Decision Tree Algorithms

Learning Objectives

- Explain the basics of the decision tree
- Describe the Gini index and entropy
- Explain information gain to build decision trees
- Discuss the construction of a decision tree
- Describe overfitting in decision trees and pruning
- List different DT algorithms
- Explain the hyperparameters and tuning for hyperparameters
- Illustrate a case study on the decision tree

Decision Trees

Example

Student	Exam Score	Pass/Fail
1	65	Pass
2	75	Pass
3	55	Fail
4	85	Pass
5	60	Fail

$$Y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

What are the parameters ?

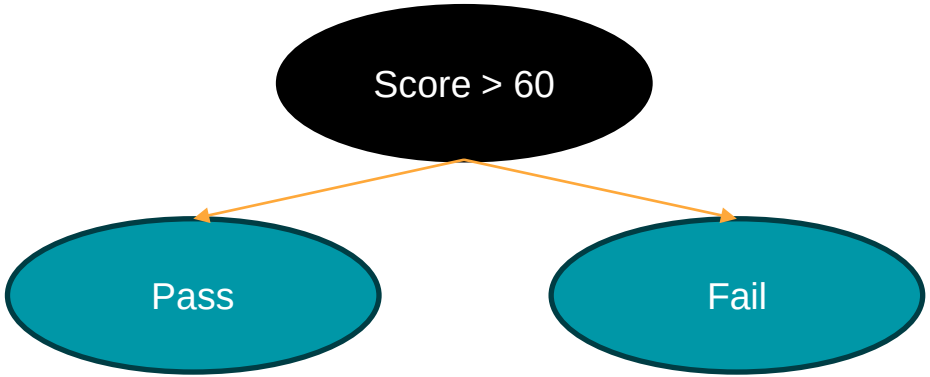
What is Logistic Regression ?

- Logistic regression is a parametric method used for binary classification tasks, where the target variable (dependent variable) is categorical with two possible outcomes (e.g., yes/no, pass/fail, 0/1).

Example

Student	Exam Score	Pass/Fail
1	65	Pass
2	75	Pass
3	55	Fail
4	85	Pass
5	60	Fail

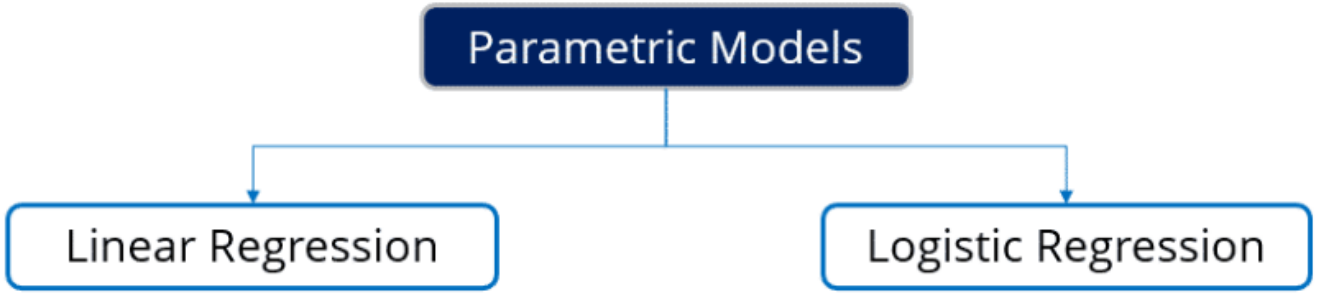
What are the parameters ?



65 – Pass
75 – Pass
85 - Pass

60 – Fail
55 – Fail

Parametric Models



- Parametric methods assume that the data is generated from a specific process, for example, usually a normal distribution.
- This allows using well-known statistics techniques to estimate the model's parameters and predict future values.

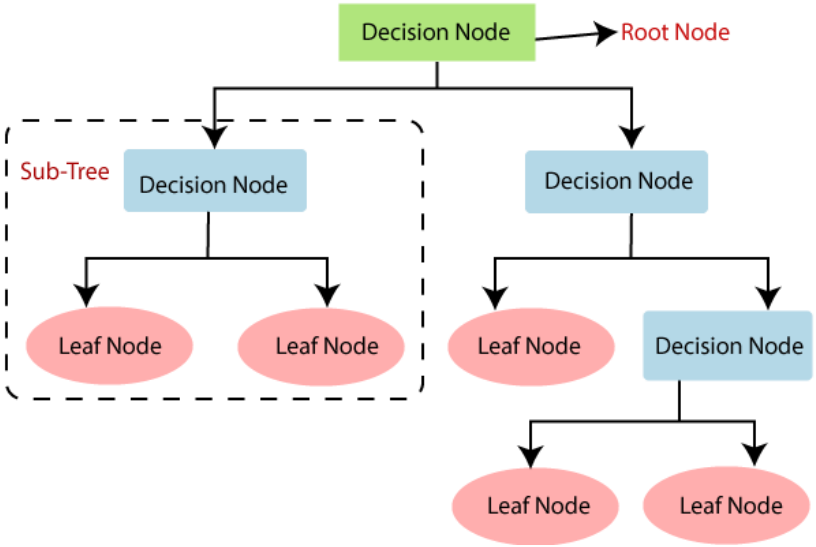
Non-parametric Models

Non-parametric models do not make explicit assumptions about the data.

Instead, non-parametric models can be seen as a set of rules or a function approximation that gets as close to the data as possible.

Decision Tree

Outlook	Temperature	Humidity	Windy	Play Tennis?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

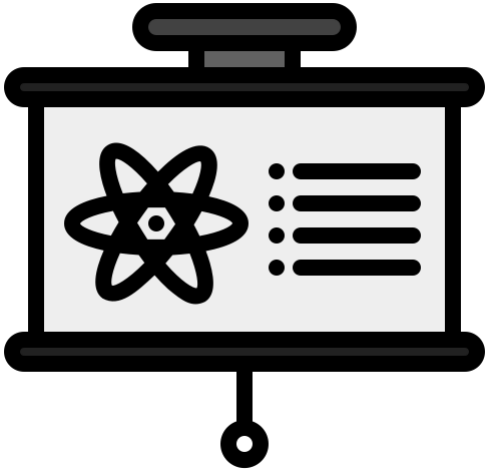


- **How select an attribute for each decision node ?**
- **What are the (im)purity metrics ?**

Information Theory

- All of these terms, purity, impurity, and information gain, are concepts that stem from a domain called information theory.

