Machine Learing Hw5

Ekta Chaudhary

30/04/2020

```
library(ISLR)
library(mlbench)
library(caret)
```

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

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

```
library(e1071)
```

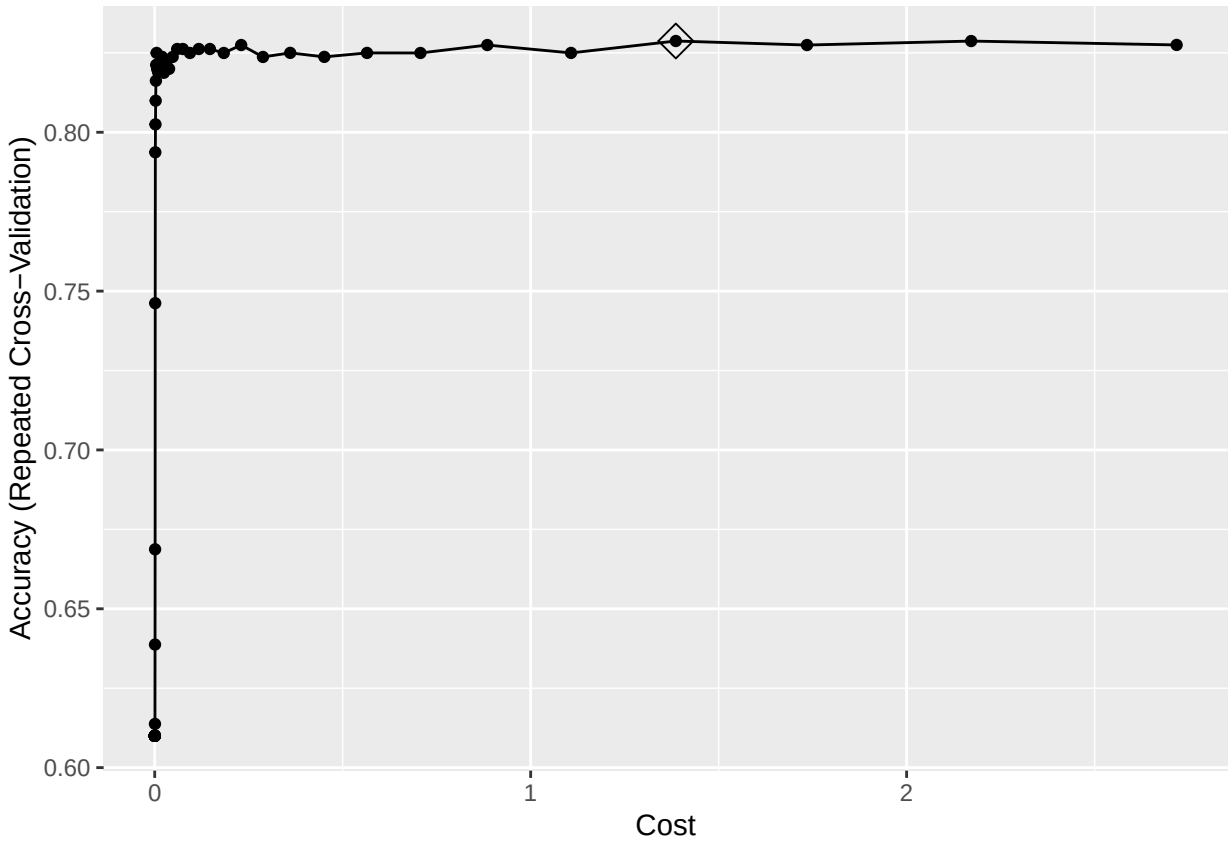
Data

This problem involves the OJ data set which is part of the ISLR package. The data contains 1070 purchases where the customer either purchased Citrus Hill or Minute Maid Orange Juice. A number of characteristics of the customer and product are recorded.Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

```
data(OJ)
set.seed(1)
rowTrain = createDataPartition(y = OJ$Purchase,
                               p = 0.747,
                               list = FALSE)
ctrl <- trainControl(method = "repeatedcv")
```

