Assignment 2

Kevin Gardner

2/15/2022

# ————————————————

# Following is the link to my GitHub account:

# <https://github.com/Kgardner22/64060_-kgardner>

# ————————————————

Import and Prepare Data:

Import the UniversalBank.csv file

UniversalBank <- read.table('C:/R/MyData/UniversalBank.csv', header = T, sep = ',')   
  
summary(UniversalBank)

## 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

Create a copy of the original data file to preserve

Original\_File <- UniversalBank

REQUIREMENT 1:

Transform categorical predictors with more than two categories into dummy variables FIRST. Need to do this for ‘Education’ and ‘Personal.Loan’.

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(class)  
  
# Remove unnecessary attributes  
UniversalBank$ID <- NULL  
UniversalBank$ZIP.Code <- NULL  
  
# Transform to factors  
UniversalBank$Education=as.factor(UniversalBank$Education)  
UniversalBank$Personal.Loan=as.factor(UniversalBank$Personal.Loan)  
  
# Use dummyVars function to create a model  
dummies <- dummyVars(Personal.Loan ~ ., data = UniversalBank)  
UniversalBank\_dummy <- as.data.frame(predict(dummies, newdata = UniversalBank))

## Warning in model.frame.default(Terms, newdata, na.action = na.action, xlev =  
## object$lvls): variable 'Personal.Loan' is not a factor

Normalize the data (I’m using z-score scaling as input method)

Norm\_model <- preProcess(UniversalBank\_dummy, method = c("center", "scale"))  
UniversalBank\_norm <- predict(Norm\_model, UniversalBank\_dummy)  
summary(UniversalBank\_norm)

## Age Experience Income Family   
## Min. :-1.94871 Min. :-2.014710 Min. :-1.4288 Min. :-1.2167   
## 1st Qu.:-0.90188 1st Qu.:-0.881116 1st Qu.:-0.7554 1st Qu.:-1.2167   
## Median :-0.02952 Median :-0.009121 Median :-0.2123 Median :-0.3454   
## Mean : 0.00000 Mean : 0.000000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.84284 3rd Qu.: 0.862874 3rd Qu.: 0.5263 3rd Qu.: 0.5259   
## Max. : 1.88967 Max. : 1.996468 Max. : 3.2634 Max. : 1.3973   
## CCAvg Education.1 Education.2 Education.3   
## Min. :-1.1089 Min. :-0.8495 Min. :-0.6245 Min. :-0.6549   
## 1st Qu.:-0.7083 1st Qu.:-0.8495 1st Qu.:-0.6245 1st Qu.:-0.6549   
## Median :-0.2506 Median :-0.8495 Median :-0.6245 Median :-0.6549   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.3216 3rd Qu.: 1.1770 3rd Qu.: 1.6010 3rd Qu.: 1.5266   
## Max. : 4.6131 Max. : 1.1770 Max. : 1.6010 Max. : 1.5266   
## Mortgage Securities.Account CD.Account Online   
## Min. :-0.5555 Min. :-0.3414 Min. :-0.2535 Min. :-1.2165   
## 1st Qu.:-0.5555 1st Qu.:-0.3414 1st Qu.:-0.2535 1st Qu.:-1.2165   
## Median :-0.5555 Median :-0.3414 Median :-0.2535 Median : 0.8219   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.4375 3rd Qu.:-0.3414 3rd Qu.:-0.2535 3rd Qu.: 0.8219   
## Max. : 5.6875 Max. : 2.9286 Max. : 3.9438 Max. : 0.8219   
## CreditCard   
## Min. :-0.6452   
## 1st Qu.:-0.6452   
## Median :-0.6452   
## Mean : 0.0000   
## 3rd Qu.: 1.5495   
## Max. : 1.5495

Add back in the target variable (Personal.Loan)

UniversalBank\_norm$Personal.Loan <- UniversalBank$Personal.Loan

We need to divide the data into training (60%) and validation (40%) sets

Train\_Index <- createDataPartition(UniversalBank$Personal.Loan, p=0.6, list = FALSE)  
Train.df <- UniversalBank\_norm[Train\_Index,]  
Validation.df <- UniversalBank\_norm[-Train\_Index,]

Create data frame with values to predict 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.

