HW1 Num3

Aaron Coates

4/17/2019

```
setwd("~/Documents/GitHub/MMSS_311_2")
pol <- read.csv("/Users/aaroncoates/Downloads/pol_data.csv")
library(broom)
library(e1071)
library(caret)
```

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

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

First, I separate the data into training and test sets.

```
pol$group <- as.factor(pol$group)
trainsize <- floor((2/3)*nrow(pol))
set.seed(100)
train_pol <- sample(nrow(pol), size = trainsize, replace=FALSE)
trainingdata <- pol[train_pol, ]
testydata <- pol[-train_pol, ]
```

Now, I will use SVM to classify the data. I use tune() to find the cost level that minimizes error.

```
tunez <- tune(svm, group ~ pol_margin + col_degree + house_income,
              data=trainingdata, kernel = "linear",
              ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))
swaggysvm <- tunez$best.model
summary(swaggysvm)
```

```
##
## Call:
## best.tune(method = svm, train.x = group ~ pol_margin + col_degree +
##     house_income, data = trainingdata, ranges = list(cost = c(0.001,
##     0.01, 0.1, 1, 5, 10, 100)), kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##       gamma:  0.3333333
##
## Number of Support Vectors:  46
##
##  ( 23 23 )
##
##
## Number of Classes:  2
##
## Levels:
##  Politicalist Socialcrat
```

Next, I will predict the results and display these in a table.

```
svmpredict <- predict(swaggysvm, testydata)
svmtable <- table("Prediction"=svmpredict, "True Party"=testydata[,1])
svmtable
```

```
##                True Party
## Prediction     Politicalist Socialcrat
##   Politicalist           41          3
##   Socialcrat              0         56
```

So, the SVM is correct for 97% of the observations.

Now, I will perform the same steps for Naive Bayes.

```
naive <- naiveBayes(group ~ pol_margin + col_degree + house_income, data=trainingdata)
```

I will use the model above to make predictions about the test data.

```
naivepred <- predict(naive, testydata)
naivetable <- table("Prediction"=naivepred, "True Party"=testydata[,1])
naivetable
```

```
##                True Party
## Prediction     Politicalist Socialcrat
##   Politicalist           41          2
##   Socialcrat              0         57
```

So, the Naive Bayes model is correct 98% of the time.