Readme

📦 README - Manuel de Build et de Lancement des Projets OpenFoodFacts

🚀 Introduction

Ce document fournit un guide détaillé pour le **build**, le **déploiement** et le **lancement** des différents projets liés au pipeline de données **OpenFoodFacts**. Il inclut la configuration des environnements, la gestion des dépendances, ainsi que la définition des arguments nécessaires pour chaque script.

📁 Structure du Projet

```
openfoodfacts_pipeline/
├── data/
│   ├── raw/
│   ├── clean/
│   └── enriched/
├── dags/
│   └── airflow_dag.py
├── scripts/
│   ├── ingestion.py
│   ├── cleaning.py
│   └── enrichment.py
├── config/
│   └── settings.yaml
├── README.md
└── requirements.txt
```

⚙️ Pré-requis

- **Python 3.8+**

- **Apache Spark 3.x**
- **Apache Airflow 2.x**
- **HDFS (Hadoop)**

📦 Installation des dépendances

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

🗂️ 1. Script d'Ingestion des Données

📜 Description

Ce script permet d'ingérer des données depuis l'API OpenFoodFacts ou des fichiers CSV et de les stocker sur HDFS.

▶️ Commande de Lancement

```
python scripts/ingestion.py --source api --output hdfs:///user/ubuntu/off_raw/
```

⚙️ Arguments

Argument	Type	Description	Valeur par défaut
`--source`	string	Source des données (`api` ou `csv`)	`api`
`--output`	string	Chemin HDFS de sortie	`hdfs:///off_raw`
`--limit`	int	Nombre d'enregistrements à ingérer	`1000`

Exemple :

```
python scripts/ingestion.py --source csv --output hdfs:///off_raw --limit 5000
```

🧹 2. Script de Nettoyage des Données

📜 Description

Ce script nettoie les données ingérées : suppression des doublons, gestion des valeurs manquantes, et standardisation des types.

▶️ Commande de Lancement

```
python scripts/cleaning.py --input hdfs:///user/ubuntu/off_raw/ --output
hdfs:///user/ubuntu/off_clean/
```

⚙️ Arguments

Argument	Type	Description	Valeur par défaut
`--input`	string	Chemin des données brutes sur HDFS	`hdfs:///off_raw`
`--output`	string	Chemin des données nettoyées sur HDFS	`hdfs:///off_clean`
`--drop-null`	bool	Supprimer les enregistrements vides	`True`

Exemple :

```
python scripts/cleaning.py --input hdfs:///off_raw --output hdfs:///off_clean
--drop-null False
```

✨ 3. Script d'Enrichissement des Données

📜 Description

Ce script enrichit les données en effectuant des jointures avec des datasets externes et en calculant de nouveaux indicateurs.

▶️ Commande de Lancement

```
python scripts/enrichment.py --input hdfs:///user/ubuntu/off_clean/ --output
hdfs:///user/ubuntu/off_enriched/ --external hdfs:///external_data/
```

⚙️ Arguments

Argument	Type	Description	Valeur par défaut
`--input`	string	Chemin des données nettoyées	`hdfs:///off_clean`
`--output`	string	Chemin des données enrichies sur HDFS	`hdfs:///off_enriched`
`--external`	string	Chemin des données externes pour jointure	`hdfs:///external_data`

Exemple :

```
python scripts/enrichment.py --input hdfs:///off_clean --output
hdfs:///off_enriched --external hdfs:///external_data
```

⏱️ 4. Orchestration avec Apache Airflow

📜 Description

Airflow est utilisé pour automatiser et orchestrer l'ensemble du pipeline.

▶️ Lancement du DAG Airflow

```
airflow dags trigger -e 2024-01-01 openfoodfacts_pipeline
```

⚙️ Configuration des DAGs

```
from airflow import DAG
from airflow.operators.bash import BashOperator
from datetime import datetime

default_args = {
    'owner': 'airflow',
    'start_date': datetime(2024, 1, 1),
    'retries': 1
}

dag = DAG(
    'openfoodfacts_pipeline',
    default_args=default_args,
    schedule_interval='@daily'
)
```

```
ingest = BashOperator(
    task_id='ingest_data',
    bash_command='python scripts/ingestion.py --source api --output
hdfs:///off_raw',
    dag=dag
)

clean = BashOperator(
    task_id='clean_data',
    bash_command='python scripts/cleaning.py --input hdfs:///off_raw --output
hdfs:///off_clean',
    dag=dag
)

enrich = BashOperator(
    task_id='enrich_data',
    bash_command='python scripts/enrichment.py --input hdfs:///off_clean --
output hdfs:///off_enriched',
    dag=dag
)

ingest >> clean >> enrich
```

📝 Variables d'Environnement

```
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export SPARK_HOME=/usr/local/spark
export AIRFLOW_HOME=~/airflow
```

🗑️ Nettoyage des Données Temporaire

Pour nettoyer les données temporaires générées :

```
hdfs dfs -rm -r /user/ubuntu/off_temp/
```

🚨 Dépannage

Problème	Cause possible	Solution
Erreur de connexion HDFS	Mauvaise config HDFS	Vérifiez `core-site.xml` et `hdfs-site.xml`
DAG Airflow bloqué	Scheduler non démarré	`airflow scheduler`
Job Spark lent	Manque de ressources	Augmentez les ressources via `spark-submit`

👤 Auteur

- **Nom :** [Votre Nom]
- **Date :** [Date de création]
- **Contact :** [Votre Email]