Group_randomfrost_lin_h2o.R

daitu

Sat Jun 25 15:58:08 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
## 使用使用随机森林模型进行预测

##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 2016-06-25 15:58:08"
```

```
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; 定义为4核同时计算;
h2o.init(nthreads=6,max_mem_size='12G')
```

```
## 
## H2O is not running yet, starting it now...
## 
## Note:  In case of errors look at the following log files:
##     /var/folders/bh/xgh997m97vl8yvm3sxvwfr5r0000gn/T//RtmpC56yU8/h2o_daitu_started_from_r.out
##     /var/folders/bh/xgh997m97vl8yvm3sxvwfr5r0000gn/T//RtmpC56yU8/h2o_daitu_started_from_r.err
## 
## 
## Starting H2O JVM and connecting: .. Connection successful!
## 
## R is connected to the H2O cluster: 
##     H2O cluster uptime:         1 seconds 606 milliseconds 
##     H2O cluster version:        3.8.2.6 
##     H2O cluster name:           H2O_started_from_R_daitu_kqf986 
##     H2O cluster total nodes:    1 
##     H2O cluster total memory:   10.67 GB 
##     H2O cluster total cores:    8 
##     H2O cluster allowed cores:  6 
##     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 2016-06-25 15:58:11"
```

```
## 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 
##   0.386   2.907  19.656
```

```
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
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]
```

```
dim(train)
```

```
## [1] 74180464       12
```

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

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

```
summary(train$target)
```

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

```
##  target
##  Min.   :0.000
##  1st Qu.:1.090
##  Median :1.380
##  Mean   :1.603
##  3rd Qu.:1.942
##  Max.   :8.517
```

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

```
## [1] "数据分区 2016-06-25 15:58:36"
```

```
## 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 evaluate 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
dim(dev)
```

```
## [1] 32790197       12
```

```
val<-train[train$Semana > 4 & train$Semana <= 8,]  ## gets Semana 5,6, 7,8
val[1:5,]
```

```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
## 1      5       1110        7     3301      15766        1212             5
## 2      5       1110        7     3301      15766        1216             3
## 3      5       1110        7     3301      15766        1220             3
## 4      5       1110        7     3301      15766        1238             1
## 5      5       1110        7     3301      15766        1242             2
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1     41.90               0           0                 5 1.7917595
## 2     25.14               0           0                 3 1.3862944
## 3     22.92               0           0                 3 1.3862944
## 4      9.83               0           0                 1 0.6931472
## 5     15.28               0           0                 2 1.0986123
##
## [5 rows x 12 columns]
```

```
dim(val)
```

```
## [1] 41596951       12
```

```
final<-train[train$Semana >=8,]                  ## gets Semana 8,9
final[1:5,]
```

```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
## 1      8       1110        7     3301      15766        1212             4
## 2      8       1110        7     3301      15766        1216             5
## 3      8       1110        7     3301      15766        1220             1
## 4      8       1110        7     3301      15766        1238             3
## 5      8       1110        7     3301      15766        1240             2
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1     33.52               0           0                 4 1.6094379
## 2     41.90               0           0                 5 1.7917595
## 3      7.64               0           0                 1 0.6931472
## 4     29.49               0           0                 3 1.3862944
## 5     16.76               0           0                 2 1.0986123
##
## [5 rows x 12 columns]
```

```
dim(final)
```

```
## [1] 20815581       12
```

```
## 模型：RandomForest####

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

```
## [1] "Model: Product Groups & randomForest 2016-06-25 15:59:03"
```

```
## 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=c("Producto_ID","Canal_ID"),mean("target"))
groups[1:5,]
```

```
##   Producto_ID Canal_ID mean_target
## 1          41        7    4.357809
## 2          53        4    5.852552
## 3          72        1    1.644182
## 4          72        6    2.378727
## 5          72        7    2.608634
##
## [5 rows x 3 columns]
```

