Group_descrip_GBMother.R

daitu

Thu Jun 23 10:51:40 2016

```
## 2016年暑期课程设计####
## 问题：Grupo Bimbo Inventory Demand
##  宾堡集团的库存需求
## 最大限度地提高销售和最大限度地减少烘焙食品的退回
## Daitu
## start:2016.06.22
## 参考借鉴kaggle上的公开程序
## 使用梯度提升机进行预测

##设置工作文件夹
setwd("/Users/Daitu/数据分析/kaggle/Grupo Bimbo")
getwd()
```

```
## [1] "/Users/daitu/数据分析/kaggle/Grupo Bimbo"
```

```
## 设置集群 ####
print(paste("Set up Cluster",Sys.time()))
```

```
## [1] "Set up Cluster 2016-06-23 10:51:40"
```

```
library(h2o) # R API is just a library
```

```
## Warning: package 'h2o' was built under R version 3.2.5
```

```
## Loading required package: statmod
```

```
##
## ----------------------------------------------------------------------
##
## Your next step is to start H2O:
##     > h2o.init()
##
## For H2O package documentation, ask for help:
##     > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit http://docs.h2o.ai
##
## ----------------------------------------------------------------------
```

```
##
## Attaching package: 'h2o'
```

```
## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
## The following objects are masked from 'package:base':
##
##     &&, %*%, %in%, ||, apply, as.factor, as.numeric, colnames,
##     colnames<-, ifelse, is.character, is.factor, is.numeric, log,
##     log10, log1p, log2, round, signif, trunc
```

```
## 启动一个集群；定一位4核同时计算;
h2o.init(nthreads=4,max_mem_size='12G')
```

```
##  Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         15 minutes 7 seconds
##     H2O cluster version:        3.8.2.6
##     H2O cluster name:           H2O_started_from_R_daitu_agg862
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   8.26 GB
##     H2O cluster total cores:    8
##     H2O cluster allowed cores:  4
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     R Version:                  R version 3.2.3 (2015-12-10)
```

```
## 加载数据####

print(paste("加载数据",Sys.time()))
```

```
## [1] "加载数据 2016-06-23 10:51:40"
```

```
## 读取整个训练数据，使用所有的核
system.time({
  train<-h2o.uploadFile("train.csv",destination_frame = "train.hex")
})
```

```

```

```
##    user  system elapsed
##   0.507   3.752  34.486
```

```
train[1:5,] ## 查看训练集的前几行
```

```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
## 1      3       1110         7     3301      15766        1212             3
## 2      3       1110         7     3301      15766        1216             4
## 3      3       1110         7     3301      15766        1238             4
## 4      3       1110         7     3301      15766        1240             4
## 5      3       1110         7     3301      15766        1242             3
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil
## 1     25.14               0           0                3
## 2     33.52               0           0                4
## 3     39.32               0           0                4
## 4     33.52               0           0                4
## 5     22.92               0           0                3
##
## [5 rows x 11 columns]
```

```
## 将训练集的(预测目标+1)取对数
train$target<-log(train$Demanda_uni_equil+1)
train[1:5,]
```

```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
## 1      3       1110         7     3301      15766        1212             3
## 2      3       1110         7     3301      15766        1216             4
## 3      3       1110         7     3301      15766        1238             4
## 4      3       1110         7     3301      15766        1240             4
## 5      3       1110         7     3301      15766        1242             3
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil   target
## 1     25.14               0           0                3 1.386294
## 2     33.52               0           0                4 1.609438
## 3     39.32               0           0                4 1.609438
## 4     33.52               0           0                4 1.609438
## 5     22.92               0           0                3 1.386294
##
## [5 rows x 12 columns]
```

```
h2o.median(train$target)
```

```
## [1] 1.386294
```

```
## 数据分区
print(paste("数据分区",Sys.time()))
```

```
## [1] "数据分区 2016-06-23 10:52:21"
```

```
## 这个模型将会把数据分为3个部分，根据星期数据进行分区：
##   one to generate product averages, a second to fit a model, and a third to evalua
te the model
## 第一个数据用来生成产品均值，第二部分数据用来拟合一个模型，第三部分数据用来计算模型
dev<-train[train$Semana <= 5,]                    ## gets Semana 3,4,5
val<-train[train$Semana > 5 & train$Semana <= 8,]  ## gets Semana 6, 7,8
val[1:5,]
```

```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
## 1      6       1110         7     3301      15766        1216             1
## 2      6       1110         7     3301      15766        1238             2
## 3      6       1110         7     3301      15766        1242             3
## 4      6       1110         7     3301      15766        1250             1
## 5      6       1110         7     3301      15766        1309             6
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1      8.38               0           0                1 0.6931472
## 2     19.66               0           0                2 1.0986123
## 3     22.92               0           0                3 1.3862944
## 4      7.64               0           0                1 0.6931472
## 5     40.56               0           0                6 1.9459101
##
## [5 rows x 12 columns]
```

```
final<-train[train$Semana == 9,]                  ## gets Semana 9
final[1:5,]
```

```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
## 1      9       1110         7     3301      15766        1212             1
## 2      9       1110         7     3301      15766        1238             2
## 3      9       1110         7     3301      15766        1240             2
## 4      9       1110         7     3301      15766        1242             1
## 5      9       1110         7     3301      15766        1250            10
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1      8.38               0           0                1 0.6931472
## 2     19.66               0           0                2 1.0986123
## 3     16.76               0           0                2 1.0986123
## 4      7.64               0           0                1 0.6931472
## 5     76.40               0           0               10 2.3978953
##
## [5 rows x 12 columns]
```

```
## 模型：产品分组&GBM####

print(paste("Model: Product Groups & GBM",Sys.time()))
```

```
## [1] "Model: Product Groups & GBM 2016-06-23 10:52:31"
```

```
## 使用测试集中用来预测的字段变量进行预测，剔除ID和星期，
predictors<-c("Agencia_ID","Canal_ID","Ruta_SAK","Cliente_ID","Producto_ID")

## first part of model: use product averages, created on the dev set
##  this is the only time we will use the dev set
## 模型的第一部分：使用产品的均值，在dev数据集上创建
## 这是dev数据集的唯一的一次使用
groups<-h2o.group_by(data=dev,by="Producto_ID",mean("target"))
groups[1:5,]
```

```
##   Producto_ID mean_target
## 1          41    4.357809
## 2          53    5.852552
## 3          72    1.651355
## 4          73    1.102258
## 5         100    1.427448
##
## [5 rows x 2 columns]
```

```
h2o.median(groups$mean_target)
```

```
## [1] 1.865737
```

```
## apply groups back into dev and validation data sets as "mean_target"
## if there are NAs for this (new products), use a constant; used median of entire tr
ain target
## 使用分组后的数据集dev，生成新的确认数据（val）
## 如果数据集中有NAS（代表新的产品），使用中位数进行代替。
newVal<-h2o.merge(x=val,y=groups,all.x = T)
newVal[1:5,]
```

```
##   Producto_ID Agencia_ID Canal_ID Ruta_SAK Cliente_ID Semana Venta_uni_hoy
## 1        1216       1110        7     3301      15766      6             1
## 2        1238       1110        7     3301      15766      6             2
## 3        1242       1110        7     3301      15766      6             3
## 4        1250       1110        7     3301      15766      6             1
## 5        1309       1110        7     3301      15766      6             6
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1      8.38               0           0                 1 0.6931472
## 2     19.66               0           0                 2 1.0986123
## 3     22.92               0           0                 3 1.3862944
## 4      7.64               0           0                 1 0.6931472
## 5     40.56               0           0                 6 1.9459101
##   mean_target
## 1    1.207841
## 2    1.257216
## 3    1.586583
## 4    1.636534
## 5    1.370717
##
## [5 rows x 13 columns]
```

```
newVal$mean_target[is.na(newVal$mean_target)]<-h2o.median(groups$mean_target)
newVal[1:5,]
```

```
##   Producto_ID Agencia_ID Canal_ID Ruta_SAK Cliente_ID Semana Venta_uni_hoy
## 1        1216       1110        7     3301      15766      6             1
## 2        1238       1110        7     3301      15766      6             2
## 3        1242       1110        7     3301      15766      6             3
## 4        1250       1110        7     3301      15766      6             1
## 5        1309       1110        7     3301      15766      6             6
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1      8.38               0           0                 1 0.6931472
