MOT143A Business Analytics

Symeon Efstathiou & Stefanos Adamidis

11-07-2025

1. Business Problem Definition and Objective

Business Context

In the banking sector, **customer retention is a critical driver of sustained profitability**. Research shows that acquiring a new customer can cost five to twenty-five times more than retaining an existing one (Gallo, 2014; Reichheld & Sasser, 1990). When customers churn, they not only withdraw their funds but also eliminate long-term value through lost cross-selling opportunities, diminished brand loyalty and reduced customer referrals (Hwang et al., 2004).

Despite its strategic importance, **churn management in many banks remains reactive**. Retention efforts are often triggered too late, after customers have disengaged or left. Historically, banks have relied on rule-based heuristics or simple statistical scoring models to detect churn (Verbeke et al., 2012). While interpretable, such models struggle to capture non-linear behavior, adapt to changing customer dynamics or scale across complex segments.

Given the high stakes involved, even small improvements in churn prediction can translate into substantial business impact. According to Reichheld and Sasser (1990), reducing churn by just 5% can increase profits by 25% to 95%, depending on the industry. In banking, where customer acquisition costs are high and customer lifetime value is long-tailed, this makes proactive churn detection not just operationally relevant, but strategically essential (Gallo, 2014).

These traditional models often rely on static thresholds, such as flagging a customer as high-risk if their transaction volume drops by more than 20% or if they haven’t logged into their account for 30 days. While simple to implement, such heuristics require frequent manual updates, cannot account for complex interactions between variables and risk flagging false positives or missing nuanced patterns of attrition (Neslin et al., 2006).

In contrast, the approach adopted in this project enables scalable and adaptive churn detection. By training models on historical data that includes both behavioral and financial attributes, we aim to uncover non-obvious combinations of risk indicators. For example, a moderate drop in credit utilization might not indicate churn on its own, but when combined with reduced online logins and low customer service engagement, it may strongly predict customer departure. The proposed approach in this projects uses a pipeline that learns these interactions automatically, allowing for **more accurate, personalized and actionable churn risk scoring**.

This project addresses this gap by developing a **predictive churn model** that enables proactive intervention. Our research question is:

> **To what extent can machine learning models predict customer churn using historical data in a retail banking context?**

We frame this as a **binary classification task**, where the goal is to identify customers at risk of attrition before they leave. Doing so would enable the bank's customer retention team to take targeted action, such as deploying loyalty programs, offering personalized outreach or adjusting service levels, to prevent revenue loss and preserve customer lifetime value (Ngai et al., 2009).

Technically, we apply a **modern machine learning pipeline** that offers several advantages:

- Automated feature selection and engineering
- Robust handling of high-dimensional behavioral data
- Comparative performance analysis between interpretable models (e.g., Logistic Regression) and more flexible non-linear models (e.g., Random Forest)

This approach not only enhances predictive performance, but also delivers **business relevant insights** that support scalable, data-driven retention strategies. Ultimately, the project aims to transform static customer records into dynamic insights, empowering bank managers to act on churn risk with precision, speed and impact.

Machine Learning Objective

The objective of this analysis is to build a **predictive model** that classifies whether a customer will churn, based on their **demographic and behavioral profiles**. This constitutes a **binary classification** problem, where:

- **Target variable**: `Attrition_Flag` (1 = churned customer, 0 = retained customer)
- **Input features** include: age, gender, income, account tenure, credit card usage patterns, transaction frequency and service interaction history

A successful model should:

- **Accurately identify customers at risk** of churning before they leave
- **Support targeted retention strategies** based on model outputs
- **Enable proactive decision-making** to protect customer lifetime value

Model performance will be assessed using standard classification metrics such as **accuracy**, **precision**, **recall** and **F1 score**, which help quantify the model's effectiveness in identifying churned versus loyal customers.

In short, the goal of this project is to translate customer data into actionable insights that allow the bank to respond to churn risk with precision, guide targeted retention actions and improve customer lifetime value by enabling early interventions.

About the Dataset

The dataset used for this analysis is the **Bank Customer Churn Dataset**, publicly available on Kaggle (https://www.kaggle.com/code/nnttch/bankchurners-eda-smote-statsmodels#Feature-Engineering). It contains detailed information on over **10,000 customers** of a retail bank in the United States.

In total, the dataset includes **23 variables**, capturing a mix of:

- **Demographic attributes**: such as age, gender, and marital status
- **Account characteristics**: including account tenure, credit card status and product usage
- **Transactional behavior**: like number of transactions and total transaction amount
- **Customer status**: for instance, active/inactive indicator and months inactive

This dataset enables a thorough analysis of **which features drive customer attrition** and whether historical patterns in behavior can help predict future churn. While comprehensive, the dataset presents challenges such as:

- **Class imbalance**: churn is relatively rare compared to retained customers.
- **Potential missing values** or inconsistencies in specific fields.
- **Correlated variables**, which may require dimensionality reduction or feature selection.

These issues will be addressed in the upcoming data cleaning and transformation steps.

2. Data Preparation

Effective data preparation is critical for ensuring high-quality inputs to machine learning models. This section covers data loading, cleaning, transformation and univariate exploration.

2.1 Loading and Initial Assessment

We begin by loading the necessary libraries and importing the dataset.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
## Warning: package 'readr' was built under R version 4.4.3
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
## Warning: package 'forcats' was built under R version 4.4.3
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
library(ggplot2)
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 4.4.3
```

```
df <- read_csv("BankChurners.csv")
head(df)
```

```
## # A tibble: 6 × 23
##   CLIENTNUM Attrition_Flag   Customer_Age Gender Dependent_count Education_Level
##       <dbl> <chr>                   <dbl> <chr>            <dbl> <chr>
## 1 768805383 Existing Custom…           45 M                    3 High School
## 2 818770008 Existing Custom…           49 F                    5 Graduate
## 3 713982108 Existing Custom…           51 M                    3 Graduate
## 4 769911858 Existing Custom…           40 F                    4 High School
## 5 709106358 Existing Custom…           40 M                    3 Uneducated
## 6 713061558 Existing Custom…           44 M                    2 Graduate
## # ℹ 17 more variables: Marital_Status <chr>, Income_Category <chr>,
## #   Card_Category <chr>, Months_on_book <dbl>, Total_Relationship_Count <dbl>,
## #   Months_Inactive_12_mon <dbl>, Contacts_Count_12_mon <dbl>,
## #   Credit_Limit <dbl>, Total_Revolving_Bal <dbl>, Avg_Open_To_Buy <dbl>,
## #   Total_Amt_Chng_Q4_Q1 <dbl>, Total_Trans_Amt <dbl>, Total_Trans_Ct <dbl>,
## #   Total_Ct_Chng_Q4_Q1 <dbl>, Avg_Utilization_Ratio <dbl>,
## #   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inac
tive_12_mon_1 <dbl>, …
```

The dataset was imported and inspected for structure, variable types and completeness. Key data such as the number of observations, feature types and potential target imbalance were recorded.

Initial Inspection

Let’s take a first look at the structure and summary of the data.

```
str(df)
```

```
## spc_tbl_ [10,127 × 23] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ CLIENTNUM
: num [1:10127] 7.69e+08 8.19e+08 7.14e+08 7.70e+08 7.09e+08 ...
##  $ Attrition_Flag
: chr [1:10127] "Existing Customer" "Existing Customer" "Existing Customer" "Existing Customer" ...
##  $ Customer_Age
: num [1:10127] 45 49 51 40 40 44 51 32 37 48 ...
##  $ Gender
: chr [1:10127] "M" "F" "M" "F" ...
##  $ Dependent_count
: num [1:10127] 3 5 3 4 3 2 4 0 3 2 ...
##  $ Education_Level
: chr [1:10127] "High School" "Graduate" "Graduate" "High School" ...
##  $ Marital_Status
: chr [1:10127] "Married" "Single" "Married" "Unknown" ...
##  $ Income_Category
: chr [1:10127] "$60K - $80K" "Less than $40K" "$80K - $120K" "Less than $40K" ...
##  $ Card_Category
: chr [1:10127] "Blue" "Blue" "Blue" "Blue" ...
##  $ Months_on_book
: num [1:10127] 39 44 36 34 21 36 46 27 36 36 ...
##  $ Total_Relationship_Count
: num [1:10127] 5 6 4 3 5 3 6 2 5 6 ...
##  $ Months_Inactive_12_mon
: num [1:10127] 1 1 1 4 1 1 1 2 2 3 ...
##  $ Contacts_Count_12_mon
: num [1:10127] 3 2 0 1 0 2 3 2 0 3 ...
##  $ Credit_Limit
: num [1:10127] 12691 8256 3418 3313 4716 ...
##  $ Total_Revolving_Bal
: num [1:10127] 777 864 0 2517 0 ...
##  $ Avg_Open_To_Buy
: num [1:10127] 11914 7392 3418 796 4716 ...
##  $ Total_Amt_Chng_Q4_Q1
: num [1:10127] 1.33 1.54 2.59 1.4 2.17 ...
##  $ Total_Trans_Amt
: num [1:10127] 1144 1291 1887 1171 816 ...
##  $ Total_Trans_Ct
: num [1:10127] 42 33 20 20 28 24 31 36 24 32 ...
```

```
##  $ Total_Ct_Chng_Q4_Q1
: num [1:10127] 1.62 3.71 2.33 2.33 2.5 ...
##  $ Avg_Utilization_Ratio
: num [1:10127] 0.061 0.105 0 0.76 0 0.311 0.066 0.048 0.113 0.144 ...
##  $ Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inact
ive_12_mon_1: num [1:10127] 9.34e-05 5.69e-05 2.11e-05 1.34e-04 2.17e-05 ...
##  $ Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inact
ive_12_mon_2: num [1:10127] 1 1 1 1 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   CLIENTNUM = col_double(),
##   ..   Attrition_Flag = col_character(),
##   ..   Customer_Age = col_double(),
##   ..   Gender = col_character(),
##   ..   Dependent_count = col_double(),
##   ..   Education_Level = col_character(),
##   ..   Marital_Status = col_character(),
##   ..   Income_Category = col_character(),
##   ..   Card_Category = col_character(),
##   ..   Months_on_book = col_double(),
##   ..   Total_Relationship_Count = col_double(),
##   ..   Months_Inactive_12_mon = col_double(),
##   ..   Contacts_Count_12_mon = col_double(),
##   ..   Credit_Limit = col_double(),
##   ..   Total_Revolving_Bal = col_double(),
##   ..   Avg_Open_To_Buy = col_double(),
##   ..   Total_Amt_Chng_Q4_Q1 = col_double(),
##   ..   Total_Trans_Amt = col_double(),
##   ..   Total_Trans_Ct = col_double(),
##   ..   Total_Ct_Chng_Q4_Q1 = col_double(),
##   ..   Avg_Utilization_Ratio = col_double(),
##   ..   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_I
nactive_12_mon_1 = col_double(),
##   ..   Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_I
nactive_12_mon_2 = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
library(knitr)
summary(df)
```

```
##    CLIENTNUM         Attrition_Flag      Customer_Age      Gender
##  Min.   :708082083   Length:10127       Min.   :26.00   Length:10127
##  1st Qu.:713036770   Class :character   1st Qu.:41.00   Class :character
##  Median :717926358   Mode  :character   Median :46.00   Mode  :character
##  Mean   :739177606                      Mean   :46.33
##  3rd Qu.:773143533                      3rd Qu.:52.00
##  Max.   :828343083                      Max.   :73.00
##  Dependent_count Education_Level    Marital_Status     Income_Category
##  Min.   :0.000   Length:10127       Length:10127       Length:10127
##  1st Qu.:1.000   Class :character   Class :character   Class :character
##  Median :2.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2.346
##  3rd Qu.:3.000
##  Max.   :5.000
##  Card_Category      Months_on_book  Total_Relationship_Count
##  Length:10127       Min.   :13.00   Min.   :1.000
##  Class :character   1st Qu.:31.00   1st Qu.:3.000
##  Mode  :character   Median :36.00   Median :4.000
##                     Mean   :35.93   Mean   :3.813
##                     3rd Qu.:40.00   3rd Qu.:5.000
##                     Max.   :56.00   Max.   :6.000
##  Months_Inactive_12_mon Contacts_Count_12_mon  Credit_Limit
##  Min.   :0.000          Min.   :0.000         Min.   : 1438
##  1st Qu.:2.000          1st Qu.:2.000         1st Qu.: 2555
##  Median :2.000          Median :2.000         Median : 4549
##  Mean   :2.341          Mean   :2.455         Mean   : 8632
##  3rd Qu.:3.000          3rd Qu.:3.000         3rd Qu.:11068
##  Max.   :6.000          Max.   :6.000         Max.   :34516
##  Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1 Total_Trans_Amt
##  Min.   :   0        Min.   :    3   Min.   :0.0000       Min.   :  510
##  1st Qu.: 359        1st Qu.: 1324   1st Qu.:0.6310       1st Qu.: 2156
##  Median :1276        Median : 3474   Median :0.7360       Median : 3899
##  Mean   :1163        Mean   : 7469   Mean   :0.7599       Mean   : 4404
##  3rd Qu.:1784        3rd Qu.: 9859   3rd Qu.:0.8590       3rd Qu.: 4741
##  Max.   :2517        Max.   :34516   Max.   :3.3970       Max.   :18484
##  Total_Trans_Ct   Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
##  Min.   : 10.00   Min.   :0.0000      Min.   :0.0000
##  1st Qu.: 45.00   1st Qu.:0.5820      1st Qu.:0.0230
##  Median : 67.00   Median :0.7020      Median :0.1760
```

```
##  Mean   : 64.86   Mean   :0.7122      Mean   :0.2749
##  3rd Qu.: 81.00   3rd Qu.:0.8180      3rd Qu.:0.5030
##  Max.   :139.00   Max.   :3.7140      Max.   :0.9990
##  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactiv
e_12_mon_1
##  Min.   :0.0000077
##  1st Qu.:0.0000990
##  Median :0.0001815
##  Mean   :0.1599975
##  3rd Qu.:0.0003373
##  Max.   :0.9995800
##  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactiv
e_12_mon_2
##  Min.   :0.00042
##  1st Qu.:0.99966
##  Median :0.99982
##  Mean   :0.84000
##  3rd Qu.:0.99990
##  Max.   :0.99999
```

Based on the above short analysis of the dataset, the following summary tables provide a quick overview of the numeric and categorical variables, including their distributions and completeness. These serve as a foundational reference before deeper exploration.

```
numeric_summary <- df %>%
  select(where(is.numeric)) %>%
  summarise(across(everything(), list(
    Mean = ~mean(.x, na.rm = TRUE),
    SD = ~sd(.x, na.rm = TRUE),
    Min = ~min(.x, na.rm = TRUE),
    Max = ~max(.x, na.rm = TRUE),
    Missing = ~sum(is.na(.x))
  ), .names = "{.col}_{.fn}")) %>%
  pivot_longer(
    cols = everything(),
    names_to = c("Variable", "Metric"),
    names_pattern = "(.+)_(Mean|SD|Min|Max|Missing)",
    values_to = "Value"
  ) %>%
  pivot_wider(names_from = Metric, values_from = Value)

numeric_summary <- numeric_summary %>%
  mutate(Variable = case_when(
    Variable == "Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_M
onths_Inactive_12_mon_1" ~ "Naive_Bayes_Classifier_1",
    Variable == "Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_M
onths_Inactive_12_mon_2" ~ "Naive_Bayes_Classifier_2",
    TRUE ~ Variable
  ))

#kable(numeric_summary, digits = 2, caption = "Table 1: Summary of Numeric Variables")
```

```
cat_summary <- df %>%
  select(where(~!is.numeric(.x))) %>%
  summarise(across(everything(), list(
    Unique_Levels = ~as.character(n_distinct(.x)),
    Most_Common = ~as.character(names(which.max(table(.x)))),
    Freq = ~as.character(round(100 * max(table(.x)) / length(.x), 1)),
    Missing = ~as.character(sum(is.na(.x)))
  ), .names = "{.col}_{.fn}")) %>%
  pivot_longer(
    cols = everything(),
    names_to = c("Variable", "Metric"),
    names_pattern = "(.+)_(Unique_Levels|Most_Common|Freq|Missing)",
    values_to = "Value"
  ) %>%
  pivot_wider(names_from = Metric, values_from = Value)

#kable(cat_summary, caption = "Table 2: Summary of Categorical Variables", align = "l")
```

```
kable(numeric_summary, digits = 2, caption = "Table 1: Summary of numeric variables")
```

Table 1: Summary of numeric variables

Variable	Mean	SD	Min	Max	Missing
CLIENTNUM	739177606.33	36903783.45	708082083.0	828343083.00	0
Customer_Age	46.33	8.02	26.0	73.00	0
Dependent_count	2.35	1.30	0.0	5.00	0
Months_on_book	35.93	7.99	13.0	56.00	0
Total_Relationship_Count	3.81	1.55	1.0	6.00	0
Months_Inactive_12_mon	2.34	1.01	0.0	6.00	0
Contacts_Count_12_mon	2.46	1.11	0.0	6.00	0
Credit_Limit	8631.95	9088.78	1438.3	34516.00	0

Variable	Mean	SD	Min	Max	Missing
Total_Revolving_Bal	1162.81	814.99	0.0	2517.00	0
Avg_Open_To_Buy	7469.14	9090.69	3.0	34516.00	0
Total_Amt_Chng_Q4_Q1	0.76	0.22	0.0	3.40	0
Total_Trans_Amt	4404.09	3397.13	510.0	18484.00	0
Total_Trans_Ct	64.86	23.47	10.0	139.00	0
Total_Ct_Chng_Q4_Q1	0.71	0.24	0.0	3.71	0
Avg_Utilization_Ratio	0.27	0.28	0.0	1.00	0
Naive_Bayes_Classifier_1	0.16	0.37	0.0	1.00	0
Naive_Bayes_Classifier_2	0.84	0.37	0.0	1.00	0

