SH Dr. Soufiane Hourri

drsoufiane.com

DUT · SEMESTER 3 · COMPUTER SCIENCE

Machine Learning

Session 05 · Decision trees: learning rules from data

Dr. Soufiane Hourri, PhD, HDR

Associate Professor & AI Consultant

École Supérieure de Technologie de Safi · Cadi Ayyad University

2026-2027

SESSION 05 · OVERVIEW

Today's session

01 How a tree makes a decision

02 How a tree chooses its questions

03 Overfitting, depth and cross-validation

04 Trees in practice

05 Lab: trees on the penguins

TODAY'S FOCUS

A tree learns a set of if-then rules. We learn how it picks each rule, and how to stop it from learning too many.

Learning objectives

By the end of this session, you will be able to

1. Read a decision tree and use it to classify a new example
2. Compute the Gini impurity of a node and of a split
3. Explain how a tree chooses its questions, one at a time
4. Recognise overfitting and limit it with max_depth
5. Choose a hyperparameter with cross-validation

PREREQUISITES

- Session 04: proportions, confusion matrix, metrics
- Session 02: pipelines, one-hot encoding
- A calculator

Warm-up from Session 04

QUESTION

A model has precision 0.90 and recall 0.40 for class 1. Which sentence is correct?

A

It finds 90% of the real positives

B

When it predicts 1, it is right 90% of the time

C

Its accuracy is 0.65

D

Lowering the threshold will lower its recall

01

CHAPTER

How a tree makes a decision

A sequence of yes-or-no questions, from the root to a leaf.

A tree is a sequence of questions

DEFINITION

A decision tree classifies an example by asking yes-or-no questions about its features, one after the other, until it reaches a leaf that gives the answer.

EXAMPLE

Is the bill shorter than 42.3 mm? If yes, is it deeper than 15.3 mm? If yes: Adelie.

VOCABULARY

Root: the first question
Node: a question
Branch: an answer, yes or no
Leaf: a final answer
Depth: questions from root to leaf

