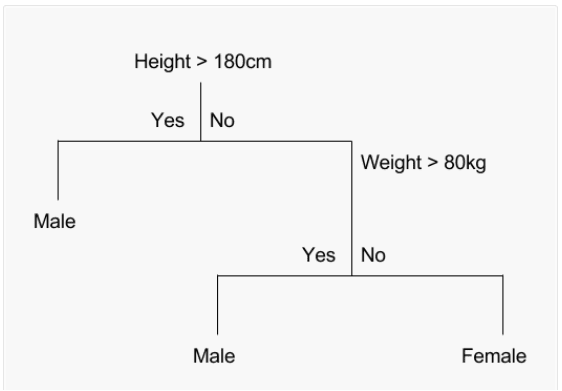
# What is a decision tree?

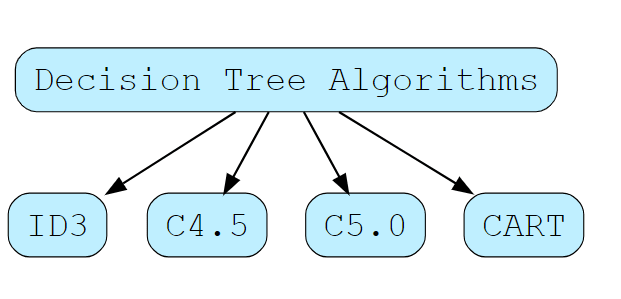
Decision Tree, as the name suggests, is constructed in a Tree manner, including a **root node**, **internal nodes**, and **leaf nodes**. Leaf nodes, also known as terminal nodes, give us the class of the instances falling in that terminal node, and the goal is to have homogeneous terminal nodes. Root Node refers to all the instances in the dataset. Interior nodes partition the set of instances. Once created, a tree can be navigated with a new row of data following each branch with the splits until a final prediction is made.Decision tree are branches emanating from nodes based on conditions. They are basically if-else statements which is used to make a decision. It forms a tree like structure.



The Decision Tree displayed above can be represented in the form of **if statements** as seen below.

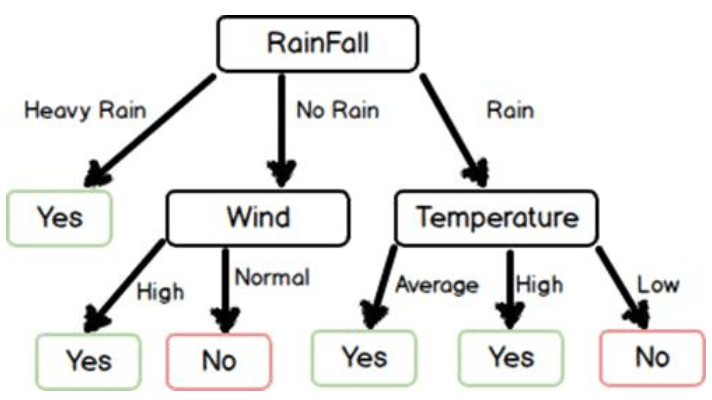
If Height > 180 cm Then Male  
 If Height <= 180 cm AND Weight > 80 kg Then Male  
 If Height <= 180 cm AND Weight <= 80 kg Then Female

# Decision Tree algorithms



#### **ID3**

ID3 stands for the Iterative Dichotomous 3 algorithm. It was developed by Ross Quinlan in 1986, and it was a predecessor of algorithms like C4.5. The algorithm works by finding categorical features for each node in the tree, which gives us the maximum information gain for the categorical targets.



#### C4.5

The C4.5 algorithm is a successor to the ID3 algorithm. It removed the restriction that features should be categorical to build the tree, unlike the ID3 algorithm. This algorithm is used by partitioning the continuous or numerical input features into a discrete set of intervals. It converts the trained trees into sets of if-then rules.

