Naive Bayes Classifier

Truong Thi An Hai

12/12/2020

*** Part I ***

In this assignment you will train a Naïve Bayes classifier on categorical data and predict individuals' incomes. Import the nbtrain.csv file. Use the first 9010 records as training data and the remaining 1000 records as testing data.

```
#Import nbtrain.csv
data = read.csv("C:/Users/hp/Desktop/nbtrain.csv")
#Split training data vs testing data
library(caret)
```

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

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

```
data_train <- head(data, 9010)
data_test <- tail(data, 1000)
```

a. Construct the Naive Bayes classifier from the training data, according to the formula "income ~ age + sex + educ". To do this, use the "naiveBayes" function from the "e1071" package. Provide the model's a priori and conditional probabilities.

```
library(e1071)
NBclassfier <- naiveBayes(as.factor(income) ~ age + sex + educ, data=data_train)
NBclassfier
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## [illegible]
##
## Conditional probabilities:
## [illegible]
```

A-priori probabilities:

- Income is in the range 10-50K: [illegible]
- Income is in the range 50-80K: [illegible]
- Income is in the range GT 80K: [illegible]

Conditional probabilities

Age

```
NBclassfier$tables$age
```

```
## [illegible]
```

Sex

```
NBclassfier$tables$sex
```

```
## [illegible]
```

Education

```
NBclassfier$tables$educ
```

```
## [illegible]
```

b. Score the model with the testing data and create the model's confusion matrix. Also, calculate the overall, 10-50K, 50-80K and GT 80K misclassification rates. Explain the variation in the model's predictive power across income classes.

```
testPred <- predict(NBclassfier, data_test, type="class")
message("Confusion Matrix for Test Data")
```

```
## Confusion Matrix for Test Data
```

```
Matrix <- confusionMatrix(testPred, as.factor(data_test$income))
Matrix
```

```
## Confusion Matrix and Statistics
##
## [illegible]
```

The overall misclassification rate: 1 - Accuracy = [illegible]

```
library([illegible])
```

```
## [illegible]
```

```
Misclassrate(testPred, as.factor(data_test$income))
```

```
## Classification table:
## [illegible]
## Misclassification errors (%):
## [illegible]
## Mean misclassification error: [illegible]
```

- The 10-50K misclassification rate: [illegible]
- The 50-80K misclassification rate: [illegible]
- The GT 80K misclassification rate: [illegible]

In this model variation is explained mostly by confusion matrix.

*** Part II ***

As in assignment I, import the nbtrain.csv file. Use the first 9010 records as training data and the remaining 1000 records as testing data.

```
#Import nbtrain.csv
data = read.csv("C:/Users/hp/Desktop/nbtrain.csv")
#Split training data vs testing data
data_train <- head(data, 9010)
data_test <- tail(data, 1000)
```

a. Construct the classifier according to the formula "sex ~ age + educ + income", and calculate the overall, female, and male misclassification rates. Explain the misclassification rates?

```
NBclassfier <- naiveBayes(as.factor(sex) ~ age + income + educ, data=data_train)
NBclassfier
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## [illegible]
##
## Conditional probabilities:
## [illegible]
```

```
testPred <- predict(NBclassfier, data_test, type="class")
Matrix <- confusionMatrix(testPred, as.factor(data_test$sex))
```

The overall misclassification rate: 1 - Accuracy = [illegible]

```
Misclassrate(testPred, as.factor(data_test$sex))
```

```
## Classification table:
## [illegible]
##    F 106  97
##    M 321 476
## Misclassification errors (%):
##    F    M
## 75.2 16.9
## Mean misclassification error: 46.1%
```

- The female misclassification rate: 75.2%
- The male misclassification rate: 16.9%

b. Divide the training data into two partitions, according to sex, and randomly select 3500 records from each partition. Reconstruct the model from part (a) from these 7000 records. Provide the model's a priori and conditional probabilities.

```
library(dplyr)

#Divide the training data into two partitions
data_female = subset(data_train, data_train$sex == 'F')
data_male = subset(data_train, data_train$sex=='M')
#Randomly select 3500 records from each partition
data_female = sample_n(data_female, 3500)
data_male = sample_n(data_male, 3500)
new_data = rbind(data_male,data_female)
model <- naiveBayes(as.factor(sex) ~ age + income + educ, data=new_data)
message("Model Navie Bayes Classifier")
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   F   M
## 0.5 0.5
##
## Conditional probabilities:
##    age
## Y       20-30     31-45     GT 45
##   F 0.1802857 0.3477143 0.4720000
##   M 0.1865714 0.3508571 0.4625714
##
##    income
## Y       10-50K     50-80K     GT 80K
##   F 0.88114286 0.08514286 0.03371429
##   M 0.74400000 0.15714286 0.09885714
##
##    educ
## Y      College     Others   Prof/Phd
##   F 0.32200000 0.65571429 0.02228571
##   M 0.28028571 0.68257143 0.03714286
```

The a priori probabilities are equal and the conditional probabilities are very similar.

c. How well does the model classify the testing data?

```
testPred <- predict(model, data_test, type="class")
Matrix <- confusionMatrix(testPred, as.factor(data_test$sex))
Matrix$table
```

```
##           Reference
## Prediction   F   M
##          F 369 412
##          M  58 161
```

```
message("Accuracy")
```

```
## Accuracy
```

```
Matrix$overall[1]
```

```
## Accuracy
##     0.53
```

d. Repeat step (b) 4 several times. What effect does the random selection of records have on the model's performance?

