Stock Price Prediction Analytics using Snowflake & Apache Airflow

Jiyoon Lee, Jeff Chong
Data Analytics, San Jose State University
One Washington Square, San Jose, CA 95192, USA
jiyoon.lee@sjsu.edu
junjie.chong@sjsu.edu

***Abstract*—This lab report details the development of a stock price prediction system. The approach employs a public stock price API for data collection, with Snowflake utilized as a cloud-based data warehouse, and Apache Airflow for managing the ETL (Extract, Transform, Load) pipeline. Snowflake's built-in machine learning capabilities are employed for time-series forecasting to predict short-term stock prices. The integration of these tools demonstrates an efficient data pipeline for real-time and historical stock price analytics, showcasing the potential for enhancing data-driven decision-making in financial markets.**

***Index Terms*—stock price prediction, ETL, data pipeline, Snowflake, Apache Airflow**

I. Problem Statement

A. System Overview

The stock market presents a dynamic environment that demands effective data management and analysis for informed decision-making. The goal of the proposed system to provide timely insights into stock market trends. To address the challenges faced in handling and analyzing large volumes of stock data, an end-to-end solution is proposed.

This system emphasizes the implementation of an automated and streamlined data pipeline and utilizes a cloud-based data warehouse to facilitate regular data collection, processing, and storage of the latest 90 days' worth of stock data. Given the frequent fluctuations in stock prices, an automated solution is crucial to ensure up-to-date information for decision-making. Additionally, a forecasting model, built using Snowflake, is incorporated to predict short-term stock prices, serving as a practical demonstration of the system's capabilities. This stock price forecasting component is also automated to always get stock price predictions for the next seven days.

As part of the objectives of this system, the prices for two stock symbols will be obtained, and their prices forecasted.

B. Importance of Database and Data Pipelines

A database (data warehouse) and data pipelines are integral components of the system. The first data pipeline handles the ETL (Extract, Transform, Load) process, converting raw stock data from a public API into a structured format and ensuring daily updates with the latest 90 days of stock prices. This constant refresh of the latest 90 days' worth of stock prices supports both historical analysis and real-time decision-making. The Snowflake data warehouse stores the transformed data in a structured and accessible format, enabling efficient querying and data analysis. There is also another data pipeline to handle the ELT (Extract, Load, Transform) process which enables the forecasting models by transforming the data again so that it obeys the format required by the model.

II. Solution Requirements

A. Functional Requirements

1) Data Collection: The system will gather stock price data from the Alpha Vantage API, a public stock market data API.

2) Data Transformation: The ETL pipeline will obtain stock data from the Alpha Vantage API, transform it from its raw JSON format into a structured format and store the transformed data into a database in Snowflake. The end result of the ETL pipeline is data ready to be used for further analysis or ELT (Extract, Load, Transform) purposes.

3) Data Storage: Snowflake will serve as the cloud-based data warehouse for storing the transformed data, providing both scalable storage and fast access for analysis.

4) Basic Forecasting: An ML (machine learning) forecasting model will be employed, demonstrating how the system can generate short-term stock price predictions using Snowflake's machine learning capabilities.

5) Automated Scheduling: Apache Airflow, a workflow management tool, will be used to automate and orchestrate the ETL pipeline and the forecasting model. Apache Airflow schedules the ETL pipeline to run daily, ensuring that the system is consistently updated with the latest stock prices over a 90-day rolling window. The forecasting model is automated to provide update-to-date stock price predictions for the next seven days.

B. System Capabilities

By achieving all the above system requirements, this proposed end-to-end solution will automate the collection, processing, and analysis of stock price data, providing users with access to updated stock information and stock price forecasting.

C. System Limitations

1) Performance and Scalability: The current implementation of the ETL pipeline inserts each record individually into the Snowflake table, which may lead to performance bottlenecks as the volume of data or the number of stock symbols increases. If the intention is to scale the volume of data in the future, implementing bulk inserts or using a staging table could significantly improve the performance and scalability of the data loading process.

2) Scope of Forecasting: The stock price prediction model is basic and serves only as a demonstration of Snowflake's ML features; it does not aim to provide highly accurate or long-term forecasts.

D. User Interaction

1) Data Access: Users will interact with the system by running SQL queries to retrieve and analyze the stored stock data and view forecast results. Snowflake's querying interface allows for easy access to both raw and transformed data.

2) Automation Management: The automated data pipeline ensures minimal manual intervention, with scheduling and execution managed by Apache Airflow.