Question 1) Fit a support vector classifier (linear kernel) to the training data with Purchase as the response and the other variables as predictors. What are the training and test error rates?

```
set.seed(1)
svml.fit <- train(Purchase ~.,
                  data = OJ[rowTrain,],
                  method = "svmLinear2",
                  preProcess = c("center", "scale"),
                  tuneGrid = data.frame(cost = exp(seq(-10, 1, len = 50))),
                  trControl = ctrl)
ggplot(svml.fit, highlight = TRUE)
```



Calculating the training error rate:

```
pred.svml_training <- predict(svml.fit, newdata = OJ[rowTrain,])
confusionMatrix(data = pred.svml_training,
                reference = OJ$Purchase[rowTrain])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 439  80
##         MM  49 232
##
##                Accuracy : 0.8388
##                  95% CI : (0.8114, 0.8636)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6549
##
##  Mcnemar's Test P-Value : 0.008258
##
##             Sensitivity : 0.8996
##             Specificity : 0.7436
```

```
##          Pos Pred Value : 0.8459
##          Neg Pred Value : 0.8256
##              Prevalence : 0.6100
##          Detection Rate : 0.5487
##    Detection Prevalence : 0.6488
##       Balanced Accuracy : 0.8216
##
##        'Positive' Class : CH
##
```

```
linear_training_error_rate = mean(pred.svml_training != OJ$Purchase[rowTrain]) * 100
linear_training_error_rate
```

```
## [1] 16.125
```

#The trainig error rate is 16.125%

Finding the test error rate:

```
pred.svml_testing <- predict(svml.fit, newdata = OJ[-rowTrain,])
confusionMatrix(data = pred.svml_testing,
                reference = OJ$Purchase[-rowTrain])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 145  21
##         MM  20  84
##
##                Accuracy : 0.8481
##                  95% CI : (0.7997, 0.8888)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.68
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8788
##             Specificity : 0.8000
##          Pos Pred Value : 0.8735
##          Neg Pred Value : 0.8077
##              Prevalence : 0.6111
##          Detection Rate : 0.5370
##    Detection Prevalence : 0.6148
##       Balanced Accuracy : 0.8394
##
##        'Positive' Class : CH
##
```

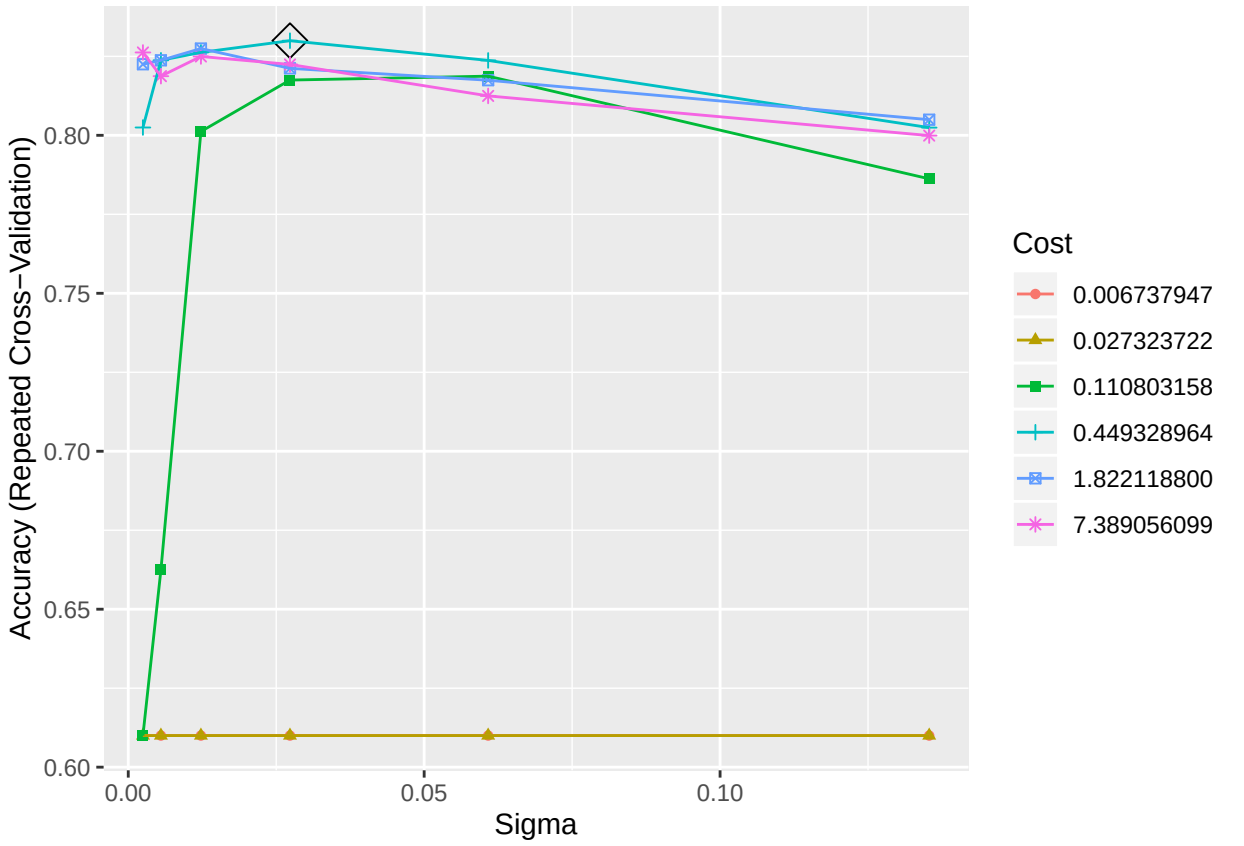
```
linear_testing_error_rate = mean(pred.svml_testing != OJ$Purchase[-rowTrain]) * 100
linear_testing_error_rate
```

```
## [1] 15.18519
```

