Building a Stock Price Prediction Analytics and Volume Analysis using Snowflake and Airflow

Gayatri Patil, Shweta Shinde

Department Of Applied Data Science, San Jose State University
San Jose, California, USA 95192 - 0250
gayatribhaskar.patil@sjsu.edu
shweta.shinde@sjsu.edu

Abstract— **Alpha Vantage offers a wealth of stock market data through its APIs, covering everything from stock prices to company news. This resource is invaluable for anyone conducting financial analysis. Using the stock price dataset from this resource, this project has two objectives- Stock Price Detection and Volume Analysis. For Stock Price Detection - The goal is to utilize the comprehensive financial data provided by Alpha Vantage APIs to analyse and predict stock market trends. The primary focus is on predicting future stock prices based on historical data and technical indicators. The project aims to create a predictive model that can assist investors and traders in making informed decisions. And the Volume Analysis component of the project aims to investigate trading volume data to understand market dynamics, specifically the buy and sell pressure in stock markets. By analysing trading volume, we can identify potential market trends, detect spikes that may indicate future price movements, and assess the correlation between trading volume and price changes. This analysis will provide valuable insights for investors and traders looking to make informed decisions based on market behaviour.**

Keywords—
ETL- Extract, Transform, Load
ARIMA - Autoregressive Integrated Moving Average
DAG – Directed Acyclic Graph

I. PROBLEM STATEMENT

This project is part of the Data Warehousing course at San Jose State University. The objective of Lab 1 is to demonstrate the process of data loading, forecasting, and data transformation using Apache Airflow and Snowflake. The lab focuses on establishing a data pipeline that connects a source database, processes the data, and loads it into a Snowflake data warehouse for analysis. In the context of this lab, the primary challenge is to build a robust data pipeline that effectively handles data extraction, transformation, and loading (ETL) from a source API into Snowflake. Additionally, we aim to perform time series forecasting on stock prices and enhance our dataset by creating lag and difference columns to better analyse trends.

II. SPECIFICATIONS

A. Data Sources

Using Alpha Vantage APIs, users are allowed to specify the stock symbol and the time range for historical data The project will leverage various endpoints from the Alpha Vantage API, which includes historical stock prices (open, high, low, close, volume).

B. Model Development

Utilizing Autoregressive Integrated Moving Average (ARIMA) models for baseline comparison. Train and assess these models using historical data. The model is implemented using Python code on Google Colab platform.

C. Data Processing

By using SQL queries created the tables in Snowflake that read data from the data source. Applied data cleaning by handling missing values and created relevant features from raw data, such as rolling averages and volume-weighted averages to enhance analysis.

D. Data Processing

Identifying significant increases for Volume spikes analysis in trading volume and analyze their context. Also helped to assess the relationship between trading volume and price changes over various timeframes. Hence, Summarized the results of the forecasting model and provided insights based on the analysis.

E. Visualization as Evaluation Metrics:

Time series plots and prediction intervals. Ans identification of patterns or anomalies in trading volume data.

III. SOLUTION REQUIREMENTS

- Alpha Vantage data source for dataset
- Google Colab platform for Python Implementation
- Development environment for Apache Airflow
- Development environment of Snowflake
- Cloud Composer from Google Cloud Platform

IV. SYSTEM ARCHITECTURE

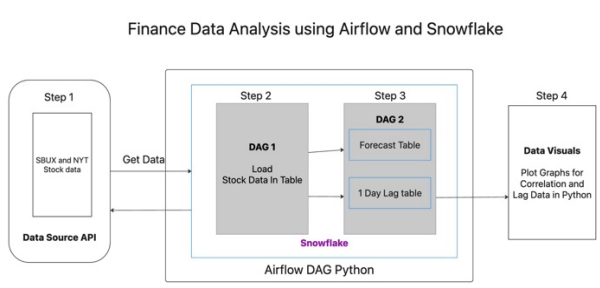


Fig. 1 System Diagram for Data Analysis process

The system diagram illustrates a data processing pipeline for financial data analysis using Airflow and Snowflake. The pipeline involves four main steps:

Step 1: Data Source:

- The pipeline starts with a data source API, likely fetching stock data for SBUX (Starbucks) and NYT (The New York Times).

