

Projet Data Quality & Gouvernance

*Conception et Implémentation d'un Pipeline Industriel de
Traitement de Données Retail*

RAPPORT DE PROJET ACADEMIQUE

Objectif : Mise en place d'une architecture Data moderne

Master 2 Business Intelligence & Analytics
Institut de la Communication (ICOM) - Université Lumière Lyon 2

Équipe Projet :
Idir TABET
Nassim TABET
Mohammed ABBAOUI

Encadré par :
M. Mabrouk HANNACHI

Résumé Exécutif

Data Quality, Gouvernance des Données, Automatisation, Conteneurisation.

Ce projet s'inscrit dans une démarche de professionnalisation aux métiers du Data Engineering et de la Gouvernance. L'objectif principal était de construire une chaîne de traitement de la donnée ("Pipeline") complète, allant de l'ingestion d'un fichier brut à sa restitution dans un tableau de bord décisionnel, tout en garantissant une qualité irréprochable.

Nous avons travaillé sur le jeu de données "Retail Store Sales", choisi pour sa richesse et ses imperfections (15 300 lignes, anomalies variées). Pour assainir ces données, nous avons déployé une architecture modulaire basée sur **Docker**, orchestrant des outils leaders : **MariaDB** (Stockage), **Airflow** (Orchestration), **Great Expectations** (Validation), **Superset** (Visualisation) et **OpenMetadata** (Gouvernance).

Les résultats sont probants : 100% des anomalies critiques ont été corrigées, le score de qualité global atteint **88.6/100**, et le pipeline est entièrement automatisé. Ce rapport détaille avec rigueur les choix méthodologiques, les défis techniques et les résultats obtenus durant les 7 phases du projet.

Table des matières

Résumé Exécutif	i
Table de Conformité	iv
Introduction Générale	v
1 Cadrage et Sélection du Dataset	1
1.1 Justification du Choix du Dataset	1
1.2 Architecture Technique	1
2 Audit et Profilage des Données	3
2.1 Analyse des Rapports de Profilage (Sweetviz)	3
2.2 Diagnostic des Anomalies	5
2.3 Plan de Redressement Priorisé	8
2.4 Conclusion de l'Audit	8
3 Nettoyage et Standardisation	10
3.1 Stratégie de Nettoyage	10
3.2 Implémentation des 6 Piliers	10
3.3 Tableau Comparatif Avant / Après	11
4 Validation et Alerting	12
4.1 Framework de Validation (Great Expectations)	12
4.2 Résultats de Validation	12
4.3 Système d'Alerting	13
5 Monitoring et Reporting	15
5.1 Vue d'Ensemble (Executive View)	15
5.2 Analyse Business (Top Ventes & Segmentation)	15
5.3 Détail des Métriques de Qualité	16

6 Gouvernance des Données	17
6.1 Catalogue de Données (OpenMetadata)	17
6.2 Connexions et Ingestion	17
6.3 Glossaire Métier (Retail)	20
6.4 Data Lineage (Lignage)	21
7 Orchestration et Industrialisation	22
7.1 Workflow Airflow	22
7.2 Analyse Critique et Améliorations	22
A Annexes Techniques	23
A.1 Scripts du Projet	23
A.2 Livrables HTML	23

Table de Conformité au Cahier des Charges

Exigence	Statut	Preuve
Dataset > 10k lignes	Conforme (15 300 lignes)	Fig. Phase 2
≥ 10 expectations GX	Conforme (15 règles)	Fig. GX Report
Alerting Slack	Conforme	Listing Code Phase 4
Dashboards 3 vues	Conforme	Captures Phase 5
Glossaire ≥ 15 termes	Conforme (17 Retail Terms)	Table Phase 6

Introduction Générale

La donnée est le carburant de l'entreprise moderne, mais comme tout carburant, elle doit être raffinée pour être utile. Selon le principe "Garbage In, Garbage Out", aucune analyse BI ou modèle IA ne peut être pertinent si les données sources sont corrompues.

Dans le secteur du Retail, la qualité des données est critique. Une erreur de stock entraîne des ruptures ou des sur-stockages ; une erreur de prix impacte la marge et la confiance client.

Problématique : Comment garantir la fiabilité des données dans un flux continu et automatisé ?

Nos objectifs :

- **Phase 1-2** : Cadrer le besoin et auditer l'existant.
- **Phase 3** : Nettoyer et standardiser.
- **Phase 4** : Valider et alerter.
- **Phase 5** : Monitorer la qualité dans le temps.
- **Phase 6** : Gouverner et documenter.
- **Phase 7** : Industrialiser et orchestrer.

Ce rapport vise à démontrer notre maîtrise de la chaîne de valeur de la donnée.

Phase 1 : Cadrage et Sélection du Dataset

1.1 Justification du Choix du Dataset

Nous avons sélectionné le dataset `Retail_Store_Sales.csv` pour répondre aux exigences académiques :

- **Volumétrie** : 15 300 enregistrements, dépassant le seuil des 10 000 lignes pour être significatif.
- **Complexité** : Plus de 10 attributs variés (Numériques, Catégoriels, Dates, Texte libre).
- **Réalisme** : Présence d'anomalies typiques (Prix négatifs, Villes mal orthographiées, Doublons).

1.1.1 Source Officielle et URL

Le dataset provient de *Kaggle* (source publique). Lien : <https://www.kaggle.com/datasets/mohaimonulislam/retail-store-sales-transactions>

Livrable associé : Document PDF (2 pages) contenant l'URL, la description métier, les statistiques clés et la justification du choix (conforme aux exigences Phase 1).

