Development project in Machine Learning

TAF MCE

Elsa Dupraz
elsa.dupraz@imt-atlantique.fr

October 8th, 2024

Objectives

Versioning with GIT

Project description

Deliverable

Objectives

Versioning with GIT

Project description

Deliverable

Objectives

- Develop good programming practices
- Use standard development tools
- Get used to collaborative work

- Work on Machine-Leaning datasets

Objectives

Versioning with GIT

Project description

Deliverable

Objectives

Versioning with GIT

Project description

Deliverable

Versioning with GIT

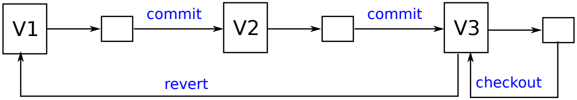
- GIT is a versioning system
- It can keep track of the successive code versions
- It allows several persons to work on the same files, and can merge the various contributions

Tutorials:

- `https://openclassrooms.com/fr/courses/1233741-gerez-vos-codes-source-avec-git` **(Course)**
- `https://github.com/girliemac/a-picture-is-worth-a-1000-words/tree/main/git-purr` **(Pictures)**
- `https://ohmygit.org/` **(Game!)**

How GIT works

- Versioning:

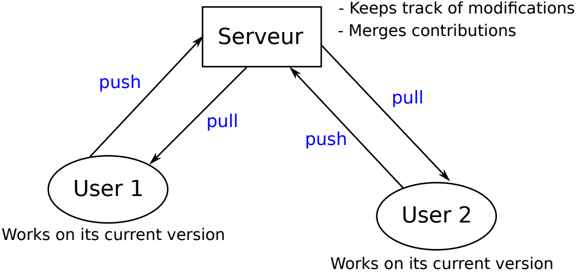


Example: coffee machine

- V1: template (repository organization, readme, etc.)
- V2: Added function to boil down water
- V3: Added function to pour water
- Then: Modified function to boil down water, but broke everything!!

How GIT works

- Architecture:

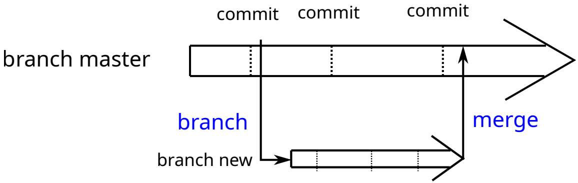

Serveur
- Keeps track of modifications
- Merges contributions
push
pull
pull
push
User 1
Works on its current version
User 2
Works on its current version

Example: coffee machine

- User1: works on function to boil down water
- User2: works on function to pour water

How GIT works

- Branches:



Example: coffee machine

- branch master: the machine can only makes coffee
- branch new: we want to add modes to choose between coffee or tea

Objectives

Versioning with GIT

Project description

Deliverable

Machine Learning Workflow

1. Import the dataset
2. Clean the data, perform pre-processing
   - Replace missing values by average or median values
   - Center and normalize the data
3. Split the dataset
   - Split between training set and test set
   - Split the training set for cross-validation
4. Train the model (including **feature selection**)
5. Validate the model

Objective: collaboratively implement this workflow and apply it to different ML datasets

Datasets

The objective of the project is to apply a Machine Learning model for Binary classification onto two different datasets:

- Banknote Authentication Dataset: `https://archive.ics.uci.edu/ml/datasets/banknote+authentication`
- Chronic Kidney Disease: `https://www.kaggle.com/mansoordaku/ckdisease`

Constitute groups of 3 to 4 students.

What I expect

- Create a git repository for your group: `https://gitlab.imt-atlantique.fr/`
- Test a few different models for classification, and implement feature selection
- Write the Python functions implementing the workflow in one single .py file.
- Write at least one unit test for one of the functions
- Apply the same workflow onto the two datasets, called from a Jupyter Notebook
- Think about good programming practices

Objectives

Versioning with GIT

Project description

Deliverable

Deliverable

Code: **one .py** file for the functions and **one .ipynb** file to run the code included in the Git repository + all what is needed to run the code (datasets, another .py file for unit tests if needed, etc.)

One Git repository per group should be “sent” by e-mail to `elsa.dupraz@imt-atlantique.fr` before the 20th of November, 11PM.

→ please give me the sufficient level of rights to see the repository (e.g. developer)

The .py file must contain...

- **One** function for pre-processing
- **One** function to prepare the dataset for training
- **One** function for training (typically applies up to 5 different methods for binary classification)
- **One** function to display all the results in a convenient form for comparison

- As many sub-functions as needed

The .ipynb file must...

- Apply the functions **of the .py file** onto the two datasets
- Show and compare the results for each dataset (e.g., numbers, curves, etc.)
- Comment on the results in both cases (which method would you choose at the end?)
- Discuss good programming practices and how they were taken into account into the project

Final comments

- Advice for good programming practices: `https://mikecroucher.github.io/reproducible_ML/`
- Register your groups before the 14th of October, at `https://semestriel.framapad.org/p/bpvm0u0niv-aagc?lang=fr`