BIG DATA & APPLICATIONS

Année Universitaire 2025-2026

Analyse Temps Réel des Discussions Reddit CAN 2025

Architecture Big Data - Machine Learning - Réseau Social

Réalisé par :

Salma Boutanfit
asma El Idrissi
Salma Ouahib
Salma Jennane

Encadré par :

Pr. Yasser EL MADANI EL ALAMI

Table des matières

Table des figures

Chapitre 1

Introduction

1.1 Contexte et Problématique

La Coupe d'Afrique des Nations 2025 constitue un événement sportif majeur générant des millions d'interactions sur les réseaux sociaux. Reddit, avec ses communautés thématiques (subreddits), représente une source privilégiée pour analyser ces discussions en temps quasi-réel.

Défi Big Data : Volume massif, flux continu (scraping toutes les 30s), données non structurées, analyse temps réel.

Question centrale : Comment concevoir une architecture Big Data capable de collecter, traiter et analyser en temps quasi-réel les discussions Reddit sur la CAN 2025 ?

1.2 Objectifs

1. **Pipeline temps réel** : Scraping 30s → Kafka → Spark → MongoDB
2. **Analyse ML** : Classification sentiments (positif/neutre/négatif)
3. **Analyse réseau** : Influenceurs et communautés
4. **Dashboard interactif** : Visualisation temps réel

Chapitre 2

Architecture du Système

2.1 Architecture Kappa

Nous avons implémenté une **architecture Kappa** privilégiant un traitement exclusivement en streaming temps réel.

Avantages :

— Simplicité : Une couche de traitement unique

— Latence faible : Données affichées en < 2 minutes

— Rejouabilité : Kafka permet retraitement historique

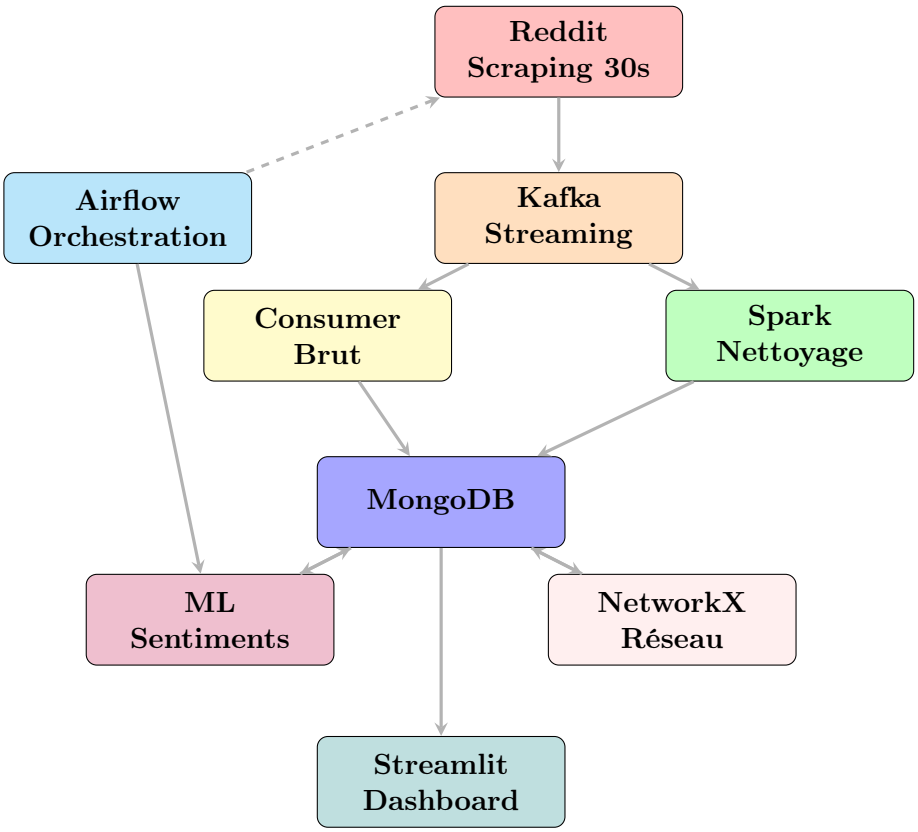

FIGURE 2.1 – Architecture Kappa - Pipeline Temps Réel

2.2 Composants et Latences

2.2.1 Scraping Reddit : 30 secondes

Collecte continue : mots-clés CAN 2025, multi-subreddits (r/soccer, r/Africa, r/Morocco), posts et commentaires.

