

Machine Learning: short introduction (trees and ANN), part 2

Sergey Korpachev on behalf of Dépôt

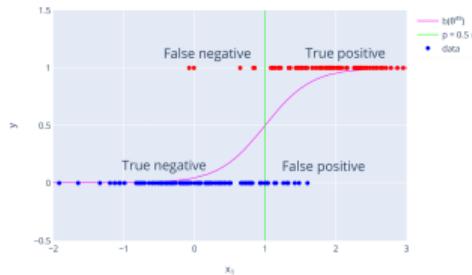
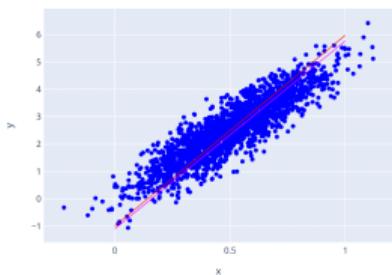
Outline

- Machine learning (ML)
 - Data
 - Features in ML
 - ML pipeline
 - Linear model
 - Decision tree
 - Neural network
 - Summary

Decision tree

Decision tree

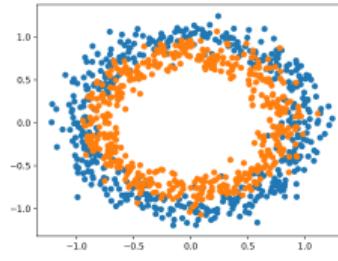
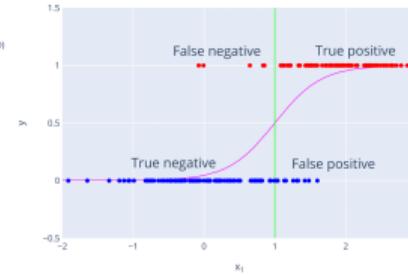
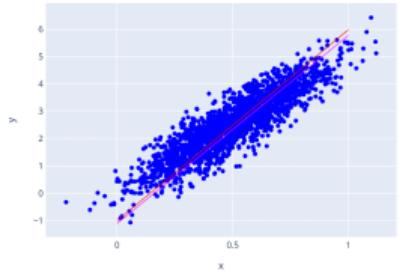
—○ motivation



- In the previous part we explored linear models:
 - nicely describe linear correlations
 - fast and easy to train

Decision tree

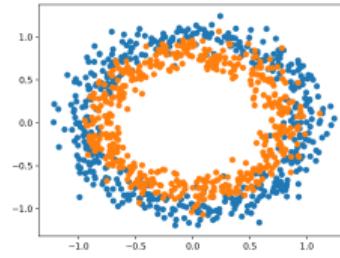
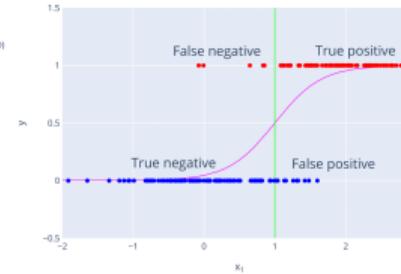
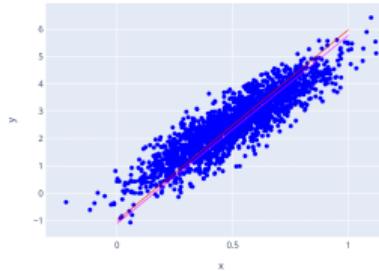
—○ motivation



- In the previous part we explored linear models:
 - nicely describe linear correlations
 - fast and easy to train
- However, they are poor in modelling non-linearities
- Additional heuristics might be applied (e.g. polynomial features) but are ad-hoc

Decision tree

—○ motivation



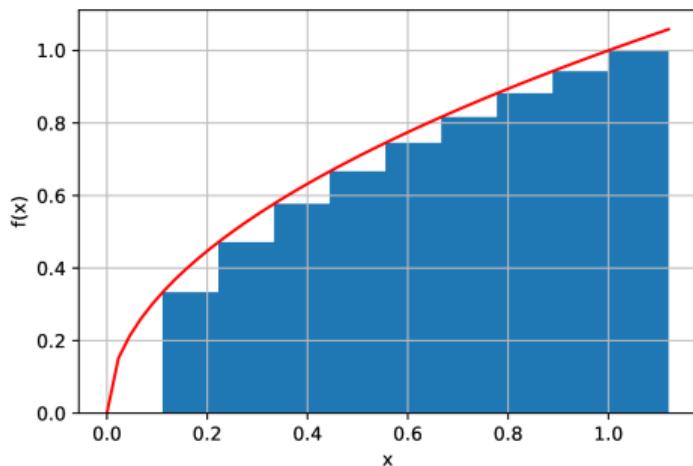
- In the previous part we explored linear models:
 - nicely describe linear correlations
 - fast and easy to train
- However, they are poor in modelling non-linearities
- Additional heuristics might be applied (e.g. polynomial features) but are ad-hoc

Is there a more general way to describe non-linearities?