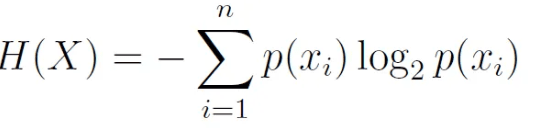
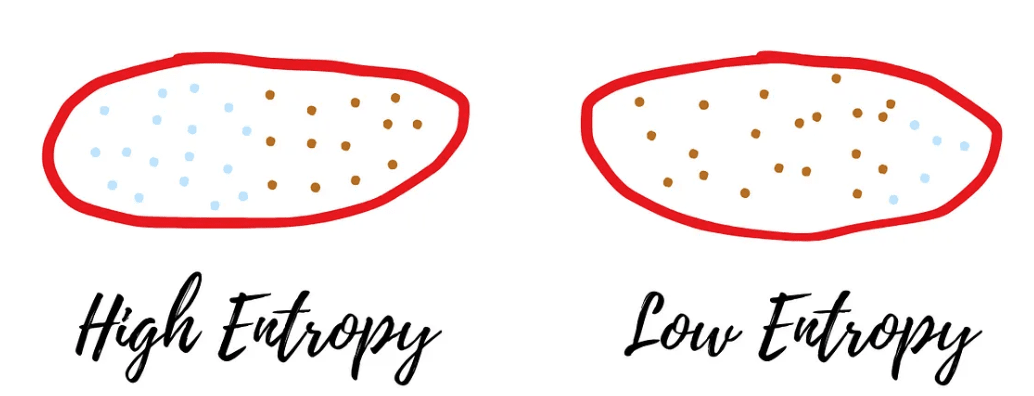
#### C5.0

C5.0 is the latest version released under a proprietary license. It uses less memory and builds smaller rule-sets than C4.5 while being more accurate.

#### CART

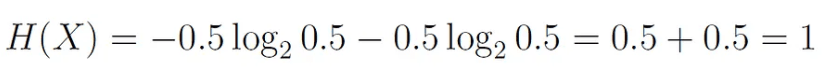
CART is an acronym for Classification and Regression Trees. It is very similar to the C4.5 algorithm. However, it provides an additional advantage because it also deals with Regression Problems and does not compute rule sets. It builds the Decision Tree using the features that yield the maximum information gain at each node. Constructing the Decision Tree involves many steps. These steps are divided into **induction** (training the model) and **pruning** of the constructed tree. Pruning is the step of removing an unnecessary structure from the Decision Tree to avoid over fitting.

**Entropy**: Entropy of a random variable is a measure of the uncertainty in the variable’s possible outcomes. The more uncertain we are about the value of the variable, the higher its entropy.



