# Binary Classification with a Bank Churn Dataset

Humphrey Afobhokhan

2024-03-12

## Introduction

This project delves into binary classification within the banking sector, focusing on customer churn prediction based on Kaggle's Bank Churn Dataset. It aims to discern patterns that influence customers' decisions to stay with or leave their bank. Utilizing statistical analysis and machine learning in R, this study addresses data preprocessing, explores key factors affecting churn, and applies predictive modeling to forecast customer behavior. Through this analysis, we seek to uncover insights that could help banks enhance customer retention strategies.

```
#import and read file
getwd()
```

```
## [1] "/Users/badboihy/Downloads/Visualizing & Analyzing Data with R - Methods & Tools/projects/Predict
```

```
setwd("/Users/badboihy/Downloads/Visualizing & Analyzing Data with R - Methods & Tools/projects/Predict
getwd()
```

```
## [1] "/Users/badboihy/Downloads/Visualizing & Analyzing Data with R - Methods & Tools/projects/Predict
```

```
df.train <- read.csv('BankChurnDataset-2.csv')
head(df.train)
```

```
##   id CustomerId        Surname CreditScore Geography Gender Age Tenure  Balance
## 1  0   15674932 Okwudilichukwu         668    France   Male  33      3      0.0
## 2  1   15749177  Okwudiliolisa         627    France   Male  33      1      0.0
## 3  2   15694510          Hsueh         678    France   Male  NA     10      0.0
## 4  3   15741417            Kao         581    France   Male  34      2 148882.5
## 5  4   15766172      Chiemenam         716     Spain   Male  33      5      0.0
## 6  5   15771669       Genovese         588   Germany   Male  36      4 131778.6
##   NumOfProducts HasCrCard IsActiveMember EstimatedSalary Exited
## 1             2         1              0       181449.97      0
## 2             2         1              1        49503.50      0
## 3             2         1              0       184866.69      0
## 4             1         1              1              NA      0
## 5             2         1              1        15068.83      0
## 6             1         1              0       136024.31      1
```

```r
str(df.train)
```

```
## 'data.frame':    165034 obs. of  14 variables:
##  $ id             : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ CustomerId     : int  15674932 15749177 15694510 15741417 15766172 15771669 15692819 15669611 1569
##  $ Surname        : chr  "Okwudilichukwu" "Okwudiliolisa" "Hsueh" "Kao" ...
##  $ CreditScore    : int  668 627 678 581 716 588 593 678 676 583 ...
##  $ Geography      : chr  "France" "France" "France" "France" ...
##  $ Gender         : chr  "Male" "Male" "Male" "Male" ...
##  $ Age            : num  33 33 NA 34 33 36 30 37 43 40 ...
##  $ Tenure         : int  3 1 10 2 5 4 8 1 4 4 ...
##  $ Balance        : num  0 0 0 148883 0 ...
##  $ NumOfProducts  : int  2 2 2 1 2 1 1 1 2 1 ...
##  $ HasCrCard      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ IsActiveMember : int  0 1 0 1 1 0 0 0 0 1 ...
##  $ EstimatedSalary: num  181450 49504 184867 NA 15069 ...
##  $ Exited         : int  0 0 0 0 0 1 0 0 0 0 ...
```

```r
summary(df.train)
```

```
##        id           CustomerId          Surname           CreditScore   
##  Min.   :     0   Min.   :15565701   Length:165034      Min.   :350.0  
##  1st Qu.: 41258   1st Qu.:15633141   Class :character   1st Qu.:597.0  
##  Median : 82516   Median :15690169   Mode  :character   Median :659.0  
##  Mean   : 82516   Mean   :15692005                      Mean   :656.5  
##  3rd Qu.:123775   3rd Qu.:15756824                      3rd Qu.:710.0  
##  Max.   :165033   Max.   :15815690                      Max.   :850.0  
##                                                                         
##   Geography            Gender               Age            Tenure     
##  Length:165034      Length:165034      Min.   :18.00   Min.   : 0.00  
##  Class :character   Class :character   1st Qu.:32.00   1st Qu.: 3.00  
##  Mode  :character   Mode  :character   Median :37.00   Median : 5.00  
##                                        Mean   :38.13   Mean   : 5.02  
##                                        3rd Qu.:42.00   3rd Qu.: 7.00  
##                                        Max.   :92.00   Max.   :10.00  
##                                        NA's   :6                      
##     Balance       NumOfProducts     HasCrCard     IsActiveMember  
##  Min.   :     0   Min.   :1.000   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:     0   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000  
##  Median :     0   Median :2.000   Median :1.000   Median :0.0000  
##  Mean   : 55478   Mean   :1.554   Mean   :0.754   Mean   :0.4978  
##  3rd Qu.:119940   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:1.0000  
##  Max.   :250898   Max.   :4.000   Max.   :1.000   Max.   :1.0000  
##                                                                   
##  EstimatedSalary         Exited      
##  Min.   :    11.58   Min.   :0.0000  
##  1st Qu.: 74637.57   1st Qu.:0.0000  
##  Median :117948.00   Median :0.0000  
##  Mean   :112575.32   Mean   :0.2116  
##  3rd Qu.:155155.25   3rd Qu.:0.0000  
##  Max.   :199992.48   Max.   :1.0000  
##  NA's   :3
```

```r
summary(df.train$EstimatedSalary)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##     11.58  74637.57 117948.00 112575.32 155155.25 199992.48         3
```

```r
# Handling missing Estimated Salary missing Estimated Salary values
df.train$EstimatedSalary[is.na(df.train$EstimatedSalary)] <-
  median(df.train$EstimatedSalary, na.rm = TRUE)

# Checking out Estimated Salary
summary(df.train$EstimatedSalary)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     11.58  74637.60 117948.00 112575.42 155152.47 199992.48
```

```r
# Age Imputation, handling missing values in age

  impute_age <- function(age,class){
    out <- age
    for (i in 1:length(age)){

      if (is.na(age[i])){

        if (class[i] == 1){
          out[i] <- 42

        }else if (class[i] == 2){
          out[i] <- 37

        }else{
          out[i] <- 32
        }
      }else{
        out[i]<-age[i]
      }
    }
    return(out)
  }

fixed.ages <- impute_age(df.train$Age, df.train$HasCrCard)
df.train$Age <- fixed.ages

summary(df.train$Age)
```

