

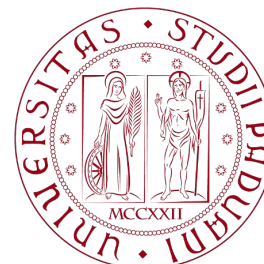
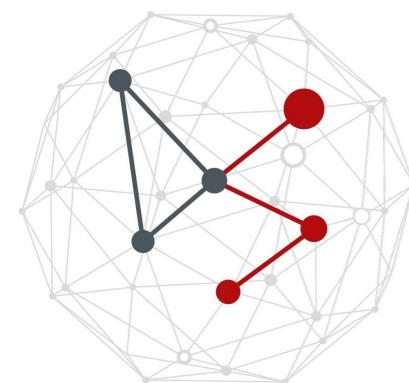
LAB 1 - MACHINE LEARNING TOOLS FOR DIMENSIONALITY REDUCTION: PCA AND CLUSTERING

Riccardo Mazzieri, Francesca Meneghello

riccardo.mazzieri@phd.unipd.it

francesca.meneghello.1@unipd.it

University of Padova, IT

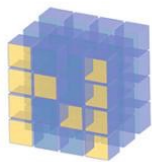


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Lab classes



- In this lab we will start with some basic tools



NumPy



matplotlib

- We suggest you use [Google Colab](#) but if you are familiar with your local Python editor feel free to use it

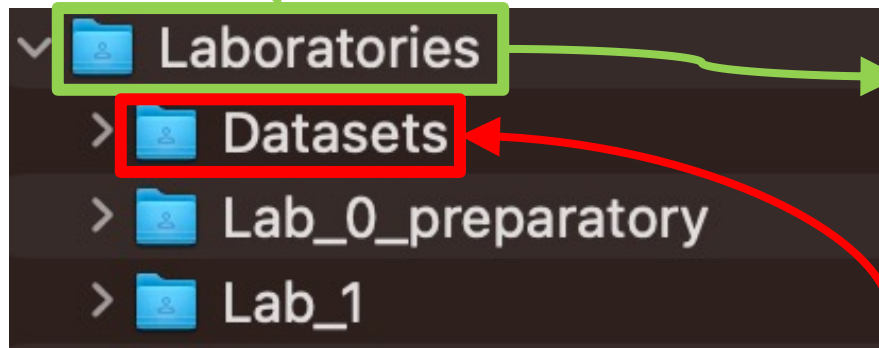


Note: Google Colab does not work with the University's Google accounts. You need to use your own Google account or create a new one.

Practical info – 1/2

Folder structure in Google Drive

- Create a folder where you will create your own copy of the labs



In the notebook we call it **HDA_labs**, feel free to chose your name and location, and then remember to modify the notebook accordingly

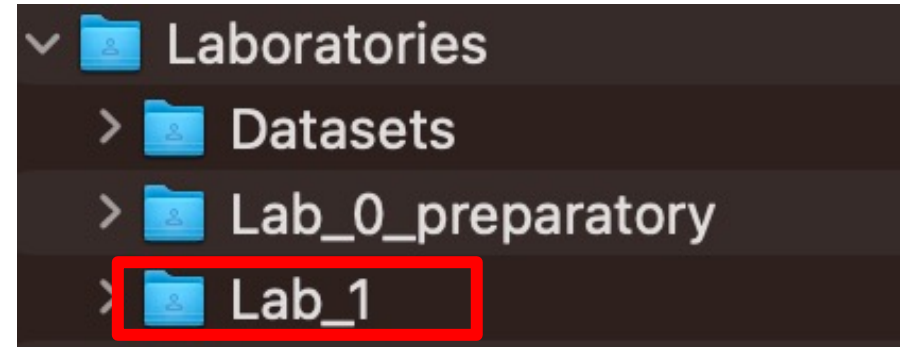
Only the first time

- Create a shortcut to the folder of the **datasets in your Google Drive** – this folder is unique for all the laboratories
→ this will allow you to directly import the shared files without the need for copying them

Practical info – 2/2

For each lab

- Create **your own copy of the entire folder of the laboratory** in your Google Drive (copy all the files, i.e., the .ipynb and .py files, and the folder of the images)