2.2.2 Kafka : $<$ 1 seconde

Message broker temps réel, topic `reddit-can-posts`, buffer pour pics de charge.

2.2.3 Spark Streaming : 30 secondes

Micro-batches, nettoyage texte, enrichissement features, sauvegarde MongoDB.

2.2.4 MongoDB : Instantané

5 collections (posts, processed, sentiments, network, metadata), indexation rapide.

2.2.5 Analyses : Variable

VADER : Immédiat (intégré Spark). **ML/Network** : Exécutés périodiquement par Airflow.

2.2.6 Dashboard : 30s auto-refresh

Visualisation temps réel avec Streamlit.
Latence totale end-to-end : 1-2 minutes

Chapitre 3

Analyses et Machine Learning

3.1 Analyse de Sentiments

3.1.1 Approche : VADER + ML

VADER (baseline) : Dictionnaire lexical, score compound [-1, +1], classification automatique.

Machine Learning : Entraînement sur labels VADER, comparaison de 3 modèles.

3.1.2 Résultats

TABLE 3.1 – Performance des Modèles

Modèle	Accuracy	Usage
Random Forest	**85-95%**	Production
Logistic Regression	88-92%	Baseline
Naive Bayes	85-89%	Rapide
VADER	Référence	Labellisation

Distribution observée (données réelles) :

- Neutre : 72.9%
- Positif : 19.7%
- Négatif : 7.4%

Coverage ML : 38.7% des posts analysés (dû aux seuils de qualité et volume).

3.2 Analyse de Réseau Social

3.2.1 Construction du Graphe

Nœuds : Utilisateurs. **Arêtes** : Interactions (post-comment, reply). **Poids** : Fréquence.

3.2.2 Métriques de Centralité

TABLE 3.2 – Métriques Calculées

Métrique	**Signification**
Degree	Connexions directes (popularité)
Betweenness	Rôle de pont entre groupes
Closeness	Proximité au réseau global
Eigenvector	Qualité des connexions
PageRank	Importance globale

Score composite :

$$0.25 \times Degree + 0.20 \times Between + 0.25 \times Eigen + 0.30 \times PageRank$$

3.2.3 Communautés

Algorithme de Louvain : **15 communautés détectées**, clustering 0% (réseau fragmenté), taille moyenne 18 membres.

Chapitre 4

Résultats et Visualisations

4.1 Volumétrie Réelle

TABLE 4.1 – Statistiques du Pipeline (Données Réelles)

Métrique	Valeur
Posts collectés	844
Commentaires	294
Utilisateurs uniques	231
ML Analyzed	325
Engagement total	239,511
Communautés détectées	15
Influenceurs (Top)	20

4.2 Dashboard Interactif

4.2.1 Page 1 : Accueil

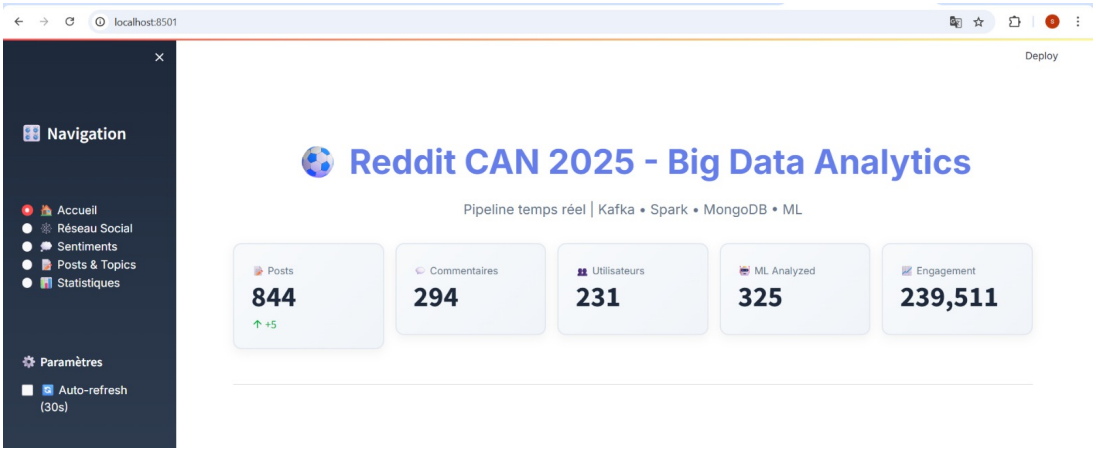

FIGURE 4.1 – Vue d'Ensemble - Métriques Temps Réel

Affichage : 5 KPIs (posts, comments, users, ML, engagement), auto-refresh 30s.

