

**Alibaba**

**天池实践手册**

前言

本手册从天池选手的角度出发，通过具体的案例场景引导用户使用御膳房平台进行数据开发、数据建模以及预测。

本手册将从以下几个步骤来描述如何使用御膳房进行数据分析和建模：

* 课题介绍
* 环境和账号
* 获取数据
* 实现步骤

由于是基于特定案例引导，本文展示的仅仅是ODPS和算法功能的冰山一角。关于详细的ODPS和算法功能，可以通过在线帮助手册了解更多。

请参考[ODPS在线帮助手册](https://help.aliyun.com/product/8314999_odps.html?spm=5176.750001.2.16.IpCSQk)。 主要参考模块：基本介绍、常用命令、SQL语法，MR语法。

关于MR的使用和配置方法请参考[御膳房用户手册](http://setting.tenant.yushanfang.com/portal/help/index.html?spm=0.0.0.0.rJSSxb)。

内容

[前言 I](#_Toc437941912)

[内容 II](#_Toc437941913)

[1 课题介绍 1](#_Toc437941914)

[1.1 课题介绍 1](#_Toc437941915)

[1.1.1 数据说明 1](#_Toc437941916)

[1.1.2 输出要求 2](#_Toc437941917)

[1.1.3 评估指标 2](#_Toc437941918)

[1.2 解题思路 3](#_Toc437941919)

[2 环境、账号 4](#_Toc437941920)

[2.1 登录御膳房 4](#_Toc437941921)

[3 获取数据 6](#_Toc437941922)

[3.1 项目首页 6](#_Toc437941923)

[3.2 查看数据 8](#_Toc437941924)

[4 实现步骤 9](#_Toc437941925)

[4.1 数据开发平台的使用（以下也称IDE） 10](#_Toc437941926)

[4.1.1 准备工作 10](#_Toc437941927)

[4.1.2 数据分析 13](#_Toc437941928)

[4.1.3 特征工程 19](#_Toc437941929)

[4.2 PAI平台 45](#_Toc437941930)

[4.2.1 PAI简介 45](#_Toc437941931)

[4.2.2 数据抽样 47](#_Toc437941932)

[4.2.3 建模和评估 52](#_Toc437941933)

[4.2.4 PAI命令及模型同步 62](#_Toc437941934)

[4.2.5 线上测试集的训练和预测 63](#_Toc437941935)

[名词解释 66](#_Toc437941936)

[修订历史 67](#_Toc437941937)

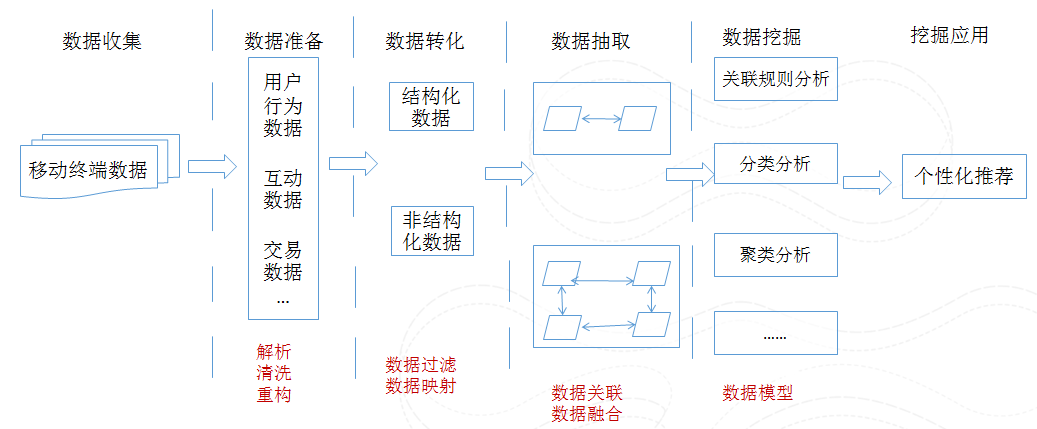
# 课题介绍

## 背景

随着移动电商业务的快速发展，用户对移动端网络的访问是随时随地的，具有更丰富的场景数据。如何在这些巨大的信息数据中快速挖掘出对我们有用的信息已成为当前急需解决的问题。

对于移动电商来说，可以运用这些有用的信息以及个性化技术的手段，对用户进行一系列的个性化推荐，帮助用户从这些网络过量的信息里面筛出他所需要的信息，提升移动端用户的用户体验，提升个性化推荐引导的成交率，从而达到精准营销(Precision marketing)的目的。

如下是一个简单的个性化推荐的流程：



## 课题介绍

本课题以阿里巴巴移动电商平台的真实用户-商品行为数据为基础，同时提供移动时代特有的位置信息。您需要通过大数据和算法构建面向移动电子商务的商品推荐模型，挖掘数据背后丰富的内涵，为移动用户在合适的时间、合适的地点精准推荐合适的内容。

在真实的业务场景下，我们往往需要对所有商品的一个子集构建个性化推荐模型。在完成这件任务的过程中，我们不仅需要利用用户在这个商品子集上的行为数据，往往还需要利用更丰富的用户行为数据。

如下是一个真实的业务场景：

给出一定量用户在时间段11月18日~12月18日内的移动端行为数据（D），

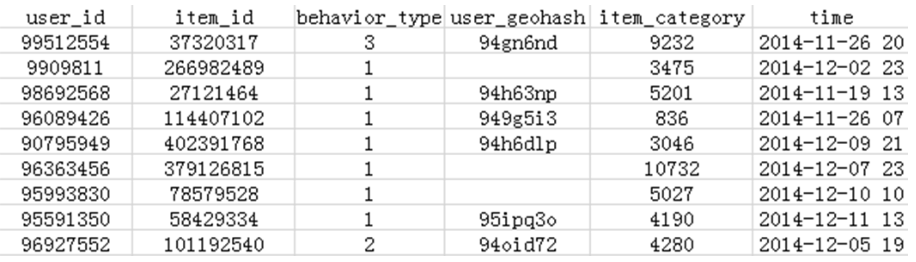
需要预测12月19日用户对商品子集（P）的购买数据。具体的数据说明下面将会详细介绍。

### 数据说明

提供的数据包含两个部分。第一部分是用户在商品全集上的移动端行为数据（D），表名为tianchi\_mobile\_recommend\_train\_user，包含如下字段：

|  |  |  |
| --- | --- | --- |
| **字段** | **字段说明** | **提取说明** |
| user\_id | 用户标识 | 抽样&字段脱敏 |
| item\_id | 商品标识 | 字段脱敏 |
| behavior\_type | 用户对商品的行为类型 | 包括浏览、收藏、加购物车、购买，对应取值分别是1、2、3、4。 |
| user\_geohash | 用户位置的空间标识，可以为空 | 由经纬度通过保密的算法生成 |
| item\_category | 商品分类标识 | 字段脱敏 |
| Time | 行为时间 | 精确到小时级别 |

D集合的记录显示如下：



每一行代表了用户 user\_id 对属于分类 item\_category 的物品 item\_id 在 time 这个时间于地点 user\_geohash 发生了交互，交互类型是 behavior\_type 。behavior\_type 包括浏览、收藏、加购物车、购买，对应取值分别是1、2、3、4。

给出的一共包含31天的交互数据，最后要预测第32天有哪些user会购买哪些item。

第二个部分是商品子集（P）,表名为tianchi\_mobile\_recommend\_train\_item，包含如下字段：

|  |  |  |
| --- | --- | --- |
| **字段** | **字段说明** | **提取说明** |
| item\_id | 商品标识 | 抽样&字段脱敏 |
| item\_ geohash | 商品位置的空间标识，可以为空 | 由经纬度通过保密的算法生成 |
| item\_category | 商品分类标识 | 字段脱敏 |

这部分的数据，是 tianchi\_mobile\_recommend\_train\_user表中出现的物品item\_id的一个子集。官方的第32天的会发生购买的 user, item 中的item都是这个数据表中的item，所以我们预测的 user, item 需要根据这个文件来过滤掉一些结果。

训练数据包含了抽样出来的一定量用户在一个月时间（11.18~12.18）之内的移动端行为数据（D），评分数据是这些用户在这个一个月之后的一天（12.19）对商品子集（P）的购买数据。您需要使用训练数据建立推荐模型，并输出用户在接下来一天对商品子集购买行为的预测结果。

### 输出要求

完成用户对商品子集P的购买预测之后，需要将结果放入指定格式的数据表（非分区表）中，要求结果表名为：tianchi\_mobile\_recommendation\_predict，包含user\_id和item\_id两列（均为string类型），要求去除重复。

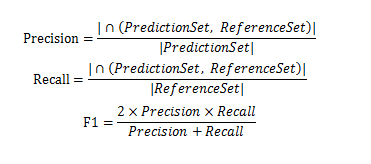
表名：tianchi\_mobile\_recommendation\_predict

结果表样例：

|  |  |  |
| --- | --- | --- |
| **字段** | **字段类型** | **示例** |
| User\_id | String | 100001 |
| Item\_id | String | 12900 |

### 评估指标

比赛采用经典的精确度(precision)、召回率(recall)和F1值作为评估指标。具体计算公式如下：



其中PredictionSet为算法预测的购买数据集合，ReferenceSet为真实的答案购买数据集合。我们以F1值作为最终的唯一评测标准。

## 解题思路

显然，该问题可以转为一个二分类问题：某个用户对某个商品是否会购买？

* 分类方法：二分类

最终只有两种结果：1：购买；0：不购买

* 样本选取

样本选取—>有交互（即：用户对商品有行为，包括浏览、收藏、加购、购买）的P子集。

分析在哪一天交互的样本可能会在19号购买？16号，17号，18号？

可以基于18号的购买数据，分析16、17号的行为数据在18号购买的概率。

* 特征

分析什么样的用户在什么时候对什么样的商品有过什么样的操作之后，可能会在19号购买？比如：是否曾经买过？商品销量？加购物车？

* 模型

分类、回归。如LR、RF、GBDT等。

下面我们将围绕这个课题，描述如何在御膳房平台完成一系列的实现步骤。

# 环境&账号

1. 登录御膳房首页[www.yushanfang.com](http://www.yushanfang.com)。点击右上角的<登录>，输入已经注册好的淘宝、阿里云账号，如下图所示:



图1御膳房登录界面

注：

* 如果您还没有淘宝账号或者阿里云账号，可以在上图点击<注册淘宝账号>或<注册阿里云账号>完成注册。
* 如果已经注册，也可以使用手机号码或邮箱登录。

1. 填写注册信息。

首次登录，需填写注册信息。

单击<登录>，页面跳转到用户信息注册页面。请使用真实信息，完成所有必填项目（手机号、Email、联系人为必填项，注：手机号应与淘宝账号绑定），阅读并勾选“同意御膳房协议”。



