Sample Answer Lab01b: k-Nearest Neighbor & Principal Components Analyses

Task 1. kNN Analysis [5 points]

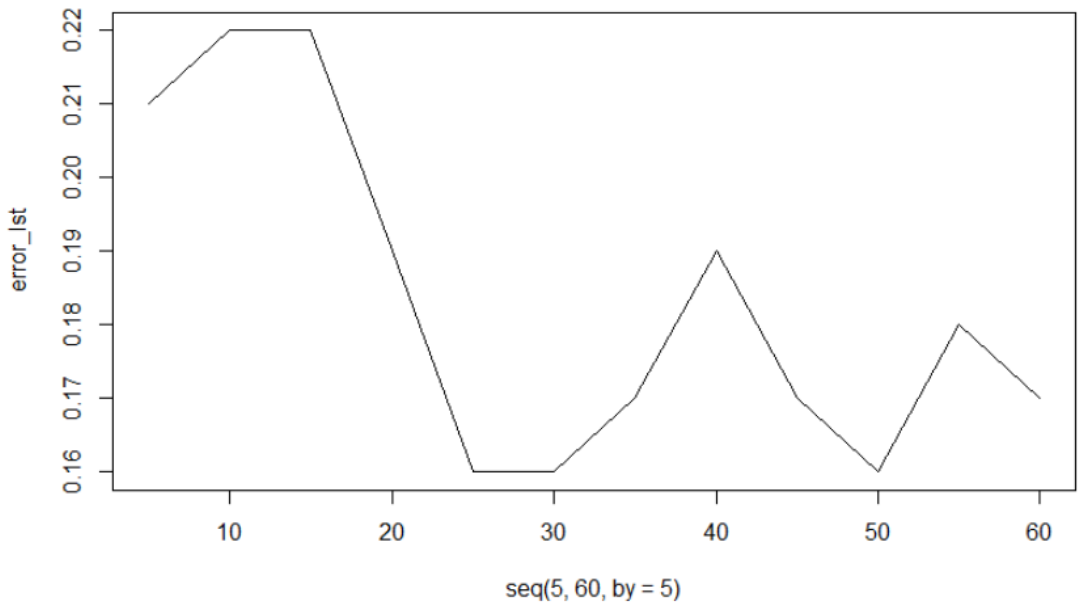
Task1.1: Normalize the data-frame **`xVars`** before splitting it up into a training and a test data set. Why do the data need to be normalized? [1 point]

```
xVars <- as.data.frame(lapply(xVars, normalize))
```

Notes: The variance of the features differs substantially. Relative to features with a low variance the inter object distances along features with a higher variance is substantially higher. Since the algorithm aims at minimizing the internal class heterogeneity features with a high variance will dominate the resulting class assignment.

Task 1.2: Vary the hyper-parameter k in the range **`k <- seq(5, 60, by=5)`** and record the ***total error rate*** in the ***test data set*** for each k-value. Plot the total error rate against the k value to identify the optimal k-value. [2 points]

```
error_lst <- c ()
for (i in seq (5,60, by = 5)) {
  k_accuracy <- caret:confusionMatrix( yVarTest,knn(train = xVarsTrain, test
= xVarsTest,cl = yVarTrain, k = i))$overall["Accuracy"]
  error_rate <- 1 - k_accuracy
  error_lst <- c(error_lst,error_rate)
  }
plot (seq (5,60,by = 5),error_lst,"l")
```



So， the lowest error rate is for $k \in \{25 \ldots 30 \text{ and } 50\}$. 30 neighbors have been subsequently selected as optimal result.

Task 1.3: Discuss the optimal solution in the light that the test data set has 28 census tracts with newer built home and 72 census tracts with older built homes. [1 point]

```
testpred <- knn(train = xVarsTrain, test = xVarsTest,cl = yVarTrain, k = 30)
CrossTable(x = yVarTest,y = testpred)

Total Observations in Table:  100
             | testpred
    yVarTest |       new |       old | Row Total |
-------------|-----------|-----------|-----------|
         new |        15 |        13 |        28 |
             |    24.703 |     4.705 |           |
             |     0.536 |     0.464 |     0.280 |
             |     0.938 |     0.155 |           |
             |     0.150 |     0.130 |           |
-------------|-----------|-----------|-----------|
         old |         1 |        71 |        72 |
             |     9.607 |     1.830 |           |
             |     0.014 |     0.986 |     0.720 |
             |     0.062 |     0.845 |           |
             |     0.010 |     0.710 |           |
-------------|-----------|-----------|-----------|
Column Total |        16 |        84 |       100 |
             |     0.160 |     0.840 |           |
-------------|-----------|-----------|-----------|
```

Notes: For neighborhoods with predominately newer homes the algorithm performs virtually not better than flipping a coin (50-50 chance). In contrast, neighborhoods with predominately older homes are almost perfectly identified. Since the test sample has only 28 neighborhoods with predominately newer homes the mis-classification of this category does not substantially impact the total error rate.

