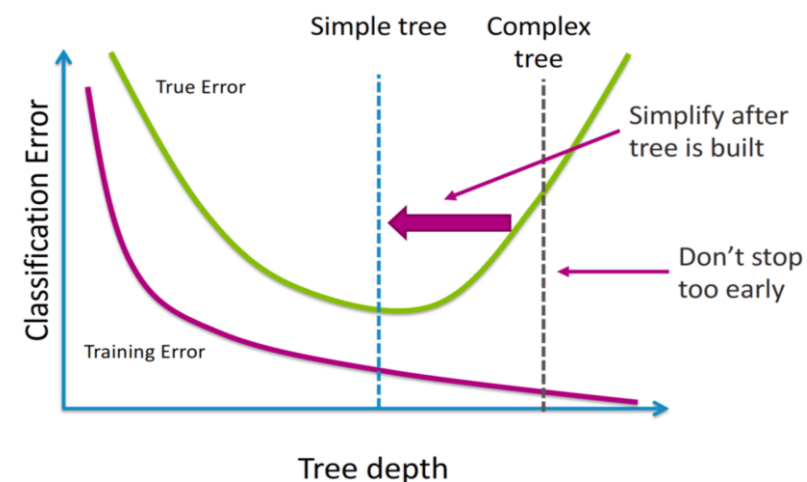
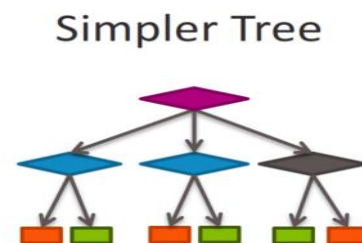
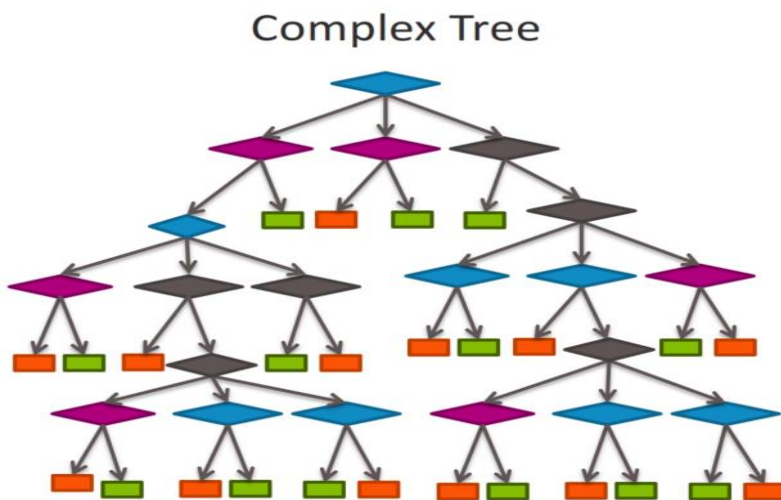
3.1 Decision Tree and Overfitting.

3.2 Pick Simpler Trees.

3.3 Challenges with Early Stopping.

3.4 Pruning.

- Pruning is when you selectively remove branches/nodes from a tree.
- The goal is to remove unwanted branches/nodes, improve the tree's structure and depth.
- By *pruning* we mean that the lower ends (the leaves) of the tree are “snipped” until the tree is much smaller.
- Intuition for Pruning: Train a complex tree, simplify later



4. Ensemble Methods.

Wisdom of crowds.

Where are we?

1.Decision Tree.

2.Learning of Decision Tree.

3.Challenges of Decision Tree.

4.Ensemble Methods.

