Project 1 - Statistical learning

August Jonasson

2024-12-01

Throughout this report, none of the actual R code is visible. Refer to code appendix to see which specific packages were used and how the coding was done. We will however plot several model summaries throughout the report, which makes readability somewhat worse, but as we believe they are relevant, we deemed this as a necessary sacrifice.

Task 1: Linear regression

(a)

Fitting a linear regression model with the stock log-returns as predictors and the log-return of the capital index as response. Below is the model summary printed out.

```
##
## Call:
## lm(formula = rOMX ~ ., data = returns)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0082894 -0.0006629  0.0000542  0.0008093  0.0076761
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.400e-04  3.624e-05  -3.864 0.000117 ***
## rABB         3.372e-02  3.836e-03   8.789  < 2e-16 ***
## rNDA.SE.ST   4.936e-02  3.619e-03  13.642  < 2e-16 ***
## HM_B.ST      4.886e-02  1.874e-03  26.070  < 2e-16 ***
## ATCO_A.ST    8.173e-02  8.625e-03   9.475  < 2e-16 ***
## ERIC_B.ST    5.023e-02  2.064e-03  24.330  < 2e-16 ***
## ESSITY_B.ST  3.229e-02  2.899e-03  11.141  < 2e-16 ***
## SAND.ST      4.661e-02  3.533e-03  13.192  < 2e-16 ***
## BOL.ST       1.798e-02  2.055e-03   8.750  < 2e-16 ***
## GETI_B.ST    1.733e-02  1.778e-03   9.745  < 2e-16 ***
## ALFA.ST      2.981e-02  2.752e-03  10.834  < 2e-16 ***
## ATCO_B.ST    3.098e-02  8.724e-03   3.552 0.000395 ***
## VOLV_B.ST    6.517e-02  3.403e-03  19.151  < 2e-16 ***
## SHB_A.ST     3.686e-02  3.735e-03   9.868  < 2e-16 ***
## ELUX_B.ST    1.174e-02  2.131e-03   5.509 4.26e-08 ***
## SEB_A.ST     4.341e-02  4.213e-03  10.304  < 2e-16 ***
## ASSA_B.ST    6.293e-02  3.276e-03  19.208  < 2e-16 ***
## AZN.ST       3.545e-02  2.664e-03  13.305  < 2e-16 ***
## SWED_A.ST    4.685e-02  3.293e-03  14.228  < 2e-16 ***
## TELIA.ST     2.922e-02  3.462e-03   8.440  < 2e-16 ***
## TEL2_B.ST    1.761e-02  2.843e-03   6.195 7.55e-10 ***
## SBB_B.ST     3.117e-03  1.004e-03   3.104 0.001947 **
```

```
## INVE_B.ST    1.074e-01  4.983e-03  21.559  < 2e-16 ***
## SINCH.ST     9.177e-03  9.677e-04   9.484  < 2e-16 ***
## SCA_B.ST     1.678e-02  2.641e-03   6.356 2.76e-10 ***
## HEXA_B.ST    5.642e-02  2.860e-03  19.728  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001399 on 1475 degrees of freedom
## Multiple R-squared:  0.9866, Adjusted R-squared:  0.9864
## F-statistic:  4346 on 25 and 1475 DF,  p-value: < 2.2e-16
```

According to the p-values of the above summary, none of the features are insignificant in their ability to predict the log-returns on the capital market index, i.e. they all have influence on the capital market index. This is not surprising at all, since we chose our predictors as the most important stocks on the market. The capital market index is modeled after these stocks. Also, the effects (coefficient estimates) of all features are very similar, i.e. no particular one feature stands out as having more or less of an impact.

By significance, we mean that under the null-hypothesis: that said coefficient has no effect on the response while keeping the others constant, the observed value would be less than 5 % likely to occur (95 % significance level). For all of our coefficients, this probability is well below 5 % and for most of them, this probability is more or less zero.

(b)

No, the results from part (a) cannot be used to answer the question of which of the stocks have to be included in the model in order to mimic the behavior of the Swedish capital market index. We have not yet validated the model, and as such we have no idea how it will perform on actual test data. It could for example be that the model is heavily overfitted.

If some of our variables weren't showing significance, we might be worried about potentially missing some joint significances, i.e. features that jointly show significance - as the significance levels we examined in the print-out above are marginal significances. However, as all of our variables already show significance, we do not have to worry about this.

