Université Mohammed V - Rabat
École Nationale Supérieure d'Informatique
et d'Analyse des Systèmes

Big Data Project Rapport

MAJOR

Business Intelligence & Analytics (BI&A)

OBJECT:

Real time social network analysis : The Palistnien cause

Prepared by:

ERRAFIA Meriem

LOUKILI Bouchra

JABIRI Abderrahmane

HDADACHE Niama

Supervised by:

Mr. EL ALAMI Yasser

Jury Member:

Mr. EL ALAMI Yasser

Academic year 2025-2026

List of Figures

Contents

General Introduction

The rapid growth of social media platforms has led to an unprecedented volume of user-generated content, making these platforms a valuable source of data for understanding public opinion, social dynamics, and information diffusion. Among these platforms, Reddit stands out due to its community-based structure, topic-oriented discussions, and high level of user engagement. Each subreddit represents a focused thematic space where users express opinions, share information, and interact through posts and comment threads.

In parallel, the increasing availability of large-scale data has driven the adoption of Big Data technologies capable of processing, storing, and analyzing massive and heterogeneous datasets. Traditional data processing approaches are no longer sufficient to handle the velocity, volume, and variety of social media data. As a result, distributed data processing frameworks and data engineering pipelines have become essential for extracting meaningful insights from such data sources.

This the perfect context to practice what we learn in the lesson of big data engineering and data analytics.

This project is situated at the intersection of social network analysis and Big Data engineering. It focuses on the collection, transformation, storage, and analysis of Reddit data related to politically and socially sensitive topic the Palestinian cause. The objective is to design an end-to-end data pipeline that enables efficient data ingestion, scalable processing, and structured storage, while ensuring data quality and reproducibility.

To achieve this, the project leverages modern data engineering tools and architectures. By structuring unstructured social media data into analyzable formats, the project aims to support further analytical tasks such as sentiement analysis.

Overall, this project provides a practical framework for handling large-scale social media data and demonstrates how Big Data technologies can be applied to real-world social network analysis scenarios.

Chapter 1

General Context of the Project

1.1 Problem Statement

Although social media data is very rich, it is also difficult to analyze. The data is unstructured, comes in large volumes, and is generated continuously. Simple data processing tools are not enough to handle this type of data, especially when real-time or near real-time analysis is needed.

Another challenge is understanding the meaning behind the text. Posts and comments often contain emotions, opinions, and multiple discussion themes. Without proper tools, it is hard to know whether users express positive, negative, or neutral feelings.

The main problem addressed in this project is how to collect and process Reddit data efficiently, and how to analyze discussions related to the Palestinian cause using sentiment analysis.

1.2 Project Objectives

The objective of this project is to build a Big Data application that analyzes Reddit discussions related to the Palestinian cause. The project aims to collect posts and comments from Reddit and process them using scalable data processing tools.

One important goal is to perform sentiment analysis in order to understand the emotions expressed in the discussions, such as positive, negative, or neutral opinions.

The project also aims to structure the processed data in a way that allows easy visualization and interpretation. This helps transform raw social media data into meaningful information that can support analysis and decision-making.

1.3 Project Planning

To ensure the smooth progress of the project, a Gantt chart was used for planning and task organization, while also highlighting the dependencies between certain tasks. After exploring different social media platforms to identify a suitable data source, the data collection process was initiated. A sample of the collected data was then shared with the other team members so they could start working on their respective tasks in parallel.

Although some tasks may appear time-consuming, the work was carried out alongside other academic projects and courses, as well as during the examination period. For this reason, careful planning and task distribution were essential to respect deadlines and maintain steady progress throughout the project.

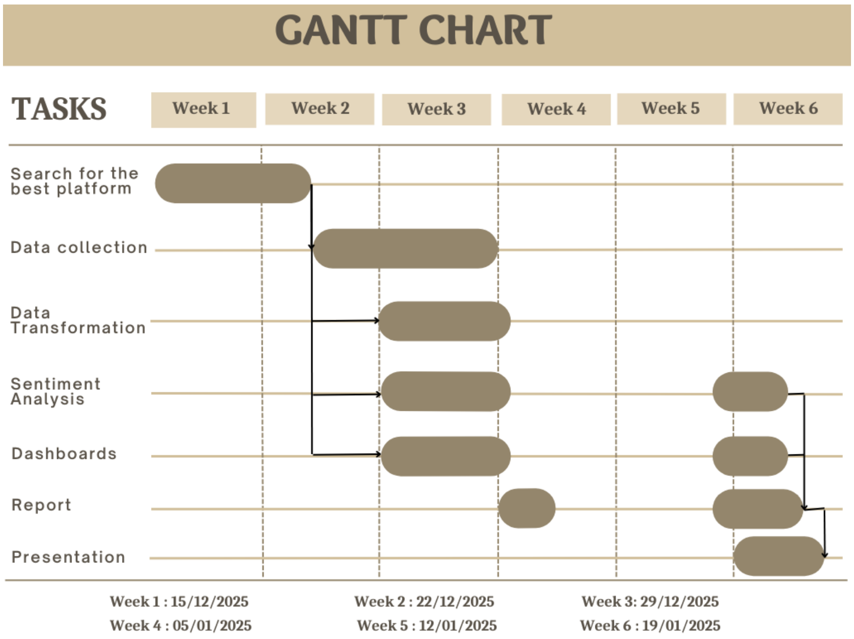


Figure 1.1: Gantt Chart

Chapter 2

Design and Implementation of the solution

2.1 Global Architecture Overview

After defining the different phases of the project, this section presents the global architecture of the system. The architecture follows a clear and linear data flow, from data collection to visualization.

