ECON412_Project 1

LIU, YIPING

LATIFI, ROYA

SUN, YIRAN

Apr 30,2021

I. Introduction

In general, machine learning problems could be divided into two categories, supervised learning and unsupervised learning. Unsupervised learning is used to solve clustering problems, density estimation, and dimensionality reduction issues. Density estimation is the construction of an estimate of an unobservable underlying probability density function, which consists of both parametric and non-parametric methods. In machine learning studies, we are often interested in a predictive modeling problem where we would like to predict a class label for a given observation. A probabilistic classifier can predict, given an observation of input, the conditional probability of a class label, and Bayes Theorem provides a principled way for calculating this conditional probability.

Bayes Theorem:

$$\Pr(L|Features) = \frac{\Pr(Features|L)\Pr(L)}{\Pr(Features)}$$

Bayes theorem provides a principled way of calculating a conditional probability without the joint probability. Under Bayes rule, sequential Bayesian learning and Naive Bayes are two popular methods of solving density estimation problems. But in practice, even with large datasets, it may be hard to find other records that exactly match the record, in terms of predictor values, with Bayes Theorem. In this situation, the Naive Bayes classification model can be used.

Naive Bayes theory is a simple supervised machine learning and data-driven algorithm that uses the Bayes theorem and it makes only simple assumptions about the data. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

A. Task Description

Our data is from Thera Bank, which has a growing customer base. The dominant part of these clients are depositors and, borrowers only account for a very small portion of the clients. Thus, Thera Bank is keen on extending its customer base quickly to acquire more loan business and all the while, procure more through the interest on loans.

Thera Bank management wants to have a model that will assist them with recognizing the potential clients who have a higher likelihood of buying the credit. Specifically, the bank needs to investigate methods of changing its liability clients over to personal loan clients.

For this project, our objective is to implement a naive Bayesian learning algorithm to the Thera Bank data set and to construct a model that will assist the bank with distinguishing the potential clients who have a higher likelihood of getting the loan.

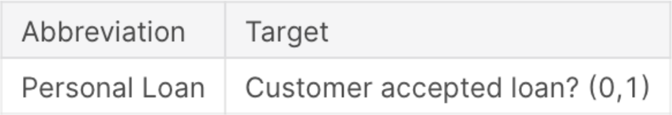
II. Data

We utilized the There Bank dataset, taken from the UCI Machine Learning Repository, which has 5000 observations with fourteen variables divided into four different measurement categories. The data incorporates customer segment data (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan).

A. Attribute Information:

The binary category has five variables, including the target variable personal loan, also securities account, CD account, online banking, and credit card. The interval category contains five variables: age, experience, income, CC avg, and mortgage. The ordinal category includes the variables family and education. The last category is nominal with ID and Zip code.

Dependent Variable:

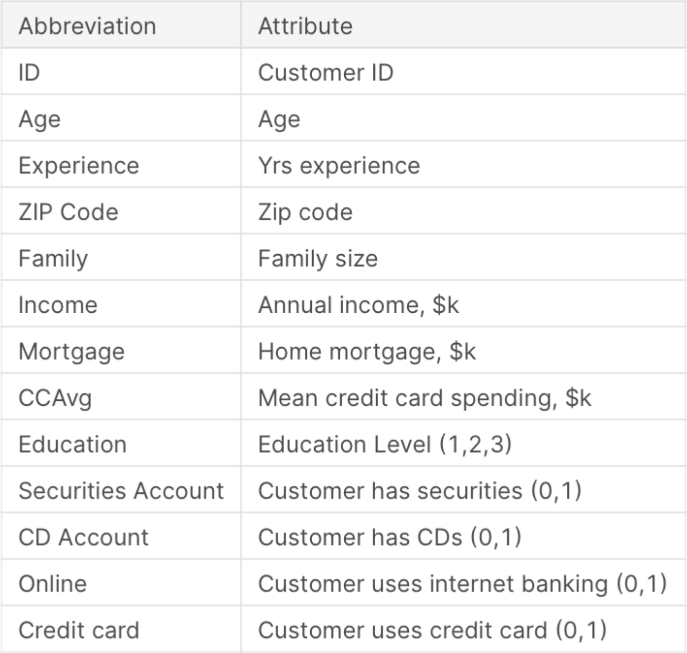


Abbreviation	Target
Personal Loan	Customer accepted loan? (0,1)

The dependent variable in this dataset is the personal loan. It is a categorical variable with values 0 and 1.

Since we use naive Bayes in data analysis, we will transform the numerical variables into categorical ones in the upcoming steps.

Independent Variables:



Abbreviation	Attribute
ID	Customer ID
Age	Age
Experience	Yrs experience
ZIP Code	Zip code
Family	Family size
Income	Annual income, $k
Mortgage	Home mortgage, $k
CCAvg	Mean credit card spending, $k
Education	Education Level (1,2,3)
Securities Account	Customer has securities (0,1)
CD Account	Customer has CDs (0,1)
Online	Customer uses internet banking (0,1)
Credit card	Customer uses credit card (0,1)

Categorical:

- Family : Family size of the customer
- Education : Education Level.

```
1. Undergrad

2. Graduate

3. Advanced/Professional
```

- Securities Account : Does the customer have a securities account with the bank?
- CD Account : Does the customer have a certificate of deposit (CD) account with the bank?
- Online : Does the customer use internet banking facilities?
- Credit card : Does the customer use a credit card issued by Thera Bank?

Numerical:

- Age : Customer's age in completed years
- Experience : #years of professional experience
- Income : Annual income of the customer
- CCAvg : Avg. spending on credit cards per month
- Mortgage : Value of house mortgage if any.

- ID : Customer ID
- ZIP Code : Home Address ZIP code.

B. Data Input

```
data <- read_excel("Bank_Personal_Loan_Modelling.xlsx",sheet ="Data", col_names = TR
UE)
head(data)
```

ID	...	Experience	Inco...	ZIP Code	Fam...	CC...	Educati...	Mortg... ▸
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	25	1	49	91107	4	1.6	1	0
2	45	19	34	90089	3	1.5	1	0
3	39	15	11	94720	1	1.0	1	0
4	35	9	100	94112	1	2.7	2	0

ID	...	Experience	Inco...	ZIP Code	Fam...	CC...	Educati...	Mortg... ▸
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
5	35	8	45	91330	4	1.0	2	0
6	37	13	29	92121	4	0.4	2	155

6 rows | 1-9 of 14 columns

There are 5000 rows and 14 columns in the dataset.

C. Simple Description

```
# check NA
a <- names(data)
b <- data.frame()
null_df <- data.frame()

for (i in a){
  c <- sum(is.na(data[,i]))
  b <- data.frame("Feature name"=i,"num of NAs"=c)
  null_df <-rbind(null_df,b)}

print(null_df)
```

```
##          Feature.name num.of.NAs
## 1                  ID          0
## 2                 Age          0
## 3          Experience          0
## 4              Income          0
## 5            ZIP Code          0
## 6              Family          0
## 7               CCAvg          0
## 8           Education          0
## 9            Mortgage          0
## 10      Personal Loan          0
## 11 Securities Account          0
## 12         CD Account          0
## 13             Online          0
## 14         CreditCard          0
```

In our data, we haven't observed any missing values.

```
#simple describe
describe(data)
```

	v...	n	mean	sd	median
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
ID	1	5000	2500.500000	1443.5200033	2500.5
Age	2	5000	45.338400	11.4631656	45.0
Experience	3	5000	20.104600	11.4679537	20.0
Income	4	5000	73.774200	46.0337293	64.0
ZIP Code	5	5000	93152.503000	2121.8521973	93437.0
Family	6	5000	2.396400	1.1476630	2.0
CCAvg	7	5000	1.937913	1.7476662	1.5
Education	8	5000	1.881000	0.8398691	2.0
Mortgage	9	5000	56.498800	101.7138021	0.0
Personal Loan	10	5000	0.096000	0.2946207	0.0

1-10 of 14 rows | 1-7 of 14 columns   Previous **1** 2 Next

```
#Delete rows with negative Experience
data <- data[data$Experience >= 0,]
describe(data)
```

