Lazy Learning - Classification Using Nearst Neighbors

Step 1 - collecting data

Step 2 - exploring and preparing the data

Import local CSV file into R

```
wbcd <- read.csv("wisc_bc_data.csv", stringsAsFactors = FALSE)
```

Drop the ID column - prevent overfitting

```
wbcd <- wbcd[-1]
```

table of interest

```
table(wbcd$diagnosis)
```

```
##
##   B   M
## 357 212
```

recode diagnosis as a factor and table or proportions with more informative labels

```
wbcd$diagnosis <- factor(wbcd$diagnosis, levels = c("B", "M"),
                         labels = c("Benign", "Malignant"))

round(prop.table(table(wbcd$diagnosis)) * 100, digits = 1)
```

```
##
##    Benign Malignant
##      62.7      37.3
```

Overview of three features

```
summary(wbcd[c("radius_mean", "area_mean", "smoothness_mean")])
```

```
##   radius_mean       area_mean      smoothness_mean
##  Min.   : 6.981   Min.   : 143.5   Min.   :0.05263
##  1st Qu.:11.700   1st Qu.: 420.3   1st Qu.:0.08637
##  Median :13.370   Median : 551.1   Median :0.09587
##  Mean   :14.127   Mean   : 654.9   Mean   :0.09636
##  3rd Qu.:15.780   3rd Qu.: 782.7   3rd Qu.:0.10530
##  Max.   :28.110   Max.   :2501.0   Max.   :0.16340
```

Create a normalize() function

```
normalize <- function(x){return ((x-min(x)) / (max(x)-min(x)))}
```

test the function on a couple of vectors:

```
normalize(c(1,2,3,4,5))
```

```
## [1] 0.00 0.25 0.50 0.75 1.00
```

```
normalize(c(10,20,30,40,50))
```

```
## [1] 0.00 0.25 0.50 0.75 1.00
```

normalize the remaining 30 numeric features

```
wbcd_n <- as.data.frame(lapply(wbcd[2:31], normalize))
```

```
summary(wbcd_n$area_mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1174  0.1729  0.2169  0.2711  1.0000
```

Data preparation - creating training and test datasets

```
wbcd_train <- wbcd_n[1:469,]
wbcd_test <- wbcd_n[470:569,]
```

```
wbcd_train_labels <- wbcd[1:469, 1]
wbcd_test_labels <- wbcd[470:569, 1]
```

Step 3 - training a model on the data

```
library(class)
```

```
## Warning: package 'class' was built under R version 3.4.4
```

```
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 3.4.4
```

KNN() function - K Nearest Neighbor

```
wbcd_test_pred <- knn(train = wbcd_train, test=wbcd_test, cl=wbcd_train_labels, k=21)
```

Step 4 - evaluating model performance

```
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        61 |         0 |        61 |
##                  |     1.000 |     0.000 |     0.610 |
```

```
##                  |     0.968 |     0.000 |           |
##                  |     0.610 |     0.000 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         2 |        37 |        39 |
##                  |     0.051 |     0.949 |     0.390 |
##                  |     0.032 |     1.000 |           |
##                  |     0.020 |     0.370 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        63 |        37 |       100 |
##                  |     0.630 |     0.370 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

Step 5 - improving model performance

Transformation - Z-score standardization

