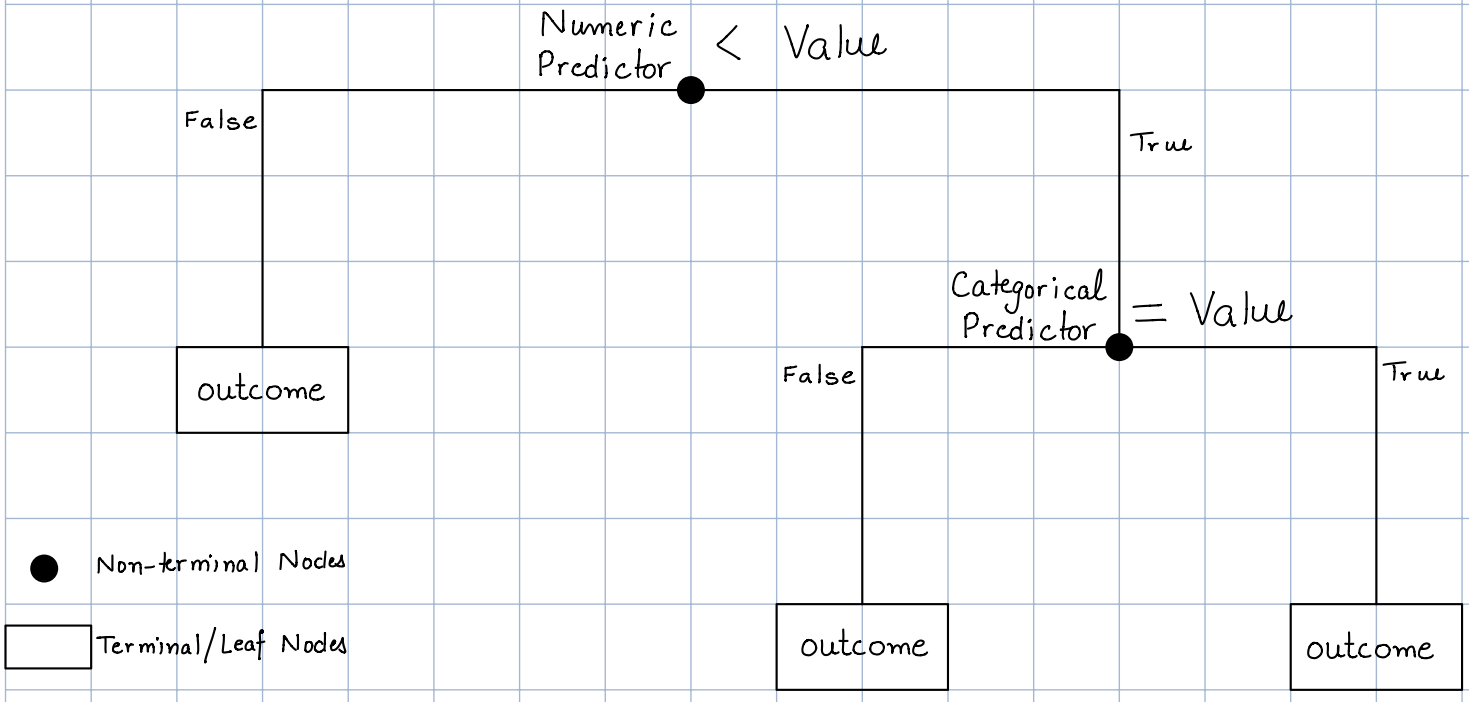
DECISION TREES

Model



A leaf node contains a region.

Region = Training data that satisfies the path from root → leaf

Prediction = Find region corresponding to a test case.

Return aggregate of response present in the region.

(mean for regression

mode or class proportion for classification)

Fitting the Model

Goal: Find non-overlapping high-dimensional rectangular regions $R_1, \ldots, R_J$

How: Use recursive binary splitting (greedy approach)

Recursive binary splitting splits incoming data into two regions R_1 and R_2:

1. Select a feature X_j and a cutpoint value v such that it minimizes splitting criterion function.

Regression:

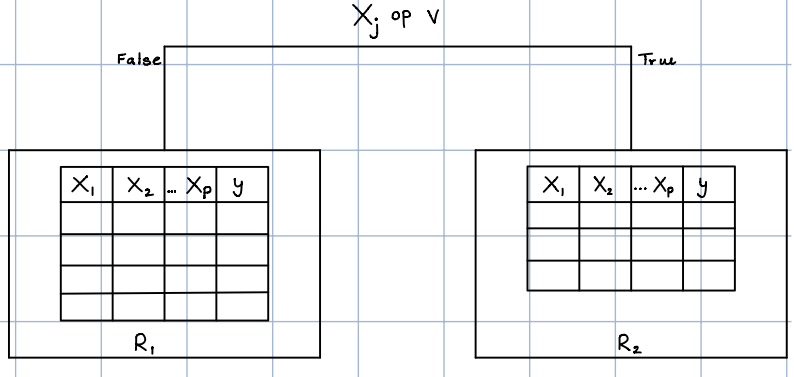
$$\sum_{i \in R_1(j,v)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i \in R_2(j,v)} \left(y_i - \hat{y}_{R_2}\right)^2$$

Classification:

$$\text{Gini} = \sum_{K=1}^{K} \hat{P}_{KR_1}\left(1 - \hat{P}_{KR_1}\right) + \sum_{K=1}^{K} \hat{P}_{KR_2}\left(1 - \hat{P}_{KR_2}\right)$$

$$\text{Cross-entropy} = -\sum_{K=1}^{K} \hat{P}_{KR_1} \log \hat{P}_{KR_1} - \sum_{K=1}^{K} \hat{P}_{KR_2} \log \hat{P}_{KR_2}$$

where $\hat{P}_{KR_i}$ is the proportion of observations that belong to class K in region R_i.



2. Keep splitting until a criteria is met such as
   - Observations in resulting region are greater than a threshold, or
   - Resulting splitting criterion function is greater than a threshold.

- Pruning: Regularization for trees.

  Overfitting occurs when tree is complex

  Extreme case: Number of leaves = Number of observations.

Let T_0 be the tree where number of leaves $|T|$ equal number of observations in training set.

Choose $T \subset T_0$ such that the

$$\text{Cost (across all regions)} + \alpha |T|$$

is minimized.

Regression cost:

$$\sum_{m=1}^{|T|} \sum_{i \in R_m(j,v)} \left(y_i - \hat{y}_{R_m} \right)^2$$

Classification cost:

$$\text{Gini} = \sum_{m=1}^{|T|} \sum_{K=1}^{K} \hat{P}_{KR_m} \left(1 - \hat{P}_{KR_m}\right)$$

$$\text{Cross-entropy} = -\sum_{m=1}^{|T|} \sum_{K=1}^{K} \hat{P}_{KR_m} \log \hat{P}_{KR_m}$$

α provides a tradeoff between quality of fit (cost) and tree complexity ($|T|$).

RANDOM FOREST

- Bootstrap Aggregation (Bagging):
  - Single tree has high variance and low bias.
  - How to reduce variance? ⇒ Use bootstrap!

Bootstrap:
- Sample B datasets from training set (each has n rows)
- Fit B (deep) trees independently (in parallel)

Aggregation:
- While predicting a test case, average out the predictions from B trees — mean for regression, mode for classification.

- Out-of-bag (OOB) Error:
  - Compute prediction for each observation i using trees that did not have i in their bootstrapped training set.
  - OOB = Use the above predictions to compute MSE or classification error.

- Feature Importance:
  - Compute mean decrease in RSS or Gini / Entropy across B trees.
  - The higher this metric, the more important a feature is.

- Random Forest:
  - Bagging + randomly sample $m = \sqrt{p}$ predictors at each split in each tree.
  - Decorrelates trees and reduces more variance than bagging.
  - The higher B, the better ⇒ does not lead to overfitting.

Boosting

- Fit B trees sequentially wher each tree is a weak learner (shallow tree with few nodes) with high bias.

- Each tree is fit on the remaining residuals from previous tree (in case of regression; classification out of scope)

Regression Algorithm

1 $\hat{f}(X) = 0$, $r = y$

2 For $b = 1$ to B:

a. Fit a shallow tree with d splits on (X, r).

b. Update the predictions $\hat{f}(X)$

$$\hat{f}(X) \leftarrow \hat{f}(X) + \lambda \hat{f}^b(X)$$

c. Update residuals

$$r \leftarrow r + \lambda \hat{f}^b(X)$$

3. Final model

$$\hat{f}(X) = \sum_{b=1}^{B} \lambda \hat{f}^b(X)$$

- Boosting hyperparameters:
  - Number of trees B : Very large B can overfit.
  - Shrinkage (learning) rate λ : Typically 0.01 or 0.001
  - Depth of each tree d : Typically 1 (leads to an additive model) or other small value.

Use cross-validation to tune all hyperparameters.