**Scenario 1: Entropy is highest**

fair coin , p(h) = 0.5, p(t)=0.5,

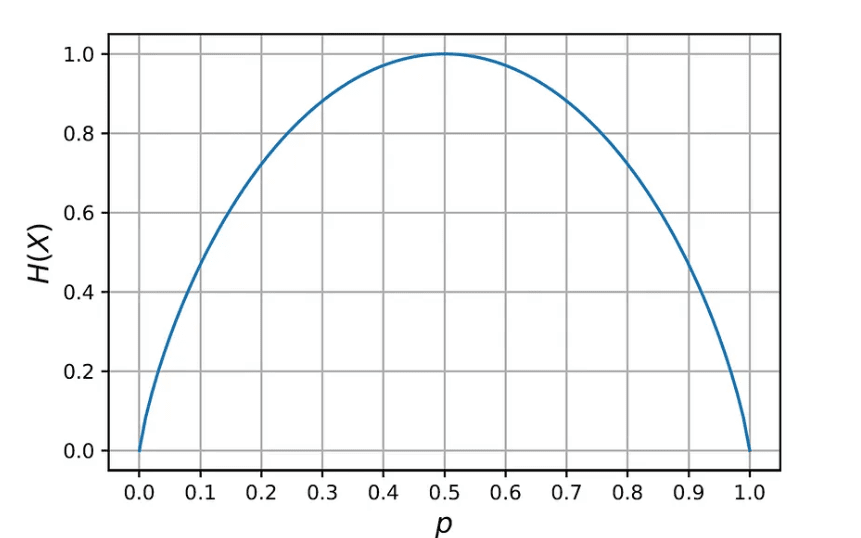


**Scenario 2: Entropy became smaller**

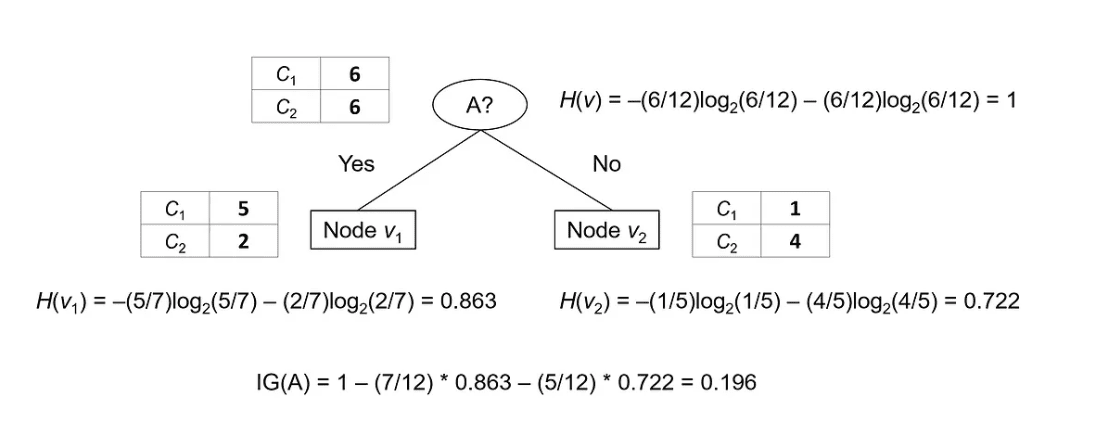
biased coin , p(h) = 0.8. p(t)=0.2,

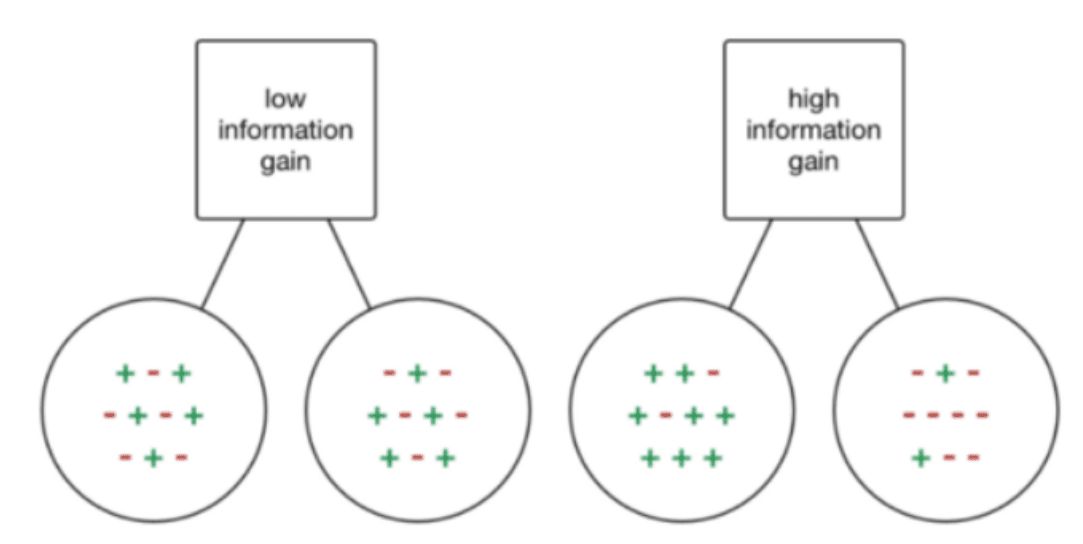


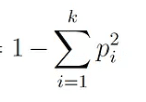


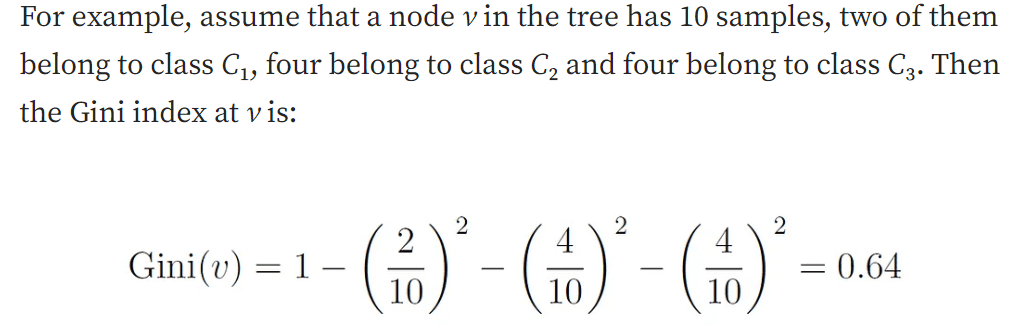


Information gain : inversely proportional to entropy. How much information is held? The goal is to maximize information gain.