图2用户信息注册

1. 点击<下一步>后，您的账号已经完成了在御膳房的注册。

# 获取数据

当用户登录后，即可进入天池大赛授予的特定项目，进行数据开发/处理/建模工作。

下面我们先来看看如何获取和查看数据。

## 项目首页

可以从两个入口进入“项目”。

1. 点击[御膳房 –> 数据中心-> 工作台]，用户可以在这个界面查看自己对应的资产，项目名称以及角色。点击项目名，即可进入项目。

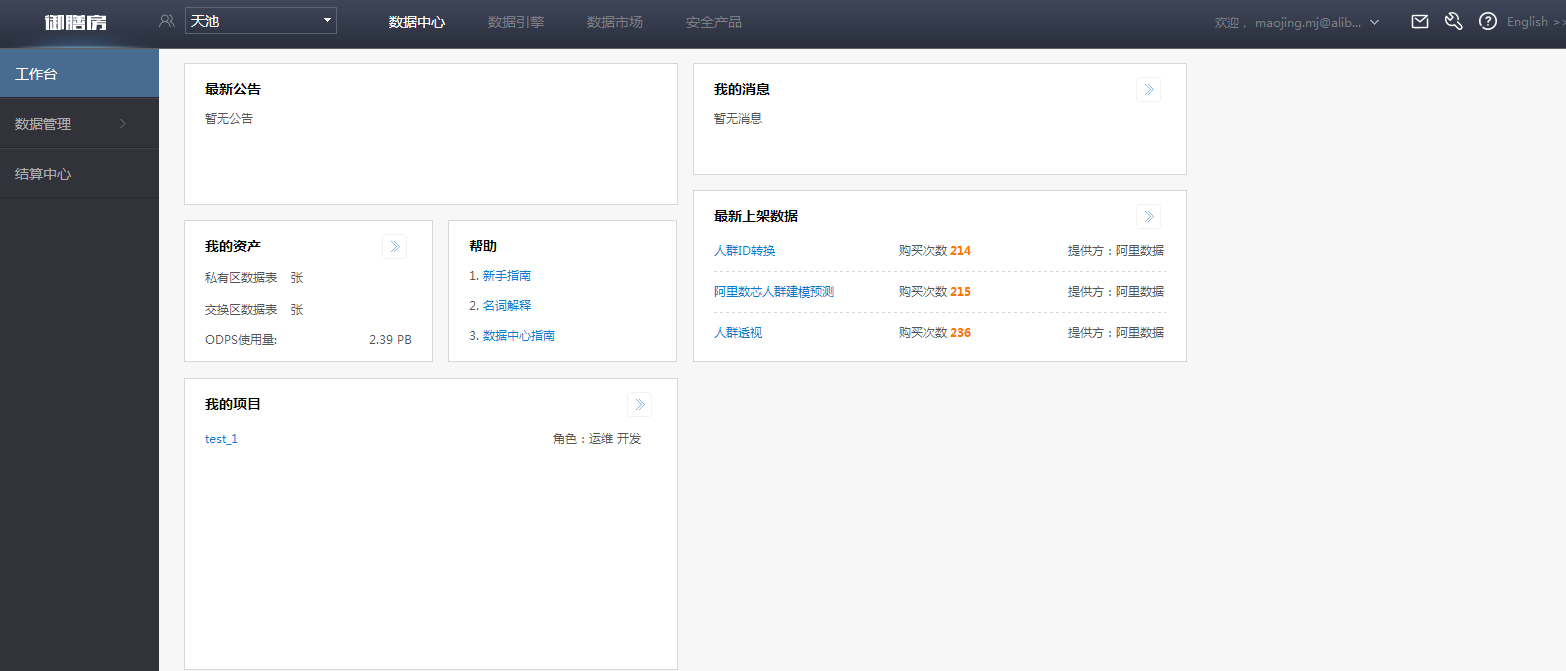


图3- 进入项目入口（1）

1. 点击[数据引擎-> 私有区]，可以看到项目列表，点击项目名称后进入项目。



图4 进入项目入口（2）



图5 进入项目入口（2）

1. 进入项目后，界面如下：



图6 进入项目

## 查看数据

1. 进入项目首页后，点击[项目数据->数据资源->组织内申请的数据]，即可查看组织内授权的表以及所属项目（来源项目）信息：

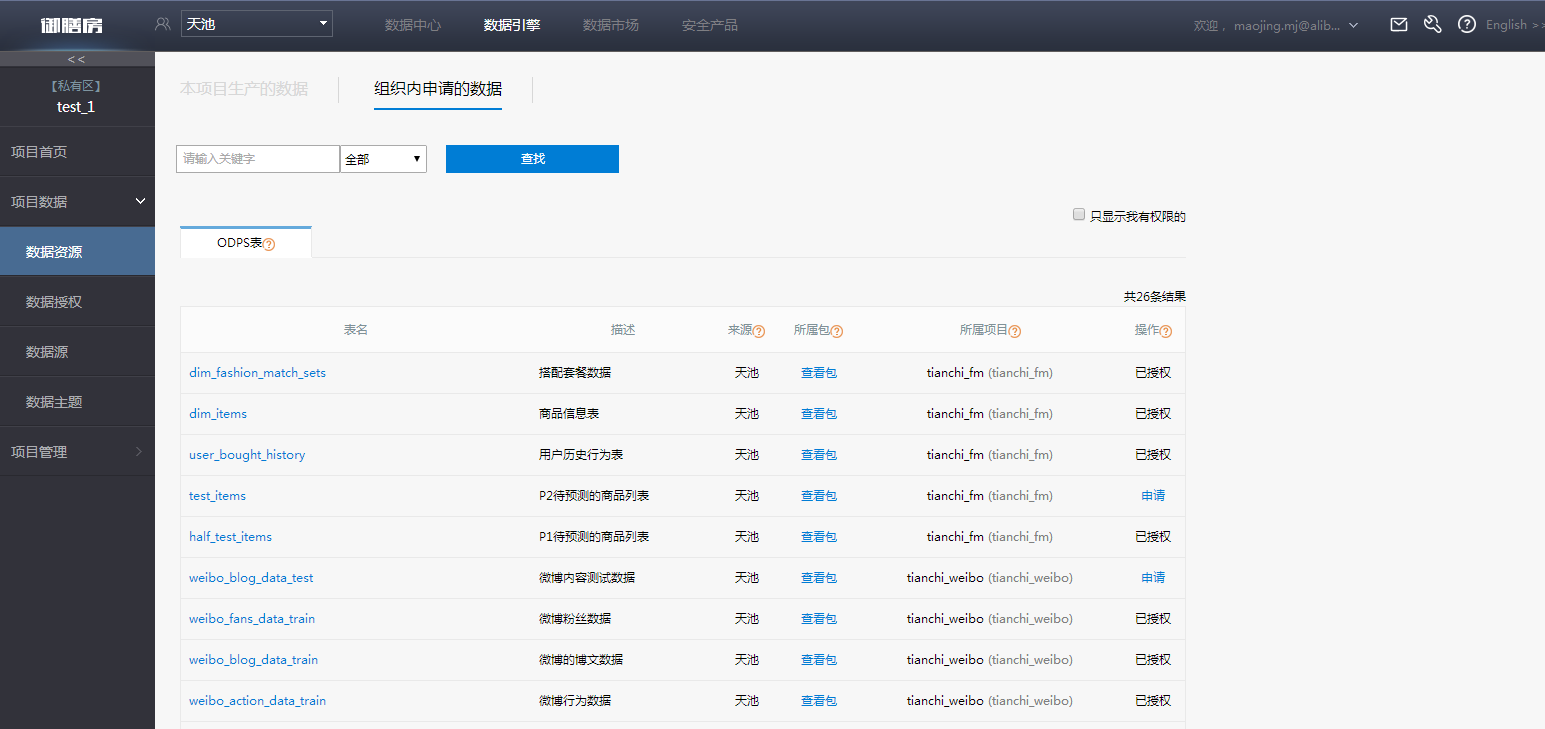


图7 查看odps表

注：

在数据开发工作台使用具体的表的时候，必须加上所属项目前缀。如：用户行为表tianchi\_mobile\_recommend\_train\_user，查看到其所属项目为tianchi\_data，则在数据开发工作台中引用时，需这样：tianchi\_data. tianchi\_mobile\_recommend\_train\_user。

# 实现步骤

接下来我们将介绍如何完成上述课题，主要包含以下几个主要的步骤：

1. 在数据开发平台利用ODPS SQL进行数据分析:

* 统计总的用户行为记录
* 统计所给数据中每天各种行为类型的记录数，了解每天会有多少浏览、收藏、加购、购买，观察是否有波动。
* 锁定在P上有过交互的用户行为。
* 分析前一天加购、收藏、浏览对后一天购买的影响。取12/18为后一天，拿12/17的数据来验证该设想。
* 计算评估指标。
* 提交结果表。

1. 特征工程：可以使用ODPS SQL进行特征提取，也可以使用MR/UDF/GRAPH进行特征工程。这里主要介绍使用UDF进行特征提取，MR/GRAPH的用法与之类似。

* Eclipse工具的下载和安装
* 新建项目(Maven Project)
* 程序开发以及提交
* 利用UDF在数据开发工作台抽取特征，详细步骤如下：
* 样本选取：（正样本）分别以2014/12/17的购买，12/18的购买来构建验证集、测试集。
* 特征工程：利用已经构建的UDF函数来统计user-item对的点击，收藏，加购，购买次数。
* 结合特征构建验证集和测试集，包含训练和测试。
* 检查正负样本比例。
* 转换特征的数据格式。

1. PAI平台的使用

* 数据抽样
* 负样本抽样1/15
* 样本合并
* 建模和评估，包括：
* 逻辑回归
* 随机森林
* 预测和评估
* PAI命令的执行以及模型同步
* 线上测试集的训练和预测

## 数据开发平台的使用（以下也称IDE）

用户在数据开发工作台中以编写和运行基于ODPS 的SQL、MapReduce、UDF和Graph。

其中，SQL、MapReduc和UDF的编写语法参考ODPS文档，请点击[这里](https://help.aliyun.com/product/8314999_odps.html?spm=5176.750001.2.18.woTmrI)。

MapReduc、UDF和Graph的配置请参考[这里](http://setting.tenant.yushanfang.com/portal/help/doc.html?file=MrUdfLocalDev)。

### 准备工作

1. 进入项目首页：



图8 项目首页

1. 点击“数据开发”，即可进入数据开发工作台：

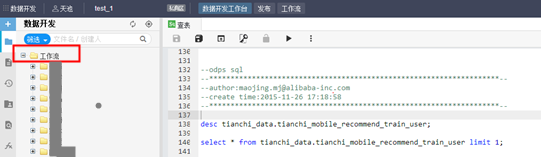


图9 数据开发工作台

为了方便管理代码，可以新建一个文件夹，把自己的代码都放在这个文件夹中（天池历届比赛中，通常都是一个队伍所有成员共享一个project）。

1. 新建文件夹。

右键点击“工作流”并点击<+文件夹>，新增一个工作文件夹,输入文件夹名称，选择文件夹位置，提交即可，如：

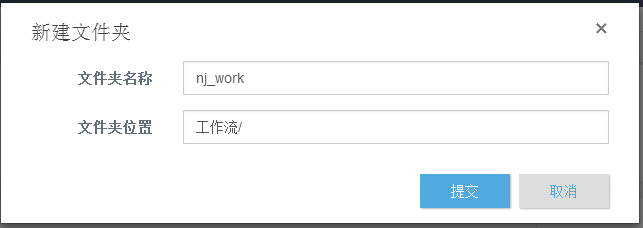


图10 新建文件夹

1. 新增工作流节点。

在数据开发工作台左侧导航选择“数据开发”。

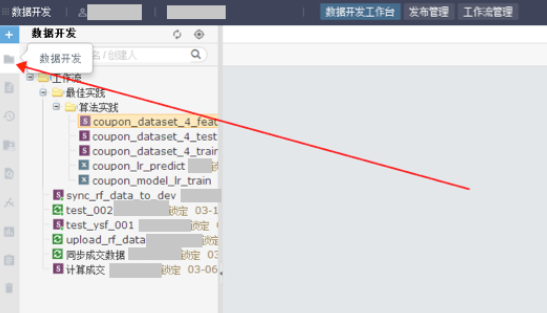


图11 数据开发图标

点击“+”图标，选择“工作流节点”。

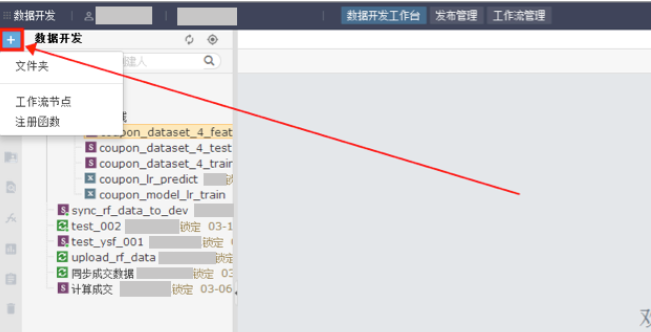


图12 “+”图标

右键点击文件夹，比如“nj\_work”，并点击<+工作流节点>，在页面配置一个ODPS SQL节点如下：

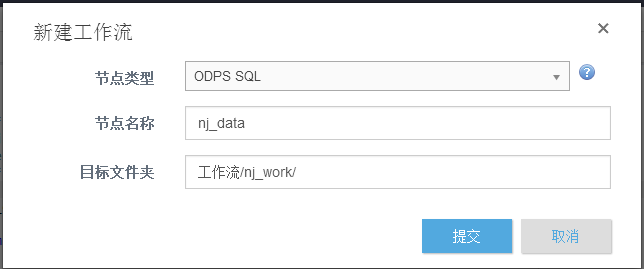


图13 新增工作流节点

点击<提交>，此时在新的页面上就可以编写ODPS SQL代码了。现在我们可以查一下表tianchi\_mobile\_recommend\_train\_user的表结构，如下：

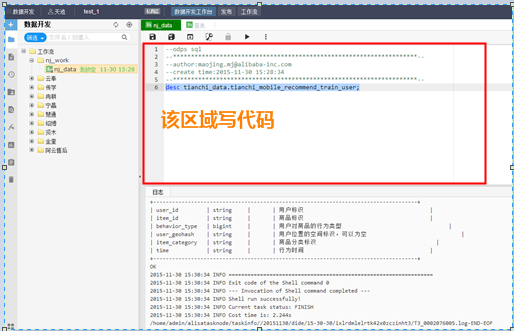


图14 运行ODPS命令

点击运行代码（或按F8键）。

注意：

对于选手而言，所有的操作只需要在开发环境运行，也就是说，不需要提交数据到生产环境，也不需要进行调度和发布。

如上述介绍，在数据开发工作台内运行的表或资源都属于开发环境，如果需要发布到生产环境，则需要通过发布管理模块将工作流节点调度并发布，方可同步到生产环境。具体描述请参考御膳房最佳实践之数据同步章节（<http://setting.tenant.yushanfang.com/portal/help/index.html?spm=0.0.0.0.6FFEVc>）。

### 数据分析

下面我们根据上面的课题在IDE上先进行相关的数据分析。

给出一定量用户在11.18~12.18之内的移动端行为数据（D），需要预测12.19对商品子集（P）的购买数据。行为分为四种：1(浏览)，2(收藏)，3(加购物车)，4(购买)。

首先，由于赛题数据来源于其他project(项目空间)，为方便起见及后续在算法平台(PAI)中使用，可以考虑将项目表拷贝到本队伍所在项目空间。

SQL如下：

Create table tianchi\_mobile\_recommend\_train\_user as

select \* from tianchi\_data. tianchi\_mobile\_recommend\_train\_user;

Create table tianchi\_mobile\_recommend\_train\_item as

select \* from tianchi\_data. tianchi\_mobile\_recommend\_train\_item;

下面我们就可以在IDE利用ODPS SQL进行相关的数据统计：

* 统计D集合的数量：

select count (\*) from tianchi\_mobile\_recommend\_train\_user;

---包含了58亿用户的行为记录。

* 统计针对P子集的用户行为数据 ：

create table t\_mj\_p\_user as

select

 t.\*, b.item\_geohash from tianchi\_mobile\_recommend\_train\_user t

  join   tianchi\_mobile\_recommend\_train\_item b on b.item\_id=t.item\_id and b.item\_category=t.item\_category;

select count(1) from t\_mj\_p\_user;

---10.7亿

* 统计所给数据中每天各种行为类型的记录数，了解每天会有多少浏览、收藏、加购、购买，观察是否有波动：

select

substr(time,1, 10) as day

,behavior\_type

,count(1) cnt

from

tianchi\_mobile\_recommend\_train\_user

group by

substr(time,1, 10)

,behavior\_type;

从用户的购买行为动机和流程来看，人们通常在购买时会经历浏览→收藏/加到购物车→下单付款这一系列动作，有的可能会搜索后就直接购买，有的可能会操作得多一点，反复挑选和比较。如果说用户都已经把宝贝加到购物车了，那么购买的概率会不会更高？最近加到购物车的是不是应该更有可能被购买？基于这个设想，下面我们示范下如何在数据平台上进行这个分析，并完成结果提交。这是一个通过简单规则来判断购买从而提交结果的示例，较为简单粗暴，但可以让我们先熟悉下从拿到课题到提交结果的整个流程，以及初尝结果验证设想。

