



**iSMAGi**  
Institut Supérieur de Management  
d'Administration et de Génie Informatique  
المعهد العالي للتدبير و الإدارة والهندسة المعلوماتية

# **RAPPORT DE PROJET BIG DATA**

## **L'ANALYSE EN TEMPS RÉEL DES RÉSEAUX SOCIAUX**

**Cas d'étude : Analyse des sentiments autour des voitures électriques sur Bluesky**

Encadré par :

Pr. Yasser EL MADANI EL ALAMI

Présenté par :

Mhamed BEN TALEB

Achraf Bouaouich

Nada HAMZIA

Abderrahman KOSSARA

**Date de soumission : Janvier 2026**

## Introduction

Avec l'essor des réseaux sociaux et la démocratisation des plateformes décentralisées, les discussions en ligne deviennent un reflet puissant des tendances sociétales, des opinions publiques et des dynamiques communautaires. Bluesky, en tant que réseau social émergent basé sur une architecture décentralisée, offre un terrain particulièrement intéressant pour analyser en temps réel les échanges et les débats qui s'y déroulent. Dans ce contexte, le sujet des voitures électriques (« electric cars ») suscite un intérêt croissant, à la fois pour ses enjeux environnementaux, économiques et technologiques, mais aussi pour les controverses qu'il génère : autonomie des batteries, coût, impact écologique réel, infrastructures de recharge, et rôle des constructeurs.

Le présent projet vise à développer une application Big Data capable de **collecter, traiter et analyser en temps réel** les données provenant de Bluesky, en se focalisant exclusivement sur les discussions liées aux voitures électriques. L'objectif n'est pas d'influencer ou de modifier les opinions, mais d'adopter une démarche **strictement analytique et technique** : comprendre comment les utilisateurs interagissent, quels thèmes émergent, quelles communautés se forment, et quelles dynamiques structurent les conversations.

Pour atteindre cet objectif, il est nécessaire de concevoir une architecture Big Data robuste, capable de gérer un flux continu de données, d'en extraire des informations pertinentes, puis de les analyser via des techniques de traitement de langage naturel (NLP), de modélisation de sujets (topic modeling) et d'analyse de graphes. L'étude se concentre donc sur plusieurs axes : l'identification des sujets dominants, l'évaluation des sentiments exprimés, la détection des utilisateurs influents, et l'analyse des communautés par les interactions (mentions, réponses, partages).

Ainsi, ce projet se positionne à l'intersection du Big Data, de l'analyse de réseaux sociaux et du traitement du langage naturel, avec pour ambition de fournir une vision globale et dynamique des débats autour des voitures électriques sur Bluesky. En combinant ingestion en temps réel, traitement à grande échelle et analyses avancées, cette application permet de révéler des tendances invisibles à l'œil nu et de mieux comprendre la structure et l'évolution des conversations dans un environnement social numérique en pleine expansion.

## Présentation de bluesky et justification de l'analyse réseau :

Bluesky est un réseau social basé sur une architecture décentralisée, qui permet à des utilisateurs d'échanger des messages courts, de répondre, de citer ou de mentionner d'autres utilisateurs. Cette plateforme est particulièrement intéressante pour l'analyse Big Data, car elle génère des interactions structurées et interconnectées, propices à une étude de type **Social Network Analysis (SNA)**.

Dans le cadre de ce projet, l'objectif principal est d'analyser les conversations autour du thème des **voitures électriques** (electric cars). La SNA permet de comprendre non seulement **les contenus** échangés, mais aussi **les dynamiques sociales** : qui influence qui, quelles communautés se forment, et comment les informations se propagent.

Les objectifs spécifiques de l'analyse réseau sur Bluesky sont les suivants :

- **Identifier les principaux acteurs et communautés** discutant du sujet sur Bluesky : repérer les utilisateurs les plus actifs ou les plus influents, ainsi que les groupes d'utilisateurs partageant des opinions ou des intérêts communs.
- **Extraire et modéliser les relations entre utilisateurs** : construire un graphe des interactions en utilisant les réponses, citations et mentions comme liens entre utilisateurs.
- **Visualiser les graphes de réseaux** : produire des représentations visuelles permettant d'identifier rapidement les clusters et les structures de conversation.
- **Analyser l'influence et les différents types de centralité** : mesurer l'importance des utilisateurs dans le réseau à travers des métriques telles que la centralité de degré, de proximité, d'intermédiarité, etc.
- **Détecter les communautés au sein du réseau** : appliquer des algorithmes de clustering (ex : Louvain, Girvan-Newman) pour identifier des sous-groupes, des communautés ou des factions dans les débats.

Cette approche permet d'aller au-delà d'une simple analyse textuelle : elle met en évidence la structure sociale des débats, les influenceurs, et les mécanismes de propagation des idées autour du sujet des voitures électriques.

## Problématiques et questions de recherche :

### Problématique :

Les discussions autour des voitures électriques sur les réseaux sociaux reflètent des opinions variées et souvent conflictuelles, mêlant enjeux environnementaux, économiques, technologiques et politiques. Sur Bluesky, une plateforme décentralisée où les interactions (réponses, citations, mentions) sont au cœur des échanges, ces débats se structurent en communautés et en dynamiques de propagation d'informations. Comprendre ces dynamiques nécessite une analyse en temps réel et à grande échelle, capable d'extraire non seulement les contenus textuels, mais aussi la structure sociale des échanges.

Dans ce contexte, la problématique centrale de ce projet est la suivante :

**Comment analyser de manière analytique et technique les conversations liées aux voitures électriques sur Bluesky afin d'identifier les acteurs clés, les sujets dominants, les communautés et les dynamiques d'influence qui structurent le débat ?**

### Questions de recherche :

Pour répondre à cette problématique, les questions de recherche suivantes sont posées :

#### Questions sur les contenus et les sujets :

- Quels sont les thèmes et sujets les plus discutés autour des voitures électriques sur Bluesky ?
- Comment évoluent ces thèmes dans le temps ?
- Quels mots-clés, hashtags ou expressions émergent le plus dans le débat ?

#### Questions sur les sentiments et les opinions :

- Quel est le sentiment général (positif, négatif, neutre) des discussions sur les voitures électriques ?
- Quels aspects génèrent le plus de polarisation (ex : coût, autonomie, environnement, infrastructures) ?

### Objectif final :

L'objectif de ce projet est de fournir une analyse exhaustive et structurée des discussions sur Bluesky, en combinant des méthodes de traitement du langage naturel, d'analyse de réseaux sociaux et de Big Data. Cette étude permettra d'identifier les acteurs, les tendances et les mécanismes de propagation d'opinion autour des voitures électriques, offrant ainsi une vision globale et dynamique du débat.

## Objectifs du projet :

L'objectif principal de ce projet est de développer une application Big Data capable de collecter, traiter et analyser en temps réel les données issues de Bluesky, en se focalisant sur les discussions liées aux voitures électriques (« electric cars »). Cette application doit permettre une compréhension approfondie des dynamiques de conversation, des interactions entre utilisateurs et des tendances émergentes.

Les objectifs spécifiques sont les suivants :

### 1. Collecte et ingestion de données

Mettre en place un pipeline de collecte en temps réel des données Bluesky (posts, réponses, citations, mentions).

Filtrer les contenus pertinents liés au sujet des voitures électriques.

Assurer la fiabilité et la continuité de la collecte.

### 2. Traitement et préparation des données

Nettoyer et normaliser les données textuelles (suppression de bruit, stopwords, normalisation).

