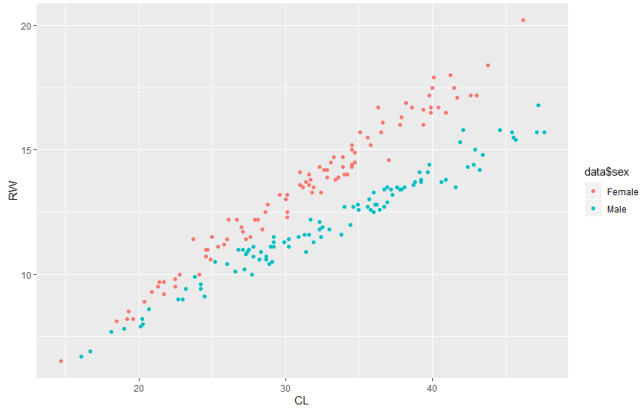
Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

Assignment 1 – LDA and Logistic Regression (vikgu708)

Task 1

```
ggplot(data,aes(x=data$CL,y=data$RW,col=data$sex)) + geom_point() + labs("Sex by RW&CL",x="CL",y="RW")
```



As the distributions are not clearly overlapping, an LDA method will be well suited to classify this data as the decision boundary can clearly split the two data sets.

Task 2

```
lda2 = lda(sex ~ RW + CL, data = data )
pred2 = predict(lda2,data)$class
table2 = table(pred2,data$sex)
mcr2 = mcr(table2)
```

```
> table2

pred2    Female Male
  Female     97    4
  Male        3   96
> mcr2
[1] 0.035
```

With only a 3.5% misclassification rate, it seems that this model is very well suited to classify the data.

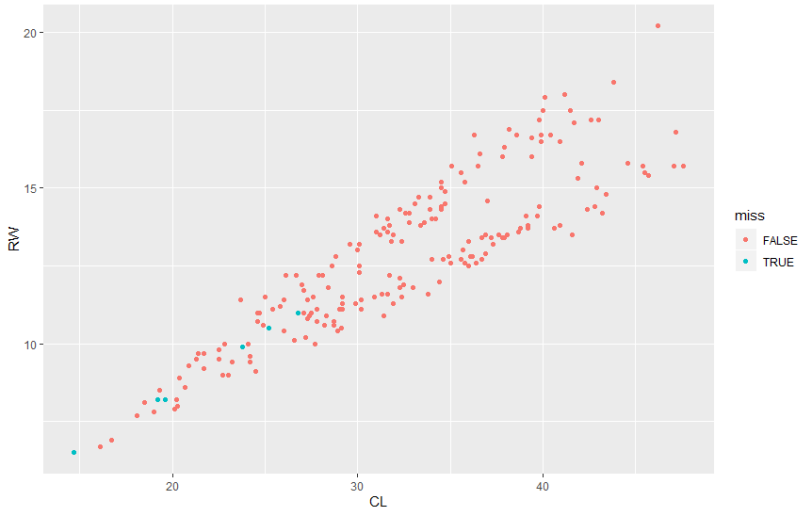
```
ggplot(data,aes(x=data$CL,y=data$RW,col=pred2)) + geom_point() + labs("LDA prediction",x="CL",y="RW")
```

The above chart is very similar to the original labels, which corresponds to the low classification error.

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

The misclassifications are found at clusters where data points of both labels are near each other's. The following plot shows the miss classifications colored in blue.

```
miss = !data$sex==pred2
ggplot(data,aes(x=data$CL,y=data$RW,col=miss)) + geom_point() + labs("LDA prediction",x="CL",y="RW")
```



Task 3

```
lda_prior = lda(sex ~ RW + CL, data = data, prior=c(0.1,0.9))
pred_prior = predict(lda_prior,data)$class
table_prior = table(pred_prior,data$sex)
mcr_prior = mcr(table_prior)
```

```
> table_prior

pred_prior Female Male
    Female     84    0
    Male       16  100
> mcr_prior
[1] 0.08
```

A prior that increases the prior likelihood of being a male (suggesting an overrepresentation of males in the population) will naturally produce a result where more observations are classified as male, which can be seen in the confusion table where 116 observations were classified as male compared to the last result without prior where 99 were classified as male.

