# AU_applied_machine_learning_health_science_execise

## Excercise Day 1

### Introduction

## Check point 1: Supervised learning example

*Example 1* Research questions: Could we predict HbA1c given various measurement of cholesterol, bmi, age, blood pressure, and sex.

Method: Linear regression

Data set: tabular data set with age, sex, body measures and biochemical measures.

*Example 2*

## Check point 2:

Find part from ECG nodes in ECG data to detect early high risk of CVD event. Mixed inference and prediction. As ECG have large variability, I believe to include many recordings ($> 1000$).

Methods: Deep learning neural network ?

Data: Time to event data - ECG data for each individuals - Hospital records of CVD events

**Checkpoint 4: Ï) Linear regression - body density data In this checkpoint you will analyse the body density data using linear regression.**

The goal is to build a linear regression model to predict body density from chest circumference measurements. • Open the script main1a.m and review it to understand the different steps in the script. Some of the code needed to answer this checkpoint is already provided in the script, but you also need to do a bit of coding yourself.

Use the script main1a.m / main1a.R and your code to answer the following:

(a) Load the data set bodyMeasurementsSingleTrainTest.txt. Describe the data, what are the sizes of the training- and test sets? how many observations and features? what is the numerical range of the variables?

*test = 244 obs with 2 variables train = 8 obs with 2 variables*

(b) Identify the part for of the script that is used for creating the polynomial expansion of the input variable. Try do create polynomial expansions with different polynomial order from 1 to 7, how many columns/features are there in the resulting data matrices?

*n+1 for each polynomial increase (1 polynum = 1 features and 2 polynum = 2 features)*

(c) Part of the code is scaling the individual columns of the polynomial regressors. Explain how this scaling is done and why the scaling may be necessary (Hint: what is the numerical range of the individual columns?).

(d) Type doc fitglm in Matlab or ?glm in R and use a bit of time to familiarize yourself with this function. What are the inputs to the function and what is the output?

*GLM is generalised linear model, and can be used for a variety of regression based defined familiy. The gaussian is used for linear regression with option for polynomial flexibility. The inputs are continuous and the output is a predicted continuous response.*

(e) Explain how training error and test error are quantified in the script?

*Training error is the mean of the squared difference between errors, and is based on the difference between observed value from the training data and predicted value from the training data set.*

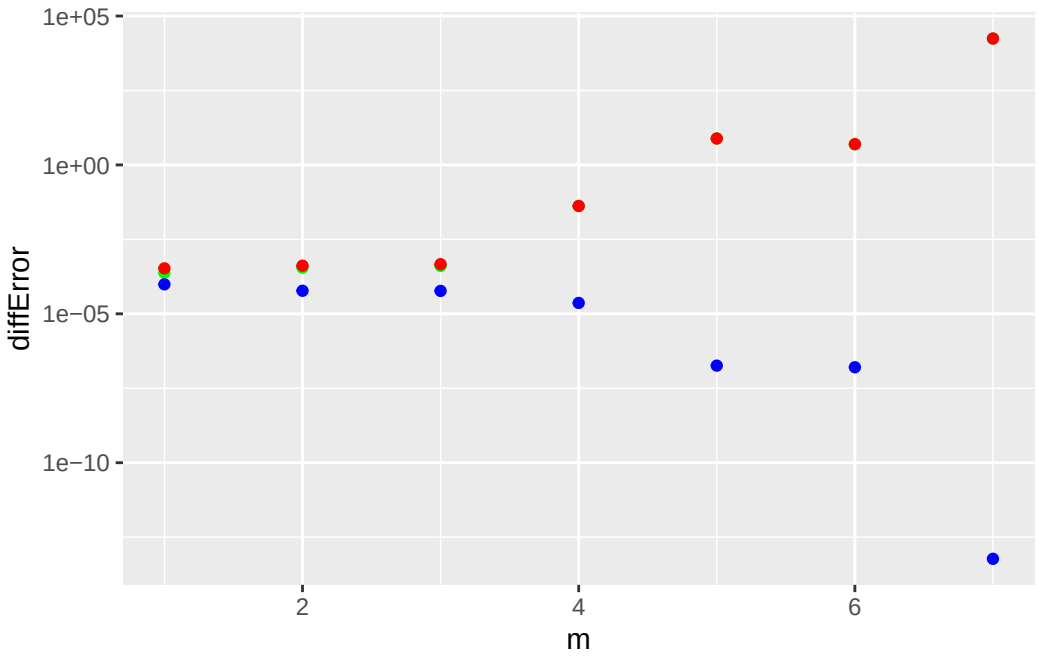
*Test error is the mean of the squared difference between errors, and is based on the difference between observed value from the test data set and predicted value from the training dataset.*

(f) Run the analysis with polynomial order ranging from 1 to 7 (Hint: include a for-loop in the script). For each of these seven models, make a plot of body density (ordinate) vs. chest circumference (abscissa) for the training- and test data as well as the model's predictions (all three in the same plot). Include the plots in your report and describe/discuss the plots.

