ALIBABA CLOUD

阿里云

机器学习PAI

PAI-Studio 可视化建模

文档版本：20210607

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.概述

PAI-Studio提供可视化的机器学习实验开发环境，实现零代码开发人工智能服务。同时，系统提供丰富且成熟的机器学习算法，覆盖商品推荐、金融风控及广告预测等场景，可以满足您不同方向的业务需求。

PAI-Studio支持阿里云主子账号登录方式。如果使用子账号，则需要主账号对其进行授权，详情请参见授权。

PAI-Studio支持使用模板或手动创建实验。通过模板可以快速创建实验，运行成功后，直接进行模型部署。手动创建实验时，系统提供百余种算法组件，并支持接入MaxCompute表数据或OSS数据等多种数据源。

进行模型训练时，系统支持AutoML自动调参及导出PMML（Predictive Model Markup Language），辅助您获得最佳模型。

PAI-Studio提供的算法组件包括：

- 传统机器学习组件

  包括数据预处理、特征工程、统计分析、时间序列、文本分析及网络分析等算法组件。

- 深度学习框架组件

  包括TensorFlow、Caffe、MXNet及PyTorch等深度学习框架。

2.授权

PAI-Studio与DataWorks共享项目空间，如果使用子账号创建MaxCompute或PAI-Studio项目，则需要为其进行授权。

子账号授权

1. 登录RAM控制台。
2. 在左侧导航栏，选择**人员管理 > 用户**。
3. 授权。
   i. 在**用户**页面，单击**操作**列下的**添加权限**。
   ii. （可选）在**添加权限**面板，单击**自定义策略**。
   iii. 在**选择权限**下的文本框，输入AliyunDataWorksFullAccess。
   iv. 单击**权限策略名称**下的AliyunDataWorksFullAccess，使其显示在**已选择**列表中。
   v. 单击**确定**。
4. 创建AccessKey。
   i. 在**用户**页面，单击**用户登录名称**。
   ii. 在用户基本信息页面的**用户AccessKey**区域，单击**创建AccessKey**。
5. 使用主账号登录DataWorks，为子账号配置MaxCompute项目权限，详情请参见添加工作空间成员和角色。

OSS授权

PAI-Studio支持OSS存储，需要主账号对服务关联角色进行授权。

1. 登录PAI控制台。
2. 在PAI控制台首页，选择**模型开发和训练 > Studio-可视化建模**。
3. 在**PAI可视化建模**页面，单击**操作**列下的**进入机器学习**。
4. 在PAI-Studio控制台首页，选择**设置 > 基本设置**。
5. 在**基本设置**页面的**OSS访问授权**区域，选中**授权机器学习读取我的OSS中的数据**复选框，其他参数采用默认配置。

GPU资源授权

PAI-Tensorflow底层使用GPU计算资源，需要对GPU计算资源进行授权。

1. 登录PAI控制台。
2. 在PAI控制台首页，选择**模型开发和训练 > Studio-可视化建模**。
3. 在**PAI可视化建模**页面，打开**开启GPU**开关。

3.常规机器学习组件

3.1. 源/目标

本文为您介绍PAI-Studio提供的源/目标组件，包括读数据表、写数据表、MySQL数据同步及OSS数据同步。

读数据表

读数据表组件用于读取MaxCompute表数据，默认读取本项目的表数据。如果跨项目读取表数据，则需要在表名前添加工程名。PAI-Studio仅支持通过可视化方式，配置该组件参数。

页签	参数	描述
表选择	表名	读取的MaxCompute表名称。如果跨项目读取表数据，则表名需要使用**工程名.表名**格式，例如**tianchi_project.weibo_data**。 注意 如果MaxCompute表字段被修改，PAI-Studio算法平台不能自动同步，您需要手动重新配置MaxCompute源。
	分区	如果输入表为分区表，则系统自动选中**分区**复选框。
	参数	分区参数，仅支持单个分区。格式为**dt=@@{yyyyMMdd}**，其中**@@{yyyyMMdd}**表示当前日期，**@@{yyyyMMdd-1d}**表示当前日期的前一天。如果未配置该参数，则表示输入全表。
字段信息	源表字段信息	输入**表名**后，系统自动读取表的结构数据。

写数据表

写数据表组件将数据写入MaxCompute表，且不支持分区操作。PAI-Studio仅支持通过可视化方式，配置该组件参数。

参数	描述
新表名	写入数据的MaxCompute表名称。如果写入分区表，则需要先创建待写入的分区表，再使用该组件写入数据。
分区	写入表是否为分区表的开关。
设置生命周期	取值范围为整数。如果该参数值为空，则表示无生命周期。

MySQL数据同步

MySQL数据同步组件将MySQL数据同步至MaxCompute项目。PAI-Studio仅支持通过可视化方式，配置该组件参数。

参数	描述
实例名称	RDS的实例名称。您可以通过以下步骤查询该参数值： 1. 使用主账号登录RDS控制台。 2. 在左侧导航栏，单击**实例列表**。 3. 在**云数据库管理**页面的**基本信息**页签，查看**实例ID/名称**。
数据库	RDS数据库名称。您可以通过以下步骤查询该参数值： 1. 设置白名单，详情请参见设置RDS实例白名单。 2. 连接数据库，详情请参见登录RDS数据库。 3. 查看数据库的datebase、table及schema。
数据表	待同步的数据表。
用户名	RDS数据库的用户名。您可以通过以下步骤查询该参数值： 1. 在**云数据库管理**页面的**基本信息**页签，单击**实例ID/名称**。 2. 在实例详情页面的左侧导航栏，单击**账号管理**。 3. 在**用户账号**页签，查看数据库账号。
密码	RDS数据库密码。
同步的字段	默认同步该数据库的所有字段。
允许脏数据阈值	允许的数据错误数，默认值为0。
同步数据带宽	单位为MB/s，默认值为1 MB/s。

OSS数据同步

OSS数据同步组件将OSS文本同步至MaxCompute数据表。PAI-Studio仅支持通过可视化方式，配置该组件参数。

参数	描述
OSS Endpoint	OSS存储服务所在的Endpoint。
OSS AccessID	OSS服务的AccessID。
OSS AccessKey	OSS服务的AccessKey。
Bucket	OSS服务的Bucket。
Object	待同步的OSS Object。
OSS Column映射	同步的字段映射格式为**Index:Name**，表示将OSS的第**Index**列同步至MaxCompute的**Name**字段。多列采用逗号分隔，例如**0:label,1:s_width,2:s_length,3:v_width,4:v_length**。
OSS文本分隔符	OSS Object的文本分隔符，默认使用英文逗号（,）分隔。
OSS文本压缩格式	支持**无压缩**、**gzip**、**zip**及**bzip2**格式。
OSS文本编码	OSS文本的编码方式，仅支持**utf-8**。
同步数据带宽	单位为MB/s，默认值为1 MB/s。
允许脏数据阈值	允许的数据错误数，默认值为0。

> **说明**
> 您可以登录OSS控制台，查询OSS相关参数。

3.2. 数据预处理

3.2.1. 采样与过滤

本文为您介绍PAI-Studio提供的采样与过滤算法，包括随机采样、加权采样、过滤与映射和分层采样。

随机采样

以随机方式生成采样数据，每次采样是各自独立的。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
参数设置	**采样个数**	取值为正整数。
	采样比例	取值为浮点数，范围(0,1)。
	放回采样	默认为不放回，勾选后变为放回。
	随机数种子	默认系统自动生成。
执行调优	**核心数**	取值为正整数，默认系统自动分配。
	核内存分配	取值为正整数，范围(1, 65536)，默认系统自动分配。

- PAI命令方式

```
PAI -name WeightedSample
  -project algo_public \
  -Dlifecycle="28" \
  -DoutputTableName="test2" \
  -DprobCol="previous" \
  -Dreplace="false" \
  -DsampleSize="500" \
  -DinputPartitions="pt=20150501" \
  -DinputTableName="bank_data_partition";
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的名称	无
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	无
outputTableName	是	输出结果表	无

参数名称	是否必选	参数描述	默认值
sampleSize	否	采样个数 说明 当sampleSize与sampleRatio都为空时，系统会报错。 当sampleSize与sampleRatio都不为空时，以sampleSize为准。	无
sampleRatio	否	采样比例，浮点数，范围(0,1)。	无
replace	否	是否放回，BOOLEAN类型。	false
randomSeed	否	随机数种子，取值范围为正整数。	系统自动分配
lifecycle	否	输出表的生命周期，取值范围为[1,3650]。	无
coreNum	否	计算的核心数目，取值范围为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存（单位是MB），取值范围为(1, 65536)。	系统自动分配

加权采样

以加权方式生成采样数据。权重列必须为DOUBLE或BIGINT类型，按照该列值的大小采样。比如所选权重列的值是1.2和1.0，则值为1.2所属样本的被采样的概率就大一些。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
参数设置	**采样个数**	取值为正整数。
	采样比例	取值为浮点数，范围(0,1)。
	放回采样	默认为不放回，勾选后变为放回。
	权重列	下拉框选择加权列，加权列支持DOUBLE型和BIGINT型。每个值代表所在记录出现的权重，不需要归一化。
	随机数种子	默认系统自动生成。
执行调优	**核心数**	取值为正在数，默认系统自动分配。
	核内存分配	取值为正整数，范围(1, 65536)，默认系统自动分配。

- PAI命令方式

```
PAI -name WeightedSample
  -project algo_public \
  -Dlifecycle="28" \
  -DoutputTableName="test2" \
  -DprobCol="previous" \
  -Dreplace="false" \
  -DsampleSize="500" \
  -DinputPartitions="pt=20150501" \
  -DinputTableName="bank_data_partition";
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的名称	无
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
outputTableName	是	输出结果表	无
sampleSize	否	采样个数 **说明** ○ 当sampleSize与sampleRatio都为空时，系统会报错。 ○ 当sampleSize与sampleRatio都不为空时，以sampleSize为准。	无
sampleRatio	否	采样比例，浮点数，范围(0,1)。	无
probCol	是	要加权的列，每个值代表所在记录出现的权重，不需要归一化，支持DOUBLE型和BIGINT型。	无
replace	否	是否放回，BOOLEAN类型。	false（默认不放回）
randomSeed	否	随机数种子，取值范围为正整数。	系统自动分配
lifecycle	否	输出表的生命周期，取值范围为[1,3650]。	无
coreNum	否	计算的核心数目，取值范围为正整数。	系统自动分配

参数名称	是否必选	参数描述	默认值
memSizePerCore	否	每个核心的内存（单位是MB），取值范围为(1, 65536)。	系统自动分配

过滤与映射

对数据按照过滤表达式进行筛选，可以修改输出字段名称。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

参数	描述
选择字段	选择要筛选的列，默认选择全部列。也可以修改输出字段名称。
过滤条件	通过where条件实现数据过滤，与SQL类似，例如age>40。 **说明** 仅支持以下操作符： ○ = ○ != ○ > ○ < ○ >= ○ <= ○ like ○ rlike

- PAI命令方式

```
PAI -name Filter
  -project algo_public \
  -DoutTableName="test_9" \
  -DinputPartitions="pt=20150501" \
  -DinputTableName="bank_data_partition" \
  -Dfilter="age>=40";
```

参数名称	是否必选	参数描述
outputTableName	是	输出表的名称
inputPartitions	否	训练输入表分区。输入表对应的输入分区，选中全表则为None。
inputTableName	是	输入表的名称
filter	否	通过where条件实现数据过滤，与SQL类似，例如age>40。 **说明** 仅支持以下操作符： ○ = ○ != ○ > ○ < ○ >= ○ <= ○ like ○ rlike

分层采样

数据集分层抽取一定比例或者一定数据的随机样本。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**分组列**	选择分组列字段，分层按照此列划分。
参数设置	**采样个数**	取值为正整数。
	采样比例	取值为浮点数，范围(0,1)。
	随机数种子值	系统自动生成，默认值为1234567。
执行调优	**核心数**	取值为正在数，默认系统自动分配。
	每个核内存大小	取值为正整数，范围(1, 65536)，默认系统自动分配。

- PAI命令方式

```
PAI -name StratifiedSample
  -project algo_public \
  -DinputTableName="test_input" \
  -DoutputTableName="test_output" \
  -DstrataColName="label" \
  -DsampleSize="A:200,B:300,C:500" \
  -DrandomSeed=1007 \
  -Dlifecycle=30;
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的名称	无
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
outputTableName	是	输出结果表	无
strataColName	是	层次列，即按照此列作为key分层。	无
sampleSize	否	采样个数 ○ 正整数：表示每个层的采样个数。 ○ 字符串：格式为strata0:n0，strata1:n1，表示每个层分别配置的采样个数。 **说明** ○ 当sampleSize与sampleRatio都为空时，系统会报错。 ○ 当sampleSize与sampleRatio都不为空时，以sampleSize为准。	无
sampleRatio	否	采样比例 ○ 数字：范围(0,1)，表示每个stratum的采样比例。 ○ 字符串：格式为strata0:r0,strata1:r1，表示每个层分别配置采样比例。	无

参数名称	是否必选	参数描述	默认值
randomSeed	否	随机数种子，取值范围为正整数。	123456
lifecycle	否	输出表的生命周期，取值范围为[1,3650]。	无
coreNum	否	计算的核心数目，取值范围为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存（单位是MB），取值范围为(1, 65536)。	系统自动分配

3.2.2. 数据合并

本文为您介绍PAI-Studio提供的数据合并算法，包括Join、合并列和合并列（UNION）。

Join

两张表通过关联信息，合成一张表，并确定输出的字段，与SQL的Join语句功能类似。

PAI-Studio仅支持通过可视化方式进行数据合并。

参数	描述
连接类型	支持左连接、内连接、右连接和全连接。
关联条件	仅支持等式。可手动添加或删除关联条件。
选择左表输出字段列	左表输出字段列
选择右表输出字段列	右表输出字段列

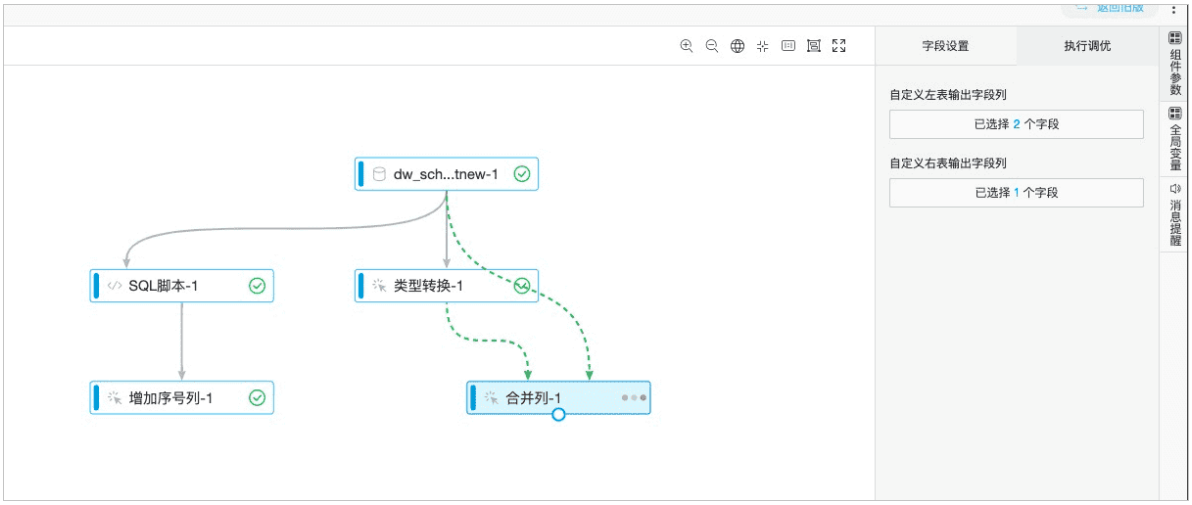
合并列

将两张表的数据按列合并，需要表的行数保持一致，否则报错。如果两张表只有一张存在分区，则分区表需要连接第二个输入端口。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

选取左表待合并的字段列，生成结果保存到右输入表的指定列。



- PAI命令方式

```
PAI -name appendColumns
  -project algo_public
  -DinputTableNames=maple_test_appendcol_basic_input1,maple_test_appendcol_basic_input2
  -DoutputTableName=maple_test_appendcol_setOutCol_output
  -DoutputTableColNames=x0,x1,x2,x3,x4,x5,x6,x7,x8,x9;
```

参数名称	是否必选	参数描述	默认值
inputTableNames	是	输入表的表名，两个表以逗号（,）分隔。	无
outputTableName	是	输出表的名称	无
selectedColNamesList	否	与输入表对应的已选中的列名列表： ◦ 同一个表的各列按逗号（,）分隔。 ◦ 不同表的各列按照分号（;）分隔。 **说明** 如果选择两张表所有的列，整个参数内容需用引号括起来，否则分号会被系统作为结束标志。如果某张表全选，可以省略所有列名，但必须保留相应分号。	无

参数名称	是否必选	参数描述	默认值
inputPartitionsInfoList	否	与输入表对应的已选择的partition列表： ◦ 同一个表的各partition按照正斜线（/）分隔。 ◦ 不同表的partition按照分号（;）分隔。 **说明** 如果选中所有partition，整个参数内容需要使用引号括起来，否则分号会被系统当做结束标志。如果某张表不用按照partition分割，则可省略所有partition名，但相应分号必须保留。	无
autoRenameCol	否	输出表各列是否自动命名。	false
outputTableColNames	否	输出表中各列的新列名，不填则输出原表中选择列的列名。 **注意** 如果autoRenameCol为Ture时，则该参数无意义。	无
lifecycle	否	输出表的生命周期，取值范围为正整数。	无
coreNum	否	节点个数，与参数memSizePerCore配对使用，正整数，取值范围[1, 9999]。	默认系统自动计算
memSizePerCore	否	单个节点内存大小，单位MB，正整数，取值范围[1024, 64*1024]。	默认系统自动计算

示例

- 源表

maple_test_appendcol_basic_input1

col0:bigint	col1:double	col2:string	col3:Datetime	col4:Boolean
10	0.0	aaaa	2015-10-01 00:00:00	TRUE
11	1.0	aaaa	2015-10-01 00:00:00	FALSE
12	2.0	aaaa	2015-10-01 00:00:00	TRUE
13	3.0	aaaa	2015-10-01 00:00:00	TRUE
14	4.0	aaaa	2015-10-01 00:00:00	TRUE

maple_test_appendcol_basic_input2

col10:bigint	col11:double	col12:string	col13:Datetime	col14:Boolean
110	10.0	2aaaa	2015-10-01 00:00:00	TRUE
111	11.0	2aaaa	2015-10-01 00:00:00	FALSE
112	12.0	2aaaa	2015-10-01 00:00:00	TRUE
113	13.0	2aaaa	2015-10-01 00:00:00	TRUE
114	14.0	2aaaa	2015-10-01 00:00:00	FALSE

- PAI命令行

```
PAI -name appendColumns
  -project algo_public
  -DinputTableNames=maple_test_appendcol_basic_input1,maple_test_appendcol_basic_input2
  -DoutputTableName=maple_test_appendcol_setOutCol_output
  -DoutputTableColNames=x0,x1,x2,x3,x4,x5,x6,x7,x8,x9;
```

- 输出表

maple_test_appendcol_setOutCol_output

x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
10	0	aaaa	2015-10-01 00:00:00	true	110	10	2aaaa	2015-10-01 00:00:00	true
11	1	aaaa	2015-10-01 00:00:00	false	111	11	2aaaa	2015-10-01 00:00:00	false
12	2	aaaa	2015-10-01 00:00:00	true	112	12	2aaaa	2015-10-01 00:00:00	true
13	3	aaaa	2015-10-01 00:00:00	true	113	13	2aaaa	2015-10-01 00:00:00	true
14	4	aaaa	2015-10-01 00:00:00	true	114	14	2aaaa	2015-10-01 00:00:00	false

合并行（UNION）

将两张表的数据按行合并，左表及右表选择输出的字段个数以及类型应保持一致。整合了union和union all的功能。

PAI-Studio仅支持通过可视化方式进行数据合并。

参数	描述
选择左表联合列	进行联合操作时，左右表选择的列数需相同，对应列的类型需保证一致。

参数	描述
输入左边表的where条件	通过where条件实现数据过滤，与SQL类似，例如age>40。 **说明** 仅支持以下操作符： • = • != • > • < • >= • <= • like • rlike
选择右表联合列	进行联合操作时，左右表选择的列数需相同，对应列的类型需保证一致。
输入右表的where条件	通过where条件实现数据过滤，与SQL类似，例如age>40。 **说明** 仅支持以下操作符： • = • != • > • < • >= • <= • like • rlike
去重	去重生成的数据表的重复行，系统默认勾选去重。

3.2.3. 增加序号列

本文为您介绍PAI-Studio提供的增加序号列组件。您可以在数据表的第一列追加ID列。

背景信息

支持1000000000*1023的算法规模。

增加序号列

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
参数设置	**默认全选**	默认全选，多余列不影响预测结果。
	序列号	默认为append_id。
	执行调优	**计算核心数**
		每个核内存数

- PAI命令方式

```
PAI -name AppendId
  -project algo_public
  -DinputTableName=maple_test_appendid_basic_input
  -DoutputTableName=maple_test_appendid_basic_output;
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无
selectedColNames	否	输入表中，参与训练的列。列名以英文逗号（,）分隔，支持INT和DOUBLE类型。如果输入为稀疏格式，则支持STRING类型的列。	所有列
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
outputTableName	是	输出结果表。	无
IDColName	否	ID列列名。	append_id
lifecycle	否	输出表生命周期。	无
coreNum	否	核心数量。	系统自动分配
memSizePerCore	否	单个核心使用的内存数。	系统自动分配

增加序列号示例

```
PAI -name AppendId
  -project algo_public
  -DinputTableName=maple_test_appendid_basic_input
  -DoutputTableName=maple_test_appendid_basic_output;
```

- 数据生成

col0	col1	col2	col3	col4
10	0.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true
11	1.0	aaaa	Thu Oct 01 00:00:00 CST 2015	false
12	2.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true
13	3.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true
14	4.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true

- 输出表

append_id	col0	col1	col2	col3	col4
0	10	0.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true
1	11	1.0	aaaa	Thu Oct 01 00:00:00 CST 2015	false
2	12	2.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true
3	13	3.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true
4	14	4.0	aaaa	Thu Oct 01 00:00:00 CST 2015	true

3.2.4. 拆分

本文为您介绍PAI-Studio提供的拆分算法。拆分是对数据进行随机拆分，用于生成训练和测试集。

拆分

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
参数设置	**拆分方式**	○ 按比例拆分 ○ 按阈值拆分
	切分比例	取值范围为(0,1)。
	随机数种子	系统默认生成。
	ID列（ID列相同的不会被拆分）	ID列。 **说明** 勾选**高级选项**时展示。
	阈值列	阈值所在列名，不支持String列。
	阈值	需要删除**切分比例**中的数据。
执行调优	**计算核心数**	系统根据输入数据量，自动分配训练的实例数量。
	每个核内存数	系统根据输入数据量，自动分配内存。单位为MB。

- PAI命令方式

```
PAI -name split -project algo_public
  -DinputTableName=wbpc
  -Doutput1TableName=wpbc_split1
  -Doutput2TableName=wpbc_split2
  -Dfraction=0.25;
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无

参数名称	是否必选	参数描述	默认值
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
output1TableName	是	输出结果表1。	无
output1TablePartition	否	输出结果表1分区名。	输出表1为非分区表
output2TableName	是	输出结果表2。	无
output2TablePartition	否	输出结果表2分区名。	输出表2为非分区表
fraction	否	切分至输出表1的数据比例，取值范围为(0,1)。	无
randomSeed	否	随机数种子，取值范围为正整数。	系统自动分配
idColName	否	ID列（ID相同的数据不会被拆分）	无
thresholdColName	否	阈值所在列名，不支持String列。	无
threshold	否	阈值。	无
lifecycle	否	输出表的生命周期，取值范围为[1,3650]。	无
coreNum	否	核心数量。	系统自动分配
memSizePerCore	否	每个核心的内存（单位是兆），取值范围为(1,65536)。	系统自动分配

3.2.5. 缺失值填充

您可以通过给定一个缺失值的配置列表，来实现将输入表的缺失值用指定的值来填充。

背景信息

- 将数值型的空值替换为最大值，最小值，均值或者一个自定义的值。

- 将字符型的空值，空字符串，空值和空字符串，指定值替换为一个自定义的值。
- 待填充的缺失值可以选择空值或空字符，也可以自定义。

  缺失值如果选择空字符串，则填充的目标列应是STRING型。
- 数值型替换可以自定义，也可以直接选择替换成数值最大值，最小值或者均值。

缺失值填充

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
参数设置	**填充的字段**	默认全选，多余列不影响预测结果。
	原值	○ Null（数值和string） ○ 空字符串（string） ○ Null和空字符串（string） ○ 自定义（string）
	替换为	○ Min（数值型） ○ Max（数值型） ○ Mean（数值型） ○ 自定义（数值型和string）
	configs	ID列。 **说明** 勾选**高级选项**时展示。
	执行调优	**计算核心数**
		每个核内存数

- PAI命令方式

```
PAI -name FillMissingValues
  -project algo_public
  -Dconfigs="poutcome,null-empty,testing" \
  -DoutputTableName="test_3"
  -DinputPartitions="pt=20150501"
  -DinputTableName="bank_data_partition";
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无

参数名称	是否必选	参数描述	默认值
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
outputTableName	是	输出结果表。	无
configs	是	缺失值填充的配置。 例如格式 `col1, null, 3.14; col2, empty, hello; col3, empty-null, world`，其中 `null` 表示空值，`empty` 表示空字符。 ○ 如果选择空字符，则填充的目标列应是STRING型。 ○ 如果采用最大值、最小值、均值，可以采用变量，其命名规范形如：min, max, mean。 ○ 如果用户自定义替换值，则使用user-defined，格式例如 `col4,user-defined,str, str123`。	无
outputParaTableName	否	配置输出表。	输出表1为非分区表
inputParaTableName	是	配置输入表。	无
lifecycle	否	输出表的生命周期，取值范围为[1,3650]。	无
coreNum	否	计算的核心数目，取值为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存（单位是兆），取值范围为(1, 65536)。	系统自动分配

缺失值填充示例

1. 使用SQL语句，生成测试数据。

```
drop table if exists fill_missing_values_test_input;
create table fill_missing_values_test_input(
  col_string string,
  col_bigint bigint,
  col_double double,
  col_boolean boolean,
  col_datetime datetime);
insert overwrite table fill_missing_values_test_input
select
  *
from
(
  select
    '01' as col_string,
    10 as col_bigint,
    10.1 as col_double,
    True as col_boolean,
    cast('2016-07-01 10:00:00' as datetime) as col_datetime
  from dual
  union all
    select
      cast(null as string) as col_string,
      11 as col_bigint,
      10.2 as col_double,
      False as col_boolean,
      cast('2016-07-02 10:00:00' as datetime) as col_datetime
    from dual
  union all
    select
      '02' as col_string,
      cast(null as bigint) as col_bigint,
      10.3 as col_double,
      True as col_boolean,
      cast('2016-07-03 10:00:00' as datetime) as col_datetime
    from dual
  union all
    select
      '03' as col_string,
      12 as col_bigint,
      cast(null as double) as col_double,
      False as col_boolean,
      cast('2016-07-04 10:00:00' as datetime) as col_datetime
    from dual
  union all
    select
      '04' as col_string,
      13 as col_bigint,
      10.4 as col_double,
      cast(null as boolean) as col_boolean,
      cast('2016-07-05 10:00:00' as datetime) as col_datetime
    from dual
```

```
    from dual
  union all
    select
      '05' as col_string,
      14 as col_bigint,
      10.5 as col_double,
      True as col_boolean,
      cast(null as datetime) as col_datetime
    from dual
) tmp;
```

输入数据说明。

```
+------------+------------+------------+-------------+--------------+
| col_string | col_bigint | col_double | col_boolean | col_datetime |
+------------+------------+------------+-------------+--------------+
| 04         | 13         | 10.4       | NULL        | 2016-07-05 10:00:00 |
| 02         | NULL       | 10.3       | true        | 2016-07-03 10:00:00 |
| 03         | 12         | NULL       | false       | 2016-07-04 10:00:00 |
| NULL       | 11         | 10.2       | false       | 2016-07-02 10:00:00 |
| 01         | 10         | 10.1       | true        | 2016-07-01 10:00:00 |
| 05         | 14         | 10.5       | true        | NULL         |
+------------+------------+------------+-------------+--------------+
```

2. 运行命令。

```
drop table if exists fill_missing_values_test_input_output;
drop table if exists fill_missing_values_test_input_model_output;
PAI -name FillMissingValues
-project algo_public
-Dconfigs="col_double,null,mean;col_string,null-empty,str_type_empty;col_bigint,null,max;col_boole
an,null,true;col_datetime,null,2016-07-06 10:00:00"
-DoutputParaTableName="fill_missing_values_test_input_model_output"
-Dlifecycle="28"
-DoutputTableName="fill_missing_values_test_input_output"
-DinputTableName="fill_missing_values_test_input";
drop table if exists fill_missing_values_test_input_output_using_model;
drop table if exists fill_missing_values_test_input_output_using_model_model_output;
PAI -name FillMissingValues
-project algo_public
-DoutputParaTableName="fill_missing_values_test_input_output_using_model_model_output"
-DinputParaTableName="fill_missing_values_test_input_model_output"
-Dlifecycle="28"
-DoutputTableName="fill_missing_values_test_input_output_using_model"
-DinputTableName="fill_missing_values_test_input";
```

3. 运行结果。
   - fill_missing_values_test_input_output

```
+------------+------------+------------+-------------+--------------+
| col_string | col_bigint | col_double | col_boolean | col_datetime |
+------------+------------+------------+-------------+--------------+
| 04         | 13         | 10.4       | true        | 2016-07-05 10:00:00 |
| 02         | 14         | 10.3       | true        | 2016-07-03 10:00:00 |
| 03         | 12         | 10.3       | false       | 2016-07-04 10:00:00 |
| str_type_empty | 11         | 10.2       | false       | 2016-07-02 10:00:00 |
| 01         | 10         | 10.1       | true        | 2016-07-01 10:00:00 |
| 05         | 14         | 10.5       | true        | 2016-07-06 10:00:00 |
+------------+------------+------------+-------------+--------------+
```

- fill_missing_values_test_input_model_output

```
+------------+------------+
| feature    | json       |
+------------+------------+
| col_string | {"name": "fillMissingValues", "type": "string", "paras":{"missing_value_type": "null-empty", "replaced_value": "str_type_empty"}} |
| col_bigint | {"name": "fillMissingValues", "type": "bigint", "paras":{"missing_value_type": "null", "replaced_value": 14}} |
| col_double | {"name": "fillMissingValues", "type": "double", "paras":{"missing_value_type": "null", "replaced_value": 10.3}} |
| col_boolean | {"name": "fillMissingValues", "type": "boolean", "paras":{"missing_value_type": "null", "replaced_value": 1}} |
| col_datetime | {"name": "fillMissingValues", "type": "datetime", "paras":{"missing_value_type": "null", "replaced_value": 1467770400000}} |
+------------+------------+
```

- fill_missing_values_test_input_output_using_model

```
+------------+------------+------------+-------------+--------------+
| col_string | col_bigint | col_double | col_boolean | col_datetime |
+------------+------------+------------+-------------+--------------+
| 04         | 13         | 10.4       | true        | 2016-07-05 10:00:00 |
| 02         | 14         | 10.3       | true        | 2016-07-03 10:00:00 |
| 03         | 12         | 10.3       | false       | 2016-07-04 10:00:00 |
| str_type_empty | 11         | 10.2       | false       | 2016-07-02 10:00:00 |
| 01         | 10         | 10.1       | true        | 2016-07-01 10:00:00 |
| 05         | 14         | 10.5       | true        | 2016-07-06 10:00:00 |
+------------+------------+------------+-------------+--------------+
```

- fill_missing_values_test_input_output_using_model_model_output

```
+------------+------------+
| feature    | json       |
+------------+------------+
| col_string | {"name": "fillMissingValues", "type": "string", "paras":{"missing_value_type": "null-empty", "replaced_value": "str_type_empty"}} |
| col_bigint | {"name": "fillMissingValues", "type": "bigint", "paras":{"missing_value_type": "null", "replaced_value": 14}} |
| col_double | {"name": "fillMissingValues", "type": "double", "paras":{"missing_value_type": "null", "replaced_value": 10.3}} |
| col_boolean | {"name": "fillMissingValues", "type": "boolean", "paras":{"missing_value_type": "null", "replaced_value": 1}} |
| col_datetime | {"name": "fillMissingValues", "type": "datetime", "paras":{"missing_value_type": "null", "replaced_value": 1467770400000}} |
+------------+------------+
```

3.2.6. 归一化

本文为您介绍PAI-Studio提供的归一化组件。

归一化

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**默认全选**	默认全选，多余列不影响预测结果。
	保留原始列	处理过的列增加“stdized_”前缀。支持DOUBLE类型与BIGINT类型。
执行调优	**计算核心数**	系统根据输入数据量，自动分配训练的实例数量。
	每个核内存	系统根据输入数据量，自动分配内存。单位为MB。

- PAI命令方式
  - 稠密数据的命令

```
PAI -name Normalize
  -project algo_public
  -DkeepOriginal="true"
  -DoutputTableName="test_4"
  -DinputTablePartitions="pt=20150501"
  -DinputTableName="bank_data_partition"
  -DselectedColNames="emp_var_rate,euribor3m"
```

- 稀疏数据的命令

```
PAI -name Normalize
  -project projectxlib4
  -DkeepOriginal="true"
  -DoutputTableName="kv_norm_output"
  -DinputTableName=kv_norm_test
  -DselectedColNames="f0,f1,f2"
  -DenableSparse=true
  -DoutputParaTableName=kv_norm_model
  -DkvIndices=1,2,8,6
  -DitemDelimiter=",";
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无
selectedColNames	否	输入表中，参与训练的列。列名以英文逗号（,）分隔，支持INT和DOUBLE类型。如果输入为稀疏格式，则支持STRING类型的列。	所有列
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
outputTableName	是	输出结果表。	无
outputParaTableName	否	配置输出表。	输出表1为非分区表
inputParaTableName	是	配置输入表。	无
keepOriginal	否	是否保留原始列： ○ true：处理过的列重命名（”normalized_”前缀），原始列保留。 ○ false：全部列保留且不重命名。	false

参数名称	是否必选	参数描述	默认值
lifecycle	否	输出表的生命周期，取值范围为[1,3650]。	无
coreNum	否	计算的核心数目，取值为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存（单位是兆），取值范围为(1, 65536)。	系统自动分配
enableSparse	否	是否打开稀疏支持： ○ true ○ false	false
itemDelimiter	否	KV对之间分隔符。	默认”,”
kvDelimiter	否	Key和Value之间分隔符。	默认”:”
kvIndices	否	KV表中需要归一化的特征索引。	无

归一化实例

- 数据生成

```
drop table if exists normalize_test_input;
create table normalize_test_input(
  col_string string,
  col_bigint bigint,
  col_double double,
  col_boolean boolean,
  col_datetime datetime);
insert overwrite table normalize_test_input
select
  *
from
(
  select
    '01' as col_string,
    10 as col_bigint,
    10.1 as col_double,
    True as col_boolean,
    cast('2016-07-01 10:00:00' as datetime) as col_datetime
  from dual
  union all
    select
      cast(null as string) as col_string,
      11 as col_bigint,
      10.2 as col_double,
      False as col_boolean,
      cast('2016-07-02 10:00:00' as datetime) as col_datetime
    from dual
```

```
    from dual
  union all
    select
      '02' as col_string,
      cast(null as bigint) as col_bigint,
      10.3 as col_double,
      True as col_boolean,
      cast('2016-07-03 10:00:00' as datetime) as col_datetime
    from dual
  union all
    select
      '03' as col_string,
      12 as col_bigint,
      cast(null as double) as col_double,
      False as col_boolean,
      cast('2016-07-04 10:00:00' as datetime) as col_datetime
    from dual
  union all
    select
      '04' as col_string,
      13 as col_bigint,
      10.4 as col_double,
      cast(null as boolean) as col_boolean,
      cast('2016-07-05 10:00:00' as datetime) as col_datetime
    from dual
  union all
    select
      '05' as col_string,
      14 as col_bigint,
      10.5 as col_double,
      True as col_boolean,
      cast(null as datetime) as col_datetime
    from dual
) tmp;
```

- PAI命令行

```
drop table if exists normalize_test_input_output;
drop table if exists normalize_test_input_model_output;
PAI -name Normalize
  -project algo_public
  -DoutputParaTableName="normalize_test_input_model_output"
  -Dlifecycle="28"
  -DoutputTableName="normalize_test_input_output"
  -DinputTableName="normalize_test_input"
  -DselectedColNames="col_double,col_bigint"
  -DkeepOriginal="true";
drop table if exists normalize_test_input_output_using_model;
drop table if exists normalize_test_input_output_using_model_model_output;
PAI -name Normalize
  -project algo_public
  -DoutputParaTableName="normalize_test_input_output_using_model_model_output"
  -DinputParaTableName="normalize_test_input_model_output"
  -Dlifecycle="28"
  -DoutputTableName="normalize_test_input_output_using_model"
  -DinputTableName="normalize_test_input";
```

- 输入说明

  normalize_test_input

col_string	col_bigint	col_double	col_boolean	col_datetime
01	10	10.1	true	2016-07-01 10:00:00
NULL	11	10.2	false	2016-07-02 10:00:00
02	NULL	10.3	true	2016-07-03 10:00:00
03	12	NULL	false	2016-07-04 10:00:00
04	13	10.4	NULL	2016-07-05 10:00:00
05	14	10.5	true	NULL

- 输出说明

- normalize_test_input_output

col_string	col_bigint	col_double	col_boolean	col_datetime	normalized_col_bigint	normalized_col_double
01	10	10.1	true	2016-07-01 10:00:00	0.0	0.0
NULL	11	10.2	false	2016-07-02 10:00:00	0.25	0.2499999999999989
02	NULL	10.3	true	2016-07-03 10:00:00	NULL	0.5000000000000022
03	12	NULL	false	2016-07-04 10:00:00	0.5	NULL
04	13	10.4	NULL	2016-07-05 10:00:00	0.75	0.7500000000000011
05	14	10.5	true	NULL	1.0	1.0

- normalize_test_input_model_output

feature	json
col_bigint	{ “name” : “normalize” , “type” :” bigint” , “paras” :{ “min” :10, “max” : 14}}
col_double	{ “name” : “normalize” , “type” :” double” , “paras” :{ “min” :10.1, “max” : 10.5}}

- normalize_test_input_output_using_model

col_string	col_bigint	col_double	col_boolean	col_datetime
01	0.0	0.0	true	2016-07-01 10:00:00
NULL	0.25	0.2499999999999989	false	2016-07-02 10:00:00
02	NULL	0.5000000000000022	true	2016-07-03 10:00:00
03	0.5	NULL	false	2016-07-04 10:00:00
04	0.75	0.7500000000000011	NULL	2016-07-05 10:00:00
05	1.0	1.0	true	NULL

- normalize_test_input_output_using_model_model_output

feature	json
col_bigint	{“name”：“normalize”，“type”:” bigint”，“paras”:{“min”:10，“max”：14}}
col_double	{“name”：“normalize”，“type”:” double”，“paras”:{“min”:10.1，“max”：10.5}}

3.2.7. 标准化

本文为您介绍PAI-Studio提供的标准化组件。

背景信息

- 对一个表的某一列或多列，进行标准化处理，将产生的数据存入新表中。
- 标准化所使用的公式：（X - Mean）/（standard deviation）。
  - Mean：样本平均值。
  - standard deviation：样本标准偏差，针对从总体抽样，利用样本来计算总体偏差，为了使算出的值与总体水平更接近，就必须将算出的标准偏差的值适度放大，即$\frac{1}{N-1}$。
  - 样本标准偏差公式：$S=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i-\bar{X})^2}$。

    其中$\bar{X}$代表所采用的样本X1，X2，...，Xn的均值。

标准化

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**默认全选**	默认全选，多余列不影响预测结果。
	保留原始列	处理过的列增加“stdized_”前缀。支持DOUBLE类型与BIGINT类型。
执行调优	**计算核心数**	系统根据输入数据量，自动分配训练的实例数量。
	每个核内存数	系统根据输入数据量，自动分配内存。单位为MB。

- PAI命令方式

- 稠密数据的命令

```
PAI -name Standardize
  -project algo_public
  -DkeepOriginal="false"
  -DoutputTableName="test_5"
  -DinputTablePartitions="pt=20150501"
  -DinputTableName="bank_data_partition"
  -DselectedColNames="euribor3m,pdays"
```

- 稀疏数据的命令

```
PAI -name Standardize
  -project projectxlib4
  -DkeepOriginal="true"
  -DoutputTableName="kv_standard_output"
  -DinputTableName=kv_standard_test
  -DselectedColNames="f0,f1,f2"
  -DenableSparse=true
  -DoutputParaTableName=kv_standard_model
  -DkvIndices=1,2,8,6
  -DitemDelimiter=",";
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无
selectedColNames	否	输入表中，参与训练的列。列名以英文逗号（,）分隔，支持INT和DOUBLE类型。如果输入为稀疏格式，则支持STRING类型的列。	所有列
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： - Partition_name=value - name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
outputTableName	是	输出结果表。	无
outputParaTableName	是	配置输出表。	无
inputParaTableName	否	配置输入表。	无

参数名称	是否必选	参数描述	默认值
keepOriginal	否	是否保留原始列： ○ true：处理过的列重命名（“stdized_”前缀），原始列保留。 ○ false：全部列保留且不重命名。	false
lifecycle	否	输出表生命周期。	无
coreNum	否	核心数量。	系统自动分配
memSizePerCore	否	单个核心使用的内存数。	系统自动分配
enableSparse	否	是否打开稀疏支持： ○ true ○ false	false
itemDelimiter	否	KV对之间分隔符。	默认”,”
kvDelimiter	否	Key和Value之间分隔符。	默认”:”
kvIndices	否	KV表中需要归一化的特征索引。	无

