

# Subreddit Meltdown

## Mental Health Trends from Reddit Data

## Data 228: Big Data Project

**Group:** 09  
**Submitted By:** Chandana Rondla,  
a Chauhan, Himaja Sree Thota,  
sh Aggarwal, Saurabh Suman,  
Shivam Mahendru  
**Submitted To:** Professor Sangjin Lee

# Introduction



## Motivation

In an age of hyper connectivity, anxiety and depression rates are soaring, challenging conventional public health monitoring. Reddit, with its 430 million monthly users, serves as an anonymous outlet for individuals to share their mental health struggles and concerns, revealing real-time societal trends.



## Research Questions

Key questions driving this analysis include:

- Did the COVID-19 pandemic cause a measurable rise in discussions related to anxiety and depression?
- Can sentiment spikes be linked to major societal events like economic crises or election cycles?
- Which Reddit communities act as early indicators of emotional distress—and how quickly do they respond after external shocks?





# Motivation

In an age of hyper-connectivity, anxiety and depression rates are soaring, challenging conventional public health monitoring. Reddit, with its 430 million monthly users, serves as an anonymous outlet for individuals to share their mental health struggles and concerns, revealing real-time societal trends.



# Research Questions

Key questions driving this analysis include:

- Did the COVID-19 pandemic cause a measurable rise in discussions related to anxiety and depression?
- Can sentiment spikes be linked to major societal events like economic crises or election cycles?
- Which Reddit communities act as early indicators of emotional distress—and how quickly do they respond after external shocks?

## Tools Used



### Frontend:

**Next.js**: Interactive dashboard UI

**Tailwind CSS**: Responsive design

### Backend:

**FastAPI**: RESTful APIs for queries

### Data Processing:

**PySpark**: Distributed processing of Reddit data

### Event Messaging:

**Apache Kafka**: Real-time data streaming

**Zookeeper**: Broker coordination

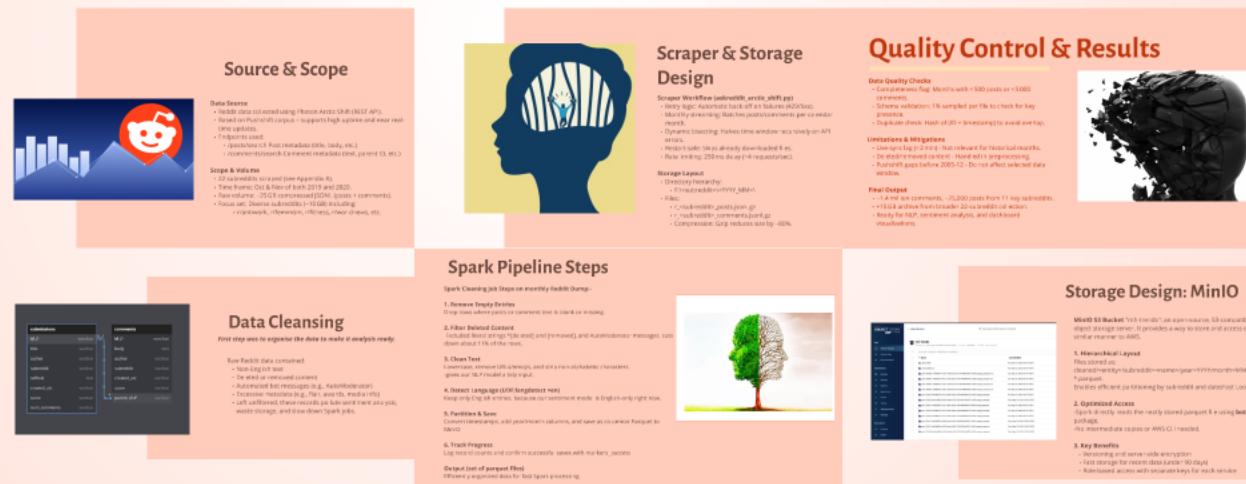
### Storage:

**MinIO**: S3-compatible object storage

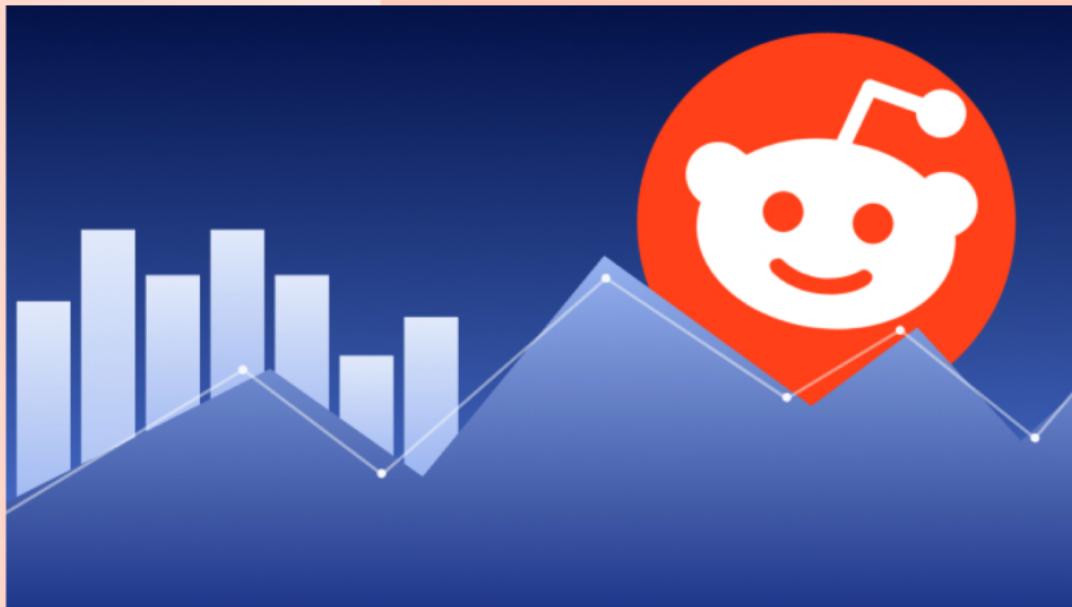
### Deployment:

**Docker**: Containerizes the entire stack

# Data Collection & Cleansing



# Source & Scope



## Data Source

- Reddit data collected using Photon Arctic-Shift (REST API).
- Based on Pushshift corpus – supports high uptime and near real-time updates.
- Endpoints used:
  - /posts/search Post metadata (title, body, etc.)
  - /comments/search Comment metadata (text, parent ID, etc.)

