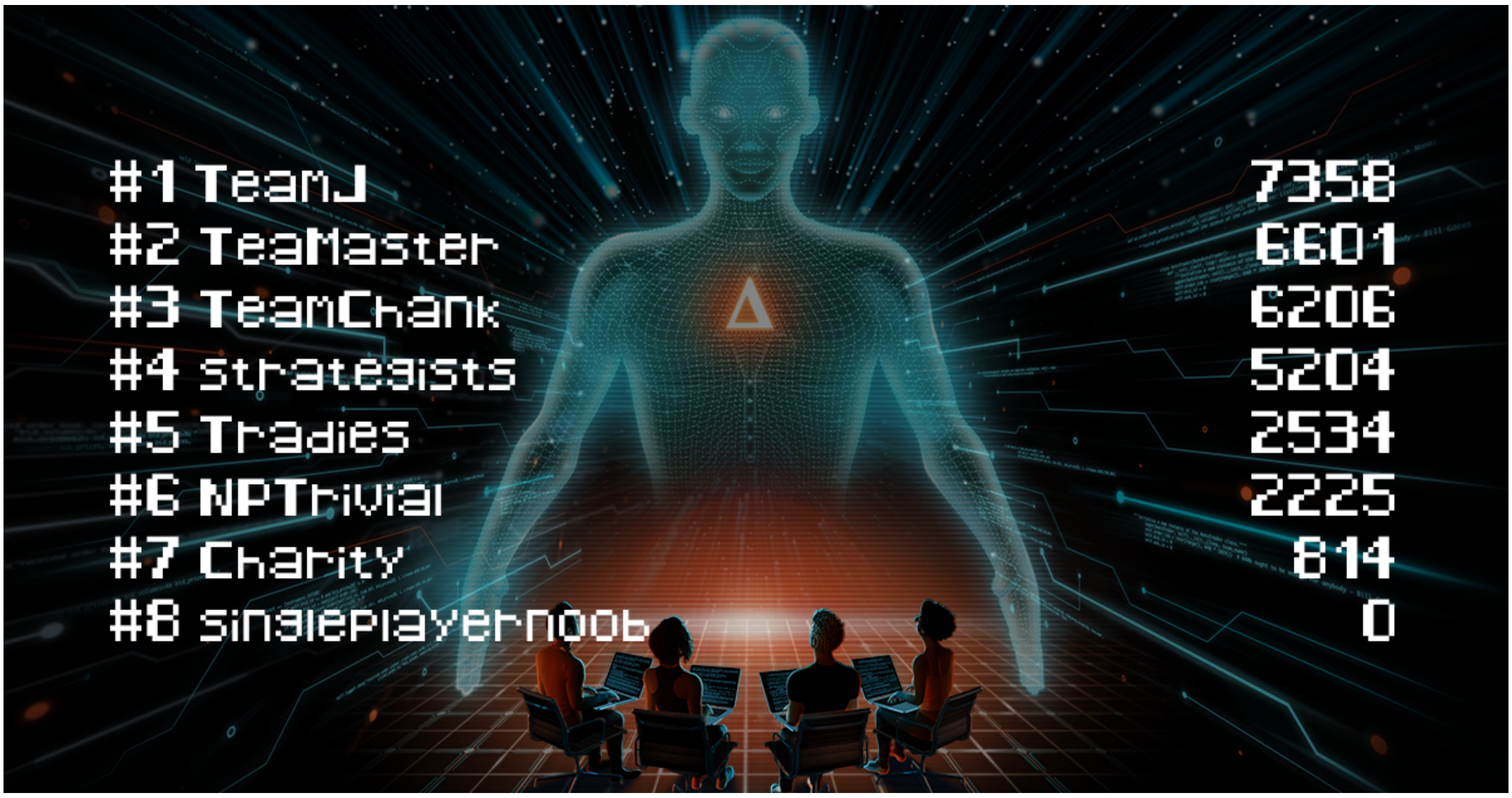
Market-making the OPTI-ETF - Optiver's Ready Trader One 2020

Team Strategists:

- Sam Breznikar - sam@sdbgroup.io (mailto:sam@sdbgroup.io)
- Nhi Huynh - *uyennhihuynhluu@gmail.com (mailto:uyennhihuynhluu@gmail.com) **withdrew from competition in Round 1*

Competition repo: github.com/s-brez/optiver-hackathon-prep



#1 TeamJ	7358
#2 TeaMaster	6601
#3 TeamChank	6206
#4 strategists	5204
#5 Tradies	2534
#6 NPTrivial	2225
#7 Charity	814
#8 singleplayernoob	0

Online Tournament 2 scoreboard pictured above.

Introduction

This notebook documents our attempt to create an optimal (most profitable) trading bot for market-making the OPTI-ETF, the synthetic financial instrument traded in Optiver's Ready Trader One (https://readytraderone.com.au/) competition.

The final and best iteration of our market making model was VolumeAdjustN15_v3 (https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%203/models/VolumeAdjustN15_v3.py). It used a naive average of levels 1-5 of the orderbook as a theoretical fair price, and adjusted its quote volume based on its current position to control inventory.

Whilst the bot was decent enough to place #4 in tournament 2, the competition grew strong, and VolumeAdjustN15_v3 (https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%203/models/VolumeAdjustN15_v3.py) did not carry us to the finals.

