## Random Forest Classifier - Overview

Random Forest is an ensemble of decision trees trained on bootstrap samples, where each split considers a random subset of features. This decorrelates trees and reduces variance. For classification, predictions are obtained by majority vote across trees.

This example uses Random Forest (scikit-learn via reticulate) to classify the Iris dataset. Workflow: split train/test, train, predict, and evaluate (classification metrics).

Prerequisites - R packages: daltoolbox, daltoolboxdp - Python accessible via reticulate (scikit-learn installed)

# Install required packages (if not already installed)  
#install.packages("daltoolboxdp")

# Loading packages  
library(daltoolbox)  
library(daltoolboxdp)

# Loading Iris dataset  
iris <- datasets::iris

# Training and evaluation with Random Forest  
  
slevels <- levels(iris$Species) # target variable levels  
  
set.seed(1)  
sr <- sample\_random() # stratified random sampling  
sr <- train\_test(sr, iris) # split data  
iris\_train <- sr$train  
iris\_test <- sr$test  
  
# Create numeric label for scikit-learn (keeping "Species" as original target)  
iris\_train$species\_encoded <- as.integer(as.factor(iris\_train$Species))  
iris\_train\_label <- iris\_train[, !names(iris\_train) %in% "Species"]  
  
# 1) Train  
model <- skcla\_rf("species\_encoded", slevels)  
model <- fit(model, iris\_train\_label)  
  
# 2) Evaluate on train  
train\_prediction <- predict(model, iris\_train\_label)  
iris\_train\_predictand <- adjust\_class\_label(iris\_train[, "Species"]) # original labels  
train\_eval <- evaluate(model, iris\_train\_predictand, train\_prediction)  
print(train\_eval$metrics)

## accuracy TP TN FP FN precision recall sensitivity specificity f1  
## 1 1 39 81 0 0 1 1 1 1 1

# 3) Evaluate on test  
iris\_test$species\_encoded <- as.integer(as.factor(iris\_test$Species))  
iris\_test\_label <- iris\_test[, !names(iris\_test) %in% "Species"]  
test\_prediction <- predict(model, iris\_test\_label)  
  
iris\_test\_predictand <- adjust\_class\_label(iris\_test[, "Species"])  
test\_eval <- evaluate(model, iris\_test\_predictand, test\_prediction)  
print(test\_eval$metrics)

## accuracy TP TN FP FN precision recall sensitivity specificity f1  
## 1 0.9666667 11 19 0 0 1 1 1 1 1

References - Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.