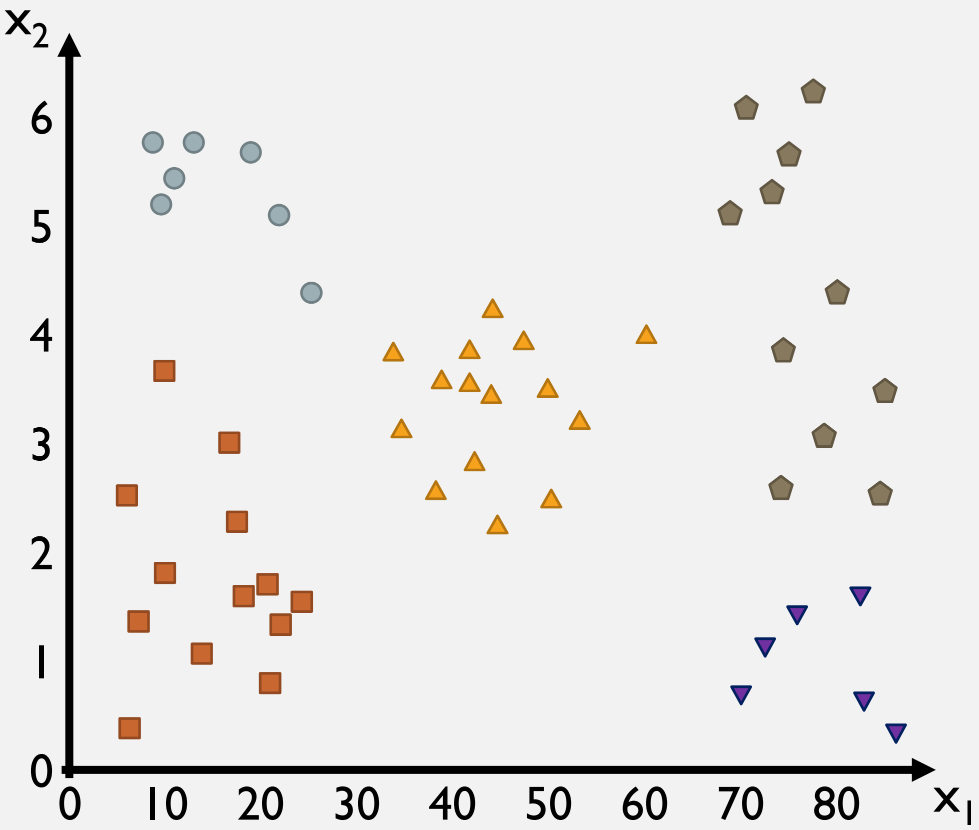
