Assign2 FML

2023-09-12

#Summary The acceptance of personal loans by customers was predicted by Universal Bank using k-NN classification and historical data. Data Partition, model training and validation, optimal "K" selection, and evaluating model performance on omitted data were all required to solve the issue. The outcomes assisted the bank in making defensible choices for a new marketing initiative designed to turn liability customers into personal loan customers. After data partitioning into training (60%) and validation (40%) sets, I did k-NN classification using all predictors (aside from ID and ZIP code) with k = 1, discovering that the consumer would not accept the Personal Loan. In order to continue with a new customer who had specific characteristics, I discovered that the ideal K would be 3. I used k-NN classification to predict the customer's loan acceptance status, and I discovered that they would not be accepting the Personal loan. Question as stated and Found Answers 1. This new client, who would be classed as 0, denies the personal loan, therefore how would we classify them? 2. What is a "K" selection that strikes a balance between overfitting and neglecting the information provided by the predictors:- According to the model we developed, the ideal "K" is 3. 3. Display the confusion matrix for the data generated by the best "K" for the validation:- The overall accuracy of the model is 0.964 when we run the model with the best "K" value of 3, and the confidence interval for the accuracy estimate is in the range (0.9549 to 0.9717). The no information rate is 0.8975. With a Kappa value of 0.7785, there is a significant degree of agreement that goes above what is expected by chance. When the model predicts class 1, it is accurate roughly 96.6% of the time (True Positive rate: 0.995, True Negative rate: 0.6927, Precision: 0.9659). 4. Sort the client2 using the top "K":- The customer2 denies the personal loan by using the best "K," which is 3 and is rated as "0." 5. Compare the confusion matrices of the training and validation sets with those of the test set. Describe the variations and their cause:- 1701 cases were properly predicted as being in class 0 by True Negatives (TN). 47 cases were accurately identified as belonging to class 1 in true positives (TP). False Positives (FP): 202 cases were misclassified as belonging to class 1 when, in fact, they do not. False Negatives (FN): 550 cases were misclassified as being in class 0, when in fact they are in class 1. With a 69.92% accuracy rate, the model was able to accurately predict class labels for around 70% of occurrences. The true accuracy is predicted to fall between 68.08% and 71.71% with a 95% confidence level. A negative Kappa score (-0.0343) indicates that the model performs worse than random chance. At 18.88%, it has a low sensitivity (ability to detect positive instances). At 75.57%, it has a moderate level of specificity (ability to recognize negative situations). Positive forecasts have a poor precision of 7.87%. Negative predictions have a strong 89.38% negative predictive value. The model's performance metrics are based on a data set where 9.96% of cases belong to the positive 1 class. Positive occurrences are detected at a rate of 1.88%. 23.88% of the dataset's cases had positive class labels predicted by the model. Taking into account both sensitivity and specificity, balanced accuracy stands at 47.22%. These statistics are provided for the class 1 (positive) category. Issue Statement Young bank Universal Bank is quickly expanding in terms of total client acquisition. The majority of these clients are liability clients (depositors), and their relationships with the bank range in size. The bank is interested in fast growing this client base in order to generate more loan business because the pool of asset customers (borrowers) is now fairly small. It wants to specifically investigate how to turn its liability consumers into personal loan customers. The bank executed a campaign for liability customers last year that had a successful conversion rate of over 9%. The retail marketing division has been inspired by this to create more intelligent campaigns with better target marketing. The objective is to anticipate a new customer's acceptance of a loan offer using k-NN. This will be the foundation for the creation of a fresh campaign. There are 5000 customers' worth of data in the file UniversalBank.csv. The information includes the customer's age, income, association with the bank (mortgage, securities account, etc.), and reaction to the previous personal loan campaign (Personal Loan). Only 480 (or 9.6%) of these 5000 clients accepted the personal loan that was presented to them during the previous campaign. Data sets should be divided into training (60%) and validation (40%).

```
UB <-read.csv("F://FML Asgn//FML Doc//UniversalBank.csv")
summary(UB)
```

```
##        ID             Age          Experience       Income          ZIP.Code    
##  Min.   :   1   Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   : 9307  
##  1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:91911  
##  Median :2500   Median :45.00   Median :20.0   Median : 64.00   Median :93437  
##  Mean   :2500   Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :93153  
##  3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:94608  
##  Max.   :5000   Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :96651  
##      Family           CCAvg          Education        Mortgage    
##  Min.   :1.000   Min.   : 0.000   Min.   :1.000   Min.   :  0.0  
##  1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0  
##  Median :2.000   Median : 1.500   Median :2.000   Median :  0.0  
##  Mean   :2.396   Mean   : 1.938   Mean   :1.881   Mean   : 56.5  
##  3rd Qu.:3.000   3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0  
##  Max.   :4.000   Max.   :10.000   Max.   :3.000   Max.   :635.0  
##  Personal.Loan   Securities.Account   CD.Account         Online      
##  Min.   :0.000   Min.   :0.0000     Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.000   1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :0.000   Median :0.0000     Median :0.0000   Median :1.0000  
##  Mean   :0.096   Mean   :0.1044     Mean   :0.0604   Mean   :0.5968  
##  3rd Qu.:0.000   3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000  
##  Max.   :1.000   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 Import and cleaning

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(e1071)
library(class)
```

