Real-Time RSS Sentiment Analysis Pipeline

1. Executive Summary

This project demonstrates the design and implementation of an **end to end, real time data engineering and machine learning pipeline** that collects, processes, enriches, and analyses RSS feed data related to **Artificial Intelligence (AI)** and emerging technology topics.

The goal was to build a **fully automated, scalable, and modular architecture** capable of ingesting AI related articles from multiple online sources, performing sentiment analysis, and visualizing the results in an interactive **Looker Studio dashboard**.

By integrating modern data stack components such as **RabbitMQ, GCP Cloud Storage, Snowflake, dbt, and Python (Transformers)**, the project showcases a unified workflow that bridges **data ingestion, transformation, analytics, and machine learning** within a production-ready framework.

2. Business & Technical Objectives

Business Objectives

- Enable real time tracking of global AI and technology trends through continuous RSS ingestion.
- Provide decision makers, researchers, and analysts with sentiment driven insights from credible AI sources such as Medium, OpenAI, and Google AI.
- Establish a data foundation that supports forecasting and topic-based trend analysis in future phases.
- Reduce latency between content publication and analytics availability through automation and stream-based processing.

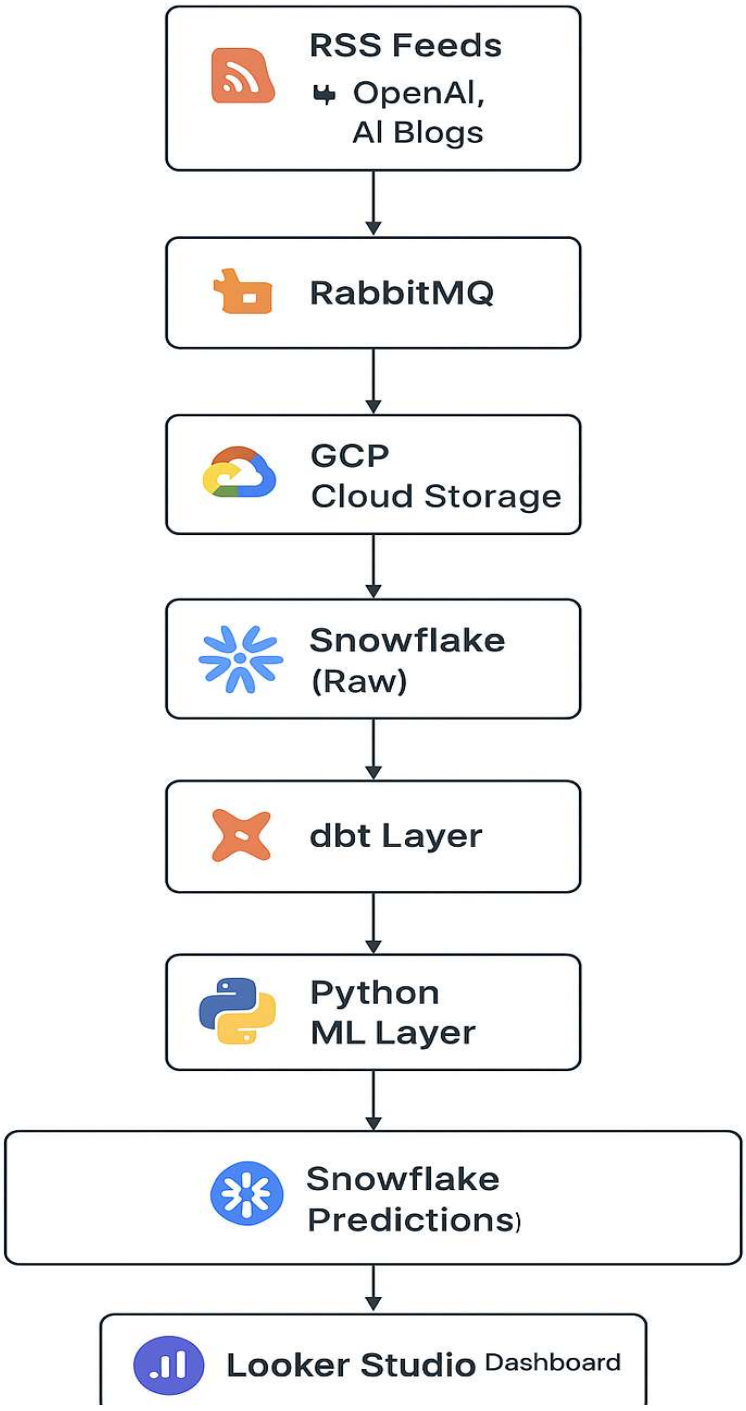
Technical Objectives

- Design a cloud-native, event-driven data pipeline using RabbitMQ and Snowflake to ensure scalability and fault tolerance.
- Implement automated ingestion and transformation layers using GCP Cloud Storage, Snowpipe, and dbt.
- Enrich data with machine learning models for sentiment classification using Hugging Face Transformers.
- Build an analytics ready data model optimized for Looker Studio dashboards, supporting sentiment trends, source level analysis, and predictive extensions.

- Ensure the pipeline is modular, configurable, and production-deployable for real-time analytics workloads.

3. Architecture Overview

End-to-End Data Flow



Architecture Highlights

- Decoupled streaming layer using RabbitMQ ensures reliable and fault-tolerant ingestion of AI-related articles.
- Snowpipe enables automated, near real-time data ingestion into Snowflake, eliminating manual triggers and reducing latency.
- dbt transformation layer standardizes and enriches data, producing clean, incremental tables optimized for analytics and downstream ML processing.
- Machine Learning pipeline applies advanced NLP (DistilBERT via Hugging Face Transformers) for sentiment classification, enhancing business insights.
