UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione

Lesson 8.

Tree-based methods

DATA SCIENCE AND AUTOMATION COURSE

MASTER DEGREE SMART TECHNOLOGY ENGINEERING

TEACHER
Mirko Mazzoleni

PLACE
University of Bergamo

Outline

1. Tree-based methods
2. Regression trees
3. Classification trees
4. Bagging and random forests

Outline

1. **Tree-based methods**

2. Regression trees

3. Classification trees

4. Bagging and random forests

Tree-based methods

- Tree-based methods involve **stratifying** or **segmenting** the predictor space into a number of simple regions [11]
- Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as **decision-tree methods**

Pros

- Simple
- Interpretable
- Can be applied to both regression and classification problems

Cons

- Not competitive with best supervised learning approaches

Bagging, random forest, boosting

Outline

1. Tree-based methods

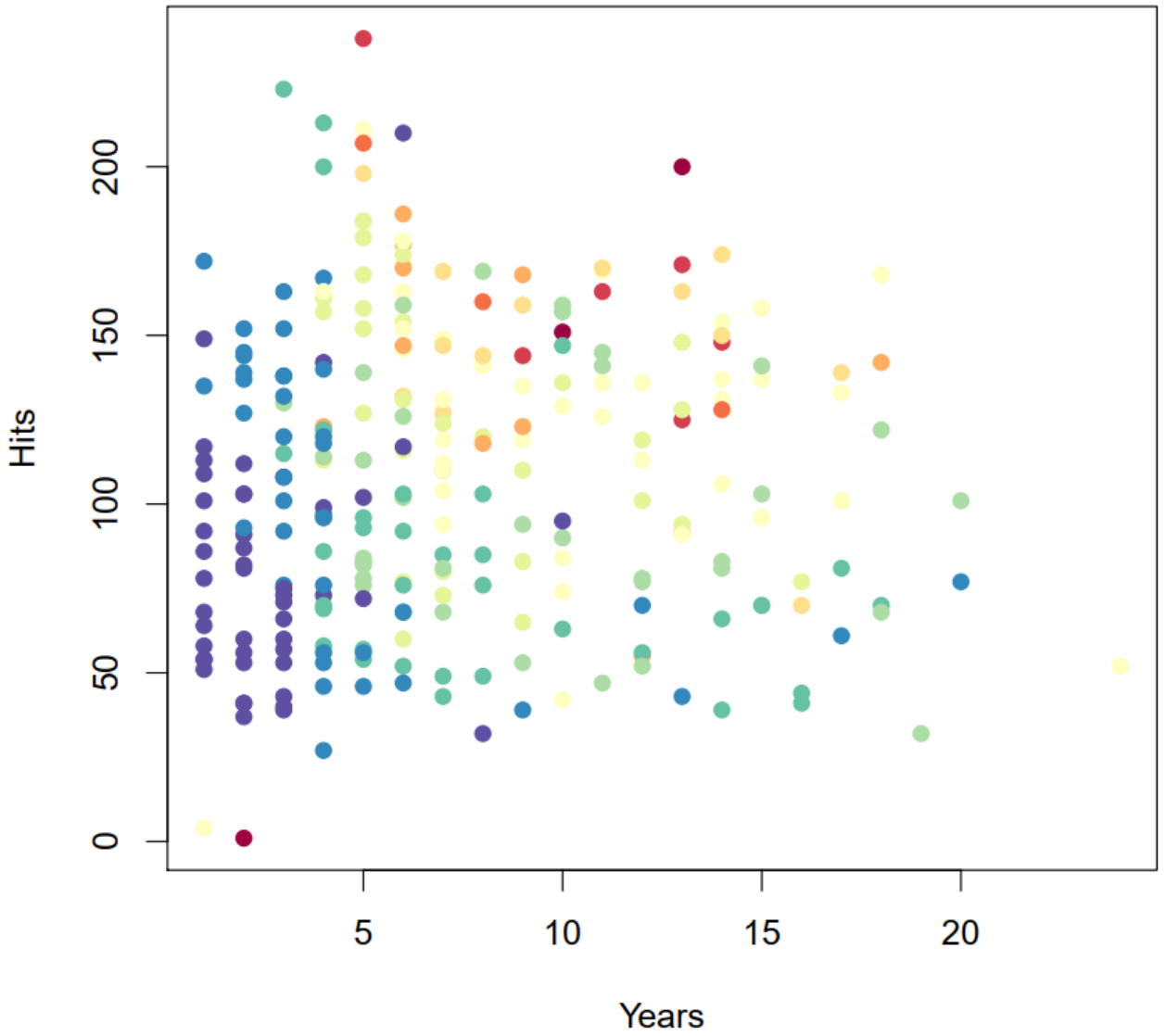
2. **Regression trees**

3. Classification trees

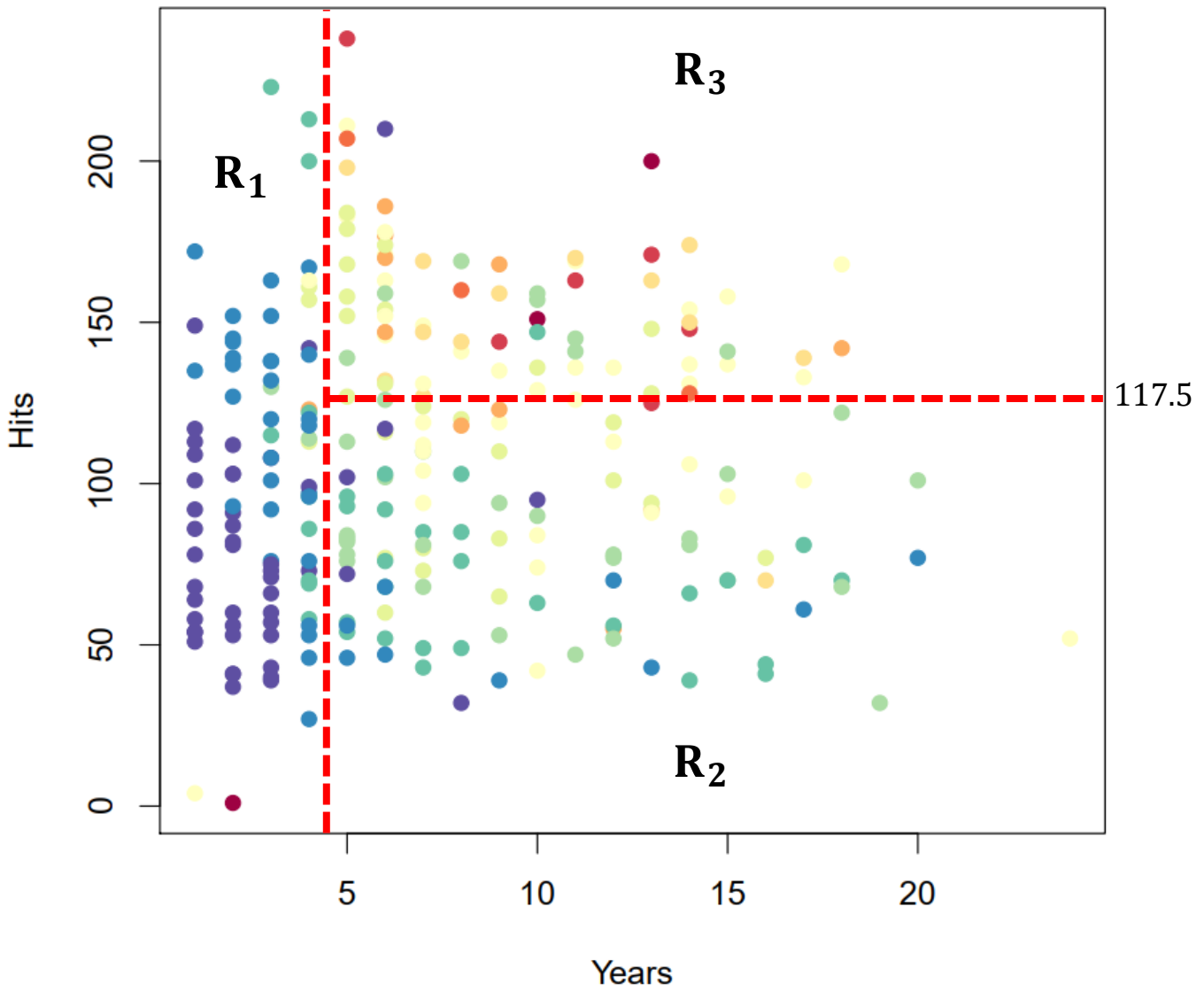
4. Bagging and random forests

Baseball salary data

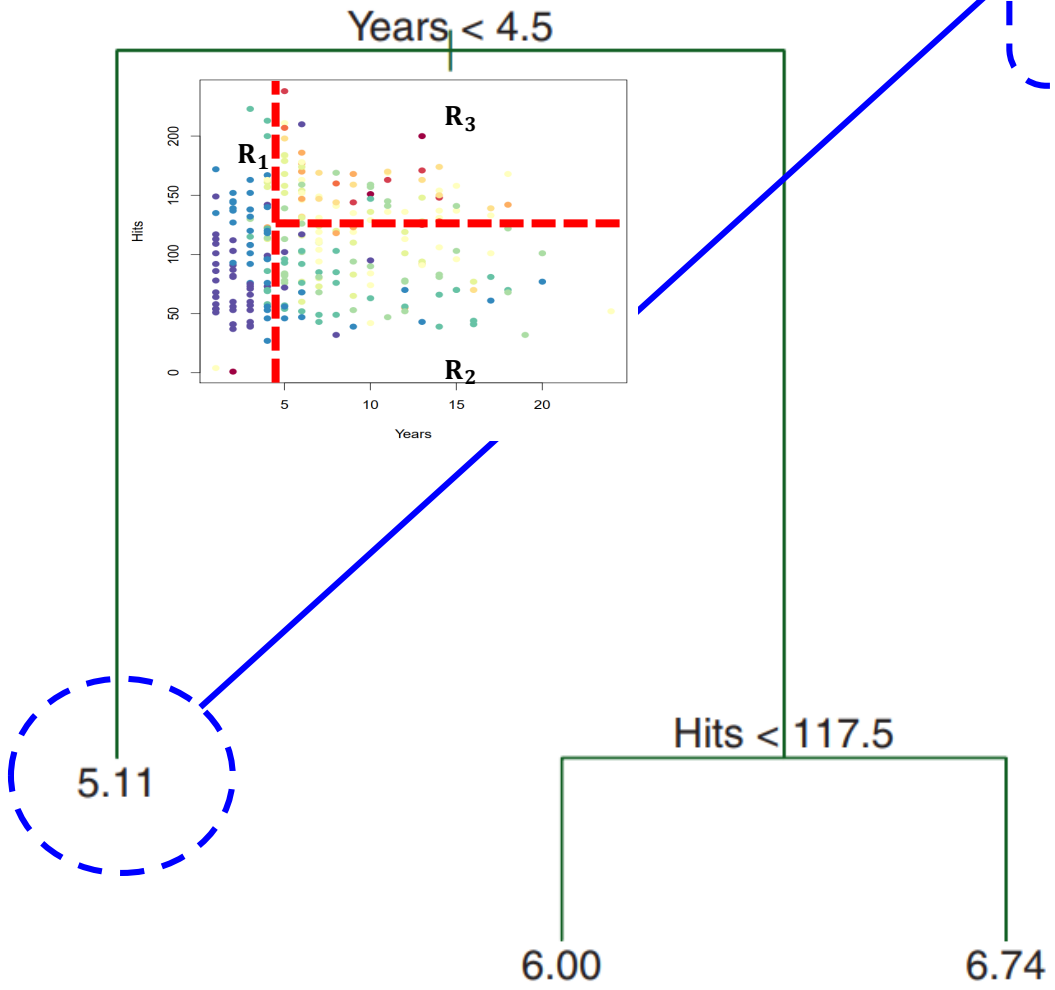
- Consider a **regression problem**