- It is a scientific study of measuring, storing, and communicating information in systems established in the early 1920s to 1940s.

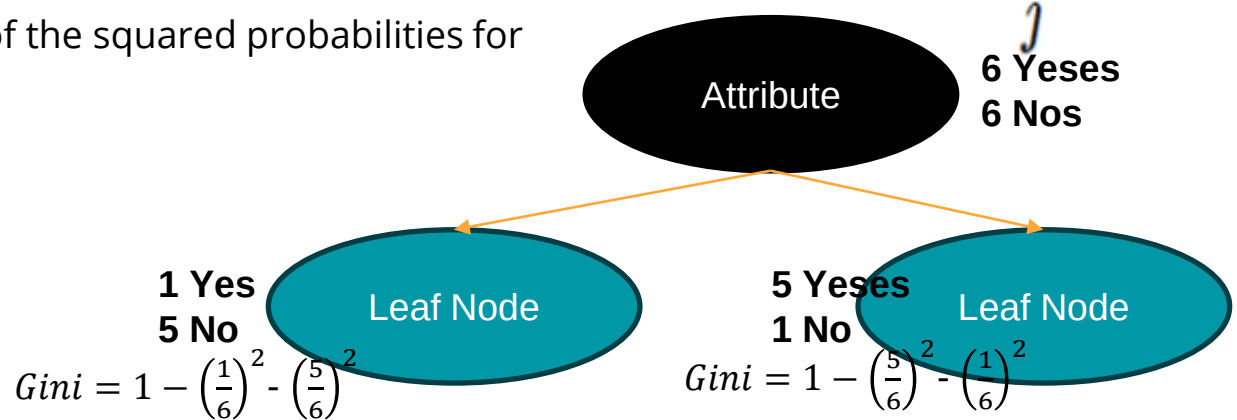


Gini Index

- The goal of the Gini index is to measure how often a randomly chosen element from a group would be incorrectly labeled based on the distribution of labels in the group.

- It is defined as the sum of the squared probabilities for each class in the group.

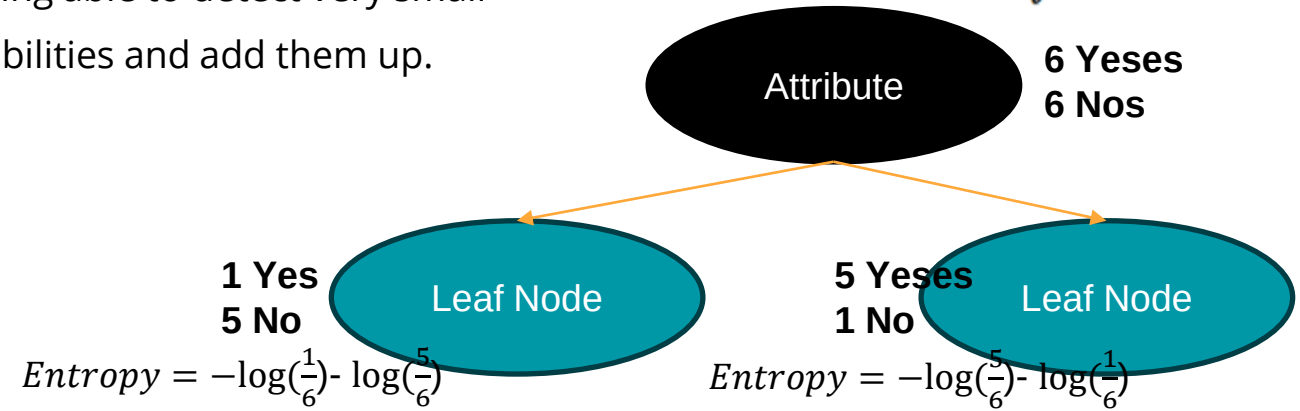
$$Gini = 1 - \sum_{j} p_j^2$$



Entropy

- Entropy measures the disorder of a group with respect to a target variable.
- The reason to use the logarithms here is that it has the handy property of being able to detect very small changes in our probabilities and add them up.

$$Entropy = -\sum_{j} p_j \log_2 p_j$$



$$Entropy = -\log(\tfrac{1}{6})\text{- }\log(\tfrac{5}{6})$$

$$Entropy = -\log(\tfrac{5}{6})\text{- }\log(\tfrac{1}{6})$$

Information Gain

- The information gained is conceptually simply the difference between the impurity of the parent group and the impurity of the resulting child groups.

- IG = Entropy(Parent) - Entropy(Children)

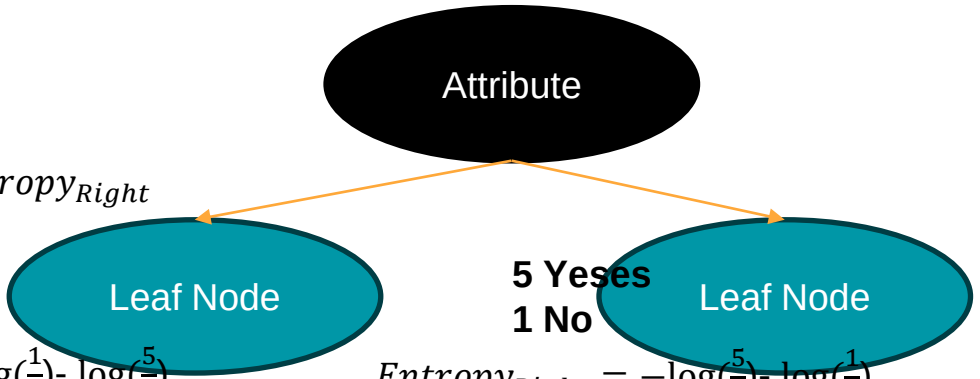
In practice

- IG = Entropy(Parent) - (weighted_avg * Entropy(Children))

$$IG = Entropy_{Parent} - \frac{6}{12}Entropy_{Left} - \frac{6}{12}Entropy_{Right}$$

