ROYAUME DU MAROC

HAUT COMMISSARIAT AU PLAN

INSTITUT NATIONAL DE STATISTIQUE ET D'ECONOMIE APPLIQUEE

INSEA

RAPPORT DE PROJET DATA WAREHOUSE

Analyse des Avis Clients des Agences Bancaires Marocaines

Utilisation d'un Stack de Données Moderne

Réalisé par :
LAOUAD Ayoub

Encadré par :
Mr. BENELALLAM Imade

Master Systèmes d'Information et Systèmes Intelligents

M2SI M1

2024-2025

Table des matières

Table des figures

Liste des tableaux

1 Introduction

Le secteur bancaire marocain, en pleine transformation digitale, fait face à un défi majeur : l'exploitation efficace des retours clients disponibles sur les plateformes numériques. Avec l'essor des services bancaires en ligne et l'augmentation des interactions clients sur Google Maps, les banques disposent d'une mine d'informations précieuses sous forme d'avis et de commentaires clients.

Ces données, bien qu'abondantes, restent largement sous-exploitées en raison de leur nature non structurée et de leur dispersion géographique. Face à cette problématique, le présent projet propose une approche moderne et automatisée pour centraliser, nettoyer, analyser et valoriser ces avis clients dans le contexte spécifique du secteur bancaire marocain.

1.1 Contexte du Projet

Les agences bancaires marocaines reçoivent quotidiennement des milliers d'avis sur Google Maps, reflétant l'expérience client réelle concernant la qualité de service, les temps d'attente, l'accueil, et d'autres aspects critiques de la relation bancaire. Cette information, cruciale pour l'amélioration continue des services, demeure dispersée et difficile à analyser de manière systématique.

1.2 Problématique

Comment peut-on transformer ces données non structurées en insights stratégiques exploitables pour améliorer la satisfaction client et optimiser les performances des agences bancaires ?

1.3 Objectifs du Projet

L'objectif principal de ce projet est de développer un pipeline de données moderne et automatisé permettant de :

- Collecter automatiquement les avis Google Maps des principales agences bancaires marocaines
- Nettoyer et enrichir ces données par des techniques d'analyse de sentiment et de modélisation de sujets
- Modéliser les données selon une architecture de Data Warehouse optimisée
- Fournir des insights visuels interactifs via des tableaux de bord décisionnels
- Automatiser l'ensemble du processus pour assurer une mise à jour continue

2 Revue de Littérature et État de l'Art

2.1 Data Warehousing et Business Intelligence

Le concept de Data Warehouse, introduit par Inmon dans les années 1990, consiste en un référentiel central de données intégrées, orientées sujet, non volatiles et historisées. L'approche moderne du Data Warehousing s'appuie sur des architectures modulaires et des outils spécialisés pour chaque étape du cycle de vie de la donnée.

2.2 Analyse de Sentiment et Traitement du Langage Naturel

L'analyse de sentiment, ou opinion mining, constitue un domaine émergent du traitement automatique du langage naturel (NLP). Les approches modernes utilisent des modèles pré-entraînés comme BERT (Bidirectional Encoder Representations from Transformers) pour classifier automatiquement la polarité des textes.

2.3 Modern Data Stack

Le concept de "Modern Data Stack" désigne l'utilisation d'outils cloud-native, modulaires et interopérables pour construire des pipelines de données robustes. Cette approche privilégie la séparation des préoccupations et l'automatisation.

3 Méthodologie

3.1 Approche Générale

Notre méthodologie s'articule autour d'une approche itérative en cinq phases distinctes, chacune correspondant à une étape clé du cycle de vie de la donnée :

1. **Collecte des données** : Extraction automatisée des avis Google Maps
2. **Transformation et enrichissement** : Nettoyage et analyse NLP
3. **Modélisation** : Création d'un schéma en étoile
4. **Analyse et visualisation** : Développement de tableaux de bord
5. **Automatisation** : Orchestration avec Apache Airflow

3.2 Architecture Technique

Le projet s'appuie sur un stack technologique moderne comprenant :

TABLE 1 – Stack technologique du projet

Étape	**Technologies**
Collecte	Python, Selenium, Google Maps
Orchestration	Apache Airflow
Stockage	PostgreSQL
Transformation	DBT (Data Build Tool), SQL
Analyse NLP	Python, Transformers, NLTK
Visualisation	Looker Studio
Versioning	Git, GitHub

4 Développement et Implémentation

4.1 Phase 1 : Collecte des Données

4.1.1 Architecture de Collecte

La collecte des données s'effectue via un scraping automatisé utilisant Selenium WebDriver pour naviguer sur Google Maps et extraire les avis des principales banques marocaines dans les grandes villes du royaume.

4.1.2 Implémentation du Scraping

Le script `Scraping.py` implémente une approche robuste de collecte avec gestion des erreurs et respect des bonnes pratiques :

```
def scrape_bank_reviews(bank_name, city):
    """
    Collecte les avis Google Maps pour une banque dans une ville donn e
    """
    driver = webdriver.Chrome()
    reviews_data = []

    try:
        # Navigation vers Google Maps
        search_query = f"{bank_name} {city} Morocco"
        driver.get(f"https://maps.google.com/search/{search_query}")

        # Extraction des avis
        # ... logique de scraping

    except Exception as e:
        logging.error(f"Erreur lors du scraping: {e}")
    finally:
        driver.quit()

    return reviews_data
```

Listing 1 – Extrait du script de scraping

4.1.3 Orchestration avec Airflow

Le DAG `Banque_Collecting_Dag.py` automatise la collecte selon une planification définie :

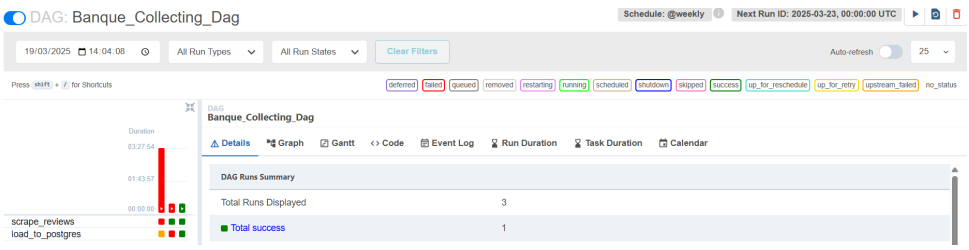


FIGURE 1 – Vue du DAG de collecte dans Airflow

4.1.4 Résultats de la Collecte

La phase de collecte a permis d'extraire un volume significatif de données :

