Fundamentals of Machine Learning : Assignment\_2

Mderangu

library(class)

## Warning: package 'class' was built under R version 4.2.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.3

##   
## 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

library(caret)

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.3

## Loading required package: lattice

library(tinytex)

## Warning: package 'tinytex' was built under R version 4.2.3

data <- read.csv("UniversalBank.csv")  
#Elimination the ID AND ZIP CODE Columns  
data$ID<-NULL  
data$ZIP.Code<-NULL  
View(data)  
#converting to factor variable  
data$Personal.Loan=as.factor(data$Personal.Loan)  
#Checking if there is any null variables  
head(is.na(data))

## Age Experience Income Family CCAvg Education Mortgage Personal.Loan  
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## Securities.Account CD.Account Online CreditCard  
## [1,] FALSE FALSE FALSE FALSE  
## [2,] FALSE FALSE FALSE FALSE  
## [3,] FALSE FALSE FALSE FALSE  
## [4,] FALSE FALSE FALSE FALSE  
## [5,] FALSE FALSE FALSE FALSE  
## [6,] FALSE FALSE FALSE FALSE

##2  
  
#Transforming Education to character  
data$Education=as.character(data$Education)  
#Creating dummy variables  
Education\_1 <- ifelse(data$Education==1 ,1,0)  
Education\_2 <- ifelse(data$Education==2 ,1,0)  
Education\_3 <- ifelse(data$Education==3 ,1,0)  
data\_2<-data.frame(Age=data$Age,Experience=data$Experience,Income=data$Income,Family=data$Family,CCAvg=data$CCAvg, Education\_1=Education\_1,Education\_2=Education\_2,Education\_3=Education\_3,Personal.Loan=data$Personal.Loan,Mortgage=data$Mortgage,Securities.Account=data$Securities.Account,CD.Account=data$CD.Account,Online=data$Online,CreditCard=data$CreditCard)  
#defining testdata  
test\_1<-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)  
#splitting data to 60:40  
set.seed(250)  
temp<- createDataPartition(data\_2$Personal.Loan,p=.6,list=FALSE,times=1)  
train\_1 <- data\_2[temp, ]  
valid\_1<- data\_2[-temp, ]  
#Normalization  
Norm\_Model=preProcess(test\_1[,-(6:9)],method=c("center","scale"))

## Warning in preProcess.default(test\_1[, -(6:9)], method = c("center", "scale")):  
## Std. deviations could not be computed for: Age, Experience, Income, Family,  
## CCAvg, Securities.Account, CD.Account, Online, CreditCard

## Warning in preProcess.default(test\_1[, -(6:9)], method = c("center", "scale")):  
## Std. deviations could not be computed for: Age, Experience, Income, Family,  
## CCAvg, Securities.Account, CD.Account, Online, CreditCard  
train\_1\_Norm =predict(Norm\_Model,train\_1)  
valid\_1\_Norm =predict(Norm\_Model,valid\_1)  
test\_1\_Norm =predict(Norm\_Model,test\_1)  
View(train\_1\_Norm)  
#running knn algorithm  
predict\_train<-train\_1\_Norm[,-9]  
train\_sample<-train\_1\_Norm[,9]  
predict\_valid<-valid\_1\_Norm[,-9]  
2

## [1] 2

valid\_sample<-valid\_1\_Norm[,9]  
predict<-knn(predict\_train, test\_1\_Norm, cl=train\_sample,k=1)  
predict

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

##3  
  
  
#The loan offer has been denied by the customer. It is determined when the k value=0  
#Finding the best value of k  
set.seed(250)  
grid\_1<-expand.grid(k=seq(1:30))  
model\_1<-train(Personal.Loan~.,data=train\_1\_Norm,method="knn",tuneGrid=grid\_1)  
model\_1

