Classification and Regression Tree

Xiaoqian Liu

North Carolina State University

Overview

- Introduction
- Construction of the tree
- Cross validation
- Python implementation

1 Introduction

Introduction

- The decision tree is one of the most popular used predictive modelling approaches
- Classification for predicting categorical labels
- Regression for numeric prediction
- The Classification And Regression Tree (CART) [1] is one commonly used algorithm to build binary decision trees

Introduction

Input:

- Years: number of years played in the major leagues
- Hits: number of hits made in the previous year

Output:

- Log salary (in thousands of dollars)

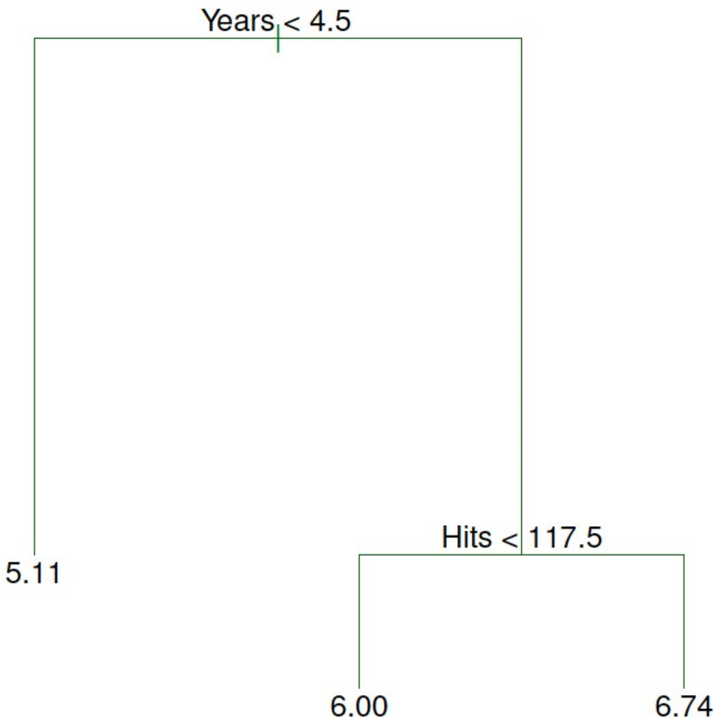


Figure1 : A regression tree for predicting the log salary of a baseball player. From Figure 8.1 in [2].

Introduction

- **Root node:** the entire training data set
- **Internal node:** a decision node to conduct splitting
- **Leaf node:** holds the decision and cannot be further split

