Machine Learning Project Specifications (M1)

Course: Python & AI
Instructor: Khouloud Chelbi
Duration: 3–4 weeks
Groups of 2–4 students

1. General Objectives

The project aims to teach students how to:

- Apply course concepts
- Work in a real ML project structure
- Use a clean environment and JupyterLab
- Collaborate effectively (Git, modular files)
- Build a complete ML pipeline

2. Technical Requirements & Environment

2.1. Create a dedicated Python environment

Using **venv** or **conda**:

Libraries required:

nginx Copy code

```nginx
numpy
pandas
matplotlib
seaborn
scikit-learn
```

Add installation steps in **README.md**.

2.2. Work exclusively in JupyterLab

All notebooks must be clean, readable, documented.

2.3. Project structure (mandatory)

```
project/
│
├── data/
├── notebooks/
│    ├── 01_EDA.ipynb
│    ├── 02_data_preparation.ipynb
│    ├── 03_modeling.ipynb
│    ├── 04_evaluation.ipynb
│
├── utils/
│    └── preprocessing.py
│    └── visualization.py
│
│
├── README.md
└── requirements.txt
```

2.4. Utils files (mandatory)

Reusable functions must be placed in `/utils`:

- Preprocessing
- Plots
- Evaluation

This teaches:

- modularity
- clean project architecture
- teamwork-friendly code

3. Project Types

Students may choose:

✔ Supervised Learning

- Classification
- Regression

✔ Unsupervised Learning

- Clustering (K-Means, Hierarchical, DBSCAN)

4. Required Work – Pipeline Steps

Notebook 01 – EDA

- Descriptive analysis
- Minimum 5 visualizations
- Correlation study
- Problem identification

Notebook 02 – Data Preparation

- Cleaning
- Encoding
- Scaling
- Outlier detection
- Train/test split (if supervised)
- Functions in `utils/preprocessing.py`

Notebook 03 – Modeling

Supervised:

- 1 baseline model
- 1 improved model

Clustering:

- 2 algorithms compared
- Cluster number selection
- Cluster visualization

Notebook 04 – Evaluation

- Metrics depending on task
- Interpretation
- Graphs
- Functions in `utils/visualization.py`

5. Collaboration Requirements

Students must demonstrate:

- Organized task distribution
- Use of shared workspace (GitHub recommended)
- Version control
- Collective validation of the final product

6. Deliverables

- Full project folder
- PDF report
- Final oral presentation (10 + 5 min)

7. Grading

Component	Weight
Project organization & structure	20%
EDA + Preparation	20%
Modeling	20%
Evaluation + Interpretation	15%
Code quality (utils required)	15%
Presentation	10%