HW5

Amin Yakubu

4/24/2019

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

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

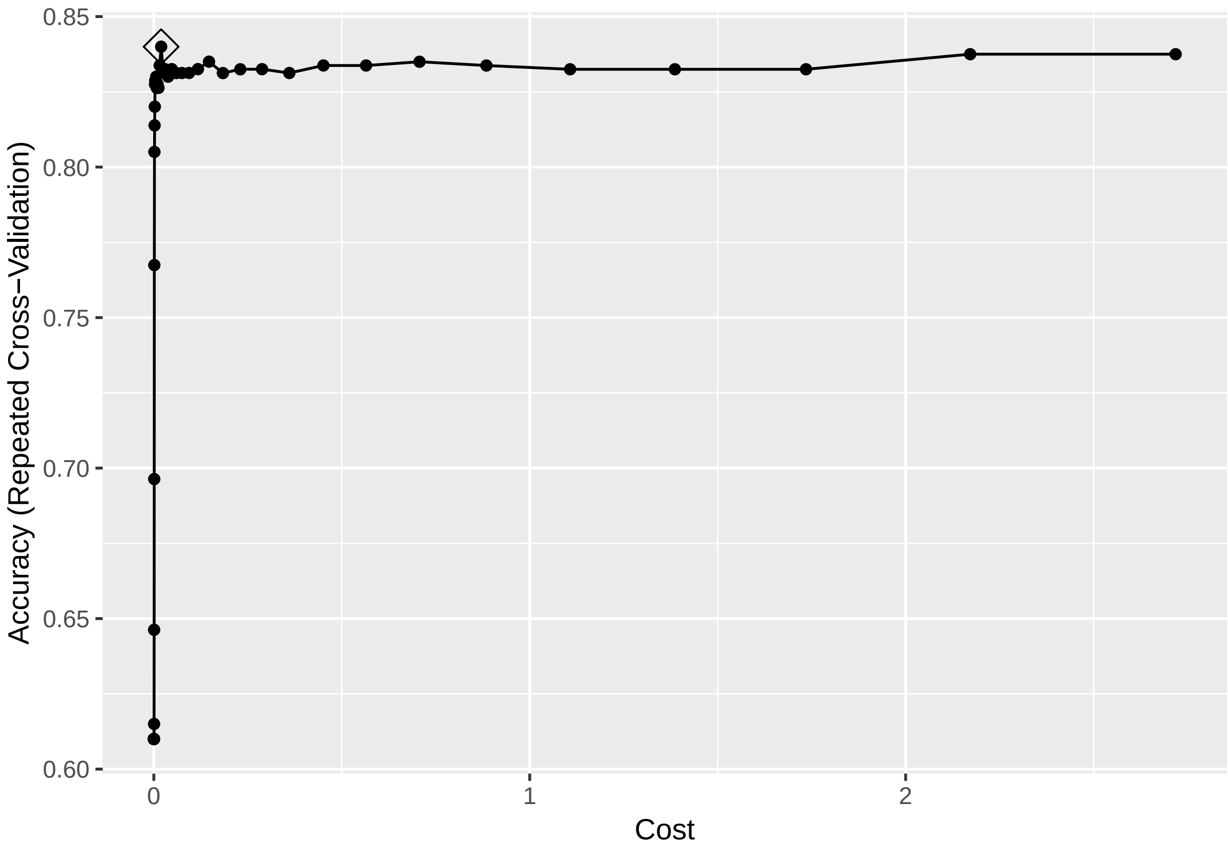
```r
library(e1071)
```

Data

```r
data(OJ)
seed = 58639

set.seed(seed)
rowTrain = createDataPartition(y = OJ$Purchase,
                               p = 0.747,
                               list = FALSE)

ctrl <- trainControl(method = "repeatedcv")
```

Question A

```r
set.seed(seed)
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)
```



Training error using the training data

```
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  75
##         MM  49 237
##
##                Accuracy : 0.845
##                  95% CI : (0.818, 0.8694)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6693
##  Mcnemar's Test P-Value : 0.02476
##
##             Sensitivity : 0.8996
##             Specificity : 0.7596
##          Pos Pred Value : 0.8541
##          Neg Pred Value : 0.8287
##              Prevalence : 0.6100
##          Detection Rate : 0.5487
```

```
##    Detection Prevalence : 0.6425
##       Balanced Accuracy : 0.8296
##
##        'Positive' Class : CH
##
```

