大数据平台御膳房

最佳实践

V1.0

2015 年 6 月

目 录

前　言

本文档为御膳房的演示案例。文档以真实线上挖掘分析案例作为牵引，详细描述了御膳房中分析、建模工具的使用，以及如何将在开发环境建立的模型部署上线到生产环境。

文档按照数据挖掘分析流程组织，以解决案例问题为主线，包括如下七个部分：

- ✧ **一、御膳房简介。**介绍御膳房的基本情况。
- ✧ **二、案例说明。**描述本文档将演示的真实线上业务案例的描述，该案例探索店铺优惠券的发放优化。
- ✧ **三、创建租户。**详细介绍了使用御膳房的前置工作，包括用户注册、创建组织、邀请成员、创建项目等。
- ✧ **四、数据上传。**介绍如何利用御膳房提供的数据同步工具，将客户存储在不同数据库的数据上传到御膳房。
- ✧ **五、数据商品购买。**简要介绍如何从数据市场购买数据商品，并授权给指定项目及个人。
- ✧ **六、数据开发。**介绍如何利用御膳房提供的工具，定义案例数据的格式和含义，如何利用 SQL 进行特征的抽取、在算法平台和数据探索环境进行数据的探索，完成满足业务需求的开发工作。
- ✧ **七、数据导出。**介绍如何将前述数据探索过程得到的结果 ,导出到客户的业务库。
- ✧ **八、总结。**对全文内容进行总结概况。

为配合本文档的演示，我们在御膳房数据市场开放了本文档案例数据集[1]，供测试和参考。文档中在演示挖掘和建模过程的同时，给出了主要的工具操作图示和相关的处理代码，可在样例数据集上直接运行测试。

[1] 该样例数据集为淘宝真实交易数据的采样，为保护用户隐私，我们对敏感字段进行了模糊处理。

一、御膳房简介

御膳房[1]是基于公共云计算的数据管理、计算和交换平台，为政府机构、企业、科研机构、第三方软件服务商等客户，提供大数据管理、开发和生产计算的能力，同时让客户间能交换数据，解决数据管理、应用、流通的场景需求，帮助客户实现商业价值。

图 1 御膳房首页

作为本文档的前置条件，御膳房文档中心[2]给出了基本操作说明，建议读者先行阅读。

[1] 御膳房入口地址：http://yushanfang.com/

[2] 文档地址：http://setting.tenant.yushanfang.com/portal/help/index.html

二、案例说明

本文档引用阿里数据挖掘团队开发的优惠券预测模型，在御膳房中进行主体流程的实现。目标定位是利用真实案例演示御膳房的使用过程[1]。

整个优惠券预测的业务目标是预测买家是否会在某个特定店铺购买某类商品（例如连衣裙），并将优惠券发放给最有可能购买的人群，从而提升转化率和客单价。业务目标落实到算法模型层面有两个核心的基本问题：一是优惠券发给谁？（即客户群选择）；二是发什么样的优惠券？（即应该满多少减多少最佳）。客户群选择实际上是预测买家的购买倾向，并依据购买倾向的强弱来给出排序的结果，落实到学习模型层面来解决；而发放何种优惠券更多是业务运营层面的考虑，可以依据卖家的满减阈值和折扣力度来综合确定。简单来讲，本案例完成了从买家的历史行为数据中计算得到最可能购买类目商品的买家群体，然后根据商家的营销策略来发放合适的优惠券，来达到提升购买率和客单价的目标。

下面给出具体示例。首先，优惠券预测模型目前已应用于江湖策[2]的精准营销功能中，产品界面如图 2.1 所示。通过后台模型可以细分确定待发放优惠券的目标客户群体，然后根据设置的方案可以执行优惠券的方法以触达买家。

图 2.1 优惠券发放落地示意图

[1] 因此这里着重在如何应用挖掘平台提供的分析工具去实现业务，而对于模型本身仅给出相应匹配程度的介绍，且在工程实现过程中对于数据和算法均进行了适当的简化。

[2] 江湖策的入口地址为：http://liuliang.taobao.com/

其次 数据样本和方法举例。设买家 U1 和 U2 近 7 天与卖家 S 交互信息如表 2.1 所示。

买家	卖家	叶子类目	浏览次数	收藏次数	添加购物车次数	购买次数
U1	S	牛仔裤	23	1	0	0
U1	S	连衣裙	11	0	0	0
U1	S	小背心	29	0	1	0
U2	S	T 恤	17	0	1	0
U2	S	牛仔裤	15	0	0	1

表 2.1 买家和买家交互行为表举例

表中显示买家 U1 和 U2 可能对卖家 S 店铺感兴趣的类目。但由于 U2 已经购买了牛仔裤，认为其短期不再需求。由于实际有交互的买家数量较多，我们需要圈定购买倾向最明显的买家来发放适合的优惠券。可以有多种方法来完成这一操作。本文档中将引用其中两种方法：一是根据业务经验设置规则；二是通过学习算法来预测。例如规则认为添加购物车是比收藏和浏览更强烈的兴趣，则筛选出买家 U1 可能买卖家 S 的小背心，买家 U2 可能买卖家 S 的 T 恤。而 U1 和 U2 比较时因为 U1 对于小背心的浏览次数高于 U2 对于 T 恤的浏览次数 ,可以认为 U1 对于小背心的购买倾向更强烈 ,因此优先选择给 U1 发放优惠券。

最后，则需要确定发放何种优惠券了。根据卖家调研，满值（如到 300 元才优惠）的设置一般是为了提升客单价，即若客户在某店铺的花费均值在 150 元左右，则可以设置满 200 才给减，吸引客户再多买一些商品；据业务经验，可设置的满值为目标商品客单价的 1.2 倍，并将结果规约到 50 元的倍数。卖家设置其折扣比例是 10%。则可以得到优惠券的发放结果。参看表 2 的示例。这里卖家 S 给买家 U1 发放 15 元的优惠券，15 元这个值的计算过程是：客单价 110 元×1.2=132 元，规约到满值 150 元，然后折扣 10%得到 15 元。

买家	卖家	二级类目	客单价	满值	折扣
U1	S	牛仔裤	120	150	15
U1	S	连衣裙	165	200	20
U1	S	小背心	110	150	15
U2	S	T 恤	100	150	15
U2	S	牛仔裤	120	150	15

表 2.2 满减折扣举例

至此完成了业务背景的简介。后文将围绕这一案例在御膳房中予以完整的实现演示。

三、创建租户

首先，让我们登陆御膳房（http://www.yushanfang.com），开启一段美妙的大数据之旅吧！注：请使用 Chrome 浏览器。

3.1 用户注册

1、点击御膳房首页右上角“登录”按钮进入登录页面，允许使用淘宝、阿里云账号登录系统。

注：也可以使用手机、邮箱登陆。

图 3.1.1 御膳房登陆界面

2、第一次登录，需填写注册信息。手机号、Email、联系人为必选项。

注：手机号应与淘宝账号绑定。

图 3.1.2 首次登陆填写注册信息

3.2 创建组织

一个组织对应于现实中的一个公司、机构或者其他可以独立承担法律责任的实体（以下称“其他实体”），它是一个多人协作的工作空间。一个组织下可以创建多个项目，一个项目可以拥有多个成员。一个用户可以创建或加入多个组织，也可以创建或加入多个项目。

并非所有用户都需要创建组织。仅是公司、机构或者其他实体等御膳房项目负责人需要创建并成为组织所有者，其他人（如开发、运维人员等）可以等待组织管理员的邀请加入。另外，淘宝子账号不能创建组织。

组织的名称必须唯一。邀请码可发邮件至 yushanfang@service.taobao.com 申请，也可以通过加入御膳房用户反馈群[1]联系御膳房小二进行申请。申请时，请说明申请理由。

[1]御膳房 1 群-用户反馈-技术支持 1：759773056。御膳房 2 群-用户反馈-技术支持 2: 1454609023。

图 3.2.1 创建组织页面

3.3 邀请成员

3.3.1 创建组织时邀请成员

邀请成员是让其他用户加入您的组织，在组织内做项目管理、数据开发、运维等工作。所邀请成员须先登录御膳房，完成上述注册环节。如暂时无成员，可点击跳过。

图 3.3.1 邀请组织成员

待所邀请的成员登录御膳房后，输入其注册时的邮箱或者手机号，即可增加成员。

3.3.2 组织创建后邀请成员

1、在御膳房首页，进入“我的工作台”。

图 3.3.2 邀请组织成员流程 1

2、点击“设置”后，点击“邀请成员加入组织”

图 3.3.3 邀请组织成员流程 2

3、输入待邀请成员注册时的账号、邮箱或手机号进行搜索，点击“添加成员，并发送邀请”

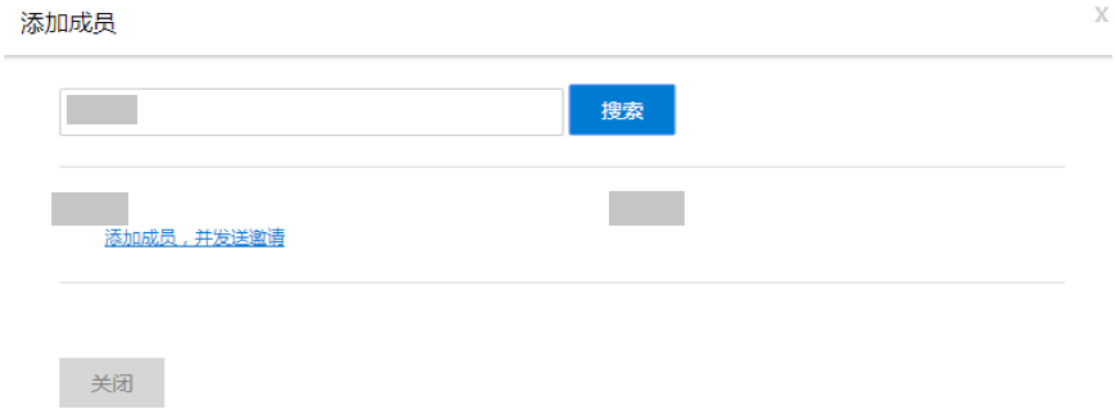


图 3.3.4 邀请组织成员流程 3

3.3.3 成员接受邀请即加入组织

邀请成员后，需要该成员登录御膳房并同意邀请。成员登录御膳房后点击“我的工作台”。然后点击接受邀请即可进入该组织。

图 3.3.5 邀请成员流程 4

3.4 创建项目

创建组织后，我们需要创建一个项目，为成员提供协同工作的空间。一般来说，你可以把要做的一件或一类事情，定义为一个项目，比如“广告精准投放系统”。

首先，进入“我的工作台”，点击“数据引擎”。

御膳房设定了“私有区”和“交换区”进行项目和数据的隔离。用户可以在御膳房帮助文档中心查看详细介绍。在本文档中，因为后续需要用户在数据市场购买数据，因此，用户需要在“交换区”建立项目。

图 3.4.1 创建项目流程 1

点击“新建项目”按钮，按需填写新建项目所需的信息。其中，“项目名称”在项目创建

后可修改；“项目标识”是代表项目空间的唯一标识，一旦填写并提交将无法修改。

图 3.4.2 创建项目流程 2

3.5 添加项目成员

项目创建后，就需要添加必要的组织内成员到项目中干活啦！

只有已被邀请到组织里的成员，才可以被添加到项目中。且本次添加动作不需要该成员进行同意审批。

3.5.1 项目首页

可以从两个入口进入“项目”。

1、在“数据中心-----工作台”首页点击项目名，即可进入项目

图 3.5.1 工作台首页

2、点击“数据引擎”，进入项目所在“交换区”，可以看到项目 list，点击后进入项目

图 3.5.2 项目列表

进入项目后界面如下：

图 3.5.3 项目首页

3.5.2 进入“成员管理”

点击左侧“项目管理”，进入“成员管理”

图 3.5.4 成员管理

3.5.3 邀请组织中成员加入本项目

进入“成员”tab，点击“邀请组织中成员加入本项目”，输入成员的账号进行查询后，便可以添加成员。

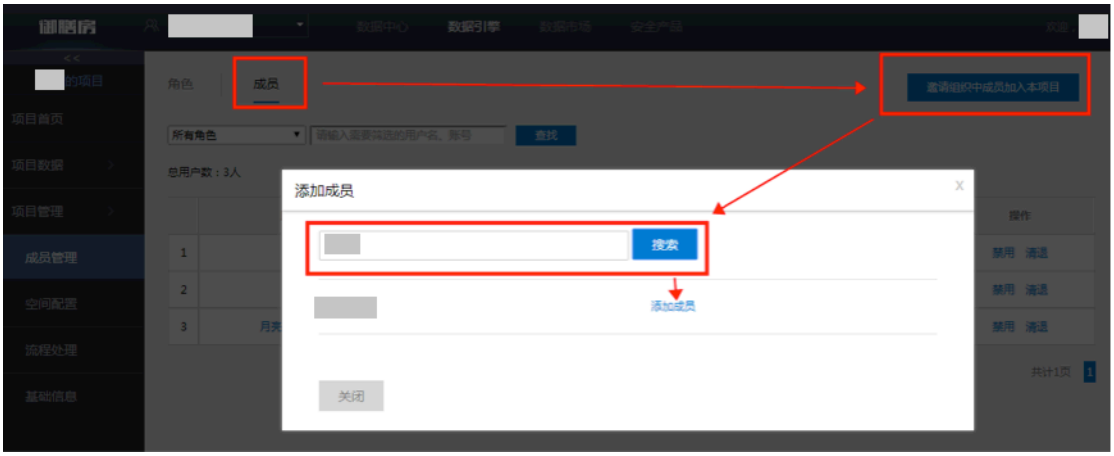


图 3.5.5 添加项目成员

3.5.4 赋予成员权限

进入“角色”tab，按需授予成员权限

图 3.5.6 项目成员权限

每个角色的权限可以点击“功能权限”进行查看。

四、数据上传

本部分演示了如何将其他数据库的数据上传到御膳房，适用于用户在日常开发及工作中的应用场景。

4.1 准备数据源

4.1.1 进入“项目”

可以从两个入口进入“项目”

1、在“数据中心-----工作台”首页点击项目名，即可进入项目

图 4.1.1 工作台首页

2、点击“数据引擎”，进入“交换区”，可以看到项目 list，点击后进入项目

图 4.1.2 项目列表

进入项目后界面如下：

图 4.1.3 项目首页

4.1.2 选择“项目数据---数据源”

点击项目空间左侧的项目数据----数据源

图 4.1.4 项目数据源

4.1.3 添加“数据源”

1、点击“+新增”

图 4.1.5 新增数据源

2、配置数据源

目前支持从阿里云 RDS、阿里云 ADS、阿里云 OSS、SFTP Server、自建 RDBMS 同步数据到御膳房。下面以 SFTP 的方式举例说明数据上传。

按照实际情况填写数据源配置：

图 4.1.6 新增数据源配置

3、配置完成后，可以看到新增的数据源

图 4.1.7 项目数据源列表

4.1.4 TIPS

1、数据源的配置信息填写有哪些要求？

数据源配置信息参考如下：

- RDS：阿里云 RDS
  - 数据库类型：选择 Mysql
  - 实例名称：RDS 给出的实例名称
  - 数据库名称：需要抽取的数据库名称
  - 用户名：对应数据库的用户名
  - 密码：对应用户名密码
- ADS：阿里云 ADS
  - IP：IP 地址
  - Port：端口号
  - 数据库名称：需要抽取的数据库名称
- OSS：阿里云 OSS
  - EndPoint: OSS Server 的 EndPoint 地址
  - accessId: OSS 的 accessId
  - acessKey: OSS 的 accessKey
  - Bucket: OSS 的 Bucket
- SFTP：自建 SFTP
  - IP：FTP 的公网 IP 地址
  - Port：端口号

- 用户名
- 密码

- 自建 RDBMS：用户自建 MySQL/SQLserver / Oracle 关系型数据库
  - 数据库类型：MySQL/SQLserver / Oracle
  - Host：主机地址
  - Port：端口号
  - 数据库名：需要抽取的数据库名称
  - Agent 组：绑定 Agent 组
  - 用户名
  - 密码
- ODPS：阿里云 ODPS
  - EndPoint：ODPS 的 EndPoint
  - tunnelEndPointP：ODPS 的 tunnelEndPoint
  - project：project 名称
  - accessId：ODPS 对应的 accessId
  - accessKey：accessId 对应的 accessKey
- OT：OpenTargeting
  - submitCallbackUrl：
  - stateCallbackUrl:
  - OT accessId：
  - OT accessKey：

2、目前支持哪些数据上传、导出方式？

御膳房提供将数据从非御膳房环境（用户自建数据库、阿里云等不同网络环境）与御膳房数据存储间的高速数据传输能力。目前支持的能力包括但不限于：

- RDS 上传导出
- ADS 导出
- OSS 上传导出
- OCS 导出
- TOP 接口上传（使用 TOP 上传数据到 UMP，然后导入）
- UMP 导出
- SFTP 上传导出

- 自建 RDBMS 上传导出
- OT 导出

4.2 建目标表

4.2.1 进入数据开发环境（以下称 IDE）

进入项目空间后，点击“数据开发”即可进入 IDE

图 4.2.1 建立目标表流程 1

4.2.2 进入 IDE 后，在左侧选择“表管理”

图 4.2.2 建立目标表流程 2

4.2.3 点击“+”，选择“新建表”

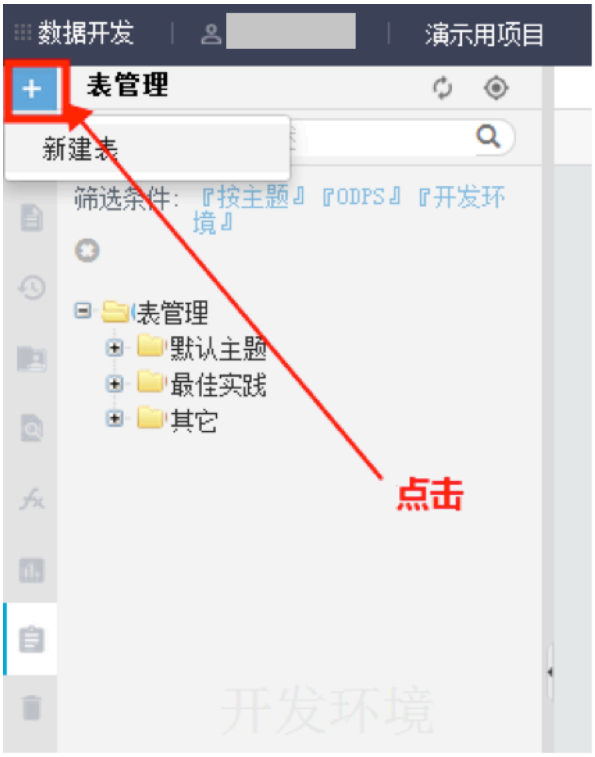


图 4.2.3 建立目标表流程 3

4.2.4 数据库类型选择“ODPS”

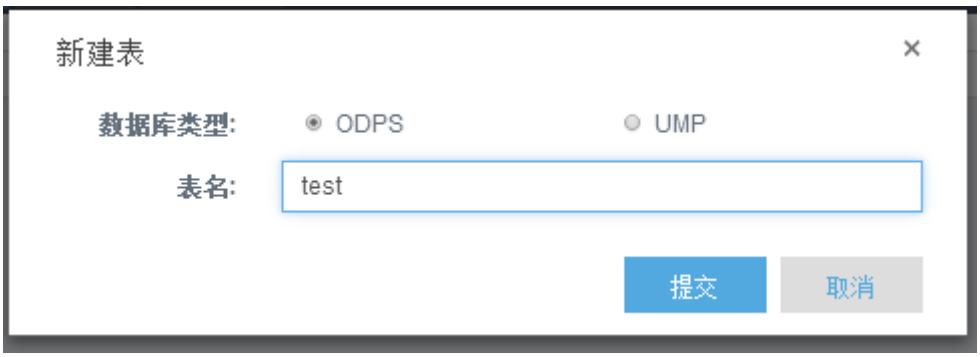


图 4.2.4 建立目标表流程 4

4.2.5 逐项填写表配置

目标表的表结构请与数据源的表结构保持一样，表类型请选择“上传表”

图 4.2.5 建立目标表流程 5

4.2.6 提交到开发环境

填写完建表配置后，将表提交到开发环境，就可以在表管理中看到新建的表了。

图 4.2.6 建立目标表流程 6

4.2.7 TIPS

- **什么是分区表和非分区表**

在御膳房中，所有的数据都被存储在表中。表中的列可以是 ODPS 支持的任意种数据类型（Bigint，Double，String，Boolean，Datetime）。御膳房中的各种不同类型计算任务的操作对象（输入、输出）都是表。用户可以创建表、删除表以及向表中导入数据。

为了提高处理效率，可以在创建表时指定表的分区（Partition），即指定表内的某几个字段作为分区列。在大多数情况下，用户可以将分区类比为文件系统下的目录。御膳房将分区列的每个值作为一个分区（目录）。用户可以指定多级分区，即将表的多个字段作为表的分区，分区之间正如多级目录的关系。

指定分区表会对用户带来诸多便利，例如：提高 SQL 运行效率，减少计算成本等。在

如下场景下使用分区表将会带来较大的收益：在 select 语句的 where 条件过滤中使用分区列作为过滤条件。然而，部分对分区操作的 SQL 的运行效率则较低，给您带来较高的计算成本，例如使用动态分区。

4.3 数据同步

下面，我们开始配置数据同步任务

4.3.1 进入 IDE

图 4.3.1 数据同步流程 1

4.3.2 在 IDE 环境左侧选择“数据开发”

图 4.3.2 数据同步流程 2

4.3.3 点击“+”，选择“工作流节点”

图 4.3.3 数据同步流程 3

4.3.4 新建“数据同步”节点

图 4.3.4 数据同步流程 4

4.3.5 配置数据同步任务，示例如下：

图 4.3.5 数据同步流程 5

- 选择同步方式
  - 跨平台导入：从外部导入数据到御膳房数据源
  - 平台内同步：御膳房数据源之间数据同步
  - 跨平台导出：从御膳房数据源导出数据
- 配置源表&目标表

  根据不同数据源类型，我们提供尽量简洁的配置项供填写和选择。

  后续按照数据源类型进行详细说明。
- 配置字段映射关系

  当选择源&目标均为：御膳房 ODPS, UMP, RDS 中一项的话，会显示字段配置选项。其他情况下 ,无法进行字段配置。需要使用者保证源&目标 ,字段数量/字段类型 一致，否则同步过程中会运行出错。

