

**天池用户手册**

**--数加平台**

内容

[内容 2](#_Toc451454368)

[1 注册 3](#_Toc451454369)

[2 报名和组队 5](#_Toc451454370)

[2.1 报名 5](#_Toc451454371)

[2.2 我的团队 7](#_Toc451454372)

[3 离线赛 9](#_Toc451454373)

[3.1 下载数据 9](#_Toc451454374)

[3.2 提交结果 9](#_Toc451454375)

[4 平台赛-数加 10](#_Toc451454376)

[3.1. 认证 10](#_Toc451454377)

[4.2 登录数加平台 11](#_Toc451454378)

[4.2.1 获取RAM账号信息 11](#_Toc451454379)

[4.3 读取/复制数据 14](#_Toc451454380)

[4.4 数据开发 14](#_Toc451454381)

[4.4.1 创建任务 14](#_Toc451454382)

[4.4.2 运行ODPS SQL 17](#_Toc451454383)

[4.4.3 运行ODPS MR 18](#_Toc451454384)

[4.5 机器学习平台 31](#_Toc451454385)

[4.5.1 简介 31](#_Toc451454386)

[4.5.2 数据准备 32](#_Toc451454387)

[4.5.3 数据预处理 34](#_Toc451454388)

[4.5.4 建模和评估 35](#_Toc451454389)

[4.6 提交结果 42](#_Toc451454390)

[5 附录 44](#_Toc451454391)

[修订历史 46](#_Toc451454392)

# 注册

对于天池新用户，第一步操作就是打开天池官网并在天池注册自己的账号信息。

步骤如下：

1. 登录天池官网 (https://tianchi.aliyun.com/) 并点击右上角的<注册>。



图 1 天池官网首页

1. 弹出如下提示：

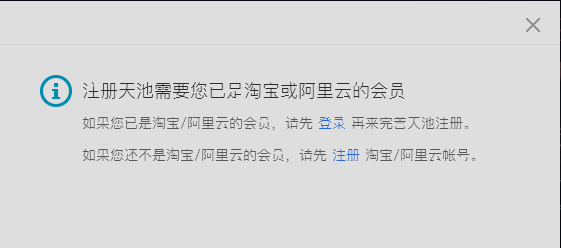


图 2 提示信息

1. 点击<注册>并在弹出的对话框内配置如下：



图 3 账号注册页面

1. 点击<确认> 弹出对话框。输入对应的手机验证码即注册成功。
2. 返回天池官网并点击右上角的 <登录>按钮，界面跳转到登录界面，输入对应的阿里云账号和密码，如下：



图4 登录主界面

1. 点击<登录>登录天池网站。

# 报名和组队

## 报名

对于天池新用户，在参加比赛前必须完善个人信息。 （老用户无需关注此步骤）

入口：登录天池->点击右上角的用户名->个人中心->编辑个人信息



图5 完善个人信息页面

1. 成功登录天池官网后，点击菜单栏的[天池大赛->算法大赛] 进入比赛列表页面，选择你感兴趣的比赛：



图6 菜单栏



图7 算法比赛列表

**注意：**

目前，我们的比赛分成三种类型：算法大赛，可视化大赛，程序设计大赛。 您可以通过菜单栏向导找到你需要参加的比赛。

1. 单击<了解更多> 进入比赛详情页面，如：



图8 报名页面

1. 点击<报名参赛> 报名对应的比赛：



图9 报名邀请码

1. 点击<下一步>成功报名。

**注意：**

如果你需要加入对方的团队，请选择[我有组队密令]并输入对方的组队密令。

## 我的团队

在上述报名示例中，如果选手选择的是[没有收到组队密令]，那就是以个人身份参赛。报名成功后，选手可以选择组建团队或者加入其他团队。

1. 在比赛详情页面，进入[我的团队]，如下图：



图10 我的团队

1. 在[组建团队]页签下，输入团队名称，并单击<组建团队>，如下：



图11 组建团队

1. 如果需要邀请队友参加比赛，点击<生成组队密令>成功生成组队密令。点击<点击复制>可以复制命令给队友。如：



图12 生成组队密令

1. 如果你想加入其它团队，在[加入其它团队]页签里输入团队的名称并点击<搜索>。在搜索结果的右侧点击<申请加入>。待对方团队的队长审批以后，方可加入对方团队。

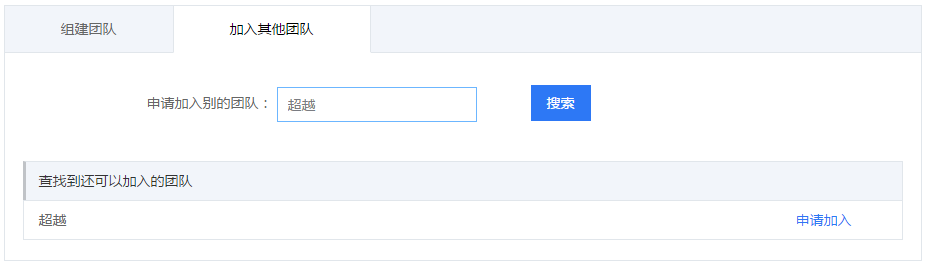


图13 加入其它团队

# 离线赛

对于离线赛，选手可以下载数据集到本地，然后通过天池网站提交本地文件上传结果。选手可以提交多次结果，但是新结果版本将覆盖旧版本。

## 下载数据

对于报名成功的离线赛选手，可以下载数据集到本地。

在比赛详情页面，打开[赛题与数据]页面，可以点击下载数据到本地。



图14 下载数据

## 提交结果

选手完成在本地进行的数据分析、特征提取、数据建模等一系列工作后，需要将本地的结果数据集通过天池网站提交。

1. 在比赛详情页面，点击左侧的[提交结果]页签，进入如下界面：



图15 提交结果界面

1. 点击<提交结果>并上传本地结果数据集到天池网站。

# 平台赛-数加

一些大赛经常需要使用分布式数据处理平台来处理比较大的数据。这里我们将介绍如何通过数加平台进行平台赛。

当选手报名平台赛后，天池官方团队将会给选手开通数加权限，并分配数加RAM账号信息。

## 认证

在进行平台赛之前，选手必须通过支付宝实名认证或者身份认证。 实名认证可以有效验证选手是否开了多个账号从而保证比赛的公平性。

1. 成功登录天池官网后，点击右上角的用户名进入到[个人中心]页面。点击[认证]，并在如下页面完成支付宝实名认证。 

图16 认证页面

1. 对于一些没有支付宝账号的选手，比如外籍选手，可以点击  进行身份认证：



图17 身份认证页面

1. 选择证件类型并按照图示的要求拍照上传。待天池官方团队审核通过后，即认证成功。

## 登录数加平台

### 获取RAM账号信息

当用户报名成功、认证通过并开通数加平台后，就可以用分配的RAM账号信息登录数加平台。

1. 登录天池官网，进入[个人中心]页面。打开[我的RAM账号]页签可以查看自己登录数加平台的账号、密码以及AK，如下图：



图18 查看RAM账号

1. 复制保存好账号信息后，进入所报名的比赛详情页面，可以看见数加页签，如：



图19 数加页签



图20数加开发平台入口

**注意：**

选手也可以直接在浏览器中打开数加的URL(<https://ide.shuju.aliyun.com/>),再进行登录。

1. 数加登录主界面如下：



图21 数加登录主界面

1. 选择[RAM用户登录]页签并输入分配的RAM账号信息，如：



图22 RAM登录主界面

1. 点击<登陆>成功登录并进入你所在的项目空间首页，如：



图23 项目首页

## 读取/复制数据

当你进入自己所在的项目后，根据赛题数据的描述，可以读取或者拷贝对应的表到自己的项目空间。

具体方法是直接在数据开发页面创建SQL节点并执行语句读取赛题数据里的表。需要注意的是，您必须在表名前面加上对应的项目空间的前缀。

例如：如果项目名称为：odps\_tc\_257100\_f673506e024，你在读取表的时候必须写成odps\_tc\_257100\_f673506e024.tianchi\_fresh\_comp\_train\_user\_online。

为了方便操作，你可以拷贝相关表到自己的项目空间，在自己的项目空间执行如下SQL：

Create table if not exists tianchi\_fresh\_comp\_train\_user\_online as select \* from odps\_tc\_257100\_f673506e024.tianchi\_fresh\_comp\_train\_user\_online;

注意：

关于如何在数据开发平台执行SQL，将在下面一章节介绍。

## 数据开发

在[数据开发]页签下，你可以使用ODPS SQL\ODPS MR\UDF\Graph\机器学习平台完成你所需要实现的数据挖掘、数据建模、提交结果等一系列步骤。

### 创建任务

1. 登录数加平台后，打开[数据开发]页签，项目首页显示如下：



图24 数据开发首页

1. 创建工作流文件目录。

打开“开发面板”并右键“任务开发”，弹出右键菜单栏。



图25 右键菜单栏

点击<新建文件夹>弹出[新建目录]界面，配置如下：:

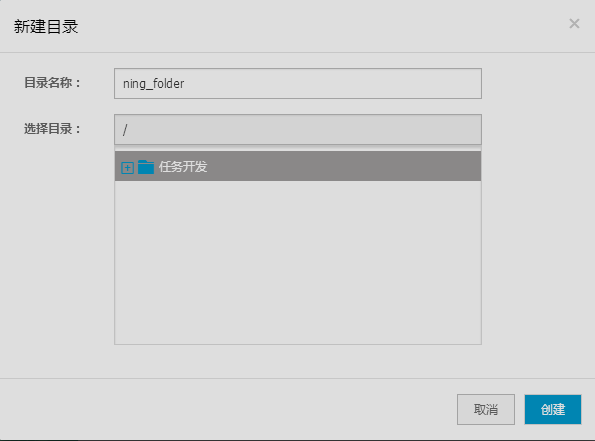


图26 新建目录

点击<创建>成功创建一个目录。

1. 新建任务

右键目录名弹出右键菜单，如：

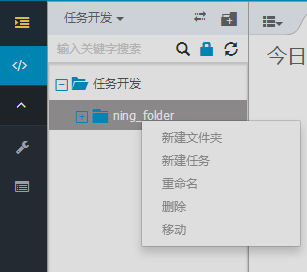


图27 右键目录菜单

点击<新建任务>弹出[新建任务]页面，配置如下：

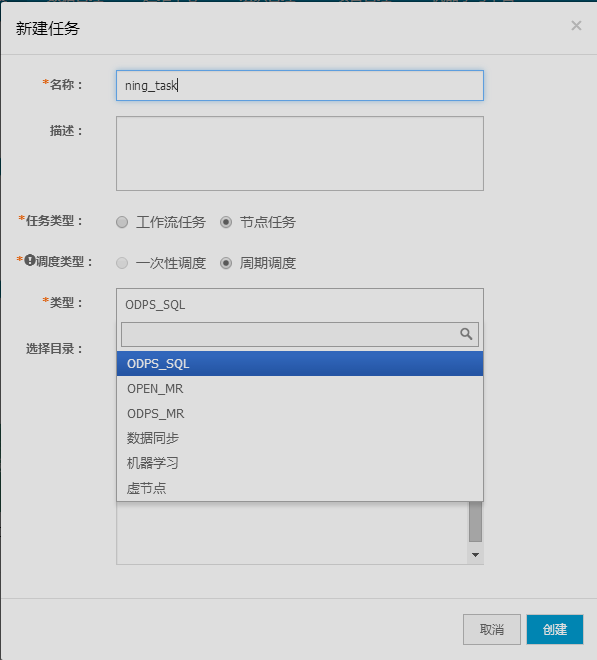


图28 新建任务

在新建任务弹出框中填写各配置项。参数说明如下：

* **名称**：支持数字、字母、下划线组合。
* **描述**：针对当前任务的简单描述，支持中文、字母、数字、下划线组合。
* **任务类型**：包括工作流任务和节点任务两种类型，工作流任务中可以包括多个节点任务。
* **调度类型**：调度类型支持一次性调度和周期调度两种类型，且成功创建工作流之后无法修改。一次性调度的工作流属性和节点属性中不包含调度属性，同时在工作流开发面板中可直接运行当前工作流。
* **类型**：类型目前支持ODPS\_SQL,ODPS\_MR,机器学习。对于天池选手，数据同步和OPEN\_MR暂不支持。
* **选择目录**：选择该任务所应归属的文件树。

此处以工作流任务为例，创建一次性调度任务。点击<创建>成功创建一个任务，如下：

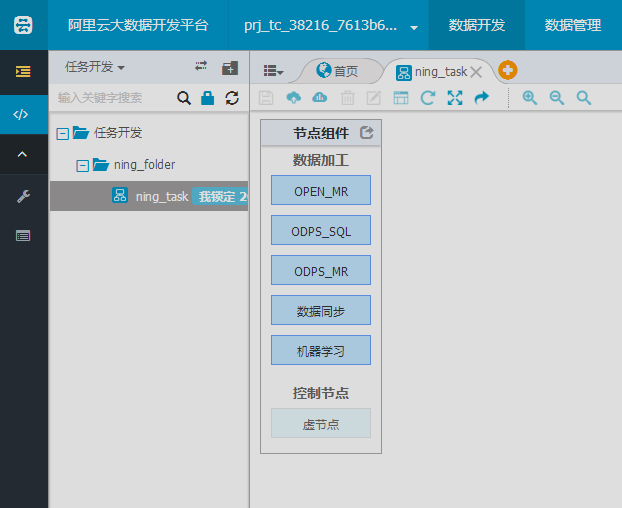


图29 成功创建工作流任务

### 运行ODPS SQL

1. 新建ODPS SQL节点。

在上图中，拽动[ODPS\_SQL]到界面右侧，自动弹出[新建节点]界面，配置如下：

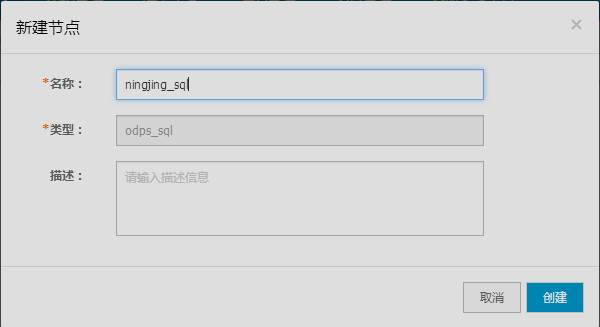


图30 新建SQL节点

点击<创建>成功创建SQL节点。双击该SQL节点，你可以运行ODPS SQL(这里我们执行拷贝表到本项目空间操作)，如下：

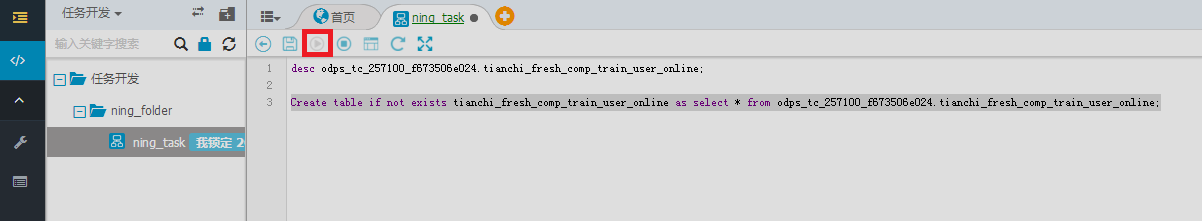


图31 运行ODPS SQL

### 运行OPEN MR

#### maven 安装

1. 下载地址<http://mirrors.hust.edu.cn/apache/maven/maven-3/3.3.3/binaries/，win7下载并解压apache-maven-3.3.3-bin.zip>。
2. 新建环境变量M2\_HOME (你的解压目录)。

目录：D:\eclipseHadoop1x\apache-maven-3.3.3

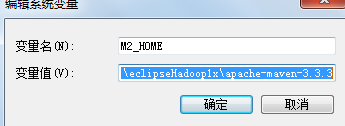


图32 配置环境变量

PATH后面添加%M2\_HOME%\bin;

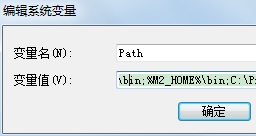


图33配置path

1. 测试cmd

mvn –v

如图显示即为安装成功：

图34 测试maven是否配置成功

1. 安装Eclipse的Maven插件：参考<http://www.eclipse.org/m2e/>(或百度eclipse安装maven插件)因为每个人eclipse版本情况无法保证每一种都能成功。

可参考：<http://jingyan.baidu.com/article/60ccbceb01de4d64cbb19756.html>

<http://blog.csdn.net/wode_dream/article/details/38052639>

现提供离线安装安装Eclipse的Maven插件过程：

* 下载eclipse-maven3-plugin.7z

网盘地址：<http://pan.baidu.com/share/link?shareid=1456505411&uk=2098767677&fid=180501298193861>

* 将解压后的子文件夹features和plugins的jar包文件分别导入Eclipse安装目录下的features和plugins文件夹内，然后重启Eclipse就可以在Preferences中看到Maven选项了。即Maven插件配置成功。