标准化示例

详细示例

```
drop table if exists standardize_test_input;
create table standardize_test_input(
  col_string string,
  col_bigint bigint,
  col_double double,
  col_boolean boolean,
  col_datetime datetime);
insert overwrite table standardize_test_input
select
  *
from
(
  select
    '01' as col_string,
    10 as col_bigint,
    10.1 as col_double,
    True as col_boolean,
    cast('2016-07-01 10:00:00' as datetime) as col_datetime
  from dual
  union all
    select
      cast(null as string) as col_string,
      11 as col_bigint,
```

```
        10.2 as col_double,
        False as col_boolean,
        cast('2016-07-02 10:00:00' as datetime) as col_datetime
      from dual
    union all
      select
        '02' as col_string,
        cast(null as bigint) as col_bigint,
        10.3 as col_double,
        True as col_boolean,
        cast('2016-07-03 10:00:00' as datetime) as col_datetime
      from dual
    union all
      select
        '03' as col_string,
        12 as col_bigint,
        cast(null as double) as col_double,
        False as col_boolean,
        cast('2016-07-04 10:00:00' as datetime) as col_datetime
      from dual
    union all
      select
        '04' as col_string,
        13 as col_bigint,
        10.4 as col_double,
        cast(null as boolean) as col_boolean,
        cast('2016-07-05 10:00:00' as datetime) as col_datetime
      from dual
    union all
      select
        '05' as col_string,
        14 as col_bigint,
        10.5 as col_double,
        True as col_boolean,
        cast(null as datetime) as col_datetime
      from dual
) tmp;
```

- PAI命令行

```
drop table if exists standardize_test_input_output;
drop table if exists standardize_test_input_model_output;
PAI -name Standardize
  -project algo_public
  -DoutputParaTableName="standardize_test_input_model_output"
  -Dlifecycle="28"
  -DoutputTableName="standardize_test_input_output"
  -DinputTableName="standardize_test_input"
  -DselectedColNames="col_double,col_bigint"
  -DkeepOriginal="true";
drop table if exists standardize_test_input_output_using_model;
drop table if exists standardize_test_input_output_using_model_model_output;
PAI -name Standardize
  -project algo_public
  -DoutputParaTableName="standardize_test_input_output_using_model_model_output"
  -DinputParaTableName="standardize_test_input_model_output"
  -Dlifecycle="28"
  -DoutputTableName="standardize_test_input_output_using_model"
  -DinputTableName="standardize_test_input";
```

- 输入说明

  standardize_test_input

col_string	col_bigint	col_double	col_boolean	col_datetime
01	10	10.1	true	2016-07-01 10:00:00
NULL	11	10.2	false	2016-07-02 10:00:00
02	NULL	10.3	true	2016-07-03 10:00:00
03	12	NULL	false	2016-07-04 10:00:00
04	13	10.4	NULL	2016-07-05 10:00:00
05	14	10.5	true	NULL

- 输出说明

- standardize_test_input_output

col_string	col_bigint	col_double	col_boolean	col_datetime	stdized_col_bigint	stdized_col_double
01	10	10.1	true	2016-07-01 10:00:00	-1.2649110640673518	-1.2649110640683832
NULL	11	10.2	false	2016-07-02 10:00:00	-0.6324555320336759	-0.6324555320341972
02	NULL	10.3	true	2016-07-03 10:00:00	NULL	0.0
03	12	NULL	false	2016-07-04 10:00:00	0.0	NULL
04	13	10.4	NULL	2016-07-05 10:00:00	0.6324555320336759	0.6324555320341859
05	14	10.5	true	NULL	1.2649110640673518	1.2649110640683718

- standardize_test_input_model_output

feature	json
col_bigint	{“name”: “standardize”, “type”:” bigint”, “paras”:{“mean”:12, “std”: 1.58113883008419}}
col_double	{“name”: “standardize”, “type”:” double”, “paras”:{“mean”:10.3, “std”: 0.1581138830082909}}

- standardize_test_input_output_using_model

col_string	col_bigint	col_double	col_boolean	col_datetime
01	-1.2649110640673515	-1.264911064068383	true	2016-07-01 10:00:00
NULL	-0.6324555320336758	-0.6324555320341971	false	2016-07-02 10:00:00
02	NULL	0.0	true	2016-07-03 10:00:00
03	0.0	NULL	false	2016-07-04 10:00:00
04	0.6324555320336758	0.6324555320341858	NULL	2016-07-05 10:00:00
05	1.2649110640673515	1.2649110640683716	true	NULL

- standardize_test_input_output_using_model_model_output

feature	json
col_bigint	{“name”：“standardize”，“type”:” bigint”，“paras”:{“mean”:12，“std”：1.58113883008419}}
col_double	{“name”：“standardize”，“type”:” double”，“paras”:{“mean”:10.3，“std”：0.1581138830082909}}

3.2.8. 类型转换

本文为您介绍PAI-Studio提供的类型转换组件。您可以将任意类型特征转成STRING、DOUBLE和INT特征，并支持转换异常时的缺失值填充。

背景信息

- 支持将表的字段类型转成另一个类型。
- 支持多个字段同时转换成不同的类型。
- 可以选择是否保持原来的转换前的数据列。

类型转换

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	描述	描述
字段设置	**转换为double类型的列**	转换所选字段为DOUBLE类型。
	转换为double异常时默认填充值	转换为DOUBLE类型异常时的默认填充值。
	转换为int类型的列	转换所选字段为INT类型。
	转换为int异常时默认填充值	转换为DOUBLE类型异常时的默认填充值。
	转换为string类型的列	转换所选字段为STRING类型。
	转换为string异常时默认填充值	转换为STRING类型异常时的默认填充值。
	是否保留原列	列名增加前缀“typed_”。
	单个节点内存大小	取值范围为1024 MB~64*1024 MB。
	节点个数	与参数**单个节点内存大小**搭配使用，取值范围为[1, 9999]。

- PAI命令方式

```
pai -project algo_public
  -name type_transform_v1
  -DinputTable=type_test
  -Dcols_to_string="f0"
  -Ddefault_double_value=0.0
  -DoutputTable=type_test_output;
```

参数名称	是否必选	参数描述	默认值
inputTable	是	输入表的表名。	无
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
outputTable	是	类型转换的结果表。	无

参数名称	是否必选	参数描述	默认值
reserveOldFeat	否	是否保持原来变换前的数据列。	无
cols_to_double	否	需要类型转换到DOUBLE的特征列。	无
cols_to_string	否	需要类型转换到STRING的特征列。	无
cols_to_int	否	需要类型转换到INT的特征列。	无
default_int_value	否	当特征字段为空时的值。	0
default_double_value	否	当特征字段为空时的值。	0.0
default_string_value	否	当特征字段为空时的值。	“”
coreNum	否	节点数量。与memSizePerCore搭配使用，取值范围为[1, 9999]。	默认自动计算
memSizePerCore	否	单个结点内存大小，单位M，取值范围为[1024, 64*1024]。	默认自动计算
lifecycle	否	输出表生命周期。	7

类型转换示例

- 测试数据生成

```
create table transform_test as
select * from
(
select true as f0,2.0 as f1,1 as f2 from dual union all
select false as f0,3.0 as f1,1 as f2 from dual union all
select false as f0,4.0 as f1,1 as f2 from dual union all
select true as f0,3.0 as f1,1 as f2 from dual union all
select false as f0,3.0 as f1,1 as f2 from dual union all
select false as f0,4.0 as f1,1 as f2 from dual union all
select true as f0,3.0 as f1,1 as f2 from dual union all
select false as f0,5.0 as f1,1 as f2 from dual union all
select false as f0,3.0 as f1,1 as f2 from dual union all
select true as f0,4.0 as f1,1 as f2 from dual union all
select false as f0,3.0 as f1,1 as f2 from dual union all
select true as f0,4.0 as f1,1 as f2 from dual
)tmp;
```

- 训练数据展示

f0	f1	f2
false	3.0	1
false	3.0	1
true	2.0	1
true	4.0	1
false	4.0	1
false	3.0	1
false	3.0	1
true	3.0	1
false	4.0	1
true	4.0	1
false	5.0	1
true	3.0	1

- 训练PAI命令

```
pai -project projectxlib4
  -name type_transform_v1
  -DinputTable=transform_test
  -Dcols_to_double=f0
  -Dcols_to_int=f1
  -Dcols_to_string=f2
  -DoutputTable=trans_test_output;
```

- 输出说明

  结果表

f0	f1	f2
0.0	3	1
0.0	3	1
1.0	2	1
1.0	4	1
0.0	4	1
0.0	3	1

f0	f1	f2
1.0	3	1
0.0	4	1
0.0	3	1
0.0	5	1
1.0	3	1
1.0	4	1

3.2.9. KV2Table

您可以使用KV2Table转化KV（Key:Value）格式的表为普通表格式。Key转换成表的某列名，Value转成该列在对应行的值。

背景信息

KV表格式定义：Key是列名的index，Value支持BIGINT，DOUBLE和STRING类型。在该组件中可以输入用户定义的key_map表，是列名和Key的映射，但无论是否输入key_map表，该组件都会输出key_map表记录转化后的列名和Key的映射。例如1:10，2:20和3:30。

key_map表格式定义：包含列名和index的映射以及类型信息的col_name，col_index和col_datatype，这三列类型要求是STRING，当col_datatype缺失时，默认值为double类型。

col_name	col_index	col_datatype
col1	1	bigint
col2	2	double

KV2Table

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**KV列名**	KV列名。
	附加列名	附加列名。
	key和value之间分隔符	Key和Value之间分隔符。默认”:”。
	kv对之间分隔符	KV对之间分隔符。默认”,”
参数设置	**是否只截取前1200列**	如果转化后列数超过表最大列数1200列后，是否只截取前1200列。

页签	参数	描述
执行调优	**计算核心数**	系统根据输入数据量，自动分配训练的实例数量。
	每个核内存大小	系统根据输入数据量，自动分配内存。单位为MB。

- PAI命令方式

```
PAI -name KVToTable
  -project algo_public
  -DinputTableName=test
  -DoutputTableName=test_out
  -DoutputKeyMapTableName=test_keymap_out
  -DkvColName=kv;
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无
kvColName	是	KV列名。	无
outputTableName	是	输出结果表。	无
outputKeyMapTableName	是	输出索引表名。	无
inputKeyMapTableName	否	输入索引表名。	无
appendColName	否	附加列名。	无
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
kvDelimiter	否	Key和Value之间分隔符。	默认”:”
itemDelimiter	否	KV对之间分隔符。	默认”,”
top1200	否	是否只截取前1200列： ○ true ○ false	true

参数名称	是否必选	参数描述	默认值
lifecycle	否	输出表的生命周期。	无
coreNum	否	计算的核心数目，取值为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存（单位是兆），取值范围为(100,64*1024)。	系统自动分配

KV2Table示例

- 数据生成

```
drop table if exists test;
  create table test as
   select
    *
   from
   (
    select '1:1,2:2,3:-3.3' as kv from dual
    union all
    select '1:10,2:20,3:-33.3' as kv from dual
   ) tmp;
```

- PAI命令行

```
PAI -name KVToTable
  -project algo_public
  -DinputTableName=test
  -DoutputTableName=test_out
  -DoutputKeyMapTableName=test_keymap_out
  -DkvColName=kv;
```

- 输出说明
  - 输出表

```
+------------+------------+------------+
| kv_1       | kv_2       | kv_3       |
+------------+------------+------------+
| 1.0        | 2.0        | -3.3       |
| 10.0       | 20.0       | -33.3      |
+------------+------------+------------+
```

  - 输出映射表

```
+------------+------------+------------+
| col_name   | col_index  | col_type   |
+------------+------------+------------+
| kv_1       | 1          | double     |
| kv_2       | 2          | double     |
| kv_3       | 3          | double     |
+------------+------------+------------+
```

算法规模

转化后的列包含Append列和KV转化的列，先输出KV列再输出Append列。当总列数超过最大列数限制，且输出top1200选项为True时，则输出最大列数，否则报错，目前输出表的最大列数为1200列。

数据量不超过1亿条记录。

常见问题

- Q：如果有输入key_map表，则转化的列内容是什么？

  A：转化的列是key_map表中的Key和KV表中的Key的交集。

- Q：如果有输入key_map表，则转化后的Key列类型是什么？

  A：转化后key列类型和key_map表中一致。如果key_map表无类型，则转化后key列类型为DOUBLE。

- Q：如果有输入key_map表，则转化后key列名称的命名规则是什么？

  A：命名规则为kv列的列名+“”+key。

  不支持以下字符：

  %&()*+-./;<>=?

- Q：列名冲突原因是什么？

  A：如果指定了Append列，且Append列名和转化后Key列名相同，则会报错。

- Q：转化的列支持什么类型？

  A：只支持数值类型。

- Q：列名长度超过128个字符时怎么办？

  A：列名会被截断成128个字符。

- Q：同一行有重复Key时，如何处理？

  A：需要将Value值相加。

3.2.10. Table2KV

本文为您介绍如何使用Table2KV转化普通表为KV（Key:Value）格式的表。

使用限制

- 转换后的结果表不会显示原表中的空值。您可以在结果表中指定需要保留的列，并且输出的列与原表的列一致。
- 如果存在输入Key_map表，则转化的列为Key_map表与KV表中Key的交集。
- 如果存在的输入Key_map表与输入表类型冲突，则输出的Key_map表使用您指定的类型。
- 输入表中需要转换为KV的列只能为BIGINT或DOUBLE类型。

Table2KV

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数，如下所示：

- 可视化方式

页签	参数	描述
字段设置	**转换列**	需要转换的列名称。
	保持原样列	不需要转换的列名称。
	key和value的分割符	Key和Value之间的分割符。默认为半角冒号（:）。
	kv间的分割符	KV对之间的分割符。默认为半角逗号（,）。
参数设置	**指定是否将列转换为编号**	指定是否将列转换为编号，取值如下： ○ 转换 ○ 不转换
执行调优	**核心数**	系统根据输入数据量，自动分配训练的实例数量。
	内存数	系统根据输入数据量，自动分配内存。单位为MB。

- PAI命令方式

```
PAI -name TableToKV
  -project algo_public
  -DinputTableName=maple_tabletokv_basic_input
  -DoutputTableName=maple_tabletokv_basic_output
  -DselectedColNames=col0,col1,col2
  -DappendColNames=rowid;
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名称。	无
inputTablePartitions	否	输入表中指定参与训练的分区，格式为 `Partition_name=value` 。 如果是多级分区，格式为 `name1=value1/name2=value2;` 。 如果指定多个分区，则需要使用半角逗号（,）分隔。	默认选择所有分区。
selectedColNames	否	选择的列名称，只能为BIGINT或DOUBLE类型。	默认选择整张表。
appendColNames	否	需要保留的列名称，该列会被原样写入至输出表中。	无
outputTableName	是	输出的KV表名称。	无

参数名称	是否必选	参数描述	默认值
kvDelimiter	否	Key和Value的分割符。	半角冒号（:）
itemDelimiter	否	KV间的分割符。默认为半角逗号（,）。	半角逗号（,）
convertColToIndexId	否	指定是否将列转换为编号。取值如下： ○ *1*为转换。 ○ *0*为不转换。	0
inputKeyMapTableName	否	输入的索引表名称。 该参数仅当 convertColToIndexId=1 时有效。如果未指定该参数，则程序自动计算一套编号。	半角引号（“”）
outputKeyMapTableName	由convertColToIndexId决定。	输出的索引表名称。当且仅当convertColToIndexId=1时该参数必选。	无
lifecycle	否	输出表的生命周期。取值为正整数。	无
coreNum	否	节点个数。取值范围为*[1, 9999]*的正整数。与memSizePerCore参数配对使用。	系统自动分配。
memSizePerCore	否	单个节点的内存大小，单位为兆。取值范围为*[1024, 64*1024]*的正整数。	系统自动分配。

示例一

- 数据生成

rowid	kv
0	col0:1,col1:1.1,col2:2
1	col0:0,col1:1.2,col2:3
2	col0:1,col1:2.3
3	col0:1,col1:0.0,col2:4

- PAI命令行

```
PAI -name TableToKV
  -project algo_public
  -DinputTableName=maple_tabletokv_basic_input
  -DoutputTableName=maple_tabletokv_basic_output
  -DselectedColNames=col0,col1,col2
  -DappendColNames=rowid;
```

- 输出说明

  输出表*maple_tabletokv_basic_output*

rowid:bigint	kv:string
0	1:1.1,2:2
1	1:1.2,2:3
2	1:2.3
3	1:0.0,2:4

示例二

- PAI命令行

```
PAI -name TableToKV
  -project projectxlib4 -DinputTableName=maple_tabletokv_basic_input
  -DoutputTableName=maple_tabletokv_basic_output
  -DselectedColNames=col0,col1,col2 -DappendColNames=rowid
  -DconvertColToIndexId=1
  -DinputKeyMapTableName=maple_test_tabletokv_basic_map_input
  -DoutputKeyMapTableName=maple_test_tabletokv_basic_map_output;
```

- 输出说明

  输出表*maple_test_tabletokv_basic_map_output*

col_name:string	col_index:string	col_datatype:string
col1	1	bigint
col2	2	double

3.3. 特征工程

3.3.1. 特征变换

本文为您介绍PAI-Studio提供的特征变换算法，包括主成分分析、特征尺度变换、特征离散、奇异值分解和特征异常平滑。

主成分分析

主成分分析（PCA）是研究如何通过少数主成分揭示多个变量间的内部结构，考察多个变量间相关性的一种多元统计方法。

PCA从原始变量中导出少数主成分，使它们尽可能多地保留原始变量的信息，并且彼此间互不相关，作为新的综合指标。

PCA使用主成分分析算法，实现了降维和降噪的功能。

> **说明** PCA目前仅支持稠密数据格式。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数，如下所示：

- 可视化方式

页签	参数	描述
字段设置	**选择特征列**	输入表中用于分析的列名称。
	附加列	附加在降维数据表后的列。
参数设置	**信息量比例**	降维后数据信息占原来的比例。
	特征分解方式	分解特征的方式，取值如下： ○ CORR ○ COVAR_SAMP ○ COVAR_POP
	数据转换方式	转换为新数据的处理方式，取值如下： ○ Simple ○ Sub-Mean ○ Normalization
执行调优	**生命周期**	指定输出表的生命周期，取值为正整数。
	节点个数	与**单个节点内存大小**参数配对使用。取值为*[1, 9999]*的正整数。
	单个节点内存大小	单位为兆。取值范围为*[1024, 64*1024]*的正整数。

- PAI命令方式

```
PAI -name PrinCompAnalysis
  -project algo_public
  -DinputTableName=bank_data
  -DeigOutputTableName=pai_temp_2032_17900_2
  -DprincompOutputTableName=pai_temp_2032_17900_1
  -DselectedColNames=pdays,previous,emp_var_rate,cons_price_idx,cons_conf_idx,euribor3m,nr_empl
oyed
  -DtransType=Simple
  -DcalcuType=CORR
  -DcontriRate=0.9;
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	进行主成分分析的输入表。	无
selectedColNames	是	输入表中用于分析的列名称。 使用逗号分隔，支持INT和DOUBLE类型。	无
eigOutputTableName	是	特征向量与特征值的输出表。	无
princompOutputTableName	是	进行主成分降维降噪后的结果输出表。	无
transType	否	转换原表为主成分分析表的方式，取值如下： ○ Simple ○ Sub-Mean ○ Normalization	*Simple*
calcuType	否	对原表进行特征分解的方式，取值如下： ○ CORR ○ COVAR_SAMP ○ COVAR_POP	*CORR*
contriRate	否	数据信息降维后保留的百分比。取值范围为*(0,1)*。	*0.9*
remainColumns	否	降维表保留原表的字段。	无
coreNum	否	节点个数，与memSizePerCore参数配对使用。取值范围为*[1, 9999]*的正整数。	系统自动分配。

参数名称	是否必选	参数描述	默认值
memSizePerCore	否	单个节点的内存大小，单位为兆。取值范围为*[1024, 64*1024]*的正整数。	系统自动分配。
lifecycle	否	指定输出表的生命周期，取值为正整数。	无

PCA输出示例

- 降维后的数据表

prin0	prin1	prin2	prin3
95.78807909...	-42.95729950747...	-67.75761249391427	-78.22763106620326
94.55356656...	-42.09090844438...	-68.61978267691727	-78.13608278651054
89.91510009...	-47.88349039874...	-70.2131838386143	-79.3464187998396
92.45302559...	-41.19278330986...	-69.35650482673981	-78.92704651042823
89.76236928...	-46.44677428041...	-66.95927201734735	-77.10906366265652
95.94066194...	-42.65895343179...	-69.30780740061341	-78.59136424888288
92.33542049...	-41.16439652633...	-69.07389275424413	-78.73867394929054
92.33143893...	-41.16317033059...	-69.07404676926707	-78.74043328209397
90.14859611...	-46.10294878666...	-69.21610483502704	-77.68690680501652

- 特征值和特征向量表

prin_name	pdays	previous	emp_var_rate	cons_price_idx	cons_conf_idx	euribor3m	nr_employed	eigenvalue	contributionrate	sumcontributionrate
prin0	0.2289...	-0.307801...	0.49043713976...	0.3676221023847...	0.103704168633...	0.49327195...	0.4723413876...	3.864397508...	0.5520567869804135	0.5520567869804135
prin1	0.6651...	-0.511030...	-0.1750362148...	-0.306844720053...	-0.38529019595...	-0.1519122...	0.0083997057...	1.333469023...	0.190495574717606...	0.7425523616980203
prin2	0.1344...	-0.248552...	-0.1027813425...	-0.355514661117...	0.882898436451...	0.01908078...	-0.057503451...	0.953306884...	0.136186697819829...	0.8787390595178498
prin3	-0.216...	0.2034146...	0.07125284421...	-0.732942136387...	-0.16599040564...	0.21796215...	0.5426739826...	0.426458636...	0.060922662369309...	0.9396617218871589

特征尺度变换

特征尺度变换的功能如下：

- 支持常见的*log2*、*log10*、*ln*、*abs*及*sqrt*等尺度变化函数。
- 支持稠密及稀疏数据格式。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数，如下所示：

- 可视化方式

页签	参数	描述
字段设置	**尺度变换特征**	需要缩放的特征。
	选择标签列	如果您设置了该字段，则可以通过可视化方式查看特征到目标变量的*x-y*分布直方图。
	是否K:V,K:V稀疏特征	训练数据是否为稀疏格式。
	保留原变换的特征	新特征加前缀scale_。

页签	参数	描述
参数设置	**尺度变化函数**	**特征尺度变换**组件支持如下尺度变化函数： ○ log2 ○ log10 ○ ln ○ abs ○ sqrt

- PAI命令方式

```
PAI -name fe_scale_runner -project algo_public
  -Dlifecycle=28
  -DscaleMethod=log2
  -DscaleCols=nr_employed
  -DinputTable=pai_dense_10_1
  -DoutputTable=pai_temp_2262_20380_1;
```

参数名称	是否必选	参数描述	默认值
inputTable	是	输入表的表名称。	无
inputTablePartitions	否	输入表中指定参与训练的分区，格式为 `Partition_name=value` 。 如果是多级分区，格式为 `name1=value1/name2=value2;` 。 如果指定多个分区，则需要使用,隔开。	输入表的所有分区。
outputTable	是	缩放尺度后的结果表。	无
scaleCols	是	勾选需要缩放的特征。 系统会自动筛选稀疏特征。您只能勾选数值类特征。	无
labelCol	否	标签字段。 如果您设置了该字段，则可以通过可视化方式查看特征到目标变量的*x-y*分布直方图。	无
categoryCols	否	将勾选的字段作为枚举特征处理，并且不支持缩放。	“”

参数名称	是否必选	参数描述	默认值
scaleMethod	否	缩放方法，取值如下： ○ log2 ○ log10 ○ ln ○ abs ○ sqrt	*log2*
scaleTopN	否	当未勾选scaleCols参数时，系统自动挑选*TopN*个需要缩放的特征。	*10*
isSparse	否	是否为*k:v*的稀疏特征。	*稠密数据*
itemSpliter	否	稀疏特征item的分隔符。	*,*
kvSpliter	否	稀疏特征item的分隔符。	*:*
lifecycle	否	结果表的生命周期。	*7*
coreNum	否	节点个数。取值范围为*[1, 9999]*的正整数。 与memSizePerCore参数配对使用。	系统自动分配。
memSizePerCore	否	单个节点的内存大小，单位为兆。取值范围为*[2048, 64 * 1024]*的正整数。	系统自动分配。

示例

- 输入数据

```
create table if not exists pai_dense_10_1 as
select
  nr_employed
from bank_data limit 10;
```

- 参数配置

  勾选nr_employed作为尺度变化特征，并且仅支持数值类特征。**尺度变化函数**选择*log2*，如下图所示。

- 运行结果

nr_employed
12.352071021075528
12.34313018339218
12.285286613666395
12.316026916036957
12.309533196497519
12.352071021075528
12.316026916036957
12.316026916036957
12.309533196497519
12.316026916036957

特征离散

离散模块的功能如下：

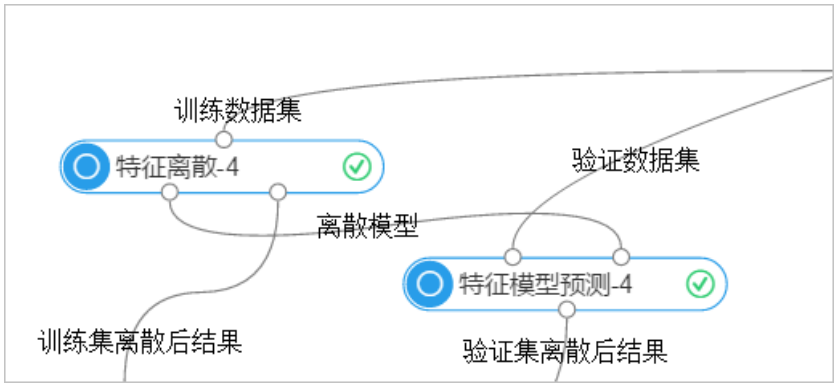
- 支持稠密数值类特征离散。
- 支持等频离散和等距离离散等无监督离散。

> **说明** 无监督离散的特征离散默认为等距离离散。

- 支持基于Gini增益离散和基于熵增益离散等有监督离散。

> **说明** 标签类特征离散必须是枚举类型STRING或BIGINT类型。

特征离散预测可以使用特征工程目录下的特征模型预测组件。详细的建模DAG图如下所示。



> **说明**
> - 您必须使用相同的离散模型做特征离散预测，才能保证特征对齐。
> - 有监督离散是根据熵增益不断遍历寻找切分断点，运行时间可能比较久。切分得到的分区数不受指定的maxBins参数限制。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数，如下所示：

- 可视化方式

页签	参数	描述
字段设置	**离散的特征**	选择需要离散的特征。如果选择的是稀疏特征，则系统会自动筛选。
	标签列	如果您设置了该字段，则可以通过可视化方式查看特征到目标变量的*x*-*y*分布直方图。
参数设置	**离散方法**	离散方法。取值如下： ○ Isometric Discretization（等距离散） ○ Isofrequecy Discretization（等频离散） ○ Gini-gain-based Discretization（基于Gini增益离散） ○ Entropy-gain-based Discretization（基于熵增益离散）
	离散区间	离散区间大小。取值为大于1的正整数。
执行调优	**计算核心数**	计算的核心数目，取值为正整数。
	每个核心内存	每个CPU分配的内存大小。

- PAI命令方式

```
PAI -name fe_discrete_runner_1 -project algo_public
  -DdiscreteMethod=SameFrequecy
  -Dlifecycle=28
  -DmaxBins=5
  -DinputTable=pai_dense_10_1
  -DdiscreteCols=nr_employed
  -DoutputTable=pai_temp_2262_20382_1
  -DmodelTable=pai_temp_2262_20382_2;
```

参数名称	是否必选	参数描述	默认值
inputTable	是	输入表的表名称。	无

参数名称	是否必选	参数描述	默认值
inputTablePartitions	否	输入表中指定参与训练的分区，格式为 `Partition_name=value` 。 如果是多级分区，格式为 `name1=value1/name2=value2;` 。 如果指定多个分区，则需要使用,隔开。	输入表的所有分区。
outputTable	是	离散后的结果表。	无
discreteCols	是	选择需要离散的特征。如果选择的是稀疏特征，则系统会自动筛选。	“”
labelCol	否	标签字段。如果您设置了该字段，则可以通过可视化方式查看特征到目标变量的*x-y*分布直方图。	无
categoryCols	否	将勾选的字段作为枚举特征处理，并且不支持离散。	默认为空
discreteMethod	否	离散方法。取值如下： ○ Isometric Discretization（等距离散） ○ Isofrequecy Discretization（等频离散） ○ Gini-gain-based Discretization（基于Gini增益离散） ○ Entropy-gain-based Discretization（基于熵增益离散）	*Isometric Discretization*
discreteTopN	否	当未勾选discreteCols参数时，系统自动挑选*TopN*个需要离散的特征。取值为正整数。	*10*
maxBins	否	离散区间大小。取值为大于1的正整数。	*100*

参数名称	是否必选	参数描述	默认值
isSparse	否	是否为k:v的稀疏特征，取值如下： ○ true ○ false 默认为稠密数据。	*false*
itemSpliter	否	稀疏特征item的分隔符。	,
kvSpliter	否	稀疏特征item的分隔符。	:
lifecycle	否	结果表生命周期。取值为正整数。	*7*
coreNum	否	节点个数。 与**memSizePerCore**参数配对使用，取值为正整数。	系统自动分配。
memSizePerCore	否	单个节点内存大小，单位为兆。取值为正整数。	系统自动分配。

示例

- 输入数据

```
create table if not exists pai_dense_10_1 as
select
  nr_employed
from bank_data limit 10;
```

- 参数配置

  输入数据为*pai_dense_10_1*。离散特征选择*nr_employed*，离散方法选择*等距离散*，离散区间配置为*5*，如下图所示。

- 运行结果

nr_employed
4.0
3.0

nr_employed
1.0
3.0
2.0
4.0
3.0
3.0
2.0
3.0

特征异常平滑

特征平滑组件可以将输入特征中包含异常的数据平滑到一定区间，支持稀疏和稠密数据格式。

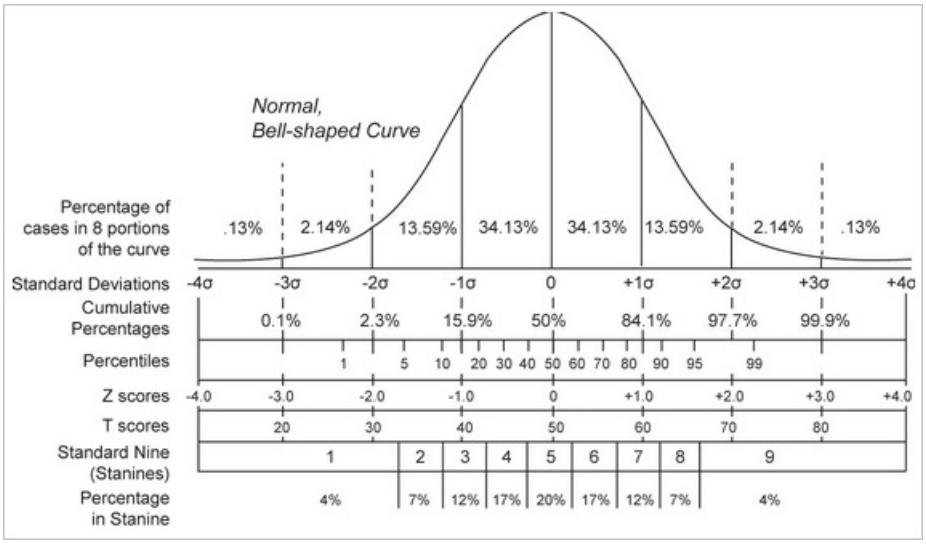
> **说明** 特征平滑组件只是将异常取值的特征值修正为正常值，本身不过滤或删除任何记录，输入数据维度和条数均不变。

平滑方法如下：

- ZScore平滑

  如果特征分布遵循正态分布，则噪音一般集中在*-3×alpha*和*3×alpha*之外，ZScore是将该范围的数据平滑到*[-3×alpha,3×alpha]*之间。

  例如，某个特征遵循正态分布，均值为*0*，标准差为*3*。因此，*-10*的特征值会被识别为异常而修正为*-3×3+0*，即为*-9*。同理*10*会被修正为*3×3+0*，即为*9*，如下图所示。



- 百分位平滑

  用于将分布不在*[minPer, maxPer]*的数据平滑到minPer和maxPer这两个分位点。

例如，age特征取值*0～200*，设置minPer为*0*，maxPer为*50%*，则不在*0～100*的特征取值都会被修正成*0*或*100*。

- 阈值平滑

  用于将分布不在*[minThresh, maxThresh]*的数据平滑到minThresh和maxThresh这两个数据点。

  例如，age特征取值*0～200*，设置minThresh为*10*，maxThresh为*80*，则不在*0～80*的特征取值都会被修正成*0*或*80*。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数，如下所示：

- 可视化方式

页签	参数	描述
字段设置	**选择平滑特征列**	需要平滑处理的特征列。
	标签列	如果您设置了该字段，则可以通过可视化方式查看特征到目标变量的*x-y*分布直方图。
参数设置	**平滑方法**	平滑方法如下： ○ ZScore平滑 ○ 百分位平滑 ○ 阈值平滑 ○ 箱线图平滑
	置信范围	置信水平。当平滑方法为**ZScore平滑**时需要配置该参数。
	阈值下限	阈值最小值，默认为*-9999*，表示不设置。 当平滑方法为**阈值平滑**时需要配置该参数。
	阈值上限	阈值最大值，默认为*-9999*，表示不设置。 当平滑方法为**阈值平滑**时需要配置该参数。
	百分位下限	最低百分位。 当平滑方法为**百分位平滑**或**箱线图平滑**时需要配置该参数。
	百分位上限	最高百分位。 当平滑方法为**百分位平滑**或**箱线图平滑**时需要配置该参数。

- PAI命令方式

```
PAI -name fe_soften_runner -project algo_public
  -DminThresh=5000
  -Dlifecycle=28
  -DsoftenMethod=min-max-thresh
  -DsoftenCols=nr_employed
  -DmaxThresh=6000
  -DinputTable=pai_dense_10_1
  -DoutputTable=pai_temp_2262_20381_1;
```

参数名称	是否必选	参数描述	默认值
inputTable	是	输入表的表名	无
inputTablePartitions	否	输入表中指定参与训练的分区，格式为 `Partition_name=value` 。 如果是多级分区，格式为 `name1=value1/name2=value2;` 。 如果指定多个分区，则需要使用,隔开。	输入表的所有分区。
outputTable	是	平滑后的结果表。	无
labelCol	否	标签字段。如果您设置了该字段，则可以通过可视化方式查看特征到目标变量的*x-y*分布直方图。	默认为空
categoryCols	否	将勾选的字段作为枚举特征处理。	默认为空
softenCols	是	选择需要平滑的特征。如果特征为稀疏特征时，系统会自动化筛选。	无
softenMethod	否	平滑方法如下： ○ ZScore平滑 ○ 百分位平滑 ○ 阈值平滑 ○ 箱线图平滑	*ZScore平滑*
softenTopN	否	当未勾选softenCols参数时，系统自动挑选*TopN*个需要平滑的特征。取值为正整数。	*10*
cl	否	置信水平。当平滑方法为**ZScore平滑**时需要配置该参数。	*10*

参数名称	是否必选	参数描述	默认值
minPer	否	最低百分位。当平滑方法为**百分位平滑**或**箱线图平滑**时需要配置该参数。	*0.0*
maxPer	否	最高百分位。当平滑方法为**百分位平滑**或**箱线图平滑**时需要配置该参数。	*1.0*
minThresh	否	阈值最小值。当平滑方法为**阈值平滑**时需要配置该参数。	*-9999*
maxThresh	否	阈值最大值。当平滑方法为**阈值平滑**时需要配置该参数。	*-9999*
isSparse	否	是否为k:v的稀疏特征，取值如下： ○ true ○ false 默认为稠密数据。	*false*
itemSpliter	否	稀疏特征item的分隔符。	*,*
kvSpliter	否	稀疏特征item的分隔符。	*:*
lifecycle	否	结果表生命周期。取值为正整数。	*7*
coreNum	否	节点个数。与**memSizePerCore**参数配对使用，取值为[1, 9999]的正整数。	系统自动分配。
memSizePerCore	否	单个节点内存大小，单位为兆。取值为[2048, 64*1024]的正整数。	系统自动分配。

示例

- 输入数据

```
create table if not exists pai_dense_10_1 as
select
  nr_employed
from bank_data limit 10;
```

nr_employed
5228.1

nr_employed
5195.8
4991.6
5099.1
5076.2
5228.1
5099.1
5099.1
5076.2
5099.1

- 参数配置

  平滑特征选择*nr_employed*，**参数设置**中平滑方法选择*阈值平滑*，阈值下限为*5000*，阈值上限为*6000*，如下图所示。

- 运行结果

nr_employed
5228.1
5195.8
5000.0
5099.1
5076.2
5228.1

nr_employed
5099.1
5099.1
5076.2
5099.1

奇异值分解

奇异值分解（Singular Value Decomposition）是线性代数中一种重要的矩阵分解，是矩阵分析中正规矩阵求对角化的推广。在信号处理、统计学等领域有重要应用。

奇异值分解的公式为 $X=USV'$ 。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数，如下所示：

- 可视化方式

页签	参数	描述
字段设置	**选择特征列**	存放k:v，k:v格式的列。
参数设置	**保留奇异值个数**	期望求解的top奇异组个数。默认求解全部奇异组。
	精度误差	允许的期望的误差精度。
执行调优	**单个节点内存大小**	单位为兆，与memSizePerCore参数配对使用，取值为*[1, 9999]*的正整数。
	节点个数	取值为*[1024, 64*1024]*的正整数。
	生命周期	指定输出表的生命周期。

- PAI命令方式

```
PAI -name svd
  -project algo_public
  -DinputTableName=bank_data
  -DselectedColNames=col0
  -DenableSparse=true
  -Dk=5
  -DoutputUTableName=u_table
  -DoutputVTableName=v_table
  -DoutputSTableName=s_table;
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	进行主成分分析的输入表。	无

参数名称	是否必选	参数描述	默认值
selectedColNames	否	输入表中用于分析的列名称，使用逗号分隔。 如果是稀疏矩阵，支持STRING类型。如果是表，支持INT和DOUBLE类型。	默认选择所有列。
inputTablePartitions	否	输入表中指定参与分析的分区，格式为 `Partition_name=value` 。 如果是多级分区，格式为 `name1=value1/name2=value2，` 。 如果指定多个分区，则需要使用,隔开。	默认选择所有分区。
outputUTableName	是	U矩阵的输出表，维度是 `m * sgNum` 。其中，*m*为数据表的行数，*sgNum*为计算出的奇异值个数。	无
outputSTableName	是	S矩阵的输出表，维度是 `sgNum * sgNum` 。其中，*sgNum*为计算出的奇异值个数。	无
outputVTableName	是	V矩阵的输出表，维度是 `n * sgNum` 。其中，*n*是矩阵的列数，*sgNum*是计算出的奇异值个数。	无
k	是	期望的奇异值个数。 输出的结果为可能比K值小的正整数。	无
tol	否	收敛误差	*1.0e-06*
enableSparse	否	输入数据是否为稀疏格式： ○ true ○ false	*false*
itemDelimiter	否	指当输入表数据为稀疏格式时，kv间的分隔符。	空格

参数名称	是否必选	参数描述	默认值
kvDelimiter	否	指当输入表数据为稀疏格式时，key和value的分隔符。	冒号
coreNum	否	节点个数。 与**memSizePerCore**参数配对使用，取值为*[1, 9999]*的正整数。	系统自动分配。
memSizePerCore	否	单个节点内存大小，单位为兆。取值为*[1024, 64*1024]*的正整数。	系统自动分配。
lifecycle	否	指定输出表的生命周期。取值为正整数。	无

示例

- 数据生成

```
drop table if exists svd_test_input;
create table svd_test_input
as
select
  *
from
(
 select
    '0:3.9079 2:0.0009 3:0.0416 4:0.17664 6:0.36460 8:0.091330' as col0
  from dual
  union all
 select
    '0:0.09229 2:0.4872172 5:0.5267 8:0.4544 9:0.23317' as col0
  from dual
  union all
  select
  '1:0.8312 3:0.9317 5:0.5680 7:0.5560 9:0.0508' as col0
  from dual
  union all
  select
  '2:0.767 5:0.01891 8:0.25235 ' as col0
  from dual
  union all
  select
  '0:0.29819 2:0.87598086 6:0.5315568 ' as col0
  from dual
  union all
  select
  '0:0.920260 2:0.5154311513 4:0.8104 5:0.188420 8:0.88' as col0
  from dual
) a;
```

- PAI命令

```
PAI -name svd
  -project algo_public
  -DinputTableName=svd_test_input
  -DselectedColNames=col0
  -DenableSparse=true
  -Dk=5
  -DoutputUTableName=u_table
  -DoutputVTableName=v_table
  -DoutputSTableName=s_table;
```

- 算法规模：10万列。

3.3.2. 特征重要性评估

本文为您介绍PAI-Studio提供的特征重要性评估，包括随机森林特征重要性和线性模型特征重要性。

随机森林特征重要性

您可以使用原始数据和随机森林模型，计算特征重要性。您可以通过以下任意一种方式，配置随机森林特征重要性组件参数：

- 可视化方式

页签	参数	描述
字段设置	**选择特征列**	输入表中，用于训练的特征列。默认选中除Label外的所有列，为可选项。
	选择目标列	该参数为必选项。 单击图标，在**选择字段**对话框中，输入关键字搜索列，选中后单击**确定**。
参数设置	**并行计算核数**	并行计算的核心数，可选。
	每个核内存大小	每个核的内存大小，单位为MB，可选。

