Atelier 4 : Exposition de la Fonction Predict via FastAPI

Prérequis :

- Modèle entraîné et sauvegardé.
- Fichier makefile fonctionnel pour toutes les étapes précédentes

1. **Objectifs :**

   Exposer la fonction predict() comme un service REST avec FastAPI.

2. **Contenu :**

   Création d'un fichier app.py contenant :

   - ✓ Une route pour la prédiction (/predict).
   - ✓ Chargement du modèle entraîné depuis le disque.
   - ✓ Exemple de requête REST pour prédire un résultat.

3. **Livrables :** Une API REST fonctionnelle.
4. **Excelence :** Exposer la fonction retrain() comme service REST.

Tutoriel : Exposition de la Fonction Predict via FastAPI

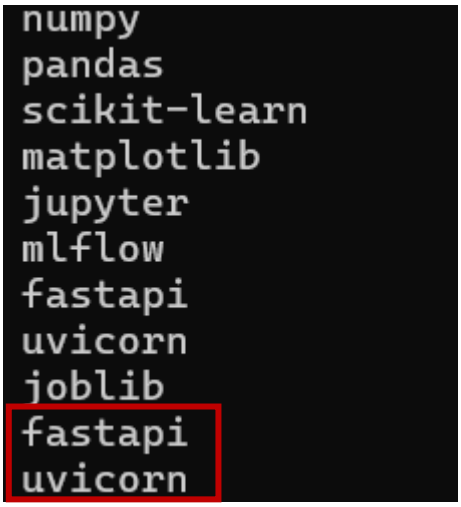
Introduction

Ce tutoriel vous guide dans la création d'un service REST permettant de réaliser des prédictions à l'aide de la fonction predict() d'un modèle machine learning. L'objectif est de transformer votre modèle en une API REST utilisable par d'autres applications.

I. Étape 01 : Préparer l'environnement

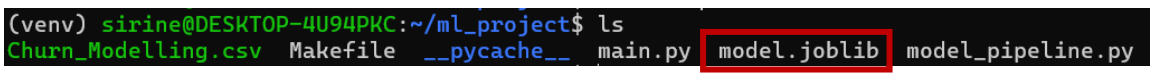
1) Installer FastAPI et Uvicorn :

- Exécuter les commandes suivantes dans votre environnement WSL :
  - « source venv/bin/activate » pour activer l'environnement virtuel
  - « pip install fastapi uvicorn » pour installer deux outils essentiels afin de développer et d'exécuter des applications web modernes et performantes avec Python.
- Ajouter si c'est pas encore fait ces dépendances au fichier requirements.txt : « fastapi » et « uvicorn »



2) Assurer que le modèle est déjà entraîné et sauvegardé :

Vous devez avoir un fichier modèle, par exemple model.joblib, sauvegardé avec la fonction save_model().



II. Étape 02 : Créer le fichier app.py

1) Créer le fichier « app.py » : `~/ml_project$ nano app.py`

2) Ajouter le code nécessaire pour:

- Définir une route HTTP POST pour effectuer des prédictions
- Utiliser le modèle pour prédire la sortie
- Retourner le résultat de la prédiction

