2017京东登录识别-R代码

GitHub：https://github.com/hallo128/Compete/tree/master/jd_game

处理数据前：先用SPSS和Excel进行数据的查看（排序、描述统计、类别数、比例、异常等）

1、代码实现功能介绍

R实现部分，我将主要解释说明，特征构建和结果导出部分；模型构建的话，不限于R语言做出的结果。

- **构建特征表**【时间类型转换、分组统计特征、数据集合并】-训练集、测试集

输入：赛题给定数据.csv

输出：可用于建模的特征表.csv

- **模型构建**【平衡样本、分训练集和验证集、模型（选参数、预测结果）】

输入：用于建模的特征表.csv

输出：测试集的预测数据.csv

- **结果导出**【数据匹配、按指定格式导出】

输入：测试集的预测数据.csv（适用于不同软件做出的结果）

输出：可用于提交的数据.csv

2、构建特征表

（1）代码文件：

由于构建特征的时候，我们有过很多思路，按天、按1/4天、按ip，但都是基于一开始的文件演变的。

而且最后结果显示第一个文件的特征说明也比较好，所以我只特别介绍一个文件。

① 文件：“构建新的特征login_trade.R”、“login_trade测试集.R”

实现逻辑：

- 合并交易数据与登录数据（为之后做准备：将交易的最后一次的登录那天的登录特征作为本次交易的登录消息）
- 交易数据特征：将交易的数据“按id，按小时”进行切分，并分组统计交易特征；
- 登录数据特征：从合并数据中取出需要的全部登录数据，对这部分登录数据“按id，按天”进行切分，并分组统计登录特征；
- “按id，按天”合并交易与登录特征，得到最终的特征表

由上述实现逻辑，你可以看出，**最关键的内容就是，分组统计与数据合并**。这作为人工构建特征最熟悉的招数，熟悉SQL的人，也可以很好很快的实现相关内容。

① 文件2：“构建特征6个小时-训练集.R”、“构建特征6个小时-测试集.R”

特别地，登录数据“按id，按1/4天”进行切分

② 文件3：“构建特征6个小时-训练集+ip.R”、“构建特征6个小时-测试集+ip.R”

特别地，在文件2的基础上，添加了ip数据的信息

R编程方法介绍【时间类型转换、分组统计特征、数据集合并】

1、时间类型转换

时间类型转换【字符串-时间戳-取时间】

R语言中可以方便实现这3个内容的相互转换

（1）为什么需要转换

- **实际记录需要：**

很多比赛都会给出时间戳或时刻点，这个指标存在的原因：一方面，是由于系统在记录用户行为的时候，肯定会记录下用户是什么时刻发生了这样的行为；另一方面，很多时候我们考虑和观测到的都是一个随着时间变化的行为，正因为有了时间的记录和排序，从而让我们有可能随着时间来得到有用的信息。

- **建模构建特征需要：**

首先，时间戳有利于你将数据按时间排序。

其次，时间戳便于你提取时间特征（年、月、日、小时、分钟、周）

最后，时间戳便于计算时间差（只需要2个时间戳相减）和平均时刻

datetime 格式定义

代码	
%Y	4位数的年
%y	2位数的年
%m	2位数的月[01,12]
%d	2位数的日[01，31]
%H	时（24小时制）[00,23]
%I	时（12小时制）[01,12]
%M	2位数的分[00,59]
%S	秒[00,61]有闰秒的存在
%w	用整数表示的星期几[0（星期天），6]
%F	%Y-%m-%d简写形式例如，2017-06-27
%D	%m/%d/%y简写形式

（2）R语言如何实现

R语言中重点介绍2个函数：as.POSIXct()和strptime()，下面是这2个函数的具体使用方法

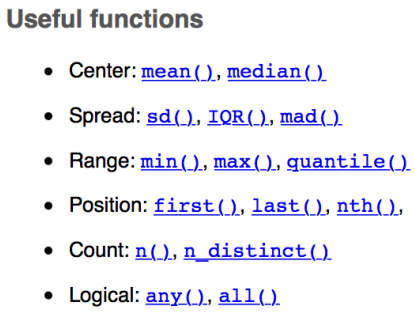
```
t1 = 1428641659              #时间戳（数值）
time1 = '2015-04-10 12:54:19'  #时刻（字符串）

#----------------时间转换
#----法一：as.POSIXct()
#用法：输入-时间戳/字符串，先转换为日期型；再按需要，转换为字符串/时间戳
#用法1：将时间戳转换字符串
l0 = as.POSIXct(t1, origin="1970-01-01 00:00:00")    #将时间戳转换为日期型
lc = as.character(l0)                                #将日期型转换为字符串
#用法2：将字符串转换时间戳
l1 = as.POSIXct(time1, origin="1970-01-01 00:00:00") #将字符串转换为日期型
l2 = as.numeric(l1)                                  #将日期型转换为时间戳
l3 = as.numeric(as.POSIXct(time1, origin="1970-01-01 00:00:00")) #将字符串转换为时间戳（最常用）

#----法二：strptime()
#特别地：输入-只能为字符串
l4 = as.numeric(strptime(time1,"%Y-%m-%d %H:%M:%S")) #将字符串转换为时间戳(用法类似)
#特点：任意取出时间点
hour = strftime(time1, format = "%m-%d %H")
```

