

---

# *Analyse des réseaux sociaux en temps réel*

## *La cause palestinienne*

---

**Réalisé par :**

Abderahmane Jabiri  
Bouchra Loukili  
Niama Hdadache  
Meriem Errafia

**Membre de jury:**

Mr. EL ALAMI Yasser

# *Plan*

---

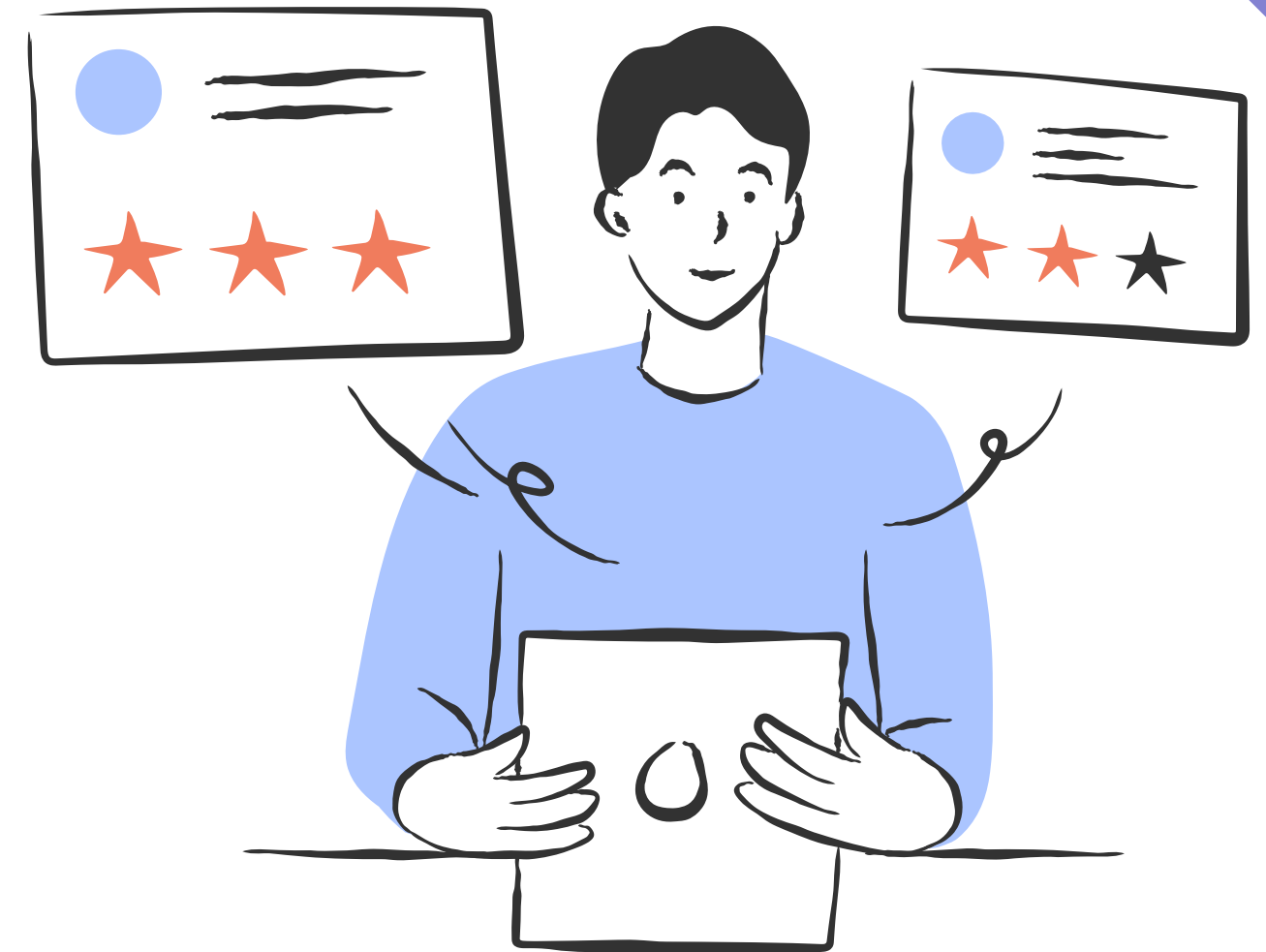
- 01 Périmètre du projet
- 02 Vue globale de l'architecture
- 03 Implémentation et Résultats
- 04 Conclusion et Perspectives



# *1. Périmètre du projet*

# Contexte Général

- Les réseaux sociaux constituent un espace majeur de partage d'opinions et de discussion des enjeux mondiaux.
- Les échanges en ligne génèrent de grandes quantités de données.
- Ces données peuvent être analysées afin de mieux comprendre l'opinion publique et les comportements sociaux.
- Ce projet analyse les discussions sur la cause palestinienne.



# *Problématique*

- Données des réseaux sociaux sont volumineuses, non structurées et générées en continu.
- Comprendre les émotions et les thématiques présentes dans les textes est difficile.

