Decision Trees

Introduction

- We now discuss *tree-based* methods.
- Our main goal is to predict a target variable based on several input variables.
- Decision trees can be applied to both regression and classification problems.

Introduction

- Thus there are of two main types:
  1. Classification tress used when the predicted outcome is a categorical variable.
  2. Regression tress used when the predicted outcome is a quantitative variable.
- The term **Classification And Regression Tree (CART)** analysis is a popular umbrella term used to refer to both of the above methods.

Regression Tree

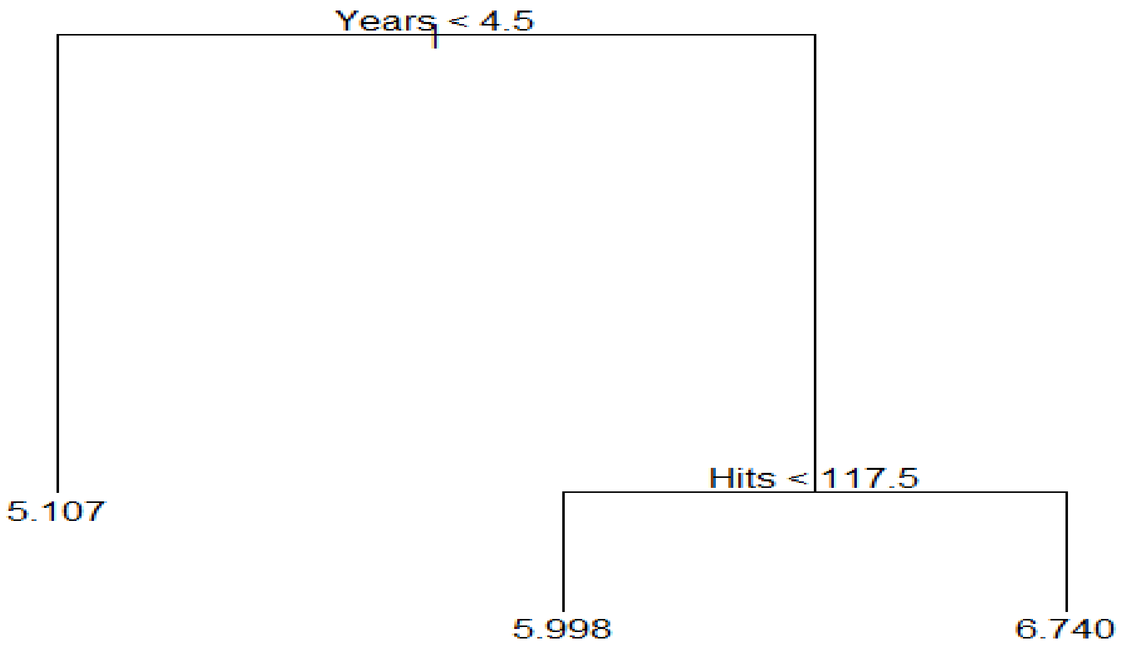
Dataset: Baseball Players' Salaries

- Major League Baseball Data for the two seasons.
- A data frame with 322 observations of major league players on 20 variables.
- Goal: To predict Salary based on a number of predictors, such as various performance indicators, number of years, etc.

Dataset: Baseball Players' Salaries

- For the time being, we will consider following three variables
    1. Salary (Thousands of dollars)
    2. Years (Number of years he has played in the major leagues)
    3. Hits (Number of hits he made in the previous year)
- Goal: To predict Salary based on Years and Hits.
- In order to reduce the skewness, we first log-transform Salary so that it has more of a typical bell-shape.

Baseball Players' Salaries



Baseball Players' Salaries

- The predictor space is segmented into three regions:

$R_1 = \{X|Years < 4.5\}$,

$R_2 = \{X|Years \geq 4.5, Hits < 117.55\}$,
$R_3 = \{X|Years \geq 4.5, Hits \geq 117.55\}$.

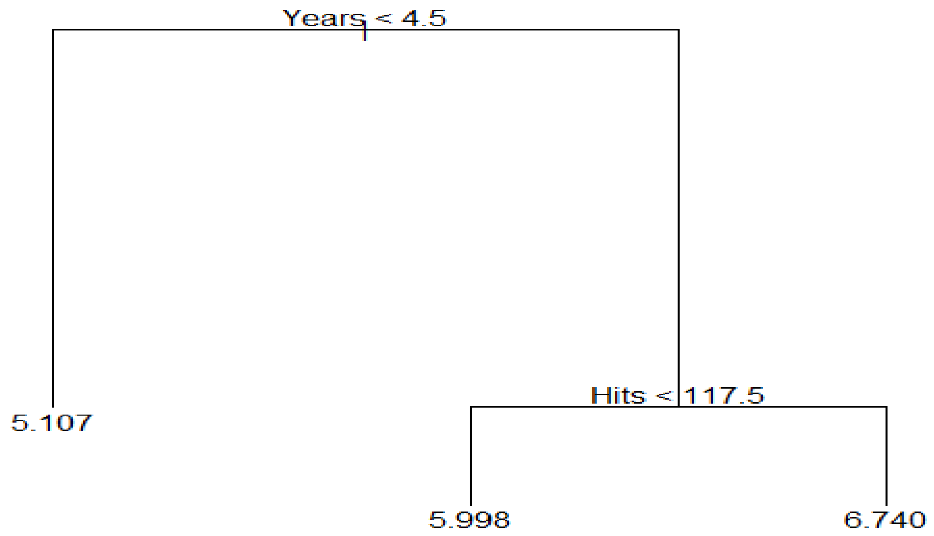
- The predicted Salary for these three groups are

$$\$1{,}000 \times e^{5.107} = \$165{,}174$$

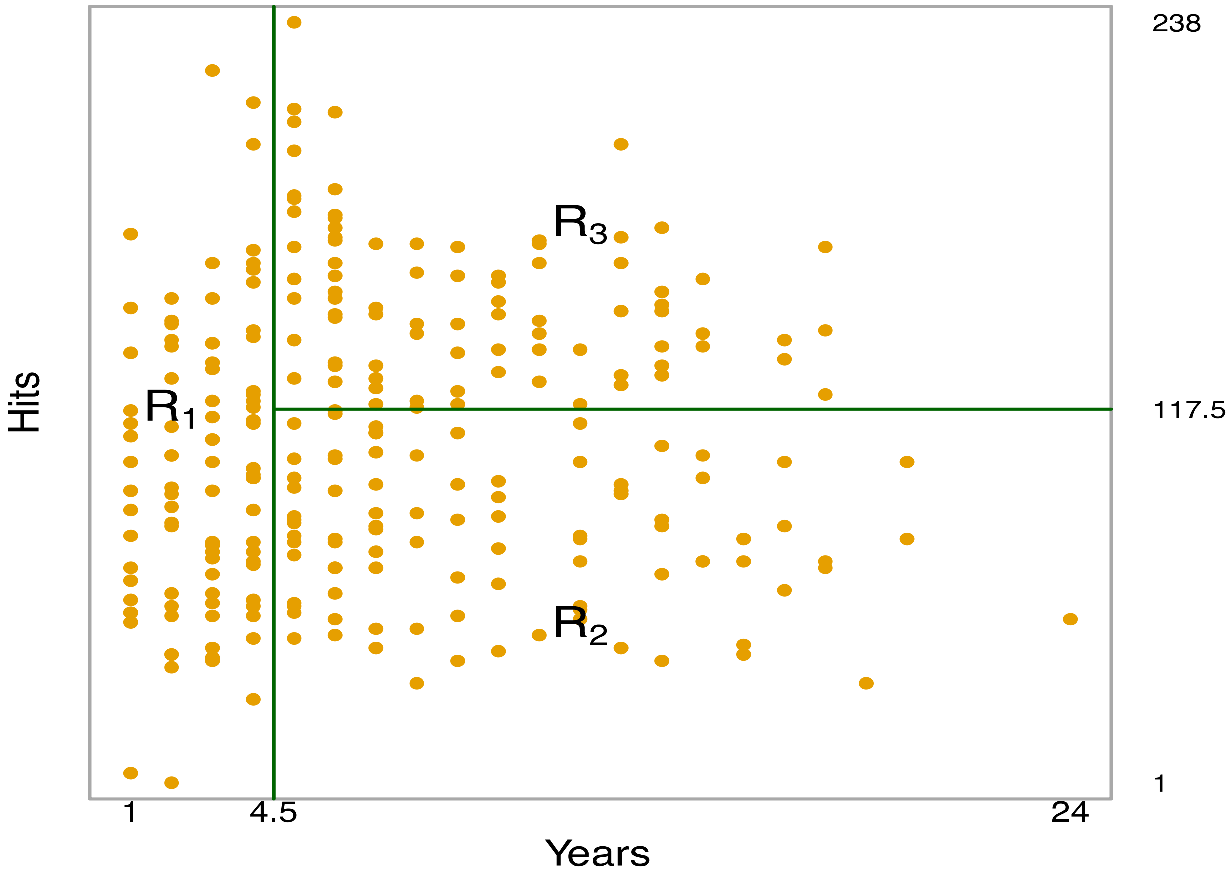
$$\$1{,}000 \times e^{5.998} = \$402{,}834$$

