HW2

David Slusser

10/4/2020

R Markdown

This is the second homework assignment Use the bank data to create a k-NN Classification Load the packages that I will need and load in data set

```
library(readr)
library(caret)
```

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

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

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 4.0.2
```

```
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.0.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
bank <- read_csv('/Users/davidslusser/Desktop/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()
## )
```

```
## Now that data is loaded, we need to transform the education variable into a set of 3
 dummies
## Education is 1, 2, or 3
## So we want education_1 = 1 if education = 1, 0 o.w.
## Create a loan acceptance variable, which is just a transformation of personal loan wh
ere 1 = accepted, 0 o.w

bank$education_1 <- ifelse(bank$Education == 1, 1, 0)
bank$education_2 <- ifelse(bank$Education == 2, 1, 0)
bank$education_3 <- ifelse(bank$Education == 3, 1, 0)
bank$acceptance <- ifelse(bank$`Personal Loan` == 1, 1, 0)
bank.1 <- bank %>%
  select(Age, Experience, Income, Family, CCAvg, Mortgage, `Securities Account`, `CD Acc
ount`, Online, CreditCard, education_1, education_2, education_3, acceptance) # Select a
subset of variables

## Now that the variables are created, we need to create a k-NN by splitting the data in
to training (60%) and
## validation (40%). Keep default value at 0.5

set.seed(1234)
train_index1 = createDataPartition(bank.1$acceptance, p = 0.6, list = FALSE)
train_data1 = bank.1[train_index1,] # create the training data; we include all columns;
 note the index is row,                                          column
```

```
## Warning: The `i` argument of ``[`()` can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
validation_data1 = bank.1[-train_index1,] # create the test set
summary(train_data1$acceptance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.09467 0.00000 1.00000
```

```
summary(validation_data1$acceptance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   0.098   0.000   1.000
```

```
## We need to normalize the data
## Use preProcess function

train1.norm.df <- train_data1
valid1.norm.df <- validation_data1
traval1.norm.df <- bank.1

norm.values <- preProcess(train_data1[, 1:6], method=c("center", "scale"))
train1.norm.df[, 1:6] <- predict(norm.values, train_data1[, 1:6]) # Replace first two co
lumns with normalized
values
valid1.norm.df[, 1:6] <- predict(norm.values, validation_data1[, 1:6])
traval1.norm.df[, 1:6] <- predict(norm.values, traval1.norm.df[, 1:6])
summary(train1.norm.df)
```

```
##       Age              Experience          Income            Family      
##  Min.   :-1.93856   Min.   :-2.00902   Min.   :-1.4229   Min.   :-1.2028  
##  1st Qu.:-0.89828   1st Qu.:-0.88070   1st Qu.:-0.7723   1st Qu.:-1.2028  
##  Median :-0.03138   Median :-0.01276   Median :-0.2302   Median :-0.3385  
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 0.92221   3rd Qu.: 0.85518   3rd Qu.: 0.5505   3rd Qu.: 0.5258  
##  Max.   : 1.87579   Max.   : 1.98350   Max.   : 3.2612   Max.   : 1.3901  
##      CCAvg             Mortgage       Securities Account   CD Account      
##  Min.   :-1.0986   Min.   :-0.5623   Min.   :0.0000     Min.   :0.000  
##  1st Qu.:-0.6993   1st Qu.:-0.5623   1st Qu.:0.0000     1st Qu.:0.000  
##  Median :-0.2429   Median :-0.5623   Median :0.0000     Median :0.000  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   :0.1087     Mean   :0.062  
##  3rd Qu.: 0.3277   3rd Qu.: 0.4445   3rd Qu.:0.0000     3rd Qu.:0.000  
##  Max.   : 4.6066   Max.   : 5.5278   Max.   :1.0000     Max.   :1.000  
##      Online         CreditCard      education_1      education_2     
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000  
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.000  
##  Mean   :0.5867   Mean   :0.2883   Mean   :0.4183   Mean   :0.286  
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.000  
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000  
##   education_3       acceptance      
##  Min.   :0.0000   Min.   :0.00000  
##  1st Qu.:0.0000   1st Qu.:0.00000  
##  Median :0.0000   Median :0.00000  
##  Mean   :0.2957   Mean   :0.09467  
##  3rd Qu.:1.0000   3rd Qu.:0.00000  
##  Max.   :1.0000   Max.   :1.00000
```

