SSMIF Quant Coding Challenge

The Quant Team

Spring, 2021

1 Introduction

The following document will outline the questions which make up our coding challenge. Completion of the below instructions is required for acceptance to the interview round. You are required to answer at least 2 questions. The questions in Part 1 are required. You may choose one or more questions to answer in Part 2.

1.1 Submission

Your submission should be written in python and consist of .py / .ipynb files in a zipped folder of name: last_first_quant21.zip. Alternatively you can create a hosted Git repository (GitHub, GitLab, etc) and share the link with us. You will have a period of one week to complete these questions (3/8/2021 - 3/13/2020). Please email your submissions to eluvishi@stevens.edu (subject line: SSMIF QUANT 2021 SUBMISSION) before 11:59pm 03/13/2021.

2 Part 1: Required Questions

The following questions are required. You must submit answers to these question for you application to be considered.

2.1 Basket Portfolio of Stocks

For the first question you will be creating a class, called Portfolio, to model different financial concepts. **You can only use numpy, pandas, and the built-in standard python libraries.** Your portfolio will consist of an arbitrary number of stocks, listed in a dictionary with their corresponding shares, which should be provided in the constructor.

Ex: {“AAPL”: 50, “GME”: 150, “TSLA”: 5, “AAL”: 200, “AMZN”: 1}

Additionally, your constructor should take a start and end date (both should be in the past) of the lifetime of your portfolio. These dates should both be datetime objects. Finally, your constructor should take a string ticker of your portfolio’s benchmark index. Ex: ”VOO”.

You will implement the following methods in your Portfolio class. Note: please use the exact method names provided:

1. averageDailyReturn()
   (a) This method should return the average daily percent return of your portfolio (one number), calculated iteratively in *your* code. Describe your implementation reasoning in the docstring.
   (b) Return type: float
2. volatility()
   (a) Calculate your portfolio’s volatility over the lifespan of the portfolio.

   (b) Return type: float

3. riskRatio()
   (a) Calculate the volatility ratio of your portfolio compared to the volatility of the benchmark over the lifetime of the portfolio.
   (b) Return type: float

4. marginalVolatility(ticker: str, shares: int)
   (a) Calculate the difference in volatility of your portfolio if you were to add the specified number of shares of the ticker passed into the method.
   (b) Return type: float

5. maxDrawDown()
   (a) Calculate the maximum drawdown of your portfolio (the largest drop from peak to trough in the lifespan of your portfolio).
   (b) If there are multiple drawdowns, return the biggest one.
   (c) Investopedia Link

$$MDD = \frac{trough\ value - peak\ value}{peak\ value}$$

   (d) Return type: float

3 Part 2: Select Questions

You are required to complete 1 of the following questions. You may complete more than 1 if time permits, though you are not required to. What you submit is a reflection of your ability, what you are proficient in. If you're not comfortable with answering one, it's totally ok to skip it. **Quality over quantity.**

3.1 Divisible

Write a function divisible(String s, int x) that takes a string s as a parameter and returns a list of all of the unique numbers contained in that string that are divisible by non-negative integer x, but do not include integer x within the number itself. For example, 24 and 16 are both divisible by 4, but 24 includes the number 4 so it is not counted. If there are no such numbers found in the string, the function will return an empty list. Consider all string inputs, even those containing numbers and letters. Letters will act as "blockers" between numbers. For example, for the string "123a4", the numbers 123 and 4 should be checked but 1234 should not be. You may make helper functions if you see them as necessary. See the test cases below

Input — Output	
divisible("tothemoon", 1)	[ ]
divisible("a465839485739b102988c30jklol4", 7)	[0, 98, 1029, 3948, 6583948, 9485, 658, 58394]
divisible("a1234567890ef", 5)	[0, 90, 890, 7890, 67890]
divisible("1782931", 1894792)	[ ]
divisible("Jennychangeyournumber8675309", 0)	[ ]

Table 1: Example Inputs and Outputs

3.2 Portfolio Dashboard

This question is designed to show off your web programming skills! We want you to make a portfolio dashboard web application. The main features should be the ability to add and remove stocks, with persistence on page reloads, and viewing the current stock price. The frontend should be written in a javascript framework (vue, react, or svelte is preferred; we use react.js in SSMIF), and the backend should be written in python (we use flask). You can feel free to use whatever you prefer (tailwind, chakra, etc, or just vanilla css; we use bootstrap). You can make the app as simple or as complicated as you want, maybe saving holdings or having multiple profile accounts would be a good idea. Show off what you can do.

3.3 Cloud Deployment

In this question, you will be spinning up an ec2 micro instance on AWS to deploy a web application. You can either use this pre-made todo app (written in flask), or use the one your portfolio dashboard. The instance should be completely free under the free tier, as long as you get rid of it when you are done. The app should be available through http on port 80. You can add https for bonus points. We should also be able to ping your instance, and there should not be any other ports open (we will check). Take a screenshot of your ec2 instance page with the ip address of the instance showing and the user's account id visible. If you want to use ecs or elastic beanstalk instead, you can do that too.

3.4 Machine Learning

This question requires a little ML / NLP knowledge. The goal is to predict if the overall market will rise or fall based off of news headlines. The dataset you will use for training and testing contains headlines from 25 different topics, gathered over the range 2009-07-22 - 2013-07-11, from the world news subreddit. The label column represents if the dow jones index closed up or down on that day. The data we will use for testing will be in the same format, from a random time period. You can use a regression, random forest, BERT, LSTM, or whatever else you want in order to build your model. We are providing you with a script to assist in data cleaning. You should use the data in the format provided to train/test your model. We're going for accuracy (and reasonable speed). Our simple solution has around 80% accuracy. You can write it in a jupyter notebook/Google colab notebook or in a python script, but a notebook is preferred. Look into using google colab or similar if you want to train your model in the cloud (here's the notebook on collab: colab; **if using colab, make sure to make a copy before you start editing, otherwise your work won't be saved**). We provided a notebook to help you get started. When submitting, please list your dependencies, either in a requirements.txt file (for pip), a Pipfile (for pipenv), or in a conda environment.yml file.

4 Before You Submit

All scripts should compile and run on the first try. Remember to comment your code and optimize your solutions where possible. Only include your source files and dependency lists in your source code. We don't want to see any "node_modules" or "__pycache__" folders. The final source code should be less than 1 mb. We wish you the best of luck!

Please reach out to Eden Luvishis, Head of Quant, with any questions.