- Represent salary of baseball players given different variables *(https://rdrr.io/cran/ISLR/man/Hitters.html)*
- Salary is color-coded from low (**blue**, **green**) to high (**yellow**, **red**)
- How would you stratify it?



Baseball salary data



Baseball salary data



- The **predicted salary** or these players is given by the **mean response value** for the players in the data set with $\text{Years} < 4.5$
- Overall, the tree stratifies or segments the players into **three regions** of predictor space

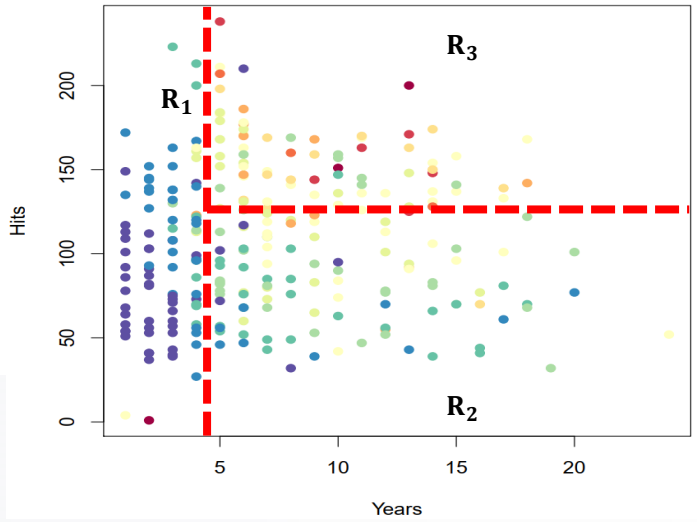
$$R_1 = \{X \mid \text{Years} < 4.5\} \quad R_2 = \{X \mid \text{Years} > 4.5, \text{Hits} < 117.5\}$$