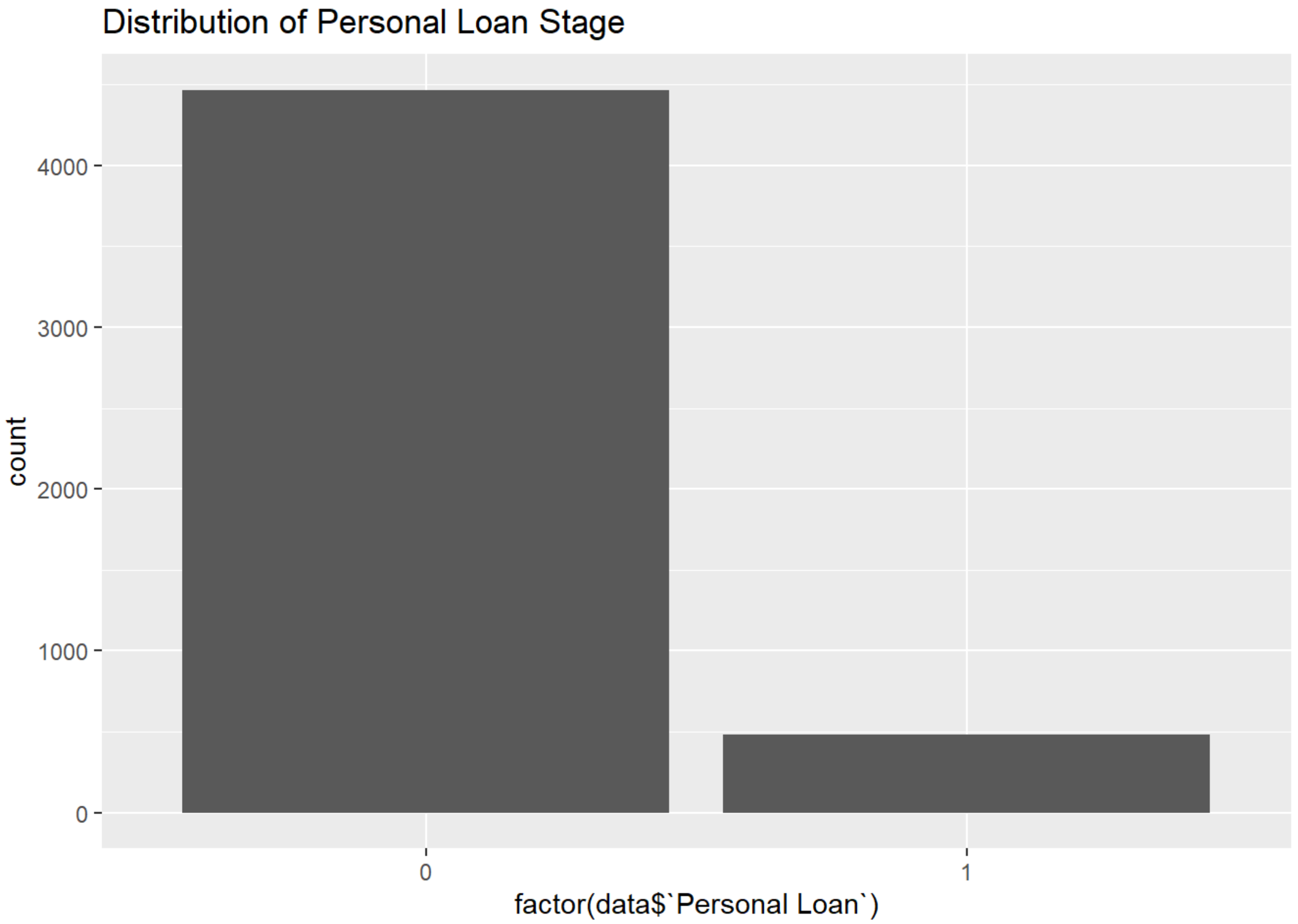
	v...	n	mean	sd	median
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
ID	1	4948	2.501269e+03	1443.2776763	2497.5
Age	2	4948	4.555719e+01	11.3207353	46.0
Experience	3	4948	2.033104e+01	11.3119730	20.0
Income	4	4948	7.381447e+01	46.1125956	64.0
ZIP Code	5	4948	9.315157e+04	2126.6690168	93437.0
Family	6	4948	2.391471e+00	1.1484444	2.0
CCAvg	7	4948	1.935900e+00	1.7476998	1.5
Education	8	4948	1.878941e+00	0.8397452	2.0
Mortgage	9	4948	5.663440e+01	101.8288850	0.0
Personal Loan	10	4948	9.700889e-02	0.2959998	0.0

1-10 of 14 rows | 1-6 of 14 columns   Previous **1** 2 Next

- The mean value of Age is 45. The majority of the customers are falling in the 51-60 age bucket followed by 41-50 and then 31-40.
- Experience feature has a minimum value of -3, which is not valid as there is not such a thing as a negative experience. Thus, it needs to be corrected by dropping those values.
- The median value for income is 64 while the maximum and minimum values of income are 224 and 8 respectively. We have observed that the standard deviation for income is very high.
- The number of people that have CD accounts is very low.
- Almost 60% of users use online banking.
- Almost 30% of the users use credit cards.
- Family Members, Education, Personal Loan, securities Account, CD Account, Online, Credit Cards seem to be factor variables.
- ID and Zip Code seem to be irrelevant for analysis as they won’t play a role in classifying the customer group.

D. Data analyis and insights

```
# distribution of dependent variable
ggplot(data, aes(x = factor(data$`Personal Loan`))) +
  geom_bar()+
  ggtitle("Distribution of Personal Loan Stage")
```



```
accept <- sum(data$`Personal Loan`==1)

cat("Number of customers who bought personal loan:",accept,
    "(",accept / length(data$`Personal Loan`) * 100 ,"%)")
```

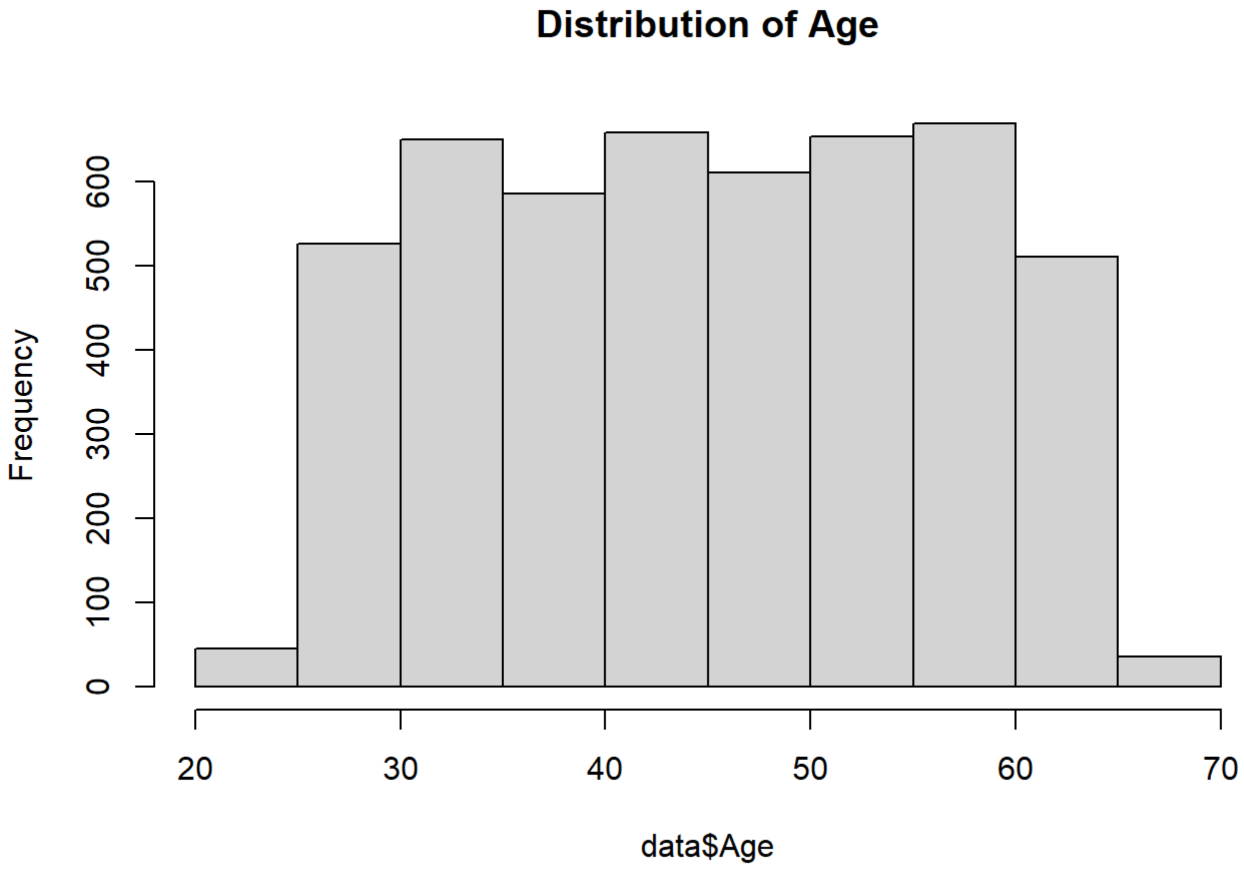
```
## Number of customers who bought personal loan: 480 ( 9.700889 %)
```

```
cat("\nNumber of customers who didn't buy personal loan:",length(data$`Personal Loan
`)-accept,
    "(",(1-accept / length(data$`Personal Loan`)) * 100 ,"%)")
```

