Assignment 01

Marc Braun

10/01/2022

Assignment 01

Task 01

The first three principal components explain about 81.29% of the variation in the data and the first four are enough to explain 91.20577% of variation in the data. Therefore, the first four principal components are needed to explain at least 90% of variation in the data.

```
## [1] "The mean of palmitic is 1231.74125874126"
```

```
## [1] "The mean of linolenic is 31.8881118881119"
```

```
## [1] "The variance of palmitic is 28423.3514996387"
```

```
## [1] "The variance of linolenic is 168.187108863116"
```

The features in the data have very different means and variances (which can be seen with the two given examples above). Therefore, scaling the data allows the PCA to explain the relevant variation in the data, as otherwise it would overestimate the importance of features with a high variance.

Task 02

```
## # weights:  15 (8 variable)
## initial  value 314.203115
## iter  10 value 88.100326
## iter  20 value 52.996752
## final  value 52.987997
## converged
```

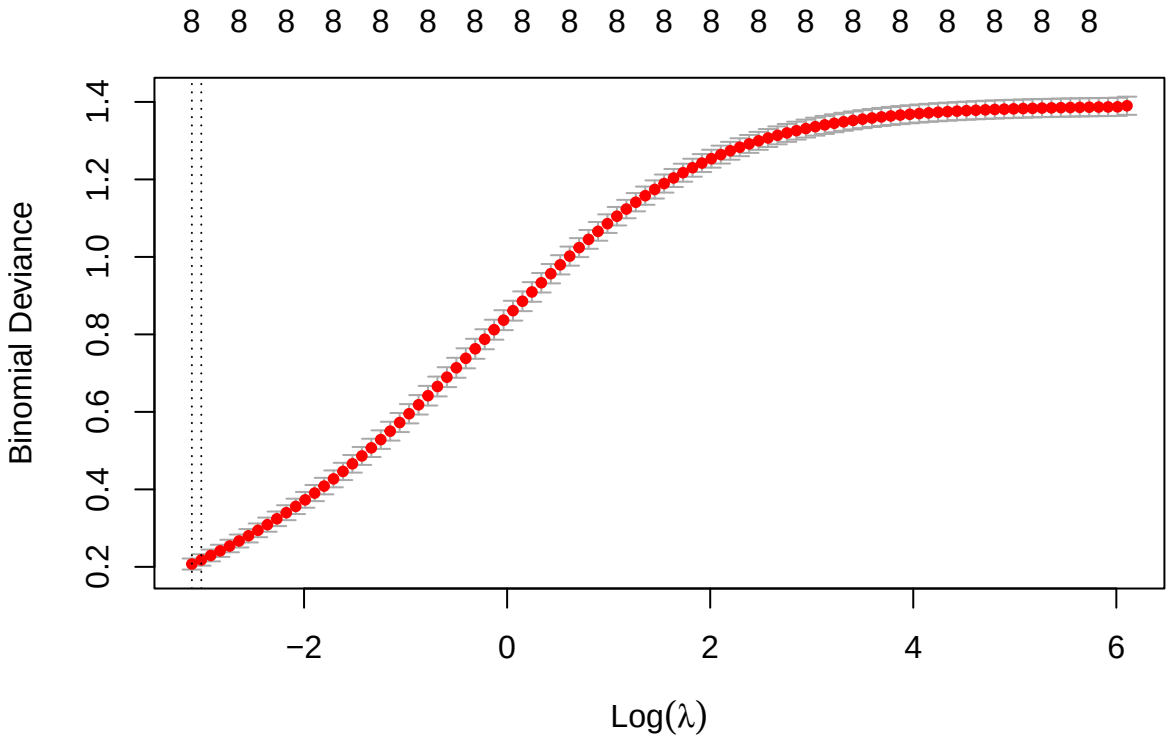
The training misclassification rate is 0.05944056 and the test misclassification rate is 0.07342657. So for the test data only about 7% of data is misclassifide which is a relatively low value, considering that only the first three principal components were used as features in the model. print(model) The decision boundary between region 1 and 2 is $-3.218617-2.992163\cdot Comp.1+3.187718\cdot Comp.2-4.000276\cdot Comp.3=0$ where the data is classified as class 1 if $-3.218617-2.992163\cdot Comp.1+3.187718\cdot Comp.2-4.000276\cdot Comp.3<0$. The decision boundary between region 1 and 3 is $-5.948747-5.677377\cdot Comp.1+2.827254\cdot Comp.2-3.045757\cdot Comp.3=0$ where the data is classified as class 1 if $-5.948747-5.677377\cdot Comp.1+2.827254\cdot Comp.2-3.045757\cdot Comp.3<0$. The decision boundary between region 2 and 3 is $-2.73013-2.685214\cdot Comp.1-0.360464\cdot Comp.2+0.954519\cdot Comp.3=0$ where the data is classified as class 2 if $-2.73013-2.685214\cdot Comp.1-0.360464\cdot Comp.2+0.954519\cdot Comp.3<0$.

