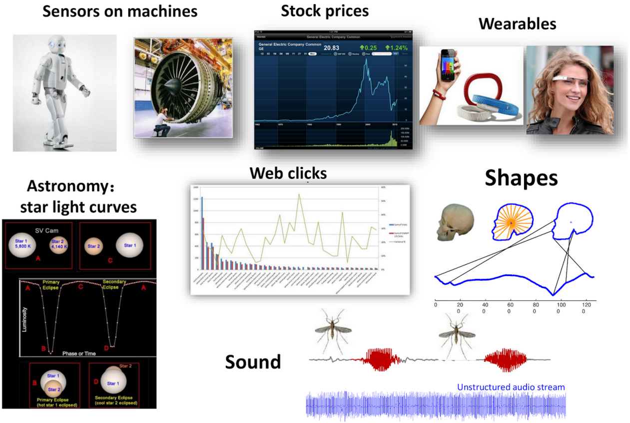
# Content

# Time Series

Time series

- describe the evolution of a process over time
- are everywhere and ubiquitous: daily life, medical, food security, financial, environmental...
- increase in quantity



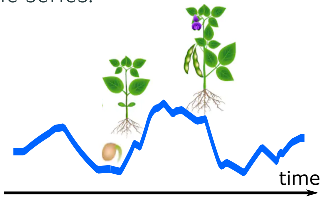
# Time Series

Formally, a time series

- is a sequence of values ordered in time
- either univariate or multivariate
- possibly of different lengths

An example univariate time series:

| time | value |
|---|---|
| t1 | 0.236 |
| t2 | 0.563 |
| t3 | 0.748 |
| t4 | 0.692 |
| ... | |
| tL | 0.167 |



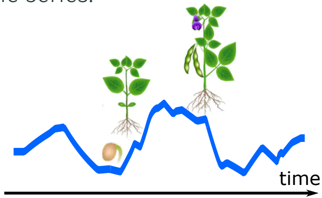
# Time Series

Formally, a time series

- is a sequence of values ordered in time
- either univariate or multivariate
- possibly of different lengths

An example univariate time series:

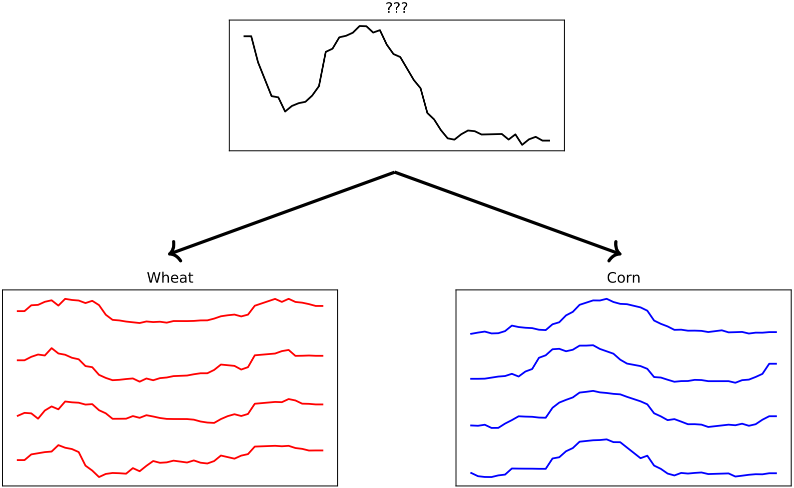
| time | value |
|---|---|
| t1 | 0.236 |
| t2 | 0.563 |
| t3 | 0.748 |
| t4 | 0.692 |
| ... | |
| tL | 0.167 |



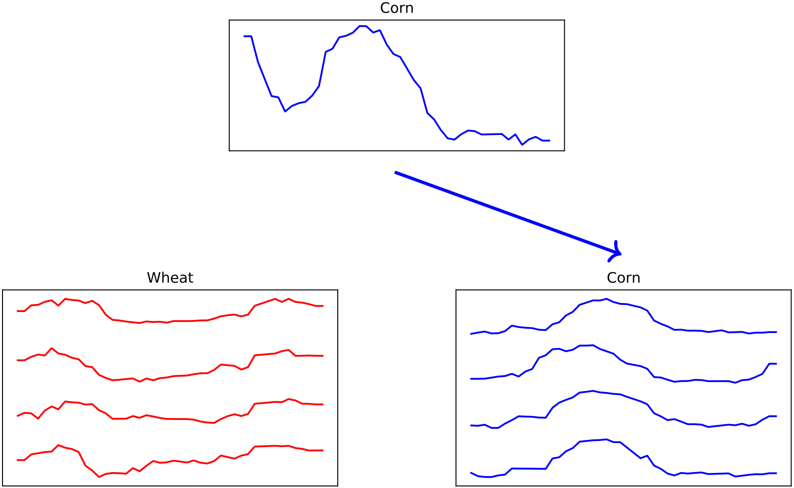

Time series analysis include

- forecasting: predicting future values
- regression: predicting a continuous scalar variable
- retrieval: finding similar time series
- segmentation: dividing a time series into "homogeneous" subseries
- classification: today's tutorial

The goal is to associate an unlabelled time series with a class with the help of some labelled time series (supervised learning).