- **Depth:** three

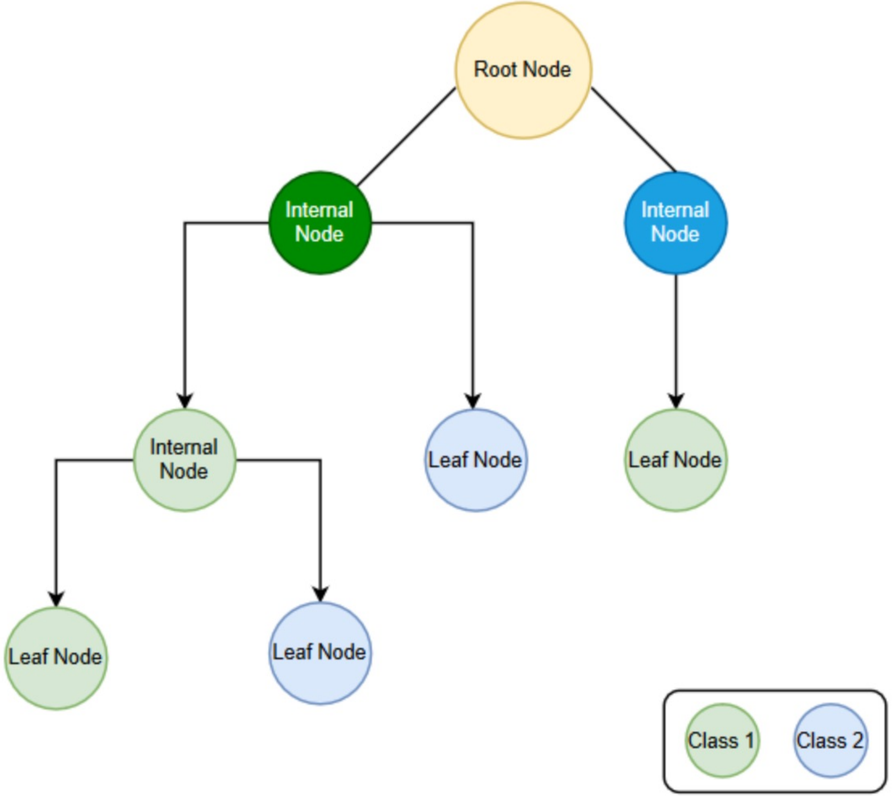


Figure 2 : Decision tree structure. Image Source.

Introduction

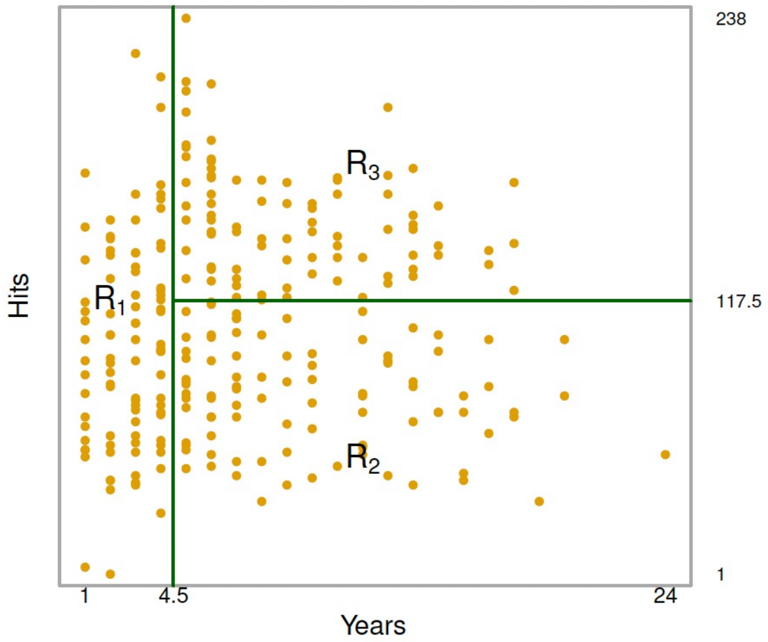


Figure3 : The three-region partition for the example in Figure 1. Adopted from [2.]

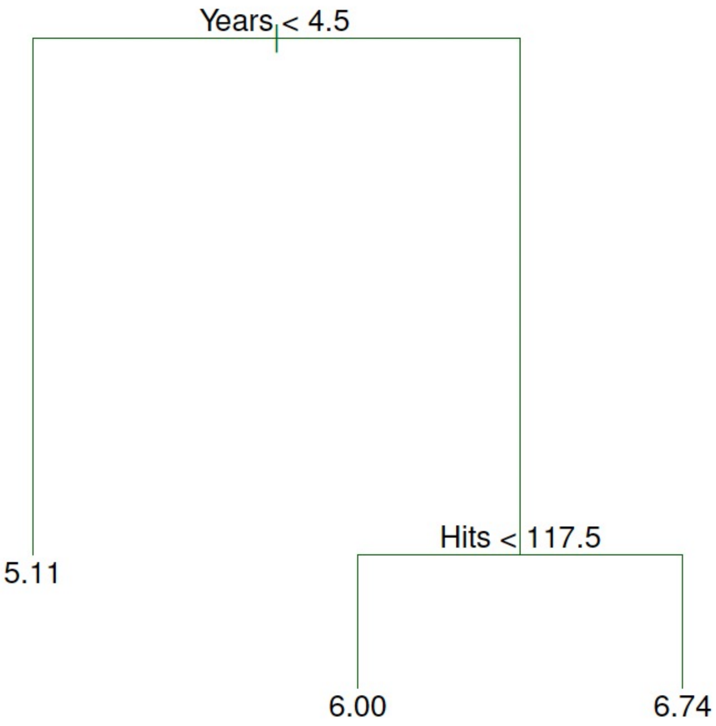


Figure1 : A regression tree for predicting the log salary of a baseball player. Adopted from [2].

