Predicting Live Stock Prices with Sentiment Analysis via Azure Stack

Team 6: Rajdeep Singh, Aidan Clark, Amruth Devineni

Introduction

Predicting stock price movements is a complex challenge faced by all investors. The volatile and fast-paced nature of the financial market increases the difficulty of this task. Traditional forecasting methods tend to rely on historical price data and technical indicators, often neglecting the influence of market sentiment. Recent research has shown that there is a correlation between sentiment from news articles and stock trends. However, the majority of this research has been focused on large stable stocks, leaving the other segments of the market unexplored.

Our project aims to fill this gap by using sentiment analysis to analyze the relationship between sentiment and stock movement across a diverse selection of stocks. Specifically, we analyzed two stable stocks (AAPL, INTC), two volatile stocks (NVDA, TSLA), and one penny stock (LAZR). This diverse selection allowed us to explore how sentiment impacts stocks with different market dynamics. Using Big Data methodologies and machine learning models, we processed and analyzed large columns of batch and streaming data. Our project's goal was to identify whether sentiment affects stock prices across different categories, and if so, to what extent.

Methodology

1. Selection of Stocks for Analysis

A key aspect of our project is to analyze how sentiment impacts stocks with varying levels of stability. For our stable stocks, we chose Apple (AAPL) and Intel (INTC). Apple has long been considered the staple stable stock and is a reliable choice for investors. Similarly, Intel has been a well-performing and trusted option, particularly for beginner investors. Despite recent declines in its stock price, Intel has maintained its reputation as a go-to stock.

For our volatile stocks, we selected Tesla (TSLA) and NVIDIA (NVDA). Since the pandemic, Tesla has demonstrated significant growth which is largely due to the acceptance and integration of electric vehicles. In the past year alone, Tesla’s stock price has increased by 86.56%. Similarly, NVIDIA has shown significant increases in stock price due to its performance in GPUs for AI and gaming. In the past year, NVIDIA’s stock price has increased by 162.86%.

Lastly, we chose Luminar Technologies (LAZR) as our penny stock. Although LAZR is not a traditional penny stock, with its price hovering around $6 following a reverse stock split in late November, it is relatively lesser-known compared to the other companies in our analysis. Luminar specializes in LiDAR sensors, a technology that has the potential for significant growth as autonomous vehicles gain popularity. While many self-driving cars, such as Tesla’s, rely on cameras, companies like Honda, Mercedes, and Waymo favor LiDAR technology, making LAZR an interesting choice for our analysis.

2. Data Gathering and API Selection

For the data gathering process, we initially planned to use the Yahoo Finance API to collect historical stock price data as our batch data and stream real-time data to validate our models and analysis. However, upon inspection, it became clear that the API access for Yahoo Finance was no longer available, and the data could only be downloaded in CSV format. Our team attempted to work around this limitation by creating a pipeline that would access an API scraping bot for live Yahoo Finance data, but this approach was unsuccessful as the API endpoint was outdated.

In addition to news sentiment, our project initially aimed to analyze the impact of social media on stock prices. Elon Musk's tweets about DogeCoin and WallStreetBets posts on GameStop and AMC have shown that social media platforms can significantly influence market price trends. When we submitted the project proposal we were under the impression that we could use the Twitter API (now X API) to gather social media data. However, we found that the current X API follows a subscription model. The free tier would not provide sufficient data for our needs, and the cost of the basic tier ($200 per month) was not a feasible option given our project's limited resources.

We ingested our data through an official partner with the Nasdaq known as Alpha Vantage. Their free-tier APIs provide raw media sentiment analysis data on a multitude of stocks, as well as stock price data, both real-time and batch. The raw news data compiles a list of articles that mention a specific stock. Alongside this data, they provide a sentiment analysis of the whole article, elements specifically mentioning the stock, and a relevance score of the stock to the full source. The historical stock ticker data provides the full stock history with custom date ranges. They provide the stock's open, close, high, and low prices. The real-time data looks similar, except it can be queried on an intraday level, allowing very specific user customization to the time intervals where data is sent. Since we were restricted by the free tier, our data was not instant, but delayed by a full day, though it was still processed and thus simulated as a stream for the purposes of this project. The training and testing of the AI model reflected this limitation and will be expanded upon later in the paper. Additionally, all previous and future mentions of "real-time" are assuming this limitation.

