Data Science

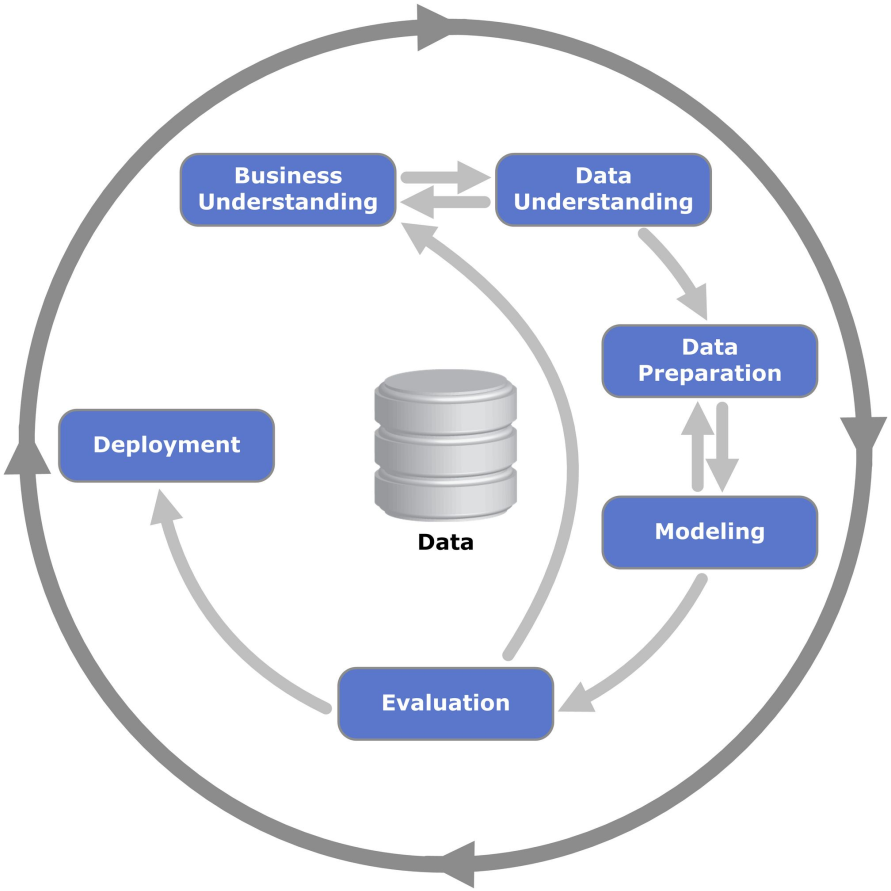
Session 5 - Data imbalance & Deidentification

hadrien.salem@centralelille.fr

introduction-to-data-science

Introduction

What did we do last time?



The CRISP-DM method

Cross-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining

- ➔ Published in 1999
- ➔ Common in the industry
- ➔ Still relevant today

The CRISP-DM method to carry out data-driven projects *(Image source: Wikipedia)*

