Machine Learning, Spring 2021
Homework 2

Daniil Bulat, Jonas Josef Huwyler, Haochen Li,Giovanni Magagnin

March 11, 2021

Exercise 1

Consider last week's R-code for the homework (on the iris data, splitting into training and test data).

```
# Set up
rm(list = ls())
set.seed(123)

# Shuffle the Iris data
Data = iris[sample(1:150),]

# and split into training and test data (80-20)
Data.Train = Data[1:120,]
Data.Test = Data[121:150,]
```

Question 1

Do a simple linear regression on the training data with feature $x_1 = Petal.Length$ and output $y = Sepal.Length$.Use the *summary* command to determine the RSE_{train} and the R^2_{train}.Based on this, would you say that the model works well? With the significance guidelines from the lecture slides, how significant is the impact of x_1 on y.

Codes and Results are listed below. The corresponding RSE_{train} is 0.4148 and the R^2_{train} is 0.7627. The R^2_{train} is relatively high considering we just use one explanatory variable. The model works well, which explains 76.27% of the variability in the response, but it still has room to improve the quality of fit.

The influence of Petal.Length on Sepal.Length is highly significant since the p-value of the variable is smaller than 0.01.

```
# Simple linear regression on the training data: x1=Petal.Length and y = Sepal.Length
Iris.Model1 = lm(Sepal.Length ~ Petal.Length, data = Data.Train)

# Determine RSE and R2
summary(Iris.Model1)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length, data = Data.Train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.23656 -0.30673 -0.03334  0.26958  1.02598
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.27856    0.08921   47.96   <2e-16 ***
## Petal.Length  0.41289    0.02120   19.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4148 on 118 degrees of freedom
## Multiple R-squared:  0.7627, Adjusted R-squared:  0.7607
## F-statistic: 379.2 on 1 and 118 DF,  p-value: < 2.2e-16
```

Question 2

Do the same as above for the model with two features (petal length and petal width) and three features (petal length, petal width and sepal width).

Codes and Results are listed below.

For two-feature model, the corresponding RSE_{train} is 0.4108 and the R^2_{train} is 0.7692. Compared to the previous model, there is not much improvement.

The influence of Petal.Length on Sepal.Length is highly significant since the p-value of the variable is smaller than 0.01. On the other hand, the influence of Petal.Width is of marginal significance since the p-value of the variable is between 0.05 and 0.1.

For three-feature model, the corresponding RSE_{train} is 0.3258 and the R^2_{train} is 0.856. The model works better than the previous ones since now it can explains more variability in the response.

All three variables show high significance levels since the p-values of the variables are smaller than 0.01.

```
# OLS on the training data: two features(petal length and petal width)
Iris.Model2 = lm(Sepal.Length ~ Petal.Length + Petal.Width, data = Data.Train)
summary(Iris.Model2)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length + Petal.Width, data = Data.Train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.17814 -0.32412 -0.04138  0.28352  1.04296
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.16844    0.10726  38.863  < 2e-16 ***
## Petal.Length  0.54270    0.07468   7.267 4.46e-11 ***
## Petal.Width  -0.31322    0.17293  -1.811   0.0727 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4108 on 117 degrees of freedom
## Multiple R-squared:  0.7692, Adjusted R-squared:  0.7652
## F-statistic: 194.9 on 2 and 117 DF,  p-value: < 2.2e-16
```

```r
# OLS on the test data: three features(petal length, petal width and sepal width)
Iris.Model3 = lm(Sepal.Length ~ Petal.Length + Petal.Width + Sepal.Width, data = Data.Train)
summary(Iris.Model3)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length + Petal.Width + Sepal.Width,
##     data = Data.Train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82577 -0.21712  0.02843  0.18999  0.85864
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.89882    0.28429   6.679 8.74e-10 ***
## Petal.Length  0.69826    0.06208  11.247  < 2e-16 ***
## Petal.Width  -0.53303    0.13964  -3.817 0.000218 ***
## Sepal.Width   0.63637    0.07606   8.367 1.52e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3258 on 116 degrees of freedom
## Multiple R-squared:  0.856,  Adjusted R-squared:  0.8523
## F-statistic: 229.9 on 3 and 116 DF,  p-value: < 2.2e-16
```

Question 3

Now use the model with one feature and predict with it the sepal lengths for the test data. Compute the corresponding RSE_{test} and the R^2_{test}. How would you now rate the model's performance?

