SVM Classification rcd180001

Joshua Durana

2022-10-23

Source: https://www.kaggle.com/datasets/danofer/law-school-admissions-bar-passage (https://www.kaggle.com/datasets/danofer/law-school-admissions-bar-passage)

Load and Clean Data

```
BARdata <- read.csv("Data/BarData.csv", header = TRUE, na.strings = c("", "NA"))

#Remove unneccessary factors
BARdata <- subset(BARdata, select = c(lsat, bar_passed, ugpa))

#Remove NAs
BARdata <- na.omit(BARdata)

#Sets columns into factors
setToFactors <- c("bar_passed")
BARdata[setToFactors] <- lapply(BARdata[setToFactors], as.factor)
```

Split Data

```
set.seed(1022)

spl <- c(train = .6, test = .2, validate = .2)
i <- sample(cut(1:nrow(BARdata), nrow(BARdata) * cumsum(c(0, spl)), labels = names(spl)))

BARTrain <- BARdata[i == "train",]
BARTest <- BARdata[i == "test",]
BARVal <- BARdata[i == "validate",]
```

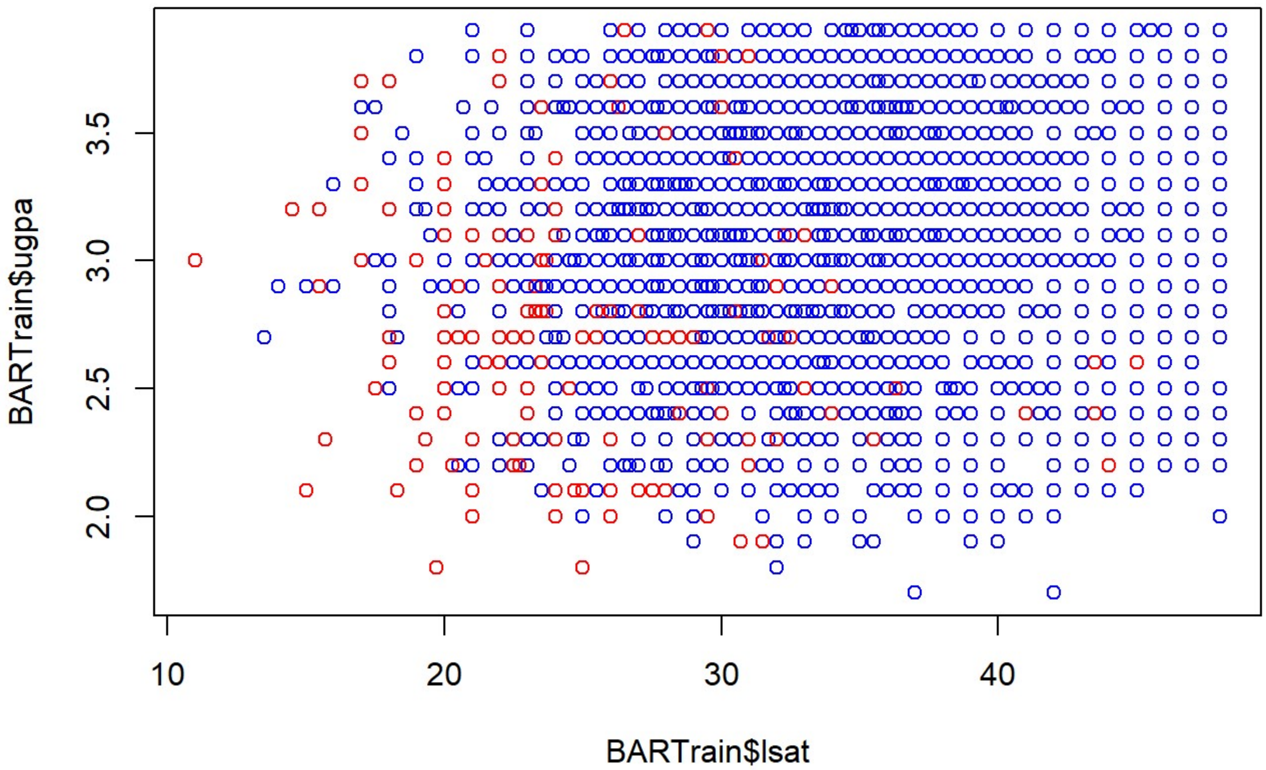
Data Exploration

```
summary(BARTrain)
```

```
##       lsat       bar_passed        ugpa
##  Min.   :11.00   FALSE:  689   Min.   :1.700
##  1st Qu.:33.00   TRUE :12755   1st Qu.:3.000
##  Median :37.00                 Median :3.200
##  Mean   :36.78                 Mean   :3.218
##  3rd Qu.:41.00                 3rd Qu.:3.500
##  Max.   :48.00                 Max.   :3.900
```

It seems that the majority of people pass the bar

```
plot(BARTrain$lsat, BARTrain$ugpa, col = c("red", "blue") [unclass(BARTrain$bar_passed)])
```



It seems like the fails and pass are somewhat mixed. But, it looks like that the passes are fully have the right side of the plot.

