https://github.com/tedinburgh/ads2023

Decision trees

Tom Edinburgh
te269

Schedule

https://github.com/tedinburgh/ads2023

1. Supervised learning
    - Regression, model fitting
    - Classification
    - k-nearest neighbours, SVM
    - Tree-based methods
2. Unsupervised learning
    - Dimensionality reduction
    - Clustering
3. Intro to neural networks
    - Neurons, perceptrons and back-propagation

Schedule

1. Supervised learning ← Irina
   - Regression, model fitting
   - Classification
   - k-nearest neighbours, SVM
   - Tree-based methods ← Tom
2. Unsupervised learning
   - Dimensionality reduction
   - Clustering
3. Intro to neural networks ← Miles
   - Neurons, perceptrons and back-propagation

Today: Decision trees

- Overview
- How to grow a tree (recursive binary partition)
- Decision trees for regression (example: baseball salary)
- Decision trees for classification (example: playing outdoors)
- Summary

Questions: halfway through, at the end, or by email (te269)

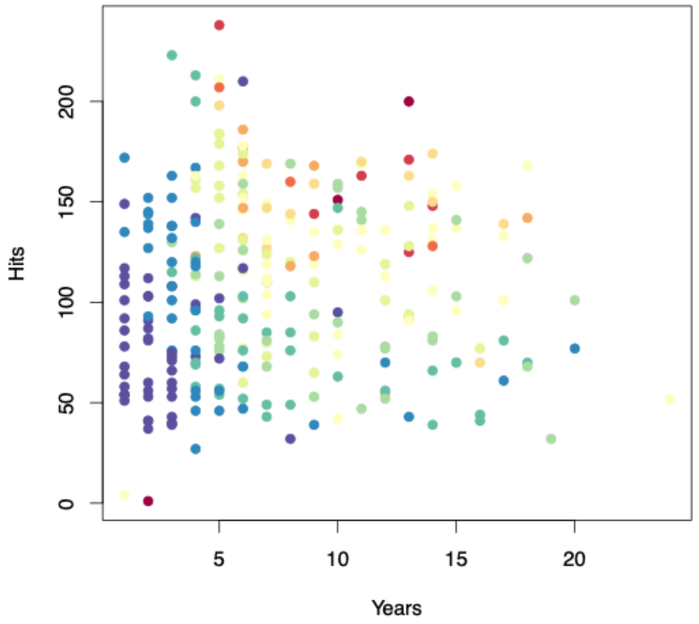
Resources

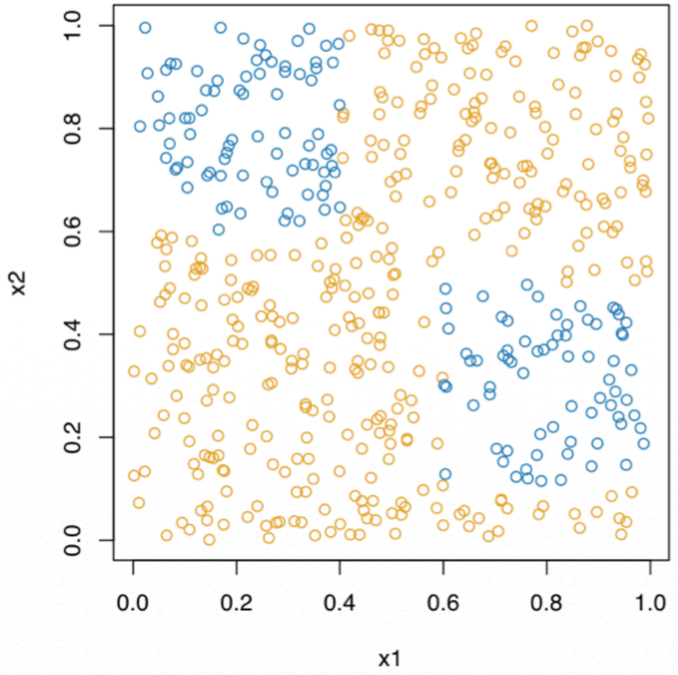
- An Introduction to Statistical Learning with Applications in R/Python (James, Witten, Hastie, Tibshirani, Taylor; 2013/2023)

- Slides adapted from:
  - Prof Alexandra Chouldechova, Carnegie Mellon
  - Prof Stephen Eglen, Cambridge

Overview

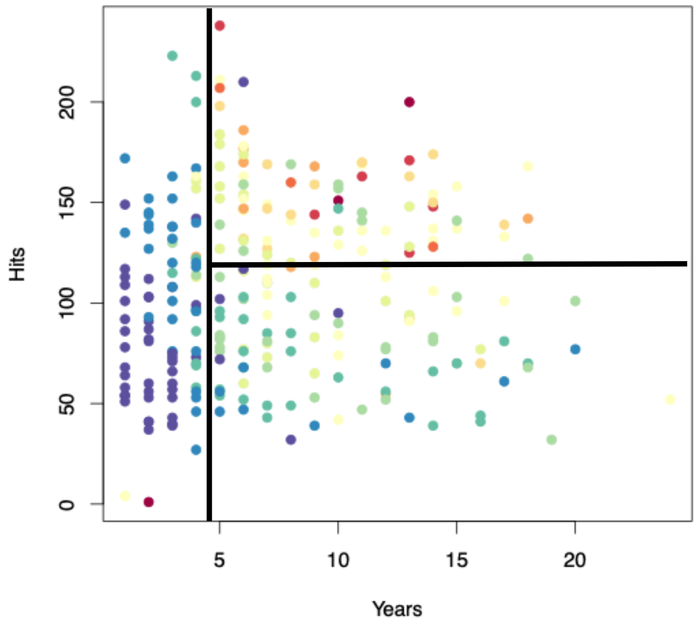
- Tree-based methods **segment** the feature space into simple regions (e.g. high-dimensional rectangles)
- **Predict** using the average (mean, mode), **classify** using the most common class
- Origins: CART (Breiman, 1984), identification trees (ID3) (Quinlan, 1986)

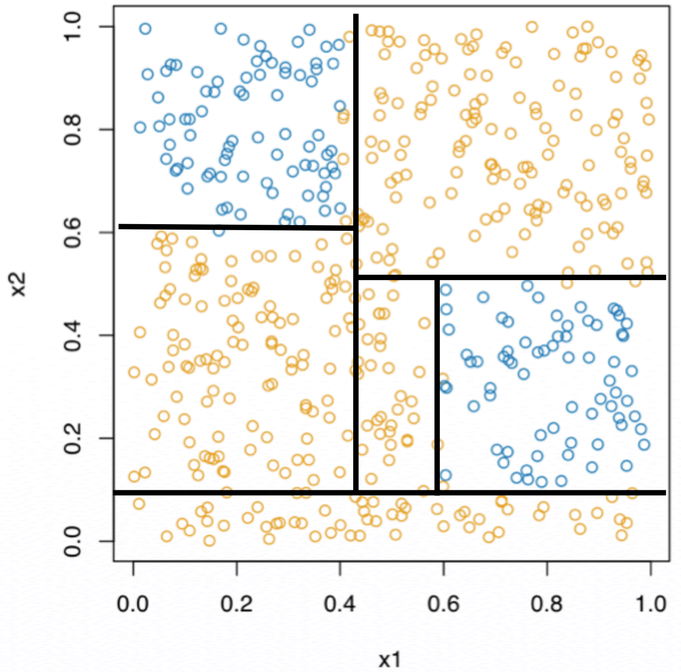




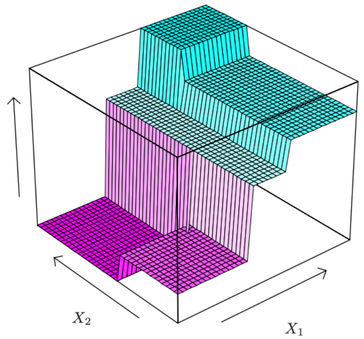
Overview

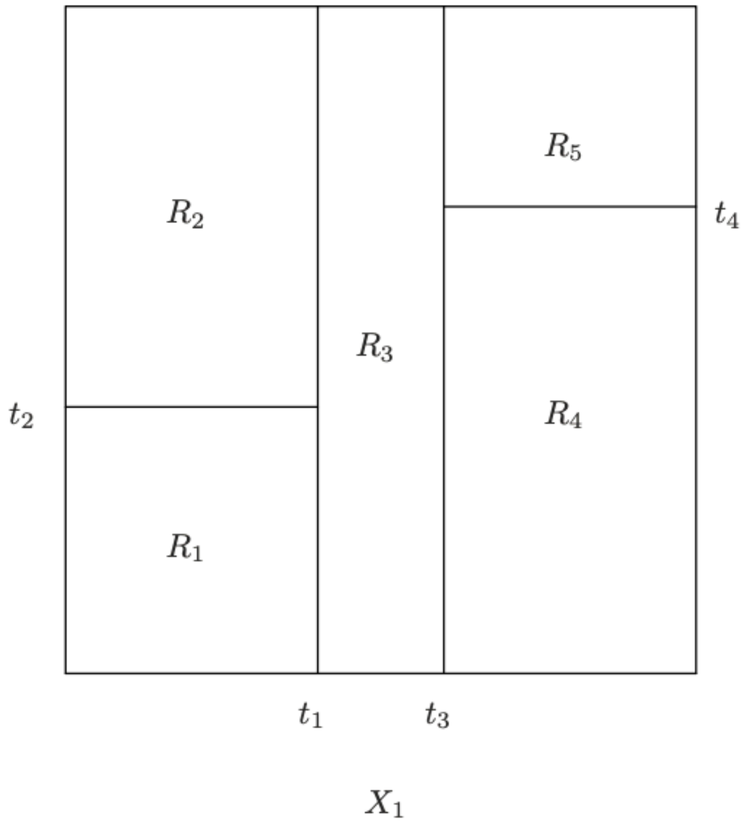
- Tree-based methods **segment** the feature space into simple regions (e.g. high-dimensional rectangles)
- **Predict** using the average (mean, mode), **classify** using the most common class
- Origins: CART (Breiman, 1984), identification trees (ID3) (Quinlan, 1986)

