problem7-2

Mehrab Atighi

11/19/2021

problem

Personal Loan Acceptance. Universal Bank is a relatively young bank growing rapidly in terms of overall customer acquisition. The majority of these customers are liability customers (depositors) with varying sizes of relationship with the bank. The customer base of asset customers (borrowers) is quite small, and the bank is interested in expanding this base rapidly to bring in more loan business. In particular, it wants to explore ways of converting its liability customers to personal loan customers (while retaining them as depositors).
A campaign that the bank ran last year for liability customers showed a healthy conversion rate of over 9% success. This has encouraged the retail marketing department to devise smarter campaigns with better target marketing. The goal is to use k-NN to predict whether a new customer will accept a loan offer. This will serve as the basis for the design of a new campaign.
The file UniversalBank.csv contains data on 5000 customers. The data include customer demographic information (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). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign. Partition the data into training (60%) and validation (40%) sets.

problem

a. Consider the following customer: Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0, Education_2 = 1, Education_3 = 0, Mortgage = 0, Securities Account = 0, CD Account = 0, Online = 1, and Credit Card = 1. Perform a k-NN classification with all predictors except ID and ZIP code using k = 1. Remember to transform categorical predictors with more than two categories into dummy variables first. Specify the success class as 1 (loan acceptance), and use the default cutoff value of 0.5. How would this customer be classified?

b. What is a choice of k that balances between overfitting and ignoring the predictor information?

c. Show the confusion matrix for the validation data that results from using the best k.

problem

d. Consider the following customer: Age $= 40$, Experience $= 10$, Income $= 84$, Family $= 2$, CCAvg $= 2$, Education_1 $= 0$, Education_2 $= 1$, Education_3 $= 0$, Mortgage $= 0$, Securities Account $= 0$, CD Account $= 0$, Online $= 1$ and Credit Card $=$ 1. Classify the customer using the best k.

e. Repartition the data, this time into training, validation, and test sets (50% : 30% : 20%). Apply the k-NN method with the k chosen above. Compare the confusion matrix of the test set with that of the training and validation sets. Comment on the differences and their reason.

solution

At the first we want to introduction our dataset in R.

```
Data<-read.csv("F:/lessons/Data mining/Data/UniversalBank.csv")
dim(Data)
```

```
## [1] 5000   14
```

```
head(Data,4)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
```

solution

Now we want to transform Educatin variable to tree dummy variables, and we names them :
Education_1 , Education_2 , Educatioin_3 .

```r
data=Data[,-c(1,5)]
data$Education_1=c(1:nrow(data))
data$Education_2=c(1:nrow(data))
data$Education_3=c(1:nrow(data))
for(i in 1:nrow(data)){
  if(data$Education[i]==1){
    data$Education_1[i] = 1
  }else(data$Education_1[i] = 0)
  if(data$Education[i]==2){
    data$Education_2[i] = 1
  }else(data$Education_2[i] = 0)
  if(data$Education[i]==3){
    data$Education_3[i] = 1
  }else(data$Education_3[i] = 0)
}
data$Education=c()
```

solution

Now we want to see new data heads and dimantion.

```
head(data)
```

```
##   Age Experience Income Family CCAvg Mortgage Personal.Loan Securities.Account
## 1  25          1     49      4   1.6        0             0                  1
## 2  45         19     34      3   1.5        0             0                  1
## 3  39         15     11      1   1.0        0             0                  0
## 4  35          9    100      1   2.7        0             0                  0
## 5  35          8     45      4   1.0        0             0                  0
## 6  37         13     29      4   0.4      155             0                  0
##   CD.Account Online CreditCard Education_1 Education_2 Education_3
## 1          0      0          0           1           0           0
## 2          0      0          0           1           0           0
## 3          0      0          0           1           0           0
## 4          0      0          0           0           1           0
## 5          0      0          1           0           1           0
## 6          0      1          0           0           1           0
```

```
dim(data)
```

```
## [1] 5000   14
```

```
summary(data)
```

```
##       Age          Experience       Income           Family     
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000  
##  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          Mortgage      Personal.Loan   Securities.Account
##  Min.   : 0.000   Min.   :  0.0   Min.   :0.000   Min.   :0.0000   
##  1st Qu.: 0.700   1st Qu.:  0.0   1st Qu.:0.000   1st Qu.:0.0000   
```

solution

we want to select the train and test subsets of dataset.

```
set.seed(5)
n=sample(c(0,1) , nrow(data) ,
         prob = c(0.6 , 0.4) ,replace = TRUE)
train_data=data[which(n==0),]
test_data=data[which(n==1),]
```