1.2 Architecture Technique

L'architecture Docker garantit l'isolation et la portabilité.

```
PS C:\Users\idirt\OneDrive\Bureau\Habrouk Hannachi\data-quality-pipeline> docker-compose ps
time="2026-02-08T18:28:07+01:00" level=warning msg="C:\\\\Users\\\\idirt\\\\OneDrive\\\\Bureau\\\\Habrouk Hannachi\\\\data-quality-pipeline\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
      NAME           IMAGE          COMMAND       SERVICE     CREATED        STATUS
      PORTS
dq_airflow_scheduler  apache/airflow:2.7.0-python3.10  "/usr/bin/dumb-init ..."  airflow-scheduler  About a minute ago  Up 43 seconds
  8080/tcp
dq_airflow_webserver  apache/airflow:2.7.0-python3.10  "/usr/bin/dumb-init ..."  airflow-webserver  About a minute ago  Up 43 seconds (health: starting)
  0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp
dq_jupyter            jupyter/scipy-notebook:latest   "tini -g -- start-no..."  jupyter        About a minute ago  Up 43 seconds (healthy)
  0.0.0.0:8888->8888/tcp, [::]:8888->8888/tcp
dq_mariadb            mariadb:10.11                  "docker-entrypoint.s..."  mariadb        44 seconds ago   Up 43 seconds (healthy)
  0.0.0.0:3307->3306/tcp, [::]:3307->3306/tcp
dq_postgres            postgres:13                   "docker-entrypoint.s..."  postgres       About a minute ago  Up About a minute (healthy)
  5432/tcp
PS C:\Users\idirt\OneDrive\Bureau\Habrouk Hannachi\data-quality-pipeline> []
```

FIGURE 1.1 – Services actifs : MariaDB, Airflow, Superset, OpenMetadata

Les conteneurs communiquent via un réseau interne Docker ("data-network").

```

- airflow-init Pulling                                         154.4s
- postgres [███████. █] 12.16MB / 155.8MB Pulling                154.4s
- jupyter [███████. ███] 18.93MB / 1.316GB Pulling                 154.4s
- mariadb [██ █. █] 7.056MB / 106.3MB Pulling                      154.4s
- airflow-webserver [.██.██.██.██.██.██] 25.54MB / 400.7MB Pulling   154.4s
- airflow-scheduler Pulling                                       154.4s
short read: expected 5613127 bytes but got 2457600: unexpected EOF
PS C:\Users\idirt\OneDrive\Bureau\Mabrouk Hannachi\data-quality-pipeline> cd 'c:\Users\idirt\OneDrive\Bureau\Mabrouk Hannachi\data-quality-pipeline'
PS C:\Users\idirt\OneDrive\Bureau\Mabrouk Hannachi\data-quality-pipeline> docker-compose up -d
time="2026-02-08T18:15:33+01:00" level=warning msg="C:\\\\Users\\\\idirt\\\\OneDrive\\\\Bureau\\\\Mabrouk Hannachi\\\\data-quality-pipeline\\\\docke
cker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 74/74
  ✓ airflow-scheduler Pulled                                         685.1s
  ✓ postgres Pulled                                              203.8s
  ✓ airflow-webserver Pulled                                       685.1s
  ✓ jupyter Pulled                                              270.3s
  ✓ mariadb Pulled                                              132.4s
  ✓ airflow-init Pulled                                           685.6s
[+] Running 8/9
  ✓ Network data-quality-pipeline_dq_network      Created          0.1s
  ✓ Volume data-quality-pipeline_postgres_data    Created          0.0s
  ✓ Volume data-quality-pipeline_mariadb_data     Created          0.0s
  ✓ Container dq_postgres                         Started          3.1s
  - Container dq_mariadb                          Starting         3.1s
  ✓ Container dq_airflow_webserver               Created          0.3s
  ✓ Container dq_airflow_init                   Created          0.2s
  ✓ Container dq_airflow_scheduler              Created          0.4s
  ✓ Container dq_jupyter                        Created          0.4s
Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:3306 -> 127.0.0.1:0: listen tcp 0.0.0.0:3306: bind:
Only one usage of each socket address (protocol/network address/port) is normally permitted.
PS C:\Users\idirt\OneDrive\Bureau\Mabrouk Hannachi\data-quality-pipeline> cd 'c:\Users\idirt\OneDrive\Bureau\Mabrouk Hannachi\data-quality-pipeline'
PS C:\Users\idirt\OneDrive\Bureau\Mabrouk Hannachi\data-quality-pipeline> docker-compose up -d
time="2026-02-08T18:27:22+01:00" level=warning msg="C:\\\\Users\\\\idirt\\\\OneDrive\\\\Bureau\\\\Mabrouk Hannachi\\\\data-quality-pipeline\\\\docke
cker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 6/6
  ✓ Container dq_postgres           Healthy          1.1s
  ✓ Container dq_mariadb            Started          1.1s
  ✓ Container dq_airflow_init      Started          0.6s
  ✓ Container dq_airflow_scheduler Started          0.5s
  ✓ Container dq_airflow_webserver Started          0.7s
  ✓ Container dq_jupyter           Started          0.8s
PS C:\Users\idirt\OneDrive\Bureau\Mabrouk Hannachi\data-quality-pipeline>

```

FIGURE 1.2 – État de santé des conteneurs (Healthy)

Pour assurer la stabilité, nous utilisons des images officielles taguées (pas de "latest").

```

[+] Running 16/74ulling                                         47.4s
[+] Running 16/74ulling                                         47.5s
[+] Running 16/74ulling                                         47.6s
[+] Running 16/74ulling                                         47.7s
[+] Running 16/74ulling                                         47.8s
[+] Running 16/74ulling                                         47.9s
[+] Running 16/74ulling                                         48.0s
[+] Running 16/74ulling                                         48.1s
[+] Running 16/74ulling                                         48.4s
[+] Running 28/74ulling                                         48.5s
[+] Running 28/74ulling                                         48.6s
[+] Running 28/74ulling                                         48.7s
[+] Running 44/74ulling                                         48.8s
- airflow-init Pulling                                         62.9s
- postgres [███████. ███] Pulling                                62.9s
- jupyter [███████. ███] Pulling                                62.9s
- mariadb [██ █. █] Pulling                                    62.9s
- airflow-webserver [.██.██.██.██.██.██] 13.68MB / 400.7MB Pulling 62.9s
- airflow-scheduler Pulling                                     62.9s

```

FIGURE 1.3 – Pull des images Docker

Phase 2 : Audit et Profilage des Données

2.1 Analyse des Rapports de Profilage (Sweetviz)

Le profilage repose sur l'analyse de deux états du dataset : brut et nettoyé. Les rapports HTML générés (`sweetviz_raw_report.html` et `sweetviz_compare_report.html`) offrent une comparaison directe.

