

I like the idea of providing more data and information about the metamodel to users, but I believe that the proposed methods may have some drawbacks.

If each week we are provided the exposures of the meta-model from the previous round, then if people are trying to change their models accordingly, it may cause the meta-model to become more volatile, as each week people are trying to optimize on changes from the previous week, creating a constantly moving target.

This also makes the assumption that the meta-model's feature exposure isn't an integral part of its performance. We don't have round performance for the meta-model, so we have no way of knowing whether it burned in round 260, or if linear exposure was the culprit. If we were provided a full model page for the meta-model, then we could compare its fnc to its corr to get a sense of whether a particular round's feature exposures matter or not, but this wouldn't solve the problem of a feedback moving target.

I think a better idea would be providing a snapshot of the meta-model predictions alongside training and validation data (not for test or live as it is a snapshot). This snapshot would not change over time, although potentially a new snapshot of it could be provided at a future point in time. This would provide us information on how the meta-model does in comparison to our models, as well as the opportunity to use it in the creation of our models. I imagine doing something like neutralizing the meta-model snapshot to the target, and then attempting to reduce correlation to the leftover component during training. Or, you could setup the problem as how do I build a model that always improves the meta-model when ensembled together with it.

I think that since this would be provided for the model building process rather as a post-processing effect, it would incentivize people to create good long-term unique models that improve the meta-model, rather than short term adjustments chasing weekly changes in the meta-model.