The following outlines how we reached the final iteration of the bot, and our testing process along the way.

Contents

---

1 - Overview

Competition

Ready Trader One was held in 4 stages, with 3 Online Tournaments followed by a grand final. Competing teams submitted their best model in the form an `autotrader.py` file to be pitted against other participants in a series of 8v8 matches in three Online Tournaments. A log file of market events was provided to competitors after each Online Tournament.

After each Online Tournament, we analysed the events in each log file, made adjustments to existing models or tested new models based on our findings, then tested our new/adjusted models on a testing set of 20 or more simulated matches with the new models competing against our previous models.

Rules

- Wash trading (trading with yourself) is forbidden. Orders that would be wash trades are rejected by the exchange.
- 20 actions (insert, modify or cancel orders) per second permitted only. Breach results in player disconnect.
- +/- 100 lot position limit. Breach results in player disconnect.
- 10 active orders only may be placed at any time. All inserted orders in excess of this limit are rejected.
- 200 lot active volume limit. I.e. the total volume of all the players orders must not exceed 200. All inserted orders which would cause this limit to be exceeded are rejected.

Assumptions

- Postions taken the ETF are automatically and instantly hedged in the future, with no fees (amazing!).

2 - Data

Data used to make decisions about our models during the competiton consisted of multiple test match results, generated with the exchange simulator provided by Optiver. Each dataset is comprised of the results of multiple runs of the simulator, ran with our best models for that stage of the competition.

Test data and models can be found here:

Online Tournament 1: https://github.com/s-brez/optiver-hackathon-prep/tree/master/Round%201 (https://github.com/s-brez/optiver-hackathon-prep/tree/master/Round%201)

Online Tournament 2: https://github.com/s-brez/optiver-hackathon-prep/tree/master/Round%202 (https://github.com/s-brez/optiver-hackathon-prep/tree/master/Round%202)

Online Tournament 3: https://github.com/s-brez/optiver-hackathon-prep/tree/master/Round%203 (https://github.com/s-brez/optiver-hackathon-prep/tree/master/Round%203)

3 - Market-Making models (trading strategies)

Model candidates for Online Tournament 1

All models tested in this round were based on the price-skewing inventory management model provided by Optiver, `Example2`.

Example2: An inventory-aware model that manages its position by skewing the price of its bid and ask quotes as it accrues long or short lots. Example2 has no theoretical price calculation, simply placing quotes at the best current bid and ask. The cost of managing inventory this way, is that profitable trades are missed if the trader accrues a large position in the other direction, as the opposing quote will be moved away from the current best price.

Symmetric-Naive: The same as Example2, but with added mechanisms to ensure hard limits are never breached, such as the 20 actions per-second limit.

Symmetric-Naive Clone: Same as Symmetric-Naive, added only to ensure there are 8 competitors for realistic testing.

Naive average of orderbook levels 1-2: Same as Symetric-Naive, but with an added theoretical price calculation where theo price = average of levels 1 and 2 of the orderbook. Bid and ask quotes are then placed at the theo price +- one minimum unit ($1), and skewed as position size increases.

Weighted average of orderbook levels 1-2: Same as Symetric-Naive, but with an added theoretical price calculation, where theo price = L1: 60% L2 40% weighted average of levels 1 and 2 of the orderbook. Bid and ask quotes are then placed at the theo price +- one minimum unit ($1), and skewed as position size increases.

Naive average of orderbook levels 1-3: Same as Symetric-Naive, but with an added theoretical price calculation where theo price = average of levels 1, 2 and 3 of the orderbook. Bid and ask quotes are then placed at the theo price +- one minimum unit ($1), and skewed as position size increases.

Weighted average of orderbook levels 1-3: Same as Symetric-Naive, but with an added theoretical price calculation, where theo price = L1: 50% L2: 35% L3: 15% weighted average of levels 1, 2 and 3 of the orderbook. Bid and ask quotes are then placed at the theo price +- one minimum unit ($1), and skewed as position size increases.

Weighted average of orderbook levels 1-5: Same as Symetric-Naive, but with an added theoretical price calculation, where theo price = L1: 37.5% L2: 27.5% L3: 15% L4: 10% L5: 5% weighted average of levels 1, 2, 3, 4 and 5 of the orderbook. Bid and ask quotes are then placed at the theo price +- one minimum unit ($1), and skewed as position size

Model selection for Online Tournament 1

20 test matches were ran with the above 8 models. Figures shown are the average of 20 matches. Figures were generated by running [this script (https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%201/testing%20data%20for%20round%201/test%20data%20-%20day%201/aggregate_scoreboard.py)](https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%201/testing%20data%20for%20round%201/test%20data%20-%20day%201/aggregate_scoreboard.py).

Model	SymmNaive	Example2	SymmNaiveClone	L12AverageNaive
PnL:	4.519	479.913	4.4.469	2284.11

Model	L12AverageWtd	L13AverageNaive	L13AverageWtd	L15AverageWtd
PnL:	1776.739	2371.9655	2116.813	2208.5385

L13AverageNaive was the best performer on average, with L12AverageNaive and L15AverageWtd close behind.

