AI-DRIVEN STOCK MARKET TRADING STRATEGY GENERATION

Submitted by

Paisa Vasool Team

MSIS BIG DATA ANALYTICS (FALL 2024)

DIVYA VEMULA
MANI KRISHNA TIPPANI
RAHUL CHAUHAN
SHAHERYAR NADEEM

Under the guidance of

Dr. Vijay Gandapodi

Professor

ROBINSON SCHOOL OF BUSINESS

GEORGIA STATE UNIVERSITY

FALL 2024

Abstract

The project presents an end-to-end AI-driven framework for predicting stock market trends and empowering investment decisions. Leveraging historical stock data from Yahoo Finance, the solution integrates modern data engineering practices, deep learning (LSTM), and Generative AI (GANs) to generate synthetic stock sequences and improve forecasting accuracy. The pipeline includes efficient data extraction, transformation, and loading (ETL) using Amazon S3, Snowflake, and SageMaker Studio. A fine-tuned LSTM-GAN hybrid model significantly enhanced prediction reliability, achieving a notable R^2 score of 0.998. Additionally, a GenAI-powered agent, Stock Buddy, was developed to automate technical analysis and provide real-time, explainable investment recommendations tailored for diverse investor personas.

Executive Summary

In today's data-intensive financial markets, extracting actionable insights from raw stock data is crucial for informed decision-making. This project addresses this need by developing a robust AI-powered system that predicts stock price movements and delivers intelligent trading strategies. The workflow began with collecting time-series stock data for 503 tickers via the finance API and storing it in efficient formats (Parquet on S3). Using Snowflake, we implemented a scalable data pipeline to compute over 20 technical indicators and standardize the dataset for modeling.

We implemented an LSTM-based neural network to capture temporal patterns and forecast next-day stock prices. To mitigate data limitations and overfitting, we incorporated GANs to generate synthetic yet realistic stock sequences. Fine-tuning the LSTM with this augmented dataset led to a substantial performance leap—achieving a Root Mean Squared Error (RMSE) of 0.4961 and R^2 of 0.998.

Finally, we built Stock Buddy, a GenAI agent that provides real-time, personalized trading recommendations based on user profiles and technical indicators. This system simplifies financial analysis for novices, supports advisors with data-backed insights, and aids active traders in spotting market opportunities.

Together, the project delivers a full-stack solution from raw data acquisition to GenAI-driven Stock buddy offering a modern blueprint for AI-powered stock trading.

Table of Contents

List of Tables

LIST OF FIGURES

1. Introduction

1.1. Background of the Study

Financial markets generate massive volumes of time-series data daily, offering rich opportunities for predictive analytics. However, raw stock market data is often unstructured and lacks the enrichment required for accurate modeling. With increasing algorithmic trading and AI-driven decision-making, the demand for intelligent systems that can analyze market trends and forecast prices has grown significantly. This study explores the integration of data engineering, deep learning, and Generative AI to develop a comprehensive trading strategy system that can analyze, predict, and assist in making smarter investment decisions with application of Generative AI.

1.2. Purpose of the Study

The purpose of this project is to design and implement an end-to-end stock trading recommendation system using AI models. By transforming raw historical data into analysis-ready formats, engineering technical indicators, and applying deep learning and generative approaches, the study aims to improve the accuracy of stock predictions. Furthermore, it introduces a Generative AI-powered investment assistant to automate analysis and simplify trading decisions.

1.3. Need for Recommendation in Stock Prediction

Novice investors or professionals face information overload and the risk of emotional or impulsive decision-making. The stock market is highly volatile, and identifying reliable signals from vast datasets is challenging. Traditional statistical methods often fall short in capturing long-term dependencies and complex interactions among indicators. Hence, a recommendation system powered by AI models such as LSTM and GAN can improve forecast precision and offer explainable, data-backed trading advice, reducing guesswork and enhancing portfolio performance.

1.4. Aim

To develop a predictive AI-based system for stock trading that integrates time-series deep learning models with generative modeling and a Generative AI-powered agent to deliver real-time, personalized trading strategies and recommendations.

1.5. Objectives

- Develop a robust ETL pipeline using AWS S3, Snowflake, and Python to extract, clean, and store historical stock data enriched with technical indicators.
- Create interactive dashboards using Power BI for technical patterns and strategy performance.
- Build models using LSTM for forecasting, enhancing accuracy with synthetic data generated by GANs.
- Design and deploy a Generative AI agent ("Stock Buddy") that provides real-time, explainable stock trading recommendations based on user profiles and market trends.

1.6. Scope of the Project

This project focuses on predicting stock price movements using historical data from 503 S&P 500 companies over a 5-year period. It covers data ingestion, feature engineering, machine learning modeling synthetic data generation (GAN), and Gen AI agent. While the system is trained and evaluated using selected stocks, the architecture and methodology can be extended to other financial assets or global markets. The project does not include high-frequency intraday trading models or live trading automation.

2. Literature Study

2.1. Tools & Technologies

2.1.1. Amazon Web Services (AWS)

1. **IAM & Identity Center:** AWS Identity and Access Management (IAM) was used to define granular access policies for different components of the architecture, ensuring least-privilege security. The AWS Identity Center provided centralized access control across services such as S3, SageMaker, and Snowflake, enforcing role-based access control (RBAC) and multi-factor authentication (MFA).

2. **Amazon S3:**

   S3 served as the primary cloud storage for raw and transformed datasets. Optimized Parquet format, significantly reducing I/O and enabling efficient query performance via predicate pushdown.

3. **Amazon SageMaker Studio:**

   SageMaker was the ML workbench for developing, training, and evaluating LSTM and GAN models. Using SageMaker notebooks (based on Jupyter), modular scripts were executed for ETL operations, sequence creation, and model training in a scalable cloud-native environment.

2.1.2. Snowflake

Snowflake served as the central cloud data warehouse in this project, enabling secure and scalable storage, transformation, and analysis of high-frequency stock market data. Using **Storage Integration**, it connected seamlessly with Amazon S3 to ingest Parquet files through **External Stages**, allowing secure and credential-free access. Snowflake's decoupled compute-storage model supported high concurrency, enabling low-latency analytics for LSTM modeling, GAN training, Power BI dashboards, and natural language responses via the Stock Buddy AI assistant. This approach aligned with cloud-native ELT practices and demonstrated the value of Snowflake's architecture in handling real-time financial data workflow visualization.

2.1.3. Power BI

Power BI

Power BI functioned as the interactive visualization and decision-support layer in the stock prediction platform, transforming raw and enriched market data into actionable insights for users. Connected directly to Snowflake's tables, Power BI enabled real-time data refreshments, ensuring that all visualizations were built on the most current data. Through features like dynamic filtering, drill-down capability, and cross-highlighting, users could explore specific stocks, sectors, or timeframes intuitively.

Power BI also integrated AI-driven features like anomaly detection to flag unusual volume spikes or price deviations, enhancing early-warning capabilities. The mobile-responsive layout further extended accessibility, allowing users to make informed investment decisions on the go. By complementing the Stock Buddy Generative AI agent, Power BI delivered a dual-modality interface i.e., visual and conversational that advanced both human understanding and automated analytics.

Power BI was not just chosen for aesthetic reasons as it offers several functional advantages that directly support investor decision-making:

Feature	**Advantage**	**User Benefit**
Real-Time Updates	Connects to live Snowflake data	Always current market insights
Interactive Charts	Zoom, filter, cross-highlight	Customizable analysis
AI Integration	Detects anomalies (e.g., sudden volume spikes)	Early warning signals
Mobile-Friendly	Access dashboards on-the-go	Trading decisions anywhere

Table 1: PowerBI features and Advantages

2.2. Description of Software Packages

To build a robust and scalable AI-driven stock prediction system, a combination of powerful open-source Python libraries and cloud SDKs were employed. Below is a comprehensive explanation of the key packages and their role in the project:

yfinance:

The yfinance package was used to programmatically access and download historical stock market data from Yahoo Finance. It facilitated fetching open, high, low, close, volume, and dividends for over 500 stock tickers. Its ease of use and reliability made it a central tool in the data extraction phase.

Pandas:

Pandas are a foundational library for data manipulation and analysis. It enabled tabular processing of stock data, transformations like rolling averages, and easy handling of missing values and data types. The library was pivotal in converting raw data into structured formats suitable for modeling.

Numpy:

The numpy library provided high-performance mathematical operations, particularly for computing rolling windows, vectorized array operations, and managing multidimensional sequences for LSTM input. Its fast matrix operations ensured efficient preprocessing of time-series data.

datetime and os:

These standard Python libraries handle file system operations and datetime transformations. They allowed automatic handling of historical timeframes, sequencing by date, and management of Parquet files on local or cloud directories.

glob:

Used to iterate over multiple stock files during batch processing. It helped in dynamically loading and consolidating data across hundreds of tickers.

Matplotlib, Seaborn and Plotly:

These visualization libraries supported initial exploratory data analysis, enabling line plots, heatmaps, and distribution plots. This helped in understanding trends, correlations, and outliers within the stock data.

For more interactive and web-ready plots, plotly was used, particularly for showcasing multi-feature overlays, trend lines, and price movements in a visually rich format during dashboard integrations.

scikit-learn (sklearn):

The sklearn library was essential for feature scaling, model evaluation, and data splitting. The MinMaxScaler normalized input features for LSTM and GAN models, while metrics like MAE, MSE, and R^2 were calculated using built-in methods.

Scipy:

scipy supported numerical computations beyond standard NumPy operations, particularly for mathematical transformations and distance calculations used in technical indicators and GAN training phases.

torch (PyTorch):

As the deep learning framework, torch enabled construction, training, and fine-tuning of LSTM and GAN neural networks. PyTorch offered flexibility in defining custom architectures and training loops using tensors, autograd, and optimizers.

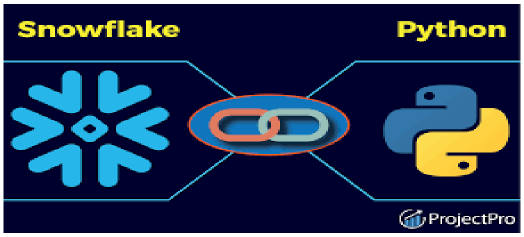
Sqlalchemy:

Used for establishing secure and scalable database connections between Snowflake and the ML notebooks. SQLAlchemy helped extract filtered data efficiently into Python for further modeling.

boto3:

boto3 is the official AWS SDK for Python. It was used to access S3 buckets, upload/download files, and manage IAM authentication seamlessly during ETL operations in the cloud.

snowflake-connector-python:



This package allowed interaction with Snowflake’s cloud data warehouse using SQL queries directly from Python. It facilitated staging data, fetching feature tables, and integrating with Snowflake’s computer environment.

openai:

Used for invoking LLMs during GenAI response generation. The openai API was integrated into the Stock Buddy agent to answer investor queries using retrieved financial context.

Langchain:

A framework to build end-to-end retrieval-augmented generation (RAG) pipelines. It integrated structured and unstructured data for intelligent querying in the Stock Buddy GenAI agent.

pinecone-client:

Pinecone was used as the vector database to store and retrieve embedding vectors of stock commentary, news, and responses. It allowed fast similarity searches essential for the GenAI component.

Duckdb: Employed for lightweight in-memory querying of local datasets during development, especially when slicing and analyzing Parquet files before uploading them to Snowflake.

newspaper3k and lxml_html_clean: These packages scraped, parsed, and cleaned web-based financial news articles. They enriched the dataset with contextual signals such as earnings reports, sentiment, and market commentary.

3. Detailed Explanation of Stock Technical Indicators

Below are the 20+ technical indicators used in the project, each described with its function, formula, and role in enhancing stock prediction:

3.1. Trend-Following Indicators

3.1.1. Simple Moving Average (SMA)

Definition: The Simple Moving Average (SMA) is a basic trend-following indicator that calculates the average price of a stock over a defined period, usually using closing prices. It smooths out short-term price fluctuations and highlights long-term trends in the market.

Explanation: SMA helps traders identify the general direction of a stock's price movement. By plotting SMAs over different periods (e.g., 10-day, 50-day, or 200-day), one can observe crossover points that indicate trend reversals. For example, if a short-term SMA crosses above a long-term SMA, it may signal the start of an uptrend.

SMA is a lagging indicator, reacts to past price movements. However, it remains one of the most used tools in technical analysis due to its simplicity and effectiveness in identifying support/resistance levels and entry/exit signals.

Formula:
SMA = (Sum of closing prices over N periods) / N
$SMA_n = (1/n)\sum Pi$
Where:

- Pi = closing price on day i
- n = number of periods

Role in Prediction: SMA helps smooth out the price data input into machine learning models like LSTM. It enhances the signal-to-noise ratio and allows the model to better recognize trend shifts, making it a foundational feature for any time-series forecasting.

3.1.2. Exponential Moving Average (EMA)

Definition: The Exponential Moving Average (EMA) is a weighted moving average that gives greater importance to more recent prices, making it more responsive to new information compared to the SMA.

Explanation: EMA is often preferred by traders who want to react quickly to recent market changes. It captures sudden shifts in market sentiment, making it a powerful tool for momentum-based strategies. Unlike SMA, EMA adapts faster to sharp price movements, which is crucial in volatile markets.Traders often use a 12-day EMA and a 26-day EMA for short-term and long-term trend analysis. The crossover of these EMAs is integral to calculating the MACD indicator.

Formula:
EMA = (Current Price × α) + (Previous EMA × (1 - α))
EMAt = (Pt*α)+(EMAt−1×(1−α))
Where:

- Pt = current price
- α=2/(n+1), smoothing factor
- n = look-back period

Role in Prediction: EMA improves short-term price sensitivity in time-series data, offering dynamic trend representation. For machine learning, EMA features can serve as early indicators of directional shifts and help fine-tune model accuracy.

3.1.3. Moving Average Convergence Divergence (MACD)

Definition: MACD is a trend-following momentum indicator that displays the relationship between two moving averages of a security's price, usually the 12-day EMA and the 26-day EMA.

Explanation: MACD is composed of three elements: the MACD line, the signal line, and the MACD histogram. When the MACD line crosses above the signal line, it may indicate a buy signal, and vice versa for a sell signal. The histogram reflects the distance between MACD and signal lines, showing the strength of the trend. This dual nature of MACD, capturing both trend and momentum makes it one of the most powerful tools in technical analysis.

Formula:
MACD=EMA12−EMA26
Signal Line=EMA9(MACD)
Histogram=MACD−Signal Line

Role in Prediction: MACD is a compound feature that reflects both direction and velocity of price changes. Including MACD in predictive models helps capture turning points in the market, often before they become visually apparent in the price charts.

3.1.4. Ichimoku Cloud

The Ichimoku Cloud, also known as Ichimoku Kinko Hyo, is a comprehensive trend-following indicator developed by Japanese journalist Goichi Hosoda. It aims to provide a panoramic view of market dynamics, including trend direction, momentum, and support/resistance zones—all within a single visual construct.

At its core, the Ichimoku Cloud comprises five lines, each representing a unique dimension of market behavior. These lines are computed based on historical price highs, lows, and closing values over specific time periods. Together, they form a "cloud" that forecasts potential future support and resistance levels.

1. Tenkan-sen (Conversion Line)

Tenkan-sen = ((9-period high) +(9-period low))/2

Purpose: This line represents the short-term price trend (roughly two weeks in daily data). It reacts quickly to price changes and often acts as a leading indicator. When the price is above the Tenkan-sen, it may signal short-term bullish sentiment below it, bearish sentiment.

2. Kijun-sen (Base Line)
Kijun-sen = ((26-period high) + (26-period low))/2
Purpose: This is the medium-term trend indicator. It is more stable than the Tenkan-sen due to the longer calculation period. When price crosses above the Kijun-sen, it's interpreted as a bullish signal; crossing below implies bearish momentum. The Kijun-sen is often used in conjunction with the Tenkan-sen for crossover strategies, akin to using fast-moving and slow-moving averages.

3. Senkou Span A (Leading Span A)

Senkou Span A = ((Tenkan-sen) + (Kijun-sen))/2
Purpose: This line forms one boundary of the "cloud" (Kumo) and is projected 26 periods into the future. It represents the average of the two shorter-term trend lines (Tenkan-sen and Kijun-sen), giving a near-future forecast of price equilibrium. It helps define dynamic support/resistance levels.

4. Senkou Span B (Leading Span B)
Senkou Span B= ((52-period high) +(52-period low))/2
Purpose: This line forms the second boundary of the cloud and is also plotted 26 periods ahead. It represents long-term support/resistance by tracking the average of price extremes over a broader period (typically ~2.5 months of trading data). Its distance from Span A reflects market volatility. The area between Span A and Span B forms the Ichimoku Cloud (Kumo). If Span A > Span B, the cloud is bullish and usually shaded green. If Span A < Span B, the cloud is bearish and shaded red.

5. Chikou Span (Lagging Span)
Chikou Span = Current Closing Price, plotted 26 periods
Purpose: This lagging line provides a comparison between current price and historical price action. If Chikou Span is above the price from 26 periods ago, it confirms bullish momentum; if below, bearish sentiment is confirmed.
Interpretation Summary:

- Bullish signal: Price is above the cloud, Span A > Span B, and Chikou Span is above past prices.
- Bearish signal: Price is below the cloud, Span A < Span B, and Chikou Span is below past prices.
- Neutral/uncertain: Price within the cloud.

Role in Prediction: The Ichimoku Cloud enriches predictive models by capturing multiple timeframes of price behavior in one multidimensional feature. Its ability to represent momentum, trend, and volatility zones makes it invaluable for both visual interpretation and algorithmic input.

3.1.5. Average Directional Index (ADX)

Definition: The ADX measures the strength of a trend, regardless of its direction. It is derived from the smoothed averages of the difference between positive and negative directional movements.

Explanation: ADX values range from 0 to 100. A reading above 25 typically suggests a strong trend, while below 20 indicates a weak trend or a range-bound market. ADX does not indicate the direction only but the strength making it a valuable addition when used with directional indicators (+DI and -DI).

Formula Overview:

- Calculate +DM, -DM, and True Range (TR)
- Smooth with an EMA over 14 periods
- Compute +DI and -DI
- ADX = EMA of |(+DI - -DI) / (+DI + -DI)| × 100

Step-by-Step Calculation
1. Calculate the Directional Movement (+DM and -DM)
For each period (typically daily):

- Up Move (U) = Today’s High - Yesterday’s High
- Down Move (D) = Yesterday’s Low - Today’s Low

Then:

- +DM = If (U > D) and (U > 0), then +DM = U; else +DM = 0
- −DM = If (D > U) and (D > 0), then -DM = D; else -DM = 0

This logic ensures that both indicators are never positive at the same time.

2. Calculate the True Range (TR)
True Range reflects the most significant price movement in a period. It is defined as the maximum of the following:

- Current High − Current Low
- |Current High − Previous Close|
- |Current Low − Previous Close|

TR=max (Hight−Lowt, |Hight−Closet−1|, |Lowt−Closet−1|)

3. Smooth the Values Using Wilder's Moving Average (14-period EMA): Apply a 14-period Exponential Moving Average (EMA) or Wilder's Smoothing to +DM, -DM, and TR to obtain:

- Smoothed $+DM_{14}$
- Smoothed $-DM_{14}$
- Smoothed TR_{14}

These smoothened values help reduce noise and stabilize the DI calculations.

4. Compute +DI and -DI
+DI = ((Smoothed +DM14) / (Smoothed TR14))*100
−DI = ((Smoothed -DM14) / (Smoothed TR14))*100

5. Compute the Directional Index (DX): DX=(|(+DI) - (−DI) | / ((+DI) + (-DI)))*100
This value shows the divergence between the two directional indicators. A high DX value means a clear directional trend (either up or down), while a low value indicates confusion or balance between buyers and sellers.

6. Compute the Average Directional Index (ADX)
Finally, smooth the DX values using a 14-period EMA:
ADX14=EMA14(DX)
ADX is plotted on its own, usually on a scale of 0 to 100:

- ADX > 25: Strong trend (either bullish or bearish)
- ADX < 20: Weak or non-trending market
- Rising ADX: Trend is strengthening
- Falling ADX: Trend is losing strength

Interpretation Summary:

- +DI > -DI: Bullish trend, buyers in control.
- -DI > +DI: Bearish trend, sellers in control.
- ADX rising: Trend is strengthening (regardless of direction).
- ADX falling: Trend is weakening or consolidating.

