SSMIF Quant Coding Challenge

Spring 2024

Contents

Introduction

Hello and welcome to the SSMIF Quant Coding Challenge! Completion of this coding challenge is required for acceptance to the interview round. **You are required to answer at least 2 out of the 5 questions detailed in this document.** We greatly emphasize that quality is better than quantity, so please concentrate on the quality of your solutions.

0.1 Challenge Structure

There are 5 optional challenges to choose from; 4 of these challenges relate to a specific team on the Quant side of the SSMIF. The challenge you complete will increase your likelihood of being considered for that specific team among other applicants. You may also complete and submit multiple of the optional challenges, which will reflect accordingly when considering your application. **Any available materials for these challenges (data, submission templates, etc.) will be provided in the zip file attached to the email.**

0.2 Rules

You may use generative AI models such as ChatGPT in your code; however, **any code written by ChatGPT must be cited by a 1-line comment above it.** This applies to all other sources, online or not, including forums like Stack Overflow, and sites similar to it. **Note that collaborating with other individuals is forbidden and will only hurt your chances of moving on to the next round.**

0.3 Submission

Your submission must be written in Python and your files for the questions must be named and formatted as instructed in each of the following questions. Please use the following Google Form to submit your files: https://forms.gle/oqDK4WDDhFFvbwnXA.

All applications should be submitted before the deadline, which is **Friday, March 8th, 2024 at 8:00AM EST.** We will not consider any late submissions.

Any additional questions regarding the challenge may be forwarded to Head of Quant, Swathi Venkateswaran, at svenka12@stevens.edu. Good luck!

Question 1 - Clue

1.4 Background

In the classic board game "Clue", players must deduce which character, weapon, and room were involved in a murder by making logical deductions based on clues provided throughout the game. Your task is to develop a Python program that simulates a simplified version of the Clue game and helps players solve the mystery.

For this problem, you will be provided a .py file that you will implement to represent this game. Within the file are functions you are required to complete to ensure the file works as intended for testing purposes. These are denoted by `#write your implementation below` in each of the required functions. You may also write helper functions if you wish to.

A player's suggestion would look like this:
`suggestion = {"character": "Miss Scarlet" , "weapon" : "Revolver", "room": "Kitchen"}`

A player's response to suggestion would look like this:
`response = {"Player 2" : "Lead Pipe", "Player 3": None}`

A player's scoring sheet would look like this:
`scoring_sheet = {"Miss Scarlet" : True, "Lead Pipe": False, "Revolver": False,...}` for all elements in the characters, weapons, and rooms.

1.5 Functions to Implement

You are required to implement the following functions in the provided .py file:

- `make_solution`
  - Randomly selects the solution by choosing one character, one weapon, and one room.
- `distribute_cards`
  - Distributes the remaining cards (excludes those in the solution) among the players
  - Returns a dictionary containing the cards distributed to each player
- `move_player`
  - Determines the number of spaces a player can move (randomly generated from 1 to number of rooms) and assigns them a room based on the number of spaces moved
- `make_suggestion`
  - Allows a player to make a suggestion, and it simulates other players' responses by checking if they have any matching cards.
  - Requires player to be in the room that is included in their suggestion
  - Returns the original suggestion and the response from other players
- `make_accusation`

  - Similar to `make_suggestion`, but if the player makes an incorrect accusation they are eliminated from the game
  - Requires player to be in the room that is included in their accusation
  - Returns `True` if accusation is correct and `False` if accusation is incorrect

- `update_scoring_sheet`
  - Updates the suggesting player's score sheet based on the suggestion and response received

Helper functions are encouraged. However, you should not alter the function headers of the defined functions above or the `__init__` function

1.6 Submission

Edit and complete the specified functions in the given .py file, aptly named `clue.py`, then rename the file as `firstname_lastname_clue.py` Submit this file to the Google Form provided in the introduction.