Codes and results are shown below. The corresponding RSE_{test} is 0.390 and R^2_{test} is 0.743.The model performs very well on the test data. The RSE is even smaller on the test than on the train data. This is also due to a lower TSS on the test data. Th R^2 is a little bit lower than in the train data but still high.

```r
# Predict with the simple linear model on test data
Yhat=cbind(1,Data.Test$Petal.Length)%*%Iris.Model1$coefficients

# Compute RSE_test and R2_test
error.test =  Data.Test$Sepal.Length - Yhat
RSS.test = sum(error.test^2)
RSE.test = sqrt(RSS.test/(length(Yhat)-2))
TSS.test = sum((Data.Test$Sepal.Length - mean(Data.Test$Sepal.Length))^2)
R2.test = 1 - RSS.test/TSS.test

# Print the result
sprintf("The corresponding RSE is %.3f and R-squared is %.3f", RSE.test,R2.test)
```

```
## [1] "The corresponding RSE is 0.390 and R-squared is 0.743"
```

Exercise 2

In this exercise we do cross validation on a rolling basis for the Olympics data (which can be found in *olympics_data.R* on StudyNet):

```
# Set up
rm(list = ls())
set.seed(123)

# Import data
L.olympics=t(data.frame(c(1896,12),c(1900,11),c(1904,11),c(1906,11.2),c(1908,10.8),
                        c(1912,10.8),c(1920,10.8),c(1924,10.6),c(1928,10.8),c(1932,10.3),
                        c(1936,10.3),c(1948,10.3),c(1952,10.4),c(1956,10.5),c(1960,10.2),
                        c(1964,10),c(1968,9.95),c(1972,10.14),c(1980,10.25),c(1984,9.99),
                        c(1988,9.92),c(1992, 9.96),c(1996, 9.84),c(2000, 9.87),c(2004,9.85),
                        c(2008, 9.69),c(2012, 9.61),c(2016, 9.83)))

Data = data.frame(L.olympics)
colnames(Data) = c("year","time")
rownames(Data) = 1:nrow(Data)
```

Question 1

Use the first 7 data points for training and find the linear regression function $f_1(x)$. Use data points 8-14 as test data and compute the corresponding RSS with the regression function $f_1(x)$.

The corresponding RSS with the regression function $f_1(x)$ is 4.347

```
# Split data into Training set (1-7) and Test set (8-14)
Data.Train1 = Data[1:7,]
Data.Test1 = Data[8:14,]

# Simple linear regression on Training set
f1 = lm(time ~ year, data = Data.Train1)

# predict with f1 on the test data
Yhat1 = cbind(1,Data.Test1$year)%*%f1$coefficients

# compute the corresponding RSS
err1 = Data.Test1$time - Yhat1
RSS1 = sum(err1^2)

# Print the result
sprintf("The corresponding RSS is %.3f",RSS1)
```

```
## [1] "The corresponding RSS is 4.347"
```

Question 2

Use the first 14 data points for training and find the linear regression function $f_2(x)$. Use data points 15-21 as test data and compute the corresponding RSS with the regression function $f_2(x)$.

The corresponding RSS with the regression function $f_2(x)$ is 0.566.

```r
# Split data into Training set (1-14) and Test set (15-21)
Data.Train2 = Data[1:14,]
Data.Test2 = Data[15:21,]

# Simple linear regression on Training set
f2 = lm(time ~ year, data = Data.Train2)

# predict with f1 on the test data
Yhat2 = cbind(1,Data.Test2$year)%*%f2$coefficients

# compute the corresponding RSS
err2 = Data.Test2$time - Yhat2
RSS2 = sum(err2^2)

# Print the result
sprintf("The corresponding RSS is %.3f",RSS2)
```

```
## [1] "The corresponding RSS is 0.566"
```

Question 3

Use the first 21 data points for training and find the linear regression function $f_3(x)$. Use data points 22-28 as test data and compute the corresponding RSS with the regression function $f_3(x)$.

The corresponding RSS with the regression function $f_3(x)$ is 0.394.

```r
# Split data into Training set (1-21) and Test set (22-28)
Data.Train3 = Data[1:21,]
Data.Test3 = Data[22:28,]

# Simple linear regression on Training set
f3 = lm(time ~ year, data = Data.Train3)

# predict with f1 on the test data
Yhat3 = cbind(1,Data.Test3$year)%*%f3$coefficients

# compute the corresponding RSS
err3 = Data.Test3$time - Yhat3
RSS3 = sum(err3^2)

# Print the result
sprintf("The corresponding RSS is %.3f",RSS3)
```

```
## [1] "The corresponding RSS is 0.394"
```

Question 4

Compute the average RSS of the three rounds.

The average RSS of the three rounds is 1.769.

```r
# compute the average RSS of the three rounds
RSS = c(RSS1,RSS2,RSS3)
RSS.ave = mean(RSS)
```

```r
# Print the result
sprintf("The average RSS of the three rounds is %.3f",RSS.ave)
```

```
## [1] "The average RSS of the three rounds is 1.769"
```