Even with a very large prior distribution for male, the mcr is still quite small, at 8%. The amount of misclassified points has increased from 7 to 16. The only slight increase can be attributed to the high separation between the data points in the training data, which results in LDA approximating quite separated distributions for the two classes.

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

Task 4

```
Y = as.numeric(data$sex)-1
threshold = 0.5
logistic = glm(Y ~ RW + CL, data = data)
pred_glm = predict(logistic,data)
prediction = pred_glm
prediction[which(prediction>threshold)] = "Male"
prediction[which(prediction<=threshold)] = "Female"

table_glm = table(as.numeric(pred_glm>threshold),data$sex)
```

```
> table_glm

    Female Male
  0     97    4
  1      3   96
```

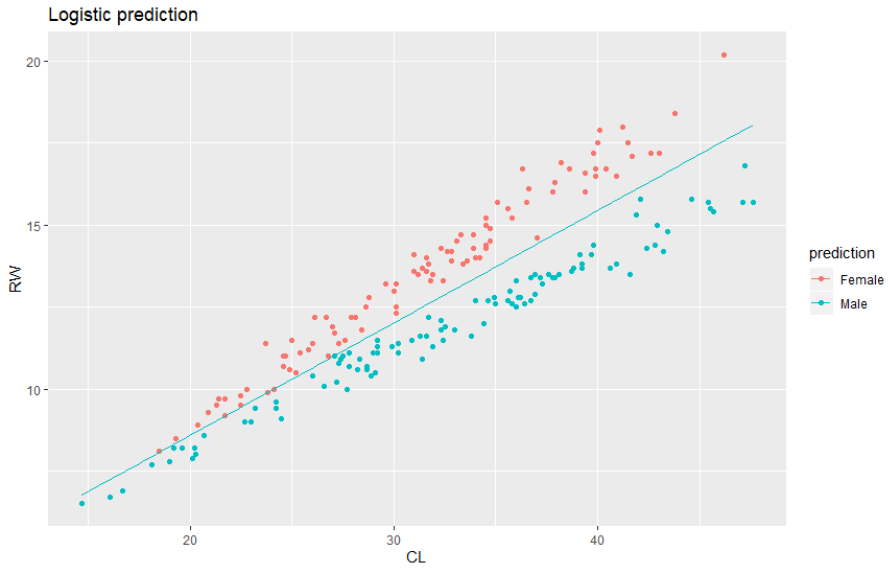
The misclassification result is the same as the LDA method.

As the decision boundary is the point the regression where both classifications have the same probability, the decision boundary equation is found from solving for RW in the equation where p=0,5.

$$p = w_1 + w_2 * RW + w_3 * CL$$

As CL is the x coordinate in our plot.

```
w = logistic$coefficients
decision = function(x)  (threshold-w[3]*x - w[1])/w[2]
glmplot + stat_function(fun=decision)
```



Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

Assignment 2 – analysis of credit scoring (felki349)