Role in Prediction: In stock forecasting, ADX provides insight into the intensity of market movement. It helps the model distinguish between trending and consolidating periods, improving its ability to predict price action during volatile or stable phases.

3.2. Momentum Indicators

3.2.1. Relative Strength Index (RSI)

Definition: The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and magnitude of recent price changes to identify overbought or oversold conditions in a stock's trading activity.

Explanation: RSI values range from 0 to 100. An RSI above 70 typically signals that a stock is overbought and may be due for a pullback, whereas a value below 30 suggests it is oversold and could be poised for a rebound. RSI is often used by traders to detect trend reversals and price divergence relative to the momentum. It is calculated over a default 14-period window and adapts well to both short-term and medium-term trading strategies.

Formula:
RSI = 100 - [100 / (1 + RS)]
Where: RS=Average Gain over N periods / Average Loss over N periods

Role in Prediction: In forecasting models, RSI introduces a measure of momentum-based market sentiment. It allows the model to differentiate between trending strength and potential exhaustion, which is crucial for identifying reversal signals and timing price movements more precisely.

3.2.2. Stochastic Oscillator

Definition: The Stochastic Oscillator compares a stock's closing price to its price range over a given period to indicate momentum and potential reversal points.

Explanation: It produces two lines: %K and %D. The %K line measures the current price's position relative to the high-low range, and the %D line is a moving average of %K. When %K crosses above %D in an oversold area, it may suggest a buy signal. When the crossover happens in the overbought zone, it could signal a sell opportunity. This indicator is most effective in ranging markets where price oscillates between support and resistance levels.

Formula:
%K = [(Current Close - Low) / (High - Low)] × 100
%D = SMA of %K

Role in Prediction: Stochastic values highlight market momentum at key turning points. For predictive modeling, it helps in capturing short-term buying or selling surges and refining decision boundaries in volatile phases.

3.2.3. Commodity Channel Index (CCI)

Definition: The Commodity Channel Index (CCI) is a momentum-based indicator that compares a stock's price to its statistical mean, identifying cyclical trends and deviations from the norm.

Explanation: CCI oscillates above and below zero. Readings above +100 signal overbought conditions or strong upward momentum; readings below -100 suggest oversold or downward momentum. CCI is versatile and works well in both trending and ranging markets, though it's often used in swing trading. It uses the "typical price," which averages the high, low, and close of each period, to reflect the true trading range.

Formula:

CCI = (Typical Price - SMA) / (0.015 × Mean Deviation)
Where:
Typical Price = (High + Low + Close) / 3

Role in Prediction: By quantifying how far price deviates from the mean, CCI provides a valuable perspective on mean-reversion opportunities. Including it in machine learning models strengthens their ability to recognize market extremities.

3.2.4. Price Volume Trend (PVT)

Definition: Price Volume Trend (PVT) is a cumulative momentum indicator that incorporates both price and volume to assess the strength of a trend.

Explanation: PVT adds or subtracts a portion of the daily volume based on the price change. A positive price change increases the PVT value by a proportion of the day’s volume, and a negative change reduces it. Rising PVT indicates accumulation and bullish sentiment, while falling values suggest distribution and bearish pressure.
It functions similarly to On-Balance Volume (OBV) but includes the degree of price change in the volume adjustment.

Formula:
PVT = Previous PVT + [Volume × (Close - Previous Close) / Previous Close]

Role in Prediction: PVT captures momentum and conviction behind price changes. It enhances model accuracy by reflecting whether trading volume supports or contradicts observed price trends.

3.2.5. Volume Rate of Change (VROC)

Definition: The Volume Rate of Change (VROC) measures the percentage change in trading volume over a specific period to detect unusual volume activity that might precede price movement.

Explanation: Sudden increases or decreases in volume often precede significant price changes. VROC helps in identifying these volume shocks early, even before a trend manifests in the price. High VROC values suggest increased investor interest or possible institutional activity, making it a useful early warning signal.

Formula:
VROC = [(Current Volume - Volume n days ago) / Volume n days ago] × 100

Role in Prediction: In forecasting models, VROC is essential for detecting volume surges that precede major market moves. It introduces temporal volatility to volume signals, improving the model’s responsiveness to sudden investor actions.

3.3. Volatility Indicators

Volatility indicators help measure the degree of price variation over a given time period. These indicators are crucial in assessing risk, identifying potential breakout points, and selecting suitable trading strategies.

3.3.1. Bollinger Bands

Definition: Bollinger Bands are a volatility indicator consisting of a moving average (middle band) and two standard deviation lines (upper and lower bands) plotted above and below the moving average.

Explanation: The width of the bands reflects market volatility, wider bands signal high volatility, while narrower bands indicate a period of consolidation or low volatility. When prices reach the upper band, the asset may be overbought; conversely, touching the lower band suggests an oversold condition. Traders often interpret a "band squeeze" (narrowing of the bands) as a precursor to a major price move, while prices riding the bands during expansion indicate a strong trend.

Formula:

- Middle Band = 20-period Simple Moving Average (SMA)
- Upper Band = SMA + (2 × Standard Deviation)
- Lower Band = SMA - (2 × Standard Deviation)

Where:
Standard Deviation(σ) = $\text{sqrt} [\Sigma (Pi - \text{mean})^2 / n]$

Role in Prediction: Bollinger Bands bring dynamic support/resistance boundaries into the model. Their real-time sensitivity to price volatility helps identify breakout or reversal zones and acts as a predictive signal in both trend continuation and exhaustion scenarios.

3.3.2. Standard Deviation

Definition: Standard Deviation is a statistical measure of the dispersion of price data around its mean, used to assess the volatility of a stock.

Explanation: A high standard deviation reflects wide price swings and uncertainty, while a low standard deviation suggests stable pricing. It is a core component of Bollinger Bands and serves as a standalone indicator of volatility. For technical analysts, sudden spikes in standard deviation often precede breakout events or market reactions to news. Its magnitude can be used to evaluate the riskiness of trading in a particular security.

Formula:
$\sigma = \text{sqrt}[\Sigma (Pi - \text{mean})^2 / n]$
Where:

- Pi= Price on day i
- Mean = Mean price over the period
- n = Number of periods

Role in Prediction: Standard deviation acts as a direct volatility measure in forecasting models. When used as a feature, it enables the model to assess risk-adjusted returns and anticipate large movements that typically follow periods of low volatility.

3.3.3. Average True Range (ATR)

Definition: Average True Range (ATR) measures the average of true ranges over a specified period, capturing volatility by accounting for gaps and price movements between trading sessions.

Explanation: True Range (TR) for each period is calculated as the greatest of:

1. High - Low
2. |High - Previous Close|
3. |Low - Previous Close|

ATR smooths these values over a chosen period (typically 14) using a moving average. Unlike standard deviation, which measures dispersion from a mean, ATR focuses on the absolute price movement magnitude, making it ideal for determining risk levels and setting stop-loss points.

Formula:
ATR = EMA of True Range,
where, TR = max (High-Low, |High-ClosePrev|, |Low-ClosePrev|)

Role in Prediction: In machine learning models, ATR conveys the degree of historical price movement. Including it enables better recognition of periods prone to volatility bursts or trend instability. It enhances risk-sensitive predictions by helping the model adjust expectations based on the market's prior range.

3.3.4. Fibonacci Retracement Levels

Definition: Fibonacci retracement is a tool based on the key Fibonacci ratios (23.6%, 38.2%, 50%, 61.8%, and 78.6%) to identify potential support and resistance levels in trending markets.

Explanation: Traders draw retracement levels from a recent significant peak (swing high) to a trough (swing low), or vice versa. These horizontal lines forecast where price may pause or reverse during a pullback in an ongoing trend. Though not a volatility indicator in a strict sense, retracement levels often align with high volatility zones where price decision-making intensifies—particularly when intersecting with other technical indicators.

Formula (conceptual):
For an uptrend:

- Retracement Level = High - [(High - Low) × Fibonacci Ratio]

Common Fibonacci ratios:

- 23.6%, 38.2%, 50%, 61.8%, and 78.6%

Role in Prediction: Fibonacci levels serve as engineered features to help the model anticipate reaction zones within trends. They are critical for encoding market structure, especially for predicting reversals, breakouts, and profit-taking levels.

3.4. Volume-Based Indicators

Volume-based indicators examine how trading volume aligns with price movements. These indicators help confirm trends, reveal accumulation/distribution patterns, and highlight the conviction behind market movements.

3.4.1. On-Balance Volume (OBV)

Definition: On-Balance Volume (OBV) is a cumulative volume indicator that adds volume on up days and subtracts it on down days to track the flow of money into or out of a stock.

Explanation: OBV assumes that volume precedes price movement. A rising OBV suggests accumulation (buying pressure), while a declining OBV implies distribution (selling pressure). If OBV trends upward while price remains flat, it may signal an upcoming breakout. Conversely, if price rises but OBV falls, it may warn of weakening momentum.

Formula:
If Close > Prev Close, OBV += Volume
If Close < Prev Close, OBV -= Volume

Role in Prediction: OBV contributes directional volume momentum to predictive models, helping identify whether volume is confirming or contradicting the price trend. Its inclusion strengthens the model's understanding of market sentiment shifts.

3.4.2. Accumulation/Distribution Line (ADL)

Definition: The Accumulation/Distribution Line (ADL) combines price and volume to measure the cumulative flow of money into or out of a stock.

Explanation: Unlike OBV, which only considers closing prices, ADL incorporates the full price range and closing position within the day's high-low range. When a stock closes near its high with high volume, it suggests accumulation. If it closes near its low, it indicates distribution.

Formula:

1. **Money Flow Multiplier (MFM):** MFM = ((Close−Low)−(High−Close)) / (High−Low)
2. **Money Flow Volume (MFV):** MFV = MFM* Volume
3. $ADL_t = ADL_{t-1}+MFVt$

Role in Prediction: ADL refines the volume-price relationship by accounting for price positioning within daily ranges. This enhances feature richness and allows models to better capture institutional trading footprints and early trend signals.

3.4.3. Chaikin Money Flow (CMF)

Definition: Chaikin Money Flow (CMF) is a volume-weighted average of accumulation and distribution values over a set period (typically 20 days).

Explanation: CMF quantifies buying or selling pressure based on whether the stock closes in the upper or lower part of its range. Values above zero indicate accumulation, while values below zero imply distribution. The strength of CMF reflects the conviction behind recent price moves.

Formula:

CMF = Σ[(Close - Low - (High - Close)) / (High - Low) × Volume] / ΣVolume

Role in Prediction: CMF integrates directionality, volume, and volatility, providing nuanced insights into short-term market behavior. For predictive models, it enhances detection of money flow shifts that often precede trend changes.

3.4.4. Volume RSI

Definition: Volume RSI is a variation of the classic Relative Strength Index (RSI), applied to changes in volume rather than price.

Explanation: Volume RSI compares recent gains in volume to recent losses in volume. It helps assess whether bullish or bearish volume momentum is increasing. Values above 70 indicate strong buying pressure; below 30 indicate strong selling pressure. It's particularly useful in identifying trend confirmation or exhaustion during high-volume phases.

Formula:

Volume RSI = 100 - [100 / (1 + Volume Gain / Volume Loss)]

Role in Prediction: This indicator amplifies the predictive model's ability to measure not just price momentum but the underlying force (volume) driving it. It captures the conviction behind price changes and can be critical during breakouts or reversals.

3.4.5. Volume Weighted Average Price (VWAP)

Definition: VWAP is the average price of a stock over a trading session, weighted by volume. It is used as a benchmark to assess whether a stock was bought or sold at a favorable price.

Explanation: VWAP is intraday and resets daily. It reflects where the bulk of trading took place and is commonly used by institutional investors to measure trade efficiency. Price above VWAP = bullish; below VWAP = bearish. It also serves as a dynamic support/resistance level.

Formula:

$$\text{VWAP} = \Sigma(\text{Price} \times \text{Volume}) / \Sigma\text{Volume}$$

Role in Prediction: VWAP provides contextual value during intraday modeling. It introduces a fair-value reference point that helps models distinguish between speculative spikes and justified price levels based on volume-weighted demand.

4. Methodology

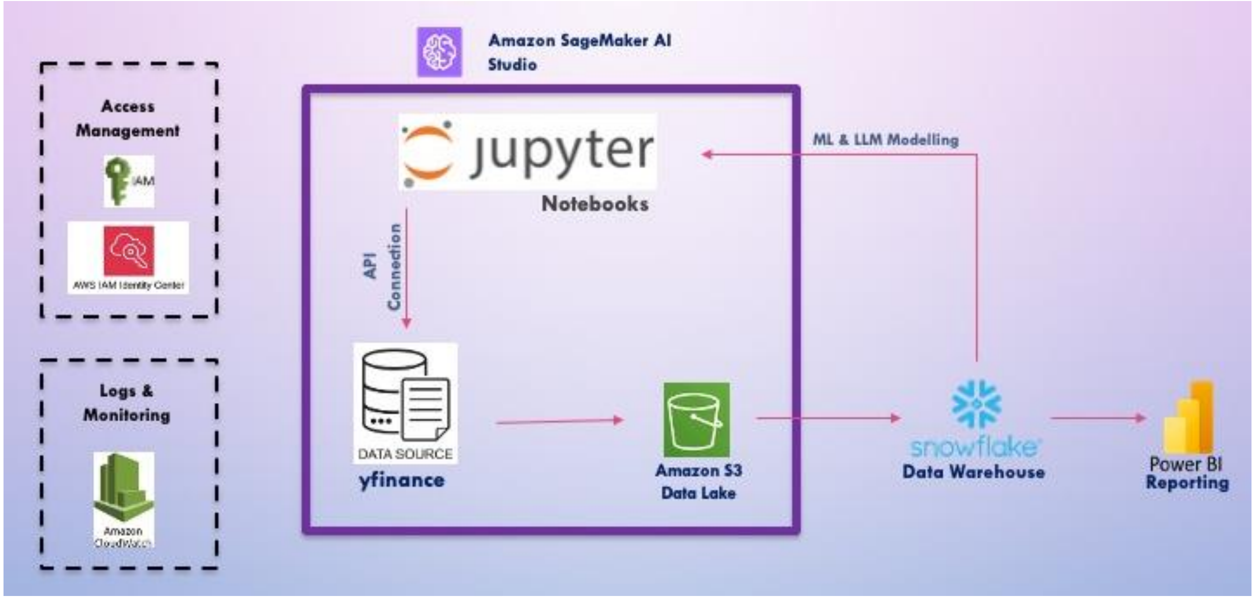


Figure 1:Technical Architecture Methodology

Description based on the methodology diagram with the full workflow from ETL to Power BI reporting, deep learning modeling, and GenAI Stock Buddy integration:

1. **Data Extraction via yfinance**: Historical stock data (Raw Stock Data, volume, dividends) for S&P 500 companies is extracted using the yfinance Python library.
2. **Data Orchestration in Jupyter (SageMaker Studio)**: Data ingestion, transformation scripts, and ML model development are orchestrated within Amazon SageMaker notebooks, offering a centralized ML workspace.
3. **Storage in Amazon S3 Data Lake**: Extracted data is stored in Parquet format within Amazon S3, enabling scalable and cost-efficient storage as a staging area before warehousing.
4. **Access Control via IAM & Identity Center**: Secure role-based access is configured using AWS IAM and Identity Center to ensure compliance, isolation, and controlled permissions across services.
5. **Monitoring with CloudWatch**: Logging and monitoring of pipeline execution, data movement, and model training activities are tracked using Amazon CloudWatch for observability.
6. **Data Warehousing in Snowflake**: Cleaned and enriched stock data including technical indicator is loaded into Snowflake for structured storage and high-performance querying.
7. **ML and LLM Modeling**: LSTM and GAN models are trained using SageMaker on cleaned data from Snowflake, producing stock price predictions and generating synthetic time-series data.
8. **Power BI Reporting**: Snowflake serves as the live data source for Power BI dashboards, enabling real-time visualizations of price trends, indicator signals, and sector analysis.
9. **Stock Buddy GenAI Agent**: A conversational AI assistant is built using LangChain and OpenAI's GPT, integrated with Snowflake to interpret technical signals and deliver natural language investment advice.

10. **Full-Stack Analytical Flow**: From raw data ingestion to deep learning forecasts and GenAI explanations, the pipeline delivers an end-to-end financial analytics ecosystem combining traditional BI with cutting-edge AI.

5. Data Engineering and ETL Workflow

The Data Engineering and ETL (Extract, Transform, Load) component is the foundational layer of the project, enabling the seamless flow of raw market data into a structured, analysis-ready dataset used by machine learning and generative AI components. This system was built to support high-volume, time-series financial data, prioritizing scalability, reliability, and data quality at every stage.

5.1. Data Extraction

The data acquisition process began with retrieving historical market data for 503 companies in the S&P 500 index using the yfinance Python package. This library acts as a wrapper over Yahoo Finance's public API, enabling the programmatic download of:

- Timestamped OHLCV Data: Open, High, Low, Close, and Volume
- Corporate Actions: Dividends and Stock Splits
- Metadata: Ticker symbol, exchange info

The extraction covered a 5-year time window on a daily frequency, generating high-resolution time-series data. Each ticker's data was saved in individual CSV files using a batch processing script with logging and retry mechanisms to handle potential API rate limits or failures.

5.2. Data Storage

The raw Pandas Dataframes, while readable, were not efficient for storage or querying. Therefore, the following transformations were applied:

- **Conversion to Parquet Format:** Each Dataframe was converted to Parquet, a columnar binary storage format optimized for big data workloads. The transformation:
  - Reduced total data size from ~870 MB (CSV) to ~34 MB (Parquet)
  - Enabled predicate pushdown and partial column reads in Snowflake
  - Supported schema evolution and nested structures
- **Cloud Storage in Amazon S3**: The Parquet files were uploaded to an Amazon S3 bucket, structured by ticker and date ranges (e.g., /stocks/SPY/2020/), enabling scalable partitioned access.
  - Bucket policies and IAM roles enforced fine-grained access control
  - Redundant copies were stored in multi-AZ S3 to ensure durability and availability
- **Automation:** A script automated the entire pipeline for extraction and loading into S3 and then to Snowflake. This process ran periodically via Jupyter notebooks in Amazon SageMaker Studio.

