Building a Finance Data Analytics – Stock Price Prediction Model

By Anshika Goel, Sai Chaitanya Munagala, San Jose State University,
Applied Data Science Department.
03/06/2025

Abstract

Stock price prediction plays a major role in business sectors to anticipate the market trends and prepare for better tomorrow. This project paper aims to present an end-to-end stock price prediction for the next 7 days using Snowflake and Apache Airflow. The pipeline extracts the stock data of past 180 days from yfinance API and stores them in a snowflake database, and leverages Machine learning capabilities to predict future stock pricing. Two DAGs are implemented for this project, one for data extraction and loading which we popularly called as ETL and other for forecasting and merging results using UNION ALL command. The final dataset integrates actual and predicted prices for complete comprehensive analysis. Although the project demonstrates the integration of cloud-based data warehousing, machine learning, and workflow orchestration to forecast data, the results represent likelihoods rather than guaranteed actual pricing.

---

1. Introduction

A. Background

Predicting stock price is a critical process in finance and investing sector. Lot of the times investors, traders and also data analysts require reliable and proper prediction tools to forecast stocks prices. General traditional forecasting models typically required of some manual intervention at every stage like collecting, storing and predicting. Hence, they are inefficient, error-prone and not scalable. Now, with the advanced cloud computing technology and availability of fully automated data pipelines, financial forecasting is a streamlined process in a much efficient way.

B. Problem Statement

The highly dynamic stock market requires a continuous data updates and forecasting. Manually retrieving stock prices and analysing one at a time is a real time-consuming and impractical for large scale financial systems especially stock industry where the data gets changed at a real rapid phase. The key challenges include:

1. *In-efficient data processing* – Stock Information needs to be extracted, transformed and stored at given real time.
2. *Scalability Issues* – Traditional go to approaches do not support real time updates.
3. *Machine Learning Integration* – Analysis and predicting part requires external ML tools.

To address these challenges, we propose a fully automated stock price forecasting system using Apache Airflow, Snowflake and yfinance API.

C. Objectives

The objective of this project are as follows:

1. To develop a fully automated stock data extraction and storage using *Apache Airflow* and *Snowflake*
2. To Apply machine learning models to forecast stock prices.
3. To *Optimize performance issues*, scalability and reliability using Airflow scheduling, *SQL transactions* and *error handling mechanisms.*
4. To merge actual and predicted prices into a final table for analysis purpose.

---

2. System Architecture

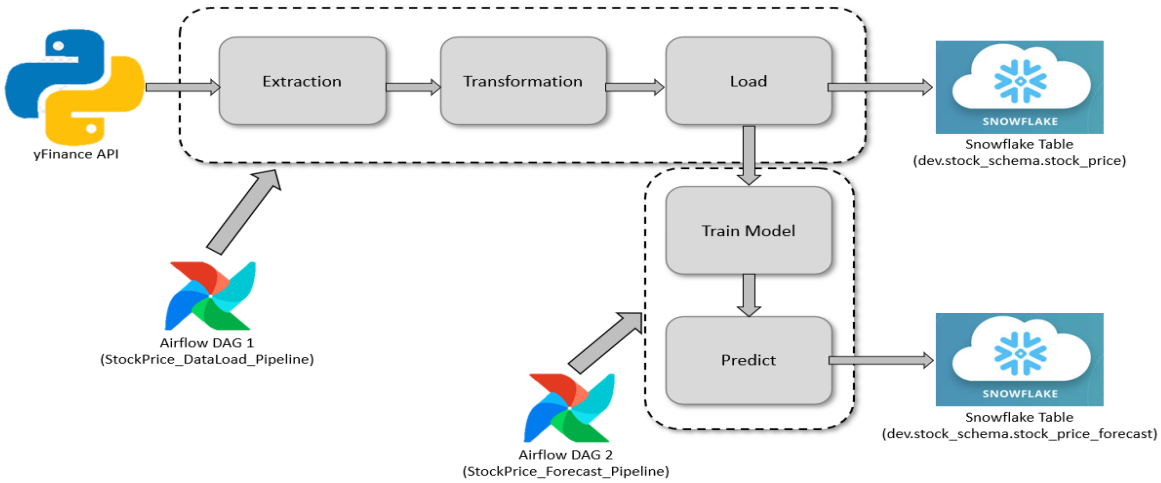
A. Technology Stack

The project is developed using the listed technologies:

1. *yfinance API* – for fetching real-time and historical stock price data.
2. *Python* – for scripting and integrating API.
3. *Apache Airflow* – for Managing automated workflows to extract and forecast data.
4. *Snowflake* – for primary data warehouse.
5. *SQL* – for data storage, querying and machine learning forecasting within Snowflake.

