Contest 3

Predicting fraud in self-checkout stores

Matt Pelz

Math/Stat 4450/8456

Spring 2020

Model development

Task

Predict occurrence of fraud using 7 variables: credit, duration, total, scans, voidedScans, attemptsWoScan, and modifiedQuantiles. Use training set including response variable ‘fraud’ to build model to predict fraud in test set.

Approach

Rely on the caret package’s train() function to tune multiple model types. Assess each individually and then explore ensemble strategies.

Assessment

After utilizing the train() functions tuning capabilities to select the optimal parameters for each model types, note the model’s accuracy in predicting fraud in the training set, deploy the model on the test set, and submit for scoring.

Model types

GBM

Tuned parameter values:

n.trees = 1500

int.depth = 5

shrinkage = 0.1

n.minobs... = 10

Training Data

Accuracy 0.9933

C5.0

Tuned parameter values:

trials = 24

model = rules

Training Data

Accuracy 0.9872

SVM

Tuned parameter values:

cost = 0.1

Training Data

Accuracy 0.9750

NNET

Tuned parameter values:

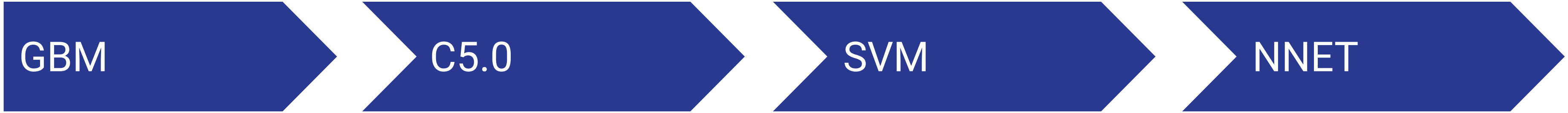
size = 6

decay = 0.1

Training Data

Accuracy 0.9802

Model types



GBM

Test Data

Accuracy 0.9904

C5.0

Test Data

Accuracy 0.9841

SVM

Test Data

Accuracy 0.9654

NNET

Test Data

Accuracy 0.9725

Simple Voting

1. GBM + C5.0 +SVM

   Test Data Accuracy

   0.9870

2. GBM + C5.0 + NNET

   Test Data Accuracy

   0.9878

Utilizing Class Probabilities

1. Average GBM and NNET

   Test Data Accuracy

   0.9853

2. Use GBM-rendered class probabilities as variable in enhanced C5.0 model

   Failure to successfully run model

Additional Boosting to GBM

Boost by fitting training data to the true value of fraud (0 or 1) minus the probability predicted by the gbm model

1. Boost with tuned nnet
2. Boost with tuned GBM

In neither case did the boosting increase the accuracy of the training data.

Threshold Selection

Identifying optimal threshold in training data

Seeking highest value of weighted sum: true pos. rate + true neg. rate

1. Run for loop to find sums for threshold values (2, 4, 6, ..., 98, 100), choose maximum
2. Use as threshold to classify test data and submit
3. Use nearby threshold values to classify test data and submit, looking for possible improvement

Final Model

GBM with n.trees = 1500, interaction depth = 5, shrinkage = 0.1, and n.minobs = 10.

Final threshold value = 0.46, adjusted downwards from the 0.48 threshold calculated as optimal for training data.

Test Data Accuracy = 0.99173