$$R_3 = \{X \mid \text{Years} > 4.5, \text{Hits} > 117.5\}$$

- The regions R_1, R_2, R_3 are known as **terminal nodes** or **leaves** of the tree
- The points along the tree where the predictor space is split are referred to as **internal nodes**
- The segments of the trees that connect the nodes are the **branches**

Intepreprationof results

- Years is the **most important factor** in determining Salary, and players with less experience earn lower salaries than more experienced players
- Given that a player is **less experienced**, the number of Hits that he made in the previous year **seems to play little role** in his Salary
- Among players who have been in the major leagues for **five or more years**, the number of Hits made in the previous year **does affect** Salary, and players who made more Hits last year tend to have higher Salary
- Compared to a regression model, it **is easier to display, interpret and explain**



Tree-building process

1. We **divide the predictor space**, that is, the set of possible values for $x_1, x_2, \dots, x_d$ into M distinct and non-overlapping regions, $R_1, R_2, \dots, R_M$
2. For every observation that falls into the region R_M, we make the same prediction, which is simply the **mean of the response values** for the training observations in R_M

Step 1

We want to divide the predictor space in **boxes** $R_1, R_2, \dots, R_M$ that minimize the Residual Sum of Squares

$$RSS = \sum_{m=1}^{M} \sum_{i \in R_m} \left(y(i) - \hat{y}_{R_m}\right)^2$$

- $\hat{y}_{R_m}$: mean response for the training observations within the $m-\text{th}$ box

Tree-building process: Step 1

- A **top-down greedy** approach that is known as **recursive binary splitting** is employed to partition the predictor space
- **Begin at the top** of the tree and then successively split the predictor space
- At each step of the tree-building process, a **best split** is made

Select the predictor x_j and the cutpoint value s such that splitting the predictor space into the regions $\{X|x_j < s\}$ and $\{X|x_j \geq s\}$ leads to the **greatest possible** RSS reduction

Tree-building process: Step 1

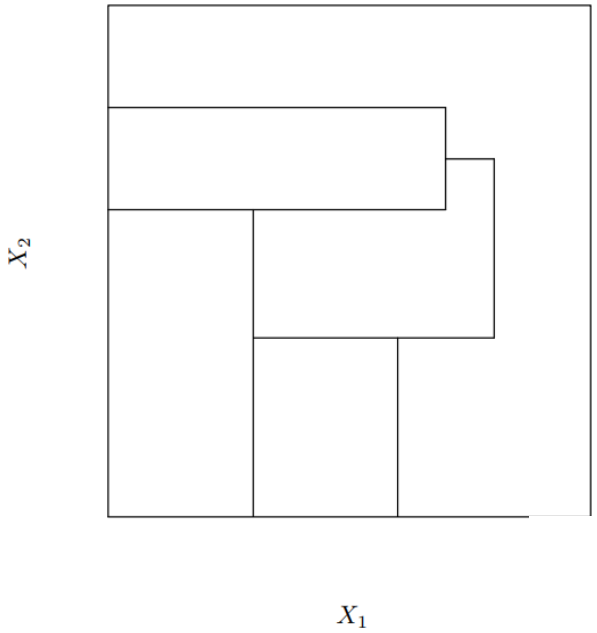
- Next, we repeat the process, looking for the best predictor and best cutpoint in order to **split the data further** so as to minimize the RSS within each of the resulting regions.