## Scope & Volume

- 22 subreddits scraped (see Appendix A).
- Time frame: Oct & Nov of both 2019 and 2020.
- Raw volume: ~25 GB compressed JSONL (posts + comments).
- Focus set: Diverse subreddits (~10 GB) including:
  - r/antiwork, r/feminism, r/fitness, r/worldnews, etc.



# Scraper & Storage Design

## Scraper Workflow (`askreddit_arctic_shift.py`)

- Retry logic: Automatic back-off on failures (429/5xx).
- Monthly streaming: Batches posts/comments per calendar month.
- Dynamic bisecting: Halves time window recursively on API errors.
- Restart-safe: Skips already downloaded files.
- Rate limiting: 250 ms delay (~4 requests/sec).

## Storage Layout

- Directory hierarchy:
  - F:\<subreddit>\<YYYY\_MM>\
- Files:
  - r\_<subreddit>\_posts.jsonl.gz
  - r\_<subreddit>\_comments.jsonl.gz
  - Compression: Gzip reduces size by ~80%.

# Quality Control & Results

## Data Quality Checks

- Completeness flag: Months with < 500 posts or < 5000 comments.
- Schema validation: 1% sampled per file to check for key presence.
- Duplicate check: Hash of (ID + timestamp) to avoid overlap.

## Limitations & Mitigations

- Live-sync lag (< 2 min) - Not relevant for historical months.
- Deleted/removed content - Handled in preprocessing.
- Pushshift gaps before 2005-12 - Do not affect selected data window.

## Final Output

- ~1.4 million comments, ~25,000 posts from 11 key subreddits.
- +15 GB archive from broader 22-subreddit collection.
- Ready for NLP, sentiment analysis, and dashboard visualizations.

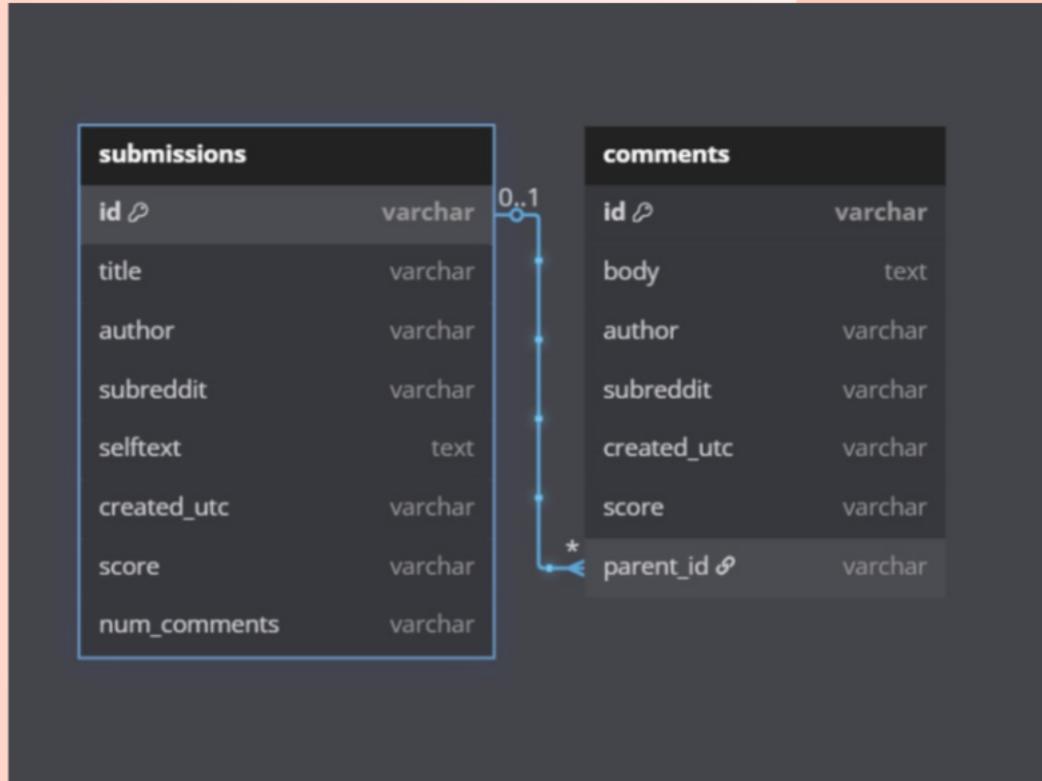


# Data Cleansing

*First step was to organise the data to make it analysis ready.*

Raw Reddit data contained:

- Non-English text
- Deleted or removed content
- Automated bot messages (e.g., AutoModerator)
- Excessive metadata (e.g., flair, awards, media info)
- Left unfiltered, these records pollute sentiment analysis, waste storage, and slow down Spark jobs.



# Spark Pipeline Steps

## Spark Cleaning Job Steps on monthly Reddit Dump:-

### 1. Remove Empty Entries

Drop rows where posts or comment text is blank or missing.

### 2. Filter Deleted Content

Excluded literal strings \*[deleted] and [removed], and AutoModerator messages. cuts down about 11% of the rows.

### 3. Clean Text

Lowercase, remove URLs/emojis, and strip non-alphabetic characters.  
-gives our NLP model a tidy input.

### 4. Detect Language (UDF;langdetect =en)

Keep only English entries, because our sentiment model is English-only right now.

### 5. Partition & Save

Convert timestamps, add year/month columns, and save as columnar Parquet to MinIO

### 6. Track Progress

Log record counts and confirm successful saves with markers \_success

### Output (set of parquet files)

Efficiently organized data for fast Spark processing.