(c)

Now using the forward selection in order to select the model. This is done by initially only using an intercept and the response variable, and then iteratively adding whichever feature would yield the most significance until no further improvement is seen.

```
##
## Call:
## lm(formula = rOMX ~ rABB + rNDA.SE.ST + HM_B.ST + ATCO_A.ST +
##     ERIC_B.ST + ESSITY_B.ST + SAND.ST + BOL.ST + GETI_B.ST +
##     ALFA.ST + ATCO_B.ST + VOLV_B.ST + SHB_A.ST + ELUX_B.ST +
##     SEB_A.ST + ASSA_B.ST + AZN.ST + SWED_A.ST + TELIA.ST + TEL2_B.ST +
##     SBB_B.ST + INVE_B.ST + SINCH.ST + SCA_B.ST + HEXA_B.ST, data = returns)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0082894 -0.0006629  0.0000542  0.0008093  0.0076761
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.400e-04  3.624e-05  -3.864 0.000117 ***
## rABB         3.372e-02  3.836e-03   8.789  < 2e-16 ***
## rNDA.SE.ST   4.936e-02  3.619e-03  13.642  < 2e-16 ***
```

```
## HM_B.ST      4.886e-02  1.874e-03  26.070  < 2e-16 ***
## ATCO_A.ST    8.173e-02  8.625e-03   9.475  < 2e-16 ***
## ERIC_B.ST    5.023e-02  2.064e-03  24.330  < 2e-16 ***
## ESSITY_B.ST  3.229e-02  2.899e-03  11.141  < 2e-16 ***
## SAND.ST      4.661e-02  3.533e-03  13.192  < 2e-16 ***
## BOL.ST       1.798e-02  2.055e-03   8.750  < 2e-16 ***
## GETI_B.ST    1.733e-02  1.778e-03   9.745  < 2e-16 ***
## ALFA.ST      2.981e-02  2.752e-03  10.834  < 2e-16 ***
## ATCO_B.ST    3.098e-02  8.724e-03   3.552 0.000395 ***
## VOLV_B.ST    6.517e-02  3.403e-03  19.151  < 2e-16 ***
## SHB_A.ST     3.686e-02  3.735e-03   9.868  < 2e-16 ***
## ELUX_B.ST    1.174e-02  2.131e-03   5.509 4.26e-08 ***
## SEB_A.ST     4.341e-02  4.213e-03  10.304  < 2e-16 ***
## ASSA_B.ST    6.293e-02  3.276e-03  19.208  < 2e-16 ***
## AZN.ST       3.545e-02  2.664e-03  13.305  < 2e-16 ***
## SWED_A.ST    4.685e-02  3.293e-03  14.228  < 2e-16 ***
## TELIA.ST     2.922e-02  3.462e-03   8.440  < 2e-16 ***
## TEL2_B.ST    1.761e-02  2.843e-03   6.195 7.55e-10 ***
## SBB_B.ST     3.117e-03  1.004e-03   3.104 0.001947 **
## INVE_B.ST    1.074e-01  4.983e-03  21.559  < 2e-16 ***
## SINCH.ST     9.177e-03  9.677e-04   9.484  < 2e-16 ***
## SCA_B.ST     1.678e-02  2.641e-03   6.356 2.76e-10 ***
## HEXA_B.ST    5.642e-02  2.860e-03  19.728  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001399 on 1475 degrees of freedom
## Multiple R-squared:  0.9866, Adjusted R-squared:  0.9864
## F-statistic:  4346 on 25 and 1475 DF,  p-value: < 2.2e-16
```

According to the forward selection method, all stocks should be included in the model.