# *Solution Proposée*



**Application Big Data pour la collecte et l'analyse  
des discussions liées à la cause palestinienne**

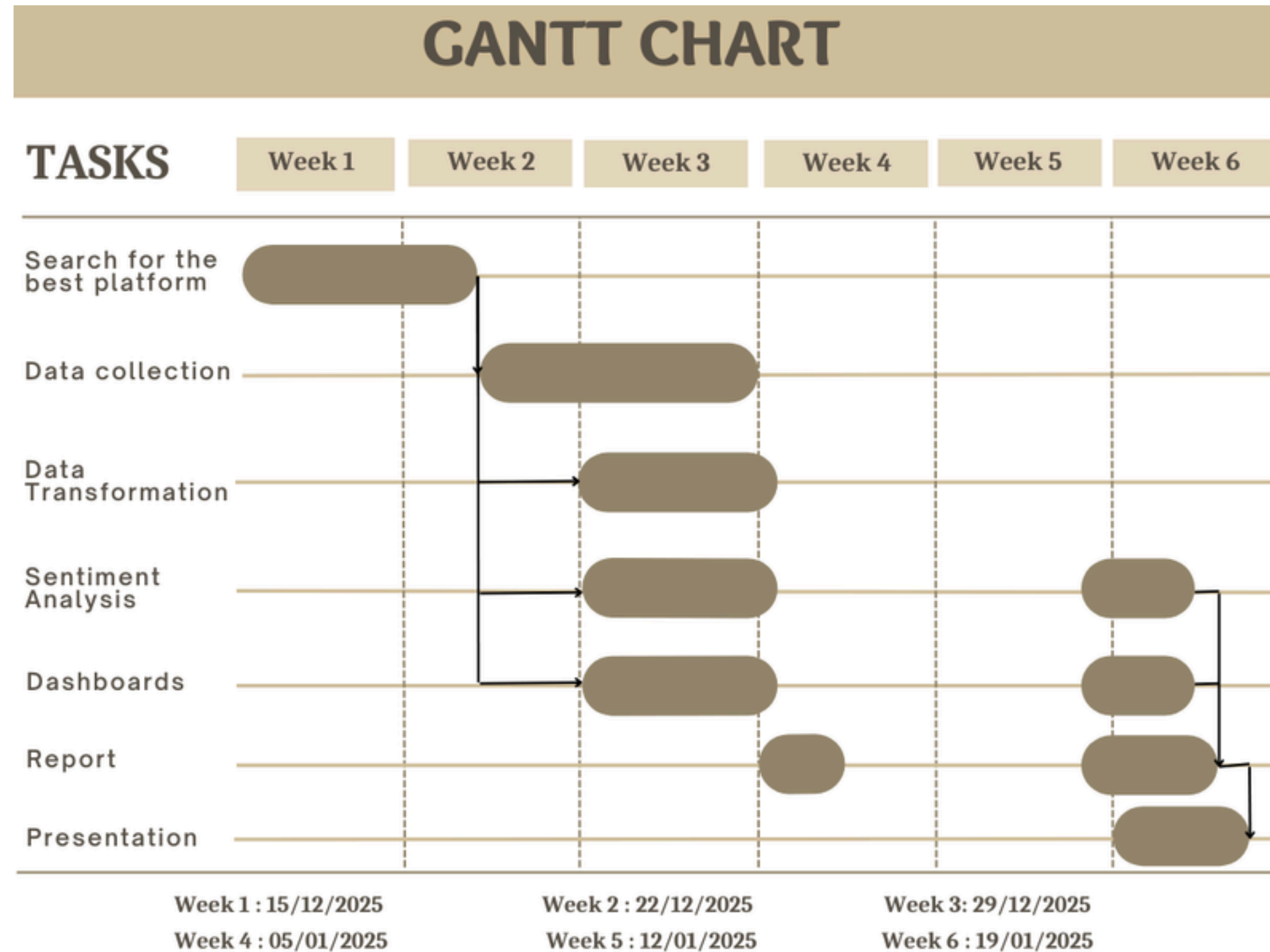




# *Objectives du projet*

- Collecter les données liées à la cause palestinienne
- Transformer et nettoyer les données collectées
- Effectuer une analyse de sentiment
- Créer des tableaux de bord pour visualiser et analyser les résultats