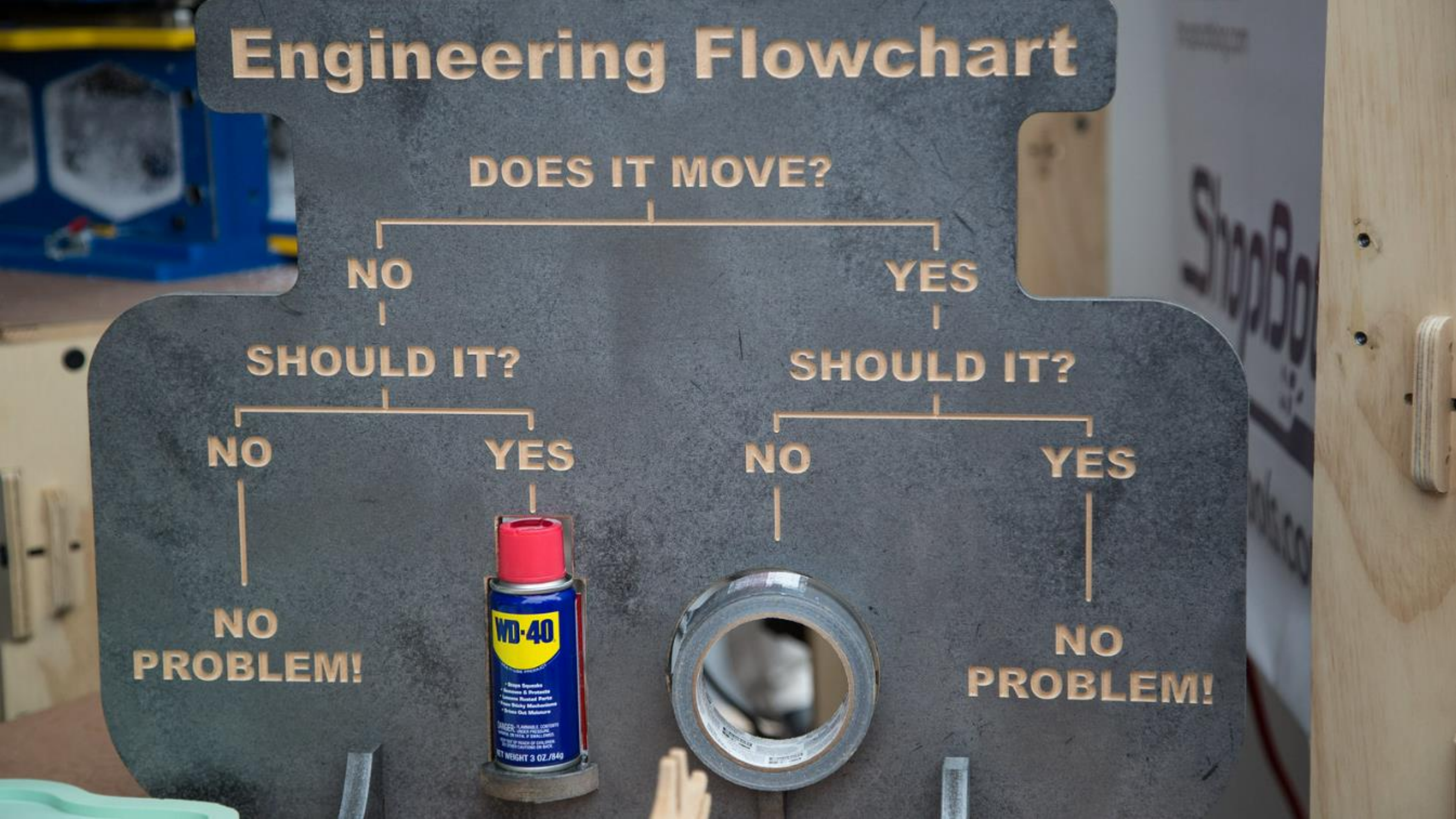
TREE-BASED MODELS