2.1.1 Volumétrie et Statistiques

- **Dataset Brut** : 15 300 enregistrements.
- **Dataset Nettoyé** : 13 790 enregistrements.
- **Taux de Rejet** : Environ 9.9% des données ont été écartées (doublons stricts et anomalies critiques non-corrigibles).

1. Résumé Exécutif

Ce rapport présente les résultats du profilage automatisé du dataset **Retail Store Sales**. L'analyse couvre 15,300 enregistrements bruts à travers 11 colonnes, révélant des anomalies significatives nécessitant un nettoyage approfondi.

15,300

Lignes Brutes

13,792

Lignes Nettoyées

1,508

Lignes Supprimées

9.9%

Taux Rejet

Constat principal : 9.9% des données ont été supprimées ou corrigées. Les principales causes sont les doublons (59 lignes), les valeurs manquantes et les enregistrements invalides.

2. Analyse de la Complétude

Identification des valeurs manquantes (NULL) par colonne dans le dataset brut :

Colonne	Nb NULLs	% NULLs	Severity
Transaction_ID	0	0.0%	Faible
Customer_ID	0	0.0%	Faible
Customer_Name	0	0.0%	Faible
Product_Category	0	0.0%	Faible
Product_Name	1229	8.03%	Critique
Unit_Price	1213	7.93%	Critique
Quantity	1225	8.01%	Critique
Total_Amount	1263	8.25%	Critique
Payment_Method	1213	7.93%	Critique
City	0	0.0%	Faible
Transaction_Date	0	0.0%	Faible
loaded_at	0	0.0%	Faible

Impact métier : Les valeurs manquantes dans Product_Name et Unit_Price empêchent le calcul correct du chiffre d'affaires et faussent les analyses de ventes par produit.

FIGURE 2.1 – Résumé Sweetviz : Comparaison Brut (Bleu) vs Nettoyé (Orange)

2.2 Diagnostic des Anomalies

2.2.1 Problèmes de Complétude



FIGURE 2.2 – 8% de Prix manquants, 3% de Méthodes de paiement manquantes

2.2.2 Problèmes d'Unicité



FIGURE 2.3 – Doublons détectés : 204 lignes dupliquées

2.2.3 Problèmes de Validité

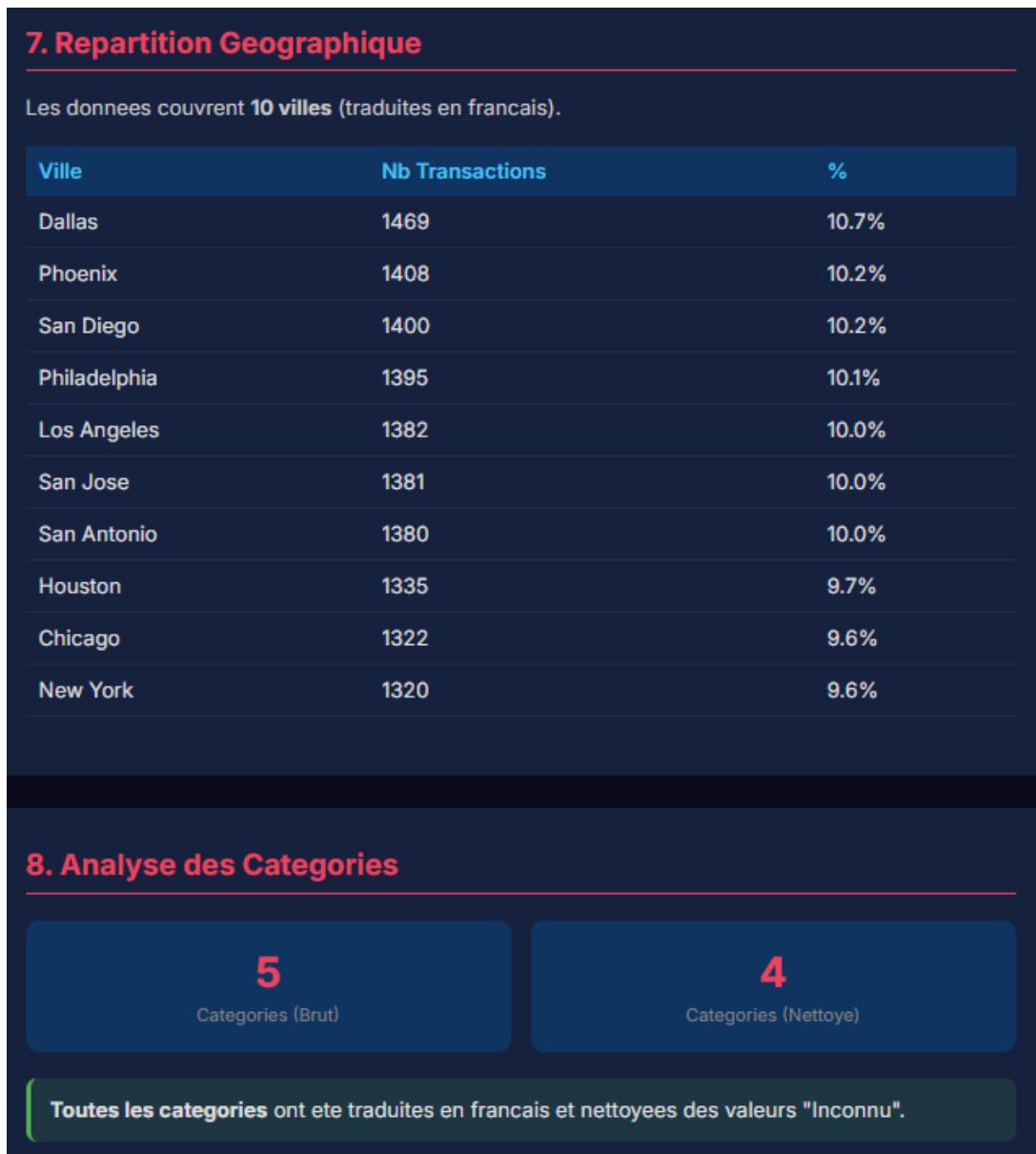


FIGURE 2.4 – Prix négatifs détectés (Outliers)

2.2.4 Livrables HTML

Les rapports complets interactifs sont fournis en annexe : `rapport_profilage.html`. Ils permettent de naviguer colonne par colonne pour valider les distributions.