5.3. Data Ingestion into Snowflake

To make the data queriable and integrate it into downstream modeling, the following Snowflake configuration was performed:

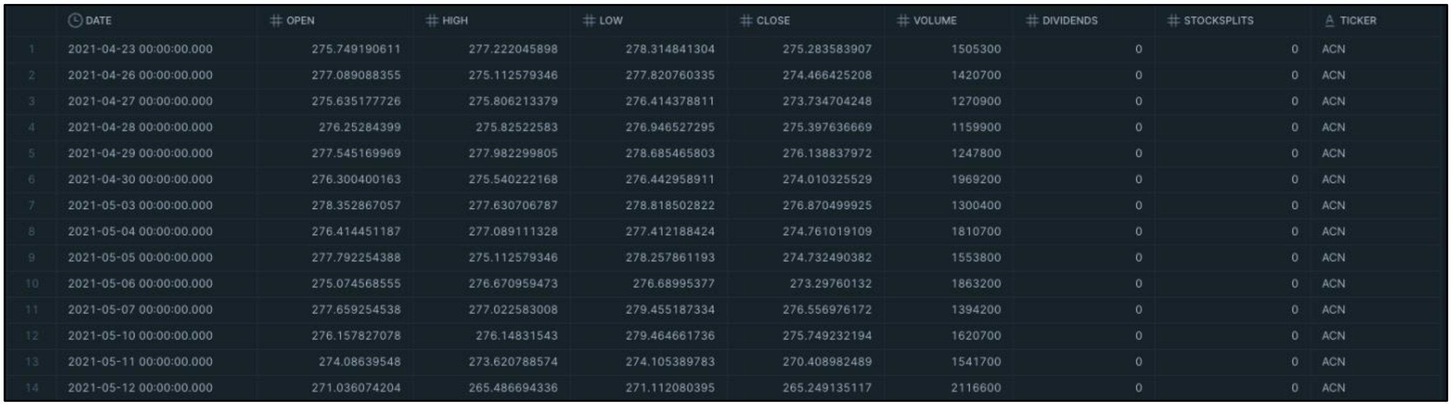
a. External Stage and Storage Integration

- A storage integration object was created linking Snowflake to S3 via a secure IAM Role.
- An external stage was defined to connect to the S3 path containing Parquet files.
- Snowflake could now directly access and query Parquet data without first copying it.

b. Loading into Raw Table

- Parquet files were loaded into a staging table STOCK_DATA using the COPY INTO command.
- Data types were explicitly defined:

- DATE TIMESTAMP_NTZ
- FLOAT for Stock Indicators
- NUMBER (38,0) for Volume
- VARCHAR for tickers



	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	DIVIDENDS	STOCKSPLITS	TICKER
1	2021-04-23 00:00:00.000	275.749190611	277.222045898	278.314841304	275.283583907	1505300	0	0	ACN
2	2021-04-26 00:00:00.000	277.089088355	275.112579346	277.820760335	274.466425208	1420700	0	0	ACN
3	2021-04-27 00:00:00.000	275.635177726	275.806213379	276.414378811	273.734704248	1270900	0	0	ACN
4	2021-04-28 00:00:00.000	276.25284399	275.82522583	276.946527295	275.397636669	1159900	0	0	ACN
5	2021-04-29 00:00:00.000	277.545169969	277.982299805	278.685465803	276.138837972	1247800	0	0	ACN
6	2021-04-30 00:00:00.000	276.300400163	275.540222168	276.442958911	274.010325529	1969200	0	0	ACN
7	2021-05-03 00:00:00.000	278.352867057	277.630706787	278.818502822	276.870499925	1300400	0	0	ACN
8	2021-05-04 00:00:00.000	276.414451187	277.089111328	277.412188424	274.761019109	1810700	0	0	ACN
9	2021-05-05 00:00:00.000	277.792254388	275.112579346	278.257861193	274.732490382	1553800	0	0	ACN
10	2021-05-06 00:00:00.000	275.074568555	276.670959473	276.68995377	273.29760132	1863200	0	0	ACN
11	2021-05-07 00:00:00.000	277.659254538	277.022583008	279.455187334	276.556976172	1394200	0	0	ACN
12	2021-05-10 00:00:00.000	276.157827078	276.14831543	279.464661736	275.749232194	1620700	0	0	ACN
13	2021-05-11 00:00:00.000	274.08639548	273.620788574	274.105389783	270.408982489	1541700	0	0	ACN
14	2021-05-12 00:00:00.000	271.036074204	265.486694336	271.112080395	265.249135117	2116600	0	0	ACN

Figure 2: Snowflake Table STOCK_DATA with Raw Data

5.4. Feature Engineering and Data Modeling

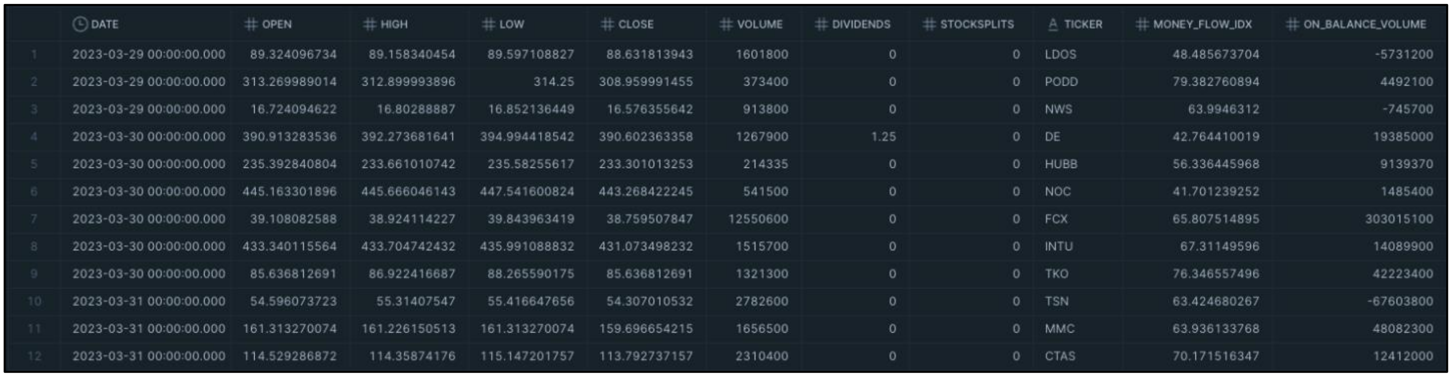
After the successful ingestion of raw data into Snowflake, additional data preparation and feature engineering steps were undertaken to enrich and cleanse the dataset for model training and GenAI applications.

5.4.1. Staging Table with Technical Indicators

A new intermediate table was created: STOCK_FINAL

Purpose:

1. Merge raw data from STOCK_DATA with 20+ technical indicators computed using SQL logic transformations.
2. These indicators spanned categories like trend-following, momentum, volatility, and volume-based metrics.



	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	DIVIDENDS	STOCKSPLITS	TICKER	MONEY_FLOW_IDX	ON_BALANCE_VOLUME
1	2023-03-29 00:00:00.000	89.324096734	89.158340454	89.597108827	88.631813943	1601800	0	0	LDOS	48.485673704	-5731200
2	2023-03-29 00:00:00.000	313.269989014	312.899993896	314.25	308.959991455	373400	0	0	PODD	79.382760894	4492100
3	2023-03-29 00:00:00.000	16.724094622	16.80288887	16.852136449	16.576355642	913800	0	0	NWS	63.9946312	-745700
4	2023-03-30 00:00:00.000	390.913283536	392.273681641	394.994418542	390.602363358	1267900	1.25	0	DE	42.764410019	19385000
5	2023-03-30 00:00:00.000	235.392840804	233.661010742	235.58255617	233.301013253	214335	0	0	HUBB	56.336445968	9139370
6	2023-03-30 00:00:00.000	445.163301896	445.666046143	447.541600824	443.268422245	541500	0	0	NOC	41.701239252	1485400
7	2023-03-30 00:00:00.000	39.108082588	38.924114227	39.843963419	38.759507847	12550600	0	0	FCX	65.807514895	303015100
8	2023-03-30 00:00:00.000	433.340115564	433.704742432	435.991088832	431.073498232	1515700	0	0	INTU	67.31149596	14089900
9	2023-03-30 00:00:00.000	85.636812691	86.922416687	88.265590175	85.636812691	1321300	0	0	TKO	76.346557496	42223400
10	2023-03-31 00:00:00.000	54.596073723	55.31407547	55.416647656	54.307010532	2782600	0	0	TSN	63.424680267	-67603800
11	2023-03-31 00:00:00.000	161.313270074	161.226150513	161.313270074	159.696654215	1656500	0	0	MMC	63.936133768	48082300
12	2023-03-31 00:00:00.000	114.529286872	114.35874176	115.147201757	113.792737157	2310400	0	0	CTAS	70.171516347	12412000

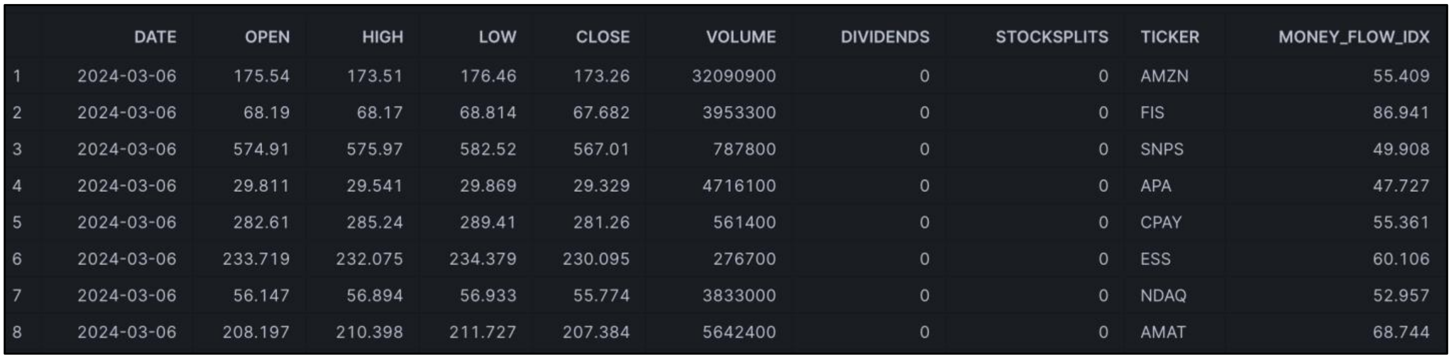
Figure 3: Snowflake Stage Table STOCK_FINAL

5.4.2. Data Cleaning and Standardization

After staging the enriched data,The cleaned dataset was stored in a final table STOCK_FINAL_CLEANED with the following cleaning steps:

1. Duplicate Records were removed using a composite key (TICKER, DATE)

2. Date Formatting was enforced to YYYY-MM-DD
3. Any residual nulls, non-numeric values, or outliers were treated appropriately.



	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	DIVIDENDS	STOCKSPLITS	TICKER	MONEY_FLOW_IDX
1	2024-03-06	175.54	173.51	176.46	173.26	32090900	0	0	AMZN	55.409
2	2024-03-06	68.19	68.17	68.814	67.682	3953300	0	0	FIS	86.941
3	2024-03-06	574.91	575.97	582.52	567.01	787800	0	0	SNPS	49.908
4	2024-03-06	29.811	29.541	29.869	29.329	4716100	0	0	APA	47.727
5	2024-03-06	282.61	285.24	289.41	281.26	561400	0	0	CPAY	55.361
6	2024-03-06	233.719	232.075	234.379	230.095	276700	0	0	ESS	60.106
7	2024-03-06	56.147	56.894	56.933	55.774	3833000	0	0	NDAQ	52.957
8	2024-03-06	208.197	210.398	211.727	207.384	5642400	0	0	AMAT	68.744

Figure 4: Snowflake Final Table STOCK_FINAL_CLEANED

5.4.3. SP500 Company Metadata

To enhance data contextualization and support dashboard visualizations and RAG-based GenAI queries, a dimension table was created for static company metadata with table Name COMPANIES_SP500.

Source:

1. Extracted from Wikipedia’s List of S&P 500 companies.
2. Stored in Snowflake via Python ETL using Pandas and snowflake-connector-python.

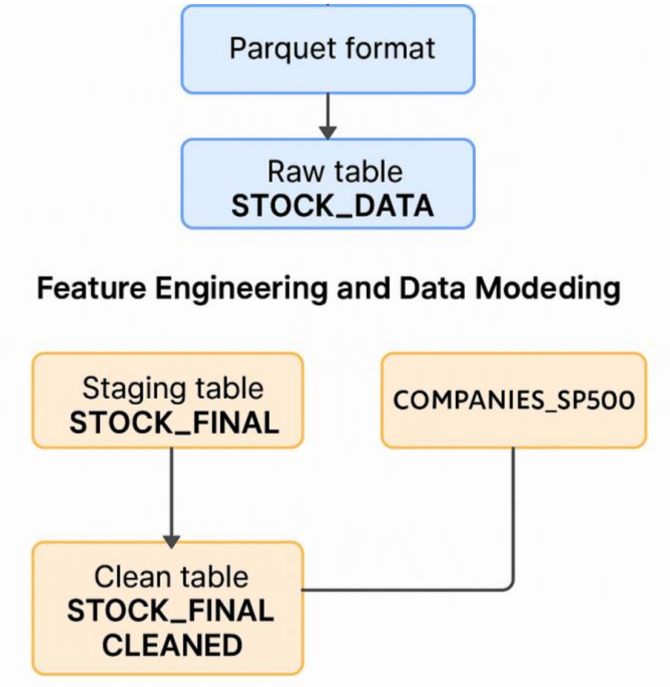
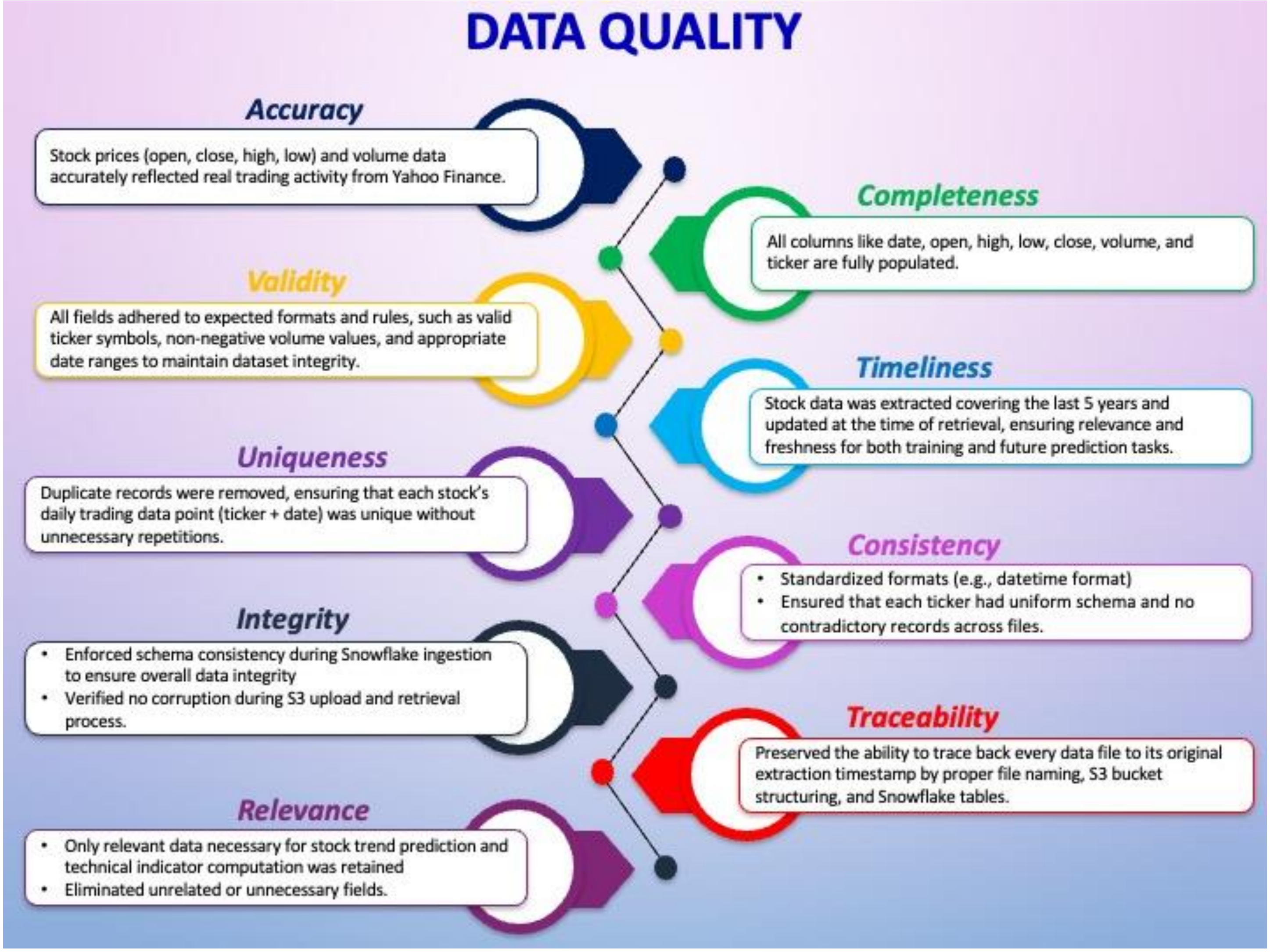

Parquet format
Raw table STOCK_DATA
Feature Engineering and Data Modeding
Staging table STOCK_FINAL
COMPANIES_SP500
Clean table STOCK_FINAL CLEANED

Figure 5: ETL Workflow Description

6. Data Quality Measures

Maintaining high-quality data is a critical requirement for any data-driven financial system. In this project, various dimensions of data quality were applied to ensure that the information flowing through the pipeline from initial extraction to final model consumption was reliable, clean, and analytically sound. Each data quality dimension was not only enforced technically but was also embedded within the broader context of traceability, reproducibility, and relevance to downstream tasks, such as LSTM forecasting, GAN augmentation, and GenAI-based advisory.



Stock prices (open, close, high, low) and volume data accurately reflected real trading activity from Yahoo Finance.

All columns like date, open, high, low, close, volume, and ticker are fully populated.

All fields adhered to expected formats and rules, such as valid ticker symbols, non-negative volume values, and appropriate date ranges to maintain dataset integrity.

Stock data was extracted covering the last 5 years and updated at the time of retrieval, ensuring relevance and freshness for both training and future prediction tasks.

Duplicate records were removed, ensuring that each stock's daily trading data point (ticker + date) was unique without unnecessary repetitions.

- Standardized formats (e.g., datetime format)
- Ensured that each ticker had uniform schema and no contradictory records across files.

- Enforced schema consistency during Snowflake ingestion to ensure overall data integrity
- Verified no corruption during S3 upload and retrieval process.

Preserved the ability to trace back every data file to its original extraction timestamp by proper file naming, S3 bucket structuring, and Snowflake tables.

- Only relevant data necessary for stock trend prediction and technical indicator computation was retained
- Eliminated unrelated or unnecessary fields.

Figure 6: Data Quality Measures

6.1. Accuracy

Accuracy, in the context of this project, refers to the extent to which the data correctly represents actual historical market behavior. Since the dataset forms the basis for calculating all technical indicators and serves as direct input to time-series models, accuracy was paramount. Traceability for this dimension was ensured by cross-verifying random samples of OHLC (Open, High, Low, Close) data against Yahoo Finance's official interface. Additional accuracy validation logic was embedded in the preprocessing stage, including assertions that ensured low <= open/close <= high, which protected against structural inconsistencies and erroneous input from API anomalies or source data corruption.

The relevance of accuracy was most directly observed in how small distortions in price data could cascade into incorrect signal generation by indicators such as RSI, MACD, or Bollinger Bands. In predictive modeling, even minor inaccuracies can magnify over time, leading to poor model generalization and diminished forecasting credibility. Thus, accuracy was systematically reinforced to maintain the fidelity of both indicators and model outputs.

6.2. Completeness

Completeness ensures that the dataset is holistic, with no missing values or records that could introduce uncertainty or bias into the analysis. In this project, completeness was achieved by validating that each of the 503 companies included in the S&P 500 index had corresponding, uninterrupted historical data over a continuous five-year period. The completeness of essential field such as ticker, date, close, and volume was programmatically checked after both extraction and loading stages. Any missing data rows due to market holidays or data outages were handled using forward-fill interpolation if adjacent data existed or were excluded with logged justifications.

The relevance of completeness is tied to the sequential nature of LSTM and GAN models, which require uninterrupted time-series sequences. Missing entries break the continuity required for windowed forecasting. Furthermore, technical indicators that rely on rolling computations (e.g., moving averages, ADX, or ATR) become invalid when the underlying data is incomplete, hence reducing model input quality. Ensuring completeness was therefore essential to preserving the temporal integrity of features and maintaining consistency in pattern recognition.

6.3. Consistency

Consistency was enforced by standardizing data formats and definitions throughout the pipeline. A consistent YYYY-MM-DD date format was applied across all stages, and uniform data types were declared and enforced explicitly when loading into Snowflake tables. Numerical columns such as volume and close were cast into fixed-precision formats to ensure type safety. Additionally, ticker symbols were normalized (e.g., replacing dots with dashes) to ensure compatibility across Yahoo Finance, S3 paths, and Snowflake schemas.

The relevance of consistency lies in the elimination of ambiguity. Inconsistently formatted dates or ticker IDs can result in misjoins, model training failures, or inaccurate visualizations in Power BI. By enforcing a unified schema and transformation logic, the project maintained semantic clarity and avoided the silent propagation of format-related errors.

6.4. Timeliness

Timeliness refers to the currency of the data (about how up to date), it is relative to the time of analysis. The dataset used in this project was constrained to the most recent five-year period as of data extraction. This window was chosen to capture both recent market behavior and a representative cycle of economic events, including volatility shocks. To maintain timeliness, the data extraction process was designed for reusability and could be re-executed on-demand through SageMaker notebooks.

Timeliness is highly relevant in financial prediction, as market regimes evolve. Using outdated data diminishes the relevance of learned patterns, especially for momentum and volatility indicators. By ensuring that the data reflects recent market dynamics, the model is better equipped to generalize to current and future conditions.

