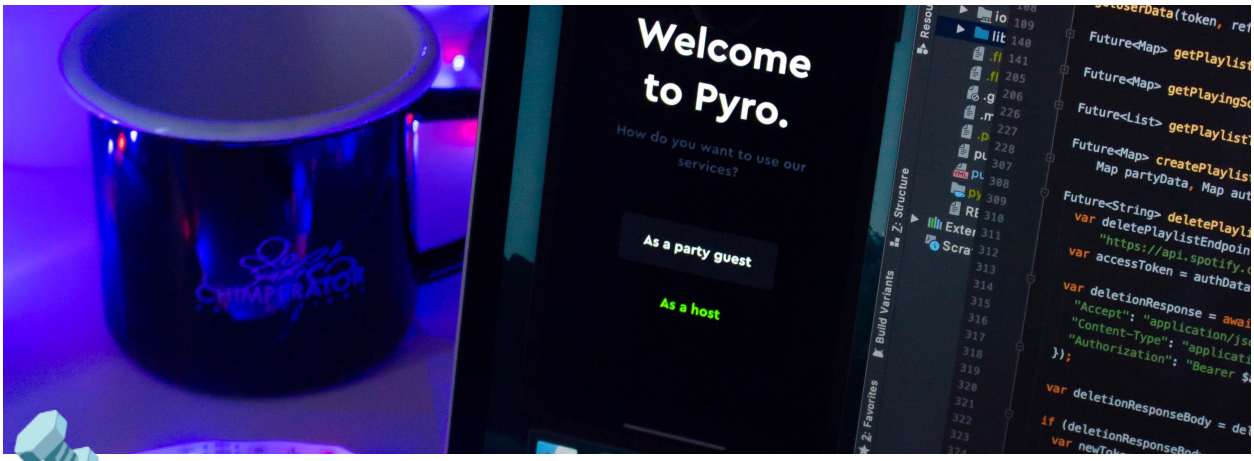

Assessment

Introduction

You have exactly 72 hours for this assessment after it has been sent to you!

This assessment evaluates the performance of a trading research role, focusing on the optimization of trading parameters for a model-driven trading strategy.

Datasets

Dataset name	Description
transaction_log-201793-ETH-1_1_2019, 10_00_00 PM_8_31_2022, 10_10_08 AM copy.csv	Shows the trades that have been made of an active trading bot. It resembles how many trades are made and what kind of trades. You could get your inspiration for determining the τ1 and τ2 from this dataset. If you want to use this. Do not spend too much time on this dataset!
eth-perp_train.csv	Contains the one-minute based price of Ethereum Perpetual starting 2021-01-01 till 2022-08-03. It is up to you how you use this dataframe.
eth-perp_test.csv	You will not get this dataframe. This dataframe contains the the one-minute based price of Ethereum Perpetual starting 2022-08-04 till 2023-04-09.

Part I: Model Volatility (σ)

In this section, we will delve into your understanding and handling of market volatility (σ), a fundamental factor in successful trading. Your ability to accurately estimate and adapt to changing levels of volatility is paramount to making informed trading decisions. We will comprehensively assess your knowledge and approach in this context.

Understanding τ1 and τ2:

τ1 and τ2 are critical time intervals that play a pivotal role in your trading strategy.

- **τ1:** This represents the first time interval and serves as the basis for estimating σ (volatility). You can see this as how many signals you use from the past. It's the timeframe during which you gather historical data and calculate the volatility that will inform your trading decisions. Explain how you determine your τ1. Examples of τ1 can be 10 minutes, 1 hour, 1 day 1 week or 1 month.
- **τ2:** On the other hand, τ2 represents the time interval for forecasting σ for the future. It's the period during which you anticipate market conditions and volatility. Clarify your process for selecting τ2. What considerations guide your choice, and how does t2 impact your trading strategy?

> 💡 The dataset 'transaction_log-201793-ETH-1_1_2019, 10_00_00 PM_8_31_2022, 10_10_08 AM copy' could be used to help you make decisions for τ1 and τ2. How? That is up to you. This file has nothing to do with the trading bot used here but you can assume similar behavior regarding **trade frequency**.

Modeling σ:

We want to understand your approach to modeling σ, the rationale behind your choice of time intervals (τ1 and τ2), and your proficiency in utilizing data to make informed decisions regarding σ, a pivotal factor in optimizing a trading strategy.

Part II: Determine parameters based on σ

In this section, we delve into the crucial aspect of determining trading parameters (υ, ψ, and χ) based on market volatility (σ). Your ability to adjust these parameters dynamically in response to changes in σ is a key determinant of your trading success. Here's how you should approach this:

Parameters	Description	Example
υ (upsilon)	percentage adjustment to buy	Current price is $1000, υ = 1.5%: Insert buy order at $985. A static υ means you always use the same percentage below the average buy price to sell.
ψ (psi)	percentage adjustment above the average buy price to sell	Current price is $1000, ψ = 1.0%: Insert sell order at $1010. A static ψ means you always use the same percentage above the average buy price to sell.
χ (Chi)	percentage of account balance that will be entered	Current price is $1000, capital is $10000, υ=1.5%, χ = 1.0%, then you insert buy order at $985 with amount $100. This means that the volume

	as amount for a buy order	will be approx 0.1015ETH. A static χ means you always use the same amount (in this example $100) to enter a buy order.

