Team Project Phase 3 - Team 7

Objective: To develop a unified, automated dashboard that consolidates real-time stock market data for user-selected stocks, updating daily after market close. The dashboard will incorporate sentiment analysis to classify the latest news as positive, negative, or neutral, enabling investors to make more informed decisions. Additionally, it leveraged Generative AI (GenAI) to summarize and analyze critical market information, including SEC filings, providing users with actionable insights and a comprehensive overview of market trends.

Phase 1 Overview: From Phase 1, we developed a unified dashboard for tracking the top 10 U.S. stocks, automatically updating daily after market close. The data pipeline was designed to ingest raw data from the Yahoo Finance API, process it into parquet files, and load it into staging tables using Cloud Run functions. These were then transformed and upserted into a cloud data warehouse for reporting via Superset. We integrated Prefect to automate the pipeline, ensuring reliable and scalable daily updates. Additionally, we tested an analytics pipeline that used web scraping and GPT-4 for summarizing news articles, providing actionable insights on stock performance. Our development practices were streamlined through GitHub, allowing efficient version control and task management.

Phase 2 Overview - Sentiment Analysis

Serverless ML (Previous)	Serverless ML (Current)
Custom-trained model (TF-IDF + Logistic Regression) created using serverless functions.	Uses a pretrained Hugging Face Transformer model for sentiment analysis - Dstilbert-Base-Uncased-Finetuned-sst-2-english
Includes a training process with a serverless cloud function using Kaggle dataset and “title” column	No training involved; leverages Hugging Face's pretrained model, uses the summary column
"Positive", "Negative", "Neutral" based on custom mapping of predictions.	"Positive", "Negative" with confidence scores.

Pipeline Design: In **Phase 3**, we expanded the pipeline further by introducing new flows to enhance functionality:

1. **Flow 1: Stock Data Processing**
    - **yfinance_dump**: Fetches stock data from Yahoo Finance, applies initial transformations, and uploads it as Parquet files to GCS.
    - **staging**: Loads Parquet files into MotherDuck and applies additional transformations. The data is then inserted into the warehouse’s primary tables.
2. **Flow 2: News Processing Flow**
    - **scrape_news**: Scrapes news articles and stores them in a dedicated table in MotherDuck.
    - **summarize_news**: Summarizes the scraped news articles using OpenAI’s APIs. Sentiment analysis is also applied at this stage (not integrated with Prefect).

    - **news_final**: Produces a cleaned and structured table of summarized articles enriched with sentiment insights.
3. **Flow 3: Podcast and Email Flow**
    - **daily_news_report**: Generates a consolidated daily report of summarized news articles by accessing data in MotherDuck.
    - **podcast_and_email**: Uses OpenAI's text-to-speech capabilities to create podcasts from the daily reports and sends them via email using API integrations.

News Processing Flow: The News Processing Flow enriches our pipeline by extracting and summarizing news articles related to the stocks in our dataset. The first stage, **scrape_news**, uses Python libraries like BeautifulSoup to collect articles from various sources, storing them in a structured format within a dedicated table in MotherDuck for easy querying. In the **summarize_news** stage, OpenAI's APIs are leveraged to condense articles into shorter summaries, highlighting key information. Sentiment analysis, performed using Hugging Face's DistilBERT model, classifies each article as positive or negative. The processed summaries, enriched with sentiment data, are stored in the **news_final** table, forming the basis for further reporting and analysis. This flow automates the processing of large volumes of news data, providing a scalable solution for tracking real-time sentiment and trends and enabling users to quickly access actionable insights.

Podcast and Email Flow: The Podcast and Email Flow enhances user engagement by delivering actionable insights in both written and audio formats. In the **daily_news_report** stage, summarized news articles from the **news_final** table are consolidated into a comprehensive daily report that highlights key developments and potential impacts on stock performance. The **podcast_and_email** function converts the report into an audio podcast using OpenAI's text-to-speech technology and automatically distributes it via email. This multi-modal delivery caters to diverse user preferences, offering insights in accessible formats for those on the go or who prefer listening over reading.

