Equity Return Predictability

Risk Management. Homework 2

Roman Kypybida

1. Linear predictive model

Model of the form: $R_{t+1} = a + \beta X_{t+1} + \varepsilon_{t+1}$, where $R_{t+1} = \ln\left(Index_{t+1} - \frac{D12_{t+1}}{12}\right) - \ln\left(Index_{t+1}\right)$.

I computed the linear regression using lm() R function, which regresses the data using the OLS approach. I evaluated the model on the training set and obtained the following results:

```
Call:
lm(formula = r ~ log(E12, base = 10) + log(D12, base = 10) +
    DE12 + svar + bm + ntis + tbl + ltr + TSPREAD + DSPREAD +
    DRSPREAD + infl, data = data[-(length(data):length(data)),
    ])

Residuals:
     Min       1Q   Median       3Q      Max
-1.37757 -0.05396  0.00856  0.06458  0.20967

Coefficients: (1 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.98601    0.03703  53.639  < 2e-16 ***
log(E12, base = 10)  0.74862    0.07171  10.439  < 2e-16 ***
log(D12, base = 10)  0.59455    0.08103   7.338 5.26e-13 ***
DE12                 0.13441    0.02940   4.572 5.57e-06 ***
svar                -1.80795    0.91655  -1.973   0.0489 *
bm                  -0.84227    0.02648 -31.807  < 2e-16 ***
ntis                 1.50555    0.23773   6.333 3.97e-10 ***
tbl                 -0.15469    0.31429  -0.492   0.6227
ltr                  0.18200    0.39002   0.467   0.6409
TSPREAD                   NA         NA      NA       NA
DSPREAD             -7.75568    1.39515  -5.559 3.68e-08 ***
DRSPREAD             0.27159    0.37863   0.717   0.4734
infl                 1.71171    1.12429   1.522   0.1283
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09624 on 815 degrees of freedom
Multiple R-squared:  0.9854,    Adjusted R-squared:  0.9852
F-statistic:  5010 on 11 and 815 DF,  p-value: < 2.2e-16
```

1.1 The summary of the OLS regression.

The regression has fitted the data very well with R squared coefficient equal to 0.9834. The standard error is 0.09624.

According to the summary, the Earnings predictor yields plus approximately ~0.74 percent to the stock return. Second highest coefficient is the coefficient of the Inflation variable, but with t value less than 2, meaning the variable is not that significant. The E12 has huge statistical significance, highest among all. The Dividends increase the return on average on 0.59 points and have third highest statistical significance among the predictors with positive effect. The dividends-earnings ratio increase the returns on 0.13 percent and is third by statistical importance.

Stock variance and Book-To-Market predictors have negative effect and decrease the returns by almost 2 percent and ~0.84 correspondingly. b/m is very statistically important. On the other hand the Stock Variance is not of big statistical significance.

Net Issues increase the returns by ~1.5 points and is second most important of the positive predictors. Treasury Bills and Long-Term Yield decrease the returns by ~0.15 and increase by ~0.18 correspondingly, both have low significance.

Among the calculated spreads, the Corporate Bonds spread significantly decrease the returns by ~7.76 percent and have the second largest significance among the predictors with negative impact. The Default return spread has low impact and significance, but positive sign and increases the return approximately by 0.27 on average.

The program has failed to compute the coefficient data for the Term spread.

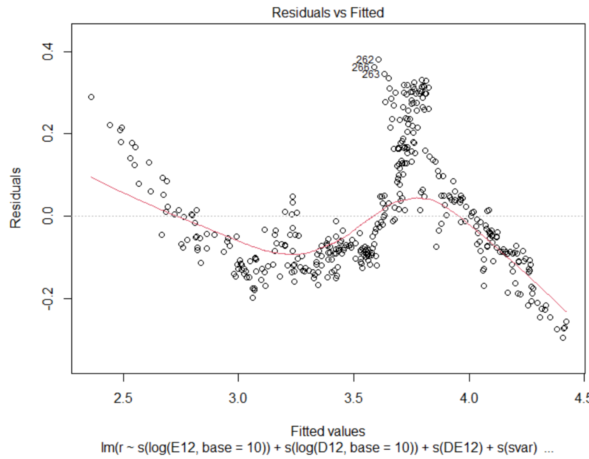
2. The OLS regression

Model of the form: $R_{t+1} = a + \beta f(X_{t+1}) + \varepsilon_{t+1}$, where $R_{t+1} = \ln\left(Index_{t+1} - \frac{D12_{t+1}}{12}\right) - \ln\left(Index_{t+1}\right)$.

The model is used in all following examples. The data is all rows of the dataset dated after 1980.

```
Call:
lm(formula = r ~ s(log(E12, base = 10)) + s(log(D12, base = 10)) +
    s(DE12) + s(svar) + s(bm) + s(ntis) + s(tbl) + s(ltr) + s(TSPREAD) +
    s(DSPREAD) + s(DRSPREAD) + s(infl), data = dtrain[-(length(dtrain):length(dtrain)),
    ])

Residuals:
     Min       1Q   Median       3Q      Max
-0.29620 -0.10305 -0.04871  0.08485  0.38167

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            2.989e+00  3.013e-02  99.190  < 2e-16 ***
s(log(E12, base = 10))  5.658e-02  1.048e-02   5.400 1.17e-07 ***
s(log(D12, base = 10))  3.113e-01  2.850e-02  10.920  < 2e-16 ***
s(DE12)                2.155e-03  2.083e-03   1.034  0.30161
s(svar)               -4.556e+02  3.804e+02  -1.198  0.23173
s(bm)                 -3.131e-01  5.354e-02  -5.847 1.07e-08 ***
s(ntis)                8.919e+02  2.884e+02   3.093  0.00213 **
s(tbl)                -7.538e+01  2.792e+01  -2.700  0.00725 **
s(ltr)                -2.717e+01  3.002e+01  -0.905  0.36601
s(TSPREAD)            -6.849e+01  2.617e+01  -2.617  0.00922 **
s(DSPREAD)             5.016e+03  3.117e+03   1.609  0.10840
s(DRSPREAD)           -6.099e+01  1.888e+01  -3.230  0.00134 **
s(infl)               -3.838e+04  2.241e+04  -1.712  0.08764 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1501 on 384 degrees of freedom
Multiple R-squared:  0.9015,    Adjusted R-squared:  0.8984
F-statistic: 292.8 on 12 and 384 DF,  p-value: < 2.2e-16
```

