feature selection w R

Jason Grahn

3/12/2019

head(iris,3)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa

# Boruta

# Perform Boruta search  
boruta\_output <- Boruta(Species ~ .,   
 data=na.omit(iris),  
 doTrace=2)

## 1. run of importance source...

## 2. run of importance source...

## 3. run of importance source...

## 4. run of importance source...

## 5. run of importance source...

## 6. run of importance source...

## 7. run of importance source...

## 8. run of importance source...

## 9. run of importance source...

## After 9 iterations, +0.81 secs:

## confirmed 4 attributes: Petal.Length, Petal.Width, Sepal.Length, Sepal.Width;

## no more attributes left.

boruta\_output

## Boruta performed 9 iterations in 0.8133054 secs.  
## 4 attributes confirmed important: Petal.Length, Petal.Width,  
## Sepal.Length, Sepal.Width;  
## No attributes deemed unimportant.

boruta\_signif <- getSelectedAttributes(boruta\_output,   
 withTentative = TRUE)  
  
print(boruta\_signif)

## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"

roughFixMod <- TentativeRoughFix(boruta\_output)

## Warning in TentativeRoughFix(boruta\_output): There are no Tentative  
## attributes! Returning original object.

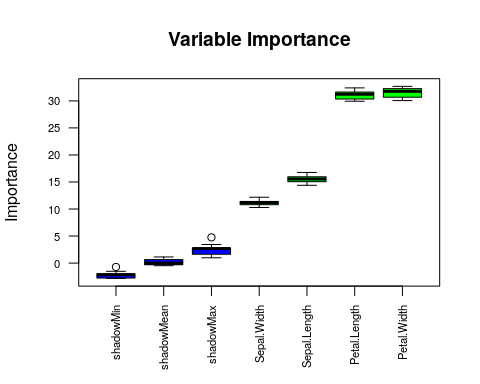
boruta\_signif <- getSelectedAttributes(roughFixMod)  
print(boruta\_signif)

## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"

# Variable Importance Scores  
imps <- attStats(roughFixMod)  
imps2 = imps[imps$decision != 'Rejected', c('meanImp', 'decision')]  
(imps2[order(-imps2$meanImp), ]) # descending sort

## meanImp decision  
## Petal.Width 31.56417 Confirmed  
## Petal.Length 31.11017 Confirmed  
## Sepal.Length 15.56325 Confirmed  
## Sepal.Width 11.14205 Confirmed

# Plot variable importance  
plot(boruta\_output,   
 cex.axis=.7,   
 las=2,   
 xlab="",   
 main="Variable Importance")



getConfirmedFormula(boruta\_output)

## Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width  
## <environment: 0x55f652918d78>

getNonRejectedFormula(boruta\_output)

## Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width  
## <environment: 0x55f6529d4a98>

Boruta feature selection is run using the Bortua package. Boruta processes feature selection by ranking the variables through random forest processing. The downside of working through random forest is that the selection process is a bit of a black-box. However, the advantage with Boruta is that it decides variable importance and selects those that are statistically significant; with allowance in determining what p-value the analyst considers significant. Boruta also processes what it considers “tentative” variables where it is unsure if a variable should be considered important or not. After running through multiple iterations of possible models, Boruta determines an “importance” score for each, and decides if they should be included in a model.

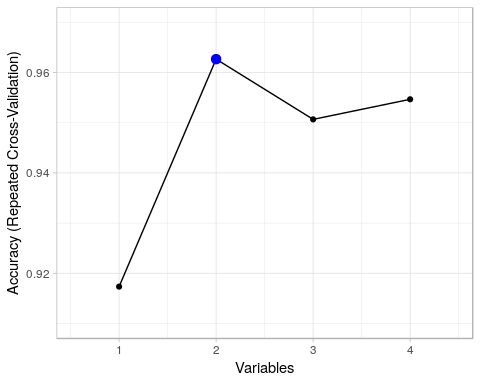
In this case, Boruta hasn’t eliminated any features, but it does state that the top *two* features are Petal.Width and Petal.Length. It did this by running 9 iterations on models in order to determine there are zero tentative features, that all were acceptiable to some degree, and the two most important were those mentioned above.

# Recursive Feature Elimination (RFE)

set.seed(100)  
options(warn=-1)  
  
subsets <- c(1:5, 10, 15, 18)  
  
ctrl <- rfeControl(functions = rfFuncs,  
 method = "repeatedcv",  
 repeats = 5,  
 verbose = FALSE)  
  
lmProfile <- rfe(x=iris[1:4], y=iris$Species,  
 sizes = subsets,  
 rfeControl = ctrl)  
  
lmProfile

##   
## Recursive feature selection  
##   
## Outer resampling method: Cross-Validated (10 fold, repeated 5 times)   
##   
## Resampling performance over subset size:  
##   
## Variables Accuracy Kappa AccuracySD KappaSD Selected  
## 1 0.9173 0.876 0.06809 0.10214   
## 2 0.9627 0.944 0.04697 0.07045 \*  
## 3 0.9507 0.926 0.05355 0.08033   
## 4 0.9547 0.932 0.05461 0.08192   
##   
## The top 2 variables (out of 2):  
## Petal.Length, Petal.Width

lmProfile %>%   
 ggplot(aes(x = Variables, y = Accuracy)) +  
 geom\_col() +  
 ylim(.91, .97) +  
 theme\_light()



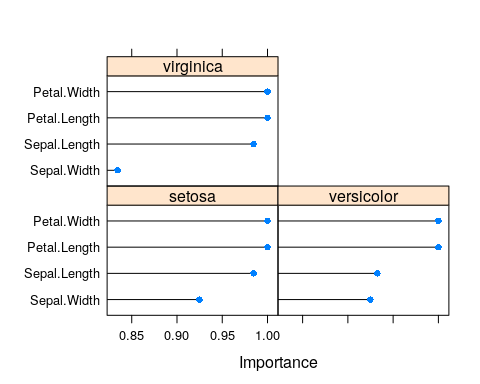
Recursive feature elimination is a computationally intensive process that cross-validates all combination of variables then repeats the pricess iteratively to find and eliminate insignificant variables. In this case, it’s settled on using two features as the most optimum model, and those two feature are Petal.Length and Petal.Width.

# Learning Vector Quantization

control <- trainControl(method="repeatedcv",   
 number=10,   
 repeats=3)  
  
# train the model  
model <- train(Species~.,   
 data=iris,   
 method="lvq",   
 preProcess="scale",   
 trControl=control)  
  
# estimate variable importance  
importance <- varImp(model,   
 scale=FALSE)  
  
# summarize importance  
print(importance)

## ROC curve variable importance  
##   
## variables are sorted by maximum importance across the classes  
## setosa versicolor virginica  
## Petal.Width 1.0000 1.0000 1.0000  
## Petal.Length 1.0000 1.0000 1.0000  
## Sepal.Length 0.9846 0.9326 0.9846  
## Sepal.Width 0.9248 0.9248 0.8344

# plot importance  
plot(importance)



With Learning Vector Quantization, we operate an artificial neural network. It uses the mlBench package. I dont know how it works. We can see from the graphical output that the top two variables for predicticting any of the three flower types are Petal.Length and Petal.Width. Sepal.Length is also third important, but isn’t necessary for modeling.

Sources: <https://www.machinelearningplus.com/machine-learning/feature-selection/> <https://machinelearningmastery.com/feature-selection-with-the-caret-r-package/>