project 4

Dipesh Poudel

1/4/2022

# Leave one Out Validation

## Reading the File

library(haven)  
bank\_loan\_df <- read\_sav("P4\_bankloan\_5000\_clients.sav")

## Changing the data type of variables

bank\_loan\_df$defaulted\_loan<-as.factor(bank\_loan\_df$defaulted\_loan)  
bank\_loan\_df$education\_level<-as.factor(bank\_loan\_df$education\_level)

## Splitting the data into train and test set

set.seed(1234)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

ind<-sample(2,nrow(bank\_loan\_df),replace=T,prob = c(0.7,0.3))  
train\_data<-bank\_loan\_df[ind==1,]  
test\_data<-bank\_loan\_df[ind==2,]

## Setting Up the Train Control

loocv\_train\_control<-trainControl(method = "LOOCV")

## Logistic Regression With LOOCV Validation

### Training Logistic Regression Model

logistic\_clf1<-train(defaulted\_loan~.,  
 data=train\_data,  
 method="glm",  
 family="binomial",  
 trControl=loocv\_train\_control  
)  
summary(logistic\_clf1)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6490 -0.6635 -0.3442 0.1409 3.2833   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.235986 0.272446 -4.537 5.72e-06 \*\*\*  
## age 0.006492 0.008297 0.782 0.4339   
## education\_level2 0.227329 0.110244 2.062 0.0392 \*   
## education\_level3 0.260781 0.156468 1.667 0.0956 .   
## education\_level4 0.285038 0.186776 1.526 0.1270   
## education\_level5 0.020994 0.447370 0.047 0.9626   
## current\_employ\_year -0.182777 0.012678 -14.416 < 2e-16 \*\*\*  
## current\_address\_year -0.094317 0.010300 -9.157 < 2e-16 \*\*\*  
## income\_household -0.002470 0.003879 -0.637 0.5244   
## debt\_income\_ratio 0.099652 0.012885 7.734 1.04e-14 \*\*\*  
## credit\_card\_debt 0.425066 0.044558 9.540 < 2e-16 \*\*\*  
## other\_debts 0.006704 0.030495 0.220 0.8260   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3994.4 on 3524 degrees of freedom  
## Residual deviance: 2850.2 on 3513 degrees of freedom  
## AIC: 2874.2  
##   
## Number of Fisher Scoring iterations: 6

### Making the Prediction

predicted\_val\_log1<-predict(logistic\_clf1,newdata = test\_data)

### Confusion Matrix for Evaluation

confusionMatrix(predicted\_val\_log1,test\_data$defaulted\_loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1038 191  
## 1 76 170  
##   
## Accuracy : 0.819   
## 95% CI : (0.7984, 0.8383)  
## No Information Rate : 0.7553   
## P-Value [Acc > NIR] : 2.487e-09   
##   
## Kappa : 0.4513   
##   
## Mcnemar's Test P-Value : 3.022e-12   
##   
## Sensitivity : 0.9318   
## Specificity : 0.4709   
## Pos Pred Value : 0.8446   
## Neg Pred Value : 0.6911   
## Prevalence : 0.7553   
## Detection Rate : 0.7037   
## Detection Prevalence : 0.8332   
## Balanced Accuracy : 0.7013   
##   
## 'Positive' Class : 0   
##

## KNN Model with LOOCV validation

### Training KNN Model

knn\_clf1<-train(defaulted\_loan~.,data = train\_data,  
 method="knn",  
 trControl=loocv\_train\_control  
 )

### Getting the Result of the Model

knn\_clf1$result

## k Accuracy Kappa  
## 1 5 0.7636879 0.3087625  
## 2 7 0.7707801 0.3112221  
## 3 9 0.7770213 0.3248772

### Confusion Matrix for Model Evaluation

predicted\_val\_knn1<-predict(knn\_clf1,newdata = test\_data)

confusionMatrix(predicted\_val\_knn1,test\_data$defaulted\_loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1018 226  
## 1 96 135  
##   
## Accuracy : 0.7817   
## 95% CI : (0.7597, 0.8025)  
## No Information Rate : 0.7553   
## P-Value [Acc > NIR] : 0.009238   
##   
## Kappa : 0.3277   
##   
## Mcnemar's Test P-Value : 6.532e-13   
##   
## Sensitivity : 0.9138   
## Specificity : 0.3740   
## Pos Pred Value : 0.8183   
## Neg Pred Value : 0.5844   
## Prevalence : 0.7553   
## Detection Rate : 0.6902   
## Detection Prevalence : 0.8434   
## Balanced Accuracy : 0.6439   
##   
## 'Positive' Class : 0   
##

## Naïve Bayes classifier

### Training the Model

library(naivebayes)

## naivebayes 0.9.7 loaded

nb\_clf1<-train(defaulted\_loan~.,  
 data=train\_data,  
 method="naive\_bayes",  
 usepoisson = TRUE,  
 trControl=loocv\_train\_control  
 )

summary(nb\_clf1)