- However, this time, instead of splitting the entire predictor space, we **split one of the two previously identified regions**. We now have three regions

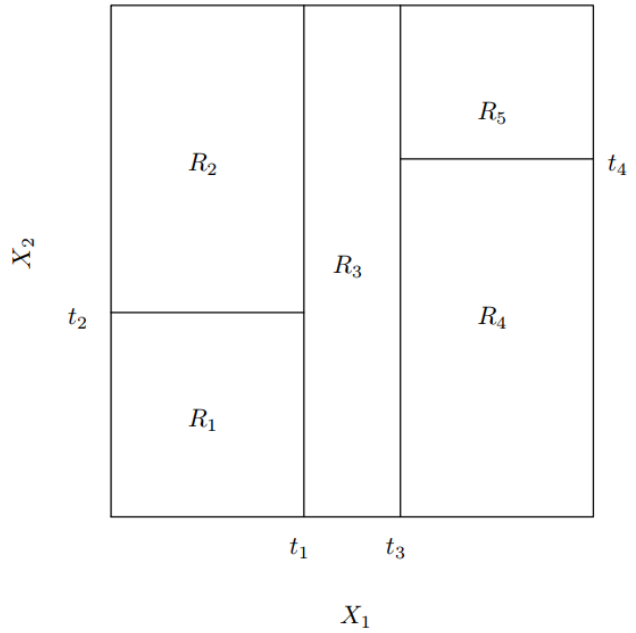
- Again, we look to split one of these three regions further, so as to minimize the RSS. The process continues **until a stopping criterion** is reached; for instance, we may continue until no region contains more than five observations

Example

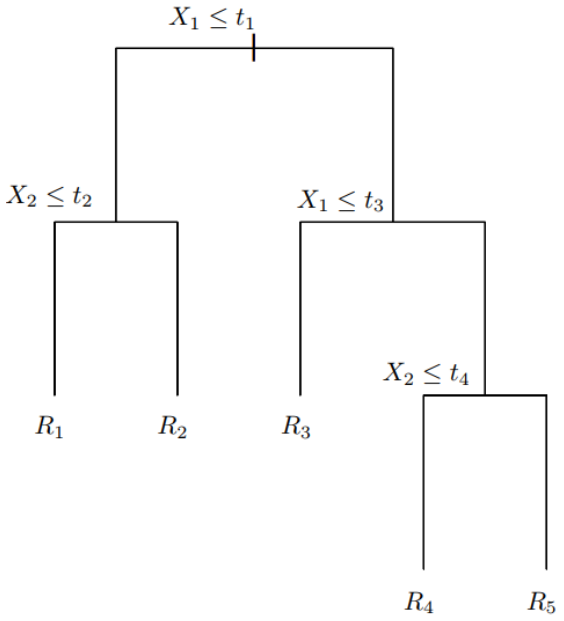
a) Partition not possible with binary recursive splitting

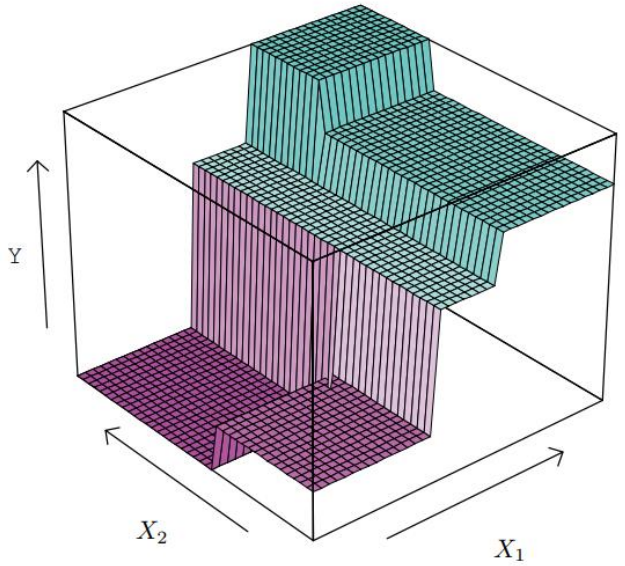


b) Output of binary recursive splitting



c) Tree corresponging to the partition b)





d) Perspective plot of the prediction surface corresponding to the tree c).

Outline

1. Tree-based methods

2. Regression trees

3. **Classification trees**

4. Bagging and random forests

Classification trees

- Very similar to a regression tree, except that it is used to predict a **qualitative** response rather than a **quantitative** one

- Each observation belongs to the **most commonly occurring class** of training observations in the region to which it belongs

- The recurvise binary splitting strategy is still employed. However, now **we can't use the RSS as metric** for node splitting

Gini index and cross-entropy

- Instead of RSS, we can minimize the following quantities

Gini index

$$G = \sum_{c=1}^{C} \hat{p}_{mc}(1 - \hat{p}_{mc})$$

- The Gini index takes on a **small value** if all of the $\hat{p}_{mc}$ are close to zero or one
- For this reason the Gini index is referred to as a measure of **node purity**: a small value indicates that a node contains predominantly observations from a single class