Adjusting Parameters Based on σ:

The crux of this section is to determine how you adjust these parameters dynamically based on market volatility (σ). Your goal is to maximize profitability, so creative and adaptive approaches are encouraged. Here's how you should think about it:

1. **υ (upsilon) Adjustment:** Consider how changes in σ affect your buying strategy. Should you increase υ in times of high volatility to capture potential price swings, or should you decrease it to minimize risk during uncertain periods? Think about specific scenarios where different υ values might be advantageous, and be prepared to explain your reasoning.
2. **ψ (psi) Adjustment:** Similar to υ, think about how changes in σ should influence your selling strategy. Should you adjust ψ upwards during high volatility to capitalize on price spikes, or is it better to lower it to secure profits more conservatively during uncertain times? Provide examples and rationale for your decisions.
3. **χ (Chi) Adjustment:** Chi introduces an additional layer of complexity by considering account balance (capital). Explore innovative ways to adjust χ based on σ. How can you balance capital preservation and profit maximization? Consider whether a dynamic χ based on σ can help you optimise the amount you insert in the buy orders and, subsequently, your returns.

💡 Note, do not make certain parameters such as υ and ψ too small, since in real life you also need to pay fees. You could refer to Deribit to backup your values if you do want to go for the as small as possible approach. Same for the parameter χ, if chosen too large you often reaches your account limit, we prefer not to have that often to happen.

💡 There's no one-size-fits-all answer. Your ability to creatively adjust these parameters based on σ and your willingness to experiment with unconventional ideas can be a significant factor in optimizing trading success. Ultimately, this section is about your ability to make informed and adaptive decisions that lead to maximizing profits in various market conditions, so if you find a way without using σ that is fine as well.

Part III: Optimize your modeling results using the predefined trading bot

For this section, you will need to use the files

```
bot.py
evaluate_performance.py
```

In this section of the assessment, you are tasked with evaluating your modeling results by implementing and optimizing the given trading bot. This bot has predefined parameters, and your challenge is to maximize its return while adhering to these constraints. Below are the fundamentals of the trading bot and key points to consider:

Bot Fundamentals:

1. **Fixed Parameters:** You should not modify certain fixed parameters or functions that changes the fundamentals of the trading bot in the code, such as capital, max_positions, and other very obvious parameters and functions.
2. **Buying Behavior:** When the bot's position is 0, it immediately buys at the current price. After that it buys or sells based on a threshold that you will need to determine.
3. **Buying and Selling Strategy:** Your main task is to determine when to buy and sell based on the given parameters. You can continue buying until the position limit is reached, but as soon as the price reaches a predefined threshold, you should sell everything. The dynamic of the trading bot is implemented already.

Running evaluate_performance.py should print the return (and the amount of ETH you virtually gained).

Your Job:

Your primary objective is to maximize the bot's return above the baseline by optimizing the parameters. In the following table you see some examples of outputs with static parameters:

υ (upsilon)	**ψ (psi)**	**χ (Chi)**	**Return (%) on eth-perp_train.csv**	**Return (%) on eth-perp_test.csv**	**Notes**
0.002	0.025	0.02	182.15%	299.69%	Default parameter set
0.01	0.05	0.03	147.81%	322.43%	
0.03	0.04	0.1	253.98%	360.42%	

You can make changes to the code, but you must ensure that it still achieves the returns using the default parameters to maintain consistency.

Important Notes:

- We assume you can always buy and sell at the desired volume and the same price.
- The order book is ignored for reducing complexity, and only the current price is considered.
- The bot should achieve exactly a 182.15%% return using the default parameters.

> 💡 Before submitting this assessment, use the default parameters again to test whether you get 182.15%% or not to make sure you did not twist the trading bot's fundamentals.

Evaluation:

After you submitted this assessment we will run your parameters on a dataset named df_test which you will not have access to.

> 💡 Make sure the code can be run by passing a new unseen dataset by just running:

```
if __name__ == '__main__':
    bot = Bot(df_test)
    bot.run()
```

Part IV: Documentation

In the final part of this assessment, compile your thoughts and results into a compact file. Include pictures or graphs if you feel necessary. Document your approach, the important changes you made to the code, the reasoning behind those changes, and the results you achieved. Clearly convey how you optimized the bot's return and any strategies you employed to achieve this goal.

Callout Option

You have the right but not the obligation to request assistance once during this assessment process. Kindly send a whatsapp message to +31624786868 indicating three distinct time slots when you are available. Following your request, you will have a maximum of 15 minutes to pose up to three questions. Please note that there is no guarantee that your questions will be answered.

The decision to utilize this option, as well as the nature of the questions asked if it is exercised, will be factored into the overall evaluation. We will not disclose whether the utilization of this option is considered an additional point in your assessment.

Evaluation

Evaluation Criteria:

You will be evaluated based on several criteria:

- **Deadline:** You have 72 hours to complete this assessment. Finishing on time is crucial.

- **Coding Style:** Maintain a clean and interpretable codebase. Use clear variable names and structure your code and files logically.
- **Approach:** Describe your overall approach to the problem. How are you planning to optimize the bot's performance? What strategies do you intend to implement?
- **Reasoning for Choices:** Document the reasoning behind your choices, especially when making modifications to the code or changing parameters. Explain why you believe your changes will improve the bot's performance.
- **Documentation and Storytelling:** Keep clear and concise documentation of your work. Narrate your journey in optimising the bot, including your thought process, experiments, and results.
- **Results:** How well did you your bot do on both the train and test data. Furthermore how did you perform against other candidates. Our researcher will do this assessment as well but in 2 days, try to beat him! 😇

Confidentiality

It is of paramount importance that the contents and details of this assessment remain strictly confidential. The information contained within this assessment is intended solely for the designated recipients and should not be shared, distributed, or disclosed to any unauthorized individuals or entities. The insights, data, and analysis presented here are proprietary and sensitive, and their protection is crucial to maintaining the integrity of our assessment process.

Submission

1. Push your code to GitHub without the datasets
2. Send both your documentation in pdf format with the GitHub link to contact@future-ed.dev

Good luck!