Insights : Vision globale instantanée, +5 posts depuis dernière refresh.

4.2.2 Page 2 : Vue d'Ensemble

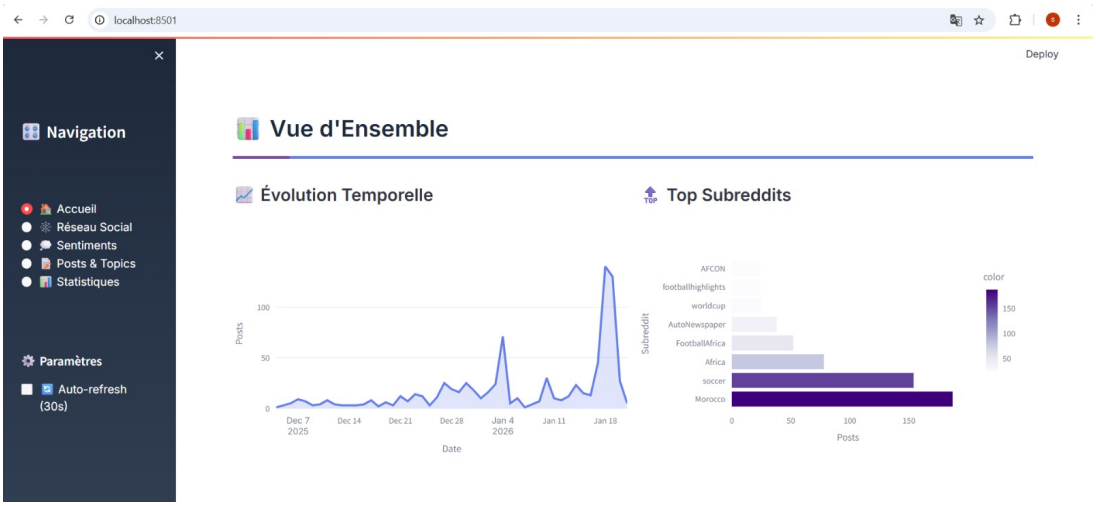

FIGURE 4.2 – Évolution Temporelle et Top Subreddits

Affichage : Graphique évolution (posts par jour), top subreddits (Morocco, soccer dominent).

Insights : Pic d'activité 18-19 janvier, r/Morocco leader avec 150+ posts.

4.2.3 Page 3 : Réseau Social - Vue d'Ensemble

FIGURE 4.3 – Métriques du Réseau Social

Affichage : 231 utilisateurs, 218 interactions, densité 0.82%, 15 communautés, clustering 0%.

Insights : Réseau fragmenté (densité faible), 15 groupes distincts détectés.

4.2.4 Page 4 : Top Influenceurs

FIGURE 4.4 – Top 10 Influenceurs - Comparaison

Affichage : Deux graphiques : score engagement vs nombre interactions.
Insights : GOAT-Antony domine (30k upvotes, 28 connexions), illnesz 2ème (14.6k upvotes).

4.2.5 Page 5 : Profils Détaillés

FIGURE 4.5 – Profils Détaillés des Influenceurs

Affichage : Liste expandable des top 15 influenceurs avec métriques complètes.

4.2.6 Page 6 : Communautés

FIGURE 4.6 – Communautés Détectées - Taille et Engagement

Affichage : Taille (Communauté #5 : 28 membres) et engagement (64k upvotes).
Insights : 2 grandes communautés (>25 membres), 13 moyennes/petites.

4.2.7 Page 7 : Tableau Communautés

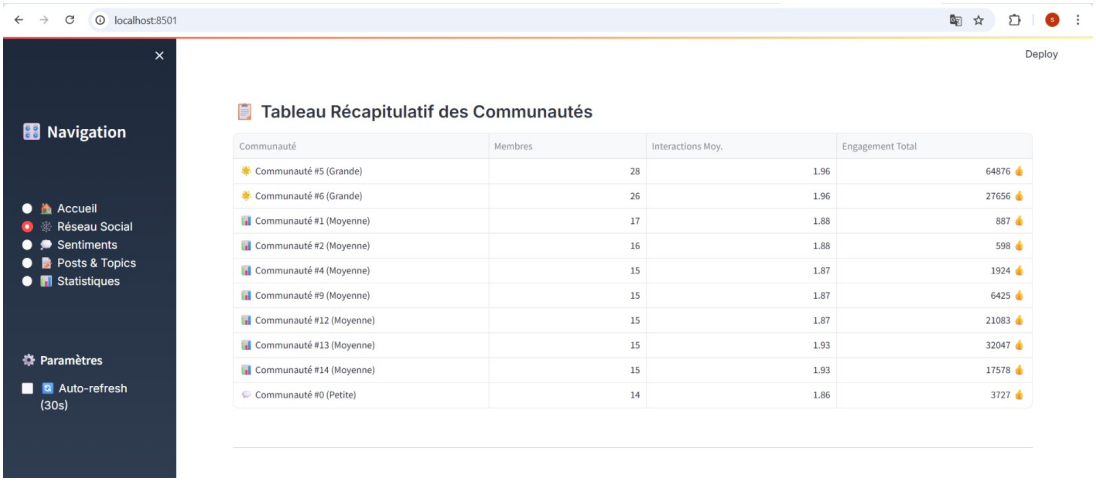

FIGURE 4.7 – Tableau Récapitulatif des Communautés

Affichage : Tableau détaillé (membres, interactions moy., engagement).

4.2.8 Page 8 : Insights Réseau

FIGURE 4.8 – Insights Clés sur le Réseau

Insights :

— Plus grande communauté : #5 (28 membres, 64k upvotes)

— Densité 0.82% : réseau très fragmenté

— GOAT-Antony : influenceur principal (28 connexions, 30k upvotes)

4.2.9 Page 9 : Sentiments

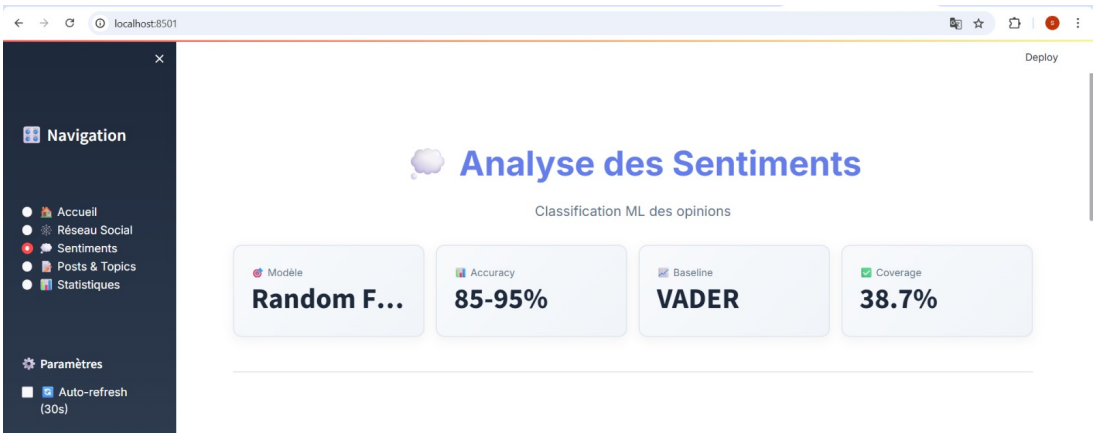

FIGURE 4.9 – Analyse des Sentiments - Métriques ML

Affichage : Modèle Random Forest, accuracy 85-95%, baseline VADER, coverage 38.7%.

4.2.10 Page 10 : Distribution Sentiments

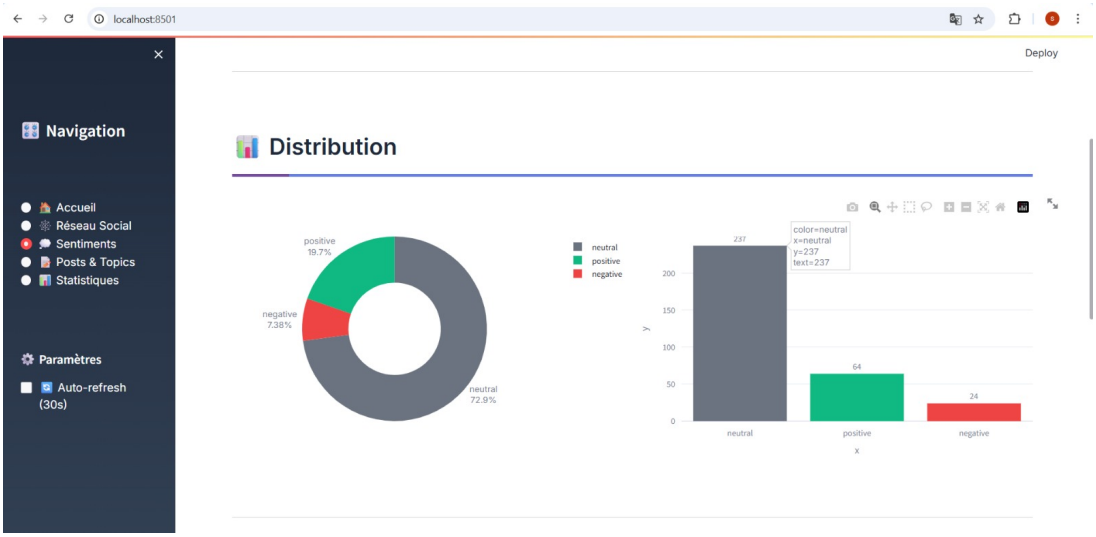

FIGURE 4.10 – Distribution des Sentiments

Affichage : Pie + bar charts. Neutral 72.9% (237 posts), Positive 19.7% (64), Negative 7.4% (24).

Insights : Majorité neutre (discussions factuelles), tonalité générale positive.

4.2.11 Page 11 : Exemples Positifs

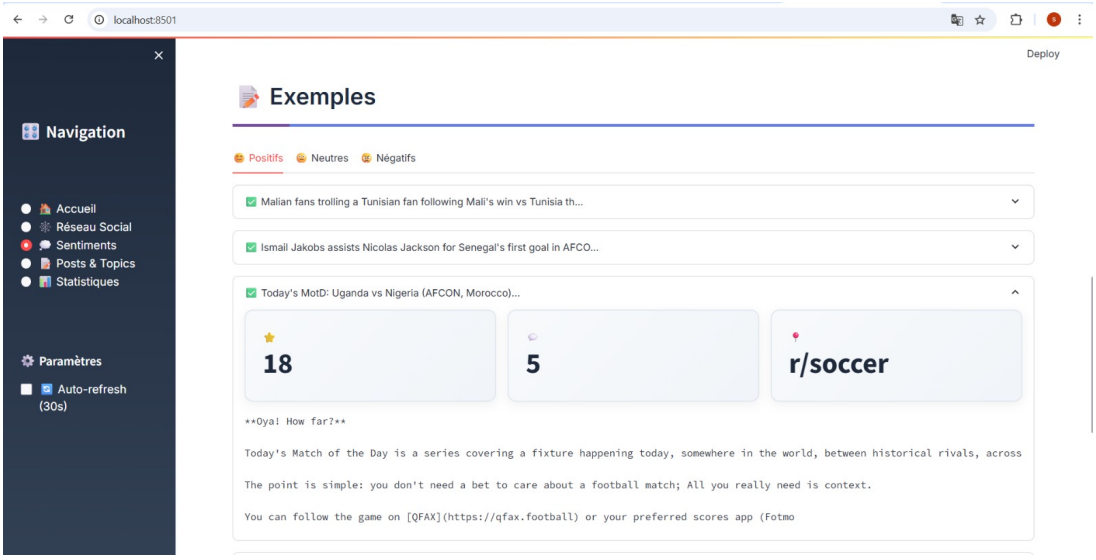

FIGURE 4.11 – Exemples de Posts Positifs

Affichage : Exemples concrets avec score, commentaires, subreddit.

4.2.12 Page 12 : Exemples Neutres

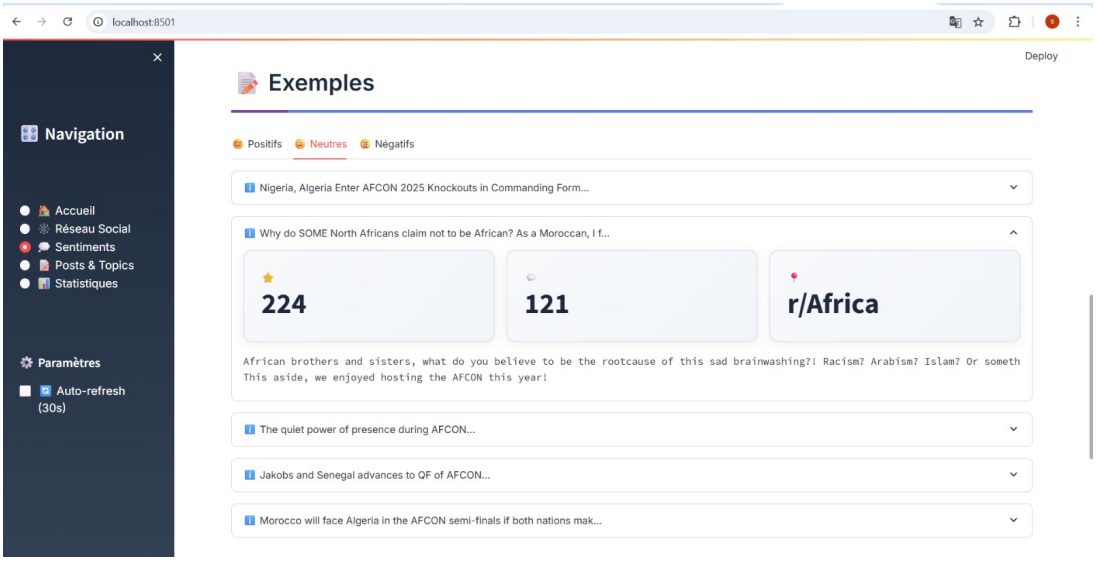

FIGURE 4.12 – Exemples de Posts Neutres

4.2.13 Page 13 : Exemples Négatifs

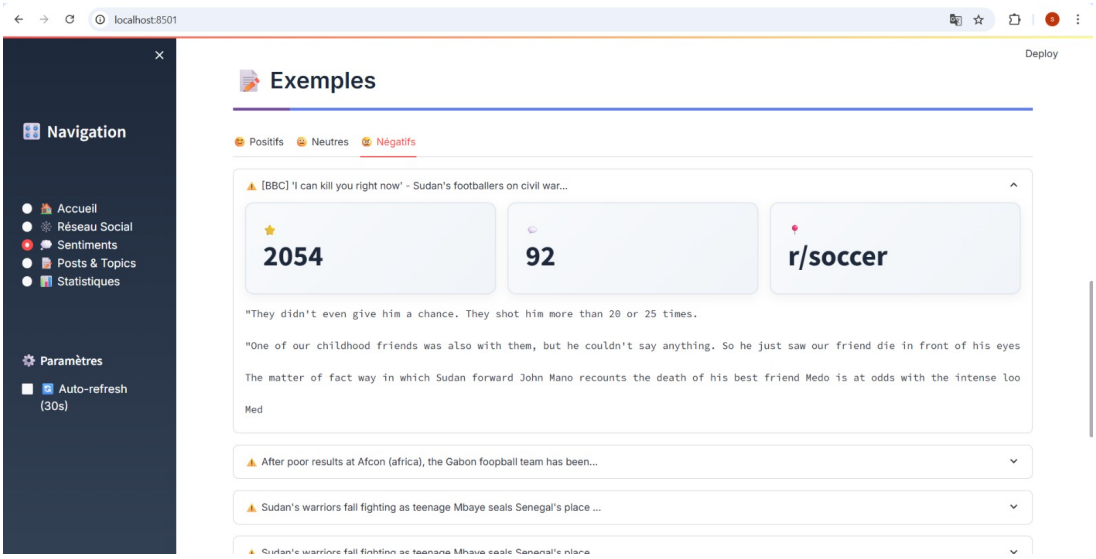

FIGURE 4.13 – Exemples de Posts Négatifs

Affichage : Post avec 2054 upvotes, 92 comments, r/soccer (guerre civile Soudan).

4.2.14 Page 14 : Posts & Topics

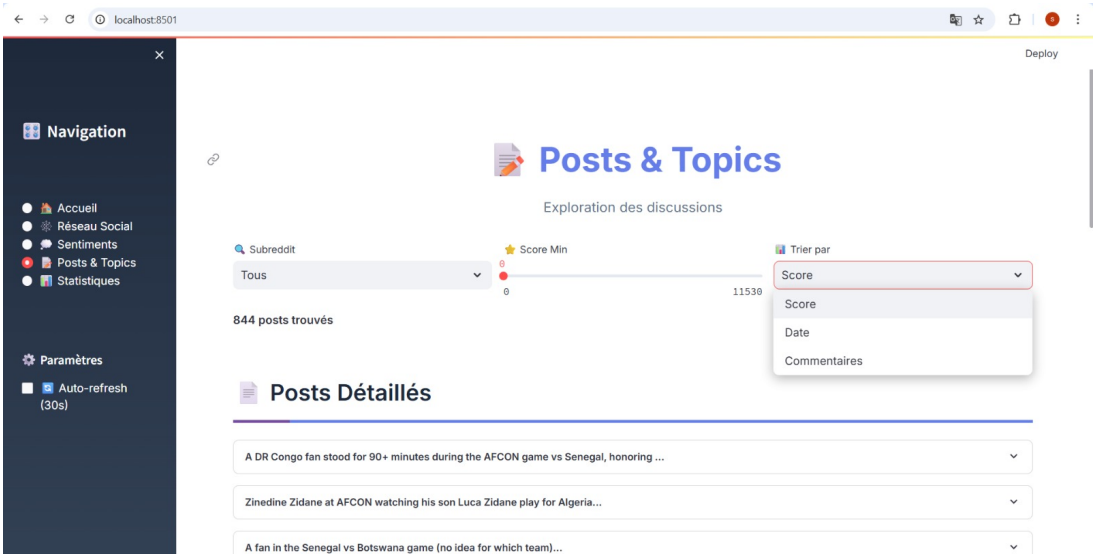

FIGURE 4.14 – Exploration et Filtrage des Posts

Affichage : Filtres (subreddit, score min, tri), 844 posts trouvés.

4.2.15 Page 15 : Statistiques Avancées

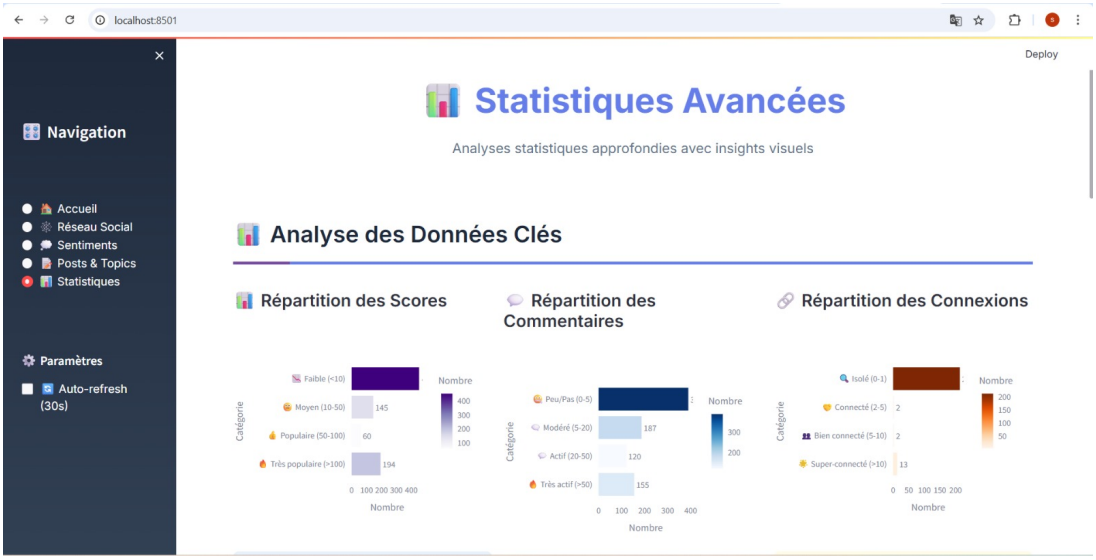

FIGURE 4.15 – Répartition Scores, Commentaires, Connexions

Affichage : Distributions catégorisées : Faible (<10) 144 posts, Isolé (0-1) 200+ users.