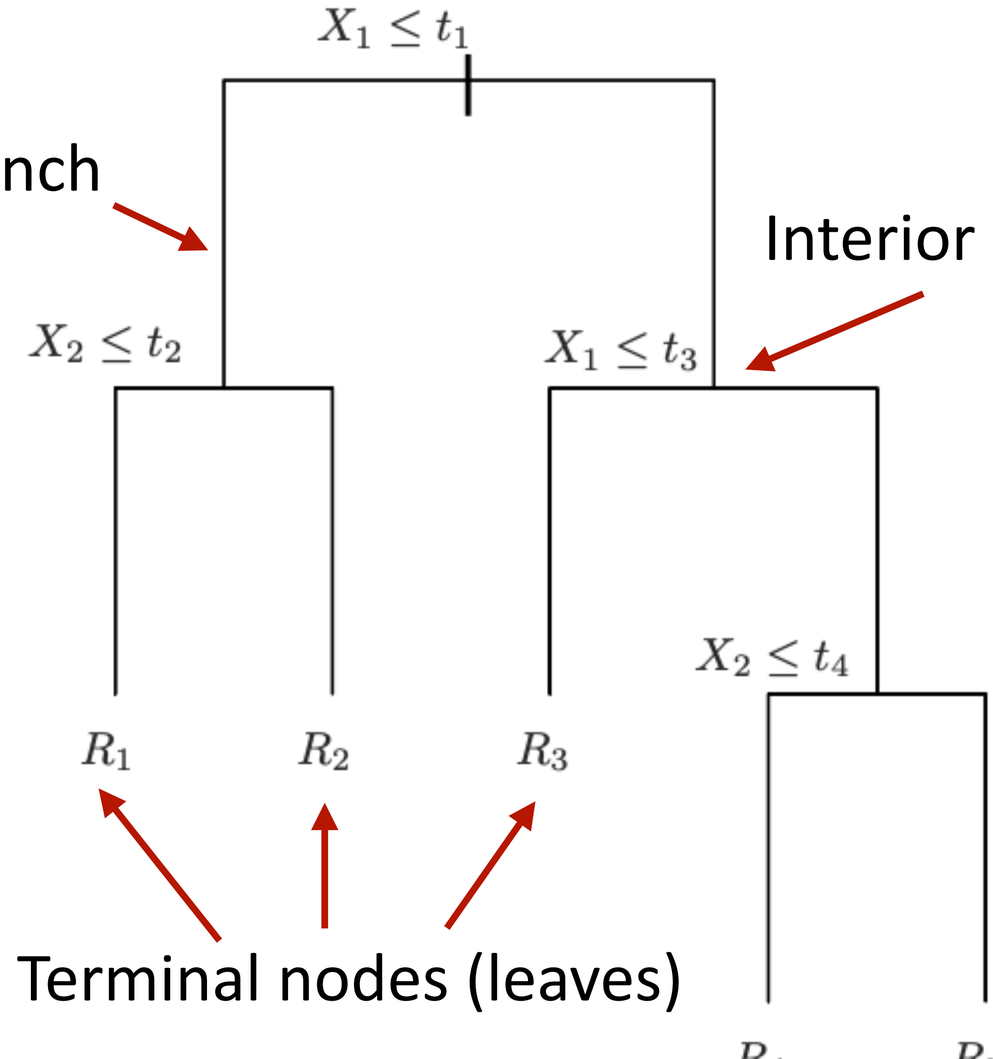


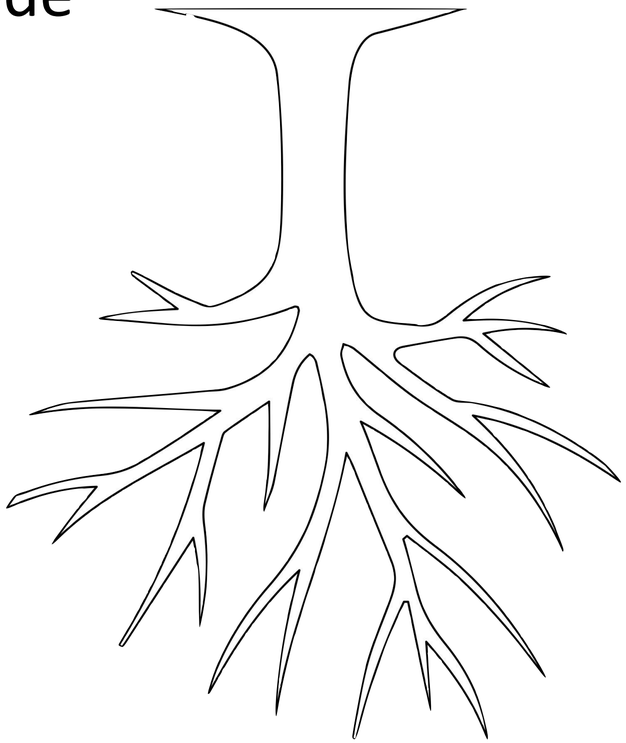


Why is this called a decision tree?









Format of the data

$$X = \begin{pmatrix} X_1 & X_2 & \cdots & X_p \end{pmatrix} = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \qquad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

- Each of the features (predictors) X_j can be qualitative or quantitative, no need to re-scale
- The output (response) y can be qualitative (for classification) or quantitative (for regression)

How to grow the tree?

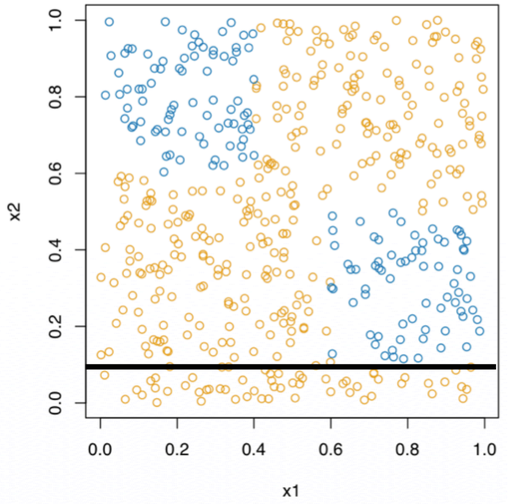
- General process:

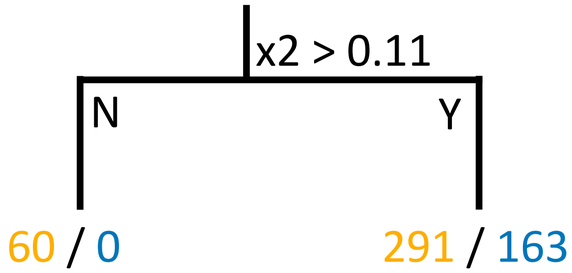
    1. Partition the predictor space (i.e. all possible values of $X_1, X_2, \ldots, X_p$) into J distinct, non-overlapping regions, labelled $R_1, \ldots, R_J$

    2. Each observation in a given region R_j is given the same predicted value (or class), which is the mean (or mode) of all response variables in that region

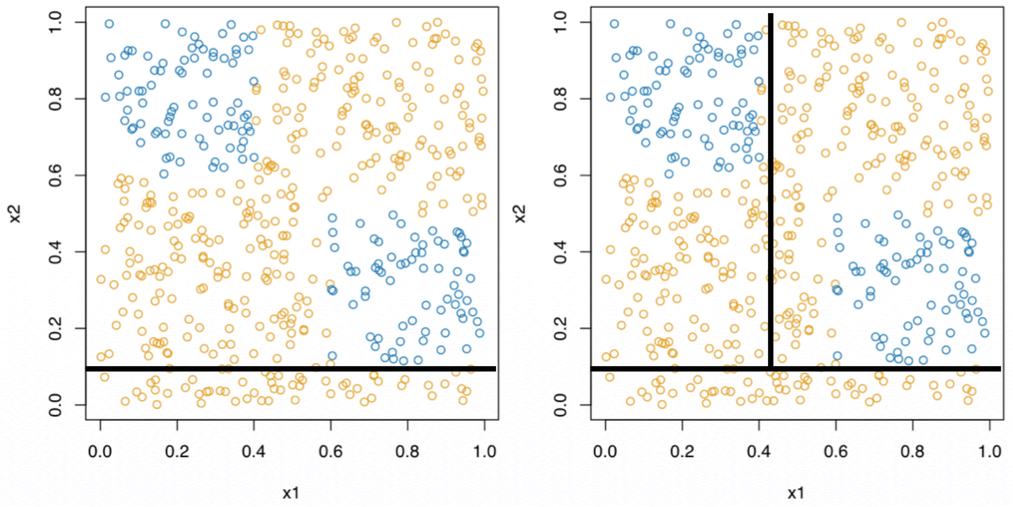
- How do we choose the partitions? e.g. high-dimensional rectangles

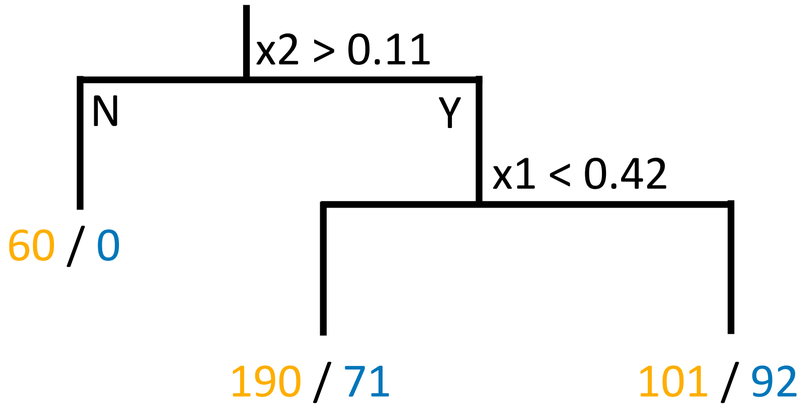
How to grow the tree?