```
var(train1.norm.df[, 1:6])
```

```
##                    Age  Experience      Income      Family       CCAvg
## Age         1.00000000  0.99445066 -0.07890015 -0.04865101 -0.06484148
## Experience  0.99445066  1.00000000 -0.06994198 -0.05384665 -0.06253098
## Income     -0.07890015 -0.06994198  1.00000000 -0.17128308  0.64420088
## Family     -0.04865101 -0.05384665 -0.17128308  1.00000000 -0.11983245
## CCAvg      -0.06484148 -0.06253098  0.64420088 -0.11983245  1.00000000
## Mortgage   -0.01257661 -0.01078686  0.19704560 -0.02942549  0.08531845
##                Mortgage
## Age        -0.01257661
## Experience -0.01078686
## Income      0.19704560
## Family     -0.02942549
## CCAvg       0.08531845
## Mortgage    1.00000000
```

```
summary(valid1.norm.df)
```

```
##       Age              Experience          Income             Family      
##  Min.   :-1.938555   Min.   :-2.00902   Min.   :-1.422924   Min.   :-1.20280
##  1st Qu.:-0.833261   1st Qu.:-0.88070   1st Qu.:-0.750660   1st Qu.:-1.20280
##  Median :-0.031382   Median :-0.01276   Median :-0.208510   Median :-0.33851
##  Mean   :-0.005115   Mean   :-0.00920   Mean   : 0.008631   Mean   : 0.01023
##  3rd Qu.: 0.835515   3rd Qu.: 0.76839   3rd Qu.: 0.528812   3rd Qu.: 0.52577
##  Max.   : 1.875792   Max.   : 1.98350   Max.   : 3.131127   Max.   : 1.39006
##      CCAvg             Mortgage        Securities Account   CD Account      
##  Min.   :-1.09863   Min.   :-0.56228   Min.   :0.000      Min.   :0.000     
##  1st Qu.:-0.69927   1st Qu.:-0.56228   1st Qu.:0.000      1st Qu.:0.000     
##  Median :-0.18580   Median :-0.56228   Median :0.000      Median :0.000     
##  Mean   : 0.01749   Mean   :-0.01154   Mean   :0.098      Mean   :0.058     
##  3rd Qu.: 0.38472   3rd Qu.: 0.40502   3rd Qu.:0.000      3rd Qu.:0.000     
##  Max.   : 4.60656   Max.   : 5.70545   Max.   :1.000      Max.   :1.000     
##      Online        CreditCard      education_1      education_2     
##  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 :1.000   Median :0.0000   Median :0.0000   Median :0.0000  
##  Mean   :0.612   Mean   :0.3025   Mean   :0.4205   Mean   :0.2725  
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
##   education_3      acceptance     
##  Min.   :0.000   Min.   :0.000  
##  1st Qu.:0.000   1st Qu.:0.000  
##  Median :0.000   Median :0.000  
##  Mean   :0.307   Mean   :0.098  
##  3rd Qu.:1.000   3rd Qu.:0.000  
##  Max.   :1.000   Max.   :1.000
```

```
var(valid1.norm.df[, 1:6])
```

```
##                    Age  Experience      Income       Family       CCAvg
## Age         0.96926598  0.96727353 -0.01871568 -0.041398584 -0.03154457
## Experience  0.96727353  0.97724627 -0.01075193 -0.048937341 -0.03037296
## Income     -0.01871568 -0.01075193  0.99188836 -0.133098451  0.64140092
## Family     -0.04139858 -0.04893734 -0.13309845  0.960126013 -0.09059215
## CCAvg      -0.03154457 -0.03037296  0.64140092 -0.090592151  0.98569828
## Mortgage   -0.01245029 -0.01032379  0.22277050 -0.006698298  0.14724702
##                Mortgage
## Age        -0.012450288
## Experience -0.010323794
## Income      0.222770499
## Family     -0.006698298
## CCAvg       0.147247021
## Mortgage    1.020263663
```