4.2.16 Page 16 : Analyses Temporelles

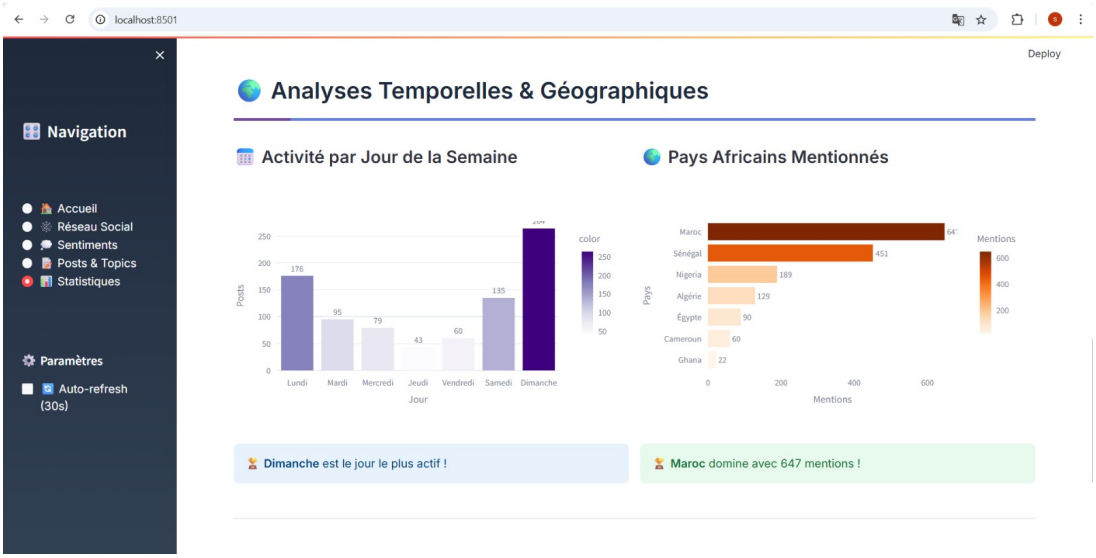

FIGURE 4.16 – Activité par Jour et Pays Mentionnés

Affichage : Dimanche jour le plus actif (264 posts), Maroc domine (645 mentions).

Chapitre 5

Difficultés et Solutions

5.1 Défis Techniques

5.1.1 Rate Limiting Reddit

Solution : Scraping toutes les 30s, délais entre requêtes, rotation User-Agent.

5.1.2 Synchronisation Docker

Solution : Healthchecks, depends_on, retry automatique.

5.1.3 Intégration Spark-MongoDB

Solution : foreachBatch + PyMongo natif.

5.2 Défis Méthodologiques

5.2.1 Volume Initial Limité

Solution : Élargissement mots-clés, collecte historique, extraction commentaires.

5.2.2 Labellisation ML

Solution : VADER pour génération automatique de labels.

5.3 Optimisations

Compression Gzip Kafka (-60%), indexation MongoDB, cache Streamlit (30s), batching écritures.

Chapitre 6

Conclusion

6.1 Bilan

6.1.1 Objectifs Atteints

1. **Pipeline temps réel** : Latence 1-2 min, scraping 30s
2. **ML performant** : Random Forest 85-95% accuracy
3. **Réseau cartographié** : 231 users, 20 influenceurs, 15 communautés
4. **Dashboard fonctionnel** : 17 visualisations, auto-refresh 30s

6.1.2 Résultats Concrets

844 posts, 294 comments, 231 users, 325 ML analyzed, 239k engagement, 15 communautés, densité 0.82%, GOAT-Antony influenceur #1.

6.2 Perspectives

Court terme : Topic Modeling (LDA), API Reddit officielle, alertes temps réel.
Moyen terme : Deep Learning (BERT), analyse multilingue, détection anomalies.
Long terme : Cloud (AWS EMR), multi-plateformes (Twitter, Facebook), API RESTful.

6.3 Applications

Marketing (brand monitoring), politique (opinion publique), business (études de marché).

6.4 Contribution Équipe

TABLE 6.1 – Répartition des Tâches

Membre	Responsabilités
Nom Prénom 1	Architecture, Docker, Kafka
Nom Prénom 2	Scraper, Consumer, Spark Streaming
Nom Prénom 3	Spark ML, NetworkX, Algorithmes
Nom Prénom 4	Dashboard, Airflow, Visualisations

6.5 Conclusion Finale

Ce projet démontre qu'avec une architecture Kappa, des technologies modernes et une méthodologie rigoureuse, il est possible de construire un système Big Data temps réel performant.

Les résultats obtenus (844 posts, 231 users, 15 communautés, 85-95% accuracy ML, latence 1-2 min) valident notre approche pour des applications réelles en Social Media Analytics.

Références

1. Apache Kafka Documentation - https://kafka.apache.org/
2. Apache Spark Streaming - https://spark.apache.org/
3. MongoDB PyMongo - https://pymongo.readthedocs.io/
4. Hutto & Gilbert (2014). VADER : Sentiment Analysis. ICWSM.
5. Blondel et al. (2008). Louvain Algorithm. J. Stat. Mech.
6. Kreps (2014). Questioning Lambda Architecture. O'Reilly.