WOTD

51CTO

全球软件开发技术峰会

World Of Tech 2017

2017年12月1日-2日 • 深圳中洲万豪酒店

DEVELOPMENT

WOTD

51CTO

大数据进行机器学习

基于Microsoft R实现

谢佳标

平安人寿 & 数据挖掘专家

目录

01 R是什么？

语言平台

- 专注于统计，分析和数据科学
- 数据可视化的框架
- 开源

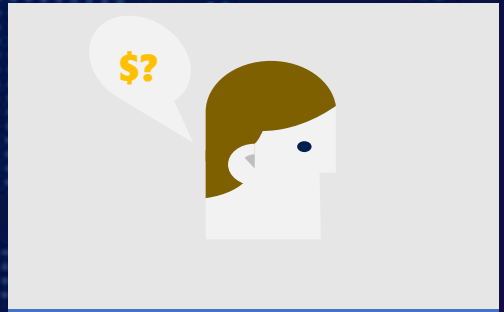
生态系统

- CRAN: 10000+ 免费的算法, 测试数据和开发包
- 许多包都可以应用到大数据计算

社区

- 上百万统计和分析学家，数据科学家正在使用R
- 大学统计学的课程
- 非常活跃的社区

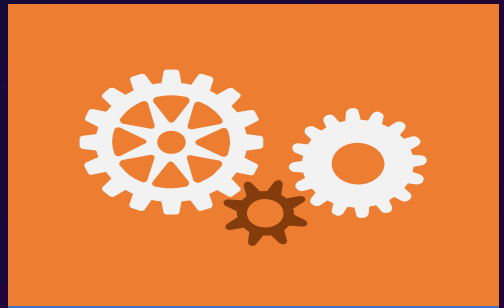
01 开源R的问题



不确定的投入成本和收益

1. 稳定的平台
2. 产品支持的挑战

如何让R和企业不断改变和发展的数据平台整合



规模和性能

01 Microsoft R Server 的好处

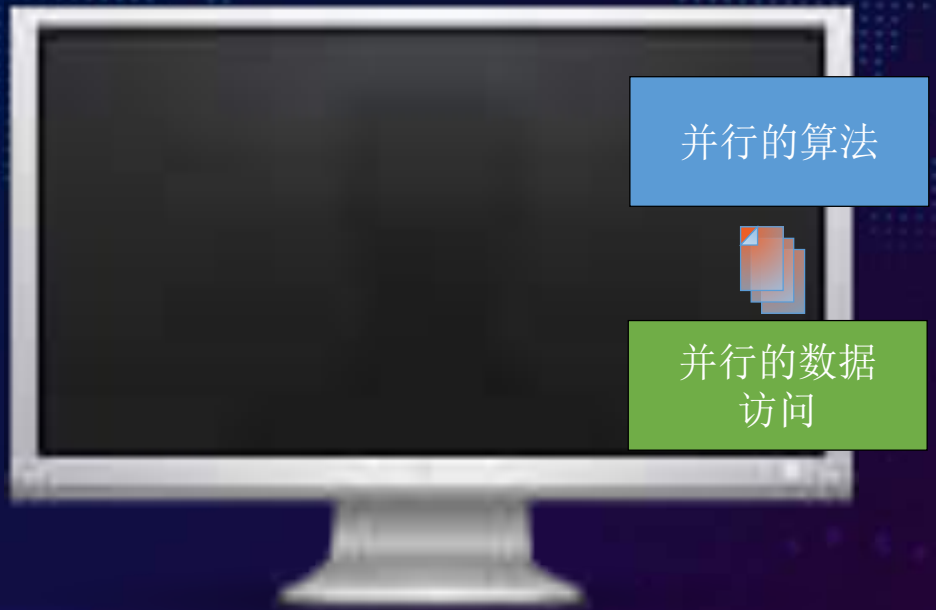
		微软R	好处
大数据	内存绑定	可扩充性的混合内存及磁盘	对大数据集的处理
分析速度	单线程	并行线程和处理	大大缩短分析时间
企业准备	社区支持	企业支持	企业级的产品服务和支持
分析的广度 & 深度	**10000+ innovative analytic packages**	**利用和优化开放源包再加上准备好的大数据包**	**企业级的R**
商业可行性	开源软件的部署风险	商业化的授权和保障	减少企业使用开源产品的投入成本和风险

