Mderangu

2023-11-05

##Question-A  
  
##Create a pivot table for the training data with Online as a column variable, CC as a row variable, and Loan as a secondary row variable. The values inside the table should convey the count. In R use functions melt() and cast(), or function table()  
  
bank <- mlba::UniversalBank  
# Load necessary libraries  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.2.3

# Assuming you have already loaded and prepared your data  
  
# Split the data into training (60%) and validation (40%) sets  
set.seed(1)  
train.index <- sample(1:nrow(bank), 0.6 \* nrow(bank))  
train <- bank[train.index, ]  
  
# Create a pivot table using dplyr and tidyr  
pivot\_table <- train %>%  
 group\_by(CreditCard, Personal.Loan, Online) %>%  
 summarise(Count = n()) %>%  
 pivot\_wider(names\_from = Online, values\_from = Count, values\_fill = 0)

## `summarise()` has grouped output by 'CreditCard', 'Personal.Loan'. You can  
## override using the `.groups` argument.

# Print the pivot table  
print(pivot\_table)

## # A tibble: 4 × 4  
## # Groups: CreditCard, Personal.Loan [4]  
## CreditCard Personal.Loan `0` `1`  
## <int> <int> <int> <int>  
## 1 0 0 805 1119  
## 2 0 1 79 119  
## 3 1 0 332 469  
## 4 1 1 30 47

##Question-B  
  
##Consider the task of classifying a customer who owns a bank credit card and is actively using online banking services. Looking at the pivot table, what is the probability that this customer will accept the loan offer? [This is the probability of loan acceptance (Loan = 1) conditional on having a bank credit card (CC = 1) and being an active user of online banking services (Online = 1)].  
  
## Answer: The probability of being approved for a loan, considering that someone possesses a bank credit card and uses online services, is 2.6%, which is equivalent to 77 out of 3,000 cases  
  
##Question-C  
  
##Create two separate pivot tables for the training data. One will have Loan (rows) as a function of Online (columns) and the other will have Loan (rows) as a function of CC.  
  
# Load necessary libraries  
library(dplyr)  
library(tidyr)  
  
# Melt and pivot the data for Personal.Loan  
Loanline <- train %>%  
 select(Personal.Loan, Online) %>%  
 group\_by(Personal.Loan, Online) %>%  
 summarise(count = n()) %>%  
 spread(key = Online, value = count, fill = 0)

## `summarise()` has grouped output by 'Personal.Loan'. You can override using the  
## `.groups` argument.

# Melt and pivot the data for CreditCard  
LoanCC <- train %>%  
 select(CreditCard, Online) %>%  
 group\_by(CreditCard, Online) %>%  
 summarise(count = n()) %>%  
 spread(key = Online, value = count, fill = 0)

## `summarise()` has grouped output by 'CreditCard'. You can override using the  
## `.groups` argument.

# Print the results  
print(Loanline)

## # A tibble: 2 × 3  
## # Groups: Personal.Loan [2]  
## Personal.Loan `0` `1`  
## <int> <dbl> <dbl>  
## 1 0 1137 1588  
## 2 1 109 166

print(LoanCC)

## # A tibble: 2 × 3  
## # Groups: CreditCard [2]  
## CreditCard `0` `1`  
## <int> <dbl> <dbl>  
## 1 0 884 1238  
## 2 1 362 516

##Question-D  
  
##Compute the following quantities [P(A | B) means “the probability ofA given B”]:  
##i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors)  
##ii. P(Online = 1 | Loan = 1)  
##iii. P(Loan = 1) (the proportion of loan acceptors)  
##iv. P(CC = 1 | Loan = 0)  
##v. P(Online = 1 | Loan = 0)  
##vi. P(Loan = 0)  
  
# Load necessary libraries  
library(e1071)

## Warning: package 'e1071' was built under R version 4.2.3

# Create contingency tables  
table1 <- table(train$Online, train$Personal.Loan)  
table2 <- table(train$CreditCard, train$Personal.Loan)  
table3 <- table(train$Personal.Loan)  
  
# Print the contingency tables  
print(table1)

##   
## 0 1  
## 0 1137 109  
## 1 1588 166

print(table2)

##   
## 0 1  
## 0 1924 198  
## 1 801 77

print(table3)

##   
## 0 1   
## 2725 275

##Question-E  
  
##Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC = 1, Online = 1).  
  
((77/(77+198))\*(166/(166+109))\*(275/(275+2725)))/(((77/(77+198))\*(166/(166+109))\*(275/(275+2725)))+((801/(801+1924))\*(1588/(1588+1137))\*2725/(2725+275)))

## [1] 0.09055758

##So, in simple terms, the entire expression calculates the conditional probability of having a bank credit card and using online services while also getting a loan approved, considering the overall probability of getting a loan approved and the probability of having these traits for both approval and non-approval cases. The result is a numerical probability value, which in this case is 2.6%.  
  
##Question-F  
  
##Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate?  
  
##Answer: The "exact method" requires that the independent variables (factors that help make a prediction) must match precisely to make a prediction. For example, if we want to predict something, we need all the exact same conditions, like having a specific income, age, and other factors.  
  
##On the other hand, the "naive Bayes method" is more flexible. It can make predictions even if we don't have all the exact conditions or if the conditions are not a perfect match. It uses probabilities and statistics to estimate the likelihood of an event happening based on the information it has, even if the conditions are not a perfect match.  
  
##So, the key difference is that the "exact method" requires a perfect match of conditions, while "naive Bayes" is more forgiving and can make predictions even when conditions are not exact.  
  
  
##Question-G  
  
##Which of the entries in this table are needed for computing P(Loan = 1 | CC = 1, Online = 1)?Run naive Bayes on the data. Examine the model output on training data, and find the entry that corresponds to P(Loan = 1 | CC = 1, Online = 1). Compare this to the number you obtained in (E).  
  
naive.train = train[,c(10,13:14)]  
naivebayes = naiveBayes(Personal.Loan~.,data=naive.train)  
naivebayes

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.90833333 0.09166667   
##   
## Conditional probabilities:  
## Online  
## Y [,1] [,2]  
## 0 0.5827523 0.4931950  
## 1 0.6036364 0.4900334  
##   
## CreditCard  
## Y [,1] [,2]  
## 0 0.293945 0.4556506  
## 1 0.280000 0.4498175

##The result obtained using the Naive Bayes model is consistent with the count we derived from the pivot table method. The probability of having a loan (Loan = 1) given that a person has a credit card (CC = 1) and uses online services (Online = 1) is calculated as (0.280)(0.603)(0.090) / (0.280 \* 0.603 \* 0.090 + 0.290 \* 0.582 \* 0.908), and it equals 0.09. This result aligns with the previously provided response.