Advantages

Advantages of the decision tree:

- **Simplicity:**

  trees can be displayed graphically and easy to understand
- **Flexibility:**

  non-parametric model;

  handle both numerical and categorical data;
- **High interpretability:**

  mirrors human decision making

2 Construction of the tree

Construction

How to construct a tree / How to divide the feature space/ how to split a node:

- Which feature
- A cut-off value
- A criterion

The CART algorithm

Construction

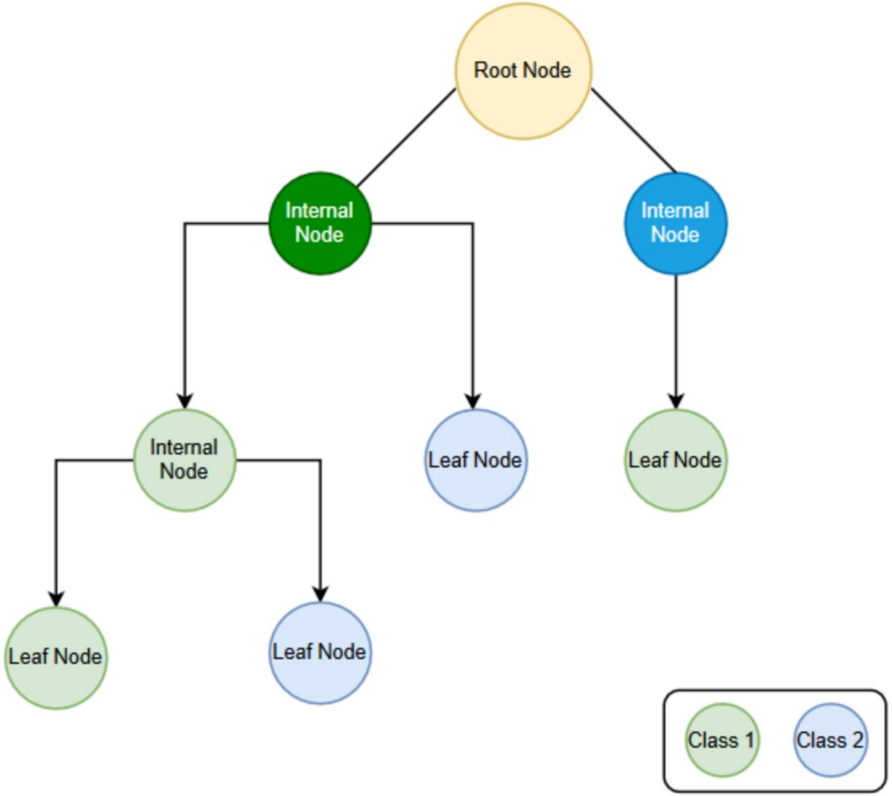
The **recursive binary splitting** approach:

- **Top-down**

  begin at the top; successively split

- **Greedy**

  find the **best** split at each node



Construction

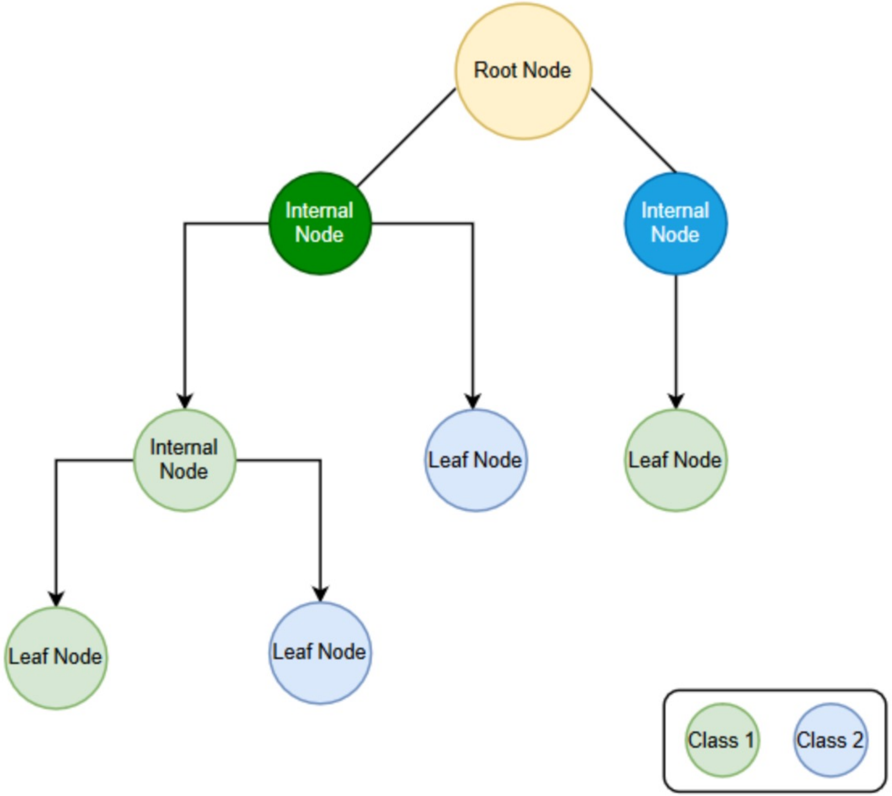
How to evaluate the “best”? (criterion)

