

# Statistical Modeling: Classification

Advanced Data Analysis with Python - Spring Semester 2026

---

Cecilia Graiff

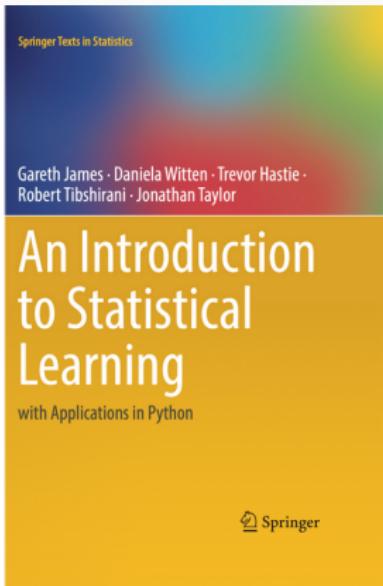
February 26, 2026

ALMAnaCH, Inria Paris - École d'Affaires Publiques, SciencesPo Paris  
[cecilia.graiff@sciencespo.fr](mailto:cecilia.graiff@sciencespo.fr)

# Table of contents

1. Homework Correction
2. Final Project Information
3. Statistical Modeling Foundations
4. Classification: Logistic Regression
5. Tree-based classification methods
6. Evaluation of the classification task

# Resources for this class



- This class follows **Chapters 4 and 8** of this book
- Refer to the book if you want to deepen today's topic
- It is a bit more mathematically intensive than this class!

[An Introduction to Statistical Learning](#)

## Homework Correction

---

## **Final Project Information**

---

# What is expected from you

## Project Description

**Due:** November 3, 2025, 23:59

**What to upload:** PDF with project proposal (2-3 research questions, planned pipeline) and group members' names

## Final Project

**Due:** December 20, 2025, 23:59

**What to upload:** Complete project report in form of a paper (5-8 pages), commented code

**Deadline cannot be delayed!**

# Resources for scientific writing

- Writing a scientific paper (ETH Zurich, 2019)
- Writing a scientific article: A step-by-step guide for beginners
- How to write your first research paper (NIH 2011)
- ... and many others!

# Project checklist

- **Code:**

- Comment each function to explain to me what it does
- Upload the code on **GitHub** and share the repository with me
- Document your repository structure in the **README** file

If you do not know how to use GitHub, you can refer to the guide I uploaded on Moodle.

# Project checklist

- **Paper:**

- 5-8 pages in English
- Structured as a **research paper**:
  - **Introduction**: introduce your research questions (RQs) and their motivation
  - **Related Work**: ground in the literature your choice of RQs and methods
  - **Methods**: explain your pipeline and how you implemented it
  - **Discussion**: present your results (e.g. in form of graphs or tables) and interpret them **qualitatively** and **quantitatively**
  - **Conclusion**: sum up your work

## Statistical Modeling Foundations

---

# What is a statistical model?

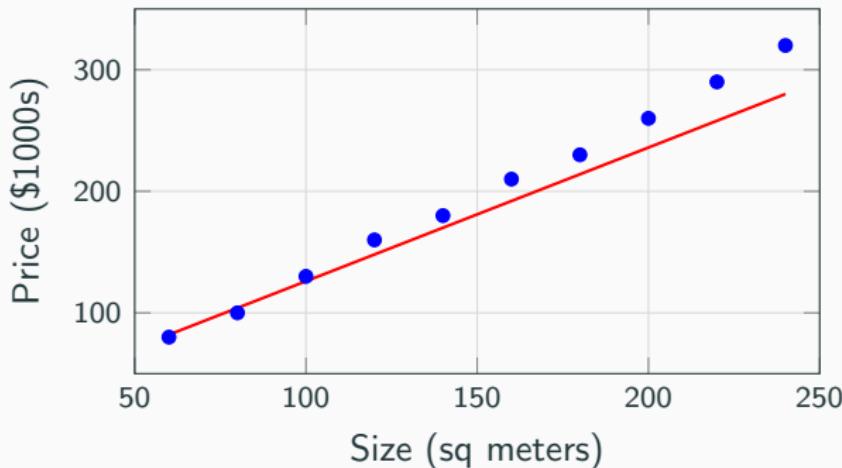
## Definition (Statistical Model)

A statistical model is a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data (and similar data from a larger population).