Unfortunately, due to time constraints, we were only able to run 3 test matches before the Online Tournament 1 submission deadline. In a sample size this small the best model actually appeated to be L12AverageWtd.

So the model we used for Online Tournament 1 was L12AverageWtd, with the remaining 17 tests conduction during and after Online Tournament 1.

Online Tournament 1 outcomes

Seventy-four teams submitted an `autotrader.py` for round 1. The draw consisted of five rounds:

Round 1 consisted of 16 matches each with a minimum of 4 and a maximum of 5 teams. The best 3 teams from each match proceeded to round 2. Round 2 consisted of 8 matches each with 6 teams. The best 4 teams from each match proceeded to round 3. Round 3 consisted of 4 matches each with 8 teams. The best 4 teams from each match proceeded to round 4. Round 4 consisted of 2 matches each with 8 teams. The best 4 teams from each match proceeded to round 5.

- Round 1: matches 1-16
- Round 2: matches 17-24
- Round 3: matches 25-28
- Round 4: matches 29-30
- Round 5: match 31

Out of the 31 matches, we competed in matches 14, 23, 28 and 30. We were eliminated in match 30.

Despite the suboptimal model choice, we still performed relatively well in round one, placing 9th out of 74 teams.

Model candidates for Online Tournament 2

Studies of competitor behavior from round 1 showed we needed to increase our quote volume significantly, and begin managing inventory with volume. Thus, all models tested in this round were based on the previous rounds best models, with modifications to quote volume. Several iterations of a volume-adjustment algorithm were tested. The bulk of the testing was conducted with versions one and two of this algorithm, running against each other.

Version one used a set of if-else conditions to set quote volume, where version two bisected a list (faster).

Example2: We left Optiver's default model `Example2` in the tests just to see how it compared.

VolumeAdjustN: This is the same as SymmetricNaive from round 1, but with the added volume adjust algo.

L13AverageWtd: Exact same model from round one, kept as a comparison.

VolumeAdjustW13: Same as round one, with added volume adjustment algo.

VolumeAdjustW12: Same as round one, with added volume adjustment algo.

VolumeAdjustN12: Same as round one, with added volume adjustment algo.

VolumeAdjustN13: Same as round one, with added volume adjustment algo.

VolumeAdjustW15: Same as round one, with added volume adjustment algo.

Model selection for Online Tournament 2

50 test matches were ran with the above 8 models. Figures shown are the average of 50 matches. Figures were generated by running this script (https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%202/testing%20data%20for%20round%202/vol%20algo%20v1%20-%20test%20data%20-%20day%201/aggregate_scoreboard.py) in the directory containing match events csv files (https://github.com/s-brez/optiver-hackathon-prep/tree/master/Round%202/testing%20data%20for%20round%202/vol%20algo%20v1%20-%20test%20data%20-%20day%201).

Model	Example2	VolumeAdjustN	L13AverageWtd	VolumeAdjustW13
PnL:	38.83	83.29	951.08	3016.63

Model	VolumeAdjustW12	VolumeAdjustN12	VolumeAdjustN13	VolumeAdjustW15
PnL:	3464.23	6192.96	5431.11	4937.47

Of 50 test matches, VolumeAdjustN12 was the best performer on average, with VolumeAdjustN13 and VolumeAdjustW15 close behind. Example2 and VolumeAdjustN got absolutely shredded when competing with the speed and volume-improved models.

Like in Round 1, due to time constraints, we were only able to run a handful of test matches before the Online Tournament 2 submission deadline. The reduced sample size indicated that the best model was VolumeAdjustW15, not VolumeAdjustN12. So we used VolumeAdjustW15 as the competition submission for round 2.

It is also worth mentioning that it was at this point we realised there were 10 different CSV files containing sample data. Until now, we were not aware of this and all testing had been conducted using the `day_1` test data. This means our testing and models were optimised to perform best in the market conditions present in day 1 - not ideal at all. It was too late to do further testing at this stage so we pressed on, but noted that more varied testing would be required in future.

Online Tournament 2 outcomes

Ninety-nine teams submitted an `autotrader.py` for round 2. The draw consisted of five rounds:

- Round 1 consisted of 16 matches each with a minimum of 6 and a maximum of 7 teams. The best 4 teams from each match proceeded to round 2.
- Round 2 consisted of 8 matches each with 8 teams. The best 4 teams from each match proceeded to round 3.
- Round 3 consisted of 4 matches each with 8 teams. The best 4 teams from each match proceeded to round 4.
- Round 4 consisted of 2 matches each with 8 teams. The best 4 teams from each match proceeded to round 5.
- Round 5 (the Final) consisted of 1 match with 8 teams.

Out of the 31 matches, we competed in matches 9, 21, 27, 30 and 31. We ranked #4 in the final match, match 31.

Despite the suboptimal model choice (...again), we performed well overall in round two, placing 4th out of 74 teams.

Model candidates for Online Tournament 3

For round three, we took the best models from the previous round, sought to improve the volume adjustment algorithm further, and introduced a few other general improvments for speed.

Version three volume adjustment algorithm used a hard-coded dictionary to set quote volumes (dictionary/hashmap being fastest python data structure), and stored any global method references as local variables to save stack lookup times. Version 3 models are denoted with `_3`, and are the same as their version two counterparts, but with volume algo and speed improvements. Version two models are suffixed with `_2`.