Step 2: Data Ingestion:

- **DAG 1:** This Apache Airflow DAG is responsible for loading the fetched data into a Snowflake table.

Step 3: Data Transformation:

- **DAG 2:** Another Airflow DAG creates a "Forecast Table" and a "1 Day Lag table." These likely involve data transformations like calculating forecasts or creating lagged versions of the data.

Step 4: Data Visualization:

- Python code is used to plot graphs for correlation and lag data analysis. This step provides visual insights into the relationships between different variables in the data.

V. TABLES STRUCTURE

TABLE I
DEV.STOCK.STOCK_PRICE_ANALYSIS

Field	Data Type	Attributes	Constraints
date	DATE	NOT NULL	PRIMARY KEY
open	NUMBER(10, 2)	NOT NULL	
high	NUMBER(10, 2)	NOT NULL	
low	NUMBER(10, 2)	NOT NULL	
close	NUMBER(10, 2)	NOT NULL	
volume	BIGINT	NOT NULL	
symbol	VARCHAR(10)	NOT NULL	

TABLE 2
DEV.STOCK.STOCK_PRICE_STAGE

Field	Data Type	Attributes	Constraints
date	DATE	NOT NULL	PRIMARY KEY
open	NUMBER(10, 2)	NOT NULL	
high	NUMBER(10, 2)	NOT NULL	
low	NUMBER(10, 2)	NOT NULL	
close	NUMBER(10, 2)	NOT NULL	
volume	BIGINT	NOT NULL	
symbol	VARCHAR(10)	NOT NULL	

TABLE 3
DEV.STOCK.STOCK_PRICE_FORECAST

Field	Data Type	Attributes	Constraints
symbol	VARCHAR(50)	NOT NULL	
date	DATE		
forecast_close	FLOAT	NOT NULL	

TABLE 4
DEV.STOCK.STOCK_VOLUME _POINTS

Field	Data Type	Attributes	Constraints
date	DATE		

volume	NUMBER(38,0)	NOT NULL	
prev_volume	NUMBER(38,0)		
volume_change_pct	NUMBER(38,0)		
symbol	VARCHAR(10)		

VI. AIRFLOW DATA PIPELINES

Airflow is a platform used to automate and manage workflows by organizing tasks into Directed Acyclic Graphs (DAGs). It enables users to schedule tasks, monitor progress, and ensure data is processed in the correct order or in parallel, make it ideal for automating ETL processes in data pipelines.

For Python code and table data reference- click here.

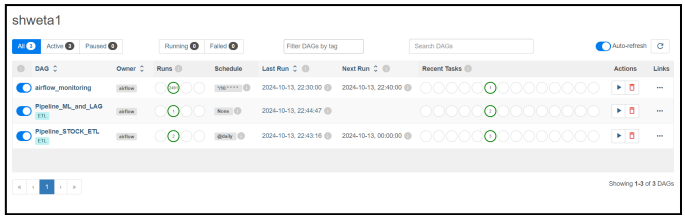


Fig. 2 Both Vantage Alpha API pipeline and ML Forecasting pipeline running as Airflow DAGs

