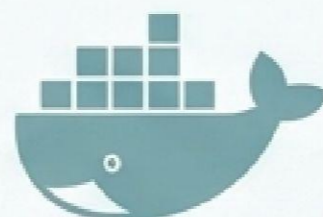
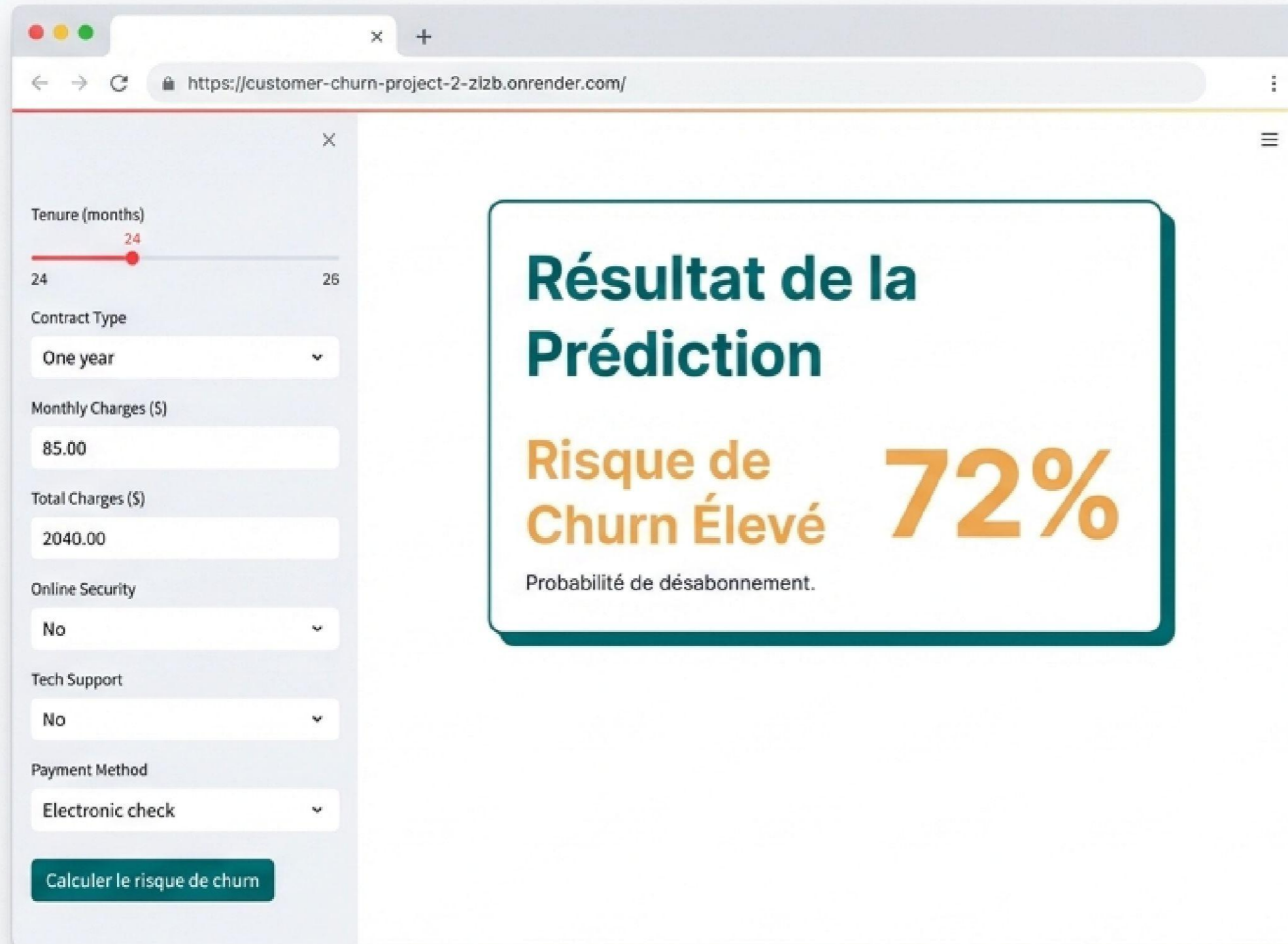


Prédiction du Churn Client

Conception et déploiement d'un pipeline
MLOps de bout en bout



Le Résultat d'Abord : L'Application en Action



https://customer-churn-project-2-zlzb.onrender.com/

Tenure (months)
24

Contract Type
One year

Monthly Charges (\$)
85.00

Total Charges (\$)
2040.00

Online Security
No

Tech Support
No

Payment Method
Electronic check

Calculer le risque de chum

Résultat de la Prédiction

Risque de Churn Élevé **72%**

Probabilité de désabonnement.

Démo en Ligne

<https://customer-churn-project-2-zlzb.onrender.com/>

Une application web interactive pour obtenir des prédictions de churn en temps réel et par lot (via fichier CSV). Hébergée sur Render.

Testez-la : Renseignez les caractéristiques d'un client et cliquez sur "Calculer le risque de churn" pour obtenir une prédiction instantanée.