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

```
## [1] 2.118703
```

```
dim(groups)
```

```
## [1] 4657    3
```

```
## 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,]
```

```
##   Canal_ID Producto_ID Semana Ruta_SAK Cliente_ID Agencia_ID Venta_uni_hoy
## 1        7        1212      5     3301      15766       1110             5
## 2        7        1216      5     3301      15766       1110             3
## 3        7        1220      5     3301      15766       1110             3
## 4        7        1238      5     3301      15766       1110             1
## 5        7        1242      5     3301      15766       1110             2
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1     41.90               0           0                 5 1.7917595
## 2     25.14               0           0                 3 1.3862944
## 3     22.92               0           0                 3 1.3862944
## 4      9.83               0           0                 1 0.6931472
## 5     15.28               0           0                 2 1.0986123
##   mean_target
## 1    1.535514
## 2    1.537746
## 3    1.532841
## 4    1.634883
## 5    1.780982
##
## [5 rows x 13 columns]
```

```
newVal$mean_target[is.na(newVal$mean_target)]<-h2o.median(groups$mean_target)
newVal[1:10,]
```

```
##   Canal_ID Producto_ID Semana Ruta_SAK Cliente_ID Agencia_ID Venta_uni_hoy
## 1        7        1212      5     3301      15766       1110             5
## 2        7        1216      5     3301      15766       1110             3
## 3        7        1220      5     3301      15766       1110             3
## 4        7        1238      5     3301      15766       1110             1
## 5        7        1242      5     3301      15766       1110             2
## 6        7        1250      5     3301      15766       1110             8
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1     41.90               0           0                 5 1.7917595
## 2     25.14               0           0                 3 1.3862944
## 3     22.92               0           0                 3 1.3862944
## 4      9.83               0           0                 1 0.6931472
## 5     15.28               0           0                 2 1.0986123
## 6     61.12               0           0                 8 2.1972246
##   mean_target
## 1    1.535514
## 2    1.537746
## 3    1.532841
## 4    1.634883
## 5    1.780982
## 6    1.994878
## 
## [10 rows x 13 columns]
```

```
dim(newVal)
```

```
## [1] 41596951       13
```

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

```
##   Canal_ID Producto_ID Semana Ruta_SAK Cliente_ID Agencia_ID Venta_uni_hoy
## 1        7        1212      8     3301      15766       1110             4
## 2        7        1216      8     3301      15766       1110             5
## 3        7        1220      8     3301      15766       1110             1
## 4        7        1238      8     3301      15766       1110             3
## 5        7        1240      8     3301      15766       1110             2
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1     33.52               0           0                 4 1.6094379
## 2     41.90               0           0                 5 1.7917595
## 3      7.64               0           0                 1 0.6931472
## 4     29.49               0           0                 3 1.3862944
## 5     16.76               0           0                 2 1.0986123
##   mean_target
## 1    1.535514
## 2    1.537746
## 3    1.532841
## 4    1.634883
## 5    1.836763
## 
## [5 rows x 13 columns]
```