Question 2 - Algorithmic Trading

2.1 Background

For those specializing in algorithmic trading, it's essential to be able to calculate different financial metrics, implement a strategy, and understand performance. In the upcoming evaluation, your skills in researching, developing, and testing algorithmic trading strategies will be assessed. We'll focus on your creativity in strategy design and your proficiency in coding (you will not be judged directly on your profit/loss).

2.2 Part 1: Strategy Development

In the "Strategy Development" section, fetch data from yfinance and develop a trading strategy (other than a simple buy-and-hold approach) aimed at maximizing profits and minimizing risks, with an initial capital of $100,000. Your strategy's success will not only be measured by its profitability but also by your innovative approach and coding skills. You are encouraged to use machine learning, however it is not necessary. Ensure your code is executed before submission to facilitate easy review of the outputs.

Historical Timeframe: Jan 1st, 2022 to December 31st, 2024

Tickers(Minimum 3 tickers, however you can use however many): `AAPL, MSFT, KO, F, PEP, TSLA, JNJ, PFE`

2.3 Part 2: Performance Visualization

In the "Performance Visualization" section, you are required to create two visualizations:

- **Profit/Loss Visualization:** Plot the profit and loss of your strategy over the specified time frame, including markers to indicate when trades were executed.
- **Buy/Hold visualization:** Plot the profit and loss using all the tickers that you used within your strategy diversified evenly.

Aim to make your charts as detailed and visually appealing as possible to allow for an easy analysis of your strategy's performance over time.

In addition, to the charts, compare your strategy's sharpe ratio to the buy and hold's sharpe ratio.

2.4 Submission

You are required to submit a jupyter notebook with the following file name:
`firstname_lastname_algo_challenge.ipynb`

Attached is a file named `algo_challenge.ipynb` that you may use to format your code.

Please ensure that your code is well-documented and that your visualizations are clear and easy to understand.

Question 3 - Factor Model

3.1 Background

The overall goal of the Factor Model team is to effectively predict the behavior of the market over a defined period of time, and more specifically, the behavior of the 11 market sectors. This helps the fund optimize our funding allocations amongst the sectors (and stocks) that we invest in. This is done through the research and development of quantitative models. This is an exciting (and difficult) challenge, and we need bright analysts to help the fund in its goal to beat the S&P 500. This coding challenge will show us your ability to research, code, and explain models of this kind - all things you will be doing as a full-time analyst.

3.2 Challenge Breakdown

Part 1: Data Retrieval and Preprocessing

a. The first part of the challenge is data retrieval and preprocessing. You have been given a csv file with 10 years worth of monthly data for 10 different assets - pull this file however you see fit and apply any appropriate preprocessing on the data (look over the data carefully). This will act as your 'portfolio' for the sake of this challenge.

Part 2: Exploratory Data Analysis (EDA)

a. This is where you should be drawing any fundamental insights from the data that may help in using it later in your model(s). This is important in finding relationships in your data and finding out how useful certain data points may be. Adjust and change your data based on these findings as you see fit.

Part 3: Model Experimentation

a. For the first part of Part 3, we would like to see you implement a CAPM model on the given assets, and use it so that you produce a set of optimal weights for the 10 assets.

b. For the second part of Part 3, we would like you to research and implement your own model to produce the same results: a set of optimal weights for the 10 assets. Be creative here; you're free to use any machine learning, statistical, or traditional models you are interested in.

Part 4: Backtesting

a. For part 4, we would like you to show a backtest of how your model performs against the S&P 500 benchmark. We aren't necessarily looking for your model to outperform here; just that you understand how to properly backtest the results of a model. You can choose which model you would like to backtest: Your own model, or the CAPM model.

    i. *Bonus points for if you are also able to produce and compare metrics between the benchmark and this made-up portfolio of stocks (sharpe, volatility, etc.).

***Each section requires a short explanation for what you did and why. Commenting throughout your code is great and highly suggested/encouraged. The last section of the challenge, titled 'References', should include any resources you used and what you used them for... CHATGPT INCLUDED.**

3.3 Submission

You will be using and submitting the given Jupyter notebook. When submitting, your file should be renamed and titled: `firstname_lastname_fm.ipynb` Remember to read all parts carefully and to submit well-documented, easy to understand code. This is an important skill for quant analysts to have for when they are collaborating on a project.