## 2     19.66               0           0                 2 1.0986123
## 3     22.92               0           0                 3 1.3862944
## 4      7.64               0           0                 1 0.6931472
## 5     40.56               0           0                 6 1.9459101
##   mean_target
## 1    1.207841
## 2    1.257216
## 3    1.586583
## 4    1.636534
## 5    1.370717
##
## [5 rows x 13 columns]
```

```
newFinal<-h2o.merge(x=final,y=groups,all.x = T)
newFinal[1:5,]
```

```
##   Producto_ID Agencia_ID Canal_ID Ruta_SAK Cliente_ID Semana Venta_uni_hoy
## 1        1212       1110        7     3301      15766      9             1
## 2        1238       1110        7     3301      15766      9             2
## 3        1240       1110        7     3301      15766      9             2
## 4        1242       1110        7     3301      15766      9             1
## 5        1250       1110        7     3301      15766      9            10
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1      8.38               0           0                 1 0.6931472
## 2     19.66               0           0                 2 1.0986123
## 3     16.76               0           0                 2 1.0986123
## 4      7.64               0           0                 1 0.6931472
## 5     76.40               0           0                10 2.3978953
##   mean_target
## 1    1.179147
## 2    1.257216
## 3    1.623425
## 4    1.586583
## 5    1.636534
##
## [5 rows x 13 columns]
```

```
newFinal$mean_target[is.na(newFinal$mean_target)]<-h2o.median(groups$mean_target)
newFinal[1:5,]
```

```
##   Producto_ID Agencia_ID Canal_ID Ruta_SAK Cliente_ID Semana Venta_uni_hoy
## 1        1212       1110        7     3301      15766      9             1
## 2        1238       1110        7     3301      15766      9             2
## 3        1240       1110        7     3301      15766      9             2
## 4        1242       1110        7     3301      15766      9             1
## 5        1250       1110        7     3301      15766      9            10
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1      8.38               0           0                1 0.6931472
## 2     19.66               0           0                2 1.0986123
## 3     16.76               0           0                2 1.0986123
## 4      7.64               0           0                1 0.6931472
## 5     76.40               0           0               10 2.3978953
##   mean_target
## 1    1.179147
## 2    1.257216
## 3    1.623425
## 4    1.586583
## 5    1.636534
##
## [5 rows x 13 columns]
```

```
## 训练 GBM; 使用参数以保持整体运行时间在20分钟内
## this model is fit on Semana 6 & 7 & 8, and evaluated on Semana 9.
g<-h2o.gbm(
  training_frame = newVal,      ## H2O frame holding the training data
  validation_frame = newFinal,  ## extra holdout piece for three layer modeling
  x=predictors,                 ## this can be names or column numbers
  y="target",                   ## target: using the logged variable created earlier
  model_id="gbm1",              ## internal H2O name for model
  ntrees = 25,                  ## use fewer trees than default (50) to speed up trai
ning
  learn_rate = 0.3,             ## lower learn_rate is better, but use high rate to o
ffset few trees
  score_tree_interval = 3,      ## score every 3 trees
  sample_rate = 0.5,            ## use half the rows each scoring round
  col_sample_rate = 0.8,        ## use 4/5 the columns to decide each split decision
  offset_column = "mean_target"
)
```

```
## 查看模型
summary(g)
```

```
## Model Details:
## ==============
##
## H2ORegressionModel: gbm
## Model Key:  gbm1
## Model Summary:
##   number_of_trees model_size_in_bytes min_depth max_depth mean_depth
## 1              25               10710         5         5    5.00000
##   min_leaves max_leaves mean_leaves
## 1         26         32    31.56000
##
## H2ORegressionMetrics: gbm
## ** Reported on training data. **
##
## MSE:  0.4036441
## R2 :  0.4312556
## Mean Residual Deviance :  0.4036441
##
##
## H2ORegressionMetrics: gbm
## ** Reported on validation data. **
##
## MSE:  0.408705
## R2 :  0.419607
## Mean Residual Deviance :  0.408705
##
##
##
##
## Scoring History:
##              timestamp          duration number_of_trees training_MSE
## 1  2016-06-23 10:53:11         0.000 sec               0      0.46541
## 2  2016-06-23 10:53:28        17.580 sec               3      0.42546
## 3  2016-06-23 10:53:47        35.789 sec               6      0.41739
## 4  2016-06-23 10:54:07        56.048 sec               9      0.41320
## 5  2016-06-23 10:54:29  1 min 18.640 sec              12      0.41126
## 6  2016-06-23 10:54:54  1 min 42.858 sec              15      0.40897
## 7  2016-06-23 10:55:19  2 min  8.434 sec              18      0.40759
## 8  2016-06-23 10:55:46  2 min 35.250 sec              21      0.40696
## 9  2016-06-23 10:56:17  3 min  6.513 sec              24      0.40411
## 10 2016-06-23 10:56:40  3 min 29.646 sec              25      0.40364
##    training_deviance validation_MSE validation_deviance
## 1            0.46541        0.47392             0.47392
## 2            0.42546        0.43293             0.43293
## 3            0.41739        0.42416             0.42416
## 4            0.41320        0.41954             0.41954
## 5            0.41126        0.41757             0.41757
## 6            0.40897        0.41543             0.41543
## 7            0.40759        0.41407             0.41407
## 8            0.40696        0.41342             0.41342
## 9            0.40411        0.40907             0.40907
## 10           0.40364        0.40871             0.40871
##
## Variable Importances: (Extract with `h2o.varimp`)
## =================================================
##
## Variable Importances:
```

```
##      variable relative_importance scaled_importance percentage
## 1    Canal_ID      1285062.000000          1.000000   0.519037
## 2    Ruta_SAK       639263.500000          0.497457   0.258199
## 3 Producto_ID       330203.656250          0.256955   0.133369
## 4  Cliente_ID       126865.828125          0.098724   0.051241
## 5  Agencia_ID        94461.750000          0.073508   0.038153
```

```
# 删除不再需要的较大的数据集
h2o.rm(train)
h2o.rm(dev)
h2o.rm(val)
h2o.rm(newVal)


## 进行预测#####

print(paste("Create Predictions",Sys.time()))
```

```
## [1] "Create Predictions 2016-06-23 10:57:00"
```

```
## 加载测试集
test<-h2o.uploadFile("test.csv",destination_frame = "test.hex")
```

```
##
  |
  |                                                                      |   0%
  |
  |======================================================================| 100%
```

```
test[1:5,] ## 查看测试集的前几行数据
```

```
##   id Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID
## 1  0     11       4037        1     2209    4639078       35305
## 2  1     11       2237        1     1226    4705135        1238
## 3  2     10       2045        1     2831    4549769       32940
## 4  3     11       1227        1     4448    4717855       43066
## 5  4     11       1219        1     1130     966351        1277
##
## [5 rows x 7 columns]
```

```
## merge in the offset column, just as with val and final
newTest<-h2o.merge(x=test,y=groups,all.x = T)
newTest[1:5,]
```

```
##   Producto_ID Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID id
## 1       35305     11       4037        1     2209    4639078  0
## 2        1238     11       2237        1     1226    4705135  1
## 3       32940     10       2045        1     2831    4549769  2
## 4       43066     11       1227        1     4448    4717855  3
## 5        1277     11       1219        1     1130     966351  4
##   mean_target
## 1         NaN
## 2    1.257216
## 3    1.450861
## 4    1.094048
## 5         NaN
##
## [5 rows x 8 columns]
```

```
newTest$mean_target[is.na(newTest$mean_target)]<-h2o.median(groups$mean_target)
newTest[1:5,]
```

```
##   Producto_ID Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID id
## 1       35305     11       4037        1     2209    4639078  0
## 2        1238     11       2237        1     1226    4705135  1
## 3       32940     10       2045        1     2831    4549769  2
## 4       43066     11       1227        1     4448    4717855  3
## 5        1277     11       1219        1     1130     966351  4
##   mean_target
## 1    1.865737
## 2    1.257216
## 3    1.450861
## 4    1.094048
## 5    1.865737
##
## [5 rows x 8 columns]
```

```
p<-h2o.predict(g,newTest)
```

```

```

```
p<-exp(p)-1
summary(p)
```

```
## Warning in summary.H2OFrame(p): Approximated quantiles computed! If you
## are interested in exact quantiles, please pass the `exact_quantiles=TRUE`
## parameter.
```

```
##  C1
##  Min.   :  -0.4573
##  1st Qu.:  -0.4573
##  Median :  -0.4573
##  Mean   :   5.4255
##  3rd Qu.:   3.8110
##  Max.   :4267.8600
```

```
# ## 创建提交文件#####
#
# print(paste("Create Submission",Sys.time()))
# submissionFrame<-h2o.cbind(test$id,p)
# colnames(submissionFrame)<-c("id","Demanda_uni_equil")
# h2o.exportFile(submissionFrame,path="h2o_gbmother.csv")  ## 输出文件
```