#Read the data.

```
universal.df <- read.csv("F://FML Asgn//FML Doc//UniversalBank.csv")
dim(universal.df)
```

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

```
t(t(names(universal.df))) #The t function creates a transport of the data frame
```

```
##       [,1]
##  [1,] "ID"
##  [2,] "Age"
##  [3,] "Experience"
##  [4,] "Income"
##  [5,] "ZIP.Code"
##  [6,] "Family"
##  [7,] "CCAvg"
##  [8,] "Education"
##  [9,] "Mortgage"
## [10,] "Personal.Loan"
## [11,] "Securities.Account"
## [12,] "CD.Account"
## [13,] "Online"
## [14,] "CreditCard"
```

Drop ID and ZIP

```
universal.df <- universal.df[,-c(1,5)]
```

##Converting Categorical Variables

```
##only education needs to be converted to factor
universal.df$Education <- as.factor(universal.df$Education)
levels(universal.df$Education)
```

```
## [1] "1" "2" "3"
```

##Converting Categorical Variables to dummy

```
##Now,convert Education to dummy variables
groups <- dummyVars(~.,data = universal.df) ##This creates the dummy groups
universal_m.df <-  as.data.frame(predict(groups,universal.df))
```

##Split the data

```
set.seed(1) #Important to ensure that we get the same sample if we run the code
training.dif = sample(row.names(universal_m.df),0.6*dim(universal_m.df)[1])
validation.dif = setdiff(row.names(universal_m.df),training.dif)
train.diff = universal_m.df[training.dif,]
valid.diff = universal_m.df[validation.dif,]
t(t(names(train.diff)))
```

```
##       [,1]
##  [1,] "Age"
##  [2,] "Experience"
##  [3,] "Income"
##  [4,] "Family"
##  [5,] "CCAvg"
##  [6,] "Education.1"
##  [7,] "Education.2"
##  [8,] "Education.3"
##  [9,] "Mortgage"
## [10,] "Personal.Loan"
## [11,] "Securities.Account"
## [12,] "CD.Account"
## [13,] "Online"
## [14,] "CreditCard"
```

```
#Now let us normalize the data
train.normal.diff <- train.diff[,-10] # Note that Personal Income is the 10th variable
valid.normal.diff <- valid.diff[,-10]

normal.values <- preProcess(train.diff[, -10], method=c("center", "scale"))
train.normal.diff <- predict(normal.values, train.diff[, -10])
valid.normal.diff <- predict(normal.values, valid.diff[, -10])
```

Solution for the given Problem

#Q1:Consider the following customer:

1. 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?

```
# We have converted all categorical variables to dummy variables
# Let's create a new sample
New_Customer1 <- 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
)

# Normalize the new customer
New.Cust.normal <- New_Customer1
New.Cust.normal <- predict(normal.values, New.Cust.normal)
New.Cust.normal
```

```
##          Age Experience    Income     Family     CCAvg Education.1 Education.2
## 1 -0.4774216 -0.8953121 0.2389084 -0.3368482 0.0492415  -0.8461728    1.583646
##   Education.3   Mortgage Securities.Account CD.Account    Online CreditCard
## 1  -0.6509102 -0.5679457         -0.3338946 -0.2380992 0.8426977   1.554365
```

#Now, let us predict using KNN

```
KNN.Predct1 <- class::knn(train = train.normal.diff,
                       test = New.Cust.normal,
                       cl = train.diff$Personal.Loan, k = 1)
KNN.Predct1
```

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

Q2. What is a choice of K that balances between over-fitting and ignoring the predictor information?