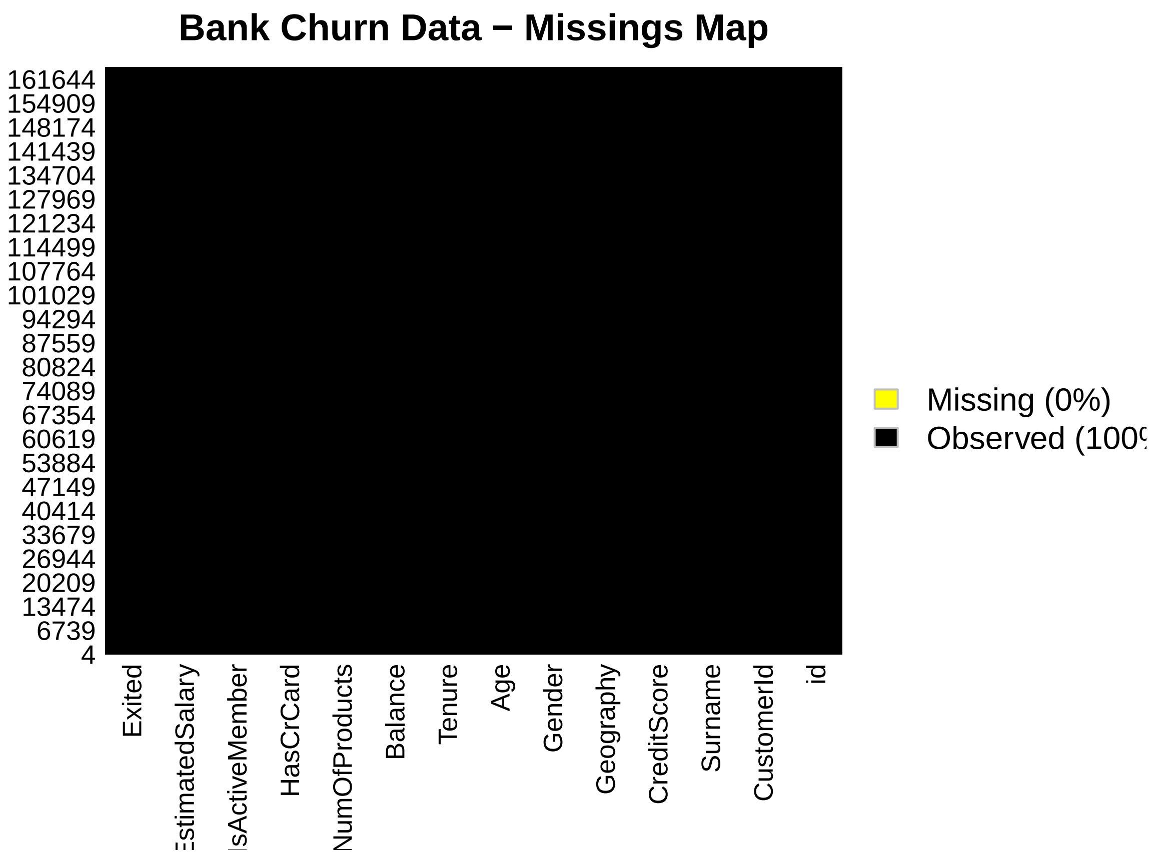
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   32.00   37.00   38.13   42.00   92.00
```

```r
#Exploratory data analysis, finding out missing value
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.1, built: 2022-11-18)
## ## Copyright (C) 2005-2024 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(df.train, main="Bank Churn Data - Missings Map",
        col=c("yellow", "black"), legend=TRUE)
```




```
# Remove ineffective features
options(repos = c(CRAN = "https://cran.rstudio.com"))

install.packages("dplyr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/zh/n835cjwn06s8mgkq7j7c8hj80000gn/T//RtmpCuW1LU/downloaded_packages
```

```
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
```

```
df.train <- select(df.train, -id, -CustomerId, -Surname)

# checking remaining columns
head(df.train,3)
```

```
##   CreditScore Geography Gender Age Tenure Balance NumOfProducts HasCrCard
## 1         668    France   Male  33      3       0             2         1
## 2         627    France   Male  33      1       0             2         1
## 3         678    France   Male  42     10       0             2         1
##   IsActiveMember EstimatedSalary Exited
## 1              0        181450.0      0
## 2              1         49503.5      0
## 3              0        184866.7      0
```

```
str(df.train)
```

```
## 'data.frame':    165034 obs. of  11 variables:
##  $ CreditScore    : int  668 627 678 581 716 588 593 678 676 583 ...
##  $ Geography      : chr  "France" "France" "France" "France" ...
##  $ Gender         : chr  "Male" "Male" "Male" "Male" ...
##  $ Age            : num  33 33 42 34 33 36 30 37 43 40 ...
##  $ Tenure         : int  3 1 10 2 5 4 8 1 4 4 ...
##  $ Balance        : num  0 0 0 148883 0 ...
##  $ NumOfProducts  : int  2 2 2 1 2 1 1 1 2 1 ...
##  $ HasCrCard      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ IsActiveMember : int  0 1 0 1 1 0 0 0 0 1 ...
##  $ EstimatedSalary: num  181450 49504 184867 117948 15069 ...
##  $ Exited         : int  0 0 0 0 0 1 0 0 0 0 ...
```

```
# Converting features to factors
df.train$Geography      <- as.factor(df.train$Geography)
df.train$Gender         <- as.factor(df.train$Gender)
df.train$HasCrCard.     <- as.factor(df.train$HasCrCard)
df.train$IsActiveMember <- as.factor(df.train$IsActiveMember)