```
wbcd_z <- as.data.frame(scale(wbcd[-1]))
```

```
wbcd_train <- wbcd_z[1:469, ]
wbcd_test <- wbcd_z[470:569, ]
wbcd_train_labels <- wbcd[1:469, 1]
wbcd_test_labels <- wbcd[470:569, 1]
```

Predict outcome by using Z-score transformation

Note: Z-score transformation failed to improve the model. Therefore, the min-max normalization method with k =21 is used for knn prediction.

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test,
cl = wbcd_train_labels, k = 21)

CrossTable(x = wbcd_test_labels, y = wbcd_test_pred,
prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        61 |         0 |        61 |
##                  |     1.000 |     0.000 |     0.610 |
```

```
##                  |     0.924 |     0.000 |           |
##                  |     0.610 |     0.000 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         5 |        34 |        39 |
##                  |     0.128 |     0.872 |     0.390 |
##                  |     0.076 |     1.000 |           |
##                  |     0.050 |     0.340 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        66 |        34 |       100 |
##                  |     0.660 |     0.340 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

start time

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test, cl = wbcd_train_labels, k=1)
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        59 |         2 |        61 |
##                  |     0.967 |     0.033 |     0.610 |
##                  |     0.952 |     0.053 |           |
##                  |     0.590 |     0.020 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         3 |        36 |        39 |
##                  |     0.077 |     0.923 |     0.390 |
##                  |     0.048 |     0.947 |           |
##                  |     0.030 |     0.360 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        62 |        38 |       100 |
##                  |     0.620 |     0.380 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test, cl = wbcd_train_labels, k=5)
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        60 |         1 |        61 |
##                  |     0.984 |     0.016 |     0.610 |
##                  |     0.968 |     0.026 |           |
##                  |     0.600 |     0.010 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         2 |        37 |        39 |
##                  |     0.051 |     0.949 |     0.390 |
##                  |     0.032 |     0.974 |           |
##                  |     0.020 |     0.370 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        62 |        38 |       100 |
##                  |     0.620 |     0.380 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test, cl = wbcd_train_labels, k=11)
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
```

```
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        60 |         1 |        61 |
##                  |     0.984 |     0.016 |     0.610 |
##                  |     0.952 |     0.027 |           |
##                  |     0.600 |     0.010 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         3 |        36 |        39 |
##                  |     0.077 |     0.923 |     0.390 |
##                  |     0.048 |     0.973 |           |
##                  |     0.030 |     0.360 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        63 |        37 |       100 |
##                  |     0.630 |     0.370 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test, cl = wbcd_train_labels, k=15)
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        61 |         0 |        61 |
##                  |     1.000 |     0.000 |     0.610 |
##                  |     0.953 |     0.000 |           |
##                  |     0.610 |     0.000 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         3 |        36 |        39 |
##                  |     0.077 |     0.923 |     0.390 |
##                  |     0.047 |     1.000 |           |
##                  |     0.030 |     0.360 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        64 |        36 |       100 |
##                  |     0.640 |     0.360 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test, cl = wbcd_train_labels, k=21)
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        61 |         0 |        61 |
##                  |     1.000 |     0.000 |     0.610 |
##                  |     0.924 |     0.000 |           |
##                  |     0.610 |     0.000 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         5 |        34 |        39 |
##                  |     0.128 |     0.872 |     0.390 |
##                  |     0.076 |     1.000 |           |
##                  |     0.050 |     0.340 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        66 |        34 |       100 |
##                  |     0.660 |     0.340 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test, cl = wbcd_train_labels, k=27)
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | wbcd_test_pred
```

```
## wbcd_test_labels |    Benign | Malignant | Row Total |
## -----------------|-----------|-----------|-----------|
##           Benign |        61 |         0 |        61 |
##                  |     1.000 |     0.000 |     0.610 |
##                  |     0.924 |     0.000 |           |
##                  |     0.610 |     0.000 |           |
## -----------------|-----------|-----------|-----------|
##        Malignant |         5 |        34 |        39 |
##                  |     0.128 |     0.872 |     0.390 |
##                  |     0.076 |     1.000 |           |
##                  |     0.050 |     0.340 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        66 |        34 |       100 |
##                  |     0.660 |     0.340 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

Accuracy of the prediction - Min_Max Normalization Method K=21

The accuracy of the predication for min-max normalization and k=21 method produce a 98% accuracy.

```
(61+37) /100
```

```
## [1] 0.98
```

Adult Example From UCI Repository

Step 2

```
library(RCurl)
```

```
## Warning: package 'RCurl' was built under R version 3.4.3
```

```
## Loading required package: bitops
```

```
urlfile <-'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
downloaded <- getURL(urlfile, ssl.verifypeer=FALSE)
connection <- textConnection(downloaded)
df <- read.csv(connection, header=FALSE, stringsAsFactors = FALSE)

str(df)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ V1: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ V2: num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ V3: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ V4: num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ V5: chr  "Iris-setosa" "Iris-setosa" "Iris-setosa" "Iris-setosa" ...
```

```
table(df$V5)
```

```
##
##     Iris-setosa Iris-versicolor  Iris-virginica
##              50              50              50
```