The goal is to associate an unlabelled time series with a class with the help of some labelled time series (supervised learning).



An example satellite image time series

Sentinel-2 images over Brittany, France

An example satellite image time series

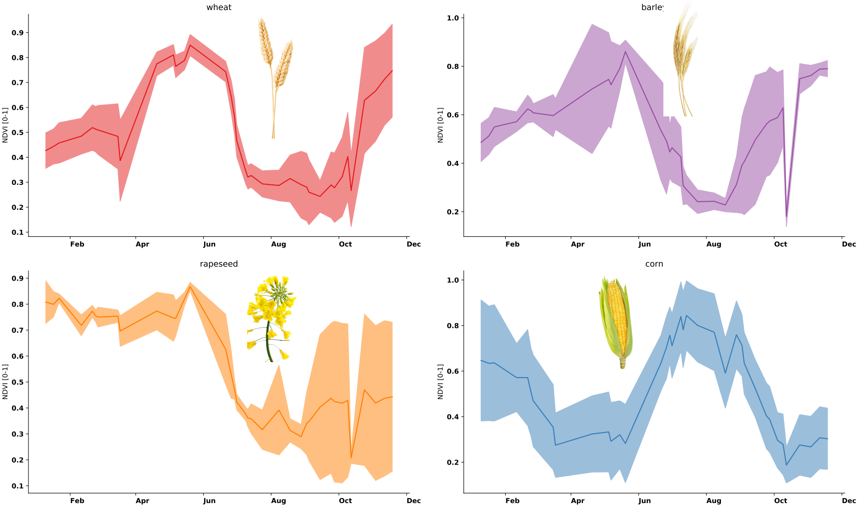
Sentinel-2 images over Brittany, France


Applications

- vegetation monitoring
- landscape changes
- large scale study

# Time Series in Remote Sensing

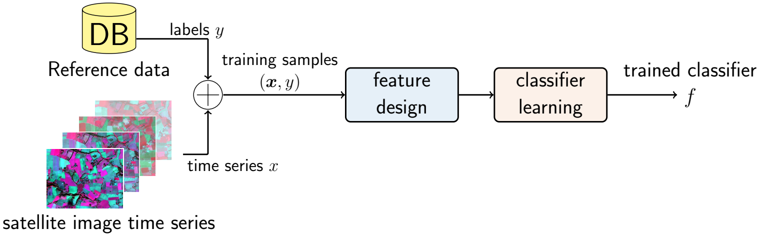
An example application: crop type mapping at large scale



# Supervised classification framework

Two main steps:

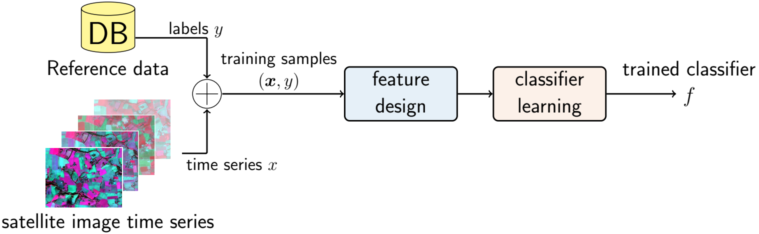
1. Learning a model $f$ such that $f(\mathbf{x}) \approx y$



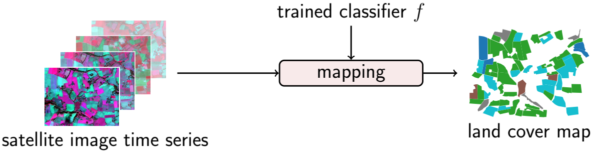
# Supervised classification framework

Two main steps:

1. Learning a model $f$ such that $f(\mathbf{x}) \approx y$




2. Using the model $f$ to map the study area



# Inputs for learning

Satellite data, *e.g.*, Sentinel-2 images

- Where to download images?
  - Sentinels Scientific Data Hub
  - Copernicus DIAS
  - cloud platforms: GEE, Amazon, Microsoft Planetary Computer
  - THEIA, USGSS, *etc.*
- Common pre-processing steps:
  - coregistration
  - atmospheric correction
  - gapfilling
  - *etc.*

# Inputs for learning

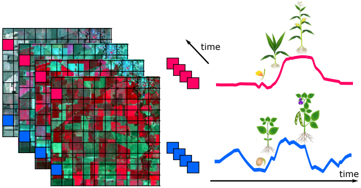
Satellite data, *e.g.*, Sentinel-2 images

- Where to download images?
  - Sentinels Scientific Data Hub
  - Copernicus DIAS
  - cloud platforms: GEE, Amazon, Microsoft Planetary Computer
  - THEIA, USGSS, *etc.*

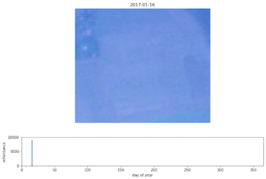
- Common pre-processing steps:
  - coregistration
  - atmospheric correction
  - gapfilling
  - *etc.*