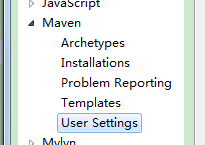


图35 安装maven插件

注意：

如果重启后显示弹窗修改Install\_JRE，修改VM参数-Dmaven.multiModuleProjectDirectory=$M2\_HOME。

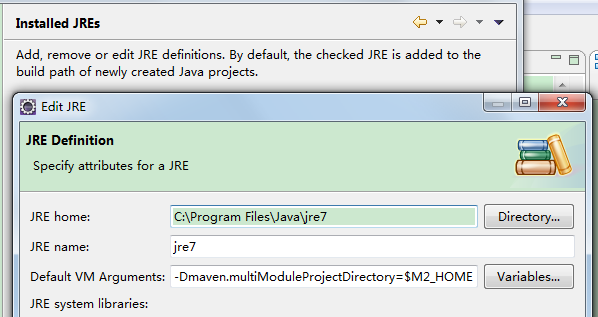
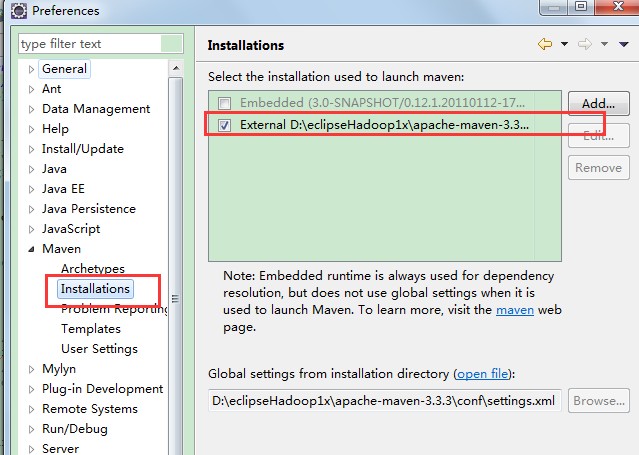


图36 安装jres

Maven的Eclipse插件配置，修改成自己的maven地址:



Maven的Eclipse插件配置，修改成自己的setting地址:

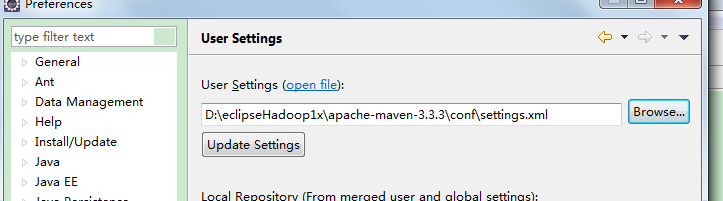


图37 user settings

1. 添加远程repository

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：

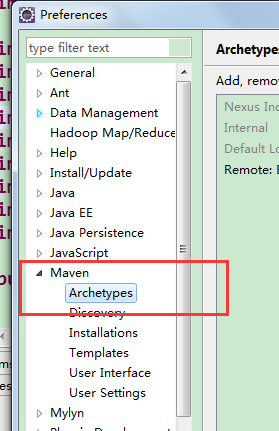


图38添加远程repository

在打开的对话框中， Catalog File 填入 http://maven.sdk.de.yushanfang.com/SNAPSHOT ， Description 填入 Base Archetypes

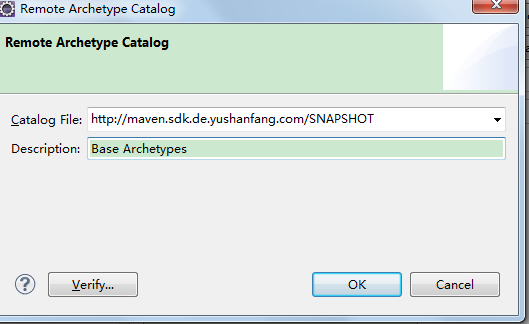


图39 配置远程repository

#### 新建项目

1. 在Eclipse中依次点击 File -> New -> Project.选择maven项目:

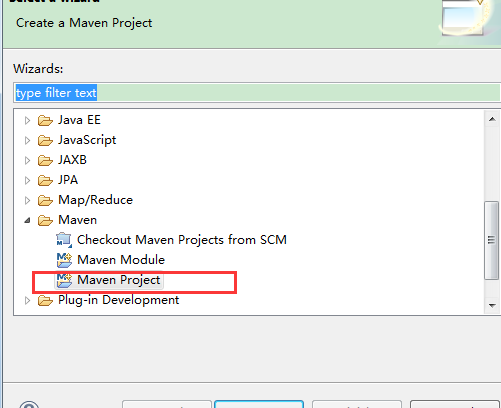


图40 选择maven工程

1. 配置并点击<Next>，如下图：

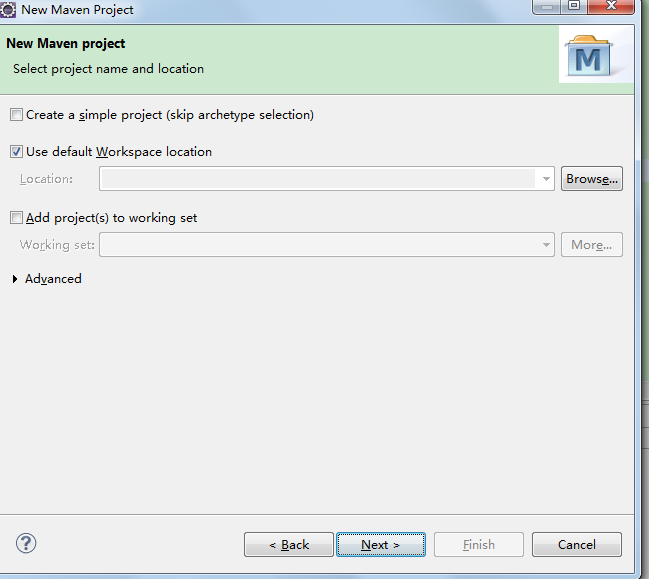


图41 Maven项目配置（1）

1. 按照下图选择：

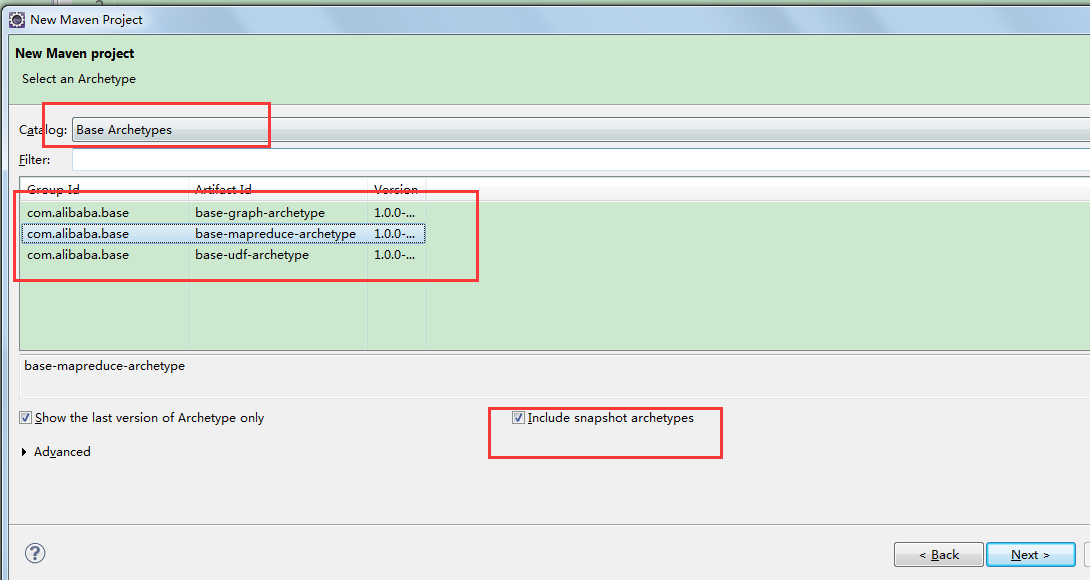


图42 项目配置（2）

1. 点击<Next>并填入项目名，其他选项默认就行，如下图：

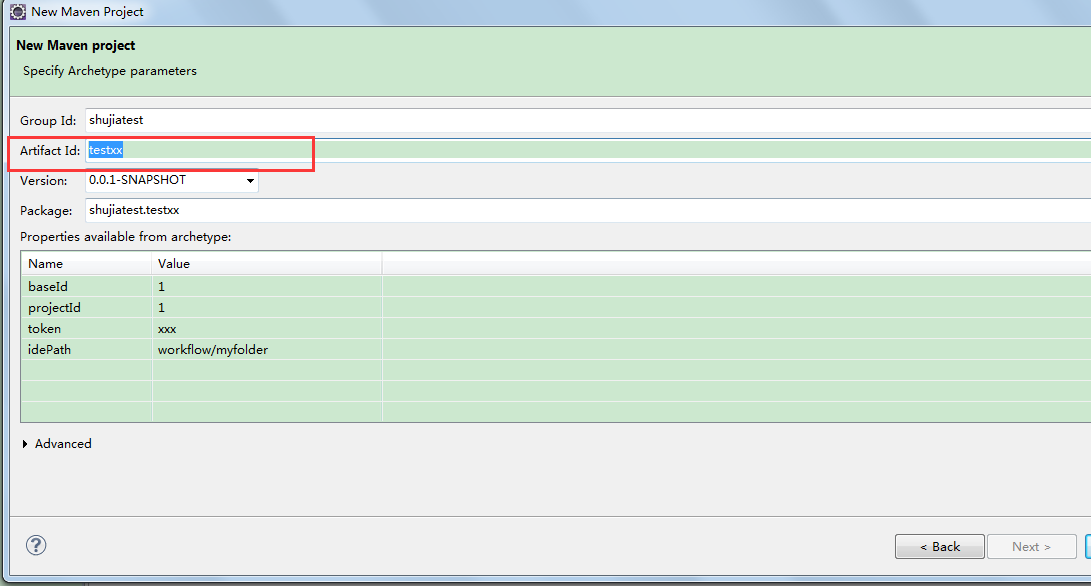


图43 项目配置（3）

至此环境搭建OK。

#### 数加平台上使用OPEN\_MR示例

1. Eclipse里编译代码

使用的代码-附上示例代码：

package hadoop.TianChiMapreduce;