$$Entropy_{Parent} = -\log(\frac{6}{12}) - \log(\frac{6}{12})$$

6 Yeses
6 Nos



1 Yes
5 Nos

$$Entropy_{Left} = -\log(\frac{1}{6}) - \log(\frac{5}{6})$$

5 Yeses
1 No

$$Entropy_{Right} = -\log(\frac{5}{6}) - \log(\frac{1}{6})$$

Gini vs. Entropy

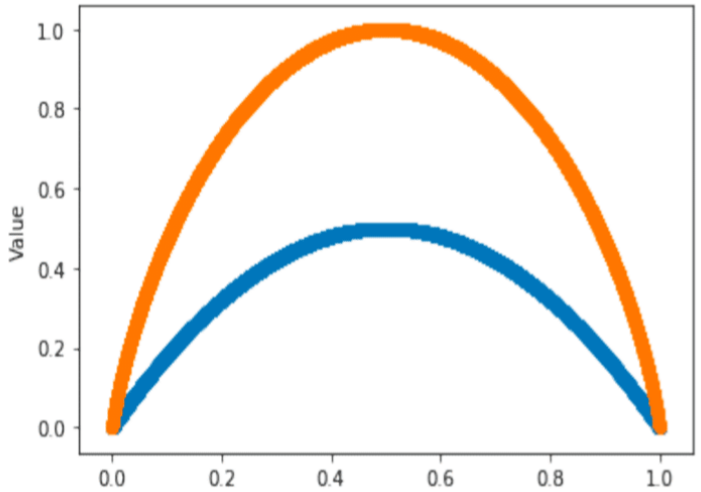
knowledgehut
upGrad

Entropy

- Entropy used to be the original impurity measure used in decision trees and therefore has a longer pedigree.

Gini

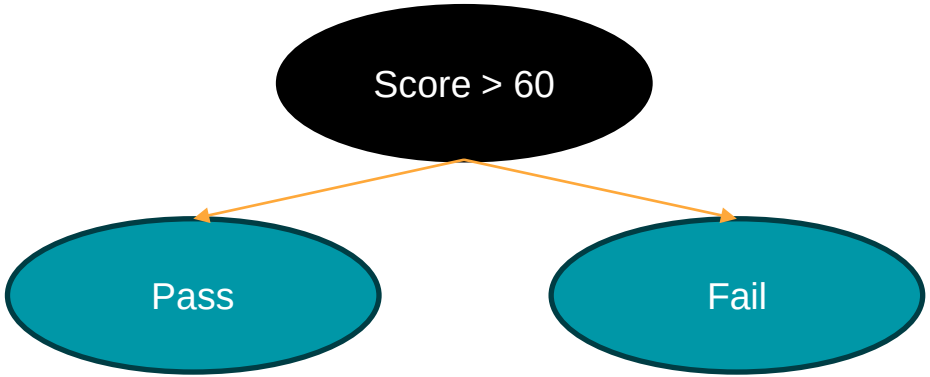
- The Gini impurity is slightly faster to compute than entropy because it doesn't require calculating a logarithm.



Dealing with Continuous Attribute

Student	Exam Score	Pass/Fail
1	65	Pass
2	75	Pass
3	55	Fail
4	85	Pass
5	60	Fail

Split Point	IG
60	…
65	…
…	…
85	…



65 – Pass
75 – Pass
85 - Pass

60 – Fail
55 – Fail

Pick the one with highest IG.

Constructing a Decision Tree

Build a Decision Tree

knowledgehut upGrad

Steps to build a decision tree:

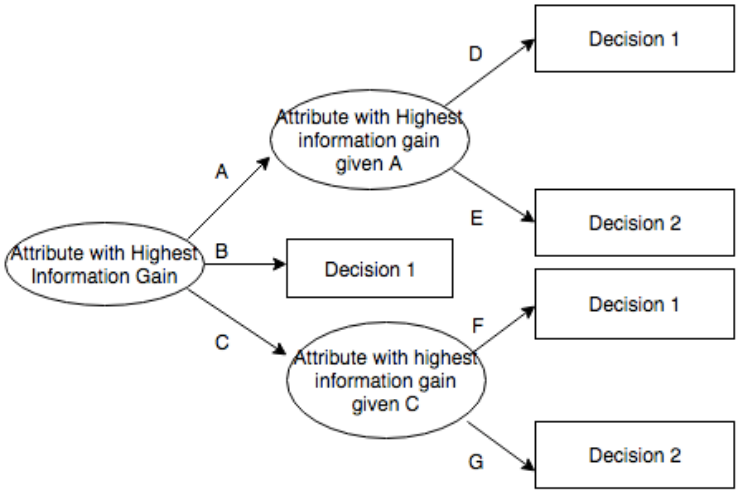
01 Find optimal threshold T that minimizes impurity metric.

02 Find optimal features by comparing impurity metrics among features.

03 Add another split when the split gives information gain.

ID3

- It is one of the oldest decision tree algorithms. Ross Quinlan developed it in 1986.
- ID3 will use entropy to find the best categorical features to do the split.
- ID3 works only with categorical features.
- ID3 is a so-called greedy algorithm that tries to find the best split at each tree level.



C4.5/C5.0

Ross Quinlan also published C4.5 in 1992. C4.5 improves on ID3 in a few ways:

- Handles categorical and numeric features (partitioning into discrete intervals).
- Uses the information gain ratio, a variant of information gain.
- Prevents overfitting by pruning; removal of branches that are not significant.
- Faster than ID3.

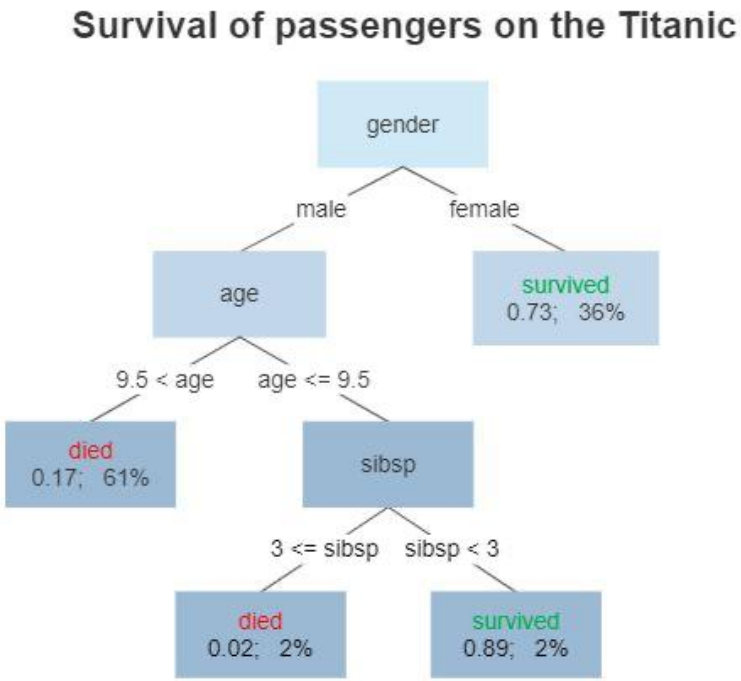
Algorithm 1.1 C4.5(D)