str(df.train)
```

```
## 'data.frame':    165034 obs. of  12 variables:
##  $ CreditScore    : int  668 627 678 581 716 588 593 678 676 583 ...
##  $ Geography      : Factor w/ 3 levels "France","Germany",..: 1 1 1 1 3 2 1 3 1 2 ...
##  $ Gender         : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 1 2 2 2 ...
##  $ Age            : num  33 33 42 34 33 36 30 37 43 40 ...
##  $ Tenure         : int  3 1 10 2 5 4 8 1 4 4 ...
##  $ Balance        : num  0 0 0 148883 0 ...
```

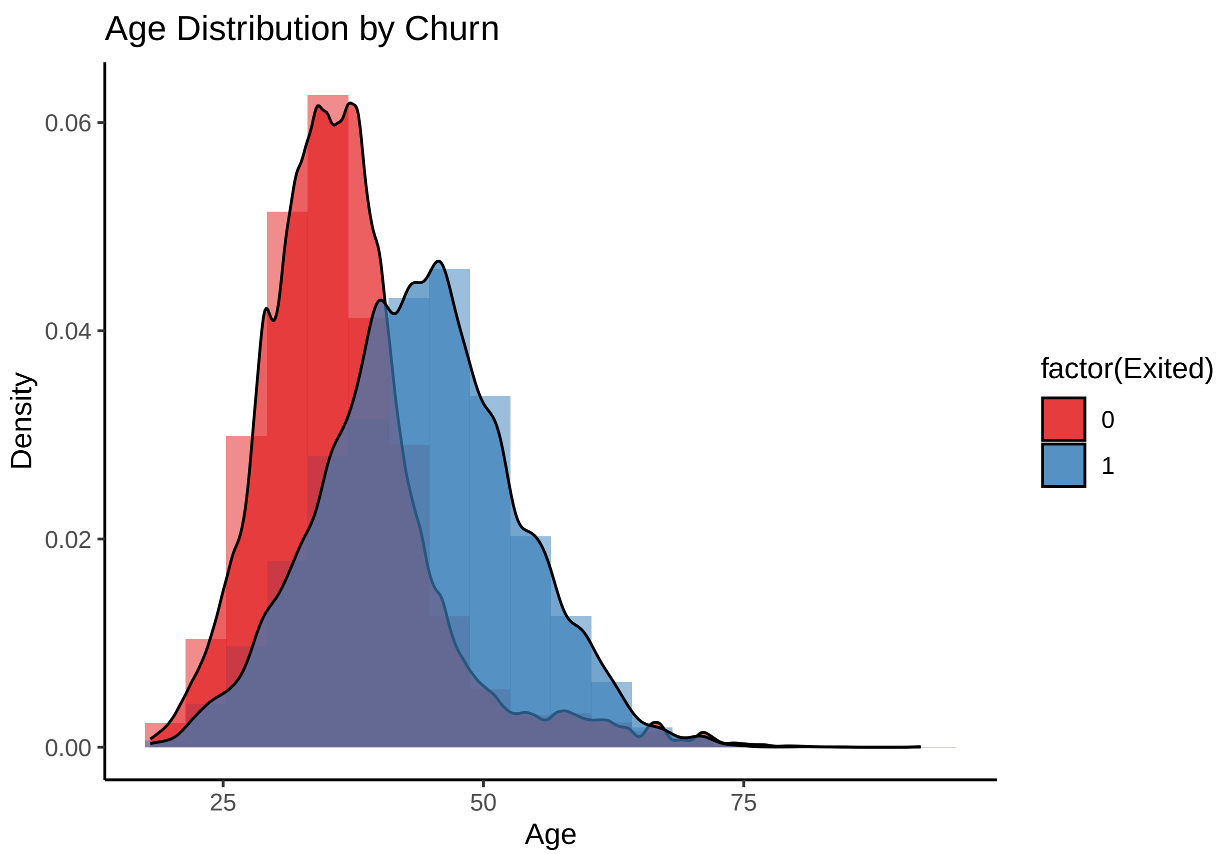
```
##  $ NumOfProducts  : int  2 2 2 1 2 1 1 1 2 1 ...
##  $ HasCrCard      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ IsActiveMember : Factor w/ 2 levels "0","1": 1 2 1 2 2 1 1 1 1 2 ...
##  $ EstimatedSalary: num  181450 49504 184867 117948 15069 ...
##  $ Exited         : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ HasCrCard.     : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
df.train <- select(df.train, -HasCrCard.)

str(df.train)
```

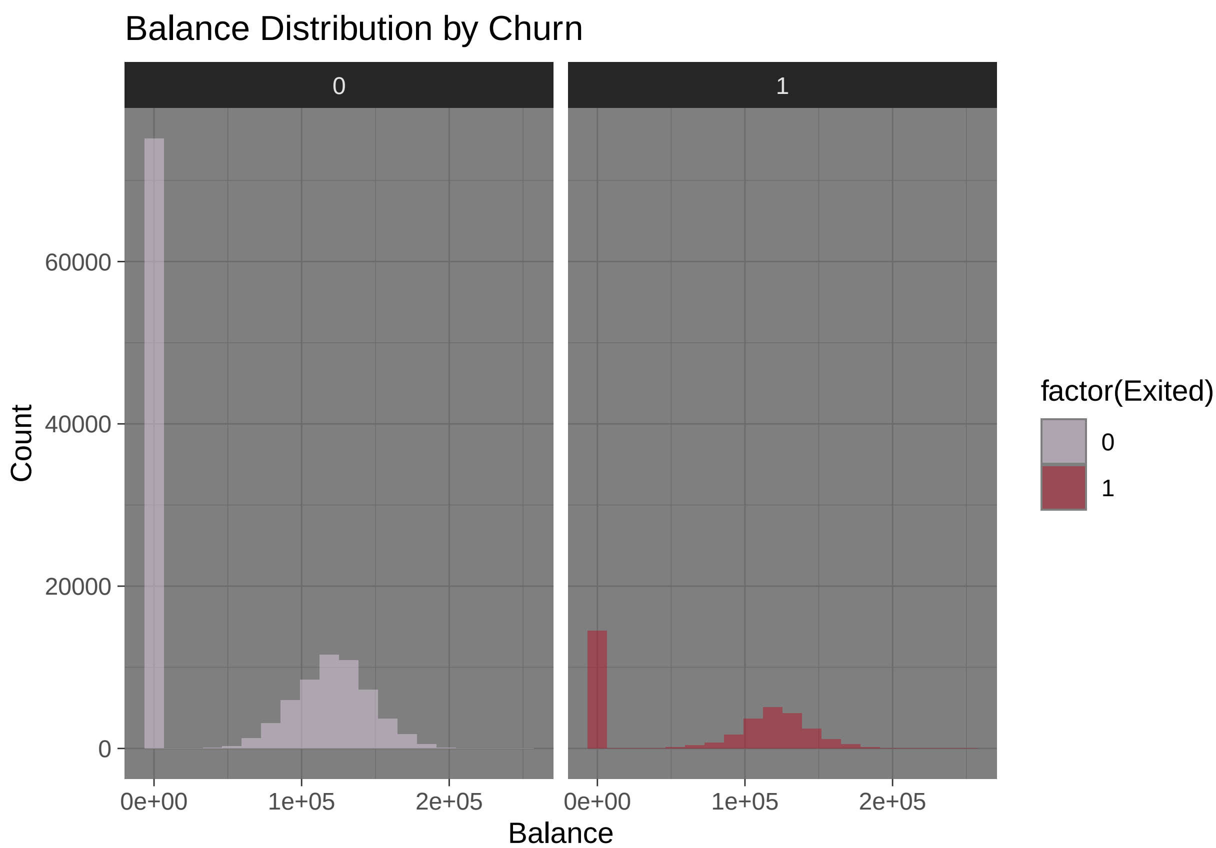
```
## 'data.frame':    165034 obs. of  11 variables:
##  $ CreditScore    : int  668 627 678 581 716 588 593 678 676 583 ...
##  $ Geography      : Factor w/ 3 levels "France","Germany",..: 1 1 1 1 3 2 1 3 1 2 ...
##  $ Gender         : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 1 2 2 2 ...
##  $ Age            : num  33 33 42 34 33 36 30 37 43 40 ...
##  $ Tenure         : int  3 1 10 2 5 4 8 1 4 4 ...
##  $ Balance        : num  0 0 0 148883 0 ...
##  $ NumOfProducts  : int  2 2 2 1 2 1 1 1 2 1 ...
##  $ HasCrCard      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ IsActiveMember : Factor w/ 2 levels "0","1": 1 2 1 2 2 1 1 1 1 2 ...
##  $ EstimatedSalary: num  181450 49504 184867 117948 15069 ...
##  $ Exited         : int  0 0 0 0 0 1 0 0 0 0 ...
```

