ML Assignment-2

Abhinay Chiranjeeth Marneni

2022-10-06

Loading Packages

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
library(class)
library(ISLR)
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(ggplot2)
library(fastDummies)
library(knitr)
```

#Importing the Universalbank.csv files to exacted the data into variables

```
getwd()
```

```
## [1] "C:/Users/abhin/Downloads"
```

```
setwd("C:/Users/abhin/Downloads")
Universal_bank <- read.csv("UniversalBank.csv")
Universal_bank <-Universal_bank[,c(2,3,4,6,7,8,9,10,11,12,13,14)]
Universal_bank$Personal.Loan<-as.factor(Universal_bank$Personal.Loan)
Universal_bank$Education<-as.factor(Universal_bank$Education)
Universal_bank <- dummy_columns(Universal_bank,select_columns = 'Education')
Universal_bank <-Universal_bank[,c("Personal.Loan",'Age','Experience','Income',"Family","CCAvg","Educat
summary(Universal_bank)
```

```
##  Personal.Loan      Age            Experience       Income           Family
##  0:4520        Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000
##  1: 480        1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000
##                Median :45.00   Median :20.0   Median : 64.00   Median :2.000
##                Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :2.396
##                3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:3.000
##                Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000
##      CCAvg          Education_1      Education_2      Education_3
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 0.700   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 1.500   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   : 1.938   Mean   :0.4192   Mean   :0.2806   Mean   :0.3002
##  3rd Qu.: 2.500   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :10.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##     Mortgage     Securities.Account   CD.Account         Online
##  Min.   :  0.0   Min.   :0.0000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.:  0.0   1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median :  0.0   Median :0.0000     Median :0.0000   Median :1.0000
##  Mean   : 56.5   Mean   :0.1044     Mean   :0.0604   Mean   :0.5968
##  3rd Qu.:101.0   3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :635.0   Max.   :1.0000     Max.   :1.0000   Max.   :1.0000
##    CreditCard
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.294
##  3rd Qu.:1.000
##  Max.   :1.000
```

To separation the collecting data into training(60%) and validation(40%) ,Normalize Data and testing data

```
set.seed(400)
Index_Train<-createDataPartition(Universal_bank$Personal.Loan, p=0.6, list=FALSE)
Universal_bank_Train <-Universal_bank[Index_Train,]
Universal_bank_Validation  <-Universal_bank[-Index_Train,]
train_label<- Universal_bank_Train[,1]
validation_label<- Universal_bank_Validation[,1]
norm_var <- c("Age","Experience","Income","Family","CCAvg","Mortgage")
norm_model<-preProcess(Universal_bank_Train[,norm_var], method = c("center", "scale"))
Universal_bank_norm_Train <-predict(norm_model,Universal_bank_Train)
```