##ADDPLOT!!!

(g) Write your own code to make a plot of training error and test error vs. polynomial order (order 1 to 7). Include the plot in your report and describe/discuss the plot. Do you observe severe overfitting for some polynomial orders?

```
plot(MSE_error_plot)
```




```
head(MSE_poly,7)
```

```
  m     errTrain      errTest i    diffError logerrTest logerrTrain
1 1 9.783850e-05 3.363816e-04 1 2.385431e-04  -7.997264   -9.232192
2 2 5.942656e-05 4.129855e-04 2 3.535590e-04  -7.792098   -9.730769
3 3 5.884494e-05 4.641982e-04 3 4.053533e-04  -7.675199   -9.740605
4 4 2.318943e-05 4.177533e-02 4 4.175214e-02  -3.175449  -10.671814
```

```
5 5 1.818859e-07 7.715585e+00 5 7.715585e+00   2.043242  -15.519886
6 6 1.605274e-07 4.971764e+00 6 4.971764e+00   1.603775  -15.644801
7 7 5.927753e-14 1.756514e+04 7 1.756514e+04   9.773672  -30.456546
```

## Checkpoint 5: b Training- and test errors

(a) Explain the difference between a training- and a test data. Also explain the difference between training- and test error.

The training data is used for training the models for predicting a certain outcome. The test data is used to test the trained model, to evaluate how well the trained model are predicting. The tools for evaluate the models performance is based on training and testing error. The training error are the error from the devolped model on the trained data point, where it is the same for testing, just focusing on the testing data points.

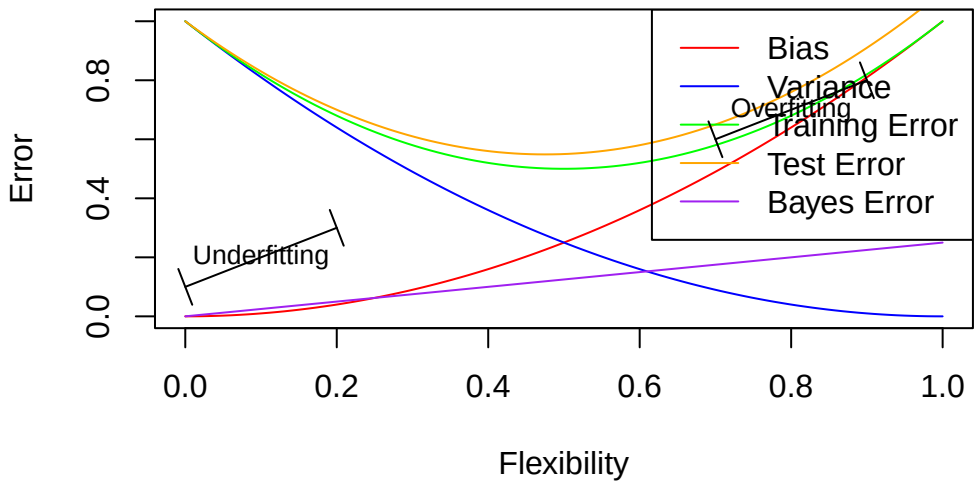
(b) Argue why we typically are interested in good model performance in terms of low test error rather than in terms of low training error.

We like to prioritize low error test results because the fitted model true evaluation depends how it fit with the validation points from test data set.

## Checkpoint 6: b Do section 2.4 conceptual exercise 3. in ISL (ISL page 53).

3. We now revisit the bias-variance decomposition.

(a) Provide a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.




**(b) Explain why each of the five curves has the shape displayed in part (a).**

```
As training error decreases because of better fitting, however overfitting is introduced and
```

## Checkpoint 7:

Explain, in your own words, what a training set is, a validation set is, and a test set is? Why is this partition of data needed? Explain how k-fold cross validation is implemented. Explain how leave-one-out (LOO) cross validation is implemented. What are the advantages and disadvantages of k-fold cross validation relative to the validation set approach and the LOO cross validation approach?

```
Training set is data that is used for training the prediction model, where validation is used
```

## Checkpoint 8:

R users: By using the functions createDataPartition and createFolds we can create random partitions of data.

• Look at the help text for these functions and use a bit of time to familiarize yourself with the functions.

(a) Run the command c = createDataPartition(y = 1:20, p = 0.8) and explain what the function does. Also describe the content of the variable c.

*It creates a list of numbers (series) of partition from 1 - 20*

(b) Run the command c = createFolds(y = 1:20, k = 10, returnTrain = TRUE) and explain what the function does. Also explain the content of the variable c.

