Real-Time User Engagement and Analytics

1. Introduction

1.1 Project Overview

The Real-Time User Engagement and Analytics project builds a scalable pipeline to ingest, process, store, and visualize social media interaction data in near real-time. It focuses on three data streams: community interactions, live streaming events, and video interactions, enabling stakeholders to monitor user engagement, platform performance, and behavioral trends.

- **Objective**: Deliver real-time insights into user engagement across social media platforms.
- **Use Case**: Analyze community engagement, live streaming metrics, and video interaction patterns to inform content strategies and user experience improvements.
- **Technologies**: Apache Kafka, Apache Spark, AWS S3, Snowflake, Metabase.
- **Outcome**: Interactive dashboards with sub-minute latency for engagement metrics.

1.2 Scope

- Ingest batch and streaming data from social media platforms.
- Clean and transform data to ensure quality and consistency.
- Store data in a dimensional model for efficient analytics.
- Visualize engagement trends and demographics.
- Ensure scalability, reliability, and fault tolerance.

2. System Architecture

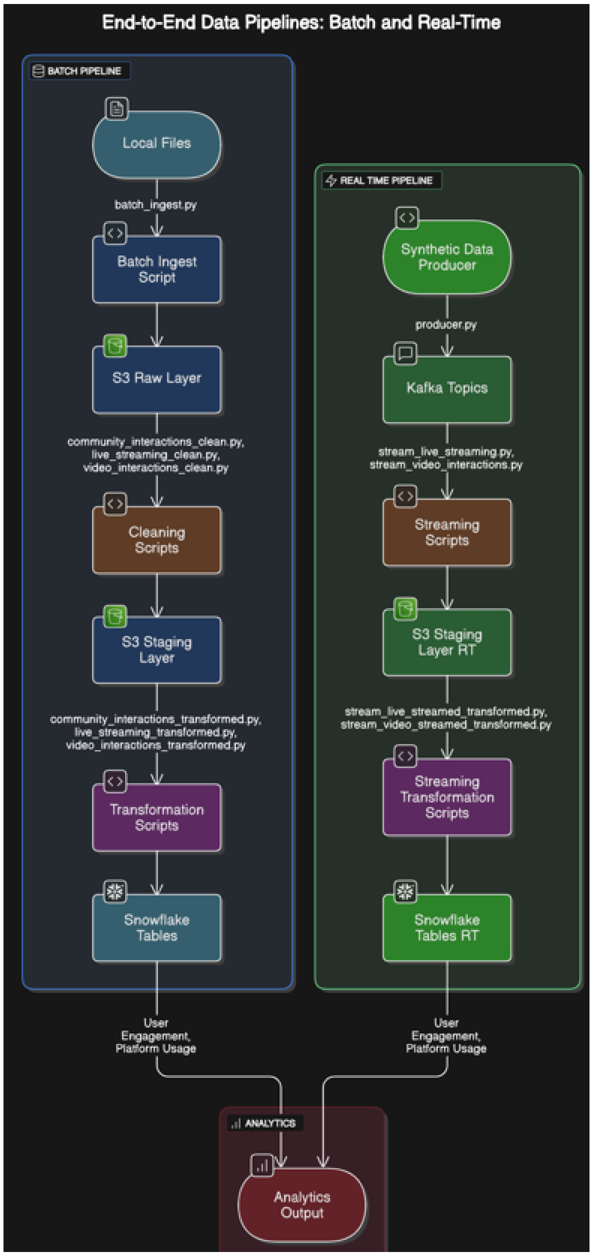
The architecture is designed for high-throughput, low-latency data processing using a layered approach.

