

Time commitment

You can spend as much time as you want on the exercise, but you are absolutely free to put a time limit (we recommend 3 hours) and tell us about it. We respect the fact that you might have other hiring processes and that you don't have a lot of time.

Deliverable

Please make sure you include the following items:

- **[README.md] (<http://README.md>)**
 - General approach
 - Technical choices
 - How did you evaluate the solution?
 - How much time did you spend on the exercise?
 - Remaining todos
 - ... and any other item you think should appear in a README.

The code, its comments and the README should all be in English. Don't forget to proofread yourself (tip: feel free to use Grammarly)!

Functional specifications

The aim of this assessment is to challenge you on a technical basis, involving machine learning, programming and research skills.

➡ ➡ Goal

Your goal is to develop an algorithm that detects anomalies in trading profiles. For such a purpose, your solution will return "anomaly" for a trade whose data is the outlier, or "regular" for normal trading data.

➡ ➡ How?

For this assessment, we provide a dataset containing the trading profiles of 4 FX traders (named "Traders"). This database is an excel file. The first column is the transaction number, the second column is the trading volume in USD, the third column is the trading price for the currency pair, the fourth column is the market price, the sixth column is the PL (profit/loss)

The next three columns are transaction time, transaction date, and Validation time. The last column specifies whether the transaction is Regular (equal to 0) or Anomaly (equal to 1).

Technical specifications

- We can read most programming languages. Just so you know, we mostly use Python and Angular/TypeScript at Upskills. Justify the language you chose in your README.
- You can use any library or framework. Justify all your important choices in your README.
- You can challenge *anything* that we've currently implemented.

How we review

The most important thing is that **you should feel proud of your work**. If you don't have time to finish, it is better to polish what you've already done than start something and risk not doing it well. You can always include a list of remaining todos in the README - we will take it into account when we review. Focus on the things that let your skills shine!

We will have a look at the following aspects:

- Correctness: does your solution work?
- Performance: what is the quality of your solution? How fast is the process?
- Maintainability: does your code include the necessary best practices to make it easy for a team to build on top of it? Have you fixed most of the issues with our implementation?
- Code style: how easy is it to read and comprehend your code? Are you following the language's code style (e.g. PEP8 for Python)?

Those aspects are **critical** to us:

- Writing skills: make sure to proofread yourself and remove all typos, orthographical mistakes, etc.
- Important technical tradeoffs need to be justified.


Our tips

If you're using Python:

- Consider using poetry to handle dependencies
- Consider using a linter (e.g. flake8) and an autoformatter (e.g. black)

Need help?

Any question? Reach out to us! We'd love to help you!

 Good luck & happy coding!