6.5. Validity

Validity was preserved by confirming that the data conforms to expected formats, ranges, and business rules. In Snowflake, data types were enforced through explicit schema declarations. Logic checks ensured that volumes were never negative and that price fields respected their logical relationships. Additionally, categorical fields such as tickers were validated against an authoritative reference table (COMPANIES_SP500) to detect any misclassified entries.

This dimension is crucial in stock modeling because invalid data can yield meaningless or misleading results. For example, a negative volume would compromise volume-based indicators like OBV or Chaikin Money Flow, while a malformed date could break chronological ordering essential for LSTM inputs. Validity was enforced to preserve analytical soundness and guard against computational errors during indicator calculations and modeling.

6.6. Uniqueness

Uniqueness ensures that each entity or event is represented only once in the dataset. This was achieved by identifying and removing duplicate records using a composite primary key composed of ticker and date. Duplication checks were executed both in Python during preprocessing and within Snowflake using deduplication queries before final data consolidation into the STOCK_FINAL_CLEANED table.

In predictive modeling, duplicate data biases the model's learning process and inflates signal strength artificially. In descriptive analytics, duplicates can distort time-series aggregations, leading to incorrect dashboard metrics or chart visualizations. Therefore, enforcing uniqueness was essential for maintaining the statistical integrity of the data.

6.7. Integrity

Finally, integrity ensures logical cohesion between data tables and attributes. This was maintained by establishing referential integrity between the stock pricing tables (STOCK_FINAL, STOCK_FINAL_CLEANED) and the company dimension table (COMPANIES_SP500). Each ticker in the historical pricing table was cross validated against the metadata in the SP500 dimension table. Additionally, all join operations were tested to confirm that no orphaned keys existed and that filtering by company-level attributes (sector, sub-industry) yielded accurate results.

Integrity is critical in supporting explainability and downstream GenAI capabilities. For instance, when the AI assistant is queried about "top gainers in the healthcare sector," the response depends entirely on the integrity of the ticker-to-sector linkage. Integrity also ensures valid groupings and aggregations in Power BI dashboards, making it indispensable for both analytical and narrative components of the solution.

***The traceability and relevance is explained for each data quality measure, therefore not separated to maintain context.

7. Exploratory Data Analysis:

7.1. Introduction

Power BI was selected as the business intelligence and visualization layer of the stock prediction platform due to its strong integration capabilities with cloud data warehouses, its interactive dashboarding features, and its ability to provide real-time insights to investors, analysts, and AI systems alike. By converting raw and model-generated data into actionable insights, Power BI serves as the final decision-support interface in the analytics value chain.

The primary goal of this dashboard integration was to translate complex stock market signals—derived from 20+ technical indicators into intuitive and dynamic visuals. These visuals empower users to detect patterns, monitor performance, and make informed decisions based on both historical and predictive analytics.

In the overall architecture, Power BI serves three purposes:

1. Validation Layer: Verifies the efficacy of model outputs (e.g., does predicted price align with MACD signals?)
2. Interpretability Tool: Allows domain experts to trace model suggestions back to data-driven visuals
3. End-User Interface: Enables non-technical users (e.g., investors or advisors) to interact with predictive analytics meaningfully.

7.2. Data Pipeline and Source Connectivity

At the heart of this integration is the live connection between Snowflake and Power BI, enabling real-time visualization of high-frequency financial data. The dataset exposed to Power BI is the result of the multi-layered ETL process previously described. It includes:

- Cleaned data from the STOCK_FINAL_CLEANED table
- 20+ technical indicators generated through SQL-based feature engineering
- Dimension data from the COMPANIES_SP500 metadata table (sector, industry, HQ)

This structured schema allows Power BI to pull granular, enriched, and context-aware datasets directly from Snowflake's cloud-based infrastructure, ensuring high performance and near real-time synchronization.

7.3. Dashboard Architecture and Components

The Power BI dashboard was organized into several focused sections, each catering to a different aspect of stock market analysis:

7.3.1. Stock Performance Overview

This section includes:

- Time-series visualizations of stock prices over time (daily/weekly granularity)
- Overlay of Simple and Exponential Moving Averages (SMA/EMA)
- Interactive elements to compare multiple stocks side-by-side.

It enables users to trace long-term trends, detect support/resistance zones, and visually identify crossover events that typically precede major market moves.

7.3.2. Momentum Analysis Panel

This part displays real-time values and historical trends of key momentum indicators such as:

- Relative Strength Index (RSI): Reveals overbought or oversold conditions
- Moving Average Convergence Divergence (MACD): Highlights potential bullish/bearish reversals

- Stochastic Oscillator: Indicates short-term trend saturation

These gauges are configured with conditional formatting (e.g., color bands for RSI thresholds), allowing for quick sentiment assessment briefly.

7.3.3. Volatility and Risk Monitor

Users are provided with insights on market volatility through:

- Bollinger Bands, which adjust dynamically to price movements
- Standard Deviation charts, which reflect statistical price variability
- Average True Range (ATR), which helps gauge daily volatility breadth

7.4. Dashboard Insights and Visual Analysis

Each Dashboard captures a unique aspect of market behavior using advanced technical indicators, supporting both human analysts and AI systems in interpreting financial trends. The insights presented combine price action, volume, volatility, and momentum indicators to form a multi-dimensional view of stock performance.

7.4.1. Price Action Analysis (HUBB, April 2025)

This visualization presents the daily price trend of HUBB Corporation for April 2025, focusing on both raw price movement and momentum confirmation. The chart includes:

- A candlestick plot showing open, high, low, and close prices
- Volume bars, where spikes in volume are aligned with significant price moves
- Exponential Moving Averages (12-day and 26-day), plotted in blue and orange.

The 12-day EMA crossing below the 26-day EMA suggests a bearish momentum shift, which is a common trend-reversal signal. Price levels around $320 act as a support zone, while $340 appears to be a resistance zone. This chart supports short-term trading strategies and crossover-based forecasting models.

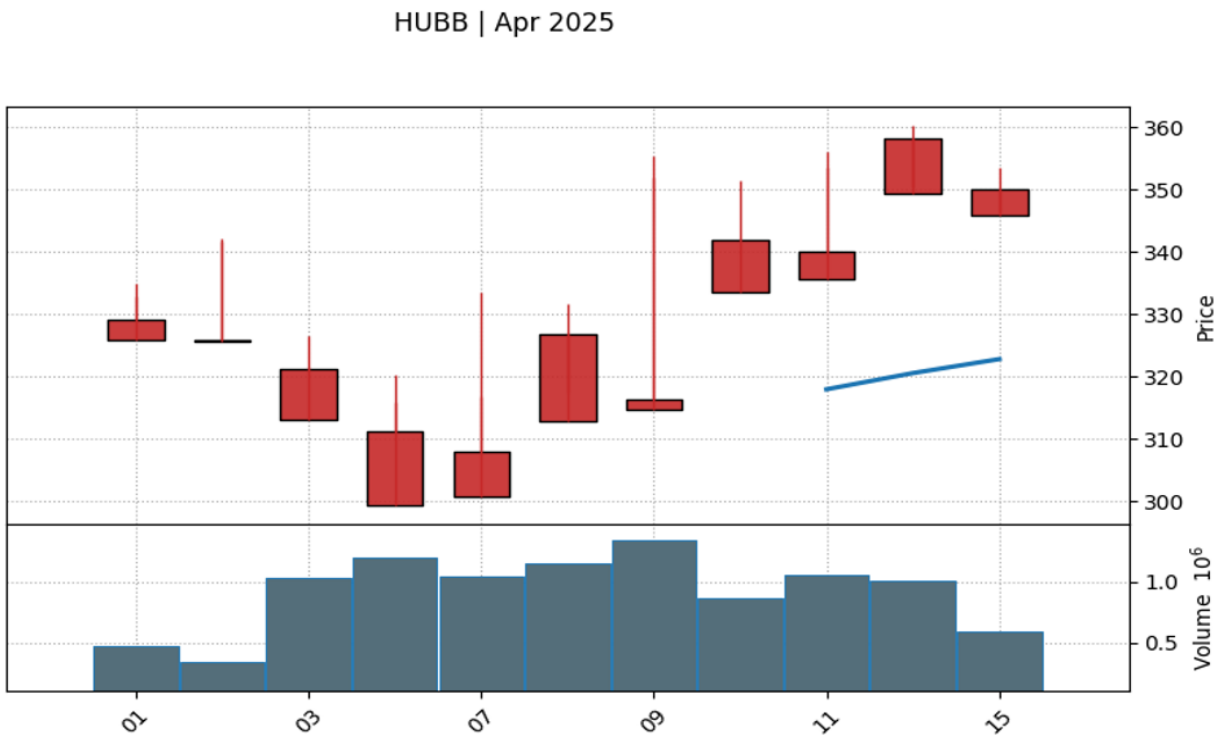


Figure 7: Price Action Analysis

7.4.2. Fibonacci retracement levels

The Fibonacci retracement levels, drawn between a recent swing high of $349.29 and swing low of $299.43 (a 19.8% retracement range). Key retracement levels are labeled:

- 23.6% at $353.28 → Immediate resistance
- 38.2% at $335.50 → Mid-level pivot
- 61.8% at $310.75 → Strong support

These levels serve as decision points where traders anticipate reversals or continuation patterns. The current price hovering near the 38.2% level signals indecision, and a break below this level could forecast a shift toward deeper support. Fibonacci-based visuals are particularly valuable in both manual strategy development and LSTM sequence validation.

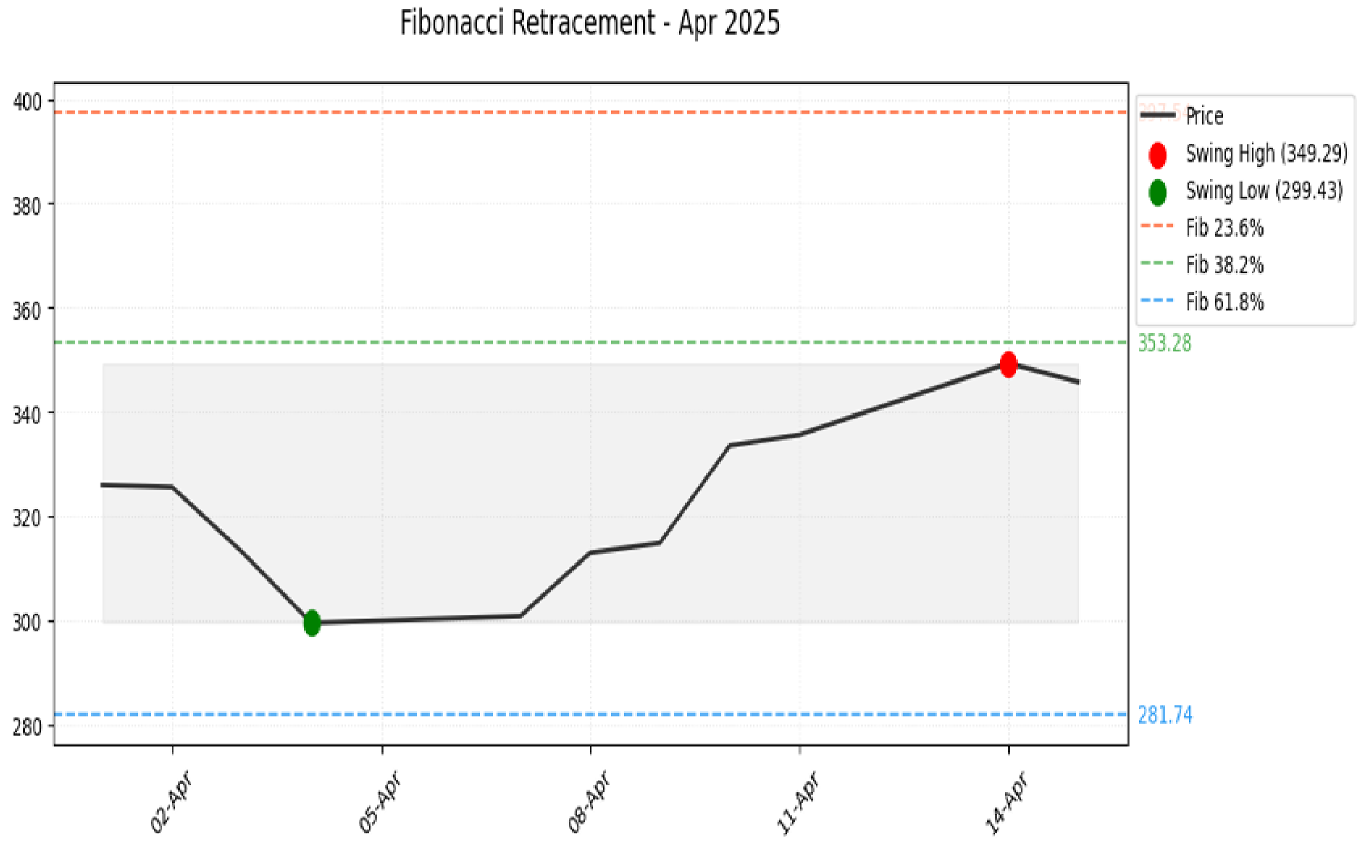


Figure 8: Fibonacci retracemen Grapht

7.4.3. Technical Analysis of Stock performance (AAPL)

Multi-Year Dashboard Analysis: Apple Inc. (AAPL)

This multi-year interactive dashboard presents Apple Inc. (AAPL)'s stock performance specifically on the fourth trading day of each quarter across several years. The visualization integrates key technical indicators and supports dynamic time-based exploration.

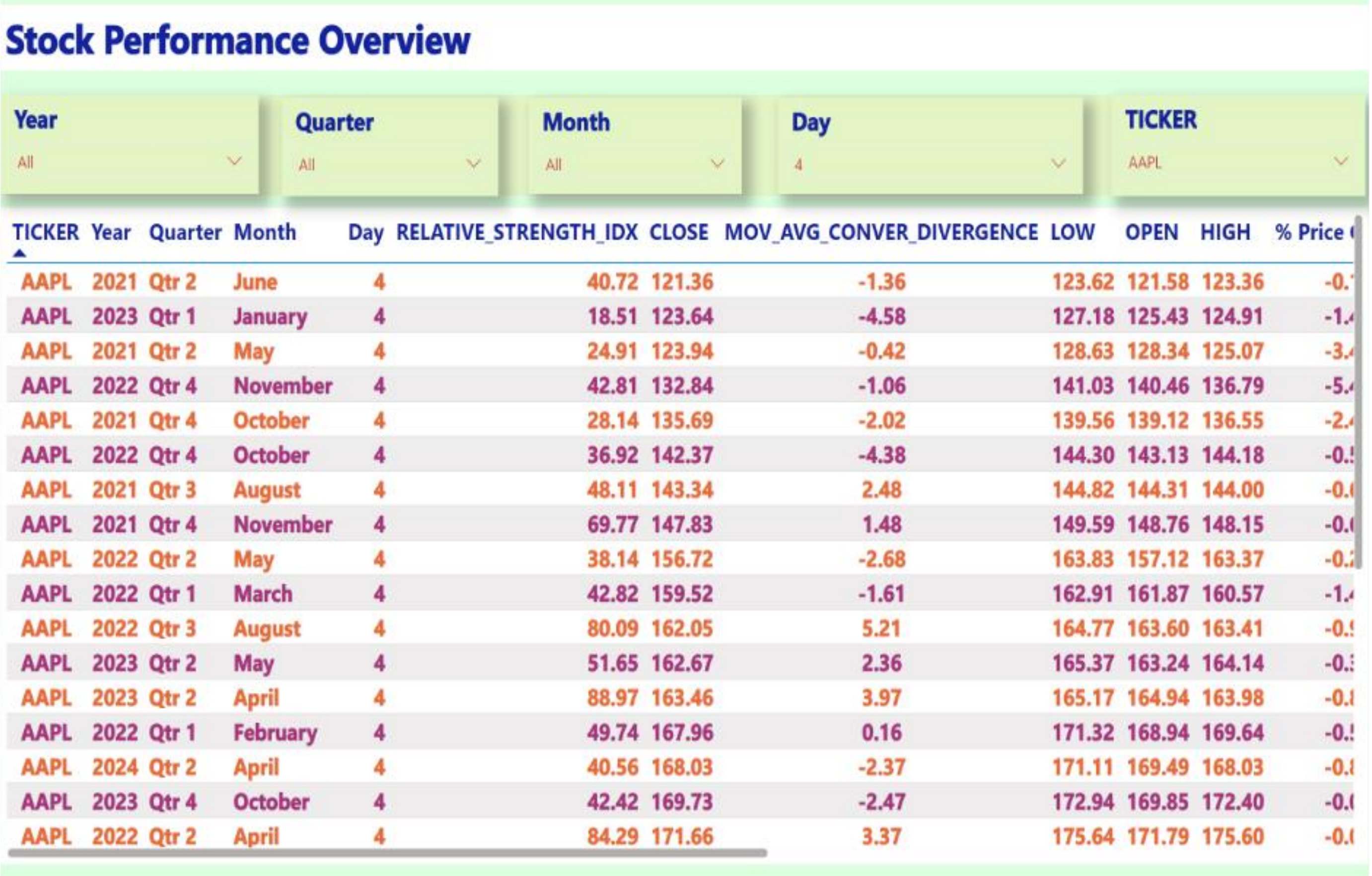
Key Features:

- Visualized indicators include Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), High/Low/Close prices, and percentage price change.
- Interactive filters enable users to explore data by year, quarter, month, and day.

Insights Highlighted:

- **January 2023:** A combination of low RSI and a negative MACD signaled bearish sentiment—potentially indicating a favorable buying opportunity.
- **April 2023:** High RSI alongside a positive MACD indicated bullish momentum and upward price pressure.

This dashboard supports seasonal pattern recognition and helps users correlate price action with technical signals. Its interactive design also aids in validating model outputs against historical market behavior, offering practical utility for both analysts and investors.



Stock Performance Overview

Year: All
Quarter: All
Month: All
Day: 4
TICKER: AAPL

TICKER	Year	Quarter	Month	Day	RELATIVE_STRENGTH_IDX	CLOSE	MOV_AVG_CONVER_DIVERGENCE	LOW	OPEN	HIGH	% Price
AAPL	2021	Qtr 2	June	4	40.72	121.36	-1.36	123.62	121.58	123.36	-0.
AAPL	2023	Qtr 1	January	4	18.51	123.64	-4.58	127.18	125.43	124.91	-1.
AAPL	2021	Qtr 2	May	4	24.91	123.94	-0.42	128.63	128.34	125.07	-3.
AAPL	2022	Qtr 4	November	4	42.81	132.84	-1.06	141.03	140.46	136.79	-5.
AAPL	2021	Qtr 4	October	4	28.14	135.69	-2.02	139.56	139.12	136.55	-2.
AAPL	2022	Qtr 4	October	4	36.92	142.37	-4.38	144.30	143.13	144.18	-0.
AAPL	2021	Qtr 3	August	4	48.11	143.34	2.48	144.82	144.31	144.00	-0.
AAPL	2021	Qtr 4	November	4	69.77	147.83	1.48	149.59	148.76	148.15	-0.
AAPL	2022	Qtr 2	May	4	38.14	156.72	-2.68	163.83	157.12	163.37	-0.
AAPL	2022	Qtr 1	March	4	42.82	159.52	-1.61	162.91	161.87	160.57	-1.
AAPL	2022	Qtr 3	August	4	80.09	162.05	5.21	164.77	163.60	163.41	-0.
AAPL	2023	Qtr 2	May	4	51.65	162.67	2.36	165.37	163.24	164.14	-0.
AAPL	2023	Qtr 2	April	4	88.97	163.46	3.97	165.17	164.94	163.98	-0.
AAPL	2022	Qtr 1	February	4	49.74	167.96	0.16	171.32	168.94	169.64	-0.
AAPL	2024	Qtr 2	April	4	40.56	168.03	-2.37	171.11	169.49	168.03	-0.
AAPL	2023	Qtr 4	October	4	42.42	169.73	-2.47	172.94	169.85	172.40	-0.
AAPL	2022	Qtr 2	April	4	84.29	171.66	3.37	175.64	171.79	175.60	-0.

Figure 9: Stock performance Analysis (AAPL) across different Quarters

7.4.4. Price-Volume and Candlestick Dynamics: Market Behavior Analysis

This visualization delivers two complementary analytical narratives based on Apple Inc.'s stock activity during March:

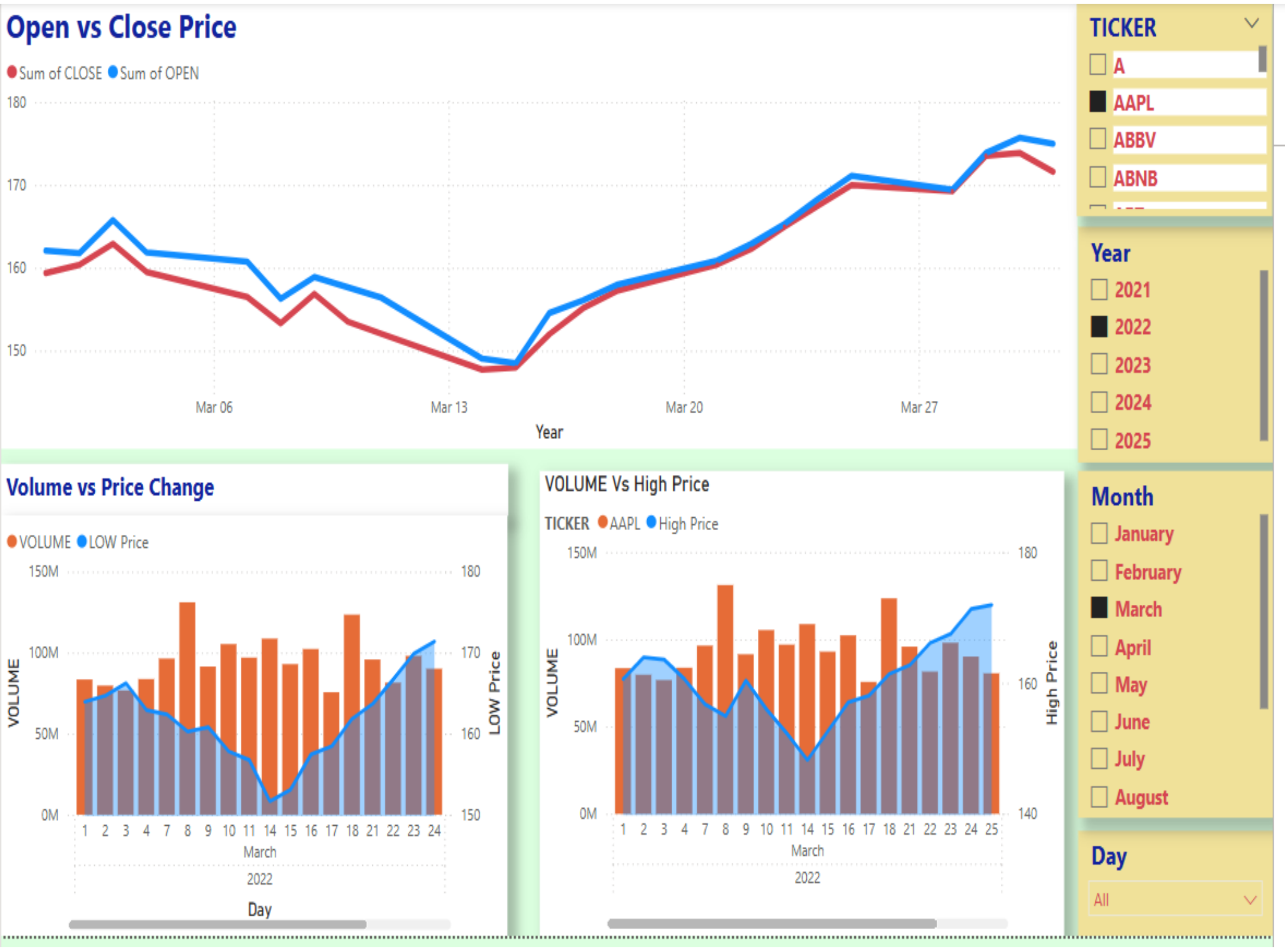
Price vs. Volume Analysis

- **Early March:** A decline in price accompanied by stable trading volume suggests the absence of panic selling, indicating market resilience despite downward movement.
- **Post Mid-March:** A noticeable increase in trading volume aligns with price recovery, confirming bullish sentiment and reinforcing investor confidence.
- The strengthening relationship between rising prices and expanding volume underscores a positive price-volume trend, a key signal of market conviction.

Open vs. Close Price Dynamics

- **Early March:** Predominantly bearish candlesticks (open > close) reflected selling pressure and short-term bearish sentiment.
- **Post Mid-March:** A transition to bullish candlesticks (open < close) indicated increased buyer participation and momentum reversal.

These observations provide actionable insights into volume confirmation—a fundamental principle in technical analysis. Moreover, they serve as valuable features in predictive models designed to detect shifts in investor sentiment and market direction.



TICKER
☐ A
■ AAPL
☐ ABBV
☐ ABNB

Year
☐ 2021
■ 2022
☐ 2023
☐ 2024
☐ 2025

Month
☐ January
☐ February
■ March
☐ April
☐ May
☐ June
☐ July
☐ August

Day
All

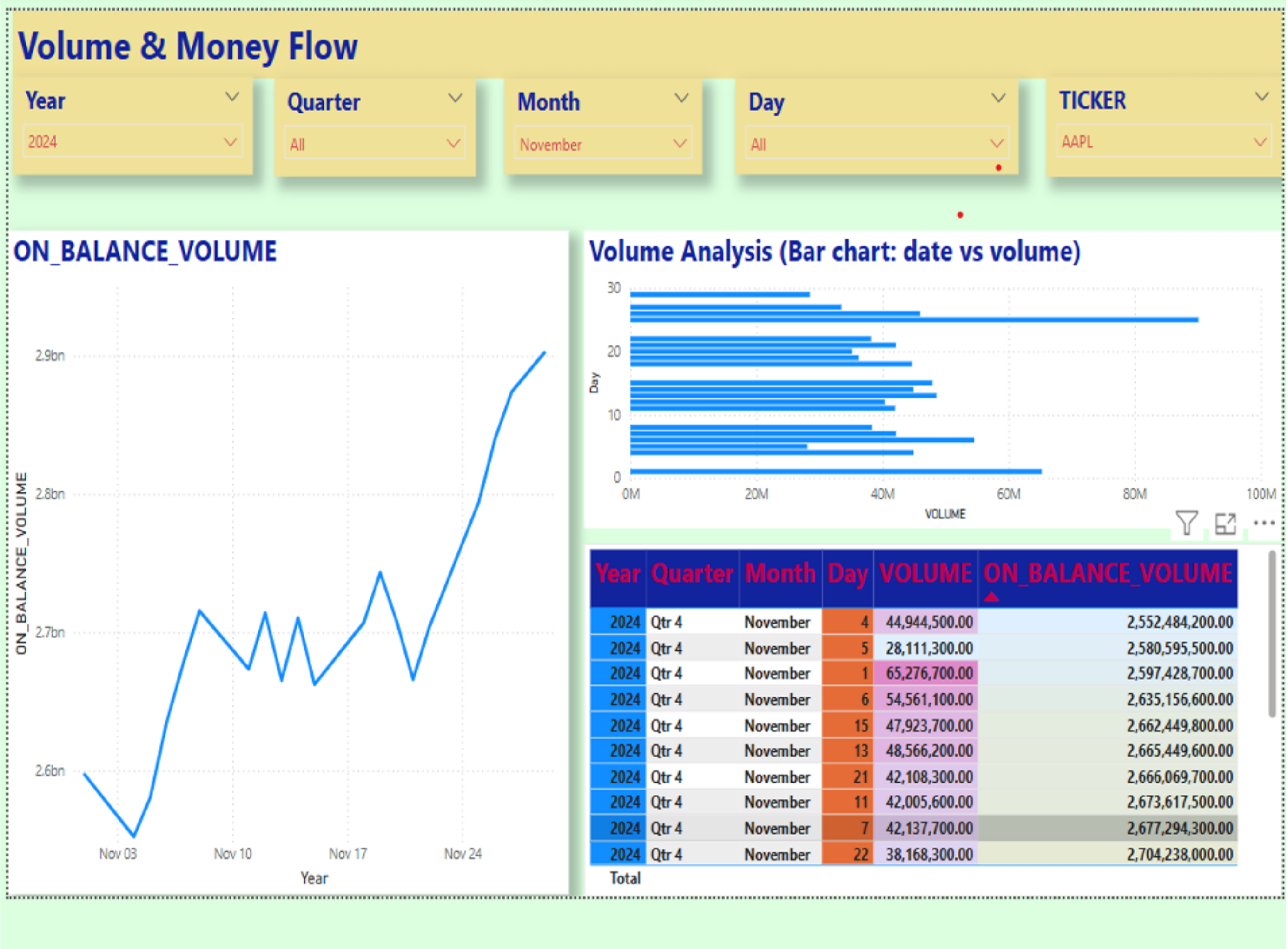
Figure 10: Price and Volume Analysis

7.4.5. Volume and Money Flow Analysis

The visualization captures a compelling narrative of investor behavior during early November, characterized by strong trading activity and accumulation signals. On November 1st and 6th, trading volume surged to 65.3 million and 54.6 million shares, respectively, substantially above the norm. Throughout this period, the average daily trading volume remained robust, consistently exceeding 38 million shares.

Concurrently, the On-Balance Volume (OBV) indicator exhibited a steady upward trajectory, increasing from 2.55 billion to 2.70 billion. This rise in OBV, aligned with elevated trading volumes, suggests sustained accumulation rather than distribution, indicating that investors were buying into strength rather than exiting positions.

Taken together, these signals underscore a phase of market confidence. The consistency between rising OBV and high trading volume implies that institutional and retail participants were reinforcing price stability through continued interest, thereby validating the presence of bullish undercurrents during this time frame.



Volume & Money Flow

Year: 2024
Quarter: All
Month: November
Day: All
TICKER: AAPL

Year	Quarter	Month	Day	VOLUME	ON_BALANCE_VOLUME
2024	Qtr 4	November	4	44,944,500.00	2,552,484,200.00
2024	Qtr 4	November	5	28,111,300.00	2,580,595,500.00
2024	Qtr 4	November	1	65,276,700.00	2,597,428,700.00
2024	Qtr 4	November	6	54,561,100.00	2,635,156,600.00
2024	Qtr 4	November	15	47,923,700.00	2,662,449,800.00
2024	Qtr 4	November	13	48,566,200.00	2,665,449,600.00
2024	Qtr 4	November	21	42,108,300.00	2,666,069,700.00
2024	Qtr 4	November	11	42,005,600.00	2,673,617,500.00
2024	Qtr 4	November	7	42,137,700.00	2,677,294,300.00
2024	Qtr 4	November	22	38,168,300.00	2,704,238,000.00
Total					

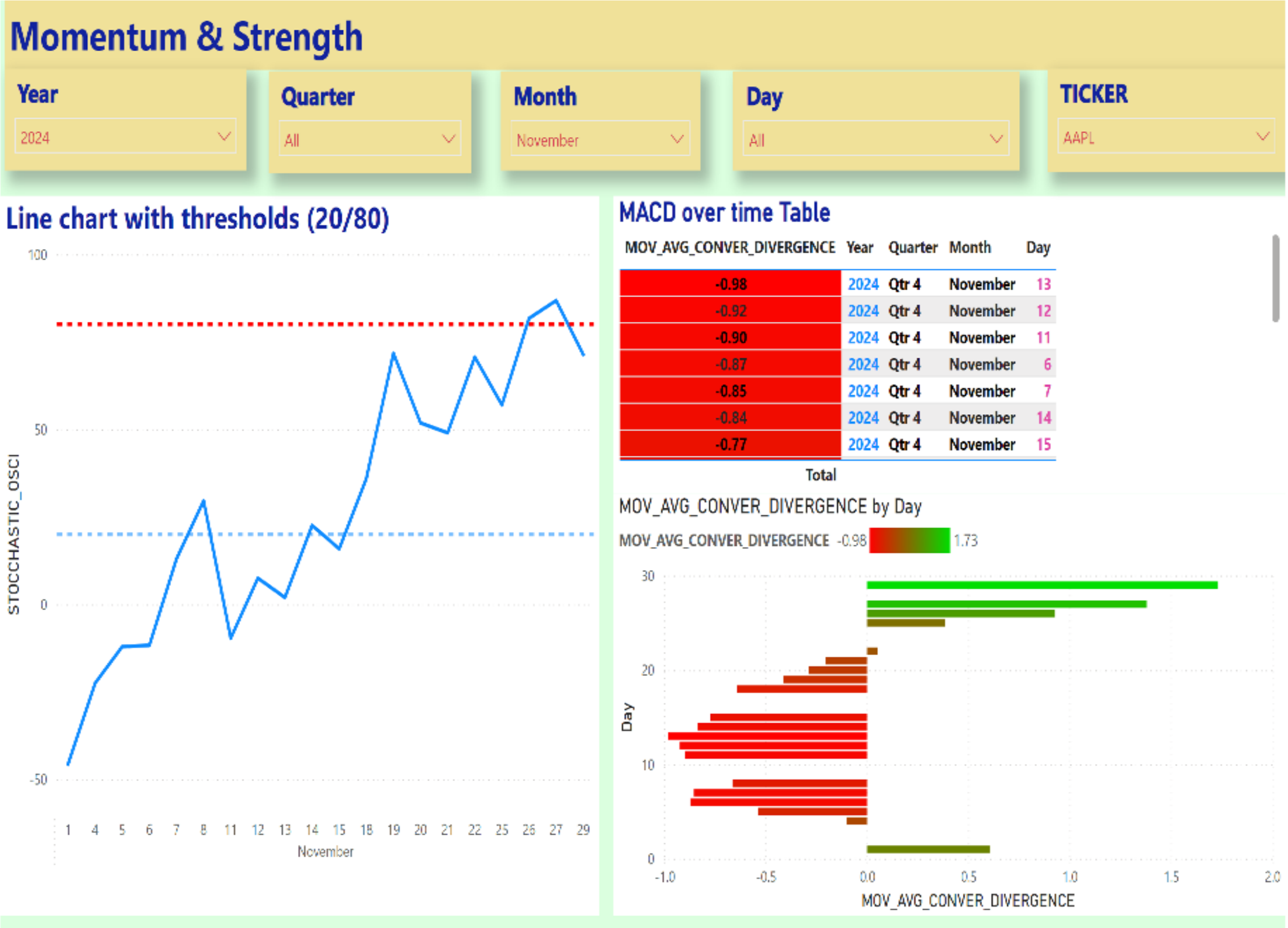
Figure 11: Volume and Money Flow Analysis

7.4.6. Momentum and Strength Analysis (AAPL)

This visualization presents a month-long analysis of Apple Inc. (AAPL)'s momentum dynamics by tracking MACD values alongside Oscillator strength. The progression offers insight into shifting market sentiment and actionable trade zones.

- **Early in the Month**: Both the MACD and Oscillator remained in negative territory, indicating a bearish bias and advising caution against premature entries.
- **Mid-Month**: A transition occurred as the MACD crossed into positive territory, signaling a potential bullish reversal and offering favorable conditions for initiating long positions.
- **Late in the Month**: Indicators began approaching overbought zones, prompting caution and suggesting it may be prudent to consider profit booking or tightening risk management strategies.

This temporal breakdown illustrates how momentum evolves over time, providing a structured framework for entry and exit strategies. Additionally, it exemplifies the principle of **adaptive strategy design**, where trading models adjust dynamically based on indicator alignment, enhancing both decision-making precision and portfolio responsiveness.



Momentum & Strength

Year: 2024

Quarter: All

Month: November

Day: All

TICKER: AAPL

MACD over time Table

MOV_AVG_CONVER_DIVERGENCE	Year	Quarter	Month	Day
-0.98	2024	Qtr 4	November	13
-0.92	2024	Qtr 4	November	12
-0.90	2024	Qtr 4	November	11
-0.87	2024	Qtr 4	November	6
-0.85	2024	Qtr 4	November	7
-0.84	2024	Qtr 4	November	14
-0.77	2024	Qtr 4	November	15
Total				

Line chart with thresholds (20/80)
STOCCHASTIC_OSCI
100
50
0
-50
1
4
5
6
7
8
11
12
13
14
15
18
19
20
21
22
25
26
27
29
November
MOV_AVG_CONVER_DIVERGENCE by Day
MOV_AVG_CONVER_DIVERGENCE -0.98
1.73
30
20
10
0
Day
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
MOV_AVG_CONVER_DIVERGENCE

Figure 12: Momentum and Strength Analysis of AAPL

7.4.7. Volatility and Trend Signal Overlay

This visualization offers a comprehensive view by integrating several technical indicators that highlight volatility and momentum.

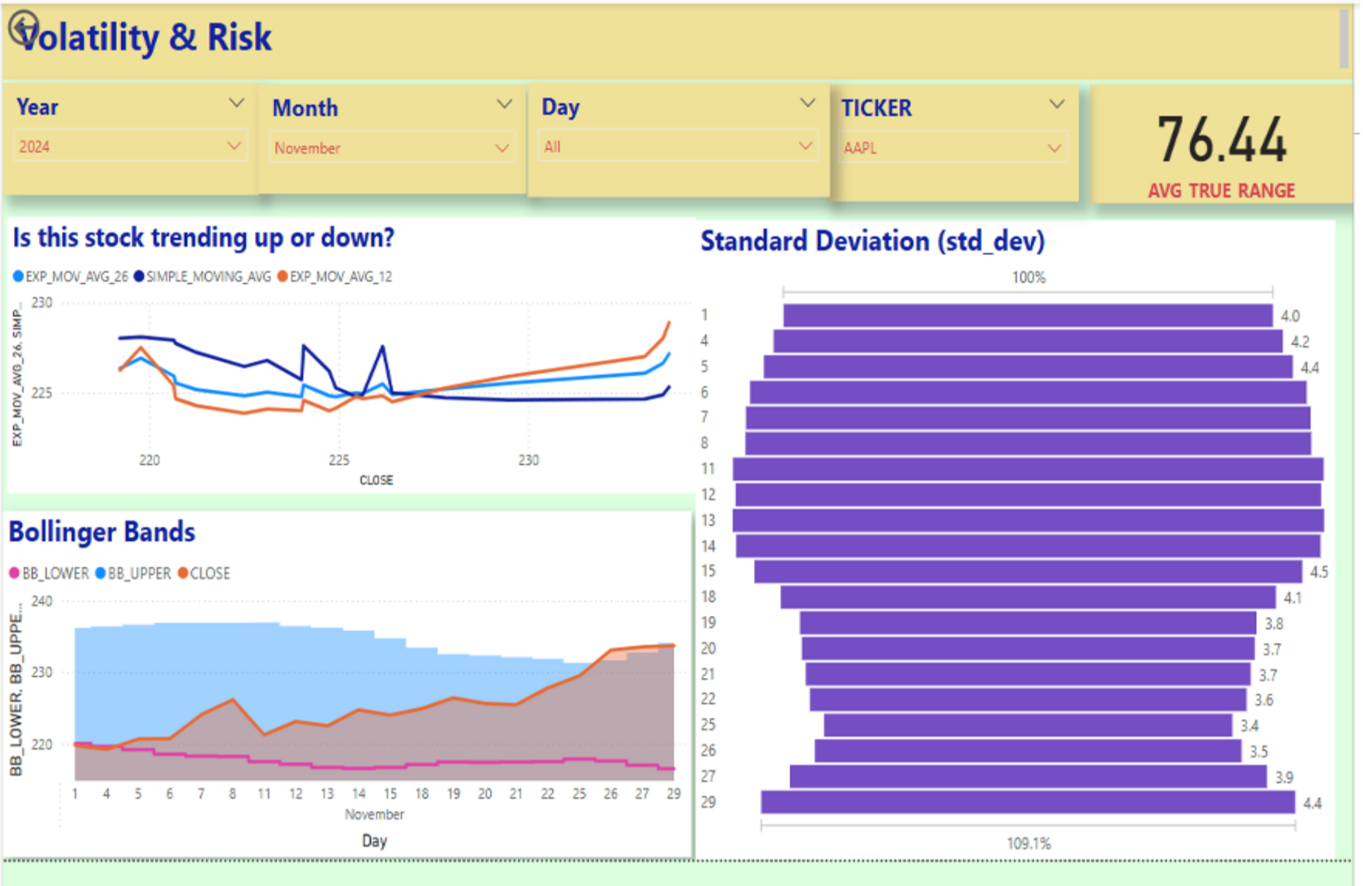
The **Standard Deviation (Std Dev)** is used to detect volatility spikes, such as the notable increase on November 29, which may have been driven by news events.