Source: [Wikipedia](#)

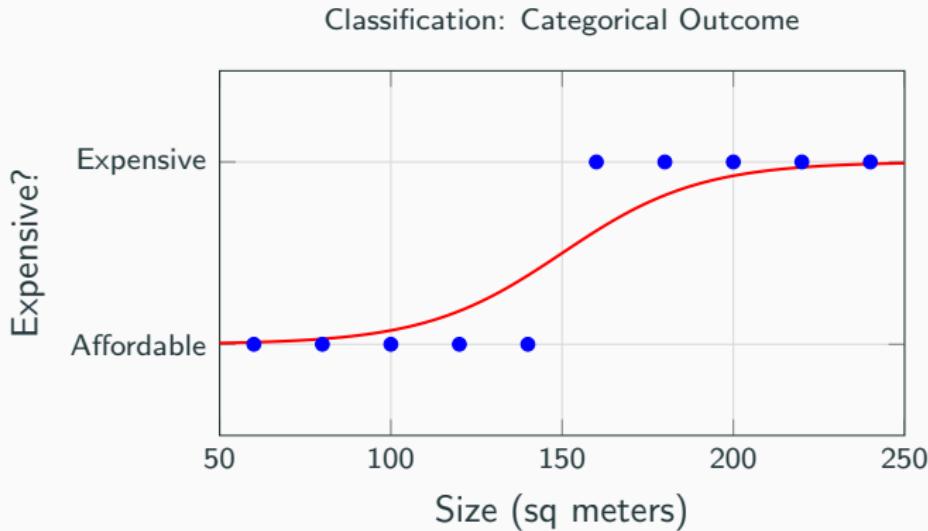
# Linear Regression Example: Predicting House Prices

Linear Regression: Continuous Outcome



- Continuous outcome: house price (\$1000s)
- Red line: regression line through the data
- Blue points: actual house prices (data points)

# Classification Example: Predicting Expensive Houses



- Categorical outcome: affordable vs expensive
- Red curve: probability of being expensive
- Blue points: observed house labels

# Classification: Logistic Regression

---

# Classification problem

## Definition

**Classification** involves assigning a label to a set of data (**categorical variables**).

- Example:

$EmploymentSector \in \{"Healthcare", "Hospitality", "Manufacturing"\}$

# Difference with Linear Regression

## Why not linear regression?

Is it possible to map the variables to integers and perform linear regression?

# Difference with Linear Regression

## Why not linear regression?

Is it possible to map the variables to integers and perform linear regression?

- Suppose we have a **binary classification question**: "Is this person in favour of the adoption of a new policy for public transportation?"

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes} \end{cases}$$

$$\bar{Y} > 0.5 = \text{Yes}$$

# Difference with Linear Regression

## Why not linear regression?

Is it possible to map the variables to integers and perform linear regression?

- Suppose we have a **binary classification question**: "Is this person in favour of the adoption of a new policy for public transportation?"

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes} \end{cases}$$

$$\bar{Y} > 0.5 = \text{Yes}$$

Linear regression can work as a **binary classifier**, but it can output probabilities bigger than 0 or smaller than 1.

### **Important:**

In the above mentioned case, you can easily demonstrate that **if you flip the two variables, the output does not change.**

## Difference with Linear Regression

- Multiclass question:

$$EmploymentSector = \begin{cases} 1 & \text{if Healthcare} \\ 2 & \text{if Hospitality} \\ 3 & \text{if Manufacturing} \end{cases}$$

This mapping implies:

- Order between the three variables
- Same relation between healthcare and hospitality and hospitality and manufacturing

**Both assumptions are not necessarily real.**

# Logistic Regression

The logistic regression model is displayed here:

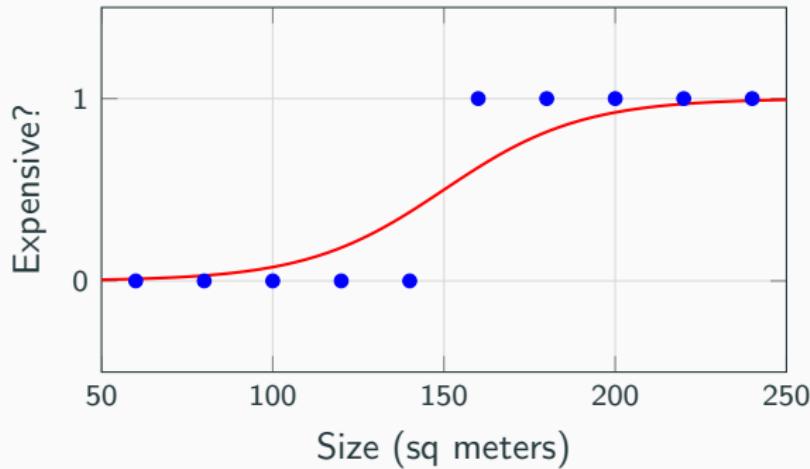
$$\Pr(Y = 1 | X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

where:

- $e = 2.71828$  is a constant (Euler's number)
- $(Y = 1 | X)$  is the conditional probability that  $Y = 1$  given  $X$
- $\beta_0$  is the intercept.
- $\beta_1$  measures the effect of  $X$  on  $Y$ .

It is easy to see that the results will always be between 0 and 1.

## Classification: Categorical Outcome



Logistic regression ensures that the estimated values lie between 0 and 1.

Adapted from: [An Introduction to Statistical Learning](#).

## Estimating the parameters: Maximum Likelihood

- For clarity, let  $p(x_i) = P(Y = 1 \mid X = x_i)$
- The **parameters**  $\beta_0$  and  $\beta_1$  are estimated based on **maximum likelihood**:

$$(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i))$$

## Estimating the parameters: Maximum Likelihood

- For clarity, let  $p(x_i) = P(Y = 1 \mid X = x_i)$
- The **parameters**  $\beta_0$  and  $\beta_1$  are estimated based on **maximum likelihood**:

$$(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i))$$

- Gives **the probability of the observed 0s and 1s** in the data
- $\beta_0$  and  $\beta_1$  picked so that  $\hat{p}(x_i)$  is as close as possible to the actual data

## Estimating the parameters: Maximum Likelihood

- For clarity, let  $p(x_i) = P(Y = 1 | X = x_i)$
- The **parameters**  $\beta_0$  and  $\beta_1$  are estimated based on **maximum likelihood**:

$$(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i))$$

- Gives **the probability of the observed 0s and 1s** in the data
- $\beta_0$  and  $\beta_1$  picked so that  $\hat{p}(x_i)$  is as close as possible to the actual data
  - **Maximizing the likelihood**
  - Example: when predicting if a house will sell (1) or not (0), trying to predict a number as close as possible to 1 if the house was sold, and to 0 if it was not

# Housing Dataset

- Binary outcome: **Sold (Yes/No)**
- Predictor: **House Price (in €100k)**
- Example records:

House Price	Sold
1.2	Yes
2.5	No
1.8	Yes
3.0	No

**Goal:** Predict probability of selling based on house price.

# Logistic Regression Model

## Equation

$$\log \frac{\Pr(Y = 1 | Price)}{1 - \Pr(Y = 1 | Price)} = \beta_0 + \beta_1 \cdot Price$$

- Parameters estimated using Maximum Likelihood

## Estimated Model

$$\hat{P}(Y = 1 | Price) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot Price)}}$$

## Results: Predicted Probabilities

House Price	Observed Sale	Predicted $P(\text{Sales})$
1.2	Yes	0.18
2.5	No	0.47
1.8	Yes	0.28
3.0	No	0.62