Input: an attribute-valued dataset D

```
Tree = {}
if D is "pure" OR other stopping criteria met then
    terminate
end if
for all attribute a ∈ D do
    Compute information-theoretic criteria if we split on a
end for
a_best = Best attribute according to above computed criteria
Tree = Create a decision node that tests a_best in the root
D_v = Induced sub-datasets from D based on a_best
for all D_v do
    Tree_v = C4.5(D_v)
    Attach Tree_v to the corresponding branch of Tree
end for
return Tree
```

CART: Classification and Regression Tree

- Leo Breiman and others in 1984 introduced CART.
- It's very similar to C4.5 but differs because it also supports numerical target variables and can be used for regression problems, as the name suggests.
- CART can also handle both categorical and numerical features, and it constructs binary trees using features and thresholds that maximize the information gained at each step using the Gini index.
- It can even handle missing values, making it very flexible and often an excellent first algorithm.



Advantages of Decision Tree

knowledgehut upGrad

- Simple to understand and interpret. It can be visualized with non-technical stakeholders.
- Require little data preparation.
- The cost of using a tree for prediction is low as the splits get smaller and smaller. Hence the prediction time is constant.
- It handles both numerical and categorical data.

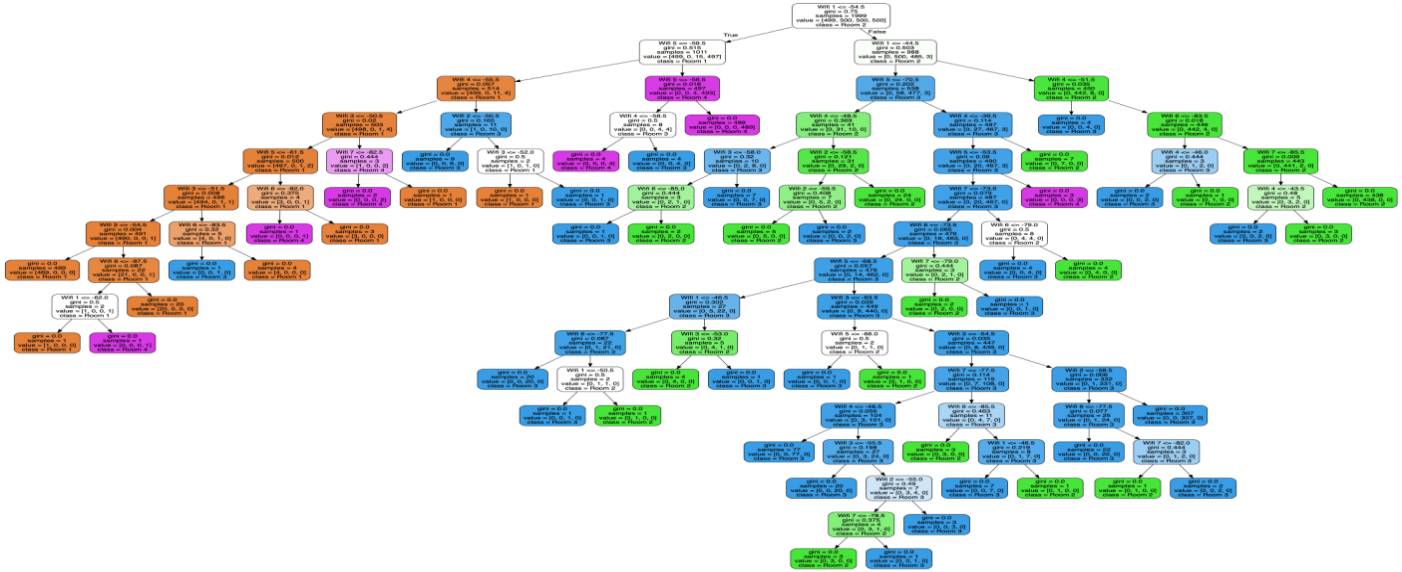
Disadvantages of Decision Tree

knowledgehut upGrad

- Easily overfits the training data and hence needs careful tuning of hyperparameters to prevent it.
- A single decision tree can be unstable.
- Performs a greedy approach, i.e., splits are optimized locally and don't guarantee global optimum.
- Imbalanced classes can become problematic.

Overfitting in Decision Trees and Pruning

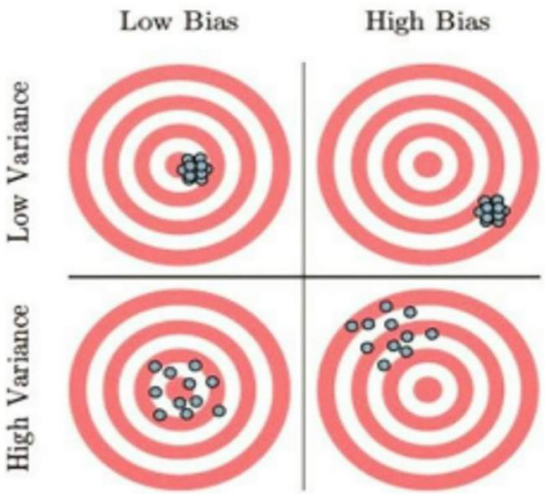
Overfitting



If a decision tree grows limitless, then the tree will keep splitting the data further, producing smaller subsets of the data, where each subset contains fewer and fewer observations.

Pruning

- To prevent overfitting, the trees are cut at some level by pruning.
- By cutting the tree more to the top allows the bias and lower variance to be introduced, whereas growing the tree deeper and deeper subsequently removes bias and adds variance.



Two options to prune the tree:

- **Pre-pruning** means stopping growing the tree earlier. Hence this process is also called early stopping.
- **Post-pruning** means we grow a tree to its full size and then cut it back.

