

Détection de Phishing par NLP

Une analyse d'URL atteignant 97% de précision en moins de 5ms.

Une performance validée sur 549 000 URLs

Précision
globale

97.2%

ROC-AUC

99.1%

F1-Score
(phishing)

97.2%

Rappel
(phishing)

97.6%

Précision
(phishing)

96.8%

Latence de prédiction ultra-faible

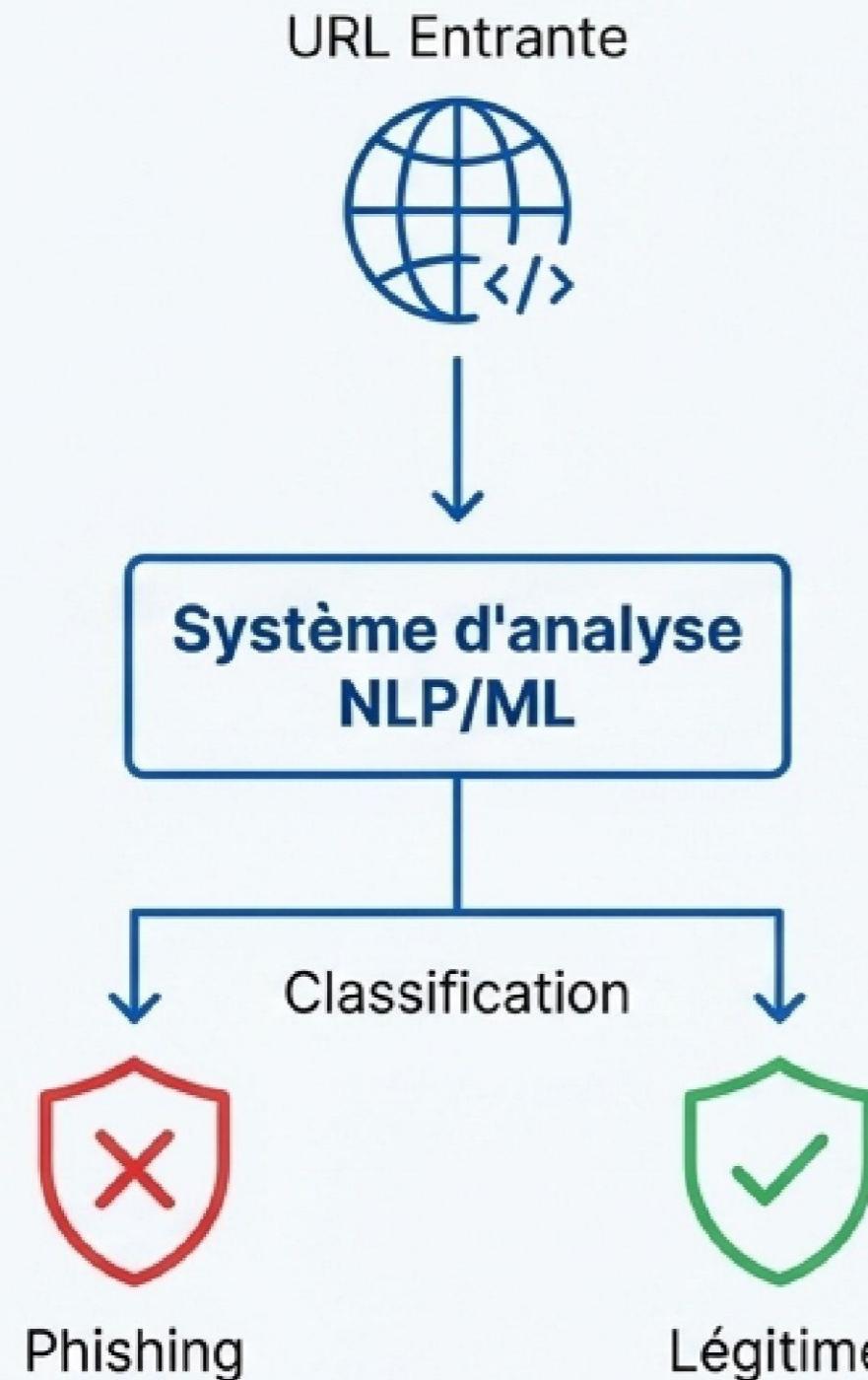
< 5ms

pour une URL unique

< 500ms

pour un batch de 1000 URLs

Un système complet, de l'analyse à l'intégration



Détection autonome

Analyse basée uniquement sur la chaîne de caractères de l'URL, sans appel réseau externe.



