Introduction to Machine Learning

Andrea De Lorenzo

A.Y.2020

Section 1

General information

Lecturers

- Andrea De Lorenzo
  - Dipartimento di Ingegneria e Architettura (DIA)
  - `http://delorenzo.inginf.units.it/`

Course materials

- Lecturer's slides
  - `http://delorenzo.inginf.units.it/project/introduction-to-machine-learning-2020`
- Suggested textbooks (for further reading)
  - Gareth James et al. *An introduction to statistical learning*. Vol. 6. Springer, 2013
- Other material:
  - I might point you to some scientific papers for discussing examples of application or specific details—just a "chat"

Everything you are required to know is in the lecturer's slides

Section 2

Introduction

What is Machine Learning?

Definition

Machine Learning is the science of getting computer to learn without being explicitly programmed.

Definition

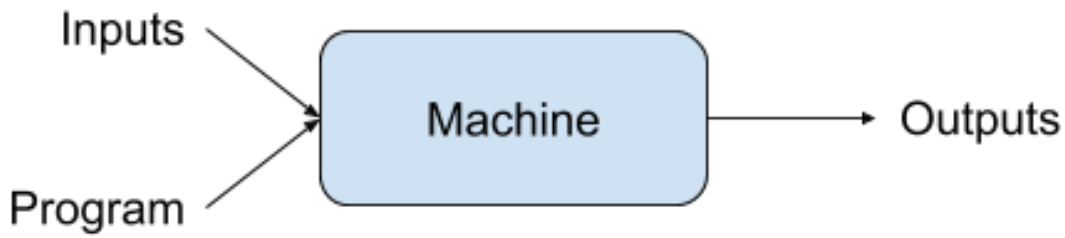
Data Mining/Analytics is the science of discovering patterns in data.

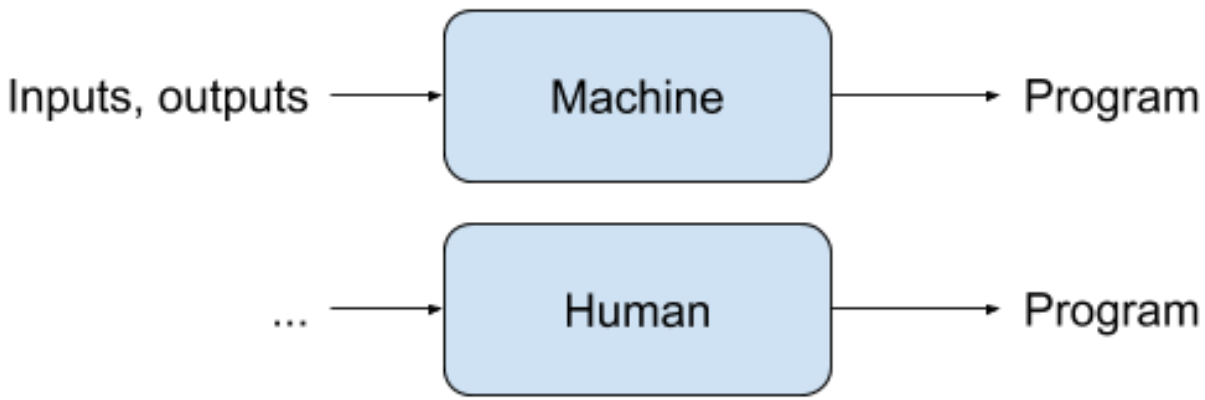
In practice

A set of mathematical and statistical tools for:

- building a model which allows to predict an output, given an input (*supervised learning*)
  - example $\langle$input, output$\rangle$ pairs are available
- learn relationships and structures in data (*unsupervised learning*)

Machine Learning: a computer science perspective





Machine Learning everyday

Example problem: spam

Discriminate between spam and non-spam emails.

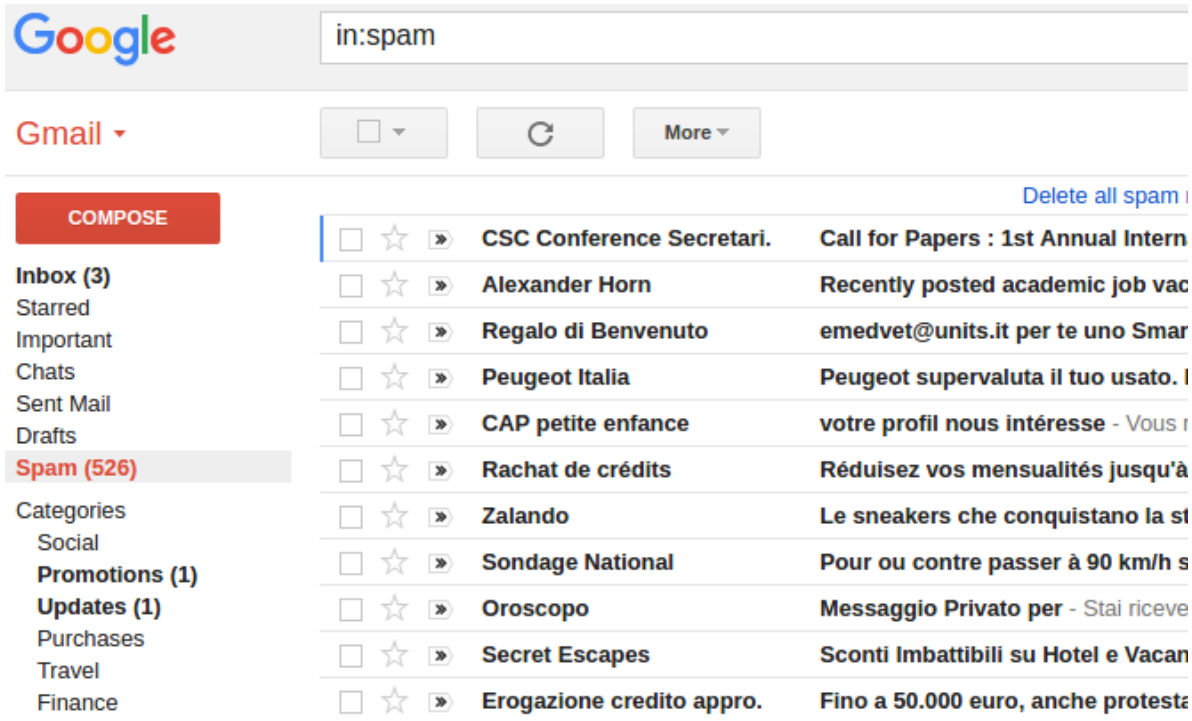


Figure: Spam filtering in Gmail.

Machine Learning everyday

Example problem: flight trajectories

Do flights over the same pair ⟨origin, destination⟩ follow the "same" trajectory? Why?

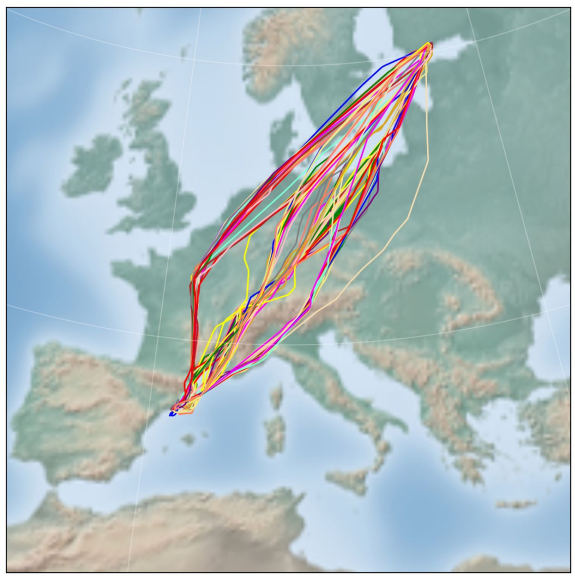


Figure: Clustering of flight trajectories.

Machine Learning everyday

Example problem: image understanding

Recognize objects in images.

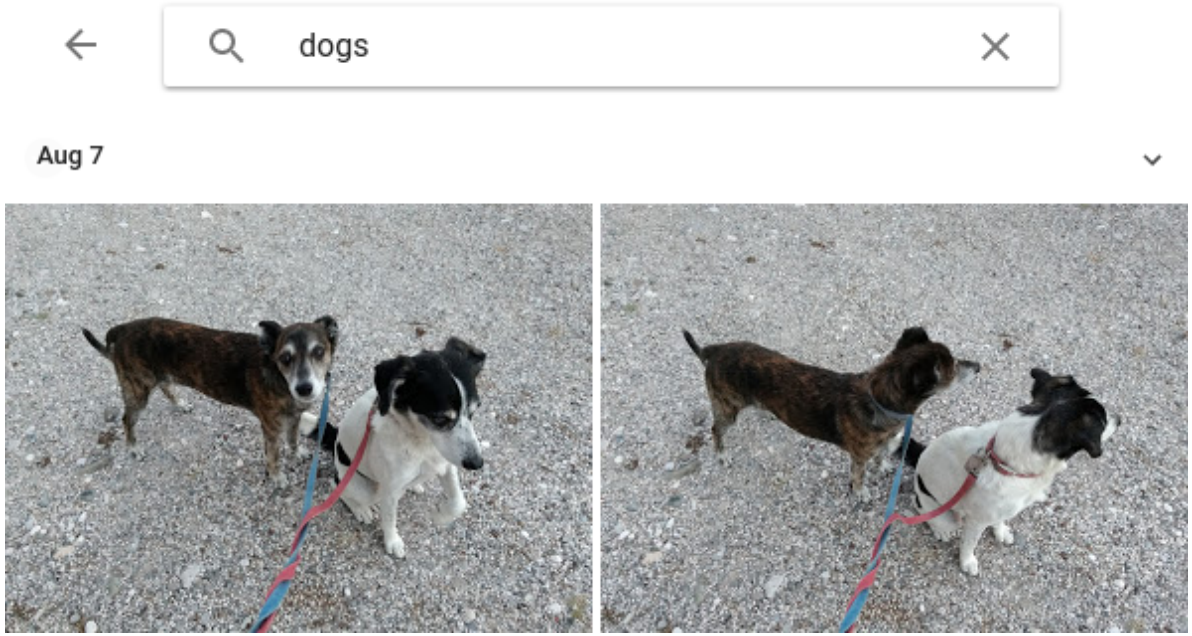


Figure: Object recognition in Google Photos.

Machine Learning everyday

Q: what type of learning (supervised/unsupervised) is in the examples?

- spam
- image understanding
- flight trajectories

Why ML/DM "today"?

- we collect more and more data (*big data*)
- we have more and more computational power

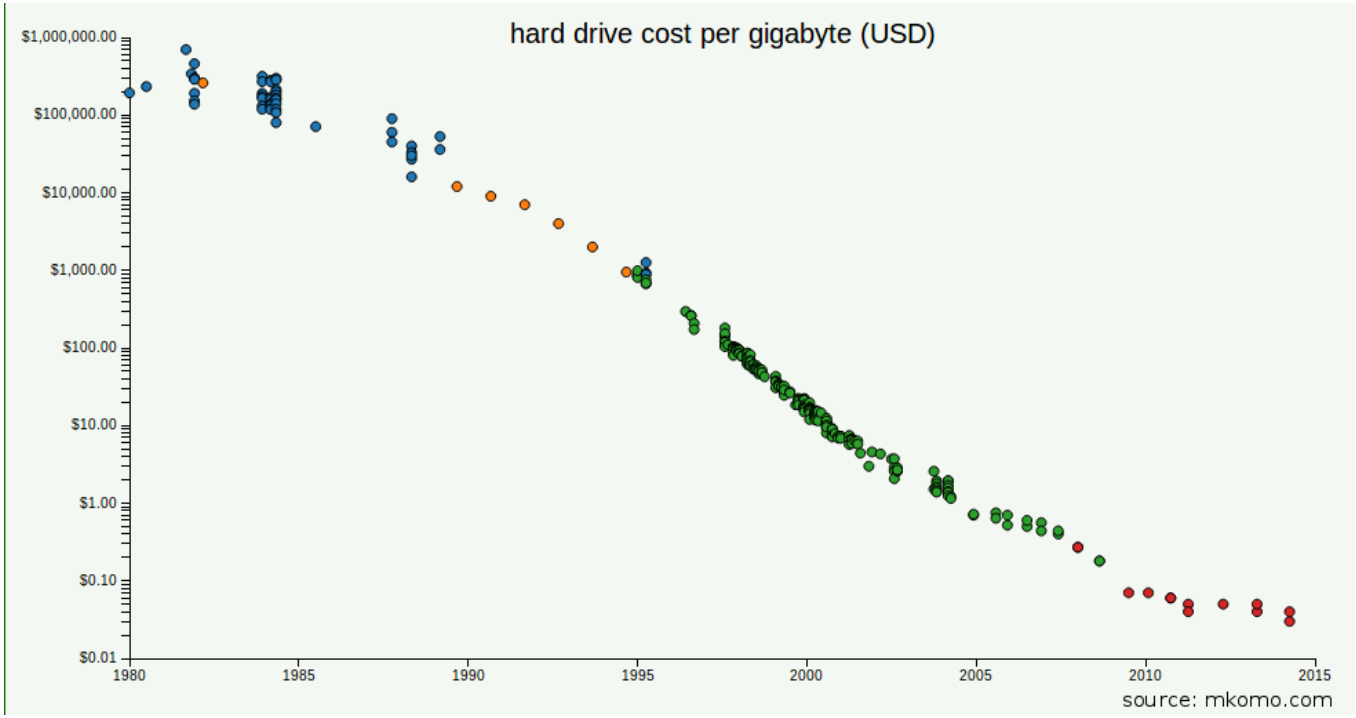


Figure: From `http://www.mkomo.com/cost-per-gigabyte-update`.

ML/DM is popular!

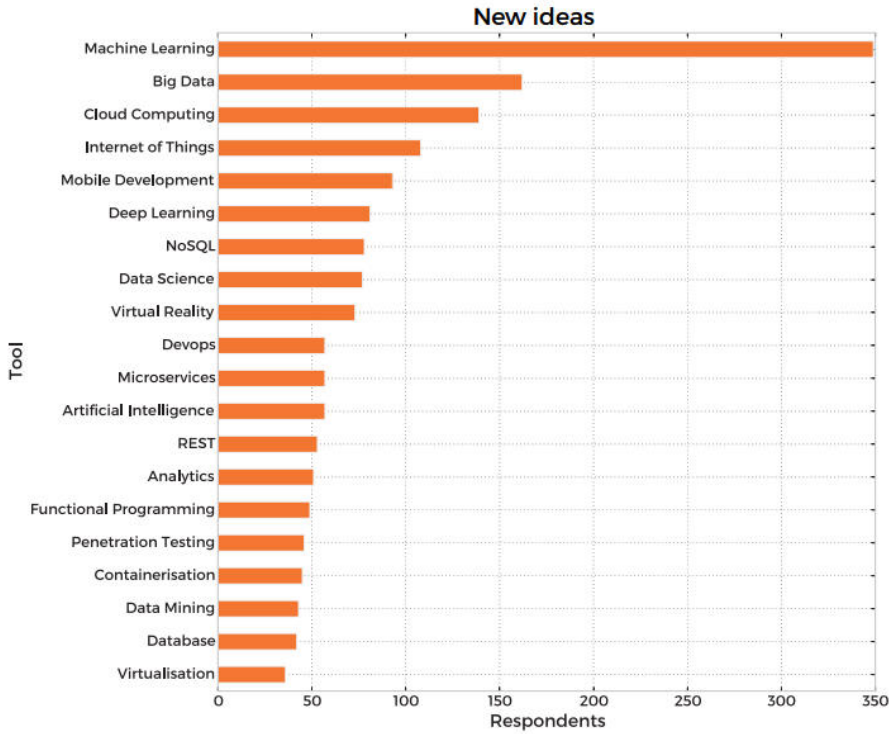


Figure: Popular areas of interest, from the Skill Up 2016: Developer Skills Report[2]

[1] https://techcus.com/p/r1zSmbXut/top-5-highest-paying-programming-languages-of-2016/.

[2] https://techcus.com/p/r1zSmbXut/top-5-highest-paying-programming-languages-of-2016/.

Aims of the course

Be able to:

1. design
2. implement
3. assess experimentally

an end-to-end Machine Learning or Data Mining system.

Aims of the course

Be able to:

1. design
2. implement
3. assess experimentally

an end-to-end Machine Learning or Data Mining system.

- Which is the problem to be solved? Which are the input and output? Which are the most suitable techniques? How should data be prepared? Does computation time matter?

Aims of the course

Be able to:

1. design
2. implement
3. assess experimentally

an end-to-end Machine Learning or Data Mining system.

- Which is the problem to be solved? Which are the input and output? Which are the most suitable techniques? How should data be prepared? Does computation time matter?
- Write some code!

Aims of the course

Be able to:

1. design
2. implement
3. assess experimentally

an end-to-end Machine Learning or Data Mining system.

- Which is the problem to be solved? Which are the input and output? Which are the most suitable techniques? How should data be prepared? Does computation time matter?
- Write some code!
- How to measure solution quality? How to compare solutions? Is my solution general?

Aims of the course

Be able to:

1. design
2. implement
3. assess experimentally

an end-to-end Machine Learning or Data Mining system.

- Which is the problem to be solved? Which are the input and output? Which are the most suitable techniques? How should data be prepared? Does computation time matter?
- Write some code!
- How to measure solution quality? How to compare solutions? Is my solution general?
  - Itself: design and implementation

Aims of the course: communication

Be able to:

1. design
2. implement
3. assess experimentally

an end-to-end Machine Learning or Data Mining system.
And be able to convince the "client" that it is:

- technically sound
- economically viable
- in its larger context

Subsection 1

Motivating example

The amateur botanist friend

He likes to collect Iris plants. He "realized" that there are 3 species, in particular, that he likes: *Iris setosa*, *Iris virginica*, and *Iris versicolor*. He'd like to have a tool to automatically *classify* collected samples in one of the 3 species.

Figure: Iris versicolor.

How to help him?

Let's help him

- Which is the problem to be solved?

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.
- Which are the input and output?

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.
- Which are the input and output?
  - Output: one species among I. setosa, I. virginica, I. versicolor.

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.
- Which are the input and output?
  - Output: one species among I. setosa, I. virginica, I. versicolor.
  - Input: the plant sample...

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.
- Which are the input and output?
  - Output: one species among I. setosa, I. virginica, I. versicolor.
  - Input: the plant sample...
    - a description in natural language?

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.
- Which are the input and output?
  - Output: one species among I. setosa, I. virginica, I. versicolor.
  - Input: the plant sample...
    - a description in natural language?
    - a digital photo?

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.
- Which are the input and output?
  - Output: one species among I. setosa, I. virginica, I. versicolor.
  - Input: the plant sample...
    - a description in natural language?
    - a digital photo?
    - DNA sequences?

Let's help him

- Which is the problem to be solved?
  - Assign exactly one specie to a sample.
- Which are the input and output?
  - Output: one species among I. setosa, I. virginica, I. versicolor.
  - Input: the plant sample...
    - a description in natural language?
    - a digital photo?
    - DNA sequences?
    - some measurements of the sample!

