Improving Model Perfromance / Tuning Parameters

Abhishek Chaturvedi

05/24/2017

Tuning Parameter

Generically and regardless of model type, what are the purposes of a model tuning parameters?

Process of adjusting vrious model options to identify the best fit model

reducing model bias

reducing errors

Caret Models

This assignment demonstrates the use of caret for constructing models. Each model should be built and compared using using `Kappa` as the performance metric calculated using 10-fold repeated cross-validation with 3 folds.

Using the rectangular data that you created for the NYCFlights to create a model for arr_delay >= 15 minutes.

- glm
- rpart
- knn
- C50
- randomForest
- adaBoost
- Two methods of your choice from the Caret Model List (you will need to install any dependencies)

Save the caret objects with the names provided.

```
# Your work here.
library('data.table')
library('rpart')
library('caret')
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
flightsDataJoined <- readRDS("flightsDataJoined.rds")
y <- "arr_delay"

# using xs generated from previous exercise
xs <- c('humid','dep_time', 'sched_dep_time','sched_arr_time','dep_delay','origin')
yx <- flightsDataJoined[,c(y,xs),with=FALSE]
yx <- na.omit(yx)
yx <- within(yx, gt15 <- ifelse(arr_delay>=15, "GT15", "LT15"))
```

```r
set.seed(333)
inTraining <- createDataPartition(yx[,gt15], p = .75, list = FALSE)
trainingData <- yx[inTraining, ]
testingData <- yx[-inTraining, ]

myCtrl <- trainControl(method="repeatedcv",number = 10,repeats = 3)

fit.glm <- train(gt15~humid+dep_time+sched_dep_time+sched_arr_time+dep_delay+origin, data = trainingDat
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
fit.knn <- train(gt15~humid+dep_time+sched_dep_time+sched_arr_time+dep_delay+origin, data = trainingDat

fit.rpart <- train(gt15~humid+dep_time+sched_dep_time+sched_arr_time+dep_delay+origin, data = trainingD

#fit.rf <- train(gt15~humid+dep_time+sched_dep_time+sched_arr_time+dep_delay+origin, data = trainingDat


#fit.myown1 <- ..
#fit.myown1 <- ..
```

Compare the models?

```
yhat.fit.glm <- predict(fit.glm,testingData,type="raw")
yhat.fit.knn <- predict(fit.knn,testingData,type="raw")
yhat.fit.rpart <- predict(fit.rpart,testingData,type="raw")
#yhat.fit.rf <- predict(fit.rf,testingData,type="raw")

confusionMatrix(data = yhat.fit.glm, reference = testingData$gt15)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction GT15 LT15
##       GT15  867  124
##       LT15  622 6220
##
##                Accuracy : 0.9048
##                  95% CI : (0.898, 0.9112)
##     No Information Rate : 0.8099
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6453
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5823
##             Specificity : 0.9805
##          Pos Pred Value : 0.8749
##          Neg Pred Value : 0.9091
```

```
##              Prevalence : 0.1901
##          Detection Rate : 0.1107
##    Detection Prevalence : 0.1265
##       Balanced Accuracy : 0.7814
##
##        'Positive' Class : GT15
##
```

```
confusionMatrix(data = yhat.fit.knn, reference = testingData$gt15)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction GT15 LT15
##       GT15  794  105
##       LT15  695 6239
##
##                Accuracy : 0.8979
##                  95% CI : (0.8909, 0.9045)
##     No Information Rate : 0.8099
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.609
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5332
##             Specificity : 0.9834
##          Pos Pred Value : 0.8832
##          Neg Pred Value : 0.8998
##              Prevalence : 0.1901
##          Detection Rate : 0.1014
##    Detection Prevalence : 0.1148
##       Balanced Accuracy : 0.7583
##
##        'Positive' Class : GT15
##
```

```
confusionMatrix(data = yhat.fit.rpart, reference = testingData$gt15)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction GT15 LT15
##       GT15  863  119
##       LT15  626 6225
##
##                Accuracy : 0.9049
##                  95% CI : (0.8982, 0.9113)
##     No Information Rate : 0.8099
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6448
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5796
```

```
##             Specificity : 0.9812
##          Pos Pred Value : 0.8788
##          Neg Pred Value : 0.9086
##              Prevalence : 0.1901
##          Detection Rate : 0.1102
##    Detection Prevalence : 0.1254
##       Balanced Accuracy : 0.7804
##
##        'Positive' Class : GT15
##
```

```
#confusionMatrix(data = yhat.fit.rf, reference = testingData$gt15)
```

Which is best? Why?

based on confusion matrix rpart is best model