Shira Rozenthal and Alex Olteanu
Professor Maus
CMPS4010
Due December 8th, 2023

Milestone 4

Our capstone project focuses on tackling Kaggle competition initiated by Optiver, a prominent electronic market maker. Optiver released extensive high-frequency market data, challenging participants to enhance pricing algorithms by predicting short-term volatility based on order book and trade information. The dataset includes order book details, executed trades, and training data with target realized volatility, while the goal is to accurately predict stock volatility in the 10-minute window following a market order using Root Mean Square Percent Error (RMSPE) as the evaluation metric.
In milestone 3, we reviewed our progress with data access, EDA, and reverse-engineered timeID. In this milestone, we will go over our early stages of feature engineering and model selection before mapping out our timeline and goals for next semester. Access to our notebook can be found here on our github under Milestone 4.

Progress Made: Feature Engineering by StockID-TimeID Pair

For each StockID and TimeID pair, we have both a Book and a Trade DataFrame, as well as a single volatility target. In the early stages, in order to effectively feed them to our model, we "aggregated" each of the two dataframes by taking the mean / max / min / range of their values and then merging them together on StockID and TimeID. This leaves us with a master Dataframe, with 1 row of book/trade data and 1 target volatility output per StockID/TimeID pair.

Given that our mode of aggregation was somewhat arbitrary, we ran a correlation on each of the aggregate features to see if they are relevant predictors of volatility. We used those with a correlation > 0.6 as features in our model.

Progress Made: Feature Engineering with Neighboring TimeIDs

Although some of the pairwise aggregate features turned out to be meaningful model inputs, the strongest predictor for volatility at TimeID(n) is going to be volatility at TimeID(n-1). Having chronologically sorted our TimeIDs last milestone, we began introducing neighboring TimeIDs to our model. So far, we have only brought in the target (volatility) of timeIDs (n-1) and (n+1), and just that made a drastic improvement on our error (this put us below 0.3). There is still so much to work with going forward, including "similar behaving" timeIDs at different points in time as well as some of the pairwise aggregate features of consecutive ones.

Progress Made: Running Models

After quite the semester, we have a few models running. We initially began with a K-NN model, which proved to be our strongest bet. After we got that one down, we decided to pull in some additional models we've learned to work with in Machine Learning this semester. The models and their RMSPEs are as follows:

K-NN RMSPE: 0.26675941703427386

Linear Regression RMSPE: 0.2913722425547761

Random Forest RMSPE: 0.27258505567872915

SVR RMSPE: 11.245415973325791 (yes 11) (not 11%) (oops)

Gradient Boosting RMSPE: 0.2687369762052166

Not surprising to see the regressions underperform given that market data is tainted with randomness, but we are happy overall to see our error below 0.27. We've adjusted our timeline accordingly, now hoping to get < 0.24 RMSPE by mardi gras.

Challenges Encountered

Our biggest weakness right now is how we are working with the structure of our data. By aggregating it down, we are losing so many of the valuable intricacies of high-frequency data and completely setting aside the relationship between Book and Trade tables.

As we move forward, we have to acknowledge a fault in our approach. We are currently using TimeID(n+1) in our model, which isn't a feasible input in real-time. We intend to continue training on future data in order to remain competitive with top submissions (who all admittedly use it in their models as well).
However, we additionally plan to run a separate "for-use" model that plays fair, and only trains on TimeIDs prior to the one being predicted.

Finally, we have been notably limited by the models we feel confident implementing. Most of the top submissions are running deep-learning models – which is a shortcoming of ours that we will attempt to tackle next semester.

For Next Semester

Our first and primary focus next semester will be to reconfigure our input, hopefully finding a way to model a sequential-to-1 relationship.

Next, We have a lot more to do with neighboring TimeIDs. As mentioned above, we want to run a model that will explore comparable "clusters" of TimeIDs to build out some more advanced pattern-based features. We haven't even looked at how the pairwise features of consecutive TimeIDs relate to one another, which has a lot of potential given that volatility is a stdev measurement of the price movements that can be observed in previous TimeIDs.

Lastly, we're going to play around with some models that are out of our comfort zone as well as continue to build upon (some of) the ones we've been running. The goal is now to be < 0.24 RMSPE by mardi gras, hopefully setting us up better for a < 0.2 RMSPE by end of semester when we would want to submit our solution.