import java.io.IOException;

import java.util.Iterator;

import com.aliyun.odps.data.Record;

import com.aliyun.odps.data.TableInfo;

import com.aliyun.odps.mapred.JobClient;

import com.aliyun.odps.mapred.MapperBase;

import com.aliyun.odps.mapred.ReducerBase;

import com.aliyun.odps.mapred.TaskContext;

import com.aliyun.odps.mapred.conf.JobConf;

import com.aliyun.odps.mapred.utils.InputUtils;

import com.aliyun.odps.mapred.utils.OutputUtils;

import com.aliyun.odps.mapred.utils.SchemaUtils;

public class WordCount {

public static class TokenizerMapper extends MapperBase {

private Record word;

private Record one;

@Override

public void setup(TaskContext context) throws IOException {

word = context.createMapOutputKeyRecord();

one = context.createMapOutputValueRecord();

one.set(new Object[] { 1L });

System.out.println("TaskID:" + context.getTaskID().toString());

}

@Override

public void map(long recordNum, Record record, TaskContext context)

throws IOException {

for (int i = 0; i < record.getColumnCount(); i++) {

word.set(new Object[] { record.get(0).toString() });

context.write(word, one);

}

}

}

/\*\*

\* A combiner class that combines map output by sum them.

\*\*/

public static class SumCombiner extends ReducerBase {

private Record count;

@Override

public void setup(TaskContext context) throws IOException {

count = context.createMapOutputValueRecord();

}

@Override

public void reduce(Record key, Iterator<Record> values, TaskContext context)

throws IOException {

long c = 0;

while (values.hasNext()) {

Record val = values.next();

c += (Long) val.get(0);

}

count.set(0, c);

context.write(key, count);

}

}

/\*\*

\* A reducer class that just emits the sum of the input values.

\*\*/

public static class SumReducer extends ReducerBase {

private Record result = null;

@Override

public void setup(TaskContext context) throws IOException {

result = context.createOutputRecord();

}

@Override

public void reduce(Record key, Iterator<Record> values, TaskContext context)

throws IOException {

long count = 0;

while (values.hasNext()) {

Record val = values.next();

count += (Long) val.get(0);

}

result.set(0, key.get(0));

result.set(1, count);

context.write(result);

}

}

public static void main(String[] args) throws Exception {

if (args.length != 2) {

System.err.println("Usage: WordCount <in\_table> <out\_table>");

System.exit(2);

}

JobConf job = new JobConf();

job.setMapperClass(TokenizerMapper.class);

job.setCombinerClass(SumCombiner.class);

job.setReducerClass(SumReducer.class);

job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));

job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));

InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(), job);

OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(), job);

JobClient.runJob(job);

}

}

1. 编译完成，项目结构如下：如下图标红的代码文件。

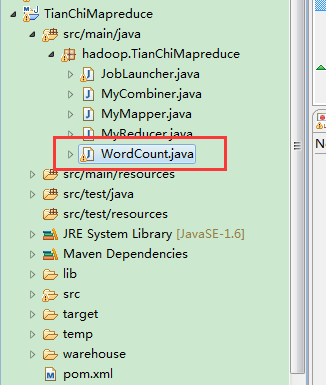


图44 代码文件

1. 右键该java文件并点击<Export>导出Jar。

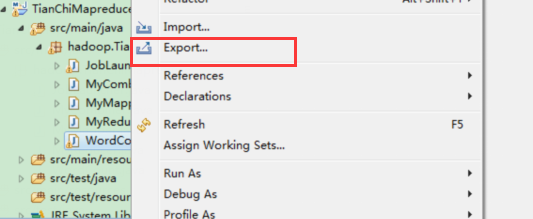


图45 导出按钮

1. 选择java目录下的jar file并点击<Next>导出jar：

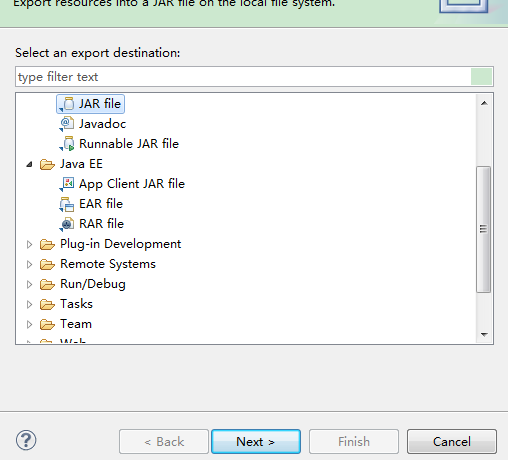


图46 导出文件选择

1. 选择导出的路径：

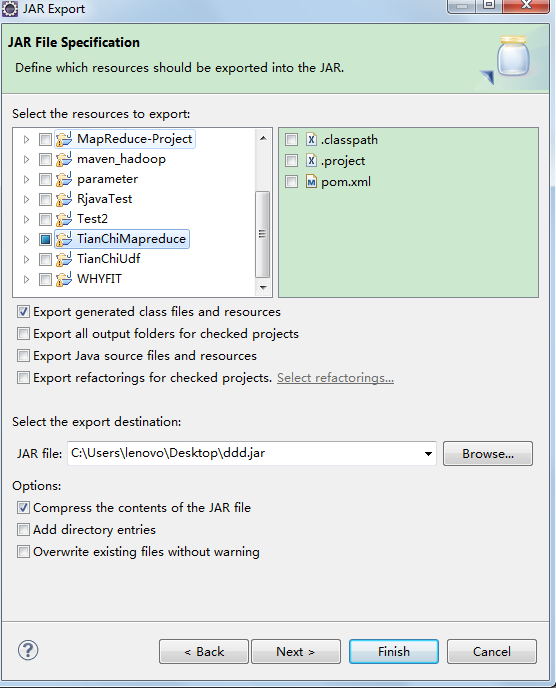


图47 选择导出路径

1. 选择运行的Main函数：

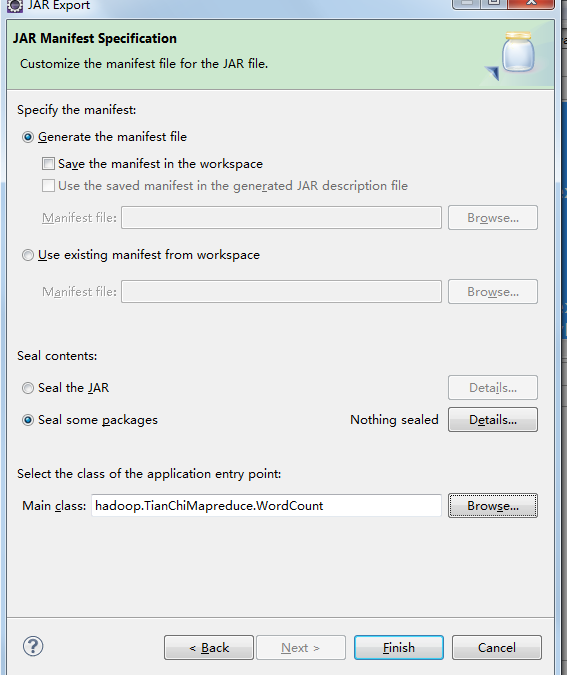


图48选择Main Class

1. 点击<Finish>。
2. 平台操作：

* 上传资源到数加平台；进入IDE后,点击加号->上传资源:

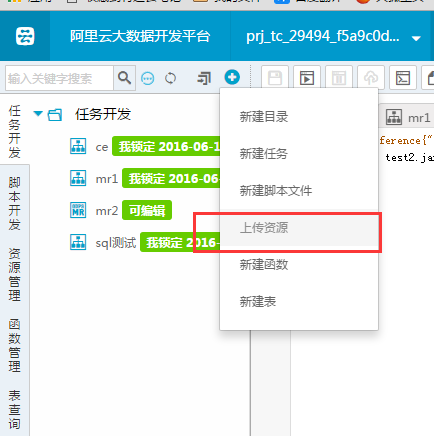


图49 上传资源按钮

* 输入名称、资源类型和上传的jar包，并提交：

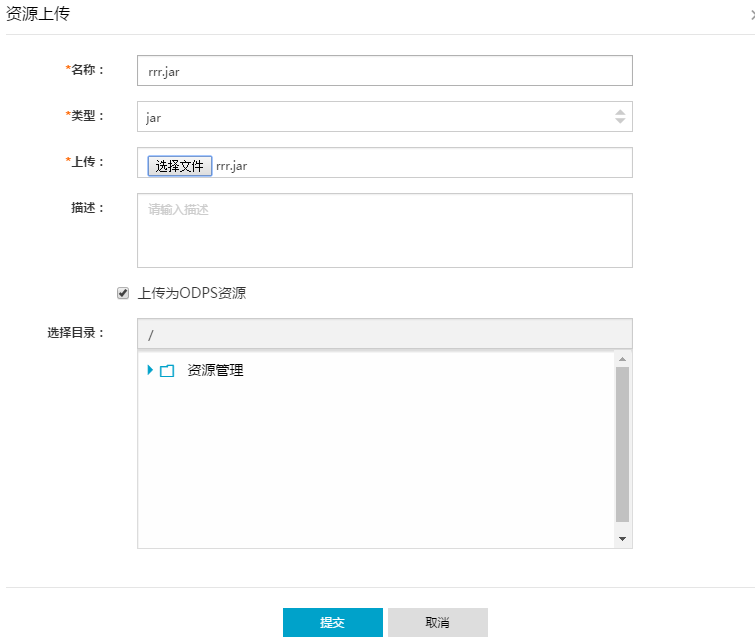


图50 上传资源配置

* 点击+号并点击<新建任务>:

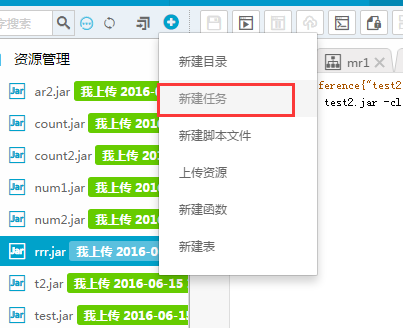


图51 新建任务按钮

* 填写任务名称，选择任务类型：



图52 新建任务配置

* 选择你新建的任务，拖动OPEN\_MR组件到右侧空白：



图53 OPEN\_MR节点

* 新建OPEN\_MR组件：



图54 新建节点

* 双击组件-按如下填写，前两个是你需要运行的jar，mapper和reducer是你要运行jar的map类和reudce类，输出表要提前创建好，key和value对应输出表的字段

建表sql：create table if not exists wcout1 (key string,cnt bigint);

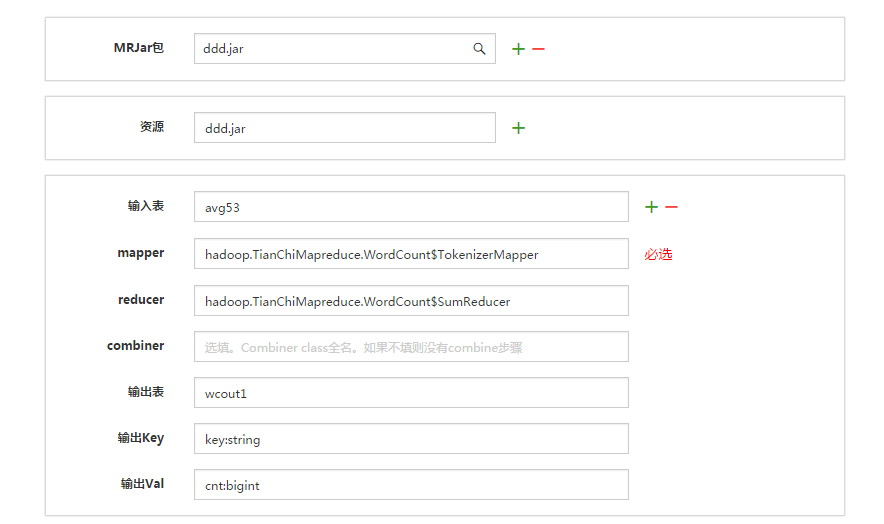
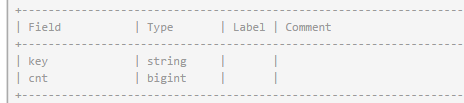


图55 open MR配置

* 填写好之后保存运行，至此wordcount程序运行完成。
* 附录：

wcout1的表结构



注意：

* 关于更多ODPS SQL的语法以及UDF、MapReduce的介绍，请参考 (<https://help.aliyun.com/product/8314999_27797.html?spm=5176.doc27901.3.1.SVp4z8>)。
* ODPS graph的用法类似与ODPS MR。具体的使用示例请参考<https://help.aliyun.com/product/8314999_27797.html?spm=5176.doc27901.3.1.SVp4z8的graph>的graph章节。
* UDF的用法类似于ODPS MR。不同之处在于上传资源后需要通过“函数管理”新建函数并与jar资源关联，具体的方法请参考(<https://help.aliyun.com/document_detail/30270.html?spm=5176.doc30281.6.144.WPNjIG>)。

## 机器学习平台

### 简介

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操作手册，请参考<https://help.aliyun.com/document_detail/30348.html?spm=5176.doc30251.6.251.rxVLj4>。

在机器学习平台上，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）
* 数据存储

在数加IDE打开“机器学习平台”页签：

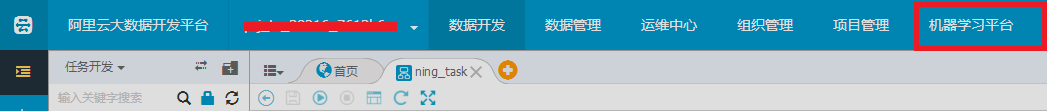


图56 机器学习平台登录入口

选择租户和空间后，进行如下主界面：

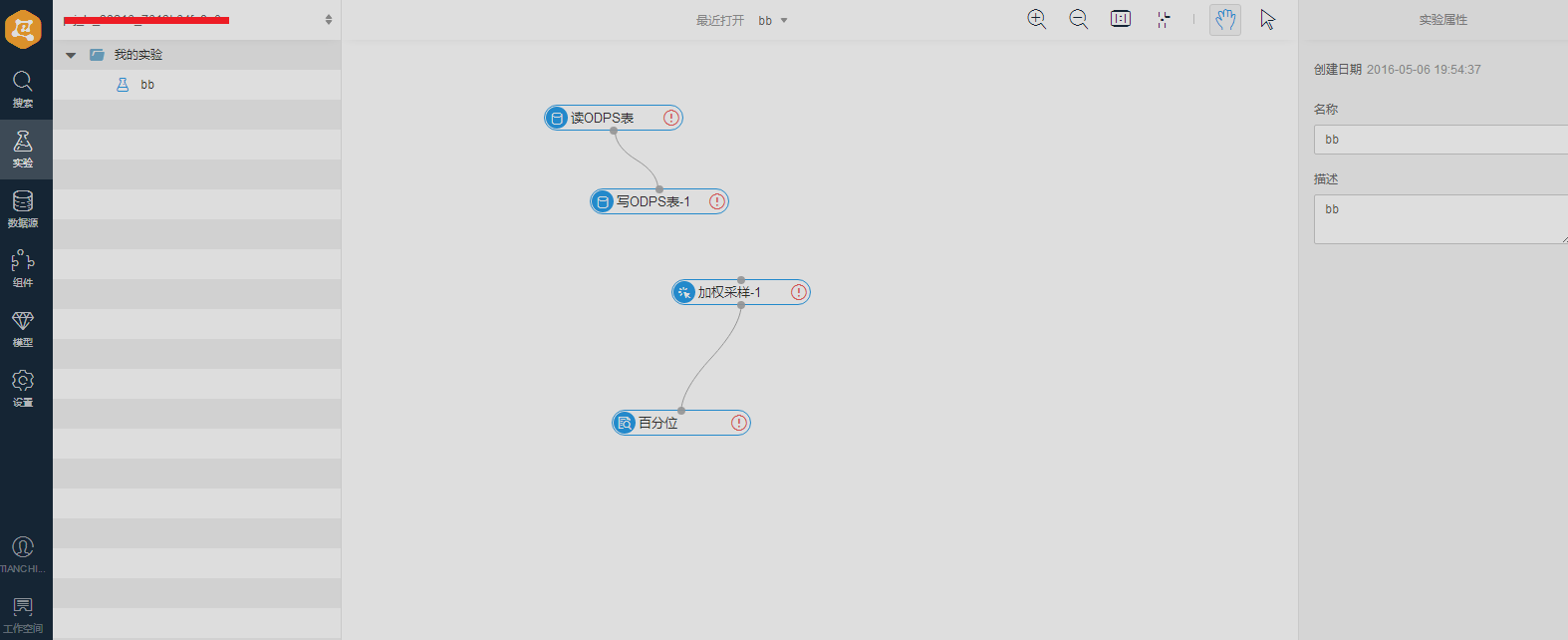


图57 机器学习平台

注意：

* 避免直接使用全量数据。建议在建模前完成数据抽样。

### 数据准备

1. 在机器学习平台页面，点击左下角的或者右键点击左上角的“我的实验”并在弹出的按钮列表里点击<新建空白实验>，弹出[新建实验]对话框， 输入名称和描述并点击<新建>：