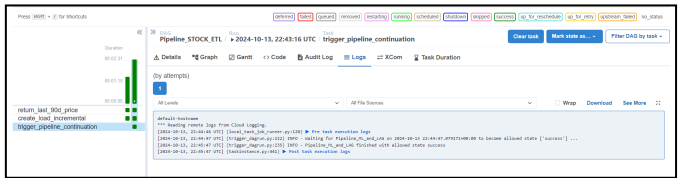


Fig. 3 Vantage Alpha API pipeline ETL - Airflow DAG

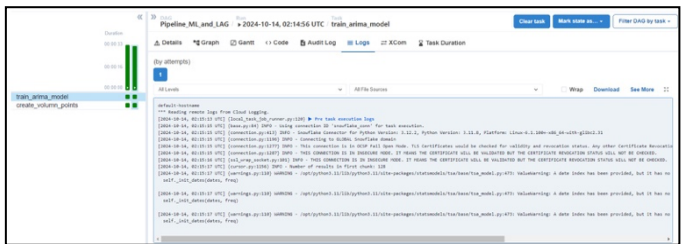


Fig. 4 LAG and ML Forecasting pipeline- Airflow DAG

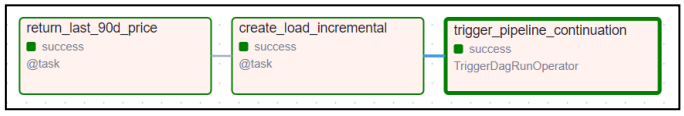
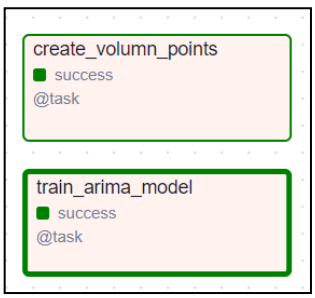


Fig. 5 Airflow Web UI- ETL Pipeline



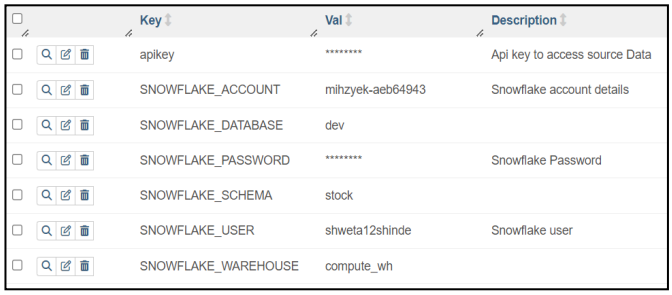
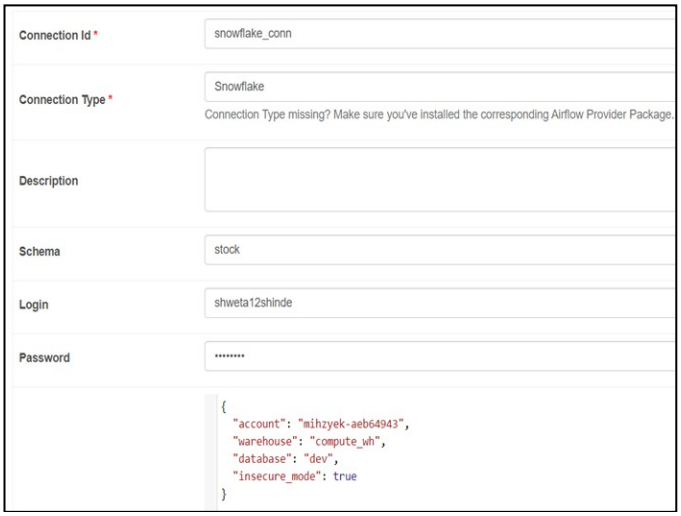


Fig. 6 Airflow Web UI- ML and LAG Pipeline

```python
def return_snowflake_conn():
    user_id = Variable.get('SNOWFLAKE_USER')
    password = Variable.get('SNOWFLAKE_PASSWORD')
    account = Variable.get('SNOWFLAKE_ACCOUNT')

    conn = snowflake.connector.connect(
        user=user_id,
        password=password,
        account=account,
        warehouse=Variable.get('SNOWFLAKE_WAREHOUSE'),
        database=Variable.get('SNOWFLAKE_DATABASE'),
        schema=Variable.get('SNOWFLAKE_SCHEMA')
    )
    return conn.cursor()
```

Fig. 7 Use of Airflow Variables in implementation




```python
def return_snowflake_conn():
    # Initialize the SnowflakeHook
    hook = SnowflakeHook(snowflake_conn_id='snowflake_conn')
    # Execute the query and fetch results
    conn = hook.get_conn()
    return conn.cursor()
```

Fig. 8 Snowflake Connection in implementation

Required libraries from the Python Package Index (PyPI)

Name	Version
snowflake-connector-python	-
requests	-
pandas	-
apache-airflow-providers-snowflake	-
snowflake-sqlalchemy	-
statsmodels	-
sensor	-
operators	-