(d)

```
##
## Call:
## lm(formula = rOMX ~ rABB + rNDA.SE.ST + HM_B.ST + ATCO_A.ST +
##     ERIC_B.ST + ESSITY_B.ST + SAND.ST + BOL.ST + GETI_B.ST +
##     ALFA.ST + ATCO_B.ST + VOLV_B.ST + SHB_A.ST + ELUX_B.ST +
##     SEB_A.ST + ASSA_B.ST + AZN.ST + SWED_A.ST + TELIA.ST + TEL2_B.ST +
##     SBB_B.ST + INVE_B.ST + SINCH.ST + SCA_B.ST + HEXA_B.ST, data = returns)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0082894 -0.0006629  0.0000542  0.0008093  0.0076761
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.400e-04  3.624e-05  -3.864 0.000117 ***
## rABB         3.372e-02  3.836e-03   8.789  < 2e-16 ***
## rNDA.SE.ST   4.936e-02  3.619e-03  13.642  < 2e-16 ***
## HM_B.ST      4.886e-02  1.874e-03  26.070  < 2e-16 ***
## ATCO_A.ST    8.173e-02  8.625e-03   9.475  < 2e-16 ***
## ERIC_B.ST    5.023e-02  2.064e-03  24.330  < 2e-16 ***
## ESSITY_B.ST  3.229e-02  2.899e-03  11.141  < 2e-16 ***
```

```
## SAND.ST      4.661e-02  3.533e-03  13.192  < 2e-16 ***
## BOL.ST       1.798e-02  2.055e-03   8.750  < 2e-16 ***
## GETI_B.ST    1.733e-02  1.778e-03   9.745  < 2e-16 ***
## ALFA.ST      2.981e-02  2.752e-03  10.834  < 2e-16 ***
## ATCO_B.ST    3.098e-02  8.724e-03   3.552 0.000395 ***
## VOLV_B.ST    6.517e-02  3.403e-03  19.151  < 2e-16 ***
## SHB_A.ST     3.686e-02  3.735e-03   9.868  < 2e-16 ***
## ELUX_B.ST    1.174e-02  2.131e-03   5.509 4.26e-08 ***
## SEB_A.ST     4.341e-02  4.213e-03  10.304  < 2e-16 ***
## ASSA_B.ST    6.293e-02  3.276e-03  19.208  < 2e-16 ***
## AZN.ST       3.545e-02  2.664e-03  13.305  < 2e-16 ***
## SWED_A.ST    4.685e-02  3.293e-03  14.228  < 2e-16 ***
## TELIA.ST     2.922e-02  3.462e-03   8.440  < 2e-16 ***
## TEL2_B.ST    1.761e-02  2.843e-03   6.195 7.55e-10 ***
## SBB_B.ST     3.117e-03  1.004e-03   3.104 0.001947 **
## INVE_B.ST    1.074e-01  4.983e-03  21.559  < 2e-16 ***
## SINCH.ST     9.177e-03  9.677e-04   9.484  < 2e-16 ***
## SCA_B.ST     1.678e-02  2.641e-03   6.356 2.76e-10 ***
## HEXA_B.ST    5.642e-02  2.860e-03  19.728  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001399 on 1475 degrees of freedom
## Multiple R-squared:  0.9866, Adjusted R-squared:  0.9864
## F-statistic:  4346 on 25 and 1475 DF,  p-value: < 2.2e-16
```

The backward selection yields the same result as the forward selection.

(e)

First, we fit the ridge regression model to the data (see code appendix for which package is used). Next, in order to choose the best λ we will use leave-one-out cross-validation. From this we can extract the best λ as

```
## [1] 0.00104933
```

We can now print the coefficient estimates against the tested values on λ. We also include a red, dashed vertical line that indicates the optimal value on λ found by the cross-validation. As can be seen from Figure 1, the λ that we found results in next to no shrinkage at all.

Fitting a new ridge model using this best λ we can compare the coefficient estimates to the simple linear regression model from the first task and see that they are more or less identical. As such, the ridge regression model is the same as the normal regression model, for which we have already deduced that these results are not enough to say which stocks definitely have to be included in order to mimic the Swedish capital market index.

```
## 25 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## rABB         0.039066278
## rNDA.SE.ST   0.047695457
## HM_B.ST      0.045931983
## ATCO_A.ST    0.059198300
## ERIC_B.ST    0.047216699
## ESSITY_B.ST  0.032029931
## SAND.ST      0.047294687
## BOL.ST       0.018790460
## GETI_B.ST    0.017646029
## ALFA.ST      0.032533178
```