01 Microsoft R Server 简介

- **Microsoft R Server** 是一款基于R的企业级大数据分析平台
- 支持各种大数据统计分析，预测性模型和机器学习功能
- 支持基于R的全套数据分析过程-探索、分析、可视化和建模等
- 通过利用和扩展开源R， R Server 能在企业级规模下进行数据分析，并与开源 R 脚本、函数和CRAN软件包，百分百兼容

01 Microsoft R Server 的革新

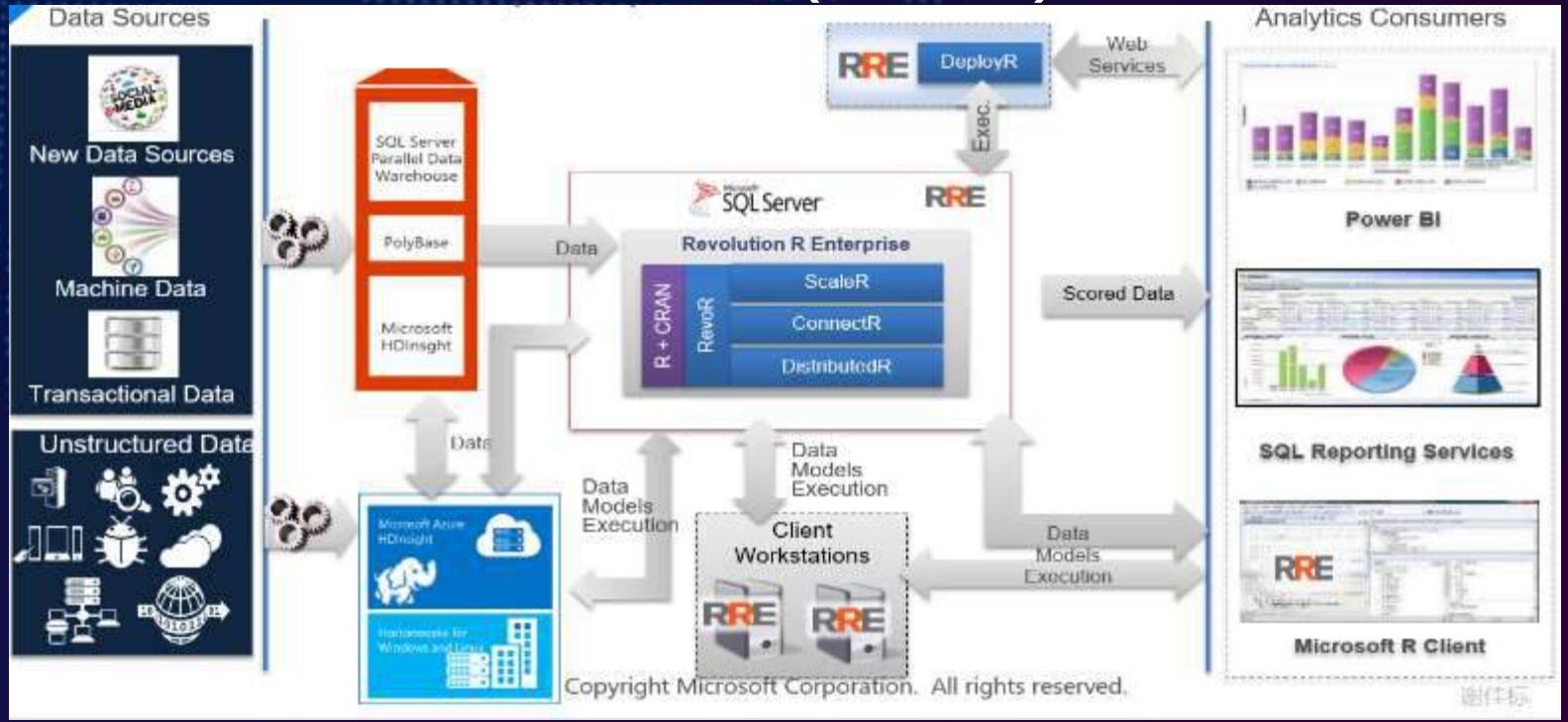
并行化 & 模块化



两个主要革新

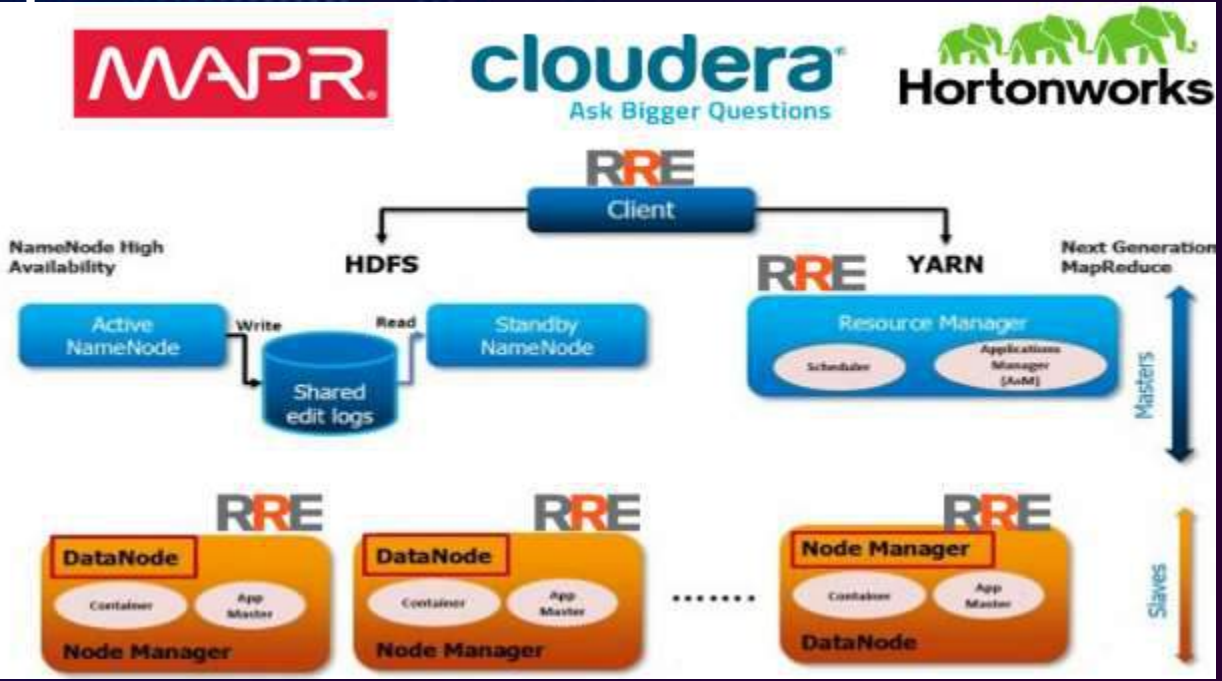
- 用户透明的并行能力加速和规模化计算
- “模块化”的处理消除了内存的限制

01 Microsoft R Server（RRE）技术架构



01 在Hadoop里的大数据分析

- 采用Hadoop节点作为R的计算引擎
- 避免大数据集的提取
- 加速模型开发进程
- 开发人员不需要学习和编写MapReduce R程序
- 利用全数据开发更精准的模型



Copyright Microsoft Corporation. All rights reserved.

目录

02 ScaleR 函数和算法

数据预处理

- Data import – Delimited, Fixed, SAS, SPSS, OBDC
- Variable creation & transformation
- Recode variables
- Factor variables
- Missing value handling
- Sort, Merge, Split
- Aggregate by category (means, sums)

描述性统计

- Min / Max, Mean, Median (approx.)
- Quantiles (approx.)
- Standard Deviation
- Variance
- Correlation
- Covariance
- Sum of Squares (cross product matrix for set variables)
- Pairwise Cross tabs
- Risk Ratio & Odds Ratio
- Cross-Tabulation of Data (standard tables & long form)
- Marginal Summaries of Cross Tabulations

统计检验

- Chi Square Test
- Kendall Rank Correlation
- Fisher's Exact Test
- Student's t-Test

抽样

- Subsample (observations & variables)
- Random Sampling

预测模型

- Sum of Squares (cross product matrix for set variables)
- Multiple Linear Regression
- Generalized Linear Models (GLM) exponential family distributions: binomial, Gaussian, inverse Gaussian, Poisson, Tweedie. Standard link functions: cauchit, identity, log, logit, probit. User defined distributions & link functions.
- Covariance & Correlation Matrices
- Logistic Regression
- Classification & Regression Trees
- Predictions/scoring for models
- Residuals for all models

变量选择

- Stepwise Regression

模拟

- Simulation (e.g. Monte Carlo)
- Parallel Random Number Generation

聚类分析

- K-Means

分类

- Decision Trees
- Decision Forests
- Gradient Boosted Decision Trees
- Naïve Bayes

