Pre-Interview Coding Challenge

Challenge 1: Backend – Identify outliers of timeseries data (Stock price)

Objective

Our company, as well as being a global fintech company, operates the British stock exchange while other companies operate other exchanges around the world.
Part of the data that the company needs to provide to its customers is price index information on the various exchanges around the world. In order to ensure that the data is of a high quality, we need to spot possible errors or "outliers" in the price data provided. For this challenge we have provided a zip file with artificial price data for a number of different global "Exchanges". For each stock exchange, select the specified number of stock files, and for each file provided, return the list of outliers present in that specific file.

Requirements

Your solution should utilize 2 APIs/Functions

•1st API/Function that, for each file provided, returns exactly 30 consecutive data points starting from a random timestamp within the file. Which means the timestamp cannot be from the last 29 data points.

•2nd API/function that gets the output from 1st one as a feed and returns the list of outliers.

Data & Inputs

Sample data is provided in .csv format. Each file has

•Stock-ID (Ticker), Timestamp (dd-mm-yyyy), stock price value.

Input parameter to your solution: *T*he recommended number of files to be sampled for each Stock Exchange. Possible input values are *1* or *2*. If there aren't enough files present for a given exchange, process whatever number of files are present even if it is lower. E.g., input is 2 but only 1 file is present, so you process 1 file.

Outlier definition: Any datapoint that is over 2 standard deviations beyond the mean of the 30 sampled data points.

Output Format

Create one .csv file for each file processed. Each .csv should have below columns on each row (1 for each outlier found). Timestamp & stock prices have same format as input file.

Stock-ID, Timestamp, actual stock price at that timestamp, mean of 30 data points, actual stock price – mean, % deviation over and above the threshold.

Error Handling

The application should gracefully handle exceptions, such as no files, empty files, invalid CSV format etc., Please free to include as much exception handling as possible. It provides insights into your ability to anticipate what can go wrong.

Documentation

Include a README file explaining how to set up and run your application.

Optional Enhancements

Feel free to add enhancements that could improve the extensibility/maintainability for future enhancement, user experience etc., Some suggestions include:

- additional functionality or checks
- more insights added in the report you generate
- Optimizations for performance and scalability.

Submission Instructions & FAQ

How to submit your challenge

Please submit your code via a publicly accessible GitHub repository created specifically for this challenge.

- It is expected that you will know how to create a repository on GitHub and check your code into it. Do not share your repository with anybody else other than the reviewer. You should ensure your repository is read-only to everybody except yourself, this is so nobody can interfere with your submission after you have completed it.
- The code for the challenge should be the only code in the repository except for the README file, explaining how to set up and run your application, providing any access related information, including language versions used.
- Estimated Time spent on the challenge should be ~2hours, if you do not finish , include any additional information, extra thoughts or things you ran out of time for in the README file.
- Ensure your code is well-commented and follows good coding practices.

Frequently Asked Questions

Q: Can I use external libraries or frameworks?

A: Yes, you can use any libraries or frameworks you find appropriate.

Q: Is it required to deploy the application online?

A: No, it's not required. However, if you choose to do so, please include access details in your README.

Q: What if I have questions during the challenge?

A: Document any assumptions you make in your README file.

Q: Are there any specific coding standards or practices I should follow?

A: While there are no enforced coding standards, your code should be readable, well-organized, and demonstrate good software development practices.

Q: Is it okay to use code snippets or libraries from the internet?

A: Yes, but ensure you understand and can explain any code you use. Also, respect code licensing and give credit where due.

Q: What will this challenge be assessing?

A: We are looking at your coding skills, problem-solving abilities, creativity, and how you approach and structure a project.

Q: Can I use online resources for help?

A: Yes, you can use online resources but make sure you understand and can explain any code you use. Plagiarism will disqualify your submission.

Q. Can I use AI generated code to complete the challenge?

A. No, the challenges are designed to test your skills and understanding of the problem presented, if we detect AI generated solutions it will be treated the same as we would treat "Plagiarism".

Q: For the UI related exercise, in case of time crunch should I prioritize look and feel or the functionality?

A: Functionality has higher precedence than look and feel, in case you have to compromise on one of them.

Q: Can I use containers?

A: Not required but additional credit may be given, if you provide a Dockerfile in the repo, that builds a container that contains all of the all of the code you produce for the challenge, the data needed to run it, and all of the software required to execute your solution. We should be able to check out your repo and run docker build and docker run to execute your solution. Make sure your Readme.md file contains instructions on how to both build and run the solution. An additional credit may be gained by uploading the container to dockerhub.com, again if you chose to do this then you should note how tomrun it from dockerhub.com in your Readme.md file.