3. Data Architecture

Our data architecture was designed to handle historical and real-time stock tickers alongside batch historical sentiment data. The architecture employs a hybrid approach, combining the structured, multi-layered Medallion Architecture for batch data with an event-driven pipeline for real-time streaming data. This design ensures the system can manage large volumes of data efficiently while providing timely insights for decision-making. The following diagram offers a high-level view of the overall architecture.

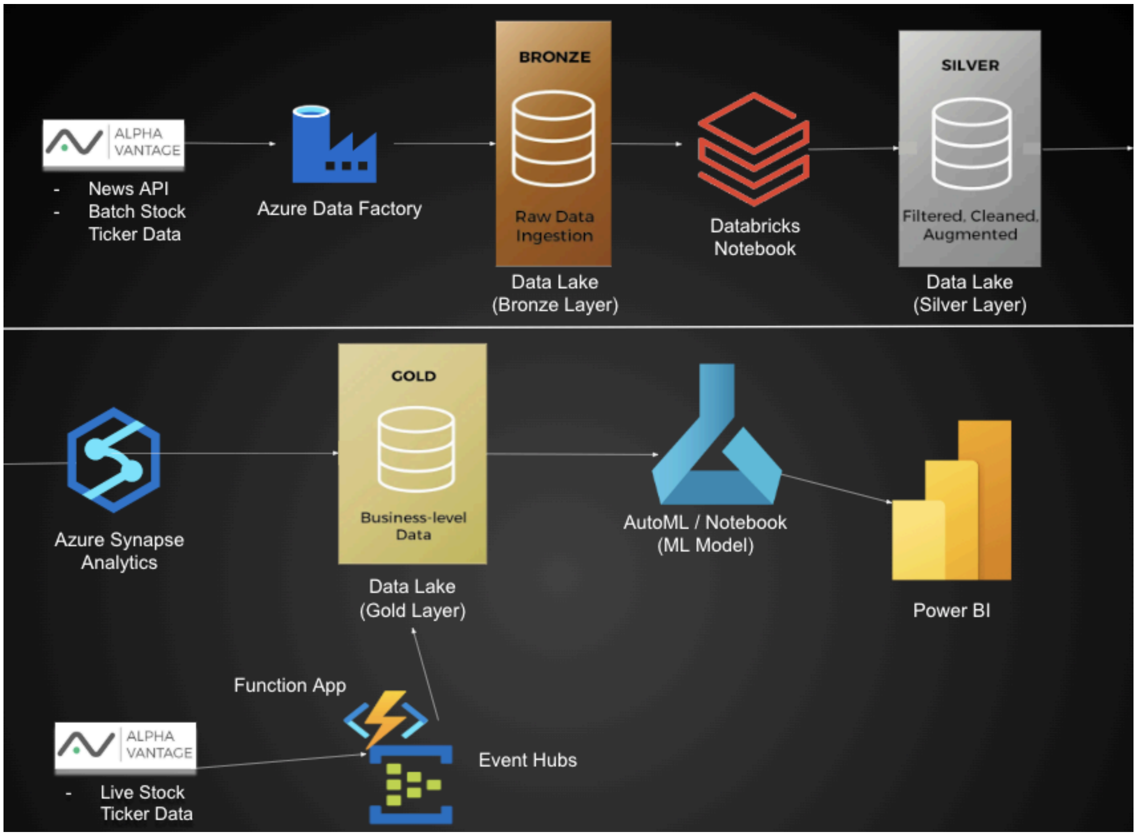


Figure 1: High-Level Data Architecture Diagram

3.1 Architectural Workflow Explained

The historical data pipeline begins by ingesting raw stock ticker and sentiment data into the Bronze Layer via Azure Data Factory. API limitations required stock ticker data to be ingested one day at a time, and required separate “copy data” activities for each stock. To address these constraints, the pipeline was scheduled to run daily at the same time, ensuring consistent ingestion and eliminating potential human error. This automated process preserved the raw format of the data, enabling reprocessing when necessary, and lowering costs by reducing the number of redundant API calls.

Once ingested, the formatting of the data was transformed using *pyspark.sql.functions*, as well as the removal of duplicate and null values to maintain data integrity for ML and reporting. This ensured the Silver Layer provided clean, structured data suitable for downstream processing.