结合

- PEMA-R API
- rxDataStep
- rxExec

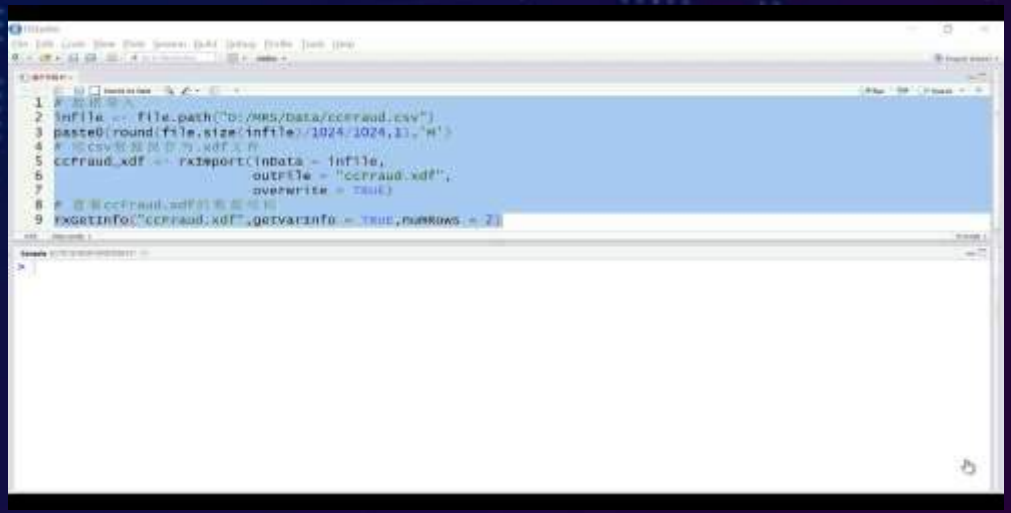
02 数据导入

- rxImport函数可以从数据源（文本、SAS、SPSS、ODBC、……）导入到R中保存为数据框；此外，如果是大数据集，MRS也提供了将数据集先保存为.xdf格式（在硬盘中），在分布式文件系统（如Hadoop的HDFS）上，XDF文件可以将数据存储在多个物理文件中，以适应非常大的数据集
- rxGetInfo函数查看xdf文件的数据结构

02

数据导入例子

```
infile <- file.path("D:/MRS/Data/ccFraud.csv")
# 将csv数据保存为.xdf文件
ccFraud_xdf <- rxImport(inData = infile,outFile = "ccFraud.xdf",overwrite = TRUE)
# 查看ccFraud.xdf的数据结构
rxGetInfo("ccFraud.xdf",getVarInfo = TRUE,numRows = 10)
```



02 数据转换

- 我们可以在保存.xdf文件时，利用rxImport函数stringsAsFactors，colClasses，和colInfo等参数改变变量的数据类型
- 我们也可以rxDataStep函数的transforms 、varsToKeep、varsToDrop、等参数进行数据转换和子集选择

02 数据转换例子

- 利用colInfo将变量gender从数值型变为因子型，且因子水平为“F”、“M”，利用colClasses将变量fraudRisk从数值型变成因子型

```
# 改变变量的数据存储类型
ccFraud_xdf <- rxImport(inData = infile,
                        outFile = "ccFraud.xdf",
                        colClasses = c(fraudRisk = "factor"),
                        colInfo = list("gender" = list(type = "factor",
                                                       levels = c("1","2"),
                                                       newLevels = c("F","M"))),
                        overwrite = TRUE)

# 查看ccFraud.xdf的数据结构
rxGetInfo(ccFraud_xdf,getVarInfo = TRUE,numRows = 5)
```

```
> rxGetInfo(ccFraud_xdf,getVarInfo = TRUE,numRows = 5)
File name: C:\Program Files\Microsoft\R Server\R_SERVER\library\RevoScaleR\rxLibs\x64\ccFraud.xdf
Number of observations: 1e+07
Number of variables: 9
Number of blocks: 20
Compression type: zlib
Variable information:
Var 1: custID, Type: integer, Low/High: (1, 1e+07)
Var 2: gender
       2 factor levels: F M
Var 3: state, Type: integer, Low/High: (1, 51)
Var 4: cardholder, Type: integer, Low/High: (1, 2)
Var 5: balance, Type: integer, Low/High: (0, 41485)
Var 6: numTrans, Type: integer, Low/High: (0, 100)
Var 7: numIntlTrans, Type: integer, Low/High: (0, 60)
Var 8: creditLine, Type: integer, Low/High: (1, 75)
Var 9: fraudRisk
       2 factor levels: 0 1
Data (5 rows starting with row 1):
  custID gender state cardholder balance numTrans numIntlTrans creditLine fraudRisk
1      1      F    35          1    3000        4           14          2         0
2      2      M     2          1       0        9            0         18         0
3      3      M     2          1       0       27            9         16         0
4      4      F    15          1       0       12            0          5         0
5      5      F    46          1       0       11           16          7         0
>
```

