Practice3_DA5030

Shivam Verma

Implementing KNN using Class package.

Step- 2 Preparing and exploring the data

```
c <- read.csv("cancer.csv", stringsAsFactors = FALSE)
# Checking if data is structured
str(c)
```

```
## 'data.frame':    100 obs. of  10 variables:
##  $ id                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ diagnosis_result : chr  "M" "B" "M" "M" ...
##  $ radius            : int  23 9 21 14 9 25 16 15 19 25 ...
##  $ texture           : int  12 13 27 16 19 25 26 18 24 11 ...
##  $ perimeter         : int  151 133 130 78 135 83 120 90 88 84 ...
##  $ area              : int  954 1326 1203 386 1297 477 1040 578 520 476 ...
##  $ smoothness        : num  0.143 0.143 0.125 0.07 0.141 0.128 0.095 0.119 0.127 0.119 ...
##  $ compactness       : num  0.278 0.079 0.16 0.284 0.133 0.17 0.109 0.165 0.193 0.24 ...
##  $ symmetry          : num  0.242 0.181 0.207 0.26 0.181 0.209 0.179 0.22 0.235 0.203 ...
##  $ fractal_dimension: num  0.079 0.057 0.06 0.097 0.059 0.076 0.057 0.075 0.074 0.082 ...
```

```
# Removing ID form data set as it don't provide useful information
c <- c[-1]
head(c)
```

```
##   diagnosis_result radius texture perimeter area smoothness compactness
## 1                M     23      12       151  954      0.143       0.278
## 2                B      9      13       133 1326      0.143       0.079
## 3                M     21      27       130 1203      0.125       0.160
## 4                M     14      16        78  386      0.070       0.284
## 5                M      9      19       135 1297      0.141       0.133
## 6                B     25      25        83  477      0.128       0.170
##   symmetry fractal_dimension
## 1    0.242             0.079
## 2    0.181             0.057
## 3    0.207             0.060
## 4    0.260             0.097
## 5    0.181             0.059
## 6    0.209             0.076
```

```
# Getting count distribution of pateints
table(c$diagnosis_result)
```

```
##
```

```
##  B  M
## 38 62
```

```
# Renaming Variables
c$diagnosis <- factor(c$diagnosis_result, levels = c("B", "M"), labels = c("Benign", "Malignant"))
# Returning results as percentage
round(prop.table(table(c$diagnosis)) * 100, digits = 1)
```

```
##
##    Benign Malignant
##        38        62
```

Normalizing the data set.

```
set.seed(1234)
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x))) }
c_n <- as.data.frame(lapply(c[2:9], normalize))
summary(c_n$radius)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1875  0.5000  0.4906  0.7500  1.0000
```

Creating training and test data set.

```
c_train <- c_n[1:65,]
c_test <- c_n[66:100,]
# Now considering daignosis factor into consideration.
c_train_labels <- c[1:65, 10]
c_test_labels <- c[66:100, 10]
```

Step-3 Training a model on data

```
# Considering k as the square root of the number of observations.
c_test_pred <- knn(train = c_train, test = c_test, cl = c_train_labels, k=10)
```

Step 4 – Evaluating the model performance

```
CrossTable(x= c_test_labels, y= c_test_pred, prop.chisq = FALSE )
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
```

```
## Total Observations in Table:  35
##
##
##               | c_test_pred
## c_test_labels |    Benign | Malignant | Row Total |
## --------------|-----------|-----------|-----------|
##        Benign |         7 |        12 |        19 |
##               |     0.368 |     0.632 |     0.543 |
##               |     0.875 |     0.444 |           |
##               |     0.200 |     0.343 |           |
## --------------|-----------|-----------|-----------|
##     Malignant |         1 |        15 |        16 |
##               |     0.062 |     0.938 |     0.457 |
##               |     0.125 |     0.556 |           |
##               |     0.029 |     0.429 |           |
## --------------|-----------|-----------|-----------|
##  Column Total |         8 |        27 |        35 |
##               |     0.229 |     0.771 |           |
## --------------|-----------|-----------|-----------|
##
##
```

Out of 35 cases 7 are True Positive, 15 are True Negative (when positive class is Benign). Accuracy is determined by (TP+TN)/Total cases i.e. approx. 63%, thus there is room for improvement.

```
#Confusion Matrix for class:knn
confusionMatrix(c_test_labels, c_test_pred)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Benign Malignant
##   Benign         7        12
##   Malignant      1        15
##
##                Accuracy : 0.6286
##                  95% CI : (0.4492, 0.7853)
##     No Information Rate : 0.7714
##     P-Value [Acc > NIR] : 0.982633
##
##                   Kappa : 0.2902
##
##  Mcnemar's Test P-Value : 0.005546
##
##             Sensitivity : 0.8750
##             Specificity : 0.5556
##          Pos Pred Value : 0.3684
##          Neg Pred Value : 0.9375
##              Prevalence : 0.2286
##          Detection Rate : 0.2000
##    Detection Prevalence : 0.5429
##       Balanced Accuracy : 0.7153
##
```

```
##        'Positive' Class : Benign
##
```