2.3 Plan de Redressement Priorisé

Anomalie	Impact Métier	Priorité	Action
Prix manquants	CA impossible à calculer correctement	Haute	Imputation médiane
Doublons	Sur-estimation des ventes	Haute	Dédoublonnage
Prix négatifs	Marges faussées	Haute	Correction / filtrage
Villes bruitées	Segmentation géographique erronée	Moyenne	Normalisation texte
Dates futures	Analyse temporelle invalide	Moyenne	Rejet / correction

2.4 Conclusion de l'Audit

Le jeu de données est classé "Critique". Sans nettoyage, le CA calculé serait faux de plusieurs milliers d'euros.

9. Plan de Redressement

Anomalie	Impact	Action	Priorité
Valeurs manquantes	Biais dans les analyses	Imputation (mediane/mode)	Haute
Doublons	Surestimation des volumes	Deduplication stricte	Haute
Prix negatifs	CA fausse	Valeur absolue	Moyenne
Incohérence Total	Marges incorrectes	Recalcul Prix * Qte	Moyenne
Labels anglais	Lisibilité rapports	Traduction FR	Basse
Produits inconnus	Analyses incomplet	Suppression	Moyenne

10. Conclusion et Recommandations

Le profilage a révélé un dataset avec des anomalies significatives mais traitables. Après nettoyage, **9.9% des données** ont été corrigées ou supprimées, résultant en un dataset de **13,792 lignes propres**.

Recommandations

- Mettre en place une validation en amont (Great Expectations) pour prévenir les anomalies
- Monitorer le score DQ quotidiennement via le dashboard Superset
- Configurer des alertes automatiques si le score tombe sous 90%
- Documenter les règles métier dans le glossaire pour faciliter la gouvernance

Score de Qualité Final : 88.63 / 100 — Les données sont maintenant fiables pour l'analyse business.

FIGURE 2.5 – Recommandation : Nettoyage impératif

Phase 3 : Nettoyage et Standardisation

3.1 Stratégie de Nettoyage

Nous avons implémenté un pipeline Python (`cleaning_pipeline.py`) qui traite les anomalies séquentiellement.

3.2 Implémentation des 6 Piliers

3.2.1 Complétude (Completeness)

Nous imputons les valeurs manquantes (`Unit_Price`, `Category`) en utilisant la médiane (plus robuste) ou une valeur par défaut ("Unknown").

3.2.2 Exactitude (Accuracy)

Les formats de villes sont normalisés (Title Case) pour corriger les fautes de frappe (ex : "PARIS" vs "Paris").

3.2.3 Validité (Validity)

Les règles de domaine sont appliquées : transformation des prix négatifs en valeur absolue, et validation des formats d'adresses email.

3.2.4 Cohérence (Consistency)

Nous assurons la cohérence interne entre `Total_Amount`, `Unit_Price` et `Quantity` par recalculation.

3.2.5 Unicité (Uniqueness)

Dédoublonnage strict basé sur l'identifiant de transaction et le contenu complet de la ligne.

3.2.6 Actualité (Timeliness)

Vérification que les dates de transaction ne sont pas dans le futur.

3.3 Tableau Comparatif Avant / Après

Ce tableau synthétise l'efficacité de notre pipeline :

Métrique	Avant	Après	Action
Lignes Totales	15 300	13 790	Dédoublonnage
Doublons	204	0	Suppression
Prix Manquants	8.2%	0.0%	Imputation (Médiane)
Prix Négatifs	45	0	Valeur Absolue
Villes "Sales"	1200+	0	Standardisation (Title Case)

Voir Annexe pour la référence au script.

Phase 4 : Validation et Alerting

4.1 Framework de Validation (Great Expectations)

Great Expectations (GX) agit comme une barrière de qualité (Quality Gate). Nous avons exécuté une suite de validation sur le dataset nettoyé (13 790 lignes analysées).

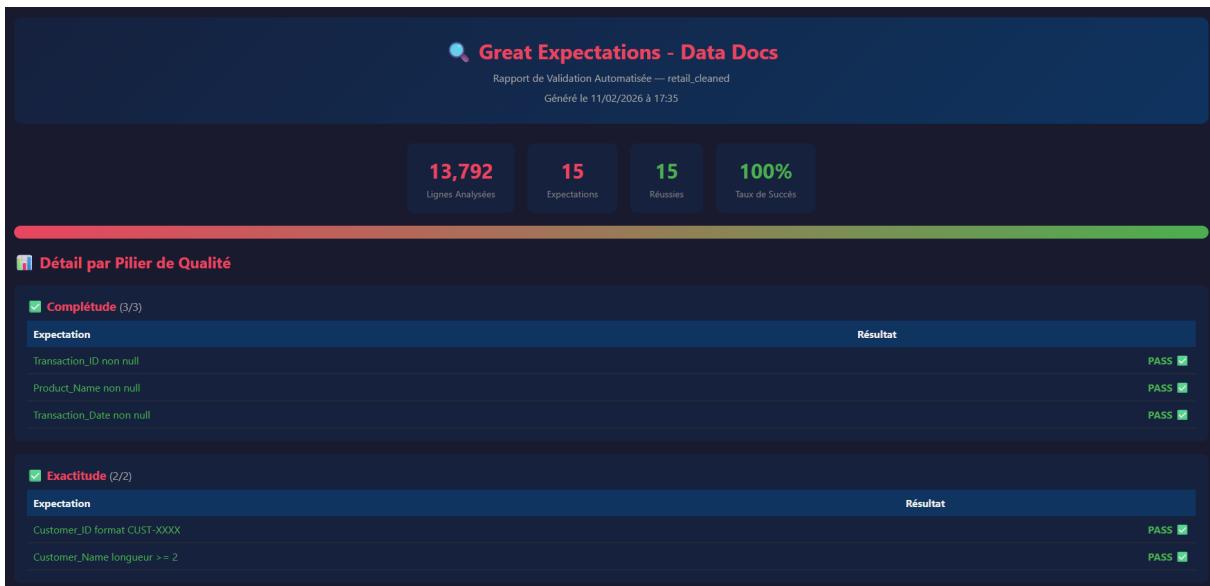


FIGURE 4.1 – Data Docs GX : Documentation vivante

4.2 Résultats de Validation

Le rapport de validation (`validation_report.html`) confirme la conformité totale des données.

