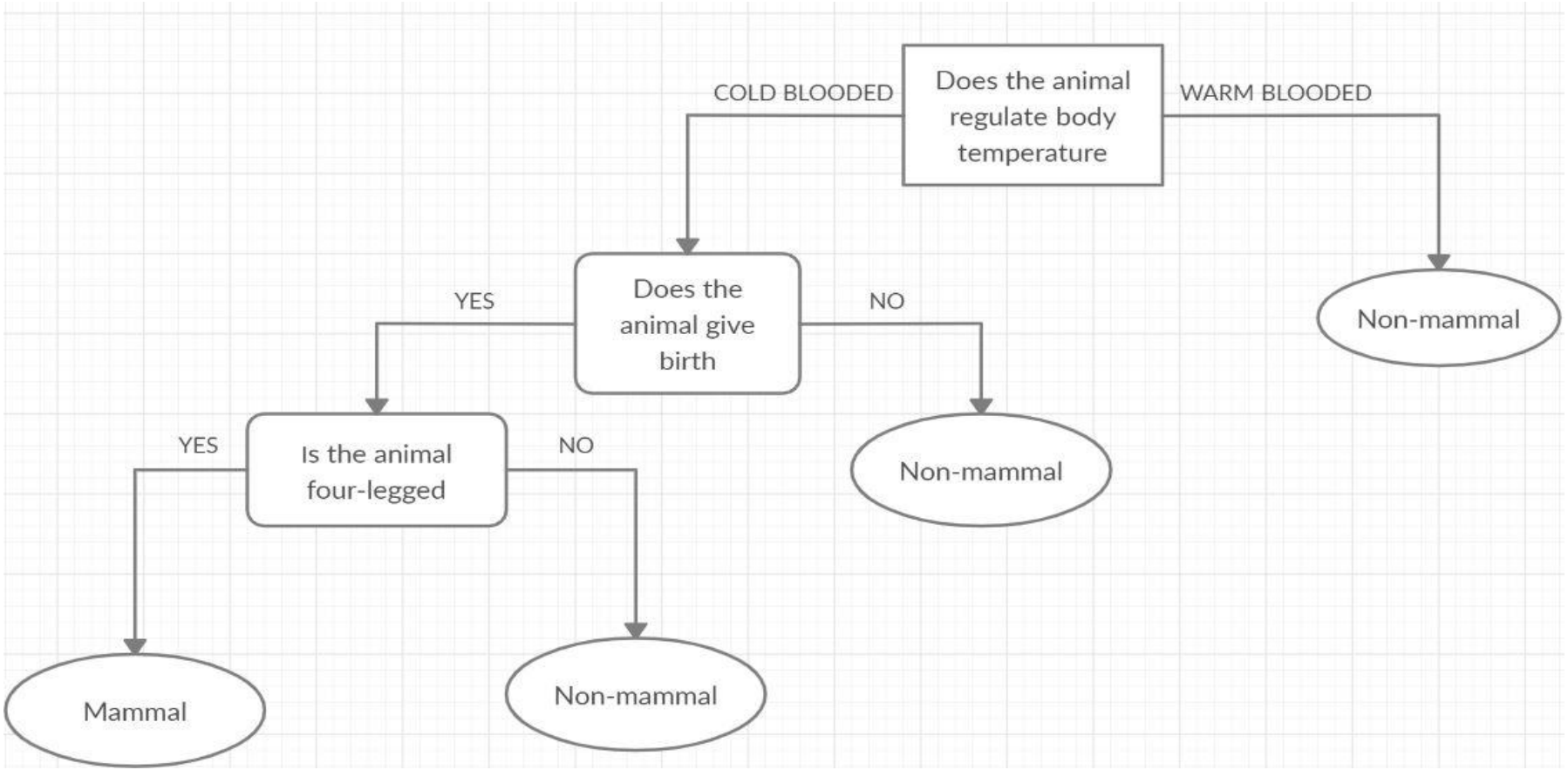
DECISION TREES

Introduction

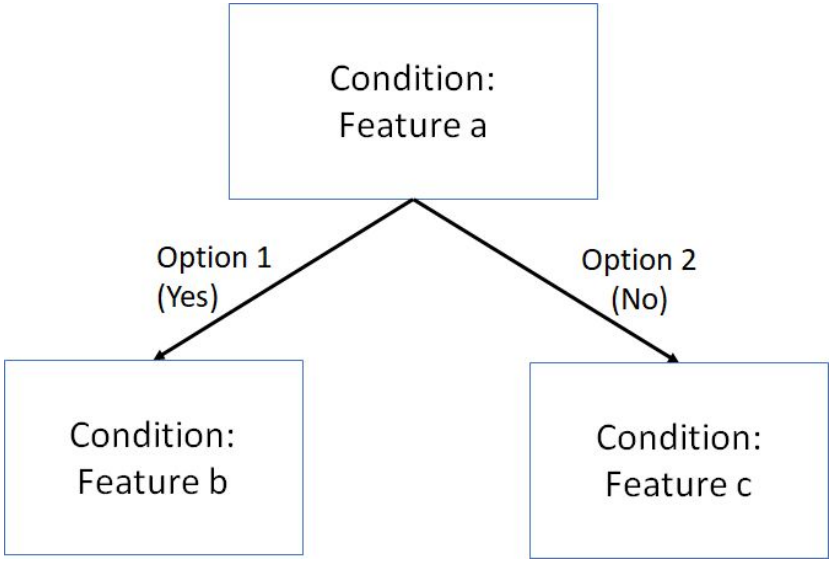
- Decision tree is one of the most powerful supervised learning tools for classification and regression.
- Decision tree has a flowchart like tree structure used to visualize the decision making process by mapping out different courses of action, as well as their potential outcomes.
- As they can be used for both classification and regression, they are collectively called Classification and Regression Trees (CART).