To\_Predict <- 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)  
  
print(To\_Predict)

## Age Experience Income Family CCAvg Education.1 Education.2 Education.3  
## 1 40 10 84 2 2 0 1 0  
## Mortgage Securities.Account CD.Account Online CreditCard  
## 1 0 0 0 1 1

Normalize this new record (To\_Predict) using the same model we applied to the original dataset

To\_Predict\_norm <- predict(Norm\_model, To\_Predict)  
print(To\_Predict\_norm)

## Age Experience Income Family CCAvg Education.1 Education.2  
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 -0.8494814 1.601024  
## Education.3 Mortgage Securities.Account CD.Account Online CreditCard  
## 1 -0.6548999 -0.5554684 -0.3413892 -0.2535149 0.8218687 1.549477

Use k-NN function to make the prediction.

Perform a k-NN classification with all predictors EXCEPT ID and Zip\_Code using k=1. Specify success class as 1 (loan acceptance) and use the default cutoff value of 0.5.

Prediction <- knn(train = Train.df[,1:13],  
 test = To\_Predict\_norm[,1:13],  
 cl=Train.df$Personal.Loan,  
 k=1)  
print(Prediction)

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

ANSWER - REQUIREMENT 1:

As shown in the above results (with k=1), the prediction for this observation is that Personal.Loan = 0 meaning, this individual is predicted to NOT accept the personal loan being offered.

REQUIREMENT 2:

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

set.seed(123)  
  
fitControl <- trainControl(method = "repeatedcv", number = 3, repeats = 2)  
  
searchGrid <- expand.grid(k = 1:15)  
  
Knn.model <- train(Personal.Loan ~ .,  
 data = Train.df,  
 method = 'knn',  
 tuneGrid = searchGrid,  
 trControl = fitControl,)  
  
Knn.model

## k-Nearest Neighbors   
##   
## 3000 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (3 fold, repeated 2 times)   
## Summary of sample sizes: 2000, 2000, 2000, 2000, 2000, 2000, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 1 0.9605000 0.7478112  
## 2 0.9548333 0.7089392  
## 3 0.9576667 0.7112729  
## 4 0.9538333 0.6774099  
## 5 0.9556667 0.6864548  
## 6 0.9546667 0.6786735  
## 7 0.9516667 0.6488616  
## 8 0.9505000 0.6400390  
## 9 0.9496667 0.6282323  
## 10 0.9471667 0.6091686  
## 11 0.9461667 0.5956549  
## 12 0.9453333 0.5877380  
## 13 0.9446667 0.5810143  
## 14 0.9438333 0.5732348  
## 15 0.9436667 0.5706980  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 1.

ANSWER - REQUIREMENT 2:

The above Knn.model indicates that the k value with the highest accuracy is a value of k=3

REQUIREMENT 3:

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

Use the predict function of the caret package to make predictions on the validation set.

predictions <- predict(Knn.model, Validation.df)

Compare predictions from the Knn.model to the actual Personal.Loan labels in the validation set to compute the confusion matrix

confusionMatrix(predictions, Validation.df$Personal.Loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1783 69  
## 1 25 123  
##   
## Accuracy : 0.953   
## 95% CI : (0.9428, 0.9619)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : 2.260e-16   
##   
## Kappa : 0.6983   
##   
## Mcnemar's Test P-Value : 9.202e-06   
##   
## Sensitivity : 0.9862   
## Specificity : 0.6406   
## Pos Pred Value : 0.9627   
## Neg Pred Value : 0.8311   
## Prevalence : 0.9040   
## Detection Rate : 0.8915   
## Detection Prevalence : 0.9260   
## Balanced Accuracy : 0.8134   
##   
## 'Positive' Class : 0   
##

REQUIREMENT 4:

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.

# The independent variables specified in this requirement are the same as those specified in requirement 1. Therefore, this data.frame is already build and normalized (To\_Predict\_norm).  
  
print(To\_Predict)

## Age Experience Income Family CCAvg Education.1 Education.2 Education.3  
## 1 40 10 84 2 2 0 1 0  
## Mortgage Securities.Account CD.Account Online CreditCard  
## 1 0 0 0 1 1

Using the normalized prediction file (To\_Predict\_norm), we will use the Knn.model to predict using the best k value (k=3)

predict(Knn.model, To\_Predict\_norm)

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

ANSWER - REQUIREMENT 4:

Using the best k value, the above results (with k=3) is predicting this observation will have Personal.Loan = 0 meaning, this individual is predicted to NOT accept the personal loan being offered.