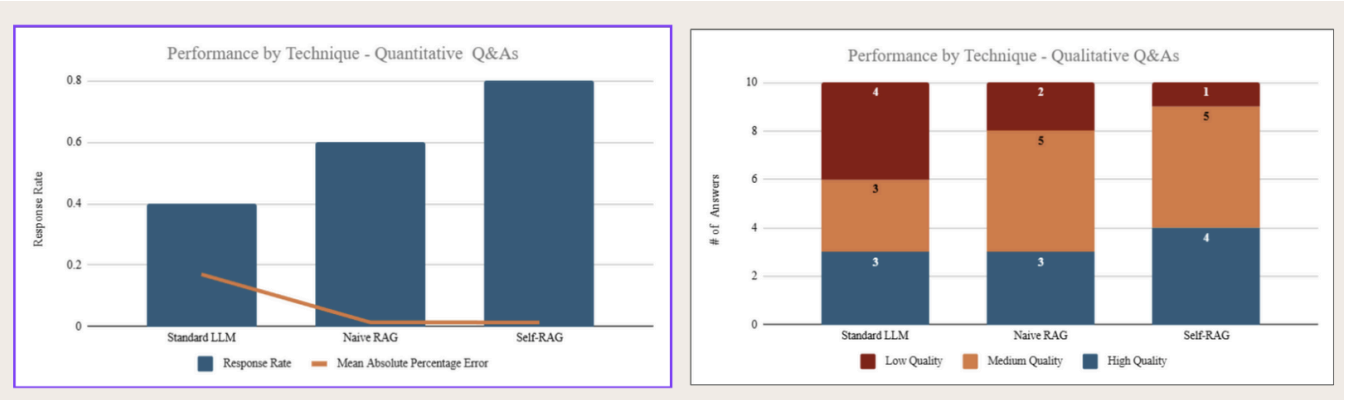
Data Product - Newsletter: The Newsletter workflow begins with essential configuration steps where the system retrieves secret tokens for both MotherDuck database access and OpenAI API authentication. After securing these credentials, the pipeline connects to the MotherDuck database to access the 'stocks.report.news_final' table.

The data processing flow continues through several stages: first retrieving all news articles from the MotherDuck database, then filtering them to include only the recent 3 days of news. These news articles are systematically grouped by their stock tickers for focused processing. For the summarization phase, we chose OpenAI's GPT-4o model over Gemini due to its better performance in handling financial content. The system processes each ticker group by concatenating article titles and summaries into a consolidated input. Using crafted prompts, OpenAI generates concise 2-3 paragraph summaries for each stock. These raw summaries then undergo a second OpenAI processing step to create optimized scripts for audio content.

The final outputs take two forms: generated raw summaries converted to script and HTML format after creating the prompt - HTML for written newsletters and audio content for podcasts, which are created

using Eleven Labs' text-to-voice technology to transform the generated scripts into professional-quality audio. These are distributed to investors daily through newsletters and podcasts. This five-minute market recap helps investors efficiently stay informed about their preferred stocks. Through the combination of summarization and dual-format delivery (newsletter and podcast), investors can quickly absorb the most significant events affecting their portfolio stocks. This streamlined approach transforms comprehensive in-depth financial news into digestible insights, making it easier for busy investors to stay on top of market movements without spending hours reading multiple news sources.

Data Product - SEC RAGbot: We create a RAG-based Chatbot with access to 10K and 10Q reports, benchmarked different RAG methods. Excel files of the reports were extracted using the EDGAR sec api, chunked at the table level, embedded using OpenAI's large embedding model, and finally upserted to Pinecone. We found that a more advanced Self-RAG with document grading and prompt optimization performed best in our benchmark.



Development Practices: Throughout our project, we used GitHub for version control and project management. We established a repository management structure to track tasks, assign responsibilities, and monitor progress, where one team member serves as the primary repository administrator, responsible for reviewing and managing merge requests to avoid overlapping changes.

Link: https://github.com/tangyum/BA-882-Pipeline-Project

Conclusions:

In conclusion, this project successfully developed a scalable and automated data pipeline that integrates stock market data and news articles, providing users with enriched datasets and actionable insights. By incorporating multi-modal user engagement through daily reports, podcasts, and a chatbot powered by Retrieval-Augmented Generation (RAG) for SEC filings, we enhanced the accessibility and usability of the platform. The use of Prefect ensured seamless orchestration and automation, enabling reliable, timely updates while minimizing manual effort. Despite challenges in integrating custom machine learning models, pivoting to serverless solutions allowed us to overcome these obstacles and deliver a robust system. With its practical applications and future potential for incorporating advanced language models and interactive features, this project demonstrates the transformative power of combining data science, automation, and generative AI to create meaningful solutions in financial analytics.