2、分组统计特征

这个内容对于经常使用数据库，写SQL语言的同学，应该不陌生。写R代码时，查到过好几包，目前我选择了**dplyr包**。接下来我将说明它的具体用法。

（1）**步骤：**

- 先把需要分组的关键字转换为“因子型”
- group_by（）——数据没有变化，为第三步做准备
- 对第二步的数据集调用summarise()，可以进行的统计指标，如右图所示【缺点：只能调用右图的函数】



（2）例子

```
#----------按id，按小时进行分组统计
#1-转换为因子
data_trade$id <- as.factor(data_trade$id)
data_trade$hour <- as.factor(data_trade$hour)
#2-分组统计
library(dplyr)
data_by_idHour <- data_trade %>% group_by(id,hour)
#最终的整理交易数据
trade_idHour = summarise(data_by_idHour,
                         timestamp = mean(timestamp),          #时间戳（平均时间）
                         is_risk = as.numeric(any(is_risk == 1)),  #是否有风险(只要出现就视为有风险)
                         time = first(time),                   #首次的时间
                         trade_count = n())                    #登录总次数
```

（3）特别地，结合dplyr::filter()使用，可以按条件进行分组统计

3、数据集合并及相关数据处理方法

- 合并数据集——merge()
- 过滤数据集——subset()与dplyr::filter()
- 检查数据完整性并填充——complete.cases()、na.omit()、is.na()
- 多指标排序——order()

- 纵向合并数据——cbind()
- 对某列数据进行切分——splitstackshape::cSplit()
- 导出数据——write.csv()

```
#1.合并数据集——merge【按关键字合并】
login_trade1 = merge(trade_idHour,data_login, by=c("time",'timestamp','id'), all =TRUE)

#2.过滤数据
#法一：选取数据子集——subset(数据集,条件,需要的X变量)
ll=subset(data_trade, is_risk == 1, select = -hour)
#法二：dplyr::filter——结合summarise使用(需要先进行group_by)
data_login_by_idDay <- data_login %>% group_by(id,day)
data_result_NOT_1 = filter(data_login_by_idDay, result != 1)
struc_data2 = summarise(data_result_NOT_1,
                        result_NOT_1_count = n(),          #不为1的result出现的总次数
                        result_NOT_1_num = n_distinct(result))  #不为1的result出现的可能情况种类

#3.检查数据完整性并填充
sum(complete.cases(tradeId_loginM))          #完整的数量
tradeId_login_full = na.omit(tradeId_loginM)  #只保留完整数据
struc_data[is.na(struc_data)] <- 0           #对na位置进行填充
#取出符合要求数据的位置——which
index_na=which(is.na(login_trade3$ip))

#4.多指标排序——order
#按id，与时间戳进行排序
login_trade2 = login_trade1[order(login_trade1$id,login_trade1$timestamp),]  #排序查看

#5.纵向合并数据——增加新的列cbind(要求2个数据的行是一一对应的，也就是最好先排过序)
trade_login_full = cbind(trade_full,tradeId_login_full[,names_use])
#变量重命名
names(trade_full)[3] = 'trade_timestamp'   #重命名时间戳

#6.对某列数据进行切分
#对一列的数据进行切分split，不能按单个切分的方法，调用cSplit可以按直接切分为几列
#splitstackshape::cSplit(数据集,变量名,切分符号)
library(splitstackshape)
last_login$'day' = cSplit(last_login, "hour", " ")$'hour_1'

#7.导出数据
write.csv(trade_login_full,'trade_login_full3.csv',row.names = FALSE)
```

4、结果导出

为了使测试集的预测数据与用于提交的数据进行匹配，按照我们构建特征的思路。就是“按id，按小时”进行匹配，即可。

对导出数据的格式进行查看（通过notepad++或者任何编辑器）。

需要注意的是，R语言中导出的数值很有可能是字符型，那么可以通过如下公式进行转换。

kk$y <- as.numeric(as.character(kk$y))