Example: A simple decision tree to classify an animal as mammal or non-mammal



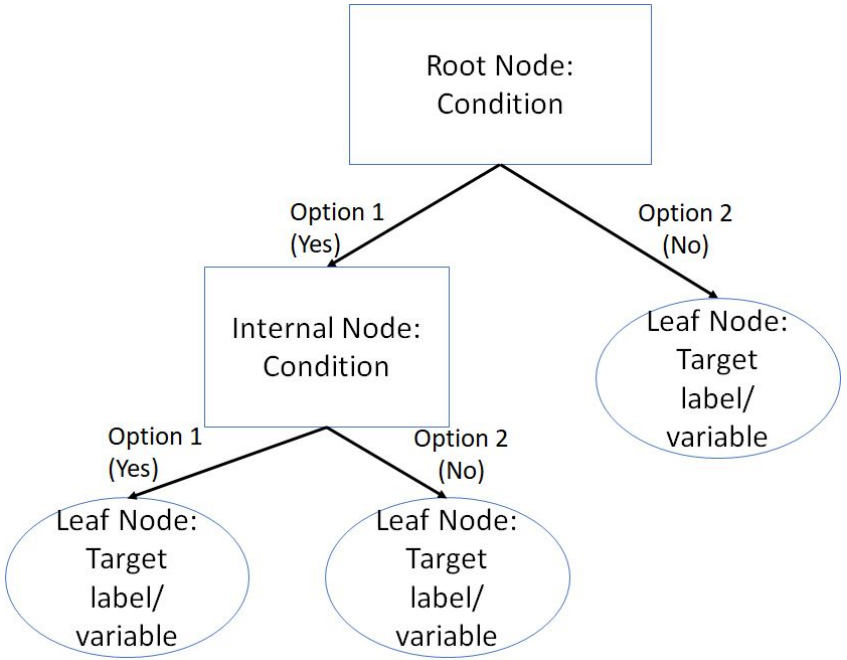
Components of a Decision Tree

- **Nodes:** Represents a condition on a feature based on which tree splits into branches
- **Branches:** Represent different options that are available based on the nodal condition
- Generally split is binary (yes/no) but multiway splits are also possible