# Storage Design: MinIO

The screenshot shows the MinIO Object Store interface. On the left, a sidebar menu includes options like User, Object Browser, Access Keys, Documentation, Administrator (Buckets, Policies, Identity, Monitoring, Events, Tiring, Site Replication, Settings), and Subscription (License, Health). The main area is titled 'Object Browser' and shows the 'mh-trends' bucket. The bucket was created on Sun, Apr 27 2025 02:14:01 (PDT) and has PRIVATE access with 1.5 GiB - 1003 Objects. The object list shows a hierarchical structure: mh-trends / cleaned / submissions / worldnews. The objects are listed by Name (sorted by Last Modified). The names include '\_SUCCESS', '\_SUCCESS.crc', and various part files (part-00000 to part-00017) with suffixes like .crc and .parquet. The last modified dates range from Fri, May 02 2025 03:47 (PDT) to Thu, May 01 2025 20:28 (PDT).

**MinIO S3 Bucket "mh-trends":** an open-source, S3-compatible object storage server. It provides a way to store and access data in a similar manner to AWS.

## 1. Hierarchical Layout

Files stored as:

cleaned/<entity>/subreddit=<name>/year=YYYY/month=MM/\*.parquet.

Enables efficient partitioning by subreddit and date(Fast Lookup)

## 2. Optimized Access

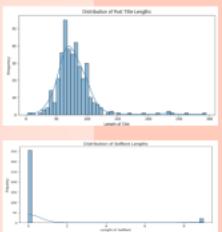
-Spark directly reads the neatly stored parquet file using **boto3** package.

-No intermediate copies or AWS CLI needed.

## 3. Key Benefits

- Versioning and server-side encryption
- Fast storage for recent data (under 90 days)
- Role-based access with separate keys for each service

# EDA & Frontend Development



## Exploratory Data Analysis

To understand mental health discussions on Reddit, we conducted EDA using pyspark, Matplotlib, and Seaborn.

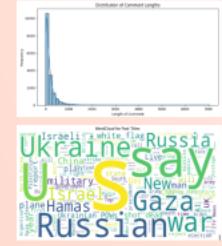
**Key Insights:**

1. Post Volume Over Time

Significant spikes in posting activity aligned with global stress events, especially during COVID-19.

2. Most Active Subreddits

Comments on r/mentalhealth, r/depression, r/stress, and r/mentalhealth had the highest activity, signifying their central role in mental health discourse.



## Exploratory Data Analysis

3. Word Clouds

Frequent terms such as "help", "a line", "stop", and "stress" indicate emotional, argyrous, and shared struggles.

4. Sentiment Distribution

Sentiment varied by time and topic, with notable shifts during significant events, highlighting the tool as a reflection of public mood.

Reddit Metrics

Sentiment Analysis Dashboard

Reddit Metrics

Comment Density Heatmap

## Frontend Development

### Frontend Architecture

- Framework: Next.js (React) + TypeScript for maintainability
- Styling: Tailwind CSS for fast, responsive design
- Visuals: React components
  - chartjs-2 for sentiment timelines
  - wordcloud for keyword displays
  - react-datepicker for time filtering
- APIs: RESTful endpoints from a FastAPI backend
- Deployment: Docker containerization for portability across systems



Reddit Metrics Dashboard

## Data Flow, State Management & UX

**Systems Integration & Data Flow:**

- Frontend sends API requests (via fetch/useState)
- Backend returns JSON, sentiment, events, keywords
- UI updates dynamically: charts, tables, word clouds

**Real-Time Updates:**

- Frontend triggers state and trigger re-renders on renders
- Visuals stay synchronized with every input change

**Design Focus:**

- Clear, clutter-free layout for readability
- Mobile responsiveness
- High performance and minimal load times
- Emphasis on user-driven discovery and cart

## Functionalities & User Experience

### Key Functionalities:

- Filtering Module:**
  - Select subreddit
  - Choose date range
  - Toggle between Posts/Comments
- Sentiment Timeline:**
  - Dynamic line chart of sentiment over time
  - Drill-down support for exact date and time
- Keywords:**
  - Display positive and negative keywords
  - Updated dynamically via socket filters
- Event Impact Table & Drill-downs:**
  - Show major events with sentiment scores
  - Click on event to drill down to related posts/comments with context
- User Benefits:**
  - Real-time insights
  - Drill-downs from macro trends to individual conversations

# Exploratory Data Analysis

To understand mental health discussions on Reddit, we conducted EDA using PySpark, Matplotlib, and Seaborn.

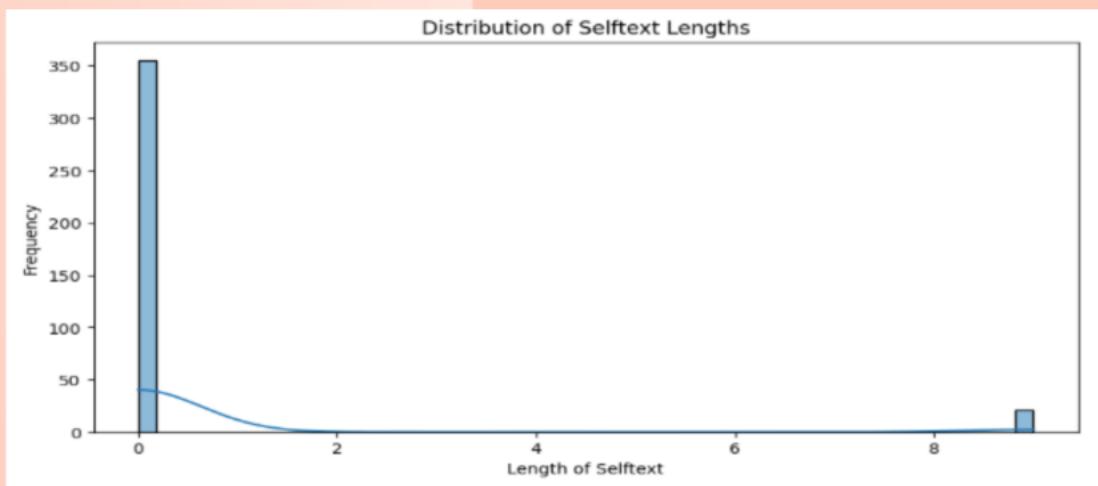
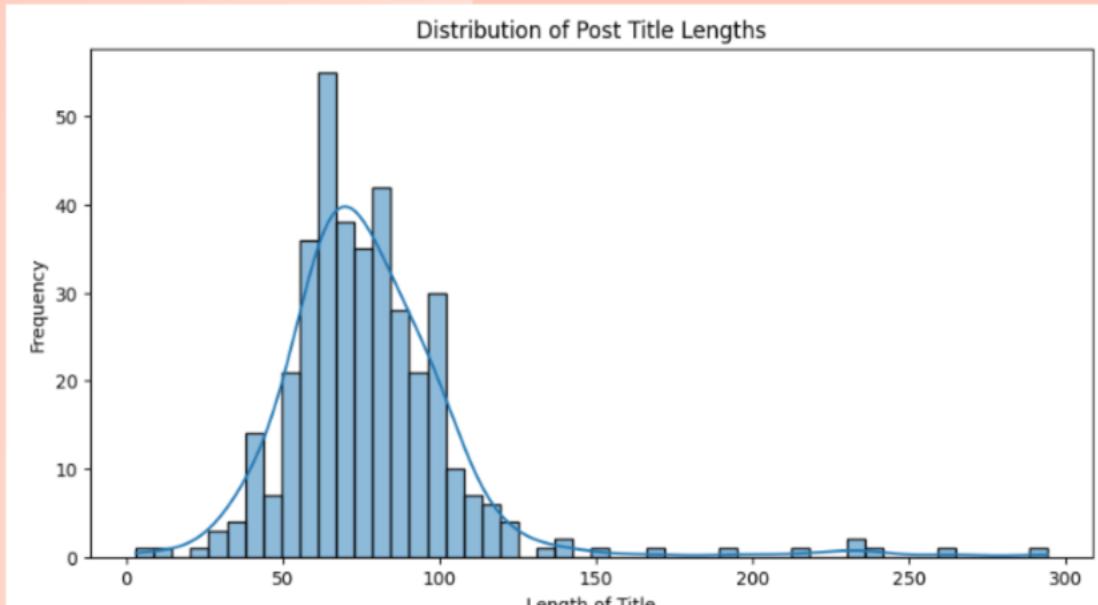
## Key Insights:

### 1. Post Volume Over Time

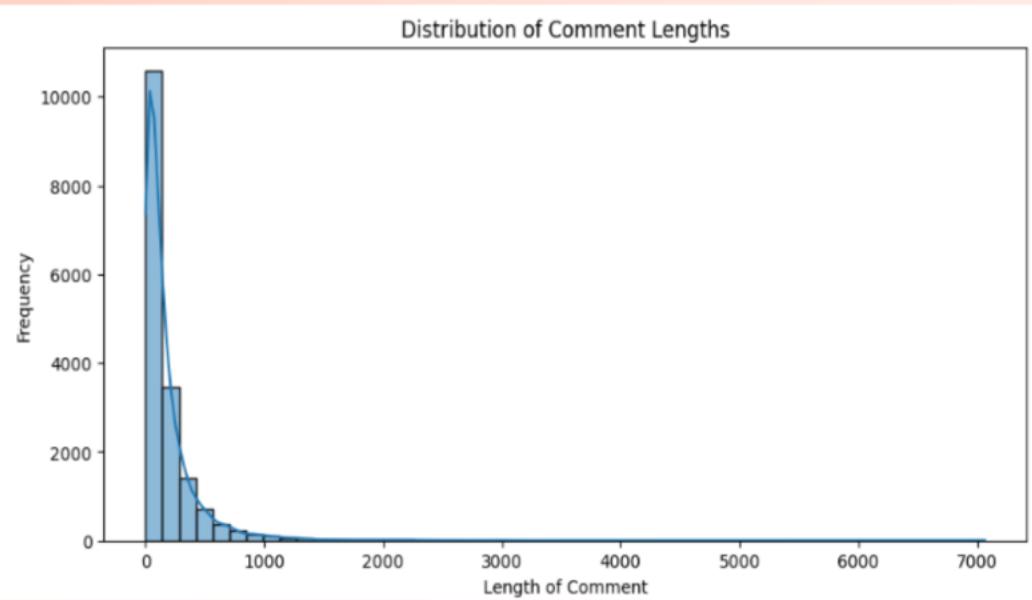
Significant spikes in posting activity aligned with global stress events, especially during COVID-19.

### 2. Most Active Subreddits

Communities like r/depression, r/anxiety, and r/mentalhealth had the highest activity, signaling their central role in mental health discourse.



# Exploratory Data Analysis



### 3. Word Clouds

Frequent terms such as "help", "alone", "cope", and "therapy" indicate emotional urgency and shared struggles.

## 4. Sentiment Distribution

Sentiment varied by time and topic, with noticeable shifts during societal events highlighting Reddit as a reflection of public mood

# Frontend Development

The screenshot shows the 'Reddit Meltdown' dashboard interface. At the top, there's a header with the title 'Reddit Meltdown' and a 'Advanced Dashboard' button. Below the header are three input fields: 'Subreddit' (set to 'r/fitness'), 'Date Range' (with start date '10/01/2019' and end date '11/30/2019'), and 'Content Type' (set to 'Comments' with a checked checkbox for 'Include Dynamic Events'). A note below the content type says 'Automatically detect significant sentiment shifts'. The main content area is titled 'Sentiment Analysis Dashboard' and subtitle 'Analyzing Reddit's sentiment evolution across major events'. It features three cards: 'AVERAGE SENTIMENT Positive' (0.23 on scale from -1 to 1), 'DATA OVERVIEW 102,615 comments' (from 2019-10-01 to 2019-11-30), and 'EVENTS ANALYZED 8 official' (plus 35 dynamically detected). Below this is a section titled 'Sentiment Timeline with Events' with a 'Detailed View' link. The bottom part of the dashboard contains a section titled 'Sentiment Analysis for r/fitness - Comments' showing 'Average Sentiment 0.239' and 'Data Points 53'. To the right is an 'About this Project' section describing the platform's purpose and methodology.