We curated the historical data into a business-ready format. External tables for each of the stocks were created with the correct data types for reporting, and these were combined into a unified table within Synapse Analytics. A data flow was used to export the unified table to the Gold Data Lake storage container. From here, our data was used to train ML models in Azure ML Notebooks. Various models were tested, leveraging the curated data to gain insights into the relationship between sentiment and stock performance. The results of the models were exported to Power BI, where the accuracy of the models was visualized alongside other business metrics.

In parallel, the system managed real-time data through a pipeline. Live stock ticker data was ingested through Azure Event Hubs and processed by an Azure Function App. The processed data bypassed the Bronze and Silver Layers, as it was clean and structured upon ingestion. Like our batch data, it was immediately imported into Azure ML for real-time predictions, and subsequently Power BI, making it easy to visualize insights on all aspects of our project.

This hybrid approach effectively combines the structured progression of the Medallion Architecture for batch processing with the immediacy of real-time pipelines. By leveraging Azure tools like Data Factory, Synapse Analytics, and Azure ML alongside Power BI, the architecture delivered actionable insights derived from both historical and real-time trends.

4. Data Governance

Maintaining a strong data architecture throughout the project ensured the system was able to meet diverse analytical needs, a necessity for our use case as our limitations continuously presented blockers and required pivots to ensure a successful outcome for our project. Some errors experienced in the process of moving data required full overhauls of whole datasets. This included missing columns, data conversion issues, and necessity for a larger sample set that required changes to previous layers to propagate high-quality business-ready data to the Gold Layer. Through our architecture, it was much easier to make these retroactive changes, as there was a clear lineage of data with strong naming conventions for easy access by any team member whenever needed.

5. Exploratory Data Analysis

After successfully pulling batch data, the next step was to process and analyze the data. We started by flattening the nested structure of the data to make it more manageable for analysis. After flattening the data, we filtered the columns to ensure we only had what we deemed necessary for our analysis. This processed and cleaned data was exported to the silver layer of our medallion architecture.

For our EDA, we generated a total of 31 graphs, divided into 7 distinct types. Six of these graph types had five iterations - one for each of the stocks. The seventh graph was a cumulative visualization. Below is a detailed explanation of the cumulative visualization, Daily Average Sentiment Score per Ticker, as well as a brief overview of the other graph types and their importance.

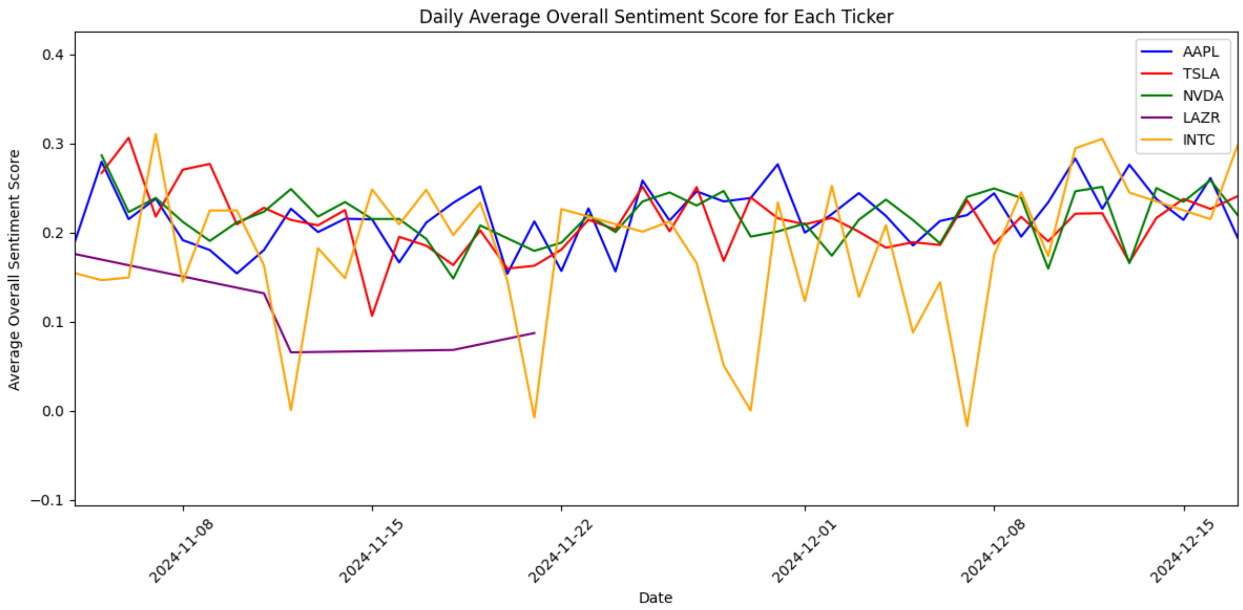


Figure 2: Graph depicting sentiment changes for the chosen stocks over time

The Daily Average Overall Sentiment Score for Each Ticker highlights trends and variability in sentiment across the five stocks over time.

