

Projet de Scoring de Crédit

Objectif du Projet

Ce projet vise à développer un modèle de scoring pour prédire la probabilité de défaut de paiement d'un client en utilisant des données historiques.

Structure du Projet

- `**`app/`` : Contient le code de l'API Flask permettant de servir les prédictions.
- `**`Notebooks/`` : Contient les notebooks utilisés pour l'analyse exploratoire, la modélisation, et le suivi des expérimentations avec MLFlow.
- `**`Data_drift/`` : Contient les fichiers d'analyse de data drift, réalisés avec Evidently.
- `**`saved_model/`` : Contient les modèles entraînés et les fichiers de données traités.
- `**`requirements.txt`` : Liste des dépendances Python nécessaires pour exécuter le projet.

3. `**Fichier `requirements.txt``

Le fichier ``requirements.txt`` doit inclure toutes les dépendances nécessaires à votre projet. Cela comprend les bibliothèques utilisées pour Flask, Streamlit, scikit-learn, etc.

`**Exemple de `requirements.txt`` :

```
```plaintext
Flask==2.0.1
streamlit==0.84.0
scikit-learn==0.24.2
joblib==1.0.1
pandas==1.3.1
shap==0.39.0
lightgbm==3.2.1
```

### ## Méthodologie

1. `**Exploration des Données` : Analyse exploratoire des données pour identifier les variables clés.
2. `**Prétraitement` : Transformation des données, gestion des valeurs manquantes, création de nouvelles features.
3. `**Modélisation` : Entraînement et évaluation de différents modèles (par exemple, LightGBM), optimisation des hyperparamètres.
4. `**Suivi des Expérimentations` : Utilisation de MLFlow pour suivre les versions du modèle et les expérimentations.
5. `**Déploiement` : Création d'une API Flask pour servir le modèle, déploiement sur le cloud.

### ## Déploiement

- L'API a été déployée en utilisant streamlit [url = "https://samiram-ux.streamlit.app/saved\_model/test\_streamlit.py"]