Task 1

Data was imported and divided

Task 2

Conf table and MCR for deviance model on test data

dev_pred_test	bad	good
bad	28	19
good	48	155

, MCR = 0.268

Conf table and MCR for deviance model on training data

gini_pred_test	bad	good
bad	18	33
good	58	141

, MCR = 0.212

Conf table and MCR for gini model on test data

gini_pred_test	bad	good
bad	18	33
good	58	141

, MCR = 0.364

Conf table and MCR for gini model on training data

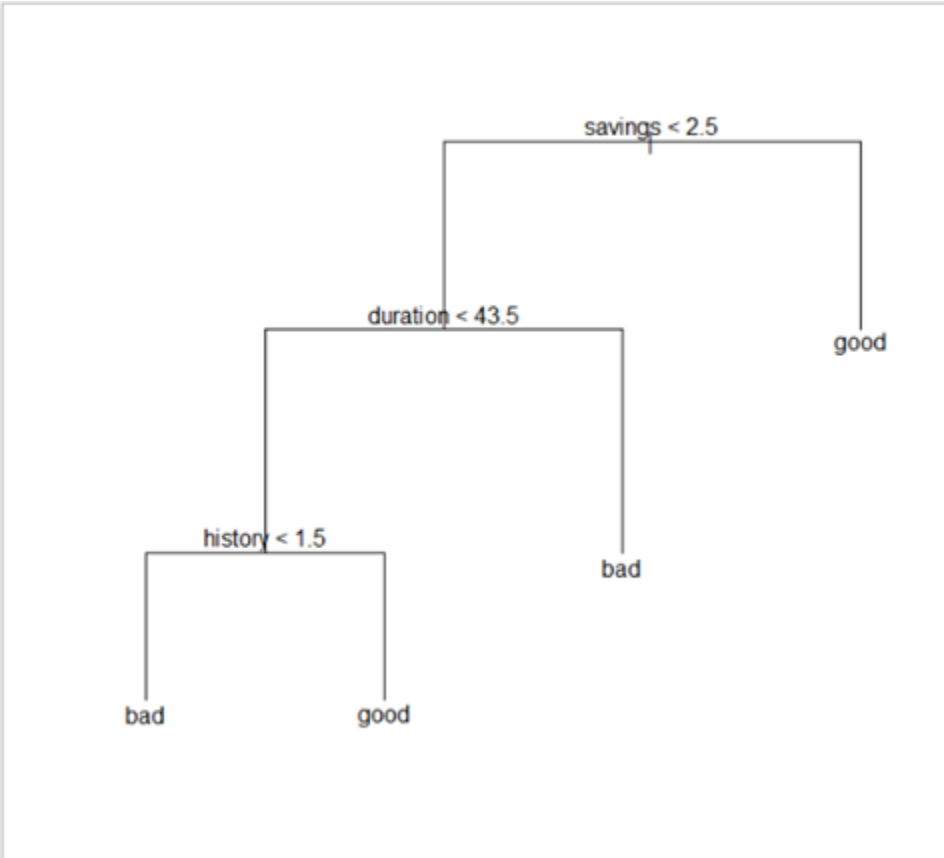
gini_pred_train	bad	good
bad	67	38
good	80	315

, MCR = 0.236

Task 3

Iterating over different tree depths and evaluating the score, deviance from training and validation data was used. To decide how much the tree should be pruned, the validation score was used, and the optimal amount of leaves were 4.

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708



The depth of the tree is 4 and the variables used are savings, duration and history. The misclassification rate on the testing data was 0.256. My interpretation of the tree structure is that savings is a very good indicator of whether the client is “good” or “bad”, and going forward, if the client doesn’t have any savings, the duration of the loan is a very important factor. If the duration is OK, then the final decision will be made on whether the client has a good history.

Task 4

Using the Naïve Bayes method on the training data, the results were as follows

```
naive_pred_train bad good
            bad   95   98
            good  52  255
```
, MCR = 0.3

Using the Naïve Bayes method on the test data, the results were as follows

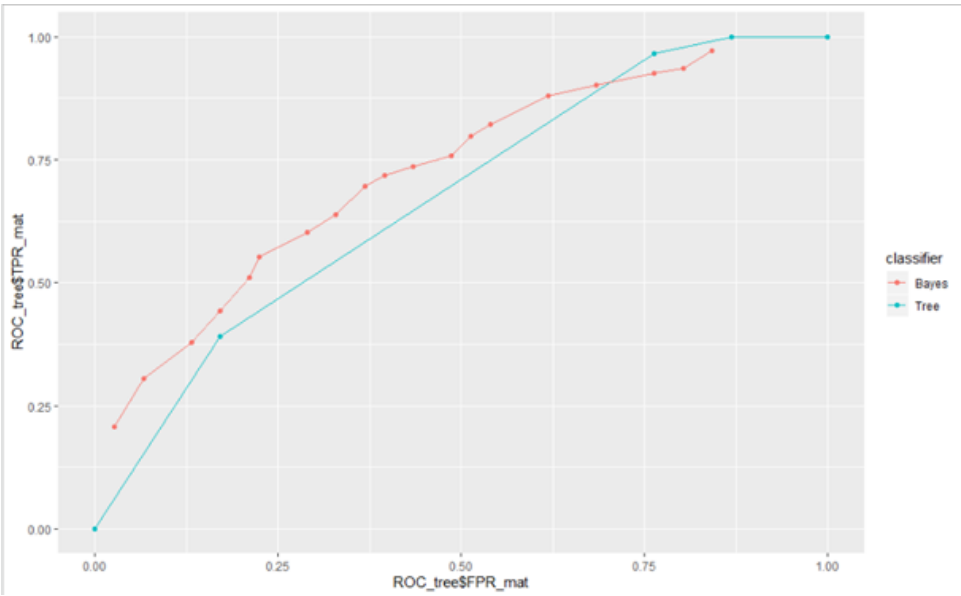
```
naive_pred_test bad good
           bad   46   49
           good  30  125
```
, MCR = 0.316

In comparison to step 3, the misclassification rate is higher.