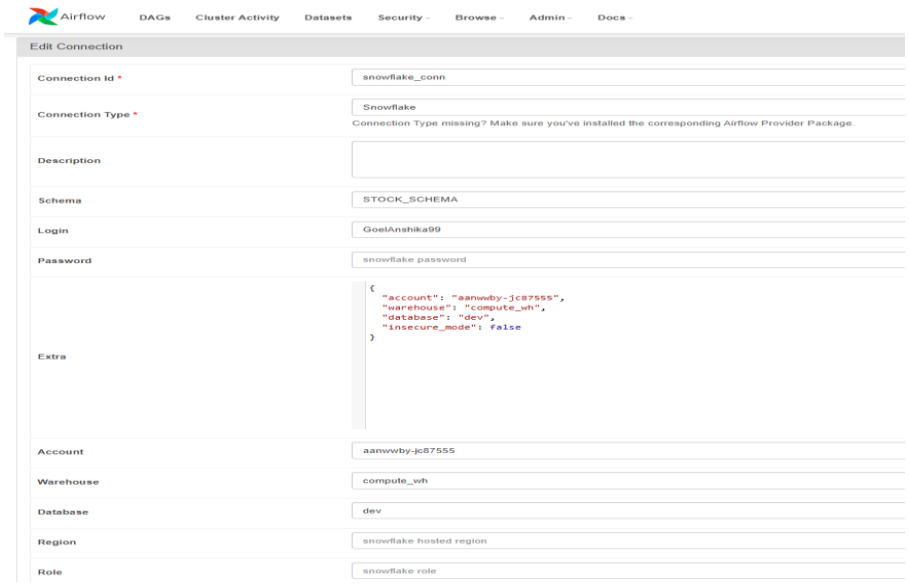
B. System Design

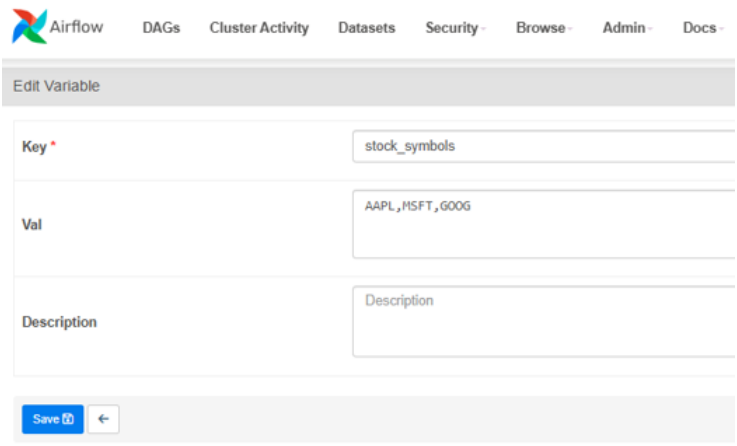
Overall Workflow diagram is constructed as follows:



C. Airflow Connections and Variables

To ensure flexibility and maintainability, Airflow connections and Airflow variables are used to manage configurations dynamically for just one real time. The snowflake connections are securely stored in Airflow under the tab snowflake_conn identifier. This is a great practise which allows securing snowflake credentials and smooth connection between Airflow and Snowflake.





In Airflow variables, stock ticker symbols are stored as an Airflow Variable in the form of Key and Values. This option eliminates the developer to tweak the code every time for stock change. The UI built option In Apache Airflow is perfectly set for the developer user to change it anytime without altering DAG script.

Thus, The Core System Designs after proper connections and *DAG scripting* ideally consists of two primary workflows:

1. *ETL Pipeline* - Extracts stock data and process it before loading into snowflake.
2. *Forecasting Pipeline* – Uses Snowflake ML model to forecast next seven-day pricing.

