Fundamentals of Machine Learning Assignment 2

Karthik Badiganti

2022-09-27

Loading Packages

```
library(class)
library(caret)
```

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

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

```
library(ISLR)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
library(fastDummies)
library(knitr)
```

Importing & Cleaning Data

```
Universal_bank <- read.csv("C://Users//gbkar//Documents//R Scripts//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
```

Data Partition and Normalization

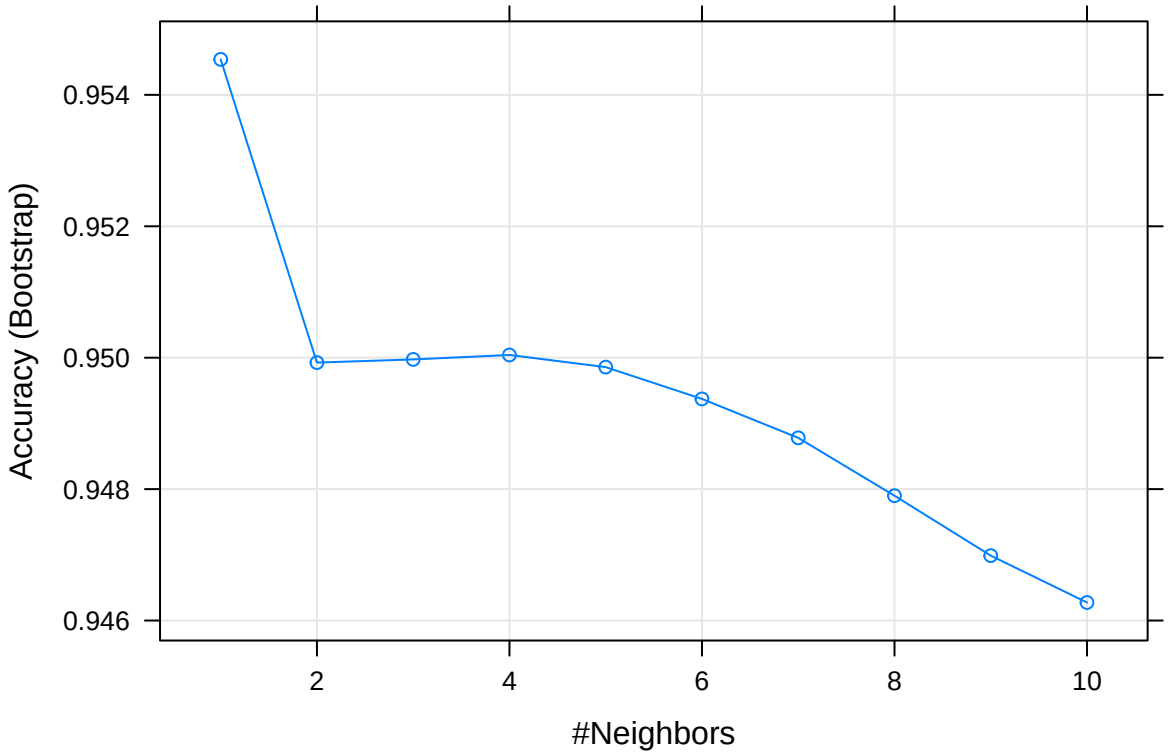
```
set.seed(123)
Index_Train<-createDataPartition(Universal_bank$Personal.Loan, p=0.6, list=FALSE)
Universal_bank_Train <-Universal_bank[Index_Train,]
Universal_bank_Validation  <-Universal_bank[-Index_Train,]

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 Classification

```
# Problem 1



set.seed(3333)



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
```

```
# Problem 2


Serach_grid <- expand.grid(k=c(1:10))
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  0.9545401  0.7104829
##    2  0.9499258  0.6750275
##    3  0.9499735  0.6673979
##    4  0.9500414  0.6640026
##    5  0.9498565  0.6552179
##    6  0.9493724  0.6449461
##    7  0.9487801  0.6335236
##    8  0.9478999  0.6229991
##    9  0.9469878  0.6107083
##   10  0.9462745  0.6028088
##
## Accuracy was used to select the optimal model using the largest value.
```