4.1 Introduction-Ensemble Methods.

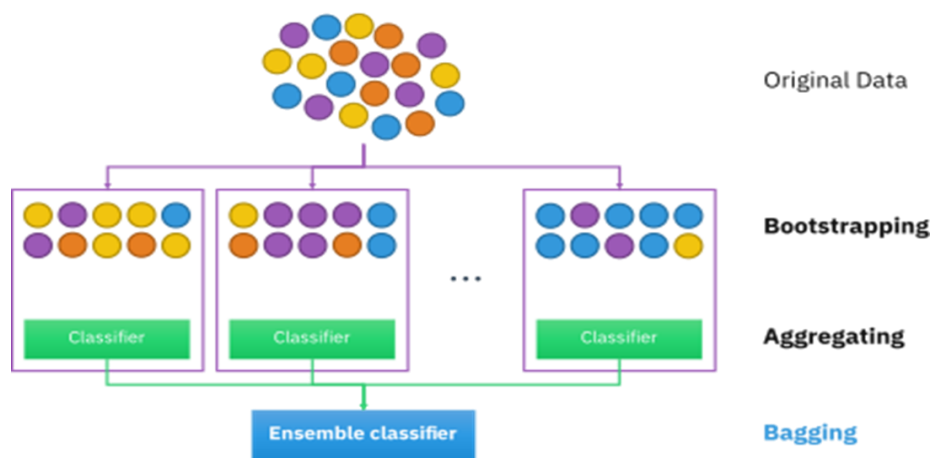
- Methods to improve the performance of weak learners.
- Shift responsibility from 1 weak learner to an “ensemble” of such learners.
 - A single decision tree often produces noisy/weak classifiers.
- Set of weak learners are combined to form a strong learner with better performance than any of them individually.
 - Let's learn multiple trees.
- How to ensure they don't all just learn the same thing??
 - --Bagging????

Where are we?

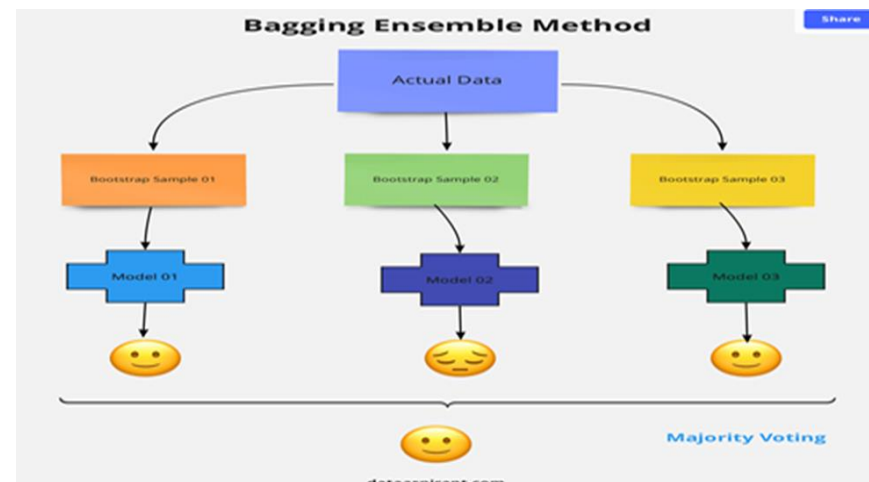
1. Decision Tree.
2. Learning of Decision Tree.
3. Challenges of Decision Tree.
4. Ensemble Methods.
 - 4.1 Introduction-Ensemble Methods.

4.2 Bagging.

- Boot-Strapping:
 - Generate multiple samples of training data, via bootstrapping, train full decision tree on each sample of data
- Aggregate:
 - For any given inputs, we output the averaged outputs of all the models for that input
 - This method is called Bagging(Brieman-1966) short for—Bootstrap Aggregating



- Benefits:
 - High Expressiveness: able to approximate complex functions and decision boundaries
 - Low Variance: averaging the prediction of all the models reduces the variance in the final prediction
- Limitations:
 - Not easily interpretable : one can no longer trace the “logic” of an output through a series of decisions based on predictor values
 - In practice Bagging, tend to be highly correlated



4.3 Random Forest.

- Definition:
 - RF is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees
 - The term was first proposed by Tim Kam Ho of Bell Labs in 1995
 - The method combines “Bagging” idea and the random selection of features

Where are we?

1.Decision Tree.

2.Learning of Decision Tree.

3.Challenges of Decision Tree.