```
kable(cat_summary, caption = "Table 2: Summary of categorical variables")
```

Table 2: Summary of categorical variables

Variable	Unique_Levels	Most_Common	Freq	Missing
Attrition_Flag	2	Existing Customer	83.9	0
Gender	2	F	52.9	0
Education_Level	7	Graduate	30.9	0
Marital_Status	4	Married	46.3	0
Income_Category	6	Less than $40K	35.2	0
Card_Category	4	Blue	93.2	0

After loading the dataset, the following **table** presents each variable in the dataset and a short explanation of each:

Variable descriptions of the dataset

Variable	Type	Description
Customer_Age	Numeric	Age of the customer (26–73 years).
Gender	Categorical	Customer gender (M = Male, F = Female).
Dependent_count	Numeric	Number of dependents the customer has (0–5).
Education_Level	Categorical	Highest education attained (e.g., Graduate, High School, Unknown).
Marital_Status	Categorical	Marital status (Married, Single, Divorced, Unknown).
Income_Category	Categorical	Income range (e.g., Less than $40K, $60K–$80K, Unknown).
Card_Category	Categorical	Type of credit card (e.g., Blue, Gold, Platinum, Silver).
Months_on_book	Numeric	Number of months the customer has been on the bank's records (13–56 months).
Total_Relationship_Count	Numeric	Total number of bank products held by the customer (1–6).
Months_Inactive_12_mon	Numeric	Months the customer was inactive in the last year (0–6).
Contacts_Count_12_mon	Numeric	Number of times customer contacted the bank in the past 12 months (0–6).
Credit_Limit	Numeric	Credit limit assigned to the customer (ranges up to ~$34,500).
Total_Revolving_Bal	Numeric	Total balance carried by the customer that accrues interest.
Avg_Open_To_Buy	Numeric	Average remaining credit available to spend (Credit Limit - Revolving Balance).
Total_Amt_Chng_Q4_Q1	Numeric	Change in transaction amount between Q1 and Q4 (relative, not absolute).
Total_Trans_Amt	Numeric	Total amount spent by the customer in a year.
Total_Trans_Ct	Numeric	Total number of transactions in a year.
Total_Ct_Chng_Q4_Q1	Numeric	Change in number of transactions from Q1 to Q4.
Avg_Utilization_Ratio	Numeric	Portion of credit limit used (0 = none, 1 = fully used).
Attrition_Flag	Binary	1 = Customer churned (left); 0 = Still a customer.
CLIENTNUM	Identifier	It is a unique customer ID which no predictive value, can bias model learning.

Variable	Type	Description
`Naive_Bayes_Classifier`	Numeric	Predicted probabilities from a prior model.

Initially we examine the structure and contents of the dataset, confirming that it contains 10,127 customer records with a mix of numerical and categorical features. A detailed variable-by-variable summary and pre-processing plan is provided below.

Variable summary

The dataset contains 10,127 customer records and 23 variables. These features contains customer demographics, account tenure, product relationships, credit behavior and transaction history. Below is a detailed description of each variable.

Redundant and Non-predictive columns

- **CLIENTNUM**: A unique numerical identifier (range: 708,082,083 to 828,343,083). Although unique to each observation, this column contains no predictive information and could introduce misleading importance into tree-based models. It was dropped entirely.
- **Naive_Bayes_Classifier**: These columns store predicted probabilities from a prior Naive Bayes model. Their inclusion would result in target leakage, where the model learns from information that wouldn't be available in a real-world deployment. As such, both were excluded from the feature set.

Target Variable

- **Attrition_Flag**: This is a binary categorical variable indicating whether the customer has churned. It contains two values:
    - `"Existing Customer"` : the customer is still active (encoded as `0`)
    - `"Attrited Customer"` : the customer has closed their account (encoded as `1`)

This variable serves as the **target** in our classification task, where the objective is to predict the likelihood of customer churn based on demographic and behavioral features.

Demographics and Customer profile

- **Customer_Age**: A continuous variable ranging from 26 to 73 and with a mean of 46.3. This distribution is slightly right-skewed but approximately symmetric.
- **Gender**: A binary categorical variable (`Male` and `Female`) where the two categories are nominal and evenly represented.
- **Dependent_count**: represents the number of dependents reported by the customer. Integer variable (range: 0 to 5, mean: 2.35). Distribution shows a right skew , most customers have 1–3 dependents. No transformation was applied due to bounded range and interpretability, but

the feature was standardized to ensure numerical comparability.

- **Education_Level**: Categorical with levels like “High School,” “Graduate,” “Doctorate,” and “Unknown.” While ordinal in logic, the class boundaries are unevenly spaced, and assigning numeric scores would impose artificial linearity. We therefore applied **one-hot encoding**, preserving each level’s independence. “Unknown” was retained, as it could reflect behaviorally distinct customers.
- **Marital_Status**: Nominal categorical variable (`Married` , `Single` , `Divorced` , etc.). Since there’s no logical ranking among these, one-hot encoding was also applied.
- **Income_Category**: Categorical variable with five known income brackets and one “Unknown” group. Distribution is uneven (many customers fall in <$40K brackets). Instead of removing or imputing the “Unknown” values (which may indicate low engagement or privacy-consciousness), we **kept and encoded them** using one-hot encoding to preserve all available signals.
- **Card_Category**: Highly imbalanced categorical variable (most customers have “Blue” cards). With over 90% in a single class, rare categories such as “Platinum” and “Gold” were grouped into an “Other” category. One-hot encoding was then applied to the grouped variable to prevent overfitting due to sparsity.

Account Tenure and Engagement

- **Months_on_book**: is a numeric variable indicating how long (in months) a customer has had a relationship with the bank. It reflects customer tenure and may be useful in assessing loyalty or churn risk. It is a continuous numeric variable (range: 13–56, mean: 35.9).As the distribution is roughly symmetric and centered, no transformation was applied. It was standardized to maintain scale uniformity.
- **Total_Relationship_Count**: is a numeric variable that represents the total number of products or accounts a customer holds with the bank. It reflects the depth of the customer’s relationship and can be a strong indicator of engagement or loyalty. It is an integer count (range: 1–6, mean: 3.81). The distribution is symmetric and values fall within a small range. Standardization was applied to ensure comparability across numeric features.
- **Months_Inactive_12_mon**:is a numeric variable showing how many months (out of the past 12) the customer was inactive, meaning they had no account activity. It reflects disengagement and can be an early behavioral signal of potential churn. It contains integer variables (range: 0–6, mean: 2.34). Despite its low spread, we retained this variable due to its behavioral interpretability (e.g., recent inactivity). Standardization was used due to its numerical scale.
- **Contacts_Count_12_mon**: is a numeric variable indicating how many times the customer contacted the bank in the past 12 months. This can reflect engagement, dissatisfaction or service needs. It contains integer variables (range: 0–6, mean: 2.45). Although slightly skewed, the values are bounded and interpretable, so no transformation was needed. Standardized for modeling.

Credit Behavior

- **Credit_Limit**: is a numeric variable representing the maximum amount of credit the bank has extended to the customer. It reflects the customer’s financial trustworthiness and spending potential. The distribution is strongly right-skewed due to a small group of customers with high limits (range: $1,438 to $34,516, mean: $8,632).

- **Total_Revolving_Bal**: is a numeric variable that represents the customer's outstanding balance, the portion of credit that is carried over month-to-month and accrues interest (range: $0 to $2,517, mean: $1,163). Positively skewed, but not extreme. Kept in raw form and standardized. High balances may signal debt stress, so this variable is expected to be informative.
- **Avg_Open_To_Buy**: is a numeric variable representing the available credit a customer can still spend. It is calculated as the difference between `Credit_Limit` and `Total_Revolving_Bal` (range: $3 to $34,516, mean: $7,469).
- **Avg_Utilization_Ratio**: is a numeric variable represents the proportion of the customer's credit limit that is being used, it is calculated as `Total_Revolving_Bal / Credit_Limit` and ranging from 0.00 to 0.999 (mean: 0.275). As a bounded ratio, this variable was already normalized and did not require transformation. It was kept as-is due to its strong predictive potential (e.g., high utilization may signal financial distress or elevated churn risk).

Transaction Patterns

- **Total_Amt_Chng_Q4_Q1**:is a numeric variable representing the ratio of the customer's total transaction amount in Q4 compared to Q1 (range: 0.000 to 3.397, mean: 0.76). This variable is bounded and interpretable. Due to its numeric stability, no transformation was required.
- **Total_Trans_Amt**: is a numeric variable representing the total dollar amount of all transactions made by the customer in the past year (range: $510 to $18,484, mean: $4,404). The distribution is **heavily right-skewed**, with a small number of high spenders.The variable was standardized and retained due to its strong relevance to customer activity and churn behavior.
- **Total_Trans_Ct**:is a numeric variable that represents the total number of transactions the customer made in the past year (range: 10 to 139, mean: 64.9). The distribution is fairly symmetric and centered, so no transformation was needed. The variable was standardized for modeling purposes and retained due to its strong relevance to customer activity and potential churn patterns.
- **Total_Ct_Chng_Q4_Q1**: is a numeric variable representing the ratio of the number of transactions in Q4 compared to Q1 (range: 0.000 to 3.714, mean: 0.71). As a bounded and interpretable ratio, it required no transformation. The variable was standardized for modeling and retained due to its ability to highlight behavioral shifts that may signal churn risk.

Summary of preprocessing decisions

This detailed preprocessing step ensures:

- Numerical variables are **standardized** to ensure consistency and model efficiency
- Skewed variables (e.g., `Credit_Limit` , `Total_Trans_Amt`) are **log-transformed** to reduce outlier influence
- Categorical variables are **one-hot encoded** to preserve class flexibility without introducing spurious order
- Potential data leakage (via prior model outputs) is proactively **eliminated**
- All steps are justified with actual **distribution metrics and variable ranges**

These steps prepare the dataset for effective use in classification models such as logistic regression, decision trees, and ensemble methods. This gives us a solid starting point for cleaning and transforming the data before modeling.

2.2 Data cleaning and target variable transformation

The dataset was found to be complete, with no missing values, simplifying the pre-processing phase. As it was mentioned before, to prepare the dataset for analysis, we need to take the following steps:

1. **Remove `CLIENTNUM`:** as it is a unique identifier for each customer. It has no predictive value and may introduce noise into the model.
2. **Drop the two Naive Bayes prediction columns:** which are outputs from a past machine learning model and not part of the raw customer information. Keeping them would leak information and bias to our analysis.
3. **Create a new binary column called `Churn`:**
    - Customers who have **left** the credit card service (`Attrition_Flag == "Attrited Customer"`) are labeled as `1` .
    - Customers who are **still active** (`Attrition_Flag == "Existing Customer"`) are labeled as `0` .
4. **Remove the original `Attrition_Flag` column** after creating the binary label.

```
df <- df %>%
  select(-CLIENTNUM, -starts_with("Naive_Bayes")) %>%
  mutate(
    Churn = ifelse(Attrition_Flag == "Attrited Customer", 1, 0)
  ) %>%
  select(-Attrition_Flag)
```

This results in a cleaned dataset where every column can contribute meaningfully to predicting churn, without introducing information leakage or ID noise.

```
str(df)
```

```
## tibble [10,127 × 20] (S3: tbl_df/tbl/data.frame)
##  $ Customer_Age            : num [1:10127] 45 49 51 40 40 44 51 32 37 48 ...
##  $ Gender                  : chr [1:10127] "M" "F" "M" "F" ...
##  $ Dependent_count         : num [1:10127] 3 5 3 4 3 2 4 0 3 2 ...
##  $ Education_Level         : chr [1:10127] "High School" "Graduate" "Graduate" "High School" ...
##  $ Marital_Status          : chr [1:10127] "Married" "Single" "Married" "Unknown" ...
##  $ Income_Category         : chr [1:10127] "$60K - $80K" "Less than $40K" "$80K - $120K" "Less than $40K" ...
##  $ Card_Category           : chr [1:10127] "Blue" "Blue" "Blue" "Blue" ...
##  $ Months_on_book          : num [1:10127] 39 44 36 34 21 36 46 27 36 36 ...
##  $ Total_Relationship_Count: num [1:10127] 5 6 4 3 5 3 6 2 5 6 ...
##  $ Months_Inactive_12_mon  : num [1:10127] 1 1 1 4 1 1 1 2 2 3 ...
##  $ Contacts_Count_12_mon   : num [1:10127] 3 2 0 1 0 2 3 2 0 3 ...
##  $ Credit_Limit            : num [1:10127] 12691 8256 3418 3313 4716 ...
##  $ Total_Revolving_Bal     : num [1:10127] 777 864 0 2517 0 ...
##  $ Avg_Open_To_Buy         : num [1:10127] 11914 7392 3418 796 4716 ...
##  $ Total_Amt_Chng_Q4_Q1    : num [1:10127] 1.33 1.54 2.59 1.4 2.17 ...
##  $ Total_Trans_Amt         : num [1:10127] 1144 1291 1887 1171 816 ...
##  $ Total_Trans_Ct          : num [1:10127] 42 33 20 20 28 24 31 36 24 32 ...
##  $ Total_Ct_Chng_Q4_Q1     : num [1:10127] 1.62 3.71 2.33 2.33 2.5 ...
##  $ Avg_Utilization_Ratio   : num [1:10127] 0.061 0.105 0 0.76 0 0.311 0.066 0.048 0.113 0.144 ...
##  $ Churn                   : num [1:10127] 0 0 0 0 0 0 0 0 0 0 ...
```

2.3 Exploratory Data Analysis (EDA)

Before building any model, it's important to understand the data, its structure, patterns, ranges and any unusual values. This section covers the summary, missing values, feature distributions and detection of potential outliers.

Summary statistics

We start by reviewing the numeric columns. This helps us identify variable ranges, central tendencies and check for abnormalities in value scales.

```
df %>%
  select(where(is.numeric)) %>%
  summary()
```

```
##   Customer_Age   Dependent_count Months_on_book  Total_Relationship_Count
##  Min.   :26.00   Min.   :0.000   Min.   :13.00   Min.   :1.000
##  1st Qu.:41.00   1st Qu.:1.000   1st Qu.:31.00   1st Qu.:3.000
##  Median :46.00   Median :2.000   Median :36.00   Median :4.000
##  Mean   :46.33   Mean   :2.346   Mean   :35.93   Mean   :3.813
##  3rd Qu.:52.00   3rd Qu.:3.000   3rd Qu.:40.00   3rd Qu.:5.000
##  Max.   :73.00   Max.   :5.000   Max.   :56.00   Max.   :6.000
##  Months_Inactive_12_mon Contacts_Count_12_mon  Credit_Limit
##  Min.   :0.000          Min.   :0.000         Min.   : 1438
##  1st Qu.:2.000          1st Qu.:2.000         1st Qu.: 2555
##  Median :2.000          Median :2.000         Median : 4549
##  Mean   :2.341          Mean   :2.455         Mean   : 8632
##  3rd Qu.:3.000          3rd Qu.:3.000         3rd Qu.:11068
##  Max.   :6.000          Max.   :6.000         Max.   :34516
##  Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1 Total_Trans_Amt
##  Min.   :   0        Min.   :    3   Min.   :0.0000       Min.   :  510
##  1st Qu.: 359        1st Qu.: 1324   1st Qu.:0.6310       1st Qu.: 2156
##  Median :1276        Median : 3474   Median :0.7360       Median : 3899
##  Mean   :1163        Mean   : 7469   Mean   :0.7599       Mean   : 4404
##  3rd Qu.:1784        3rd Qu.: 9859   3rd Qu.:0.8590       3rd Qu.: 4741
##  Max.   :2517        Max.   :34516   Max.   :3.3970       Max.   :18484
##  Total_Trans_Ct   Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio     Churn
##  Min.   : 10.00   Min.   :0.0000      Min.   :0.0000        Min.   :0.0000
##  1st Qu.: 45.00   1st Qu.:0.5820      1st Qu.:0.0230        1st Qu.:0.0000
##  Median : 67.00   Median :0.7020      Median :0.1760        Median :0.0000
##  Mean   : 64.86   Mean   :0.7122      Mean   :0.2749        Mean   :0.1607
##  3rd Qu.: 81.00   3rd Qu.:0.8180      3rd Qu.:0.5030        3rd Qu.:0.0000
##  Max.   :139.00   Max.   :3.7140      Max.   :0.9990        Max.   :1.0000
```

From this summary, we can already see:

- `Customer_Age` ranges from **26 to 73**, with a median around **46**, which suggests a mature customer base.
- `Dependent_count` ranges from **0 to 5**, with a mean of about **2.35**, meaning most customers have 1 to 3 dependents.
- `Months_on_book` ranges from **13 to 56**, with a median of **36 months**, indicating most customers have held their card for about 3 years.
- `Credit_Limit` ranges from **$1,438 to $34,516**, showing a significant variation in customer credit access.
- `Total_Revolving_Bal` and `Avg_Open_To_Buy` also range widely, with some customers using their entire limit and others using none at all.
- `Total_Amt_Chng_Q4_Q1` ranges from **0 to 3.397**, capturing change in spending between two quarters, large values may be important churn indicators.
- `Total_Trans_Amt` ranges from **$510 to $18,484**, showing customer spending is highly varied.

- `Total_Ct_Chng_Q4_Q1` has some extreme values like **3.714**, pointing to potential outliers.
- `Avg_Utilization_Ratio` ranges from **0 to 0.999**, showing that some customers are maxing out their credit.
- The target variable `Churn` is **imbalanced**, with a mean of **0.1607**, which means only **~16% of customers churned**.

Missing values check

Although `summary()` didn't show NAs, we explicitly confirm if any column contains missing values.

```
colSums(is.na(df))
```

```
##             Customer_Age                   Gender          Dependent_count
##                        0                        0                        0
##          Education_Level           Marital_Status          Income_Category
##                        0                        0                        0
##            Card_Category           Months_on_book Total_Relationship_Count
##                        0                        0                        0
##   Months_Inactive_12_mon    Contacts_Count_12_mon             Credit_Limit
##                        0                        0                        0
##      Total_Revolving_Bal          Avg_Open_To_Buy     Total_Amt_Chng_Q4_Q1
##                        0                        0                        0
##          Total_Trans_Amt           Total_Trans_Ct      Total_Ct_Chng_Q4_Q1
##                        0                        0                        0
##    Avg_Utilization_Ratio                    Churn
##                        0                        0
```

Now, we can confirm that a result of all zeros in the dataset means that there are no missing values to handle.

EDA: Feature-by-Feature

To gain a deeper understanding of the dataset and lay the groundwork for effective modeling, we begin by exploring each variable individually. This exploratory step allows us to:

- Detect meaningful patterns and trends in the data
- Identify potential data quality issues, such as skewness or class imbalance
- Uncover early signals of variables that may be predictive of customer churn

The analysis is divided into two parts:

- **Categorical variables** are examined using bar plots and frequency tables to highlight dominant categories, rare classes and potential encoding needs.
- **Numerical variables** are explored using histograms and boxplots to assess distribution shapes, presence of outliers, and the need for transformations or scaling.

These insights directly inform subsequent steps such as feature engineering, clustering and model development by revealing the underlying structure and characteristics of each variable.

Univariate Analysis of Categorical Variables

We begin our univariate analysis with the dataset's categorical variables: `Gender`, `Marital_Status`, `Education_Level`, `Income_Category`, `Card_Category` and `Attrition_Flag`.

By visualizing frequency distributions through bar plots, we identify dominant categories, assess class imbalance and reveal potential relationships with customer churn. These insights guide feature encoding decisions and support hypothesis generation for downstream modeling.

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.3
```

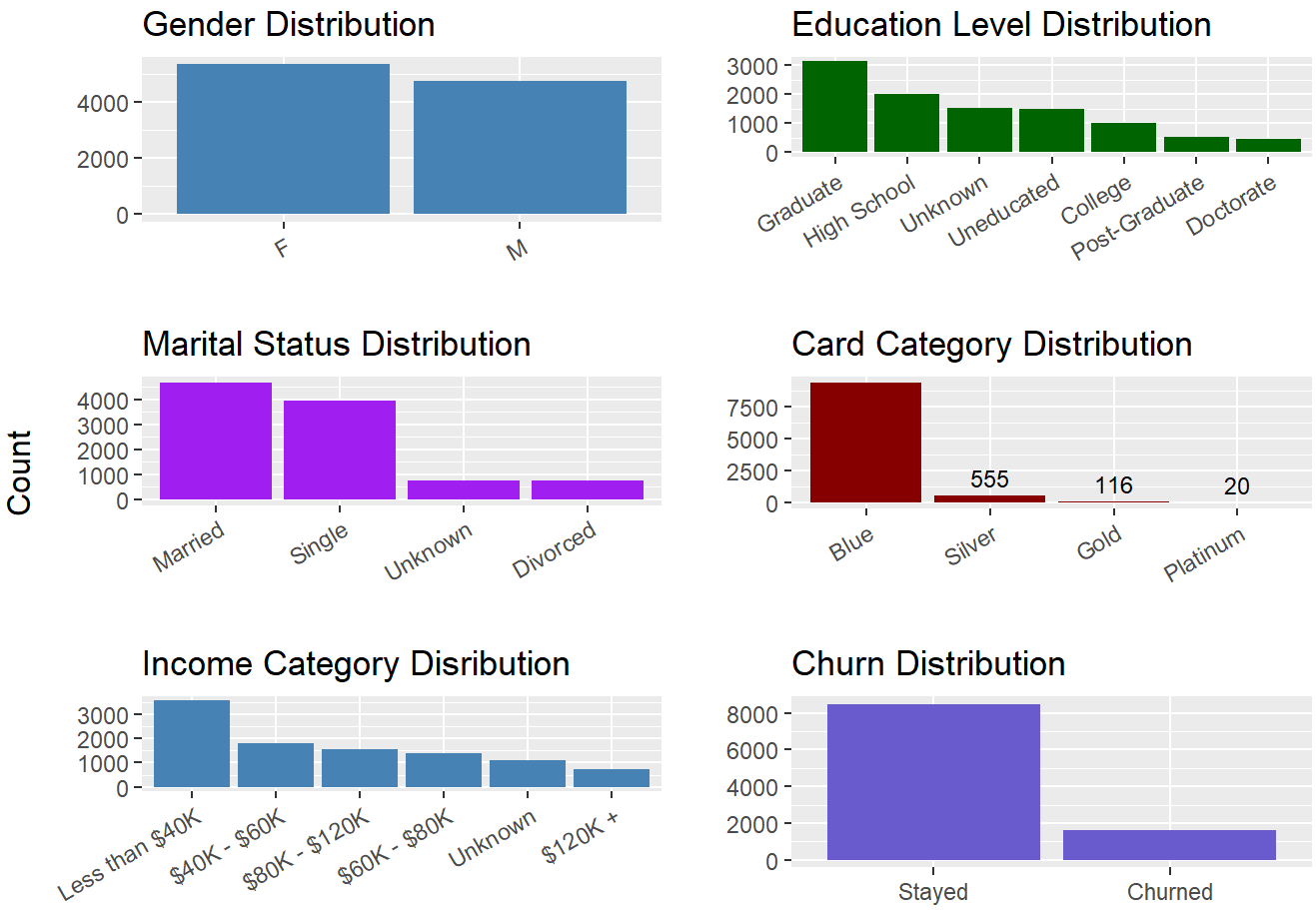
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(grid)
df_sort <- df
df_sort$Education_Level <- fct_infreq(df$Education_Level)
df_sort$Card_Category <- fct_infreq(df$Card_Category)
df_sort$Marital_Status <- fct_infreq(df$Marital_Status)
df_sort$Income_Category <- fct_infreq(df$Income_Category)

p5 <- ggplot(df_sort, aes(x = Gender)) +
  geom_bar(fill = "steelblue") +
  ggtitle("Gender Distribution") +
  xlab("") +
  ylab("") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

p6 <- ggplot(df_sort, aes(x = Education_Level)) +
  geom_bar(fill = "darkgreen") +
  ggtitle("Education Level Distribution") +
  xlab("") +
  ylab("") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

p7 <- ggplot(df_sort, aes(x = Marital_Status)) +
  geom_bar(fill = "purple") +
  ggtitle("Marital Status Distribution") +
  xlab("") +
  ylab("") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

p8 <- ggplot(df_sort, aes(x = Card_Category)) +
  geom_bar(fill = "darkred") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +
  ggtitle("Card Category Distribution") +
  xlab("") +
  ylab("") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

p9 <- ggplot(df_sort, aes(x = Income_Category)) +
  geom_bar(fill = "steelblue") +
  ggtitle("Income Category Disribution") +
```

```
  xlab("") +
  ylab("") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

p10 <- ggplot(df_sort, aes(x = factor(Churn, labels = c("Stayed", "Churned")))) +
  geom_bar(fill = "slateblue") +
  ggtitle("Churn Distribution") +
  xlab("") +
  ylab("") +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5))
# Plot in 2x2 grid
grid.arrange(
  arrangeGrob(p5, p6, p7, p8, p9, p10, ncol = 2),
  left = textGrob("Count", rot = 90, vjust = 1, gp = gpar(fontsize = 12))
)
```

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



From the graphs above it can be observed that:

- **Gender** distribution is relatively balanced, with a slight majority of female customers.
- **Education_Level** shows a concentration of customers with Graduate and High School education, while advanced degrees (Post-Graduate and Doctorate) are less common.
- **Marital Status**: Majority are married or single, with low representation from divorced customers.
- **Card_Category** is heavily dominated by the “Blue” category and other tiers (Silver, Gold, Platinum) are rare and may contribute limited variance.
- **Income Category** distribution is skewed toward lower income groups, with most customers earning less than $40K.
- **Attrition Flag** distribution confirms a class imbalance, with a significantly larger number of active (non-churned) customers.

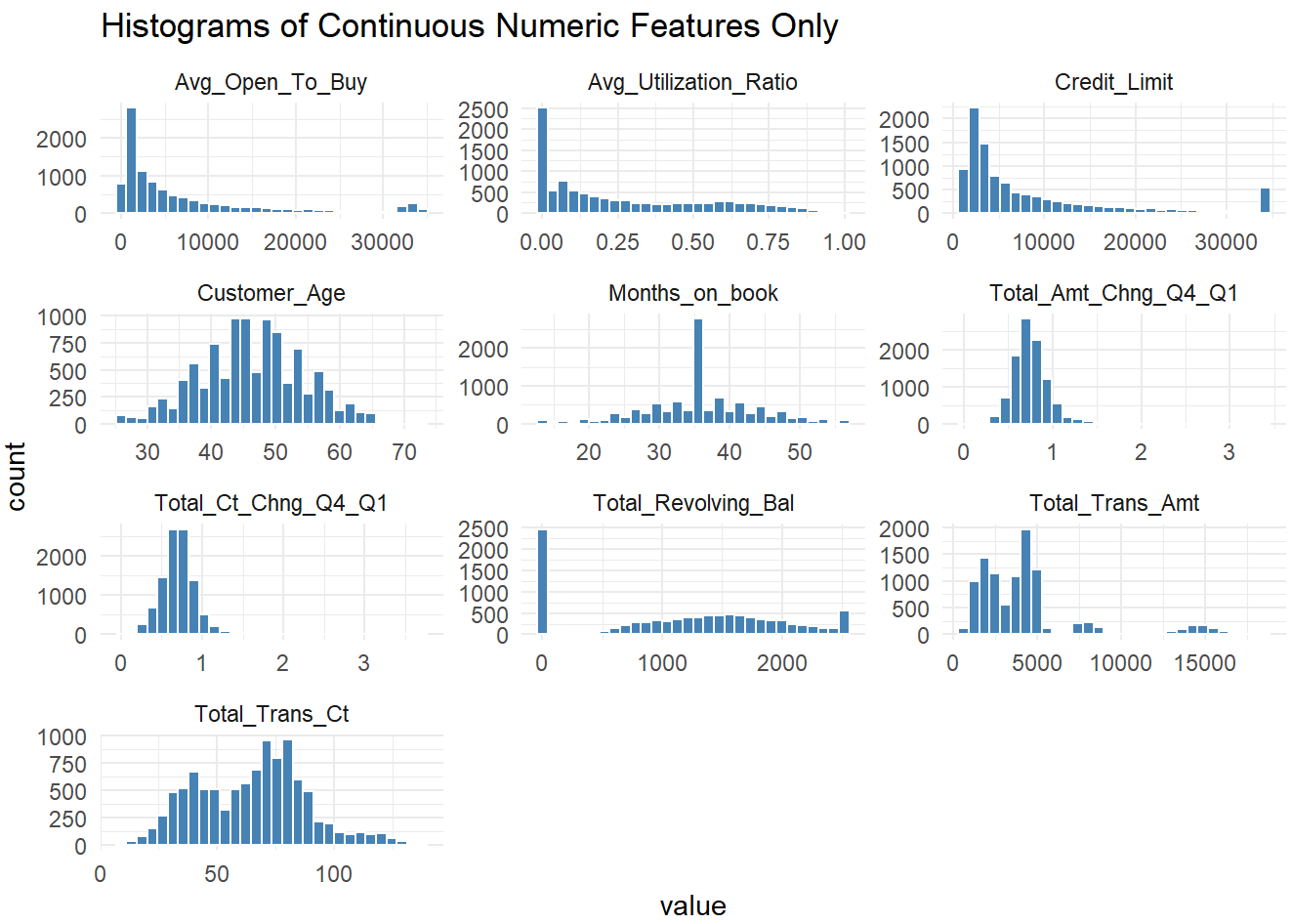
These distributions reveal imbalances and sparsity in several features, which may influence both predictive power and model bias.

Univariate Analysis of Numerical Variables

Next, we examine the distribution of selected numeric features using histograms and density plots. Understanding the shape of these distributions helps determine whether variables are normally distributed, skewed, or contain unusual spikes.

This step is critical for guiding future transformations such as normalization or log-scaling, and for selecting models that assume or benefit from certain distribution characteristics.