Figure: ETL DAG Pipeline with 3 tasks [Extract, Transform and Load]

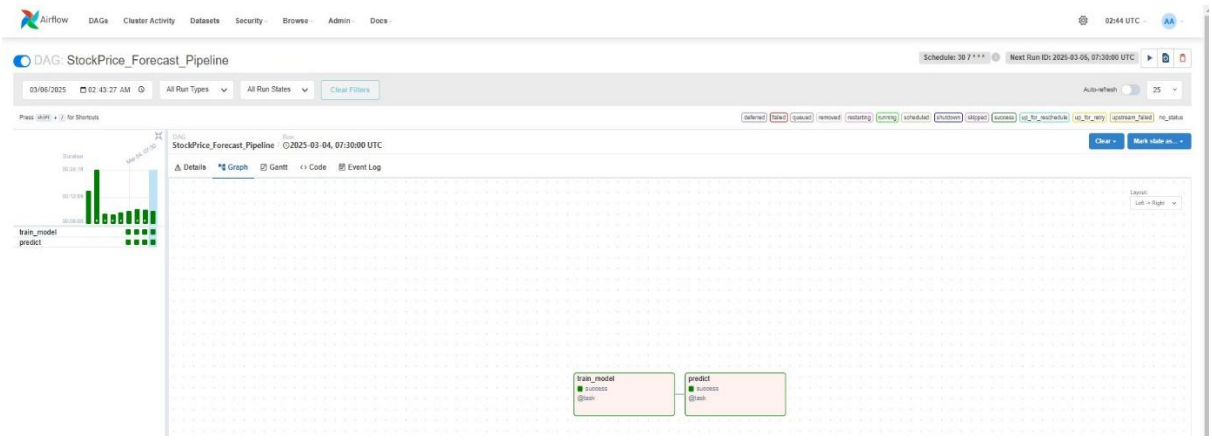


Figure: Forecasting Pipeline with 2 tasks [Train and Predict]

D. GitHub Repository for Airflow DAG

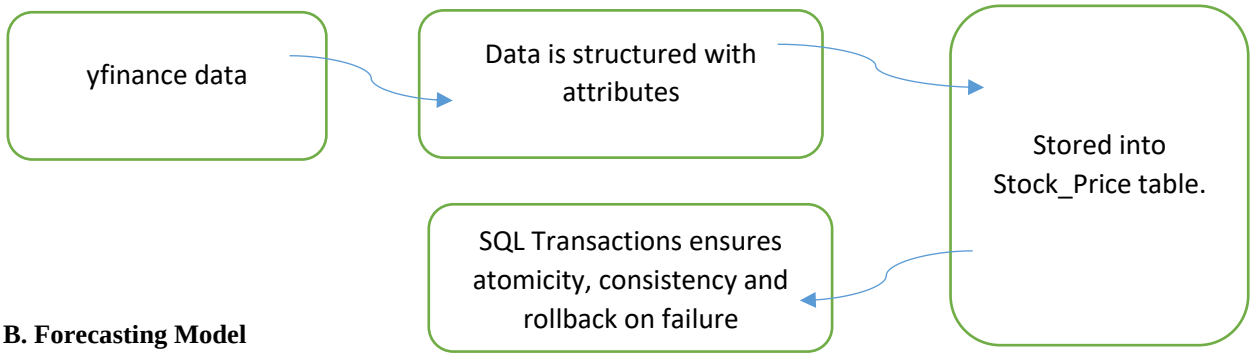
ETL DAG – https://github.com/GoelAnshika99/Lab-1_Group-16/blob/main/ETL%20DAG.py

Forecast DAG - https://github.com/GoelAnshika99/Lab-1_Group-16/blob/main/Forecast%20DAG.py