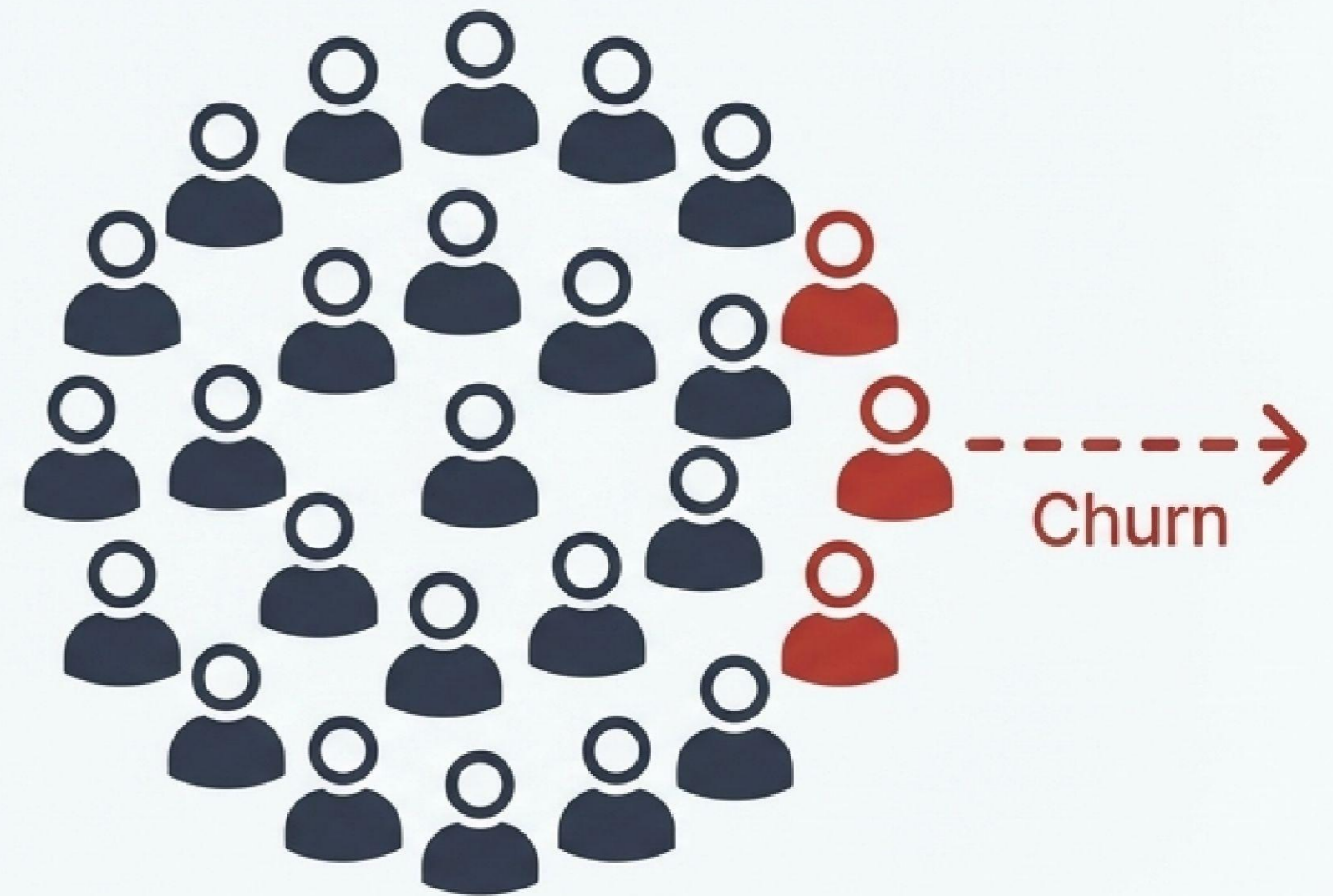
Le premier chargement peut prendre quelques secondes si le service est en veille (plan gratuit Render).

L'Enjeu Commercial : Anticiper l'Attrition pour Mieux Retenir

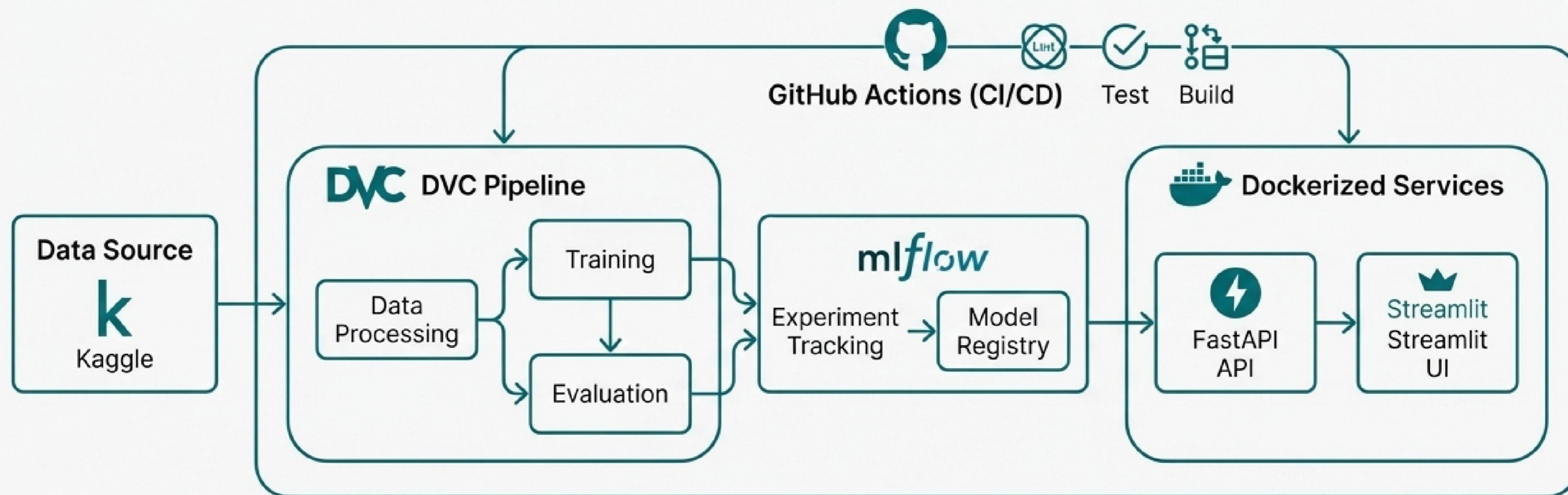
Problème : Le 'churn' client, ou attrition, est un défi majeur pour les entreprises basées sur l'abonnement. Perdre un client coûte bien plus cher que d'en acquérir un nouveau.