Course outline

Data science course

Session 1: Understanding data

Session 2: Clean code & Git

Session 3: Preparing data - Cleaning & Missingness

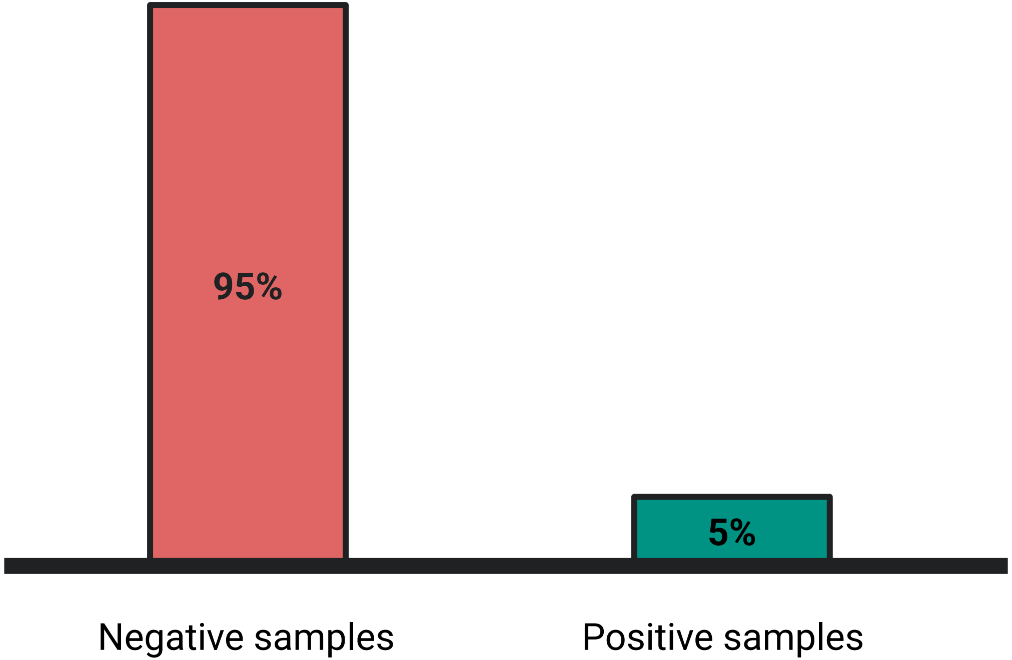
Session 4: Preparing data - Dimensionality reduction

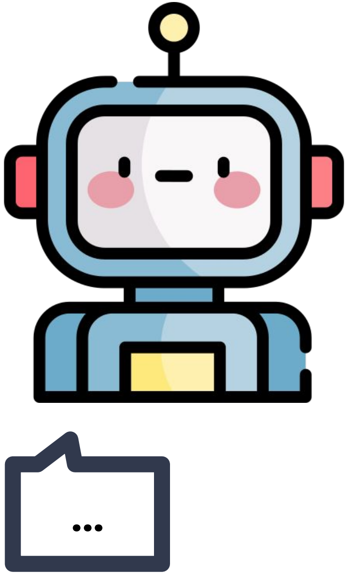
Session 5: Preparing data - Data imbalance