The testing error rate is 15.18519%

Question 2) Fit a support vector machine with a radial kernel to the training data. What are the training and test error rates?

```
svmr.grid <- expand.grid(C = exp(seq(-5, 2, len = 6)),
                         sigma = exp(seq(-6,-2, len = 6)))
set.seed(1)
svmrad.fit <- train(Purchase ~., OJ,
                    subset = rowTrain,
                    method = "svmRadial",
                    preProcess = c("center", "scale"),
                    tuneGrid = svmr.grid,
                    trControl = ctrl)
ggplot(svmrad.fit, highlight = TRUE)
```



Calculating the Training error rate:

```
pred.svmrad_training <- predict(svmrad.fit, newdata = OJ[rowTrain,])
confusionMatrix(data = pred.svmrad_training,
                reference = OJ$Purchase[rowTrain])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 439  78
##         MM  49 234
##
##                Accuracy : 0.8412
##                  95% CI : (0.8141, 0.8659)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6607
##
##  Mcnemar's Test P-Value : 0.01297
##
##             Sensitivity : 0.8996
##             Specificity : 0.7500
##          Pos Pred Value : 0.8491
##          Neg Pred Value : 0.8269
##              Prevalence : 0.6100
##          Detection Rate : 0.5487
##    Detection Prevalence : 0.6462
##       Balanced Accuracy : 0.8248
##
##        'Positive' Class : CH
##
```

```
radial_training_error_rate = mean(pred.svmrad_training != OJ$Purchase[rowTrain]) * 100
radial_training_error_rate
```

```
## [1] 15.875
```

The training error rate is 15.875%

Calculating the Testing error rate

```
pred.svmrad_testing <- predict(svmrad.fit, newdata = OJ[-rowTrain,])
confusionMatrix(data = pred.svmrad_testing,
                reference = OJ$Purchase[-rowTrain])
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction  CH  MM
##         CH 147  25
##         MM  18  80
##
##                Accuracy : 0.8407
##                  95% CI : (0.7915, 0.8823)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6608
##
##  Mcnemar's Test P-Value : 0.3602
##
##             Sensitivity : 0.8909
##             Specificity : 0.7619
##          Pos Pred Value : 0.8547
##          Neg Pred Value : 0.8163
##              Prevalence : 0.6111
##          Detection Rate : 0.5444
##    Detection Prevalence : 0.6370
##       Balanced Accuracy : 0.8264
##
##        'Positive' Class : CH
##
```

```
radial_testing_error_rate = mean(pred.svmrad_testing != OJ$Purchase[-rowTrain]) * 100
radial_testing_error_rate
```

```
## [1] 15.92593
```

The testing error rate is 15.92%