mbruner3_2

Mark Bruner

9/22/2020

Libraries needed for this assignment.

```
library(ggplot2)
library(caret)
```

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

```
library(ISLR)
library(readr)
```

Imported dataset.

```
unibank.df <- read_csv("UniversalBank.csv")
```

```
## Parsed with column specification:
## cols(
##   ID = col_double(),
##   Age = col_double(),
##   Experience = col_double(),
##   Income = col_double(),
##   `ZIP Code` = col_double(),
##   Family = col_double(),
##   CCAvg = col_double(),
##   Education = col_double(),
##   Mortgage = col_double(),
##   `Personal Loan` = col_double(),
##   `Securities Account` = col_double(),
##   `CD Account` = col_double(),
##   Online = col_double(),
##   CreditCard = col_double()
## )
```

Cleaning dataset.

Removed Zip Code and ID columns. Also, made dummy categories for Education and removed the Education column. Lastly, converted dataset into a data frame.

```
unibank.df <- unibank.df[, -1]
unibank.df <- unibank.df[, -4]
unibank.df$Education_1 <- ifelse(unibank.df$Education == '1', 1, 0)
unibank.df$Education_2 <- ifelse(unibank.df$Education == '2', 1, 0)
unibank.df$Education_3 <- ifelse(unibank.df$Education == '3', 1, 0)
unibank.df <- unibank.df[, -6]
unibank.df <- data.frame(unibank.df)
head(unibank.df)
```

```
##   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
```

Partitions

Created partitions for the dataset into Train & Validation set.

```
set.seed(123)
Train_index <- createDataPartition(unibank.df$Personal.Loan, p = .6, list = FALSE)
Train_data = unibank.df[Train_index, ]
Valid_data = unibank.df[-Train_index, ]
```

Customer Data Frame.

Created a data frame for the customer.

```
customer.df <- data.frame("Age" = as.integer(40), "Experience" = as.integer(10), Income = as.integer(84
```

Copy original data

```
Train_norm <- Train_data
Valid_norm <- Valid_data
```

Normalization

Used the training data except for the personal loan column to normalize the rest of the train, test, and validation set.

```
normalized_values <- preProcess(Train_data[, -7], method = "center", "scale")
Train_norm[, -7] <- predict(normalized_values, Train_data[, -7])
Valid_norm[, -7] <- predict(normalized_values, Valid_data[, -7])
```

kNN Modeling

```
library(FNN)
nn <- knn(train = Train_norm[, -7], test = Valid_norm[, -7], cl = Train_norm[, 7], k = 1, prob = TRUE)

valid_levels <- factor(Valid_norm[, 7], levels = c("0", "1"))
```

#Created Accuracy Data Frame to hold "K" Values.

```
accuracyTest.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
```

Hypertuning using Validation

Used an accuracy test for k{i} = 1:14 and a confusion matrix.

```
for(i in 1:14) {
  knn.pred <- knn(train = Train_norm[, -7], test = Valid_norm[, -7],
                  cl = Train_norm[, 7], k = i, prob = TRUE)
  accuracyTest.df[i, 2] <- confusionMatrix(knn.pred, valid_levels)$overall[1]
}
confusionMatrix(knn.pred, valid_levels)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1774  165
##          1   24   37
##
##                Accuracy : 0.9055
##                  95% CI : (0.8918, 0.918)
##     No Information Rate : 0.899
##     P-Value [Acc > NIR] : 0.1771
```

```
##
##                   Kappa : 0.246
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9867
##             Specificity : 0.1832
##          Pos Pred Value : 0.9149
##          Neg Pred Value : 0.6066
##              Prevalence : 0.8990
##          Detection Rate : 0.8870
##    Detection Prevalence : 0.9695
##       Balanced Accuracy : 0.5849
##
##        'Positive' Class : 0
##
```

```
accuracyTest.df
```

```
##     k accuracy
## 1   1   0.9040
## 2   2   0.9045
## 3   3   0.9035
## 4   4   0.9025
## 5   5   0.9075
## 6   6   0.9015
## 7   7   0.9005
## 8   8   0.9030
## 9   9   0.9020
## 10 10   0.8990
## 11 11   0.9020
## 12 12   0.9040
## 13 13   0.9030
## 14 14   0.9055
```

k = 5 is the value that provide the best performance.

Prediction

Normalized all of Universal Bank Dataset.

```
normalized_values <- preProcess(unibank.df[, -7], method = c("center", "scale"))

unibank.df[, -7] <- predict(normalized_values, unibank.df[, -7])
```

#Prediction of Customer

```
knn.pred.new <- knn(unibank.df[, -7], customer.df, cl = unibank.df[, 7], k = 5, prob = TRUE)
knn.attr <- attributes(knn.pred.new)
knn.attr[1]
```

```
## $levels
## [1] "1"
```

```
knn.attr[3]
```

```
## $prob
## [1] 0.8
```

If I am interpreting these results correctly then I would have to say that the customer would accept the personal loan.