- Intel (INTC) demonstrated significant variance in sentiment scores, which aligns with its sporadic stock performance over the past month.
- Tesla (TSLA) and NVIDIA (NVDA) exhibited the tightest and most consistent sentiment trends, reflecting their strong performance in the stock market and positive news coverage surrounding them.
- Apple (AAPL) had moderate variance in sentiment scores, this performance can be attributed to the mixed reviews the recent iPhone 16 launch had.
- Luminar (LAZR) had data only up till November 20th because that was the last time this stock was mentioned in the news. This limitation emphasizes its penny stock nature.

Beyond this graph, the following additional visualizations were generated to explore sentiment and validate our data:

1. **Sentiment Histograms**: These visualizations allow us to verify the distribution of positive, negative, and neutral sentiments in the news. These visualizations also allowed us to verify that the API's data is valid
2. **Daily Average Sentiment Scores**: These visualizations provide a more detailed view of the variance in average sentiment scores. Notably, LAZR had an incomplete line graph due to its limited news coverage.
3. **Whisker Plots for Daily Sentiment**: These visualizations demonstrated the extent of variance in sentiment scores for each of the stocks and offered insights into the consistency of the sentiment trends.
4. **Bar Plots for Source Distribution**: These visualizations allowed us to identify the sources contributing news data for each of the stocks. These visualizations also helped us validate the API and analyze which stocks were most frequently mentioned overall and on specific platforms

5. **Sentiment Label Breakdown:** These area plots displayed how sentiment distribution evolved over time for each of the stocks and provided the most granular insights on sentiment.
6. **Intraday Sentiment Graphs:** These visualizations captured changes in sentiment throughout the day, and helped us to understand how sentiment fluctuated in real time.

These visualizations were chosen and generated because they allowed us to identify patterns, understand the dynamics of sentiment across stocks, and validate our data source.

6. Stream Data

We utilized Azure Function App and Event Hubs, along with Alpha Vantage's Free Tier API to set up our streaming data pipeline. The Alpha Vantage's Free Tier API has a limitation of 100 pulls per day. Since we were pulling data for five stocks we decided to set the pull intervals to 15-minutes throughout the day to ensure we did not exceed the API's limit. The Function App was configured to not only fetch the data but also clean and process it, ensuring that the data would be ready to land directly in the Gold Layer of our medallion architecture.

As for storing the stream data, we used Stream Analytics Job, which allowed us to output the data into our container. A limitation of Stream Analytics Job is that it does not allow users to specify a directory within a container. This resulted in our data being output directly into *storage_account/container* rather than *storage_account/container/Gold*. Prior to implementing the Stream Analytics Job, we attempted to use the native Capture Feature that is built into Event Hubs to stream data directly to the container. Unfortunately, our resource group lacked the required permissions to enable direct capture bringing us to Stream Analytics Job as an alternative approach.

Another significant limitation we encountered was with the API. In the Free Tier subscription, the API defines "latest ticker data" as the closing price from the previous day's market. Even though we configured the Function APP and Event Hubs to pull data at 15-minute intervals, the API did not support real-time data streaming unless we upgraded to the Basic Tier Subscription.

This limitation required us to adjust our modeling approach. Initially, we were planning on configuring the model to predict live stock prices using batch sentiment data. The stream data would be used to validate the model's result and would allow us to have an interactive real-time dashboard. However, due to the API constraint, we had to shift our approach to now predict the closed stock price based on the sentiment data and use the "latest ticker data" from the API to validate our findings. This change to our modeling approach was unexpected but it was the only way our group could make use of the streaming data. Despite the limitations of the API, we were successfully able to set up a functioning streaming data pipeline.