Lecture 3

MAL1, 2024

DECISION TREES





Engineering Flowchart

DOES IT MOVE?

NO

SHOULD IT?

NO

NO PROBLEM!

YES

YES

SHOULD IT?

NO

YES

NO PROBLEM!

HOW DECISION TREES WORK

- Step 1

  Find the feature that is the best predictor of your data

- Step 2

  Partition instances of your training set according to that feature

- Step 3

  Repeat 1-2 recursively

- Stop when

  All instances in a given node belong to the same class
  or
  There are no more ways to split

A LOAN IN THE BANK

A FICTITIOUS EXAMPLE

id	salary	savings	debt	class
1	Low	High	True	Approved
2	Low	Low	False	Declined
3	High	Low	False	Approved
4	Low	Low	True	Declined
5	High	Low	True	Approved
6	High	High	False	Approved
7	High	Low	False	Approved
8	Low	Low	True	Declined
9	High	High	True	Approved
10	Low	Low	False	Declined
11	Low	High	False	Approved
12	Low	Low	True	Declined

How do we decide which feature to branch off on?

A LOAN IN THE BANK

A FICTITIOUS EXAMPLE

id	salary	savings	debt	class
1	Low	High	True	Approved
2	Low	Low	False	Declined
3	High	Low	False	Approved
4	Low	Low	True	Declined
5	High	Low	True	Approved
6	High	High	False	Approved
7	High	Low	False	Approved
8	Low	Low	True	Declined
9	High	High	True	Approved
10	Low	Low	False	Declined
11	Low	High	False	Approved
12	Low	Low	True	Declined

The Gini impurity index

$$G(D) = 1 - \sum_j p_j^2 = 1 - \left(\frac{\quad}{\quad}\right)^2 - \left(\frac{\quad}{\quad}\right)^2 =$$

$$G_k(D) = \sum_i \frac{n_i}{n} G(D_i)$$

$$G_{\text{salary}}(D) = \frac{\quad}{\quad}\left(1 - \left(\frac{\quad}{\quad}\right)^2 - \left(\frac{\quad}{\quad}\right)^2\right) + \frac{\quad}{\quad}\left(1 - \left(\frac{\quad}{\quad}\right)^2 - \left(\frac{\quad}{\quad}\right)^2\right)$$

A LOAN IN THE BANK

A FICTITIOUS EXAMPLE

id	salary	savings	debt	class
1	Low	High	True	Approved
2	Low	Low	False	Declined
3	High	Low	False	Approved
4	Low	Low	True	Declined
5	High	Low	True	Approved
6	High	High	False	Approved
7	High	Low	False	Approved
8	Low	Low	True	Declined
9	High	High	True	Approved
10	Low	Low	False	Declined
11	Low	High	False	Approved
12	Low	Low	True	Declined

The Gini impurity index

$$G_{\text{salary}}(D) = 0.24$$
$$G_{\text{savings}}(D) = 0.31$$
$$G_{\text{debt}}(D) = 0.47$$

A LOAN IN THE BANK

A FICTITIOUS EXAMPLE

id	salary	savings	debt	class
1	Low	High	True	Approved
2	Low	Low	False	Declined
3	High	Low	False	Approved
4	Low	Low	True	Declined
5	High	Low	True	Approved
6	High	High	False	Approved
7	High	Low	False	Approved
8	Low	Low	True	Declined
9	High	High	True	Approved
10	Low	Low	False	Declined
11	Low	High	False	Approved
12	Low	Low	True	Declined

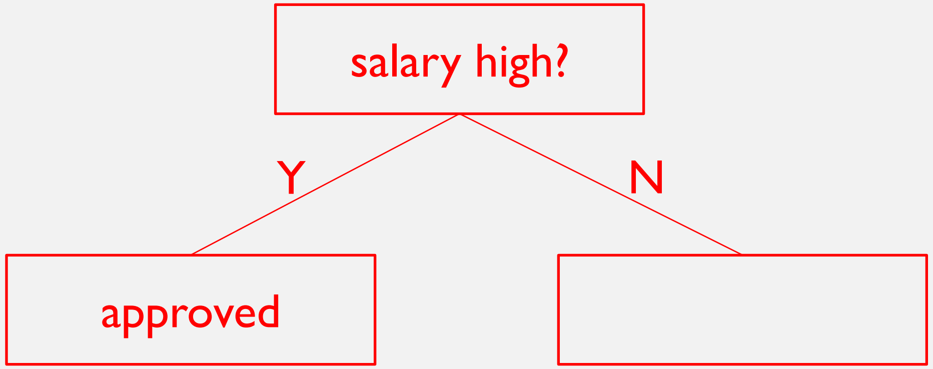
Beginning to draw the tree

A LOAN IN THE BANK

A FICTITIOUS EXAMPLE

id	salary	savings	debt	class
1	Low	High	True	Approved
2	Low	Low	False	Declined
3	High	Low	False	Approved
4	Low	Low	True	Declined
5	High	Low	True	Approved
6	High	High	False	Approved
7	High	Low	False	Approved
8	Low	Low	True	Declined
9	High	High	True	Approved
10	Low	Low	False	Declined
11	Low	High	False	Approved
12	Low	Low	True	Declined

Finishing the tree



salary high?