- Higher house price = higher predicted probability of selling

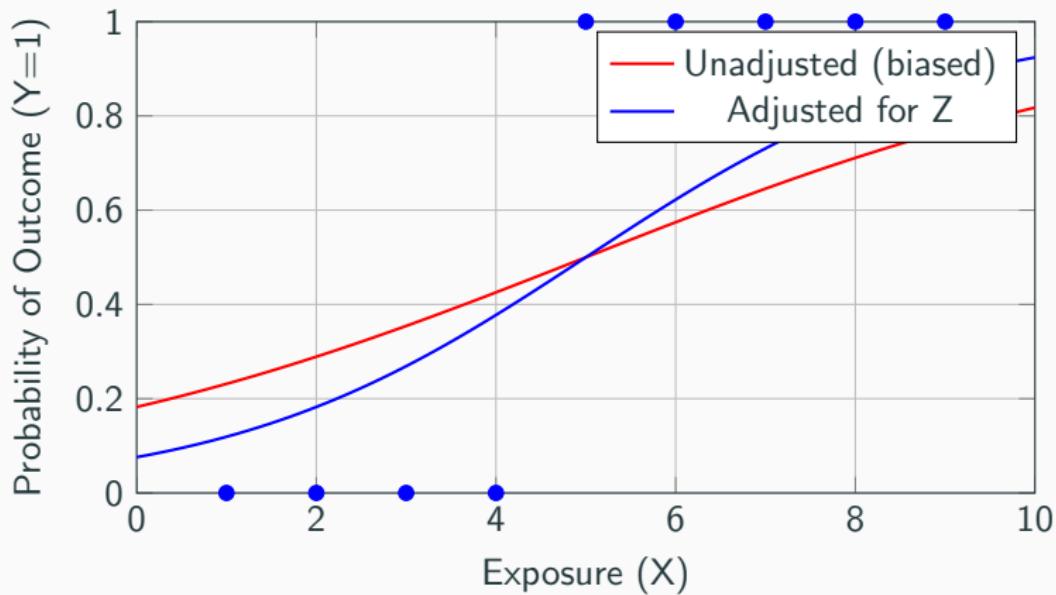
## Caveat: Confounding

**Problem:** We plot the relationship between **credit card balance** and **default** (setting: US). The statistics reveal that students have a higher credit card balances, but for each balance, they tend to default less. This means that depending on the analysis we conduct, we apparently get contradictory results.

- **Student** and **Balance** are correlated
- **Balance** has an effect on the output (**Default**) as well
- Consequence: **confounding**
- (One possible) solution: **Multiple Logistic Regression**

Source of example: [An Introduction to Statistical Learning - Chapter 4](#)

## Effect of Confounder (Z) in Logistic Regression



# Multiple Logistic Regression

- Same as simple, but with more than one predictor:

$$\log \frac{P(Y = 1 | X)}{1 - P(Y = 1 | X)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

# Multinomial Logistic Regression

- Used for **multiclass classification problems**

$$\Pr(Y = k | X) = \frac{e^{\beta_{0k} + \beta_{1k}X_1 + \dots + \beta_{pk}X_p}}{\sum_{j=1}^K e^{\beta_{0j} + \beta_{1j}X_1 + \dots + \beta_{pj}X_p}}$$

- Approach: Model the distribution of  $X$  in each class **separately**, and deduct  $\Pr(Y | X)$

We will only focus on **normal distribution** in this class.

## Binary Example: Housing

We want to predict whether a house will **sell within 30 days** (Yes = 1, No = 0) based on its characteristics:

- Price
- Size
- Bedrooms

**Question:** Given a house with 200000, 3 bedrooms, and 100 m<sup>2</sup>, what is the probability it sells within 30 days?

# Logistic Regression Formula

The probability a house sells is estimated as:

$$P(\text{Sold} = 1) = \frac{e^{\beta_0 + \beta_1 \text{Price} + \beta_2 \text{Size} + \beta_3 \text{Bedrooms}}}{1 + e^{\beta_0 + \beta_1 \text{Price} + \beta_2 \text{Size} + \beta_3 \text{Bedrooms}}}$$

- $\beta_0$ : baseline probability (intercept)
- $\beta_1, \beta_2, \beta_3$ : coefficients showing effect of each feature

## Example Houses and Predicted Probability

House	Price (\$k)	Size (m <sup>2</sup> )	Bedrooms	Probability	Sold	Sold?
A	100	80	2	0.3		No
B	150	100	3	0.6		Yes
C	200	120	4	0.8		Yes

Logistic regression predicts probabilities, not just yes/no.