7. Machine Learning

For the machine learning process, we divided the dataset into two key parts. The first part, without sentiment features, used historical price data, including open, high, low, and volume, as features to predict the closing price. The second part, with sentiment features, augmented the historical price data with sentiment metrics such as overall sentiment score and ticker sentiment score, allowing us to evaluate the added value of sentiment analysis. For model selection, we opted for Linear Regression as our baseline model due to its simplicity and interpretability. We also leveraged Azure's AutoML to identify optimal models for predicting stock prices using both historical and sentiment data, with the Voting Ensemble model delivering the best performance. The data was split into training and testing datasets using a 70-30 ratio to avoid overfitting. Model 1 excluded sentiment features and was trained on open, high, low, and volume, while Model 2, incorporating sentiment features, was trained on all variables, including sentiment scores. We ran Azure AutoML on both configurations and compared the results. Additionally, we developed our own model using AzureML notebooks, which yielded better results.

To evaluate the models, we used key metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R^2 Score, and Explained Variance. Model 1 performed reasonably well for stable stocks but struggled with predictions for volatile stocks like Tesla and NVIDIA due to the absence of sentiment data. In contrast, Model 2, which included sentiment features, significantly improved performance, particularly for volatile stocks, as sentiment data helped capture the market mood and sharp price movements influenced by news. Azure AutoML identified the Voting Ensemble model as the best performer, achieving an R^2 score of 0.99975 on testing data. The RMSE of 2.8393 and MAE of 1.0991 further validated its accuracy and the importance of sentiment analysis in stock prediction. Meanwhile, our own model achieved an R^2 value of 0.999977, which was used to predict findings on the stream data, providing us with the current stock values.

8. Findings

The impact of sentiment on stock predictions varied across different stock categories. For stable stocks like AAPL and INTC, sentiment data had minimal influence due to the steady and predictable nature of these stocks. However, for volatile stocks such as TSLA and NVDA, sentiment significantly improved prediction accuracy by capturing market trends influenced by media coverage and news events. In contrast, sentiment data for the penny stock LAZR reflected irregular trends, likely due to the limited and inconsistent media coverage often associated with penny stocks.

When comparing the performance of the models, we observed that the model incorporating sentiment data outperformed the model that relied solely on historical price trends, delivering better prediction accuracy and capturing market behavior more effectively.

Using a Power BI dashboard, we visualized the high correlation between sentiment and stock price movements for volatile stocks, highlighting the role of sentiment in capturing sharp price fluctuations. Additionally, the statistical results from the analysis further validated the enhanced performance achieved with sentiment data integration.

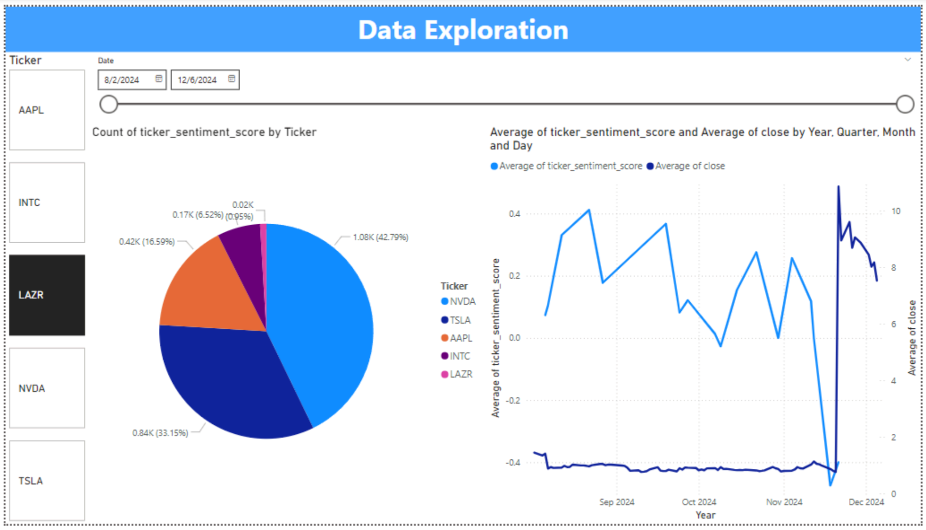
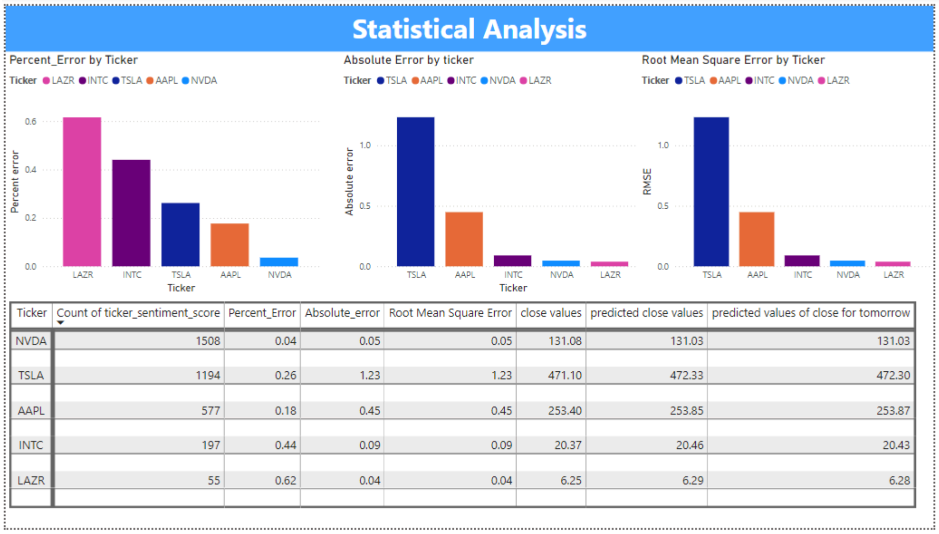