02 数据转换例子

对航空飞行数据集，利用rxDataStep函数中的varsToKeep函数对所需变量进行筛选，且通过transforms参数增加两个新变量：

flightDate和speed

```
airlinesXdf <- rxDataStep(inData = "D:/MRS/Data/flights.csv",
                  outFile = "flights.xdf",
                  varsToKeep =
c("year","month","day","distance","air_time","carrier","arr_delay"),
                  transforms = list(
                    flightDate = as.Date(paste(year, month, day, sep = "-")),
                    speed = distance / (air_time / 60)))
rxGetInfo(airlinesXdf,getVarInfo = TRUE,numRows = 6)
```

```
> rxGetInfo(airlinesXdf,getVarInfo = TRUE,numRows = 6)
File name: C:\Program Files\Microsoft\R Server\R_SERVER\library\RevoScaleR\rxLibs\x64\flights.xdf
Number of observations: 336776
Number of variables: 9
Number of blocks: 1
Compression type: zlib
Variable information:
Var 1: year, Type: integer, Low/High: (2013, 2013)
Var 2: month, Type: integer, Low/High: (1, 12)
Var 3: day, Type: integer, Low/High: (1, 31)
Var 4: distance, Type: integer, Low/High: (17, 4983)
Var 5: air_time, Type: integer, Low/High: (20, 695)
Var 6: carrier, Type: character
Var 7: arr_delay, Type: integer, Low/High: (-86, 1272)
Var 8: flightDate, Type: Date, Low/High: (2013-01-01, 2013-12-31)
Var 9: speed, Type: numeric, Low/High: (76.8000, 703.5846)
Data (6 rows starting with row 1):
  year month day distance air_time carrier arr_delay flightDate    speed
1 2013     1   1     1400      227      UA        11 2013-01-01 370.0441
2 2013     1   1     1416      227      UA        20 2013-01-01 374.2731
3 2013     1   1     1089      160      AA        33 2013-01-01 408.3750
4 2013     1   1     1576      183      B6       -18 2013-01-01 516.7213
5 2013     1   1      762      116      DL       -25 2013-01-01 394.1379
6 2013     1   1      719      150      UA        12 2013-01-01 287.6000
>
```

02 数据排序及合并

- 利用rxSort函数对数据框或者.xdf文件进行排序
- 利用rxSplit函数将.xdf文件或数据框分割成多个.xdf文件或数据框
- 利用rxMerge函数使用各种合并类型合并两个.xdf文件或数据框

02 数据排序例子

对信用卡欺诈数据，利用rxSort函数按照balance变量进行降序排序

```
head(rxSort(ccFraud_xdf,sortByVars = "balance",
        decreasing = TRUE))
```

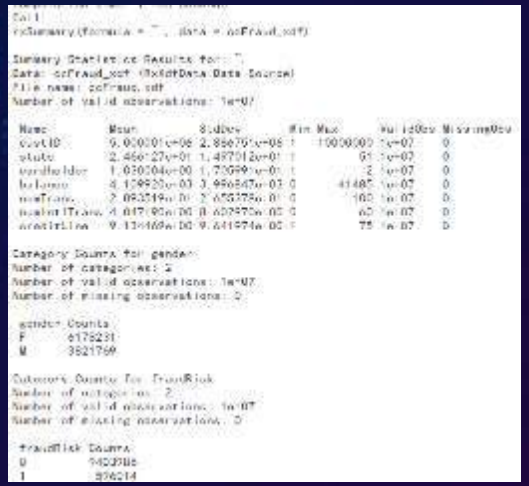
```
Time to sort data file: 9.126 seconds
   custID gender state cardholder balance numTrans numIntlTrans creditLine fraudRisk
1 3086052      F    10          1   41485       35            0         56         1
2 9957409      M    48          1   39987       84            1         56         1
3  471478      F    35          1   39725       61            0         41         1
4  162445      M    39          1   39554        6           16         52         1
5 7202754      F    51          1   37557       10            0         40         1
6 9123140      F     3          1   37000       10            1         36         1
```

