# Hannah Roach

# 5-12-2019

# Lab Exercise 8: Naïve Bayesian Classifier

|  |  |
| --- | --- |
| **Purpose:** | This lab is designed to investigate and practice the Naïve Bayesian Classifier analytic technique. After completing the tasks in this lab you should be able to:   * Use R functions for Naïve Bayesian Classification * Apply the requirements for generating appropriate training data * Validate the effectiveness of the Naïve Bayesian Classifier with the big data |
|  | |
| **Tasks:** | Tasks you will complete in this lab include:   * Use R –Studio environment to code the Naïve Bayesian Classifier * Use the ODBC connection to the “census” database to create a training data set for Naïve Bayesian Classifier from the big data * Use the Naïve Bayesian Classifier program and evaluate how well it predicts the results using the training data and then compare the results with original data |
|  | |

Part 1 – Building Naïve Bayesian Classifier

### Workflow Overview

### LAB Instructions

| **Step** | **Action** |
| --- | --- |
| 1 | Log in with GPADMIN credentials on to R-Studio. |
| 2 | **Set working directory and review training and test data**  1. Set the working directory using the following command:  **> setwd("~/LAB08")**   * The **“sample1.csv”** file in this directory represents the data worked with in the instructor led training session. The file has a header row, followed by 14 rows of training data. * The **testing data** on which you will predict the results should be appended after the **training data**. The data set should read:   **Age,Income,Jobstaisfaction,Desire,Enrolls 🡨-------Header**  **<=30,High,No,Fair,No**  **<=30,High,No,Excellent,No**  **31 to 40,High,No,Fair,Yes**  **>40,Medium,No,Fair,Yes**  **>40,Low,Yes,Fair,Yes**  **>40,Low,Yes,Excellent,No**  **31 to 40,Low,Yes,Excellent,Yes**  **<=30,Medium,No,Fair,No**  **<=30,Low,Yes,Fair,Yes**  **>40,Medium,Yes,Fair,Yes**  **<=30,Medium,Yes,Excellent,Yes**  **31 to 40,Medium,No,Excellent,Yes**  **31 to 40,High,Yes,Fair,Yes**  **>40,Medium,No,Excellent,No**  **<=30,Medium,Yes,Fair, 🡨---------testing data** |
| 3 | **Install and load library “e1071”**  Execute the following command to install the required packages and load the libraries:  **> install.packages("e1071")**  **> library("e1071")** |
| 4 | **Read in and review data**   1. Execute the following to read in the data.     **> # read the data into a table from the file**  **> sample <- read.table("sample1.csv",header=TRUE,sep=",")**  **> # we will now define the data frames to use the NB classifier**  **> # we will now define the data frames to use the NB classifier**  **> traindata <- as.data.frame(sample[1:14,])**  **> testdata <- as.data.frame(sample[15,])**  You now have two data frame objects “**traindata**” and “**testdata**” for running the NB Classifier.   1. Execute the following command to display the data frames, to ensure they are loaded properly.   **> #Display data frames**  **> traindata**  **> testdata**   1. Screenshot the output for traindata and test data |
| 5 | **Build the Naïve Bayesian classifier Model from First Principles:**   1. The first step in building the model is the computation of prior probabilities. The independent variables here are the “Age”, “Income”, “Jobsatisfaction” and “Desire”. The dependent variable is “Enrolls”   Compute the prior probabilities of enrollment, P(no), P(yes) first, the counts :  **> tprior <- table(traindata$Enrolls)**  then, normalize over the total number of instances to get the probabilities  **> tprior <- tprior/sum(tprior)**  **> tprior**  Screenshots the results of prior probabilities     1. Compute the summaries that you need to create a Bayes model: P(A|b), b={no, yes}   First, count up "no" and "yes" by Age:  **> ageCounts <-table(traindata[,c("Enrolls", "Age")])**  3. Then, normalize by the total number of "no" and "yes" each to get the conditional probabilities  **> ageCounts <- ageCounts/rowSums(ageCounts)**  Display the results on the console and review the conditional probabilities  **> ageCounts**  Screenshot the results     1. Do the same for the other variables.   **> incomeCounts <- table(traindata[,c("Enrolls", "Income")])**  **> incomeCounts <- incomeCounts/rowSums(incomeCounts)**  **>incomeCounts**  **Screenshot**    **> jsCounts <- table(traindata[,c("Enrolls", "Jobsatisfaction")])**  **> jsCounts<-jsCounts/rowSums(jsCounts)**  **>jsCounts**  **Screenshot**    **> desireCounts <- table(traindata[,c("Enrolls", "Desire")])**  **> desireCounts <- desireCounts/rowSums(desireCounts)**  >desireCounts  Screenshot |
| 6 | **Predict the Results:**   1. Use the Naïve Bayesian Classifier formula to compute product of P(A|b), for b={no, yes}. The maximum of the two is the “predicted” result of the dependent variable. In the test data we need to predict the “Enrolls” given the for Age<=30, Income = Medium, Jobsatisfaction = yes and Desire = Fair   **> pyes <-**  **ageCounts["Yes","<=30"]\***  **incomeCounts["Yes","Medium"]\***  **jsCounts["Yes","Yes"]\***  **desireCounts["Yes","Fair"]\***  **tprior["Yes"]**  followed by  **> pno <-**  **ageCounts["No","<=30"]\***  **incomeCounts["No","Medium"]\***  **jsCounts["No","Yes"]\***  **desireCounts["No","Fair"]\***  **tprior["No"]**   1. The prediction will be max(pyes,pno).   **> pyes**  **> pno**  **> max(pyes,pno)**   1. What is the predicted result for “Enrolls” for someone’s age less than 30, income medium ,JobSatisfaction yes, and desidre Fiar? |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | **Execute the Naïve Bayesian Classifier with e1071 package:**  The Naïve Bayes function computes the conditional a-posterior probabilities of a categorical class variable given independent categorical predictor variables using the Bayes rule. The usage takes the form of naiveBayes(formula, data,…) where the arguments are defined as follows:   |  |  | | --- | --- | | * **formula** | A formula of the form class ~ x1 + x2 + .... Interactions are not allowed. | | * **data** | Either a data frame of factors or a contingency table. |  * You are modeling for attribute “ Enrolls”.  1. Use the following commands to execute the model and display the results.   **> # use the NB classifier**  **> model <- naiveBayes(Enrolls ~.,traindata)**  **> # display model**  **> model**   1. SCREENSHOT THE RESULTS and compare these results to the **apriori probabilities** you manually computed earlier in step 5. Are they the same or different ?   They are different |
| 8 | **Predict the Outcome of “Enrolls” with the Testdata:**   1. To use the predict function, type in the following:   **> # predict with testdata**  **> results <- predict (model,testdata)**  **> # display results**  **> results**   1. Review the results (Prediction for “Enrolls”) on the console. What is the prediction Yes or No? |
| 9 | **Review results**   1. Look at P(age=31-40 | Enrolls = No). You will observe a zero probability. Is this a problem?   No, because the probability is not an absolute zero. The probability is just very small.   1. Build another NB model, with Laplace smoothing model2 = naiveBayes(Enrolls ~.,traindata, laplace=0.01)      1. Compare the probabilities here with those of the first model   Note down your observations in the space provided below:  The probabilities are very similar. The probability between 31 and 40 for the second model is slightly higher than the first model. In the second model there probability is measured at higher levels of precision with five numbers trailing the decimal point for age. |
|  |  |

*End of Lab Exercise*