Figure 3: "Sentiment Distribution and Correlation with Stock Price Trends"

The pie chart shows the count of sentiment scores, with NVDA having the highest sentiment score count, indicating significant media and public attention, while LAZR has the lowest, reflecting its limited news coverage. The line graph on the right correlates the average sentiment scores with the average close prices over time, showing that NVDA and TSLA exhibit stronger sentiment-price relationships, whereas LAZR displays irregular patterns due to its low sentiment data and penny stock nature.



Ticker	Count of ticker_sentiment_score	Percent_Error	Absolute_error	Root Mean Square Error	close values	predicted close values	predicted values of close for tomorrow
NVDA	1508	0.04	0.05	0.05	131.08	131.03	131.03
TSLA	1194	0.26	1.23	1.23	471.10	472.33	472.30
AAPL	577	0.18	0.45	0.45	253.40	253.85	253.87
INTC	197	0.44	0.09	0.09	20.37	20.46	20.43
LAZR	55	0.62	0.04	0.04	6.25	6.29	6.28

Figure 4: "Statistical Analysis of Model Performance Across Stocks"

The bar charts display the percent error, absolute error, and root mean square error (RMSE) for each stock. NVDA exhibited the lowest error rates across all metrics due to its stable historical trends and ample sentiment data. TSLA, being a highly volatile stock, has higher absolute and RMSE values due to its sharp fluctuations. LAZR demonstrates the highest percent error, driven by its irregular sentiment patterns and high volatility as a penny stock. The accompanying table summarizes these metrics, providing detailed insights into the model's performance, and highlighting sentiment's impact on prediction accuracy.

Conclusion

This project provided our team with a comprehensive overview of the Data Lifecycle, as it began with a problem that needed to be solved, providing more information for traders through the use of sentiment analysis and machine learning. Though we were limited by budget constraints, our finished product reflected an exploration of various datasets, ingested, cleaned and prepared for business use. We explored the data in heavy detail, coming up with extensive insights to understand the relationship between media sentiment and stock prices. Our training of multiple machine learning models to ensure we reached peak performance is yet another reflection of the experience data engineers in the field face on a day-to-day basis. On top of this, we were tasked with presenting our findings in an effective and appealing manner to support our findings and recommendations. Though many of the alternatives in the same medium would have suffered from much of the same limitations we faced, a real data engineer with financial backing could expand on our findings with more granularity. Many of our initial ideas, such as leveraging the X API, are still highly interesting subjects of study in terms of understanding how social media's impact on stock behavior reflects that of traditional news media. On top of this, Alpha Vantage hosts many paid APIs that would only stand to increase the true "real-time" aspect that we missed out on due to the restriction of the free-tier API.

Ultimately, this project serves as a great experiment to understand the real-world implications of the work of a Big Data Engineer. Working with Azure services like Event Hubs, Stream Analytics, and AutoML gave our team a comprehensive understanding of the Azure platform, a cloud stack used by millions worldwide. The learned experience of having run through the full data lifecycle was as valuable as the results themselves.