*CreateFold split up the data in k groups. Here c are devided into 10 k-folds with 20 vector outcomes.*

(c) Explain why it is generally recommended to run e.g. the command set.seed(0) before creating the random partitions.

*To set up the seed before deviding the dataset and doing the analysis, hence obtain reproduceability.*

## Check point 9

Linear regression - body density data - cross validation In this checkpoint you will analyse the body density data using linear regression (with polynomial regressors), and the goal is to build a linear regression model to predict body density from chest circumference measurements. You will use k-fold cross validation to evaluate model performance for different polynomial order.

• Open the script main1b.m and review it to understand the different steps in the script. Some of the code needed to answer this checkpoint is already provided in the script, but you also need to do a bit of coding yourself. Use the script main1b.m / main1b.R and your code to answer the following:

(a) Load the data set bodyMeasurementsSingleCV.txt. Describe the data, what is the size of the data set? how many observations and features?

*252 obs and 2 features*

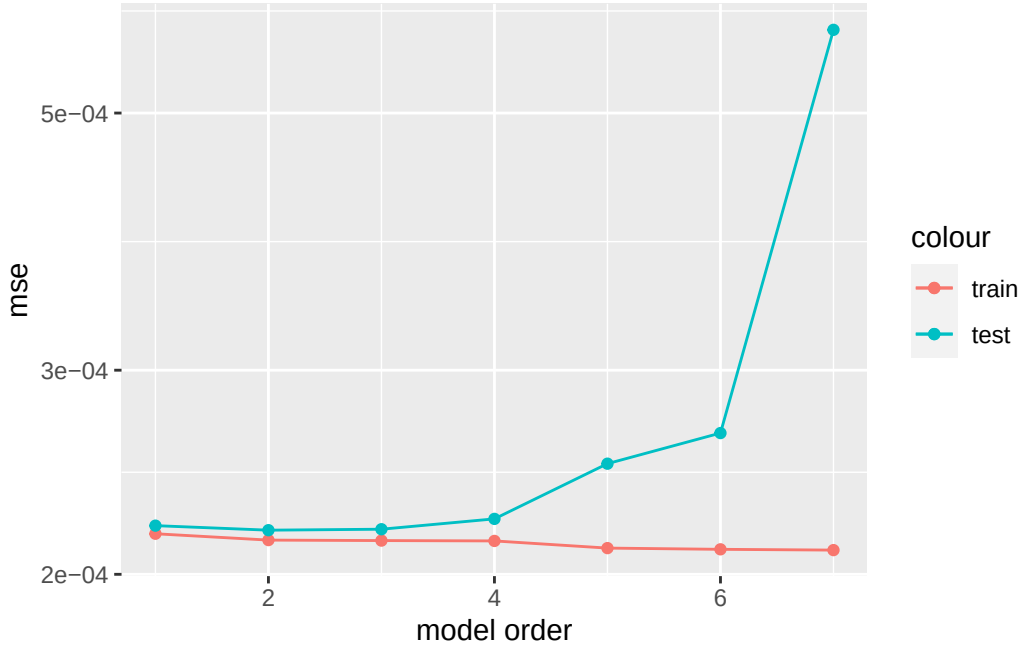
(b) Identify the part of the script where the random partition is performed. How many folds are used in the k-fold cross validation?

*10 k-folds*

(c) The script contains two for-loops (one nested within the other). Explain, in your own words, how the analysis is performed/structured in the script.

*The fist part of the loop is splitting the training data and this will be done K times (10 in this case). Each data set will be included another loop with conducting training a linear model with flexibility of order m ( up till 7 in this case). Each model will tested on each test data set and errors estimates from train and test evaluation will be extracted.*

(d) Run the script. Look at the plot of training and test error (cross-validation error) (ordinate) vs. polynomial order (abscissa) (both error curves in same plot). Include the plot in your report and describe/discuss it. Are the curves as expected? do you observe severe overfitting (compare with your result from Checkpoint 4)? if not, try to explain why not (Hint: look at the number of training observations and model flexibility). For which polynomial order do you observe the lowest test error?




*We do not observe the same severity as the earlier models. The overfitting seem first to be slightly introduced after 5 orders in polynomial. Again the best model would be using 2 orders in polynomial as this has the lowest value in MSEtest. The improvement of the model is expected as we are using more data for training dataset (90%) and we are including cross-validation.*