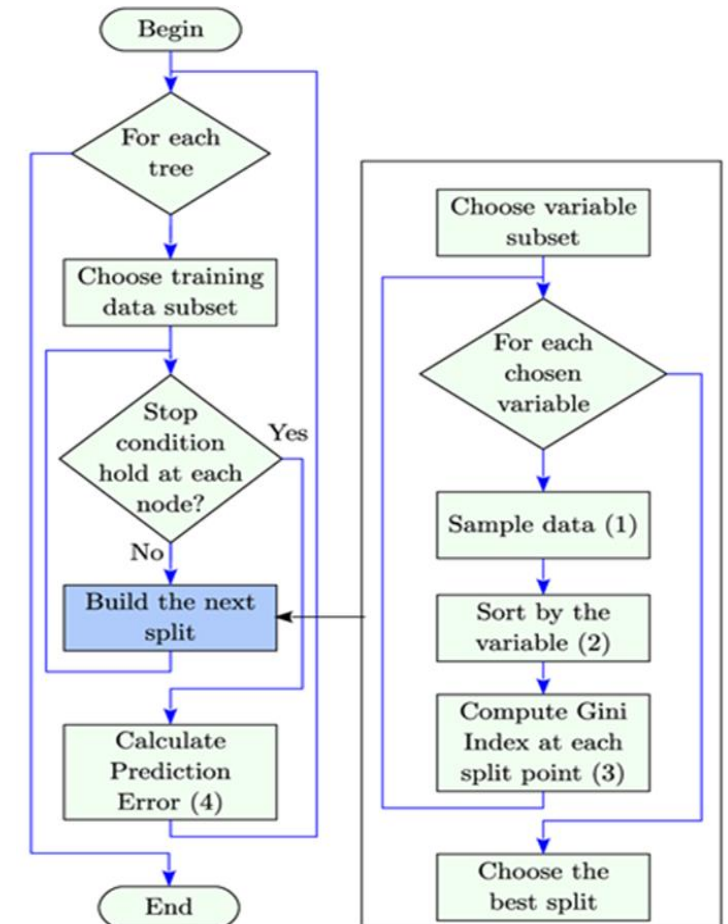
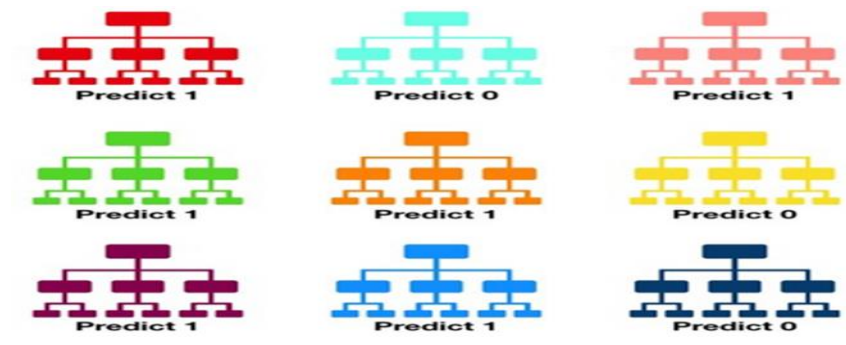
4.Ensemble Methods.

4.1 Introduction-Ensemble Methods.

4.2 Bagging.

4.3 Random Forest-Algorithm.

- It works in four steps:
 - Select Random samples from a given dataset (i.e. Bootstrapping.)
 - Construct a decision tree for each sample and get a prediction result from each decision tree
 - While building the decision tree some of the attributes are not considered intentionally.
 - Perform a vote for each predicted result (i.e. Aggregating).
 - Select the prediction result with the most votes as the final prediction



Where are we?

1. Decision Tree.
2. Learning of Decision Tree.
3. Challenges of Decision Tree.
4. Ensemble Methods.
 - 4.1 Introduction-Ensemble Methods.
 - 4.2 Bagging.
 - 4.3 Random Forest-Algorithm.

4.4 Pros and Cons.

- Pros:
 - considered as a highly accurate and robust method because of the number of decision trees participating in the process.
 - does not suffer from the overfitting problem. The main reason is that it takes the average of all the predictions, which cancels out the biases.
 - can be used in both classification and regression problems.
- Cons
 - is slow in generating predictions because it has multiple decision trees
 - is difficult to interpret compared to a decision tree, where you can easily make a decision by following the path in the tree.

Where are we?

1.Decision Tree.

2.Learning of Decision Tree.

3.Challenges of Decision Tree.

4.Ensemble Methods.

