Data Science Methods in Finance

Take Home Exam

Jens Kvaerner, Ole Wilms

Submission deadline: April 6, 2022 at 23.59

Important Instructions

- This is an **individual assignment** and hence, you must work on it by yourself.
- Copying codes from others will be counted as fraud and lead to a failure of the exam as well as a report to the exam committee.
- You are allowed to use ment.io to discuss questions with other students. However, the questions must be about the general task of a problem. For example asking other students to look for errors in your code is not allowed.
- To submit your final report and code, please use the corresponding Assignment on Canvas.
- **Late submissions will not be considered and result in a grade of 0.**
- **Please do NOT submit datafiles. Only use the original datafile from Canvas and upload your codes and report.**
- The code must run without errors and generate the same solutions as you have reported.
- The data you will use is “crsp_data_exam2022.RDS" available on Canvas by April 4.
- The exam has a total of 5 questions with multiple sub-questions which account for a total of 100 points:

Question	1	2	3	4	Total
Points	35	25	20	20	100

- The bonus points you earned for the participation on ment.io will be added to the points you achieve in this exam (maximum 10).
- We care about the layout, and we want a “consultant" professional report. That means you should prepare a set of slides, and not a long document (more on this below). We give up to 5 bonus points for good layouts.

General Information

For this assignment, please create a report in power point or with Beamer (we recommend overleaf) where you summarize the results. Each item should be answered on a different slide. Only if it is mentioned otherwise in the subquestion or if you think it is really necessary, you can use more slides. Try to be precise. Ambiguous and overly long answers will lead to a reduction of points. Put additional explanations or calculations in the appendix. Start all headers with the question you are answering. For example, when presenting summary statistics for the first question, write “Q1.1: Summary statistics", etc. Make sure that the font size is well readable (not too small) and that you do not put too much information (text) on one slide. You might want to use wide format.[1]

[1] The template “example_format.pdf" on Canvas is a good starting point. We strongly encourage you to follow this structure as close as possible.

Dataset and Setup

In this exam you are asked to predict US stock returns using return data from the Center for Research in Security Prices (CRSP). The dataset (“crsp_data_exam2022.RDS") is similar to the data used in Tutorial 2 and starts in 1989. It contains firm level data on stock returns including dividends (ret), company identifiers (permno), stock prices (prc) and the number of shares (shrout). The dataset is preprocessed such that it doesn’t contain any missing values and prices are already adjusted to be positive. Your task is to predict one-month ahead cross-sectional stock returns using features that are constructed from past returns. That is, you use company level data from period t to predict the return of the company in period $t+1$ as in Gu et al. (2020). Furthermore you are asked to build a trading strategy based on the predictions of your machine learning model.

Question 1 (35 points): Construct features and preprocess data

The first task is to construct features based on past stock returns. Let $R_{i,t}$ denote the stock return of company i in period t (variable *ret* in the dataset).

1.1 Construct the variable short-term reversal which is given by the one-month lagged return. We call this variable $\mathit{mom1}_{i,t}$:

$$\mathit{mom1}_{i,t} = R_{i,t-1}$$

The next step is to construct several momentum variables which leave out the short term reversal. We will call these variables $\mathit{mom2\text{-}1}_{i,t}$, $\mathit{mom3\text{-}1}_{i,t}$, $\ldots$ $\mathit{mom12\text{-}1}_{i,t}$. They are constructed as follows:

$$\begin{aligned}
\mathit{mom2\text{-}1}_{i,t} &= (1+R_{i,t-2})-1 \\
\mathit{mom3\text{-}1}_{i,t} &= (1+R_{i,t-2})(1+R_{i,t-3})-1 \\
\mathit{mom4\text{-}1}_{i,t} &= (1+R_{i,t-2})(1+R_{i,t-3})(1+R_{i,t-4})-1 \\
&\vdots \\
\mathit{mom12\text{-}1}_{i,t} &= (1+R_{i,t-2})(1+R_{i,t-3})(1+R_{i,t-4})\ldots(1+R_{i,t-12})-1
\end{aligned}$$

Provide a table containing the mean and number of missing observations for each feature.

1.2 Impute missing features with their cross-sectional median as follows: (i) for each time t calculate the cross-sectional median for each stock-level predictive characteristic, (ii) check whether the stock-level predictive characteristic is missing in that period, and (iii) replace the stock-level predictive characteristic in that period with its cross-sectional median if it is missing. This problem can be solved for all stock-level predictive characteristics at once by using the mutate_at function from dplyr. If you cannot solve this question, simply remove all observations (rows) that contain missing values. If you do so, please highlight this in your presentation. Provide the mean and number of missing observations for each feature. You can add these numbers to the table from Question 1.1.

1.3 Replace the stock-level predictive characteristics with 0 if missing after the previous steps and provide the mean and number of missing observations for each feature. You can add these numbers to the table from Question 1.1/1.2.

1.4 Normalize all features between -1 and 1 in the cross-section. So at each time t you compute the minimum and maximum values of the feature over all firms and normalize your features as follows:

$$\tilde{x}_{i,t} = \frac{2 \times (x_{i,t} - \min_t(x_{i,t}))}{(\max_t(x_{i,t}) - \min_t(x_{i,t}))} - 1$$

where $x_{i,t}$ denotes the feature for firm i in period t, $\tilde{x}_{i,t}$ is the normalized feature and $\min_t(x_{i,t})$ is the minimum of that feature over all firms for a given period t (note that I omitted the index to allow for different features to simplify notation). If you cannot solve this question, simply scale all features such that they have mean 0 and standard deviation 1. If you do so, please highlight this in your presentation. Provide the mean and standard deviation for each feature.