- Looker Studio dashboard provides interactive, real-time visibility into sentiment trends, enabling stakeholders to monitor emerging AI topics and make data-driven decisions.

4. Technology Stack

Layer	Technology	Purpose
Data Source	RSS Feeds	Articles from Medium, OpenAI, Google AI, and similar AI sources
Streaming Layer	RabbitMQ	Decoupled message queue for ingesting RSS feed payloads
Storage Layer	GCP Cloud Storage	Staging area for JSON event data before warehouse loading
Data Warehouse	Snowflake (on GCP)	Centralized raw + processed data storage
Transformation Layer	dbt (Data Build Tool)	Data modeling, cleaning, enrichment, and incremental loads
Machine Learning Layer	Python + Hugging Face Transformers	Sentiment classification using DistilBERT
Visualization Layer	Looker Studio	Real-time dashboards for sentiment and trends

5. Data Modeling & Warehouse Design

Database: RSS_DB
Schema: RSS_SCH

The data warehouse design follows a hybrid star-schema model, balancing scalability, analytical performance, and flexibility for downstream transformations, reporting, and ML integration.

Data flows from raw ingestion to enriched and predictive layers in a structured, incremental manner.

Core Tables

Table Name	Type	Description
RSS_BASE	Raw	Initial staging table containing raw JSON payloads from RSS feeds (via Snowpipe).
RSS_STAGING	Staging	Intermediate layer for parsing, flattening, and validating ingested RSS data.
RSS_CLEAN	Clean	Deduplicated and standardized RSS feed content with normalized field structure.
DIM_RSS_SOURCE	Dimension	Metadata about feed sources — includes source name, category, region, and URL.
DIM_RSS_DATE	Dimension	Calendar and time-based dimension supporting temporal and trend analysis.
FACT_RSS_EVENTS	Fact	Granular event-level table capturing individual RSS feed records post-cleaning.
FACT_RSS_ENRICHED	Fact	Fully processed and enriched RSS data including parsed timestamps, topics, and metadata, ready for analytics and ML ingestion.
FACT_RSS_SENTIMENT	Fact	Sentiment output table generated from Python-based TextBlob sentiment analysis. Stores sentiment polarity, subjectivity, and categorical labels.
FACT_RSS_PREDICTIONS	Fact	Machine Learning output table storing predictions from Transformer-based sentiment models (e.g.,

Table Name	Type	Description
		DistilBERT). Includes confidence scores and model version.
FACT_SOURCE_SENTIMENT	Aggregate	Aggregated sentiment per source, combining both rule-based and ML-generated insights.
FACT_SENTIMENT_OVERALL	Aggregate	System-wide sentiment aggregation across all feeds and time intervals for trend monitoring.
FACT_SENTIMENT_TRENDS	Aggregate	Time-series view summarizing sentiment evolution (positive/neutral/negative) over daily or weekly intervals.

