M2 Data Engineering
Applications of BigData
2022-2023

Project

Goals

The goal of this project is to apply some concepts & tools seen in the 3 sessions of this course, this project is organized into 3 parts :

- Part 1 : Building Classical ML projects with respect to basic ML Coding best practices
- Part 2 : Integrate MLFlow to your project
- Part 3 : Integrate ML Interpretability to your project

DataSet (Finance use case)

DataSet of Home Credit Risk Classification:

https://www.kaggle.com/c/home-credit-default-risk/data

you'll not use all the datasets available on Kaggle, only the main data set :

⇒application_train.csv

⇒application_test.csv

You may also use a reduced version of these datasets

Requirements

Linux OS is recommended, an IDE with last python version (use anaconda environment for example)

Part 1

Build an ML Project for **Home Credit Risk Classification** based on the given Dataset with respect to coding best practice for production ready code :

- Use GIT for team collaboration, code & model versioning
- Separate your ML project workflow into different scripts (data preparation, feature engineering, models training, predict)
- Use a template cookie cutter or adapt/define your own (Example : https://drivendata.github.io/cookiecutter-data-science/)
- Use a conda environment for all your libraries (or any other package/environnement management like poetry)
- Use a documentation library (Sphinx recommended)

For this project, you can choose one of these classical ML algorithms : Xgboost, Random Forest or Gradient Boosting, **having the best ML performances <u>is not</u> the goal of this project**

- Propose a solution to Schedule your ML pipeline **(Optional)**

<u>Part 2</u>

Integrate **MLFlow** Library to your Project :

- Install MLFlow in your python environment (don't forget to add it to your lib requirements)
- Track parameters & metrics of your model and display the results in your local mlflow UI (multiple runs)
- Package your code in a reusable and reproducible model format with ML Flow projects
- Deploy your model into a local REST server that will enable you to score predictions **(Optional)**

<u>Part 3</u>

Integrate **SHAP Library** to your Project :

- Install SHAP in your python environment (don't forget to add it to your lib requirements)
- Use it to explain your model predictions :
    - Build a TreeExplainer and compute Shaplay Values
    - Visualize explanations for a specific point of your data set,
    - Visualize explanations for all points of your data set at once,
    - Visualize a summary plot for each class on the whole dataset.
    -

<u>Report</u>

Your project must be structured in a report, you can write a separate report or integrate it in your GIT repository.

Project conditions & evaluation

-Work in teams of 2 or 3 max

-Delivery due date : **24/01/2023**

-Project Defense **: 31/01/2023 (10 minutes per group)**

The evaluation of your project will be based on your Project GIT Repository (please share it with this email address : **dinamedy@hotmail.com)** contraining :

-Project code (git repo organization, notebooks, scripts, etc.)

-Project Outputs (predictions on test dataset, Automatic Documentation, MLflow outputs, SHAP Output)

-Synthetic Report