## When Logistic Regression isn't enough?

1. The classes are well separated
2. The distribution of the predictors  $X$  is approximately normal in each of the classes and the sample size is small

# When Logistic Regression isn't enough?

1. The classes are well separated
2. The distribution of the predictors X is approximately normal in each of the classes and the sample size is small

Instead: **Linear Discriminant Analysis (LDA)**.

- LDA is also more popular **when we have more than 2 response classes.**

# Discriminative and Generative Models

## Discriminative Models

- Model the **conditional distribution**  $p(Y|X)$  directly
- Focus on assigning labels to the data
- Examples: Logistic Regression, SVM, Neural Networks
- Usually better for classification accuracy

## Generative Models

- Model the **joint distribution**  $p(X, Y)$
- Learn  $p(X|Y)$  and  $p(Y)$
- Examples: Naive Bayes, LDA, QDA

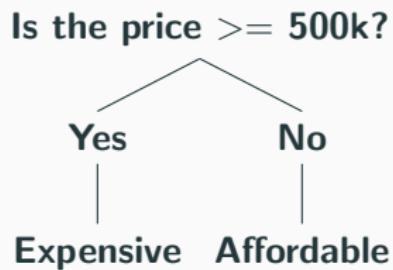
## **Tree-based classification methods**

---

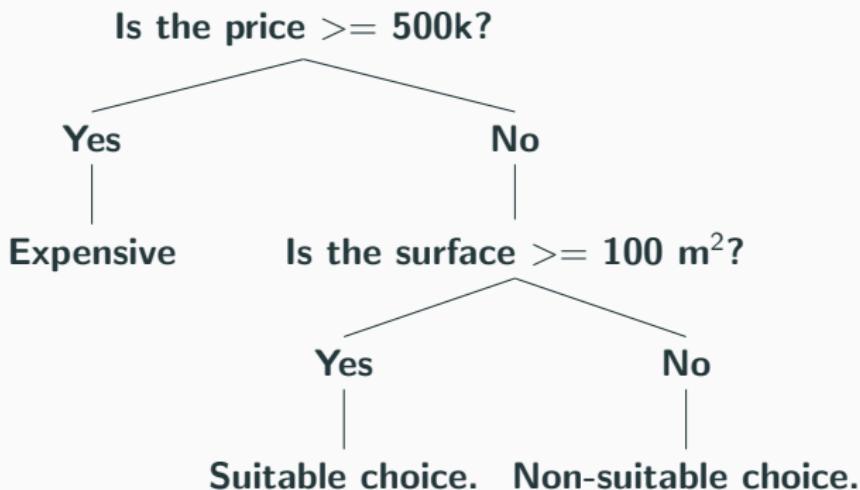
## Decision trees

- **Decision trees** model the classification problem in a tree structure
- They break down the problem into smaller and smaller subsets
- A tree is incrementally built from these smaller subsets
- **Decision nodes** (two branches or more) and **leaf nodes** (last nodes, correspond to classification)
- Uppest node is the **root**
- Applied to both **regression** and **classification**