$$\$1{,}000 \times e^{6.740} = \$845{,}346,$$

respectively.

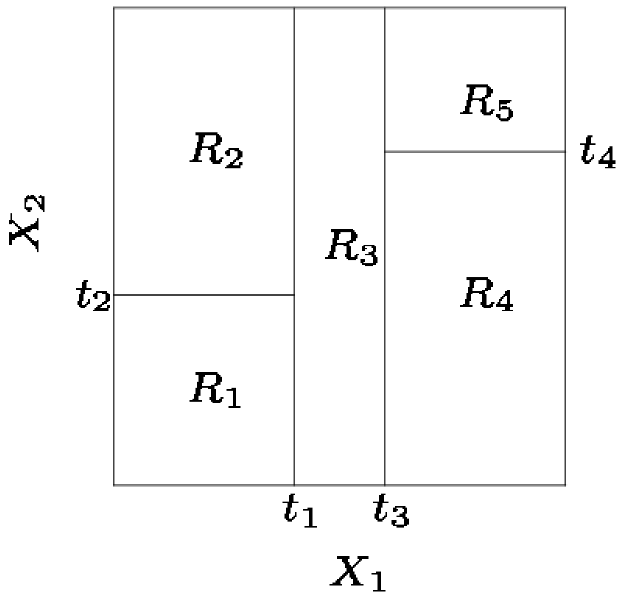


Baseball Players' Salaries: Another Representation



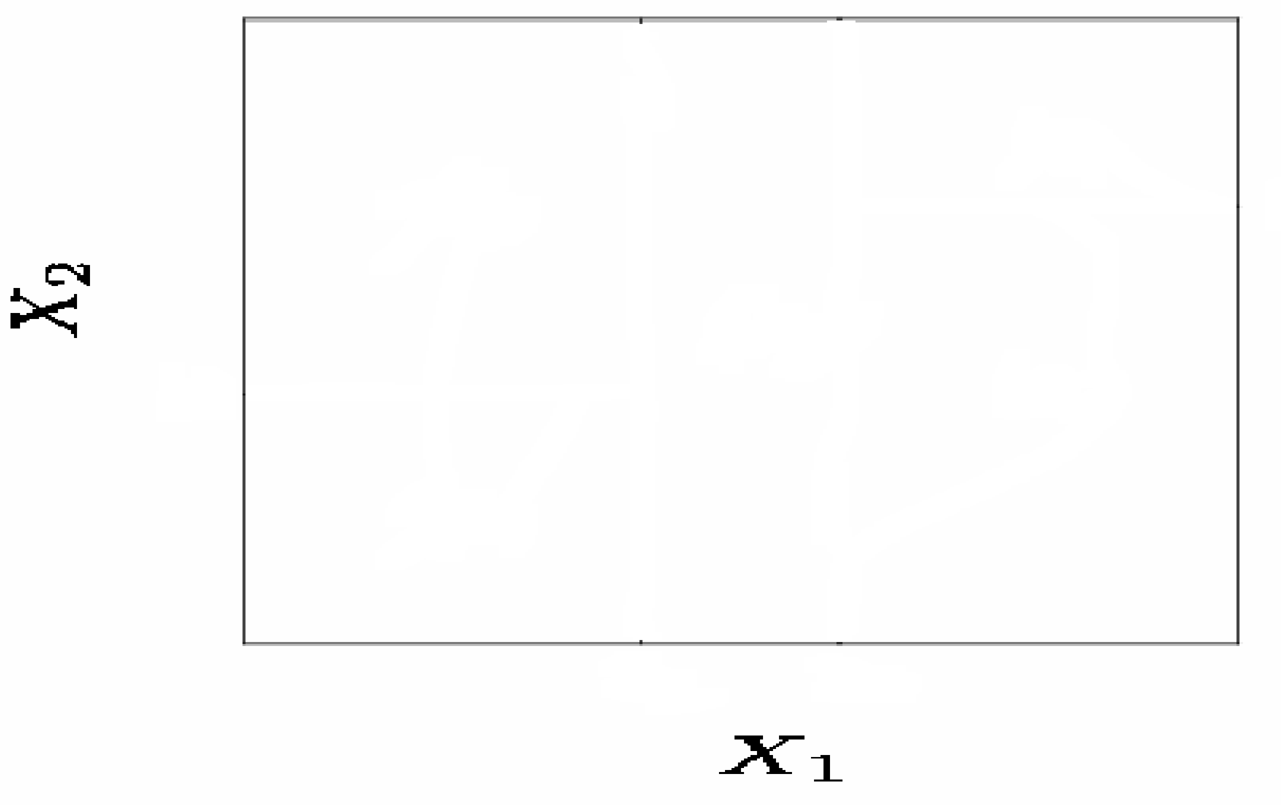
The General View

- Consider two predictors X_1 and X_2.
- The predictor space is segmented into five distinct regions.
- Depending upon which region our observation comes from, we would make one of five possible predictions for Y.
- We typically use the mean of the training observations belonging to a particular region as the predicted value for the region.



