Distributed Daily Stock Price Forecasting for Tech Companies using Spark and Distributed Storage

Keyan Luo
Kennesaw State University
Kennesaw, Georgia, USA

CCS Concepts

• **Information systems → Data mining**; • **Computing methodologies → Machine learning**.

Keywords

Spark, Distributed Computing, Stock Price Prediction, Financial Big Data

1 Project Overview

1.1 Project Title

Distributed Daily Stock Price Forecasting for Tech Companies using Spark and Distributed Storage

1.2 Domain

Financial Market Analysis, Stock Trading Prediction.

1.3 Problem Statement

In the increasingly complex and volatile financial markets, accurate stock price prediction is crucial for informed investment decisions. However, traditional single-machine forecasting methods face limitations in efficiency and scalability when dealing with massive historical stock data, technical indicators, and inter-company influences. This project aims to build a distributed system that processes a dataset of technology stocks (FAANG+) with 10 years of daily data (OHLCV, technical indicators) using the Spark framework, and forecasts the next-day closing price, thereby validating the feasibility and advantages of distributed processing for large-scale time-series prediction.

1.4 Scope

Will Implement:

- Develop a system capable of loading and processing a dataset of FAANG+ companies' stocks (AAPL, MSFT, AMZN, GOOGL, META, NVDA) with 10 years of daily data from distributed storage (e.g., HDFS/S3).
- Utilize Spark for data grouping (by company) and basic analysis.
- Explore and apply at least 2-3 course stack concepts (e.g., Distributed Storage like HDFS/S3, Spark RDD/DataFrame processing, and potentially Data Models like Parquet or Wide-column stores).
- Train a machine learning model (e.g., a regression model from Spark MLlib or a simple LSTM) to predict the next-day closing price, based on the technical indicators provided in the dataset.
- Generate and store (e.g., in CSV or Parquet files) the prediction results for the test set.
- Present the system's architecture diagram and data flow diagram in the proposal.

Will Not Implement:

- Real-time data stream processing (this project is based on a static daily dataset).
- Complex NLP techniques for news sentiment analysis (although relevant information might be present in the dataset, this project focuses on structured data and technical indicators for prediction).
- Deployment of a production-grade, highly available forecasting service.
- Automatic adjustment for stock splits or dividends (the dataset provides "Adj Close", but we will primarily focus on "Close" as input and "Next_Day_Close" as the target).

2 System Description

2.1 Data Sources

- **Primary Source:** The "Stock Price Dataset & Forecasting" dataset from Kaggle [6]. This dataset contains 10 years of daily stock market data for leading tech companies, enhanced with pre-calculated technical analysis indicators and the next-day closing price as the prediction target.
- **Data Format:** CSV file.
- **Companies Covered:** Apple (AAPL), Microsoft (MSFT), Amazon (AMZN), Google (GOOGL), Meta (META), Nvidia (NVDA).

2.2 Data Characteristics

- **Volume:** 10 years of daily data covering 6 companies. Assuming approximately 252 trading days per year, this results in roughly 15,120 rows. When loaded into Spark, it can be converted to Parquet format for optimized storage and read efficiency.
- **Variety:**
  - Structured Data: Date, stock ticker, prices (Open, High, Low, Close, Adj Close), trading volume.
  - Numerical Features: Pre-calculated technical indicators, daily returns, volatility.
  - Categorical Data: Stock Ticker.
- **Velocity:** The data is daily, making it suitable for Batch Processing. The prediction target is based on the previous day's closing price, thus suitable for offline prediction based on historical data.

2.3 Stack Layers

This project will focus on and apply key layers of the Big Data Stack:

1. **Storage:**
   - *Concept:* Distributed File Storage (e.g., HDFS or S3).
   - *Application:* The initial CSV dataset will be uploaded to a distributed file system and converted into a more optimized format.
2. **Syntax / Data Formats:**
   - *Concept:* Data Formats (e.g., CSV, Parquet).
   - *Application:* After loading the raw CSV data, we will convert it to Parquet format. Parquet is a columnar storage format that offers better compression and query performance, ideal for Spark's large-scale data processing [1].
3. **Processing:**
   - *Concept:* Spark RDDs / DataFrames, Distributed Computation, Grouping (by Ticker).
   - *Application:* Using Spark DataFrames API to load, filter, group (by company Ticker), and preprocess the data. We will use Spark to train predictive models [2].
4. **(Optional 4th Concept) Data Stores:**
   - *Concept:* Wide-column model or Document stores.
   - *Application:* Processed features for model training can be saved as Parquet files, or potentially loaded into HBase for fast retrieval.