Analysis of competitor behavior from the past round showed our quotes were often wider than the teams who scored above us, suggesting we could improve or refine our pricing method to quote a tighter spread. At this stage, considering I was working by myself, I decided to just try and improve the models we already had. To develop and test a new pricing method would be a risk, and eat up time I didn't have as a solo participant.

OB = orderbook

VolumeAdjustN15_v2: Naive avg. of levels 1-5 OB for pricing, version two volume adjust algo, no speed improvements.

VolumeAdjustW12_v2: Weighted avg. of levels 1-2 OB for pricing, version two volume adjust algo, no speed improvements.

VolumeAdjustN12_v2: Naive avg. of levels 1-2 OB for pricing, version two volume adjust algo, no speed improvements.

VolumeAdjustN13_v2: Naive avg. of levels 1-3 OB for pricing, version two volume adjust algo, no speed improvements.

VolumeAdjustN15_v3: Naive avg. of levels 1-5 OB for pricing, version three volume adjust algo, has speed improvements.

VolumeAdjustW12_v3: Weighted avg. of levels 1-2 OB for pricing, version three volume adjust algo, has speed improvements.

VolumeAdjustN12_v3: Naive avg. of levels 1-2 OB for pricing, version three volume adjust algo, has speed improvements.

VolumeAdjustN13_v3: Naive avg. of levels 1-3 for OB pricing, version three volume adjust algo, has speed improvements.

Model selection for Online Tournament 3

Noting that the testing for the last two rounds was insufficient, round three model testing was conducted using `day_1` , `day_2` , `day_3` , `day_4` , `day_5` , `round_1` , and `round_2` test data. At least 15 matches were run for each set of test data and an average pnl figure taken for each model for each set of test matches. In some cases, there was only time to run six matches, so this will have had an effect on the statistical reliabilty of the numbers below.

Average highest scoring model for each test data:

- `day_1` : **VolumeAdjustN15_v3** - avg pnl: 4976.57
- `day_2` : **VolumeAdjustN15_v3** - avg pnl: 5633.31
- `day_3` : **VolumeAdjustN15_v3** - avg pnl: 6416.2
- `day_4` : **VolumeAdjustN13_v3** - avg pnl: 8562.09
- `day_5` : **VolumeAdjustN13_v2** - avg pnl: 7856.08
- `round_1` : **VolumeAdjustN12_v3** - avg pnl: 5744.36
- `round_2` : **VolumeAdjustN15_v3** - avg pnl: 5780.34

Of the seven lots of test matches, model VolumeAdjustN15_v3 won the most matches. This suggested that VolumeAdjustN15_v3 was better equipped to deal with more market conditions than the other models. As such VolumeAdjustN15_v3 was our submission for round 3.

Online Tournament 3 outcomes

One hundred and one teams submitted an Autotrader for Online Tournament 3. The draw consisted of five rounds:

- Round 1 consisted of 16 matches each with a minimum of 6 and a maximum of 7 teams. The best 4 teams from each match proceeded to round 2.
- Round 2 consisted of 8 matches each with 8 teams. The best 4 teams from each match proceeded to round 3.
- Round 3 consisted of 4 matches each with 8 teams. The best 4 teams from each match proceeded to round 4.
- Round 4 consisted of 2 matches each with 8 teams. The best 4 teams from each match proceeded to round 5.
- Round 5 (the Final) consisted of 1 match with 8 teams.

Out of 31 matches, we competed in matches 8, 20, and 26.