A tree that recognises penguin species

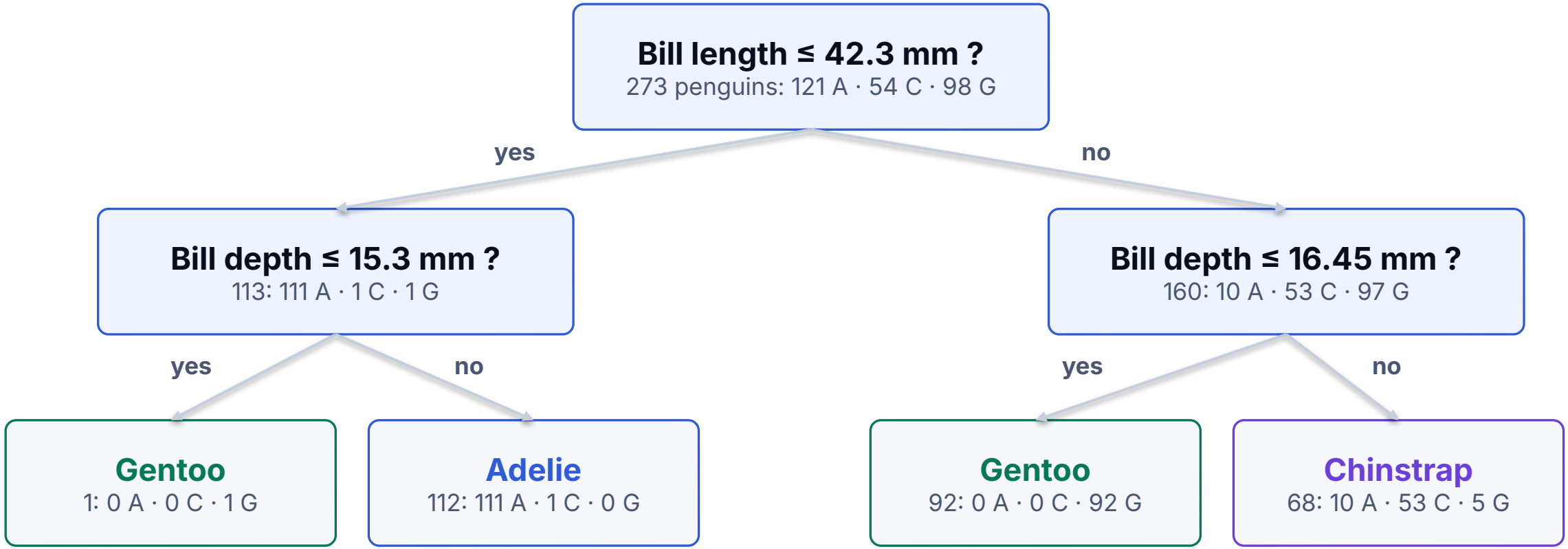


Figure 1. Tree with max_depth = 2 trained on 273 penguins (bill length and depth only). A Adelie, C Chinstrap, G Gentoo. scikit-learn 1.9.1.

Classify three new penguins

PROBLEM

Use the tree of Figure 1 to classify: P1 bill 39.0 × 18.5 mm; P2 bill 48.0 × 18.0 mm; P3 bill 47.0 × 15.0 mm (length × depth).

REASONING

1. P1: 39.0 ≤ 42.3 → left; 18.5 > 15.3 → Adelie leaf
2. P2: 48.0 > 42.3 → right; 18.0 > 16.45 → Chinstrap leaf
3. P3: 47.0 > 42.3 → right; 15.0 ≤ 16.45 → Gentoo leaf
4. Two questions per penguin, always

RESULT

P1: Adelie

P2: Chinstrap (p = 0.78)