This project will explicitly demonstrate the application of **Distributed File Storage (Storage)**, **Data Formats (Syntax - Parquet)**, and **Spark RDD/DataFrame Processing (Processing)**.

2.4 Assumptions

- The accuracy and integrity of the provided dataset are assumed.
- Past price trends and technical indicator patterns have predictive power for future price movements.
- The selected machine learning model(s) can learn effective prediction patterns from the data.
- Stock splits and dividends are not precisely accounted for in the "Close" price used for prediction.
- Real-world trading factors such as transaction costs and taxes are ignored.

3 Implementation Approach

3.1 Technology Choices

- **Core Framework:** Apache Spark (using PySpark).
- **Storage:** HDFS (or Amazon S3).
- **Data Format:** Parquet.
- **Machine Learning Library:** Spark MLlib.
- **Diagramming Tool:** draw.io.
- **Programming Language:** Python.

3.2 Processing Model

Batch Processing: Due to the daily historical nature of the dataset, we will adopt a batch processing model. Data will be loaded, processed, models trained, predictions made, and results stored offline.

3.3 Scalability Plan

- **Data Storage:** Using HDFS/S3 ensures horizontal scalability for data storage.
- **Data Processing:** Spark's distributed computing architecture inherently supports horizontal scaling. By adding more nodes to the Spark cluster, we can handle larger datasets or accelerate computation tasks.
- **Model Training:** Spark MLlib's distributed model training capabilities are crucial for large datasets. To ensure scalability in distributed prediction, the system will go beyond simple data partitioning. As emphasized in [3], true scalability lies in distributed computation. We will leverage Spark's Pandas UDFs or Spark ML Pipelines to train independent model instances in parallel for each stock ticker. This approach ensures that both data processing and model training are genuinely distributed, enabling the system to handle larger volumes and more complex models efficiently.

3.4 Metrics

Prediction Accuracy:

- Mean Squared Error (MSE) / Root Mean Squared Error (RMSE).
- Mean Absolute Error (MAE).
- R-squared (R^2).

System Performance:

- Processing Time: Total time for data loading, feature processing, model training, and prediction.

4 Literature Review

This section provides a detailed examination of the foundational academic literature that informs our project's methodology and technical choices. Each cited work is discussed individually, highlighting its core contributions, methodologies, and relevance to our work, in a style that emulates the thoroughness of MLA format documentation.

4.1 Brahmane and Krishna (2021): Big Data Classification using Deep Learning and Apache Spark Architecture

Anilkumar V. Brahmane and B. Chaitanya Krishna, in their 2021 publication "Big Data Classification Using Deep Learning and Apache Spark Architecture" in *Neural Computing and Applications* (Vol. 33, No. 22, pp. 15253-15266), propose a novel architecture for big data classification that integrates deep learning with Apache Spark. The authors address the challenge of efficiently handling large datasets by leveraging Spark's distributed processing capabilities. Their core technical contribution is a specialized "feature selection and arrangement" process, termed the RCBO algorithm, implemented within the initial stages of Spark processing. This methodology is particularly relevant to our project as it addresses the need for effective feature engineering on extensive datasets. We plan to adapt their approach by utilizing Spark's DataFrame API for feature selection and arrangement on our 10 years of tech stock data, including pre-computed technical indicators like `RSI_14` and `MACD`.

4.2 Behera et al. (2020): Comparative Study of Real-Time Machine Learning Models for Stock Prediction through Streaming Data

Ranjan Behera and colleagues, in their 2020 paper "Comparative Study of Real-Time Machine Learning Models for Stock Prediction through Streaming Data," published in *JUCS - Journal of Universal Computer Science* (Vol. 26, No. 9, pp. 1128-1147), present a comparative study of machine learning models for stock prediction using Spark as their distributed platform. Although their work focuses on real-time streaming data, which differs from our batch processing approach, the fundamental principle of using a distributed system for stock price prediction is directly applicable. They highlight the importance of scaling models to handle multiple companies and growing datasets, proposing data grouping by ticker as a key strategy. We will adopt this strategy, using Spark to group our data by stock ticker (AAPL, MSFT, etc.) to ensure our models can effectively process and predict for each company independently and efficiently. Their comparative analysis of different machine learning models also provides valuable insights for our model selection process, guiding our use of Spark MLlib for regression and potentially other models.