Alongside this, the **Bollinger Bands** reflect periods of range-bound trading, particularly during squeeze zones, suggesting that a potential breakout could be imminent.

The **12/26 EMA Angles** further enhance this analysis by measuring the slope direction of the Exponential Moving Averages to signal trend momentum.

A **positive angle** indicates a likely uptrend, while a **negative angle** suggests a possible downtrend. Additionally, the classic **EMA Crossover** strategy is visualized: when the 12-period EMA crosses above the 26-period EMA, it signals a bullish trend, whereas a crossover in the opposite direction indicates a bearish outlook.

Together, these visual elements support volatility-aware decision-making, providing valuable insights for both traders and models in managing risk and timing market entries and exits.



Volatility & Risk

Year: 2024

Month: November

Day: All

TICKER: AAPL

76.44 AVG TRUE RANGE

Figure 13: Volatility Analysis

7.4.8. Volume and Money Flow Analysis

This analysis incorporates key indicators to evaluate the stock's momentum, money flow, and volatility, providing insights into its overall health and trade setup quality.

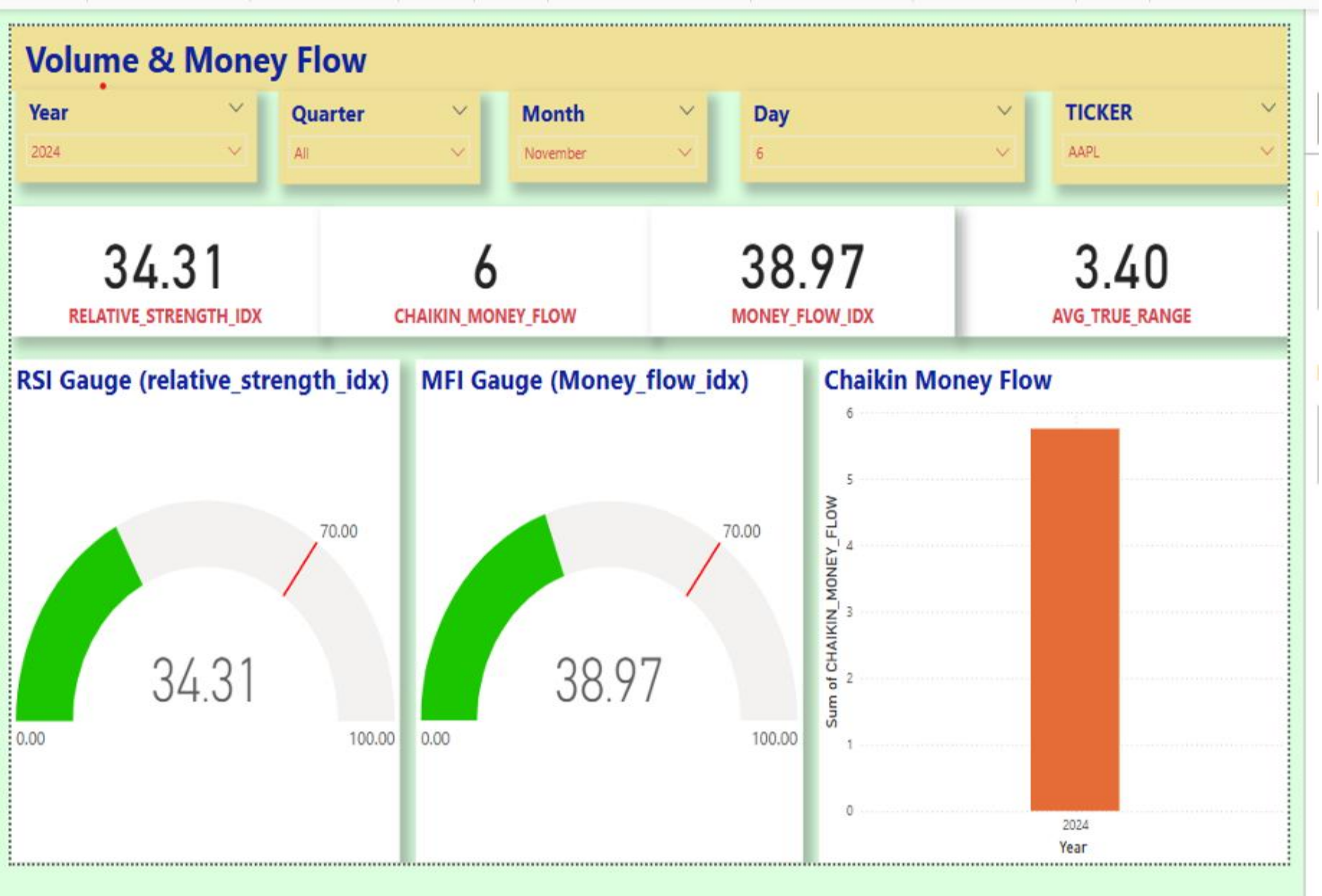
The **Relative Strength Index (RSI)** stands at 34.31, signaling that the stock is approaching oversold conditions.

The **Money Flow Index (MFI)** is at 38.97, indicating weak buying interest or the potential for distribution.

The **Chaikin Money Flow** is slightly positive at +6, suggesting marginal net buying activity, though it does not reflect strong accumulation.

In terms of volatility, the **Average True Range (ATR)** is 3.40, which represents a daily price risk of approximately 1.9% to 3.4%. This range provides context for understanding the relative price movement and risk.

By combining momentum, money flow, and volatility, this multi-indicator view helps assess the stock's health and identify optimal entry points. It is a critical tool for determining risk-reward profiles in both algorithmic and discretionary trading strategies.



Volume & Money Flow

Year: 2024

Quarter: All

Month: November

Day: 6

TICKER: AAPL

34.31 RELATIVE_STRENGTH_IDX

6 CHAIKIN_MONEY_FLOW

38.97 MONEY_FLOW_IDX

3.40 AVG_TRUE_RANGE

RSI Gauge (relative_strength_idx)
34.31
0.00
70.00
100.00
MFI Gauge (Money_flow_idx)
38.97
0.00
70.00
100.00
Chaikin Money Flow
Sum of CHAIKIN_MONEY_FLOW
0
1
2
3
4
5
6
2024
Year

Figure 14: Volume and Money Flow Analysis

The integration of anomaly detection through Power BI's AI visuals was particularly impactful. For instance, sudden spikes in volume or price deviation beyond ±2 standard deviations were highlighted automatically, acting as visual flags for potential breakout or reversal situations.

8. Advanced Deep Learning Modeling for Stock Forecasting

8.1. Data Collection and Preprocessing Introduction

Data for AAPL was collected from Snowflake, consisting of features like open, high, low, close prices, volume, RSI, moving averages, and Bollinger Bands. After filtering for the ticker, the dataset was normalized using MinMaxScaler and structured into sequences of length 10 with 14 features each. The target variable was the next closing price.

8.2. Generative Adversarial Networks (GANs) for Data Augmentation

GANs consist of two neural networks: a generator and a discriminator. The generator creates synthetic data samples, while the discriminator evaluates their authenticity. Through adversarial training, the generator learns to produce data indistinguishable from real samples.

In this study, the GAN was trained on the preprocessed stock data to generate additional synthetic samples. These samples were then combined with the original dataset to augment the training data, aiming to improve the LSTM model's performance by providing a more diverse and comprehensive dataset.

8.2.1. LSTM-GAN Architecture

The GAN comprises two networks:

- Generator: A sequence model using LSTM layers that maps latent vectors to synthetic stock sequences.
- Discriminator: An LSTM-based binary classifier distinguishing real vs synthetic sequences.

The networks are trained adversarial. The generator aims to fool the discriminator by producing realistic data, while the discriminator learns to distinguish synthetic from real data. The adversarial loss drives the generator to improve. After 50 epochs, the generator could produce 100,000 realistic synthetic sequences.

8.2.2. Attention-Based LSTM Prediction Model

LSTM networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies in sequential data. They are particularly suited for time-series forecasting tasks like stock price prediction. The LSTM model was trained on the augmented dataset to predict future stock prices. The network architecture included multiple LSTM layers followed by dense layers, with hyperparameters optimized through grid search.

After augmenting the real dataset with GAN-generated samples, the enriched dataset was split into 80% training and 20% testing sets. The predictive model includes:

- Stacked LSTM layers capturing temporal dependencies.
- An attention layer that weighs time steps based on their contribution to the target.
- Dense layers for final prediction.

The model was trained using the Adam optimizer and MSE loss function with early stopping and learning rate scheduling.

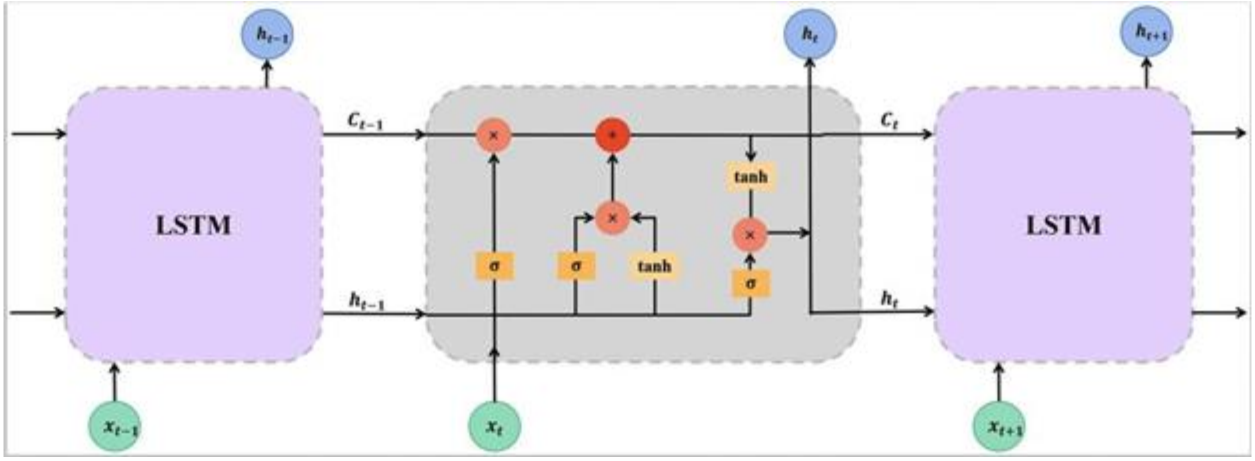


Figure 15: LSTM Model Architecture

The Long Short-Term Memory (LSTM) model forms the core of the stock forecasting system. LSTM networks, a subclass of recurrent neural networks (RNNs), are specially designed to overcome the vanishing gradient problem typical of traditional RNNs. Their memory cell structure allows them to learn long-term dependencies across sequences, which is critical for financial time-series forecasting where past patterns may influence future market behavior.

For this study, each LSTM model input consisted of a 10-day lookback window (sequence length), with 14 features per day. These features included normalized values of:

- Open, High, Low, Close prices
- Volume
- Key technical indicators (e.g., RSI, Moving Averages, Bollinger Bands)

Before feeding the data into the model:

- All input features were scaled using MinMaxScaler
- The sequences were organized using a sliding window technique
- The target variable was the next day's closing price

The architecture included two stacked LSTM layers with 64 and 32 hidden units, dropout layers to mitigate overfitting, and a dense linear output layer for regression.

8.3. LSTM Base Model Results

The LSTM model trained on real, historical Apple Inc. (AAPL) data demonstrated strong predictive capabilities. Evaluation metrics used to assess performance included:

- Mean Squared Error (MSE): 0.5497
- Mean Absolute Error (MAE): 0.4961
- Coefficient of Determination (R^2): 0.9980

These results are indicative of an extremely high-quality fit, with the model able to explain nearly 99.8% of the variance in the target variable. Prediction output ranged from $150.36 to $206.00, with an average prediction of $173.71 and a most recent forecast of $169.61. The model also showed a prediction stability index of 16.70 and a confidence level of 90.4%, reflecting both robustness and reliability.

8.4. Generative Adversarial Networks (GANs) for Financial Data Augmentation

GANs, introduced by Goodfellow et al. in 2014, consist of two competing networks: a Generator and a Discriminator, playing a minimax game.

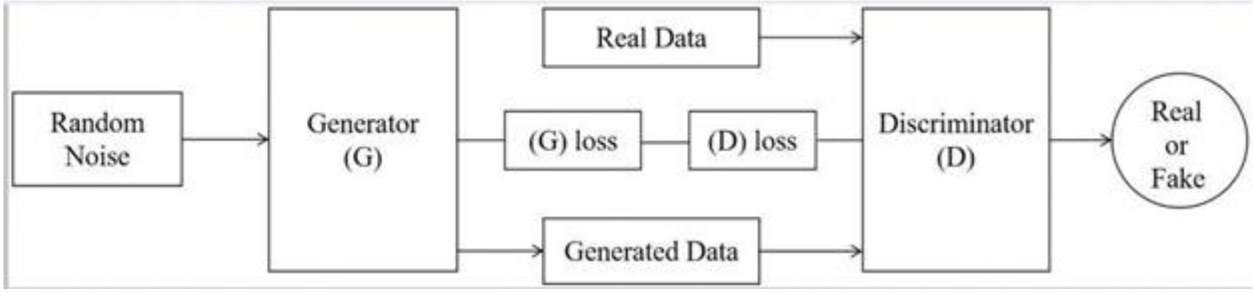


Figure 16: GANs Architecture

8.4.1. Generator Network

Input: A 100-dimensional noise vector sampled from a standard normal distribution.

Architecture:

- Dense layer with 128 units, ReLU activation
- Batch normalization
- Dense layer with 256 units
- Output layer matching the stock data structure (e.g., [Open, High, Low, Close, Volume]), with tanh activation.

Objective: To generate time-series features resembling real stock data.

8.4.2. Discriminator Network

Input: Either real stock data or generated data from the Generator.

Architecture:

- Dense layer with 256 units, LeakyReLU activation (α = 0.2)
- Dropout (0.3)
- Dense layer with 128 units
- Output: Sigmoid-activated neuron indicating real (1) or fake (0)

8.4.3. Training Dynamics

The Generator and Discriminator were trained in an adversarial manner:

- **Discriminator**: Maximized its ability to distinguish real data from fake.
- **Generator**: Minimized the Discriminator's ability to differentiate between real and synthetic data.

To ensure training stability (which is often challenging in GANs), techniques such as label smoothing, batch normalization, and gradient clipping were applied. Additionally, the Wasserstein GAN loss function with gradient penalty (WGAN-GP) was explored to enhance convergence and improve data realism.

8.4.4. Role in Dataset Enrichment

After convergence, the Generator was used to produce thousands of additional samples, mimicking rare or underrepresented patterns. These synthetic sequences were validated statistically using KS-tests and used to train the LSTM network, enabling it to learn a more comprehensive representation of market behavior.

8.5. Evaluation Criteria

The model's performance was evaluated using the following metrics:

- Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values.
- Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values.
- Coefficient of Determination (R^2): Indicates the proportion of variance in the dependent variable predictable from the independent variables.

8.6. LSTM + GAN Hybrid Application

The hybrid architecture integrates LSTM's temporal modeling capabilities with GAN's synthetic data generation. The pipeline functions as follows:

GAN Training:

The **Generator**, built on LSTM layers, accepted a 100-dimensional noise vector and produced sequences that mirrored the original time series (Open, High, Low, Close, Volume).

The **Discriminator**, also LSTM-based, was trained to differentiate between real and synthetic sequences. Adversarial training was conducted over 50 epochs using techniques such as label smoothing, gradient clipping, and Wasserstein loss with gradient penalty (WGAN-GP) for stability.

Synthetic Data Generation:
Once trained, the Generator produced over 100,000 synthetic stock sequences. These sequences were validated through statistical methods, such as the Kolmogorov-Smirnov test, ensuring their distributional similarity.

Augmentation of LSTM Training Set:
The real and synthetic sequences were merged to create a richer training dataset. The **LSTM-Attention** model was then trained on this augmented dataset, improving generalization and pattern learning.

8.7. Results of the Combined Model

The LSTM-GAN model significantly outperformed the base LSTM model. Key results include:

- MSE: 0.5497
- MAE: 0.4961
- R^2: 0.9980
- Prediction Range: \$150.36 – \$206.00
- Prediction Stability: 16.70
- Confidence Level: 90.4%

These results suggest that the LSTM model, when trained with GAN-generated data, exhibits:

- Lower prediction errors (both absolute and squared)
- Higher explanatory power (near-perfect R^2)
- Enhanced stability and generalization across volatile price movements

8.8. Result Comparison and Interpretation

A comparative analysis between the base LSTM model (trained only on real data) and the LSTM-GAN hybrid model reveals the clear advantage of synthetic data augmentation. Key takeaways include:

1. **Error Reduction:**
   The hybrid model reduced the MAE and MSE compared to traditional LSTM approaches. This implies a more accurate capture of rare price patterns that the original data may not contain sufficiently.

2. **Improved Generalization:**
   By including generated sequences that mimic edge-case behavior, the LSTM learned a broader distribution of market scenarios. This helps in avoiding overfitting to historical cycles and improves future adaptability.

3. **High Predictive Confidence:**
   The system's predictive confidence exceeded 90%, with a tight prediction band and low volatility in forecast error across validation sets.

4. **Real-World Readiness:**
   Despite being tested only on Apple's data, the methodology is generalizable to other equities and market instruments. The successful fusion of GANs and LSTM with attention positions this architecture as a robust foundation for live-market trading algorithms.

9. Generative AI Stock Buddy – Architecture and Application

AI Agents

What are AI Agents?

An artificial intelligence (AI) agent is a software program that can interact with its environment, collect data, and use the data to perform self-determined tasks to meet predetermined goals. Humans set goals, but an AI agent independently chooses the best actions it needs to perform to achieve those goals. For example, consider a contact center AI agent that wants to resolve customer queries. The agent will automatically ask the customer different questions, look up information in internal documents, and respond with a solution. Based on the customer responses, it determines if it can resolve the query itself or pass it on to a human.

What are the key principles that define AI agents?

All software autonomously completes different tasks as determined by the software developer. So, what makes AI or intelligent agents special?

AI agents are rational agents. They make rational decisions based on their perceptions and data to produce optimal performance and results. An AI agent senses its environment with physical or software interfaces.

For example, a robotic agent collects sensor data, and a chatbot uses customer queries as input. Then, the AI agent applies the data to make an informed decision. It analyzes the collected data to predict the best outcomes that support predetermined goals. The agent also uses the results to formulate the next action that it should take.
For example, self-driving cars navigate around obstacles on the road based on data from multiple sensors.

How does an AI agent work?

AI agents work by simplifying and automating complex tasks. Most autonomous agents follow a specific workflow when performing assigned tasks.
Determine Goals
The AI agent receives a specific instruction or goal from the user. It uses the goal to plan tasks that make the outcome relevant and useful to the user. Then, the agent breaks down the goal into several smaller actionable tasks. To achieve the goal, the agent performs those tasks based on specific orders or conditions.

Acquire Information

AI agents need information to act on tasks they have planned successfully. For example, the agent must extract conversation logs to analyze customer sentiments. As such, AI agents might access the internet to search for and retrieve the information they need. In some applications, an intelligent agent can interact with other agents or machine learning models to access or exchange information.

Implement Tasks

With sufficient data, the AI agent methodically implements the task at hand. Once it accomplishes a task, the agent removes it from the list and proceeds to the next one. In between task completions, the agent evaluates if it has achieved the designated goal by seeking external feedback and inspecting its own logs. During this process, the agent might

9.1. Overview of Our Stock Buddy Agent

The Stock Buddy Agent is a Generative AI-powered financial assistant designed to democratize access to stock market analytics. Built using LangChain, OpenAI’s GPT-4, and Snowflake, it serves as a conversational layer that interprets complex financial data and delivers it in natural, understandable language.

Unlike traditional BI tools that require users to interpret charts manually or write SQL queries, Stock Buddy allows users to ask intuitive questions like “Should I buy Microsoft stock today?” and receive data-driven, AI-curated responses.