III. Functional Analysis

A. Overall System Diagram

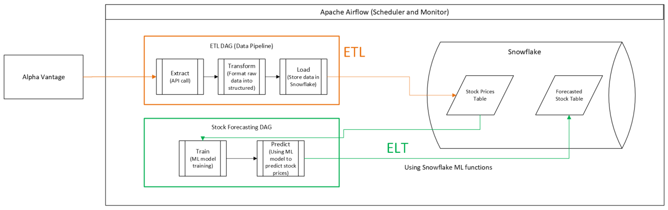

Apache Airflow (Scheduler and Monitor)
Alpha Vantage
ETL DAG (Data Pipeline)
Extract (API call)
Transform (Format raw data into structured)
Load (Store data in Snowflake)
ETL
Snowflake
Stock Prices Table
Forecasted Stock Table
Stock Forecasting DAG
Train (ML model training)
Predict (Using ML model to predict stock prices)
ELT
Using Snowflake ML functions

Fig. 1. Overall System Diagram.

The full-sized image of the system diagram can be viewed here.

B. Data Source: Alpha Vantage API

The Alpha Vantage API provides financial data for stocks, forex, and cryptocurrencies. The Daily Time Series endpoint (TIME_SERIES_DAILY) returns historical daily data for stock, including open, high, low, close, and volume for each day the market is open. To access this data, an HTTP GET request with the stock symbol and API key is required.

C. Data Warehouse: Snowflake

Within the Snowflake database, the raw_data schema in the Snowflake database contains data sourced from the Alpha Vantage API. The adhoc schema enables dynamic analysis and forecasting using statistical models and machine learning on this raw data. Finalized results are then stored in the analytics schema, which serves as a centralized repository for insights and dashboards. This structured approach effectively transforms raw data into actionable insights, supporting informed decision-making across the organization.

D. Apache Airflow Pipelines

It allows users to define data pipelines as Directed Acyclic Graphs (DAGs) in Python, making it easy to visualize, manage, and execute tasks. Here are the components of Apache Airflow that are used in the system:

1) Variables: Airflow Variables are versatile for managing small key-value pairs within workflows. In this implementation, credentials such as API keys are stored using Airflow Variables.

2) Snowflake Connection: In Apache Airflow, connections allow tasks to interact with external services like Snowflake. The apache-airflow-providers-snowflake package offers operators and hooks for executing queries and managing workflows, enabling a connection to the Snowflake database to be established. A Snowflake Connection object was created in the Apache Airflow interface using Snowflake credentials such as the username, password and account ID.

3) ETL Pipeline: An ETL (Extract, Transform, Load) pipeline is a process used in data integration to collect data from data sources, transform it into a suitable format, and load it into Snowflake data warehouse.

In this ETL pipeline, each phase has specific tasks that handle different aspects of data movement and preparation.

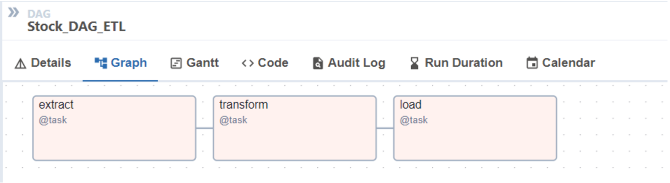


Fig. 2. Graph for ETL DAG.

1) Task 1: Extract data from API
2) Task 2: Restructure the 90 days' worth of data
3) Task 3: Load transformed data into Snowflake

To retrieve the stock data, a HTTP (Hypertext Transfer Protocol) GET request is made to the endpoint URL key along with the API key and the stock symbol; the json() method will then return the data as a Python dictionary.

Once the data is obtained, each record is ensured to have the stock symbol and date appended, which will later serve as a composite primary key, and then loaded it into the Snowflake database.

4) ELT Pipeline: Once the data is loaded into the warehouse by the ETL pipeline, transformations can be applied, such as cleaning, aggregating, and enriching the data to make it suitable for analysis. The primary goal of this ELT pipeline is to enable forecasting models that predict future stock prices.

1) Task 1: Train a model
2) Task 2: Forecast the stock prices for the next 7 days

By using the Time-Series Forecasting, a predictive model is trained based on historical data loaded through the ETL pipeline.

Once the model is trained, forecasts for the next seven days can be generated by invoking the !FORECAST function. The results provide valuable insights for decision-making, enabling us to anticipate trends and adjust business strategies.

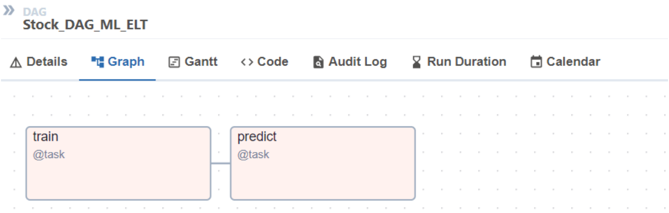


Fig. 3. Graph for ML ELT DAG.

E. Collective Functionality of System Components

The components of the system work collectively to address the problem of stock price prediction by integrating data collection, transformation, and analysis into a cohesive workflow. The Alpha Vantage API serves as the data source, providing historical stock data that is extracted and structured into a suitable format through the ETL pipeline, which is managed by Apache Airflow. This transformed data is loaded into Snowflake, organizing the data across different schemas for raw storage, dynamic analysis, and final results. Apache Airflow's orchestration capabilities ensure that data pipelines are automated and efficient, while Snowflake's machine learning functionality is leveraged to apply time-series forecasting models. Together, these components form a streamlined system that automates the entire process, from data collection to forecasting, enabling timely stock price predictions.