2.1.1 The regression summary. The R squared equals 0.9015, the standard error is 0.1501. Good result for a regression.

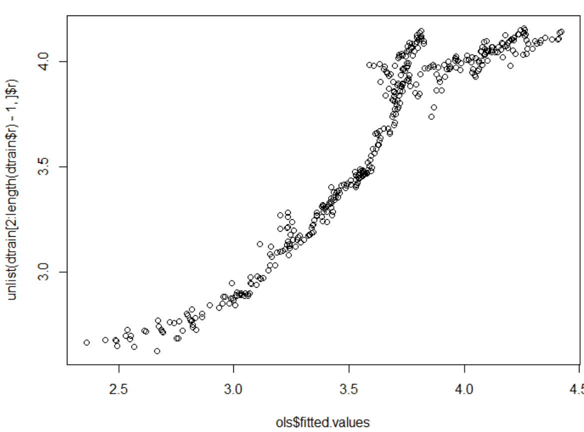


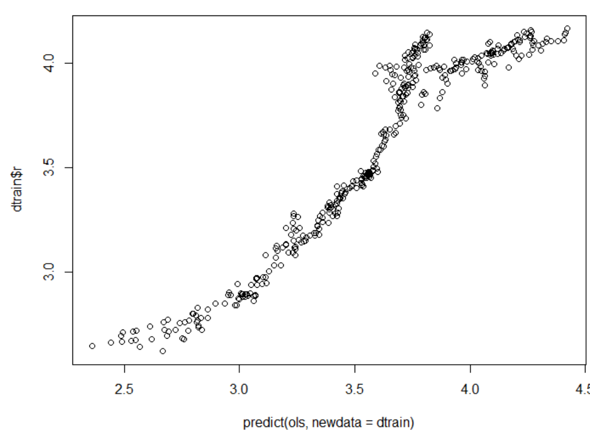
2.1.2 A graph of residual values in relation to fitted values. We can see that the values are worse fitted in the middle of the regression and the regression has better results till the end of the set.

The RMSE and R squared of the model for predictions:

$$R2 = 0.8780935;$$
$$RMSE \ = 0.1646251;$$

The model has good explanatory value and quite small error.





2.1.3 A graph of the training data and values fitted by the regression.

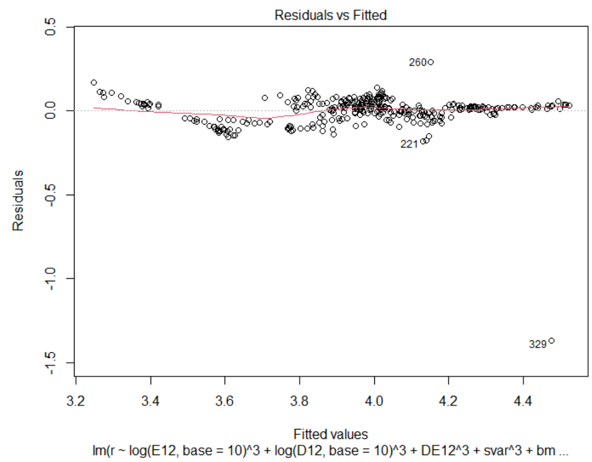
2.1.4 A graph of the training data and values predicted by the regression for the training set.

The graphs show that the regressions results are approaching the linear, although are a bit scattered in the beginning and the end of the set.

Next, I performed the following procedure:

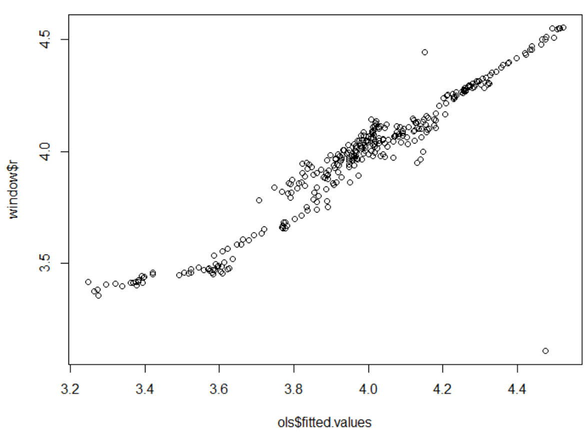
1) I predicted a value on a test set using a model, trained on a window over the dataset;
2) Updated the corresponding entry in the dataset with the returned value;
3) Retrained the model on the window shifted right;
4) Repeated the process till the end of the test set;

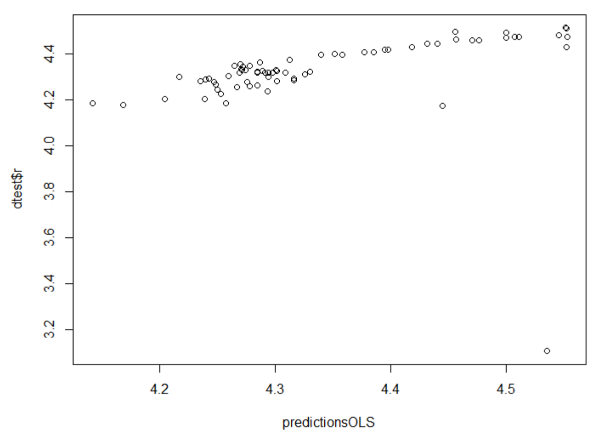
```
Call:
lm(formula = r ~ log(E12, base = 10)^3 + log(D12, base = 10)^3 +
    DE12^3 + svar^3 + bm^3 + ntis^3 + tbl^3 + ltr^3 + TSPREAD^3 +
    DSPREAD^3 + DRSPREAD^3 + infl^3, data = window)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36857 -0.03183  0.01509  0.04126  0.29240

Coefficients: (1 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.52000    0.21236  11.867  < 2e-16 ***
log(E12, base = 10)  0.66424    0.09359   7.098 8.36e-12 ***
log(D12, base = 10)  0.53805    0.11309   4.758 2.98e-06 ***
DE12                 0.09087    0.03575   2.542  0.01150 *
svar                -1.82483    1.68951  -1.080  0.28092
bm                  -1.53554    0.09085 -16.902  < 2e-16 ***
ntis                 1.50949    0.54424   2.774  0.00587 **
tbl                 -1.10551    0.62092  -1.780  0.07596 .
ltr                 -1.62265    1.40995  -1.151  0.25066
TSPREAD                   NA         NA      NA       NA
DSPREAD            -10.56221    3.20672  -3.294  0.00110 **
DRSPREAD            -1.59226    1.39521  -1.141  0.25463
infl                 4.10881    1.89349   2.170  0.03075 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1005 on 317 degrees of freedom
Multiple R-squared:  0.8805,    Adjusted R-squared:  0.8764
F-statistic: 212.4 on 11 and 317 DF,  p-value: < 2.2e-16
```



2.1.5 The results of the model regressed on the last window of data in the sliding window procedure.

2.1.6 The residuals of the aforementioned model versus the values fitted by it.





2.1.7 Values of the last window versus the values fitted by the model regressed on the last window of data in the sliding window procedure.

