Assignment_2

2025-09-26

Loading packages

```
library(readr)
library(dplyr)

##
## Attaching package: 'dplyr'

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

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

library (caret)

## Loading required package: ggplot2

## Loading required package: lattice

library (ISLR)
library(class)
library(gmodels)
```

Importing the dataset

```
UniversalBank <- read_csv("./UniversalBank.csv")

## Rows: 5000 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (14): ID, Age, Experience, Income, ZIP Code, Family, CCAvg, Education,
M...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

Task-1: Perform kNN

```
#Removing ID and ZIP Code from the data
UniversalBank <- UniversalBank %>% select(-ID, -`ZIP Code`)



#checking the datatype of all variables to identify the categorical and
```

```
numeric for creating dummy and normalization
str(UniversalBank)

## tibble [5,000 × 12] (S3: tbl_df/tbl/data.frame)
##  $ Age               : num [1:5000] 25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : num [1:5000] 1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : num [1:5000] 49 34 11 100 45 29 72 22 81 180 ...
##  $ Family            : num [1:5000] 4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num [1:5000] 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9
...
##  $ Education         : num [1:5000] 1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : num [1:5000] 0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal Loan     : num [1:5000] 0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities Account: num [1:5000] 1 1 0 0 0 0 0 0 0 0 ...
##  $ CD Account        : num [1:5000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : num [1:5000] 0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : num [1:5000] 0 0 0 0 1 0 0 1 0 0 ...

#Creating dummy variables for categorical data
#As Education defined as numeric, converting it to factor/categorical to make
dummy variables
UniversalBank$Education <- as.factor(UniversalBank$Education)
edu.dummy <- dummyVars(~Education, data=UniversalBank)
edu.dummy <- predict(edu.dummy,UniversalBank)
head(edu.dummy)

##   Education.1 Education.2 Education.3
## 1           1           0           0
## 2           1           0           0
## 3           1           0           0
## 4           0           1           0
## 5           0           1           0
## 6           0           1           0

#Adding dummy variables to the original dataset
bank.dummy <- UniversalBank %>% select(-Education)
bank.dummy <- cbind(bank.dummy, edu.dummy)


#normalizing the numeric data
bank.numeric <- UniversalBank %>% select(Age, Experience, Income, Family,
CCAvg, Mortgage)
bank.norm <- preProcess(bank.numeric, method=c('range'))
bank.normalized <- predict(bank.norm,bank.numeric)
summary(bank.normalized)

##       Age           Experience          Income            Family
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2727   1st Qu.:0.2826   1st Qu.:0.1435   1st Qu.:0.0000
##  Median :0.5000   Median :0.5000   Median :0.2593   Median :0.3333
##  Mean   :0.5077   Mean   :0.5023   Mean   :0.3045   Mean   :0.4655
```

```
##  3rd Qu.:0.7273   3rd Qu.:0.7174   3rd Qu.:0.4167   3rd Qu.:0.6667
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##      CCAvg           Mortgage
##  Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0700   1st Qu.:0.00000
##  Median :0.1500   Median :0.00000
##  Mean   :0.1938   Mean   :0.08897
##  3rd Qu.:0.2500   3rd Qu.:0.15906
##  Max.   :1.0000   Max.   :1.00000
```

```
#Adding two dataset: one with dummy variables and the normalized variables
into one dataframe
Unibank <- bank.dummy %>% select(-Age, -Experience, -Income, -Family, -CCAvg,
-Mortgage)
Unibank <- cbind(Unibank, bank.normalized)


#Train and Valid data partition
set.seed(135)  #help to keep the random with the same seed number
train.index=createDataPartition(Unibank$`Personal Loan`,p=0.6,list=FALSE)
train.data=Unibank[train.index,]     #60%(3000)
valid.data=Unibank[-train.index,]    #40%(2000)