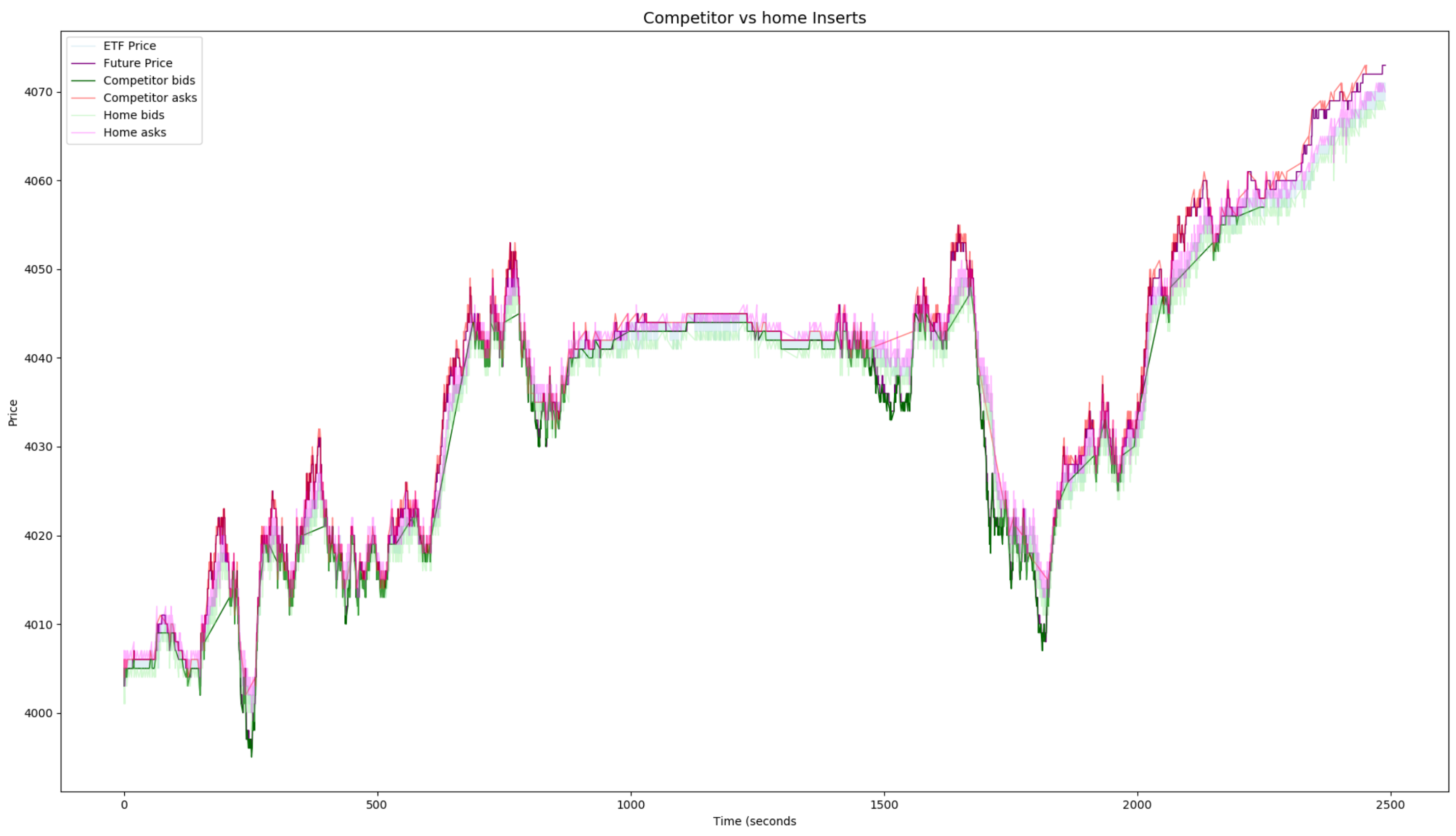
We were eliminated in match 26, and thus did not make the finals.