The project is based on a streaming approach rather than batch processing. Data is processed in near real time, which allows faster analysis and more up-to-date results. The architecture follows an ETL pipeline composed of three main steps: Extract, Transform, and Load.

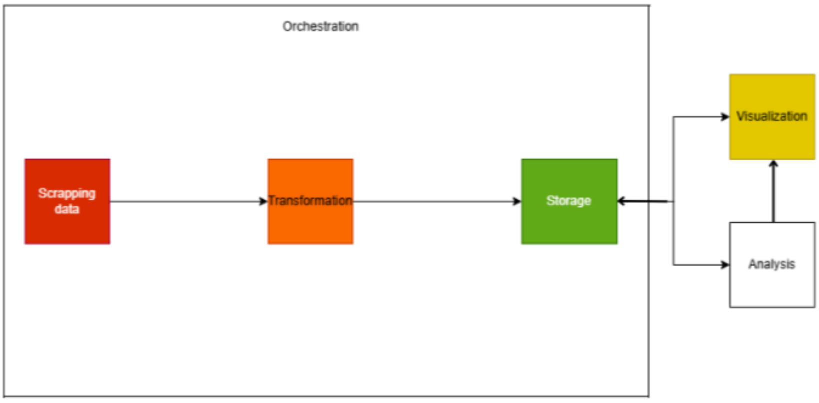


Figure 2.1: Global Architecture of the Project

This architecture ensures a clear separation between data ingestion, processing, analysis, and visualization, making the system scalable, maintainable, and easy to extend.

2.2 Phase 0: Choosing the Best Platform for Data Collection

The first phase of the project focuses on selecting the most suitable social media platform for data collection. At the beginning, several platforms were considered, such as Reddit, Discord, Telegram, and others. Different APIs were tested in order to access data from these platforms.

However, most of these APIs require special permissions, private access, or strict authentication rules, which made them difficult or impossible to use within the scope of this project. In addition, some platforms limit data access or restrict the type of information that can be collected.

For these reasons, Reddit was selected as the main data source. Reddit provides public discussions and is more accessible for data extraction. Although some API limitations still existed, data collection was made possible by scraping public Reddit content.

2.3 Phase 1: Data Collection

After deciding to scrape data from Reddit, this approach allowed us to retrieve posts, comments, and replies related to the Palestinian cause without accessing private or restricted information. The Reddit platform architecture was taken into consideration, particularly its organization into topic-based communities called subreddits.

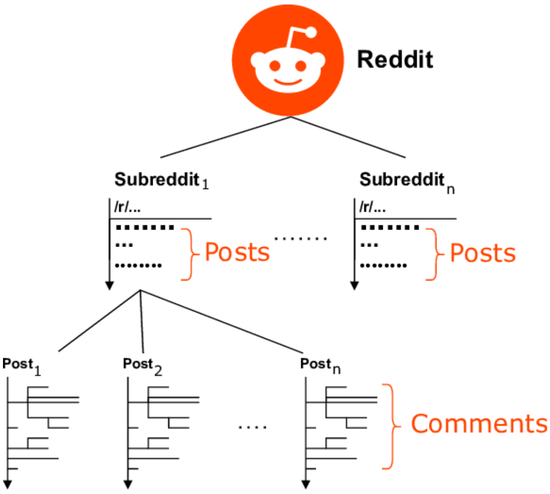


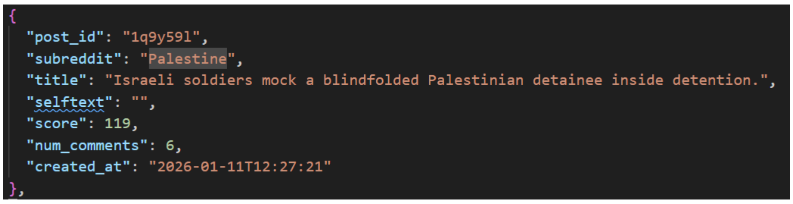
Figure 2.2: Reddit Structure

To ensure rich and relevant data, subreddits with a large number of members and frequent discussions related to Palestine and Israel were selected. The main subreddits used in this project are `r/Palestine`, `r/IsraelPalestine`, and `r/IsraelUnderAttack` because these subreddits are among the most popular and long-established, each with hundreds of thousands of members; for example, `r/Palestine`, which was created in 2008 and has over 315 k members.

The scraping process was designed with several important considerations. To simulate near real-time data collection, the system scrapes data every 20 minutes. During each execution, up to 20 posts are collected, along with their associated comments, and up to 10 replies per comment. These parameters are configurable and can be increased or reduced at any time depending on system performance or analysis needs.

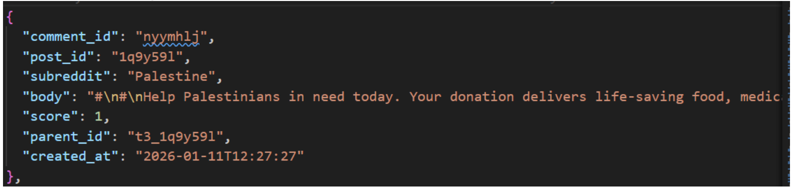
The data collection process is based on an incremental scraping strategy using a sliding time window. For each subreddit, the system maintains a checkpoint corresponding to the most recently processed timestamp. During each execution, posts from the recent past are intentionally revisited within a one-week revision window in order to capture new comments or replies that may have been added after the initial ingestion.