```
## The final value used for the model was k = 1.
```

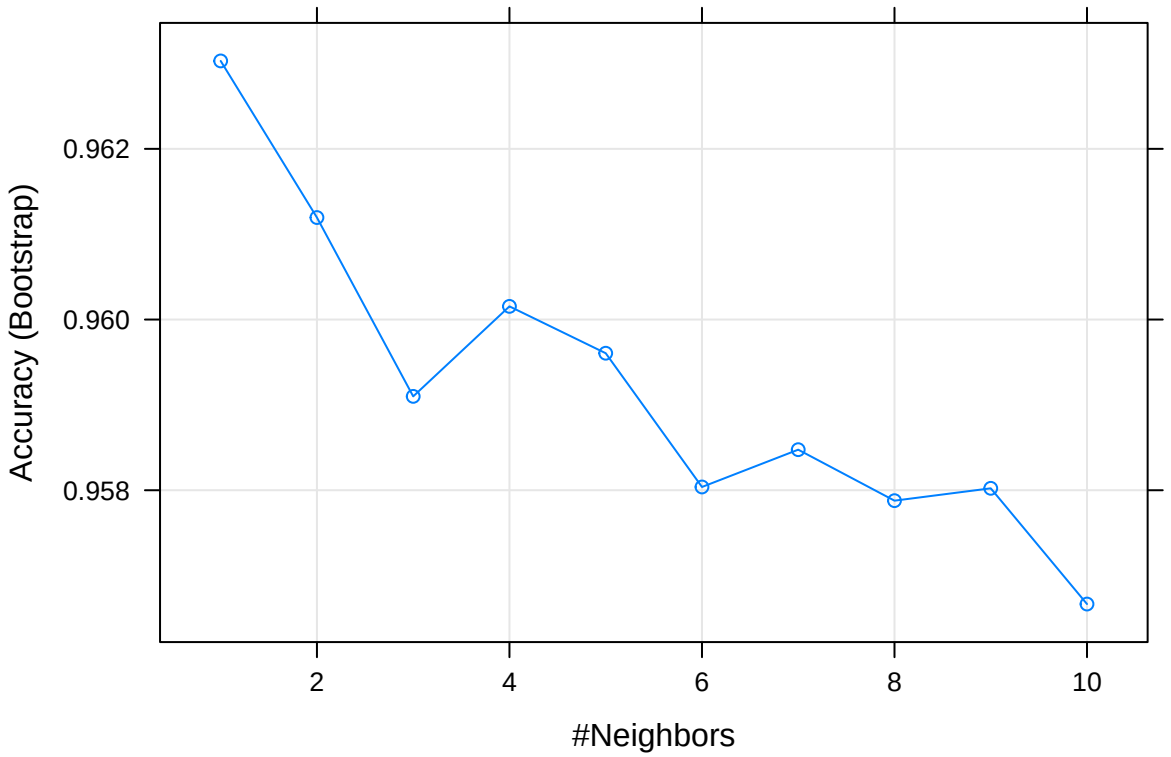
```
model$bestTune[[1]]
```

```
## [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  0.9630304  0.6968415
##    2  0.9611952  0.6789246
##    3  0.9590998  0.6542558
##    4  0.9601544  0.6572087
##    5  0.9596053  0.6418207
##    6  0.9580379  0.6196657
##    7  0.9584752  0.6168242
##    8  0.9578778  0.6063563
##    9  0.9580222  0.6036339
##   10  0.9566663  0.5846940
##
## 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 |      1796 |        12 |      1808 |
##                  |     0.993 |     0.007 |     0.904 |
##                  |     0.971 |     0.079 |           |
##                  |     0.898 |     0.006 |           |
## -----------------|-----------|-----------|-----------|
```

```
##                1 |        53 |       139 |       192 |
##                  |     0.276 |     0.724 |     0.096 |
##                  |     0.029 |     0.921 |           |
##                  |     0.026 |     0.070 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |      1849 |       151 |      2000 |
##                  |     0.924 |     0.075 |           |
## -----------------|-----------|-----------|-----------|
##
##
```