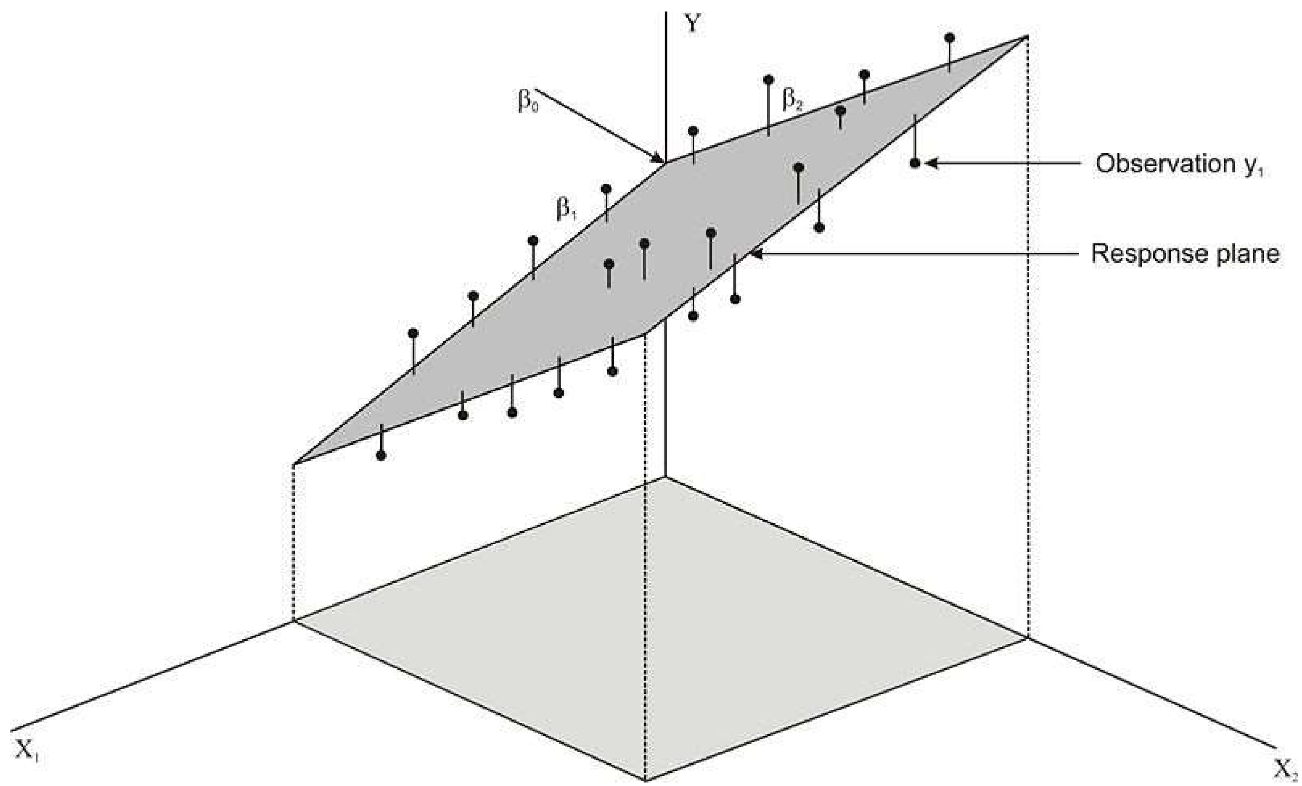
REGRESSION TREES

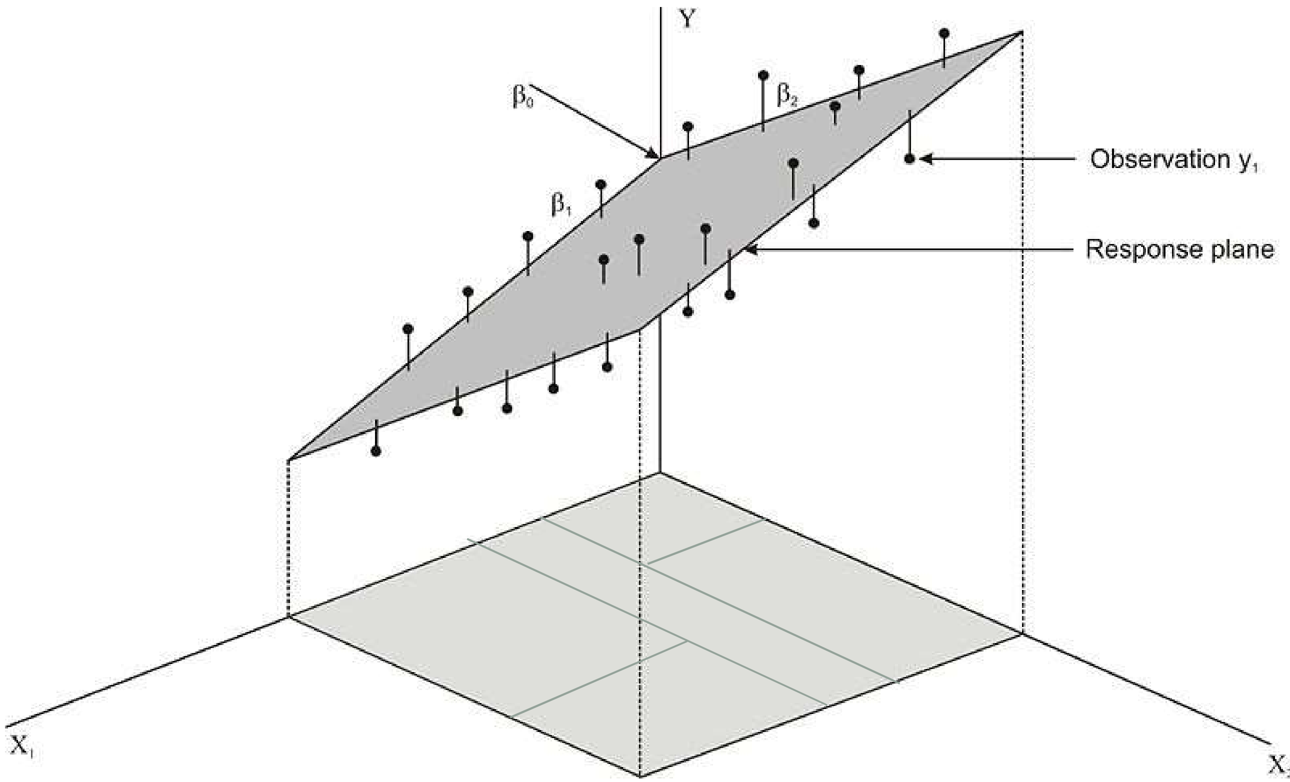
Outline

- Linear Regression vs RegressionTrees
- Predictors space
- Splitting predictors
- Bagging
- Random Forest

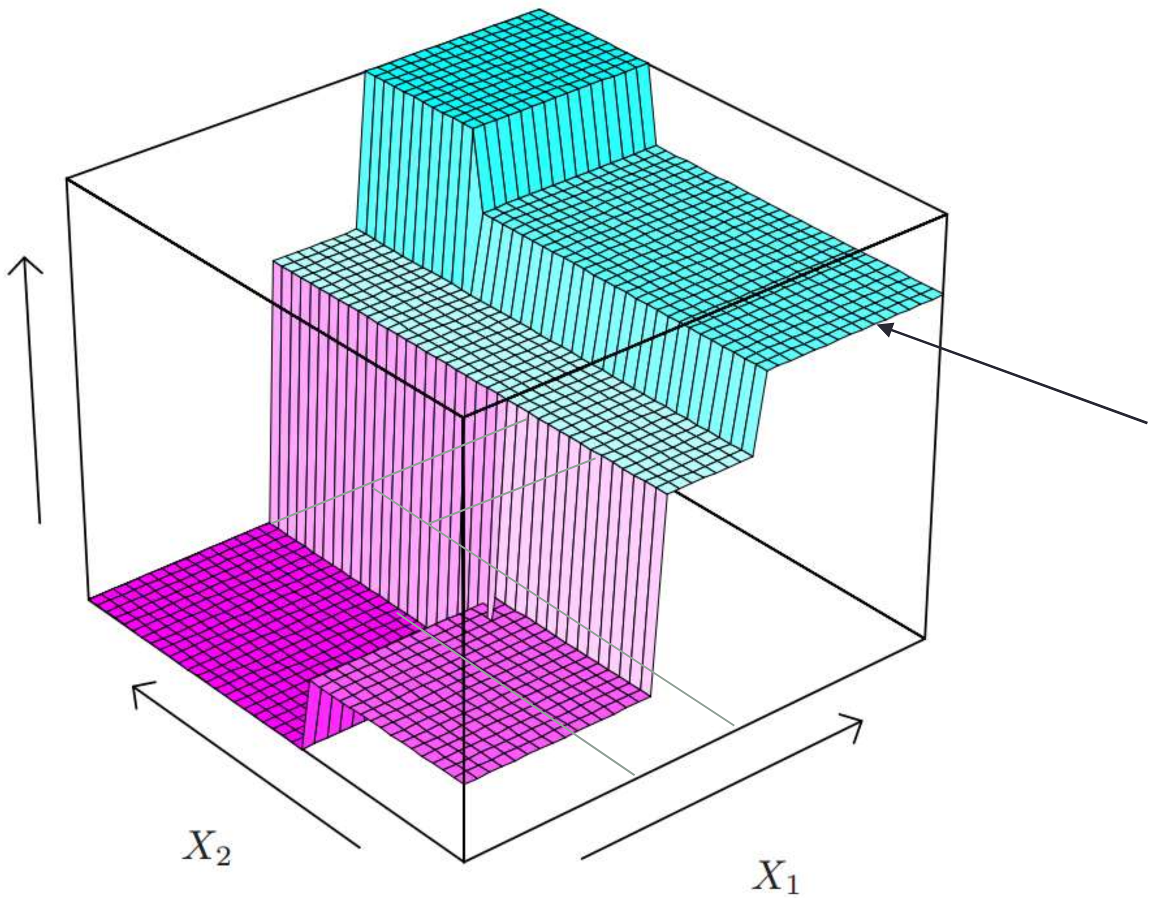
Linear Regression vs. Regression Tree



Linear Regression vs. Regression Tree



Linear Regression vs. Regression Tree



response surface

Partitioning Up the Predictor Space

- Split the predictor space into disjoint regions, R_1, R_2,…,R_k
- For each region R_j the prediction is the mean response of all observations in that region