1. 由于所给数据tianchi\_mobile\_recommend\_train\_user为用户在所有商品类目上的行为，而要预测的是在商品子集P上的购买，故先锁定在P上有过交互的用户行为：

create table t\_mj\_p\_user as

select

 t.\*, b.item\_geohash from tianchi\_mobile\_recommend\_train\_user t

  join   tianchi\_mobile\_recommend\_train\_item b on b.item\_id=t.item\_id and b.item\_category=t.item\_category;

select count(1) from t\_mj\_p\_user;

------ 1074139328 (用户对P子集商品的交互记录数)

select count(1) from t\_mj\_p\_user where behavior\_type=4;

-----9309353(其中对P子集商品有购买的记录数)

select count(1) from

(select

user\_id

,item\_id

,count(1) cnt

from

t\_mj\_p\_user

group by

user\_id

,item\_id

)t;

-----202430141 [user,item]对

1. 分析前一天加购、收藏、浏览对后一天购买的影响。取12/18为后一天，拿12/17的数据来验证该设想。

create table t\_mj\_u\_i\_1day as

select

user\_id

,item\_id

,case when substr(time,1, 10)='2014-12-17' and behavior\_type=1 then 1 else 0 end as is\_1day\_view

,case when substr(time,1, 10)='2014-12-17' and behavior\_type=2 then 1 else 0 end as is\_1day\_fav

,case when substr(time,1, 10)='2014-12-17' and behavior\_type=3 then 1 else 0 end as is\_1day\_cart

,case when substr(time,1, 10)='2014-12-17' and behavior\_type=4 then 1 else 0 end as is\_1day\_buy

,case when substr(time,1, 10)='2014-12-18' and behavior\_type=4 then 1 else 0 end as is\_buy

from

t\_mj\_p\_user;

计算评估指标：

select

precision

,recall

,2\*precision\*recall/(precision+recall) as f1

from

(select

hit\_cnt/p\_cnt as precision

,hit\_cnt/r\_cnt as recall

from

(select

sum(if(a.user\_id is not null and a.item\_id is not null and b.user\_id is not null and b.item\_id is not null, 1, 0)) as hit\_cnt --命中

,if(sum(a.r\_cnt) is null, 0, sum(a.r\_cnt)) as r\_cnt --实际

,if(sum(b.p\_cnt) is null, 0, sum(b.p\_cnt)) as p\_cnt --预测

from

(

select user\_id,item\_id, count(1) as r\_cnt

from

t\_mj\_u\_i\_1day

where

is\_buy=1

and user\_id is not null

and item\_id is not null

group by user\_id, item\_id

) a

full outer join

(

select user\_id,item\_id, count(1) as p\_cnt

from

t\_mj\_u\_i\_1day

where

is\_1day\_cart=1

and user\_id is not null

and item\_id is not null

group by user\_id, item\_id

) b

on a.user\_id = b.user\_id and a.item\_id = b.item\_id

)t

)tb;

返回如下结果,可以看到F1=2.99%，召回率高于准确率，均有较大提升空间：

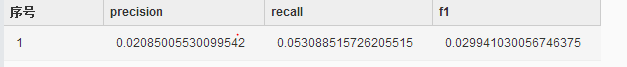


图15 返回结果

1. 提交结果：

以比赛要求的12/19为预测日期，将用户在12/18对商品子集P加购最多的商品作为预测结果tianchi\_mobile\_recommendation\_predict，生成该表即完成了提交，注：大赛进行的时候，天池的评测系统会自动去扫描各project中指定命名的表（本例中即：tianchi\_mobile\_recommendation\_predict）并评测。

drop table if exists tianchi\_mobile\_recommendation\_predict;

create table tianchi\_mobile\_recommendation\_predict

as

select

user\_id

,item\_id

from

(

select

user\_id

,item\_id

,row\_number() over(partition by user\_id order by num desc) as rank

from

(

select

a.user\_id

,a.item\_id

,a.num

from

(

select

user\_id

,item\_id

,count(1) as num

from

tianchi\_mobile\_recommend\_train\_user

where

substr(time, 1, 10)='2014-12-18'

and behavior\_type=3

group by

user\_id

,item\_id

) a

join

(

select distinct

item\_id

from

tianchi\_mobile\_recommend\_train\_item

) b

on a.item\_id = b.item\_id

where

b.item\_id is not null

) c

) d

where rank<=1;

count tianchi\_mobile\_recommendation\_predict;-----166068（与上述数据分析时所统计的每日购买数据量相仿）

基于这个提交的结果，可以拿到这样一个分数：准确率=1%， 召回率=0.9%，F1=0.96%。

说明：

上述SQL语法涉及到的主要知识点包括：

* 创建表/删除表
* 聚合函数count
* 字符串函数substr
* Join语法
* row\_number函数
* case when/if条件函数

关于SQL的更多语法，请参考ODPS用户手册<http://help.aliyun.com/document_detail/odps/SQL/summary.html>。

### 特征工程

经过一番数据探查和分析，可大致确定深入的解题方案。假设考虑采取“按用户-商品对抽取特征，再进行分类”这样一种方案，那么接下来就要抽取特征。假设我们的特征使用近1天，3天，7天，10天的用户行为数量（点击数，浏览数…），我们可以直接使用sql来抽取特征，或者使用MapReduce（简称MR），再或者使用UDF(user-defined function)，这些都可以在数据开发下添加相应的节点来完成。以下演示在平台上如何配置和使用MR/UDF/Graph实现特征抽取，以UDF为例详细操作，MR同理，只是一两处有些许差别，相应的地方也会特别注明。

#### 工具下载

请前往http://www.eclipse.org/m2e/ 下载Eclipse 插件。

请前往http://maven.apache.org/ 下载Maven 插件，建议下载3.2.5 版本。

下载完毕并解压安装后，打开Eclipse 插件，会看到以下界面：

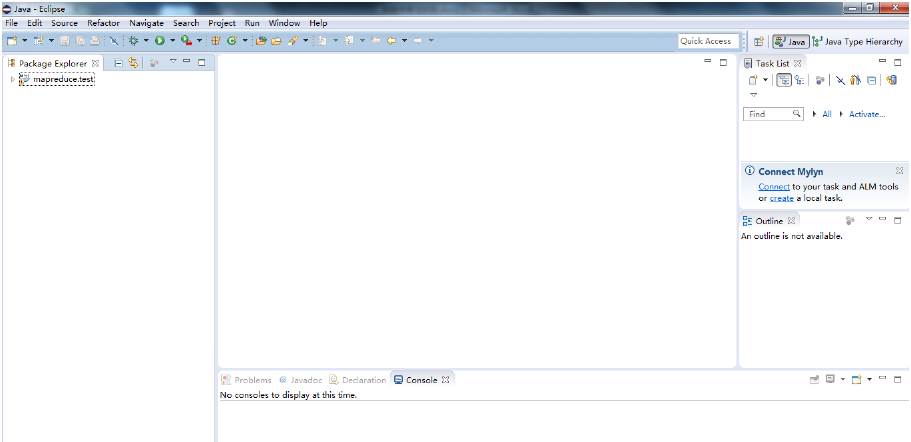


图16 Eclipse主页面

#### 新建项目

MR&UDF开发工具全部基于Maven开发，因此使用此工具前应该对maven的概念及基本使用有一定了解。使用IDE来开发MR&UDF，还要了解如何在IDE中使用相关maven插件。

MR&UDF开发工具通过maven archetype机制新建程序项目，御膳房提供base-mapreduce-archetype、base-udf-archetype和base-graph-archetype三个archetypes，通过远程repository可直接使用。

* 添加远程repository

注：请确保机器是上装有maven，并且Eclipse中已安装M2Eclipse插件。

1. 如果您使用的是MAC 系统，请在Eclipse 中依次点击 <Preferences -> Maven ->Arthetypes> ，在打开的对话框中点击 <Add Remote Catalog...> 按钮；如果您使用的是WINDOWS 或其他系统，请在Eclipse 中依次点击<Window ->Preferences -> Maven -> Arthetypes> ，在打开的对话框中点击 <Add Remote Catalog...> 按钮。

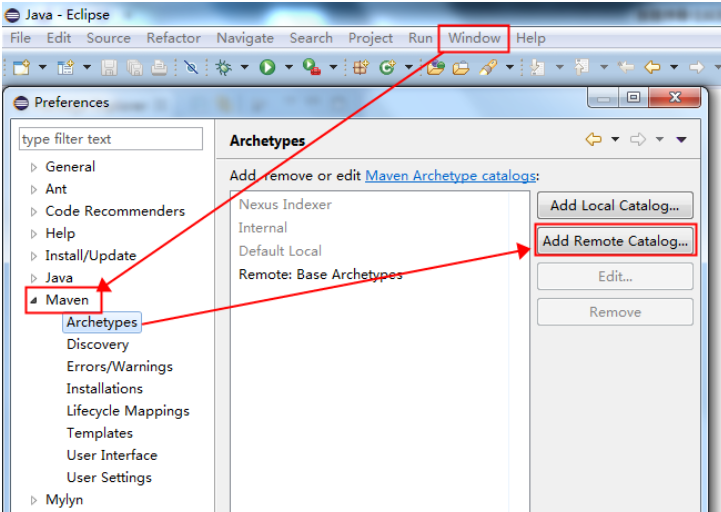


图17添加 Maven Arthetypes

1. 在打开的对话框中， Catalog File 填入http://maven.sdk.de.yushanfang.com/SNAPSHOT， Description 填入 Base Archetypes，然后一路点击 <OK> 就完成了远程repository 添加。

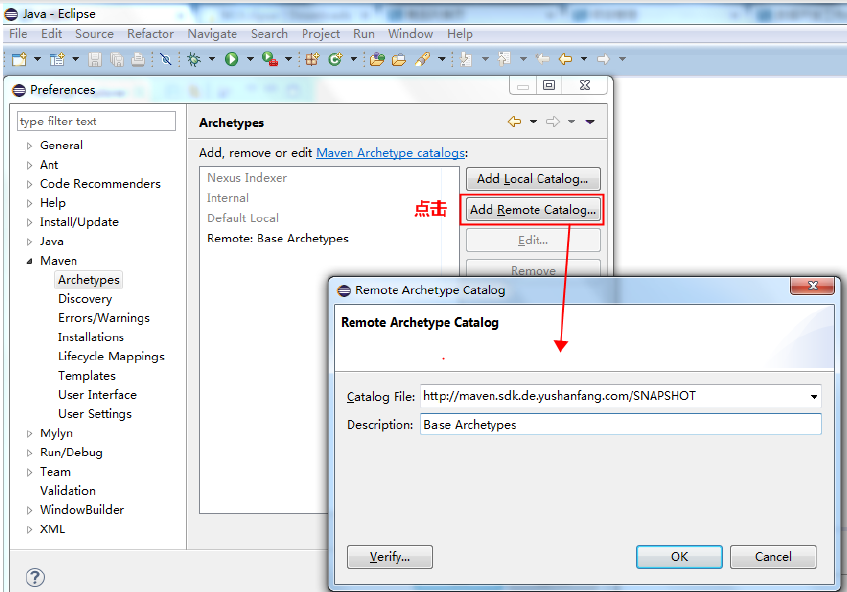


图18 设定远程目录

* 新建项目

1. 如同新建一般maven项目一样，在Eclipse中依次点击< File -> New -> Project...>，在打开的对话框中选择[Maven -> Maven Project]:

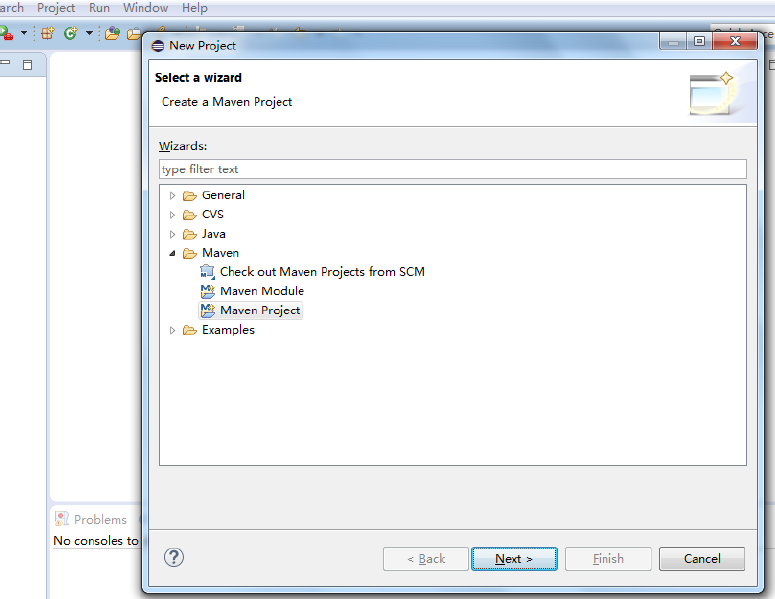


图19 新建Maven 项目

1. 一路点击 <Next> ，在 [Select an Archetypes] 界面， Catalog 选择 Base Archetypes ，并选中 Include snapshot archetypes 复选框，此时可以看到列表中出现base-graph-archetype、base-mapreduce-archetype和base-udf-archetype三个archetypes，选择需要创建的程序类型，点击<Next>。

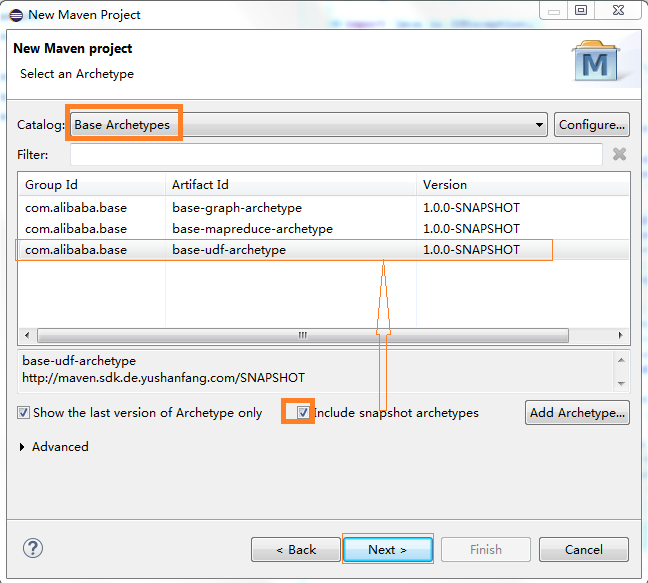


图20 新建base-udf-archetype

1. 以base-udf-archetype为例，点击<Next> 后，会进入常规的maven项目初始化配置界面。注意Properties列表中会出现几个御膳房特有的配置项，分别是：

* baseId - 用户在御膳房上的用户标识
* projectId - 用户要创建程序到哪个御膳房项目ID
* token – 项目证书
* idePath - 程序将要上传到IDE的哪个工作目录/文件夹下，需要文件夹已经在IDE中创建好，例如“工作流/宁晶”。

获取这些配置项的操作步骤如下：

1. 点击<我的账号>，如下图：



图21 获取我的账号

1. 点击证书管理，看到项目ID（对应到 projectId）、用户标识（对应到baseId）、项目证书（对应到token）三个信息：



图22 证书管理

1. 配置示例如下图：（其中，Group ID和Artifact Id， idePath， functionName为自定义，注：命名中请勿使用-字符，否则会报错；另外3个ID项按照上述方式获取并填入）。若为UDF, 相应的配置会保存于配置文件src/main/resources/META-INF/base.udf.xml中.