Cross-entropy

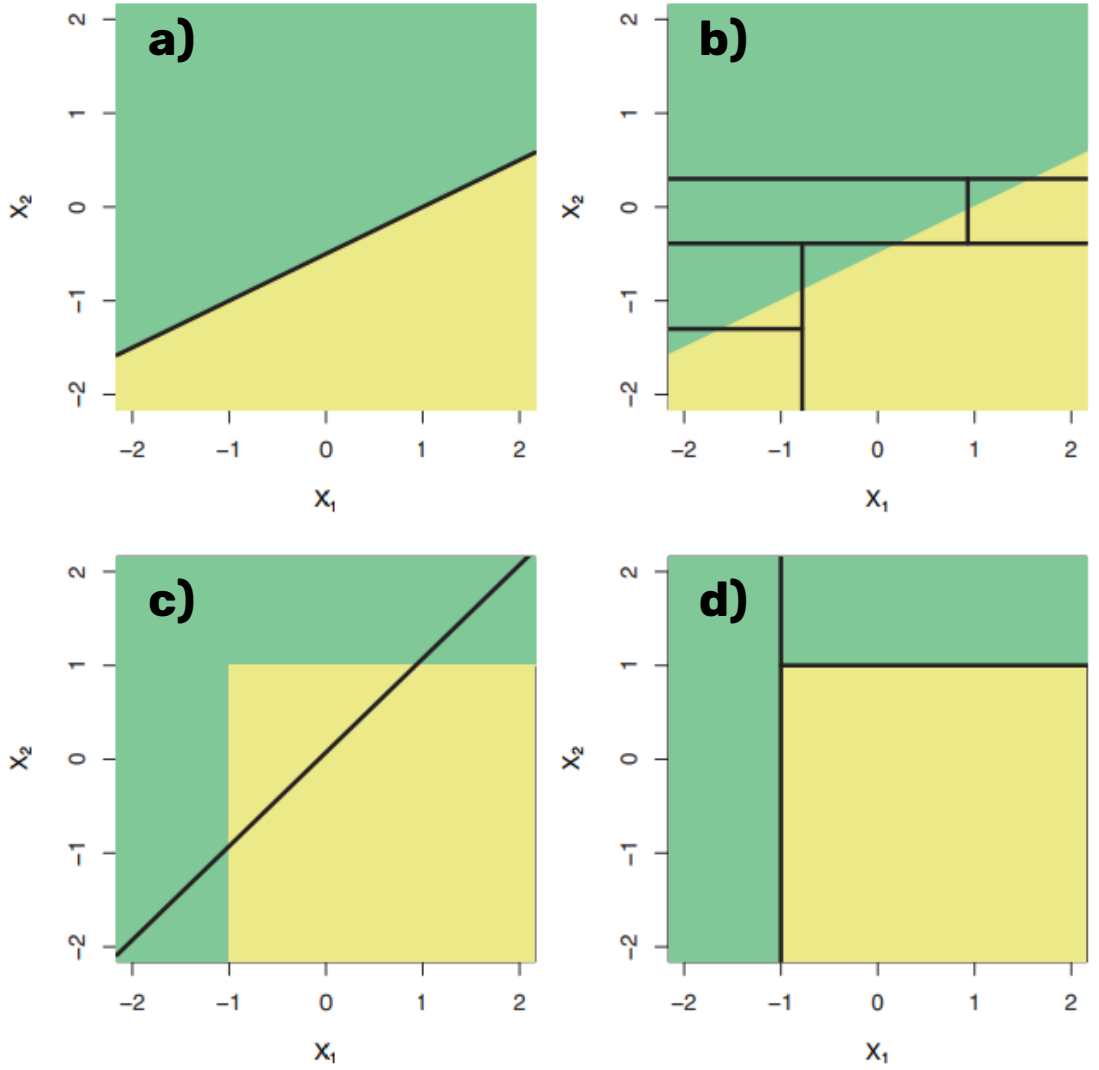
$$D = -\sum_{c=1}^{C} \hat{p}_{mc} \cdot \log[\hat{p}_{mc}]$$

- It turns out that the Gini index and the cross-entropy are **very similar numerically**
- $\hat{p}_{mc}$: proportion of training observations in the m-th region that are from the c-th class. Obviously, $0 \leq \hat{p}_{mc} \leq 1$