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

```
##   Canal_ID Producto_ID Semana Ruta_SAK Cliente_ID Agencia_ID Venta_uni_hoy
## 1        7        1212      8     3301      15766       1110             4
## 2        7        1216      8     3301      15766       1110             5
## 3        7        1220      8     3301      15766       1110             1
## 4        7        1238      8     3301      15766       1110             3
## 5        7        1240      8     3301      15766       1110             2
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil    target
## 1     33.52               0           0                 4 1.6094379
## 2     41.90               0           0                 5 1.7917595
## 3      7.64               0           0                 1 0.6931472
## 4     29.49               0           0                 3 1.3862944
## 5     16.76               0           0                 2 1.0986123
##   mean_target
## 1    1.535514
## 2    1.537746
## 3    1.532841
## 4    1.634883
## 5    1.836763
##
## [5 rows x 13 columns]
```

```
## 训练 randomForest; 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.
rf<-h2o.randomForest(
  x=predictors,                 ## 建立模型的预测变量
  y="target",                   ## target: using the logged variable created earlier
  training_frame = newVal,      ## H2O frame holding the training data
  validation_frame = newFinal,  ## extra holdout piece for three layer modeling
  model_id="randomForest1",     ## internal H2O name for model
  ntrees = 200,                  ## 使用100棵树建立模型
  sample_rate = 0.8,            ## use 80% of the rows each scoring round
  weights_column = "mean_target"
)
```

```
## 查看模型
summary(rf)
```

```
## Model Details:
## ==============
##
## H2ORegressionModel: drf
## Model Key:  randomForest1
## Model Summary:
##   number_of_trees model_size_in_bytes min_depth max_depth mean_depth
## 1             200            60660176        20        20   20.00000
##   min_leaves max_leaves mean_leaves
## 1        368     104998 26256.45500
##
## H2ORegressionMetrics: drf
## ** Reported on training data. **
## Description: Metrics reported on Out-Of-Bag training samples
##
## MSE:  0.5565291
## R2 :  0.3818045
## Mean Residual Deviance :  0.5565291
##
##
## H2ORegressionMetrics: drf
## ** Reported on validation data. **
##
## MSE:  0.557135
## R2 :  0.3774114
## Mean Residual Deviance :  0.557135
##
##
##
##
## Scoring History:
##             timestamp          duration number_of_trees training_MSE
## 1 2016-06-25 15:59:39         0.007 sec               0
## 2 2016-06-25 15:59:59        20.118 sec               1      0.54708
## 3 2016-06-25 16:00:40  1 min  0.640 sec               4      0.58149
## 4 2016-06-25 16:02:17  2 min 38.377 sec              11      0.60043
## 5 2016-06-25 16:06:22  6 min 43.491 sec              26      0.58038
## 6 2016-06-25 16:16:59 17 min 19.760 sec              56      0.55842
## 7 2016-06-25 16:42:33 42 min 54.495 sec             139      0.55277
## 8 2016-06-25 17:04:46       1:05:06.746             200      0.55653
##   training_deviance validation_MSE validation_deviance
## 1
## 2           0.54708        0.55148             0.55148
## 3           0.58149        0.53614             0.53614
## 4           0.60043        0.56081             0.56081
## 5           0.58038        0.56545             0.56545
## 6           0.55842        0.55364             0.55364
## 7           0.55277        0.55272             0.55272
## 8           0.55653        0.55713             0.55713
##
## Variable Importances: (Extract with `h2o.varimp`)
## =================================================
##
## Variable Importances:
##      variable relative_importance scaled_importance percentage
## 1 Producto_ID   1136510336.000000          1.000000   0.385112
## 2    Canal_ID    985370368.000000          0.867014   0.333897
## 3    Ruta_SAK    581313536.000000          0.511490   0.196981
## 4  Agencia_ID    153383392.000000          0.134960   0.051975
## 5  Cliente_ID     94541512.000000          0.083186   0.032036
```

```
# 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 2016-06-25 17:13:16"
```

```
## 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,]
```

```
##   Canal_ID Producto_ID Agencia_ID id Ruta_SAK Cliente_ID Semana
## 1        1       35305       4037  0     2209    4639078     11
## 2        1        1238       2237  1     1226    4705135     11
## 3        1       32940       2045  2     2831    4549769     10
## 4        1       43066       1227  3     4448    4717855     11
## 5        1        1277       1219  4     1130     966351     11
##   mean_target
## 1         NaN
## 2    1.216841
## 3    1.433608
## 4    1.043424
## 5         NaN
## 
## [5 rows x 8 columns]
```

```
dim(newTest)
```

```
## [1] 6999251       8
```

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

```
##   Canal_ID Producto_ID Agencia_ID id Ruta_SAK Cliente_ID Semana
## 1        1       35305       4037  0     2209    4639078     11
## 2        1        1238       2237  1     1226    4705135     11
## 3        1       32940       2045  2     2831    4549769     10
## 4        1       43066       1227  3     4448    4717855     11
## 5        1        1277       1219  4     1130     966351     11
##   mean_target
## 1    2.118703
## 2    1.216841
## 3    1.433608
## 4    1.043424
## 5    2.118703
## 
## [5 rows x 8 columns]
```

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

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

```
## [1] 6999251       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.   :  0.000
##  1st Qu.:  3.471
##  Median :  4.084
##  Mean   :  4.967
##  3rd Qu.:  4.900
##  Max.   :102.092
```

```
## 对预测值四舍五入
p <- round(p)
p[p<0]<- 0
sum(p == 0)
```

```
## [1] 120
```