P3: Gentoo

A tree cuts the space into rectangles

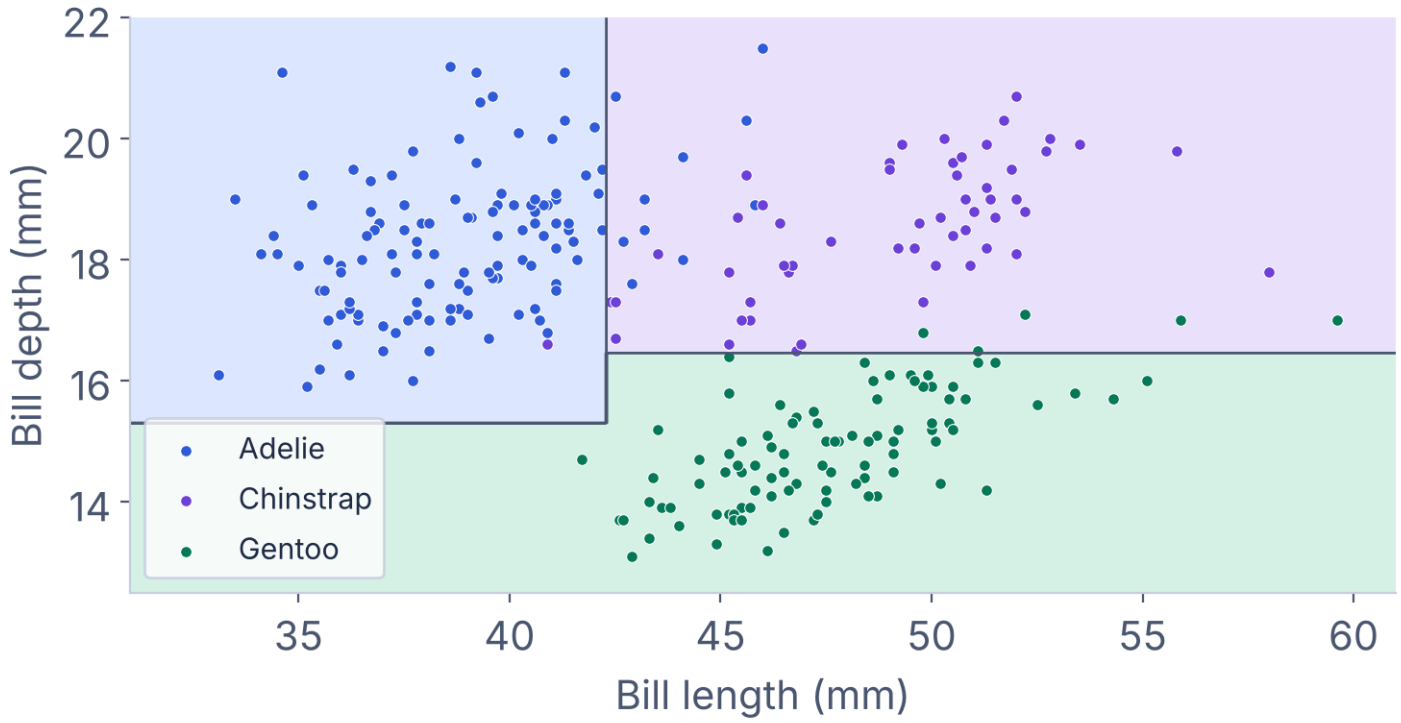


Figure 2. Regions of the tree of Figure 1. Points: the 273 training penguins.

WHAT TO NOTICE

- Each question is a vertical or horizontal cut
- Each leaf is a rectangle
- A few penguins fall in the wrong region

Knowledge check

QUESTION

In the tree of Figure 1, a penguin has a bill of 44.0 × 17.5 mm. What does the tree predict, and with what probability?

A

Adelie, 0.99

B

Chinstrap, 0.78

C

Gentoo, 1.00

D

It cannot decide: 44.0 is close to 42.3

02

CHAPTER

How a tree chooses its questions

At each node, the question that makes the groups as pure as possible.

A good question creates pure groups

DEFINITION

A group is pure when all its examples belong to the same class. The impurity measures how mixed a group is: 0 for a pure group, larger for a mixed one.

EXAMPLE

The Gentoo leaf (92 G) is pure. The root (121 A, 54 C, 98 G) is very mixed. A good question splits a mixed group into purer ones.

THE RULE

Try every possible question.
Keep the one whose groups are the purest.
Repeat in each group.

The Gini impurity

$$\text{Gini} = 1 - (p_1^2 + p_2^2 + \ldots + p_k^2)$$

SYMBOLS

p_k : proportion of class k in the group

Pure group: Gini = 0

Two classes, 50/50: Gini = 0.5 (the maximum)