Objectif : Identifier de manière proactive les clients les plus susceptibles de résilier leur contrat pour permettre des actions de rétention ciblées et efficaces.

Dataset Utilisé : Basé sur le dataset public **Telco Customer Churn** de Kaggle, simulant un cas d'usage réel.



Une Architecture MLOps Robuste et Reproductible



1. Reproductibilité

Pipelines de données et d'entraînement versionnés.

2. Automatisation

Intégration et déploiement continus pour la qualité et la vitesse.

3. Déploiement

Services conteneurisés pour un déploiement fiable et isolé.

L'Arsenal Technologique : Les Meilleurs Outils pour Chaque Tâche



Hydra

Gestion centralisée et modulaire des configurations.



DVC

Versioning des données et orchestration du pipeline.

mlflow MLflow

Suivi des expériences, logging et registre de modèles.



Optuna

Optimisation automatique des hyperparamètres.



API de prédiction haute performance.



Interface utilisateur web interactive.



Orchestration multi-services pour le déploiement.

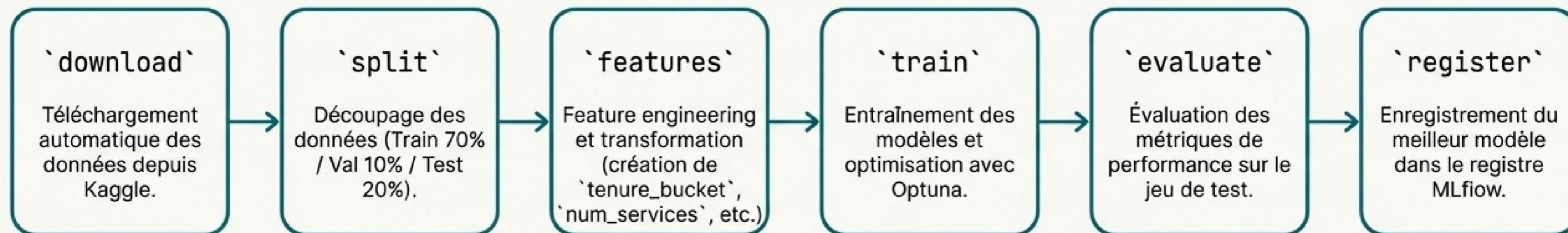


GitHub Actions

Automatisation des tests et du déploiement (CI/CD).

Un Pipeline de Données Maîtrisé de Bout en Bout avec DVC

Chaque étape est versionnée, reproductible et exécutable avec une seule commande : `dvc repro`.



```
stages:
  train:
    cmd: python src/models/train_model.py
    deps:
      - src/models/train_model.py
      - data/processed/train_features.csv
    params:
      - train
    outs:
      - artifacts/models/
  evaluate:
    cmd: python src/models/evaluate_model.py
    deps:
      - artifacts/models/
      - data/processed/test_features.csv
    metrics:
      - reports/metrics.json:
          cache: false
```

La Configuration Centralisée : Pilotage Précis avec Hydra



Modularité : Séparez les configurations (données, modèle, entraînement) pour plus de clarté.



Flexibilité : Modifiez les paramètres depuis la ligne de commande sans toucher au code.



Reproductibilité : La configuration est versionnée avec le code, garantissant la traçabilité des expériences.

```
configs/model.yaml
```

```
# @package _group_
```

```
# Model configuration
```

```
model_name: lightgbm
```

```
# Features to be used for training
```

```
features:
```