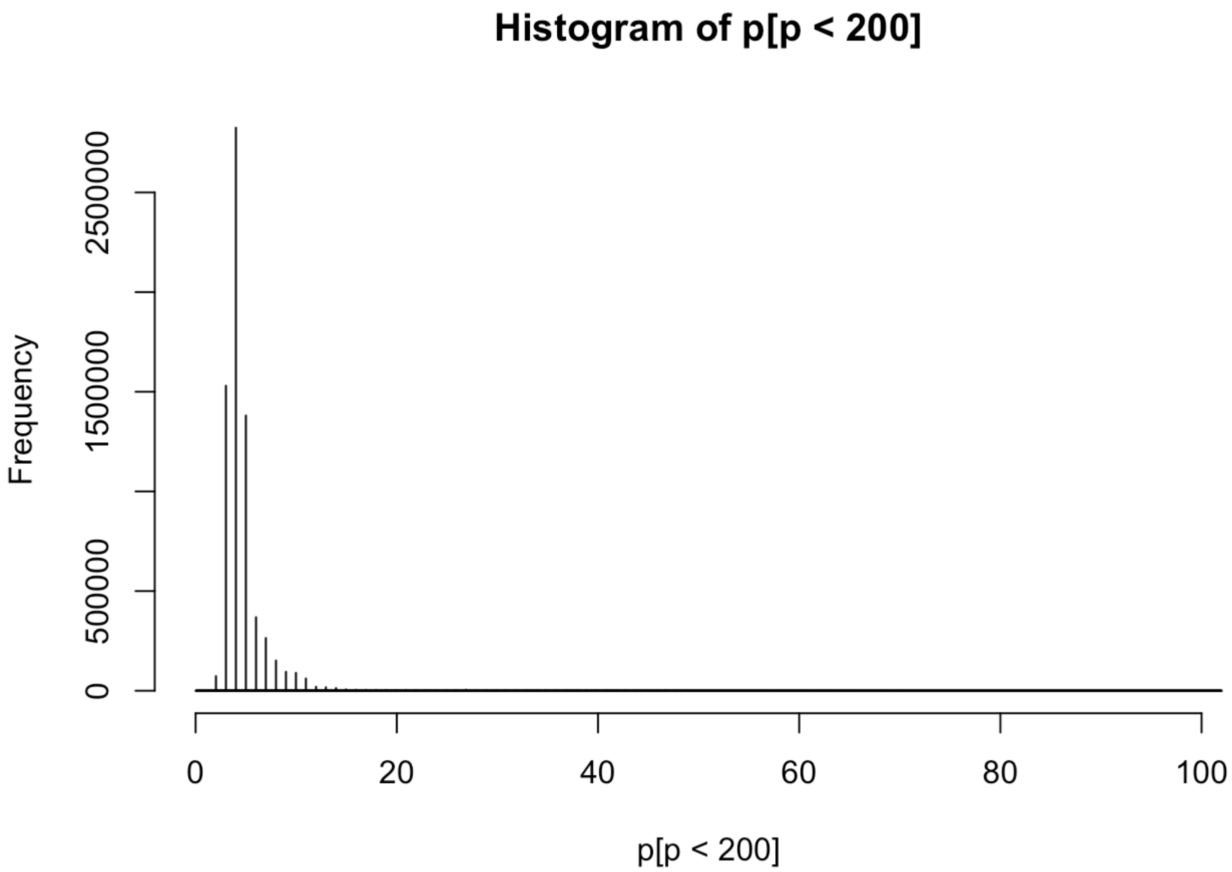
```
## 预测值的分布
summary(p)
```

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

```
##  C1
##  Min.   :  0.000
##  1st Qu.:  4.000
##  Median :  4.000
##  Mean   :  4.965
##  3rd Qu.:  5.000
##  Max.   :102.000
```