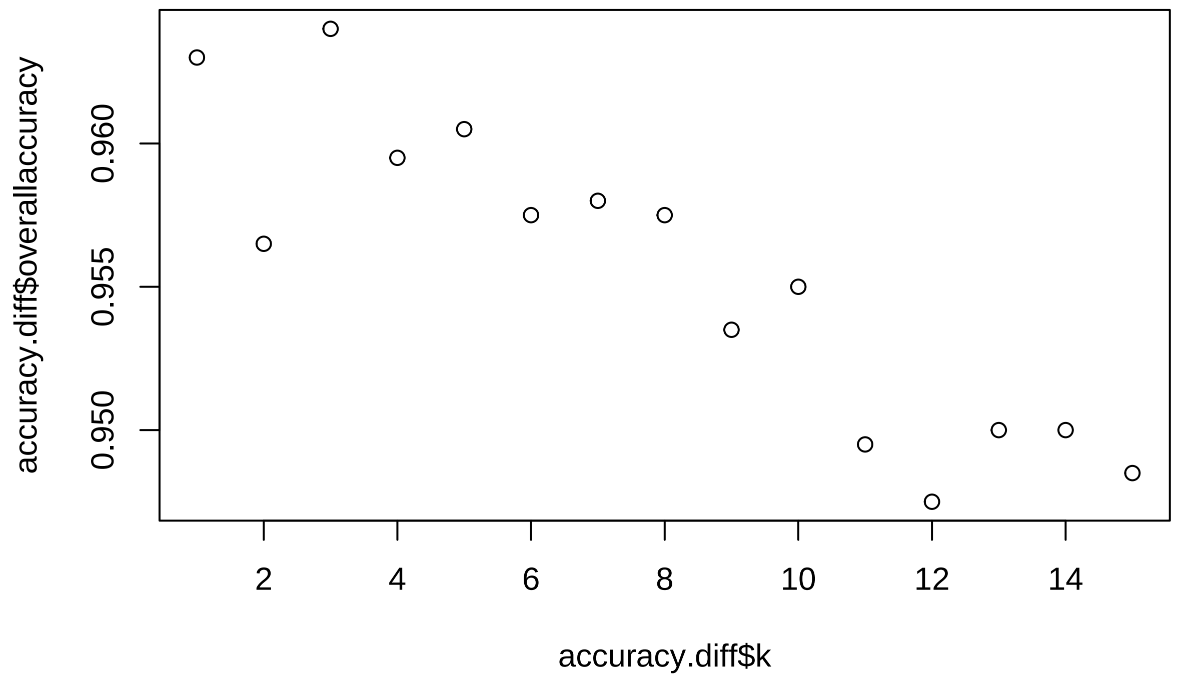
```
#Calculate the accuracy for each value of k
#Set the range of k values to consider

accuracy.diff <- data.frame(k = seq(1, 15, 1), overallaccuracy = rep(0, 15))
for(i in 1:15) {
  KNN.Predct <- class::knn(train = train.normal.diff,
                         test = valid.normal.diff,
                         cl = train.diff$Personal.Loan, k = i)
  accuracy.diff[i, 2] <- confusionMatrix(KNN.Predct,
                                    as.factor(valid.diff$Personal.Loan),positive = "1")$overall[1]
}

which(accuracy.diff[,2] == max(accuracy.diff[,2]))
```

```
## [1] 3
```

```
plot(accuracy.diff$k,accuracy.diff$overallaccuracy)
```



#Q 3. Show the confusion matrix for the validation data that results from using the best k

```
KNN.Predct2 <- class::knn(train = train.normal.diff,
                          test = valid.normal.diff,
                          cl = train.diff$Personal.Loan, k = 3)

confusionMatrix(KNN.Predct2,as.factor(valid.diff$Personal.Loan))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1786   63
##          1    9  142
##
##                Accuracy : 0.964
##                  95% CI : (0.9549, 0.9717)
##     No Information Rate : 0.8975
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7785
##
##  Mcnemar's Test P-Value : 4.208e-10
##
##             Sensitivity : 0.9950
##             Specificity : 0.6927
##          Pos Pred Value : 0.9659
##          Neg Pred Value : 0.9404
##              Prevalence : 0.8975
##          Detection Rate : 0.8930
##    Detection Prevalence : 0.9245
##       Balanced Accuracy : 0.8438
##
##        'Positive' Class : 0
##
```

