model1\_LogR

2024-12-08

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

data <- read.csv("/Users/ahzenthu/Downloads/Bank Customer Churn Prediction.csv")  
head(data)

## customer\_id credit\_score country gender age tenure balance products\_number  
## 1 15634602 619 France Female 42 2 0.00 1  
## 2 15647311 608 Spain Female 41 1 83807.86 1  
## 3 15619304 502 France Female 42 8 159660.80 3  
## 4 15701354 699 France Female 39 1 0.00 2  
## 5 15737888 850 Spain Female 43 2 125510.82 1  
## 6 15574012 645 Spain Male 44 8 113755.78 2  
## credit\_card active\_member estimated\_salary churn  
## 1 1 1 101348.88 1  
## 2 0 1 112542.58 0  
## 3 1 0 113931.57 1  
## 4 0 0 93826.63 0  
## 5 1 1 79084.10 0  
## 6 1 0 149756.71 1

data$country <- factor(data$country)  
data$gender <- factor(data$gender)  
  
sample\_size <- floor(0.8 \* nrow(data))  
train\_ind <- sample(seq\_len(nrow(data)), size = sample\_size)  
train <- data[train\_ind, ]  
test <- data[-train\_ind, ]  
  
glm.fit <- glm(  
 churn ~ credit\_score + country + gender + age + tenure + balance +   
 products\_number + credit\_card + active\_member + estimated\_salary,  
 data = train,  
 family = binomial  
)  
  
predictedprob <- predict(glm.fit, newdata = test, type = "response")  
  
glm.pred <- ifelse(predictedprob > 0.5, "1", "0")   
glm.pred <- factor(glm.pred, levels = c("0", "1"))   
  
test$churn <- factor(test$churn, levels = c("0", "1"))  
CM <- confusionMatrix(glm.pred, test$churn) # Predicted first, reference second  
print(CM)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1559 302  
## 1 55 84  
##   
## Accuracy : 0.8215   
## 95% CI : (0.804, 0.8381)  
## No Information Rate : 0.807   
## P-Value [Acc > NIR] : 0.05213   
##   
## Kappa : 0.2426   
##   
## Mcnemar's Test P-Value : < 2e-16   
##   
## Sensitivity : 0.9659   
## Specificity : 0.2176   
## Pos Pred Value : 0.8377   
## Neg Pred Value : 0.6043   
## Prevalence : 0.8070   
## Detection Rate : 0.7795   
## Detection Prevalence : 0.9305   
## Balanced Accuracy : 0.5918   
##   
## 'Positive' Class : 0   
##

the model performs pretty well at predicting not churn cases with a high sensitivity and decent accuracy. the model is weak at correctly predicting churn cases, seen from the low specificity. the models precision is good at predicting not churn, 80+. but when the model predicts churn, it is only right 60 percent of the time. overall the model is better at predicting not churn than churn cases.

The model performs well for predicting non-churn cases due to high sensitivity (96.6%) and decent accuracy (82.1%). It is good at predicting when a customer will not churn.

The model struggles to correctly predict churn cases, as seen from the low specificity (23.1%). Low Kappa indicates the model is not very reliable when accounting for chance.