Mandatory Assignment 2

Starting note As I struggled with some of the exercises, and my codes did not work, I have removed my trial attempts, since one of the requirements is that there must be no errors or interruptions when running the codes. I would really appreciate tips and hints to solve the exercises I didn't manage to solve. (This section will of course be removed when I revisit the assignment after the peer feedback). Furthermore, the random_forest is not tuned, as the code took forever to run. If you have any suggestions, feel free to give hints here as well.

Exercise 1 The data used in the mandatory assignment is the Fidy_Finance_ML SQL database containg the predictors from *Empirical Asset Pricing with Machine Learning* and macroeconomic predictors from Goyal Welsh Homepage. The variables used in this assignment are 5 stock characteristics: 1-month momentum (mom1m), 12-month momentum (mom12m), share turnover (turn), return volatility (retvol), and change in 6-month momentum (chmom) , the 4 macroeconomic variables: Net Equity Expansion (macro_ntis), Treasury-bill rate (macro_tbl), book-to-market ratio (macro_bm), and dividend-price ratio (macro_dp), and the stock identifier (permno), the month states year-month-date (month), the excess returns (ret_excess), the lagged market capitalization (mkt_cap), and industry classifications (sic2).
The 5 stock characteristics are chosen based on *figure 4: Variable importance by model* and *figure 5: Characteristic importance*. Momentum is one of the characteristics, that is important in the models. To capture both short term, medium-term, and long-term momentum, both 1-month momentum, change in 6-month momentum and 12-month momentum is included. Furthermore, the share turnover and return volatility is chosen as the two last characteristics.
The 4 macroeconomic variables are chosen based on *table 4: Variable importance for macroeconomic predictors*, where the variable importance is measured. Looking at the graph to table 4, book-to-market ratio have high importance in each model used in *Empirical Asset Pricing with Machine Learning*. The Treasury-bill rate is chosen such that we have a proxy for the risk-free rate in the dataset. The fourth variable, dividend-price ratio, is the difference between dividends and prices, both in logs. It was chosen as the fourth variable to have some price related variable in the dataset. The net equity expansion is chosen due to its high importance in the random forest model.

In the figure below, the summary statistics are presented:

Table 1: Table 1: Summary statistics

permno	month	ret_excess	mktcap_lag	sic2	characteristic_mom1m
Min. :10001	Min. :2005-01-01	Min. :-1.000000	Min. : 0.1	Min. : 1.00	Min. :-0.989853
1st Qu.:37154	1st Qu.:2008-04-01	1st Qu.:-0.063805	1st Qu.: 103.4	1st Qu.:33.00	1st Qu.:-0.550985
Median :80193	Median :2012-02-01	Median : 0.001585	Median : 452.0	Median :48.00	Median :-0.007826
Mean :64861	Mean :2012-05-09	Mean : 0.007135	Mean : 4908.7	Mean :47.76	Mean :-0.002784
3rd Qu.:88791	3rd Qu.:2016-04-01	3rd Qu.: 0.065830	3rd Qu.: 2031.7	3rd Qu.:61.00	3rd Qu.: 0.546059
Max. :93436	Max. :2020-11-01	Max. :19.881489	Max. :2206911.1	Max. :99.00	Max. : 0.990180

As seen in the summary output of table 1, the mean of the excess returns and the market capitalization are 0.007 and 4908.7, respectively. Further, the summary statistic shows sign of skewness, as the average excess return and the average market capitalization are greater than the median of the excess return and market capitalization. This is often seen in financial data. Hence, this will not be investigated further in this

assignment. The mean of 1-month momentum and 12-month momentum are -0.003 and -0.01, respectively, which indicates both short-term and long-term reversal effect. The mean of share turnover is 0.04, and the average return volatility is 0.128. The mean of the change in 6-month momentum is slightly negative.
The dataset also contains 4 macroeconomic predictors, which is illustrated in Figure 1 - 4.