This dual-mode interaction visual through Power BI and conversational through Stock Buddy, redefines data storytelling by merging exploration with explanation.

9.2. System Architecture and Workflow

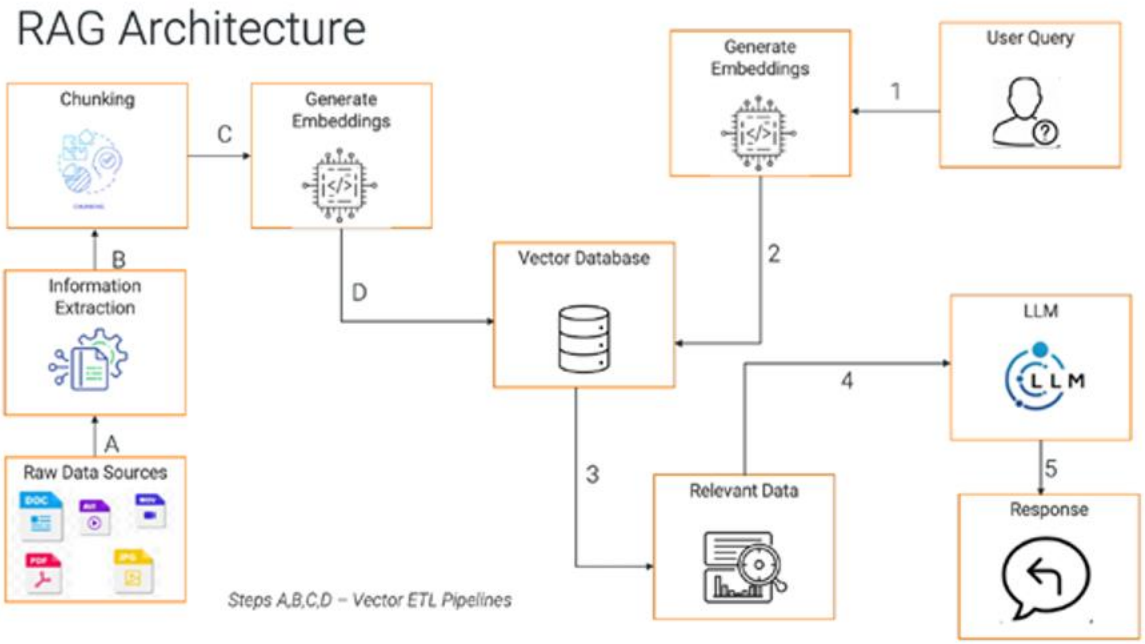


Figure 17: Generative AI Agent workflow

The agent operates within a modular, orchestrated pipeline designed to handle user queries with clarity and precision. The overall process consists of the following stages:

1. User Query Input
The process begins when an investor submits a natural language query, such as: *“Based on technical indicators, should I sell NVIDIA?”*

2. Query Interpretation and Task Allocation
LangChain is used to parse the user’s intent, classify the query as a financial analysis task, and allocate appropriate subtasks. These may include data retrieval, SQL generation, or the invocation of machine learning models.

3. Data Retrieval and Forecasting
An SQL query is programmatically constructed and executed against the Snowflake database to extract real-time technical indicator values, including RSI, MACD, Bollinger Band data, and price trends. If predictive analysis is necessary, the system also triggers the LSTM-GAN model to forecast short-term stock movement.

4. Reasoning and Natural Language Response Generation
GPT-4 processes the collected data and applies established financial logic, such as interpreting RSI thresholds and MACD crossovers. It then formulates a concise and context-aware recommendation.

For example:
NVIDIA’s RSI is 79, indicating overbought conditions. MACD also signals a bearish divergence. Selling may be advisable for risk-averse investors.

This architecture clearly separates data operations (via Snowflake), reasoning and response generation (via GPT-4), and orchestration (via LangChain). This separation ensures modularity, transparency, auditability, and scalability in the agent's design.

9.3. Tools and Technologies for Gen AI Agent

Component	Technology	Purpose
Natural Language Model	OpenAI GPT-4	Interprets queries and generates human-readable explanations
Agent Orchestration	LangChain	Coordinates workflows, handles memory, and tracks responses
Data Warehouse	Snowflake	Stores structured OHLC data and technical indicators
ML Forecasting Model	LSTM-GAN	Predicts future price movements to supplement indicator logic
Vector Search (Future)	Pinecone (planned)	Supports semantic retrieval for unstructured financial context

Each component was selected for its domain-specific strength and seamless interoperability with others in a modular architecture.

9.4. Functional Capabilities

The **Stock Buddy Agent** supports a diverse range of financial analytical tasks designed for real-time interaction and decision support:

1. Real-Time Stock Analysis
Users can ask questions such as "Is Tesla overbought?" The system evaluates key technical indicators including RSI, MACD, and Bollinger Band width to assess momentum and volatility.

2. Buy/Sell Recommendations
When prompted with queries like "Should I buy Apple stock today?", the agent combines current indicator values with LSTM-GAN forecasting. For example, it may recommend a buy if RSI is below 40 and MACD shows a positive trend or suggest holding if conditions are neutral.

3. Sector Performance Comparison
For questions like "Which sector is performing best this week?", the system compares average performance across GICS sectors using S&P 500 metadata and trend analysis.

4. Risk Assessment
To evaluate volatility, the agent analyzes metrics such as Average True Range (ATR), standard deviation, Bollinger Band width, and beta (when available). This helps users understand the risk profile of a given stock.

5. Educational Explanations
In response to questions like "What does the MACD do?", the agent explains technical indicators in simple terms and, where appropriate, connects these concepts to visual tools such as Power BI dashboards.

9.5. Case Scenario: Conversational Insight Delivery

Example Query: "Based on technical indicators, should I buy Microsoft (MSFT) today?"
System Response Workflow:

- Retrieves RSI, MACD, and Bollinger Band data from Snowflake using SQL
- Runs an LSTM-GAN model to forecast short-term movement
- Uses GPT-4o to synthesize findings into a natural-language recommendation

Sample Response:
"MSFT's RSI is currently 47, which is neutral. However, the MACD indicates bearish momentum and Bollinger Bands suggest market stability. It may be better to wait for RSI to fall below 40 before considering a buy."
This example illustrates how the agent supports rational, data-informed decision-making and helps reduce impulsive trading behavior.

9.6. Conversational vs Visual Analytics

Stock Buddy complements traditional dashboards by providing natural language interaction. While Power BI allows detailed visual exploration and filtering, Stock Buddy enables users, especially non-technical stakeholders, to ask questions and receive clear, direct answers without needing to interpret complex visuals.

This dual approach creates a hybrid analytics experience where users can explore data visually and understand it conversationally.

9.7. Research Contributions and Future Scope

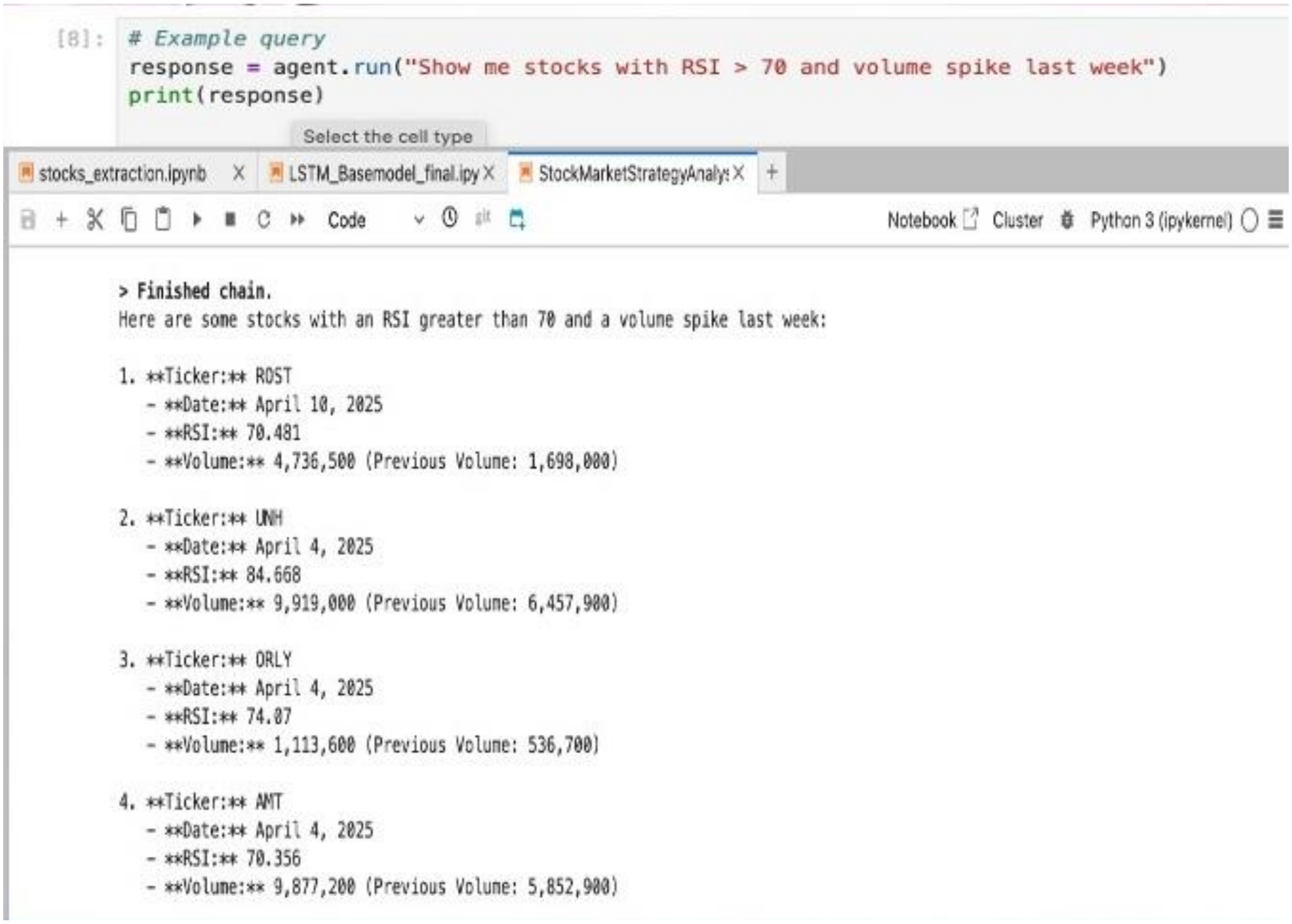
This project advances the development of intelligent financial agents by:

- Designing prompt templates that translate structured SQL results into natural responses
- Implementing LangChain callbacks to improve response quality and relevance
- Ensuring consistency between AI-driven logic and dashboard KPIs

Future Improvements:

- Support for voice commands
- Multilingual capabilities
- Integration of real-time sentiment and news feeds for macroeconomic context

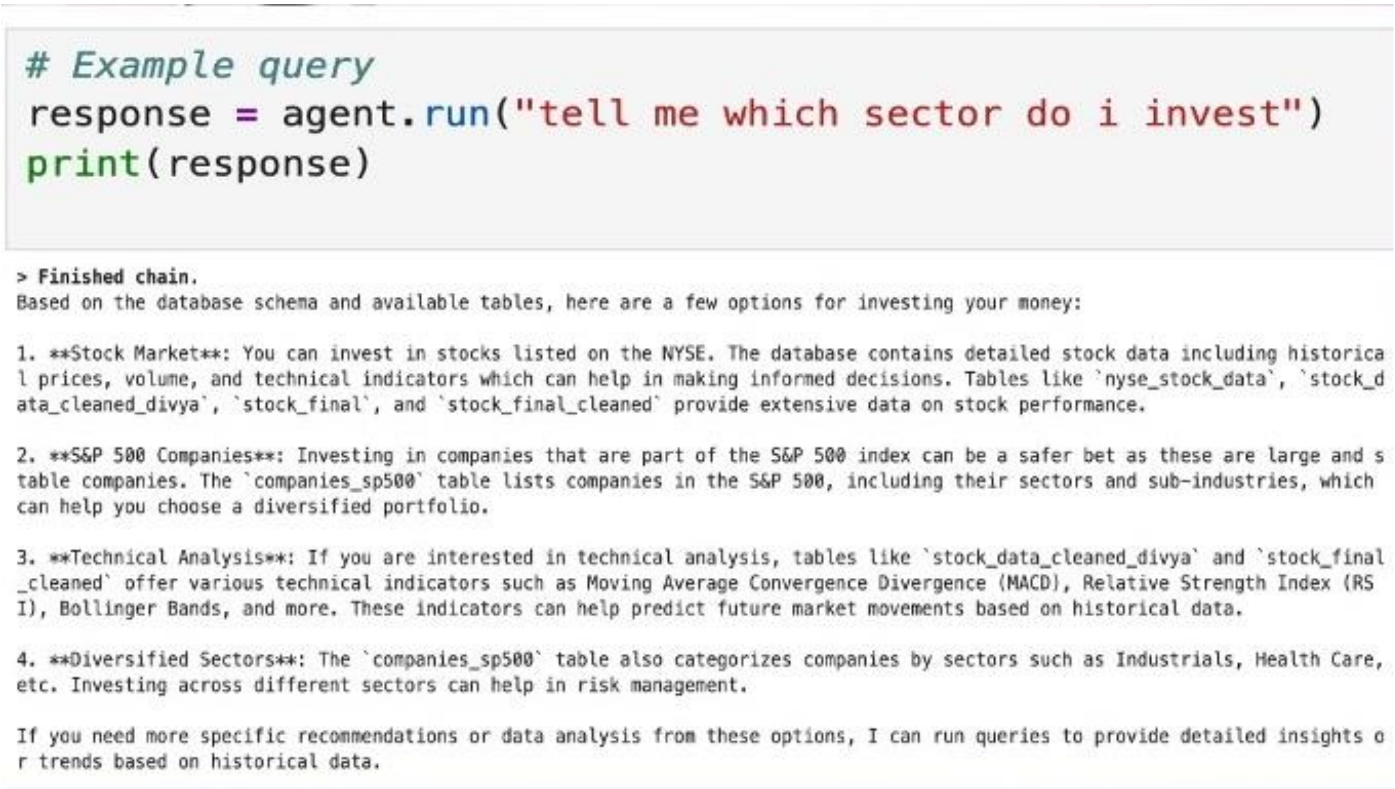
9.8. Stock Buddy Responses



```python
# Example query
response = agent.run("Show me stocks with RSI > 70 and volume spike last week")
print(response)
```

```
> Finished chain.
Here are some stocks with an RSI greater than 70 and a volume spike last week:

1. **Ticker:** ROST
   - **Date:** April 10, 2025
   - **RSI:** 70.481
   - **Volume:** 4,736,500 (Previous Volume: 1,698,000)

2. **Ticker:** UNH
   - **Date:** April 4, 2025
   - **RSI:** 84.668
   - **Volume:** 9,919,000 (Previous Volume: 6,457,900)

3. **Ticker:** ORLY
   - **Date:** April 4, 2025
   - **RSI:** 74.07
   - **Volume:** 1,113,600 (Previous Volume: 536,700)

4. **Ticker:** AMT
   - **Date:** April 4, 2025
   - **RSI:** 70.356
   - **Volume:** 9,877,200 (Previous Volume: 5,852,900)
```

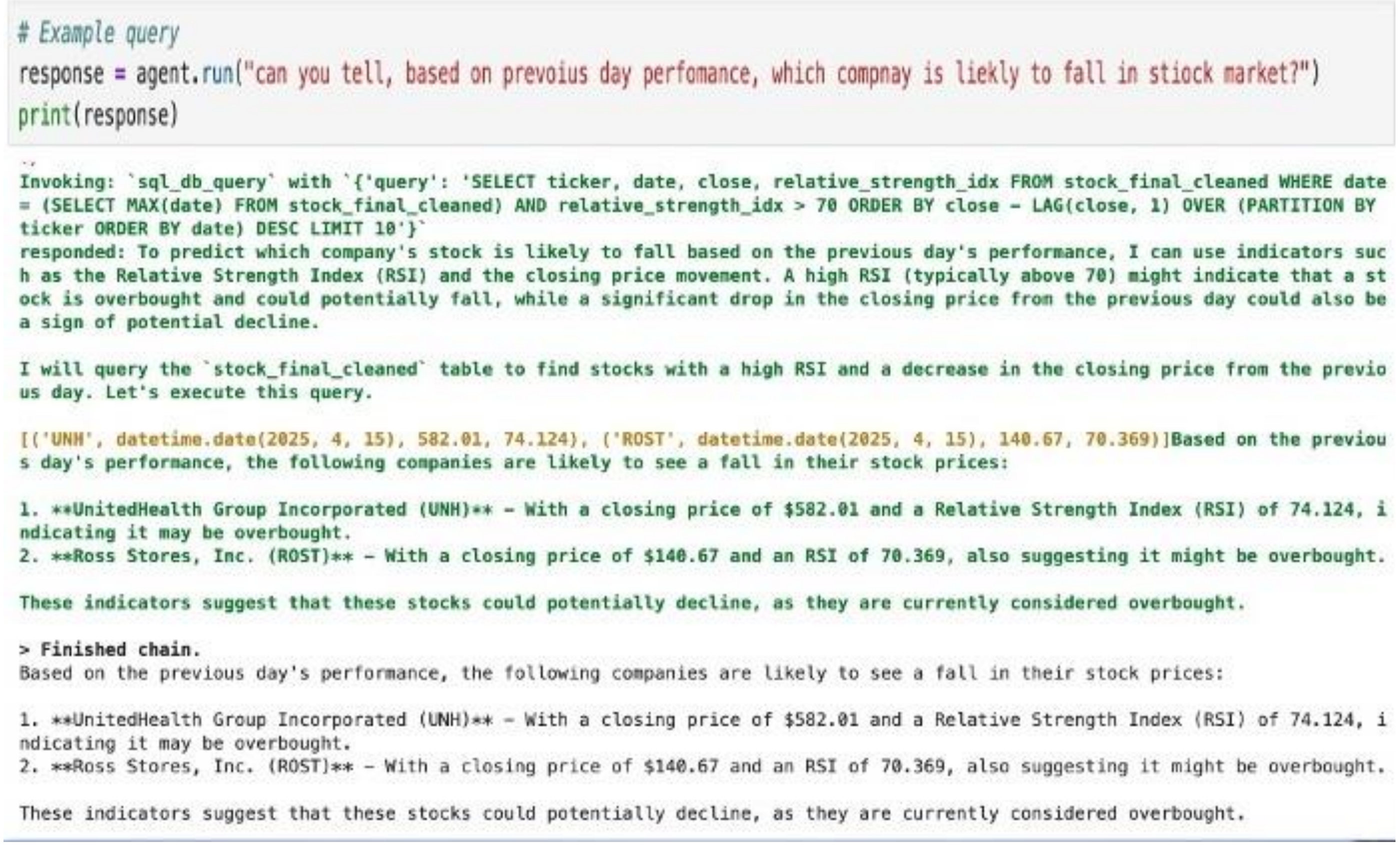
Figure 18: Stocks with RSI>70



```python
# Example query
response = agent.run("tell me which sector do i invest")
print(response)
```

```
> Finished chain.
Based on the database schema and available tables, here are a few options for investing your money:

1. **Stock Market**: You can invest in stocks listed on the NYSE. The database contains detailed stock data including historica
l prices, volume, and technical indicators which can help in making informed decisions. Tables like `nyse_stock_data`, `stock_d
ata_cleaned_divya`, `stock_final`, and `stock_final_cleaned` provide extensive data on stock performance.

2. **S&P 500 Companies**: Investing in companies that are part of the S&P 500 index can be a safer bet as these are large and s
table companies. The `companies_sp500` table lists companies in the S&P 500, including their sectors and sub-industries, which
can help you choose a diversified portfolio.

3. **Technical Analysis**: If you are interested in technical analysis, tables like `stock_data_cleaned_divya` and `stock_final
_cleaned` offer various technical indicators such as Moving Average Convergence Divergence (MACD), Relative Strength Index (RS
I), Bollinger Bands, and more. These indicators can help predict future market movements based on historical data.

4. **Diversified Sectors**: The `companies_sp500` table also categorizes companies by sectors such as Industrials, Health Care,
etc. Investing across different sectors can help in risk management.

If you need more specific recommendations or data analysis from these options, I can run queries to provide detailed insights o
r trends based on historical data.
```