API REST

Endpoint FastAPI robuste pour une intégration facile dans d'autres services.



Interface Web

Dashboard Streamlit pour des démonstrations et des tests manuels rapides.



Tracking ML

Suivi et reproductibilité des expériences avec MLflow.



Optimisation Automatisée

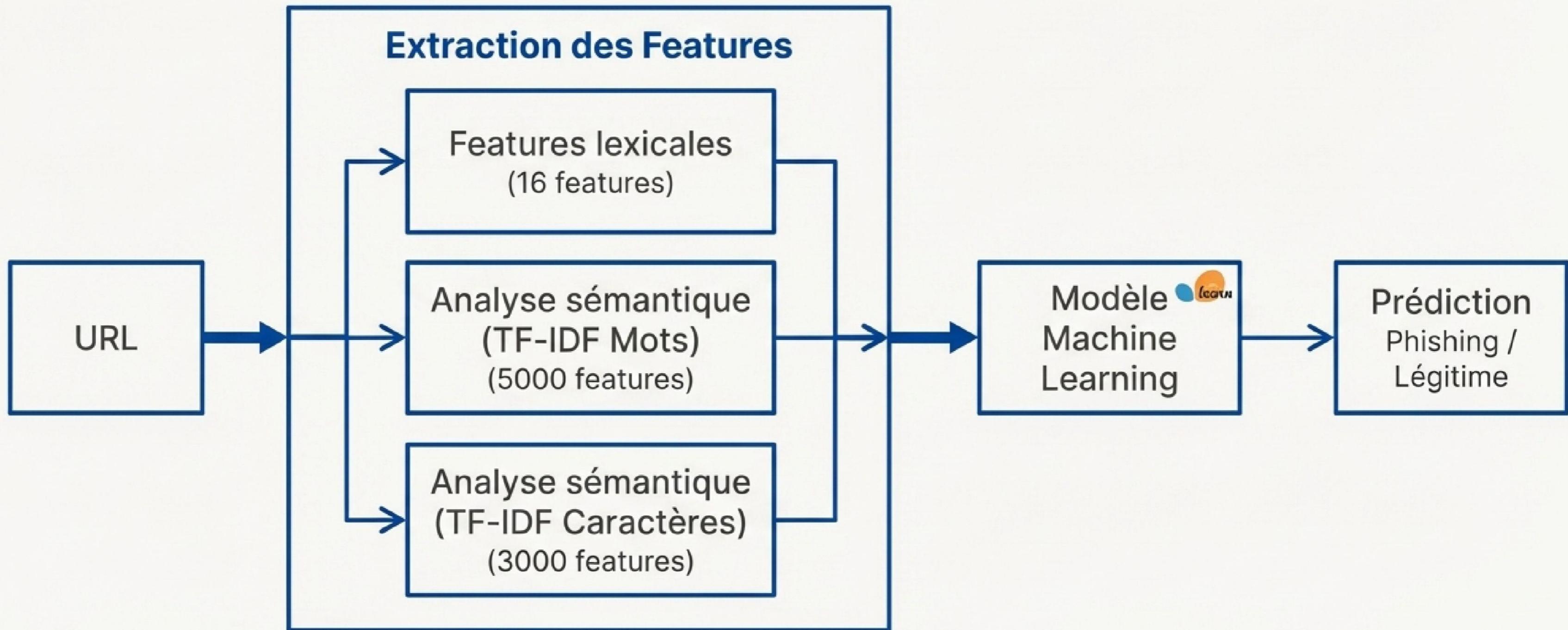
Recherche des meilleurs hyperparamètres via Optuna.



Prêt pour le déploiement

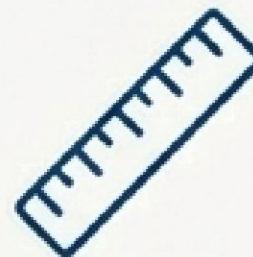
Configurations pour Docker et Render incluses.

Architecture du pipeline de prédition



L'analyse lexicale : déceler les signatures structurelles

16 features conçues pour capturer les caractéristiques non-sémantiques d'une URL.



Longueur de l'URL

Le nombre total de caractères.



Ratio de chiffres

La proportion de chiffres par rapport aux lettres.



