Classifying if a customer accepts or rejects loan using K-NN Algorithm

Niharika D

Problem Statement: *The goal of this project is to use k-NN to predict whether a new customer of Universal bank will accept a loan offer.*

Dataset: *Universal Bank datset contains data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage,securitiesaccount, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.*

Loading Required Libraries:

```
library(caret)
library(class)
```

Reading the dataset:

```
UniversalBank<-read.csv("./UniversalBank.csv")
```

Data Cleaning:

```
#Removing variables which are not used
remove_data<-subset(UniversalBank,select=-c(ID,ZIPCode))
UniversalBank<-remove_data

#checking for null values
null_values <- is.na(UniversalBank)
```

Data Transformation:

```
#Converting PersonalLoan to factor datatype
UniversalBank$PersonalLoan = as.factor(UniversalBank$PersonalLoan)

#Creating dummy variables for categorical variable Education
Education_1<-ifelse(UniversalBank$Education == '1',1,0)
Education_2<-ifelse(UniversalBank$Education == '2',1,0)
Education_3<-ifelse(UniversalBank$Education == '3',1,0)

#Adding newly created dummy variables back to the dataset

UniBank<-data.frame(Age=UniversalBank$Age,Experience=UniversalBank$Experience,Income=UniversalBank$Inco
```

```r
#Partitioning data into 60% and 40%

set.seed(123)
Split_data <- createDataPartition(UniBank$PersonalLoan,p=.6,list=FALSE,times=1)
Training <- UniBank[Split_data,]
Validation <- UniBank[-Split_data,]
```

```r
#Normalizing the data

Normalization <- preProcess(Training[,-(6:9)],method = c("center","scale"))
Training_norm = predict(Normalization,Training)

Validation_norm = predict(Normalization,Validation)
```

```r
#Creating TEST Dataset with given values

Test_predictor<-data.frame(Age=40,Experience=10,Income=84,Family=2,CCAvg=2,Education_1=0,Education_2=1,E

Normalization_test = predict(Normalization,Test_predictor)
```

Data Modelling:

```r
#Creating Train and Validation predictors
Train_predictor = Training_norm[,-9]
Validate_predictor = Validation_norm[,-9]
Train_label<-Training_norm[,9]
Validate_label<-Validation_norm[,9]

#Running K-NN Algorithm
Prediction <-knn(Train_predictor,
                 Normalization_test,
                 cl=Train_label,
                 k=1)
Prediction
```

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

```r
#Finding the best k value
set.seed(321)
SearchGrid <- expand.grid(k=seq(1:30))
model <- train(PersonalLoan~.,data=Training_norm,method="knn",tuneGrid=SearchGrid)
model
```