Types of Nodes

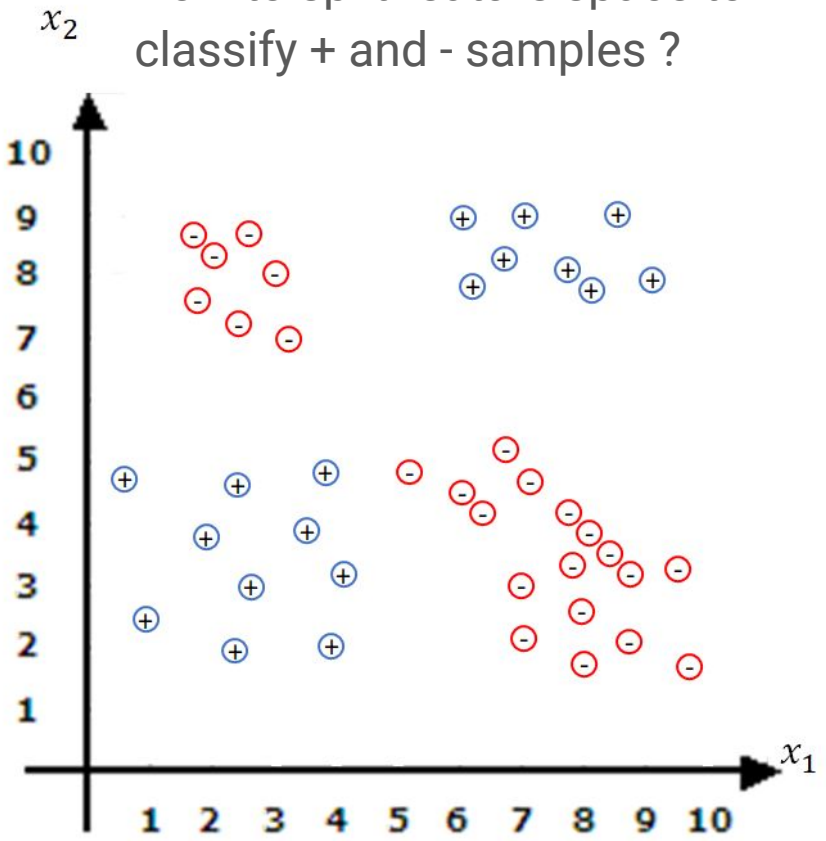
- **Root Node:** Top-level node in the decision tree where the first conditional split happens
- **Internal Node:** Node with incoming and outgoing branches having conditional splits
- **Leaf Node:** Terminal of a branch that doesn't split anymore and represents a target label or target variable