↓

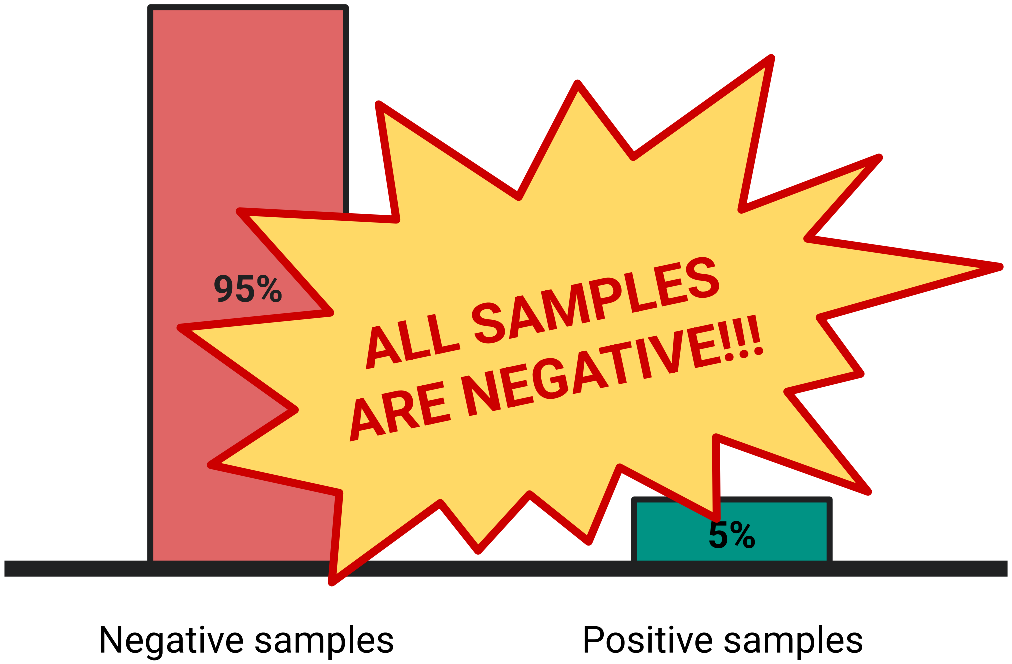
Machine learning course

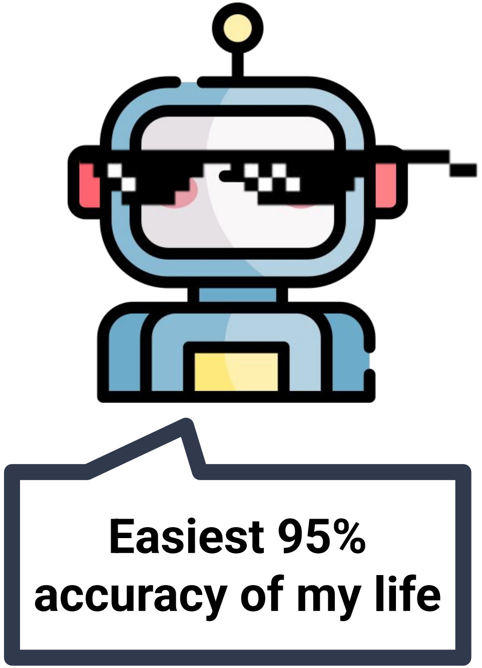
What is class imbalance?