```
# Exploratory data analysis using GGPlot

library(ggplot2)

# For Age and Churn
ggplot(df.train, aes(x = Age, fill = factor(Exited))) +
  geom_histogram(aes(y = ..density..), position = "identity",
                 bins = 20, alpha = 0.5) +
  geom_density(alpha = 0.7) +
  scale_fill_brewer(palette = "Set1") +
  labs(title = "Age Distribution by Churn", x = "Age", y = "Density") +
  theme_classic()
```

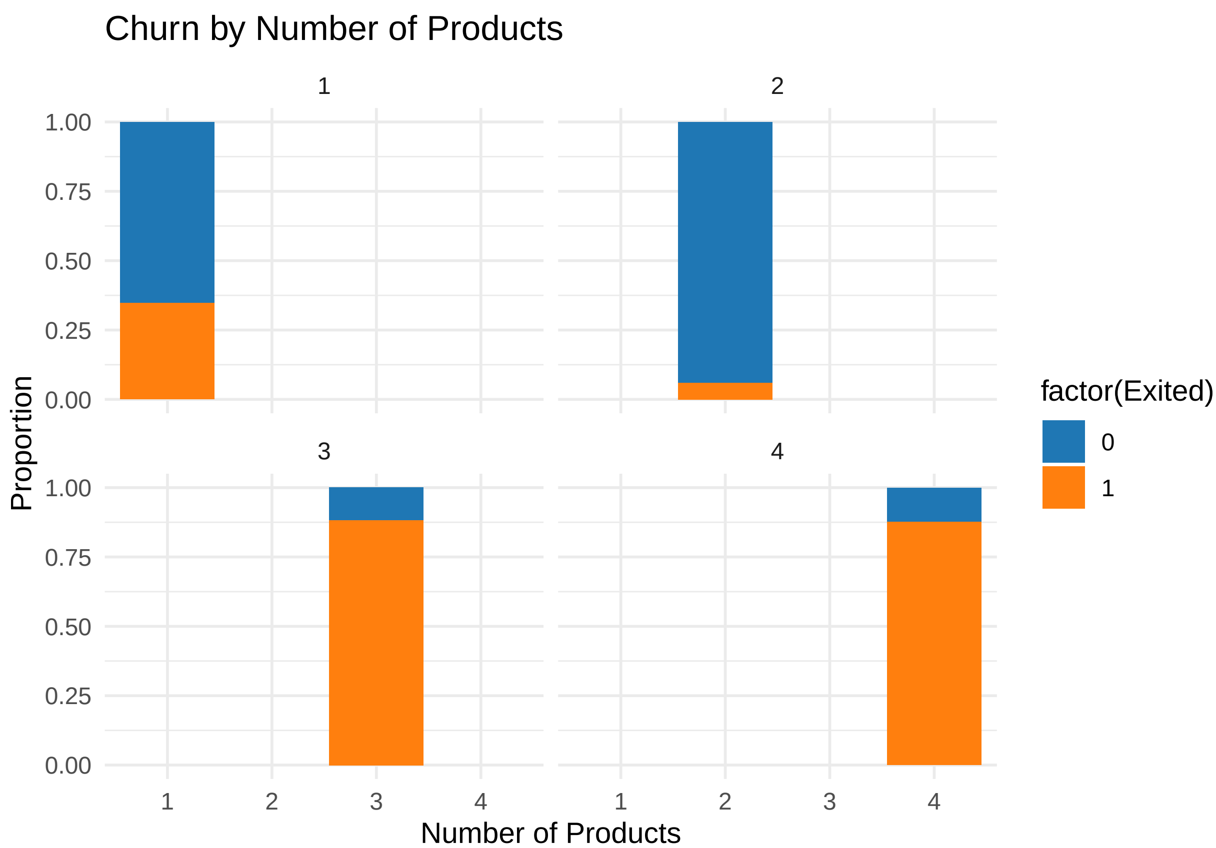
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



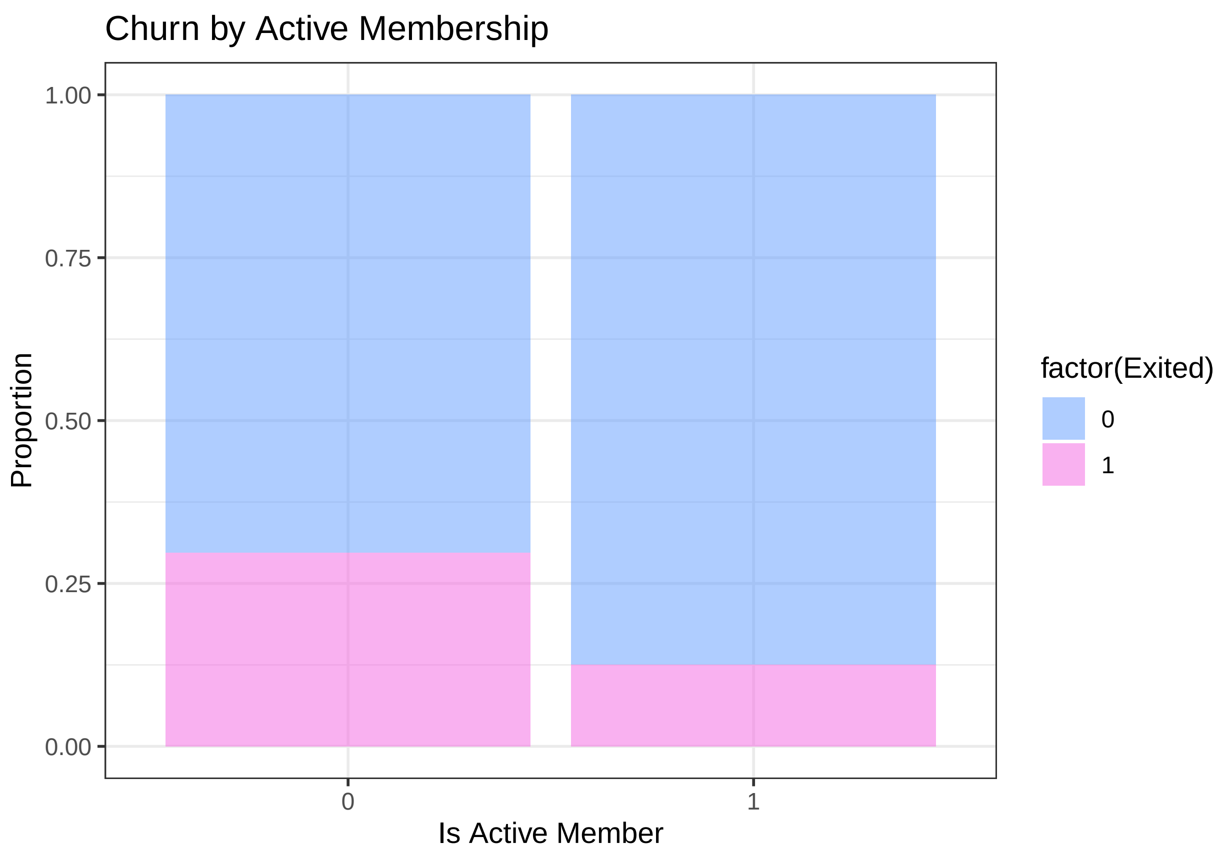