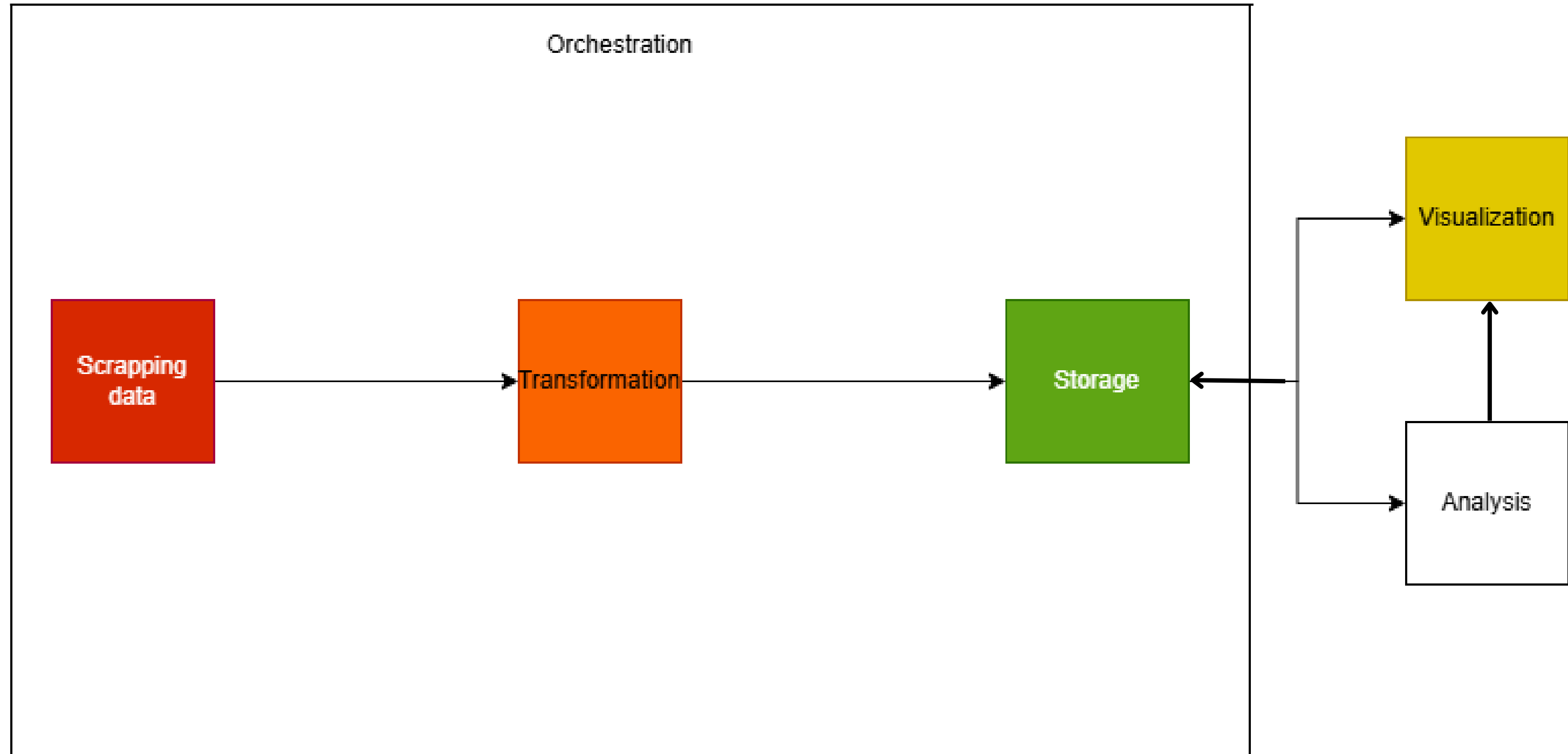
# Planification du projet





## 2. *Vue globale de l'architecture*

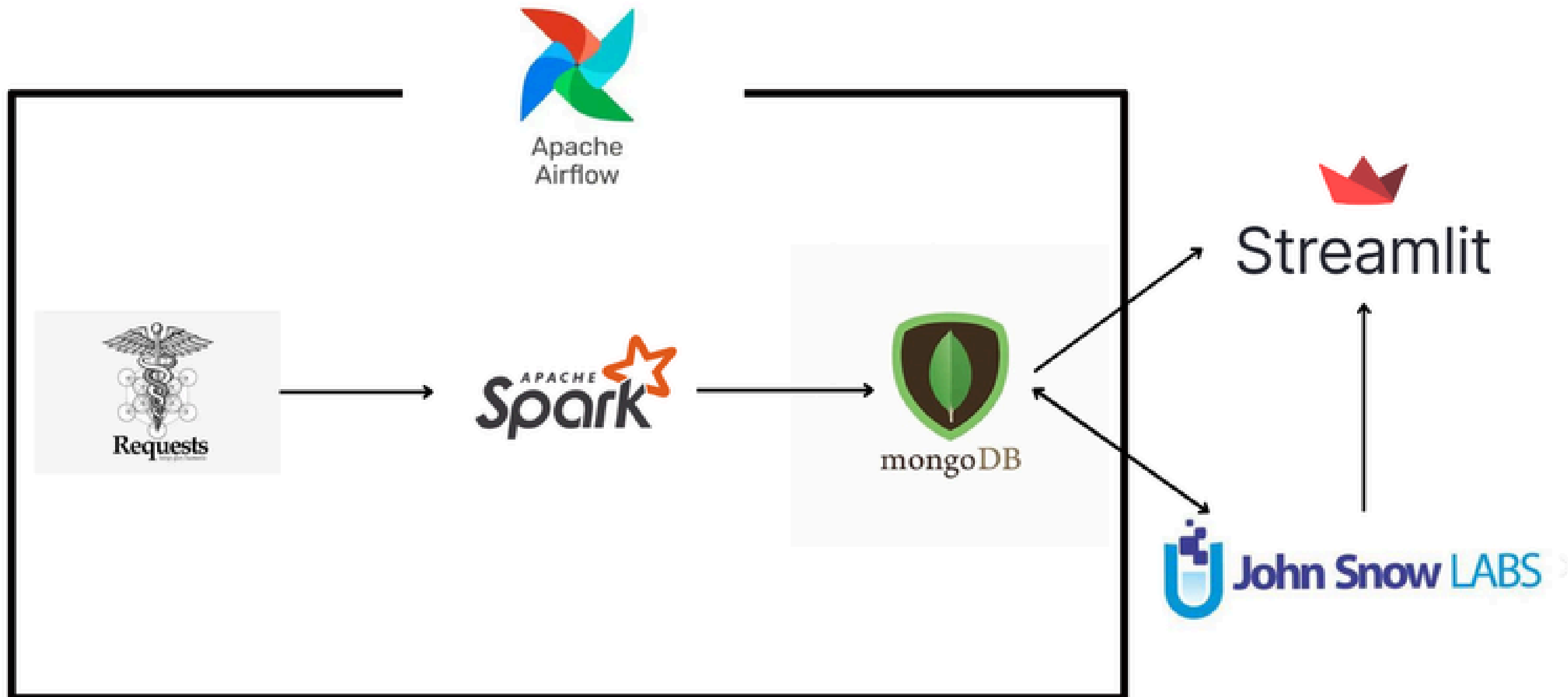
# Vue globale de l'architecture





## *4. Implementation and Results*

# *Technologies utilisées*



# *Phase 0 : Choix de la plateforme*

- Sélectionner la plateforme de réseaux sociaux
- Différentes API ont été testées afin d'accéder aux données de ces plateformes.
- Reddit a été choisi comme principale source de données pour le scraping.



# *Phase 1: Collection des données*

---

## **Méthode de collecte :**

- Web scraping des pages de subreddits publics

## **Subreddits analysés (Top 3):**

- r/Palestine
- r/IsraelPalestine
- r/IsraelUnderAttack

## **Types de données collectées :**

- Publications (posts)
- Commentaires (comments)
- Collecte incrémentale pour éviter les doublons

# *Phase 1: Collection des données*

## **Champs extraits:**

```
{
  "comment_id": "o0jckk9",
  "post_id": "lqhbgkz",
  "subreddit": "Palestine",
  "body": "Wow.. Israel gets more and more desperate by the day. \n\nThey are going to burn the wor",
  "score": 39,
  "parent_id": "t3_lqhbgkz",
  "created_at": "2025-08-12T20:04:07"
},
```

```
{
  "post_id": "lqh7znt",
  "subreddit": "Palestine",
  "title": "Israeli settler attempts to prevent the Abu Hammam family from accessing their property",
  "selftext": "",
  "score": 15,
  "num_comments": 1,
  "created_at": "2026-01-19T15:57:57"
},
```

# *Phase 2: Transformation des données*

---

## **Mode de traitement :**

- Micro-batch (quasi temps réel, chaque 15 min)

## **Opérations appliquées :**

- Nettoyage du texte (URLs, caractères spéciaux \n \t...)
- Filtrage du bruit (bots, messages automatiques, spam)
- Normalisation (minuscules, espaces)
- Détection de la langue (anglais uniquement)
- Suppression des doublons

## **Données traitées :**

- Publications (posts)
- Commentaires (comments)

## *Phase 3: Analyse de sentiments*

### *Objectif:*

*Analyser le sentiment des commentaires Reddit en utilisant trois modèles NLP complémentaires, intégrés dans un pipeline Spark + Airflow.*

### *Models :*

- *Vivekn Sentiment (Spark NLP)*
- *VADER*
- *TextBlob*

# Phase 3: Analyse de sentiments

## Vivekn Sentiment (Spark NLP)

- Inspiré d'un Naive Bayes amélioré
- Algorithme original testé sur dataset IMDB Movie Reviews
- Accuracy rapportée  $\approx 88,8\%$
- Résultat:
  - "positive"
  - "negative"
  - "neutral"

## VADER

Introduit en 2014 par C. J. Hutto et E. Gilbert.

- Structure de réponse :  
{'neg': 0.0, 'neu': 0.238, 'pos': 0.762, 'compound': 0.6369}

Cela signifie :

neg — score du sentiment négatif (0.0 à 1.0)  
neu — score du sentiment neutre  
pos — score du sentiment positif  
compound — score global de sentiment normalisé entre -1 (fortement négatif) et +1 (fortement positif)

☞ Généralement, on interprète le compound score comme suit :

$\geq +0.05 \rightarrow$  positif  
 $\leq -0.05 \rightarrow$  négatif  
entre -0.05 et +0.05  $\rightarrow$  neutre

## Textblob

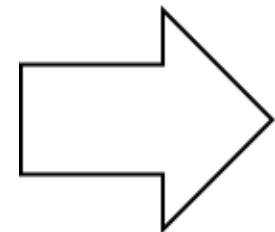
- 📌 Nature du modèle
  - Lexical (rule-based)
  - Pas de machine learning
  - N'apprend pas des données
- 📊 Performances
  - Textes simples / avis courts :  $\sim 55-65\%$  précision
- ⚖ Comparaison
  - Moins précis que VADER (réseaux sociaux)
- 💻 Format de sortie

`TextBlob("I really love this product").sentiment`  
 $\rightarrow$  `Sentiment(polarity=0.5, subjectivity=0.6)`
- 📈 Interprétation des scores
  - polarity  $\in [-1, +1]$  : -1 négatif | 0 neutre | +1 positif
  - subjectivity  $\in [0, 1]$  : 0 objectif | 1 subjectif

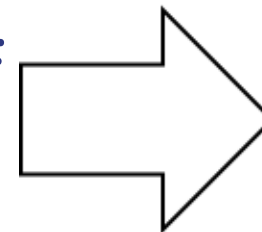
# Phase 3: Analyse de sentiments

*collection clean comments*

```
{
  "_id": {
    "$oid": "6970889bba854c2436bbbec"
  },
  "comment_id": "o0rppjf",
  "post_id": "1qid59k",
  "clean_comment": "maybe he should just disappear",
  "comment_hash": "1996e8434e17f303afd8be6bd1583a95c",
  "score": 9,
  "created_at": "2026-01-21T00:38:07"
}
```



