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)</pre> data_test <- tail(data, 1000)</pre>

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**

a. Construct the Naïve Bayes classifier from the training data, according to the

Naive Bayes Classifier for Discrete Predictors ## ## Call: ## naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities: 10-50K 50-80K GT 80K ## 0.80266371 0.12563818 0.07169811 ## ## Conditional probabilities: ## ## Y 20-30 31-45 GT 45 ## 10-50K 0.20796460 0.34457965 0.44745575 ## 50-80K 0.08303887 0.39752650 0.51943463 ## GT 80K 0.06811146 0.34055728 0.59133127 ## ## ## Y ## 10-50K 0.4798119 0.5201881 ## 50-80K 0.2871025 0.7128975 ## GT 80K 0.2058824 0.7941176 ## educ College Others Prof/Phd ## Y ## 10-50K 0.24585177 0.73976770 0.01438053 ## 50-80K 0.49558304 0.44257951 0.06183746 ## GT 80K 0.53869969 0.29566563 0.16563467 A-priori probabilities: • Income is in the range 10-50K: 0.803 • Income is in the range 50-80K: 0.126 • Income is in the range GT 80K: 0.072

Conditional probabilities Age NBclassfier\$tables\$age ## ## Y 20-30 31-45 ## 10-50K 0.20796460 0.34457965 0.44745575 ## 50-80K 0.08303887 0.39752650 0.51943463

GT 80K 0.06811146 0.34055728 0.59133127 NBclassfier\$tables\$sex

Sex ## ## 10-50K 0.4798119 0.5201881 ## 50-80K 0.2871025 0.7128975 ## GT 80K 0.2058824 0.7941176 Education NBclassfier\$tables\$educ ## educ ## Y College Others Prof/Phd ## 10-50K 0.24585177 0.73976770 0.01438053

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.

50-80K 0.49558304 0.44257951 0.06183746 ## GT 80K 0.53869969 0.29566563 0.16563467

95% CI : (0.7686, 0.8196)

Kappa: 0.0709

No Information Rate : 0.793

P-Value [Acc > NIR] : 0.4564

Mcnemar's Test P-Value : <2e-16

 The 50-80K misclassification rate: 100% The GT 80K misclassification rate: 89.3%

*** Part II ***

#Import nbtrain.csv

In this model variation is explaeined mostly by confusion matrix

data = read.csv("C:/Users/hp/Desktop/nbtrain.csv")

#Split training data vs testing data

Y 20-30 31-45 GT 45 ## F 0.1802444 0.3475051 0.4722505 ## M 0.1837859 0.3536009 0.4626131

Y 10-50K 50-80K GT 80K ## F 0.88340122 0.08273931 0.03385947 ## M 0.74025974 0.15879575 0.10094451

Y College Others Prof/Phd ## F 0.32128310 0.65707739 0.02163951 M 0.28040142 0.68103109 0.03856749

income

library(dplyr)

model

##

Call:

Matrix\$table

##

Prediction F M

message("Accuracy")

Matrix\$overall[1]

Accuracy

#Divide the training data into two partitions

data_female = sample_n(data_female, 3500) data_male = sample_n(data_male, 3500) new_data = rbind(data_male,data_female)

message("Model Navie Bayes Classifier")

Naive Bayes Classifier for Discrete Predictors

naiveBayes.default(x = X, y = Y, laplace = laplace)

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

Matrix <- confusionMatrix(testPred, as.factor(data_test\$sex))</pre>

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

testPred <- predict(model, data_test, type="class")</pre>

Reference

F 369 412

M 58 161

M 0.73600000 0.16000000 0.10400000

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")</pre>

#Divide the training data into two partitions

Naive Bayes Classifier for Discrete Predictors

naiveBayes.default(x = X, y = Y, laplace = laplace)

GT 80K

##

Call:

Y

##

##

##

##

F M ## 0.5 0.5

income

educ

A-priori probabilities:

Conditional probabilities:

Y 10-50K 50-80K

Y 20-30 31-45 GT 45 ## F 0.1814286 0.3528571 0.4657143 ## M 0.1894286 0.3497143 0.4608571

F 0.88428571 0.08257143 0.03314286 ## M 0.74028571 0.16028571 0.09942857

data_female = sample_n(data_female, 3500) data_male = sample_n(data_male, 3500) new_data = rbind(data_male, data_female)

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

model <- naiveBayes(as.factor(sex) ~ age + income + educ, data=new_data)</pre>

Matrix <- confusionMatrix(testPred, as.factor(data_test\$sex))</pre>

##

##

educ

message("Accuracy")

Matrix\$overall[1]

0.53

Accuracy

Accuracy

2.