##   
## ================================== Naive Bayes ==================================   
##   
## - Call: naive\_bayes.default(x = x, y = y, laplace = param$laplace, usekernel = TRUE, usepoisson = TRUE, adjust = param$adjust)   
## - Laplace: 0   
## - Classes: 2   
## - Samples: 3525   
## - Features: 11   
## - Conditional distributions:   
## - KDE: 11  
## - Prior probabilities:   
## - 0: 0.7461  
## - 1: 0.2539  
##   
## ---------------------------------------------------------------------------------

### Making Prediction on Test Data

predicted\_val\_nb1<-predict(nb\_clf1,newdata = test\_data)

### Confusion Matrix for Model Evaluation

confusionMatrix(predicted\_val\_nb1,test\_data$defaulted\_loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1094 308  
## 1 20 53  
##   
## Accuracy : 0.7776   
## 95% CI : (0.7555, 0.7986)  
## No Information Rate : 0.7553   
## P-Value [Acc > NIR] : 0.02363   
##   
## Kappa : 0.1764   
##   
## Mcnemar's Test P-Value : < 2e-16   
##   
## Sensitivity : 0.9820   
## Specificity : 0.1468   
## Pos Pred Value : 0.7803   
## Neg Pred Value : 0.7260   
## Prevalence : 0.7553   
## Detection Rate : 0.7417   
## Detection Prevalence : 0.9505   
## Balanced Accuracy : 0.5644   
##   
## 'Positive' Class : 0   
##

## Support Vector Machine (SVM) Model

### Training the Model

#ctrl <- trainControl(method = "LOOCV", savePred=T)  
#svm\_clf1<-train(defaulted\_loan~.,  
# data=train\_data,  
# method="svmLinear",  
# trControl=ctrl,  
# )  
#svm\_clf

### Making the Prediction for test data

#predicted\_val\_svm1<-predict(svm\_clf1,newdata = test\_data)

### Confusion Matrix for Model Evaluation

#confusionMatrix(predicted\_val\_svm1,test\_data$defaulted\_loan)

**The Model did not Converge to a solution. Leaving it as is for now.**

## Decision Tree Model

dtree\_clf1<-train(defaulted\_loan~.,  
 data = train\_data,  
 method="rpart",  
 parms = list(split = "information"),  
 tuneLength=10,  
 trControl=loocv\_train\_control  
 )  
dtree\_clf1

## CART   
##   
## 3525 samples  
## 8 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Leave-One-Out Cross-Validation   
## Summary of sample sizes: 3524, 3524, 3524, 3524, 3524, 3524, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.002793296 0.7926241 0.3538152  
## 0.002979516 0.7863830 0.3320428  
## 0.003072626 0.7852482 0.3267308  
## 0.003351955 0.7900709 0.3357440  
## 0.004469274 0.7690780 0.2966642  
## 0.005586592 0.7804255 0.3451509  
## 0.006703911 0.7790071 0.3422901  
## 0.024581006 0.7880851 0.3481796  
## 0.027374302 0.7602837 0.2924469  
## 0.060335196 0.6669504 -0.1372405  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.002793296.

### Making the Prediction for test data

predicted\_val\_dtree1<-predict(dtree\_clf1,newdata = test\_data)

### Confusion Matrix for Model Evaluation

confusionMatrix(predicted\_val\_dtree1,test\_data$defaulted\_loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1037 235  
## 1 77 126  
##   
## Accuracy : 0.7885   
## 95% CI : (0.7667, 0.8091)  
## No Information Rate : 0.7553   
## P-Value [Acc > NIR] : 0.001443   
##   
## Kappa : 0.3285   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9309   
## Specificity : 0.3490   
## Pos Pred Value : 0.8153   
## Neg Pred Value : 0.6207   
## Prevalence : 0.7553   
## Detection Rate : 0.7031   
## Detection Prevalence : 0.8624   
## Balanced Accuracy : 0.6400   
##   
## 'Positive' Class : 0   
##

## Artifical Neural Network (ANN) Model

### Training the Model

#ann\_clf1 <- train(defaulted\_loan ~ ., data = train\_data,   
# method = "nnet",  
# preProcess = c("center","scale"),   
# maxit = 250, # Maximum number of iterations  
# tuneGrid = data.frame(size = 1, decay = 0),  
# tuneGrid = data.frame(size = 0, decay = 0),skip=TRUE, # Technically, this is log-reg  
# metric = "Accuracy",  
# trControl=loocv\_train\_control)

### Making the Predictions for Test data

#predicted\_val\_ann1<-predict(ann\_clf1,newdata = test\_data)

### Confusion Matrix for the Model Evaluation

#confusionMatrix(predicted\_val\_ann1,test\_data$defaulted\_loan)

The ANN Also Crashed the R Session for Multiple time so we discard this model for now.