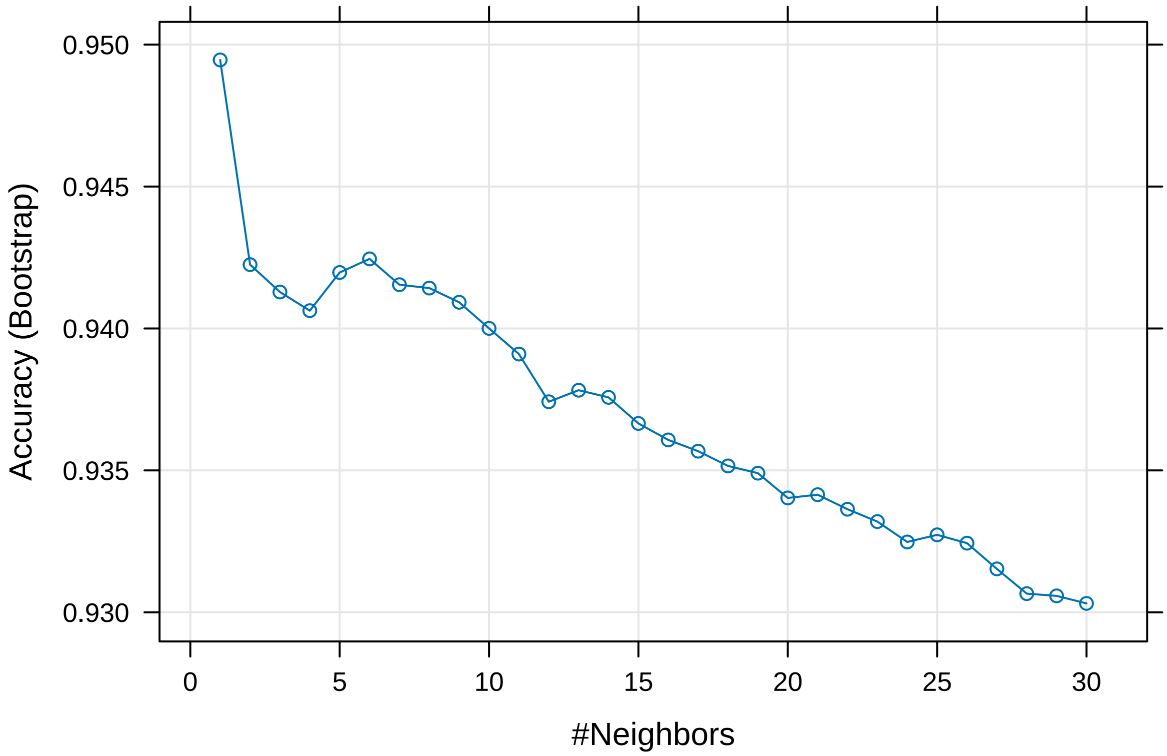
```
## k-Nearest Neighbors
##
## 3000 samples
##   13 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
```

```
## Summary of sample sizes: 3000, 3000, 3000, 3000, 3000, 3000, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9494606  0.6790540
##    2  0.9422495  0.6280947
##    3  0.9412847  0.6134538
##    4  0.9406270  0.6016192
##    5  0.9419724  0.6002352
##    6  0.9424516  0.5964779
##    7  0.9415414  0.5813768
##    8  0.9414230  0.5739398
##    9  0.9409234  0.5622319
##   10  0.9400033  0.5504214
##   11  0.9390992  0.5405417
##   12  0.9374193  0.5250147
##   13  0.9378248  0.5264158
##   14  0.9375736  0.5209409
##   15  0.9366545  0.5095868
##   16  0.9360740  0.5031104
##   17  0.9356776  0.4995591
##   18  0.9351571  0.4930217
##   19  0.9349029  0.4902960
##   20  0.9340303  0.4808043
##   21  0.9341448  0.4819498
##   22  0.9336325  0.4767749
##   23  0.9331972  0.4711490
##   24  0.9324792  0.4637587
##   25  0.9327311  0.4626125
##   26  0.9324367  0.4611510
##   27  0.9315305  0.4514573
##   28  0.9306597  0.4421025
##   29  0.9305808  0.4389133
##   30  0.9303154  0.4362343
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```r
best_k <- model$bestTune[[1]]
best_k
```

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

```r
plot(model)
```



K=1 is the choice that balances between overfitting and ignoring the predictor information.

Data Validation:

```
Prediction_new <- predict(model,Validate_predictor)

confusionMatrix(Prediction_new,Validate_label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1789   54
##          1   19  138
##
##                Accuracy : 0.9635
##                  95% CI : (0.9543, 0.9713)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7711
##
##  Mcnemar's Test P-Value : 6.909e-05
##
##             Sensitivity : 0.9895
##             Specificity : 0.7188
```

```
##          Pos Pred Value : 0.9707
##          Neg Pred Value : 0.8790
##              Prevalence : 0.9040
##          Detection Rate : 0.8945
##    Detection Prevalence : 0.9215
##       Balanced Accuracy : 0.8541
##
##        'Positive' Class : 0
##
```

```r
testing_best_k <- knn(Train_predictor,Normalization_test , cl=Train_label, k=best_k)
head(testing_best_k)
```

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

Based on the K value- we can conclude that the customer won't accept a personal loan.