Posts that fall outside this time window are considered outdated and are no longer reprocessed for comment collection. This strategy helps reduce unnecessary processing while ensuring that late interactions are not missed. those are some exemples



```
{
  "post_id": "1q9y591",
  "subreddit": "Palestine",
  "title": "Israeli soldiers mock a blindfolded Palestinian detainee inside detention.",
  "selftext": "",
  "score": 119,
  "num_comments": 6,
  "created_at": "2026-01-11T12:27:21"
},
```

Figure 2.3: Sample of scraped post information



```
{
  "comment_id": "nyymhlj",
  "post_id": "1q9y591",
  "subreddit": "Palestine",
  "body": "#\n#\nHelp Palestinians in need today. Your donation delivers life-saving food, medic
  "score": 1,
  "parent_id": "t3_1q9y591",
  "created_at": "2026-01-11T12:27:27"
},
```

Figure 2.4: Sample of scraped comment information

The following tables summarize the attributes that were scraped from Reddit for posts, comments, and replies.

2.3.1 Post Attributes

Attribute	Description
`post_id`	Unique identifier of the Reddit post
`subreddit`	Name of the subreddit where the post was published
`title`	Title of the post
`selftext`	Textual content or description of the post
`score`	Number of votes received by the post
`num_comments`	Total number of comments on the post
`created_at`	Publication date of the post

Table 2.1: Attributes Collected for Reddit Posts

2.3.2 Comment and Reply Attributes

Attribute	Description
`comment_id`	Unique identifier of the comment or reply
`post_id`	Identifier of the associated post
`subreddit`	Name of the subreddit
`body`	Textual content of the comment or reply
`score`	Number of votes received by the comment or reply
`parent_id`	Identifier of the parent element; `t3` indicates a comment linked to a post, while `t1` indicates a reply to another comment
`created_at`	Publication date of the comment or reply

Table 2.2: Attributes Collected for Reddit Comments and Replies

2.4 Phase 2: Data Transformation

After data ingestion, a dedicated transformation phase was implemented to convert raw Reddit data into clean, structured, and analytics-ready datasets. This phase was carried out using Apache Spark in micro-batch mode and was applied consistently to both *comments* and *posts*, with minor adaptations depending on the nature of each data type.

The objectives of this phase were fourfold:

- Remove noise and artifacts introduced during data scraping
- Normalize textual content to ensure consistency
- Eliminate duplicated and spam-like records
- Prepare high-quality datasets for downstream storage and visualization

2.4.1 Schema Parsing and Validation

Messages were consumed from Kafka topics and deserialized from JSON format using predefined schemas. For comments, fields such as `comment_id`, `post_id`, textual content, author information, and timestamps were extracted. For posts, the schema included identifiers, subreddit information, post titles, post bodies, and associated metadata.

Records missing essential identifiers were filtered out to ensure data integrity and prevent the propagation of incomplete data.

2.4.2 Text Normalization and Cleaning

A comprehensive text-cleaning pipeline was applied to both comments and posts. For posts, the textual content was constructed by concatenating the post title and body, while for comments, the transformation focused on the comment body.

The following transformations were applied:

- Conversion of text to lowercase to enable case-insensitive analysis
- Replacement of URLs with a placeholder token `<url>` to reduce noise while preserving semantic

intent

- Removal of escaped characters introduced during scraping (e.g., `\n`, `\r`, `\t`)
- Removal of Reddit-specific formatting and markdown symbols
- Removal of Reddit usernames (e.g., `u/username`)
- Normalization of hashtags by removing the hash symbol while retaining the underlying word
- Whitespace normalization to ensure uniform spacing

Stop words were intentionally preserved to maintain linguistic context, which is particularly important for sentiment and discourse analysis.

2.4.3 Spam and Noise Filtering

To improve data quality, rule-based filters were introduced to remove non-informative and noisy content. These included automated moderation messages, deleted or removed content, and repetitive fundraising or donation-related spam commonly observed in social media discussions.

This filtering step significantly reduced dataset noise and ensured that retained records reflected genuine user-generated content.

2.4.4 Language Detection

A language detection step was applied to retain only English-language records. This was implemented using a Python User Defined Function (UDF) with built-in exception handling to prevent runtime failures on malformed or empty text. Non-English records were excluded to maintain consistency in downstream analysis.

2.4.5 Deduplication Strategy

Social media platforms frequently exhibit duplicated or near-duplicated content, particularly in the case of coordinated messaging or spam. To address this, a content-based deduplication strategy was implemented.

For both posts and comments, a SHA-256 hash was computed from the cleaned textual content. Records sharing the same hash value were considered duplicates, and only a single instance was retained. This approach ensured robustness even when identical content was associated with different identifiers.

2.4.6 Final Dataset Preparation

After completing all transformation steps, a final selection of relevant attributes was performed. Cleaned and deduplicated datasets were written to MongoDB, with separate collections maintained for posts and comments.

This transformation phase ensured that raw, heterogeneous social media data was converted into structured, reliable, and analytics-ready datasets suitable for real-time analysis and visualization.