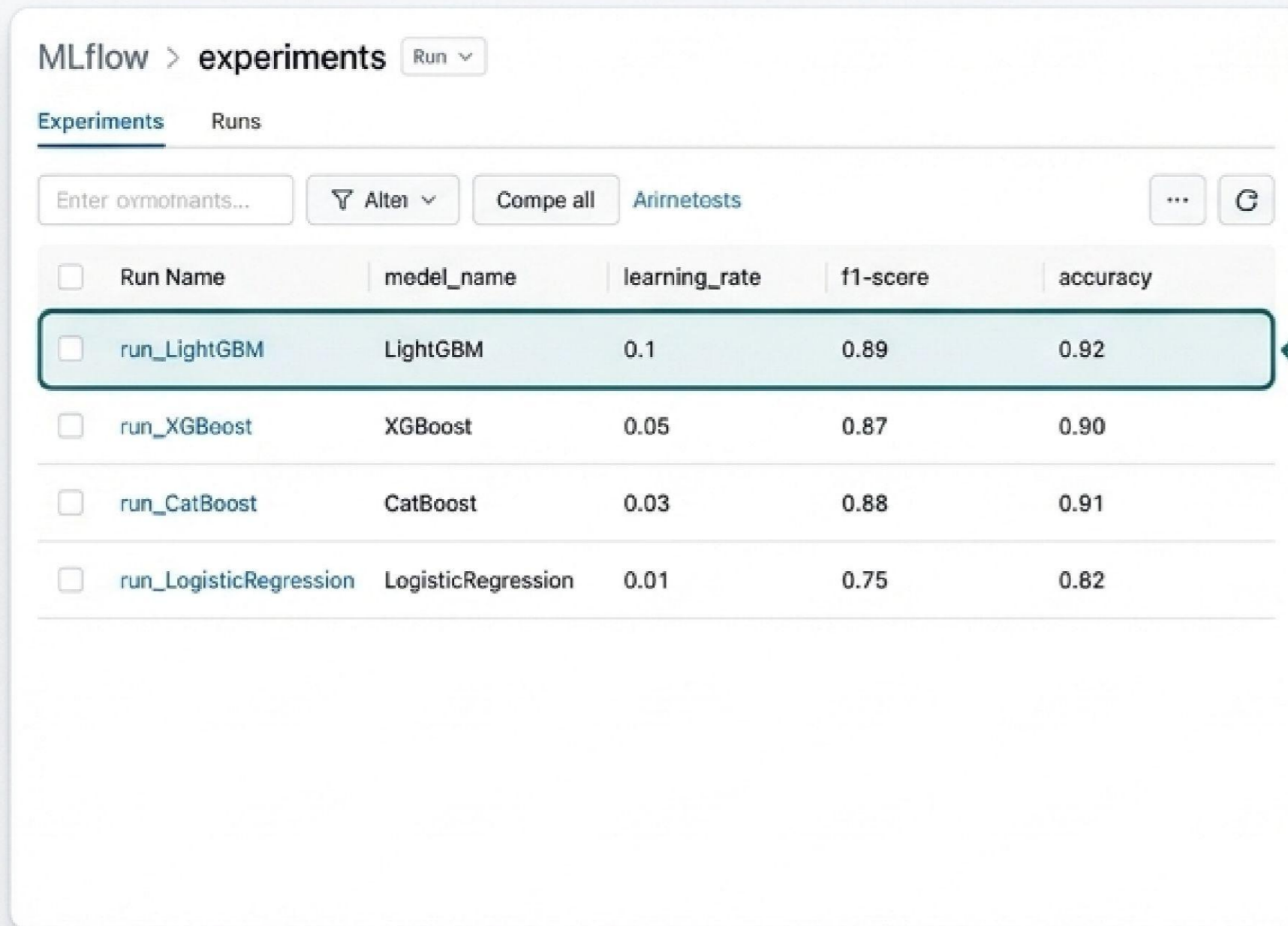
- tenure
- MonthlyCharges
- TotalCharges
- Contract
- tenure_bucket
- num_services

```
# Target variable
```

```
target_col: Churn
```

Exemple de configuration du modèle. Hydra assemble dynamiquement les configurations nécessaires à l'exécution.

Maîtriser le Chaos : Suivi d'Expériences Rigoureux avec MLflow



MLflow > experiments Run ▾

Experiments Runs

Enter keywords... Alter ▾ Compare all Arithmetic ... G

<input type="checkbox"/>	Run Name	model_name	learning_rate	f1-score	accuracy
<input type="checkbox"/>	run_LightGBM	LightGBM	0.1	0.89	0.92
<input type="checkbox"/>	run_XGBoost	XGBoost	0.05	0.87	0.90
<input type="checkbox"/>	run_CatBoost	CatBoost	0.03	0.88	0.91
<input type="checkbox"/>	run_LogisticRegression	LogisticRegression	0.01	0.75	0.82

Meilleur
Modèle

Logging Automatique

Chaque exécution du pipeline (`dvc repro`) logue automatiquement les paramètres, les métriques et les artefacts (modèle, graphiques).