#Organizing the test customer information
#Defining new customer for test
new.customer <- data.frame(
  Age = 40,
  Experience = 10,
  Income = 84,
  Family = 2,
  CCAvg = 2,
  Mortgage = 0,
  Securities.Account = 0,
  CD.Account = 0,
  Online = 1,
  CreditCard = 1,
  Education1 = 0,
  Education2 = 1,
  Education3 = 0
)
#normalizing data for new_customer
ncustomer.normalized <- predict(bank.norm,new.customer)
summary(ncustomer.normalized)
```

```
##       Age           Experience         Income           Family
##  Min.   :0.3864   Min.   :0.2826   Min.   :0.3519   Min.   :0.3333
##  1st Qu.:0.3864   1st Qu.:0.2826   1st Qu.:0.3519   1st Qu.:0.3333
##  Median :0.3864   Median :0.2826   Median :0.3519   Median :0.3333
##  Mean   :0.3864   Mean   :0.2826   Mean   :0.3519   Mean   :0.3333
```

```
##  3rd Qu.:0.3864   3rd Qu.:0.2826   3rd Qu.:0.3519   3rd Qu.:0.3333
##  Max.   :0.3864   Max.   :0.2826   Max.   :0.3519   Max.   :0.3333
##      CCAvg        Mortgage Securities.Account   CD.Account     Online
##  Min.   :0.2   Min.   :0   Min.   :0           Min.   :0    Min.   :1
##  1st Qu.:0.2   1st Qu.:0   1st Qu.:0           1st Qu.:0    1st Qu.:1
##  Median :0.2   Median :0   Median :0           Median :0    Median :1
##  Mean   :0.2   Mean   :0   Mean   :0           Mean   :0    Mean   :1
##  3rd Qu.:0.2   3rd Qu.:0   3rd Qu.:0           3rd Qu.:0    3rd Qu.:1
##  Max.   :0.2   Max.   :0   Max.   :0           Max.   :0    Max.   :1
##    CreditCard   Education1   Education2   Education3
##  Min.   :1    Min.   :0    Min.   :1    Min.   :0
##  1st Qu.:1    1st Qu.:0    1st Qu.:1    1st Qu.:0
##  Median :1    Median :0    Median :1    Median :0
##  Mean   :1    Mean   :0    Mean   :1    Mean   :0
##  3rd Qu.:1    3rd Qu.:0    3rd Qu.:1    3rd Qu.:0
##  Max.   :1    Max.   :0    Max.   :1    Max.   :0

#separating the predictors and labels for kNN function
#predictors
train.predictors <- train.data[,2:14]
ncustomer.predictors <- ncustomer.normalized
#labels
train.labels <- train.data[,1]


#Model testing
ncustomer.label <- knn(train.predictors, ncustomer.predictors,
cl=train.labels, k=1)
ncustomer.label

## [1] 0
## Levels: 0 1
```

##**The result came negative: the value is 0 which depicts that the model predicts that the customer won't accept the loan**.

Task-2: The choice of k

```
##As in last task, I didn't create test data, I am going to use the valid
data to identify the k value

#separating the predictors and labels for kNN function
#predictors
train.predictors <- train.data[,2:14]
valid.predictors <- valid.data[,2:14]
#labels
train.labels <- train.data[,1]
valid.labels <- valid.data[,1]

#Exploring k values with the confusion matrix
```

```r
for (i in 1:15) {
  model.prediction <- knn(train.predictors, valid.predictors,
cl=train.labels, k=i)
  CrossTable(x=valid.labels,y=model.prediction,prop.chisq = FALSE)
}
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1788 |        15 |      1803 |
##              |     0.992 |     0.008 |     0.901 |
##              |     0.962 |     0.106 |           |
##              |     0.894 |     0.007 |           |
## -------------|-----------|-----------|-----------|
##            1 |        70 |       127 |       197 |
##              |     0.355 |     0.645 |     0.098 |
##              |     0.038 |     0.894 |           |
##              |     0.035 |     0.064 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1858 |       142 |      2000 |
##              |     0.929 |     0.071 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
```

