---

title: "Assignment 2" author: "Jacob Fabian" date: "2022-10-01" output: html_document

###How would this customer need to be classified?

```
###Load Data
setwd("~/Downloads")
bank = read.csv("UniversalBank.csv")
```

```
###Dummy Varriables
bank$Education = as.factor(bank$Education)

bank_dummy<-bank[ ,-c(1,5)]
bank_dummy$Personal.Loan = as.factor(bank_dummy$Personal.Loan)
bank_dummy$CCAvg = as.integer(bank_dummy$CCAvg)
set.seed(1)
train.index <- sample(row.names(bank_dummy), 0.6*dim(bank_dummy)[1])
test.index <- setdiff(row.names(bank_dummy), train.index)
train.df <- bank_dummy[train.index, ]
valid.df <- bank_dummy[test.index, ]
```

```
###New Customer
new.cust <- data.frame(Age = 40,
                       Experience = 10,
                       Income = 84,
                       Family = 2,
                       CCAvg = 2,
                       Education = 2,
                       Mortgage = 0,
                       Securities.Account = 0,
                       CD.Account = 0,
                       Online = 1,
                       CreditCard = 1)
```

```
###knn
library(class)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
new.cust$Education <- as.factor(new.cust$Education)

train.norm.df <- train.df[,-8]
valid.norm.df <- valid.df[,-8]

new.cust.norm <- new.cust
norm.values <- preProcess(train.df[, -8], method=c("center", "scale"))
train.norm.df <- predict(norm.values, train.df[, -8])
valid.norm.df <- predict(norm.values, valid.df[, -8])
```

```r
new.cust.norm <- predict(norm.values, new.cust.norm)
knn.pred <- class::knn(train = train.norm.df,
                       test = new.cust.norm,
                       cl = train.df$Personal.Loan, k = 1)

###Prediction
knn.pred
```

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

```r
knn.pred[3]
```

```
## [1] <NA>
## Levels: 0 1
```

###The customer will be classified as zero since the nearest neighbors are classified as zero.

```r
library(class)
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
for(i in 1:14) {
 knn.2 <- knn(train = train.df[,-8],test = valid.df[,-8], cl = train.df[,8], k=i, prob=TRUE)
 accuracy.df[i, 2] <- confusionMatrix(knn.2, valid.df[,8])$overall[1]
}
accuracy.df
```

```
##     k accuracy
## 1   1   0.8955
## 2   2   0.8945
## 3   3   0.9010
## 4   4   0.8955
## 5   5   0.8970
## 6   6   0.8995
## 7   7   0.8960
## 8   8   0.8920
## 9   9   0.8940
## 10 10   0.8990
## 11 11   0.9010
## 12 12   0.8975
## 13 13   0.9000
## 14 14   0.9010
```

###The best choice of k is 3

###Confusion Matrix for 3

```r
knn.3 <- knn(train = train.df[,-8],test = valid.df[,-8], cl = train.df[,8], k=3, prob=TRUE)
confusionMatrix(knn.3, valid.df[,8])
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction    0    1
##          0 1729  130
##          1   66   75
##
##                Accuracy : 0.902
##                  95% CI : (0.8881, 0.9147)
##     No Information Rate : 0.8975
##     P-Value [Acc > NIR] : 0.2674
##
##                   Kappa : 0.3819
##
##  Mcnemar's Test P-Value : 6.795e-06
##
##             Sensitivity : 0.9632
##             Specificity : 0.3659
##          Pos Pred Value : 0.9301
##          Neg Pred Value : 0.5319
##              Prevalence : 0.8975
##          Detection Rate : 0.8645
##    Detection Prevalence : 0.9295
##       Balanced Accuracy : 0.6645
##
##        'Positive' Class : 0
##
```

```
library(class)
customer.df= data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2, Education_1 = 0,
knn.4 <- knn(train = train.df[,-8],test = valid.df[,-8], cl = train.df[,8], k=3, prob=TRUE)
head(knn.4)
```