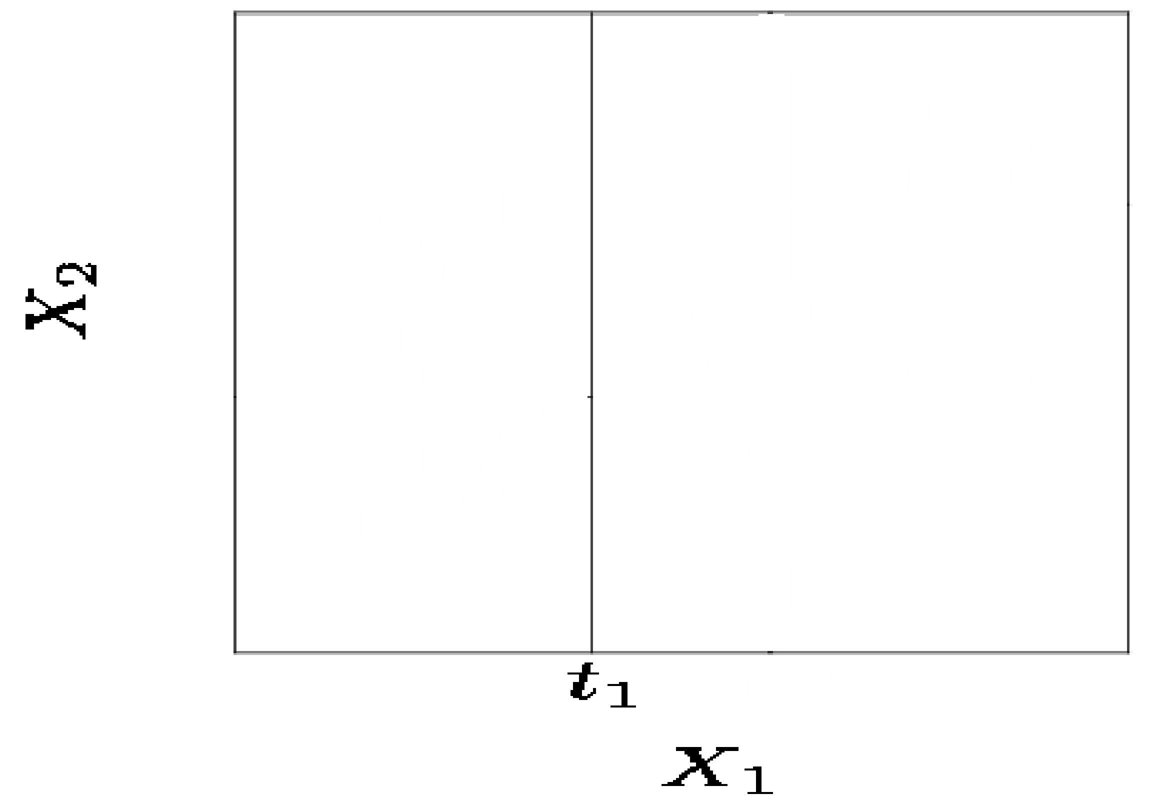
The General View

- Typically we create the partitions by iteratively splitting one of the X variables into two regions.