Q4. 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

```
#Classifying the customer using the best K.

New_Customer2 = 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
)

# Normalize the New_Customer2

New.Cust.normal2 <- New_Customer2
New.Cust.normal2 <- predict(normal.values, New.Cust.normal2)

KNN.Predct3 <- class::knn(train = train.normal.diff,
                         test = New.Cust.normal2,
                         cl = train.diff$Personal.Loan, k = 3)

KNN.Predct3
```

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

```
#The customer has been classified as will not accept the the personal loan

print("As the out put is 0, The Customer will not accept the Personal Loan offer")
```

```
## [1] "As the out put is 0, The Customer will not accept the Personal Loan offer"
```

#Q5

```r
set.seed(2)
#Let's take 50% of the entire modified data as Training data
train.diff2 = sample(row.names(universal_m.df), 0.5*dim(universal_m.df)[1])

#Let's take 30% of the data from the remaining 50% as Validation Data
valid.diff2 = sample(setdiff(row.names(universal_m.df), train.diff2), 0.3*dim(universal_m.df)[1])

#Let's take remaining 20% of the modified data as Test Data
test.diff2 = setdiff(row.names(universal_m.df), union(train.diff2,valid.diff2))

train.normal.diff2 = universal_m.df[train.diff2,]
valid.normal.diff2 = universal_m.df[valid.diff2,]
test.normal.diff2 = universal_m.df[test.diff2,]

#transporting the data
t(t(names(train.normal.diff2)))
```

```
##       [,1]
##  [1,] "Age"
##  [2,] "Experience"
##  [3,] "Income"
##  [4,] "Family"
##  [5,] "CCAvg"
##  [6,] "Education.1"
##  [7,] "Education.2"
##  [8,] "Education.3"
##  [9,] "Mortgage"
## [10,] "Personal.Loan"
## [11,] "Securities.Account"
## [12,] "CD.Account"
## [13,] "Online"
## [14,] "CreditCard"
```

```r
# Applying the k-NN method with the chosen K.

trainknn2 = knn(train = train.normal.diff2[,-8], test = train.normal.diff2[,-8], cl = train.normal.diff

validknn2 = knn(train = train.normal.diff2[,-8], test = valid.normal.diff2[,-8], cl = train.normal.diff

testknn2 = knn(train = train.normal.diff2[,-8], test = test.normal.diff2[,-8], cl = train.normal.diff2[
```

Comparing the confusion matrix of the training set, validation set and test set

```r
Confusionmatrix_trainknn2 = confusionMatrix(trainknn2, as.factor(train.normal.diff2$Personal.Loan),posi

Confusionmatrix_trainknn2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1702  202
##          1  549   47
##
##                Accuracy : 0.6996
##                  95% CI : (0.6812, 0.7175)
##     No Information Rate : 0.9004
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.034
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.18876
##             Specificity : 0.75611
##          Pos Pred Value : 0.07886
##          Neg Pred Value : 0.89391
##              Prevalence : 0.09960
##          Detection Rate : 0.01880
##    Detection Prevalence : 0.23840
##       Balanced Accuracy : 0.47243
##
##        'Positive' Class : 1
##
```

```
Confusionmatrix_validknn2 = confusionMatrix(validknn2, as.factor(valid.normal.diff2$Personal.Loan),posi

Confusionmatrix_trainknn2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1702  202
##          1  549   47
##
##                Accuracy : 0.6996
##                  95% CI : (0.6812, 0.7175)
##     No Information Rate : 0.9004
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.034
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.18876
##             Specificity : 0.75611
##          Pos Pred Value : 0.07886
##          Neg Pred Value : 0.89391
##              Prevalence : 0.09960
##          Detection Rate : 0.01880
##    Detection Prevalence : 0.23840
##       Balanced Accuracy : 0.47243
##
##        'Positive' Class : 1
##
```

```
Confusionmatrix_testknn2 = confusionMatrix(testknn2, as.factor(test.normal.diff2$Personal.Loan),positiv

Confusionmatrix_trainknn2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1702  202
##          1  549   47
##
##                Accuracy : 0.6996
##                  95% CI : (0.6812, 0.7175)
##     No Information Rate : 0.9004
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.034
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.18876
##             Specificity : 0.75611
##          Pos Pred Value : 0.07886
##          Neg Pred Value : 0.89391
##              Prevalence : 0.09960
##          Detection Rate : 0.01880
##    Detection Prevalence : 0.23840
##       Balanced Accuracy : 0.47243
##
##        'Positive' Class : 1
##
```