**Reddit Meltdown**

Advanced Dashboard

Subreddit  
Select Subreddit  
r/fitness

Date Range  
Start Date  
10/01/2019  
End Date  
11/30/2019

Content Type  
Comments  
 Include Dynamic Events  
Automatically detect significant sentiment shifts

**Sentiment Analysis Dashboard**  
Analyzing Reddit's sentiment evolution across major events

AVERAGE SENTIMENT  
**Positive**  
0.23 on scale from -1 to 1

DATA OVERVIEW  
**102,615** comments  
From 2019-10-01 to 2019-11-30

EVENTS ANALYZED  
**8 official**  
Plus 35 dynamically detected

**Sentiment Timeline with Events** [Detailed View](#)

**About this Project**

Reddit Meltdown analyzes sentiment trends across different subreddits, mapping them against significant events.

The platform combines data from millions of Reddit posts and comments, applying natural language processing to extract sentiment and key topics.

Sentiment Analysis for r/fitness - Comments

Average Sentiment  
0.239

Data Points  
53

## Frontend Architecture:

- Framework: Next.js (React) + TypeScript for maintainability
- **Styling:** Tailwind CSS for fast, responsive design
- **Visualization Tools:**
  - chartjs-2 for sentiment timelines
  - wordcloud for keyword displays
  - react-datepicker for time filtering
- **APIs:** RESTful endpoints from a FastAPI backend
- **Deployment:** Docker containerization for portability across systems

Reddit Meltdown Dashboard

Select Subreddit: r/antiwork

Date Range: Start Date: 10/01/2019, End Date: 11/30/2019

Content Type: Submissions

Event Sources: Include Dynamic Events (checkbox checked)

**Event Impact Analysis for r/antiwork**

Date	Event	Category	Impact	Keywords
2019-10-23	Reddit's mobile app reaches 50 million downloads	platform	+0.23	
2019-10-01	Reddit launches AITA (Am I The Asshole) community awards	platform	-0.21	
2019-11-05	Reddit introduces 'Community Points' in test subreddits	platform	-0.19	wonderful working school workers
2019-11-25	Major Reddit outage affecting multiple subreddits	platform	+0.19	work found boss credit able
2019-11-20	Cybertruck unveiling by Tesla	technology	+0.17	work quality long certifications working
2019-11-13	Reddit's 'Popular' tab algorithm changes announced	platform	-0.17	people tired company work life
2019-10-15	Reddit implements new award system sitewide	platform	+0.08	
2019-10-10	Blizzard bans Hearthstone player for supporting Hong Kong protests	gaming	+0.08	

**About Event Impact Analysis**

This analysis shows how key events correlate with changes in sentiment across r/antiwork. A positive impact score indicates a positive shift in sentiment following the event, while a negative score indicates a decrease in sentiment.

Click on any event row to view conversations from that date.



**Recent Events** 38 dynamic events found

- 2019-10-01 Reddit launches AITA (Am I The Asshole) community awards (PLATFORM) Impact: 0.28
- 2019-10-04 Negative sentiment shift detected in r/antiwork (COMMUNITY) Impact: 0.26
- 2019-10-05 Negative sentiment shift detected in r/antiwork (COMMUNITY) Impact: 0.21
- 2019-10-06 Negative sentiment shift detected in r/antiwork (COMMUNITY) Impact: 0.37
- 2019-10-07 Negative sentiment shift detected in r/antiwork (COMMUNITY) Impact: 0.21
- 2019-10-09 Positive sentiment shift detected in r/antiwork (COMMUNITY) Impact: 0.46

# Functionalities & User Experience

## Key Functionalities:

### • Filtering Module:

- Select subreddit
- Choose date range
- Toggle between Posts/Comments

### • Sentiment Timeline:

- Dynamic line chart of sentiment over time
- Tooltip support for exact score and date

### • Keyword Panel:

- Displays positive and negative keywords
- Updated dynamically via backend filters

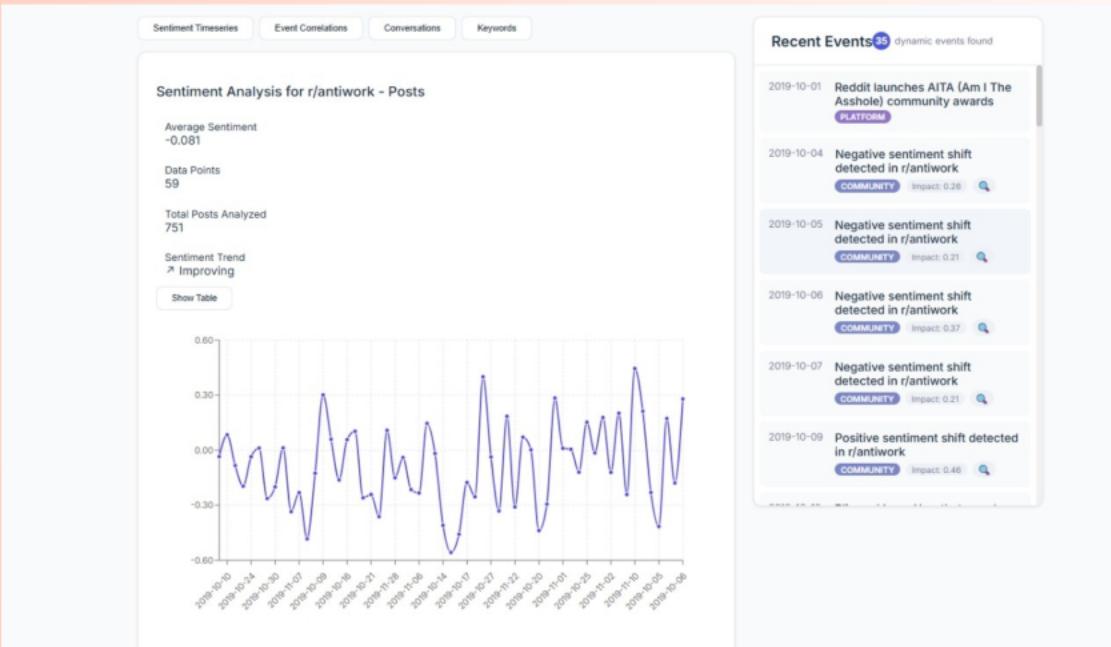
### • Event Impact Table & Drill-down:

- Shows major events with sentiment scores
- Click-through to related posts/comments with context

### • User Benefits:

- Real-time insights
- Drill-downs from macro trends to individual conversations

# Data Flow, State Management & UX



## System Integration & Data Flow:

- User selects filters in dashboard
- Frontend sends API request (via fetch/axios)
- Backend returns JSON: sentiment, events, keywords
- UI updates dynamically: charts, tables, wordclouds

## Real-Time Updates:

- React hooks manage state and trigger seamless re-renders
- Visuals stay synchronized with every input change

## Design Focus:

- Clean, clutter-free layout for readability
- Mobile-responsive design
- High performance and minimal load time
- Emphasis on user-driven discovery and clarity



# Sentiment Analysis

## Methodology & Insights

To analyze emotional tone across Reddit mental health discussions, we applied VADER (Valence Aware Dictionary and sEntiment Reasoner)—a fast, lexicon-based sentiment tool built for social media text. VADER calculates a compound score for each text, classifying it as positive, neutral, or negative.