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

###New Loan Accepted

```
library(caret)
library(class)
Test.1 = createDataPartition(bank$Age, p= 0.2 , list=FALSE)
TD1  = bank [Test.1,]
Rem.d = bank[-Test.1,]
TI1 = createDataPartition(Rem.d$Age, p= 0.5 , list=FALSE)
TD1 = Rem.d[TI1,]
VD1 = Rem.d[-TI1,]
TND1 <- TD1
VND1 <- VD1
TND1 <- TD1
rem_data.norm.df_1 <- Rem.d
norm.values_1 <- preProcess(TD1[-8], method=c("center", "scale"))
TND1[-8] <- predict(norm.values_1, TD1[-8])
VND1[-8] <- predict(norm.values_1, VD1[-8])
TND1[-8] <- predict(norm.values_1, TND1[-8])
TND1[-8] <- predict(norm.values_1, TD1[-8])
```

```r
rem_data.norm.df_1[-8] <- predict(norm.values_1,Rem.d[-8])
head(TND1)
```

```
##           ID        Age Experience     Income   ZIP.Code    Family       CCAvg
## 3  -1.725036 -0.5495679 -0.4451884 -1.3542411  0.8794650 -1.207349 -0.55855940
## 8  -1.721584  0.4121322  0.3418271 -1.1186334  0.4377325 -1.207349 -0.95816039
## 14 -1.717441  1.1989777  1.0413964 -0.7330935  0.9931671  1.410683  0.29772842
## 15 -1.716750  1.8983959  1.8284119  0.8090662 -0.8141272 -1.207349  0.01229915
## 16 -1.716060  1.2864050  0.8665041 -1.1186334  1.0693474 -1.207349 -0.27313013
## 17 -1.715369 -0.6369952 -0.5326346  1.1946062  1.0443330  1.410683  1.55361723
##    Education   Mortgage Personal.Loan Securities.Account CD.Account     Online
## 3          1 -0.5583718    -0.3304659         -0.3396929 -0.2635981 -1.2218914
## 8          3 -0.5583718    -0.3304659         -0.3396929 -0.2635981 -1.2218914
## 14         2 -0.5583718    -0.3304659         -0.3396929 -0.2635981  0.8179941
## 15         1 -0.5583718    -0.3304659          2.9423638 -0.2635981 -1.2218914
## 16         3 -0.5583718    -0.3304659         -0.3396929 -0.2635981  0.8179941
## 17         3  0.7228347     3.0245178         -0.3396929 -0.2635981 -1.2218914
##    CreditCard
## 3   -0.645153
## 8    1.549245
## 14  -0.645153
## 15  -0.645153
## 16   1.549245
## 17  -0.645153
```

```r
set.seed(2019)
prediction_Q5 <- knn(train = TND1[,-8], test = VND1[,-8],
          cl = TND1[,8], k = 3, prob=TRUE)
actual= VND1$Personal.Loan
prediction_prob = attr(prediction_Q5,"prob")
table(prediction_Q5,actual)
```

```
##                actual
## prediction_Q5 -0.330465897443695 3.02451783294915
##             1                832               23
##             2                498               81
##             3                476               89
```

```r
mean(prediction_Q5==actual)
```

```
## [1] 0
```

```r
set.seed(2019)
prediction_Q5 <- knn(train = rem_data.norm.df_1[,-8], test = TND1[,-8],
          cl = rem_data.norm.df_1[,8], k = 3, prob=TRUE)
actual= TND1$Personal.Loan
prediction_prob = attr(prediction_Q5,"prob")
table(prediction_Q5,actual)
```

```
##                actual
## prediction_Q5 -0.330465897443695 3.02451783294915
```

```
##              1               813              36
##              2               469              83
##              3               521              78
```

```
mean(prediction_Q5==actual)
```

```
## [1] 0
```

#####The test set preformed better with 80% of the data, verses the training data that only used 50% of the data. The predictions are close but seems with more data then the formula runs better.