4.1 Introduction-Ensemble Methods.

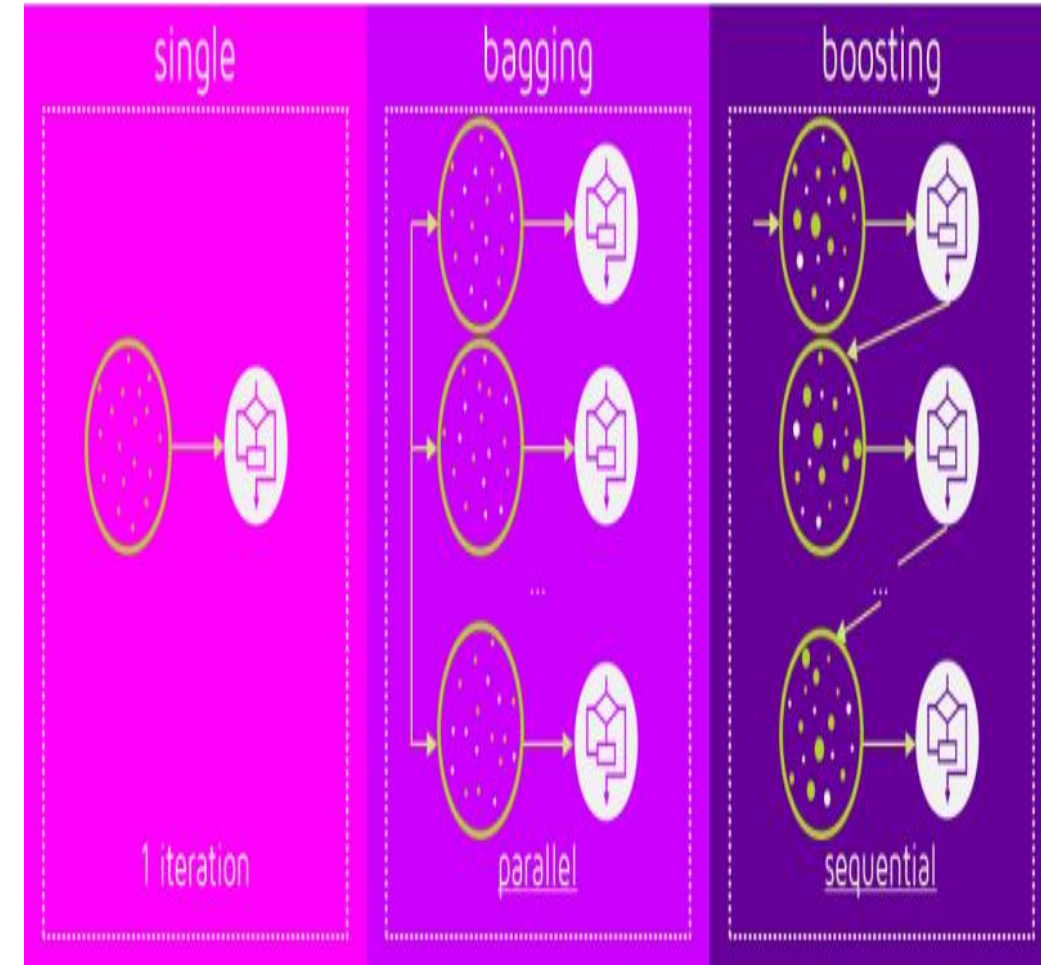
4.2 Bagging.

4.3 Random Forest-Algorithm.

4.4 Pros and Cons-Fandom Forest.

4.3 Boosting.

- Boosting does not involve bootstrap sampling
- Trees are grown sequentially:
 - each tree is grown using information from previously grown trees-**weights**.
- Like bagging, boosting involves combining many decision trees,



