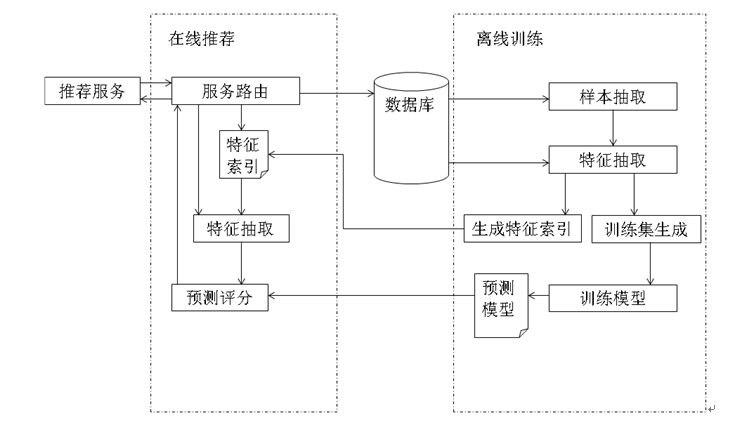
推荐系统搭建全程图文攻略

# 推荐系统架构简介

整体推荐架构图：



1. 推荐整体从数据处理开始，默认数据从关系型数据到每天增量导入到hive，在hive中通过中间表和调用python文件等一系列操作，将数据处理为算法数学建模的入口数据，这里只是模拟一下，所以用一个scala文件产生所有准备数据，并直接load到hive中去做数据处理
2. 数据处理完以后开始数学建模，通过recommend.scala文件对逻辑回归算法的调用，产生模型文件，将三个模型文件拷贝到dubbo项目的响应目录，启动项目，访问测试

**整个过程默认已经有hive环境，intellij idea的环境，并且可以执行scala文件**

流程如下：

Scala文件产生数据🡺load到hive，处理数据🡺recommond.scala调用逻辑回归算法计算模型，生成模型文件🡺将模型文件拷贝到项目制定目录，运行项目🡺浏览器访问测试

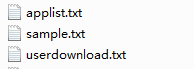
# 二．数据预处理

## 1.创建测试数据

通过DataGenerator类创建数据，参见附件DataGenerator.scala文件，传入参数两个，数据条数和输出目录

比如：100000 E:\推荐系统\资料\hitop

会输出三个文件



## 2.hive建表

真实的生产场景涉及到大概五十张表的字段，这里全部简化流程，直接给出最终的三张表：

应用词表

用户历史下载表

正负例样本表

建表语句：

**应用词表**：

CREATE EXTERNAL TABLE IF NOT EXISTS dim\_rcm\_hitop\_id\_list\_ds

(

hitop\_id STRING,

name STRING,

author STRING,

sversion STRING,

ischarge SMALLINT,

designer STRING,

font STRING,

icon\_count INT,

stars DOUBLE,

price INT,

file\_size INT,

comment\_num INT,

screen STRING,

dlnum INT

)row format delimited fields terminated by '\t';

**用户历史下载表**：

CREATE EXTERNAL TABLE IF NOT EXISTS dw\_rcm\_hitop\_userapps\_dm

(

device\_id STRING,

devid\_applist STRING,

device\_name STRING,

pay\_ability STRING

)row format delimited fields terminated by '\t';

**正负例样本表：**

CREATE EXTERNAL TABLE IF NOT EXISTS dw\_rcm\_hitop\_sample2learn\_dm

(

label STRING,

device\_id STRING,

hitop\_id STRING,

screen STRING,

en\_name STRING,

ch\_name STRING,

author STRING,

sversion STRING,

mnc STRING,

event\_local\_time STRING,

interface STRING,

designer STRING,

is\_safe INT,

icon\_count INT,

update\_time STRING,

stars DOUBLE,

comment\_num INT,

font STRING,

price INT,

file\_size INT,

ischarge SMALLINT,

dlnum INT

)row format delimited fields terminated by '\t';