Task 03

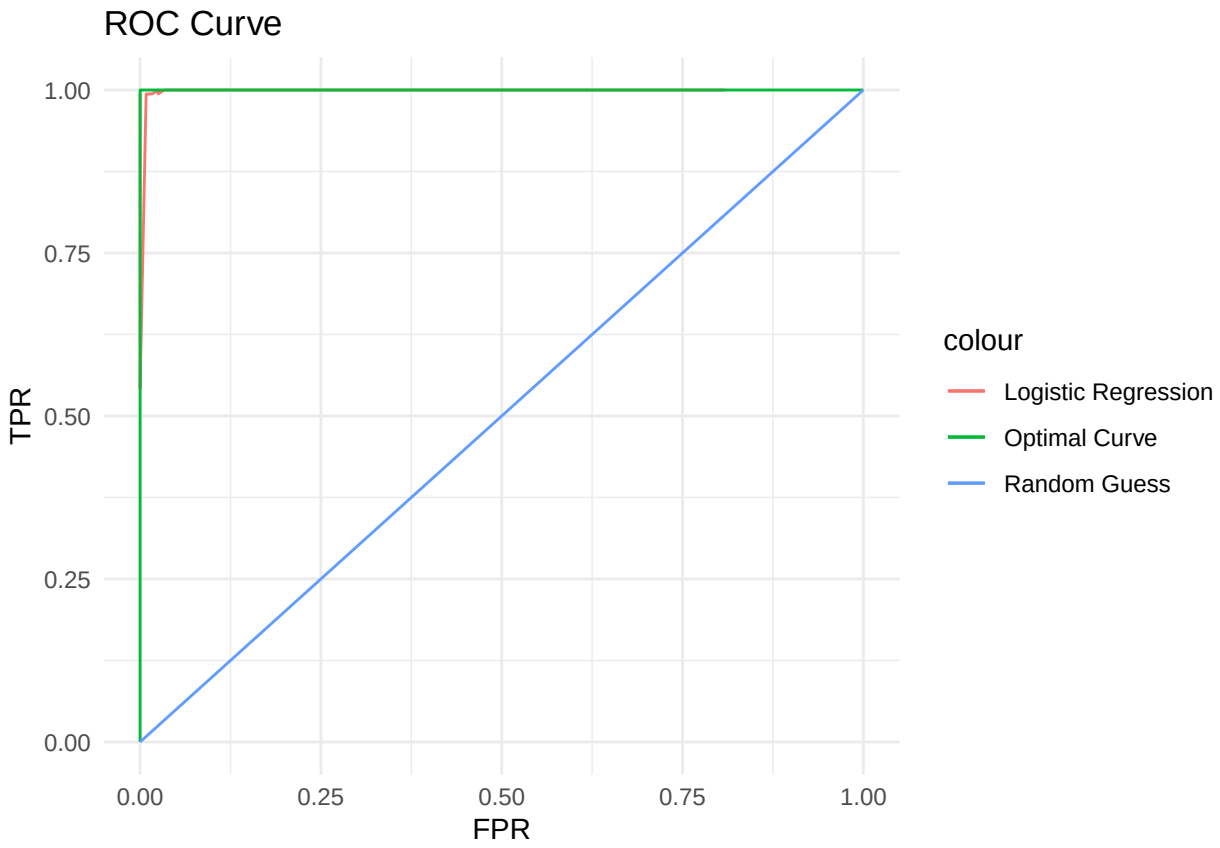
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

The optimal penalty factor according to the model is $\lambda = 0.04481765$.



