

Graded Project : Part 2 (AI Project Technical Methodologies)

⇒Goal

The goal of this part 2 of the graded project is to be more familiar with technical best practices of Data & AI projects through a hands on project related to part 1 (customer churn prediction)

⇒Tools

It is recommended to use a linux or mac OS, an IDE with a **recent python version**.

⇒References

Add all sources you may rely on in your report

⇒Restitution

- A report describing your technical project and containing your code repository URL (on github or gitlab), the report must be available in the **same group dedicated repository** as for part 1:
📁 AI PM_Graded Project_Groups_092024
- You must share your github project with email : dinamedy@hotmail.com or gitlab project with hadjem.m@gmail.com

Implement AI Customer Churn Project with coding best practices

Build an AI Project based on the [E-Commerce churn Dataset](#) (also available [here](#)) and on a ML model of your choice with respect for technical best practice for production ready project, mainly :

Versioning & collaboration

Use GIT for code & models versioning, you can use github or gitlab as a collaborative platform

Code Structure

Structure your project :

- Separate your ML project workflow into different scripts (data preparation, feature engineering, models training, predict)
- Use a cookie cutter as a template to structure all components of your project (Example : <https://drivendata.github.io/cookiecutter-data-science/> you can use it or adapt it or propose your own cookie cutter)

Standardization

Use the same python version for your group with respect to the [PEP8](#) style guide for python (you may use [pylint](#), [flake8](#) or [black](#) for that)

Reproducibility

Use a dependency management solution for your libraries & packages (Ex : conda or poetry)

Documentation

Use a code documentation library to automate the documentation generation ([Sphinx](#) recommended) you can find documentation here : [sphinx documentation](#)

Model tracking & local deployment

Experiment [MLflow](#) with your project for :

- **Track parameters & metrics of your model** and display the results in your local MLflow UI (multiple runs), you can find a documentation here : [MLflow tracking](#)
- **Package your code** in a reusable and reproducible model format with [MLFlow projects](#)
- **Deploy your model into an API** that will enable you to score predictions using [ML Models](#)

Cloud Deployment

Propose a solution of your choice to deploy your AI project (data, code, models) on a cloud infrastructure (Ex : you can use a free version of google cloud). explain your solution in the report (even if you couldn't implement it).

Remark : having the best ML/AI performances is not the goal of this project

The evaluation of this part 2 will be mainly based on the report + your project repository on github or gitlab containing :

- All project code and best practices (notebooks, scripts, requirements, etc.)
- Generated Models
- MLFlow outputs