Stock Prediction Model Report

Aditya Tekale, Tanya Yadav
Department of Applied Data Science
San Jose State University
San Jose CA 95112
aditya.tekale@sjsu.edu,
tanya.yadav@sjsu.edu

***Abstract*—This project presents the development of a stock prediction model system using Snowflake and Airflow to predict stock prices for two companies over the next seven days. The system uses the Alpha Vantage API to retrieve stock market data, which is stored in Snowflake database. An airflow DAG is implemented to run the data extraction every day. Machine Learning forecasting tasks are performed in Snowflake to predict the stock prices. The report outlines the system architecture, database schema, and key Python and SQL implementations, highlighting the efficiency of using cloud-based data warehousing and automated pipelines in financial forecasting.**

Keywords— Stock Price Prediction, Snowflake, Airflow, Data Pipelines, Machine Learning, Alpha Vantage API, Data Warehousing, Cloud Computing, SQL Forecasting

I. Introduction

In this project, a stock prediction model system is developed to predict the stock prices based on historical data. The system is built using Snowflake, a cloud data platform, and Airflow, an open-source platform to automate scheduling and monitoring of the workflows. Using Alpha Vantage API to collect stock market data, we created a data pipeline to store and process this data for forecasting future stock prices. The focus of this project is to build a scalable solution that provides reliable predictions using Machine Learning techniques within the Snowflake environment.

II. Problem Statement

The objective of this project is to build a stock price prediction system for financial analysis using database and data pipelines. The prediction system will be forecasting stock prices using the 90 days historical data for each stock retrieved from the Alpha Vantage API. The need for this system arises from the vast amount of stock data available, which requires an automated process for collection, storage, and analysis.

The system relies critically on databases and data pipelines, to ensure it performs effectively. A database, implemented using Snowflake, will efficiently store the historical stock price data and enable quick access for analysis. Data pipelines, implemented using Airflow, will automate the data retrieval process and ensure that the data is consistently updated daily, which is crucial for maintaining accurate forecasting models.

Using Airflow for automation ensures that data extraction from Alpha Vantage occurs daily, while Snowflake serves as the data warehouse for organizing and analyzing the data.

III. Solution Requirements

To develop an effective solution, the system must meet the following requirements:

- Data Retrieval: The system should connect to the Alpha Vantage API to fetch 90 days historical stock prices for selected companies and transform the data to match our table structure.
- Data Storage: The retrieved data must be stored in a Snowflake database in a well-structured table format.
- Automated Pipelines: The data retrieval process must be automated using Airflow DAGs, ensuring daily updates of stock price data.
- Forecasting Capability: The system must implement machine learning tasks in Snowflake to predict stock prices for the next seven days based on historical data.

The system's main limitation is the dependency on the accuracy and availability of the data provided by the Alpha Vantage API. Furthermore, the predictive accuracy of machine learning models may vary based on market conditions and external factors not accounted for in the historical data.

This system is intended for data analysts, financial planners and investors who want to make wise decisions of investment. Similar to other scheduler tools, the Airflow interface allows users to track data extraction and processing workflows for updates. Users can quickly and directly obtain historical and predicted stock data within Snowflake, which can be used in creating additional reports or Dashboards. Moreover, with daily updated forecasts, the system offers the users short-term stock predictions to help them make immediate decisions in a volatile market.

IV. Functional Analysis

The proposed application system consists of several functional components that collectively solve the problem of stock prediction.

A. Data Source

1) Alpha Vantage API: API key retrieved from Alpha vantage is added to Airflow variables for security so that the variable can be used in the dag instead of directly placing API in code. The API key serves as a primary data source for historical stock prices. The API provides comprehensive market data, including stock prices (open, claose, high, low, and volume).

2) Stock Symbols: In the extract, we select two stock symbols to do the prediction for next 7 days. For this project, GOOGL and TTWO stock has been used.

B. *Database*

To connect Snowflake in Airflow, snowflake is established in Airflow environment using Airflow connections, which is called in each DAG. The Snowflake database is structured into three schemas to facilitate efficient storage, analysis, and forecasting of stock data extracted from the Alpha Vantage API.

1) Raw_data: The raw_data schema contains a table named stock_price, which stores the raw stock data retrieved from the Alpha Vantage API. The table contains columns- Symbol (varchar), Date (timestamp_ntz) which is the primary key, Open (float), Close (float), Low (float), High (float) and Volume (float).