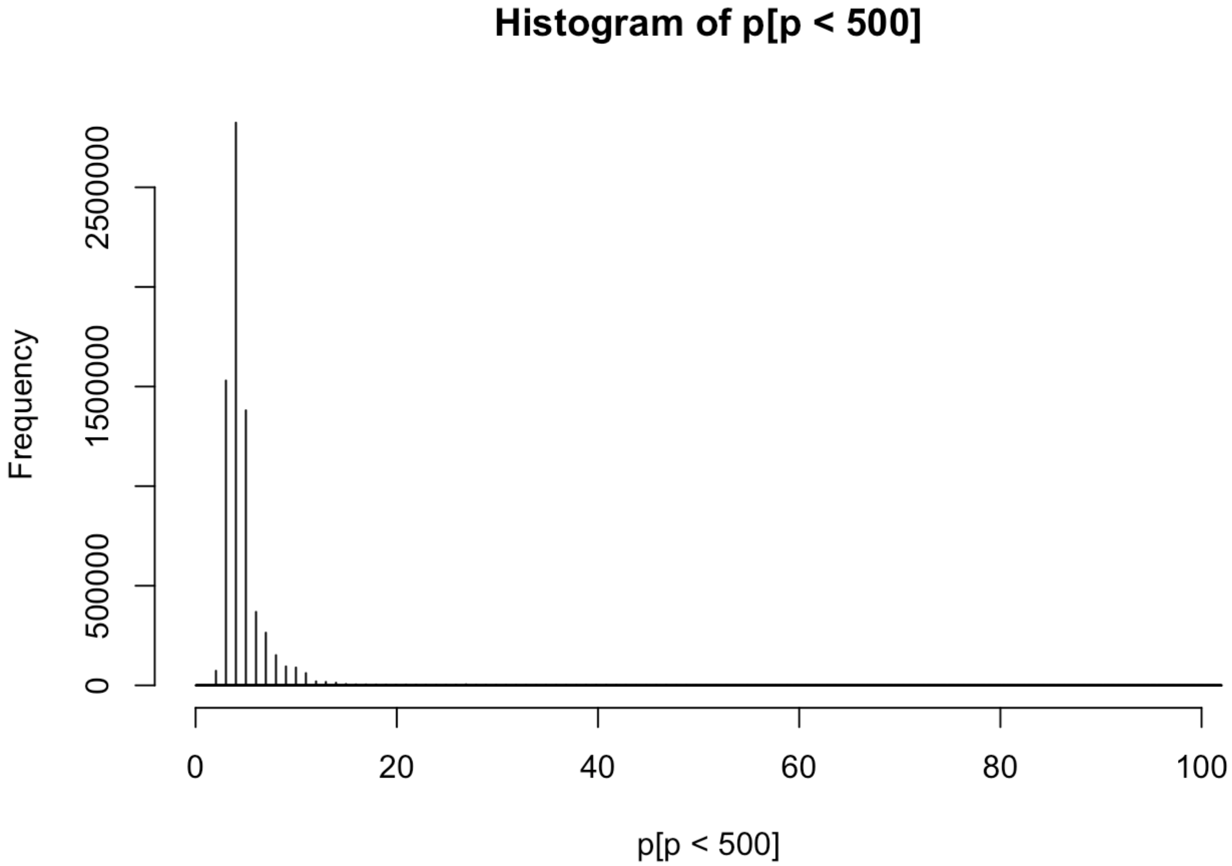
```
h2o.hist(p[p<200],breaks = "FD")
```

```
## Warning in histo$counts/sum(histo$counts) * 1/diff(histo$breaks): 长的对象
## 长度不是短的对象长度的整倍数
```