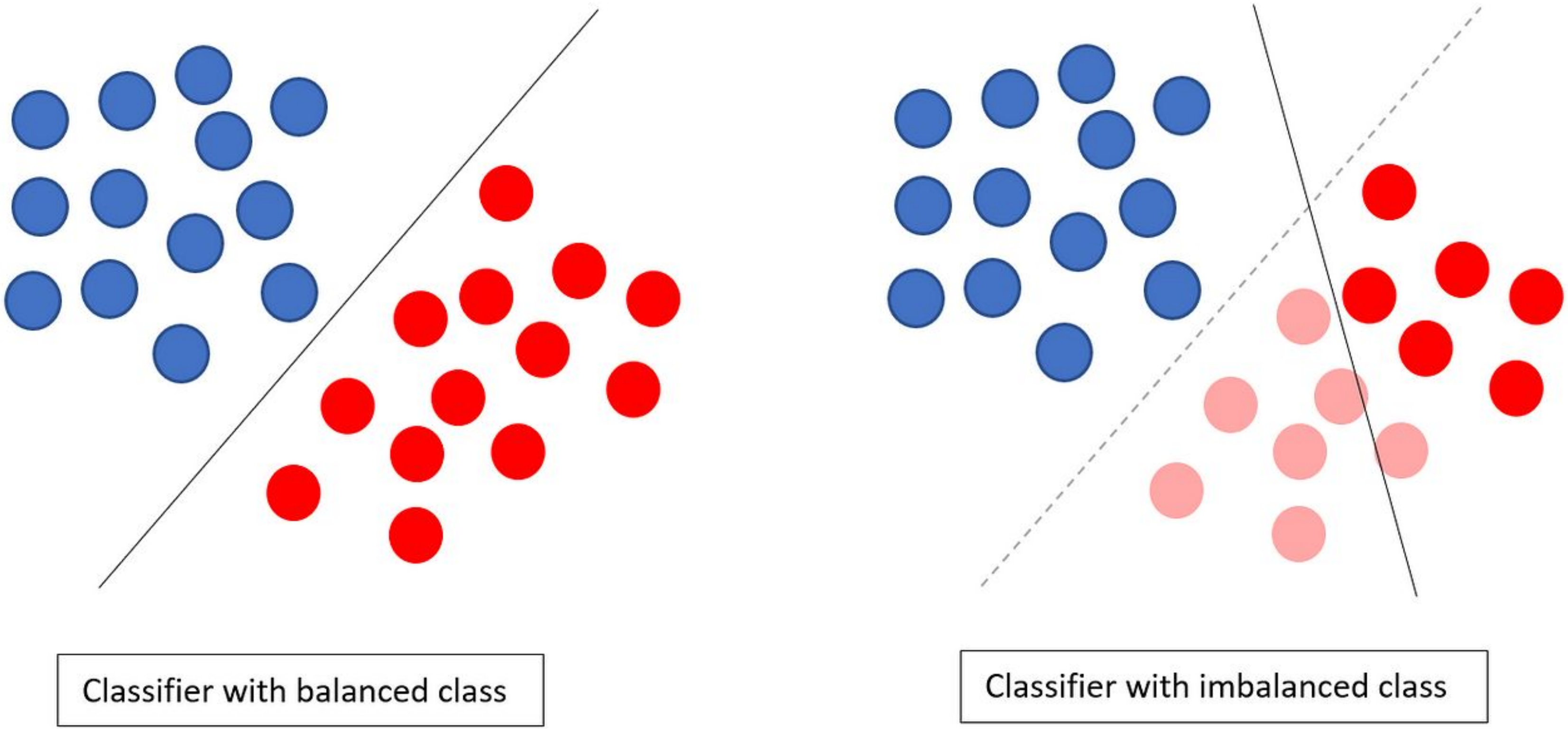




Class imbalance happens when one class has many more instances than the other(s)


95%
ALL SAMPLES ARE NEGATIVE!!!
5%
Negative samples
Positive samples


Easiest 95% accuracy of my life



Class imbalance tends to skew the decision boundary of algorithms

How to deal with class imbalance

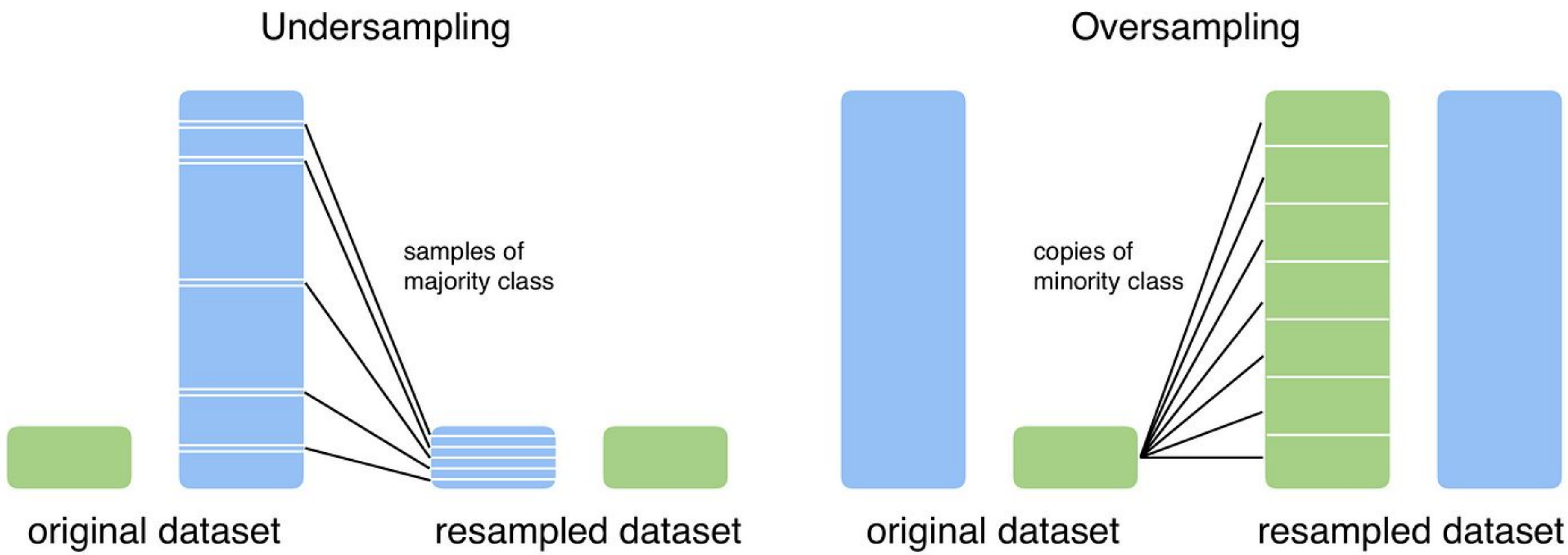
How can you deal with class imbalance?

How can you deal with class imbalance?