```
## Now train the k-NN from the carret package
## By setting the random seed, we can reproduce the results

set.seed(1234)
model1 <- train(as.factor(acceptance) ~ Age + Experience + Income + Family + CCAvg + edu
cation_1 + education_2
                + education_3 + Mortgage + `Securities Account` + `CD Account` + Online
 + CreditCard,
                data = train1.norm.df, method = "knn")
model1
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   13 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3000, 3000, 3000, 3000, 3000, 3000, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9545240  0.6852693
##   7  0.9529855  0.6641167
##   9  0.9519810  0.6492145
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```
## Test the model onto a new data frame with a given customer
customer.df = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = as.
integer(84),
                         Family = as.integer(2), CCAvg = as.integer(2), Mortgage = as.in
teger(0),
                         `Securities Account` = as.integer(0), `CD Account` = as.integer
(0),
                         Online = as.integer(1), CreditCard = as.integer(1), education_1
= as.integer(0),
                         education_2 = as.integer(1), education_3 = as.integer(0))

customer.df[, 1:6] <- predict(norm.values, customer.df[, 1:6])

model1.1 <- knn(train = train1.norm.df[, 1:13, drop = FALSE], test = customer.df[, 1:13,
drop = FALSE],
                cl = train1.norm.df$acceptance,
                k = 1) # Instructions say to use k = 1

model1.1
```

```
## [1] 0
## attr(,"nn.index")
##      [,1]
## [1,]  977
## attr(,"nn.dist")
##           [,1]
## [1,] 0.7038998
## Levels: 0
```

According to the model, the consumer would be in the 0 level, which is the not accepted loan

```
## Look for which k is best balancing of overfitting and underfitting
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))

for(i in 1:14) { # Want to loop through
  knn.1 <- knn(train = train1.norm.df[, 1:13, drop = FALSE], test = valid1.norm.df[, 1:1
3, drop = FALSE],
               cl = as.factor(train1.norm.df$acceptance), k = i)
  accuracy.df[i, 2] <- confusionMatrix(knn.1, as.factor(valid1.norm.df$acceptance))$over
all[1] # Get the
                                                                                # conf
usion matrix as well
}
accuracy.df
```

```
##     k accuracy
## 1   1   0.9585
## 2   2   0.9525
## 3   3   0.9610
## 4   4   0.9505
## 5   5   0.9565
## 6   6   0.9525
## 7   7   0.9570
## 8   8   0.9495
## 9   9   0.9525
## 10 10   0.9480
## 11 11   0.9515
## 12 12   0.9450
## 13 13   0.9485
## 14 14   0.9460
```

We find the best choice to be k = 3, which we will apply to the test set because that has the highest accuracy

```
## Get the confusion matrix for when k = 3
knn.3 <- knn(train = train1.norm.df[, 1:13, drop = FALSE], test = valid1.norm.df[, 1:13,
drop = FALSE],
             cl = as.factor(train1.norm.df$acceptance), k = 3, prob=TRUE)
confusionMatrix(knn.3, as.factor(valid1.norm.df$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1797   71
##          1    7  125
##
##                Accuracy : 0.961
##                  95% CI : (0.9516, 0.9691)
##     No Information Rate : 0.902
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7418
##
##  Mcnemar's Test P-Value : 9.796e-13
##
##             Sensitivity : 0.9961
##             Specificity : 0.6378
##          Pos Pred Value : 0.9620
##          Neg Pred Value : 0.9470
##              Prevalence : 0.9020
##          Detection Rate : 0.8985
##    Detection Prevalence : 0.9340
##       Balanced Accuracy : 0.8169
##
##        'Positive' Class : 0
##
```

The confusion matrix shows that the model is 96.0% accurate. This means that we predict correctly if the person accepts the personal loan 96% of the time. The model is 99.56% sensative, meaning we find a true acceptance of the loan 99.56% of the time. The precision was only 64.18%, meaning only 64.18% of what we predicted of who would accept the loan actually accepted the loan.

