Data226 Building a Stock Price Prediction Analytics using Snowflake & Airflow

Lab is for a group project (2 person team).

Lab 1

This lab project is based on analyzing the finance data from yfinance API and titled as 'Building a Finance Data Analytics'

yfinance provide comprehensive stock market data. This dataset is a valuable resource for anyone interested in various financial analysis (examples to follow). The APIs include information about stock data (open, high, low, close, volume)

With this dataset, you can do the following:

1. **Stock Price Prediction**

- Objective: Predict future stock prices based on historical data.
- Data Used: Historical stock prices and technical indicators.
- Example Analysis: Develop predictive models using machine learning (e.g., LSTM, ARIMA) for short-term or long-term stock price forecasting.

2. Stock Trend Analysis

- Objective: Analyze stock price patterns to identify upward and downward trends.
- Data Used: Historical stock data (Open, High, Low, Close, Volume).
- Example Analysis: Use moving averages, Bollinger Bands, and trend lines to analyze key stock trends and price volatility.

3. Technical Indicators Analysis

- Objective: Utilize technical indicators like RSI, MACD, and SMA to identify buy and sell signals.
- Data Used: Stock price and volume data.
- Example Analysis: Analyze oversold/overbought conditions or crossover points to predict optimal trading times.

5. Portfolio Optimization

- Objective: Analyze correlations between multiple stocks to construct an optimal investment portfolio.
- Data Used: Historical stock data from multiple stocks.

- Example Analysis: Create a portfolio that maximizes returns while minimizing risk by optimizing the risk-return ratio.

6. Volatility Analysis

- Objective: Measure stock volatility to assess risk levels.
- Data Used: Daily returns data.
- Example Analysis: Compare historical volatility with market volatility indices like VIX to assess the risk of specific stocks.

7. Volume Analysis

- Objective: Analyze trading volume to understand market buy/sell pressure.
- Data Used: Stock trading volume data.
- Example Analysis: Analyze spikes in volume to determine if they indicate impending price movements or assess the correlation between volume and price changes.

8. Sentiment Analysis and Stock Reaction

- Objective: Analyze the impact of news or social media sentiment on stock prices.
- Data Used: News articles, Twitter data, etc.
- Example Analysis: Use text mining to classify news as positive or negative and correlate it with stock price movements.

These analyses provide valuable insights for investment decision-making, risk management, market forecasting, and company financial health assessment. Depending on the project's specific requirements, the scope and depth of each analysis can be tailored to meet the desired outcomes. For Building a Stock Data Analysis, use the yfinance APIs: Click Here

Project Details

We will all implement the same data analytics system ("**Stock Price Prediction")** which is to forecast the stock prices of a company for the next 7+ days using the last 180 days of the prices:

1. Choose at least two companies (for example, NVDA & APPL)
2. Get their stock prices of the last 180 days from yfinance API (https://pypi.org/project/yfinance/) and populate the info in a table in your Snowflake.
3. (Table schema) The table should have the following info:
    a. Stock symbol
    b. Date
    c. Open, Close, Min, Max, Volume
4. (Python codes) Step 2 needs to be running **daily** in the form of Airflow DAG (data pipeline)
5. (SQL queries) Set up ML forecasting tasks in your Snowflake to predict the next 7 days of the stock prices
6. (Python codes) Step 5 should be implemented as Airflow data pipelines and the final table needs to be unioned as demonstrated here

Project Report

The project proposal should clearly define the application system that your team is proposing and the results of your requirements analysis and conceptual design. The proposal does not have to be long, three to five typed pages plus supporting diagrams and the like is probably sufficient. All the Python codes and SQL queries should be in a Github repo and the links should be provided. Be concise, but thorough.

The project report should include the following sections:

1. Problem Statement - what application system does your team to build and why; why are a database and data pipelines needed as part of the system.

2. Solution Requirements - what are the requirements for a solution, what will the system do, what are its limitations, how will people use the system.

3. Functional Analysis - discuss the functional components of the application system that you are proposing and how they collectively solve the problem. Include database & data pipeline interactions for each.

4. Tables structure, Screenshots, Python codes, and SQL queries

Grading Criteria

1. Problem statement, requirements and specifications

2. Overall system diagram

3. Table(s) structure (Fields, attributes, constraints): all tables should be covered!

4. Provide the following:

- Airflow data pipeline codes in Python
    - Both yfinance pipeline and ML Forecasting pipeline should be running as Airflow DAGs
- Screenshot of the Airflow Web UI showing two pipelines you created
- Demonstrate proper use of Airflow connections and variables in your implementation
- As demonstrated (here), forecasting job needs to union two tables (one from ETL and the other from forecasting) to create a final table
- Should use SQL transaction along with try/except

5. Submit Lab 1 report in IEEE format (Single Column format is fine)

SQL & Airflow codes should be in Github and place the links in step 4