Exercise 1

OBS: We were not able to make the tuning in the models work, so we have not estimated any models. We have also tried LASSO, Ridge and Elastic Net, but it kept returning errors we could not find a solution to.

1.1

The macro and stock influential characteristics chosen as predictors are presented in the tables below:

Table 1: Influential characteristics

Characteristic	Description	Code
mom1m	short-term reversal	characteristic_mom1m
mvel1	log market equity	characteristic_mvel1
mom12m	stock momentum	characteristic_mom12m
chmom	momentum change	characteristic_chmom
maxret	recent maximum return	characteristic_maxret

Table 2: Macroeconomic Variables

Variable	Description	Code
bm	Book to Market	macro_bm
ntis	Net Equity Expansion	macro_ntis
tbl	T-bill Rate	macro_tbl
tms	Term Spread	macro_tms

The selection of the above stock characteristic are based on Figure 5 in Gu, Kelly and Xiu (2020), in which these five characteristics are ranked highest in terms of overall model contribution. *mom1m*, *mom12m*,*chmom* and *maxret* are based on recent price trends, where as *log market equity* is a measure of market liquidity. The choices of macroeconomic variables are based on Table 4 presented in the same paper. According to their importance measure, all models agree that the aggregated *book-to-market* ratio is a critical predictor. Bond market variables such as the *tbl*, the treasure bill rate, are important for linear and generalized linear models, where as non-linear (machine-learning based) methods place greater emphasis on term spread, *tms*, and net equity expansion, *ntis*. Hence, these predictors should have the most influential power when either estimating with linear or non-linear methods.

Table 3: Summary Statistics

	Mean	Sd	skewness	kurtosis	min	max
mom1m	-0.002	0.604	0.011	-1.309	-0.989	0.99
mvel1	0.033	0.587	-0.06	-1.24	-0.99	1
Bm	0.30	0.05	0.23	-0.57	0.22	0.44
ntis	-0.01	0.02	-0.41	-0.01	-0.05	0.02
tbl	0.01	0.01	1.02	-0.3	1e-04	0.05
tms	0.02	0.01	-0.03	-1.13	-0.00	0.04

1.2

The summary statistic of the most meaningfully variables are presented in the table below. The book-to-market variable has the highest mean on 0,30 while the short-term reversal variable has the lowest mean. Further, the log market equity has the highest max value and the T-bill rate has the lowest.

1.3

We perform a one-hot encoding that mutates column *sic2* into 70 columns which are 1 if *permno* at the given month corresponds to the industry code and zero otherwise. We create this using:

```
datadmy <- data %>% mutate(value = 1) %>% spread(sic2, value, fill = 0 )
```

This gives us a new dataframe with dimensions 744088 × 83.

Exercise 2

The model in Gu, Kelly and Xiu (2020)

$$r_{i,t+1} = E_t\left(r_{i,t+1}\right) + \varepsilon_{i,t+1} \quad \text{where} \quad E_t\left(r_{i,t+1}\right) = g\left(z_{i,t}\right) \tag{A.1}$$

describes an asset's excess return as an additive prediction error model. The potential limitations is that $g(\cdot)$ does not depend on either i nor t. Gu, Kelly and Xiu (2020) the model is flexible as it leverages information from the entire panel which lends stability to estimates of risk premia for any individual assets due to the fact that it maintains the same form across time t and across different stocks i. The linear factor model representation of the expected asset's excess return is written as

$$r_i = a_i + \lambda_{i1} f_1 + \lambda_{i2} f_2 + \cdots + \lambda_{in} f_n + \varepsilon_j \tag{A.2}$$

where a_i is a constant, f_n a systematic factor and λ_{in} the sensitivity of the ith asset to factor n. Here, the expected return of an asset i is a linear function of the asset's sensitivities to the n factors. Thus, the functional form of $g(\cdot)$ would also be linear in this case, but $z_{i,t}$ itself in the factor model would then correspond to the f_n's as these represent factors containing controls, e.g. macroeconomic variables (the state of the economy).