```
Universal_bank_norm_Validation  <-predict(norm_model,Universal_bank_Validation)
Universal_bank_test<-Universal_bank[0,-1]
test_data<-c(40,10,84,2,2,0,1,0,0,0,0,1,1)
Universal_bank_test[nrow(Universal_bank_test) + 1, ] <- test_data
Universal_bank_norm_test<-predict(norm_model,Universal_bank_test)
```

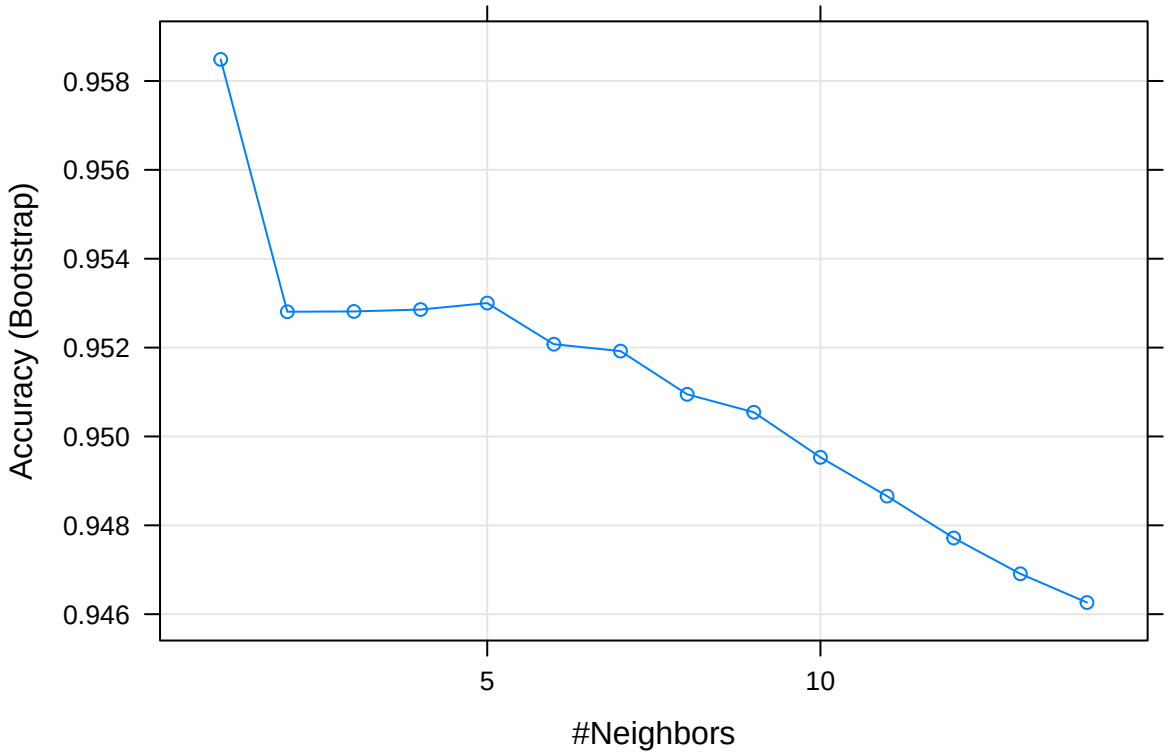
KNN model Classification

```
set.seed(500)
train_predictor<- Universal_bank_norm_Train[-1]
Loan_predicted <-knn(train_predictor,Universal_bank_norm_test, cl=train_label,
               k=1)
print(Loan_predicted)
```

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

Again separated the data sets to apply the K-NN method

```
Serach_grid <- expand.grid(k=c(1:14))
trctrl <- trainControl(method = "boot")
model<-train(Personal.Loan~.,data=Universal_bank_norm_Train,trControl=trctrl,
             method="knn", tuneGrid=Serach_grid
            )
plot(model)
```




```
model
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   13 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3000, 3000, 3000, 3000, 3000, 3000, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9584862  0.7303111
##    2  0.9528069  0.6886659
##    3  0.9528136  0.6839532
##    4  0.9528560  0.6793611
##    5  0.9530020  0.6725484
##    6  0.9520759  0.6601402
##    7  0.9519213  0.6535064
##    8  0.9509482  0.6420979
##    9  0.9505441  0.6349643
##   10  0.9495303  0.6245540
##   11  0.9486573  0.6147601
##   12  0.9477150  0.6057204
```

```
##   13  0.9469096  0.5954234
##   14  0.9462632  0.5886258
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