Comparaison Facile

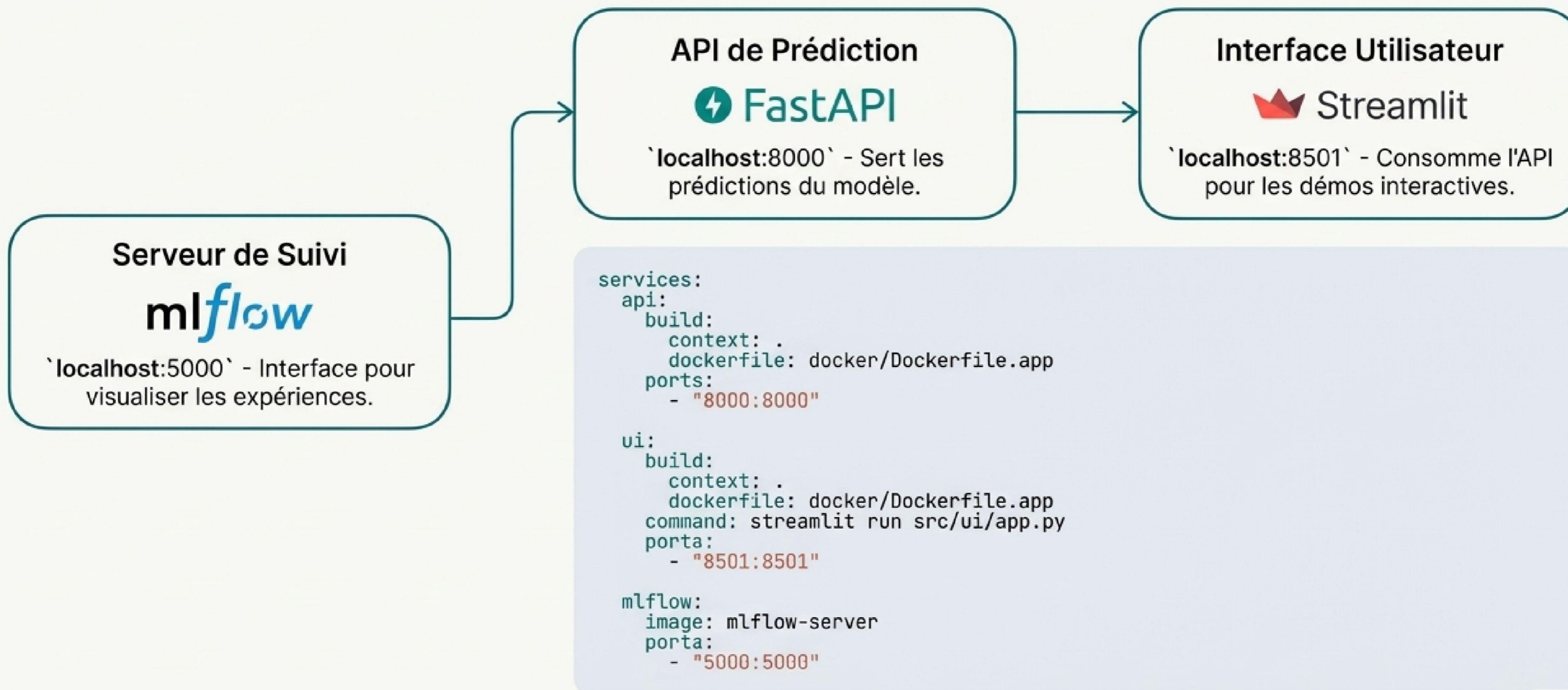
L'interface permet de comparer les performances des différents modèles testés (LightGBM, XGBoost, CatBoost, Régression Logistique).

Registre de Modèles

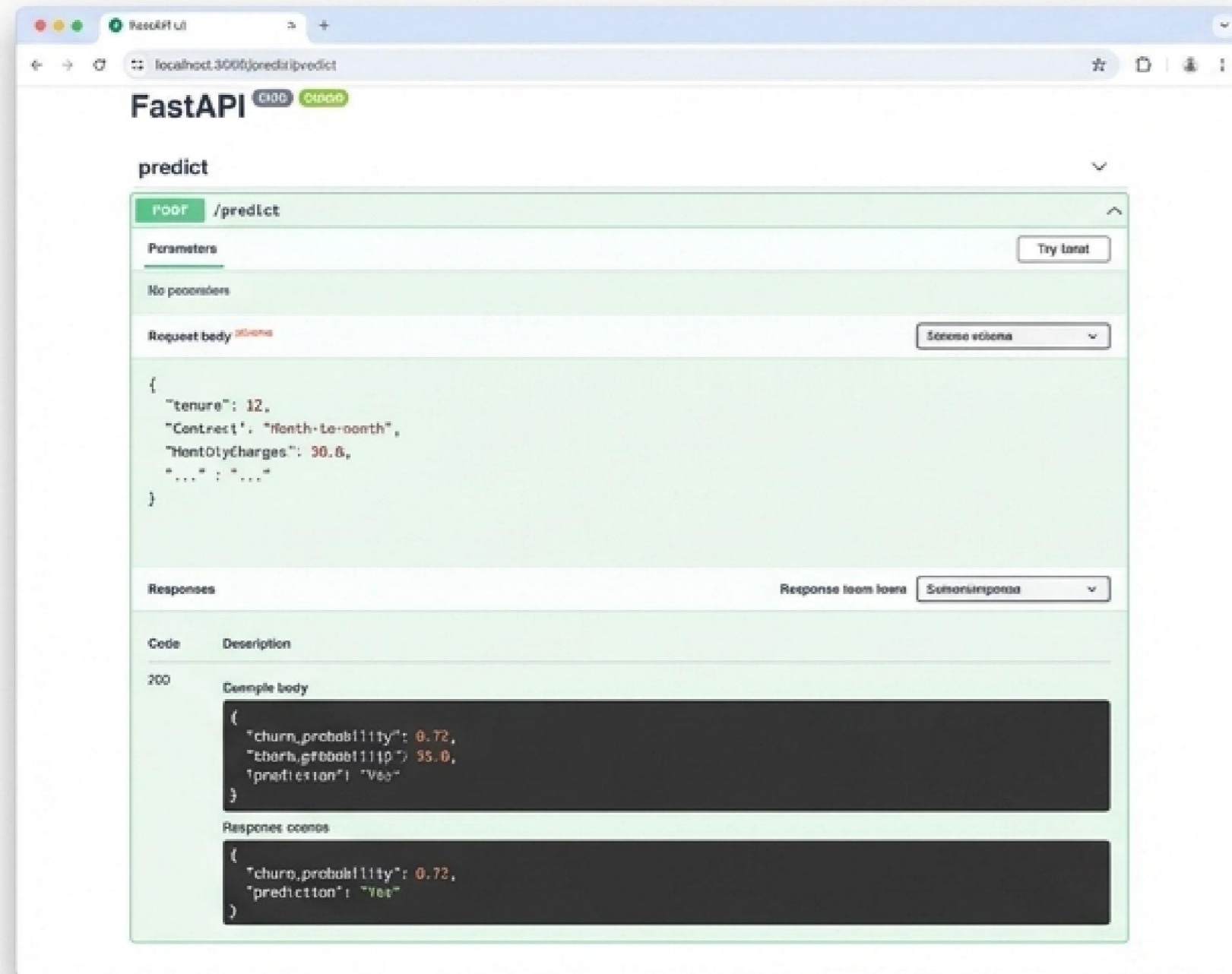
Le meilleur modèle est promu dans le registre, le versionnant pour le déploiement.

