Support Vector Machines (SVM)

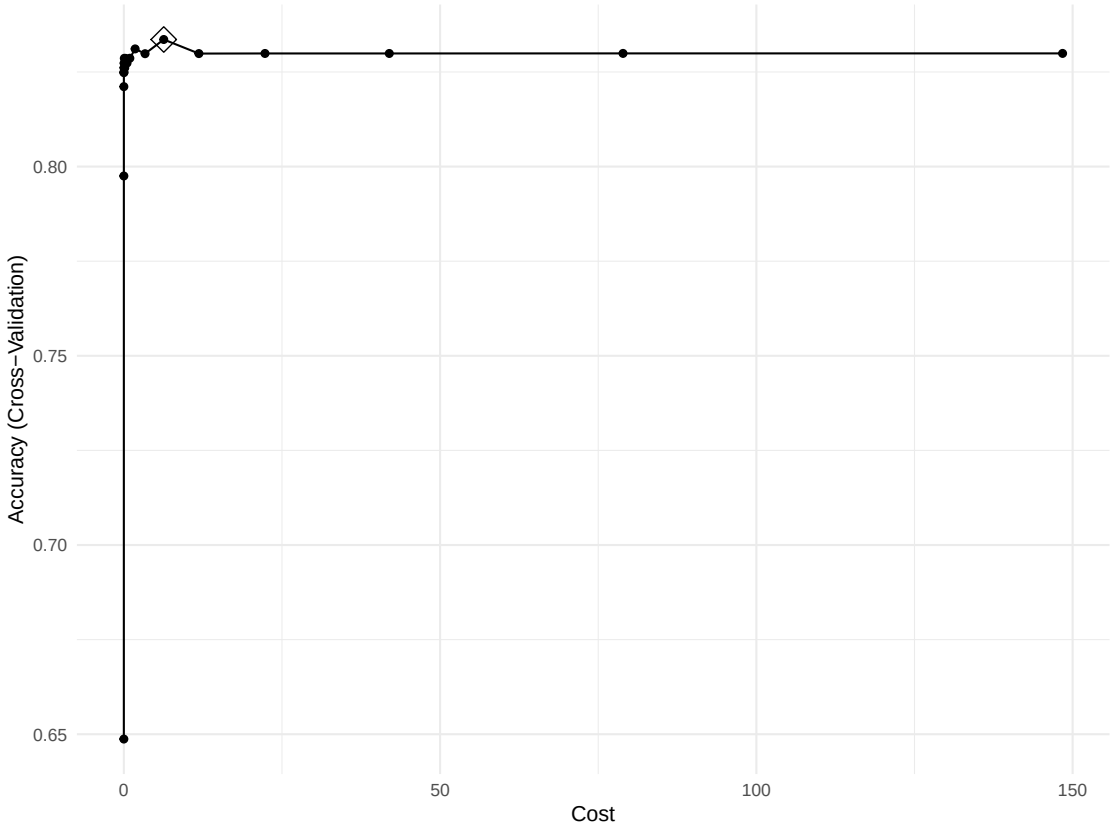
Hana Akbarnejad

5/10/2020

First I fit a support vector classifier (linear kernel) to the training data.

```
set.seed(2020)
# SWM with linear kernel, caret
svml_fit = train(purchase~.,
                 data = train_data,
                 method = "svmLinear2",
                 preProcess = c("center", "scale"),
                 tuneGrid = data.frame(cost = exp(seq(-7,5,len=20))),
                 trControl = ctrl)

ggplot(svml_fit, highlight = TRUE)
```



```
svml_fit$bestTune
```

```
##        cost
## 15 6.309808
```

```
summary(svml_fit)
```