---

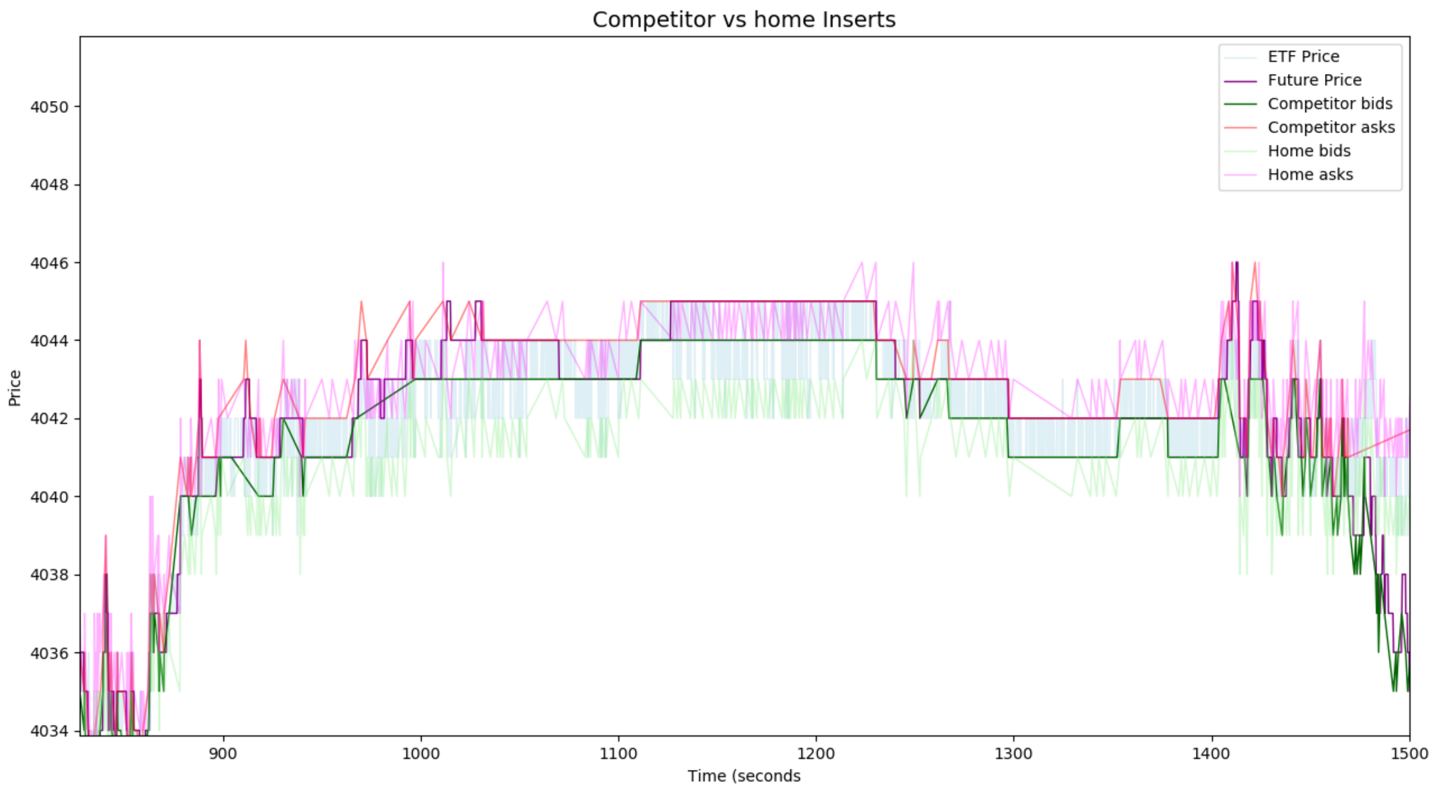
4 - Competitor Analysis

Online Tournament 1 competitor analysis

We examined the quoting behaviour of the competitors who scored above us in the highest tier of match we competed in, match 30. Team "NowUCMe" was the highest scoring team. We plotted our own quote inserts against their quote inserts against the ETF price over time to determine how they were quoting differently to us, using this script (https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%201/results_round_1/visualise_quotes.py).

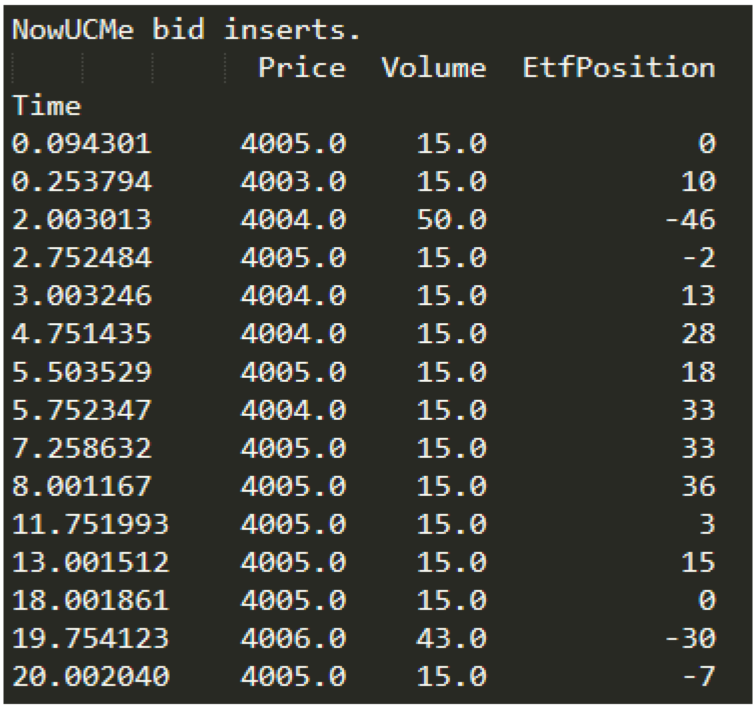


When we zoomed in on the plot above, we see that team NowUCMe was managing their postition not by skewing their quotes directionally, but by adjusting the volume of their quotes.



We had been managing inventory by skewing quotes, so this had to change.

We also examined NowUCMe's quote volumes by printing their quote inserts and current ETF position, and observing the relationship between the two. [This script (https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%201/results_round_1/visualise_quotes.py)](https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%201/results_round_1/visualise_quotes.py) was used.



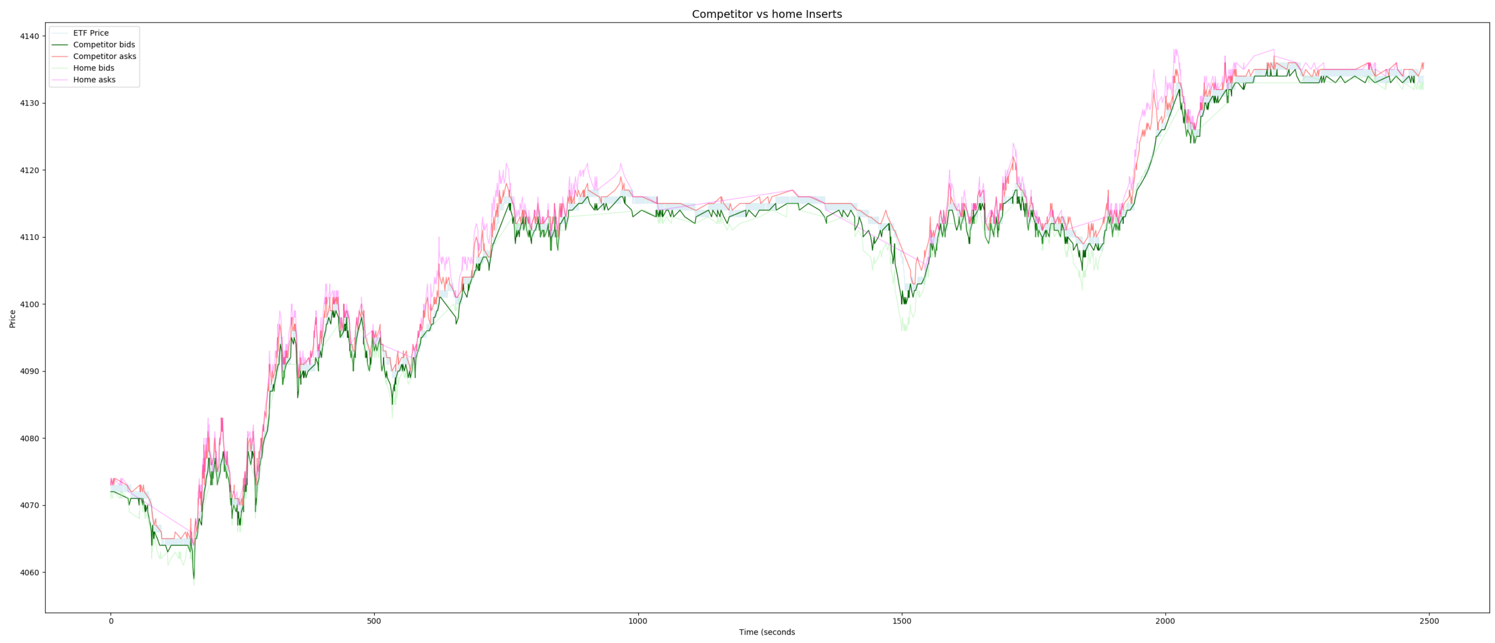
```
NowUCMe bid inserts.
                Price  Volume  EtfPosition
Time
0.094301       4005.0    15.0            0
0.253794       4003.0    15.0           10
2.003013       4004.0    50.0          -46
2.752484       4005.0    15.0           -2
3.003246       4004.0    15.0           13
4.751435       4004.0    15.0           28
5.503529       4005.0    15.0           18
5.752347       4004.0    15.0           33
7.258632       4005.0    15.0           33
8.001167       4005.0    15.0           36
11.751993      4005.0    15.0            3
13.001512      4005.0    15.0           15
18.001861      4005.0    15.0            0
19.754123      4006.0    43.0          -30
20.002040      4005.0    15.0           -7
```

