Project Proposal
ID2221 Data Intensive Computing

Tangyujun Han Gabriel Françon Syed Arif Rahman
tanghan | gfrancon | sarahman @kth.se

September 27, 2023

1 Problem Statement

The financial market is a complex ecosystem, with stock prices influenced by a myriad of factors ranging from company-specific news to global economic trends. Accurate prediction of stock prices is a challenge that has significant implications for investors, traders, and financial institutions. Traditional methods, relying on fundamental and technical analysis, often fall short in capturing the dynamic nature of the market, especially in the short term. With the advent of big data technologies and machine learning, there is an opportunity to harness vast amounts of real-time data to improve the accuracy of stock price predictions.

2 Tools

The following tools will be used in this project:

- Programming Language : Pyhton and Scala
- Distributed Messaging System: Kafka
- Storage: Cassandra
- Stream Processing: Spark
- Machine learning model: Spark MLlib

3 Data

For this project, our primary data source is yfinance[1], a reliable and widely-used library that provides access to historical and real-time stock data from Yahoo Finance. This ensures that our predictive models are built upon comprehensive and up-to-date market information, enhancing the accuracy and relevance of our stock price predictions.

4 Methodology and algorithm

4.1 Data Collection and Preprocessing

Use yfinance to fetch historical and real-time stock data. Preprocess the data to handle missing values, outliers, and any noise. This might involve techniques like imputation, normalization, and transformation. Feature engineering: Derive new features from the existing data, such as moving averages, volatility, trading volume changes, etc.

4.2 Data Streaming and Storage

Use Kafka to stream real-time stock data. This will allow the system to process data in real time and make predictions on the fly. Store historical and real-time data in a distributed storage system like Cassandra for efficient retrieval and processing.

4.3 Model training

Split the preprocessed data into training and testing sets. Use Spark MLlib to train various machine learning models on the training set. This might include regression models, time series models, and ensemble models. Evaluate each model's performance on the testing set using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2.

4.4 Model Deployment

Deploy the best-performing model in a distributed environment. Use Kafka to stream real-time data to the model for real-time predictions.

References

[1] Ran Aroussi. Download market data from Yahoo! Finance's API, September 2023. original-date: 2017-05-21T10:16:15Z. URL: https://github.com/ranaroussi/yfinance.