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

Task 5

Computing the FPR and TPR for the models with the interval proposed in the exercise, the following is the resulting ROC-curve



The conclusion made is that the Naïve Bayesian method is better since the TPR to FPR ratio is generally better for the Naïve Bayesian method than the Tree (the area under the line is larger).

Task 6

The introduced loss matrix implies that it is much more expensive to have a false positive than a false negative.

```
bayes_pred_loss bad good
           bad   62  164
           good  14   10
```
, MCR = 0.712

```
naive_pred_test bad good
           bad   46   49
           good  30  125
```
, MCR = 0.316

In comparison to the Naïve Bayesian method without a loss matrix, the new fit will predict a lot more “bad” clients, and the since the loss is very high in a false positive (reasonable since this is credit scoring where a false positive means the client will default on the loan) the following MCR is very high – better safe than sorry.

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

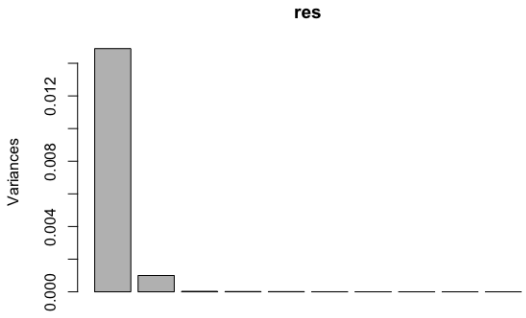
Assignment 4 – Principal Components (lukbo262)

Task 1

The following code is used to import the data to R and conduct a standard PCA:

```
library(fastICA)
RNGversion("3.5.1")

spectraData = read.csv2("NIRSpectra.csv")
data1 = spectraData
data1$Viscosity = c()

res = prcomp(data1)
lambda = res$sdev^2
sprintf("%2.3f", lambda/sum(lambda)*100)
screeplot(res)
plot(res$x[, 1], res$x[, 2], xlab = "PC1", ylab = "PC2")
```