Y

N

approved

LEARNING DECISION TREES

- means learning the sequence of questions that gets us to the best answer most quickly

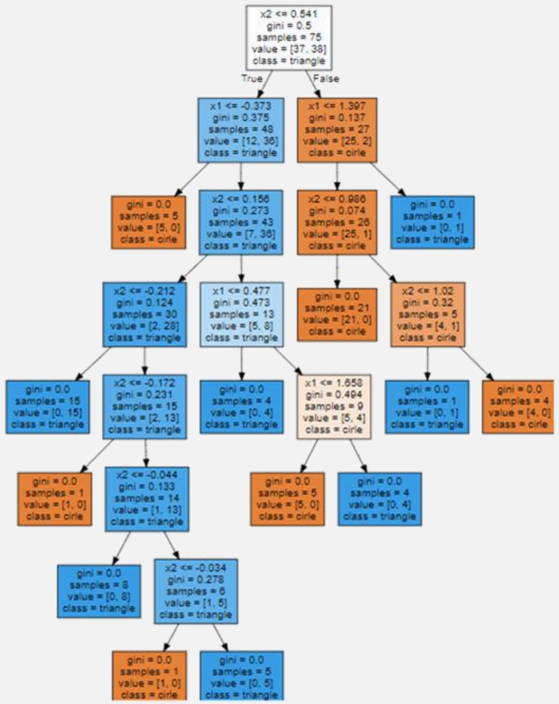
- the questions may be yes/no but usually of the form " "

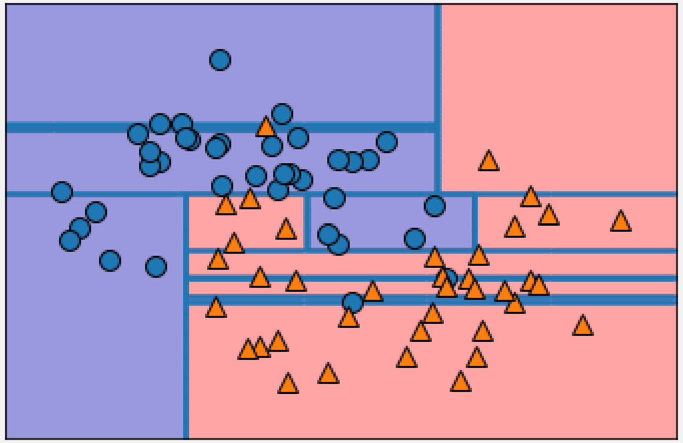
- the algorithm searches over all possible and finds the most one

VISUALIZATION

Jupyter Notebook **Decision Trees I: Visualization and hyperparameters**

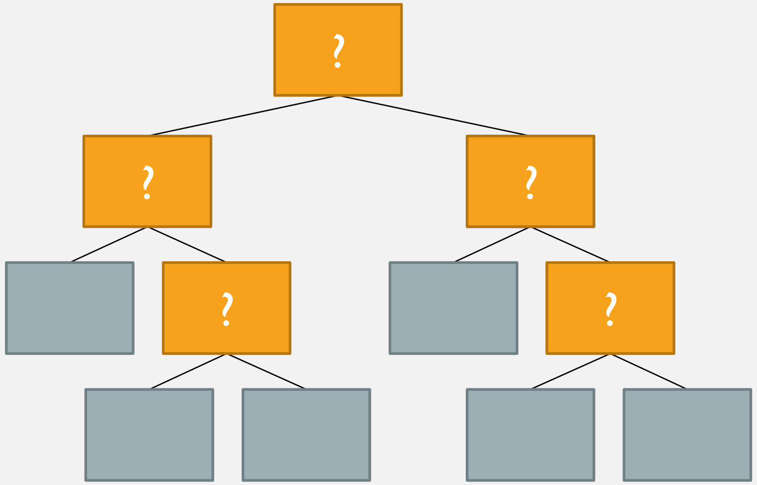
VISUALIZATION





OVERFITTING AND HYPERPARAMETERS

```
Accuracy on training data: 1.0
Accuracy on testing data: 0.92
```



max_depth

max_leaf_notes

min_samples_split

(**criterion**)