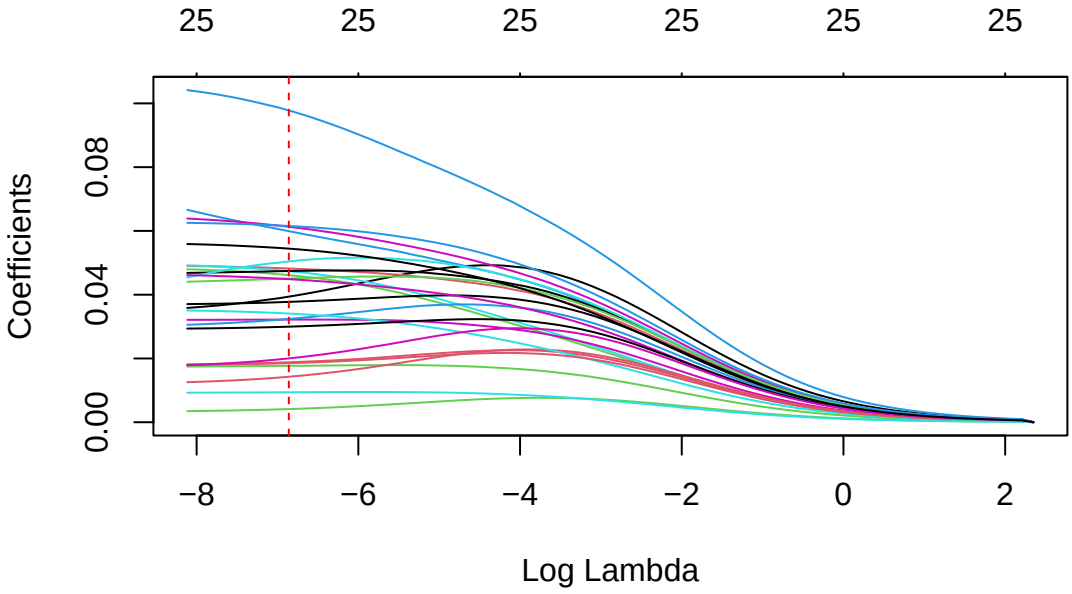

Figure 1: Ridge regression coefficients against values on log-lambda.

```
## ATCO_B.ST   0.050948995
## VOLV_B.ST   0.061424079
## SHB_A.ST    0.037806800
## ELUX_B.ST   0.014343230
## SEB_A.ST    0.045263887
## ASSA_B.ST   0.061729409
## AZN.ST      0.034144881
## SWED_A.ST   0.045026216
## TELIA.ST    0.030028568
## TEL2_B.ST   0.018556346
## SBB_B.ST    0.004146002
## INVE_B.ST   0.098201973
## SINCH.ST    0.009399505
## SCA_B.ST    0.020118011
## HEXA_B.ST   0.054480328
```

(f)

Using the Lasso regression and performing the same steps as in the ridge regression task, we get the best λ as

```
## [1] 3.95066e-05
```

Again, let us examine the coefficients of the model that uses this best lambda for the lasso regression. Again, we can see from Figure 2 that next to no shrinkage has been applied and that all 25 stocks are still included in the model. The answer will therefore be the same here as for all the previous answers. Below are the resulting coefficient estimates using the best λ, which again can be compared to previous models to see that they are unchanged.

```
## 25 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
```

