Homework 2 - Machine Learning

Maximilian Arrich, Florian Benkhalifa, Raphael Eigenmann

March 6 2019

Exercise 1

Exercise 1.1

We split the data into testing and training data:

```
set.seed(100) #allows for reproducibility
data   <- Boston

#we declare 70% of the data to be training data
sample <- sample(nrow(data), floor(0.7*nrow(data)), replace = FALSE)
train  <- data[sample,][,13:14]
test   <- data[-sample,][,13:14]
```

Exercise 1.2

We run a linear regression to obtain the estimate coefficients:

```
fit_train <- lm(medv~lstat, data=train)
pander(fit_train)
```

Table 1: Fitting linear model: medv ~ lstat

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	35.01	0.6784	51.6	3.176e-166
lstat	-0.9752	0.0471	-20.7	7.539e-63

Hence, the prediction function is given by the estimated coefficients::

$$medv = 35.0073 - 0.9752\, lstat$$

Exercise 1.3

We insert into the above stated prediction function the lstat data to obtain the fitted values for the testing data. Alternatively, we can use the R function `predict`. Both methods give the same fitted values:

```
pred_test<- fit_train$coefficients[1] + fit_train$coefficients[2]*test$lstat

pred_test_alternative <- predict(fit_train, newdata = test)

all(pred_test == pred_test_alternative)
```

```
## [1] TRUE
```

Exercise 1.4

We compute the residuals by calculating the difference of the empirical values and the predictions:

```
residuals <- test$medv-pred_test
```

Exercise 1.5

For the sake of handiness concerning the following Exercises we decided to condense the previous commands to one single function. This allows us to compute RSS, RSE and R-squared, while diffentiating between linear and quadratic regression models by choosing the poly variable accordingly. Further, specifying the seed variable allows to collect different random samples in Exercise 2.2 and 2.3. while attaining reproducibility. We did not fully understand if the notion of "different random splittings into training and test data" in question 2.2 was related to randomly varying the test and training sizes, or to keep the splitting ratio (70%-30%) and to simply take 5 different random samples. Because of that, we included the possibility to also vary the partition sizes fixed splitting by setting the training size randomly between 0.5 and 1. The function looks as follows:

```
fun <- function(poly, seed, varying_partition = F){
  set.seed(seed)    # Setting seed to create multiple reproducible samples
   # Create random partition size.
  if(varying_partition == T) training_size <- 0.5 + runif(1,0,0.5) else training_size <- 0.7
  # Creating sample
  sample    <- sample(nrow(data), floor(training_size*nrow(data)), replace = FALSE)
  train     <- data[sample,][,13:14]
  test      <- data[-sample,][,13:14]
  # Regression formula
  formula   <- paste("medv~poly(lstat,", as.character(poly), ",raw=T)")
  lm        <- lm(as.formula(formula), data=train)
  predict   <- predict(lm, newdata = test)
  residuals <- test$medv - predict
  RSS       <- sum(residuals^2)
  RSE       <- sqrt(RSS/(nrow(data)-2))
  R_squared <- 1 - RSS/(var(test$medv)*(nrow(data)-1))
  aic       <- AIC(lm)
  # organizing output in matrix
  mat       <- matrix(c(RSS, RSE, R_squared,aic), nrow=4,
                      dimnames = list(c("RSS", "RSE", "R_squared","AIC")))
  return(mat)
}
```

We use the above stated function to obtain the R^2 of the linear model:

```
pander(fun(poly=1, seed=100))
```

RSS	5720
RSE	3.369
R_squared	0.8586
AIC	2307

Exercise 2

Exercise 2.1

Again, and since we set the seed equal to 100 at the beginning, we can use our function to compute the statistics for the quadratic model by specifying poly = 2 and by specifying the correct seed:

```
pander(fun(poly=2, seed= 100))
```

RSS	4513
RSE	2.992
R_squared	0.8885
AIC	2225

Exercise 2.2

Here, we interpreted "different random splittings into training and test data" as keeping the partition size fixed while simply taking 5 different random samples. Below, we also added an alternative solution where we allowed the partition size to be randomly selected. We compute the desired statistic for five different random data splittings:

```
# Linear Model
R2_linear <- do.call(cbind,lapply(1:5, fun, poly = 1)) #values for linear model
pander(R2_linear %>% set_colnames(1:5))
```

	1	2	3	4	5
RSS	6194	5622	5418	6273	6986
RSE	3.506	3.34	3.279	3.528	3.723
R_squared	0.8564	0.8688	0.8667	0.8569	0.8569
AIC	2294	2309	2314	2292	2273

```
# Quadratic Model
R2_quadratic <- do.call(cbind,lapply(1:5, fun, poly = 2)) #values for quadratic model
pander(R2_quadratic %>% set_colnames(1:5))
```

	1	2	3	4	5
RSS	4449	3742	4629	4699	5608
RSE	2.971	2.725	3.031	3.053	3.336
R_squared	0.8969	0.9127	0.8861	0.8928	0.8851
AIC	2228	2249	2221	2218	2188

Exercise 2.3

Compute the mean R^2 for of the two different model specifications:

```
mean(R2_linear[3,])
```

```
## [1] 0.8611437
```

```
mean(R2_quadratic[3,])
```

```
## [1] 0.8947199
```

Since, on average, the quadratic model explains more of the variance on the testing data (higher R^2), it seems eligible to choose the quadratic model. This choice is supportet by the AIC of the quadratic model being lower than the one of the linear model, implying a lower Kullback-Leiber divergence:

```
mean(R2_linear[4,])
```

```
## [1] 2296.432
```

```
mean(R2_quadratic[4,])
```

```
## [1] 2220.84
```

Alternative to Exercise 2.2 with random splitting ratios

Now, we present an alternative way of understanding "different random splittings" by also varying the size of the partitions. However, even by using this alternative interpretation, the quadratic model prevails as we will see below.

```
# Linear Model
R2_linear_alt <- do.call(cbind,lapply(1:5, fun, poly = 1, varying_partition = 1))
pander(R2_linear_alt %>% set_colnames(1:5))
```

	1	2	3	4	5
RSS	7555	8448	10422	3927	8252
RSE	3.872	4.094	4.547	2.791	4.046
R__squared	0.825	0.7994	0.7992	0.9018	0.8179
AIC	2072	1934	1861	2613	1961

```
# Quadratic Model
R2_quadratic_alt <- do.call(cbind,lapply(1:5, fun, poly = 2, varying_partition = 1))
pander(R2_quadratic_alt %>% set_colnames(1:5))
```

	1	2	3	4	5
RSS	6100	6141	8300	2809	5913
RSE	3.479	3.491	4.058	2.361	3.425
R__squared	0.8587	0.8542	0.8401	0.9297	0.8695
AIC	1994	1883	1793	2527	1910

Exercise 2.3

Compute the mean R^2 for of the two different model specifications:

```
mean(R2_linear_alt[3,])
```

```
## [1] 0.8286668
```

```
mean(R2_quadratic_alt[3,])
```

```
## [1] 0.8704592
```

Since, on average, the quadratic model explains more of the variance on the testing data (higher R^2), it seems eligible to choose the quadratic model. This choice is supportet by the AIC of the quadratic model being lower than the one of the linear model, implying a lower Kullback-Leiber divergence:

```
mean(R2_linear_alt[4,])
```

```
## [1] 2088.197
```

```
mean(R2_quadratic_alt[4,])
```

```
## [1] 2021.294
```