ML Assignment 2

Navid Farzanpay

2026-02-13

```
bank<-read.csv("UniversalBank-1.csv")
dim(bank)
```

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

```
names(bank)
```

```
##  [1] "ID"                 "Age"                "Experience"
##  [4] "Income"             "ZIP.Code"           "Family"
##  [7] "CCAvg"              "Education"          "Mortgage"
## [10] "Personal.Loan"      "Securities.Account" "CD.Account"
## [13] "Online"             "CreditCard"
```

```
bank<-bank[,!(names(bank)%in%c("ID","ZIP.Code"))]
names(bank)
```

```
##  [1] "Age"                "Experience"         "Income"
##  [4] "Family"             "CCAvg"              "Education"
##  [7] "Mortgage"           "Personal.Loan"      "Securities.Account"
## [10] "CD.Account"         "Online"             "CreditCard"
```

```
str(bank$Education)
```

```
##  int [1:5000] 1 1 1 2 2 2 2 3 2 3 ...
```

```
table(bank$Education)
```

```
##
##    1    2    3
## 2096 1403 1501
```

```
bank$Education<-as.factor(bank$Education)
library(caret)
```

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

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

```
dummies<-dummyVars(Personal.Loan~.,data=bank)
bank_dummy<-data.frame(predict(dummies,newdata=bank))
names(bank_dummy)
```

```
##  [1] "Age"                "Experience"         "Income"
##  [4] "Family"             "CCAvg"              "Education.1"
##  [7] "Education.2"        "Education.3"        "Mortgage"
## [10] "Securities.Account" "CD.Account"         "Online"
## [13] "CreditCard"
```

```
y<-bank$Personal.Loan
x<-bank_dummy
table(y)
```

```
## y
##    0    1
## 4520  480
```

```
library(caret)
set.seed(123)
idx_train<-createDataPartition(y,p=0.60,list=FALSE)
```

```
x_train<-x[idx_train,]
x_valid<-x[-idx_train,]
y_train<-y[idx_train]
y_valid<-y[-idx_train]
dim(x_train)
```

```
## [1] 3000   13
```

```
dim(x_valid)
```

```
## [1] 2000   13
```

```
prop.table(table(y_train))
```

```
## y_train
##          0          1
## 0.90733333 0.09266667
```

```
prop.table(table(y_valid))
```

```
## y_valid
##     0     1
## 0.899 0.101
```

```
norm_model<-preProcess(x_train, method = "range")
x_train_n<-predict(norm_model, x_train)
x_valid_n<-predict(norm_model, x_valid)
```

```
ls()
```

```
##  [1] "bank"       "bank_dummy" "dummies"    "idx_train"  "norm_model"
##  [6] "x"          "x_train"    "x_train_n"  "x_valid"    "x_valid_n"
## [11] "y"          "y_train"    "y_valid"
```

```
new_customer<-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)
new_customer_n<-predict(norm_model,new_customer)
new_customer_n
```

```
##         Age Experience    Income    Family CCAvg Education.1 Education.2
## 1 0.3863636  0.2826087 0.3518519 0.3333333   0.2           0           1
##   Education.3 Mortgage Securities.Account CD.Account Online CreditCard
## 1           0        0                   0          0      1          1
```

```
library(class)
pred_k1<-knn(
  train = x_train_n,
  test = new_customer_n,
  cl=y_train,
  k=1,
  prob = TRUE
)
pred_k1
```

```
## [1] 0
## attr(,"prob")
## [1] 1
## Levels: 0 1
```

Question 1: Using k=1, the customer is classified as 0, meaning the loan is not accepted. Therefore, based on the nearest neighbor in the training data, the customer would not be predicted to accept the personal loan offer.

```r
library(class)
k_values<-seq(1,51,by=2)
val_acc<-numeric(length(k_values))
for(i in seq_along(k_values)){
  k<-k_values[i]
  pred<-knn(train = x_train_n,test = x_valid_n,cl=y_train,k=k)
  val_acc[i]<-mean(pred==y_valid)
}
best_k<-k_values[which.max(val_acc)]
best_k
```

