# HW5

*Amin Yakubu*

*4/24/2019*

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

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

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

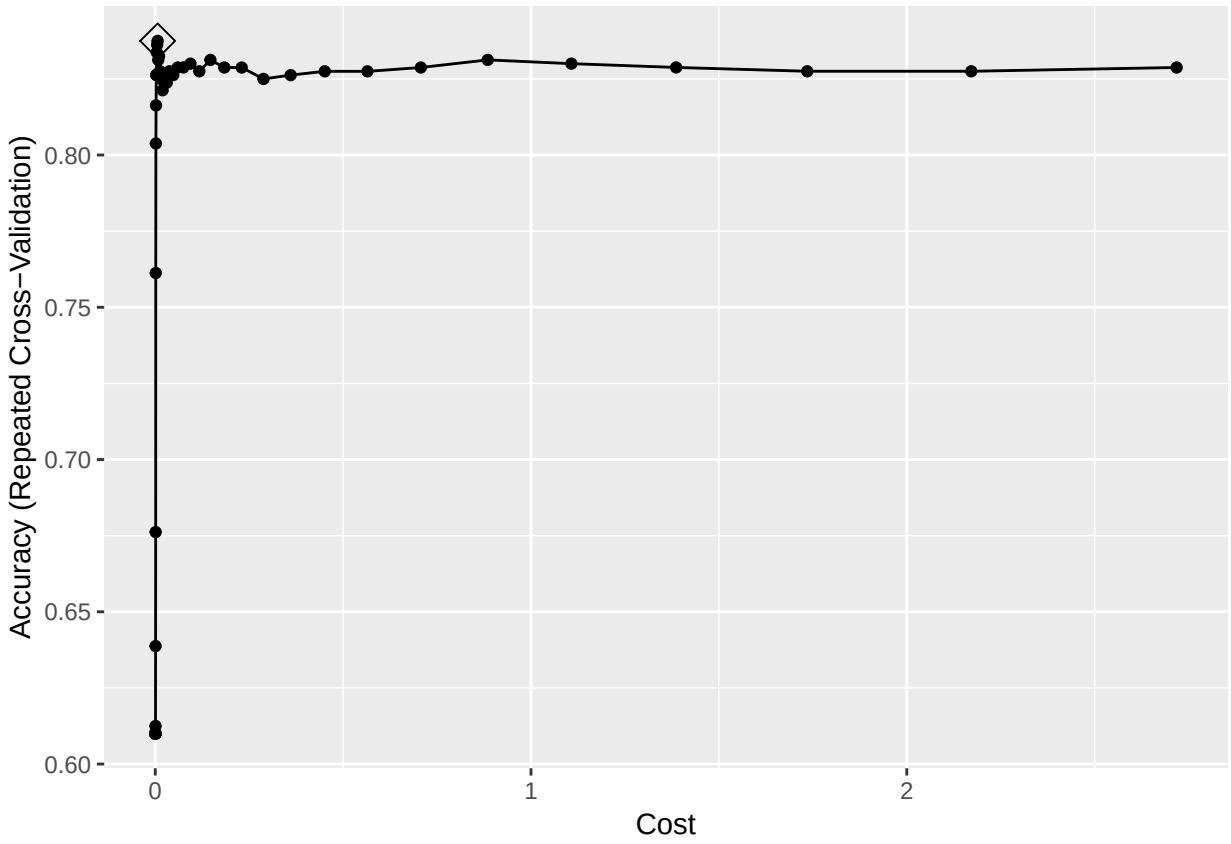
```
library(e1071)
```

## Data

```
data(OJ)
seed = 1

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

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

## Question A

```
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)
```




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 437  81
##         MM  51 231
##
##                Accuracy : 0.835
##                  95% CI : (0.8074, 0.8601)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6471
##  Mcnemar's Test P-Value : 0.0116
##
##             Sensitivity : 0.8955
##             Specificity : 0.7404
##          Pos Pred Value : 0.8436
##          Neg Pred Value : 0.8191
##              Prevalence : 0.6100
##          Detection Rate : 0.5463
```

```
##    Detection Prevalence : 0.6475
##       Balanced Accuracy : 0.8179
##
##        'Positive' Class : CH
##
```

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

```
## [1] 16.5
```

Test error using the held-out data

```
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 147  28
##         MM  18  77
##
##                Accuracy : 0.8296
##                  95% CI : (0.7794, 0.8725)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 5.295e-15
##
##                   Kappa : 0.6352
##  Mcnemar's Test P-Value : 0.1845
##
##             Sensitivity : 0.8909
##             Specificity : 0.7333
##          Pos Pred Value : 0.8400
##          Neg Pred Value : 0.8105
##              Prevalence : 0.6111
##          Detection Rate : 0.5444
##    Detection Prevalence : 0.6481
##       Balanced Accuracy : 0.8121
##
##        'Positive' Class : CH
##
```