4.3.6 配置调度

调度配置完成后点击“提交”即可。配置的示例如下：

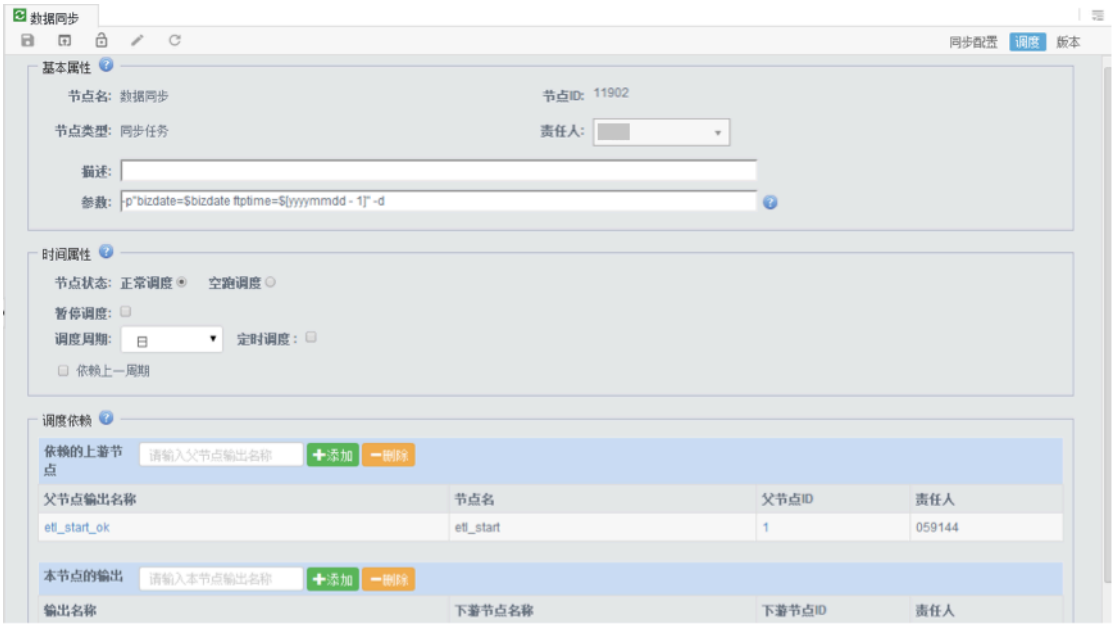


图 4.3.6 数据同步流程 6

需要填写的各项描述如下：

- 节点名称：新建工作流节点的时候填写的节点名，需要修改可以在目录树，鼠标对节点右键”重命名”。
- 节点 id：任务提交后会生成唯一的节点 id，不可修改。
- 节点类型：新建工作流节点的时候选择的节点类型，不可修改。
- 责任人：即节点 owner,新建起来的节点责任人默认是当前登录工作台的用户，需要修

改可以点击输入框，输入搜索或者直接选择其他用户（都是当前应用的成员）。

- 描述：一般用于描述节点业务、用途。
- 参数：任务调度时给代码里的变量赋值。使用场景:代码里用变量表示时间，如pt=${datetime}，在这里可以给代码里的变量赋值，赋的值可以用调度内置时间参数datetime=$bizdate（调度内置了一些常用的时间参数，具体参数说明请看参数说明）。
- 调度周期：按需选择，支持月、周、日、小时、分钟级的周期。
- 上游节点：即调度任务依赖的上游任务，没有上游节点的任务可以填写“etl_start_ok”

4.3.7 发布任务

1、点击“发布管理”

图 4.3.7 数据同步流程 7

2、点击“打包发布”

图 4.3.8 数据同步流程 8

注：

- 有“开发”角色才可以创建发布包；
- 创建发布包 tab 下，任务列表里，已经加入发布包未发布成功的节点，不显示在这列表，且不能再编辑提交，必须要把发布包执行成功或者从那个发布包里删除，才能编辑提交并显示在创建发布包 tab 的任务列表里。
- 创建发布包 tab 下，任务列表里，开发环境测试信息只是提示作用，不会阻塞任务发布。

3、任务发布成功后，我们即完成了数据上传的工作。

五、数据商品购买

本文档第六部分的数据开发过程中使用的数据表，已被御膳房店铺上架在数据市场以供购买使用。用户无需再进行数据同步，可直接通过在数据市场购买数据商品并授权给“交换区”项目，进行本文档后续的数据开发示范操作。因此这部分将简单介绍数据购买的流程，详细的数据市场操作方法可以在御膳房文档中心 - 数据市场进行学习和查阅。

御膳房有严格的权限规则，有权限在数据市场购买数据商品的组织角色包括：组织所有者、数据中心管理员、采购者。因此用户首先需要确认组织角色，然后再进行购买。

5.1 购买数据商品

1、在御膳房工作台点击“数据市场”，在搜索栏输入要购买的表名，本文档使用表名为：买家类目交互表。点击表名可进入商品详情页。

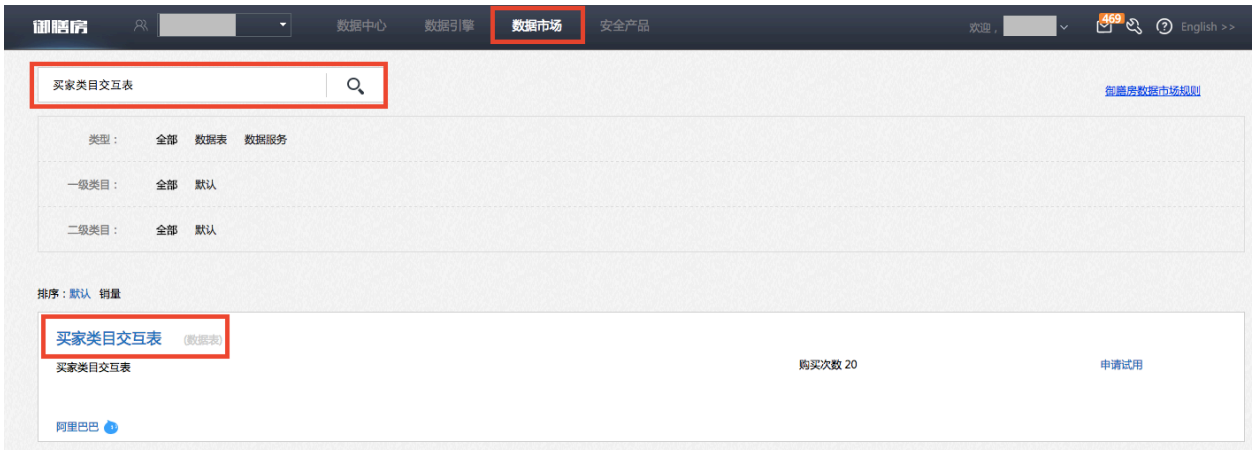


图 5.1.1 购买数据流程 1

2、在商品详情页选择周期，并点击申请试用。

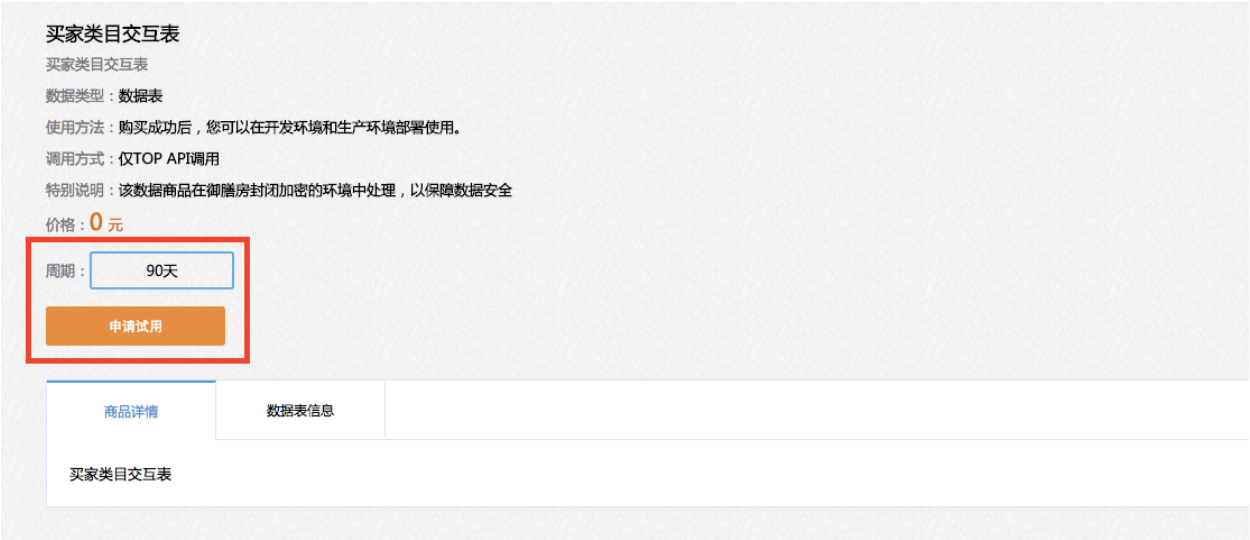


图 5.1.2 购买数据流程 2

3、在申请页面填写申请理由并选择目标组织，最后点击申请。我们会及时处理订单，订单通过审核后，御膳房以系统信息方式提醒用户。

图 5.1.3 购买数据流程 3

5.2 数据授权

5.2.1 授权给项目

御膳房有严格的权限规则，有权限进行**数据授权给项目**的组织角色包括：组织所有者、数据中心管理员、组织管理员。因此用户首先需要确认组织角色，然后再将数据授权给项目。想了解详细的御膳房权限管理，请进入文档中心 - 权限服务。

注 御膳房设定项目必须绑定 Appkey 才能被组织授权使用数据市场购买的数据商品。

请用户在项目首页 - 项目管理 - 项目配置中绑定 Appkey，再进行后续操作。

1、进入数据中心 - 数据管理 - 交换区数据 - 购买的数据，授权方式选择“两方授权”，找到“买家类目交互表”，点击授权给项目。

图 5.2.1 授权数据给项目流程 1

2、在授权弹框中选择指定的交换区项目，进行授权

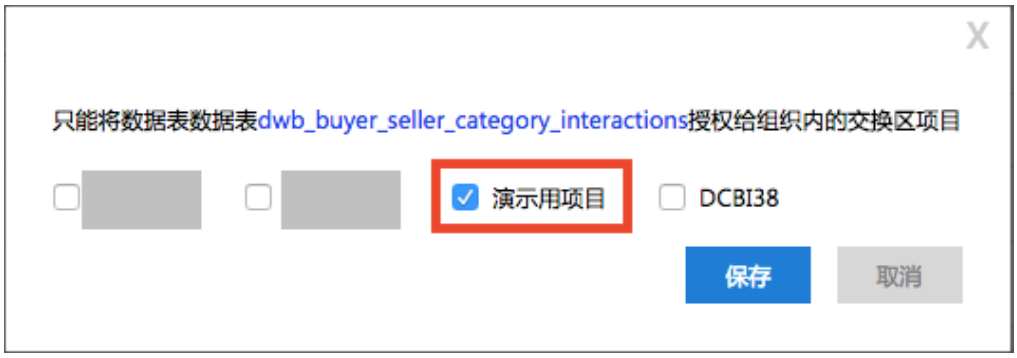


图 5.2.2 授权数据给项目流程 2

5.2.2 项目内部授权

现在，需要将项目的数据授权给项目内相关人员，用于后续的数据开发工作。有权限进行**项目内部**授权的项目角色包括：项目所有者、项目管理员。因此用户首先需要确认项目角色，然后再将数据进行内部授权。

注 1：项目内只有项目管理员、开发和运维才有数据权限，要使用购买的表，请用户先将数据开发人员添加这三个角色中的至少一个。

1、进入数据引擎 - 交换区

图 5.2.3 项目内数据授权流程 1

2、进入项目首页

图 5.2.4 项目内数据授权流程 2

3、选择“项目数据---数据授权”，点击右上角“历史授权”

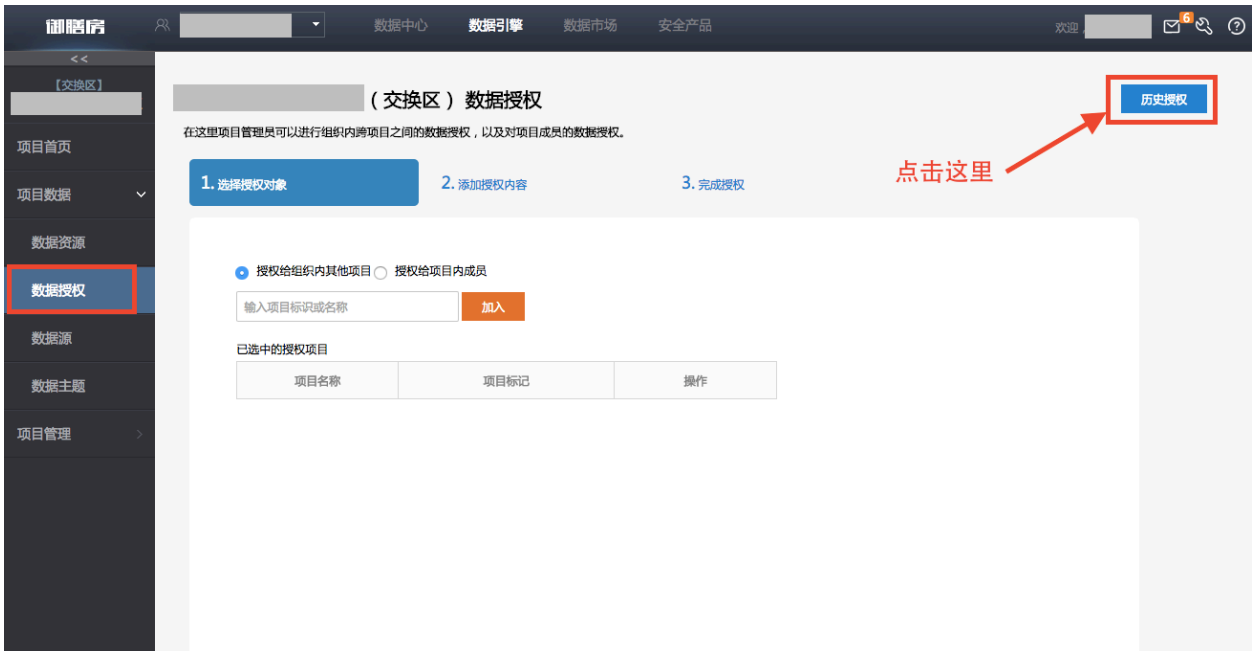


图 5.2.5 项目内数据授权流程 3

4、点击“申请得到的”，来源选择“来自数据市场”，找到购买的表名，点击下方“内部授权”

图 5.2.6 项目内数据授权流程 4

5、在内部授权界面，添加授权对象（角色或账号）。

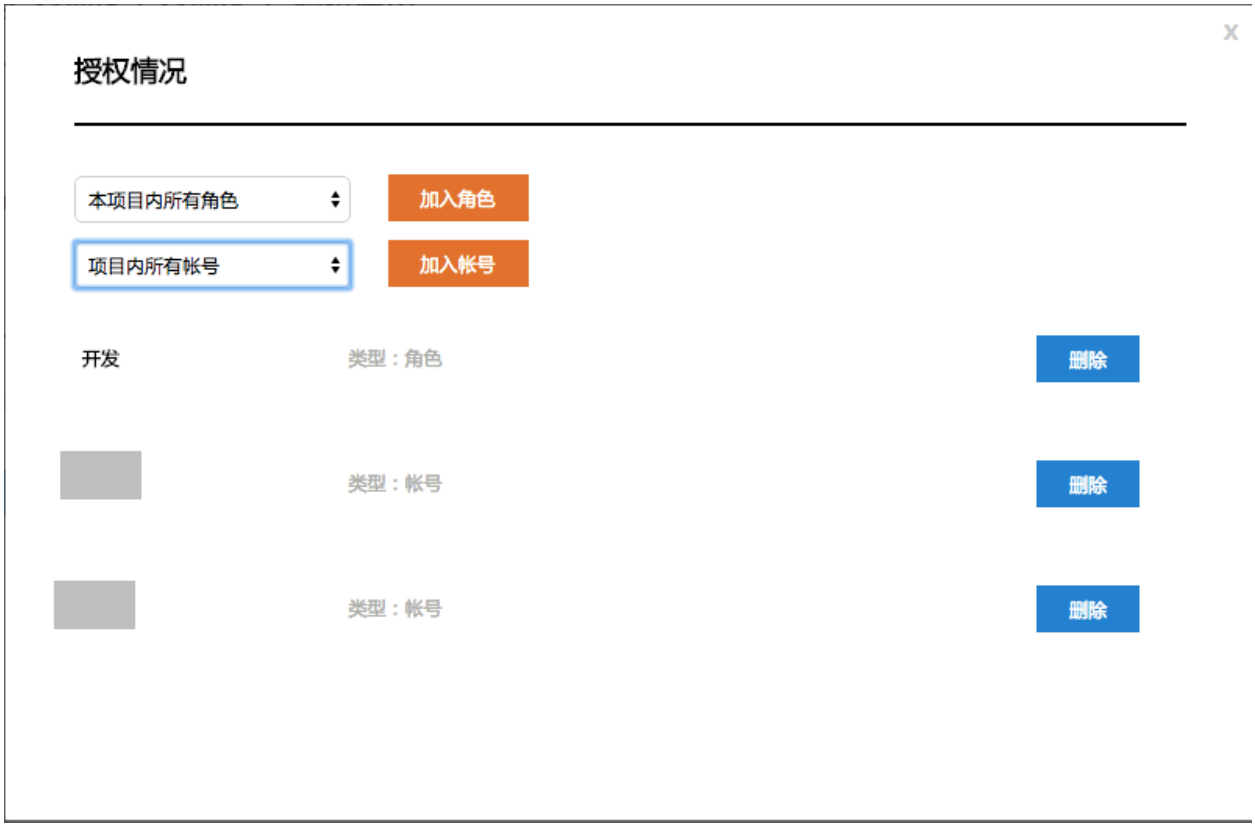


图 5.2.7 项目内数据授权流程 5

5.3 查看数据

授权完成后，被授权成员可以在 IDE 环境中以只读方式使用这些资源。

5.3.1 进入 IDE

进入项目空间后，点击“数据开发”即可进入 IDE 环境

图 5.3.1 创建临时查询流程 1

5.3.2 创建临时查询任务

1、点击左侧的临时查询后，点击“+”新建临时查询，示例如下：

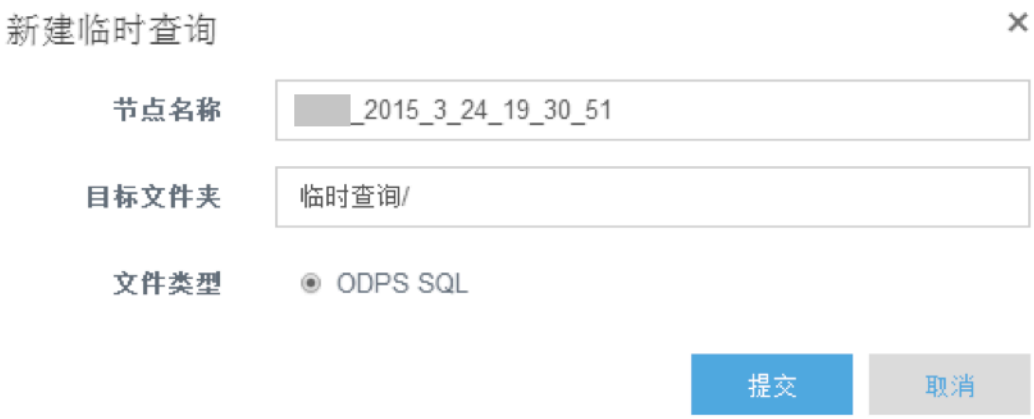


图 5.3.2 创建临时查询流程 2

2、提交后即可在 IDE 里写 SQL 语句查看数据

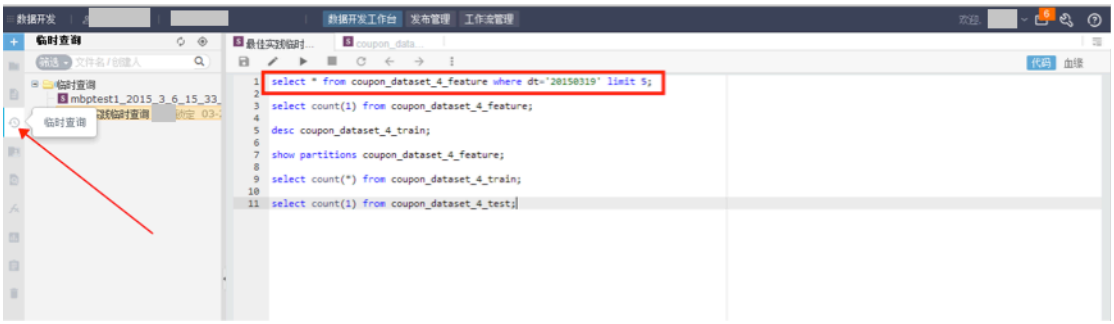


图 5.3.3 创建临时查询流程 3

六、数据开发

下面，我们将会以优惠券预测模型的开发流程，来演示御膳房的使用过程。数据开发的全过程在“演示用户 01”组织中的“演示用项目”项目中进行。

注：在御膳房中，只有“交换区”能够使用市场购买来的数据商品，因此用户如要按照本文档示范进行操作，请在“交换区”建立项目，并在该项目下进行数据开发及测试。

6.1 建表

首先，我们需要先建三张表，用于后续的开发中。这三张表分别是：

表名	用途
coupon_dataset_4_feature	原始特征集表
coupon_dataset_4_train	训练集表
coupon_dataset_4_test	测试集表

表 6.1 需要建立三张表的信息

下面以创建表 coupon_dataset_4_feature 为例，介绍如何在御膳房中建表。

6.1.1 进入项目空间

在“数据中心-工作台”首页点击项目空间名，即可进入项目空间。

图 6.1.1 建表流程 1

也可以先进入“数据引擎”，然后在项目列表中，点击对应的项目名称进入项目空间。

图 6.1.2 建表流程 1

6.1.2 进入 IDE

进入项目空间后，点击“数据开发”即可进入数据开发环境（以下称 IDE）。

图 6.1.3 建表流程 2

6.1.3 进入 IDE 后，在左侧导航栏中选择“表管理”。

图 6.1.4 建表流程 3

6.1.4 点击“+”图标，选择“新建表”。

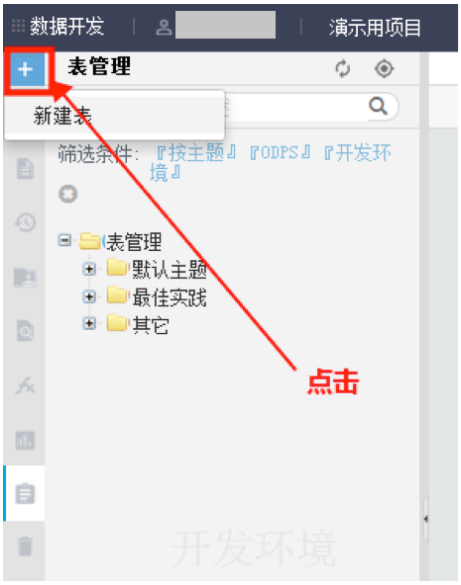


图 6.1.5 建表流程 4

6.1.5 数据库类型选择“ODPS”，表名填写为“coupon_dataset_4_feature”。

图 6.1.6 建表流程 5

6.1.6 逐项填写表配置，基本属性、物理模型设计如下所示。

图 6.1.7 建表流程 6

表结构设计的内容如下所示。

图 6.1.8 表结构设计

表结构数据，也可以使用 DDL 语句快速输入。点击顶部工具条的“DDL 模式”，在弹出的对话框中输入以下 DDL 语句，点击“生成表结构”即可。

```
CREATE TABLE coupon_dataset_4_feature (
  masked_buyer_id STRING COMMENT '模糊用户 id',
  masked_seller_id STRING COMMENT '模糊卖家 id',
  masked_shop_id STRING COMMENT '模糊店铺 id',
  cat_id BIGINT COMMENT '类目 id',
  pv DOUBLE COMMENT '流量',
  add_cart_num DOUBLE COMMENT '添加购物车次数',
  auction_collect_num DOUBLE COMMENT '商品收藏数',
  alipay_trade_num DOUBLE COMMENT '购买次数',
  target BIGINT COMMENT '模型预测目标'
)
COMMENT '最佳实践特征模型表'
PARTITIONED BY (
  dt STRING
)
LIFECYCLE 10;
```