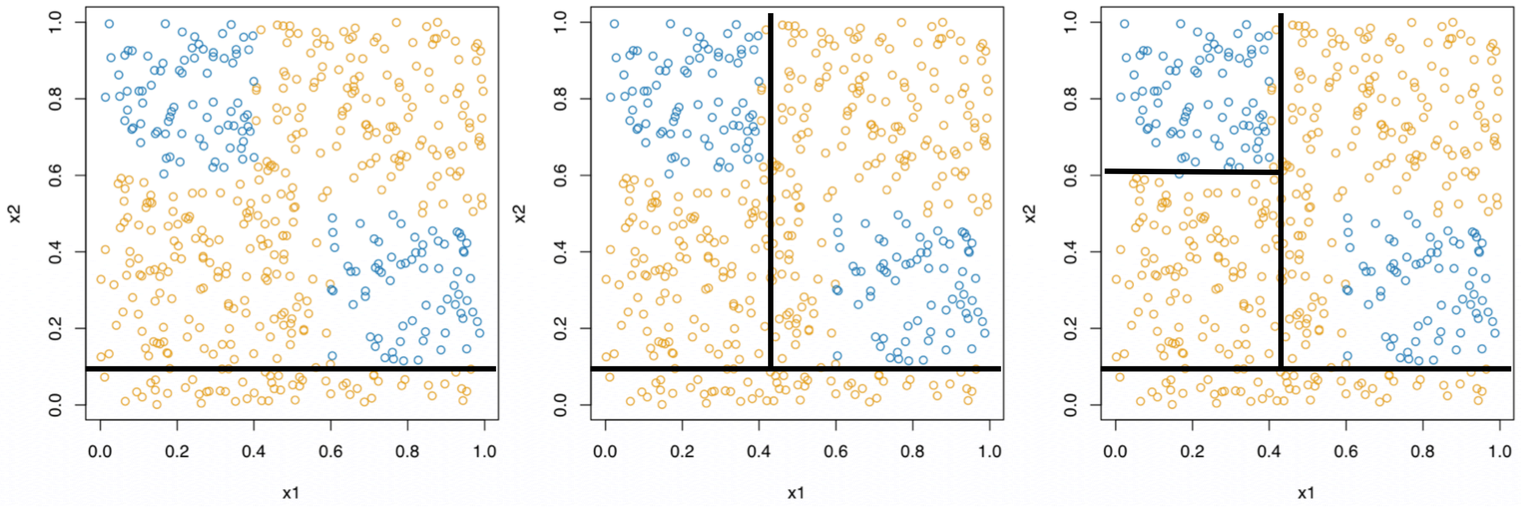


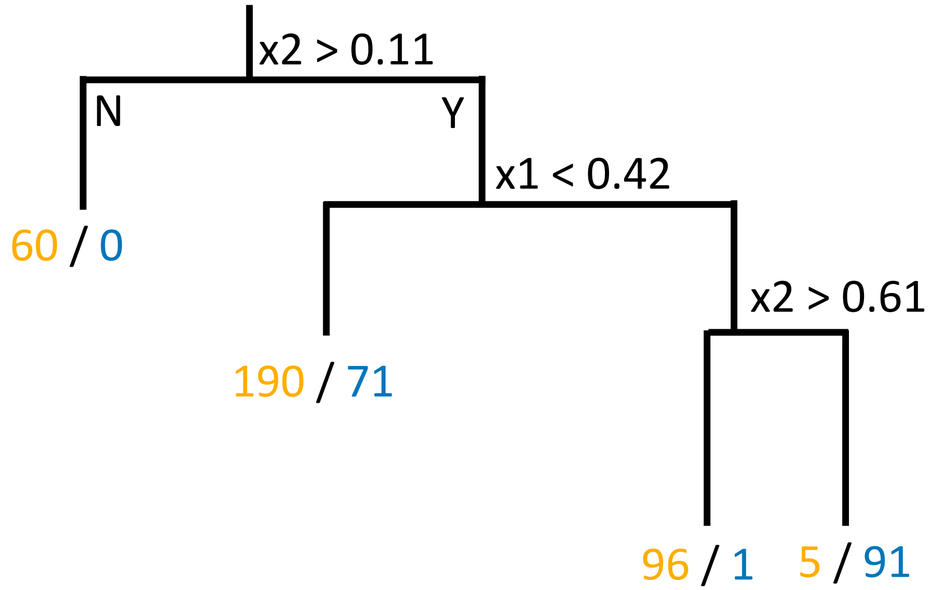
How to grow the tree?



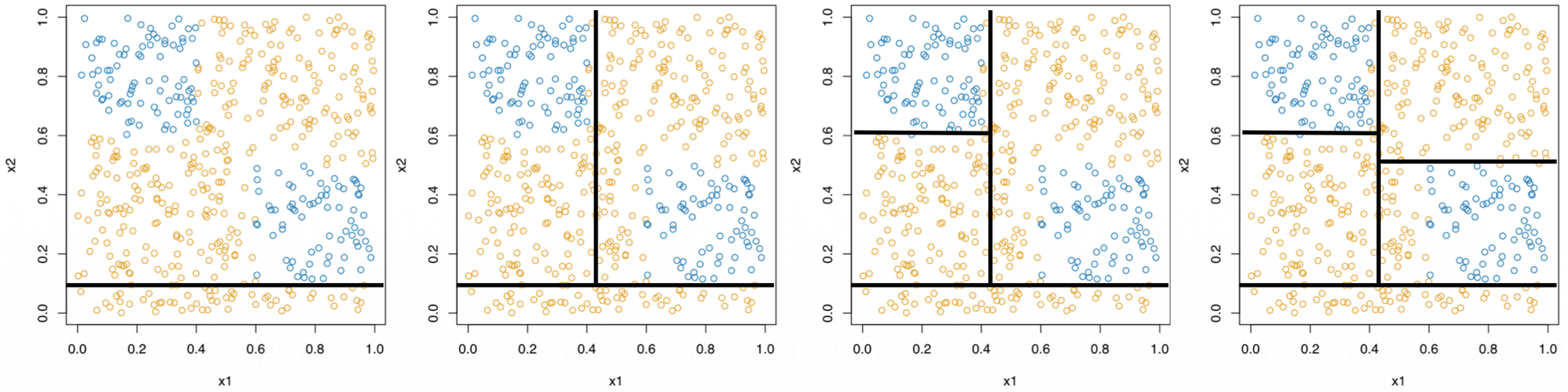


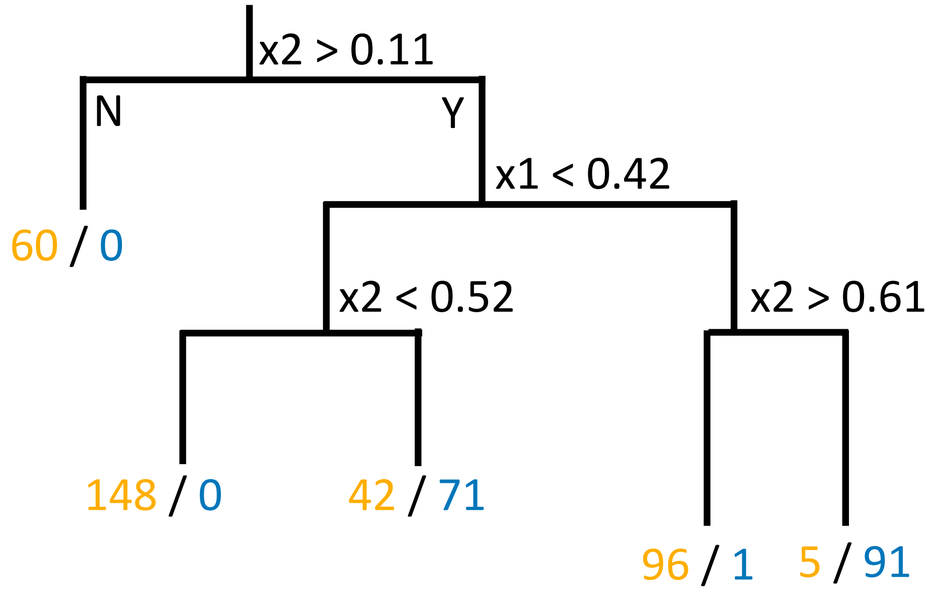
How to grow the tree?



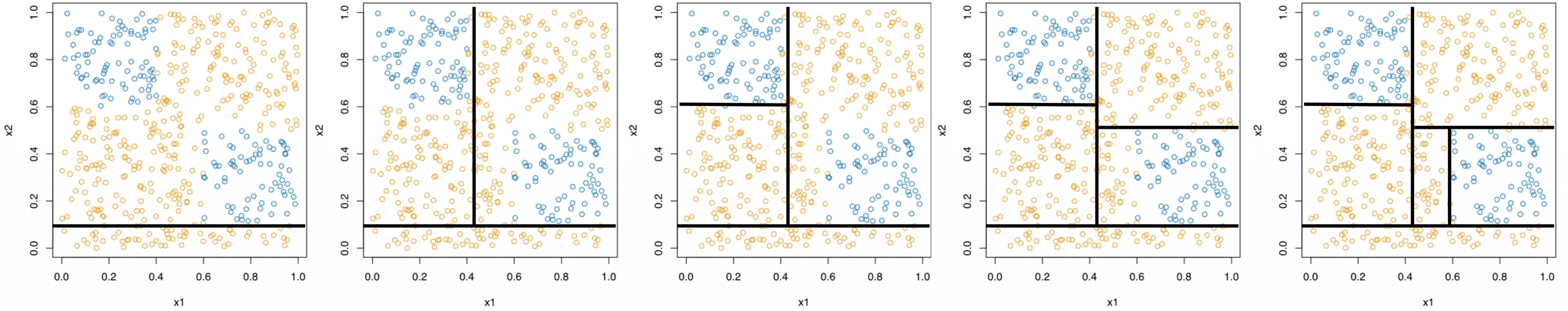


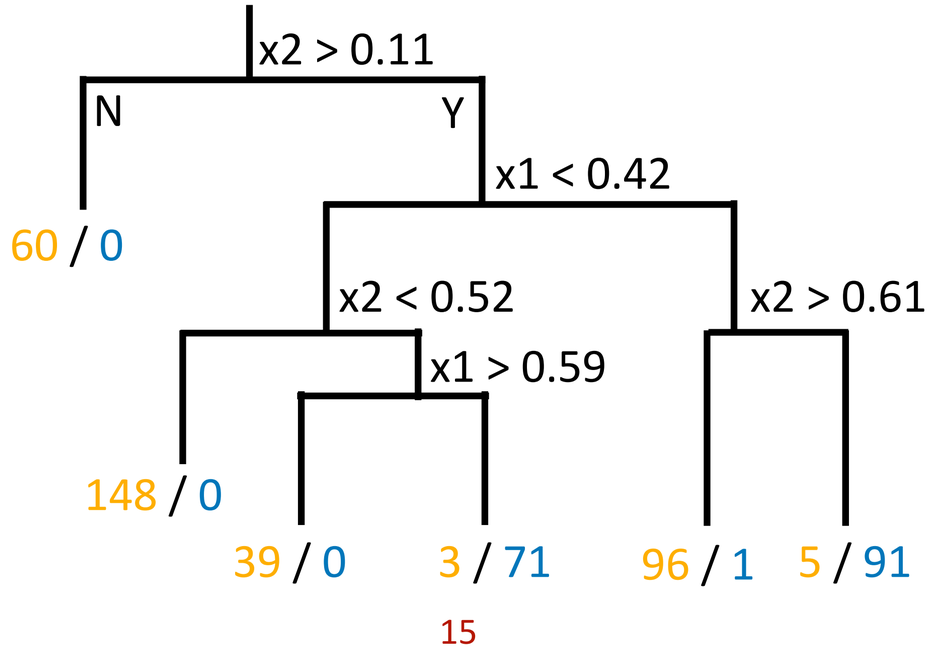
How to grow the tree?