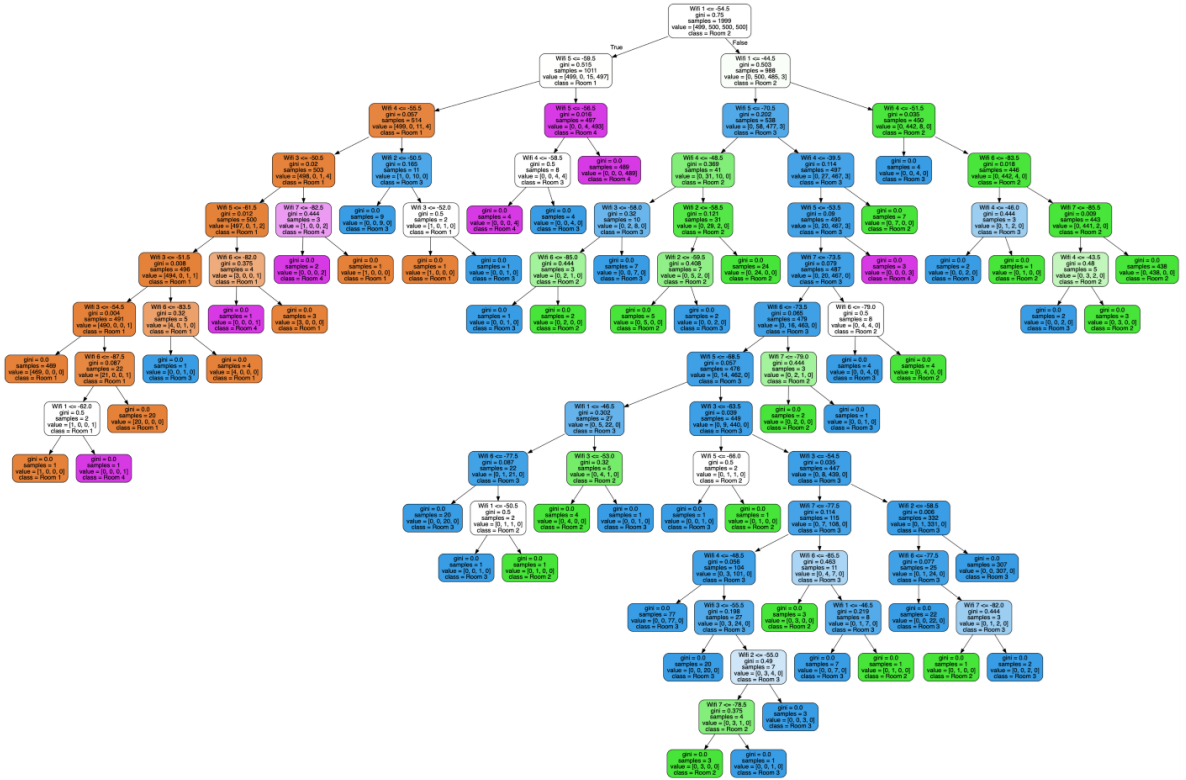
Hyperparameters and Tuning

Common Hyperparameters (criterion)

knowledgehut upGrad

1. criterion

Gini or Entropy is used as the impurity measure.



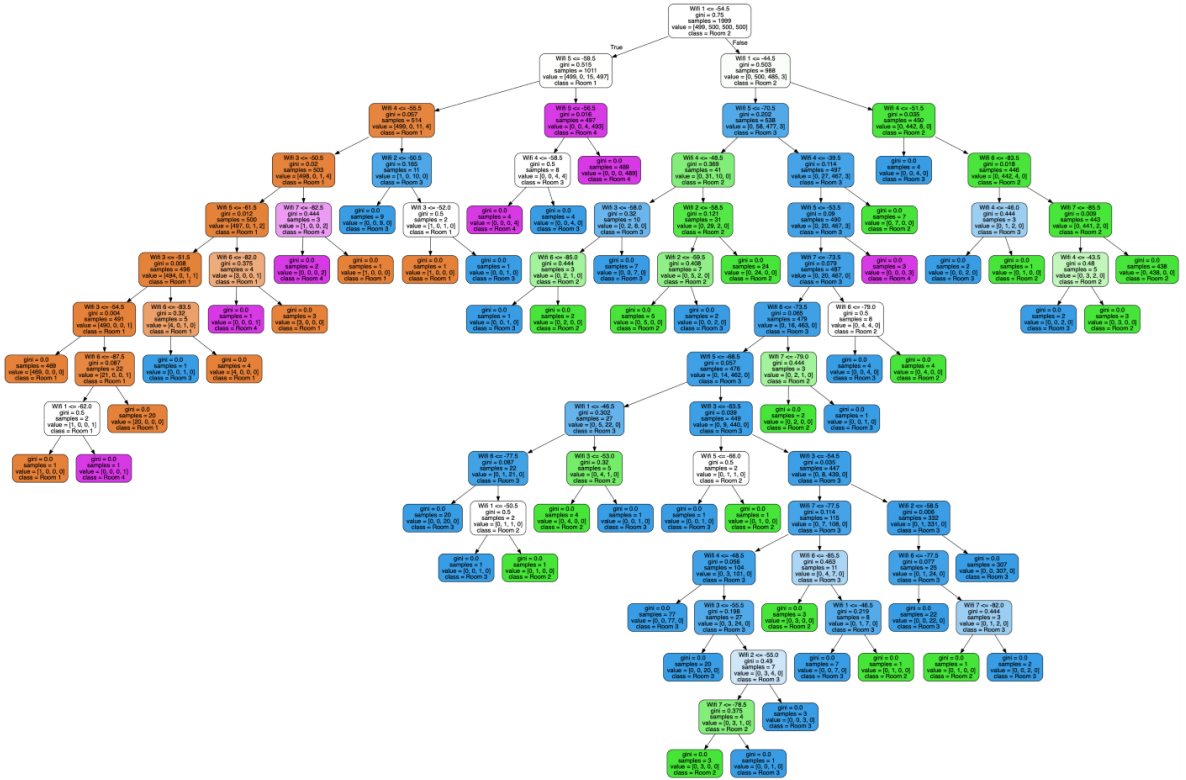
Common Hyperparameters (max_depth)

knowledgehut upGrad

2. max_depth

Controls the maximum depth of the tree.

If not specified, the tree will be expanded until all leaves contain less than the maximum number of samples in one split.