2.1.8 Values of the test set versus the prediction of the regression.

The average metrics in the sliding window procedure:

$$R2 = 0.9643921;$$
$$RMSE = 0.1790797;$$

The metrics of the model in prediction:

$$R2 = -0.09295666;$$
$$RMSE = 0.06793074;$$

The average metrics show that model fitted the data well and commits a small error. From the latter metrics we observe negative R2, but smaller RMSE error. It could indicate overfitting.

2.2 Lasso regression

I trained the Lasso regression using "caret" R package with an alpha parameter equal to 1 and tune hyperparameter λ. I used resampling with sliding_window() function for tuning (same procedure I applied to Ridge and Elastic-Net regressions) and the tuned $\lambda = 0.1$.

The average RMSE over the samples for trained model is ~0.46.

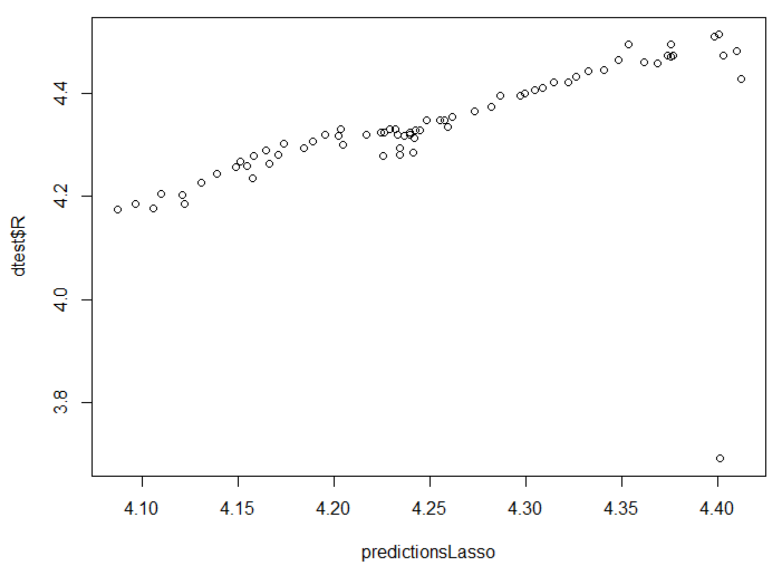
The metrics for prediction using the trained model are the following:

RMSE = 0.1429008;

R2 = 0.9226518;

The model shows normal results on the training set with a small error rate.

Next I compute the sliding window procedure with the following results:



2.2.1 A graph of testing data in relation to predictions on the test data. The model's results are linear although are positioned lower than the values in the test set.

The average metrics in the sliding window procedure:

$R2 = 0.894606$;
$RMSE = 0.1359905$;

The metrics of the model in prediction:

$R2 = - 0.2079244$;
$RMSE = 0.1290986$;

The model has normal results in forecasting on the window used for training, but much poorer performance in prediction of real data. The negative sign of R2 indicates the overfitting. On the other hand, the model has small errors.

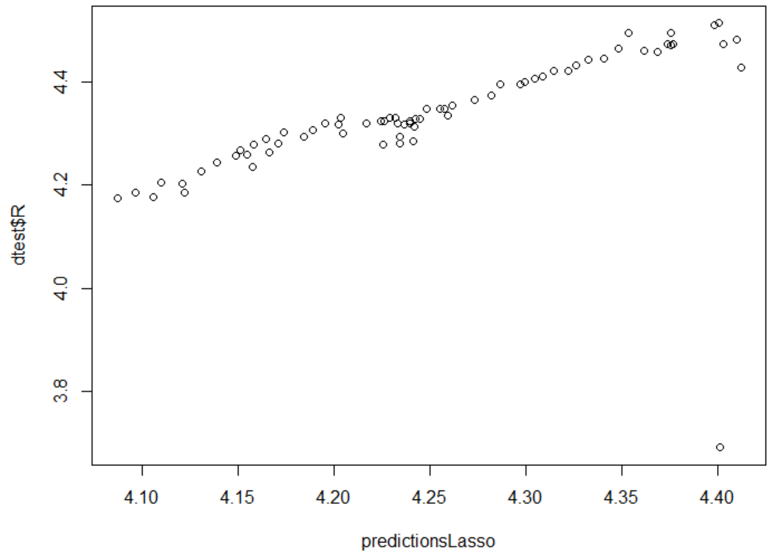
2.3 Ridge regression

Here I used the same "caret" package, but for Ridge regression I used alpha equal to 0 and tuned λ with the same resampling procedure as the Lasso regression. The tuned λ equals to 0.1.

The average R2 of the model over for the training data is 0.9306287, which indicates good fit for the training set. The average RMSE metric is 0.162074 – model has normally small error.

The metrics for predicting on the training set are similar:

$R2 = 0.9308128$;
$RMSE = 0.1239588$;



2.3.1 A relation of the predictions of the model and test data, the predictions were performed on.

The average metrics in the sliding window procedure:

$R2 = 0.955255;$
$RMSE \ = 0.09010101;$

The metrics of the model in prediction:

$R2 = 0.2466609;$
$RMSE \ = 0.1019523;$

The model has great performance on the training windows, but very poor on the test data – model explains only 24% and has 10 times greater error on the test data than on the training set.

2.4 Elastic-Net regression

Here, I trained the Elastic-Net using "caret" package and varying the λ and α parameters. The tuned parameters are λ = 0.02 and α = 1.

The model has a relatively high RMSE of ~0.36.

The metrics of the model in prediction on the training set:

$R2 = 0.8616516;$
$RMSE \ = 0.1752877;$

The average metrics in the sliding window procedure:

$R2 = 0.9586489;$

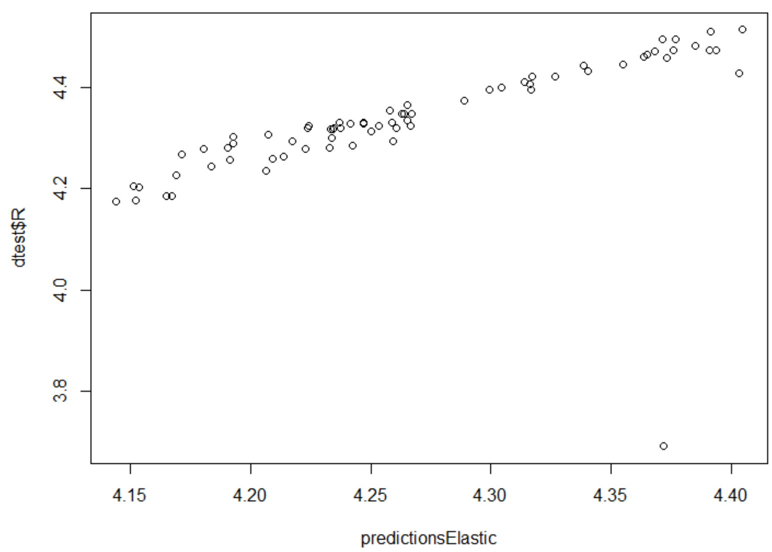
$RMSE\ = 0.007357927;$

The metrics of the model in prediction:

$R2 = 0.03134184;$

$RMSE\ = 0.1156078;$

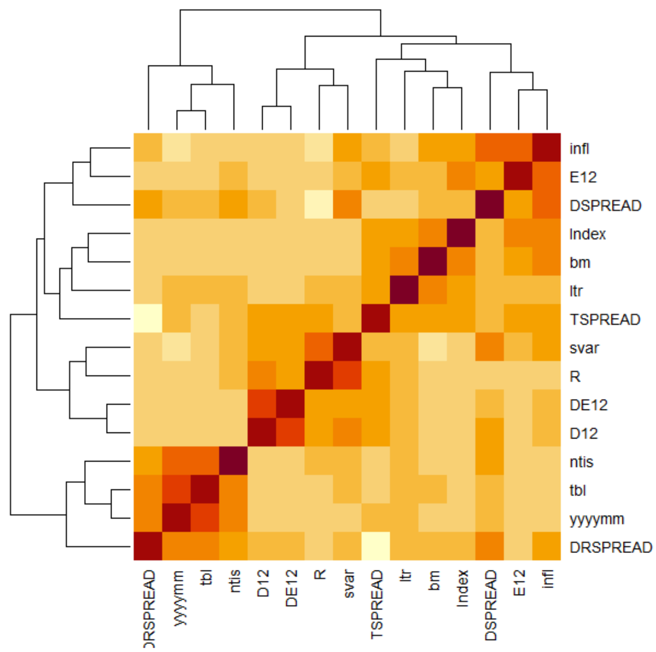
The average metrics indicate great performance over the training process, but the metrics calculated as test data in the relation to the predictions shows very bad explanatory performance of the model. On the other hand model does not have very large error, although the error is a 100 times higher than on the training set.



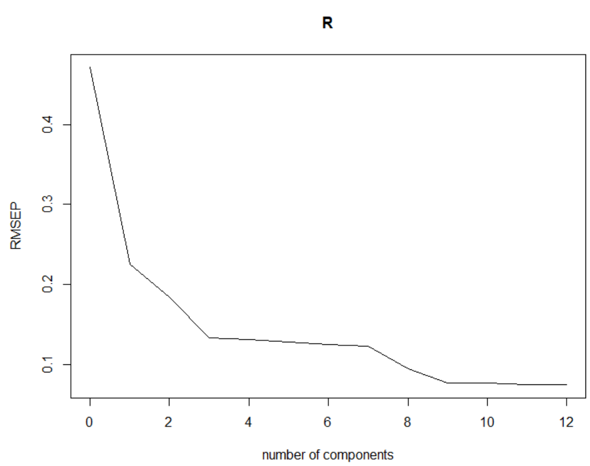
2.4.1 Predictions to test data relation. The model is scattered over a linearity in the top part of the graph. The model approaches linear results, but in fact is quite poor in performance.

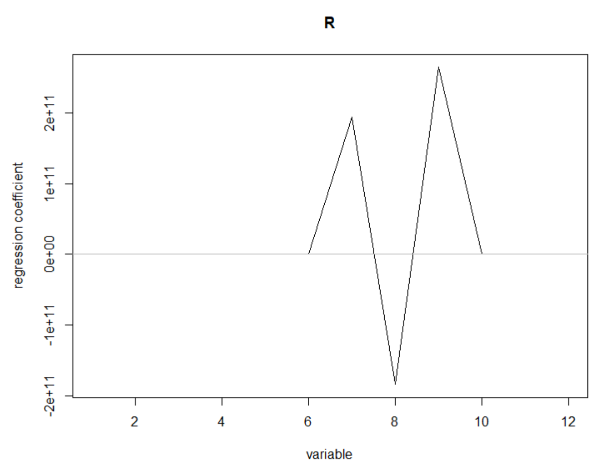
3. PCA

I computed the Principal Component Analysis and regression based on PCA using R packages “FactoMineR”, “factoextra” and “pls”.



3.1 A heatmap of the correlation matrix of columns the training data.

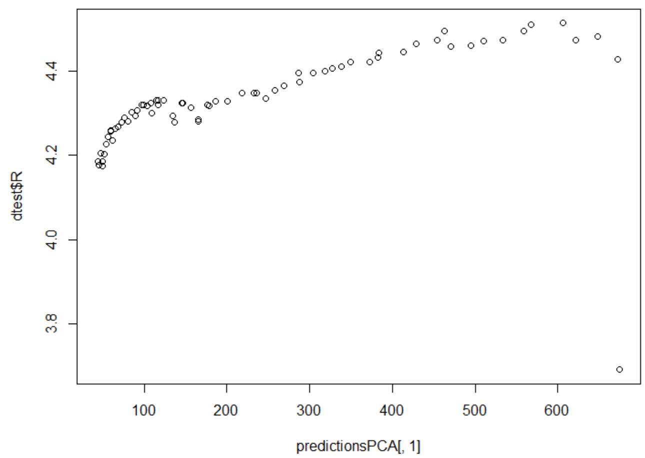




3.2 A validation plot of the PCA regression. The RMSEP error falls as the number of components grows.

3.3 A coefficient plot of the PCA regression. We can see that the highest regression coefficients is on the variable 10 and 8. The lowest is on the variable 8.

Regression with 3 components, results:



3.4 Predictions versus the test data. The data is far from predicting the data very accurately.

The average metrics in the sliding window procedure:

$RMSE$ = 943.1337;

$R2$ = -101.61;

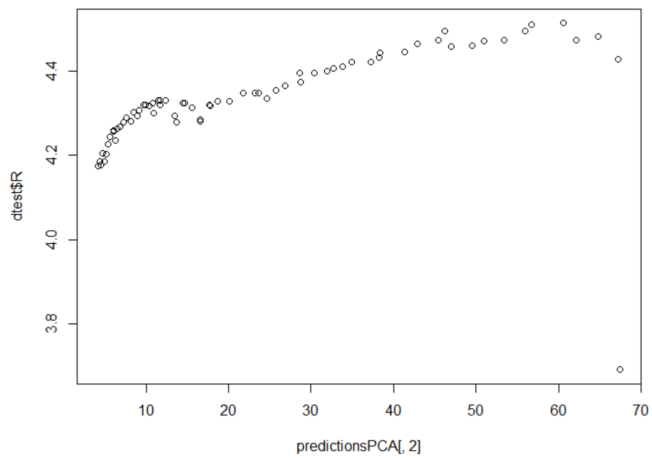
The metrics of the model in prediction:

$R2$ = -6557426.8009;

$RMSE$ = 300.7936;

The model overfits the data. The metrics are uninterpretable. The graph shows us that the model becomes better till the end of the set, but is still weak in estimating the values.