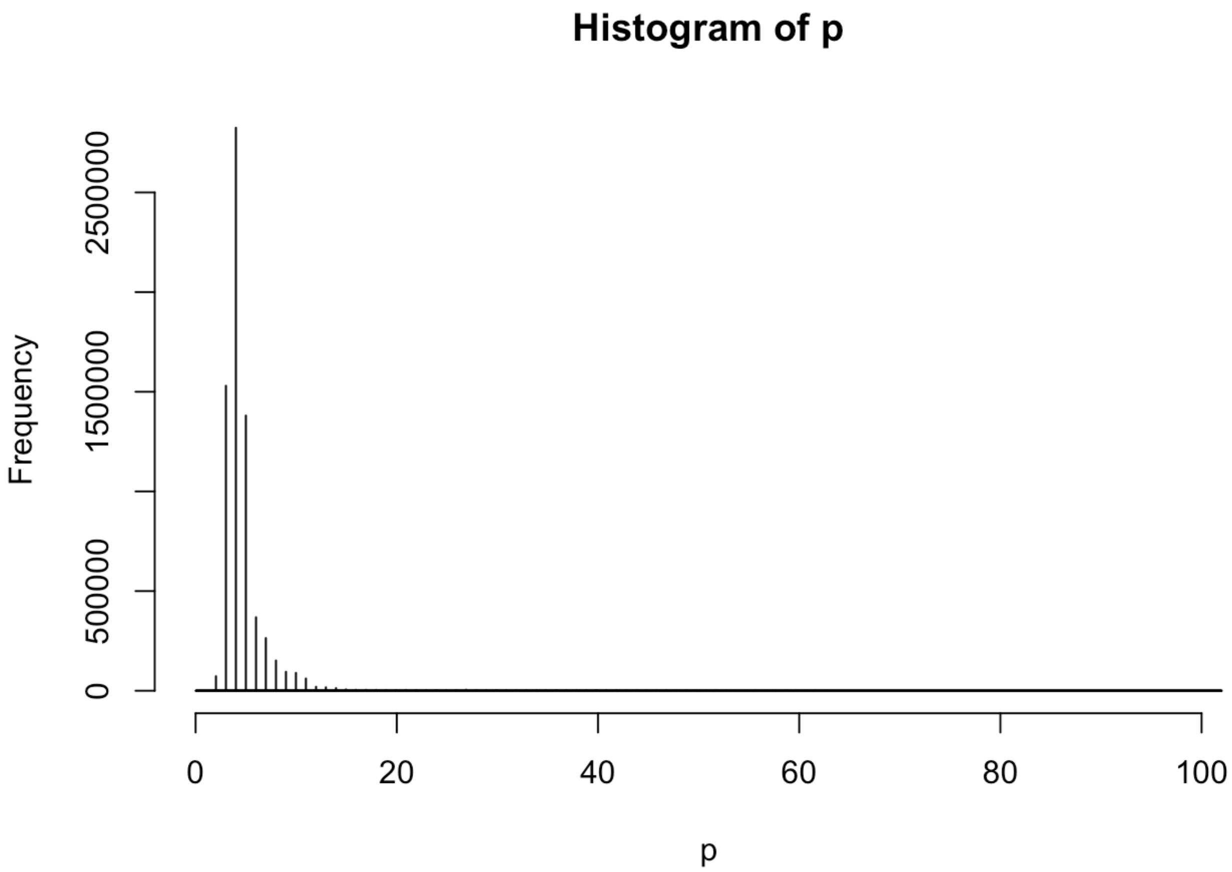
```
h2o.hist(p[p<500],breaks = "FD")
```

```
## Warning in histo$counts/sum(histo$counts) * 1/diff(histo$breaks): 长的对象
## 长度不是短的对象长度的整倍数
```



```
h2o.hist(p,breaks = "FD" )
```

```
## Warning in histo$counts/sum(histo$counts) * 1/diff(histo$breaks): 长的对象
## 长度不是短的对象长度的整倍数
```



```
## 创建提交文件#####

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

```
## [1] "Create Submission 2016-06-25 17:16:35"
```

```
submissionFrame<-h2o.cbind(test$id,p)
dim(submissionFrame)
```

```
## [1] 6999251       2
```

```
colnames(submissionFrame)<-c("id","Demanda_uni_equil")
submissionFrame[1:10,]
```

```
##   id Demanda_uni_equil
## 1  0                 4
## 2  1                 3
## 3  2                 4
## 4  3                 4
## 5  4                 7
## 6  5                 5
##
## [10 rows x 2 columns]
```

```
h2o.exportFile(submissionFrame,path="sh2o_rf.csv")  ## 输出文件
```