Task 1.4: Compare the predicted probability of being a census tract with predominately newer homes against proportions of newer build homes in the variable **`validTractShp$post1980[testSel]`** in a scatter plot and interpret the plot. Hint: you get the probabilities with the syntax **`attr(testPred, "prob")`** and you need to carefully adjust this probability because it measure the probability of the highest predicted class. [1 point]

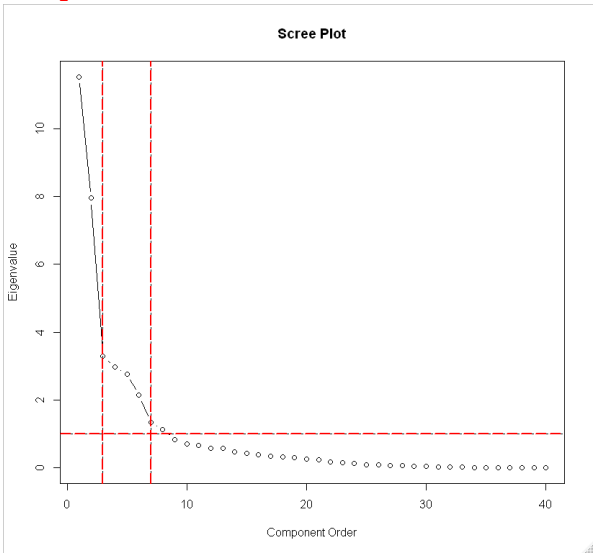
```
predProbs <- ifelse(testPred == "old", 1-attr(testPred, "prob"),
                                       attr(testPred, "prob")  )
plot(validTractShp$post1980[testSel], predProbs,
     xlab="Proportion of newer homes",
     ylab="Predicted Probablity of new homes",
     main="Newer Versus Older Homes by Test Sample Census Tract")
abline(lm(predProbs~validTractShp$post1980[testSel]),
       lty=5, lwd=3, col="red")
abline(v=50, h=0.5, lty=3, lwd=2, col="red")
```

Notes: There is a clear relationship between the predicted probability of newer homes and the observed proportion of newer homes in the test sample census tracts. It is again also apparent that the predict in census tracts with newer homes does not perform well.

Task 2: Principal Component Analysis [2 points]

Task 2.1: Perform a principal component analysis on the data-frame **`xVars`** and provide the scree-plot and proportion of explained variance plot. [1 point]

```
pc <-prcomp(xVars, retx=TRUE, scale.=TRUE)
plot(pc$sdev^2, type="b", main="Scree Plot",
     xlab="Component Order", ylab="Eigenvalue")
abline(h=1, v=c(3,7), lty=5,lwd=2, col="red")
```

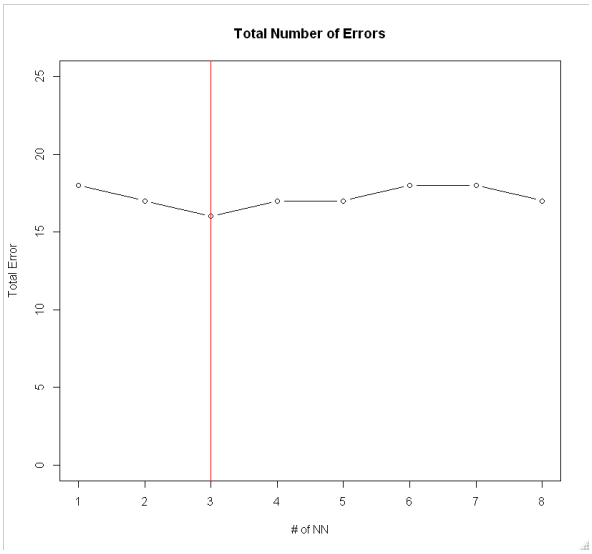


Notes: In total 8 components have eigenvalues greater than the critical value of one. However, using the elbow rule the slope changes noticeable either at 3 or 7 components.

Task 2.2: Perform a sequence of kNN analyses with the hyper parameter $k = 30$ against an increasing number of component score vectors starting from the one with the largest eigenvalue to the one with its eigenvalue barely larger than one. Plot the total test error rate against the number of component score vectors. [1 point]

```
newVars <- pc$x[, 1:8]
newVars <- as.data.frame(apply(newVars, 2, normalize))

xVarsTest <- newVars[testSel, ]
xVarsTrain <- newVars[-testSel, ]
result <- NULL
for (i in 1:8){
  testPred <- knn(train = xVarsTrain, test = xVarsTest, prob = TRUE,
                  cl = yVarTrain, k = 30)
  totalErr <- sum(yVarTest != testPred)
  result <- rbind(result, c(i, totalErr))
}
## Identify k-value with lowest total error
(kValue <- which.min(result[, 2]))
plot(result[,c(1,2)], type="b", xlab="# of NN", ylab="Total Error",
     main="Total Number of Errors", ylim=c(0,25))
abline(v=kValue,col="red")
```



Notes: Only the first three component score vectors are sufficient to best perform the class membership predication. However, the number of misclassified test census tracts is now 16, whereas, in task 1.3 only 14 tracts were misclassified.