```
# Test the model onto a new data frame with a given customer
customer.df1 = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = a
s.integer(84),
                          Family = as.integer(2), CCAvg = as.integer(2), Mortgage = as.in
teger(0),
                          `Securities Account` = as.integer(0), `CD Account` = as.integer
(0),
                          Online = as.integer(1), CreditCard = as.integer(1), education_1
= as.integer(0),
                          education_2 = as.integer(1), education_3 = as.integer(0))

customer.df1[, 1:6] <- predict(norm.values, customer.df1[, 1:6])

model1.2 <- knn(train = train1.norm.df[, 1:13, drop = FALSE], test = customer.df1[, 1:13
, drop = FALSE],
                cl = as.factor(train1.norm.df$acceptance),
                k = 3, prob = TRUE) # 3 is what we found to be the best k

model1.2
```

```
## [1] 0
## attr(,"prob")
## [1] 1
## attr(,"nn.index")
##      [,1] [,2] [,3]
## [1,]  977 2463 1610
## attr(,"nn.dist")
##           [,1]      [,2]     [,3]
## [1,] 0.7038998 0.8323018 1.027151
## Levels: 0
```

We find that the customer is still in level 0: the loan will not be accepted

```
## We now want to do 50% for training, 30% for validation and 20% for testing
## Resplit the data
m_bank <- bank %>%
  select(Age, Experience, Income, Family, CCAvg, Mortgage, `Securities Account`, `CD Acc
ount`, Online, CreditCard, education_1, education_2, education_3, acceptance) # Select a
subset of variables

set.seed(1234)
test_index2 = createDataPartition(m_bank$acceptance, p = 0.2, list = FALSE) # 20% reserv
ed for Test
test_data2 = m_bank[test_index2,]
traval_data2 = m_bank[-test_index2,] # Validation and Training data is rest
train_index2 = createDataPartition(traval_data2$acceptance, p = 0.50, list=FALSE) # 50%
 of remaining data as training
train_data2 = traval_data2[train_index2,]
validation_data2 = traval_data2[-train_index2,] # rest as validation

summary(train_data2)
```

```
##       Age             Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 64.00   Median :2.000
##  Mean   :45.08   Mean   :19.84   Mean   : 74.82   Mean   :2.405
##  3rd Qu.:55.00   3rd Qu.:29.00   3rd Qu.:100.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.00   Max.   :224.00   Max.   :4.000
##      CCAvg            Mortgage      Securities Account   CD Account
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.000      Min.   :0.0000
##  1st Qu.: 0.700   1st Qu.:  0.00   1st Qu.:0.000      1st Qu.:0.0000
##  Median : 1.600   Median :  0.00   Median :0.000      Median :0.0000
##  Mean   : 1.966   Mean   : 57.95   Mean   :0.103      Mean   :0.0665
##  3rd Qu.: 2.600   3rd Qu.:103.25   3rd Qu.:0.000      3rd Qu.:0.0000
##  Max.   :10.000   Max.   :587.00   Max.   :1.000      Max.   :1.0000
##      Online         CreditCard      education_1     education_2
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :1.000   Median :0.0000   Median :0.000   Median :0.0000
##  Mean   :0.576   Mean   :0.2915   Mean   :0.429   Mean   :0.2625
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000
##   education_3       acceptance
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.3085   Mean   :0.1015
##  3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
summary(validation_data2)
```