Tuning these parameters is called *pre-pruning*

PRE-PRUNING

Jupyter Notebook **Decision Trees I: Visualization and hyperparameters**

PROS AND CONS OF DECISION TREES

Pros

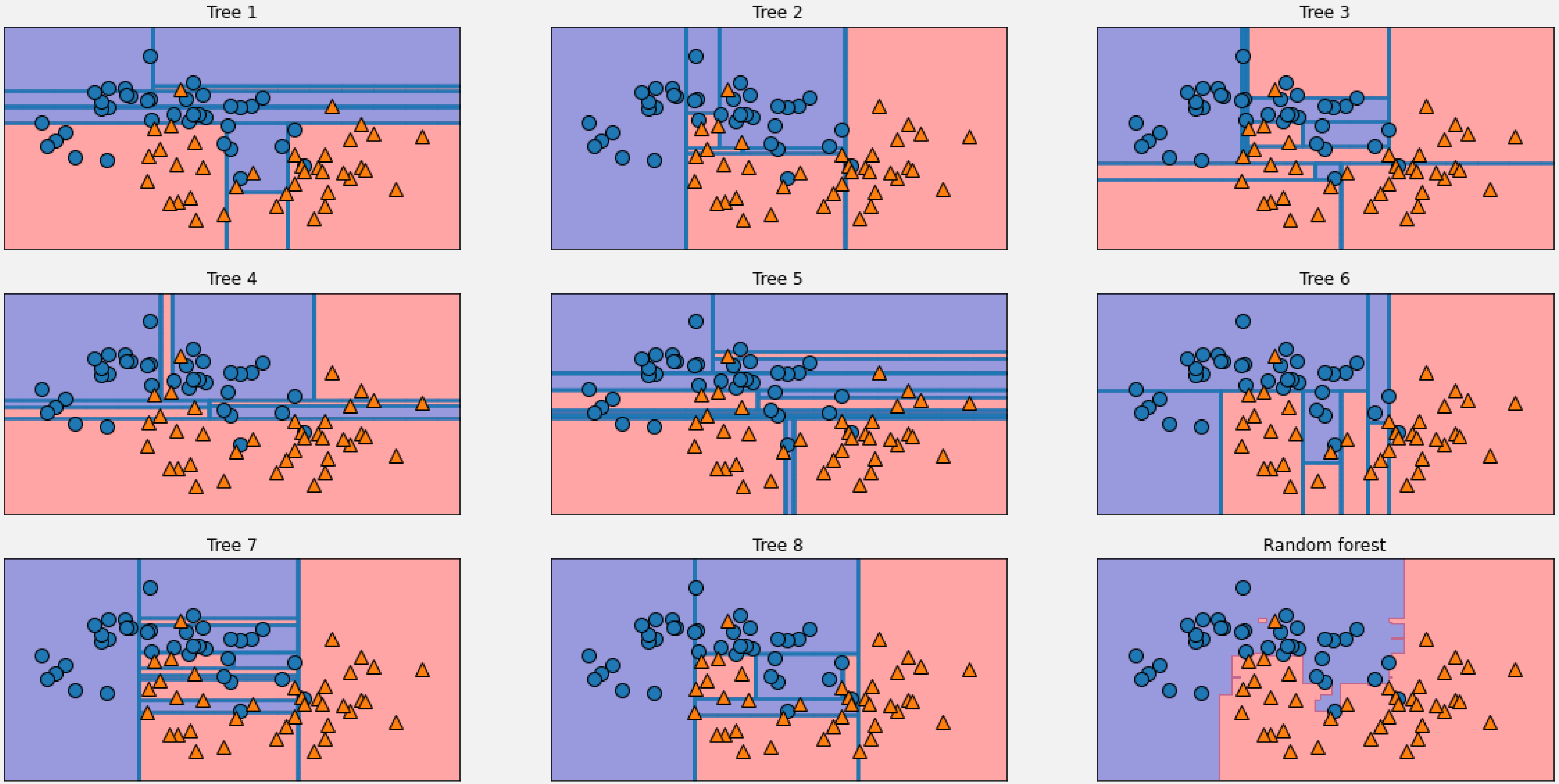
Cons

ENSEMBLES OF DECISION TREES

- **Random forests** *(bagging)*

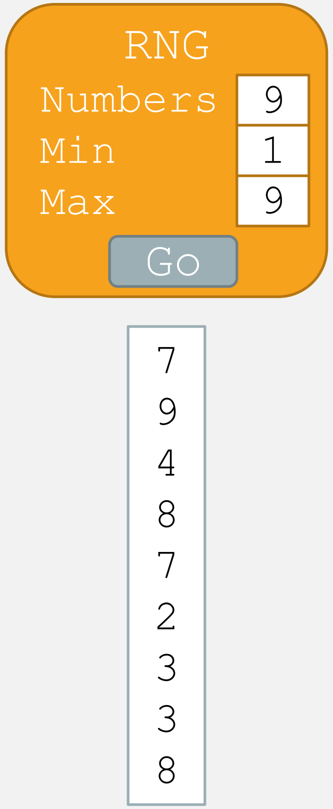
- **Gradient boosted decision trees** *(boosting)*

RANDOM FORESTS



RANDOMIZATION I: BOOTSTRAPPING

	f_1	f_2	f_3	f_4	f_5	f_6
x_1	45	5	21	45	15	1
x_2	87	2	12	44	64	2
x_3	24	8	15	43	36	3
x_4	67	7	17	44	87	2
x_5	13	5	12	44	65	3
x_6	87	4	16	42	34	1
x_7	89	7	13	42	2	2
x_8	68	3	14	43	54	3
x_9	35	6	11	41	63	2



A bootstrap dataset

	f_1	f_2	f_3	f_4	f_5	f_6
x_7	89	7	13	42	2	2
x_9	35	6	11	41	63	2
x_4	67	7	17	44	87	2
x_8	68	3	14	43	54	3
x_7	89	7	13	42	2	2
x_2	87	2	12	44	64	2
x_3	24	8	15	43	36	3
x_3	24	8	15	43	36	3
x_8	68	3	14	43	54	3

RANDOMIZATION 1: BOOTSTRAPPING

Dataset for tree 1

	f_1	f_2	f_3	f_4	f_5	f_6
x_7	89	7	13	42	2	2
x_9	35	6	11	41	63	2
x_4	67	7	17	44	87	2
x_8	68	3	14	43	54	3
x_7	89	7	13	42	2	2
x_2	87	2	12	44	64	2
x_3	24	8	15	43	36	3
x_3	24	8	15	43	36	3
x_8	68	3	14	43	54	3

Dataset for tree 2

	f_1	f_2	f_3	f_4	f_5	f_6
x_6	87	4	16	42	34	1
x_8	68	3	14	43	54	3
x_2	87	2	12	44	64	2
x_2	87	2	12	44	64	2
x_3	24	8	15	43	36	3
x_7	89	7	13	42	2	2
x_4	67	7	17	44	87	2
x_2	87	2	12	44	64	2
x_8	68	3	14	43	54	3

Dataset for tree 3

	f_1	f_2	f_3	f_4	f_5	f_6
x_3	24	8	15	43	36	3
x_3	24	8	15	43	36	3
x_8	68	3	14	43	54	3
x_7	89	7	13	42	2	2
x_1	45	5	21	45	15	1
x_1	45	5	21	45	15	1
x_6	87	4	16	42	34	1
x_5	13	5	12	44	65	3
x_7	89	7	13	42	2	2

RANDOMIZATION II: FEATURE SELECTION

Dataset for tree 1

	f_1	f_2	f_3	f_4	f_5	f_6
x_7	89	7	13	42	2	2
x_9	35	6	11	41	63	2
x_4	67	7	17	44	87	2
x_8	68	3	14	43	54	3
x_7	89	7	13	42	2	2
x_2	87	2	12	44	64	2
x_3	24	8	15	43	36	3
x_3	24	8	15	43	36	3
x_8	68	3	14	43	54	3