1.5 Split the dataset into training (used for estimating the model), validation (used for tuning the hyperparameters), and testing (used for evaluating the performance) data. The outcomes are given by the firm level returns and please use all twelve features created in Question 1.1 and 1.2. The training dataset ranges from 1990-01-01 up to and including 1999-12-31, the validation dataset ranges from 2000-01-01 up to and including 2007-06-30. As it is well known that many return predictors did not work well during the financial crisis, we will consider two test samples. One including the financial crisis ranging from 2007-07-01 up to and including 2021-12-31 and one excluding the financial crisis ranging from 2010-01-01 up to and including 2021-12-31. Provide the average monthly return in the training, validation and the two test data sets.

1.6 Compute and report the correlation matrix of the data in the training sample and interpret the findings. Is there anything worrisome about the features?

You might want to look at other summary statistics as well to obtain more information about the data but you do not have to report these in the write-up.

Question 2 (25 points): Linear models

2.1 Fit a linear regression model in the training data (we simply pool all observations) using all 12 features:

$$R_{i,t} = \beta_0 + \beta_1 \mathit{mom1}_{i,t-1} + \beta_2 \mathit{mom2\text{-}1}_{i,t-1} + \ldots + \beta_{12} \mathit{mom12\text{-}1}_{i,t-1} + \epsilon_{i,t}$$

Report the regression coefficients including t-statistics. Compute the R-Squared from Gu et al. (2020) for the training and validation data as well as the two test datasets. It is given by

$$R^2_{Gu} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i)^2}.$$

You can use the function in *R2_ Gu.R* available on Canvas to compute it. The function takes outcomes y_i and predictions $\hat{y}_i$ from any machine learning model as inputs and returns R^2_{Gu} as its output. Interpret your findings.

2.2 Next we use lasso regressions to improve the predictive power. Use the training sample to fit your lasso model and tune the hyperparamter λ by using the R^2_{Gu} in the validation data. For this, set up a (large) grid for λ. Fit the model for a given λ using the training data. Use the fitted model to compute R^2_{Gu} in the validation data and repeat this exercise for all values of λ. Choose the model with the highest R^2_{Gu} in the validation data as your final model. Report a figure that plots R^2_{Gu} against λ as well as the optimal value for λ.

2.3 Report the regression coefficients including significance levels of your final model and interpret your findings.

2.4 Compute the R^2_{Gu} for the final lasso model in the two test datasets and compare to the linear model from Question 2.2. Interpret your findings.

Question 3 (20 points): Portfolios

In this exercise, you are asked to use the predictions from Question 2 to construct portfolios and compute returns of your trading strategy.

3.1 Use your predicted returns from the lasso model to form portfolios. That is, construct a portfolio that takes a long position in the stocks that are in the top 20% of the distribution of the predicted returns in a specific month. Put differently you go long in the 20% of stocks with the with the highest predicted returns. Take a short position in the stocks that are in the top 20% of the distribution of the predicted returns in the same month. We recommend that you use the quantile function to create the cut-off points you need to allocate stocks into different portfolios.

Compute the portfolio returns by taking the average (equal-weighted) return over all stocks in the respective portfolio. Report the average monthly return for both, the long and the short portfolio in the two test datasets.

3.2 Create the "factor" as the return of a long-short portfolio strategy. That is, you buy the long portfolio and sell the short portfolio. Compute the mean and standard deviation of this strategy for both test datasets. Interpret your findings.

3.3 Show that this strategy delivers a positive alpha relative to the Capital Asset Pricing Model (CAPM). For this, you first need to compute the value-weighted market return which you can simply do using the codes from Tutorial 2, Question 3 (available on Canvas). Regress the return of the long-short portfolio on the market return (and a constant) and report the regression outcomes including significance levels. Interpret your findings.

Question 4 (20 points): Neural Networks

In this exercise, you are asked to fit a neural network to predict returns and build a trading strategy based on those predictions.

4.1 Build a nwetwork with one hidden layer and 4 neurons. Use Relu for the activation function of the hidden layer. In the slides, please provide the architecture of your network (simply provide the code that you use to set up the your *keras_ model_ sequential*) and motivate the choice of activation function for the output layer as well as your choice of loss function. List the main advantage and disadvantage of neural nets over the lasso approach used in Question 2. Note that you can solve this exercise without actually implementing and fitting the network.

4.2 Train the network for 5 epochs using a batch size of 200 and report the mean squared error in the training data. Also report the R^2_{Gu} in the training, validation and test data and interpret your findings.

Note: If you run into memory/computing time problems here, only use half of the training data starting in 1995-01-01.

4.3 To prevent the network from overfitting, add a kernel regularizer with a lasso penalty (l1) and $\lambda = 0.001$ to the hidden layer. Again, train the new network for 5 epochs using a batch size of 200. Report the mean squared error in the training data as well as the R^2_{Gu} in the training, validation and test data. Interpret your findings.

4.4 Use the predictions from this neural network to build a trading strategy as described in Question 3.1-3.2 (you can reuse the same code). Compute the mean and standard deviation of this strategy for both test datasets and compare it to the strategy based on the lasso predictions.

References

Gu, S., B. Kelly, and D. Xiu (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies 33*(5), 2223–2273.