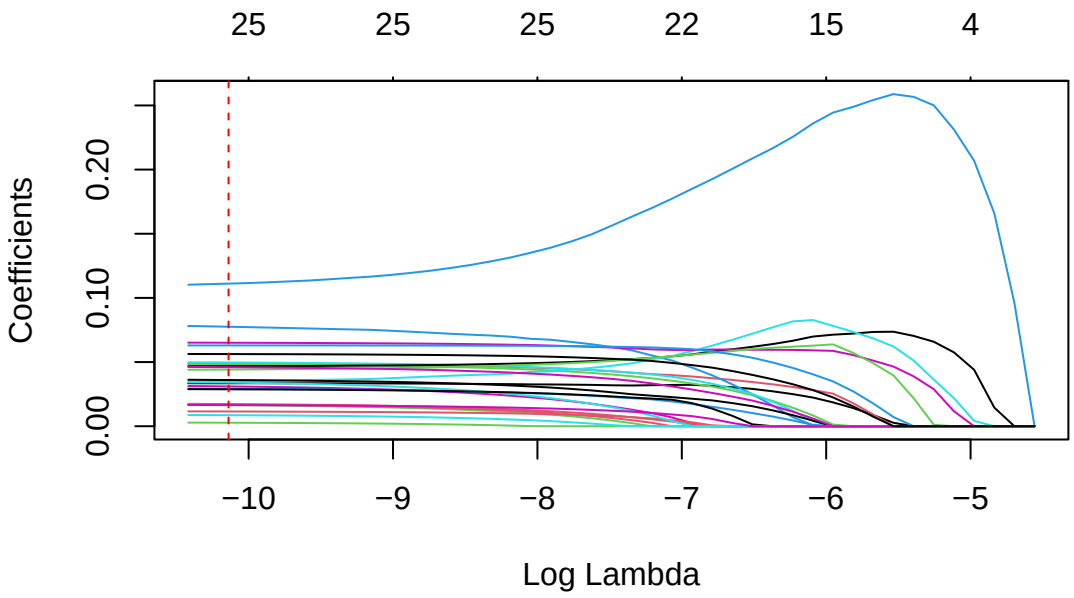

Figure 2: Lasso regression coefficients against values on log-lambda.

```
## rABB         0.033158335
## rNDA.SE.ST   0.048654667
## HM_B.ST      0.048171614
## ATCO_A.ST    0.080077455
## ERIC_B.ST    0.049677332
## ESSITY_B.ST 0.031029946
## SAND.ST      0.047054995
## BOL.ST       0.017319318
## GETI_B.ST    0.016449536
## ALFA.ST      0.029394106
## ATCO_B.ST    0.032348573
## VOLV_B.ST    0.065019177
## SHB_A.ST     0.036408771
## ELUX_B.ST    0.011569412
## SEB_A.ST     0.043738152
## ASSA_B.ST    0.062981027
## AZN.ST       0.034039375
## SWED_A.ST    0.046298408
## TELIA.ST     0.028847738
## TEL2_B.ST    0.016883898
## SBB_B.ST     0.002765300
## INVE_B.ST    0.111117268
## SINCH.ST     0.008621693
## SCA_B.ST     0.016465694
## HEXA_B.ST    0.056062706
```

(g)

According to the results from all of the previous tasks, we can deduce that all stocks should be included in order to mimic the Swedish capital market index as well as possible. Refer to each of the above sub-task model summaries for the respective regression coefficients.

Task 2: Linear classification

(a)

We start by splitting the data into 80/20 (preserving chronology) and also convert our response variable to binary categorical such that if the return on capital market index on day t is positive we assign 1, and 0 otherwise.

Since the purpose of this task is to train two different models - one using all of the 25 stocks as predictors and the other only using the subset of stocks selected by the lasso regression in the previous task - but our lasso also selected all of the stocks as the best model, we decide to arbitrarily remove half of the coefficients for the second model. We remove the half which has the lowest coefficient estimates, as these should affect the model the least. We do this in order for the task at hand to be compatible with us.

The dimensions of the training and test data after the split:

```
## [1] 1201   26
```

```
## [1] 300  26
```

We can now fit the two models and print out their corresponding summaries.

Starting with the model using all of the stocks as predictors. From the below print-out we can now see that all predictors are no longer significant. This indicates that we should consider dropping the insignificant variables from the model. Either directly or through some of the selection methods used in the previous task.

```
##
## Call:
## glm(formula = rOMX ~ ., family = binomial, data = train_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.2520     0.1806  -1.395 0.162952
## rABB         75.7898    25.9477   2.921 0.003491 **
## rNDA.SE.ST   66.1452    16.3720   4.040 5.34e-05 ***
## HM_B.ST      62.5701    11.8672   5.273 1.35e-07 ***
## ATCO_A.ST    71.8039    44.1095   1.628 0.103555
## ERIC_B.ST    76.8052    13.9773   5.495 3.91e-08 ***
## ESSITY_B.ST  25.0554    16.3415   1.533 0.125216
## SAND.ST      41.1655    20.2616   2.032 0.042184 *
## BOL.ST       34.3595    12.2223   2.811 0.004935 **
## GETI_B.ST    32.8153    11.6140   2.825 0.004721 **
## ALFA.ST      55.0731    16.1454   3.411 0.000647 ***
## ATCO_B.ST    64.6164    41.8572   1.544 0.122653
## VOLV_B.ST    70.1201    19.1958   3.653 0.000259 ***
## SHB_A.ST     60.6144    20.7080   2.927 0.003421 **
## ELUX_B.ST    34.8463    15.3830   2.265 0.023497 *
## SEB_A.ST     78.8468    24.6824   3.194 0.001401 **
## ASSA_B.ST    66.8203    20.2082   3.307 0.000944 ***
## AZN.ST       68.4038    14.2453   4.802 1.57e-06 ***
## SWED_A.ST    57.0167    19.0162   2.998 0.002715 **
## TELIA.ST     43.1229    16.5310   2.609 0.009091 **
```

```
## TEL2_B.ST    18.7656    17.2242   1.089 0.275939
## SBB_B.ST     -8.4204     8.7810  -0.959 0.337589
## INVE_B.ST   154.9253    30.8342   5.024 5.05e-07 ***
## SINCH.ST     14.9996     5.3101   2.825 0.004732 **
## SCA_B.ST     32.8206    14.2964   2.296 0.021692 *
## HEXA_B.ST    68.2770    19.0559   3.583 0.000340 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1660.50  on 1200  degrees of freedom
## Residual deviance:  233.32  on 1175  degrees of freedom
## AIC: 285.32
##
## Number of Fisher Scoring iterations: 10
```