02 描述性统计

- 可以利用rxSummary函数对数据进行基本汇总统计，包括按组计算

rxSummary(~.,ccFraud_xdf) # 对全部变量进行统计

rxSummary(~creditLine:fraudRisk,data = ccFraud_xdf) # 根据fraudRisk变量对creditLine变量进行分组统计



```
Call:
rxSummary(formula = ~creditLine:fraudRisk, data = ccFraud_xdf)

Summary Statistics Results for: ~creditLine:fraudRisk
Data: ccFraud_xdf (RxXdfData Data Source)
File name: ccFraud.xdf
Number of valid observations: 1e+07

 Name                  Mean     StdDev   Min Max ValidObs MissingObs
 creditLine:fraudRisk 9.134469 9.641974 1   75  1e+07    0

Statistics by category (2 categories):

 Category                    fraudRisk Means     StdDev   Min Max ValidObs
 creditLine for fraudRisk=0 0          8.219344  7.92463 1   75  9403986
 creditLine for fraudRisk=1 1         23.573429 18.63502 1   75   596014
>
```

02 基本图形功能

- 可以利用rxHistogram创建直方图
- 可以利用rxLinePlot创建线图
- 可以利用rxLorenz计算可绘制的洛伦兹曲线
- 可以利用rxRocCurve计算和绘制来自实际和预测数据的ROC曲线。

目录

03 机器学习算法

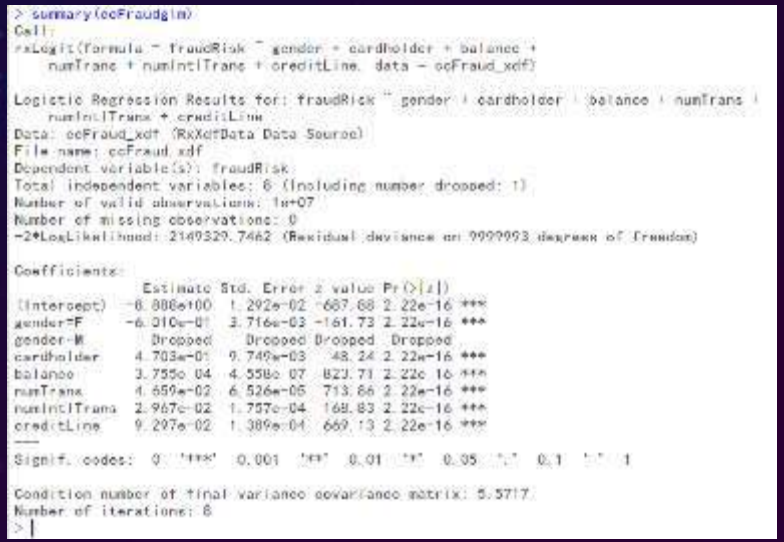
R有丰富的机器学习算法 ，MRS中的RevoScaleR包也包含了各种机器学习算法，常见算法如下表：

rxLinMod	线性回归模型	rxLogit	逻辑回归模型
rxGlm	广义线性回归模型	rxDTree	适用于数据分类或回归树
rxBTrees	使用随机梯度增强算法对数据进行分类或回归决策	rxDForest	随机森林
rxPredict	计算拟合模型的预测。输出必须是XDF数据源	rxRoc	ROC曲线

03 逻辑回归模型

利用rxLogit函数构建逻辑回归，summary函数查看模型结果

```
# logitic回归模型
ccFraudglm <- rxLogit(fraudRisk ~ gender + cardholder + balance + numTrans
                      + numIntlTrans + creditLine,data = ccFraud_xdf)
# 查看模型结果
summary(ccFraudglm)
```



```
> summary(ccFraudglm)
Call:
rxLogit(formula = fraudRisk ~ gender + cardholder + balance +
    numTrans + numIntlTrans + creditLine, data = ccFraud_xdf)

Logistic Regression Results for: fraudRisk ~ gender + cardholder + balance + numTrans +
    numIntlTrans + creditLine
Data: ccFraud_xdf (RxXdfData Data Source)
File name: ccFraud.xdf
Dependent variable(s): fraudRisk
Total independent variables: 6 (Including number dropped: 1)
Number of valid observations: 1e+07
Number of missing observations: 0
-2*LogLikelihood: 2149329.7462 (Residual deviance on 9999993 degrees of freedom)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.808e+00  1.292e-02 -687.68 2.22e-16 ***
gender=F    -6.910e-01  3.716e-03 -161.73 2.22e-16 ***
gender=M        Dropped     Dropped Dropped  Dropped
cardholder   4.703e-01  9.749e-03   48.24 2.22e-16 ***
balance      3.755e-04  4.558e-07  823.71 2.22e-16 ***
numTrans     4.659e-02  6.526e-05  713.86 2.22e-16 ***
numIntlTrans 2.967e-02  1.757e-04  168.83 2.22e-16 ***
creditLine   9.297e-02  1.389e-04  669.13 2.22e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Condition number of final variance covariance matrix: 5.5717
Number of iterations: 6
>
```