- PAI命令格式

```
pai -name feature_importance -project algo_public
  -DinputTableName=pai_dense_10_10
  -DmodelName=xlab_m_random_forests_1_20318_v0
  -DoutputTableName=erkang_test_dev.pai_temp_2252_20319_1
  -DlabelColName=y
  - DfeatureColNames="pdays,previous,emp_var_rate,cons_price_idx,cons_conf_idx,euribor3m,nr_employed,age,campaign,poutcome"
  -Dlifecycle=28;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称	无
outputTableName	是	输出表的名称	无
labelColName	是	输入表的标签列名	无
modelName	是	输入的模型名称	无
featureColNames	否	输入表选择的特征列	除Label外的所有列
inputTablePartitions	否	输入表选择的分区名称	选择全表
lifecycle	否	输出表的生命周期	不设置
coreNum	否	核心数	自动计算
memSizePerCore	否	内存数	自动计算

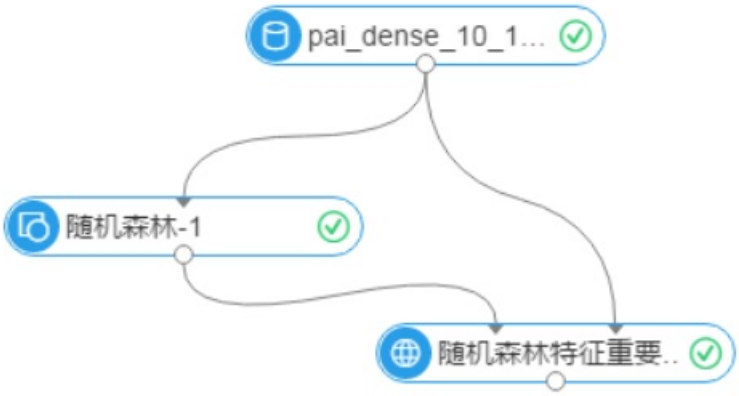
随机森林特征重要性示例

1. 使用SQL语句，生成训练数据。

```
drop table if exists pai_dense_10_10;
create table if not exists pai_dense_10_10 as
select
  age,campaign,pdays, previous, poutcome, emp_var_rate, cons_price_idx, cons_conf_idx, euribor3m,
nr_employed, y
from  bank_data limit 10;
```

2. 构建如下实验，详情请参见算法建模。

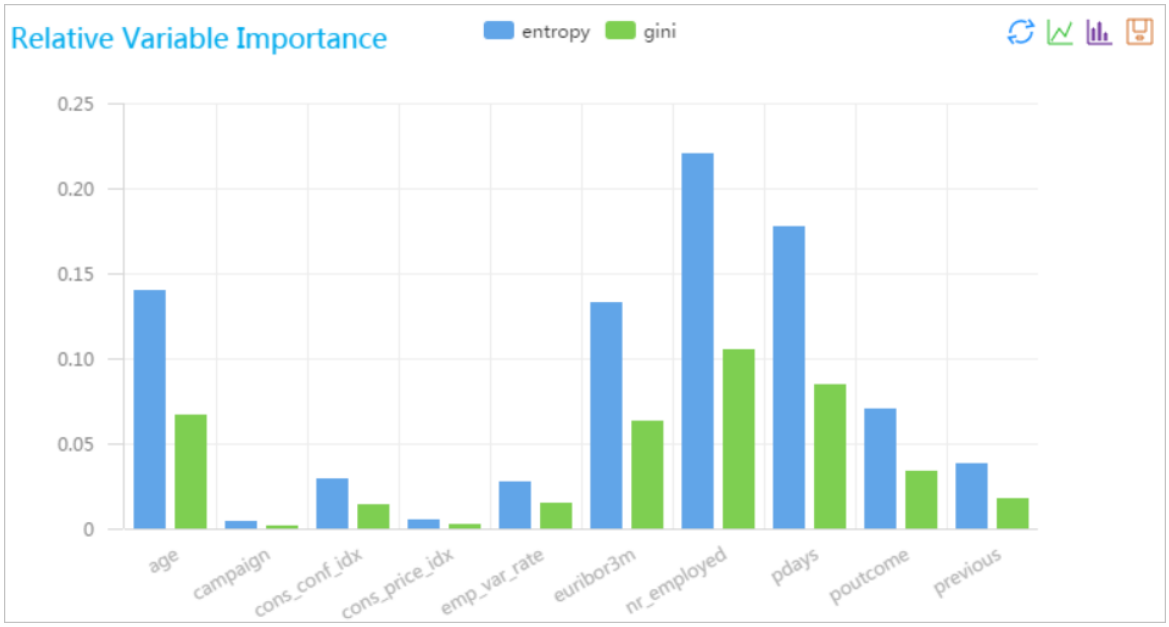
   数据源为pai_dense_10_10，y为随机森林的标签列，其它列为特征列。**强制转换列**选择age和campaign，表示将这两列作为枚举特征处理，其它使用默认参数。



3. 运行实验，查看预测结果。

colname	gini	entropy
age	0.06625000000000003	0.13978726292803723
campaign	0.0017500000000000003	0.004348515545596772
cons_conf_idx	0.013999999999999999	0.02908409497018851
cons_price_idx	0.002	0.0049804499913461255
emp_var_rate	0.014700000000000003	0.026786360680260933
euribor3m	0.06300000000000003	0.1321936348846039
nr_employed	0.10499999999999998	0.2203227248076733
pdays	0.0845	0.17750329234397513
poutcome	0.03360000000000001	0.07050327193845542
previous	0.017700000000000004	0.03810381005801592

4. 运行完成后，右键单击随机森林特征重要性组件，选择**查看分析报告**，查看结果。



线性模型特征重要性

计算线性模型的特征重要性，包括线性回归和二分类逻辑回归，支持稀疏和稠密数据格式。您可以通过以下任意一种方式，配置线性模型特征重要性组件参数：

- 可视化方式

页签	参数	描述

页签	参数	描述
字段设置	选择特征列	输入表中，用于训练的特征列。默认选中除Label外的所有列，为可选项。
	选择目标列	该参数为必选项。 单击图标，在**选择字段**对话框中，输入关键字搜索列，选中后单击**确定**。
	输入表数据是否为稀疏格式	可选项。
执行调优	计算核心数	计算的核心数，可选。
	每个核内存大小	每个核的内存大小，单位为MB，可选。

- PAI命令方式

```
PAI -name regression_feature_importance -project algo_public
  -DmodelName=xlab_m_logisticregressi_20317_v0
  -DoutputTableName=pai_temp_2252_20321_1
  -DlabelColName=y
 -DfeatureColNames=pdays,previous,emp_var_rate,cons_price_idx,cons_conf_idx,euribor3m,nr_employed,age,campaign
  -DenableSparse=false -DinputTableName=pai_dense_10_9;
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的表名	无
outputTableName	是	输出表的表名	无
labelColName	是	输入表的标签列名	无
modelName	是	输入的模型名称	无
featureColNames	否	输入表选择的特征列	除Label外的所有列
inputTablePartitions	否	输入表选择的分区名称	选择全表
enableSparse	否	输入表是否为稀疏格式	false
itemDelimiter	否	当输入表数据为稀疏格式时，KV对之间的分隔符。	空格
kvDelimiter	否	当输入表数据为稀疏格式时，Key和Value之间的分隔符。	英文冒号（:）
lifecycle	否	输出表的生命周期	不设置

参数	是否必选	描述	默认值
coreNum	否	核心数	自动计算
memSizePerCore	否	内存数	自动计算

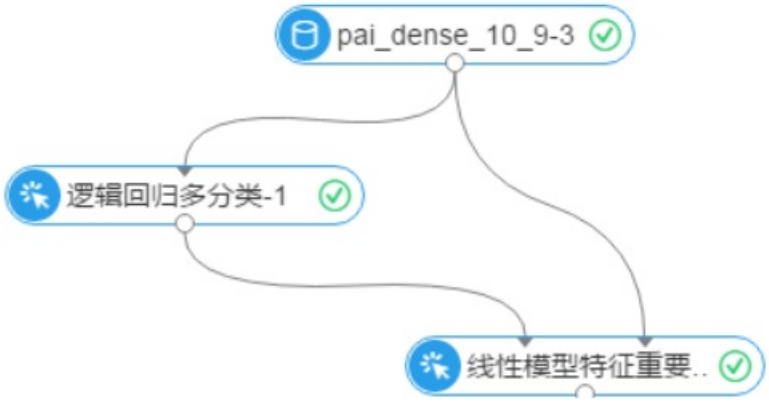
线性模型特征重要性示例

1. 使用SQL语句，生成训练数据。

```
create table if not exists pai_dense_10_9 as
select
  age,campaign,pdays, previous, emp_var_rate, cons_price_idx, cons_conf_idx, euribor3m, nr_employ
ed, y
from  bank_data limit 10;
```

2. 构建如下实验，详情请参见算法建模。

   y为逻辑回归多分类组件的标签列，其它字段为特征列，其它参数使用默认值。



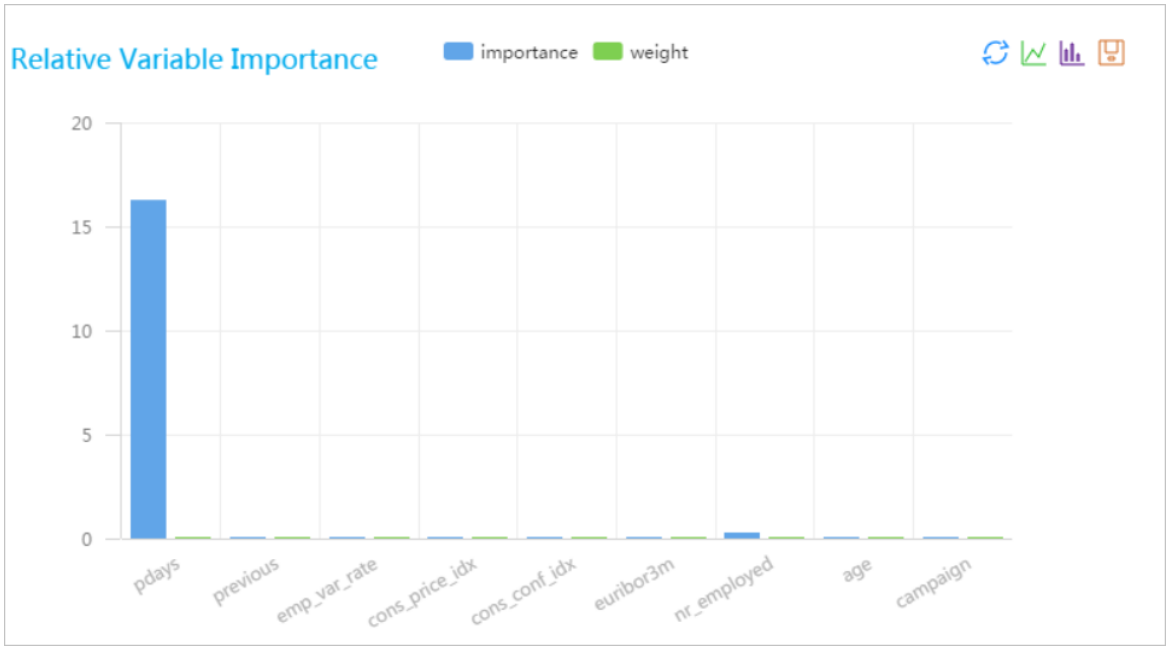
3. 运行实验，查看预测结果。

colname	weight	importance
pdays	0.033942600256583334	16.31387797440866
previous	0.00004248130342485344	0.000030038817725357177
emp_var_rate	0.00006720242617694611	0.00010554561260753949
cons_price_idx	0.00012311047229142307	0.00006581255124425219
cons_conf_idx	0.00017227965471819213	0.0008918770542818432
euribor3m	0.00006113758212679113	0.00010427128177450988
nr_employed	0.0034541377310490697	0.26048098230126043
age	0.00009618162708080744	0.0009267659744232966
campaign	0.000019142551785274455	0.00004179335366052985 5

指标计算公式如下。

列名	公式
weight	abs(w_)
importance	abs(w_j) * STD(f_i)

4. 运行完成后，右键单击线性模型特征重要性组件，选择**查看分析报告**，查看结果。



3.3.3. 特征重要性过滤

特征重要性过滤组件为线性特征重要性、GBDT特征重要性和随机森林特征重要性等组件提供过滤功能，支持过滤TopN的特征。

组件配置

PAI命令

```
PAI -name fe_filter_runner -project algo_public
  -DselectedCols=pdays,previous,emp_var_rate,cons_price_idx,cons_conf_idx,euribor3m,nr_employed,age
,campaign,poutcome
  -DinputTable=pai_dense_10_10
  -DweightTable=pai_temp_2252_20319_1
  -DtopN=5
  -DmodelTable=pai_temp_2252_20320_2
  -DoutputTable=pai_temp_2252_20320_1;
```

参数	描述	是否必选
inputTable	输入表名。	是

参数	描述	是否必选
inputTablePartitions	系统默认选择所有分区。指定输入表的分区： • 指定单个分区，格式为 partition_name=value 。 • 指定多个分区，格式为 name1=value1,name2=value2 。 说明 多个分区之间用英文逗号（,）分隔。 • 指定多级分区，格式为 name1=value1/name2=value2 。	否
weightTable	特征重要性的权重表（即线性特征重要性、GBDT特征重要性、随机森林特征重要性的输出表）。	是
outputTable	过滤出TopN特征的输出表。	是
modelTable	特征过滤产出的模型文件。	是
selectedCols	默认输入表的所有字段列名。	否
topN	TopN特征，默认10。 说明 仅支持输入正整数。	否
lifecycle	输出表生命周期，默认7。 说明 仅支持输入正整数。	否

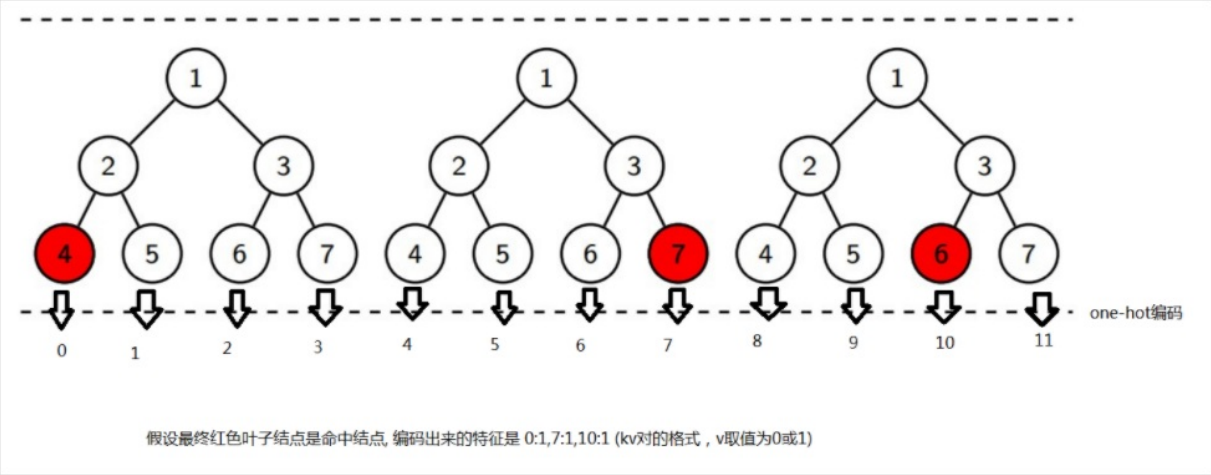
3.3.4. 特征生成

本文为您介绍PAI-Studio提供的特征生成算法，包括特征编码和one-hot编码。

特征编码

特征编码是由决策树和Ensemble算法挖掘新特征的一种策略，特征来自一个或多个特征组成的决策树叶子结点的one-hot结果。

例如，下图有三棵树，共有34个构成的叶子结点。根据树的顺序依次编码为0~11号特征，其中第0棵树的叶子结点占据0~3号特征，第二棵树占据4~7号特征，第三棵树占据8~11号特征。该编码策略可以有效转换GBDT非线性特征为线性特征。



您可以通过以下任意一种方式，配置特征编码组件参数：

- 可视化方式

页签	参数	描述
字段设置	**特征列**	输入表中，用于训练的特征列。
	标签列	该参数为必选项。 单击图标，在**选择字段**对话框中，输入关键字搜索列，选中后单击**确定**。
	附加输出列	可选，保留原特征至输出结果表。
参数设置	**计算核心数**	计算的核心数，格式为正整数。
	每个核心内存数	每个核心的内存数量，格式为正整数。

- PAI命令格式

```
PAI -name fe_encode_runner -project algo_public
  -DinputTable="pai_temp_2159_19087_1"
  -DencodeModel="xlab_m_GBDT_LR_1_19064"
  -DselectedCols="pdays,previous,emp_var_rate,cons_price_idx,cons_conf_idx,euribor3m,nr_employed,
age,campaign"
  -DlabelCol="y"
  -DoutputTable="pai_temp_2159_19061_1";
 -DcoreNum=10
 -DmemSizePerCore=1024
```

参数名称	是否必选	描述	默认值
inputTable	是	输入表的名称。	无
inputTablePartitions	否	输入表中指定参与训练的分区，格式为partition_name=value。 如果是多级，格式为name1=value1/name2=value2。 如果指定多个分区，使用英文逗号（,）分隔。	输入表的所有分区
encodeModel	是	编码的输入GBDT二分类的模型。	无
outputTable	是	缩放尺度后的结果表。	无
selectedCols	是	勾选GBDT参与编码的特征，通常是GBDT组件的训练特征。	无
labelCol	是	标签字段。	无
lifecycle	否	结果表的生命周期。	7
coreNum	否	指定Instance的总数，支持BIGINT类型。	-1，会根据输入数据量计算需要的Instance的数量。
memSizePerCore	否	指定memory大小。	-1，会根据输入数据量计算需要的内存大小。

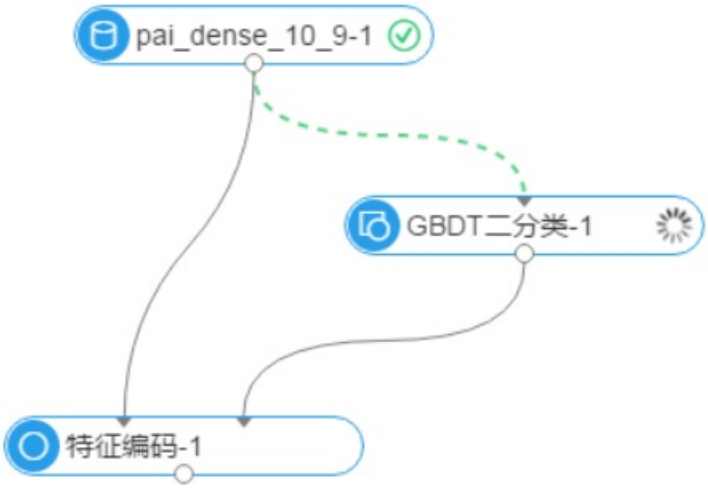
特征编码示例

1. 使用SQL语句，生成训练数据。

```
CREATE TABLE IF NOT EXISTS tdl_pai_bank_test1
(
  age        BIGINT COMMENT '',
  campaign     BIGINT COMMENT '',
  pdays       BIGINT COMMENT '',
  previous     BIGINT COMMENT '',
  emp_var_rate   DOUBLE COMMENT '',
  cons_price_idx DOUBLE COMMENT '',
  cons_conf_idx  DOUBLE COMMENT '',
  euribor3m     DOUBLE COMMENT '',
  nr_employed    DOUBLE COMMENT '',
  y          BIGINT COMMENT ''
)
LIFECYCLE 7;
insert overwrite table tdl_pai_bank_test1
select * from
(select 53 as age,1 as campaign,999 as pdays,0 as previous,-0.1 as emp_var_rate,
    93.2 as cons_price_idx,-42.0 as cons_conf_idx, 4.021 as euribor3m,5195.8 as nr_employed,0 as y
from dual
union all
select 28 as age,3 as campaign,6 as pdays,2 as previous,-1.7 as emp_var_rate,
    94.055 as cons_price_idx,-39.8 as cons_conf_idx, 0.729 as euribor3m,4991.6 as nr_employed,1 as y
from dual
union all
select 39 as age,2 as campaign,999 as pdays,0 as previous,-1.8 as emp_var_rate,
    93.075 as cons_price_idx,-47.1 as cons_conf_idx, 1.405 as euribor3m,5099.8 as nr_employed,0 as y
from dual
union all
select 55 as age,1 as campaign,3 as pdays,1 as previous,-2.9 as emp_var_rate,
    92.201 as cons_price_idx,-31.4 as cons_conf_idx, 0.869 as euribor3m,5076.2 as nr_employed,1 as y
from dual
union all
select 30 as age,8 as campaign,999 as pdays,0 as previous,1.4 as emp_var_rate,
    93.918 as cons_price_idx,-42.7 as cons_conf_idx, 4.961 as euribor3m,5228.2 as nr_employed,0 as y
from dual
union all
select 37 as age,1 as campaign,999 as pdays,0 as previous,-1.8 as emp_var_rate,
    92.893 as cons_price_idx,-46.2 as cons_conf_idx, 1.327 as euribor3m,5099.1 as nr_employed,0 as y
from dual
union all
select 39 as age,1 as campaign,999 as pdays,0 as previous,-1.8 as emp_var_rate,
    92.893 as cons_price_idx,-46.2 as cons_conf_idx, 1.313 as euribor3m,5099.1 as nr_employed,0 as y
from dual
union all
select 36 as age,1 as campaign,3 as pdays,1 as previous,-2.9 as emp_var_rate,
    92.963 as cons_price_idx,-40.8 as cons_conf_idx, 1.266 as euribor3m,5076.2 as nr_employed,1 as y
from dual
union all
select 27 as age,2 as campaign,999 as pdays,1 as previous,-1.8 as emp_var_rate,
    93.075 as cons_price_idx,-47.1 as cons_conf_idx, 1.41 as euribor3m,5099.1 as nr_employed,0 as y
from dual
) a
```

2. 构建如下实验，通常与GBDT二分类组件配合使用。详情请参见算法建模。

设置GBDT二分类组件的参数，树的数目为5，树的最大深度为3，y为标签列，其它字段为特征列。



3. 运行实验，查看预测结果。

kv	y
2:1,5:1,8:1,12:1,15:1,18:1,28:1,34:1,41:1,50:1,53:1,63:1,72:1	0.0
2:1,5:1,6:1,12:1,15:1,16:1,28:1,34:1,41:1,50:1,51:1,63:1,72:1	0.0
2:1,3:1,12:1,13:1,28:1,34:1,36:1,39:1,55:1,61:1	1.0
2:1,3:1,12:1,13:1,20:1,21:1,22:1,42:1,43:1,46:1,63:1,64:1,67:1,68:1	0.0
0:1,10:1,28:1,29:1,32:1,36:1,37:1,55:1,56:1,59:1	1.0

输出的效果可以直接输入至逻辑回归二分类或多分类组件，通常效果会比单独使用LR或GDBT的效果好，且不易拟合。

one-hot编码

one-hot编码组件包括训练和预测功能：

- one-hot编码组件训练功能：
  - 输入节点：该组件第一个输入节点（左）是训练数据的输入，训练时无需设置输入节点（右）。
  - 输出节点：该组件有两个输出节点，一个是编码后的结果数据表（左），另一个是模型表（右），该模型表用于对同类型的新数据进行one-hot编码。
- one-hot编码组件预测功能

  one-hot组件的第二个输入节点（右）是one-hot的模型输入，可以利用已有的one-hot模型对新数据进行one-hot编码。

您可以通过以下任意一种方式，配置one-hot组件参数：

- 可视化方式

页签	参数	描述
字段设置	选择二值化列	选择二值化的字段，必选项。
	其他保留特征	勾选特征保留输出在KV对。勾选的字段当作特征（不进行one-hot编码）输出至KV字段中，保留的特征会从0开始编码，该字段必须为DOUBLE类型。
	附加列	附加在输出表的列，可选项。
参数设置	生命周期	表的生命周期，默认值为7。
	输出表的类型	包括kv和table。当离散特征较多时，建议输出kv格式，table仅支持1024列，超出将报错。
	节点个数	节点的个数。
	单个结点内存大小	单个结点的内存大小，单位为MB。
	删除最后一个枚举量的编码	该参数为true时，可以保证编码后数据的线性无关性。
	忽略待编码数据中的空元素	该参数为true时，表示不对空元素编码。

- PAI命令方式

```
PAI -name one_hot
 -project algo_public
  -DinputTable=one_hot_test
  -DbinaryCols=f0,f1,f2
  -DmodelTable=one_hot_model
  -DoutputTable=one_hot_output
  -Dlifecycle=28;
```

参数名称	是否必选	描述	默认值
inputTable	是	输入表的名称	无
inputTablePartitions	否	输入表选择的分区名称	输入表的所有分区
binaryCols	是	one-hot编码字段，必须是枚举类特征，字段可以是任意类型。	无
reserveCols	否	勾选的字段当作特征（不进行one-hot编码）输出至KV字段中，保留的特征会从0开始编码，该字段必须为DOUBLE类型。	空字符串

参数名称	是否必选	描述	默认值
appendCols	否	勾选的字段原封不动地输出输入表的字段至输出表中。	无
outputTable	是	one-hot后的结果表，编码结果保存在KV字段中。	无
inputModelTable	否	one-hot编码的输入模型表。 **说明** inputModelTable和outputModelTable中必须有一个为非空字符串。	空字符串
outputModelTable	否	one-hot编码的输出模型表。 **说明** inputModelTable和outputModelTable中必须有一个为非空字符串。	空字符串
lifecycle	否	输出表的生命周期	7
dropLast	是	是否删除最后一个枚举量的编码。该参数为true时，可以保证编码后数据的线性无关性。	*false*
outputTableType	是	输出表的类型，包括稀疏表和稠密表。当离散特征较多时，建议输出kv格式.table仅支持1024列，超出将报错。	*kv*
ignoreNull	是	是否忽略待编码数据中的空元素。该参数为*true*时，表示不对空元素编码。	*false*
coreNum	否	节点的个数。	自动计算

参数名称	是否必选	描述	默认值
memSizePerCore	否	单个结点的内存大小，单位为MB，范围为[2048, 64*1024]。	自动计算

说明如下：

- inputModelTable、outputModelTable至少一个参数非空。当inputModelTable非空时，其对应的表应该为非空的模型表。
- 编码字段列中，离散值个数可以支持至千万量级。
- 当训练的模型作为下次编码使用的模型时，由于dropLast、ignoreNull和reserveCols编码参数的效果已封装到模型中，无法进行调整。如果您需要调整，则必须重新训练。
- 建议输出表使用kv格式。当使用表时，列数最多为1024列。当超出该值则会报错，无法完成编码任务。
- one-hot编码生成的kv表默认从0开始编号。
- 当使用模型编码新数据时，如果在模型映射表中无法找到数据中的离散量，则忽略该离散量，即不对该离散量编码。如果需要对其进行编码，请重新训练模型映射表。

one-hot编码示例

1. 使用SQL语句，生成训练数据。

```
PAI -project projectxlib4
 -name one_hot
 -DinputTable=one_hot_yh
 -DbinaryCols=f0,f2,f4
 -DoutputModelTable=one_hot_model_8
 -DoutputTable=one_hot_in_table_1_output_8
 -DdropLast=false
 -DappendCols=f0,f2,f4
 -DignoreNull=false
 -DoutputTableType=table
 -DreserveCols=f3
 -DcoreNum=4
 -DmemSizePerCore=2048;
```

2. 测试输入表如下。

f0	f1	f2	f3	f4
12	prefix1	1970-09-15 12:50:22	0.1	true
12	prefix3	1971-01-22 03:15:33	0.4	false
NULL	prefix3	1970-01-01 08:00:00	0.2	NULL
3	NULL	1970-01-01 08:00:00	0.3	false

f0	f1	f2	f3	f4
34	NULL	1970-09-15 12:50:22	0.4	NULL
3	prefix1	1970-09-15 12:50:22	0.2	true
3	prefix1	1970-09-15 12:50:22	0.3	false
3	prefix3	1970-01-01 08:00:00	0.2	true
3	prefix3	1971-01-22 03:15:33	0.1	false
NULL	prefix3	1970-01-01 08:00:00	0.3	false

该输入表中，f0为BIGINT类型、f1为STRING类型、f2为DATATIME类型、f3为DOUBLE类型、f4为BOOL类型。

3. 测试结果为映射模型表。

col_name	col_value	mapping
reserve	f3	0
f0	12	1
f0	3	2
f0	34	3
f0	null	4
f2	22222222222	5
f2	33333333333	6
f2	4	7
f4	0	8
f4	1	9
f4	null	10

结果表中最上面一行为reserve行，列名值固定为reserve，保存reserve信息。其余行对应的是编码映射信息：

- 编码后的表（table）

f0	f1	f3	f4	_reserve__f3_0	f0_12_1	f0_3_2	f0_34_3	f0_null_4	f2_22222_5	f2_33333_6	f2_4_7	f4_0_8	f4_1_9	f4_null_10
12	prefix1	0.1	true	0.1	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
12	prefix3	0.4	false	0.4	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
NULL	prefix3	0.2	NULL	0.2	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
3	NULL	0.3	false	0.3	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
34	NULL	0.4	NULL	0.4	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
3	prefix1	0.2	true	0.2	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
3	prefix1	0.3	false	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
3	prefix3	0.2	true	0.2	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
3	prefix3	0.1	false	0.1	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
NULL	prefix3	0.3	false	0.3	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0

- 编码后的表（kv）

f0	f1	f3	f4	kv
12	prefix1	0.1	true	0:0.1,1:1,5:1,9:1
12	prefix3	0.4	false	0:0.4,1:1,6:1,8:1

f0	f1	f3	f4	kv
NULL	prefix3	0.2	NULL	0:0.2,4:1,7:1,10:1
3	NULL	0.3	false	0:0.3,2:1,7:1,8:1
34	NULL	0.4	NULL	0:0.4,3:1,5:1,10:1
3	prefix1	0.2	true	0:0.2,2:1,5:1,9:1
3	prefix1	0.3	false	0:0.3,2:1,5:1,8:1
3	prefix3	0.2	true	0:0.2,2:1,7:1,9:1
3	prefix3	0.1	false	0:0.1,2:1,6:1,8:1
NULL	prefix3	0.3	false	0:0.3,4:1,7:1,8:1

扩展性测试

测试数据：样本数为2亿，枚举量为10万。测试数据表如下。

f0	f1
94	prefix3689
9664	prefix5682
2062	prefix5530
9075	prefix9854
9836	prefix1764
5140	prefix1149
3455	prefix7272
2508	prefix7139
7993	prefix1551
5602	prefix4606
3132	prefix5767

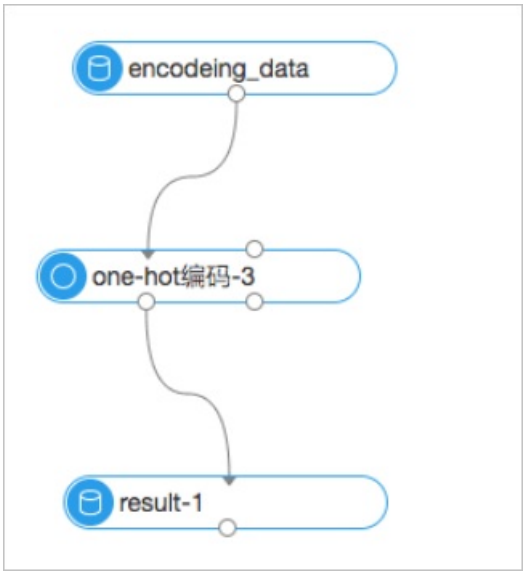
测试结果如下。

core num	train time	predict time	加速比
5	84s	181s	1/1

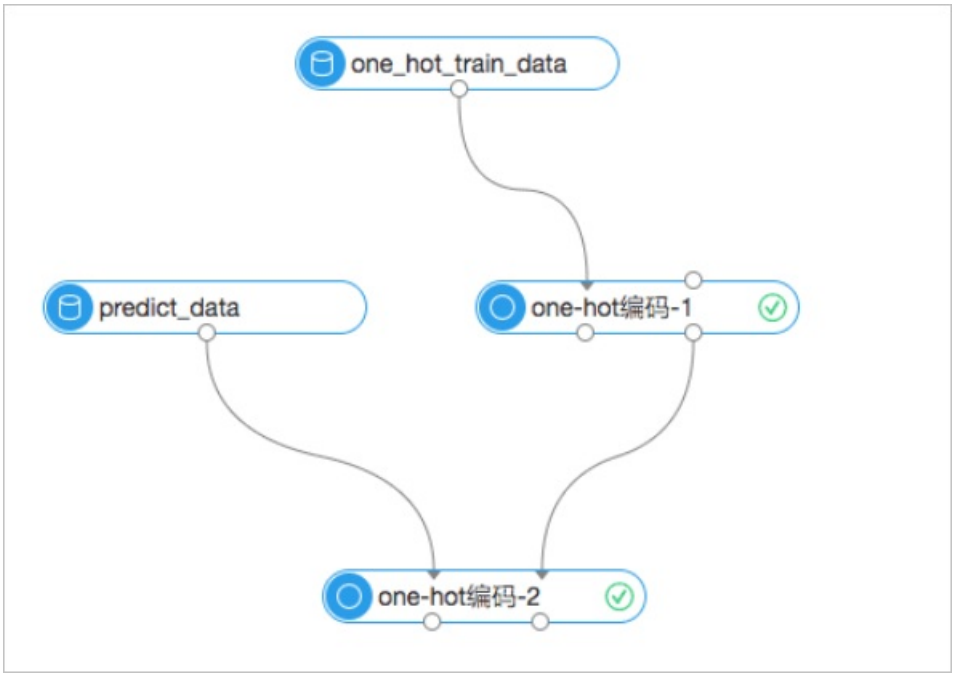
core num	train time	predict time	加速比
10	60s	93s	1.4/1.95
20	46s	56s	1.8/3.23

Web端的使用说明：

- 直接使用该组件进行编码，实验流程如下。



- 使用该组件的训练模型，再通过模型对数据进行编码，实验流程如下。



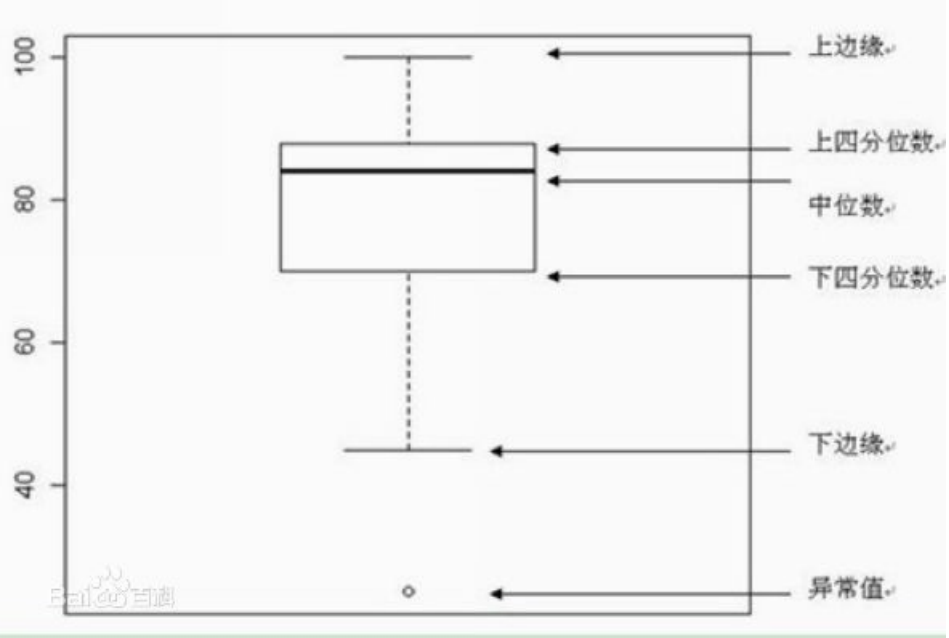
3.3.5. 异常检测

异常检测用于检测连续值和枚举值类特征的数据，帮助您挖掘数据中的异常点。

背景信息

异常检测的方法包括箱型图（Box-plot）和AVF（Attribute Value Frequency）：

- 箱型图用于检测连续值类特征的数据，根据箱线图最大值和最小值检测异常特征。



- AVF用于检测枚举值类特征的数据，根据枚举特征的取值频率及阈值检测异常特征。

组件配置

- 可视化方式

页签	参数	描述
字段设置	特征列	选择需要分析的字段。
	异常检测方法	选择检测的方法。箱型图用于检测连续类特征。AVF用于检测枚举类特征。

- PAI命令

```
PAI -name fe_detect_runner -project algo_public
    -DselectedCols="emp_var_rate,cons_price_rate,cons_conf_idx,euribor3m,nr_employed" \
    -Dlifecycle="28"
    -DdetectStrategy="boxPlot"
    -DmodelTable="pai_temp_2458_23565_2"
    -DinputTable="pai_bank_data"
    -DoutputTable="pai_temp_2458_23565_1";
```

参数名称	参数描述	是否必选
inputTable	输入表的表名。	是

参数名称	参数描述	是否必选
inputTablePartitions	系统默认选择所有分区。指定输入表的分区： ○ 指定单个分区，格式为 partition_name=value 。 ○ 指定多个分区，格式为 name1=value1,name2=value2 。 说明 多个分区之间用英文逗号（,）分隔。 ○ 指定多级分区，格式为 name1=value1/name2=value2 。	否
selectedCols	输入特征，字段类型没有限制。	是
detectStrategy	系统支持Box-plot和AVF选项。 Box-plot用于检测连续值类特征。 AVF用于检测枚举值类特征。	是
outputTable	异常检测结果数据集，即检测到异常特征的数据集。	是
modelTable	异常检测模型。	是
lifecycle	输出表的生命周期，系统默认为7。	否
coreNum	节点个数，与参数memSizePerCore配对使用，取值范围[1,9999]。 说明 仅支持配置正整数。	否
memSizePerCore	单个结点内存大小，取值范围[2048,64 *1024]，单位MB。	否

3.3.6. 基于分箱组件实现连续特征离散化

本文为您介绍如何使用分箱组件进行连续特征离散化。

前提条件

完成项目的创建，详情请参见创建项目。

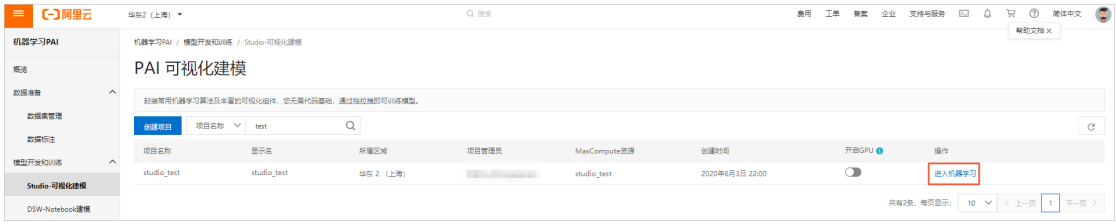
背景信息

特征离散是将连续的数据进行分段，使其变为多个离散化区间。针对该场景，PAI推出了分箱组件，支持等频分箱、等宽分箱及自动分箱。

本文首先使用**读数据表**组件，读取公共数据表pai_online_project.iris_data。然后使用**分箱**组件生成离散数据。最后使用**数据转换模块**将原始数据从连续值转换为离散值。

操作步骤

1. 进入PAI-Studio控制台。
   i. 登录PAI控制台。
   ii. 在左侧导航栏，选择**模型开发和训练 > Studio-可视化建模**。
   iii. 在**PAI可视化建模**页面，单击**进入机器学习**。

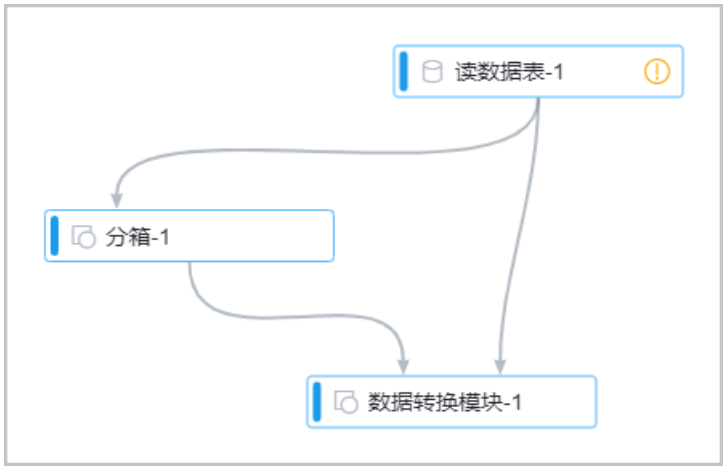


2. 创建空白实验。
   i. 在左侧导航栏，单击**首页**。
   ii. 单击**新建实验 > 新建空白实验**。
   iii. 在**新建实验**对话框，配置参数。