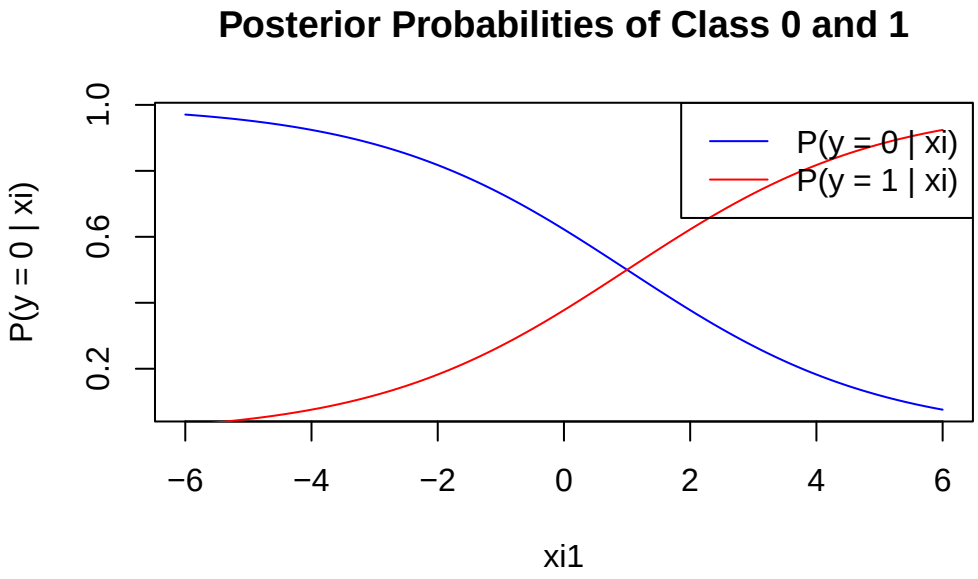
## Excercise Day 2

## Logistic Regression

### Checkpoint 10: The logistic regression model Suppose that your input data xi has a single predictor xi1, and suppose that the logistic regression model has parameters $_0 = 1$ and $_1 = 1$.

(a) Make a drawing/plot with curves of i) the posterior probability of class 0 P (y = 0|xi) as a function of xi1 and ii) the posterior probability of class 1 P (y = 1|xi) as a function of

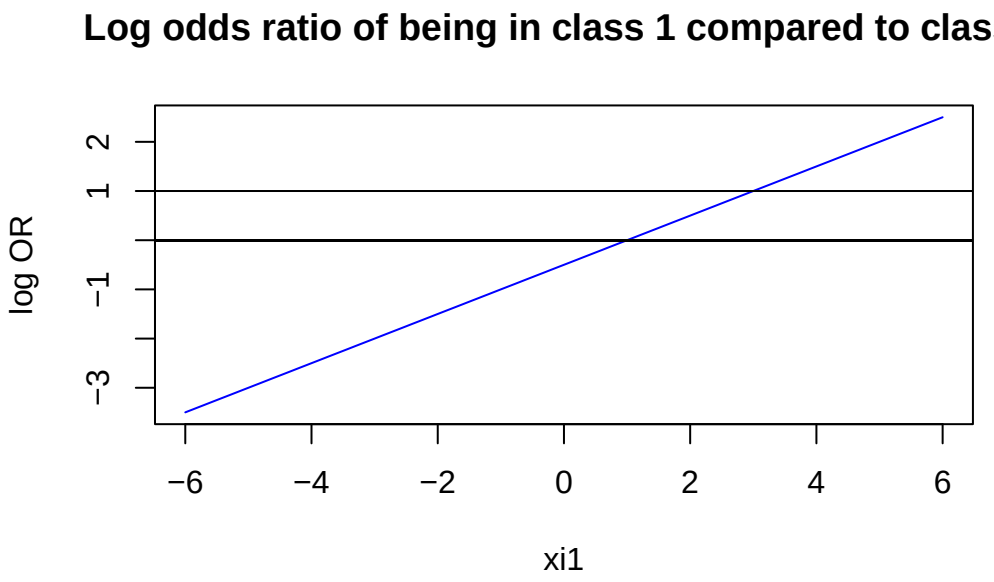
xi1, with xi1 ranging from -6 to 6. Remember to label each of the curves, to label axes in your drawing, and also remember to put tick labeling (numeric) on the axes.




(b) Explain how you can classify a given input xi by using the posterior probabilities, and indicate the decision boundary/threshold on your drawing/plot above.

*Based on xi will the classification be 0 from -6 to 1 because the have the highest probability wihch is > 0.5. Afterwards, beyond xi > 1 the probability for y=1 is >0.5, and hence will classify to category 1.*

(c) Make another drawing/plot with the log-odds ratio as a function of xi1.




(d) Explain how you can classify a given input xi by using the log-odds ratio, and indicate the decision boundary/threshold on your drawing/plot above.

*When OR are >1, the probability for being in class 1 is higher than the probability for being in class 0. Like the first plot the threshold can be found at xi = 1.*

(e) Suppose that we have three test samples For each of these three samples, compute P (y = 1|xi) and compute the predicted the class label.