Un Écosystème de Services Conteneurisés avec Docker

L'ensemble de l'application est orchestré localement via `docker compose` pour un développement et un déploiement cohérents.



Une API de Prédiction Performante et Documentée via FastAPI



`POST /predict`

Exemple de Requête

```
{
  "tenure": 12,
  "Contract": "Month-to-month",
  "MonthlyCharges": 50.0,
  "...": "..."
}
```

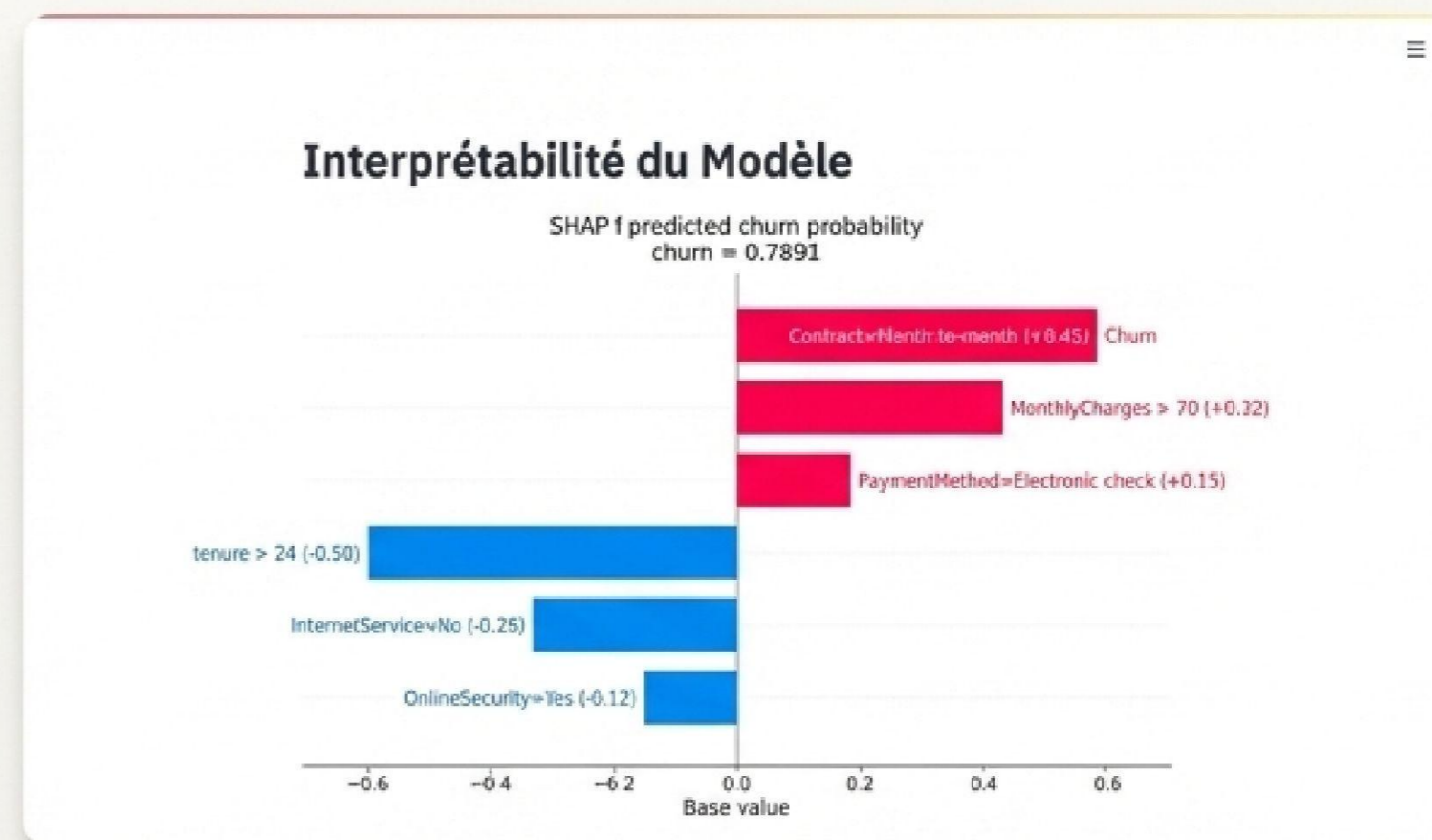