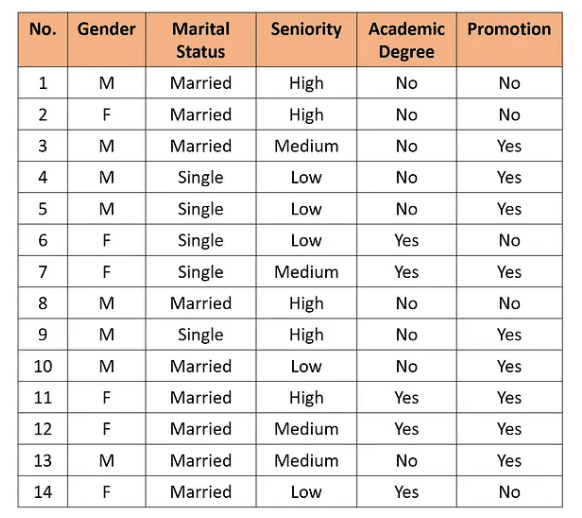




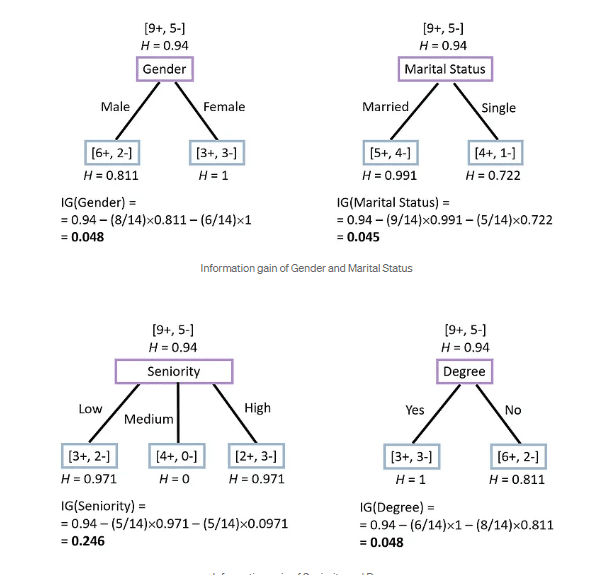
Gini Index : Measure of impurity. It measures how often a randomly chosen sample at a node is incorrectly labeled

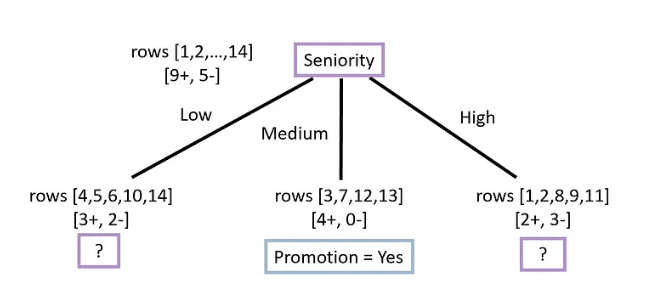


**Exercise:** Our objective is to predict whether an employee will get a promotion based on four attributes

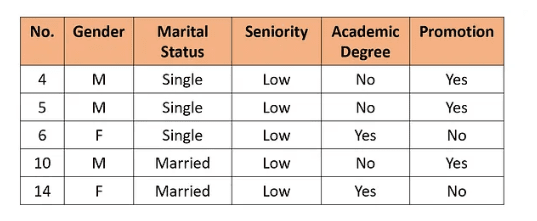


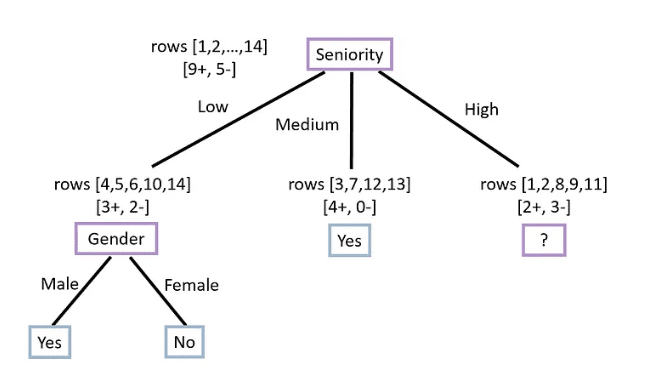
Step 1: we need to select the attribute for the first split at the root node. To that end, we compute the information gain for each of the four attributes