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

```
## [1] 17.03704
```

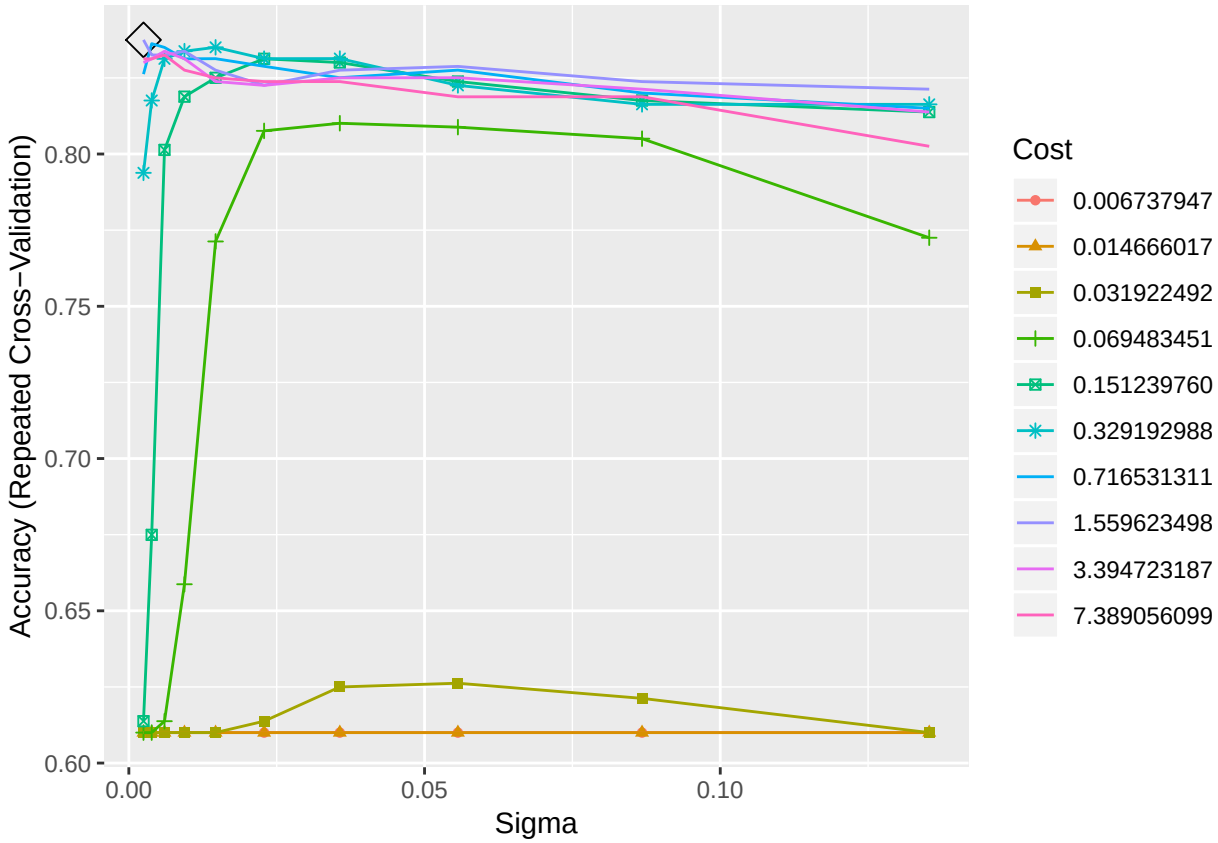
# Question B

```
svmr.grid <- expand.grid(C = exp(seq(-5, 2, len = 10)),
                         sigma = exp(seq(-6,-2, len = 10))) # This sigma is the same as gamma in the sv
set.seed(1)
```

```
svmr.fit <- train(Purchase ~., OJ,
                  subset = rowTrain,
                  method = "svmRadial",
                  preProcess = c("center", "scale"),
                  tuneGrid = svmr.grid,
                  trControl = ctrl)

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

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have
## 10. Consider specifying shapes manually if you must have them.
```

```
## Warning: Removed 40 rows containing missing values (geom_point).
```