2.1 Components

- **Data Sources**:
    - Batch: Community interactions (Parquet), live streaming (NDJSON), video interactions (CSV, NDJSON).
    - Streaming: Live streaming and video interactions via Kafka topics.
- **Apache Kafka**: Streams live streaming and video interaction data into topics (live_streaming, video_interactions).
- **AWS S3**:
    - Raw Bucket: Stores unprocessed data (s3a://datastreaming-analytics-1/raw/).
    - Staging Bucket: Holds cleaned, transformed data (s3a://datastreaming-analytics-1/staging/).
- **Apache Spark**: Processes batch and streaming data using Structured Streaming for cleaning, transformation, and deduplication.
- **Snowflake**: Cloud data warehouse for storing dimensional tables and fact tables.
- **Metabase**: Visualizes data through dashboards connected to Snowflake views.

2.2 Data Flow

1. **Ingestion**:
    a. Batch data (Parquet, CSV, NDJSON) is ingested into S3 raw bucket.
    b. Streaming data is ingested via Kafka and written to S3 raw bucket.
2. **Cleaning**:
    a. Spark cleans raw data (handles nulls, standardizes formats, validates ranges).
    b. Cleaned data is written to S3 staging bucket.
3. **Transformation**:
    a. Spark transforms staged data into a star schema (dimension and fact tables).
    b. Transformed data is loaded into Snowflake.
4. **Visualization**:
    a. Metabase queries Snowflake views to create dashboards.
    b. Dashboards auto-refresh every 60 seconds.



3. Data Pipeline Steps

3.1 Step 1: Data Collection

- **Sources**:
    - Community interactions: Parquet files with user engagement metrics.
    - Live streaming: NDJSON files and Kafka streams with event metrics.
    - Video interactions: CSV, NDJSON files, and Kafka streams with user behavior data.

- **Mechanism**:
    - Batch: Spark reads files and writes to S3 raw bucket with deduplication.
    - Streaming: Kafka producers publish to topics; Spark consumers write to S3.
- **Key Features**:
    - Fault-tolerant ingestion with Kafka replication.
    - Schema enforcement for data consistency.
- **Output**: Raw data in S3 (community_interactions, live_streaming, video_interactions).

3.2 Step 2: Data Cleaning

- **Tool**: Apache Spark (Structured Streaming for streaming data).
- **Processes**:
    - **Null Handling**: Drop rows with nulls in critical columns (e.g., CommunityID, UserID); fill non-critical nulls with defaults (e.g., Unknown, 0).
    - **Validation**: Filter invalid data (e.g., negative engagement, age outside 13–100).
    - **Standardization**: Normalize string fields (e.g., Gender to Male/Female/Other, Platform to valid platforms like Instagram).
    - **Deduplication**: Remove duplicates based on unique keys (e.g., CommunityID+UserID).
    - **Formatting**: Trim strings, capitalize platforms, and derive fields (e.g., AgeGroup).
- **Output**: Cleaned data in S3 staging bucket, partitioned by IngestionTimestamp for streaming data.

3.3 Step 3: Data Transformation

- **Tool**: Apache Spark with Snowflake connector.
- **Processes**:
    - **Dimensional Modeling**: Create dimension tables (DIM_USER, DIM_COMMUNITY, DIM_PLATFORM, etc.) and fact tables (FACT_COMMUNITY_INTERACTIONS, FACT_LIVE_STREAMING_INTERACTIONS).
    - **Surrogate Keys**: Generate MD5-based surrogate IDs for dimensions (e.g., User_S_ID).
    - **Joins**: Link fact tables to dimensions using natural keys (e.g., UserID).
    - **Deduplication**: Ensure unique InteractionID in fact tables.

    - **Type Casting**: Convert metrics to FLOAT for analytics (e.g., CommunityEngagement).
- **Output**: Structured data in Snowflake public schema.

3.4 Step 4: Data Storage

- **Tool**: Snowflake data warehouse.
- **Process**: Spark writes transformed data to Snowflake tables using JDBC connector.
- **Schema**: Star schema with fact and dimension tables (see Section 4).
- **Benefits**:
    - Optimized for analytical queries.
    - Scalable compute and storage.
    - Supports real-time updates via append/overwrite modes.
- **Output**: Query-ready data in Snowflake.