2.5 Phase 3: Sentiment Analysis

2.5.1 Objectives

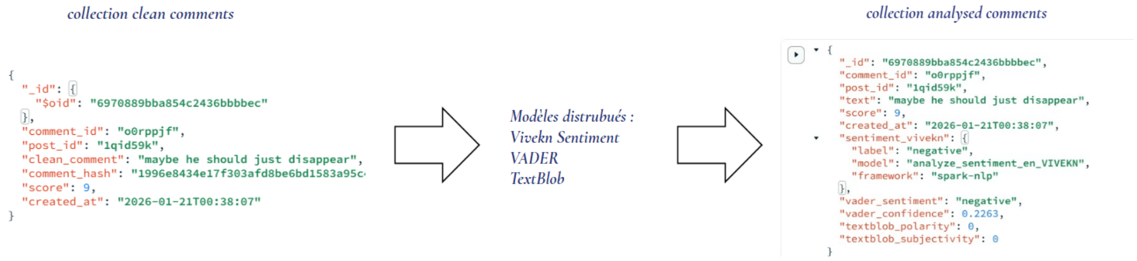
The sentiment analysis phase aims to:

- Analyze the sentiment of Reddit comments in near real time
- Capture complementary sentiment signals by combining multiple NLP models
- Ensure scalability and efficiency through integration with Apache Spark and Airflow
- Provide structured sentiment outputs for downstream aggregation and visualization

To achieve these objectives, three complementary sentiment analysis models were integrated into a unified Spark-based pipeline orchestrated by Apache Airflow:

- Vivekn Sentiment Model (Spark NLP)
- VADER (Valence Aware Dictionary and sEntiment Reasoner)
- TextBlob

Each model brings a different methodological perspective, allowing cross-validation of sentiment signals and increased robustness of the analysis.



```
{
  "_id": {
    "$oid": "6970889bba854c2436bbbbec"
  },
  "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"
}
```

```
{
  "_id": "6970889bba854c2436bbbbec",
  "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
}
```

Figure 2.5: Global sentiment analysis pipeline.

Figure 2.5 illustrates the logical flow of the sentiment analysis pipeline. Each cleaned Reddit comment is analyzed independently by three complementary sentiment models. The Vivekn model produces a categorical sentiment label, while VADER and TextBlob generate continuous sentiment scores. The outputs are then merged into a unified schema, enabling cross-model comparison, robustness analysis, and statistical aggregation.

2.5.2 Vivekn Sentiment Model (Spark NLP)

The Vivekn Sentiment model is part of the Spark NLP ecosystem and is implemented natively on the JVM, making it highly efficient and well-suited for large-scale distributed processing.

This model is inspired by an enhanced Naive Bayes approach and was originally evaluated on the IMDB Movie Reviews dataset. Reported accuracy on this benchmark is approximately **88.8%**, making it one of the most reliable classical sentiment classifiers available in Spark NLP.

The model produces a categorical sentiment label for each input text:

- `positive`
- `negative`
- `neutral`

Thanks to its native Spark integration, Vivekn was applied directly within the Spark NLP pipeline without any Python overhead, ensuring optimal performance and scalability.

2.5.3 VADER Sentiment Analysis

VADER (Valence Aware Dictionary and sEntiment Reasoner) was introduced in 2014 by C. J. Hutto and Eric Gilbert and is specifically designed for sentiment analysis of social media text. It relies on a lexicon-based approach enriched with rules that capture punctuation, capitalization, intensifiers, negations, and emoticons.

For each input text, VADER returns a structured score dictionary of the form:

```
{'neg': 0.0, 'neu': 0.238, 'pos': 0.762, 'compound': 0.6369}
```

The scores are interpreted as follows:

- `neg`: proportion of negative sentiment
- `neu`: proportion of neutral sentiment
- `pos`: proportion of positive sentiment
- `compound`: normalized overall sentiment score in the range [-1, +1]

The compound score is commonly interpreted using the following thresholds:

- `compound` ≥ 0.05 → Positive sentiment
- `compound` ≤ -0.05 → Negative sentiment
- $-0.05 <$ `compound` < 0.05 → Neutral sentiment

In this project, VADER was executed using **Pandas UDFs** within Spark, enabling vectorized processing while maintaining high performance.

2.5.4 TextBlob Sentiment Analysis

TextBlob is a lightweight, rule-based sentiment analysis library built on top of classical NLP techniques. Unlike machine learning models, TextBlob does not learn from data and relies entirely on predefined lexical rules.

2.5.4.0.1 Model Characteristics

- Lexical and rule-based approach
- No machine learning or model training
- Simple and interpretable sentiment scores

2.5.4.0.2 Observed Performance Empirical studies show that TextBlob achieves:

- Approximately **55% to 65% accuracy** on simple texts and short reviews
- Lower performance on complex discourse, sarcasm, or social media-specific language

Compared to VADER, TextBlob is generally less accurate for social media content but remains useful as a complementary baseline.

2.5.4.0.3 Output Format TextBlob returns a sentiment object containing two continuous scores:

```
Sentiment(polarity=0.5, subjectivity=0.6)
```

The scores are interpreted as:

- `polarity` $\in [-1, +1]$: negative to positive sentiment
- `subjectivity` $\in [0, 1]$: objective to subjective content

As with VADER, TextBlob was executed using **Pandas UDFs** to ensure efficient parallel execution within Spark.