[Three samples]here::here("doc/pics/Screenshot%202023-01-11%20at%2008.30.51.png"))

```
log_funk <- function(x) {
post_prob_1 <- exp(1+1*x)
post_prob_0 <- 1+post_prob_1
Y_Ix <- post_prob_1/post_prob_0

return(Y_Ix)
}
```

```
p_v <- c(-2,0,2)

for (i in p_v) {
  y_val <- log_funk(i)
```

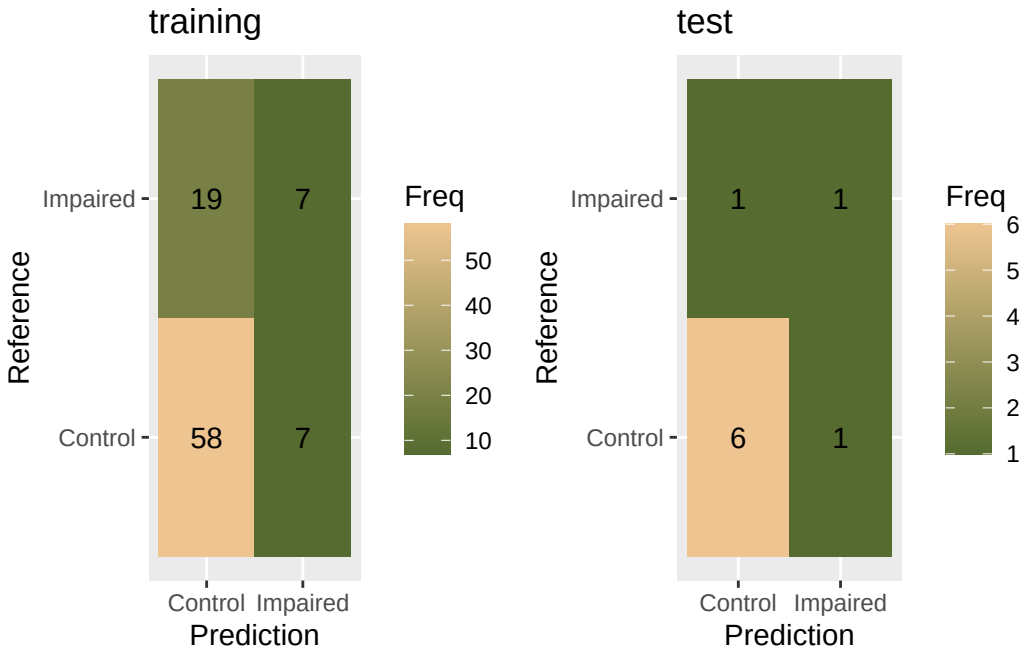
```
  print(y_val)
}
```

```
[1] 0.2689414
[1] 0.7310586
[1] 0.9525741
```

### Checkpoint 11: Logistic regression - CSF biomarker data

In this checkpoint you will analyse the CSF biomarker data using logistic regression. The goal of the analysis is to build a logistic regression model to predict group membership (control/impaired) from a single CSF feature tau. • Open the script main2a.m / main2a.R and review it to understand the different steps in the script.

```
source(here::here("R/day_2_code/main2a.R"))
```




(a) Run the first code section %% Import data etc.. What is the size of the data set? How many features and observations? Describe the response variable y, what type of variable is it and what is its content? How many subjects are there in each group?

*100 obs and 131 features*

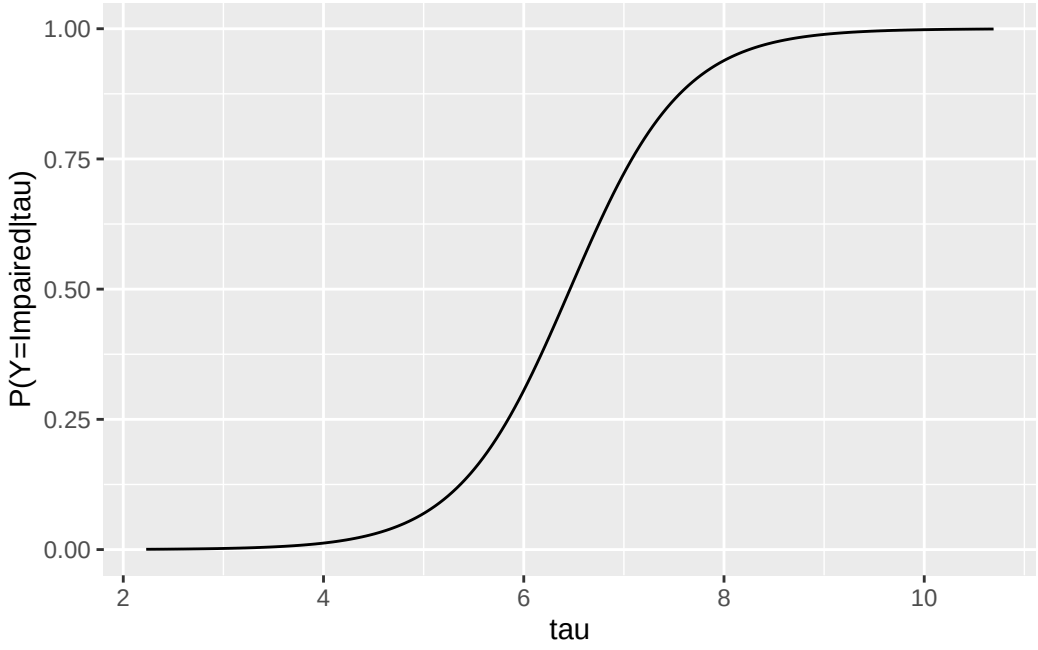
```
Control Impaired
     72       28
```