### Key Aspects of Our Approach:

- Applied to titles and comments across subreddits like: r/antiwork, r/feminism, r/fitness, r/personalfinance, etc.
- Focused on daily sentiment averages per subreddit
- Enabled real-time analysis using Spark and distributed processing

2019-10-04 Negative sentiment shift detected in r/antiwork

COMMUNITY

Impact: 0.26



2019-10-05 Negative sentiment shift detected in r/antiwork

COMMUNITY

Impact: 0.21



2019-10-06 Negative sentiment shift detected in r/antiwork

COMMUNITY

Impact: 0.37



# Temporal Shifts & VADER Benefits

Bringing up anti work ideas makes people angry

This is the only place online or offline where I can share my feelings genuinely, and I find this really sad. I grew up before the internet was widespread. Reddit did not exist for most of my lifetime. Before the internet, the chances of finding two woke people running into each other was God damn near impossible. I wonder how long this sub is going to keep existing before someone shuts it down... In RL, I get a lot of angry responses. Other people react with disgust. "So you're just lazy/self..."

u/HierEncore 145 points Post  
Sentiment: -0.14

2019-11-29 +37 days

Radical opinion: being alive shouldn't cost you money.

I am so depressed I can barely hold my job and i'll probably lose it in the next few months. Therapist said a weekly appointment would not be enough for me so I have to wait till April where a hospital has a free space for me. Until then I have to pay for rent, electricity, food and health insurance. My savings should last that long hopefully, but after the hospital therapy it's completely unsure if the money lasts. People shouldn't have the right to tell me that I have to stay alive when the...

u/Kivijakotakou 123 points Post  
Sentiment: -0.50

2019-11-29 +37 days

## Temporal Sentiment Shifts:

- Daily tracking revealed sentiment anomalies
- Example: Early Oct 2019, r/antiwork saw consecutive drops—matching emotionally charged posts

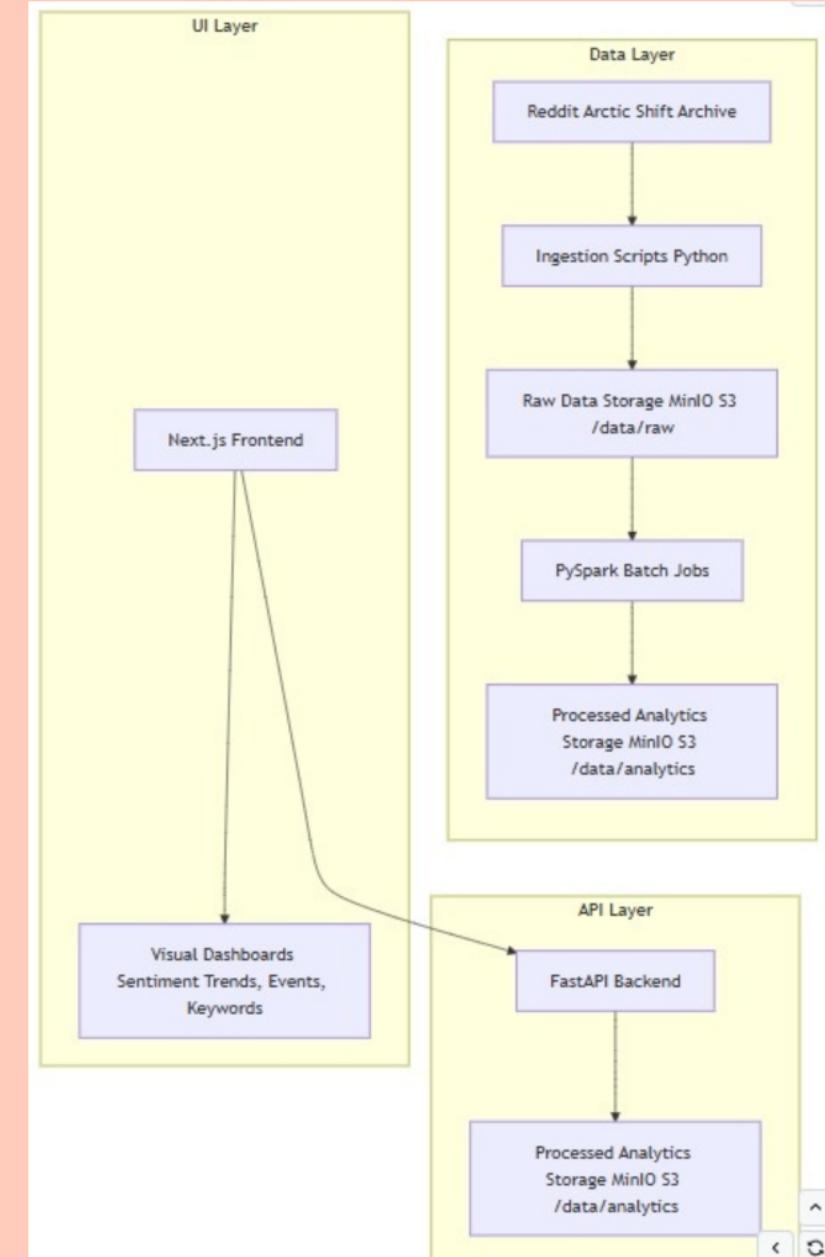
## Why VADER is a Good Fit:

- Designed for social media—understands emojis, slang, and negations
- Extremely fast and scalable
- Seamless integration with PySpark enabled processing of millions of posts efficiently