Iris: input and output

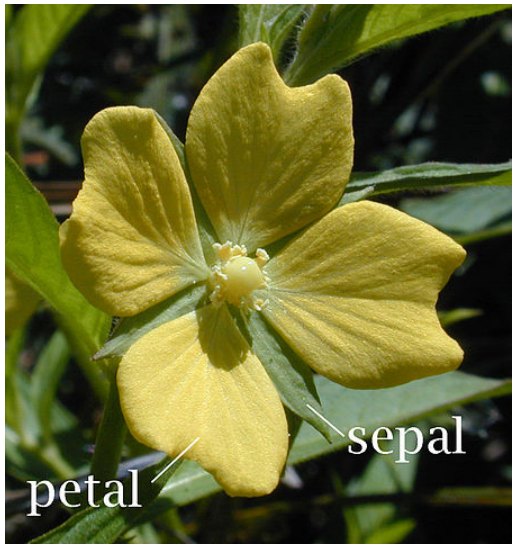


Figure: Sepal and petal.

Input: sepal length and width, petal length and width (in cm)
Output: the class
Example: $(5.1, 3.5, 1.4, 0.2) \rightarrow$ I. setosa

Other information

The botanist friend asked a senior botanist to inspect several samples and label them with the corresponding species.

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
7.0	3.2	4.7	1.4	I. versicolor
6.0	2.2	5.0	1.5	I. virginica

Notation and terminology

- Sepal length, sepal width, petal length, and petal width are input variables (or independent variables, or features, or attributes).
- Species is the output variable (or dependent variable, or response).

Notation and terminology

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

- $x_1^T = (x_{1,1}, x_{1,2}, \ldots, x_{1,p})$ is an observation (or instance, or data point), composed of p variable values;

Notation and terminology

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

- $x_1^T = (x_{1,1}, x_{1,2}, \ldots, x_{1,p})$ is an observation (or instance, or data point), composed of p variable values; y_1 is the corresponding output variable value

Notation and terminology

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

- $x_1^T = (x_{1,1}, x_{1,2}, \ldots, x_{1,p})$ is an observation (or instance, or data point), composed of p variable values; y_1 is the corresponding output variable value
- $\mathbf{x}_2^T = (x_{1,2}, x_{2,2}, \ldots, x_{n,2})$ is the vector of all the n values for the 2nd variable (X_2).

Notation and terminology

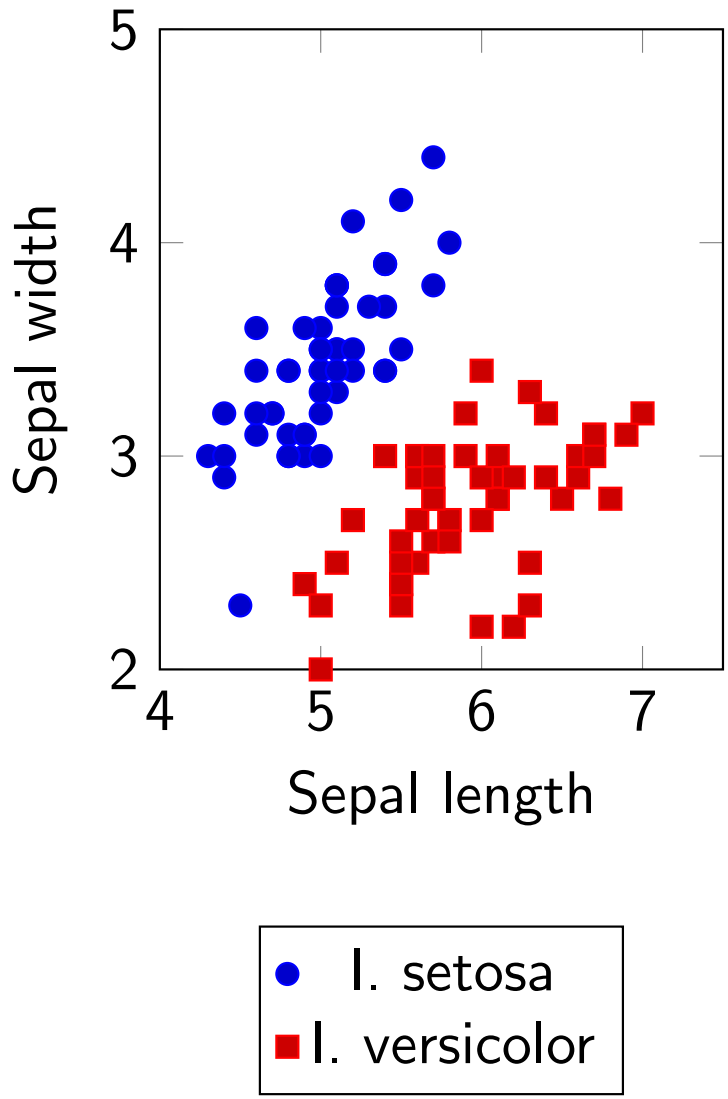
Different communities (e.g., statistical learning vs. machine learning vs. artificial intelligence) use different terms and notation:

- $x_j^{(i)}$ instead of $x_{i,j}$ (hence $x^{(i)}$ instead of x_i)
- m instead of n and n instead of p
- ...

Focus on the meaning!

Iris: visual interpretation

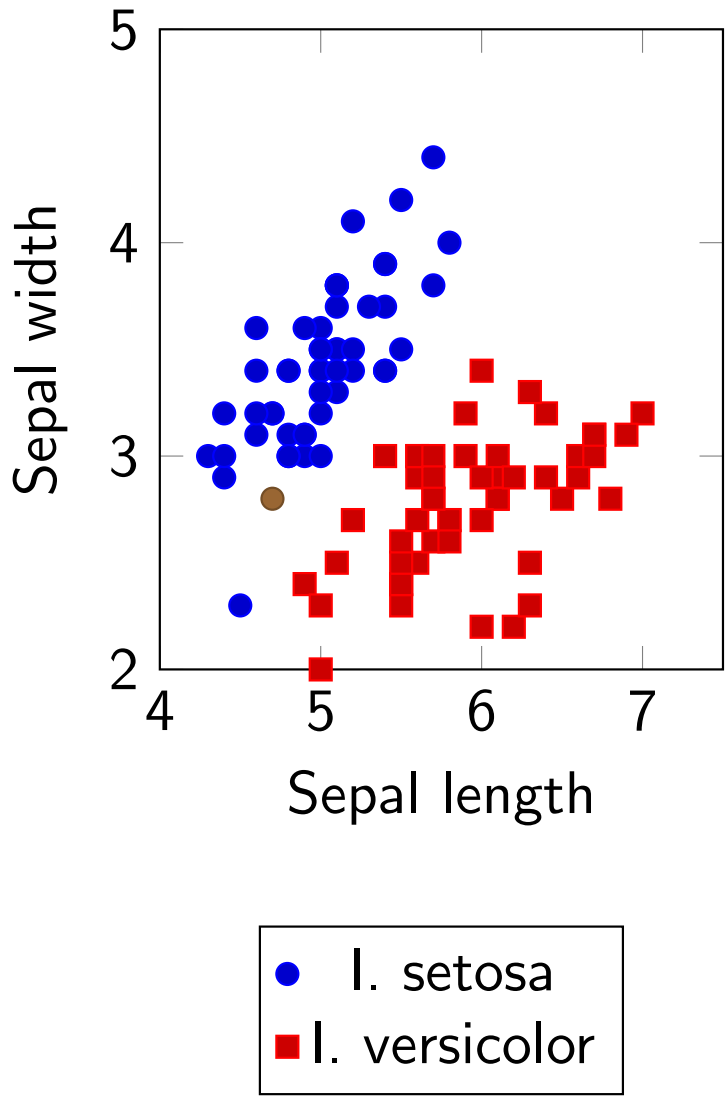
Simplification: forget petal and I. virginica $\rightarrow$ 2 variables, 2 species (binary classification problem).



Iris: visual interpretation

Simplification: forget petal and I. virginica $\rightarrow$ 2 variables, 2 species (binary classification problem).

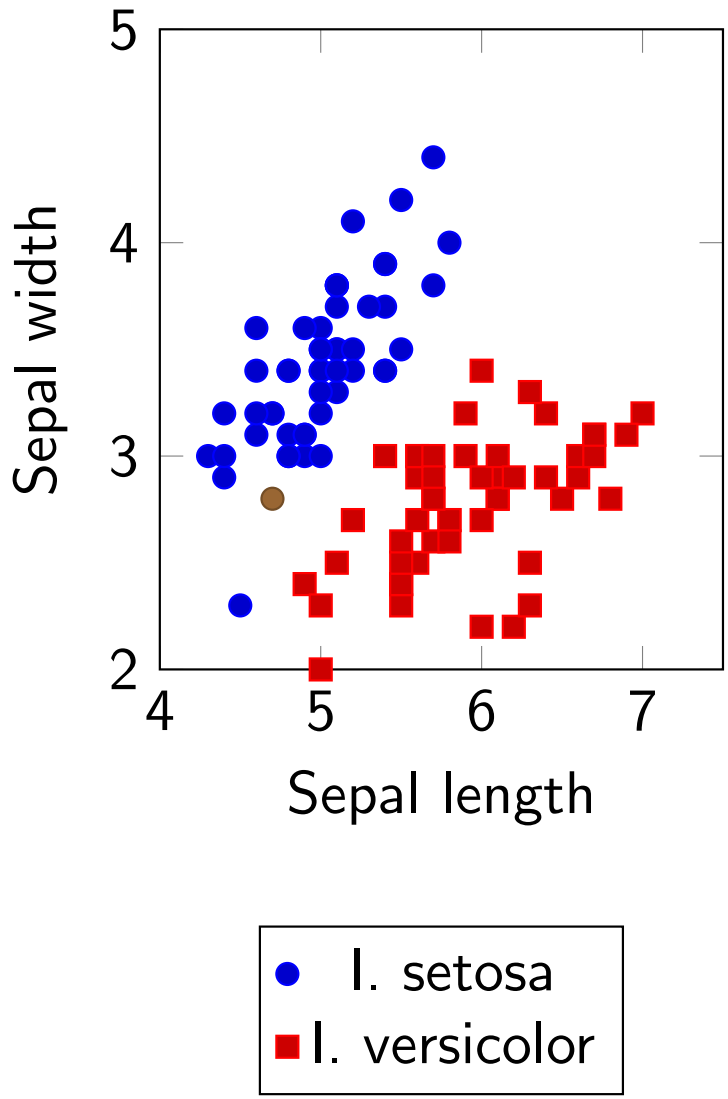
- *Problem*: given any new observation, we want to automatically assign the species.



Iris: visual interpretation

Simplification: forget petal and I. virginica $\rightarrow$ 2 variables, 2 species (binary classification problem).

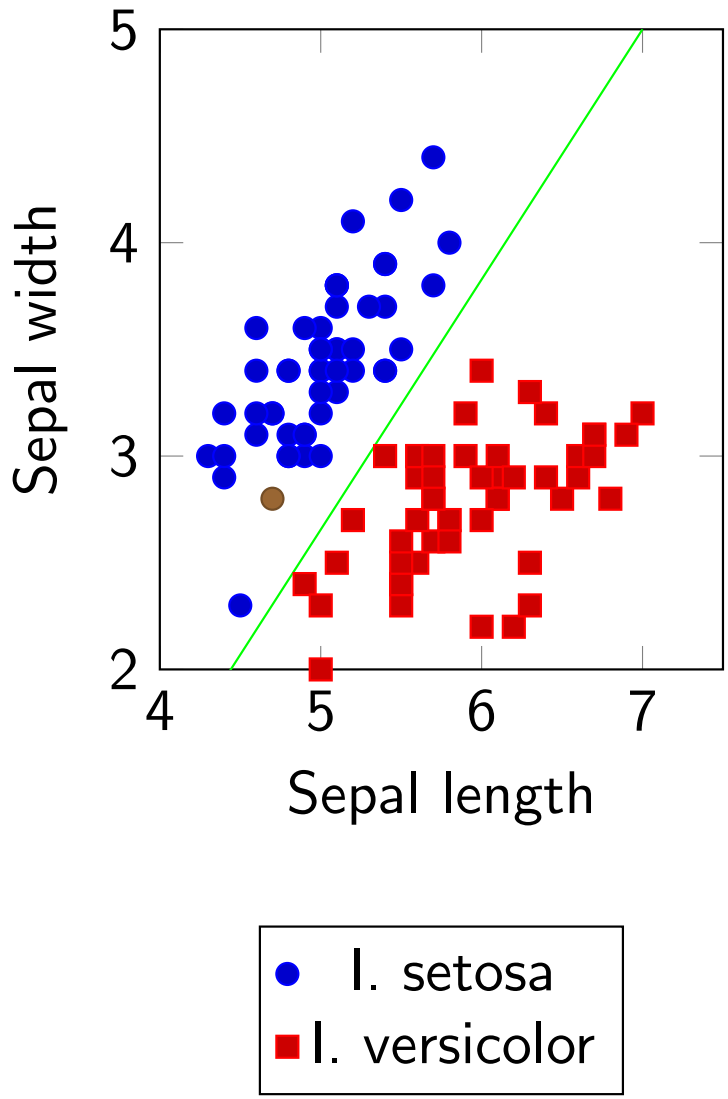
- *Problem*: given any new observation, we want to automatically assign the species.
- *Sketch of a possible solution*:



Iris: visual interpretation

Simplification: forget petal and I. virginica $\rightarrow$ 2 variables, 2 species (binary classification problem).

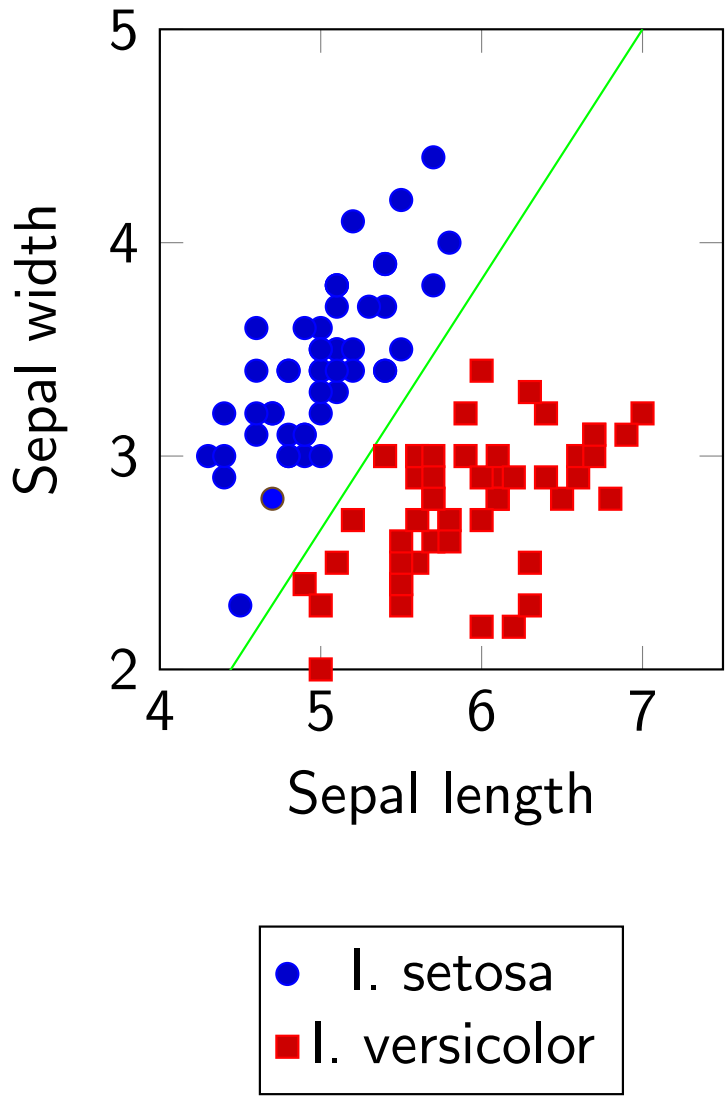
- *Problem*: given any new observation, we want to automatically assign the species.
- *Sketch of a possible solution*:
  1. learn a model (classifier)



Iris: visual interpretation

Simplification: forget petal and I. virginica $\rightarrow$ 2 variables, 2 species (binary classification problem).

- *Problem*: given any new observation, we want to automatically assign the species.
- *Sketch of a possible solution*:
  1. learn a model (classifier)
  2. "use" model on new observations



"A" model?

There could be many possible models:

- how to choose?
- how to compare?

Q: a model of what?

Choosing the model

The choice of the model/tool/technique to be used is determined by many factors:

- Problem size (n and p)
- Availability of an output variable (y)
- Computational effort (when learning or "using")
- Explicability of the model
- ...

We will see some options.

Comparing many models

Experimentally: does the model work well on (new) data?

Comparing many models

Experimentally: does the model work well on (new) data?
Define "works well":

- a single performance index?
- how to measure?
- repeatability/reproducibility...
  - **Q:** what's the difference?

We will see/discuss some options.

It does not work well...

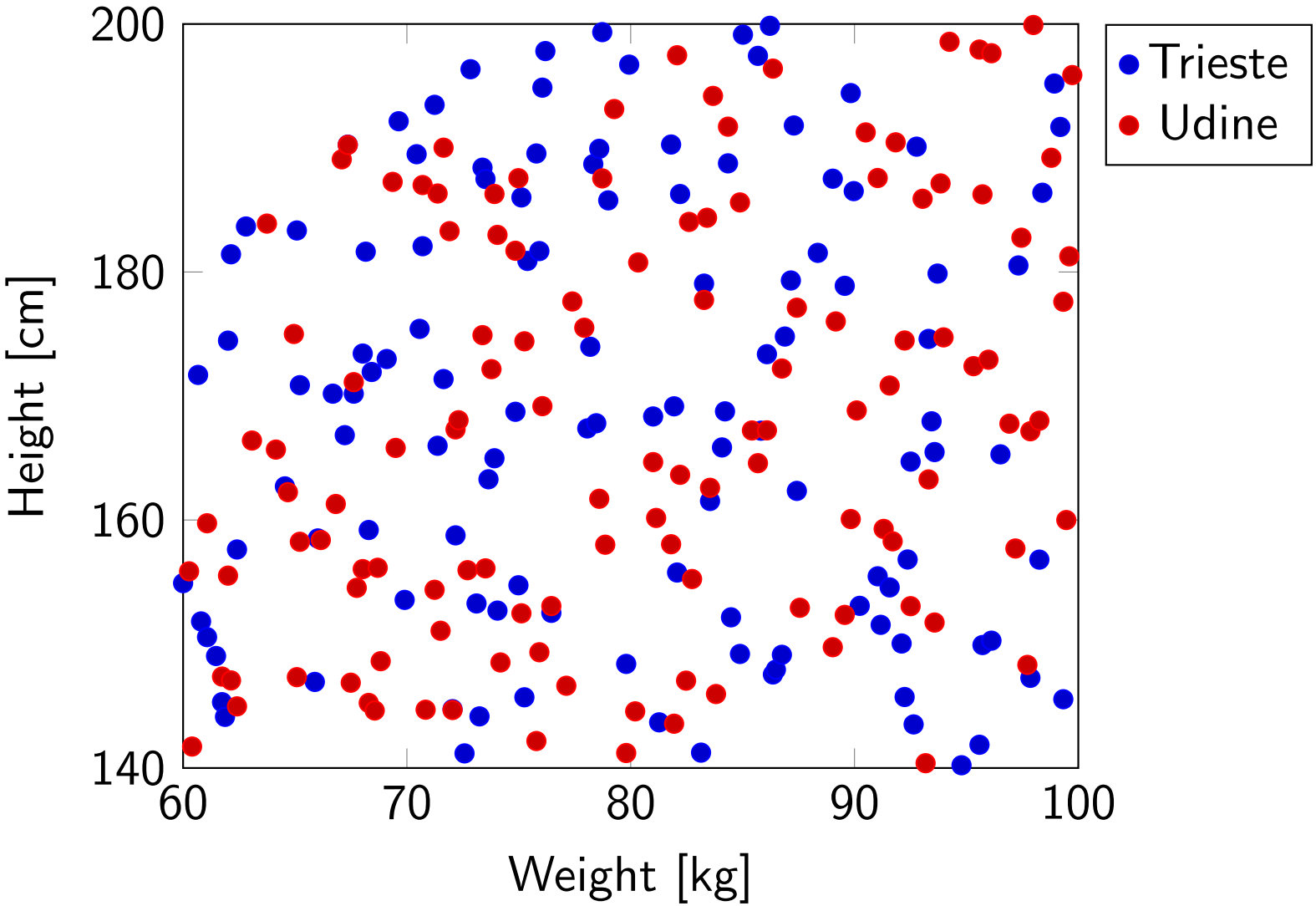
Why?

- the data is not informative
- the data is not representative
- the data has changed
- the data is too noisy

We will see/discuss these issues.

ML is not magic

Problem: find birth town from height/weight.



Q: which is the data issue here?

Implementation

When "solving" a problem, we usually need:

- explore/visualize data
- apply one or more ML technique
- assess learned models

"By hands?" No, with software!

ML/DM software

Many options:

- libraries for general purpose languages:
  - Java: e.g., `http://haifengl.github.io/smile/`
  - Python: e.g., `http://scikit-learn.org/stable/`
  - ...
- specialized sw environments:
  - Octave: `https://en.wikipedia.org/wiki/GNU_Octave`
  - R: `https://en.wikipedia.org/wiki/R_(programming_language)`
- from scratch

ML/DM software: which one?

- production/prototype
- platform constraints
- degree of (data) customization
- documentation availability/community size
- ...
- previous knowledge/skills

ML/DM software: why?

In all cases, sw allows to be more productive and concise.
E.g., learn and use a model for classification, in Java+Smile:

```
1 double[][] instances = ...;
2 int[] labels = ...;
3 RandomForest classifier = (new RandomForest.Trainer()).train(
      instances, labels);
4 double[] newInstance = ...;
5 int newLabel = classifier.predict(newInstance);
```

In R:

```
1 d = ...
2 classifier = randomForest(label~., d)
3 newD = ...
4 newLabels = predict(classifier, newD)
```

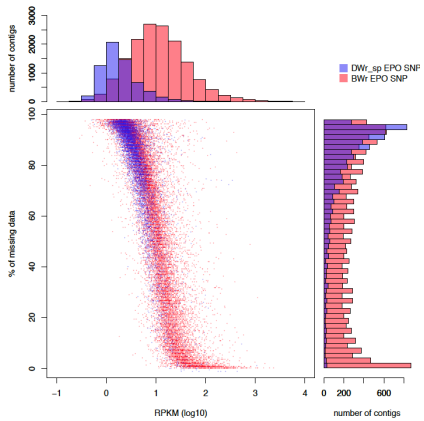
Section 3

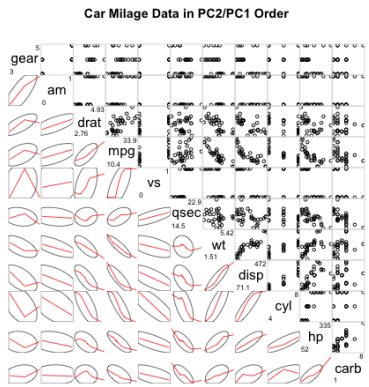
Plotting data: an overview

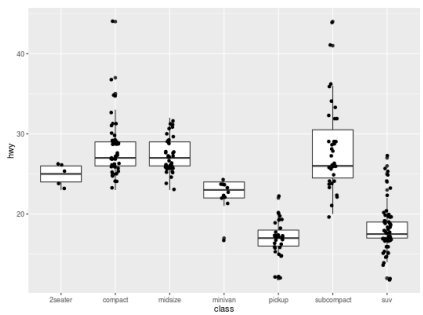
Advanced plotting

- many packages (e.g., ggplot2)
- many options

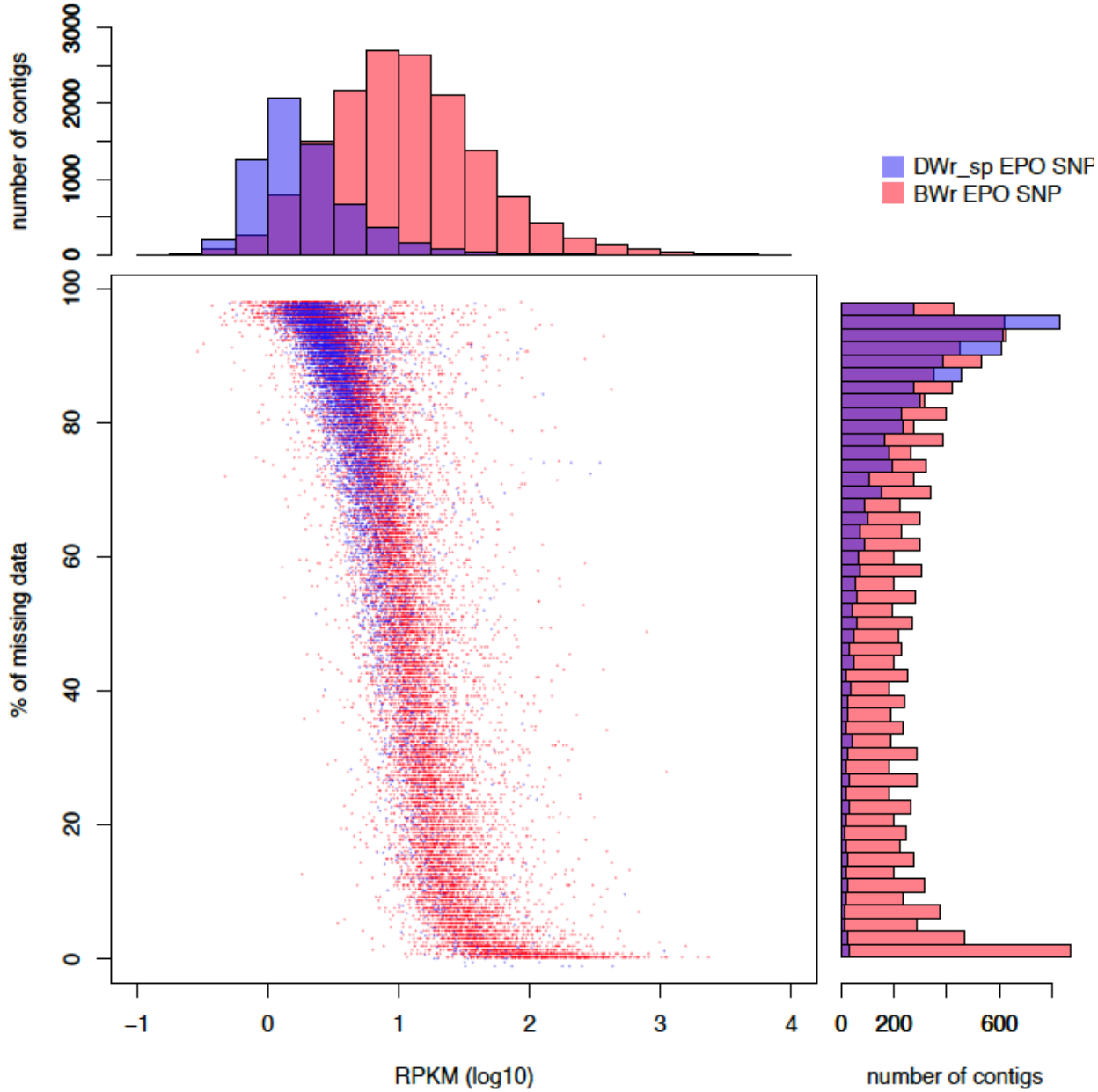
Which is the most proper chart to support a thesis?





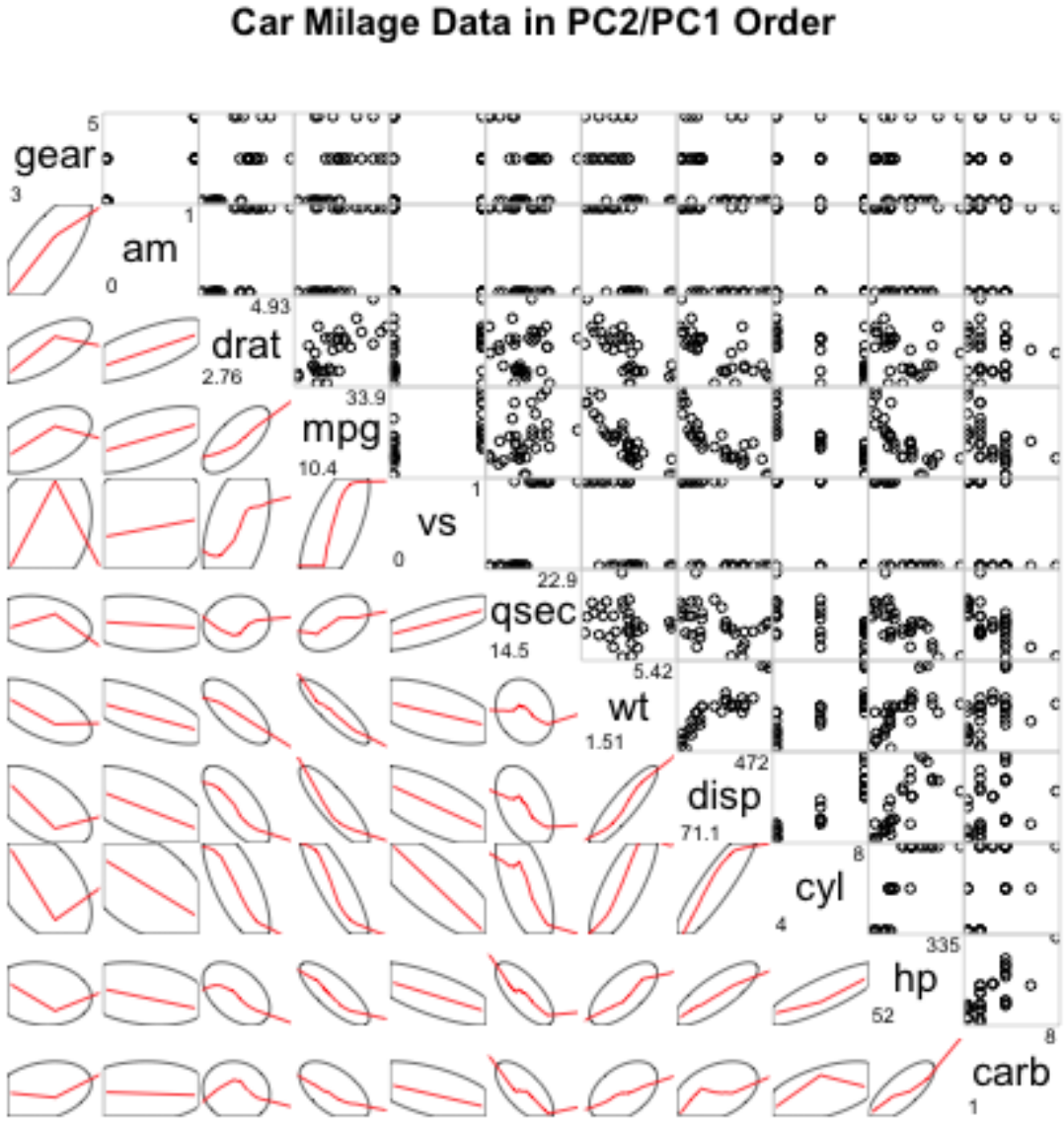


Aim of a plot: examples

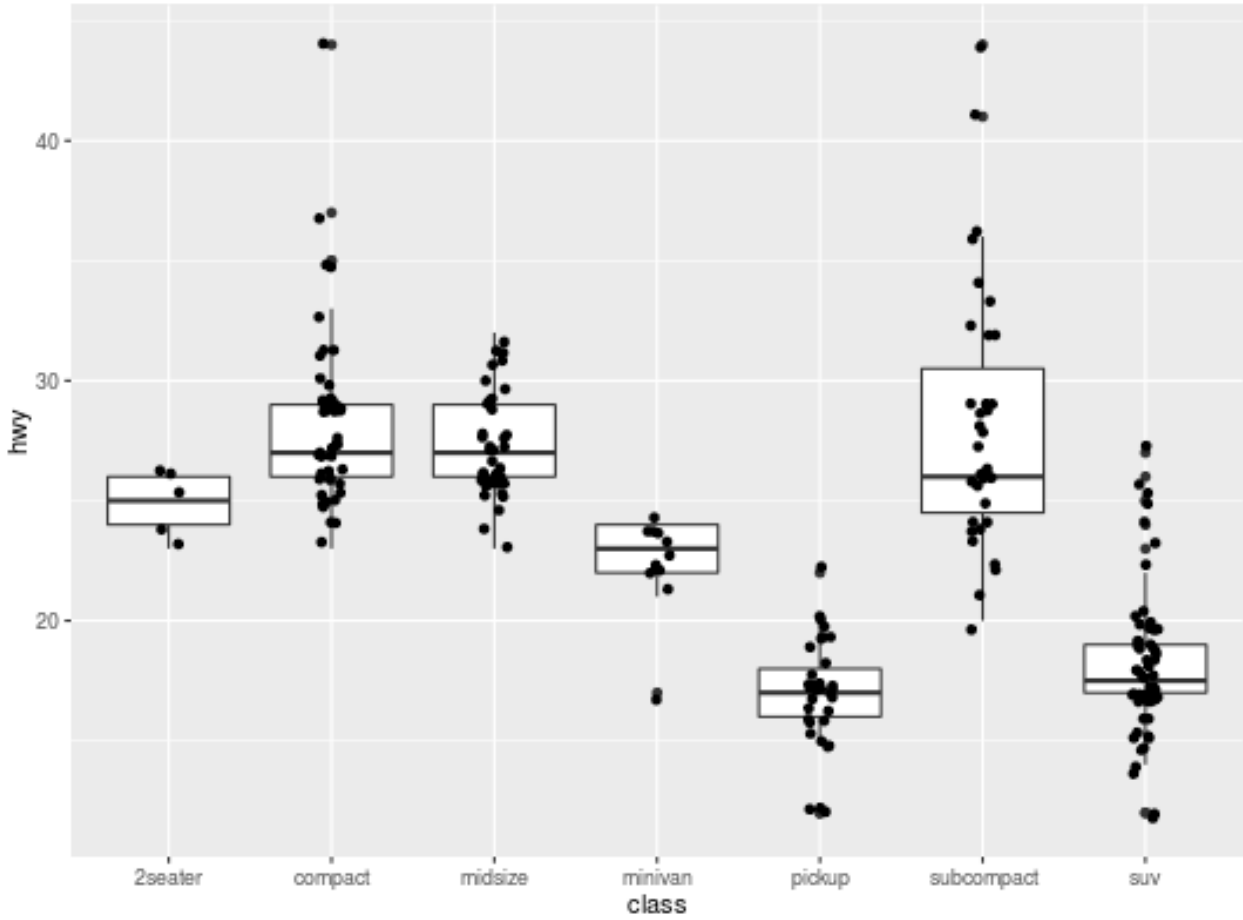


Aim of a plot: examples

Aim of a plot: examples



Aim of a plot: examples



Section 4

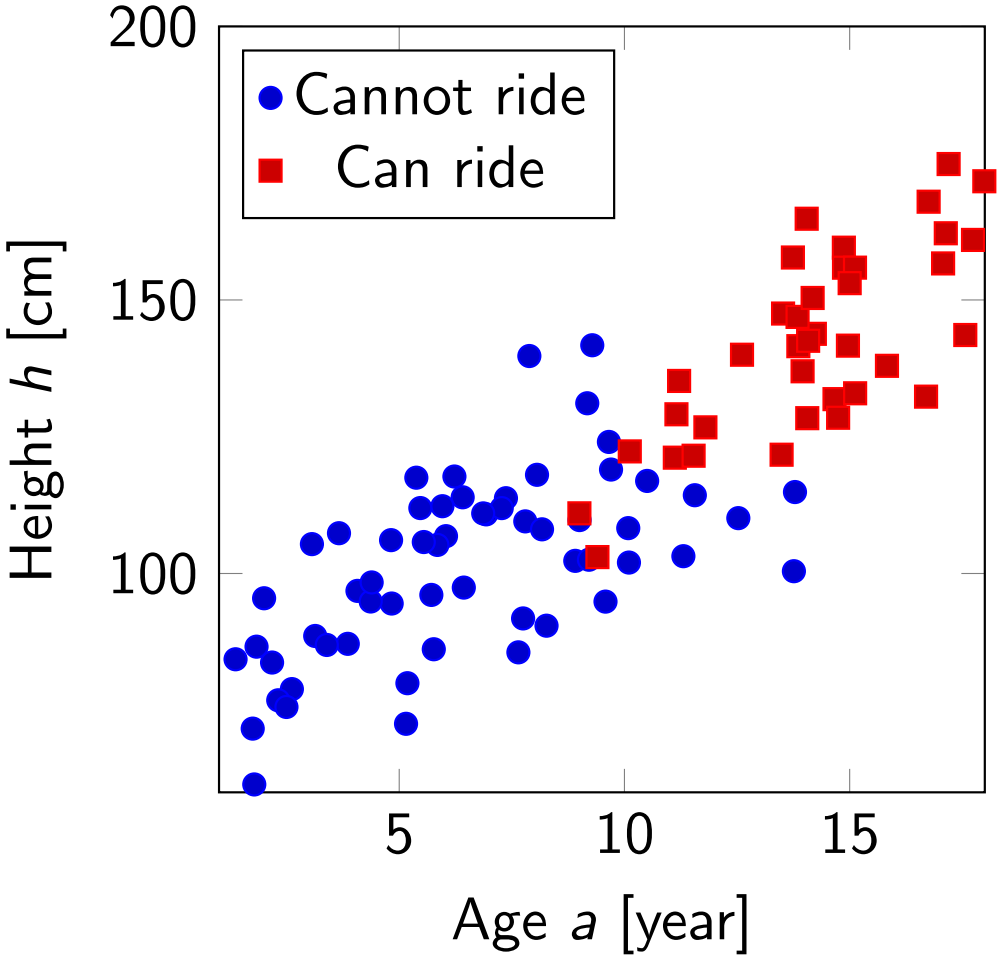
Tree-based methods

The carousel robot attendant

Problem: replace the carousel attendant with a robot which automatically decides who can ride the carousel.

Carousel: data

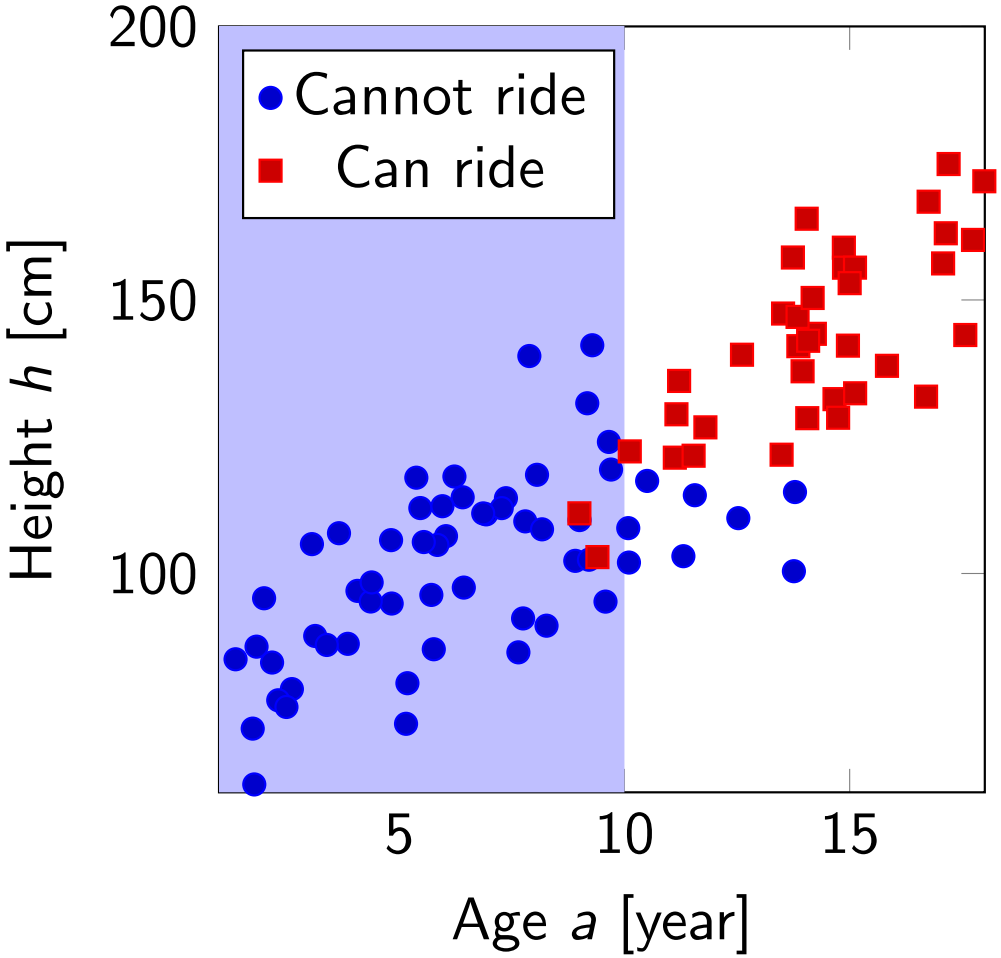
Observed human attendant's decisions.



How can the robot take the decision?

Carousel: data

Observed human attendant's decisions.

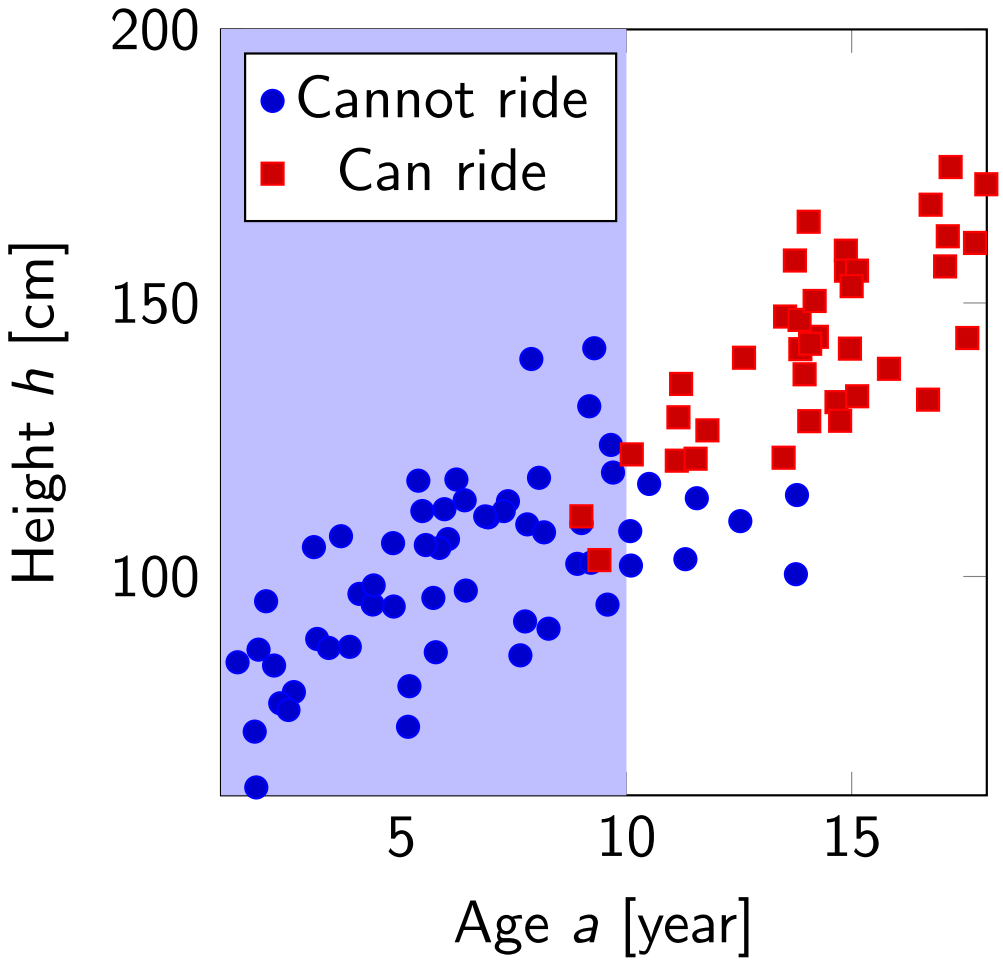


How can the robot take the decision?

- if younger than 10 $\rightarrow$ can't!

Carousel: data

Observed human attendant's decisions.

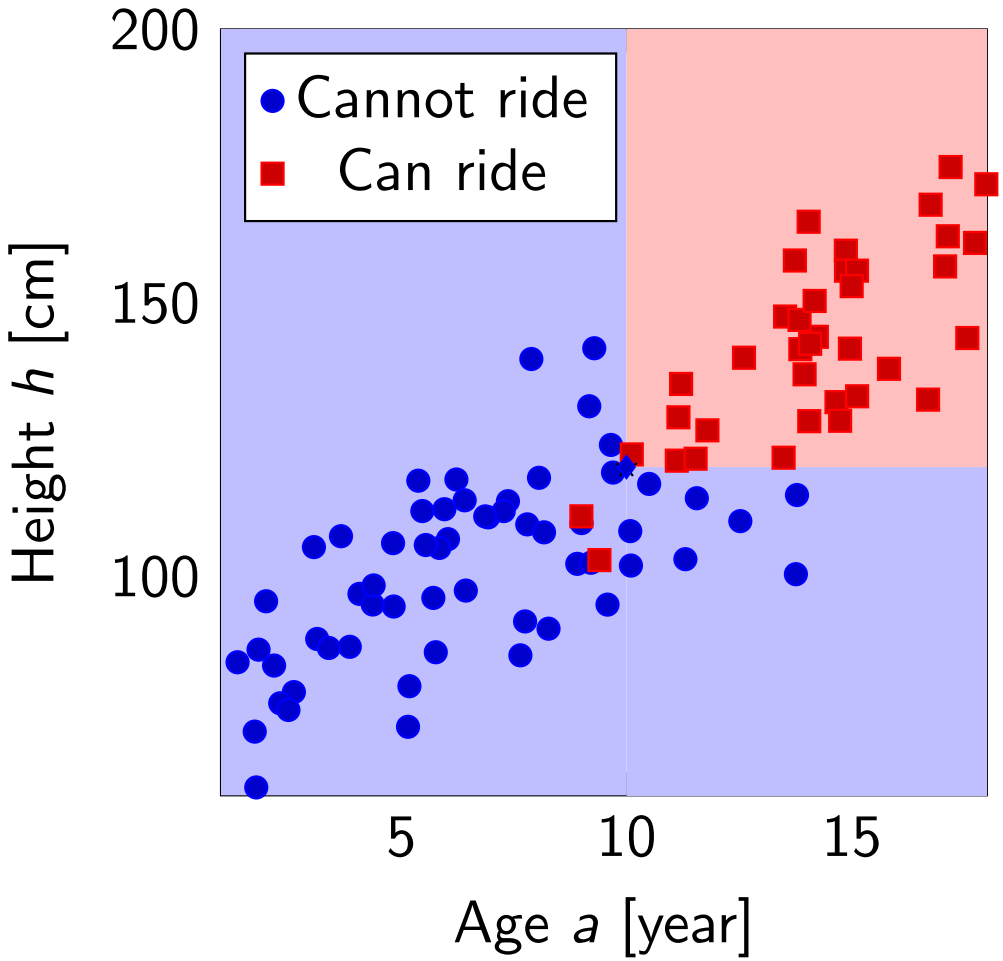


How can the robot take the decision?

- if younger than 10 $\rightarrow$ can't!
- otherwise:

Carousel: data

Observed human attendant's decisions.

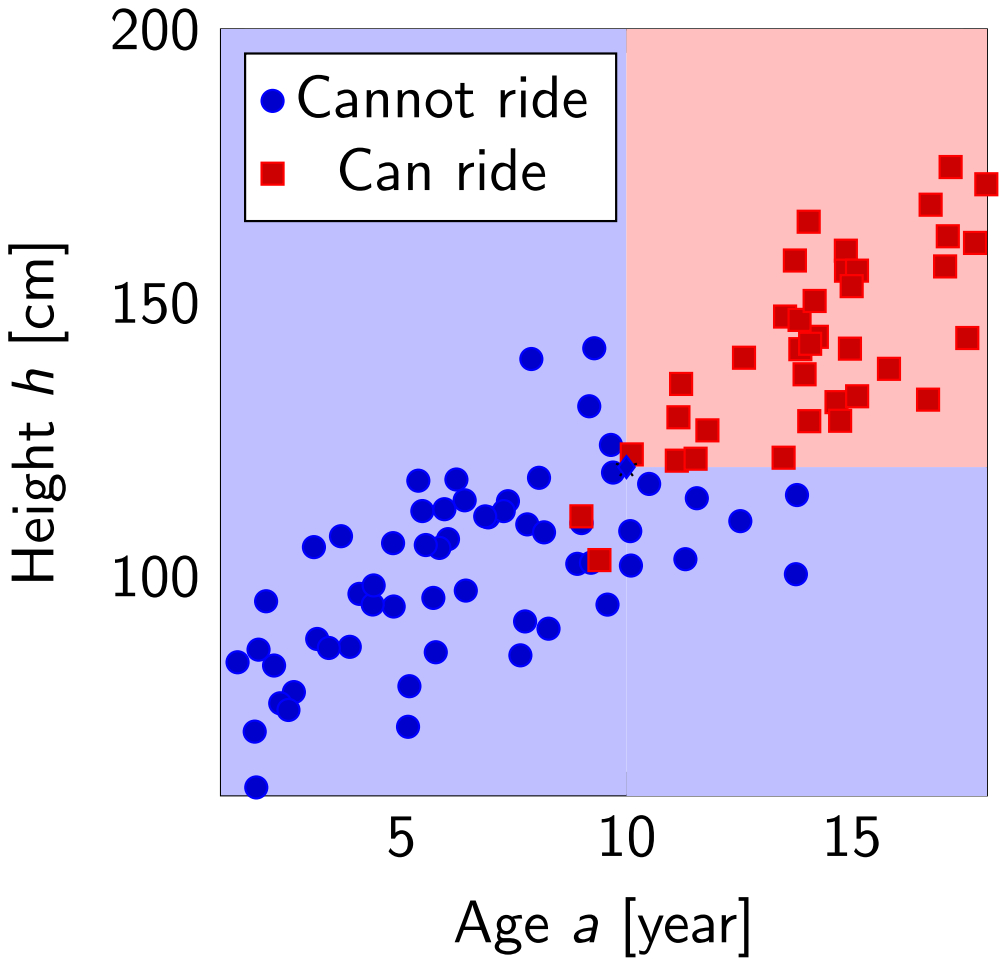


How can the robot take the decision?

- if younger than 10 $\rightarrow$ can't!
- otherwise:
  - if shorter than 120 $\rightarrow$ can't!
  - otherwise $\rightarrow$ can!

Carousel: data

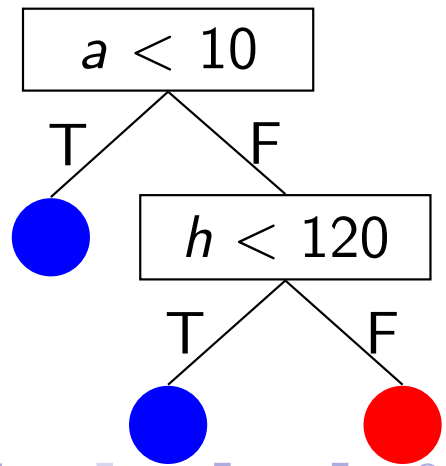
Observed human attendant's decisions.



How can the robot take the decision?

- if younger than 10 $\rightarrow$ can't!
- otherwise:
  - if shorter than 120 $\rightarrow$ can't!
  - otherwise $\rightarrow$ can!

Decision tree!



How to build a decision tree

Dividi-et-impera (recursively):

- find a cut variable and a cut value
- for left-branch, dividi-et-impera
- for right-branch, dividi-et-impera

How to build a decision tree: detail

Recursive binary splitting

function BUILDDECISIONTREE($\mathbf{X}, \mathbf{y}$)
  if SHOULDSTOP($\mathbf{y}$) **then**
    $\hat{y} \leftarrow$ most common class in $\mathbf{y}$
    return new terminal node with $\hat{y}$
  else
    $(i, t) \leftarrow$ BESTBRANCH($\mathbf{X}, \mathbf{y}$)
    $n \leftarrow$ new branch node with (i, t)
    append child BUILDDECISIONTREE($\mathbf{X}|_{\mathbf{x}_i < t}, \mathbf{y}|_{\mathbf{x}_i < t}$) to n
    append child BUILDDECISIONTREE($\mathbf{X}|_{\mathbf{x}_i \geq t}, \mathbf{y}|_{\mathbf{x}_i \geq t}$) to n
    return n
  end if
end function

- Recursive binary splitting
- Top down (start from the "big" problem)

Best branch

function BestBranch($\mathbf{X}, \mathbf{y}$)
  $(i^\star, t^\star) \leftarrow \arg\min_{i,t} E(\mathbf{y}|_{\mathbf{x}_i \geq t}) + E(\mathbf{y}|_{\mathbf{x}_i < t})$
  return $(i^\star, t^\star)$
end function

Classification error on subset:

$$E(\mathbf{y}) = \frac{|\{y \in \mathbf{y} : y \neq \hat{y}\}|}{|\mathbf{y}|}$$

$\hat{y} =$ the most common class in $\mathbf{y}$

- Greedy (choose split to minimize error now, not in later steps)

Best branch

$$(i^\star, t^\star) \leftarrow \arg\min_{i,t} E(\mathbf{y}|_{\mathbf{x}_i \geq t}) + E(\mathbf{y}|_{\mathbf{x}_i < t})$$

The formula say what is done, not how is done!

Q: "how" can different methods differ?

Stopping criterion

function ShouldStop(**y**)
    if **y** contains only one class **then**
        return true
    else if $|\mathbf{y}| < k_{\text{min}}$ **then**
        return true
    else
        return false
    end if
end function

Other possible criterion:

- tree depth larger than d_{max}

Best branch criteria

Classification error $E()$ works, but has been shown to be "not sufficiently sensitive for tree-growing".

$$E(\mathbf{y}) = \frac{|\{y \in \mathbf{y} : y \neq \hat{y}\}|}{|\mathbf{y}|} = 1 - \max_c \frac{|\{y \in \mathbf{y} : y = c\}|}{|\mathbf{y}|} = 1 - \max_c p_{\mathbf{y},c}$$

Other two option:

- Gini index

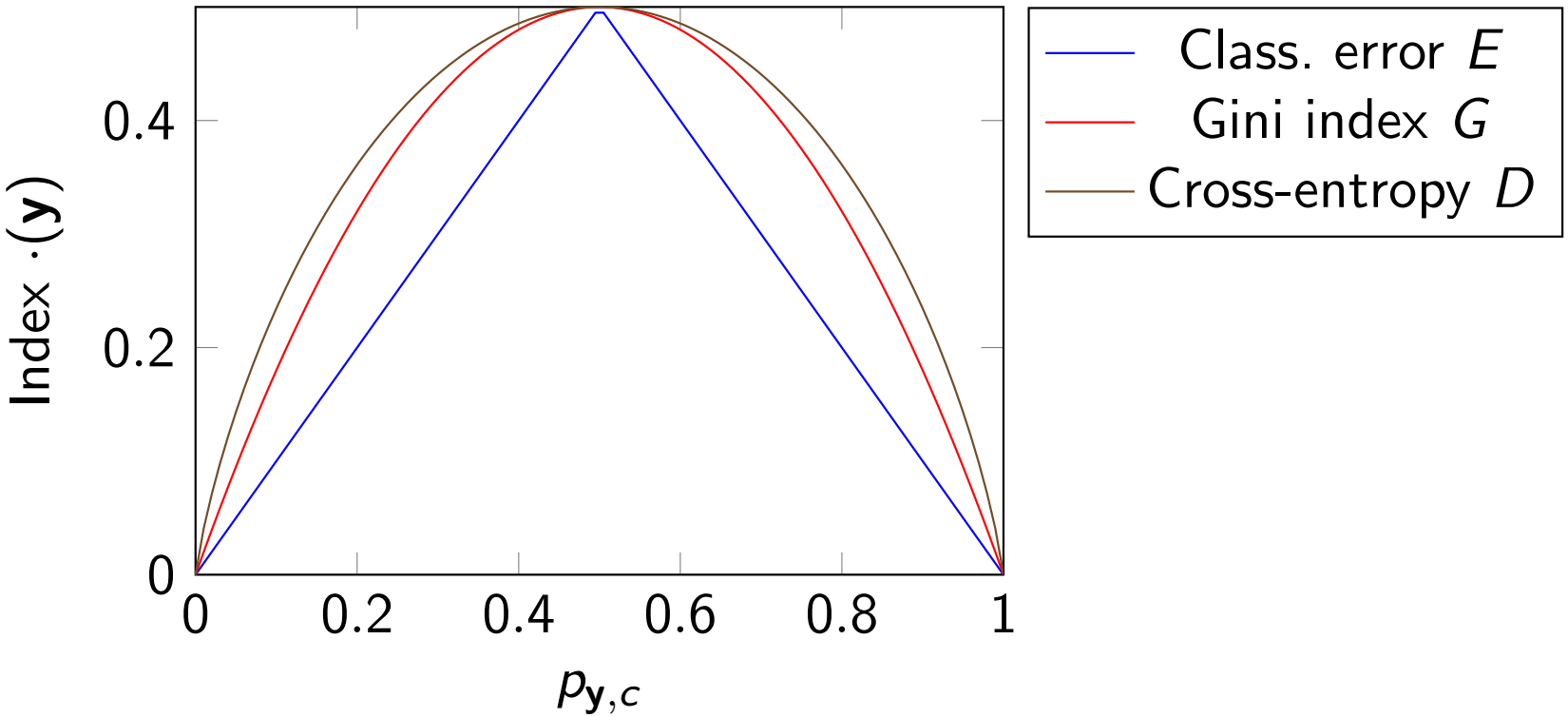
$$G(\mathbf{y}) = \sum_c p_{\mathbf{y},c}(1 - p_{\mathbf{y},c})$$

- Cross-entropy

$$D(\mathbf{y}) = -\sum_c p_{\mathbf{y},c} \log p_{\mathbf{y},c}$$

For all indexes, the lower the better (**node impurity**).