Trees vs. linear models

Two dimensional classification example

- **a) – b)** The true decision boundary is **linear**
  - ✓ **a)** Linear classification is perfect
  - ✓ **b)** Decision trees are less accurate
- **c) – d)** The true decision boundary is **not linear**
  - ✓ **a)** Linear classification is unable to find the correct boundary
  - ✓ **b)** decision trees are successful



Pictures taken from [11]

Decision trees summary

- Trees are **very easy to explain** to people. They can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small)
- Some people believe that decision trees **more closely mirror human decision-making** than do the regression and classification approaches seen previously
- Trees can easily **handle qualitative predictors** without the need to create dummy variables
- Unfortunately, trees generally **do not have the same level of predictive accuracy** as some of the other regression and classification approaches

Outline

1. Tree-based methods
2. Regression trees
3. Classification trees
4. **Bagging and random forests**

Bagging

- Aggregating different trees can **substantially improve** the predictive accuracy
- Bootstrap aggregation, or **bagging**, is a general-purpose procedure for reducing the variance of a statistical learning method.
  - ✓ It is particularly useful and frequently used in the context of decision trees
- Recall that given a set of N **independent** observations $z(1), \dots, z(N)$, each with variance σ^2, the **variance of the mean** $\bar{z}$ of the observations is given by σ^2/N
- In other words, **averaging** a set of observations **reduces variance**. Of course, this is not practical because we generally do not have access to multiple training sets.

Bagging

- Instead, we can **bootstrap**, by taking repeated samples from the (single) training data set → extraction with resampling of the training data
- In this **bagging** approach we generate B different bootstrapped training data sets
  - ✓ Train our method on the b-th bootstrapped training set in order to get $\hat{f}^{*b}(\boldsymbol{x})$, the prediction at a point $\boldsymbol{x}$.
  - ✓ **Regression trees**
    - **Average** all the predictions to obtain $\hat{f}_{bag}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(\boldsymbol{x})$
  - ✓ **Classification trees**
    - **Majority vote:** the overall prediction is the most commonly occurring class among the B predictions.

B is not critical and can be set to a sufficiently high number, such as $B = 100$

Out-of-Bag error estimation

- Estimate the **validation error** of a bagged model
- In bagging, trees are fit to bootstrapped **subsets** of the observations
  - ✓ on average, each bagged tree makes use of around of the $2/3$ of observations
- The remaining $1/3$ of the observations **not used to fit** a given bagged tree are referred to as the **out-of-bag (OOB)** observations
- We can **predict the response** for the i-th observation using each of the trees in which that observation was OOB. This will yield around $B/3$ predictions for the i-th observation, which we **average**
- This estimate is essentially the Leave-One-Out cross-validation error for bagging, if B is large

Random forests

- Random forests provide an improvement over bagged trees by **decorrelating** the trees
  - ✓ This reduces the variance when we average the trees

- When building these decision trees, **each time a split** in a tree is considered, a **random** selection of $q < d$ predictors is chosen as split candidates from the full set of d predictors
  - ✓ The split **is allowed to use only one** of those q predictors

- A different selection of q predictors is taken at each split, and typically we choose