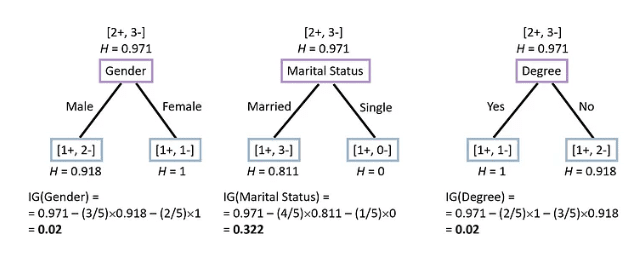
Step 2: Left Node

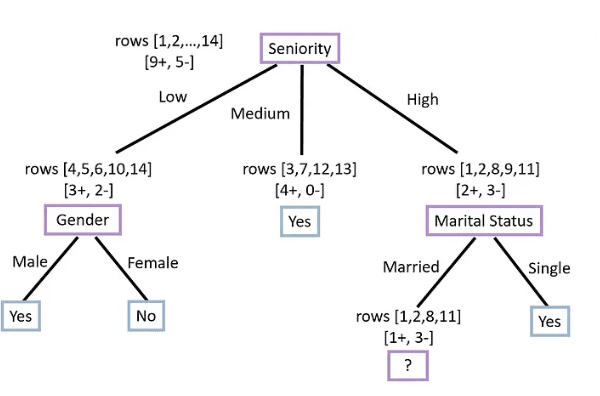




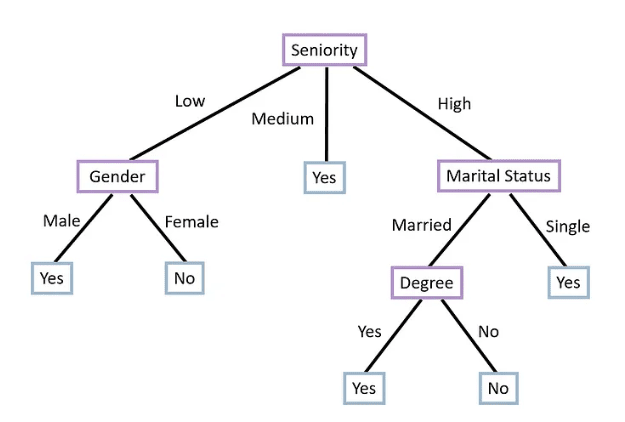
Step 3: Right Node



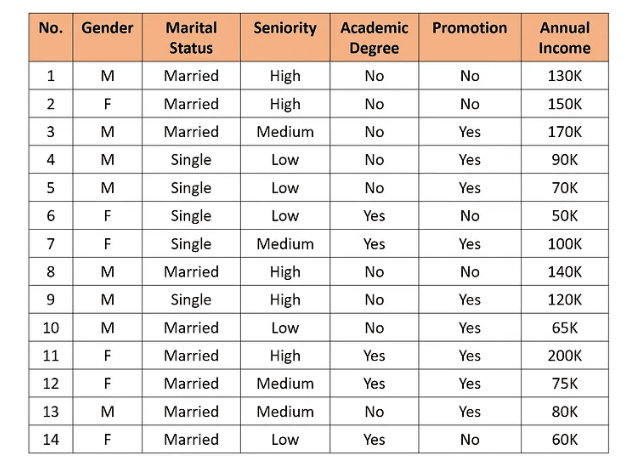


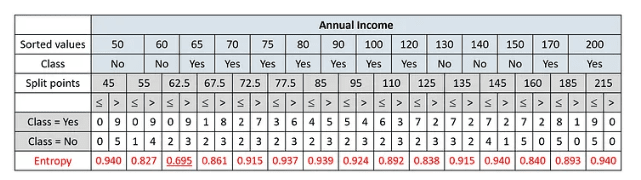


Step 4:

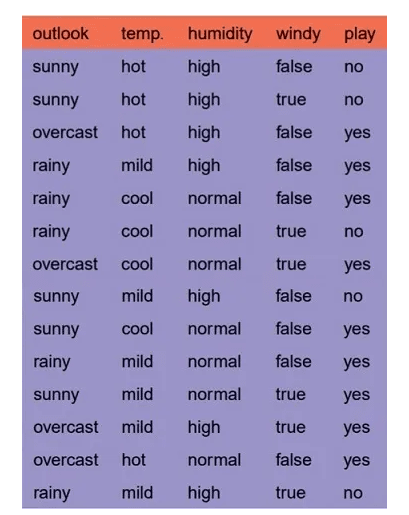


What if there was a continuous feature?





