

“Information Valuation” Specification

1 Overview

You will trade four symbols, A, B, C, D , where A, B are ETFs on related baskets of stocks and C, D are uncorrelated equities. There are four hedge funds (conveniently named A, B, C, D) that respectively trade symbols A, B, C, D . An executive director in the corporate development division of each of these funds has decided to expand their firm’s business by disclosing their firm’s position for a fee. You will need to determine which of these sources of information might be useful in order to maximize trading expectancy. The directors of these funds are also holding auctions. Winning an auction gives you the ability to buy the hedge fund’s data a time step ahead of time.

Coming up with a successful strategy for this case might involve:

1. Performing statistical analysis to deduce the relationship between hedge fund positions and symbol returns.
2. Determining which information is worth the price and the most profitable way to trade on that information.
3. Examining the additional value of having the hedge funds’ positions one time step in advance and using that to profitably participate in the per-round auction.

2 Case Information

2.1 Summary

The case consists of 7 rounds, each of which consists of 252 timesteps. At each timestep of every round, you (or rather, your Python bot) will have the opportunity to decide which hedge fund positions you want to buy—the corresponding fees will be deducted from your total PNL. Afterwards, you’ll receive the data you purchased and will submit the quantities of each symbol you wish to hold for that round, subject to position limits. By default, the data you receive will correspond to the hedge fund’s ending position on the previous timestep.

At the end of each round, you’ll receive a .csv file that contains the price history of the symbols for that round as well as the hedge fund position data *for the time steps in which you purchased that information*. In particular, you begin the case with no data, so we strongly recommend you purchase position information for most or all of the first round to have something to work with (in fact, that’s the default behaviour of the skeleton bot we provide you).

Additionally, for each round, there will be an auction where participants can submit a sealed bid to each of the different hedge funds. For every fund, the top 8 bids will

be considered winners of the auction and will pay out at the price of the 9th bid. The payout price will be published at the end of the auction. In the following round, whenever an auction winner for hedge fund *X* purchases the data from fund *X*, the data will correspond to the hedge funds desired ending position for the current timestep instead of the ending position from the previous timestep. Note that you still need to pay the per-timestep fee to receive the data even if you win the auction—the auction just allows you to look at this data one timestep in advance.

2.2 Pricing

The per-time-step pricing for the hedge fund data is \$200:

Fund	Cost / timestep (USD)
A	100
B	100
C	100
D	100

2.3 Input

When we test your strategy, we will grade a `Trader` object on the timesteps for the given round. The starter code we provide initializes the class as follows:

```
class Trader:
    def __init__(self):
        """
        The Trader class is instantiated once for each round.
        """

        # You may want to keep track of history.
        self.symbols = ['A', 'B', 'C', 'D']
        self.px_history = {sym: [] for sym in self.symbols}
        self.pos_history = {sym: [] for sym in self.symbols}

        self.pnl_this_round = 0

        # Keep in mind these params.
        self.TIMESTEPS_PER_ROUND = 252
        self.POS_LIMIT_BY_SYMBOL = 600_000
        self.POS_LIMIT_TOTAL = 1_000_000

        # You may want your bot to remember if it won the auction.
        self.won_auctions = {
            sym: False for sym in self.symbols
        }
```

You should feel free to make changes as necessary. We will only interact with your bot via the following three methods.

2.4 Output

Your trader bot should contain the following methods:

- `WonAuctions(wins)`

- `time`: a dictionary mapping each symbol to a Boolean.

This function will also be called once per round and will tell your bot which auctions you won. This function need not return anything.

- `BuyInfo(time, prices)`: controls how much information you buy at each timestep in a given round.

- `time`: an integer argument indicating the current timestep
- `stock prices`: a dictionary argument with keys in the symbols and float values which will be taken from the information pricing data for that round.
- `fund prices`: a dictionary argument with keys in the symbols and float values which will be the price of buying hedge fund data. (This will always be \$200 for each symbol.)

This will output a dictionary of Boolean values in each symbol "A", "B", "C", "D" corresponding to whether or not you purchased information for each at a given timestep.

- `MakeTrades(time, info, prices)`: gives how much of each symbol you will choose to purchase in given round.

- `time`: an integer argument indicating the current timestep.
- `fund info`: a dictionary argument with keys in the symbols corresponding to either the position information for each symbols at a given timestep if you have purchased it or `None` otherwise.
- `stock prices`: a dictionary argument with keys in the symbols and float values which will be taken from the price data in the symbols for that round.

This will output a dictionary of floats in each symbol "A", "B", "C", "D" corresponding to how much of each symbol you choose to trade at a given timestep.