```
df %>%
  select(
    Customer_Age,
    Months_on_book,
    Credit_Limit,
    Avg_Open_To_Buy,
    Total_Revolving_Bal,
    Total_Amt_Chng_Q4_Q1,
    Total_Trans_Amt,
    Total_Trans_Ct,
    Total_Ct_Chng_Q4_Q1,
    Avg_Utilization_Ratio
  ) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white") +
  facet_wrap(~variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Histograms of Continuous Numeric Features Only")
```



Observations:

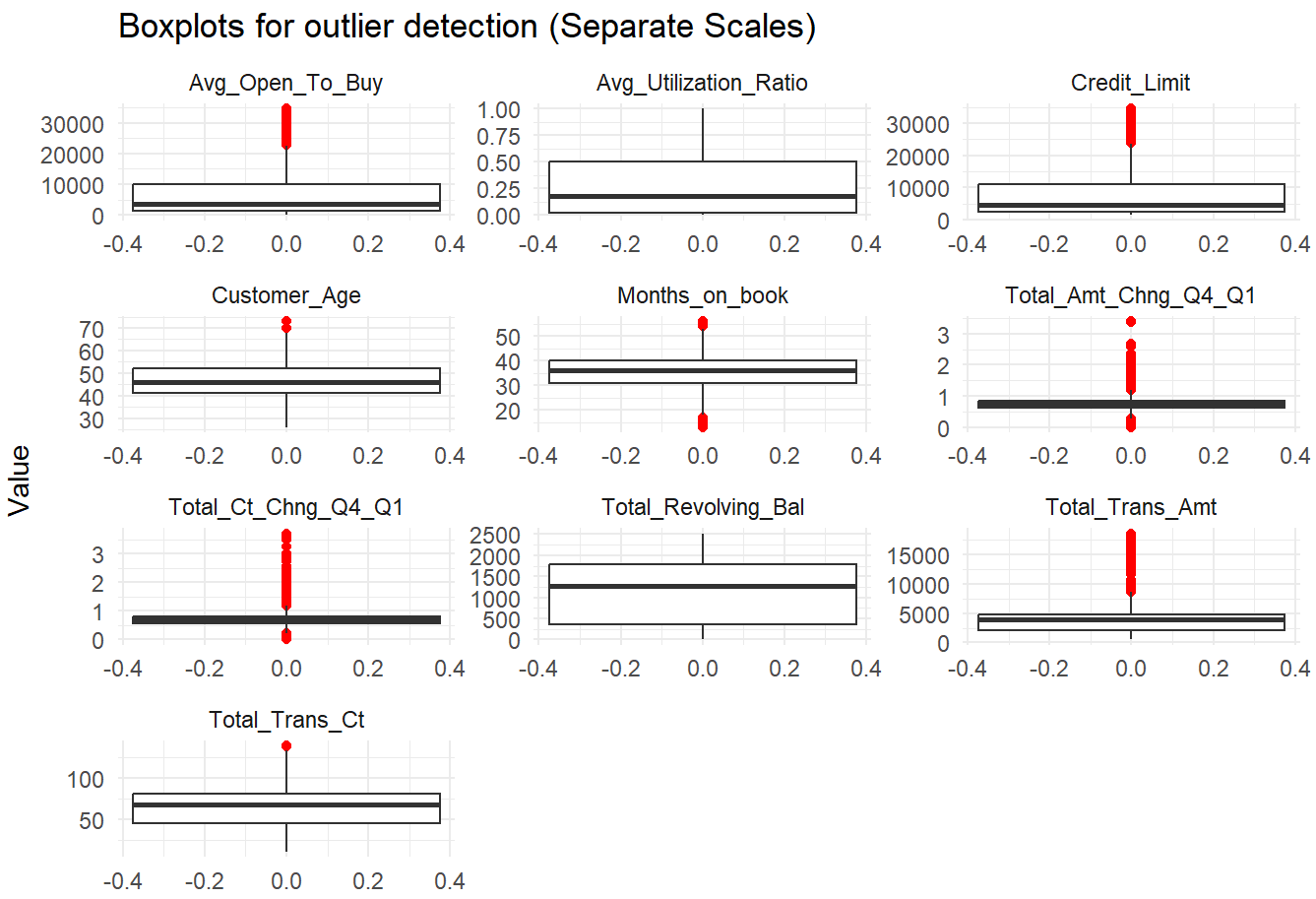
- `Customer_Age` is nearly normal, centered around 46 years, with minor left skew.
- `Months_on_book` is mostly symmetric but with a **noticeable spike** at 36, likely a common tenure.
- `Credit_Limit` and `Avg_Open_To_Buy` are **highly right-skewed**, many customers have small limits while a few have very high credit ceilings.
- `Total_Revolving_Bal` shows a large number of customers at 0 balance, followed by a flat spread, this indicates many customers pay off their full balance.
- `Total_Amt_Chng_Q4_Q1` and `Total_Ct_Chng_Q4_Q1` both show a **concentrated unimodal peak**, meaning most customers fall in a narrow range of quarter-over-quarter change.
- `Total_Trans_Amt` and `Total_Trans_Ct` have a **bi-modal or clustered shape**, indicating different usage groups (e.g. low vs high spenders).
- `Avg_Utilization_Ratio` is heavily right-skewed with a **long tail**, showing most customers use a small portion of their limit, but a few max it out.

These distribution insights will guide appropriate pre-processing steps and inform model selection.

Outlier Detection

To identify potential anomalies or extreme values that could distort model performance, we inspect all numerical variables using boxplots. Outliers are visualized as data points that fall significantly outside the interquartile range.

```
df %>%
  select(
    Customer_Age,
    Months_on_book,
    Credit_Limit,
    Avg_Open_To_Buy,
    Total_Revolving_Bal,
    Total_Amt_Chng_Q4_Q1,
    Total_Trans_Amt,
    Total_Trans_Ct,
    Total_Ct_Chng_Q4_Q1,
    Avg_Utilization_Ratio
  ) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
  ggplot(aes(y = value)) +
  geom_boxplot(outlier.color = "red") +
  facet_wrap(~variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Boxplots for outlier detection (Separate Scales)", x = "", y = "Value")
```



From the boxplots above, we observe that several numeric features contain notable outliers:

- `Credit_Limit` , `Avg_Open_To_Buy` , and `Total_Trans_Amt` show strong positive skew with many high-end outliers.
- `Total_Amt_Chng_Q4_Q1` and `Total_Ct_Chng_Q4_Q1` also show concentration around low values, with a few unusually high changes.
- `Customer_Age` and `Months_on_book` contain milder outliers, likely due to edge cases (very old or new customers).

At this stage, we **retain all outliers**, as they may represent valid high-activity or high-risk customer behavior. However, their presence will be considered when applying scaling, clustering, or linear models, where sensitivity to extreme values could affect performance.

Summary of Data Preparation

At this stage, we have conducted a comprehensive review and cleaning of the dataset to ensure it is suitable for downstream modeling and analysis:

- **Numeric features** were analyzed through histograms and boxplots to understand their distributions, value ranges, and outlier behavior. Several variables (e.g., `Credit_Limit`, `Total_Trans_Amt`) exhibit right skew and high-end outliers, which were retained under the assumption that they reflect genuine customer behavior.
- **Categorical features** were visualized using bar plots to assess dominant categories, class imbalance and sparsity. These findings will inform appropriate encoding strategies during feature engineering.
- The dataset contained **no missing values**, eliminating the need for imputation and reducing potential preprocessing bias.
- **Non-informative or leakage-prone variables** such as the unique identifier `CLIENTNUM` and pre-generated Naive Bayes scores were removed to prevent overfitting and maintain model integrity.
- The target variable `Attrition_Flag` was recoded into a binary variable (`Churn`) to clearly frame the task as a binary classification problem.

These data preparation steps ensure a strong foundation for downstream tasks such as feature engineering, dimensionality reduction, clustering, and predictive modeling.

2.4 Bivariate Analysis: Churn Relationships

This section explores how customer features relate to churn, providing insight into which variables are likely to be predictive in classification modeling

Categorical Variable Counts

We now explore the **categorical variables** and how they relate to churn.

```
df %>%
  select(Gender, Education_Level, Marital_Status, Income_Category, Card_Category) %>%
  map(~as.data.frame(table(.)))
```

```
## $Gender
##   . Freq
## 1 F 5358
## 2 M 4769
##
## $Education_Level
##               . Freq
## 1       College 1013
## 2     Doctorate  451
## 3      Graduate 3128
## 4   High School 2013
## 5 Post-Graduate  516
## 6    Uneducated 1487
## 7       Unknown 1519
##
## $Marital_Status
##          . Freq
## 1 Divorced  748
## 2  Married 4687
## 3   Single 3943
## 4  Unknown  749
##
## $Income_Category
##                . Freq
## 1        $120K +  727
## 2    $40K - $60K 1790
## 3    $60K - $80K 1402
## 4   $80K - $120K 1535
## 5 Less than $40K 3561
## 6        Unknown 1112
##
## $Card_Category
##          . Freq
## 1     Blue 9436
## 2     Gold  116
## 3 Platinum   20
## 4   Silver  555
```

These tables reveal useful patterns:

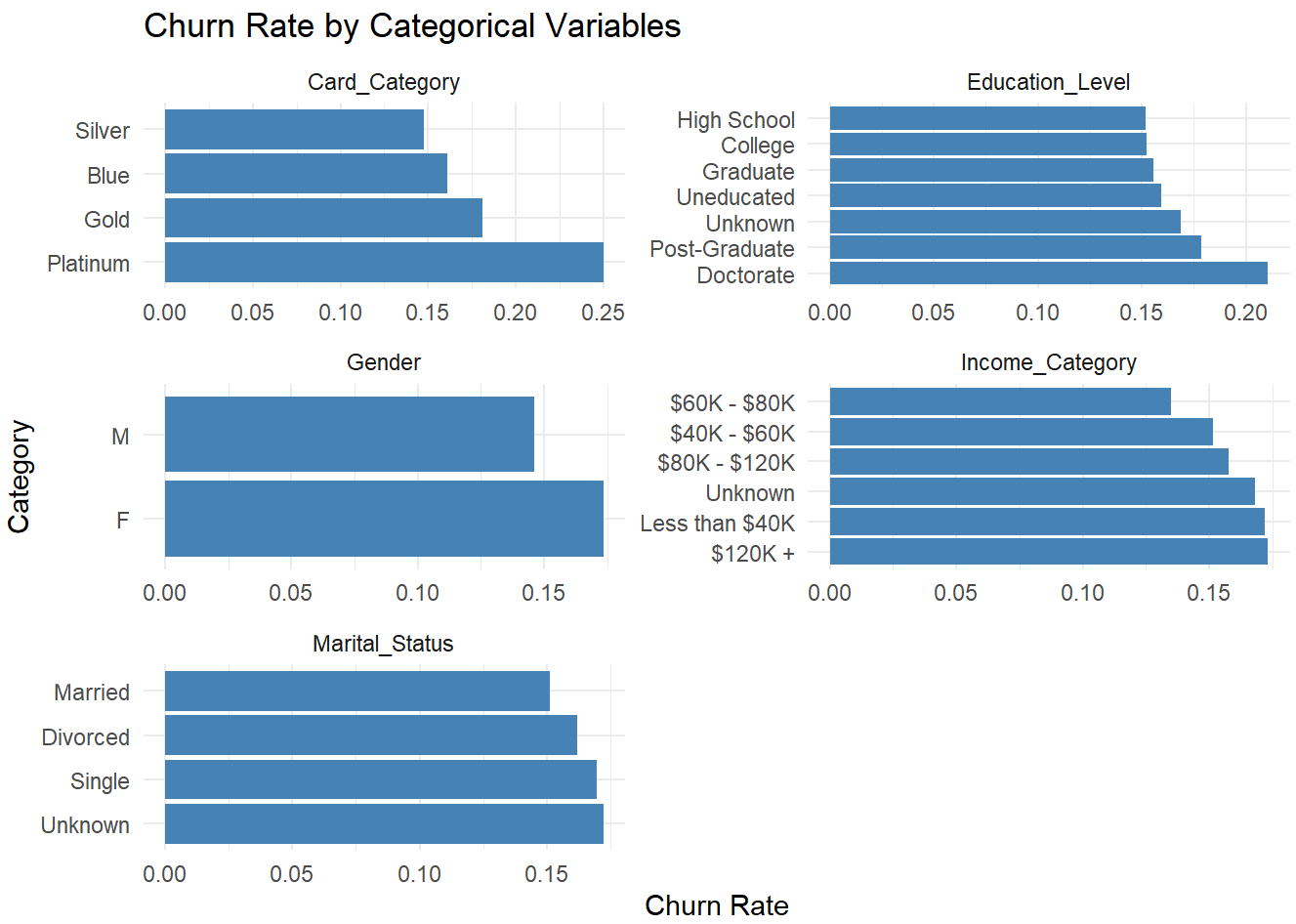
- `Gender` is fairly balanced, with slightly more female customers (53%).
- `Education_Level` shows most users are “Graduate” or “High School” educated; 15% are “Unknown.”
- `Marital_Status` is dominated by “Married” and “Single”, while “Divorced” and “Unknown” are minor.
- `Income_Category` shows a skew toward “Less than $40K”; 11% have “Unknown” income.
- `Card_Category` is extremely imbalanced because 93% of customers hold “Blue” cards; other types are rare but may indicate VIP customers.

Churn Rate by Categorical Variables

To explore how churn behavior varies across different customer segments, we visualize churn rates by category for key features such as `Education_Level`, `Marital_Status`, `Income_Category`, `Gender`, and `Card_Category`.

These bar plots highlight whether specific subgroups (e.g., “Uneducated”, “Single”, or “Platinum” cardholders) are more likely to churn. This analysis helps identify early signals of churn prone segments and informs feature selection for modeling.

```
df %>%
  select(Gender, Education_Level, Marital_Status, Income_Category, Card_Category, Churn) %>%
  pivot_longer(-Churn, names_to = "variable", values_to = "category") %>%
  group_by(variable, category) %>%
  summarise(
    churn_rate = mean(Churn),
    count = n(),
    .groups = "drop"
  ) %>%
  ggplot(aes(x = reorder(category, -churn_rate), y = churn_rate)) +
  geom_col(fill = "steelblue") +
  facet_wrap(~variable, scales = "free", ncol = 2) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Churn Rate by Categorical Variables", x = "Category", y = "Churn Rate")
```



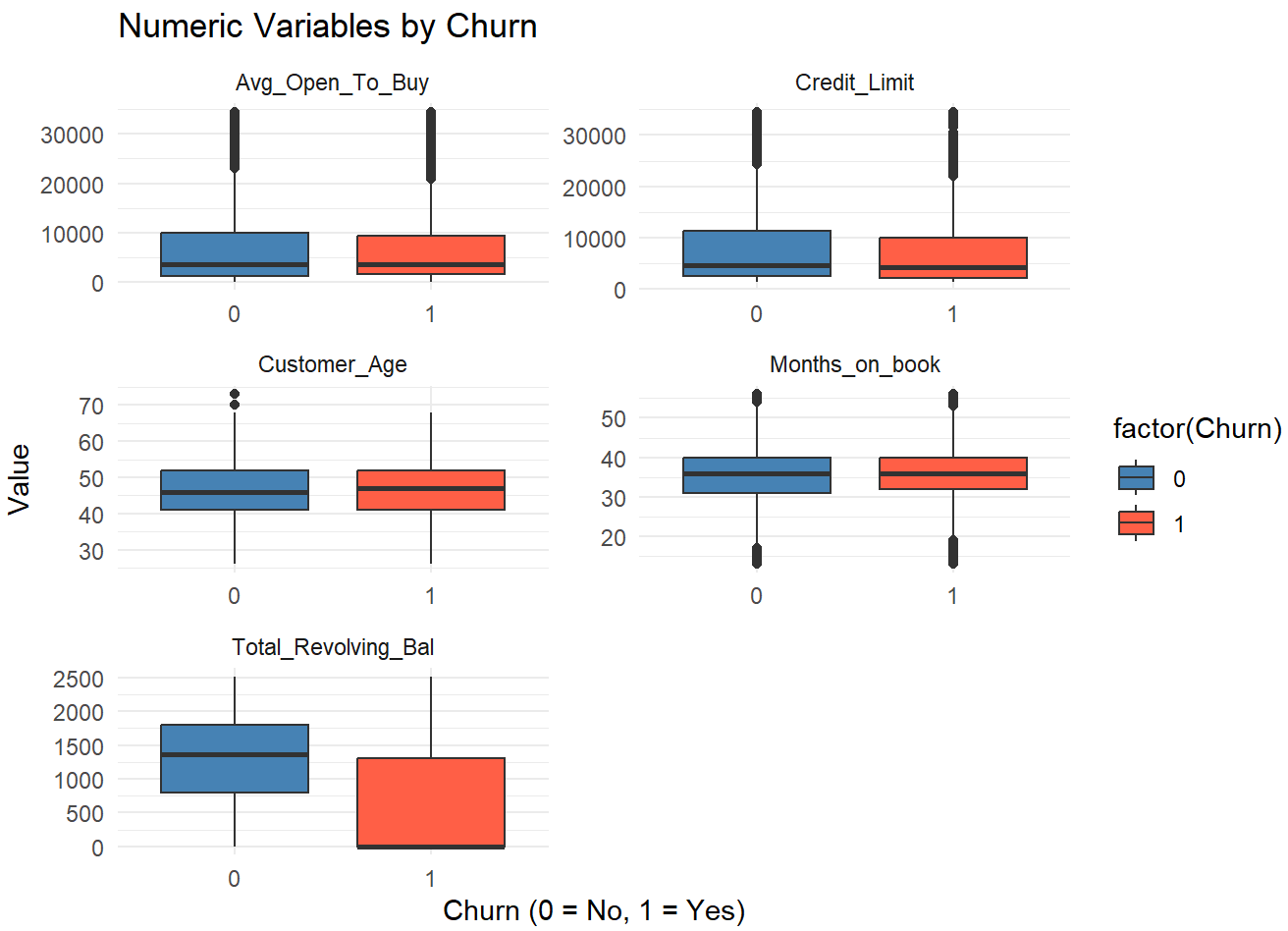
The churn rate plots reveal several important patterns:

- **Card Category**: Customers with **Platinum cards churn at approximately 25%**, the highest among all categories. **Gold** cardholders churn at around **17%**, while **Blue (~15%)** and **Silver (~14%)** customers show the lowest churn. This may indicate higher dissatisfaction or unmet expectations in premium tiers.
- **Education Level**: Churn is highest among **Doctorate holders (~21%)** and **Post-Graduate (~18%)** customers. In contrast, those with **College (~14%)**, **High School (~13%)**, and **Uneducated (~15%)** backgrounds exhibit lower churn, suggesting that churn risk does not linearly follow education level.
- **Gender**: **Female customers churn at ~17%**, slightly higher than **male customers (~15%)**.
- **Income Category**: The **highest churn is seen in the $120K+ group (~18%)**, followed closely by **< $40K (~17%)** and **Unknown (~16%)**. In contrast, **middle-income segments ($60K–$80K)** show lower churn rates around **13–14%**, indicating better retention.
- **Marital Status**: **Single customers churn at ~16%**, followed by **Unknown (~17%)**. **Married customers churn at the lowest rate (~13%)**, which may reflect greater financial stability or longer-term loyalty.

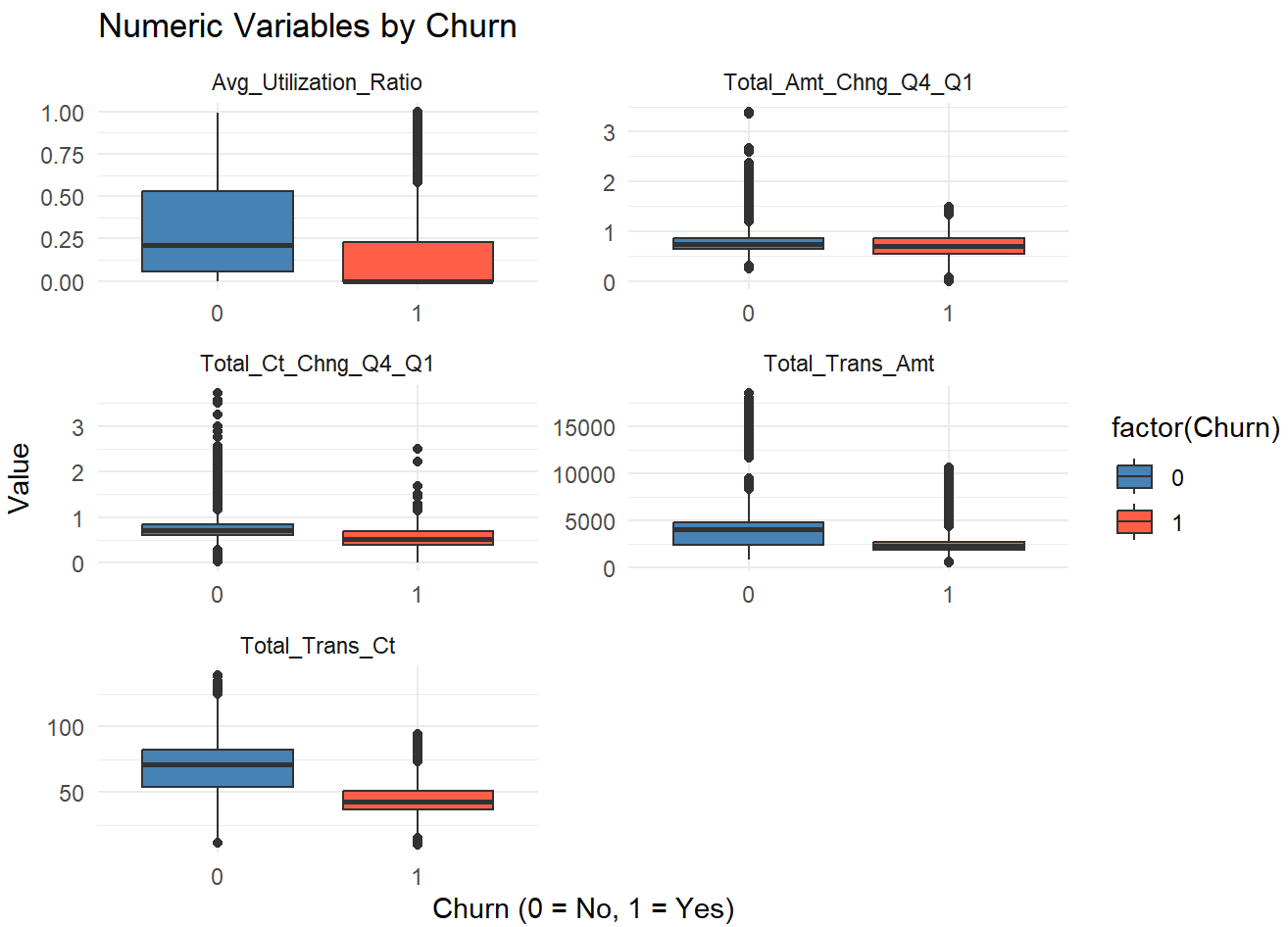
Churn Comparison for Numeric Features

To explore how numerical variables differ between churned and non-churned customers, we use boxplots segmented by the binary `Churn` variable. This helps identify features that may have strong predictive relationships with customer attrition.

```
df %>%
select(
  Churn,
  Customer_Age,
  Months_on_book,
  Credit_Limit,
  Avg_Open_To_Buy,
  Total_Revolving_Bal,
) %>%
pivot_longer(-Churn, names_to = "variable", values_to = "value") %>%
ggplot(aes(x = factor(Churn), y = value, fill = factor(Churn))) +
geom_boxplot() +
facet_wrap(~variable, scales = "free", ncol = 2) +
theme_minimal() +
labs(title = "Numeric Variables by Churn", x = "Churn (0 = No, 1 = Yes)", y = "Value") +
scale_fill_manual(values = c("0" = "steelblue", "1" = "tomato"))
```


Numeric Variables by Churn
Avg_Open_To_Buy
Credit_Limit
Customer_Age
Months_on_book
Total_Revolving_Bal
Value
Churn (0 = No, 1 = Yes)
factor(Churn)
0
1

```
df %>%
 select(
   Churn,
   Total_Amt_Chng_Q4_Q1,
   Total_Trans_Amt,
   Total_Trans_Ct,
   Total_Ct_Chng_Q4_Q1,
   Avg_Utilization_Ratio
 ) %>%
 pivot_longer(-Churn, names_to = "variable", values_to = "value") %>%
 ggplot(aes(x = factor(Churn), y = value, fill = factor(Churn))) +
 geom_boxplot() +
 facet_wrap(~variable, scales = "free", ncol = 2) +
 theme_minimal() +
 labs(title = "Numeric Variables by Churn", x = "Churn (0 = No, 1 = Yes)", y = "Value") +
 scale_fill_manual(values = c("0" = "steelblue", "1" = "tomato"))
```



The numeric variable boxplots reveal key churn related differences:

- **Avg_Utilization_Ratio**: Churned customers tend to have lower utilization ratios, suggesting reduced credit engagement before leaving.
- **Credit_Limit**: Churners generally have lower credit limits. They may represent lower-tier customers with less financial flexibility and weaker institutional loyalty.
- **Customer_Age**: Churned customers are slightly older on average, possibly reflecting a late-stage decision to exit or less inclination to adapt to new offerings.
- **Months_on_book**: Churners typically have shorter relationships with the bank, reinforcing the idea that **newer customers churn faster**, potentially due to unmet expectations or poor onboarding.
- **Total_Trans_Amt**: One of the strongest signals, churners spend significantly less than loyal customers, making this a highly predictive behavioral feature.
- **Avg_Open_To_Buy**: Churners show a lower average available credit, which may be tied to both lower credit limits and less willingness to spend. High-value, high-capacity customers appear more stable.

- **Total_Revolving_Bal**: Non-churners carry higher revolving balances, indicating consistent use of credit. Churners often maintain low or zero balances, reflecting disengagement.
- **Total_Ct_Chng_Q4_Q1**: Churners display limited change in transaction counts across quarters. This lack of growth may reflect fading activity and a transition toward disengagement.
- **Total_Trans_Ct**: Churned customers consistently make fewer transactions, indicating declining usage patterns prior to attrition.
- **Total_Amt_Chng_Q4_Q1**: This quarterly change in amount is lower for churners, implying reduced financial activity growth compared to loyal clients.

These differences will inform our feature selection process and highlight which variables may provide meaningful signal when building classification models.

Correlation Analysis of Numeric Variables

To better understand the relationships between numeric features, we visualize the **Pearson correlation matrix** using a heatmap. This analysis helps to:

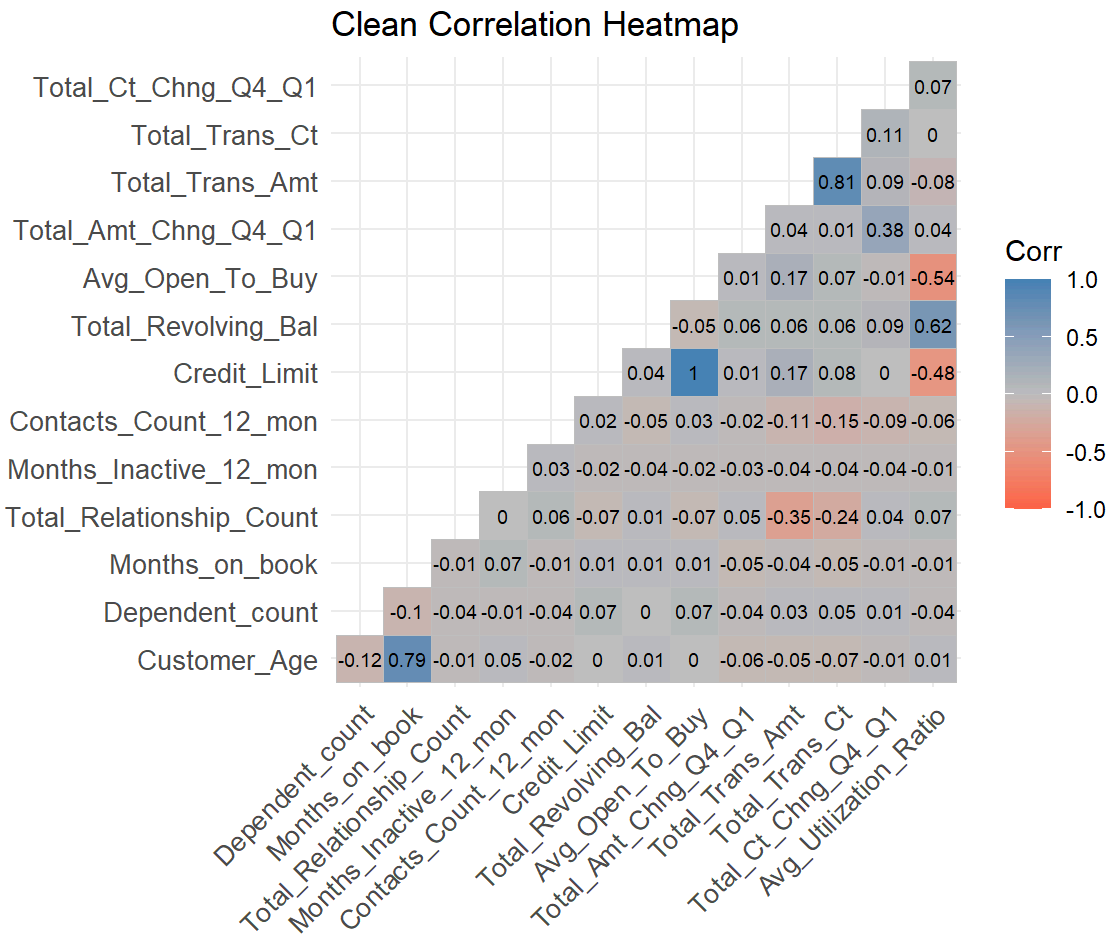
- **Detect multicollinearity**, which can negatively impact linear models and distort variable importance.
- **Identify redundant features** that may carry overlapping information.

This heatmap provides a structured overview of which variables are closely related and which are largely independent. Such insights guide both **feature selection** and **model design** decisions later in the analysis pipeline.

```
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.4.3
```

```
cor_matrix <- df %>%
  select(where(is.numeric)) %>%
  select(-Churn) %>%
  cor(method = "pearson")

ggcorrplot(cor_matrix,
           type = "lower",
           method = "square",
           lab = TRUE,
           lab_size = 2.5,
           tl.cex = 10,
           title = "Clean Correlation Heatmap",
           colors = c("tomato", "gray", "steelblue"),
           ggtheme = theme_minimal())
```



Correlation Insights:

- **Total_Trans_Ct** and **Total_Trans_Amt** show the strongest positive correlation (0.81), which makes sense because more transactions usually mean higher total amounts.
- **Credit_Limit** is strongly correlated (0.95) with **Avg_Open_To_Buy**, which is expected since open credit is derived from total limit minus usage.
- **Avg_Utilization_Ratio** is negatively correlated with **Avg_Open_To_Buy** (-0.54), meaning high utilization typically reduces available credit.
- Most other variables are weakly correlated (< 0.4), suggesting limited multicollinearity across features. This supports the idea that **dimensionality reduction (PCA)** may not be strictly necessary, but could still be beneficial to simplify the feature space and improve model generalization.

Variables excluded from analysis

The following analysis explores four count based numerical variables: `Dependent_count` , `Total_Relationship_Count` , `Months_Inactive_12_mon` , and `Contacts_Count_12_mon` . These variables were initially evaluated for inclusion in the churn prediction model. While ultimately excluded from the final feature set, due to low predictive power, redundancy or inconsistent relationships with churn, they still provide valuable insight into customer behaviors and interaction patterns.

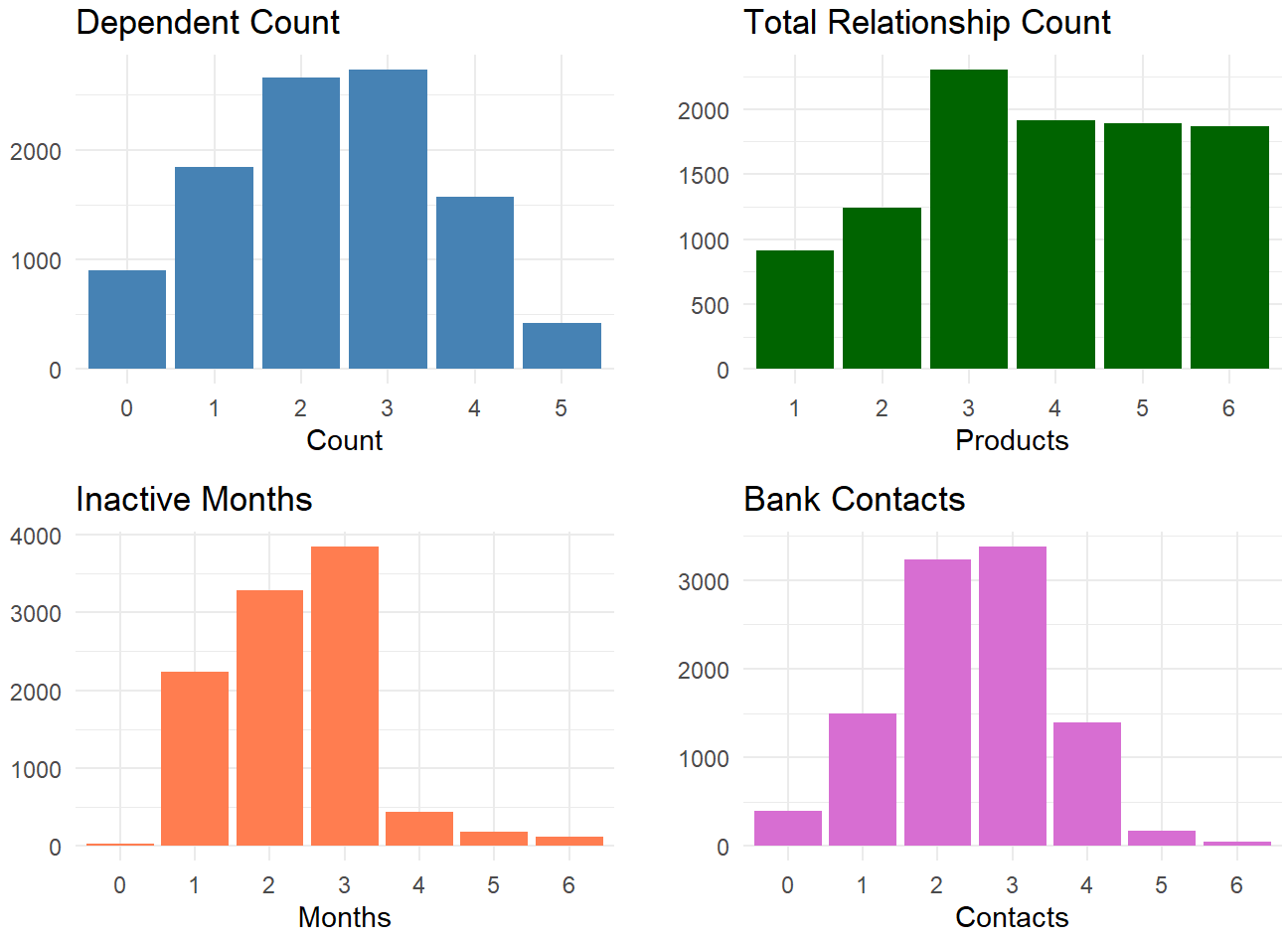
```
p1 <- ggplot(df, aes(x = factor(df$Dependent_count))) +
  geom_bar(fill = "steelblue") +
  labs(title = "Dependent Count", x = "Count", y = "") +
  theme_minimal()

p2 <- ggplot(df, aes(x = factor(df$Total_Relationship_Count))) +
  geom_bar(fill = "darkgreen") +
  labs(title = "Total Relationship Count", x = "Products", y = "") +
  theme_minimal()

p3 <- ggplot(df, aes(x = factor(df$Months_Inactive_12_mon))) +
  geom_bar(fill = "coral") +
  labs(title = "Inactive Months", x = "Months", y = "") +
  theme_minimal()

p4 <- ggplot(df, aes(x = factor(df$Contacts_Count_12_mon))) +
  geom_bar(fill = "orchid") +
  labs(title = "Bank Contacts", x = "Contacts", y = "") +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```

```
## Warning: Use of `df$Dependent_count` is discouraged.
## ℹ Use `Dependent_count` instead.
```

```
## Warning: Use of `df$Total_Relationship_Count` is discouraged.
## ℹ Use `Total_Relationship_Count` instead.
```

```
## Warning: Use of `df$Months_Inactive_12_mon` is discouraged.
## ℹ Use `Months_Inactive_12_mon` instead.
```

```
## Warning: Use of `df$Contacts_Count_12_mon` is discouraged.
## ℹ Use `Contacts_Count_12_mon` instead.
```



Observations:

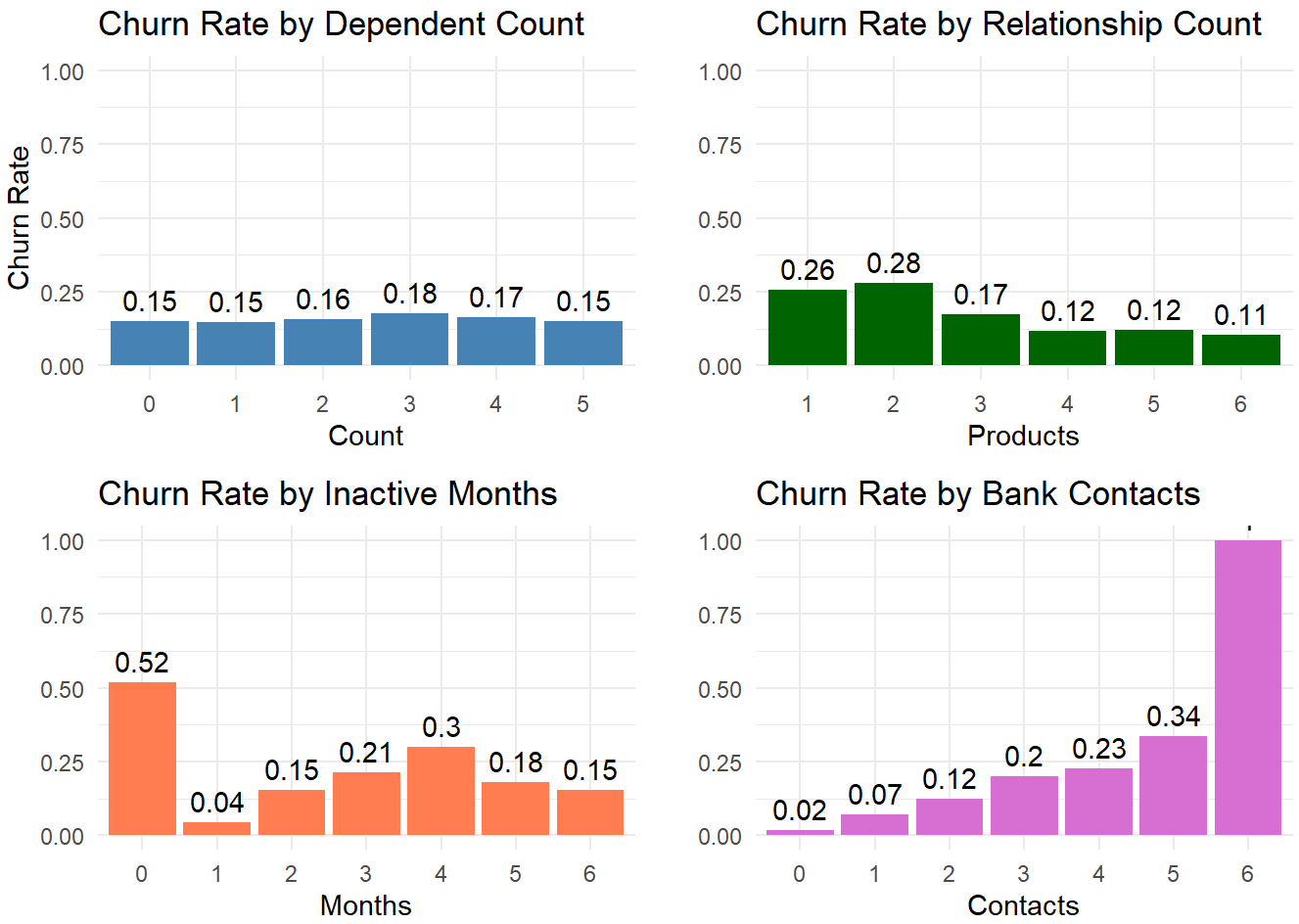
- **Dependent Count**: This variable captures how many dependents a customer has. Most customers report having 2–3 dependents, while counts of 0 or 5 are relatively rare. This distribution suggests moderate variation in family size, but clustering around the center could limit its discriminative power.

- **Total Relationship Count**: This refers to the number of bank products held by each customer. The distribution is skewed toward 3–6 products, indicating that most customers have a moderate-to-high level of engagement. Very few have only one product.
- **Inactive Months**:Most customers were inactive for 2 to 3 months in the past year, with a sharp drop after month 3. This suggests a fairly consistent level of engagement. The long tail may represent a small group of highly disengaged customers.
- **Bank Contacts**:Contact counts are tightly centered around 2 to 3 contacts, with very few customers having 5 or more. The right tail is long but sparse, possibly indicating customers who required more attention or support.