```r
# For Balance and Churn
ggplot(df.train, aes(x = Balance, fill = factor(Exited))) +
  geom_histogram(position = "identity", bins = 20, alpha = 0.5) +
  facet_grid(. ~ Exited) +
  scale_fill_manual(values = c("0" = "#DECBE4", "1" = "#B2182B")) +
  labs(title = "Balance Distribution by Churn", x = "Balance", y = "Count") +
  theme_dark()
```



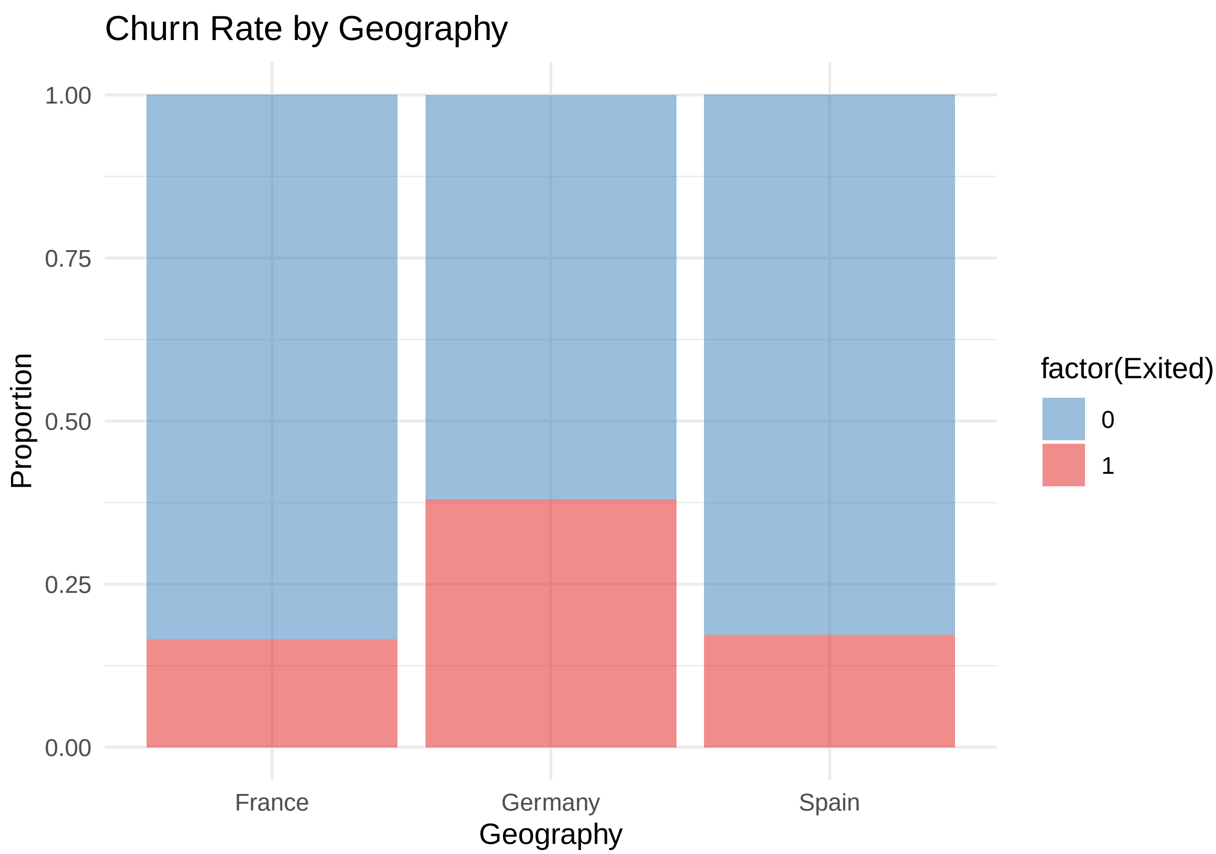