```
## 
## 
##              | model.prediction 
## valid.labels |         0 |         1 | Row Total | 
## -------------|-----------|-----------|-----------|
##            0 |      1787 |        16 |      1803 | 
##              |     0.991 |     0.009 |     0.901 | 
##              |     0.956 |     0.122 |           | 
##              |     0.893 |     0.008 |           | 
## -------------|-----------|-----------|-----------|
##            1 |        82 |       115 |       197 | 
##              |     0.416 |     0.584 |     0.098 | 
##              |     0.044 |     0.878 |           | 
##              |     0.041 |     0.058 |           | 
## -------------|-----------|-----------|-----------|
## Column Total |      1869 |       131 |      2000 | 
##              |     0.934 |     0.066 |           | 
## -------------|-----------|-----------|-----------|
## 
## 
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  2000 
## 
## 
##              | model.prediction 
## valid.labels |         0 |         1 | Row Total | 
## -------------|-----------|-----------|-----------|
##            0 |      1798 |         5 |      1803 | 
##              |     0.997 |     0.003 |     0.901 | 
##              |     0.954 |     0.043 |           | 
##              |     0.899 |     0.002 |           | 
## -------------|-----------|-----------|-----------|
##            1 |        86 |       111 |       197 | 
##              |     0.437 |     0.563 |     0.098 | 
##              |     0.046 |     0.957 |           | 
##              |     0.043 |     0.056 |           | 
## -------------|-----------|-----------|-----------|
## Column Total |      1884 |       116 |      2000 | 
##              |     0.942 |     0.058 |           | 
## -------------|-----------|-----------|-----------|
```

```
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1798 |         5 |      1803 |
##              |     0.997 |     0.003 |     0.901 |
##              |     0.952 |     0.045 |           |
##              |     0.899 |     0.002 |           |
## -------------|-----------|-----------|-----------|
##            1 |        91 |       106 |       197 |
##              |     0.462 |     0.538 |     0.098 |
##              |     0.048 |     0.955 |           |
##              |     0.045 |     0.053 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1889 |       111 |      2000 |
##              |     0.945 |     0.056 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
```

```
## -------------|-----------|-----------|-----------|
##            0 |      1800 |         3 |      1803 |
##              |     0.998 |     0.002 |     0.901 |
##              |     0.950 |     0.028 |           |
##              |     0.900 |     0.002 |           |
## -------------|-----------|-----------|-----------|
##            1 |        94 |       103 |       197 |
##              |     0.477 |     0.523 |     0.098 |
##              |     0.050 |     0.972 |           |
##              |     0.047 |     0.051 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1894 |       106 |      2000 |
##              |     0.947 |     0.053 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1800 |         3 |      1803 |
##              |     0.998 |     0.002 |     0.901 |
##              |     0.947 |     0.030 |           |
##              |     0.900 |     0.002 |           |
## -------------|-----------|-----------|-----------|
##            1 |       100 |        97 |       197 |
##              |     0.508 |     0.492 |     0.098 |
##              |     0.053 |     0.970 |           |
##              |     0.050 |     0.048 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1900 |       100 |      2000 |
##              |     0.950 |     0.050 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1801 |         2 |      1803 |
##              |     0.999 |     0.001 |     0.901 |
##              |     0.945 |     0.021 |           |
##              |     0.900 |     0.001 |           |
## -------------|-----------|-----------|-----------|
##            1 |       105 |        92 |       197 |
##              |     0.533 |     0.467 |     0.098 |
##              |     0.055 |     0.979 |           |
##              |     0.052 |     0.046 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1906 |        94 |      2000 |
##              |     0.953 |     0.047 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1800 |         3 |      1803 |
##              |     0.998 |     0.002 |     0.901 |
##              |     0.944 |     0.032 |           |
```