The above confirmed that they were indeed adjusting their position with volume and not by skewing price. It also showed us we needed to significantly increase our quote volumes on average as our default quote volume was 1 lot in either direction.

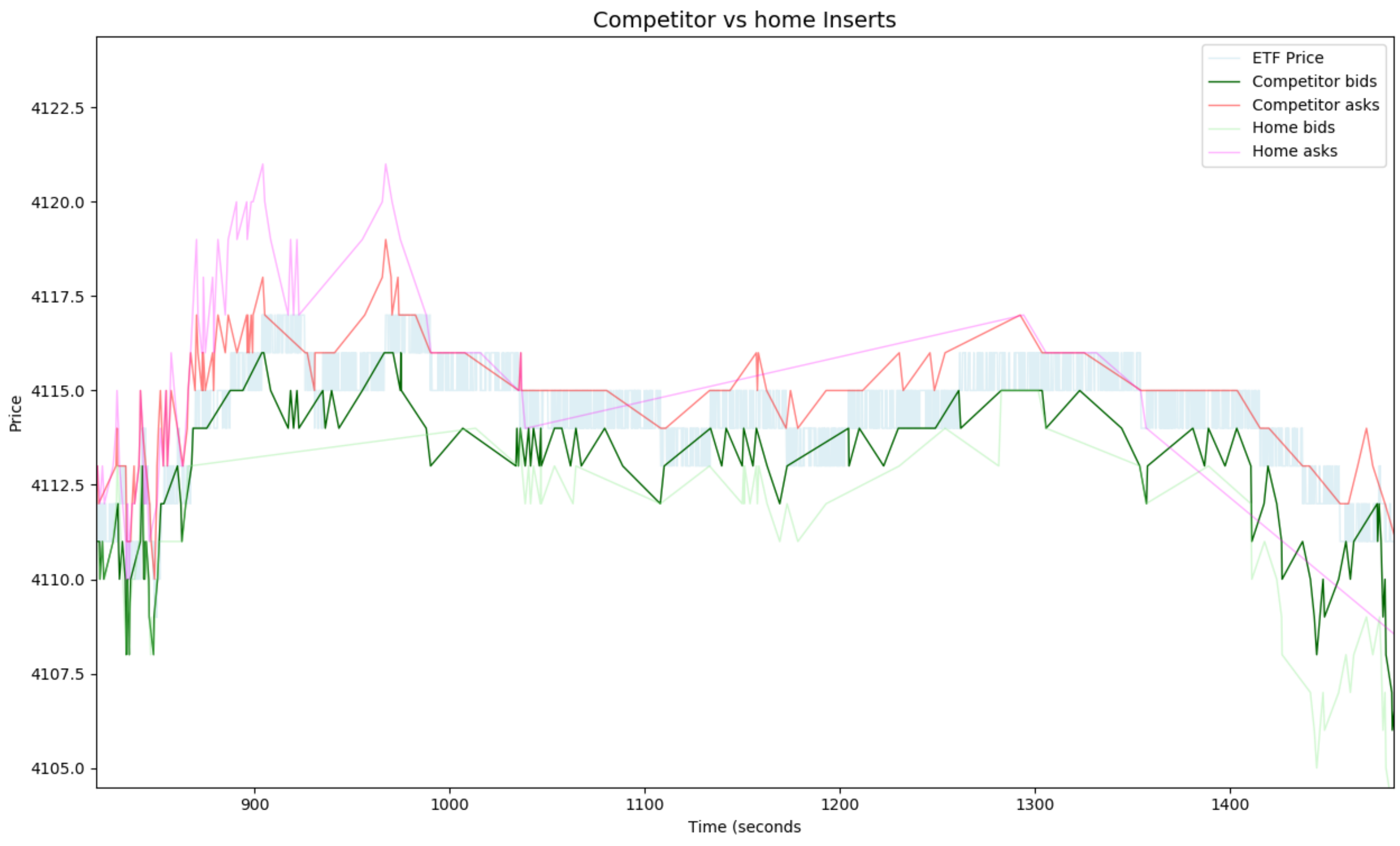
Online Tournament 2 best competitor analysis

As in round one, We examined the quoting behaviour of the competitors who scored above us in the highest tier of match we competed in, match 31. Team "TeamJ" were the best scoring team.

We plotted our own quote inserts against their quote inserts against the ETF price over time to determine how they were quoting differently to us, using [this script (https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%201/results_round_1/visualise_quotes.py)](https://github.com/s-brez/optiver-hackathon-prep/blob/master/Round%201/results_round_1/visualise_quotes.py). Please run the script for an interactive, zoomable plot as the resolution here doesnt show the quotes with accuracy.

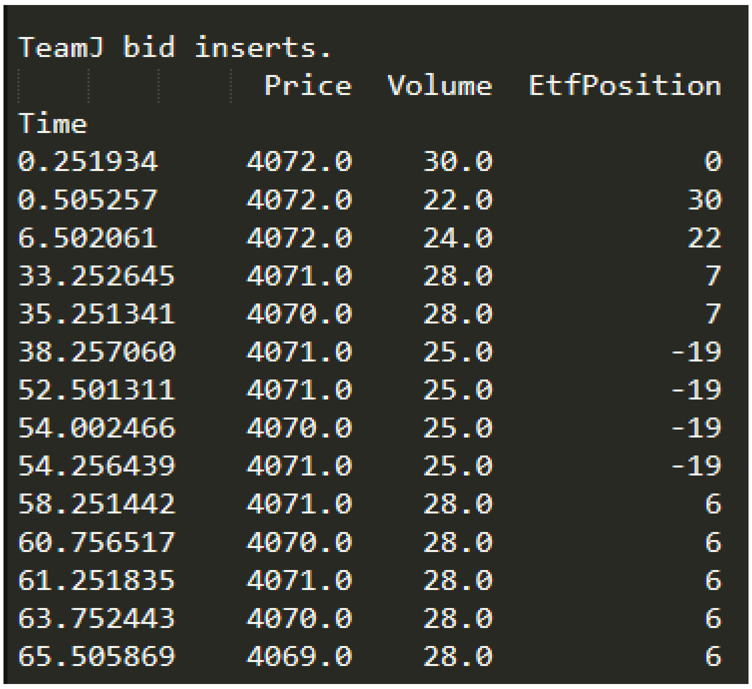


The plot above showed us that again, our quotes are not quite as tight as the competition, despite having adopting the volume-adjusting inventory method previously described. This would suggest a new pricing method is required, to be truly competitive for the next round



Zooming in on the plot reveals another difference between us and TeamJ, there were periods where we were not actually offering quotes, as a result of being at the risk limit (inventory limit).

A quick look at the quote-position relationship below shows that TeamJ's quote volume is quite a bit higher on average than anything we saw in the last round, but still not as high as our laddered quote volume system, and that their bot tries to stay close to flat, not going near the risk limit. This would suggest our volumes were perhaps too high, and that we reached the risk limit too fast when offering quotes, missing chunks of time where we could have profited from the spread.



```
TeamJ bid inserts.
                Price  Volume  EtfPosition
Time
0.251934       4072.0    30.0            0
0.505257       4072.0    22.0           30
6.502061       4072.0    24.0           22
33.252645      4071.0    28.0            7
35.251341      4070.0    28.0            7
38.257060      4071.0    25.0          -19
52.501311      4071.0    25.0          -19
54.002466      4070.0    25.0          -19
54.256439      4071.0    25.0          -19
58.251442      4071.0    28.0            6
60.756517      4070.0    28.0            6
61.251835      4071.0    28.0            6
63.752443      4070.0    28.0            6
65.505869      4069.0    28.0            6
```

Online Tournament 3 best competitor analysis

No analysis for this round due to early elimination.

5 - Conclusions

Findings:

- Based on analysis of the better performing bots, we can safely say that the ideal method to manage inventory (current position) is through adjusting quote volume, and not skewing quote prices.
- Being at the inventory limit for periods of time is detrimental to our market-making behaviour overall, with spread-taking opportunties missed due to an inability to quote.
- We were not able to idenftify a stronger performing theoretical pricing method than a simple naive average of levels 1-5 of the orderbook, though competitor analysis implies there are certainly better methods.

What we would do different next time:

- For testing, use a cloud vps service to spin up a few dozen individual instances to run many simultaneous test matches. Testing was a major bottleneck, given I was one person working on two machines.
- Pay more attention to competitor behaviour at earlier stages. I.e, it was clear that our pricing method was not quite good enough after round 2, but chose not to improve due to time/manpower constraints.
- Recruit more teammates!