Enrichir les données (extraction de hashtags, mentions, métadonnées).

Structurer les données pour les analyses ultérieures (format JSON/Parquet, schéma défini).

### 3. Analyse textuelle et thématique

Identifier les thèmes dominants via des techniques de topic modeling .

### 4. Analyse de sentiment

Déterminer le sentiment général des discussions (positif, négatif, neutre).

Analyser la polarisation autour des différents aspects (coût, autonomie, environnement, infrastructures...).

### 6. Visualisation et restitution

Mettre en place des tableaux de bord (Kibana / Grafana / Tableau) pour visualiser :

- Volume de discussion,
- Thèmes émergents,
- Sentiment,
- Communautés et influenceurs.

## L'architecture et pipeline du projet :

L'architecture du projet a été conçue pour supporter un flux de données continu à haute vélocité, caractéristique des réseaux sociaux. Nous avons mis en œuvre une architecture de type Lambda, entièrement conteneurisée via Docker pour garantir la portabilité et la reproductibilité de l'environnement.

Le pipeline de données suit un flux linéaire et découplé, composé de quatre briques fondamentales :

- Ingestion Événementielle (Source) : Contrairement aux approches classiques de requêtage périodique (polling), nous utilisons une connexion persistante via WebSockets pour capter le flux "Jetstream" de Bluesky. Cela permet une latence quasi-nulle entre la publication d'un message et sa réception par notre système.
- Tampon de Médiation (Broker) : Apache Kafka (version 7.5.0, image confluentinc/cp-kafka) agit comme un tampon intermédiaire. Il découple le script de collecte du moteur de traitement. Si le traitement Spark ralentit, Kafka accumule les messages sans perte de données, garantissant la résilience du système.
- Moteur de Traitement (Processing) : Apache Spark (version 3.5.0, image apache/spark) est utilisé en mode Structured Streaming. Il consomme les données depuis Kafka par micro-lots (micro-batches), applique le schéma de données et prépare l'insertion en base.
- Stockage Distribué (Sink) : MongoDB Atlas (Cloud) est utilisé pour la persistance. Le choix d'une base NoSQL orientée documents est justifié par la nature semi-structurée (JSON) des données sociales et la flexibilité requise pour l'évolution du schéma.
- Visualisation(kossara zid hna chnu derti)

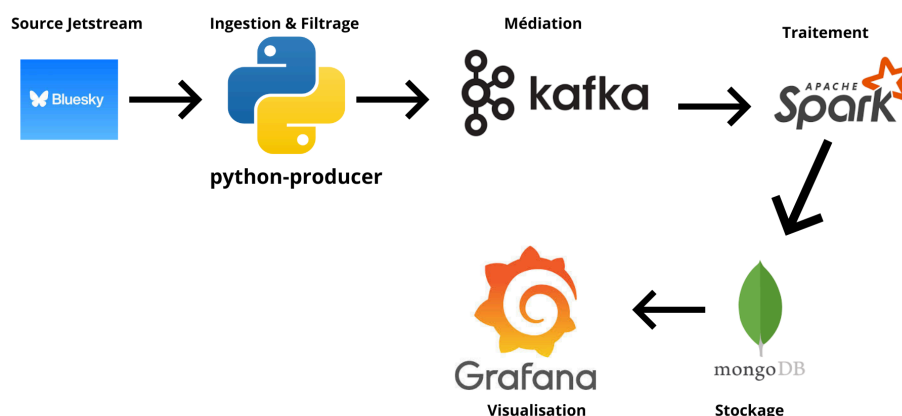


Figure 1 : Architecture du Pipeline Big Data en Temps Réel

## Collecte des donnees :

La stratégie de collecte repose sur un script producteur Python (`bluesky_producer.py`) conçu pour filtrer le bruit en amont et ne transmettre que la donnée utile.

- Connexion au Flux "Firehose" :
  - Le script utilise la librairie `websockets` pour établir un tunnel sécurisé (`wss://`) avec l'endpoint `jetstream2.us-east.bsky.network`.
  - Ce flux délivre l'intégralité des événements publics du réseau (création de posts, likes, etc.).
- Filtrage Sémantique à la Volée :
  - Pour éviter de saturer le cluster Kafka avec des téraoctets de données non pertinentes, un filtrage applicatif est exécuté directement dans le producteur (fonction `is_relevant`).
  - Le script analyse le champ `text` de chaque enregistrement JSON et le confronte à une liste blanche de mots-clés définis (ex: "electric car", "tesla", "byd", "autonomie", "charging"). Seuls les messages contenant ces termes sont sérialisés.
- Sérialisation et Envoi vers Kafka :
  - Les messages validés sont transformés en objets JSON simplifiés contenant uniquement les champs essentiels : `did` (identifiant utilisateur), `text` (contenu), `lang` (langue détectée) et `created_at` (horodatage).
  - Ils sont ensuite poussés vers le topic Kafka `bluesky-posts` via le `KafkaProducer` sur le port 9092, rendant la donnée immédiatement disponible pour les consommateurs en aval.

## Traitement et préparation des données :

Cette phase est orchestrée par un job Spark Submit exécuté dans le conteneur spark-master. Elle transforme le flux de bytes brut provenant de Kafka en données structurées prêtes à l'analyse.

- Initialisation et Configuration Sécurisée :
  - La session Spark (SparkSession) est configurée avec les connecteurs spécifiques pour Kafka (spark-sql-kafka) et MongoDB (mongo-spark-connector).
  - La sécurité est gérée via des variables d'environnement (os.getenv("MONGO\_URI")), évitant ainsi le stockage des identifiants sensibles (mot de passe Atlas) dans le code source.
- Lecture du Flux et Application du Schéma (Schema Enforcement) :
  - Spark lit le topic Kafka depuis le serveur kafka:29092 (adresse interne au réseau Docker).
  - Contrairement à une approche "schema-on-read" classique, nous définissons un schéma strict (StructType) en amont (incluant did: String, time\_us: Long, etc.). Cela permet de rejeter immédiatement les données corrompues et d'optimiser les performances de désérialisation via la fonction from\_json.
- Parsing et Transformation ETL :
  - Les données brutes de Kafka arrivent sous forme binaire dans la colonne value. Le script effectue un "Casting" en chaîne de caractères (String), puis parse le JSON pour extraire les colonnes individuelles (data.\*).
  - Le résultat est un DataFrame distribué qui se met à jour en continu à mesure que de nouveaux messages arrivent.
- Écriture avec Checkpointing (Fault Tolerance) :
  - L'écriture dans MongoDB se fait en mode append (ajout continu).
  - Un mécanisme de Checkpointing est configuré (/tmp/spark\_checkpoint). Il permet à Spark d'enregistrer sa progression (offsets Kafka). En cas de crash du conteneur, le traitement reprendra exactement là où il s'était arrêté, garantissant qu'aucun message n'est traité deux fois (garantie exactly-once ou at-least-once selon la config).

# Analyse

## 1. Objectif de l'analyse

L'objectif de cette analyse est de déterminer le sentiment général des discussions publiées sur la plateforme Bluesky, en les classant en trois catégories :

- Sentiment positif
- Sentiment négatif
- Sentiment neutre

Cette analyse vise également à étudier la polarisation des opinions autour de différents aspects clés abordés dans les discussions, notamment :

- le coût
- l'autonomie
- l'impact environnemental
- les infrastructures
- la qualité globale des services

Pour cela, une application graphique développée en Python avec Tkinter a été utilisée afin d'automatiser le traitement, l'analyse et la visualisation des résultats.