```r
# For NumOfProducts and Churn
ggplot(df.train, aes(x = factor(NumOfProducts), fill = factor(Exited))) +
  geom_bar(position = "fill") +
  facet_wrap(~NumOfProducts) +
  scale_fill_manual(values = c("0" = "#1f77b4", "1" = "#ff7f0e")) +
  labs(title = "Churn by Number of Products",
       x = "Number of Products", y = "Proportion") +
  theme_minimal()
```



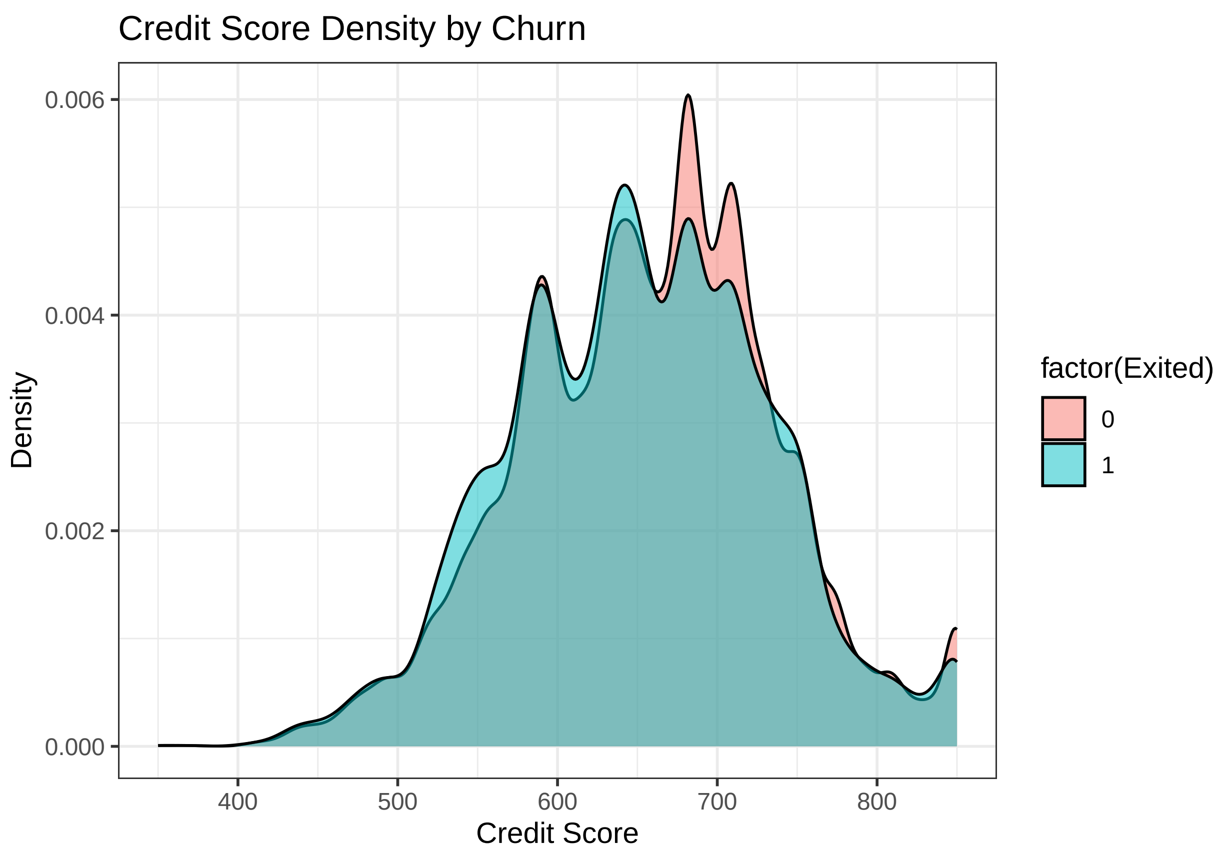

```r
# For IsActiveMember and Churn
ggplot(df.train, aes(x = factor(IsActiveMember), fill = factor(Exited))) +
  geom_bar(position = "fill", alpha = 0.5) +
  scale_fill_manual(values = c("0" = "#619CFF", "1" = "#F564E3")) +
  labs(title = "Churn by Active Membership",
       x = "Is Active Member", y = "Proportion") +
  theme_bw()
```




```r
# For Geography and Churn
ggplot(df.train, aes(x = Geography, fill = factor(Exited))) +
  geom_bar(position = "fill", alpha = 0.5) +
  scale_fill_brewer(palette = "Set1", direction = -1) +
  labs(title = "Churn Rate by Geography",
       x = "Geography", y = "Proportion") +
  theme_minimal()
```




```r
# For CreditScore and Churn
ggplot(df.train, aes(x = CreditScore, fill = factor(Exited))) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("0" = "#F8766D", "1" = "#00BFC4")) +
  labs(title = "Credit Score Density by Churn",
       x = "Credit Score", y = "Density") +
  theme_bw()
```




```r
# 'Exited' is the target variable

# Load the necessary package for sampling
install.packages("caTools")
```

```
##
## The downloaded binary packages are in
##  /var/folders/zh/n835cjwn06s8mgkq7j7c8hj80000gn/T//RtmpCuW1LU/downloaded_packages
```

```r
library(caTools)

# Set a seed for reproducibility
set.seed(101)

# Split the data into training and testing sets
split = sample.split(df.train$Exited, SplitRatio = 0.70)
final.train = subset(df.train, split == TRUE)
final.test = subset(df.train, split == FALSE)
```

```r
# Train the logistic regression model
final.log.model <- glm(Exited ~ ., family = binomial(link = 'logit'),
                       data = df.train)

