Statistical Learning. Task 1

Prediction with Trees and Ensemble-based trees

Alex Sanchez

2025-03-03

Contents

Objective

The goal of this exercise is to predict the flowering type of a plant (fast or slow) based on its genotype and morphological characteristics using tree-based classification models.

Dataset

We have data from 697 plants, where the following information has been recorded:

- **Morphological Factors**: Flowering time (in days), length (in cm), and coverage area (in cm^2).
- **Genetic Factors**: Genotypes of 19 genes with three possible states:
  - 0: Homozygous dominant
  - 1: Heterozygous
  - 2: Homozygous recessive

The data is stored in the following files:

- `data.csv`: Contains the 19 genotypes, length, and coverage of each plant (dimensions: 697×21).
- `flowering_time.csv`: Contains the flowering time in days for each plant (dimensions: 697×1).

Target Variable Definition

The response variable is defined as:

- **Fast flowering**: Less than or equal to 40 days (encoded as 0).
- **Slow flowering**: More than 40 days (encoded as 1).

Procedure

1. Data Preprocessing

- Load and merge both datasets into a single data frame.
- Randomly take a sample of 600 rows so that each group's dataset is unique.
- Transform the flowering time variable into a categorical binary variable according to the definition above.
- Split the dataset into a **training set (2/3)** and a **test set (1/3)**, using `set.seed(12345)` (or its python equivalent) to ensure reproducibility.

2. Model Construction

You must train and evaluate at least three tree-based classification models:

1. **Classification Tree (CART)**
2. **Random Forest**, properly tuned.
3. **A a traditional boosting model (e.g., XGBoost, LightGBM) or the C5.0 algorithm for tree-based boosting.**
4. **Additional Model**: You may add a fourth model of your choice.

3. Model Evaluation

- Evaluate model performance using a **confusion matrix**.
- Calculate performance metrics including:
  - Accuracy, Sensitivity, Specificity, and F1-score.
  - Compare models using **ROC curves**.

4. Hyperparameter Tuning

- Optimize model parameters using techniques such as:
  - Grid search or random search.
  - Cross-validation (e.g., 10-fold cross-validation).
- For Random Forest, you may test performance improvement by increasing the number of trees (e.g., 1000 trees).
- For C5.0, you may check the performance change when applying boosting with a value of 10.

Implementation Guidelines

- The implementation can be done in either **R or Python**.
- Yo should ensure their code is structured and well-documented to facilitate reproducibility.

Submission Requirements

Each group must submit **two files**:

1. **A report** (PDF or HTML) including:

- Description of the problem and dataset.
- Explanation of the models and their parameters.
- Results and comparison of model performance.
- Conclusions on the best-performing model.

2. **A script file** (`.R` or `.py`) containing all the necessary code to reproduce the analysis.

Submission Instructions

- Upload the files to **Atenea**.
- Additionally, submit a folder containing all files to a **GitHub repository**, whose link will be provided later.