F. Table Structure

TABLE I
ATTRIBUTES AND DATA TYPES FOR STOCK PRICE TABLE

Attribute	Data Type
`date`	DATE
`open`	FLOAT
`high`	FLOAT
`low`	FLOAT
`close`	FLOAT
`volume`	INTEGER
`symbol`	VARCHAR

1) Stock Price Table: Shown in Table I are the attributes and their respective data types of the stock prices table resulting from ETL pipeline. There is a constraint on the table - the primary key. It is a composite key made from the attributes 'date' and 'symbol'. The reasoning is that as the table consists of records for multiple stock symbols, this composite primary key allows us to have records for the same day, but the stock symbols have to be unique.

TABLE II
ATTRIBUTES AND DATA TYPES FOR FORECAST TABLE

Attribute	Data Type
`symbol`	VARCHAR
`date`	TIMESTAMP
`actual`	FLOAT
`forecast`	FLOAT
`lower_bound`	FLOAT
`upper_bound`	FLOAT

2) Forecast Table: The table containing the predicted stock prices, as described in Table II, is the end result of the ML ELT pipeline. The first seven records hold the predicted stock prices for the next seven days and the remaining records hold the stock prices for the last 90 days. The ML model forecasts the attributes 'forecast', 'lower_bound' and 'upper_bound' and as such, these remaining records have null values for these attributes. The main concern of this proposed system is the value in the attribute 'actual' for the next seven days of stock price prediction. Addtionally, as these seven records are for future dates, there are null values for the attribute 'actual'. The attribute 'actual' represents the ground truth, i.e. the actual stock prices for past data. As there are two stock symbols, this means that there will be a total of 14 records of stock price predictions.

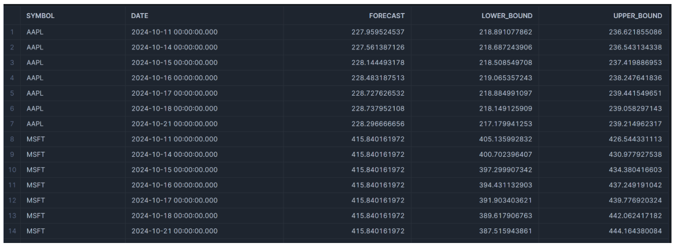
IV. SYSTEM USAGE & ACCESS

This section describes how a user will engage with the system to get the end results of the system.

A. Accessing Stock Price Predictions

To get the stock price predictions, the user will execute a SQL query in a Snowflake worksheet, provided they have access to the database. The query can be found here. The query will fetch only the records of stock price prediction, resulting in 14 records of predictions, comprised of seven records for each stock symbol.

Shown below in Figure 4 is the result of the query execution. For visual clarity, here is access to the full-sized image.



	SYMBOL	DATE	FORECAST	LOWER_BOUND	UPPER_BOUND
1	AAPL	2024-10-11 00:00:00.000	227.959524537	218.891077862	236.621855086
2	AAPL	2024-10-14 00:00:00.000	227.561387126	218.687243906	236.543134338
3	AAPL	2024-10-15 00:00:00.000	228.144493178	218.508549708	237.419886953
4	AAPL	2024-10-16 00:00:00.000	228.483187513	219.065357243	238.247641836
5	AAPL	2024-10-17 00:00:00.000	228.727626532	218.884991097	239.441549651
6	AAPL	2024-10-18 00:00:00.000	228.737952108	218.149125909	239.058297143
7	AAPL	2024-10-21 00:00:00.000	228.296666656	217.179941253	239.214962317
8	MSFT	2024-10-11 00:00:00.000	415.840161972	405.135992832	426.544331113
9	MSFT	2024-10-14 00:00:00.000	415.840161972	400.702396407	430.977927538
10	MSFT	2024-10-15 00:00:00.000	415.840161972	397.299907342	434.380416603
11	MSFT	2024-10-16 00:00:00.000	415.840161972	394.431132903	437.249191042
12	MSFT	2024-10-17 00:00:00.000	415.840161972	391.903403621	439.776920324
13	MSFT	2024-10-18 00:00:00.000	415.840161972	389.617906763	442.062417182
14	MSFT	2024-10-21 00:00:00.000	415.840161972	387.515943861	444.164380084

Fig. 4. Result of executing the SQL query to obtain stock price predictions.

B. Source Code

All code is published in a GitHub repository, accessible here. Also included in the repository are additional images that showcase the use of Snowflake Variables and Connections. All data pipeline code is written in Python, with segments of SQL code used to interact with the Snowflake database.