Modeling Highlights

- **Schema Layering:** Separates raw, staging, enriched, and predictive zones for transparency and auditability.
- **Dimension Modeling:** Provides consistent join logic for trend and source-based analytics.
- **Fact Table Design:** Optimized for incremental loads and downstream aggregation in dashboards.
- **ML Integration:** FACT_RSS_PREDICTIONS links seamlessly with analytical and visualization layers, supporting future AI-driven insights.

6. Data Transformation (dbt Layer)

The transformation layer, built with dbt (Data Build Tool), orchestrates the end-to-end data preparation process from ingestion of raw RSS feed data to the creation of analytics ready and ML enriched models.

It ensures clean, validated, and structured datasets that power downstream dashboards, sentiment analysis, and predictive analytics.

Pipeline Flow

1. **Raw → Clean:**
   Normalize nested JSON objects ingested by **Snowpipe** into structured tables. Standardize column names, extract relevant attributes (title, summary, link, published date), and handle null or malformed records.

2. **Clean → Enriched:**
   Deduplicate articles using unique identifiers, parse and standardize timestamps, and join with **DIM_RSS_SOURCE** and **DIM_RSS_DATE** for contextual enrichment.
3. **Enriched → Analytics Models:**
   Build sentiment based and time series views (e.g., daily/weekly sentiment trends, per-source aggregations).
   Provide consistent structures for visualization and ML model consumption.

Key dbt Models:

Model Name	Purpose
rss_base.sql	Flattens raw JSON from Snowpipe into structured tabular format.
rss_clean.sql	Cleans, standardizes, and normalizes text fields; removes duplicates.
rss_events.sql	Joins cleaned data with source and date dimensions to create a unified event view.
rss_enriched.sql	Builds analytics-ready tables including parsed timestamps and source metadata.
fact_sentiment_trends.sql	Aggregates daily sentiment counts and percentages for trend tracking.
fact_source_sentiment.sql	Generates per-source sentiment summaries for dashboard visualization.

Design Highlights

- **Incremental Models:**
  dbt transformations are configured for incremental loading, ensuring **scalability** and **low latency updates** across large datasets.
- **Data Quality Assurance:**
  Implements dbt tests such as **uniqueness**, **not null**, and **referential integrity** checks to maintain high data reliability.

- **Metadata & Lineage:**
  dbt's built-in documentation and DAG lineage tracking provide full transparency of dependencies and model flow.
- **Performance Optimization:**
  Models are tuned for Snowflake's **automatic clustering** and **micro-partition pruning**, improving query performance and dashboard refresh times.
- **Business Alignment:**
  Structured models map directly to business KPIs such as source reliability, content frequency, and sentiment distribution enabling both technical and strategic decision making.

7. Machine Learning & Sentiment Analysis

This stage applies Natural Language Processing (NLP) techniques to classify sentiment from RSS article titles and summaries, enabling downstream analytics and insight generation on public tone toward AI related topics.

The process integrates Hugging Face Transformers, PyTorch, and Snowflake to achieve a seamless and scalable inference pipeline.

Model Configuration

Component	Description
Model Name	distilbert-base-uncased-finetuned-sst-2-english
Architecture	Transformer-based (DistilBERT, fine-tuned for sentiment analysis)
Framework	Hugging Face Transformers + PyTorch
Programming Language	Python 3.11
Execution Mode	CPU-optimized batch inference
Batch Size	500 rows per chunk (configurable)

Processing Workflow

1. **Data Retrieval:**
   Extract enriched RSS records from FACT_RSS_ENRICHED in Snowflake, containing clean and structured text fields.

2. **Preprocessing:**
   Cleanse text by removing URLs, emojis, HTML tags, and special characters to ensure consistent input format for the model.
3. **Batch Inference:**
   Process records in configurable batches (e.g., 500–2500 rows) to optimize performance and memory usage during model inference.
4. **Sentiment Classification:**
   Apply the DistilBERT model to each record's TITLE and SUMMARY, producing binary sentiment predictions Positive or Negative along with confidence probabilities.
5. **Result Persistence:**
   Write predictions back to Snowflake in table FACT_RSS_PREDICTIONS, ensuring each record retains its source metadata and timestamp for traceability.