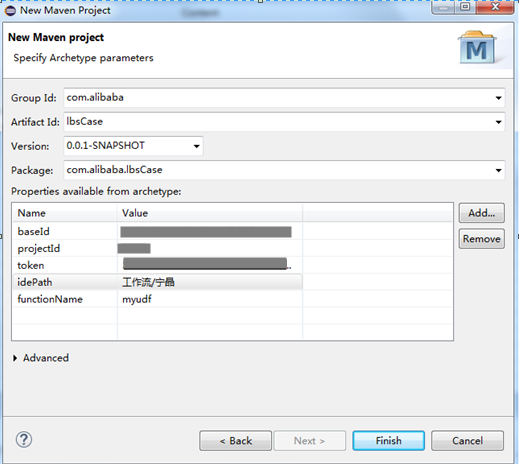


图23 设定archetype参数

注：这里将idePath自定义配置成：工作流/宁晶（再次强调下，这个文件需要在IDE中先建好，否则后续会报错），那么当Maven项目开发完，在eclipse直接提交后，提交的节点和对应源代码就会自动在IDE 的该目录下创建。

1. 填写完成后，点击 <Finish> 就会完成项目创建，随后即可在Eclipse的项目管理器中看到新建的项目了。

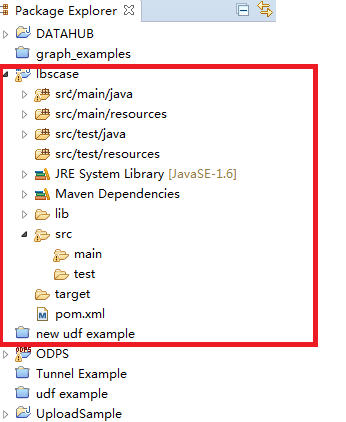


图24 配置好的项目

#### 程序开发及提交

##### 项目结构

通过archetypes新创建的程序具有完整的maven project骨架（如初始化好的pom文件等）、脚手架代码（用户可直接在其中实现自己的业务）和示例代码；其中mapreduce程序还带有本地运行环境和mock数据，用于通过local模式运行mapreduce程序，以便本地调试。

下面我们将介绍UDF程序开发步骤。

##### UDF程序开发步骤

1. 项目结构

一个新创建的udf程序项目结构如下：（与mapreduce程序类似，但是udf程序目前不支持本地调试）

├── lib

│ ├── bouncycastle.provider-1.38-jdk15.jar

│ ├── commons-codec-1.9.jar

│ ├── commons-io-2.4.jar

│ ├── commons-lang-2.4.jar

│ ├── commons-logging-1.1.1.jar

│ ├── gson-2.2.4.jar

│ ├── jackson-core-asl-1.9.13.jar

│ ├── jackson-mapper-asl-1.9.13.jar

│ ├── javacsv-2.0.jar

│ ├── odps-commons-internal-0.13.2.jar

│ ├── odps-sdk-commons-0.13.2.jar

│ ├── odps-sdk-core-0.13.2.jar

│ ├── odps-sdk-udf-0.13.2.jar

│ └── odps-sdk-udf-internal-0.13.2.jar

├── pom.xml

└── src

├── main

│ ├── java

│ │ └── my

│ │ └── group

│ │ └── myudf

│ │ ├── MyUDAF.java

│ │ ├── MyUDF.java

│ │ ├── MyUDTF.java

│ │ └── examples

│ │ ├── GcdUDF.java

│ │ ├── NormalizeUDTF.java

│ │ └── VarianceUDAF.java

│ └── resources

│ ├── META-INF

│ │ └── base.udf.xml

│ └── credential.properties

└── test

├── java

│ └── my

│ └── group

│ └── myudf

└── resources

其中，用户的程序写在src/main/java 下。

1. 证书配置

证书文件位于 src/main/resources/credential.properties ，其中存储Base ID、Project ID和Token，是完成鉴权及身份认证的重要文件。

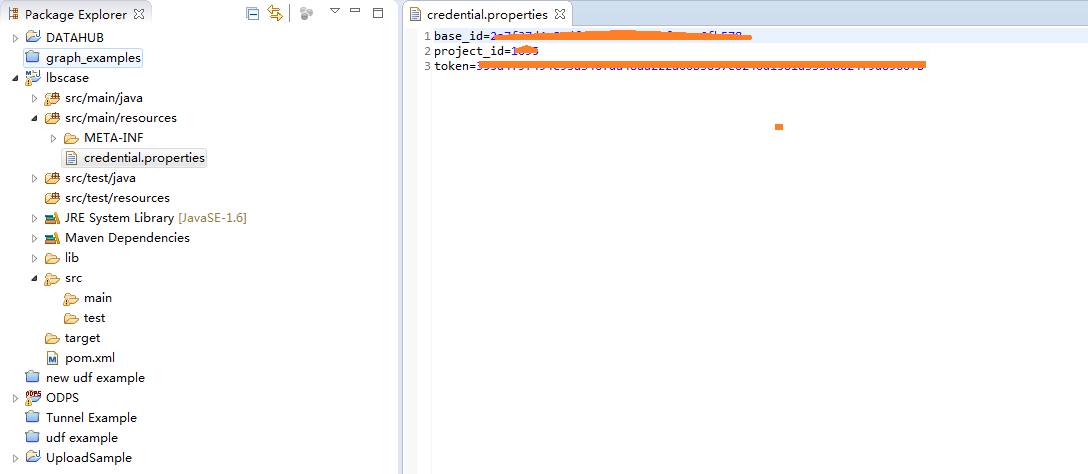


图25 证书配置页面

1. 配置文件

不论是mapreduce还是udf程序，都有一个主配置文件。这个配置文件是整个项目的核心，许多重要信息都需要在这里配置。

UDF程序配置文件位于 src/main/resources/META-INF/base.udf.xml，形如下图。其中：baseId、projectId来源于maven项目初始化配置； functionName必填，为最终注册的UDF名称，即需要在SQL中调用的函数名；className - 必填，为实现UDF的class全名。



图26 UDF程序配置文件

1. 程序开发

在src/main/java中新建class，右击src/main/java并点击<New-> Package>:

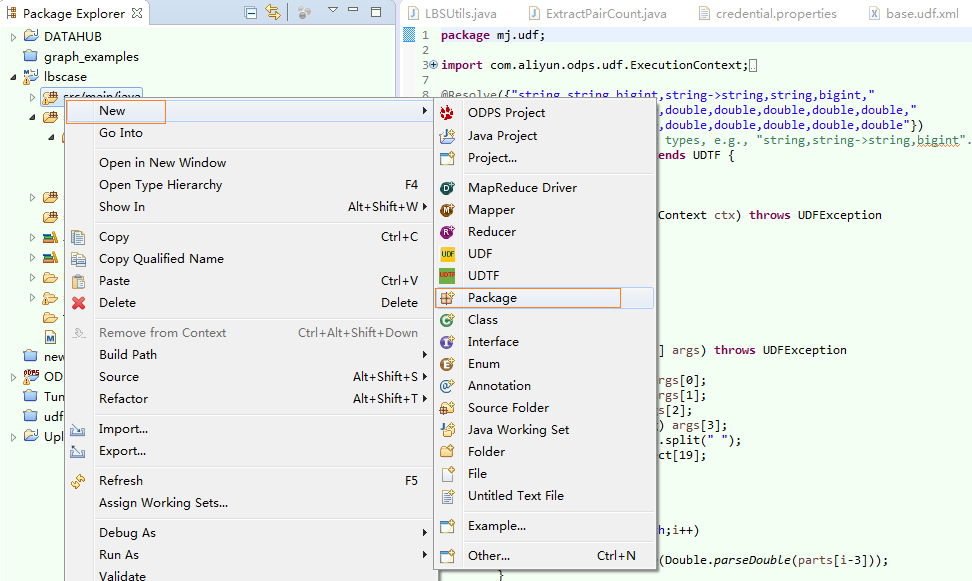


图27 新建package

在弹出的New Java Package对话框中填入基本信息，如：

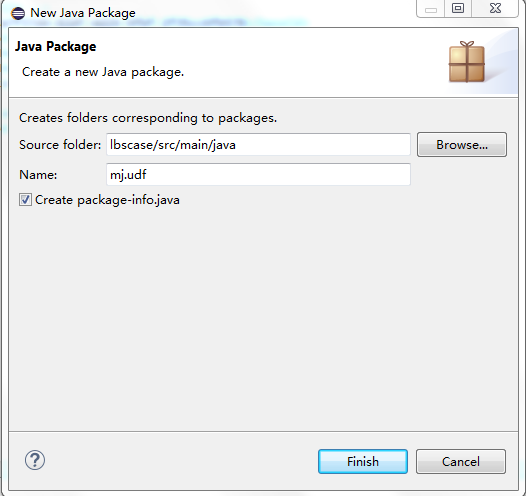


图28 新建Java package

点击<Finish>之后，将会在src/main/java节点下新增”mj.udf”这个节点，如：

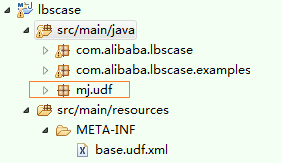


图29 新增的package

右击mj.udf这个package，并点击<New-> Class>。填入Name(LBSUtils)后即可打开程序编辑框。

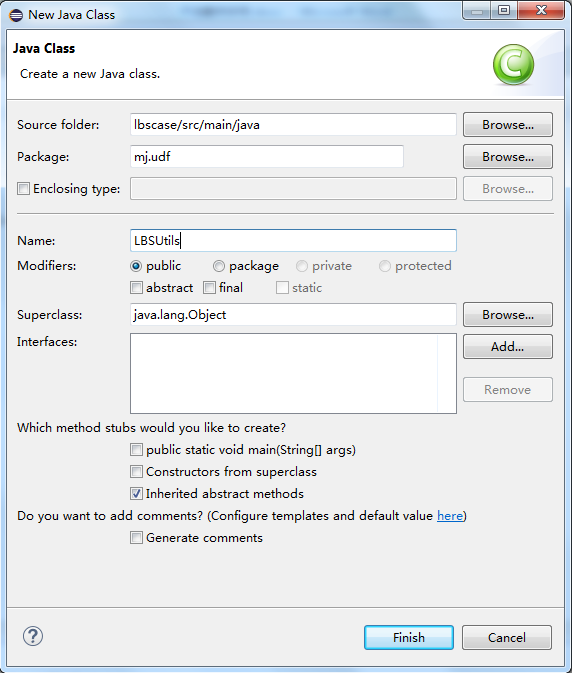


图30 新建Class

在编辑框编写程序代码：

package taijixiong.udf;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class LBSUtils

{

public static int DateSubtract(String d1, String d2) throws ParseException

{

SimpleDateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");

Date date1 = fmt.parse(d1);

Date date2 = fmt.parse(d2);

return (int) ((date1.getTime()-date2.getTime())/(1000L\*60\*60\*24));

}

public static String Join(int[] arr, String sep)

{

if ( arr.length==0)

{

return "";

}

StringBuffer sb = new StringBuffer();

sb.append(arr[0]);

for(int i=1;i<arr.length;i++)

{

sb.append(sep);

sb.append(arr[i]);

}

return sb.toString();

}

public static void main(String args[]) throws ParseException

{

System.out.println(DateSubtract("2014-02-19 20", "2014-12-18"));