填写完表配置后，将表提交到开发环境，就可以在表管理中看到新建的表了。

图 6.1.9 建表流程 7

表 coupon_dataset_4_train 和 coupon_dataset_4_test 的建表步骤类似，表信息如下所示。

1）coupon_dataset_4_train

表的基本属性和物理模型设计信息如下。

图 6.1.10 表基本属性和物理模型设计

表结构设计如下：

表结构设计

添加字段 | 删除字段 | 上移 | 下移

字段英文名	中文名	字段类型	描述	主键	非空	操作
masked_buyer_id		STRING	模糊用户id		✓	
masked_seller_id		STRING	模糊卖家id		✓	
masked_shop_id		STRING	模糊店铺id		✓	
cat_id		BIGINT	类目id		✓	
pv		DOUBLE	流量		✓	
add_cart_num		DOUBLE	加车数		✓	
auction_collect_num		DOUBLE	收藏数		✓	
alipay_trade_num		DOUBLE	alipay数		✓	
target		BIGINT	目标		✓	

图 6.1.11 表结构设计

对应的 DDL 语句如下：

```
CREATE TABLE coupon_dataset_4_train (
 masked_buyer_id STRING COMMENT '模糊用户 id',
 masked_seller_id STRING COMMENT '模糊卖家 id',
 masked_shop_id STRING COMMENT '模糊店铺 id',
 cat_id BIGINT COMMENT '类目 id',
 pv DOUBLE COMMENT '流量',
 add_cart_num DOUBLE COMMENT '加车数',
 auction_collect_num DOUBLE COMMENT '收藏数',
 alipay_trade_num DOUBLE COMMENT 'alipay 数',
 target BIGINT COMMENT '目标'
)
COMMENT '用于训练模型的数据'
;
```

2）coupon_dataset_4_test

表的基本属性和物理模型设计信息如下。

图 6.1.12 表基本属性和物理模型设计

表结构设计如下：

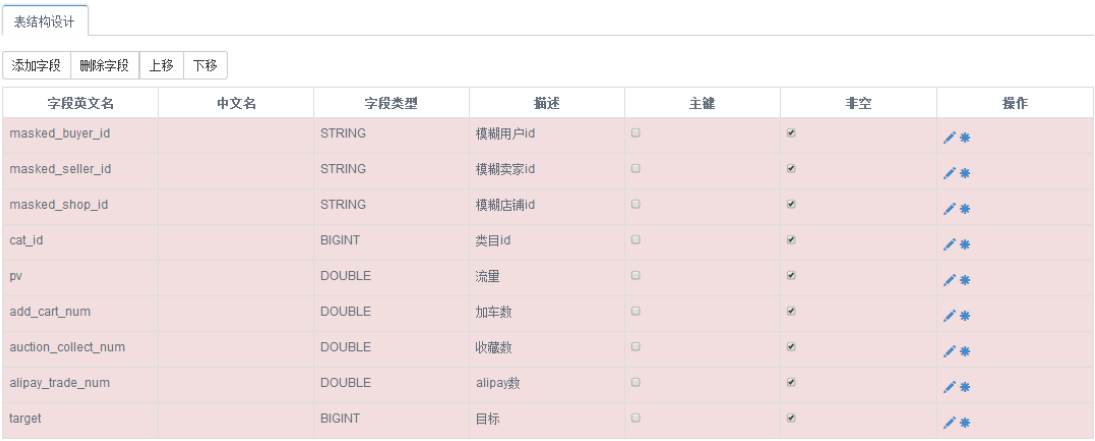


图 6.1.13 表结构设计

对应的 DDL 语句如下：

```
CREATE TABLE coupon_dataset_4_test (
 masked_buyer_id STRING COMMENT '模糊用户 id',
 masked_seller_id STRING COMMENT '模糊卖家 id',
 masked_shop_id STRING COMMENT '模糊店铺 id',
 cat_id BIGINT COMMENT '类目 id',
 pv DOUBLE COMMENT '流量',
 add_cart_num DOUBLE COMMENT '加车数',
 auction_collect_num DOUBLE COMMENT '收藏数',
 alipay_trade_num DOUBLE COMMENT 'alipay 数',
 target BIGINT COMMENT '目标'
)
;
```

6.1.7 TIPS

- **什么是生产和开发环境**

类似于网站开发里的日常开发环境和线上生产环境，隔离日常开发工作和线上稳定运行的业务。

在数据引擎里，数据、调度都分开发、生产两个环境。在开发环境里，每个用户可以自由做数据开发、调试，对开发环境的数据表有读写权限，但对生产环境的数据表数据只有读权限，开发环境下无法修改生产数据。御膳房上系统默认的开发、生产环境分别对应两个 ODPS Project，例如生产 Project 名称为 ysf，那么开发环境 Project 名称则为 ysf_dev。

开发环境的所有变更只有通过发布管理中发布上线，才会真正作用于生产环境。简单来讲，例如 SQL 代码改动后，只有完成发布，在生产环境里每天自动调度运行时，才会跑

改动后的代码。

通过提供这两个环境，为用户确保灵活自由的数据开发、挖掘和稳定支撑线上业务的数据处理可以安全隔离。

- **什么是节点**

节点是指对数据进行处理的基本单元（任务）。御膳房工作流节点的主要包括：ODPS SQL、ODPS MR、数据同步、算法实验等。

- **数据开发中的节点、临时查询、手动任务有什么区别**

- 节点：可以自定义调度时间
- 临时查询：不需要发布到生产环境，只是做一个临时用的查询
- 手动任务：需要发布到生产环境，但需要手动发布

- **可不可以直接 DDL 建表**

可以在 IDE 环境中新建 ODPS SQL 节点后，写 DDL 语句建表，等 1-2 分钟即可在表管理中查看到新建的表。

- **可视化建表和手写 DDL 有什么区别**

- 可视化建表界面，允许输入更多属性，例如主键。
- 可视化建表界面，最终系统也会生成 DDL。

- **怎么查看数据**

查看数据的具体办法请参阅“数据上传—数据查看”。

6.2 数据探查

我们可以数据探索环境里进行数据探查。

6.2.1 数据探索环境

御膳房数据探索环境是基于远程虚拟桌面技术，构建在 Ubuntu 操作系统之上的一站式数据探索环境。数据探索环境用于数据预处理、数据观察、特征处理、数据建模、模型

评测等研究性质的工作。御膳房数据探索环境与御膳房数据仓库是天然连通的，可以直接访问与同步用户在御膳房中已经获得授权的系统表以及个人表到数据探索环境。

6.2.2 申请开通

御膳房数据探索环境目前还是邀请制，我们欢迎有志于大数据分析与挖掘的用户申请试用，具体请联系御膳房的运营小二（可发邮件至 yushanfang@service.taobao.com 申请，也可以通过加入御膳房用户反馈群[1]申请）。具体流程如下：

组织的管理者向御膳房运营小二提交业务需求进行申请，通过申请后，提交需要使用数据探索环境的 “组织标识”与需要开通数据探索环境的“项目空间” ;确认通过邀请后在项目管理者的项目空间配置的虚拟机管理中即可发现虚拟机的 IP 地址与端口号。一个 IP 与一个端口组成一个数据探索环境实例，即一个远程桌面。

图 6.2.1 虚拟机管理

对成员进行虚拟机绑定后，我们可以从项目首页的数据探索处进入数据探索环境。

图 6.2.2 进入数据探索

[1]御膳房 1 群-用户反馈-技术支持 1：759773056。御膳房 2 群-用户反馈-技术支持 2: 1454609023。

6.2.3 同步数据

1、进入“数据探索”的远程虚拟桌面页面。

图 6.2.3 数据探索 - 数据同步流程 1

2、点击“数据同步”，填入项目信息登录。

图 6.2.4 数据探索 - 数据同步流程 2

登录使用到了用户标识、项目 ID、项目证书三个信息，这些信息的获取地址如下：

首先，点击自己登录名弹出下拉菜单，点击“我的账号”。

图 6.2.5 数据探索 - 数据同步流程 3

然后，点击“证书管理”，我们就可以看到项目 ID、用户标识、项目证书三个信息。由于用户标示和项目证书字符较长，数据探索页面提供“剪切板”方便用户复制粘贴操作，具体操作参考 Tips 中详细介绍。

图 6.2.6 数据探索 - 数据同步流程 4

3、双击需要同步的表，即可打开表的详细信息页面。在这里可以查看表列信息、分区信息，可以预览前 1000 条记录。点击“表下载”，设置数据文件存储路径，选择下载的分区、起始位置和下载的记录数，点击“同步数据”即可同步 ODPS 数据表到远程桌面的本地磁盘。数据同步之后，就可以使用合适的工具进行探查和分析。

注：在下载页面，防止数据量过大造成延迟，用户可将数据条数改为 200000 条。分区选择 dt=20150625

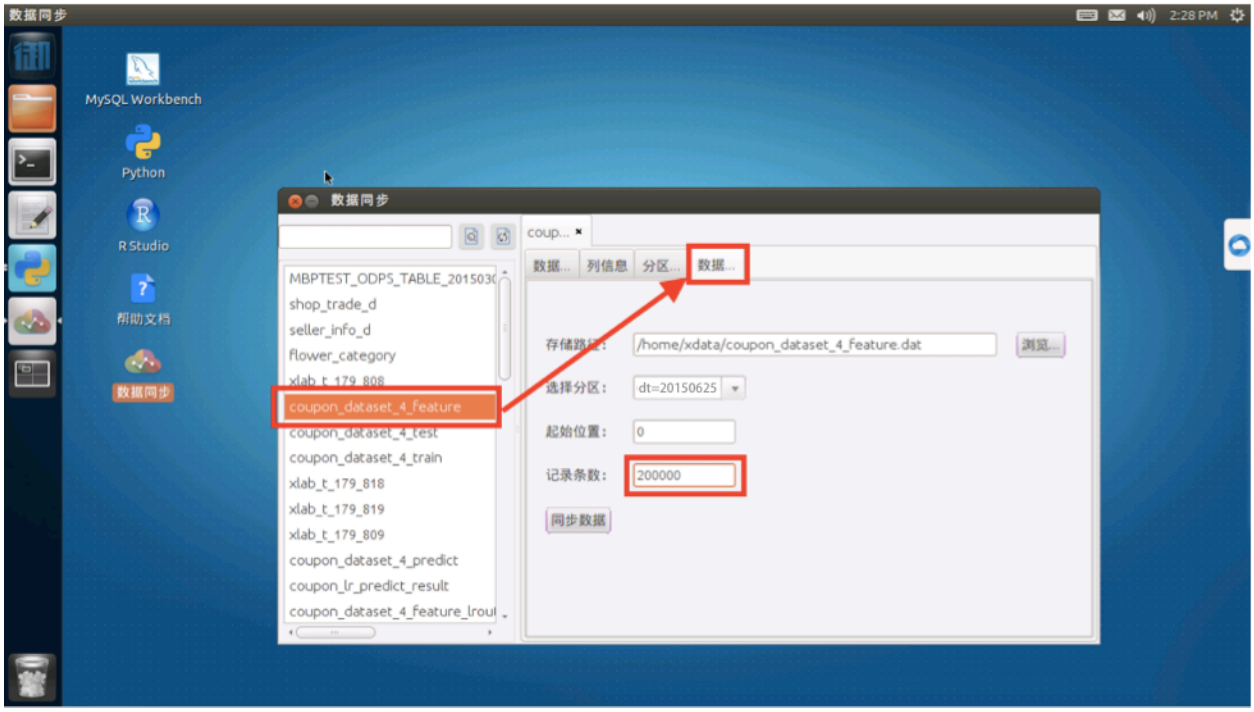


图 6.2.7 数据探索 - 数据同步流程 4

6.2.4 数据探查

1、同步后，我们可以用 Python 中的 pandas 载入数据。

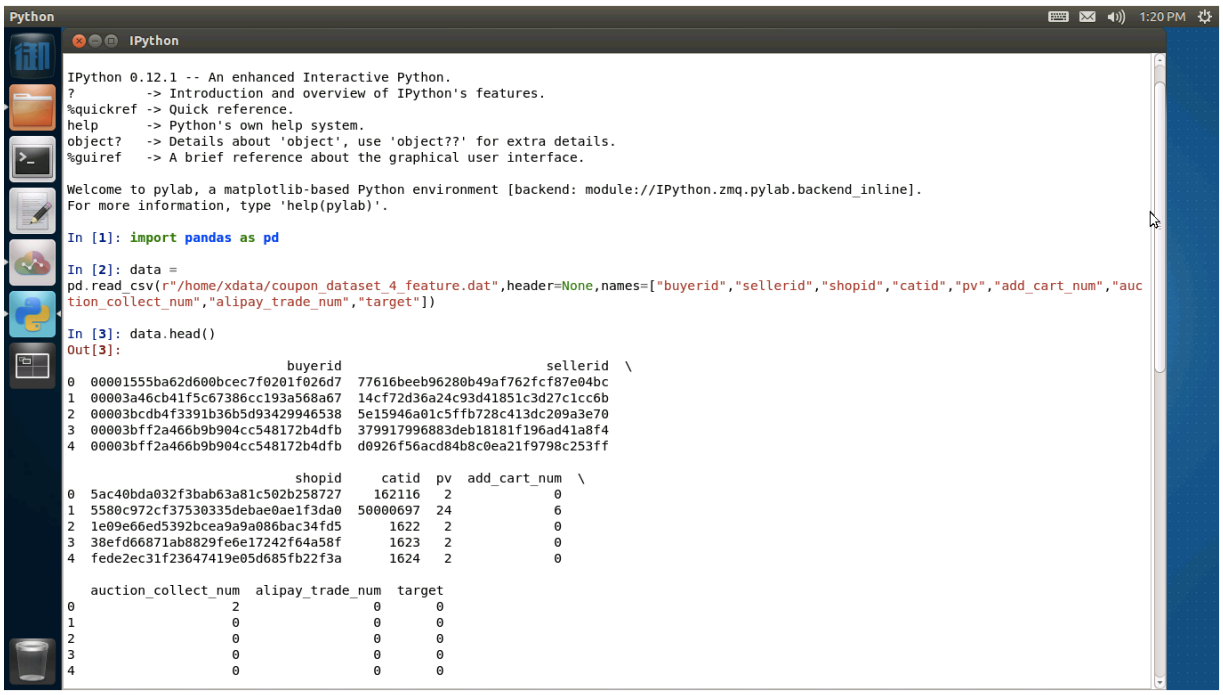


图 6.2.8 数据探查示例 1

2、使用 python 简单计算各个列均值。

```
In [6]: data.mean()
Out[6]:
catid                  30455612.844745
pv                            3.510970
add_cart_num                  0.154440
auction_collect_num           0.154980
alipay_trade_num              0.055290
target                        0.001305
dtype: float64
```

图 6.2.9 数据探查示例 2

当然你也可以使用 R Studio 进行数据探查和分析，在此不再演示。

6.2.5 TIPS

- **什么是剪切板**

目前只支持外部上传文件到数据探索环境，数据探索环境的一切数据及文件不可以导出。剪贴板提供了外部往远程桌面上复制文本的功能。点击页面右侧的御膳房小图标，即可弹出小工具页面。

图 6.2.10 剪切版使用流程 1

点击后示例如下：

图 6.2.11 剪切版使用流程 2

6.3 特征工程

下面，我们会对原始数据进行特征工程。

6.3.1 新建文件夹

为了方便管理代码，可以先新建一个文件夹，然后把这个案例的所有代码都放在一个文件夹中。

1. 在 IDE 环境左侧选择“数据开发”。

图 6.3.1 新建文件夹流程 1

2. 点击“+”，选择“文件夹”。

图 6.3.2 新建文件夹流程 2

3. 输入文件夹名称和路径，点击“提交”创建一个文件夹。

图 6.3.3 新建文件夹流程 3

我们可以用这种方式，建立“工作流/最佳实践/算法实践”的文件夹，在后续的操作中，所有相关的代码和任务都会放到该文件夹中。

6.3.2 新建 ODPS SQL 工作流节点

1. 在 IDE 环境左侧选择“数据开发”。

图 6.3.4 新建工作流节点流程 1

2. 点击“+”图标，选择“工作流节点”。

图 6.3.5 新建工作流节点流程 2

3. 节点配置如下。

图 6.3.6 新建工作流节点流程 3

节点类型：ODPS SQL

节点名称：coupon_dataset_4_feature

目标文件夹：工作流/最佳实践/算法实践

4. 配置完毕后，点击提交，进入下一步。

6.3.3 编写代码

提交新建的工作流节点后，我们即可在 IDE 环境中编写代码。

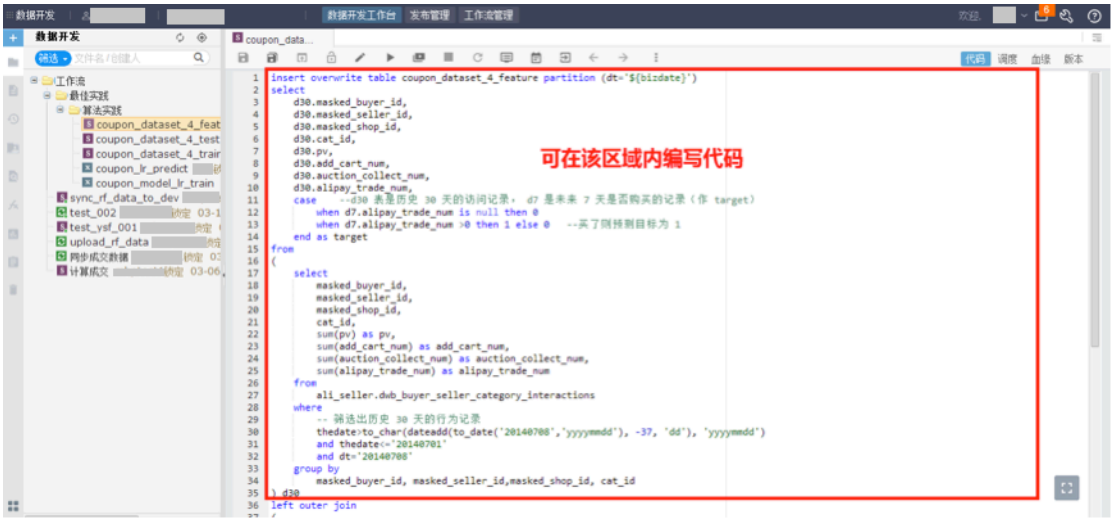


图 6.3.7 IDE 代码编写示例

代码示例如下：

```
   insert overwrite table coupon_dataset_4_feature partition
(dt='${bizdate}')
   select
       d30.masked_buyer_id,
       d30.masked_seller_id,
       d30.masked_shop_id,
       d30.cat_id,
       d30.pv,
       d30.add_cart_num,
       d30.auction_collect_num,
       d30.alipay_trade_num,
       case    --d30 表是历史 30 天的访问记录， d7 是未来 7 天是否购买的记录
(作 target)
           when d7.alipay_trade_num is null then 0
           when d7.alipay_trade_num >0 then 1 else 0   --买了则预测目标
为 1
       end as target
   from
   (
       select
           masked_buyer_id,
           masked_seller_id,
           masked_shop_id,
           cat_id,
           sum(pv) as pv,
           sum(add_cart_num) as add_cart_num,
           sum(auction_collect_num) as auction_collect_num,
           sum(alipay_trade_num) as alipay_trade_num
```

```
      from
         ali_seller_dwb_buyer_seller_category_interactions
      where
         -- 筛选出历史 30 天的行为记录
         thedate>to_char(dateadd(to_date('20150625','yyyymmdd'),
-37, 'dd'), 'yyyymmdd')
         and thedate<='20150618'
         and dt='20150625'
      group by
         masked_buyer_id, masked_seller_id,masked_shop_id, cat_id
   ) d30
   left outer join
   (
      -- 未来 7 天的行为记录
      select
         masked_buyer_id,
         masked_seller_id,
         masked_shop_id,
         cat_id,
         sum(pv) as pv,
         sum(add_cart_num) as add_cart_num,
         sum(auction_collect_num) as auction_collect_num,
         sum(alipay_trade_num) as alipay_trade_num
      from
         ali_seller_dwb_buyer_seller_category_interactions
      where
         thedate>to_char(dateadd(to_date('20150625','yyyymmdd'), -7,
'dd'), 'yyyymmdd')
         and thedate<='20150625' and dt='20150625'
      group by
         masked_buyer_id, masked_seller_id,masked_shop_id, cat_id
   ) d7
   on
      d30.masked_buyer_id = d7.masked_buyer_id and
      d30.masked_seller_id = d7.masked_seller_id and
      d30.masked_shop_id = d7.masked_shop_id and
      d30.cat_id = d7.cat_id;
```

6.3.4 运行代码

1、代码编写完成后，点击高级运行。

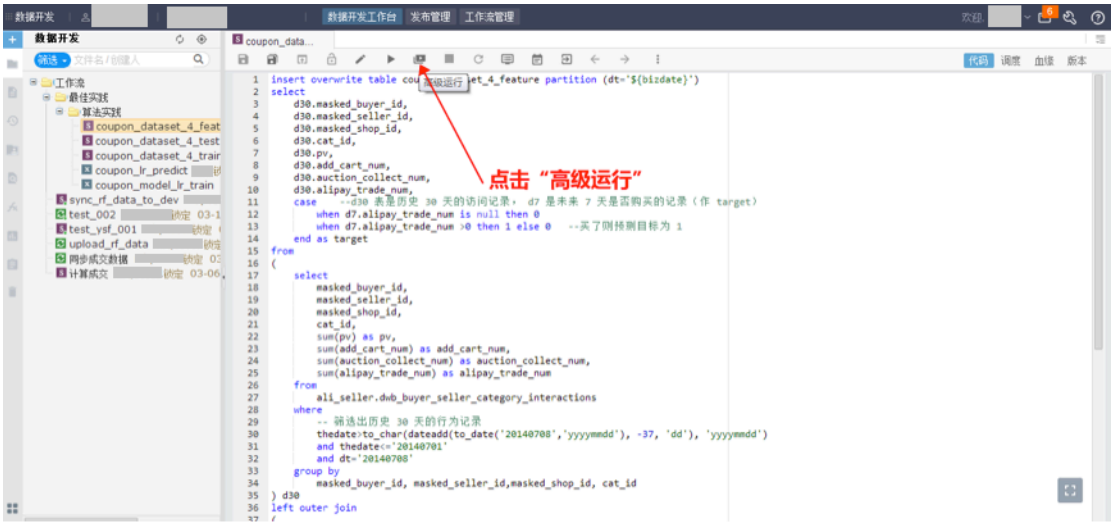


图 6.3.8 运行代码流程 1

2、选择分区（2015/06/25），运行后，对应的数据会写到 coupon_dataset_4_feature 表的对应分区。

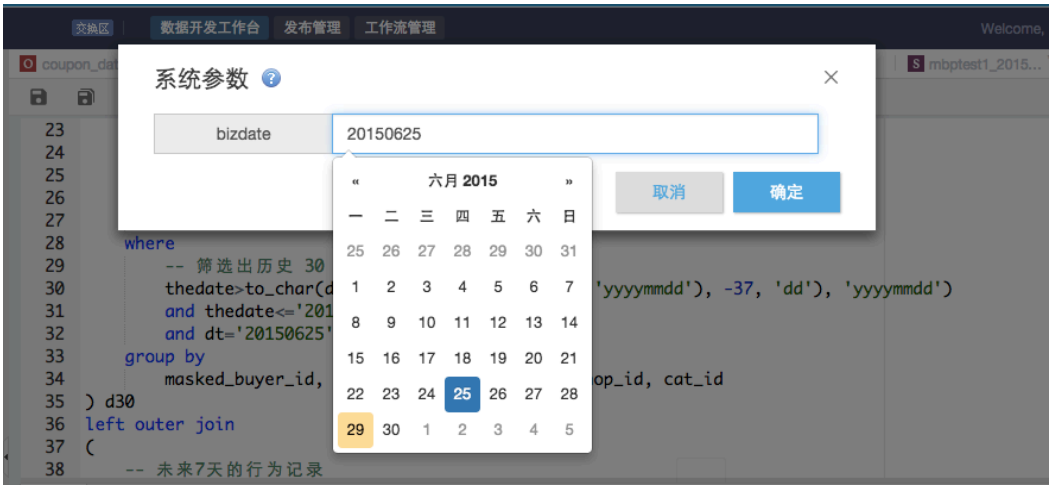


图 6.3.9 运行代码流程 2

3、可以通过临时查询简单查看数据。

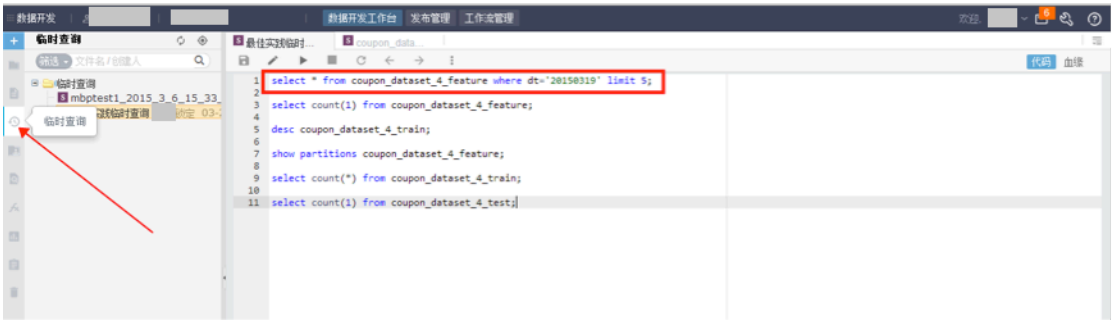


图 6.3.10 临时查询

代码如下：

select * from coupon_dataset_4_feature where dt='xxx' limit 5;