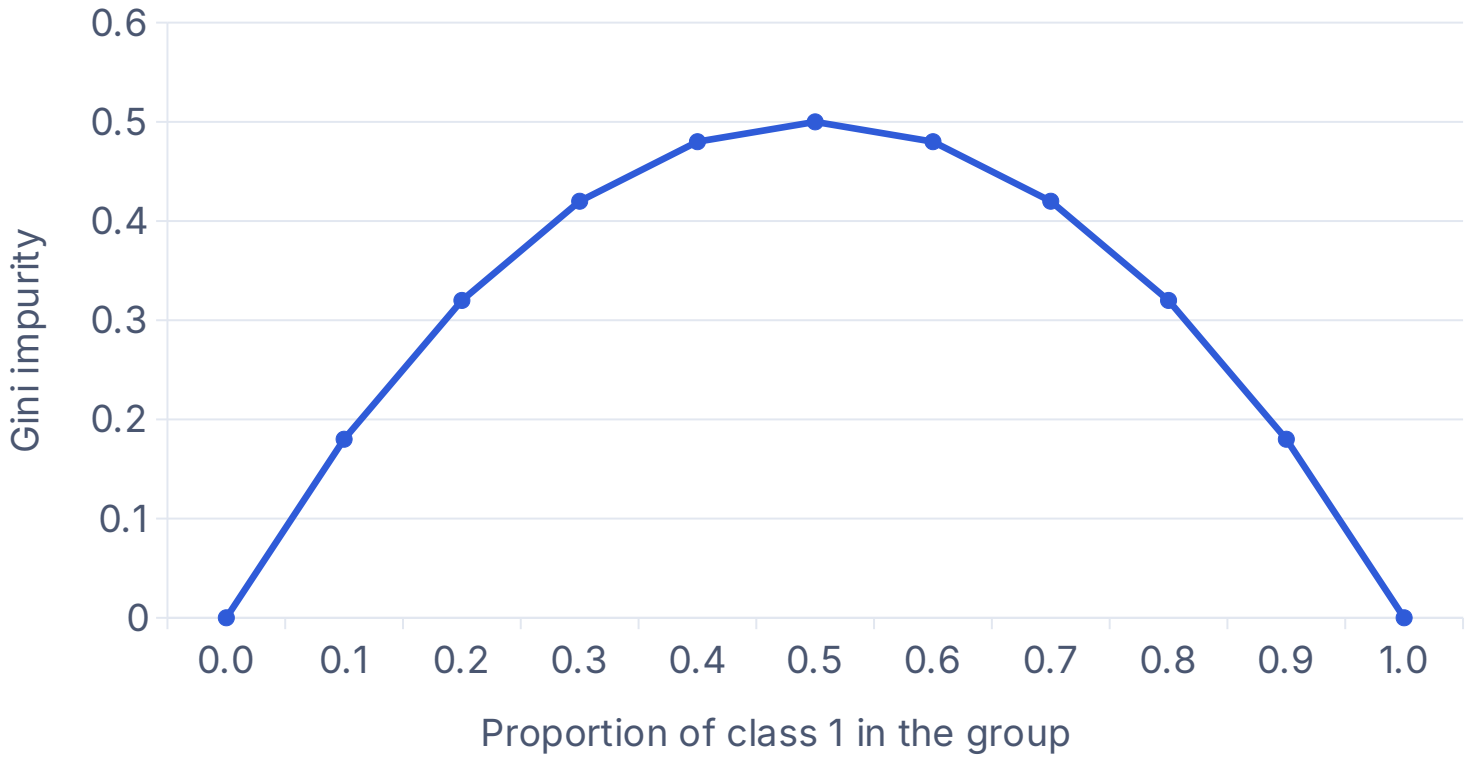
EXAMPLES

10 A, 0 G: $1 - (1^2 + 0^2) = 0$

5 A, 5 G: $1 - (0.5^2 + 0.5^2) = 0.5$

8 A, 2 G: $1 - (0.8^2 + 0.2^2) = 0.32$

Gini for two classes



Gini = $1 - (p^2 + (1 - p)^2) = 2p(1 - p)$ for two classes.

MAIN FINDING

Gini is 0 when the group is pure (p = 0 or p = 1) and largest at p = 0.5.

Purer groups give lower values.

Gini of a split: sex on the Titanic

PROBLEM

Titanic (Session 04): 549 died, 342 survived. Women 81 died, 233 survived; men 468 died, 109 survived. Compute the Gini before and after splitting on sex.

REASONING

1. Root: $1 - (0.616^2 + 0.384^2) = 0.473$
2. Women (314): $1 - (0.258^2 + 0.742^2) = 0.383$
3. Men (577): $1 - (0.811^2 + 0.189^2) = 0.306$
4. Weighted: $(314 \times 0.383 + 577 \times 0.306) / 891$

RESULT

After the split: 0.333

Decrease: 0.473 − 0.333 = 0.140

Which question is better?

PROBLEM

Second candidate: "Did the passenger travel in third class?" Third class: 372 died, 119 survived (491). Other classes: 177 died, 223 survived (400). Compare with the sex split.

REASONING

1. Third (491): $1 - (0.758^2 + 0.242^2) = 0.367$
2. Others (400): $1 - (0.443^2 + 0.558^2) = 0.493$
3. Weighted: $(491 \times 0.367 + 400 \times 0.493) / 891 = 0.424$
4. Decrease: $0.473 - 0.424 = 0.049$

RESULT

Sex: 0.140

Third class: 0.049

The tree asks about sex first

Numerical features: choosing a threshold

Penguin	1	2	3	4	5	6	7	8
Flipper (mm)	181	190	195	197	209	210	215	222
Species	Adelie	Adelie	Chinstrap	Adelie	Gentoo	Chinstrap	Gentoo	Gentoo
Gentoo?	no	no	no	no	yes	no	yes	yes

Table 1. Eight penguins from penguins.csv, sorted by flipper length. Task: separate Gentoo from the others.

Two thresholds, two Gini values

PROBLEM

Root: 5 not Gentoo, 3 Gentoo, Gini 0.469. Compare the thresholds flipper ≤ 203 mm and flipper ≤ 209.5 mm.