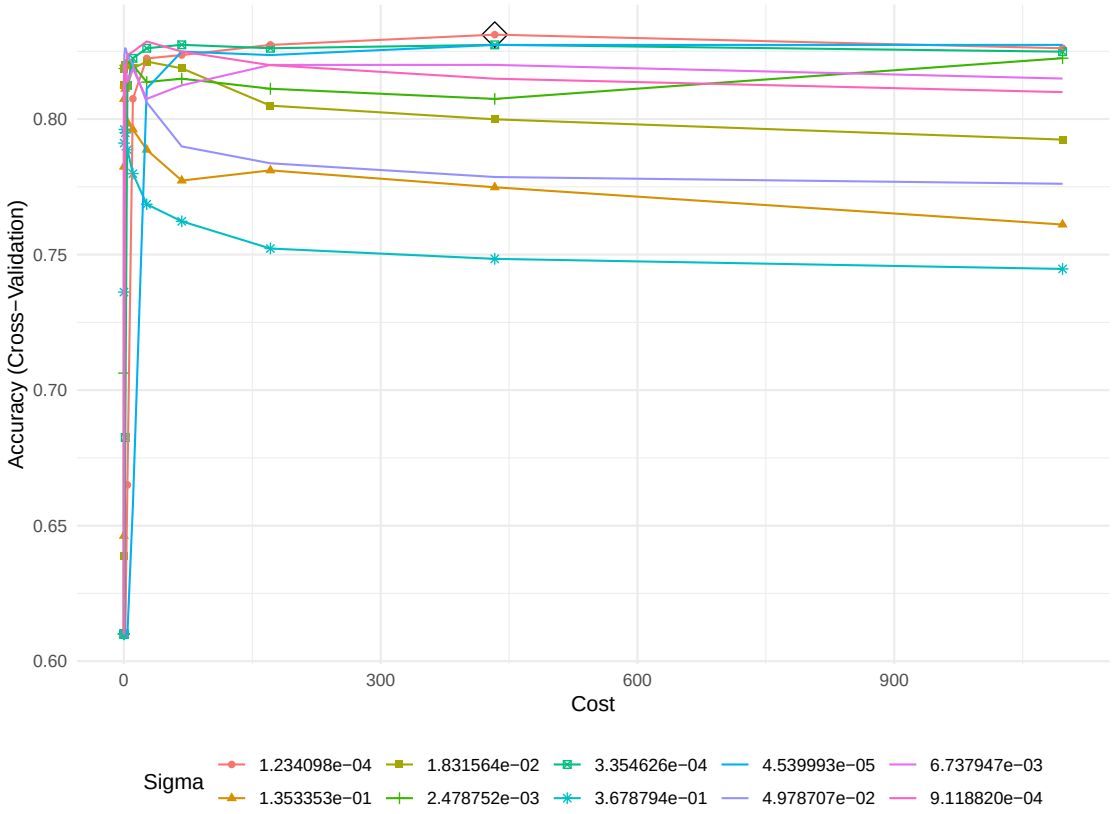
```
##
## Call:
## svm.default(x = as.matrix(x), y = y, kernel = "linear", cost = param$cost,
##     probability = classProbs)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  6.309808
##
## Number of Support Vectors:  337
##
##  ( 170 167 )
##
##
## Number of Classes:  2
##
## Levels:
##  CH MM
```

Then, I fit a support vector machine with a radial kernel to the training data.

```
set.seed(2020)
svmr_grid = expand.grid(C = exp(seq(-6,7,len=15)),
                        sigma = exp(seq(-10,-1,len=10)))

svmr_fit = train(purchase~., train_data,
                 method = "svmRadial",
                 preProcess = c("center", "scale"),
                 tuneGrid = svmr_grid,
                 trControl = ctrl)

ggplot(svmr_fit, highlight = TRUE)
```




```
svmr_fit$bestTune
```

```
##            sigma        C
## 132 0.0001234098 433.2992
```

```
summary(svmr_fit)
```

```
## Length  Class   Mode
##      1   ksvm     S4
```

Then, I look at training error rates of both SVM models with linare and radial kernels

```
set.seed(2020)

#looking at train performance of linear kernel
svml_pred2 = predict(svml_fit, newdata = train_data)

cm_linear = confusionMatrix(data = svml_pred2,
                reference = oj_data$purchase[train_rows])


#looking at train performance of radial kernel
svmr_pred2 = predict(svmr_fit, newdata = train_data)
```

```
cm_radial = confusionMatrix(data = svmr_pred2,
                reference = oj_data$purchase[train_rows])
```

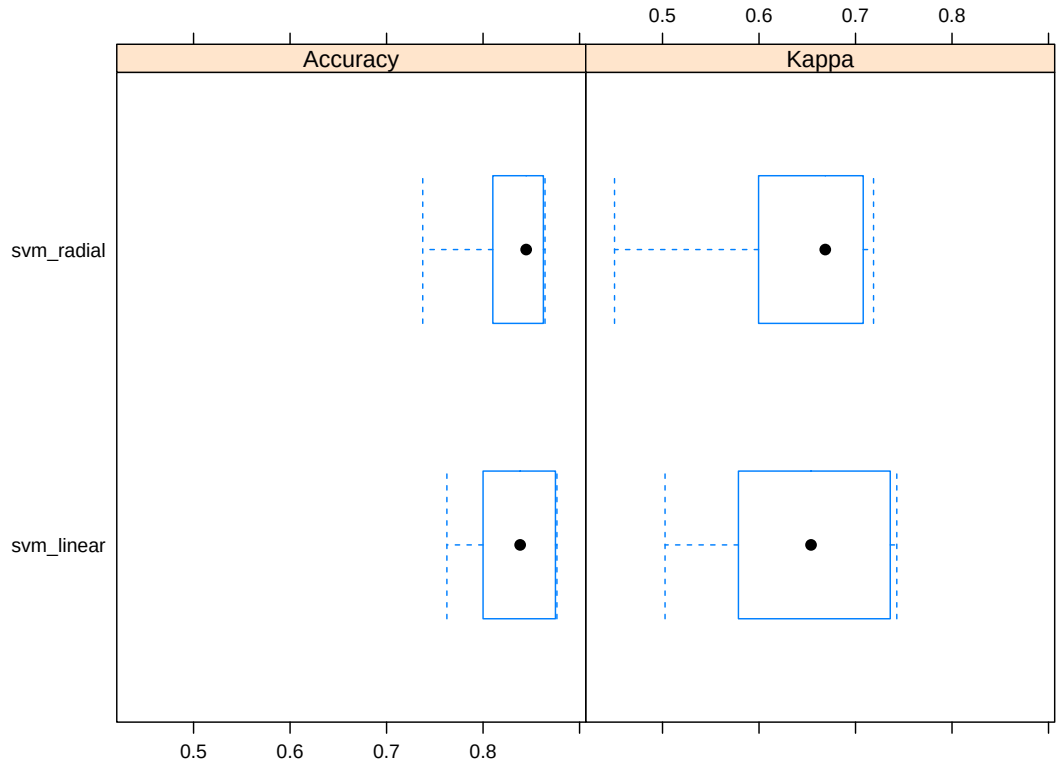
From confusion matrices above, we can observe that SVM with linear kernel has training error rate of 16.12% and SVM with radial kernel has training error rate of 16.25%. We can see that these values are very close to each other with less than 0.5% difference.

```
set.seed(2020)

resamp = resamples(list(svm_radial = svmr_fit, svm_linear = svml_fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: svm_radial, svm_linear
## Number of resamples: 10
##
## Accuracy
##              Min.   1st Qu.    Median      Mean 3rd Qu.      Max. NA's
## svm_radial 0.7375 0.8107199 0.8446759 0.8311303  0.8625 0.8641975    0
## svm_linear 0.7625 0.8025316 0.8384259 0.8335832  0.8750 0.8765432    0
##
## Kappa
##                 Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## svm_radial 0.4502618 0.6003074 0.6686347 0.6412861 0.7054240 0.7186701    0
## svm_linear 0.5026178 0.5815294 0.6538992 0.6467705 0.7337289 0.7427653    0
```

```
bwplot(resamp)
```



Also, the summary of *resamples()* function show the mean accuracy of 0.8335 (cross validation error = 16.65%) for SVM with linear kernel and the mean accuracy of of 0.8311 (cross validation error rate = 16.89%) for SVM with radial kernel. So, if we want to select one of these models, I think the one with linear kernel is a better choice because the model is less complicated while the cross validated error is not that different.

Finally, I look at test performance of both SVM models with linare and radial kernels

```
set.seed(2020)

#looking at test performance of linear kernel
svml_pred = predict(svml_fit, newdata = test_data)

cm_linear2 = confusionMatrix(data = svml_pred,
                reference = oj_data$purchase[-train_rows])

#looking at test performance of radial kernel
svmr_pred = predict(svmr_fit, newdata = test_data)

cm_radial2 = confusionMatrix(data = svmr_pred,
                reference = oj_data$purchase[-train_rows])
```

Looking at confusion matrix of two SVM models, we can see that test error rate of the one with linear kernel is 16.3% and the test error rate of the one with radial kernel is 15.56%. So, it seems that linear kernel and

radial kernel models' performance on test dataset are pretty much the same and pretty close to training errors.