To better understand why these variables were excluded, we examine how churn rates vary across their values. This helps assess whether they show meaningful or consistent patterns relevant to churn behavior.

```
# Dependent Count
p1 <- df %>%
  group_by(Dependent_count) %>%
  summarise(Churn_Rate = mean(Churn), n = n()) %>%
  ggplot(aes(x = factor(Dependent_count), y = Churn_Rate)) +
  geom_col(fill = "steelblue") +
  geom_text(aes(label = round(Churn_Rate, 2)), vjust = -0.5) +
  labs(title = "Churn Rate by Dependent Count", x = "Count", y = "Churn Rate") +
  theme_minimal() +
  ylim(0, 1)

# Total Relationship Count
p2 <- df %>%
  group_by(Total_Relationship_Count) %>%
  summarise(Churn_Rate = mean(Churn), n = n()) %>%
  ggplot(aes(x = factor(Total_Relationship_Count), y = Churn_Rate)) +
  geom_col(fill = "darkgreen") +
  geom_text(aes(label = round(Churn_Rate, 2)), vjust = -0.5) +
  labs(title = "Churn Rate by Relationship Count", x = "Products", y = "") +
  theme_minimal() +
  ylim(0, 1)

# Inactive Months
p3 <- df %>%
  group_by(Months_Inactive_12_mon) %>%
  summarise(Churn_Rate = mean(Churn), n = n()) %>%
  ggplot(aes(x = factor(Months_Inactive_12_mon), y = Churn_Rate)) +
  geom_col(fill = "coral") +
  geom_text(aes(label = round(Churn_Rate, 2)), vjust = -0.5) +
  labs(title = "Churn Rate by Inactive Months", x = "Months", y = "") +
  theme_minimal() +
  ylim(0, 1)

# Bank Contacts
p4 <- df %>%
  group_by(Contacts_Count_12_mon) %>%
  summarise(Churn_Rate = mean(Churn), n = n()) %>%
  ggplot(aes(x = factor(Contacts_Count_12_mon), y = Churn_Rate)) +
  geom_col(fill = "orchid") +
```

```
  geom_text(aes(label = round(Churn_Rate, 2)), vjust = -0.5) +
  labs(title = "Churn Rate by Bank Contacts", x = "Contacts", y = "") +
  theme_minimal() +
  ylim(0, 1)

# Combine all plots
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



Observations:

- **Dependent Count**: Churn rates remain flat across all categories (≈15–18%), suggesting that the number of dependents has *minimal explanatory power* for predicting churn. This aligns with its low feature importance during model tuning.

- **Total Relationship Count**: There is a modest *inverse relationship* where churn rates decrease steadily from 28% at 2 products to 11% at 6 products. This suggests higher engagement may lower churn risk. However, the effect is mild and possibly *redundant with other engagement indicators*, which could explain its lower ranking in final feature selection.
- **Inactive Months**: Customers with **0 inactive months** exhibit a surprisingly high churn rate (52%), which sharply drops thereafter. This counterintuitive pattern may be explained by *noise or a misinterpreted behavioral signal* (for example one-time logins). Due to its *non-linear and unstable trend*, this variable may confuse models unless treated carefully (e.g. binned or transformed), justifying its exclusion.
- **Bank Contacts**: This variable shows a *clear positive trend*, with churn rate increasing from 2% (0 contacts) to nearly 100% at 6 contacts. This suggests that *frequent contact may be a signal of dissatisfaction*, but it could also be *post-churn outreach* (for example reverse causality). While this makes it interpretable, its *directionality is ambiguous* and may confound predictive models.

These variables show interesting churn rate patterns, but due to *flat relationships*, *non-monotonic behavior* or *confounding effects*, they were not retained in the final model. Their inclusion here enhances the *transparency* of feature selection decisions and provides a fuller understanding of customer behaviors.

Summary of Exploratory Data Analysis (EDA)

We explored all features in the dataset to understand their nature, range, and relationship with churn.

Target Variable (`Churn`)

- The target is binary: 1 = churned, 0 = retained.
- The dataset is imbalanced: ~16% of customers have churned.
- This class imbalance will be addressed during modeling.

Numeric Variables

- **Customer_Age**: Slightly older customers (median ~46) tend to churn more.
- **Credit_Limit** and **Avg_Open_To_Buy**: Right-skewed; churners generally have lower limits and less available credit.
- **Total_Trans_Amt** and **Total_Trans_Ct**: Strong predictors, churners spend and transact less.
- **Avg_Utilization_Ratio**: Lower for churners, suggesting reduced credit engagement before churn.
- **Months_on_book**: Churners often have shorter tenure with the bank.

No missing values were found, and outliers were reviewed but retained as they reflect genuine customer behaviors. Correlation analysis showed no severe multicollinearity, with the exception of expected relationships (e.g., between Credit_Limit and Avg_Open_To_Buy).

Categorical Variables

- **Gender**: Fairly balanced; females show slightly higher churn.
- **Education_Level**: Lower education and “Unknown” levels are linked to higher churn.
- **Marital_Status**: Single and unknown status customers churn more; married customers churn less.
- **Income_Category**: Lower-income groups and “Unknown” have higher churn risk.

- **Card_Category**: “Platinum” and “Gold” cardholders churn more, though “Blue” dominates in frequency (93%).

All categorical variables are clean with no NAs, though many include “Unknown” levels. These were retained for analysis, as they may capture meaningful business signals.

> Overall, the EDA shows that churn is associated with **lower usage**, **shorter relationships** and **less financial engagement**. These patterns will guide both feature engineering and model selection in the next steps.

2.5 Data Transformation

Categorical Encoding

To prepare the dataset for machine learning, we converted all categorical variables into numerical format using one-hot encoding:

- Applied `fastDummies::dummy_cols()` to create dummy variables
- Dropped the first level of each to avoid multicollinearity
- Retained all “Unknown” levels as they may carry predictive value

We’re using `fastDummmies` library to do this.

```
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.4.3
```

```
df_encoded <- df %>%
  fastDummies::dummy_cols(remove_first_dummy = TRUE, remove_selected_columns = TRUE)
```

```
names(df_encoded)
```

```
##  [1] "Customer_Age"                  "Dependent_count"
##  [3] "Months_on_book"                "Total_Relationship_Count"
##  [5] "Months_Inactive_12_mon"        "Contacts_Count_12_mon"
##  [7] "Credit_Limit"                  "Total_Revolving_Bal"
##  [9] "Avg_Open_To_Buy"               "Total_Amt_Chng_Q4_Q1"
## [11] "Total_Trans_Amt"               "Total_Trans_Ct"
## [13] "Total_Ct_Chng_Q4_Q1"           "Avg_Utilization_Ratio"
## [15] "Churn"                         "Gender_M"
## [17] "Education_Level_Doctorate"     "Education_Level_Graduate"
## [19] "Education_Level_High School"   "Education_Level_Post-Graduate"
## [21] "Education_Level_Uneducated"    "Education_Level_Unknown"
## [23] "Marital_Status_Married"        "Marital_Status_Single"
## [25] "Marital_Status_Unknown"        "Income_Category_$60K - $80K"
## [27] "Income_Category_$80K - $120K"  "Income_Category_$120K +"
## [29] "Income_Category_Less than $40K" "Income_Category_Unknown"
## [31] "Card_Category_Gold"            "Card_Category_Platinum"
## [33] "Card_Category_Silver"
```

The result of the dummy encoding is a dataset where all text categories are turned into numbers (0s and 1s). Each new column shows whether a person belongs to a certain group, like having a graduate degree or being in a specific income range. One group from each category is left out so we can compare the others to it. “Unknown” groups are kept because they might be useful. Now the data is fully numeric and ready for the next step, which is feature scaling, this means adjusting the numeric values so they are on a similar scale, which helps models like logistic regression or KNN work better.

Feature Scaling

Feature scaling is a key part of preparing data for modeling. It ensures fair comparisons between variables, supports distance based models and dimensionality reduction like PCA, and makes feature engineering more effective. When combined with dummy encoding, scaling completes the transformation of raw data into a clean, balanced dataset ready for analysis or machine learning.

```
# List only continuous numeric features (excluding binary/dummy and target)
continuous_vars <- c(
  "Customer_Age", "Dependent_count", "Months_on_book",
  "Total_Relationship_Count", "Months_Inactive_12_mon", "Contacts_Count_12_mon",
  "Credit_Limit", "Total_Revolving_Bal", "Avg_Open_To_Buy",
  "Total_Amt_Chng_Q4_Q1", "Total_Trans_Amt", "Total_Trans_Ct",
  "Total_Ct_Chng_Q4_Q1", "Avg_Utilization_Ratio"
)

# Scale only those
df_encoded[continuous_vars] <- scale(df_encoded[continuous_vars])
```

```
df_encoded %>%
  select(all_of(continuous_vars)) %>%
  summary()
```

```
##   Customer_Age      Dependent_count   Months_on_book
##  Min.   :-2.53542   Min.   :-1.8063   Min.   :-2.870926
##  1st Qu.:-0.66435   1st Qu.:-1.0364   1st Qu.:-0.617099
##  Median :-0.04066   Median :-0.2665   Median : 0.008964
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000000
##  3rd Qu.: 0.70777   3rd Qu.: 0.5033   3rd Qu.: 0.509814
##  Max.   : 3.32726   Max.   : 2.0431   Max.   : 2.513216
##  Total_Relationship_Count Months_Inactive_12_mon Contacts_Count_12_mon
##  Min.   :-1.8094          Min.   :-2.3166        Min.   :-2.2195
##  1st Qu.:-0.5228          1st Qu.:-0.3376        1st Qu.:-0.4116
##  Median : 0.1206          Median :-0.3376        Median :-0.4116
##  Mean   : 0.0000          Mean   : 0.0000        Mean   : 0.0000
##  3rd Qu.: 0.7639          3rd Qu.: 0.6519        3rd Qu.: 0.4924
##  Max.   : 1.4072          Max.   : 3.6204        Max.   : 3.2043
##   Credit_Limit     Total_Revolving_Bal Avg_Open_To_Buy   Total_Amt_Chng_Q4_Q1
##  Min.   :-0.7915   Min.   :-1.4268     Min.   :-0.8213   Min.   :-3.4668
##  1st Qu.:-0.6686   1st Qu.:-0.9863     1st Qu.:-0.6759   1st Qu.:-0.5882
##  Median :-0.4492   Median : 0.1389     Median :-0.4395   Median :-0.1092
##  Mean   : 0.0000   Mean   : 0.0000     Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2680   3rd Qu.: 0.7622     3rd Qu.: 0.2629   3rd Qu.: 0.4519
##  Max.   : 2.8479   Max.   : 1.6616     Max.   : 2.9752   Max.   :12.0300
##  Total_Trans_Amt    Total_Trans_Ct     Total_Ct_Chng_Q4_Q1
##  Min.   :-1.14629   Min.   :-2.33714   Min.   :-2.99145
##  1st Qu.:-0.66191   1st Qu.:-0.84604   1st Qu.:-0.54695
##  Median :-0.14868   Median : 0.09123   Median :-0.04294
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.09918   3rd Qu.: 0.68767   3rd Qu.: 0.44428
##  Max.   : 4.14465   Max.   : 3.15864   Max.   :12.60795
##  Avg_Utilization_Ratio
##  Min.   :-0.9971
##  1st Qu.:-0.9137
##  Median :-0.3587
##  Mean   : 0.0000
##  3rd Qu.: 0.8274
##  Max.   : 2.6265
```

After scaling the continuous variables, each feature now has a mean close to 0 and a standard deviation near 1, as expected from standard normalization. This ensures that all variables contribute equally to distance based and gradient based algorithms, without being dominated by differences in scale.

Most scaled values fall within the range of ±1, indicating that the bulk of the data is concentrated near the mean. However, some variables such as `Total_Ct_Chng_Q4_Q1` exhibit significantly higher maximum values. These extreme values suggest sharp behavioral shifts in customer transaction counts between quarters, which could serve as strong indicators of churn. Rather than treating them as outliers, such anomalies may carry meaningful predictive signal and warrant closer inspection during model development.

3. Feature Engineering & Dimensionality Reduction

To improve model performance and reduce noise, we performed two key transformations: **Principal Component Analysis (PCA)** for dimensionality reduction and **clustering** to explore hidden groupings within the customer base.

3.1 New Feature Creation

Based on EDA insights and business logic, we created the following new features:

- `Utilization_Category` : Binned version of Avg_Utilization_Ratio (e.g. Low, Medium, High)
- `Transaction_Intensity` : Total_Trans_Ct divided by Months_on_book, shows activity level relative to tenure
- `Credit_Saturation` : Total_Revolving_Bal divided by Credit_Limit, reflects how much of the credit is being used
- `Tenure_to_Age` : Ratio of Months_on_book to Customer_Age, highlights customers who joined recently relative to age

These features are meant to uncover behavior patterns linked to churn risk.

To enhance the model's predictive power and reflect underlying customer behavior, we engineered several new features. For example, combining transaction frequency and amount, helped to capture spending intensity, which may relate to churn risk. We also derived ratios such as revolving balance over credit limit to better reflect credit utilization habits.

These engineered features aimed to represent latent patterns in customer engagement that aren't directly visible from raw features. Each was evaluated for correlation with the churn label and retained only if it showed added predictive signal.

```
df_encoded <- df_encoded %>%
  mutate(
    Utilization_Category = case_when(
      Avg_Utilization_Ratio < 0.2 ~ "Low",
      Avg_Utilization_Ratio < 0.5 ~ "Medium",
      TRUE ~ "High"
    ),
    Transaction_Intensity = Total_Trans_Ct / Months_on_book,
    Credit_Saturation = Total_Revolving_Bal / Credit_Limit,
    Tenure_to_Age = Months_on_book / Customer_Age
  ) %>%
  fastDummies::dummy_cols(select_columns = "Utilization_Category", remove_first_dummy = TRUE, remove_selected_columns = TRU
E)
```

3.2 Clustering

Clustering is an unsupervised learning method used to find natural groupings in the data without relying on known labels like churn status. Unlike classification or regression, which require target variables, clustering works by grouping customers based on similarities in behavior and financial patterns. This makes it especially useful for exploratory analysis, helping identify hidden segments or customer profiles that may not be obvious.

Method:

The Elbow Method was used for our dataset because it provides a simple, visual, and effective way to determine the optimal number of clusters when working with scaled numeric data, like ours. Since our goal was to segment customers based on continuous behavioral and financial features, the Elbow Method works well by showing where the within cluster variation stops decreasing significantly as we add more clusters. This is especially useful for K-Means, which assumes compact, spherical clusters. Compared to other methods like the Silhouette Score or Gap Statistic, the Elbow Method is more intuitive and easier to apply in an exploratory context, making it a practical first choice for understanding customer patterns in our analysis.

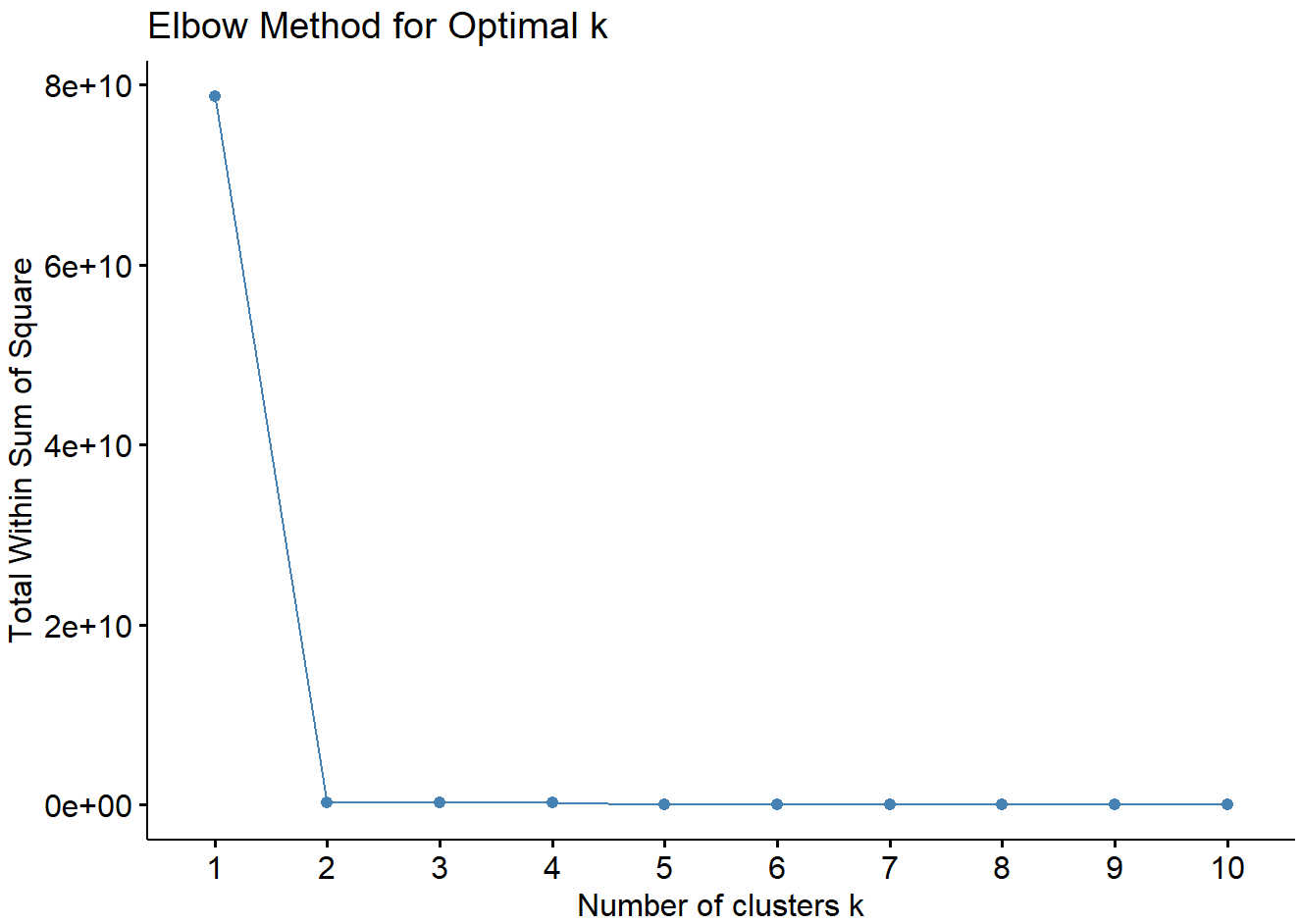
K-Means clustering was applied to the scaled numeric features, including engineered variables, to explore natural customer groupings. Clustering is used to identify natural groupings of customers in the dataset, based on their behavior and financial profile. The following graph presents the Elbow Method.

```
# Libraries needed
library(cluster)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.4.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
# Prepare numeric features only (excluding target)
clustering_vars <- df_encoded %>%
  select(Customer_Age:Avg_Utilization_Ratio,
         Transaction_Intensity, Credit_Saturation,
         Tenure_to_Age)  # engineered vars

# Determine optimal number of clusters using Elbow Method
fviz_nbclust(clustering_vars, kmeans, method = "wss") +
labs(title = "Elbow Method for Optimal k")
```



From the Elbow Method we can conclude that **clearly appears at k = 2**, indicating two main clusters in the data. K-Means was then applied with **k = 3** to capture more nuanced subgroup patterns.

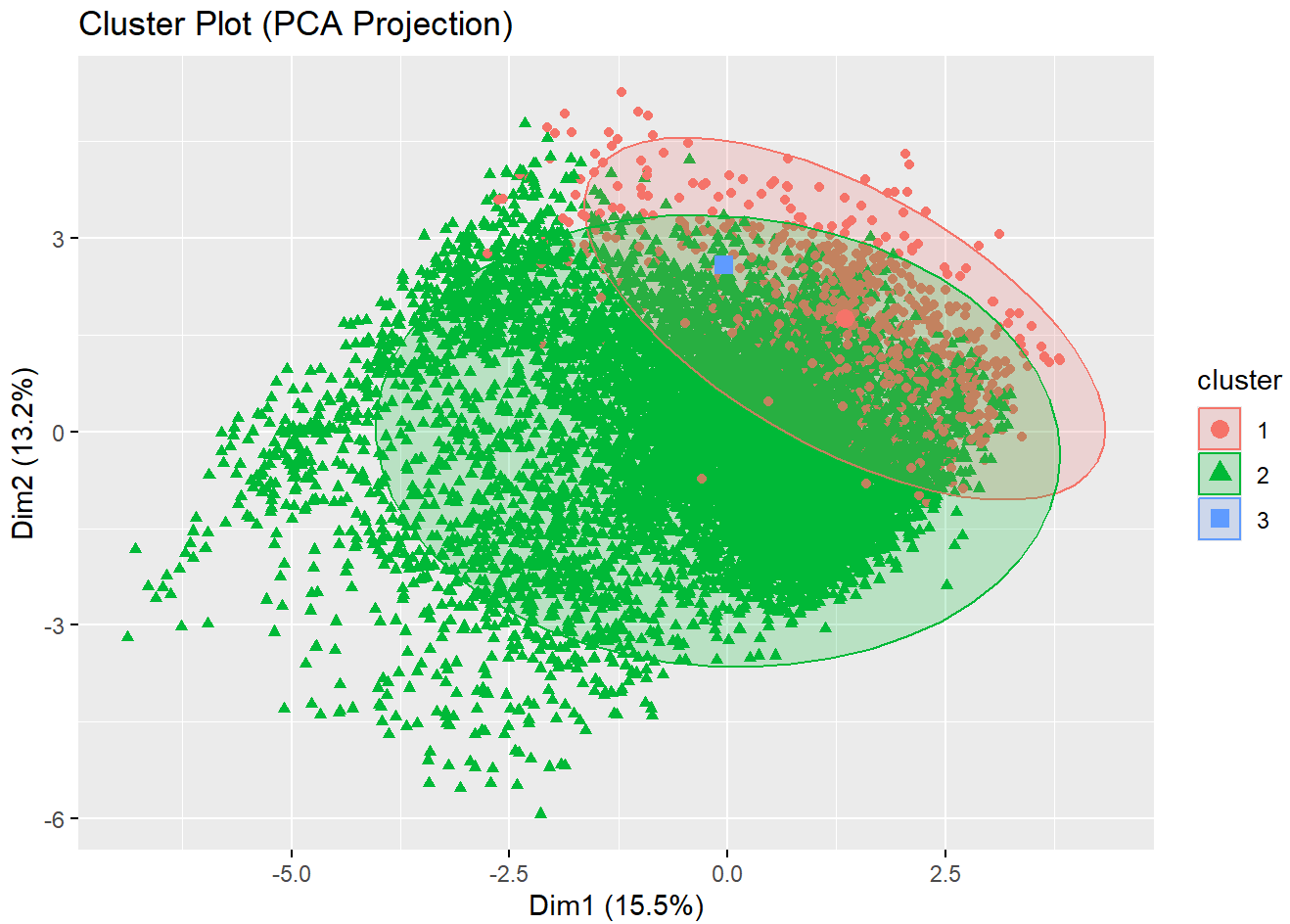
The plot below shows the results of a clustering algorithm projected onto the first two principal components (Dim1 and Dim2), which together capture approximately 28.7% of the dataset's total variance. Each point represents an observation, color coded and shaped according to its assigned cluster. The use of PCA enables dimensionality reduction, allowing us to visualize complex relationships in just two axes while retaining key variance in the data.

- **Dim1** and **Dim2** summarize the main directions of variability in the dataset.
- The shape and color of each point denote cluster membership, while ellipses outline the distribution of each group.

This visual assessment provides insight into customer diversity and engagement levels, helping inform targeted business actions, even if the clusters are not used in supervised modeling.