Decision tree

—○ motivation

- What is the simplest way to approximate a function?
- Using **piecewise linear function**

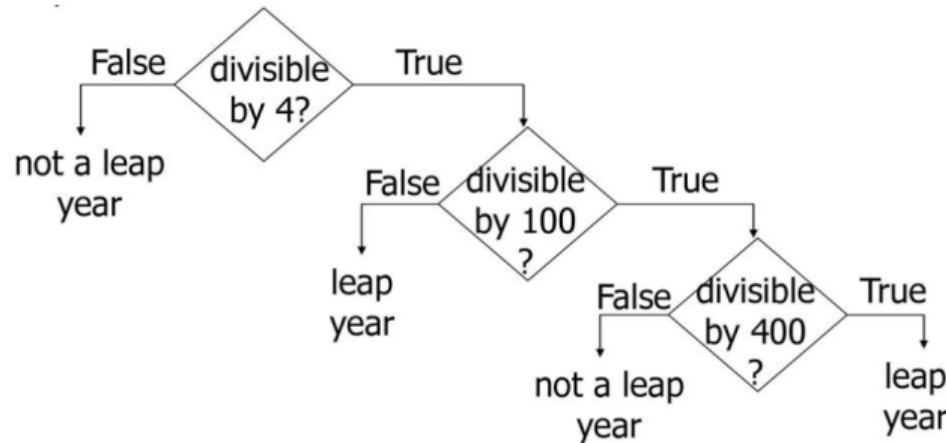


$$f(x) = \begin{cases} 0, & x < 0.12 \\ 0.35, & 0.12 \leq x < 0.22 \\ 0.47, & 0.22 \leq x < 0.36 \\ \dots \end{cases}$$

Decision tree

— o intuition

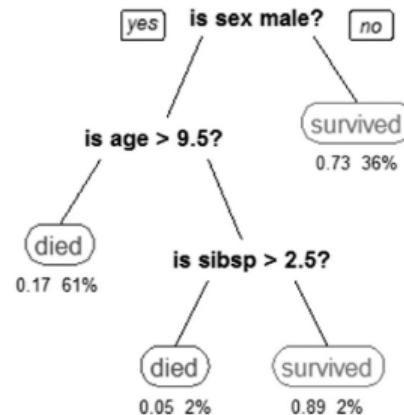
- What is the simplest way to categorize things?
- Ask yes/no questions



Decision tree

— o intuition

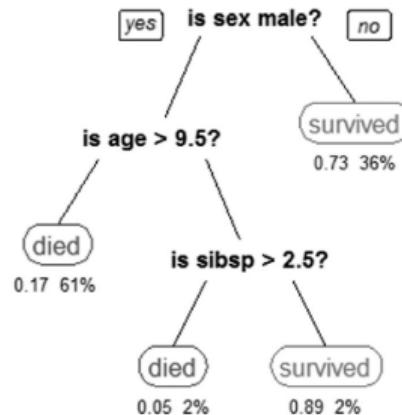
- What is the simplest way to categorize things?
- Ask yes/no questions



Decision tree

— o intuition

- What is the simplest way to categorize things?
- Ask yes/no questions



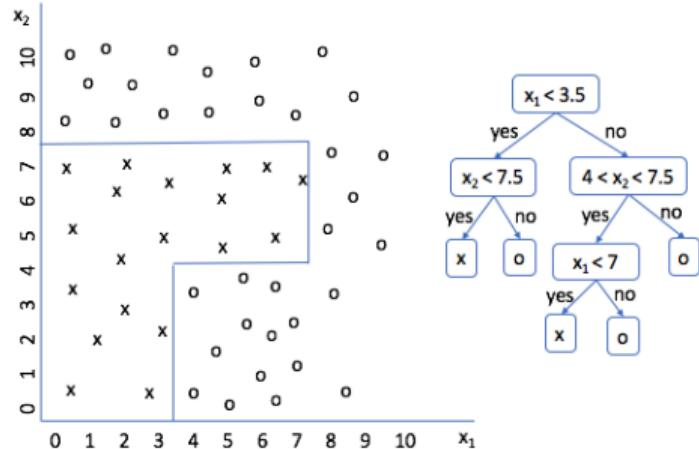
Can we mathematically formulate this?