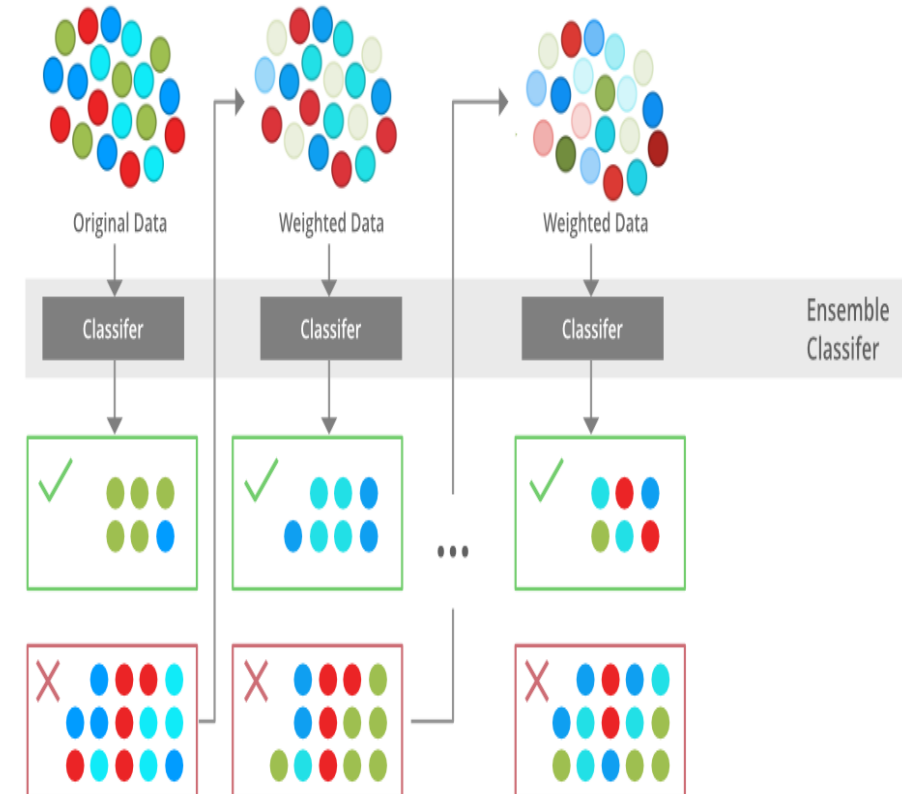
Where are we?

1. Decision Tree.
2. Learning of Decision Tree.
3. Challenges of Decision Tree.
4. Ensemble Methods.
 - 4.1 Introduction-Ensemble Methods.
 - 4.2 Bagging.
 - 4.3 Random Forest-Algorithm.
 - 4.4 Pros and Cons-Fandom Forest.

4.5 Adaboost..

- **Algorithm:**

- *Initialize the dataset and assign equal weight to each of the data point.*
- *Provide this as input to the model and identify the wrongly classified data points.*
- *Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points. And then normalize the weights of all data points.*
- *if (got required results)*
 Go to step 5
 else
 Go to step 2
- *End*



Where are we?

1. Decision Tree.
2. Learning of Decision Tree.
3. Challenges of Decision Tree.
4. Ensemble Methods.
 - 4.1 Introduction-Ensemble Methods.
 - 4.2 Bagging.
 - 4.3 Random Forest-Algorithm.
 - 4.4 Pros and Cons-Random Forest.
 - 4.5 Adaboost

4.6 Bagging Vs. Boosting.

- **Bagging** is the simplest way of combining predictions that belong to the same type while **Boosting** is a way of combining predictions that belong to the different types.
- **Bagging** aims to decrease variance, not bias while **Boosting** aims to decrease bias, not variance.
- In **Bagging** each model receives equal weight whereas in **Boosting** models are weighted according to their performance.
- In **Bagging** each model is built independently whereas in **Boosting** new models are influenced by performance of previously built models.
- In **Bagging** different training data subsets are randomly drawn with replacement from the entire training dataset. In **Boosting** every new subsets contains the elements that were misclassified by previous models.
- **Bagging** tries to solve over-fitting problem while **Boosting** tries to reduce bias.
- If the classifier is unstable (high variance), then we should apply **Bagging**. If the classifier is stable and simple (high bias) then we should apply **Boosting**.
- **Bagging** is extended to Random forest model while **Boosting** is extended to **Gradient boosting**.

Where are we?

1. Decision Tree.

2. Learning of Decision Tree.

3. Challenges of Decision Tree.

4. Ensemble Methods.

4.1 Introduction-Ensemble Methods.

4.2 Bagging.

4.3 Random Forest-Algorithm.

4.4 Pros and Cons-Random Forest.

4.5 Adaboost

4.7 Selecting Best Techniques.

- Whether to select Bagging or Boosting for a particular problem.
 - It depends on the **data, the simulation and the circumstances**.
- Bagging and Boosting *decrease the variance of your single estimate* as they combine several estimates from different models. So the result may be a *model with higher stability*.
- If the problem is that the *single model gets a very low performance*, *Bagging will rarely get a better bias*.
 - However, Boosting could generate a combined model with *lower errors as it optimizes the advantages and reduces pitfalls of the single model*.
- By contrast, *if the difficulty of the single model is over-fitting*, then *Bagging is the best option*.
 - Boosting for its part doesn't help to avoid over-fitting.

Thank You!!!!