	id [PK] inte	bank_name text	branch_name text	location text	city text	review_text text	rating text	review_date text
1	1	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC83+JV6, Agadir 80000	CC83+JV6, Agadir 80000	AGADIR	Le pure service bancaire qu'il m'ait été donné de voir!!!!! ...	1 étoile	il y a 2 semaines
2	2	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC83+JV6, Agadir 80000	CC83+JV6, Agadir 80000	AGADIR	Il est plus facile d'ouvrir un compte que de le clôturer. T...	1 étoile	il y a un an
3	3	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC83+JV6, Agadir 80000	CC83+JV6, Agadir 80000	AGADIR	Service médiocre attendre pls qu 40 min !!!!!!!	1 étoile	il y a 8 mois
4	4	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC83+JV6, Agadir 80000	CC83+JV6, Agadir 80000	AGADIR	Vraiment les agents no professionel ainsi le service de ...	1 étoile	il y a 10 mois
5	5	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC83+JV6, Agadir 80000	CC83+JV6, Agadir 80000	AGADIR	Excellente expérience	5 étoiles	il y a un an
6	6	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC6R+328, Av. Moulay H...	CC6R+328, Av. Moulay Hass...	AGADIR	La pire agence de tout le royaume,	1 étoile	il y a 10 mois
7	7	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC6R+328, Av. Moulay H...	CC6R+328, Av. Moulay Hass...	AGADIR	Service déplorable, à chaque fois que je viens dans cett...	1 étoile	il y a 2 ans
8	8	ATTIJARIWAFA BANK	ATTIJARIWAFA BANK CC6R+328, Av. Moulay H...	CC6R+328, Av. Moulay Hass...	AGADIR	Très bon service par la directrice qui m'a aidé pour plus...	5 étoiles	il y a 2 ans

Nombre total de lignes : 9523 Requête terminée 00:00:00.277 CRLF Lgn 1, Col 15

FIGURE 2 – Extrait des données collectées

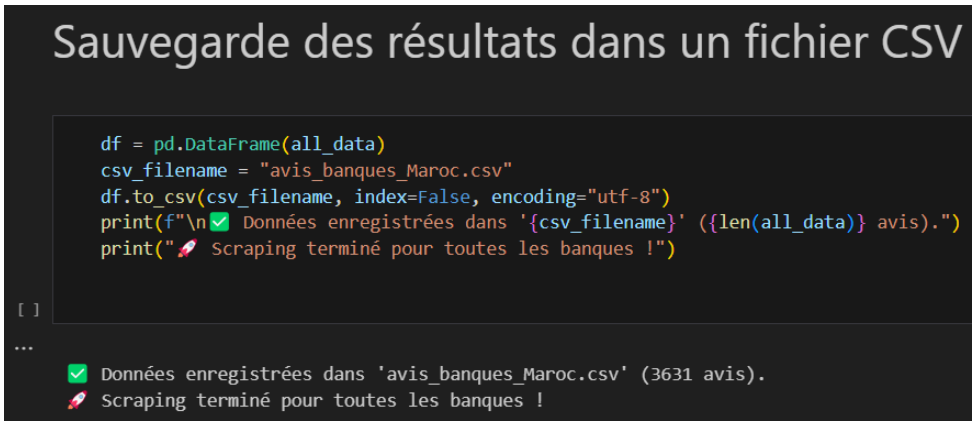


FIGURE 3 – Interface de scraping en cours d'exécution

Les données collectées comprennent :
— Nom de la banque et de l'agence
— Localisation géographique
— Texte de l'avis client
— Note attribuée (1-5 étoiles)
— Date de publication

4.2 Phase 2 : Nettoyage et Transformation

4.2.1 Pipeline de Nettoyage

Cette phase critique transforme les données brutes en information exploitable via plusieurs étapes :

1. **Dédoublonnage** : Suppression des avis dupliqués
2. **Normalisation** : Uniformisation du format des données
3. **Détection de langue** : Identification automatique de la langue
4. **Analyse de sentiment** : Classification positive/négative/neutre
5. **Extraction de topics** : Identification des thèmes récurrents

4.2.2 Analyse de Sentiment avec BERT

L'analyse de sentiment utilise un modèle BERT multilingue pré-entraîné :

```
from transformers import pipeline

def analyze_sentiment(text):
```

```
    """
    Analyse le sentiment d'un texte avec BERT multilingue
    """
    classifier = pipeline(
        "sentiment-analysis",
        model="nlptown/bert-base-multilingual-uncased-sentiment"
    )

    result = classifier(text)
    return {
        'sentiment': result[0]['label'],
        'confidence': result[0]['score']
    }
```

Listing 2 – Implémentation de l'analyse de sentiment

4.2.3 Orchestration du Nettoyage

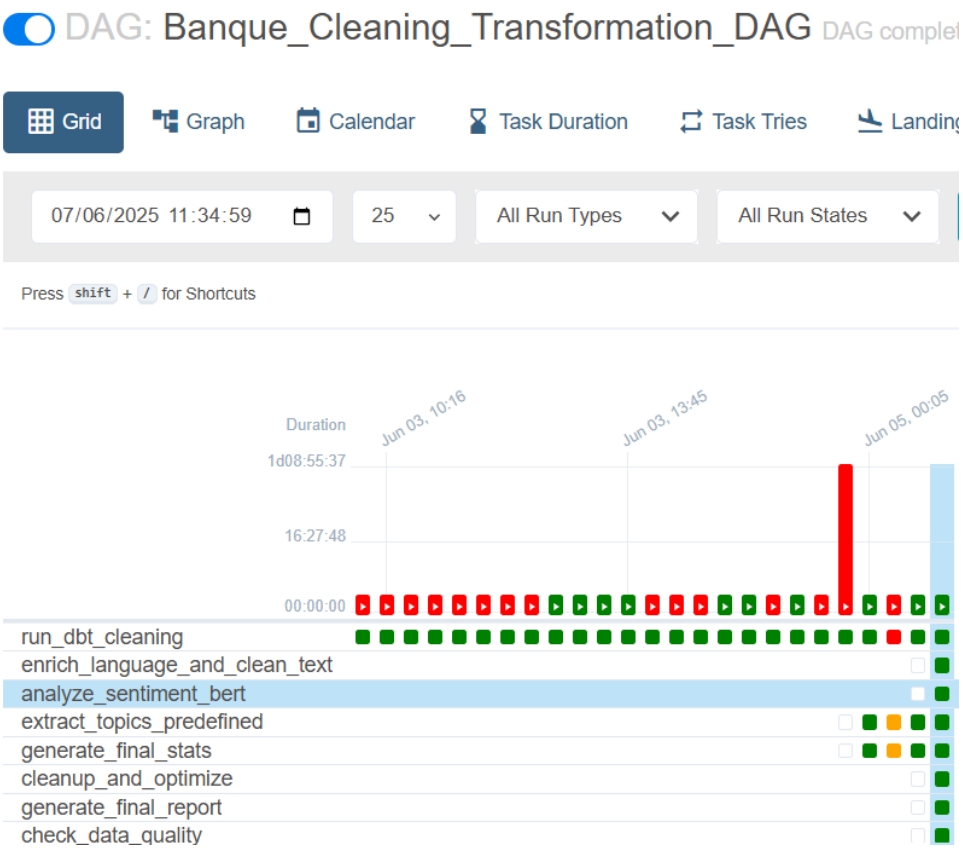
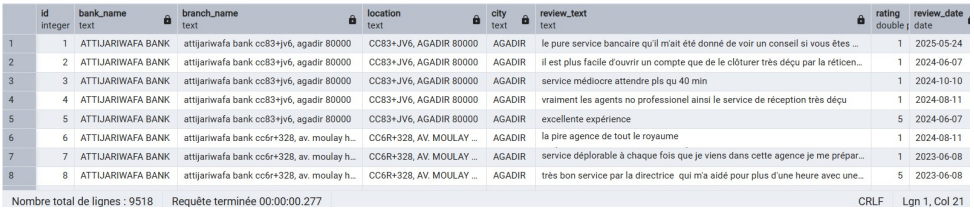