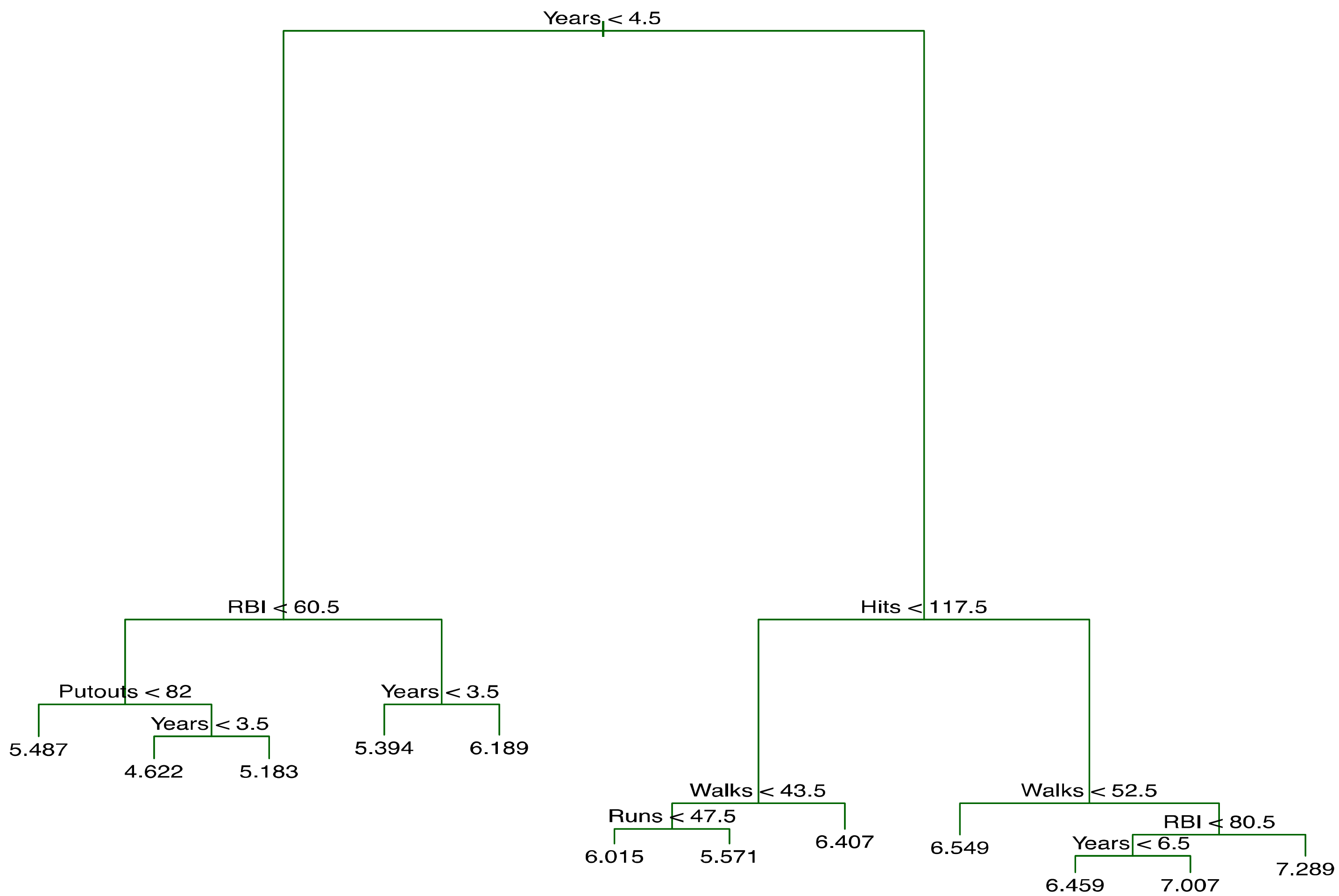
Regression Trees

- Suppose that we have two regions R_1 and R_2

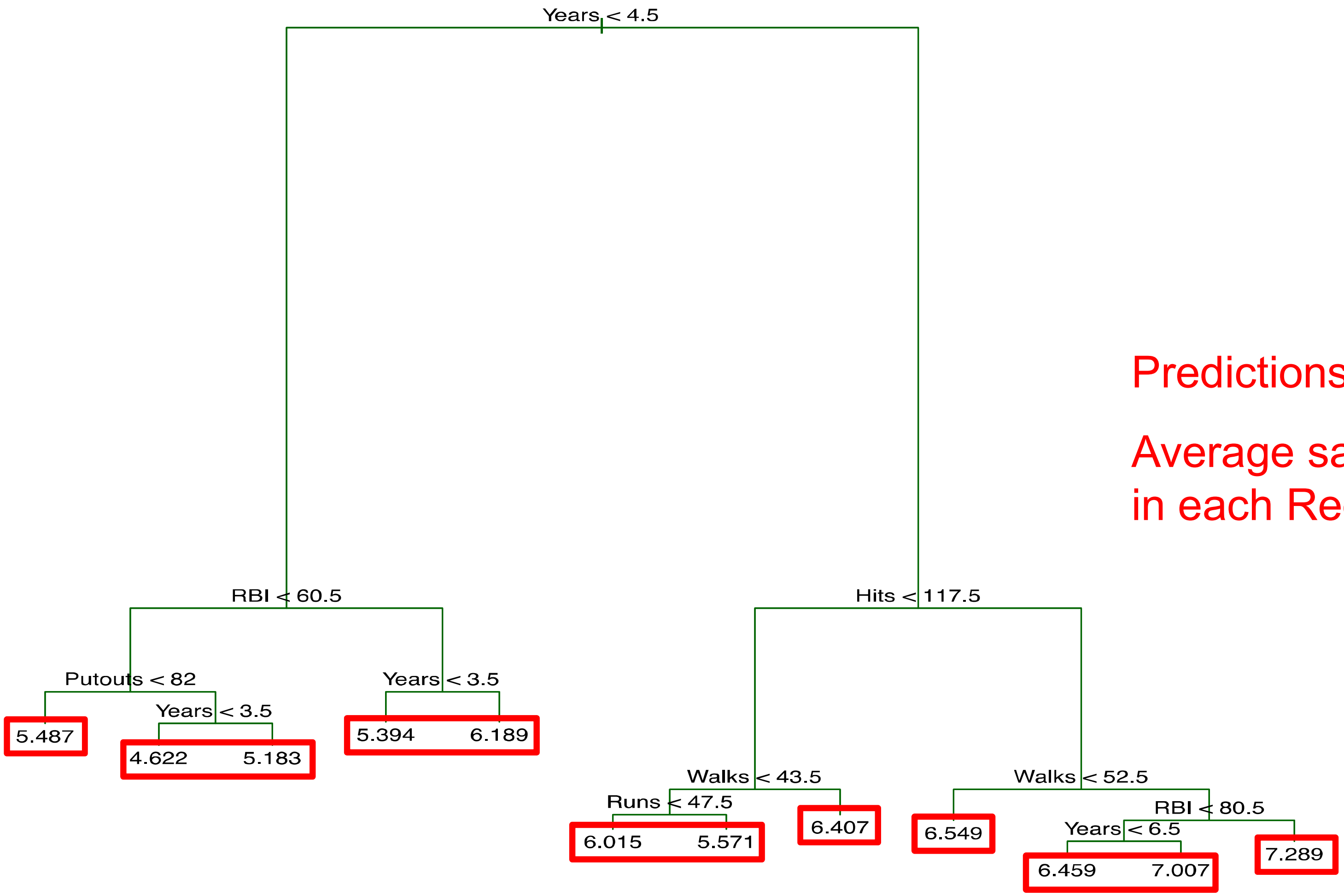
- The observations in Region R_1 have mean response 10
- The observations in Region R_2 have mean response 21
- Then

for any new observation of X in region R_1 we would predict 10, and would predict 21 for new observations in region R_2