参数	描述
名称	输入基于分箱组件实现连续特征离散化。
项目	不支持修改。
描述	输入使用PAI提供的分箱组件，实现连续特征离散化。
位置	选择我的实验。

   iv. 单击**确定**。
3. 构建实验流程。
   i. 在左侧导航栏，单击**组件**。
   ii. 在组件列表，将**源/目标**下的**读数据表**组件拖入画布中。
   iii. 在组件列表，将**金融板块**下的**分箱**和**数据转换模块**组件拖入画布中。

iv. 将以上组件拼接为如下实验。



4. 配置组件参数。

i. 单击画布中的**读数据表**组件，在右侧面板，配置实验数据源。

页签	参数	描述
表选择	**表名**	输入pai_online_project.iris_data。
	分区	该公共数据表为非分区表，因此**分区**复选框不支持选中。
字段信息	**源表字段信息**	配置**表名**后，系统会自动同步该数据表的**源表字段信息**，无需手动配置。

ii. 单击画布中的**分箱**组件，在右侧面板，配置参数（仅配置如下参数，其他参数使用默认值即可）。

页签	参数	描述
IO/字段设置	**特征列**	选择f1、f2、f3及f4列。
参数设置	**分箱个数**	配置为10，表示将连续特征离散化至10个区间中。
	分箱方式	支持**等频**、**等宽**及**自动分箱**。使用**自动分箱**时，数据源必须包含label字段，且为二分类场景。本文以**等频**分箱为例。

iii. 单击画布中的**数据转换模块**组件，在右侧面板，配置参数（仅配置如下参数，其他参数使用默认值即可）。

页签	参数	描述
IO/字段设置	**不进行转换的数据列**	选择type列，该列会原样输出。
	数据转换的类型	选择**Index**。

5. 单击画布上方的**运行**。

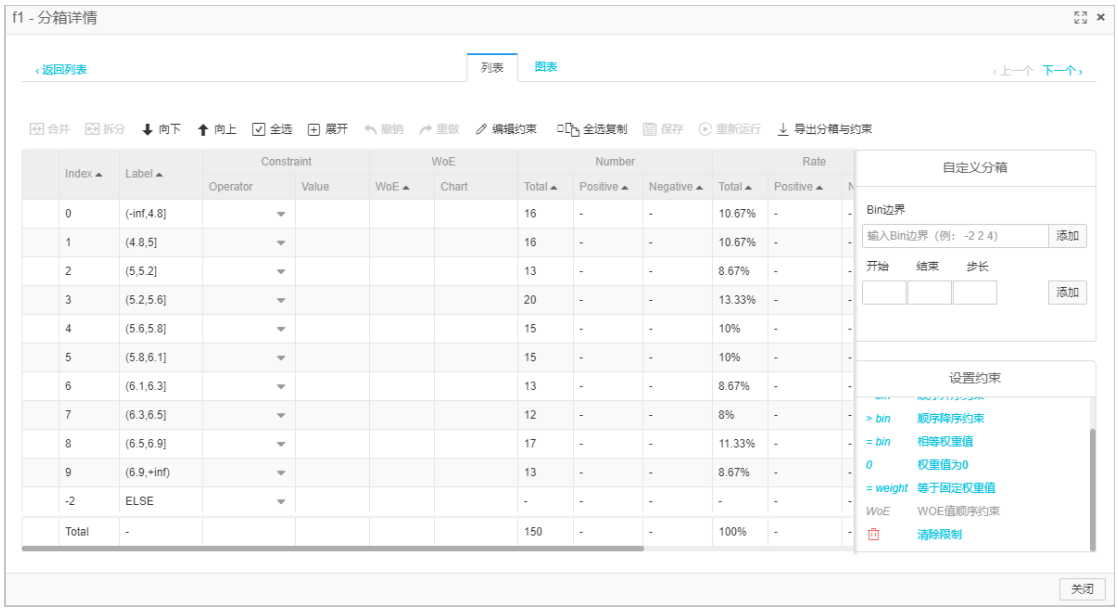
6. 查看实验结果。

i. 实验运行结束后，右键单击画布中的**数据转换模块**组件，在快捷菜单，单击**查看数据**，即可查看离散化结果。

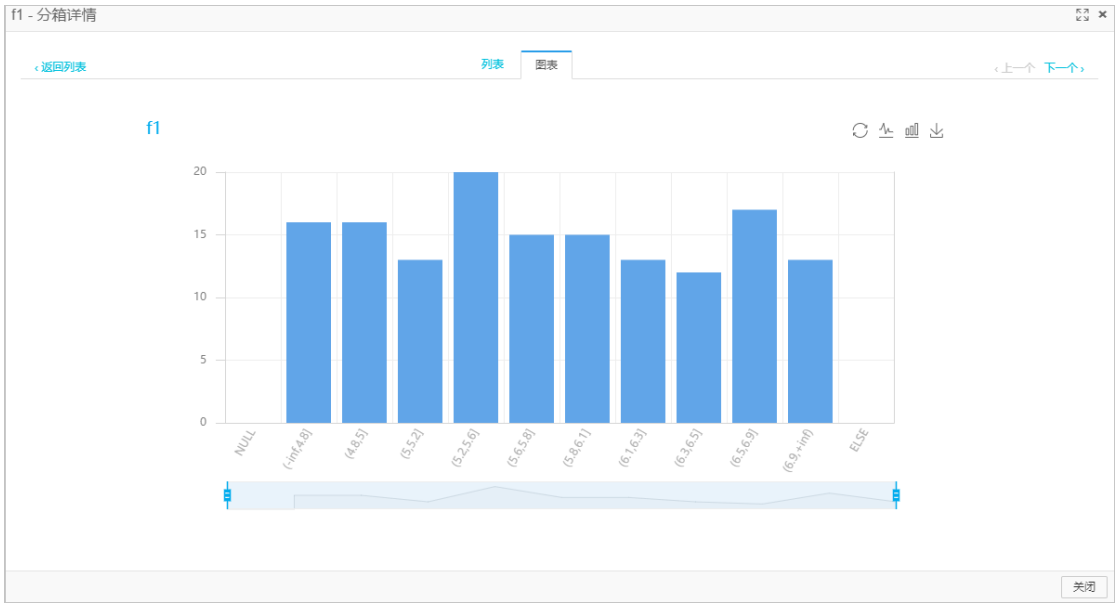
type	f1	f2	f3	f4
Iris-setosa	2	7	0	0
Iris-setosa	1	3	0	0
Iris-setosa	0	5	0	0
Iris-setosa	0	4	1	0
Iris-setosa	1	7	0	0
Iris-setosa	3	8	2	1
Iris-setosa	0	6	0	1
Iris-setosa	1	6	1	0
Iris-setosa	0	3	0	0
Iris-setosa	1	4	1	0
Iris-setosa	3	8	1	0
Iris-setosa	0	6	2	0
Iris-setosa	0	3	0	0
Iris-setosa	0	3	0	0
Iris-setosa	4	8	0	0
Iris-setosa	4	8	1	1
Iris-setosa	3	8	0	1

ii. 右键单击画布中的**分箱**组件，在快捷菜单，单击**我要分箱**。

iii. 单击待查看特征（以f1特征列为例）**操作**列下的**详情**，即可查看该特征的分箱详情，如下图所示。



iv. 单击**图表**页签，以图表的形式查看分箱结果。



3.4. 统计分析

3.4.1. 直方图

直方图（Histogram）又称质量分布图，是一种统计报告图，由一系列高度不等的纵向条纹或线段表示数据分布的情况。 通常，横轴表示数据类型，纵轴表示分布情况。

组件配置

您可以通过以下方式，配置离散值特征分析组件参数：

- 可视化方式

页签	参数	描述
字段设置	选择字段	选择需要分析的字段。支持double和bigint类型。 **说明** 最大不超过1024个字段。
参数设置	区间个数	直方图的区间个数。
执行调优	计算核心数	计算的核心数，取值范围为正整数。
	每个核内存数	每个核心的内存，取值范围为1 MB~65536 MB。

- PAI命令

```
PAI -name histogram
    -project algo_public
    -DinputTableName=maple_histogram_1to20_input
    -DoutputTableName=maple_histogram_1to20_output
    -DselectedColNames=col0,col1 -DintervalNum=20;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 说明 指定多个分区时，分区之间使用英文逗号（,）分隔。	无
outputTableName	是	输出表名称。	无
selectedColNames	是	输入表中用于训练的列名，以逗号分隔，支持int和double类型。 说明 最大不超过1024列。	无
intervalNum	否	直方图区间个数。	100
lifecycle	否	表的声明周期。	无
coreNum	否	计算的核心数，取值范围为正整数。取值范围[1, 9999]。	系统自动分配
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	系统自动分配

示例

- 输入说明

col0	col1
1	1.0
2	2.0
3	3.0
4	4.0

col0	col1
5	5.0
6	6.0
7	7.0
8	8.0
9	9.0
10	10.0
11	11.0
12	12.0
13	13.0
14	14.0
15	15.0
16	16.0
17	17.0
18	18.0
19	19.0
20	20.0

- PAI命令

```
PAI -name histogram
  -project algo_public
  -DinputTableName=maple_histogram_1to20_input
  -DoutputTableName=maple_histogram_1to20_output
  -DselectedColNames=col0,col1 -DintervalNum=20;
```

- 输出说明

colname	histogram
col0	[1, 1.95):1;[1.95, 2.9):1;[2.9, 3.85):1;[3.85, 4.8):1;[4.8, 5.75):1;[5.75, 6.7):1;[6.7, 7.65):1;[7.65, 8.6):1;[8.6, 9.55):1;[9.55, 10.5):1;[10.5, 11.45):1;[11.45, 12.4):1;[12.4, 13.35):1;[13.35, 14.3):1;[14.3, 15.25):1;[15.25, 16.2):1;[16.2, 17.15):1;[17.15, 18.1):1;[18.1, 19.05):1;[19.05, 20]:1

colname	histogram
col1	[1, 1.95):1;[1.95, 2.9):1;[2.9, 3.85):1;[3.85, 4.8):1;[4.8, 5.75):1;[5.75, 6.7):1;[6.7, 7.65):1;[7.65, 8.6):1;[8.6, 9.55):1;[9.55, 10.5):1;[10.5, 11.45):1;[11.45, 12.4):1;[12.4, 13.35):1;[13.35, 14.3):1;[14.3, 15.25):1;[15.25, 16.2):1;[16.2, 17.15):1;[17.15, 18.1):1;[18.1, 19.05):1;[19.05, 20]:1

3.4.2. 皮尔森系数

皮尔森系数是一种线性相关系数，用于反映两个变量线性相关程度的统计量。机器学习中，皮尔森系数用于计算输入表或分区两列（数值列）的Pearson相关系数，计算结果输出至输出表。

组件配置

您可以通过以下方式，配置离散值特征分析组件参数：

- 可视化方式

页签	参数	描述
IO/字段设置	输入列1	输入计算相关系数列名。
	输入列2。	输入计算相关系数列名。

- PAI命令

```
pai -name pearson
  -project algo_public
  -DinputTableName=wpbc
  -Dcol1Name=f1
  -Dcol2Name=f2
  -DoutputTableName=wpbc_pear;
```

参数名称	参数描述	是否必选
inputTableName	输入表的表名。	是
inputTablePartitions	系统默认选择所有分区。指定输入表的分区： ○ 指定单个分区，格式为 `partition_name=value` 。 ○ 指定多个分区，格式为 `name1=value1,name2=value2` 。 说明 多个分区之间用英文逗号（,）分隔。 ○ 指定多级分区，格式为 `name1=value1/name2=value2` 。	否

参数名称	参数描述	是否必选
col1Name	输入列1的列名。	是
col2Name	输入列2的列名。	是
outputTableName	输出结果表的表名。	是
lifecycle	输出表的生命周期。系统默认无生命周期。 **说明** 仅支持输入正整数。	否

示例

- 输入表

```
create table pai_pearson_test_input as
select * from
(
select 1.0 as f0,0.11 as f1
union all
select 2.0 as f0,0.12 as f1
union all
select 3.0 as f0,0.13 as f1
union all
select 5.0 as f0,0.15 as f1
union all
select 8.0 as f0,0.18 as f1
)tmp;
```

- PAI命令

```
pai -name pearson
  -project algo_public
  -DinputTableName=pai_pearson_test_input
  -Dcol1Name=f0
  -Dcol2Name=f1
  -DoutputTableName=pai_pearson_test_output;
```

- 输出表

```
+------------+------------+------------+------------+------------+------------+--------------------+
| src_table  | src_parts  | col1_name  | col2_name  | count_total| count_valid| pearson_coefficient|
+------------+------------+------------+------------+------------+------------+--------------------+
| sre_mpi_algo_dev.pai_pearson_test_input |        | f0      | f1      | 5       | 5       | 0.9999999999999973 |
+------------+------------+------------+------------+------------+------------+--------------------+
```

3.4.3. 百分位

百分位是统计学术语，用于计算数据表列数据的百分位。一组数据从小到大排序，并计算相应数据的百分位，则某百分位所对应数据的值称为该百分位的百分位数。

背景信息

- 系统仅支持Bigint，Double和Datetime类型的数据计算百分位。
- 计算百分位时，空列，默认跳过。如果全部为空列，则运行报错。
- colName支持配置多列数据。

组件配置

- 可视化方式

页签	参数	描述
参数设置	输入列	选择输入列的字段。
执行调优	核数量	节点个数。
	每个核的内存大小	单个节点内存大小。

- PAI命令

```
PAI -name Percentile
    -project algo_public
    -DinputTableName=maple_test_percentile_3col_input
    -DcolName=col0,col1,col2 -DoutputTableName=maple_test_percentile_3col_output;
```

参数名称	参数描述	是否
inputTableName	输入表名。	是
outputTableName	输出表名。	是
colName	计算的列名，默认值选择所有列。 **说明** 多个列名之间使用英文逗号（,）分割。	否

参数名称	参数描述	是否
inputPartitions	系统默认选择所有分区。指定输入表的分区： ○ 指定单个分区，格式为 `partition_name=value` 。 ○ 指定多个分区，格式为 `name1=value1,name2=value2` 。 **说明** 多个分区之间用英文逗号（,）分隔。 ○ 指定多级分区，格式为 `name1=value1/name2=value2` 。	否
predictInputTableName	预测表表名，配置该项参数后即可输出预测结果。	否
predictInputTablePartitions	输入预测表分区。	否
predictSelectedColNames	预测表列名，系统默认选取全表。该列名需要与训练表中对应的列相同。	否
predictSelectedOriginalColNames	输入表中保持原有数据的列名，默认选择所有列，以英文逗号（,）分隔。	否
predictOutputTableName	输出预测表名，与predictInputTableName参数配对使用。	否
lifecycle	输出表的生命周期，系统默认无生命周期。 **说明** 仅支持输入正整数。	否
coreNum	节点个数，取值范围为[1, 9999]。与参数memSizePerCore配对使用。 **说明** 仅支持输入正整数。	否

参数名称	参数描述	是否
memSizePerCore	单个节点内存大小，取值范围为范围[1024, 64*1024]，单位MB。 **说明** 仅支持输入正整数。	否

示例

- 输入表

col0:double（1000行）	col1:bigint（100行）	col2:bigint（300行）
962	88	Tue Oct 15 00:26:40 CST 1974
218	99	Thu Jan 04 20:53:20 CST 1973
565	44	Sat Mar 09 02:40:00 CST 1974
314	68	Mon Aug 11 22:40:00 CST 1975
583	13	Sat Aug 23 12:26:40 CST 1975
615	87	Tue May 25 14:13:20 CST 1971
70	53	Fri Mar 23 09:20:00 CST 1979
929	63	Mon Jul 03 16:26:40 CST 1972
249	48	Thu Mar 15 07:33:20 CST 1973
428	62	Wed Mar 17 03:33:20 CST 1971
119	1	Thu Jun 26 15:33:20 CST 1975
756	27	Mon Jan 30 17:20:00 CST 1978
490	75	Wed Dec 11 21:20:00 CST 1974
957	12	Sun Jul 05 12:26:40 CST 1970
80	22	Wed Oct 04 06:40:00 CST 1972
681	57	Wed Nov 03 15:06:40 CST 1971
13	95	Sat Sep 12 23:06:40 CST 1970

- PAI命令

```
PAI -name Percentile
  -project algo_public
  -DinputTableName=maple_test_percentile_3col_input
  -DcolName=col0,col1,col2 -DoutputTableName=maple_test_percentile_3col_output;
```

- 输出表

quantile:bigint	col0:double	col1:bigint	col2:datetime
0	0.0	0	Thu Jan 01 08:00:00 CST 1970
1	9.0	0	Sat Jan 24 11:33:20 CST 1970
2	19.0	1	Sat Feb 28 04:53:20 CST 1970
3	29.0	2	Fri Apr 03 22:13:20 CST 1970
4	39.0	3	Fri May 08 15:33:20 CST 1970
5	49.0	4	Fri Jun 12 08:53:20 CST 1970
6	59.0	5	Fri Jul 17 02:13:20 CST 1970
7	69.0	6	Thu Aug 20 19:33:20 CST 1970
8	79.0	7	Thu Sep 24 12:53:20 CST 1970
9	89.0	8	Thu Oct 29 06:13:20 CST 1970
10	99.0	9	Wed Dec 02 23:33:20 CST 1970
11	109.0	10	Wed Jan 06 16:53:20 CST 1971
12	119.0	11	Wed Feb 10 10:13:20 CST 1971
13	129.0	12	Wed Mar 17 03:33:20 CST 1971
14	139.0	13	Tue Apr 20 20:53:20 CST 1971

quantile:bigint	col0:double	col1:bigint	col2:datetime
15	149.0	14	Tue May 25 14:13:20 CST 1971
16	159.0	15	Tue Jun 29 07:33:20 CST 1971
...	...	...	...
84	839.0	83	Thu Dec 15 10:13:20 CST 1977
85	849.0	84	Thu Jan 19 03:33:20 CST 1978
86	859.0	85	Wed Feb 22 20:53:20 CST 1978
87	869.0	86	Wed Mar 29 14:13:20 CST 1978
88	879.0	87	Wed May 03 07:33:20 CST 1978
89	889.0	88	Wed Jun 07 00:53:20 CST 1978
90	899.0	89	Tue Jul 11 18:13:20 CST 1978
91	909.0	90	Tue Aug 15 11:33:20 CST 1978
92	919.0	91	Tue Sep 19 04:53:20 CST 1978
93	929.0	92	Mon Oct 23 22:13:20 CST 1978
94	939.0	93	Mon Nov 27 15:33:20 CST 1978
95	949.0	94	Mon Jan 01 08:53:20 CST 1979
96	959.0	95	Mon Feb 05 02:13:20 CST 1979
97	969.0	96	Sun Mar 11 19:33:20 CST 1979
98	979.0	97	Sun Apr 15 12:53:20 CST 1979

quantile:bigint	col0:double	col1:bigint	col2:datetime
99	989.0	98	Sun May 20 06:13:20 CST 1979
100	999.0	99	Sat Jun 23 23:33:20 CST 1979

3.4.4. 全表统计

全表统计用于统计全表，或某些选中的列。

组件配置

- 可视化方式

页签	参数	描述
参数设置	输入列	选择输入列的字段，系统默认选择全部列。
执行调优	核数目	节点个数。
	内存数	单个节点内存大小。

- PAI命令

```
PAI -name stat_summary
-project algo_public
-DinputTableName=test_data
-DoutputTableName=test_summary_out
-DinputTablePartitions="ds='20160101'"
-DselectColNames=col0,col1,col2
-Dlifecycle=1
```

参数名称	参数描述	是否必选
inputTableName	输入表名。	是
outputTableName	输出表名。	是
inputTablePartitions	输入表的分区，系统默认选择所有分区。	否
selectColNames	需要统计的列名。	否

参数名称	参数描述	是否必选
lifecycle	输出结果表的生命周期，系统默认不设置生命周期。 说明 仅支持输入正整数。	否
coreNum	节点数量。 说明 仅支持输入正整数。	否
memSizePerCore	每个节点的内存数，取值范围[1024, 64*1024]，单位MB。 说明 仅支持输入正整数。	否

输出格式

输出统计结果的全部字段，如下表所示。

列名	描述
colname	列名
datatype	类型
totalcount	总数
count	非NULL数量
missingcount	NULL数量
nancount	NAN数量
positiveinfinitycount	正无穷数量
negativeinfinitycount	负无穷数量
min	最小值
max	最大值
mean	平均值

列名	描述
variance	方差
standarddeviation	标准差
standarderror	标准误差
skewness	偏度
kurtosis	峰度
moment2	二阶矩
moment3	三阶矩
moment4	四阶矩
centralmoment2	二阶中心距
centralmoment3	三阶中心距
centralmoment4	四阶中心距
sum	总和
sum2	平方和
sum3	立方和
sum4	四次方和

示例

- 输入数据

```
drop table if exists summary_test_input;
create table summary_test_input as
select
 *
from
(
 select 'a' as col1, 1 as col2, 0.001 as col3 from dual
  union all
 select 'b' as col1, 2 as col2, 100.01 as col3 from dual
) tmp;
```

- PAI命令

```
PAI -name stat_summary
-project algo_public
-DinputTableName=summary_test_input
-DoutputTableName=summary_test_input_out
-DselectColNames=col1,col2,col3
-Dlifecycle=1;
```

- 输出结果

```
 | colname  | datatype  | totalcount | count  | missingcount | nancount  | positiveinfinitycount | negativeinfinitycount | min  | max  | mean  | variance  | standarddeviation | standarderror | skewness  | kurtosis  | moment2  | moment3  | moment4  | centralmoment2 | centralmoment3 | centralmoment4 | sum | sum2  | sum3  | sum4  |
 | col1  | string  | 2  | 2  | 0  | 0  | 0  | 0  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  | NULL  |
 | col2  | bigint  | 2  | 2  | 0  | 0  | 0  | 0  | 1  | 2  | 1.5  | 0.5  | 0.7071067811865476 | 0.5  | 0  | -2  | 2.5  | 4.5  | 8.5  | 0.25  | 0  | 0.0625  | 3  | 5  | 9  | 17  |
 | col3  | double  | 2  | 2  | 0  | 0  | 0  | 0  | 0.001  | 100.01  | 50.0055  | 5000.900040500001 | 70.71704207968544 | 50.00450000000001 | 2.327677906939552e-16 | -1.999999999999999 | 5001.000050500001 | 500150.0150005006 | 50020003.00020002 | 2500.45002025  | 2.91038304567337e-11 | 6252250.303768232 | 100.011  | 10002.000101 | 1000300.030001001 | 100040006.0004 |
```

3.4.5. 离散值特征分析

本文为您介绍PAI-Studio提供的离散值特征分析。

离散值特征分析统计离散特征的分布情况。包括gini，entropy，gini gain，infomation gain，infomation gain ratio等指标。计算每个离散值对应的gini，entropy，计算单列对应的gini gain，infomation gain，infomation gain ratio。

- gini index:

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - fi)$$

- entropy:

$$I_E(f) = -\sum_{i=1}^{m} f_i log_2 f_i$$

配置组件

您可以通过以下任意一种方式，配置离散值特征分析组件参数：

- 可视化方式

参数	描述
特征列	用来表现训练样本数据特征的列。

参数	描述
标签列	标签字段。
稀疏矩阵	当输入表数据为稀疏格式时，需要设置KV格式的特征。

- PAI命令方式

```
PAI
-name enum_feature_selection
-project algo_public
-DinputTableName=enumfeautreselection_input
-DlabelColName=label
-DfeatureColNames=col0,col1
-DenableSparse=false
-DoutputCntTableName=enumfeautreselection_output_cntTable
-DoutputValueTableName=enumfeautreselection_output_valuetable
-DoutputEnumValueTableName=enumfeautreselection_output_enumvaluetable;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 说明 指定多个分区时，分区之间使用英文逗号（,）分隔。	默认选择全表
featureColNames	否	输入表中，用于训练的特征列名。	无
labelColName	否	输入表中，标签列的名称。	无
enableSparse	否	输入数据是否为稀疏格式，取值范围为{ture,false}。	false
kvFeatureColNames	否	KV格式的特征。	默认选择全表
kvDelimiter	否	当输入表数据为稀疏格式时，key和value之间的分隔符。	英文冒号（:）
itemDelimiter	否	当输入表数据为稀疏格式时，KV对之间的分隔符。	英文逗号（,）
outputCntTableName	否	输出离散特征的枚举值分布数表。	不涉及

参数名称	是否必选	描述	默认值
outputValueTableName	否	输出离散特征的gini、entropy表。	不涉及
outputEnumValueTableName	否	输出离散特征枚举值gini、entropy表。	不涉及
lifecycle	否	表的生命周期。	无
coreNum	否	计算的核心数，取值范围为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	系统自动分配

示例

使用如下SQL语句，生成输入数据。

```
drop table if exists enum_feature_selection_test_input;
create table enum_feature_selection_test_input
as
select
  *
from
(
  select
    '00' as col_string,
    1 as col_bigint,
    0.0 as col_double
  from dual
  union all
    select
      cast(null as string) as col_string,
      0 as col_bigint,
      0.0 as col_double
    from dual
  union all
    select
      '01' as col_string,
      0 as col_bigint,
      1.0 as col_double
    from dual
  union all
    select
      '01' as col_string,
      1 as col_bigint,
      cast(null as double) as col_double
    from dual
  union all
    select
      '01' as col_string,
      1 as col_bigint,
      1.0 as col_double
    from dual
  union all
    select
      '00' as col_string,
      0 as col_bigint,
      0.0 as col_double
    from dual
) tmp;
```

输入数据如下所示。

```
+------------+------------+------------+
| col_string | col_bigint | col_double |
+------------+------------+------------+
| 01         | 1          | 1.0        |
| 01         | 0          | 1.0        |
| 01         | 1          | NULL       |
| NULL       | 0          | 0.0        |
| 00         | 1          | 0.0        |
| 00         | 0          | 0.0        |
+------------+------------+------------+
```

- PAI命令方式
  - 运行命令

```
drop table if exists enum_feature_selection_test_input_enum_value_output;
drop table if exists enum_feature_selection_test_input_cnt_output;
drop table if exists enum_feature_selection_test_input_value_output;
PAI -name enum_feature_selection -project algo_public -DitemDelimiter=":" -Dlifecycle="28" -DoutputValueTableName="enum_feature_selection_test_input_value_output" -DkvDelimiter="," -DlabelColName="col_bigint" -DfeatureColNames="col_double,col_string" -DoutputEnumValueTableName="enum_feature_selection_test_input_enum_value_output" -DenableSparse="false" -DinputTableName="enum_feature_selection_test_input" -DoutputCntTableName="enum_feature_selection_test_input_cnt_output";
```

- 运行结果
  - enum_feature_selection_test_input_cnt_output

```
+------------+------------+------------+------------+
| colname    | colvalue   | labelvalue | cnt        |
+------------+------------+------------+------------+
| col_double | NULL       | 1          | 1          |
| col_double | 0          | 0          | 2          |
| col_double | 0          | 1          | 1          |
| col_double | 1          | 0          | 1          |
| col_double | 1          | 1          | 1          |
| col_string | NULL       | 0          | 1          |
| col_string | 00         | 0          | 1          |
| col_string | 00         | 1          | 1          |
| col_string | 01         | 0          | 1          |
| col_string | 01         | 1          | 2          |
+------------+------------+------------+------------+
```

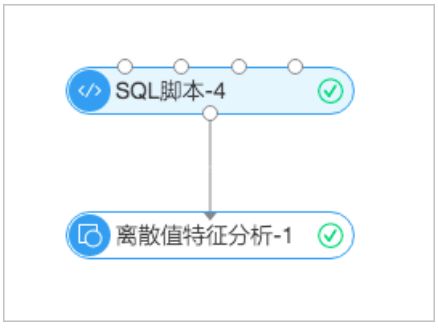
  - enum_feature_selection_test_input_value_output

```
+------------+------------+------------+------------+------------+---------------+
| colname    | gini       | entropy    | infogain   | ginigain   | infogainratio |
+------------+------------+------------+------------+------------+---------------+
| col_double | 0.3888888888888889 | 0.792481250360578 | 0.20751874963942196 | 0.1111111111111111 | 0.14221913160264427 |
| col_string | 0.38888888888888884 | 0.792481250360578 | 0.20751874963942196 | 0.1111111111111116 | 0.14221913160264427 |
+------------+------------+------------+------------+------------+---------------+
```

  - enum_feature_selection_test_input_enum_value_output

```
+------------+------------+------------+------------+
| colname    | colvalue   | gini       | entropy    |
+------------+------------+------------+------------+
| col_double | NULL       | 0.0        | 0.0        |
| col_double | 0          | 0.22222222222222224 | 0.4591479170272448 |
| col_double | 1          | 0.16666666666666666 | 0.3333333333333333 |
| col_string | NULL       | 0.0        | 0.0        |
| col_string | 00         | 0.16666666666666666 | 0.3333333333333333 |
| col_string | 01         | 0.2222222222222222 | 0.4591479170272448 |
+------------+------------+------------+------------+
```

- 可视化方式
  - 组件界面



- 参数设置

- 运行结果

3.4.6. 单样本T检验

本文为您介绍PAI-Studio提供的单样本T检验。

单样本T检验旨在检验某个变量的总体均值与某个指定值之间是否存在显著差异，其检验的样本必须总体服从正态分布。

- 可视化方式

页签	参数	描述
字段设置	样本1所在列	进行训练的样本1所在列。
	对立假设类型	对立假设的类型。
	置信度	检测结果的置信度。
	假设均值大小	假设均值的大小。

页签 参数设置	参数	描述
	两总体方差是否相等	两个总体值的方差是否相等。可选true或者false。
	节点个数	节点个数，正整数格式。
	单个节点内存大小	每个节点的内存大小。取值范围1 MB~65536 MB。

- PAI命令方式

```
pai -name t_test -project algo_public
  -DxTableName=pai_t_test_all_type
  -DxColName=col1_double
  -DoutputTableName=pai_t_test_out
  -DxTablePartitions=ds=2010/dt=1
  -Dalternative=less
  -Dmu=47
  -DconfidenceLevel=0.95
```

参数	是否必须	描述	默认值
xTableName	是	表名称	无
xColName	是	需要进行T检验的列	无
outputTableName	是	输出表名称	无
xTablePartitions	否	表的分区列表	空
alternative	否	对立假设。取值包括：two.sided、less、greater。	two.sided
mu	否	假设的均值	0
confidenceLevel	否	置信度。取值包括：0.8、0.9、0.95、0.99、0.995、0.999。	0.95

输出说明

输出一个一行一列的JSON格式表。

```
{
  "AlternativeHypthesis": "mean not equals to 0",
  "ConfidenceInterval": "(44.72234194006504, 46.27765805993496)",
  "ConfidenceLevel": 0.95,
  "alpha": 0.05,
  "df": 99,
  "mean": 45.5,
  "p": 0,
  "stdDeviation": 3.919647479510927,
  "t": 116.081867662439
}
```


3.4.7. 卡方拟合性检验

本文为您介绍PAI-Studio提供的卡方拟合性检验。

配置组件

卡方拟合性检验适用于变量为类别型变量的场景，旨在检验单个多项分类型变量在各分类间的实际观测次数与理论次数是否一致，其零假设为观测次数与理论次数无差异。

- 可视化方式

参数	描述
检验列	进行训练的检验数据列。
类别概率	类别概率配置，格式为 `类别：概率` ，所有概率和为1。

- PAI命令方式

```
PAI -name chisq_test
  -project algo_public
  -DinputTableName=pai_chisq_test_input
  -DcolName=f0
  -DprobConfig=0:0.3,1:0.7
  -DoutputTableName=pai_chisq_test_output0
  -DoutputDetailTableName=pai_chisq_test_output0_detail
```

参数	是否必须	描述	默认值
inputTableName	是	输入表的名称。	无
colName	是	列名称	无
outputTableName	是	输出表名称	无
outputDetailTableName	是	输出详细表名称。	无

参数	是否必须	描述	默认值
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 说明 指定多个分区时，分区之间使用英文逗号（,）分隔。	默认为空
probConfig	否	类别概率配置，格式为 类别：概率 ，所有概率和为1。	默认所有概率值相等

示例

- 测试数据

```
create table pai_chisq_test_input as
select * from
(
 select '1' as f0,'2' as f1 from dual
 union all
 select '1' as f0,'3' as f1 from dual
 union all
 select '1' as f0,'4' as f1 from dual
 union all
 select '0' as f0,'3' as f1 from dual
 union all
 select '0' as f0,'4' as f1 from dual
)tmp;
```

- PAI命令

```
PAI -name chisq_test
  -project algo_public
  -DinputTableName=pai_chisq_test_input
  -DcolName=f0
  -DprobConfig=0:0.3,1:0.7
  -DoutputTableName=pai_chisq_test_output0
  -DoutputDetailTableName=pai_chisq_test_output0_detail
```

- 输出说明

- 输出JSON格式的表outputTableName，只有一行一列。

```
{
  "Chi-Square": {
    "comment": "皮尔逊卡方",
    "df": 1,
    "p-value": 0.75,
    "value": 0.2380952380952381
  }
}
```

- 输出详细表outputDetailTableName，字段如下。

column name	comment
参数colName	类别
observed	观察频率
expected	期望频率
residuals	标准误差（residuals = (observed-expected) / sqrt(expected)）

- 数据展示

f0	observed	expected	residuals
0	2.0	1.5	0.4082482904638631
1	3.0	3.5	-0.2672612419124244

3.4.8. 数据视图

通过数据视图组件，您可以可视化地了解特征与标签列的分布情况及特征的特点，以便后续进行数据分析。该组件支持稀疏和稠密数据格式。本文为您介绍PAI-Studio提供的数据视图算法的参数配置方式及使用示例。

配置组件

- 可视化方式

页签	参数	描述
字段设置	选择特征列	用来表现训练样本数据特征的列。
	选择目标列	用来进行训练样本数据的目标列。
	枚举特征	勾选的特征将被视作枚举特征处理。
	k:v，k:v稀疏数据格式	是否采用KV格式的稀疏数据。

页签	参数	描述
参数设置	连续特征离散区间数	连续性特征等距离划分最大区间数。
执行调优	计算核心数	计算的核心数，取值范围为正整数。
	每个核心内存	每个核心的内存，取值范围为1 MB~65536 MB。

- PAI命令

```
PAI
-name fe_meta_runner
-project algo_public
-DinputTable="pai_dense_10_10"
-DoutputTable="pai_temp_2263_20384_1"
-DmapTable="pai_temp_2263_20384_2"
-DselectedCols="pdays,previous,emp_var_rate,cons_price_idx,cons_conf_idx,euribor3m,nr_employed,age,campaign,poutcome"
-DlabelCol="y"
-DcategoryCols="previous"
-Dlifecycle="28"-DmaxBins="5";
```

参数名称	是否必选	描述	默认值
inputTable	是	输入表的名称。	无
inputTablePartitions	是	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区之间使用英文逗号（,）分隔。	无
outputTable	是	输出表名称。	无
mapTable	是	输出映射表，数据视图对String类字符串会做一个统计，映射成数字（转换成Int方便机器学习识别和训练）	无
selectedCols	是	输入表选择列名类型。	无
categoryCols	否	把Int或者Double字段当做枚举特征。	无
maxBins	否	连续性特征等距离划分最大区间数。	100

参数名称	是否必选	描述	默认值
isSparse	否	输入数据是否为稀疏格式，取值范围为{ture,false}。	false
itemSpliter	否	当输入表数据为稀疏格式时，KV对之间的分隔符。	英文逗号（,）
kvSpliter	否	当输入表数据为稀疏格式时，key和value之间的分隔符。	英文冒号（:）
lifecycle	否	表的声明周期。	28
coreNum	否	计算的核心数，取值范围为正整数。取值范围[1, 9999]。	系统自动分配
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	系统自动分配

3.4.9. 协方差

本文为您介绍PAI-Studio提供的协方差。

在概率论和统计学中，协方差用于衡量两个变量的总体误差。方差是协方差的一种特殊情况，即当两个变量是相同的情况。期望值分别为E(X) =μ与E(Y)=v的两个实数，其随机变量X与Y之间的协方差定义为：cov(X, Y)=E((X - μ) (Y - v))。

您可以通过以下任意一种方式，配置协方差组件参数：

- 可视化方式

页签	参数	描述
字段设置	**输入列**	选择输入列，只支持bigint与double类型。
进行调优	**核心数**	计算的核心数，如果未配置，则系统自动分配。
	内存数	每个核心的内存，如果未配置，则系统自动分配。单位为MB。

- PAI命令方式

```
PAI -name cov
  -project algo_public
  -DinputTableName=maple_test_cov_basic12x10_input
  -DoutputTableName=maple_test_cov_basic12x10_output
  -DcoreNum=6
  -DmemSizePerCore=110;
```

参数名称	是否必须	描述	默认值
inputTableName	是	输入表的名称。	无

参数名称	是否必须	描述	默认值
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区之间使用英文逗号（,）分隔。	输入表的所有分区
outputTableName	是	输出表名称。	无
selectedColNames	否	输入表选择列名类型。	选择全部列
lifecycle	否	指定输出表的生命周期。	无
coreNum	否	计算的核心数，取值范围为正整数。取值范围[1, 9999]。	默认自动计算
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	默认自动计算

3.4.10. 经验概率密度图

本文为您介绍PAI-Studio提供的经验概率密度图。

经验概率密度图采用经验分布和内核分布两种算法：

- 经验分布

  当无法得到精确的参数分布时，需要从数据中估计概率分布从而得到非参数分布。更详细介绍请参见经验分布。

- 内核分布

  算法中采用内核分布估计样本数据的概率密度，与直方图类似，都是代表样本数据的分布情况。区别是内核分布通过叠加各部分数据而产生连续平滑的分布曲线，而直方图呈现的是离散地数据分布。采用内核分布时，非样本数据点的概率密度并非0，而是各样本抽样点在内核分布下的概率密度的加权叠加。当前算法的内核分布采用高斯分布。更详细介绍请参见内核分布。

配置组件

您可以通过以下任意一种方式，配置离散值特征分析组件参数：

- 可视化方式

页签	参数	描述
字段设置	输入列	选择输入列，只支持bigint与double类型。

字段设置 页签	参数	描述
	标签列	标签字段。
参数设置	计算频次区间数	值越大精度越高，会根据各列数据的取值范围进行区间划分计算区间数。
执行调优	核数目	计算的核心数，取值范围为正整数。
	内存数	每个核心的内存，取值范围为1 MB~65536 MB。

- PAI命令方式

```
PAI -name empirical_pdf
-project algo_public
-DinputTableName="test_data"
-DoutputTableName="test_epdf_out"
-DfeatureColNames="col0,col1,col2"
-DinputTablePartitions="ds='20160101'"
-Dlifecycle=1
-DintervalNum=100
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
outputTableName	是	输出表名。	无
featureColNames	是	输入表中，用于训练的特征列名。	无
labelColName	否	输入表中，标签列的名称。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区之间使用英文逗号（,）分隔。	无
intervalNum	否	计算频次区间数，越大精度越高。取值范围[1,1E14)。	无
lifecycle	否	表的生命周期。	无
coreNum	否	计算的核心数，取值范围为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	系统自动分配

示例

使用如下SQL语句，生成输入数据。

```
drop table if exists epdf_test;
create table epdf_test as
select
 *
from
(
 select 1.0 as col1 from dual
  union all
 select 2.0 as col1 from dual
  union all
 select 3.0 as col1 from dual
  union all
 select 4.0 as col1 from dual
  union all
 select 5.0 as col1 from dual
) tmp;
```

执行如下PAI命令。

```
PAI -name empirical_pdf
-project algo_public
-DinputTableName=epdf_test
-DoutputTableName=epdf_test_out
-DfeatureColNames=col1;
```

- 输入说明

  选择需要计算的列，可选择多列。同时可选择label列，按照每个label值把这些列切分成多组。例如label列中包含的值为0和1，需要计算的列会被分成label=0和label=1两组，分别画出概率密度。

  > **说明** 选择的label列不能超过100个。

- 输出说明

  图和结果表，结果表的字段如下。不设置label列时，label字段输出NULL。

列名	数据类型
colName	string
label	string
x	double
pdf	double

输出表

```
+------------+------------+------------+------------+
| colname    | label      | x          | pdf        |
```

colname	label	x	pai
col1	NULL	1.0	0.12775155176809325
col1	NULL	1.0404050505050506	0.1304256933829622
col1	NULL	1.0808101010101012	0.13306325897429525
col1	NULL	1.1212151515151518	0.1356613897616418
col1	NULL	1.1616202020202024	0.1382173796574596
col1	NULL	1.202025252525253	0.1407286844875733
col1	NULL	1.2424303030303037	0.14319293014274642
col1	NULL	1.2828353535353543	0.14560791960033242
col1	NULL	1.3232404040404049	0.14797163876379316
col1	NULL	1.3636454545454555	0.1502822610772349
col1	NULL	1.404050505050506	0.1525381508819247
col1	NULL	1.4444555555555567	0.1547378654919243
col1	NULL	1.4848606060606073	0.1568801559764068
col1	NULL	1.525265656565658	0.15896396664681753
col1	NULL	1.5656707070707085	0.16098843325768245
col1	NULL	1.6060757575757592	0.1629528799404685
col1	NULL	1.6464808080808098	0.16485681490034038
col1	NULL	1.6868858585858604	0.16669992491584543
col1	NULL	1.727290909090911	0.16848206869138338
col1	NULL	1.7676959595959616	0.17020326912168932
col1	NULL	1.8081010101010122	0.17186370453638117
col1	NULL	1.8485060606060628	0.17346369900080946
col1	NULL	1.8889111111111134	0.17500371175692428
col1	NULL	1.929316161616164	0.17648432589456017
col1	NULL	1.9697212121212146	0.17790623634938396
col1	NULL	2.0101262626262653	0.1792702373286898
col1	NULL	2.050531313131316	0.18057720927022053
col1	NULL	2.0909363636363665	0.18182810544221673
col1	NULL	2.131341414141417	0.18302393829491406
col1	NULL	2.1717464646464677	0.18416576567472337
col1	NULL	2.2121515151515183	0.1852546770123305
col1	NULL	2.252556565656569	0.18629177959496213
col1	NULL	2.2929616161616195	0.18727818503109434
col1	NULL	2.33336666666667	0.18821499601297229
col1	NULL	2.3737717171717208	0.18910329347850022
col1	NULL	2.4141767676767714	0.18994412426940221
col1	NULL	2.454581818181822	0.19073848937711185
col1	NULL	2.4949868686868726	0.19148733286168018
col1	NULL	2.535391919191923	0.1921915315221827
col1	NULL	2.575796969696974	0.19285188538972659
col1	NULL	2.6162020202020244	0.19346910910630113
col1	NULL	2.656607070707075	0.19404382424446043
col1	NULL	2.6970121212121256	0.1945765526142701
col1	NULL	2.7374171717171762	0.19506771059517916
col1	NULL	2.777822222222227	0.19551760452158667
col1	NULL	2.8182272727272775	0.19592642714194602
col1	NULL	2.858632323232328	0.1962942551623821
col1	NULL	2.8990373737373787	0.1966210478770638
col1	NULL	2.9394424242424293	0.1969066468790639
col1	NULL	2.97984747474748	0.19715077683721793
col1	NULL	3.0202525252525305	0.19735304731663747
col1	NULL	3.060657575757581	0.19751295561309964
col1	NULL	3.1010626262626317	0.19762989056457925