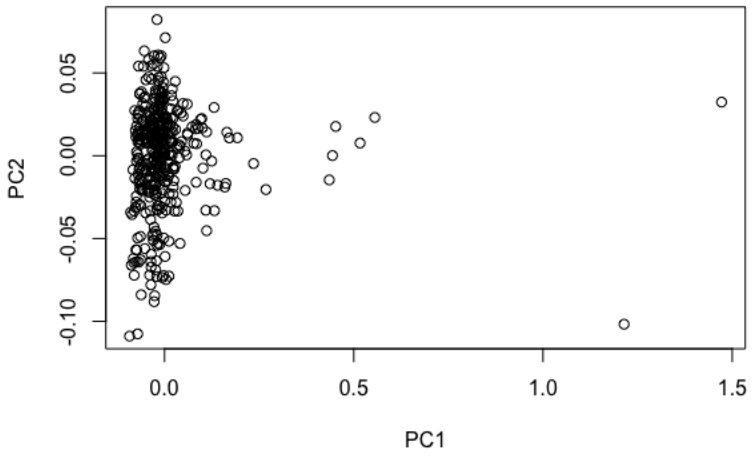
Below to the left is a plot showing how much variation is explained by each feature. It is evident that two features explain the vast majority of the dataset's variance, making it appropriate to remove the less-contributing features. Looking at the table below to the right, we can see that feature 1 and 2 account for 99.595 % of the total variance. Henceforward, these will be our principal components PC1 and PC2.




```
  [1] "93.332" "6.263"  "0.185"  "0.101"  "0.068"  "0.025"  "0.009"  "0.003"
  [9] "0.003"  "0.002"  "0.001"  "0.001"  "0.001"  "0.001"  "0.000"  "0.000"
 [17] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [25] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [33] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [41] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [49] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [57] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [65] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [73] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [81] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [89] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
 [97] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
[105] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
[113] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
[121] "0.000"  "0.000"  "0.000"  "0.000"  "0.000"  "0.000"
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

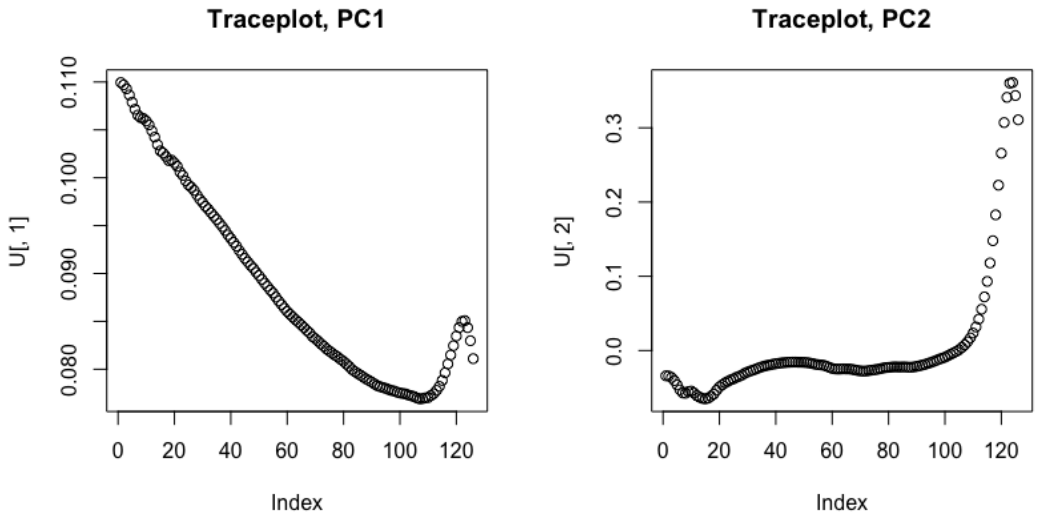
Below is a plot of the scores, using PC1 and PC2 as coordinates. Most observations are somewhat clustered, but there are a few outliers located to the right in the plot. These could be considered "unusual" diesel fuels or they could be badly measured observations.



Task 2

```
U = res$rotation
plot(U[, 1], main="Traceplot, PC1")
plot(U[, 2], main="Traceplot, PC2")
```

Below are the trace plots for PC1 and PC2 generated by the above R code. We can discern that PC1 is explained mostly by original features with lower indexes, even though there are few features with higher indexes offering some explanation as well. It is evident that PC2 is explained by mainly a few of the original features. These are the features with indexes between approximately 115 and 125.



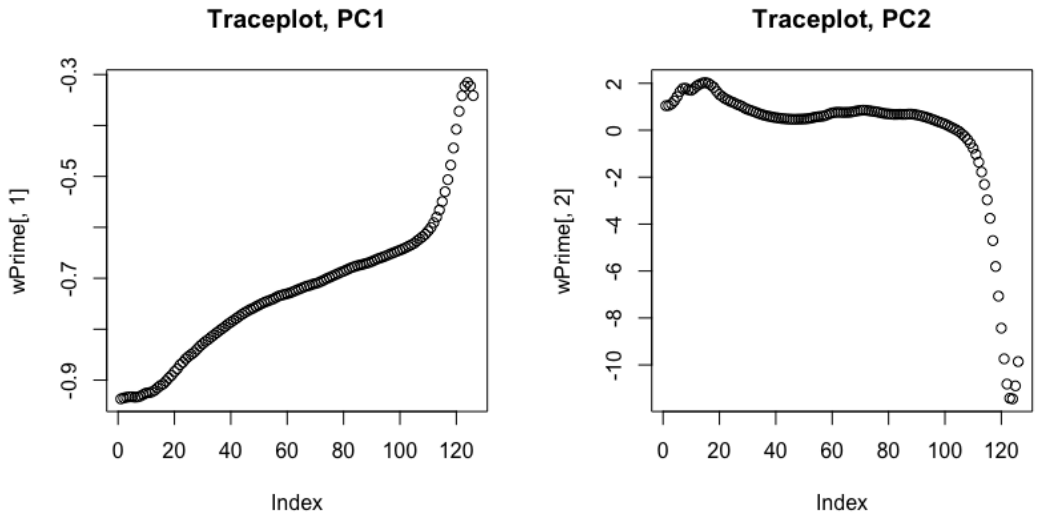
Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

Task 3

The code below was used to perform the ICA and plotting:

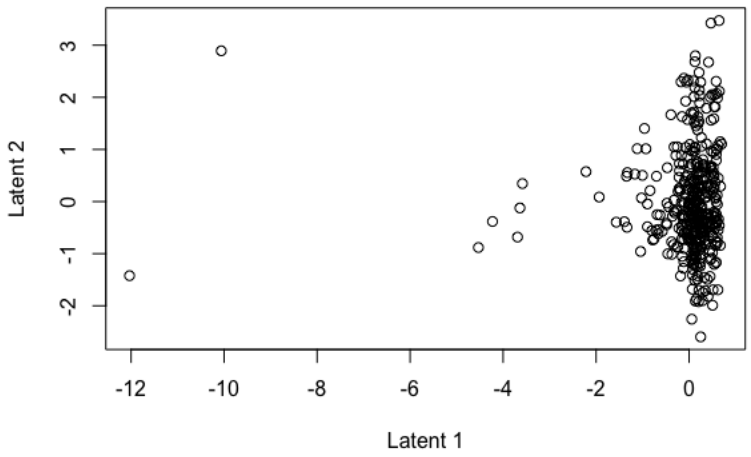
```
set.seed(12345)
ica = fastICA(data1, n.comp = 2, alg.typ = "parallel", fun = "logcosh", alpha = 1,
              method = "R", row.norm = F, maxit = 200, tol = 0.0001, verbose = T)
wPrime = ica$K %*% ica$W
plot(wPrime[, 1], main="Traceplot, PC1")
plot(wPrime[, 2], main="Traceplot, PC2")

plot(ica$S[, 1], ica$S[, 2], xlab = "Latent 1", ylab = "Latent 2")
```

Below are the trace plots for PC1 and PC2. These plots are much alike the previous ones, but we can see that they are both essentially amplified and mirrored on the x-axis. The matrix *W'* shows how dependent a feature is on the given principal components.



Below is a plot of the score, using the two first latent features. Compared to the earlier score plot, we can see that the data is mirrored and amplified here as well.

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708



Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

Appendix

Assignment 1

```
library("MASS")

library("ggplot2")


data = read.table("australian-crabs.csv",header=TRUE,sep=",")


n=dim(data)[1]

set.seed(12345)

id=sample(1:n, floor(n*0.5))

train=data[id,]

test=data[-id,]


mcr = function (table){

  return (1-sum(diag(table))/sum(table))

}


#1.

ggplot(data,aes(x=data$CL,y=data$RW,col=data$sex)) + geom_point() + labs("Sex by
RW&CL",x="CL",y="RW")




#2.

lda2 = lda(sex ~ RW + CL, data = data )

pred2 = predict(lda2,data)$class

table2 = table(pred2,data$sex)

mcr2 = mcr(table2)
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
miss = !data$sex==pred2

ggplot(data,aes(x=data$CL,y=data$RW,col=pred2)) + geom_point() + labs("LDA
prediction",x="CL",y="RW")


#3

lda_prior = lda(sex ~ RW + CL, data = data, prior=c(0.1,0.9))

pred_prior = predict(lda_prior,data)$class

table_prior = table(pred_prior,data$sex)

mcr_prior = mcr(table_prior)

table_prior

mcr_prior

miss = !data$sex==pred_prior


ggplot(data,aes(x=data$CL,y=data$RW,col=pred_prior)) + geom_point() + labs(title="LDA w
Prior",x="CL",y="RW")