SVM Linear

```
library(e1071)

#SVM
linearBARSVM <- svm(bar_passed ~ ., data = BARTrain, kernel = "linear", scale = TRUE)
summary(linearBARSVM)
```

```
##
## Call:
## svm(formula = bar_passed ~ ., data = BARTrain, kernel = "linear",
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  1387
##
##  ( 698 689 )
##
##
## Number of Classes:  2
##
## Levels:
##  FALSE TRUE
```

```
#Predictions and Accuracy
linearBARPredict <- predict(linearBARSVM, newdata = BARTest, type = "response")
mean(linearBARPredict == BARTest$bar_passed)
```

```
## [1] 0.9462174
```

```
linearTuned <- tune(svm, bar_passed~., data = BARVal, kernel = "linear", ranges = list(cost =
c(.01, .1, 1, 10)), gamma = c(.5,1,2,3,4))
summary(linearTuned$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = bar_passed ~ ., data = BARVal,
##     ranges = list(cost = c(0.01, 0.1, 1, 10)), kernel = "linear",
##     gamma = c(0.5, 1, 2, 3, 4))
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.01
##
## Number of Support Vectors:  482
##
##  ( 242 240 )
##
##
## Number of Classes:  2
##
## Levels:
##  FALSE TRUE
```

```
#Predictions and Accuracy
linearSVMTuned <- predict(linearTuned$best.model, newdata = BARTest)
mean(linearSVMTuned == BARTest$bar_passed)
```

```
## [1] 0.9462174
```

SVM Polynomial

```
#SVM
polyBARSVM <- svm(bar_passed ~ ., data = BARTrain, kernel = "polynomial", scale = TRUE)
summary(polyBARSVM)
```

```
##
## Call:
## svm(formula = bar_passed ~ ., data = BARTrain, kernel = "polynomial",
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  1401
##
##  ( 712 689 )
##
##
## Number of Classes:  2
##
## Levels:
##  FALSE TRUE
```

```
#Predictions and Accuracy
polyBARPredict <- predict(polyBARSVM, newdata = BARTest)
mean(polyBARPredict == BARTest$bar_passed)
```

```
## [1] 0.9462174
```

```
polyTuned <- tune(svm, bar_passed~., data = BARVal, kernel = "polynomial", ranges = list(cost
= c(.001, .01, .1, 1, 10)), gamma = c(.5,1,2,3,4))
summary(polyTuned$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = bar_passed ~ ., data = BARVal,
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10)), kernel = "polynomial",
##     gamma = c(0.5, 1, 2, 3, 4))
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  0.001
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  484
##
##  ( 244 240 )
##
##
## Number of Classes:  2
##
## Levels:
##  FALSE TRUE
```

```
#Predictions and Accuracy
polySVMTuned <- predict(polyTuned$best.model, newdata = BARTest)
mean(polySVMTuned == BARTest$bar_passed)
```

```
## [1] 0.9462174
```

SVM Radial

```
#SVM
radialBARSVM <- svm(bar_passed ~ ., data = BARTrain, kernel = "radial", scale = TRUE)
summary(radialBARSVM)
```

```
##
## Call:
## svm(formula = bar_passed ~ ., data = BARTrain, kernel = "radial",
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  1660
##
##  ( 975 685 )
##
##
## Number of Classes:  2
##
## Levels:
##  FALSE TRUE
```

```
#Predictions and Accuracy
radialBARPredict <- predict(radialBARSVM, newdata = BARTest)
mean(radialBARPredict == BARTest$bar_passed)
```

```
## [1] 0.9464405
```

```
radialTuned <- tune(svm, bar_passed~., data = BARVal, kernel = "radial", ranges = list(cost =
c(.001, .01, .1, 1, 10, 10)), gamma = c(.5,1,2,3,4))
summary(radialTuned$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = bar_passed ~ ., data = BARVal,
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10, 10)), kernel = "radial",
##     gamma = c(0.5, 1, 2, 3, 4))
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.001
##
## Number of Support Vectors:  481
##
##  ( 241 240 )
##
##
## Number of Classes:  2
##
## Levels:
##  FALSE TRUE
```

```
#Predictions and Accuracy
radialSVMTuned <- predict(radialTuned$best.model, newdata = BARTest)
mean(radialSVMTuned == BARTest$bar_passed)
```

```
## [1] 0.9462174
```

Conclusions

The best kernel seems to be radial by a tiny bit, most likely due to the fails seems to be more congregated on the lower values of ugpa and lsat scores. The accuracy seems to be identical for polynomial and linear, most likely due to the shape of the hyperplane being very similar to both of the kernels. All of their accuracy seem pretty good.