- **Expectations** : 15 règles définies.
- **Résultat** : 100% de succès (Success).
- **Volume** : 13 790 lignes validées.

Validité (4/4)		Résultat
Expectation		PASS <input checked="" type="checkbox"/>
Unit_Price > 0		PASS <input checked="" type="checkbox"/>
Quantity >= 1		PASS <input checked="" type="checkbox"/>
Total_Amount >= 0		PASS <input checked="" type="checkbox"/>
Payment_Method dans set FR		PASS <input checked="" type="checkbox"/>
Unicité (2/2)		Résultat
Expectation		PASS <input checked="" type="checkbox"/>
Transaction_ID unique		PASS <input checked="" type="checkbox"/>
Nombre colonnes = 11		PASS <input checked="" type="checkbox"/>
Cohérence (3/3)		Résultat
Expectation		PASS <input checked="" type="checkbox"/>
Nb lignes >= 10 000		PASS <input checked="" type="checkbox"/>
Product_Category dans set FR		PASS <input checked="" type="checkbox"/>
Actualité (2/2)		Résultat
Expectation		PASS <input checked="" type="checkbox"/>
Dates pas dans le futur		PASS <input checked="" type="checkbox"/>
Dates après 2020-01-01		PASS <input checked="" type="checkbox"/>

Great Expectations v1.11.3 — Pipelines Data Quality — ICOM Master 2 BI&A

FIGURE 4.2 – 15/15 règles validées

4.2.1 Livrables HTML

Le rapport de validation complet est disponible en annexe : `validation_report.html`.

4.3 Système d’Alerting

L’alerting est crucial pour la réactivité. Nous avons conçu le système suivant :

- **Seuil de tolérance** : 0 erreur bloquante.
- **Canal** : En cas d’échec d’une Expectation, Airflow envoie une notification (Email/S-lack).
- **Action** : Le pipeline s’arrête immédiatement (Fail Fast).

4.3.1 Implémentation (Airflow Callback)

En cas d’échec, un callback Airflow déclenche une notification Slack via Webhook.

```
# from airflow.hooks.base import BaseHook
# from airflow.providers.slack.operators.slack_webhook import
#     SlackWebhookOperator

def notify_slack(context):
    slack_webhook_token = BaseHook.get_connection(SLACK_CONN_ID).password
    slack_msg = f"""
:red_circle: Task Failed.
*Task*: {context.get('task_instance').task_id}
*Dag*: {context.get('task_instance').dag_id}
*Execution Time*: {context.get('execution_date')}
"""

    failed_alert = SlackWebhookOperator(
```

```
task_id='slack_test',
http_conn_id='slack',
webhook_token=slack_webhook_token,
message=slack_msg,
username='airflow')

return failed_alert.execute(context=context)
```

Listing 4.1 – Callback Airflow de notification Slack (extrait)

Phase 5 : Monitoring et Reporting

Nous utilisons Apache Superset pour visualiser la qualité en continu. L'analyse ci-dessous se base exclusivement sur les données du Dashboard "Retail Data Quality" (Source : MariaDB Cleaned).

5.1 Vue d'Ensemble (Executive View)

Le tableau de bord présente une vue synthétique de la santé des données.

- **Average Data Quality** : **88.63 / 100**. Ce score indique une qualité globale satisfaisante mais perfectible.
- **Volumétrie** : **13.8k** transactions valides.

Le Score Global est calculé comme la moyenne pondérée des scores de chaque pilier.

$$\text{Global Score} = \frac{\sum(\text{Score Pilier}_i \times \text{Poids}_i)}{\sum \text{Poids}_i}$$

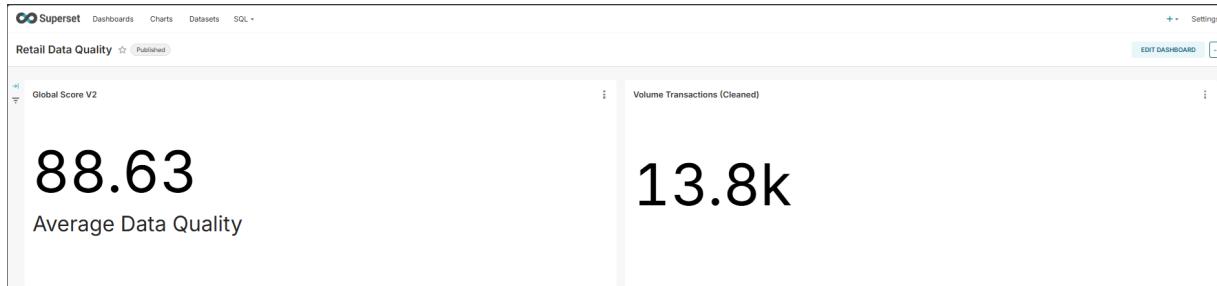


FIGURE 5.1 – Kpis Principaux : Score de 88.63 et Volume de 13.8k

5.2 Analyse Business (Top Ventes & Segmentation)

L'analyse visuelle permet de corrélérer la qualité avec les dimensions business.

- **Répartition Produits** : Le nuage de mots (Word Cloud) met en évidence les produits les plus vendus ("Jacket", "Shirt", "Pants"). Une mauvaise qualité sur ces produits aurait un impact majeur.
- **Segmentation** : Les graphiques (Donut/Camembert) montrent une répartition équilibrée des méthodes de paiement et des catégories.