3. Methodology

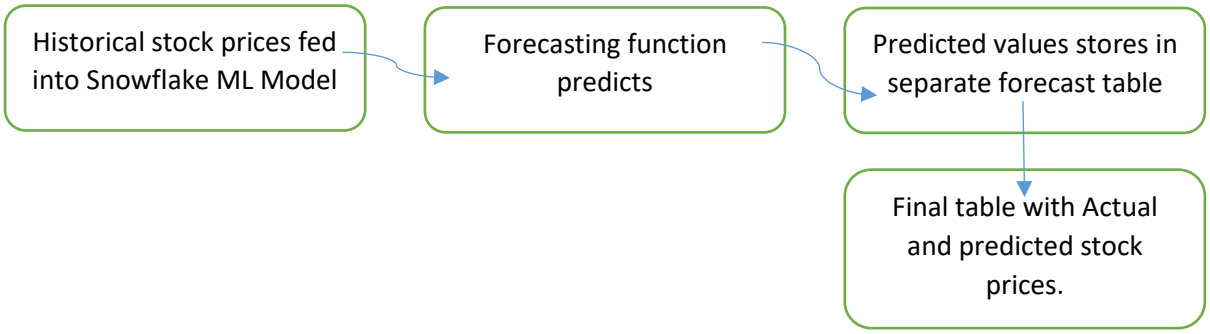
A. Data Extraction and Loading

The *ETL Pipeline* extracts stock data using *yfinance API* and it loads into structured table in Snowflake. The process typically follows,



B. Forecasting Model

The forecasting pipeline applies built-in *Machine Learning Model* to predict stock prices seamlessly as mentioned for the next seven days. The process typically follows,



---

4. Database Schema and Data Storage

A. Stock Price Table

This table contains *historical stock prices* fetched from yfinance.

Field Name	Data Type	Attributes	Constraints
Symbol	Varchar	Unique identifier	Not Null, Primary Key
Date	Date	Represents trade date	Not Null, Primary Key
Open	Float	Numeric, Financial Value	Nullable
High	Float	Numeric, Financial Value	Nullable
Low	Float	Numeric, Financial Value	Nullable
Close	Float	Numeric, Financial Value	Nullable
Volume	Float	Numeric, Financial Value	Nullable

B. Forecasted Stock Price Table

This table stores *predicted stock prices* for the next seven days.

Field Name	Data Type	Attributes	Constraints
Symbol	Varchar	Unique identifier	Not Null, Primary Key
Stock_ts	Date	Timestamp of record	Not Null, Primary Key

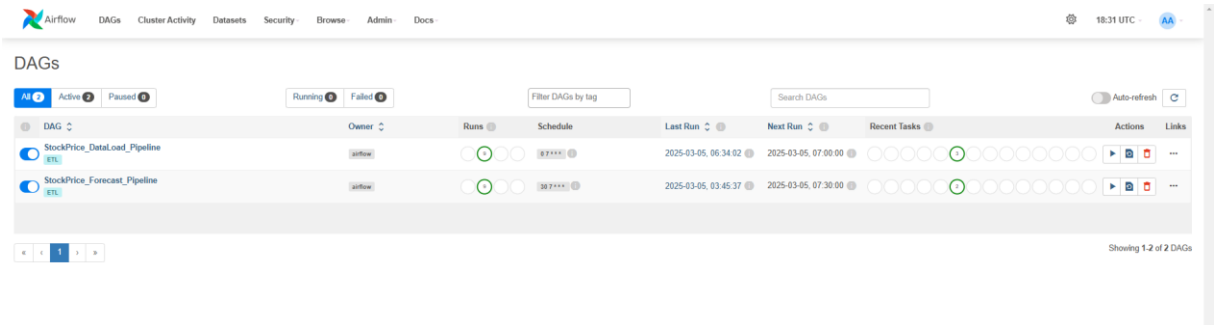
Open	Float	Numeric, Financial Value	Nullable
High	Float	Numeric, Financial Value	Nullable
Low	Float	Numeric, Financial Value	Nullable
Volume	Float	Numeric, Financial Value	Nullable
Actual_close	Float	Numeric, Financial Value	Nullable
Forecast_close	Float	Numeric, Financial Value	Nullable
Forecast_Lower_Bound	Float	Numeric, Financial Value	Nullable
Forecast_Upper_Bound	Float	Numeric, Financial Value	Nullable