- Regression:

$$\text{RSS} = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2$$

- Classification:

$$\text{Gini} = \sum_{j=1}^{J} \sum_{k=1}^{K} \hat{p}_{jk} (1 - \hat{p}_{jk})$$



Construction

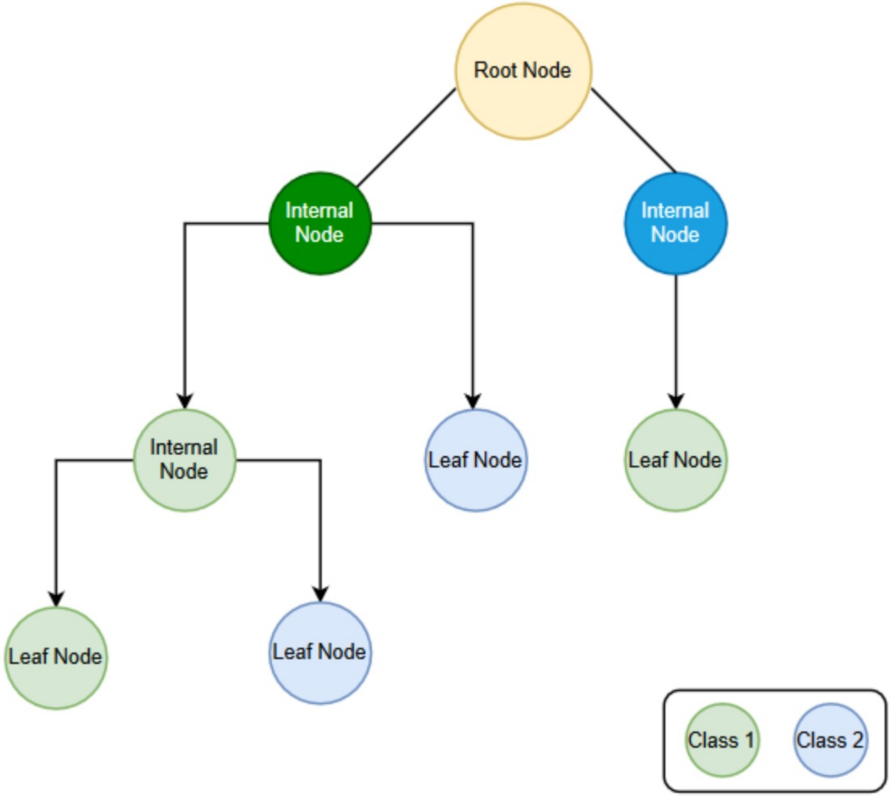
How to predict?