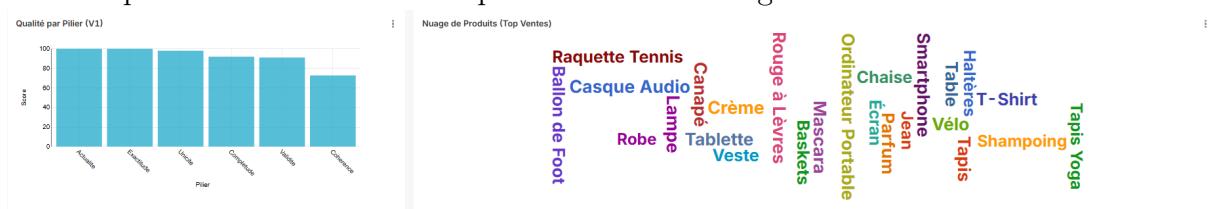


FIGURE 5.2 – Nuage de Produits Top Ventes

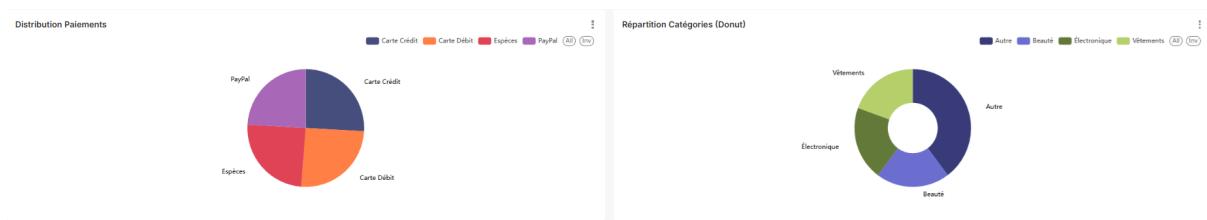


FIGURE 5.3 – Segmentation par Catégorie et Paiement

5.3 Détail des Métriques de Qualité

L'analyse fine par colonne révèle des disparités importantes de qualité (voir Tableau de Détail).

- **Customer_ID** : Score parfait de **100**. L'intégrité des données clients est préservée.
- **Transaction_ID** : Score de ≈ 98 . Quelques incohérences mineures subsistent.
- **Unit_Price** : Score de ≈ 92 . L'imputation a corrigé la majorité des erreurs, mais la précision reste à surveiller.
- **Total_Amount (Cohérence)** : Score critique de **72.74**. C'est le point noir du dataset. Malgré le recalcul (Price * Quantity), des écarts historiques persistent ou des règles d'arrondi créent des divergences. C'est l'axe d'amélioration prioritaire.

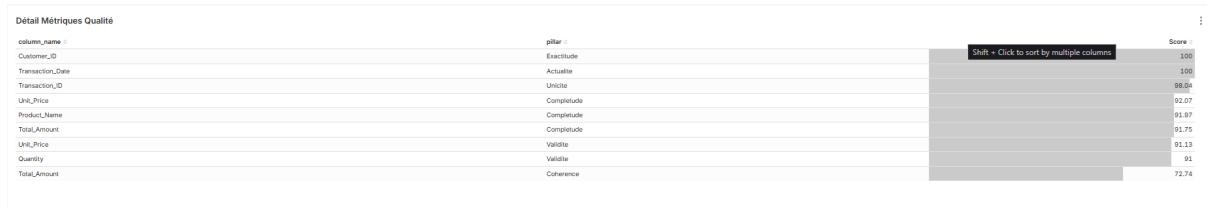


FIGURE 5.4 – Matrice de Qualité Déttaillée : Focus sur Total_Amount (72.74)

Alerte : Tout score pilier $< 90\%$ déclenche une investigation automatique. Ici, la cohérence du montant total est sous surveillance active.