From the plot one can see that the model seems to start getting bad results very quickly for increasing values of lambda. Therefore, it can be concluded that the unpenalized model has a good trade-off between bias and variance, because introducing a high penalty factor lambda would decrease the variance and increase the bias and the error would therefore be less for these high values of lambda, if the variance of the unpenalized model was too high (which is called overfit).



From the plot one can see that the Logistic Regression performs very well on the test data. The area under the curve is almost 1 which is the optimal value a model could theoretically get. A model that would only output random guesses of the two classes for the prediction would have a value for the area under the curve of 0.5, so the proposed model performs way better.
In this case, the ROC curve with true positive rate (TPR) and false positive rate (FPR) was used, because the two target classes are appear about equally in the data (otherwise another metric than FPR and TPR would be more insightful).

Appendix

```
data_original <- read.csv("olive.csv")
pca_data <- data_original[,3:10]
# Scaling the data for PCA
pca_data_scaled <- scale(pca_data)
res <- princomp(pca_data_scaled)
eigenvals <- res$sdev^2
percentage_eigenvals <- eigenvals / sum(eigenvals)
cumsum_perc_eigenvals <- cumsum(percentage_eigenvals)

print(paste0("The mean of palmitic is ", mean(data_original$palmitic)))
print(paste0("The mean of linolenic is ", mean(data_original$linolenic)))
print(paste0("The variance of palmitic is ", var(data_original$palmitic)))
print(paste0("The variance of linolenic is ", var(data_original$linolenic)))

data <- as.data.frame(res$scores[,1:3])
data[,4] <- as.factor(data_original$Region)
colnames(data)[4] <- "Region"
```

```
n = dim(data)[1]
set.seed(12345)
id = sample(1:n, floor(n * 0.5))
train = data[id, ]
test = data[-id, ]

library(nnet)
model <- multinom(Region~., train)

train_predictions <- predict(model, train)
test_predictions <- predict(model, test)

calc_misclass <- function(pred, actual){
  res <- 0
  for (i in 1:length(pred)) {
    if(pred[i] != actual[i]) res <- res + 1
  }
  return(res / length(pred))
}

train_misclass <- calc_misclass(train_predictions, train$Region)
test_misclass <- calc_misclass(test_predictions, test$Region)

data <- data_original[,c(1,3:10)]
data$Region <- sapply(data$Region, function(x){if(x==1)"South" else "Other"})
data$Region <- as.factor(data$Region)
n = dim(data)[1]
set.seed(12345)
id = sample(1:n, floor(n * 0.5))
train = data[id, ]
test = data[-id, ]
library(glmnet)
ridge_model <- cv.glmnet(as.matrix(subset(train, select=-Region)), train$Region,
                         alpha=0, family = "binomial")
#ridge_model$lambda.min

plot(ridge_model)

TPR <- c()
FPR <- c()
test_predict <- predict(ridge_model, as.matrix(subset(test, select=-Region)),
                        type = "response", s="lambda.min")
for (i in seq(0.05, 0.95, 0.01)) {
  test_predict_temp <- sapply(test_predict,
                              function(x,p){if (x>p) "South" else "Other"},
                              p = i)
  test_predict_temp <- as.factor(test_predict_temp)
  confMat <- table(test_predict_temp, test$Region)
  TPR <- append(TPR, confMat[2, 2]/(confMat[1,2] + confMat[2, 2]))
  FPR <- append(FPR, confMat[2, 1]/(confMat[1,1] + confMat[2, 1]))
}

# Plot
```

```r
plot_data <- data.frame(TPR = TPR, FPR = FPR)
library(ggplot2)
ggplot() +
  geom_line(data=plot_data, mapping=aes(FPR,
                                        TPR,
                                        color = "Logistic Regression")) +
  geom_line(data=data.frame(FPR=c(0,0,1),
                            TPR=c(0,1,1)),
            mapping=aes(FPR, TPR, color = "Optimal Curve")) +
  geom_line(data=data.frame(FPR=c(0,1), TPR=c(0,1)),
            mapping=aes(FPR, TPR, color = "Random Guess")) +
  theme_minimal() + labs(title = "ROC Curve")
```