```
| col1 | NULL | 3.1414676767676823 | 0.19770313729675995 |
| col1 | NULL | 3.181872727272733 | 0.19773188285349683 |
| col1 | NULL | 3.2222777777777836 | 0.19771522265793107 |
| col1 | NULL | 3.262682828282834 | 0.19765216774530828 |
| col1 | NULL | 3.303087878787885 | 0.19754165270453194 |
| col1 | NULL | 3.3434929292929354 | 0.19738254426210697 |
| col1 | NULL | 3.383897979797986 | 0.19717365043938664 |
| col1 | NULL | 3.4243030303030366 | 0.19691373021193162 |
| col1 | NULL | 3.4647080808080872 | 0.1966015035982942 |
| col1 | NULL | 3.505113131313138 | 0.19623566210464843 |
| col1 | NULL | 3.5455181818181885 | 0.19581487945135703 |
| col1 | NULL | 3.585923232323239 | 0.19533782250778076 |
| col1 | NULL | 3.6263282828282897 | 0.1948031623623475 |
| col1 | NULL | 3.6667333333333403 | 0.1942095854560816 |
| col1 | NULL | 3.707138383838391 | 0.19355580470939734 |
| col1 | NULL | 3.7475434343434415 | 0.19284057057394655 |
| col1 | NULL | 3.787948484848492 | 0.19206268194364004 |
| col1 | NULL | 3.8283535353535427 | 0.19122099686158253 |
| col1 | NULL | 3.8687585858585933 | 0.19031444296253852 |
| col1 | NULL | 3.909163636363644 | 0.1893420275936375 |
| col1 | NULL | 3.9495686868686946 | 0.18830284755928747 |
| col1 | NULL | 3.989973737373745 | 0.1871960984396676 |
| col1 | NULL | 4.030378787878796 | 0.18602108343567092 |
| col1 | NULL | 4.070783838383846 | 0.18477722169674377 |
| col1 | NULL | 4.111188888888897 | 0.1834640560916829 |
| col1 | NULL | 4.151593939393948 | 0.1820812603860928 |
| col1 | NULL | 4.191998989898998 | 0.18062864579383914 |
| col1 | NULL | 4.232404040404049 | 0.179106166873458 |
| col1 | NULL | 4.272809090909099 | 0.17751392674406796 |
| col1 | NULL | 4.31321414141415 | 0.17585218159888508 |
| col1 | NULL | 4.353619191919201 | 0.17412134449794325 |
| col1 | NULL | 4.394024242424251 | 0.1723219884250765 |
| col1 | NULL | 4.434429292929302 | 0.17045484859762067 |
| col1 | NULL | 4.4748343434343525 | 0.16852082402064342 |
| col1 | NULL | 4.515239393939403 | 0.1665209782808102 |
| col1 | NULL | 4.555644444444454 | 0.16445653957824907 |
| col1 | NULL | 4.596049494949504 | 0.16232889999798905 |
| col1 | NULL | 4.636454545454555 | 0.16013961402571825 |
| col1 | NULL | 4.6768595959596055 | 0.1578903963157465 |
| col1 | NULL | 4.717264646464656 | 0.15558311872216193 |
| col1 | NULL | 4.757669696969707 | 0.1532198066072439 |
| col1 | NULL | 4.798074747474757 | 0.1508026344442397 |
| col1 | NULL | 4.838479797979808 | 0.14833392073462115 |
| col1 | NULL | 4.878884848484859 | 0.14581612226291346 |
| col1 | NULL | 4.919289898989909 | 0.1432518277151203 |
| col1 | NULL | 4.95969494949496 | 0.1406437506896507 |
| col1 | NULL | 5.00010000000001 | 0.13799472213247665 |
+------------+------------+------------+------------+
```

3.4.11. 箱线图

本文为您介绍PAI-Studio提供的箱线图。

箱形图是一种用作显示一组数据分散情况的统计图。它主要用于反映原始数据分布的特征，还可以进行多组数据分布特征的比较。

配置组件

您可以通过以下任意一种方式，配置箱线图组件参数：

- 可视化方式

参数	描述
选择连续类型特征	选择连续类型的特征。
选择枚举类型特征	选择枚举类型的特征。
分层样本采用数	分层样本的采用数。

- PAI命令

```
PAI -name box_plot -project algo_public
  -DinputTable="boxplot"
  -DcontinueCols="age"
  -DcategoryCol="y"
   -DoutputTable="pai_temp_6075_97181_1"
  -DsampleSize="1000"
  -Dlifecycle="7";
```

参数名称	是否必选	描述	默认值
inputTable	是	输入表的名称。	无
inputTablePartitions	是	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 说明 指定多个分区时，分区之间使用英文逗号（,）分隔。	无
outputTable	是	输出表名，存放箱线图和采样的样本。	无
continueCols	是	连续值特征。	无
categoryCol	是	枚举特征列。	无
sampleSize	否	绘制每个特征的扰动情况的样本采样数。	1000
lifecycle	否	输出表生命周期。单位：天。	28
coreNum	否	计算的核心数，取值范围为正整数。	系统自动分配

参数名称	是否必选	描述	默认值
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	系统自动分配

示例

- 输入数据

```
create table boxplot as select age, y from bank_data limit 100;
```

age	y
50	0
53	0
28	1
39	0
55	1
30	0
37	0
39	0
36	1
27	0
34	0
41	0
55	1
33	0
26	0
52	0
35	1
27	1
28	0
26	0
41	0

age	y
35	0
40	0
32	0
41	0
34	0
49	0
37	0
35	0
38	0
47	0
46	0
27	0
29	1
32	0
36	0
29	0
47	0
44	0
54	0
36	0
42	0
44	0
72	1
48	0
36	0
35	0

age	y
43	0
56	0
42	0
31	0
32	0
33	0
31	0
39	0
30	1
24	0
24	0
38	0
26	0
41	0
34	0
30	0
37	0
68	0
31	0
48	0
33	0
59	0
44	0
28	0
50	0
33	0

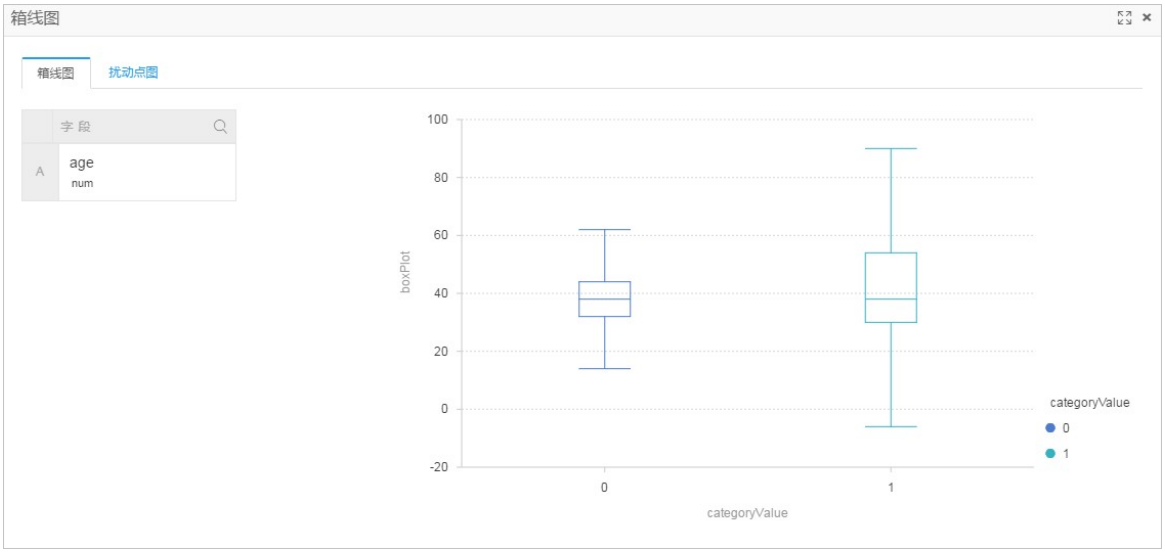
age	y
45	0
40	0
45	0
43	0
54	0
53	0
35	0
30	0
25	0
35	0
54	1
30	0
38	0
35	0
47	0
32	0
27	0
40	1
31	0
42	0
40	0
31	0
57	0
38	1
39	0
37	0

age	y
44	0

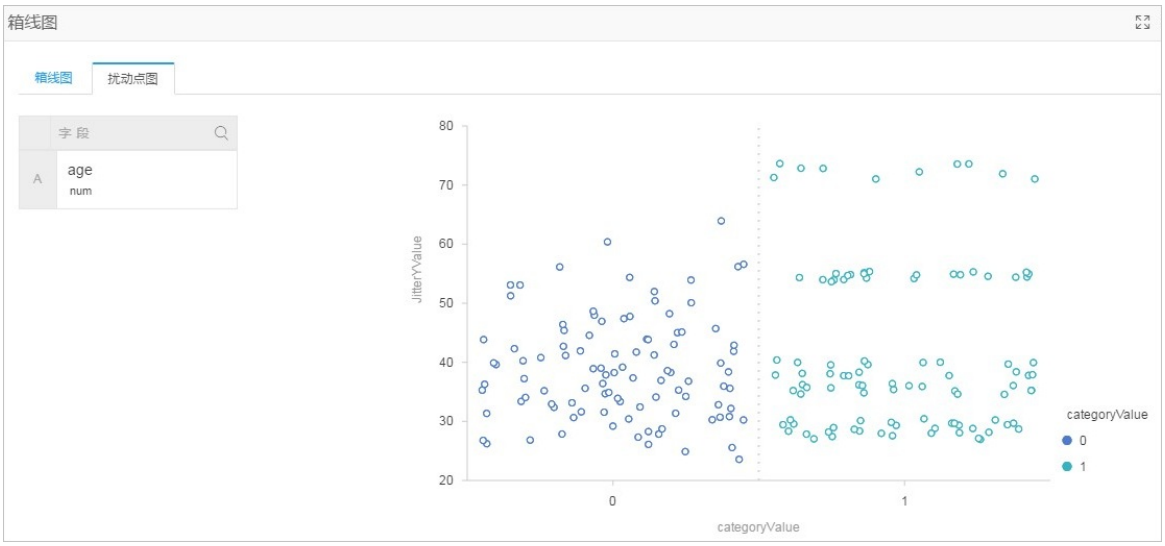
- 参数配置

  选择age为连续类型特征，y为枚举类特征，其他保持默认值。

- 运行效果
  - 箱线图分布情况如下图所示。



  - 扰动点图分布情况如下图所示。



3.4.12. 散点图

本文为您介绍PAI-Studio提供的散点图。

散点图是指在回归分析中，数据点在直角坐标系平面上的分布图。

配置组件

您可以通过以下任意一种方式，配置散点图组件参数：

- 可视化方式

参数	描述
选择特征列	选择用来表现训练样本数据特征的列。
分类标签列	标签字段。
抽样样本数	抽样的样本数量。

- PAI命令方式

```
PAI -name scatter_diagram -project algo_public
 -DselectedCols=emp_var_rate,cons_price_rate,cons_conf_idx,euribor3m
 -DlabelCol=y
 -DmapTable=pai_temp_2447_22859_2
 -DinputTable=scatter_diagram
 -DoutputTable=pai_temp_2447_22859_1;
```

参数名称	是否必选	描述	默认值
inputTable	是	输入表的名称。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 说明 指定多个分区时，分区之间使用英文逗号（,）分隔。	无
outputTable	是	输出表名称。	无
mapTable	是	输出信息表，存放每个特征的最小值、最大值和枚举取值等。	无
selectedCols	是	选择列名类型，用于绘制两两特征之间的散点图，最多勾选5个特征。	无
labelCol	是	把Int或者String字段当做枚举标签列。	空
lifecycle	是	输出表生命周期。单位：天。	28

示例

- 输入数据

```
create table scatter_diagram as select emp_var_rate,cons_price_rate, cons_conf_idx,euribor3m,y from pai_bank_data limit 10
```

emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
1.4	93.918	-42.7	4.962	0
-0.1	93.2	-42.0	4.021	0
-1.7	94.055	-39.8	0.729	1
-1.8	93.075	-47.1	1.405	0
-2.9	92.201	31.4	0.869	1
1.4	93.918	-42.7	4.961	0
-1.8	92.893	-46.2	1.327	0
-1.8	92.893	92.893	1.313	0
-2.9	92.963	-40.8	1.266	1
-1.8	93.075	-47.1	1.41	0
1.1	93.994	-36.4	4.864	0
1.4	93.444	-36.1	4.964	0
1.4	93.444	-36.1	4.965	1
-1.8	92.893	-46.2	1.291	0
1.4	94.465	-41.8	4.96	0
1.4	93.918	-42.7	4.962	0
-1.8	93.075	-47.1	1.365	1
-0.1	93.798	-40.4	4.86	1
1.1	93.994	-36.4	4.86	0
1.4	93.918	-42.7	4.96	0
-1.8	93.075	-47.1	1.405	0
1.4	94.465	-41.8	4.967	0
1.4	93.918	-42.7	4.963	0
1.4	93.918	-42.7	4.968	0
1.4	93.918	-42.7	4.962	0
-1.8	92.893	-46.2	1.344	0

emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
-3.4	92.431	-26.9	0.754	0
-1.8	93.075	-47.1	1.365	0
-1.8	92.893	-46.2	1.313	0
1.4	93.918	-42.7	4.961	0
1.4	94.465	-41.8	4.961	0
-1.8	92.893	-46.2	1.327	0
-1.8	92.893	-46.2	1.299	0
-2.9	92.963	-40.8	1.268	1
1.4	93.918	-42.7	4.963	0
-1.8	92.893	-46.2	1.334	0
1.4	93.918	-42.7	4.96	0
-1.8	93.075	-47.1	1.405	0
1.4	94.465	-41.8	4.96	0
1.4	93.444	-36.1	4.962	0
1.1	93.994	-36.4	4.86	0
1.1	93.994	-36.4	4.857	0
1.4	93.918	-42.7	4.961	0
-3.4	92.649	-30.1	0.715	1
1.4	93.444	-36.1	4.966	0
-0.1	93.2	-42.0	4.076	0
1.4	93.444	-36.1	4.965	0
-1.8	92.893	-46.2	1.354	0
1.4	93.444	-36.1	4.967	0
1.4	94.465	-41.8	4.959	0
-1.8	92.893	-46.2	1.354	0
1.4	94.465	-41.8	4.958	0

emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
-1.8	92.893	-46.2	1.354	0
1.4	94.465	-41.8	4.864	0
1.1	93.994	-36.4	4.859	0
1.1	93.994	-36.4	4.857	0
-1.8	92.893	-46.2	1.27	0
1.1	93.994	-36.4	4.857	0
1.1	93.994	-36.4	4.859	0
1.4	94.465	-41.8	4.959	0
1.1	93.994	-36.4	4.856	0
-1.8	93.075	-47.1	1.405	0
-1.8	92.843	-50.0	1.811	1
-0.1	93.2	-42.0	4.021	0
-2.9	92.469	-33.6	1.029	0
1.4	93.918	-42.7	4.962	0
-1.8	93.075	-47.1	1.365	0
1.1	93.994	-36.4	4.857	0
-1.8	92.893	-46.2	1.259	0
1.1	93.994	-36.4	4.857	0
1.4	94.465	-41.8	4.866	0
-2.9	92.201	-31.4	0.883	0
-0.1	93.2	-42.0	4.076	0
1.1	93.994	-36.4	4.857	0
1.4	93.918	-42.7	4.96	0
1.4	93.444	-36.1	4.962	0
1.1	93.994	-36.4	4.858	0
1.1	93.994	-36.4	4.857	0

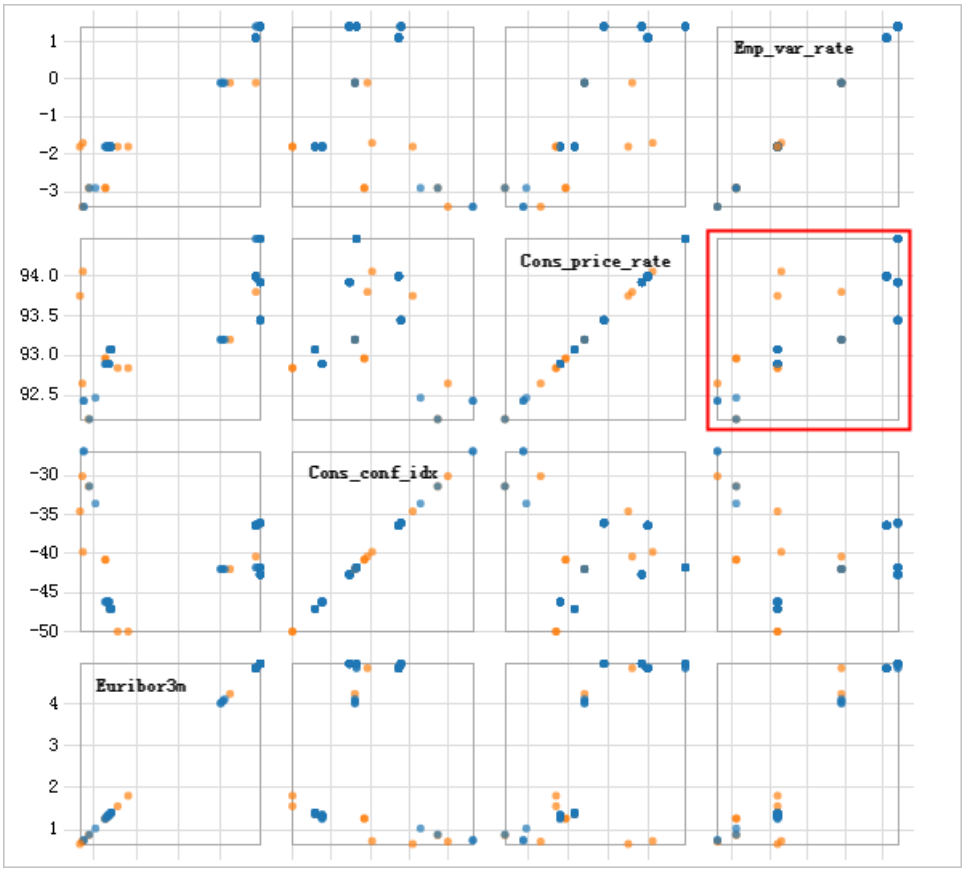
emp_var_rate	cons_price_rate	cons_conf_idx	euribor3m	y
1.1	93.994	-36.4	4.856	0
1.4	93.918	-42.7	4.968	0
1.4	93.444	-36.1	4.966	0
1.4	94.465	-41.8	4.962	0
1.4	93.444	-36.1	4.963	0
-1.8	92.843	-50.0	1.56	1
1.4	93.918	-42.7	4.96	0
1.4	93.444	-36.1	4.963	0
-3.4	92.431	-26.9	0.74	0
1.1	93.994	-36.4	4.856	0
1.4	93.918	-42.7	4.962	0
1.1	93.994	-36.4	4.856	0
-0.1	93.2	-42.0	4.245	1
1.1	93.994	-36.4	4.857	0
-1.8	93.075	-47.1	1.405	0
-1.8	92.893	-46.2	1.327	0
-0.1	93.2	-42.0	4.12	0
1.4	94.465	-41.8	4.958	0
-1.8	93.749	-34.6	0.659	1
1.1	93.994	-36.4	4.858	0
1.1	93.994	-36.4	4.858	0
1.4	93.444	-36.1	4.963	0

- 参数配置

  选择y为散点图可选的标签列，选择select emp_var_rate，cons_price_rate，cons_conf_idx，euribor3m为特征列。

- 运行结果

  直观的展示了特征与特征之间分类标签的分布情况。



3.4.13. 相关系数矩阵

本文为您介绍PAI-Studio提供的相关系数矩阵。

相关系数算法用于计算一个矩阵中每列之间的相关系数，取值范围为[-1,1]。系统计算时，count数按两列间同时非空的元素个数计算，两两列之间可能不同。

配置组件

您可以通过以下任意一种方式，配置相关系数矩阵组件参数：

- 可视化方式

页签	参数	描述
字段设置	默认全选	无
执行调优	核心数	与内存数同时设置后，该参数才生效。
	内存数	与核心数同时设置后，该参数才生效。

- PAI命令方式

```
PAI -name corrcoef
  -project algo_public
  -DinputTableName=maple_test_corrcoef_basic12x10_input
  -DoutputTableName=maple_test_corrcoef_basic12x10_output
  -DcoreNum=1
  -DmemSizePerCore=110;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 说明 指定多个分区时，分区之间使用英文逗号（,）分隔。	无
outputTableName	是	输出表名称列表。	无
selectedColNames	否	输入表选择列名类型。	默认选择全部列
lifecycle	否	指定输出表的生命周期。	无
coreNum	否	与参数memSizePerCore配对使用，正整数。范围为[1, 9999]。	默认自动计算
memSizePerCore	否	单个节点内存大小，单位MB。正整数，范围为[1024, 64*1024]。	默认自动计算

示例

- 数据生成

col0:double	col1:bigint	col2:double	col3:bigint	col4:double	col5:bigint	col6:double	col7:bigint	col8:double	col9:double
19	95	33	52	115	43	32	98	76	40
114	26	101	69	56	59	116	23	109	105
103	89	7	9	65	118	73	50	55	81
79	20	63	71	5	24	77	31	21	75
87	16	66	47	25	14	42	99	108	57

col0:double	col1:bigint	col2:double	col3:bigint	col4:double	col5:bigint	col6:double	col7:bigint	col8:double	col9:double
11	104	38	37	106	51	3	91	80	97
84	30	70	46	8	6	94	22	45	48
35	17	107	64	10	112	53	34	90	96
13	61	39	1	29	117	112	2	82	28
62	4	102	88	100	36	67	54	12	85
49	27	44	93	68	110	60	72	86	58
92	119	0	113	41	15	74	83	18	111

- PAI命令

```
PAI -name corrcoef
  -project algo_public
  -DinputTableName=maple_test_corrcoef_basic12x10_input
  -DoutputTableName=maple_test_corrcoef_basic12x10_output
  -DcoreNum=1
  -DmemSizePerCore=110;
```

- 运行结果

columnsnames	col0	col1	col2	col3	col4	col5	col6	col7	col8	col9
col0	1	-0.21156572518207 24	0.05983062597065 61	0.25999035706846 93	-0.34832491882255 86	-0.28716254396809 926	0.47880162127435 116	-0.13646519484213 326	-0.19500158764680 092	0.38973902409490 85
col1	-0.21156572518207 24	1	-0.84444773778985 85	-0.17507636221594 533	0.40943384150571 377	0.09135976026101 403	-0.30185063746265 74	0.40733726912808 044	-0.11827739124590 071	0.12433851389455 183
col2	0.05983062597065 61	-0.84444773778985 85	1	0.18518346647293 102	-0.20934839228057 014	-0.18964175123896 59	0.17993774988632 13	-0.38588856764699 48	0.20254569203773 892	0.13476160753756 655

columnsnames	col0	col1	col2	col3	col4	col5	col6	col7	col8	col9
col3	0.25999035706846 93	-0.17507636221594533	0.18518346647293102	1	0.03988018649854009	-0.43737887418329147	-0.05381829642526 7184	0.29008564415869 86	-0.36075479100756 88	0.49120190749304 49
col4	-0.34832491882255 86	0.40943384150571377	-0.20934839228057014	0.03988018649854009	1	0.14656052092468 75	-0.50160303643479 55	0.54960243257111 17	0.01374325611539 4122	0.07497231559184887
col5	-0.28716254396809926	0.09135976026101403	-0.18964175123896 59	-0.43737887418329147	0.14656052092468 75	1	0.16729809310873522	-0.29890655828796964	0.36185181010146 17	-0.17139609572868 85
col6	0.47880162127435116	-0.30185063746265 74	0.17993774988632 13	-0.05381829642526 7184	-0.50160303643479 55	0.16729809310873522	1	-0.81650198801564 62	-0.11173420918721436	-0.10363860378347944
col7	-0.13646519484213326	0.40733726912808044	-0.38588856764699 48	0.29008564415869 86	0.54960243257111 17	-0.29890655828796964	-0.81650198801564 62	1	0.07435907471544469	0.11711976051999162
col8	-0.19500158764680092	-0.11827739124590071	0.20254569203773892	-0.36075479100756 88	0.01374325611539 4122	0.36185181010146 17	-0.11173420918721436	0.07435907471544469	1	-0.18463012549540175
col9	0.38973902409490 85	0.12433851389455183	0.13476160753756655	0.49120190749304 49	0.07497231559184887	-0.17139609572868 85	-0.10363860378347944	0.11711976051999162	-0.18463012549540175	1

3.4.14. 正态检验

正态性检验通过观测值判断总体是否服从正态分布，是统计判决中重要的一种特殊的拟合优度假设检验。本文为您介绍PAI-Studio提供的正态检验。

正态检验组件由Anderson-Darling Test、Kolmogorov-Smirnov Test和QQ图检验方法组成，您可以选择一种或多种检验方法。

- Anderson-Darling Test是将样本数据的经验累积分布函数与假设数据呈正态分布时期望的分布进行比较。

如果实测差异足够大，该检验将否定总体呈正态分布的假设。

- Kolmogorov-Smirnov是比较两个观测值分布的检验方法。
- QQ图通过把测试样本数据的分位数与已知分布相比较，从而来检验数据的分布情况。在样本量 > 1000时，系统会采样进行计算并输出QQ图，因此图中的数据点不一定覆盖所有样本。

配置组件

您可以通过以下任意一种方式，配置正态检验参数：

- 可视化方式

页签	参数	描述
字段设置	选择字段列	无
参数设置	Anderson-Darling检验	取值为： ○ 是 ○ 否 默认值为是。
	Kolmogorov-Smirnov检验	取值为： ○ 是 ○ 否 默认值为是。
	使用QQ图	取值为： ○ 是 ○ 否 默认值为是。
执行调优	计算的核心数	计算的核心数，取值为正整数。
	每个核心的内存（MB）	每个核心的内存。

- PAI命令方式

```
PAI -name normality_test
  -project algo_public
  -DinputTableName=test
  -DoutputTableName=test_out
  -DselectedColNames=col1,col2
  -Dlifecycle=1;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
outputTableName	是	输出表名称列表。	无

参数名称	是否必选	描述	默认值
selectedColNames	否	输入表选择列名类型。支持选择多列，类型为DOUBLE或BIGINT。	无
inputTablePartitions	否	输入表分区名称。	””
enableQQplot	否	使用QQ图检验。取值为true或false。	ture
enableADtest	否	使用Anderson-Darling检验。取值为true或false。	ture
enableKStest	否	使用Kolmogorov-Smirnov检验。取值为true或false。	ture
lifecycle	否	指定输出表的生命周期。取值为整数，且≥-1。	-1
coreNum	否	与参数memSizePerCore配对使用，取值为正整数。系统会根据输入数据量计算所起instance的数量。	-1
memSizePerCore	否	单个节点内存大小，单位MB。取值为正整数，范围为(100, 64*1024)。系统会根据输入数据量计算所需内存大小。	-1

示例

- 数据生成

```
drop table if exists normality_test_input;
create table normality_test_input as
select
 *
from
(
 select 1 as x from dual
  union all
 select 2 as x from dual
  union all
 select 3 as x from dual
  union all
 select 4 as x from dual
  union all
 select 5 as x from dual
  union all
 select 6 as x from dual
  union all
 select 7 as x from dual
  union all
 select 8 as x from dual
  union all
 select 9 as x from dual
  union all
 select 10 as x from dual
) tmp;
```

- PAI命令

```
PAI -name normality_test
  -project algo_public
  -DinputTableName=normality_test_input
  -DoutputTableName=normality_test_output
  -DselectedColNames=x
  -Dlifecycle=1;
```

- 输入说明

  输入格式：选择需要计算的列，支持选择多列。类型为DOUBLE或BIGINT。

- 输出说明

  输出格式：图和结果表，结果表的字段如下。结果表有两个分区：

  - `p=test` 的分区是Anderson-Darling检验或Kolmogorov-Smirnov检验的结果。当 `enableADtest` 为true或 `enableKStest` 为true时会输出数据。
  - `p=plot` 是QQ图的数据，当 `enableQQplot` 为true时会输出数据，并复用 `p=test` 的列。即当 `p=plot` 时，testvalue列记录原观测数据（QQ图的x轴），pvalue列记录如果服从正态分布的预期数据（QQ图的y轴）。

列名	数据类型	含义
colName	STRING	列名
testname	STRING	检验名

列名	数据类型	含义
testvalue	DOUBLE	检验值/QQ图x轴
pvalue	DOUBLE	检验的p值/QQ图y轴
p	DOUBLE	分区名

输出表如下。

```
+------------+------------+------------+------------+------------+
| colname    | testname   | testvalue  | pvalue     | p          |
+------------+------------+------------+------------+------------+
| x          | NULL       | 1.0        | 0.8173291742279805 | plot       |
| x          | NULL       | 2.0        | 2.470864450785345  | plot       |
| x          | NULL       | 3.0        | 3.5156067948020056 | plot       |
| x          | NULL       | 4.0        | 4.3632330349313095 | plot       |
| x          | NULL       | 5.0        | 5.128868067945126  | plot       |
| x          | NULL       | 6.0        | 5.871131932054874  | plot       |
| x          | NULL       | 7.0        | 6.6367669650686905 | plot       |
| x          | NULL       | 8.0        | 7.4843932051979944 | plot       |
| x          | NULL       | 9.0        | 8.529135549214654  | plot       |
| x          | NULL       | 10.0       | 10.182670825772018 | plot       |
| x          | Anderson_Darling_Test | 0.1411092332197832 | 0.9566579606430077 | test       |
| x          | Kolmogorov_Smirnov_Test | 0.09551932503797644 | 0.9999888659426232 | test       |
+------------+------------+------------+------------+------------+
```

3.4.15. 洛伦兹曲线

通过洛伦兹曲线，您可以直观地看到一个国家或地区收入分配状况。洛伦兹曲线的弯曲程度反映了收入分配的不平等程度。弯曲程度越大，收入分配越不平等。

画一个矩形，矩形的高用于衡量社会财富的百分比，将之分为N等份，每一等分为1/N的社会总财富。在矩形的长上，将所有家庭从最贫者到最富者自左向右排列，也分为N等分，第一个等份代表收入最低的1/N的家庭。在这个矩形中，将每1/N的家庭所有拥有的财富的占比累积起来，并将相应的点画在图中，便得到了一条曲线就是洛伦兹曲线。

配置组件

您可以通过以下任意一种方式，配置洛伦兹曲线参数：

- 可视化方式

页签	参数	描述
字段设置	选择字段列	无
参数设置	分位数	默认值为100。
执行调优	计算的核心数	计算的核心数，取值为正整数。
	每个核心的内存（MB）	每个核心的内存。

- PAI命令方式

```
PAI -name LorenzCurve
  -project algo_public
  -DinputTableName=maple_test_lorenz_basic10_input
  -DcolName=col0
  -DoutputTableName=maple_test_lorenz_basic10_output -DcoreNum=20
  -DmemSizePerCore=110;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
outputTableName	是	输出表名称列表。	无
ColName	否	输入表选择列名称。支持选择多列，列之间使用英文逗号（,）分隔。	无
N	否	分位数。	100
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ◦ Partition_name=value ◦ name1=value1/name2=value2：多级分区 说明 指定多个分区时，分区之间使用英文逗号（,）分隔。	无
lifecycle	否	指定输出表的生命周期。取值为整数。单位：天。	28
coreNum	否	与参数memSizePerCore配对使用，取值为正整数。系统会根据输入数据量计算所起instance的数量。	默认自动计算
memSizePerCore	否	单个节点内存大小，单位MB。取值为正整数，范围为(1024, 64*1024)。	默认自动计算

示例

1. 生成如下测试数据。

col0:double
4
7
2

col0:double
8
6
3
9
5
0
1
10

2. 执行如下PAI命令。

```
PAI -name LorenzCurve
  -project algo_public
  -DinputTableName=maple_test_lorenz_basic10_input
  -DcolName=col0
  -DoutputTableName=maple_test_lorenz_basic10_output
  -DcoreNum=20
  -DmemSizePerCore=110;
```

3. 查看输出结果，如下表所示。

quantile	col0
0	0
1	0.01818181818181818
2	0.01818181818181818
3	0.01818181818181818
4	0.01818181818181818
5	0.01818181818181818
6	0.01818181818181818
7	0.01818181818181818
8	0.01818181818181818
9	0.01818181818181818
10	0.01818181818181818

quantile	col0
11	0.05454545454545454
12	0.05454545454545454
13	0.05454545454545454
14	0.05454545454545454
...	...
85	0.8181818181818182
86	0.8181818181818182
87	0.8181818181818182
88	0.8181818181818182
89	0.8181818181818182
90	1
91	1
92	1
93	1
94	1
95	1
96	1
97	1
98	1
99	1
100	1

3.5. 机器学习

3.5.1. 二分类

本文为您介绍PAI-Studio提供的二分类算法，包括线性支持向量机、逻辑回归二分类、GBDT二分类、PS-SMART二分类及PS逻辑回归二分类。

线性支持向量机

支持向量机SVM（Support Vector Machine）是基于统计学习理论的一种机器学习方法，通过寻求结构风险最小化，提高学习机泛化能力，从而实现经验风险和置信范围最小化。该线性支持向量机算法不通过核函数方式实现，具体实现理论请参见算法原理中的Trust Region Method for L2-SVM部分。该算法仅支持二分类场景。

您可以通过以下任意一种方式，配置线性支持向量机组件参数：

- 可视化方式

页签	参数	描述
字段设置	**特征列**	输入列，支持BIGINT和DOUBLE类型。
	标签列	支持BIGINT、DOUBLE及STRING类型。
参数设置	**正样本的标签值**	目标基准值。如果未指定，则系统随机选定。如果正负例样本差异大，建议手动指定。
	正例惩罚因子	正例权重值。默认值为1.0，取值范围为(0, +∞)。
	负例惩罚因子	负例权重值。默认值为1.0，取值范围为(0, +∞)。
	收敛系数	收敛误差。默认值为0.001，取值范围为(0, 1)。
执行调优	**计算的核心数**	如果未配置，则系统自动分配。
	每个核心的内存	如果未配置，则系统自动分配。单位为MB。

- PAI命令方式

```
PAI -name LinearSVM -project algo_public
  -DinputTableName="bank_data"
  -DmodelName="xlab_m_LinearSVM_6143"
  -DfeatureColNames="pdays,emp_var_rate,cons_conf_idx"
  -DlabelColName="y"
  -DpositiveLabel="0";
  -DpositiveCost="1.0"
  -DnegativeCost="1.0"
  -Depsilon="0.001"
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
inputTableParitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区之间使用英文逗号（,）分隔。	输入表的所有分区

参数名称	是否必选	描述	默认值
modelName	是	输出的模型名称。	无
featureColNames	是	输入表中，用于训练的特征列名。	无
labelColName	是	输入表中，标签列的名称。	无
positiveLabel	否	正例的值。	从label取值中，随机选择一个。
positiveCost	否	正例权重值，即正例惩罚因子。取值范围为(0, +∞)。	1.0
negativeCost	否	负例权重值，即负例惩罚因子。取值范围为(0, +∞)。	1.0
epsilon	否	收敛系数，取值范围为(0,1)。	0.001
enableSparse	否	输入数据是否为稀疏格式，取值范围为{ture,false}。	false
itemDelimiter	否	当输入表数据为稀疏格式时，KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	当输入表数据为稀疏格式时，key和value之间的分隔符。	英文冒号（:）
coreNum	否	计算的核心数，取值范围为正整数。	系统自动分配
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	系统自动分配

线性支持向量机示例

1. 导入如下训练数据。

id	y	f0	f1	f2	f3	f4	f5	f6	f7
1	-1	-0.294118	0.487437	0.180328	-0.292929	-1	0.00149028	-0.53117	-0.0333333
2	+1	-0.882353	-0.145729	0.0819672	-0.414141	-1	-0.207153	-0.766866	-0.666667
3	-1	-0.0588235	0.839196	0.0491803	-1	-1	-0.305514	-0.492741	-0.633333
4	+1	-0.882353	-0.105528	0.0819672	-0.535354	-0.777778	-0.162444	-0.923997	-1

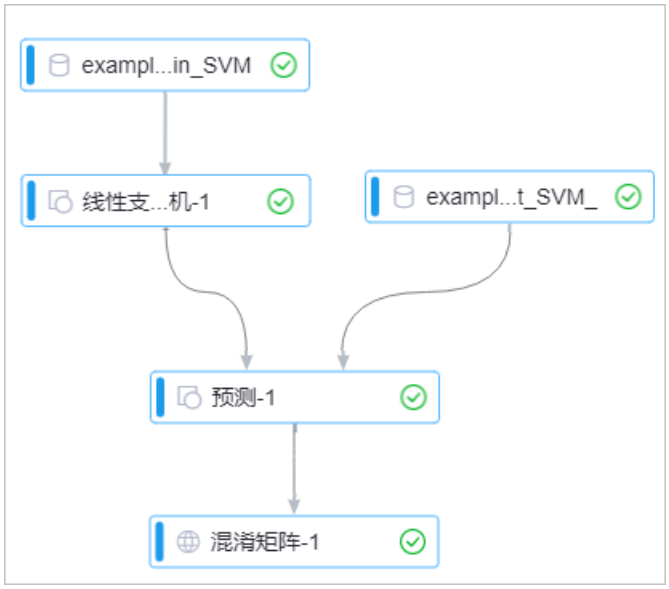
id	y	f0	f1	f2	f3	f4	f5	f6	f7
5	-1	-1	0.376884	-0.344262	-0.292929	-0.602837	0.28465	0.887276	-0.6
6	+1	-0.411765	0.165829	0.213115	-1	-1	-0.23696	-0.894962	-0.7
7	-1	-0.647059	-0.21608	-0.180328	-0.353535	-0.791962	-0.0760059	-0.854825	-0.833333
8	+1	0.176471	0.155779	-1	-1	-1	0.052161	-0.952178	-0.733333
9	-1	-0.764706	0.979899	0.147541	-0.0909091	0.283688	-0.0909091	-0.931682	0.0666667
10	-1	-0.0588235	0.256281	0.57377	-1	-1	-1	-0.868488	0.1

2. 导入如下测试数据。

id	y	f0	f1	f2	f3	f4	f5	f6	f7
1	+1	-0.882353	0.0854271	0.442623	-0.616162	-1	-0.19225	-0.725021	-0.9
2	+1	-0.294118	-0.0351759	-1	-1	-1	-0.293592	-0.904355	-0.766667
3	+1	-0.882353	0.246231	0.213115	-0.272727	-1	-0.171386	-0.981213	-0.7
4	-1	-0.176471	0.507538	0.278689	-0.414141	-0.702128	0.0491804	-0.475662	0.1
5	-1	-0.529412	0.839196	-1	-1	-1	-0.153502	-0.885568	-0.5
6	+1	-0.882353	0.246231	-0.0163934	-0.353535	-1	0.0670641	-0.627669	-1

id	y	f0	f1	f2	f3	f4	f5	f6	f7
7	-1	-0.882353	0.819095	0.278689	-0.151515	-0.307329	0.19225	0.00768574	-0.966667
8	+1	-0.882353	-0.0753769	0.0163934	-0.494949	-0.903073	-0.418778	-0.654996	-0.866667
9	+1	-1	0.527638	0.344262	-0.212121	-0.356974	0.23696	-0.836038	-0.8
10	+1	-0.882353	0.115578	0.0163934	-0.737374	-0.56974	-0.28465	-0.948762	-0.933333

3. 创建如下实验，详情请参见算法建模。



4. 配置线性支持向量机组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**特征列**	选择f0、f1、f2、f3、f4、f5、f6及f7列。
	标签列	选择y列。

5. 运行实验，查看预测结果。

id	y	prediction_result	prediction_score	prediction_detail
1	+1	+1	0.157594271402295...	{"+1": 0.1575942714022959, "-1": -0.1575942714022959}
2	+1	+1	0.6614698832250897	{"+1": 0.6614698832250897, "-1": -0.6614698832250897}
3	+1	-1	-0.033364749587674...	{"+1": -0.03336474958767432, "-1": 0.03336474958767432}
4	-1	-1	-0.7993936496277533	{"+1": -0.7993936496277533, "-1": 0.7993936496277533}
5	-1	-1	-0.062511633256538...	{"+1": -0.06251163325653875, "-1": 0.06251163325653875}
6	+1	+1	0.099662488068409...	{"+1": 0.09966248806840972, "-1": -0.09966248806840972}
7	-1	-1	-0.7519810414882623	{"+1": -0.7519810414882623, "-1": 0.7519810414882623}
8	+1	+1	0.180293970573986...	{"+1": 0.1802939705739862, "-1": -0.1802939705739862}
9	+1	-1	-0.1196222721567562	{"+1": -0.1196222721567562, "-1": 0.1196222721567562}
10	+1	+1	0.4110490612942082	{"+1": 0.4110490612942082, "-1": -0.4110490612942082}

逻辑回归二分类

逻辑回归二分类组件支持稀疏及稠密数据格式。PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	训练特征列	输入数据源中，用于训练的特征列。支持DOUBLE及BIGINT类型。 说明 特征数量不能超过两千万。
	目标列	输入数据源中，目标列名称。
	正类值	无。
	是否稀疏数据	输入数据是否为稀疏格式。
参数设置	正则项	支持None、L1及L2类型。
	最大迭代次数	默认值为100。
	正则系数	如果 正则项为None，则该参数失效。
	最小收敛误差	默认值为0.000001。
执行调优	核数目	系统自动分配。
	每个核内存数	系统自动分配。

- PAI命令方式

```
PAI -name logisticregression_binary
  -project algo_public
  -DmodelName="xlab_m_logistic_regression_6096"
  -DregularizedLevel="1"
  -DmaxIter="100"
  -DregularizedType="l1"
  -Depsilon="0.000001"
  -DlabelColName="y"
  -DfeatureColNames="pdays,emp_var_rate"
  -DgoodValue="1"
  -DinputTableName="bank_data"
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无
featureColNames	否	输入表中，用于训练的特征列名。 **说明** 特征数量不能超过两千万。	所有数值列
labelColName	是	输入表的标签列名。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持的格式包括： ○ partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区间使用英文逗号（,）分隔。	全表
modelName	是	输出的模型名。	无
regularizedType	否	正则化类型，取值范围为{'l1','l2','None'}。	l1
regularizedLevel	否	正则化系数。如果 regularizedType为None，则该参数失效。	1.0
maxIter	否	L-BFGS的最大迭代次数。	100
epsilon	否	收敛误差。该参数是L-BFGS算法的终止条件，即两次迭代的log-likelihood之差小于该值，迭代终止。	1.0e-06
goodValue	否	目标基准值。二分类时，指定训练系数针对的label值。如果为空，则系统随机分配。	无
enableSparse	否	输入数据是否为稀疏格式，取值范围为{true,false}。	false

