2025 Quant Modeling Challenge: Market Making

Yale Quantitative Financial Modeling Group
Wharton Investment and Trading Group QIS
February 2025

0. Introduction

Welcome to our second annual modeling challenge! This year, you will play the role of a liquidity provider in a corporate bond market. The challenge is designed to be a fun introduction to market making and a financially-themed exercise in data science. Students from any university are welcome to participate. We are awarding cash prizes for top scorers, as detailed below. To join the contest distribution list, sign up here!

1. The Dataset

A large number of corporations in the green energy sector are simultaneously issuing 3-year revenue bonds to finance various expansionary projects around the world. Each bond pays an annual coupon, but the coupon amount may fluctuate based on the revenue brought in by the project. A small set of features is available to investors regarding each bond:

1. **Credit Rating**: A letter grade (e.g. AA or BBB) that indicates the issuer's perceived credit risk.
2. **Debt-to-Equity Ratio**: A quantity representing how leveraged the company is.
3. **Energy Capacity (MW)**: An estimate of the number of megawatts of renewable energy the firm has infrastructure to provide. A proxy for production potential.
4. **ESG Score**: A third-party environmental, social, and governance rating in [0, 100].
5. **Region:** The geographic continent of the firm's project (e.g. Europe or Asia).

The historical_bonds.csv file contains this information for 1000 historical bonds, along with the actual cash flows which they yielded the bondholder over the following 3 years. Assume the bonds have no principal repayment. The simulation_bonds.csv file contains these features for another 1000 bonds drawn from the same distribution, but these are not labeled with cash flows. Your goal is to design a market making agent to provide liquidity to this bond market by providing a *bid* and an *ask* price for each bond at the beginning of each year. For each bond, then, your agent must provide 3 bids and 3 ask prices: a bid and an ask at issuance, a bid and an ask after 1 year (with the knowledge of the first cash flow), and a bid and an ask after 2 years (with the knowledge of the first two cash flows). Assume that the volume on both sides of your spread is fixed at 1 bond and that cash earns an interest rate of **5% per year** in this fictional economy.

2. Evaluation

Your market-making agent will be competing with other submissions in a market trading the 1000 simulation bonds. In order to prevent any cross-agent collusion, agents will not trade with each other but rather against well-informed traders that know the *expected* cash flows of each bond (cash flows are sampled from log-normal distributions with means that are known to the traders). One trader is looking to take a short position on all of the bonds and the other is going long. Because they are well-informed, these traders will always buy a bond at your ask price if it is below the expected present value or sell one to you if your bid is above the expected present value. But the traders will also collectively sell to the next 5 best bids and buy at the next 5 lowest asks. Here is pseudocode for the algorithm dictating the collective behavior of these traders for a single bond in a single period:

```
evaluation_traders:
    Let exp_pv be the expected fair price of the bond
    if ask < bid for any submission, Buy and Sell
    for every ask with ask < exp_pv, Buy
    for every bid with bid > exp_pv, Sell
    Buy from next 5 lowest asks
    Sell to next 5 highest bids
```

So even though these traders have an information advantage, you can make money because you have the advantage of setting the spread. If your bid is below *exp_pv* but above the bids of other submissions, you can force the short trader to make an unfavorable trade. Setting good spreads will require producing strong projections of future cash flows. The dataset is not meant to be complex; most regressors should perform sufficiently to earn profit with reasonable spreads. Algorithms will be scored by total profit after trading on the simulation bonds for enough time to eliminate any variance in the winner introduced by randomness.

3. Specifications

For maximum accessibility of the challenge, we have decided to collect submissions as Python methods. Your method should have the following header:

```
def [name]_mm_agent(assets, current_time):
    # produce a bid and an ask for every asset at current_time
    return bids, asks
```

The parameter *current_time* is an integer in {0, 1, 2} referring to the current period, and assets will be a pandas dataframe with 5, 6, or 7 columns respectively. The first 5 columns will be the features, and the remaining columns refer to cash flows from earlier periods. The columns will be named and ordered exactly as in historical_bonds.csv. *Bids* and *asks* should be lists of floating point values ordered to correspond to the assets in the data frame.

To submit your algorithm, please upload it as a python file to **this form** before **Friday, February 21st at 11:59pm ET**. You may also include other files (ex. pickle files) to save any models used by your algorithm. Your algorithm should be able to price 1000 assets in at most 10 seconds.

4. Prizes: First Place: $250; Second Place: $100; Third Place: $50

5. Office Hours

We plan to hold office hours for the tournament both in-person for students at Yale and on zoom for others. Please register for the competition email distribution for more details. Also feel free to email any of us at YQFM or WITG QIS with any questions.

Contact: yaleqfm@gmail.com