- Regression:

  $\hat{y}_{R_j}$ = sample mean

- Classification:

  $\hat{y}_{R_j}$ = most common class

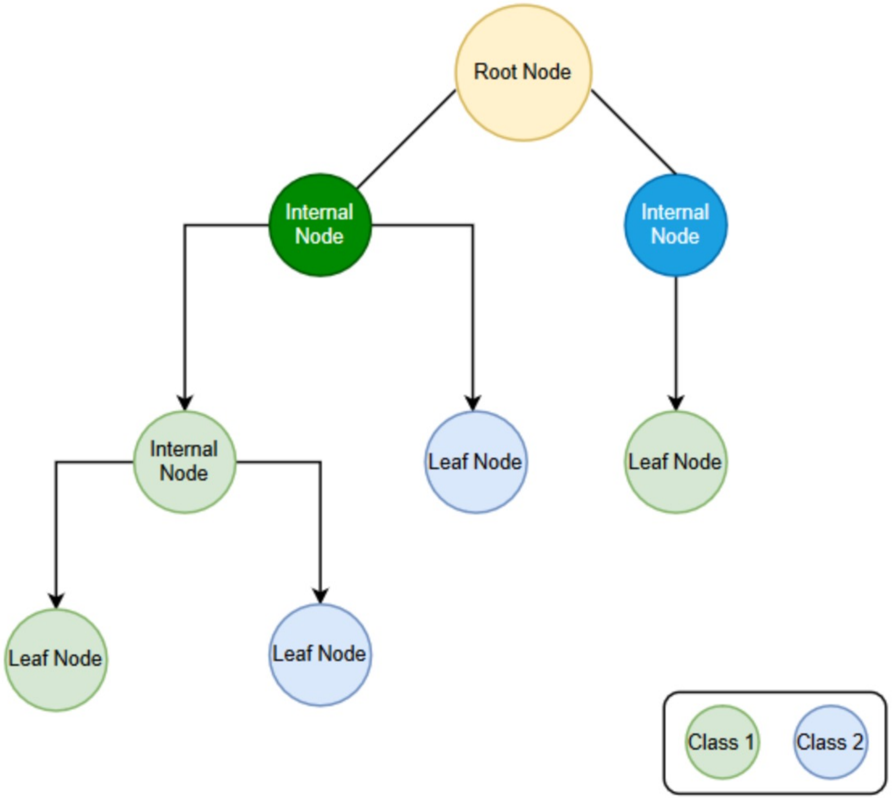


Construction

How to stop?

- Min_samples_leaf
- Min_samples_spilt
- Max_depth

Important to avoid overfitting!



Construction

- A set of p feature variables $S = \{x_1, \cdots, x_p\}$, a response variable y
- Recursively create binary partitions (for regression):
  - Start at the root node.
  - Consider a splitting variable $x \in S$ and a cut-off value c to divide the space into $\{x \leq c\}$ and $\{x > c\}$, then model the response y by its sample mean over each region.
  - Choose the splitting variable and cut-off value that achieves the best fit in a least squares sense.
  - Stop the splitting once a stopping rule is satisfied.

Advantages

Advantages of the decision tree:

- **Simplicity:**

  trees can also be displayed graphically
- **Flexibility:**

  non-parametric model;

  handle both numerical and categorical data;
- **High interpretability:**

  mirrors human decision making

3 Cross validation

Cross Validation

Tuning tree complexity: cross-validation

- As the number of features increases, the size of the tree grows rapidly
  - An overly complex model
  - Nullify the model's attractive interpretability
  - Overfitting problem
- Balance between the tree complexity and the model's goodness-of-fit

Cross Validation

- K-Fold Cross-Validation Procedure:
  - Suppose we wish to select a maximal tree depth γ from a set $\{\gamma_1, \cdots, \gamma_m\}$
  - Given a sample of data, randomly split the full dataset into K roughly equal-sized groups. Set aside one group as the validation set and use the remaining K-1 groups as the training set.
  - Build a tree model on the training set for each γ_j for $j = 1, \cdots, m$. Then calculate the mean squared prediction error of each fitted model on the hold-out validation set.
  - The process is repeated K times, and we obtain K estimates of the prediction error for each γ_j for $j = 1, \cdots, m$.
  - Select the maximal depth γ_j that minimizes the average prediction error.

4 Python implementation

Disadvantages

- **Non-robustness:**

  A small change in the training data could cause a large change in the tree

- **Predictive accuracy:**

  Trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches

  bagging or boosting

References

[1]. Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. Classification and Regression Trees. CRC press, 1984.

[2]. Gareth, James, Witten Daniela, Hastie Trevor, and Tibshirani Robert. An introduction to statistical learning: with applications in R. Spinger, 2013.

Thanks!