Example: Baseball Players' Salaries



Example: Baseball Players' Salaries



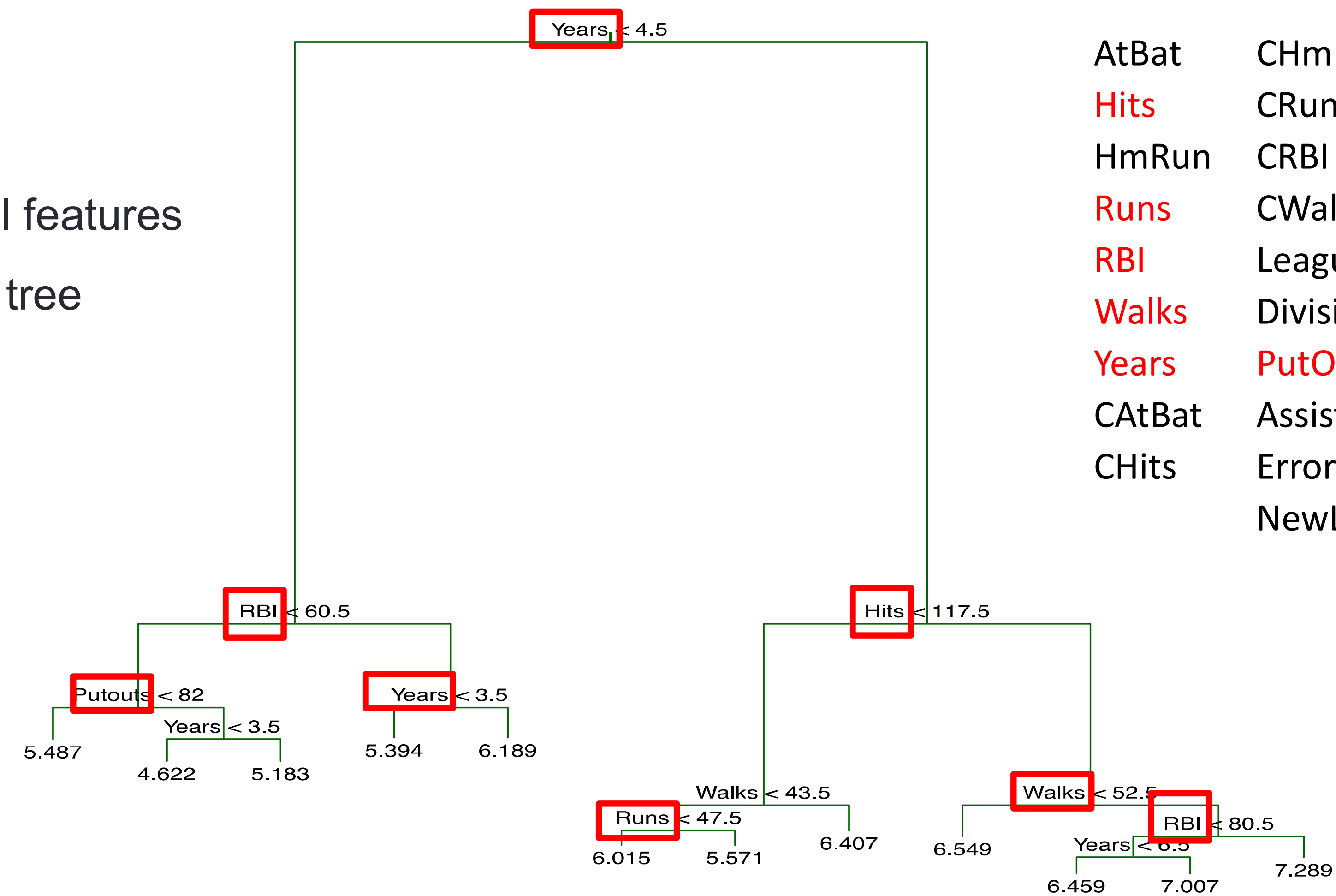
Predictions are

Average salary in each Region

Example: Baseball Players' Salaries

Not all features

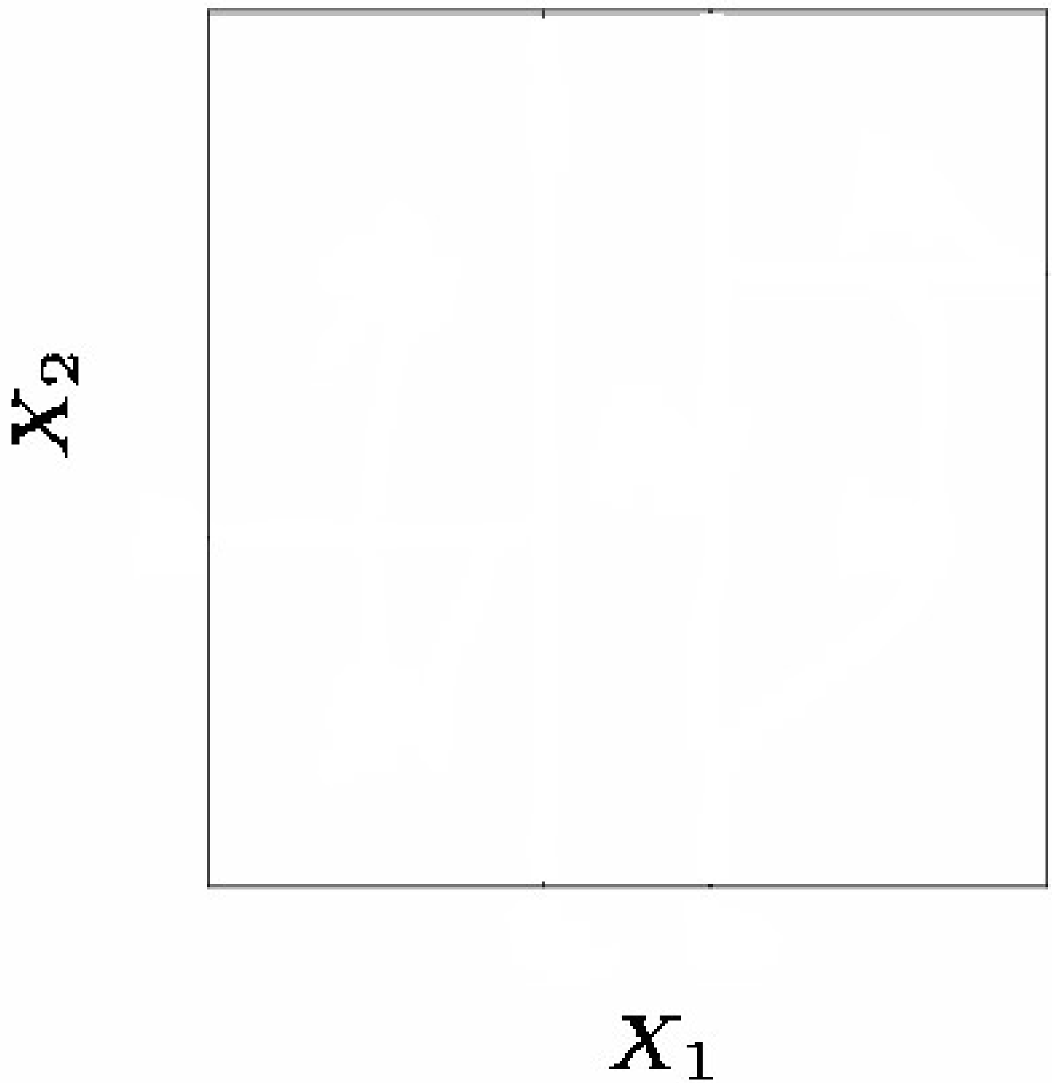
in the tree



AtBat	CHmRun
Hits	CRuns
HmRun	CRBI
Runs	CWalks
RBI	League
Walks	Division
Years	PutOuts
CAtBat	Assists
CHits	Errors
	NewLeague

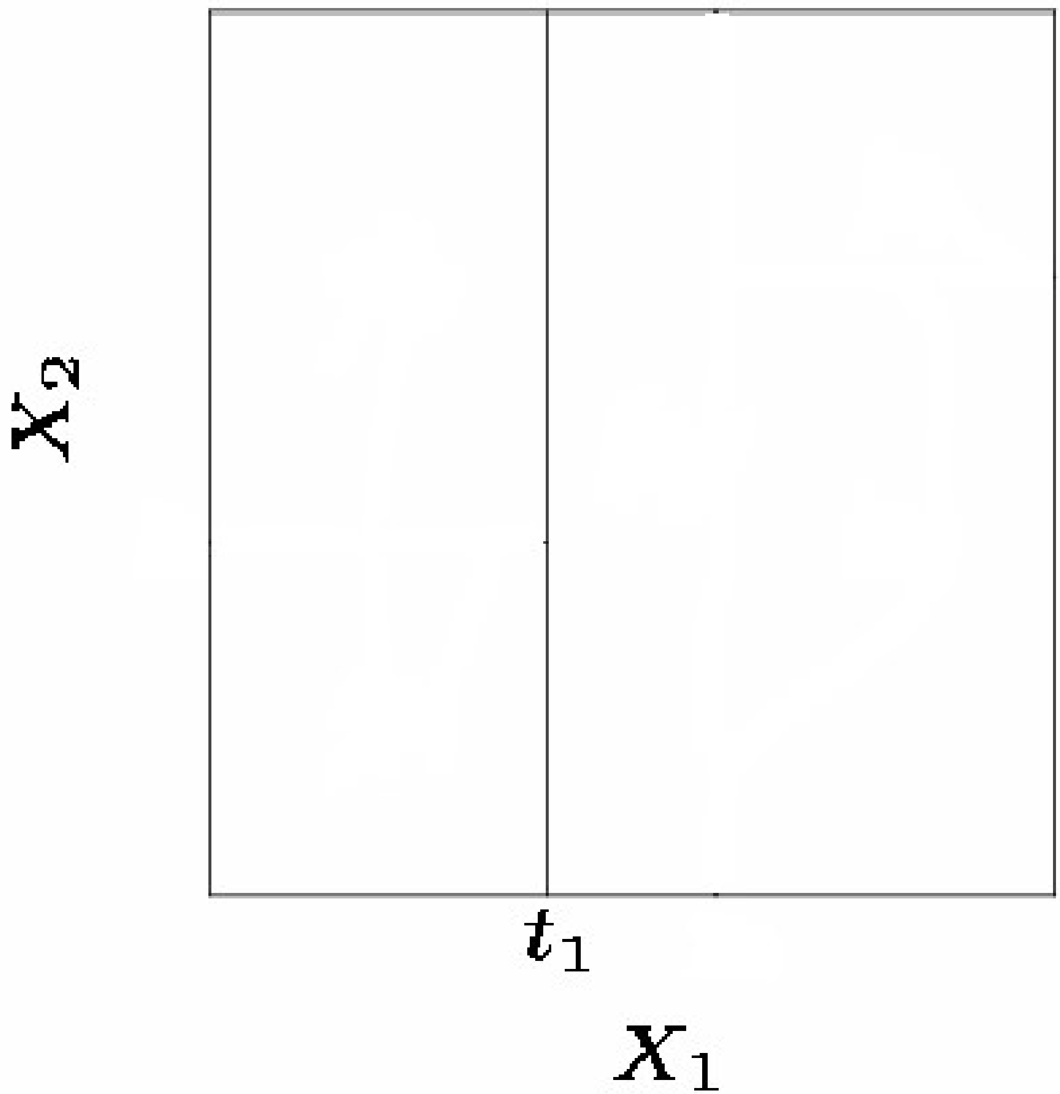
Partitioning Up the Predictor Space

- Consider two predictors X_1, X_2
- Regions are created by iteratively splitting one of the X variables into two regions
- MSE is



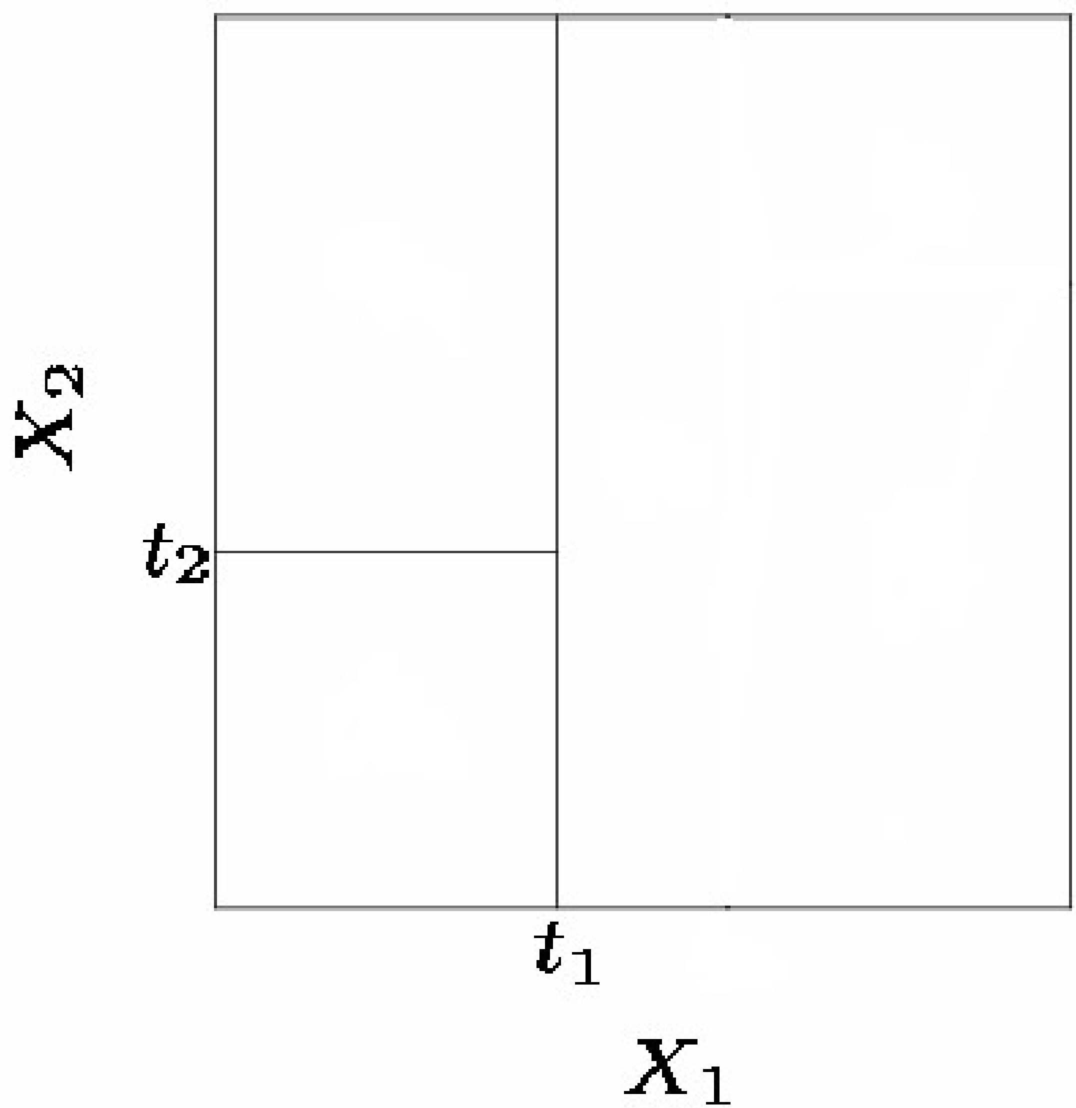
Partitioning Up the Predictor Space

1. First split on $X_1 = t_1$



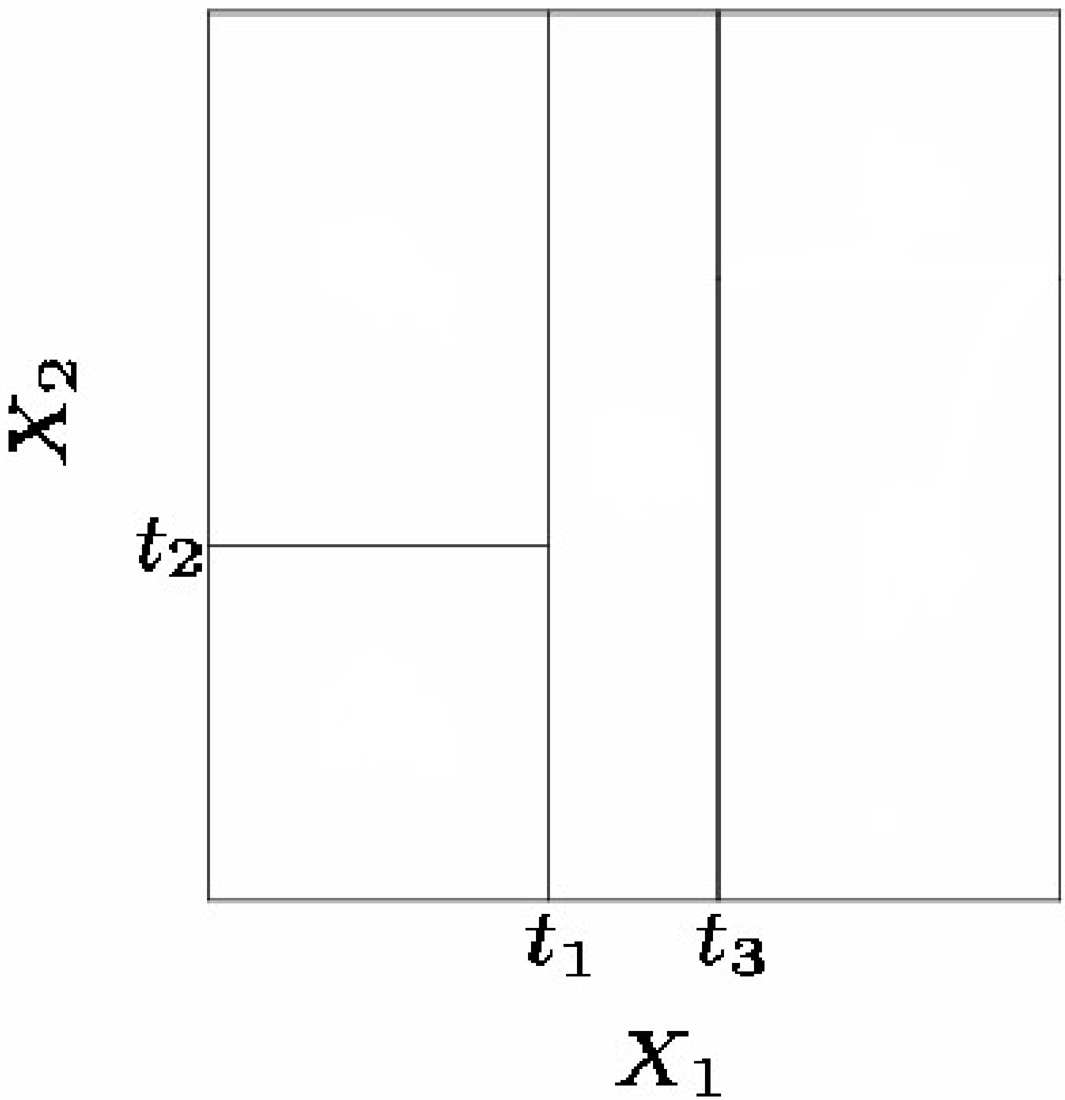
Partitioning Up the Predictor Space

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



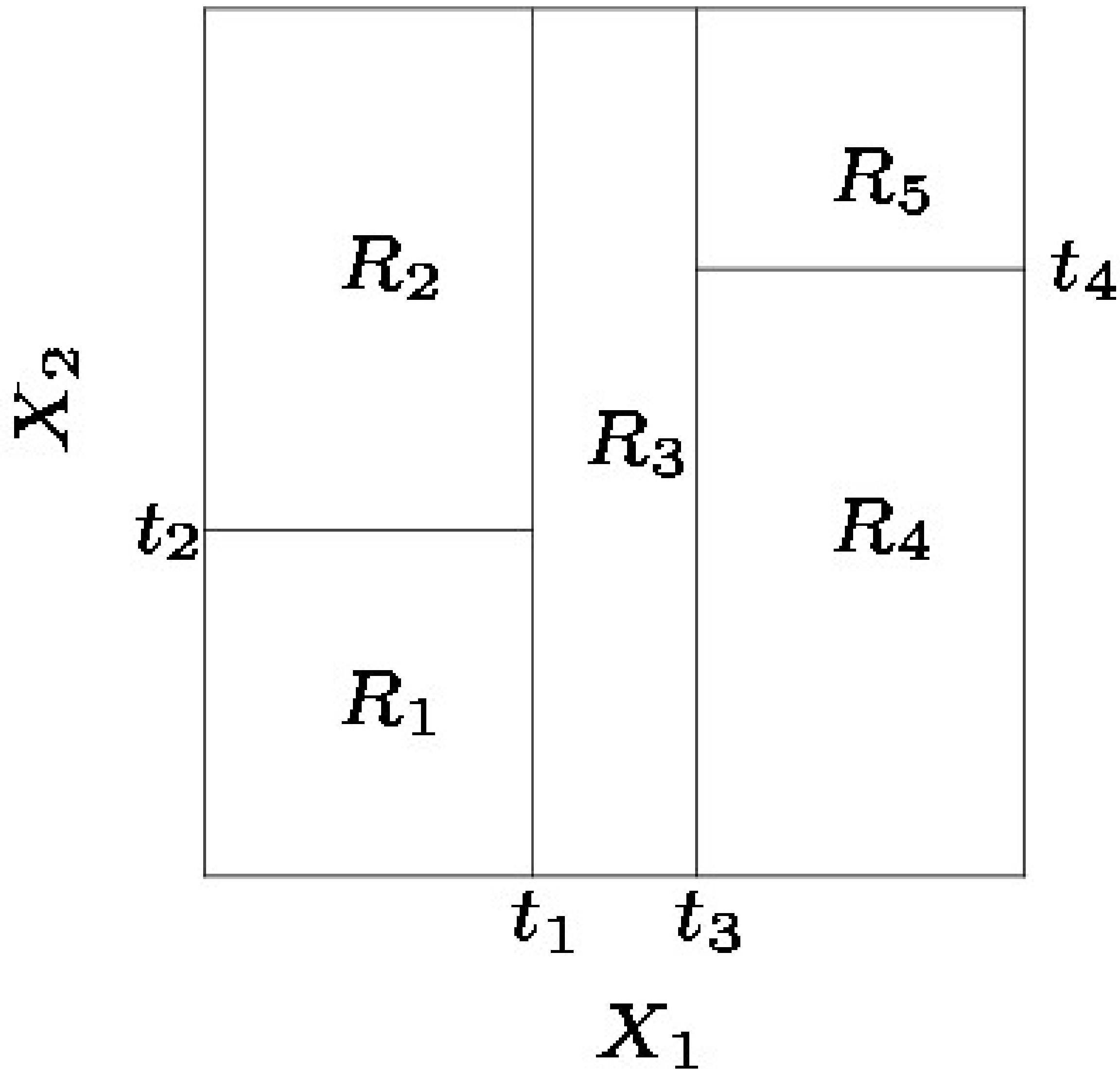
Partitioning Up the Predictor Space

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



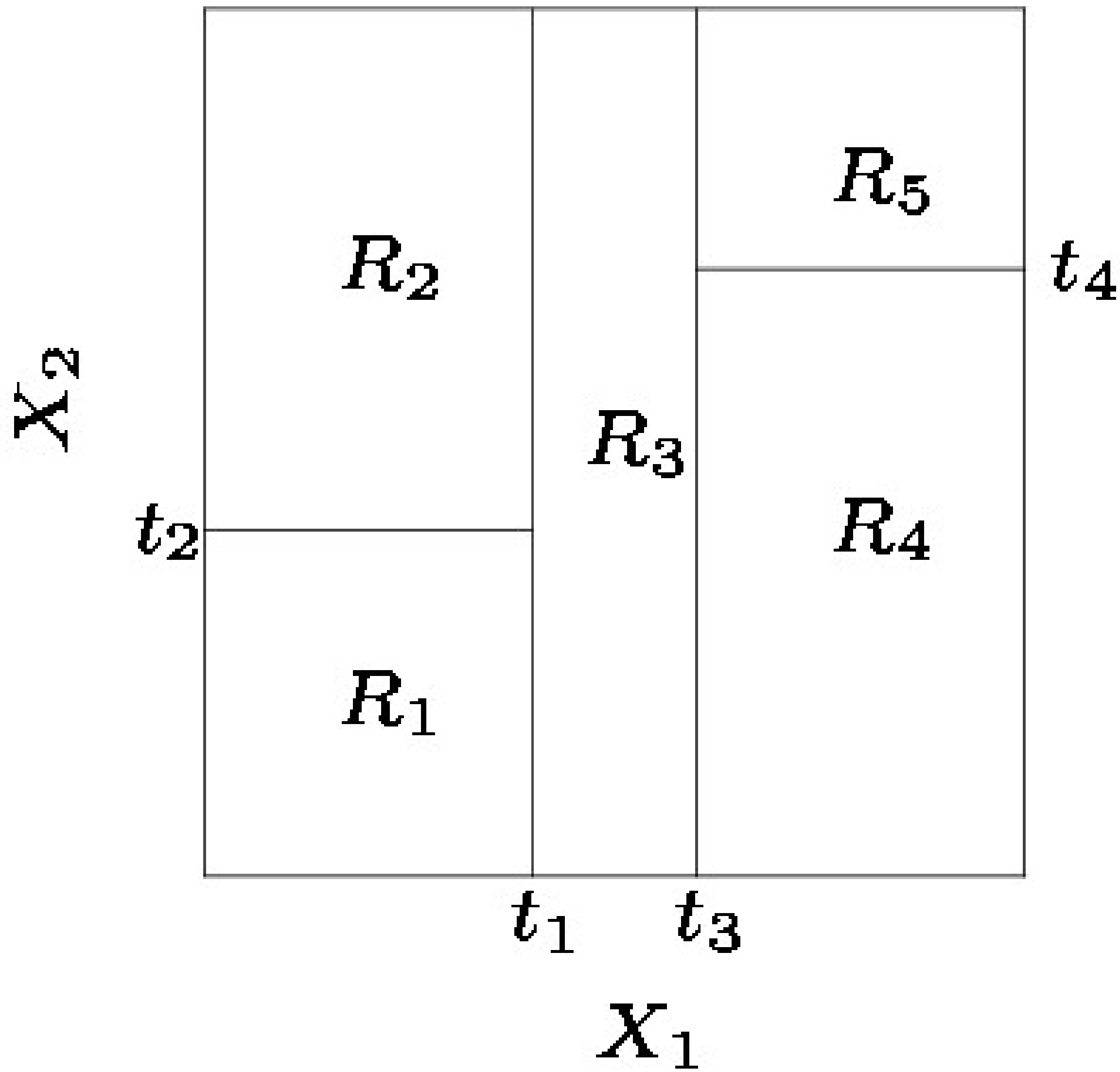
Partitioning Up the Predictor Space

1. First split on $X_1 = t_1$