```
summary(df)
```

```
##        V1              V2              V3              V4
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##       V5
##  Length:150
##  Class :character
##  Mode  :character
##
##
##
```

```
df <- df[sample(1:nrow(df)), ]

df$V5 <- factor(df$V5, levels = c("Iris-setosa", "Iris-virginica", "Iris-versicolor"))
round(prop.table(table(df$V5)) * 100, digits = 1)
```

```
##
##     Iris-setosa  Iris-virginica Iris-versicolor
##            33.3            33.3            33.3
```

```
str(df$V5)
```

```
##  Factor w/ 3 levels "Iris-setosa",..: 1 1 1 2 3 3 2 2 3 1 ...
```

```
levels(df$V5)
```

```
## [1] "Iris-setosa"     "Iris-virginica"  "Iris-versicolor"
```

```
normalize <- function(x){return ((x-min(x)) / (max(x)-min(x)))}
df_n <- as.data.frame(lapply(df[1:4], normalize))
summary(df_n)
```

```
##        V1               V2               V3               V4
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333
##  Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000
##  Mean   :0.4287   Mean   :0.4392   Mean   :0.4676   Mean   :0.45778
##  3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
```

```
df_train <- df_n[1:100,]
df_test <- df_n[101:150,]
df_train_labels <- df[1:100, 5]
df_test_labels <- df[101:150, 5]
```

Step 3 - training a model on the data

```
library(class)
library(gmodels)
```

KNN() function - K Nearest Neighbor

```
df_test_pred <- knn(train = df_train, test=df_test, cl=df_train_labels, k=11)
```

Step 4 - evaluating model performance

```
CrossTable(x = df_test_labels, y = df_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  50
##
##
##                  | df_test_pred
##  df_test_labels |     Iris-setosa |  Iris-virginica | Iris-versicolor |       Row Total |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##     Iris-setosa |              18 |               0 |               0 |              18 |
##                 |           1.000 |           0.000 |           0.000 |           0.360 |
##                 |           1.000 |           0.000 |           0.000 |                 |
##                 |           0.360 |           0.000 |           0.000 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##  Iris-virginica |               0 |              12 |               1 |              13 |
##                 |           0.000 |           0.923 |           0.077 |           0.260 |
##                 |           0.000 |           0.923 |           0.053 |                 |
##                 |           0.000 |           0.240 |           0.020 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
## Iris-versicolor |               0 |               1 |              18 |              19 |
##                 |           0.000 |           0.053 |           0.947 |           0.380 |
##                 |           0.000 |           0.077 |           0.947 |                 |
##                 |           0.000 |           0.020 |           0.360 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##    Column Total |              18 |              13 |              19 |              50 |
##                 |           0.360 |           0.260 |           0.380 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##
##
```

Step 5 - improving model performance

Adjust K value

```
df_test_pred <- knn(train = df_train, test=df_test, cl=df_train_labels, k=9)
CrossTable(x = df_test_labels, y = df_test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  50
##
##
##                  | df_test_pred
##  df_test_labels |     Iris-setosa |  Iris-virginica | Iris-versicolor |       Row Total |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##     Iris-setosa |              18 |               0 |               0 |              18 |
##                 |           1.000 |           0.000 |           0.000 |           0.360 |
##                 |           1.000 |           0.000 |           0.000 |                 |
##                 |           0.360 |           0.000 |           0.000 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##  Iris-virginica |               0 |              12 |               1 |              13 |
##                 |           0.000 |           0.923 |           0.077 |           0.260 |
##                 |           0.000 |           0.923 |           0.053 |                 |
##                 |           0.000 |           0.240 |           0.020 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
## Iris-versicolor |               0 |               1 |              18 |              19 |
##                 |           0.000 |           0.053 |           0.947 |           0.380 |
##                 |           0.000 |           0.077 |           0.947 |                 |
##                 |           0.000 |           0.020 |           0.360 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##    Column Total |              18 |              13 |              19 |              50 |
##                 |           0.360 |           0.260 |           0.380 |                 |
## ----------------|-----------------|-----------------|-----------------|-----------------|
##
##
```

In conclusion, the min-max normalization with k=9 is the optimal method for the prediction which produce a 100% accuracy based on a 50 test sample.