(b) Explain how data is divided into a training set and a test set, and explain the meaning of stratification. Run the second code section %% Divide into training and test sets. Compute the class proportions in the training set and in the test set and report these, and verify that class proportions are preserved after the data partitioning.

*By using the code createDataPartition, and specify that Y should be account for the two outcomes, and define that 90% of data goes to training.*

(c) Run the code section %% Train model, predict, and plot model. Describe the variable catInfo and its content. Describe the model outputs yhatTrainProb and yhatTestProb, what does these represent? Describe the variables yhatTrain and yhatTest, what do these represent? Include the plot of model output vs. input, describe the plot, and explain how classification can be performed based on this plot.

*catInfo is the variable including the labels of the group response (impaired/control). yhatTrainProb is the predicted groups in training data set based on the trained model, and yhatTestProb is the predicted groups in the testing data set based on the trained model (validation of the model). yhatTrain and yhatTest are rounded the probabilities of yhatTrainProb and yhatTestProb to a single digit, a 0 or 1 to be classified as imparied or control. Approximately, if the tau value was above 6.5 you had a higher probability for being in imparired compared to control*



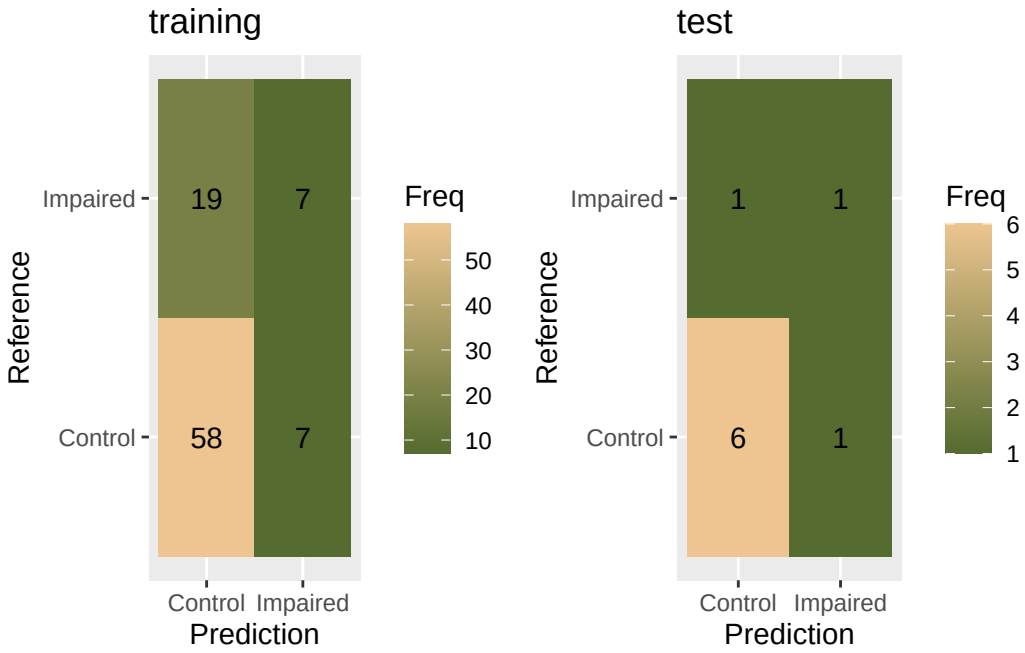
(d) In the third code section, the model predictions are converted to categorical data to be used as input to the confusionchart (Matlab) confusionMatrix (R) function for plotting the confusion matrix. Look at the help text for this function and use a bit of time to familiarize yourself with the function. Run the forth code section %% Plot confusion matrix, include the plot in your report, and describe/discuss it. Based on the numbers in the confusion matrix, compute and report the following performance metrics for the test set: classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity.

