Fichier explicatif dossier Github

Voici un fichier introductif permettant de comprendre l'objectif du projet et le découpage des dossiers, et un fichier listant les packages utilisés seront présents dans le dossier GitHub.

I. Objectif du projet 7 (formation Data scientist)

L'entreprise « Prêt à dépenser » propose des crédits à la consommation. Elle souhaite mettre en œuvre un «scoring credit» pour accorder ses crédits selon la probabilité qu'un client rembourse son crédit. Elle souhaite donc développer un algorithme de classification en s'appuyant sur des sources de données variées (données comportementales, données provenant d'autres institutions financières, etc.).

Dans le cadre de ce projet, les missions qui me sont accordées sont :

1. Construire un modèle de scoring qui donnera une prédiction sur la probabilité de faillite d'un client de façon automatique.

2. Analyser les features qui contribuent le plus au modèle, d'une manière générale (feature importance globale) et au niveau d'un client (feature importance locale), afin, dans un soucis de transparence, de permettre à un chargé d'études de mieux comprendre le score attribué par le modèle.

3. Mettre en production le modèle de scoring de prédiction à l'aide d'une API et réaliser une interface de test de cette API.

4. Mettre en œuvre une approche globale MLOps de bout en bout, du tracking des expérimentations à l'analyse en production du data drift.

II. Organisation du dossier

Ce dossier est constitué de 4 dossiers et 2 fichiers :
- Dossier "Data"
- Dossier "mlflow_model"
- Dossier "notebooks"
- Dossier "tests_unitaires"
- Fichier "api.py"
- Fichier "requirements.txt"

1. Dossier « Data »

Ce dossier est constitué d'un unique sous-dossier intitulé "cleaned" et dans lequel se trouve 2 fichiers:
- application_train_cleaned.zip : fichier contenant la totalité des clients de l'entreprise « Prêt à dépenser ».

- application_train_cleaned_frac_10%.zip : fichier contenant un échantillon de 10% des clients de l'entreprise « Prêt à dépenser » pris aléatoirement.

2. Dossier « mlflow_model »

Ce dossier est constitué de 5 fichiers générés lors de l'export du modèle de machine learning par ML flow :

- mlmodel : contient les métadonnées du modèle.
- conda.yaml : spécifications de l'environnement Conda dans lequel le modèle doit être exécuté.
- model.pkl : modèle de machine learning qui va réaliser la prédiction.
- python_env.yaml : spécifications de l'environnement Python dans lequel le modèle doit être exécuté.
- requirements.txt : liste des dépendances Python nécessaires pour exécuter le modèle.

3. Dossier « notebooks »

Ce dossier est constitué de 2 fichiers :

- Berthe_Pierrick_2_notebook_nettoyage_022024.ipynb : Notebook de nettoyage et début de feature engineering du dataset initial du projet.
- Berthe_Pierrick_2_notebook_nettoyage_022024.ipynb : Notebook de fin de feature engineering du dataset initial du projet et test de différents modèles de machine learning pour parvenir à réaliser la classification des clients par le « score credit ».

4. Dossier « tests_unitaires »

Ce dossier est constitué d'un fichier unique :

- test_unitaire_api.py : script python permettant de réaliser 4 tests unitaires sur l'API déployée sur le cloud :
  - Test de la route d'accueil de l'application Flask
  - Test de la route POST de la prédiction de l'application Flask
  - Test de la prédiction de l'API en utilisant les données du 1er client
  - Test de la réponse de l'API en cas d'erreur 415.

5. Fichier « api.py »

Ce script python permet de créer une API en utilisant une application de la librairie Flask. L'application télécharge les données et le modèle de machine learning puis elle va définir des endpoints pour les requêtes GET et POST qu'elle va recevoir et qui vont permettent d'extraire les id des clients, d'extraire les données d'un client en obtenant son id, de prédire sa classe avec un CustomModelWrapper qui permet de personnaliser le seuil de prédiction et de calculer la feature importance globale. Pour information, l'API déployée est configurée pour fonctionner avec le dataset qui comprend seulement 10% des clients de l'entreprise « Prêt à dépenser » pris aléatoirement. Cette démarche n'est pas la finalité du projet qui consisterait à faire fonctionner l'API avec 100% des clients mais ce nombre restreint de client permet d'obtenir des temps d'affichage des résultats convenables sur le dahboard Streamlit.

6. Fichier « requirements.txt »

Ce fichier texte contient toutes les dépendances nécessaires à installer lors du déploiement de l'API et pour la faire fonctionner en utilisant le dashboard streamlit qui lui envoie des requêtes.