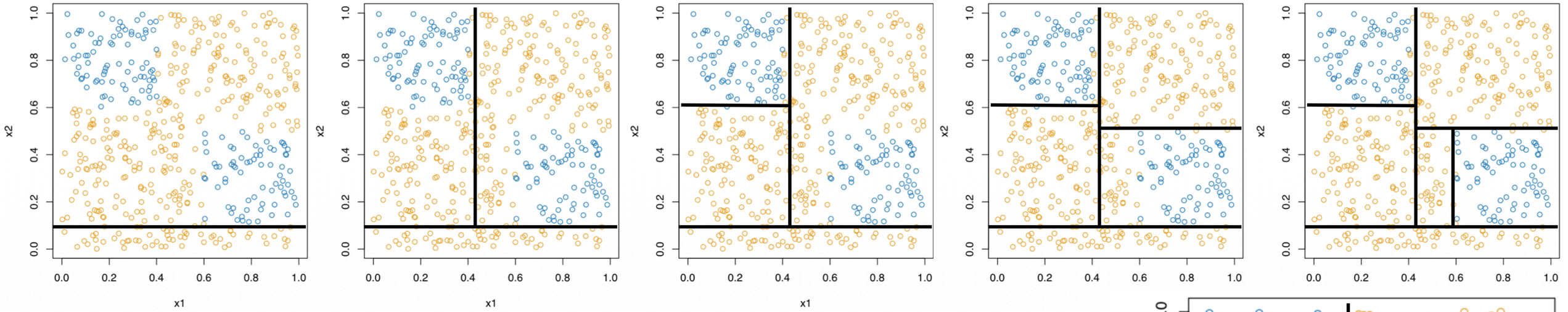


How to grow the tree?



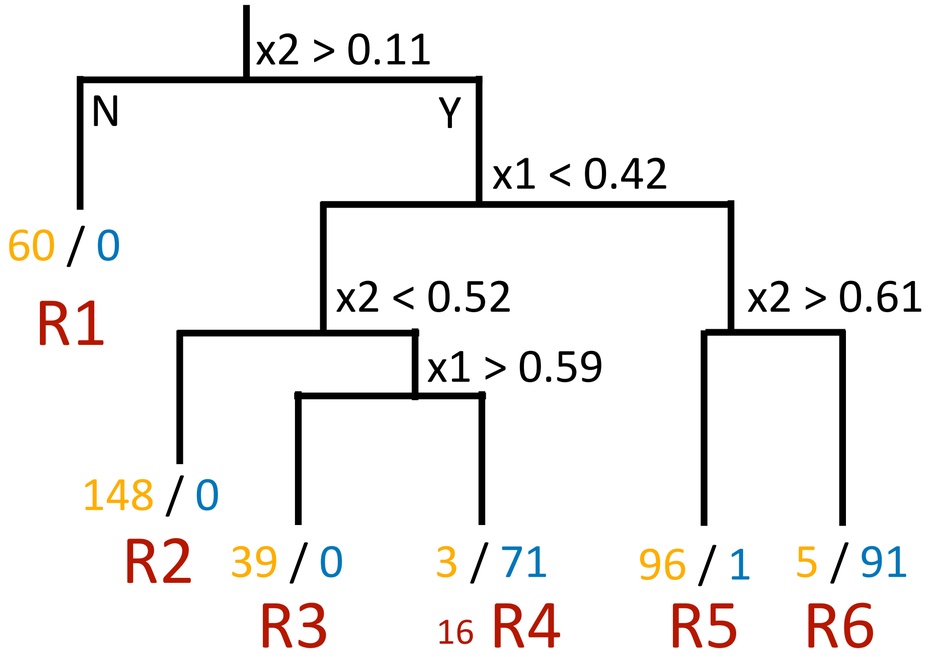


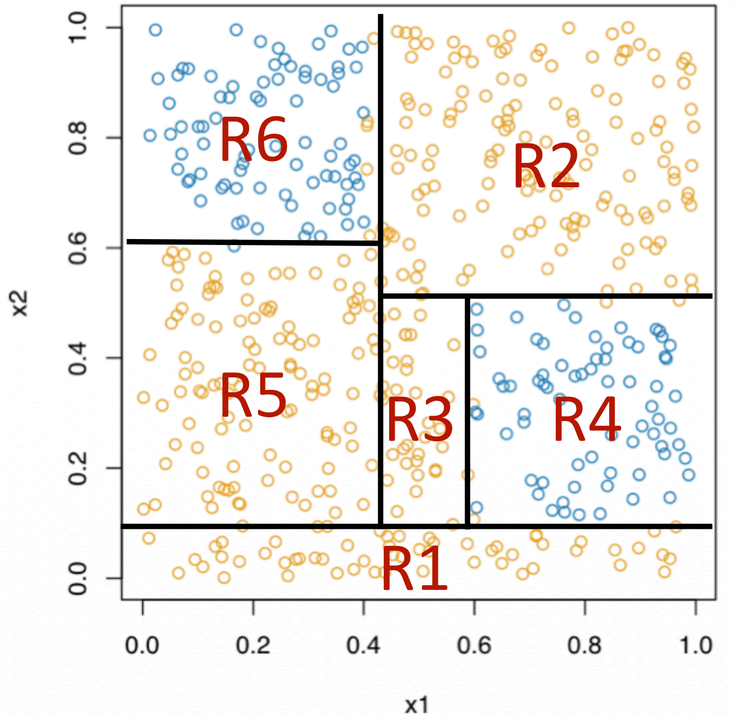
How to grow the tree?



Confusion matrix

True \ Predicted	Yellow	Blue
Yellow	343	8
Blue	1	162





How to grow the tree?

- This is called recursive binary partitioning
- A **greedy** algorithm
- At each step, we only care about the best partition at that moment
    - All past partitions are fixed
    - It doesn’t worry about any future partitions
- When do we stop? e.g.
    - Same number of leaves as observations, or
    - Each leaf has fewer than a small fixed number of observations

Regression trees

- For one of the predictors X_j with a cutoff threshold c, we define two regions:
  $R_1(j, c) = \{X \mid X_j < c\}$ and $R_2(j, c) = \{X \mid X_j \geq c\}$
- An observation x_i is in region $R_1(j, c)$ if $x_{ij} < c$ and in region $R_2(j, c)$ if $x_{ij} \geq c$
- For the first iteration, $R_1(j, c)$ and $R_2(j, c)$ partitions the entire predictor space
- For later iterations, $R_1(j, c)$ and $R_2(j, c)$ partitions a previous 'parent' region (so inherits any previous conditions of that 'parent' region)

Regression trees

- Suppose there are n_1 observations in $R_1(j, c)$ and n_2 observations in $R_2(j, c)$
- The prediction in each region is the mean response for observations in that region, i.e. $\hat{y}_{R_1} = \frac{1}{n_1} \sum_{i:\ x_i \in R_1(j,c)} y_i$ and $\hat{y}_{R_2} = \frac{1}{n_2} \sum_{i:\ x_i \in R_2(j,c)} y_i$
- Select the current partition to minimise the residual sum of squares (RSS), i.e. choose j and c to minimise:

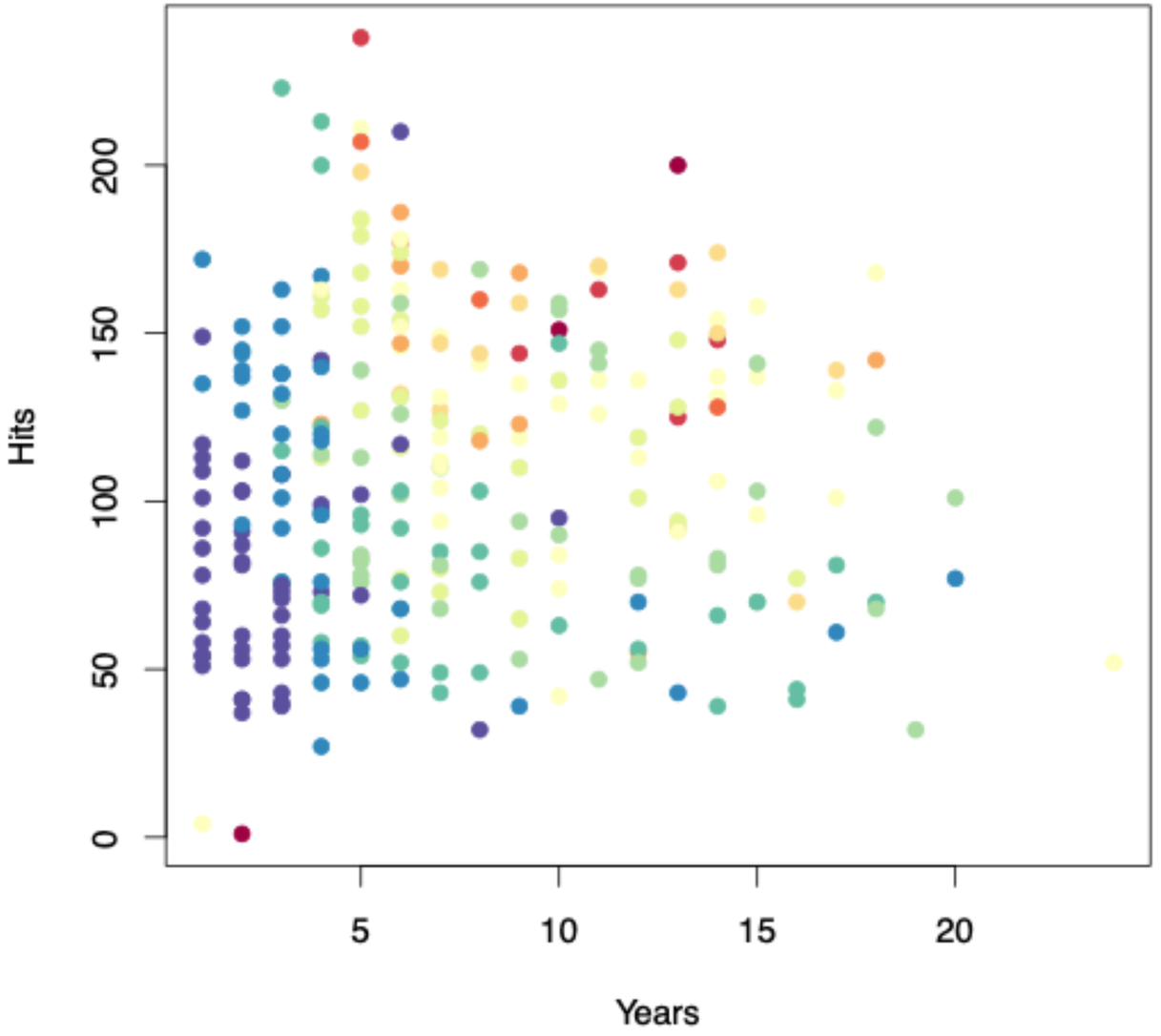
$$RSS(j, c) = \sum_{i:\ x_i \in R_1(j,c)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\ x_i \in R_2(j,c)} (y_i - \hat{y}_{R_2})^2$$