III. Étape 03 : Tester l'API localement

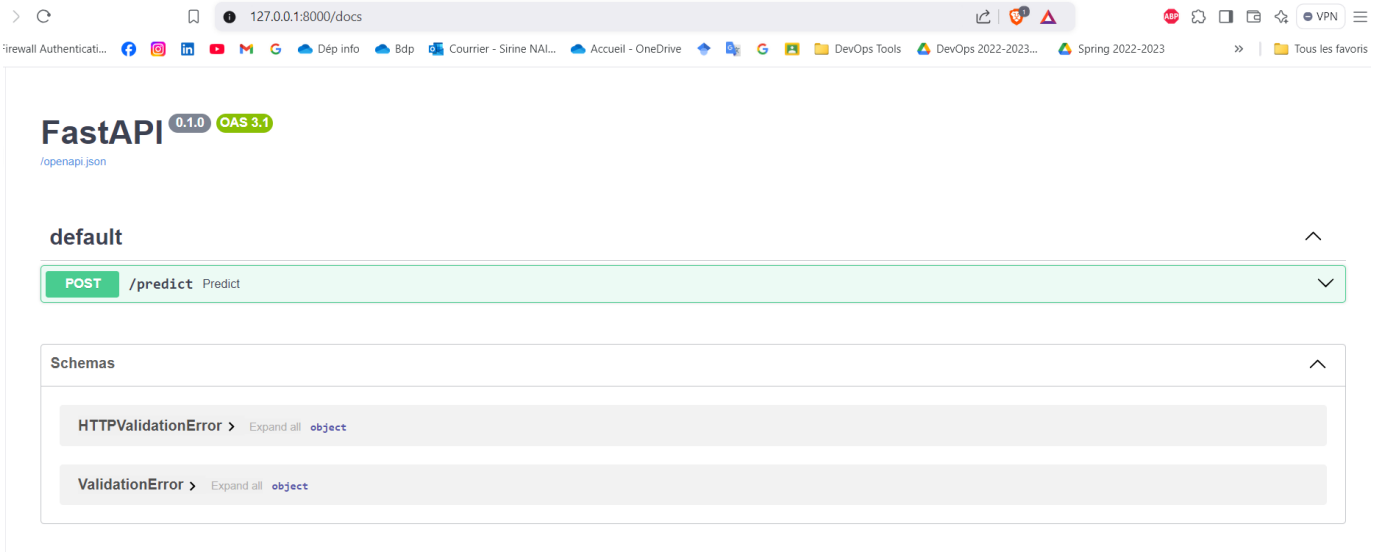
1) Démarrer le serveur FastAPI :

Lancer la commande « uvicorn app:app --reload --host 0.0.0.0 --port 8000 ». Cette commande démarre l'application FastAPI avec **Uvicorn**, active le rechargement automatique du serveur (--reload), et la rend accessible sur toutes les interfaces réseau (--host 0.0.0.0) au port 8000. Cela permet de tester l'API localement ou sur un réseau externe.

```
(venv) sirine@DESKTOP-4U94PKC:~/ml_project$ uvicorn app:app --reload --host 0.0.0.0 --port 8000
INFO:     Will watch for changes in these directories: ['/home/sirine/ml_project']
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [1563] using StatReload
Model loaded from model.joblib
INFO:     Started server process [1565]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

2) Accéder à la documentation interactive :

Ouvrir un navigateur et rendez-vous à : http://127.0.0.1:8000/docs



L'adresse http://127.0.0.1:8000/docs est donnée à titre indicatif. Merci de l'adapter selon l'adresse de votre machine dans l'environnement adéquat et le port utilisé.

3) Tester votre prédiction en exposant le service Rest avec FastAPI

Vérifier que votre end point est fonctionnel.

Excellence : Vous pouvez utiliser une solution web pour l'affichage du résultat selon les critères saisies (django, flask ou autres). Vous êtes libres dans ce cas d'exposer et de consommer le service avec les technologies de votre choix.

IV. Étape 04 : Ajout de l'étape de test API au niveau du Makefile

Ajouter la commande nécessaire dans le « Makefile » pour exécuter l'API (ouvrir l'interface swagger pour le test) :

V. Étape 05 : Vérification et Debug

1) Vérifier les logs : Si le modèle ne se charge pas correctement, vérifiez que le chemin dans MODEL_PATH est correct.
2) Gestion des erreurs : Le point de terminaison /predict renvoie des erreurs claires via HTTPException en cas de problème.

Livrables

1. **Fichier app.py** : Contient le service REST exposant la fonction predict().
2. **Documentation interactive** : Accessible via http://127.0.0.1:8000/docs.
3. **Commande Makefile** : Permet de démarrer l'API facilement.

Excellence :

Exposer la fonction retrain() :

a. Ajouter un point de terminaison /retrain qui permettra de réentraîner le modèle avec de nouveaux hyperparamètres.
b. Gérer les paramètres via des requêtes POST.