*When plotting the confusion matrix, we observe the the abbility of the trained model to predict outcome, hence its performance compared with th actual observation in both the traing and testing dataset, retrospectively*

*Based on visual observing the plots and estimates of classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity, the models performance are quite poor especially to detect observation with impaired status. This are also reflected by the low sensitivity and true positive predictive value. On the other hand the model performs well to detect, observations in control group, which are the reason the error rate accuracy does not have too bad performance.*

```
#|eecho: false

grid.arrange(cm_train_plot, cm_test_plot, ncol = 2)
```



```
#Training data
report_prec_train
```

```
     true_pr false_pr   true_nr  false_nr      sens      spec
[1,]     0.5      0.5 0.7532468 0.2467532 0.2692308 0.8923077
```

```
accTrain #accuracy
```

```
[1] 0.7142857
```

```
errTrain #error rate
```

```
[1] 0.2857143
```

```
# Testing data
report_prec_test
```

```
     true_pr false_pr   true_nr  false_nr sens      spec
[1,]     0.5      0.5 0.8571429 0.1428571  0.5 0.8571429
```

```
accTest #accuracy
```

```
[1] 0.7777778
```

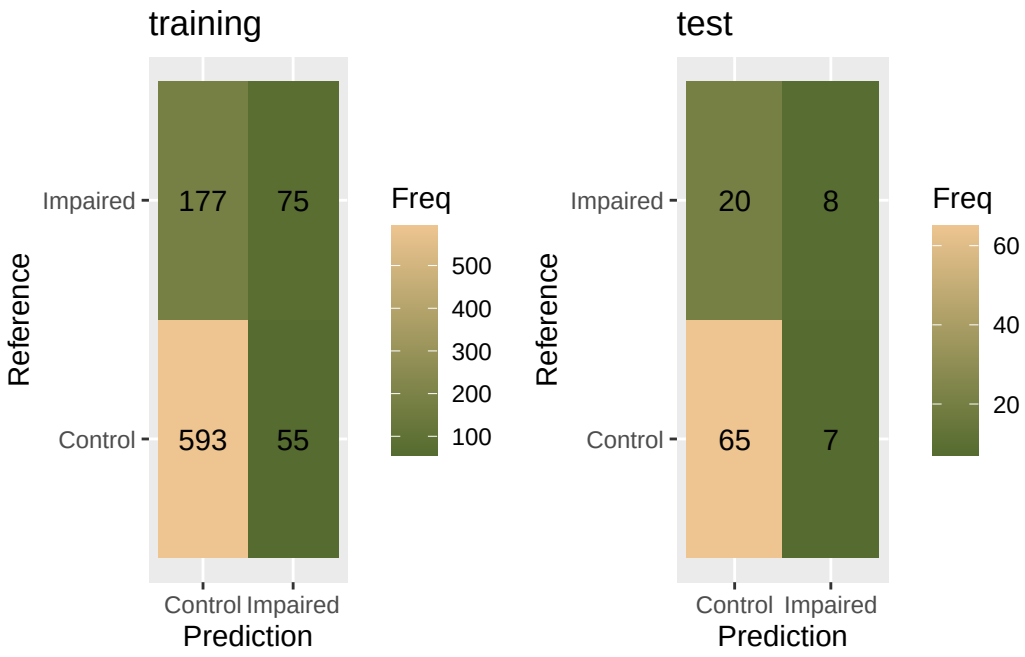
```
errTest #error rate
```

```
[1] 0.2222222
```

(e) You will now do almost the same analysis, but you will now use 10-fold cross-validation for model evaluation. Open the script main2b.m / main2b.R and review it to understand the different steps in the script. Run the script. Include the plot of the confusion matrices for the trainingand test/validation data in you report and describe/discuss it. Also explain how the confusion matrices are computed across the cross-validation iterations. Based on the numbers in the confusion matrix, compute and report the following performance metrics for the test set: classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity.

*When using the cross-validation, we are having a loop with K (10 in this case) different training and testing data set. Based on the loop, we estimate the average values from the loop estimates into the confusion matrix. The new developed model with cross-validation have higher accuracy and lower error rate in training data, compare to the simpler model from main2a, however the testing data are more imprecise compared to earlier model. Again, the model performs well, to find true control and have high specificity. The sensitivity was improved in the training, but performed horrible in the testing data set, with no sensitivity at all to capture impaired*

```
#|echo: false
grid.arrange(cm_train_plot,cm_test_plot, ncol = 2)
```




```
#Training data
report_prec_train
```

```
      true_pr  false_pr   true_nr  false_nr sens      spec
[1,] 0.6666667 0.3333333 0.7820513 0.2179487 0.32 0.9384615
```

```
accTrain[idx1] #accuracy
```

```
[1] 0.7666667
```

```
errTrain[idx1] #error rate
```

```
[1] 0.2333333
```

```
# Testing data
report_prec_test
```

```
     true_pr false_pr   true_nr  false_nr sens      spec
[1,]       0        1 0.6666667 0.3333333    0 0.8571429
```

```
accTest[idx1] #accuracy
```

```
[1] 0.6
```

```
errTest[idx1] #error rate
```

```
[1] 0.4
```

## Checkpoint 12: Regularization

(a) Explain when and why it may be necessary to use model regularization.