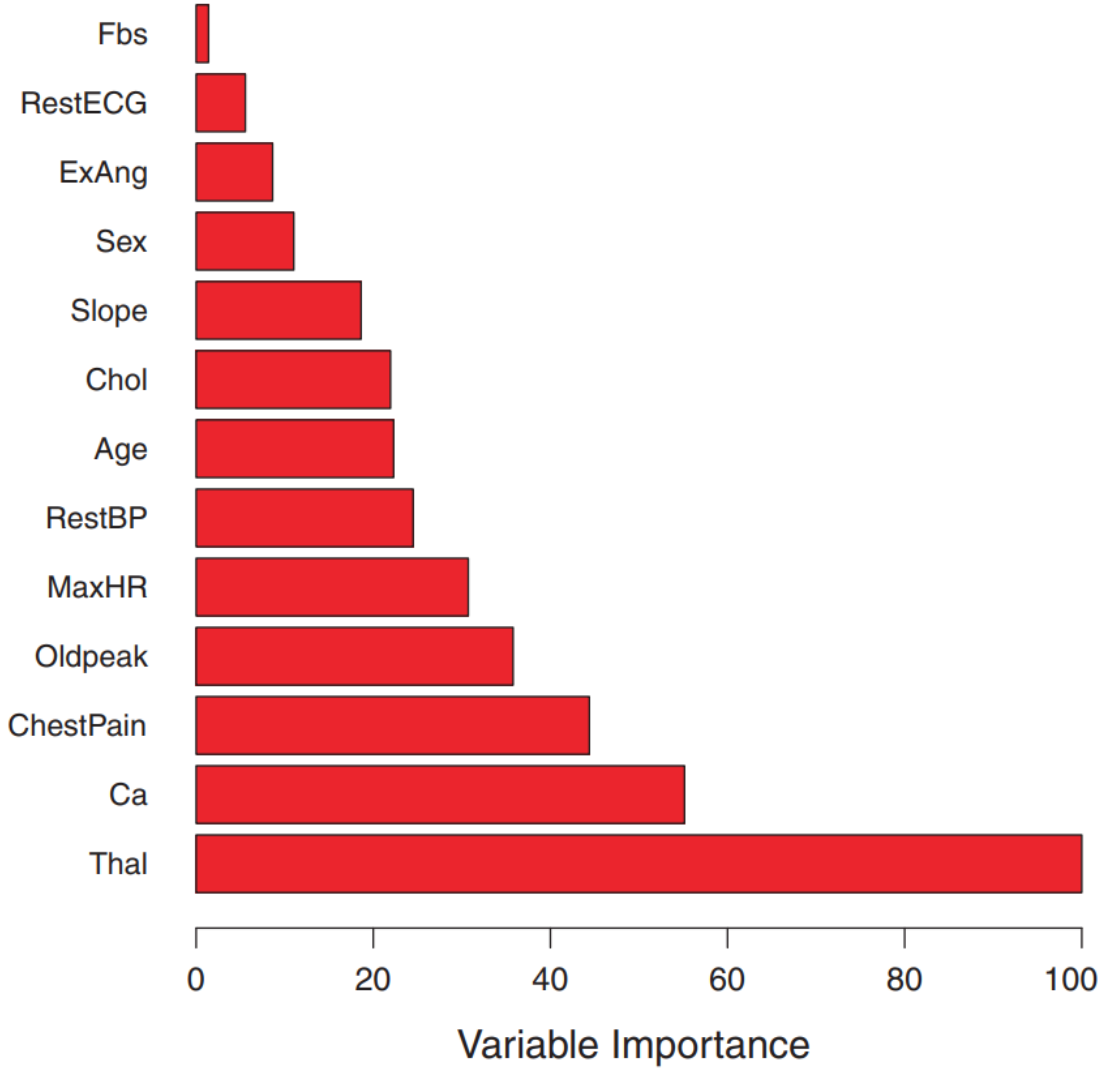
$$q = \sqrt{d}$$

Variable importance measure

- Tree methods can automatically assess the **importance** of each feature in predicting the output

- For bagged/RF regression trees, we record the **total amount** that the **RSS is decreased** due to splits over a **given predictor**, averaged over all B trees. A **large value** indicates an **important** predictor

- Similarly, for bagged/RF classification trees, we add up the total amount that the Gini index is decreased by splits over a given predictor, averaged over all *B* trees



References

1. Provost, Foster, and Tom Fawcett. "*Data Science for Business: What you need to know about data mining and data-analytic thinking*". O'Reilly Media, Inc., 2013.
2. Brynjolfsson, E., Hitt, L. M., and Kim, H. H. *"Strength in numbers: How does data driven decision making affect firm performance?"* Tech. rep., available at SSRN: http://ssrn.com/abstract=1819486, 2011.
3. Pyle, D. "*Data Preparation for Data Mining*". Morgan Kaufmann, 1999.
4. Kohavi, R., and Longbotham, R. *"Online experiments: Lessons learned".* Computer, 40 (9), 103–105, 2007.
5. Abu-Mostafa, Yaser S., Malik Magdon-Ismail, and Hsuan-Tien Lin. *"Learning from data"*. AMLBook, 2012.
6. Andrew Ng. *"Machine learning"*. Coursera MOOC. (https://www.coursera.org/learn/machine-learning)
7. Domingos, Pedro. "*The Master Algorithm".* Penguin Books, 2016.
8. Christopher M. Bishop, "*Pattern recognition and machine learning", Springer-Verlag New York, 2006.*
9. Hastie, T., Tibshirani, R.,, Friedman, J. *"The Elements of Statistical Learning"*. New York, NY, USA: Springer New York Inc, 2001.
10. Tom Fawcett, *"An introduction to ROC analysis",* Pattern Recognition Letters, Volume 27, Issue 8, 2006, Pages 861-874, ISSN 0167-8655,
11. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *"An Introduction to Statistical Learning: With Applications in R".* Springer Publishing Company, Incorporated, 2014.