There are many methods to deal with class imbalance

- Undersampling your data
- Oversampling your data
- Generating artificial data
- Using imbalance-aware machine learning algorithms

  ⇒ More on that in the ML course



Undersampling and oversampling

Image source

Undersampling and oversampling

Undersampling

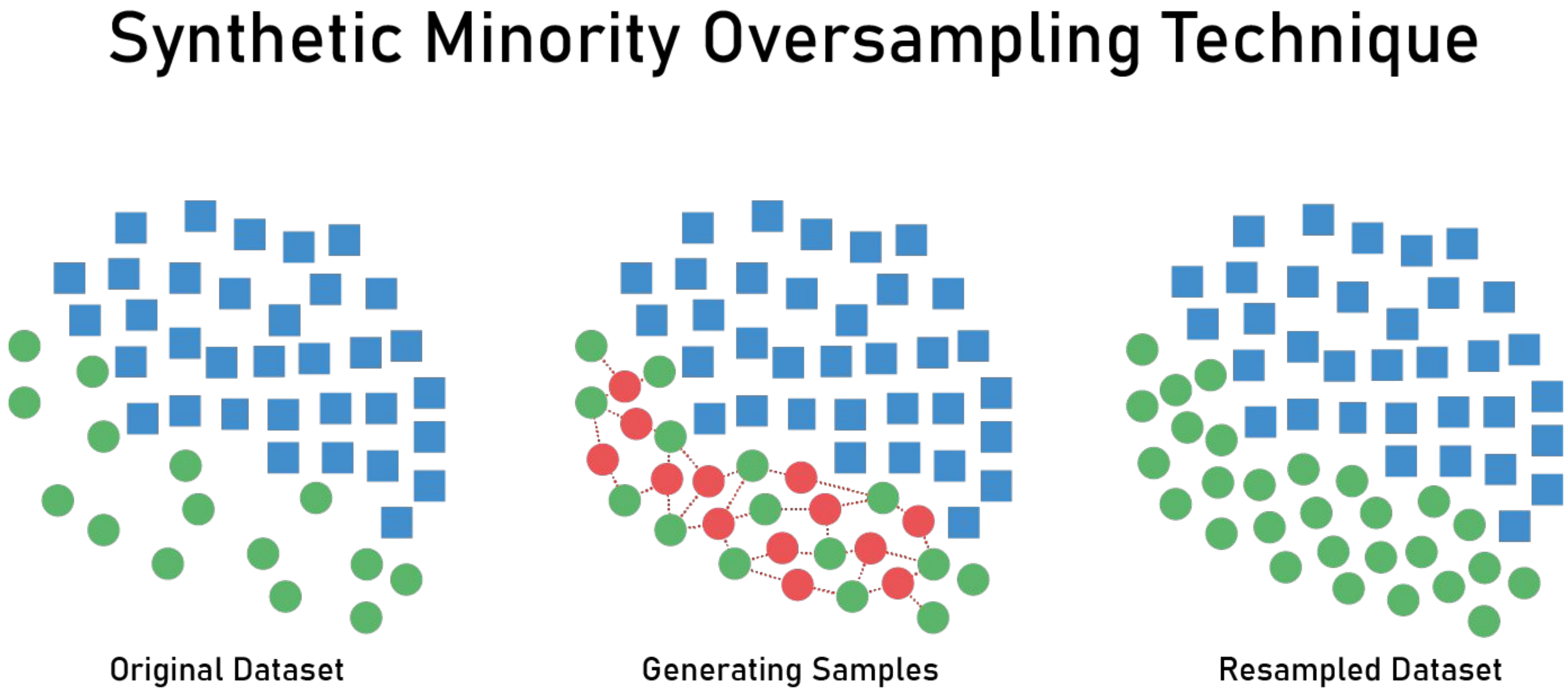
⇒ Removing data from the majority class

Addresses class imbalance
Reduces computational charge
Loss of information due to removing instances
Can introduce bias
Risk of underfitting when the imbalance is severe

Oversampling

⇒ Duplicating data from the minority class

Addresses class imbalance
No loss of information
Risk of overfitting
May introduce noise from the minority class