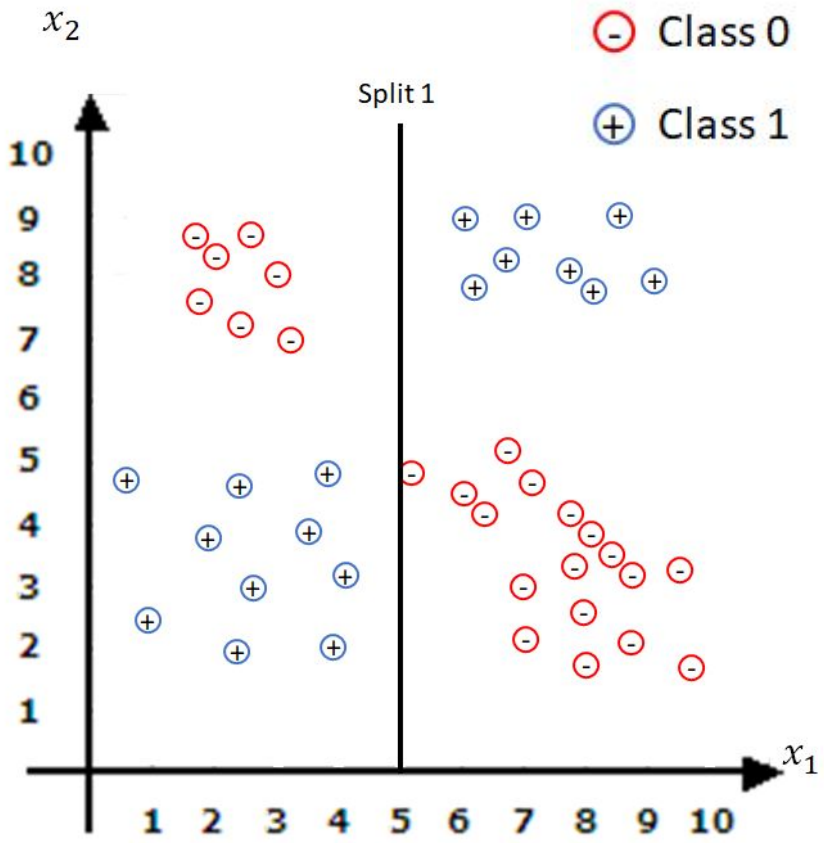
Principle of a Decision Tree

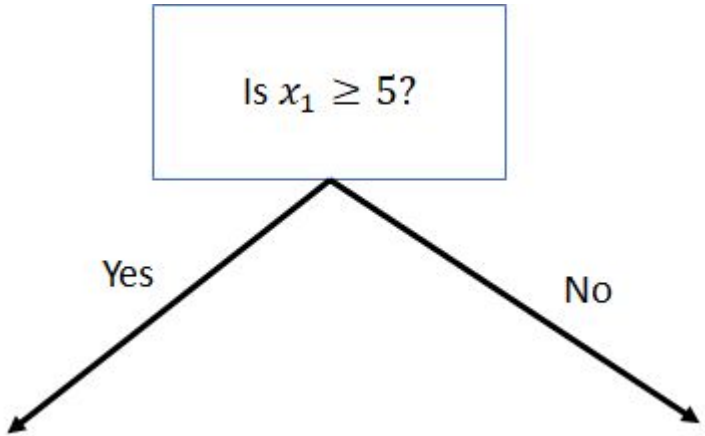
- Feature space is split into smaller areas based on different features
- Leaf nodes should correspond to distinct, non-overlapping regions
- Each region should contain only samples belonging to one particular class or atleast significant majority should belong to one class

How to split feature space to classify + and - samples ?



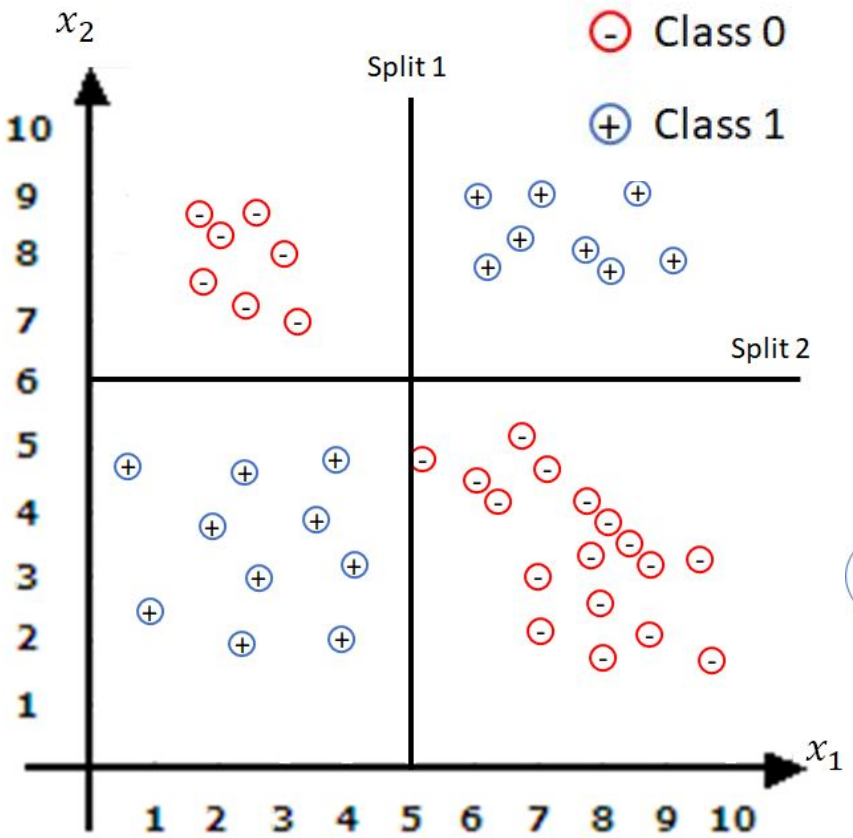
Principle of a Decision Tree: Illustration