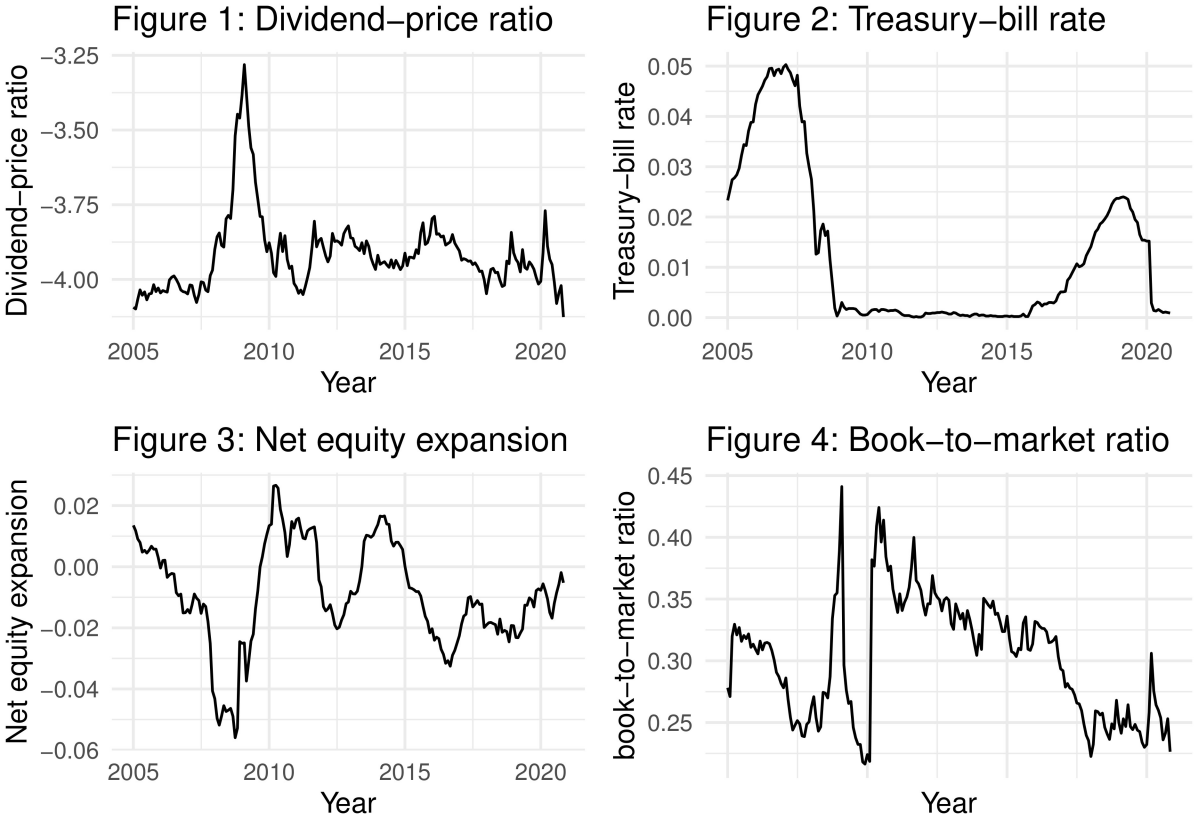

Figure 1 shows the dividend-price ratio from 2005 to 2020. As the graph shows, the dividend-price ratio increased sharply during the financial crisis, but remained negative throughout the period. Figure 2 shows the Treasury-bill rate from 2005 to 2020, which increases in the first years until about 2008, where it decreases until the middle of 2015. It then increases again until the last part of 2019, where it takes a dive. Figure 3 shows the net equity expansion, which fluctuates throughout the period with a large decrease after 2007 followed by a increase in around 2009. Figure 4 shows the book-to-market ratio, which fluctuates throughout the period with a downward trend beyond around 2010, where it takes a dive.

Before we start working with the data, two further cleaning steps will be performed. In the recipe code, all interactions between the 5 stock characteristics and the 4 macroeconomic variables are created, and dummies for the variable sic2 will be implemented in the dataset.

Exercise 2 In Gu, Kelly and Xiu (2020), an asset's excess return is described as an additive prediction error model:

$$r_{i,t+1} = E_t(r_{i,t+1}) + \varepsilon_{i,t+1}$$

where

$$E_t(r_{i,t+1}) = g(z_{i,t})$$

One of the limitations of this modeling approach is, that the function $g(z_{i,t})$ does not depend on i or t, which means that the function maintains the same form for different stocks and months. This is in direct conflict with standard asset pricing approaches. Furthermore, $g(z_{i,t})$ does not use information prior to t or for other stocks than i.

Exercise 3 The purpose of hyperparameter tuning is to make sure, that the model does not overfit. A

limitation/concern with hyperparameter tuning is that there is limit theoretical guidance on how to perform the hyperparameter tuning.

The data is splitted into a training and a testing subsample. where the most recent 20% of the observations will be used for the out-of-sample testing (the testing sample). For the remaining 80% of the sample (the training sample), 60% of the sample goes to the training set and the last 20% of the sample goes to the validation set in the report. The training and the testing subsamples have 595,270 and 148,818 observations, respectively.

Exercise 4 In this assignment, the random forests and elastic net is implemented. Random forests: Random forests are a further development of the decision trees that addresses the shortcoming for example the high variance in decision trees. Random forests work by creating multiple decision trees and take the average of the predictions. Bootstrapping is used to induce randomness by assuring that the trees are not similar. Using the training sample, B decision trees are created, where features are randomly selected for each tree. The prediction of each observation is computed as:

$$\hat{y} = \frac{1}{B}\sum_{i=1}^{B} \hat{y}_{T_i}$$

In the hyperparameter tuning, the number of trees and the observations in each branch are generated to give the smallest possible mean square error (MSE).

Elastic net: The elastic net combines L_1 with L_2 penalization. The general framework considers the optimization problem:

$$\hat{\beta}^{EN} = argmin_{\beta}(Y - X\beta)'(Y - X\beta) + \lambda(1-\rho)\sum_{k=1}^{K}|\beta_k| + \frac{1}{2}\lambda\rho\sum_{k=1}^{K}\beta_k^2$$

For the elastic net, two parameters has to be chosen, λ and ρ, the shrinkage factor and the weighting parameter. In the tuning part, this are the two parameters, we have to find the optimal value of. The optimal value of λ and ρ is the ones, that minimizes the root mean square error (rmse). The elastic net will be implemented with the *glmnet* package.

First, the random forest is computed with 50 trees and at least 20 observations in each node. As my code for the hyperparameter tuning did not work, I kept the model with 50 trees and at least 20 observations in each node. Random forest delivers a MSE of 0.025. Table 2 shows the first 6 observation of the random forest, which will be used in exercise 5.

Table 2: Table 2: Head of Random Forest

permno	month	.pred	ret_excess	mktcap_lag
14542	2017-03-01	0.0314	0.0074	285598.7226
14543	2017-03-01	0.0176	0.0085	391.5100
14544	2017-03-01	0.0148	0.1560	594.1379
14545	2017-03-01	0.0368	-0.0155	73.0897
14547	2017-03-01	0.0440	0.0953	1813.6685
14548	2017-03-01	-0.0188	0.1277	63.3825

The other method used in this assignment is the elastic net. The parameters λ and ρ is tuned in the elastic net with a 20-fold cross-validation and a $10 * 3$ hyperparameters grid. Table 3 shows the optimal values of λ and ρ, which delivers the lowest RMSE. The optimal value of λ is 0.006 and 0 for ρ. Thus, the optimal model is Elastic net (0.006). Table 4 shows the first 6 observation of the elastic net, which will be used in exercise 5.

Table 3: Table 3: Lowest RMSE for the Elastic net

penalty	mixture	.metric	.estimator	mean	n	std_err	.config	.best	.bound
0.006	0	rmse	standard	0.1544	20	9e-04	Preprocessor1_Model08	0.1536	0.1545

Table 4: Table 4: Head of Elastic Net (0.006)

permno	month	.pred	ret_excess	mktcap_lag
14542	2017-03-01	0.0138	0.0074	285598.7226
14543	2017-03-01	0.0238	0.0085	391.5100
14544	2017-03-01	0.0104	0.1560	594.1379
14545	2017-03-01	0.0144	-0.0155	73.0897
14547	2017-03-01	0.0161	0.0953	1813.6685
14548	2017-03-01	0.0207	0.1277	63.3825

Exercise 5 Using the random forest and elastic net from the previous exercise, machine learning portfolios are created. Using the predicted excess return as the sorting variable. The period breakpoints are the deciles of the predicted excess returns from the two models. The output of the portfolio sorting is the value-weighted monthly excess returns for each decile portfolio. Creating the long-short portfolio as in Gu, Kelly and Xiu (2020), where thw lowest decile is shorted and we go long in the highest decile. Figure 5 shows the results of the two long-short strategies based on random forest and elastic net.

Table 5: Table 5: Long-short strategy

	CAPM alpha	Market beta	Excess return
Random Forest	0.00623	0.46926	0.01187
Elastic Net	-0.00645	-0.02553	-0.00675

Both machine learning portfolios perform worse than the market excess return. The portfolio based on random forest has an excess return of -0.00016, where the portfolio based on the elastic net has a negative excess return of -0.0068. Both portfolios have negative CAPM alphas. It means that the two machine learning portfolios underperform compared to the market.