model2\_svm

2024-12-08

library("caret")

## Loading required package: ggplot2

## Loading required package: lattice

library("kernlab")

##   
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':  
##   
## alpha

library(ggplot2)  
  
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)  
data$churn <- factor(ifelse(as.numeric(data$churn) < 0.2, 'no', 'yes'))  
  
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, ]  
  
# had to change the CV bc my computer couldnt handle it well  
fitControl <- trainControl(method = "cv", number = 3, verboseIter = TRUE)  
  
grid <- expand.grid(degree = c(2), scale = c(0.01, 0.1), C = c(1))  
  
svmpoly <- train(  
 churn ~ credit\_score + country + gender + age + tenure + balance +   
 products\_number + credit\_card + active\_member + estimated\_salary,  
 data = train,  
 method = "svmPoly",  
 trControl = fitControl,  
 tuneGrid = grid  
)

## + Fold1: degree=2, scale=0.01, C=1   
## - Fold1: degree=2, scale=0.01, C=1   
## + Fold1: degree=2, scale=0.10, C=1   
## - Fold1: degree=2, scale=0.10, C=1   
## + Fold2: degree=2, scale=0.01, C=1   
## - Fold2: degree=2, scale=0.01, C=1   
## + Fold2: degree=2, scale=0.10, C=1   
## - Fold2: degree=2, scale=0.10, C=1   
## + Fold3: degree=2, scale=0.01, C=1   
## - Fold3: degree=2, scale=0.01, C=1   
## + Fold3: degree=2, scale=0.10, C=1   
## - Fold3: degree=2, scale=0.10, C=1   
## Aggregating results  
## Selecting tuning parameters  
## Fitting degree = 2, scale = 0.1, C = 1 on full training set

print(svmpoly)

## Support Vector Machines with Polynomial Kernel   
##   
## 8000 samples  
## 10 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (3 fold)   
## Summary of sample sizes: 5334, 5332, 5334   
## Resampling results across tuning parameters:  
##   
## scale Accuracy Kappa   
## 0.01 0.8067502 0.05566947  
## 0.10 0.8588742 0.45343043  
##   
## Tuning parameter 'degree' was held constant at a value of 2  
## Tuning  
## parameter 'C' was held constant at a value of 1  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were degree = 2, scale = 0.1 and C = 1.

prediction\_svmpoly<- predict(svmpoly, newdata = test)  
confusionMatrix(prediction\_svmpoly, test$churn)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1531 270  
## yes 35 164  
##   
## Accuracy : 0.8475   
## 95% CI : (0.831, 0.863)  
## No Information Rate : 0.783   
## P-Value [Acc > NIR] : 1.977e-13   
##   
## Kappa : 0.442   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9777   
## Specificity : 0.3779   
## Pos Pred Value : 0.8501   
## Neg Pred Value : 0.8241   
## Prevalence : 0.7830   
## Detection Rate : 0.7655   
## Detection Prevalence : 0.9005   
## Balanced Accuracy : 0.6778   
##   
## 'Positive' Class : no   
##

the accuracy is high 85+, which show this model is pretty good. the kappa shows moderate agreement, which shows that the model is strong. the sensitivity is very high which shows its strong at predicting when customers will not churn The model is excellent at identifying customers who will not churn (high sensitivity of 97). It achieves a high overall accuracy of 86.05%. The model struggles with identifying churn cases (low specificity of 36.6%). The low specificity could lead to missed opportunities for targeting at-risk customers.