REASONING

1. ≤ 203: left 4 not G (Gini 0); right 3 G + 1 not (Gini 0.375)
2. Weighted: (4 × 0 + 4 × 0.375) / 8 = 0.188
3. ≤ 209.5: left 4 not + 1 G (0.32); right 2 G + 1 not (0.444)
4. Weighted: (5 × 0.32 + 3 × 0.444) / 8 = 0.367

RESULT

≤ 203 mm wins: 0.188

≤ 209.5 mm: 0.367

Penguin 6 stays misclassified

Building a tree: the algorithm

INPUT

Training examples with features and a class

OUTPUT

A tree of yes-or-no questions with a class in each leaf

KEY IDEA

Greedy: the best question now, without looking ahead

1. Start with all training examples in the root
2. For every feature and every threshold, compute the weighted Gini of the split
3. Keep the question with the lowest weighted Gini
4. Send each example to the left or right child
5. Repeat steps 2 to 4 in each child
6. Stop when a node is pure or a limit is reached (max_depth, min_samples_leaf)
7. Each leaf predicts its majority class

Another impurity: entropy

$$\text{Entropy} = -(p_1 \log_2 p_1 + \ldots + p_k \log_2 p_k)$$

SYMBOLS

p_k : proportion of class k (terms with p_k = 0 are skipped)

Pure group: entropy = 0

Two classes, 50/50: entropy = 1

IN PRACTICE

criterion="gini" (default) or "entropy" in scikit-learn.

Both usually choose very similar questions.

Knowledge check

QUESTION

A node contains 6 survivors and 2 non-survivors. What is its Gini impurity?

A

0.25

B

0.375

C

0.5

D

0.75

Break - 15 minutes

After the break: what happens when a tree never stops growing?

03

CHAPTER

Overfitting, depth and cross-validation

A tree can learn its training data by heart. We choose how far it may go.