}

}

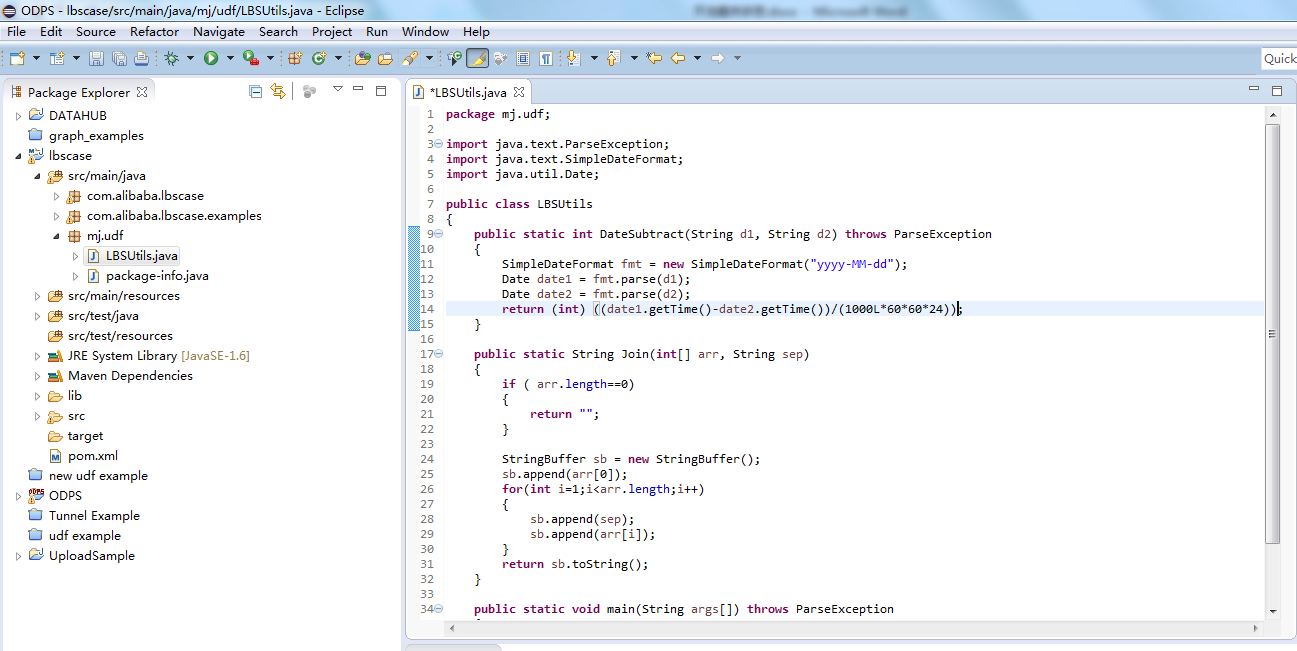


图31 编写UDF代码

采用同样的方法，开发ExtractPairCount.java用于统计user-item对的点击，收藏，购物车，购买次数。代码如下：

package mj.udf;

import java.text.ParseException;

import java.util.Arrays;

import com.aliyun.odps.udf.ExecutionContext;

import com.aliyun.odps.udf.UDTF;

import com.aliyun.odps.udf.annotation.Resolve;

import com.aliyun.odps.udf.UDFException;

@Resolve({"string,string,bigint,string,string,bigint->string,string,string,string"})

// TODO define input and output types, e.g., "string,string->string,bigint".

public class ExtractPairCount extends UDTF {

public String lastUid = "";

public String lastIid = "";

public int[] intervals = {1,3,7,100};

public int currentIndex = 0;

public int[] counts = new int[4\*4];

public String ptime = "";

@Override

public void setup(ExecutionContext ctx) throws UDFException

{

}

/\*\*

\* UDTF Process接口

\*

\* 每条记录都会调用此接口。

\*/

public void process(Object[] args) throws UDFException

{

String uid = (String) args[0];

String iid = (String) args[1];

int behavior = ((Long) args[2]).intValue();

String atime = (String) args[3];

String ptime = (String) args[4];

int cnt = ((Long) args[5]).intValue();

this.ptime = ptime;

if (uid.compareTo(this.lastUid)==0 && iid.compareTo(this.lastIid)==0)

{

}

else

{

//new pair

//output old pair

if (this.lastUid.length()>0 && this.lastIid.length()>0)

{

this.output();

}

//add new pair

this.lastUid = uid;

this.lastIid = iid;

this.clearStatus();

}

int dateDiff;

try {

dateDiff = LBSUtils.DateSubtract(ptime,atime);

} catch (ParseException e) {

// TODO Auto-generated catch block

throw new RuntimeException(e);

}

while (dateDiff > this.intervals[this.currentIndex])

{

this.currentIndex+=1;

}

this.counts[this.currentIndex\*4+behavior-1]+=cnt;

}

public void clearStatus()

{

Arrays.fill(this.counts, 0);

this.currentIndex = 0;

}

public void output() throws UDFException

{

if (this.lastUid.length()>0 && this.lastIid.length()>0)

{

for(int i=0;i<this.intervals.length-1;i++)

{

for(int k=0;k<4;k++)

{

this.counts[(i+1)\*4 + k] += this.counts[i\*4+k];

}

}

String feature = LBSUtils.Join(this.counts, " ");

this.forward(this.lastUid,this.lastIid,

this.ptime

,feature);

}

}

/\*\*

\* UDTF Close接口

\*

\* 任务最后调用此接口，规格化所有数据并输出。forward方法用于输出结果

\*/

public void close() throws UDFException

{

this.output();

}

}

1. 提交程序

UDF程序开发完毕，提交到IDE中运行远程试跑和部署。提交是通过maven插件完成的，以Eclipse为例，提交程序步骤如下：

* 配置settings.xml

御膳房依赖的maven plugin不属于官方插件，所以首先需要配置maven以使得maven运行时可以找到base-maven-plugin。

配置需要放在.m2/settings.xml中，.m2的位置根据不同操作系统不一致，可查阅maven的官方文档确定本机.m2目录所在位置。

一般而言，.m2 文件夹是隐藏文件夹，需要首先在文件夹选项中“显示隐藏的文件、文件夹和驱动器”，然后检索“.m2”文件夹所在位置。

以Windows 7系统为例：

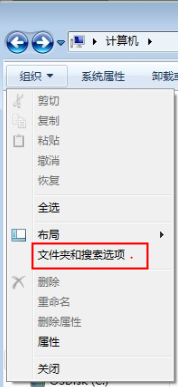


图32 文件夹和搜索选项

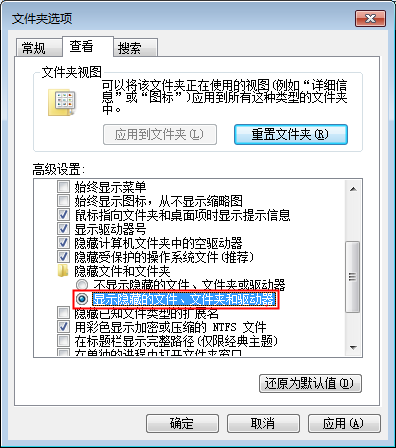


图33 显示隐藏的文件、文件夹和驱动器

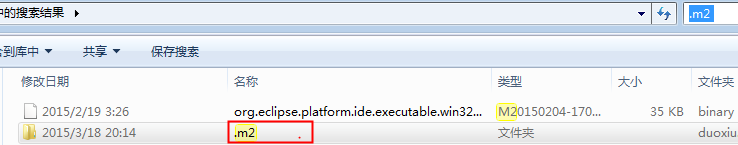


图34 搜索.m2文件夹

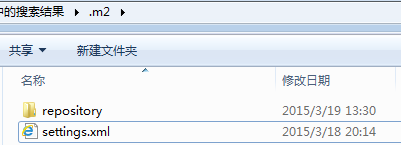


图35 settings.xml

配置的settings.xml项目内如如下：（若用户系统中没有settings.xml可将下面的内容直接新建为settings.xml；否则请将如下配置项合并进已有的settings.xml。）

<?xml version="1.0" encoding="UTF-8"?>

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

<pluginGroups>

<pluginGroup>com.alibaba.base.plugins</pluginGroup>

</pluginGroups>

<profiles>

<profile>

<id>base</id>

<pluginRepositories>

<pluginRepository>

<id>base-snapshots</id>

<url>http://maven.sdk.de.yushanfang.com/SNAPSHOT</url>

<releases>

<enabled>false</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

<activeProfiles>

<activeProfile>base</activeProfile>

</activeProfiles>

</settings>

* base-maven-plugin

御膳房的maven plugin有三个功能：

# 检查项目是否完整

mvn base:check

# 对项目进行打包并生成哈希文件，为提交做好准备

mvn base:zip

# 提交项目到IDE

mvn base:submit

base-maven-plugin的几个特有的配置项如下：

base.program.type - 程序类型。可以是 mapreduce 或 udf

base.endpoint- 御膳房的提交服务地址 http://api.sdk.de.yushanfang.com

base.ide.url - 仅mapreduce需要。御膳房的IDE地址 http://ide.de.yushanfang.com

base.ide.resource.url - 仅mapreduce需要。御膳房实例的资源地址模式，用于生成mr script。 http://@{env}.codebase.de.yushanfang.com/scheduler/res?id={rid}

* 提交：在Eclipse中通过base-maven-plugin提交程序：

1. 在项目上点击右键点击，依次选择 Run As -> Run Configurations...

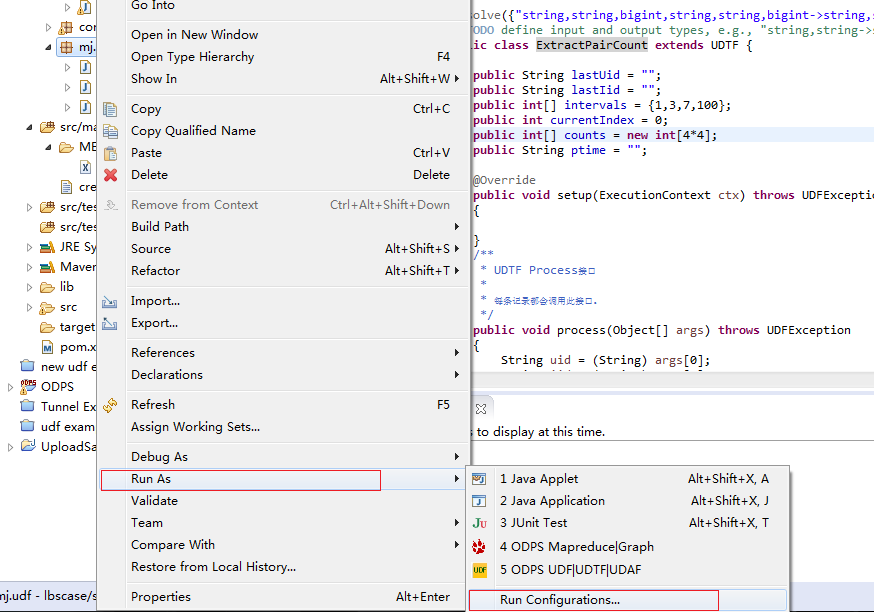


图36 运行按钮

1. 在Run Configurations对话框左侧列表中右键点击< Maven Build -> New>。
2. 右侧窗口中，Base directory 选择项目所在目录。可以点击 <Broswe Workspace... >或 <Browse File System...> 选择项目所在目录。
3. Goals 中输入 base:check base:zip base:submit。
4. Parameters列表中填入上文提到的base-maven-plugin配置项。
5. 点击 <Apply> ，再点击< Run> ，Eclipse就会自动调用base-maven-plugin提交程序。

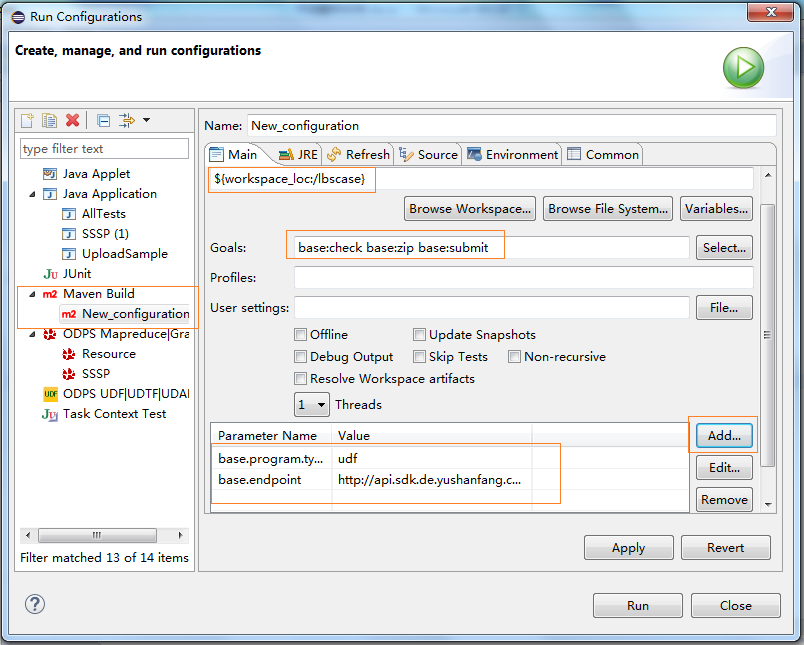


图37 运行配置

注：

对于mapreduce程序，如果成功后会自动打开IDE，并自动新建好mr程序节点，用户可以在IDE中完成后续工作。如果是udf，则会自动将UDF注册好，在IDE中可直接看到并使用。

因为这里示例的是UDF，提交后会弹出Console显示运行情况，成功会在Console中提示BUILD SUCCESS。

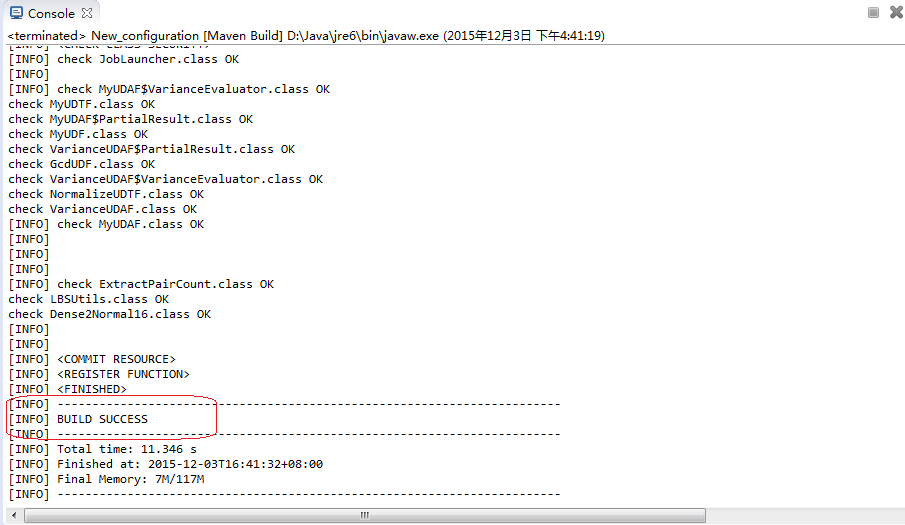


图38 运行成功

1. 回到御膳房- 数据开发工作台，在相应目录下可以看到上传了的UDF JAR包并注册了myudf函数。

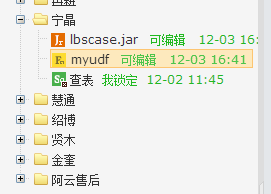


图39 注册好的udf

1. 双击myudf函数，可看到函数的注册信息，是对应上配置文件src/main/resources/META-INF/base.udf.xml的信息的：



图40 注册函数

注：如果JAR包中包含多个UDF，可在御膳房-数据开发工作台中手动注册函数，注册函数编辑框中的资源列表要填写工程生成的jar包。



图41 手动注册函数

接下来即可调用udf来抽取特征了，新建一个sql节点，代码写好后点击运行。

#### 抽取特征

1. 样本选取：（正样本）分别以2014/12/17的购买，12/18的购买来构建验证集、测试集

相关SQL如下：

DROP TABLE IF EXISTS t\_label\_valid\_train;