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

```
pred.svmr_training <- predict(svmr.fit, newdata = OJ[rowTrain,])

confusionMatrix(data = pred.svmr_training,
                reference = OJ$Purchase[rowTrain])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 438  79
##         MM  50 233
```

```
##
##                Accuracy : 0.8388
##                  95% CI : (0.8114, 0.8636)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6553
##  Mcnemar's Test P-Value : 0.01369
##
##             Sensitivity : 0.8975
##             Specificity : 0.7468
##          Pos Pred Value : 0.8472
##          Neg Pred Value : 0.8233
##              Prevalence : 0.6100
##          Detection Rate : 0.5475
##    Detection Prevalence : 0.6462
##       Balanced Accuracy : 0.8222
##
##        'Positive' Class : CH
##
```

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

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

Now let's find out the testing error rate

```
pred.svmr_testing <- predict(svmr.fit, newdata = OJ[-rowTrain,])

confusionMatrix(data = pred.svmr_testing,
                reference = OJ$Purchase[-rowTrain])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 146  27
##         MM  19  78
##
##                Accuracy : 0.8296
##                  95% CI : (0.7794, 0.8725)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 5.295e-15
##
##                   Kappa : 0.6365
##  Mcnemar's Test P-Value : 0.302
##
##             Sensitivity : 0.8848
##             Specificity : 0.7429
##          Pos Pred Value : 0.8439
##          Neg Pred Value : 0.8041
##              Prevalence : 0.6111
##          Detection Rate : 0.5407
##    Detection Prevalence : 0.6407
##       Balanced Accuracy : 0.8139
```

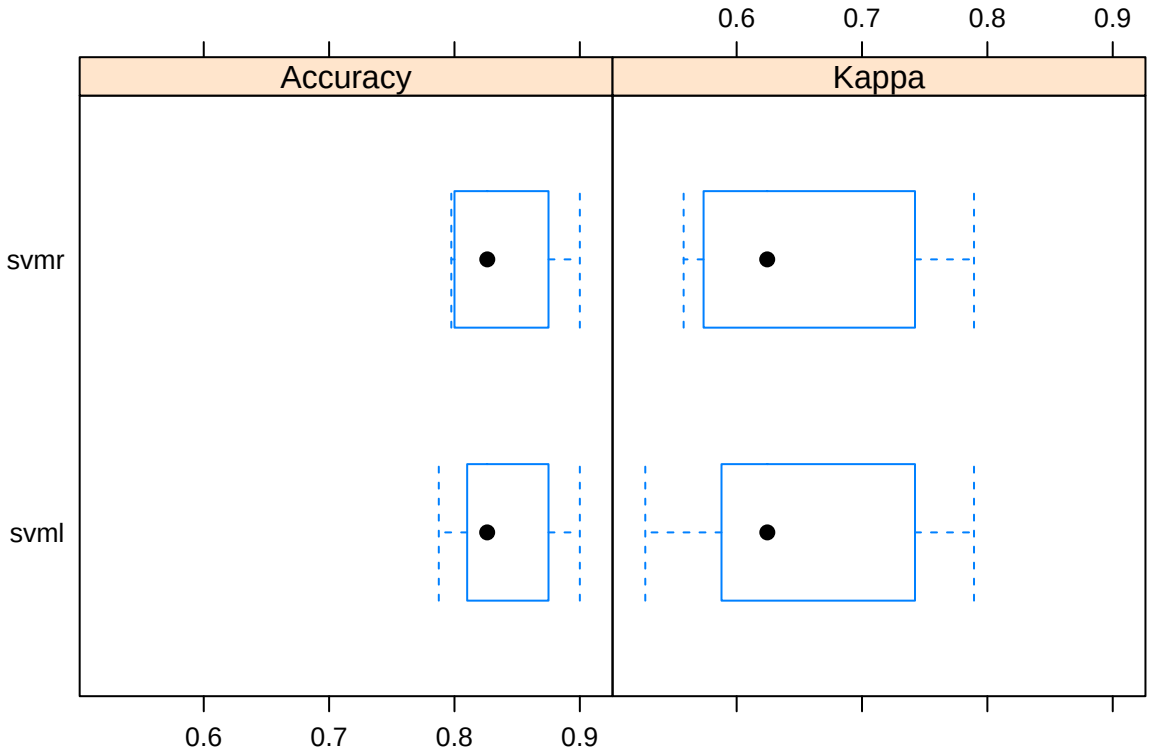
```
##
##        'Positive' Class : CH
##
```

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

```
## [1] 17.03704
```

## Question C

```
resamp <- resamples(list(svmr = svmr.fit, svml = svml.fit))
bwplot(resamp)
```




Based on the cross validation results from resamples, we see that the radial kernel has a higher a accuracy and therefore will be the preffered model in this case.