Another Exercise:

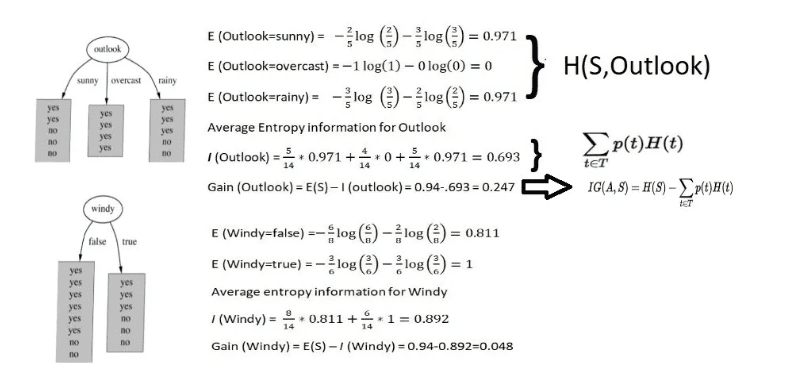


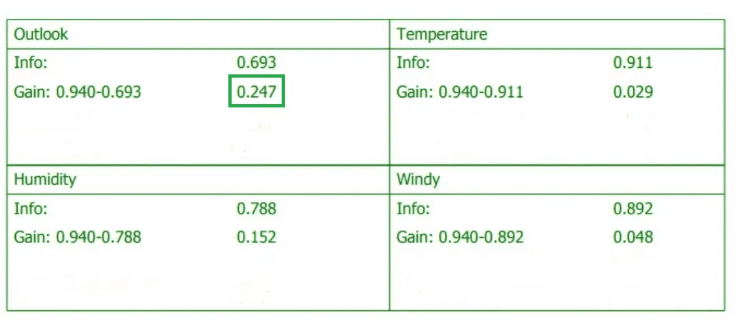
Compute Entropy:

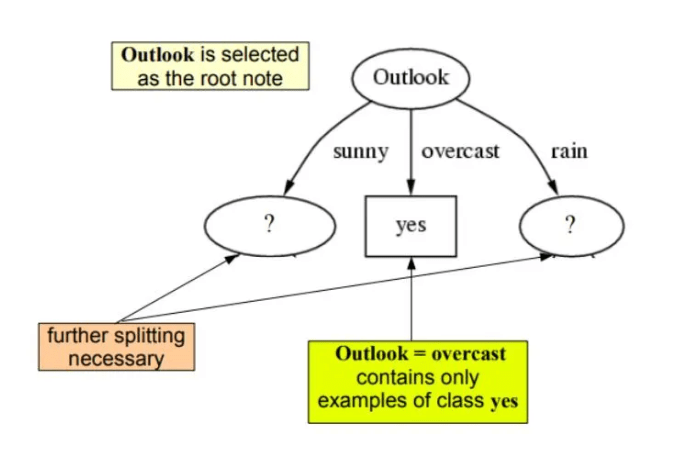
Information gain:

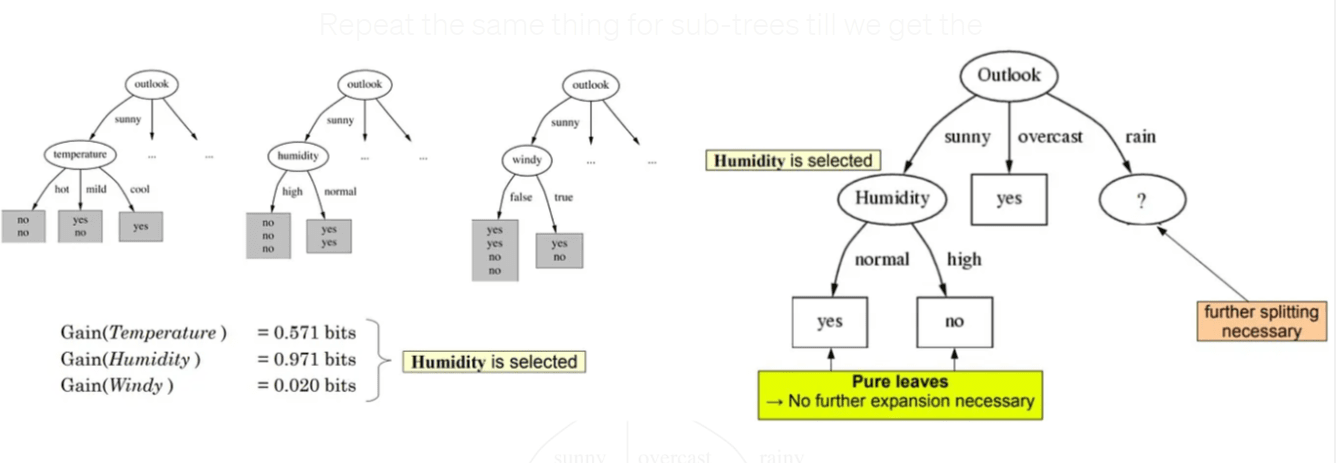


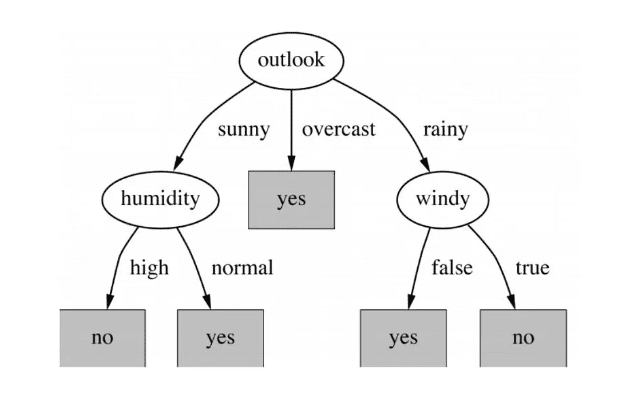
Calculate Entropy for the features:





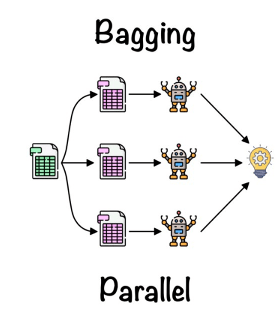


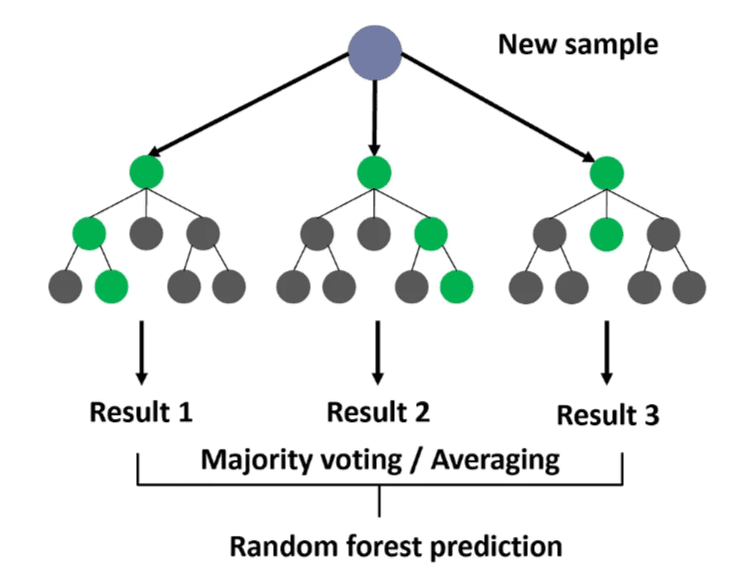


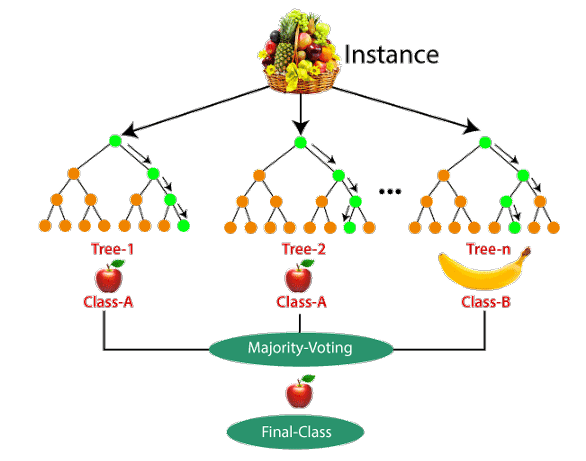


Random Forrest

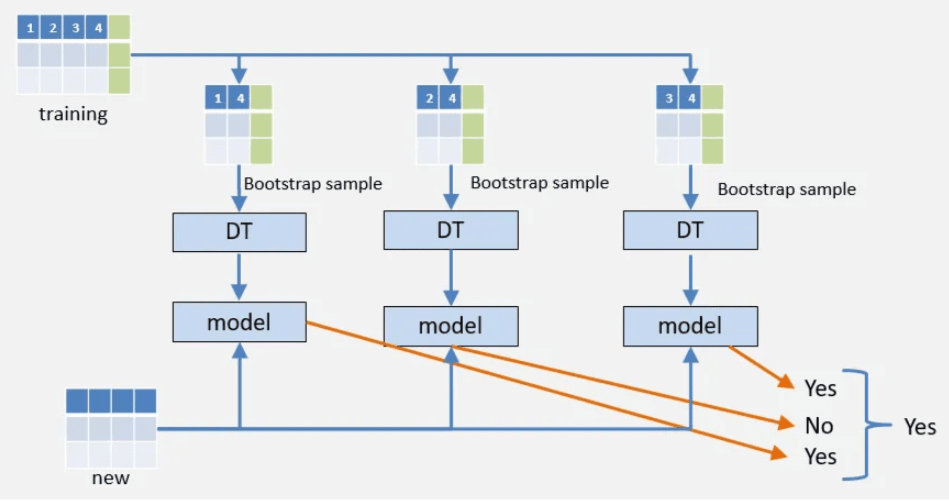
Bagging is a simple and a compelling ensemble method. It is a general procedure that can be used to reduce our model’s variance.







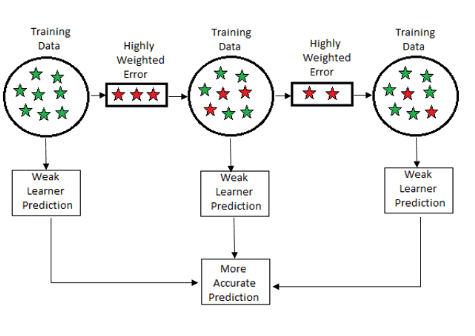
Sampling of the original data is done by a process called bootstrapping - with and without replacement



Gradient Boosting

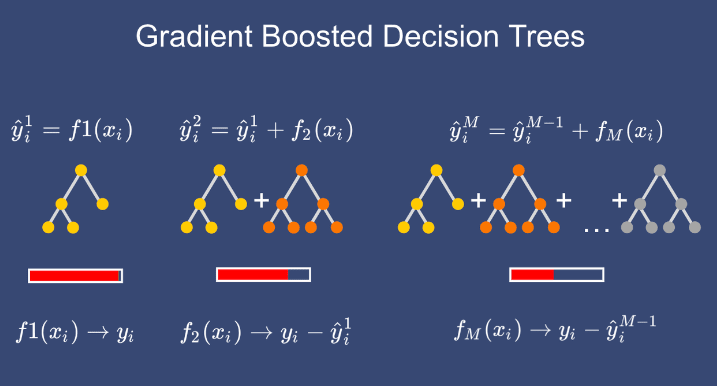
Gradient boosting is a powerful machine learning technique used for regression and classification tasks, among others. It builds a prediction model in the form of an ensemble of weak prediction models, which are typically simple decision trees. The key idea behind gradient boosting is to add new models to the ensemble sequentially.

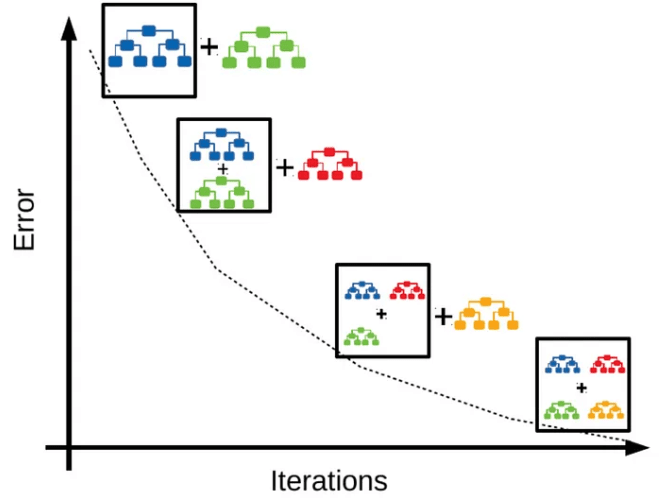
At each step, a new model is trained to predict the residuals (or "errors") of the current ensemble. The residuals are calculated as the difference between the actual output and the prediction of the current ensemble. By fitting each new model to the residuals, we allow each new model to focus on the mistakes of the current ensemble.



Here's a simplified version of the process:

1. Fit a decision tree to the data: F1(x) = y
2. Fit the next decision tree to the residuals of the previous: h1(x) = y - F1(x)
3. Add this new tree to our algorithm: F2(x) = F1(x) + h1(x)
4. Fit the next decision tree to the residuals of F2: h2(x) = y - F2(x)
5. Add this new tree to our algorithm: F3(x) = F2(x) + h2(x)
6. Continue this process until a stopping condition is met, such as a maximum number of trees or when the improvement in error becomes negligible.





Exercise:

