UNIVERSITY OF AMSTERDAM
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 13.4 - Combining Models
Decision Trees - Random Forests

Erik Bekkers

(Bishop 14.4, Hastie-Tibshirani-Friedman 9.2)

Image credit: Kirillm | Getty Images

Slide credits: Patrick Forré, Rianne van den Berg and the ***MOOC by Hastie and Tibshirani***

Regression with GP's

- Combining models: (Bishop 4.1-4.4)
  - Bayesian model averaging vs. model combination methods
  - Committees:
    - Bootstrap aggregation
    - Random subspace methods
    - Boosting
  - **Decision trees**
  - Random forests

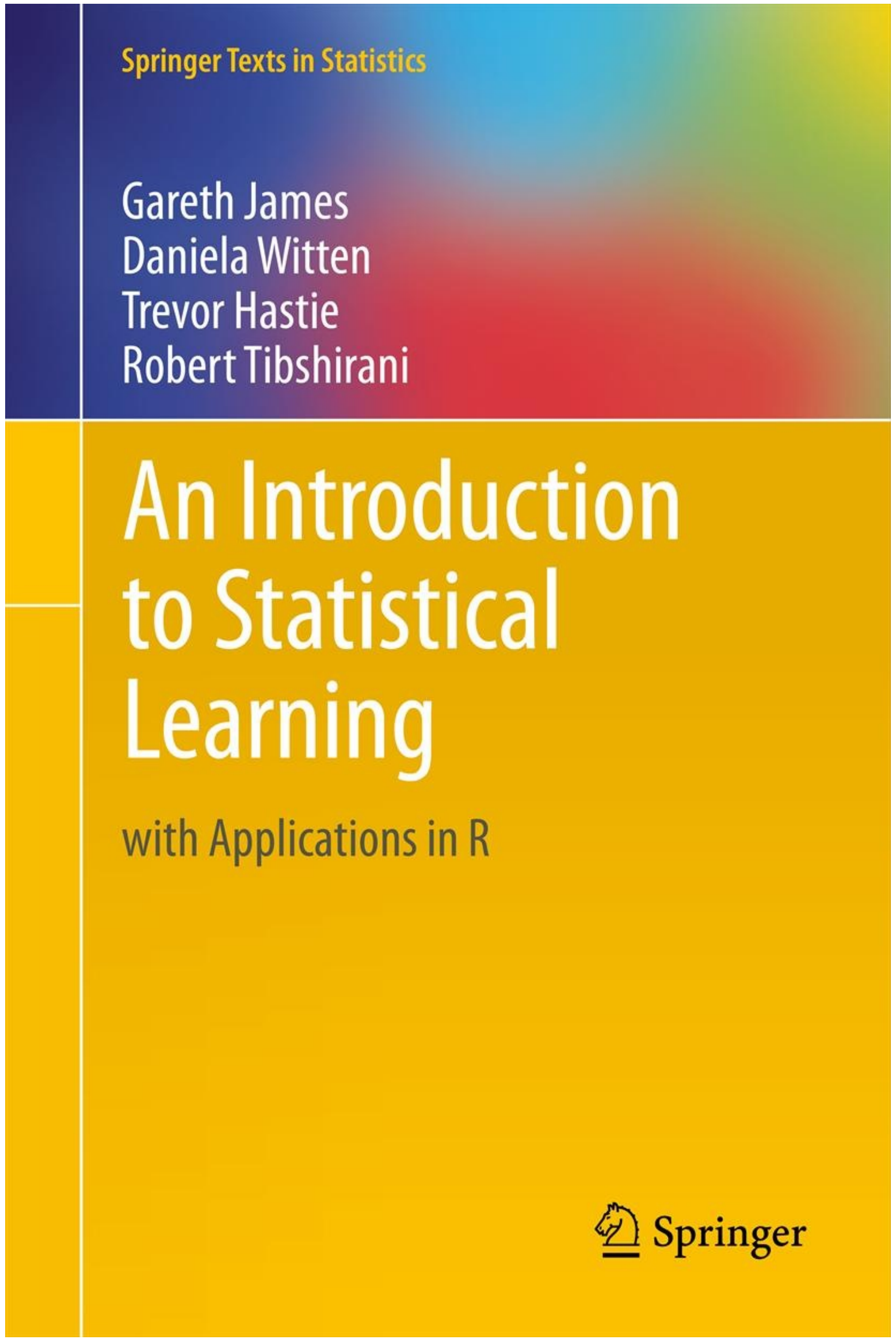
Introduction to Statistical learning (ch 8)

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani,

Introduction to Machine learning as a statistical tool.

See:
http://www-bcf.usc.edu/~gareth/ISL/

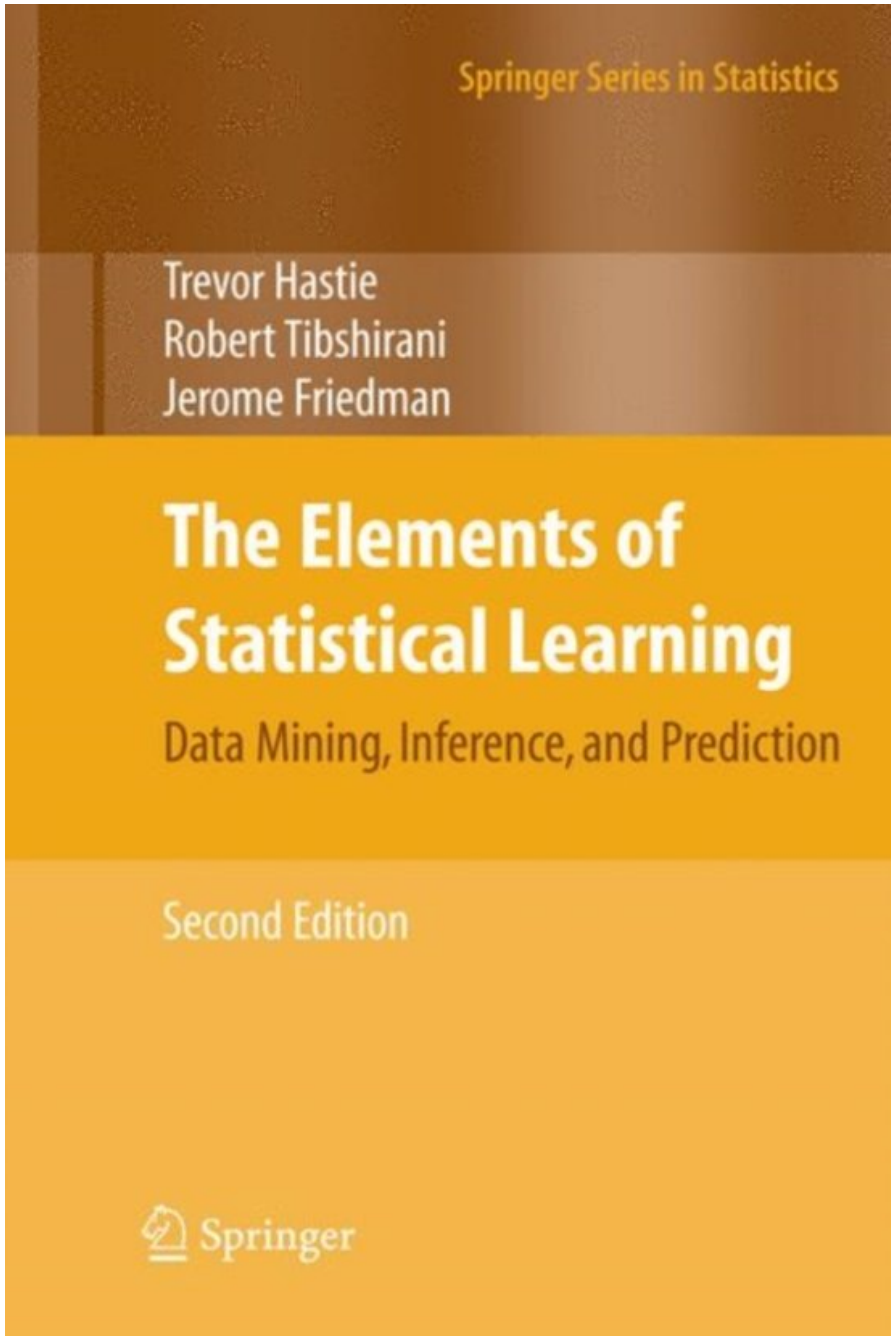
for pdf of book and MOOC by Hastie and Tibshirani



The elements of statistical learning (ch 9.2)

Trevor Hastie, Robert Tibshirani, Jerome Friedman

More advanced view of Machine learning as a statistical tool.



Decision Trees

Slides based on Stanford MOOC Statistical Learning (Ch 8)

- Applications: Regression & Classification
- Stratify/Segment input space into rectangular regions
- Splitting rules of input space can be summarized in tree
- **Pros and cons**
  - Simple and useful for interpretation
  - Not competitive with state of the art algorithms
  - Extensions such as bagging, random forests and boosting are ensemble methods that improve performance

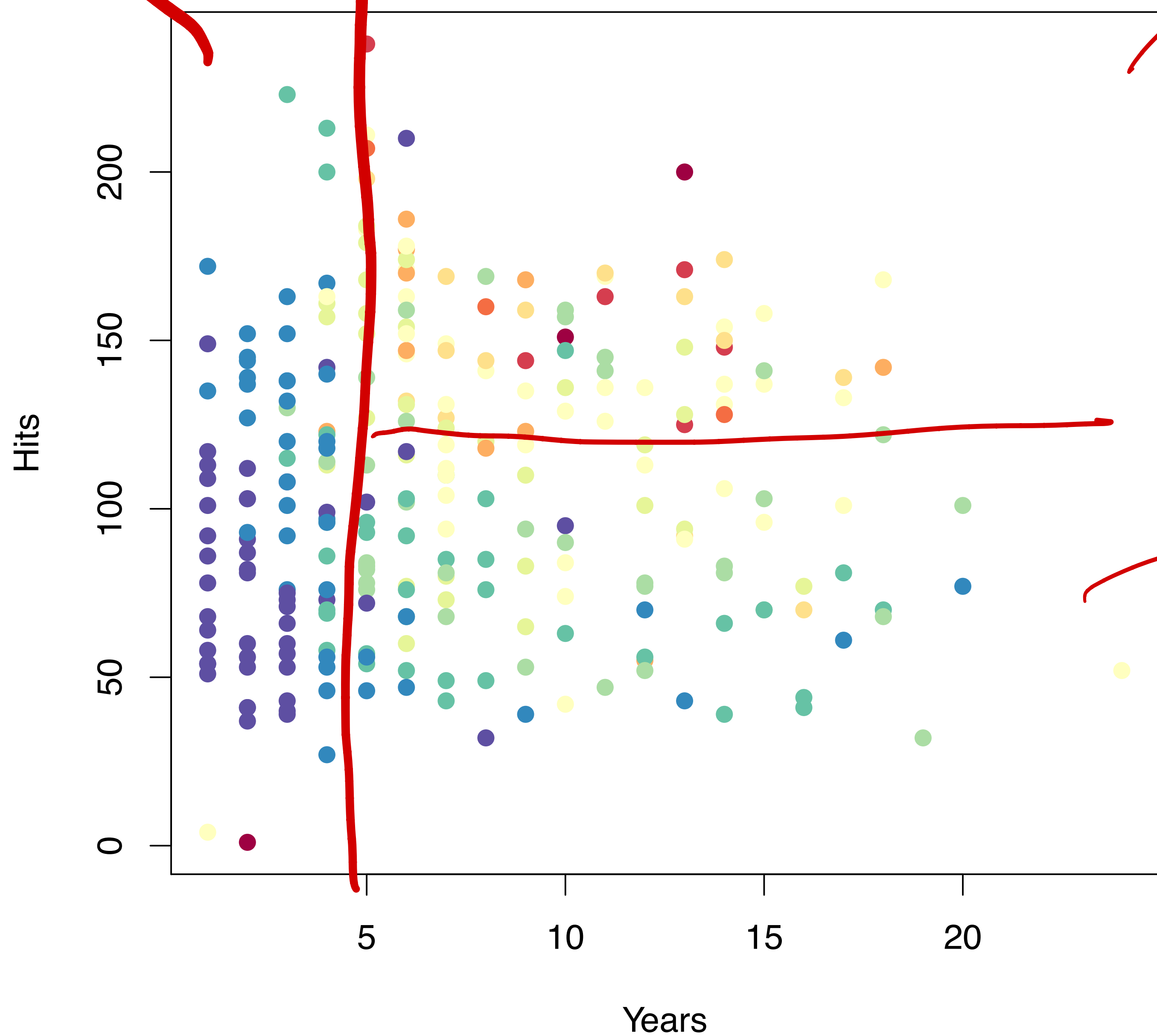
Decision Trees: Regression

R_1

$\underline{x} \in R_1$

$\rightarrow \hat{y}_{R_1}$

Baseball salary data:



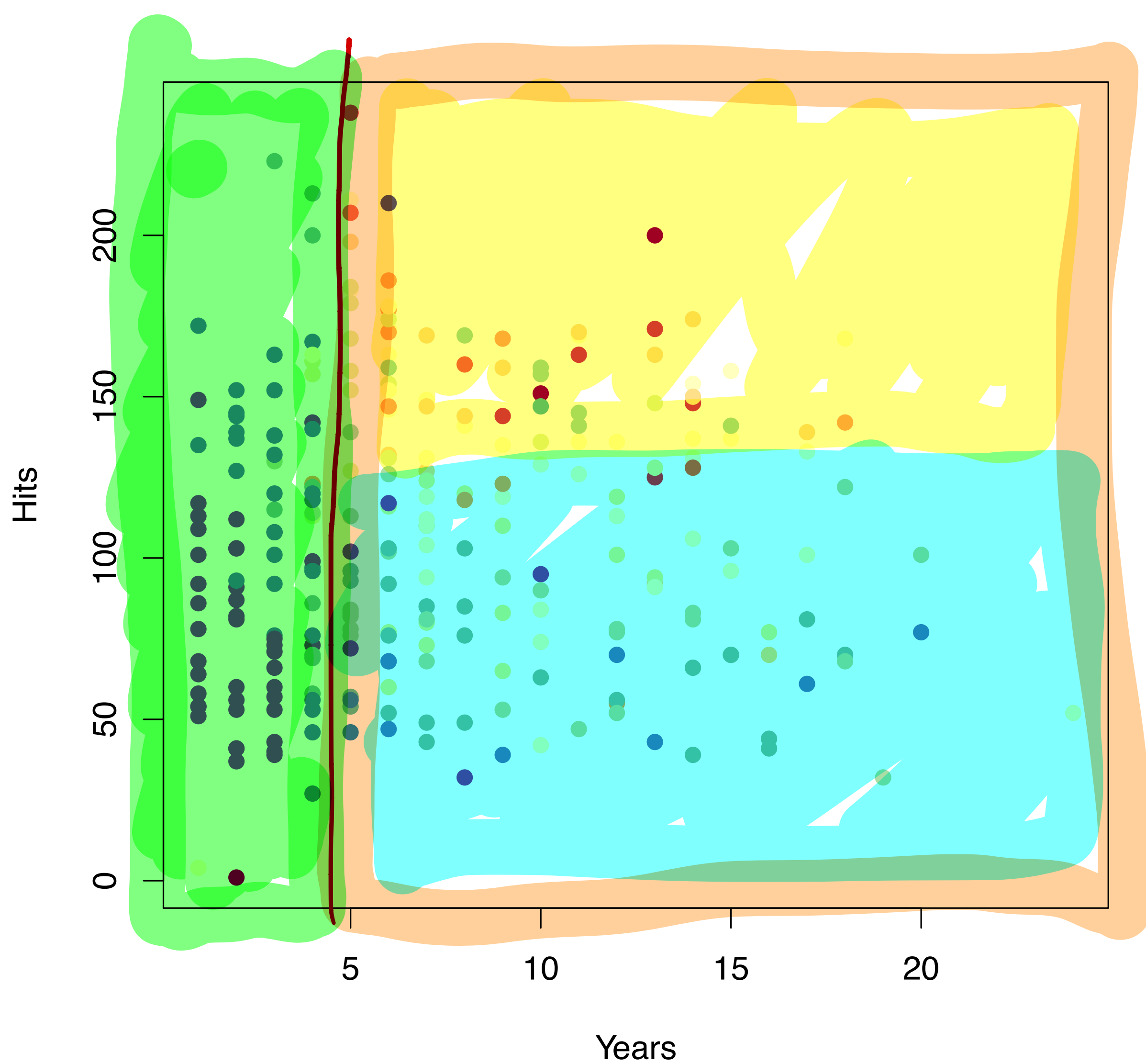
R_3

R_2

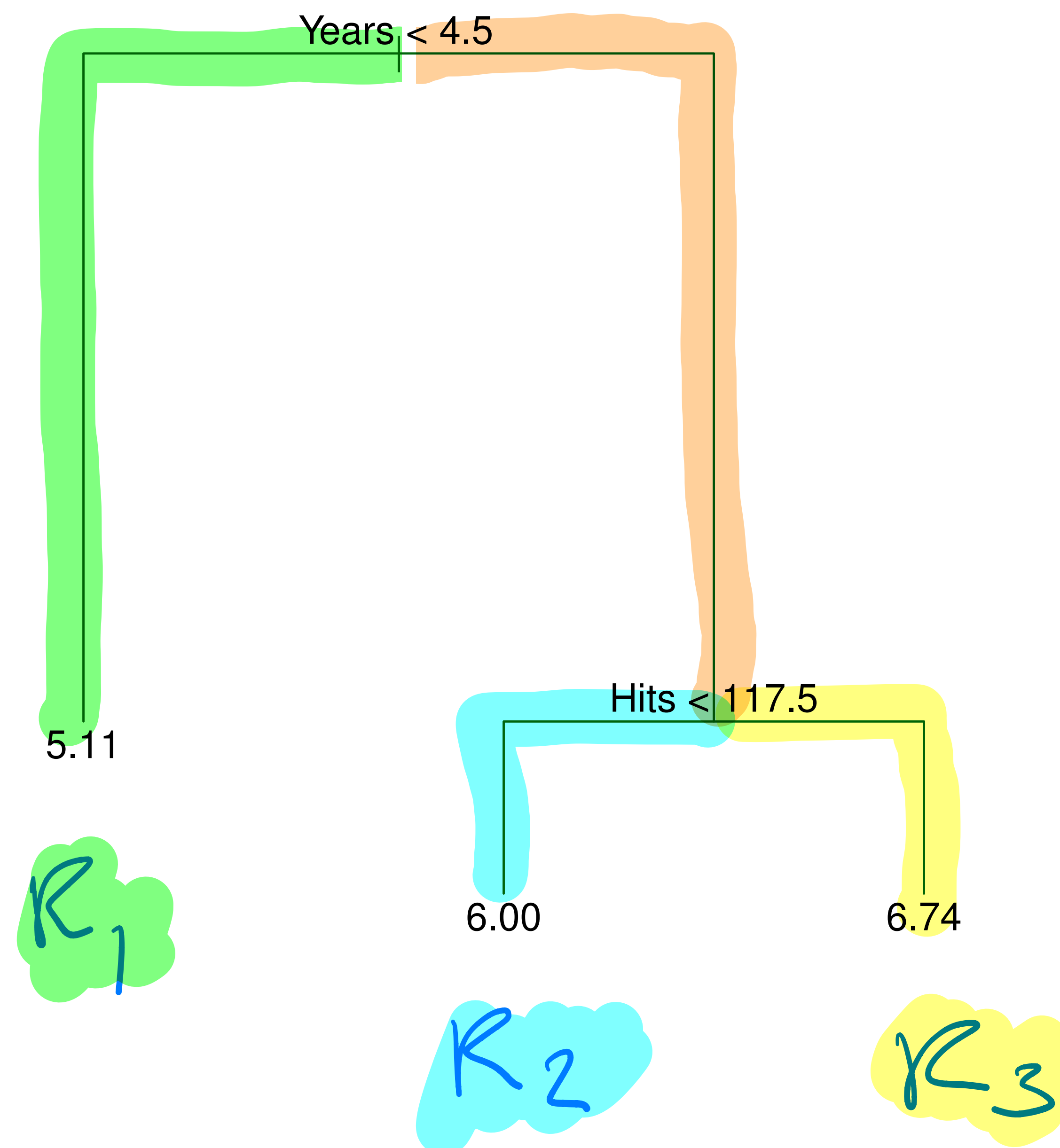
Salary is color-coded from low (blue, green) to high (yellow, red)

Basketball salary dataset [source: ISL Chapter 8]

Baseball salary dataset

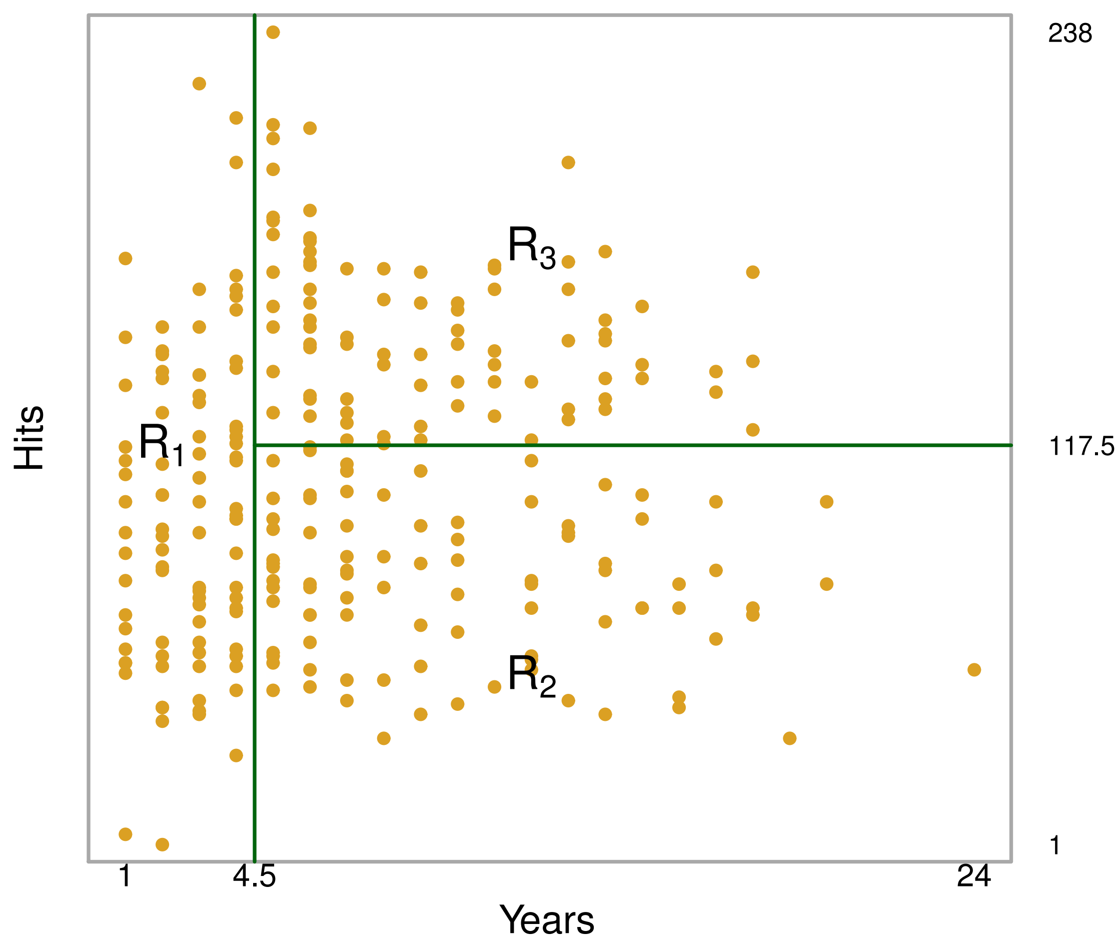


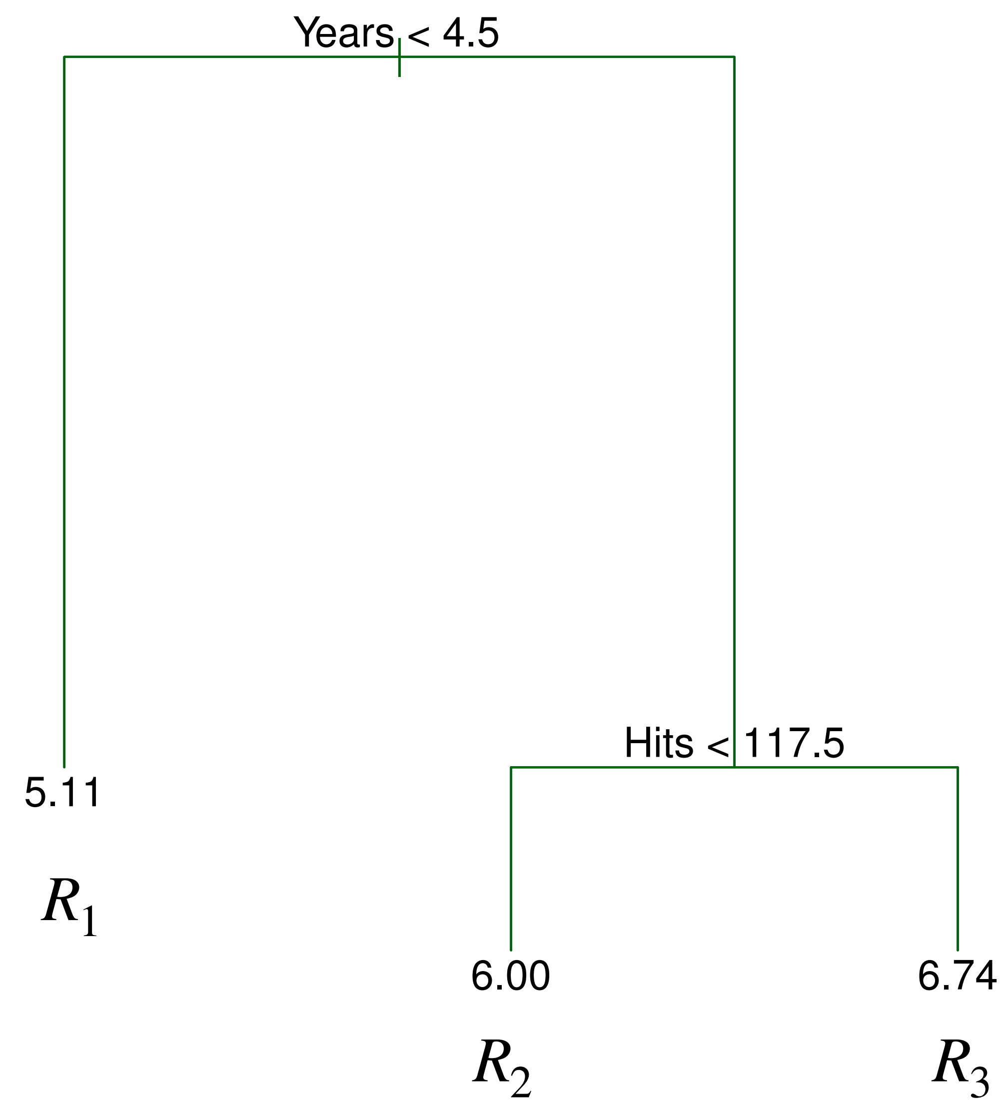
Basketball salary dataset [source: ISL Chapter 8]



Example decision tree [source: ISL Chapter 8]

Baseball salary dataset





$$R_1 = \{\mathbf{X} | \text{years} < 4.5\}$$

$$R_2 = \{\mathbf{X} | \text{years} \geq 4.5, \text{hits} < 117.5\}$$

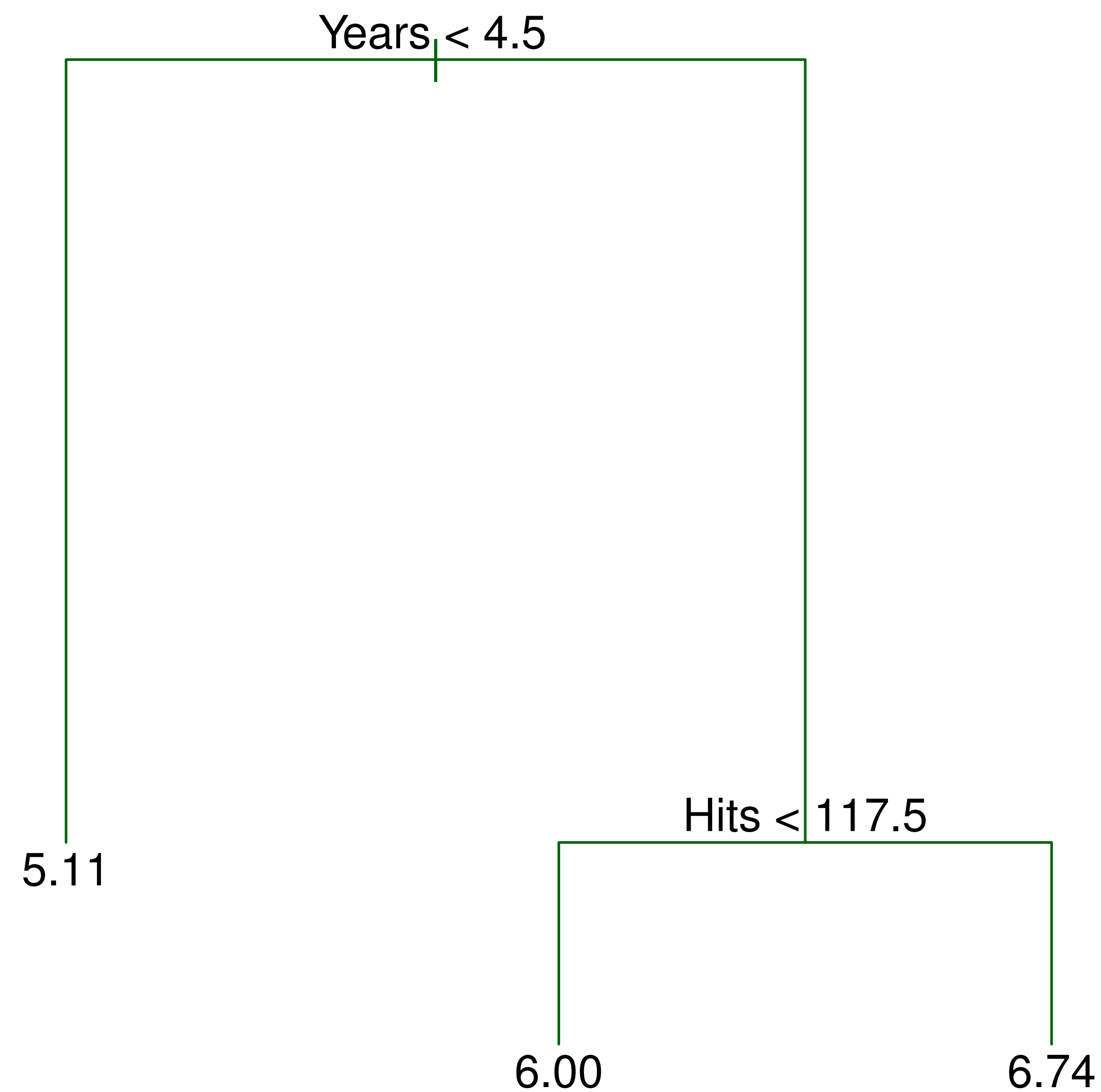
$$R_3 = \{\mathbf{X} | \text{years} \geq 4.5, \text{hits} \geq 117.5\}$$

Basketball salary dataset [source: ISL Chapter 8]

Example decision tree [source: ISL Chapter 8]

Interpretation

- **Years** is the most important factor in determining Salary. Players with less experience earn lower salaries than more experienced players.
- For less experienced players, the **#hits** in previous year is of little importance.
- More experience players get rewarded for a larger #hits.

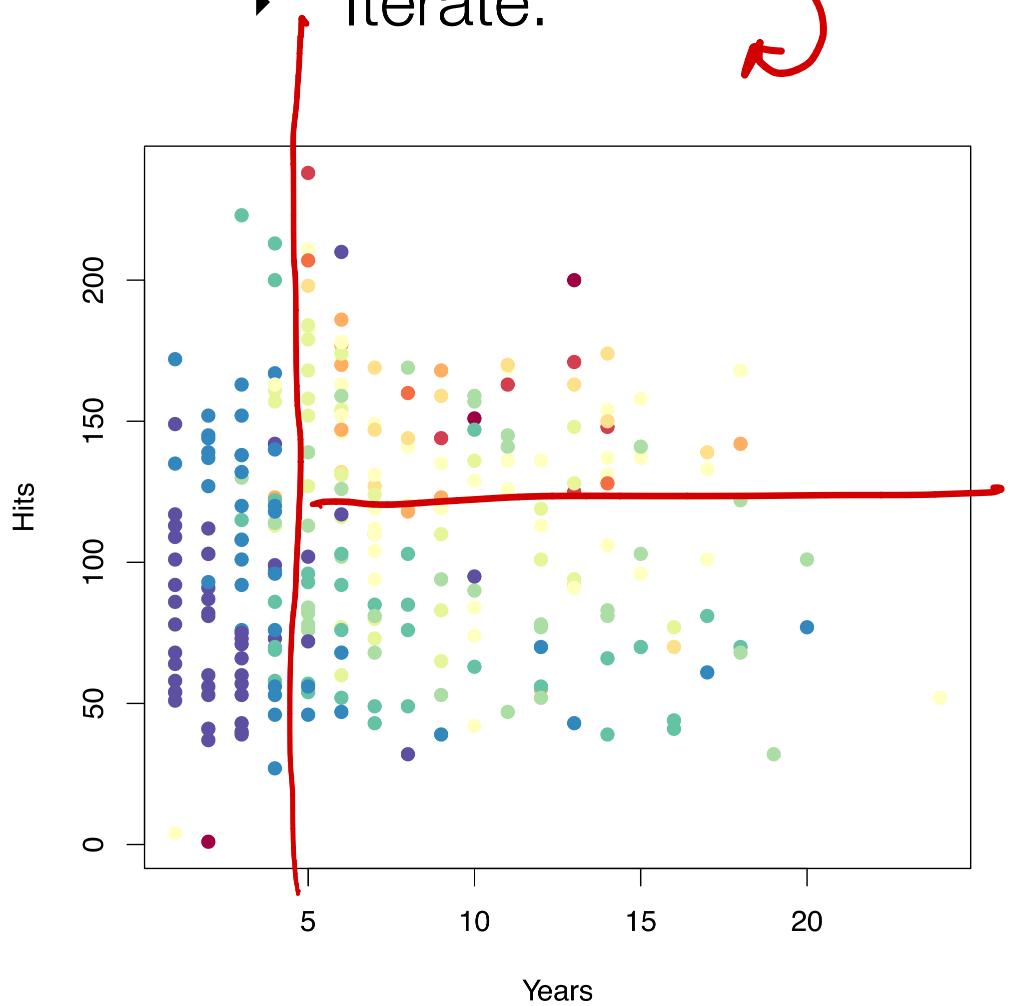


Tree building process

- Recursive binary splitting: minimize $\sum_{j=1}^{J} \sum_{i:\mathbf{x}_i \in R_j} (y_i - \hat{y}_{R_j})^2$

  with $\hat{y}_{R_j}$ mean response for training observations in j^{th} box

- Iterate:



1. Select the predictor/feature x_j and the cutpoint s, such that splitting $\{\mathbf{x} \,|\, x_j < s\}$ and $\{\mathbf{x} \,|\, x_j \geq s\}$ leads to largest decrease in SoSE. (greedy)

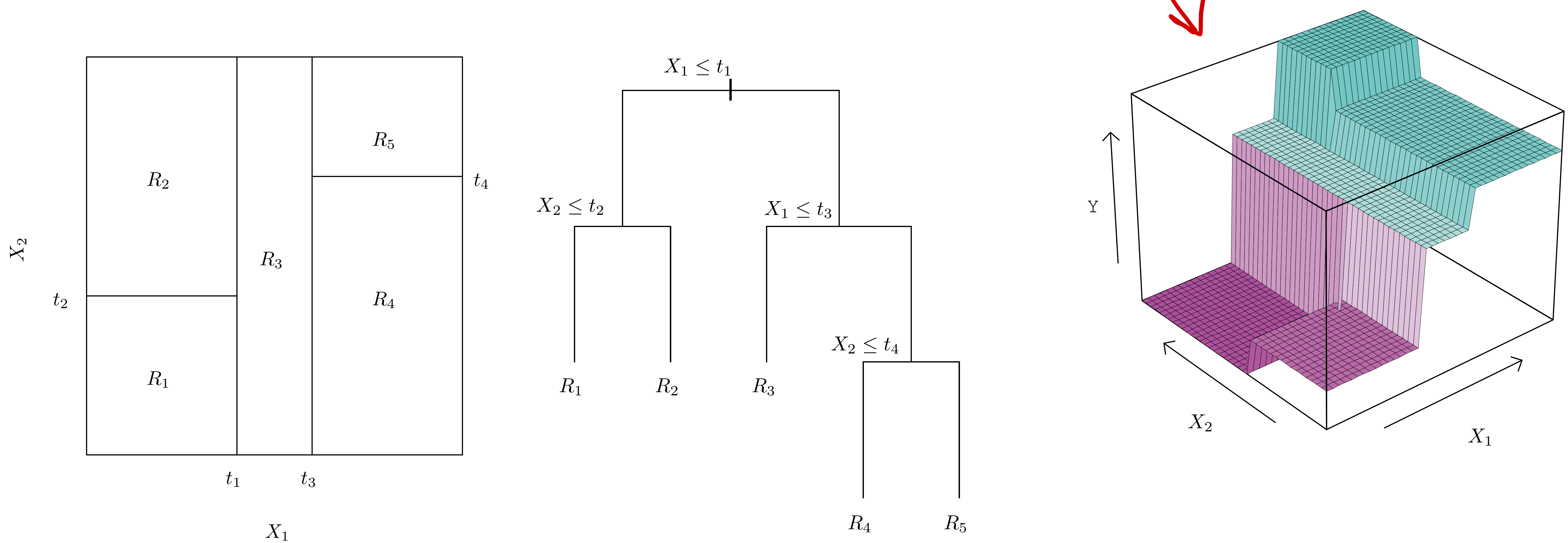
2. For each of the two regions: Select the best predictor/feature x_j and the cutpoint s that lead to largest decrease in SoSE. Split the region that has largest decrease in SoSE.

3. Example stopping criterion: Every region should contain at most 5 observations

Predictions

(Regression)

- For new datapoint: if $\mathbf{x}' \in R_j$ predict $t' = \frac{1}{|R_j|} \sum_{\mathbf{x}_i \in R_j} y_i$



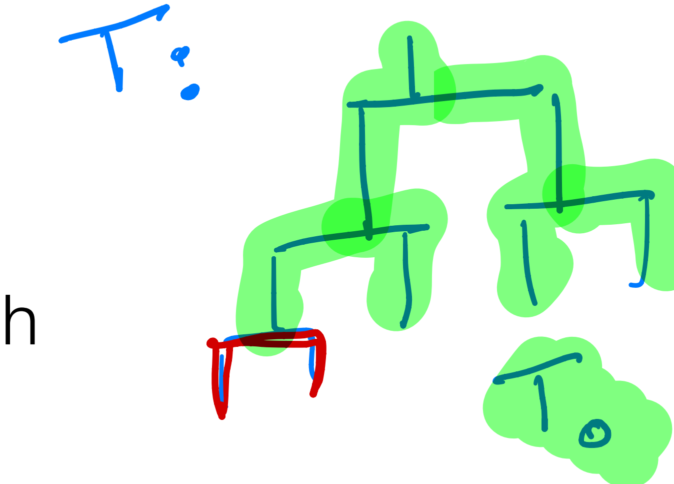
For classification: prediction is majority vote!

Example decision tree [source: ISL Chapter 8]

Decision trees: overfitting

- Large trees might overfit to the training set.
- A small variation in the training dataset can cause different splitting higher up the tree.
- Smaller trees can underfit.
- Strategy: stop splitting when the decrease in SoSE no longer exceeds a threshold
- Short-sided (greedy). A split with a small decrease can lead to larger decreases later on.

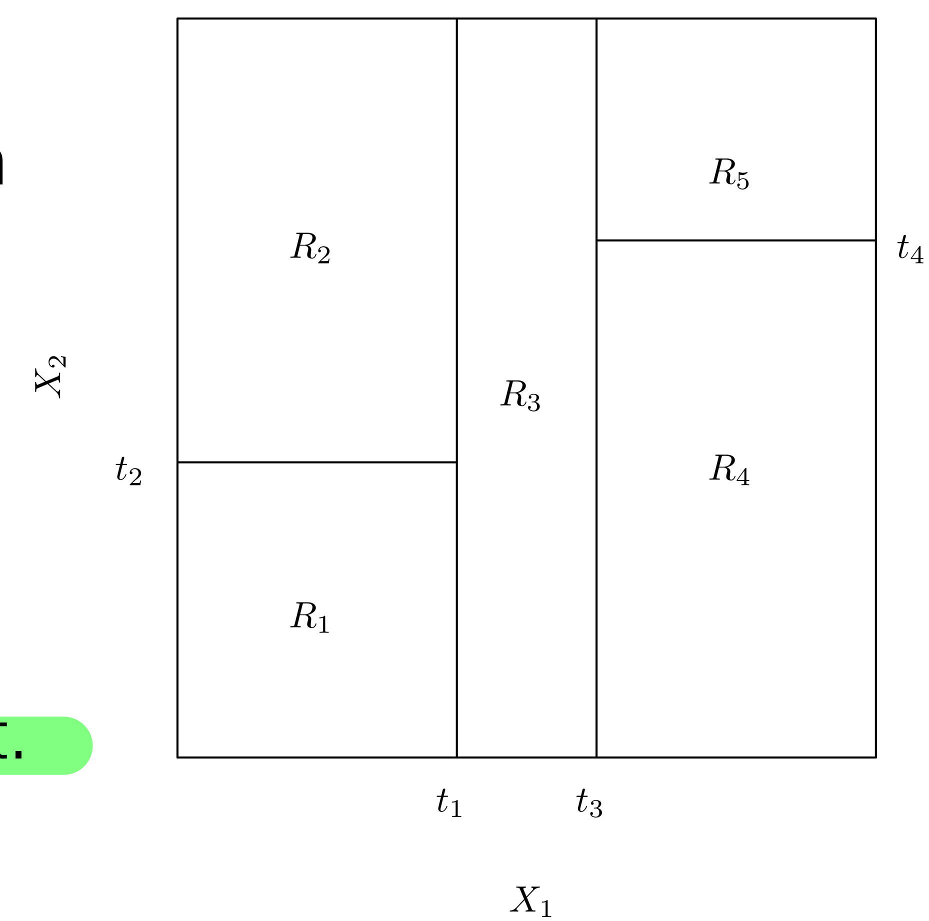
Pruning decision trees



- **Strategy 1**: Grow the tree only until a maximum depth
- **Strategy 2**: Grow a large tree T_0 and prune it to a subtree T with a smaller number of terminal nodes $|T|$.
- Residual error at j^{th} leaf node: $Q_j = \sum_{i:\mathbf{x}_i \in R_j} (y_i - \hat{y}_{R_j})^2$
- Increase α slowly starting from zero and for each value find T that minimizes:

$$\sum_{j=1}^{|T|} Q_j + \alpha |T|$$

- Select the optimal value of α with a validation set.
- Cost complexity pruning/weakest link pruning

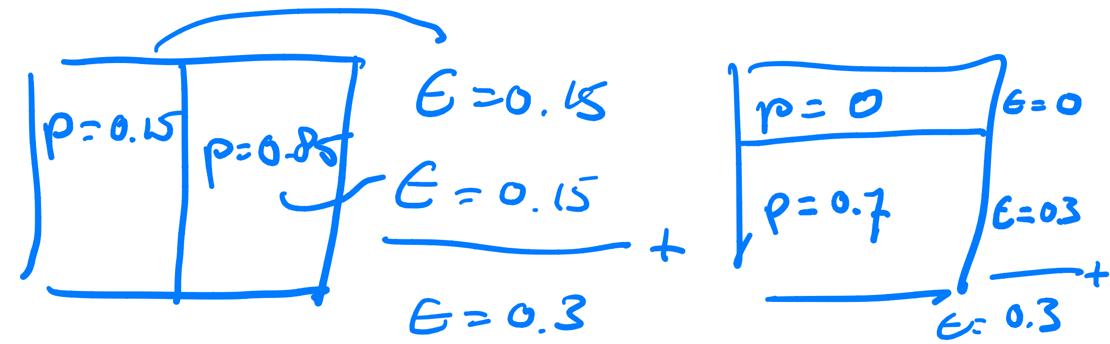


Classification decision trees

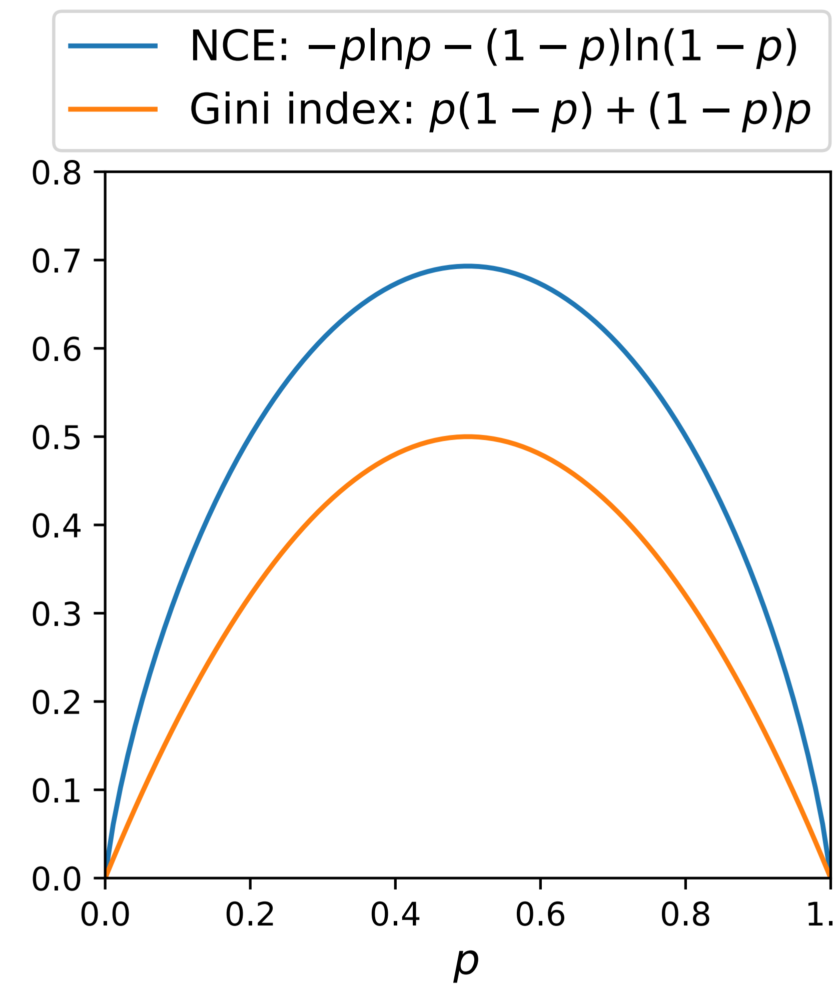
$p = \frac{|C_1|}{|C_2|}$

- Recursive binary splitting for classification with K classes

$$\min \sum_{j=1}^{J} Q_j$$



- The sum-of-squares error is replaced by one of the following options:
  - Misclassification rate: $Q_j = \frac{1}{N} \sum_{i:\mathbf{x}_i \in R_j} I[y(\mathbf{x}_i) \neq t_n]$
  - Negative cross entropy: $Q_j = -\sum_{k=1}^{K} p_{jk} \ln p_{jk}$
  - Gini index: $Q_j = \sum_{k=1}^{K} p_{jk}(1 - p_{jk})$
- NCE & Gini encourage regions with high proportions of data points for one of the classes



Ensemble methods

- Decision trees are easily interpretable and nice to visualize.
- Performance is usually suboptimal.
- Solution: Create ensembles of trees!
    - Bagging / bootstrap aggregation with trees
    - Random Forests: bagging + random subspace method
    - Boosting

Regression with GP's

- Combining models: (Bishop 4.1-4.4)
  - Bayesian model averaging vs. model combination methods
  - Committees:
    - Bootstrap aggregation
    - Random subspace methods
    - Boosting
  - Decision trees
  - **Random forests**

Random Forests

(Bootstrapping)

- Bagged trees can be highly correlated: if there are a few very strong predictors in the dataset, then all bagged trees will use these predictors in top splits

- **Solution**

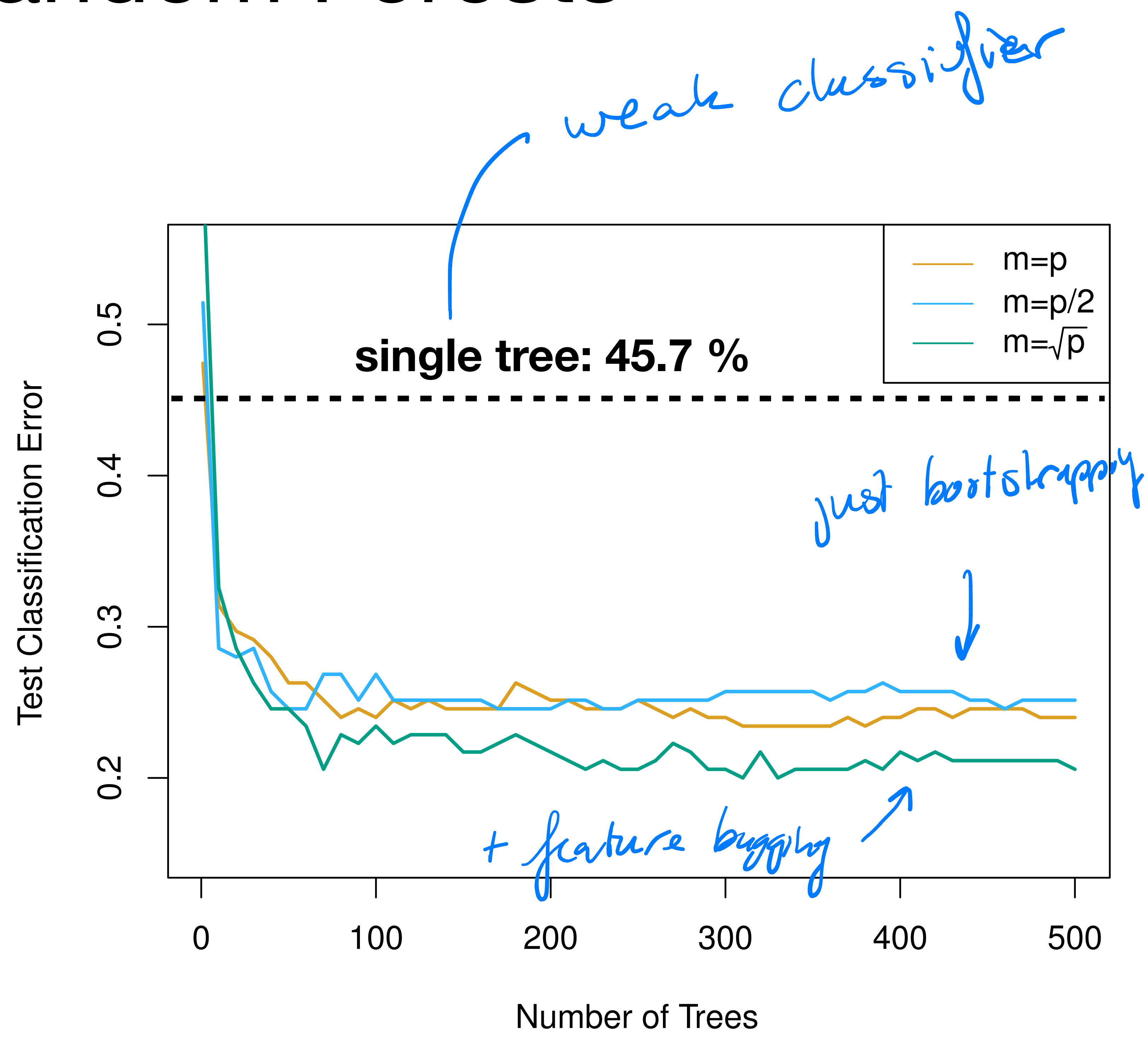
    - Build an ensemble of trees by bootstrapping the dataset

    - Feature bagging: for each tree, every time a split is considered, a random selection of m (out of p) predictors is chosen as a split candidate.

    - At each split a new selection is made, where typically $M = \sqrt{D}$

Bagging vs Random Forests

- Gene expression dataset
- Task: classify cancer type based on $p = 500$ gene expressions
- Random forests ($m < p$) show small improvement over just bootstrapping ($m = p$)



Bagging versus random forests for the gene expression dataset [source: ISL Chapter 8]