

BEEM062 Main Assignment Part B Brief

March 27, 2024

Abstract

Your main assignment (80%) must be handed in by Friday 26th April 2024. It consists of two equally weighted parts: part A) A 1,600 word essay; and part B) A technical task-based assignment. This document outlines your tasks for Part B, which on its own contributes 40% to your overall module grade. For Python based tasks, you **MUST** solve them using Jupyter Notebooks, with each line of code stored. You will submit your assignment as a set of documents with your notebooks stored separately (with the .ipynb extension so that they can be easily verified). You are welcome to store your own code on your own github repository or elsewhere, but the .ipynb files must be submitted. Since you submit one file, the best way to do this is to submit your part A document, and include a link to a repository (you can use your onedrive or any other) that contains all your files in one folder.

1 Front End Coding

(40 marks)

1.1 Wallet (15 marks)

Building further on the Bitcoin wallet constructed for beem061, construct a webpage that generates a private key, shows an associated address and QR code, but in addition add the following:

- Balance associated with the address
- Mnemonics

1.2 Web Page with Financial Literacy Game (20 marks)

Create a combination of .html .css and .js files that when placed in the same folder allow a browser to render a web page that present the user with a game that teaches the user something about financial literacy. Write down the concept from finance that you are teaching, and then create the game. There are many examples you can find online - any code you adapt, just acknowledge where it came from. The game should involve the user having to solve some kind of puzzle, generated randomly, with indication as to what the player scored (whether they got something right or wrong). Note that you will submit a separate .css file, so it is important that you include some attention to styling as well, while the JavaScript part will be dealing with the more dynamic elements of your page.

1.3 TypeScript (5 marks)

Explain in words the difference between JavaScript and TypeScript. Your answer must demonstrate an understanding of the pros and cons of both.

2 Time Series and GD with Python and NumPy

(15 marks)

From a Jupyter notebook in Python, extract one single time series of your choice from:

- the Quandl API, call it x_t
- the CryptoCompare API, call it y_t

Depending on availability, choose your own frequency and time period, making sure that you can analyse both time series by treating them each as a one dimensional array of the same length. Use the numpy library to find OLS estimates of α and β in the following specification, where e_t is an assumed white noise error:

$$y_t = \alpha + \beta x_t + e_t$$

- Analytically from standard OLS formulae
- By trial and error Machine Learning with a Gradient Descent (GD) Algorithm

One challenge you may encounter is obtaining convergence, depending on the underlying data. In words, describe why convergence to a Loss Function minimum may be difficult to obtain in practice, and how this might be overcome.

3 AI and the Blockchain (15 marks)

In 500 words for each bullet point, explain each of the following (be sure to reference where appropriate):

- Explain what ChatGPT is, and what effects it may have in the future.
- To what extent do you think Blockchain technology could provide a solution to any systemic problems coming from AI?

4 Time Series Forecasting

(30 marks)

In this section you are set an open ended mission of using everything you have learned to create a Python notebook that produces a time series forecast of a financial time series of your choice. Pick one single financial time series that you are interested in (eg the Bitcoin price in Sterling), and conduct your analysis in stages: In the first stage, you must describe why you have chosen the series, and what factors you think affect it. Where relevant, describe any general trends, specific events, and what variables you think may drive it. Distinguish between variables that you should be able to obtain, versus those you cannot. In the second stage, conduct your analysis. There are many potential difficulties to overcome, including extracting the relevant time series (you could use the quandl or cryptocompare API calls already used, or any other).

Your analysis should reduce a larger model with many variables (more than 10) down to a smaller model by applying a Machine Learning Technique like L1/L2 Regularisation. Scikit learn has examples of implementing the combination of L1 and L2 called elastic nets. This is the recommended approach, but you can also choose other machine learning approaches as long as they are properly documented. Include an assessment of how good your implementation is (confusion matrix, AUC-ROC).