2.5.5 Execution Strategy and Integration

The sentiment analysis pipeline was designed to leverage the strengths of both JVM-based and Python-based components:

- Vivekn Sentiment runs natively in Spark NLP on the JVM for maximum scalability
- VADER and TextBlob are applied using Pandas UDFs for vectorized Python execution
- All models are orchestrated using Apache Airflow to ensure reproducibility

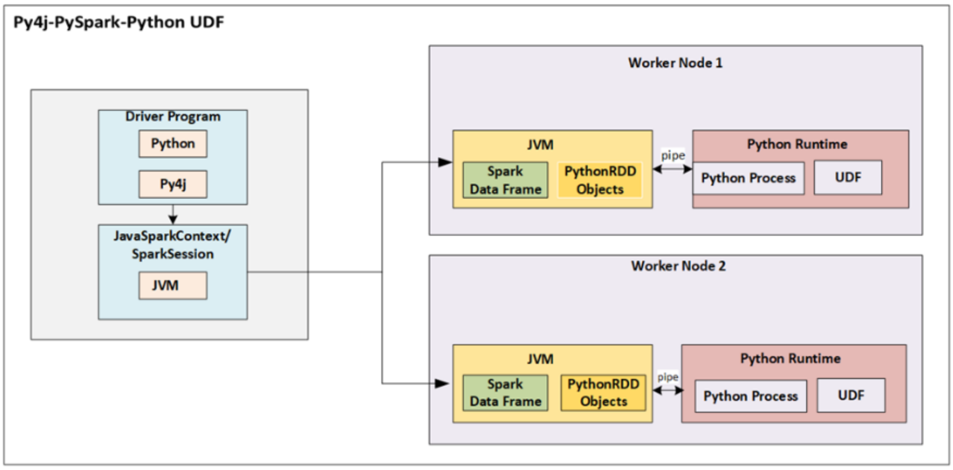


Figure 2.6: Execution architecture of Pandas UDFs in PySpark using Py4J.

As shown in Figure 2.6, Pandas UDF execution relies on the Py4J bridge to enable communication between the Spark JVM and the Python runtime. For each executor, data is serialized into Python processes where vectorized sentiment analysis is applied. The results are then returned to the JVM and integrated back into the distributed Spark DataFrame.

This hybrid execution strategy allows the system to balance scalability, flexibility, and analytical depth while maintaining near real-time sentiment analysis capabilities.

2.6 Phase 4: Dashboards and Visualization

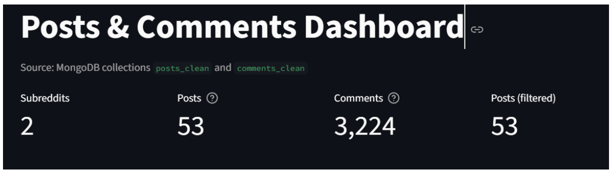
The dashboards presented in this section provide an analytical view of the data collected and processed throughout the project. They are designed to transform raw and processed data into clear, interpretable insights, supporting the understanding of discussion dynamics, user engagement, and sentiment trends. Two complementary dashboards are presented: the first focuses on posts and comments stored in MongoDB, highlighting activity and engagement patterns, while the second analyzes sentiment results generated offline and stored as JSON files. Together, these dashboards illustrate both the structural and emotional aspects of the social network data.

Chapter 3

Implementation and Results

3.1 Dashboards

3.1.1 Posts and Comments Activity Dashboard



Posts & Comments Dashboard

Source: MongoDB collections `posts_clean` and `comments_clean`

Subreddits	Posts	Comments	Posts (filtered)
2	53	3,224	53

Figure 3.1: Global Overview of Reddit Activity

This visualization provides a high-level summary of the collected Reddit data. It shows the number of subreddits analyzed, the total number of posts, and the total number of comments stored in the database. The metrics highlight the scale of the dataset and the level of user engagement: although the analysis covers a limited number of subreddits, the high volume of comments compared to posts indicates intensive discussion activity. This overview sets the context for the detailed analyses presented in the following visualizations.

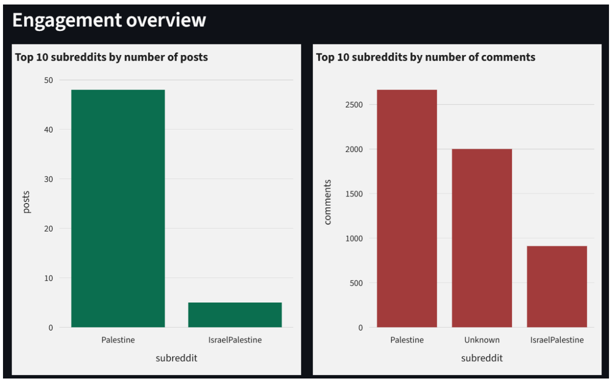


Figure 3.2: Engagement Distribution Across Subreddits

This visualization compares content creation and user interaction across subreddits. The Palestine subreddit clearly dominates in both the number of posts and comments, indicating high activity and engagement. The presence of a significant number of comments under Unknown suggests missing or aggregated metadata. Overall, the charts show that a small number of subreddits generate most of the discussions.