DROP TABLE IF EXISTS t\_label\_test\_train;

CREATE TABLE t\_label\_valid\_train AS

SELECT

user\_id

,item\_id

,1 AS label

FROM

t\_mj\_p\_user

WHERE

time>'2014-12-17' AND time<'2014-12-18' AND behavior\_type=4

GROUP BY user\_id,item\_id;

CREATE TABLE t\_label\_test\_train AS

SELECT

user\_id

,item\_id

,1 AS label

FROM

t\_mj\_p\_user

WHERE

time>'2014-12-18' AND time<'2014-12-19' AND behavior\_type=4

GROUP BY user\_id,item\_id;

1. 特征工程：构建UDF函数来统计user-item对的点击，收藏，加购，购买次数。该udf已经在上述的章节成功构建。调用该函数进行统计的SQL代码如下：

DROP TABLE IF EXISTS tfea\_ui\_count;

CREATE TABLE tfea\_ui\_count(user\_id STRING,item\_id STRING, time STRING, feature STRING);

INSERT INTO TABLE tfea\_ui\_count

SELECT

myudf(user\_id,item\_id,behavior\_type,time,'2014-12-19',1) AS (user\_id,item\_id,time,feature)

FROM

(

SELECT

\*

FROM t\_mj\_p\_user tb2

WHERE time < '2014-12-19'

DISTRIBUTE BY user\_id,item\_id

SORT BY user\_id,item\_id,time DESC

) tb1;

INSERT INTO TABLE tfea\_ui\_count

SELECT

myudf(user\_id,item\_id,behavior\_type,time,'2014-12-18',1) AS (user\_id,item\_id,time,feature)

FROM

(

SELECT

\*

FROM

t\_mj\_p\_user tb2

WHERE

time < '2014-12-18'

DISTRIBUTE BY user\_id,item\_id

SORT BY user\_id,item\_id,time DESC

) tb1;

INSERT INTO TABLE tfea\_ui\_count

SELECT

myudf(user\_id,item\_id,behavior\_type,time,'2014-12-17',1) AS (user\_id,item\_id,time,feature)

FROM

(

SELECT

\*

FROM

t\_mj\_p\_user tb2

WHERE time < '2014-12-17'

DISTRIBUTE BY user\_id,item\_id

SORT BY user\_id,item\_id,time DESC

) tb1;

1. 结合特征构建验证集和测试集，包含训练和测试。验证集用本地验证，测试集用于提交。

* --valid\_train：验证训练集,~20141217
* --valid\_test：验证测试集,~20141218
* --test\_train：测试训练集,~20141218
* --test\_test：测试测试集,all

相关SQL如下：

DROP TABLE IF EXISTS t\_valid\_train\_fea;

DROP TABLE IF EXISTS t\_valid\_test\_fea;

DROP TABLE IF EXISTS t\_test\_train\_fea;

DROP TABLE IF EXISTS t\_test\_test\_fea;

CREATE TABLE t\_valid\_train\_fea AS

SELECT

tb1.user\_id

,tb1.item\_id

,tb1.time

,CASE

WHEN tb2.label IS NOT NULL THEN tb2.label

ELSE 0

END AS label

,feature

FROM

(select \* from tfea\_ui\_count where time='2014-12-17') tb1

LEFT OUTER JOIN t\_label\_valid\_train tb2

ON tb1.user\_id=tb2.user\_id AND tb1.item\_id=tb2.item\_id;

CREATE TABLE t\_valid\_test\_fea AS

SELECT

tb1.\*

FROM tfea\_ui\_count tb1

WHERE tb1.time='2014-12-18';

CREATE TABLE t\_test\_train\_fea AS

SELECT

tb1.user\_id

,tb1.item\_id

,tb1.time

,CASE

WHEN tb2.label IS NOT NULL THEN tb2.label

ELSE 0

END AS label

,feature

FROM

tfea\_ui\_count tb1

LEFT OUTER JOIN t\_label\_test\_train tb2

ON tb1.user\_id=tb2.user\_id AND tb1.item\_id=tb2.item\_id

WHERE tb1.time='2014-12-18';

CREATE TABLE t\_test\_test\_fea AS

SELECT

tb1.\*

FROM tfea\_ui\_count tb1

WHERE tb1.time='2014-12-19';

* 检查正负样本比例

select label, count(1) from t\_valid\_train\_fea group by label;

select label, count(1) from t\_valid\_test\_fea group by label;

下面一步就是数据抽样。我们可以通过SQL来实现，也可以通过PAI平台。关于数据抽样的介绍，请参考下面章节的描述。

## PAI平台

### PAI简介

阿里PAI机器学习平台是构建在阿里云ODPS计算平台之上，集数据处理、特征工程、建模、离线预测为一体的机器学习平台。该平台提供了一套极易操作的可视化编辑页面，同时汇集了阿里集团的大量优质算法，不仅大大降低了数据挖掘的门槛，而且帮你快速洞察出大数据掩藏的秘密。

新手快速入门及完整详细的PAI操作手册，参见[这里](http://setting.tenant.yushanfang.com/portal/help/doc.html?file=SuanFaPingTai)。

在PAI上，用户可以按照CRISP-DM（cross-industry standard process for data mining，跨行业数据挖掘标准流程）来完成完整的数据挖掘实验，包括如下几类主要操作：

* 数据准备
* 数据预处理（如：采样及过滤、数据合并、特征变换、缺失值补充、切分等）
* 数据可视化（如：直方图）
* 算法建模（如：分类、聚类、回归、文本分析、网络分析等）
* 多模型评估（如：ROC曲线、混淆矩阵计算、预测等）

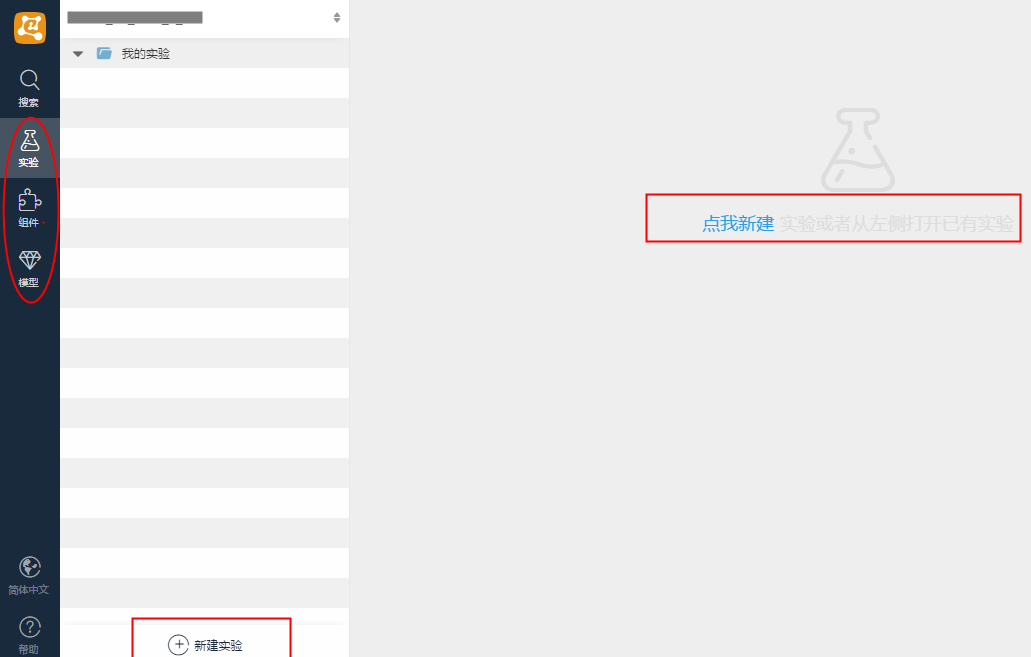


图42 算法平台主页面

算法平台在御膳房上有2个入口：

1. 数据引擎 – 项目首页，选择“算法平台”：



图43：如何进入算法平台方式一

1. 数据开发工作台 – 数据开发下拉选项

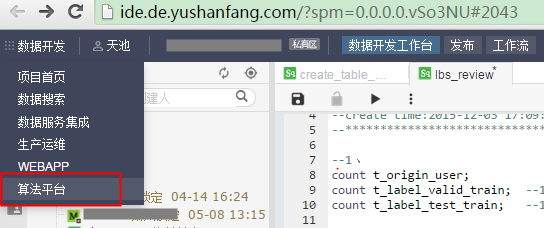


图44如何进入算法平台方式二

由于上述训练集中正负样本比例极其不均衡，为此需要做样本均衡处理（注：真实的业务场景中，样本不均衡是极其常见的，如何做好样本均衡也是数据挖掘的一个知识点。以下只做操作示例，不对样本的比例做深入探究，这个留与读者去思考和实践）。对正样本复制4倍，负样本抽样1/15。 其中正样本复制4倍可以通过SQL完成。

以下将会详细介绍如何在算法平台上完成。

### 数据抽样

1. 负样本抽样1/15
2. 在算法平台页面，点击左下角的弹出[新建实验]对话框， 输入名称和描述：

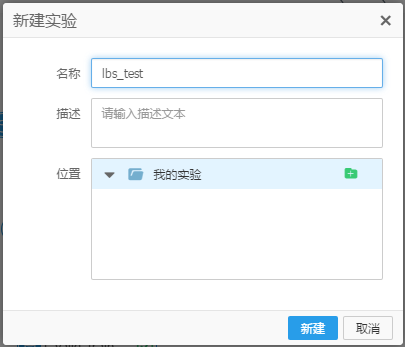


图45 新建实验

1. 左侧导航切换到组件栏；向画布中拖入odps源，随机采样，ODPS目标；点击odps源，在右侧表选择栏填入odps表t\_valid\_train\_fea。（注：这里的表必须为本项目空间的表，不能引用跨项目的表；表名无需带上project名称）

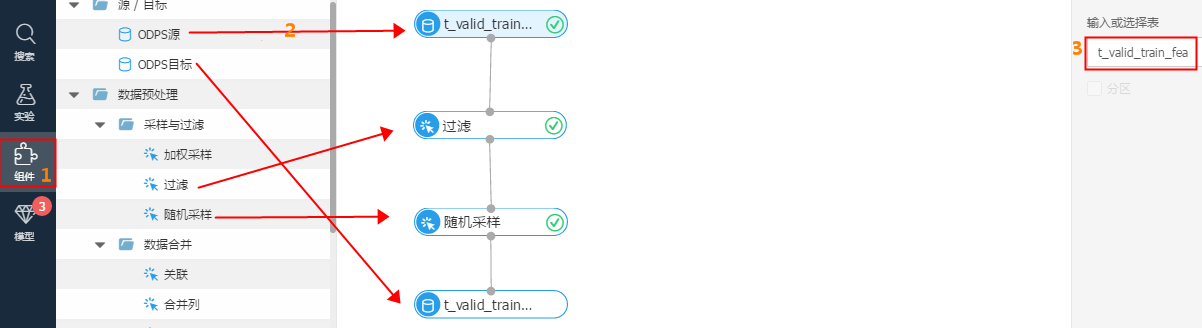


图46 添加组件

1. 选择完表后，切换到字段信息栏，可以查看输入表的字段名、数据类型和前100行数据的数值分布。（也可右击ODPS源组件→查看数据，进行数据探查）



图47 字段信息栏

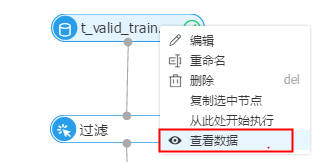


图48 查看数据按钮

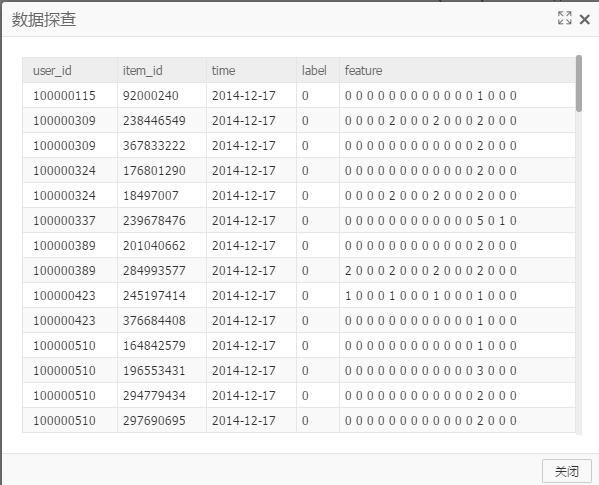


图49 数据探查结果

1. 过滤组件，设置条件为：label=0



图50 过滤条件设置

1. 随机采样组件，设置参数：（1/15约为0.067）

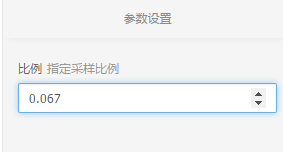


图51 设定采样比例

1. 向画布拖入一个odps目标组件，输入新表名t\_valid\_train\_fea\_neg。（注：如果目标表已经存在，会提示“目标表已存在，继续操作会重建该表”）