```
##               |     0.900 |     0.002 |           |
## --------------|-----------|-----------|-----------|
##             1 |       107 |        90 |       197 |
##               |     0.543 |     0.457 |     0.098 |
##               |     0.056 |     0.968 |           |
##               |     0.053 |     0.045 |           |
## --------------|-----------|-----------|-----------|
## Column Total |      1907 |        93 |      2000 |
##               |     0.954 |     0.046 |           |
## --------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##               | model.prediction
## valid.labels |         0 |         1 | Row Total |
## --------------|-----------|-----------|-----------|
##             0 |      1801 |         2 |      1803 |
##               |     0.999 |     0.001 |     0.901 |
##               |     0.941 |     0.023 |           |
##               |     0.900 |     0.001 |           |
## --------------|-----------|-----------|-----------|
##             1 |       113 |        84 |       197 |
##               |     0.574 |     0.426 |     0.098 |
##               |     0.059 |     0.977 |           |
##               |     0.056 |     0.042 |           |
## --------------|-----------|-----------|-----------|
## Column Total |      1914 |        86 |      2000 |
##               |     0.957 |     0.043 |           |
## --------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
```

```
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1800 |         3 |      1803 |
##              |     0.998 |     0.002 |     0.901 |
##              |     0.937 |     0.038 |           |
##              |     0.900 |     0.002 |           |
## -------------|-----------|-----------|-----------|
##            1 |       121 |        76 |       197 |
##              |     0.614 |     0.386 |     0.098 |
##              |     0.063 |     0.962 |           |
##              |     0.060 |     0.038 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1921 |        79 |      2000 |
##              |     0.961 |     0.040 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1801 |         2 |      1803 |
##              |     0.999 |     0.001 |     0.901 |
##              |     0.935 |     0.027 |           |
##              |     0.900 |     0.001 |           |
## -------------|-----------|-----------|-----------|
##            1 |       125 |        72 |       197 |
##              |     0.635 |     0.365 |     0.098 |
```

```
##               |     0.065 |     0.973 |           |
##               |     0.062 |     0.036 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1926 |        74 |      2000 |
##               |     0.963 |     0.037 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##               | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##             0 |      1802 |         1 |      1803 |
##               |     0.999 |     0.001 |     0.901 |
##               |     0.933 |     0.014 |           |
##               |     0.901 |     0.000 |           |
## -------------|-----------|-----------|-----------|
##             1 |       129 |        68 |       197 |
##               |     0.655 |     0.345 |     0.098 |
##               |     0.067 |     0.986 |           |
##               |     0.064 |     0.034 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1931 |        69 |      2000 |
##               |     0.966 |     0.034 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
```

```
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1803 |         0 |      1803 |
##              |     1.000 |     0.000 |     0.901 |
##              |     0.933 |     0.000 |           |
##              |     0.901 |     0.000 |           |
## -------------|-----------|-----------|-----------|
##            1 |       129 |        68 |       197 |
##              |     0.655 |     0.345 |     0.098 |
##              |     0.067 |     1.000 |           |
##              |     0.064 |     0.034 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1932 |        68 |      2000 |
##              |     0.966 |     0.034 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1801 |         2 |      1803 |
##              |     0.999 |     0.001 |     0.901 |
##              |     0.933 |     0.029 |           |
##              |     0.900 |     0.001 |           |
## -------------|-----------|-----------|-----------|
##            1 |       130 |        67 |       197 |
##              |     0.660 |     0.340 |     0.098 |
##              |     0.067 |     0.971 |           |
##              |     0.065 |     0.034 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1931 |        69 |      2000 |
```

```
##              |     0.966 |     0.034 |           |
## -------------|-----------|-----------|-----------|
##
##
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1803 |         0 |      1803 |
##              |     1.000 |     0.000 |     0.901 |
##              |     0.930 |     0.000 |           |
##              |     0.901 |     0.000 |           |
## -------------|-----------|-----------|-----------|
##            1 |       136 |        61 |       197 |
##              |     0.690 |     0.310 |     0.098 |
##              |     0.070 |     1.000 |           |
##              |     0.068 |     0.030 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1939 |        61 |      2000 |
##              |     0.970 |     0.030 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```r
#Finding the k best value from accuracy data
accuracy <- c()
for (i in 1:15) {
  model.prediction <- knn(train.predictors, valid.predictors,
cl=train.labels, k=i)
  accuracy <- mean(model.prediction == valid.labels)  # validation accuracy
  accuracy <- print(accuracy)
}
```