Best branch criteria: binary classification



Cross-entropy is rescaled.

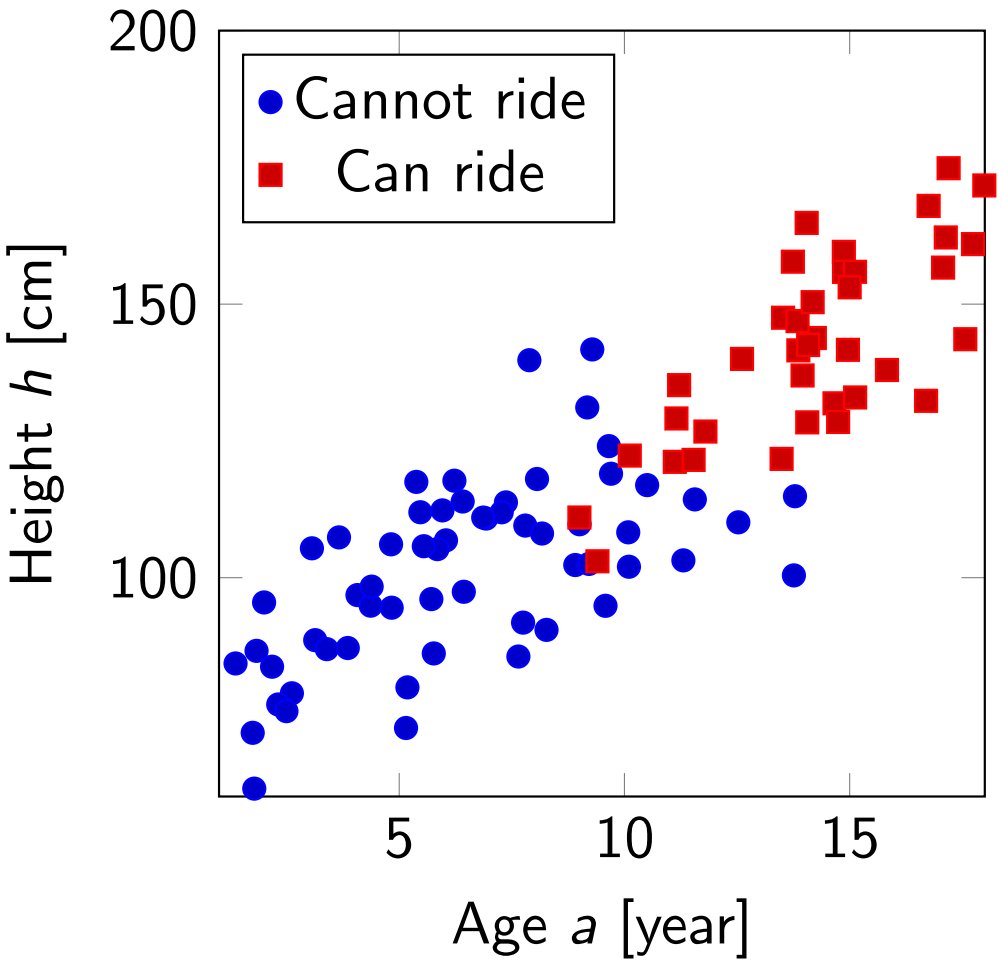
Q: what happens with multiclass problems?

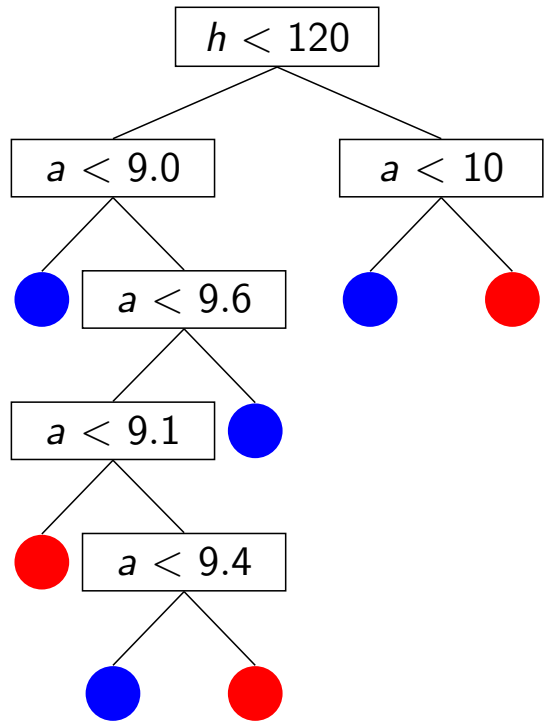
Categorical independent variables

- Trees can work with categorical variables
- Branch node is $x_i = c$ or $x_i \in C' \subset C$ (c is a class)
- Can mix categorical and numeric variables

Stopping criterion: role of $k_{\min}$

Suppose $k_{\min} = 1$ (never stop for $\mathbf{y}$ size)





Q: what's wrong? (recall: "a model of what?")

Tree complexity

When the tree is "too complex"

- less readable/understandable/explicable
- maybe there was noise into the data

Q: what's noise in carousel data?

Tree complexity is not related (only) with k_{min}, but also with data

Tree complexity: other interpretation

- maybe there was noise into the data

The tree *fits* the learning data too much:

- it overfits (**overfitting**)
- does not generalize (high **variance**: model varies if learning data varies)

High variance

"model varies if learning data varies": what? why data varies?

- learning data is about the system/phenomenon/nature S
  - a collection of *observations* of S
  - a point of view on S

High variance

"model varies if learning data varies": what? why data varies?

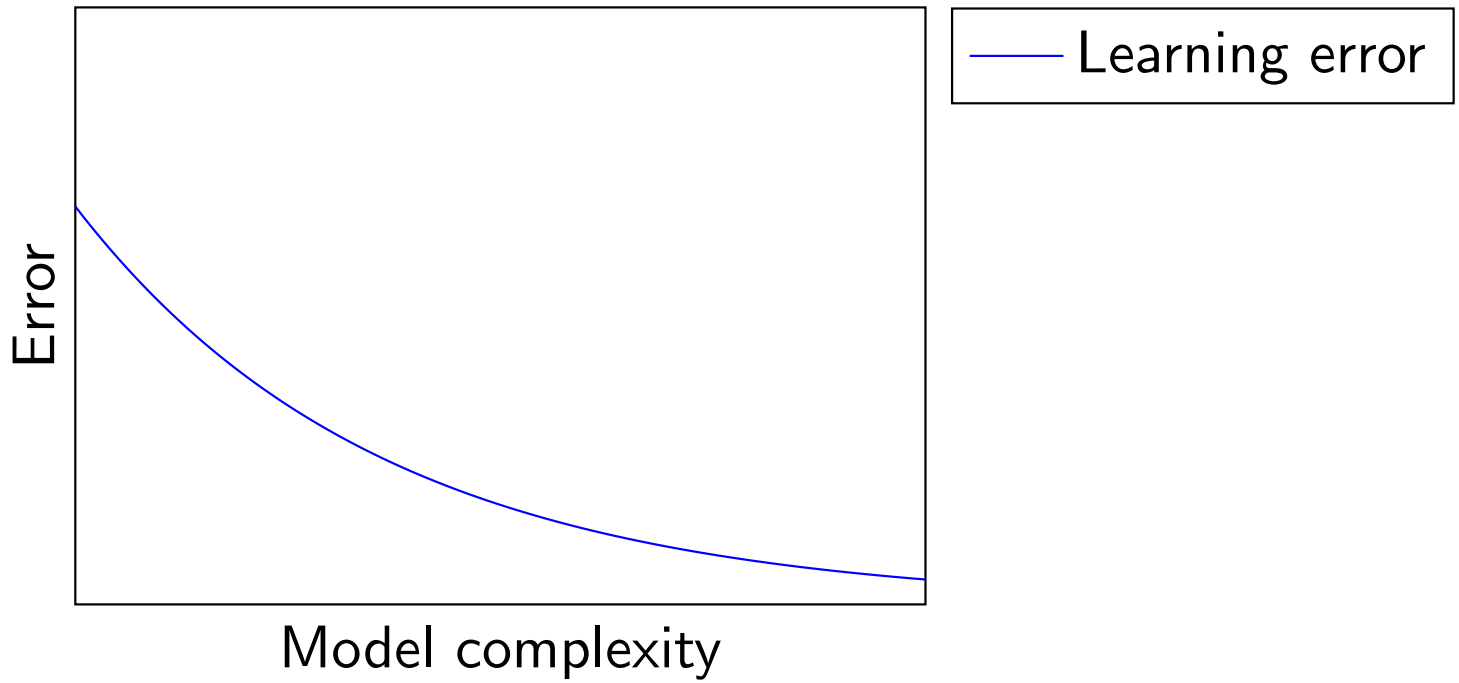
- learning data is about the system/phenomenon/nature S
  - a collection of *observations* of S
  - a point of view on S
- learning is about understanding/knowing/explaining S

High variance

"model varies if learning data varies": what? why data varies?

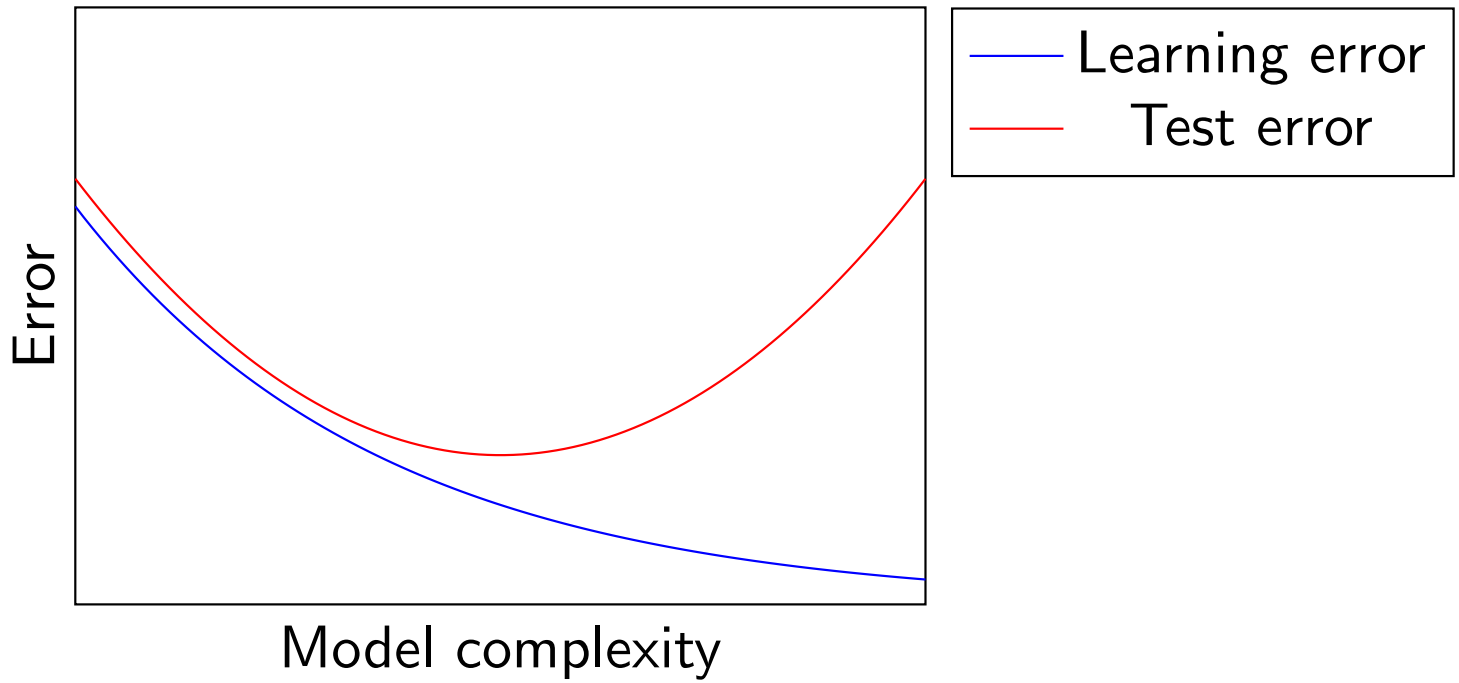
- learning data is about the system/phenomenon/nature S
  - a collection of *observations* of S
  - a point of view on S
- learning is about understanding/knowing/explaining S
  - if I change the point of view on S, my knowledge about S should remain the same!

Spotting overfitting



Test error: error on unseen data

Spotting overfitting



Test error: error on unseen data

k-fold cross-validation

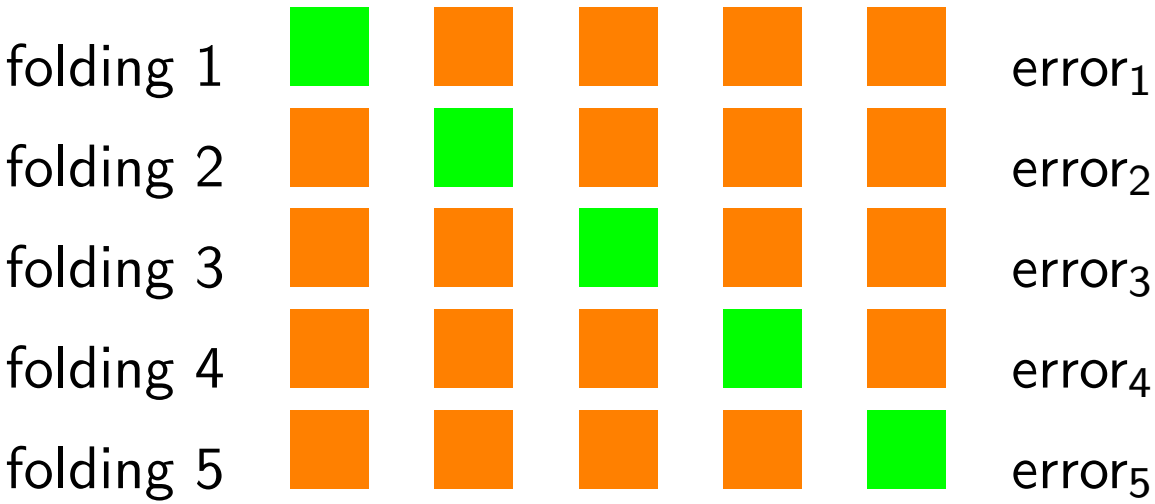
Where can I find "unseen data"? Pretend to have it!

1. split learning data ($\mathbf{X}$ and $\mathbf{y}$) in k equal slices (each of $\frac{n}{k}$ observations/elements)
2. for each split (i.e., each $i \in \{1, \ldots, k\}$)
   - 2.1 learn on all but k-th slice
   - 2.2 compute classification error on **unseen** k-th slice
3. average the k classification errors

In essence:

- can the learner generalize beyond available data?
- how the learned artifact will behave on unseen data?

k-fold cross-validation



$$\text{error} = \frac{1}{k} \sum_{i=1}^{i=k} \text{error}_i$$

Or with any other meaningful (effectiveness) measure

Q: how should data be split?

Fighting overfitting with trees

- large k_{min} (large w.r.t. what?)
- when building, limit depth
- when building, don't split if low overall impurity decrease
- after building, *prune*

Pruning: high level idea

1. learn a full tree t_0
2. build from t_0 a sequence $T = \{t_0, t_1, \ldots, t_n\}$ of trees such that
   - t_i is a root-subtree of t_{i-1} ($t_i \subset t_{i-1}$)
   - t_i is always less complex than t_{i-1}
3. choose the $t \in T$ with minimum classification error with k-fold cross-validation

k-fold cross-validation: data splitting

Q: how should data be split?

Example: Android Malware detection

- Gerardo Canfora et al. "Effectiveness of opcode ngrams for detection of multi family android malware". In: *Availability, Reliability and Security (ARES), 2015 10th International Conference on*. IEEE. 2015, pp. 333–340
- Gerardo Canfora et al. "Detecting android malware using sequences of system calls". In: *Proceedings of the 3rd International Workshop on Software Development Lifecycle for Mobile*. ACM. 2015, pp. 13–20

Using cross-validation (CV) for assessment (I)

How the learned artifact will behave on unseen data?

More precisely:
How an artifact learned with **this learning technique** will behave on unseen data?

Using CV for assessment (II)

"This learning technique" $=$ BuildDecisionTree() with $k_{\text{min}} = 10$

1. repeat k times
   1.1 BuildDecisionTree() with $k_{\text{min}} = 10$ on all but one slice
   - $\frac{k-1}{k} n$ observations in each $\mathbf{X}$ passed to BuildDecisionTree()
   1.2 compute classification error on left out slice
2. average computed classification errors

k invocations of BuildDecisionTree()

Using CV for assessment (III)

"This learning technique" = BuildDecisionTree() with $k_{\min}$ chosen automatically with a 10-fold CV

For assessing this technique, we do two nested CVs:

1. repeat k times
   - 1.1 choose $k_{\min}$ among m values with 10-CV (repeat BuildDecisionTree() $10m$ times) on all but one slice
     - $\frac{k-1}{k}\frac{9}{10}n$ observations in each $\mathbf{X}$ passed to BuildDecisionTree()!
   - 1.2 compute classification error on left out slice
     - usually, a new tree is built on $\frac{k-1}{k}n$ observations
2. average computed classification errors

$(10m+1)k$ invocations of BuildDecisionTree()

Using CV for assessment: "cheating"

"This learning technique" $=$ BuildDecisionTree() with k_{min} chosen automatically with a 10-fold CV

Using just one CV is cheating (cherry picking)!

- k_{min} is chosen exactly to minimize error on the full dataset
- conceptually, this way of "fitting" k_{min} is similar to the way we build the tree

Subsection 1

Regression trees

Regression with trees

Trees can be used for regression, instead of classification.

decision tree vs. regression tree

Tree building: decision → regression

function BUILDDECISIONTREE($\mathbf{X}, \mathbf{y}$)
  if SHOULDSTOP($\mathbf{y}$) **then**
    $\hat{y} \leftarrow$ most common class in $\mathbf{y}$
    return new terminal node with $\hat{y}$
  else
    $(i, t) \leftarrow$ BESTBRANCH($\mathbf{X}, \mathbf{y}$)
    $n \leftarrow$ new branch node with (i, t)
    append child BUILDDECISIONTREE($\mathbf{X}|_{\mathbf{x}_i < t}, \mathbf{y}|_{\mathbf{x}_i < t}$) to n
    append child BUILDDECISIONTREE($\mathbf{X}|_{\mathbf{x}_i \geq t}, \mathbf{y}|_{\mathbf{x}_i \geq t}$) to n
    return n
  end if
end function

Q: what should we change?