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

```r
max(val_acc)
```

```
## [1] 0.962
```

```r
data.frame(k_values,val_acc)
```

```
##    k_values val_acc
## 1         1  0.9620
## 2         3  0.9600
## 3         5  0.9500
## 4         7  0.9420
## 5         9  0.9385
## 6        11  0.9340
## 7        13  0.9270
## 8        15  0.9250
## 9        17  0.9220
## 10       19  0.9225
## 11       21  0.9190
## 12       23  0.9165
## 13       25  0.9145
## 14       27  0.9150
## 15       29  0.9130
## 16       31  0.9125
## 17       33  0.9115
## 18       35  0.9115
## 19       37  0.9100
## 20       39  0.9085
## 21       41  0.9065
## 22       43  0.9060
## 23       45  0.9050
## 24       47  0.9035
## 25       49  0.9025
## 26       51  0.9015
```

Question 2: Although k=1 yields the highest validation accuracy (0.962), it's likely to overfit the training data. more balanced choice is k=3, which achieves nearly the same validation accuracy (0.960) while reducing model variance. Therefore, k=3 provides a better balance between overfitting and ignoring predictor information.

```
pred_k3<-knn(
  train = x_train_n,
  test = x_valid_n,
  cl=y_train,
  k=3
)
conf_matrix<-table(Predicted=pred_k3,Actual=y_valid)
conf_matrix
```

```
##          Actual
## Predicted    0    1
##         0 1792   74
##         1    6  128
```

Question 3: Using k=3, the confusion matric for the validation set shows 1792 true negatives, 128 true positives, 74 false negatives, and 6 false positives. The model correctly classifies the majority of customers, with strong performance in identifying both loan acceptances and non acceptances. The overall validation accuracy is 96%

```
pred_best<-knn(
  train = x_train_n,
  test = new_customer_n,
  cl=y_train,
  k=3,
  prob = TRUE
)
pred_best
```

```
## [1] 0
## attr(,"prob")
## [1] 1
## Levels: 0 1
```

Question 4: Using the selected value k=3, the customer is classified as 0. Therefore, based on the three nearest neighbors in the training data, this customer is predicted not to accept the personal loan offer.

```
set.seed(123)
idx_train2<-createDataPartition(y,p=0.50,list=FALSE)
x_train2<-x[idx_train2,]
y_train2<-y[idx_train2]
x_remaining<-x[-idx_train2,]
y_remaining<-y[-idx_train2]
idx_valid2<-createDataPartition(y_remaining,p=0.60,list=FALSE)
x_valid2<-x_remaining[idx_valid2,]
y_valid2<-y_remaining[idx_valid2]
x_test2<-x_remaining[-idx_valid2,]
y_test2<-y_remaining[-idx_valid2]
dim(x_train2)
```

```
## [1] 2500   13
```

```r
dim(x_valid2)
```

```
## [1] 1500   13
```

```r
dim(x_test2)
```

```
## [1] 1000   13
```

```r
norm_model2<-preProcess(x_train2,method = "range")
x_train2_n<-predict(norm_model2, x_train2)
x_valid2_n<-predict(norm_model2, x_valid2)
x_test2_n<-predict(norm_model2, x_test2)
```

```r
pred_train2 <- knn(train = x_train2_n, test = x_train2_n, cl = y_train2, k = 3)
pred_valid2 <- knn(train = x_train2_n, test = x_valid2_n, cl = y_train2, k = 3)
pred_test2  <- knn(train = x_train2_n, test = x_test2_n,  cl = y_train2, k = 3)

cm_train2 <- table(Predicted = pred_train2, Actual = y_train2)
cm_valid2 <- table(Predicted = pred_valid2, Actual = y_valid2)
cm_test2  <- table(Predicted = pred_test2,  Actual = y_test2)

cm_train2
```

```
##          Actual
## Predicted    0    1
##         0 2265   58
##         1    6  171
```

```r
cm_valid2
```

```
##          Actual
## Predicted    0    1
##         0 1349   57
##         1    8   86
```

```r
cm_test2
```

```
##          Actual
## Predicted   0   1
##         0 891  40
##         1   1  68
```

Question 5: Using k=3 with a 50% training, 30% validation, and 20% test split, the training accuracy is approximately 97.4%, while the validation and test accuracies are approximately 95.7% and 95.9%, respectively. The training accuracy is slightly higher because the model is evaluated on data it was trained on. However, the validation and test accuracies are very similar, indicating that the model generalizes well to unseen data and does not appear to be overfitting. The small differences across the three sets are expected due to sampling variability and the bias variance trade off.