Question 4 - Risk Management

4.1 Background

Risk management analysts are capable of understanding, using, and applying various statistical metrics to assess portfolio performance and strength. Risk metrics are crucial in managing a portfolio as they give a clear look into the potential downside of investments and allow for more informed decision-making. Some examples of preexisting metrics that are widely utilized across the industry are Beta, Maximum Drawdown, 95% Value at Risk(VaR), Volatility, Sharpe Ratio, and many more.

4.2 Part 1

A key component of joining the SSMIF is showing creativity for problem-solving and bringing a unique perspective to your respective team and the fund as a whole.

For the first part of this assessment, you will create your own unique statistical measurement that you believe captures risk efficiently. This risk metric can be based on preexisting metrics, or it can be entirely brand new. The purpose of this metric is to give clear, unbiased insight into the potential downside of a security and will highlight your creativity, finance knowledge, and statistics understanding. More information will be provided in the risk.ipynb file attached.

Additionally, comment on your code/functions explaining why this metric is an effective and usable risk metric.

4.3 Part 2

A crucial component of risk management is being able to understand and identify trends in data. The data in the risk-challenge CSV file consists of data points that were randomly generated by some function.

For the second part of this assessment, use the data from the risk-challenge CSV file and fit a Linear Model to predict the y values of the function. Identify the coefficients of the x values and briefly describe if the model has any degree of predictive power. Following this, you must complete a full hypothesis test to show how statistically significant the model's predictive power is using a confidence level $\alpha = 0.05$.

If you are able to complete the task without the use of any regression libraries, this would add additional points to your risk challenge score. However, using libraries would not negatively impact your score in any way.

Additionally, comment on your code/functions explaining the results of your model fitting and interpret the results of the hypothesis test.

4.4 General Notes

The only allowed libraries for the Risk Management Challenge are NumPy, Pandas, SKLearn, YFinance, or any related machine learning/regression libraries. Please note that the use of any

unauthorized libraries, packages, or APIs may remove your code from consideration into further rounds.

4.5 Submission

Please submit your solution in a Jupyter notebook(ipynb) and name your notebook as `first_last_risk.ipynb`.

Question 5 - Development

5.1 Background

The development team is responsible for maintaining SSMIF's internal web applications and tools. This coding challenge will assess your ability to develop a shopping catalog app using Flask and Next.js. The app will have a set of required features and a set of optional features that will help you stand out from other applicants. This challenge is designed to allow you to showcase your creativity, attention to detail, and proficiency with the frameworks we use.

5.2 Part 1: Backend

The backend will maintain a catalog of items that are currently available, serve product names & prices to the front end, and host api endpoints that will be used to add/remove items in the catalog.

5.2.1 Requirements

- Must have endpoints for adding items, removing items, and getting the current catalog (product names & prices)
- Must be build using Python & Flask
- Code must be legible and well commented

5.3 Part 2: Frontend

The frontend will display the catalog of products and their corresponding prices, appear appealing to potential customers, and provide an intuitive interface for adding/removing items from the catalog with your api/backend.

5.3.1 Requirements

- Front end should make full use of any additional features that were added to the Backend
- Must use TailwindCSS for styling (not just plain CSS)
- Code must be legible and well commented (where applicable)

5.4 Additional Information

Please refer to the README.md file provided in the Question 5 folder for further instructions and details about the challenge. We have also provided a starter code for the backend and frontend in the Question 5 folder.

5.5 Submission

You must submit both parts within a `.zip` file named `first_last_dev.zip`. You should make a new `README.md` with your name, details about any additional features, and instructions on how to install & run it. Please do not include the `/.next`, `/node_modules` or `/.venv` directories. Make sure to update your python package requirements with `pip freeze > requirements.txt` before submitting, so that the reviewer can easily install the required packages.