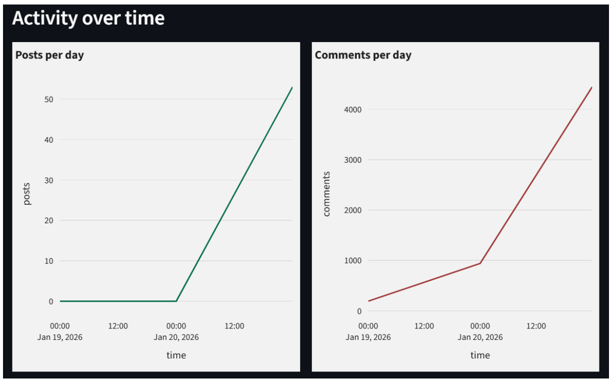


Figure 3.3: Posts and Comments Activity Over Time

These time-series charts show how activity evolves over the analyzed period. The number of posts increases sharply on the second day, which is followed by a much larger rise in comments. This pattern indicates that content publication triggers delayed but intense user engagement, with discussions growing rapidly after posts are published.

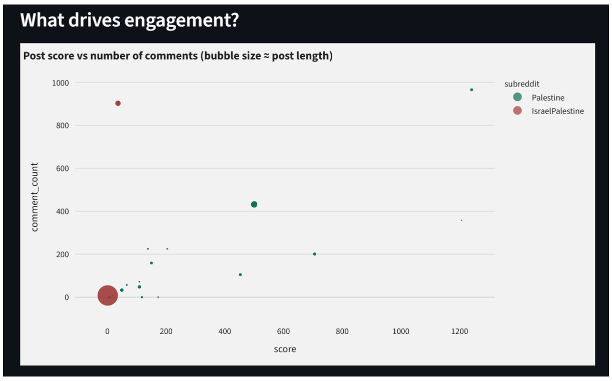


Figure 3.4: Relationship Between Post Score and User Engagement

This scatter plot explores how post popularity relates to engagement. Posts with higher scores generally attract more comments, indicating a positive relationship between visibility and discussion volume. However, some low-score posts also generate significant engagement, suggesting that controversial or sensitive topics can drive discussions independently of popularity. Bubble size shows that longer posts do not necessarily lead to more comments, highlighting that content relevance matters more than length.

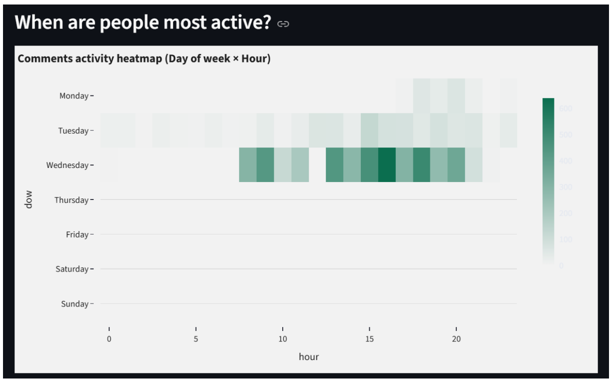
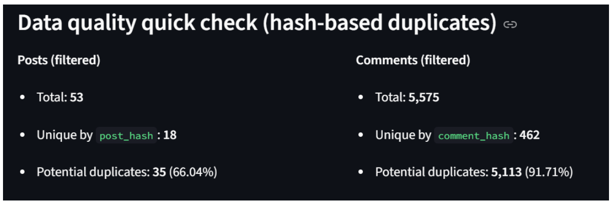


Figure 3.5: Temporal Patterns of Comment Activity

This heatmap shows when users are most active based on the day of the week and hour of the day. Comment activity peaks during mid-week, particularly on Wednesdays, and is concentrated in the afternoon and early evening hours. This pattern suggests that discussions intensify during typical working-day hours, reflecting synchronized user engagement around shared events or topics.



Data quality quick check (hash-based duplicates)

Posts (filtered)

- Total: 53
- Unique by `post_hash` : 18
- Potential duplicates: 35 (66.04%)

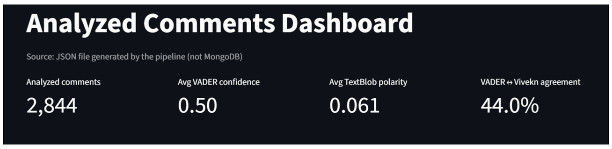
Comments (filtered)

- Total: 5,575
- Unique by `comment_hash` : 462
- Potential duplicates: 5,113 (91.71%)

Figure 3.6: Data Quality Assessment Using Hash-Based Deduplication

This visual provides a quick assessment of data quality by identifying potential duplicates using hash values. The results show that a large proportion of both posts and comments share identical content, especially comments, where duplication is particularly high. This highlights the presence of repeated or copied content and justifies the use of deduplication and cleaning steps before performing analytical or sentiment-based analyses.

3.1.2 Analyzed Comments Sentiment Dashboard



Analyzed Comments Dashboard

Source: JSON file generated by the pipeline (not MongoDB)

Analyzed comments	Avg VADER confidence	Avg TextBlob polarity	VADER ↔ Vivekn agreement
2,844	0.50	0.061	44.0%

Figure 3.7: Overall Sentiment Analysis Summary

This visualization provides a summary of the sentiment analysis results obtained from the processed comments. It shows the total number of analyzed comments, the average confidence of the VADER model, the average polarity score from TextBlob, and the level of agreement between VADER and Vivekn models. The metrics indicate moderate sentiment intensity and a relatively low agreement between models, highlighting the complexity and subjectivity of opinions expressed in the discussions.