03 决策树模型

使用rxDTree函数实现决策树建模，如果响应变量是因子型，构建的是分类树，如果是数值型则是回归树

```
# 构建决策树模型
ccFraudTree <- rxDTree(fraudRisk ~ gender + cardholder + balance
                       + numTrans
                       + numIntlTrans + creditLine,data = ccFraud_xdf,
                       blocksPerRead = 30, maxDepth = 5, cp = 1e-5)
ccFraudTree
```

```
> ccFraudTree
Call:
rxDTree(formula = fraudRisk ~ gender + cardholder + balance +
    numTrans + numIntlTrans + creditLine, data = ccFraud_xdf,
    maxDepth = 5, cp = 1e-05, blocksPerRead = 30)
File:  C:\Users\daniel.xie\Documents\ccFraud.xdf
Number of valid observations:  1e+07
Number of missing observations:  0

Tree representation:
n= 1e+07

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1e+07 596014 0 (0.94039860 0.05960140)
   2) balance< 11045 9419495 328058 0 (0.96517244 0.03482756)
     4) creditLine< 42.5 9285931 273585 0 (0.97053769 0.02946231)
       8) numTrans< 74.5 8421728 131787 0 (0.98435155 0.01564845) *
       9) numTrans>=74.5 864203 141798 0 (0.83592050 0.16407950)
        18) balance< 6005 694081  66616 0 (0.90402273 0.09597727) *
        19) balance>=6005 170122  75182 0 (0.55807009 0.44192991)
          38) creditLine< 8.5 84187  27760 0 (0.67025788 0.32974212) *
          39) creditLine>=8.5 85935  38513 1 (0.44816431 0.55183569) *
     5) creditLine>=42.5 133564  54473 0 (0.59215807 0.40784193)
      10) balance< 4460 81264  21809 0 (0.73162778 0.26837222)
        20) numTrans< 49.5 66751  12993 0 (0.80535123 0.19464877) *
        21) numTrans>=49.5 14513   5697 1 (0.39254462 0.60745538)
          42) creditLine< 62.5 9352   4583 0 (0.50994440 0.49005560) *
          43) creditLine>=62.5 5161    928 1 (0.17981011 0.82018989) *
```

03 查看cp信息

查看模型的cp等相关信息，其中cp是每次分类对应的复杂度系数。

```
> ccFraudTree$cptable
           CP nsplit rel error    xerror       xstd
1  0.0644179499      0 1.0000000 1.0000000 0.001256110
2  0.0408899791      2 0.8711641 0.8715936 0.001177456
3  0.0121529584      3 0.8302741 0.8305023 0.001150849
4  0.0109292735      6 0.7938152 0.7940032 0.001126563
5  0.0052330986      8 0.7719567 0.7730624 0.001112336
6  0.0049825452      9 0.7667236 0.7626700 0.001105193
7  0.0037381672     12 0.7517760 0.7487861 0.001095561
8  0.0020158922     14 0.7442996 0.7450697 0.001092966
9  0.0018506277     16 0.7402678 0.7411604 0.001090228
10 0.0003120732     18 0.7365666 0.7367595 0.001087135
11 0.0000100000     19 0.7362545 0.7365012 0.001086953
>
```