Fig. 9 Use of PY Packages in Airflow

VII. SNOWFLAKE ML FORECAST

Snowflake's **ML Forecasting** model allows users to make time series predictions directly within the Snowflake platform. The model can predict future values based on historical data by using machine learning algorithms such as ARIMA The model requires input data, a timestamp column, and a target variable (such as stock prices or volume) to forecast future periods. This integrated approach simplifies the process, providing scalable machine learning capabilities for accurate forecasting within the cloud data warehouse. Click here to refer SQL.

```
-- Union your predictions with your historical data, then view the results in a chart.
SELECT SYMBOL, DATE, CLOSE AS actual, NULL AS forecast, NULL AS lower_bound, NULL AS upper_bound
    FROM STOCK_PRICE_ANALYSIS
UNION ALL
SELECT replace(series, '"', '') as SYMBOL, ts as DATE, NULL AS actual, forecast, lower_bound, upper_bound
    FROM My_stock_forecasts_2024_10_14;
```

Results | Chart

	SYMBOL	DATE	ACTUAL	FORECAST	LOWER_BOUND	UPPER_BOUND
121	NYT	2024-07-24 00:00:00.000	54.53	null	null	null
122	NYT	2024-07-23 00:00:00.000	55.29	null	null	null
123	NYT	2024-07-22 00:00:00.000	55.09	null	null	null
124	NYT	2024-07-19 00:00:00.000	52.39	null	null	null
125	NYT	2024-07-18 00:00:00.000	52.64	null	null	null
126	NYT	2024-07-17 00:00:00.000	53.29	null	null	null
127	NYT	2024-07-16 00:00:00.000	53.74	null	null	null
128	NYT	2024-07-15 00:00:00.000	53.58	null	null	null
129	SBUX	2024-10-14 00:00:00.000	null	95.678764521	93.47069554	97.93486481
130	SBUX	2024-10-15 00:00:00.000	null	95.802488924	93.31069526	98.269811301
131	SBUX	2024-10-16 00:00:00.000	null	95.775655797	93.104400402	98.374248838
132	SBUX	2024-10-17 00:00:00.000	null	95.70756026	92.906506339	98.546098951
133	SBUX	2024-10-18 00:00:00.000	null	95.745587624	92.832777108	98.623863536
134	SBUX	2024-10-21 00:00:00.000	null	95.649108222	92.702469998	98.549106636
135	SBUX	2024-10-22 00:00:00.000	null	95.752316341	92.610045019	98.867405041
136	SBUX	2024-10-23 00:00:00.000	null	95.792634325	92.545213598	98.9466446

```
-- Inspect the accuracy metrics of your model.
CALL stock_forecast!SHOW_EVALUATION_METRICS();
```

Results | Chart

	SERIES	ERROR_METRIC	METRIC_VALUE	STANDARD_DEVIATION	LOGS
1	"NYT"	MAE	0.853	NaN	null
2	"NYT"	MAPE	0.015	NaN	null
3	"NYT"	MDA	0.417	NaN	null
4	"NYT"	MSE	1.161	NaN	null
5	"NYT"	COVERAGE_INTERVAL=0.95	0.923	NaN	null
6	"SBUX"	MAE	1.309	NaN	null
7	"SBUX"	MDA	0.75	NaN	null
8	"SBUX"	MSE	2.17	NaN	null
9	"SBUX"	SMAPE	0.014	NaN	null
10	"SBUX"	COVERAGE_INTERVAL=0.95	1	NaN	null
11	"NYT"	WINKLER_ALPHA=0.05	3.661	NaN	null
12	"NYT"	SMAPE	0.016	NaN	null
13	"SBUX"	WINKLER_ALPHA=0.05	5.769	NaN	null
14	"SBUX"	MAPE	0.014	NaN	null

Fig. 10 Screenshot for Query execution results

VIII. VOLUME ANALYSIS- CHARTS

Objective: Understand market buy/sell pressure by analyzing stock trading volume. For plot code click here.

Methodology:

- **Data**: Stock trading volume and price data.
- **Correlation**: A correlation analysis was performed to determine the relationship between volume and price changes.
- **Findings**: Significant spikes in trading volume were observed to correlate with price movements, helping predict market trends.