## Decision trees: Classification example (simplified)

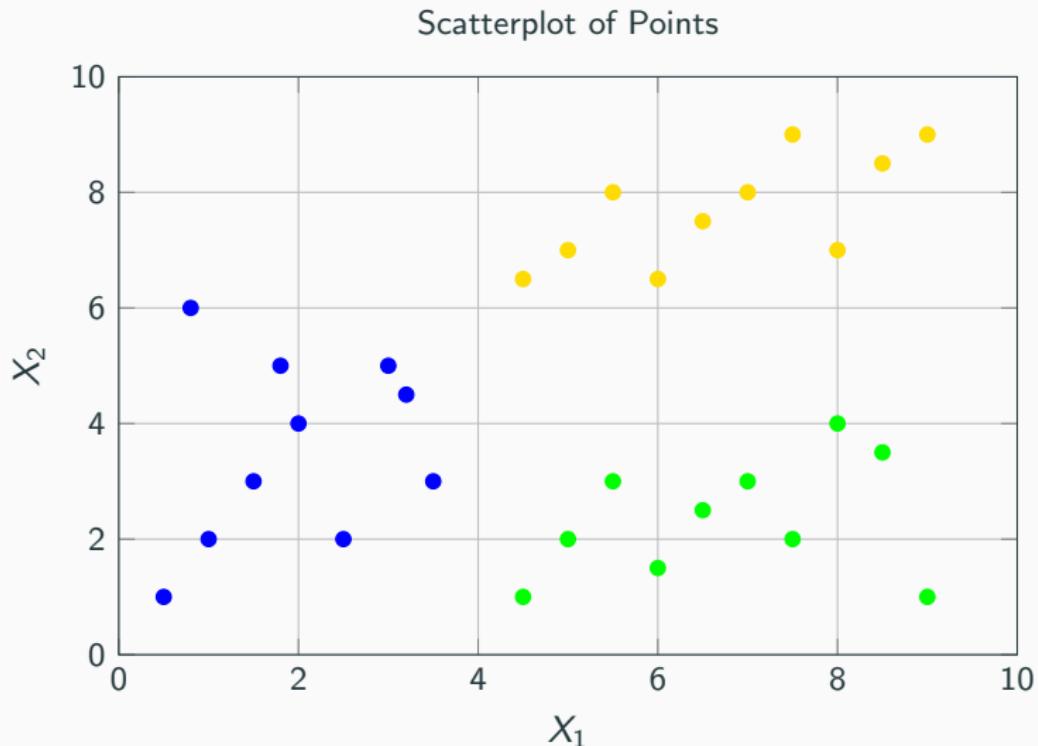


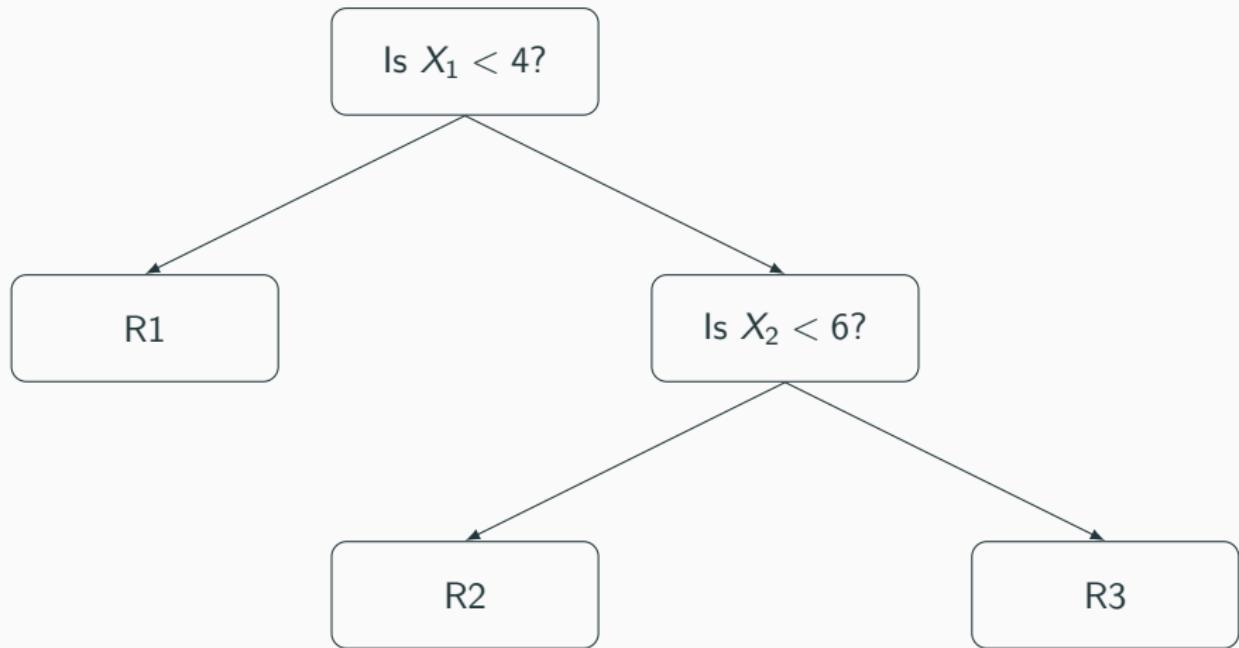
## Decision trees: Classification example (simplified)