参数名称	是否必选	参数描述	默认值
itemDelimiter	否	输入表数据为稀疏格式时，KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	输入表数据为稀疏格式时，key和value之间的分隔符。	英文冒号（:）
coreNum	否	核心数量。	系统自动分配
memSizePerCore	否	单个核心使用的内存数，单位为MB。	系统自动分配

PAI使用KV格式表示稀疏数据，如下表所示。其中itemDelimiter表示KV对之间的分隔符，kvDelimiter表示key和value之间的分隔符。

key_value
1:100,4:200,5:300
1:10,2:20,3:30

> **说明** KV格式使用索引（从0开始）表示key。如果使用字符表示key，则系统报错。

逻辑回归二分类示例

1. 使用SQL语句，生成训练数据。

```
drop table if exists lr_test_input;
create table lr_test_input
as
select
  *
from
(
select
    cast(1 as double) as f0,
    cast(0 as double) as f1,
    cast(0 as double) as f2,
    cast(0 as double) as f3,
    cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(1 as double) as f1,
      cast(0 as double) as f2,
      cast(0 as double) as f3,
      cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(0 as double) as f1
```

```
        cast(0 as double) as f1,
        cast(1 as double) as f2,
        cast(0 as double) as f3,
        cast(1 as bigint) as label
    from dual
    union all
      select
        cast(0 as double) as f0,
        cast(0 as double) as f1,
        cast(0 as double) as f2,
        cast(1 as double) as f3,
        cast(1 as bigint) as label
    from dual
    union all
      select
        cast(1 as double) as f0,
        cast(0 as double) as f1,
        cast(0 as double) as f2,
        cast(0 as double) as f3,
        cast(0 as bigint) as label
    from dual
    union all
      select
        cast(0 as double) as f0,
        cast(1 as double) as f1,
        cast(0 as double) as f2,
        cast(0 as double) as f3,
        cast(0 as bigint) as label
    from dual
) a;
```

生成的训练数据表lr_test_input如下。

f0	f1	f2	f3	label
1.0	0.0	0.0	0.0	0
0.0	0.0	1.0	0.0	1
0.0	0.0	0.0	1.0	1
0.0	1.0	0.0	0.0	0
1.0	0.0	0.0	0.0	0
0.0	1.0	0.0	0.0	0

2. 使用PAI命令，提交逻辑回归二分类组件的训练参数。

```
drop offlinemodel if exists lr_test_model;
PAI -name logisticregression_binary
  -project algo_public
  -DmodelName="lr_test_model"
  -DitemDelimiter=","
  -DregularizedLevel="1"
  -DmaxIter="100"
  -DregularizedType="None"
  -Depsilon="0.000001"
  -DkvDelimiter=":"
  -DlabelColName="label"
  -DfeatureColNames="f0,f1,f2,f3"
  -DenableSparse="false"
  -DgoodValue="1"
  -DinputTableName="lr_test_input";
```

3. 使用PAI命令，提交预测组件参数。

```
drop table if exists lr_test_prediction_result;
PAI -name prediction
  -project algo_public
  -DdetailColName="prediction_detail"
  -DmodelName="lr_test_model"
  -DitemDelimiter=","
  -DresultColName="prediction_result"
  -Dlifecycle="28"
  -DoutputTableName="lr_test_prediction_result"
  -DscoreColName="prediction_score"
  -DkvDelimiter=":"
  -DinputTableName="lr_test_input"
  -DenableSparse="false"
  -DappendColNames="label";
```

4. 查看预测结果表lr_test_prediction_result。

label	prediction_result	prediction_score	prediction_detail
0	0	0.9999998793434426	{“0”：0.9999998793434426，“1”：1.206565574533681e-07}
1	1	0.9999997995741735	{“0”：2.004258650156743e-07，“1”：0.999999799574135}
1	1	0.9999997995741735	{“0”：2.004258650156743e-07，“1”：0.999999799574135}
0	0	0.9999998793434426	{“0”：0.9999998793434426，“1”：1.206565574533681e-07}
0	0	0.9999998793434426	{“0”：0.9999998793434426，“1”：1.206565574533681e-07}
0	0	0.9999998793434426	{“0”：0.9999998793434426，“1”：1.206565574533681e-07}

GBDT二分类

GBDT（Gradient Boosting Decesion Tree）二分类算法的原理是设置阈值，如果特征值大于阈值，则为正例，反之为负例。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	选择特征列	输入数据源中，参与训练的特征列。支持DOUBLE及BIGINT类型。 说明 特征列数量不能超过800。
	选择标签列	仅支持BIGINT类型。
	选择分组列	支持DOUBLE及BIGINT类型，默认将全表作为一组。
参数设置	metric类型	支持NDCG及DCG类型。
	树的数目	取值范围为1~10000。
	学习速率	取值范围为(0,1)。
	训练采集样本比例	取值范围为(0,1)。
	训练采集特性比例	取值范围为(0,1)。
	最大叶子数	取值范围为1~1000。
	测试数据比例	取值范围为[0,1)。
	树最大深度	取值范围为1~100。
	叶子点最少样本数	取值范围为1~1000。
	随机数产生器种子	取值范围为[0,10]。
	一个特征分裂的最大数量	取值范围为1~1000。
执行调优	核数目	系统根据输入数据量，自动分配训练的实例数量。
	内存	系统根据输入数据量，自动分配内存。取值范围为1024 MB~64*1024 MB。

- PAI命令方式

```
PAI -name gbdt_lr
  -project algo_public
  -DfeatureSplitValueMaxSize="500"
  -DrandSeed="0"
  -Dshrinkage="0.5"
  -DmaxLeafCount="32"
  -DlabelColName="y"
  -DinputTableName="bank_data_partition"
  -DminLeafSampleCount="500"
  -DgroupIDColName="nr_employed"
  -DsampleRatio="0.6"
  -DmaxDepth="11"
  -DmodelName="xlab_m_GBDT_LR_21208"
  -DmetricType="2"
  -DfeatureRatio="0.6"
  -DinputTablePartitions="pt=20150501"
  -DtestRatio="0.0"
  -DfeatureColNames="age,previous,cons_conf_idx,euribor3m"
  -DtreeCount="500"
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
featureColNames	否	输入表中，用于训练的特征列名。	所有数值列
labelColName	是	输入表中的标签列名。	无
inputTablePartitions	否	输入表中，参与训练的分区。支持的格式包括： ◦ Partition_name=value ◦ name1=value1/name2=value2：多级分区 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
modelName	是	输出的模型名称。	无
outputImportanceTableName	否	输出特征重要性的表名。	无
groupIDColName	否	数据分组列。	全表

参数	是否必选	描述	默认值
lossType	否	损失函数包括以下类型： ○ 0：GBRANK ○ 1：LAMBDAMART_DCG ○ 2：LAMBDAMART_NDCG ○ 3：LEAST_SQUARE ○ 4：LOG_LIKELIHOOD	0
metricType	否	Metric包括以下类型： ○ 0：NDCG（Normalized Discounted Cumulative Gain） ○ 1：DCG（Discounted Cumulative Gain） ○ 2：AUC，仅适用于label取值为0/1的场景	0
treeCount	否	树数量，取值范围为1~10000。	500
shrinkage	否	学习速率，取值范围为(0,1)。	0.05
maxLeafCount	否	最大叶子数，取值范围为1~1000。	32
maxDepth	否	树的最大深度，取值范围为1~100。	10
minLeafSampleCount	否	叶子节点容纳的最少样本数，取值范围为1~1000。	500
sampleRatio	否	训练采集的样本比例，取值范围为(0,1)。	0.6
featureRatio	否	训练采集的特征比例，取值范围为(0,1)。	0.6
tau	否	GBRank Loss中的Tau参数，取值范围为[0,1]。	0.6
p	否	GBRank Loss中的p参数，取值范围为[1,10]。	1
randSeed	否	随机数种子，取值范围为[0,10]。	0
newtonStep	否	使用Newton迭代法的开关，取值范围为{0,1}。	1
featureSplitValueMaxSize	否	特征分裂的最大数量，取值范围为1~1000。	500
lifecycle	否	输出表的生命周期。	无

> 说明
- GBDT与GBDT_LR默认的损失函数类型不同。因为GBDT默认为regression loss:mean squared error loss，GBDT_LR默认为logistic regression loss，所以GBDT_LR无需手动配置损失函数，系统会自动配置损失函数。
- GBDT的特征列、标签列及分组列仅支持数值类型。
- 连接ROC曲线时，预测组件需要选择目标基准值。

GBDT二分类示例

1. 使用SQL语句，生成训练数据。

```
drop table if exists gbdt_lr_test_input;
create table gbdt_lr_test_input
as
select
  *
from
(
  select
    cast(1 as double) as f0,
    cast(0 as double) as f1,
    cast(0 as double) as f2,
    cast(0 as double) as f3,
    cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(1 as double) as f1,
      cast(0 as double) as f2,
      cast(0 as double) as f3,
      cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(0 as double) as f1,
      cast(1 as double) as f2,
      cast(0 as double) as f3,
      cast(1 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(0 as double) as f1,
      cast(0 as double) as f2,
      cast(1 as double) as f3,
      cast(1 as bigint) as label
  from dual
  union all
    select
      cast(1 as double) as f0,
```

```
        cast(0 as double) as f1,
        cast(0 as double) as f2,
        cast(0 as double) as f3,
        cast(0 as bigint) as label
    from dual
    union all
      select
        cast(0 as double) as f0,
        cast(1 as double) as f1,
        cast(0 as double) as f2,
        cast(0 as double) as f3,
        cast(0 as bigint) as label
    from dual
) a;
```

生成的训练数据表gbdt_lr_test_input如下。

f0	f1	f2	f3	label
1.0	0.0	0.0	0.0	0
0.0	0.0	1.0	0.0	1
0.0	0.0	0.0	1.0	1
0.0	1.0	0.0	0.0	0
1.0	0.0	0.0	0.0	0
0.0	1.0	0.0	0.0	0

2. 使用PAI命令，提交GBDT二分类组件的训练参数。

```
drop offlinemodel if exists gbdt_lr_test_model;
PAI -name gbdt_lr
  -project algo_public
  -DfeatureSplitValueMaxSize="500"
  -DrandSeed="1"
  -Dshrinkage="1"
  -DmaxLeafCount="30"
  -DlabelColName="label"
  -DinputTableName="gbdt_lr_test_input"
  -DminLeafSampleCount="1"
  -DsampleRatio="1"
  -DmaxDepth="10"
  -DmodelName="gbdt_lr_test_model"
  -DmetricType="0"
  -DfeatureRatio="1"
  -DtestRatio="0"
  -DfeatureColNames="f0,f1,f2,f3"
  -DtreeCount="5"
```

3. 使用PAI命令，提交预测组件参数。

```
drop table if exists gbdt_lr_test_prediction_result;
PAI -name prediction
  -project algo_public
  -DdetailColName="prediction_detail"
  -DmodelName="gbdt_lr_test_model"
  -DitemDelimiter=","
  -DresultColName="prediction_result"
  -Dlifecycle="28"
  -DoutputTableName="gbdt_lr_test_prediction_result"
  -DscoreColName="prediction_score"
  -DkvDelimiter=":"
  -DinputTableName="gbdt_lr_test_input"
  -DenableSparse="false"
  -DappendColNames="label";
```

4. 查看预测结果表gbdt_lr_test_prediction_result。

label	prediction_result	prediction_score	prediction_detail
0	0	0.9984308925552831	{“0”：0.9984308925552831，“1”：0.001569107444716943}
0	0	0.9984308925552831	{“0”：0.9984308925552831，“1”：0.001569107444716943}
1	1	0.9982721832240973	{“0”：0.001727816775902724，“1”：0.9982721832240973}
1	1	0.9982721832240973	{“0”：0.001727816775902724，“1”：0.9982721832240973}
0	0	0.9984308925552831	{“0”：0.9984308925552831，“1”：0.001569107444716943}
0	0	0.9984308925552831	{“0”：0.9984308925552831，“1”：0.001569107444716943}

PS-SMART二分类

参数服务器PS（Parameter Server）致力于解决大规模的离线及在线训练任务，SMART（Scalable Multiple Additive Regression Tree）是GBDT（Gradient Boosting Decesion Tree）基于PS实现的迭代算法。PS-SMART支持百亿样本及几十万特征的训练任务，可以在上千节点中运行。同时，PS-SMART支持多种数据格式及直方图近似等优化技术。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
	是否稀疏格式	稀疏格式的KV之间使用空格分隔，key与value之间使用英文冒号（:）分隔。例如1:0.3 3:0.9。

页签	参数	描述
字段设置	**选择特征列**	输入表中，用于训练的特征列。如果输入数据是Dense格式，则只能选择数值（BIGINT或DOUBLE）类型。如果输入数据是Sparse KV格式，且key和value是数值类型，则只能选择STRING类型。
	选择标签列	输入表的标签列，支持STRING及数值类型。如果是内部存储，则仅支持数值类型。例如二分类中的0和1。
	选择权重列	列可以对每行样本进行加权，支持数值类型。
参数设置	**评估指标类型**	支持**negative loglikelihood for logistic regression**、**binary classiffication error**及**Area under curve for classification**类型。
	树数量	训练数据量与**树数量**成正比。
	树最大深度	默认值为5，即最多32个叶子节点。
	数据采样比例	构建每棵树时，采样部分数据进行学习，构建弱学习器，从而加快训练。
	特征采样比例	构建每棵树时，采样部分特征进行学习，构建弱学习器，从而加快训练。
	L1惩罚项系数	控制叶子节点大小。该参数值越大，叶子节点规模分布越均匀。如果过拟合，则增大该参数值。
	L2惩罚项系数	控制叶子节点大小。该参数值越大，叶子节点规模分布越均匀。如果过拟合，则增大该参数值。
	学习速率	取值范围为(0,1)。
	近似Sketch精度	构造Sketch的切割分位点阈值。该参数值越小，获得的桶越多。一般使用默认值0.03，无需手动配置。
	最小分裂损失变化	分裂节点所需要的最小损失变化。该参数值越大，分裂越保守。
	特征数量	特征数量或最大特征ID。如果估计使用资源时，未配置该参数，则系统会启动SQL任务自动计算。
	全局偏置项	所有样本的初始预测值。
	特征重要性类型	支持**模型中，该特征做为分裂特征的次数**、**模型中，该特征带来的信息增益**及**模型中，该特征在分裂节点覆盖的样本数**类型。
执行调优	**计算核心数**	默认为系统自动分配。
	每个核内存大小	单个核心使用的内存，单位为MB。通常无需手动配置，系统会自动分配。

- PAI命令方式

```
#训练。
PAI -name ps_smart
  -project algo_public
  -DinputTableName="smart_binary_input"
  -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
  -DoutputTableName="pai_temp_24515_545859_2"
  -DoutputImportanceTableName="pai_temp_24515_545859_3"
  -DlabelColName="label"
  -DfeatureColNames="f0,f1,f2,f3,f4,f5"
  -DenableSparse="false"
  -Dobjective="binary:logistic"
  -Dmetric="error"
  -DfeatureImportanceType="gain"
  -DtreeCount="5";
  -DmaxDepth="5"
  -Dshrinkage="0.3"
  -Dl2="1.0"
  -Dl1="0"
  -Dlifecycle="3"
  -DsketchEps="0.03"
  -DsampleRatio="1.0"
  -DfeatureRatio="1.0"
  -DbaseScore="0.5"
  -DminSplitLoss="0"
#预测。
PAI -name prediction
  -project algo_public
  -DinputTableName="smart_binary_input";
  -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
  -DoutputTableName="pai_temp_24515_545860_1"
  -DfeatureColNames="f0,f1,f2,f3,f4,f5"
  -DappendColNames="label,qid,f0,f1,f2,f3,f4,f5"
  -DenableSparse="false"
  -Dlifecycle="28"
```

模块	参数	是否必选	描述	默认值
	featureColNames	是	输入表中，用于训练的特征列。如果输入表是Dense格式，则只能选择数值（BIGINT或DOUBLE）类型。如果输入表是Sparse KV格式，且KV格式中key和value是数值类型，则只能选择STRING类型。	无
	labelColName	是	输入表的标签列，支持STRING及数值类型。如果是内部存储，则仅支持数值类型。例如二分类中的0和1。	无
	weightCol	否	列可以对每行样本进行加权，支持数值类型。	无
	enableSparse	否	是否为稀疏格式，取值范围为{true,false}。稀疏格式的KV之间使用空格分隔，key与value之间使用英文冒号（:）分隔。例如1:0.3 3:0.9。	false
数据参数				

数据参数模块	参数	是否必选	描述	默认值
	inputTableName	是	输入表的名称。	无
	modelName	是	输出的模型名称。	无
	outputImportanceTableName	否	输出特征重要性的表名。	无
	inputTablePartitions	否	格式为ds=1/pt=1。	无
	outputTableName	否	输出至MaxCompute的表，二进制格式，不支持读取，只能通过SMART的预测组件获取。	无
	lifecycle	否	输出表的生命周期。	3
算法参数	objective	是	目标函数类型。如果进行二分类训练，则选择binary:logistic。	无
	metric	否	训练集的评估指标类型，输出在Logview文件Coordinator区域的stdout。支持以下类型： ◦ logloss：对应可视化方式的**negative loglikelihood for logistic regression**类型。 ◦ error：对应可视化方式中的**binary classification error**类型。 ◦ auc：对应可视化方式中的**Area under curve for classification**类型。	无
	treeCount	否	树数量，与训练时间成正比。	1
	maxDepth	否	树的最大深度，取值范围为1~20。	5
	sampleRatio	否	数据采样比例，取值范围为(0,1]。如果取值为1.0，则表示不采样。	1.0
	featureRatio	否	特征采样比例，取值范围为(0,1]。如果取值为1.0，则表示不采样。	1.0
	l1	否	L1惩罚项系数。该参数值越大，叶子节点分布越均匀。如果过拟合，则增大该参数值。	0
	l2	否	L2惩罚项系数。该参数值越大，叶子节点分布越均匀。如果过拟合，则增大该参数值。	1.0
	shrinkage	否	取值范围为(0,1)。	0.3
	sketchEps	否	构造Sketch的切割分位点阈值，桶数为O(1.0/sketchEps)。该参数值越小，获得的桶越多。一般使用默认值，无需手动配置。取值范围为(0,1)。	0.03

模块	参数	是否必选	描述	默认值
	minSplitLoss	否	分裂节点所需要的最小损失变化。该参数值越大，分裂越保守。	0
	featureNum	否	特征数量或最大特征ID。如果估计使用资源时，未配置该参数，则系统会启动SQL任务自动计算。	无
	baseScore	否	所有样本的初始预测值。	0.5
	featureImportanceType	否	计算特征重要性的类型，包括： ○ weight：在模型中，该特征作为分裂特征的次数。 ○ gain：在模型中，该特征带来的信息增益。 ○ cover：在模型中，该特征在分裂节点覆盖的样本数。	gain
调优参数	coreNum	否	核心数量，该参数值越大，算法运行越快。	系统自动分配
	memSizePerCore	否	每个核心使用的内存，单位为MB。	系统自动分配

PS-SMART二分类示例

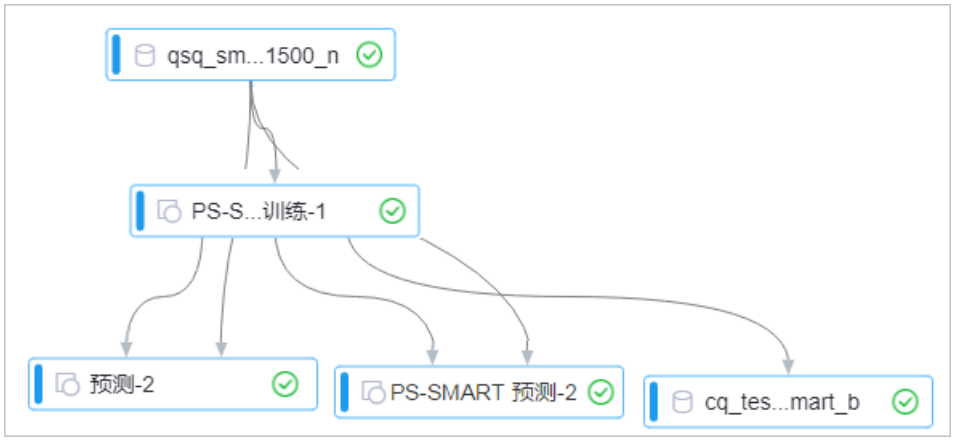
1. 使用SQL语句，生成训练数据（以Dense格式数据为例）。

```
drop table if exists smart_binary_input;
create table smart_binary_input lifecycle 3 as
select
*
from
(
select 0.72 as f0, 0.42 as f1, 0.55 as f2, -0.09 as f3, 1.79 as f4, -1.2 as f5, 0 as label from dual
union all
select 1.23 as f0, -0.33 as f1, -1.55 as f2, 0.92 as f3, -0.04 as f4, -0.1 as f5, 1 as label from dual
union all
select -0.2 as f0, -0.55 as f1, -1.28 as f2, 0.48 as f3, -1.7 as f4, 1.13 as f5, 1 as label from dual
union all
select 1.24 as f0, -0.68 as f1, 1.82 as f2, 1.57 as f3, 1.18 as f4, 0.2 as f5, 0 as label from dual
union all
select -0.85 as f0, 0.19 as f1, -0.06 as f2, -0.55 as f3, 0.31 as f4, 0.08 as f5, 1 as label from dual
union all
select 0.58 as f0, -1.39 as f1, 0.05 as f2, 2.18 as f3, -0.02 as f4, 1.71 as f5, 0 as label from dual
union all
select -0.48 as f0, 0.79 as f1, 2.52 as f2, -1.19 as f3, 0.9 as f4, -1.04 as f5, 1 as label from dual
union all
select 1.02 as f0, -0.88 as f1, 0.82 as f2, 1.82 as f3, 1.55 as f4, 0.53 as f5, 0 as label from dual
union all
select 1.19 as f0, -1.18 as f1, -1.1 as f2, 2.26 as f3, 1.22 as f4, 0.92 as f5, 0 as label from dual
union all
select -2.78 as f0, 2.33 as f1, 1.18 as f2, -4.5 as f3, -1.31 as f4, -1.8 as f5, 1 as label from dual
) tmp;
```

生成的训练数据如下。

序号	f0	f1	f2	f3	f4	f5	label
1	0.72	0.42	0.55	-0.09	1.79	-1.2	0
2	1.23	-0.33	-1.55	0.92	-0.04	-0.1	1
3	-0.2	-0.55	-1.28	0.48	-1.7	1.13	1
4	1.24	-0.68	1.82	1.57	1.18	0.2	0
5	-0.85	0.19	-0.06	-0.55	0.31	0.08	1
6	0.58	-1.39	0.05	2.18	-0.02	1.71	0
7	-0.48	0.79	2.52	-1.19	0.9	-1.04	1
8	1.02	-0.88	0.82	1.82	1.55	0.53	0
9	1.19	-1.18	-1.1	2.26	1.22	0.92	0
10	-2.78	2.33	1.18	-4.5	-1.31	-1.8	1

2. 创建如下实验，详情请参见算法建模。



3. 配置PS-SMART二分类组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	特征列	选择f0、f1、f2、f3、f4、及f5列。
	标签列	选择label列。
参数设置	评估指标类型	选择Area under curve for classification。
	树数量	输入5。

4. 查看统一预测组件的预测结果。

序号	f0	f1	f2	f3	f4	f5	label	prediction_result	prediction_score	prediction_detail
1	0.72	0.42	0.55	-0.09	1.79	-1.2	0	0	0.5563381910324097	{"0": 0.5563382208347321, "1": 0.4436617791652679}
2	1.23	-0.33	-1.55	0.92	-0.04	-0.1	1	1	0.6076213121414185	{"0": 0.3923786878585815, "1": 0.6076213121414185}
3	-0.2	-0.55	-1.28	0.48	-1.7	1.13	1	1	0.5768264532089233	{"0": 0.4231735467910767, "1": 0.5768264532089233}
4	1.24	-0.68	1.82	1.57	1.18	0.2	0	0	0.7096900939941406	{"0": 0.709690123796463, "1": 0.290309876203537}
5	-0.85	0.19	-0.06	-0.55	0.31	0.08	1	1	0.7265769839286804	{"0": 0.2734230160713196, "1": 0.7265769839286804}
6	0.58	-1.39	0.05	2.18	-0.02	1.71	0	0	0.5573073625564575	{"0": 0.5573073327541351, "1": 0.4426926672458649}
7	-0.48	0.79	2.52	-1.19	0.9	-1.04	1	1	0.5777847170829773	{"0": 0.4222152829170227, "1": 0.5777847170829773}
8	1.02	-0.88	0.82	1.82	1.55	0.53	0	0	0.7096900939941406	{"0": 0.709690123796463, "1": 0.290309876203537}
9	1.19	-1.18	-1.1	2.26	1.22	0.92	0	0	0.7096900939941406	{"0": 0.709690123796463, "1": 0.290309876203537}
10	-2.78	2.33	1.18	-4.5	-1.31	-1.8	1	1	0.7265769839286804	{"0": 0.2734230160713196, "1": 0.7265769839286804}

其中prediction_detail列的1表示正例，0表示负例。

5. 查看PS-SMART预测组件的预测结果。

序号	original_label	prediction_score	leaf_index
1	0	0.4066115617752075	2 2 2 2 2
2	1	0.5709302425384521	2 2 1 1 2
3	1	0.6125051379203796	1 1 1 1 1
4	0	0.3217407166957855	2 1 2 2 1
5	1	0.6954338550567627	1 2 1 1 2
6	0	0.4794751703739166	2 1 1 1 1
7	1	0.5404139757156372	1 2 2 2 2
8	0	0.3217407166957855	2 1 2 2 1
9	0	0.3217407166957855	2 1 2 2 1
10	1	0.6954338550567627	1 2 1 1 2

其中：

- **prediction_score**：表示预测正例的概率。如果该值大于0.5，则表示预测结果为正例，反之预测结果为负例。
- **leaf_index**：表示预测的叶子节点编号，每个样本有N（树数量）个，每棵树对应一个数字，该数字表示样本落在这棵树叶子节点的编号。

> **说明** PS-Smart预测组件需要一列string类型数据作为label，且该列不能为空或者NULL，可以将某特征列通过类型转换组件转换为string类型。

6. 右键单击**PS-SMART二分类**组件，在快捷菜单，选择**查看数据 > 查看输出桩3**，查看特征重要性表。

序号	id	value
1	0	0.5690338015556335
2	1	0.21714292466640472
3	4	0.21382322907447815

其中：

- **id**：表示传入的特征序号。因为该示例传入的特征为f0、f1、f2、f3、f4及f5，所以**id**列的0表示f0特征列，**id**列的4表示f4特征列。如果输入数据是KV格式，则**id**列表示KV中的key。
- **value**：表示特征重要性类型，默认为gain，即该特征对模型带来的信息增益之和。
- 该特性重要性表中仅有3个特性，表示树在分裂过程中仅使用了这三个特性，可以认为其他特性的特征重要性为0。

相关说明：

- PS-SMART二分类组件的目标列仅支持数值类型，且0表示负例，1表示正例。如果MaxCompute表数据是STRING类型，则需要进行类型转换。例如，分类目标是Good/Bad字符串，需要转换为1/0。
- 如果数据是KV格式，则特征ID必须为正整数，特征值必须为实数。如果特征ID为字符串类型，则需要使用序列化组件进行序列化。如果特征值为类别型字符串，需要进行特征离散化等特征工程处理。
- 虽然PS-SMART二分类组件支持数十万特征任务，但是消耗资源大且运行速度慢，可以使用GBDT类算法进行训练。GBDT类算法适合直接使用连续特征进行训练，除需要对类别特征进行One-Hot编码（筛除低频特征）以外，不建议对其他连续型数值特征进行离散化。
- PS-SMART算法会引入随机性。例如，data_sample_ratio及fea_sample_ratio表示的数据和特征采样、算法使用的直方图近似优化及局部Sketch归并为全局Sketch的顺序随机性。虽然多个worker分布式执行时，树结构不同，但是从理论上可以保证模型效果相近。如果您在训练过程中，使用相同数据和参数，多次得到的结果不一致，属于正常现象。

- 如果需要加速训练，可以增大**计算核心数**。因为PS-SMART算法需要所有服务器获得资源后，才能开始训练，所以集群忙碌时，申请较多资源会增加等待时间。

PS逻辑回归二分类

数服务器PS（Parameter Server）致力于解决大规模的离线及在线训练任务，逻辑回归（Logistic Regression）是经典的二分类算法，广泛应用于广告及搜索场景。PS逻辑回归支持千亿样本、十亿特征的二分类训练任务。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**选择特征列**	输入数据源中，用于训练的特征列。如果输入数据为Dense格式，则支持DOUBLE及BIGINT类型。如果输入数据为Sparse KV格式，则仅支持STRING类型。 **说明** 特征数量不能超过两千万。
	选择标签列	输入数据源中，标签列的名称。
	是否稀疏格式	如果**是否稀疏格式**为true，则特征ID不能使用0，建议特征从1开始编号。
参数设置	**L1 weight**	L1正则化系数。该参数值越大，模型非零元素越少。如果过拟合，则增大该参数值。
	L2 weight	L2正则化系数。该参数值越大，模型参数绝对值越小。如果过拟合，则增大该参数值。
	最大迭代次数	算法的最大迭代次数，0表示迭代次数无限制。
	最小收敛误差	优化算法的终止条件，通常取值为10次迭代Loss相对变化率的平均值。该参数值越小，算法执行时间越长。
	最大特征ID	最大的特征ID或特征维度，取值可以大于实际值。该参数值越大，内存占用越大。如果未配置该参数，则系统启动SQL任务自动计算。
执行调优	**核心数**	核心数量。
	每个核的内存大小	每个核心的内存，单位为MB。

- PAI命令方式

```
#训练
PAI -name ps_lr
  -project algo_public
  -DinputTableName="lm_test_input"
  -DmodelName="logistic_regression_model"
  -DlabelColName="label"
  -DfeatureColNames="f0,f1,f2,f3,f4,f5"
  -Dl1Weight=1.0
  -Dl2Weight=0.0
  -DmaxIter=100
  -Depsilon=1e-6
  -DenableSparse=false
#预测
drop table if exists logistic_regression_predict;
PAI -name prediction
  -DmodelName="logistic_regression_model"
  -DoutputTableName="logistic_regression_predict"
  -DinputTableName="lm_test_input"
  -DappendColNames=label
  -DfeatureColNames="f0,f1,f2,f3,f4,f5"
  -DenableSparse=false
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无
featureColNames	是	输入表中，用于训练的特征列。 如果输入数据为Dense格式，则支持DOUBLE及BIGINT类型。如果输入数据为Sparse KV格式，则仅支持STRING类型。 **说明** 特征数量不能超过两千万。	无
labelColName	是	输入表的标签列名，支持DOUBLE及BIGINT类型。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持的格式包括： ◦ partition_name=value ◦ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区间使用英文逗号（,）分隔。	全表
modelName	是	输出的模型名。默认输出至OfflineModel。如果enableModelIo为false，则输出至MaxCompute。	无

参数名称	是否必选	参数描述	默认值
enableModelIo	否	是否输出至OfflineModel，取值范围为{true,false}。如果enableModelIo为false，则输出至MaxCompute。	true
l1Weight	否	L1正则化系数。该参数值越大，模型非零元素越少。如果过拟合，则增大该参数值。	1.0
l2Weight	否	L2正则化系数。该参数值越大，模型参数绝对值越小。如果过拟合，则增大该参数值。	0
maxIter	否	算法的最大迭代次数。	100
epsilon	否	算法终止条件。	1.0e-06
modelSize	否	最大的特征ID或特征维度，取值可以大于实际值。该参数值越大，内存占用越大。如果未配置该参数，则系统启动SQL任务自动计算。	0
enableSparse	否	输入数据是否为稀疏格式，取值范围为{true,false}。	false
itemDelimiter	否	输入表数据为稀疏格式时，KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	输入表数据为稀疏格式时，key和value之间的分隔符。	英文冒号（:）
coreNum	否	核心数量。	系统自动分配
memSizePerCore	否	单个核心使用的内存数，单位为MB。	系统自动分配

PS逻辑回归二分类示例

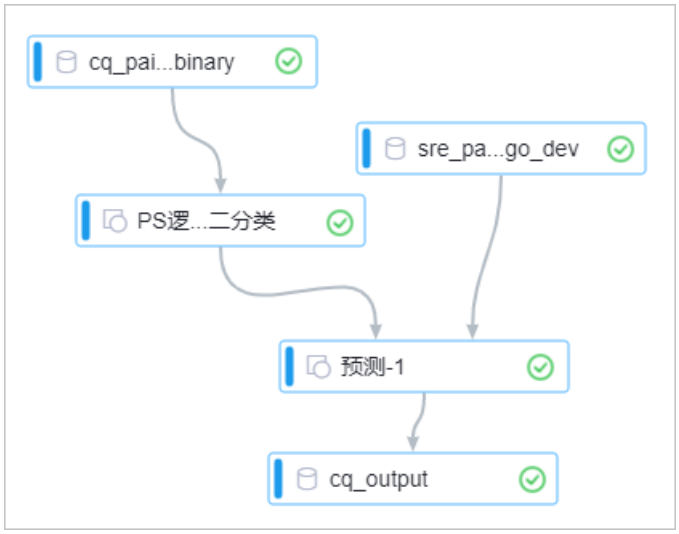
1. 使用SQL语句，生成训练数据（以Dense格式数据为例）。

```
drop table if exists lm_test_input;
create table lm_test_input as
select
*
from
(
select 0.72 as f0, 0.42 as f1, 0.55 as f2, -0.09 as f3, 1.79 as f4, -1.2 as f5, 0 as label from dual
union all
select 1.23 as f0, -0.33 as f1, -1.55 as f2, 0.92 as f3, -0.04 as f4, -0.1 as f5, 1 as label from dual
union all
select -0.2 as f0, -0.55 as f1, -1.28 as f2, 0.48 as f3, -1.7 as f4, 1.13 as f5, 1 as label from dual
union all
select 1.24 as f0, -0.68 as f1, 1.82 as f2, 1.57 as f3, 1.18 as f4, 0.2 as f5, 0 as label from dual
union all
select -0.85 as f0, 0.19 as f1, -0.06 as f2, -0.55 as f3, 0.31 as f4, 0.08 as f5, 1 as label from dual
union all
select 0.58 as f0, -1.39 as f1, 0.05 as f2, 2.18 as f3, -0.02 as f4, 1.71 as f5, 0 as label from dual
union all
select -0.48 as f0, 0.79 as f1, 2.52 as f2, -1.19 as f3, 0.9 as f4, -1.04 as f5, 1 as label from dual
union all
select 1.02 as f0, -0.88 as f1, 0.82 as f2, 1.82 as f3, 1.55 as f4, 0.53 as f5, 0 as label from dual
union all
select 1.19 as f0, -1.18 as f1, -1.1 as f2, 2.26 as f3, 1.22 as f4, 0.92 as f5, 0 as label from dual
union all
select -2.78 as f0, 2.33 as f1, 1.18 as f2, -4.5 as f3, -1.31 as f4, -1.8 as f5, 1 as label from dual
) tmp;
```

生成的训练数据如下。

序号	f0	f1	f2	f3	f4	f5	label
1	0.72	0.42	0.55	-0.09	1.79	-1.2	0
2	1.23	-0.33	-1.55	0.92	-0.04	-0.1	1
3	-0.2	-0.55	-1.28	0.48	-1.7	1.13	1
4	1.24	-0.68	1.82	1.57	1.18	0.2	0
5	-0.85	0.19	-0.06	-0.55	0.31	0.08	1
6	0.58	-1.39	0.05	2.18	-0.02	1.71	0
7	-0.48	0.79	2.52	-1.19	0.9	-1.04	1
8	1.02	-0.88	0.82	1.82	1.55	0.53	0
9	1.19	-1.18	-1.1	2.26	1.22	0.92	0
10	-2.78	2.33	1.18	-4.5	-1.31	-1.8	1

2. 构建如下实验，详情请参见算法建模。



3. 配置PS逻辑回归二分类组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**选择特征列**	选择f0、f1、f2、f3、f4及f5列。
	选择标签列	选择label列。
执行调优	**核心数**	输入3。
	每个核的内存大小	输入1024。

4. 配置预测组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**特征列**	选择f0、f1、f2、f3、f4及f5列。
	原样输出列	选择f0、f1、f2、f3、f4、f5及label列。

5. 运行实验，查看预测结果。

序号	prediction_result	prediction_score	prediction_detail
1	0	0.9083687005394571	{"0": 0.9083687005394571, "1": 0.09163129946054296}
2	0	0.9018497676649229	{"0": 0.9018497676649229, "1": 0.09815023233507712}
3	1	0.8688191866303958	{"0": 0.1311808133696042, "1": 0.8688191866303958}
4	0	0.970607987507053	{"0": 0.970607987507053, "1": 0.02939201249294704}
5	1	0.8705510193907658	{"0": 0.1294489806092342, "1": 0.8705510193907658}
6	0	0.8426501566563311	{"0": 0.8426501566563311, "1": 0.1573498433436689}
7	1	0.7559534361845501	{"0": 0.2440465638154499, "1": 0.7559534361845501}
8	0	0.9722574685125349	{"0": 0.9722574685125349, "1": 0.02774253148746508}
9	0	0.9788787481815239	{"0": 0.9788787481815239, "1": 0.0211212518184761}
10	1	0.999816509540659	{"0": 0.0001834904593409625, "1": 0.999816509540659}

其中prediction_detail列的1表示正例，0表示负例。

相关说明：

- PS逻辑回归二分类组件的目标列仅支持数值类型，且0表示负例，1表示正例。如果MaxCompute表数据是STRING类型，则需要进行类型转换。例如，分类目标是Good/Bad字符串，需要转换为1/0。
- 如果数据是KV格式，则特征ID必须为正整数，特征值必须为实数。如果特征ID为字符串类型，则需要使用序列化组件进行序列化。如果特征值为类别型字符串，需要进行特征离散化等特征工程处理。

3.5.2. 多分类

本文为您介绍PAI-Studio提供的多分类算法，包括PS-SMART多分类、K近邻、逻辑回归多分类、随机森林及朴素贝叶斯。

PS-SMART多分类

参数服务器PS（Parameter Server）致力于解决大规模的离线及在线训练任务，SMART（Scalable Multiple Additive Regression Tree）是GBDT（Gradient Boosting Decesion Tree）基于PS实现的迭代算法。PS-SMART支持百亿样本及几十万特征的训练任务，可以在上千节点中运行。同时，PS-SMART支持多种数据格式及直方图近似等优化技术。

PAI-Studio支持通过可视化或PAI命令方式，配置PS-SMART多分类组件的参数：

- 可视化方式

页签	参数	描述
字段设置	**是否稀疏格式**	稀疏格式的KV之间使用空格分隔，key与value之间使用英文冒号（:）分隔。例如1:0.3 3:0.9。
	选择特征列	输入表中，用于训练的特征列。如果输入数据是Dense格式，则只能选择数值（BIGINT或DOUBLE）类型。如果输入数据是Sparse KV格式，且key和value是数值类型，则只能选择STRING类型。
	选择标签列	输入表的标签列，支持STRING及数值类型。如果是内部存储，则仅支持数值类型。例如二分类中的0和1。
	选择权重列	列可以对每行样本进行加权，支持数值类型。
	类别数	多分类的类别数量。如果**类别数**为n，则标签列的取值为{0,1,2,...,n-1}。
	评估指标类型	支持**multiclass negative log likelihood**和**multiclass classification error**类型。
	树数量	训练数据量与**树数量**成正比。
	树最大深度	默认值为5，即最多32个叶子节点。
	数据采样比例	构建每棵树时，采样部分数据进行学习，构建弱学习器，从而加快训练。
	特征采样比例	构建每棵树时，采样部分特征进行学习，构建弱学习器，从而加快训练。

页签	参数	描述
参数设置	**L1惩罚项系数**	控制叶子节点大小。该参数值越大，叶子节点规模分布越均匀。如果过拟合，则增大该参数值。
	L2惩罚项系数	控制叶子节点大小。该参数值越大，叶子节点规模分布越均匀。如果过拟合，则增大该参数值。
	学习速率	取值范围为(0,1)。
	近似Sketch精度	构造Sketch的切割分位点阈值。该参数值越小，获得的桶越多。一般使用默认值0.03，无需手动配置。
	最小分裂损失变化	分裂节点所需要的最小损失变化。该参数值越大，分裂越保守。
	特征数量	特征数量或最大特征ID。如果需要估计使用资源，则必须手动配置该参数。
	全局偏置项	所有样本的初始预测值。
	特征重要性类型	支持**模型中，该特征做为分裂特征的次数**、**模型中，该特征带来的信息增益**及**模型中，该特征在分裂节点覆盖的样本数**类型。
执行调优	**核心数**	默认为系统自动分配。
	每个核的内存大小	单个核心使用的内存，单位为MB。通常无需手动配置，系统自动分配。

- PAI命令方式

```
#训练。
PAI -name ps_smart
  -project algo_public
  -DinputTableName="smart_multiclass_input"
  -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
  -DoutputTableName="pai_temp_24515_545859_2"
  -DoutputImportanceTableName="pai_temp_24515_545859_3"
  -DlabelColName="label"
  -DfeatureColNames="features"
  -DenableSparse="true"
  -Dobjective="multi:softprob"
  -Dmetric="mlogloss"
  -DfeatureImportanceType="gain"
  -DtreeCount="5";
  -DmaxDepth="5"
  -Dshrinkage="0.3"
  -Dl2="1.0"
  -Dl1="0"
  -Dlifecycle="3"
  -DsketchEps="0.03"
  -DsampleRatio="1.0"
  -DfeatureRatio="1.0"
  -DbaseScore="0.5"
  -DminSplitLoss="0"
#预测。
PAI -name prediction
  -project algo_public
  -DinputTableName="smart_multiclass_input";
  -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
  -DoutputTableName="pai_temp_24515_545860_1"
  -DfeatureColNames="features"
  -DappendColNames="label,features"
  -DenableSparse="true"
  -DkvDelimiter=":"
  -Dlifecycle="28"
```