图52 输入目标表

1. 连接所有组件，点击<运行>。状态栏显示算法模型的运行进展。



图53 运行界面

1. 运行成功后，可以右键ODPS目标组件→查看数据，看到抽样后的结果（也可以返回数据开发工作台进行校验）。
2. 如果失败，可以选中具体的节点，右键，查看日志，形如下图，通过查看日志定位出错原因从而做相应调整。



图54 查看日志

1. 样本合并

负抽样完成后，与复制了4倍的正样本合并。合并后的结果为：

2014-12-17的训练集t\_valid\_train\_fea\_sub ；

2014-12-18的训练集t\_test\_train\_fea\_sub。

上述“负样本抽样1/15，正样本复制4倍，”的动作也可以直接通过加权采样+ 数据合并 来完成，不再赘述，留给用户自己来尝试。另外，样本不均衡在现实业务场景是极其常见的，如何做好合理的样本均衡处理，留给大家去深入探索。

1. 转换特征的数据格式

由于之前用UDF抽取的特征为字符串格式存储于一个字段中，现转换为每个特征一列的格式。类似之前一样，写了个UDF函数udf\_d2n16，调用之完成数据格式的转换。SQL代码如下：

DROP TABLE IF EXISTS t\_valid\_train\_fea\_sub\_n;

DROP TABLE IF EXISTS t\_test\_train\_fea\_sub\_n;

DROP TABLE IF EXISTS t\_valid\_test\_fea\_n;

DROP TABLE IF EXISTS t\_test\_test\_fea\_n;

CREATE TABLE t\_valid\_train\_fea\_sub\_n AS

SELECT udf\_d2n16(user\_id,item\_id,label,feature) AS

(user\_id,item\_id,label,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16)

FROM t\_valid\_train\_fea\_sub;

CREATE TABLE t\_test\_train\_fea\_sub\_n AS SELECT udf\_d2n16(user\_id,item\_id,label,feature) AS

(user\_id,item\_id,label,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16)

FROM t\_test\_train\_fea\_sub;

CREATE TABLE t\_valid\_test\_fea\_n AS SELECT udf\_d2n16(user\_id,item\_id,0,feature) AS

(user\_id,item\_id,label,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16)

FROM t\_valid\_test\_fea;

CREATE TABLE t\_test\_test\_fea\_n AS SELECT udf\_d2n16(user\_id,item\_id,0,feature) AS

(user\_id,item\_id,label,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16)

FROM t\_test\_test\_fea;

### 建模和评估

特征提好，数据集准备好，那么就可以跑模型了。跑模型除了用MR和SQL自己写以外，算法平台还提供了丰富的数据挖掘和处理算法。

一个典型的训练预测流程如下图所示：

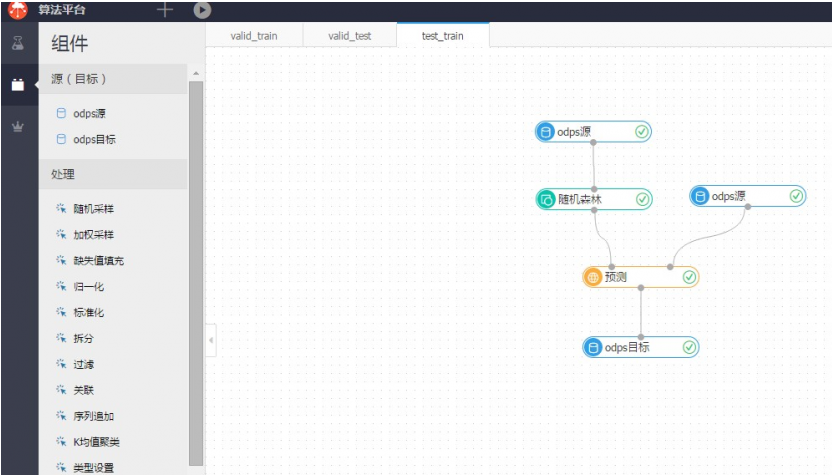


图55 典型的训练预测流程

建模前可以再次对样本及特征进行检查，可以使用”全表统计“来做个概览，拖入一个全表统计组件，形如：

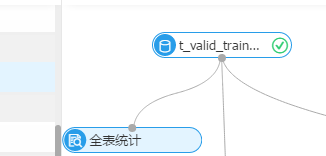


图56全表统计组件

运行成功后，右键”全表统计“ →查看分析报告。可以查看到表的记录数、各字段的缺失值、空值、最小值、最大值、平均值、方差等。

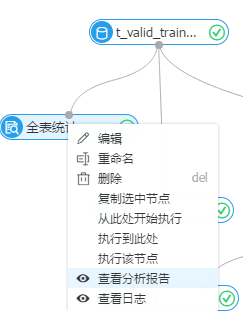


图57 查看分析报告

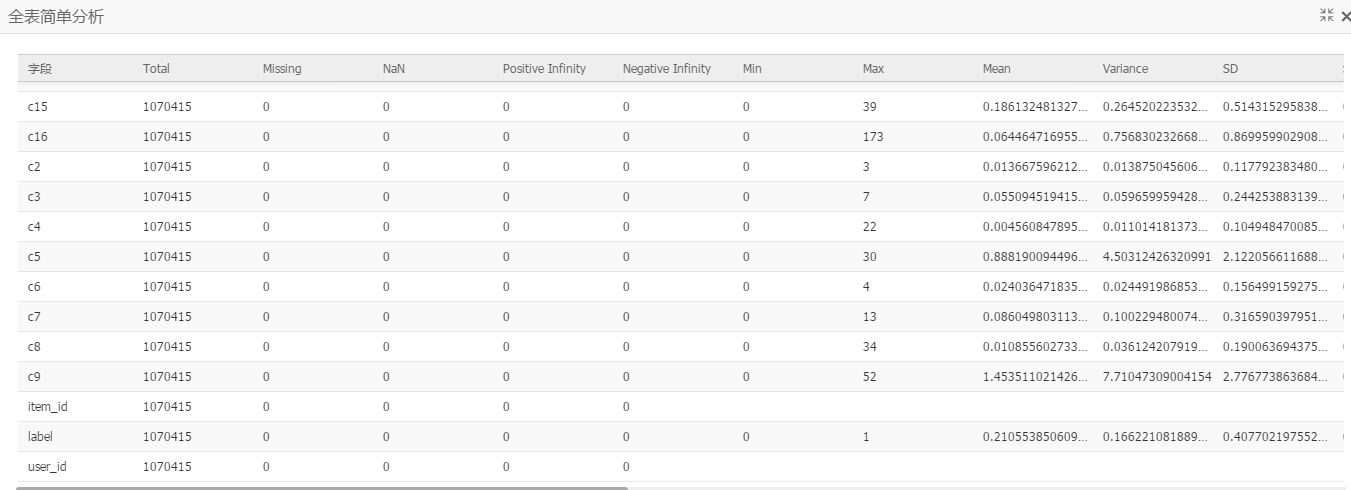


图58 分析结果

以下采用2种训练方式：逻辑回归、随机森林，进行建模的对比。

#### 逻辑回归

经典逻辑回归是一个二分类算法，算法平台的逻辑回归可以支持多分类。模型包括训练和生成模型、模型预测两个部分。 该算法支持稀疏矩阵的格式，最大支持的稀疏特征数为100万。

1. 新建实验，往画布里拖入ODPS源组件， 机器学习/分类/逻辑回归组件，
2. ODPS源组件，选择表t\_valid\_train\_fea\_sub\_n
3. 逻辑回归组件，“字段设置”TAB中，选择特征列和标签列（标签列即目标变量、分类标签），其中特征列选择上述特征工程所构建的C1-C16这16个特征，标签列选择label字段（即：特定的User\_id是否购买了特定的Item\_id， 1为购买，0未购买）; 切换到“参数设置”Tab, 勾选“是否二分类”，设置目标基准值（正例值）为1，其它几个参数直接沿用了默认值。

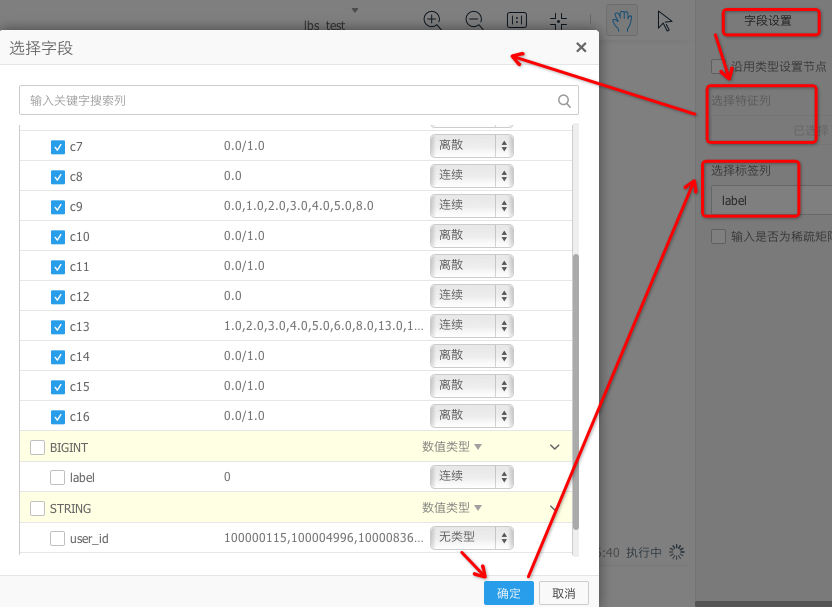


图59 逻辑回归-字段选择

注：

* 沿用类型设置节点必须在输入表进行类型设置处理后才能勾选。
* 输入列：支持double类型与bigint类型。
* 标签列：只能选择非输入列的其它列，支持string类型、double类型和bigint类型。
* 权重列：（可选）只能选择非输入列和非标签列，支持double类型和bigint类型。
* 系统会对字段类型进行初步判断（离散，连续，无类型），可根据算法的需要调整所选输入列的类型。
* LR的体征不能为空值，否则会报错。



图60 逻辑回归参数设置

附：参数说明

* 是否二分类：（可选）默认多分类，勾选后表示选择二分类
* 目标基准值：（可选）二分类时，指定训练系数针对的label值；如果填空，会随机选择一个
* 最大迭代数：（可选）L-BFGS的最大迭代次数，默认是100
* 收敛误差：（可选）L-BFGS的终止条件，即两次迭代之间log-likelihood的差，默认为1.0e-06
* 正则化类型：（可选）正则化类型，可以选择‘l1、‘l2’、‘None’，默认为‘l1’
* 正则化系数：（可选）正则项系数，默认为 1.0；当 regularizedType 为 None 时，该项会被忽略。

#### 随机森林

1. 拖入随机森林组件，选择字段（与上述逻辑回归一样，但值得注意的是，随机森林的输入列较之于逻辑回归，还支持string类型），设置参数如下：



图61 随机森林参数设置

注：

* 随机森林输入列支持连续型和离散型输入，选择的数据类型不一样，计算方式也不一样
* 算法类型：（可选）可供选择的算法类型有id3算法、cart算法、c4.5算法以及默认情况下的将上述三种算法均分的混合算法
* 树的数目：森林中树的个数, 范围(0, 1000]
* 随机属性类型：（可选）单颗树在生成时，随机选择的特征个数。可供选择的类型有logN，N/3，sqrtN，N四种类型；即如果有8个特征，选了logN，那么训练每个节点时随机选3（log8）个特征用于该节点处的拆分特征;注：该参数不是每个树的特征个数。
* 树最大深度：（可选）单颗树的最大深度。-1表示完全生长。范围[1, ∞)
* 叶子节点最少记录数：（可选）叶节点数据的最小个数。最小个数为2
* 叶子节点最少记录百分比：（可选）叶节点数据个数占父节点的最小比例，-1表示无这一限制。默认-1，范围[0,100]
* 每棵树最大记录数：（可选）森林中单颗树输入的随机数据的个数。范围为(1000, 1000000)

1. 连接组件后，点击运行按钮，状态栏显示出开始时间及正在执行中，如下：



图62 运行界面

运行成功后可以看到左侧导航栏的模型中显示生成了2个模型。

#### 预测和评估

##### 预测

预测组件是专门用于模型预测的组件，两个输入：训练模型和预测数据；输出为预测结果；

传统的数据挖掘算法一般都采用该组件进行预测操作。

预测组件的参数设置形如：



图63预测组件参数设置

预测组件的唯一参数：‘是否为稀疏矩阵’，该参数表示输入预测集的数据格式是否为稀疏矩阵 。

若数据是稀疏格式，需要选择稀疏矩阵的特征列，稀疏的格式如：

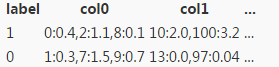


图64 稀疏格式

选择验证测试集t\_test\_train\_fea\_sub\_n来对2个模型进行评估。

分类模型常用的评估方式有ROC曲线，混淆矩阵等。

评估类组件需连接预测组件。

##### ROC曲线

ROC曲线是根据一系列不同的二分类方式（分界值或决定阈），以真阳性率（灵敏度）为纵坐标，假阳性率（1-特异度）为横坐标绘制的曲线。ROC曲线是一种座标图式的分析工具，可用于 (1) 选择最佳的信号侦测模型、舍弃次佳的模型。 (2) 在同一模型中设定最佳阈值。