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

```
## [1] 15.5
```

Trainig error rate is 15.5%

Test error using the held-out data

```r
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 140  27
##         MM  25  78
##
##                Accuracy : 0.8074
##                  95% CI : (0.7552, 0.8527)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 3.059e-12
##
##                   Kappa : 0.5934
##  Mcnemar's Test P-Value : 0.8897
##
##             Sensitivity : 0.8485
##             Specificity : 0.7429
##          Pos Pred Value : 0.8383
##          Neg Pred Value : 0.7573
##              Prevalence : 0.6111
##          Detection Rate : 0.5185
##    Detection Prevalence : 0.6185
##       Balanced Accuracy : 0.7957
##
##        'Positive' Class : CH
##
```

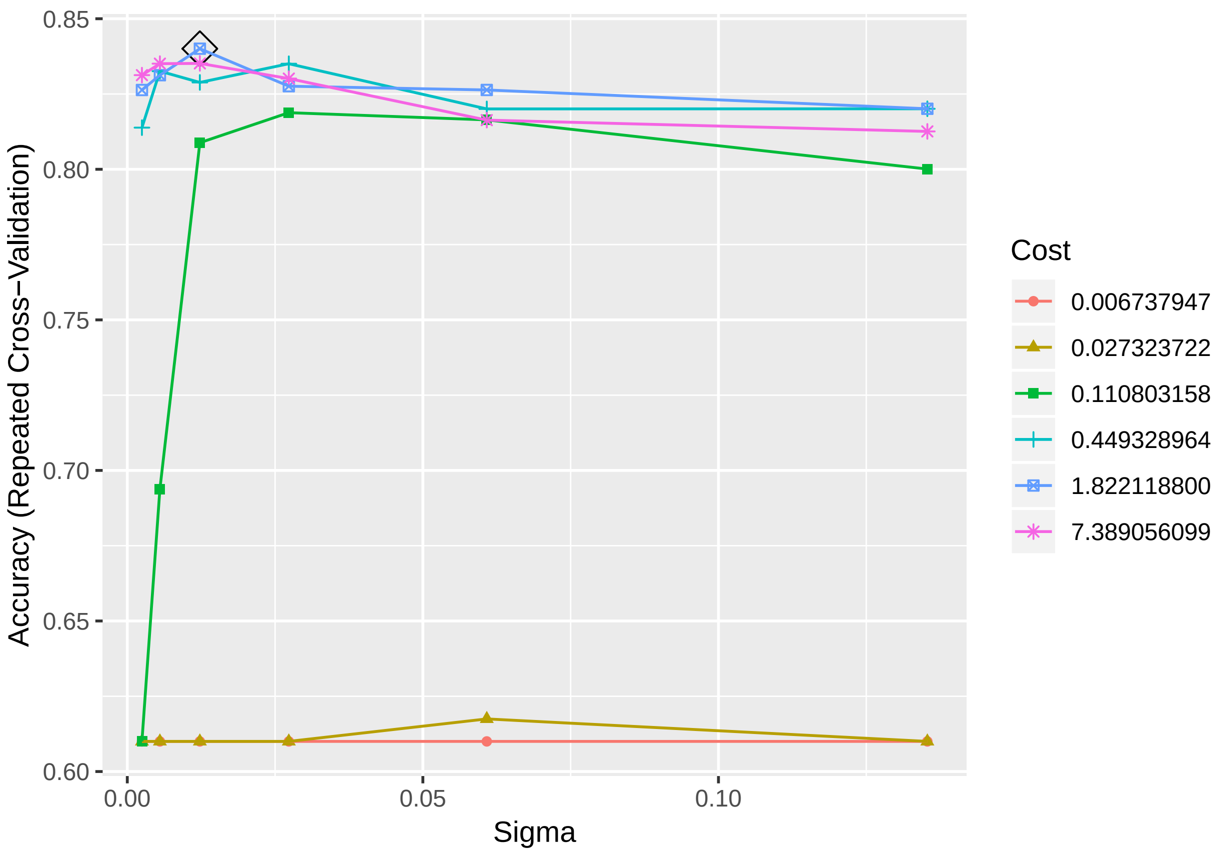
```r
linear_testing_error_rate = mean(pred.svml_testing != OJ$Purchase[-rowTrain]) * 100
linear_testing_error_rate
```

```
## [1] 19.25926
```

The testing error rate is 19.259%

Question B

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

ggplot(svmrad.fit, highlight = TRUE)
```



Now let's see what the training error rate is for the support vector machine with a radial kernel.