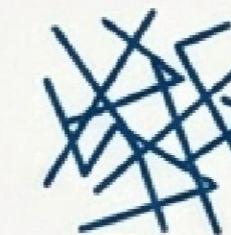
Profondeur du chemin

Le nombre de '/' dans le chemin.



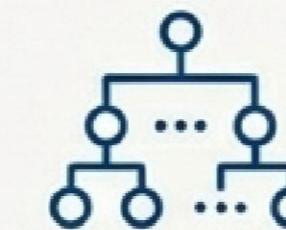
TLD suspect

Vérifie si le Top-Level Domain est dans une liste à risque.



Entropie

La complexité et l'aléa de la chaîne de caractères.



Nombre de sous-domaines

Le nombre de ':' dans le nom d'hôte.

1.2.3.4

Présence d'une adresse IP

Déetecte si le nom d'hôte est une IP.

@ ?
& =

Présence de caractères spéciaux

Déetecte la présence de '@', '?', '&', '='.

L'analyse sémantique : comprendre le langage du phishing

Transformer le texte de l'URL en un espace vectoriel de plus de 8000 dimensions pour le modèle.

Analyse par Mots (TF-IDF)

Capture les mots-clés fréquemment associés au phishing (ex: 'login', 'secure', 'account', 'verify').

5000

features

Analyse par Caractères (TF-IDF)

Déetecte les patterns subtils et les tentatives d'obfuscation (ex: 'paypal' vs 'paypal') qui échappent à l'analyse par mots.

3000

features

Un processus d'entraînement rigoureux et automatisé



Suivi des expériences avec MLflow

Chaque entraînement est tracé, enregistrant les paramètres, les métriques et le modèle final pour une reproductibilité totale.



Optimisation des hyperparamètres avec Optuna

Le système recherche automatiquement la meilleure combinaison d'hyperparamètres pour maximiser la performance du modèle.



Personnalisation facile

Les paramètres clés comme le nombre d'essais Optuna (`train.n_trials=100`) ou le seed (`train.seed=42`) sont configurables via la ligne de commande.

Un modèle entraîné sur un dataset vaste et équilibré

Source des données

Kaggle - "Phishing Site URLs"