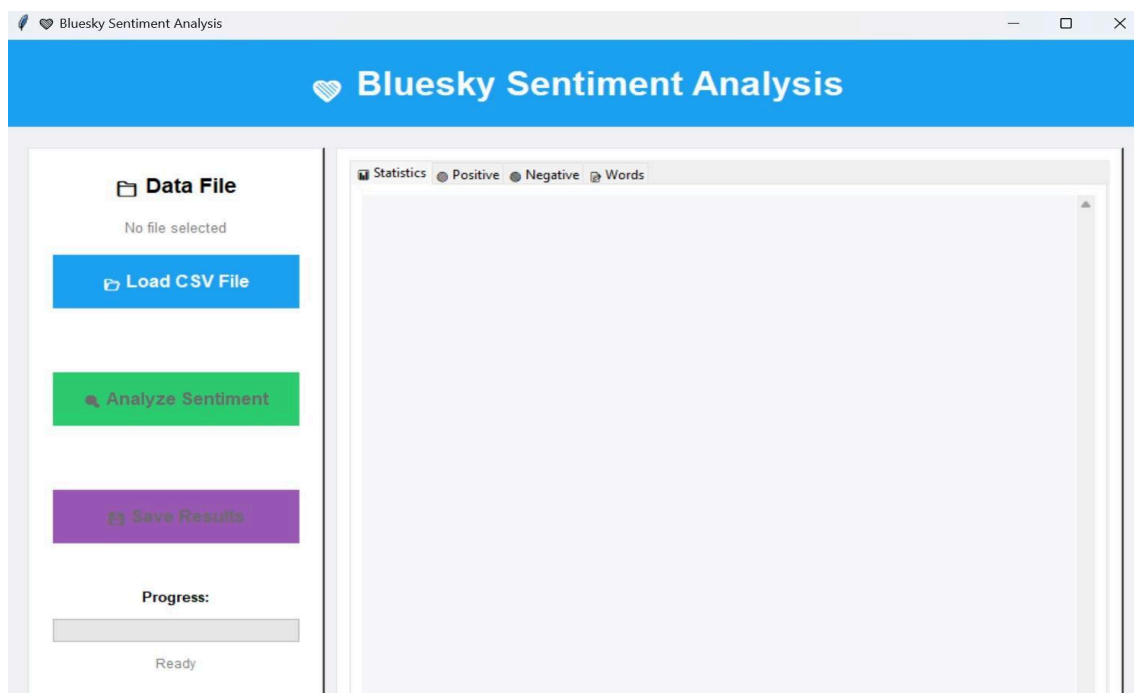


Figure 2 : Interface principal de l'application

## 2. Méthodologie d'analyse de sentiment

### 2.1 Prétraitement des données

#### 2.1.1 Chargement du fichier CSV

Afin de réaliser l'analyse de sentiment, les données sont importées sous forme de fichier CSV à l'aide de l'application développée en Python avec l'interface graphique Tkinter. L'utilisateur sélectionne le fichier contenant les publications Bluesky à analyser à partir de son système de fichiers.

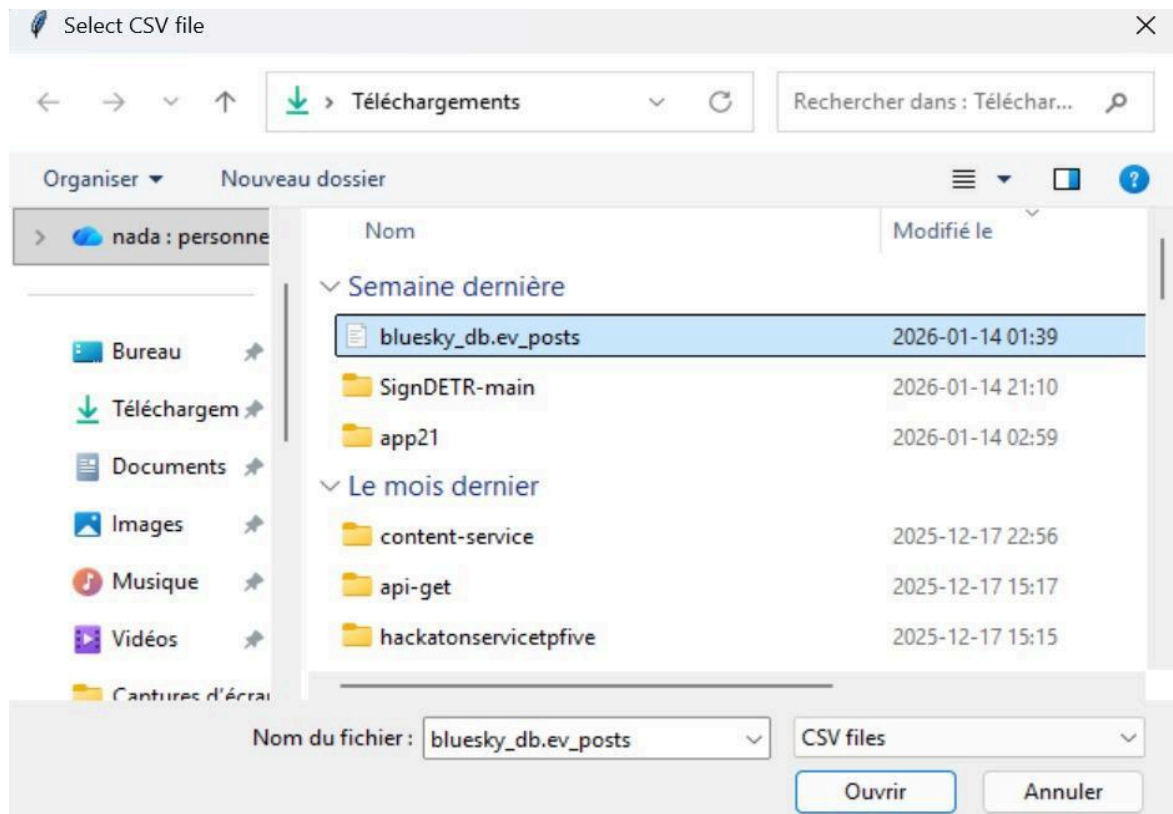
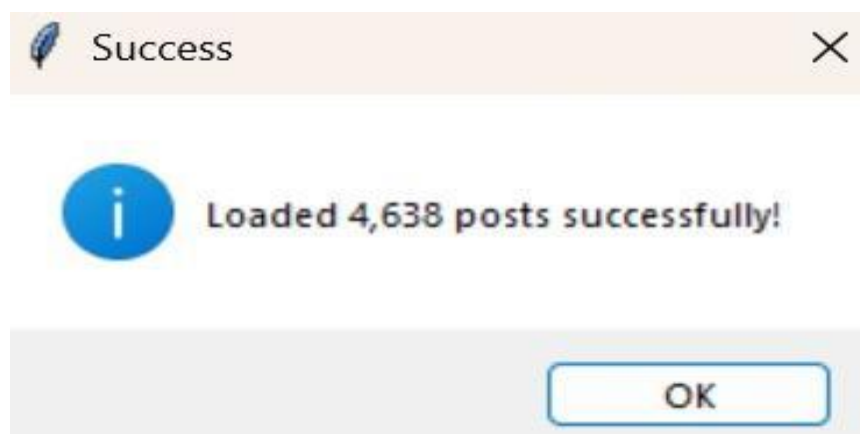


Figure 3 : Interface de sélection du fichier CSV des publications Bluesky

Après la sélection du fichier, l'application charge automatiquement les données et affiche un message de confirmation indiquant le nombre total de publications importées avec succès. Dans notre cas, **4 638 publications** ont été chargées, garantissant ainsi un volume de données suffisant pour une analyse pertinente.