FIGURE 4 – DAG de nettoyage et transformation

4.2.4 Intégration DBT

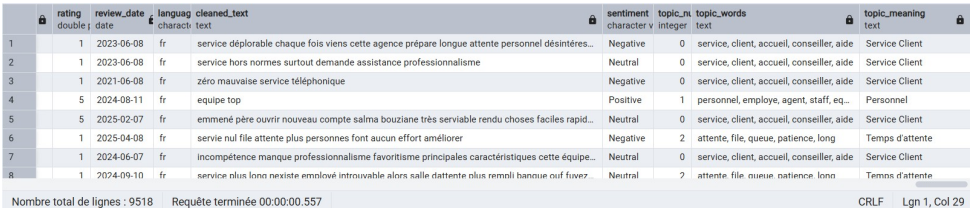
DBT (Data Build Tool) orchestre les transformations SQL complexes :



	id integer	bank_name text	branch_name text	location text	city text	review_text text	rating double	review_date date
1	1	ATTIJARIWAFA BANK	attijariwafa bank cc83+jv6, agadir 80000	CC83+JV6, AGADIR 80000	AGADIR	le pure service bancaire qu'il m'ait été donné de voir un conseil si vous êtes ...	1	2025-05-24
2	2	ATTIJARIWAFA BANK	attijariwafa bank cc83+jv6, agadir 80000	CC83+JV6, AGADIR 80000	AGADIR	il est plus facile d'ouvrir un compte que de le clôturer très déçu par la réticen...	1	2024-06-07
3	3	ATTIJARIWAFA BANK	attijariwafa bank cc83+jv6, agadir 80000	CC83+JV6, AGADIR 80000	AGADIR	service médiocre attendre pls qu 40 min	1	2024-10-10
4	4	ATTIJARIWAFA BANK	attijariwafa bank cc83+jv6, agadir 80000	CC83+JV6, AGADIR 80000	AGADIR	vraiment les agents no professionel ainsi le service de réception très déçu	1	2024-08-11
5	5	ATTIJARIWAFA BANK	attijariwafa bank cc83+jv6, agadir 80000	CC83+JV6, AGADIR 80000	AGADIR	excellente expérience	5	2024-06-07
6	6	ATTIJARIWAFA BANK	attijariwafa bank cc6r+328, av. moulay h...	CC6R+328, AV. MOULAY ...	AGADIR	la pire agence de tout le royaume	1	2024-08-11
7	7	ATTIJARIWAFA BANK	attijariwafa bank cc6r+328, av. moulay h...	CC6R+328, AV. MOULAY ...	AGADIR	service déplorable à chaque fois que je viens dans cette agence je me prépar...	1	2023-06-08
8	8	ATTIJARIWAFA BANK	attijariwafa bank cc6r+328, av. moulay h...	CC6R+328, AV. MOULAY ...	AGADIR	très bon service par la directrice qui m'a aidé pour plus d'une heure avec une...	5	2023-06-08

Nombre total de lignes : 9518 Requête terminée 00:00:00.277 CRLF Lgn 1, Col 21

FIGURE 5 – Résultats des transformations DBT



	rating double	review_date date	languag charact	cleaned_text text	sentiment character v	topic_nu integer	topic_words text	topic_meaning text
1	1	2023-06-08	fr	service déplorable chaque fois viens cette agence prépare longue attente personnel désintéres...	Negative	0	service, client, accueil, conseiller, aide	Service Client
2	1	2023-06-08	fr	service hors normes surtout demande assistance professionnalisme	Neutral	0	service, client, accueil, conseiller, aide	Service Client
3	1	2021-06-08	fr	zéro mauvaise service téléphonique	Negative	0	service, client, accueil, conseiller, aide	Service Client
4	5	2024-08-11	fr	equipe top	Positive	1	personnel, employe, agent, staff, eq...	Personnel
5	5	2025-02-07	fr	emmené père ouvrir nouveau compte salma bouziane très serviable rendu choses faciles rapid...	Neutral	0	service, client, accueil, conseiller, aide	Service Client
6	1	2025-04-08	fr	servie nul file attente plus personnes font aucun effort améliorer	Negative	2	attente, file, queue, patience, long	Temps d'attente
7	1	2024-06-07	fr	incompétence manque professionnalisme favoritisme principales caractéristiques cette équipe...	Neutral	0	service, client, accueil, conseiller, aide	Service Client
8	1	2024-09-10	fr	service plus long pexiste employé introuvable alors salle dattente plus rempli banque ouf fuyez...	Neutral	2	attente, file, queue, patience, long	Temps d'attente

Nombre total de lignes : 9518 Requête terminée 00:00:00.557 CRLF Lgn 1, Col 29

FIGURE 6 – Données après transformation

4.3 Phase 3 : Modélisation des Données

4.3.1 Conception du Schéma en Étoile

La modélisation suit une approche dimensionnelle classique avec un schéma en étoile optimisé pour l'analyse :

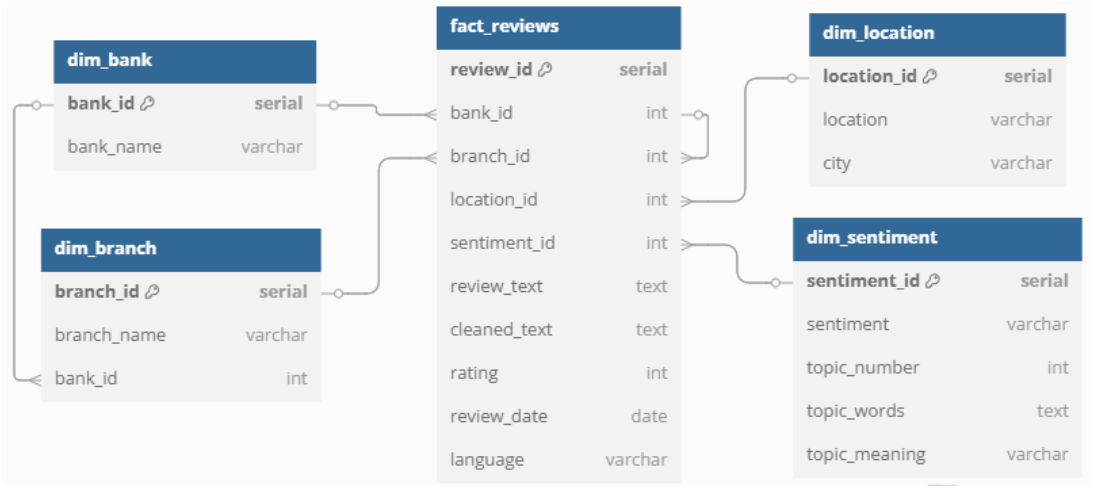


FIGURE 7 – Schéma en étoile du Data Warehouse

4.3.2 Table de Faits

La table `fact_reviews` centralise les métriques quantifiables :

	review_id text	bank_id text	branch_id text	location_id text	sentiment_id text	rating double	review_date date
1	8f14e45fceea...	7b34ab946...	20dfdacc70...	e023a44eb...	356730ce000...	1	2023-06-08
2	45c48cce2e2d...	7b34ab946...	20dfdacc70...	e023a44eb...	3e6afc5190ad...	1	2023-06-08
3	d3d9446802a...	7b34ab946...	20dfdacc70...	e023a44eb...	356730ce000...	1	2021-06-08
4	6512bd43d9c...	7b34ab946...	20dfdacc70...	e023a44eb...	b76482039b4...	5	2024-08-11
5	c51ce410c124...	7b34ab946...	20dfdacc70...	e023a44eb...	3e6afc5190ad...	5	2025-02-07