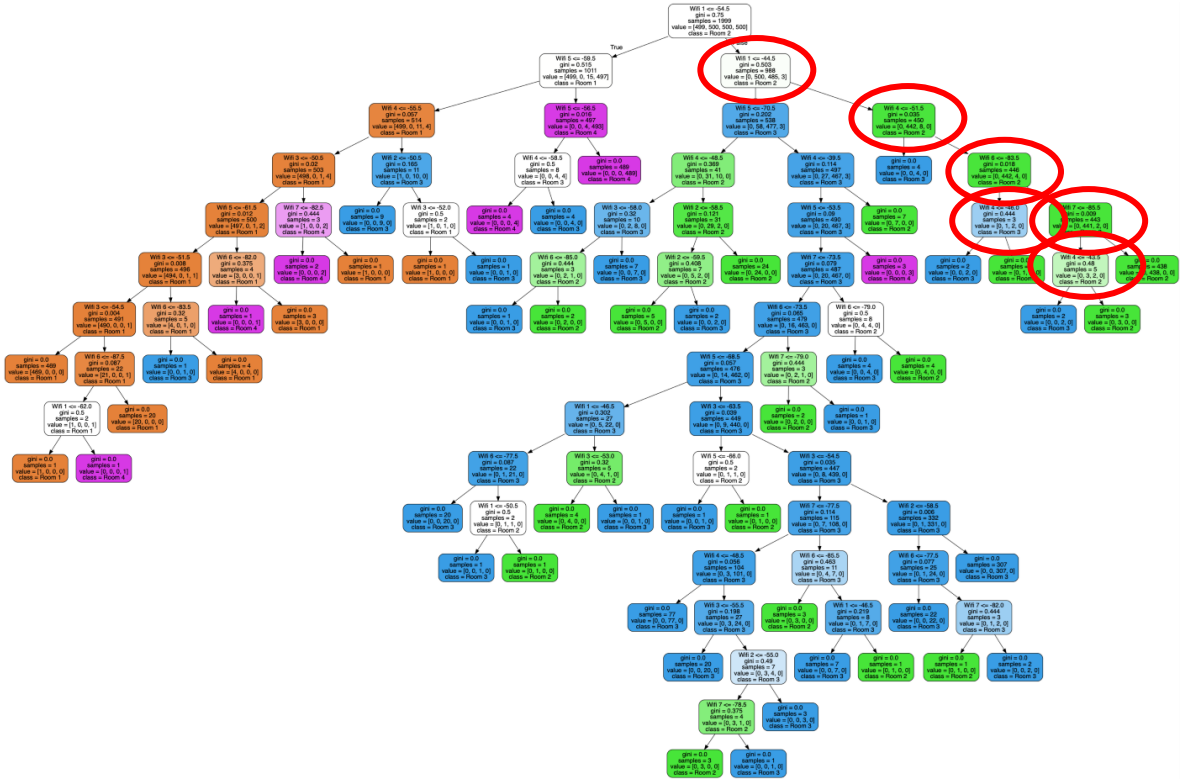
Common Hyperparameters (min_samples_split)

knowledgehut upGrad

3. min_samples_split

The minimum samples required to make a split.

- If the node contains values less than the minimum sample, the split will not happen.
- The default value for the split is 2.



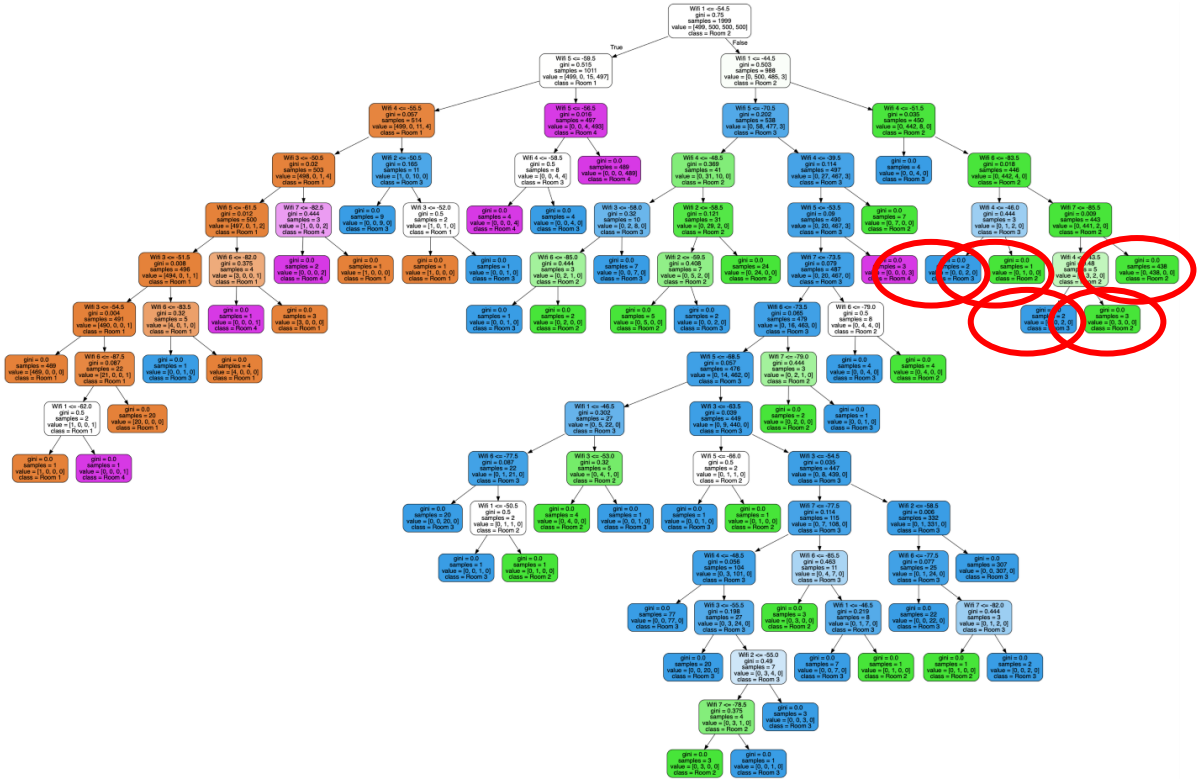
Common Hyperparameters (min_samples_leaves)

knowledgehut upGrad

4. min_samples_leaves

The minimum samples are required on a leaf node.

- A split takes place only if it leaves a min_samples_leaves at both branches.
- Applied to the leaves of the tree.