*To constrain the flexibility/complexity of the model and decrease the MSE. Here our goal is to minimize the error caused by variance and bias, and in the end improve the robustness of the models ability to predict, and prevent overfitting*

(b) (1) Write down the penalized cost function for linear regression for each of the following penalty/shrinkage terms: (i) ridge ( 2), (ii) lasso ( 1).

*In ridge, the values shrinkage close to zero and MSE decrease until a certain value. In lasso, some coefficients shrink to zero and the most important features are left.*

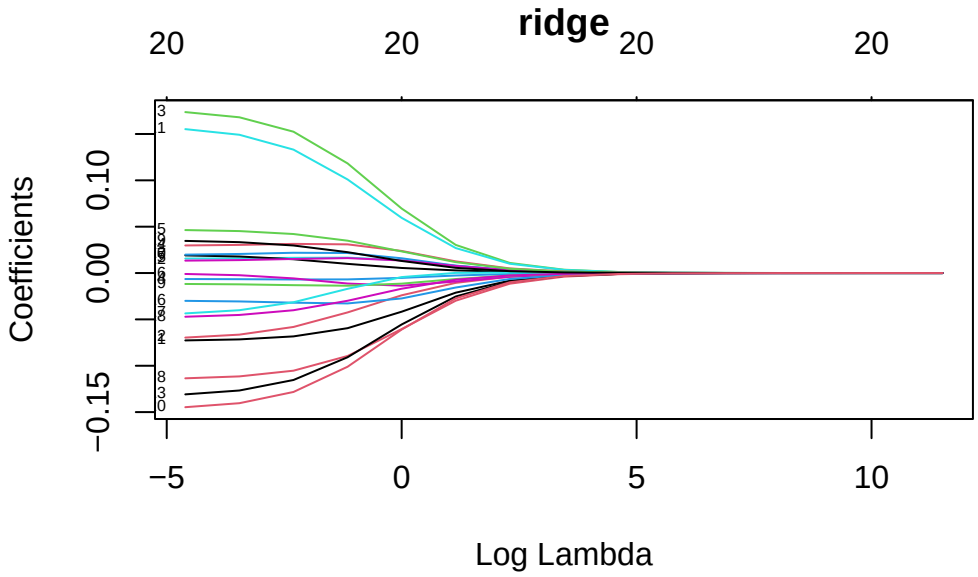
(2) Explain meaning of the different elements of the expressions in (1).

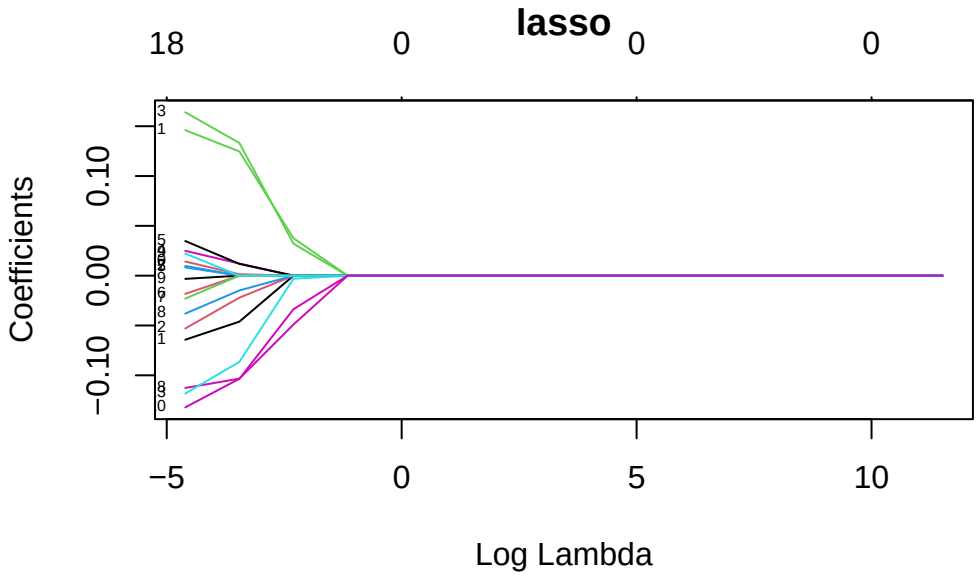
*The residual sum of sqaure that, sum each obesevation true value minus their predicted value.*

**Checkpoint 13: b Regularization, training- and test errors, and biasvariance trade-off**

(a) Provide a sketch of how coefficient estimates typically change with the strength of the regularization parameter  for the ridge and the lasso penalty, respectively. Describe/discuss the curves and their similarities/differences.

*Ridge / Lasso*






*Both regularization methods are shrinking values to improve model performance. In ridge, all coefficients are shrinkage collectively, where in lasso some coefficients are shrinkage to 0 and some increases in beta value.*