4.3 Koduppanapolackal et al. (2025): Real-Time Stock Price Prediction and Visualization Using LSTM and Matplotlib

James J. Koduppanapolackal and collaborators, in their 2025 *International Conference on Next Generation Communication & Information Processing (INCIP)* paper "Real-Time Stock Price Prediction and Visualization Using LSTM and Matplotlib" (pp. 605-609), address the increasing complexity and volatility of stock markets by applying a big data-based approach using the Long Short-Term Memory (LSTM) algorithm. Their study focuses on analyzing historical stock data to predict market trends and the next-day closing price ("Next_Day_Close"), which directly supports our project's primary objective. We intend to adapt their LSTM approach for our use case by configuring the model to incorporate our pre-computed technical indicators, such as `SMA` and `EMA`. By doing so, we aim to potentially enhance prediction accuracy, building on their findings that these indicators can provide valuable context for market trend analysis and volatility prediction.

4.4 Liang (2024): ARIMA with Attention-Based CNN-LSTM and XGBoost Hybrid Model for Stock Prediction in the US Stock Market

Luocheng Liang, in the 2024 *SHS Web of Conferences* publication (Vol. 196, Article 02001), titled "ARIMA with Attention-Based CNN-LSTM and XGBoost Hybrid Model for Stock Prediction in the US Stock Market," introduces a sophisticated hybrid modeling approach for stock prediction. This hybrid strategy combines the ARIMA model with Attention-based CNN-LSTM and XGBoost, designed to improve predictive capabilities for complex, nonlinear time series data. This approach is particularly relevant given the high volatility of tech stocks like Nvidia (NVDA). The study's focus on companies in the "US stock market" aligns well with our FAANG+ dataset's domain. We plan to leverage the concept of using `XGBoost` for its proven accuracy, especially in volatile market conditions. Our Spark pipeline will be configured to support experimentation with such advanced ensemble techniques, enabling us to explore hybrid modeling logic for enhanced prediction performance.

4.5 Li (2024): Study on the Fitness of ARIMA Model in Stock Forecasting

Jingwen Li, in the 2024 *SHS Web of Conferences* publication (Vol. 208, Article 01028), titled "Study on the Fitness of ARIMA Model in Stock Forecasting," investigates the suitability of the ARIMA model for stock forecasting. The paper emphasizes the importance of technical analysis and statistical modeling for predicting asset prices in international markets, using the SP500 as an example. It highlights the concept of "accuracy of asset price prediction" and the crucial role of data preparation. We will apply their validation methodologies to assess the accuracy of our "Next_Day_Close" predictions. Additionally, we will adapt the ARIMA model's data preparation steps by configuring Spark to perform necessary data cleaning and statistical preprocessing. This includes conducting stationarity tests and applying differencing to our 10-year tech stock data, as discussed in relation to ARIMA's requirements and general time-series data quality considerations [4, 5], ensuring our data is suitable for robust statistical modeling.

5 Architecture Diagrams

5.1 Stack Architecture Diagram

5.2 Data Flow / Pipeline Diagram

References

[1] Apache Parquet Project. Apache parquet documentation. https://parquet.apache.org/, 2024.

[2] Apache Spark Project. Machine learning library (mllib) guide. https://spark.apache.org/docs/latest/ml-guide.html, 2024.

[3] Ranjan Behera et al. Comparative study of real time machine learning models for stock prediction through streaming data. *JUCS - Journal of Universal Computer Science*, 26(9):1128–1147, 2020. doi: 10.3897/jucs.2020.059.

[4] Anilkumar V. Brahmane and B. Chaitanya Krishna. Big data classification using deep learning and apache spark architecture. *Neural Computing and Applications*, 33(22):15253–15266, 2021. doi: 10.1007/s00521-021-06145-w.

[5] Jingwen Li. Study on the fitness of arima model in stock forecasting. *SHS Web of Conferences*, 208:01028, 2024. doi: 10.1051/shsconf/202420801028.

[6] Vishard Mehta. Faang stock market data with technical indicators. https://www.kaggle.com/datasets/vishardmehta/faang-stock-market-data-with-technical-indicators, 2024.