Tree building: decision → regression

function BUILDDECISIONTREE($\mathbf{X}, \mathbf{y}$)
  if SHOULDSTOP($\mathbf{y}$) **then**
    $\hat{y} \leftarrow \bar{y}$ ▷ mean $\mathbf{y}$
    return new terminal node with $\hat{y}$
  else
    $(i, t) \leftarrow$ BESTBRANCH($\mathbf{X}, \mathbf{y}$)
    $n \leftarrow$ new branch node with (i, t)
    append child BUILDDECISIONTREE($\mathbf{X}|_{\mathbf{x}_i < t}, \mathbf{y}|_{\mathbf{x}_i < t}$) to n
    append child BUILDDECISIONTREE($\mathbf{X}|_{\mathbf{x}_i \geq t}, \mathbf{y}|_{\mathbf{x}_i \geq t}$) to n
    return n
  end if
end function

Q: what should we change?

Best branch

function BestBranch($\mathbf{X}, \mathbf{y}$)
  $(i^\star, t^\star) \leftarrow \arg\min_{i,t} E(\mathbf{y}|_{\mathbf{x}_i \geq t}) + E(\mathbf{y}|_{\mathbf{x}_i < t})$
  return $(i^\star, t^\star)$
end function

Q: what should we change?

Best branch

function BESTBRANCH($\mathbf{X}, \mathbf{y}$)
  $(i^\star, t^\star) \leftarrow \arg\min_{i,t} \sum_{y_i \in \mathbf{y}|_{\mathbf{x}_i \geq t}} (y_i - \bar{y})^2 + \sum_{y_i \in \mathbf{y}|_{\mathbf{x}_i < t}} (y_i - \bar{y})^2$
  return $(i^\star, t^\star)$
end function

Q: what should we change?

Minimize sum of residual sum of squares (RSS) (the two $\bar{y}$ are different)

Stopping criterion

function ShouldStop(**y**)
    if **y** contains only one class **then**
        return true
    else if $|\mathbf{y}| < k_{\text{min}}$ **then**
        return true
    else
        return false
    end if
end function

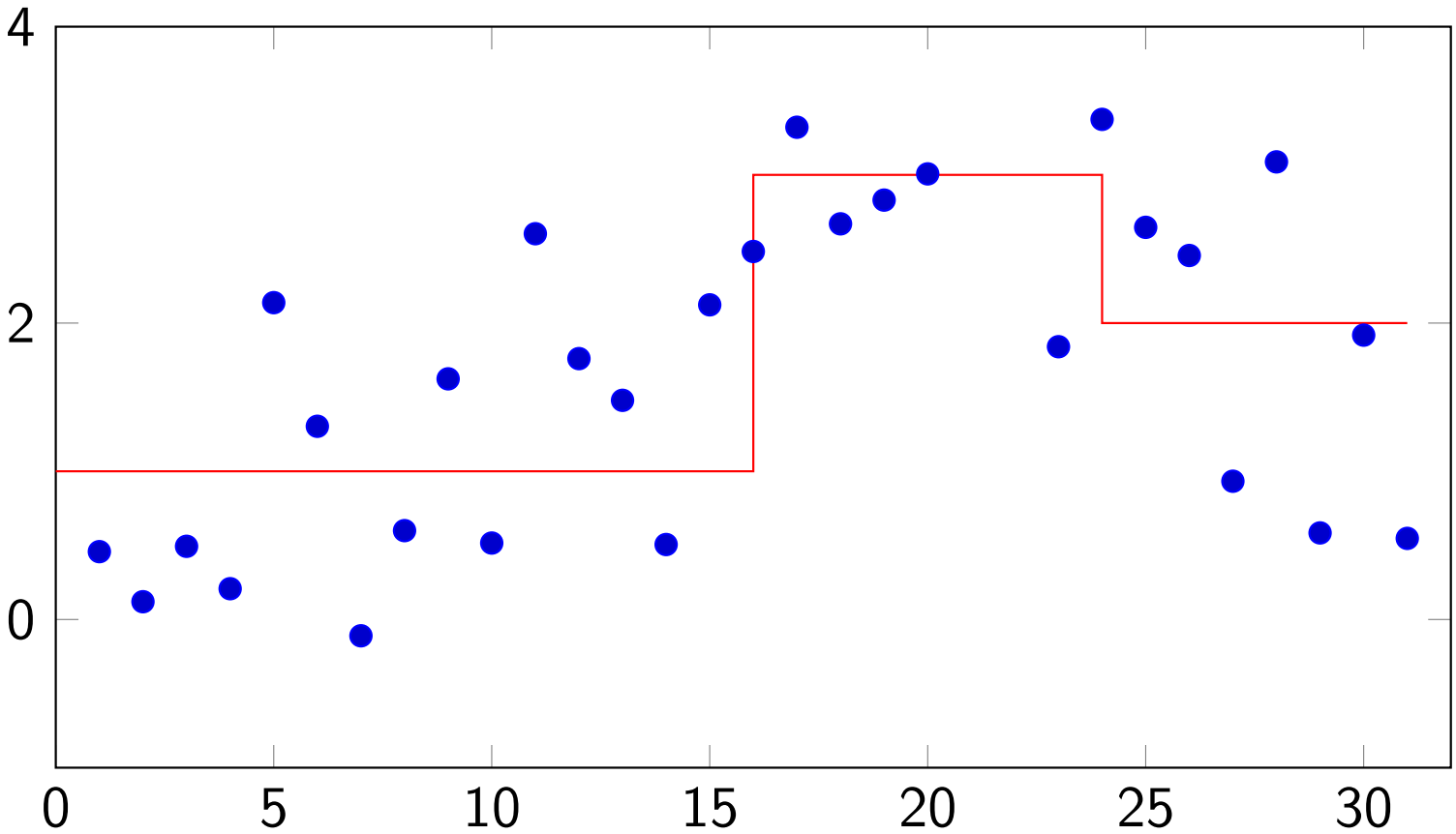
Q: what should we change?

Stopping criterion

function SHOULDSTOP(**y**)
    if RSS is 0 **then**
        return true
    else if $|\mathbf{y}| < k_{\min}$ **then**
        return true
    else
        return false
    end if
end function

Q: what should we change?

Interpretation



Regression and overfitting

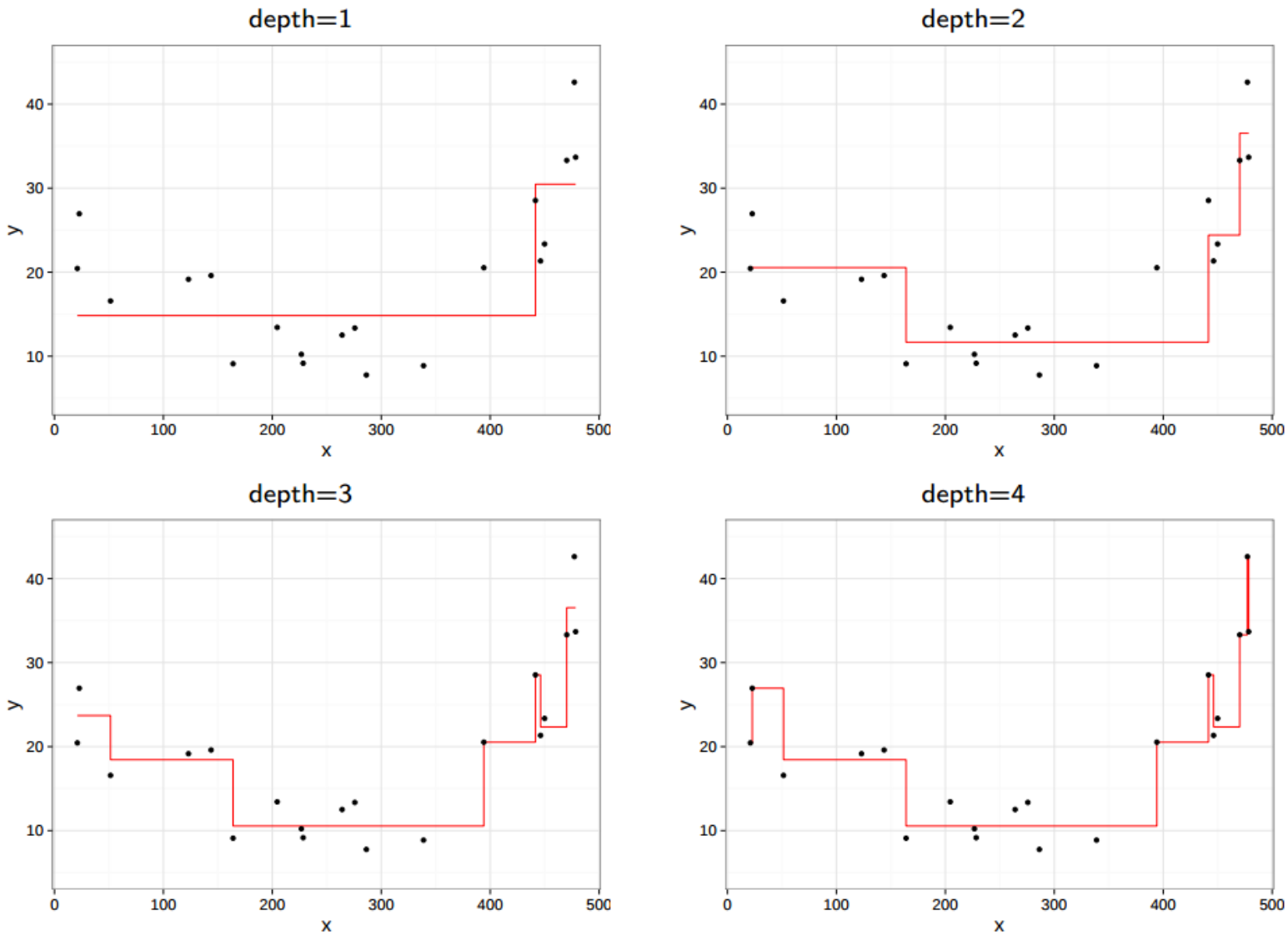


Image from F. Daolio

Trees in summary

Pros:

- ▲ easily interpretable/explicable
- ▲ learning and regression/classification easily understandable
- ▲ can handle both numeric and categorical values

Cons:

- ▼ not so accurate (**Q:** always?)

Tree accuracy?

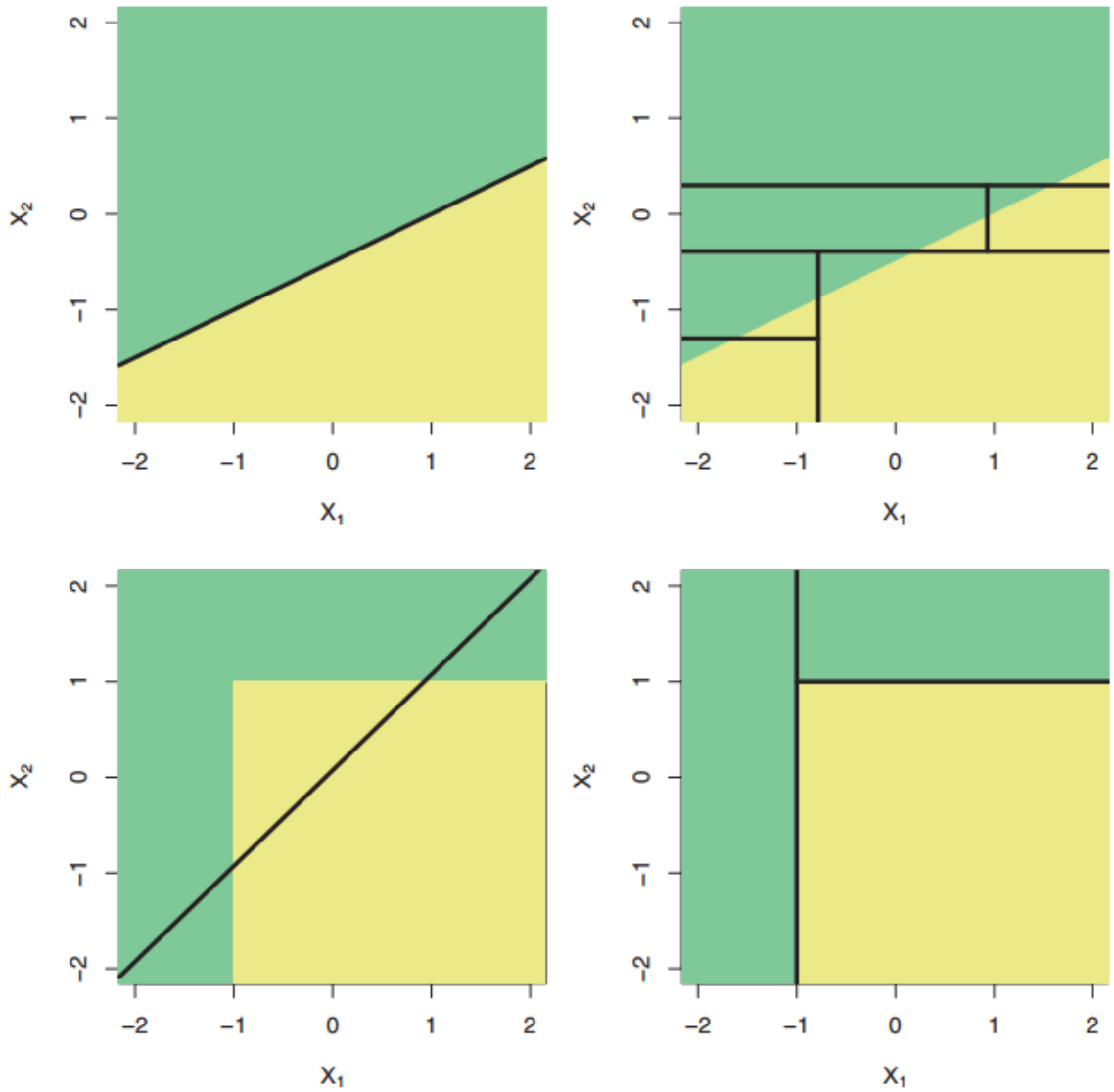
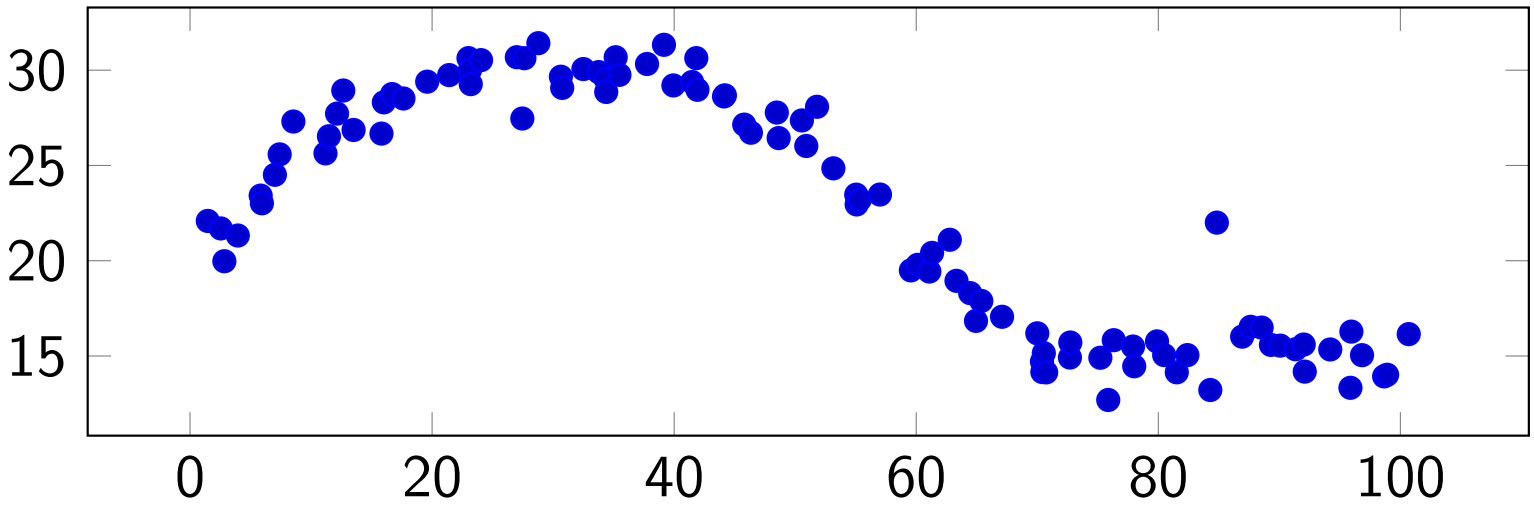


Image from An Introduction to Statistical Learning

Subsection 2

Trees aggregation

Weakness of the tree



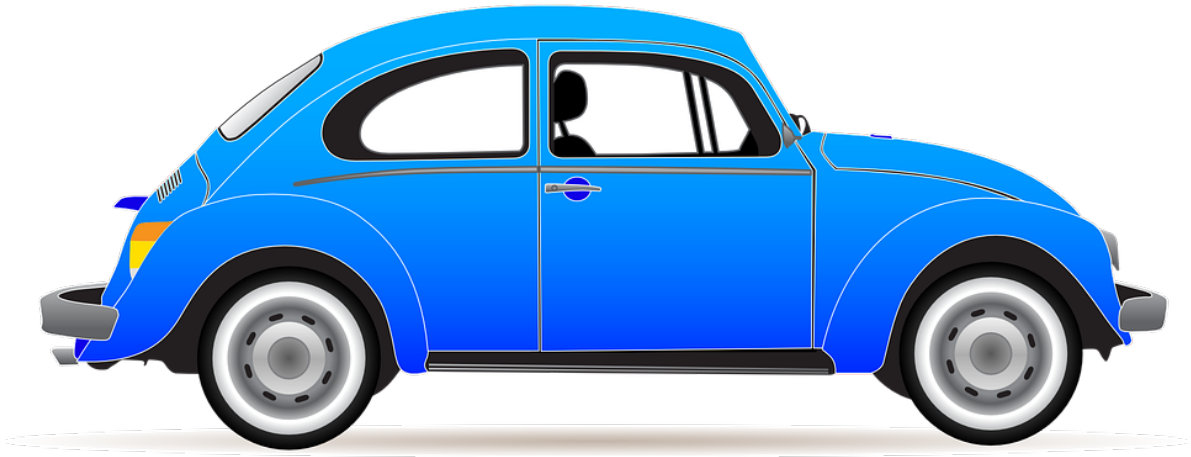
Small tree:

- low complexity
- will hardly fit the "curve" part
- *high bias, low variance*

Big tree:

- high complexity
- may overfit the noise on the right part
- *low bias, high variance*

The trees view



Small tree:

- "a car is something that moves"

Big tree:

- "a car is a made-in-Germany blue object with 4 wheels, 2 doors, chromed fenders, curved rear enclosing engine"

Big tree view

A big tree:

- has a detailed view of the learning data (high complexity)
- "trusts too much" the learning data (high variance)

What if we "combine" different big tree views and ignore details on which they disagree?