*Modèles distribués :*  
*Vivekn Sentiment*  
*VADER*  
*TextBlob*

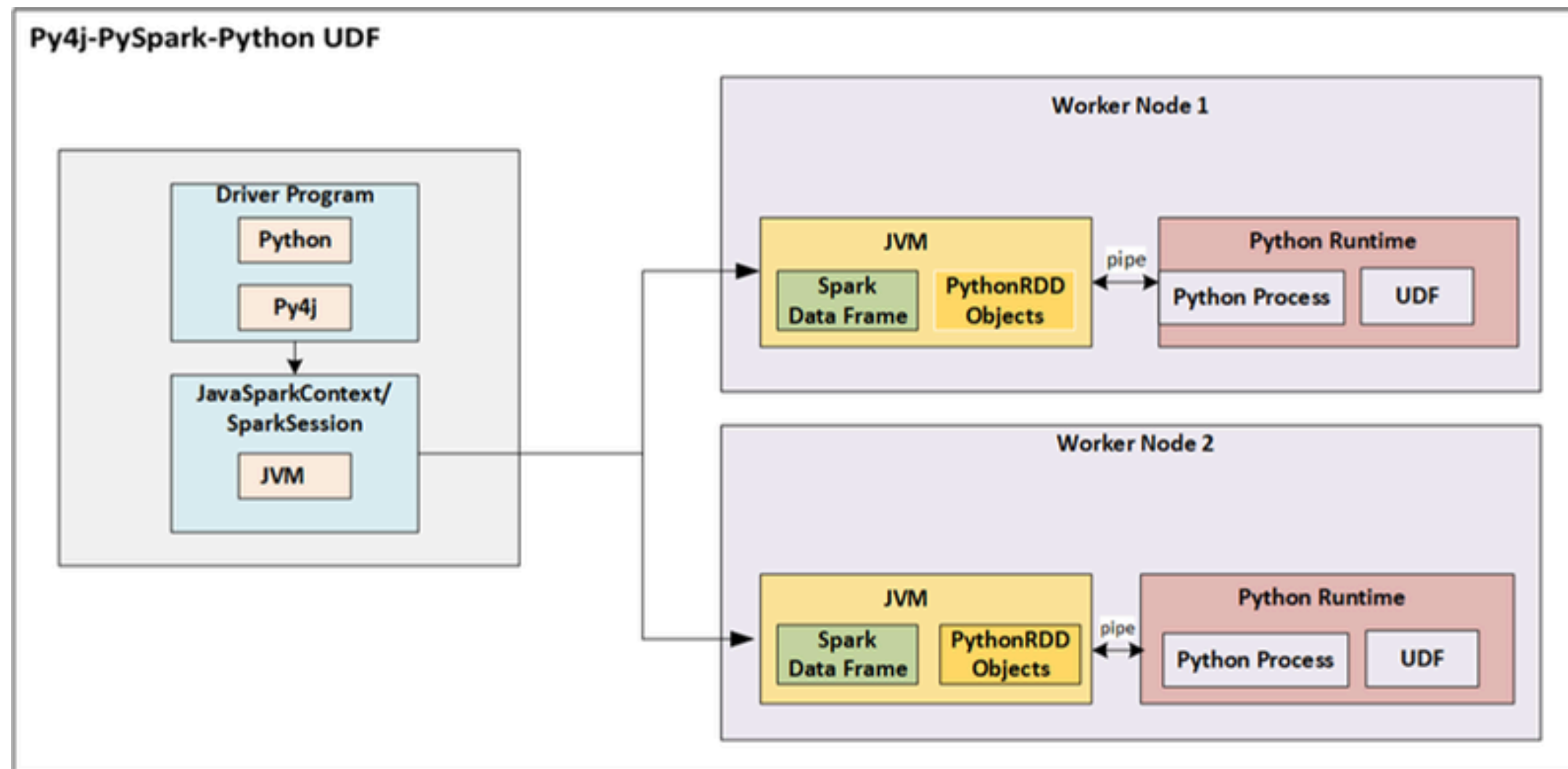


*collection analysed comments*

```
{
  "_id": "6970889bba854c2436bbbec",
  "comment_id": "o0rppjf",
  "post_id": "1qid59k",
  "text": "maybe he should just disappear",
  "score": 9,
  "created_at": "2026-01-21T00:38:07",
  "sentiment_vivekn": {
    "label": "negative",
    "model": "analyze_sentiment_en_VIVEKN",
    "framework": "spark-nlp"
  },
  "vader_sentiment": "negative",
  "vader_confidence": 0.2263,
  "textblob_polarity": 0,
  "textblob_subjectivity": 0
}
```

# Phase 3: Analyse de sentiments

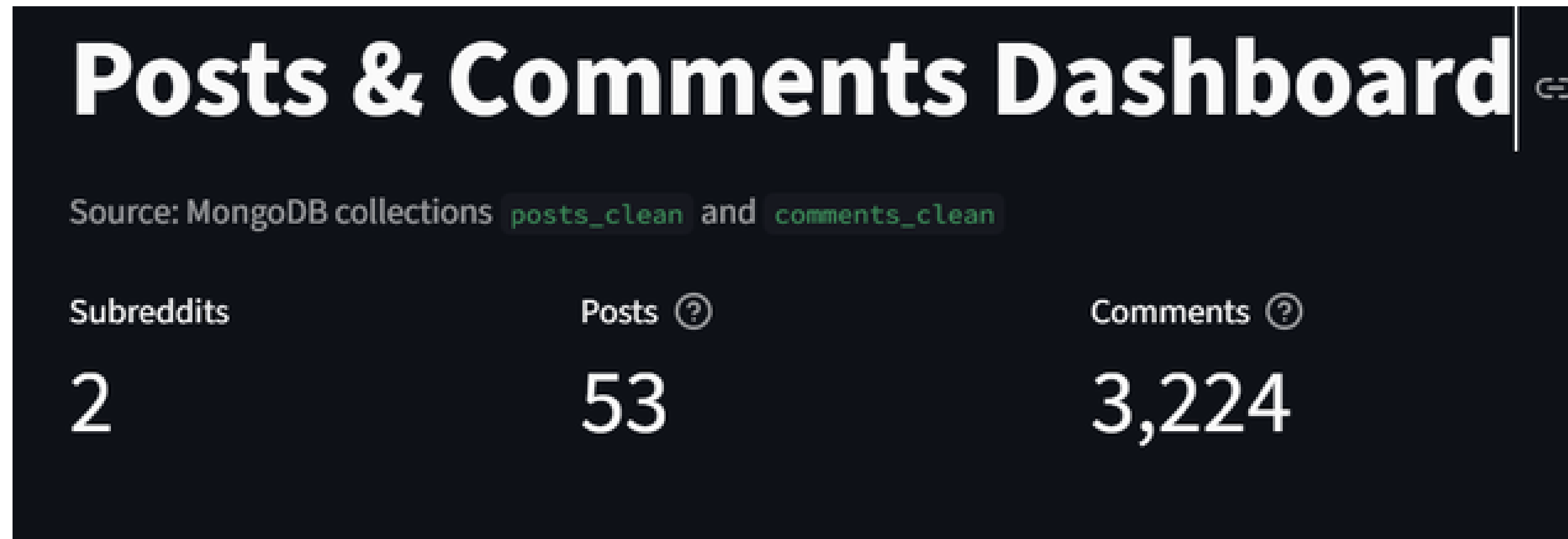
Execution of the models : Vader & Textblob : *pandas-udf*