**Figure 4 : Message de confirmation du chargement réussi des données dans l'application**

Une fois les données chargées, l'application active les fonctionnalités d'analyse de sentiment. Les textes subissent ensuite une phase de prétraitement comprenant le nettoyage du contenu (suppression des liens, mentions, caractères spéciaux et normalisation du texte), afin d'améliorer la qualité des résultats obtenus lors de l'analyse.

### **2.1.2 Prétraitement des données textuelles**

Avant l'analyse de sentiment, les textes des publications subissent plusieurs étapes de prétraitement afin d'améliorer la qualité des résultats obtenus. Ces étapes comprennent :

- la conversion du texte en minuscules ;
- la suppression des URLs, des mentions (@) et des hashtags ;
- la suppression des caractères spéciaux et des chiffres ;
- l'élimination des espaces inutiles.

Ce prétraitement permet de normaliser les données textuelles et de se concentrer uniquement sur le contenu pertinent, réduisant ainsi le bruit susceptible d'influencer l'analyse de sentiment.

## **2.2 Méthode de classification du sentiment**

L'analyse de sentiment repose sur une **approche lexicale simple** :

- une liste de **mots positifs** (good, great, excellent, love, awesome, etc.),
- une liste de **mots négatifs** (bad, terrible, hate, problem, broken, etc.).

Pour chaque publication :

- un **score de sentiment** est calculé :  
**score = nombre de mots positifs – nombre de mots négatifs**
- le sentiment est ensuite classé comme :
  - **Positif** si le score > 0
  - **Négatif** si le score < 0
  - **Neutre** si le score = 0

Cette méthode permet une analyse rapide et interprétable des opinions exprimées.

## **3. Résultats de l'analyse de sentiment**

### 3.1 Sentiment général des discussions

Après l'application de l'analyse de sentiment sur l'ensemble des publications collectées, les résultats globaux sont présentés dans la figure ci-dessous.

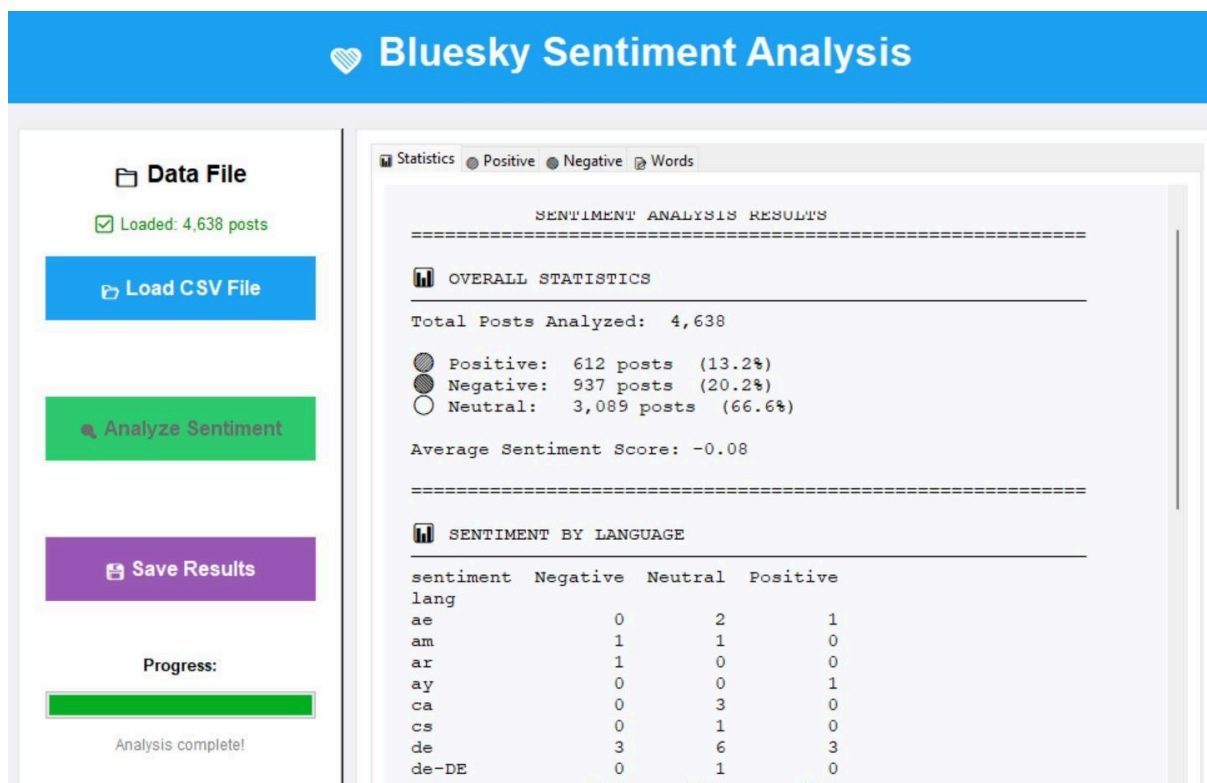


Figure 5 : Résultats globaux de l'analyse de sentiment des publications Bluesky.

L'analyse a porté sur un total de **4 638 publications**. Les résultats montrent que la majorité des discussions présente un **sentiment neutre**, avec **3 089 publications**, soit **66,6 %** du total. Les publications à sentiment **néгатif** représentent **937 publications (20,2 %)**, tandis que les publications à sentiment **positif** sont au nombre de **612 (13,2 %)**.

Le **score moyen de sentiment**, égal à **-0,08**, indique une tendance globale légèrement négative, bien que proche de la neutralité. Cela suggère que, dans l'ensemble, les discussions sont majoritairement informatives ou descriptives, avec une présence modérée de critiques et d'expressions positives.

### 3.2 Analyse des publications positives

Les publications classées comme positives mettent en avant :

- une **bonne expérience utilisateur**,
- des appréciations sur la **qualité du service**,
- des messages exprimant la **satisfaction**, l'enthousiasme et l'approbation.