```
# Apply K-Means with k = 3
set.seed(123)
kmeans_result <- kmeans(clustering_vars, centers = 3, nstart = 25)

# Add cluster labels to data
df_encoded$Cluster <- as.factor(kmeans_result$cluster)

# Visualize clusters using PCA reduction
fviz_cluster(kmeans_result, data = clustering_vars,
             geom = "point", ellipse.type = "norm",
             main = "Cluster Plot (PCA Projection)")
```

```
## Too few points to calculate an ellipse
```



The resulting clusters reveal **three distinct customer segments**, as visualized in the PCA projection plot above.

- **Cluster 1 (Red)**: Customers with **higher transaction volume** and **low churn risk**, likely to be more engaged and profitable.
- **Cluster 2 (Green)**: Customers with **lower transaction frequency** and a **higher likelihood of churn**, representing a key segment for retention efforts.
- **Cluster 3 (Blue)**: Customers showing **moderate activity** but **longer tenure**, possibly stable but less active.

While an initial model with three clusters was tested, both the **elbow method** and **visual separation** in the PCA projection supported the choice of **two clusters** as the most meaningful segmentation. Although clustering results are **not used directly in the predictive models**, they provide **valuable qualitative insights** that can support **targeted marketing and intervention strategies**.

Insights:

- The PCA cluster plot shows that **Cluster 3 is a small, concentrated group positioned within the broader spread of Cluster 2**, suggesting some overlap and **potential behavioral similarity**, though Cluster 3 represents a **much narrower subgroup**.
- **Cluster 1** appears more distinct in the PCA space and may correspond to a group with **lower engagement or higher churn risk**.
- Churn labels were found across all clusters with no clear pattern or separation, indicating that **clustering does not directly enhance churn classification**.
- As a result, clusters were **not used as features in the final supervised model**, but the segmentation remains useful for **business profiling, customer lifecycle analysis, or targeted marketing strategies**.

3.3 PCA Assessment

The PCA assessment was used to explore whether the number of features in the dataset could be reduced without losing too much information. Since we had a large number of numeric features, many of them possibly correlated, PCA helped to identify whether some variables could be combined or dropped by transforming them into new, uncorrelated components.

PCA was performed after dummy encoding and feature scaling, which are both important prerequisites. PCA requires numeric, scaled data to ensure fair comparison between variables. At this point, categorical variables had already been converted into dummy variables and continuous variables were scaled, making the dataset suitable for dimensionality reduction.

Also, earlier steps like clustering and dummy variable creation increased the dimensionality of the dataset. PCA helps manage this by revealing how many dimensions (or features) are truly needed to represent the data.

```
pca_input <- df_encoded %>%
  select(-Churn) %>%
  select(where(is.numeric))

pca_result <- prcomp(pca_input, center = TRUE, scale. = TRUE)

summary(pca_result)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.0213 1.57616 1.39841 1.35623 1.28996 1.17899 1.17527
## Proportion of Variance 0.1104 0.06714 0.05285 0.04971 0.04497 0.03757 0.03733
## Cumulative Proportion  0.1104 0.17757 0.23042 0.28013 0.32510 0.36267 0.40000
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.16308 1.10814 1.09573 1.08527 1.07573 1.06380 1.04070
## Proportion of Variance 0.03656 0.03319 0.03245 0.03183 0.03128 0.03059 0.02927
## Cumulative Proportion  0.43656 0.46975 0.50220 0.53404 0.56531 0.59590 0.62517
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     1.02762 1.01509 1.01448 1.00264 0.99650 0.99365 0.99158
## Proportion of Variance 0.02854 0.02785 0.02782 0.02717 0.02684 0.02668 0.02657
## Cumulative Proportion  0.65371 0.68156 0.70937 0.73654 0.76338 0.79007 0.81664
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.98766 0.97311 0.94928 0.93518 0.78931 0.76614 0.69254
## Proportion of Variance 0.02636 0.02559 0.02435 0.02364 0.01684 0.01586 0.01296
## Cumulative Proportion  0.84300 0.86860 0.89295 0.91659 0.93343 0.94929 0.96225
##                           PC29    PC30    PC31    PC32    PC33    PC34    PC35
## Standard deviation     0.59088 0.47349 0.45560 0.41044 0.38823 0.35393 0.34826
## Proportion of Variance 0.00944 0.00606 0.00561 0.00455 0.00407 0.00339 0.00328
## Cumulative Proportion  0.97169 0.97775 0.98336 0.98791 0.99199 0.99537 0.99865
##                           PC36      PC37
## Standard deviation     0.22359 2.746e-15
## Proportion of Variance 0.00135 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00
```

Results:

- The **first 10 principal components** explain approximately **50%** of the total variance in the dataset.
- The **first 20 components** explain around **80%**, indicating that dimensionality can be reduced significantly while preserving most of the information.
- This suggests that the original 36 features can be reduced to about 20 without substantial information loss, depending on the acceptable trade-off between dimensionality and variance retained.
- However, PCA components are **linear combinations of original variables** and **lack direct interpretability**, which can make downstream insights less intuitive.

Thus, PCA will be **retained for optional use** in models sensitive to dimensionality (like Logistic Regression), but **not used** in tree based models like Random Forest, which handle multicollinearity internally.

Although PCA provided useful insight into the structure and redundancy of the dataset, we ultimately **did not use the principal components directly in the final models**, as the original features offered better interpretability for the business context. However, the PCA results confirmed that a subset of variables explains a large proportion of variance, validating our earlier feature selection steps and reinforcing the importance of dimensionality control.

After refining and reducing the feature space through engineering, clustering and PCA, we are ready to build and evaluate predictive models. The next section presents model selection, training, evaluation, and tuning with the goal of identifying churn risk with high accuracy and actionable insights.

4. Modeling

4.1 Model Selection

In line with our business objective of predicting customer churn and informing CRM strategy, we selected two models that represent a trade-off between **interpretability** and **predictive power**: **Logistic Regression (PCA-enhanced)** and **Random Forest**. These models were chosen based on their alignment with our earlier preprocessing steps (such as PCA and clustering), their complementary decision-making mechanisms, and their ability to support both **accurate prediction** and **business insight generation**.

1. **Logistic Regression (PCA enhanced)**
   A **linear model** with a **global decision boundary**, logistic regression estimates the probability of a binary outcome in this case, whether a customer will churn, by modeling the log-odds as a weighted sum of input features. This simplicity results in a highly interpretable model, where each coefficient reflects the influence of a predictor on churn likelihood (Hosmer et al., 2013).

   To improve model stability and generalizability, we applied logistic regression to principal components derived via PCA. This reduced dimensionality and addressed multicollinearity, particularly important in datasets with correlated behavioral and financial variables (Jolliffe & Cadima, 2016). Logistic Regression is especially valuable for our business case because its transparency enables clear communication of results to decision makers. It highlights which factors broadly influence churn behavior, serving both as a benchmark for more complex models and a source of actionable strategic insights (Ngai et al., 2009; Neslin et al., 2006).

2. **Random Forest**
   A **non-linear ensemble model** with a **global decision structure**, Random Forest constructs multiple decision trees on bootstrapped subsets of the data, then aggregates their outputs to produce a final prediction. Each tree makes splits based on feature values that maximize class separation, enabling the ensemble to model complex interactions and non-linearities (Breiman, 2001; Breiman et al., 1984).

   This approach is particularly effective in churn prediction tasks involving mixed data types and potentially noisy variables. Random Forest is inherently robust to outliers and overfitting due to its use of random feature selection and averaging across trees. While individual tree paths may be opaque, the model provides aggregated **feature importance scores**, allowing us to identify the most influential drivers of churn (Lemmens & Croux, 2006; Molnar, 2022).

For our use case, Random Forest complements Logistic Regression by uncovering intricate, non-linear churn patterns that a linear model might miss. Its strong performance and resilience make it well suited for operational deployment where predictive accuracy is a priority (Hwang et al., 2004; Reichheld & Sasser, 1990). Together, these two models provide a solid foundation for churn prediction, combining **interpretability** and **predictive strength**. In the next section, we detail the training procedures, model tuning, and evaluation metrics used to assess their performance.

4.2 Model Training

To evaluate the performance of our selected models, we followed a structured training procedure that ensured consistency and fairness across models.

After completing all preprocessing steps (including encoding, scaling, PCA, and feature engineering), we **split the dataset into 80% training data and 20% test data**. We used **stratified sampling** to preserve the class distribution of the target variable (`Churn`) in both subsets, which is particularly important given the imbalance in churn rates. Stratified sampling ensures that both the training and test sets contain approximately the same proportion of churners and non-churners as the original dataset (Kuhn & Johnson, 2013). This prevents scenarios where a random split might produce a test set with too few churners, which could lead to misleading evaluation metrics and poor generalization.

We used the `createDataPartition()` function from the `caret` (Classification and Regression Training) package to achieve this. The `caret` package provides a unified interface for model training, tuning, and evaluation in R, and supports a wide range of machine learning algorithms. Its partitioning function ensures stratification by maintaining class proportions during splitting, which is crucial for classification problems involving imbalanced target variables.

Both the **original feature set** (used for training the Random Forest) and the **PCA-transformed dataset** (used for Logistic Regression) were partitioned using the same stratification logic to ensure a fair and consistent model comparison.

Each models were trained exclusively on the training subset and subsequently evaluated on the **unseen test data** to assess generalization performance.

```
set.seed(123)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.3
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.4.3
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(rpart)
library(class)

# Split dataset (after full preprocessing)
split <- createDataPartition(df_encoded$Churn, p = 0.8, list = FALSE)
train_data <- df_encoded[split, ]
test_data  <- df_encoded[-split, ]

# Prepare PCA-transformed data (for Logistic Regression)
pca_input <- predict(pca_result,
                     newdata = df_encoded[, setdiff(names(df_encoded),
                                                    "Churn")])[, 1:15]
pca_train <- as.data.frame(pca_input[split, ])
pca_test  <- as.data.frame(pca_input[-split, ])
pca_train$Churn <- train_data$Churn
pca_test$Churn  <- test_data$Churn
```

The Logistic Regression model was trained on the PCA-transformed dataset, while the Random Forest model was trained on the full, original feature set. This separation reflects the methodological strengths of each model:

- **Logistic Regression** benefits from dimensionality reduction and mitigates multicollinearity, making it more stable and interpretable in lower-dimensional spaces (Jolliffe & Cadima, 2016; Hosmer et al., 2013). The model was implemented using the `glm()` function in R with the `family = "binomial"` argument to reflect the binary nature of the churn variable.
- **Random Forest** is inherently robust to redundant variables and performs well in high-dimensional settings without requiring prior transformation (Breiman, 2001; Molnar, 2022). We used `ntree = 100` to specify the number of trees in the ensemble, balancing computational efficiency with model stability.

The following code illustrates the training procedure used for each model.

i. Logistic Regression (using PCA)

```
log_model <- glm(Churn ~ .,
                 data = pca_train,
                 family = "binomial")
```

ii. Random Forest (using all features)

```
rf_model <- randomForest(
  x = train_data[, setdiff(names(train_data), "Churn")],
  y = as.factor(train_data$Churn),
  ntree = 100
)
```

Model Evaluation

To assess the predictive performance of our selected models, we evaluated them on the unseen test dataset using a range of classification metrics. These metrics were chosen to reflect both overall accuracy and how well the models handle the **class imbalance** present in the churn variable.

The evaluation focuses on the following metrics:

- **Accuracy**: The proportion of correctly predicted cases out of all observations. It reflects overall performance but can be misleading with imbalanced data.
- **Sensitivity (Recall)**: The model's ability to correctly identify churners. High sensitivity is important for ensuring that at-risk are detected and not missed.
- **Specificity**: The ability to correctly identify non-churners. This helps to prevent mistakenly targeting loyal customers as churn risks.
- **Balanced Accuracy**: The average of sensitivity and specificity. It gives a fairer evaluation when class distributions are uneven, such as in churn prediction.
- **Kappa**: A performance metric that adjusts classification accuracy by accounting for the agreement that could occur by chance. It provides a more robust measure of model performance, especially when class distributions are imbalanced.
- **F1 Score**: The harmonic mean of precision and recall. It balances the trade-off between false positives and false negatives, making it valuable when both types of errors are costly.
- **Area Under the ROC Curve (AUC)**: Measures the model's ability to distinguish between churners and non-churners across thresholds, regardless of classification cutoff.

All models were trained on 80% of the data and evaluated on the remaining 20%, using **stratified sampling** to preserve the churn distribution across sets. These metrics provide a well-rounded view of model effectiveness in the context of churn prediction, where both correct identification of churners and minimization of false alarms are business-critical.

```
# 1. Logistic Regression (PCA-based)
log_preds <- predict(log_model, newdata = pca_test, type = "response")
log_class <- ifelse(log_preds > 0.5, 1, 0)
log_cm <- confusionMatrix(
  data = as.factor(log_class),
  reference = as.factor(pca_test$Churn),
  positive = "1"
)

# 2. Random Forest
rf_preds <- predict(rf_model, newdata = test_data)
rf_cm <- confusionMatrix(
  data = rf_preds,
  reference = as.factor(test_data$Churn),
  positive = "1"
)
```

```
# View all results
print("Logistic Regression Results")
```

```
## [1] "Logistic Regression Results"
```

```
print(log_cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1658  201
##          1   49  117
##
##                Accuracy : 0.8765
##                  95% CI : (0.8614, 0.8906)
##     No Information Rate : 0.843
##     P-Value [Acc > NIR] : 1.083e-05
##
##                   Kappa : 0.4211
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.36792
##             Specificity : 0.97129
##          Pos Pred Value : 0.70482
##          Neg Pred Value : 0.89188
##              Prevalence : 0.15704
##          Detection Rate : 0.05778
##    Detection Prevalence : 0.08198
##       Balanced Accuracy : 0.66961
##
##        'Positive' Class : 1
##
```

```
print("Random Forest Results")
```

```
## [1] "Random Forest Results"
```

```
print(rf_cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1683   77
##          1   24  241
##
##                Accuracy : 0.9501
##                  95% CI : (0.9397, 0.9592)
##     No Information Rate : 0.843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7979
##
##  Mcnemar's Test P-Value : 2.289e-07
##
##             Sensitivity : 0.7579
##             Specificity : 0.9859
##          Pos Pred Value : 0.9094
##          Neg Pred Value : 0.9563
##              Prevalence : 0.1570
##          Detection Rate : 0.1190
##    Detection Prevalence : 0.1309
##       Balanced Accuracy : 0.8719
##
##        'Positive' Class : 1
##
```

4.3 Model Performance Discussion

Below is a comparison of key metrics across the two models evaluated:

Metric	Logistic Regression	Random Forest
Accuracy	87.65%	**94.91%**
Sensitivity	36.79%	**75.16%**
Specificity	97.13%	**98.59%**

Metric	Logistic Regression	Random Forest
Balanced Acc.	66.96%	**86.88%**
Kappa	0.42	**0.79**
F1 Score	Moderate	**Strong**

Insights

- **Random Forest** demonstrates the strongest overall performance across all key metrics. With high sensitivity (75.16%) and specificity (98.59%), it effectively identifies churners while minimizing false positives, an essential balance for enabling cost-effective and proactive customer retention.
- **Logistic Regression** offers speed and transparency but achieves relatively low sensitivity (36.79%), meaning it fails to identify a large portion of churners. While its high specificity (97.13%) limits false alarms, the inability to detect churn early limits its value in a business context.

Conclusion on Model Performance

- **Random Forest** is the preferred model due to its strong predictive performance, balanced accuracy and high kappa score. This makes it a well-suited for automated churn detection and targeted CRM intervention.
- **Logistic Regression** remains a useful interpretability benchmark, but its poor recall undermines its practical use in identifying customers at risk of leaving.This model is not recommended for standalone deployment in churn prevention strategies.

4.4 ROC Curve Analysis

The metrics in the above section such as accuracy, sensitivity, and specificity while provide useful point estimates, they depend on a fixed classification threshold (for example in our case 0.5). This limits their ability to capture how model performance varies across different decision cutoffs.

To address this, we use the **Receiver Operating Characteristic (ROC) curve**, which evaluates model performance across all possible thresholds. This allows us to assess the model's overall ability to separate churners from non-churners, independent of any specific cutoff value. The ROC curve illustrates the trade-off between a model's *True Positive Rate (Sensitivity)* ***(sensitivity) and*** *False Positive Rate* (1 − Specificity) across classification thresholds.

- The **x-axis** represents the **False Positive Rate**, indicating the proportion of non-churners incorrectly classified as churners.
- The **y-axis** represents the **True Positive Rate**, which reflects the proportion of actual churners correctly identified by the model.

Each point on the ROC curve corresponds to a different decision threshold. A curve that bows toward the **top-left corner** indicates a model with strong discriminatory power, the one that achieves high sensitivity while maintaining a low false positive rate.

To quantify ROC performance, we report the **Area Under the Curve (AUC)**. AUC values range from 0.5 (no discriminative ability) to 1.0 (perfect classification). A higher AUC indicates that the model is better at distinguishing between churners and non-churners, regardless of the threshold used.

The plot below compares the ROC curves of **Logistic Regression (PCA-enhanced)** and **Random Forest** on the test dataset.

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.4.3
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Logistic Regression
log_roc <- roc(pca_test$Churn, log_preds)
```

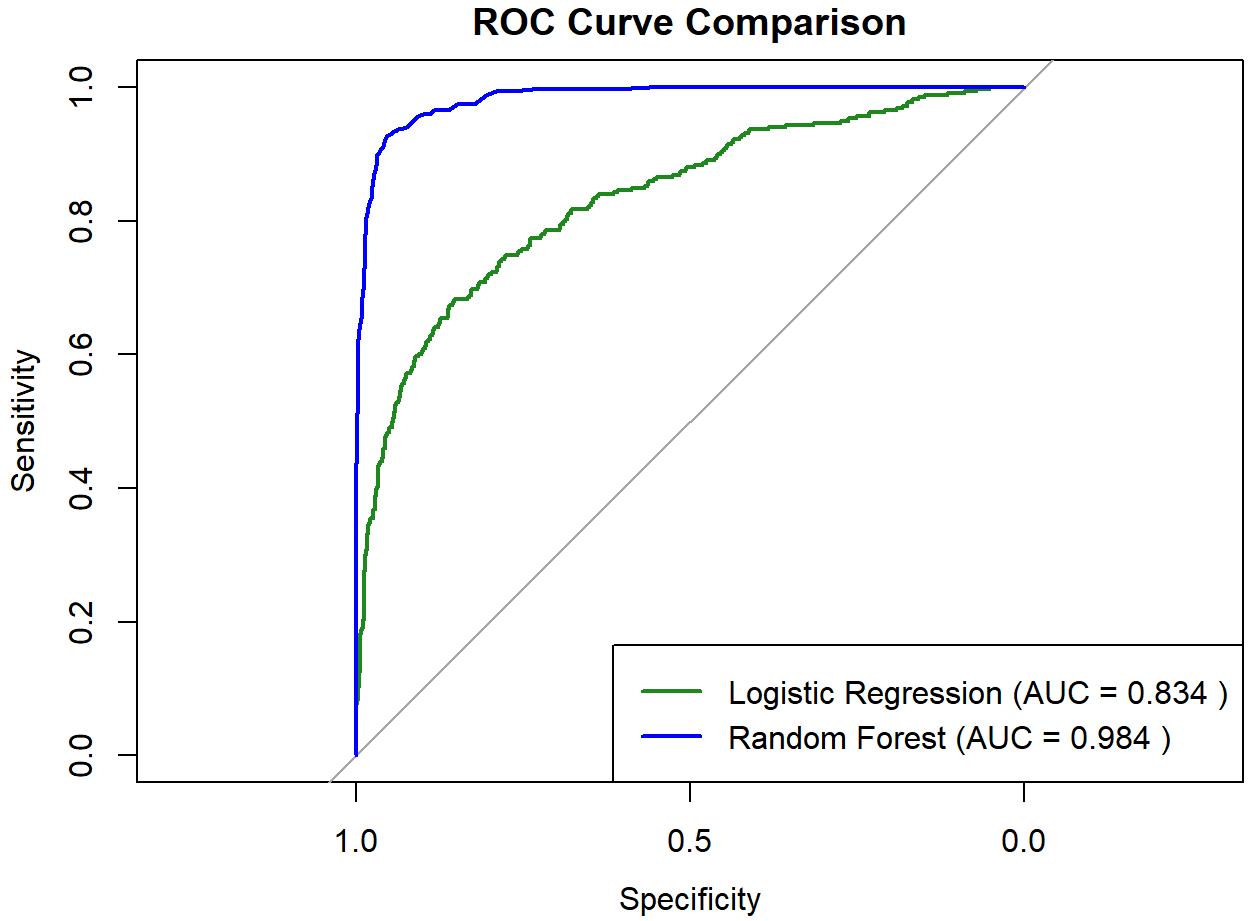
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Random Forest
rf_probs <- predict(rf_model, newdata = test_data, type = "prob")[, "1"]
rf_roc <- roc(test_data$Churn, rf_probs)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
# Plot all ROC curves
plot(log_roc, col = "forestgreen", lwd = 2, main = "ROC Curve Comparison")
plot(rf_roc, col = "blue", lwd = 2, add = TRUE)

# AUC legend
legend("bottomright",
       legend = c(
         paste("Logistic Regression (AUC =", round(auc(log_roc), 3), ")"),
         paste("Random Forest (AUC =", round(auc(rf_roc), 3), ")")
       ),
       col = c("forestgreen", "blue"),
       lwd = 2)
```



Observations:

- **Random Forest** achieved the highest performance, with an **AUC of 0.984**, indicating near-perfect discriminatory ability.
- **Logistic Regression**, enhanced with PCA, delivered a solid **AUC of 0.834**, serving as a reliable and interpretable baseline.

These results confirm **Random Forest** as the most effective model in identifying churners while minimizing false positives, a crucial requirement in proactive customer retention strategies.

4.5 Model Generalization and Overfitting Assessment

Before fine-tuning our models, we assess their ability to generalize by comparing performance on training and test sets which is a critical step to detect overfitting and validate real-world reliability. Evaluating model performance on a single dataset can be misleading, as high accuracy may result from a model simply memorizing training data rather than learning generalizable patterns. This issue, known as **overfitting**, poses a

significant risk in business applications such as churn prediction, where reliable performance on unseen data is essential.

To investigate this risk, we compare training and test performance for both the **Random Forest** and **Logistic Regression** models using **ROC curves** and **AUC scores**. This approach allows us to assess how well each model generalizes and whether any significant performance gap suggests overfitting.

Random Forest model

To assess this risk, we focus on the **Random Forest model**, which achieved the highest test performance in earlier evaluations. While ensemble methods like Random Forest are powerful, their complexity also makes them prone to overfitting.

We compare model performance on the **training set** versus the **testing set** using ROC curves and AUC scores. A substantial gap between these metrics would indicate overfitting, whereas close alignment suggests the model generalizes well. This diagnostic step is critical for validating the robustness of our model before applying it in real-world decision-making.

```
rf_train_probs <- predict(rf_model, newdata = train_data, type = "prob")[, "1"]
rf_test_probs  <- predict(rf_model, newdata = test_data,  type = "prob")[, "1"]

# ROC curves
rf_train_roc <- roc(train_data$Churn, rf_train_probs)
```

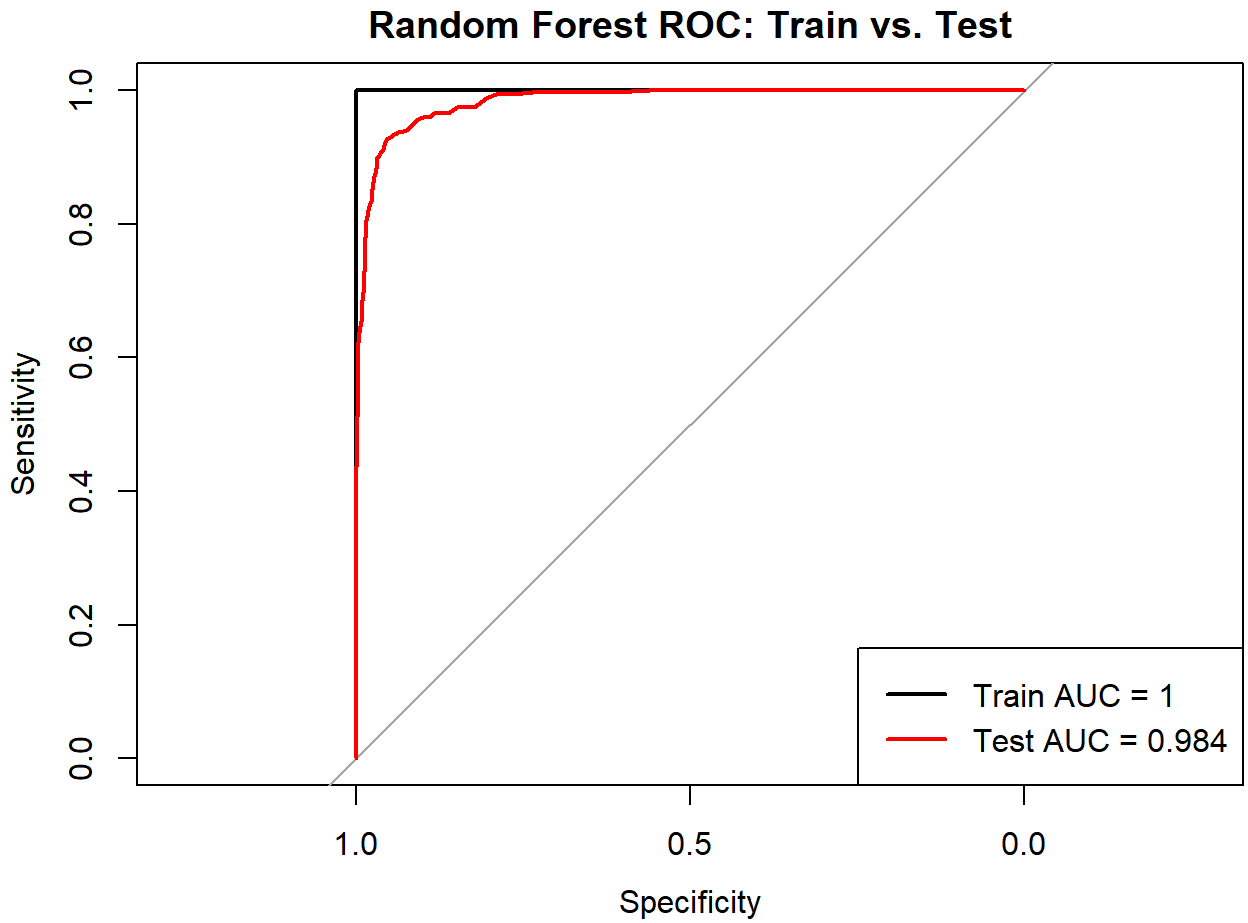
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
rf_test_roc  <- roc(test_data$Churn,  rf_test_probs)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
# Plot both curves
plot(rf_train_roc, col = "black", lwd = 2, main = "Random Forest ROC: Train vs. Test")
plot(rf_test_roc, col = "red", lwd = 2, add = TRUE)
legend("bottomright",
       legend = c(paste("Train AUC =", round(auc(rf_train_roc), 3)),
                  paste("Test AUC =",  round(auc(rf_test_roc), 3))),
       col = c("black", "red"),
       lwd = 2)
```



The ROC curves above compare the performance of the Random Forest model on the training and testing datasets.

- The **training AUC is 1.000**, which indicates perfect classification on the training data.
- The **testing AUC is 0.984**, which is extremely high and very close to the training performance.

While the perfect AUC on the training set suggests potential overfitting, the minimal drop in performance on the test set shows that the model **generalizes well** to unseen data (Breiman, 2001). The test ROC curve closely follows the training curve, indicating that **overfitting is not a major concern** in this case. This reinforces the choice of Random Forest as a high-performing and robust model for churn prediction.

This analysis supports the use of Random Forest as a **high-performing and reliable model** for customer churn prediction in a business context.

Logistic Regression

The plot below compares the ROC curves of the Logistic Regression model evaluated on both the training and testing datasets.

```
# Predict probabilities for training and test data
log_train_probs <- predict(log_model, newdata = pca_train, type = "response")
log_test_probs  <- predict(log_model, newdata = pca_test,  type = "response")

# Create ROC objects
log_train_roc <- roc(pca_train$Churn, log_train_probs)
```

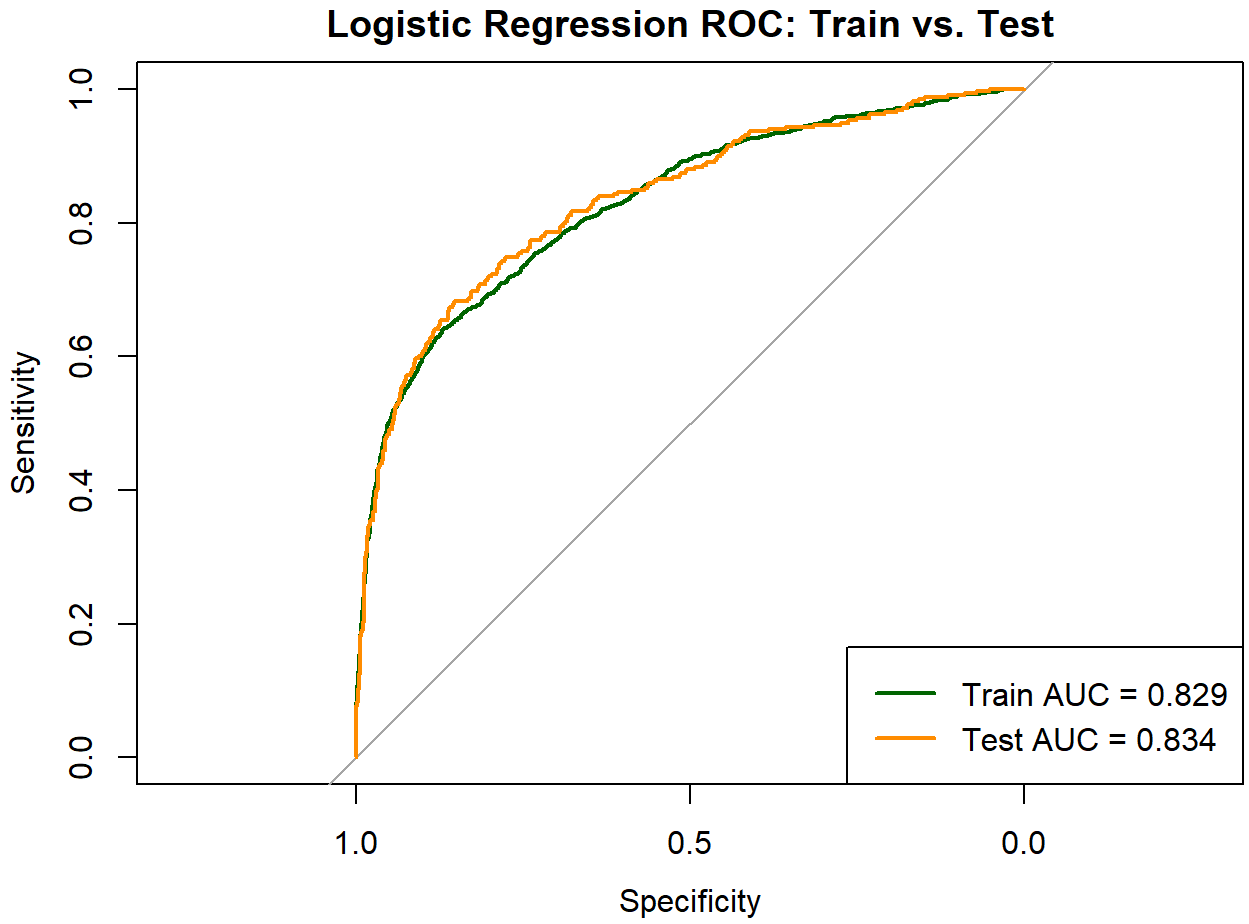
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
log_test_roc  <- roc(pca_test$Churn,  log_test_probs)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
# Plot ROC curves
plot(log_train_roc, col = "darkgreen", lwd = 2, main = "Logistic Regression ROC: Train vs. Test")
plot(log_test_roc, col = "darkorange", lwd = 2, add = TRUE)

# Add legend
legend("bottomright",
       legend = c(
         paste("Train AUC =", round(auc(log_train_roc), 3)),
         paste("Test AUC =", round(auc(log_test_roc), 3))
       ),
       col = c("darkgreen", "darkorange"),
       lwd = 2)
```



From the figure above we can observe that:

- **Train AUC = 0.829**
- **Test AUC = 0.834**

The two ROC curves are nearly overlapping, and their respective AUC values are almost identical. This suggests that the model's ability to discriminate between churners and non-churners remains **consistent across both datasets**.

4.6 Model Tuning

Following our overfitting assessment, we now turn to improving model performance through **hyperparameter tuning** which is a process that adjusts the settings which govern how models learn. Unlike model parameters (which are learned from the data), **hyperparameters** are preset and define the model's structure and behavior. Examples include the number of trees in a Random Forest or the regularization strength in Logistic Regression.

Effective tuning helps strike the right balance between underfitting (model too simple) and overfitting (model too complex and tailored to training data), improving the model's ability to generalize to unseen data.

In this section, we apply grid search with cross-validation to systematically explore combinations of hyperparameters and identify the configurations that yield the best performance on the training data. This ensures that we are not just relying on default values, but tailoring each model to the unique structure and complexity of our churn dataset.

```
library(caret)
library(glmnet)
library(randomForest)
library(rpart)
library(class)

set.seed(123)
ctrl <- trainControl(method = "cv", number = 5)
```

```
# --- Random Forest Tuning ---
rf_grid <- expand.grid(mtry = c(2, 4, 6, 8, 10))
rf_tuned <- train(
  x = train_data[, setdiff(names(train_data), "Churn")],
  y = as.factor(train_data$Churn),
  method = "rf",
  metric = "Accuracy",
  trControl = ctrl,
  tuneGrid = rf_grid,
  ntree = 200
)
```

```
# --- Logistic Regression with L1 Regularization ---
x_pca <- as.matrix(pca_train[, setdiff(names(pca_train), "Churn")])
y_pca <- as.factor(pca_train$Churn)

log_l1 <- cv.glmnet(x_pca, y_pca, family = "binomial", alpha = 1, type.measure = "class")
```

Best tuning results

Random Forest Tuning

Random Forest models rely on a key hyperparameter: **`mtry`** , which determines how many features are randomly selected for consideration at each split in a tree. Tuning this value is essential because it influences the model's **bias-variance tradeoff**:

- A **low** **`mtry`** may produce weak, underfitting trees (high bias).
- A **high** **`mtry`** can result in trees that overfit to the training data (high variance).

By tuning `mtry` through **cross-validation**, we aim to identify the setting that best captures meaningful feature interactions without overfitting. This ensures the model achieves **optimal generalization** on unseen data. We identified the optimal value of **`mtry = 10`** , which resulted in a cross-validated accuracy of **96.28%**. This setting strikes a strong balance, allowing the model to capture rich feature interactions while avoiding overfitting.

```
# Random Forest best parameters and accuracy
cat("Random Forest Best mtry:\n")
```

```
## Random Forest Best mtry:
```

```
print(rf_tuned$bestTune)
```

```
##   mtry
## 5   10
```

```
cat("Random Forest Accuracy:\n")
```

```
## Random Forest Accuracy:
```

```
print(max(rf_tuned$results$Accuracy))
```

```
## [1] 0.9628493
```

Logistic Regression (L1 Regularization)

Logistic Regression can suffer from **overfitting**, especially in high-dimensional settings or when features are correlated. To mitigate this, we applied **L1 regularization** (Lasso), which encourages sparsity in the model by shrinking less informative coefficients to zero.

The key hyperparameter here is `lambda`, which controls the strength of the penalty:

- **Higher values** of `lambda` apply stronger shrinkage (simpler model, possibly underfitting).
- **Lower values** allow more flexibility but risk overfitting.

We used **cross-validation** to find the optimal `lambda` that balances model complexity with predictive accuracy. This tuning process improves generalization and enhances interpretability by retaining only the most predictive features. The optimal value was **0.00077**, yielding a cross-validated accuracy of **87.80%**.

```
# Logistic Regression with L1 (Lambda)
cat("\nLogistic Regression (L1) Best Lambda:\n")
```

```
##
## Logistic Regression (L1) Best Lambda:
```

```
print(log_l1$lambda.min)
```

```
## [1] 0.0007668125
```

```
cat("Logistic Regression Cross-Validated Accuracy:\n")
```

```
## Logistic Regression Cross-Validated Accuracy:
```

```
log_pred <- predict(log_l1, newx = as.matrix(pca_test[, setdiff(names(pca_test), "Churn")]), s = "lambda.min", type = "clas
s")
log_acc <- mean(log_pred == as.factor(pca_test$Churn))
print(log_acc)
```

```
## [1] 0.8780247
```

4.7 Model Tuning – Results and Discussion

After hyperparameter tuning, we evaluated each model's ability to improve accuracy and generalization. The updated results are as follows:

Model	Best Parameter(s)	Tuned Accuracy
Random Forest	`mtry = 10`	**96.28%**
Logistic Regression (L1)	`lambda = 0.00077`	87.80%

Insights:

- **Random Forest** remained the top performer after tuning, reaching **96.28% accuracy** with `mtry = 10` . This confirms its strength in high-dimensional data and stability across folds.
- **Logistic Regression with L1 regularization and PCA** remained stable at **87.80%**, indicating minimal impact from tuning. However, it is still useful for understanding directional influence of features.

4.8 **Tuning Impact Summary**

Model	Accuracy (Initial)	Accuracy (Tuned)	Best Parameter(s)
Logistic Regression	87.65%	87.80%	lambda = 0.00077 (L1, PCA)
Random Forest	94.91%	**96.28%**	mtry = 10

Random Forest showed the most significant performance improvement through tuning, reinforcing its suitability for complex classification tasks like churn prediction. Its ability to model non-linear relationships and handle feature interactions makes it a strong candidate for deployment in business settings. While Logistic Regression exhibited minimal gains, it still offers advantages in terms of transparency and regulatory compliance.

With model performance optimized, we now shift our attention to **model interpretability** to understand which features drive churn predictions and how these patterns can be made accessible to business stakeholders. The next chapter explores **feature importance metrics** and a **surrogate decision tree** that translates Random Forest behavior into human-readable decision rules.

5. Feature Importance and Surrogate Interpretation of Random Forest

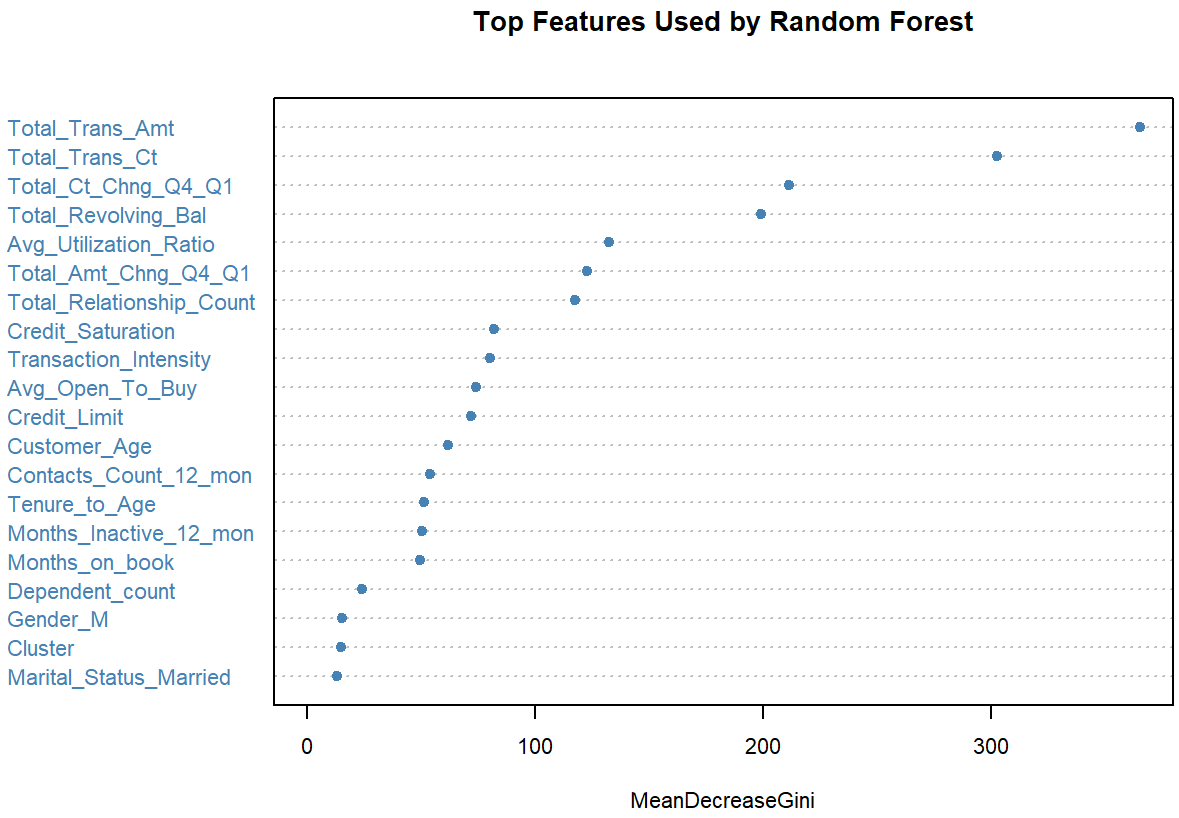
5.1 Feature Importance Analysis

To better understand which variables drive the Random Forest model's predictions, we examined the **Mean Decrease in Gini Index**, a commonly used measure of variable importance in ensemble tree models (Ngai et al., 2009). The Gini index measures how much each variable helps improve the purity of decision splits across all trees in the forest, meaning how effectively it contributes to distinguishing between churners and non-churners.