3.5 Step 5: Visualization

- **Tool**: Metabase.
- **Process**: Connect Metabase to Snowflake, query views (e.g., ENGAGEMENT_OVERVIEW), and create bar chart dashboards.
- **Dashboards**:
    - **Engagement Overview**: Engagement by platform.
    - **Community Trends**: Community engagement by community name.
    - **Live Streaming**: Live engagement by device type.
    - **Video Interactions**: Engagement by watch reason.
    - **Time Trends**: Engagement by hour.
    - **Demographics**: Engagement by age group.
- **Features**:
    - Auto-refresh every 60 seconds.
    - Interactive filters for dimensions (e.g., platform, age group).
- **Output**: Real-time dashboards with engagement insights.

3.6 Step 6: Results Discussion

- **Activities**: Analyze dashboard trends, identify high-engagement platforms, and share insights.
- **Example Insights**:
    - Peak engagement on YouTube during evening hours.

    - Higher live streaming engagement on mobile devices.
    - Productivity loss correlated with addiction levels in video interactions.
- **Outcome**: Data-driven recommendations for content optimization and user retention.

4. Dimensional Modeling

The data is organized in a star schema to optimize analytical queries in Snowflake.

4.1 Fact Tables

- **FACT_COMMUNITY_INTERACTIONS**:
    - **Purpose**: Stores community engagement metrics.
    - **Columns**:
        - InteractionID (STRING): Unique interaction ID (MD5 hash).
        - UserID_Surrogate (STRING): Links to DIM_USER.
        - CommunityID_Surrogate (STRING): Links to DIM_COMMUNITY.
        - PlatformID (STRING): Links to DIM_PLATFORM.
        - MembershipStatusID (STRING): Links to DIM_MEMBERSHIP_STATUS.
        - CommunityEngagement (FLOAT): Engagement score.
        - TotalTimeSpent (FLOAT): Time spent in community.
        - IngestionTimestamp (TIMESTAMP): Data ingestion time.
- **FACT_LIVE_STREAMING_INTERACTIONS**:
    - **Purpose**: Stores live streaming engagement metrics.
    - **Columns**:
        - InteractionID (STRING): Unique interaction ID.
        - UserID_Surrogate (STRING): Links to DIM_USER.
        - EventID_Surrogate (STRING): Links to DIM_EVENT.
        - PlatformID (STRING): Links to DIM_PLATFORM.
        - DeviceTypeID (STRING): Links to DIM_DEVICE_TYPE.
        - TimeID (STRING): Links to DIM_TIME.
        - LiveEngagement (FLOAT): Engagement score.
        - ViewerCount (FLOAT): Number of viewers.
        - AddictionLevel (FLOAT): Addiction score.
        - IngestionTimestamp (TIMESTAMP): Data ingestion time.