- How much does this reduce the RSS before the partition? $RSS = \sum_i (y_i - \hat{y})^2$

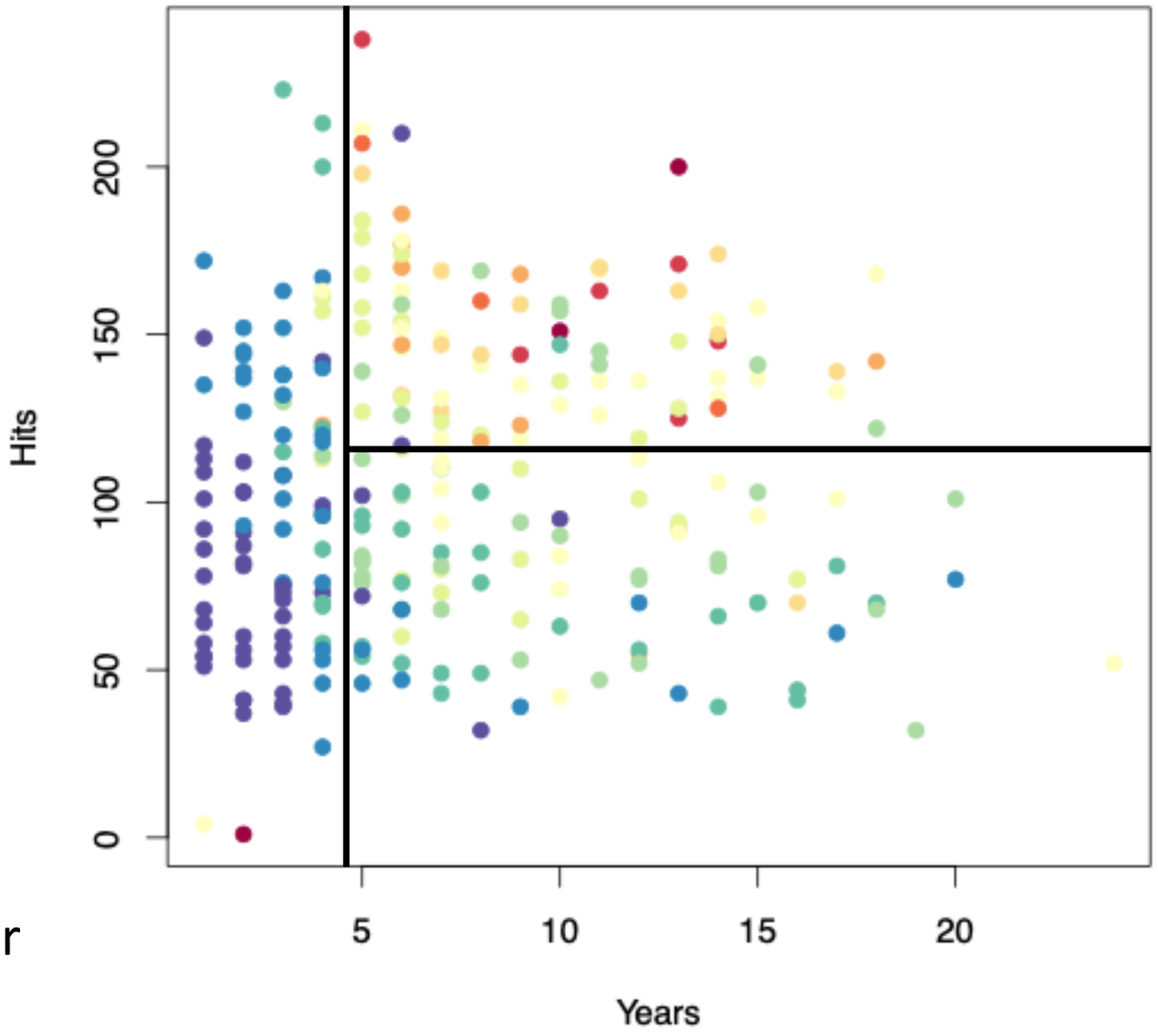
Example: baseball salaries

- X_1 is number of years experience
- X_2 is number of hits last year
- y is a log-transform of salary, measured in thousands of dollars i.e. the player's salary is $\$1000e^y$

- In the figure, y is represented by the colour of the dot (low - medium - high)

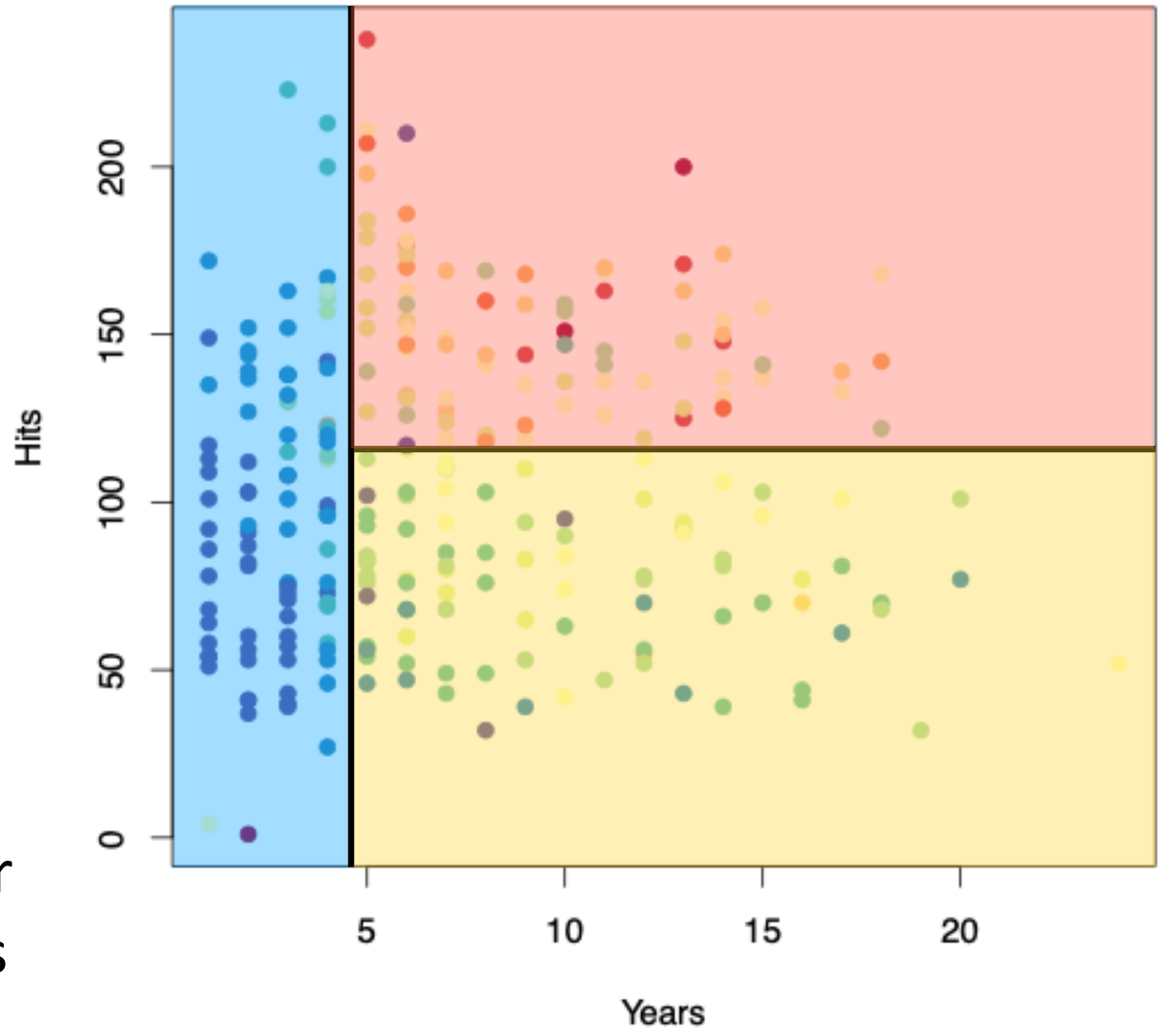


Baseball salary example



- How to interpret this?
  - Experience most important feature
  - If limited experience, hits isn't a big factor
  - If lots of experience, hits is a big factor

Baseball salary example



- How do we make predictions?
- How much do we expect a new player with 10 years experience and 200 hits to make? $\$1000e^{6.74} = \$845{,}346$

Questions?