## 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 0.8969700 0.3673762  
## 2 0.8943920 0.3536964  
## 3 0.8939792 0.3490313  
## 4 0.8962044 0.3552457  
## 5 0.8971912 0.3457064  
## 6 0.8968486 0.3335567  
## 7 0.8987524 0.3337443  
## 8 0.8990759 0.3245677  
## 9 0.8985465 0.3124463  
## 10 0.8995060 0.3140504  
## 11 0.9003620 0.3112579  
## 12 0.9007992 0.3041664  
## 13 0.9002200 0.2899583  
## 14 0.9017022 0.3002135  
## 15 0.9015162 0.2936509  
## 16 0.9007611 0.2842111  
## 17 0.9020131 0.2896747  
## 18 0.9030497 0.2899980  
## 19 0.9039421 0.2920068  
## 20 0.9040625 0.2911595  
## 21 0.9048045 0.2890007  
## 22 0.9045942 0.2851167  
## 23 0.9044351 0.2791885  
## 24 0.9041508 0.2779244  
## 25 0.9049948 0.2790018  
## 26 0.9054645 0.2828797  
## 27 0.9060156 0.2819463  
## 28 0.9055660 0.2743655  
## 29 0.9052043 0.2673272  
## 30 0.9060776 0.2721931  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 30.

#confusion matrix  
predicted<-predict(model\_1,valid\_1\_Norm[-9])  
confusionMatrix(predicted,valid\_sample)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1787 158  
## 1 21 34  
##   
## Accuracy : 0.9105   
## 95% CI : (0.8971, 0.9227)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : 0.1716   
##   
## Kappa : 0.2429   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9884   
## Specificity : 0.1771   
## Pos Pred Value : 0.9188   
## Neg Pred Value : 0.6182   
## Prevalence : 0.9040   
## Detection Rate : 0.8935   
## Detection Prevalence : 0.9725   
## Balanced Accuracy : 0.5827   
##   
## 'Positive' Class : 0   
##

##4  
  
  
#5 data is splitted to 50:30:20 ratio again  
set.seed(400)  
label\_1<-createDataPartition(data\_2$Personal.Loan,p=0.5,list=FALSE)  
4

## [1] 4

label\_2<-createDataPartition(data\_2$Personal.Loan,p=0.3,list=FALSE)  
label\_3<-createDataPartition(data\_2$Personal.Loan,p=0.2,list=FALSE)  
train\_2<-data\_2[label\_1,]  
valid\_2<-data\_2[label\_2,]  
test\_2<-data\_2[label\_3,]  
#normalizing new dataset  
normal\_1<-preProcess(train\_1[,-(6:9)],method=c("center","scale"))  
normalized\_train\_1 <- predict(normal\_1,train\_1)  
normalized\_valid\_1<-predict(normal\_1,valid\_1)  
normalized\_test\_1<-predict(normal\_1,test\_1)  
#running knn for train,validation and test data  
predict\_new\_train= normalized\_train\_1[,-9]  
predict\_new\_train\_1= normalized\_train\_1[,9]  
predict\_new\_valid=normalized\_valid\_1[,-9]  
predict\_new\_valid\_1=normalized\_valid\_1[,9]  
predict\_new\_test=normalized\_test\_1[,-9]  
predict\_new\_test\_1=normalized\_test\_1[,9]  
View(predict\_new\_test\_1)  
value\_k <- 4 # Replace 5 with your desired value  
Predict\_train\_new <- knn(predict\_new\_train, predict\_new\_train, cl = predict\_new\_train\_1, k = value\_k)  
Predict\_train\_new<-knn(predict\_new\_train,predict\_new\_train,cl=predict\_new\_train\_1,k=value\_k)  
Predict\_valid\_new<-knn(predict\_new\_train,predict\_new\_valid,cl=predict\_new\_train\_1,k=value\_k)  
#training ,validation and test data confusion matrix  
confusionMatrix(Predict\_train\_new,predict\_new\_train\_1)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 2708 91  
## 1 4 197  
##   
## Accuracy : 0.9683   
## 95% CI : (0.9614, 0.9743)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7891   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9985   
## Specificity : 0.6840   
## Pos Pred Value : 0.9675   
## Neg Pred Value : 0.9801   
## Prevalence : 0.9040   
## Detection Rate : 0.9027   
## Detection Prevalence : 0.9330   
## Balanced Accuracy : 0.8413   
##   
## 'Positive' Class : 0   
##