4.2 Dimension Tables

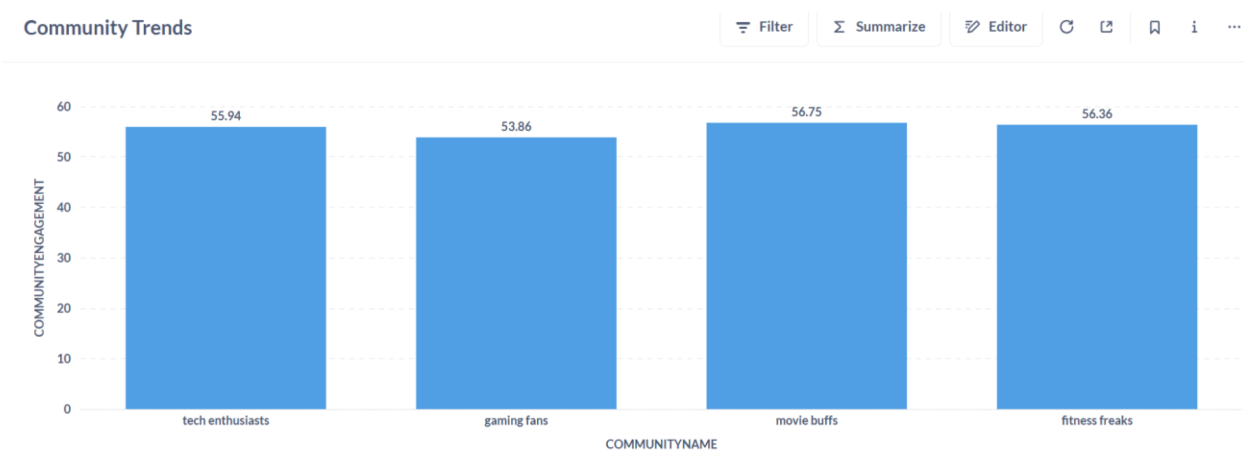
- **DIM_USER**:
  - User_S_ID (STRING): Surrogate key.
  - UserID (STRING): Natural key.
  - Age (LONG), Gender (STRING), AgeGroup (STRING), Location (STRING), etc.
- **DIM_COMMUNITY**:
  - Community_S_ID (STRING): Surrogate key.
  - CommunityID (LONG), CommunityName (STRING).
- **DIM_PLATFORM**:
  - PlatformID (STRING): Surrogate key.
  - Platform (STRING): e.g., Instagram, YouTube.
- **DIM_EVENT**:
  - Event_S_ID (STRING): Surrogate key.
  - EventID (LONG), EventType (STRING), StreamDuration (LONG).
- **DIM_DEVICE_TYPE**:
  - DeviceTypeID (STRING): Surrogate key.
  - DeviceType (STRING): e.g., Mobile, Desktop.
- **DIM_TIME**:
  - TimeID (STRING): Surrogate key.
  - WatchTime (STRING), Hour (LONG).
- **DIM_MEMBERSHIP_STATUS**:
  - MembershipStatusID (STRING): Surrogate key.
  - MembershipStatus (STRING): e.g., Member, Admin.