2. If $X_1 < t_1$, 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 $X_2 = t_4$

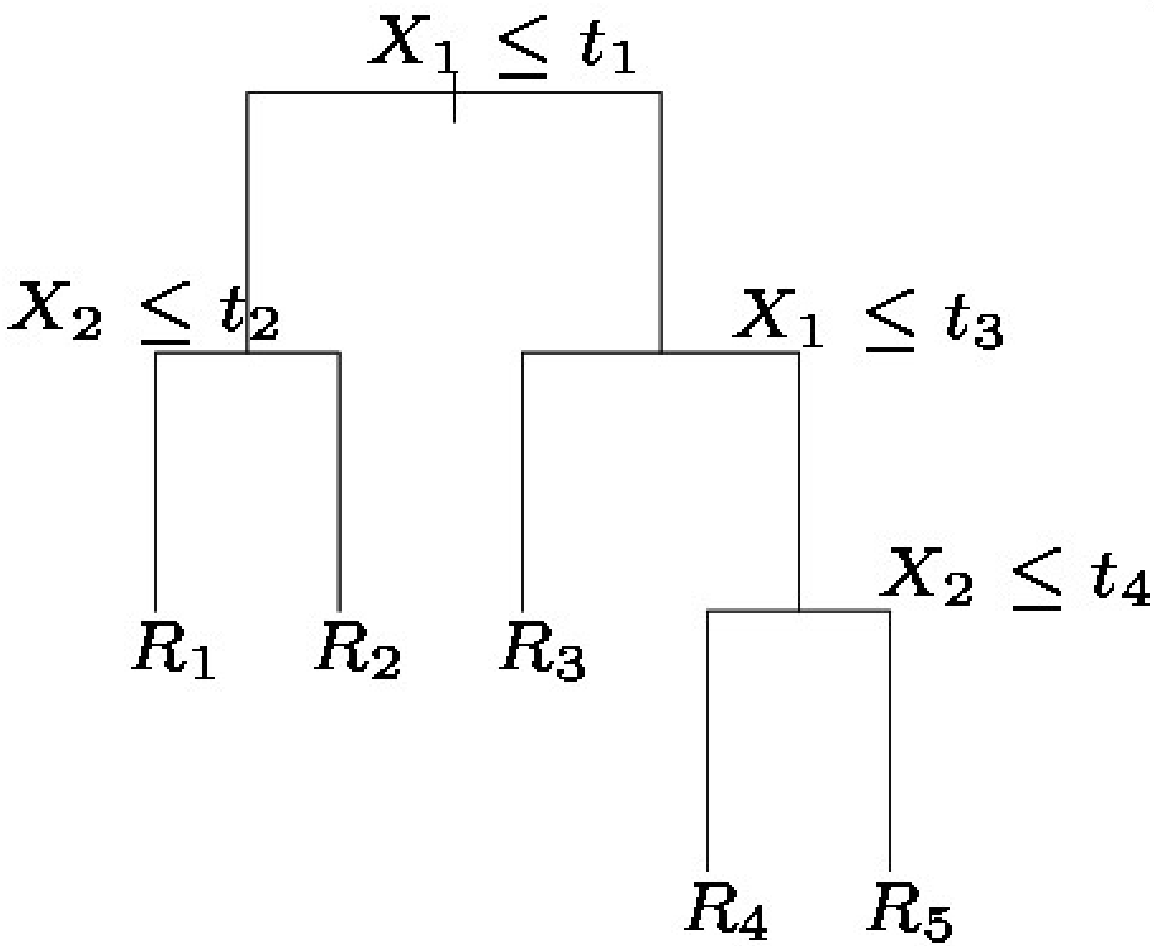


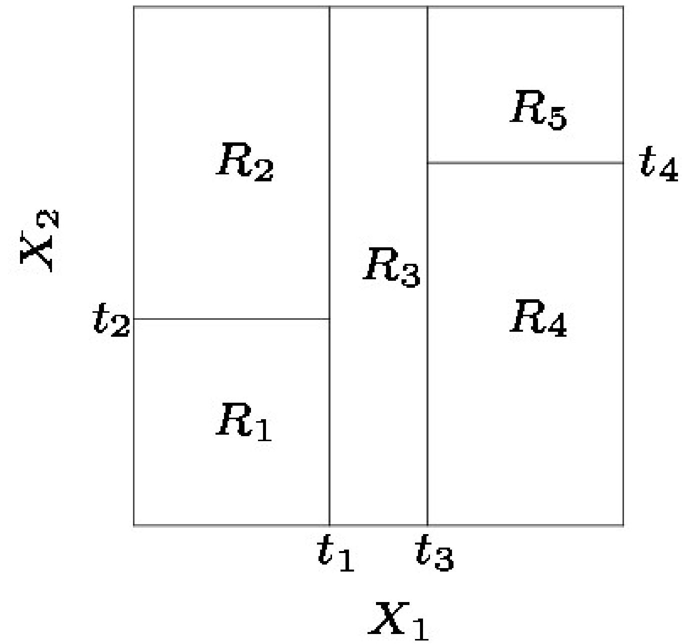
Partitioning Up the Predictor Space

1. First split on $X_1=t_1$
2. If $X_1<t_1$, 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 $X_2=t_4$
5. stop



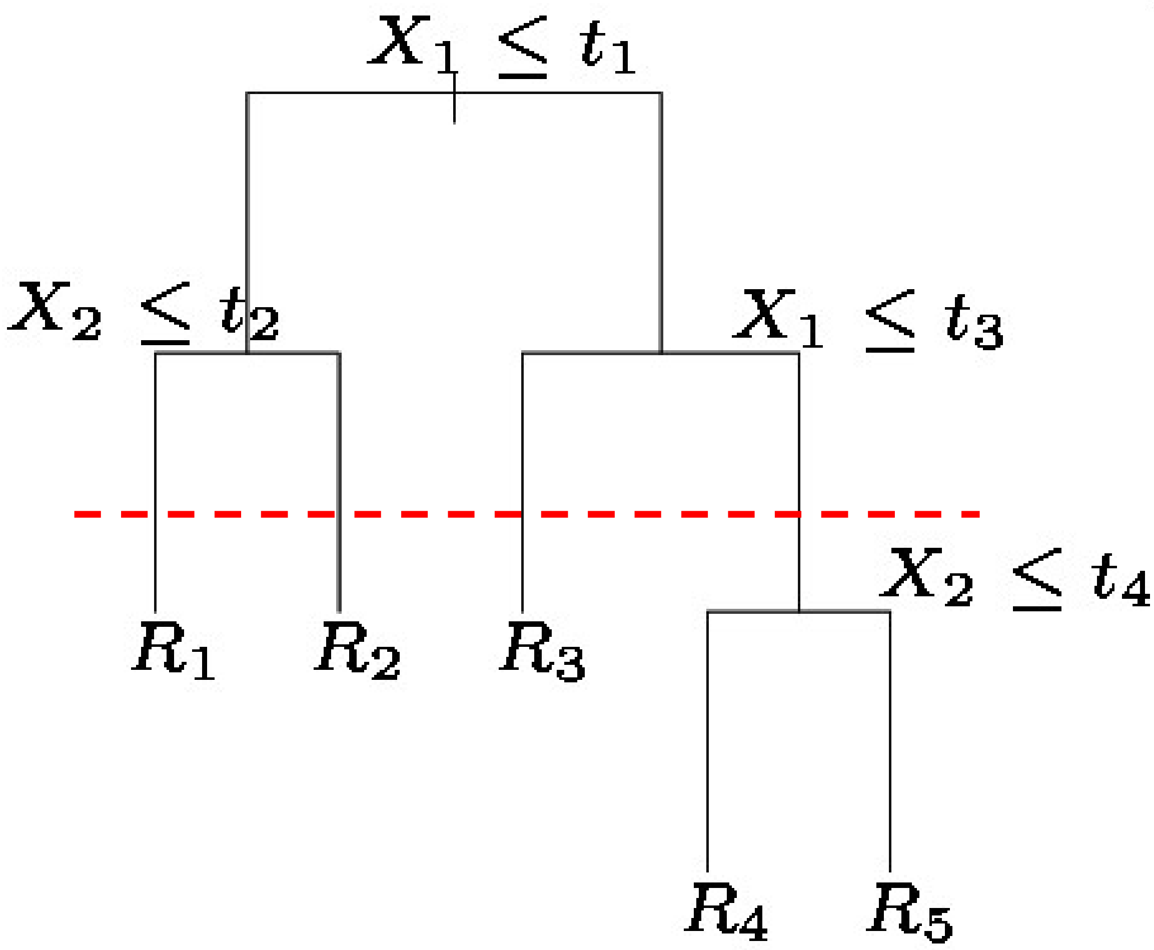
Decision Tree

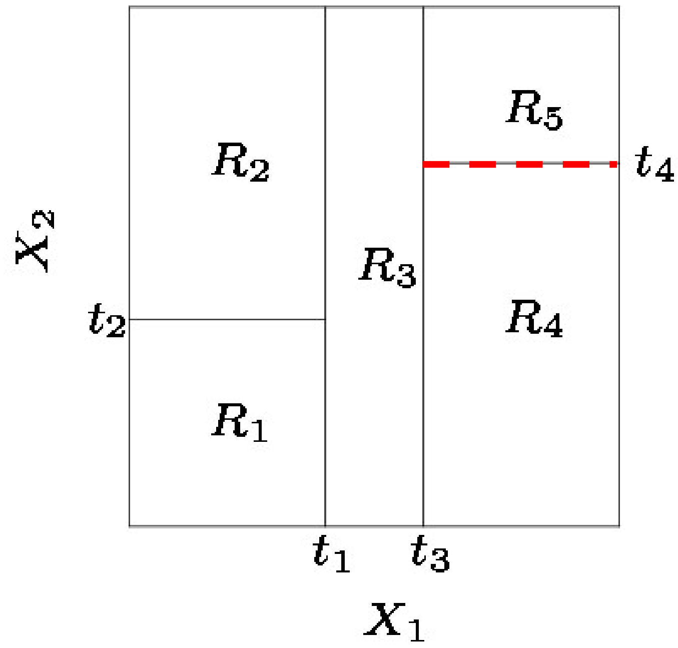




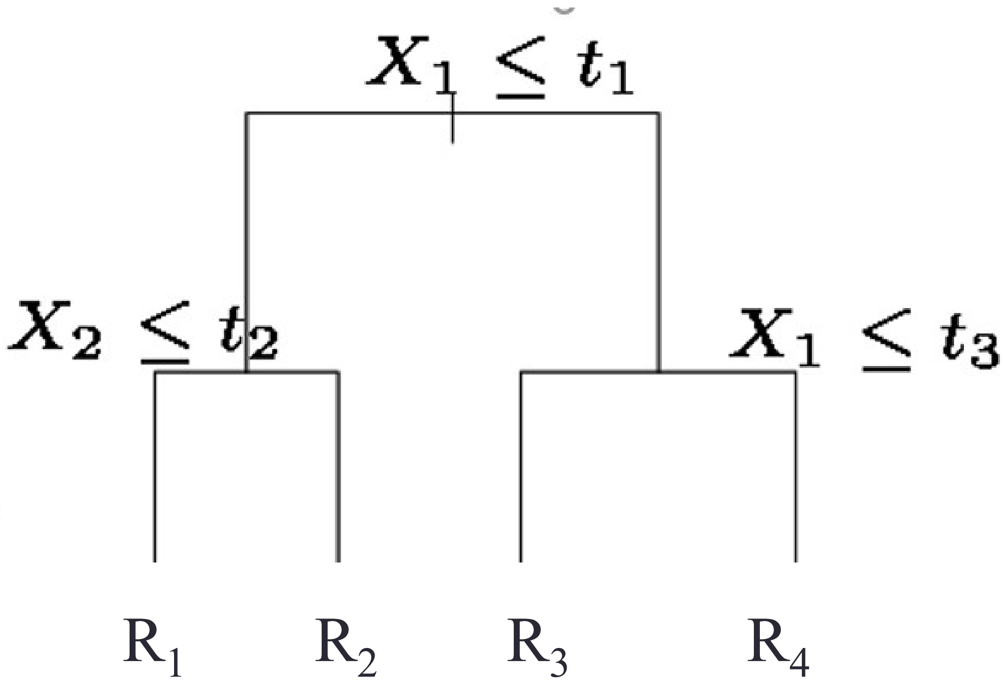
n. terminal nodes
=
n. regions

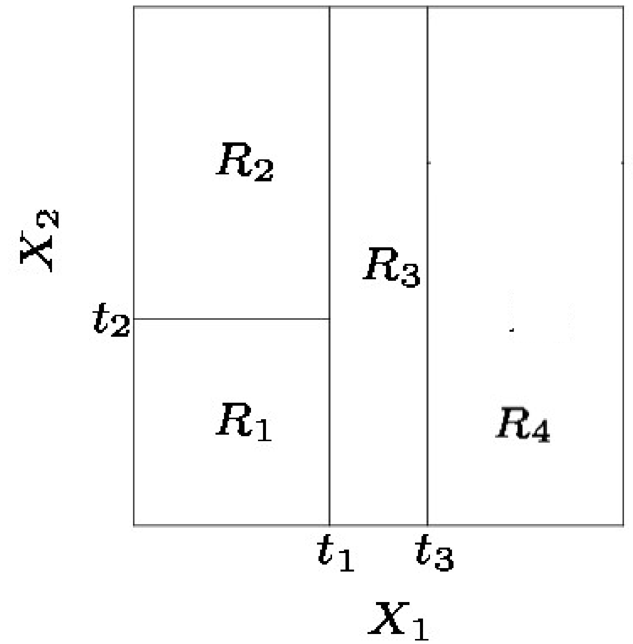
Decision Tree



