Exchanged-Traded Funds

Chenting Zhang, Changrong Li, Duc Minh Pham

October 20, 2023

1 Introduction

In an ever-evolving financial landscape, the selection of investment vehicles is a pivotal decision that can significantly impact an investor's portfolio performance. Among these options, Exchange-Traded Funds (ETFs) have gained widespread popularity due to their diversified nature and ease of trading.

Our project focuses on developing a quantitative ETF selection strategy tailored to the US market. We utilized market data sourced from Yahoo! Finance's API to analyze and identify optimal ETFs for our investment. Our project was executed on the Databricks notebook platform, including Apache Spark Streaming, Kafka, and Python, to streamline data processing and enhance our decision our decision-making capabilities.

To execute the project, one simply needs to run the IPython notebook file on Databricks. The setup for Kafka is already configured and hosted in the Confluent cloud, so there are no additional configuration steps required.

2 Data

For this project, we utilized datasets from Yahoo Finance, collecting the data through the 'yfinance' module, which offers convenient access to historical financial data. Our dataset consists of information on six Exchange-Traded Funds (ETFs) selected for analysis, specifically:

- "AXP" (American Express Company)
- "DIA" (SPDR Dow Jones Industrial Average ETF Trust)
- "IWM" (iShares Russell 2000 ETF)
- "KWEB" (KraneShares CSI China Internet ETF)
- "QQQ" (Invesco QQQ Trust)
- "SPY" (SPDR S&P 500 ETF Trust)

For each of these ETFs, we retrieved the following columns when fetching the data from the source:

- Date: The date corresponding to the data entry.
- Adjusted Close: The adjusted closing price of the ETF on the given date.
- Close: The closing price of the ETF on the given date.
- High: The highest trading price of the ETF on the given date.
- Low: The lowest trading price of the ETF on the given date.
- Open: The opening price of the ETF on the given date.
- Volume: The trading volume of the ETF on the given date.

Our data collection period began on January 1st, 2019, and extends up to the current date. We collected data at a daily interval for each ETF, providing us with a comprehensive historical dataset to base our analysis and selection strategy upon. The historical market data is essential for evaluating the performance and behavior of these ETFs over time. This dataset will be a fundamental component of our project, helping us develop and test our ETF selection strategy.

3 Methodology and Algorithm

Our project involves a multi-step process that leverages various tools and techniques to select and analyze Exchange-Traded Funds (ETFs) in the US market. The methodology can be summarized as follows:

1. **Data Collection**: We initiated the project by fetching historical market data using the Python `yfinance` module. This data includes key metrics for a selection of six ETFs over the specified time period.

2. **Data Streaming to Kafka**: The collected data was then streamed into a Kafka topic called "assets". The Kafka queue in confluent cloud is displayed in figure **??**

3. **Data Processing with Apache Spark**: We consumed the data from the Kafka topic using Apache Spark. The data was structured into a Spark DataFrame for subsequent analysis.

4. **Rate of Change (ROC) Calculation**: One of the primary steps in our analysis was to calculate the Rate of Change (ROC) for the financial dataset. ROC is a key metric that helps us understand the momentum and volatility of each ETF.

5. **Daily Returns Computation**: Using the ROC values, we computed the daily returns for each ETF. These daily returns provide insights into the daily performance of the ETFs.

6. **Top ETF Selection**: We identified the top three ETFs with the highest ROC values. This selection was based on the ROC metric and serves as a foundation for our portfolio strategy.

7. **Monthly Portfolio Returns**: To evaluate the performance of our selected ETFs over a longer horizon, we calculated daily portfolio returns. This allowed us to assess the effectiveness of our ETF selection strategy over time.

8. **Data Visualisation**: We visualise the investing returns of our dynamic portfolio constructed and run according to our strategy as well as the baseline (The Standard and Poor's 500 index anchored ETF SPY here), to make the effect of ETF selection clear. The result is illustrated in figure 1.