From satellite images to time series

Pixel-based analysis




Object-based analysis, *e.g.*, averaging the reflectance values within an agricultural parcel



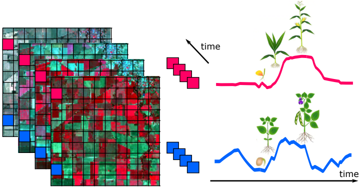
# Inputs for learning

## Satellite data, *e.g.*, Sentinel-2 images

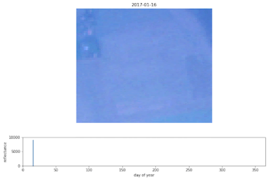
- Where to download images?
  - Sentinels Scientific Data Hub
  - Copernicus DIAS
  - cloud platforms: GEE, Amazon, Microsoft Planetary Computer
  - THEIA, USGSS, *etc.*
- Common pre-processing steps:
  - coregistration
  - atmospheric correction
  - gapfilling
  - *etc.*

From satellite images to time series

Pixel-based analysis




Object-based analysis, *e.g.*, averaging the reflectance values within an agricultural parcel




## Reference data

Usually vector files

↳ label $y \in \{1, \cdots, \mathcal{C}\}$

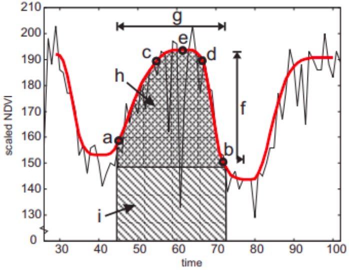
- photo-interpretation
- field campaigns
- governmental data (*e.g.* Corine Land Cover)
- collaborative data (*e.g.* Open Street Map)

# From time series to feature vectors

Feature design is a key step when using traditional machine learning algorithms

- flatten reflectance time series
- compute spectral features, *e.g.*, Normalized Difference Vegetation Index
- extract temporal features: statistical and phenological features
- and even compute spatial features, *e.g.* Haralick or attribute profiles

TIMESAT example: extraction of key phenological stages [1]




[1] Jönsson, P., & Eklundh, L. (2004). TIMESAT—a program for analyzing time-series of satellite sensor data. *Computers & Geosciences*, 30(8), 833-845.

# Evaluation

Types of evaluation: quantitative (accuracy, computational complexity, explainability), visual, evaluation on a downstream task

# Evaluation

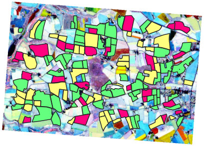
Types of evaluation: quantitative (accuracy, computational complexity, explainability), visual, evaluation on a downstream task

Quantitative evaluation: split the labeled data into 3 spatially independent sets

- a train set to learn the model's parameters
- a validation set to tune the hyperparameter values of the model
- a test set to obtain a non-biased estimation of the model's performance

Labeled data

Polygon-split



Grid-split

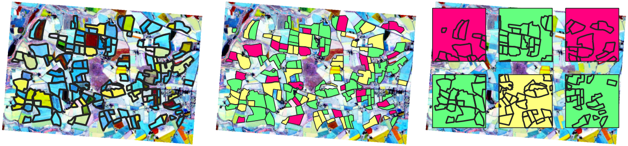
# Evaluation

Types of evaluation: quantitative (accuracy, computational complexity, explainability), visual, evaluation on a downstream task

Quantitative evaluation: split the labeled data into 3 spatially independent sets

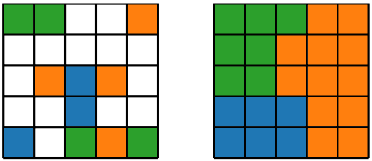
- a train set to learn the model's parameters
- a validation set to tune the hyperparameter values of the model
- a test set to obtain a non-biased estimation of the model's performance

Labeled data Polygon-split Grid-split



The confusion matrix:

Ground truth Dense predictions



*predicted*

| *real* | | | |
|---|---|---|---|
| | 2 | 1 | 0 |
| | 0 | 3 | 1 |
| | 1 | 1 | 2 |

# Practical activity

How to implement this framework?

- develop your own Python code
- use dedicated libraries, *e.g.*, snap, OTB
- use existing frameworks, *e.g.*, ***`iota2`*** or R-SITS package

# Practical activity

How to implement this framework?

- ◇ develop your own Python code
- ◇ use dedicated libraries, *e.g.*, snap, OTB
- ◇ use existing frameworks, *e.g.*, ***`iota2`*** or R-SITS package

Let us now move to our first practical activity!

**Notebook 1: Feature Engineering**

| | | pre-processing | | feature design | | classification | |
|---|---|---|---|---|---|---|---|
| **feat. engineering** | raw data $X$ | $\longrightarrow$ | processed data | $\longrightarrow$ | features | $\longrightarrow$ | labels $y$ |
| **realization** | Sentinel-2 Series | $\longrightarrow$ | cloud filtering | $\longrightarrow$ | NDVI-statistics | $\longrightarrow$ | crop type $y$ |

Link for the notebooks: *`https://tinyurl.com/isprs2022`*