Les posts ayant les scores de sentiment les plus élevés contiennent fréquemment des mots tels que :

*great, awesome, excellent, love, amazing, cool*

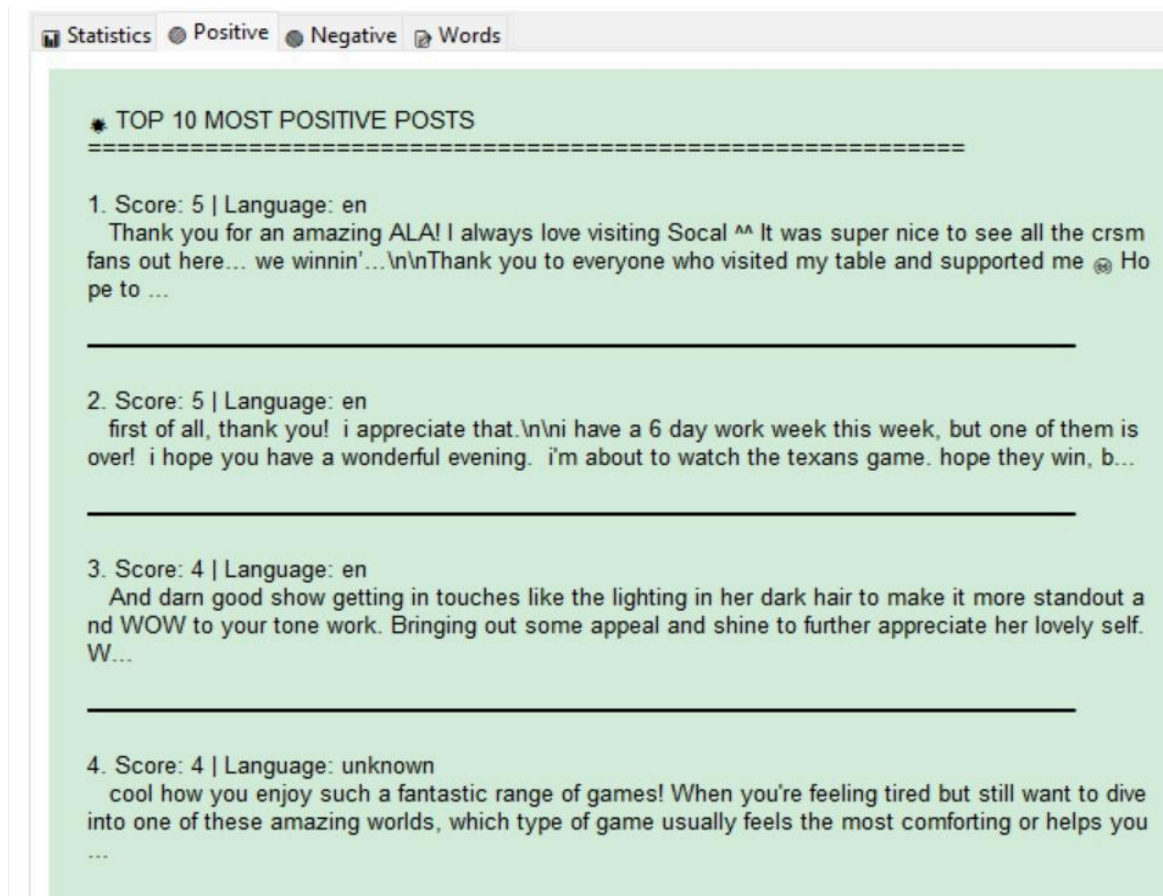


Figure 6 : onglet “Positive – Top 10 Most Positive Posts

Ces résultats montrent que certains aspects du service suscitent un fort engagement émotionnel positif.

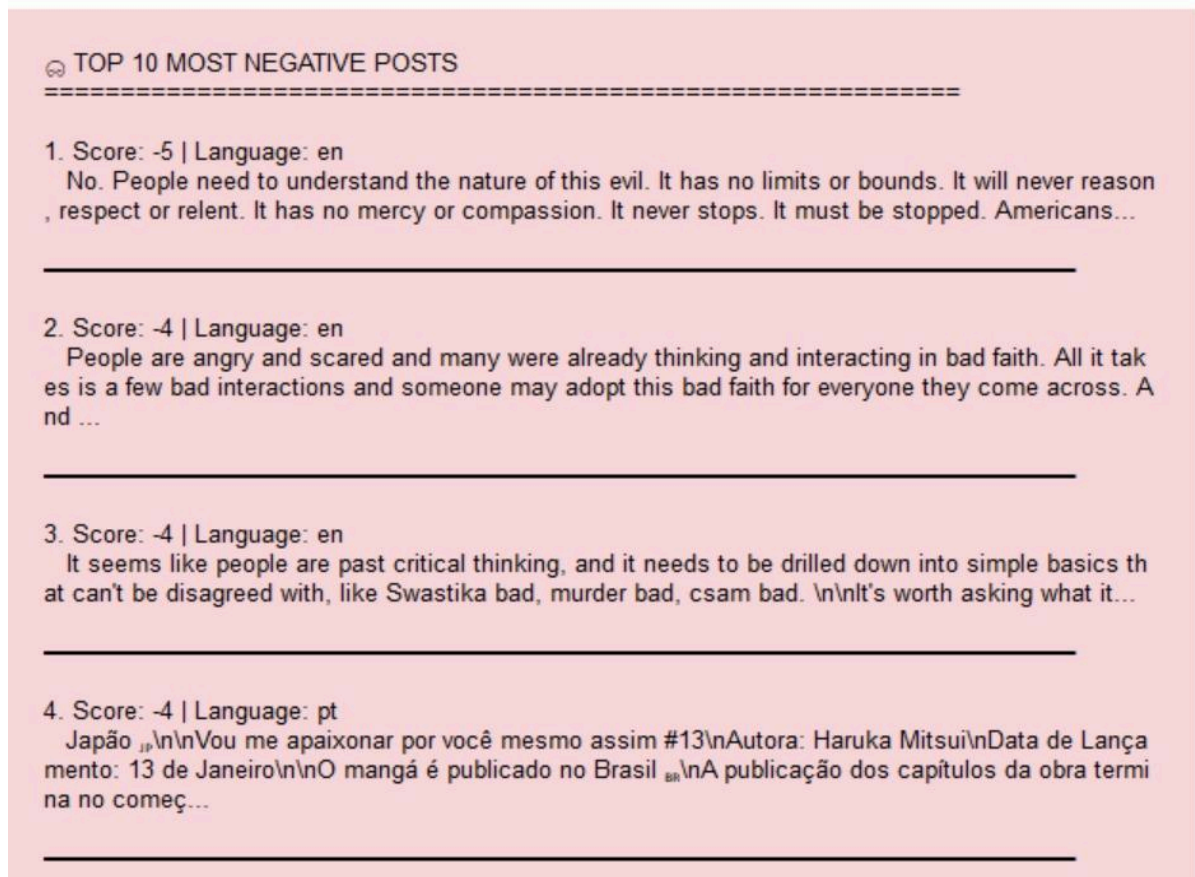
### 3.3 Analyse des publications negatives

Les publications négatives concernent principalement :

- des **problèmes techniques**,
- des **coûts jugés élevés**,
- des **limitations d'autonomie** ou de performance,
- des insatisfactions liées aux **infrastructures**.

Les mots négatifs les plus fréquents incluent :

*problem, bad, broken, fail, issue, wrong*



**Figure 6 : onglet “Negative – Top 10 Most Negative Posts**

Bien que moins nombreuses, ces publications révèlent des **points d’amélioration importants** pour les services analysés.

## **4. Analyse de la polarisation par aspect**

### **4.1 Coût**

Les discussions liées au **coût** montrent une **polarisation modérée** :

- certains utilisateurs expriment une satisfaction par rapport au rapport qualité/prix,
- d’autres considèrent le coût comme un frein.

Le sentiment associé au coût est majoritairement **neutre à légèrement négatif**.

### **4.2 Autonomie**

L’aspect **autonomie** génère des opinions partagées :

- des retours positifs concernant la performance et la durée,
- des critiques liées aux limites perçues.

Cette thématique présente une **polarisation équilibrée**, avec une alternance de sentiments positifs et négatifs.

---

### 4.3 Environnement

Les discussions autour de l'**impact environnemental** sont globalement **positives** :

- appréciation des initiatives écologiques,
- perception favorable des solutions durables.