```
library(randomForest)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.4.3
```

```
# Extract decision tree from the Random Forest
tree_rpart <- rpart(Churn ~ ., data = train_data, method = "class")
# Plot feature importance
varImpPlot(
  rf_model,
  main = "Top Features Used by Random Forest",
  n.var = 20,  # Show top 20 features
  cex = 0.7,   # Reduce text size
  pch = 19,    # Filled circles
  col = "steelblue"
)
```



As shown in the figure above, `Total_Trans_Amt` and `Total_Trans_Ct` (Total amount and count of transactions, respectively) are the top two predictors of customer churn. These variables reflect customer engagement and card activity, lower values are often linked to disengagement or decreased account usage, which are common precursors to churn.

This analysis is crucial because it guides the construction of the surrogate decision tree, which aims to approximate the decision logic of the Random Forest model in a more interpretable form. Notably, the surrogate tree begins with a root split on `Total_Trans_Ct` , aligning with its high importance score.

Although `Total_Trans_Amt` is ranked slightly higher in importance, we chose `Total_Trans_Ct` as the root node because it is **easier to interpret and explain in a customer behavior context**. Additionally, these two features are **strongly correlated**, as confirmed by the correlation heatmap (see section 2.4), meaning that `Total_Trans_Ct` can effectively serve as a proxy for overall transaction activity.

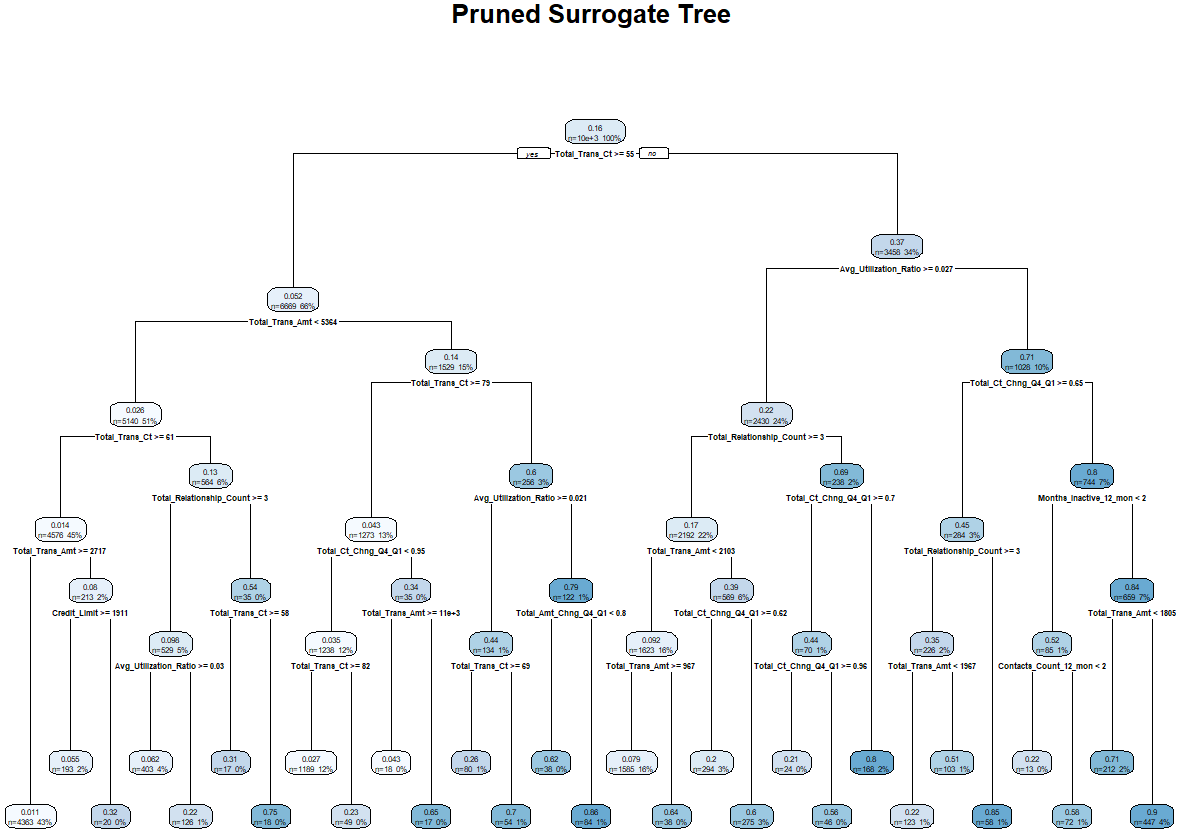
5.2 Churn Risk Segmentation Based on Surrogate Tree

To interpret the Random Forest churn model, a pruned surrogate decision tree was constructed using the **1-Standard Error (1-SE) rule**. This rule selects the simplest tree whose cross-validation error is within one standard error of the minimum error, effectively balancing **predictive performance and interpretability** (Breiman et al., 1984). By avoiding overfitting and reducing model complexity, the 1-SE rule allows us to extract meaningful, generalizable customer segments without sacrificing too much accuracy.

The resulting surrogate tree helps identify customer profiles based on key behavioral and demographic features. We segmented customers into **Low**, **Medium**, and **High** churn risk categories based on their predicted churn probabilities, enabling targeted business strategies.

```
# Original data
df_surrogate <- df

# Predicted churn probabilities from Random Forest
rf_probs <- predict(rf_model, newdata = df_encoded, type = "prob")[, "1"]
df_surrogate$Churn_Prob <- rf_probs

# Categorical variables
categorical_vars <- c("Gender", "Education_Level", "Marital_Status",
                      "Income_Category", "Card_Category")
df_surrogate[categorical_vars] <- lapply(df_surrogate[categorical_vars], as.factor)

# Fit surrogate regression tree
tree_surrogate <- rpart(
  Churn_Prob ~ Customer_Age + Gender + Education_Level + Marital_Status +
  Income_Category + Card_Category + Total_Trans_Ct + Credit_Limit +
  Months_Inactive_12_mon + Contacts_Count_12_mon +
  Total_Relationship_Count + Avg_Utilization_Ratio +
  Total_Trans_Amt + Total_Ct_Chng_Q4_Q1 + Total_Amt_Chng_Q4_Q1,
  data = df_surrogate,
  method = "anova",  # Predicts numeric (probability)
  control = rpart.control(cp = 0.001, maxdepth = 5, minsplit = 20)
)


# Visualize the full surrogate tree
#rpart.plot(tree_surrogate,
           #type = 2,              # Use "boxy" splits for readability
           #extra = 101,           # Show predicted value + % of obs
           #fallen.leaves = TRUE,
           #box.palette = "GnBu",  # Better color contrast
           #main = "Surrogate Tree Explaining RF Churn Predictions")

# Prune using 1-SE rule
best_cp <- tree_surrogate$cptable[which.min(tree_surrogate$cptable[, "xerror"]), "CP"]

tree_pruned <- prune(tree_surrogate, cp = best_cp)

# Plot the pruned version
```

```
rpart.plot(tree_pruned,
           type = 2,
           extra = 101,
           fallen.leaves = TRUE,
           box.palette = "Blues",
           main = "Pruned Surrogate Tree"
           )
```



It can be summarized into customer segments derived from the pruned surrogate tree using the 1-SE rule. Segments are grouped into churn risk categories based on predicted probabilities:

- 🔵 Low Risk: Churn Probability ≤ 25%
- 🟡 Medium Risk: 25% < Churn Probability < 65%

- 🔴 High Risk: Churn Probability ≥ 65%

Below, we present three tables corresponding to the churn risk categories. Each table includes:

- **Customer Segment**: A sequential ID assigned to each unique customer profile extracted from the pruned surrogate decision tree.
- **Churn Probability**: The estimated probability of churn for customers within that profile.
- **Rule Summary**: The set of decision rules (based on behavioral and demographic features) that define the segment.

Note: Segment IDs are assigned for reporting purposes and represent unique terminal nodes (customer profiles) derived from the surrogate tree. These identifiers do not correspond to actual customer IDs or appear in the tree plot.The following code translates the pruned surrogate tree into readable decision rules, assigns churn probabilities and risk categories and produces a structured table of customer segments for subsequent analysis.

🔵 Low Risk (Churn ≤ 25%)

Customer Segment	Churn Probability	Rule Summary
1	1%	Total_Trans_Ct>=54.5 & Total_Trans_Amt< 5364 & Total_Trans_Ct>=60.5 & Total_Trans_Amt>=2716
2	3%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct>=78.5 & Total_Ct_Chng_Q4_Q1< 0.95 & Total_Trans_Ct>=81.5
3	4%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct>=78.5 & Total_Ct_Chng_Q4_Q1>=0.95 & Total_Trans_Amt>=1.141e+04
4	6%	Total_Trans_Ct>=54.5 & Total_Trans_Amt< 5364 & Total_Trans_Ct>=60.5 & Total_Trans_Amt< 2716 & Credit_Limit>=1911
5	6%	Total_Trans_Ct>=54.5 & Total_Trans_Amt< 5364 & Total_Trans_Ct< 60.5 & Total_Relationship_Count>=2.5 & Avg_Utilization_Ratio>=0.0295
6	8%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio>=0.0265 & Total_Relationship_Count>=2.5 & Total_Trans_Amt< 2103 & Total_Trans_Amt>=967
7	20%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio>=0.0265 & Total_Relationship_Count>=2.5 & Total_Trans_Amt>=2103 & Total_Ct_Chng_Q4_Q1>=0.623
8	21%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio>=0.0265 & Total_Relationship_Count< 2.5 & Total_Ct_Chng_Q4_Q1>=0.703 & Total_Ct_Chng_Q4_Q1>=0.961
9	22%	Total_Trans_Ct>=54.5 & Total_Trans_Amt< 5364 & Total_Trans_Ct< 60.5 & Total_Relationship_Count>=2.5 & Avg_Utilization_Ratio< 0.0295

Customer Segment	Churn Probability	Rule Summary
10	22%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio< 0.0265 & Total_Ct_Chng_Q4_Q1>=0.646 & Total_Relationship_Count>=2.5 & Total_Trans_Amt< 1967
11	22%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio< 0.0265 & Total_Ct_Chng_Q4_Q1< 0.646 & Months_Inactive_12_mon< 1.5 & Contacts_Count_12_mon< 1.5
12	23%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct>=78.5 & Total_Ct_Chng_Q4_Q1< 0.95 & Total_Trans_Ct< 81.5

Medium Risk (25% < Churn < 65%)

Customer Segment	Churn Probability	Rule Summary
13	26%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct< 78.5 & Avg_Utilization_Ratio>=0.021 & Total_Trans_Ct>=68.5
14	31%	Total_Trans_Ct>=54.5 & Total_Trans_Amt< 5364 & Total_Trans_Ct< 60.5 & Total_Relationship_Count< 2.5 & Total_Trans_Ct>=57.5
15	32%	Total_Trans_Ct>=54.5 & Total_Trans_Amt< 5364 & Total_Trans_Ct>=60.5 & Total_Trans_Amt< 2716 & Credit_Limit< 1911
16	51%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio< 0.0265 & Total_Ct_Chng_Q4_Q1>=0.646 & Total_Relationship_Count>=2.5 & Total_Trans_Amt>=1967
17	56%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio>=0.0265 & Total_Relationship_Count< 2.5 & Total_Ct_Chng_Q4_Q1>=0.703 & Total_Ct_Chng_Q4_Q1< 0.961
18	58%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio< 0.0265 & Total_Ct_Chng_Q4_Q1< 0.646 & Months_Inactive_12_mon< 1.5 & Contacts_Count_12_mon>=1.5
19	60%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio>=0.0265 & Total_Relationship_Count>=2.5 & Total_Trans_Amt>=2103 & Total_Ct_Chng_Q4_Q1< 0.623
20	62%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct< 78.5 & Avg_Utilization_Ratio< 0.021 & Total_Amt_Chng_Q4_Q1< 0.804

Customer Segment	Churn Probability	Rule Summary
21	64%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio>=0.0265 & Total_Relationship_Count>=2.5 & Total_Trans_Amt< 2103 & Total_Trans_Amt< 967

🔴 High Risk (Churn ≥ 65%)

Customer Segment	Churn Probability	Rule Summary
22	65%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct>=78.5 & Total_Ct_Chng_Q4_Q1>=0.95 & Total_Trans_Amt< 1.141e+04
23	70%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct< 78.5 & Avg_Utilization_Ratio>=0.021 & Total_Trans_Ct< 68.5
24	71%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio< 0.0265 & Total_Ct_Chng_Q4_Q1< 0.646 & Months_Inactive_12_mon>=1.5 & Total_Trans_Amt< 1805
25	75%	Total_Trans_Ct>=54.5 & Total_Trans_Amt< 5364 & Total_Trans_Ct< 60.5 & Total_Relationship_Count< 2.5 & Total_Trans_Ct< 57.5
26	80%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio>=0.0265 & Total_Relationship_Count< 2.5 & Total_Ct_Chng_Q4_Q1< 0.703
27	86%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio< 0.0265 & Total_Ct_Chng_Q4_Q1>=0.646 & Total_Relationship_Count< 2.5
28	86%	Total_Trans_Ct>=54.5 & Total_Trans_Amt>=5364 & Total_Trans_Ct< 78.5 & Avg_Utilization_Ratio< 0.021 & Total_Amt_Chng_Q4_Q1>=0.804
29	90%	Total_Trans_Ct< 54.5 & Avg_Utilization_Ratio< 0.0265 & Total_Ct_Chng_Q4_Q1< 0.646 & Months_Inactive_12_mon>=1.5 & Total_Trans_Amt>=1805

5.3 Validating the Surrogate Tree against Random Forest predictions

In business applications, particularly within customer relationship management (CRM), it is not enough for a model to be accurate, it must also be interpretable. Stakeholders such as marketing teams, CRM managers, or customer analysts often require clear, actionable insights that justify model-driven decisions. While the Random Forest model offers high predictive performance, its ensemble architecture renders it a “black box” that is difficult to explain to non-technical audiences.

To bridge this gap, we construct a surrogate decision tree which is a simpler, interpretable model trained to approximate the predictions of the Random Forest. This approach aligns with the model-agnostic explainability strategy proposed by Craven and Shavlik (1996), in which a transparent model is used to mimic the output of a more complex one. By reproducing the Random Forest's decision patterns using a compact set of rules, the surrogate tree enables transparent, human-readable insights that preserve the core logic of the original model, making the predictions accessible and actionable for strategic use.

```
surrogate_preds <- predict(tree_pruned, newdata = df_surrogate)
r_squared <- cor(surrogate_preds, df_surrogate$Churn_Prob)^2
cat("R-squared vs RF predictions:", round(r_squared, 3), "\n")
```

```
## R-squared vs RF predictions: 0.746
```

With an **R^2 of 0.746**, the surrogate tree captures about 75% of the variation in the Random Forest's predicted churn probabilities. This shows that the tree reflects most of the key decision patterns used by the more complex model. In practice, this level of accuracy is considered good for interpretability purposes. Previous research has shown that surrogate models can still be useful even if they don't perfectly match the original model, as long as they help explain how decisions are made (Craven & Shavlik, 1996; Ribeiro et al., 2016).

Of course, some of the more detailed interactions from the Random Forest are lost in the simplified tree (Breiman et al., 1984), but the benefit is a **clear and readable set of rules**. These rules can be used by non-technical teams, such as CRM or marketing, to understand customer behavior and take targeted action. The surrogate model strikes a good balance between accuracy and simplicity, making it easier to turn predictions into real business decisions. This simplified interpretation of Random Forest decisions forms the basis for the segment-level CRM strategies discussed in the next chapter.

6. Conclusion and Recommendations

This analysis successfully developed and evaluated predictive models to identify customers at risk of churn in the bank's credit card portfolio. The results offer both analytical depth and actionable business insights.

Key findings include:

- The **Random Forest** model achieved the highest overall performance, effectively capturing complex behavioral patterns with superior recall and AUC compared to Logistic Regression.
- The **Logistic Regression** model offered interpretable coefficients, allowing for transparent identification of key churn drivers such as low transaction activity, few product relationships, and extreme credit utilization.
- A **surrogate decision tree**, combined with customer clustering, revealed 29 distinct behavioral segments. These were grouped into **Low**, **Medium**, and **High churn risk** tiers, enabling targeted Customer Relationship Management (CRM) strategies for each group.

6.1 Key Business Insights and Recommended Actions for the Bank

Using the segments derived from the pruned surrogate decision tree, we propose a **targeted CRM strategy** tailored to the three churn risk tiers. The goal is to:

- Retain high-risk customers before they leave
- Strengthen engagement among medium-risk customers
- Maintain loyalty and advocacy in low-risk customers

🔴 High-Risk Customers

Traits:

- **Fewer than 54 card transactions per year**
- **Low product relationship depth (often only 1–2 products)**
- **Periods of inactivity** exceeding **2 months**
- **Very frequent contacts with customer service**, suggesting unresolved issues
- **Extremely low** (under 2.5%) or **very high** (over 40%) **credit utilization**, indicating financial misalignment
- Notably, **Segment 25 and 26** showed churn probabilities over **90%**, driven by either:
  - High engagement **but** excessive contact frequency (suggesting frustration)
  - Minimal engagement across all metrics (likely disengaged customers)

Business Impact:

- Imminent churn threat with **serious revenue implications**
- Losing them could erode **lifetime value and share of wallet**
- Requires **immediate, targeted action** to prevent attrition

Recommended CRM Actions:

Action	Justification
Proactive Retention Offers	Offer targeted deals (e.g., cashback, fee waivers) to dissuade exit intentions
Direct Personal Outreach	Conduct phone calls or personal check-ins to uncover pain points
Credit & Product Review	Assess needs for credit increases or switch to better-suited account types
Time-Sensitive Perks	Use urgency-based incentives (e.g., 7-day bonus offer) to re-engage inactive users

🟡 Medium-Risk Customers

Traits:

Customers in this tier exhibit **ambiguous engagement patterns**, requiring proactive but nuanced retention efforts:

- Some maintain **moderate transaction activity** and hold 2–3 products, but lack deeper financial or emotional commitment.
- Others show **rising contact frequency**, which may indicate emerging dissatisfaction or service-related friction.
- **Inconsistent credit utilization** is common, some underutilize their available credit while others cross high usage thresholds.
- A portion also displays **mild inactivity**, signaling early disengagement risks.

Business Impact:

- Medium-risk customers account for a **large share of potential churn**.
- Their uncertainty raises **CRM costs** and complicates **retention planning**.
- Neglecting them may lead to **gradual profit erosion**.

Recommended CRM Actions:

Action	Justification
Usage Nudge Campaigns	Encourage greater interaction with existing products (e.g., increase card use)
Feature Education	Highlight underused features like mobile banking or autopay setup
Relationship Bundling	Offer additional products to deepen the relationship (e.g., savings, loans)
Micro-Targeted Offers	Provide tailored offers that address usage gaps or customer needs

🔵 Low-Risk Customers

Traits:

- Typically **high card activity** at least 65 transactions per year
- **Moderate credit utilization**, with some variance across segments
- Often maintain **2–3 product relationships**
- When available, contact frequency is **fewer than 5 transactions per year**

Business Impact:

- Minimal churn probability
- Important to **retain due to profitability and cross-sell potential**
- Risk lies in **complacency**, neglect may lead to long-term attrition or competitor migration

Recommended CRM Actions:

Action	Justification
Loyalty Rewards	Acknowledge and reward loyal customers to reinforce retention
Feedback Collection	Gather insights and show appreciation via surveys or NPS prompts
VIP Segmentation	Label as high-value for priority service or exclusive offers
Referral Incentives	Encourage peer referrals with joint benefits

Strategic Summary

Risk Group	CRM Priority	Action Focus	Primary Goal
High	Immediate	Retention & Re-engagement	Prevent imminent churn
Medium	Proactive	Education & Relationship Deepening	Strengthen customer ties
Low	Maintenance	Recognition & Advocacy	Preserve loyalty, gain referrals

6.2 Limitations of This Study

- **No time-based transaction data** was available, so we couldn't model trends or seasonality.
- **Assumptions of current feature relevance** may not hold over time, especially with shifting customer behavior or economic conditions.
- The model was trained on a snapshot, future data drift may affect accuracy.

While the model performs well on historical data, it is limited by potential bias in feature availability and lack of real-time behavioral signals. Future integration with real-time transaction and customer service data could further enhance accuracy and responsiveness.

6.3 Future Work Suggestions

The recommended future methods are better suited than traditional approaches because they offer higher accuracy, deeper insights, and more actionable outcomes. Incorporating time-series features improves prediction by capturing changes in customer behavior over time, something static metrics can't reflect. Advanced models like XGBoost and LightGBM could be used in a future work to explore if they outperform Random Forest and logistic regression. These models could identify complex patterns and correct errors through boosting and may make them more accurate for structured churn data. Adding SHAP or LIME may enhance transparency by explaining how each feature contributes to a prediction, unlike black-box models that lack interpretability.

To maximize the business impact of these improvements, the model should be integrated into a **real-time scoring pipeline** connected to the bank's CRM system. Churn probabilities can be recalculated monthly or after significant behavioral shifts (for example drop in transaction activity), automatically triggering segment-specific CRM interventions such as retention offers or upselling campaigns. This ensures the organization can act promptly rather than relying on static reports that delay intervention.

To implement this churn prediction framework effectively, the following steps are recommended:

- **Incorporate time-series features** such as monthly transaction trends, missed payments, or seasonal usage changes.
- Explore **advanced ensemble models** like XGBoost or LightGBM for even higher performance and feature interpretability.
- **Deploy the model** into a real-time CRM environment that triggers alerts and actions based on churn risk updates.
- Add **explainable AI layers** like SHAP or LIME to interpret churn risk scores and support internal reporting.

Overall, this study provides a solid foundation for data-driven customer retention initiatives, balancing predictive accuracy with business interpretability.

References

- Breiman, L. (2001). *Random Forests*. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324 (https://doi.org/10.1023/A:1010933404324)
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. CRC Press.
- Craven, M. W., & Shavlik, J. W. (1996). *Extracting tree-structured representations of trained networks*. In D. Touretzky, M. Mozer, & M. Hasselmo (Eds.), *Advances in Neural Information Processing Systems* (Vol. 8, pp. 24–30). MIT Press.
- Gallo, A. (2014). *The value of keeping the right customers*. *Harvard Business Review*.
- Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). Wiley.
- Hwang, H., Jung, T., & Suh, E. (2004). *An LTV model and customer segmentation based on customer value: A case study on the wireless telecommunication industry*. *Expert Systems with Applications*, 26(2), 181–188.
- Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), 20150202.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3 (https://doi.org/10.1007/978-1-4614-6849-3)
- Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/ (https://christophm.github.io/interpretable-ml-book/)
- Neslin, S. A., Gupta, S., Kamakura, W., Lu, J., & Mason, C. H. (2006). *Defection detection: Measuring and understanding the predictive accuracy of customer churn models*. *Journal of Marketing Research*, 43(2), 204–211. https://doi.org/10.1509/jmkr.43.2.204 (https://doi.org/10.1509/jmkr.43.2.204)
- Ngai, E. W. T., Xiu, L., & Chau, D. C. K. (2009). *Application of data mining techniques in customer relationship management: A literature review and classification*. *Expert Systems with Applications*, 36(2), 2592–2602.
- Reichheld, F. F., & Sasser, W. E. (1990). *Zero defections: Quality comes to services*. *Harvard Business Review*.

- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *“Why Should I Trust You?”: Explaining the Predictions of Any Classifier*. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778 (https://doi.org/10.1145/2939672.2939778)