```
##       Age           Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.0   Min.   :1.000
##  1st Qu.:36.00   1st Qu.:11.00   1st Qu.: 39.0   1st Qu.:1.000
##  Median :46.00   Median :21.00   Median : 64.0   Median :2.000
##  Mean   :45.49   Mean   :20.24   Mean   : 73.2   Mean   :2.404
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 95.0   3rd Qu.:3.000
##  Max.   :67.00   Max.   :42.00   Max.   :202.0   Max.   :4.000
##      CCAvg            Mortgage      Securities Account   CD Account
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.0000     Min.   :0.000
##  1st Qu.: 0.700   1st Qu.:  0.00   1st Qu.:0.0000     1st Qu.:0.000
##  Median : 1.500   Median :  0.00   Median :0.0000     Median :0.000
##  Mean   : 1.895   Mean   : 53.73   Mean   :0.1055     Mean   :0.057
##  3rd Qu.: 2.500   3rd Qu.: 94.00   3rd Qu.:0.0000     3rd Qu.:0.000
##  Max.   :10.000   Max.   :635.00   Max.   :1.0000     Max.   :1.000
##      Online         CreditCard      education_1      education_2
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.6205   Mean   :0.2965   Mean   :0.4035   Mean   :0.2945
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   education_3      acceptance
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000
##  Mean   :0.302   Mean   :0.0955
##  3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :1.000   Max.   :1.0000
```

```
summary(test_data2)
```

```
##       Age            Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:36.00   1st Qu.:10.00   1st Qu.: 37.25   1st Qu.:1.000
##  Median :45.50   Median :20.00   Median : 62.00   Median :2.000
##  Mean   :45.56   Mean   :20.35   Mean   : 72.85   Mean   :2.364
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 99.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :42.00   Max.   :203.00   Max.   :4.000
##      CCAvg           Mortgage      Securities Account   CD Account
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.000      Min.   :0.000
##  1st Qu.: 0.700   1st Qu.:  0.00   1st Qu.:0.000      1st Qu.:0.000
##  Median : 1.500   Median :  0.00   Median :0.000      Median :0.000
##  Mean   : 1.967   Mean   : 59.12   Mean   :0.105      Mean   :0.055
##  3rd Qu.: 2.500   3rd Qu.:106.00   3rd Qu.:0.000      3rd Qu.:0.000
##  Max.   :10.000   Max.   :617.00   Max.   :1.000      Max.   :1.000
##      Online        CreditCard     education_1     education_2     education_3
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.00
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.00
##  Median :1.000   Median :0.000   Median :0.000   Median :0.000   Median :0.00
##  Mean   :0.591   Mean   :0.294   Mean   :0.431   Mean   :0.289   Mean   :0.28
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.00
##  Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.00
##    acceptance
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.086
##  3rd Qu.:0.000
##  Max.   :1.000
```

```
## Now that the data is split, normalize the data
## Copy the original data

train.norm.df2 <- train_data2
valid.norm.df2 <- validation_data2
traval.norm.df2 <- traval_data2
test.norm.df2 <- test_data2

## Use preProcess() from the caret package to normalize Sales and Age.
norm.values <- preProcess(train_data2[, 1:6], method=c("center", "scale"))
train.norm.df2[, 1:6] <- predict(norm.values, train_data2[, 1:6]) # Replace first two co
lumns with normalized
                                                                  #values
valid.norm.df2[, 1:6] <- predict(norm.values, validation_data2[, 1:6])
traval.norm.df2[, 1:6] <- predict(norm.values, traval.norm.df2[, 1:6])
test.norm.df2[, 1:6] <- predict(norm.values, test_data2[, 1:6])
summary(train.norm.df2)
```