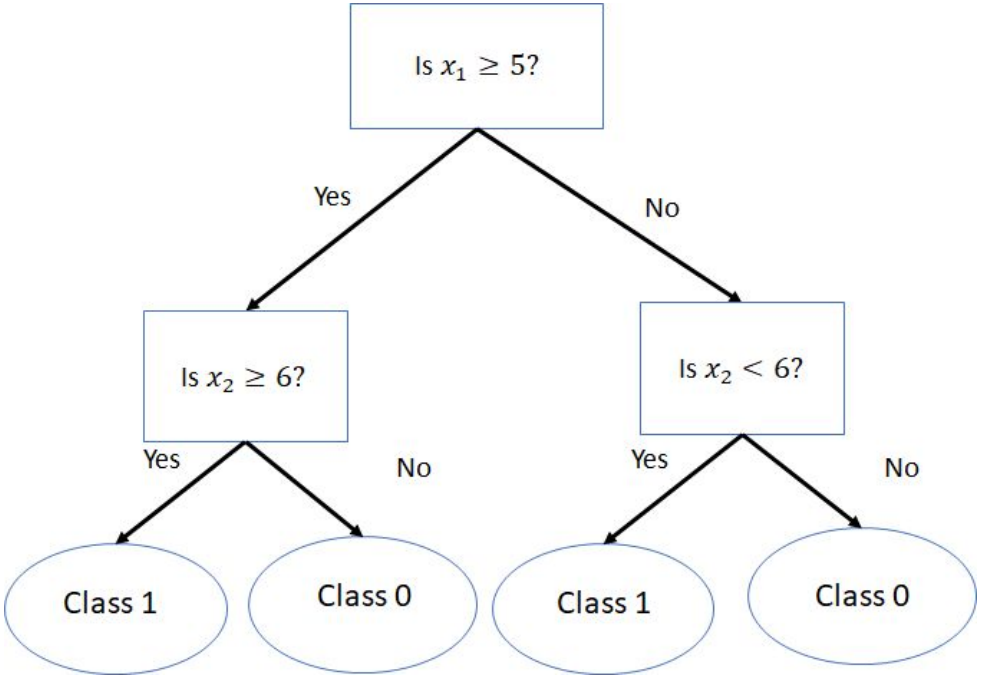




After split 1, the regions still contain samples of both classes

Principle of a Decision Tree: Illustration





After split 2, every distinct region has samples from either class 1 or 0

Building a Decision Tree

- Questions to be answered while building a decision tree:
  - How to select the feature and condition to split upon at a node ?
  - When to stop splitting or fixing a node to be the leaf node ? (Stopping Criteria)

Building a Decision tree

1. **Feature and Condition to split:** Features and conditions can be selected randomly but for efficient split, following are some measures used to select a feature to be split at a node
    - Entropy or Gini Index for Clasification
    - Squared error for Regression

Feature split which has maximum entropy or Gini index is selected or feature split which minimises squared error is selected (For more info on entropy and Gini index refer :

https://towardsdatascience.com/gini-index-vs-information-entropy-7a7e4fed3fcb)

2. Split the tree based on the attribute selected which divides the feature space in multiple regions

3. Perform further splits based on the measures given in step 1

4. Repeat the splits until the stopping criteria is reached

5. For classification trees , the stopping criteria is generally based on the purity of regions after split i.e., whether the region has samples belonging to same class or not

