Risk Analysis Project

2024-11-27

```
library(readxl)

## Warning: package 'readxl' was built under R version 4.3.2

# fetch training data
file_path <-
"C:/Users/nandi/Documents/Personal/Academics/Final/kaggle/archive/Credit_Risk
_Data_Set/train_FIN_ANA_DATA.xls"
file.exists(file_path)

## [1] TRUE

# Load the data
financial_train_data <- read_excel(file_path)
head(financial_train_data)

## # A tibble: 6 × 11
##   ACC_NO        INVESTMENT_TOTAL ACCCURRENTBALANCE INF_MARITAL_STATUS
INF_GENDER
##   <chr>                    <dbl>             <dbl> <chr>
<chr>
## 1 0027010017245         10720596            585913 M                  F
## 2 0027010017436         43455000            585913 M                  F
## 3 0027010017458         22012402             68348 M                  F
## 4 0027010017493          4893983                 0 M                  M
## 5 0027010017515         46254814             68348 M                  F
## 6 0027010017537         54562500             68348 M                  F
## # ℹ 6 more variables: INSTALL_SIZE <dbl>, DUE_PAYMENT <dbl>,
## #   COMPENSATION_CHARGED <chr>, CLIENT_TYPE <chr>, QUALITY_OF_LOAN <chr>,
## #   REPAY_MODE <chr>

dim(financial_train_data)

## [1] 37408    11

##37408 rows and   11 attributes

##preprocessing steps

##identifying na values
sum(is.na(financial_train_data))

## [1] 947
```

```
# There are 947 NA values identified in the training dataset
sum(is.na(financial_train_data$REPAY_MODE))
```

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

```
# INF_MARITAL_STATUS, INF_GENDER, INSTALL_SIZE, COMPENSATION_CHARGED,
CLIENT_TYPE

colSums(is.na(financial_train_data))
```

```
##               ACC_NO     INVESTMENT_TOTAL    ACCCURRENTBALANCE
##                    0                    0                    0
##   INF_MARITAL_STATUS           INF_GENDER         INSTALL_SIZE
##                    2                    2                  838
##          DUE_PAYMENT COMPENSATION_CHARGED          CLIENT_TYPE
##                    0                    2                  103
##      QUALITY_OF_LOAN           REPAY_MODE
##                    0                    0
```

```
# Applying Imputation Approach to fill the missing values in the dataset

# Updating INF_MARITAL_STATUS column NA values
mode_marital_status <-
names(sort(table(financial_train_data$INF_MARITAL_STATUS), decreasing =
TRUE))[1]
financial_train_data$INF_MARITAL_STATUS[is.na(financial_train_data$INF_MARITA
L_STATUS)] <- mode_marital_status



# Updating INF_GENDER column NA values
mode_INF_GENDER <- names(sort(table(financial_train_data$INF_GENDER),
decreasing = TRUE))[1]
financial_train_data$INF_GENDER[is.na(financial_train_data$INF_GENDER)] <-
mode_INF_GENDER

# The following command can be used to study the data set, fetch the unique
values in a column

# unique(financial_train_data$INSTALL_SIZE)


# Updating COMPENSATION_CHARGED column NA values
mode_COMPENSATION_CHARGED <-
names(sort(table(financial_train_data$COMPENSATION_CHARGED), decreasing =
TRUE))[1]
financial_train_data$COMPENSATION_CHARGED[is.na(financial_train_data$COMPENSA
TION_CHARGED)] <- mode_COMPENSATION_CHARGED
```

```r
financial_train_data$CLIENT_TYPE[financial_train_data$CLIENT_TYPE == "0"] <-
NA


#since CLIENT_TYPE has 103 missing values, we can impute the missing values
with the consistent distribution available in sample dataset.

distribution <- table(financial_train_data$CLIENT_TYPE, useNA = "no")
probabilities <- prop.table(distribution)
set.seed(123)  # For reproducibility
financial_train_data$CLIENT_TYPE[is.na(financial_train_data$CLIENT_TYPE)] <-
sample(
  names(probabilities),
  size = sum(is.na(financial_train_data$CLIENT_TYPE)),
  replace = TRUE,
  prob = probabilities
)



# Updating INSTALL_SIZE column NA values
# here we are updating the NA values with median of the column, with
respecctive of their client type

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.3

##
## Attaching package: 'dplyr'

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

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

financial_train_data <- financial_train_data %>%
  group_by(CLIENT_TYPE) %>%
  mutate(INSTALL_SIZE = ifelse(is.na(INSTALL_SIZE),
                               median(INSTALL_SIZE, na.rm = TRUE),
                               INSTALL_SIZE))

# reChecking for NA values
colSums(is.na(financial_train_data))

##               ACC_NO     INVESTMENT_TOTAL    ACCCURRENTBALANCE
##                    0                    0                    0
##   INF_MARITAL_STATUS           INF_GENDER         INSTALL_SIZE
##                    0                    0                    0
##          DUE_PAYMENT COMPENSATION_CHARGED          CLIENT_TYPE
##                    0                    0                    0
##      QUALITY_OF_LOAN           REPAY_MODE
##                    0                    0
```

```r
# after imputation there are no NA values for the complete training dataset


# head(financial_train_data)
# colnames(financial_train_data)


# fixing the column datatypes, converting factor columns


financial_train_data <- financial_train_data %>%
  mutate(
    INF_MARITAL_STATUS = as.factor(INF_MARITAL_STATUS),
    INF_GENDER = as.factor(INF_GENDER),
    CLIENT_TYPE = as.factor(CLIENT_TYPE),
    QUALITY_OF_LOAN = as.factor(QUALITY_OF_LOAN),
    REPAY_MODE = as.factor(REPAY_MODE)
  )


summary(financial_train_data)
```

```
##     ACC_NO            INVESTMENT_TOTAL    ACCCURRENTBALANCE
INF_MARITAL_STATUS
##  Length:37408       Min.   :5.002e+05   Min.   :        0   M:35414
##  Class :character   1st Qu.:8.368e+05   1st Qu.:     2178   O:   31
##  Mode  :character   Median :1.635e+06   Median :    24484   U: 1963
##                     Mean   :6.204e+06   Mean   :  1174438
##                     3rd Qu.:4.365e+06   3rd Qu.:   341639
##                     Max.   :1.509e+09   Max.   :217415344
##  INF_GENDER  INSTALL_SIZE        DUE_PAYMENT          COMPENSATION_CHARGED
##  F: 9648    Min.   :       0   Min.   :        0   Length:37408
##  M:27758    1st Qu.:       0   1st Qu.:        0   Class :character
##  O:    2    Median :       0   Median :        0   Mode  :character
##              Mean   :   44674   Mean   :   375821
##              3rd Qu.:       0   3rd Qu.:        0
##              Max.   :59844373   Max.   :370192428
##      CLIENT_TYPE     QUALITY_OF_LOAN REPAY_MODE
##  Rural     :26219   B: 4154         I: 5825
##  Semi-urban: 8553   G:33254         N:31583
##  Urban     : 2636
##
```

```
##
##

# checking for duplicate record - 0 duplicates
count(financial_train_data)

## # A tibble: 3 × 2
## # Groups:   CLIENT_TYPE [3]
##   CLIENT_TYPE     n
##   <fct>       <int>
## 1 Rural       26219
## 2 Semi-urban   8553
## 3 Urban        2636

count(financial_train_data %>% distinct())

## # A tibble: 3 × 2
## # Groups:   CLIENT_TYPE [3]
##   CLIENT_TYPE     n
##   <fct>       <int>
## 1 Rural       26219
## 2 Semi-urban   8553
## 3 Urban        2636

financial_train_data %>%
  group_by(ACC_NO) %>%
  filter(n() > 1)

## # A tibble: 0 × 11
## # Groups:   ACC_NO [0]
## # ℹ 11 variables: ACC_NO <chr>, INVESTMENT_TOTAL <dbl>,
## #   ACCCURRENTBALANCE <dbl>, INF_MARITAL_STATUS <fct>, INF_GENDER <fct>,
## #   INSTALL_SIZE <dbl>, DUE_PAYMENT <dbl>, COMPENSATION_CHARGED <chr>,
## #   CLIENT_TYPE <fct>, QUALITY_OF_LOAN <fct>, REPAY_MODE <fct>

# Feature Engineering

investment_bins <- c(-Inf, 8.368e+05, 4.365e+06, Inf)
investment_labels <- c("Small", "Medium", "Large")

due_payment_bins <- c(-Inf, 0, 1e+06, Inf)
due_payment_labels <- c("No Payment Due", "Low Due", "High Due")

install_size_bins <- c(-Inf, 0, 1e+05, Inf)
install_size_labels <- c("No Install", "Small", "Large")

account_balance_bins <- c(-Inf, 0, 2.178e+03, 3.416e+05, Inf)
account_balance_labels <- c("Zero", "Low", "Moderate", "High")

# Binning INVESTMENT_TOTAL
financial_train_data$INVESTMENT_BIN <- cut(
```

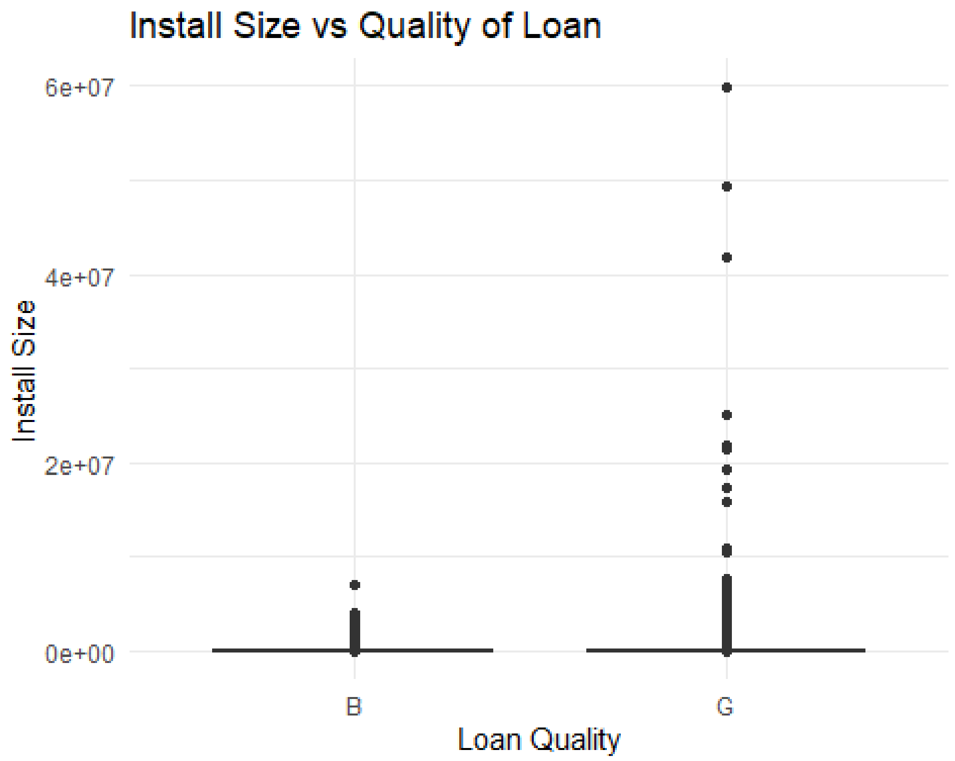
```r
  financial_train_data$INVESTMENT_TOTAL,
  breaks = investment_bins,
  labels = investment_labels,
  right = FALSE
)

# Binning DUE_PAYMENT
financial_train_data$DUE_PAYMENT_BIN <- cut(
  financial_train_data$DUE_PAYMENT,
  breaks = due_payment_bins,
  labels = due_payment_labels,
  right = FALSE
)

# Binning INSTALL_SIZE
financial_train_data$INSTALL_SIZE_BIN <- cut(
  financial_train_data$INSTALL_SIZE,
  breaks = install_size_bins,
  labels = install_size_labels,
  right = FALSE
)

# Binning ACCCURRENTBALANCE
financial_train_data$BALANCE_BIN <- cut(
  financial_train_data$ACCCURRENTBALANCE,
  breaks = account_balance_bins,
  labels = account_balance_labels,
  right = FALSE
)

# converting Y or N values in COMPENSATION_CHARGED column to 1 and 0
respectively

financial_train_data <- financial_train_data %>%
  mutate(COMPENSATION_CHARGED = ifelse(COMPENSATION_CHARGED == "Y", 1, 0))


summary(financial_train_data[, c("INVESTMENT_BIN", "DUE_PAYMENT_BIN",
"INSTALL_SIZE_BIN", "BALANCE_BIN")])

##  INVESTMENT_BIN       DUE_PAYMENT_BIN    INSTALL_SIZE_BIN   BALANCE_BIN
##  Small : 9352   No Payment Due:    0   No Install:    0   Zero    :    0
##  Medium:18551   Low Due       :35688   Small     :35437   Low     : 9349
##  Large : 9505   High Due      : 1720   Large     : 1971   Moderate:18700
##                                                           High    : 9359

# setting  "Rural", "Semi-urban", "Urban"to 1,2,3 numeric levels
financial_train_data$CLIENT_TYPE <- factor(financial_train_data$CLIENT_TYPE,
                                           levels = c("Rural", "Semi-urban",
"Urban"))
```

```
financial_train_data$CLIENT_TYPE <-
as.numeric(financial_train_data$CLIENT_TYPE)

# Preprocessing test data separately



# Load the test data
test_file_path <-
"C:/Users/nandi/Documents/Personal/Academics/Final/kaggle/archive/Credit_Risk
_Data_Set/test_FIN_ANA_DATA.xls"
file.exists(test_file_path)

## [1] TRUE

financial_test_data <- read_excel(test_file_path)
head(financial_test_data)

## # A tibble: 6 × 11
##   ACC_NO        INVESTMENT_TOTAL ACCCURRENTBALANCE INF_MARITAL_STATUS
INF_GENDER
##   <chr>                    <dbl>             <dbl> <chr>
<chr>
## 1 1598350000464           641740              1038 M                  M
## 2 1598350000475           532125              4310 M                  M
## 3 1598350000486           632625              4310 M                  M
## 4 1598350000622          1967250              5114 M                  M
## 5 1598350000655          1636875              2787 M                  M
## 6 1598350000666          1636875              2787 M                  M
## # ℹ 6 more variables: INSTALL_SIZE <dbl>, DUE_PAYMENT <dbl>,
## #   COMPENSATION_CHARGED <chr>, CLIENT_TYPE <chr>, QUALITY_OF_LOAN <chr>,
## #   REPAY_MODE <chr>

# Check dimensions of test data
dim(financial_test_data)

## [1] 4310   11

# Checking NA values in test data
colSums(is.na(financial_test_data))

##               ACC_NO     INVESTMENT_TOTAL    ACCCURRENTBALANCE
##                    0                    0                    0
##   INF_MARITAL_STATUS           INF_GENDER         INSTALL_SIZE
##                    0                    0                    1
##          DUE_PAYMENT COMPENSATION_CHARGED          CLIENT_TYPE
##                    0                    0                   79
##      QUALITY_OF_LOAN           REPAY_MODE
##                    0                    0
```

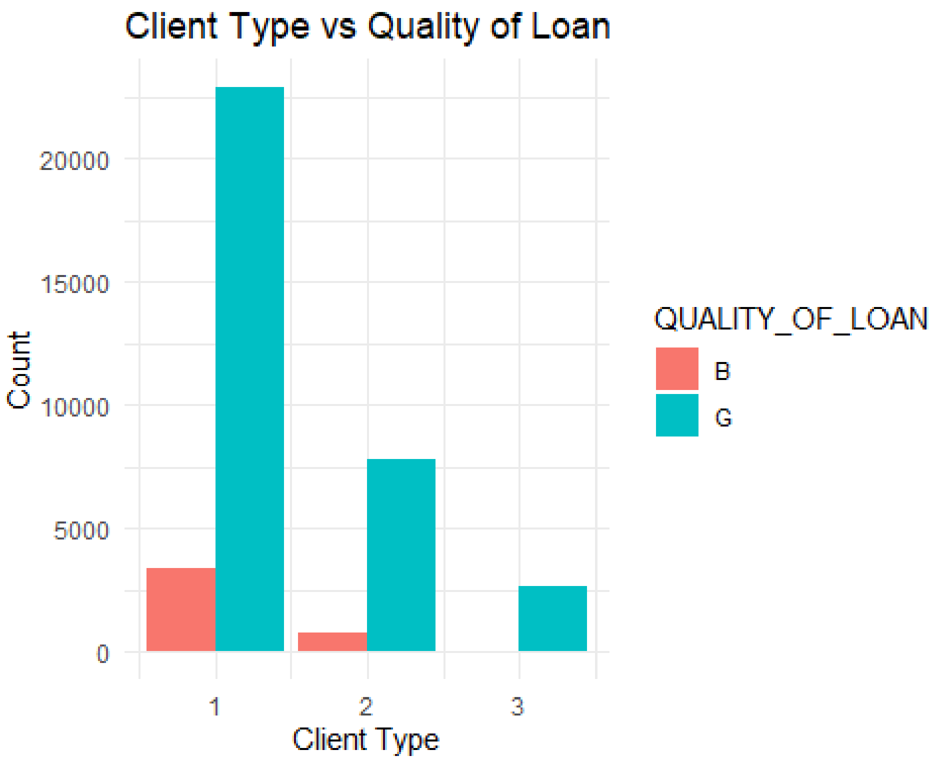
```
# === Preprocessing Test Data ===

## INF_MARITAL_STATUS: Impute with mode from training data
financial_test_data$INF_MARITAL_STATUS[is.na(financial_test_data$INF_MARITAL_
STATUS)] <- mode_marital_status

## INF_GENDER: Impute with mode from training data
financial_test_data$INF_GENDER[is.na(financial_test_data$INF_GENDER)] <-
mode_INF_GENDER

## COMPENSATION_CHARGED: Impute with mode from training data
financial_test_data$COMPENSATION_CHARGED[is.na(financial_test_data$COMPENSATI
ON_CHARGED)] <- mode_COMPENSATION_CHARGED



## INSTALL_SIZE: Impute missing values with median by CLIENT_TYPE from
training data
financial_test_data <- financial_test_data %>%
  group_by(CLIENT_TYPE) %>%
  mutate(INSTALL_SIZE = ifelse(is.na(INSTALL_SIZE),
                               median(INSTALL_SIZE, na.rm = TRUE),
                               INSTALL_SIZE)) %>%
  ungroup()

# Recheck for missing values in test data
colSums(is.na(financial_test_data))

##               ACC_NO     INVESTMENT_TOTAL    ACCCURRENTBALANCE
##                    0                    0                    0
##   INF_MARITAL_STATUS           INF_GENDER         INSTALL_SIZE
##                    0                    0                    0
##          DUE_PAYMENT COMPENSATION_CHARGED          CLIENT_TYPE
##                    0                    0                   79
##      QUALITY_OF_LOAN           REPAY_MODE
##                    0                    0

# === Fixing Column Datatypes ===
financial_test_data <- financial_test_data %>%
  mutate(
    INF_MARITAL_STATUS = as.factor(INF_MARITAL_STATUS),
    INF_GENDER = as.factor(INF_GENDER),
    CLIENT_TYPE = as.factor(CLIENT_TYPE),
    QUALITY_OF_LOAN = as.factor(QUALITY_OF_LOAN),
    REPAY_MODE = as.factor(REPAY_MODE)
  )

# === Feature Engineering for Test Data ===
financial_test_data$INVESTMENT_BIN <- cut(
  financial_test_data$INVESTMENT_TOTAL,
  breaks = investment_bins,
  labels = investment_labels,
  right = FALSE
)

financial_test_data$DUE_PAYMENT_BIN <- cut(
  financial_test_data$DUE_PAYMENT,
  breaks = due_payment_bins,
  labels = due_payment_labels,
  right = FALSE
)

financial_test_data$INSTALL_SIZE_BIN <- cut(
  financial_test_data$INSTALL_SIZE,
  breaks = install_size_bins,
  labels = install_size_labels,
  right = FALSE
)

financial_test_data$BALANCE_BIN <- cut(
  financial_test_data$ACCCURRENTBALANCE,
  breaks = account_balance_bins,
  labels = account_balance_labels,
  right = FALSE
)

# converting Y or N values in COMPENSATION_CHARGED column to 1 and 0
respectively

financial_test_data <- financial_test_data %>%
  mutate(COMPENSATION_CHARGED = ifelse(COMPENSATION_CHARGED == "Y", 1, 0))

unique(financial_test_data$CLIENT_TYPE)

## [1] Rural      Urban      Semi-Urban <NA>
## Levels: Rural Semi-Urban Urban

## CLIENT_TYPE: Impute missing values using the high frequency value as test
data

frequency_table <- table(financial_test_data$CLIENT_TYPE)

# Get the most frequent category (mode)
mode_value <- names(frequency_table)[which.max(frequency_table)]

# Replace NA values with the mode
financial_test_data$CLIENT_TYPE[is.na(financial_test_data$CLIENT_TYPE)] <-
mode_value
```

```
#
# setting  "Rural", "Semi-urban", "Urban"to 1,2,3 numeric levels
financial_test_data$CLIENT_TYPE <- factor(financial_test_data$CLIENT_TYPE,
                                           levels = c("Rural", "Semi-Urban",
"Urban"))


financial_test_data$CLIENT_TYPE <-
as.numeric(financial_test_data$CLIENT_TYPE)

# === Summary of Test Data ===
summary(financial_test_data)

##     ACC_NO           INVESTMENT_TOTAL    ACCCURRENTBALANCE
INF_MARITAL_STATUS
##  Length:4310        Min.   :   500089   Min.   :       0   M:4077
##  Class :character   1st Qu.:   654750   1st Qu.:    1992   O:   1
##  Mode  :character   Median :  1094250   Median :   11134   U: 232
##                     Mean   :  2586344   Mean   :  640463
##                     3rd Qu.:  2424323   3rd Qu.:  274304
##                     Max.   :633229341   Max.   :20122327
##  INF_GENDER  INSTALL_SIZE       DUE_PAYMENT       COMPENSATION_CHARGED
##  F: 655     Min.   :       0   Min.   :       0   Min.   :0.0000
##  M:3655     1st Qu.:       0   1st Qu.:       0   1st Qu.:0.0000
##             Median :       0   Median :       0   Median :0.0000
##             Mean   :   15190   Mean   :  206648   Mean   :0.0891
##             3rd Qu.:       0   3rd Qu.:       0   3rd Qu.:0.0000
##             Max.   :12270150   Max.   :75000000   Max.   :1.0000
##   CLIENT_TYPE    QUALITY_OF_LOAN REPAY_MODE INVESTMENT_BIN
##  Min.   :1.000   B :  19         I: 548     Small :1445
##  1st Qu.:1.000   DF:   4         N:3762     Medium:2276
##  Median :1.000   G :4286                    Large : 589
##  Mean   :1.433   SS:   1
##  3rd Qu.:1.000
##  Max.   :3.000
##         DUE_PAYMENT_BIN   INSTALL_SIZE_BIN   BALANCE_BIN
##  No Payment Due:   0   No Install:   0    Zero    :   0
##  Low Due       :4172   Small     :4238    Low     :1150
##  High Due      : 138   Large     :  72    Moderate:2190
##                                           High    : 970
##
##

#Exploratory Data Analysis


# Creating Visualizations to understand the relationships between independent
variables and the target variable

# Target variable -  (QUALITY_OF_LOAN).
```

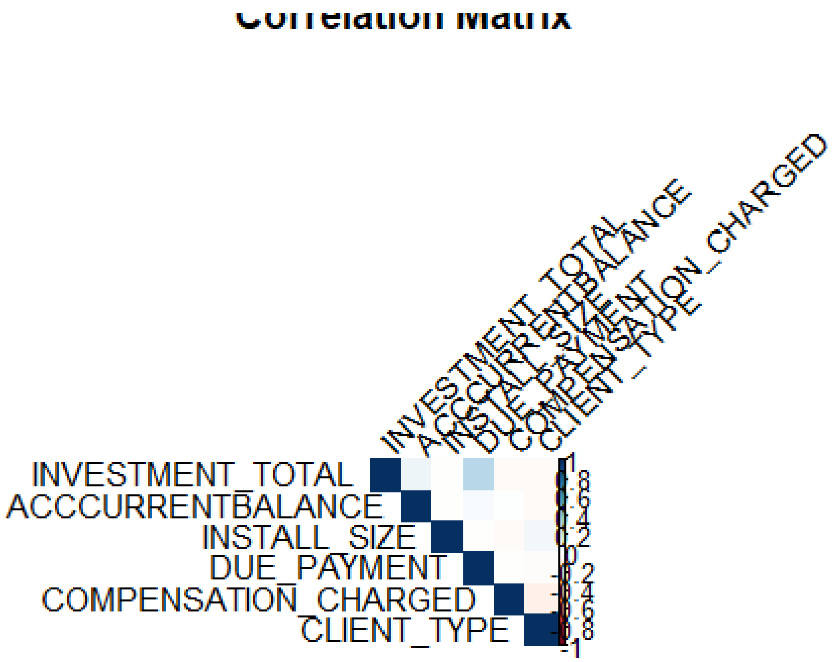
```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

ggplot(financial_train_data, aes(x = QUALITY_OF_LOAN, y = INSTALL_SIZE)) +
  geom_boxplot(fill = "skyblue") +
  labs(title = "Install Size vs Quality of Loan", x = "Loan Quality", y =
"Install Size") +
  theme_minimal()
```



```
ggplot(financial_train_data, aes(x = CLIENT_TYPE, fill = QUALITY_OF_LOAN)) +
  geom_bar(position = "dodge") +
  labs(title = "Client Type vs Quality of Loan", x = "Client Type", y =
"Count") +
  theme_minimal()
```



```
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.3.3

## corrplot 0.92 loaded

numeric_data <- financial_train_data %>%
  select_if(is.numeric)
# correlation matrix
cor_matrix <- cor(numeric_data, use = "complete.obs")

# Visualize the correlation matrix
library(corrplot)
corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45, title = "Correlation Matrix")
```



```
financial_train_data$CLIENT_TYPE <-
as.numeric(financial_train_data$CLIENT_TYPE)
sapply(numeric_data, class)

##     INVESTMENT_TOTAL    ACCCURRENTBALANCE          INSTALL_SIZE
##            "numeric"            "numeric"             "numeric"
##          DUE_PAYMENT COMPENSATION_CHARGED           CLIENT_TYPE
##            "numeric"            "numeric"             "numeric"

summary(financial_train_data)

##     ACC_NO           INVESTMENT_TOTAL    ACCCURRENTBALANCE
INF_MARITAL_STATUS
##  Length:37408       Min.   :5.002e+05   Min.   :        0   M:35414
##  Class :character   1st Qu.:8.368e+05   1st Qu.:     2178   O:   31
##  Mode  :character   Median :1.635e+06   Median :    24484   U: 1963
##                     Mean   :6.204e+06   Mean   :  1174438
##                     3rd Qu.:4.365e+06   3rd Qu.:   341639
##                     Max.   :1.509e+09   Max.   :217415344
##  INF_GENDER  INSTALL_SIZE        DUE_PAYMENT         COMPENSATION_CHARGED
##  F: 9648    Min.   :       0   Min.   :        0   Min.   :0.0000
##  M:27758    1st Qu.:       0   1st Qu.:        0   1st Qu.:0.0000
##  O:    2    Median :       0   Median :        0   Median :0.0000
##             Mean   :   44674   Mean   :   375821   Mean   :0.4467
##             3rd Qu.:       0   3rd Qu.:        0   3rd Qu.:1.0000
##             Max.   :59844373   Max.   :370192428   Max.   :1.0000
##   CLIENT_TYPE   QUALITY_OF_LOAN REPAY_MODE INVESTMENT_BIN
##  Min.   :1.00   B: 4154         I: 5825    Small : 9352
##  1st Qu.:1.00   G:33254         N:31583    Medium:18551
##  Median :1.00                              Large : 9505
##  Mean   :1.37
##  3rd Qu.:2.00
##  Max.   :3.00
##        DUE_PAYMENT_BIN     INSTALL_SIZE_BIN    BALANCE_BIN
##  No Payment Due:    0   No Install:     0   Zero    :    0
##  Low Due       :35688   Small     :35437   Low     : 9349
##  High Due      : 1720   Large     : 1971   Moderate:18700
##                                            High    : 9359
##
##
```

```
head(financial_train_data)
```

```
## # A tibble: 6 × 15
## # Groups:   CLIENT_TYPE [2]
##   ACC_NO        INVESTMENT_TOTAL ACCCURRENTBALANCE INF_MARITAL_STATUS
INF_GENDER
##   <chr>                    <dbl>             <dbl> <fct>
<fct>
## 1 0027010017245         10720596            585913 M                  F
## 2 0027010017436         43455000            585913 M                  F
## 3 0027010017458         22012402             68348 M                  F
## 4 0027010017493          4893983                 0 M                  M
## 5 0027010017515         46254814             68348 M                  F
## 6 0027010017537         54562500             68348 M                  F
## # ℹ 10 more variables: INSTALL_SIZE <dbl>, DUE_PAYMENT <dbl>,
## #   COMPENSATION_CHARGED <dbl>, CLIENT_TYPE <dbl>, QUALITY_OF_LOAN <fct>,
## #   REPAY_MODE <fct>, INVESTMENT_BIN <fct>, DUE_PAYMENT_BIN <fct>,
## #   INSTALL_SIZE_BIN <fct>, BALANCE_BIN <fct>
```

```
financial_train_data$QUALITY_OF_LOAN <-
ifelse(financial_train_data$QUALITY_OF_LOAN == "G", 1, 0)
financial_test_data$QUALITY_OF_LOAN <-
ifelse(financial_test_data$QUALITY_OF_LOAN == "G", 1, 0)


unique(financial_train_data$CLIENT_TYPE)
```

```
## [1] 2 1 3
```

```
head(financial_train_data[,2:15])
```

```
## # A tibble: 6 × 14
## # Groups:   CLIENT_TYPE [2]
##   INVESTMENT_TOTAL ACCCURRENTBALANCE INF_MARITAL_STATUS INF_GENDER
INSTALL_SIZE
##              <dbl>             <dbl> <fct>              <fct>
<dbl>
```

```
## 1         10720596           585913 M                  F
0
## 2         43455000           585913 M                  F
0
## 3         22012402            68348 M                  F
0
## 4          4893983                0 M                  M
0
## 5         46254814            68348 M                  F
0
## 6         54562500            68348 M                  F
0
## # ℹ 9 more variables: DUE_PAYMENT <dbl>, COMPENSATION_CHARGED <dbl>,
## #   CLIENT_TYPE <dbl>, QUALITY_OF_LOAN <dbl>, REPAY_MODE <fct>,
## #   INVESTMENT_BIN <fct>, DUE_PAYMENT_BIN <fct>, INSTALL_SIZE_BIN <fct>,
## #   BALANCE_BIN <fct>
```

```
head(financial_test_data)
```

```
## # A tibble: 6 × 15
##   ACC_NO        INVESTMENT_TOTAL ACCCURRENTBALANCE INF_MARITAL_STATUS
INF_GENDER
##   <chr>                    <dbl>             <dbl> <fct>
<fct>
## 1 1598350000464           641740              1038 M                  M
## 2 1598350000475           532125              4310 M                  M
## 3 1598350000486           632625              4310 M                  M
## 4 1598350000622          1967250              5114 M                  M
## 5 1598350000655          1636875              2787 M                  M
## 6 1598350000666          1636875              2787 M                  M
## # ℹ 10 more variables: INSTALL_SIZE <dbl>, DUE_PAYMENT <dbl>,
## #   COMPENSATION_CHARGED <dbl>, CLIENT_TYPE <dbl>, QUALITY_OF_LOAN <dbl>,
## #   REPAY_MODE <fct>, INVESTMENT_BIN <fct>, DUE_PAYMENT_BIN <fct>,
## #   INSTALL_SIZE_BIN <fct>, BALANCE_BIN <fct>
```

```
unique(financial_train_data[,2:15])
```

```
## # A tibble: 32,478 × 14
## # Groups:   CLIENT_TYPE [3]
##    INVESTMENT_TOTAL ACCCURRENTBALANCE INF_MARITAL_STATUS INF_GENDER
INSTALL_SIZE
##               <dbl>             <dbl> <fct>              <fct>
<dbl>
##  1         10720596            585913 M                  F
0
##  2         43455000            585913 M                  F
0
##  3         22012402             68348 M                  F
0
##  4          4893983                 0 M                  M
0
```

```
##  5         46254814             68348 M                  F
0
##  6         54562500             68348 M                  F
0
##  7         21825000             68348 M                  F
0
##  8         10912500             68348 M                  F
0
##  9         11299894             68348 M                  F
0
## 10         11310806             68348 M                  F
0
## # ℹ 32,468 more rows
## # ℹ 9 more variables: DUE_PAYMENT <dbl>, COMPENSATION_CHARGED <dbl>,
## #   CLIENT_TYPE <dbl>, QUALITY_OF_LOAN <dbl>, REPAY_MODE <fct>,
## #   INVESTMENT_BIN <fct>, DUE_PAYMENT_BIN <fct>, INSTALL_SIZE_BIN <fct>,
## #   BALANCE_BIN <fct>
```

```
unique(financial_test_data[,2:15])
```

```
## # A tibble: 3,857 × 14
##    INVESTMENT_TOTAL ACCCURRENTBALANCE INF_MARITAL_STATUS INF_GENDER
INSTALL_SIZE
##               <dbl>             <dbl> <fct>              <fct>
<dbl>
##  1           641740              1038 M                  M
0
##  2           532125              4310 M                  M
0
##  3           632625              4310 M                  M
0
##  4          1967250              5114 M                  M
0
##  5          1636875              2787 M                  M
0
##  6          2185500              2787 M                  M
0
##  7          1091250              2787 M                  M
0
##  8           545625             18588 M                  M
0
##  9          1091250              2787 M                  M
7961
## 10           660380                 0 M                  M
7960
## # ℹ 3,847 more rows
## # ℹ 9 more variables: DUE_PAYMENT <dbl>, COMPENSATION_CHARGED <dbl>,
## #   CLIENT_TYPE <dbl>, QUALITY_OF_LOAN <dbl>, REPAY_MODE <fct>,
## #   INVESTMENT_BIN <fct>, DUE_PAYMENT_BIN <fct>, INSTALL_SIZE_BIN <fct>,
## #   BALANCE_BIN <fct>
```

```r
# predictors column numbers - 4,5,8,9, 11, 12, 13, 14, 15
# 10 target

selected_data <- financial_train_data[, c(10, 4, 5, 8, 9, 11, 12, 13, 14,
15)]
library(dplyr)
pca_train_data <- selected_data %>%
  mutate(
    INF_MARITAL_STATUS = recode(INF_MARITAL_STATUS, "M" = 1, "U" = 2, "O" =
3),
    INF_GENDER = recode(INF_GENDER, "F" = 1, "M" = 2, "O" = 3),
    REPAY_MODE = recode(REPAY_MODE, "N" = 0, "I" = 1),
    INVESTMENT_BIN = recode(INVESTMENT_BIN, "Large" = 3, "Medium" = 2,
"Small" = 1),
    DUE_PAYMENT_BIN = recode(DUE_PAYMENT_BIN, "Low Due" = 0, "High Due" = 1),
    INSTALL_SIZE_BIN = recode(INSTALL_SIZE_BIN, "Small" = 1, "Large" = 2),
    BALANCE_BIN = recode(BALANCE_BIN, "High" = 3, "Moderate" = 2, "Low" = 1)
  )


x <- pca_train_data[, -1]  # Exclude the target variable

x_scaled <- scale(x)  # Scaling the features

# Perform PCA
pca <- prcomp(x_scaled, center = TRUE, scale. = TRUE)

# View the proportion of variance explained by each principal component
summary(pca)

## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5    PC6     PC7
## Standard deviation     1.3385 1.2364 1.0050 0.9859 0.9661 0.9256 0.90357
## Proportion of Variance 0.1991 0.1699 0.1122 0.1080 0.1037 0.0952 0.09072
## Cumulative Proportion  0.1991 0.3689 0.4812 0.5892 0.6929 0.7881 0.87877
##                            PC8    PC9
## Standard deviation     0.83586 0.6264
## Proportion of Variance 0.07763 0.0436
## Cumulative Proportion  0.95640 1.0000

# 87% variance
selected_components <- pca$x[, 1:7]

final_data <- cbind(selected_components, pca_train_data$QUALITY_OF_LOAN)

final_data = as.data.frame(final_data)

colnames(final_data)[ncol(final_data)] <- "QUALITY_OF_LOAN"
```

```r
#  logistic regression model
log_model <- glm(QUALITY_OF_LOAN ~ ., data = final_data, family = binomial)

test_data <- financial_test_data[, c(4, 5, 8, 9, 11, 12, 13, 14, 15)]
library(dplyr)
test_data <- test_data %>%
  mutate(
    INF_MARITAL_STATUS = recode(INF_MARITAL_STATUS, "M" = 1, "U" = 2, "O" =
3),
    INF_GENDER = recode(INF_GENDER, "F" = 1, "M" = 2, "O" = 3),
    REPAY_MODE = recode(REPAY_MODE, "N" = 0, "I" = 1),
    INVESTMENT_BIN = recode(INVESTMENT_BIN, "Large" = 3, "Medium" = 2,
"Small" = 1),
    DUE_PAYMENT_BIN = recode(DUE_PAYMENT_BIN, "Low Due" = 0, "High Due" = 1),
    INSTALL_SIZE_BIN = recode(INSTALL_SIZE_BIN, "Small" = 1, "Large" = 2),
    BALANCE_BIN = recode(BALANCE_BIN, "High" = 3, "Moderate" = 2, "Low" = 1)
  )
test_data = as.data.frame(test_data)
test_target <- financial_test_data$QUALITY_OF_LOAN


test_data_scaled <- scale(test_data)
test_data_pca <- predict(pca, newdata = test_data_scaled)
test_data_pca_7 <- test_data_pca[, 1:7]
test_data_pca_7 = as.data.frame(test_data_pca_7)

predictions_prob <- predict(log_model, newdata = test_data_pca_7, type =
"response")
predictions <- ifelse(predictions_prob > 0.5, 1, 0)
confusion_matrix <- table(Predicted = predictions, Actual = test_target)
print(confusion_matrix)
```

```
##            Actual
## Predicted    0    1
##          1   24 4286
```

```r
correct_predictions <- sum(predictions == test_target)

accuracy <- correct_predictions / length(test_target)

# Print accuracy

print(paste("Accuracy: ", round(accuracy * 100, 2), "%", sep = ""))
```

```
## [1] "Accuracy: 99.44%"
```

```r
# Applying SMOTE
#install.packages("DMwR2")
library(DMwR2)
```

```
## Warning: package 'DMwR2' was built under R version 4.3.3
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#install.packages("smotefamily")
library(smotefamily)
```

```
## Warning: package 'smotefamily' was built under R version 4.3.3
```

```
selected_data <- financial_train_data[, c(10, 4, 5, 8, 9, 11, 12, 13, 14,
15)]

# Step 2: Encode categorical variables
pca_train_data <- selected_data %>%
  mutate(
    INF_MARITAL_STATUS = recode(INF_MARITAL_STATUS, "M" = 1, "U" = 2, "O" =
3),
    INF_GENDER = recode(INF_GENDER, "F" = 1, "M" = 2, "O" = 3),
    REPAY_MODE = recode(REPAY_MODE, "N" = 0, "I" = 1),
    INVESTMENT_BIN = recode(INVESTMENT_BIN, "Large" = 3, "Medium" = 2,
"Small" = 1),
    DUE_PAYMENT_BIN = recode(DUE_PAYMENT_BIN, "Low Due" = 0, "High Due" = 1),
    INSTALL_SIZE_BIN = recode(INSTALL_SIZE_BIN, "Small" = 1, "Large" = 2),
    BALANCE_BIN = recode(BALANCE_BIN, "High" = 3, "Moderate" = 2, "Low" = 1)
  )
x <- pca_train_data[,2:10]   # Exclude target column
y <- pca_train_data[,1]

y <- as.factor(y$QUALITY_OF_LOAN)

smote_result <- SMOTE(x, y, K = 5, dup_size = 2)

smote_x <- smote_result$data[, -ncol(smote_result$data)]  # Remove the last
column (class column)
smote_y <- smote_result$data[, ncol(smote_result$data)]
smote_y <- as.factor(smote_y$class)
smote_x <- smote_x %>%
  mutate(across(everything(), ~ as.numeric(as.character(.))))

pca <- prcomp(smote_x, center = TRUE, scale. = TRUE)
smote_x_pca <- predict(pca, newdata = smote_x)
smote_x_pca_7 <- smote_x_pca[, 1:7]

smote_x_pca_7 <- as.data.frame(smote_x_pca_7)

log_model_smote <- glm(smote_y ~ ., data = smote_x_pca_7, family = binomial)

predictions_prob_smote <- predict(log_model_smote, newdata = smote_x_pca_7,
type = "response")
predictions_smote <- ifelse(predictions_prob_smote > 0.5, 1, 0)
```

```
confusion_matrix_smote <- table(Predicted = predictions_smote, Actual =
smote_y)
print(confusion_matrix_smote)
```

```
##          Actual
## Predicted     0     1
##         0    26    71
##         1 12436 33183
```

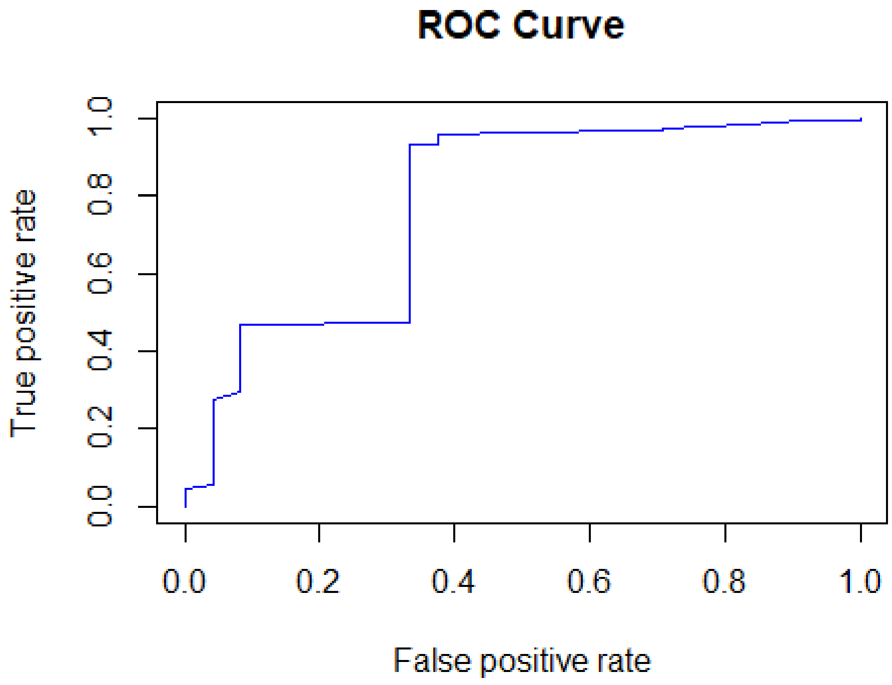
```
accuracy_smote <- sum(predictions_smote == smote_y) / length(smote_y)
print(paste("Accuracy: ", round(accuracy_smote * 100, 2), "%", sep = ""))
```

```
## [1] "Accuracy: 72.64%"
```

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.3.2
```

```
pred <- prediction(predictions_prob, test_target)
perf <- performance(pred, "tpr", "fpr")
plot(perf, col = "blue", main = "ROC Curve")
```



```
auc <- performance(pred, "auc")
print(auc@y.values)
```

```
## [[1]]
## [1] 0.7794078
```