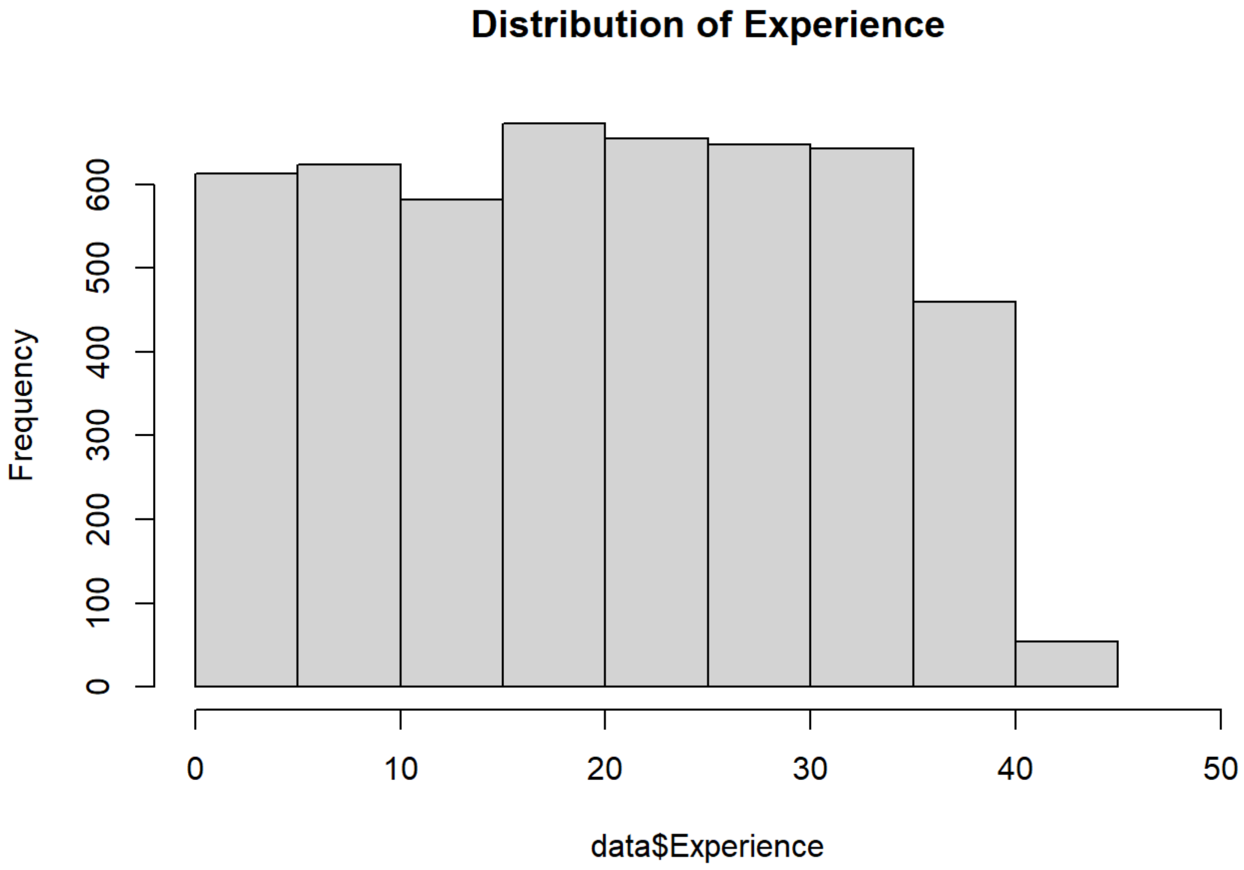
```
##
## Number of customers who didn't buy personal loan: 4468 ( 90.29911 %)
```

Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.

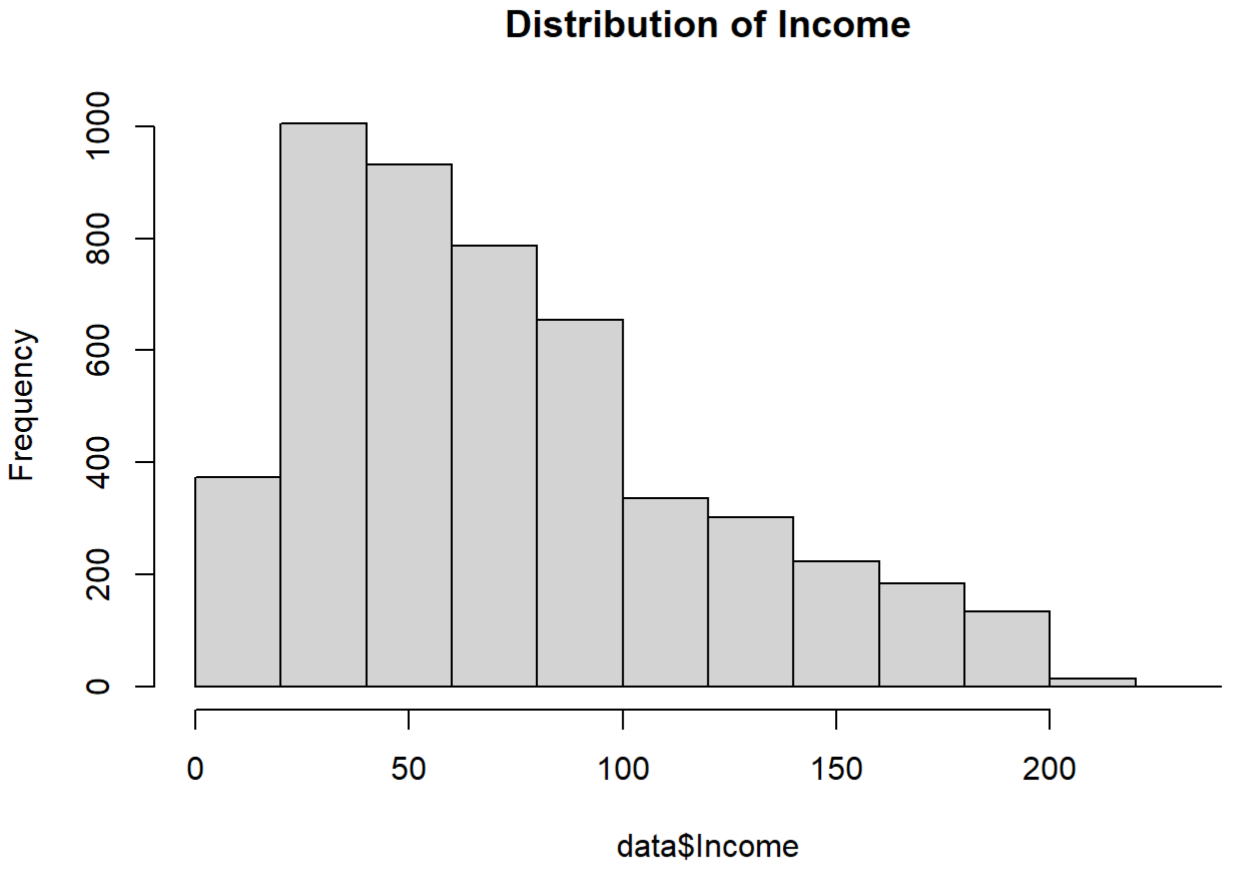
```
# distribution of independent variables
hist(data$Age,main="Distribution of Age")
```