solution

we are going to work with normalized data so we should make normal data frame, and using preProcess function to normalizing variables.

```
train_normal<- train_data
test_normal<-test_data
data_normal<-data
new.df=data.frame(Age = 40, Experience = 10 , Income = 84,
                  Family = 2 , CCAvg = 2, Education_1= 0 ,Education_2= 1,
                  Education_3=0 , Mortgage = 0 , Securities.Account = 0 ,
                  CD.Account = 0 , Online = 1 , CreditCard =1)

#install.packages("caret")
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
norm.values<-preProcess(train_data[,-7] , method = c("center" , "scale"))
train_normal<-predict(norm.values , train_data[,-7])
test_normal<-predict(norm.values , test_data[,-7])
data_normal<-predict(norm.values , data[,-7])
new.norm.df <-predict(norm.values, new.df)
n2=sample(c(0,1) , nrow(train_normal) ,
         prob = c(0.6 , 0.4) ,replace = TRUE)
train_normal_2=train_normal[which(n2==0),]
train_test_normal=train_normal[which(n2==1),]
```

solution

Now we want to select K value, we should work with all of data

```
#b)
memory.limit(size = 9999999999999)
```

```
## [1] 1e+13
```

```
library(caret);library(e1071)
accuracy_data=data.frame(k = seq(1,nrow(train_normal_2),1) , accuracy=rep(0,nrow(train_normal_2)))
system.time(
  for(i in 1:nrow(accuracy_data)){
    knn.predict<-FNN::knn(train =train_normal_2[,-7] ,test = train_test_normal[,-7] ,
                          cl = train_normal_2[,7] ,k=i)
    accuracy_data[i , 2] = confusionMatrix(data = knn.predict[1:nrow(train_test_normal)],
                                           factor(train_test_normal[,7]) ,)$overall[1]
  })
```

```
##    user  system elapsed
## 6608.91   52.50 6793.08
```

```
head(accuracy_data)
```

```
##   k  accuracy
## 1 1 0.8358714
## 2 2 0.8942470
## 3 3 0.8824027
## 4 4 0.8959391
## 5 5 0.8959391
## 6 6 0.8967851
```

```
(best = which.max(accuracy_data$accuracy))
```

```
## [1] 15
```

solution

```r
#c)
knn.predict<-FNN::knn(train =train_normal[,-7] ,
                      test = test_normal[,-7] ,
                      cl = train_data[,7] ,k=best)
confusionMatrix(data = knn.predict[1:nrow(test_data)],
                factor(test_data[,7]) )$overall[1]
```

```
## Accuracy
## 0.940482
```

solution

```
#d)
library(FNN)
new.df=data.frame(Age = 40, Experience = 10 , Income = 84,
                  Family = 2 , CCAvg = 2, Education_1= 0 ,
                  Education_2= 1,Education_3=0,Mortgage = 0,
                  Securities.Account = 0 ,CD.Account = 0 ,
                  Online = 1 , CreditCard =1)

nn<-knn(train = train_normal , test = new.df,
        cl =train_data[,7] ,k=best)
r=row.names(train_data)[attr(nn, 'nn.index')]
train_data[r,7]
```

```
##  [1] 1 1 0 1 0 1 1 0 1 1 0 1 1 0 1
```

solution

```
#e)
n=sample(c(0,1,2) , nrow(data) ,prob = c(0.5 , 0.3 ,0.2) ,replace = TRU
train_data_2=data[which(n==0),]
valid_data=data[which(n==1),]
test_data_2=data[which(n==2),]

norm.values<-preProcess(train_data_2[,-7] , method = c("center" , "scal
train_normal_2<-predict(norm.values , train_data_2[,-7])
test_normal_2<-predict(norm.values , test_data_2[,-7])
valid_normal<-predict(norm.values , valid_data[,-7])
data_normal_2<-predict(norm.values , data[,-7])
```

solution

```
#for test data:
knn.predict<-FNN::knn(train = train_normal_2[,-7] ,test = test_normal_2[,-7] ,
                      cl = train_data_2[,7] ,k=best)
confusionMatrix(data = knn.predict[1:nrow(test_data_2)],
                factor(test_data_2[,7]) )$overall[1]
```

```
##  Accuracy
## 0.9478958
```

```
#for valid data:
knn.predict<-FNN::knn(train =train_normal_2[,-7] ,test = valid_normal[,-7] ,
                      cl = train_data_2[,7] ,k=best)
confusionMatrix(data = knn.predict[1:nrow(valid_normal)],
                factor(valid_data[,7]) )$overall[1]
```

```
##  Accuracy
## 0.9423979
```

```
#for train data:
knn.predict<-FNN::knn(train =train_normal_2[,-7] ,test = train_normal_2[,-7] ,
                      cl = train_data_2[,7] ,k=best)
confusionMatrix(data = knn.predict[1:nrow(train_normal_2)],
                factor(train_data_2[,7]) )$overall[1]
```

```
##  Accuracy
## 0.9517736
```