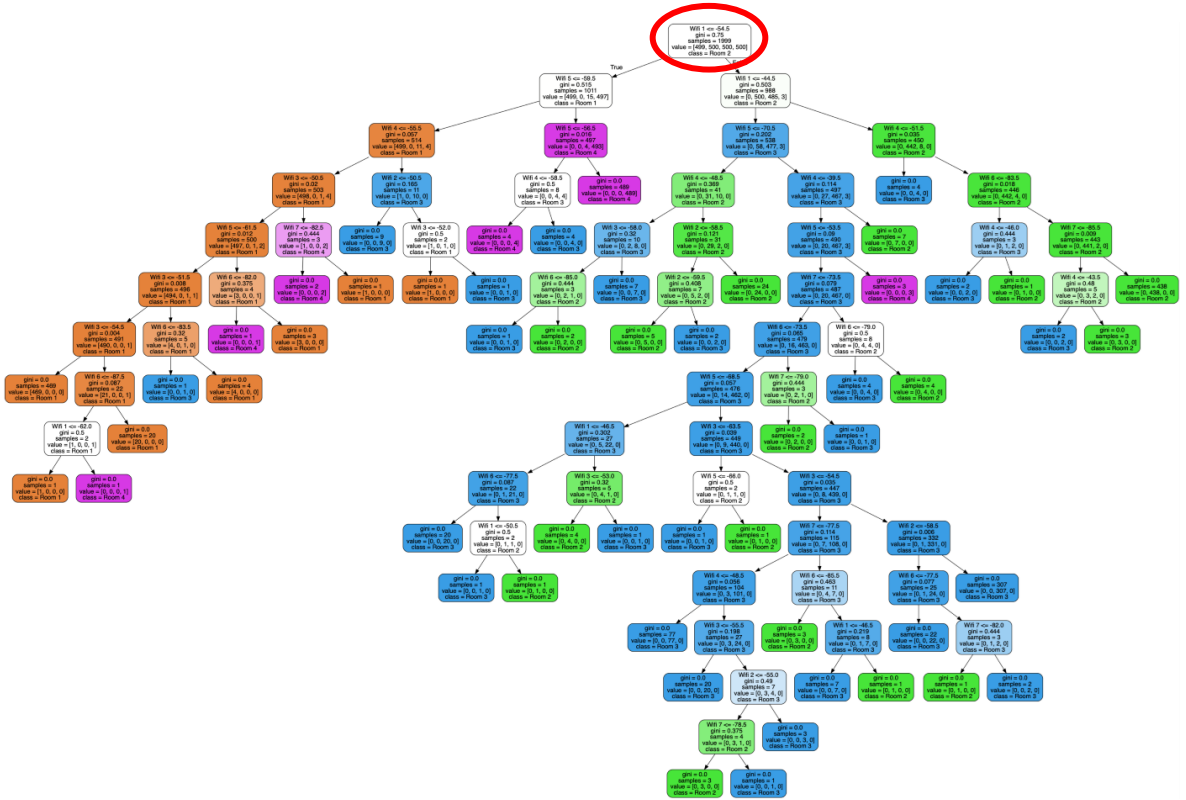
Common Hyperparameters (max_features)

knowledgehut upGrad

5. max_features

The maximum number of features considered for the split.

- All features are considered for split by default and can be limited using the subset of features.
- It prevents overfitting.



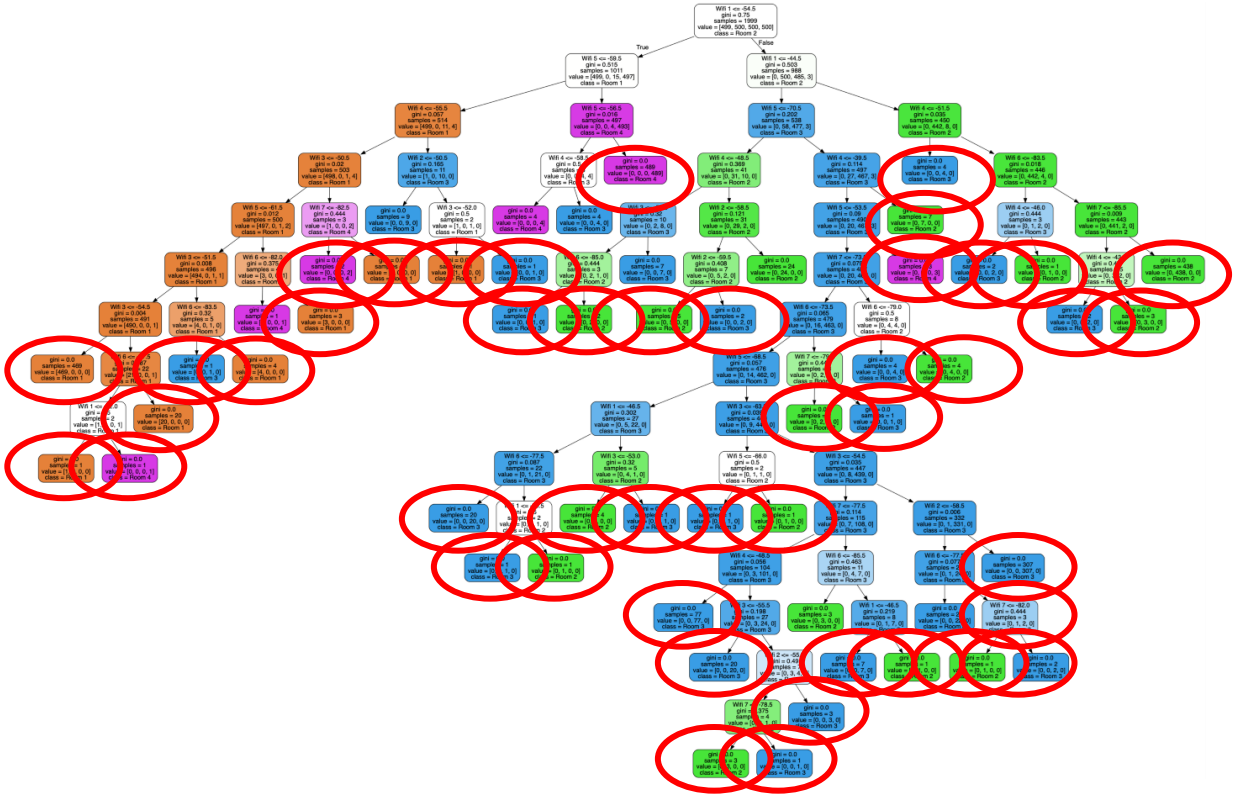
Common Hyperparameters (max_leaf_node)

knowledgehut upGrad

6. max_leaf_node

The maximum number of leaves generated.

- Similar to max_depth, but specifies the number of leaves.