Generating artificial data with SMOTE

Image source

Synthetic Minority Oversampling Technique

Principle

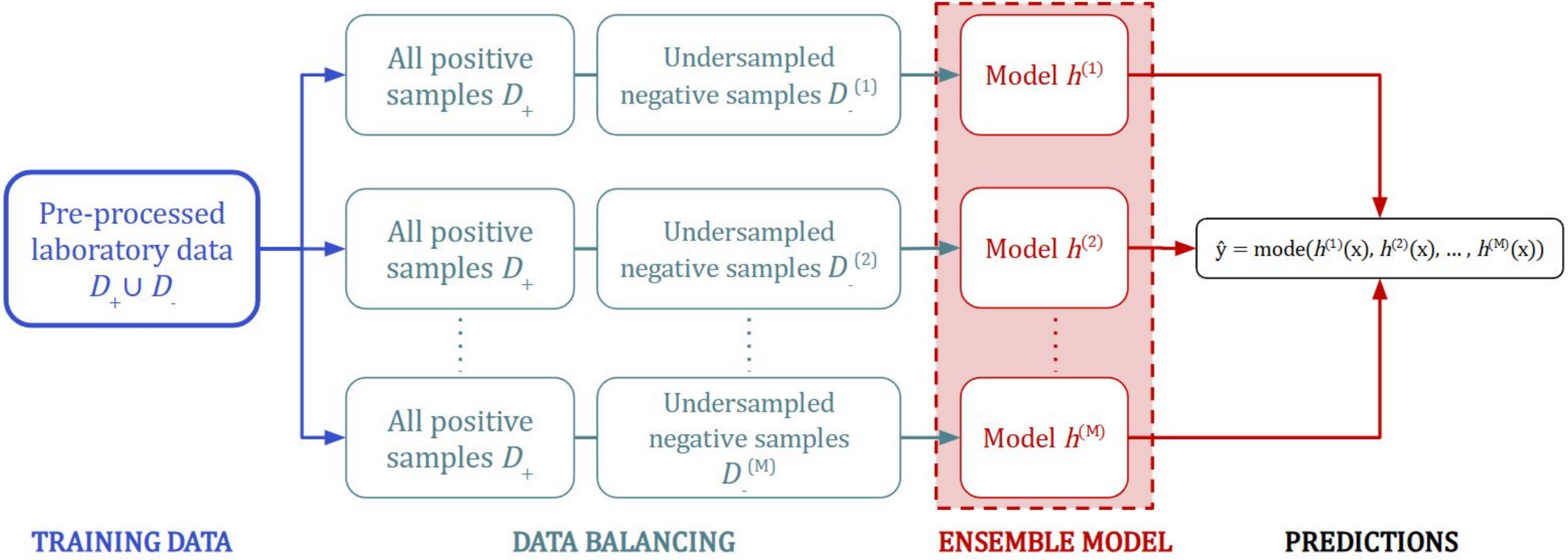
- Choose a value for *k*
- For each instance in the minority class, identify the *k* nearest neighbours
- Interpolate new values linearly

Variations

- ADASYN: Focuses on examples in low-density areas
- SMOTE-Tomek: Removes borderline noisy instances
- Borderline-SMOTE: Focuses on borderline instances

Generating artificial data with Generative Adversarial Networks (GAN)