Execution of the model Vivekn : *Spark-nlp* :run directly on JVM

## Phase 4: Dashboards

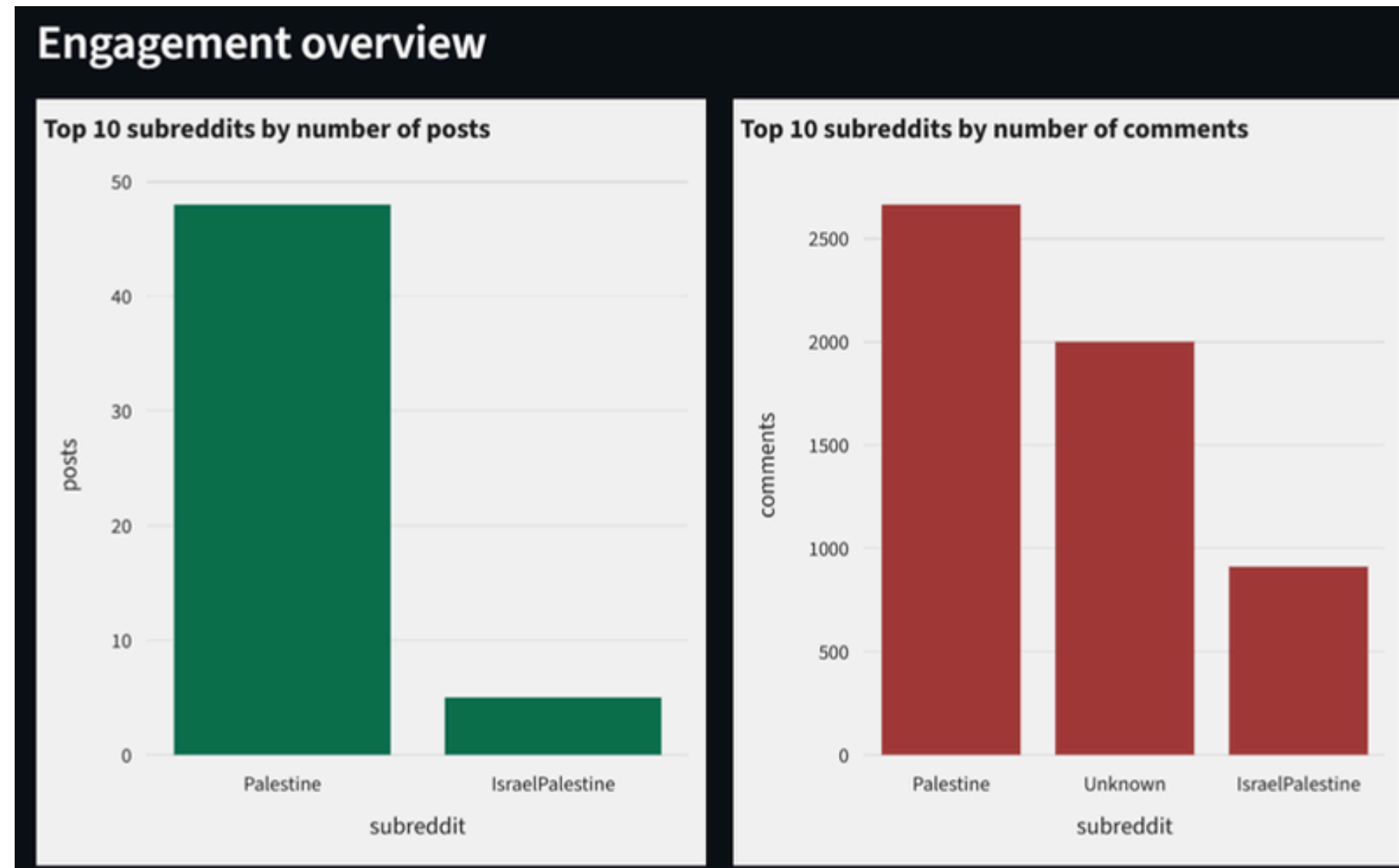
**Dashboard 1:** Vue globale de l'activité des posts et commentaires Reddit



- Nombre total de subreddits analysés
- Volume global de posts collectés
- Volume total de commentaires associés

# Phase 4: Dashboards

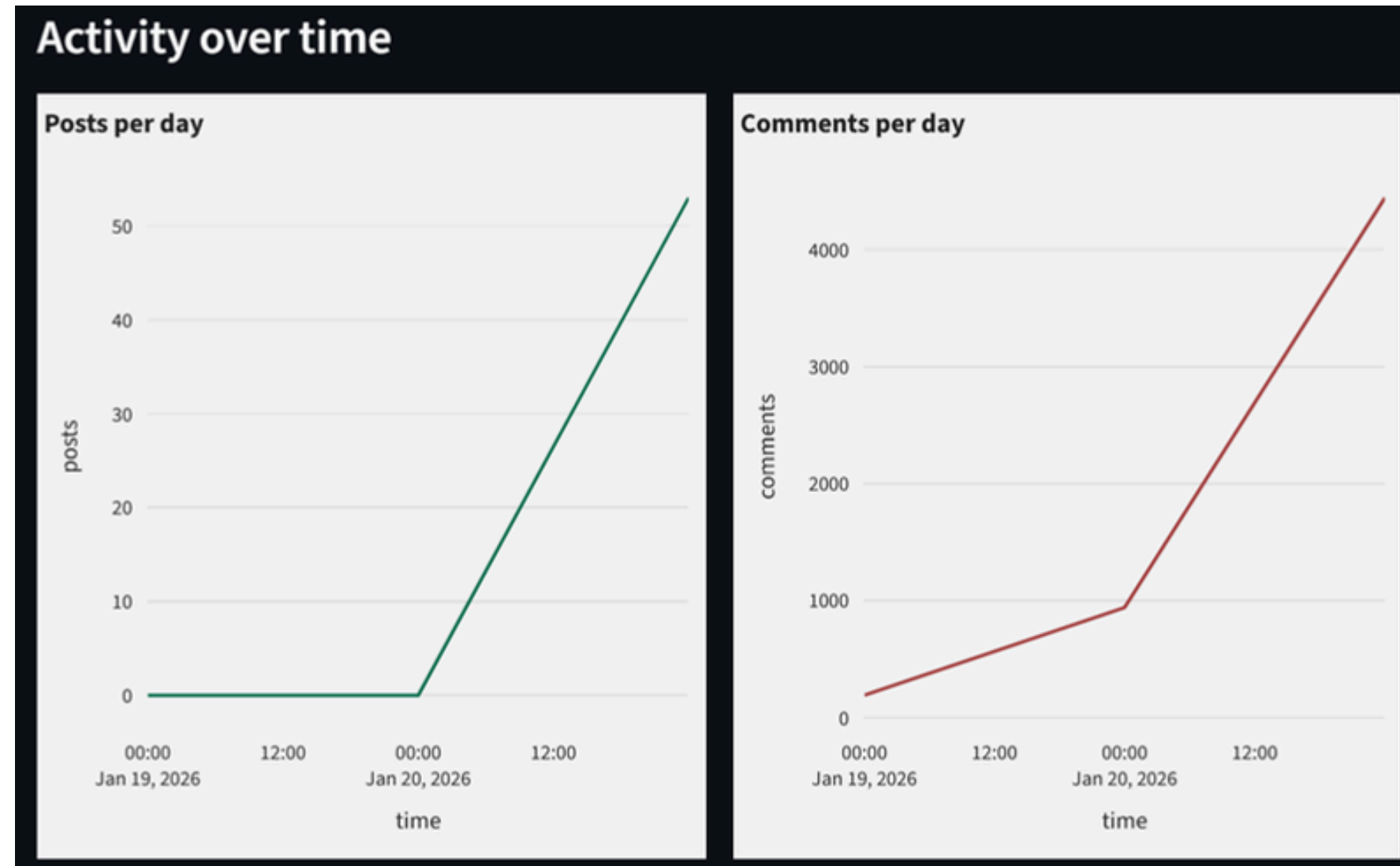
## Répartition de l'engagement par subreddit



- Ce visuel montre le nombre de posts et de commentaires par subreddit
- Le graphique de gauche représente la création de posts
- Le graphique de droite représente l'engagement via les commentaires
- Il permet d'identifier les subreddits les plus actifs et les plus engageants

