Opening-gap Model – 14/08/19

The training consisting of DAX equities up to year 2018, all combined into a single matrix

Validation data set of separate DAX equities, starting from 2018

Regularized regression:

lasso\_model <-  
 train\_data %>%  
 sample\_frac( # 2/3 train samples w/o replacement  
 size = 2 / 3,  
 replace = F,  
 weight = normalize01(train\_data$Recency)  
 ) %>%  
 train\_lasso()

gradient boosted trees:

xgboost\_model <-  
 train\_data %>%  
 sample\_frac( # 2/3 train samples w/o replacement  
 size = 2 / 3,  
 replace = F,  
 weight = normalize01(train\_data$Recency)  
 ) %>%  
 train\_xgb()

We distinguish between two kinds on weights:

1. Selection weights: the likelyhood of given observation to be taken into training set
2. Training weights: The importance of selected observtions features for the regression

Variation of weights between underlying models should reduce the corelation between them. The weights selection remains unsolved, it’s hard to optimize due to the time complexity (it’s super slow). The results are extremely sensitive, especially to the training weights, in the current state of implementation they aren’t considered by the model.

The general idea behind the model is to leverage the bootstrap aggregating to the combination of regularized regression (very general, no overfitting, but only capable of capturing the global trend) and boosted trees (hopefully capable of infering richer concepts, but more prone to overfitting), both trained on random subset of the initial dataset.

As the approach of the basic linear model was maybe too simple, I was hoping to improve it with momentum sensitive boosted trees, without losing the perception of the global trend. As of now it’s actually yielding worse results than the simple version, however it looks like it already manages to incorporate the concepts of tighter time frames. Some work on the feature engineering still has to be done (Even in it’s current form it has a slight edge over the market.

TODO:

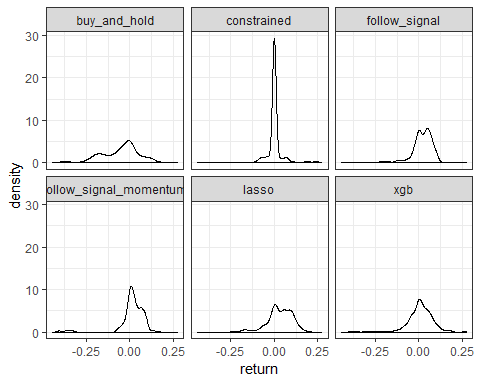
* bootstrap aggregating non-TA features
* investigate equities on which the model repeatedly fails
* Find better constrains for the constrained signal handler
* Evaluate adding ta trend reversal patterns

# DAX

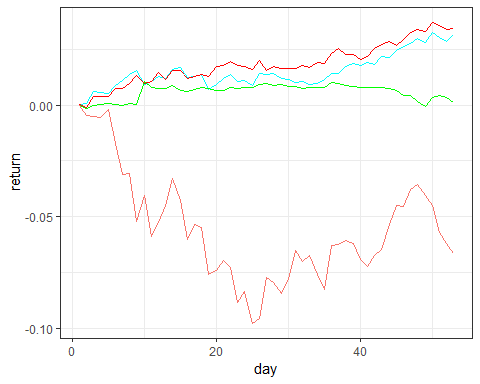
Predicted returns:

dax\_returns <-   
 dax\_test\_data %>%  
 get\_backtesting\_data(  
 dax\_lasso\_model,  
 dax\_xgboost\_model,  
 dax\_equities  
 ) %>%  
 map(get\_returns)

returns densities:



mean equity:



separate performances:

