SVM

```
train <- readRDS("./data/cog_train_preproc.RDS") %>% dplyr::select(-subject)
test <- readRDS("./data/cog_test_preproc.RDS") %>% dplyr::select(-subject)
```

With e1071

First I look at an SVM with a lienar kernel.

```
set.seed(1)
linear.tune <- e1071::tune.svm(cdr~.,
                               data = train,
                               kernel = "linear",
                               cost = exp(seq(-5, 1, len = 20)))
# summary(linear.tune)
plot(linear.tune)

best.linear <- linear.tune$best.model
summary(best.linear)

train.pred <- predict(best.linear, newdata = train)

confusionMatrix(data = train.pred,
                reference = train$cdr)

pred.linear <- predict(best.linear, newdata = test)

confusionMatrix(data = pred.linear,
                reference = test$cdr)
```

The training accuracy is 0.7729 (0.7392, 0.8043), with a Kappa or 0.389. The test accuracy is 0.762 (0.7125, 0.8068), with a Kappa of 0.369.

Next, I will look at an SVM with a radial kernel to see whether it is a better fit.

```
set.seed(1)
radial.tune <- e1071::tune.svm(cdr~.,
                               data = train,
                               kernel = "radial",
                               cost = exp(seq(-4, 5, len = 10)),
                               gamma = exp(seq(-8, -3, len = 5)))

# summary(radial.tune)


best.radial <- radial.tune$best.model
summary(best.radial)

train.pred.radial <- predict(best.radial, newdata = train)

confusionMatrix(data = train.pred.radial,
                reference = train$cdr)
```

```
pred.radial <- predict(best.radial, newdata = test)

confusionMatrix(data = pred.radial,
                reference = test$cdr)
```

The training accuracy is 0.7865 (0.7533, 0.817), with a Kappa of 0.4317. The test accuracy is 0.762 (0.7125, 0.8068), which is the same as with the linear kernel. The Kappa value is 0.381.

Overall, it looks like the predictive capability on test data is similar between models. The difference lies in the training accuracy, which is slightly higher in the SVM with the radial kernel. Since we're interested in accuracy, I would choose the SVM with radial kernel.

Now with caret

Linear kernel:

```
ctrl <- trainControl(method = "repeatedcv",
                     repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

set.seed(1)
svml.fit <- train(cdr~.,
                  data = train,
                  method = "svmLinear2",
                  metric = "roc",
                  tuneGrid = data.frame(cost = exp(seq(1,7,len=20))),
                  trControl = ctrl)

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

Radial kernel:

```
svmr.grid <- expand.grid(C = exp(seq(-4,5,len=10)),
                         sigma = exp(seq(-8,-3,len=5)))
set.seed(1)
svmr.fit <- train(cdr~.,
                  data = train,
                  method = "svmRadial",
                  metric = "roc",
                  tuneGrid = svmr.grid,
                  trControl = ctrl)

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

Model comparison with `resamples`.

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

It looks like the two SVMs are very similar in their performance with the training data, with an AUC around 0.8. The SVM with linear kernel gives a more precise (narrower CI) estimate of AUC.

Looking at the test data performance:

```
pred.svml <- predict(svml.fit, newdata = test)
pred.svmr <- predict(svmr.fit, newdata = test)

confusionMatrix(data = pred.svml,
                reference = test$cdr)

confusionMatrix(data = pred.svmr,
                reference = test$cdr)
```

Looking at the test data performance, the SVM with linear kernel is better (higher accruacy).

```
saveRDS(svml.fit, "./data/SVM.RDS")
```