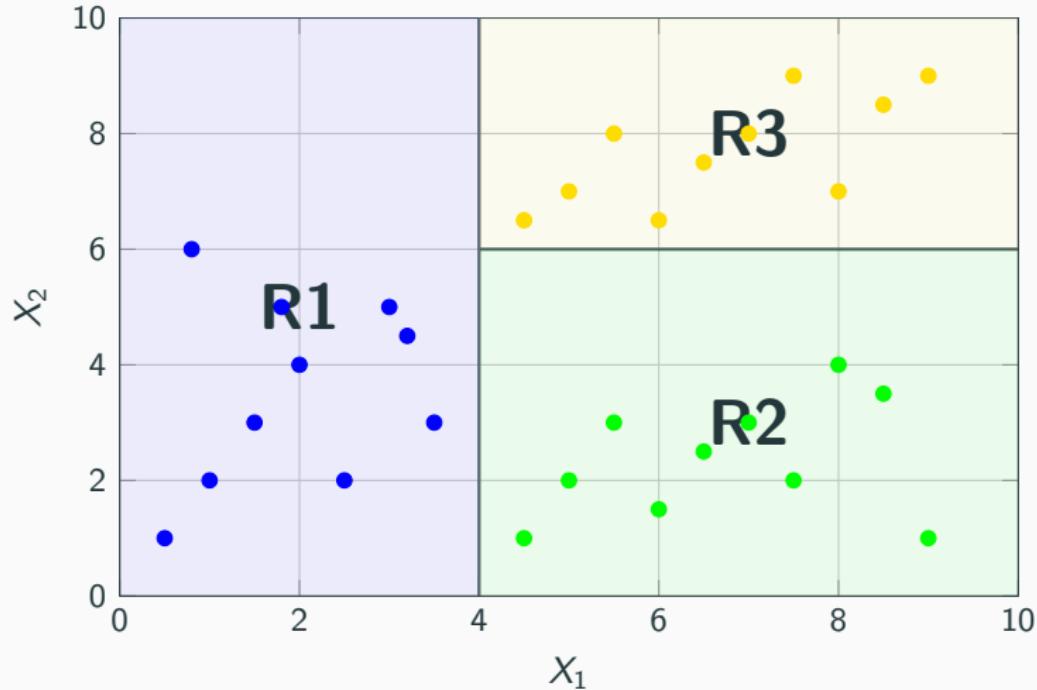
(Clearly not in Paris.)

# Decision trees: Regression Example





Visualization of the "splitting" in different areas done by tree-based methods.



Readapted from: [An Introduction to Statistical Learning](#).

## How to build a decision tree

- Divide the predictor space into  $N$  non-overlapping regions  
 $R_1, R_2, \dots, R_n$

## How to build a decision tree

- Divide the predictor space into  $N$  non-overlapping regions  $R_1, R_2, \dots, R_n$
- Goal: Find the best regions  $R_1$  to  $R_n$  according to an optimization criterium
  - Regression: Minimize RSS
  - Classification: Gini or entropy

# How to build a decision tree

- Divide the predictor space into  $N$  non-overlapping regions  $R_1, R_2, \dots, R_n$
- Goal: Find the best regions  $R_1$  to  $R_n$  according to an optimization criterium
  - Regression: Minimize RSS
  - Classification: Gini or entropy
- Make predictions in leaves
  - Regression: mean of the response values for the training observations in  $R_n$
  - Classification: majority class in the leaf

## Regression tree

- Divide the predictor space into  $N$  non-overlapping regions  
 $R_1, R_2, \dots, R_n$