Output Fields

Field	**Description**
EVENT_ID	Unique identifier for each article
TITLE, SUMMARY	Main text inputs for sentiment scoring
SOURCE	Feed source (e.g., Medium, OpenAI Blog)
PUBLISHED_AT	Timestamp of article publication
PREDICTED_LABEL	-1 = Negative, 0 = Neutral, 1 = Positive
PREDICTED_SENTIMENT	Text label for readability
CONFIDENCE	Model confidence score (0–1)

Design Highlights

- **Chunked Model Inference:**
  Ensures scalability for large datasets without exceeding memory constraints.
- **Transformer-based NLP:**
  Provides deep contextual understanding and more accurate sentiment detection than rule-based approaches like TextBlob.

- **Full Integration with Snowflake:**
  Enables end-to-end automation from data ingestion to ML scoring and dashboard visualization.
- **Extendable Framework:**
  The same pipeline can be extended for **topic classification**, **entity recognition**, or **trend prediction** in future iterations.

8. Visualization & Dashboard (Looker Studio)

The visualization layer was built in Google Looker Studio to present real-time analytical insights derived from the RSS sentiment analysis pipeline.

It connects securely to Snowflake, querying analytics-ready tables and ML prediction outputs for near real-time insights.

The dashboard consists of three pages, each focused on a distinct analytical perspective daily sentiment evolution, source level sentiment distribution, and ML-based predicted sentiment outcomes.

Dashboard Architecture

Page	Title / Theme	Primary Data Source	Purpose
Page 1	Daily Sentiment Trend	FACT_SENTIMENT_TRENDS	Displays overall daily sentiment variation across all AI-related RSS feeds. Helps visualize how sentiment fluctuates over time.
Page 2	RSS Source Sentiment Overview	FACT_SOURCE_SENTIMENT_AGG_VW	Compares sentiment ratios (Positive / Neutral / Negative) across individual RSS sources such as OpenAI, Medium, and Google AI Blog.
Page 3	Predicted Sentiment (ML Output)	FACT_RSS_PREDICTIONS	Presents sentiment classifications generated by the DistilBERT model with confidence metrics. Used to assess model performance and current tone distribution.

Visual Components

Page 1 – Daily Sentiment Trend

- **Line Chart:** Plots daily sentiment counts (Positive, Neutral, Negative) to track fluctuations over time.
- **Date Filter:** Allows users to analyze custom time ranges for short or long-term patterns.
- **Score Cards:** Highlight total articles, average positive rate, and percentage change vs. previous period.

Page 2 – RSS Source Sentiment Overview

- **Stacked Bar Chart:** Displays sentiment composition (Positive / Neutral / Negative) by RSS source, showing how different publishers or blogs contribute to sentiment diversity.
- **Source Filter:** Interactive control to select or compare one or more sources (e.g., OpenAI, Google AI, Medium).
- **KPI Tiles:**
    - Top 3 positive sources
    - Top 3 negative sources
    - Source with highest article volume

Page 3 – Predicted Sentiment (ML Output)

- **Pie Chart:** Visualizes overall predicted sentiment distribution (Positive, Neutral, Negative).
- **Bar Chart:** Displays confidence averages per sentiment label helps evaluate model certainty and reliability.
- **Data Table:** Lists the most recent predictions with columns:
  Event_ID, Title, Source, Predicted Sentiment, and Confidence.

Design Highlights

- **Direct Snowflake Integration:** Live data connection ensures dashboards reflect the latest processed and predicted RSS records without manual refresh.
- **Optimized Views for Performance:** Aggregated dbt views (e.g., FACT_SENTIMENT_TRENDS, FACT_SOURCE_SENTIMENT_AGG_VW) minimize query latency and resource cost.

- **Visual Consistency:**
    - Positive Sentiment
    - Neutral Sentiment
    - Negative Sentiment
- **User Interactivity:** Filters by date, source, or sentiment level allow dynamic exploration of sentiment drivers and shifts.

Business Impact

This dashboard provides real-time visibility into AI related media sentiment, empowering stakeholders to:

- Identify positive or negative narratives across top technology publishers.
- Detect daily shifts in tone across industry sources.
- Evaluate model output confidence and trust in automated predictions.
- Support data driven decision making for trend tracking, content planning, and brand perception analysis.