## 3.load数据

分别往三张表load数据：

**商品词表**：

load data local inpath '/opt/msb/recommender/data/applist.txt' into table dim\_rcm\_hitop\_id\_list\_ds;

**用户历史下载表**：

load data local inpath '/opt/msb/recommender/data/userdownload.txt' into table dw\_rcm\_hitop\_userapps\_dm;

**正负例样本表**：

load data local inpath '/opt/msb/recommender/data/sample.txt' into table dw\_rcm\_hitop\_sample2learn\_dm;

## 4.构建训练数据

### 1.创建临时表

处理数据时所需要的临时表

CREATE TABLE IF NOT EXISTS tmp\_dw\_rcm\_hitop\_prepare2train\_dm

(

device\_id STRING,

label STRING,

hitop\_id STRING,

screen STRING,

ch\_name STRING,

author STRING,

sversion STRING,

mnc STRING,

interface STRING,

designer STRING,

is\_safe INT,

icon\_count INT,

update\_date STRING,

stars DOUBLE,

comment\_num INT,

font STRING,

price INT,

file\_size INT,

ischarge SMALLINT,

dlnum INT,

idlist STRING,

device\_name STRING,

pay\_ability STRING

)row format delimited fields terminated by '\t';

最终保存训练集的表

CREATE TABLE IF NOT EXISTS dw\_rcm\_hitop\_prepare2train\_dm

(

label STRING,

features STRING

)row format delimited fields terminated by '\t';

### 2.训练数据预处理过程

首先将数据从正负例样本和用户历史下载表数据加载到临时表中

INSERT OVERWRITE TABLE tmp\_dw\_rcm\_hitop\_prepare2train\_dm

SELECT

t2.device\_id,

t2.label,

t2.hitop\_id,

t2.screen,

t2.ch\_name,

t2.author,

t2.sversion,

t2.mnc,

t2.interface,

t2.designer,

t2.is\_safe,

t2.icon\_count,

to\_date(t2.update\_time),

t2.stars,

t2.comment\_num,

t2.font,

t2.price,

t2.file\_size,

t2.ischarge,

t2.dlnum,

t1.devid\_applist,

t1.device\_name,

t1.pay\_ability

FROM

(

SELECT

device\_id,

devid\_applist,

device\_name,

pay\_ability

FROM

dw\_rcm\_hitop\_userapps\_dm

) t1

RIGHT OUTER JOIN

(

SELECT

device\_id,

label,

hitop\_id,

screen,

ch\_name,

author,

sversion,

IF (mnc IN ('00','01','02','03','04','05','06','07'), mnc,'x') AS mnc,

interface,

designer,

is\_safe,

IF (icon\_count <= 5,icon\_count,6) AS icon\_count,

update\_time,

stars,

IF ( comment\_num IS NULL,0,

IF ( comment\_num <= 10,comment\_num,11)) AS comment\_num,

font,

price,

IF (file\_size <= 2\*1024\*1024,2,

IF (file\_size <= 4\*1024\*1024,4,

IF (file\_size <= 6\*1024\*1024,6,

IF (file\_size <= 8\*1024\*1024,8,

IF (file\_size <= 10\*1024\*1024,10,

IF (file\_size <= 12\*1024\*1024,12,

IF (file\_size <= 14\*1024\*1024,14,

IF (file\_size <= 16\*1024\*1024,16,

IF (file\_size <= 18\*1024\*1024,18,

IF (file\_size <= 20\*1024\*1024,20,21)))))))))) AS file\_size,

ischarge,

IF (dlnum IS NULL,0,

IF (dlnum <= 50,50,

IF (dlnum <= 100,100,

IF (dlnum <= 500,500,

IF (dlnum <= 1000,1000,

IF (dlnum <= 5000,5000,

IF (dlnum <= 10000,10000,

IF (dlnum <= 20000,20000,20001)))))))) AS dlnum

FROM

dw\_rcm\_hitop\_sample2learn\_dm

) t2

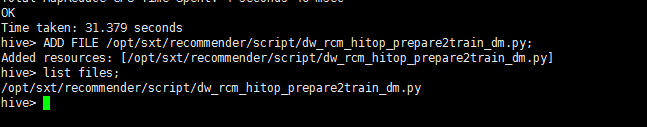
ON (t1.device\_id = t2.device\_id);

