Assignment 4

Mohamed Megahed

4/18/2020

Predicting the type of a breast tumor (benign or malignant) – Support Vector Machine Model

The data is loaded using the mlbench library, data(BreastCancer) A data frame with 699 observations on 11 variables, one being a character variable, 9 being ordered or nominal, and 1 target class.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

*** Understanding Data structure

```
summary(BreastCancer)
```

```
##       Id             Cl.thickness   Cell.size     Cell.shape  Marg.adhesio
n
##  Length:699         1      :145   1      :384   1      :353   1      :407
##  Class :character   5      :130   10     : 67   2      : 59   2      : 58
##  Mode  :character   3      :108   3      : 52   10     : 58   3      : 58
##                     4      : 80   2      : 45   3      : 56   10     : 55
##                     10     : 69   4      : 40   4      : 44   4      : 33
##                     2      : 50   5      : 30   5      : 34   8      : 25
##                     (Other):117   (Other): 81   (Other): 95   (Other): 63
##   Epith.c.size  Bare.nuclei   Bl.cromatin  Normal.nucleoli    Mitoses
##  2      :386   1      :402   2      :166   1      :443     1      :579
##  3      : 72   10     :132   3      :165   10     : 61     2      : 35
##  4      : 48   2      : 30   1      :152   3      : 44     3      : 33
##  1      : 47   5      : 30   7      : 73   2      : 36     10     : 14
##  6      : 41   3      : 28   4      : 40   8      : 24     4      : 12
##  5      : 39   (Other): 61   5      : 34   6      : 22     7      :  9
##  (Other): 66   NA's   : 16   (Other): 69   (Other): 69     (Other): 17
##        Class
##  benign   :458
##  malignant:241
##
##
##
##
##
```

```
levels(BreastCancer$Class)

## [1] "benign"    "malignant"
```

** checking if there any missing data

```
# Check if there are any missing values:
anyNA(BreastCancer)

## [1] TRUE

sum(is.na(BreastCancer))

## [1] 16
```

** we have 16 missing values in our dataset. ### Cleaning missing and excluding the ID variable

```
B_Cancer <- na.omit(BreastCancer)[,c(2:11)]

set.seed(123)
intrain <- createDataPartition(y = B_Cancer$Class, p= 0.7, list = FALSE)
train_data <- B_Cancer[intrain,]
test_data <- B_Cancer[-intrain,]

set.seed(123)
svm.model<-train(Class~.,data=train_data,method='svmLinear', scale = FALSE)
svm.model

## Support Vector Machines with Linear Kernel
##
## 479 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 479, 479, 479, 479, 479, 479, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9447694  0.8788296
##
## Tuning parameter 'C' was held constant at a value of 1
```

Examining cost values (1,2.5,6.25) and Predicting the test data

```
set.seed(123)
Grid_Serach <- expand.grid(.C=c(1,2.5,6.25))
#Building a support vector machine model
svm_Grid<-train(Class~.,
                data=train_data,
                method='svmLinear',
```

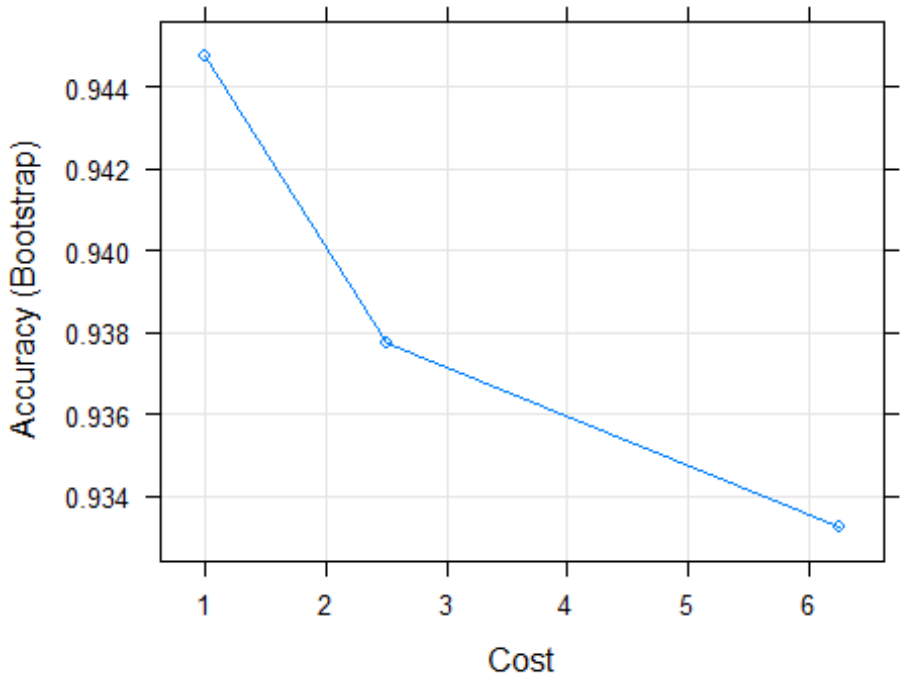
```
                tuneGrid=Grid_Serach,
                scale = FALSE)
svm_Grid
```

```
## Support Vector Machines with Linear Kernel
##
## 479 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 479, 479, 479, 479, 479, 479, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   1.00  0.9447694  0.8788296
##   2.50  0.9377325  0.8633873
##   6.25  0.9332128  0.8533543
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 1.
```

```
preds_svm_test <- predict(svm_Grid, test_data[1:9])  # predicting with the ne
w SVM model
table(pred = preds_svm_test, true = test_data$Class)
```

```
##            true
## pred        benign malignant
##   benign       130         5
##   malignant      3        66
```

```
plot(svm_Grid)
```



After examining the (1,2.5,6.25) values of cost it is clear that when Cost is equal 1 it has the best accuracy which is 94.47%