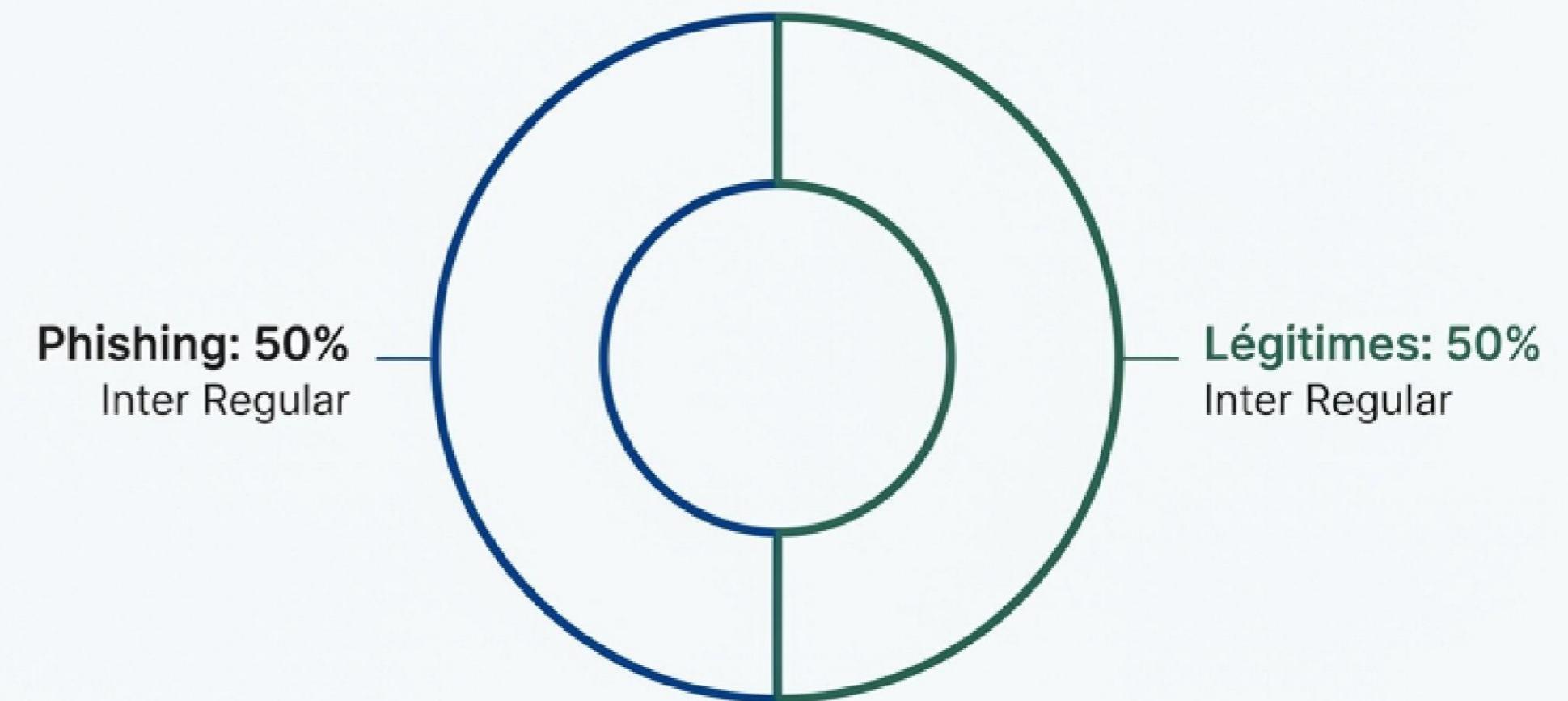
Volume

549 000

URLs uniques

Composition

Parfaitement équilibré avec **50%** d'URLs de phishing et **50%** d'URLs légitimes.



Intégrez la détection via une API REST performante

Construit avec **FastAPI** pour une haute performance et une documentation automatique.

 Base URL: `http://localhost:8000`

Endpoints principaux

 **GET /health**

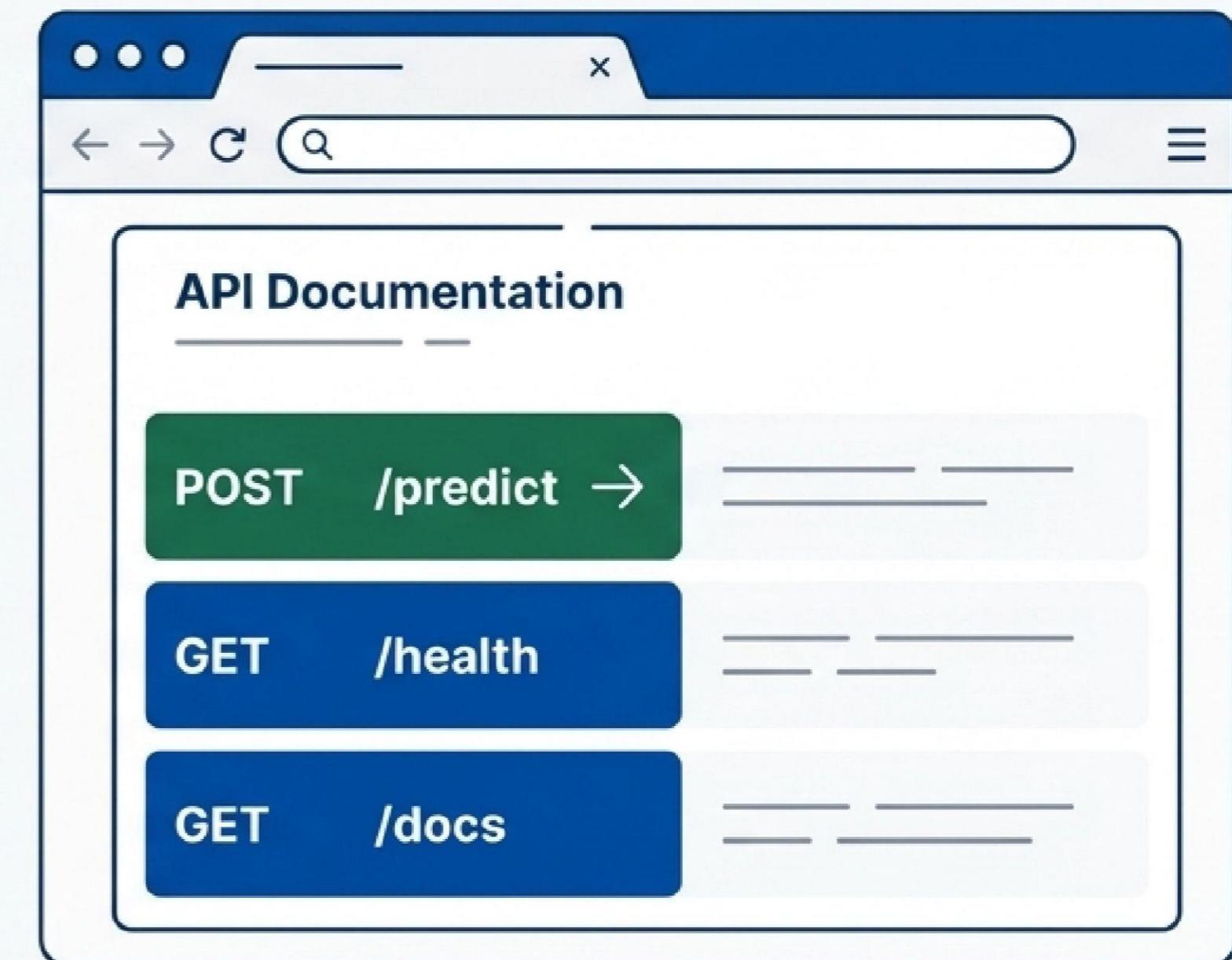
Pour la vérification de santé du service
(monitoring, health checks).