注：

ROC曲线只支持二分类，不支持多分类；不支持GBDT回归与排序模型。

操作如下：

1. 向画布中拖入ODPS源、预测组件、ROC曲线，连接。

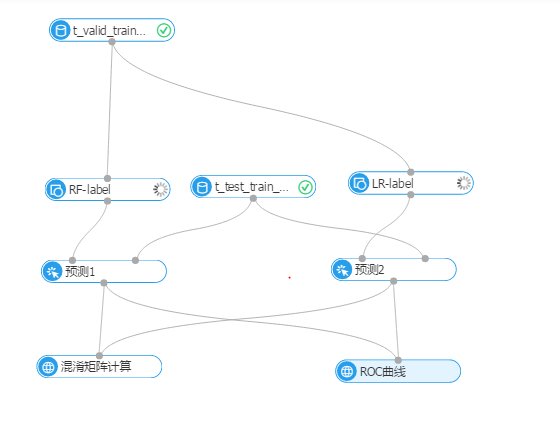


图65 ROC曲线

1. 新加入的ODPS源中选择t\_test\_train\_fea\_sub\_n表；
2. 设置ROC曲线参数：可默认沿用设置，"选择标签列"和"目标基准值"都将继承父节点中的相关参数；"目标基准值"是针对逻辑回归组件的二分类问题下的参数值。

注：若父节点不是逻辑回归且不是二分类的情况，ROC将运行报错，这是需要手动设置ROC的"目标基准值"。



图66 ROC参数设置

1. 点击运行
2. 运行成功后，右键ROC曲线→ 查看评估报告。

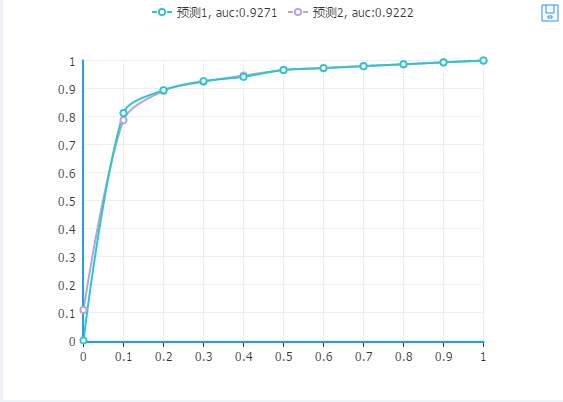


图67 ROC曲线评估报告

从auc结果来看，预测1（随机森林RF）略优于预测2（逻辑回归）。

另外模型参数的设置对结果也会有影响，至于如何调参到最优效果，需要结合和业务数据和模型经验，逐步尝试寻找最优参数。

##### 混淆矩阵

混淆矩阵（confusion matrix）是可视化工具，特别用于监督学习，在无监督学习一般叫做匹配矩阵,主要用于比较分类结果和实际测得值，可以把分类结果的精度显示在一个混淆矩阵里面。只有分类模型才可进行混淆矩阵分析。

1. 向画布中拖入ODPS源、预测组件、混淆矩阵计算，连接。

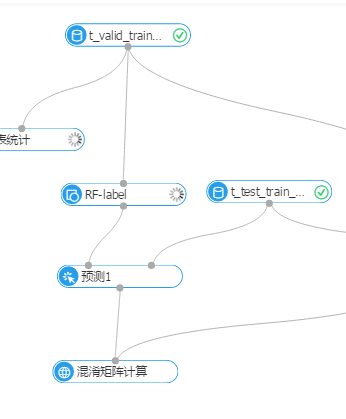


图68 混淆矩阵组件设置

1. 新加入的ODPS源中选择t\_test\_train\_fea\_sub\_n表；
2. 设置混淆矩阵计算的参数：一般情况下勾选默认的沿用设置，用户也可自行选择目标列和预测概率列，预测概率列指预测组件生成预测目标列名。



图69混淆矩阵参数设置

1. 点击运行。
2. 运行成功后，右键<混淆矩阵组件→ 查看评估报告>。



图70 查看评估报告结果（混淆矩阵）

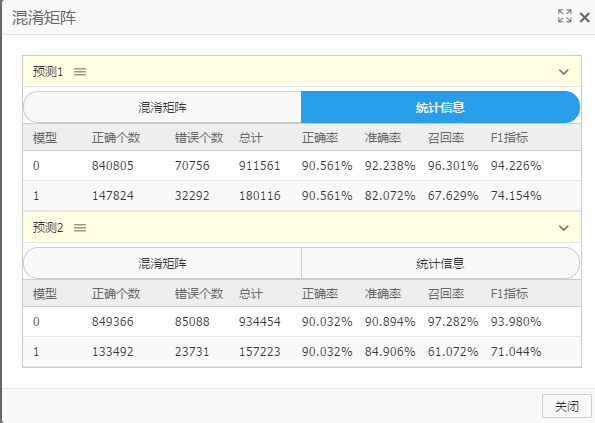


图71 统计信息

整体正确率计算结果：正确个数总数/总数

预测1（随机森林）整体正确率= (840805+147824)/(911561+180116)=90.56%

预测2（逻辑回归）整体正确率=(849366+133492)/(934454+157223)=90.03%

从混淆矩阵结果来看，两种算法的参数采取默认值的情况下，模型的准确率几乎相同。

更改默认参数后准确率会有改变吗？答案是肯定的，大家可以一试。

### PAI命令及模型同步

1. PAI命令

除了通过IDE组件及参数设置方式外，还可以用PAI命令来完成相应操作。

以混淆矩阵计算为例，上述操作混淆矩阵的PAI命令如下：

PAI -name *confusionmatrix* -project *algo\_public* -DoutputTableName="*pai\_temp\_2954\_24178\_1*"

-DlabelColName="*label*" -DpredictionColName="*prediction\_result*" -DinputTableName=" *t\_test\_train\_fea\_sub\_n*";

其中，

* name: 组件名字。
* project: (可选)默认project是algo\_public。如果选择其它的Project，需要指定algo\_public下的算法包，否则报错。
* outputTableName：混淆矩阵结果表。
* labelColName：输入表的标签列列名。
* predictionColName：预测结果列。
* inputTableName：输入预测结果表名。

1. 模型同步

通过PAI命令生成的模型，可以在算法平台中实现同步：模型 → 同步模型，右键，选择”立即同步”即可（注：同步的模型暂时还不支持模型查看）。



图72 模型同步

### 线上测试集的训练和预测

经过上述ROC曲线和混淆矩阵的评估，可以选择随机森林，对测试集（t\_test\_train\_fea\_sub\_n和t\_test\_test\_fea\_n）进行训练和测试，生成赛题需求的结果提交。

1. 依次向画布中拖入组件：ODPS源，随机森林，ODPS源，预测，ODPS目标，并连接好所有组件：

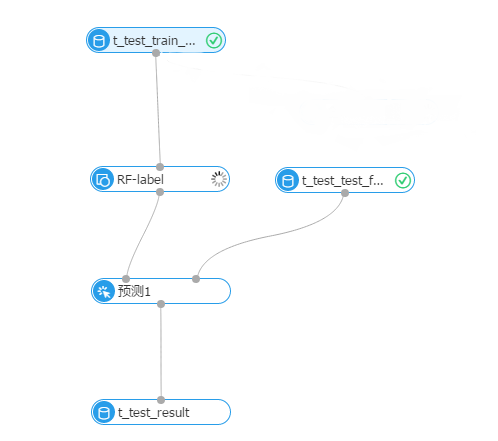


图73 拖入组件并连接

1. 设置2个ODPS源组件，分别选择表：

点击运行，运行完成后可以用sql查看结果表：t\_test\_train\_fea\_sub\_n和t\_test\_test\_fea\_n。

1. 设置随机森林参数，选择字段和标签列。
2. 设置ODPS目标表为t\_test\_result（将模型预测出来的结果表命名为t\_test\_result）。
3. 点击运行。运行完成后可以右击，查看结果；也可返回数据开发工作台中通过SQL来查看结果。可以看到结果表会追加3列，predict\_result是预测结果，prediction\_score是预测为predict\_result的概率，这里要注意一下，这个概率是跟predict\_result有关的。

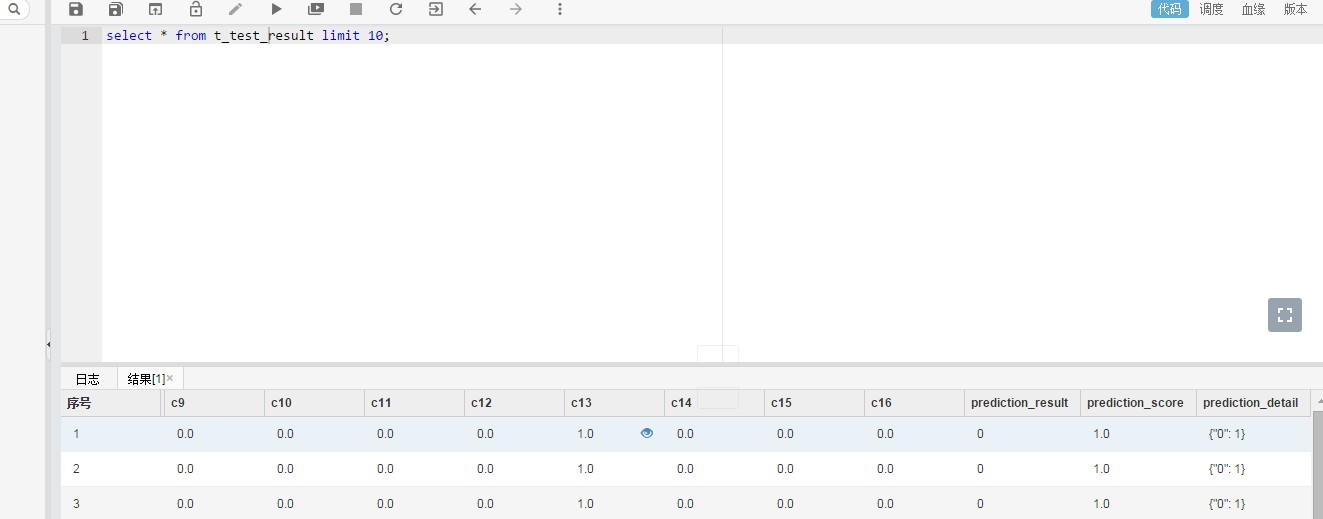


图74 运行结果

1. 提交结果

根据上述模型生成的预测结果，按照概率值从大到小排序，选择合适的量来提交到赛题要求的命名tianchi\_mobile\_recommendation\_predict结果表中，完成。

如，选择概率较高的TOP 200000：代码如下：

DROP TABLE IF EXISTS tianchi\_mobile\_recommendation\_predict;

CREATE TABLE tianchi\_mobile\_recommendation\_predict AS SELECT user\_id, item\_id FROM

(SELECT user\_id,item\_id,prediction\_score FROM t\_test\_result WHERE

prediction\_result=1 ORDER BY prediction\_score DESC LIMIT 200000) tb1



图75 结果表

就这样，评测的时候评测程序会自动找到工程下的这个表名（具体请参见比赛说明）进行评分排行。本结果可以获得的分数为： 准确率2.298%， F1为3.22%51。

这2个结果提交的案例只是初级的baseline，意在由浅入深完整地示范如何使用平台来参加算法大赛或者数据挖掘实践，希望这样step by step的操作能使大家迅速地上手。然而，功夫在诗外，要想取得好成绩，实践有突破，还需要大家夯实基础，深入地理解数据挖掘/机器学习的各算法原理，结合对业务的了解和分析，在实践中逐步提高。

名词解释

* **ODPS（Open Data Processing Service）**：开放数据处理服务由阿里云自主研发，提供针对TB/PB级数据、实时性要求不高的分布式处理能力，应用于数据分析、挖掘、商业智能等领域。
* **项目(Project)**: 项目（也称项目空间）是ODPS最基本的组织对象。其他对象，例如表(Table)和实例(Instance)等都归属于一个项目。
* **实验（Experiment）**: 实验是指PAI平台用户搭建的数据工作流程或者数据应用。用户需要先建立一个实验实例，然后在实验画布上搭建数据流程。
* **ODPS源表与ODPS目标表(Table)**: 表(Table)是ODPS中数据存储对象。与常见的关系型数据类似，ODPS中的表逻辑上也是二维结构。源表指一个算法节点的输入，目标表指算法节点的输出。
* **组件（Nodes）**： 组件是用户可以在PAI平台上调用执行的最小操作单元， 例如数据导入导出、数据处理、数据分析、模型训练或者预测。
* **模型（Model）**： 模型是特指一个算法或者机器学习训练组件产生的结果数据。模型是一类特殊的组件。
* **分区(partition)** ： ODPS表分区（Partition）， 是指一张表下，根据分区字段（一个或多个组合）对数据存储进行划分。也就是说，如果表没有分区，数据是直接放在表所在的目录下；而如果表有 Partition，每个 Partition 对应表下的一个目录，数据是分别存储在不同的分区目录下。比如，假设前面给出的各个字段创建一张表，表名为 page\_view，指定其分区字段为 dt（日期）和 country，则 对 于 分 区 dt=20150101,country=US 下 的 数 据 ， 就 会 存 放 在 page\_view/dt=20150101/country=US/的目录下。分区的最大好处在于可以加快查询，比如要查找满足 dt=20150101 且 country=US 的数据，只需要扫描相应的分区即dt=20150101/country=US/ 目录下的数据即可；如果没有分区，则需要扫描表page\_view/下的所有数据。
* **数据挖掘：**数据挖掘一般是指从大量的数据中通过算法搜索隐藏于其中信息的过程。数据挖掘通常与计算机科学有关，并通过统计、在线分析处理、情报检索、机器学习、专家系统（依靠过去的经验法则）和模式识别等诸多方法来实现上述目标。
* **机器学习：**机器学习(Machine Learning, ML)是一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。专门研究计算机怎样模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能。
* **CRISP-DM：**cross-industry standard process for data mining，跨行业数据挖掘标准流程。

修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| Revision NO. | Revised Date | Revised by | Reason for Revision |
| V5 | 2015-12-09 |  |  |
| V6 | 2015-12-15 |  | 实现步骤添加概述 |
| V7 | 2016-2-28 |  | 新增背景介绍，修改课题介绍部分，改成通俗易懂的白话文 |
| V8 | 2016-3-3 | 宁晶 | 更新准备工作的注意事项，更新PAI逻辑回归的注意事项。 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |