WQD7004 - Group 10 Project - Credit Risk Prediction

Group No.: 10

Group Members

- LINGYU MENG (S2131391)
- HUIJUN LIU (S2142285)
- ZUOGE CHEN (S2125783)
- KELVIIN RAJJ KARUPAYA (S2151665)
- RUIXUE ZHANG (S2142119)

Dataset

- Title: German Credit Risk
- Year: 2020
- Content: This dataset classifies people described by a set of attributes as good or bad credit risks.
- Source: https://www.kaggle.com/datasets/kabure/german-credit-data-with-risk

1 Introduction

In this dataset, each entry represents a person who takes a credit by a bank. Each person is classified as good or bad credit risks according to the set of attributes.

1.1 Project Objective

To manage and avoid the credit risk of users.

1.2 Project Question

-
-

2 Data Pre-processing

2.1 Data Understanding

Import libraries

```
library(dplyr)
library(readr)
library(VIM)
library(missForest)
library(Hmisc)
library(caret)
library(ggplot2)
```

```
library(e1071)
library(klaR)
library(nnet)
library(Metrics)
library(rpart)
library(tidyverse)
```

Read in dataset

```
df=read.csv('german_credit_data.csv')
# Delete the 1st column which is used for indexing.
df=df[,-1]
head(df)
```

```
##   Age    Sex Job Housing Saving.accounts Checking.account Credit.amount
## 1  67   male   2     own            <NA>           little          1169
## 2  22 female   2     own          little         moderate          5951
## 3  49   male   1     own          little             <NA>          2096
## 4  45   male   2    free          little           little          7882
## 5  53   male   2    free          little           little          4870
## 6  35   male   1    free            <NA>             <NA>          9055
##   Duration             Purpose Risk
## 1        6            radio/TV good
## 2       48            radio/TV  bad
## 3       12           education good
## 4       42 furniture/equipment good
## 5       24                 car  bad
```

```
## 6      36          education good
```

See the structure of dataset

```
str(df)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Age             : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ Sex             : chr  "male" "female" "male" "male" ...
##  $ Job             : int  2 2 1 2 2 1 2 3 1 3 ...
##  $ Housing         : chr  "own" "own" "own" "free" ...
##  $ Saving.accounts : chr  NA "little" "little" "little" ...
##  $ Checking.account: chr  "little" "moderate" NA "little" ...
##  $ Credit.amount   : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ Duration        : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ Purpose         : chr  "radio/TV" "radio/TV" "education" "furniture/equipment" ...
##  $ Risk            : chr  "good" "bad" "good" "good" ...
```

2.2 Handle missing data

Check missing values

```
print(paste('Complete obs.:',sum(complete.cases(df))))
```

```
## [1] "Complete obs.: 522"
```

- Distribution of NAs (by column):

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

```
##            Age             Sex             Job         Housing 
##              0               0               0               0 
##  Saving.accounts Checking.account    Credit.amount        Duration 
##            183             394               0               0 
##        Purpose            Risk 
##              0               0
```

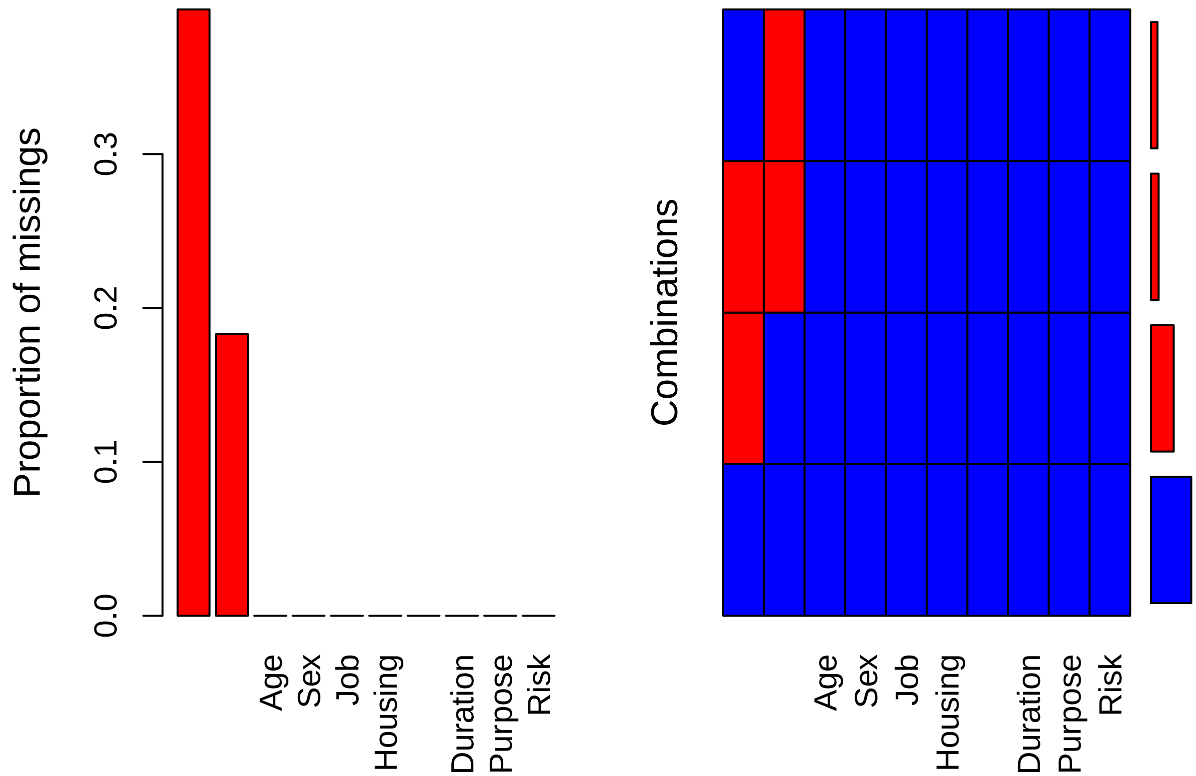
- Check if any missing value is “ ” type

```
colSums(df=="")
```

```
##            Age             Sex             Job         Housing 
##              0               0               0               0 
##  Saving.accounts Checking.account    Credit.amount        Duration 
##             NA              NA               0               0 
##        Purpose            Risk 
##              0               0
```

Visualisation of missing part

```
aggr(df,labels=names(df),col=c('blue','red'),
     numbrs=T,sortVars=T)
```




```
##
## Variables sorted by number of missings:
##          Variable Count
##  Checking.account 0.394
##   Saving.accounts 0.183
##               Age 0.000
##               Sex 0.000
##               Job 0.000
##           Housing 0.000
##     Credit.amount 0.000
##          Duration 0.000
##           Purpose 0.000
##              Risk 0.000
```

As shown above, Saving.accounts & Check.account contain missing values, and red color presents missing part.

Predict and impute NAs

Step 1: Convert chr variables to factor type

```
df1 <- df
df1$Sex <- as.factor(df1$Sex)
df1$Job <- as.ordered(df1$Job)
df1$Housing <- as.factor(df1$Housing)
df1$Saving.accounts <- as.ordered(df1$Saving.accounts)
df1$Checking.account <- as.ordered(df1$Checking.account)
df1$Purpose <- as.factor(df1$Purpose)
df1$Risk <- as.ordered(df1$Risk)
```

Step 2: Impute NAs using missForest

missForest is used to impute missing values particularly in the case of mixed-type data. It can be used to impute **continuous and/or categorical** data including complex interactions and nonlinear relations. It yields an out-of-bag (OOB) imputation error estimate. Moreover, it can be run parallel to save computation time.

```
df.mis <- df1[!complete.cases(df1),]
df.train <- df1[complete.cases(df1),]
set.seed(42)
df.imp <- missForest(xmis = df.mis, xtrue = df.train, maxiter = 10, ntree = 200)
message('Out of Bag error: ', df.imp$OOBerror)
```

```
## Out of Bag error: 00.089277066758907
```

Save imputation result

```
df.nomis <- df1
df.nomis[!complete.cases(df.nomis),] <- df.imp$ximp
```

SavAcct values distribution after imputation:

```
table(df.nomis$Saving.accounts)
```

```
##
##     little   moderate quite rich       rich
##        739        113         92         56
```

CheckAcct values distribution after imputation:

```
table(df.nomis$Checking.account)
```

```
##
##   little moderate     rich
##      360      489      151
```

2.3 Smooth noisy data (Not yet decided)

- Save cleaned dataset

```
write.csv(df.nomis,file = 'german_credit_data_rmna.csv',row.names=F)
```

- Summary dataset

```
summary(df.nomis)
```

```
##      Age           Sex      Job     Housing     Saving.accounts
##  Min.   :19.00   female:310   0: 22   free:108   little    :739
##  1st Qu.:27.00   male  :690   1:200   own :713   moderate  :113
##  Median :33.00                2:630   rent:179   quite rich: 92
##  Mean   :35.55                3:148              rich      : 56
##  3rd Qu.:42.00
##  Max.   :75.00
##
##  Checking.account Credit.amount     Duration                  Purpose
##  little  :360     Min.   :  250   Min.   : 4.0   car                :337
##  moderate:489     1st Qu.: 1366   1st Qu.:12.0   radio/TV           :280
##  rich    :151     Median : 2320   Median :18.0   furniture/equipment:181
##                   Mean   : 3271   Mean   :20.9   business           : 97
##                   3rd Qu.: 3972   3rd Qu.:24.0   education          : 59
##                   Max.   :18424   Max.   :72.0   repairs            : 22
##                                                  (Other)            : 24
##    Risk
##  bad :300
##  good:700
##
##
##
##
##
```

Descriptions of cleaned dataset

- Age: (quantitative, in years)
- Sex: (dichotomous: female, male)
- Job: (ordinal: 0 - unskilled and non-resident, 1 - unskilled and resident, 2 - skilled, 3 - highly skilled)
- Housing: (nominal: own, rent, or free)
- SavAcct: (ordinal: little, moderate, quite rich, rich) - Status of existing saving account.
- CheckAcct: (ordinal: little, moderate, rich) - Status of existing checking account.
- CredAmt: (quantitative, in D-mark) The maximum amount that the bank is committed to lend.
- Duration: (quantitative, in month)
- Purpose: (nominal: car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/others) - Reasons to get a loan.
- Risk: (dichotomous : good, bad)