#4

Y = as.numeric(data$sex)-1

threshold = 0.5

logistic = glm(Y ~ RW + CL, data = data)

pred_glm = predict(logistic,data)

prediction = pred_glm

prediction[which(prediction>threshold)] = "Male"

prediction[which(prediction<=threshold)] = "Female"


pred_glm>threshold


glmplot = ggplot(data,aes(x=data$CL,y=data$RW,col=prediction)) + geom_point() +
labs(x="CL",y="RW",title="Logistic prediction")
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
table_glm = table(as.numeric(pred_glm>threshold),data$sex)

mcr(table_glm)


w = logistic$coefficients

decision = function(x)  (threshold-w[3]*x - w[1])/w[2]

glmplot + stat_function(fun=decision)



summary(lda2)
```

Assignment 2

```
library(readxl)

library(tree)

library(e1071)

library(MASS)

library(ggplot2)

library(pROC)

RNGversion("3.5.1")



mcr = function (table){

  return (1-sum(diag(table))/sum(table))

}


## 1


data = read_excel("creditscoring.xls")


n=dim(data)[1]
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
RNGversion("3.5.1")

set.seed(12345)

id=sample(1:n, floor(n*0.5))

train=data[id,]


id1=setdiff(1:n, id)

RNGversion("3.5.1")

set.seed(12345)

id2=sample(id1, floor(n*0.25))

valid=data[id2,]


id3=setdiff(id1,id2)

test=data[id3,]


## 2


deviance_model = tree(as.factor(good_bad) ~ ., data = train, split = "deviance")

gini_model = tree(as.factor(good_bad) ~ ., data = train, split = "gini")


RNGversion("3.5.1")

set.seed(12345)

dev_pred_train = predict(deviance_model, newdata = train, type = "class")

RNGversion("3.5.1")

set.seed(12345)

dev_pred_test = predict(deviance_model, newdata = test, type = "class")

RNGversion("3.5.1")

set.seed(12345)

gini_pred_train = predict(gini_model, newdata = train, type = "class")
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
RNGversion("3.5.1")

set.seed(12345)

gini_pred_test = predict(gini_model, newdata = test, type = "class")


summary(deviance_model)

summary(gini_model)


conf_dev = table(dev_pred_test, test$good_bad)

conf_gini = table(gini_pred_test, test$good_bad)

conf_dev_train = table(dev_pred_train, train$good_bad)

conf_gini_train = table(gini_pred_train, train$good_bad)


conf_dev

conf_gini

conf_dev_train

conf_gini_train


mcr(conf_dev)

mcr(conf_gini)

mcr(conf_dev_train)

mcr(conf_gini_train)
```

3

```
score_train = rep(0,9)

score_valid = rep(0,9)
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
# Loop over different tree depths


for (i in 2:9) {
  tree = prune.tree(deviance_model, best=i)
  RNGversion("3.5.1")
  set.seed(12345)
  prediction = predict(tree, newdata = valid, type = "tree") # Predict with the pruned tree and val set
  score_train[i] = deviance(tree) # Calculate deviance of test set
  score_valid[i] = deviance(prediction) # Calculate deviance of val set
}


plot(2:9, score_train[2:9], type="b", col="red", ylim=c(250,600), ylab="Deviance", xlab="Leaves") # Plot deviance to number of leaves
points(2:9, score_valid[2:9], type="b", col="blue")
legend("top", legend=c("Train = red", "Valid = blue"))


optimal_leaves = match(min(score_valid[2:9]), score_valid)


plot(deviance_model) # Plot non-pruned tree
text(deviance_model, pretty=0)


optimal_tree = prune.tree(deviance_model, best=4) # Prune tree and plot
plot(optimal_tree)
text(optimal_tree, pretty=0)


# Predict with pruned tree


RNGversion("3.5.1")
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
set.seed(12345)

optimal_tree_predict = predict(optimal_tree, newdata = test, type = "class")

conf_optimal_tree = table(optimal_tree_predict, test$good_bad)

conf_optimal_tree

mcr(conf_optimal_tree)
```

4

```
naive_bayes = naiveBayes(as.factor(good_bad) ~ ., data = train)

print(naive_bayes)

RNGversion("3.5.1")

set.seed(12345)

naive_pred_train = predict(naive_bayes, newdata = train)

naive_pred_test = predict(naive_bayes, newdata = test)

conf_naive_train = table(naive_pred_train, train$good_bad)

conf_naive_test = table(naive_pred_test, test$good_bad)

conf_naive_train

conf_naive_test

mcr(conf_naive_train)

mcr(conf_naive_test)
```

5

```
RNGversion("3.5.1")

set.seed(12345)
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
raw_naive_pred_train = predict(naive_bayes, newdata = train, type = "raw")