然后再利用python脚本处理格式

这里要先讲python脚本加载到hive中

ADD FILE /opt/msb/recommender/script/dw\_rcm\_hitop\_prepare2train\_dm.py;

可以通过list files;查看是不是python文件加载到了hive



目的：将当前浏览的appID与用户历史下载表中的appid形成关联特征

Python文件：

在hive语句中调用python脚本

在hive中使用python脚本处理数据的原理：

Hive会以输出流的形式将数据交给python脚本，python脚本以输入流的形式来接受数据，接受来数据以后，在python中进行一系列的数据处理，处理完毕后，又以输出流的形式交给Hive，交给了hive就说明了就处理后的数据成功保存到hive表中了

INSERT OVERWRITE TABLE dw\_rcm\_hitop\_prepare2train\_dm

SELECT

TRANSFORM (t.\*)

USING 'python dw\_rcm\_hitop\_prepare2train\_dm.py'

AS (label,features)

FROM

(

SELECT

label,

hitop\_id,

screen,

ch\_name,

author,

sversion,

mnc,

interface,

designer,

icon\_count,

update\_date,

stars,

comment\_num,

font,

price,

file\_size,

ischarge,

dlnum,

idlist,

device\_name,

pay\_ability

FROM

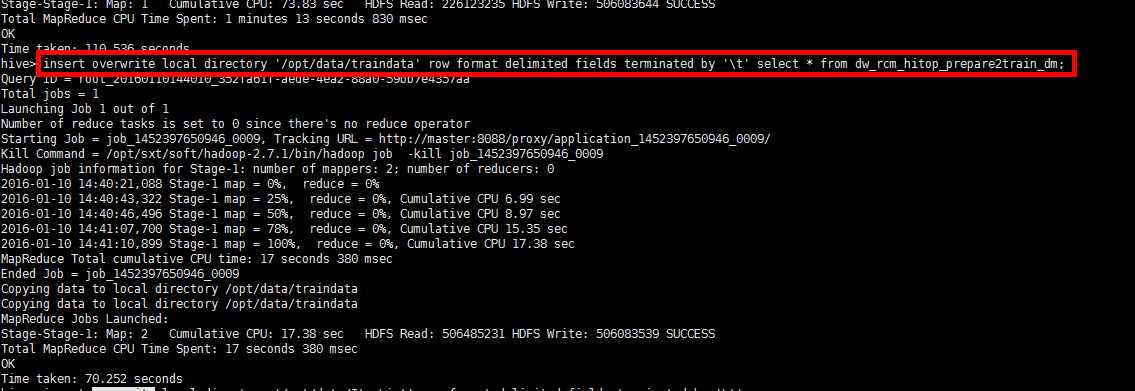
tmp\_dw\_rcm\_hitop\_prepare2train\_dm

) t;

### 3.导出训练数据

将处理完成后的训练数据导出用做线下训练的源数据

`注：这里是将数据导出到本地，方便后面再本地模式跑数据，导出模型数据。**这里是方便演示真正的生产环境是直接用脚本提交spark任务，从hdfs取数据结果仍然在hdfs，再用ETL工具将训练的模型结果文件输出到web项目的文件目录下，用来做新的模型，web项目设置了定时更新模型文件，每天按时读取新模型文件**



