mproving Model Perfromance / Tuning Parameters

Vani P jyothy

5/13/2017

Tuning Parameter

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

Tuning paameter decides how a particular model is fit.It can effect the bias-variance trade-off.BIAS measures how close the model comes to the true value.High bias ???means poor fit. VARIANCE is the stability of the model, susceptibility to new values.High variance means ???poor fit

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.

```
##       tailnum year                    type                  manufacturer
##    1:  N10156 2004 Fixed wing multi engine                      EMBRAER
##    2:  N102UW 1998 Fixed wing multi engine             AIRBUS INDUSTRIE
##    3:  N103US 1999 Fixed wing multi engine             AIRBUS INDUSTRIE
##    4:  N104UW 1999 Fixed wing multi engine             AIRBUS INDUSTRIE
##    5:  N10575 2002 Fixed wing multi engine                      EMBRAER
##   ---
## 3318:  N997AT 2002 Fixed wing multi engine                       BOEING
## 3319:  N997DL 1992 Fixed wing multi engine MCDONNELL DOUGLAS AIRCRAFT CO
## 3320:  N998AT 2002 Fixed wing multi engine                       BOEING
## 3321:  N998DL 1992 Fixed wing multi engine MCDONNELL DOUGLAS CORPORATION
## 3322:  N999DN 1992 Fixed wing multi engine MCDONNELL DOUGLAS CORPORATION
##           model engines seats speed    engine
##    1: EMB-145XR       2    55    NA Turbo-fan
##    2:  A320-214       2   182    NA Turbo-fan
##    3:  A320-214       2   182    NA Turbo-fan
##    4:  A320-214       2   182    NA Turbo-fan
##    5: EMB-145LR       2    55    NA Turbo-fan
##   ---
## 3318:   717-200       2   100    NA Turbo-fan
## 3319:     MD-88       2   142    NA Turbo-fan
## 3320:   717-200       2   100    NA Turbo-fan
## 3321:     MD-88       2   142    NA Turbo-jet
## 3322:     MD-88       2   142    NA Turbo-jet
```

Data set used after doing the necessary preprocessing and only selecting the required fiels

```
str(Aviation1)
```

```
## 'data.frame':    327346 obs. of  15 variables:
##  $ arr_delay     : int  11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ arr_delay_15  : num  1 0 0 1 1 1 0 1 1 1 ...
##  $ visib         : num  NA NA NA NA 10 NA 10 10 10 10 ...
##  $ precip        : num  NA NA NA NA 0 NA 0 0 0 0 ...
##  $ humid         : num  NA NA NA NA 57.3 ...
##  $ temp          : num  NA NA NA NA 39.9 ...
##  $ dep_delay     : int  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
##  $ carrier       : chr  "UA" "UA" "AA" "B6" ...
##  $ origin        : chr  "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr  "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : int  227 227 160 183 116 150 158 53 140 138 ...
##  $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
##  $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ distance      : int  1400 1416 1089 1576 762 719 1065 229 944 733 ...
```

```
summary(Aviation1)
```

```
##    arr_delay         arr_delay_15        visib           precip
##  Min.   : -86.000   Min.   :0.0000   Min.   : 0.00   Min.   :0.0000
##  1st Qu.: -17.000   1st Qu.:1.0000   1st Qu.:10.00   1st Qu.:0.0000
##  Median :  -5.000   Median :1.0000   Median :10.00   Median :0.0000
##  Mean   :   6.895   Mean   :0.7629   Mean   : 9.21   Mean   :0.0027
##  3rd Qu.:  14.000   3rd Qu.:1.0000   3rd Qu.:10.00   3rd Qu.:0.0000
##  Max.   :1272.000   Max.   :1.0000   Max.   :10.00   Max.   :1.1800
##                                      NA's   :1186    NA's   :1186
##      humid             temp          dep_delay          arr_time
##  Min.   : 12.74   Min.   : 10.94   Min.   : -43.00   Min.   :   1
##  1st Qu.: 46.09   1st Qu.: 41.00   1st Qu.:  -5.00   1st Qu.:1104
##  Median : 60.77   Median : 55.94   Median :  -2.00   Median :1535
##  Mean   : 61.63   Mean   : 55.68   Mean   :  12.56   Mean   :1502
##  3rd Qu.: 77.96   3rd Qu.: 71.06   3rd Qu.:  11.00   3rd Qu.:1940
##  Max.   :100.00   Max.   :100.04   Max.   :1301.00   Max.   :2400
##  NA's   :1213     NA's   :1213
##    carrier             origin              dest              air_time
##  Length:327346      Length:327346      Length:327346      Min.   : 20.0
##  Class :character   Class :character   Class :character   1st Qu.: 82.0
##  Mode  :character   Mode  :character   Mode  :character   Median :129.0
##                                                           Mean   :150.7
##                                                           3rd Qu.:192.0
##                                                           Max.   :695.0
##
##  sched_arr_time     month           distance
##  Min.   :   1   Min.   : 1.000   Min.   :  80
##  1st Qu.:1122   1st Qu.: 4.000   1st Qu.: 509
##  Median :1554   Median : 7.000   Median : 888
##  Mean   :1533   Mean   : 6.565   Mean   :1048
##  3rd Qu.:1944   3rd Qu.:10.000   3rd Qu.:1389
```

```
##  Max.   :2359   Max.   :12.000   Max.   :4983
##
```

Splitting the data into tesing and training data set

```
set.seed(1)
Train <- createDataPartition(Aviation1$arr_delay_15, p=0.1, list=FALSE,times=1)
training <- Aviation1[ Train, ]
testing <- Aviation1[ -Train, ]

#Converting the output variable to a factor variable
training$arr_delay_15=factor(training$arr_delay_15)
testing$arr_delay_15=factor(testing$arr_delay_15)

x=na.omit(training)

y=sample_n(x,3000)
```

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

```
train_control<- trainControl(method="cv", number=3, savePredictions = TRUE)

fit_glm=train(arr_delay_15 ~ dep_delay+carrier+air_time+visib+precip+temp+humid,data=training,trControl=
```

```
## 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_glm
```

```
## Generalized Linear Model
##
## 32585 samples
##     7 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 21823, 21824, 21823
## Resampling results:
```

```
##
##   Accuracy   Kappa
##   0.9030825  0.7099717
##
##
```

- rpart

```
train_control_rpart<- trainControl(method="cv", number=3, savePredictions = TRUE)


fit_rpart=train(arr_delay_15 ~ dep_delay+carrier+air_time+visib+precip+temp+humid,data=training,trContr




fit_rpart
```

```
## CART
##
## 32585 samples
##     7 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 21823, 21824, 21823
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.001157705  0.9025203  0.7060636
##   0.001243461  0.9019705  0.7054336
##   0.588757396  0.8542511  0.4677846
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was cp = 0.001157705.
```

knn

```
train_control_knn<- trainControl(method="cv", number=3, savePredictions = TRUE)


fit_knn=train(arr_delay_15 ~ dep_delay+carrier+air_time+visib+precip+temp+humid,data=y, method="knn",na


fit_knn
```

```
## k-Nearest Neighbors
##
## 3000 samples
##    7 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3000, 3000, 3000, 3000, 3000, 3000, ...
```

```
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.8726203  0.6376529
##   7  0.8835490  0.6629381
##   9  0.8869411  0.6687255
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was k = 9.
```

Random forest

```
fit_rf=train(arr_delay_15 ~ dep_delay+carrier+air_time+visib+precip+temp+humid,data=y,method="rf",
               trControl=trainControl(method="cv",number=3),
               prox=TRUE,allowParallel=TRUE)
```

```
## Loading required package: randomForest
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
fit_rf
```

```
## Random Forest
##
## 3000 samples
##    7 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 2000, 2000, 2000
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8633333  0.5536677
##   11    0.8956667  0.6957877
##   21    0.8940000  0.6926334
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 11.
```

C50

```
fit_c50=train(arr_delay_15 ~ dep_delay+carrier+air_time+visib+precip+temp+humid,data=y,method="C5.0",
               trControl=trainControl(method="cv",number=3))
```

```
## Loading required package: plyr
```

```
## ------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------
##
## Attaching package: 'plyr'
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
fit_c50
```

```
## C5.0
##
## 3000 samples
##    7 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 2001, 2000, 1999
## Resampling results across tuning parameters:
##
##   model  winnow  trials  Accuracy   Kappa
##   rules  FALSE    1      0.8996636  0.7117872
##   rules  FALSE   10      0.9009963  0.7131702
##   rules  FALSE   20      0.8959983  0.6999150
##   rules   TRUE    1      0.8996636  0.7117872
##   rules   TRUE   10      0.8993276  0.7068729
##   rules   TRUE   20      0.8996653  0.7091098
##   tree   FALSE    1      0.8996636  0.7117872
##   tree   FALSE   10      0.8979933  0.6990099
##   tree   FALSE   20      0.8946669  0.6951227
##   tree    TRUE    1      0.8996636  0.7117872
##   tree    TRUE   10      0.8989939  0.7057362
##   tree    TRUE   20      0.8973326  0.7011649
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were trials = 10, model = rules
##  and winnow = FALSE.
```

Which model is the best?

Tree model is the best because of its interpretebility.