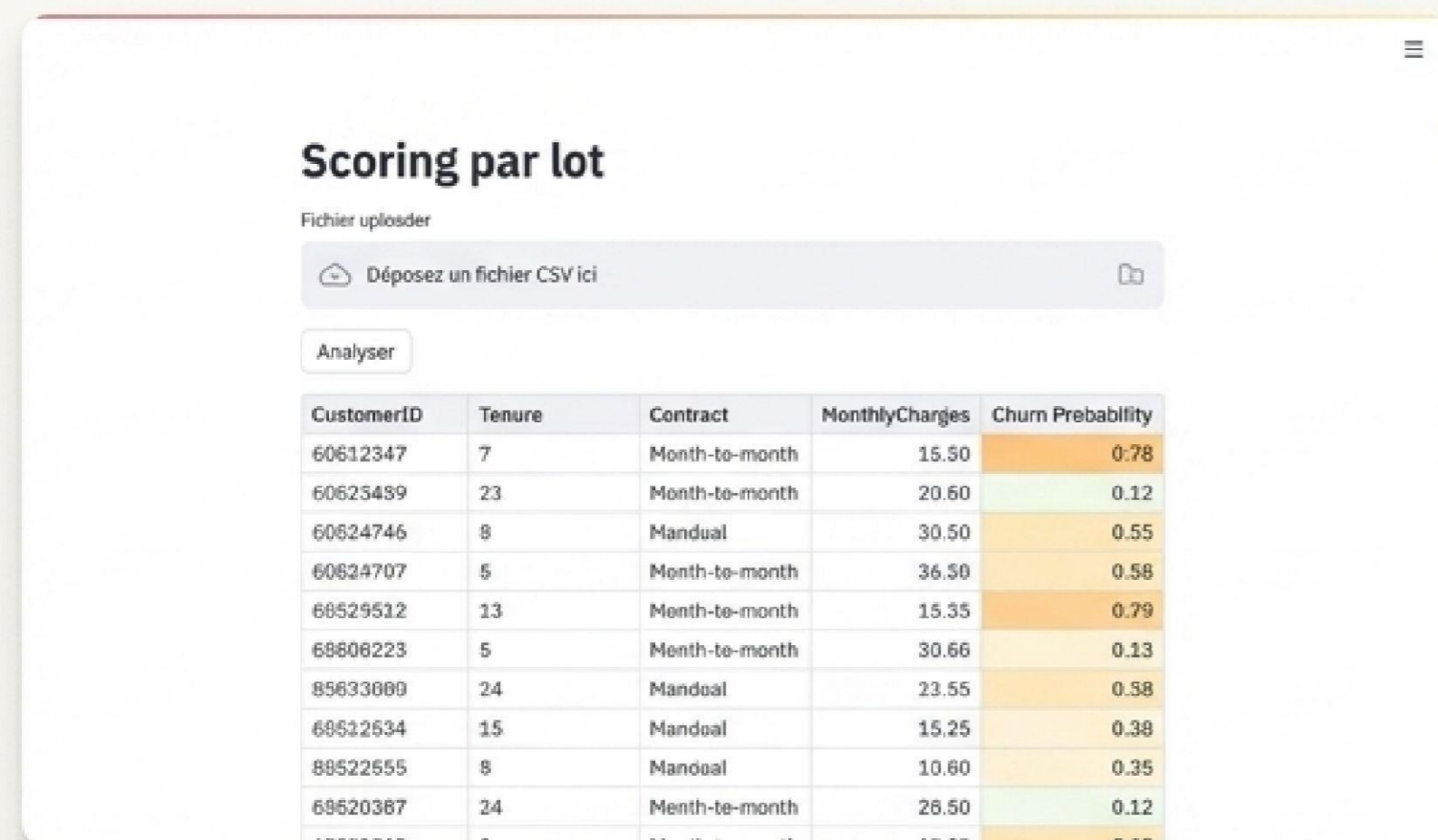
Exemple de Réponse

```
{
  "churn_probability": 0.72,
  "prediction": "Yes"
}
```

FastAPI offre des performances élevées, une validation automatique des données et une documentation interactive (Swagger UI) générée automatiquement.

Au-delà de la Prédiction : Interprétabilité et Analyse par Lot

L'application Streamlit n'est pas seulement une vitrine, mais un outil d'analyse.