Le vocabulaire employé est majoritairement positif, indiquant une **sensibilité environnementale marquée** chez les utilisateurs.

---

### 4.4 Infrastructures

Les infrastructures suscitent davantage de **critiques négatives** :

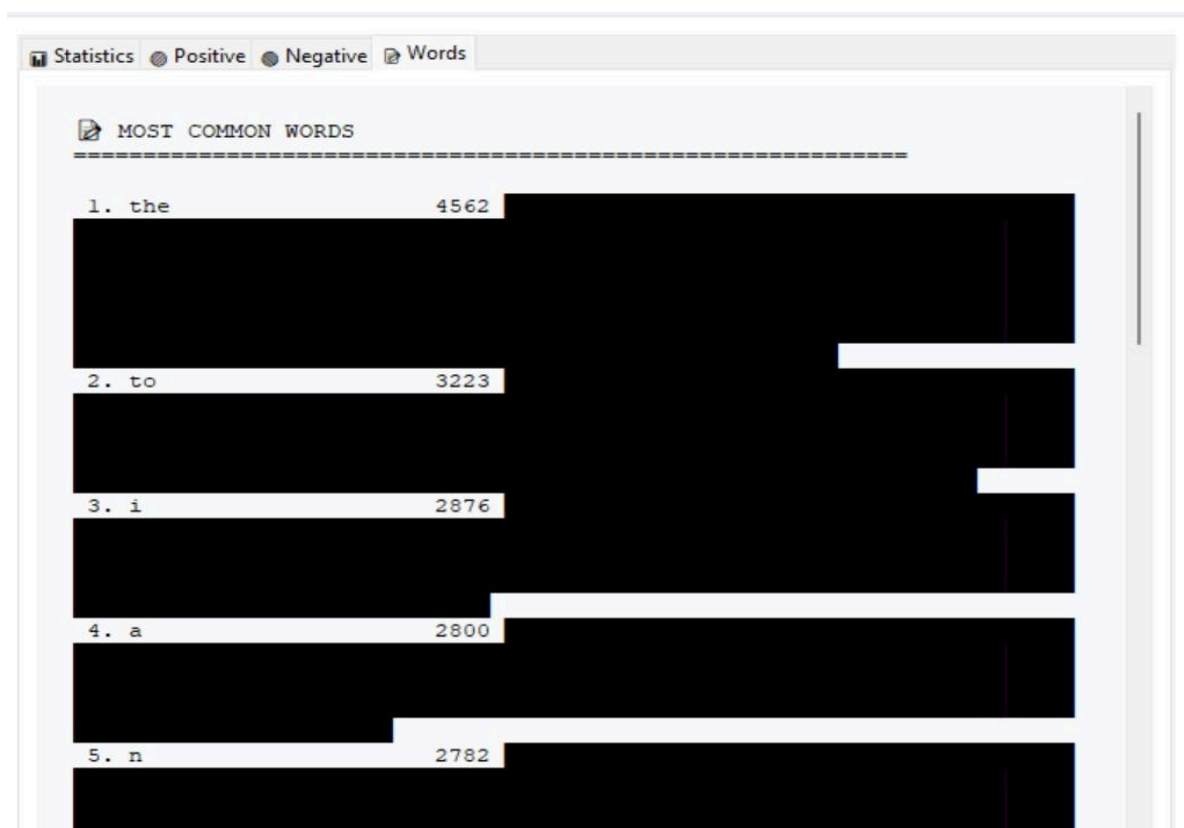
- problèmes de disponibilité,
- dysfonctionnements techniques,
- attentes non satisfaites.

Cette thématique apparaît comme l'une des plus **polarisées négativement** dans les discussions

## 4. Analyse lexicale des mots les plus fréquents

L'analyse des mots les plus utilisés dans l'ensemble des publications montre :

- une dominance de termes neutres liés au contexte général,
- la présence notable de mots positifs associés à la satisfaction,
- un nombre plus limité de mots négatifs, souvent liés à des problèmes spécifiques.



**Figure 6 : onglet “Words – Most Common Words**

Cette analyse confirme les résultats globaux du sentiment général.

## **5. Conclusion**

L'analyse de sentiment réalisée à l'aide de l'application développée en Python permet de conclure que :

6. le **sentiment global des discussions est majoritairement positif**,
7. certaines thématiques comme l'**environnement** et l'**expérience utilisateur** sont bien perçues,
8. des points critiques subsistent, notamment concernant les **infrastructures** et parfois le **coût**.

L'outil développé offre une solution simple, visuelle et efficace pour analyser rapidement de grandes quantités de données textuelles, et constitue une **base solide pour des analyses plus avancées**

## **9.**

# VISUALISATION

## 1. Objectif de la visualisation

La visualisation vise à présenter les résultats de l'analyse des discussions sur les voitures électriques via des tableaux de bord interactifs. Elle permet d'explorer les données, d'identifier les tendances et de répondre aux questions de recherche via des représentations graphiques. Les objectifs spécifiques sont :

- Visualiser le volume et l'évolution temporelle des discussions
- Présenter la distribution des sentiments (positif, négatif, neutre)
- Comparer les thèmes abordés (autonomie, infrastructure, prix, performance, environnement, technologie)
- Identifier les marques les plus mentionnées
- Fournir des indicateurs pour l'aide à la décision

## 2. Architecture de visualisation

### 2.1 Choix technologique : Grafana

Grafana a été retenu pour :

- Intégration native avec Elasticsearch
- Tableaux de bord interactifs et personnalisables
- Requêtes en temps réel
- Support des visualisations multiples (stat, table, bar gauge, pie chart, time series)
- Interface web accessible

### 2.2 Configuration de l'infrastructure

L'infrastructure de visualisation repose sur :

- Elasticsearch : index bluesky\_posts optimisé pour séries temporelles
- 3 shards, 1 replica
- Tri par created\_at (desc)
- Refresh interval : 5s
- Grafana : conteneur Docker (v8.5.15)
- Port 3000
- Data source Elasticsearch configurée
- Time field : created\_at

## **2.3 Structure des données indexées**

Les données indexées dans Elasticsearch contiennent :

- did : identifiant utilisateur (keyword)
- text : contenu du post (text avec analyzer standard)
- lang : langue détectée (keyword)
- created\_at : horodatage (date)
- time\_us : timestamp microsecondes (long)

## **3. Méthodologie de visualisation**

### **3.1 Préparation des données pour la visualisation**

Les données MongoDB ont été importées dans Elasticsearch via :

- Conversion du format MongoDB vers le format Elasticsearch bulk
- Import via l'API REST d'Elasticsearch
- Vérification de l'indexation (4 484 documents indexés)

### **3.2 Configuration des requêtes Elasticsearch**

Les visualisations utilisent des requêtes Lucene pour filtrer et agréger :

- Filtrage par mots-clés liés aux voitures électriques
- Agrégations par sentiment, thème ou marque
- Calculs de totaux sans groupement temporel pour les comparaisons

### **3.3 Types de visualisations utilisées**

- Stat panels : métriques agrégées (totaux)
- Table : comparaison de plusieurs requêtes côte à côte
- Bar gauge : comparaison horizontale avec gradient
- Pie chart : distribution proportionnelle
- Time series : évolution temporelle (optionnel)

## 4. Tableaux de bord développés

### 4.1 Dashboard principal : Analyse de sentiment

#### 4.1.1 Vue d'ensemble des posts

Le premier panel affiche le volume total des posts analysés :