Exercise 3

3.1

The purpose of the "hyperparameter" selection procedure or the choice of tuning parameters is to avoid overfitting a model. More so, the choice of hyperparameter governs the extent of estimator regularization.

The objective function consists of a cost function and regularization function. We want to minimize this in order to enhance prediction accuracy and thus minimize the error term. The objective function depends on the framework, i.e. LASSO, Ridge, Elastic Net etc. The reason you avoid fitting a model or tuning the hyperparameters based on the entire data set is to keep some observations for out-of-sample testing. That way, we are able to evaluate the chosen method's predictive performance.

3.2

The "fixed" model scheme splits the data into training, validation and testing samples. It estimate the model once from the training and validation samples and also attempts to fit all points in the testing sample using this fixed model estimate. Further, the tuning parameters are then selected from the data in a validation sample. An alternative to this is the a "rolling" scheme. The training and validation samples gradually shift forward in time to include more recent data but also holds the total number of time periods in each training and validation sample fixed. The third alternative is the "recursive" performance evaluation scheme. As the rolling approach, it gradually includes more recent observations in the training and validation windows, but always retains the entire history in the training scheme despite the fact its window size gradually increases.

3.3

We divide our data into 3 disjoint time periods. The first sub sample is the "training sub sample". This is used to estimate the model subject to a specific set of tuning parameter values. The second sub sample is the "validation sub sample" which is used for tuning the hyperparameters. The third subsampel is the "testing" sample, which is a out-of sample. It is neither used for estimation nor tuning, but used to evaluate a method's predictive performance.

Exercise 4

We start by implementing the random forest method. Random forest is a ensemble method that combines forecast from a various different trees. It is a variation on a more general procedure known as bootstrap aggregation or also called "bragging". To assure that the individual tress are not the same, we use the bootstrap to induce randomness. Bagging combines and averages multiple models. We begin with creating m bootstrap samples from the training data. The bootstrapped samples allows to create various different data set, but each data set has the same distribution as the overall training set. Then for each of the bootstrap samples we train a single, unpruned regression tree and then averages their forecast. The tress for each individual bootstrap sample tends to be deep and overfit which makes their individual predictions inefficiently variable. When averaging across multiple trees, this reduces the variability of one tree and reduces overfitting, which improves the predictive performance.

To grow a forest, we start with building **B** decision tress T_1,...T_B using the training sample. We then create a bootstrap training set by sampling **N** observations from the training set with replacement. For each observation in the test set from a prediction:

$$\hat{y} = \frac{1}{B} \sum_{i=1}^{B} \hat{y}_{T_i} \tag{A.3}$$

The hyperparameters in the ramdom forest approch is the number for random trees. We initializes in our code a random forest with 50 trees contained in the ensemble where we require at least 20 observations in a node.

The second machine learning method we implement is the elastic net method. Elastic net combines L_1 with L_2 penalization and therefore encourage a grouping effect where strongly correlated predictors tend to be in or out of the model together. This framework considers the following general optimization problem:

$$\hat{\beta}^{EN} = \text{arg min } (Y - X\beta)^{'}(Y - X\beta) + \lambda(1-\rho) \sum_{k=1}^{K} \parallel \beta_k + \frac{1}{2}\lambda\rho \sum_{k=1}^{K} \beta_k^2 \tag{A.4}$$

The elastic net involves two nonnegative hyperparameters, the shrinkage λ and the weighting parameter ρ. The Elastic Net resemble Lasso for ρ=1 and Ridge regression for ρ=0. We then adaptively optimize the tunining parameters, λ and ρ using the validation sample. Our implementation of penalized regression uses the accelerated proximal gradient algorithm and accommodate both the least squares and Huber objective functions.

4.1

We chose the model paramters that optimize the objective function

4.3

Exercise 5

5.1

5.2

5.3