其中，xxx 对应选定的分区。

6.4 数据拆分

下面，我们会将原始特征表（coupon_dataset_4_feature）的数据按照 8:2 的比例随机分配到训练集表和测试集表中，这两个表没有分区。

1、新建 ODPS SQL 工作流节点，名为 coupon_dataset_4_train，用来存储训练集表。

图 6.4.1 数据拆分流程 1

2、编写代码，拆分出训练集表。

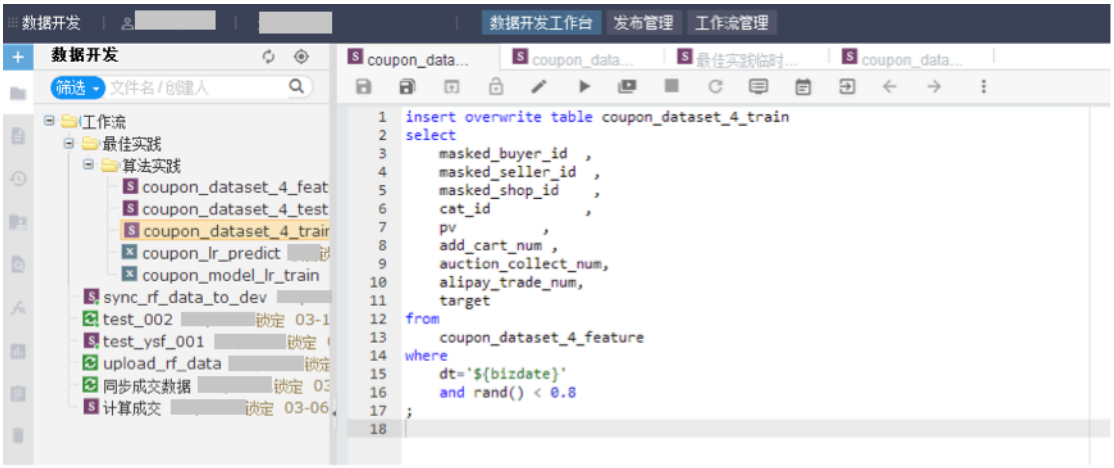


图 6.4.2 数据拆分流程 2

代码示例如下：

```
insert overwrite table coupon_dataset_4_train
select
```

```
    masked_buyer_id  ,
    masked_seller_id  ,
    masked_shop_id    ,
    cat_id          ,
    pv          ,
    add_cart_num ,
    auction_collect_num,
    alipay_trade_num,
    target
from
    coupon_dataset_4_feature
where
    dt='${bizdate}'
    and rand() < 0.8
;
```

3、点击运行后，数据即写入训练集表中；

4、创建 ODPS SQL 工作流节点，名为 coupon_dataset_4_test，用来存储测试集表。

图 6.4.3 数据拆分流程 3

5、编写代码，拆分出测试集表。

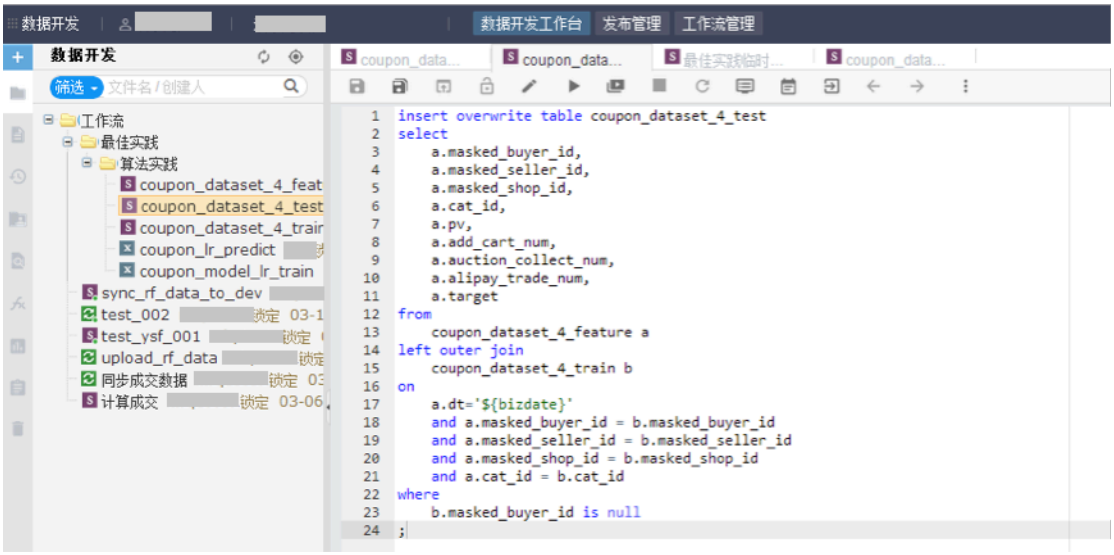


图 6.4.4 数据拆分流程 4

代码示例如下：

```
insert overwrite table coupon_dataset_4_test
select
    a.masked_buyer_id,
    a.masked_seller_id,
    a.masked_shop_id,
    a.cat_id,
    a.pv,
    a.add_cart_num,
    a.auction_collect_num,
    a.alipay_trade_num,
    a.target
from
    coupon_dataset_4_feature a
left outer join
    coupon_dataset_4_train b
on
    a.dt='${bizdate}'
    and a.masked_buyer_id = b.masked_buyer_id
    and a.masked_seller_id = b.masked_seller_id
    and a.masked_shop_id = b.masked_shop_id
    and a.cat_id = b.cat_id
where
    b.masked_buyer_id is null
;
```

6、点击运行后，数据即写入测试集表中；

6.5 算法平台建模

完成数据拆分后，利用算法平台进行模型建立。御膳房数据算法平台目前还是邀请制，我们欢迎有志于大数据分析与挖掘的用户申请试用，具体请联系御膳房的运营小二（可发邮件至 yushanfang@service.taobao.com 申请，也可以通过加入御膳房用户反馈群[1]）。用户可以进入文档中心 - 算法平台详细查阅使用方法。

6.5.1 实验模型建立及训练：

pv（流量）、add_cart_num（添加购物车次数）、auction_collect_num（商品收藏）、alipay_trade_num（支付数）与 target（预测目标）之间的回归模型。

1、新建 ODPS SQL 工作流节点，选择“算法实验”，名为 coupon_model_lr_train。

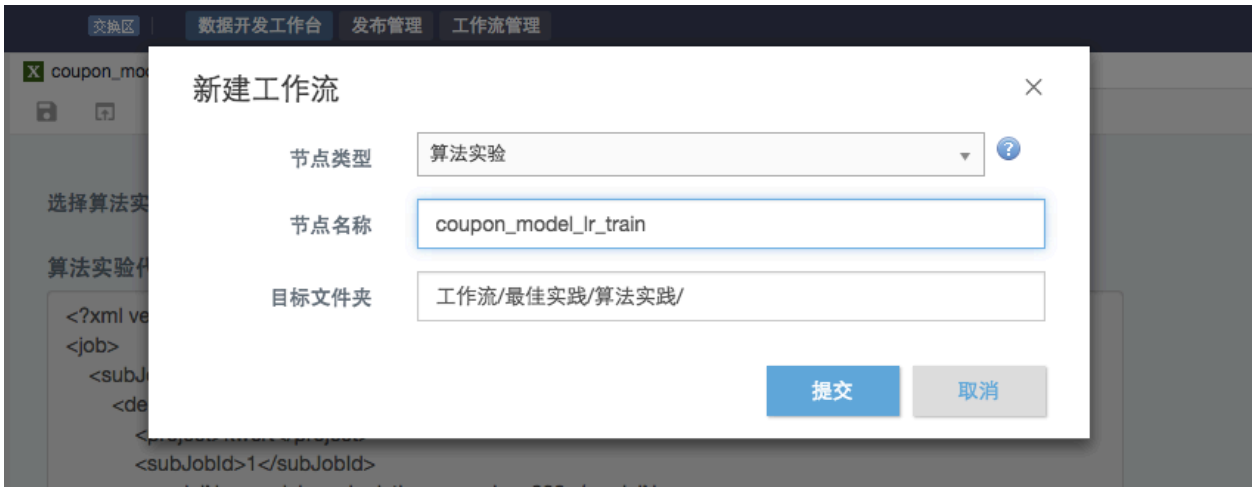


图 6.5.1 算法平台实验模型建立流程 1

2、在新节点的界面中点击“在算法平台中查看实验”

[1]御膳房 1 群-用户反馈-技术支持 1：759773056。御膳房 2 群-用户反馈-技术支持 2: 1454609023。

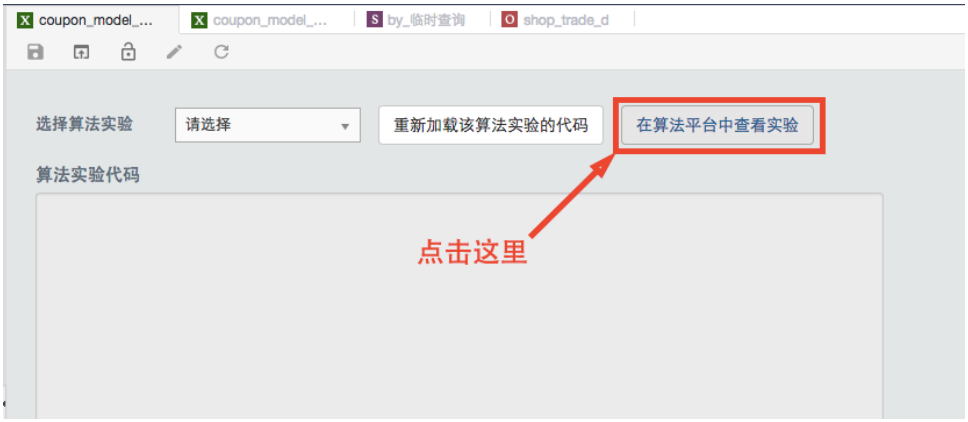


图 6.5.2 算法平台实验模型建立流程 2

3、进入算法实验的界面，新建算法实验，“coupon_lr_exp”

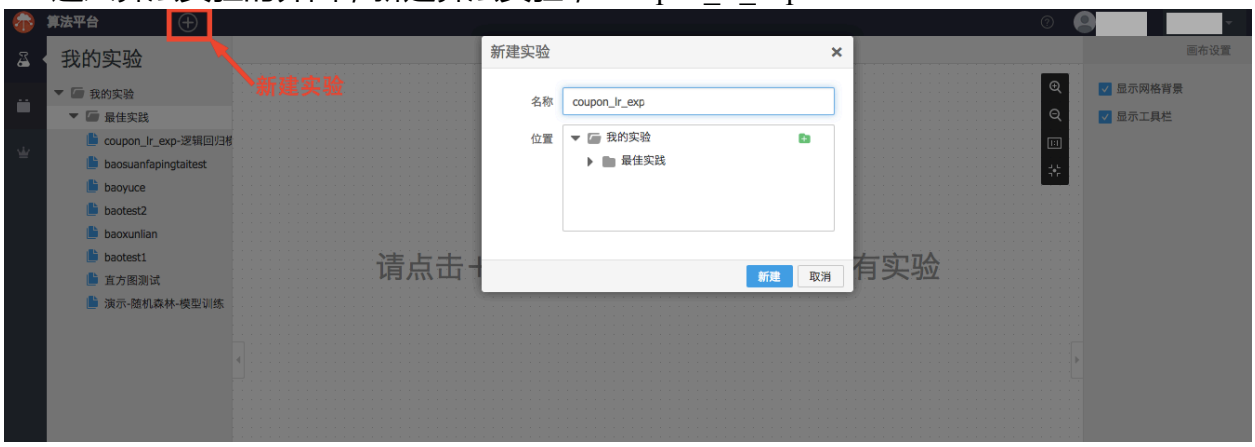


图 6.5.3 算法平台实验模型建立流程 3

4、然后选择模型组件，在组件栏中分别拖拽“ODPS 源”和“逻辑回归”至右侧开发面板中。

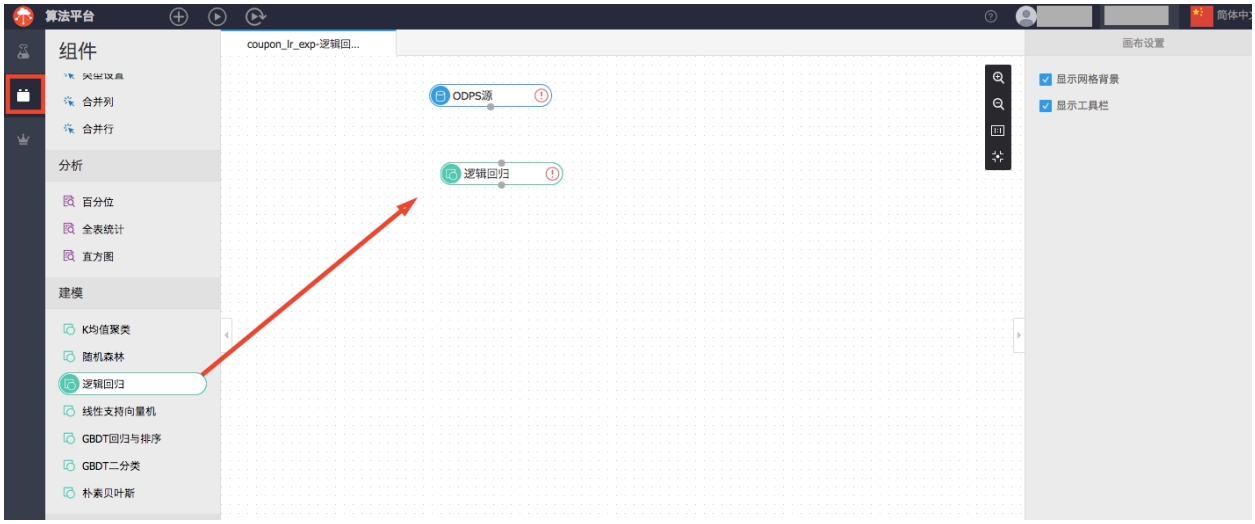


图 6.5.4 算法平台实验模型建立流程 4

5、双击“ODPS 源”，在右边栏内进行组件参数配置。

- 表名：coupon_dataset_4_train

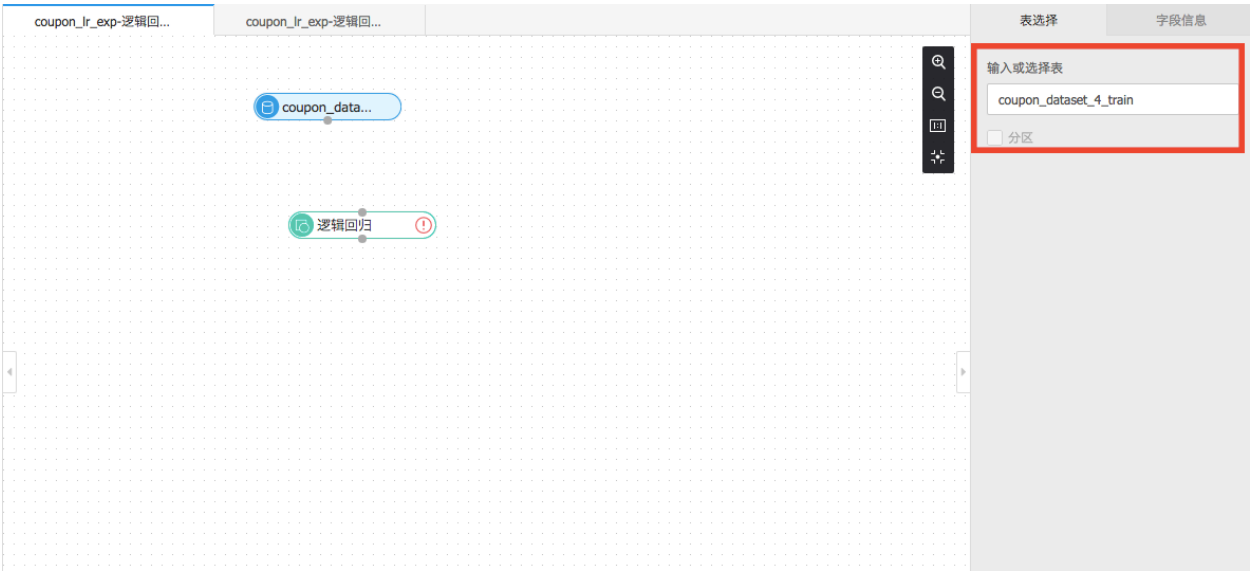


图 6.5.5 算法平台实验模型建立流程 5

6、将“ODPS 源”和“逻辑回归”相连接然后双击“逻辑回归”进行组件参数配置。

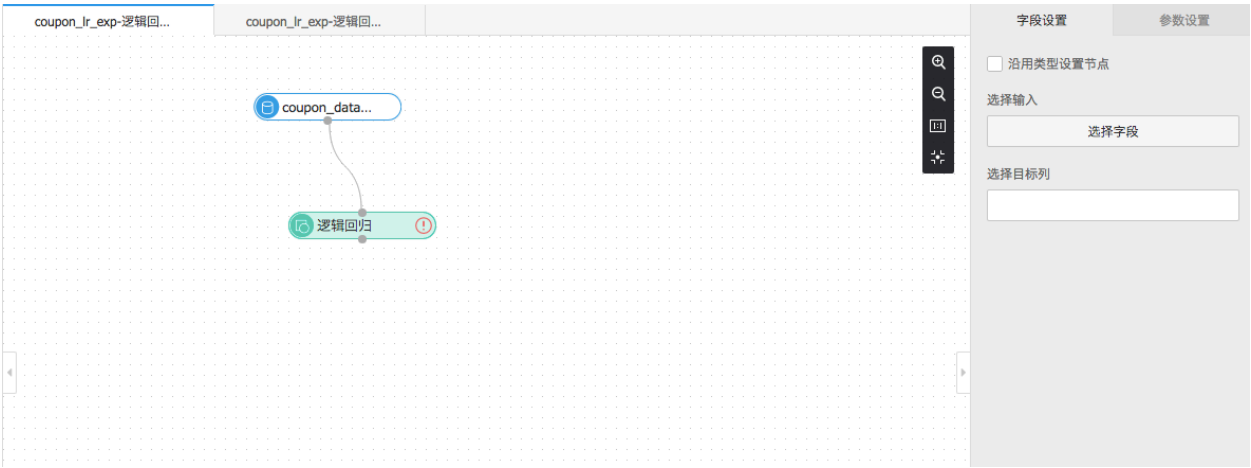


图 6.5.6 算法平台实验模型建立流程 6

字段设置：

- 选择输入：pv、add_cart_num、auction_collect_num、alipay_trade_num
- 选择目标列：target

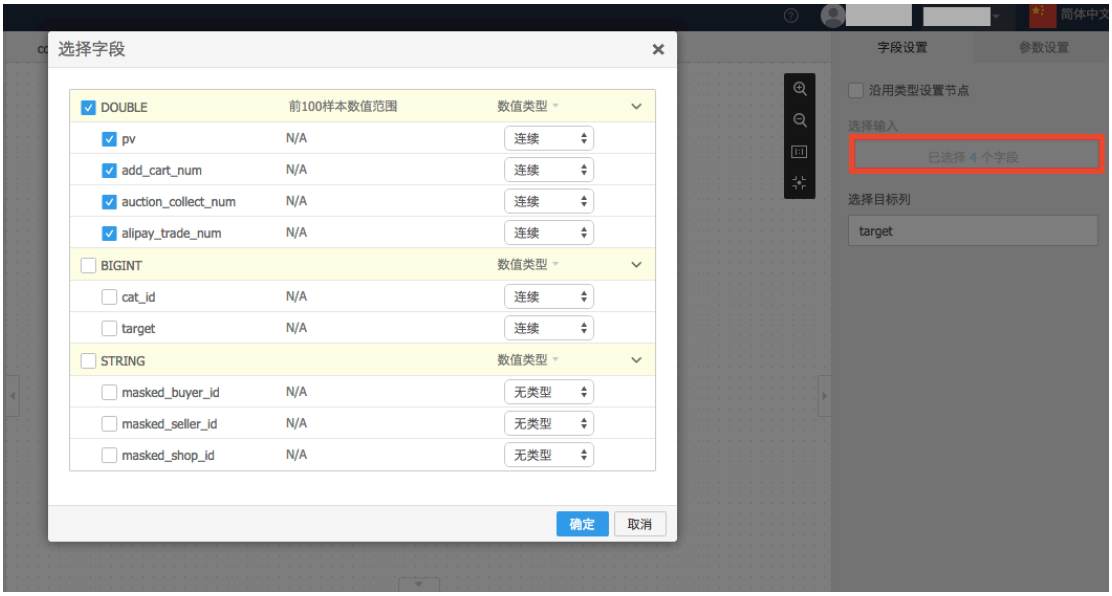


图 6.5.7 算法平台实验模型建立流程 7

6.5.2 模型测试及评估

1、新建算法实验，命名为：coupon_lr_evaluation。并在模型栏中找到上一步训练好的模型（coupon_lr_exp-LR-target 模型），拖拽到开发面板。

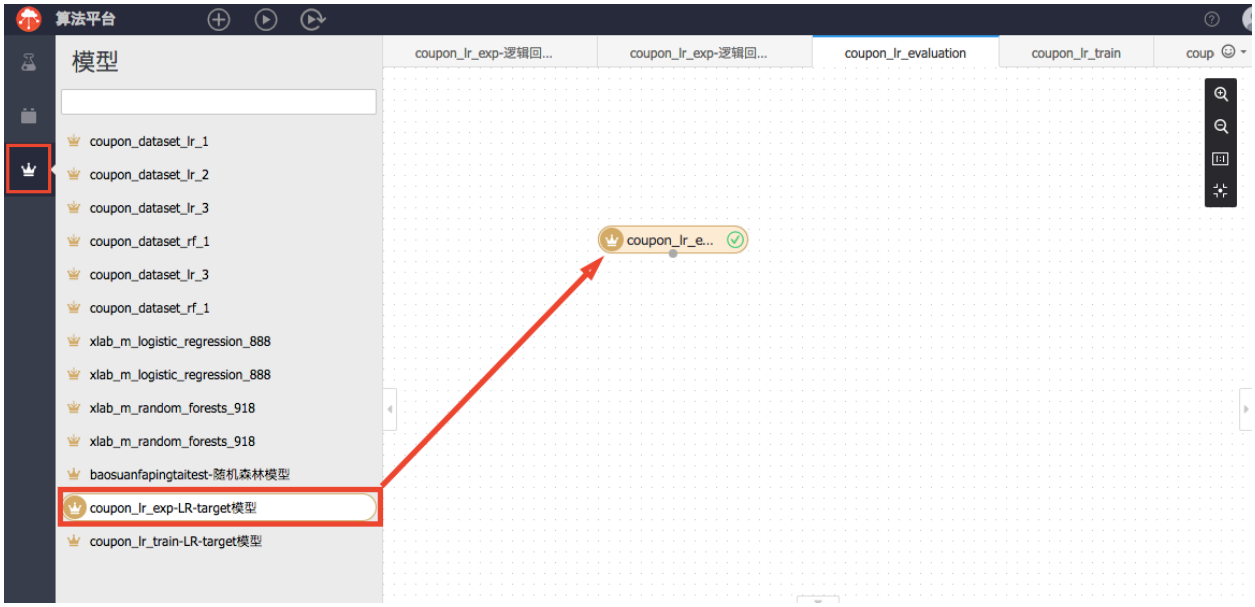


图 6.5.8 模型测试及评估流程 1

2、在组件栏中拖拽“ODPS 源”至右侧开发面板中，并双击进行配置。

- 输入或选择表：coupon_dataset_4_test

图 6.5.9 模型测试及评估流程 2

3、预测 :在组件栏中将“预测”拖拽至开发面板 ,将“模型”与“ODPS 源”分别与“预测”连接，并双击“预测”进行参数设置。

- 自定义
- 二分类
- 目标基准值：1

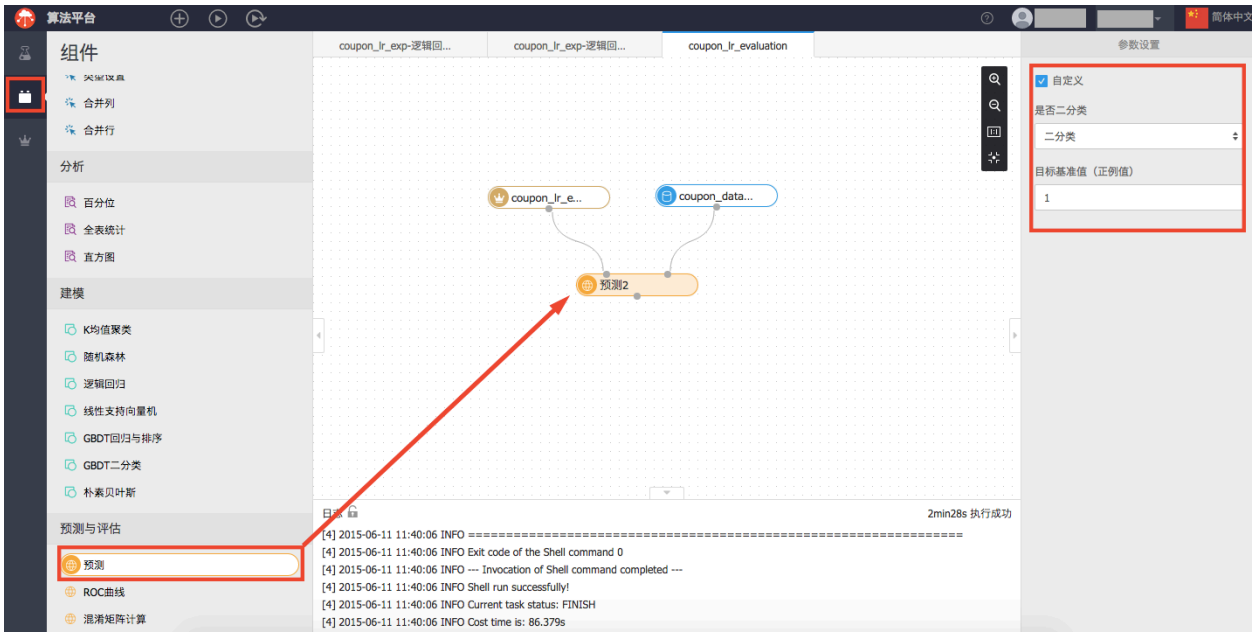


图 6.5.10 模型测试及评估流程 3

4、模型评估：在组件中将“ROC 曲线”拖拽至开发面板，双击进行参数设置。

- 沿用设置

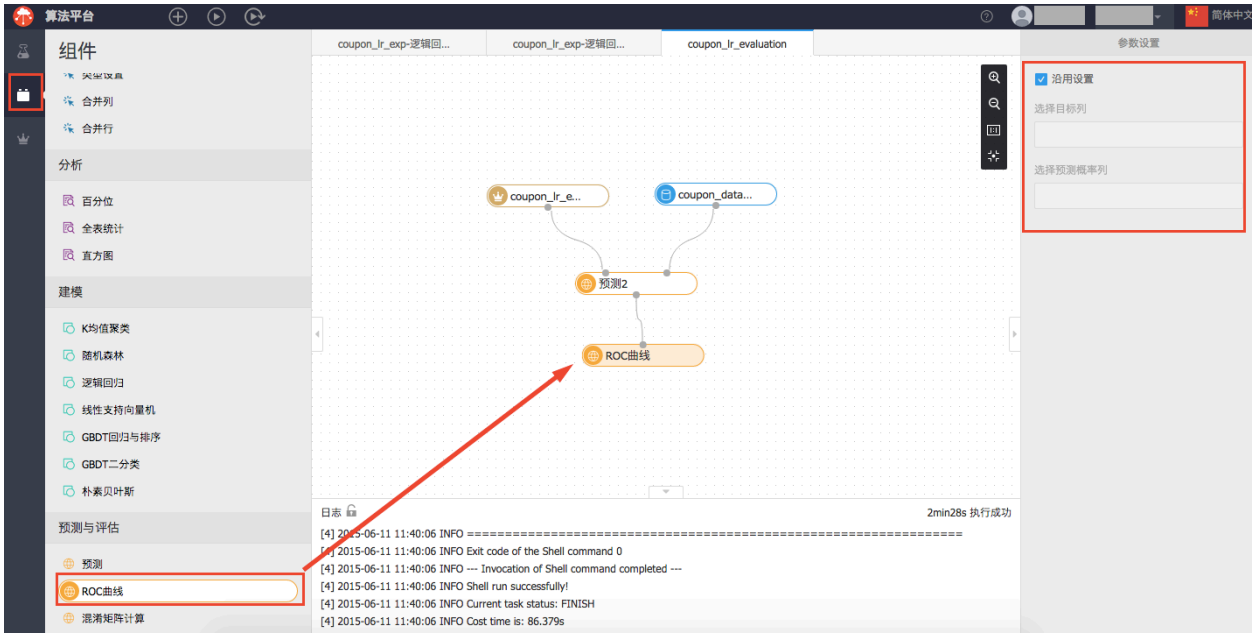


图 6.5.11 模型测试及评估流程 4

5、点击运行，然后右键点击“ROC 曲线”查看评估报告

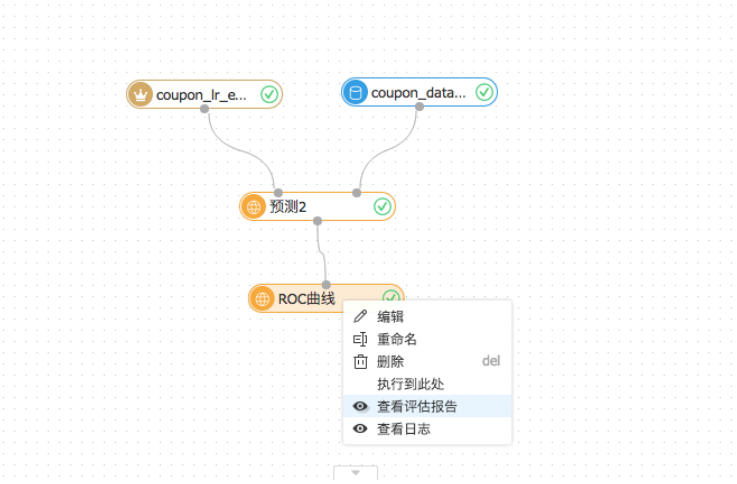


图 6.5.12 模型测试及评估报告 1

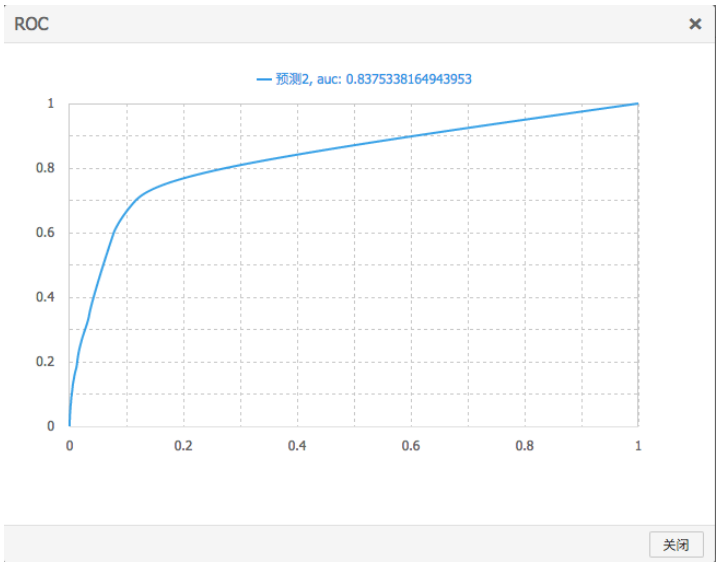


图 6.5.13 模型测试及评估报告 2

6、右键点击“预测”，查看预测数据

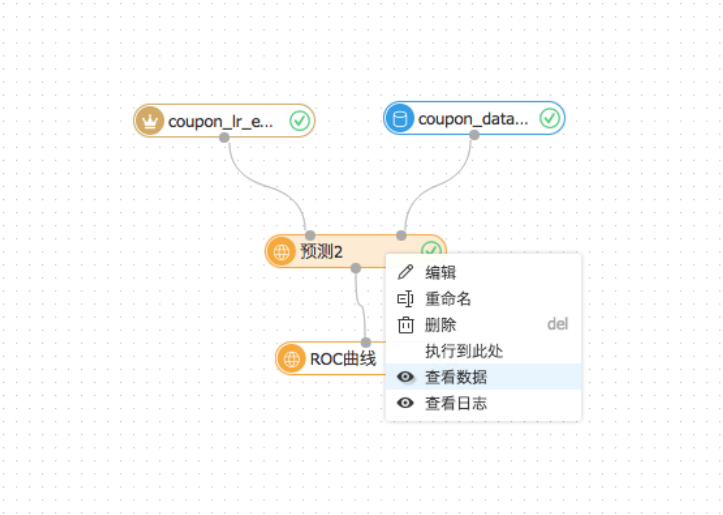


图 6.5.14 查看预测数据 1

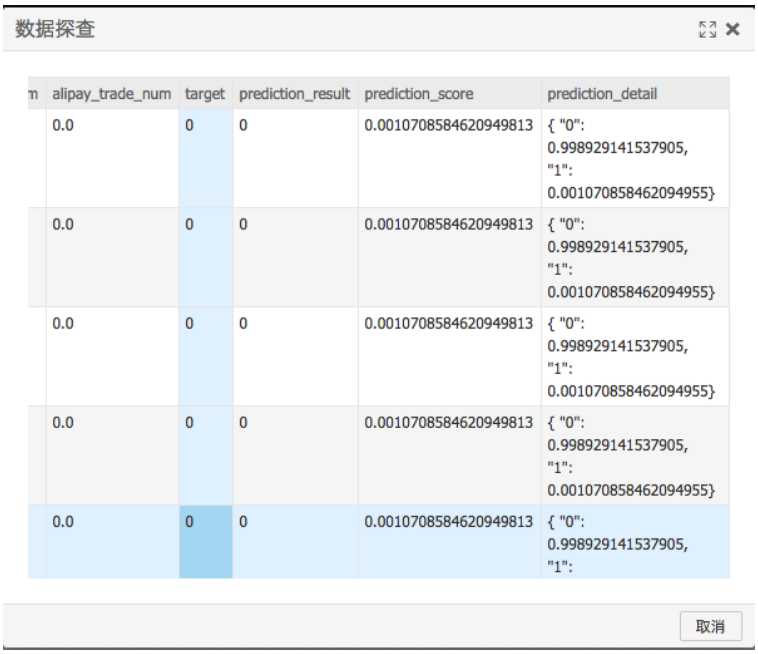


图 6.5.15 查看预测数据 2

6.5.3 上线模型的建立和训练

1、新建算法实验，命名：coupon_lr_train。用于使用原始特征集训练，该实验是最后上线训练模型的实验。

ODPS 源配置

- 输入或选择表：coupon_dataset_4_feature
- 勾选分区，输入表分区

逻辑回归配置参考上文的实验模型建立和训练

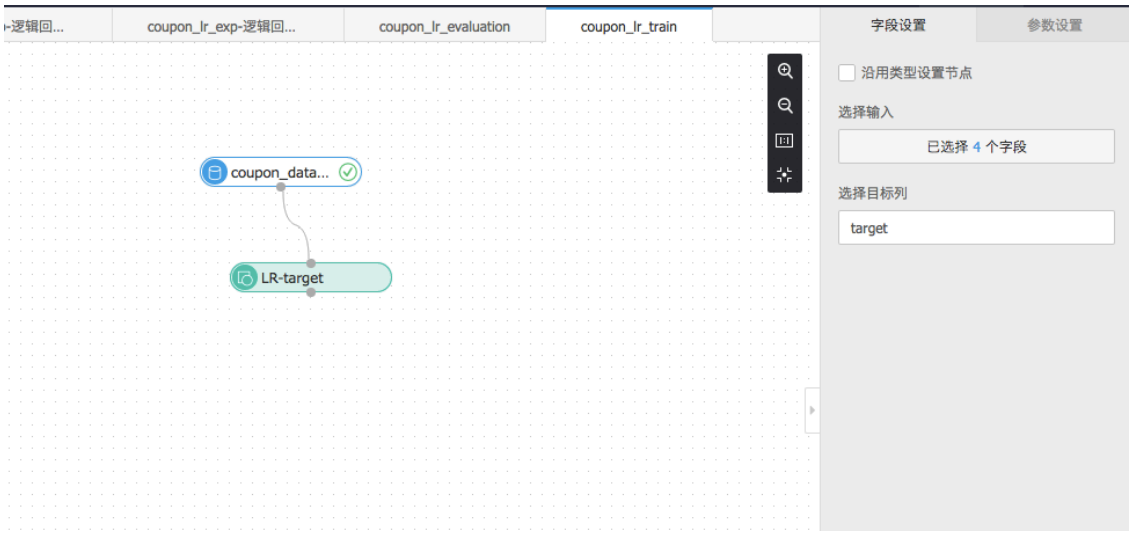


图 6.5.16 上线模型的建立和训练

6.5.4 上线模型的预测及结果导出

1、新建一个算法实验，命名为：coupon_lr_online_prediction。需要组件包括：

- ODPS 源：选择表为“coupon_dataset_4_test”

注：在日常生产开发中，应该使用未拆分的预测全量表。但由于本文档为演示目的，没有专门的业务逻辑，所以在上线预测的 ODPS 源节点中，直接使用了训练集中拆分出来的测试集。

- 预测：配置参考实验模型
- 模型：选择“coupon_lr_train-LR-target 模型”（上一步训练的模型）
- ODPS 目标：输入表为“coupon_lr_predict_result”

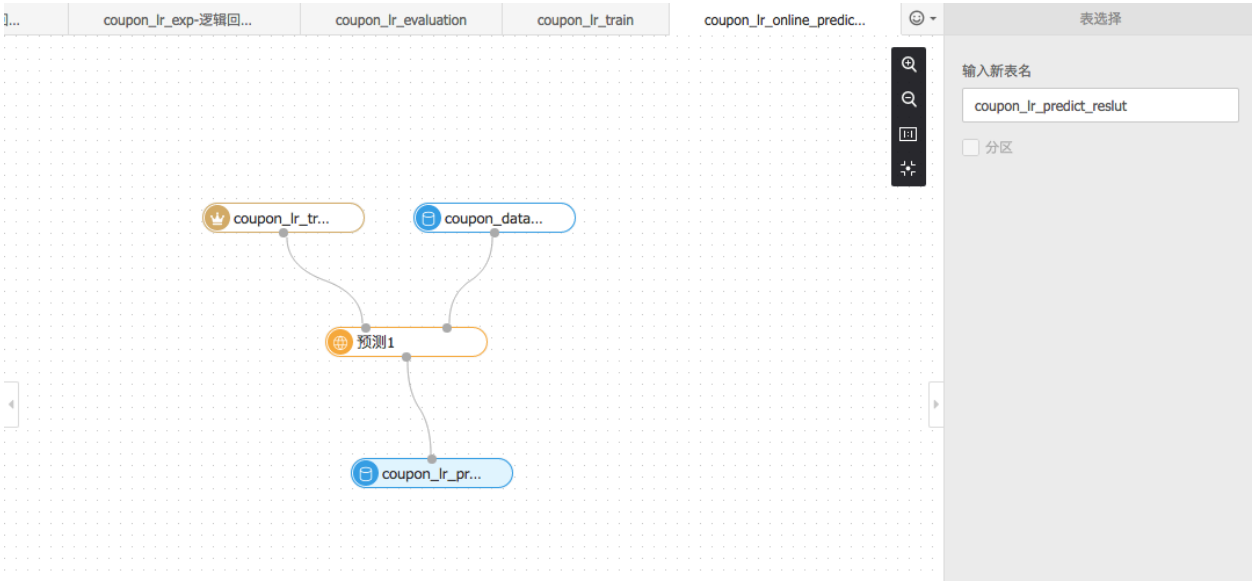


图 6.5.17 上线模型结果导出

2、节点配置

1、在 IDE 环境配置 coupon_lr_model_train 节点，在其中选择算法实验为：coupon_lr_train

图 6.5.18 节点配置 1

2、新建工作流节点为算法实验，节点名叫“coupon_lr_predict”，配置其中的算法实验为 coupon_lr_online_prediction

图 6.5.19 节点配置 2

6.6 MR 实现算法

通过 ODPS SQL 实现的训练集抽取是通过随机采样抽取的，在实际的应用中，可能我们需要使用别的方式来抽取训练样本，这里假设我们希望每个店铺都能有样本进入训练集，因此，我们希望通过 shop_id 分组后采样，这里介绍如何使用 MR 来实现。

御膳房 MR 目前还是邀请制，我们欢迎有志于大数据分析与挖掘的用户申请试用，具体请联系御膳房的运营小二（可发邮件至 yushanfang@service.taobao.com 申请，也可以通

过加入御膳房用户反馈群[1])。用户可以进入文档中心 - MR 文档详细查阅使用方法。

6.6.1 下载工具

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：

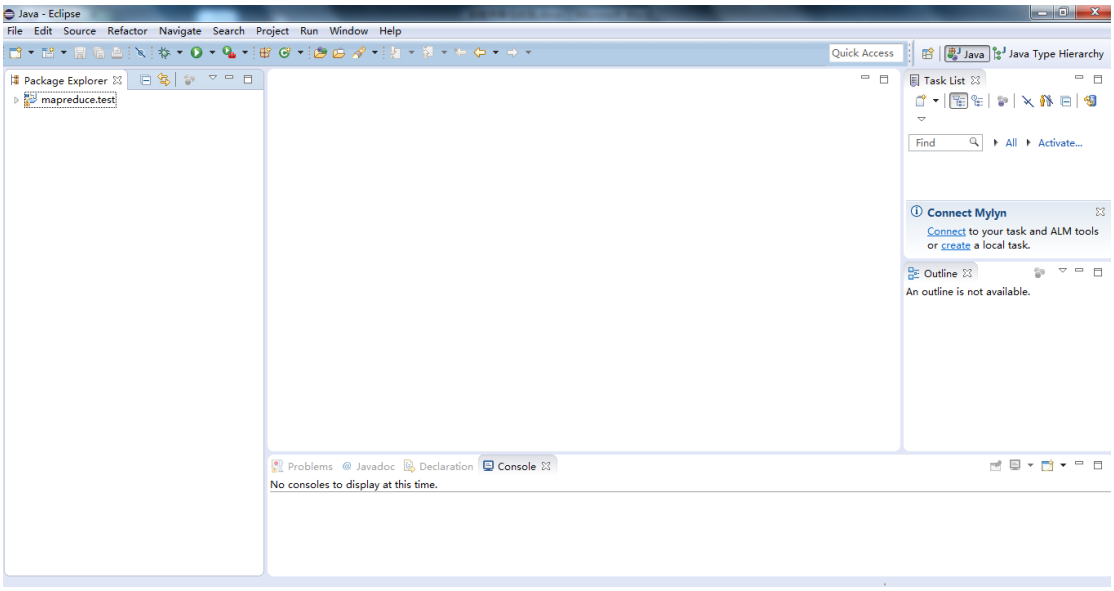


图 6.6.1 Eclipse 界面

6.6.2 新建程序

MR 开发工具通过 maven archetype 机制新建程序项目，御膳房提供 base-mapreduce-archetype 和 base-udf-archetype 两个 archetypes，通过远程 repository 可直接使用。