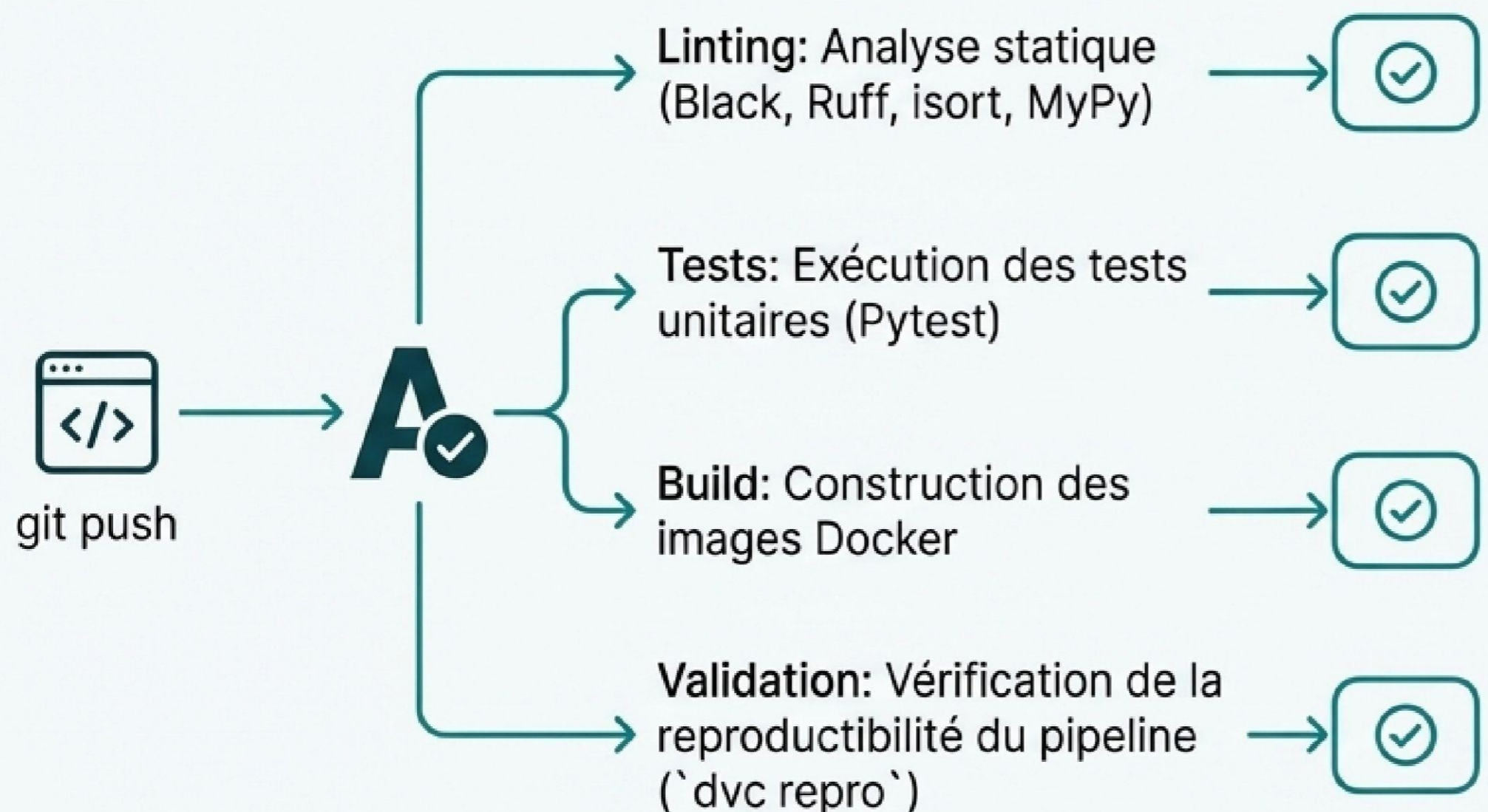


✓ **Prédictions Individuelles :**
Saisie manuelle des caractéristiques d'un client.

✓ **Scoring par Lot :** Import d'un fichier CSV pour obtenir des prédictions pour plusieurs clients.

✓ **Interprétabilité du Modèle :**
Visualisation des facteurs de risque principaux via l'intégration de SHAP, expliquant le 'pourquoi' derrière chaque prédiction.

L'Excellence Automatisée : Qualité et Déploiement Continu (CI/CD)



Objectifs du Workflow

- “Garantir la Qualité”:
Chaque modification est automatiquement validée.
- “Prévenir les Régressions”:
Les tests unitaires assurent la stabilité du code.
- “Assurer la Reproductibilité”:
Le pipeline est testé à chaque commit pour garantir des résultats cohérents.

Une Fondation Solide : Structure, Qualité et Tests

Structure du Projet

```
Customer_Churn_Project/  
├── configs/  
├── data/  
├── src/  
├── tests/  
├── compose.yaml  
├── dvc.yaml  
└── pyproject.toml
```

Une structure claire et modulaire pour une maintenance et une évolution facilitées.

Gestion des Dépendances



Gestion des dépendances isolée et déterministe avec Poetry.

```
poetry install
```

Qualité du Code







Un ensemble d'outils pour garantir la robustesse et la lisibilité du code.

```
make test
```

```
make lint
```

Un Projet MLOps Complet, Prêt à l'Emploi

Résumé des Points Forts

-  • **Pipeline de Bout en Bout:** De la donnée brute à l'application web déployée.
-  • **100% Reproducible:** Le versioning des données, du code et des configurations garantit des résultats cohérents.
-  • **Automatisé et Robuste:** CI/CD et tests pour une qualité continue.
-  • **Prêt à Déployer:** Conteneurisé avec Docker et configurable pour le cloud (Render).

Appel à l'Action



Explorez le Code

github.com/Souley225/Customer_Churn_Project



Contribuez: Les contributions sont les bienvenues. Forkez le repository et ouvrez une Pull Request.