模块	参数	是否必选	描述	默认值
	featureColNames	是	输入表中，用于训练的特征列。如果输入表是Dense格式，则只能选择数值（BIGINT或DOUBLE）类型。如果输入表是Sparse KV格式，且KV格式中key和value是数值类型，则只能选择STRING类型。	无
	labelColName	是	输入表的标签列，支持STRING及数值类型。如果是内部存储，则仅支持数值类型。例如多分类的{0,1,2,...,n-1}，其中n表示类别数量。	无
	weightCol	否	列可以对每行样本进行加权，支持数值类型。	无
	enableSparse	否	是否为稀疏格式，取值范围为{true,false}。稀疏格式的KV之间使用空格分隔，key与value之间使用英文冒号（:）分隔。例如1:0.3 3:0.9。	false

数据参数	参数	是否必选	描述	默认值
模块参数	inputTableName	是	输入表的名称。	无
	modelName	是	输出的模型名称。	无
	outputImportanceTableName	否	输出特征重要性的表名。	无
	inputTablePartitions	否	格式为ds=1/pt=1。	无
	outputTableName	否	输出至MaxCompute的表，二进制格式，不支持读取，只能通过SMART的预测组件获取。	无
	lifecycle	否	输出表的生命周期。	3
算法参数	classNum	是	多分类的类别数量。如果类别数量为n，则标签列取值为{0,1,2,...,n-1}。	无
	objective	是	目标函数类型。如果进行多分类训练，则选择multi:softprob。	无
	metric	否	训练集的评估指标类型，输出在Logview文件Coordinator区域的stdout。支持以下类型： ○ mlogloss：对应可视化方式的**multiclass negative log likelihood**类型。 ○ merror：对应可视化方式中的**multiclass classification error**类型。	无
	treeCount	否	树数量，与训练时间成正比。	1
	maxDepth	否	树的最大深度，取值范围为1~20。	5
	sampleRatio	否	数据采样比例，取值范围为(0,1]。如果取值为1.0，则表示不采样。	1.0
	featureRatio	否	特征采样比例，取值范围为(0,1]。如果取值为1.0，则表示不采样。	1.0
	l1	否	L1惩罚项系数。该参数值越大，叶子节点分布越均匀。如果过拟合，则增大该参数值。	0
	l2	否	L2惩罚项系数。该参数值越大，叶子节点分布越均匀。如果过拟合，则增大该参数值。	1.0
	shrinkage	否	取值范围为(0,1)。	0.3
	sketchEps	否	构造Sketch的切割分位点阈值，桶数为O(1.0/sketchEps)。该参数值越小，获得的桶越多。一般使用默认值，无需手动配置。取值范围为(0,1)。	0.03

模块	参数	是否必选	描述	默认值
	minSplitLoss	否	分裂节点所需要的最小损失变化。该参数值越大，分裂越保守。	0
	featureNum	否	特征数量或最大特征ID。如果需要估计使用资源，则必须手动配置该参数。	无
	baseScore	否	所有样本的初始预测值。	0.5
	featureImportanceType	否	计算特征重要性的类型，包括： ○ weight：在模型中，该特征作为分裂特征的次数。 ○ gain：在模型中，该特征带来的信息增益。 ○ cover：在模型中，该特征在分裂节点覆盖的样本数。	gain
调优参数	coreNum	否	核心数量，该参数值越大，算法运行越快。	系统自动分配
	memSizePerCore	否	每个核心使用的内存，单位为MB。	系统自动分配

PS-SMART多分类示例

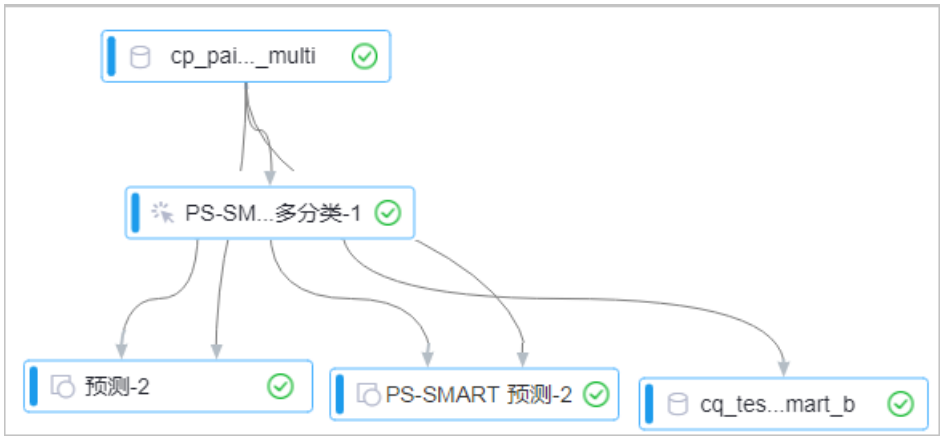
1. 使用如下SQL语句，生成输入数据（以KV格式数据为例）。

```
drop table if exists smart_multiclass_input;
create table smart_multiclass_input lifecycle 3 as
select
*
from
(
select 2 as label, '1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17' as features from dual
  union all
select 1 as label, '1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41' as features from dual
  union all
select 1 as label, '1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91' as features from dual
  union all
select 2 as label, '1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60' as features from dual
  union all
select 1 as label, '1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86' as features from dual
  union all
select 1 as label, '1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84' as features from dual
  union all
select 0 as label, '1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30' as features from dual
  union all
select 1 as label, '1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41' as features from dual
  union all
select 0 as label, '1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44' as features from dual
  union all
select 1 as label, '1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35' as features from dual
) tmp;
```

生成的数据如下。

序号	label	features
1	2	1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17
2	1	1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41
3	1	1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91
4	2	1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60
5	1	1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86
6	1	1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84
7	0	1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30
8	1	1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41
9	0	1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44
10	1	1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35

2. 构建实验，详情请参见算法建模。



3. 配置PS-SMART多分类组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**特征列**	选择features列。
	标签列	选择label列。
	是否稀疏格式	选中**是否稀疏格式**复选框。
参数设置	**类别数**	输入3。
	评估指标类型	选择**multiclass negative log likelihood**。
	树数量	输入5。

4. 配置统一预测组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**特征列**	默认全选，多余列不影响预测结果。
	原样输出列	选择label列。
	稀疏矩阵	选中**稀疏矩阵**复选框。
	key与value分隔符	输入英文冒号（:）。
	kv对间的分隔符	输入\u0020。

5. 配置PS-SMART预测组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**特征列**	默认全选，多余列不影响预测结果。
	原样输出列	选择label列。
	稀疏矩阵	选中**稀疏矩阵**复选框。

页签	参数	描述
	key与value分隔符	输入英文冒号（:）。
	kv对间的分隔符	输入\u0020。

6. 运行实验，查看统一预测组件的预测结果。

序号	label	features	prediction_result	prediction_score	prediction_detail
1	2	1:0.55 2:-0....	1	0.46681538224220276	{"0": 0.1409690380096436, "1": 0.4668153822422028, "2": 0.3922155499458313}
2	1	1:-1.26 2:1....	1	0.7185402512550354	{"0": 0.1393321454524994, "1": 0.7185402512550354, "2": 0.1421276330947876}
3	1	1:-0.77 2:0....	1	0.6726031303405762	{"0": 0.1620725691318512, "1": 0.6726031303405762, "2": 0.165324330329895}
4	2	1:0.86 2:-0....	2	0.4884149134159088	{"0": 0.1755447387695312, "1": 0.3360402882099152, "2": 0.4884149134159088}
5	1	1:-0.76 2:0....	1	0.6707357168197632	{"0": 0.1629969924688339, "1": 0.6707357168197632, "2": 0.1662672907114029}
6	1	1:2.22 2:-0....	0	0.4126577377319336	{"0": 0.4126577377319336, "1": 0.2393952459096909, "2": 0.3479470014572144}
7	0	1:-1.21 2:0....	0	0.541054368019104	{"0": 0.541054368019104, "1": 0.2916863262653351, "2": 0.1672592610120773}
8	1	1:2.17 2:-0....	1	0.5768934488296509	{"0": 0.1118654236197472, "1": 0.5768934488296509, "2": 0.311241090297699}
9	0	1:-0.40 2:0....	0	0.5392904877662659	{"0": 0.5392904877662659, "1": 0.2939955592155457, "2": 0.1667139828205109}
10	1	1:0.17 2:0....	1	0.7185402512550354	{"0": 0.1393321454524994, "1": 0.7185402512550354, "2": 0.1421276330947876}

其中：

- **prediction_detail**列中的0、1及2表示多分类的类别。
- **predict_result**列表示预测的结果类别。
- **predict_score**列表示预测为**predict_result**类别的概率。

7. 查看PS-Smart预测组件的预测结果。

序号	original_label	score_class_0	score_class_1	score_class_2	leaf_index
1	2	0.14096903800964355	0.46681538224220276	0.3922155499458313	1 1 2 1 1 2 1 1 2 1 2 2 1 2 2
2	1	0.1393321305513382	0.7185402512550354	0.1421276330947876	1 1 1 1 1 1 1 1 1 1 3 1 1 3 1
3	1	0.1620725691318512	0.6726031303405762	0.16532433032989502	1 1 1 1 1 1 1 1 1 1 3 1 1 2 1
4	2	0.17554473876953125	0.33604028820991516	0.4884149134159088	1 2 2 1 2 2 1 2 2 1 4 2 1 4 2
5	1	0.16299699246883392	0.6707357168197632	0.1662672907114029	1 1 1 1 1 1 1 1 1 1 2 1 1 3 1
6	1	0.4126577377319336	0.23939524590969086	0.34794700145721436	2 2 2 2 2 2 2 2 2 2 4 2 2 4 2
7	0	0.541054368019104	0.2916863262653351	0.16725926101207733	2 2 1 2 2 1 2 2 1 2 4 1 2 2 1
8	1	0.11186541616916656	0.5768935084342957	0.3112410604953766	1 1 2 1 1 2 1 1 2 1 3 2 1 3 2
9	0	0.5392904877662659	0.29399555921554565	0.16671398282051086	2 2 1 2 2 1 2 2 1 2 2 1 2 4 1
10	1	0.1393321305513382	0.7185402512550354	0.1421276330947876	1 1 1 1 1 1 1 1 1 1 3 1 1 3 1

其中：

- score_class_k列表示预测为第K类别的概率。
- **leaf_index**列表示预测的叶子节点编号。如果树数量为N，类别数量为M，则每个样本的**leaf_index**取值为N*M个数。例如，该示例的**leaf_index**取值为5*3=15。每棵树对应一个数字，该数字表示样本落在这棵树叶子节点的编号。

8. 右键单击**PS-SMART多分类**组件，在快捷菜单，选择**查看数据 > 查看输出桩3**，查看特征重要性。

序号	id	value
1	1	0.276059627532959
2	3	0.20854459702968597
3	4	0.31002077460289
4	5	0.20537501573562622

其中：

- id列表示传入的特征序号。因为该示例的输入数据是KV格式，所以id列表示KV对中的key。
- value列表示特征重要性类型，默认为gain，即该特征对模型带来的信息增益之和。

相关说明：

- PS-SMART多分类组件的目标列仅支持数值类型。如果MaxCompute表数据是STRING类型，则需要进行类型转换。例如，分类目标是Good/Medium/Bad字符串，需要转换为0/1/2。
- 如果数据是KV格式，则特征ID必须为正整数，特征值必须为实数。如果特征ID为字符串类型，则需要使用序列化组件进行序列化。如果特征值为类别型字符串，需要进行特征离散化等特征工程处理。
- 虽然PS-SMART多分类组件支持数十万特征任务，但是消耗资源大且运行速度慢，可以使用GBDT类算法进行训练。GBDT类算法适合直接使用连续特征进行训练，除需要对类别特征进行One-Hot编码（筛除低频特征）外，不建议对其他连续型数值特征进行离散化。
- PS-SMART算法会引入随机性。例如，data_sample_ratio及fea_sample_ratio表示的数据和特征采样、算法使用的直方图近似优化及局部Sketch归并为全局Sketch的顺序随机性。虽然多个Worker分布式执行时，树结构不同，但是从理论上可以保证模型效果相近。如果您在训练过程中，使用相同数据和参数，多次得到的结果不一致，属于正常现象。
- 如果需要加速训练，可以增大**计算核心数**。因为PS-SMART算法需要所有服务器获得资源后，才能开始训练，所以集群忙碌时，申请较多资源会增加等待时间。

K近邻

K近邻算法进行分类的原理是针对预测表的每行数据，从训练表中选择与其距离最近的K条记录，将这K条记录中类别数量最多的类，作为该行的类别。

PAI-Studio支持通过可视化或PAI命令方式，配置K近邻算法组件的参数：

- 可视化方式

页签	参数	描述
字段设置	**选择训练表特征列**	用于训练的特征列。
	选择训练表的标签列	训练的目标列。
	选择预测表特征列	如果未配置该参数，则表示其与训练表特征列相同。
	产出表附加ID列	用于标识该列的身份，从而获得某列对应的预测值。系统默认使用预测表特征列，作为附加ID列。
	输入表数据是稀疏格式	使用KV格式表示稀疏数据。
	kv间的分隔符	默认为英文逗号（,）。
	key和value的分隔符	默认为英文冒号（:）。

页签	参数	描述
参数设置	**近邻个数**	默认值为100。
执行调优	**核心数**	默认系统自动分配。
	内存数	默认系统自动分配。

- PAI命令方式

```
PAI -name knn
  -DtrainTableName=pai_knn_test_input
  -DtrainFeatureColNames=f0,f1
  -DtrainLabelColName=class
  -DpredictTableName=pai_knn_test_input
  -DpredictFeatureColNames=f0,f1
  -DoutputTableName=pai_knn_test_output
  -Dk=2;
```

参数	是否必选	描述	默认值
trainTableName	是	训练表的表名。	无
trainFeatureColNames	是	训练表的特征列名。	无
trainLabelColName	是	训练表的标签列名。	无
trainTablePartitions	否	训练表中，参与训练的分区。	所有分区
predictTableName	是	预测表的表名。	无
outputTableName	是	输出表的表名。	无
predictFeatureColNames	否	预测表的特征列名。	与trainFeatureColNames相同
predictTablePartitions	否	预测表中，参与预测的分区。	所有分区
appendColNames	否	输出表中，附加预测表的列名。	与predictFeatureColNames相同
outputTablePartition	否	输出表的分区。	全表
k	否	最近邻的数量。取值范围为1~1000。	100
enableSparse	否	输入表数据是否为稀疏格式。取值范围为{true,false}。	false
itemDelimiter	否	如果输入表数据为稀疏格式，则KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	如果输入表数据为稀疏格式，则key和value之间的分隔符。	英文冒号（:）

参数	是否必选	描述	默认值
coreNum	否	节点数量。与memSizePerCore搭配使用，取值范围为1~20000。	系统自动计算
memSizePerCore	否	单个节点的内存，取值范围为1024 MB~64*1024 MB。	系统自动计算
lifecycle	否	输出表的生命周期。	无

K近邻示例

1. 生成训练数据。

```
create table pai_knn_test_input as
select * from
(
 select 1 as f0,2 as f1, 'good' as class from dual
 union all
 select 1 as f0,3 as f1, 'good' as class from dual
 union all
 select 1 as f0,4 as f1, 'bad' as class from dual
 union all
 select 0 as f0,3 as f1, 'good' as class from dual
 union all
 select 0 as f0,4 as f1, 'bad' as class from dual
)tmp;
```

2. 使用PAI命令，提交K近邻算法组件参数。

```
pai -name knn
  -DtrainTableName=pai_knn_test_input
  -DtrainFeatureColNames=f0,f1
  -DtrainLabelColName=class
  -DpredictTableName=pai_knn_test_input
  -DpredictFeatureColNames=f0,f1
  -DoutputTableName=pai_knn_test_output
  -Dk=2;
```

3. 查看训练结果。

f0	f1	prediction_result	prediction_score	prediction_detail
1	4	bad	1.0	{"bad": 1}
0	4	bad	1.0	{"bad": 1}
0	3	bad	0.5	{"bad": 0.5, "good": 0.5}
1	3	good	1.0	{"good": 1}
1	2	good	1.0	{"good": 1}

其中：

- f0和f1表示结果附件列。
- prediction_result表示分类结果。

- prediction_score表示分类结果对应的概率。
- prediction_detail表示最近的K个分类及其对应的概率。

逻辑回归多分类

逻辑回归多分类组件支持稀疏及稠密数据格式。PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	训练特征列	输入数据源中，用于训练的特征列。支持DOUBLE及BIGINT类型。 说明 特征数量不能超过两千万。
	目标列	输入数据源中，目标列名称。
	是否稀疏数据	输入数据是否为稀疏格式。
参数设置	正则项类型	支持L1、L2及None类型。
	最大迭代次数	默认值为100。
	正则系数	如果 正则项为None，则该参数失效。
	最小收敛误差	默认值为0.000001。

- PAI命令方式

```
PAI -name logisticregression_multi
  -project algo_public
  -DmodelName="xlab_m_logistic_regression_6096"
  -DregularizedLevel="1"
  -DmaxIter="100"
  -DregularizedType="l1"
  -Depsilon="0.000001"
  -DlabelColName="y"
  -DfeatureColNames="pdays,emp_var_rate"
  -DgoodValue="1"
  -DinputTableName="bank_data"
```

参数名称	是否必选	参数描述	默认值
inputTableName	是	输入表的表名。	无
featureColNames	否	输入表中，用于训练的特征列名。 说明 特征数量不能超过两千万。	所有数值列

参数名称	是否必选	参数描述	默认值
labelColName	是	输入表的标签列名。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持的格式包括： ○ partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区间使用英文逗号（,）分隔。	全表
modelName	是	输出的模型名。	无
regularizedType	否	正则化类型，取值范围为{'l1','l2','None'}。	l1
regularizedLevel	否	正则化系数。如果 regularizedType为None，则该参数失效。	1.0
maxIter	否	L-BFGS的最大迭代次数。	100
epsilon	否	收敛误差。该参数是L-BFGS算法的终止条件，即两次迭代的log-likelihood之差小于该值，迭代终止。	1.0e-06
enableSparse	否	输入数据是否为稀疏格式，取值范围为{true,false}。	false
itemDelimiter	否	输入表数据为稀疏格式时，KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	输入表数据为稀疏格式时，key和value之间的分隔符。	英文冒号（:）
coreNum	否	核心数量。	系统自动分配
memSizePerCore	否	单个核心使用的内存数，单位为MB。	系统自动分配

逻辑回归多分类示例

1. 使用SQL语句，生成训练数据。

```
drop table if exists multi_lr_test_input;
create table multi_lr_test_input
as
select
  *
from
(
  select
    cast(1 as double) as f0,
    cast(0 as double) as f1,
    cast(0 as double) as f2,
    cast(0 as double) as f3,
    cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(1 as double) as f1,
      cast(0 as double) as f2,
      cast(0 as double) as f3,
      cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(0 as double) as f1,
      cast(1 as double) as f2,
      cast(0 as double) as f3,
      cast(2 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(0 as double) as f1,
      cast(0 as double) as f2,
      cast(1 as double) as f3,
      cast(1 as bigint) as label
  from dual
) a;
```

生成的训练数据（multi_lr_test_input表）如下。

f0	f1	f2	f3	label
1.0	0.0	0.0	0.0	0
0.0	0.0	1.0	0.0	2
0.0	0.0	0.0	1.0	1
0.0	1.0	0.0	0.0	0

2. 使用PAI命令，提交逻辑回归多分类算法组件参数。

```
drop offlinemodel if exists multi_lr_test_model;
PAI -name logisticregression_multi
  -project algo_public
  -DmodelName="multi_lr_test_model"
  -DitemDelimiter=","
  -DregularizedLevel="1"
  -DmaxIter="100"
  -DregularizedType="None"
  -Depsilon="0.000001"
  -DkvDelimiter=":"
  -DlabelColName="label"
  -DfeatureColNames="f0,f1,f2,f3"
  -DenableSparse="false"
  -DinputTableName="multi_lr_test_input";
```

3. 使用PAI命令，提交预测组件参数。

```
drop table if exists multi_lr_test_prediction_result;
PAI -name prediction
  -project algo_public
  -DdetailColName="prediction_detail"
  -DmodelName="multi_lr_test_model"
  -DitemDelimiter=","
  -DresultColName="prediction_result"
  -Dlifecycle="28"
  -DoutputTableName="multi_lr_test_prediction_result"
  -DscoreColName="prediction_score"
  -DkvDelimiter=":"
  -DinputTableName="multi_lr_test_input"
  -DenableSparse="false"
  -DappendColNames="label";
```

4. 查看输出结果（multi_lr_test_prediction_result表）。

label	prediction_result	prediction_score	prediction_detail
0	0	0.9999997274902165	{“0”：0.9999997274902165，“1”：2.324679066261573e-07，“2”：2.324679066261569e-07}
0	0	0.9999997274902165	{“0”：0.9999997274902165，“1”：2.324679066261573e-07，“2”：2.324679066261569e-07}
2	2	0.9999999155958832	{“0”：2.018833979850994e-07，“1”：2.324679066261573e-07，“2”：0.9999999155958832}
1	1	0.9999999155958832	{“0”：2.018833979850994e-07，“1”：0.9999999155958832，“2”：2.324679066261569e-07}

随机森林

随机森林是一个包括多决策树的分类器，其分类结果由单棵树输出类别的众数决定。PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**选择特征列**	默认为除标签列和权重列外的所有列。
	排除列	不参与训练的列，不能与**选择特征列**同时使用。
	强制转换列	解析规则如下： ○ STRING、BOOLEAN及DATETIME类型的列，解析为离散类型。 ○ DOUBLE和BIGINT类型的列，解析为连续类型。 **说明** 如果需要将BIGINT类型的列解析为CATEGORICAL，则必须使用forceCategorical参数指定类型。
	权重列的列名	列可以对每行样本进行加权，支持数值类型。
	标签列	输入表的标签列，支持STRING及数值类型。
参数设置	**森林中树的个数**	取值范围为1~1000。
	单颗树的算法在森林中的位置	如果有N棵树，且algorithmTypes=[a,b]，则： ○ [0,a)为ID3算法。 ○ [a,b)为CART算法。 ○ [b,n]为C4.5算法。 例如，在一个拥有5棵树的森林中，如果[2,4]表示0，则1为ID3算法，2,3为CART算法，4为C4.5算法。如果输入None，则算法在森林中均分。
	单棵树随机特征数	取值范围为[1,N]，N表示Feature数量。
	叶节点数据的最小个数	取值范围为正整数，默认值为2。
	叶节点数据个数占父节点的最小比例	取值范围为[0,1]，默认值为0。
	单颗树的最大深度	取值范围为[1,+∞)，默认值为无穷。
	单颗树输入的随机数据个数	取值范围为(1000,1000000]，默认值为100000。

- PAI命令方式

```
PAI -name randomforests
  -project algo_public
  -DinputTableName="pai_rf_test_input"
  -DmodelName="pai_rf_test_model"
  -DforceCategorical="f1"
  -DlabelColName="class"
  -DfeatureColNames="f0,f1"
  -DmaxRecordSize="100000"
  -DminNumPer="0"
  -DminNumObj="2"
  -DtreeNum="3";
```

参数	是否必选	描述	默认值
inputTableName	是	输入表。	无
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
labelColName	是	输入表中，标签列的列名。	无
modelName	是	输出的模型名。	无
treeNum	是	森林中树的数量，取值范围为1~1000。	100
excludedColNames	否	不参与训练的列，不能与featureColNames同时使用。	空
weightColName	否	输入表中的权重列名。	无
featureColNames	否	输入表中，用于训练的特征列名。	除labelColName与weightColName外的所有列
forceCategorical	否	解析规则如下： ○ STRING、BOOLEAN及DATETIME类型的列，解析为离散类型。 ○ DOUBLE和BIGINT类型的列，解析为连续类型。 **说明** 如果需要将BIGINT类型的列解析为CATEGORICAL，则必须使用forceCategorical参数指定类型。	INT为连续类型

参数	是否必选	描述	默认值
algorithmTypes	否	单棵树的算法在森林中的位置。如果有N棵树，且algorithmTypes=[a,b]，则： ○ [0,a)为ID3算法。 ○ [a,b)为CART算法。 ○ [b,n]为C4.5算法。 例如，在一个拥有5棵树的森林中，[2,4]表示0，则1为ID3算法，2,3为CART算法，4为C4.5算法。如果输入None，则算法在森林中均分。	算法在森林中均分
randomColNum	否	生成单棵树时，每次分裂选择的随机特征数量。取值范围为[1,N]，N表示Feature数量。	$\log_2 N$
minNumObj	否	叶节点数据的最小个数，取值范围为正整数。	2
minNumPer	否	叶节点数据个数占父节点的最小比例，取值范围为[0,1]。	0.0
maxTreeDeep	否	单颗树的最大深度，取值范围为[1,+∞)。	无穷
maxRecordSize	否	单棵树输入的随机数据个数，取值范围为(1000,1000000]。	100000

随机森林示例

1. 使用SQL语句，生成训练数据。

```
create table pai_rf_test_input as
select * from
(
 select 1 as f0,2 as f1, "good" as class from dual
 union all
 select 1 as f0,3 as f1, "good" as class from dual
 union all
 select 1 as f0,4 as f1, "bad" as class from dual
 union all
 select 0 as f0,3 as f1, "good" as class from dual
 union all
 select 0 as f0,4 as f1, "bad" as class from dual
)tmp;
```

2. 使用PAI命令，提交随机森林算法组件参数。

```
PAI -name randomforests
  -project algo_public
  -DinputTableName="pai_rf_test_input"
  -Dmodelname="pai_rf_test_model"
  -DforceCategorical="f1"
  -DlabelColName="class"
  -DfeatureColNames="f0,f1"
  -DmaxRecordSize="100000"
  -DminNumPer="0"
  -DminNumObj="2"
  -DtreeNum="3";
```

3. 查看模型PMML（Predictive Model Markup Language）。

```
<?xml version="1.0" encoding="utf-8"?
<PMML xmlns="http://www.dmg.org/PMML-4_2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="4.2" xsi:schemaLocation="http://www.dmg.org/PMML-4_2 http://www.dmg.org/v4-2/pmml-4-2.xsd"
 <Header copyright="Copyright (c) 2014, Alibaba Inc." description=""
  <Application name="ODPS/PMML" version="0.1.0"/
  <TimestampTue, 12 Jul 2016 07:04:48 GMT</Timestamp
 </Header
 <DataDictionary numberOfFields="2"
  <DataField name="f0" optype="continuous" dataType="integer"/
  <DataField name="f1" optype="continuous" dataType="integer"/
  <DataField name="class" optype="categorical" dataType="string"
   <Value value="bad"/
   <Value value="good"/
  </DataField
 </DataDictionary
 <MiningModel modelName="xlab_m_random_forests_1_75078_v0" functionName="classification" algorithmName="RandomForests"
  <MiningSchema
   <MiningField name="f0" usageType="active"/
   <MiningField name="f1" usageType="active"/
   <MiningField name="class" usageType="target"/
  </MiningSchema
  <Segmentation multipleModelMethod="majorityVote"
   <Segment id="0"
    <True/
    <TreeModel modelName="xlab_m_random_forests_1_75078_v0" functionName="classification" algorithmName="RandomForests"
     <MiningSchema
      <MiningField name="f0" usageType="active"/
      <MiningField name="f1" usageType="active"/
      <MiningField name="class" usageType="target"/
     </MiningSchema
     <Node id="1"
      <True/
      <ScoreDistribution value="bad" recordCount="2"/
      <ScoreDistribution value="good" recordCount="3"/
      <Node id="2" score="good"
       <SimplePredicate field="f1" operator="equal" value="2"/
       <ScoreDistribution value="good" recordCount="1"/
```

```
        </Node
        <Node id="3" score="good"
         <SimplePredicate field="f1" operator="equal" value="3"/
         <ScoreDistribution value="good" recordCount="2"/
        </Node
        <Node id="4" score="bad"
         <SimplePredicate field="f1" operator="equal" value="4"/
         <ScoreDistribution value="bad" recordCount="2"/
        </Node
       </Node
      </TreeModel
     </Segment
     <Segment id="1"
      <True/
      <TreeModel modelName="xlab_m_random_forests_1_75078_v0" functionName="classification" alg
orithmName="RandomForests"
       <MiningSchema
        <MiningField name="f0" usageType="active"/
        <MiningField name="f1" usageType="active"/
        <MiningField name="class" usageType="target"/
       </MiningSchema
       <Node id="1"
        <True/
        <ScoreDistribution value="bad" recordCount="2"/
        <ScoreDistribution value="good" recordCount="3"/
        <Node id="2" score="good"
         <SimpleSetPredicate field="f1" booleanOperator="isIn"
          <Array n="2" type="integer"2 3</Array
         </SimpleSetPredicate
         <ScoreDistribution value="good" recordCount="3"/
        </Node
        <Node id="3" score="bad"
         <SimpleSetPredicate field="f1" booleanOperator="isNotIn"
          <Array n="2" type="integer"2 3</Array
         </SimpleSetPredicate
         <ScoreDistribution value="bad" recordCount="2"/
        </Node
       </Node
      </TreeModel
     </Segment
     <Segment id="2"
      <True/
      <TreeModel modelName="xlab_m_random_forests_1_75078_v0" functionName="classification" alg
orithmName="RandomForests"
       <MiningSchema
        <MiningField name="f0" usageType="active"/
        <MiningField name="f1" usageType="active"/
        <MiningField name="class" usageType="target"/
       </MiningSchema
       <Node id="1"
        <True/
        <ScoreDistribution value="bad" recordCount="2"/
        <ScoreDistribution value="good" recordCount="3"/
        <Node id="2" score="bad"
         <SimplePredicate field="f0" operator="lessOrEqual" value="0.5"/
```

```
        <SimplePredicate field="f0" operator="lessOrEqual" value="0.5"/
        <ScoreDistribution value="bad" recordCount="1"/
        <ScoreDistribution value="good" recordCount="1"/
       </Node
       <Node id="3" score="good"
        <SimplePredicate field="f0" operator="greaterThan" value="0.5"/
        <ScoreDistribution value="bad" recordCount="1"/
        <ScoreDistribution value="good" recordCount="2"/
       </Node
      </Node
     </TreeModel
    </Segment
   </Segmentation
  </MiningModel
</PMML
```

4. 查看模型可视化输出。

朴素贝叶斯

朴素贝叶斯是一种基于独立假设的贝叶斯定理的概率分类算法。PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**特征列**	默认为除标签列外的所有列，支持DOUBLE、STRING及BIGINT数据类型。
	排除列	不参与训练的列，不能与**选择特征列**同时使用。
	强制转换列	解析规则如下： ○ STRING、BOOLEAN及DATETIME类型的列，解析为离散类型。 ○ DOUBLE和BIGINT类型的列，解析为连续类型。 **说明** 如果需要将BIGINT类型的列解析为CATEGORICAL，则必须使用forceCategorical参数指定类型。
	标签列	输入表的标签列，只能选择非特征列。支持STRING、DOUBLE及BIGINT类型。

页签	参数	描述
	输入数据是否为稀疏格式	使用KV格式表示稀疏数据。
	当输入为稀疏时，K:V间的分隔符	默认为英文逗号（,）。
	当输入为稀疏时，key和value的分隔符	默认为英文冒号（:）。
执行调优	**计算核心数**	默认为系统自动分配。
	每个核心内存数	默认为系统自动分配。

- PAI命令方式

```
PAI -name NaiveBayes -project algo_public
  -DinputTablePartitions="pt=20150501"
  -DmodelName="xlab_m_NaiveBayes_23772"
  -DlabelColName="poutcome"
  -DfeatureColNames="age,previous,cons_conf_idx,euribor3m"
  -DinputTableName="bank_data_partition";
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的表名。	无
inputTablePartitions	否	输入表中，参与训练的分区。	所有分区
modelName	是	输出的模型名称。	无
labelColName	是	输入表中，标签列的名称。	无
featureColNames	否	输入表中，用于训练的特征列名。	除标签列外的所有列
excludedColNames	否	用于反选特征列，该参数不能与featureColNames同时使用。	空
forceCategorical	否	解析规则如下： ○ STRING、BOOLEAN及DATETIME类型的列，解析为离散类型。 ○ DOUBLE和BIGINT类型的列，解析为连续类型。 **说明** 如果需要将BIGINT类型的列解析为CATEGORICAL，则必须使用forceCategorical参数指定类型。	INT为连续类型
coreNum	否	计算的核心数。	系统自动分配

参数	是否必选	描述	默认值
memSizePerCore	否	每个核心的内存，取值范围为1 MB~65536 MB。	系统自动分配

朴素贝叶斯示例

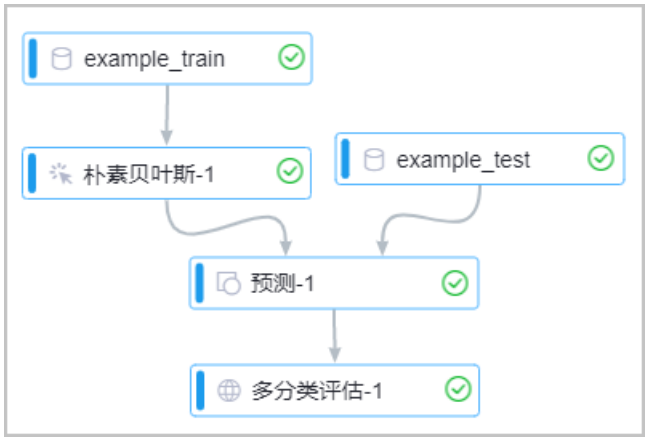
1. 准备如下训练数据。

id	y	f0	f1	f2	f3	f4	f5	f6	f7
1	-1	-0.294118	0.487437	0.180328	-0.292929	-1	0.00149028	-0.53117	-0.0333333
2	+1	-0.882353	-0.145729	0.0819672	-0.414141	-1	-0.207153	-0.766866	-0.666667
3	-1	-0.0588235	0.839196	0.0491803	-1	-1	-0.305514	-0.492741	-0.633333
4	+1	-0.882353	-0.105528	0.0819672	-0.535354	-0.777778	-0.162444	-0.923997	-1
5	-1	-1	0.376884	-0.344262	-0.292929	-0.602837	0.28465	0.887276	-0.6
6	+1	-0.411765	0.165829	0.213115	-1	-1	-0.23696	-0.894962	-0.7
7	-1	-0.647059	-0.21608	-0.180328	-0.353535	-0.791962	-0.0760059	-0.854825	-0.833333
8	+1	0.176471	0.155779	-1	-1	-1	0.052161	-0.952178	-0.733333
9	-1	-0.764706	0.979899	0.147541	-0.0909091	0.283688	-0.0909091	-0.931682	0.0666667
10	-1	-0.0588235	0.256281	0.57377	-1	-1	-1	-0.868488	0.1

2. 准备如下测试数据。

id	y	f0	f1	f2	f3	f4	f5	f6	f7
1	+1	-0.882353	0.0854271	0.442623	-0.616162	-1	-0.19225	-0.725021	-0.9
2	+1	-0.294118	-0.0351759	-1	-1	-1	-0.293592	-0.904355	-0.766667
3	+1	-0.882353	0.246231	0.213115	-0.272727	-1	-0.171386	-0.981213	-0.7
4	-1	-0.176471	0.507538	0.278689	-0.414141	-0.702128	0.0491804	-0.475662	0.1
5	-1	-0.529412	0.839196	-1	-1	-1	-0.153502	-0.885568	-0.5
6	+1	-0.882353	0.246231	-0.0163934	-0.353535	-1	0.0670641	-0.627669	-1
7	-1	-0.882353	0.819095	0.278689	-0.151515	-0.307329	0.19225	0.00768574	-0.966667
8	+1	-0.882353	-0.0753769	0.0163934	-0.494949	-0.903073	-0.418778	-0.654996	-0.866667
9	+1	-1	0.527638	0.344262	-0.212121	-0.356974	0.23696	-0.836038	-0.8
10	+1	-0.882353	0.115578	0.0163934	-0.737374	-0.56974	-0.28465	-0.948762	-0.933333

3. 构建实验，详情请参见算法建模。



4. 配置朴素贝叶斯组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**特征列**	在训练表中，选择f0、f1、f2、f3、f4、f5、f6及f7列。
	标签列	在训练表中，选择y列。

5. 运行实验，查看预测结果。

id	y	prediction_result	prediction_score	prediction_detail
1	+1	+1	1.1253341740304466	{"+1": 1.125334174030447, "-1": -4.116083356256129}
2	+1	+1	2.1096017126455773	{"+1": 2.109601712645577, "-1": -9.431111061253313}
3	+1	+1	1.0119555555558313	{"+1": 1.011955555555831, "-1": -3.065398632191728}
4	-1	-1	-2.435966936637894	{"+1": -33.18118539397123, "-1": -2.435966936637894}
5	-1	-1	-8.569186264886811	{"+1": -10.87241674956984, "-1": -8.569186264886811}
6	+1	-1	-3.7653558619317407	{"+1": -4.842928791659737, "-1": -3.765355861931741}
7	-1	-1	-4.895262140022778	{"+1": -89.31454432308786, "-1": -4.895262140022778}
8	+1	+1	-2.701748842959141	{"+1": -2.701748842959141, "-1": -3.683850999783979}
9	+1	-1	-4.321549531283424	{"+1": -20.76204601810873, "-1": -4.321549531283424}
10	+1	+1	-3.088051422948976	{"+1": -3.088051422948976, "-1": -3.667255025859788}

3.5.3. 聚类

本文为您介绍PAI-Studio提供的聚类算法，包括K均值聚类和DBSCAN。

K均值聚类

K均值聚类首先随机选择K个对象作为每个簇的初始聚类中心，然后计算剩余对象与各簇中心的距离，将其分配至距离最近的簇，再重新计算每个簇的聚类中心。该算法假设聚类对象为空间向量，且以各聚类内部的均方误差和最小为目标，不断地进行计算迭代，直到准则函数收敛。

PAI-Studio支持通过可视化或PAI命令方式，配置K均值聚类组件的参数：

- 可视化方式

页签	参数	描述
字段设置	特征列	支持DOUBLE及INT数据类型。
	附加列	附加输出至聚类结果表的输入列，列名以英文逗号（,）分隔。
	输入为稀疏矩阵	使用KV格式表示稀疏数据。
	kv键间分隔符	默认为英文逗号（,）。
	kv键内分隔符	默认为英文冒号（:）。
参数设置	聚类数	取值范围为1~1000。
	距离度量方式	支持Euclidean、Cosine及Cityblock方式。
	质心初始化方法	支持Random、First K、Uniform、K-Means++及使用初始质心表方法。
	最大迭代次数	取值范围为1~1000。
	收敛标准	迭代终止条件。
	初始随机种子	默认值为当前时间。如果seed为固定值，则聚类结果稳定。
执行调优	核心数	默认为系统自动分配。
	每个核的内存大小	默认为系统自动分配。

- PAI命令方式

```
pai -name kmeans
  -project algo_public
  -DinputTableName=pai_kmeans_test_input
  -DselectedColNames=f0,f1
  -DappendColNames=f0,f1
  -DcenterCount=3
  -Dloop=10
  -Daccuracy=0.01
  -DdistanceType=euclidean
  -DinitCenterMethod=random
  -Dseed=1
  -DmodelName=pai_kmeans_test_output_model_
  -DidxTableName=pai_kmeans_test_output_idx
  -DclusterCountTableName=pai_kmeans_test_output_couter
  -DcenterTableName=pai_kmeans_test_output_center;
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的表名。	无

参数	是否必选	描述	默认值
selectedColNames	否	输入表中，参与训练的列。列名以英文逗号（,）分隔，支持INT和DOUBLE类型。如果输入为稀疏格式，则支持STRING类型的列。	所有列
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 说明 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
appendColNames	否	附加输出至聚类结果表的输入列，列名以英文逗号（,）分隔。	无
enableSparse	否	输入表是否为稀疏矩阵，取值范围为{true,false}。	false
itemDelimiter	否	KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	key和value之间的分隔符。	英文冒号（:）
centerCount	是	聚类数，取值范围为1~1000。	10
distanceType	否	距离度量方式，支持以下类型： ○ euclidean：欧式距离 d(x-c)=(x-c)(x-c)' 。 ○ cosine：夹角余弦的计算方式如下。 $$d(x-c)=0.5-0.5*\frac{xc'}{\sqrt{xx'}\sqrt{cc'}}$$ ○ cityblock：曼哈顿距离 d(x-c)=|x-c| 。	euclidean
initCenterMethod	否	质心初始化的方法，支持以下方法： ○ random：从输入数据表中，随机采样K个初始中心点，初始随机种子由参数seed指定。 ○ topk：从输入数据表，读取前K行作为初始中心点。 ○ uniform：从最小值到最大值，均匀地计算K个初始中心点。 ○ kmpp：通过K-Means++算法，获得K个初始中心点。 ○ external：指定额外的初始质心表。	random
initCenterTableName	否	初始质心表的名称。如果initCenterMethod为external，则该参数生效。	无
loop	否	最大迭代次数，取值范围为1~1000。	100

参数	是否必选	描述	默认值
accuracy	否	算法终止条件。如果两次迭代的目标差小于该值，则算法终止。	0.1
seed	否	初始随机种子。	当前时间
modelName	否	输出模型的名称。	无
idxTableName	是	输出聚类结果表，包括聚类后每条记录所属的类号。	无
idxTablePartition	否	聚类结果表的分区。	无
clusterCountTableName	否	聚类统计表，统计各聚类包含的点数量。	无
centerTableName	否	聚类中心表。	无
coreNum	否	节点数量，与memSizePerCore搭配使用。取值范围为1~9999。	系统自动分配
memSizePerCore	否	每个节点的内存大小，取值范围为1024 MB~64*1024 MB。	系统自动分配
lifecycle	否	输出表的生命周期。	无