▼ LABORATORIES (Francesca Meneghello, Riccardo Mazzieri) ✎



FEEDBACK

Laboratory classes attendance ✎



FILE

Instructions for the labs

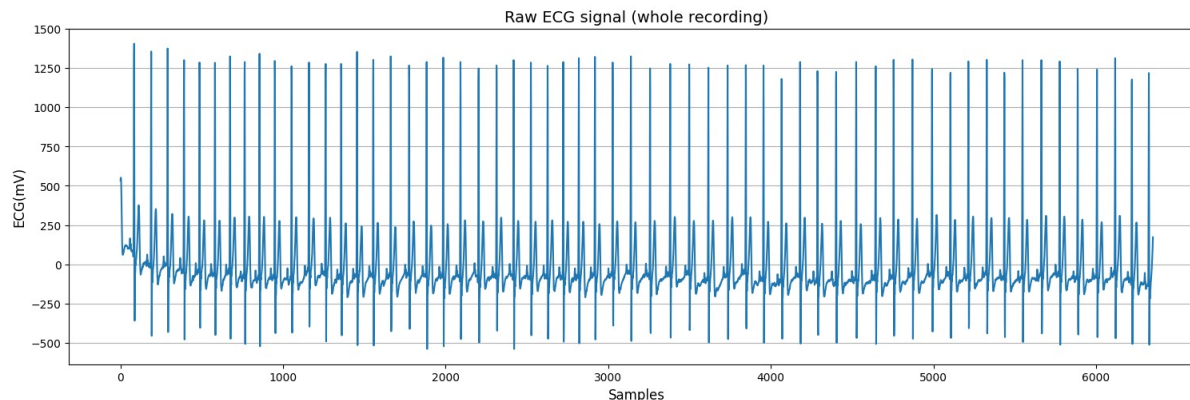


Lab 0 - preparatory material

https://drive.google.com/drive/folders/1d_BgR38zDVWPspjIptQ_mZp1DwB0uYTc?usp=sharing

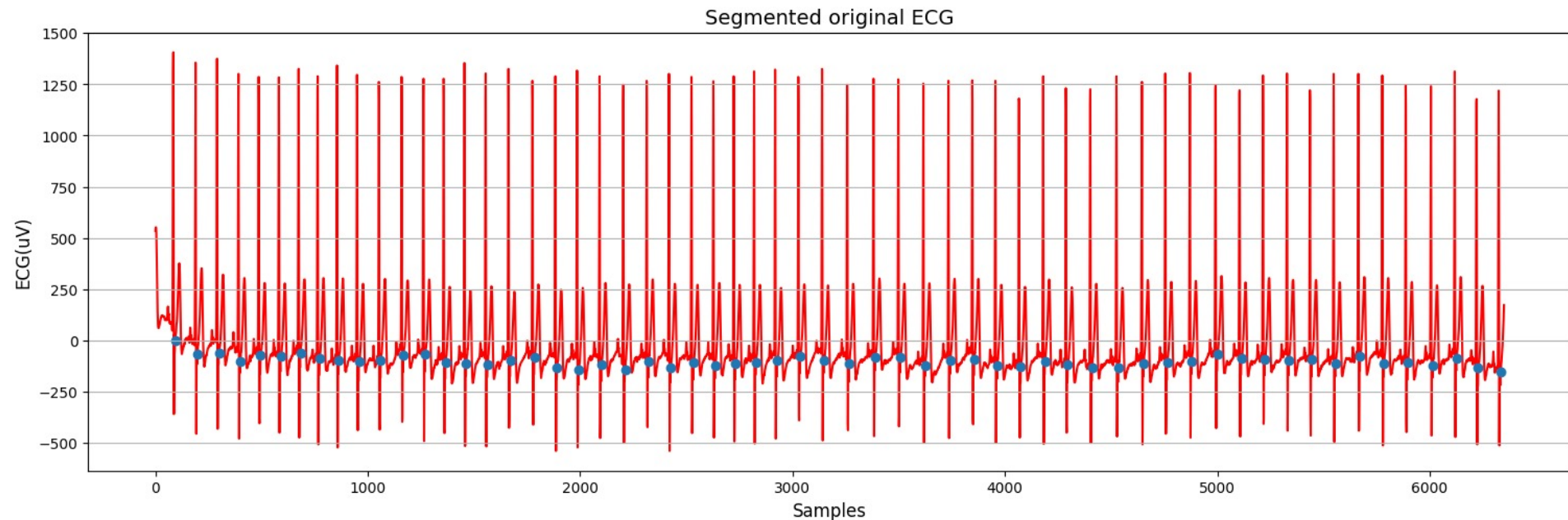
Lab 1 – Oct. 25

- Machine learning tools for dimensionality reduction: PCA and clustering
 - **The challenge:**
 - ECG signals dimensionality reduction
 - **You will learn to:**
 - implement a dimensionality reduction algorithm based on PCA from sketch and using the implementation from Python libraries
 - implement dimensionality reduction based on clustering
 - use **Scikit-learn** Python library



Preprocessing

1. Split the ECG signal into the beats → dimensionality reduction will be applied to each beat
2. **Normalize** the data → very important step to be applied at the input of machine learning algorithms!