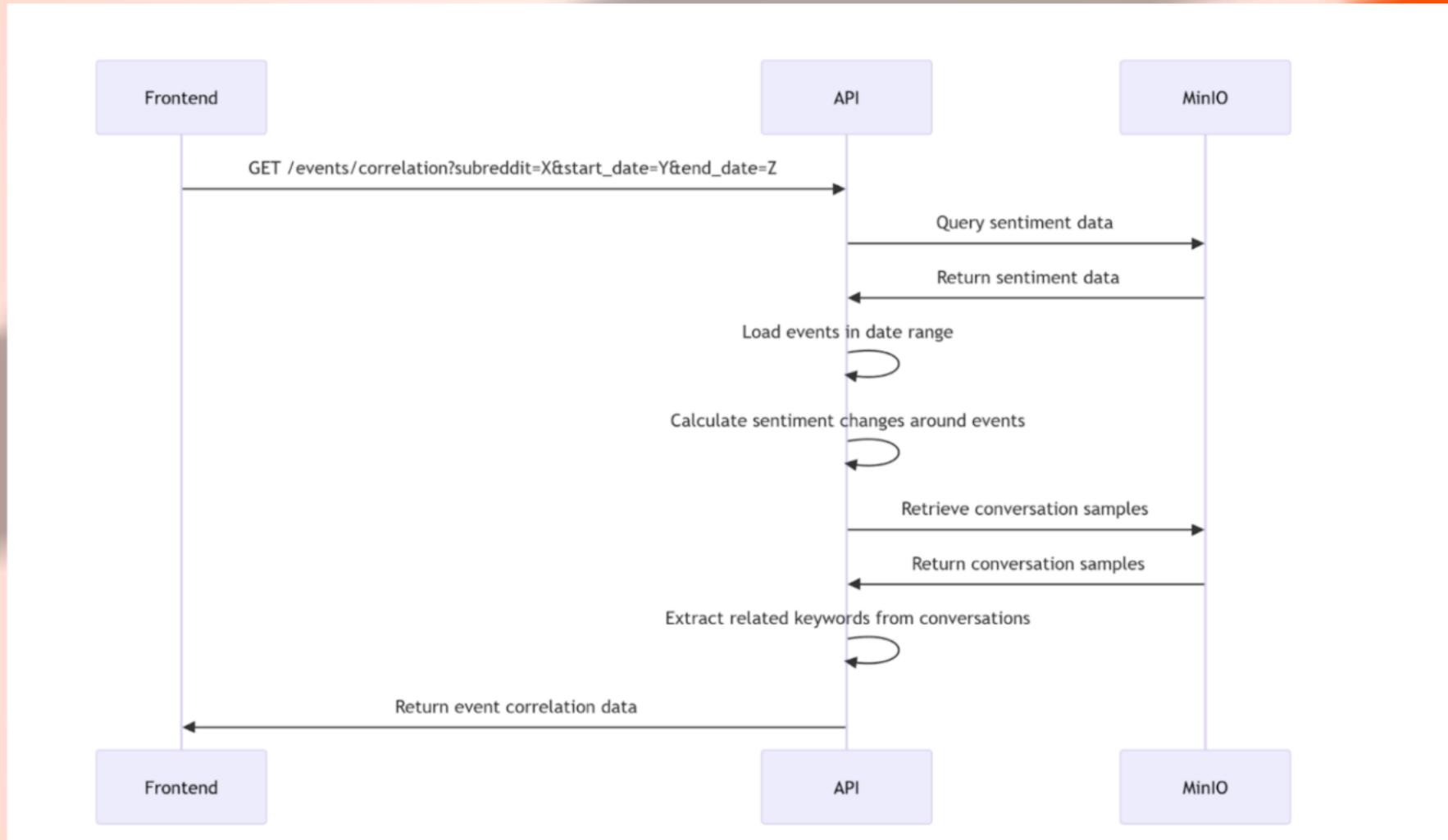
# System Design Overview

## Data Flow Summary:

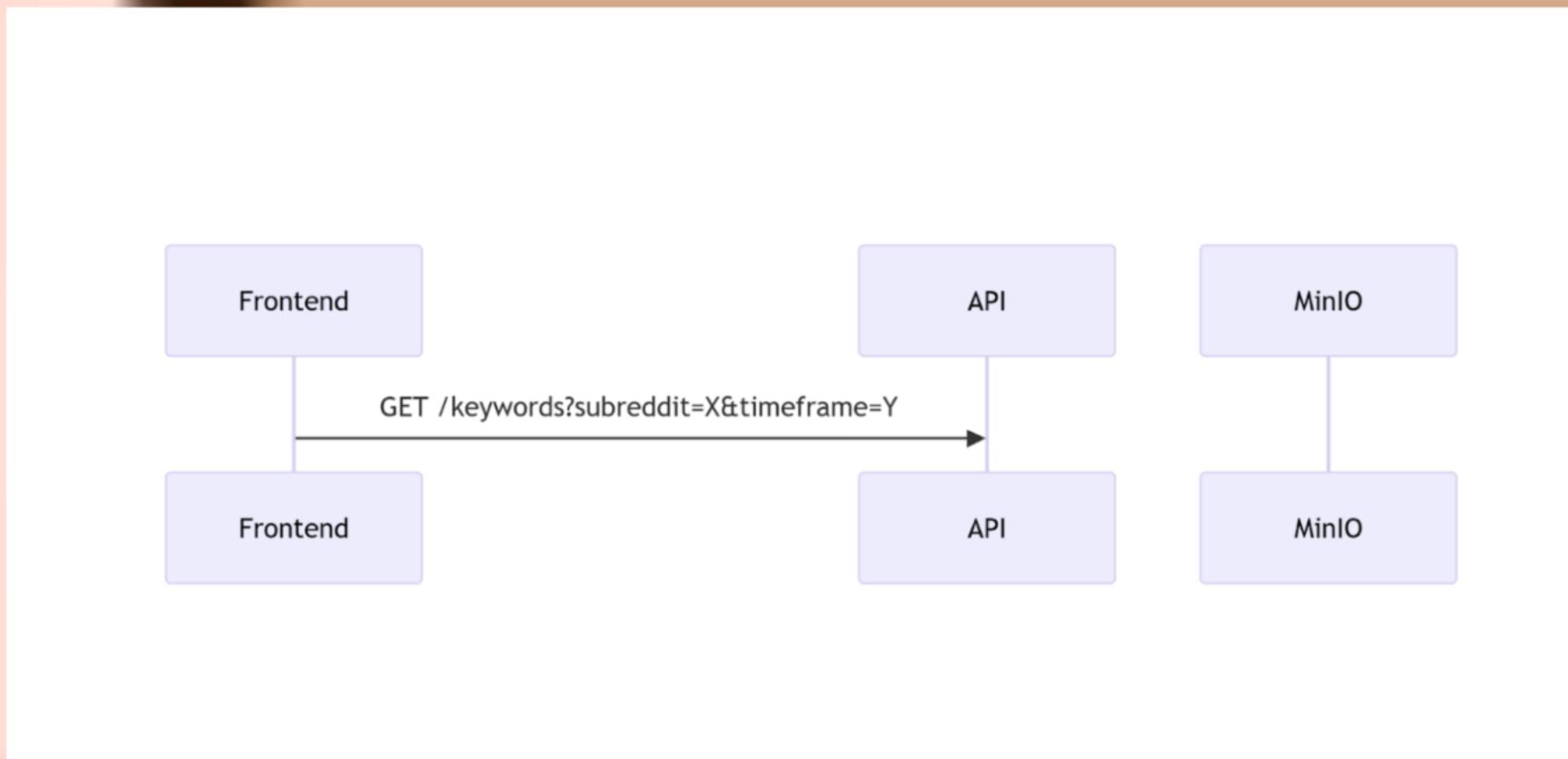
- Python scripts ingest Reddit data -> publish to MinIO
- Spark consumes, processes, writes Parquet to MinIO.
- FastAPI serves results from MinIO.
- Next.js frontend visualizes data via backend APIs.