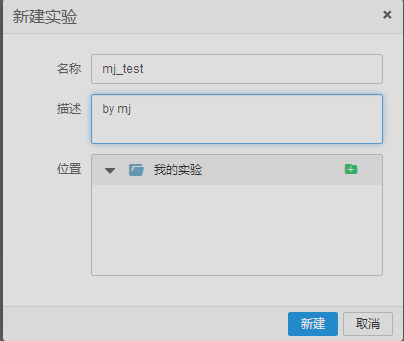


图58新建实验

1. 左侧导航切换到组件栏；向画布中拖入“读odps表”组件。点击该组件，在右侧表选择栏填入odps表。（注：这里的表一般为本项目空间的表，表名无需带上project名称。如果需要引用跨项目空间的表，需要在表前加入项目前缀，对天池选手来说，此功能用得不多。）

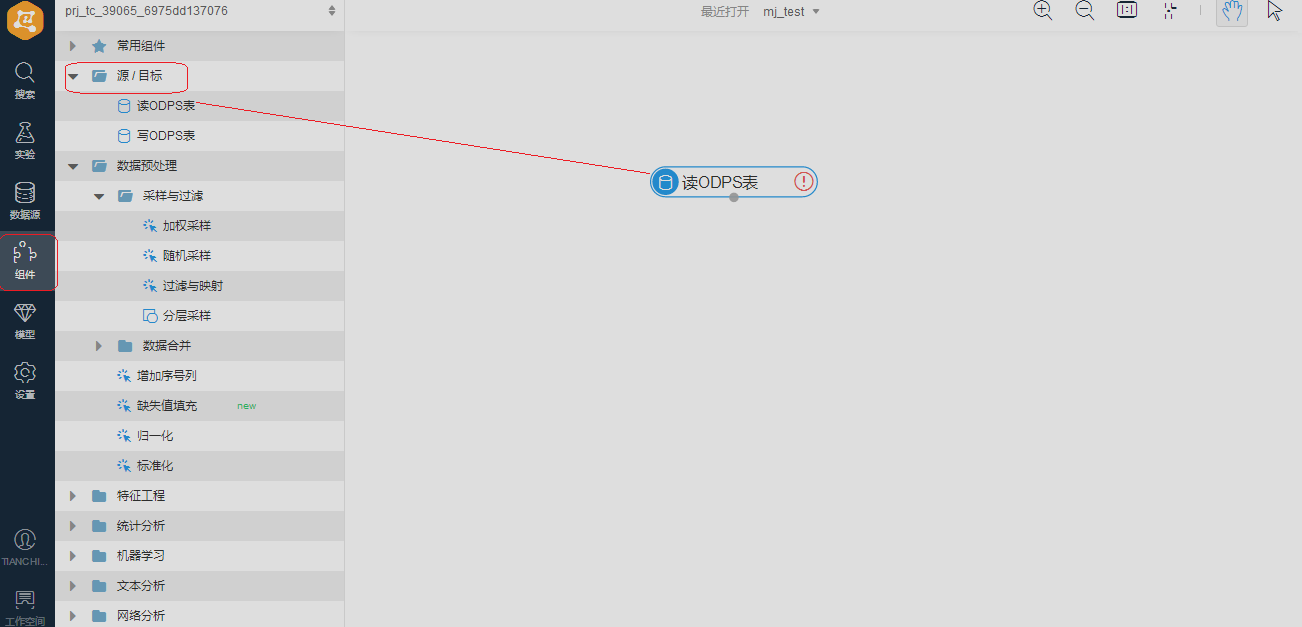


图59 读ODPS表组件

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60字段信息

### 数据预处理

数据预处理包括采样过滤和数据合并。下面以“缺失值填充”为例描述步骤。

1. 从数据预处理栏拖入**缺失值填充**组件到画布。

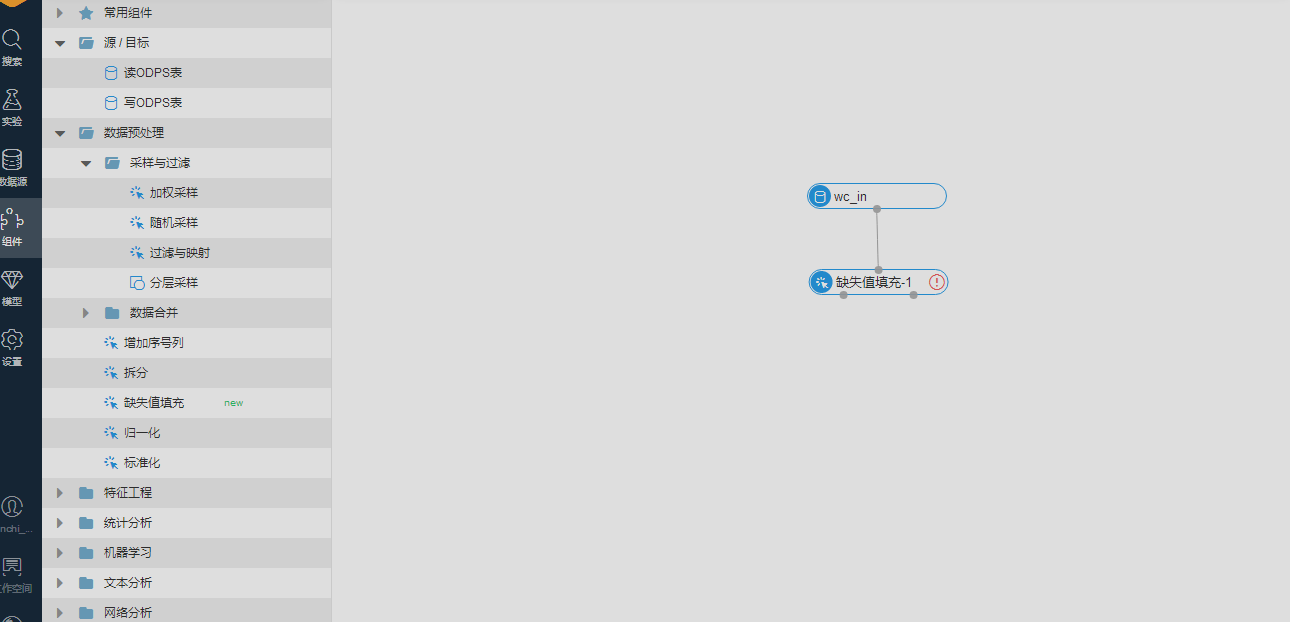


图61缺失值填充组件

1. 右击**缺失值填充**，进行**字段设置**。点击**选择字段列**，勾选所有数值型列，并用最大值进行填充。

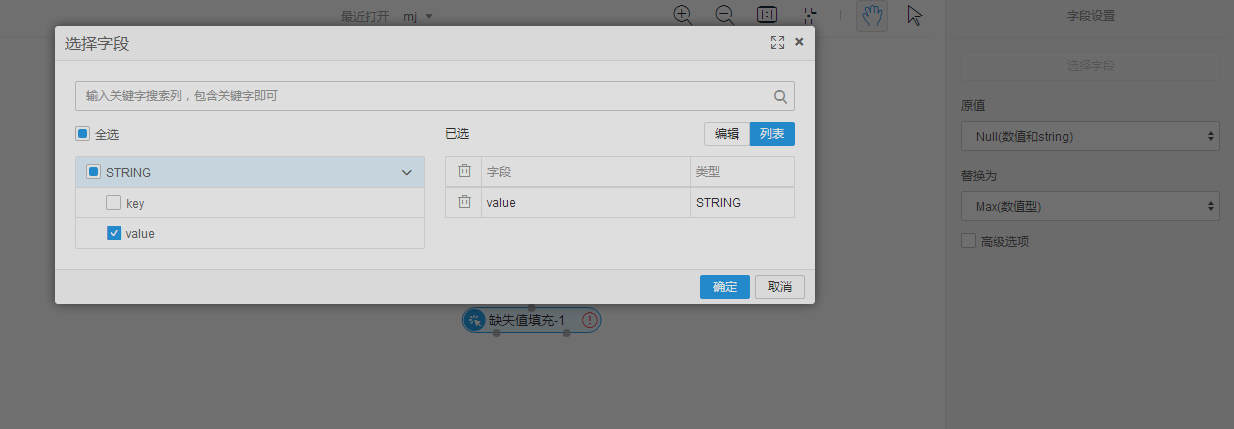


图62选取填充字段

注意：将数字进行缺失值填充是模型训练前数据处理很重要的步骤之一

1. 将**拆分**组件拖到画布，步骤同上。

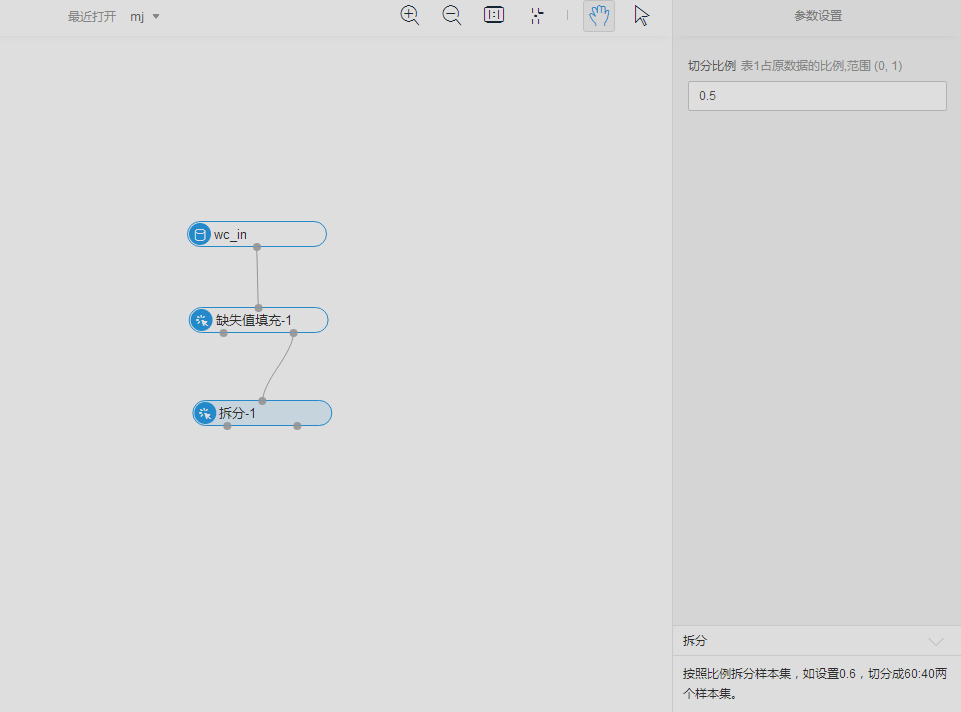