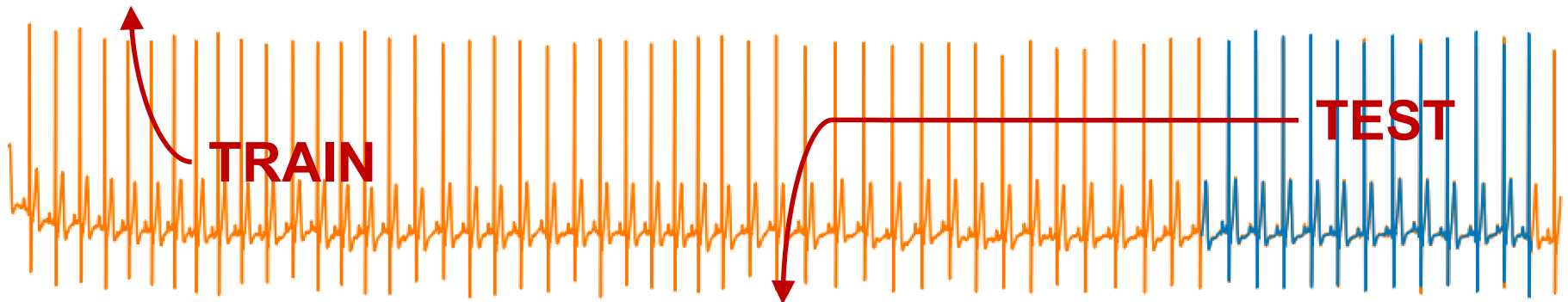


PCA

Obtain the principal components through two approaches:

- Implement PCA from scratch following the procedure detailed in **L01_PCA** (slides from Prof. Rossi) → will allow you to digest the procedure seen during the lectures
- Use the PCA algorithm already implemented in Python and offered as part of the **Scikit-learn** library

<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html#sklearn.decomposition.PCA>

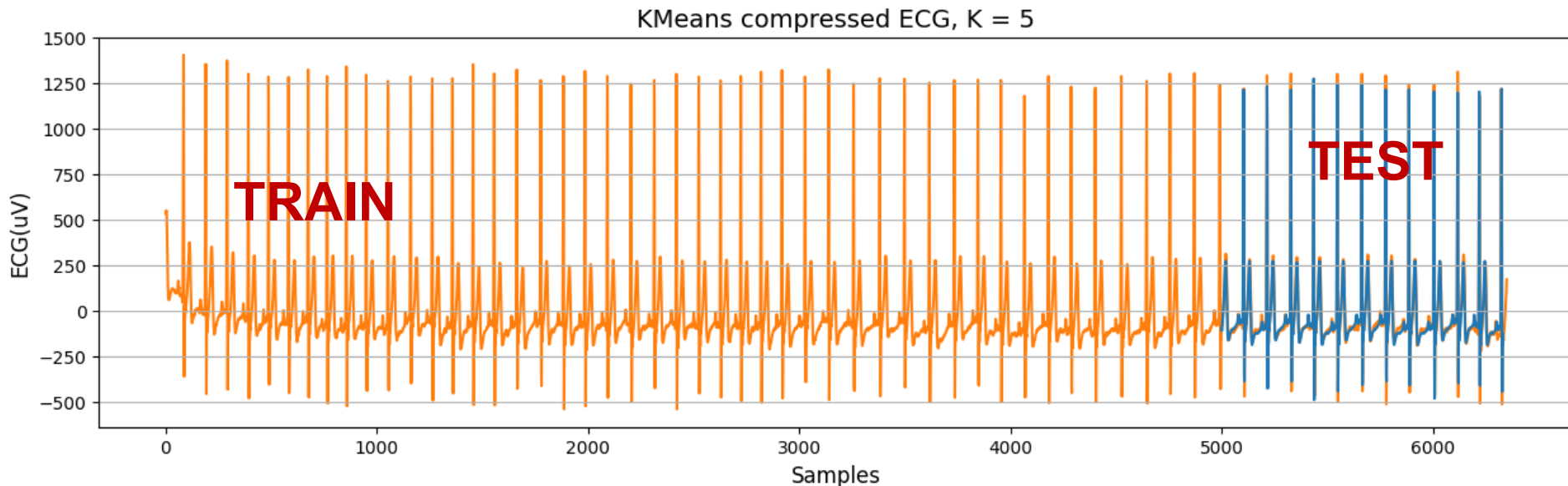


Evaluate the effectiveness of the components in representing the ECG beats (test on data not used during training!)

Clustering: K-means

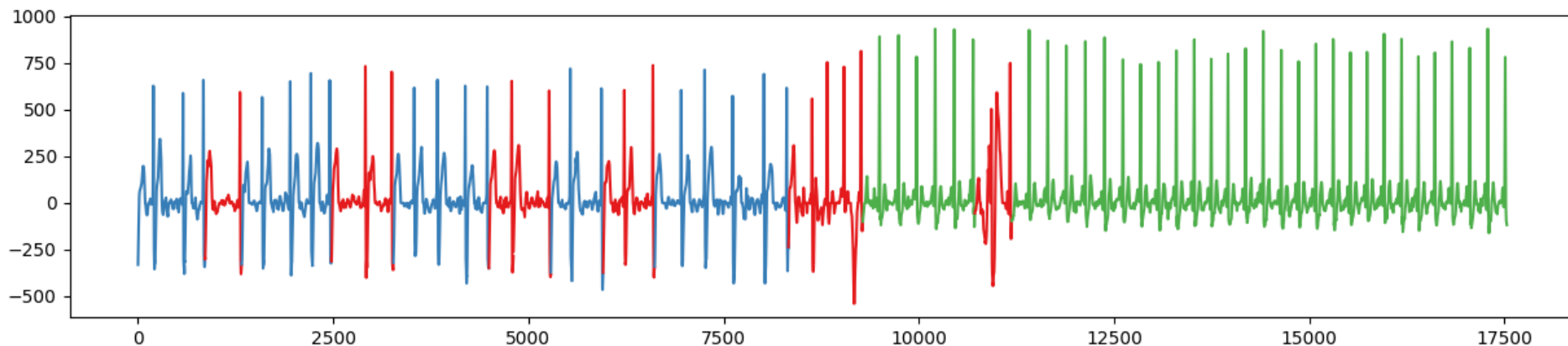
- Use K-means with 5 classes to obtain some representative beats (class centroids) to be used as codebook for dimensionality reduction
- Use the **Scikit-learn** library and evaluate the reconstruction error

<https://scikit-learn.org/stable/modules/clustering.html#k-means>



Clustering: DBSCAN

- You will use a different ECG signal composed of
 1. A first part collected from a patient with **atrial fibrillation**
 2. A second part with normal beats
- Implement DBSCAN using the **Scikit-learn** library
- Find the optimal **Eps** to separate the two classes (and outliers) using the procedure explained during the classes (see **L03_DBSCAN**)



How can I fill the notebooks?

1. Read the text before the code: you can find the objective of that portion of code and details about the procedure to implement
2. Click on the links of the documentation and read it
 - What does the function do? Is it the right function for your objective?
 - What are the **inputs** of the function? Matrix/array, shape, type...
 - What are the **outputs** of the function?
 - How should the function be **imported** at the beginning of the notebook? Usually we already do it for you but for the final project you will need to do it on your own, take a look at it
3. If the link to the documentation is not provided...search the documentation on your own 😊

Import section

Two approaches:

https://docs.python.org/3/reference/simple_stmts.html#import

```
# Scientific packages
```

```
import numpy as np
```

```
import scipy
```

```
import matplotlib
```

```
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.decomposition import PCA
```

```
from sklearn.cluster import KMeans
```

```
from sklearn.metrics import mean_squared_error
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.cluster import DBSCAN
```

```
from kneebow.rotor import Rotor
```

```
from scipy.io import loadmat
```

```
from scipy.signal import medfilt
```

```
from scipy.signal import resample
```

```
# Plot packages
```

```
import matplotlib.pyplot as plt
```

```
from matplotlib.gridspec import GridSpec
```

```
# Custom functions
```

```
from hda_utils import load_dataset, load_dataset_DBSCAN, segment_ECG, segment_ECG2, matrix_to_signal
```