# Event Correlation



# Keywords Analysis



# Backend

Framework: FastAPI (+ Uvicorn)

Auto-docs / validation: Swagger UI & Pydantic models

Key endpoints (GET):

**/sentiment (daily scores)    /events (external stressors)**

**/keywords (trending terms)    /events/correlation (match mood w. events)**

Domain models:

**SentimentData, EventData, KeywordData, EventCorrelationData**

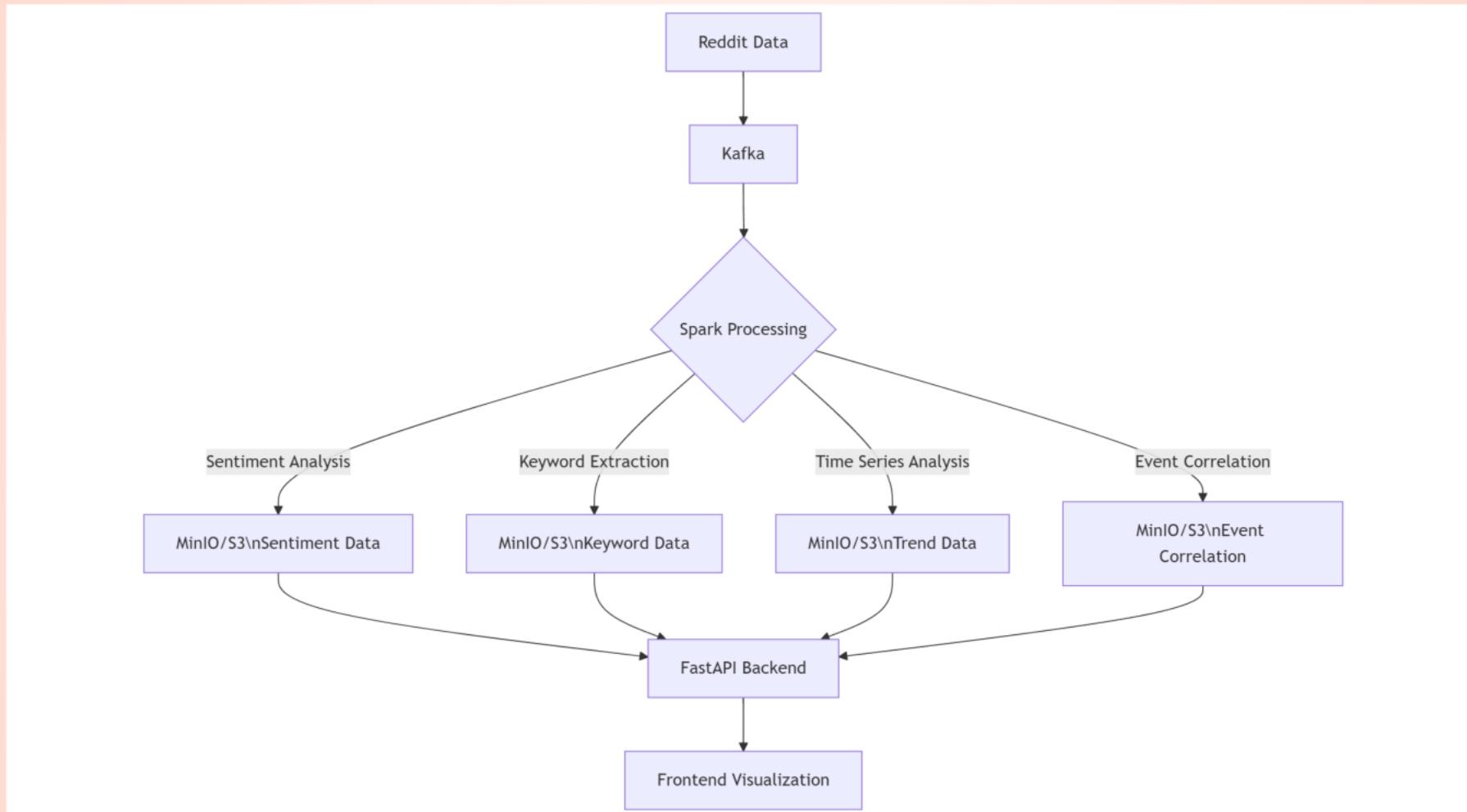
Data source:

**Parquet in MinIO S3 bucket mh-trends**

Access layer:

**boto3 reads only requested partitions**

# Pyspark Jobs



# DEMO