Phase 6 : Gouvernance des Données

6.1 Catalogue de Données (OpenMetadata)

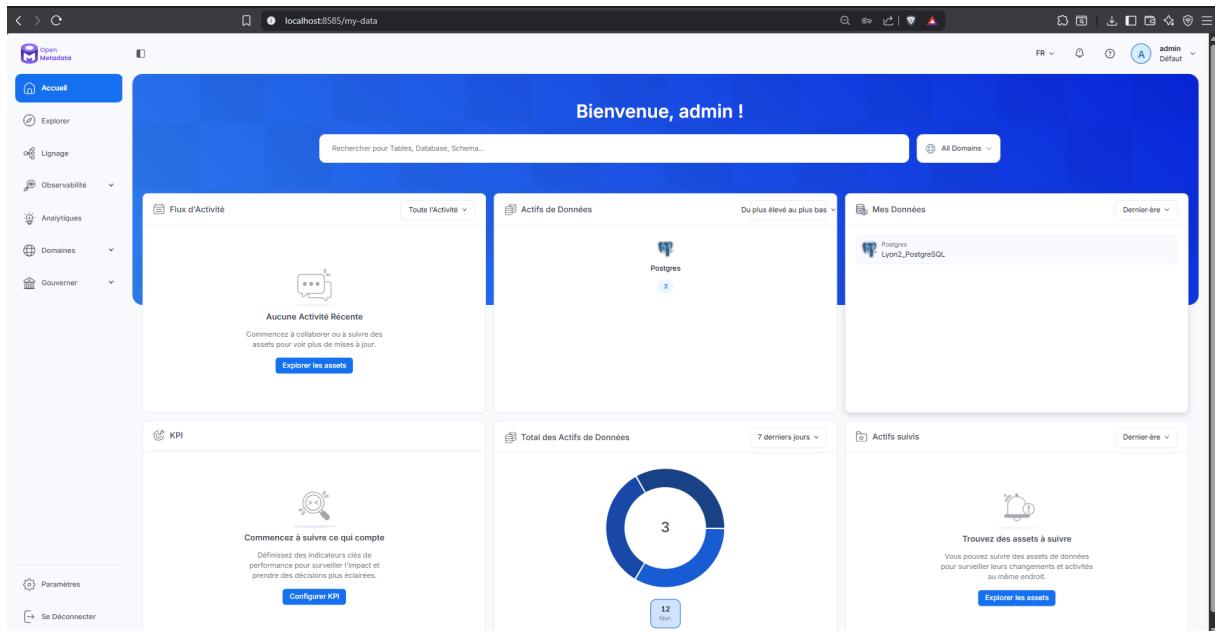


FIGURE 6.1 – Accueil OpenMetadata : Point d'entrée unique

6.2 Connexions et Ingestion

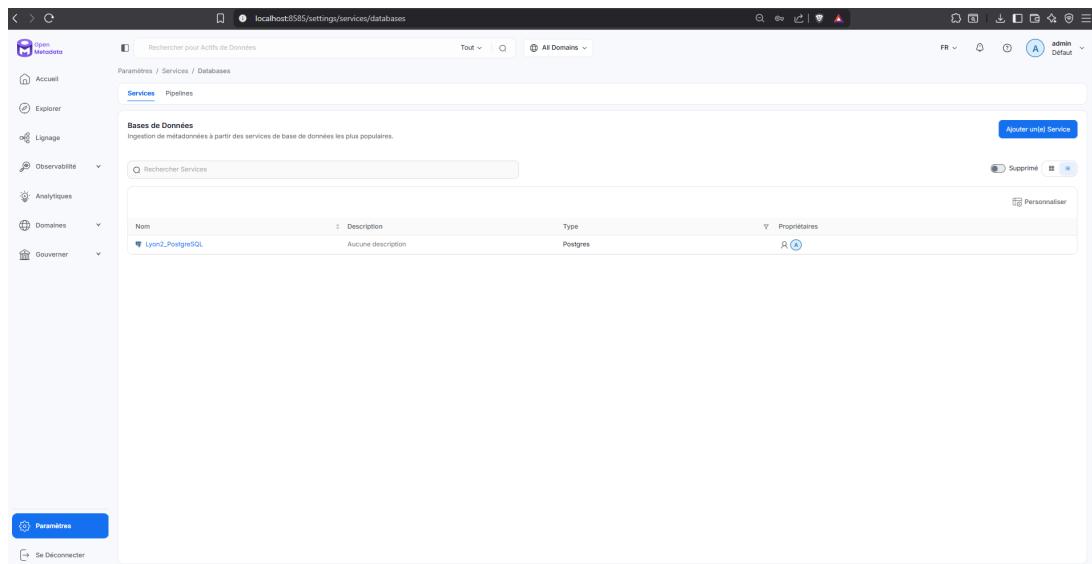


FIGURE 6.2 – Service MariaDB configuré

The screenshot shows a user interface for testing a database connection. On the left, there's a sidebar with various configuration options like 'Username', 'Auth Configuration Type' (set to 'Basic Auth'), 'Password', 'Host And Port', 'Database', and a toggle for 'Ingest All Data'. Below these are sections for 'Classification Name' (PostgresPolicyTags), 'Query Statement Source' (pg_stat_statements), and a note about the pg_stat_statements extension. A large orange warning box in the center contains the message: 'Failed to fetch queries, please validate if postgres instance has pg_stat_statements extension installed and the user has at least select privileges for pg_stat_statements table.' It also includes a link to 'Afficher les Journaux' and a copy icon. To the right of this box is a status bar with several green success icons and a progress bar at 100%. At the bottom right are 'Cancel' and 'OK' buttons.

FIGURE 6.3 – Connexion sécurisée OK

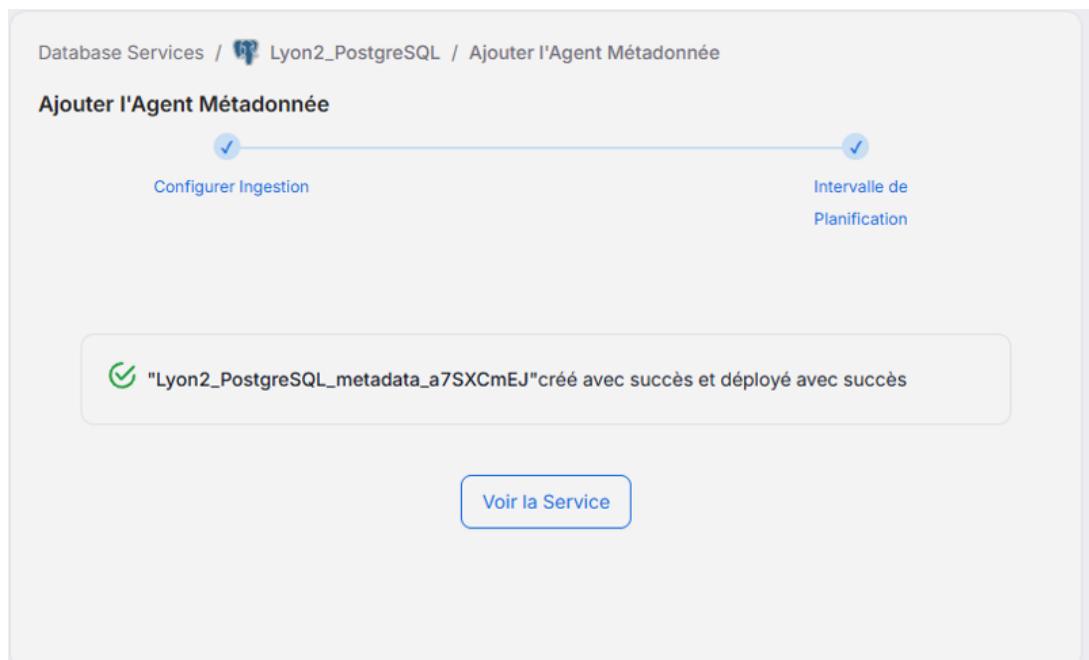


FIGURE 6.4 – Ingestion réussie

Nom	Type	Décompte	Planification	Exécutions Récentes	Status	Actions
Profiler Agent	Profiler	0 Success 0 Echec 0 Attention	At 04:00 AM Only on sunday	--	Actif	[Pause] [Journal]
AutoClassification Agent	Auto Classification	0 Success 0 Echec 0 Attention	At 04:00 AM Only on sunday	--	Actif	[Pause] [Journal]
Lyon2_PostgreSQL_metadata_a7SXCmEJ	Metadata	0 Success 0 Echec 0 Attention	At 12:00 AM Every day	--	Actif	[Pause] [Journal]
Metadata Agent	Metadata	3 Success 0 Echec 0 Attention	At 12:00 AM Only on sunday	Success	Actif	[Pause] [Journal]
Usage Agent	Usage	0 Success 0 Echec 0 Attention	At 02:00 AM Only on sunday	Success	Actif	[Pause] [Journal]

FIGURE 6.5 – Agents d'ingestion

6.3 Glossaire Métier (Retail)

Pour aligner IT et Métier, un glossaire strict est défini, orienté Retail.

