Willem Wang

Part 1

My program is heavily reliant on the SortedDict data structure in the sortedcontainers package, which helps continuously maintain bids and offers in increasing order. The program is structured such that for each update in the provided input files, either the bid or ask dictionary will be referenced and modified. After the addition, modification, or deletion, price levels without any quantity are deleted with the clean() function. This allows for continuous access to the top of book.

Admittedly, the program could be more object oriented; if I were to refactor it, I would define a class called orderbook and keep the methods and attributes local. Other improvements would include keeping the orderbook in memory and writing only after processing all the updates, rather than writing on each update. Finally, the naming could be improved.

Part 2

I have chosen to predict the best bid price, or bp0. My thesis is quantity helps predict the product’s true price. Correspondingly, most predictors have some combination of both price and quantity. First, the book is filtered for both a bid and ask. This is to eliminate possible outliers after calculating predictors. Next, random training and testing sets are automatically generated with sklearn. After using lasso, the dominant predictors are a weighted average, and a variant called book pressure which switches the bid and ask quantities in the numerator. Interestingly, lasso eliminates predictors that only have size and no price such as the imbalance ratio in [Lipton 2013](https://arxiv.org/pdf/1312.0514).

After predicting the price, we use it to make buy and sell decisions. A basic strategy would be to take from the top of book once there is enough theoretical edge with our prediction. For example, if the predicted price is 0.2 points below the best bid, we would sell. A buy signal would be triggered when the predicted price is at least 0.4 points above the best offer. These decisions are summarized as two columns of Booleans in the output files. There is much tuning and backtesting to be done considering the signal results in many trades (and fees). Risk management is also of concern as the signal may put on very large positions. I envision this as an intraday strategy.

Statistically, there are issues of multicollinearity among my predictors, as well as autocorrelation in the data. The high r2 may be an artificial side effect of these two details. Ultimately, there are numerous diagnostics to be conducted to improve the model.