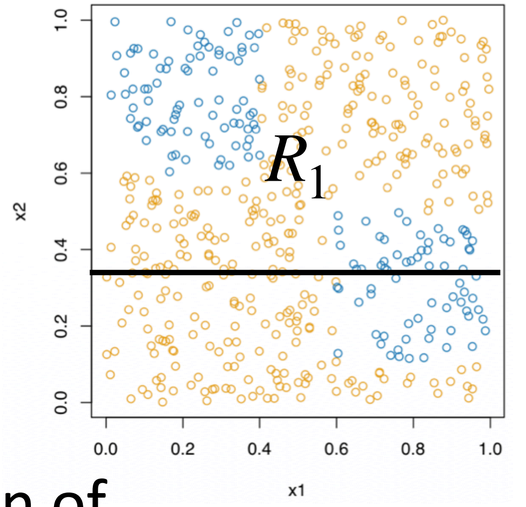
Classification trees

- The tree building process is very similar to before
- Now the output y is qualitative rather than quantitative, e.g. with K classes
- Instead of RSS, there are various criteria for making each partition, where $\hat{p}_{mk}$ is the proportion of observations within the region $R_m(j, c)$ belonging to the k^{th} class:
  - Classification error, $E_m(j, c) = 1 - \max_k(\hat{p}_{mk})$
  - Gini index, $G_m(j, c) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) = 1 - \sum_{k=1}^{K} \hat{p}_{mk}^2$
  - Entropy, $H_m(j, c) = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

Classification trees

- What is the total classification error or Gini index after the partition?
- A weighted average of the scores for each region (weighted by the number of observations in each region, n_1 and n_2)
  - Classification error, $E(j,c) = \frac{n_1}{n} E_1(j,c) + \frac{n_2}{n} E_2(j,c)$
  - Gini index, $G(j,c) = \frac{n_1}{n} G_1(j,c) + \frac{n_2}{n} G_2(j,c)$
- We'll come back to entropy shortly

Classification trees



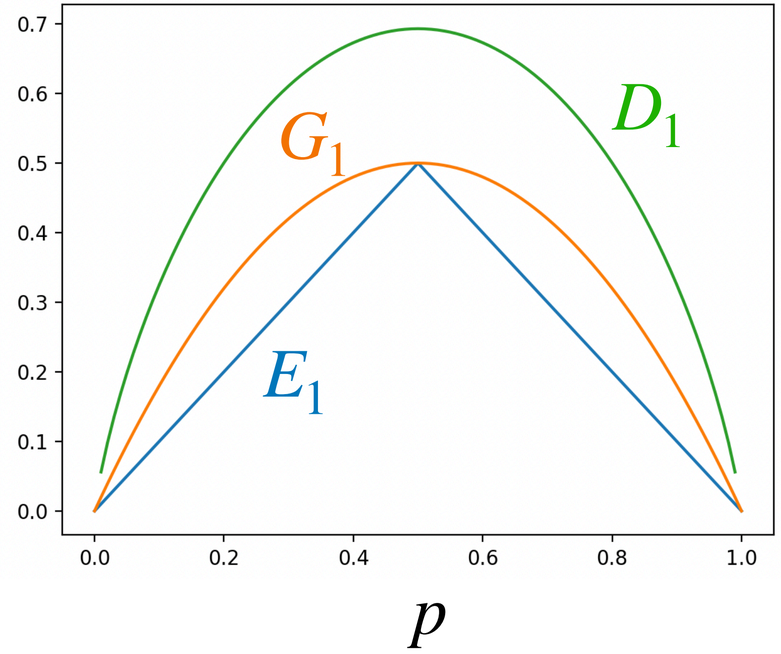
- What do this look like if there are $K = 2$ classes?

- Consider just region R_1 (with j and c fixed). If we write the proportion of observations in class $k = 1$ (e.g. yellow) as $\hat{p}_{11} = p$, then the proportion of observations in class $k = 2$ (e.g. blue) is $\hat{p}_{12} = (1 - p)$.

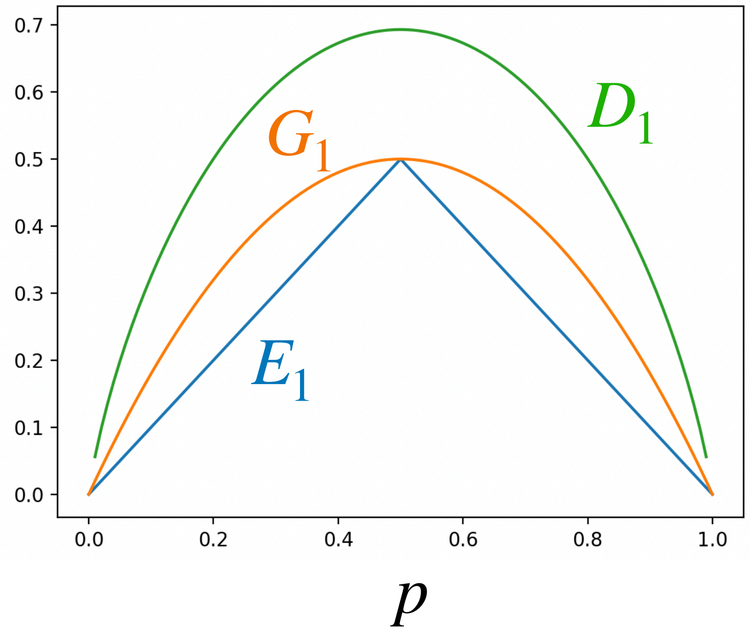
  - Classification error, $E_1 = \begin{cases} 1 - p & \text{if } p > 0.5 \\ p & \text{if } p < 0.5 \end{cases}$

  - Gini index, $G_1 = 1 - p^2 - (1 - p)^2 = 2p(1 - p)$

  - Entropy, $H_1 = -p \log p - (1 - p)\log(1 - p)$



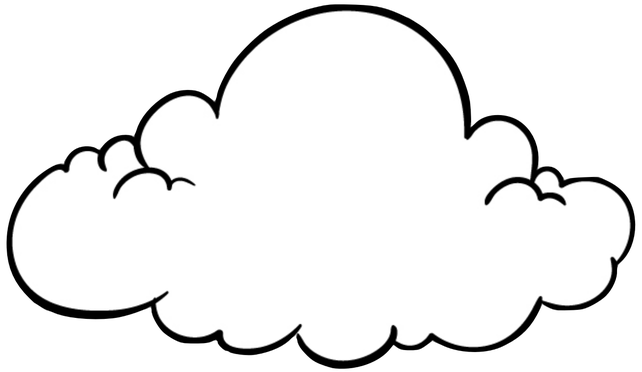
Classification trees



- We want **pure nodes** (almost all observations from one class), i.e. p close to 0 or close to 1
- The classification error is not particularly good at this!
- E.g. we would rather choose a partition that goes from $\hat{p}_{mk} = 0.8$ to $\hat{p}_{mk} = 0.9$ than a partition that goes from $\hat{p}_{mk} = 0.5$ to $\hat{p}_{mk} = 0.6$. The classification error improves by 0.1 in both cases.
- The Gini index improves by 0.14 when the partition goes from $\hat{p}_{mk} = 0.8$ to $\hat{p}_{mk} = 0.9$ and only by 0.02 when the partition goes from $\hat{p}_{mk} = 0.5$ to $\hat{p}_{mk} = 0.6$

Example: playing outdoors

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No



Example: playing outdoors

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

- Discrete predictors, binary outcome
- Class $k = 1$ is Play?=Yes, with proportion $\hat{p}_{11} = p = 0.5$
- Class $k = 2$ is Play?=No, with proportion $\hat{p}_{12} = 0.5$
- Initial classification (assume we classify all observations as Play?=Yes):
  - $E(\text{Init}) = 1 - 0.5 = 0.5$
  - $G(\text{Init}) = 1 - 0.5^2 - 0.5^2 = 0.5$

Example: playing outdoors

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

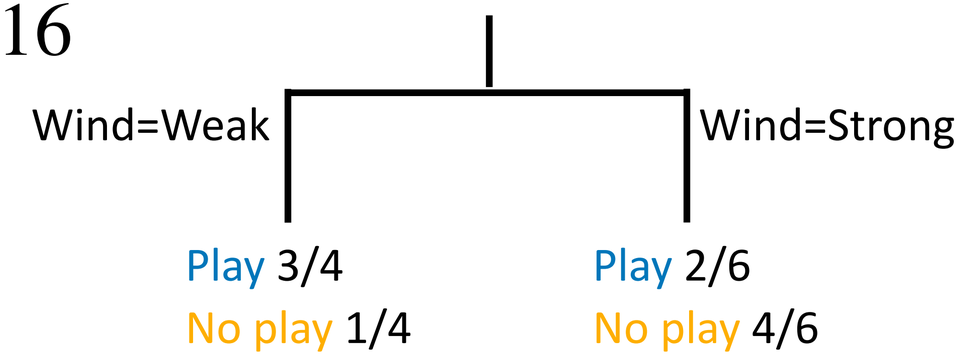
- Partition into R_1 (Wind=Weak) vs R_2 (Wind=Strong)
- $E_1 = 1 - 3/4 = 0.25$

  $E_2 = 1 - 2/3 \approx 0.33$

  $E(\text{Wind}) = 4/10 \times 0.25 + 6/10 \times 0.33 = 0.3$
- $G_1 = 1 - (3/4)^2 - (1/4)^2 = 0.375$

  $G_2 = 1 - (1/3)^2 - (2/3)^2 = 4/9 \approx 0.44$

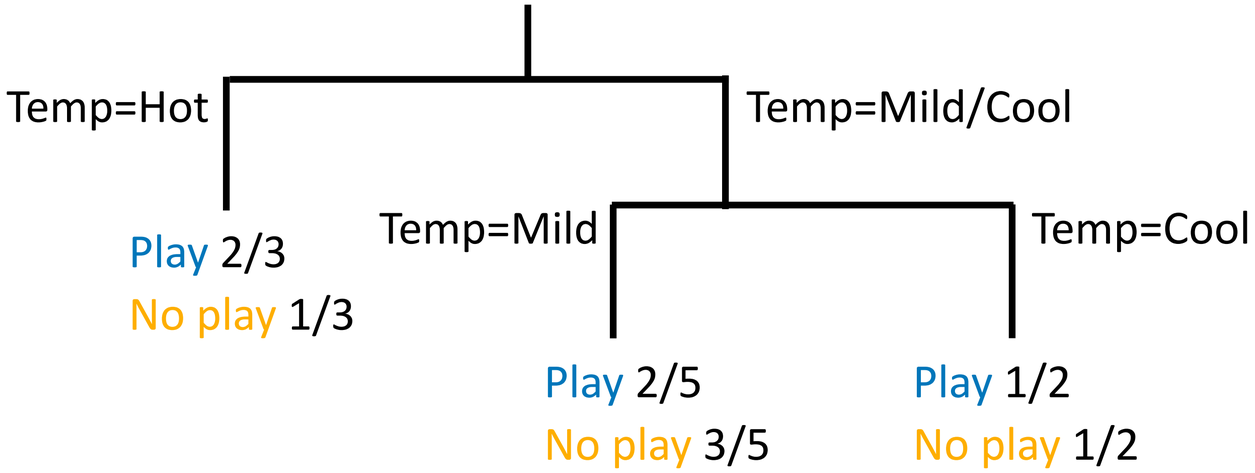
  $G(\text{Wind}) = 4/10 \times 0.375 + 6/10 \times 0.44 \approx 0.416$