1.

```
#Divide the training data into two partitions
data_female = subset(data_train, data_train$sex == 'F')
data_male = subset(data_train, data_train$sex=='M')
#Randomly select 3500 records from each partition
data_female = sample_n(data_female, 3500)
data_male = sample_n(data_male, 3500)
new_data = rbind(data_male,data_female)
model <- naiveBayes(as.factor(sex) ~ age + income + educ, data=new_data)
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   F   M
## 0.5 0.5
##
## Conditional probabilities:
##    age
## Y       20-30     31-45     GT 45
##   F 0.1774286 0.3480000 0.4745714
##   M 0.1831429 0.3488571 0.4680000
##
##    income
## Y       10-50K     50-80K     GT 80K
##   F 0.88200000 0.08485714 0.03314286
##   M 0.73600000 0.16000000 0.10400000
##
##    educ
## Y      College     Others   Prof/Phd
##   F 0.31800000 0.66057143 0.02142857
##   M 0.28085714 0.68028571 0.03885714
```

```
testPred <- predict(model, data_test, type="class")
Matrix <- confusionMatrix(testPred, as.factor(data_test$sex))
message("Accuracy")
```

```
## Accuracy
```

```
Matrix$overall[1]
```

```
## Accuracy
##     0.53
```

2.

```
#Divide the training data into two partitions
data_female = subset(data_train, data_train$sex == 'F')
data_male = subset(data_train, data_train$sex=='M')
#Randomly select 3500 records from each partition
data_female = sample_n(data_female, 3500)
data_male = sample_n(data_male, 3500)
new_data = rbind(data_male,data_female)
model <- naiveBayes(as.factor(sex) ~ age + income + educ, data=new_data)
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   F   M
## 0.5 0.5
##
## Conditional probabilities:
##    age
## Y       20-30     31-45     GT 45
##   F 0.1817143 0.3442857 0.4740000
##   M 0.1837143 0.3517143 0.4645714
##
##    income
## Y       10-50K     50-80K     GT 80K
##   F 0.88742857 0.08057143 0.03200000
##   M 0.74257143 0.15800000 0.09942857
##
##    educ
## Y      College     Others   Prof/Phd
##   F 0.31771429 0.66000000 0.02228571
##   M 0.27685714 0.68400000 0.03914286
```

```
testPred <- predict(model, data_test, type="class")
Matrix <- confusionMatrix(testPred, as.factor(data_test$sex))
message("Accuracy")
```

```
## Accuracy
```

```
Matrix$overall[1]
```

```
## Accuracy
##     0.53
```

3.

```
#Divide the training data into two partitions
data_female = subset(data_train, data_train$sex == 'F')
data_male = subset(data_train, data_train$sex=='M')
#Randomly select 3500 records from each partition
data_female = sample_n(data_female, 3500)
data_male = sample_n(data_male, 3500)
new_data = rbind(data_male,data_female)
model <- naiveBayes(as.factor(sex) ~ age + income + educ, data=new_data)
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   F   M
## 0.5 0.5
##
## Conditional probabilities:
##    age
## Y       20-30     31-45     GT 45
##   F 0.1814286 0.3528571 0.4657143
##   M 0.1894286 0.3497143 0.4608571
##
##    income
## Y       10-50K     50-80K     GT 80K
##   F 0.88428571 0.08257143 0.03314286
##   M 0.74028571 0.16028571 0.09942857
##
##    educ
## Y      College     Others   Prof/Phd
##   F 0.32028571 0.65857143 0.02114286
##   M 0.28371429 0.67885714 0.03742857
```

```
testPred <- predict(model, data_test, type="class")
Matrix <- confusionMatrix(testPred, as.factor(data_test$sex))
message("Accuracy")
```

```
## Accuracy
```

```
Matrix$overall[1]
```

```
## Accuracy
##     0.53
```

4.

```
#Divide the training data into two partitions
data_female = subset(data_train, data_train$sex == 'F')
data_male = subset(data_train, data_train$sex=='M')
#Randomly select 3500 records from each partition
data_female = sample_n(data_female, 3500)
data_male = sample_n(data_male, 3500)
new_data = rbind(data_male,data_female)
model <- naiveBayes(as.factor(sex) ~ age + income + educ, data=new_data)
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##   F   M
## 0.5 0.5
##
## Conditional probabilities:
##    age
## Y       20-30     31-45     GT 45
##   F 0.1837143 0.3440000 0.4722857
##   M 0.1848571 0.3551429 0.4600000
##
##    income
## Y       10-50K     50-80K     GT 80K
##   F 0.88285714 0.08257143 0.03457143
##   M 0.74000000 0.15742857 0.10257143
##
##    educ
## Y      College     Others   Prof/Phd
##   F 0.32114286 0.65771429 0.02114286
##   M 0.27542857 0.68371429 0.04085714
```

```
testPred <- predict(model, data_test, type="class")
Matrix <- confusionMatrix(testPred, as.factor(data_test$sex))
message("Accuracy")
```

```
## Accuracy
```

```
Matrix$overall[1]
```

```
## Accuracy
##     0.53
```

Conditional probabilities are very close over the entire sample.