Regression with 5 components, results:



3.5 Predictions versus the test data.

The average metrics in the sliding window procedure:

$R2 =$ -206.2911;
$RMSE \ =$ 919.1553;

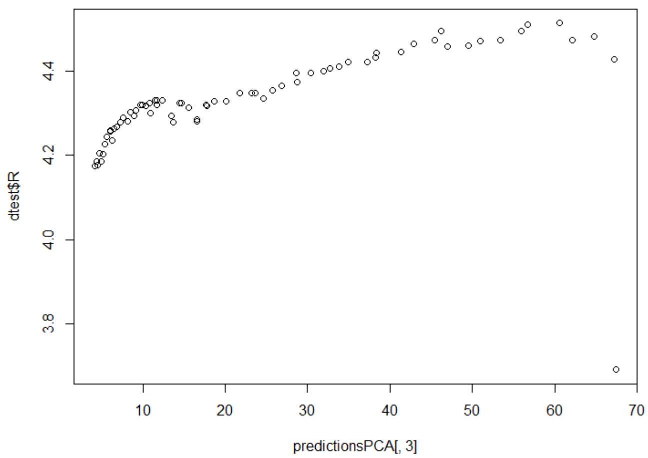
The metrics of the model in prediction:

$R2 =$ -53113.78886;

$RMSE \ =$ 27.07133;

The R2 have negative sign, but lower modulo value than in the case of 3 components. The results on the graph are a bit shifted to the right in relation to the aforementioned case. The model is still poor and metrics are hard to interpret.

Regression with 10 components, results:



3.6 Predictions versus the test data.

The average metrics in the sliding window procedure:

$R2$ = -427.8245;

$RMSE$ = 862.6522;

The metrics of the model in prediction:

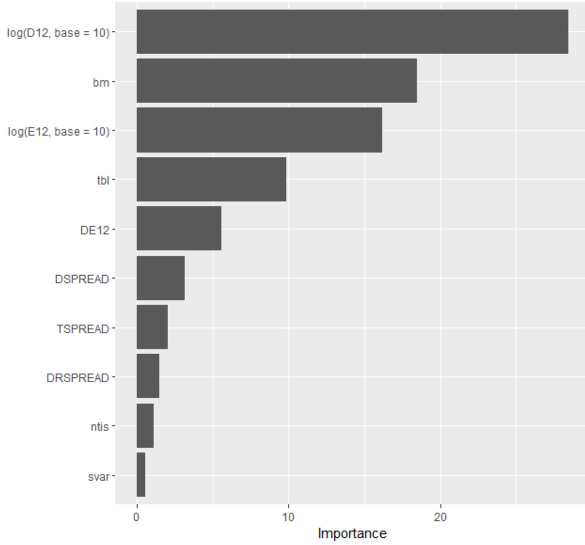
$R2$ = -53113.78886;

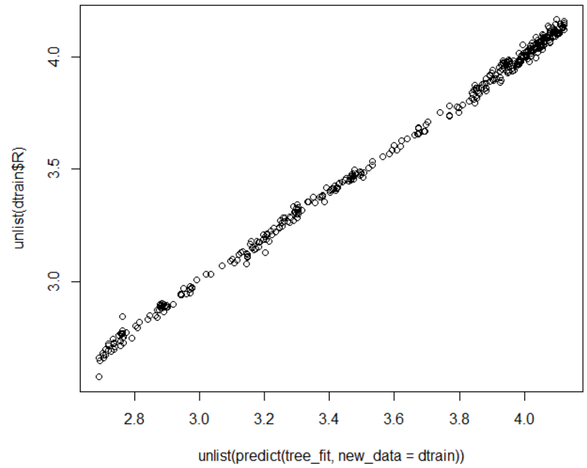
$RMSE$ = 27.07133;

The model has same results in prediction as the previous one. The model is bad in performance and has terrible overfit.

4. Random forests

I trained random forest using a R engine “ranger” with number of nodes to sample on each split (hyperparameter “mtry”) $mtry = \sqrt{n}$, where n is the number of predictors and hyperparameters “trees” (number of trees) and “min_n” (minimum number of nodes) with tuned values. The tuning process was performed with resamples formed using a function sliding_window(), which forms the resamples with a sliding window over the given data. A random forest trained on the training data has yielded the following importance graph:





4.1.Importance of the predictors, random forest trained on the training sample.

4.2. The relation of the training data to results of the model.

From the graph we observe that the most important predictors are logs of Earnings (E12) and Dividends (D12). The least important are Stock Variance (svar) and Net Issues (ntis).

The model used the following hyperparameters: $mtry = 3, number\ of\ trees = 722, target\ node\ size = 16$.

The MSE and R squared of the model:

OOB prediction error (MSE): 0.001189692;

R squared (OOB): 0.9946492;

Manually computed metrics of the model:

$R2$ = 0.9979273;

$RMSE$ = 0.0004585288;

The average RMSE and R squared over the windows:

$R2$ = 0.997299;

$RMSE$ = 0.000433801;

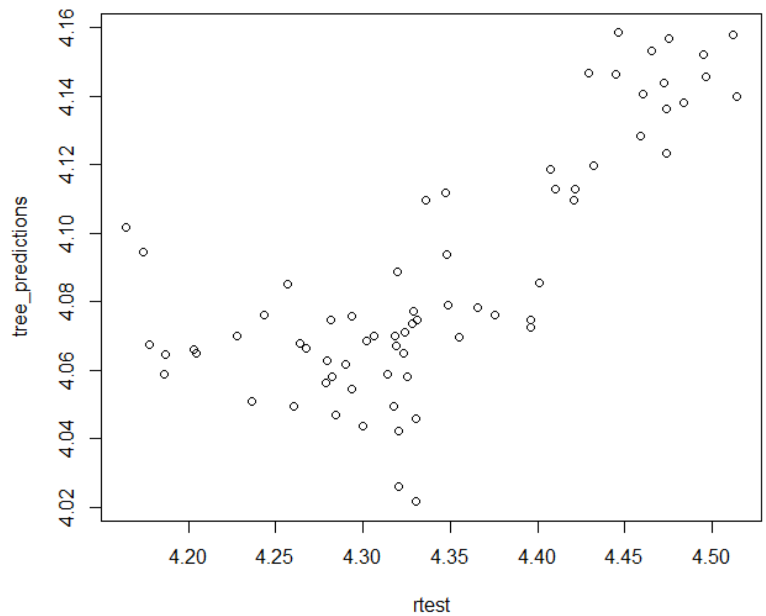
The model shows very good explanation for the sliding window tests and has little error.

