Machine Learning for Finance 2024

Exercise 1

Assistant professor Niclas Meyer

March 25, 2024

Due 7 April 2024 at 23.59pm.

General instructions:

- The Exercises are designed to be completed in groups/teams. The suggested maximum and minimum group size is three persons.
- Please include R scripts and output from running the scripts. You can use for example the R Markdown option in R to output codes and results to a PDF or similar document. Including only the code or only the text in your answers is not sufficient (except for in Question 1).
- The data you need for completing the Exercise is found on the course´s Teams group (General - Files). The file is named *Data_2024.dta*. You can import the data into R using "File - Import Data Sets - From Stata", after saving the data on your computer/OneDrive.
- You will find all the codes you need to complete the Exercise in the course book by James, Witten, Hastie, and Tibshirani (2017).
- Use the code "set.seed(1)" for randomization so that we can replicate them.

1 Empirical questions

Use the data *Data_2024.dta* which can be found on the course's Teams page. The file includes data from COMPUSTAT for North American companies for fiscal years 2005-2020. The dependent variable *trt1m* is the monthly total stock return measured in month m of year t. There are a total of 301 independent (lagged monthly stock return as well as quarterly accounting) variables in the data sets. These are winsorized at the 5th and 95th percentiles, and scaled by total assets.

The raw data includes 1,950,726 observations for 309 variables. The data needs to be cleaned (follow the steps in the order given below):

- First, **remove variables that are not needed**. Keep variables in columns 2, 6, 8, 10-309. This includes the variable *year* which we need to construct training and test sets. You can use the "select" command in R.

- Second, **remove rows with missing observations** for any of independent (or dependent) variables. You can use the "na.omit" command in R. After cleaning the data, the dimensions (*dim* code in R) of the data should be 471,521 observations for 303 variables.

- Third, in all Questions in Exercise 1, use **years 2005-2015 (inclusive) as training set**, and **years 2016-2020 (inclusive) as test set**. Use the "subset" command. The dimensions of the training data set should be 325,505 rows with 303 variables, and for the test data set they should be 146,016 rows with 303 variables.

  In other words:
  Train=Data_2024[2005-2015]
  Test=Data_2024[2016-2020]

1. **Predictions using simple models**

   a) Estimate an *intercept-only* model, where the dependent variable is $trt1m$, and calculate the *test MSE.* That is, estimate the model using the training set (data for years 2005-2015), and test the prediction accuracy using the test set (data for years 2016-2020).

   b) Estimate a *least squares* (OLS) model, where you use $trt1m$ as dependent variable and all the variables in the list above as independent variables (i.e. all independent variables but excluding *year*). Calculate the *test-MSE.*

2. **Predictions using Forward Stepwise Selection**

   a) Estimate a *Forward Selection Model* on the training set to find the variables included in the $\mathcal{M}_1, \ldots, \mathcal{M}_{301}$ models. Report the results (or at least a subset of them) (output from R or Python is sufficient as answer). Which variable enters all the $\mathcal{M}_i$ models?

b) Adjust the training error rate using the BIC (Bayesian Information Criterion) value. Which model of the $\mathcal{M}_i$ models is the "best" according to BIC (i.e. how many variables should we use)?

c) Calculate the test-MSE for the best model chosen using *Forward Selection* and BIC.

d) Use *Forward Stepwise Selection* and *10-fold cross-validation* on the training set to decide which of the $\mathcal{M}_1, \ldots, \mathcal{M}_{40}$ models is the best model. **Note! Set nvmax=40**, otherwise it will take a long time to estimate the models. You can find the codes for cross-validation in the course book.

e) Calculate the test-MSE for the model you have chosen using *Forward Stepwise Selection* and cross-validation in step 3d) using the test set.

3. **Predictions using Backward Stepwise Selection**

a) Use *Backward Stepwise Selection* to find the variables included in the $\mathcal{M}_1, \ldots, \mathcal{M}_{40}$ models. Use *Backward Stepwise Selection* and *10-fold cross-validation* on the training set to decide which of the $\mathcal{M}_1, \ldots, \mathcal{M}_{40}$ models is the best model. **Note! Set nvmax=40**, otherwise it will take a long time to estimate the models. Report the output (from R or Python) as well as the the best model.

b) Report the coefficients for the variables included in the final best model that you have chosen using *Backward Stepwise Selection* in step 4b). Use the full training set. **Set nvmax=40.**

c) Calculate the test-MSE for the model you have chosen in 3b) using *Backward Stepwise Selection* and *10-fold cross-validation* using the test set.

4. **Predictions using Lasso**

a) Estimate a *Lasso regression* using the training set with tuning parameter $\lambda = 0.5$. Show the coefficients for this specification (use the full training set sample).

b) Then use a grid of lambda values (using e.g. the method in James, Witten, Hastie, and Tibshirani (2017)). Show a plot of the coefficients (y-axis) and log lambdas (x-axis), or of the coefficients (y-axis) and *L1 norm* (x-axis), or similar.

c) Use *10-fold cross-validation* on the training set to find the tuning parameter that yields the model with the lowest MSE. Report this *best* tuning parameter, and show a plot of how the MSE depends on the value of the tuning parameter (R produces this plot automatically).

d) Using the best (lambda)model chosen using *10-fold cross-validation* in 4c), calculate the test-MSE using the test set.

5. **Predictions using Ridge**

a) Use *10-fold cross-validation* and *Ridge regression* on the training set to find the tuning parameter that yields the model with the lowest MSE. Report this *best* tuning parameter, and show a plot of how the MSE depends on the value of the tuning parameter.

b) Using the best (lambda)model chosen using *10-fold cross-validation* and *Ridge regression* in 5a), calculate the test-MSE using the test set.

6. **Results: Predictions**

a) Compare the test-MSEs for all the models in this Exercise to each other! Which machine learning model (ML) ($Forward\,Stepwise\,Selection$, $Backward\,Stepwise\,Selection$, *Lasso*, or *Ridge*) yields the best prediction of the test sample? Which one yields the worst?

b) Do the ML methods beat the $intercept-only$ and the OLS models in prediction? Why, why not?

7. **Best subset selection method and cross-validation**
Because the sample is so large and includes so many independent variables, we used Forward and Backward Stepwise selection models instead of the Best Subset Selection method.

a) Estimate a *Best subset selection* method (”exhaustive”) for up to the best four-variable models, i.e. **set nvmax=4**! Otherwise, it will take an extremely long time to estimate the model. Which are the variables included in the best model with four variables? Show the R output in your answer.

8. **Predictions using Ridge in Python**

a) Based on python *Lasso* example of predicting *wage* presented in the lecture, similarly, using *Ridge* model to predict *wage* with the same data uploaded on Moodle. In your answers, at least do the following tasks: examine the relationship between ridge coefficient and lambda; use K-fold cross validation to fine tune and choose the best hyperparameters and the best model. What can you conclude about the prediction accuracy of linear OLS model, Ridge and Lasso? The question must be answered in Python.

Hint: you can use Ridge from Sklearn package, please refer to the links below:

https://www.tutorialspoint.com/scikit_learn/scikit_learn_ridge_regression.htm

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html