6.6.3 添加远程 repository

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。

[1]御膳房 1 群-用户反馈-技术支持 1：759773056。御膳房 2 群-用户反馈-技术支持 2: 1454609023。

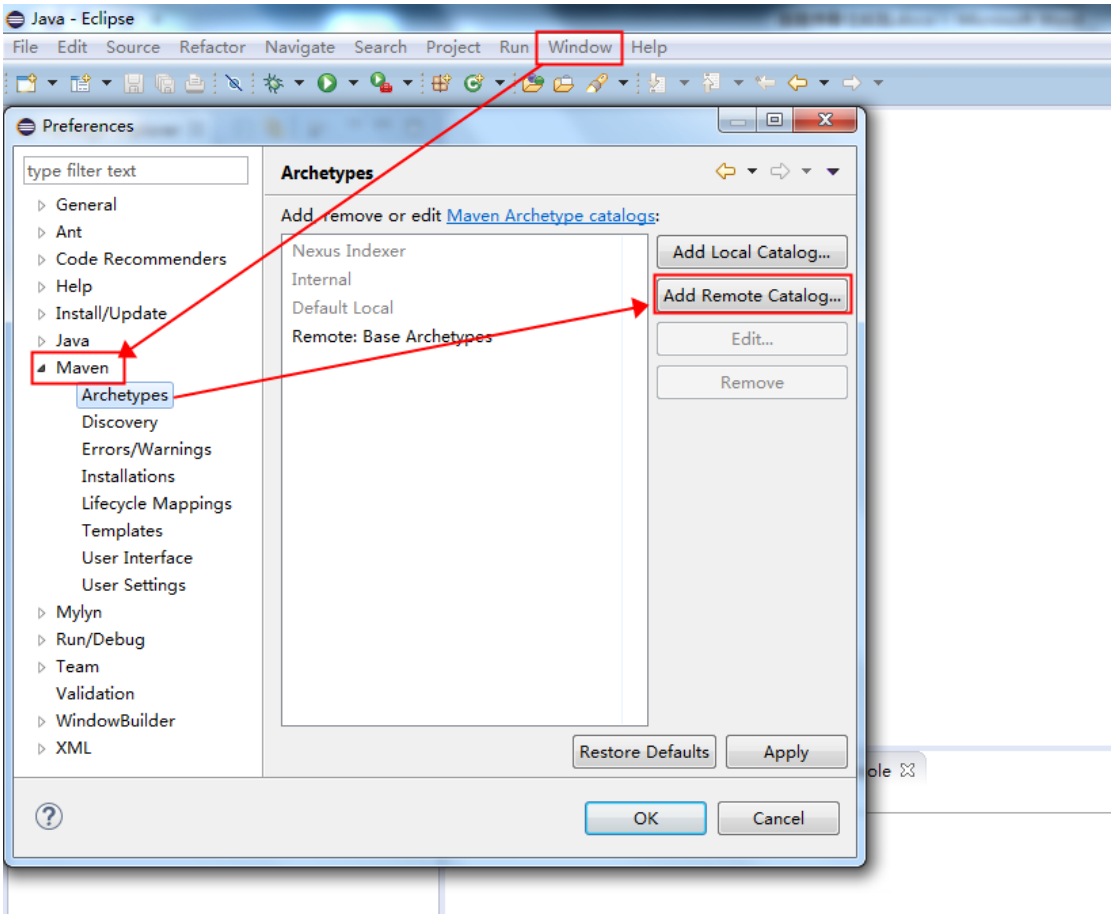


图 6.6.2 添加远程 repository 流程 1

2、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 OK 就完成了远程 repository 添加。

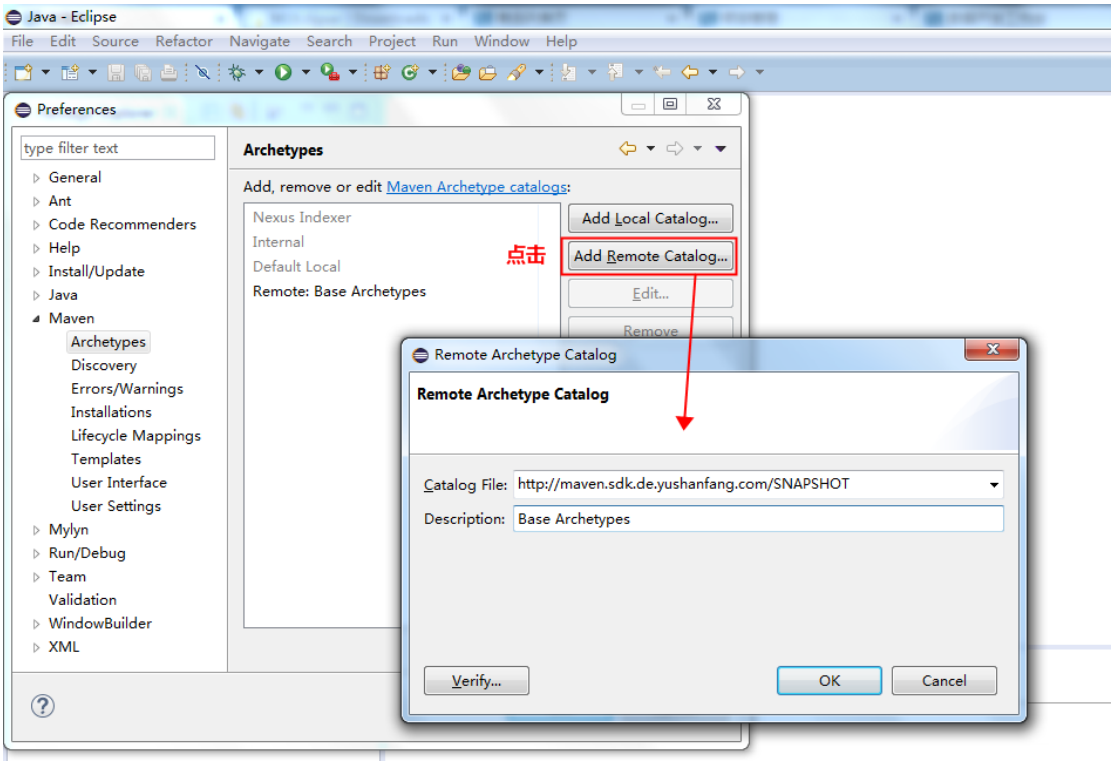


图 6.6.3 添加远程 repository 流程 2

6.6.4 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...

2、在打开的对话框中选择 Maven -> Maven Project，示意如下：

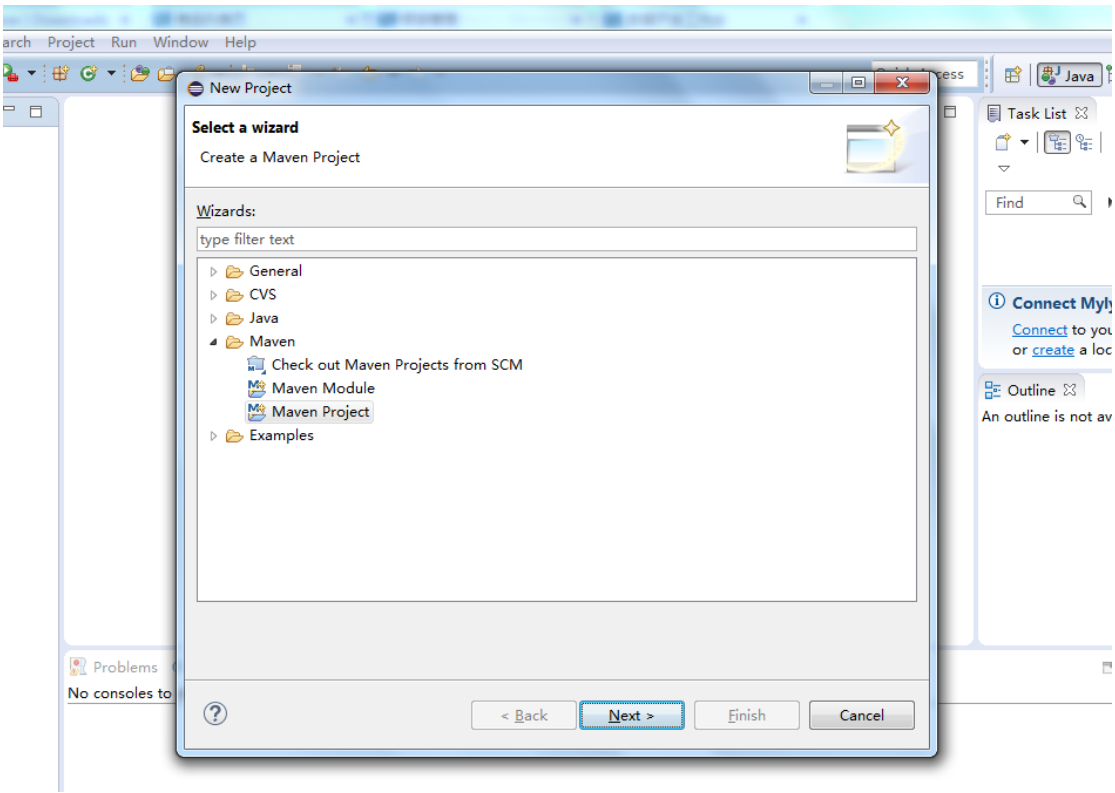


图 6.6.4 新建项目流程 1

3、一路点击 Next ，到达 Select an Archetypes 界面，示例如下：

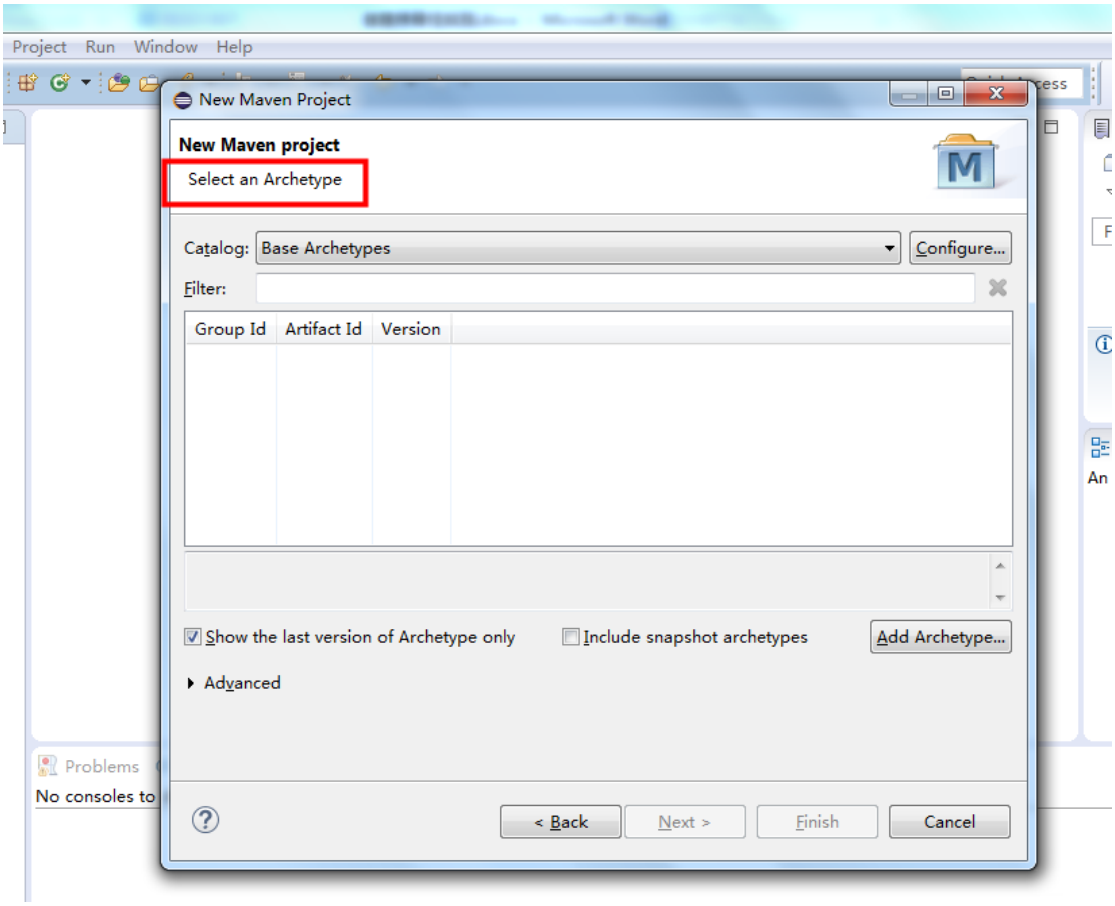


图 6.6.5 新建项目流程 2

4、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-mapreduce-archetype 和 base-udf-archetype 两个 archetypes ,选择需要创建的程序类型，点击 Next

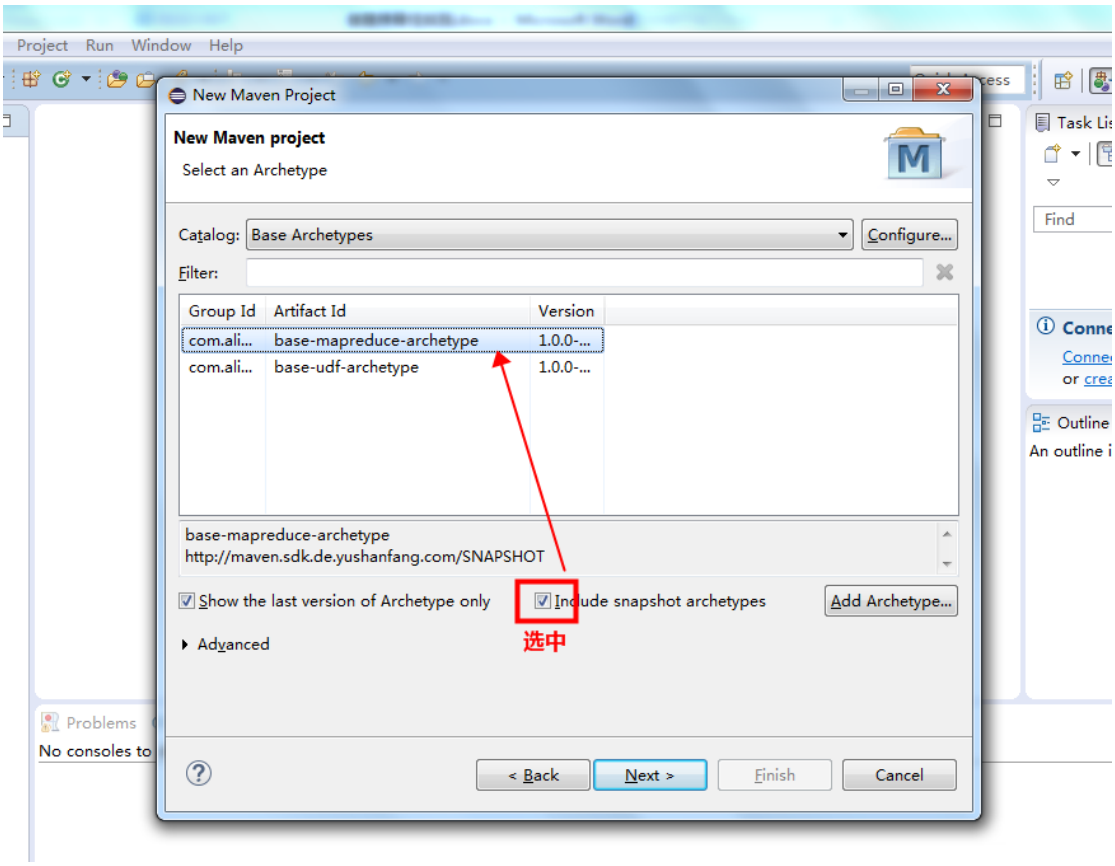


图 6.6.6 新建项目流程 3

5、以 base-mapreduce-archetype 为例，当点击 Next 后，会进入常规的 maven 项目初始化配置界面。Group Id、Artifact Id 填下示例如下：

Group Id：com.alibaba

Artifact Id：bestPractice

6、注意 Properties 列表中会出现几个御膳房特有的配置项，分别是：

- baseId – 御膳房用户的用户标识
- 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 的哪个工作夹下，需要工作夹已经在 IDE 中创建好，例如“工作流/最佳实践/算法实践”

上述 ID 的获取地址如下：

首先，点击“我的账号”

图 6.6.7 新建项目流程 4

然后，点击“证书管理”，我们就可以看到项目 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 6.6.8 新建项目流程 5

配置如下图：

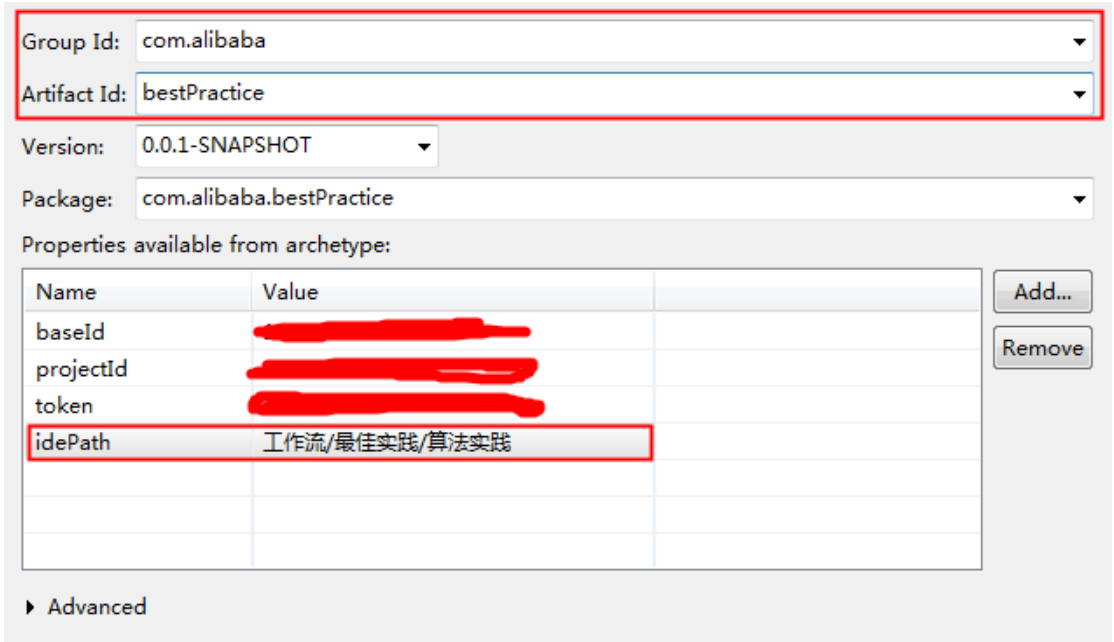


图 6.6.9 新建项目流程 6

这里将 idePath 配置成了“工作流/最佳实践/算法实践”，项目开发完，在 eclipse 直接提交后，提交的节点和对应运营代码就会自动在 IDE 的该目录下创建。

7、填写完成后，点击 Finish 就会完成项目创建，然后就可以在 Eclipse 的项目管理器中看

到新建的项目了。示例如下：

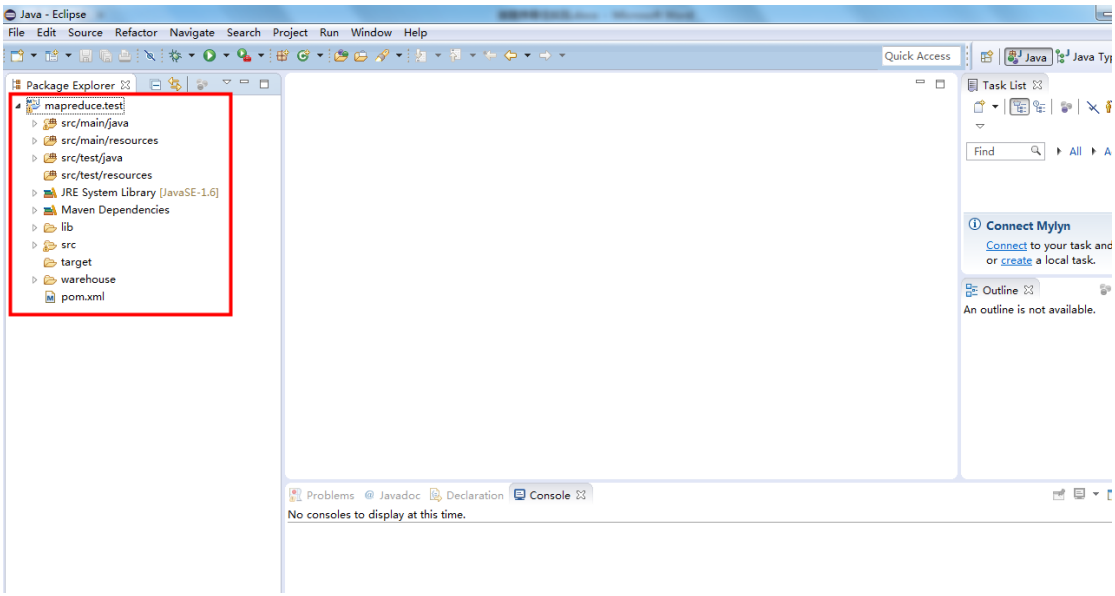


图 6.6.10 新建项目流程 7

6.6.5 程序开发

1、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。

下面介绍 mapreduce 项目的结构。

2、MapReduce

一个新创建的 mapreduce 程序项目结构如下（这里假设项目的 groupId 为 my.group，artifactId 为 mymr）：

```
.
├── lib
│   ├── bouncycastle.provider-1.38-jdk15.jar
│   ├── commons-cli-1.2.jar
│   ├── commons-codec-1.9.jar
│   ├── commons-collections-3.2.1.jar
│   ├── commons-digester3-3.2.jar
│   ├── commons-io-2.4.jar
│   ├── commons-lang-2.4.jar
│   ├── commons-logging-1.1.1.jar
```

```
|   ├── gson-2.2.4.jar
|   ├── guava-15.0.jar
|   ├── jackson-core-asl-1.9.13.jar
|   ├── jackson-mapper-asl-1.9.13.jar
|   ├── javacsv-2.0.jar
|   ├── junit-4.4.jar
|   ├── odps-mapred-bridge-0.14.0-rc1.jar
|   ├── odps-mapred-local-0.14.0-rc1.jar
|   ├── odps-sdk-commons-0.14.0-rc1.jar
|   ├── odps-sdk-core-0.14.0-rc1.jar
|   ├── odps-sdk-lot-0.14.0-rc1.jar
|   ├── odps-sdk-mapred-0.14.0-rc1.jar
|   ├── odps-sdk-udf-0.14.0-rc1.jar
|   ├── protobuf-java-2.4.1.jar
|   └── velocity-1.7.jar
├── pom.xml
├── src
|   ├── main
|   |   ├── java
|   |   |   └── my
|   |   |       └── group
|   |   |           └── mymr
|   |   |               ├── JobLauncher.java
|   |   |               ├── MyCombiner.java
|   |   |               ├── MyMapper.java
|   |   |               └── MyReducer.java
|   |   └── resources
|   |       ├── META-INF
|   |       |   └── base.mapred.xml
|   |       └── credential.properties
|   └── test
|       ├── java
|       |   └── my
|       |       └── group
|       |           └── mymr
|       └── resources
└── warehouse
    └── local
        ├── wordcount_in
        |   ├── __schema__
        |   ├── dt=20140101
        |   |   └── data
        |   └── dt=20140102
        |       └── data
```

```
└──── wordcount_out
    ├──── __schema__
    └──── dt=20140102
        └──── R_000000
```

项目主要分以下几部分：

- lib - 这个目录下放置 ODPS local 模式运行所需的库，用户不要动其中的东西，可以不关注
- pom.xml - 这个标准的 Maven POM 文件
- src - 按照 maven 规范组织好的项目源码目录结构，并且在${groupId}/${artifactId}目录下生成了脚手架代码和本地运行用的 JobLauncher
- warehouse - 这个目录下是按 ODPS 规范组织而成的 local warehouse，其中放置 mock 数据，用于本地运行和调试