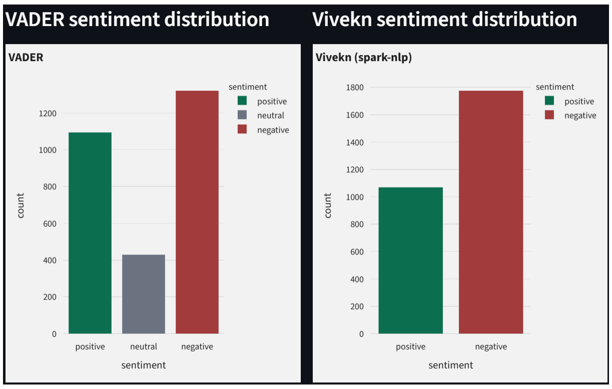
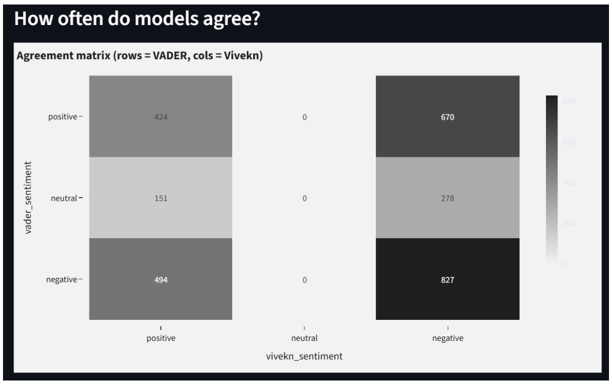


Figure 3.8: Sentiment Distribution by Model (VADER vs Vivekn)

These charts compare how two different sentiment analysis models classify the same set of comments. Both VADER and Vivekn identify a predominance of negative sentiment, indicating that discussions are largely emotionally charged. VADER additionally detects a notable proportion of neutral comments, while Vivekn produces a more polarized classification. This difference reflects the distinct modeling approaches and highlights why comparing multiple sentiment models is important for robust interpretation.



vader_sentiment \ vivekn_sentiment	positive	neutral	negative
positive	424	0	670
neutral	151	0	278
negative	494	0	827

Figure 3.9: Agreement Between Sentiment Models

This agreement matrix compares the sentiment labels assigned by VADER and Vivekn for the same comments. The strongest agreement occurs for negative sentiment, while neutral sentiment shows little alignment between the models. Overall, the matrix reveals frequent disagreement, reflecting differences in model design and sensitivity, and reinforcing the need to use multiple sentiment analysis approaches for a more balanced interpretation.

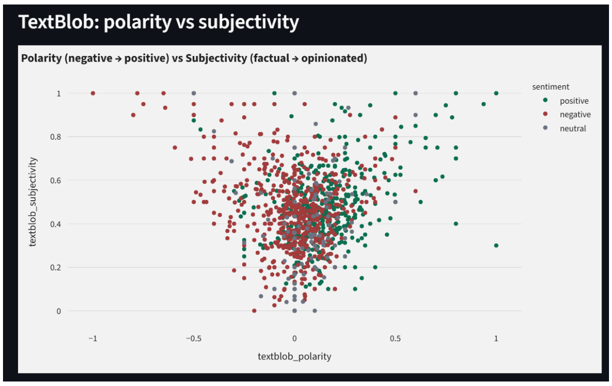


Figure 3.10: Emotional Tone and Subjectivity of Comments

This scatter plot illustrates the relationship between polarity and subjectivity in comments based on TextBlob analysis. Most comments cluster around neutral polarity with medium to high subjectivity, indicating that discussions are largely opinion-based rather than factual. Negative comments appear slightly more subjective, while positive comments are more dispersed across polarity values. This suggests that emotionally charged opinions dominate the conversations.

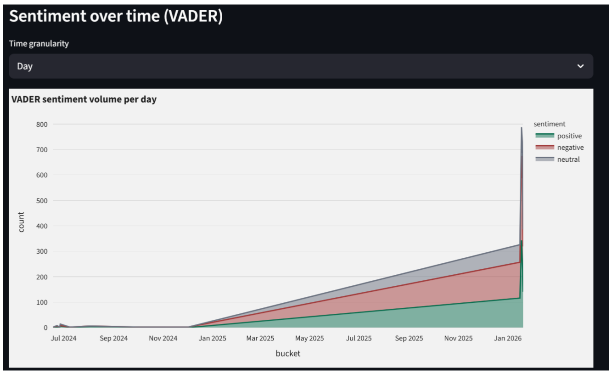


Figure 3.11: Evolution of Sentiment Over Time (VADER)

This area chart shows how the volume of sentiments evolves over time based on the VADER model. Negative sentiment consistently dominates the discussion, followed by neutral and positive sentiments. A sharp increase toward the end of the period indicates a surge in commenting activity, suggesting heightened public attention or triggering events that intensified emotional reactions.

3.2 Technologies Used

To implement the proposed solution, several tools and technologies were required. Although the project statement suggested some technologies to be used, this does not automatically justify their selection. For this reason, this section presents the chosen technologies and explains why they were suitable for this project.

This is the architecture of the solution, including the technologies used. We will clarify each of them afterward.