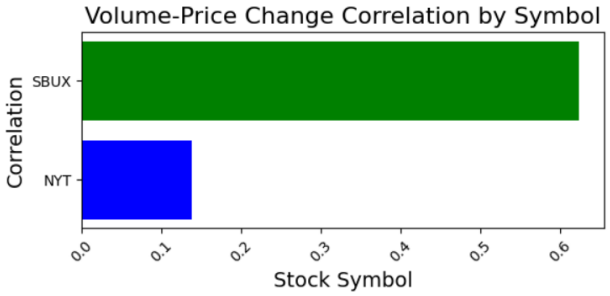


Fig. 11 Volume Price Change Correlation Analysis

Scatter Plot Analysis

Objective: Analyze trading volume changes over time to assess market buy/sell pressure for different stock symbols.

Data Used: Stock trading volume data for symbols "NYT" and "SBUX".

Analysis:

- The scatter plot visualizes volume changes across various dates.
- Significant volume spikes, particularly the one over 1000 units, can indicate potential price movements or unusual trading activity.
- The plot highlights the variability in volume changes between different symbols, suggesting varying market interest.

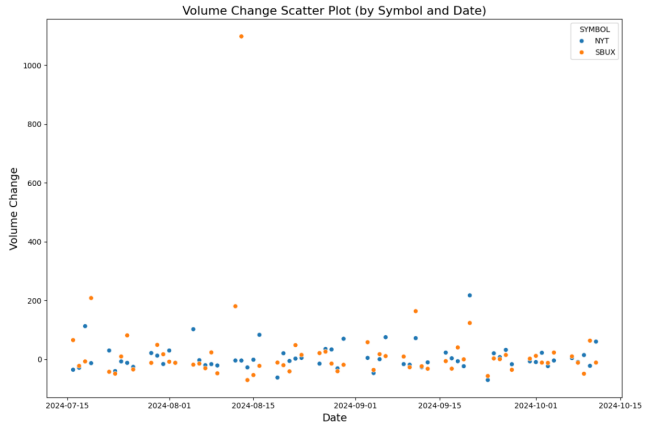


Fig.12 Volume Change Analysis by Scatter Plot.

IX. GITHUB LINK

https://github.com/ShindeShwetaK/DW_LAB1

X. FUNCTIONAL ANALYSIS

This project aims to empower users with data-driven insights into stock price movements, aiding them in making more informed financial decisions. Detailed functional analysis for each step of project can be considered as below:

A. *Data Sources*

Implementation of API calls to fetch stock data from Alpha Vantage. Scheduling regular data updates by implementing incremental load to ensure the dataset remains current.

B. *Data Analysis*

Conducted exploratory data analysis (EDA) to understand historical trends and patterns in stock prices and volume trends in data. Visualized the predicted data using charts and graphs to illustrate key findings.

C. *Model Development and Evaluation*

Split data into training and testing sets. Developed the ARIMA predictive models based on historical data. Evaluated the models using test data which helped to find out forecasted stock prices for next 10 days.

D. *Visualization*

Create visualizations by using plotting graphs through matplotlib library for model performance.

XI. CONCLUSIONS

The lab effectively showed how to create a data pipeline, from extracting data to loading it into a Snowflake data warehouse. We built a machine learning model to forecast prices and made a table to analyse volume changes using the LAG function. The visualizations helped us see how stocks are related and how trends shift. Overall, this experience improved our understanding of data warehousing and its real-world uses.

ACKNOWLEDGMENT

The lab project of Building a Stock Price Prediction Analytics using Snowflake and Airflow, was an enriching experience because of the guidance and support of professor Keeyong Hun and TA Revanth Kumar Bondada. We would like to extend our gratitude towards them for their insights and expertise which turned out to be instrumental in helping us navigate the complexities of the project. Thank you for fostering a collaborative and engaging learning environment that encouraged us to explore and innovate. We greatly appreciate your dedication and encouragement.

REFERENCES

[1] Professor Keeyong Hun, DATA226 Lecture Notes Week 4, Week 5, Week 6, Week 7

[2] Alpha Vantage API Key & Symbol Selection, https://www.alphavantage.co/documentation/