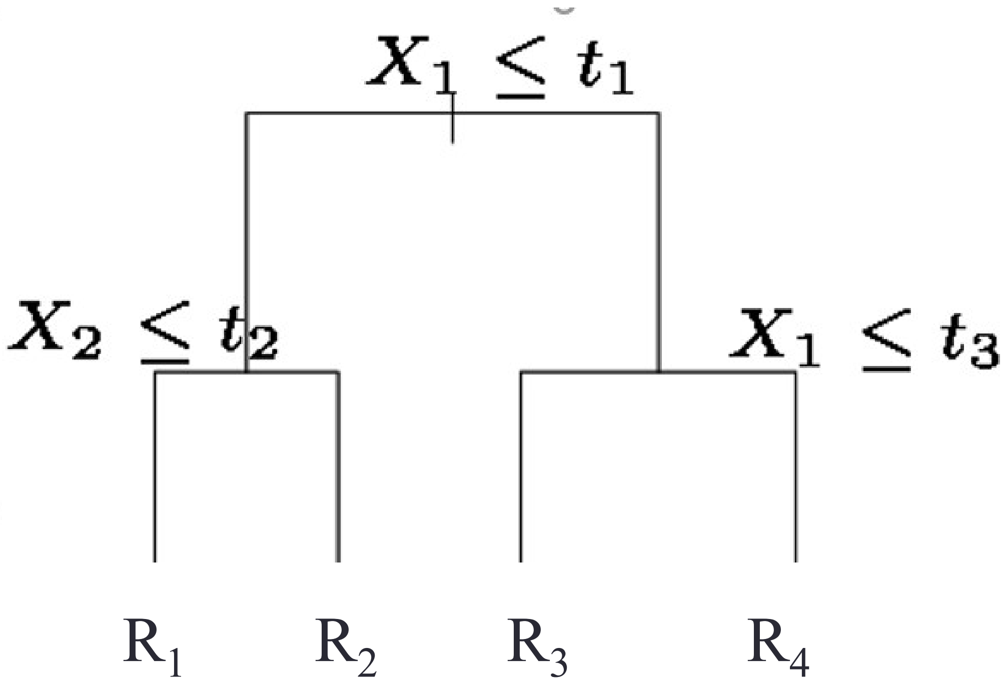


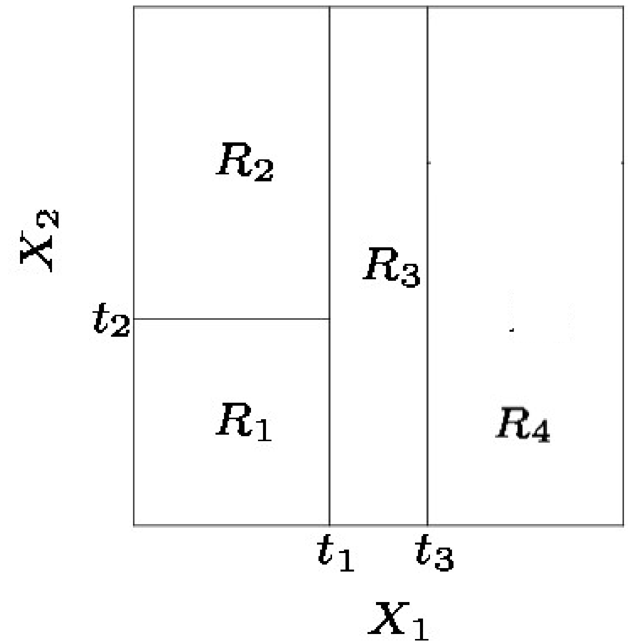
Pruned Tree





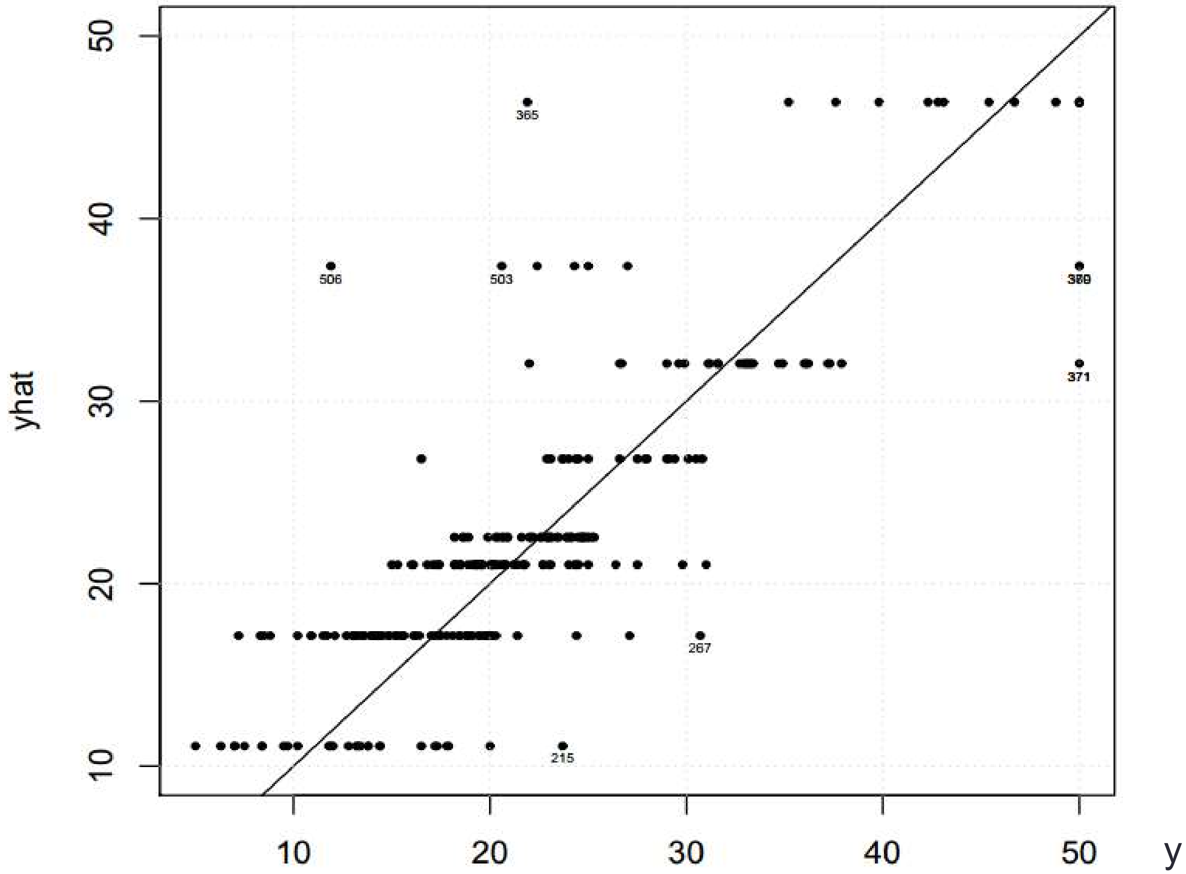
Decision Tree





n. of terminal regions = n. of predicted values

Decision Tree



n. of terminal regions = n. of predicted values

Stopping criteria

- As the number of splits increase,

  the number of observations in the splited regions decrease

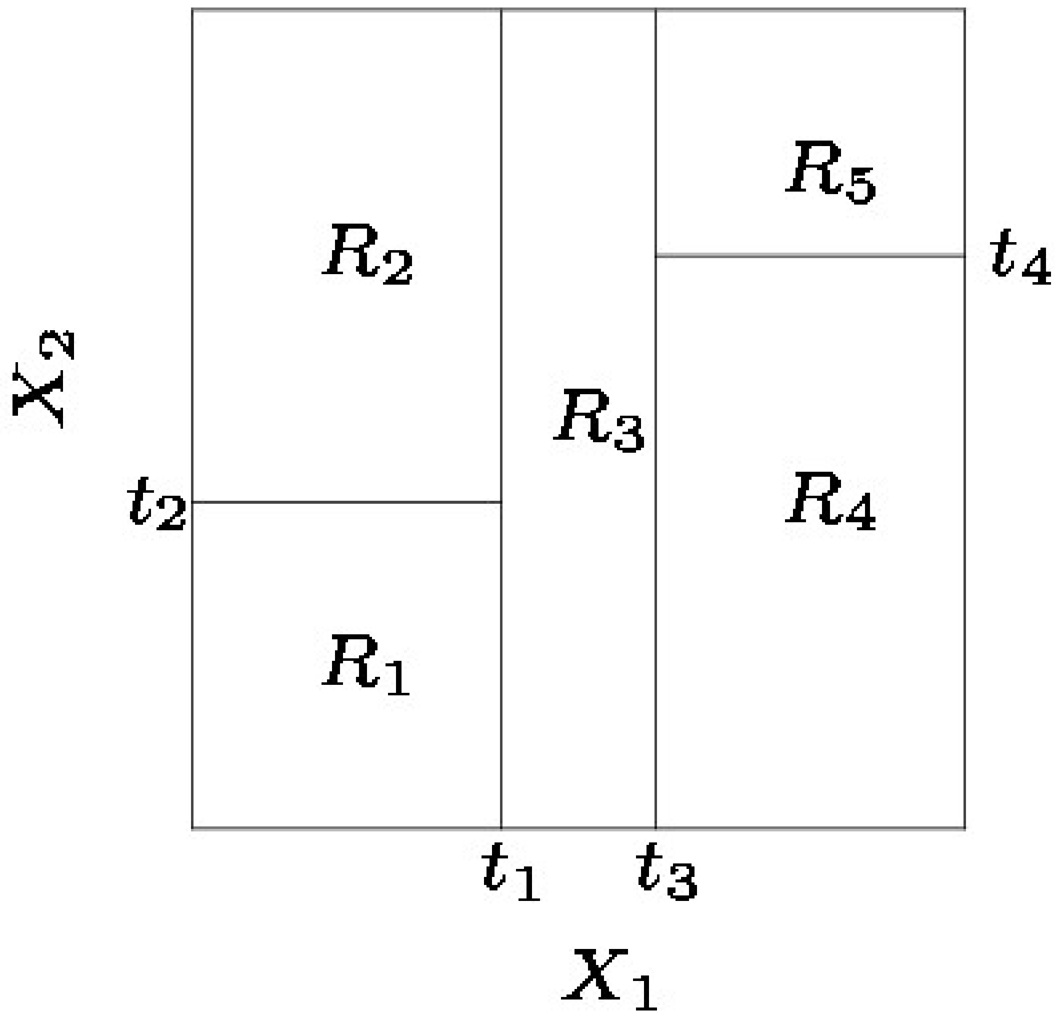
- Criteria 1: Stop when the max number of obs is equal to a threshold number

- Criteria 2: Fix the number of splits

Rectangular regions

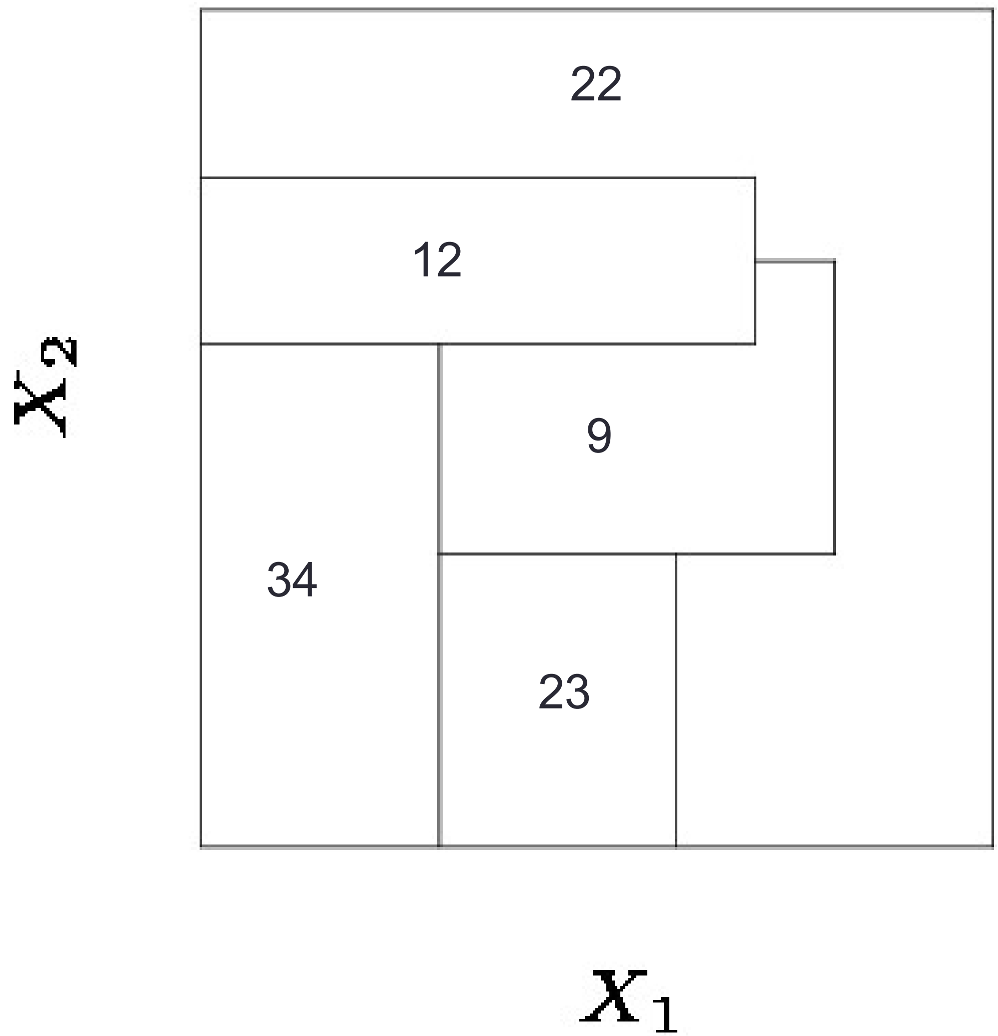
- CART models partition the predictor space into regions with special shape

- Regions are always rectangular and disjoint



Not possible

- This partitioning cannot result from a regression tree
- Region 9 is not rectangular



CLASSIFICATION TREES

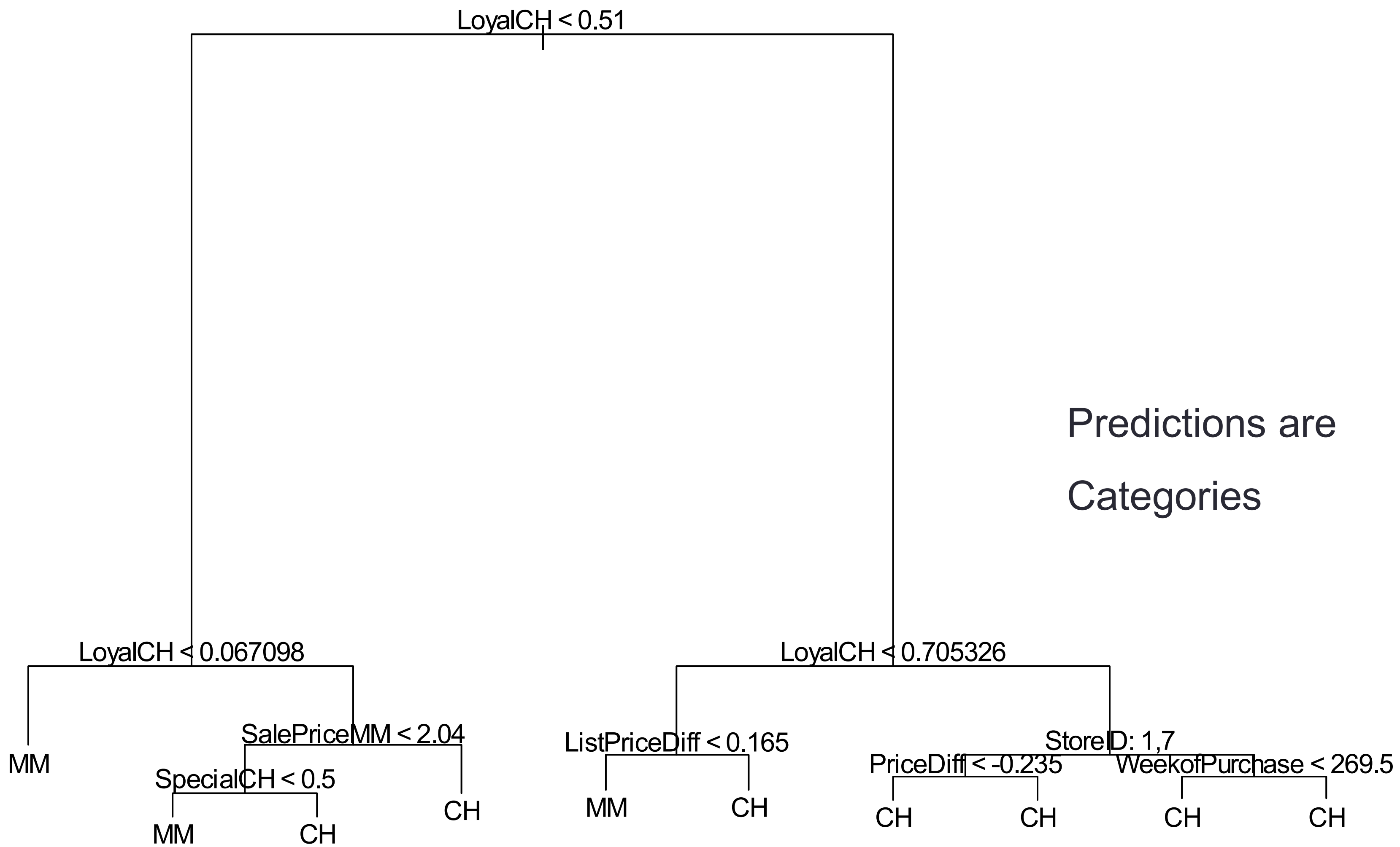
Classification Tree

- Classification Trees can be used with a categorical response with two or more categories

- The tree is grown (i.e. the splits are chosen) the same way as a regression tree is grown

- For each region the prediction is the most common category in that region.

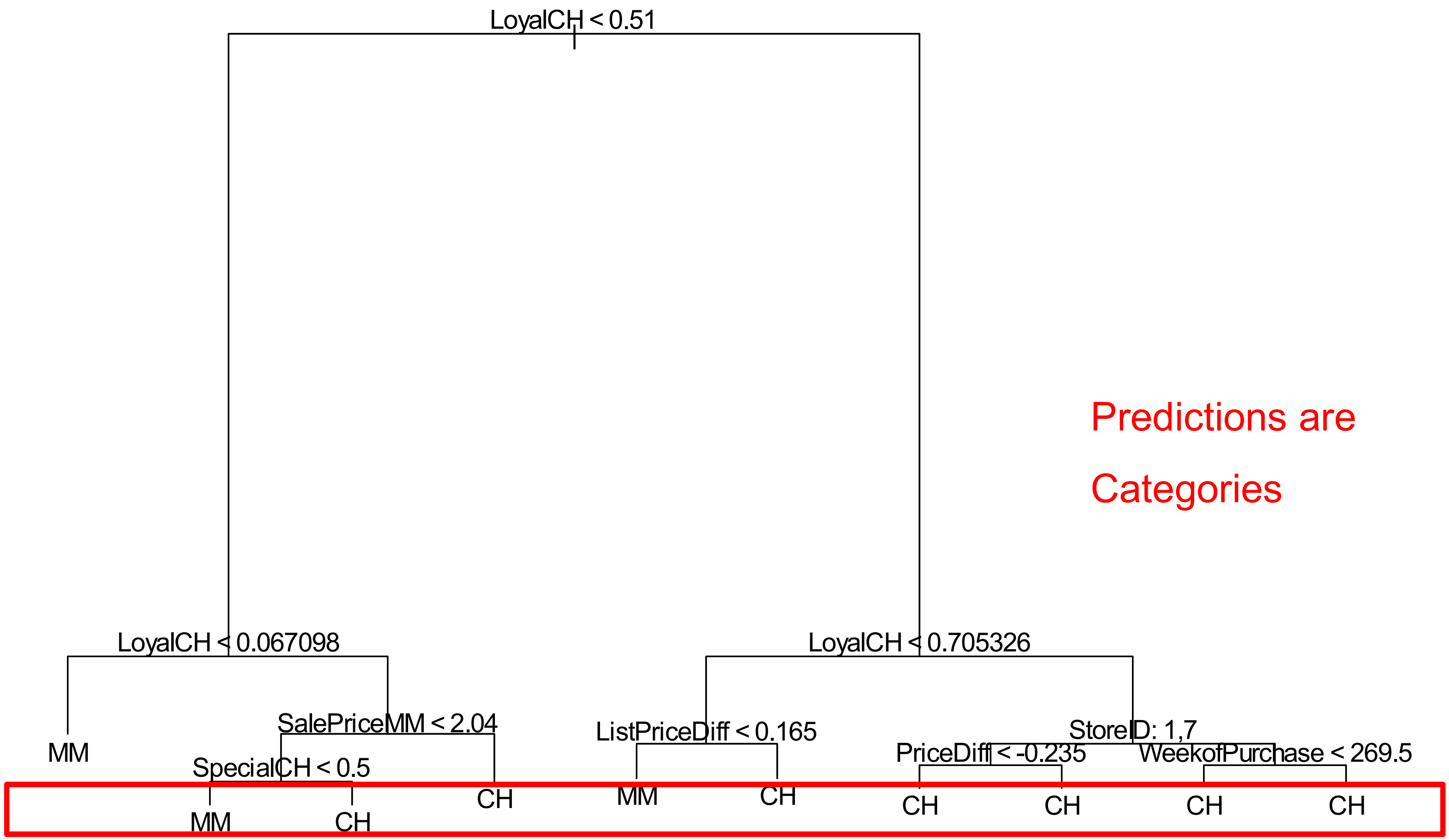
Example



Predictions are

Categories

Example



Predictions are

Categories

BAGGING

Bagging for Regression Trees

- Construct B regression trees using B bootstrapped training datasets
- Average the resulting predictions

- These trees are not pruned, so each individual tree has high variance but low bias. Averaging these trees reduces variance, and thus we end up lowering both variance and bias

Bagging for Classification Trees

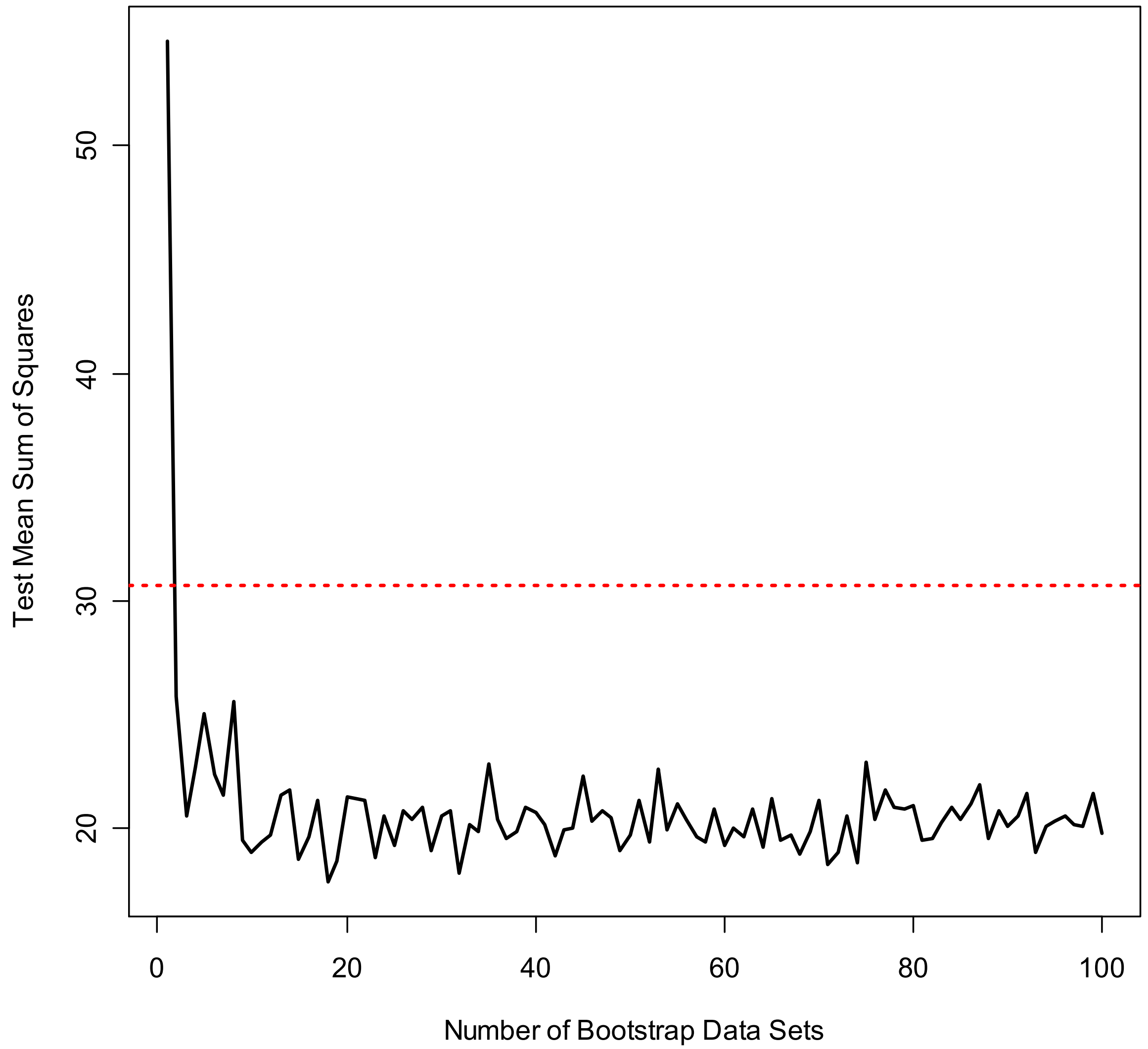
- Construct B regression trees using B bootstrapped training datasets

- For prediction, there are two approaches
  1. Record the category that each bootstrapped data set predicts and provide an overall prediction to the most commonly occurring category (majority vote)
  2. If the classifier produces probability estimates we can just average the probabilities and then predict to the class with the highest probability

Example

- The red dashed line represents the test MSE using a single tree

- The black line shows changes in the test MSE as B increases



Variable Importance Measure

- Bagging typically improves the accuracy over prediction using a single tree, but it is harder to interpret the model!
- With hundreds of trees, and it is no longer clear which variables are most important for prediction
- Thus bagging improves prediction accuracy at the expense of interpretability
- We can still get an overall summary of the importance of each predictor using *Relative Influence Plots*

Relative Influence Plots

- How do we decide which predictors (features) are most important in predicting the response?

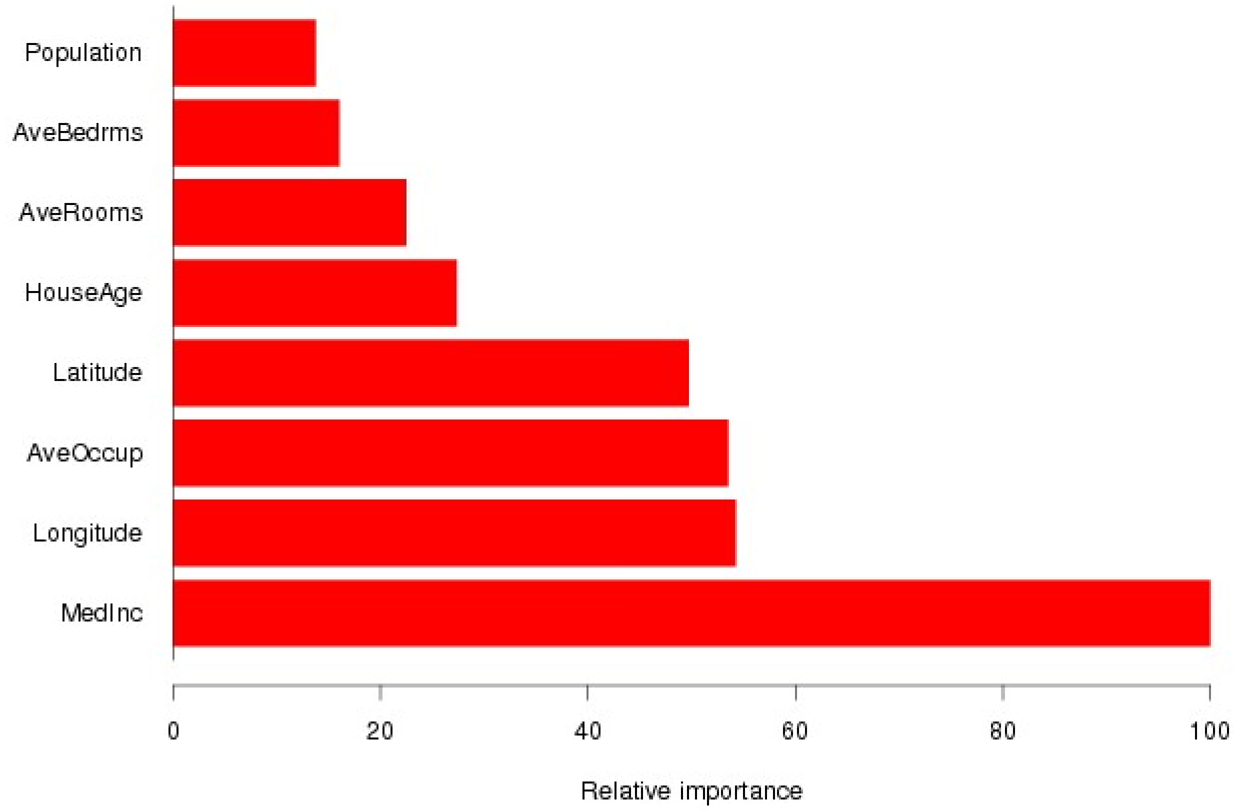
- For each feature
  - measure the decrease in MSE when splitting on a particular feature
  - Sum all decrements
  - A number close to zero indicates the feature is not important and could be dropped
  - The larger the score, the more influential and important, the feature is

- Plot the sum of decrements of each predictor

Example: Housing Data

- Median Income is by far the most important variable

- Longitude, Latitude and Average occupancy are the next most important.



RANDOM FORESTS

Random Forests

- It builds on the idea of bagging, but it provides an improvement because it de-correlates the trees

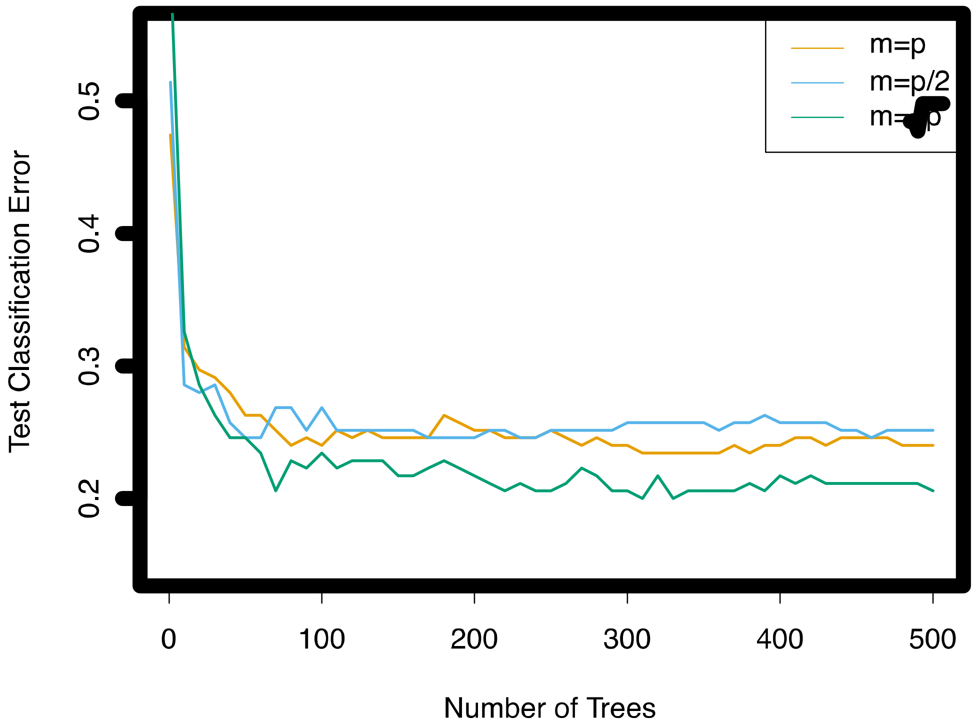
- How does it work?
  - each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors (Usually $m \approx \sqrt{p}$)

Why are we considering a random sample of m predictors instead of all p predictors for splitting?

- Suppose that we have a very strong predictor in the data set along with a number of other moderately strong predictor, then in the collection of bagged trees, most or all of them will use that very strong predictor for the first split!
- All bagged trees will look similar. Hence all the predictions from the bagged trees will be highly correlated
- Averaging many highly correlated quantities does not lead to a large variance reduction
- By selecting the predictors from different subsets of predictors, Random Forest “de-correlates” the bagged trees leading to reduction in variance

Random Forest with different values of “m”

- When random forests are built using m = p, then this amounts simply to bagging



Random Forest with different values of "m"

- As the number *m* decreases the test error rate decrease