Image source



Imbalance-aware machine learning algorithms - the example of EasyEnsemble



PERFORM RESAMPLING AFTER THE TRAIN-TEST SPLIT

Data leakage will <u>artificially inflate</u> your results



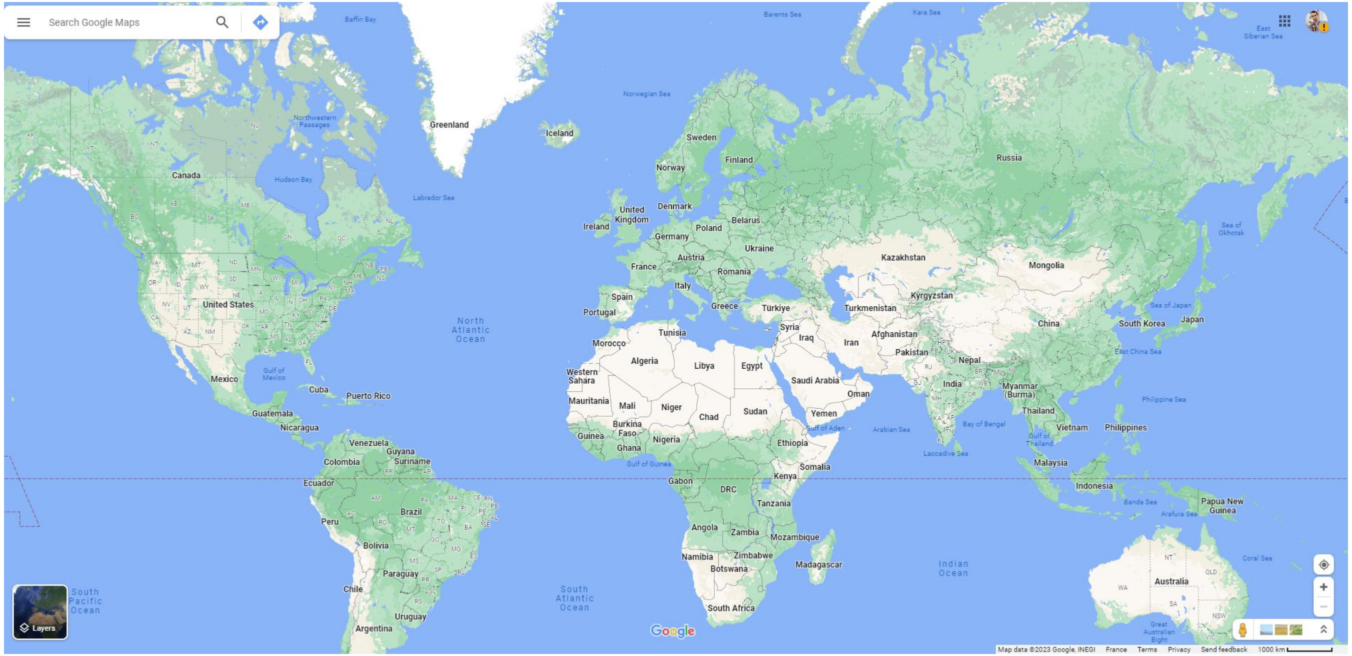
Practical work

Get the latest version of the notebook from GitHub

Don’t forget to upload your work!

What is the deidentification of data?

I am looking for a man…



Question 1 : 8 billion people ⇒ 4 billion men

Hiding someone’s name is not enough to hide their identity

Who teaches in Centrale Lille…

Question 2 : 4 billion men ⇒ < ~300 male teachers

Hiding someone's name is not enough to hide their identity

And is pursuing a PhD for AI in healthcare!

Question 3 : 300 people ⇒ 1 person

Hiding someone’s name is not enough to hide their identity

Deidentification is more complex than simple anonymization

Anonymization is not enough to hide someone's identity

- **Data linkage** can lead to reidentification
- Unique features can let you identify some people easily (e.g. few people are over 100 years old)

There are several techniques to deidentify data

- **Data masking**: hiding part of the value
- **Aggregation**: e.g. grouping ages within ranges
- **Generalization**: e.g. replacing dates with years
- **Data perturbation**: e.g. introducing noise
- **Data swapping**
- **Removing isolated data** (sometimes legally required)

⚠️The more you modify the data, the higher risk of reducing the algorithms' performance ⇒ find a compromise

Closing words on this first course

What we saw so far

1. Understanding data

- Asking the right questions
- How to visualize data

2. Clean code & Git

- How to use Git and GitHub
- The basics of collaborative development

3. Cleaning data & Missingness

- Sources of missing data
- Removing or imputing missing values

4. Dimensionality reduction

- Feature selection and feature extraction
- Principal Component Analysis

5. Data imbalance & Deidentification

- Undersampling and oversampling
- Deidentifying data

What we will see next

Debrief

Debrief

What did we learn today?

What could we have done better?

What are we doing next time?

Data Science

Session 5 - Data imbalance & Deidentification

hadrien.salem@centralelille.fr

introduction-to-data-science