```
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 442  69
##         MM  46 243
```

```
##
##                Accuracy : 0.8562
##                  95% CI : (0.83, 0.8798)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6938
##  Mcnemar's Test P-Value : 0.04022
##
##             Sensitivity : 0.9057
##             Specificity : 0.7788
##          Pos Pred Value : 0.8650
##          Neg Pred Value : 0.8408
##              Prevalence : 0.6100
##          Detection Rate : 0.5525
##    Detection Prevalence : 0.6388
##       Balanced Accuracy : 0.8423
##
##        'Positive' Class : CH
##
```

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

```
## [1] 14.375
```

The training error rate is 14.375%

Now let's find out 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 141  26
##         MM  24  79
##
##                Accuracy : 0.8148
##                  95% CI : (0.7633, 0.8593)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 4.049e-13
##
##                   Kappa : 0.609
##  Mcnemar's Test P-Value : 0.8875
##
##             Sensitivity : 0.8545
##             Specificity : 0.7524
##          Pos Pred Value : 0.8443
##          Neg Pred Value : 0.7670
##              Prevalence : 0.6111
##          Detection Rate : 0.5222
```

```
##    Detection Prevalence : 0.6185
##       Balanced Accuracy : 0.8035
##
##        'Positive' Class : CH
##
```

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

```
## [1] 18.51852
```

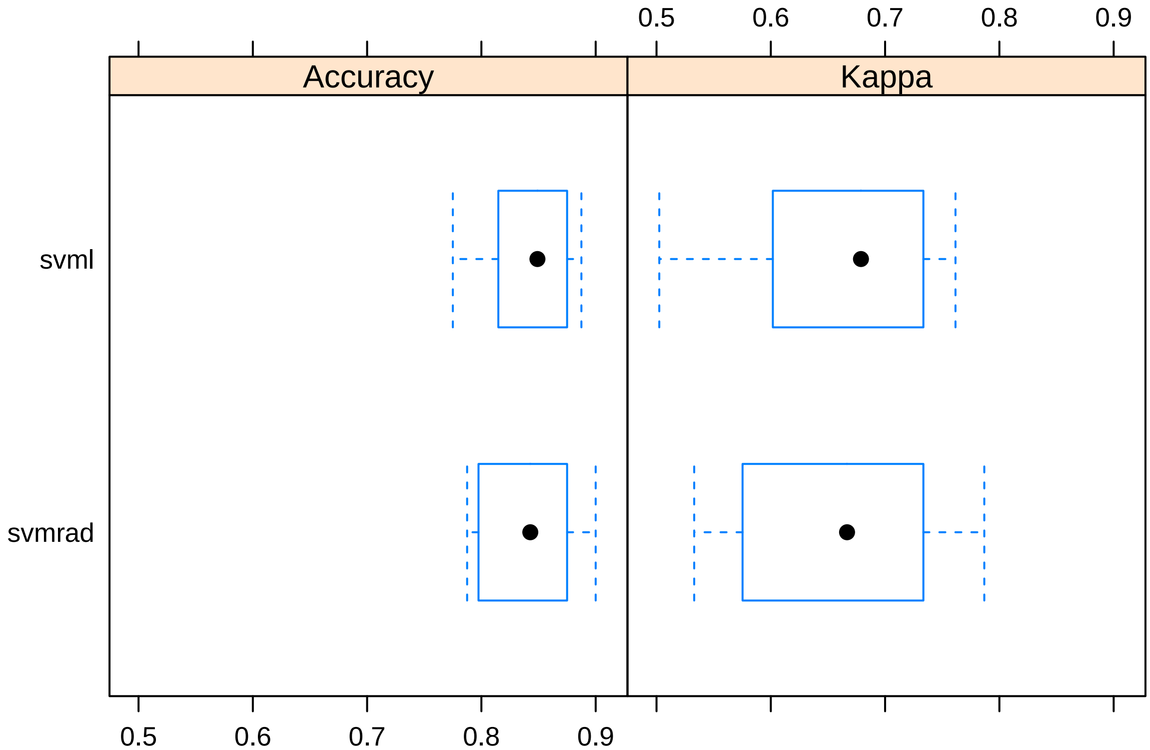
We expect the testing error to higher than the training error rate. The testing error rate is 18.5185%

Question C

```r
resamp <- resamples(list(svmrad = svmrad.fit, svml = svml.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: svmrad, svml
## Number of resamples: 10
##
## Accuracy
##          Min.   1st Qu.    Median      Mean  3rd Qu.   Max. NA's
## svmrad 0.7875 0.8018050 0.8428006 0.8400508 0.871875 0.9000    0
## svml   0.7750 0.8179012 0.8490506 0.8399887 0.871875 0.8875    0
##
## Kappa
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## svmrad 0.5329670 0.5829913 0.6666997 0.6564983 0.7264067 0.7868088    0
## svml   0.5024188 0.6068274 0.6788917 0.6584110 0.7281437 0.7615894    0
```

```r
bwplot(resamp)
```



In model selection we don't use the testing or training error, rather we use the cross validation error. Based on the median cross validation results from resamples, we see that the linear kernel has a higher a accuracy and seems to give better results, and therefore will be the preffered model in this case. Also, from the resamples summary, we see that the median cross validation error is slightly higher for the linear kernel method. The median kappa values are also very similar.