Lipscomb University MSDS 5213: Naive Bayes

Olivia Samples

6/22/2020

4.1 Dataset Description

We will use the data set vote2.csv (in canvas). This data set has 16 predictor variables. The variable party is the target variable. Description about this data set can be found here http://archive.ics.uci.edu/ml/machine-learning-databases/voting-records/house-votes-84.names

4.2 Your tasks

- split the data into two sets (train and test)
- use the train dataset to train a Naive Bayes classifier (10 points)
- test the classifier with the test dataset (10 points)
- produce a confusion matrix (10 points)

Using RMarkDown, submit your work in .html or .pdf, including list of the commands you have used as well as the output generated with each command if any (20 points)

R Markdown

For this example we will use the e1071 package in R to illustrate the use of Naive Bayes classification.

Start by loading the class library

```
library(e1071)
library(caret)
```

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

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

Read the csv file:

```
votes <- read.csv(file='/Users/osamples/Documents/Lipscomb/MSDS 5213/Assignments/vote2.csv', head=TRUE,
```

Let's create 80-20 split of the data. 80% training data and 20% testing data.

```
split_size <- 0.8
train_size <- floor(nrow(votes) * split_size)
set.seed(1)
train_indices <- sample(1:nrow(votes), train_size)
```

Extract the class label from the testing data.

```
cl = votes[-train_indices,]$party
```

Extract the training data and the testing data. Then, train the Naive Bayes classifier

```
train = subset(votes[train_indices,])
test = subset(votes[-train_indices,],select =-party)
model <- naiveBayes(party ~.,data=train)
```

Test the model on the test data.

```
pred <- predict(model,test)
```

To see the confusion matrix, let us compare what is in pred with what is in cl.

```
confusionMatrix(pred, factor(cl))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   democrat republican
##   democrat         48          1
##   republican        3         35
##
##                Accuracy : 0.954
##                  95% CI : (0.8864, 0.9873)
##     No Information Rate : 0.5862
##     P-Value [Acc > NIR] : 3.914e-15
##
##                   Kappa : 0.906
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9412
##             Specificity : 0.9722
##          Pos Pred Value : 0.9796
##          Neg Pred Value : 0.9211
##              Prevalence : 0.5862
##          Detection Rate : 0.5517
##    Detection Prevalence : 0.5632
##       Balanced Accuracy : 0.9567
##
##        'Positive' Class : democrat
##
```

To see the probability distribution over the classes, we need to execute

```
pred <- predict(model,test,type="raw")
head(pred)
```

```
##          democrat   republican
## [1,] 1.000000e+00 2.198925e-08
## [2,] 9.999940e-01 5.953361e-06
## [3,] 4.389259e-07 9.999996e-01
## [4,] 9.362851e-11 1.000000e+00
## [5,] 8.611881e-13 1.000000e+00
## [6,] 1.000000e+00 4.581915e-20
```