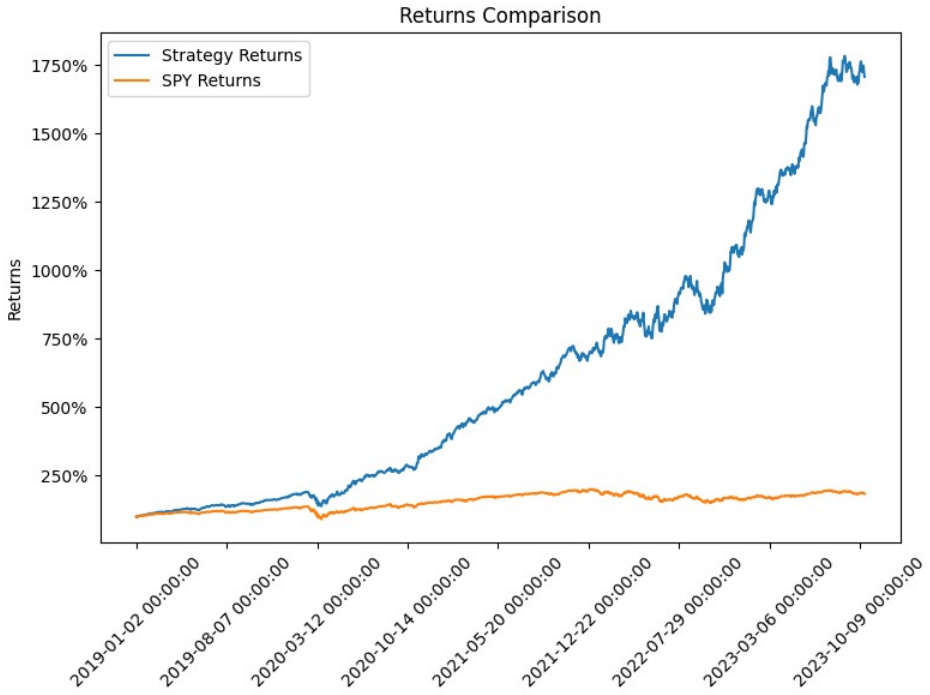

Figure 1: Result

4 Challenges

During the course of our project, we encountered several challenges that required us to adapt and find solutions. These challenges are integral to the learning process and the evolution of our project. Some of the key challenges we faced include:

1. **Docker Configuration**: Initially, we attempted to implement a full application using Docker containers. While this approach offers many advantages, we faced challenges related to Docker configuration, particularly concerning Spark Streaming. To work with Spark Streaming, we needed to set up both our local development environment and Docker instances. This was due to version mismatches that arose when attempting to use Spark in the Docker container.

2. **Spark-Kafka Integration**: Although we successfully used 'spark-submit' to deploy our Spark application to the Spark cluster, we encountered a perplexing issue where Spark could not locate the Kafka cluster, despite our application functioning correctly before submission. This problem stemmed from Docker configuration, which we struggled to resolve effectively.

The initial structure of the project, as depicted in Figure 2, provided us with a foundation, but these challenges prompted us to reassess our technical approach and continually refine our strategies to overcome them. Such challenges, while demanding, offer valuable opportunities for learning and growth in our project journey.

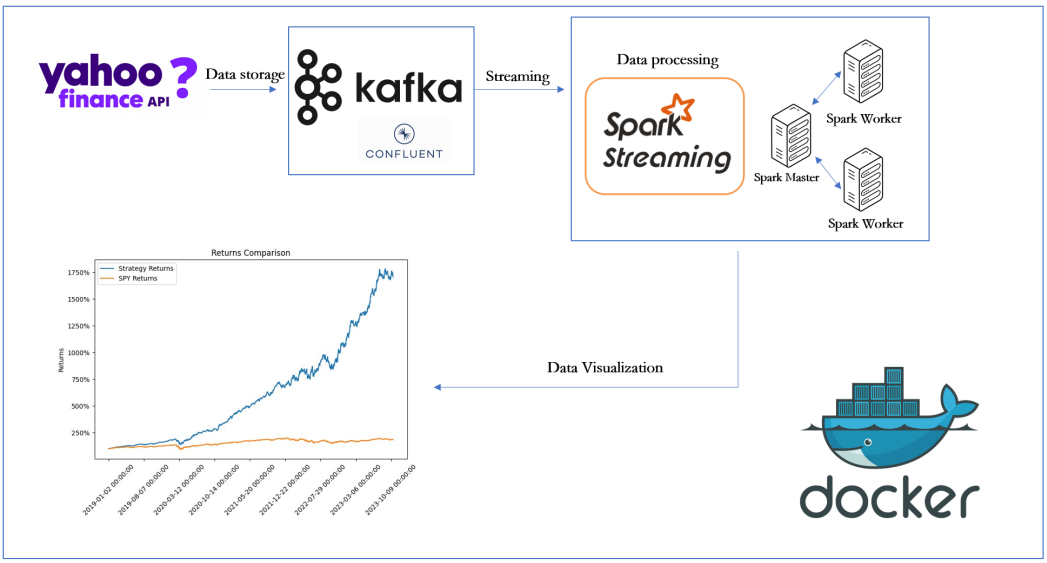

Figure 2: Inital architecture