Mandatory Assignment 2 - Hand In

pxq563

2022-04-17

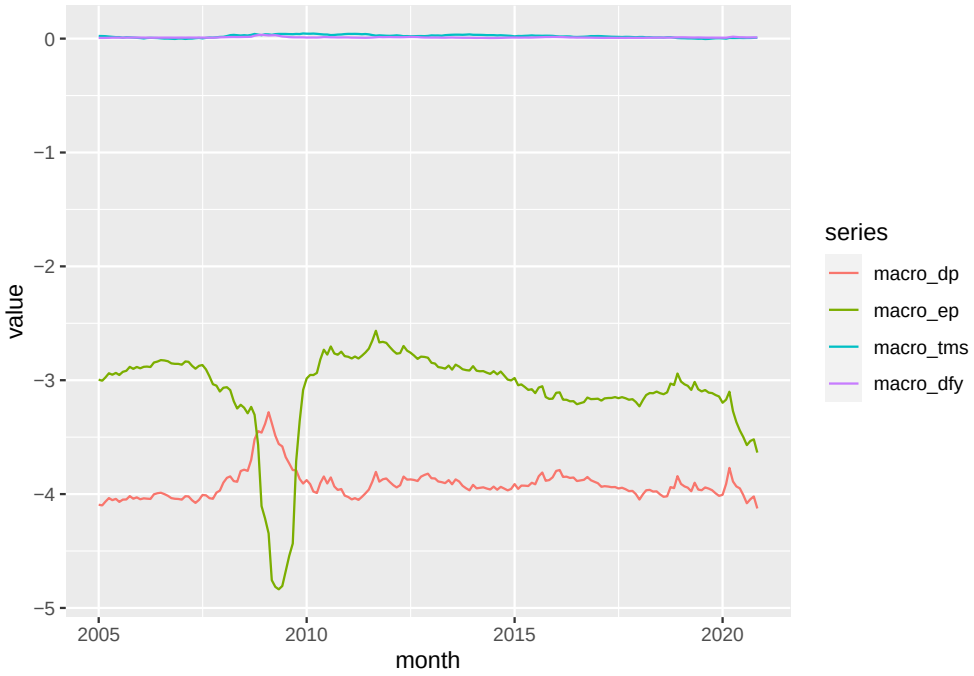
Ad 1

In this assignment we will be examining the conditional expected excess returns of a large selection of US stocks. This empirical problem is of great importance within the literature of asset pricing, as well as for practitioners in the industry. For this analysis, we will be working with a subset of the data from the paper *Empirical Asset Pricing with Machine Learning*, both with regards to sample length as well as number of predictors. In Table 1, the 5 stock characteristic variables and 4 macroeconomic variables that we have chosen for our analysis can be seen, in addition to preliminary summary statistics.

Table 1: Average cross-sectional summary statistics

name	mean	sd	min	q05	q25	q50	q75	q95	max	n
characteristic_maxret	0.12	0.53	-0.99	-0.75	-0.31	0.14	0.57	0.92	0.99	3895.75
characteristic_mom12m	-0.01	0.59	-0.99	-0.91	-0.52	-0.02	0.52	0.91	0.99	3895.75
characteristic_mom1m	0.00	0.60	-0.99	-0.91	-0.54	0.00	0.54	0.91	0.99	3895.75
characteristic_mvel1	0.04	0.59	-1.00	-0.89	-0.49	0.06	0.55	0.92	1.00	3895.75
characteristic_retvol	0.13	0.52	-1.00	-0.73	-0.30	0.15	0.57	0.91	0.99	3895.75
macro_dfy	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	3895.75
macro_dp	-3.92	0.00	-3.92	-3.92	-3.92	-3.92	-3.92	-3.92	-3.92	3895.75
macro_ep	-3.09	0.00	-3.09	-3.09	-3.09	-3.09	-3.09	-3.09	-3.09	3895.75
macro_tms	0.02	0.00	0.02	0.02	0.02	0.02	0.02	0.02	0.02	3895.75

For our subset of data we have chosen to work with the following variables. For stock characteristics, we've included *1 month momentum* (mom1m) and *12 month momentum (mom12m)*. Especially the first, representing short-term reversal, have historically proven to be a strong predictor. Furthermore, we have chosen *size* (mvel1), *return volatility* (retvol) and *maximum daily return* (maxret), in order to capture information about how large the firms are, as well as how dynamic their stock price is. For macro variables, we choose *dividend-price ratio* (dp), which has historically been a cornerstone in asset pricing theories. *Earnings-price ratio*, which is an important marker for how good firms are at generating cash. The *term spread* (tms) and *default spread* (dfy) are related to the business cycle of the economy.



Ad 2

The empirical approach conducted here, where a set of covariates are used to explain excess stock returns, have certain limitations. Statistical irregularities and low signal-to-noise ratios may obscure the *true* (should such exist) relationship between returns and other variables. In other words, the statistical models fit the data (which may contain irregularities or abnormalities), but does not necessarily touch upon the theoretical underlying market equilibrium price dynamics (as you would in theoretical models). Furthermore, for this particular methodology, we have assumed that the mapping $g(\cdot)$ is constant over time. This may be convenient from a implementation point-of-view, but not necessarily theoretically plausible.

The prediction error problem in Gu, Kelly and Xiu (2020) can be seen as a generalization of the Arbitrage Pricing Theory (Ross, 1976). The regressors, $z_{i,t}$, i.e the mix of stock characteristic variables, macroeconomic variables, and interactions inbetween, play the role of factors in APT. Here, the functional form of $g(\cdot)$ would simply be linear, while in our empirical setup, this form is more flexible with possible non-linear specifications (although not time-varying).

Ad 3

Machine learning models often come with certain parameters that needs to be specified a priori to the usual estimation procedure. These often go by the name of *hyperparameters*. These hyperparameters are determined by measuring some estimation error on a *validation* set of the data, which kept separate from the training and testing sets. This procedure is often denoted cross-validation, when multiple validation sets are used. In general, the hyperparameter tuning is done by choose the values from a discrete grid that minimizes the following loss function

$$\mathcal{L}_{CV}(\theta) = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(r_{i,t+1} - g(z_{i,t};\theta))^2$$

where N refers to the number of stocks at a given time, T the number of time periods and B the number of random data draws. In our assignment however, we will use a simpler setup where (roughly) 60 pct. of the data is used for training, the following 20 pct. used for hyperparameter tuning and finally the remaining 20 pct for actual performance measuring. This is elaborated below.

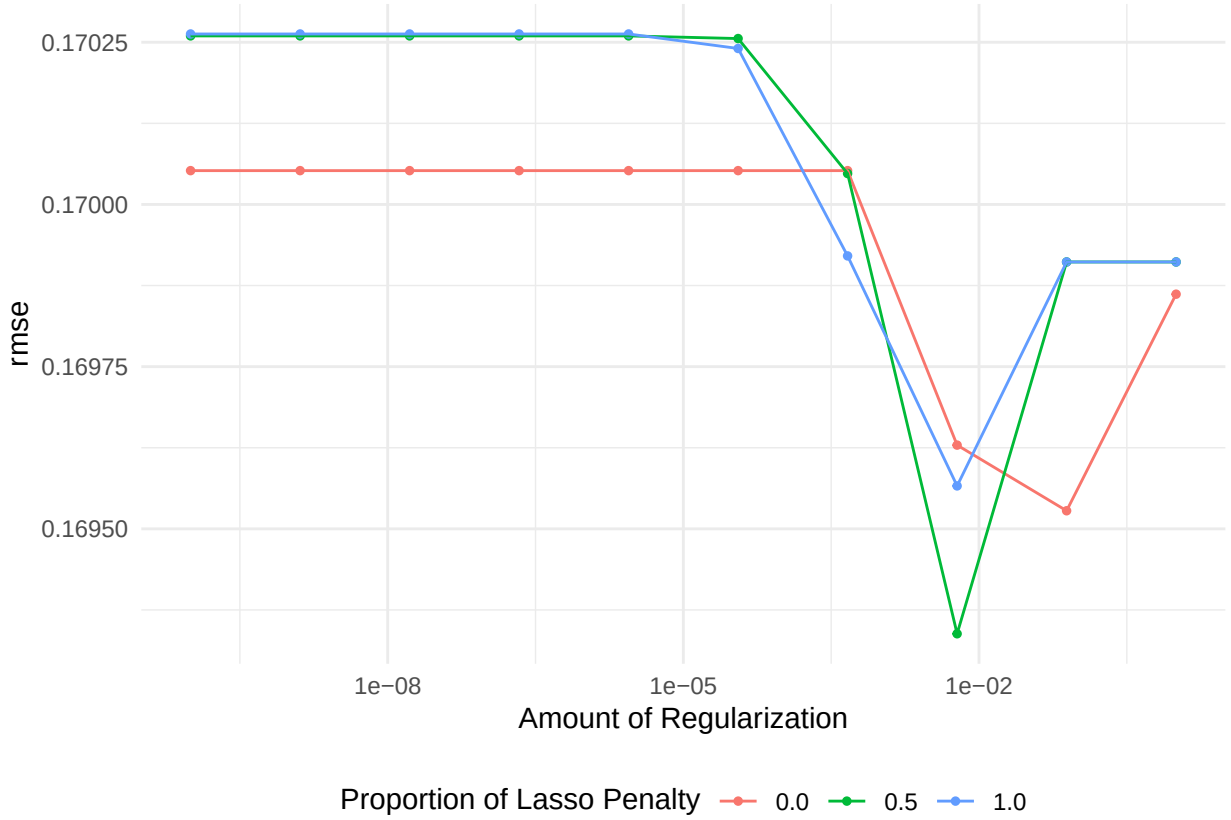
Through hyperparameter tuning on the validation set, we try to find the best parameter values for the given data, without introducing bias (i.e. data-snooping). Some machine learning models however, are less sensitive than others with respect to their hyperparameters. The performance of the Random Forest model is often quite robust despite its parameter specifications. In such a case with pre-specified values, there's a gain of obtaining an additional large amount of training data, potentially leading to better predictive performance.

```
# Split main data into "training" and "validation" part - 80/20
split <- initial_time_split(
  data,
  prop = 4 / 5
)

# Initial - set training data size to 60 pct of the original size
initial <- floor((training(split) %>% count(month) %>% nrow() +
                    testing(split) %>% count(month) %>% nrow())*0.6)

# Split training into training and validation, such that overall distr. is roughly 60/20/20
data_folds <- time_series_cv(
  data = training(split),
  date_var = month,
  initial = initial,
  assess = training(split) %>% count(month) %>% nrow() - initial,
  cumulative = FALSE,
  slice_limit = 1
)
```

```
## Overlapping Timestamps Detected. Processing overlapping time series together using sliding windows.
```

The code snippet above splits the data three-way, into a training, validation and test set. We have opted for a rough 60/20/20 split. The large portion of training data is essential for our models to learn the structure of the data. At the same time, the validation set and test set needs to be large enough to consistently estimate the optimal hyperparameters as well as the predictive performance respectively.

Ad 4



We've chosen to implement an linear regression with an Elastic-Net regulization term. That is, a mixture of both the Ridge and Lasso models. Additionally, we've also implemented a Random Forest model. It appears that the mixture-specification of the Elastic-Net model is able to achieve a lower root mean squared error.