Nombre total de lignes : 9518 Requête terminée 00:00:00.309

FIGURE 8 – Structure de la table de faits

4.3.3 Tables de Dimensions

	bank_name text	bank_id text
1	ATTIJARIWAFA BANK	7b34ab9465407973f34773c9b5e1780a
2	UMNIA BANKARAB BANK	8de993231c09d30f9619792f093b8f2f
3	CIH BANK	7e6a8e178bbfb3c6431c27f80df6fd05
4	BANQUE POPULAIRE	901f1593e6fe321fa5552438c5e3c57b
5	AL BARID BANK	58863e522be33feb07d339e2166f2f8e

Nombre total de lignes : 23 Requête terminée 00:00:00.193

FIGURE 9 – Table de dimension des banques

`dim_bank` - Dimension Banque

	branch_name text	branch_id text
1	crédit du maroc 06 rue dayat aoua, rabat	dd0c331a52f4dca9c2e15076a6910957
2	bmci bank 22r3+v6w, fès 30050	cdcb1e1cf8d992159097778ff21a85b0
3	banque centrale populaire x9wm+g3q, av. des far, settat	70f3021ec8e53185d8305bd397e16cf7
4	banque populaire 62j9+vp6, taza	79fc416b802a4f6ecc556ca01950b08f
5	credit du maroc rue lalla fatima zahra, fes 30050	027cc901a8529120235ef2138292ffa7

Nombre total de lignes : 1282 Requête terminée 00:00:00.248

FIGURE 10 – Table de dimension des agences

`dim_branch` - Dimension Agence

	location text	city text	location_id text
1	J2FF+VFV, MARRAKESH 40000	MARRAKECH	8686bb1b2dc22275f1beec231b2b25e1
2	W35H+2CH, TÉMARA 12000	TÉMARA	825e382200a0b68cc3a8bf77f6c4420d
3	AV. MOULAY HASSAN, OUJDA 60000	OUJDA	d24eb1f6583efdf66db4f4ea3560af32
4	LOTISSEMENT TOUABEL, LOT 239 AV. DES FAR, TETOUAN 93000	TÉTOUAN	1a66558ea39da34853b5ecfe0abd4b34
5	AVENUE LA RÉSISTANCE, BP 9246, RABAT 10040	RABAT	b24477b5c6d909640d69d968b306c5...

Nombre total de lignes : 1292 Requête terminée 00:00:00.133

FIGURE 11 – Table de dimension géographique

`dim_location` - Dimension Géographique

	sentiment character var	topic_number integer	topic_words text	topic_meaning text	sentiment_id text
1	Neutral	2	attente, file, queue, patience, long	Temps d'attente	28b63190604207b684a01031dc5b05...
2	Negative	3	guichet, caisse, comptoir, depot, retr...	Guichet/Caisse	26fc1f54d8d2f5dd64a3789c25131ce8
3	Positive	8	horaire, ouvert, ferme, weekend, dim...	Horaires/Accès	23bb3eedf5dcce4359547f1078c365de
4	Negative	4	telephone, appel, tel, ligne, contact	Téléphone/Digital	f87ef33bb7d743d5ba72460b349e14...
5	Negative	-1	avis general	General	aea71864dc7002a84276865ea3209e...

Nombre total de lignes : 34 Requête terminée 00:00:00.107

FIGURE 12 – Table de dimension du sentiment

`dim_sentiment` **- Dimension Sentiment**

4.3.4 Base de Données PostgreSQL

```
avis_banques=> \dt
        List of relations
 Schema |  Name   | Type  | Owner
--------+---------+-------+-------
 public | reviews | table | bank
(1 row)

avis_banques=> select count(*) from reviews;
 count
-------
  3631
(1 row)
```

FIGURE 13 – Vue de la base de données PostgreSQL

4.4 Phase 4 : Analyse et Visualisation

4.4.1 Développement des Tableaux de Bord

Les tableaux de bord Looker Studio offrent une vue d'ensemble interactive des insights extraits :

- **Analyse temporelle** : Évolution du sentiment client dans le temps
- **Comparaison inter-banques** : Classement des institutions par satisfaction
- **Analyse géographique** : Performance par région et ville
- **Topics Analysis** : Thèmes les plus fréquents dans les avis
- **KPIs de satisfaction** : Métriques de performance synthétiques

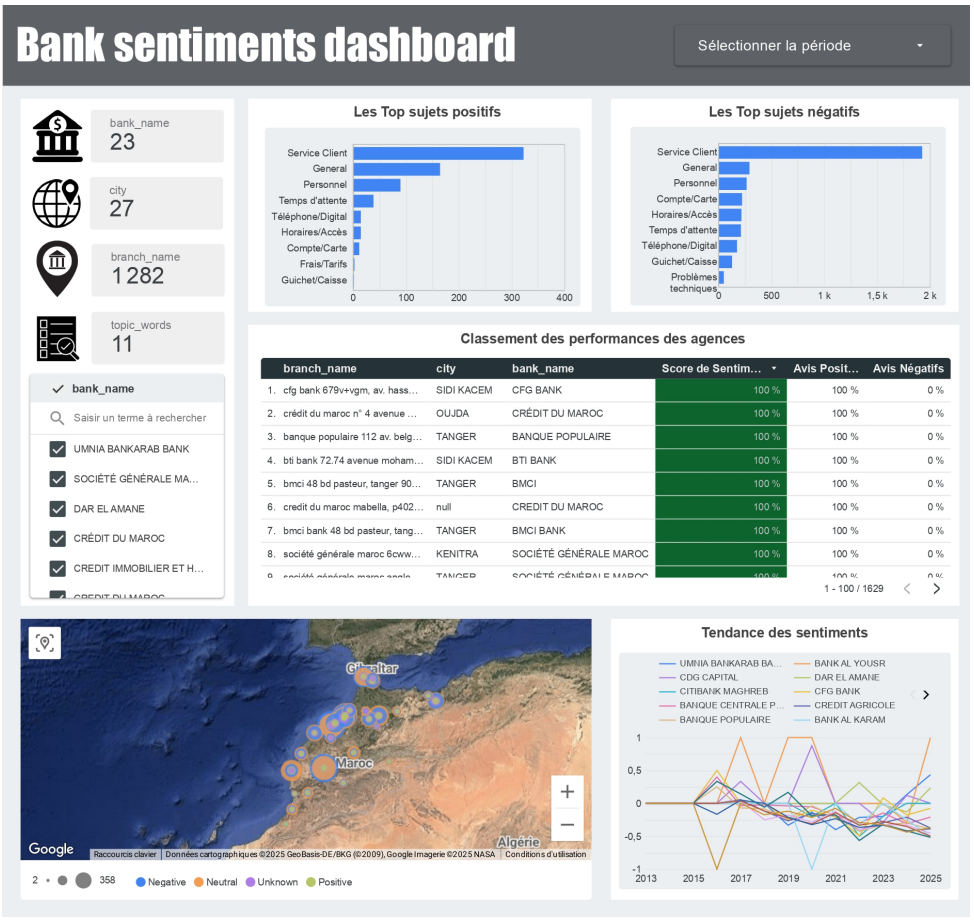


FIGURE 14 – BI Dashboard en Looker Studio

4.5 Phase 5 : Automatisation Complète

L'orchestration globale du pipeline s'effectue via Apache Airflow avec une planification adaptée aux besoins métier :

- **Collecte** : Quotidienne pour les nouvelles données
- **Transformation** : Déclenchée après chaque collecte
- **Modélisation** : Mise à jour du Data Warehouse
- **Alertes** : Notifications en cas d'échec ou d'anomalie

5 Résultats et Analyse

5.1 Volume de Données Traitées

Le pipeline a permis de traiter un volume significatif de données :

TABLE 2 – Statistiques des données collectées

Métrique	Valeur
Nombre total d'avis collectés	9523
Nombre de banques analysées	23
Nombre d'agences couvertes	1282
Villes analysées	27
Période couverte	2015-2025

5.2 Distribution des Sentiments

L'analyse de sentiment révèle une répartition intéressante :

TABLE 3 – Répartition des sentiments

Sentiment	Nombre	Pourcentage
Positif	667	7,0%
Neutre	4 284	45,0%
Négatif	3 557	37,4%
Unknown	1 010	10,6%

5.3 Insights Métier Extraits

5.3.1 Thématiques Récurrentes

L'analyse des topics révèle les préoccupations principales des clients :
— **Temps d'attente** : Thème le plus fréquent dans les avis négatifs
— **Qualité d'accueil** : Factor différenciant entre agences
— **Digitalisation** : Appréciation des services en ligne
— **Accessibilité** : Importance de la localisation géographique

5.3.2 Performance par Banque

Le classement des banques selon l'indice de satisfaction composite révèle des disparités significatives, permettant d'identifier les meilleures pratiques.

5.3.3 Analyse Géographique

Les résultats montrent des variations régionales importantes, liées aux spécificités locales et à la densité du réseau d'agences.

6 Discussion

6.1 Apports du Projet

Ce projet démontre la faisabilité et la valeur ajoutée d'une approche moderne du Data Warehousing appliquée à l'analyse de sentiment dans le secteur bancaire marocain. Les principales contributions sont :
— **Automatisation complète** : Pipeline end-to-end sans intervention manuelle

— **Scalabilité** : Architecture modulaire extensible
— **Insights actionnables** : Métriques directement exploitables par le métier
— **Méthodologie reproductible** : Approche généralisable à d'autres secteurs

6.2 Défis Rencontrés

6.2.1 Défis Techniques

— **Robustesse du scraping** : Gestion des changements d'interface Google Maps
— **Performance NLP** : Optimisation des traitements sur de gros volumes
— **Qualité des données** : Gestion de la variabilité linguistique

6.2.2 Défis Méthodologiques

— **Modélisation du sentiment** : Adaptation aux spécificités culturelles
— **Validation des résultats** : Absence de ground truth
— **Biais de collecte** : Représentativité géographique et démographique

6.3 Limitations

— **Couverture temporelle** : Données limitées aux avis récents
— **Biais de sélection** : Utilisateurs actifs sur Google Maps
— **Contexte linguistique** : Complexité du multilinguisme marocain
— **Évolution des interfaces** : Maintenance requise pour le scraping

7 Perspectives et Améliorations

7.1 Améliorations Techniques

7.1.1 Court Terme

— **Tests unitaires** : Amélioration de la robustesse du code
— **Monitoring avancé** : Alertes proactives et métriques de qualité
— **Optimisation des performances** : Parallélisation des traitements
— **Sécurisation** : Chiffrement des données sensibles

7.1.2 Moyen Terme

— **Machine Learning avancé** : Modèles personnalisés pour le contexte marocain
— **Analyse prédictive** : Anticipation des tendances de satisfaction
— **Détection d'anomalies** : Identification automatique des problèmes
— **API REST** : Exposition des données via interface programmatique

7.2 Extensions Métier

— **Élargissement sectoriel** : Application à d'autres industries
— **Sources multiples** : Intégration Facebook, Twitter, avis spécialisés
— **Analyse concurrentielle** : Benchmarking automatisé
— **Recommandations automatiques** : Suggestions d'amélioration

8 Conclusion

Ce projet de Data Warehouse dédié à l'analyse des avis clients bancaires illustre parfaitement l'application pratique des concepts de Business Intelligence moderne dans un contexte métier spécifique. L'approche développée démontre comment les technologies émergentes peuvent transformer des données non structurées en insights stratégiques exploitables.

8.1 Contributions Principales

Le projet apporte plusieurs contributions significatives :

1. **Méthodologique** : Démonstration d'une approche moderne du Data Warehousing
2. **Technique** : Implémentation d'un pipeline de données robuste et scalable
3. **Métier** : Génération d'insights actionnables pour le secteur bancaire
4. **Académique** : Documentation complète d'un cas d'usage réel

8.2 Impact et Applicabilité

Les résultats obtenus confirment la pertinence de l'approche pour :
— Les institutions bancaires souhaitant améliorer leur satisfaction client
— Les régulateurs cherchant à monitorer la qualité de service
— Les chercheurs s'intéressant à l'analyse de sentiment dans le contexte marocain
— Les praticiens du Data Engineering recherchant des patterns reproductibles

8.3 Leçons Apprises

Cette expérience souligne l'importance de :
— La robustesse et la maintenance des pipelines de données
— L'adaptation des modèles NLP aux spécificités culturelles locales
— L'automatisation comme facteur clé de succès des projets Data
— La collaboration entre expertise technique et connaissance métier

8.4 Vision Future

Ce projet constitue une base solide pour le développement d'une plateforme d'intelligence cliente plus large, intégrant l'analyse multi-sources et l'aide à la décision automatisée. L'approche modulaire adoptée facilite l'évolution vers des architectures cloud-native et l'intégration de nouvelles sources de données.

En définitive, ce travail illustre comment l'alliance entre technologies modernes et rigueur méthodologique peut transformer l'exploitation des données clients en avantage concurrentiel durable pour les institutions financières marocaines.