图63 拆分

此步骤的目的是将数据拆分成两份，50%作为模型训练集，50%作为模型预测集。

### 建模和评估

数据集准备好，就可以跑模型。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：

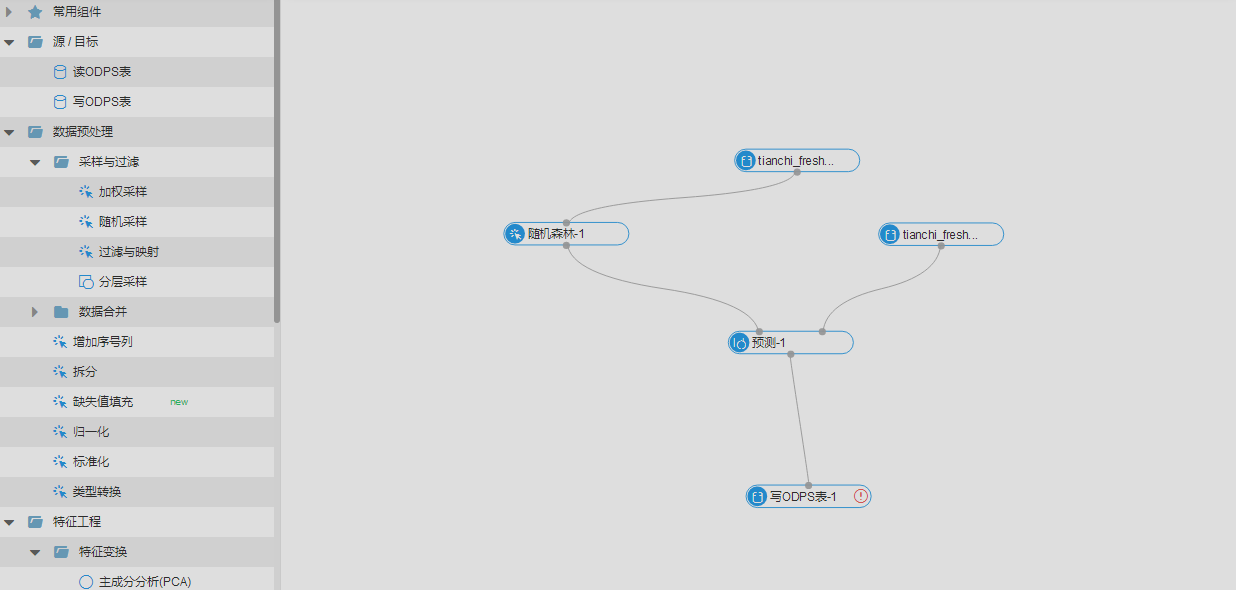


图64 典型的训练预测流程

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：

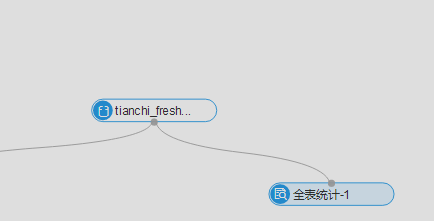


图65全表统计组件

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66 查看分析报告

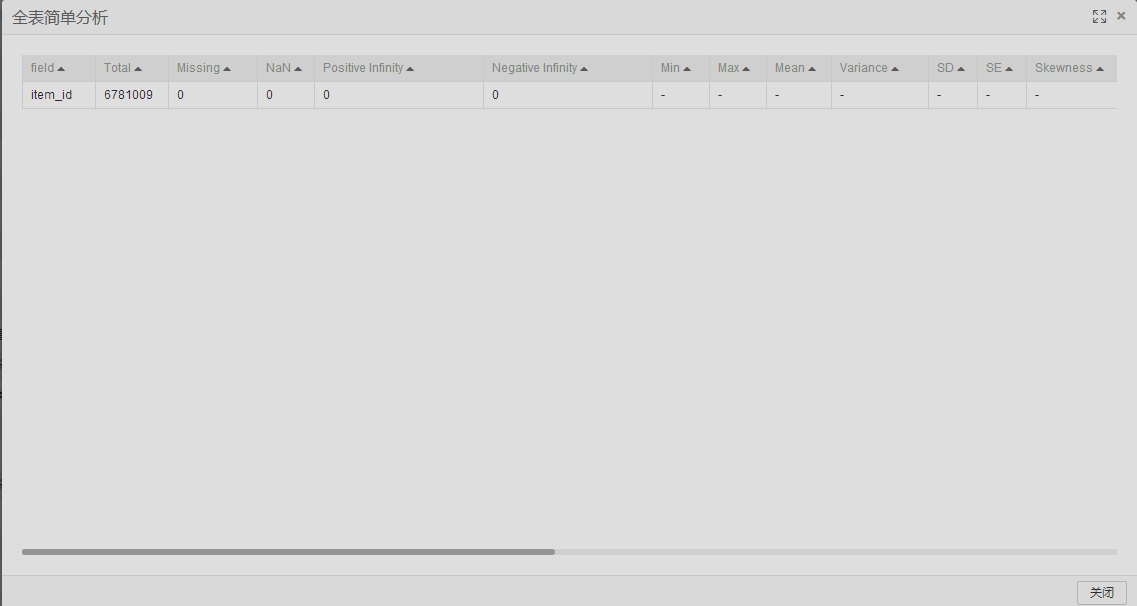


图67 分析结果

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/二分类/逻辑回归二分类**组件，
2. **读ODPS表**组件，选择输入表。
3. 逻辑回归组件，“字段设置”TAB中，选择训练特征列和目标列（目标列即目标变量、分类标签），其中训练特征列选择特征工程所构建的特征，目标列选择label字段；切换到“参数设置”Tab，设置目标基准值（正则系数）为1，其它几个参数直接沿用了默认值。