A small tree and a tree with no limit

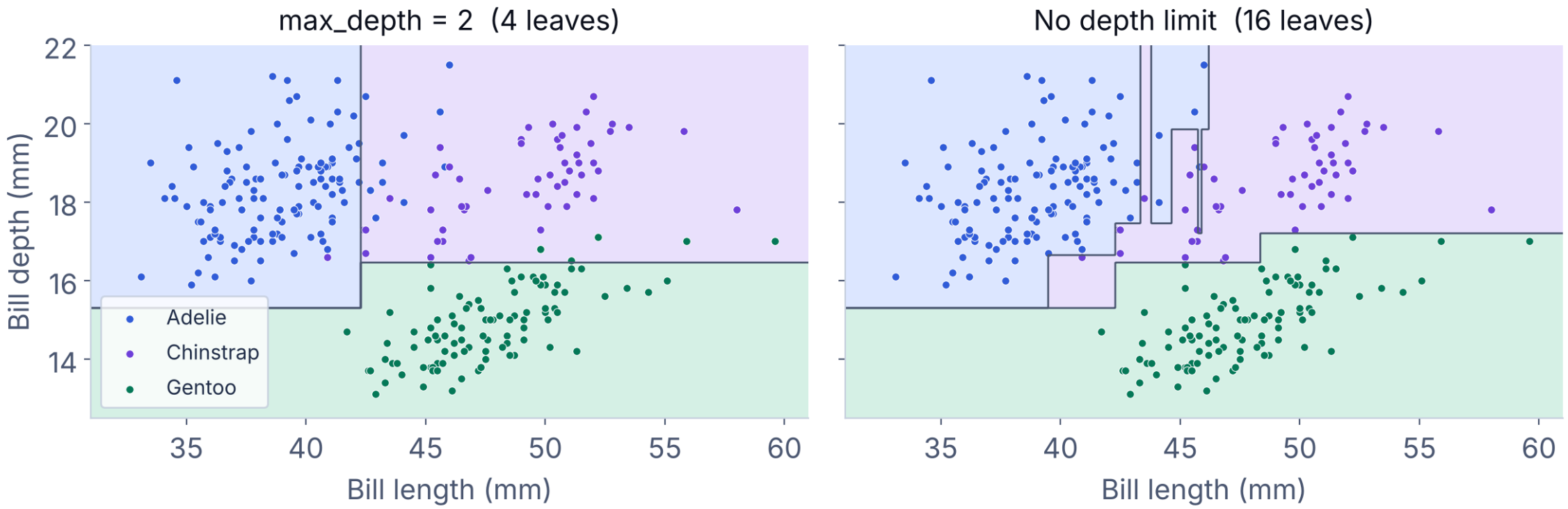
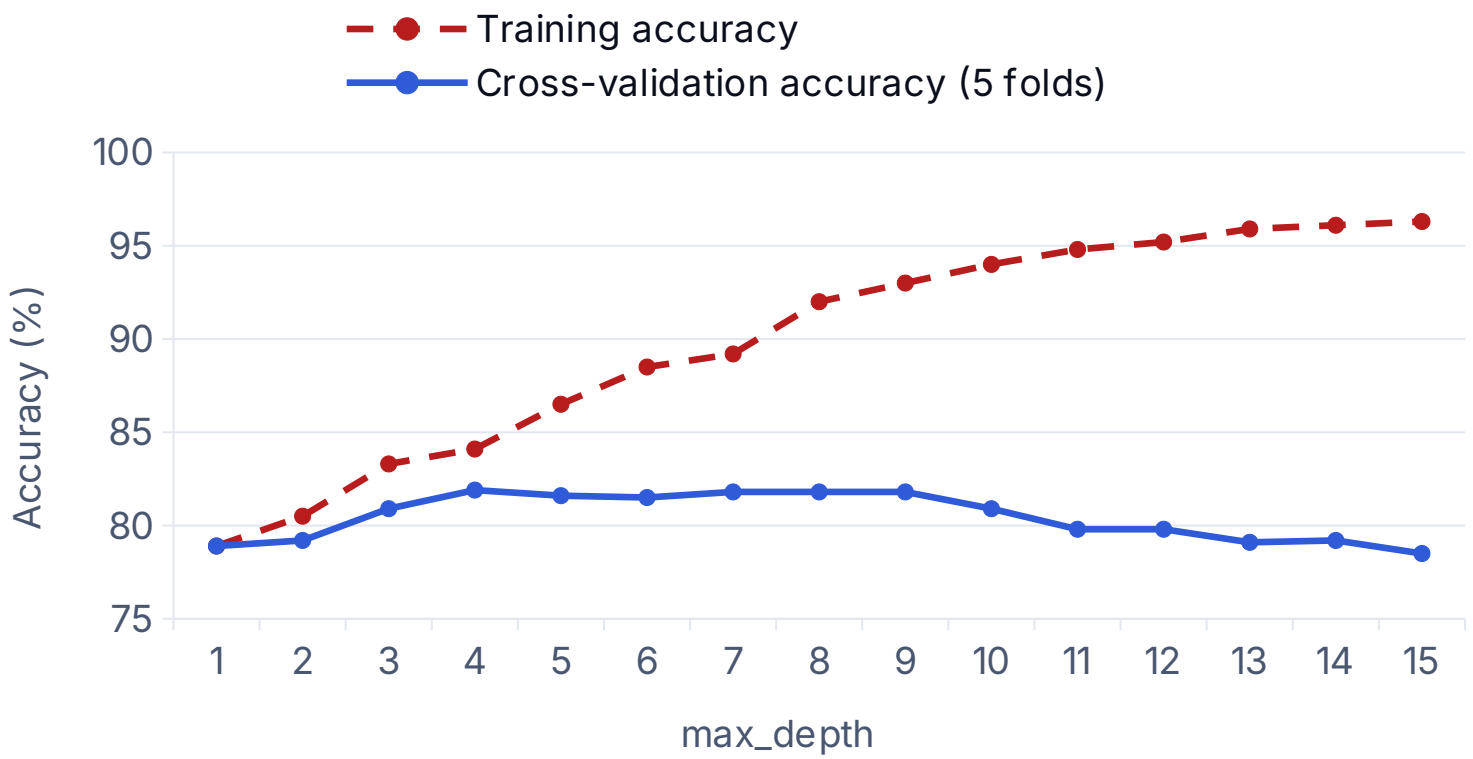


Figure 3. Same 273 training penguins. Training accuracy: 94.1% (left) and 100% (right). Test accuracy: 91.3% and 95.7%.

Deeper trees: training up, validation down



Titanic, 712 training passengers (Session 04 split). Same features as the Session 04 model. scikit-learn 1.9.1.

MAIN FINDING

Training accuracy keeps rising with depth.

Accuracy on unseen data peaks at depth 4 (81.9%), then falls.

The gap between the curves is overfitting.