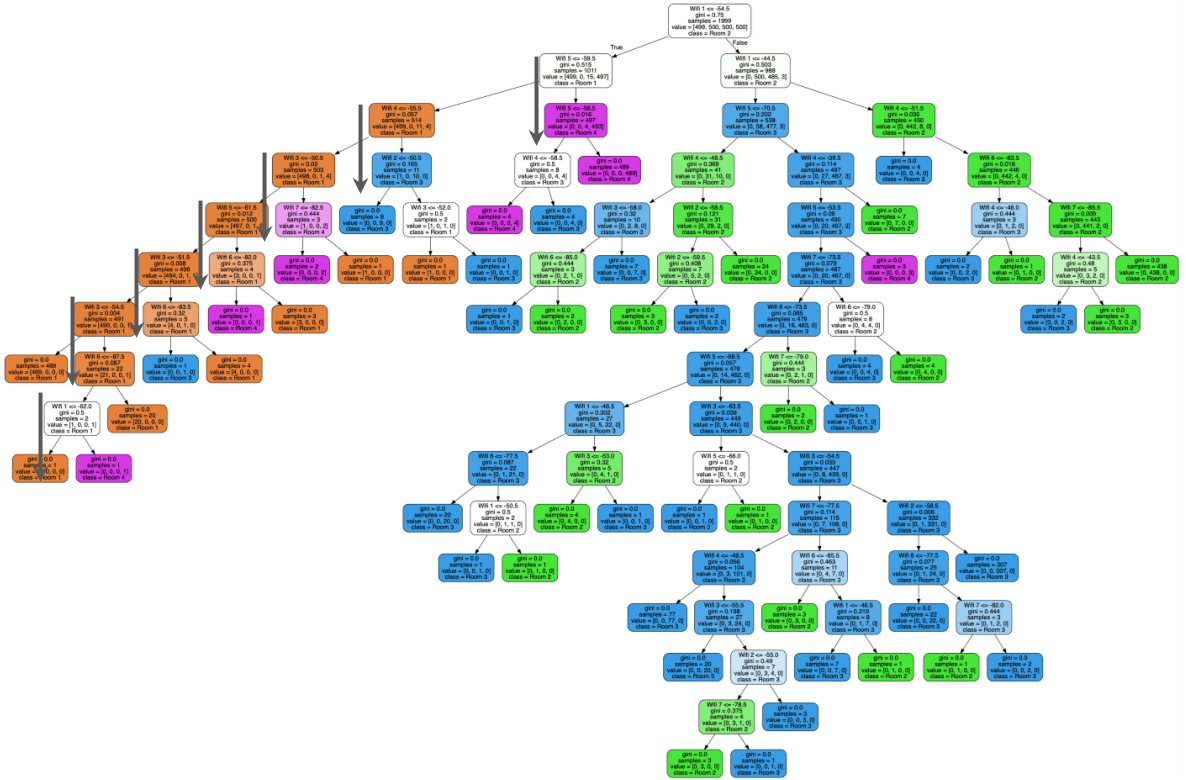
Common Hyperparameters (min_impurity_decreases)

knowledgehut upGrad

7. min_impurity_decreases

The threshold for early stopping of tree growth.

- A node splits if it induces a decrease of impurity greater than or equal to its value.

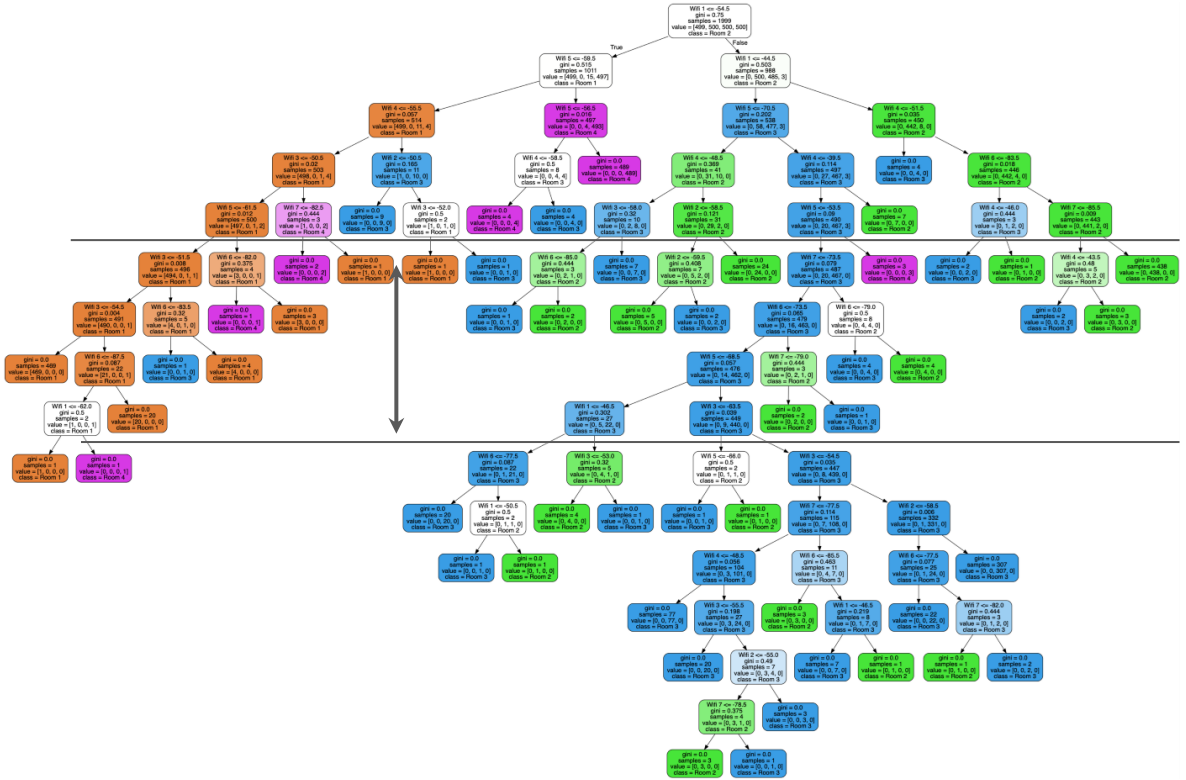


Common Hyperparameters (ccp_aplha)

knowledgehut upGrad

8. ccp_aplha

A post-pruning parameter.



Tuning Hyperparameter

knowledgehut upGrad

Grid Search

Grid Search is more efficient and faster.

Randomized Search

- Randomized search is more effective in some cases.
- With hyperparameters, the number of possible combinations increases, and the grid search becomes slow.
- It is better to start with the randomized search and then move to the grid search.

Thank you