Now moving on to the model that only uses the half of the stocks that yielded the largest coefficient estimates in the initial linear model. From this model print-out we can see that each stock is regarded as significant.

```
##
## Call:
## glm(formula = rOMX ~ ., family = binomial, data = train_data[c("rOMX",
##     subset_stocks)])
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1079     0.1368  -0.789 0.430303
## INVE_B.ST   148.9714    23.4481   6.353 2.11e-10 ***
## ATCO_A.ST   127.7869    17.7138   7.214 5.43e-13 ***
## VOLV_B.ST    61.2188    13.5019   4.534 5.79e-06 ***
## ASSA_B.ST    70.5982    16.7696   4.210 2.55e-05 ***
## HEXA_B.ST    67.8489    13.9916   4.849 1.24e-06 ***
## ERIC_B.ST    55.8617     9.8106   5.694 1.24e-08 ***
## rNDA.SE.ST   50.9674    14.3683   3.547 0.000389 ***
## HM_B.ST      47.9623     9.0661   5.290 1.22e-07 ***
## SWED_A.ST    24.8381    11.4942   2.161 0.030701 *
## SAND.ST      54.1852    14.0494   3.857 0.000115 ***
## SEB_A.ST     76.4457    21.3346   3.583 0.000339 ***
## SHB_A.ST     38.8053    17.4656   2.222 0.026296 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1660.5  on 1200  degrees of freedom
## Residual deviance:  362.5  on 1188  degrees of freedom
## AIC: 388.5
##
## Number of Fisher Scoring iterations: 9
```

(b)

Now moving on to the prediction using the models on the same test data.

The misclassification proportion of the full model is:

```
## [1] 0.05666667
```

and the misclassification proportion of the subset model is:

```
## [1] 0.07333333
```

(c)

The conclusions drawn from the results on the accuracy of the models is that the full model performed slightly better than the one using only half of the predictors. As we removed predictors somewhat arbitrarily from the latter model, this is not very surprising. What can be said about both models, however, is that they both tested very well on unseen data, i.e. that the logistic model based on the top stocks on the Swedish market form relatively strong evidence of whether or not the capital market index will go up or down on a given day.

