Validation of Predicted Values

Classification Rate

When developing models for prediction, the most critical metric regards how well the model does in predicting the target variable on out of sample observations. The process involves using the model estimates to predict values on the training set. Afterwards, we will compared the predicted target variable versus the observed values for each observation. In the example below, you’ll notice that our model accurately predicted 67 of the observations in the testing set.

pred = predict(mod\_fit, newdata=testing)

accuracy <- table(pred, testing[,"Class"])

sum(diag(accuracy))/sum(accuracy)

## [1] 0.705

pred = predict(mod\_fit, newdata=testing)

confusionMatrix(data=pred, testing$Class)

ROC Curve

The receiving operating characteristic is a measure of classifier performance. Using the proportion of positive data points that are correctly considered as positive and the proportion of negative data points that are mistakenly considered as positive, we generate a graphic that shows the trade off between the rate at which you can correctly predict something with the rate of incorrectly predicting something. Ultimately, we’re concerned about the area under the ROC curve, or AUROC. That metric ranges from 0.50 to 1.00, and values above 0.80 indicate that the model does a good job in discriminating between the two categories which comprise our target variable. Bear in mind that ROC curves can examine both target-x-predictor pairings and target-x-model performance. An example of both are presented below.

library(pROC)

# Compute AUC for predicting Class with the variable CreditHistory.Critical

f1 = roc(Class ~ CreditHistory.Critical, data=training)

plot(f1, col="red")

##

## Call:

## roc.formula(formula = Class ~ CreditHistory.Critical, data = training)

##

## Data: CreditHistory.Critical in 180 controls (Class Bad) < 420 cases (Class Good).

## Area under the curve: 0.5944

library(ROCR)

# Compute AUC for predicting Class with the model

prob <- predict(mod\_fit\_one, newdata=testing, type="response")

pred <- prediction(prob, testing$Class)

perf <- performance(pred, measure = "tpr", x.measure = "fpr")

plot(perf)

auc <- performance(pred, measure = "auc")

auc <- auc@y.values[[1]]

auc

## [1] 0.6540625

K-Fold Cross Validation

When evaluating models, we often want to assess how well it performs in predicting the target variable on different subsets of the data. One such technique for doing this is k-fold cross-validation, which partitions the data into k equally sized segments (called ‘folds’). One fold is held out for validation while the other k-1 folds are used to train the model and then used to predict the target variable in our testing data. This process is repeated k times, with the performance of each model in predicting the hold-out set being tracked using a performance metric such as accuracy. The most common variation of cross validation is 10-fold cross-validation.

ctrl <- trainControl(method = "repeatedcv", number = 10, savePredictions = TRUE)

mod\_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own +

CreditHistory.Critical, data=GermanCredit, method="glm", family="binomial",

trControl = ctrl, tuneLength = 5)

pred = predict(mod\_fit, newdata=testing)

confusionMatrix(data=pred, testing$Class)