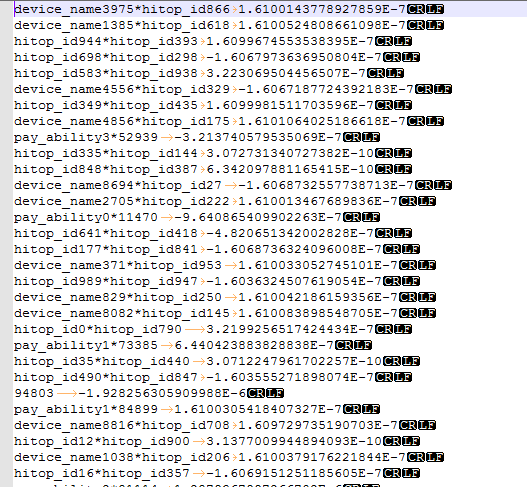
# 三．模型训练

将导出的数据作为输入放在recommend类中执行，参见附件recommond.scala文件，参数为四个，分别是spark执行的模式，输入数据文件路径，分隔符和输出数据路径，注意这里分割是tab键或者是逗号，因为源数据中的分隔符号不统一

这里的输入文件为前面导出的训练数据，地址为linux本地路径/opt/data/traindata/000000\_0

例如：**local E:/推荐系统/资料/hitop/000000\_0 "\t|," E:/推荐系统/资料/hitop/model.csv**

得到结果文件为特征和权重，如图



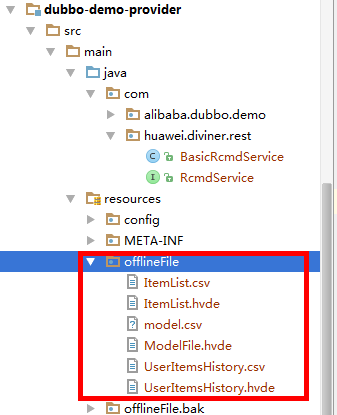
后面的权重小数为科学计数法

# 四．线上模型使用

## 1.拷贝模型文件

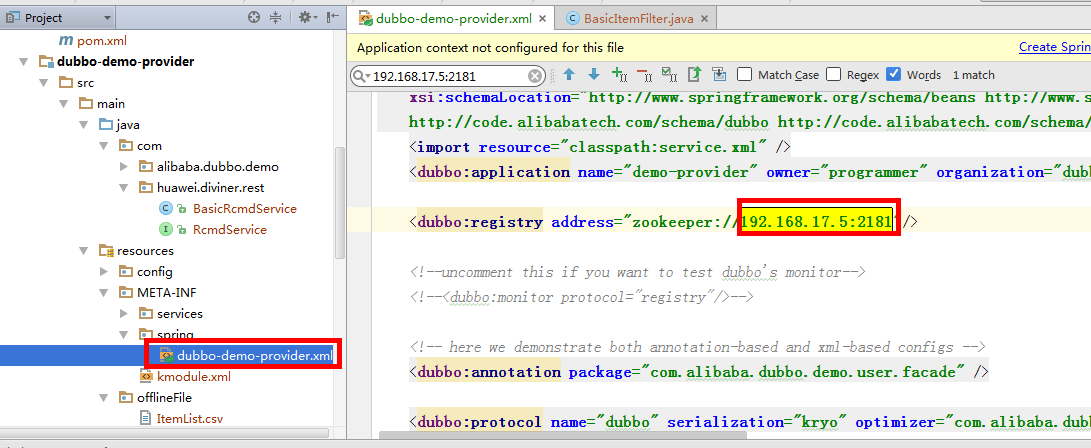
解压项目文件dubbox.rar

将产生的模型文件放到项目资源文件下的offlineFile目录下（如：D:\dubbox\dubbo-demo-provider\src\main\resources\offlineFile），将前面产生的applist.txt改名为ItemList.csv，将userdownload.txt文件改名为UserItemsHistory.csv，看下hvde文件和csv文件的字段描述是否匹配



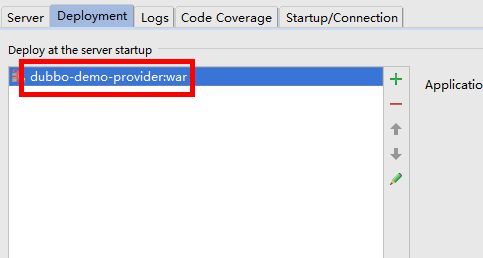
## 2.修改配置文件

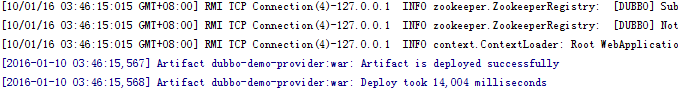
修改资源文件中的zookeeper的ip和端口配置



## 3.启动项目

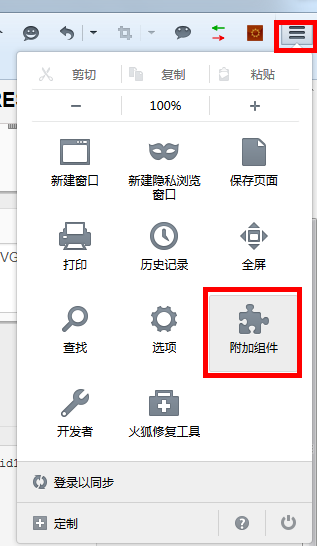
将dubbo-demo-provider项目发布到tomcat，并启动成功



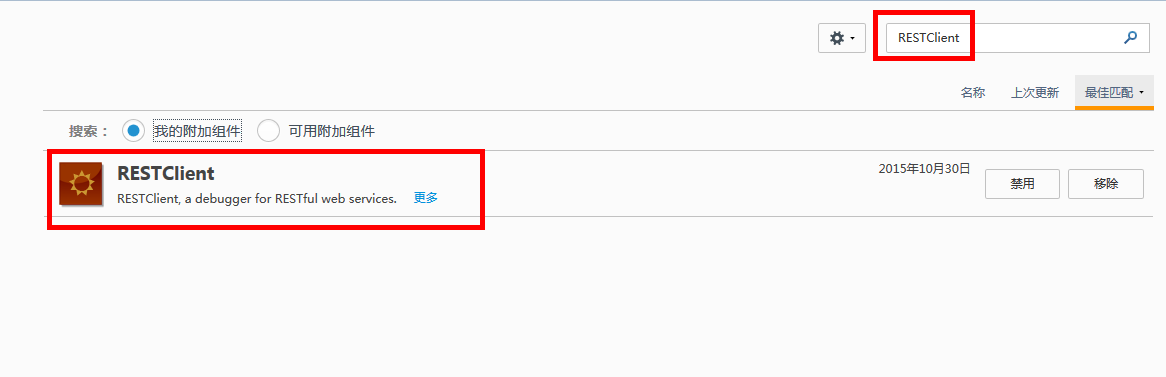


## 4.访问测试

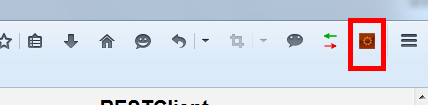
安装火狐浏览器，点击右上角安装附加组件



在右上角输入搜索RESTClient，如果没安装会在可用附件组件中搜到，安装一下，我这里是安装过了



安装完成以后会出现在右上角

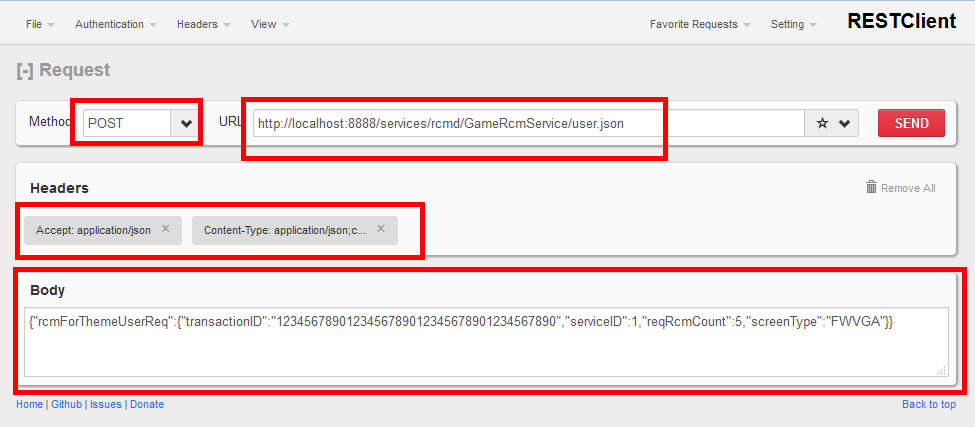


点击此图标

进入RESTClient的页面，并编辑如下参数

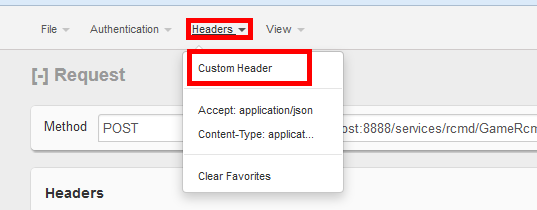