Code appendix

```
library(tidyverse)
library(quantmod)
library(corrplot)
library(glmnet)
library(magrittr)
tickers = c("^OMX",
            "ABB.ST",
            "NDA-SE.ST",
            "HM-B.ST",
            "ATCO-A.ST",
            "ERIC-B.ST",
            "ESSITY-B.ST",
            "SAND.ST",
            "BOL.ST",
            "GETI-B.ST",
            "ALFA.ST",
            "ATCO-B.ST",
            "VOLV-B.ST",
            "SHB-A.ST",
            "ELUX-B.ST",
            "SEB-A.ST",
            "ASSA-B.ST",
            "AZN.ST",
            "SWED-A.ST",
            "TELIA.ST",
            "TEL2-B.ST",
            "SBB-B.ST",
            "INVE-B.ST",
            "SCA-B.ST",
            "HEXA-B.ST",
            "SINCH.ST")

getSymbols(tickers, src="yahoo", from = '2017-12-31', to = "2023-12-31")

# adjusting "bad" names
NDA_SE.ST = `NDA-SE.ST`
HM_B.ST = `HM-B.ST`
ATCO_A.ST = `ATCO-A.ST`
ERIC_B.ST = `ERIC-B.ST`
```

```
ESSITY_B.ST = `ESSITY-B.ST`
GETI_B.ST = `GETI-B.ST`
ATCO_B.ST = `ATCO-B.ST`
VOLV_B.ST = `VOLV-B.ST`
SHB_A.ST = `SHB-A.ST`
ELUX_B.ST = `ELUX-B.ST`
SEB_A.ST = `SEB-A.ST`
ASSA_B.ST = `ASSA-B.ST`
SWED_A.ST = `SWED-A.ST`
TEL2_B.ST = `TEL2-B.ST`
SBB_B.ST = `SBB-B.ST`
INVE_B.ST = `INVE-B.ST`
NDA_SE.ST = `NDA-SE.ST`
SCA_B.ST = `SCA-B.ST`
HEXA_B.ST = `HEXA-B.ST`

OMX_ad = OMX$OMX.Adjusted
ABB.ST_ad = ABB.ST$ABB.ST.Adjusted
NDA_SE.ST_ad = NDA_SE.ST$`NDA-SE.ST.Adjusted`
HM_B.ST_ad = HM_B.ST$`HM-B.ST.Adjusted`
ATCO_A.ST_ad = ATCO_A.ST$`ATCO-A.ST.Adjusted`
ERIC_B.ST_ad = ERIC_B.ST$`ERIC-B.ST.Adjusted`
ESSITY_B.ST_ad = ESSITY_B.ST$`ESSITY-B.ST.Adjusted`
SAND.ST_ad = SAND.ST$SAND.ST.Adjusted
BOL.ST_ad = BOL.ST$BOL.ST.Adjusted
GETI_B.ST_ad = GETI_B.ST$`GETI-B.ST.Adjusted`
ALFA.ST_ad = ALFA.ST$ALFA.ST.Adjusted
ATCO_B.ST_ad = ATCO_B.ST$`ATCO-B.ST.Adjusted`
VOLV_B.ST_ad = VOLV_B.ST$`VOLV-B.ST.Adjusted`
SHB_A.ST_ad = SHB_A.ST$`SHB-A.ST.Adjusted`
ELUX_B.ST_ad = ELUX_B.ST$`ELUX-B.ST.Adjusted`
SEB_A.ST_ad = SEB_A.ST$`SEB-A.ST.Adjusted`
ASSA_B.ST_ad = ASSA_B.ST$`ASSA-B.ST.Adjusted`
AZN.ST_ad = AZN.ST$AZN.ST.Adjusted
SWED_A.ST_ad = SWED_A.ST$`SWED-A.ST.Adjusted`
TELIA.ST_ad = TELIA.ST$TELIA.ST.Adjusted
TEL2_B.ST_ad = TEL2_B.ST$`TEL2-B.ST.Adjusted`
SBB_B.ST_ad = SBB_B.ST$`SBB-B.ST.Adjusted`
INVE_B.ST_ad = INVE_B.ST$`INVE-B.ST.Adjusted`
SINCH.ST_ad = SINCH.ST$SINCH.ST.Adjusted
SCA_B.ST_ad = SCA_B.ST$`SCA-B.ST.Adjusted`
HEXA_B.ST_ad = HEXA_B.ST$`HEXA-B.ST.Adjusted`

market_data=cbind(OMX_ad,ABB.ST_ad,NDA_SE.ST_ad, HM_B.ST_ad, ATCO_A.ST_ad,
                  ERIC_B.ST_ad, ESSITY_B.ST_ad, SAND.ST_ad, BOL.ST_ad,
                  GETI_B.ST_ad, ALFA.ST_ad, ATCO_B.ST_ad, VOLV_B.ST_ad,
                  SHB_A.ST_ad, ELUX_B.ST_ad, SEB_A.ST_ad, ASSA_B.ST_ad,
                  AZN.ST_ad, SWED_A.ST_ad, TELIA.ST_ad, TEL2_B.ST_ad,
                  SBB_B.ST_ad, INVE_B.ST_ad, SINCH.ST_ad, SCA_B.ST_ad,
                  HEXA_B.ST_ad)

k=nrow(market_data)
returns=as.data.frame(log(as.matrix(market_data[2:k,]))-log(
```

```
  as.matrix(market_data[1:(k-1),])))
names(returns)=c("rOMX","rABB","rNDA.SE.ST", "HM_B.ST", "ATCO_A.ST",
                 "ERIC_B.ST", "ESSITY_B.ST", "SAND.ST", "BOL.ST",
                 "GETI_B.ST", "ALFA.ST", "ATCO_B.ST", "VOLV_B.ST",
                 "SHB_A.ST", "ELUX_B.ST", "SEB_A.ST", "ASSA_B.ST",
                 "AZN.ST", "SWED_A.ST", "TELIA.ST", "TEL2_B.ST",
                 "SBB_B.ST", "INVE_B.ST", "SINCH.ST", "SCA_B.ST",
                 "HEXA_B.ST")

save(returns, file = "returns.Rda")
load("returns.Rda")
returns <- na.omit(returns)
model <- lm(data = returns, rOMX ~ .)
summary(model)
forward_model <- step(model, direction = "forward",
                      scope = formula(~ .))
summary(forward_model)
backward_model <- step(model, direction = "backward")
summary(backward_model)
x_var <- as.matrix(returns[,2:26])
y_var <- as.matrix(returns[,1])
ridge_fit <- glmnet(x_var, y_var, alpha = 0, lambda.min.ratio = 1e-6)
# printing the best lambda for ridge
ridge_best_lambda <- ridge_cv$lambda.min
print(ridge_best_lambda)