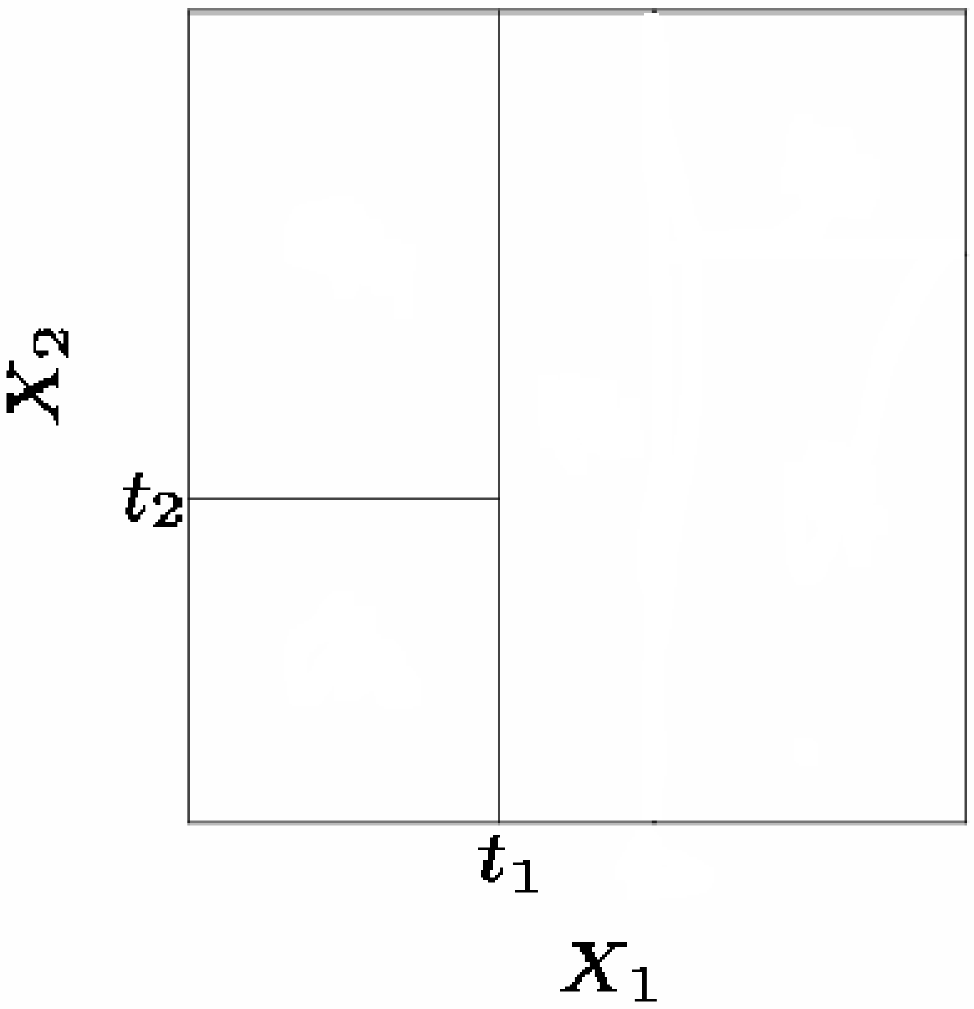
The General View

1. First split on $X_1 = t_1$.



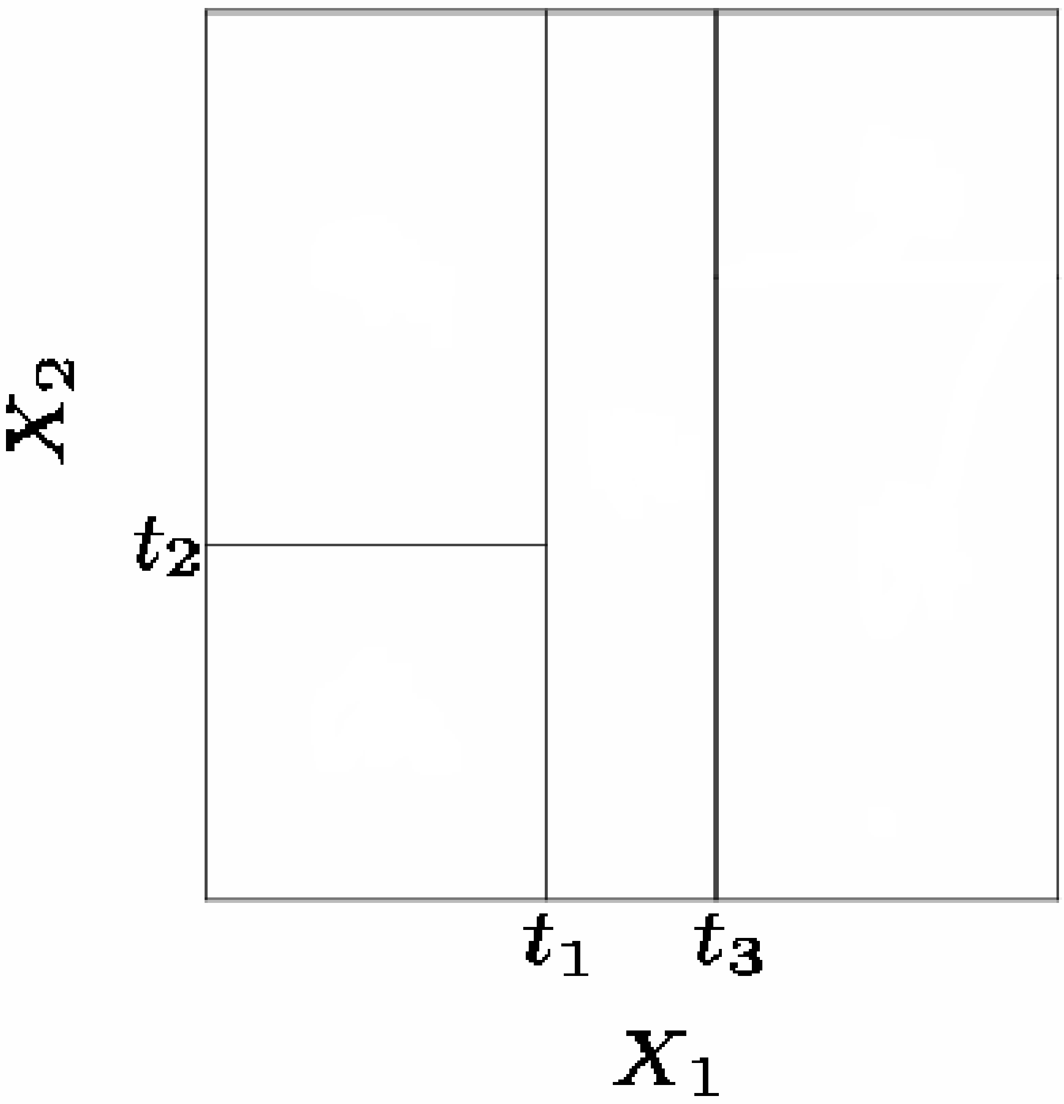
The General View

1. First split on $X_1 = t_1$.
2. If $X_1 \leq t_1$, split on $X_2 = t_2$.



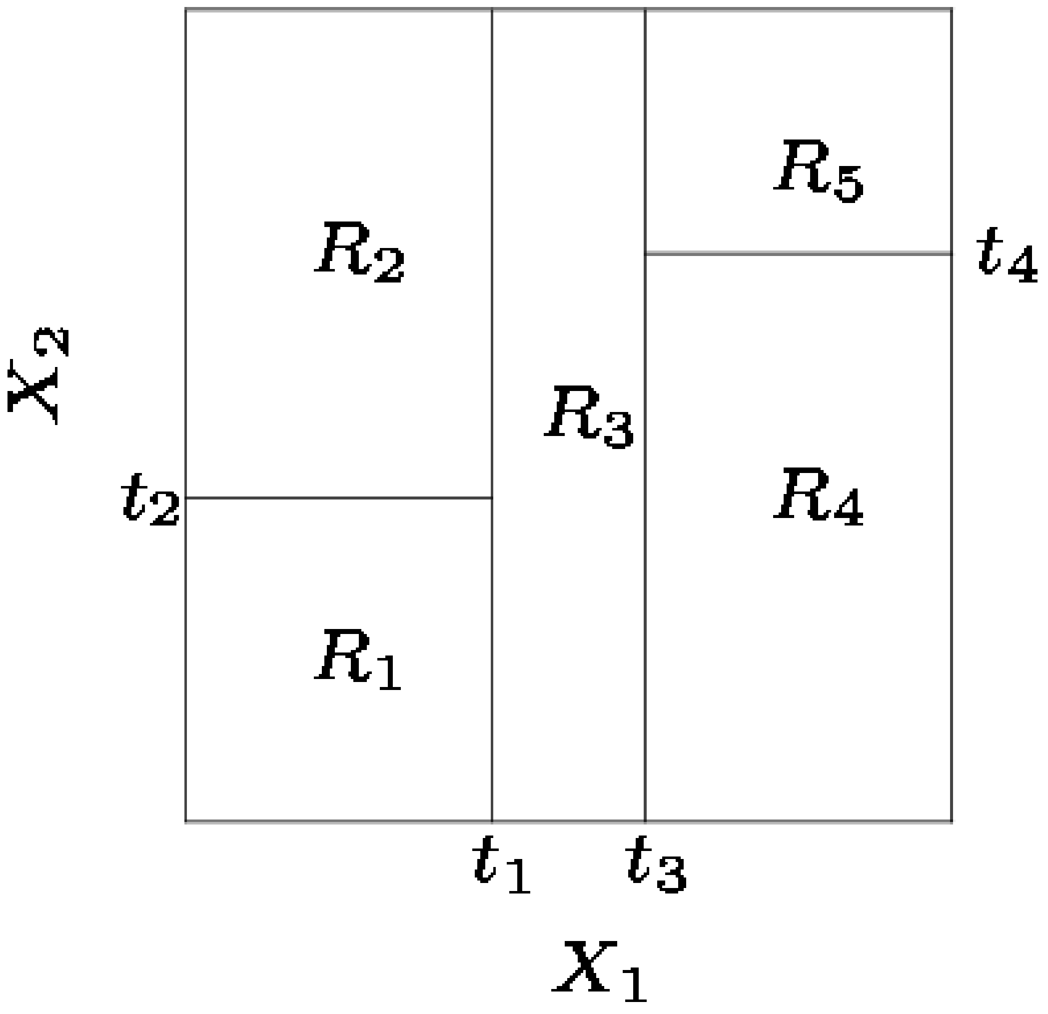
The General View

1. First split on $X_1 = t_1$.
2. If $X_1 \leq t_1$, split on $X_2 = t_2$.
3. If $X_1 > t_1$, split on $X_1 = t_3$.

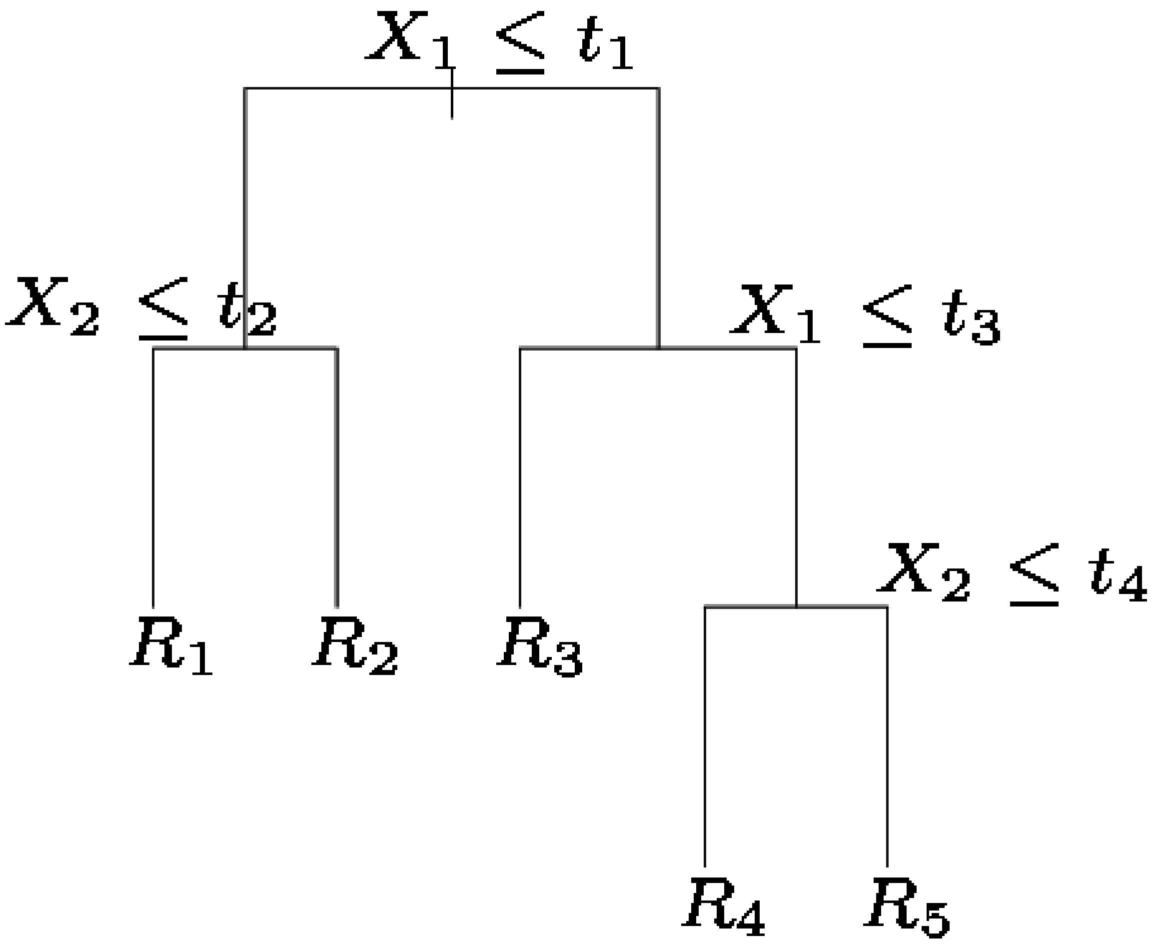


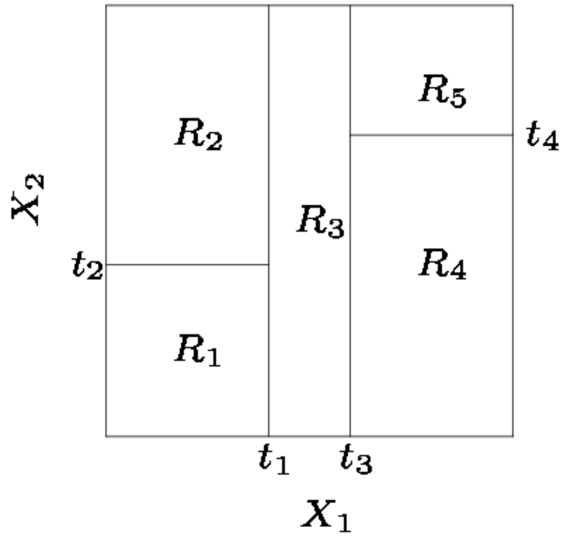
The General View

1. First split on $X_1 = t_1$.
2. If $X_1 \leq t_1$, split on $X_2 = t_2$.
3. If $X_1 > t_1$, split on $X_1 = t_3$.
4. If $X_1 > t_3$, split on $X_2 = t_4$.



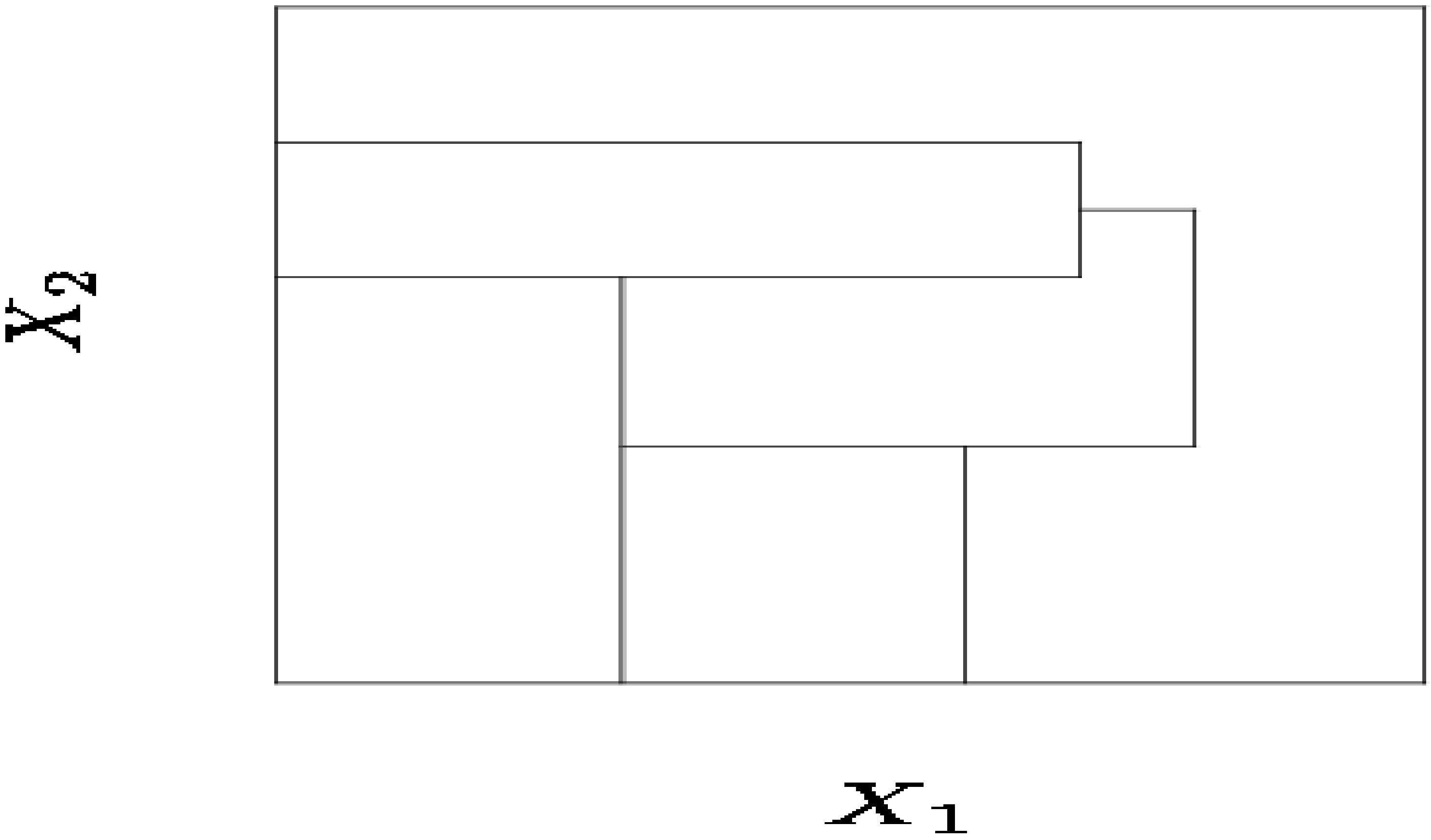
The General View





- When we create partitions like this, we can always represent them using a tree-like structure.
- This tree-like representation provides a very simple way to explain the model to a non-expert!!!

Can this partition arise in a similar fashion?



Regression Tree: Two steps

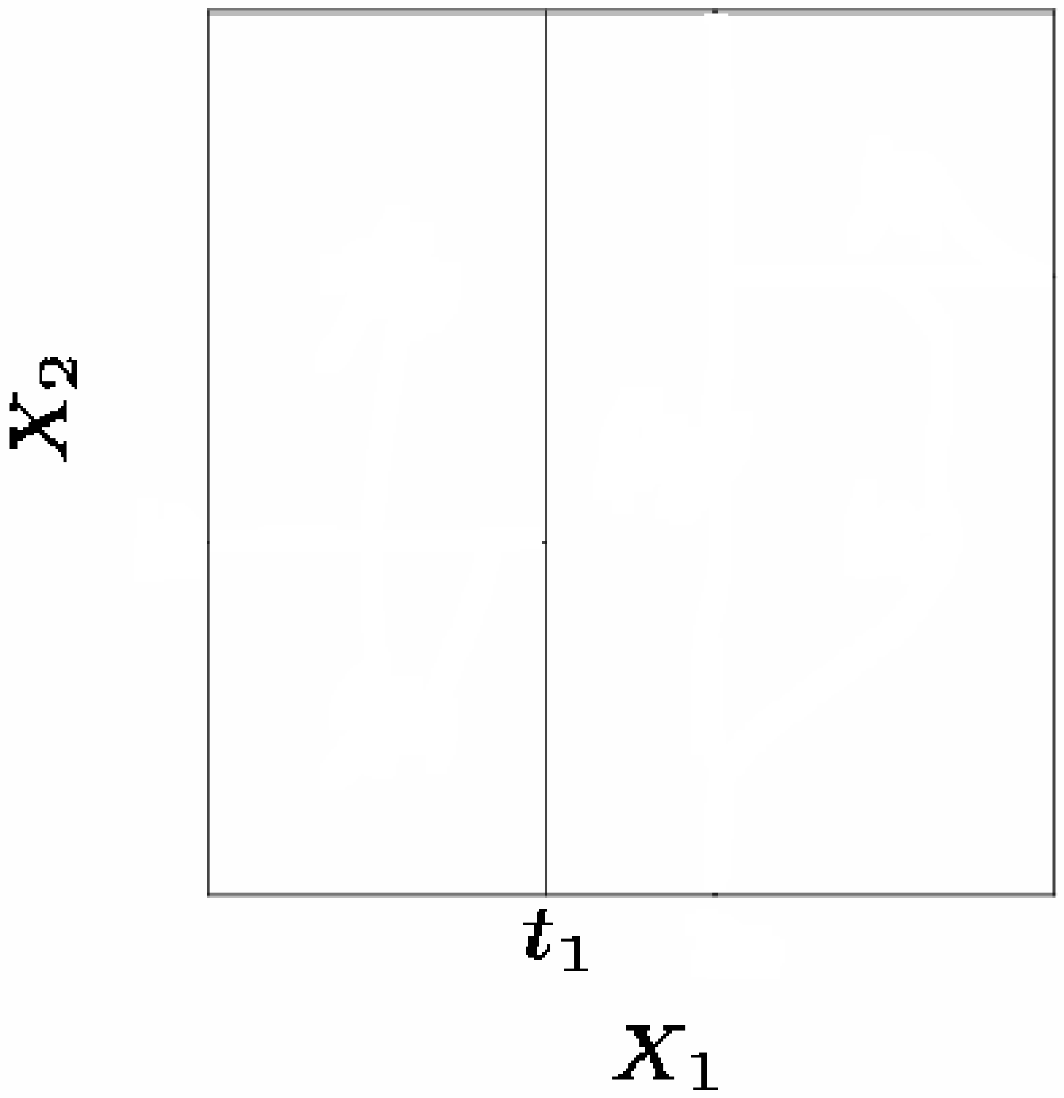
1. Divide the predictor space, i.e., the set of possible values of the predictors, into J distinct and non-overlapping regions, namely $R_1, R_2, \dots, R_J$.
2. For every observation that falls into the region R_j, we make the same predictions, which is simply the mean of the response values for the training observations in R_j.

Some Natural Questions about Step 1

- How do we construct the regions $R_1, R_2, \dots, R_J$?
  - ✓Though these regions could have any shape in theory, we choose to segment the predictor space into high-dimensional rectangles, or boxes.
  - ✓This is mainly done for simplicity and for ease of interpretation.
- How should we decide where to split?
  - ✓ We take a *top-down, greedy* approach, that is known as recursive binary splitting.
  - ✓The approach is top-down, because it begins at the top of the tree.
  - ✓The approach is greedy because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead.

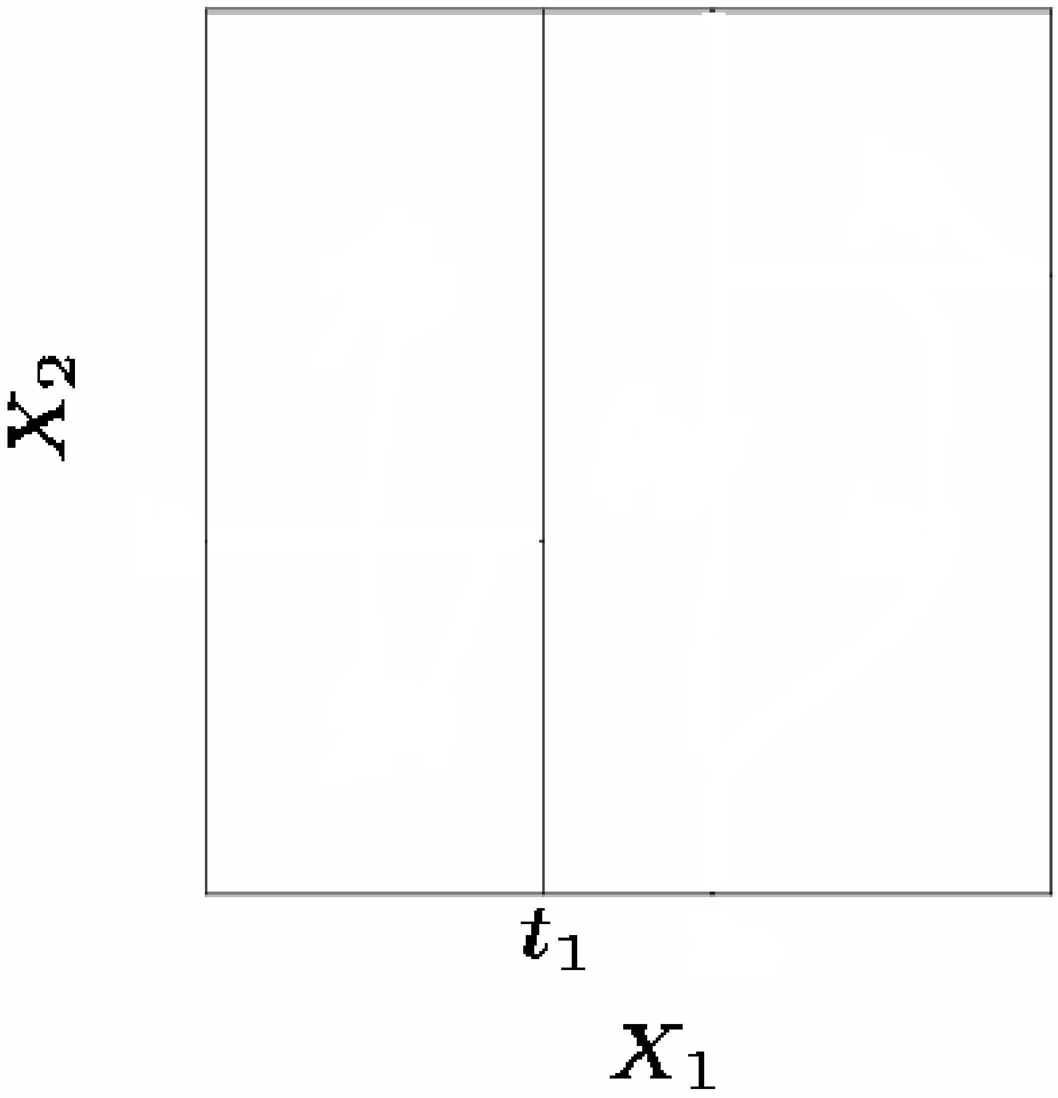
Where to split?

- We consider splitting into two regions, $X_j \leq s$ and $X_j > s$ for all possible values of s and $j = 1,2$.
- We then choose the s and j that results in the lowest MSE on the training data.



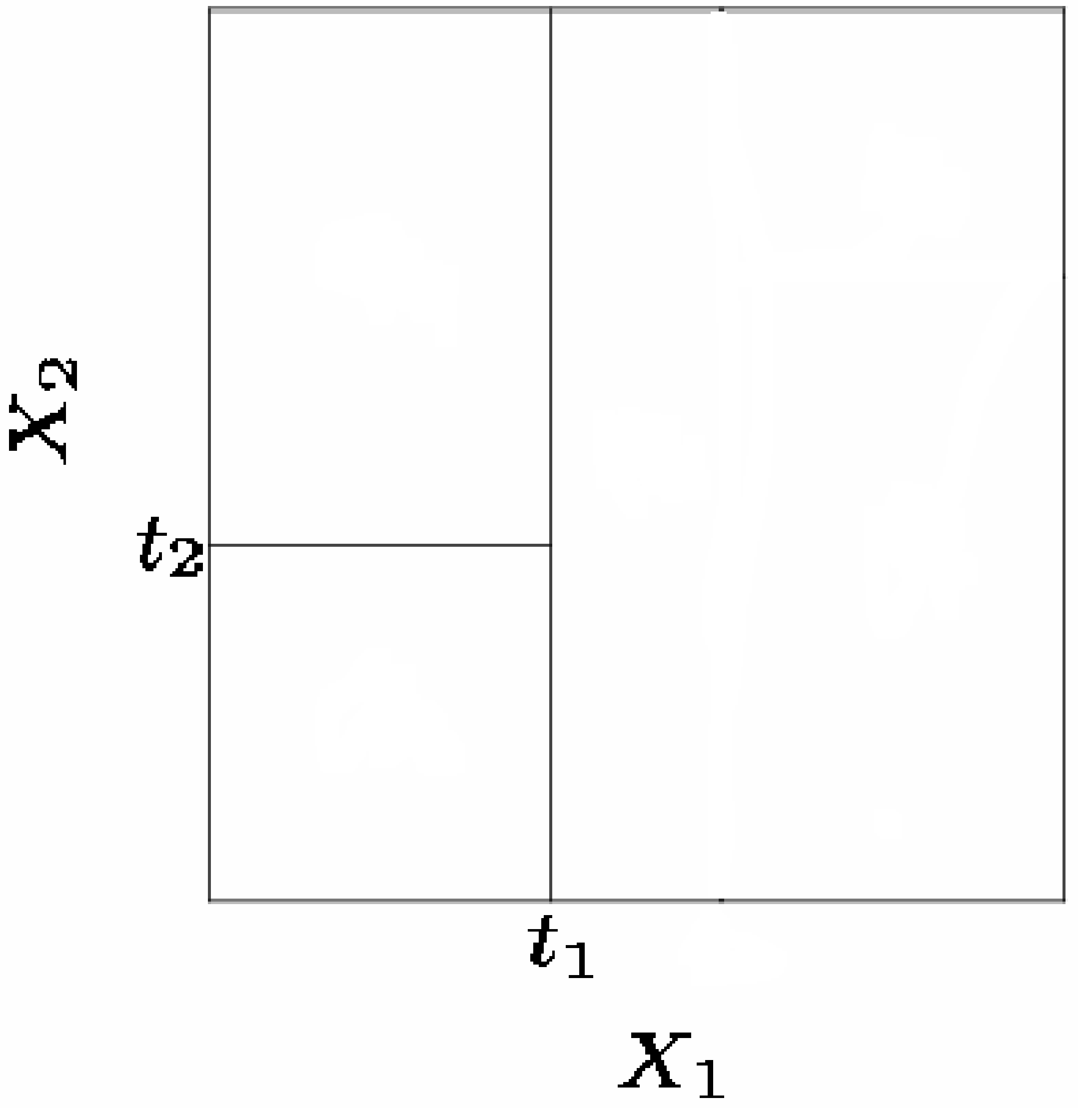
Where to split?