http://localhost:9888/services/rcmd/GameRcmService/user.json

{"rcmForThemeUserReq":{"transactionID":"1234567890","serviceID":"1","reqRcmCount":"5","screenType":"FWVGA"}}

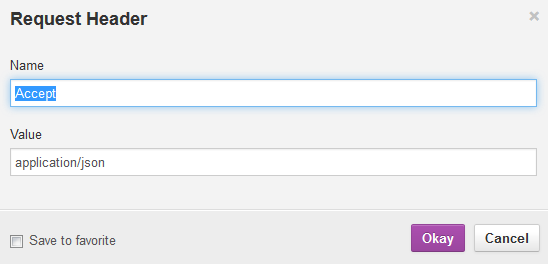


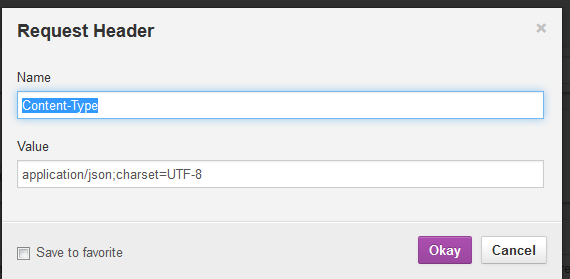
其中head部分参数编辑

点击上面headers部分添加Custom Header

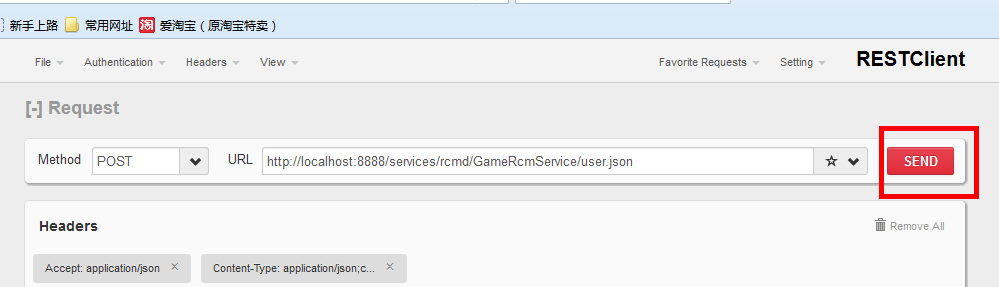


分别添加如下两个Header

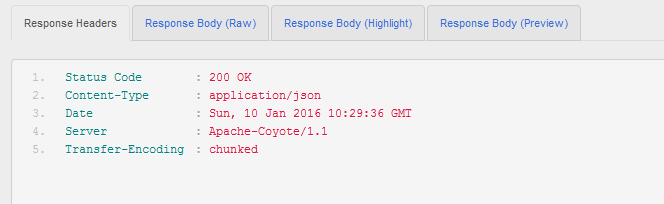




点击右边send按钮



返回状态



返回结果

到此为止整个推荐系统部署完成，如果有web展示端可以通过restful接口访问线上服务，将推荐的信息在web端展示