Decision tree

—○ algorithm

Algorithm 1: Decision tree

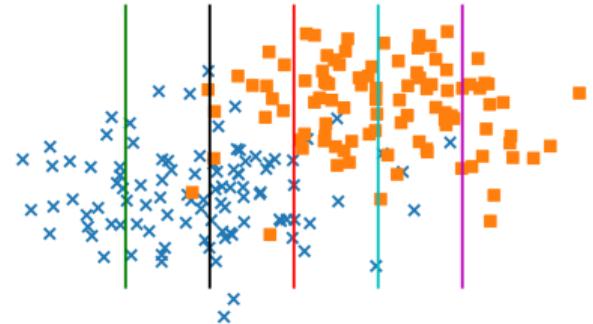
- 1: Initialize: hyperparameters, leaf set = $\{X_{\text{train}}\}$
- 2: **while** not stopping criteria **do**
- 3: leaf to split = Choose(leaf set)
- 4: left leaf, right leaf = Split(leaf to split)
- 5: Add left leaf, right leaf to leaf set
- 6: Remove leaf to split from leaf set
- 7: **end while**



Decision tree

—○ split criteria

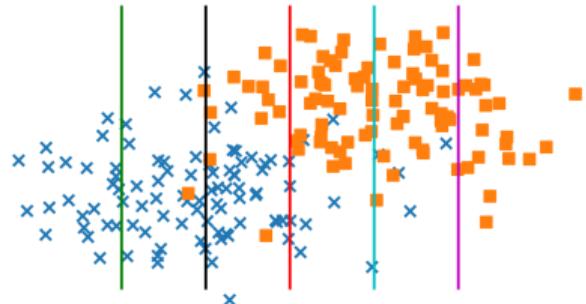
- Suppose there is a classification problem
- So we want to separate one class from the other by constructing **decision boundary**
- And using decision tree algorithm described earlier
- To do that we need to start splitting on features
- Which **split** is the best?



Decision tree

—○ split criteria

- Suppose there is a classification problem
- So we want to separate one class from the other by constructing **decision boundary**
- And using decision tree algorithm described earlier
- To do that we need to start splitting on features
- Which **split** is the best?
- We need some criteria



Decision tree

—○ split criteria

[more about IG](#)

- Suppose we have n features $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ in our dataset X
- If we want to split on some feature $x^{(j)}$ at threshold t then we propose to maximize **Information Gain** (IG):

$$IG(X, a) = H(X) - H(X | a) \rightarrow \max_a$$

$$H(X | a) = \sum_{v \in vals(a)} \frac{|S_a(v)|}{|X|} \cdot H(S_a(v))$$

- In case of a binary tree with $L(R)$ elements in the left (right) split we've got:

$$H(X, j, t) = \frac{|L|}{|X|} H(L) + \frac{|R|}{|X|} H(R) \rightarrow \min_{j,t}$$

Decision tree

—○ split criteria

[more about IG](#)

- Suppose we have n features $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ in our dataset X
- If we want to split on some feature $x^{(j)}$ at threshold t then we propose to maximize **Information Gain** (IG):

$$IG(X, a) = H(X) - H(X | a) \rightarrow \max_a$$

$$H(X | a) = \sum_{v \in vals(a)} \frac{|S_a(v)|}{|X|} \cdot H(S_a(v))$$

- In case of a binary tree with $L(R)$ elements in the left (right) split we've got:

$$H(X, j, t) = \frac{|L|}{|X|} H(L) + \frac{|R|}{|X|} H(R) \rightarrow \min_{j,t}$$

How to choose $H(X)$?

Decision tree



split criteria

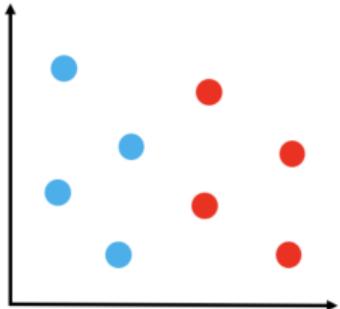


The "best" split is a central one because it introduces **purity** in the best way. And we have some functions to measure the purity!

Decision tree

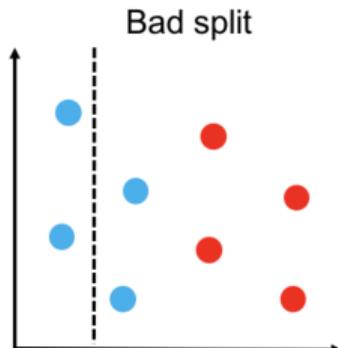


split criteria



$$p = \frac{4}{8}, p = \frac{4}{8}$$

$$I = \frac{4}{8} \left(1 - \frac{4}{8}\right) + \frac{4}{8} \left(1 - \frac{4}{8}\right) = \frac{1}{2}$$