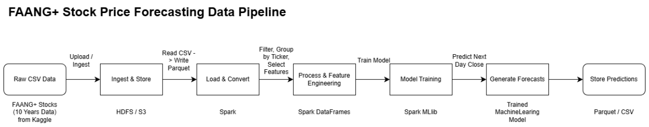
FAANG+ Stock Price Forecasting Data Pipeline
Raw CSV Data
Upload / Ingest
Ingest & Store
Read CSV -> Write Parquet
Load & Convert
Filter, Group by Ticker, Select Features
Process & Feature Engineering
Train Model
Model Training
Predict Next Day Close
Generate Forecasts
Store Predictions
FAANG+ Stocks (10 Years Data) from Kaggle
HDFS / S3
Spark
Spark DataFrames
Spark MLlib
Trained MachineLearing Model
Parquet / CSV

Figure 2: FAANG+ Stock Price Forecasting Data Pipeline

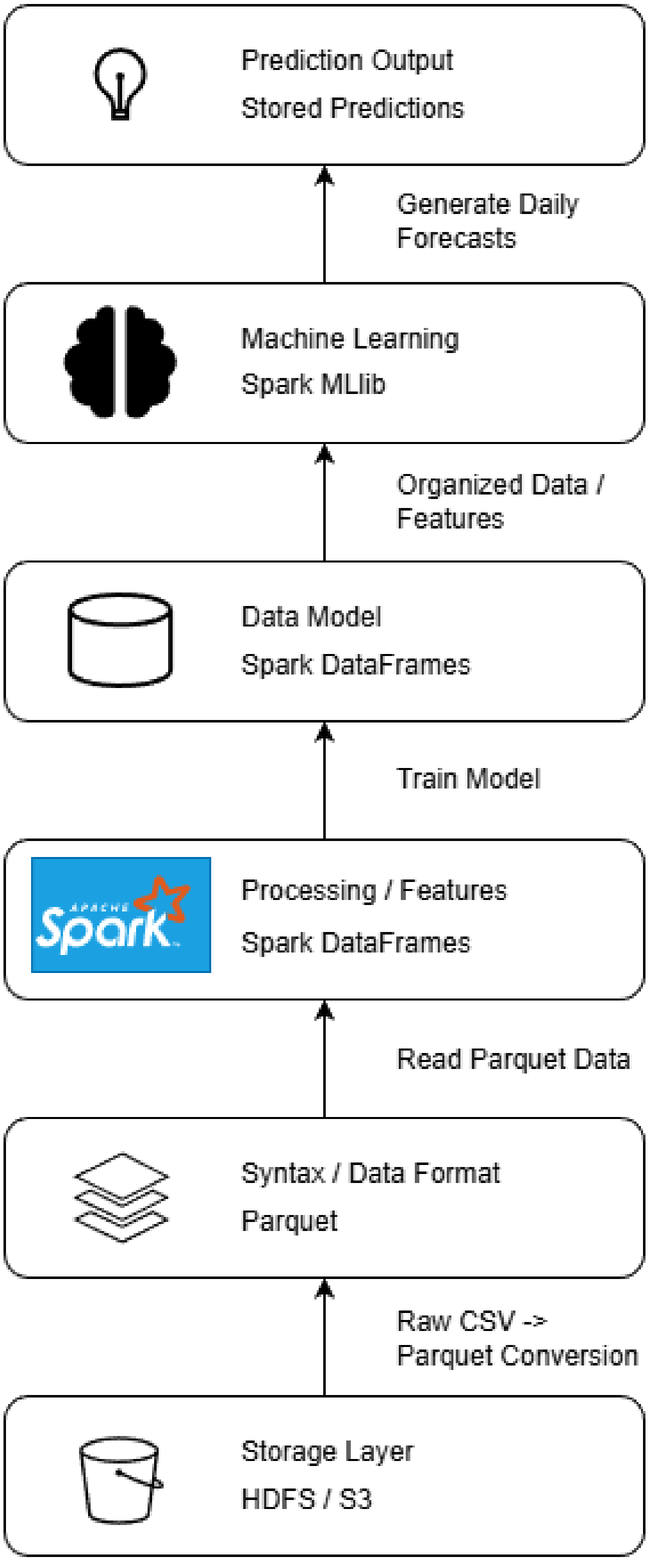

Figure 1: System Stack Architecture for FAANG+ Stock Price Forecasting