默认情况下，所生成的脚手架代码是一个完整的 wordcount 程序，可直接以本地模式运行。运行方式如下：

1、首先通过 mvn clean package 命令编译项目

2、然后在 Eclipse 中的项目管理器中选择 JobLauncher.java，点击右键，选择 Run As -> Java Application。如果一切正常，应该可以在 Eclipse 中看到运行类似下面的输出

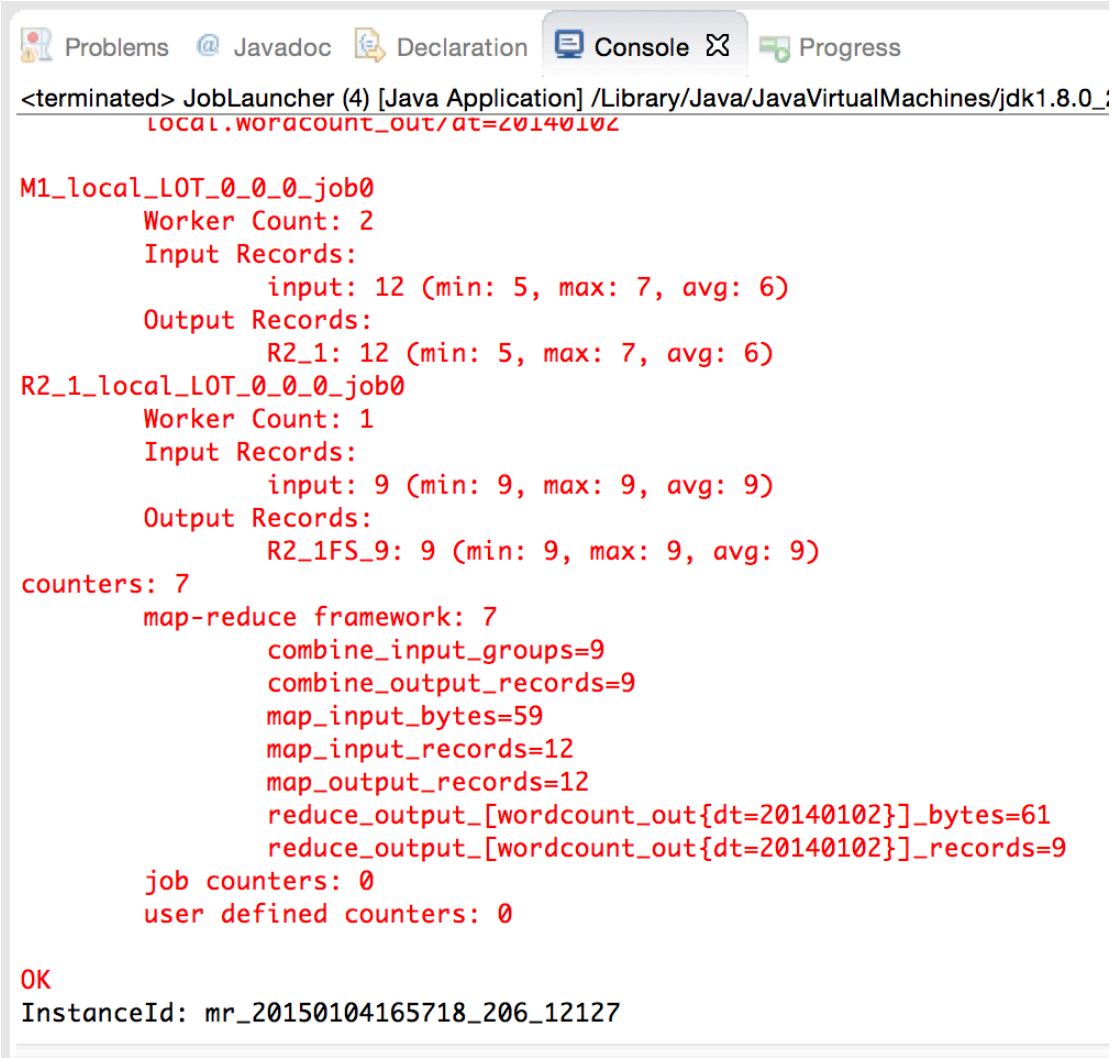


图 6.6.11 输出结果

接下来，我们可以开发自己的代码，在原有的数据集中，为了保证所有的店铺都有能样本成为训练集，我们可以利用 MapReduce 来进行分组抽样构成我们的训练集，新建的项目的目录结构如下：

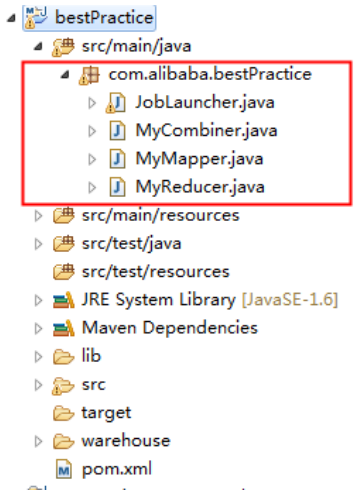


图 6.6.12 新建项目目录结构

可以看到，模板自动生成了对应的 MapReduce 需要的代码，其中启动填写的是一个 wordcount 的例子，我们可以直接修改其中的 MyMapper.java 和 MyReducer.java 来实现我们需要的逻辑。MyMapper.java 的实现代码如下：

```
public class MyMapper implements Mapper {

    private Record key;

private Record value;




    public void setup(TaskContext context) throws IOException {

    key = context.createMapOutputKeyRecord();

    value = context.createMapOutputValueRecord();

    }



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

        String shop_id = record.getString(2);

    key.set(0,shop_id);          // map  时以shop_id为key

    value = record;

    context.write(key,value);

    }




public void cleanup(TaskContext arg0) throws IOException {
```

```
    }

        }
```

对应的 MyReducer.java 的代码如下：

```
    public class MyReducer implements Reducer {

        private Record result;



        public void setup(TaskContext context) throws IOException {

            result = context.createOutputRecord();

        }



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

            double rand = 0.0;

            Random r = new Random();

         while (values.hasNext()) {

                Record val = values.next();

                rand = r.nextDouble();

                if( rand < 0.8 ){ //  按照80%的概率输出分组下的样本

                 result = val;

                 context.write(result);

                }

            }

        }



        public void cleanup(TaskContext arg0) throws IOException {
```

```
    }

     }
```

在这个实例中，没有用到 Combiner，因此，MyCombiner.java 不需要调整。

6.6.6 修改配置文件

我们需要通过 base.mapred.xml 来修改对应的 MapReduce 的任务配置，从而模板可以自动的去通过 jobLauncher.java 可以通过该配置来设置。具体的配置如下（请注意配置用户自有 baseID、projectID、resourceName 和 idePath 信息）：

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

<mapred>

    <!-- profile -->

    <baseId>742745df443fb8ad4c1b0ce77f758735</baseId>

    <projectId>1102</projectId>

    <resourceName>bestPractice</resourceName>

    <idePath>工作流/最佳实践/算法实践</idePath>



    <!-- classes -->

    <jobLauncher>com.alibaba.bestPractice.JobLauncher</jobLauncher>

    <mapper>com.alibaba.bestPractice.MyMapper</mapper>

    <reducer>com.alibaba.bestPractice.MyReducer</reducer>

    <!-- <combiner>com.alibaba.bestPractice.MyCombiner</combiner>    -->



    <!--task-->

    <mapOutputKey>masked_shop_id:string</mapOutputKey>


<mapOutputValue>masked_buyer_id:string,masked_seller_id:string,masked_shop_id:string,cat_id:bigint,pv:double
```

```
,add_cart_num:double,auction_collect_num:double,alipay_trade_num:double,target:bigint</mapOutputValue>
        <!--
        <partitionColumns>col1,col2</partitionColumns>
        <outputKeySortColumns>col1,col2</outputKeySortColumns>
        <outputKeySortOrders>ASC,DESC</outputKeySortOrders>
        <outputGroupingColumns>col1,col2</outputGroupingColumns>
        <numReduceTask>8</numReduceTask>
        <memoryForMapTask>2048</memoryForMapTask>
        <memoryForReduceTask>2048</memoryForReduceTask>
        -->

        <!-- tables -->
        <inputTables>
            <table>
                <name>coupon_dataset_4_feature</name>
                <partitions>
                    <partition>dt=20150625</partition>
                </partitions>
            </table>
        </inputTables>
        <outputTable>
            <name>coupon_dataset_4_train_grouped</name>
        </outputTable>
    </mapred>
```

可以看到，在配置中，我们使用了一个新的输出表 coupon_dataset_4_train_grouped，我们需要在数据引擎中先建好这张表，表的 schema 与 coupon_dataset_4_train 完全相同，这里不赘述。

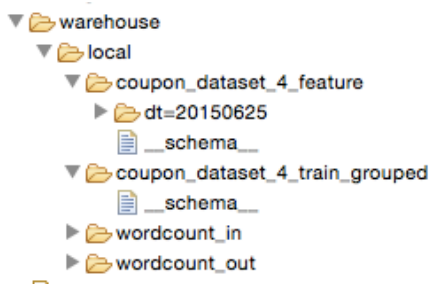
配置文件还可以配置很多其他的信息，具体的配置说明可以参照平台的御膳房 MR 文档。

6.6.7 本地调试

Mapreduce 程序支持本地模式运行，用于调试程序。程序在本地运行时行为与线上完全一致，只是数据采用本地 mock 数据，输出也输出到本地 mock 库。

1、配置 local warehouse

要使得程序在本地运行，需要 mock 一个本地的数据仓库，这个 local 仓库被指定为项目根目录下的 warehouse 。我们需要在本地手动的将 mock 的数据创建在本地的 warehouse 目录下，具体的创建规范可以参照平台帮助文档。本例中，我们创建本地测试数据之后的目录如下。(文件名为:__schema__)



其中，分区目录 dt=20150625（选择 20150625 的原因是，我们构造的 coupon_dataset_4_feature 是在 20150625 这一天创建的，所以分区也刚好是这一天）与我们配置中的分区日期是一致的。输入表 coupon_dataset_4_feature 的_schema_如下：

```
project=local

table=coupon_dataset_4_feature

columns=masked_buyer_id:string,masked_seller_id:string,masked_shop_id:string,cat_id:bigint,pv:double,add_cart_num:double,auction_collect_num:double,alipay_trade_num:double,target:bigint

partitions=dt:STRING
```

对应的在 data 文件中的数据如下，我们 mock 了 100 行数据：

```
00000b5961a3fb627e84afe11d8b6221,97540f0f6068d3d178243c317b3198e8,be25f3d3e228816d4926d0f5bb674ebc,50010850,2,0,0,0,0
```

00001555ba62d600bcec7f0201f026d7,a6a4b276e37fd3c4cab6238c7c9146d1,c2f1b695325f8f3e331813ba53ad0680,50000671,2,0,0,0,0

0000278dea037ae49ec2adf586f288c6,f4608ef9dabfc57191d4d2fbadecc578,314a28b38b3b3f9ec04ee5d74fe30f93,50000671,4,0,0,0,0

00004126a9a9827173bfce15b3e69c0f,31606859fadeff94343919c80846c566,d7b940307bd899559b2f840c14aad33b,162116,2,0,0,0,0

00006297e85bd67ef3aa0640ad5fcf9d,20b23da90796c056aeadb021bb37c59e,0087d709d81349d8fcbe7fc9d149cc58,162116,2,0,0,0,0

000070fbc68c8778fd7379ec3aa5dfb8,b73b8d0ba168151962fa5f4f80f5048d,bceaa7b8e8fb08b00bded637b54d3fa6,50010850,2,0,0,0,0

000073219d058394f9cc31ce7d727bda,0706611cbccbf29579314d5e7d33e1c1,637e923d02781bf05876e8e8eaf4c874,50010850,2,0,0,0,0

000079a1ccb363c86ead508722cba56f,0ba49d2f631e04277de322102ebe3eb6,49e7aa575e5466be5db619edc6984af5,1622,2,0,0,0,0

00008831c7c205a4eff3acfb655e5efe,5fd490500f6aa9ba258906f38d9f460c,af3e5c8bbafe4711df1eea4d1fdfc76b,50011277,2,0,0,0,0

00009e6b78adfb5e72e0032047db2af5,5a7a596911fd1bf68b5428abddf7110f,000d6386c2b38c1c6d261e877ed4bad0,121434004,2,0,0,0,0

0000aae28226ff8a5236729a1762064e,13b326b102beae105b29022f04296743,c01833c8c306f884447581027e692dfe,1622,4,0,0,0,0

0000ead9121ccd572e6fb9ab0e79dc6f,74e17ff16efa22d0dc2e56431fa61c39,034c1ffbcd5dddc0585af59270abc531,50010850,6,0,0,0,0

000117f16902a4f376cbc4d2dec1b302,9dff604399a37550652b628c4a128206,4846e4c2c6aba047cfea5020f765ee02,50000671,12,2,0,0,0

000127170e6f1f0b3f583093e955d9c4,2316b9dc4c0323254db4ed0bc1ffbbc2,c95a66448af574aba4898f64839d3f8b,121412004,2,0,0,0,0

00012925676d4da9e5c666897efa5b6a,b39ddf3b3e4bbc0462c5a843d416c586,75940c61f5096b0f988d631b36f7bb90,50010850,2,0,0,0,0

00012925676d4da9e5c666897efa5b6a,f2cb5462e7e7caf8a922d23746384b91,81c9f5f45075e500ff40d6add744

36de,162104,4,0,0,0,0

00012a54440d8e4d4db0f360e2ce246a,8655fa13ec38d0e28dadc31b3b8345ae,95b2428bc9a5badb854e531ca3eb755b,50010850,2,0,0,0,0

000144149dde7b5c101135ad69ac5385,ed0f6ccabeee5c510f113472321b0896,d9b3b70a1f039fa2b3a0aaac50722c2f,50010850,2,0,0,0,0

000148cd88cf1b3a5e701de1da13be47,2f488a625749b55dd432c6023b084200,4ad20b4fd2d0e87e524d1a56cb32c408,50000671,2,0,2,0,0

00015695d674ed3c55ce426e96a0c206,548615f9d332854fa8430a569c57b0b9,34615acb7f1e2c44f94862c4f0083a15,50010850,2,0,0,0,0

000158090458b83bf20eff296e07be1e,8fd694f4967a94b21b4881059c63705d,fe770bbee4d1458237192850436 10f02,162116,2,0,0,0,0

00015d77d9fd5f9dd0af1d137bcf633f,f1a1a3d89c734c2b5aafc55b4bf8bda6,2a831ada5b62e8cc63ecbb285ff737cb,1624,2,0,0,0,0

000165d3e3053ea1683f9807335b20d9,971c5d006b5f3f79b5c39f1a7b7be6df,31c97621b9b7c57e212f8e75144839b3,162104,2,0,0,0,0

000182ab22f9b0e4a5f62ba36e0b50d3,3e51797598ef48d7924b5070fa93962b,e72cfbc61d205acef5588b6272a2db68,50000697,2,0,0,0,0

0001c6794fa0d42118f8118804b6a42e,d4fb9a845239123efc78fd3d16fd7493,d3c7bdeaa23293ecb7dde5e58cf06fe7,1624,4,4,0,0,0

0001d07aa7c3916c481414ddc032368f,bc22ec9d95d47b982fb7ff43299b40a5,0ce6cf025a673c8adff8f5510dcefa13,50000852,2,0,0,0,0

0001d07aa7c3916c481414ddc032368f,ddb50748aa66f95a7ab781a2f9bbca61,d415342a989090b697e90b280daffef3,50010850,2,0,0,0,0

0001d5d334e5349c8b595a5250014add,5785d962f6d944d1dfd122fa503b6264,298a176a1f65152e72f9931f12e990cb,50011277,2,0,0,0,0

0001e6c119d3166c0bde1210bb0af6c8,498800af348612d1e9df8c62b54cbdea,552dee91002de08e58bbc40242231f5a,162104,2,0,0,0,0

0001e6c119d3166c0bde1210bb0af6c8,6f72e945ec380e22562975e0d6160ed3,d681afd5137ba69703e804e68116d8b1,1622,2,0,0,0,0

0001f9219c850635cc4da49afff8f58b,654fea80d47bf6168e5cac3b2756308c,1bb3145849d5a85b7fd3fb0cac0d46a8,50000852,2,0,0,0,0

000209e8a911116520345e25e7af334c,67c8b1c2a13f2eb82015b2bd6bdefaec,3e89cb660279517106f4c0b21145acba,1622,2,0,0,0,0

000209e8a911116520345e25e7af334c,9c63ccd017b4e154d4d0774a64dffc82,19f8ae637f5e171df5c0e9ed571e84fc,162116,2,0,0,0,0

000220b85b8538fcc3c9572b1eead28a,83297354af2f0330f937e70ee0fba5da,494085bcccc2eb413ab6fadc761de566,1624,2,0,0,0,0

00022c4d1e6c62ba817bfde8cd55b0e0,b5fe2363ab01d6394ee62f32c8a87185,f84e727457430808c14dd4cda1cc6219,50011404,10,0,4,0,0

00023290e372b6469327df4409bdf81e,5a19c07f33729337eb18f6e3fc32138d,2a376d4aadb709064f6cbed0c3a0854d,50000697,2,2,0,0,0

00023811429bbc11711d32396f562b43,ba6ee1a67189f122331506ac783c27f6,22db7c99f3ee7615f876c8ad126dab68,162104,6,0,0,0,0

00023811429bbc11711d32396f562b43,c57497b85861b87fed4cb11444d6cad3,dda7be2aa41552d5dac1ae6432082ead,121412004,2,0,0,0,0

0002800bd7e617e718d62e2b4d34dfc5,ff971269691ad07a19dae014c215ca34,102f5a6a11bfb10be04bafd6b98032e2,50010850,2,0,0,0,0

0002914cf687aaacd5d1c2517fce7112,abd9041c32e05a3d822195a3cd5a26b4,f4501963089adc1e6704e78c8134919f,50000671,2,0,0,0,0

0002c73c41b64cd21be8b7b29a29e7c0,a41cde3ad59c5e09ad0c6a40d4bdcb83,d7bf0056d7eb69eeb0d8add0aded33ff,50011277,2,0,0,0,0

0002e186b5426666af647bcdd2042ab6,babec6efc89808c8bb47a6e6b6869a58,4521c2345743c86dc80e0f67fc3248e3,50010850,2,2,0,0,0

0002f2af308bd0edfbbf6de7ab6c991e,b68a10310a62713febc4dc79f329704d,04171bde55b0ea70427db501c8202234,162116,2,0,0,0,0

000309712702a21ef48e8e9145ec35ea,bfa9d206982ff4180a3937ea29916fa1,634d8c23d00cd23c9965cb5577fa7ea4,50010850,2,0,0,0,0

0003177f495e958b35e9c18fe6d55605,cf39fa888a73f662f0d2c4be8cfe9599,bd47d9e5a632fc37644a5f1e9990f

6b4,162116,2,0,0,0,0

00031c8abe91cff564e27d011dfb25d4,7bfb43c033893e3c42bada878c374f7d,b7bc810de3a35f9add2f3a2e451f24ad,50000671,2,2,0,0,0

000330be33c51b413917e000ed019fbb,c109a0e31372d3121f3742e546f76bc6,bbed60d77fa996a6f346ed4203998907,1623,2,0,0,0,0

000340306be47af296e298476b56d5cf,ed75cbdf27681783b03b0f1b474ca0b5,d6d53c34f9fc6dd71a3ceb714bc2b255,1624,2,0,0,0,0

000354b2653460208c2ecaf4fcb960c7,2e0eb42e323da311f657f33117f13c0c,d60a3fb6b6f0f0a9f55705b3f2a314f1,50011277,2,0,0,0,0

00035bce9d60d7775e8a2636564a0c4e,6dad14bafb8016cf0a7f20e106db779a,d2187f455c91d026b908228158e1185d,50000671,2,0,0,0,0

0003697f9e2cb5b656d81206b62ade00,2610f6fbe88190b3dfb92d1ad936eb28,ed92063d093bc77e861f605000b33467,50010850,2,0,0,0,0

000380f66e95e098639c8cf3af1f75f0,bde3f748d275860404a9d06a7ffeaa7f,a85c6c91ceb498148df5aa39d8adccab,162104,30,0,0,0,0

000382b616c6bb0f7257d189d071b6bd,8f681f2adb01b4f0c1118c808aedfdd6,2f67ddcae1f34f427cbcca13c53f11a4,162116,2,0,0,0,0

000385e39034057e9fe55321e593d0a4,f54469c215adcf724e73f4a84715ee47,d72d8d62716e1d37825b9417c19bf3a5,50010850,2,0,0,0,0

000392d9291e77321a11d5356cd5c1f9,1e766985f26ad5fc541c45805e97ba3e,9210351d6e883f69356e8274ba8aad4d,1623,2,0,0,0,0

00039bb117ac7b070f65398441a195aa,3bd0df5ee3f440cdedd5a2b7ecb39b3c,4304566098300d1fb57e71117a10a115,162116,2,0,2,0,0

0003cfb99e625536b45e8b576c814992,f0280a1b0301303da47c1d72332c4503,5533666ebe9ffe08668e66cb3daebbcc,162116,2,0,0,0,0

0003e39979bde64adcfba5ef26399fda,bdc6f0a6382585eeadc6daa3116364d1,81ae0af423d38ee6ef07d50bff882658,50010850,2,0,0,0,0

0003ed95f6e846d0c60cf2257fbc4ca8,0c3123f5515c6b8bba549173140db7da,be5ad8e90da4feee72be3790ff29cd51,1629,4,0,0,0,0

00040556ef2f6afdd8427ce42b95c041,d8bb6b57ed29c273057323065999d833,46e24df0e6d331bcc313213f4011648d,1622,2,0,0,0,0

0004065d443961ede1827b6cc0a66f84,0e5610e8e29bf9e9af29189482c5e717,59babaaf8fad0aee5ee158764137b6a6,1622,2,0,0,0,0

000408ac112882b01992709152f55265,2d15aaa308badb2e584bdafb84f9f1ff,190936c9de0d77ea0577fbf05e945e46,1623,2,0,0,0,0

0004175c111d6372dc68fcc69d683ec3,e10c5688b3208e4455315f0eb46f4aa3,9c3806f0fcb495e49673a0084c26dba0,1624,6,0,0,0,0

000425b9aa8267200bcb8672e7a872e3,258fd61d8f5d4f32cdbef1e2e638d555,101fc693a61786e627fd03641b6ae410,50010850,2,0,0,0,0

000425b9aa8267200bcb8672e7a872e3,6a1b001350dbd9a08697238586967ae1,ec5121f2d1e76d248fd12912d978755d,50010850,2,0,0,0,0

000425b9aa8267200bcb8672e7a872e3,b28f7488ba7497661c90aa0b61baf79e,0711c264946a693bf5111165d4660c64,50011277,4,0,0,0,0

00045135fe6762337f1f09002d9c49a0,4003ee78dc0ad8e2f75ea66d58838810,96efcaf0e323a7cfb5214963bacbe1c1,50000671,2,0,0,0,0

00045135fe6762337f1f09002d9c49a0,4d8c377cc262180bc35b9ff5361d971d,70bed5b210ab564f21079538fff5d266,162116,2,0,0,0,0

00046d9276b8f40349bde1a24362fa21,f94582e7b3a0ae8d5aacd807c35620d2,b37f4be2aa27333d3350eec46ce21069,162116,2,0,0,0,0

00047df19ca2e42fb5578a150159e9d8,dac68faafd66ffa3e9c5fd2c9c8d5700,fab63986757a0f5915b131dad32274a9,1629,2,0,0,0,0

00048a75437bc85351e8a0d0c553fe2e,a5a589a2e9fd9f3b676d133472dd25fe,df7e1f048f671c02445cef9e68b696e0,162116,2,0,0,0,0

00048e2508fd5a7ed302d37ea7893e0a,ad24d8d28fe0ca84223098d0213a9bb2,9853b5618165252feef59992cb4feaac,50010850,2,0,0,0,0

0004ab07da916d1c7e3bf66539d5b871,444a61d78c4bf7cf0d5b275f98726628,2882f2a614357a457faf61cdc57579a6,1622,2,0,0,0,0

0004b0ae267a68e9f286fed01ec23e01,fcfdc41ddd8de3dac3eda4f46996f55d,4adda479863f2e62f2c6ce03a523bf

1a,162205,2,0,0,0,0

0004d1b09c9bcddcc0fd671d7fe0d6c2,69e6c3c370aaf089d0002713cd87e635,76d46d410fbc25da19ec742dc688de77,50010850,2,0,0,0,0

0004d3c8ffebe9d39c1c23bb72f83d4f,4f60f0a7e554082c16245e76db89d8a8,e102e08580d99d2285542928f18fdc5c,50010850,2,0,0,0,0

0004d5613c5ff762d81412116f0b0fa0,0d668c12b08e335d972365732b5b804d,ac2c23c92ed73418edbe875ac8114c44,50010850,2,0,0,0,0

0004d5613c5ff762d81412116f0b0fa0,b083e20a5fedcae8d91a5f57948f836e,805677d7280dd369b018374c92e8c4fe,50010850,44,0,14,0,1

0004e436e3fa031ed854b2d1d58180df,077017c8e05ddd1f273bcf2e1fbae99c,00c43048752adad80ba97dddfa13b9b1,50010850,2,0,0,0,0

0004ec25dc82db1754e5f391c0d877bd,203294bd531b5932def80d6e1661ccea,bc792e777fccc5edf6a44145e0f9e8c0,1622,4,0,2,0,0

0004ec25dc82db1754e5f391c0d877bd,a2894a27b1a459db9b5ad7efece1d23e,46d58ffcca787580d7dcefc26b567343,1622,2,0,0,0,0

000516d34586777cb7dfc265dd2f86a2,2be2ebb340ca55fff44b80e4c65ead12,1e36f3448e513ef64fd03da6031d44d1,1622,2,0,0,0,0

00053589829a09c35d29ed28d94f34cb,518cbc8ad08dfbf386c16d7fc1d0ccd5,2ce0ab3f4cdb03be705af9b302607b32,50000671,2,0,0,0,0

00054abf06c1170b08cef9eeb80772ff,29227eb4a39f8d9a1c8aa1c36e253ab5,ec3190a04fdb35e303964870450a3691,50000671,2,0,0,0,0

00055560219bcd06903ba1298c305c9d,89ea8d4e8fde092e6b6ac295140e7ccb,9855deef24a4e861ae49ef0100988eab,1624,2,0,0,0,0

000566ac75e46203fbada1f3f889afd0,fa80af77398dbd714912f5d96186a951,70cca034ab1a619adf7b58cea4602dbc,162116,2,0,0,0,0

00056b7f19c5e79cb9f23c8fb16225b8,9b3973301171aafc6cd074a9235183dd,3d52570248d6f697e8ed09f674160b98,162116,2,0,0,0,0

0005bce487502cb83da8aa1b0cd38a27,77a4d4843cbadfddd0de08fde9ea185d,a377eb625fa3dee76aa651135a7c880c,1622,2,0,0,0,0

0005c0a98c9060199e44e612aaea7bdc,f091210e7400a70a0999568fa7c6d75e,5c733c43d181b646a9cc5f2063d81037,1624,8,0,0,0,0

0005cd97ad622aff4c941e3f40c02bae,c612048a91f923917a187fe4e6dce3fa,432709f2a9eae4be89e35d60870e5d04,50000697,2,0,0,0,0

0005ce586324225f7d81e3b47e6dc5d8,61f420af4deab179caba15aad734e6cb,c2bd42cf6607d25d62e1f8f3028fd4c7,50010850,2,0,0,0,0

0005cfc333e3a38bf5ce8f69dd23aeb6,6090c9803b3f7632c5572c3dd3b429b2,c70aea895a73c91f96630405921be040,50000671,2,0,0,0,0

0005d498745d5d142ce0014f2071bd63,2aad7b2ebd25a8bf600804f118109401,7ed3e64fc9c780b35667dd6173d21a5f,1622,2,0,0,0,0

0005d498745d5d142ce0014f2071bd63,cdad5f2746aa51f4b135fe77c5fcafbc,2df67125b6221bb2fbdc9d2486343053,50010850,2,0,0,0,0

0005dc31ae923b7596d0595018cdfc7b,340d883f77627da683d57132c04638b9,3aea603e8407755230947291da2e5b93,50011277,2,0,0,0,0

00062cbfed659f89911fd5625de21da8,579c1e11702a9999b9fee88665eaf1a4,22990b9a70c38e632b49ae2b7f40cf6f,1622,2,0,0,0,0

00067975b783a43754be3f91f6fd7b72,b9bd20adfda68286160fabf4fb6016a7,5d004ee8ca3c3947abc89a4d7e95615d,1622,2,0,0,0,0

00067b2bcf23474e0d491b09c28936dc,71966adf196be24544f289711ece0cd9,88efefe5b806b937eaf38d5bb2f0a5c3,50010850,2,0,0,0,0

00067b387f13e7299902d0254e482c8d,5b910adeec175713fc9d083b6ff8555f,a77130ed7391cb9f73016807a3bc188e,162116,2,0,0,0,0

00068e46ca4d5beb45009d8fa279af3e,47a6d5376644518604be4a7df53b81e4,11e90a796269369ef491e4d6291b2196,1624,2,0,0,0,0

在输出表 coupon_dataset_4_train_grouped 的_schema_文件如下：

project=local

table=coupon_dataset_4_train_grouped

columns=masked_buyer_id:string,masked_seller_id:string,masked_shop_id:string,cat_id:bigint,pv:double,add_