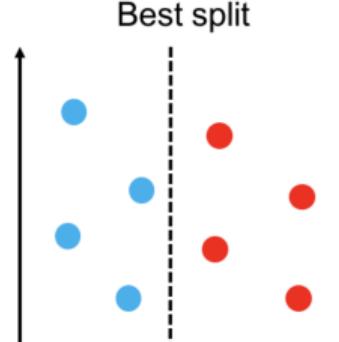
$$p = \frac{2}{2}, p = \frac{0}{2}$$

$$I_{left} = \frac{2}{2} \left(1 - \frac{2}{2}\right) + \frac{0}{2} \left(1 - \frac{0}{2}\right) = 0$$

$$p = \frac{2}{6}, p = \frac{4}{6}$$

$$I_{right} = \frac{2}{6} \left(1 - \frac{2}{6}\right) + \frac{4}{6} \left(1 - \frac{4}{6}\right) = \frac{4}{9}$$

$$\Delta I = \frac{1}{2} - 0 \frac{2}{8} - \frac{4}{9} \frac{6}{8} = \frac{1}{6}$$



$$p = \frac{4}{4}, p = \frac{0}{4}$$

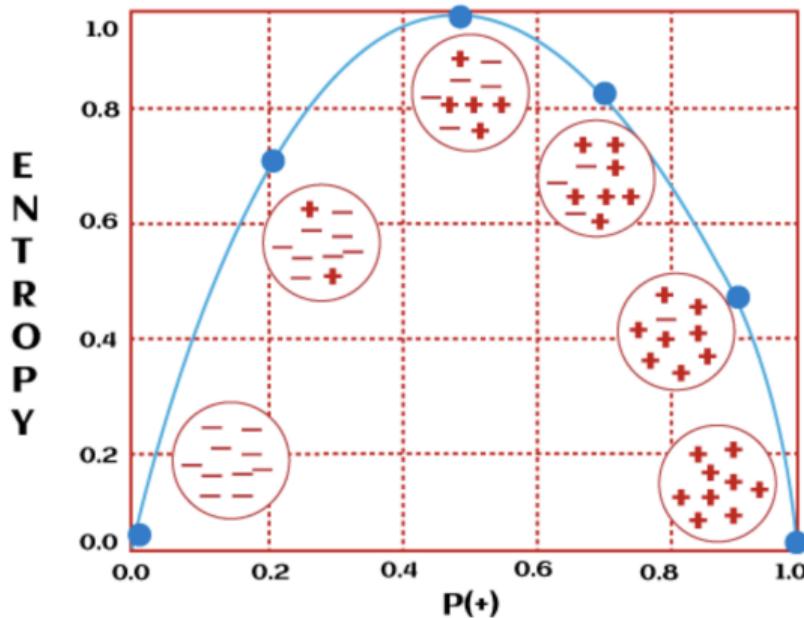
$$I_{left} = \frac{4}{4} \left(1 - \frac{4}{4}\right) + \frac{0}{4} \left(1 - \frac{0}{4}\right) = 0$$

$$p = \frac{0}{4}, p = \frac{4}{4}$$

$$I_{right} = \frac{0}{4} \left(1 - \frac{0}{4}\right) + \frac{4}{4} \left(1 - \frac{4}{4}\right) = 0$$