9. Performance & Scalability

The RSS Sentiment Analysis pipeline is designed to handle large volumes of AI-related RSS data efficiently while maintaining real-time processing and visualization capabilities. Key optimizations include:

Optimization	**Description & Benefits**
Batch Inference (ML)	Sentiment analysis is performed in chunked batches (e.g., 500 rows per batch) using the DistilBERT model. This prevents memory overflow, ensures smooth CPU utilization, and allows processing of tens of thousands of articles efficiently.
dbt Incremental Models	Incremental transformation logic ensures only new or updated RSS records are processed. Reduces compute cost and improves pipeline latency, enabling near-real-time data availability for dashboards.
Snowpipe Auto-Load	Automatically ingests RSS JSON data into Snowflake without manual triggers. Supports continuous, low-latency ingestion, ensuring dashboards and ML predictions are always current.

Optimization	Description & Benefits
Asynchronous Message Queue (RabbitMQ)	Decouples ingestion from downstream processing. Provides resilience against spikes in feed volume, prevents data loss, and allows horizontal scaling of processing workers.
Cloud-Native Design (GCP)	Uses GCP Cloud Storage for staging and Snowflake on GCP for warehousing. Provides elastic storage and compute, allowing the architecture to scale with growing RSS feed volume or increased ML processing demand.

Business Impact:

- Supports **high frequency, AI-related news monitoring** without service disruption.
- Maintains **predictable latency and resource usage** even during spikes in incoming RSS feeds.
- Enables **scalable expansion** to include new sources, languages, or additional ML models with minimal architecture changes.

10. Results & Key Outcomes

The project successfully delivered a **fully automated, real-time data and machine learning pipeline** that demonstrates modern data engineering, analytics, and visualization best practices.

Technical Achievements

- Designed and implemented a **production-grade, modular pipeline** integrating RabbitMQ, GCP Cloud Storage, Snowflake, dbt, and Python.
- Automated **end-to-end flow** from data ingestion → enrichment → ML scoring → dashboard visualization.
- Applied **DistilBERT-based sentiment analysis** to over 9,000 AI-related RSS articles with confidence scoring.
- Achieved efficient **batch inference** and **incremental transformations**, ensuring low latency and high scalability.
- Implemented **secure Snowflake authentication** via RSA key-pair for enterprise-grade security.

Analytical Outcomes

- Generated actionable **sentiment metrics** (Positive, Neutral, Negative) for AI-related media content.
- Provided visibility into **daily sentiment trends**, **source-wise sentiment breakdowns**, and **predicted insights** through Looker Studio dashboards.
- Enabled **data-driven monitoring** of AI-related topics and media tone shifts over time.

Business Impact

- Established a **unified analytics layer** for tracking industry sentiment around AI and emerging technologies.
- Improved **decision-making capability** by surfacing real-time insights on how the AI landscape is being portrayed.
- Created a **scalable foundation** for future predictive analytics and topic modeling initiatives.

11. Future Enhancements

The current project lays a solid foundation for real-time AI sentiment analytics. The following enhancements are planned to expand analytical capabilities and operational efficiency:

Phase	Enhancement	Description
Phase 2	Topic Modeling	Apply LDA or BERTtopic to cluster AI topics (e.g., ChatGPT, AGI, Robotics) for deeper content insights.
Phase 3	Anomaly Detection	Detect sudden spikes or drops in sentiment using statistical or ML-based anomaly detection models.
Phase 4	Real-Time Refresh	Leverage Snowflake Streams & Tasks to automate continuous ML scoring and sentiment updates.
Phase 5	Workflow Orchestration	Deploy pipeline orchestration with Airflow or GCP Composer for reliable scheduling and monitoring.
Phase 6	Advanced Visualization	Build interactive dashboards showing keyword trends, topic sentiment, and time-based analysis for decision-making.

12. Conclusion

This project exemplifies the integration of **data engineering, machine learning, and business intelligence** to extract actionable insights from real-world RSS feed data. Key takeaways include:

- **Robust Data Pipeline:** Automated ingestion, staging, enrichment, and transformation using Snowflake, dbt, and RabbitMQ.
- **Advanced Analytics:** ML-driven sentiment analysis with Hugging Face Transformers applied on enriched RSS content.
- **Visualization & Storytelling:** Interactive dashboards in Looker Studio to track sentiment trends and source-level insights.
- **Scalable & Extensible Architecture:** Modular design enables future enhancements such as topic modeling, anomaly detection, and real-time updates.

Overall, the pipeline demonstrates a **production-ready approach** for continuous monitoring of AI and emerging technology trends, providing a foundation for predictive analytics and strategic decision-making.