- Here the optimal split was on X_1 at point t_1.
- We now repeat the process looking for the next best split except that we must also consider whether to split the first region or the second region up.
- Again the criteria is smallest MSE.



Where to split?

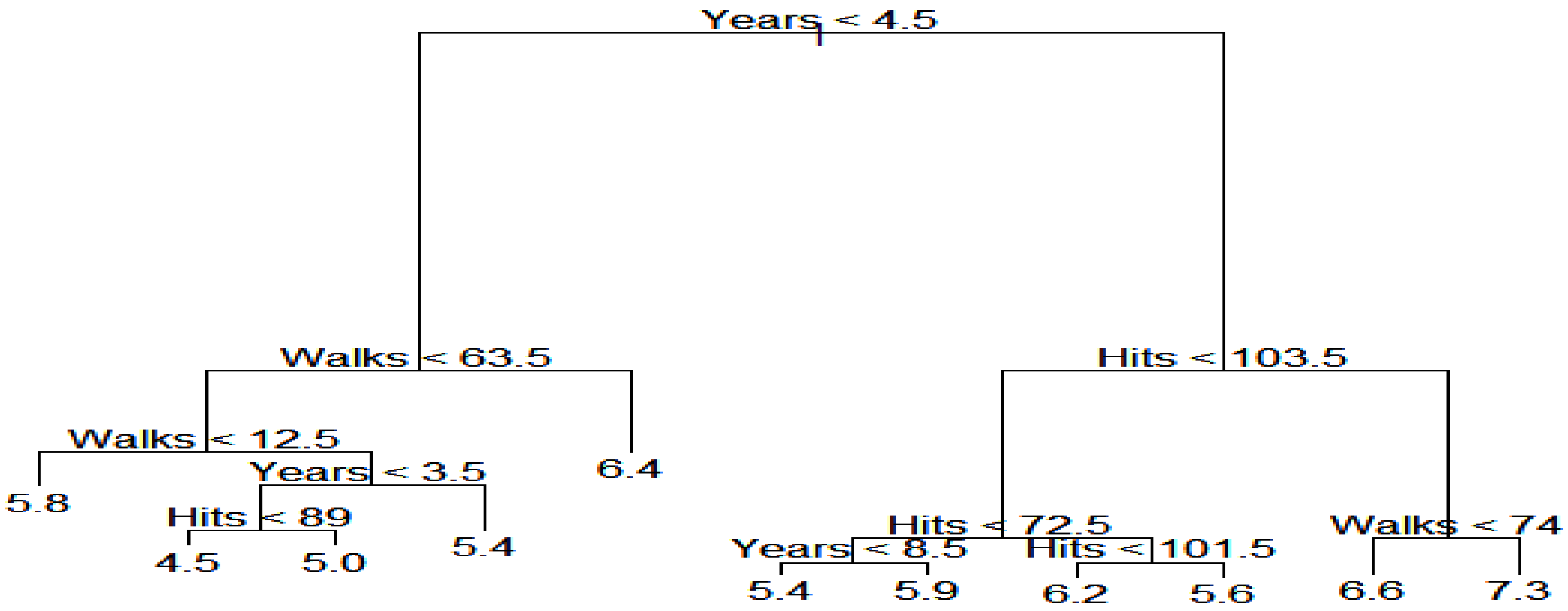
- The optimal split was the left region on X_2 at point t_2.
- This process continues until our regions have too few observations to continue e.g. all regions have 5 or fewer points.



Baseball Players' Salaries

- We first remove all the rows that have missing values in any variable.
- This reduces the number of observations to 263.
- We then randomly split the observations into two parts- the training set containing 132 observations and the test set containing 131 observations.
- We will first build a tree based on the training data set.