plot(ridge_fit, xvar = "lambda")
abline(v = log(ridge_best_lambda), lty = "dashed", col = "red")
best_ridge_fit <- glmnet(x_var, y_var, alpha = 0, lambda = ridge_best_lambda)
best_ridge_fit$beta
# fitting the model
lasso_fit <- glmnet(x_var, y_var, alpha = 1, lambda.min.ratio = 1e-6)

# cross-validation to get best lambda
lasso_cv <- cv.glmnet(x_var, y_var, alpha = 1)

# printing the best lambda for lasso
lasso_best_lambda <- lasso_cv$lambda.min
print(lasso_best_lambda)
plot(lasso_fit, xvar = "lambda")
abline(v = log(lasso_best_lambda), lty = "dashed", col = "red")
best_lasso_fit <- glmnet(x_var, y_var, alpha = 1, lambda = lasso_best_lambda)
best_lasso_fit$beta
# converting capital market index variable to binary categorical
returns_cat <- returns %>%
  mutate(rOMX = ifelse(rOMX > 0, 1, 0))

# length of data in order make split
n_data <- nrow(returns_cat)

# creating integer representing roughly 80 % split of data.
training_length <- round(n_data * 0.8)

# splitting the data into training and test
```

```r
train_data <- returns_cat[1:training_length, ]
test_data <- returns_cat[(training_length+1):n_data, ]
# selecting the 12 stocks which coeff estimates were the highest in the lin mod
subset_stocks <- model$coefficients %>%
  data.frame() %>%
  set_colnames("coeff") %>%
  arrange(desc(coeff)) %>%
  t()

subset_stocks <- colnames(subset_stocks)[1:12]
# the dimensions of the resulting data sets
dim(train_data)
dim(test_data)
# all stocks model
logmodel_full <- glm(rOMX ~ ., family = binomial, data = train_data)

# stock subset model
logmodel_subset <- glm(rOMX ~ ., family = binomial,
                    data = train_data[c("rOMX", subset_stocks)])
summary(logmodel_full)
summary(logmodel_subset)
# predictions of the full model
predictions_full <- predict(logmodel_full,
                        newdata = select(test_data, !rOMX),
                        type = "response")
pred_labels_full <- ifelse(predictions_full > 0.5, 1, 0)

# predictions of the subset model
predictions_subset <- predict(logmodel_subset,
                        newdata = test_data[subset_stocks],
                        type = "response")
pred_labels_subset <- ifelse(predictions_subset > 0.5, 1, 0)
n_test_data <- nrow(test_data)
true_labels <- test_data["rOMX"]

full_model_misclassifications <- abs(true_labels - pred_labels_full) %>% sum()
sub_model_misclassifications <- abs(true_labels - pred_labels_subset) %>% sum()

full_model_accuracy <- full_model_misclassifications / n_test_data
sub_model_accuracy <- sub_model_misclassifications / n_test_data

full_model_accuracy
sub_model_accuracy
```

```
## R version 4.4.0 (2024-04-24 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
```

```
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: Europe/Stockholm
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] magrittr_2.0.3  glmnet_4.1-8    Matrix_1.7-0    corrplot_0.95
##  [5] quantmod_0.4.26 TTR_0.24.4      xts_0.14.1      zoo_1.8-12
##  [9] lubridate_1.9.3 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
## [13] purrr_1.0.2     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1
## [17] ggplot2_3.5.1   tidyverse_2.0.0 knitr_1.46
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    shape_1.4.6.1     stringi_1.8.3
##  [5] lattice_0.22-6    hms_1.1.3         digest_0.6.35     evaluate_0.23
##  [9] grid_4.4.0        timechange_0.3.0  iterators_1.0.14  fastmap_1.1.1
## [13] jsonlite_1.8.8    foreach_1.5.2     survival_3.5-8    fansi_1.0.6
## [17] scales_1.3.0      codetools_0.2-20  cli_3.6.2         rlang_1.1.3
## [21] splines_4.4.0     munsell_0.5.1     withr_3.0.0       yaml_2.3.8
## [25] tools_4.4.0       tzdb_0.4.0        colorspace_2.1-0  curl_5.2.1
## [29] vctrs_0.6.5       R6_2.5.1          lifecycle_1.0.4   pkgconfig_2.0.3
## [33] pillar_1.9.0      gtable_0.3.5      Rcpp_1.0.13       glue_1.7.0
## [37] highr_0.10        xfun_0.43         tidyselect_1.2.1  rstudioapi_0.16.0
## [41] htmltools_0.5.8.1 rmarkdown_2.26    compiler_4.4.0
```