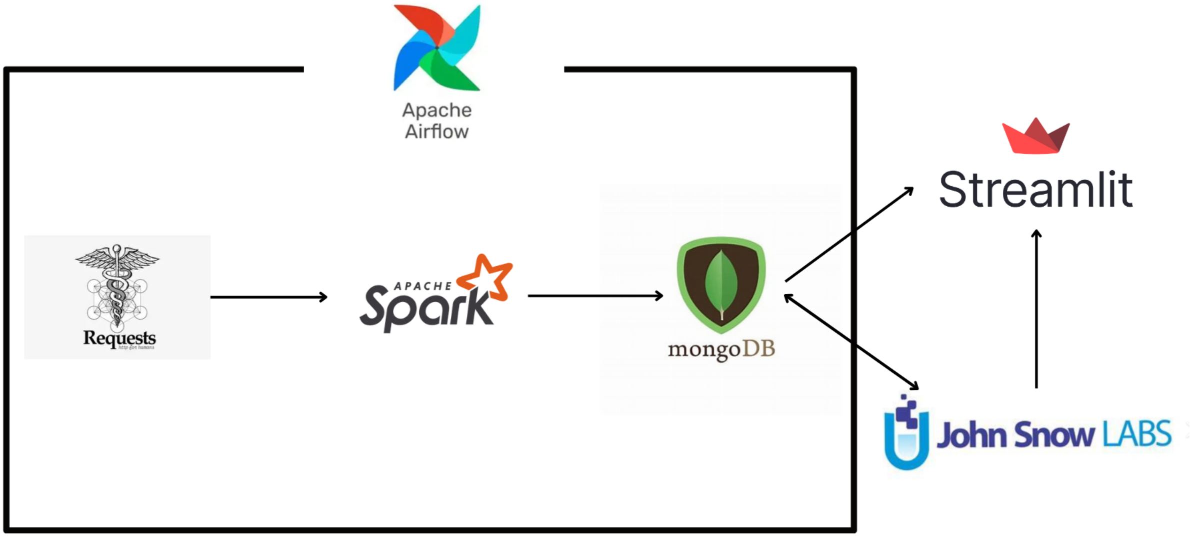


Figure 3.12: Architecture with technologies used

3.2.1 Requests Library

After several attempts to use the Reddit API, which were unsuccessful due to access and permission limitations, web scraping was chosen as the solution to collect data from Reddit. The Python `Requests` library was sufficient to scrape public posts, comments, and related information.

This library is simple to use, lightweight, and well adapted for retrieving web content. It allowed reliable data collection without requiring special authentication or private access.

3.2.2 Apache Spark

Apache Spark was used mainly for data processing and transformation. One reason for choosing Spark was to apply the concepts learned during the course in a practical project. In addition, Spark is one of the most widely used technologies in Big Data engineering, especially for handling large-scale data efficiently.

Spark allows fast processing and supports distributed computation, which makes it suitable for transforming and preparing large volumes of social media data for further analysis.

3.2.3 MongoDB

The collected data is mainly unstructured text data. MongoDB was chosen as the storage solution because it is a NoSQL database well suited for handling unstructured and semi-structured data.

In addition, MongoDB is flexible, easy to integrate with Python and Spark, and efficient for storing transformed data. It allows fast access to data for analysis and visualization purposes.

3.2.4 Airflow

To ensure orchestration between the different phases of the project, Airflow was used because it is the most suitable tool and we are familiar with it.

3.2.5 John Snow Labs

John Snow Labs is the company behind *Spark NLP*, a production-grade natural language processing library built on Apache Spark. Spark NLP runs natively on the JVM and enables scalable, distributed NLP pipelines with high performance.

In this project, Spark NLP was used to apply sentiment analysis at scale directly within Spark, eliminating Python overhead and ensuring efficient processing of large volumes of unstructured text data.

3.2.6 Streamlit

Streamlit is an open-source Python framework designed for building interactive data applications and dashboards with minimal overhead. It allows rapid development of data-driven interfaces directly from Python code, without requiring front-end development skills.

In this project, Streamlit was used to create interactive dashboards for exploring posts, comments, and sentiment analysis results. It enabled fast visualization of data stored in MongoDB and static JSON files, facilitating real-time insight generation and effective communication of analytical results.

Conclusion and Perspectives

Conclusion

This project demonstrated the design and implementation of a complete data pipeline for real-time social network analysis, from data ingestion to visualization. Reddit data was collected, processed, and stored using scalable technologies such as Kafka, Spark, and MongoDB. Advanced natural language processing techniques were applied to extract sentiment information from large volumes of unstructured text, enabling a deeper understanding of user engagement and emotional trends.

The developed dashboards provided an interactive and intuitive way to explore both structural data (posts and comments) and analytical results (sentiment analysis). The combination of multiple sentiment models highlighted the complexity of online discussions and emphasized the importance of model comparison for reliable interpretation. Overall, the project successfully transformed raw social media data into actionable insights while ensuring data quality and scalability.

Perspectives

Several improvements and extensions can be considered for future work. Additional social media platforms could be integrated to enable cross-platform analysis and broader insights. The sentiment analysis component could be enhanced by incorporating domain-specific or multilingual models to improve accuracy. Furthermore, real-time dashboard updates and alerting mechanisms could be added to monitor emerging trends or sudden changes in sentiment. Finally, extending the analysis to include topic modeling or network-based interaction analysis would provide a richer understanding of discourse dynamics and influence patterns.