The RMSE and R squared of the model for predictions:

$R2$ = -7.435457;

$RMSE$ = 0.2635666;

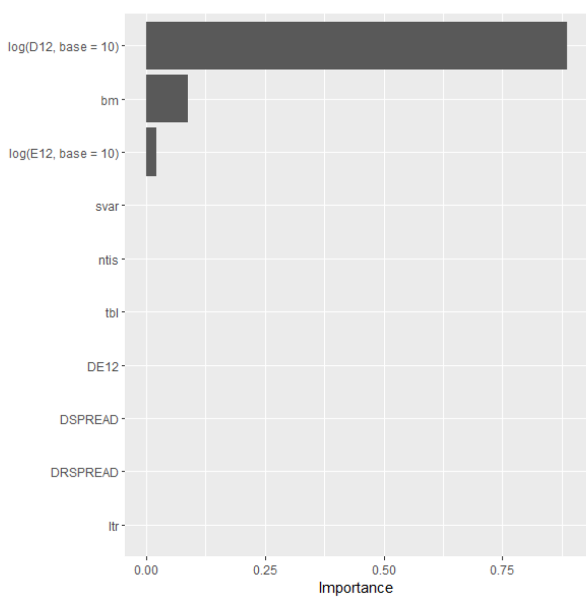
The former metrics indicate that the model is strongly overfitted and due to that has substantially larger error in predicting new values rather than fitting the data.

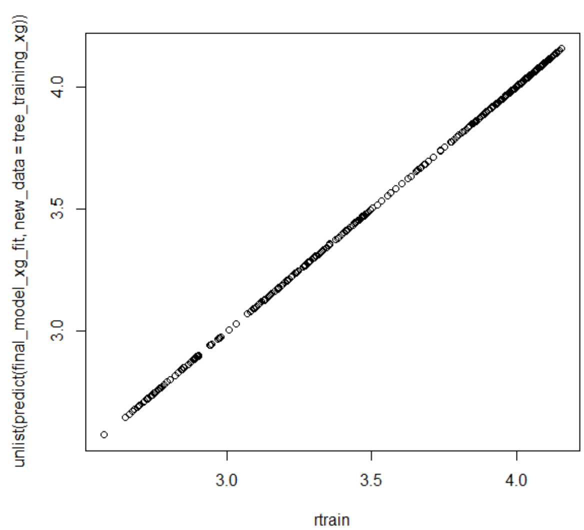


4.3 The relation of testing data to the predicted values. The values are much more scattered and indicate the bad performance of the model.

5. Boosted trees

I trained the boosted tree of engine “xgboost” and trees number equals to 500 with learning rate, tree depth and sample size set for tuning. I used the same sliding window procedure as in the previous models.





5.1 The importance graph for the boosted tree. The Dividends and Earnings are among the most important again. Although the most important variable D12 is far more important than the next two predictors. The E12 has little importance, although stays among top 3 most important variables. The second place takes the Book-To-Market ratio.

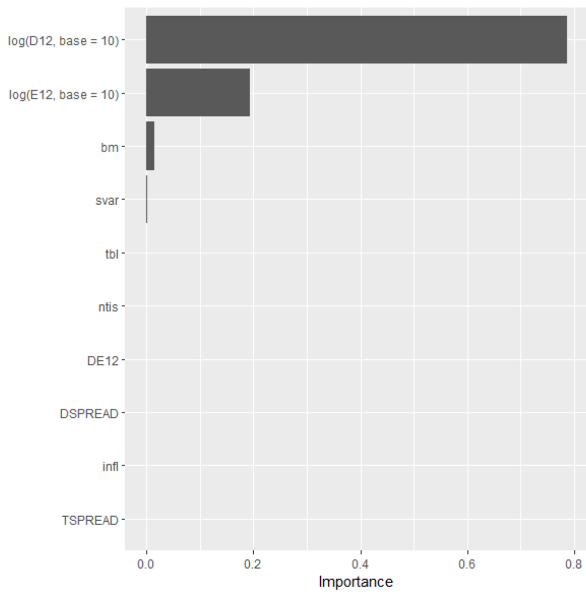
5.2 The second graph is relation of the predictions and the training data. The is totally linear and indicates strong fit and overfit of the data.

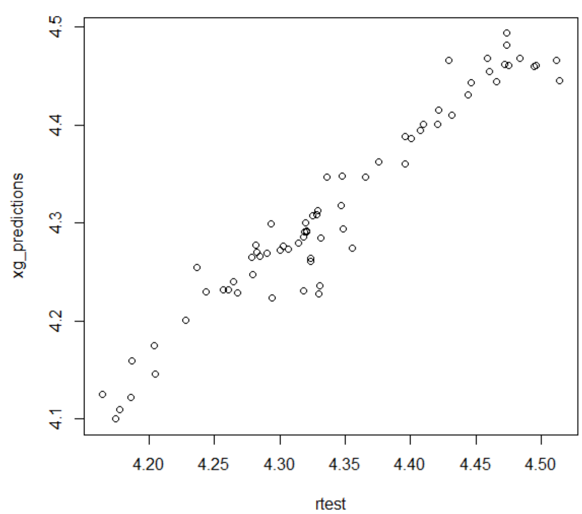
The mean RMSE over the training process (over the sliding windows): $RMSE = 0.01873257$. A small error, good approximate results.

Manually computed metrics of the model:

$$R2 = 0.9999985;$$
$$RMSE \; = 3.242749e\text{-}07;$$

The R2 confirms the former results: the model is very well fit and overfits the data.





5.3 The importance graph for the last fit model (the model with the highest number of predictions used for training).

The virtually the same results as in the training, but the Earnings predictor is interchanged with bm and has grown more importance than in the aforementioned analysis. Also the Stock variance has some importance in this case.

5.4 The graph of relation between the testing data and predictions of the model.

The average RMSE and R squared over the windows:

$$R2 = 0.9999983;$$
$$RMSE \; = 3.074149e\text{-}07;$$

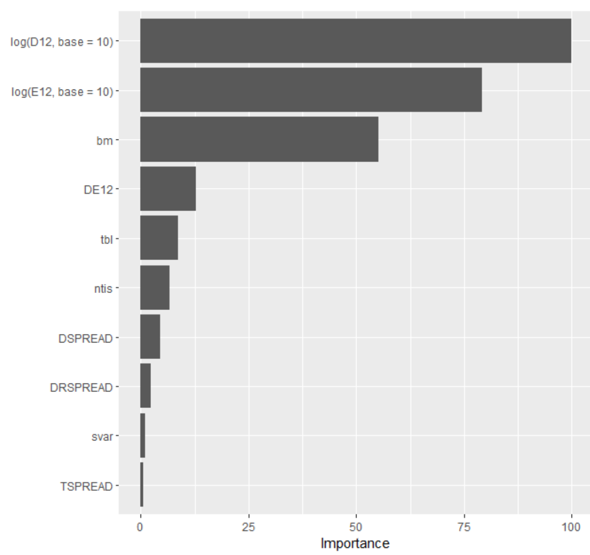
The data has very good, almost perfect, in fact, fit of the model with a very small error, close to overfitting.
The RMSE and R squared of the model for predictions:

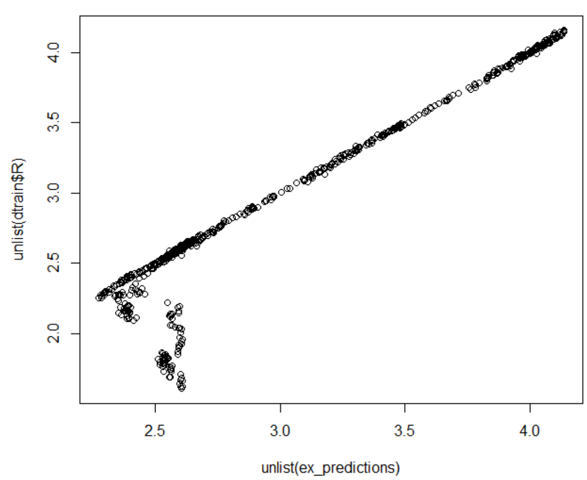
$$R2 = 0.819159;$$
$$RMSE \ = 0.03859086;$$

The model performs worse in the realm of predictions, R2 falls almost by more almost 20 points and error grows sub stantially. Overall good performance and fit for the data.

6. Extremely randomized trees

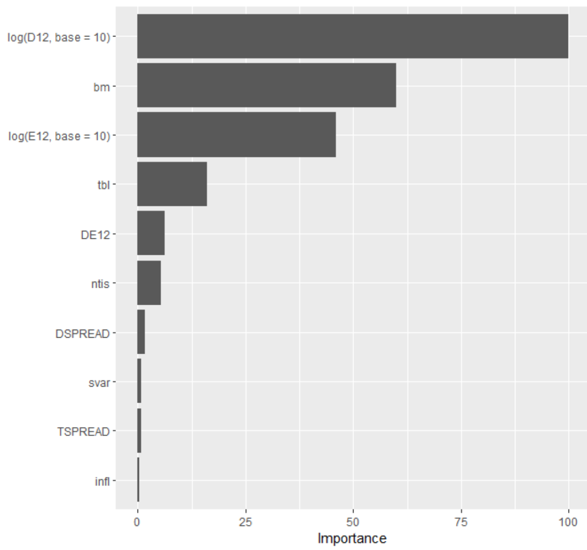
The extremely randomized trees split the data randomly instead of choosing the most optimal split and constructs the trees over the entire dataset instead of using the samples. Here I used R package “caret” and splitrule set to “extratrees” to evaluate the models. I have not used any resampling for the training on the training data, but applied “sliding window” procedure for the prediction process.

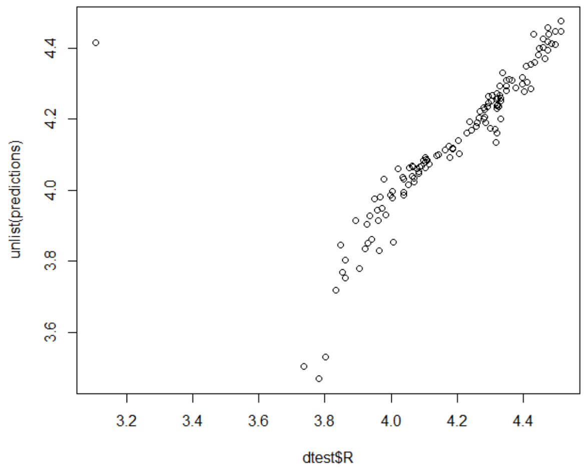




6.1 The importance graph of the predictors. The most important as in the previous models are D12 and E12.
The third place is still taken by the Book-To-Market ratio. The Svar variable still has some importance, but remains among the least important predictors.
6.2 The relation of the predictions and training data. The model has problems in evaluating the first part of the training set, but has linear close to perfect performance till the end of the set.

Mtry hyperparameter value of the model is 4. The next are results of the prediction procedure.





6.3 The importance graph for the last model of the sliding window process over the predictions. The bm is on the second place now and the Svar predictor has more importance than before. The inflation predictor has gained some importance now.

6.4 The predictions to test data relation. The model approximates the results and approaches the linearity and the test ing data to the end of the process, but is substantially scattered.

The average RMSE and R squared over the windows:

$$R2 = 0.9996017;$$
$$RMSE \ = 0.01354936;$$

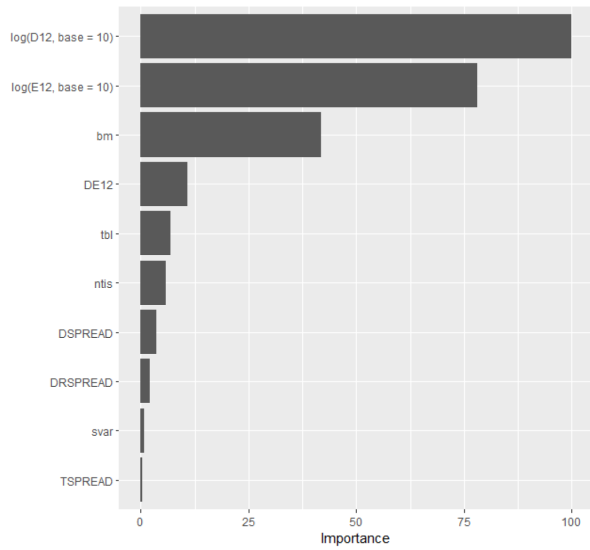
Similar results to the boosted trees and random forest, but these models have higher RMSE and a bit lower R2. Less overfitting, but lower regression accuracy performance.

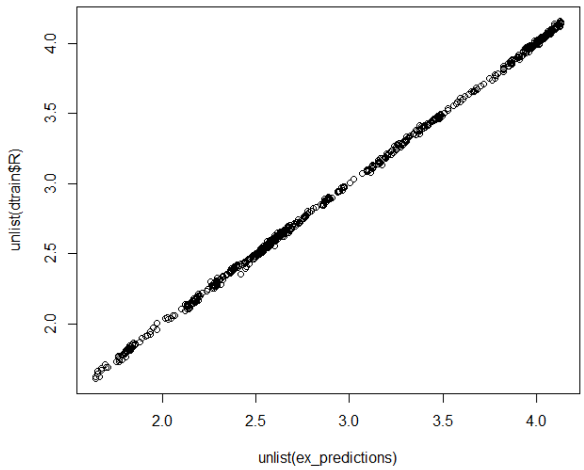
The RMSE and R squared of the model for predictions:

$$R2 = 0.5737774;$$
$$RMSE \ = 0.1418618;$$

The model has relatively low R2 and is poor in explaining the data. It could be to no use of resampling using a sliding window, which could yield the time effect. The predictions RMSE is even higher than average over the sliding window.