2) Adhoc: The adhoc schema contains a table named stock_prices_forecast which consists of forecast (float), lower_bound (float), series (variant), ts (timestamp_ntz), upper_bound (float) generated by forecasting the stock prices for the next seven days using machine learning algorithms and a view called stock_prices_view, created from the raw_data.stock_price table to extarct relevant columns- Date (timestamp_ntz), Close (float), and Symbol (varchar) for training machine learning models.

3) Analytics: The analytics schema contains one table named market_data which combines both historical stock data and predicted stock prices using a union on both. It includes columns for symbol (varchar), date (timestamp_ntz), actual (float), forecast (float), and the upper (float) and lower (float) bound. This table provides a comprehensive view for tracking and analyzing both past trends and future predictions.

C. *Data Pipeline*

The Airflow DAGs automates the process of retrieving data from the API and storing it in Snowflake and then performing the machine learning forecasting tasks. This pipeline consists of the following steps:

1) Extract, Transform and Load : The ETL process consists of first fetching the last 90 days of stock price data by connecting to the Alpha Vantage API for the selected stock symbols, then transforming the data into the required format to match the table in snowflake and lastly loading it into a Snowflake table.

2) Machine Learning Forecasting: SQL queries will be used to set up machine learning forecasting tasks in Snowflake, enabling predictions of stock prices for the next seven days based in the historical data stored in the database.

V. Overall System Diagram

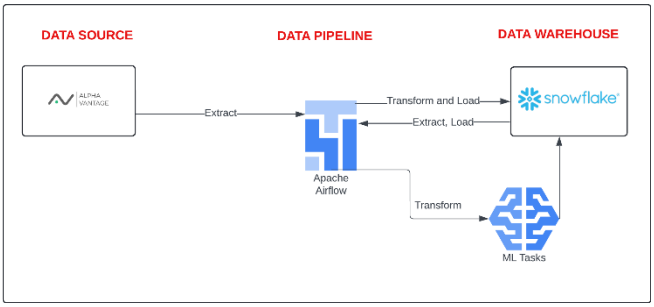

Fig. 1. System Diagram

VI. Table Structure

A. *dev.raw_data.stock_price*

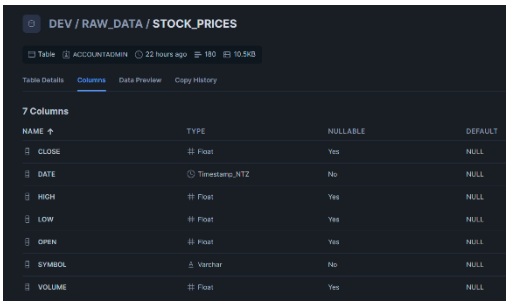

Fig. 2. stock_prices table

B. *dev.adhoc.stock_prices_forecast*

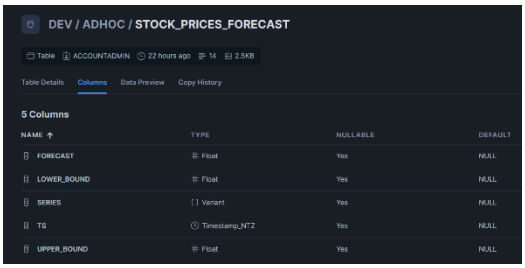

Fig. 3. stock_prices_forecast table

C. *dev.adhoc.stock_prices_view*

Fig. 4. stock_prices_view table

D. *dev.analytics.stock_prices_with_forecast*

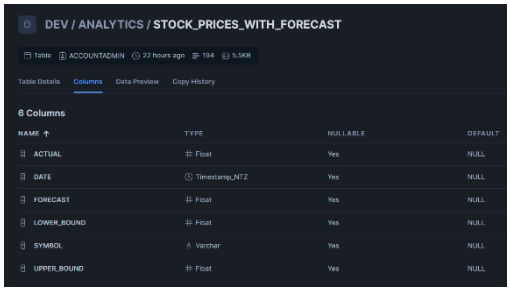

Fig. 5. stock_prices_with_forecast table

VII. Snowflake

A. *SQL query execution results*

1) Extracted Data:

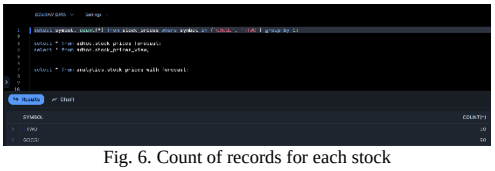

Fig. 6. Count of records for each stock

2) Stock forecast:

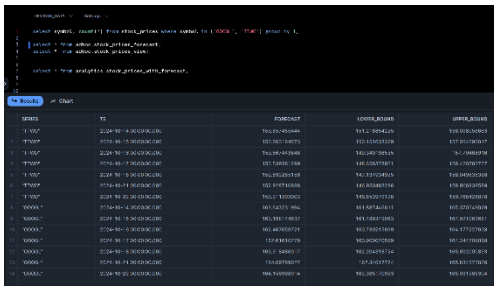

Fig. 7. Predicted stock prices for each stock

3) Count of records after union in analytics schema:

Fig. 8. Count of records after union

VIII. AIRFLOW

A. Codes

1) Single DAG: In this approach, single dag with multiple tasks were created and implemented that performs the following:

a) Extract (extract_stock_data()): Extracts 90 days stock data from api using api key which was provided in airflow variables.

b) Transform (transform_stock_data()): Transforms data to match the table structure in snowflake

c) Load (load_to_snowflake()): Loads the data into snowflake using snowflake connection which was configured in airflow connections.

d) Train (train_forecast_model()): Uses ML feature in snowflake and uses the data to train the prediction model and returns predcited stock prices for 7 days

e) Predict(predict_stock_prices()):Combines historical data and predicted data for further analysis

GitHub Link: https://github.com/aditya-tekale-99/Stock-Prediction/blob/main/stock_prediction_dag.py

2) Multiple DAGs: In this approach, two dags with relevant tasks were created and implemented. The first DAG (etl.py) performs the following tasks:

a) Extract (extract())

b) Transform (transform())

c) Load (load_to_snowflake())

The second DAG (trainpredict.py) performs the following tasks:

a) Train (train())

b) Predict (predict())

Github Link: https://github.com/tyadav2/Stock-ML-Forecasting

B. Web UI Screenshot

1) Single DAGs:

Fig. 9. Web UI showing the dag overview

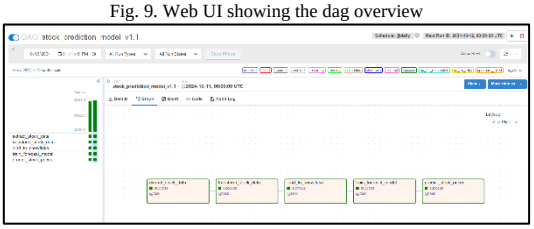

Fig. 10. Graph showing flow of task executions

Fig. 11. Log showing success predict task (final task)

2) Multiple DAGs:

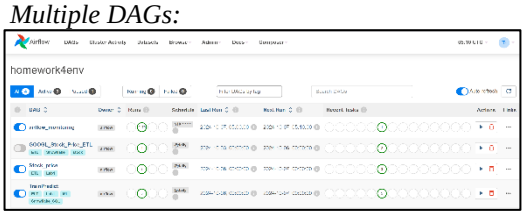

Fig. 12. Web UI showing two dags overview

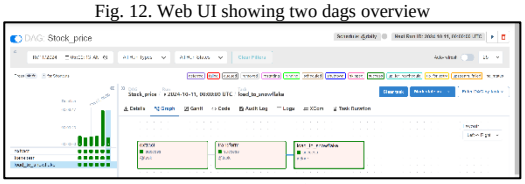

Fig. 13. Graph of ETL

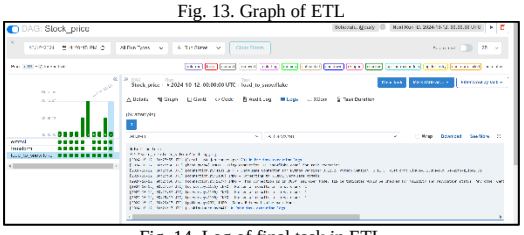

Fig. 14. Log of final task in ETL

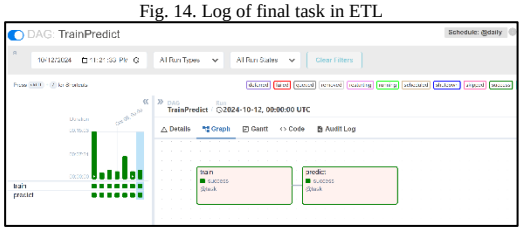

Fig. 15. Graph of train, predict tasks

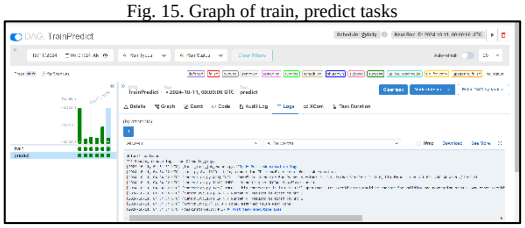

Fig. 16. Log of final task in dag