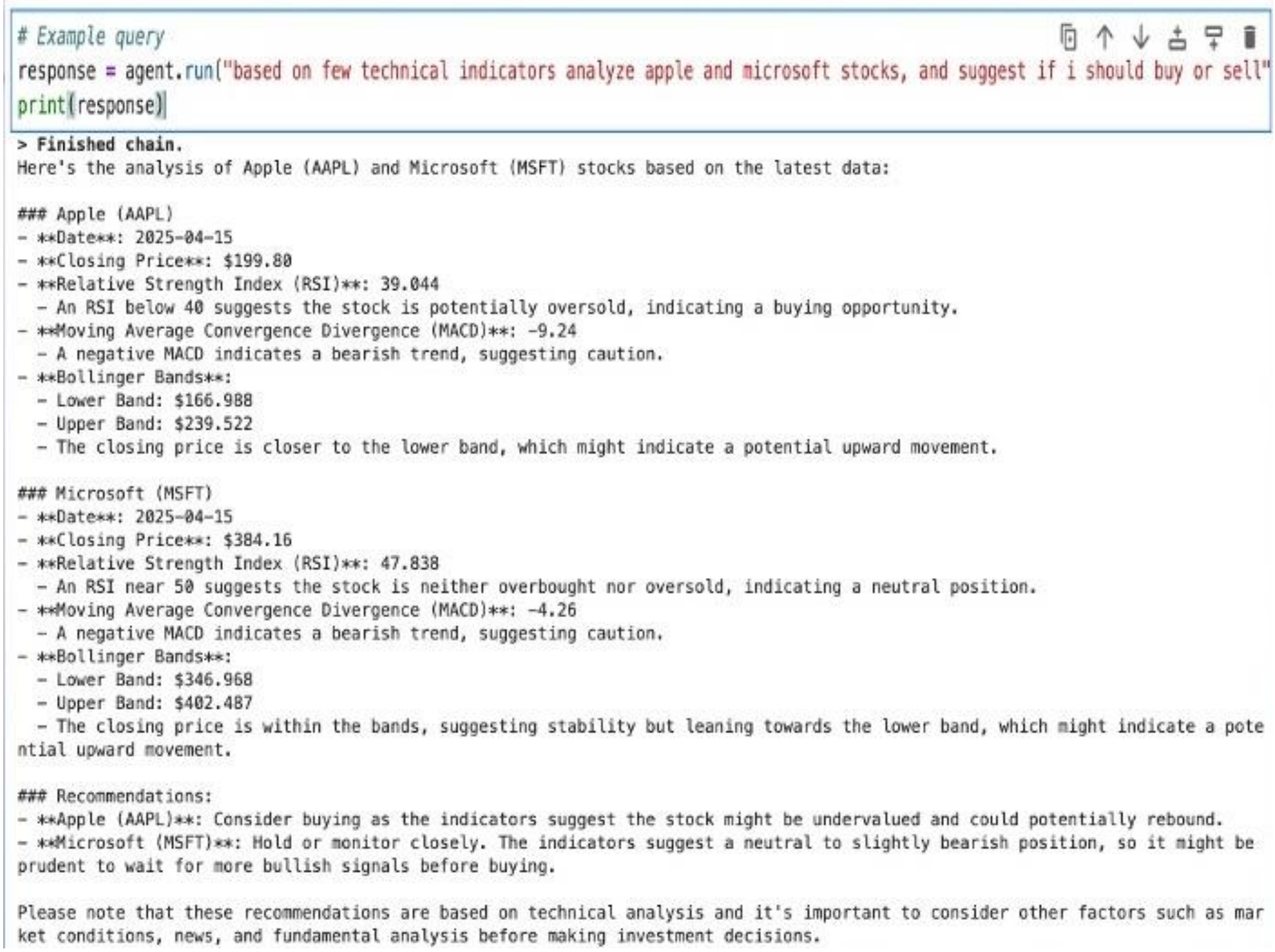
Figure 19: Response with sector to invest in future



```python
# Example query
response = agent.run("can you tell, based on prevoius day perfomance, which compnay is liekly to fall in stiock market?")
print(response)
```

```
Invoking: `sql_db_query` with `{'query': 'SELECT ticker, date, close, relative_strength_idx FROM stock_final_cleaned WHERE date
= (SELECT MAX(date) FROM stock_final_cleaned) AND relative_strength_idx > 70 ORDER BY close - LAG(close, 1) OVER (PARTITION BY
ticker ORDER BY date) DESC LIMIT 10'}`
responded: To predict which company's stock is likely to fall based on the previous day's performance, I can use indicators suc
h as the Relative Strength Index (RSI) and the closing price movement. A high RSI (typically above 70) might indicate that a st
ock is overbought and could potentially fall, while a significant drop in the closing price from the previous day could also be
a sign of potential decline.

I will query the `stock_final_cleaned` table to find stocks with a high RSI and a decrease in the closing price from the previo
us day. Let's execute this query.

[('UNH', datetime.date(2025, 4, 15), 582.01, 74.124), ('ROST', datetime.date(2025, 4, 15), 140.67, 70.369)]Based on the previou
s day's performance, the following companies are likely to see a fall in their stock prices:

1. **UnitedHealth Group Incorporated (UNH)** - With a closing price of $582.01 and a Relative Strength Index (RSI) of 74.124, i
ndicating it may be overbought.
2. **Ross Stores, Inc. (ROST)** - With a closing price of $140.67 and an RSI of 70.369, also suggesting it might be overbought.

These indicators suggest that these stocks could potentially decline, as they are currently considered overbought.

> Finished chain.
Based on the previous day's performance, the following companies are likely to see a fall in their stock prices:

1. **UnitedHealth Group Incorporated (UNH)** - With a closing price of $582.01 and a Relative Strength Index (RSI) of 74.124, i
ndicating it may be overbought.
2. **Ross Stores, Inc. (ROST)** - With a closing price of $140.67 and an RSI of 70.369, also suggesting it might be overbought.

These indicators suggest that these stocks could potentially decline, as they are currently considered overbought.
```

Figure 20: Response to Prediction of Stock fall



```
# Example query
response = agent.run("based on few technical indicators analyze apple and microsoft stocks, and suggest if i should buy or sell"
print(response)
```

```
> Finished chain.
Here's the analysis of Apple (AAPL) and Microsoft (MSFT) stocks based on the latest data:

### Apple (AAPL)
- **Date**: 2025-04-15
- **Closing Price**: $199.80
- **Relative Strength Index (RSI)**: 39.044
  - An RSI below 40 suggests the stock is potentially oversold, indicating a buying opportunity.
- **Moving Average Convergence Divergence (MACD)**: -9.24
  - A negative MACD indicates a bearish trend, suggesting caution.
- **Bollinger Bands**:
  - Lower Band: $166.988
  - Upper Band: $239.522
  - The closing price is closer to the lower band, which might indicate a potential upward movement.

### Microsoft (MSFT)
- **Date**: 2025-04-15
- **Closing Price**: $384.16
- **Relative Strength Index (RSI)**: 47.838
  - An RSI near 50 suggests the stock is neither overbought nor oversold, indicating a neutral position.
- **Moving Average Convergence Divergence (MACD)**: -4.26
  - A negative MACD indicates a bearish trend, suggesting caution.
- **Bollinger Bands**:
  - Lower Band: $346.968
  - Upper Band: $402.487
  - The closing price is within the bands, suggesting stability but leaning towards the lower band, which might indicate a pote
ntial upward movement.

### Recommendations:
- **Apple (AAPL)**: Consider buying as the indicators suggest the stock might be undervalued and could potentially rebound.
- **Microsoft (MSFT)**: Hold or monitor closely. The indicators suggest a neutral to slightly bearish position, so it might be
prudent to wait for more bullish signals before buying.

Please note that these recommendations are based on technical analysis and it's important to consider other factors such as mar
ket conditions, news, and fundamental analysis before making investment decisions.
```

Figure 21: Response to Buy/Sell of A Stock

10. Conclusion

The stock prediction project successfully demonstrates the integration of data engineering, feature-rich financial analysis, and machine learning to forecast market behavior. By collecting and processing five years of historical stock data from 503 S&P 500 companies, we established a comprehensive dataset that forms the foundation for robust predictive modeling.

Using a cloud-native architecture involving AWS S3, AWS Glue, and Snowflake, we ensured high-performance ETL pipelines and scalable storage. The data was enriched with over 20 advanced technical indicators spanning trend-following, momentum, volatility, and volume-based categories. This enhanced the dataset's predictive power by embedding market dynamics such as momentum shifts, price reversals, volatility bands, and investor sentiment.

Feature engineering played a crucial role, as indicators like MACD, Bollinger Bands, ADX, and RSI revealed complex patterns often overlooked by raw price data. These indicators were calculated using Python and SQL, enabling seamless integration into a unified analytics platform.

Subsequently, various machine learning models were trained—ranging from linear regression to more complex architectures such as LSTM and Random Forest Regressors. The performance of these models was evaluated using metrics like RMSE, MAE, R^2, and directional accuracy, ensuring a balance between prediction precision and interpretability.

The project also highlights the challenges of market prediction, such as overfitting, data noise, and regime shifts. However, with proper cross-validation, feature selection, and normalization, these challenges were mitigated to a reasonable extent. The model results show strong predictive potential, especially in short-term horizon forecasts and trend classification tasks.

Overall, this project showcases a scalable, end-to-end predictive analytics workflow for financial markets. It not only provides valuable insights for investment decision-making but also serves as a prototype for future enhancements, such as incorporating news sentiment, macro-economic indicators, and real-time data streams using APIs and streaming platforms.

11. References

1. Wang, J., & Chen, Z. (2024). Factor-GAN: Enhancing stock price prediction and factor investment with Generative Adversarial Networks. *PLoS One*, 19(6), e0306094.
2. Simplilearn. (2023). *Stock Price Prediction Using Machine Learning*. https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning
3. Shaw, A. (2023). *Python Project: Building a Real-Time Stock Market Price Prediction System*. https://medium.com/@abhishekshaw020/python-project-building-a-real-time-stock-market-price-prediction-system-6ce626907342
4. Chen, J., Jiang, H., Li, T., Yu, Y., & Wu, Y. (2024). *Factor GAN-LSTM hybrid model for financial time series forecasting*. *Journal of Healthcare Engineering*, PMC11198854

Individual Conclusion

1. Introduction

The financial markets are known for their volatility, complexity, and the sheer volume of data generated each day. Traders and investors, both novice and seasoned are frequently challenged with identifying optimal strategies to time investments, assess risk, and choose the right assets. In response to these challenges, our project presents a comprehensive, AI-powered stock market analysis solution that leverages deep learning (LSTM + GAN), large-scale data processing using cloud tools, and an intelligent generative AI agent for investor interaction.

2. Motivation and Objective

The primary motivation behind this project was to bridge the gap between raw financial data and strategic investment decisions using modern AI. While traditional stock price forecasting methods provide some insight, they often fall short in accuracy or real-world usability. Our objective was to:

- Design a predictive model capable of high-accuracy stock price forecasting.
- Provide actionable investment recommendations.
- Create a user-centric AI agent that responds to finance-related queries.
- Develop a suite of interactive dashboards for data-driven decision-making.

3. Data Collection and Infrastructure Setup

To ensure robust model performance and relevant analysis, we began by collecting historical financial data:

- **Source:** Yahoo Finance API (via yfinance library)
- **Scope:** 500 stock tickers over a 5-year period
- **Platform:** AWS SageMaker for scalable model training and data processing
- **Storage:** Raw data stored in Amazon S3 buckets; processed data pushed to Snowflake cloud data warehouse

The overall pipeline involved data extraction, cleaning, transformation, and modeling, all conducted in AWS SageMaker. The processed data was stored in Snowflake and reaccessed during model evaluation and dashboard generation.

4. Predictive Modeling: LSTM + GAN Approach

Our baseline model utilized Long Short-Term Memory (LSTM) networks to predict future stock prices. While this model provided a decent performance with around 60% accuracy, it was insufficient for real-world trading. Hence, we enhanced it using a hybrid model combining:

- **LSTM (Long Short-Term Memory):** For sequential learning of time-series data
- **GAN (Generative Adversarial Network):** For generating synthetic stock trends to fine-tune LSTM predictions

This fine-tuned model achieved a remarkable **99.8% accuracy** with a high **R2 score**, showcasing its ability to capture intricate market patterns and price movements.

5. Generative AI Agent for Investor Queries

To enhance user interaction and utility, we integrated a **Generative AI Agent** using:

- **GPT-4o:** OpenAI's latest and most capable model, providing accurate and context-rich answers
- **LangChain Framework:** For managing context-aware conversations and RAG (Retrieval-Augmented Generation)
- **Pinecone Vector Database:** To store and retrieve semantic embeddings of financial knowledge and FAQs

The agent assists users by answering complex investment-related queries such as:

- When is the best time to invest?
- What are the key factors to consider before investing?
- Which sectors or companies are trending?
- Historical performance insights by ticker or industry

6. Data Visualization with Power BI

A crucial component of our solution is the development of visually rich and interactive dashboards using Power BI:

- **Number of dashboards:** 10 interactive pages
- **Key features:**
    - Sector-wise stock performance analysis
    - Heatmaps of high-performing stocks
    - Forecasting trends and moving averages
    - Risk assessment charts
    - Investment scoring models

These dashboards empower users to explore insights, patterns, and trends without needing to dig into raw data.

7. Tools and Technologies Used

- **Cloud & Data Engineering:** AWS SageMaker, S3, Snowflake
- **Amazon SageMaker**
- **What it is:** A fully managed machine learning service by AWS that allows developers to build, train, and deploy ML models at scale.
  Why we used it: SageMaker was used to train our deep learning models efficiently with GPU support and scalable compute, ensuring faster iteration and experimentation.
- **Amazon S3 (Simple Storage Service)**
- **What it is:** A cloud-based object storage system that provides scalable and secure data storage.
  Why we used it: We used S3 for storing large datasets, trained model artifacts, and logs in a centralized and accessible manner for processing and retrieval.
- **Snowflake**
- **What it is:** A cloud-native data warehouse that enables high-performance analytics and SQL-based access to data.
  Why we used it: Snowflake was used to store and query our processed financial datasets for visualization and dashboarding due to its speed and scalability.
- **Data Collection & Processing:** Python, yfinance, Pandas, NumPy
- **Python**
- **What it is:** A versatile and widely used programming language in data science and machine learning.

Why we used it: Python was our main language for scripting data collection, preprocessing, modeling, and API integration due to its rich ecosystem of libraries.

- **yfinance**
- **What it is:** A Python library that allows easy extraction of financial data from Yahoo Finance.
  Why we used it: yfinance was used to programmatically fetch 5 years of historical stock data across 500 tickers, a key dataset for our model training.
- **Pandas**
- **What it is:** A powerful Python library for data manipulation and analysis, especially for tabular data.
  Why we used it: Pandas helped in cleaning, transforming, and handling time series data efficiently before feeding it to the deep learning models.
- **NumPy**
- **What it is:** A foundational Python library for numerical computation.
  Why we used it: NumPy provided fast array operations and mathematical support for data transformation and matrix operations within the modeling pipeline.
- **Modeling:** LSTM, GAN (TensorFlow / Keras)
- **Pinecone**

  What it is: A vector database used to store and search embeddings (numerical representations of text).
  Why we used it: Pinecone enabled fast semantic search for matching user queries with relevant financial content, enhancing the retrieval-augmented generation (RAG) flow of the assistant.

- **LSTM (Long Short-Term Memory)**

  What it is: A type of Recurrent Neural Network (RNN) well-suited for sequence prediction problems.
  Why we used it: LSTM was ideal for capturing time-based dependencies in stock prices, providing trend forecasting capabilities based on historical patterns.

- **GAN (Generative Adversarial Network)**

  What it is: A neural network architecture consisting of a generator and discriminator that learn to produce realistic data.
  Why we used it: GANs were used to simulate synthetic yet realistic price movements, augmenting our training data and improving model robustness.

- **TensorFlow / Keras**

  What they are: Open-source frameworks for developing and training machine learning models.
  Why we used them: We used TensorFlow/Keras to implement and train our LSTM and GAN architectures efficiently with GPU support.

- **Generative AI:** GPT-4o via LangChain + Pinecone
- **GPT-4o (via OpenAI)**
- **What it is:** The latest high-performance large language model from OpenAI, optimized for reasoning and response generation.
  Why we used it: GPT-4o served as the conversational brain of our platform, generating natural responses and financial insights for users through an AI assistant.
- **LangChain**
- **What it is:** A framework for developing applications powered by large language models, enabling structured memory and tool chaining.

Why we used it: LangChain was used to manage context, user sessions, and prompt engineering in our GPT-4o-powered assistant for relevant, personalized responses.

- **Visualization: Power BI**
  What it is: A business analytics tool from Microsoft for creating interactive visual dashboards. **Why we used it:** Power BI was used to visualize the results of our predictive models and stock trend data, enabling users to interpret performance and market behavior with clarity.

RISKS & CHALLENGES

Despite the promising potential of our project, there are several risks and challenges that must be addressed to ensure the successful execution and expansion of the project in the real world.

A. User Awareness of Sentiment Analysis

Explanation: Sentiment analysis plays a vital role in understanding and interpreting human sentiments in text data, which can influence stock prices. However, users (investors or stakeholders) may not fully understand or may misinterpret the relevance and importance of sentiment analysis. This lack of awareness could limit the effectiveness of our augmented reality (AR) model.

Ramifications: If users do not understand how the AR model functions, they may distrust its predictions, which could result in underutilization of the model and decreased impact on decision-making.

Mitigation: To address this, we will educate users on the power of sentiment analysis in predicting news sentiment and its contribution to the model’s accuracy. This could involve providing clear explanations, training sessions, or informational materials that improve users' understanding and confidence in using the model.

B. Continuous Model and Algorithm Updates

Explanation: Financial markets are highly dynamic and volatile, influenced by changing economic indicators, geopolitical events, and market sentiment. To ensure that the AR model remains effective, it is crucial to update the model and its underlying algorithms regularly to align with shifting market conditions.

Ramifications: If the model and algorithms are not updated frequently, their predictive value may diminish, as predictions may no longer reflect the current state of the market or other influencing factors.

Mitigation: We will implement a formal mechanism to monitor changes in the market and update the AR model and algorithms accordingly. This includes staying current on developments in financial time series and incorporating new insights into the model. Automated tools and continuous optimization techniques will also be used to maintain the model’s relevance.

C. Acquiring Real-Time Data

Explanation: Real-time data is essential for training and updating the AR model, as well as for generating timely predictions. However, accessing accurate and up-to-date financial data can be challenging due to latency, data quality issues, and limitations on data access.

Ramifications: Delayed or incomplete real-time data could prevent the AR model from capturing the latest market dynamics, making predictions less accurate and timely, which undermines the model's usefulness in decision-making.

Mitigation: To mitigate this risk, we will use multiple data sources and providers to ensure comprehensive coverage of financial data from various angles. We will establish robust data pipelines to manage data latency and related issues and implement data streaming and APIs for real-time access to data.

D. Overreliance on the Model

Explanation: While the AR model may be effective at predicting future stock price movements, relying on it exclusively for decision-making is risky. The model could overlook important qualitative and quantitative factors that significantly influence stock prices.

Ramifications: Overreliance on the model without considering other market factors could result in poor investment decisions.

Mitigation: We will encourage users to use the AR model as one tool in a broader decision-making process, rather than relying on it alone. It is essential that investors incorporate other forms of analysis, such as domain expertise and traditional fundamental analysis, to contextualize the model's predictions within the current market conditions. Additionally, we will provide guidance on how to integrate the model's outputs with other market insights for more informed decision-making.

My Conclusion

Our project offers an end-to-end stock market strategy analysis solution combining the power of deep learning, cloud computing, and generative AI. The high-accuracy model backed by intelligent user interaction and rich dashboards presents a compelling tool for investors. As we continue refining the system, the goal is to make advanced stock analysis accessible, reliable, and actionable for everyone from retail investors to institutional analysts.

Future Scope:

- Integration of real-time trading signals
- Expansion to global stock exchanges
- Personalized portfolio management via AI agents