I repeated the same steps, but using the resamples with sliding_window() function.





6.5 The importance graph has similar structure as in the previous case. The DE12 predictor is now the fourth and tbl is below it. The Svar has more importance than in the model without the resampling.
6.6 Now the relation of real data and the models returns is almost linear, although is not completely perfect. There is no down tail in the beginning as in the case with no resampling.

The average RMSE and R squared over the windows:

$$R2 = 0.9996015;$$
$$RMSE = 0.01355961;$$

The model almost the same R2 and RMSE.
The RMSE and R squared of the model for predictions:

$$R2 = 0.5679007;$$
$$RMSE = 0.1428364;$$

The R2 falls one point lower and RMSE grows one point higher. Although the model has better performance on the t raining set it is clearly overfitted and has bad explanatory performance on the real data.

7. Shallow neural network

I trained the neural network with 8 nodes in the hidden layer, using a “neuralnet” package. I used no resampling and tuned one hyperparameter - learning rate.

The results of the tuning process: $learning\ rate = 0.4$.

Maximum RMSE on the tuning grid = 0.03270865.

Mean RMSE on the tuning grid = 0.02928022.

Minimum RMSE on the tuning grid = 0.02554181.

Maximum R2 on the tuning grid = 0.9970724.

Mean R2 on the tuning grid = 0.9961303.

Minimum R2 on the tuning grid = 0.995199.

The results show very good explanatory performance of the network and small errors.

Than I retrained the model with the tuned value of the learning rate:

R2 = 0.9953048;

RMSE = 0.03234642;

The average RMSE and R squared over the windows:

R2 = 0.99437;

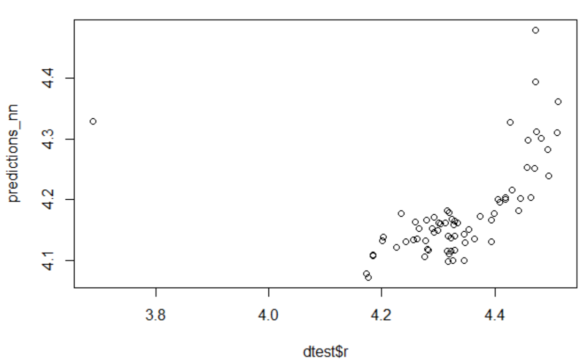
RMSE = 0.03056259;

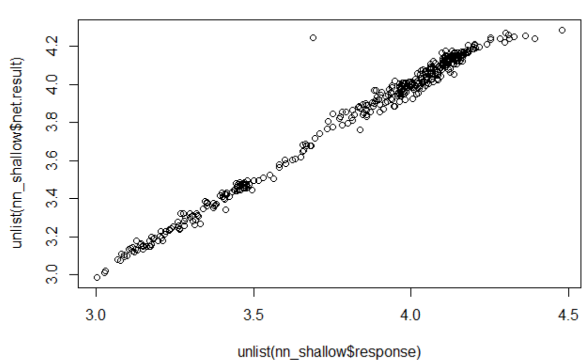
The RMSE and R squared of the model for predictions:

$R2$ = -1.641417;

$RMSE$ = 0.190998;

The model has great explanatory performance in the training process and on the sliding windows. But the results on the test set indicate the overfit, with errors 10 times larger than in the training process.





7.1 Predictions versus the test data.

7.2 The values put in the neural network versus the response of the neural network. The network shows great results and approximates linear performance.

8. Deep neural network

I trained the neural network with two layers – 16 nodes and 8 nodes in the hidden layers, using the same "neuralnet" package. As in the previous case I used no resampling and tuned only the learning rate.

The results of the tuning process: $learning\ rate = 0.05$.

Maximum RMSE on the tuning grid = 0.02902179.
Mean RMSE on the tuning grid = 0.02592802.
Minimum RMSE on the tuning grid = 0.02361659.

Maximum R2 on the tuning grid = 0.9974971.
Mean R2 on the tuning grid = 0.99696.
Minimum R2 on the tuning grid = 0.9962204.

The results show very good explanatory performance of the network and small errors.

Than I retrained the model with the tuned value of the learning rate:

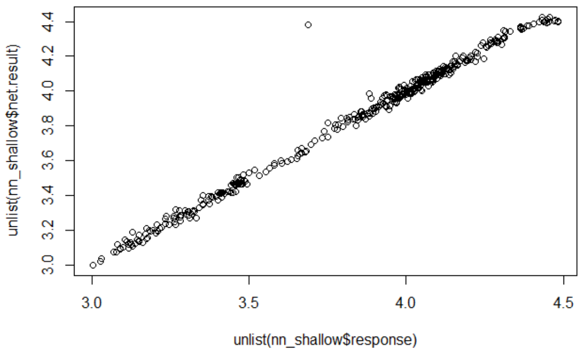
R2 = 0.9971206;
RMSE = 0.02249042;

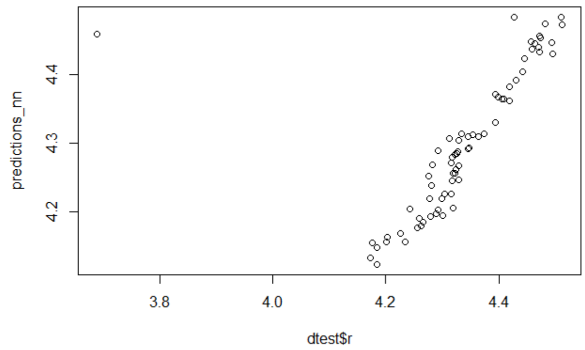
The average RMSE and R squared over the windows:

R2 = 0.99437;
RMSE = 0.03056259;

The RMSE and R squared of the model for predictions:

$R2 = 0.1749873$;
$RMSE = 0.1067433$;





8.1 The values put in the network versus the values given out by the network.

8.2 Predictions versus the test data.

The model has great performance on the dataset and very small error, but explains only 17% of the variance in the te st set and has much higher error on it. It clearly indicates overfit, but with better performance and of smaller magnitude than in the case of the shallow network.

Summary

The traditional regression methods have lower performance on the training data set than the trees and much higher errors. But the trees are prone to overfit and have negative R square metrics when it comes to new data. On the other hand, the errors of the trees are substantially smaller than in the traditional regressions. The extremely randomized trees show great performance on the training data, but have relatively high errors and worse performance than other regression trees techniques like random forests or boosted trees, which have better fit results, although are more prone to the overfitting. The shallow neural network is able to produce decent results, but in my case it has only 8 nodes with 12 predictors, which leads to the overfitting in the data. On the contrary, the deeper network with 16 nodes on the first layer and 8 nodes on the second layer does not overfit the data, but has higher error and better explanatory performance.