 **POST /predict**

Pour soumettre une URL et recevoir une
prédiction en temps réel.

 **GET /docs**

Pour accéder à la documentation interactive de
l'API (Swagger UI).



L'API en action : un exemple concret

Effectuez une prédiction avec une simple commande `curl`.

```
curl -X POST http://localhost:8000/predict \
  -H "Content-Type: application/json" \
  -d '{"url": "http://paypal-verify.tk/account"}'
```

```
{
  "url": "http://paypal-verify.tk/account",
  "prediction": 1,
  "label": "phishing",
  "proba_phishing": 0.94,
  "proba_legitimate": 0.06
}
```

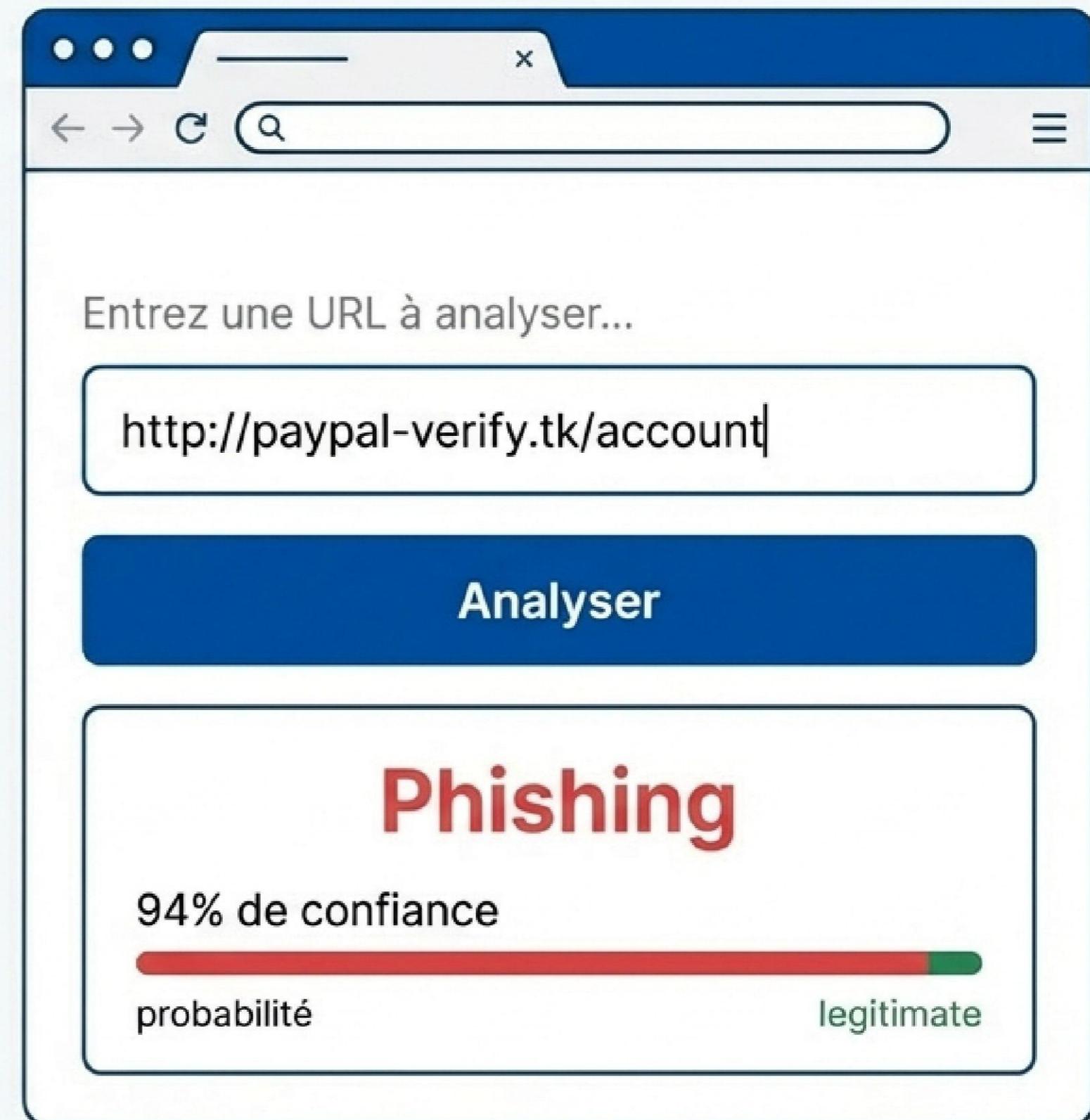
Testez le modèle interactivement avec l'interface web