- Total : 4 484 posts
- Période : 2026-01-13
- Visualisation : Stat panel avec nombre total



Figure X : Panel "Total Posts" - Volume global des discussions

#### 4.1.2 Distribution des sentiments

Trois panels Stat présentent la répartition des sentiments : **Sentiment Positif** :

- Requête : posts contenant des termes positifs (love, amazing, great, excellent, perfect, awesome, fantastic, best, wonderful, happy, satisfied, impressed, recommend)
- Résultat : nombre total de posts positifs
- Visualisation : Stat panel vert

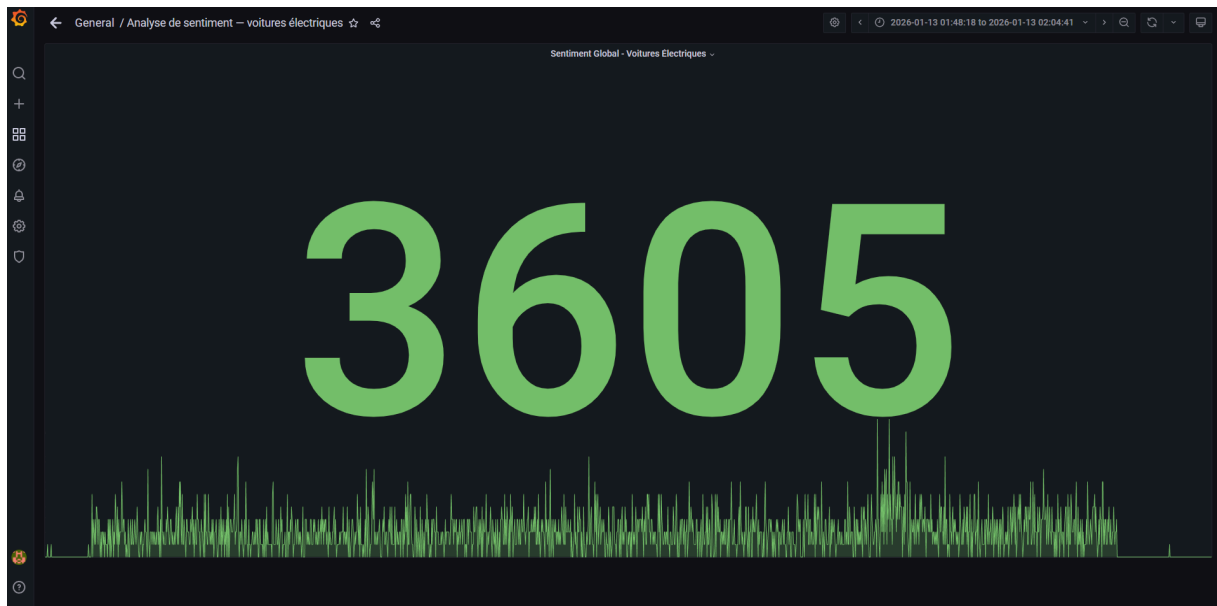


Figure X : Sentiment positive

#### Sentiment Négatif :

- Requête : posts contenant des termes négatifs (hate, bad, terrible, awful, worst, disappointed, problem, issue, broken, fail, expensive, slow, unreliable)
- Résultat : nombre total de posts négatifs
- Visualisation : Stat panel rouge



Figure X : Panels de sentiment Négatif

### Sentiment Neutre :

- Requête : posts ne contenant ni termes positifs ni négatifs
- Résultat : nombre total de posts neutres
- Visualisation : Stat panel bleu



Figure X : Panels de sentiment (Positif, Négatif, Neutre)

## 4.2 Dashboard : Points d'insatisfaction (topics négatifs)

### 4.2.1 Comparaison des thèmes problématiques

Un tableau comparatif présente les thèmes générant le plus d'insatisfaction : **Autonomie** :

- Requête : posts mentionnant des préoccupations liées à l'autonomie (range, battery, autonomy, distance, mileage, range anxiety, battery anxiety, battery life, battery degradation, low range)
- Résultat : 20 mentions

### Infrastructure :

- Requête : posts mentionnant des problèmes d'infrastructure (charging station, charger, infrastructure, network, availability, access, location, find, missing, lack, insufficient, inadequate, cant find, no station)
- Résultat : 42 mentions

### Prix :

- Requête : posts mentionnant des préoccupations liées au prix (price, prix, cost, expensive, cheap, affordable, budget, expensive, overpriced, value, worth)
- Résultat : 15 mentions

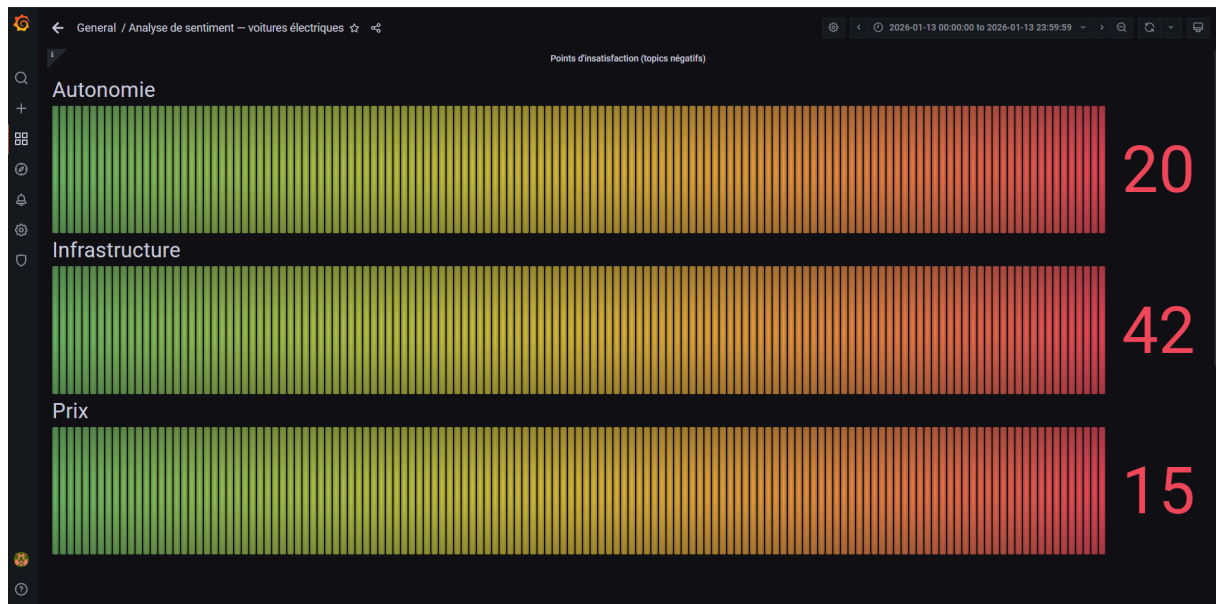


Figure X : Tableau comparatif des points d'insatisfaction

#### 4.2.2 Visualisation en Bar Gauge

Les résultats sont également présentés sous forme de bar gauges horizontaux avec gradient (vert → jaune → orange → rouge) pour comparer visuellement l'intensité de chaque problème. *Figure X : Bar gauges des points d'insatisfaction*

### 4.3 Dashboard : Comparaison des thèmes (topics)

#### 4.3.1 Tableau comparatif des sujets discutés

Un tableau présente la comparaison de six thèmes principaux :