# Phase 4: Dashboards

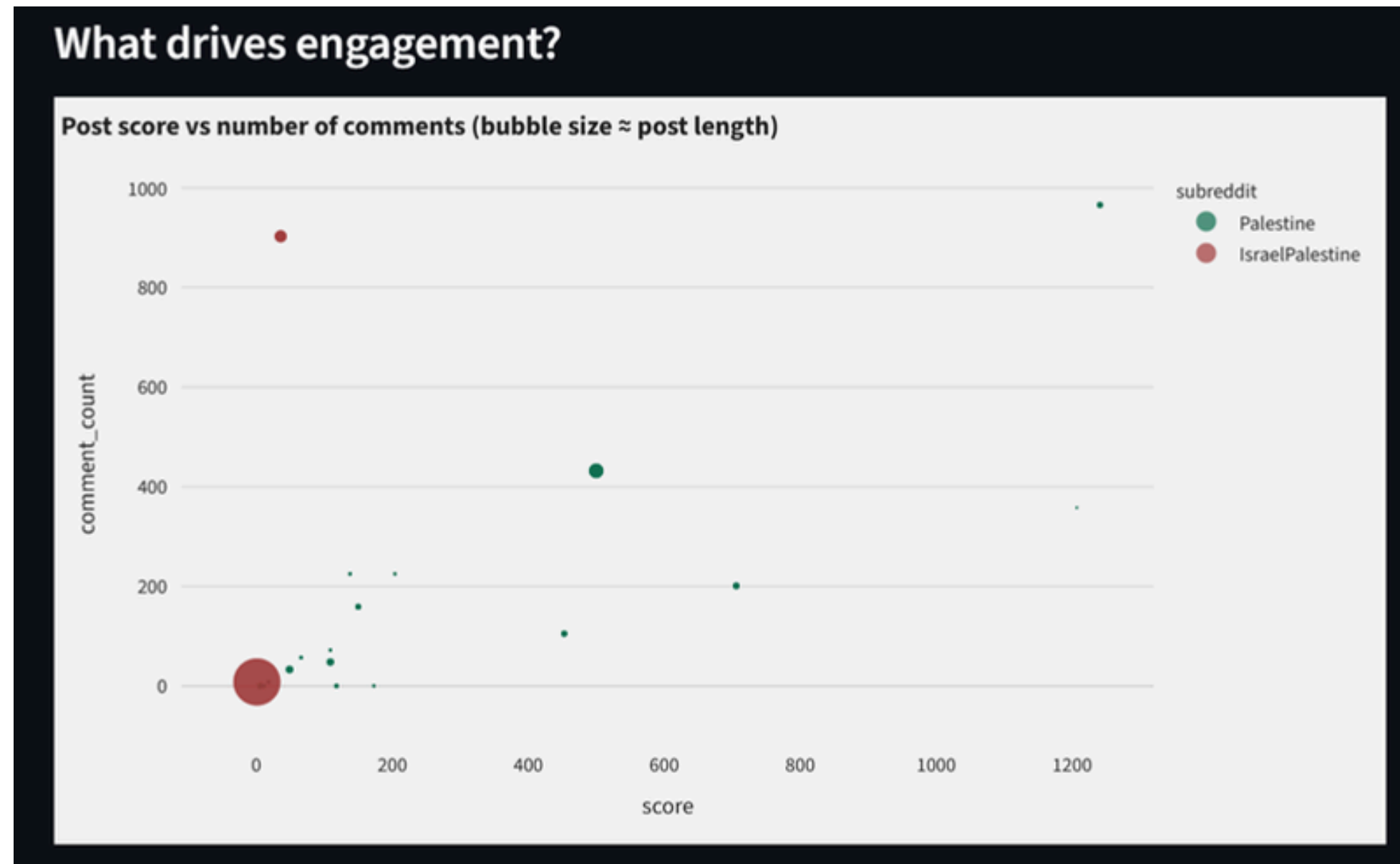
Évolution de l'activité dans le temps



- Ce visuel montre l'évolution du nombre de posts et de commentaires par jour
- Le graphique de gauche représente les posts publiés
- Le graphique de droite représente les commentaires générés
- On observe que l'augmentation des posts entraîne une hausse plus importante des commentaires

# Phase 4: Dashboards

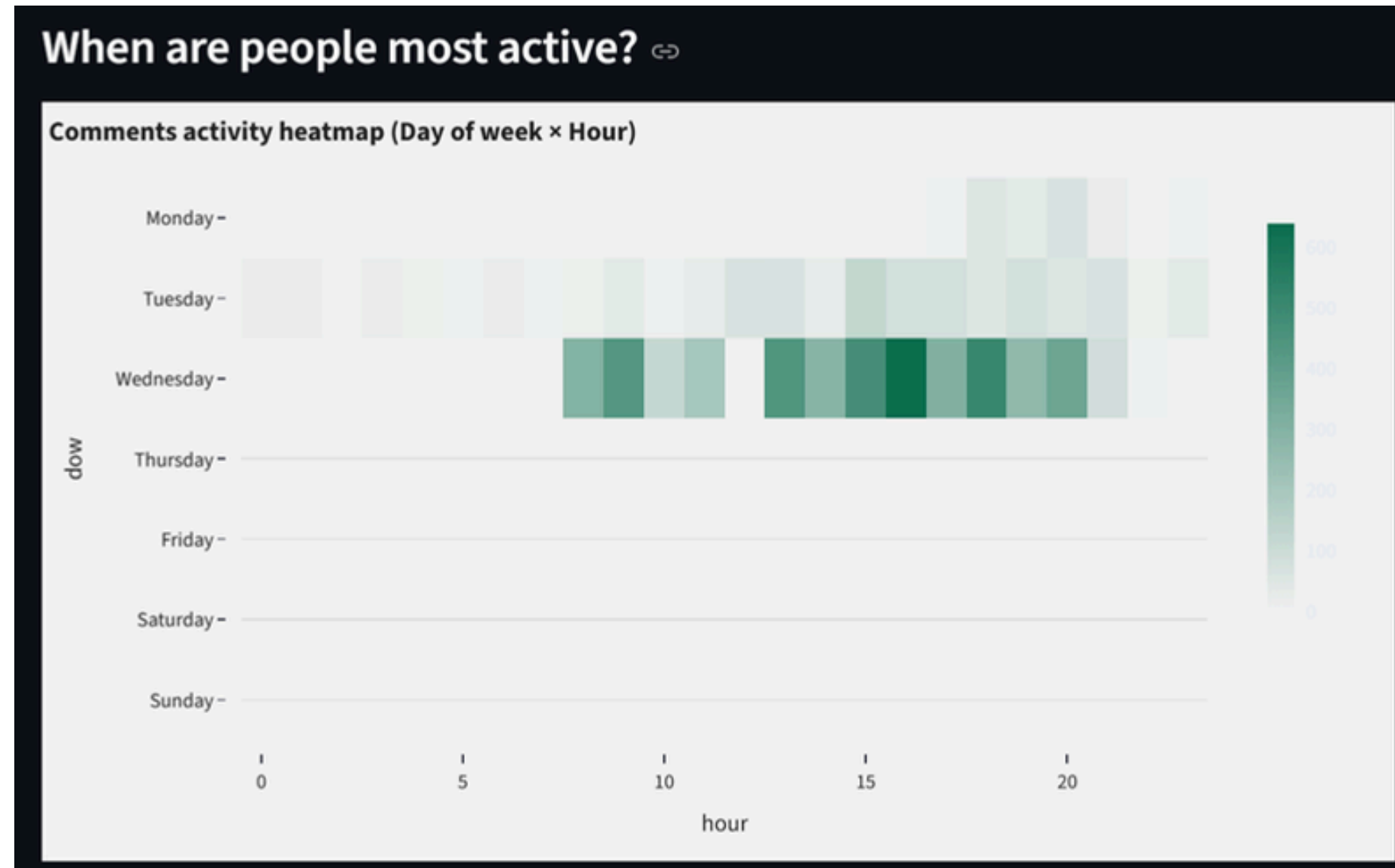
## Facteurs influençant l'engagement



- Ce visuel analyse la relation entre le score d'un post et le nombre de commentaires
- Chaque point représente un post
- La taille des bulles correspond à la longueur du post
- Il permet d'observer si les posts populaires génèrent plus d'engagement

# Phase 4: Dashboards

## Moments de forte activité des utilisateurs



- Ce visuel montre quand les utilisateurs commentent le plus, selon le jour de la semaine et l'heure.
- Les couleurs foncées indiquent des périodes de forte activité.
- On observe que l'activité est surtout concentrée en milieu de semaine, principalement entre la fin de matinée et l'après-midi.

