

ML3

Diabeast API — Prédiction du diabète avec FastAPI

API développée avec **FastAPI**, utilisant un modèle **RandomForestClassifier** sauvegardé avec **Joblib** pour prédire la présence probable de diabète chez un patient.

Structure du projet

```
app/
├── api/
│   ├── api.py # Code principal de l'API FastAPI
│   └── model/
│       ├── diabeast.pkl # Modèle entraîné sauvegardé avec joblib
│       └── data/
│           └── ...
└── README.md # Documentation
```

Installation

1. Créer un environnement virtuel

```
python3 -m venv .venv
source .venv/bin/activate # Linux/macOS
```

2. Installer les dépendances

```
pip install fastapi[standard] joblib numpy pydantic scikit-learn
```

Lancer l'API

Mode développement :

```
fastapi dev app/api/api.py
```

L'API sera accessible sur :

- <http://127.0.0.1:8000>

Documentation interactive :

- <http://127.0.0.1:8000/docs>

Navigation du code

1. Initialisation de l'application

```
app = FastAPI(title="Diabeast API - FastAPI", version="1.0")
```

Crée une **instance FastAPI** avec un titre et une **version visibles** dans la documentation Swagger.

2. Gestionnaire d'erreurs personnalisées

```
@app.exception_handler(RequestValidationError)
async def validation_exception_handler(request: Request, exc: RequestValidationError):
    return JSONResponse(
        status_code=422,
        content=jsonable_encoder({
            "error": "Invalid input data",
            "detail": exc.errors(),
            "body": exc.body
        })
    )
```

♦ Intercepte toutes les **erreurs** de validation (ex: mauvais type de données). ♦ Retourne un message JSON clair pour aider au **debug**.

3. Définition du modèle d'entrée (Pydantic)

```
class PatientData(BaseModel):
    age: int
    gender: int
    polyuria: int
    polydipsia: int
    sudden_weight_loss: int
    weakness: int
    polyphagia: int
    genital_thrush: int
    visual_blurring: int
    itching: int
    irritability: int
    delayed_healing: int
```

```
partial_paresis: int
muscle_stiffness: int
alopecia: int
obesity: int
```

Décrit le format exact attendu dans le **JSON d'entrée**. Chaque champ doit être un entier (0 ou 1).

4. Chargement du modèle entraîné

```
diabeast = joblib.load("model/diabeast.pkl")
```

Charge le modèle **RandomForestClassifier** entraîné pour la **prédiction**.

5. Routes principales

```
GET /
@app.get("/")
def root():
    return {"message": "Bienvenue sur l'API de prédiction du diabeast"}
```

Test rapide pour vérifier que le serveur est en ligne.

```
POST /predict
@app.post("/predict")
async def predict(data: PatientData):
```

Reçoit un objet **JSON** conforme au modèle PatientData.

Convertit les valeurs en **tableau NumPy**.

Prédit la classe (0 ou 1).

Retourne les **probabilités associées**.

Exemple de requête 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,
  "vaginal_discharge": 0
}
```

```
{
  "visual_blurring": 0,
  "itching": 0,
  "irritability": 1,
  "delayed_healing": 0,
  "partial_paresis": 0,
  "muscle_stiffness": 0,
  "alopecia": 0,
  "obesity": 1
}
```

Réponse attendue :

```
{
  "prediction": 1,
  "probabilities": {
    "negative": 0.12,
    "positive": 0.88
  }
}
```

Exemple d'erreur :

```
{
  "error": "Invalid input data",
  "detail": [
    {
      "type": "int_parsing",
      "loc": ["body", "age"],
      "msg": "Input should be a valid integer, unable to parse string as an integer",
      "input": "quarante"
    }
  ],
  "body": { "age": "quarante", "gender": 1, ... }
}
```




Bibliothèques ajoutées

Librairie	Description courte	Commande d'installation	Utilisation principale
FastAPI	Framework web moderne, rapide et asynchrone pour construire des API avec Python. Il repose sur les standards OpenAPI et JSON Schema	<code>pip install fastapi[standard]</code>	Création et déploiement d'API performantes pour exposer des modèles de machine learning ou tout autre service web, avec documentation interactive générée automatiquement

	JSON Schema, facilitant la documentation automatique et les interactions avec les modèles de machine learning.		automatiquement.
Pydantic	Bibliothèque pour la validation et la sérialisation de données en Python, utilisée par FastAPI pour gérer les modèles de données d'entrée et de sortie.	<code>pip install pydantic</code>	Définition de schémas de données stricts et validation automatique des requêtes JSON dans les API FastAPI.
NumPy	Bibliothèque fondamentale pour le calcul scientifique en Python, utilisée pour manipuler efficacement des tableaux et matrices numériques.	<code>pip install numpy</code>	Manipulation de tableaux et matrices, préparation des données pour les modèles de machine learning et calculs numériques rapides.
Joblib	Outil performant pour la sérialisation et la sauvegarde de modèles Python, notamment ceux créés avec Scikit-learn.	<code>pip install joblib</code>	Sauvegarde et restauration rapide de modèles entraînés pour le déploiement ou la réutilisation dans une API.
Pydantic	Bibliothèque pour la validation et la sérialisation de données en Python, utilisée par FastAPI pour gérer les modèles de données d'entrée et de sortie.	<code>pip install pydantic</code>	Définition de schémas de données stricts et validation automatique des requêtes JSON dans les API FastAPI.
Pathlib	Module standard de Python (intégré à partir de la version 3.4) permettant une gestion moderne et orientée objet des	<i>(inclus par défaut avec Python ≥ 3.4)</i>	Gestion simple et lisible des chemins vers les modèles, données et ressources dans le projet FastAPI.

chemins de fichiers
et répertoires.

Auteur

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

Liens Documentation

- **Joblib** : gestion et sérialisation des modèles Python
[Documentation officielle Joblib - load et dump](#)
- **Pydantic** : validation et sérialisation des données pour FastAPI
[Documentation officielle Pydantic - BaseModel](#)