

ML4

Diabeast — Prédiction du diabète

Application web permettant de prédire la probabilité de diabète à partir de données médicales.

Elle combine une API **FastAPI** pour le backend et une interface **Streamlit** pour le frontend.

La communication entre les deux se fait via le module `requests`, et la configuration est gérée avec `python-dotenv`.

Démo en ligne

L'application est hébergée sur **Streamlit Cloud** :

👉 <https://diabetesufo-yrbtpc94zrlgmzt2j7bv4u.streamlit.app/>

Structure du projet

```
app/
├── api/
│   └── api.py           # Code principal de l'API FastAPI
├── model/
│   └── diabeast.pkl     # Modèle entraîné sauvegardé avec joblib
├── data/
│   └── ...              # Données sources ou exemples
├── front/
│   └── app.py           # Application Streamlit (frontend)
├──
└── README.md           # Documentation
```

Fonctionnement général

Le modèle de machine learning (`diabeast.pkl`) est chargé par le backend FastAPI.

L'application Streamlit (`front/app.py`) envoie les données saisies par l'utilisateur à l'API via `requests`.

L'URL de l'API est configurée via un fichier `.env` grâce à `python-dotenv`.

L'API renvoie la prédiction, affichée instantanément dans l'interface Streamlit.

Installation et exécution locale

1. Cloner le dépôt

```
git clone https://github.com/AnnaVitry/DiabetesUFO
cd app
```

2. Créer et activer un environnement virtuel

```
python -m venv venv
source venv/bin/activate      # macOS / Linux
venv\Scripts\activate        # Windows
```

3. Installer les dépendances

```
pip install -r requirements.txt
```

Le fichier `requirements.txt` doit inclure :

```
fastapi
streamlit
requests
python-dotenv
joblib
scikit-learn
pandas
numpy
```

4. Configurer les variables d'environnement

Crée un fichier `.env` dans le dossier `front/` avec, par exemple :

```
API_URL="http://127.0.0.1:8000/"
```

Dans `app.py`, l'URL de l'API est chargée via :

```
from dotenv import load_dotenv
import os
load_dotenv()
API_URL = os.getenv("API_URL")
```

5. Lancer l'API FastAPI

Depuis le dossier `app/api` :

```
fastapi run api.py
```

- L'API sera disponible sur `http://127.0.0.1:8000`

6. Lancer l'application Streamlit

Dans un autre terminal, depuis le dossier `app/front` :

```
streamlit run app.py
```

- L'application sera accessible sur `http://localhost:8501`

Exemple appel CURL

```
curl -X POST "http://127.0.0.1:8000/predict" \  
-H "Content-Type: application/json" \  
-d '{  
  "age": 45,  
  "gender": 1,  
  "polyuria": 1,  
  "polydipsia": 0,  
  "sudden_weight_loss": 0,  
  "weakness": 1,  
  "polyphagia": 0,  
  "genital_thrush": 1,  
  "visual_blurring": 0,  
  "itching": 0,  
  "irritability": 1,  
  "delayed_healing": 0,  
  "partial_paresis": 0,  
  "muscle_stiffness": 0,  
  "alopecia": 0,  
  "obesity": 1  
}'
```

Technologies utilisées

- **Python 3.10+**
- **FastAPI** — Backend API
- **Streamlit** — Interface utilisateur
- **Requests** — Communication entre frontend et backend
- **python-dotenv** — Gestion des variables d'environnement
- **Joblib / Scikit-learn** — Modélisation prédictive

- **Pandas / NumPy** — Manipulation des données

Librairie	Description courte	Commande d'installation	Utilisation principale
Streamlit	Bibliothèque open-source pour créer et partager rapidement des applications web interactives avec Python.	<code>pip install streamlit</code>	Interface utilisateur de l'application et visualisation des prédictions.
FastAPI	Framework web rapide et moderne pour construire des API REST avec Python.	<code>pip install fastapi</code>	Backend / API pour le modèle de prédiction.
Requests	Librairie simple et élégante pour effectuer des requêtes HTTP .	<code>pip install requests</code>	Communication entre Streamlit (frontend) et FastAPI (backend) .
python-dotenv	Charge les variables d'environnement depuis un fichier <code>.env</code> .	<code>pip install python-dotenv</code>	Gestion des paramètres de configuration (ex : URL de l'API).
Joblib	Outil pour la sérialisation efficace d'objets Python (ex : modèles ML).	<code>pip install joblib</code>	Chargement du modèle entraîné <code>diabeast.pkl</code> .
Scikit-learn	Bibliothèque phare du Machine Learning en Python.	<code>pip install scikit-learn</code>	Entraînement et évaluation du modèle prédictif.
Pandas	Bibliothèque pour la manipulation et l'analyse de données tabulaires .	<code>pip install pandas</code>	Gestion des données d'entrée et prétraitement.
NumPy	Outil de calcul scientifique pour les opérations numériques rapides .	<code>pip install numpy</code>	Support mathématique du modèle et des calculs.

Licence

Projet open-source.

Auteur

Projet réalisé par : 🧑‍🤝‍🧑 Groupe **"UFO"** composé de **Anna, Fidel et Olivier** — Promotion Simplon 2025 📅 17 Octobre 2025 📁 Module Machine Learning / Streamlit & FastAPI



Liens & Documentation

- [Application en ligne](#)
- [Documentation Streamlit](#) — pour comprendre comment créer et déployer des applications interactives en Python.
- [Dépôt GitHub](#) — code source du projet.