## Regression tree

- Divide the predictor space into  $N$  non-overlapping regions  $R_1, R_2, \dots, R_n$
- For each observation in region  $R_n$ , make the same predictions: **mean of the response values for the training observations in  $R_n$**

## Regression tree

- Divide the predictor space into  $N$  non-overlapping regions  $R_1, R_2, \dots, R_n$
- For each observation in region  $R_n$ , make the same predictions: **mean of the response values for the training observations in  $R_n$**
- Goal: Find regions  $R_1$  to  $R_n$  so to minimize the Residual Sum of Squares (RSS)

## Regression tree

Problem: it is infeasible to compute RSS for every possible partition of the space in  $n$  regions.

Solution: **top-down, greedy approach.**

- **top-down:** begins **at the top** of the tree and then successively splits the predictor space.
- **greedy:** at each step of the tree-building process, the best split is made **at that particular step.**

# Classification Trees

- Same as regression, but for **qualitative values**

For a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.

- The measures for the binary splits are **Gini index** and **entropy**

Source: [An Introduction to Statistical Learning](#).

# Classification trees

## 1. Gini Index:

Measures the total variance across the classes.

$$G = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk})$$

- $p_i$  = proportion of training observations in the mth region that are from the kth class  $i$
- $K$  = number of classes

# Classification Trees

2. **Entropy:** An alternative measure to Gini.

$$D = - \sum_{k=1}^K \hat{p}_{mk} (\log(\hat{p}_{mk}))$$

Lower Gini or entropy = predominantly observations from a single class (node "purity") = better

## Considerations on decision trees

- **Pro:** easy to compute, represent, and interpret
- **Contra:** accuracy is way lower; oversimplification

The risk of **overfitting** is particularly high. This means that the model performs **well on the training set, but poorly on the test set**. One common approach to reduce overfitting is **pruning**. An alternative is also **random forest**, a learning method that is more accurate and reduces overfitting, but is also more complicated to implement.

## **Evaluation of the classification task**

---

## Confusion Matrix

- Visualization of **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)** and **False Negatives (FN)** values

# Confusion Matrix

- Visualization of **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)** and **False Negatives (FN)** values

		Predicted	Predicted
		Actual	
		Actual	Actual
		4 TN	1 FP
		2 FN	3 TP

# Accuracy

- Ratio between **correctly predicted values** and **all values**.

# Accuracy

- Ratio between **correctly predicted values** and **all values**.
- Accuracy =  $\frac{TP+TN}{TP+TN+FP+FN}$

# Precision

- Ratio between **correctly predicted positive values** and **all predicted positive values**.

# Precision

- Ratio between **correctly predicted positive values** and **all predicted positive values**.
- Precision =  $\frac{TP}{TP+FP}$

## Recall

- Ratio between **correctly predicted positive values** and **all actual positive values**.
- $\text{Recall} = \frac{TP}{TP+FN}$

## F1 score

- F1 Score =  $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

## ROC curve

- **ROC (Receiver Operating Characteristic)**: plots *True Positive Rate* (TPR = recall) vs *False Positive Rate*

- $\text{TPR} = \frac{TP}{TP + FN}$

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

## ROC curve

- **ROC (Receiver Operating Characteristic)**: plots *True Positive Rate* ( $\text{TPR} = \text{recall}$ ) vs *False Positive Rate*

- $\text{TPR} = \frac{TP}{TP + FN}$

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

- **AUC (Area Under Curve)**: probability a randomly selected positive ranks above a randomly selected negative.

## ROC curve

- **ROC (Receiver Operating Characteristic)**: plots *True Positive Rate* ( $\text{TPR} = \text{recall}$ ) vs *False Positive Rate*

- $\text{TPR} = \frac{TP}{TP + FN}$

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

- **AUC (Area Under Curve)**: probability a randomly selected positive ranks above a randomly selected negative.

- **Interpretation:**

- AUC = 0.5: random guessing (diagonal).
- AUC close to 1: perfect!
- AUC < 0.5: predictions inverted.

## ROC curve: example