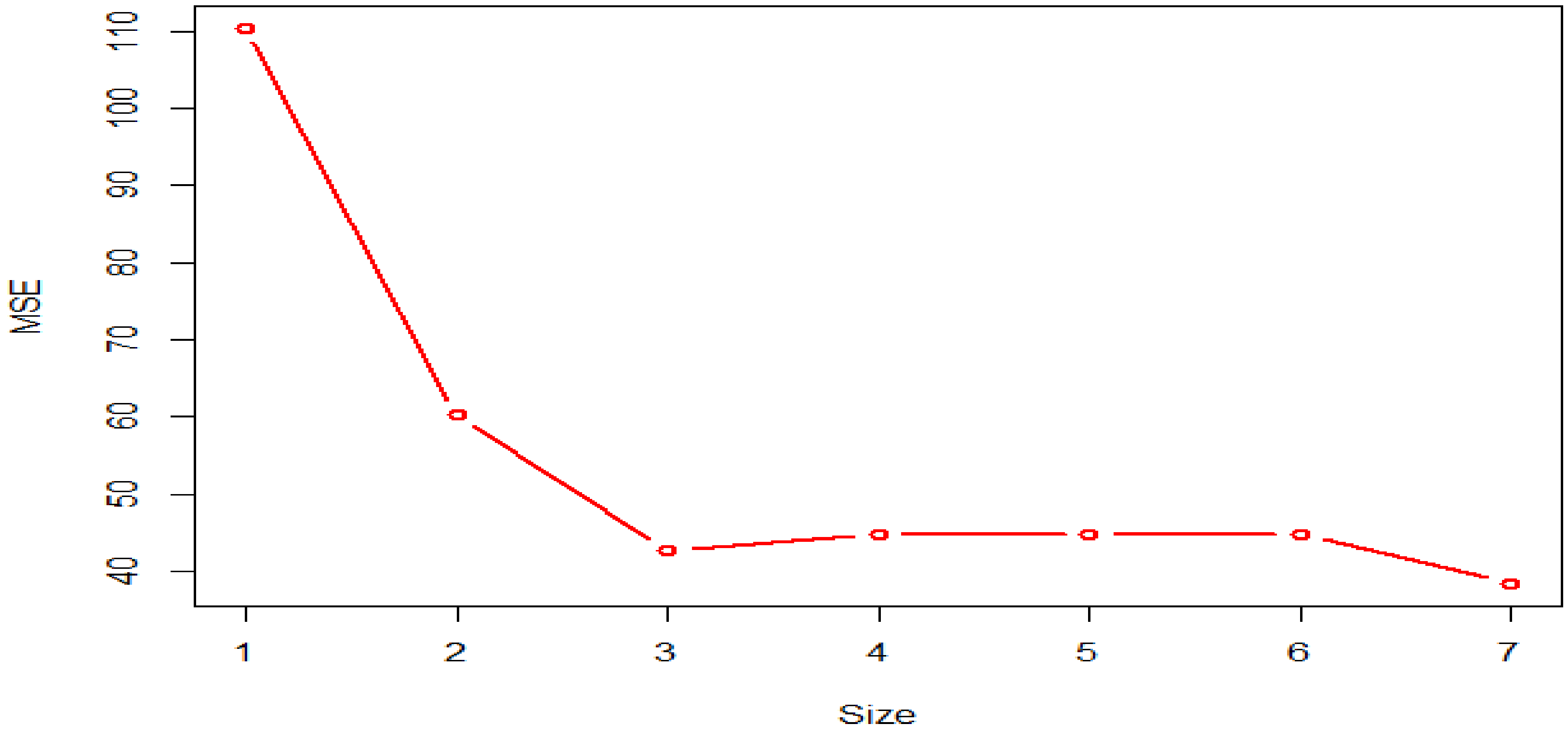
Baseball Players' Salaries



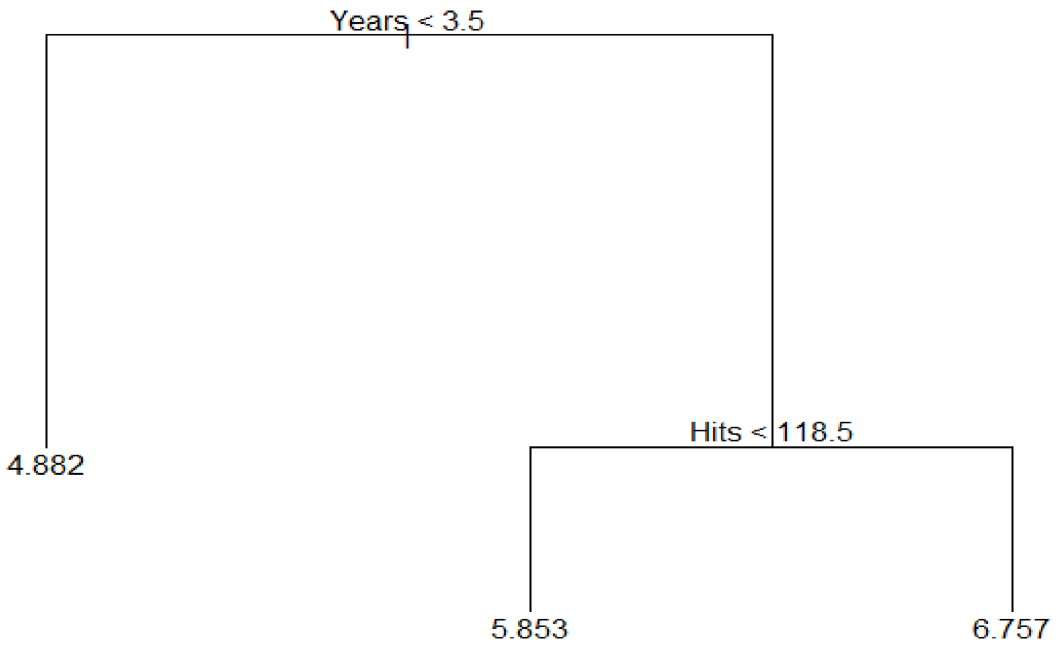
Tree Pruning

- A large tree (i.e., one with many terminal nodes) may tend to over-fit the training data.
- It may lead to poor test set performance.
- A smaller tree with fewer splits may be easy to interpret.
- Generally, we can improve accuracy by “pruning” the tree i.e. cutting off some of the terminal nodes.
- How do we know how far back to prune the tree?
- We use six-fold cross validation to see which tree has the lowest error rate.

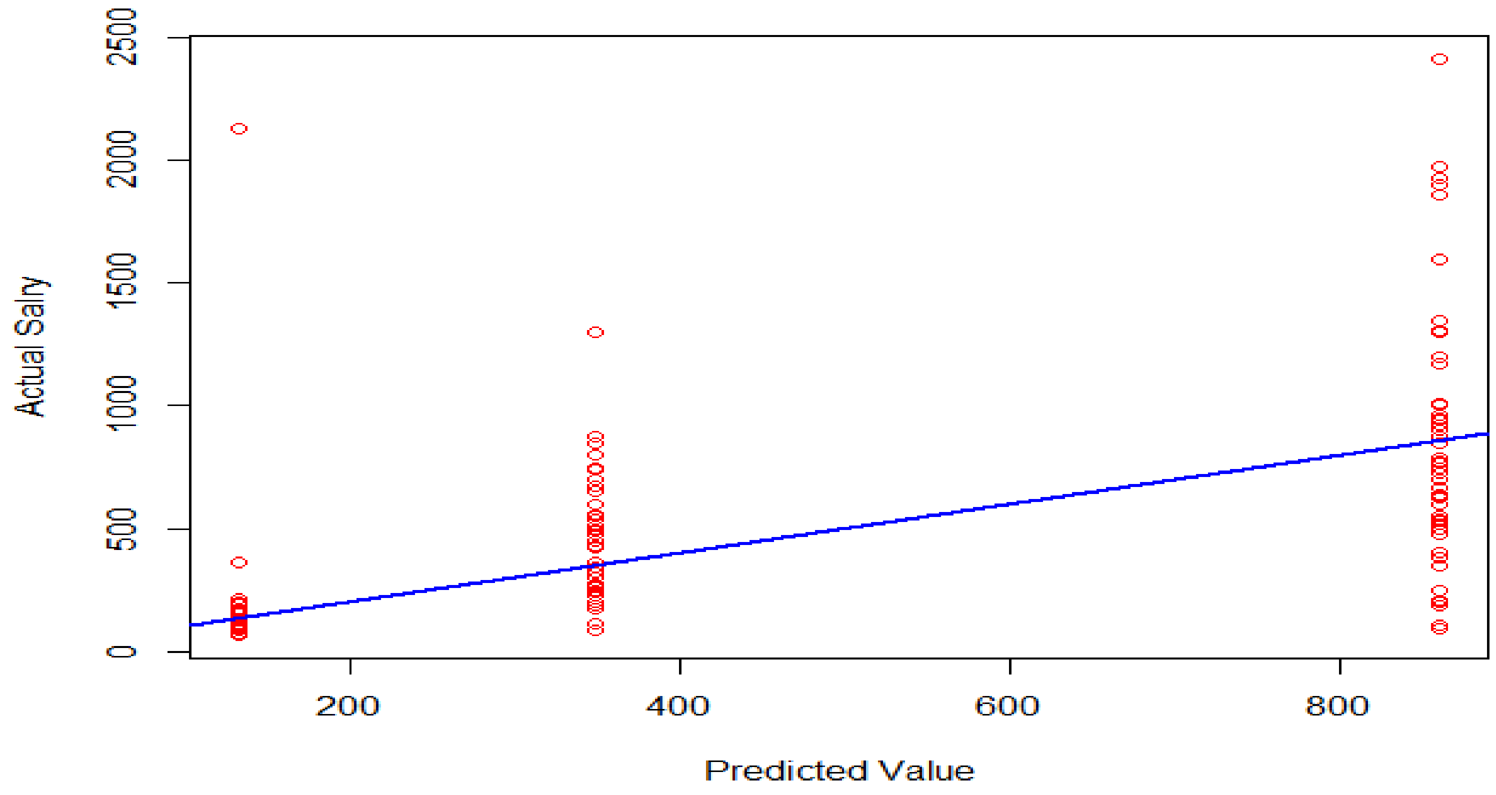
Cross-Validation



Pruned Tree



Test Error



RMSE=408.29