## Phase 4: Dashboards

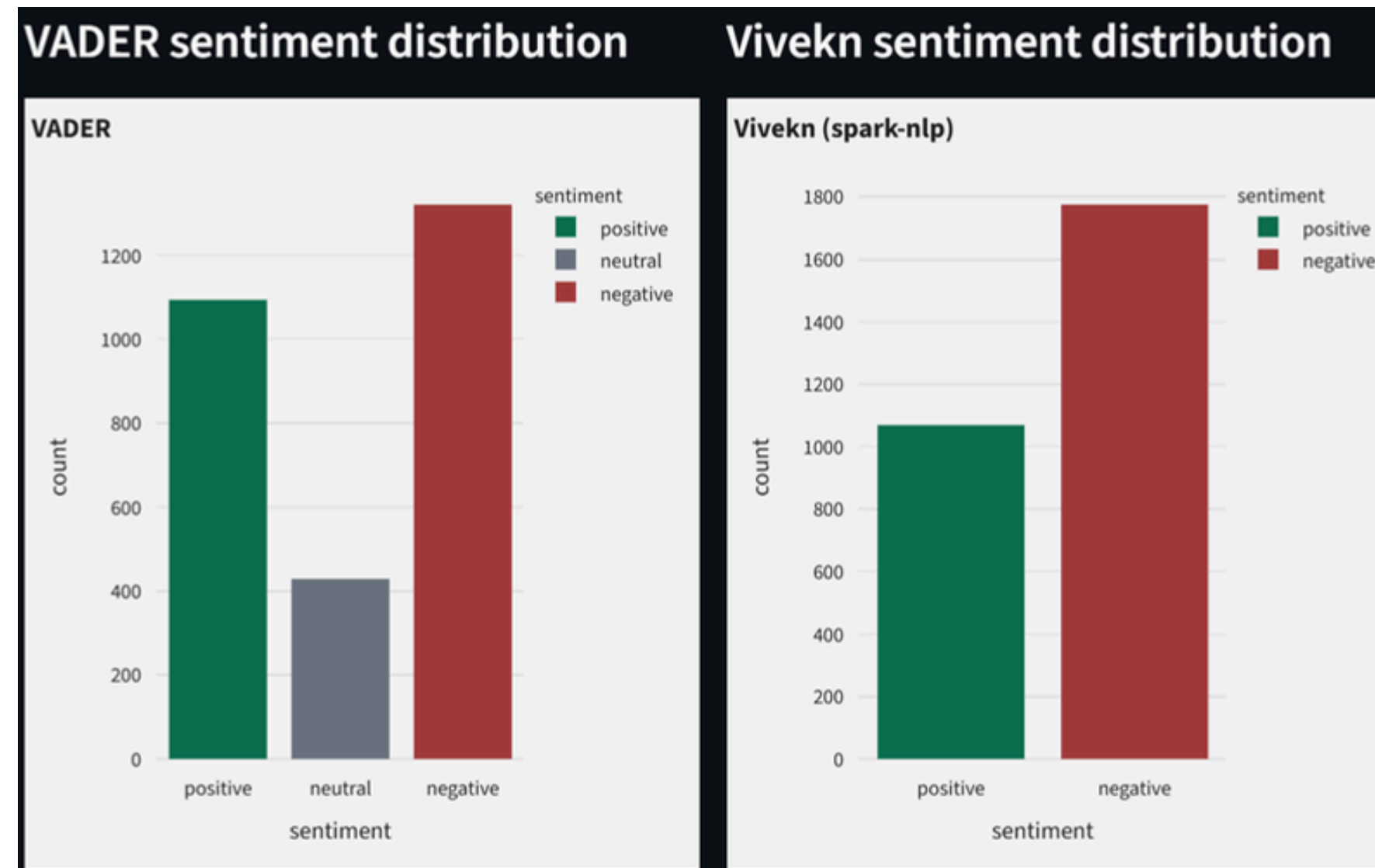
### Dashboard 2: Analyse des sentiments des commentaires



- **Analyzed comments** : nombre total de commentaires analysés.
- **Avg VADER confidence** : niveau moyen de confiance du modèle VADER dans ses prédictions.
- **Avg TextBlob polarity** : polarité moyenne des commentaires (plutôt négative, neutre ou positive).
- **VADER ↔ Vivekn agreement** : taux d'accord entre les deux modèles de sentiment.

# Phase 4: Dashboards

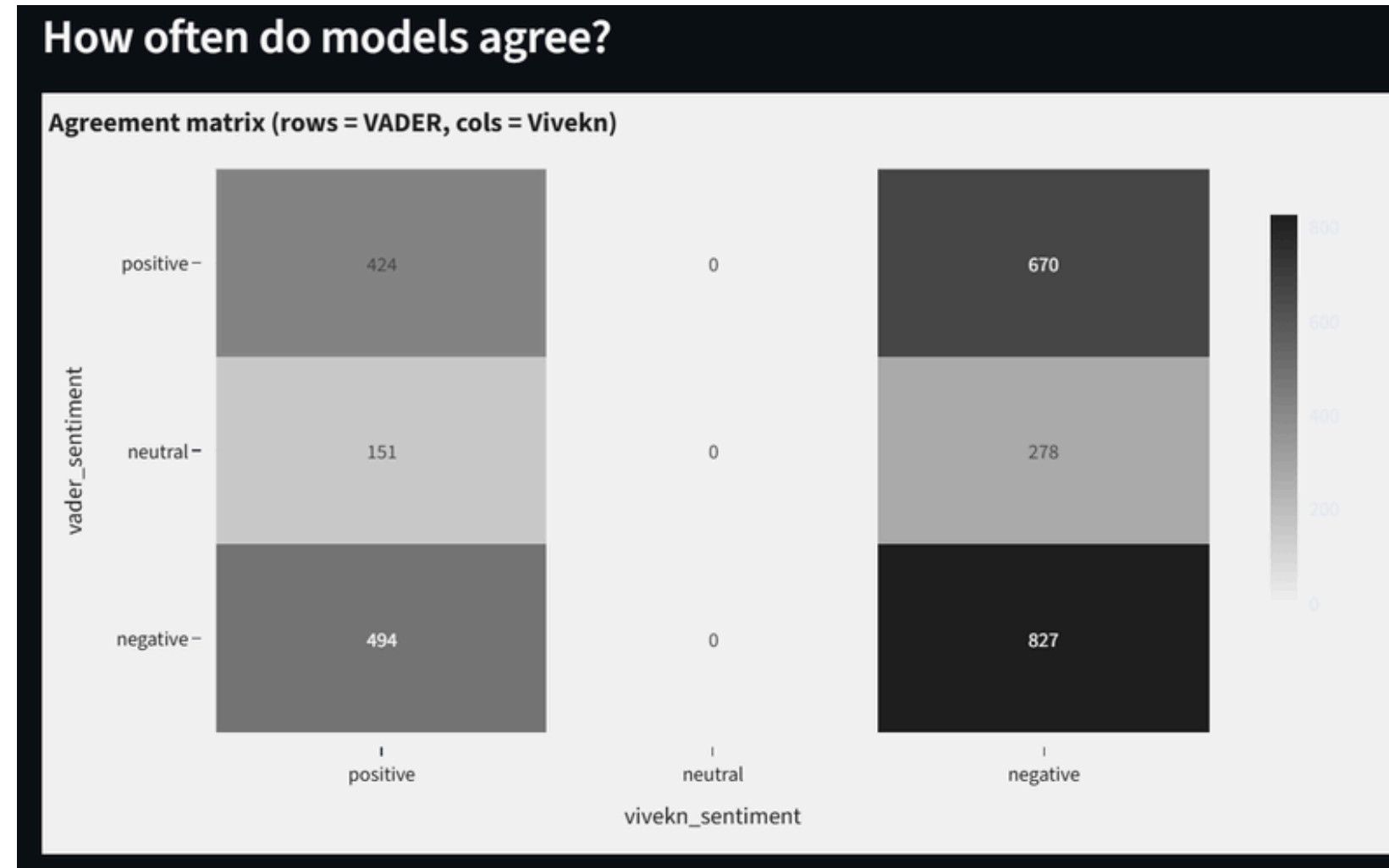
Répartition des sentiments selon les modèles VADER et Vivekn