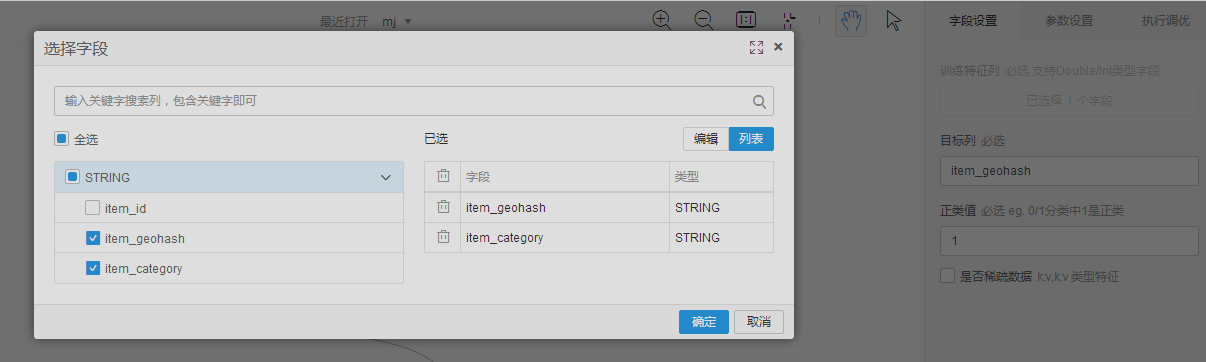


图68 逻辑回归-字段选择

注：

* 训练特征列：支持double类型与bigint类型。
* 目标列：只能选择非输入列的其它列，支持string类型、double类型和bigint类型。
* 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。

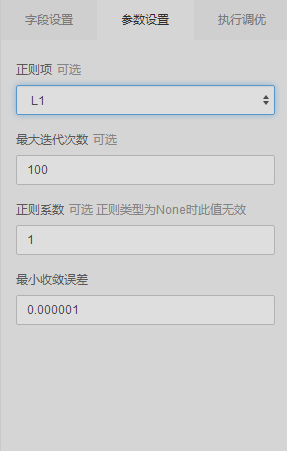


图69 逻辑回归参数设置

附：参数说明

* 最大迭代数：（可选）L-BFGS的最大迭代次数，默认是100
* 最小收敛误差：（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
* 正则项：（可选）正则化类型，可以选择‘L1、‘L2’、‘None’，默认为‘None’。
* 正则系数：（可选）正则项系数，默认为 1.0；当正则项为 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70 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。
* 森林中树的个数：范围(0, 1000]
* 单颗树随机特征数：（可选）单颗树在生成时，随机选择的特征个数。范围[1,N],默认log2N。
* 单颗树的最大深度：（可选）单颗树的最大深度。-1表示完全生长。范围[1, ∞)
* 叶节点数据的最小个数：（可选）叶节点数据的最小个数。最小个数为2
* 叶节点数据个数占父节点的最小比例：（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[0,1]
* 单颗2输入的随机数据个数：（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。

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：

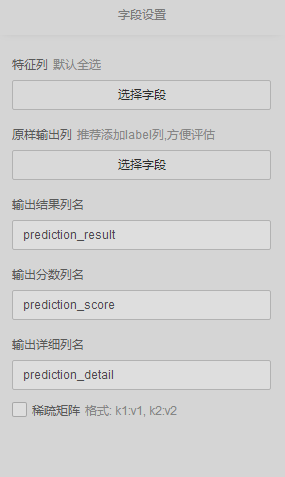


图71预测组件参数设置

注意：如果勾选‘稀疏矩阵’，表示输入预测集的数据格式为稀疏矩阵 。

##### 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。

下面以混淆矩阵为例说明。

* 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。

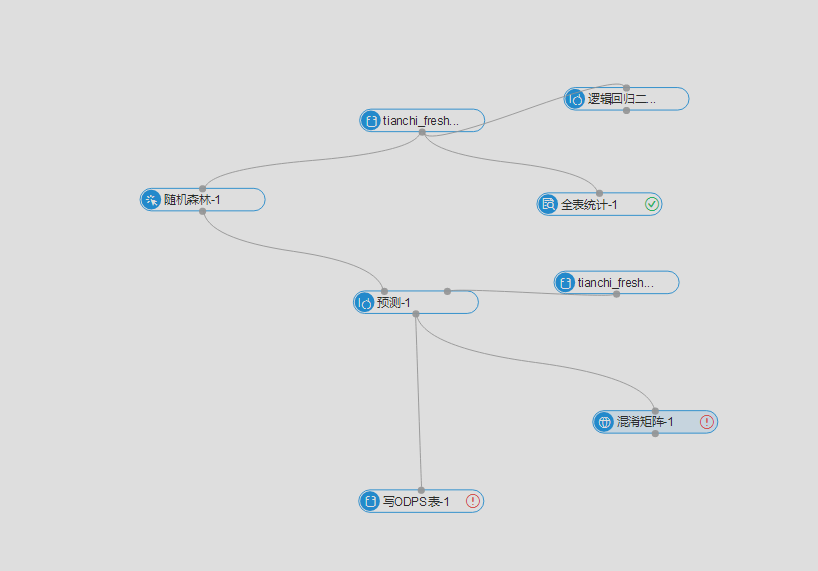


图72 混淆矩阵组件设置

1. 在读ODPS表组件中选择输入表；
2. 设置混淆矩阵计算的参数：用户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。

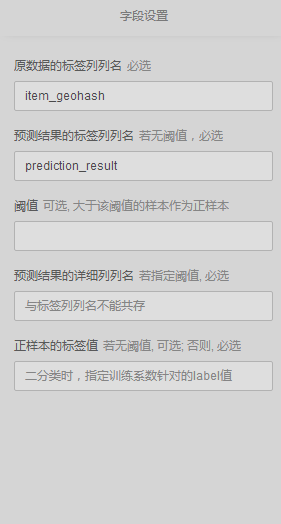


图73混淆矩阵参数设置

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>。

## 提交结果

经过混淆矩阵的评估，可以选择随机森林，对测试集进行训练和测试，生成赛题需求的结果提交。

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

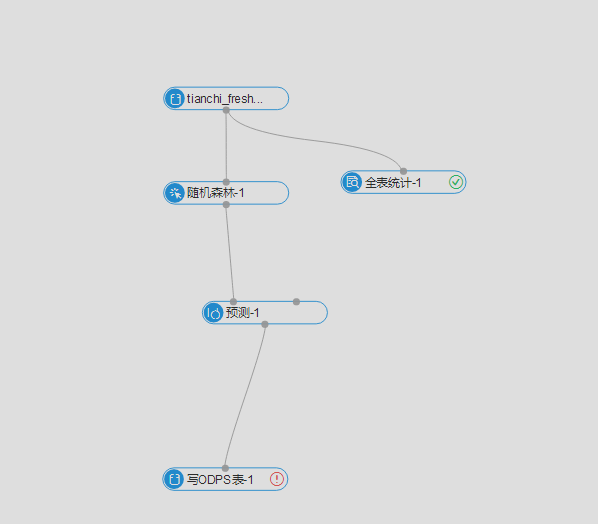


图74 拖入组件并连接

1. 选择输入表，点击运行，运行完成后可以用sql查看结果表。
2. 设置随机森林参数，选择字段和标签列。
3. 设置写ODPS表为t\_test\_result（将模型预测出来的结果表命名为t\_test\_result）。
4. 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。predict\_score是预测结果。

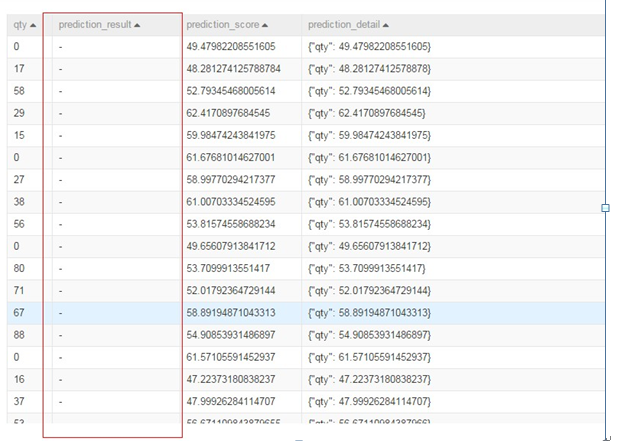


图68预测结果

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的结果表中。

# 附录

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。

常用命令

show tables;--显示项目空间中的表名。

desc table\_name; --检查表结构。

show partitions table\_name; -- 查看表的分区信息。

文档参考

1. 数加文档

<https://help.aliyun.com/product/9091750_30212.html>

1. 算法平台相关文档



1. ODPS用户手册

<https://help.aliyun.com/product/8314999_27797.html?spm=5176.doc27901.3.1.Wbs6qm>

修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| Revision NO. | Revised Date | Revised by | Reason for Revision |
| V1 | 2016-5-19 | 宁晶 | 新增 |
| V2 | 2016-6-16 | 宁晶 | 更改mr为open mr的示例 |
|  |  |  |  |
|  |  |  |  |