For each node, randomly
select a of features
and ask the question
involving

RANDOMIZATION II: FEATURE SELECTION

Dataset for tree 1

	f_1	f_2	f_3	f_4	f_5	f_6
x_7	89	7	13	42	2	2
x_9	35	6	11	41	63	2
x_4	67	7	17	44	87	2
x_8	68	3	14	43	54	3
x_7	89	7	13	42	2	2
x_2	87	2	12	44	64	2
x_3	24	8	15	43	36	3
x_3	24	8	15	43	36	3
x_8	68	3	14	43	54	3

`max_features`

`max_features = n_features`

`max_features = 1`

RANDOMIZATION II: FEATURE SELECTION

Dataset for tree 1

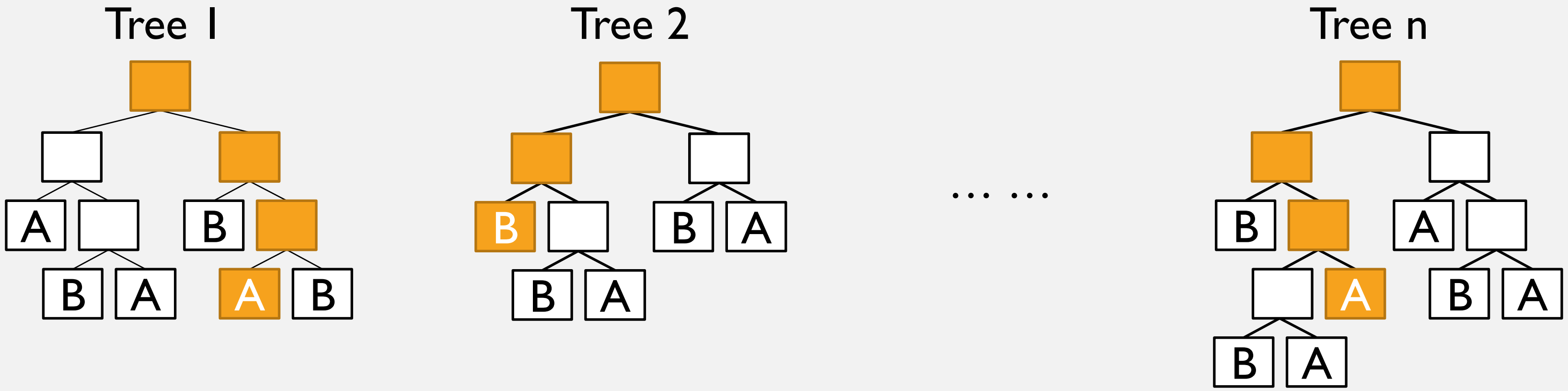
	f_1	f_2	f_3	f_4	f_5	f_6
x_7	89	7	13	42	2	2
x_9	35	6	11	41	63	2
x_4	67	7	17	44	87	2
x_8	68	3	14	43	54	3
x_7	89	7	13	42	2	2
x_2	87	2	12	44	64	2
x_3	24	8	15	43	36	3
x_3	24	8	15	43	36	3
x_8	68	3	14	43	54	3

A low value of `max_features`

A high value of `max_features`

A rule of thumb

PREDICTIONS USING RANDOM FORESTS



PROS AND CONS OF RANDOM FORESTS

Pros

Cons

TREES VS. FORESTS

Jupyter Notebook **Decision Trees 2: Feature importance and ensembles of trees**

GRADIENT BOOSTED DECISION TREES

OR GRADIENT BOOSTED REGRESSION TREES *OR* GRADIENT BOOSTING MACHINES

HYPERPARAMETERS

`n_estimators`

`max_depth`

`learning_rate`

CODING BOOSTED TREES

Jupyter Notebook **Decision Trees 2: Feature importance and ensembles of trees**

PROS AND CONS OF GRADIENT BOOSTED DECISION TREES

Pros

Cons

WHEN TO USE WHAT

Tree

Forest

Boosted tree

WHERE DOES A DATA SCIENTIST CAMP?
IN A RANDOM FOREST