Bibliographie

1. Inmon, W.H. (2005). *Building the Data Warehouse*. 4th Edition, Wiley.
2. Kimball, R., Ross, M. (2013). *The Data Warehouse Toolkit : The Definitive Guide to Dimensional Modeling*. 3rd Edition, Wiley.
3. Devlin, B., Cote, L. (1996). Data warehouse : from architecture to implementation. *IBM Systems Journal*, 35(3-4), 510-551.
4. Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.
5. Vaswani, A., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
6. Devlin, J., et al. (2018). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv :1810.04805*.
7. Apache Airflow Documentation. (2024). `https://airflow.apache.org/docs/`
8. DBT Documentation. (2024). `https://docs.getdbt.com/`
9. PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*. `https://www.postgresql.org/docs/`
10. Google Cloud. (2024). *Looker Studio Documentation*. `https://support.google.com/looker-studio`

Annexes

Annexe A : Structure Détaillée du Code

```
DW_Project-LAOUAD_Ayoub/
          Banque_General_Dag.py           # DAG principal orchestrateur
          LookerStudio-Dashboard.pdf      # Export du dashboard
          Projet DW INSEA.pdf            # Cahier des charges
          README.md                       # Documentation utilisateur
          requirements.txt                # D pendances Python

          Phase1-DataCollection/
                Banque_Collecting_Dag.py    # DAG de collecte
                Scraping.py                 # Script de scraping
                config.py                   # Configuration
                Resultat/
                    Dag_collection.png
                    extrait_donn es_DBT.jpg
                    extrait_donn es_Transformation.jpg

          Phase3-DataModeling/
                Banque_Modeling_Dag.py      # DAG de mod lisation
                schema_creation.sql         # Scripts SQL
                DBT_reviews_DW/
                      dbt_project.yml
                      models/
                            staging/
                            intermediate/
                            marts/
                                dim_bank.sql
                                dim_branch.sql
                                dim_location.sql
                                dim_sentiment.sql
                                fact_reviews.sql
                      tests/
                Resultat/
                    dim_bank.png
                    dim_branch.png
                    dim_location.png
                    dim_sentiment.png
                    fact_reviews.png
                    Star_Schema.png
                    psql.png

          Phase4-Analytics/
                dashboard_config.json       # Configuration Looker
                sql_queries/                # Requ tes d'analyse
                    kpi_satisfaction.sql
                    trend_analysis.sql
                    comparative_analysis.sql

          docs/
              technical_documentation.md
              user_guide.md
              deployment_guide.md
```

Listing 3 – Arborescence complète du projet

Annexe B : Configuration Technique

B.1 Configuration Apache Airflow

```
# airflow.cfg - Extrait de configuration
[core]
dags_folder = /opt/airflow/dags
base_log_folder = /opt/airflow/logs
remote_logging = False
executor = LocalExecutor

[database]
sql_alchemy_conn = postgresql://airflow:password@localhost/airflow

# Configuration des connexions via Airflow UI
BANQUES_DB_CONN = {
    "conn_type": "postgres",
    "host": "localhost",
    "schema": "bank_reviews",
    "login": "postgres",
    "password": "password",
    "port": 5432
}
```

Listing 4 – Configuration des connexions Airflow

B.2 Configuration DBT

```
name: 'bank_reviews_dw'
version: '1.0.0'
config-version: 2

profile: 'bank_reviews'

model-paths: ["models"]
analysis-paths: ["analysis"]
test-paths: ["tests"]
seed-paths: ["data"]
macro-paths: ["macros"]
snapshot-paths: ["snapshots"]

target-path: "target"
clean-targets:
  - "target"
  - "dbt_packages"

models:
  bank_reviews_dw:
    staging:
      +materialized: view
    intermediate:
      +materialized: ephemeral
    marts:
      +materialized: table
      +post-hook: "GRANT SELECT ON {{ this }} TO analytics_role"

vars:
```

```
  sentiment_threshold: 0.7
  min_review_length: 10
```

Listing 5 – dbt$_project.yml$

B.3 Modèles DBT - Exemples

```
{{ config(
    materialized='table',
    indexes=[
      {'columns': ['bank_key'], 'unique': True},
      {'columns': ['bank_name']}
    ]
) }}

WITH bank_data AS (
    SELECT DISTINCT
        bank_name,
        bank_type,
        headquarters_city,
        CASE
            WHEN bank_name LIKE '%BMCE%' THEN 'Priv e'
            WHEN bank_name LIKE '%CIH%' THEN 'Publique'
            ELSE 'Mixte'
        END AS ownership_type,
        established_year,
        CURRENT_TIMESTAMP AS created_at,
        CURRENT_TIMESTAMP AS updated_at
    FROM {{ ref('stg_reviews') }}
    WHERE bank_name IS NOT NULL
)

SELECT
    {{ dbt_utils.surrogate_key(['bank_name']) }} AS bank_key,
    bank_name,
    bank_type,
    headquarters_city,
    ownership_type,
    established_year,
    created_at,
    updated_at
FROM bank_data
```

Listing 6 – models/marts/dim$_bank.sql$

```
{{ config(
    materialized='table',
    indexes=[
      {'columns': ['review_date']},
      {'columns': ['bank_key', 'branch_key']},
      {'columns': ['sentiment_key']}
    ]
) }}

WITH review_facts AS (
    SELECT
        r.review_id,
```

```
        db.bank_key,
        dbr.branch_key,
        dl.location_key,
        ds.sentiment_key,
        r.review_date,
        r.rating,
        r.review_length,
        r.sentiment_score,
        r.confidence_score,
        CASE
            WHEN r.rating >= 4 THEN 1
            ELSE 0
        END AS is_positive_rating,
        CASE
            WHEN r.sentiment_label = 'POSITIVE' THEN 1
            ELSE 0
        END AS is_positive_sentiment,
        1 AS review_count,
        CURRENT_TIMESTAMP AS created_at
    FROM {{ ref('stg_reviews') }} r
    LEFT JOIN {{ ref('dim_bank') }} db
        ON r.bank_name = db.bank_name
    LEFT JOIN {{ ref('dim_branch') }} dbr
        ON r.branch_name = dbr.branch_name
        AND db.bank_key = dbr.bank_key
    LEFT JOIN {{ ref('dim_location') }} dl
        ON r.city = dl.city
        AND r.region = dl.region
    LEFT JOIN {{ ref('dim_sentiment') }} ds
        ON r.sentiment_label = ds.sentiment_label
    WHERE r.review_text IS NOT NULL
      AND LENGTH(r.review_text) >= {{ var('min_review_length') }}
)

SELECT * FROM review_facts
```

Listing 7 – models/marts/fact$_r$*eviews.sql*

Annexe C : Requêtes d'Analyse

C.1 KPIs de Satisfaction

```
-- KPI de satisfaction globale par banque
WITH bank_metrics AS (
    SELECT
        db.bank_name,
        COUNT(*) as total_reviews,
        AVG(fr.rating) as avg_rating,
        AVG(fr.sentiment_score) as avg_sentiment,
        SUM(fr.is_positive_sentiment) * 100.0 / COUNT(*) as
   positive_sentiment_pct,
        SUM(fr.is_positive_rating) * 100.0 / COUNT(*) as
   positive_rating_pct
    FROM {{ ref('fact_reviews') }} fr
    JOIN {{ ref('dim_bank') }} db ON fr.bank_key = db.bank_key
    WHERE fr.review_date >= CURRENT_DATE - INTERVAL '12 months'
    GROUP BY db.bank_name
```

```
),
ranking AS (
    SELECT
        *,
        ROW_NUMBER() OVER (ORDER BY avg_rating DESC,
   positive_sentiment_pct DESC) as satisfaction_rank
    FROM bank_metrics
)
SELECT * FROM ranking ORDER BY satisfaction_rank;
```