$$\Delta I = \frac{1}{2} - 0 \frac{4}{8} - 0 \frac{4}{8} = \frac{1}{2}$$

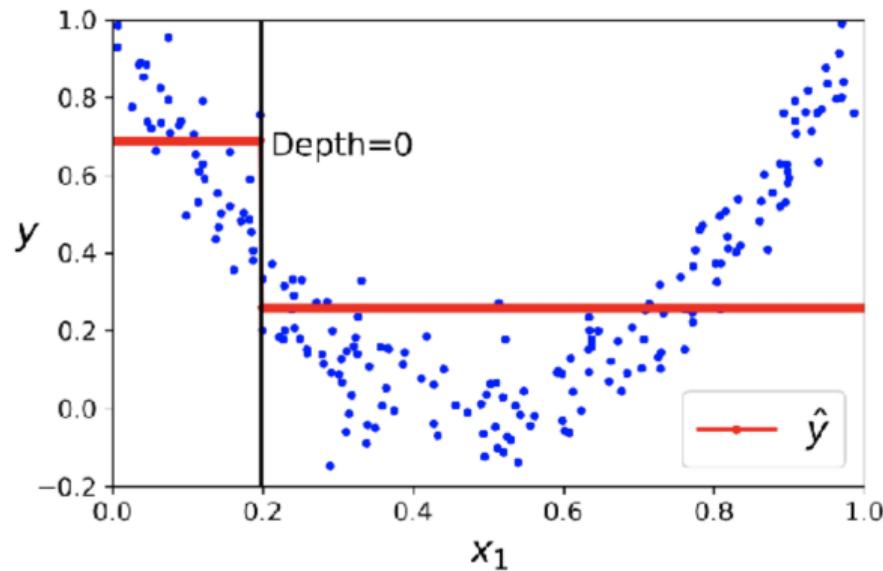
Decision tree —○ split criteria



Михаил Гущин, НИУ ВШЭ

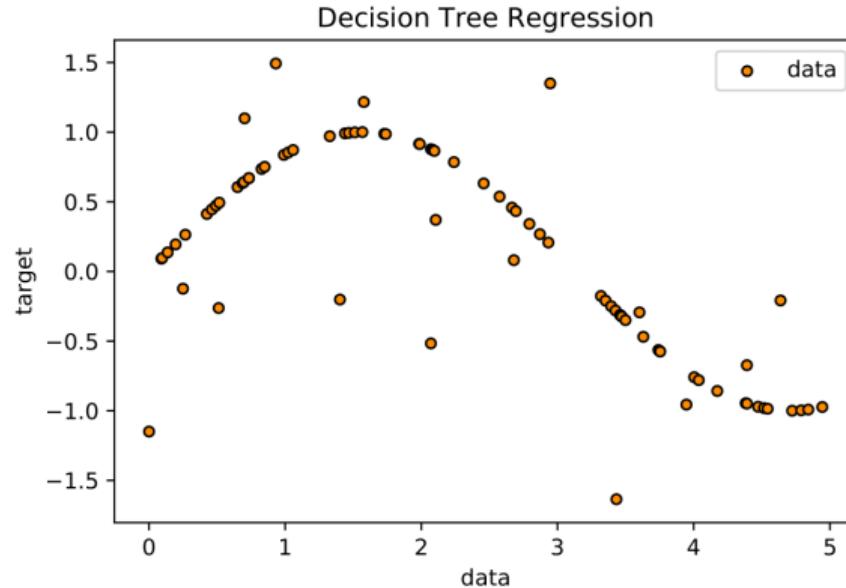
Источник: <https://www.javatpoint.com/entropy-in-machine-learning>

Decision tree —○ split criteria



Decision tree

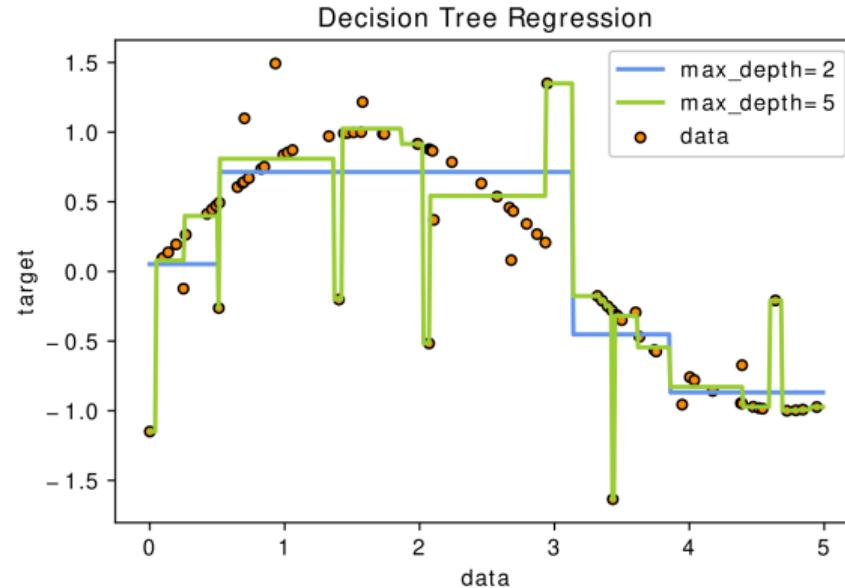
—○ regression



Can we use decision tree approach for regression problem?

Decision tree

—○ regression



Can we use decision tree approach for regression problem? Yes

Decision tree

—○ growing

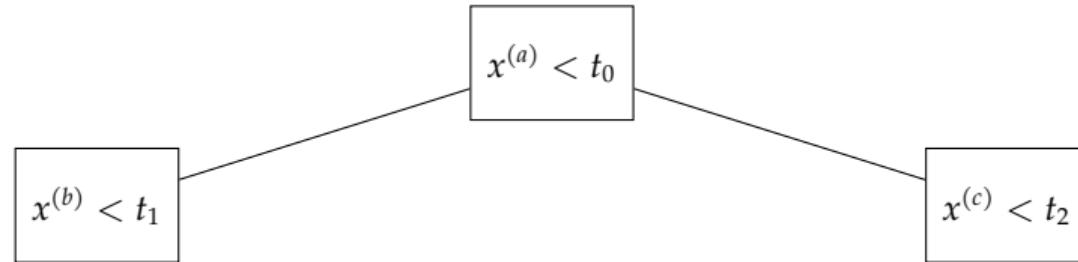
What about tree construction?

$$x^{(a)} < t_0$$

Decision tree

—○ growing

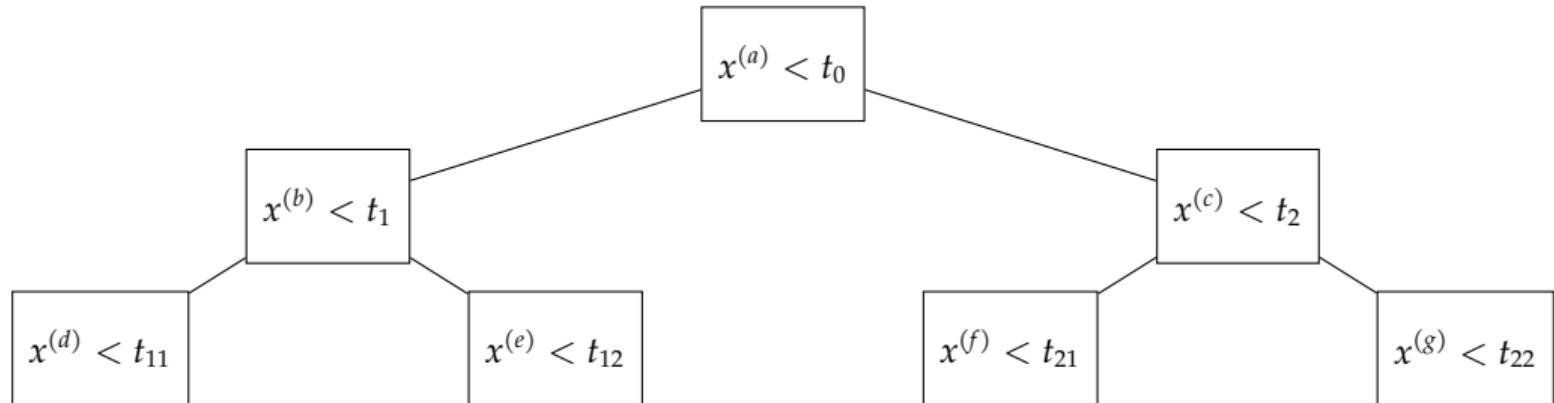
What about tree construction? Do it recursively by greedy algorithm



Decision tree

—○ growing

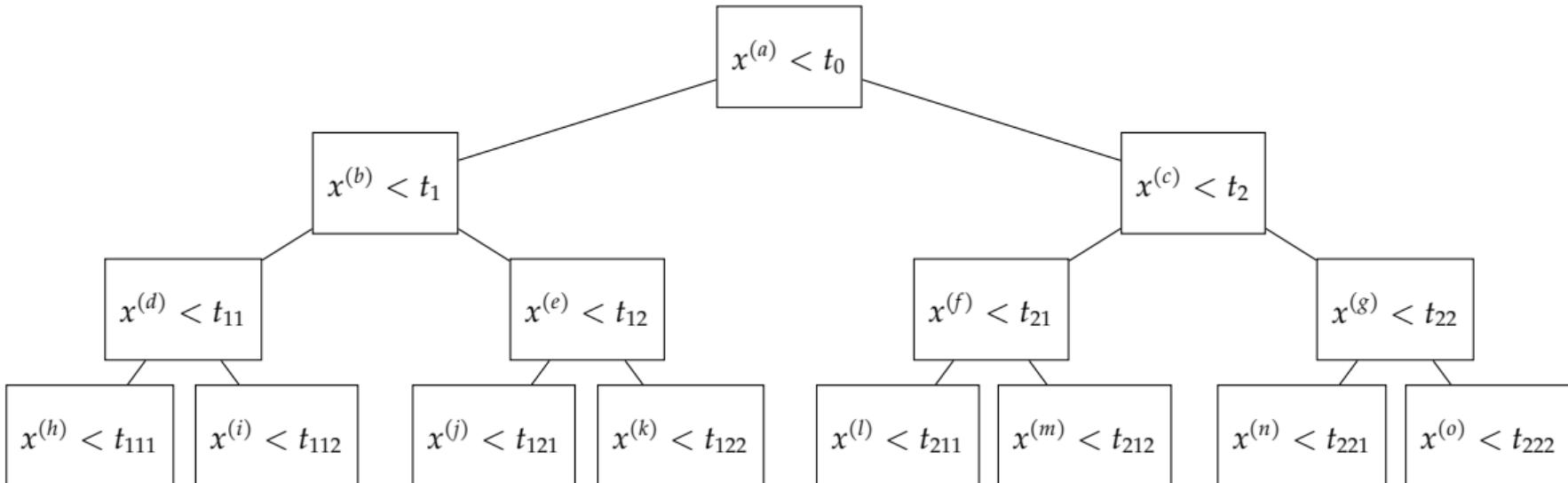
What about tree construction? Do it recursively by **greedy** algorithm