2.5 Position Limits

At each timestep, you are allowed to hold a position of maximum 1,000,000 USD across all four instruments. In order to manage risk, the maximum position you can hold in any particular instrument is capped at 600,000 USD. You may hold short (negative) or

long (positive) positions as you see fit—these limits apply to the sum of the absolute values of your signed positions.

If your desired position for a particular timestep exceeds any of these limits, it will be ignored, and you will instead hold zero position for that timestep.

2.6 Structure and Scoring

There will be seven rounds of submissions. The amount of time you will have for each round will be approximately: 15, 30, 25, 15, 15, 15, and 15 minutes respectively.

For each round of submissions, your code will be run and evaluated on 252 consecutive timesteps. Each time step will involve a single call to your bot's `BuyInfo` method followed by a single call to your bot's `MakeTrades` method. We will then update your PNL by:

1. Subtracting the total cost of hedge fund information you purchased.
2. Adding your profits/losses due to trades, calculated as the difference between the current price and the next time step's price, multiplied by your position.

Essentially, this means that your position is liquidated from one time step to the next, and you will always have a maximum of 1,000,000 USD to trade. There are no spreads, no other traders on the market, and no other elements of market micro structure you need to consider in this case.

Your final score will be the sum of your PNLs in the seven rounds. **Note: Due to the structure of the case, you should expect for your PNL per time step to be on the order of hundreds or low thousands.** The symbol and hedge fund price characteristics won't change from round to round—the data is generated consecutively and artificially partitioned into rounds. The amount you pay in the auction will be subtracted from your PNL for the corresponding round.

Having multiple rounds of submissions allows you to modify and improve your code in between rounds. You may also decide to update your auction bids after seeing the results of the previous round. Having seven rounds might seem overwhelming, but don't let that bother you. Rounds are spaced further apart in the beginning to allow you more time to iterate on your strategy as data comes in. It's totally viable for you to not significantly change your bot between submissions for the later rounds and use that time to perform more sophisticated statistical analysis that might lead to a more substantial revision at a later round.

2.7 Technical Requirements

Your submission will be run on our AWS server using Python 3.7.9 and will be able to import NumPy (v1.19.4), SciPy (v1.5.4), Pandas (v1.1.4), Scikit-learn (v0.23.2), and

Statsmodels (v0.12.1). You should not assume that any other packages are available, nor should you try to use any other packages. **Attempts at abusing our system will result in immediate disqualification. This includes any attempts at reading or writing files.**

Your code is expected to run quickly. If your code takes more than 10 seconds to execute the 252 timesteps, it may not run to completion. In such a situation, our system does its best to count all timesteps executed before timing out, but we make no guarantees.

You will be provided a testing file that will run some basic tests to help you debug. We cannot guarantee that these tests are comprehensive, though we've tried our best to make sure they are. As long as you follow the specifications above, you will be fine. Please be sure every dictionary you return contains every symbol; on the other hand, be aware that the dictionary of hedge info we provide you might *not* contain every symbol.

2.8 Submission

All of your bots and bids will be submitted via the following link:

<http://3.94.167.111:8080/submit>

You will be using the same Team ID as for HFT. Be careful to enter your username correctly. Once you have submitted your code, you can go to the website below to check that your code uploaded properly. Your latest bot/bid submission is the only one that will be considered. Rounds are zero-indexed, so the round number will be 0 at first, and the last round will be round 6. Do not share your Team ID with other teams.

You will use the same website to download your team's information (e.g. csvs containing hedge fund information and price data, and PNL and auction information). To do so, use this URL with the correct substitutions:

http://3.94.167.111:8080/results/<team_id>/round/<round_id>/

Once again, be sure to substitute your Team ID in correctly, and remember to zero-index the round number.

3 Auction

Before each round except the first, there will be an auction for extra information to be given in that round. The auctioned information is as follows: for a given symbol/hedge pair, at every given timestep, the hedge fund will sell to you their position data one timestep in advance. **Note that winning the auction does not affect the pricing of the data; in order to obtain the position data, you would still have to pay the corresponding fee at each timestep.**

This information may be helpful in improving predictions in the behavior of the symbols, and you may find it beneficial to take a long or short position in each of the individual symbols.

3.1 Auction Format

There will be one opportunities for obtaining advance-timestep information each round, for each symbol. At the start of each round, teams may submit their bids through the submission website at the same time that they submit their bot.

The payment structure is as follows: the teams with the 8 highest bids will each pay a price equal to the 9th highest bid. This amount will be subtracted from your PNL that round. Ties will be broken arbitrarily– **you may submit non-integer bids, but all bids must be non-negative!**

As an implementation detail, note that if you win an auction, your bot will have been fed the information for that symbol one time step ahead. This will be reflected in the CSV you receive with purchase information. This means that, if you won some auctions but not all, your hedge fund data will not be aligned. You are responsible for fixing this.