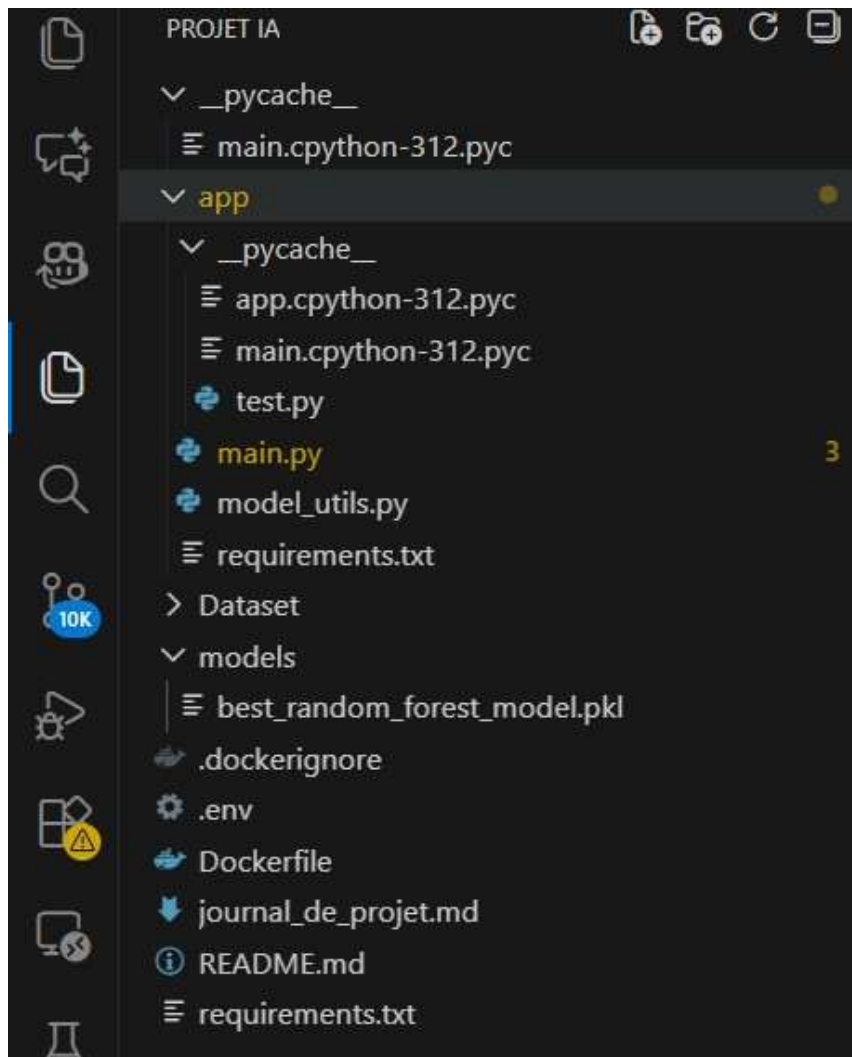


1) La structure du projet



2) Aperçu du démarrage de l'API (Fast API)

4) Structure du Docker

```
Dockerfile > ...
1 FROM python:3.10-slim (last pushed 1 week ago)
2
3 ENV PYTHONDONTWRITEBYTECODE=1
4 ENV PYTHONUNBUFFERED=1
5
6 WORKDIR /app
7
8 # Installer dépendances depuis la racine
9 COPY requirements.txt /app/requirements.txt
10 RUN pip install --upgrade pip
11 RUN pip install fastapi uvicorn
12 RUN pip install --no-cache-dir -r requirements.txt
13
14 # Copier le code
15 COPY app /app/app
16 COPY models /app/models
17
18 # Lancer l'API
19 CMD ["sh", "-c", "uvicorn app.main:app --host 0.0.0.0 --port ${PORT}"]
20
21
```

5) Run du docker

PROBLÈMES 1 SORTIE CONSOLE DE DÉBOGAGE TERMINAL PORTS JUPYTER

PS C:\Users\IBRAHIM TRAORE\Downloads\Projet IA> docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f8e864c12419	shopnow-api:latest	"uvicorn app.main:ap..."	10 hours ago	Up 40 seconds		stupefied_wilson
144c914f6d25	shopnow-api	"uvicorn app.main:ap..."	10 hours ago	Up 10 hours	0.0.0.0:8000->8000/tcp	shopnow-api

PS C:\Users\IBRAHIM TRAORE\Downloads\Projet IA>

6) Extrait du journal de projet.md

```

# journal_de_projet.md > ## # journal_de_projet.md > ## ## Date / version
1  ✓ # journal_de_projet.md
2
3  ✓ ## Objectif
4  Préparer et dockeriser l'API de prédiction de la satisfaction client (ShopNow).
5  📌
6  ✓ ## Date / version
7  2025-11-26 - Version 1.0
8
9  ✓ ## Étapes réalisées
10 1. Nettoyage et fusion des tables (notebook)
11 2. Entraînement RandomForest, optimisation RandomizedSearchCV
12 3. Export du modèle : models/best_random_forest_model.pkl
13 4. Conception de l'API FastAPI (app/app.py)
14 5. Dockerfile et .dockerignore créés
15 6. Tests locaux : uvicorn app:app --reload + endpoints /health et /predict OK
16 7. Construction et test du conteneur Docker (docker build, docker run)
17
18 ✓ ## Problèmes rencontrés
19 - MemoryError lors des merges + solution : Dask / lecture chunkée et sélection de colonnes
20 - Multiple errors Mflow 403 + solution : configuration token DAGshub
21 - Gestion de l'encodage des catégories : j'ai sauvegardé labelEncoder et intégré son chargement dans l'API.
22
23 ## Commandes clés
24 ✓ # build
25 docker build -t shopnow-api:latest .
26 ✓ # run
27 docker run -d --name shopnow-api -p 8000:8000 shopnow-api:latest
28 ✓ # test
29 curl http://127.0.0.1:8000/health
30 curl -X POST http://127.0.0.1:8000/predict -H "Content-Type: application/json" -d '{"delai_livraison":3,"retard_livraison":0,
    "nb_versements":1,"valeur_paiement":120.5,"prix_total":120.5,"nb_articles":1,"longueur_commentaire":50,
    "product_category_name":"electronics"}'

```