Decision tree

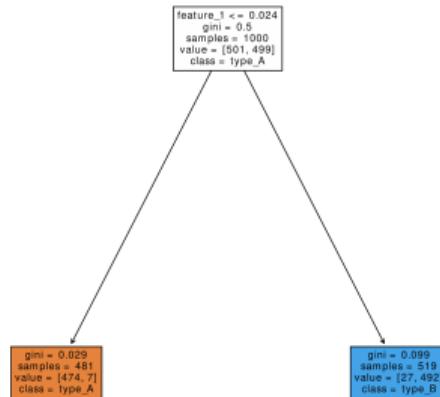
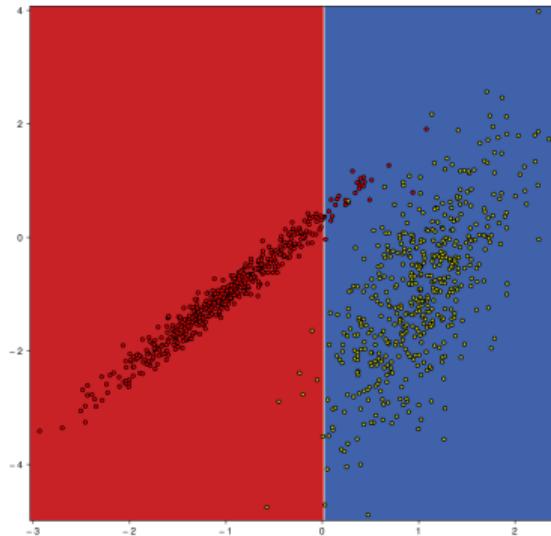
—○ growing

What about tree construction? Do it recursively by greedy algorithm



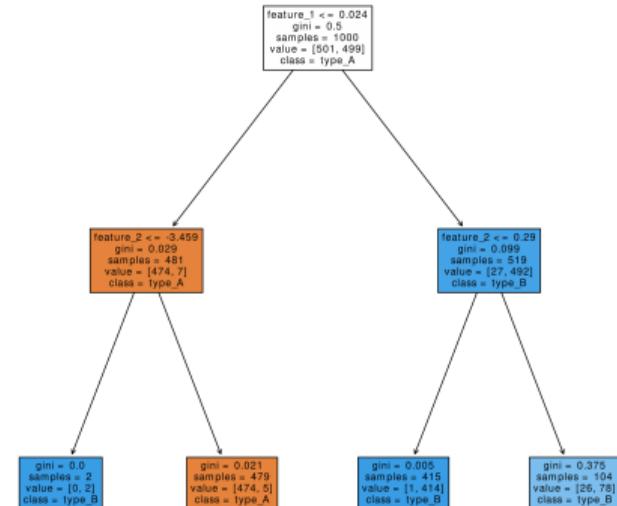
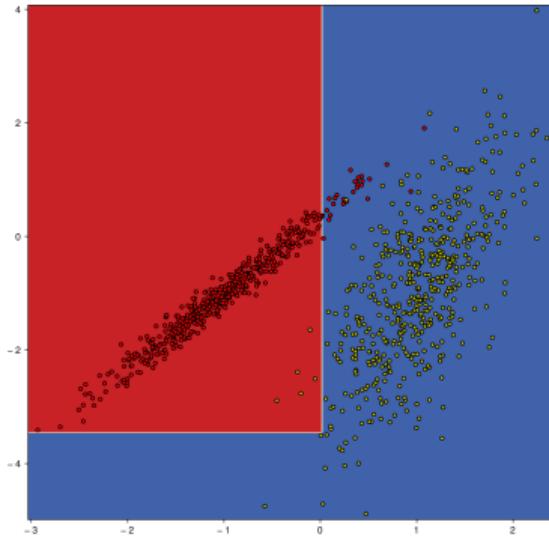
Decision tree growing