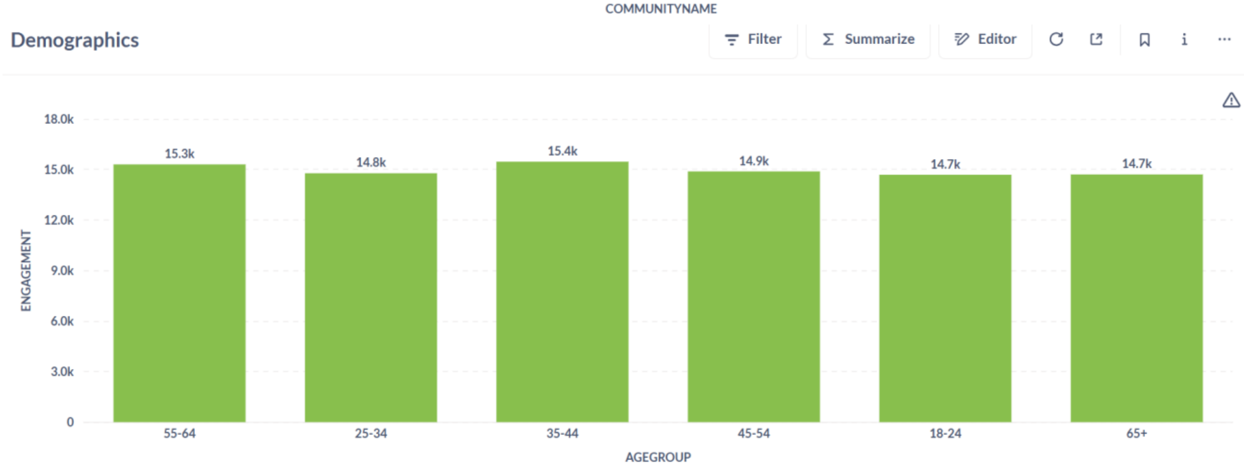
4.3 Benefits

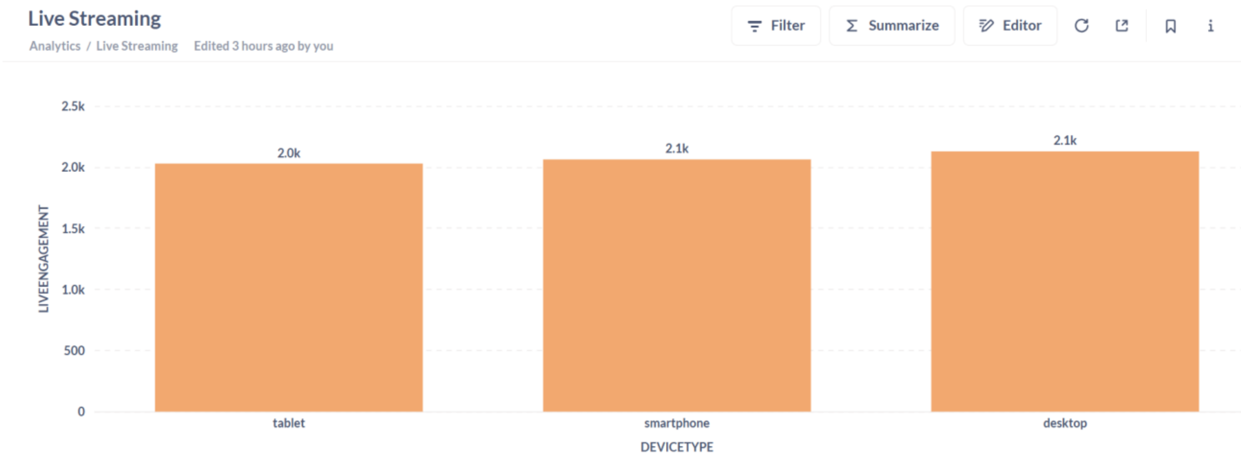
- Simplifies joins for analytical queries.
- Supports aggregations (e.g., engagement by platform).
- Enhances query performance in Snowflake.

5. Visualization and Insights

5.1 Dashboards







6. Operational Considerations

6.1 Scalability

- **Kafka**: Partition topics for parallel streaming.

- **Spark**: Adjust partitions (spark.sql.shuffle.partitions=1) and use checkpointing.
- **Snowflake**: Scale warehouse compute dynamically.
- **S3**: Partition staging data by IngestionTimestamp for efficient reads.

6.3 Challenges and Solutions

- **Challenge**: Inconsistent data formats (e.g., Gender as M/male).
    - **Solution**: Standardize fields in Spark cleaning scripts.
- **Challenge**: High streaming data velocity.
    - **Solution**: Kafka partitioning and Spark streaming with 10-second triggers.
- **Challenge**: Duplicate records.
    - **Solution**: Deduplicate using unique keys and watermarks.

7. Results and Impact

- **Achievements**:
    - End-to-end pipeline with sub-10-second latency for streaming data.
    - Six interactive dashboards for stakeholder insights.
- **Impact**:
    - Enabled real-time content strategy adjustments.
    - Reduced analysis time from hours to minutes.
    - Provided foundation for predictive analytics (e.g., engagement forecasting).

8. Future Work

- **Enhancements**:
    - Integrate machine learning for addiction prediction.
    - Add real-time alerts for engagement spikes.
    - Support additional platforms
- **Optimizations**:
    - Optimize Spark memory usage for larger streams.
    - Implement Snowflake clustering for faster queries.
    - Explore AWS Glue for metadata management.

9. Conclusion

The Real-Time User Engagement and Analytics project delivers a robust pipeline for processing and visualizing high-velocity social media data. By integrating Kafka, Spark, S3, Snowflake, and Metabase, it provides low-latency, actionable insights into user engagement, empowering stakeholders to optimize content and enhance user experiences. The scalable, reliable design ensures adaptability for future growth and advanced analytics.