---

5. Results and Discussion

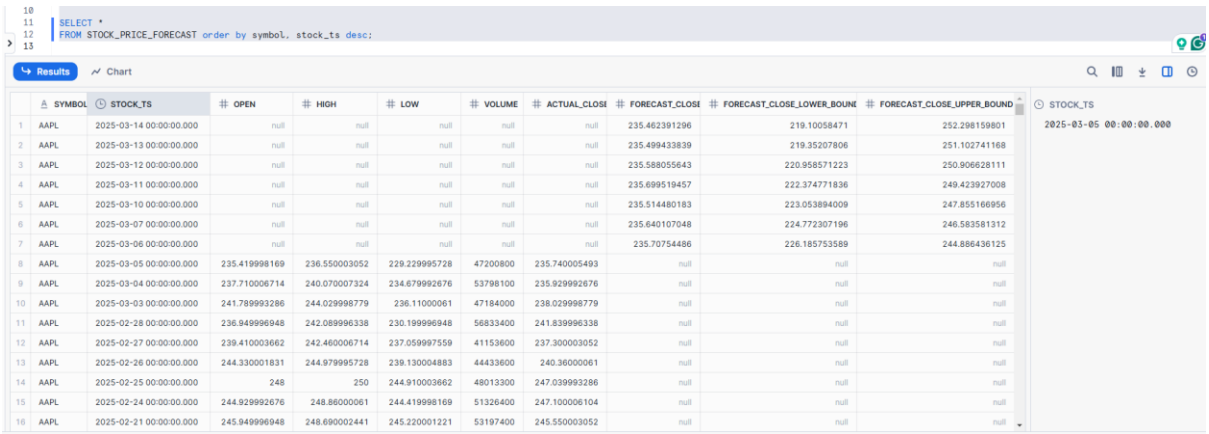
A. Airflow Execution Results

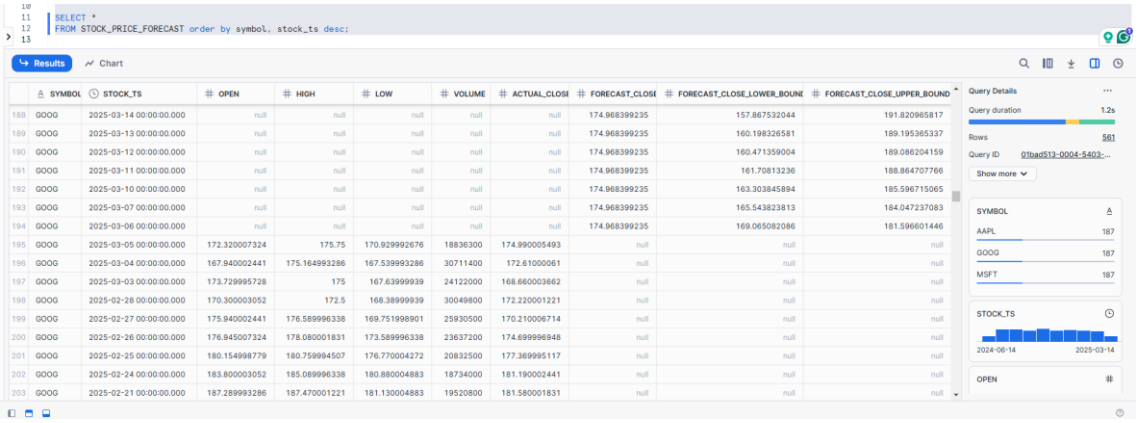
Both the *Airflow DAGs* are executed successfully, fetching the stock price data, loading them into the tables in snowflake and generating a seven-day forecast.



B. Final Consolidated Table

The final table is consolidated with actual and predicted stock prices with the help of UNION ALL command and stores them into table, allowing for an easy visualization.





Note: The Final table above is *generated at a timestamp 05th March 2025*, there by showing next 7 days results.

---

6. Conclusion

Thus, finally at the end of the project, A fully automated pipeline is developed and built to automate stock price prediction with the help of *Apache Airflow*, *Snowflake* and *yfinance API*. This implementation eliminates manual intervention at every stage of its life cycle and ensures that stock prices are fetched, stored, processed, and forecasted seamlessly on a scheduled basis.

With Airflow orchestrating the workflows, the ETL pipeline reliably pulls stock price data and stores it in Snowflake**,** while the forecasting pipeline applies machine learning models to generate future stock price predictions. The use of Airflow connections and variables makes the pipeline flexible and easily configurable**,** allowing updates without modifying the DAG scripting. Therefore, the final output, which combines actual and predicted stock prices**,** provides valuable insights for financial analysis**.**

---

7. References

[1] Snowflake Documentation: https://docs.snowflake.com/
[2] Apache Airflow Documentation: https://airflow.apache.org/docs/
[3] yFinance API Documentation: https://pypi.org/project/yfinance/
[4] IEEE Paper Formatting Guidelines: https://www.ieee.org/conferences/publishing/templates.html