Wisdom of the crowds

What if we "combine" different big tree views and ignore details on which they disagree?

- many views
- independent views
- aggregation of views

$\approx$ *the wisdom of the crowds*: a collective opinion may be better than a single expert's opinion

Wisdom of the trees

- many views
- independent views
- aggregation of views

Wisdom of the trees

- many views
  - just use many trees
- independent views

- aggregation of views

Wisdom of the trees

- many views
  - just use many trees
- independent views

- aggregation of views
  - just average prediction (regression) or take most common prediction (classification)

Wisdom of the trees

- many views
  - just use many trees
- independent views
  - ??? learning is deterministic: same data $\Rightarrow$ same tree $\Rightarrow$ same view
- aggregation of views
  - just average prediction (regression) or take most common prediction (classification)

Independent views

Independent views $\equiv$ different points of view $\equiv$ *different* learning data

But we have only *one* learning data!

Independent views: idea! (**Bootstrap**)

Like in cross-fold, consider only a part of the data, but:

- instead of a subset
- a sample with repetitions

Independent views: idea! (**Bootstrap**)

Like in cross-fold, consider only a part of the data, but:

- instead of a subset
- a sample with repetitions

$$\mathbf{X} = (x_1^T\, x_2^T\, x_3^T\, x_4^T\, x_5^T) \qquad \text{original learning data}$$
$$\mathbf{X_1} = (x_1^T\, x_5^T\, x_3^T\, x_2^T\, x_5^T) \qquad \text{sample 1}$$
$$\mathbf{X_2} = (x_4^T\, x_2^T\, x_3^T\, x_1^T\, x_1^T) \qquad \text{sample 2}$$
$$\mathbf{X_i} = \ldots \qquad \text{sample } i$$

- (**y** omitted for brevity)
- learning data size is not a limitation (differently than with subset)

Tree **bagging**

When learning:

1. Repeat B times
   1.1 take a sample of the learning data
   1.2 learn a tree (unpruned)

When predicting:

1. Repeat B times
   1.1 get a prediction from ith learned tree
2. predict the average (or most common) prediction

For classification, other aggregations can be done: majority voting (most common) is the simplest
Using independent, possibly different classifiers together: *ensemble* of classifiers

How many trees?

B is a parameter:

- when there is a parameter, there is the problem of finding a good value
- remember $k_{\min}$, depth (**Q:** impact on?)

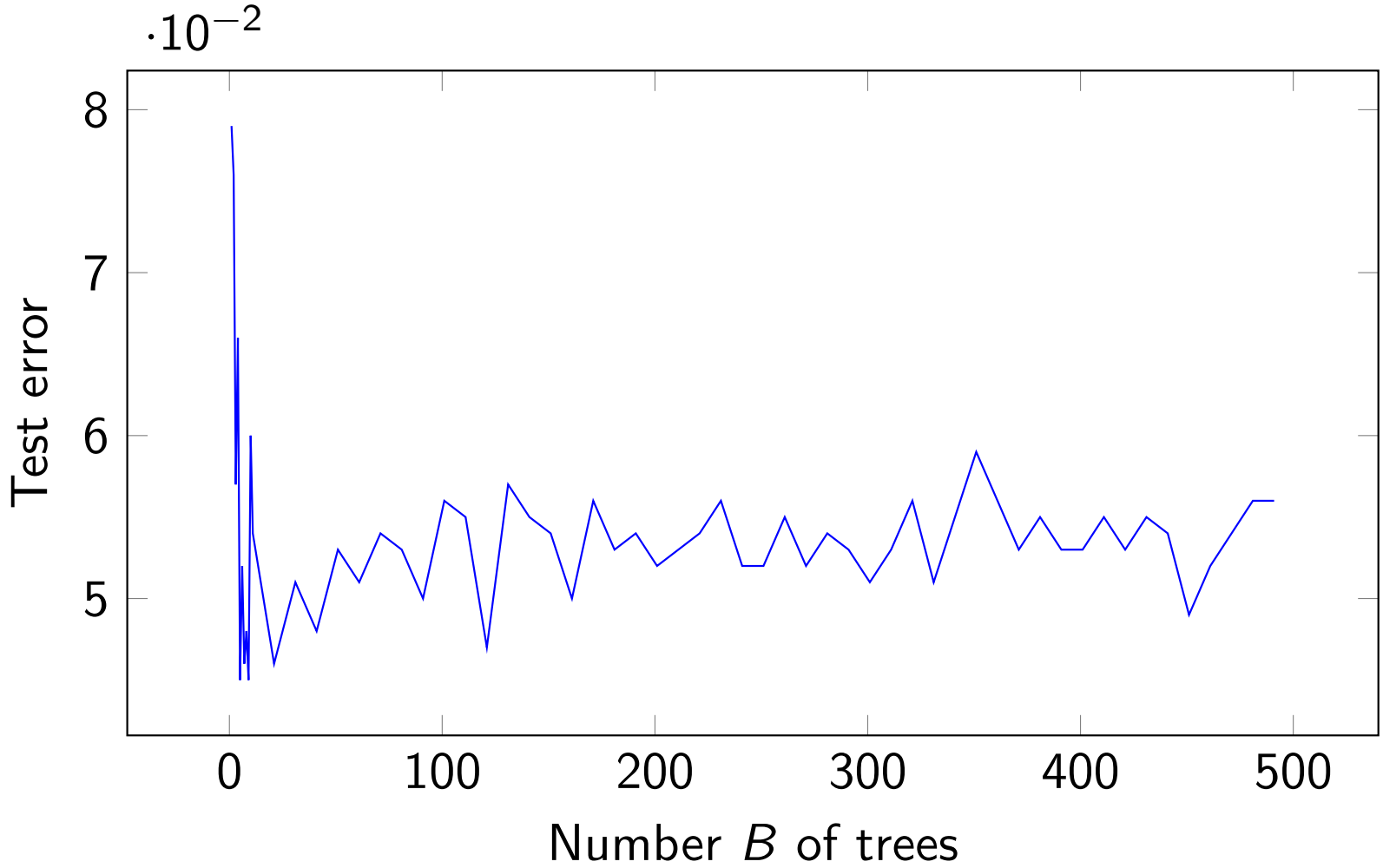
How many trees?

B is a parameter:

- when there is a parameter, there is the problem of finding a good value
- remember k_{min}, depth (**Q:** impact on?)
- it has been shown (experimentally) that
  - for "large" B, bagging is better than single tree
  - increasing B does not cause overfitting
  - (for us: default B is ok! "large" $\approx$ hundreds)

Q: how better? at which cost?

Bagging: impact of B



Independent view: improvement

Despite being learned on different samples, bagging trees may be correlated, hence views are not very independent

- e.g., one variable is much more important than others for predicting (*strong predictor*)

Idea: force point of view differentiation by "hiding" variables

Random forest

When learning:

1. Repeat B times
   - 1.1 take a sample of the learning data
   - 1.2 consider only m on p independent variables
   - 1.3 learn a tree (unpruned)

When predicting:

1. Repeat B times
   - 1.1 get a prediction from ith learned tree
2. predict the average (or most common) prediction

- (observations and) variables are **random**ly chosen...
- ...to learn a **forest** of trees

Q: are missing variables a problem?

Random forest: parameter m

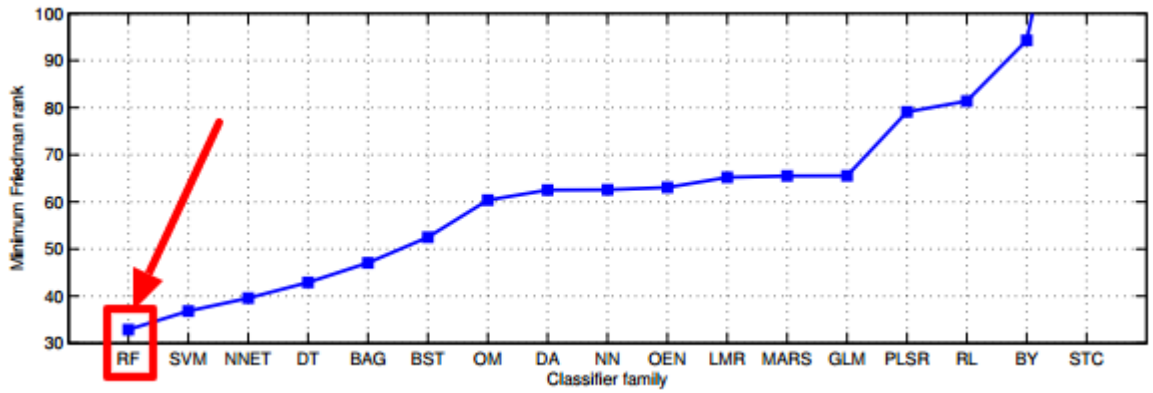
How to choose the value for m?

- $m = p \rightarrow$ bagging
- it has been shown (experimentally) that
  - m does not relate with overfitting
  - $m = \sqrt{p}$ is good for classification
  - $m = \frac{p}{3}$ is good for regression
  - (for us, default m is ok!)

Random forest

Experimentally shown: one of the "best" multi-purpose supervised classification methods

- Manuel Fernández-Delgado et al. "Do we need hundreds of classifiers to solve real world classification problems". In: *J. Mach. Learn. Res* 15.1 (2014), pp. 3133–3181



but...

No free lunch!

"Any two optimization algorithms are equivalent when their performance is averaged across all possible problems"

- David H Wolpert. "The lack of a priori distinctions between learning algorithms". In: *Neural computation* 8.7 (1996), pp. 1341–1390

Why free lunch?

- many restaurants, many items on menus, many possibly prices for each item: where to go to eat?
- no general answer
- but, if you are a vegan, or like pizza, then a best choice could exist

Q: problem? algorithm?

Observation sampling

When learning:

1. Repeat B times
   - 1.1 take a sample of the learning data
   - 1.2 consider only m on p independent variables (only for RF)
   - 1.3 learn a tree (unpruned)

Each learned tree uses only a portion of the observation in the learning data:

- for each observation, $\approx \frac{B}{3}$ trees did not considere it when learned

Observation sampling

When learning:

1. Repeat B times
   - 1.1 take a sample of the learning data
   - 1.2 consider only m on p independent variables (only for RF)
   - 1.3 learn a tree (unpruned)

Each learned tree uses only a portion of the observation in the learning data:

- for each observation, $\approx \frac{B}{3}$ trees did not considere it when learned
- those observation were *unseen* for those trees, like in cross-validation (**OOB = out-of-bag**)

Bonus 1: OOB error

- for unseen each observation there are $\frac{B}{3}$ predictions
- can "average" prediction among trees, observation and obtain an estimate of the testing error (OOB error)
  - like with cross-fold validation
  - for free!

OOB error

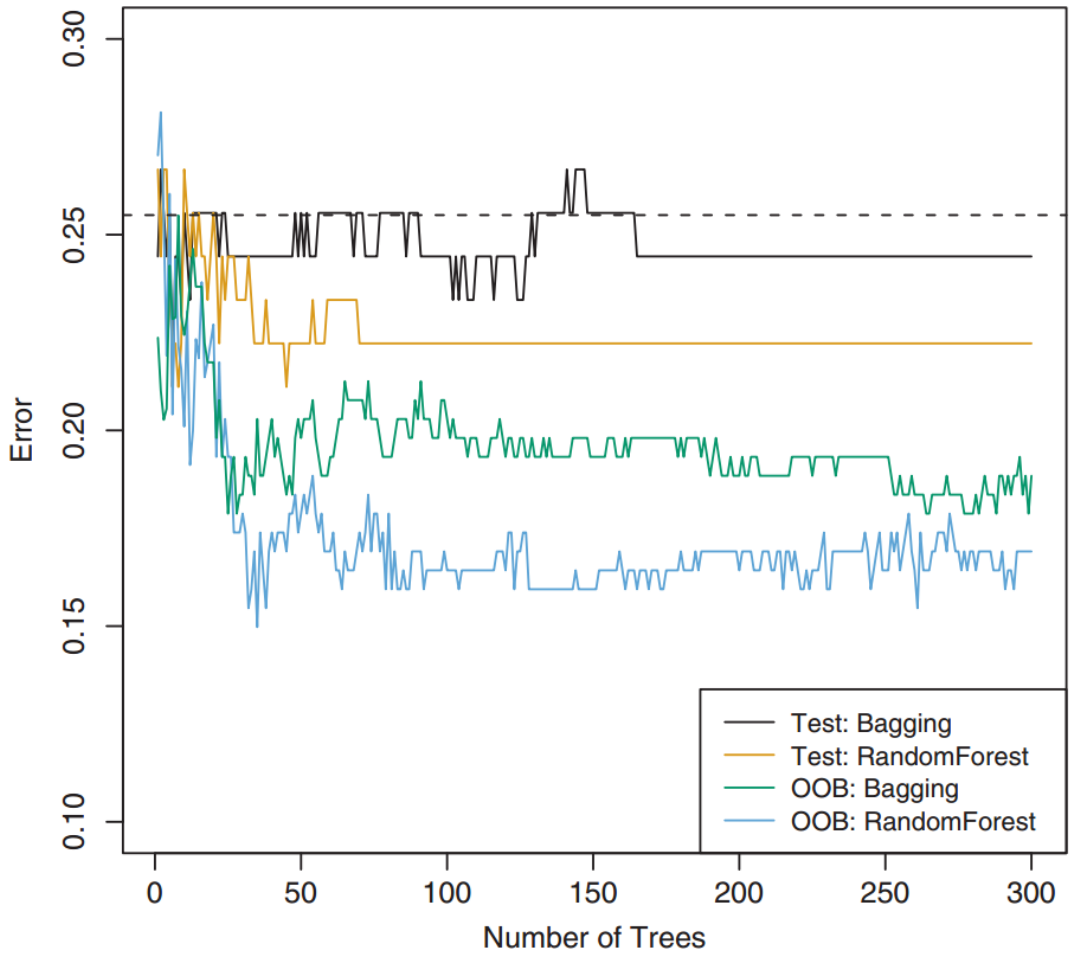


Image from An Introduction to Statistical Learning

Why estimating the test error?

Because the test data, in real world, is not available!

- will my ML solution work?

Bagging/RF and explicability

- Trees are easily understandable $\rightarrow$ explicability
- Hundreds of trees are not!

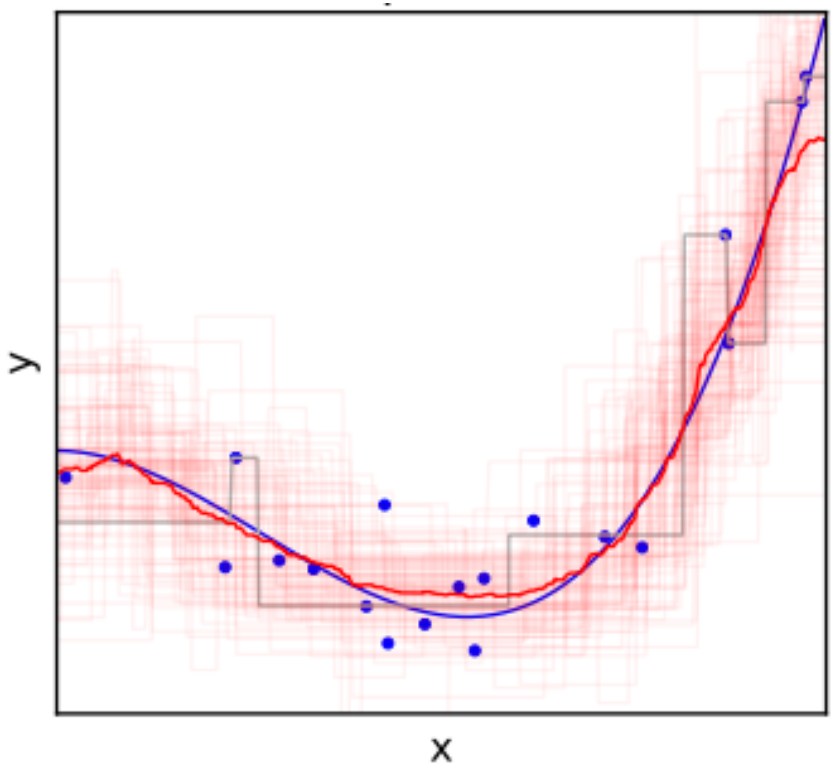


Image from F. Daolio

Bagging/RF and explicability: idea!

While learning:

1. for each tree, at each split
   - 1.1 keep note of the split variable
   - 1.2 keep note of RSS/Gini reduction
2. for each variable, sum reductions

The largest reduction, the more important the variable!

Bonus 2: variable importance

Instead of explicability based on tree shape:

- importance of variables based on RSS/Gini reduction

Nature of the prediction

Consider classification:

- tree $\rightarrow$ the class
- forest $\rightarrow$ the class, as resulting from a voting

Nature of the prediction

Consider classification:

- tree $\rightarrow$ the class
  - "virginica" is just "virginica"
- forest $\rightarrow$ the class, as resulting from a voting
  - "241 virginica, 170 versicolor, 89 setosa" is different than "478 virginica, 10 versicolor, 2 setosa"

Different **confidence** in the prediction

Bonus 3: confidence/tunability

Voting outcome:

- in classification, a measure of confidence of the decision
- in binary classification, voting threshold can be tuned to adjust bias towards one class (*sensitivity*)

Q: in regression?

Subsection 3

Binary classification

Binary classification

Binary classification:

- one of the most common classes of problems
- (comparative) evaluation is important!

Binary classification: evaluation

Consider the problem of classifying a person ('s data) as suffering or not suffering from a disease X.

Suppose we have "an accuracy of 99.99%". **Q:** is it good?

Binary classification: positives/negatives

Consider the problem of classifying a person ('s data) as suffering or not suffering from a disease X.

- **positive**: an observation of "suffering" class
- **negative**: an observation of "not suffering" class

In other problems, positive may mean a different thing: define it!