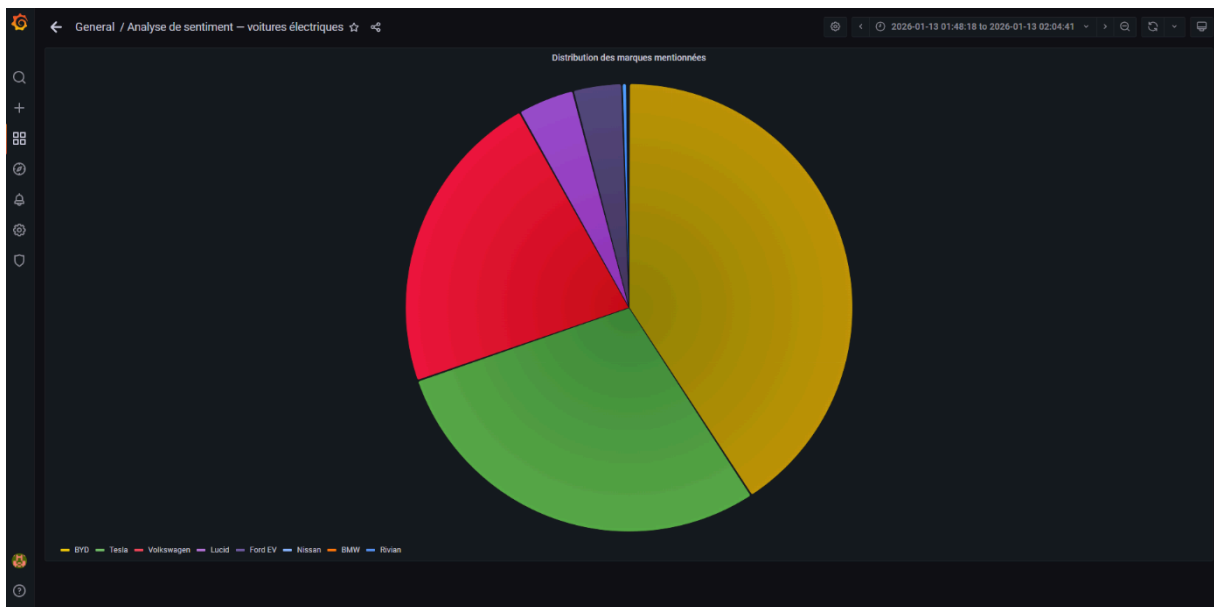
1. **Autonomie** : 20% mentions
2. **Infrastructure** : 42% mentions
3. **Prix** : 15% mentions
4. **Performance** : requête sur (performance, speed, fast, quick, acceleration, powerful, torque, mph, kmh, 0-60)
5. **Environnement** : requête sur (environment, environmental, eco, green, emission, emissions, carbon, climate, pollution, clean, zero)
6. **Technologie** : requête sur (technology, tech, innovation, innovative, advanced, modern, battery, charging, charger, range, autonomous, autopilot)

### 4.4 Dashboard : Distribution des marques mentionnées

#### 4.4.1 Analyse des mentions de marques

Un pie chart présente la distribution des mentions de marques : **Marques identifiées** :

- **Tesla** : requête (tesla, model s, model 3, model x, model y, cybertruck) — proportion la plus élevée
- **Volkswagen** : requête (volkswagen, vw, id.3, id.4, id.buzz) — deuxième position
- **Lucid** : mentions significatives
- **Ford EV** : mentions modérées
- **Nissan** : requête (nissan, leaf) — mentions faibles
- **BMW** : requête (bmw, i3, i4, iX) — mentions faibles
- **BYD** : requête (BYD, byd) — mentions faibles
- **Rivian** : mentions très faibles



*Figure X : Pie chart de la distribution des marques mentionnées*

## 5. Résultats de la visualisation

### 5.1 Volume global des discussions

- Total de posts analysés : 4 484
- Période couverte : 2026-01-13
- Couverture : discussions liées aux voitures électriques sur Bluesky

### 5.2 Répartition des sentiments

Les visualisations montrent :

- Sentiment neutre dominant (majorité des posts)
- Sentiment négatif : 20,2% (937 posts)
- Sentiment positif : 13,2% (612 posts)

- Score moyen de sentiment : -0,08 (légèrement négatif, proche de la neutralité)

### **5.3 Thèmes les plus discutés**

Ordre d'importance (par nombre de mentions) :

1. Infrastructure : 42% mentions (problèmes de disponibilité, accès, localisation)
2. Autonomie : 20% mentions (préoccupations sur la portée, l'anxiété de la batterie)
3. Prix : 15% mentions (coût, accessibilité, valeur)

### **5.4 Marques les plus mentionnées**

- Tesla : mentionnée le plus fréquemment
- Volkswagen : deuxième position
- Autres marques (Lucid, Ford EV, Nissan, BMW, BYD, Rivian) : mentions plus faibles

## **6. Interprétation des résultats**

### **6.1 Insights sur les sentiments**

- Discussions majoritairement informatives (neutres)
- Présence modérée de critiques (négatif : 20,2%)
- Expressions positives limitées (positif : 13,2%)
- Tendance globale légèrement négative mais proche de la neutralité

### **6.2 Insights sur les thèmes problématiques**

- Infrastructure : point critique principal (42 mentions)
- Autonomie : préoccupation significative (20 mentions)
- Prix : préoccupation modérée (15 mentions)

### **6.3 Insights sur les marques**

- Tesla : visibilité la plus élevée
- Diversification : plusieurs marques mentionnées, mais concentration sur Tesla et Volkswagen
- Émergence : marques comme Lucid et Rivian présentes mais moins visibles

## **7. Avantages de la visualisation Grafana**

### **7.1 Interactivité**

- Filtrage temporel via le time picker
- Actualisation automatique des données
- Navigation entre dashboards

### **7.2 Flexibilité**

- Personnalisation des requêtes

- Ajout/modification de panels
- Export des données

### **7.3 Performance**

- Requêtes optimisées sur Elasticsearch
- Affichage rapide des résultats
- Gestion efficace de grands volumes

## **8. Limitations et améliorations futures**

### **8.1 Limitations actuelles**

- Requêtes basées sur des mots-clés (pas de NLP avancé)
- Sentiment simplifié (lexical)
- Pas d'analyse temporelle approfondie (focus sur les totaux)

### **8.2 Améliorations proposées**

- Intégration de modèles NLP pour un sentiment plus précis
- Analyse de l'évolution temporelle des tendances
- Détection automatique de nouveaux thèmes émergents
- Analyse de corrélations entre thèmes et sentiments
- Visualisation des réseaux d'influence entre utilisateurs

**Lien Github:**

<https://github.com/bentalebdev/bluesky-ev-project>

## **CONCLUSION :**

Ce projet a permis de déployer avec succès une architecture Big Data robuste (utilisant Docker, Kafka, Spark et MongoDB) pour analyser en temps réel les flux du réseau social Bluesky . L'étude de plus de 4 600 publications sur les voitures électriques révèle un sentiment globalement neutre (66,6 %), mais met en évidence des points de friction majeurs : les critiques se concentrent principalement sur l'infrastructure de recharge (42 % des mentions) et l'autonomie (20 %), éclipsant souvent les aspects positifs liés à l'environnement . En identifiant les acteurs clés comme Tesla et les dynamiques de polarisation, cette application démontre l'efficacité de l'analyse technique pour comprendre les tendances sociétales sur les plateformes décentralisées .