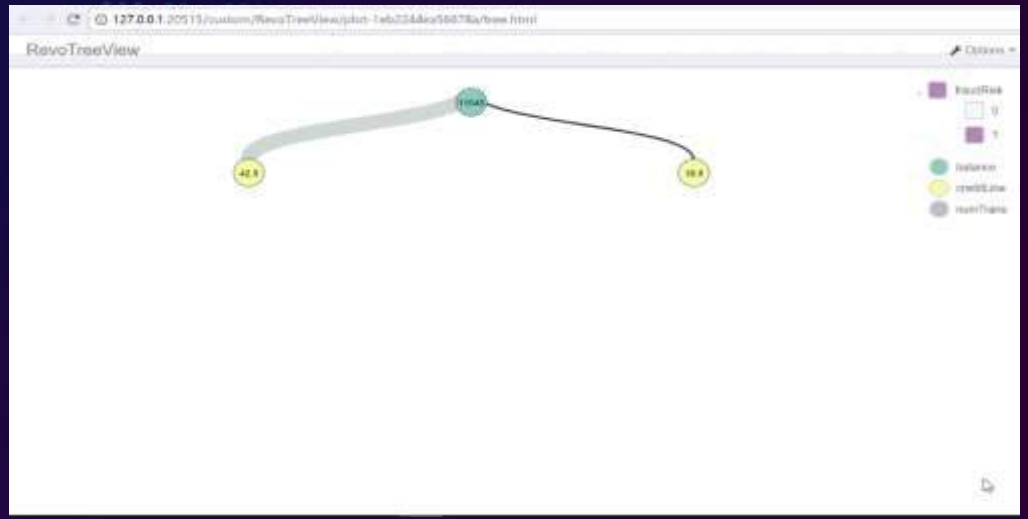
随着分割数的增加，交叉验证错误（xerror）稳步下降，注意到当nsplit=12时，变化显著减慢

03 决策树剪枝及可视化

我们可以使用prune.rxDTree函数进行决策树剪枝，利用

RevoTreeView包中的createTreeView函数对生成的树进行交互可视化

```
# 对决策树进行剪枝
ccFraudTree1 <- prune.rxDTree(ccFraudTree,
                    cp = 0.0037381672)
# 对决策树进行可视化
library(RevoTreeView)
plot(createTreeView(ccFraudTree1))
```



03 MicrosoftML包介绍

MicrosoftML包提供了新的机器学习功能，具有更高的速度，性能和可扩展性，特别是处理大量的文本数据或高维分类数据。

Algorithm	ML tasks supported	Scalability	Application Examples
rxFastLinear() Fast Linear model (SDCA)	binary classification, linear regression	#rows: ~1B #cols: ~2B CPU: multi-proc	Mortgage default prediction Email spam filtering
rxOneClassSvm() OneClass SVM	anomaly detection	cols: ~1K #rows: RAM-bound CPU: single proc	Credit card fraud detection
rxFastTrees() Fast Tree	binary classification, regression	#cols: ~30K #rows: RAM-bound CPU: multi proc	Demography prediction
rxFastForest() Fast Forest	binary classification, regression	#cols: ~30K #rows: RAM-bound CPU: multi proc	Churn Prediction
rxNeuralNet() Neural Network	binary and multiclass classification, regression	#cols: ~10Ks #rows: Inf CPU: multi proc, CUDA GPU	Check signature recognition, OCR, Click Prediction
rxLogisticRegression() Logistic regression	binary and multiclass classification	#cols: ~100M #rows: Inf for single proc CPU #rows: RAM-bound for multi proc CPU	Classifying sentiments from feedback

```
> # 模型一：利用MicrosoftML包的rxFastTrees()函数构建快速决策树模型
> (a <- Sys.time()) #模型运行前时间
> treeModel <- rxFastTrees(fraudRisk ~ gender + cardholder + balance + numTrans
+ + numIntlTrans + creditLine,data = ccFraud_xdf)
> (b <- Sys.time()) #模型运行后时间
> b-a # 模型运行时长
Time difference of 1.086313 mins
> # 模型二：利用MicrosoftML包的rxFastForest()函数构建快速随机森林模型
> (a <- Sys.time()) #模型运行前时间
> forestModel <- rxFastForest(fraudRisk ~ gender + cardholder + balance + numTrans
+ + numIntlTrans + creditLine,data = ccFraud_xdf)
> (b <- Sys.time()) #FastTrees模型运行后时间
> b-a # 模型运行时长
Time difference of 1.433823 mins
> # 模型三：利用MicrosoftML包的rxLogisticRegression()函数构建快速逻辑回归模型
> (a <- Sys.time()) #模型运行前时间
> logitModel <- rxLogisticRegression(fraudRisk ~ gender + cardholder + balance + numTrans
+ + numIntlTrans + creditLine,data = ccFraud_xdf)
> (b <- Sys.time()) #模型运行后时间
> b-a # 模型运行时长
Time difference of 20.27396 secs
```

Thank you!