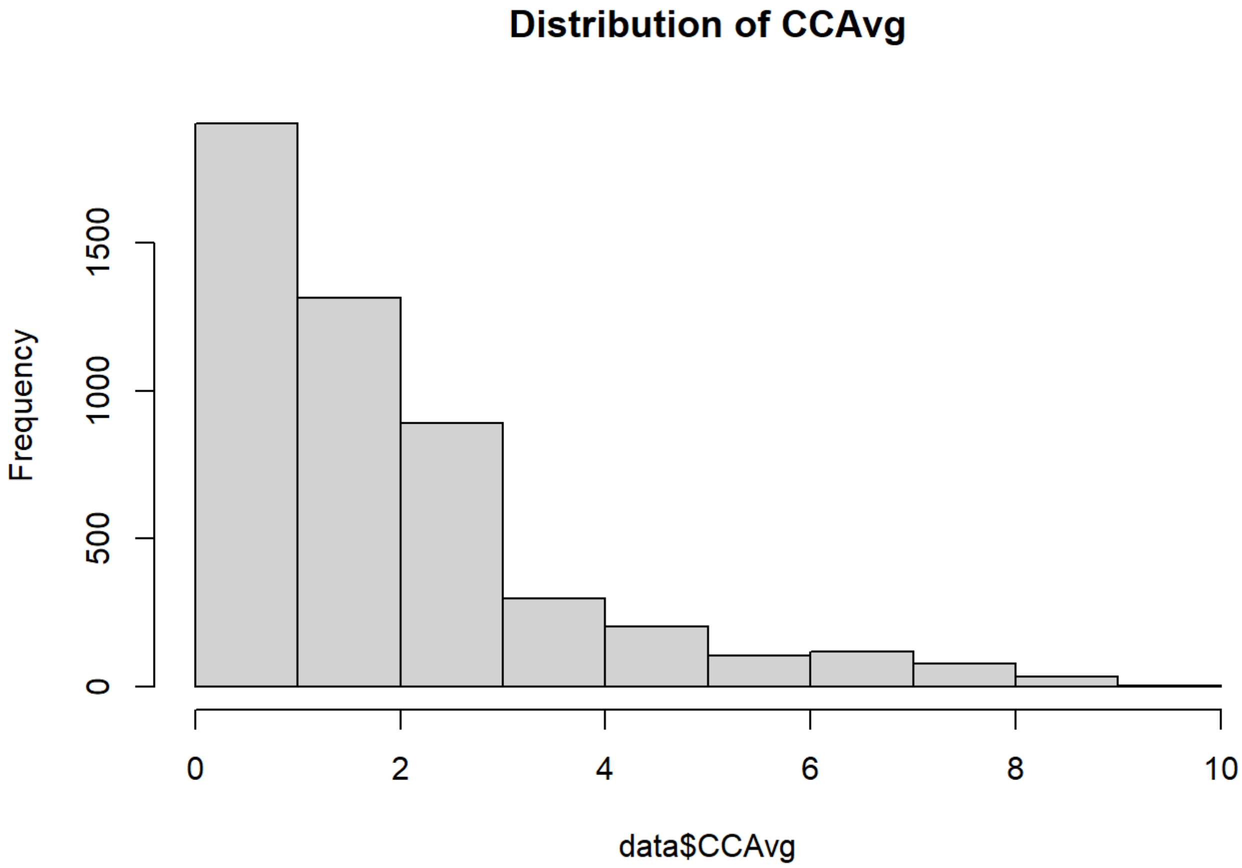
```
hist(data$Experience,xlim=c(0,50),main="Distribution of Experience")
```



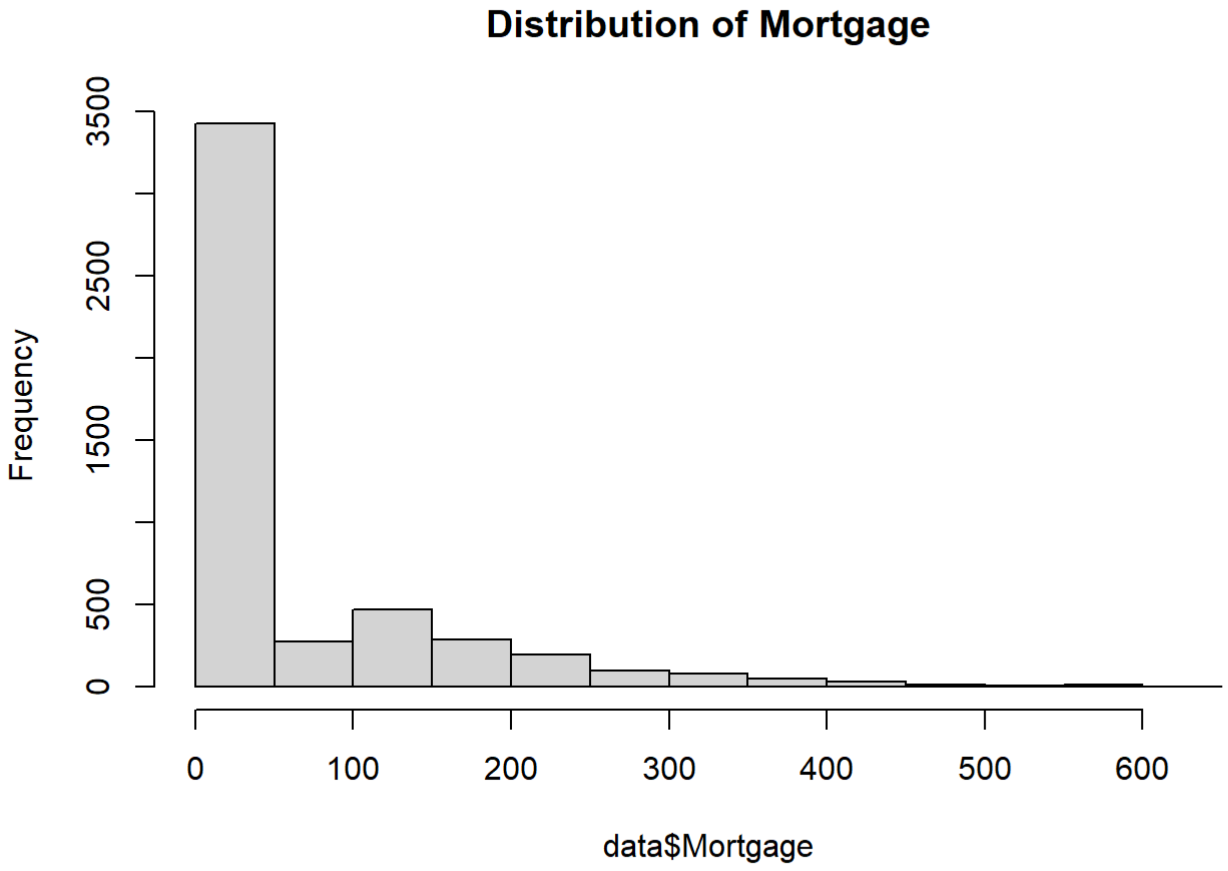
```
hist(data$Income,main="Distribution of Income")
```



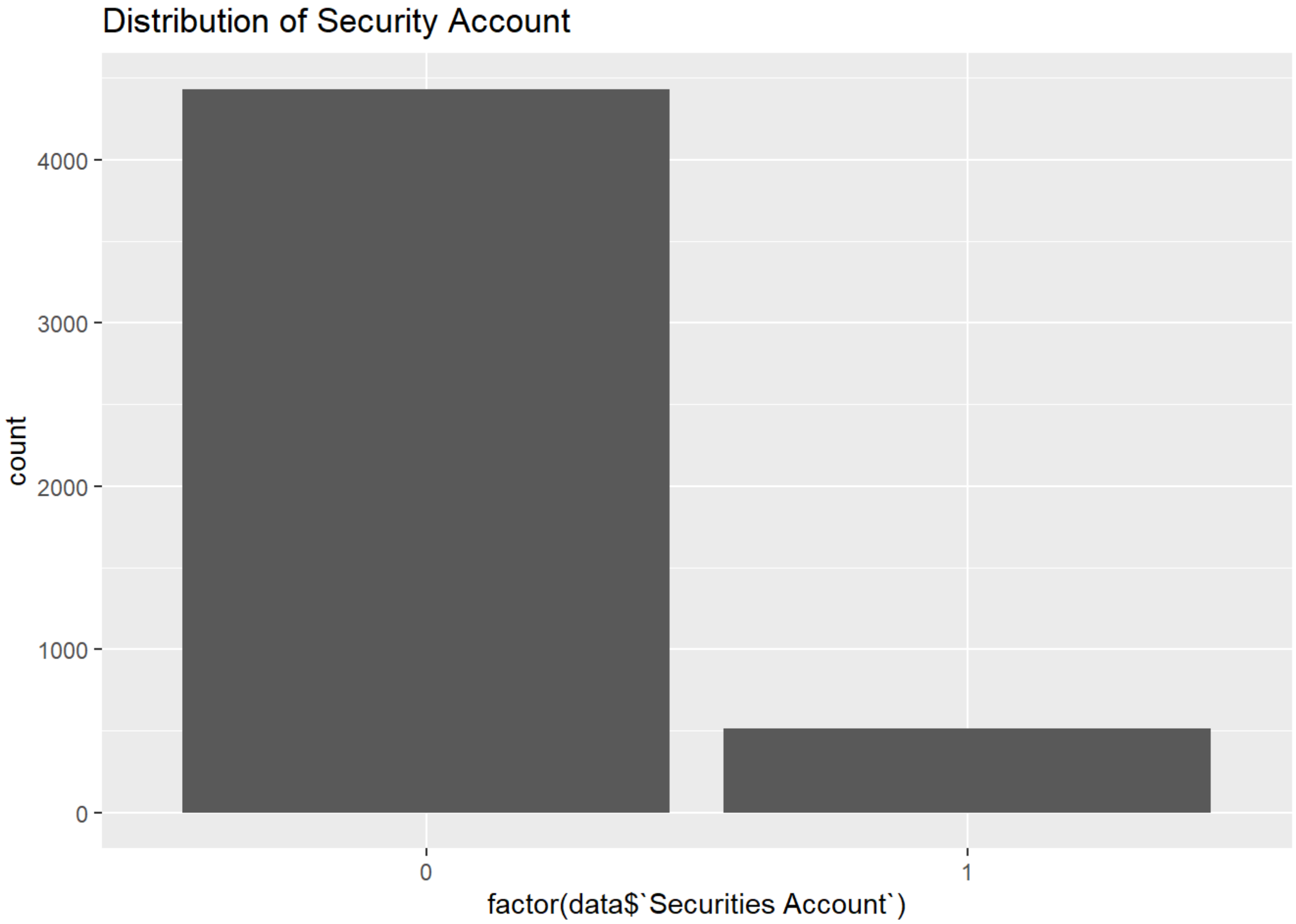
```
hist(data$CCAvg,main="Distribution of CCAvg")
```



```
hist(data$Mortgage,main="Distribution of Mortgage")
```



```
ggplot(data, aes(x = factor(data$`Securities Account`)) +
  geom_bar()+
  ggtitle("Distribution of Security Account")
```

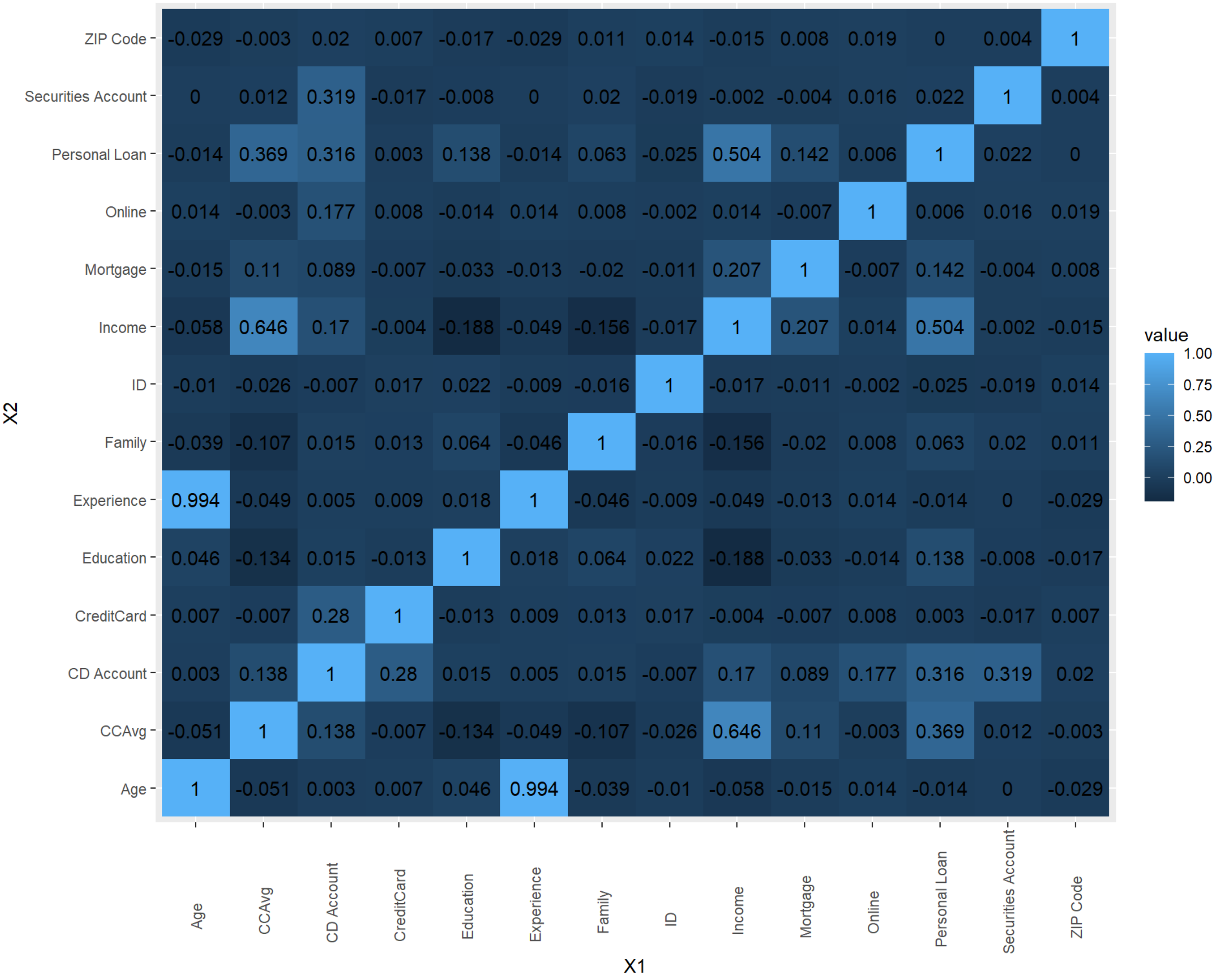


- Age: Senior citizens are relatively low when compared to other age buckets. The density graph shows that the age variable has an almost normal distribution.
- Experience: Most customers in the Thera Bank dataset have 21-30 years of experience. The density graph shows that the experience variable has an almost normal distribution.
- Income: Most customers are earning an average income of fewer than 50k dollars per year. Customers earning more than 100k dollars are relatively low when compared to other buckets. The density graph shows that the income variable is rightly skewed.
- CCAvg: The number of customers that have average spending on credit cards per month less than $1000 is relatively low. The density graph shows that the average spending on credit cards per month variable is rightly skewed.
- Mortgage: The majority of the customers are not having any mortgage. The density graph shows that the mortgage variable is rightly skewed.

E. Correlation Matrix Analysis

```
# Heatmap
corr <- melt(cor(as.matrix(data)))

par(mar=c(0,0,0,0))
ggplot(corr, aes(x=X1,y=X2,fill=value))+
  geom_tile()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=0.25))+
  geom_text(aes(fill = round(corr$value,2),label = round(corr$value,3)),size=4)
```



	Age	CCAvg	CD Account	CreditCard	Education	Experience	Family	ID	Income	Mortgage	Online	Personal Loan	Securities Account	ZIP Code
ZIP Code	-0.029	-0.003	0.02	0.007	-0.017	-0.029	0.011	0.014	-0.015	0.008	0.019	0	0.004	1
Securities Account	0	0.012	0.319	-0.017	-0.008	0	0.02	-0.019	-0.002	-0.004	0.016	0.022	1	0.004
Personal Loan	-0.014	0.369	0.316	0.003	0.138	-0.014	0.063	-0.025	0.504	0.142	0.006	1	0.022	0
Online	0.014	-0.003	0.177	0.008	-0.014	0.014	0.008	-0.002	0.014	-0.007	1	0.006	0.016	0.019
Mortgage	-0.015	0.11	0.089	-0.007	-0.033	-0.013	-0.02	-0.011	0.207	1	-0.007	0.142	-0.004	0.008
Income	-0.058	0.646	0.17	-0.004	-0.188	-0.049	-0.156	-0.017	1	0.207	0.014	0.504	-0.002	-0.015
ID	-0.01	-0.026	-0.007	0.017	0.022	-0.009	-0.016	1	-0.017	-0.011	-0.002	-0.025	-0.019	0.014
Family	-0.039	-0.107	0.015	0.013	0.064	-0.046	1	-0.016	-0.156	-0.02	0.008	0.063	0.02	0.011
Experience	0.994	-0.049	0.005	0.009	0.018	1	-0.046	-0.009	-0.049	-0.013	0.014	-0.014	0	-0.029
Education	0.046	-0.134	0.015	-0.013	1	0.018	0.064	0.022	-0.188	-0.033	-0.014	0.138	-0.008	-0.017
CreditCard	0.007	-0.007	0.28	1	-0.013	0.009	0.013	0.017	-0.004	-0.007	0.008	0.003	-0.017	0.007
CD Account	0.003	0.138	1	0.28	0.015	0.005	0.015	-0.007	0.17	0.089	0.177	0.316	0.319	0.02
CCAvg	-0.051	1	0.138	-0.007	-0.134	-0.049	-0.107	-0.026	0.646	0.11	-0.003	0.369	0.012	-0.003
Age	1	-0.051	0.003	0.007	0.046	0.994	-0.039	-0.01	-0.058	-0.015	0.014	-0.014	0	-0.029

- Usage of credit card is positively correlated to income of a person.
- The number of customers with higher education are buying Personal Loan compared to other groups.
- Customers who operate online are more likely to take loans compared to non online users.
- Family with size more than 2 are more interested in personal loans.
- Customers with no credit card are more interested to buy personal loans.
- Customers with no security accounts are more interested in buying personal loans.
- There is a higher correlation in Age and Experience features so we can drop one of them.
- Correlation coefficient of ID and target variable Personal Loan is negative and close to zero so we can drop the variable.
- Correlation coefficients of Age and Experience are negative and close to zero so we can drop these variables as well.

- Correlation coefficient of Zip code variable is also close to zero so we can drop this variable

F. Data Processing

Variable Selection

Based on data correlation and classification intuition, we have chosen Age, Experience, Income, Family, CCAvg, Education, Mortgage, Securities Account, and CD Account as features for classification and use the features to determine Personal Loan category which will help the bank with distinguishing the potential clients who have a higher likelihood of getting the loan.

Variable Transformation

In our dataset, all of the variables are of numeric data type. Naive Bayes requires categorical variables. Thus, numerical variables must be binned and converted to categorical.

We have converted the below variables into factors:

We use the "factor" function to transform family members, securities account, CD Account, and education into categorical variables. As for Age, Experience, Mortgage, Income, CCAvg, we split them into several groups based on data distribution and then converted them into categorical variables.

```
# Transfer into Categorical variable
data$`Personal Loan` <- factor(data$`Personal Loan`)
data$Family <- factor(data$Family)
data$`Securities Account`<- factor(data$`Securities Account`)
data$`CD Account`<- factor(data$`CD Account`)
data$Education <- factor(data$Education)
```

```
# Change numerical variable into Categorical ones
data$AgeG <- factor(cut(data$Age,breaks = c(22,30,40,50,60,70)
             ,labels=c("<=30","31-40","41-50","51-60","more than 60")))

data$ExperienceG <- factor(cut(data$Experience,breaks = c(0,10,20,30,50)
             ,labels=c("<=10y","11-20","21-30","more than 30")))

data$MortgageG <- factor((data$Mortgage>0)*1)

data$IncomeG <- factor(cut(data$Income,breaks = c(0,50,100,150,300)
                    ,labels = c("0-50$","51-100$","101-150$","151-$")))

data$CCAvgG <- factor(cut(data$CCAvg,breaks = c(0,1,4,6,20)
                   ,labels = c("1","2-4","5-6", "more than 6")))

head(data)
```

ID	...	Experience	Inco...	ZIP Code	Fam...	CC...	Educati...	Mortg... ▸
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>	<dbl>	<fct>	<dbl>
1	25	1	49	91107	4	1.6	1	0
2	45	19	34	90089	3	1.5	1	0
3	39	15	11	94720	1	1.0	1	0
4	35	9	100	94112	1	2.7	2	0
5	35	8	45	91330	4	1.0	2	0
6	37	13	29	92121	4	0.4	2	155

6 rows | 1-9 of 19 columns

Dataset Split

We split 60% of the data for training and the left is used for validation.

```
# Create training and validation sets.
selected.var <- c(6,8,10,12,15,16,17,18,19)

set.seed(412)
train.index <- sample(c(1:dim(data)[1]), dim(data)[1]*0.6)

train.data <- data[train.index, selected.var]
valid.data <- data[-train.index, selected.var]
```

```
# check if the dependent variable is distributed evenly.
accept1 <- sum(train.data$`Personal Loan`==1)
accept2 <- sum(valid.data$`Personal Loan`==1)

cat("Number of customers who bought personal loan in Training set:",accept1,
    "(",accept1 / length(train.data$`Personal Loan`) * 100 ,"%)")
```

```
## Number of customers who bought personal loan in Training set: 292 ( 9.838275 %)
```

```
cat("\nNumber of customers who bought personal loan in Validation set:",accept2,
    "(",accept2 / length(valid.data$`Personal Loan`) * 100 ,"%)")
```

```
##
## Number of customers who bought personal loan in Validation set: 188 ( 9.494949 %)
```

III. Model Fitting

A. Fitting Naive Bayes

We have used a Naive Bayes classifier to conduct the classification and estimate the personal loan category of potential clients (unknown sample) with attributed features. The Naive Bayes function will help us compute a categorical class variable's prior and posterior probabilities using the Bayes rule and predict the class of a new sample from its features. It finds the probability of a given set of features for all possible values of the class variable Y (Potential Client Class) and picks up the output with maximum probability. With this algorithm, we can determine the potential client class of a sample by its highest posterior probability.

Naive Bayes Method

$$\Pr(X|C=i) = \prod_{n=1}^{t} \Pr(X_n|C=i)$$

$$y = argmax_y P(y) \prod_{i=1}^{n} \Pr(X_i|y)$$

The above equation of Naive Bayes helps us to obtain the class of potential clients, given the predictors/features.

```
model <- naiveBayes(`Personal Loan` ~ ., data = train.data)
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90161725 0.09838275
##
## Conditional probabilities:
##    Family
## Y           1         2         3         4
##   0 0.3109118 0.2578475 0.2002990 0.2309417
##   1 0.2123288 0.2534247 0.2739726 0.2602740
##
##    Education
## Y           1         2         3
##   0 0.4409567 0.2731689 0.2858744
##   1 0.1952055 0.3664384 0.4383562
##
##    CD Account
## Y            0          1
##   0 0.95889387 0.04110613
##   1 0.75684932 0.24315068
##
##    AgeG
## Y        <=30     31-40     41-50     51-60 more than 60
##   0 0.1136024 0.2443946 0.2608371 0.2690583    0.1121076
##   1 0.1506849 0.2465753 0.2431507 0.2328767    0.1267123
##
##    ExperienceG
## Y       <=10y     11-20     21-30 more than 30
##   0 0.2351382 0.2548277 0.2711094    0.2389246
##   1 0.2951389 0.2222222 0.2430556    0.2395833
##
##    MortgageG
## Y           0         1
##   0 0.6875934 0.3124066
##   1 0.6506849 0.3493151
##
##    IncomeG
## Y        0-50$    51-100$   101-150$      151-$
##   0 0.42488789 0.40358744 0.12107623 0.05044843
##   1 0.00000000 0.08219178 0.46232877 0.45547945
##
##    CCAvgG
## Y            1        2-4        5-6 more than 6
##   0 0.39281620 0.53381735 0.03782958  0.03553687
##   1 0.09965636 0.46048110 0.27491409  0.16494845
```

```
#we could check one by one
prop.table(table(train.data$`Personal Loan`, train.data$Family), margin = 2)
```

```
## 
##              1          2          3          4
##   0 0.93064877 0.90314136 0.87012987 0.89048991
##   1 0.06935123 0.09685864 0.12987013 0.10951009
```

```
prop.table(table(train.data$`Personal Loan`, train.data$Education), margin = 2)
```

```
## 
##              1          2          3
##   0 0.95392078 0.87231504 0.85666293
##   1 0.04607922 0.12768496 0.14333707
```

```
prop.table(table(train.data$`Personal Loan`, train.data$`CD Account`), margin = 2)
```

```
## 
##              0          1
##   0 0.92070327 0.60773481
##   1 0.07929673 0.39226519
```

```
prop.table(table(train.data$`Personal Loan`, train.data$AgeG), margin = 2)
```

```
## 
##           <=30      31-40      41-50      51-60 more than 60
##   0 0.87356322 0.90082645 0.90767230 0.91370558   0.89020772
##   1 0.12643678 0.09917355 0.09232770 0.08629442   0.10979228
```

```
prop.table(table(train.data$`Personal Loan`, train.data$ExperienceG), margin = 2)
```

```
## 
##          <=10y      11-20      21-30 more than 30
##   0 0.87960340 0.91316147 0.91094148   0.90142857
##   1 0.12039660 0.08683853 0.08905852   0.09857143
```

```
prop.table(table(train.data$`Personal Loan`, train.data$MortgageG), margin = 2)
```

```
## 
##              0          1
##   0 0.90640394 0.89125800
##   1 0.09359606 0.10874200
```

```
prop.table(table(train.data$`Personal Loan`, train.data$IncomeG), margin = 2)
```

```
##
##          0-50$    51-100$   101-150$      151-$
##   0 1.00000000 0.97826087 0.70588235 0.50373134
##   1 0.00000000 0.02173913 0.29411765 0.49626866
```

```
prop.table(table(train.data$`Personal Loan`, train.data$CCAvgG), margin = 2)
```

```
##
##               1        2-4        5-6 more than 6
##   0 0.97256386 0.91247551 0.55307263  0.65957447
##   1 0.02743614 0.08752449 0.44692737  0.34042553
```

The columns give the posterior probabilities of the labels.

- As for the results of Naive Bayes, we can see that for categorical variables clients with higher education levels, or without a CD account are more likely to accept loan services. Families with more members are more likely to accept loan services overall but the number of family members won't strictly affect the acceptance.
- And for initial numerical variables which we transformed by assigning groups, clients who are younger than 30 are more likely to be involved in loan services, as well as those with less working experience(less than 10 years). And higher salaries and higher spending on credit cards also indicate that they are potential loan clients.
- The mortgage conditions do not have a significant impact on loan needs.

B. Predict Probabilities

```
# Predict Probabilities
pred.prob <- predict(model, newdata = valid.data, type = "raw")

## predict class membership
pred.class <- predict(model, newdata = valid.data)

df <- data.frame(actual = valid.data$`Personal Loan`, predicted = pred.class, pred.p
rob)
df
```

actual	predicted	X0	X1
<fct>	<fct>	<dbl>	<dbl>
0	0	0.999906839	9.316113e-05

actual	predicted	X0	X1
<fct>	<fct>	<dbl>	<dbl>
0	0	0.999923522	7.647820e-05
0	0	0.985563389	1.443661e-02
0	0	0.988571049	1.142895e-02
1	1	0.180848450	8.191516e-01
0	0	0.747308552	2.526914e-01
0	0	0.915760673	8.423933e-02
1	1	0.197503149	8.024969e-01
0	0	0.999965407	3.459257e-05
0	0	0.999924558	7.544225e-05

1-10 of 1,980 rows Previous **1** 2 3 4 5 6 ... 198 Next

From the results, we could get the probability that the customers may accept the personal loan and the predicting classification of the client from the validation part of the dataset.

C. Evaluating Performace

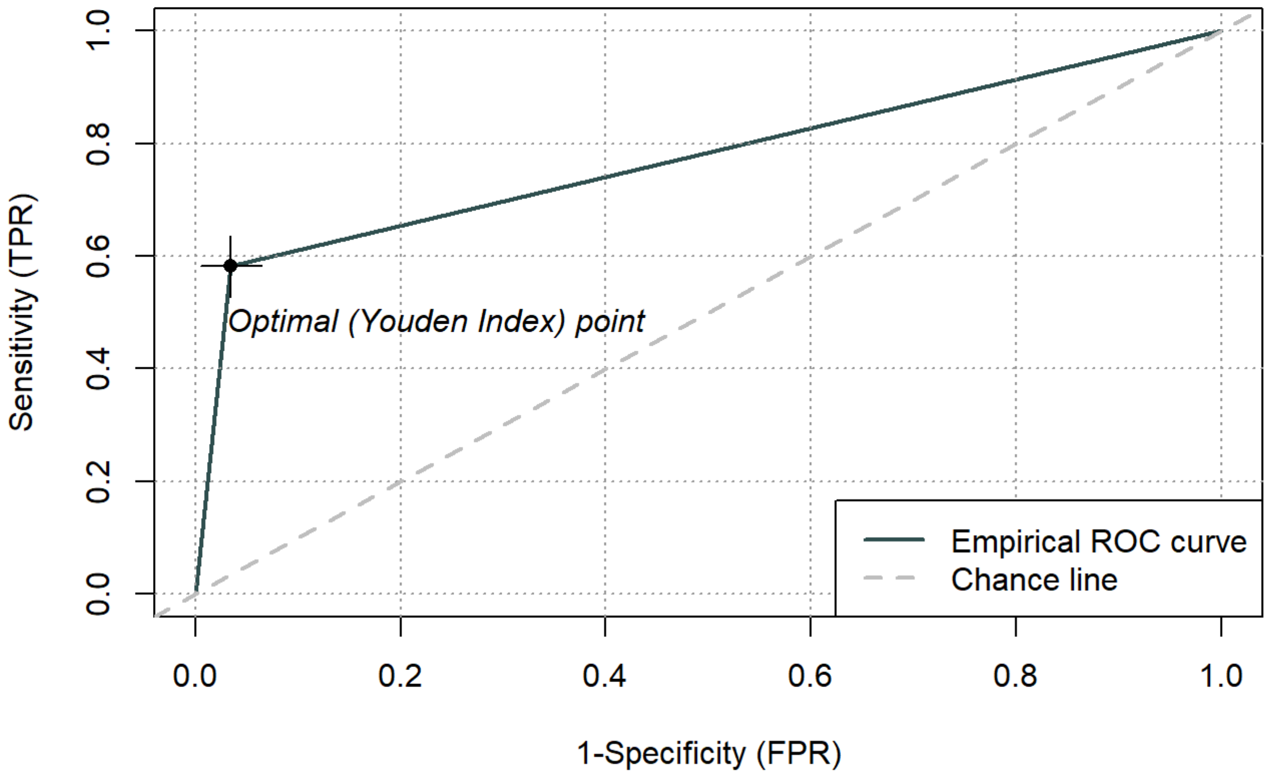
```
#Confusion Matrix for Training
fit.class <- predict(model, newdata = train.data)
confusionMatrix(as.factor(fit.class), as.factor(train.data$`Personal Loan`))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    0    1
##          0 2583  122
##          1   93  170
##                                           
##                Accuracy : 0.9276          
##                  95% CI : (0.9176, 0.9366)
##     No Information Rate : 0.9016          
##     P-Value [Acc > NIR] : 4.466e-07       
##                                           
##                   Kappa : 0.5728          
##                                           
##  Mcnemar's Test P-Value : 0.05619         
##                                           
##             Sensitivity : 0.9652          
##             Specificity : 0.5822          
##          Pos Pred Value : 0.9549          
##          Neg Pred Value : 0.6464          
##              Prevalence : 0.9016          
##          Detection Rate : 0.8703          
##    Detection Prevalence : 0.9114          
##       Balanced Accuracy : 0.7737          
##                                           
##        'Positive' Class : 0               
## 
```

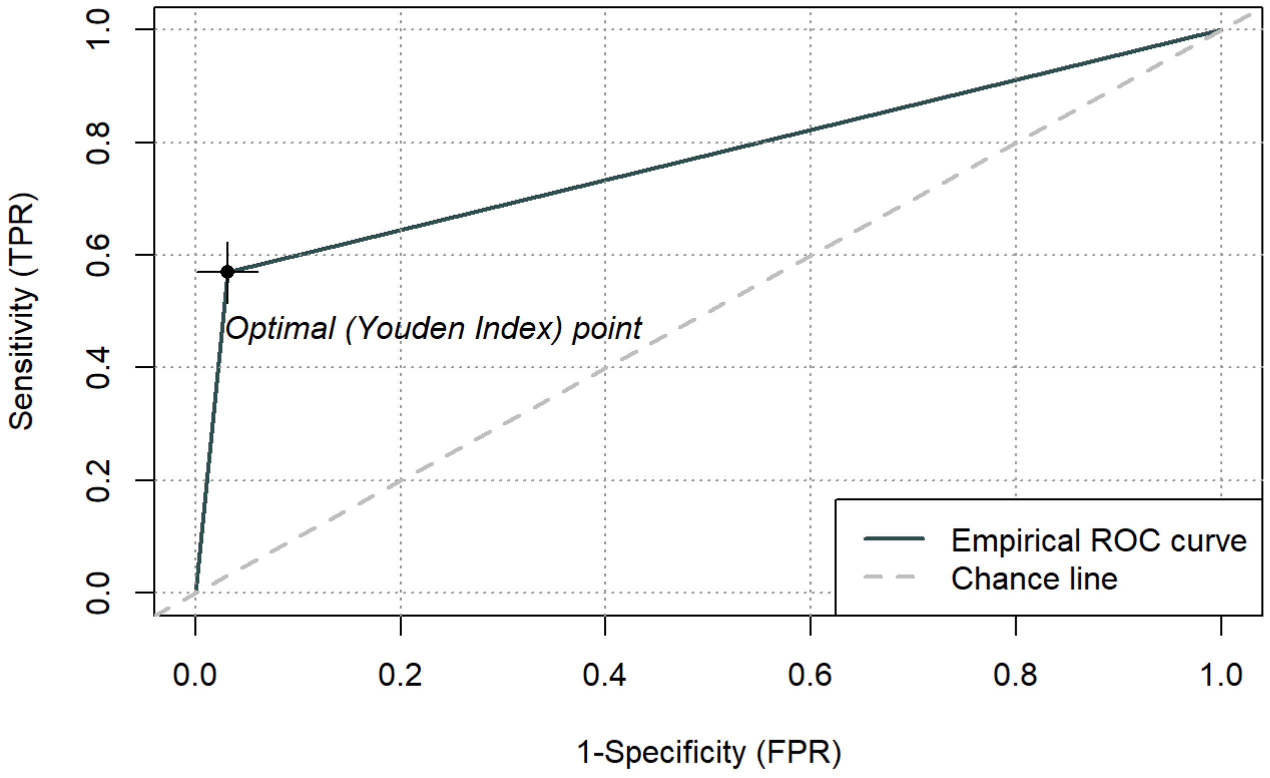
```
#Confusion Matrix for Validation
confusionMatrix(pred.class, as.factor(valid.data$`Personal Loan`))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1736   81
##          1   56  107
##
##                Accuracy : 0.9308
##                  95% CI : (0.9187, 0.9416)
##     No Information Rate : 0.9051
##     P-Value [Acc > NIR] : 2.709e-05
##
##                   Kappa : 0.5719
##
##  Mcnemar's Test P-Value : 0.04032
##
##             Sensitivity : 0.9688
##             Specificity : 0.5691
##          Pos Pred Value : 0.9554
##          Neg Pred Value : 0.6564
##              Prevalence : 0.9051
##          Detection Rate : 0.8768
##    Detection Prevalence : 0.9177
##       Balanced Accuracy : 0.7689
##
##        'Positive' Class : 0
##
```

```
#ROC for Training
plot(rocit(score=as.numeric(fit.class),class=as.numeric(train.data$`Personal Loan
`)))
```



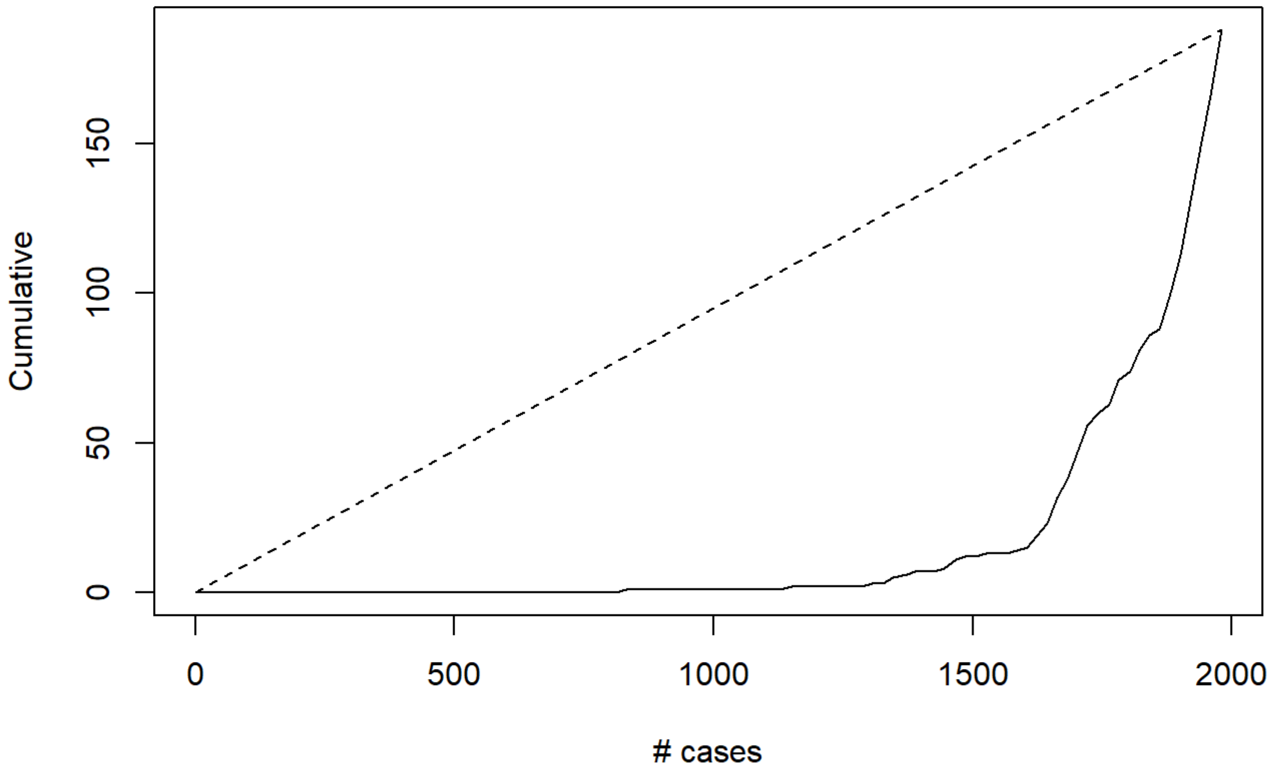
```
#ROC for Testing
plot(rocit(score=as.numeric(pred.class),class=as.numeric(valid.data$`Personal Loan
`)))
```



```
gain <- gains(ifelse(valid.data$`Personal Loan`==1,1,0), pred.prob[,1], groups=100)

# Plot the Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(valid.data$`Personal Loan`==1))~c(0,gain$cume.obs),
     xlab="# cases", ylab="Cumulative", main="", type="l")

lines(c(0,sum(valid.data$`Personal Loan`==1))~c(0, dim(valid.data)[1]), lty=2)
```



From the Confusion Matrix results, we know that the accuracy score in the training dataset is 0.9276 (0.9176, 0.9366), and in the validation dataset is 0.9308 (0.9187, 0.9416) meaning that the probability of determining the target correctly is very high.

Cohen’s kappa coefficient (κ) is a statistic that is used to measure inter-rater reliability for categorical items, taking into account the possibility of the agreement occurring by chance. As the k values here are both about 0.57, we can say our model does a fairly good job. And Mcnemar’s Test also tells us that our model is reliable since the P-Value is lower than 0.05.

Besides the confusion matrix, we also plotted the ROC curve and Lift Charts. The receiver operating characteristic (ROC) curve is another common tool used with binary classifiers. The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner). When AUC = 1, the classifier can perfectly distinguish between all the Positive and the

Negative class points correctly. If AUC = 0, it means that the classifier would be predicting all Negatives as Positives, and all Positives as Negatives. This means that, the higher the AUC, the better the performance of the model. From the ROC curve, we could see the AUC (Area under the ROC Curve) space of both training and validation data is similar, proven by the accuracy score of data, and the optimal point located around (0.02, 0.58). The TPR (true-positive rate) of training data is a little bit higher than the validation data.

V.Conclusion

In this project, we have used the `Naive Bayes method` to classify and predict the probability of clients from Thera Bank accepting the personal loan service and classify potential clients. Based on the correlation of variables, we chose eight variables (family, education, CD account, age, experience, mortgage, income, and CCAvg) and transformed the quantitative variables into categorical variables to run a Naive Bayes model and classify clients. As for model results, the accuracy of both the training and classification part of data were quite high (above 90%) and Naive Bayes generated decent classification results for this dataset.

VI. Reference

- Bank_Personal_Loan_Modeling data can be found here: https://www.kaggle.com/krantiswalke/bank-personal-loan-modelling (https://www.kaggle.com/krantiswalke/bank-personal-loan-modelling)