PART III

Here we are tasked with using the data in returns/volume/risk data in the "MLTestRiskDataUS.csv" file to predict forward returns.

Admittedly, I did not get to spend as much time on this task as I would have liked, but there were a few things that I thought were somewhat challenging, largely related to the lack of explicit timestamps on the data (I supposed that perhaps time is monotonically increasing with the rows, but without explicitly knowing that I could not assume it). So, among other things, I decided I could not:

- Do cross-sectional modeling, and/or use covariances among assets in any way.
- Do rolling (backward-looking) model training.
- Normalize the returns inversely with any custom trailing volatility estimated over some period (the "risk" feature we're provided could suffice, but we don't know how that's estimated either). To be fair I think vol/risk-normalization of returns is more important for futures contracts.
- Construct new features based on trailing measures... e.g. a 5-day z-score of the price.
- Construct features based on changes in quantities, e.g. change in volatility over some recent period.
- Build trading systems based on certain classes of simple rules (e.g. defining trends or reversion indicators), which rely on having a notion of a time axis that we don't have access to here.

Another issue which confused me was the question of why we seem to only have in-sample data for the "ret" feature. I would assume this is a daily (backward-looking) return; it's not labeled "fwd," and it is highly correlated with the "prior" returns. I may be missing something here. it's not what we're being asked to predict, but it's not provided out-of-sample….

So, I found this situation to be a bit limiting in terms of what I felt I could do. I did construct some new features, including risk-normalized returns, but nothing I added seemed to confer any significant predictive advantage.

Another issue was the large size of this dataset, so batch methods with poor scaling (e.g. SVM/SVR) would be impractical. Nevertheless, for noisy data with potential outliers, I am often a fan of the epsilon-insensitive/hinge-loss function used in SVR. Rather than using SVR explicitly, we settled for an online stochastic gradient variant where we train a linear regressor to minimize the hinge loss. Unfortunately, this only seemed to give a barely positive R^2 and a barely above 50% accuracy in correctly forecasting the sign (when cross-validated on in-sample data). However, it appeared to work more reliably than squared-loss methods like ridge regression.

I briefly played with feature selection as well as using a couple of nonlinear methods (XGBoost and a single-layer neural network), but without a principled reason for doing so and without a

good way of going about investigating these methods (in my limited amount of time), I did not pursue them at any real depth.

So I am not satisfied with the result—a forecast of the 5-day returns using the linear hinge-loss regressor—but it is all that I could manage in the limited time.