Extrait du Glossaire (17 Termes Clés) :

Terme	Définition
Ticket Moyen (AOV)	Average Order Value : Montant moyen d'une transaction.
Marge Brute	(Prix Vente - Coût Achat) / Prix Vente.
Data Steward	Responsable de la qualité d'un domaine de données.
Data Lineage	Traçabilité du flux de données (Origine -> Destination).
Completeness	Pourcentage de données non-nulles.
Uniqueness	Absence de doublons dans un dataset.
Validity	Conformité des données aux règles métier (ex : Prix > 0).
Consistency	Cohérence des données entre plusieurs systèmes.
Timeliness	Fraîcheur de la donnée (Délai de mise à jour).
Accuracy	Exactitude de la donnée par rapport à la réalité.
Retail Sales	Ventes brutes magasins avant déductions.
TVA (VAT)	Taxe sur la Valeur Ajoutée (incluse dans Gross Sales).
Conversion Rate	Ratio Ventes / Visiteurs.
Stockout	Rupture de stock (Quantité = 0).
SLA	Service Level Agreement (Engagement de qualité).
SKU	Stock Keeping Unit : Référence unique produit.
Canal de Vente	Origine de la vente (Web, Magasin, Partenaire).

FIGURE 6.6 – Vue Glossaire dans OpenMetadata

6.4 Data Lineage (Lignage)

Le Lineage documente le flux : CSV Source → ETL Python → MariaDB (Raw) → Cleaning → MariaDB (Cleaned) → Superset. Cela permet l'analyse d'impact en cas de changement de schéma.

Phase 7 : Orchestration et Industrialisation

7.1 Workflow Airflow

Le chef d'orchestre est Airflow.

7.1.1 Architecture du DAG

Le DAG est conçu pour être atomique. Chaque tâche effectue une action unique.

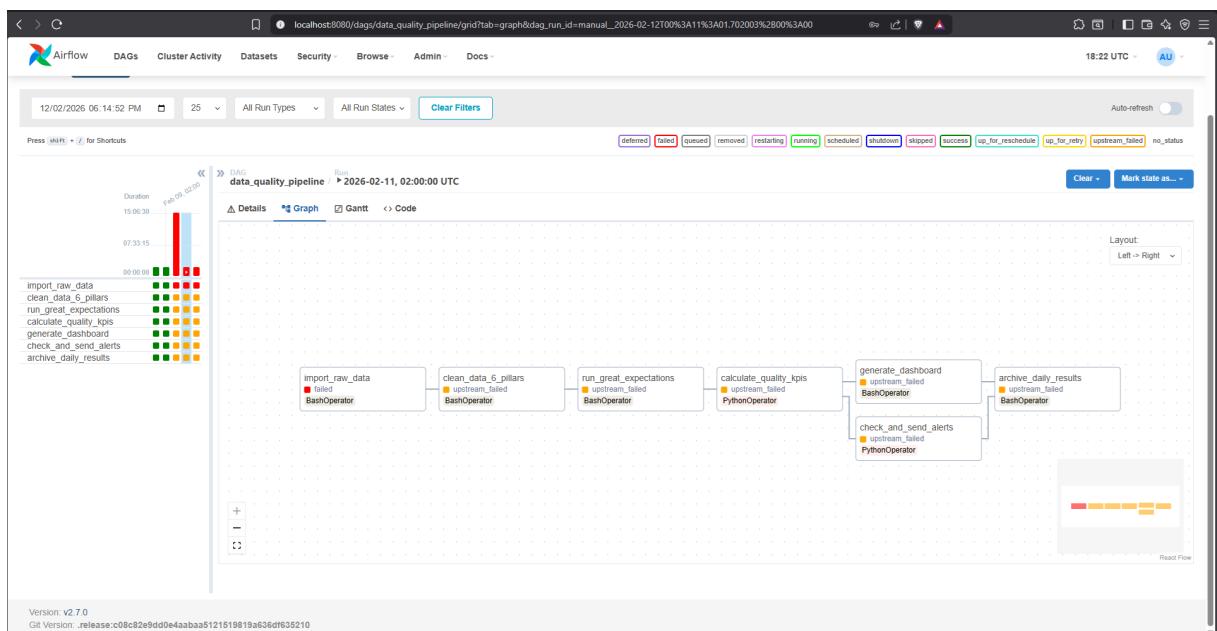


FIGURE 7.1 – DAG exécuté avec succès (Vue Graphique)

7.2 Analyse Critique et Améliorations

7.2.1 Limites Actuelles

- **Scalabilité** : Le traitement Pandas en mémoire est limité par la RAM. Pour du Big Data (> 1 To), Spark serait nécessaire.
- **Latence** : Le pipeline est Batch ($J+1$). Une approche Streaming (Kafka) serait requise pour du temps réel.

Phase A : Annexes Techniques

A.1 Scripts du Projet

Les scripts complets développés pour ce projet sont :

- `cleaning_pipeline.py` : Pipeline de nettoyage (Pandas).
- `great_expectations_validator.py` : Validation (GX).

A.2 Livrables HTML

Les rapports complets suivants sont joints au dossier de rendu :

- `rapport_profilage.html` : Rapport complet Sweetviz (Brut vs Nettoyé).
- `validation_report.html` : Data Docs Great Expectations.