Dashboard simple et réactif construit avec Streamlit.

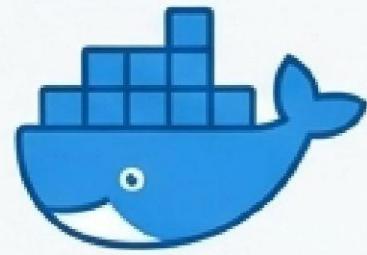
Fonctionnalité

Permet de coller une URL dans un champ de texte et d'obtenir instantanément la prédiction du modèle, la probabilité et une explication simple.

Commande de lancement : `make ui` (lance le service sur le port 8501)



Déployez le système en quelques minutes, localement ou sur le cloud



Déploiement Local avec Docker

Lancez l'ensemble des services (API, UI, MLflow) avec une seule commande.

```
docker compose up --build
```

- API: <http://localhost:8000>
- UI: <http://localhost:8501>
- MLflow: <http://localhost:5000>



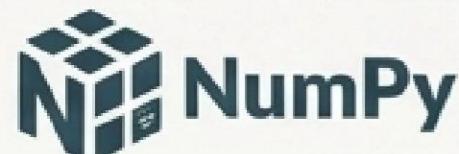
Déploiement Cloud sur Render

Le fichier `render.yaml` configure automatiquement les services sur la plateforme Render après connexion d'un repository GitHub.

1. Pousser le code sur GitHub.
2. Connecter le repo sur Render.
3. Ajouter les secrets KAGGLE_USERNAME et KAGGLE_KEY.

La stack technique complète

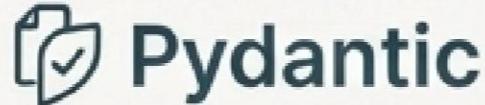
ML / NLP



MLOps



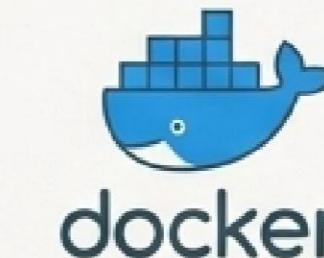
API



UI



DevOps



Qualité du code



Un workflow local optimisé avec Makefile

Des commandes simples pour gérer l'ensemble du cycle de vie du projet.

-  make download Télécharger le dataset depuis Kaggle.
-  make train Lancer l'entraînement complet du modèle.
-  make evaluate Évaluer les performances du modèle sur le jeu de test.
-  make serve Démarrer l'API FastAPI.
-  make ui Démarrer l'interface Streamlit.
-  make mlflow Démarrer l'interface de tracking MLflow.
-  pytest tests/ -v Exécuter la suite de tests unitaires.

Ressources du projet et contact

Projet



- Repository GitHub:
github.com/Souley225/NLP_Phishing_detection_Project
- Dataset Source: Phishing Site URLs sur Kaggle



Inspirations



Articles de recherche sur
'BERT for Phishing
Detection'



'URL-based Features'



'Hybrid NLP Approach'

Auteur

Souleymane Sall



sallsouleymane
2207@gmail.com



LinkedIn



Medium