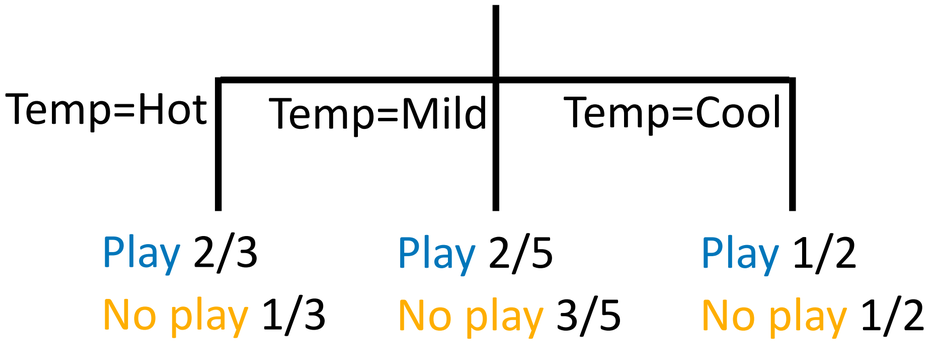


Example: playing outdoors

- Temperature has 3 categories, so partition into 3 branches
- E.g. partition into R_1 (Temp=Hot) vs R_2 (Temp=Mild/Cool), then partition R_2 into R_3 (Temp=Cool) vs R_4 (Temp=Mild)

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No





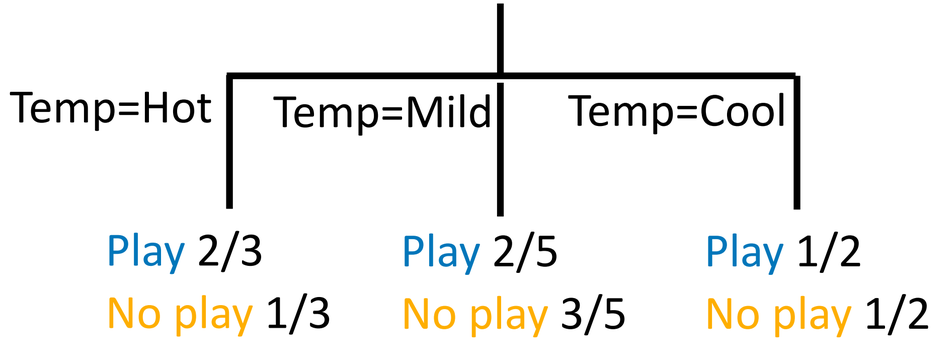
Example: playing outdoors

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

- Partition into R_1 (Temp=Hot) vs R_2 (Temp=Mild) vs R_3 (Temp=Cool) (renaming R_4 from previous slide to R_2)
- $E_1 = 1 - 2/3 \approx 0.33, \quad E_2 = 1 - 3/5 = 0.4$

  $E_3 = 1 - 1/2 = 0.5$

  $E(\text{Temp}) = 3/10 \times 0.33 + 5/10 \times 0.4 + 2/10 \times 0.5 = 0.4$



- $G_1 = 1 - (2/3)^2 - (1/3)^2 \approx 0.44$

  $G_2 = 1 - (2/5)^2 - (3/5)^2 = 0.48$

  $G_3 = 1 - (1/2)^2 - (1/2)^2 = 0.5$

  $G(\text{Temp}) = 3/10 \times 0.44 + 5/10 \times 0.48 + 2/10 \times 0.5 \approx 0.473$

Example: playing outdoors

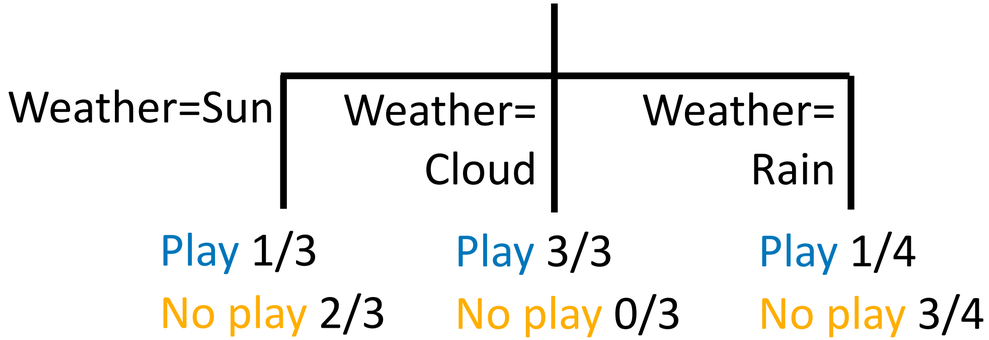
Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

- Partition into R_1 (Weather=Sun) vs R_2 (Weather=Cloud) vs R_3 (Weather=Rain)

- $E_1 = 1 - 2/3 \approx 0.33, \quad E_2 = 1 - 3/3 = 0$

  $E_3 = 1 - 3/4 = 0.25$

  $E(\text{Weather}) = 3/10 \times 0.33 + 3/10 \times 0 + 4/10 \times 0.25 = 0.2$



- $G_1 = 1 - (1/3)^2 - (2/3)^2 \approx 0.44$

  $G_2 = 1 - (3/3)^2 - (0/3)^2 = 0$

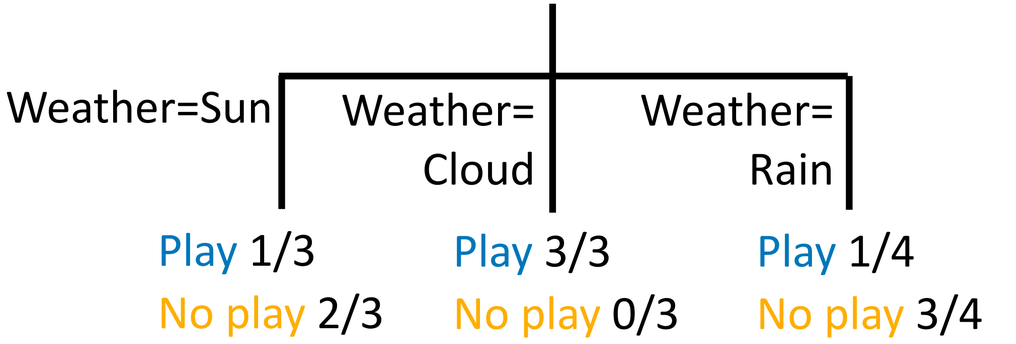
  $G_3 = 1 - (1/4)^2 - (3/4)^2 = 0.375$

  $G(\text{Weather}) = 3/10 \times 0.44 + 3/10 \times 0 + 4/10 \times 0.375 \approx 0.282$

Example: playing outdoors

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

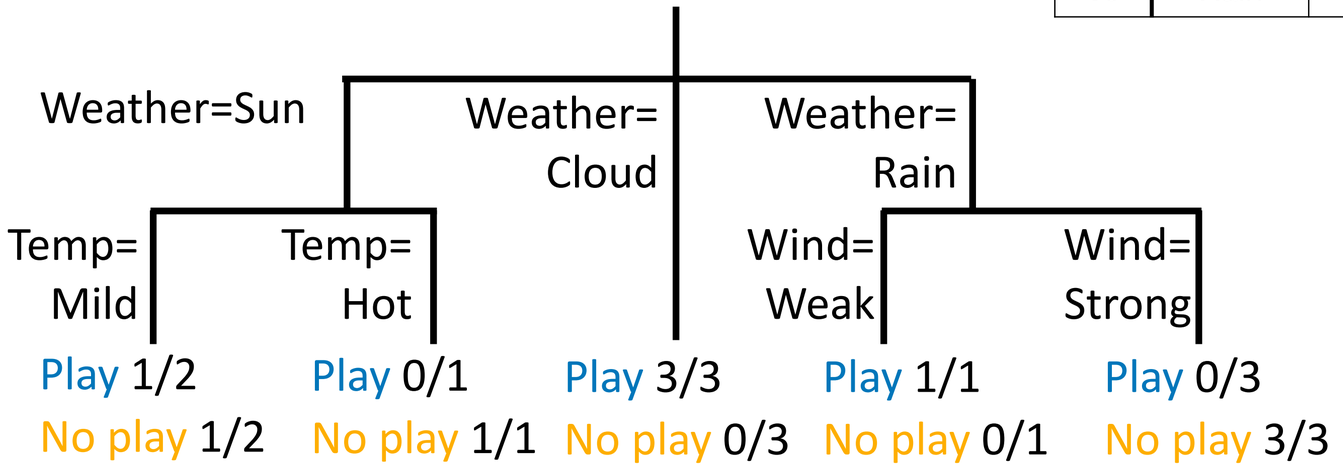
- $E(\text{Init}) = 0.5$
  $E(\text{Wind}) = 0.3,\ E(\text{Temp}) = 0.4,\ E(\text{Weather}) = 0.2$
- $G(\text{Init}) = 0.5$
  $G(\text{Wind}) \approx 0.416,\ G(\text{Temp}) \approx 0.473$
  $G(\text{Weather}) \approx 0.282$
- The best partition (biggest decrease) is by Weather, using either the classification error or the Gini index



Example: playing outdoors

- Next: consider partitioning R_1 (Weather=Sun) by Temp or Wind OR partitioning R_3 (Weather=Rain) by Temp or Wind
- A final decision tree:

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

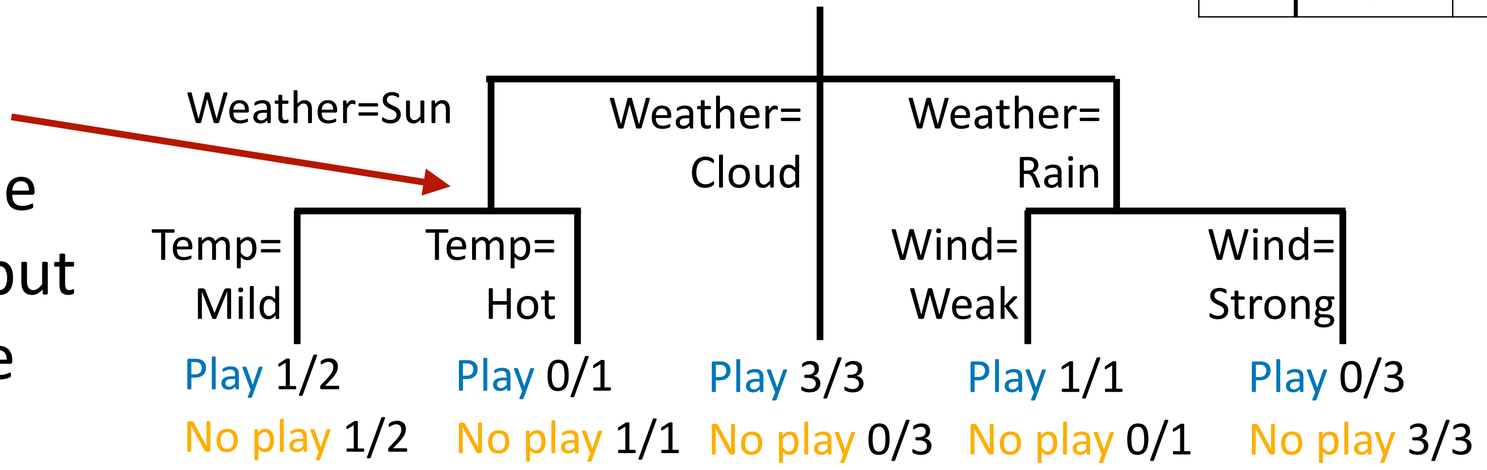


Example: playing outdoors

- Next: consider partitioning R_1 (Weather=Sun) by Temp or Wind OR partitioning R_3 (Weather=Rain) by Temp or Wind
- A final decision tree:

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

Note: this partition doesn't decrease the classification error but it does decrease the Gini index!



Entropy and information gain

- Entropy, measured in nats:

$$H(Y) = -\sum_k P(Y=k)\log P(Y=k) = -\sum_k p_k \log p_k$$

- Entropy, measured in bits: $H(Y) = -\sum_k p_k \log_2 p_k$ (we'll use this for now)

- Suppose we know $X_j = x$, then

$$H(Y \mid X_j = x) = -\sum_k P(Y=k \mid X_j = x)\log_2 P(Y=k \mid X_j = x)$$

- The **information gain** is the amount that this reduces the entropy of Y, i.e.

$$IG(Y, X_j = x) = H(Y) - H(Y \mid X_j = x)$$

Entropy and information gain

- We can use the **average information gain** as another way to calculate the 'best' partition, i.e. $IG(Y, X_j) = H(Y) - \sum_x p(X_j = x) H(Y \mid X_j = x)$

Entropy and information gain

- We can use the **average information gain** as another way to calculate the 'best' partition, i.e. $IG(Y, X_j) = H(Y) - \sum_x p(X_j = x) H(Y \mid X_j = x)$

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

- For the example, $H(\text{Play}) = -\sum_k P(\text{Play} = k) \log_2 P(\text{Play} = k)$

  $= -5/10 \log_2(5/10) - 5/10 \log_2(5/10) = 1$

- $H(\text{Play} \mid \text{Wind} = \text{Strong}) = -3/4 \log(3/4) - 1/4 \log(1/4) \approx 0.811$

  $H(\text{Play} \mid \text{Wind} = \text{Weak}) = -2/3 \log(2/3) - 1/3 \log(1/3) \approx 0.918$

Entropy and information gain

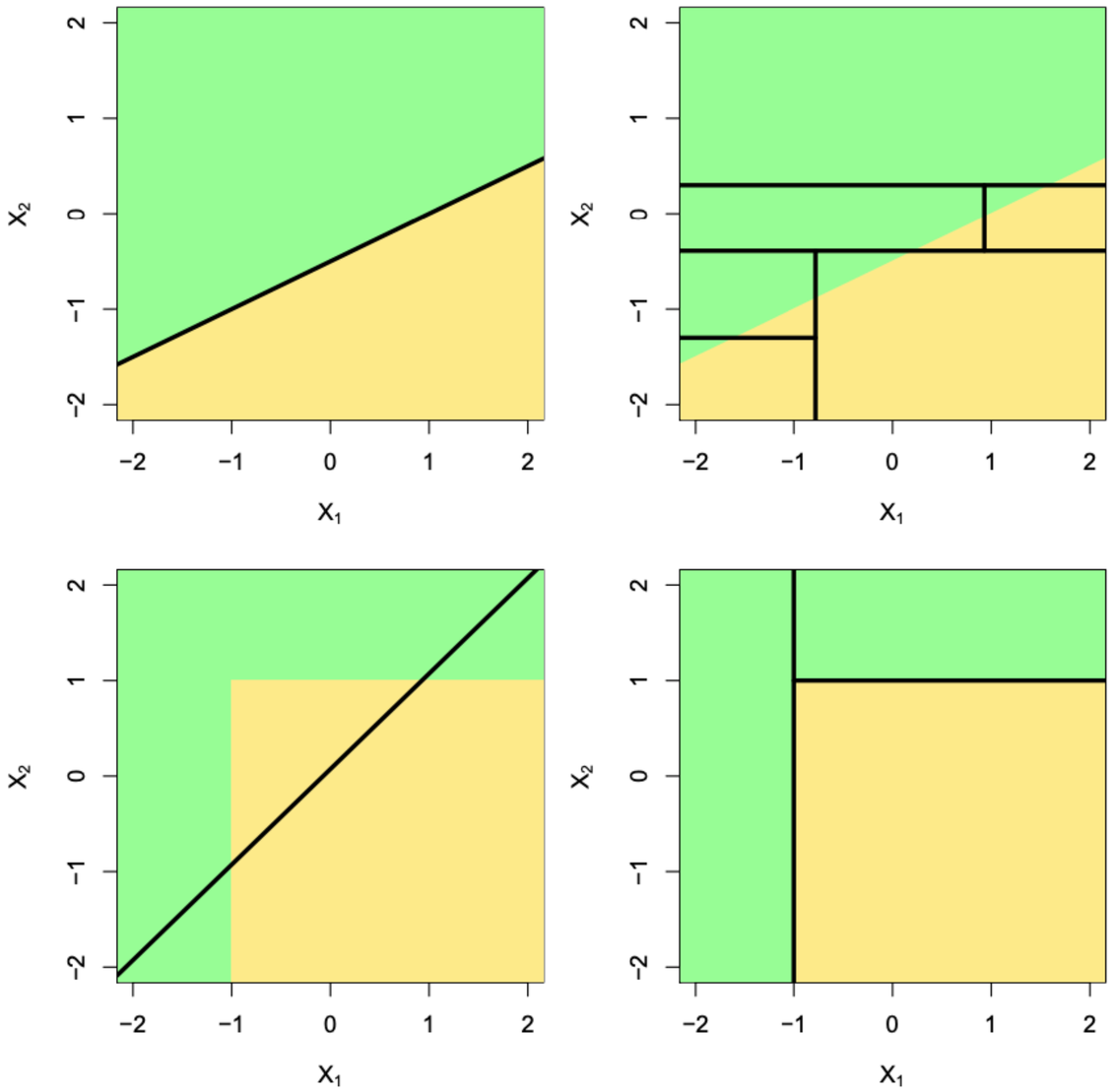
- We can use the **average information gain** as another way to calculate the 'best' partition, i.e. $IG(Y, X_j) = H(Y) - \sum_x p(X_j = x) H(Y \mid X_j = x)$

Day	Weather	Temperature	Wind	Play?
1	Sun	Hot	Weak	No
2	Cloud	Hot	Weak	Yes
3	Sun	Mild	Strong	Yes
4	Cloud	Mild	Strong	Yes
5	Rain	Mild	Strong	No
6	Rain	Cool	Strong	No
7	Rain	Cool	Weak	Yes
8	Sun	Mild	Strong	No
9	Cloud	Hot	Weak	Yes
10	Rain	Mild	Strong	No

- For the example, $H(\text{Play}) = -\sum_k P(\text{Play} = k) \log_2 P(\text{Play} = k)$
  $= -5/10 \log_2(5/10) - 5/10 \log_2(5/10) = 1$
- $H(\text{Play} \mid \text{Wind} = \text{Strong}) = -3/4 \log(3/4) - 1/4 \log(1/4) \approx 0.811$
  $H(\text{Play} \mid \text{Wind} = \text{Weak}) = -2/3 \log(2/3) - 1/3 \log(1/3) \approx 0.918$
- $IG(\text{Play}, \text{Wind}) = 1 - 4/10 \times 0.811 - 6/10 \times 0.918 \approx 0.1245$
- Similarly, $IG(\text{Play}, \text{Temp}) \approx 0.2116$, $IG(\text{Play}, \text{Weather}) = 0.4$

Largest information gain

Decision trees vs linear regression



$$y \approx f(X) = \beta_0 + \sum_j X_j \beta_j$$

$$y \approx f(X) = \sum_m c_m 1\{X \in R_m\}$$

Decision trees: pros and cons

- Trees are very interpretable! You can easily read off why the model gave a particular prediction (it's a white box model) ✓
- Trees are non-parametric, they don't require assumptions about the data or residuals ✓
- Trees handle mixed data types (both predictors and response variables) ✓
- Trees can handle missing data (e.g. using surrogate variables) ✓
- Trees are not very robust. Small changes to the input data can completely change the tree structure. ✘
- Most of the time, trees aren't particularly good at prediction. ✘
- Does this mirror human intuition better than other methods for prediction or classification?

When do we stop growing the tree?

- Which is better: small trees or big trees?

Questions?

- Feel free to email me at te269@cam.ac.uk

- These slides will go on GitHub later today,

  https://github.com/tedinburgh/ads2023

Next time

- Pruning
- Ensembles and aggregates
  - Bagging
  - Boosting
  - Random forests