```
cart_num:double,auction_collect_num:double,alipay_trade_num:double,target:bigint

        partitions=dt:STRING
```

2、配置完成之后，我们可以在本地，run 一下 jobLauncher.java 了，运行之后应该能看到如下的执行信息：

```
Summary:
Inputs:
        local.coupon_dataset_4_feature/dt=20150625
Outputs:
        local.coupon_dataset_4_train_grouped

M1_local_LOT_0_0_0_job0
        Worker Count: 1
        Input Records:
                input: 100 (min: 100, max: 100, avg: 100)
        Output Records:
                R2_1: 100 (min: 100, max: 100, avg: 100)
R2_1_local_LOT_0_0_0_job0
        Worker Count: 1
        Input Records:
                input: 100 (min: 100, max: 100, avg: 100)
        Output Records:
                R2_1FS_9: 82 (min: 82, max: 82, avg: 82)
counters: 5
        map-reduce framework: 5
                map_input_bytes=11657
                map_input_records=100
                map_output_records=100
                reduce_output_[coupon_dataset_4_train_grouped]_bytes=10220
                reduce_output_[coupon_dataset_4_train_grouped]_records=82
        job counters: 0
        user defined counters: 0

OK
InstanceId: mr_20150626153843_680_5634
```

此时，我们刷新一下 warehouse 目录，我们可以看到 coupon_dataset_4_train_grouped 目录下出现了一个新的文件 R_000000，这就是程序执行之后输出的文件，我们可以打开看一下，内容就是我们的训练集文件，应该有 79 行数据。

6.6.8 提交程序

当程序开发完毕并在本地调试通过，就可以提交到 IDE 中运行远程试跑和部署了。提交是通过 maven 插件完成的，下面以 Eclipse 为例说明如何提交程序。

1、配置 settings.xml

由于 BASE 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，请查阅 maven 的文档确定本机.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。

需要配置的 settings.xml 项目如下：

```
<?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>
```

如之前系统中没有 settings.xml 可将上面内容直接新建为 settings.xml ,否则请将配置项合并进已有的 settings.xml。

2、base-maven-plugin

御膳房的 maven plugin 有三个 goal：

```
# 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
```

另外 base-maven-plugin 有几个特有的配置项：

- base.program.type - 程序类型。可以是 mapreduce 或 udf
- base.endpoint - 特定御膳房实例的 MR/UDF 提交服务地址 http://api.sdk.de.yushanfang.com
- base.ide.url - 仅 mapreduce 需要。特定御膳房实例的 IDE 地址 http://ide.de.yushanfang.com
- base.ide.resource.url - 仅 mapreduce 需要。特定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}

3、提交程序

下面举例说明如何在 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...

2、在左侧列表中右键点击 Maven Build -> New

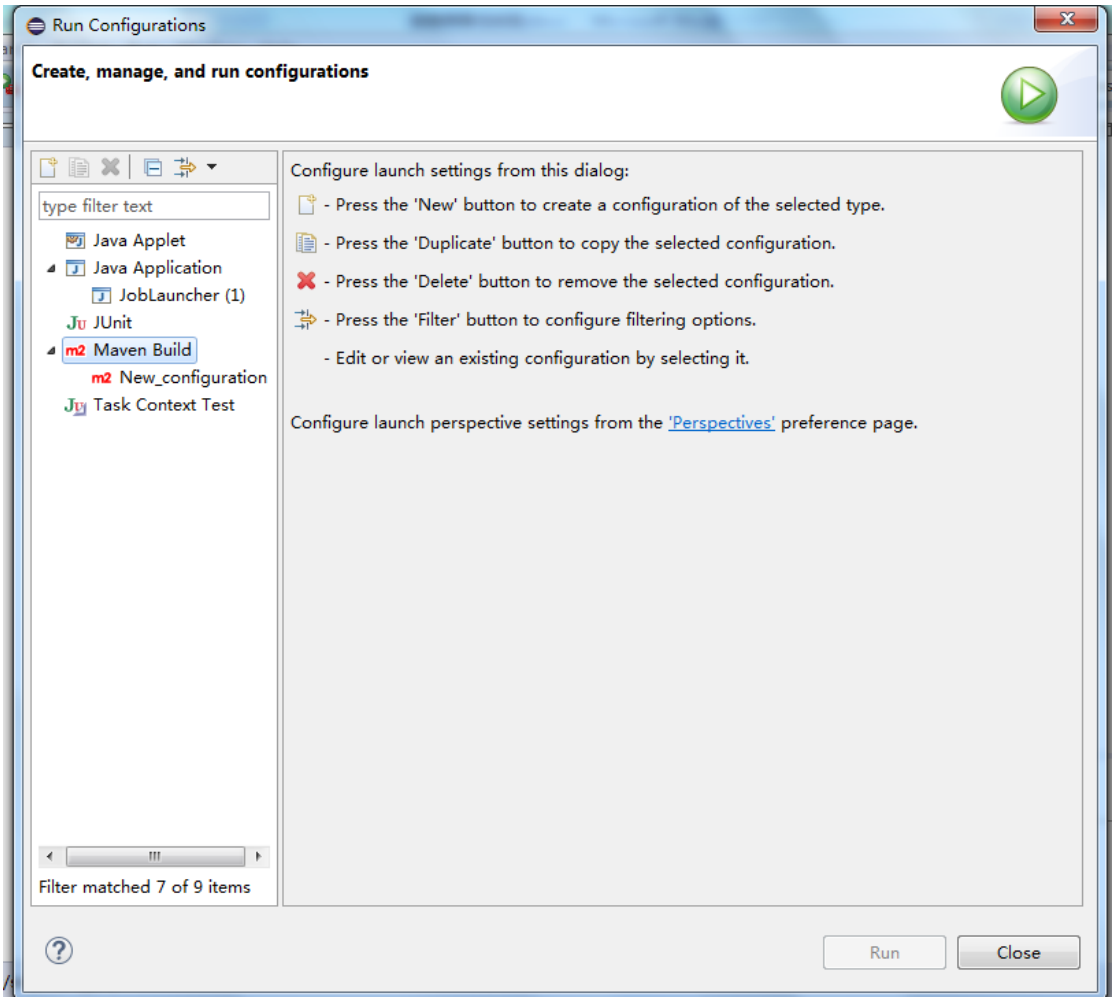


图 6.6.13 提交程序流程 1

3、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

4、Goals 中输入 base:check base:zip base:submit

5、Parameters 列表中填入上文提到的 base-maven-plugin 配置项

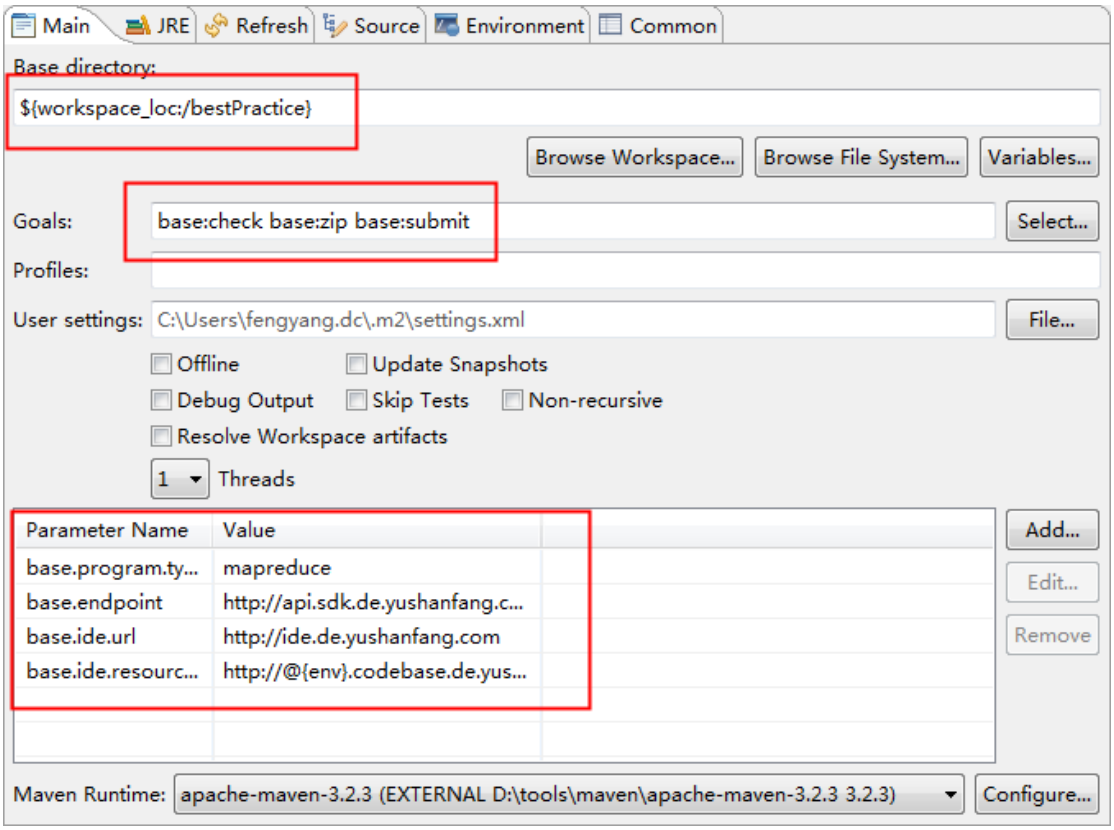


图 6.6.14 提交程序流程 2

7、然后点击 Apply ，再点击 Run ，Eclipse 就会自动调用 base-maven-plugin 提交程序。

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。创建好的目录结构应该如下：

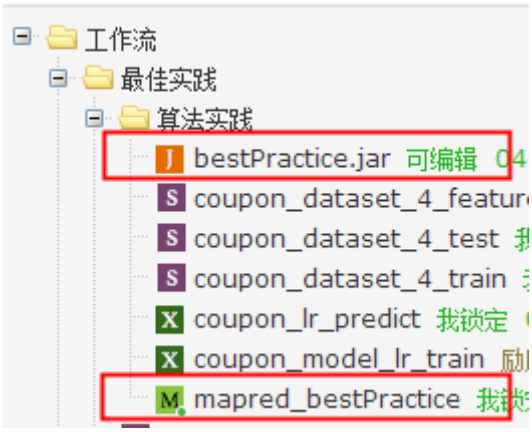


图 6.6.15 目录结构

打开 mapred_bestPractice，结果如下：

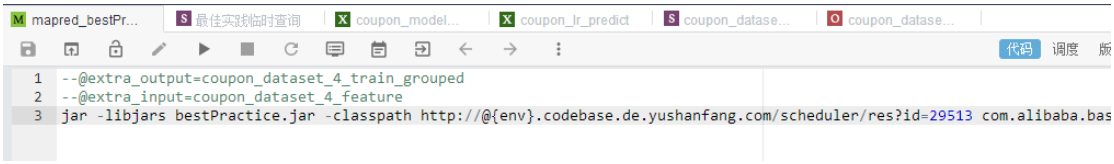


图 6.6.16 提交程序流程 3

系统自动生成了执行 mr 的命令，因此我们只需要直接执行该程序，就可以在开发环境使用真实数据运行了。

6.7 部署任务

6.7.1 提交表到生产环境

依次将以下三张表提交到生产环境。

- 原始特征集表 coupon_dataset_4_feature
- 训练集表 coupon_dataset_4_train
- 测试集表 coupon_dataset_4_test

图 6.7.1 部署任务流程 1

6.7.2 依次发布三张表的开发节点

将三张表的开发节点发布生产环境，并在生产环境中测试（注意测试顺序）。

1、在“数据开发”中进入每张表的节点。

图 6.7.2 部署任务流程 2

2、点击“调度”，进入配置调度。

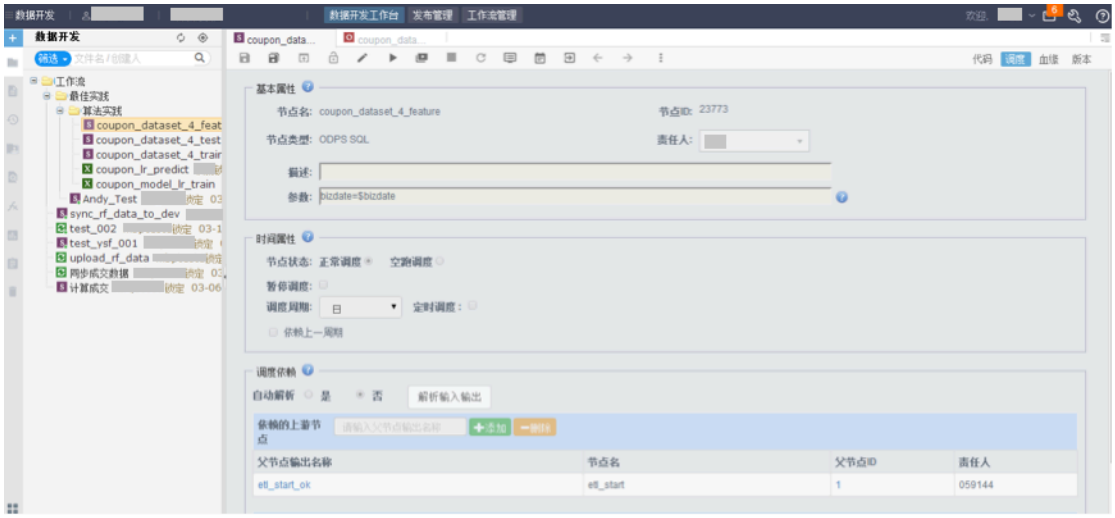


图 6.7.3 部署任务流程 3

3、打包发布。

图 6.7.4 部署任务流程 4

4、进入“工作流管理”。

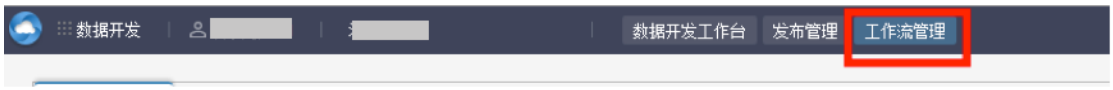


图 6.7.5 部署任务流程 5

5、右键点击节点，选择“在生产环境测试”。

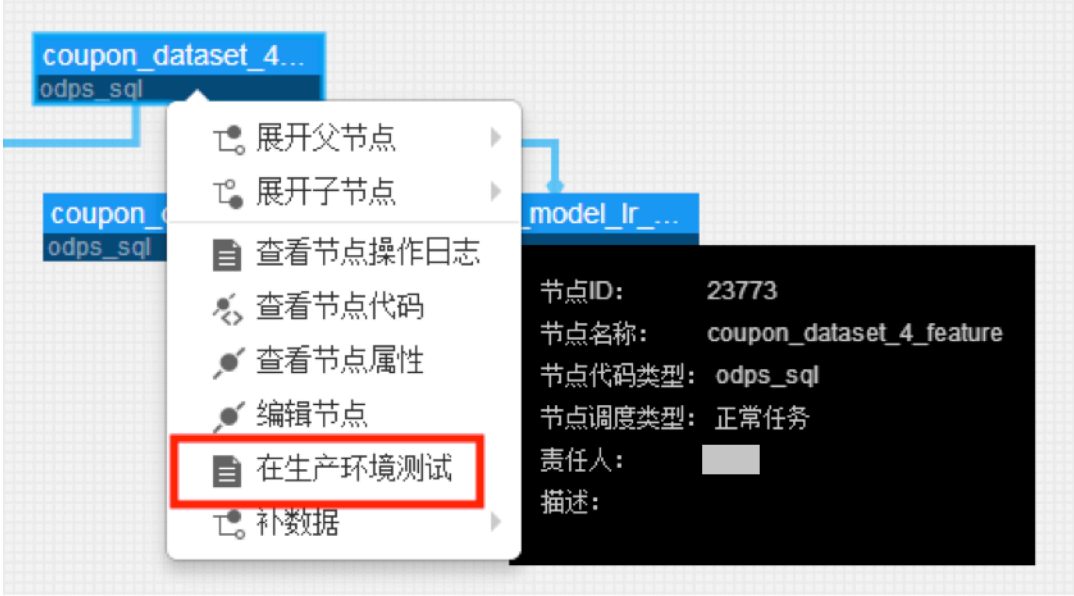


图 6.7.6 部署任务流程 6

6.7.3 将算法实验的两个节点上线

将算法实验的两个节点上线，并在生产环境测试（注意测试顺序）。步骤同上。

七、数据导出

7.1 准备数据目标

目前御膳房支持将数据导出到阿里云 RDS、阿里云 ADS、阿里云 OSS、阿里云 OCS、OT、UMP、SFTP 和自建 RDBMS。

目标数据源的添加步骤详见“数据上传----准备数据源”

7.2 数据同步

数据导出任务的配置和数据上传任务的配置方式一致。

至此完成了程序开发的所有工作。

八、总结

本文档对御膳房的应用案例进行了详细的演示，侧重论述了如何利用御膳房提供的数据开发、建模功能，进行数据的分析、模型的建立与评估、以及模型的优化，涵盖大数据分析中的主要流程和应用。在完成线下模型开发之后，文档给出了如何将开发的模型部署到线上生产环境中的操作演示。模型最终产出的结果可以直接导出到线上数据库中供应用方调用，也可以使用 ODPS SQL 脚本等继续开发后续逻辑。

文档中使用的案例来源于线上系统“江湖策”的精准营销功能，演示时利用简化的数据和模型完成了对于用户购买行为的预测。可以看到，应用平台各类工具，能够方便地完成大规模数据的处理、探索、分析和建模工作，支撑应用需求。本文档的这一实践，可作为基本应用样例给开发者和数据分析师提供参考。