check out also [these](#) visuals



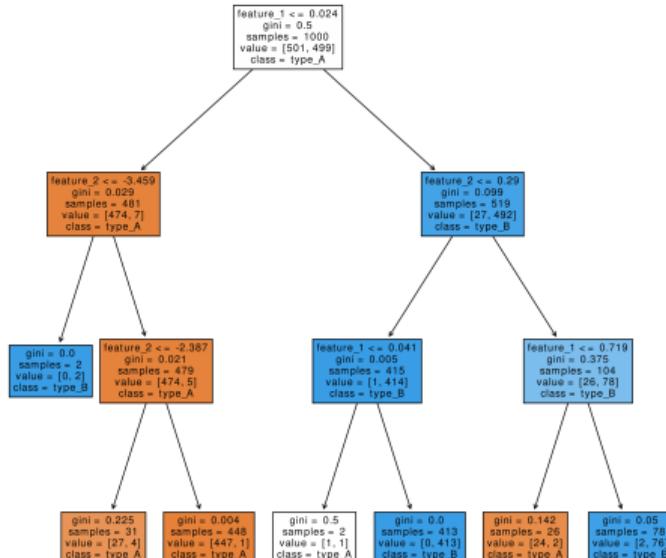
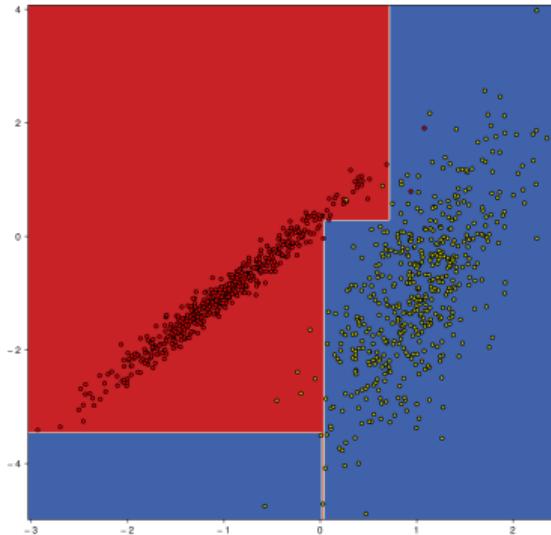
Decision tree growing

check out also these visuals



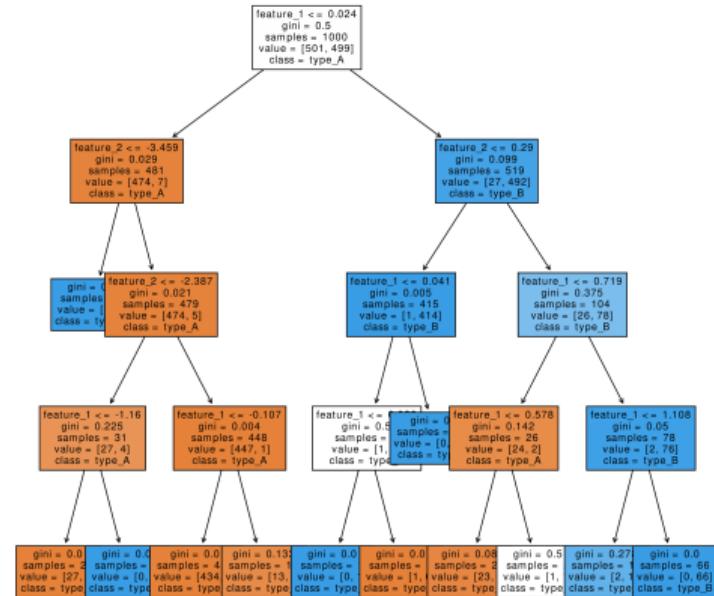
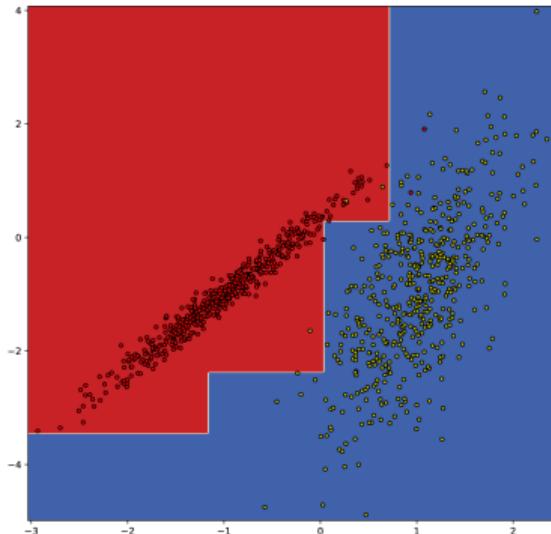
Decision tree \longrightarrow growing

check out also [these](#) visuals



Decision tree — growing

check out also [these](#) visuals



Random forest and gradient boosting

Random forest and gradient boosting

—○ what's next after decision trees?