# Summarize the model
summary(final.log.model)
```

```
##
## Call:
## glm(formula = Exited ~ ., family = binomial(link = "logit"),
##     data = df.train)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.431e+00  7.330e-02 -33.172  < 2e-16 ***
## CreditScore       -7.957e-04  8.679e-05  -9.168  < 2e-16 ***
## GeographyGermany   1.149e+00  1.971e-02  58.287  < 2e-16 ***
## GeographySpain     3.167e-02  1.842e-02   1.720   0.0855 .
## GenderMale        -6.669e-01  1.399e-02 -47.666  < 2e-16 ***
## Age                9.411e-02  7.886e-04 119.345  < 2e-16 ***
## Tenure            -1.543e-02  2.480e-03  -6.220 4.97e-10 ***
## Balance           -1.986e-06  1.422e-07 -13.968  < 2e-16 ***
## NumOfProducts     -9.130e-01  1.378e-02 -66.263  < 2e-16 ***
## HasCrCard         -1.610e-01  1.596e-02 -10.084  < 2e-16 ***
## IsActiveMember1   -1.282e+00  1.500e-02 -85.479  < 2e-16 ***
## EstimatedSalary    9.414e-07  1.391e-07   6.767 1.31e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 170337  on 165033  degrees of freedom
## Residual deviance: 130536  on 165022  degrees of freedom
## AIC: 130560
##
## Number of Fisher Scoring iterations: 5
```

```r
# Predict on the test set
fitted.probabilities <- predict(final.log.model,
                                newdata = final.test, type = 'response')
fitted.results <- ifelse(fitted.probabilities > 0.5, 1, 0)

# Calculate and print the accuracy
misClasificError <- mean(fitted.results != final.test$Exited)
print(paste('Accuracy:', 1 - misClasificError))
```

```
## [1] "Accuracy: 0.83474045647344"
```

```r
# Create a confusion matrix
confusionMatrix <- table(final.test$Exited, fitted.results)

head(confusionMatrix)
```

```
##    fitted.results
##         0     1
##   0 37288  1746
##   1  6436  4040
```

```r
# Load the necessary package for calculating sensitivity and specificity
install.packages("caret")
```

```
##
## The downloaded binary packages are in
##  /var/folders/zh/n835cjwn06s8mgkq7j7c8hj80000gn/T//RtmpCuW1LU/downloaded_packages
```

```r
library(caret)
```

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

```r
# Convert to factors for confusion matrix calculations
final.test$Exited <- factor(final.test$Exited, levels = c(0, 1))
fitted.results <- factor(fitted.results, levels = c(0, 1))

# Calculate sensitivity and specificity
sensitivity <- sensitivity(confusionMatrix, positive = "1")
specificity <- specificity(confusionMatrix, positive = "1")

# Print the sensitivity and specificity
print(paste('Sensitivity:', sensitivity))
```

```
## [1] "Sensitivity: 0.698237124092637"
```

```r
print(paste('Specificity:', specificity))
```

```
## [1] "Specificity: 0.698237124092637"
```

```r
# Load the new dataset
new_customer_data <- read.csv("NewCustomerDataset-2.csv")

str(new_customer_data)
```

```
## 'data.frame':    110023 obs. of  13 variables:
##  $ id             : int  165034 165035 165036 165037 165038 165039 165040 165041 165042 165043 ...
##  $ CustomerId     : int  15773898 15782418 15807120 15808905 15607314 15672704 15647838 15775307 1565
##  $ Surname        : chr  "Lucchese" "Nott" "K?" "O'Donnell" ...
##  $ CreditScore    : int  586 683 656 681 752 593 682 539 845 645 ...
##  $ Geography      : chr  "France" "France" "France" "France" ...
##  $ Gender         : chr  "Female" "Female" "Female" "Male" ...
##  $ Age            : num  23 46 34 36 38 22 45 47 47 30 ...
##  $ Tenure         : int  2 2 7 8 10 9 4 8 3 5 ...
##  $ Balance        : num  0 0 0 0 121264 ...
##  $ NumOfProducts  : int  2 1 2 1 1 2 2 2 1 2 ...
##  $ HasCrCard      : int  0 1 1 1 1 0 1 1 1 0 ...
##  $ IsActiveMember : int  1 0 0 0 0 0 1 1 0 1 ...
##  $ EstimatedSalary: num  160977 72549 138882 113932 139431 ...
```

```r
head(new_customer_data, 5)
```

```
##       id CustomerId   Surname CreditScore Geography Gender Age Tenure  Balance
## 1 165034   15773898  Lucchese         586    France Female  23      2      0.0
## 2 165035   15782418      Nott         683    France Female  46      2      0.0
## 3 165036   15807120        K?         656    France Female  34      7      0.0
## 4 165037   15808905 O'Donnell         681    France   Male  36      8      0.0
## 5 165038   15607314   Higgins         752   Germany   Male  38     10 121263.6
##   NumOfProducts HasCrCard IsActiveMember EstimatedSalary
## 1             2         0              1       160976.75
## 2             1         1              0        72549.27
## 3             2         1              0       138882.09
## 4             1         1              0       113931.57
## 5             1         1              0       139431.00
```

```r
summary(new_customer_data)
```

```
##        id           CustomerId          Surname           CreditScore
##  Min.   :165034   Min.   :15565701   Length:110023      Min.   :350.0
##  1st Qu.:192540   1st Qu.:15632859   Class :character   1st Qu.:597.0
##  Median :220045   Median :15690175   Mode  :character   Median :660.0
##  Mean   :220045   Mean   :15692097                      Mean   :656.5
##  3rd Qu.:247550   3rd Qu.:15756926                      3rd Qu.:710.0
##  Max.   :275056   Max.   :15815690                      Max.   :850.0
##   Geography            Gender                Age            Tenure
##  Length:110023      Length:110023      Min.   :18.00   Min.   : 0.000
##  Class :character   Class :character   1st Qu.:32.00   1st Qu.: 3.000
##  Mode  :character   Mode  :character   Median :37.00   Median : 5.000
##                                        Mean   :38.12   Mean   : 4.997
##                                        3rd Qu.:42.00   3rd Qu.: 7.000
##                                        Max.   :92.00   Max.   :10.000
##     Balance       NumOfProducts     HasCrCard      IsActiveMember
##  Min.   :     0   Min.   :1.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:     0   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :     0   Median :2.000   Median :1.000   Median :0.0000
##  Mean   : 55334   Mean   :1.553   Mean   :0.753   Mean   :0.4952
##  3rd Qu.:120146   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :250898   Max.   :4.000   Max.   :1.000   Max.   :1.0000
##  EstimatedSalary
##  Min.   :    11.58
##  1st Qu.: 74440.32
##  Median :117832.23
##  Mean   :112315.15
##  3rd Qu.:154631.35
##  Max.   :199992.48
```

```r
# Preprocess the data Handling missing values
new_customer_data$EstimatedSalary[is.na(new_customer_data$EstimatedSalary)] <-
  median(df.train$EstimatedSalary, na.rm = TRUE)

summary(new_customer_data)
```

```
##        id           CustomerId        Surname           CreditScore
##  Min.   :165034   Min.   :15565701   Length:110023      Min.   :350.0
##  1st Qu.:192540   1st Qu.:15632859   Class :character   1st Qu.:597.0
##  Median :220045   Median :15690175   Mode  :character   Median :660.0
##  Mean   :220045   Mean   :15692097                      Mean   :656.5
##  3rd Qu.:247550   3rd Qu.:15756926                      3rd Qu.:710.0
##  Max.   :275056   Max.   :15815690                      Max.   :850.0
##   Geography           Gender               Age            Tenure
##  Length:110023      Length:110023      Min.   :18.00   Min.   : 0.000
##  Class :character   Class :character   1st Qu.:32.00   1st Qu.: 3.000
##  Mode  :character   Mode  :character   Median :37.00   Median : 5.000
##                                        Mean   :38.12   Mean   : 4.997
##                                        3rd Qu.:42.00   3rd Qu.: 7.000
##                                        Max.   :92.00   Max.   :10.000
##     Balance       NumOfProducts     HasCrCard     IsActiveMember
##  Min.   :     0   Min.   :1.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:     0   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :     0   Median :2.000   Median :1.000   Median :0.0000
##  Mean   : 55334   Mean   :1.553   Mean   :0.753   Mean   :0.4952
##  3rd Qu.:120146   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :250898   Max.   :4.000   Max.   :1.000   Max.   :1.0000
##  EstimatedSalary
##  Min.   :    11.58
##  1st Qu.: 74440.32
##  Median :117832.23
##  Mean   :112315.15
##  3rd Qu.:154631.35
##  Max.   :199992.48
```

```
str(new_customer_data)
```

```
## 'data.frame':    110023 obs. of  13 variables:
##  $ id             : int  165034 165035 165036 165037 165038 165039 165040 165041 165042 165043 ...
##  $ CustomerId     : int  15773898 15782418 15807120 15808905 15607314 15672704 15647838 15775307 156
##  $ Surname        : chr  "Lucchese" "Nott" "K?" "O'Donnell" ...
##  $ CreditScore    : int  586 683 656 681 752 593 682 539 845 645 ...
##  $ Geography      : chr  "France" "France" "France" "France" ...
##  $ Gender         : chr  "Female" "Female" "Female" "Male" ...
##  $ Age            : num  23 46 34 36 38 22 45 47 47 30 ...
##  $ Tenure         : int  2 2 7 8 10 9 4 8 3 5 ...
##  $ Balance        : num  0 0 0 0 121264 ...
##  $ NumOfProducts  : int  2 1 2 1 1 2 2 2 1 2 ...
##  $ HasCrCard      : int  0 1 1 1 1 0 1 1 1 0 ...
##  $ IsActiveMember : int  1 0 0 0 0 0 1 1 0 1 ...
##  $ EstimatedSalary: num  160977 72549 138882 113932 139431 ...
```

```
# Remove unnecessary features
new_customer_data <- new_customer_data %>% select(-id, -CustomerId, -Surname)

# Convert categorical variables to factors
new_customer_data$Geography <- as.factor(new_customer_data$Geography)
new_customer_data$Gender <- as.factor(new_customer_data$Gender)
new_customer_data$IsActiveMember <- as.factor(new_customer_data$IsActiveMember)
```

```r
# Ensure categorical variables are factorized as in the training set
new_customer_data$Geography <- as.factor(new_customer_data$Geography)
new_customer_data$Gender <- as.factor(new_customer_data$Gender)

str(new_customer_data)
```

```
## 'data.frame':    110023 obs. of  10 variables:
##  $ CreditScore    : int  586 683 656 681 752 593 682 539 845 645 ...
##  $ Geography      : Factor w/ 3 levels "France","Germany",..: 1 1 1 1 2 1 3 3 1 3 ...
##  $ Gender         : Factor w/ 2 levels "Female","Male": 1 1 1 2 2 1 2 1 1 2 ...
##  $ Age            : num  23 46 34 36 38 22 45 47 47 30 ...
##  $ Tenure         : int  2 2 7 8 10 9 4 8 3 5 ...
##  $ Balance        : num  0 0 0 0 121264 ...
##  $ NumOfProducts  : int  2 1 2 1 1 2 2 2 1 2 ...
##  $ HasCrCard      : int  0 1 1 1 1 0 1 1 1 0 ...
##  $ IsActiveMember : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 2 1 2 ...
##  $ EstimatedSalary: num  160977 72549 138882 113932 139431 ...
```

```r
# Predict churn
new_customer_data$predicted_churn <- predict(final.log.model,
                                             newdata = new_customer_data,
                                             type = 'response')
new_customer_data$predicted_churn <-
  ifelse(new_customer_data$predicted_churn > 0.5, 1, 0)

# View the predictions
head(new_customer_data$predicted_churn)
```

```
## [1] 0 1 0 0 0 0
```

# Report

The project analysis on predicting bank customer churn through machine learning reveals the following:

- Data Preparation: Missing values were addressed, and datasets divided into training and testing sets.
- Model Training: A logistic regression model was trained with an accuracy of 83.47%.
- Model Evaluation: Sensitivity and specificity were calculated, both approximately 69.82%.
- Predictions: The model predicted churn for a subset of customers from a new dataset.

This process exemplifies a data-driven approach to understand and mitigate customer attrition.

The accuracy of the model is approximately 83.47%, meaning it correctly predicts customer churn 83.47% of the time. The confusion matrix provides a more detailed breakdown:

- True negatives (correctly predicted non-churn): 37,288
- False positives (incorrectly predicted churn): 1,746
- False negatives (incorrectly predicted non-churn): 6,436
- True positives (correctly predicted churn): 4,040

The sensitivity (true positive rate) is approximately 69.82%, indicating that the model correctly identifies 69.82% of the customers who will churn. The specificity (true negative rate) is also about 69.82%, showing the model correctly identifies 69.82% of the customers who will not churn.

The predictions for a new customer dataset indicate that out of six customers, the model predicts one will churn (the second customer), and the remaining five will not churn. This information can be vital for targeted customer retention strategies.