Assignment Part B

Matic Kristanc (720082718)

University of Exeter

BEEM061 Fundamentals of Financial Technology

Jack Rogers

16th January 2023

Contents

Table of figures

Introduction to part B

All the relevant code for the technical part of this assignment is available through the following link: assignment files

1 Exploring the Bitcoin Blockchain and basic Web Coding

1.1 Extracting information form the transaction

Our chosen transaction has a block height of "758992". In our case we were tasked with extracting the following data:

- **txcoun**t – includes the number of transactions in the block
- **time** – includes information on when the block was mined, with date and hour
- **totalFees** – tells us the total amount of fees in BSV used for all the transactions in the block
- **confirmations** – gives us the number of times the block was confirmed to the current time – every new block mined in the chain serves as an additional confirmation
- **miner** – gives us the alias of the miner who mined this block

2 Time series investigation of asset prices

For our analysis we have chosen Apple stock (AAPL) to represent our risky asset, Coca-cola stock (KO) as our relatively safe asset, and the S&P 500 index (GSPC) as a market proxy. Our baseline for choosing the risky and safe asset was the S&P 500 index. We compared the volatility represented by the standard deviation and the market betas to select our two chosen stocks. Apple stock has volatility and beta that is higher than the market, while inversely Coca-cola has both metrics lower than the market.

We chose the start date of our analysis to be the 15th December 1980, as this is as far as the history for AAPL goes, which has the least historical data of the three.

2.1 Data analysis

We first compared the correlations of the assets. As seen in the figure below the lowest correlation is between the AAPL and KO stock, which is likely due to them being in completely different industries. One being in a tech industry which is generally considered more volatile, while the other one is in a stable industry where we don't expect big increases or decreases in demand through market cycles.

Figure 1: Asset Correlations

```
In [29]: AAPL_KO_correl = AAPL_returns.corr(KO_returns)
         AAPL_SP500_correl = AAPL_returns.corr(SP500_returns)
         KO_SP500_correl = KO_returns.corr(SP500_returns)
         print(AAPL_KO_correl)
         print(AAPL_SP500_correl)
         print(KO_SP500_correl)

         0.2407511736593989
         0.48616235166457633
         0.570525923108984
```

We continued by running a linear regression on the data in order to get the α and β, presented in the figure below.

Figure 2: Alpha and Beta of stocks

```
In [34]: lr.fit(sp, a)
         print(lr.intercept_, lr.coef_)
         lr.fit(sp, k)
         print(lr.intercept_, lr.coef_)

         [0.000287] [[1.22048152]]
         [0.00030763] [[0.73913927]]
```

In the CAPM model β represents the measurement of systemic risk in a particular asset/portfolio. Our results show us that Apple has a β of 1.22, while Coca-cola has a β of 0.74. If we were to assume that the risk-free rate is 3% and the market expected return is 8%, we can calculate the expected returns of the two stocks with the CAPM formula:

$$R_i = R_f + \beta_i * (R_m - R_f)$$

This gives us a hypothetical expected return for Apple stock of 9.1% and an expected return of 6.7% for Coca-Cola.

3 Machine learning in practice

3.1 High level description of FinTech firm

The firm aims to issue peer-to-peer loans to individuals. Based on set parameters (loan term, income, length of employment, …) that a potential client inputs and the firms' machine learning model a loan is either issued or denied. The application itself is dockerized and can thus run on AWS, Google Cloud, Azure, and even a laptop. Dockerizing is a process that allows developers to pack, deploy, and run applications using Docker containers. Developers will include everything they need to run the application successfully in the container. Dockerizing is desirable as it allows for a simple deployment of application, it can be ran in seconds on the kernel, and provides a consistent work environment on any operating system or cloud service. (Developer Experience, 2022)

The firms' application includes four structural parts that are interacting with each other in order to provide the lending service. Those components are namely: Streamlit, Postgres, Jupyter, and FastAPI.

Streamlit provides the UI, it is basically the firms' website where clients input their data. It communicates with FastAPI which is the firms' credit risk model, that provides a credit score based on the clients input and gives us an estimate of how likely it is that the customer pays back the loan, and whether we issue it or not. The decision is then sent back to Streamlit and shown to the client.

Postgres is an open-source SQL database in which all of the client's data is stored and with time also the actual results of whether individual loans were paid back or not. This data is later used for analysis to improve the model. The machine learning model analysis is done in Jupyter, which provides us with a predictive model through a logistic regression on the data set from Postgres. In order to properly prepare the data for the analysis we have to turn it from linear to non-linear with the process of normalization. Additionally, tools are implemented such as ROC – which explains how good the model is, and Accuracy which compares the models' performance to that of just blindly lending to anyone.

With time as the inputs are adjusted or replaced with better ones, and more data is gathered, we can expect that the model will become better at predicting the risk of loss and thus provide the firm with better guesses of whether a loan should be issued and increase the firms' profitability.

3.2 Written description of Python Code

The lines of the code are explained with comments in the file "model_building.ipynb".

Bibliography

Developer Experience. (2022, May 6). *Dockerizing*. Retrieved from Developer Experience: https://developerexperience.io/practices/dockerizing