Implementing KNN using Caret package

Using the same data splitted into 65:35 as train and test, as used before.

```
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(1234)
# Data Scaling Performed (Standardization using z-scores)
knn_fit <- train(c_train_labels ~., data = cbind(c_train_labels, c_train), method = "knn",
 trControl=trctrl,
 preProcess = c("center", "scale"),
 tuneLength = 10)
#printing the model
knn_fit
```

```
## k-Nearest Neighbors
##
## 65 samples
##  8 predictor
##  2 classes: 'Benign', 'Malignant'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 59, 60, 58, 58, 59, 58, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.8333333  0.5610983
##    7  0.8785714  0.6591657
##    9  0.8777778  0.6526838
##   11  0.8579365  0.5924597
##   13  0.8730159  0.6455458
##   15  0.8626984  0.6068904
##   17  0.8515873  0.5699765
##   19  0.8373016  0.5234547
##   21  0.8412698  0.5234547
##   23  0.8119048  0.4228945
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

```
#Predicted values
c_caret_pred <- predict(knn_fit, newdata = c_test)
```

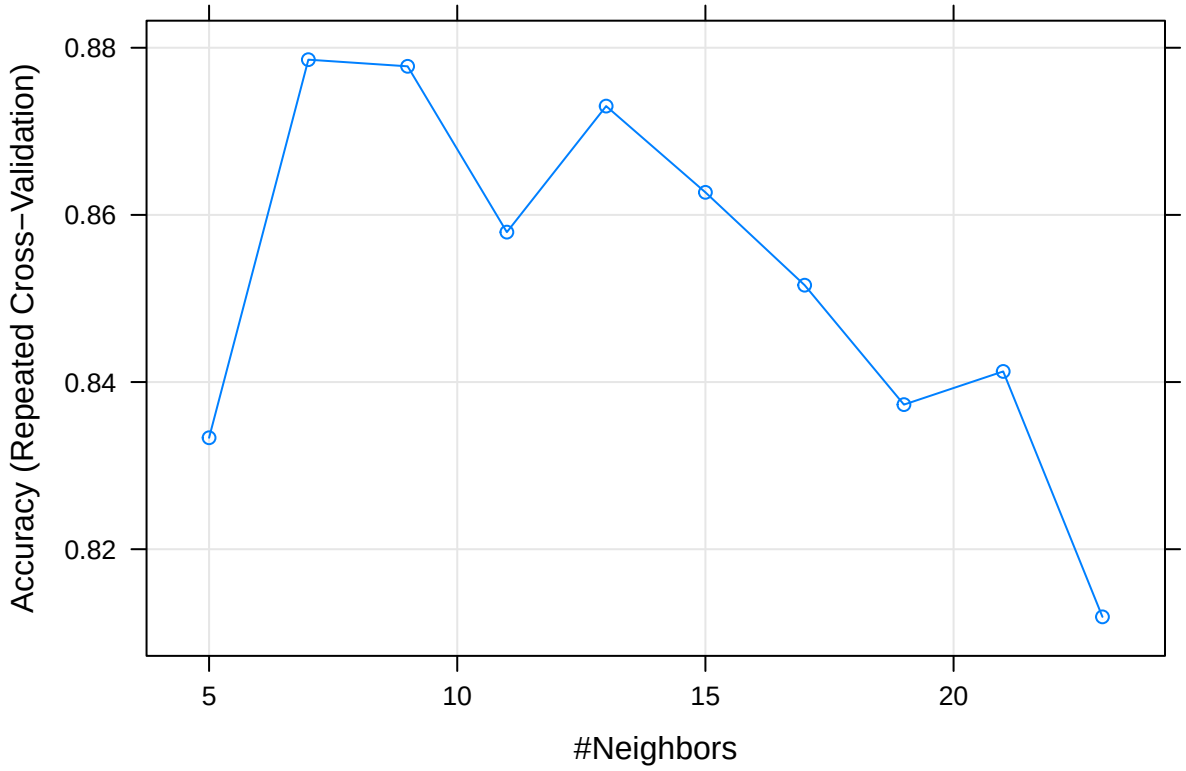
```
CrossTable(x= c_test_labels, y= c_caret_pred, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
```

```
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  35
##
##
##                | c_caret_pred
## c_test_labels |    Benign | Malignant | Row Total |
## --------------|-----------|-----------|-----------|
##        Benign |        10 |         9 |        19 |
##               |     0.526 |     0.474 |     0.543 |
##               |     1.000 |     0.360 |           |
##               |     0.286 |     0.257 |           |
## --------------|-----------|-----------|-----------|
##     Malignant |         0 |        16 |        16 |
##               |     0.000 |     1.000 |     0.457 |
##               |     0.000 |     0.640 |           |
##               |     0.000 |     0.457 |           |
## --------------|-----------|-----------|-----------|
##  Column Total |        10 |        25 |        35 |
##               |     0.286 |     0.714 |           |
## --------------|-----------|-----------|-----------|
##
##
```

Out of 35 cases 10 are True Positive, 16 True Negative (when positive class is Benign). Accuracy is determined by (TP+TN)/Total cases i.e. approx. 75%.

```
plot(knn_fit)
```




```
confusionMatrix(c_test_labels, c_caret_pred)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Benign Malignant
##   Benign        10         9
##   Malignant      0        16
##
##                Accuracy : 0.7429
##                  95% CI : (0.5674, 0.8751)
##     No Information Rate : 0.7143
##     P-Value [Acc > NIR] : 0.436332
##
##                   Kappa : 0.5039
##
##  Mcnemar's Test P-Value : 0.007661
##
##             Sensitivity : 1.0000
##             Specificity : 0.6400
##          Pos Pred Value : 0.5263
##          Neg Pred Value : 1.0000
##              Prevalence : 0.2857
##          Detection Rate : 0.2857
##    Detection Prevalence : 0.5429
##       Balanced Accuracy : 0.8200
##
##        'Positive' Class : Benign
##
```