Listing 8 – Calcul des KPIs principaux

C.2 Analyse Temporelle

```
SELECT
    DATE_TRUNC('month', fr.review_date) as review_month,
    db.bank_name,
    COUNT(*) as review_count,
    AVG(fr.rating) as avg_rating,
    SUM(CASE WHEN ds.sentiment_label = 'POSITIVE' THEN 1 ELSE 0 END) *
   100.0 / COUNT(*) as positive_pct,
    SUM(CASE WHEN ds.sentiment_label = 'NEGATIVE' THEN 1 ELSE 0 END) *
   100.0 / COUNT(*) as negative_pct,
    SUM(CASE WHEN ds.sentiment_label = 'NEUTRAL' THEN 1 ELSE 0 END) *
   100.0 / COUNT(*) as neutral_pct
FROM {{ ref('fact_reviews') }} fr
JOIN {{ ref('dim_bank') }} db ON fr.bank_key = db.bank_key
JOIN {{ ref('dim_sentiment') }} ds ON fr.sentiment_key = ds.
   sentiment_key
WHERE fr.review_date >= CURRENT_DATE - INTERVAL '24 months'
GROUP BY DATE_TRUNC('month', fr.review_date), db.bank_name
ORDER BY review_month DESC, db.bank_name;
```

Listing 9 – Évolution mensuelle du sentiment

Annexe D : Scripts d'Automatisation

D.1 Script de Monitoring

```
import logging
import psycopg2
from datetime import datetime, timedelta
import smtplib
from email.mime.text import MIMEText

class PipelineMonitor:
    def __init__(self, db_config, email_config):
        self.db_config = db_config
        self.email_config = email_config
        self.logger = self._setup_logging()

    def _setup_logging(self):
        logging.basicConfig(
            level=logging.INFO,
            format='%(asctime)s - %(levelname)s - %(message)s',
```

```
            handlers=[
                logging.FileHandler('/var/log/pipeline_monitor.log'),
                logging.StreamHandler()
            ]
        )
        return logging.getLogger(__name__)

    def check_data_freshness(self):
        """V rifie la fra cheur des donn es"""
        try:
            conn = psycopg2.connect(**self.db_config)
            cursor = conn.cursor()

            query = """
            SELECT
                MAX(review_date) as last_review_date,
                COUNT(*) as total_reviews,
                COUNT(CASE WHEN review_date >= CURRENT_DATE - INTERVAL
'1 day' THEN 1 END) as recent_reviews
            FROM star_schema.fact_reviews
            """

            cursor.execute(query)
            result = cursor.fetchone()

            last_date, total_reviews, recent_reviews = result
            days_since_last = (datetime.now().date() - last_date).days

            if days_since_last > 2:
                self._send_alert(
                    f"Donn es obsol tes d tect es",
                    f"Derni re donn e: {last_date}, il y a {
days_since_last} jours"
                )

            self.logger.info(f"Contr le fra cheur: {recent_reviews}
nouveaux avis sur {total_reviews} total")

        except Exception as e:
            self.logger.error(f"Erreur lors du contr le de fra cheur:
{e}")
            self._send_alert("Erreur de monitoring", str(e))
        finally:
            if 'conn' in locals():
                conn.close()

    def check_data_quality(self):
        """V rifie la qualit  des donn es"""
        quality_checks = [
            {
                'name': 'Pourcentage de valeurs nulles',
                'query': """
                SELECT
                    COUNT(CASE WHEN sentiment_score IS NULL THEN 1 END)
* 100.0 / COUNT(*) as null_pct
                FROM star_schema.fact_reviews
                WHERE review_date >= CURRENT_DATE - INTERVAL '7 days'
                """,
```

```
                'threshold': 5.0,
                'operator': '<'
            },
            {
                'name': 'Distribution des sentiments',
                'query': """
                SELECT
                    SUM(CASE WHEN is_positive_sentiment = 1 THEN 1 ELSE
0 END) * 100.0 / COUNT(*) as positive_pct
                FROM star_schema.fact_reviews
                WHERE review_date >= CURRENT_DATE - INTERVAL '7 days'
                """,
                'threshold': 20.0,
                'operator': '>'
            }
        ]

        try:
            conn = psycopg2.connect(**self.db_config)
            cursor = conn.cursor()

            for check in quality_checks:
                cursor.execute(check['query'])
                result = cursor.fetchone()[0]

                if check['operator'] == '<' and result >= check['
threshold']:
                    self._send_alert(
                        f"Probl me de qualit : {check['name']}",
                        f"Valeur: {result:.2f}%, Seuil: {check['
threshold']}%"
                    )
                elif check['operator'] == '>' and result <= check['
threshold']:
                    self._send_alert(
                        f"Probl me de qualit : {check['name']}",
                        f"Valeur: {result:.2f}%, Seuil: {check['
threshold']}%"
                    )

                self.logger.info(f"{check['name']}: {result:.2f}%")

        except Exception as e:
            self.logger.error(f"Erreur lors du contr le qualit : {e}")
        finally:
            if 'conn' in locals():
                conn.close()

    def _send_alert(self, subject, message):
        """Envoie une alerte par email"""
        try:
            msg = MIMEText(message)
            msg['Subject'] = f"[Pipeline Alert] {subject}"
            msg['From'] = self.email_config['from']
            msg['To'] = self.email_config['to']

            server = smtplib.SMTP(self.email_config['smtp_server'])
            server.starttls()
```

```
            server.login(self.email_config['username'], self.
    email_config['password'])
            server.send_message(msg)
            server.quit()

            self.logger.info(f"Alerte envoy e: {subject}")

        except Exception as e:
            self.logger.error(f"Erreur envoi email: {e}")

if __name__ == "__main__":
    db_config = {
        'host': 'localhost',
        'database': 'bank_reviews',
        'user': 'postgres',
        'password': 'password'
    }

    email_config = {
        'smtp_server': 'smtp.gmail.com',
        'from': 'pipeline@insea.ac.ma',
        'to': 'admin@insea.ac.ma',
        'username': 'pipeline@insea.ac.ma',
        'password': 'app_password'
    }

    monitor = PipelineMonitor(db_config, email_config)
    monitor.check_data_freshness()
    monitor.check_data_quality()
```