6. For regression trees, the stopping criteria is generally maximum depth of the tree that is pre-defined

Advantages

- Easy to interpret and explain
- They do not require any domain knowledge for feature selection
- Can easily handle both numeric and categorical data
- Does not require any scaling
- The type of relation (linear or non-linear) between features and targets does not affect tree performance

Disadvantages

- Decision tree learners can create complex tree that overfit the data and do not generalise well to unseen data
- Trees can be very non-robust. In other words, a small change in the data can cause a large change in the tree built
- Decision trees can be biased if some classes dominate. Therefore balanced datasets are recommended
- Greedy algorithms cannot return globally optimal decision tree

Minimise Over-Fitting

- To minimise overfitting, a balance should be ensured between depth of a the tree and purity of leaf nodes
- Two options to minimize overfitting:
  - Early Stopping of training by restricting tree depth or the number of leaf nodes in the tree
  - Pruning a tree based on some cost function after building a large tree

For more details on pruning refer to :
https://en.wikipedia.org/wiki/Decision_tree_pruning

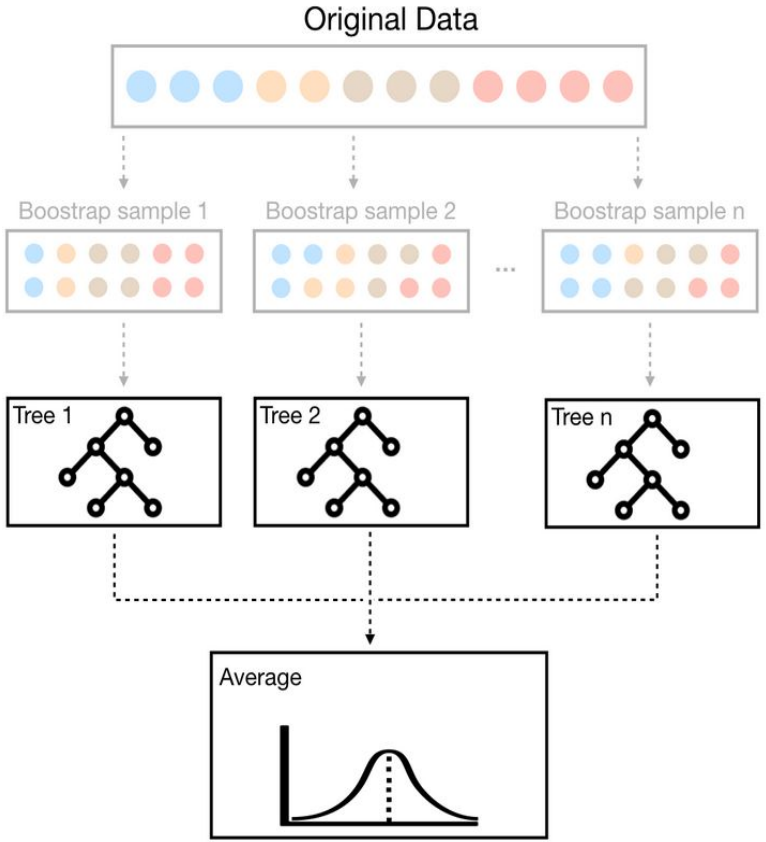
Tuning in Decision Trees

- In practice, when learning decision trees, there exist many hyperparameters which need to be tuned for better learning
- They include parameters which affect the decision tree complexity, stopping criteria and pruning
- For more details on hyper-parameter tuning in decision trees one may refer to : https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680

Tree Bagging

Bagging Algorithm

- Bagging is an ensemble classifier that consists of many decision trees
- Create multiple datasets from the given dataset by sampling data points with replacement
  (also known as ‘bootstraping’)
- Fit a decision tree to each re-sampled dataset
- On a test set, average of the predictions of all the trees is taken to make a prediction