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

model <- naiveBayes(as.factor(sex) ~ age + income + educ, data=new_data)</pre>

##

data_train <- head(data, 9010)</pre> data_test <- tail(data, 1000)</pre>

data and the remaining 1000 records as testing data.

Statistics by Class:

##

##

##

##

##

##

##

testPred <- predict(NBclassfier, data_test, type="class")</pre> message("Confusion Matrix for Test Data") ## Confusion Matrix for Test Data Matrix <- confusionMatrix(testPred, as.factor(data_test\$income))</pre> Matrix ## Confusion Matrix and Statistics ## Reference ## Prediction 10-50K 50-80K GT 80K ## 10-50K 787 127 67 50-80K 0 0 0 GT 80K 6 5 8 ## ## ## Overall Statistics ## ## Accuracy: 0.795

##
Class: 10-50K Class: 50-80K Class: GT 80K
Sensitivity 0.9924 0.000 0.1067
Specificity 0.0628 1.000 0.9881
Pos Pred Value 0.8022 NaN 0.4211
Neg Pred Value 0.6842 0.868 0.9317
Prevalence 0.7930 0.132 0.0750
Detection Rate 0.7870 0.000 0.0080
Detection Prevalence 0.9810 0.000 0.0190
Balanced Accuracy 0.5276 0.500 0.5474 The overall misclassification rate: 1 - Accuracy = 0.205 library(shipunov) ## package 'shipunov', version 1.12 Misclass(testPred, as.factor(data_test\$income)) ## Classification table: ## pred 10-50K 50-80K GT 80K ## 10-50K 787 127 67 ## 50-80K 0 0 0 ## GT 80K 6 5 8 ## Misclassification errors (%): ## 10-50K 50-80K GT 80K 0.8 100.0 89.3 ## Mean misclassification error: 63.4% • The 10-50K misclassification rate: 0.8%

calculate the overall, female, and male misclassification rates. Explain the misclassification rates? NBclassfier <- naiveBayes(as.factor(sex) ~ age + income + educ, data=data_train) ## Naive Bayes Classifier for Discrete Predictors ## naiveBayes.default(x = X, y = Y, laplace = laplace) ## A-priori probabilities: ## 0.43596 0.56404 ## Conditional probabilities:

a. Construct the classifier according to the formula "sex ~ age + educ + income", and

As in assignment I, import the nbtrain.csv file. Use the first 9010 records as training

testPred <- predict(NBclassfier, data_test, type="class")</pre> Matrix <- confusionMatrix(testPred, as.factor(data_test\$sex))</pre> The overall misclassification rate: 1 - Accuracy = 0.418 Misclass(testPred, as.factor(data_test\$sex)) ## Classification table: obs ## pred F M ## F 106 97 ## M 321 476 ## Misclassification errors (%): ## 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.

A-priori probabilities: ## F M ## 0.5 0.5 ## ## Conditional probabilities: 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

Accuracy 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)</pre> ## 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: ## Y 20-30 31-45 ## 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

Naive Bayes Classifier for Discrete Predictors ## naiveBayes.default(x = X, y = Y, laplace = laplace) ## A-priori probabilities: ## F M ## 0.5 0.5 ## Conditional probabilities: ## ## Y GT 45 20-30 31-45 ## F 0.1817143 0.3442857 0.4740000 ## M 0.1837143 0.3517143 0.4645714 ## ## income 10-50K 50-80K ## Y 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")</pre> Matrix <- confusionMatrix(testPred, as.factor(data_test\$sex))</pre> 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)</pre> model

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")</pre> Matrix <- confusionMatrix(testPred, as.factor(data_test\$sex))</pre> 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)</pre> 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:

20-30 31-45 ## F 0.1837143 0.3440000 0.4722857 M 0.1848571 0.3551429 0.4600000 ## ## income 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")</pre> Matrix <- confusionMatrix(testPred, as.factor(data_test\$sex))</pre> message("Accuracy") ## Accuracy Matrix\$overall[1] ## Accuracy 0.53 Conditional probabilities are very close over the entire sample.