```
## [1] 0.9575
## [1] 0.951
## [1] 0.9545
## [1] 0.955
```

```
## [1] 0.9515
## [1] 0.9485
## [1] 0.9465
## [1] 0.941
## [1] 0.9425
## [1] 0.9385
## [1] 0.9365
## [1] 0.9355
## [1] 0.9355
## [1] 0.9325
## [1] 0.932
```

```r
#Finding the best k value from Kappa, as this is kind of imbalanced dataset
for(i in 1:15) {
  model.prediction <- knn(train = train.predictors, test = valid.predictors,
cl = train.labels, k = i)
  valid.labels  <- factor(valid.labels, levels = c("0", "1"))
  confusion.matrix <- confusionMatrix(model.prediction, valid.labels,
positive = "1")
  print(confusion.matrix$overall['Kappa'])
}
```

```
##    Kappa
## 0.726711
##     Kappa
## 0.6582501
##    Kappa
## 0.686367
##     Kappa
## 0.6776899
##     Kappa
## 0.6561722
##    Kappa
## 0.640421
##     Kappa
## 0.6073135
##     Kappa
## 0.5702249
##     Kappa
## 0.5677634
##    Kappa
## 0.518553
##     Kappa
## 0.5047227
##     Kappa
## 0.5179318
##     Kappa
## 0.4872896
##     Kappa
## 0.4770351
```

```
##     Kappa
## 0.4471163
```

#I tried multiple way to find the best k-created confusion matrix, calculated accuracy, and also the kappa to find the best k, as to me, it is an imbalanced dataset. The best k value I got is 1; only in this value I got highest accuracy, highest kappa and lowest False Negative cases, which is the most costly or risky case in this this model.

Task-3: confusion matrix for the validation data that results from using the best k

```
#separating the predictors and labels for kNN function
#predictors
train.predictors <- train.data[,2:14]
valid.predictors <- valid.data[,2:14]
#labels
train.labels <- train.data[,1]
valid.labels <- valid.data[,1]

#According to the result of my last task, k=1 is the best choice for this
model
model.prediction <- knn(train.predictors, valid.predictors, cl=train.labels,
k=1)
CrossTable(x=valid.labels,y=model.prediction,prop.chisq = FALSE)

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | model.prediction
## valid.labels |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1788 |        15 |      1803 |
##              |     0.992 |     0.008 |     0.901 |
##              |     0.962 |     0.106 |           |
##              |     0.894 |     0.007 |           |
## -------------|-----------|-----------|-----------|
##            1 |        70 |       127 |       197 |
##              |     0.355 |     0.645 |     0.098 |
##              |     0.038 |     0.894 |           |
```

```
##               |     0.035 |     0.064 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1858 |       142 |      2000 |
##              |     0.929 |     0.071 |           |
## -------------|-----------|-----------|-----------|
##
##
```

Task-4: Classify the customer using the best k.

```
#Organizing the test customer information
#Defining new customer for test
new.customer <- data.frame(
  Age = 40,
  Experience = 10,
  Income = 84,
  Family = 2,
  CCAvg = 2,
  Mortgage = 0,
  Securities.Account = 0,
  CD.Account = 0,
  Online = 1,
  CreditCard = 1,
  Education1 = 0,
  Education2 = 1,
  Education3 = 0
)
#normalizing data for new_customer
ncustomer.normalized <- predict(bank.norm,new.customer)


#separating the predictors and labels for kNN function
#predictors
train.predictors <- train.data[,2:14]
ncustomer.predictors <- ncustomer.normalized
#labels
train.labels <- train.data[,1]


#Model testing
ncustomer.label <- knn(train.predictors, ncustomer.predictors,
cl=train.labels, k=1)
ncustomer.label

## [1] 0
## Levels: 0 1
```

#**From task-2, the best k value I got is 1, also, in task-1, we were supposed to use the k=1, and the customer detail for both of the task-1 & 4 is ame, so the result and the code of this task will be same as task-1. The customer won't accept the loan**

Task-5: Comparing the confusion matrix

```
#Train, Valid, and Test data partition
set.seed(246)  #help to keep the random with the same seed number
train.index5=createDataPartition(Unibank$`Personal Loan`,p=0.5,list=FALSE)
train.data5=Unibank[train.index5,]          #train(50%)
split.data5=Unibank[-train.index5,]
split.index5=createDataPartition(split.data5$`Personal
Loan`,p=0.6,list=FALSE)
valid.data5=split.data5[split.index5,]      #valid(30%)
test.data5=split.data5[-split.index5,]      #test(20%)

#Comparing train and test
#separating the predictors and labels for kNN function
#predictors
train.predictors5 <- train.data5[,2:14]
valid.predictors5 <- valid.data5[,2:14]
test.predictors5 <- test.data5[,2:14]
#labels
train.labels5 <- train.data5[,1]
valid.labels5 <- valid.data5[,1]
test.labels5 <- test.data5[,1]

#Model testing
train.model <- knn(train = train.predictors5, test = train.predictors5, cl =
train.labels5, k = 1)
valid.model <- knn(train = train.predictors5, test = valid.predictors5, cl =
train.labels5, k = 1)
test.model  <- knn(train = train.predictors5, test = test.predictors5, cl =
train.labels5, k = 1)

CrossTable(x=train.labels5,y=train.model,prop.chisq = FALSE)

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2500
##
##
##               | train.model
## train.labels5 |         0 |         1 | Row Total |
## --------------|-----------|-----------|-----------|
```

```
##              0 |      2261 |         0 |      2261 |
##                |     1.000 |     0.000 |     0.904 |
##                |     1.000 |     0.000 |           |
##                |     0.904 |     0.000 |           |
## ---------------|-----------|-----------|-----------|
##              1 |         0 |       239 |       239 |
##                |     0.000 |     1.000 |     0.096 |
##                |     0.000 |     1.000 |           |
##                |     0.000 |     0.096 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |      2261 |       239 |      2500 |
##                |     0.904 |     0.096 |           |
## ---------------|-----------|-----------|-----------|
##
##
```

```
CrossTable(x=valid.labels5,y=valid.model,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1500
##
##
##                | valid.model
## valid.labels5 |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |      1343 |        17 |      1360 |
##                |     0.988 |     0.012 |     0.907 |
##                |     0.970 |     0.148 |           |
##                |     0.895 |     0.011 |           |
## ---------------|-----------|-----------|-----------|
##              1 |        42 |        98 |       140 |
##                |     0.300 |     0.700 |     0.093 |
##                |     0.030 |     0.852 |           |
##                |     0.028 |     0.065 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |      1385 |       115 |      1500 |
##                |     0.923 |     0.077 |           |
## ---------------|-----------|-----------|-----------|
##
##
```

```r
CrossTable(x=test.labels5,y=test.model,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1000
##
##
##              | test.model
## test.labels5 |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       890 |         9 |       899 |
##              |     0.990 |     0.010 |     0.899 |
##              |     0.962 |     0.120 |           |
##              |     0.890 |     0.009 |           |
## -------------|-----------|-----------|-----------|
##            1 |        35 |        66 |       101 |
##              |     0.347 |     0.653 |     0.101 |
##              |     0.038 |     0.880 |           |
##              |     0.035 |     0.066 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       925 |        75 |      1000 |
##              |     0.925 |     0.075 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```r
train.acc <- mean(train.model == train.labels5)
valid.acc <- mean(valid.model == valid.labels)
```

```
## Warning in `==.default`(valid.model, valid.labels): longer object length
is not
## a multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple
of
## shorter object length
```

```r
test.acc <- mean(test.model == test.labels5)

train.acc
```

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

```
valid.acc

## [1] 0.838

test.acc

## [1] 0.956
```

#**From the confusion matrix, it is clearly visible that there is no error for train data model. The accuracy is 100% for this model, because all the samples have been used to train the model. On the other hand, the total error in valid data model is 59 and in test data model is 44 with False Negative cases 42 and 35 respectively. So, if we want to prioritize models based on their correct prediction or FN cases, test model is better than the valid data model. Also, in terms of accuracy, test model's accuracy rate is higher than the valid data model, but still the vaalues are closer to each other, which represents a good generalization**