```
##       Age              Experience          Income            Family      
##  Min.   :-1.93131   Min.   :-1.99153   Min.   :-1.4331   Min.   :-1.2212  
##  1st Qu.:-0.88164   1st Qu.:-0.85810   1st Qu.:-0.7682   1st Qu.:-1.2212  
##  Median :-0.00691   Median : 0.01378   Median :-0.2320   Median :-0.3520  
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 0.86782   3rd Qu.: 0.79846   3rd Qu.: 0.5402   3rd Qu.: 0.5172  
##  Max.   : 1.91749   Max.   : 2.01908   Max.   : 3.1998   Max.   : 1.3864  
##      CCAvg              Mortgage       Securities Account   CD Account      
##  Min.   :-1.1098   Min.   :-0.5736   Min.   :0.000      Min.   :0.0000  
##  1st Qu.:-0.7147   1st Qu.:-0.5736   1st Qu.:0.000      1st Qu.:0.0000  
##  Median :-0.2067   Median :-0.5736   Median :0.000      Median :0.0000  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   :0.103      Mean   :0.0665  
##  3rd Qu.: 0.3577   3rd Qu.: 0.4483   3rd Qu.:0.000      3rd Qu.:0.0000  
##  Max.   : 4.5343   Max.   : 5.2358   Max.   :1.000      Max.   :1.0000  
##      Online        CreditCard      education_1     education_2     
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000  
##  Median :1.000   Median :0.0000   Median :0.000   Median :0.0000  
##  Mean   :0.576   Mean   :0.2915   Mean   :0.429   Mean   :0.2625  
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000  
##  Max.   :1.000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000  
##   education_3       acceptance     
##  Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000  
##  Mean   :0.3085   Mean   :0.1015  
##  3rd Qu.:1.0000   3rd Qu.:0.0000  
##  Max.   :1.0000   Max.   :1.0000
```

```
var(train.norm.df2[, 1:6])
```

```
##                     Age  Experience      Income      Family       CCAvg
## Age         1.000000000  0.99448544 -0.03953604 -0.06359312 -0.05752006
## Experience  0.994485440  1.00000000 -0.03359696 -0.07252296 -0.05764899
## Income     -0.039536041 -0.03359696  1.00000000 -0.16755126  0.66310366
## Family     -0.063593119 -0.07252296 -0.16755126  1.00000000 -0.11580076
## CCAvg      -0.057520056 -0.05764899  0.66310366 -0.11580076  1.00000000
## Mortgage   -0.002167449 -0.00227706  0.20254366 -0.01532623  0.10838946
##                Mortgage
## Age        -0.002167449
## Experience -0.002277060
## Income      0.202543661
## Family     -0.015326230
## CCAvg       0.108389462
## Mortgage    1.000000000
```

```
summary(valid.norm.df2)
```

```
##       Age              Experience          Income             Family
##  Min.   :-1.93131   Min.   :-1.99153   Min.   :-1.43312   Min.   :-1.2212144
##  1st Qu.:-0.79417   1st Qu.:-0.77091   1st Qu.:-0.76820   1st Qu.:-1.2212144
##  Median : 0.08056   Median : 0.10096   Median :-0.23198   Median :-0.3520226
##  Mean   : 0.03586   Mean   : 0.03492   Mean   :-0.03474   Mean   :-0.0008692
##  3rd Qu.: 0.86782   3rd Qu.: 0.88565   3rd Qu.: 0.43293   3rd Qu.: 0.5171691
##  Max.   : 1.91749   Max.   : 1.93190   Max.   : 2.72796   Max.   : 1.3863608
##      CCAvg              Mortgage        Securities Account   CD Account
##  Min.   :-1.10977   Min.   :-0.57357   Min.   :0.0000     Min.   :0.000
##  1st Qu.:-0.71468   1st Qu.:-0.57357   1st Qu.:0.0000     1st Qu.:0.000
##  Median :-0.26315   Median :-0.57357   Median :0.0000     Median :0.000
##  Mean   :-0.04016   Mean   :-0.04177   Mean   :0.1055     Mean   :0.057
##  3rd Qu.: 0.30125   3rd Qu.: 0.35673   3rd Qu.:0.0000     3rd Qu.:0.000
##  Max.   : 4.53431   Max.   : 5.71088   Max.   :1.0000     Max.   :1.000
##      Online         CreditCard      education_1      education_2
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.6205   Mean   :0.2965   Mean   :0.4035   Mean   :0.2945
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   education_3      acceptance
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000
##  Mean   :0.302   Mean   :0.0955
##  3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :1.000   Max.   :1.0000
```

```
var(valid.norm.df2[, 1:6])
```

```
##                    Age  Experience      Income      Family       CCAvg
## Age         1.00757971  0.99832301 -0.03236524 -0.03347452 -0.03483634
## Experience  0.99832301  1.00153369 -0.02010025 -0.03931060 -0.03137237
## Income     -0.03236524 -0.02010025  0.94487094 -0.13064554  0.56427181
## Family     -0.03347452 -0.03931060 -0.13064554  0.98247057 -0.09254754
## CCAvg      -0.03483634 -0.03137237  0.56427181 -0.09254754  0.89247215
## Mortgage   -0.02509699 -0.02024017  0.19780384 -0.01824608  0.10834096
##                Mortgage
## Age        -0.02509699
## Experience -0.02024017
## Income      0.19780384
## Family     -0.01824608
## CCAvg       0.10834096
## Mortgage    0.99081710
```

```
## Now combine the training and validation
## Before we predict for the test set, we should combine the Training and Validation se
t, normalize the data,
## and then do the prediction.
norm.values <- preProcess(traval_data2[, 1:6], method=c("center", "scale")) # Use combin
ed set to normalize
traval.norm.df2[, 1:6] <- predict(norm.values, traval_data2[, 1:6])
test.norm.df2[, 1:6] <- predict(norm.values, test_data2[, 1:6])
summary(traval.norm.df2)
```

```
##       Age               Experience            Income             Family
##  Min.   :-1.94550   Min.   :-2.008167   Min.   :-1.4356   Min.   :-1.2263
##  1st Qu.:-0.89784   1st Qu.:-0.875197   1st Qu.:-0.7614   1st Qu.:-1.2263
##  Median :-0.02479   Median :-0.003682   Median :-0.2176   Median :-0.3532
##  Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.84825   3rd Qu.: 0.867833   3rd Qu.: 0.5219   3rd Qu.: 0.5200
##  Max.   : 1.89591   Max.   : 2.000803   Max.   : 3.2624   Max.   : 1.3931
##      CCAvg             Mortgage       Securities Account   CD Account
##  Min.   :-1.1201   Min.   :-0.5539   Min.   :0.0000     Min.   :0.00000
##  1st Qu.:-0.7140   1st Qu.:-0.5539   1st Qu.:0.0000     1st Qu.:0.00000
##  Median :-0.1918   Median :-0.5539   Median :0.0000     Median :0.00000
##  Mean   : 0.0000   Mean   : 0.0000   Mean   :0.1042     Mean   :0.06175
##  3rd Qu.: 0.3303   3rd Qu.: 0.4305   3rd Qu.:0.0000     3rd Qu.:0.00000
##  Max.   : 4.6816   Max.   : 5.7444   Max.   :1.0000     Max.   :1.00000
##      Online         CreditCard     education_1      education_2
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.000   Median :0.0000   Median :0.0000
##  Mean   :0.5982   Mean   :0.294   Mean   :0.4163   Mean   :0.2785
##  3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
##   education_3       acceptance
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.3053   Mean   :0.0985
##  3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000
```

```
summary(test.norm.df2)
```

```
##       Age              Experience           Income             Family      
##  Min.   :-1.94550   Min.   :-2.008167   Min.   :-1.43563   Min.   :-1.22632  
##  1st Qu.:-0.81054   1st Qu.:-0.875197   1st Qu.:-0.79944   1st Qu.:-1.22632  
##  Median : 0.01886   Median :-0.003682   Median :-0.26113   Median :-0.35318  
##  Mean   : 0.02375   Mean   : 0.027169   Mean   :-0.02518   Mean   :-0.03536  
##  3rd Qu.: 0.84825   3rd Qu.: 0.867833   3rd Qu.: 0.54363   3rd Qu.: 0.51995  
##  Max.   : 1.89591   Max.   : 1.913651   Max.   : 2.80563   Max.   : 1.39309  
##      CCAvg              Mortgage        Securities Account   CD Account      
##  Min.   :-1.12012   Min.   :-0.55390   Min.   :0.000      Min.   :0.000     
##  1st Qu.:-0.71400   1st Qu.:-0.55390   1st Qu.:0.000      1st Qu.:0.000     
##  Median :-0.24987   Median :-0.55390   Median :0.000      Median :0.000     
##  Mean   : 0.02107   Mean   : 0.03245   Mean   :0.105      Mean   :0.055     
##  3rd Qu.: 0.33031   3rd Qu.: 0.49747   3rd Qu.:0.000      3rd Qu.:0.000     
##  Max.   : 4.68158   Max.   : 5.56590   Max.   :1.000      Max.   :1.000     
##      Online        CreditCard     education_1     education_2     education_3   
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.00  
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.00  
##  Median :1.000   Median :0.000   Median :0.000   Median :0.000   Median :0.00  
##  Mean   :0.591   Mean   :0.294   Mean   :0.431   Mean   :0.289   Mean   :0.28  
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.00  
##  Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.00  
##    acceptance     
##  Min.   :0.000  
##  1st Qu.:0.000  
##  Median :0.000  
##  Mean   :0.086  
##  3rd Qu.:0.000  
##  Max.   :1.000
```

```
## Predict and Get the confusion matrix
knn.train2 <- knn(train = train.norm.df2[, 1:13],test = train.norm.df2[, 1:13],
                  cl = train.norm.df2$acceptance, k = 3) # k = 3 is what we found to be
 best

knn.val2 <- knn(train = train.norm.df2[, 1:13],test = valid.norm.df2[,1:13],
               cl = train.norm.df2$acceptance, k = 3) # k = 3 is what we found to be bes
t

knn.test2 <- knn(train = train.norm.df2[,1:13],test = test.norm.df2[, 1:13],
                 cl = train.norm.df2$acceptance, k = 3) # k = 3 is what we found to be be
st


confusionMatrix(knn.train2, as.factor(train.norm.df2$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1795   47
##          1    2  156
##
##                Accuracy : 0.9755
##                  95% CI : (0.9677, 0.9818)
##     No Information Rate : 0.8985
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.851
##
##  Mcnemar's Test P-Value : 3.263e-10
##
##             Sensitivity : 0.9989
##             Specificity : 0.7685
##          Pos Pred Value : 0.9745
##          Neg Pred Value : 0.9873
##              Prevalence : 0.8985
##          Detection Rate : 0.8975
##    Detection Prevalence : 0.9210
##       Balanced Accuracy : 0.8837
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(knn.val2, as.factor(valid.norm.df2$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1799   71
##          1   10  120
##
##                Accuracy : 0.9595
##                  95% CI : (0.9499, 0.9677)
##     No Information Rate : 0.9045
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7265
##
##  Mcnemar's Test P-Value : 2.617e-11
##
##             Sensitivity : 0.9945
##             Specificity : 0.6283
##          Pos Pred Value : 0.9620
##          Neg Pred Value : 0.9231
##              Prevalence : 0.9045
##          Detection Rate : 0.8995
##    Detection Prevalence : 0.9350
##       Balanced Accuracy : 0.8114
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(knn.test2, as.factor(test.norm.df2$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 913  33
##          1   1  53
##
##                Accuracy : 0.966
##                  95% CI : (0.9528, 0.9763)
##     No Information Rate : 0.914
##     P-Value [Acc > NIR] : 3.382e-11
##
##                   Kappa : 0.7399
##
##  Mcnemar's Test P-Value : 1.058e-07
##
##             Sensitivity : 0.9989
##             Specificity : 0.6163
##          Pos Pred Value : 0.9651
##          Neg Pred Value : 0.9815
##              Prevalence : 0.9140
##          Detection Rate : 0.9130
##    Detection Prevalence : 0.9460
##       Balanced Accuracy : 0.8076
##
##        'Positive' Class : 0
##
```

The model is 96.2% accurate with a sensativity of 99.78%, we find a true acceptance of the loan 99.78% of the time. We get a precision of 59.55%, or 59.55% of the time we preidct somebody will get a personal loan they do.

The accuracy, sensativity and specificty of the model are the highest on the training data when compared to the validation and testing data. This makes sense given that the model is being trained on data that it sees in the training model and tested on data that it does not in the testing model. By being able to see the underlying data, we would expect the model to perform better than when it does out of sample.