Projet Final

Début 9h, retour 17h30

(Aucun retard ne sera permis)

Objectif

- Simuler un projet d'entreprise réel où plusieurs développeurs travaillent sur le même projet et chacun développe une fonctionnalité dans une branche différente puis fusionne sa version finale avec la branche principale (main).
- Créer ensuite un docker qui se base sur le repository Github (branche main) et se met à jour automatiquement en production.

Consignes

Un repository Github contenant des templates vous aidera à avancer : MLDS-AF/Examen-M1

- Analyser les données *train_diabetes_health_indicators* (analyse descriptive, correlations entre les features, proportions des classes (déséquilibre), etc.) et développer un modèle de classification pour prédire le diabète (la colonne *Diabetes_012*). Les modèles à tester sont : Random forest, SVM et GradientBoosting présents dans scikit-learn.

- Le point de départ est un repository GitHub que vous créez contenant un README et un .gitignore. Rajouter ensuite un fichier main.py (dans la branche main commune) qui servira à évaluer les prédictions de chacun des modèles après l'entrainement (vous pouvez vous appuyez sur le template_main.py).

- L'objectif est de travailler en collaboration pour remplir le fichier main.py mais de manière indépendante et asynchrone en développant un modèle de classification par personne dans une nouvelle branche (vous pouvez vous appuyer sur le notebook *template_branche.ipynb* pour développer et tester le modèle). Une fois la modèle entrainé et testé vous pouvez l'exporter et fusionner votre branche avec la branche main (en utilisant un merge pull request) en rajoutant votre notebook dans un dossier commun appelé experiments par exemple, et en mettant à jour le fichier main.py pour tester votre modèle sur les données finales *validation_diabetes_health_indicators*.

- Faire un clone de la branche main finale sur une machine et créer une image docker dans le but d'executer le fichier main.py et retourner le résultat de classification (prédiction) de chaque méthode.

- Pousser l'image dans docker Hub.

Rendu :

- Le lien vers Github et dockerHub dans le fichier : lien groupes.
- Si vous avez créé un volume en local dans la partie docker, spécifier sur le README comment l'utiliser, et mettre la commande pour le lancer en local.

Important : Ne plus faire de push après avoir rendu le projet

Évaluation :

Le projet sera évalué en fonction de :

- Fonctionnement de la solution et la capacité à interpréter les résultats.
- Integration des fonctionnalités et pratiques vues en cours.
- Répondre aux besoins en travaillant en collaboration et faciliter l'intégration de la solution finale.

Bonus :

- Développer le modèle en local avec docker et monter un volume sur le projet, ça vous permet de remplir le Dockerfile au fur et à mesure et de travailler ensemble sur la même version de python et des librairies.
- Améliorer la qualité du rendu final, par exemple : permettre à l'utilisateur de choisir le(s) modèle(s) qu'il souhaite tester, ou bien executer le modèle dans un backend. Toute autre proposition sera appréciée.
- Rédaction d'une bonne documentation (README) et des commentaires dans le code (notamment pour interpreter les résultats).
- Réaliser des cross validations.
- Optimiser les hyperparamètres à l’aide de grid search.
- Mettre à jour un des modèles après l’avoir dockerisé (à travers un volume).

Bon courage !