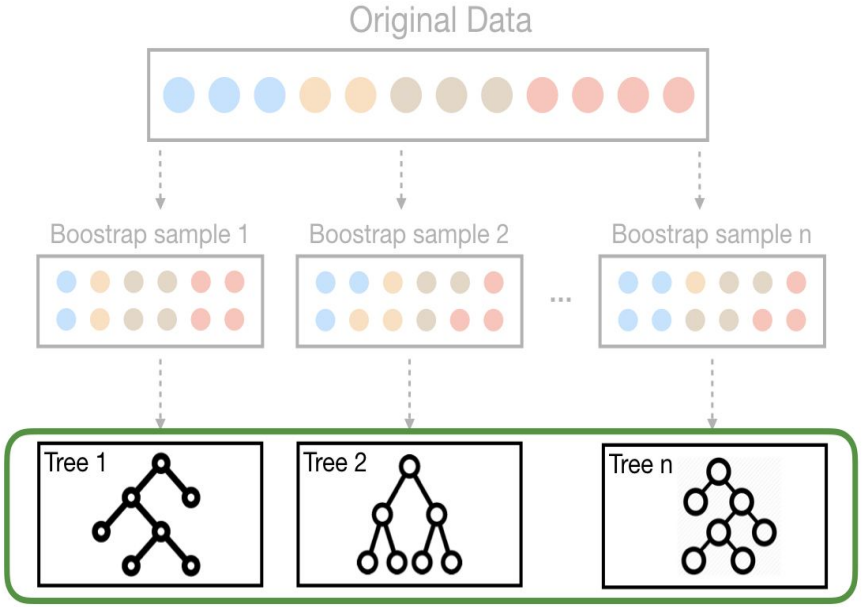


Source: https://bradleyboehmke.github.io/random-forest-training/slides-source.html#33