RNGversion("3.5.1")

set.seed(12345)

raw_naive_pred_test = predict(naive_bayes, newdata = test, type = "raw")

RNGversion("3.5.1")

set.seed(12345)

raw_optimal_pred_train = predict(optimal_tree, newdata = train, type = "vector")

RNGversion("3.5.1")

set.seed(12345)

raw_optimal_pred_test = predict(optimal_tree, newdata = test, type = "vector")



ROCMatrix = function(pred_vec, pi_vec){
  TPR_mat = matrix(nrow = 19, ncol = 1)
  FPR_mat = matrix(nrow = 19, ncol = 1)

  positives = sum(test$good_bad == "good")
  negatives = sum(test$good_bad == "bad")

  index = 0

  for(i in pi_vec) {
    index = index + 1
    classification = ifelse(pred_vec[, 2] > i, "good", "bad")
    levs = sort(union(classification, test$good_bad))
    conf_t = table(factor(classification,levs), factor(test$good_bad, levs))

    TP = conf_t[4]
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
    FP = conf_t[2]


    TPR = (TP / positives)
    FPR = (FP / negatives)
    TPR_mat[index, 1] = TPR
    FPR_mat[index, 1] = FPR
  }
  return(data.frame(pi_vec, TPR_mat, FPR_mat))
}


pi_vec = seq(0.05, 0.95, 0.05)
ROC_bayes = ROCMatrix(raw_naive_pred_test, pi_vec)
ROC_tree = ROCMatrix(raw_optimal_pred_test, pi_vec)

ROC_tree$TPR_mat
ROC_tree$FPR_mat
ROC_bayes$TPR_mat
ROC_bayes$FPR_mat

ggplot(data = NULL, aes(col = classifier)) +
  geom_point(data = ROC_tree, aes(x = ROC_tree$FPR_mat, y = ROC_tree$TPR_mat, col="Tree")) +
  geom_line(data = ROC_tree, aes(x = ROC_tree$FPR_mat, y = ROC_tree$TPR_mat, col="Tree")) +
  geom_point(data = ROC_bayes, aes(x = ROC_bayes$FPR_mat, y = ROC_bayes$TPR_mat, col="Bayes"))
+
  geom_line(data = ROC_bayes, aes(x = ROC_bayes$FPR_mat, y = ROC_bayes$TPR_mat, col="Bayes"))
```

6

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
loss_mat = matrix(c(0,1,10,0), nrow = 2, ncol = 2)

bayes_pred_loss = ifelse((raw_naive_pred_test[,2]/raw_naive_pred_test[,1]) <
(loss_mat[2,1]/loss_mat[1,2]), "good", "bad")

conf_bayes_loss = table(bayes_pred_loss, test$good_bad)

conf_bayes_loss

mcr(conf_bayes_loss)
```

Assignment 4

```
library("ggplot2")

library("fastICA")


set.seed(12345)


data = data.frame(read.csv2("NIRSpectra.csv"))

feats = data

feats$Viscosity = c()


pca =prcomp(x = feats, scale. = TRUE, center =TRUE)


feats

plot(pca, type = "l" )


# PC1 + PC2 explains >99% of the variance

qplot(x = pca$x[,1], y = pca$x[,2])


# 2.2

plot(pca$rotation[,1], main =" PC1 traceplot")

plot(pca$rotation[,2], main ="PC2 traceplot")


# 2.3
```

Felix Kimiaei – felki349
Lukas Borggren – lukbo262
Viktor Gustafsson – vikgu708

```
RNGversion("3.51")

set.seed(12345)

ICA = fastICA(X = feats, 2, fun = "logcosh",   alpha = 1, method = "R", maxit = 200, alg.typ = "parallel", tol = 0.0001, row.norm = FALSE)

W = ICA$K %*% ICA$W

plot(W[,1], main = " W' 1",col="blue")

plot(W[,2], main = " W' 2",col="red")

qplot(x = a$S[,1], y = a$S[,2], xlab = "Latent feat 1", ylab = "Latent feat 2" , data = feats)
```