Untitled

Lona Uprety

2025-10-13

```r
library(readr)
bank <- read_csv("C:/Users/lona2/Downloads/UniversalBank (1).csv")
```

```
## Rows: 5000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): ID, Age, Experience, Income, ZIP Code, Family, CCAvg, Education, M...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
View(bank)
library(caret)
```

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

```r
library(e1071)
```

```
##
## Attaching package: 'e1071'
##
## The following object is masked from 'package:ggplot2':
##
##     element
```

```r
library(dplyr)
```

```
##
## 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
```

```r
library(reshape2)
```

```r
#Keep only the relevant columns
bank1 <- bank %>% select(Online, CreditCard, `Personal Loan`)
```

```r
#Rename columns
colnames(bank1) <- c("Online", "CC", "Loan")
```

```r
#Partition data
set.seed(123)
train_index <- createDataPartition(bank1$Loan, p = 0.6, list = FALSE)
train <- bank1[train_index, ]
valid <- bank1[-train_index, ]
```

```r
#A) Pivot table
pivot_table <- as.data.frame(table(train$CC, train$Loan, train$Online))
colnames(pivot_table) <- c("CC", "Loan", "Online", "Count")
pivot_table_cast <- dcast(pivot_table, CC + Loan ~ Online, value.var = "Count")
pivot_table_cast
```

```
##   CC Loan   0    1
## 1  0    0 785 1145
## 2  0    1  65  122
## 3  1    0 317  475
## 4  1    1  34   57
```

```r
#B) Conditional probability
cond_table <- table(train$Loan, train$CC, train$Online)
p_L1_CC1_O1 <- cond_table["1", "1", "1"] / sum(cond_table[, "1", "1"])
p_L1_CC1_O1
```

```
## [1] 0.1071429
```

```r
#The probability that a customer who owns a bank credit card and actively uses online banking will acce
```

```r
#C) Separate pivot table
pivot_online <- table(train$Loan, train$Online)
pivot_online
```

```
##
##        0    1
##   0 1102 1620
##   1   99  179
```

```r
pivot_cc <- table(train$Loan, train$CC)
pivot_cc
```

```
##
##        0    1
##   0 1930  792
##   1  187   91
```

```r
#D) Compute probabilities
# 1. P(CC = 1 | Loan = 1)
P_CC1_L1 <- sum(train$CC == 1 & train$Loan == 1) / sum(train$Loan == 1)

# 2. P(Online = 1 | Loan = 1)
P_Online1_L1 <- sum(train$Online == 1 & train$Loan == 1) / sum(train$Loan == 1)

# 3. P(Loan = 1)
P_L1 <- mean(train$Loan == 1)

# 4. P(CC = 1 | Loan = 0)
P_CC1_L0 <- sum(train$CC == 1 & train$Loan == 0) / sum(train$Loan == 0)

# 5. P(Online = 1 | Loan = 0)
P_Online1_L0 <- sum(train$Online == 1 & train$Loan == 0) / sum(train$Loan == 0)

# 6. P(Loan = 0)
P_L0 <- mean(train$Loan == 0)

# Display all
data.frame(
  Metric = c("P(CC=1|Loan=1)", "P(Online=1|Loan=1)", "P(Loan=1)",
             "P(CC=1|Loan=0)", "P(Online=1|Loan=0)", "P(Loan=0)"),
  Probability = c(P_CC1_L1, P_Online1_L1, P_L1, P_CC1_L0, P_Online1_L0, P_L0)
)
```

```
##                Metric Probability
## 1     P(CC=1|Loan=1)  0.32733813
## 2 P(Online=1|Loan=1)  0.64388489
## 3          P(Loan=1)  0.09266667
## 4     P(CC=1|Loan=0)  0.29096253
## 5 P(Online=1|Loan=0)  0.59515062
## 6          P(Loan=0)  0.90733333
```

```r
#E) Naive bayes probability
num <- P_L1 * P_CC1_L1 * P_Online1_L1
den <- num + (P_L0 * P_CC1_L0 * P_Online1_L0)
P_L1_given_CC1_O1 <- num / den
P_L1_given_CC1_O1
```

```
## [1] 0.1105637
```

```r
#A customer who owns a bank credit card and actively uses online banking services has about an 11.1% ch
```

```r
#F) Compare results
comparison <- data.frame(
  Source = c("Pivot Table", "Naive Bayes Formula"),
  Probability = c(p_L1_CC1_O1, P_L1_given_CC1_O1)
)
comparison
```

```
##                 Source Probability
```

```
## 1         Pivot Table   0.1071429
## 2 Naive Bayes Formula   0.1105637
```

```r
#While both estimates are similar, the Naive Bayes result is typically considered more reliable because
```

```r
#G) Run naive bayes model
train$Online <- as.factor(train$Online)
train$CC <- as.factor(train$CC)
train$Loan <- as.factor(train$Loan)

model_nb <- naiveBayes(Loan ~ Online + CC, data = train)
model_nb
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90733333 0.09266667
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4048494 0.5951506
##   1 0.3561151 0.6438849
##
##    CC
## Y           0         1
##   0 0.7090375 0.2909625
##   1 0.6726619 0.3273381
```

```r
# Predict probability for CC=1 and Online=1
new_customer <- data.frame(Online = factor(1, levels = c(0,1)),
                           CC = factor(1, levels = c(0,1)))
predict(model_nb, new_customer, type = "raw")
```

```
##              0         1
## [1,] 0.8894363 0.1105637
```

```r
#The probability computed manually using the Naive Bayes formula is identical to the probability produc
```