Hyperparameters that limit a tree

- max_depth: the maximum number of questions from root to leaf
- min_samples_leaf: the minimum number of examples in a leaf
  - Titanic, no limit: 152 leaves, 98.3% on the training set
  - min_samples_leaf=20: 25 leaves, 83.7% on the training set
- These are hyperparameters: we choose them, the tree does not learn them
- Too small: underfitting. Too large: overfitting.

KEY TAKEAWAY

Choosing a hyperparameter means choosing how complex the model may be.

k-fold cross-validation

Training set (712 passengers) split into 5 folds

Round 1	**Validation**	Train	Train	Train	Train	score 1
Round 2	Train	**Validation**	Train	Train	Train	score 2
Round 3	Train	Train	**Validation**	Train	Train	score 3
Round 4	Train	Train	Train	**Validation**	Train	score 4
Round 5	Train	Train	Train	Train	**Validation**	score 5

Result
CV score = mean of the 5 scores
Every passenger is used for validation exactly once
The test set is not touched

Figure 4. 5-fold cross-validation on the training set. The model is trained 5 times; each time one fold is kept aside to measure it.

Choosing max_depth with GridSearchCV

CODE · PYTHON

```python
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier

tree = make_pipeline(prep,
    DecisionTreeClassifier(random_state=42))
grid = GridSearchCV(tree,
    {"decisiontreeclassifier__max_depth": range(1, 16)},
    cv=5)
grid.fit(train[num + cat], train["survived"])
print(grid.best_params_)
print(grid.best_score_)
```

OUTPUT

```
{'decisiontreeclassifier__
    max_depth': 4}
0.819
```

For each depth, 5 trees are trained: 75 fits. Then the best depth is refitted on all 712 passengers.

Choosing the depth on the test set

× INCORRECT APPROACH

"I tried depths 1 to 15 on the test set and kept the best (82.7%, no limit)"

✓ CORRECTION

Choose with cross-validation on the training set (depth 4), then measure once on the test set

WHY IT FAILS

With 179 test passengers, one passenger is 0.56 points. Picking the best of 16 noisy test scores rewards luck, and the test score is no longer honest.

Knowledge check

QUESTION

A tree has 99% accuracy on its training set and 72% in cross-validation. What should you do?

A

Nothing: 99% is excellent

B

Increase max_depth

C

Decrease max_depth or raise min_samples_leaf

D

Choose the depth that maximises the test accuracy

04

CHAPTER

Trees in practice

The Titanic again: reading the rules, comparing with logistic regression.

The rules learned on the Titanic (depth 3)

EXPORT_TEXT OUTPUT

```
|--- sex_female <= 0.50
|   |--- age <= 3.50
|   |   |--- sibsp <= 2.50: survived  [0, 12]
|   |   |--- sibsp >  2.50: died      [2, 1]
|   |--- age >  3.50
|   |   |--- pclass_1 <= 0.50: died    [314, 39]
|   |   |--- pclass_1 >  0.50: died    [58, 33]
|--- sex_female >  0.50
|   |--- pclass_3 <= 0.50
|   |   |--- fare <= 28.86: survived   [6, 48]
|   |   |--- fare >  28.86: survived   [1, 82]
|   |--- pclass_3 >  0.50
|   |   |--- embarked_S <= 0.50: survived [13, 32]
|   |   |--- embarked_S >  0.50: died   [45, 26]
```

HOW TO READ IT

[died, survived] counts in each leaf, training set

Root question: sex, as computed by hand

Boys under 4 with few siblings: 12 of 12 survived

Women in 3rd class: depends on the port

Feature importance

CODE · PYTHON

```python
best = grid.best_estimator_
names = best[0].get_feature_names_out()
imp = best[-1].feature_importances_

pd.Series(imp, index=names)
  .sort_values(ascending=False)
  .head(5)
```

OUTPUT

```
sex_female   0.576
pclass_3     0.144
age          0.115
fare         0.063
pclass_1     0.052
```

Rounded; cat__ and num__ prefixes removed. Share of the total Gini decrease brought by each feature (depth-4 tree). The values add up to 1.

Tree or logistic regression?