Listing 10 – monitoring.py - Surveillance du pipeline

Annexe E : Tests et Validation

E.1 Tests DBT

```
# tests/assert_data_quality.yml
version: 2

models:
  - name: fact_reviews
    description: "Table de faits des avis clients"
    tests:
      - dbt_utils.recency:
          datepart: day
          field: review_date
          interval: 7
    columns:
      - name: review_id
        description: "Identifiant unique de l'avis"
        tests:
          - unique
          - not_null

      - name: rating
        description: "Note attribu e (1-5)"
        tests:
```

```
          - not_null
          - accepted_values:
              values: [1, 2, 3, 4, 5]

      - name: sentiment_score
        description: "Score de sentiment (-1    1)"
        tests:
          - not_null
          - dbt_utils.accepted_range:
              min_value: -1
              max_value: 1

      - name: bank_key
        description: "Cl      trangre    vers dim_bank"
        tests:
          - not_null
          - dbt_utils.relationships_where:
              to: ref('dim_bank')
              field: bank_key

  - name: dim_bank
    columns:
      - name: bank_key
        tests:
          - unique
          - not_null
      - name: bank_name
        tests:
          - not_null
          - dbt_utils.not_empty_string

macros:
  - name: test_sentiment_distribution
    description: "V rifie la distribution des sentiments"
    sql: |
      SELECT
        SUM(CASE WHEN is_positive_sentiment = 1 THEN 1 ELSE 0 END) *
   100.0 / COUNT(*) as positive_pct
      FROM {{ ref('fact_reviews') }}
      HAVING positive_pct < 10 OR positive_pct > 90
```

Listing 11 – Tests de qualité des données

E.2 Tests Unitaires Python

```
import unittest
import pandas as pd
from sentiment_analysis import SentimentAnalyzer

class TestSentimentAnalysis(unittest.TestCase):

    def setUp(self):
        self.analyzer = SentimentAnalyzer()

    def test_positive_sentiment(self):
        """Test de d tection sentiment positif"""
        text = "Service excellent, personnel tr s accueillant"
```

```
        result = self.analyzer.analyze_sentiment(text)

        self.assertEqual(result['sentiment'], 'POSITIVE')
        self.assertGreater(result['confidence'], 0.7)

    def test_negative_sentiment(self):
        """Test de d tection sentiment n gatif"""
        text = "Attente tr s longue, service d cevant"
        result = self.analyzer.analyze_sentiment(text)

        self.assertEqual(result['sentiment'], 'NEGATIVE')
        self.assertGreater(result['confidence'], 0.7)

    def test_multilingual_support(self):
        """Test support multilingue"""
        texts = [
            "                              ",  # Arabe
            "Service tr s bon",   # Fran ais
            "Very good service"   # Anglais
        ]

        for text in texts:
            result = self.analyzer.analyze_sentiment(text)
            self.assertIn(result['sentiment'], ['POSITIVE', 'NEGATIVE',
   'NEUTRAL'])
            self.assertIsInstance(result['confidence'], float)

    def test_batch_processing(self):
        """Test traitement par lots"""
        texts = [
            "Excellent service",
            "Service moyen",
            "Service d cevant"
        ]

        results = self.analyzer.analyze_batch(texts)

        self.assertEqual(len(results), len(texts))
        for result in results:
            self.assertIn('sentiment', result)
            self.assertIn('confidence', result)

if __name__ == '__main__':
    unittest.main()
```

Listing 12 – test$_{s}entiment_{a}nalysis.py$

Annexe F : Guide de Déploiement

F.1 Instructions d'Installation

```
#!/bin/bash
# setup.sh - Script d'installation automatis e

echo "=== Installation du pipeline Data Warehouse ==="

# 1. Cr ation de l'environnement virtuel
```

```
python3 -m venv venv_dw
source venv_dw/bin/activate

# 2. Installation des d pendances
pip install --upgrade pip
pip install -r requirements.txt

# 3. Configuration PostgreSQL
createdb bank_reviews
psql -d bank_reviews -f schema/init_database.sql

# 4. Configuration Airflow
export AIRFLOW_HOME=$(pwd)/airflow
airflow db init
airflow users create \
    --username admin \
    --firstname Admin \
    --lastname User \
    --role Admin \
    --email admin@insea.ac.ma \
    --password admin

# 5. Configuration DBT
cd dbt_project
dbt deps
dbt debug
dbt seed
dbt run
dbt test

# 6. Configuration des connexions Airflow
python setup_connections.py

echo "=== Installation termin e ==="
echo "D marrage:"
echo "1. airflow scheduler"
echo "2. airflow webserver"
echo "3. Ouvrir http://localhost:8080"
```

Listing 13 – Installation et configuration

F.2 Configuration de Production

```
version: '3.8'

services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow
      POSTGRES_DB: airflow
    volumes:
      - postgres_db_volume:/var/lib/postgresql/data
      - ./sql/init.sql:/docker-entrypoint-initdb.d/init.sql
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "airflow"]
```

```
      interval: 5s
      retries: 5
    restart: always

  redis:
    image: redis:latest
    expose:
      - 6379
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 5s
      timeout: 30s
      retries: 50
    restart: always

  airflow-webserver:
    build: .
    command: webserver
    ports:
      - "8080:8080"
    depends_on:
      postgres:
        condition: service_healthy
      redis:
        condition: service_healthy
    environment:
      AIRFLOW__CORE__EXECUTOR: CeleryExecutor
      AIRFLOW__DATABASE__SQL_ALCHEMY_CONN: postgresql+psycopg2://airflow
   :airflow@postgres/airflow
      AIRFLOW__CELERY__RESULT_BACKEND: db+postgresql://airflow:
   airflow@postgres/airflow
      AIRFLOW__CELERY__BROKER_URL: redis://:@redis:6379/0
    volumes:
      - ./dags:/opt/airflow/dags
      - ./logs:/opt/airflow/logs
      - ./plugins:/opt/airflow/plugins
    restart: always

  airflow-scheduler:
    build: .
    command: scheduler
    depends_on:
      postgres:
        condition: service_healthy
      redis:
        condition: service_healthy
    environment:
      AIRFLOW__CORE__EXECUTOR: CeleryExecutor
      AIRFLOW__DATABASE__SQL_ALCHEMY_CONN: postgresql+psycopg2://airflow
   :airflow@postgres/airflow
      AIRFLOW__CELERY__RESULT_BACKEND: db+postgresql://airflow:
   airflow@postgres/airflow
      AIRFLOW__CELERY__BROKER_URL: redis://:@redis:6379/0
    volumes:
      - ./dags:/opt/airflow/dags
      - ./logs:/opt/airflow/logs
      - ./plugins:/opt/airflow/plugins
    restart: always
```

```
  airflow-worker:
    build: .
    command: celery worker
    depends_on:
      postgres:
        condition: service_healthy
      redis:
        condition: service_healthy
    environment:
      AIRFLOW__CORE__EXECUTOR: CeleryExecutor
      AIRFLOW__DATABASE__SQL_ALCHEMY_CONN: postgresql+psycopg2://airflow
   :airflow@postgres/airflow
      AIRFLOW__CELERY__RESULT_BACKEND: db+postgresql://airflow:
   airflow@postgres/airflow
      AIRFLOW__CELERY__BROKER_URL: redis://:@redis:6379/0
    volumes:
      - ./dags:/opt/airflow/dags
      - ./logs:/opt/airflow/logs
      - ./plugins:/opt/airflow/plugins
    restart: always

volumes:
  postgres_db_volume:
```

Listing 14 – docker-compose.yml - Déploiement Docker

Ce rapport académique détaillé présente une analyse complète du projet de Data Warehouse pour l'analyse des avis clients bancaires. Il suit une structure académique rigoureuse et intègre tous les éléments techniques, méthodologiques et pratiques du projet, ainsi qu'une documentation technique complète en annexes pour faciliter la reproduction et l'extension du travail.