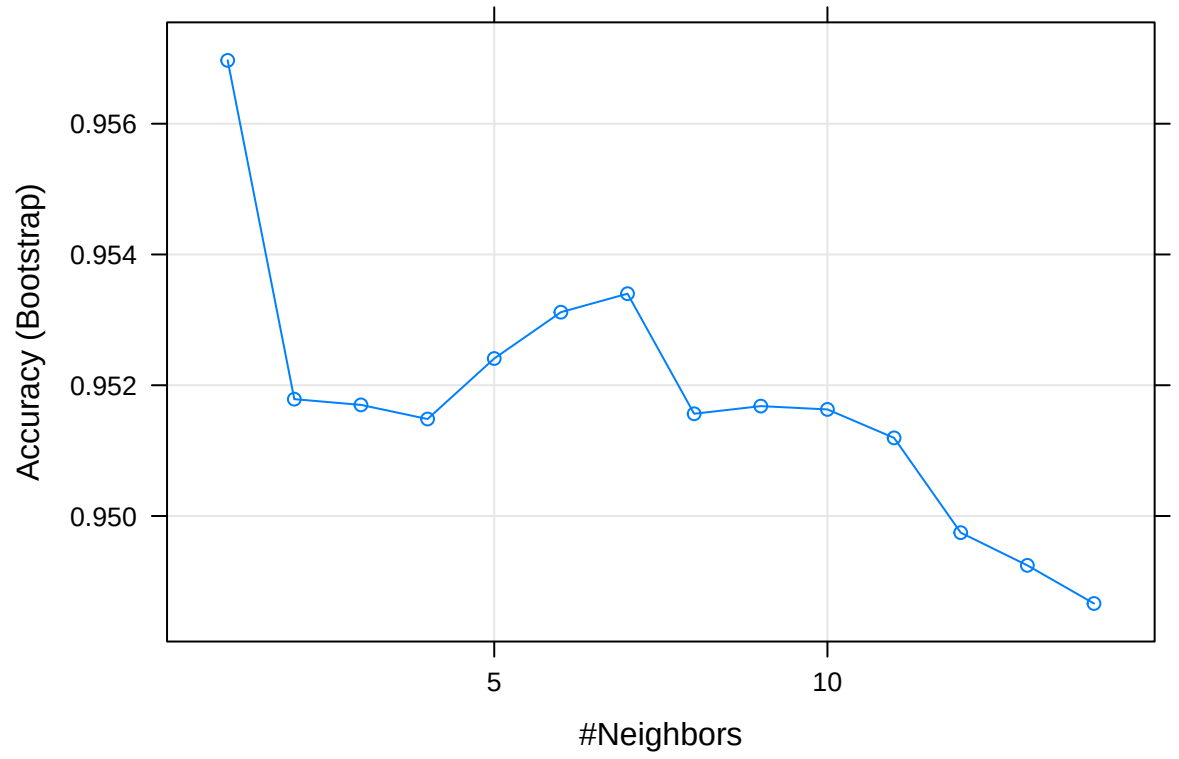
```
model$bestTune[[1]]
```

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

```
Loan_predicted_1 <-knn(train_predictor,Universal_bank_norm_test, cl=train_label,
               k=model$bestTune[[1]])
Loan_predicted_1
```

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

```
validation_model<-train(Personal.Loan~.,data=Universal_bank_norm_Validation,trControl=trctrl,
             method="knn", tuneGrid=Serach_grid
            )
Loan_predicted_2 <-knn(train_predictor,Universal_bank_norm_Validation[-1], cl=train_label,
               k=model$bestTune[[1]])
valid.model<-train(Universal_bank_norm_Validation[-1],Loan_predicted_2,trControl=trctrl,
             method="knn", tuneGrid=Serach_grid
            )
plot(valid.model)
```



```
valid.model
```

```
## k-Nearest Neighbors
##
## 2000 samples
##   13 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 2000, 2000, 2000, 2000, 2000, 2000, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9569687  0.6785295
##    2  0.9517870  0.6373869
##    3  0.9516996  0.6266615
##    4  0.9514832  0.6121821
##    5  0.9524087  0.6134099
##    6  0.9531173  0.6141760
##    7  0.9534002  0.6097572
##    8  0.9515638  0.5881753
##    9  0.9516808  0.5840943
##   10  0.9516303  0.5806399
##   11  0.9511948  0.5720940
##   12  0.9497454  0.5566718
##   13  0.9492452  0.5460756
##   14  0.9486618  0.5364442
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
library("gmodels")
s<-CrossTable(x=validation_label,y=Loan_predicted_2, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##                  | Loan_predicted_2
## validation_label |         0 |         1 | Row Total |
## -----------------|-----------|-----------|-----------|
##                0 |      1790 |        18 |      1808 |
```

```
##                  |     0.990 |     0.010 |     0.904 |
##                  |     0.974 |     0.110 |           |
##                  |     0.895 |     0.009 |           |
## -----------------|-----------|-----------|-----------|
##                1 |        47 |       145 |       192 |
##                  |     0.245 |     0.755 |     0.096 |
##                  |     0.026 |     0.890 |           |
##                  |     0.024 |     0.072 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |      1837 |       163 |      2000 |
##                  |     0.918 |     0.082 |           |
## -----------------|-----------|-----------|-----------|
##
##
```