Model (Titanic test set, 179 passengers)	Accuracy	Precision	Recall	F1
Baseline: majority class	0.615	undefined	0.000	undefined
Logistic regression (Session 04)	0.804	0.793	0.667	0.724
Decision tree, max_depth = 4 (CV)	0.793	0.864	0.551	0.673

Table 2. Same split and features as Session 04. Tree confusion matrix: TN 104, FP 6, FN 31, TP 38. The baseline never predicts survival, so its precision is 0 / 0.

Strengths and weaknesses of trees

Strengths

Easy to read and explain as rules

No scaling needed; numbers and categories mixed

Captures interactions and steps

Weaknesses

Overfits without limits

Unstable: small data changes, different tree

Steps only: cannot extrapolate

What comes next

Control depth with cross-validation

Many trees together are more stable

Always compare with a baseline

05

LAB

Lab: trees on the penguins

Species first, then sex: does a tree beat Session 04's logistic regression?

LAB · GUIDED CODE

Lab · Part A: species with a small tree

CODE · PYTHON

```python
from sklearn.tree import (
    DecisionTreeClassifier, export_text)
df = pd.read_csv("penguins.csv")
df = df.dropna(subset=["bill_length_mm"])
X = ["bill_length_mm", "bill_depth_mm",
     "flipper_length_mm", "body_mass_g"]
tr, te = train_test_split(df, test_size=0.2,
    random_state=42, stratify=df["species"])
tree = DecisionTreeClassifier(
    max_depth=2, random_state=42)
tree.fit(tr[X], tr["species"])
print(tree.score(te[X], te["species"]))
print(export_text(tree, feature_names=X))
```

OUTPUT

```
0.986

|--- flipper_length_mm <= 206.50
|   |--- bill_length_mm <= 43.35
|   |   |--- class: Adelie
|   |--- bill_length_mm >  43.35
|   |   |--- class: Chinstrap
|--- flipper_length_mm >  206.50
|   |--- bill_depth_mm <= 17.55
|   |   |--- class: Gentoo
|   |--- bill_depth_mm >  17.55
|   |   |--- class: Chinstrap
```

LAB · EXERCISE

Lab · Questions

GOAL

Read and evaluate trees; choose a depth with cross-validation; compare with Session 04.

TIME

35 min

QUESTIONS (PART A: SPECIES; PART B: SEX)

1. A. 3 × 3 confusion matrix: which penguin is wrong?
2. A. Classify two new penguins by hand, then with predict
3. B. Baseline, then a tree with no limit: train vs test
4. B. GridSearchCV on max_depth from 1 to 10
5. B. Best tree on the test set vs Session 04 (0.910)
6. B. Which features does the tree use most?

DELIVERABLE

Your notebook, one sentence per question.

Exit question

QUESTION

Why do we choose max_depth with cross-validation rather than with the test set?

A

Cross-validation is faster

B

The test set must stay unseen for an honest final score

C

The test set is too large

D

Trees cannot be evaluated on a test set

Recap

KEY TAKEAWAYS

- A tree classifies with yes-or-no questions; each leaf predicts its majority class
- Gini = $1 - \Sigma\, p^2$: 0 for a pure group
- Each question minimises the weighted Gini (greedy)
- Without limits, a tree overfits: control max_depth, min_samples_leaf
- Choose hyperparameters with k-fold cross-validation, never on the test set

NEXT

- Practice (Friday): finish the lab
- Read: ISLP, section 8.1
- Bring a calculator next time

References

(1) James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). An Introduction to Statistical Learning, with Applications in Python, ch. 8.1 (Decision trees) and 5.1 (Cross-validation). Springer. statlearning.com

(2) Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and Regression Trees. Wadsworth.

(3) scikit-learn 1.9 User Guide: Decision Trees; Cross-validation; Tuning the hyper-parameters of an estimator. scikit-learn.org/stable

(4) Horst, A. M., Hill, A. P., & Gorman, K. B. (2020). palmerpenguins: Palmer Archipelago (Antarctica) penguin data. doi.org/10.5281/zenodo.3960218

(5) Titanic passenger data, as distributed with seaborn (seaborn-data repository, "titanic"). github.com/mwaskom/seaborn-data

Course page
learn.drsoufiane.com

SH

Thank you · Questions?

Finish the lab notebook for Friday's practice.

Dr. Soufiane Hourri, PhD, HDR

drsoufiane.com