Effectiveness indexes: FPR, FNR

Given some labeled data and a classifier for the disease X problem, we can measure:

- the number of negative observations *wrongly* classified as positives: False Positives (**FP**)
- the number of positive observations *wrongly* classified as negatives: False Negatives (**FN**)

To decouple FP, FN from data size:

$$\text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$\text{FNR} = \frac{\text{FN}}{\text{P}} = \frac{\text{FN}}{\text{FN} + \text{TP}}$$

Accuracy and error rate

Relation of FPR, FNR with accuracy and error rate

$$\text{Accuracy} = 1 - \text{Error Rate}$$

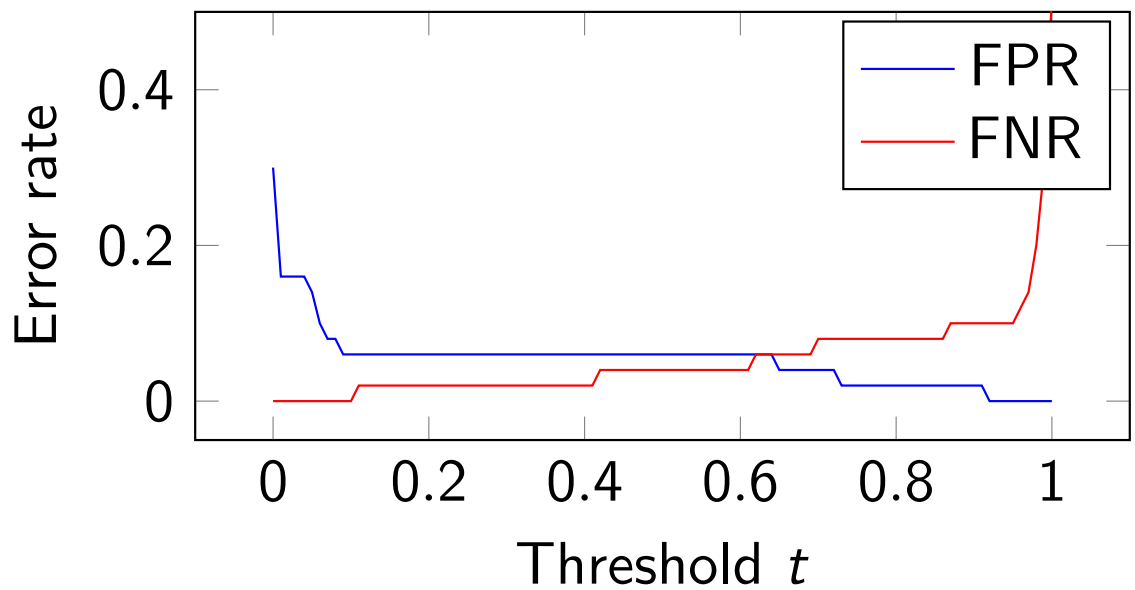
$$\text{Error Rate} = \frac{\text{FN} + \text{FP}}{\text{P} + \text{N}}$$

Q: Error Rate $\stackrel{?}{=} \frac{\text{FPR}+\text{FNR}}{2}$

FPR, FNR and sensitivity

- Suppose FPR $= 0.06$, FNR $= 0.04$ with threshold set to 0.5 (default for RF)
- One could be interested in "limiting" the FNR $\rightarrow$ change the threshold

Experimentally:



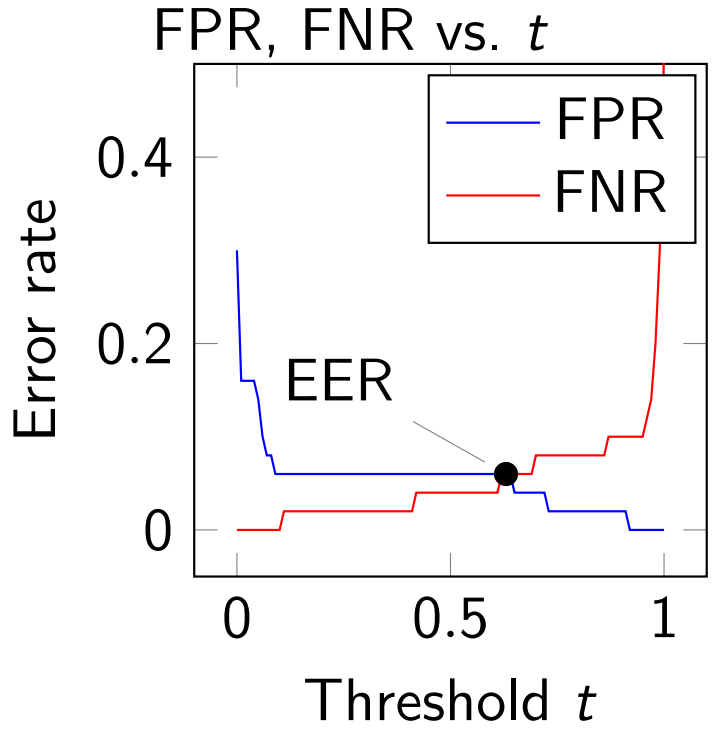
Comparing classifiers with FPR, FNR

- Classifier A: FPR $= 0.06$, FNR $= 0.04$
- Classifier B: FPR $= 0.10$, FNR $= 0.01$

Which one is the better?

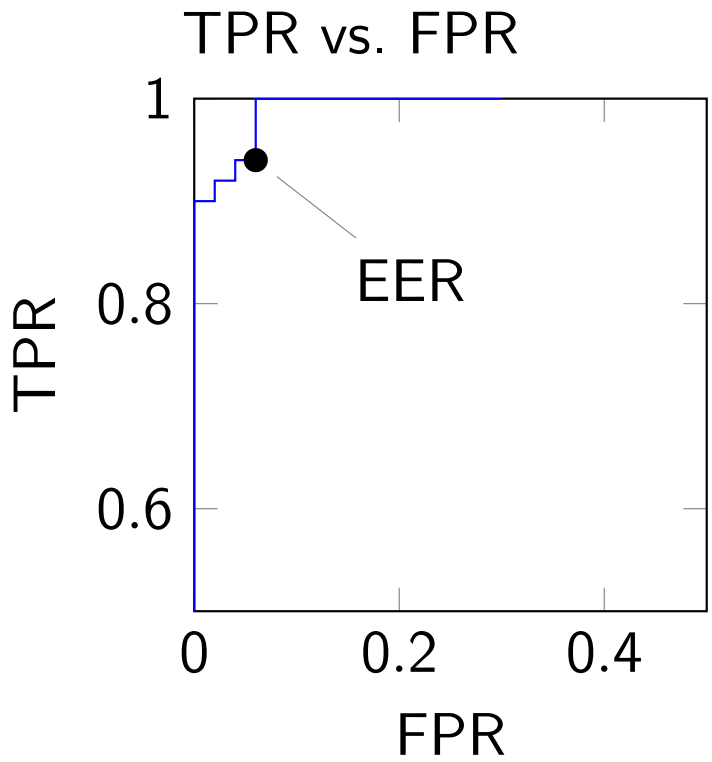
We'd like to have one single index $\rightarrow$ EER, AUC

Equal Error Rate (EER)



EER: the FPR at the value of t for which FPR $=$ FNR

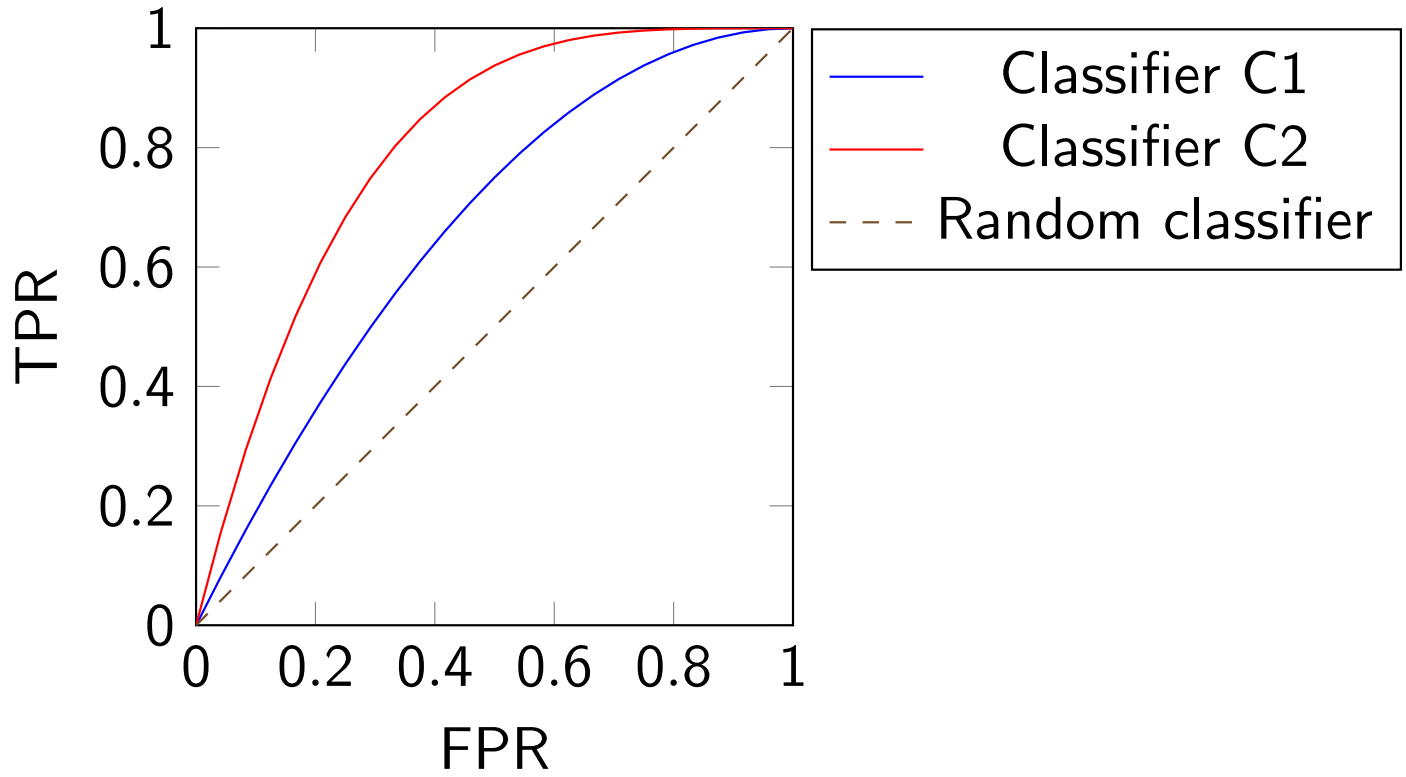
AUC: Area Under the Curve



AUC: the area under the TPR vs. FPR curve, plotted for different values of threshold t

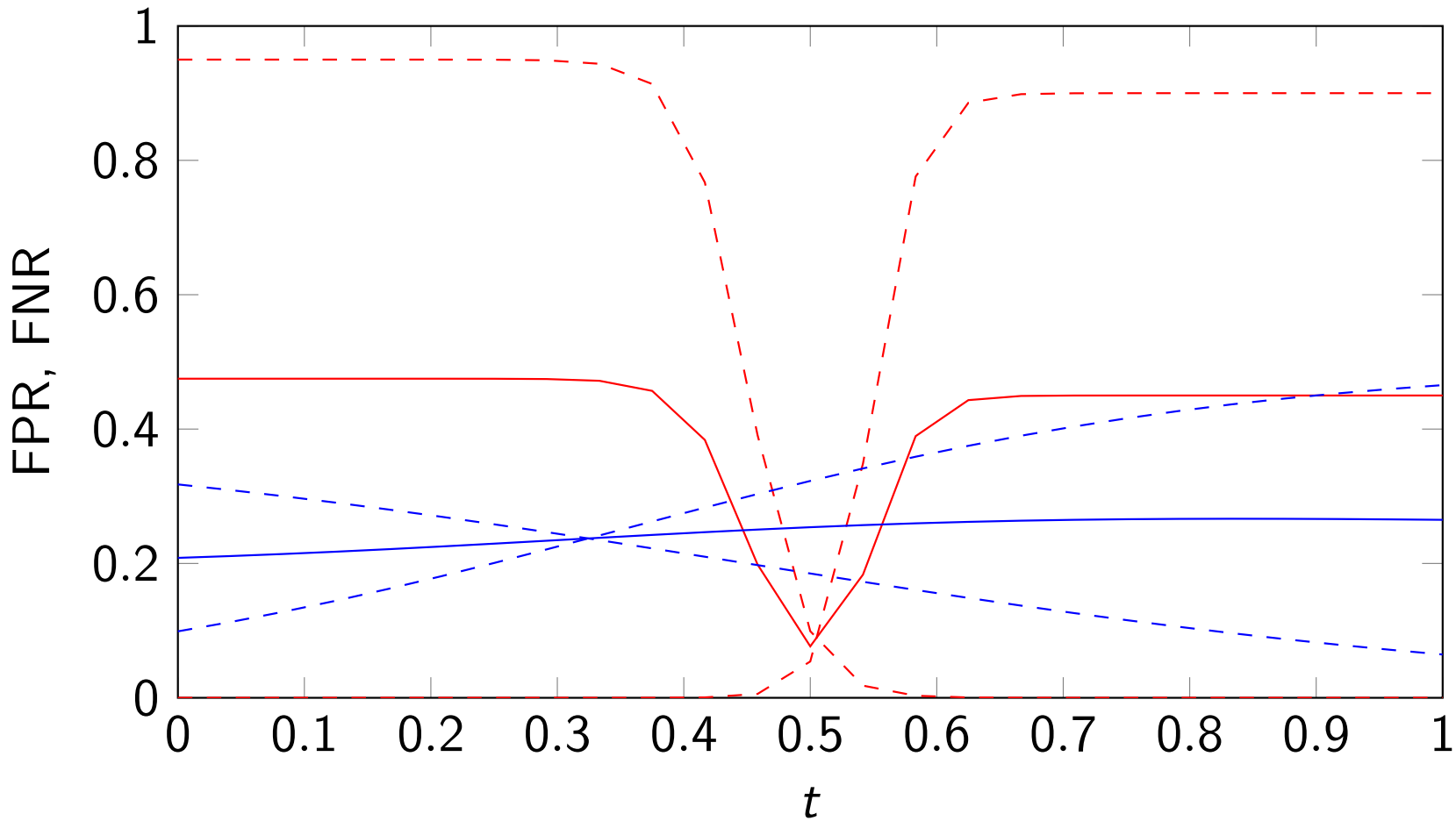
- the curve is called the *Receiver operating characteristic* (ROC)

ROC and comparison



Q: what does the bisector represent?

Other issues: robustness w.r.t. the threshold



"Same" with other parameters

Other issues: robustness w.r.t. random components

Consider A vs. B, AUC measured with cross-fold validation:

- A: $0.85, 0.73, 0.91, \cdots \rightarrow \mu = 0.83, \sigma = 0.15$
- B: $0.81, 0.78, 0.79, \cdots \rightarrow \mu = 0.81, \sigma = 0.03$

Can we say that A is better than B? (for effectiveness only)

In general, other sources of performance variability:

- random seed
- subclass of problem class (e.g., image recognition of dogs, cats, . . .)

Comparing techniques

Technique A, B; different index (e.g., AUC) values:

- A $\rightarrow (x_a^1, x_a^2, \dots) \rightarrow$ random variable X_a
- B $\rightarrow (x_b^1, x_b^2, \dots) \rightarrow$ random variable X_b

Do X_a, X_b follow different distributions?

- yes: A and B are different (concerning the AUC)
- no: difference in μ_a, μ_b might be due to randomness $\rightarrow$ A, B are not *significantly* different

Statistical significance in a nutshell

Just the way of thinking:

1. State a set of assumptions (the *null hypothesis* H_0), e.g.:
   - X_a, X_b are normally distributed and independent
   - $\bar{x}_a = \bar{x}_b$ (or $\bar{x}_a \geq \bar{x}_b$)
   - any other assumption in the *statistical model*

Statistical significance in a nutshell

Just the way of thinking:

1. State a set of assumptions (the *null hypothesis* H_0), e.g.:
   - X_a, X_b are normally distributed and independent
   - $\bar{x}_a = \bar{x}_b$ (or $\bar{x}_a \geq \bar{x}_b$)
   - any other assumption in the *statistical model*
2. Perform a statistical test, appropriate choice depending on many factors, e.g.:
   - Wilcoxon test (many versions)
   - Friedman (many versions)
   - ...

Statistical significance in a nutshell

Just the way of thinking:

1. State a set of assumptions (the *null hypothesis* H_0), e.g.:
   - X_a, X_b are normally distributed and independent
   - $\bar{x}_a = \bar{x}_b$ (or $\bar{x}_a \geq \bar{x}_b$)
   - any other assumption in the *statistical model*
2. Perform a statistical test, appropriate choice depending on many factors, e.g.:
   - Wilcoxon test (many versions)
   - Friedman (many versions)
   - ...
3. ...which outputs a p-value $\in [0, 1]$
   - 0 is "good", 1 is "bad"

p-value: meaning

0 is "good", 1 is "bad"
The p-value is the degree to which the data conform to the pattern predicted by the null hypothesis

- p-value $= P(x_a^1, x_a^2, \ldots, x_b^1, x_b^2, \ldots | H_0)$

If p-value is low:

- we've been very (un)lucky in having observed $x_a^1, x_a^2, \ldots, x_b^1, x_b^2, \ldots$
- "maybe" because H_0 is not true

p-value: meaning

0 is "good", 1 is "bad"
The p-value is the degree to which the data conform to the pattern predicted by the null hypothesis

- p-value $= P(x_a^1, x_a^2, \ldots, x_b^1, x_b^2, \ldots | H_0)$

If p-value is low:

- we've been very (un)lucky in having observed $x_a^1, x_a^2, \ldots, x_b^1, x_b^2, \ldots$
- "maybe" because H_0 is not true
  - **Warning!** Any part of H_0, not necessarily the $\bar{x}_a = \bar{x}_b$ part!

Statistical significance

Things are much more complex than this. . .

Some interesting papers:

- Joaquín Derrac et al. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms". In: *Swarm and Evolutionary Computation* 1.1 (2011), pp. 3–18
- Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. "How Many Random Seeds? Statistical Power Analysis in Deep Reinforcement Learning Experiments". In: *arXiv preprint arXiv:1806.08295* (2018)
- Sander Greenland et al. "Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations". In: *European journal of epidemiology* 31.4 (2016), pp. 337–350

Subsection 4

Boosting

Many views and aggregation

In bagging/RF (regression):

- many views are different samples
- aggregation is average

Alternative:

- many views are subsequent residuals
- aggregation is the sum

Boosting

When learning:

1. Current data is learning data
2. Repeat B times
   - 2.1 learn a tree on current data
   - 2.2 current data becomes residuals of learned tree $(\mathbf{y} - \hat{\mathbf{y}})$

When predicting:

1. Repeat B times
   - 1.1 get a prediction from ith learned tree
2. sum prediction

Q: implementation differences w.r.t. RF?

Boosting (regression)

function BoostTrees($\mathbf{X}, \mathbf{y}$)
  $t(\mathbf{X}) \leftarrow \mathbf{0}$
  for $i \in \{1, 2, \ldots, B\}$ **do**
    $t_i \leftarrow$ BuildRegressionTree($\mathbf{X}, \mathbf{y}, d$)
    $t(\mathbf{X}) \leftarrow t(\mathbf{X}) + \lambda t_i(\mathbf{X})$
    $\mathbf{y} \leftarrow \mathbf{y} - \lambda t_i(\mathbf{X})$
  end for
  return t
end function

- Each learned tree should be simple (maximum splits d)
- λ slows down learning

Trickier with classification.

Boosting parameters

- λ usually set to 0.01 or 0.001
- λ and B interact: for small λ, B should be large
- large B can lead to overfitting (unlike bagging/RF, **Q:** why)

Find a good value for B with cross-validation

(Both boosting and bagging general techniques)

Bagging/RF/boosting in summary

	Tree	Bagging	RF	Boosting
interpretability	▲			
numeric/categorical	▲	▲	▲	▲
accuracy	▼		▲	▲
test error estimate		▲	▲	
variable importance		▲	▲	▲
confidence/tunability		▲	▲	
fast to learn	▲*			▼
(almost) non-parametric		▲	▲	

*: **Q:** how faster? when? does it matter?