- Ce visuel montre la distribution des sentiments détectés dans les commentaires.
- VADER distingue trois catégories : positif, neutre et négatif.
- Vivekn (Spark NLP) classe les commentaires uniquement en positif ou négatif.
- Les deux modèles indiquent une dominance globale des sentiments négatifs, malgré une présence significative de commentaires positifs.

# Phase 4: Dashboards

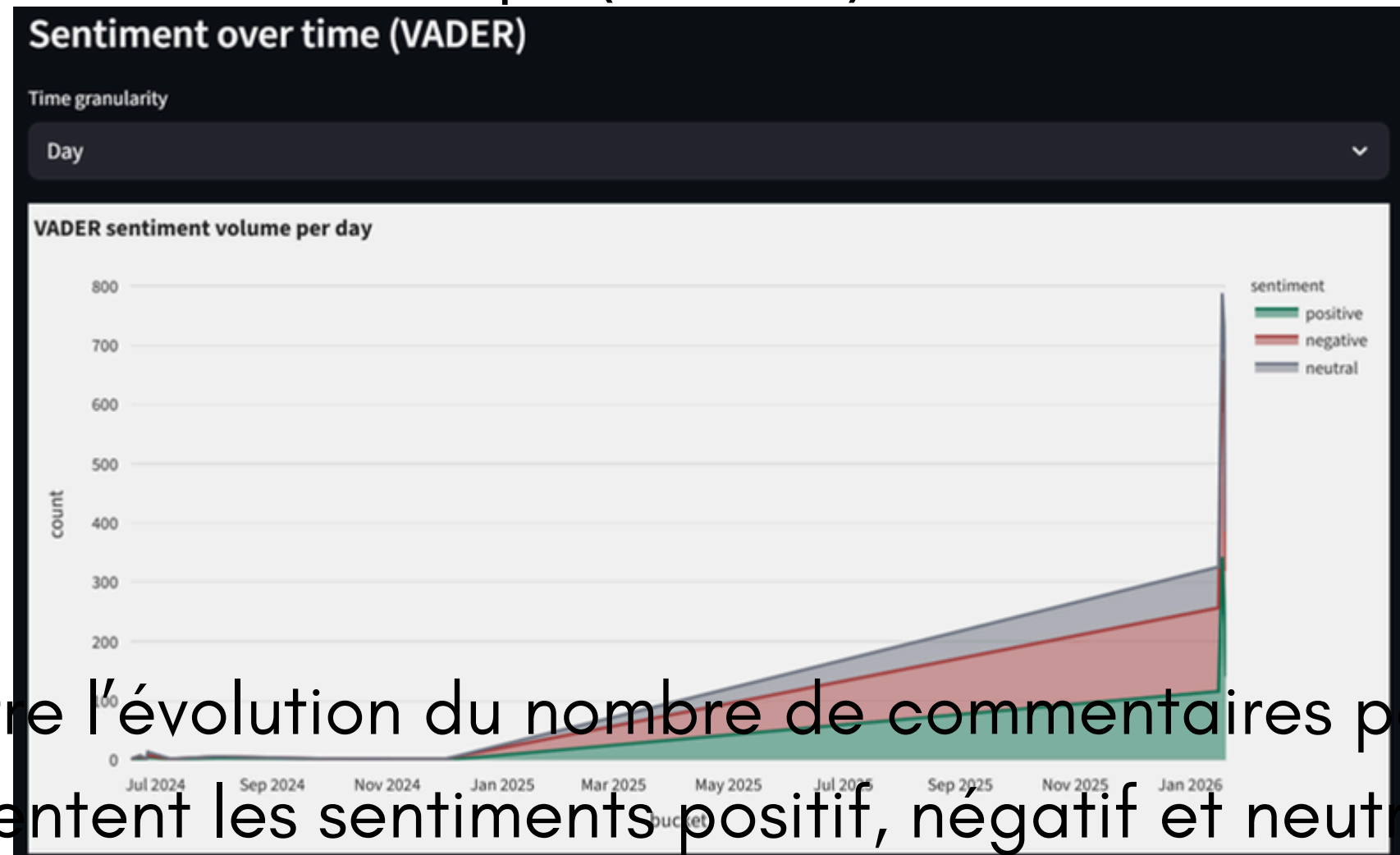
## Accord entre les modèles de sentiment (VADER vs Vivekn)



- Ce visuel montre à quelle fréquence les deux modèles donnent le même sentiment pour un commentaire.
- Les lignes représentent les prédictions de VADER, les colonnes celles de Vivekn.
- Les cases foncées indiquent un fort accord entre les deux modèles.
- On observe que l'accord est partiel, surtout pour les sentiments négatifs, ce qui montre que les modèles n'analysent pas toujours le texte de la même manière.

# Phase 4: Dashboards

## Évolution du sentiment dans le temps (VADER)



- Ce graphique montre l'évolution du nombre de commentaires par sentiment au fil du temps.
- Les couleurs représentent les sentiments positif, négatif et neutre détectés par VADER.
- On observe une augmentation progressive de l'activité, avec une forte hausse vers la fin de la période.
- Les sentiments négatifs et neutres sont majoritaires, ce qui reflète un contexte de discussion très polarisé.



## *8. Conclusion et Perspectives*



## Conclusion

- Collecte des données Reddit (posts et commentaires)
- Nettoyage et préparation des données
- Stockage dans MongoDB et fichiers JSON
- Analyse de sentiment (VADER, Spark NLP, TextBlob)
- Visualisation des résultats avec Streamlit



## *Perspectives*

- Intégrer plus de subreddits et une période plus longue
- Ajouter une analyse multilingue (arabe, français)
- Comparer d'autres modèles de sentiment

The left side of the slide features a decorative graphic composed of several overlapping triangular and quadrilateral shapes in various shades of purple, ranging from a deep indigo to a very light lavender. These shapes create a dynamic, layered effect that tapers towards the right.

*Merci pour  
votre attention!*