Random Forest

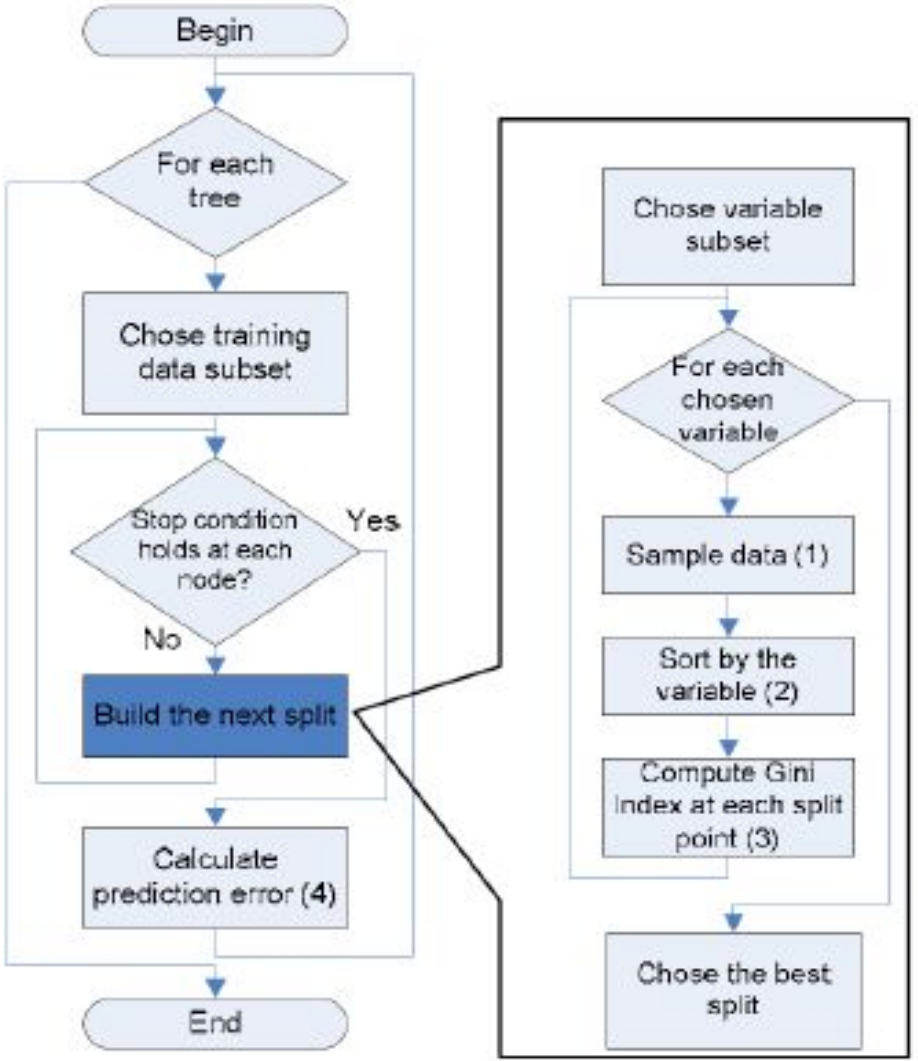
Random Forests

- Random forest is an ensemble classifier that consists of many decision trees similar to tree bagging
- However, each time a split is to be performed, the search for the split feature is limited to a random subset of m of the p features



Source: https://bradleyboehmke.github.io/random-forest-training/slides-source.html#33

Flowchart of Random Forest Training Algorithm



Advantages

- One of the most accurate learning algorithms available.
- It generalises well to unseen data and generally does not overfit
- It is robust to outliers
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- Not much of preprocessing of data is required

Disadvantages

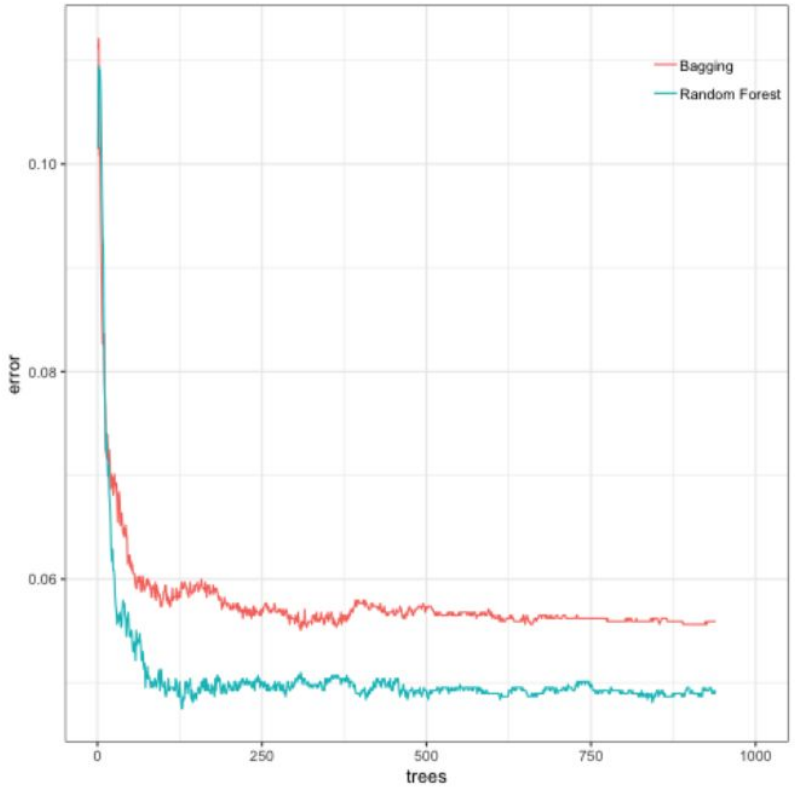
- Slow on large datasets
- Less interpretable compared to individual decision trees
- They are found to not very effective for regression tasks

Tree Bagging & Random Forests - Common Points

- Splits are chosen according to a class purity measure:
  - E.g. squared error (regression), Gini index or entropy (classification)
- How many number of decision trees to create?
  - Sample datasets & build trees until the validation error no longer decreases
- How to select m features while training random forest?
  - Generally, for classification trees, m = p/3
  - Generally, for regression trees, m = sqrt(p)

Bagging Vs Random Forests

- Bagging introduces randomness into rows (samples) of data
- RF introduced randomness in both rows (samples) and columns (features) of data
- Therefore, random forests create more diverse set of trees and result in lower prediction error



Prediction Error: Bagging Vs RF

Source: https://bradleyboehmke.github.io/random-forest-training/slides-source.html#33