K均值聚类输出数据说明

K均值聚类输出聚类结果表、聚类统计表及聚类中心表。输出格式如下：

- 聚类结果表

列名	描述
appendColNames	附加列。
cluster_index	训练表中，每个样本被分配到的簇。
distance	训练表中，每个样本到簇中心的距离。

- 聚类统计表

列名	描述
cluster_index	簇编号。
cluster_count	每个簇中的样本数量。

- 聚类中心表

列名	描述
cluster_index	簇编号。
selectedColNames	训练表中，参与训练的列。

K均值聚类示例

以稠密格式数据作为输入：

1. 您可以通过以下任何一种方式，生成测试数据：
   - 使用初始质心表的方式

```
create table pai_kmeans_test_init_center as
select * from
(
select 1 as f0,2 as f1 from dual
union all
select 1 as f0,3 as f1 from dual
union all
select 1 as f0,4 as f1 from dual
)tmp;
```

   - 使用其他初始质心的方式

```
create table pai_kmeans_test_input as
select * from
(
 select 'id1' as id,1 as f0,2 as f1 from dual
 union all
 select 'id2' as id,1 as f0,3 as f1 from dual
 union all
 select 'id3' as id,1 as f0,4 as f1 from dual
 union all
 select 'id4' as id,0 as f0,3 as f1 from dual
 union all
 select 'id5' as id,0 as f0,4 as f1 from dual
)tmp;
```

2. 使用PAI命令，提交K均值聚类算法组件参数：
   - 使用初始质心表的方式

```
drop table if exists pai_kmeans_test_output_idx;
yes
drop table if exists pai_kmeans_test_output_couter;
yes
drop table if exists pai_kmeans_test_output_center;
yes
drop offlinemodel if exists pai_kmeans_test_output_model_;
yes
pai -name kmeans
  -project algo_public
  -DinputTableName=pai_kmeans_test_input
  -DinitCenterTableName=pai_kmeans_test_init_center
  -DselectedColNames=f0,f1
  -DappendColNames=f0,f1
  -DcenterCount=3
  -Dloop=10
  -Daccuracy=0.01
  -DdistanceType=euclidean
  -DinitCenterMethod=external
  -Dseed=1
  -DmodelName=pai_kmeans_test_output_model_
  -DidxTableName=pai_kmeans_test_output_idx
  -DclusterCountTableName=pai_kmeans_test_output_couter
  -DcenterTableName=pai_kmeans_test_output_center;
```

- 使用随机初始质心的方式

```
drop table if exists pai_kmeans_test_output_idx;
yes
drop table if exists pai_kmeans_test_output_couter;
yes
drop table if exists pai_kmeans_test_output_center;
yes
drop offlinemodel if exists pai_kmeans_test_output_model_;
yes
pai -name kmeans
  -project algo_public
  -DinputTableName=pai_kmeans_test_input
  -DselectedColNames=f0,f1
  -DappendColNames=f0,f1
  -DcenterCount=3
  -Dloop=10
  -Daccuracy=0.01
  -DdistanceType=euclidean
  -DinitCenterMethod=random
  -Dseed=1
  -DmodelName=pai_kmeans_test_output_model_
  -DidxTableName=pai_kmeans_test_output_idx
  -DclusterCountTableName=pai_kmeans_test_output_couter
  -DcenterTableName=pai_kmeans_test_output_center;
```

3. 查看聚类结果表、聚类统计表及聚类中心表：
   - 聚类结果表idxTableName

```
+------------+------------+---------------+------------+
| f0     | f1     | cluster_index | distance  |
+------------+------------+---------------+------------+
| 1      | 2      | 0         | 0.0     |
| 1      | 3      | 1         | 0.5     |
| 1      | 4      | 2         | 0.5     |
| 0      | 3      | 1         | 0.5     |
| 0      | 4      | 2         | 0.5     |
+------------+------------+---------------+------------+
```

- 聚类统计表clusterCountTableName

```
+---------------+---------------+
| cluster_index | cluster_count |
+---------------+---------------+
| 0         | 1         |
| 1         | 2         |
| 2         | 2         |
+---------------+---------------+
```

- 聚类中心表centerTableName

```
+---------------+------------+------------+
| cluster_index | f0     | f1     |
+---------------+------------+------------+
| 0         | 1.0    | 2.0    |
| 1         | 0.5    | 3.0    |
| 2         | 0.5    | 4.0    |
+---------------+------------+------------+
```

以稀疏格式数据作为输入：

1. 生成测试数据。

```
create table pai_kmeans_test_sparse_input as
select * from
(
 select 1 as id,"s1" as id_s,"0:0.1,1:0.2" as kvs0,"2:0.3,3:0.4" as kvs1 from dual
 union all
 select 2 as id,"s2" as id_s,"0:1.1,2:1.2" as kvs0,"4:1.3,5:1.4" as kvs1 from dual
 union all
 select 3 as id,"s3" as id_s,"0:2.1,3:2.2" as kvs0,"6:2.3,7:2.4" as kvs1 from dual
 union all
 select 4 as id,"s4" as id_s,"0:3.1,4:3.2" as kvs0,"8:3.3,9:3.4" as kvs1 from dual
 union all
 select 5 as id,"s5" as id_s,"0:5.1,5:5.2" as kvs0,"10:5.3,6:5.4" as kvs1 from dual
)tmp;
```

稀疏格式数据作为输入时，使用0填充缺失的列。如果多个列同时作为输入，则会被合并。例如，kvs0和kvs1同时作为输入，则第一行的实际数据如下。

```
0:0.1,1:0.2,2:0.3,3:0.4,4:0,5:0,6:0,7:0,8:0,9:0,10:0
```

示例中的稀疏矩阵列从0开始编号，矩阵共5行11列。如果kvs中的某列包含 `123456789:0.1` ，则稀疏矩阵变为5行123456789列，该矩会消耗大量CPU和内存。对于kvs中存在异常列编号的原始数据，建议重

新进行列编号以减小矩阵规模。

2. 使用PAI命令，提交K均值聚类组件的参数。

```
pai -name kmeans
 -project algo_public
 -DinputTableName=pai_kmeans_test_sparse_input
 -DenableSparse=true
 -DselectedColNames=kvs0,kvs1
 -DappendColNames=id,id_s
 -DitemDelimiter=,
 -DkvDelimiter=:
 -DcenterCount=3
 -Dloop=100
 -Daccuracy=0.01
 -DdistanceType=euclidean
 -DinitCenterMethod=topk
 -Dseed=1
 -DmodelName=pai_kmeans_test_input_sparse_output_model
 -DidxTableName=pai_kmeans_test_sparse_output_idx
 -DclusterCountTableName=pai_kmeans_test_sparse_output_couter
 -DcenterTableName=pai_kmeans_test_sparse_output_center;
```

3. 查看聚类结果表、聚类统计表及聚类中心表：
   - 聚类结果表idxTableName

```
+------------+------------+---------------+------------+
| id      | id_s     | cluster_index | distance  |
+------------+------------+---------------+------------+
| 4       | s4       | 0             | 2.90215437218629 |
| 5       | s5       | 1             | 0.0      |
| 1       | s1       | 2             | 0.7088723439378913 |
| 2       | s2       | 2             | 1.1683321445547923 |
| 3       | s3       | 0             | 2.0548722588034516 |
+------------+------------+---------------+------------+
```

   - 聚类统计表clusterCountTableName

```
+---------------+---------------+
| cluster_index | cluster_count |
+---------------+---------------+
| 0             | 2             |
| 1             | 1             |
| 2             | 2             |
+---------------+---------------+
```

   - 聚类中心表centerTableName

```
+---------------+------------+------------+
| cluster_index | kvs0     | kvs1     |
+---------------+------------+------------+
| 0             | 0:2.6,1:0,2:0,3:1.1,4:1.6,5:0 | 6:1.15,7:1.2,8:1.65,9:1.7,10:0 |
| 1             | 0:5.1,1:0,2:0,3:0,4:0,5:5.2 | 6:5.4,7:0,8:0,9:0,10:5.3 |
| 2             | 0:0.6,1:0.1,2:0.75,3:0.2,4:0.65,5:0.7 | 6:0,7:0,8:0,9:0,10:0 |
+---------------+------------+------------+
```

相关说明：

- 如果使用夹角余弦距离，则某些聚类可能为空，即聚类数量小于K。因为初始化的K个中心点（向量）可能是平行向量，所以按顺序遍历中心点时，样本不会被分配至后面的中心点（平行向量）。建议通过外部输入中心表的方式，使用线下准备好的K个中心点。
- 如果输入表中存在NULL或空值，则系统报错 `Algo Job Failed-System Error-Null feature value found` 。建议使用缺省值进行填充。
- 使用稀疏格式数据作输入时，如果最大列编号超过2,000,000，则系统报错 `Algo Job Failed-System Error-Feature count can’t be more than 2000000` 。建议从0或1开始，重新对列进行编号。
- 如果中心点模型过大导致写失败，则系统报错 `Algo Job Failed-System Error-kIOError:Write failed for message: comparison_measure` 。建议将稀疏矩阵的列从0或1开始，重新编号。如果模型规模 `col*centerCount >27,0000,000` ，则只能通过命令行的方式，去除modelName参数，再重新执行聚类。
- 如果输入表的列名存在SQL关键字，则系统报错 `FAILED: Failed Task createCenterTable:kOtherError:ODPS-0130161:[1,558] Parse exception - invalid token ‘,’, expect “)” ` 。

DBSCAN

DBSCAN（Density Based Spatial Clustering of Applications with Noise）是一种基于数据密度的无监督聚类算法。在聚类空间的特定区域内，使用指定的半径阈值和数量阈值，筛选核心点及其领域点。通过密度可达及密度相连理论，实现数据点聚类。

PAI-Studio支持通过可视化或PAI命令方式，配置DBSCAN组件的参数：

- 可视化方式

页签	参数	描述
参数设置	**输入数据类型**	支持**邻接表**和**向量**类型。
	数据向量的维度	输入数据的向量维度。如果**输入数据类型**为**向量**，则该参数必选。 说明 如果**输入表的格式**为**多列**，则**数据向量的维度**必须与**选择数据所在列**的数量一致。
	邻域点的距离阈值	如果两点之间的距离小于该阈值，则互为邻域点。如果**输入数据类型**为**向量**，则该参数必选。
	核心对象密度阈值	如果某点邻域内的点数量大于该阈值，则该点为核心对象。
	输入表的格式	如果**输入数据类型**为**向量**，则该参数必选。系统支持以下格式： ○ **多列**：使用多列表示向量。 ○ **两列**：使用一列表示向量，各维度通过英文逗号（,）分隔。
字段信息	**选择数据所在列**	如果**输入表的格式**为**多列**，则该参数必选。
	Server机器数量	Server服务器的数量。
	Worker机器数量	Worker服务器的数量。

页签	参数	描述
执行调优	Server每台机器的CPU数	每台Server服务器的CPU数量。
	Worker每台机器的CPU数	每台Worker服务器的CPU数量。
	Worker每台机器内存数	每台Worker服务器的内存，单位为MB。
	Server每台机器内存数	每台Server服务器的内存，单位为MB。

- PAI命令方式

  DBSCAN算法支持以邻接表或向量作为输入数据，且支持以多列或两列的格式表示向量。具体的PAI命令如下：

  - 以邻接表作为输入

```
pai -name ps_dbscan
-DinputTable=hxdb_neighbor_data_order
-DinputType="1"
-DoutputTable="hxtmp2"
-DminPoints="4"
-DserverNum="1"
-DserverCpu="300"
-DserverMemory="3000"
-DworkerNum="2"
-DworkerCpu="800"
-DworkerMemory="2000"
```

  - 以多列格式的向量作为输入

```
pai -name ps_dbscan
-DinputTable=hxdb_multicols_data
-DinputType="0"
-DoutputTable="hxtmp"
-DdataType="DenseMultiCols"
-DpointDim="12"
-Deps="4"
-DminPoints="20"
-DselectedColIds="all"
-DserverNum="2"
-DserverCpu="300"
-DserverMemory="3000"
-DworkerNum="10"
-DworkerCpu="800"
-DworkerMemory="2000"
```

- 以两列格式的向量作为输入

```
pai -name ps_dbscan
-DinputTable="hxdb_sample_60w"
-DinputType="0"
-DoutputTable="hxtmp1"
-DdataType="Dense2Cols"
-DpointDim="2"
-Deps="0.01"
-DminPoints="10"
-DselectedColIds="all"
-DserverNum="2"
-DserverCpu="300"
-DserverMemory="3000"
-DworkerNum="10"
-DworkerCpu="800"
-DworkerMemory="2000"
```

参数	是否必选	描述	默认值
inputTable	是	输入表的名称。	无
outputTable	是	输出表的名称。	无
inputType	否	输入数据的类型。支持以下类型： ○ 0：表示以向量形式输入。 ○ 1：表示以邻接表形式输入。	0
pointDim	否	输入数据的向量维度。如果inputType为0，则该参数必选。 **说明** 如果dataType为DenseMultiCols，则pointDim必须与selectedColIds的数量一致。	10
eps	否	邻域点的距离阈值。如果两点之间的距离小于该阈值，则互为邻域点。如果inputType为0，则该参数必选。	1.0
minPoints	否	核心对象的密度阈值。如果某点邻域内的点数量大于该阈值，则该点为核心对象。	10
dataType	否	输入表的格式。如果inputType为0，则该参数必选。系统支持以下格式： ○ DenseMultiCols：使用多列表示向量。 ○ Dense2Cols：使用一列表示向量，各维度通过英文逗号（,）分隔。	Dense2Cols

参数	是否必选	描述	默认值
selectedColIds	否	数据所在的列。如果dataType为DenseMultiCols，则该参数必选。取值格式为all或0,1,3。 说明 ID列必须位于首位。	all
serverNum	是	Server服务器的数量。	5
workerNum	是	Worker服务器的数量。	30
serverCpu	是	每台Server服务器的CPU数量。	8
workerCpu	是	每台Worker服务器的CPU数量。	8
workerMemory	是	每台Worker服务器的内存，单位为MB。	10000
serverMemory	是	每台Server服务器的内存，单位为MB。	10000

DBSCAN输入数据说明

DBSCAN算法支持以邻接表或向量作为输入数据，且支持以多列或两列的格式表示向量。输入数据的示例如下：

- 邻接表

```
+-------------+------------+
| mid(bigint) | f1(string) |
+-------------+------------+
| 0           | 2,3,0      |
| 1           | 1,2,3,4    |
| 2           | 2,1,5      |
| 3           | 1,3        |
| 4           | 1,4        |
| 5           | 2,5,1,0    |
+-------------+------------+
```

> 说明 某点的邻域点必须包含该点本身。例如，0号点的邻域点必须包含0点。

- 多列格式的向量（以二维向量为例）

```
+--------------+------------+------------+
| mid(bigint)  | f1(double) | f2(double) |
+--------------+------------+------------+
| 0            | 0.0        | 0.3        |
| 1            | 0.0        | 1.0        |
| 2            | 0.0        | 0.1        |
| 3            | 1.0        | 0.0        |
| 4            | 0.0        | 0.2        |
| 5            | 1.0        | 0.2        |
+--------------+------------+------------+
```

第一列为样本ID。第二列和第三列分别为向量各维度的值。

- 两列格式的向量（以二维向量为例）

```
+--------------+------------+
| mid(bigint) | f1(string) |
+--------------+------------+
| 0        | 0.0,0.3   |
| 1        | 0.0,1.0   |
| 2        | 0.0,0.1   |
| 3        | 1.0,0.0   |
| 4        | 0.0,0.2   |
| 5        | 1.0,0.2   |
+--------------+------------+
```

第一列为样本ID。第二列为向量各维度的值，通过英文逗号（,）分隔。

> **说明** 通过**数据预处理**下的**增加序号列**组件，可以为每条样本增加ID列。

3.5.4. 回归

本文为您介绍PAI-Studio提供的回归算法，包括GBDT回归、线性回归、PS-SMART回归及PS线性回归。

GBDT回归

梯度渐进回归树GBDT（Gradient Boosting Decesion Tree）是一种迭代决策树算法，适用于线性及非线性回归场景。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**输入列**	输入数据源中，参与训练的特征列。支持DOUBLE及BIGINT类型。 **说明** 特征列数量不能超过800。
	标签列	支持DOUBLE及BIGINT类型。
	分组列	支持DOUBLE及BIGINT类型，默认将全表作为一组。
	损失函数类型	支持gbrank loss、lambdamart dcg loss、lambdamart ndcg loss及regression loss类型。
	gbrank loss中的Tau参数	取值范围为[0,1]。
	gbrank与regression loss中的指数底数	取值范围为[1,10]。

页签	参数	描述
参数设置	metric类型	支持NDCG及DCG类型。
	树数量	取值范围为1~10000。
	学习速率	取值范围为(0,1)。
	最大叶子数	取值范围为1~1000。
	一棵树的最大深度	取值范围为1~100。
	叶子节点容纳的最少样本数	取值范围为1~1000。
	样本采样比例	取值范围为(0,1)。
	训练中采集的特征比例	取值范围为(0,1)。
	测试样本数比例	取值范围为[0,1)。
	随机数产生器种子	取值范围为[0,10]。
	是否使用newton方法来学习	使用Newton方法的开关。
	一个特征分裂的最大数量	取值范围为1~1000。
执行调优	计算核心数	系统根据输入数据量，自动分配训练的实例数量。
	每个核内存	系统根据输入数据量，自动分配内存。单位为MB。

- PAI命令方式

```
PAI -name gbdt
  -project algo_public
  -DfeatureSplitValueMaxSize="500"
  -DlossType="0"
  -DrandSeed="0"
  -DnewtonStep="0"
  -Dshrinkage="0.05"
  -DmaxLeafCount="32"
  -DlabelColName="campaign"
  -DinputTableName="bank_data_partition"
  -DminLeafSampleCount="500"
  -DsampleRatio="0.6"
  -DgroupIDColName="age"
  -DmaxDepth="11"
  -DmodelName="xlab_m_GBDT_83602"
  -DmetricType="2"
  -DfeatureRatio="0.6"
  -DinputTablePartitions="pt=20150501"
  -Dtau="0.6"
  -Dp="1"
  -DtestRatio="0.0"
  -DfeatureColNames="previous,cons_conf_idx,euribor3m"
  -DtreeCount="500"
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
featureColNames	否	输入表中，用于训练的特征列名。支持DOUBLE及BIGINT类型。	所有数值列
labelColName	是	输入表中的标签列名，支持DOUBLE及BIGINT类型。	无
inputTablePartitions	否	输入表中，参与训练的分区。支持的格式包括： ◦ Partition_name=value ◦ name1=value1/name2=value2：多级分区 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	所有分区
modelName	是	输出的模型名称。	无
outputImportanceTableName	否	输出特征重要性的表名。	无
groupIDColName	否	数据分组列。	全表

参数	是否必选	描述	默认值
lossType	否	损失函数包括以下类型： ○ 0：GBRANK ○ 1：LAMBDAMART_DCG ○ 2：LAMBDAMART_NDCG ○ 3：LEAST_SQUARE ○ 4：LOG_LIKELIHOOD	0
metricType	否	包括以下类型： ○ 0：NDCG（Normalized Discounted Cumulative Gain） ○ 1：DCG（Discounted Cumulative Gain） ○ 2：AUC，仅适用于label取值为0/1的场景	0
treeCount	否	树数量，取值范围为1~10000。	500
shrinkage	否	学习速率，取值范围为(0,1)。	0.05
maxLeafCount	否	最大叶子数，取值范围为1~1000。	32
maxDepth	否	树的最大深度，取值范围为1~100。	10
minLeafSampleCount	否	叶子节点容纳的最少样本数，取值范围为1~1000。	500
sampleRatio	否	训练采集的样本比例，取值范围为(0,1)。	0.6
featureRatio	否	训练采集的特征比例，取值范围为(0,1)。	0.6
tau	否	GBRank Loss中的Tau参数，取值范围为[0,1]。	0.6
p	否	GBRank Loss中的p参数，取值范围为[1,10]。	1
randSeed	否	随机数种子，取值范围为[0,10]。	0
newtonStep	否	使用Newton迭代法的开关，取值范围为{0,1}。	1
featureSplitValueMaxSize	否	特征分裂的最大数量，取值范围为1~1000。	500
lifecycle	否	输出表的生命周期。	无

GBDT回归示例

1. 使用SQL语句，生成测试数据。

```
drop table if exists gbdt_ls_test_input;
create table gbdt_ls_test_input
as
```

```
select
  *
from
(
  select
    cast(1 as double) as f0,
    cast(0 as double) as f1,
    cast(0 as double) as f2,
    cast(0 as double) as f3,
    cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(1 as double) as f1,
      cast(0 as double) as f2,
      cast(0 as double) as f3,
      cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(0 as double) as f1,
      cast(1 as double) as f2,
      cast(0 as double) as f3,
      cast(1 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(0 as double) as f1,
      cast(0 as double) as f2,
      cast(1 as double) as f3,
      cast(1 as bigint) as label
  from dual
  union all
    select
      cast(1 as double) as f0,
      cast(0 as double) as f1,
      cast(0 as double) as f2,
      cast(0 as double) as f3,
      cast(0 as bigint) as label
  from dual
  union all
    select
      cast(0 as double) as f0,
      cast(1 as double) as f1,
      cast(0 as double) as f2,
      cast(0 as double) as f3,
      cast(0 as bigint) as label
  from dual
) a;
```

生成的测试数据表gbdt_ls_test_input如下。

f0	f1	f2	f3	label
1.0	0.0	0.0	0.0	0
0.0	0.0	1.0	0.0	1
0.0	0.0	0.0	1.0	1
0.0	1.0	0.0	0.0	0
1.0	0.0	0.0	0.0	0
0.0	1.0	0.0	0.0	0

2. 使用PAI命令，提交GBDT回归组件的训练参数。

```
drop offlinemodel if exists gbdt_ls_test_model;
PAI -name gbdt
  -project algo_public
  -DfeatureSplitValueMaxSize="500"
  -DlossType="3"
  -DrandSeed="0"
  -DnewtonStep="1"
  -Dshrinkage="0.5"
  -DmaxLeafCount="32"
  -DlabelColName="label"
  -DinputTableName="gbdt_ls_test_input"
  -DminLeafSampleCount="1"
  -DsampleRatio="1"
  -DmaxDepth="10"
  -DmetricType="0"
  -DmodelName="gbdt_ls_test_model"
  -DfeatureRatio="1"
  -Dp="1"
  -Dtau="0.6"
  -DtestRatio="0"
  -DfeatureColNames="f0,f1,f2,f3"
  -DtreeCount="10"
```

3. 使用PAI命令，提交预测组件参数。

```
drop table if exists gbdt_ls_test_prediction_result;
PAI -name prediction
  -project algo_public
  -DdetailColName="prediction_detail"
  -DmodelName="gbdt_ls_test_model"
  -DitemDelimiter=","
  -DresultColName="prediction_result"
  -Dlifecycle="28"
  -DoutputTableName="gbdt_ls_test_prediction_result"
  -DscoreColName="prediction_score"
  -DkvDelimiter=":"
  -DinputTableName="gbdt_ls_test_input"
  -DenableSparse="false"
  -DappendColNames="label"
```

4. 查看预测结果表gbdt_ls_test_prediction_result。

label	prediction_result	prediction_score	prediction_detail
0	NULL	0.0	{ “label” : 0}
0	NULL	0.0	{ “label” : 0}
1	NULL	0.9990234375	{ “label” : 0.9990234375}
1	NULL	0.9990234375	{ “label” : 0.9990234375}
0	NULL	0.0	{ “label” : 0}
0	NULL	0.0	{ “label” : 0}

线性回归

线性回归（Linear Regression）是分析因变量和多个自变量之间的线性关系模型。PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**选择特征列**	输入数据源中，参与训练的特征列。
	选择标签列	支持DOUBLE及BIGINT类型。
	是否稀疏格式	使用KV格式表示稀疏格式。
	kv对间分隔符	默认使用英文逗号（,）分隔。
	key与value分隔符	默认使用英文冒号（:）分隔。
	最大迭代轮数	算法进行的最大迭代次数。
	最小似然误差	如果两次迭代间的Log Likelihood之差小于该值，则算法终止。

页签	参数	描述
参数设置	**正则化类型**	支持L1、L2及None类型。
	正则系数	如果**正则化类型**为None，则该参数失效。
	生成模型评估表	指标包括R-Squared、AdjustedR-Squared、AIC、自由度、残差的标准差及偏差。
	回归系数评估	指标包括T值、P值及置信区间[2.5%,97.5%]。只有选中**生成模型评估表**复选框，该参数才生效。
执行调优	**计算核心数**	默认为系统自动分配。
	每核内存大小	默认为系统自动分配。

- PAI命令方式

```
PAI -name linearregression
  -project algo_public
  -DinputTableName=lm_test_input
  -DfeatureColNames=x
  -DlabelColName=y
  -DmodelName=lm_test_input_model_out;
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
modelName	是	输出模型的名称。	无
outputTableName	否	输出的模型评估表名称。如果enableFitGoodness为true，则该参数必选。	无
labelColName	是	因变量，支持DOUBLE及BIGINT类型。只能选择一列作为因变量。	无
featureColNames	是	自变量。如果输入数据为稠密格式，则支持DOUBLE及BIGINT类型。如果输入数据为稀疏格式，则支持STRING类型。	无
inputTablePartitions	否	输入表的分区。	无
enableSparse	否	输入数据是否为稀疏格式，取值范围为{true,false}。	false
itemDelimiter	否	KV对之间的分隔符。如果enableSparse为true，则该参数生效。	英文逗号（,）

参数	是否必选	描述	默认值
kvDelimiter	否	keyvalue之间的分隔符。如果enableSparse为true，则该参数生效。	英文冒号（:）
maxIter	否	算法进行的最大迭代次数。	100
epsilon	否	最小似然误差。如果两次迭代间的Log Likelihood之差小于该值，则算法终止。	0.000001
regularizedType	否	正则化类型，取值范围为{l1,l2,None}。	None
regularizedLevel	否	正则系数。如果regularizedType为**None**，则该参数失效。	1
enableFitGoodness	否	是否生成模型评估表。指标包括R-Squared、AdjustedR-Squared、AIC、自由度、残差的标准差及偏差。取值范围为{true,false}。	false
enableCoefficientEstimate	否	是否进行回归系数评估。评估指标包括T值、P值及置信区间[2.5%,97.5%]。如果enableFitGoodness为true，则该参数生效。取值范围为{true,false}。	false
lifecycle	否	模型评估输出表的生命周期。	-1
coreNum	否	计算的核心数量。	系统自动分配
memSizePerCore	否	每个核心的内存，取值范围为1024 MB~20*1024 MB。	系统自动分配

线性回归示例

1. 使用SQL语句，生成测试数据。

```
drop table if exists lm_test_input;
create table lm_test_input as
select
 *
from
(
 select 10 as y, 1.84 as x1, 1 as x2, '0:1.84 1:1' as sparsecol1 from dual
  union all
 select 20 as y, 2.13 as x1, 0 as x2, '0:2.13' as sparsecol1 from dual
  union all
 select 30 as y, 3.89 as x1, 0 as x2, '0:3.89' as sparsecol1 from dual
  union all
 select 40 as y, 4.19 as x1, 0 as x2, '0:4.19' as sparsecol1 from dual
  union all
 select 50 as y, 5.76 as x1, 0 as x2, '0:5.76' as sparsecol1 from dual
  union all
 select 60 as y, 6.68 as x1, 2 as x2, '0:6.68 1:2' as sparsecol1 from dual
  union all
 select 70 as y, 7.58 as x1, 0 as x2, '0:7.58' as sparsecol1 from dual
  union all
 select 80 as y, 8.01 as x1, 0 as x2, '0:8.01' as sparsecol1 from dual
  union all
 select 90 as y, 9.02 as x1, 3 as x2, '0:9.02 1:3' as sparsecol1 from dual
  union all
 select 100 as y, 10.56 as x1, 0 as x2, '0:10.56' as sparsecol1 from dual
) tmp;
```

2. 使用PAI命令，提交线性回归组件参数。

```
PAI -name linearregression
  -project algo_public
  -DinputTableName=lm_test_input
  -DlabelColName=y
  -DfeatureColNames=x1,x2
  -DmodelName=lm_test_input_model_out
  -DoutputTableName=lm_test_input_conf_out
  -DenableCoefficientEstimate=true
  -DenableFitGoodness=true
  -Dlifecycle=1;
```

3. 使用PAI命令，提交预测组件参数。

```
pai -name prediction
  -project algo_public
  -DmodelName=lm_test_input_model_out
  -DinputTableName=lm_test_input
  -DoutputTableName=lm_test_input_predict_out
  -DappendColNames=y;
```

4. 查看输出的模型评估表lm_test_input_conf_out。

```
+------------+------------+------------+------------+--------------------+------------+
| colname    | value      | tscore     | pvalue     | confidenceinterval | p          |
+------------+------------+------------+------------+--------------------+------------+
| Intercept  | -6.42378496687763 | -2.2725755951390028 | 0.06       | {"2.5%": -11.964027, "97.5%": -0.883543
} | coefficient |
| x1         | 10.260063429838898 | 23.270944360826963 | 0.0        | {"2.5%": 9.395908, "97.5%": 11.124219} | co
efficient |
| x2         | 0.35374498323846265 | 0.2949247320997519 | 0.81       | {"2.5%": -1.997160, "97.5%": 2.704650} | c
oefficient |
| rsquared   | 0.9879675667384592 | NULL       | NULL       | NULL           | goodness   |
| adjusted_rsquared | 0.9845297286637332 | NULL       | NULL       | NULL           | goodness   |
| aic        | 59.331109494251805 | NULL       | NULL       | NULL           | goodness   |
| degree_of_freedom | 7.0        | NULL       | NULL       | NULL           | goodness   |
| standardErr_residual | 3.765777749448906 | NULL       | NULL       | NULL           | goodness   |
| deviance   | 99.26757440771128 | NULL       | NULL       | NULL           | goodness   |
+------------+------------+------------+------------+--------------------+------------+
```

5. 查看预测结果表lm_test_input_predict_out。

```
+------------+-------------------+------------------+-------------------+
| y          | prediction_result | prediction_score | prediction_detail |
+------------+-------------------+------------------+-------------------+
| 10         | NULL              | 12.808476727264404 | {"y": 12.8084767272644} |
| 20         | NULL              | 15.43015013867922 | {"y": 15.43015013867922} |
| 30         | NULL              | 33.48786177519568 | {"y": 33.48786177519568} |
| 40         | NULL              | 36.565880804147355 | {"y": 36.56588080414735} |
| 50         | NULL              | 52.674180388994415 | {"y": 52.67418038899442} |
| 60         | NULL              | 62.82092871092313 | {"y": 62.82092871092313} |
| 70         | NULL              | 71.34749583130122 | {"y": 71.34749583130122} |
| 80         | NULL              | 75.75932310613193 | {"y": 75.75932310613193} |
| 90         | NULL              | 87.1832221199846 | {"y": 87.18322211998461} |
| 100        | NULL              | 101.92248485222113 | {"y": 101.9224848522211} |
+------------+-------------------+------------------+-------------------+
```

PS-SMART回归

参数服务器PS（Parameter Server）致力于解决大规模的离线及在线训练任务，SMART（Scalable Multiple Additive Regression Tree）是GBDT（Gradient Boosting Decesion Tree）基于PS实现的迭代算法。PS-SMART支持百亿样本及几十万特征的训练任务，可以在上千节点中运行。同时，PS-SMART支持多种数据格式及直方图近似等优化技术。

PAI-Studio支持通过可视化或PAI命令方式，配置PS-SMART回归组件的参数：

- 可视化方式

页签	参数	描述
字段设置	**是否稀疏格式**	稀疏格式的KV之间使用空格分隔，key与value之间使用英文冒号（:）分隔。例如1:0.3 3:0.9。
	选择特征列	输入表中，用于训练的特征列。如果输入数据是Dense格式，则只能选择数值（BIGINT或DOUBLE）类型。如果输入数据是Sparse KV格式，且key和value是数值类型，则只能选择STRING类型。

页签	参数	描述
	选择标签列	输入表的标签列，支持STRING及数值类型。如果是内部存储，则仅支持数值类型。例如二分类中的0和1。
	选择权重列	列可以对每行样本进行加权，支持数值类型。
参数设置	**目标函数类型**	支持Linear regression、Logistic regression、Poisson regression、Gamma regression及Tweedie regression类型。
	Tweedie分布指数	Tweedie分布的方差和均值关系指数。
	评估指标类型	支持rooted mean square error、mean absolute error、negative loglikelihood for logistic regression、negative loglikelihood for poisson regression、residual deviance for gamma regression、negative log-likelihood for gamma regression、negative log-likelihood for Tweedie regression及无类型。
	树数量	训练数据量与**树数量**成正比。
	树最大深度	默认值为5，即最多32个叶子节点。
	数据采样比例	构建每棵树时，采样部分数据进行学习，构建弱学习器，从而加快训练。
	特征采样比例	构建每棵树时，采样部分特征进行学习，构建弱学习器，从而加快训练。
	L1惩罚项系数	控制叶子节点大小。该参数值越大，叶子节点规模分布越均匀。如果过拟合，则增大该参数值。
	L2惩罚项系数	控制叶子节点大小。该参数值越大，叶子节点规模分布越均匀。如果过拟合，则增大该参数值。
	学习速率	取值范围为(0,1)。
	近似Sketch精度	构造Sketch的切割分位点阈值。该参数值越小，获得的桶越多。一般使用默认值0.03，无需手动配置。
	最小分裂损失变化	分裂节点所需要的最小损失变化。该参数值越大，分裂越保守。
	特征数量	特征数量或最大特征ID。如果估计使用资源时，未配置该参数，则系统会启动SQL任务自动计算。
	全局偏置项	所有样本的初始预测值。
	特征重要性类型	支持**模型中，该特征做为分裂特征的次数**、**模型中，该特征带来的信息增益**及**模型中，该特征在分裂节点覆盖的样本数**类型。

页签	参数	描述
执行调优	**核心数**	默认为系统自动分配。
	每个核的内存大小	单个核心使用的内存，单位为MB。通常无需手动配置，系统会自动分配。

- PAI命令方式

```
#训练。
PAI -name ps_smart
  -project algo_public
  -DinputTableName="smart_regression_input"
  -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
  -DoutputTableName="pai_temp_24515_545859_2"
  -DoutputImportanceTableName="pai_temp_24515_545859_3"
  -DlabelColName="label"
  -DfeatureColNames="features"
  -DenableSparse="true"
  -Dobjective="reg:linear"
  -Dmetric="rmse"
  -DfeatureImportanceType="gain"
  -DtreeCount="5";
  -DmaxDepth="5"
  -Dshrinkage="0.3"
  -Dl2="1.0"
  -Dl1="0"
  -Dlifecycle="3"
  -DsketchEps="0.03"
  -DsampleRatio="1.0"
  -DfeatureRatio="1.0"
  -DbaseScore="0.5"
  -DminSplitLoss="0"
#预测。
PAI -name prediction
  -project algo_public
  -DinputTableName="smart_regression_input";
  -DmodelName="xlab_m_pai_ps_smart_bi_545859_v0"
  -DoutputTableName="pai_temp_24515_545860_1"
  -DfeatureColNames="features"
  -DappendColNames="label,features"
  -DenableSparse="true"
  -Dlifecycle="28"
```

模块	参数	是否必选	描述	默认值
	featureColNames	是	输入表中，用于训练的特征列。如果输入表是Dense格式，则只能选择数值（BIGINT或DOUBLE）类型。如果输入表是Sparse KV格式，且KV格式中key和value是数值类型，则只能选择STRING类型。	无

模块	参数	是否必选	描述	默认值
数据参数	labelColName	是	输入表的标签列，支持STRING及数值类型。如果是内部存储，则仅支持数值类型。例如二分类中的0和1。	无
	weightCol	否	列可以对每行样本进行加权，支持数值类型。	无
	enableSparse	否	是否为稀疏格式，取值范围为{true,false}。稀疏格式的KV之间使用空格分隔，key与value之间使用英文冒号（:）分隔。例如1:0.3 3:0.9。	false
	inputTableName	是	输入表的名称。	无
	modelName	是	输出的模型名称。	无
	outputImportanceTableName	否	输出特征重要性的表名。	无
	inputTablePartitions	否	格式为ds=1/pt=1。	无
	outputTableName	否	输出至MaxCompute的表，二进制格式，不支持读取，只能通过SMART的预测组件获取。	无
	lifecycle	否	输出表的生命周期。	3
	objective	是	目标函数类型。支持以下类型： ○ reg:linear：Linear Regression ○ reg:logistic：Logistic Regression ○ count:poisson：Poisson Regression ○ reg:gamma：Gamma Regression ○ reg:tweedie：Tweedie Regression	reg:linear

模块	参数	是否必选	描述	默认值
算法参数	metric	否	训练集的评估指标类型，输出在logview文件coordinator区域的stdout。支持以下类型： ○ rmse：对应可视化方式的**rooted mean square error**类型。 ○ mae：对应可视化方式中的**mean absolute error**类型。 ○ logistic-nloglik：对应可视化方式中的**negative loglikelihood for logistic regression**类型。 ○ poisson-nloglik：对应可视化方式中的**negative loglikelihood for poisson regression**类型。 ○ gamma-deviance：对应可视化方式中的**residual deviance for gamma regression**类型。 ○ gamma-nloglik：对应可视化方式中的**negative log-likelihood for gamma regression**类型。 ○ tweedie-nloglik：对应可视化方式中的**negative log-likelihood for Tweedie regression**类型。	无
	treeCount	否	树数量，与训练时间成正比。	1
	maxDepth	否	树的最大深度，取值范围为1~20。	5
	sampleRatio	否	数据采样比例，取值范围为(0,1]。如果取值为1.0，则表示不采样。	1.0
	featureRatio	否	特征采样比例，取值范围为(0,1]。如果取值为1.0，则表示不采样。	1.0
	l1	否	L1惩罚项系数。该参数值越大，叶子节点分布越均匀。如果过拟合，则增大该参数值。	0
	l2	否	L2惩罚项系数。该参数值越大，叶子节点分布越均匀。如果过拟合，则增大该参数值。	1.0
	shrinkage	否	学习速率，取值范围为(0,1)。	0.3
	sketchEps	否	构造Sketch的切割分位点阈值，桶数为O(1.0/sketchEps)。该参数值越小，获得的桶越多。一般使用默认值，无需手动配置。取值范围为(0,1)。	0.03
	minSplitLoss	否	分裂节点所需要的最小损失变化。该参数值越大，分裂越保守。	0

模块	参数	是否必选	描述	默认值
	featureNum	否	特征数量或最大特征ID。如果估计使用资源时，未配置该参数，则系统会启动SQL任务自动计算。	无
	baseScore	否	所有样本的初始预测值。	0.5
	featureImportanceType	否	计算特征重要性的类型，包括： ○ weight：在模型中，该特征作为分裂特征的次数。 ○ gain：在模型中，该特征带来的信息增益。 ○ cover：在模型中，该特征在分裂节点覆盖的样本数。	gain
	tweedieVarPower	否	Tweedie分布的方差和均值关系指数。	1.5
调优参数	coreNum	否	核心数量，该参数值越大，算法运行越快。	系统自动分配
	memSizePerCore	否	每个核心使用的内存，单位为MB。	系统自动分配

PS-SMART回归示例

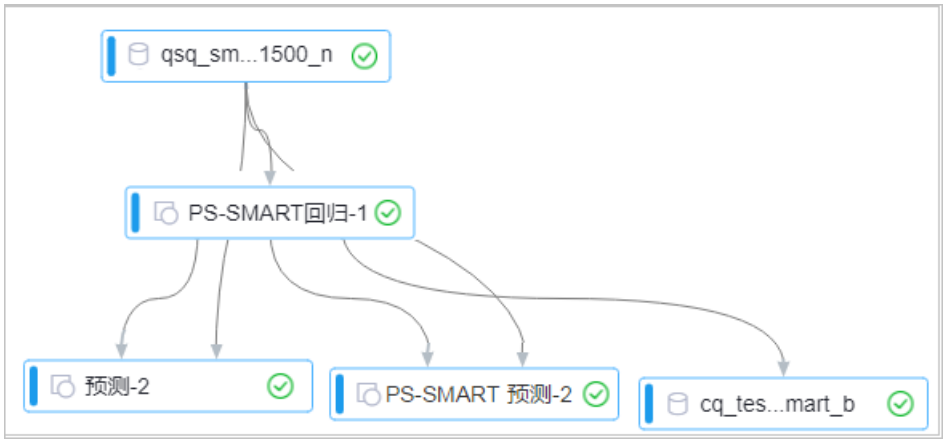
1. 使用如下SQL语句，生成输入数据（以KV格式数据为例）。

```
drop table if exists smart_regression_input;
create table smart_regression_input as
select
*
from
(
select 2.0 as label, '1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17' as features from dual
  union all
select 1.0 as label, '1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41' as features from dual
  union all
select 1.0 as label, '1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91' as features from dual
  union all
select 2.0 as label, '1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60' as features from dual
  union all
select 1.0 as label, '1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86' as features from dual
  union all
select 1.0 as label, '1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84' as features from dual
  union all
select 0.0 as label, '1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30' as features from dual
  union all
select 1.0 as label, '1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41' as features from dual
  union all
select 0.0 as label, '1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44' as features from dual
  union all
select 1.0 as label, '1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35' as features from dual
) tmp;
```

生成的数据如下。

序号	label	features
1	2.0	1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17
2	1.0	1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41
3	1.0	1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91
4	2.0	1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60
5	1.0	1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86
6	1.0	1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84
7	0.0	1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30
8	1.0	1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41
9	0.0	1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44
10	1.0	1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35

2. 构建实验，详情请参见算法建模。



3. 配置PS-SMART回归组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	是否稀疏格式	选中是否稀疏格式复选框。
	特征列	选择features列。
	标签列	选择label列。
参数设置	目标函数类别	配置为Linear regression。
	评估指标类型	选择rooted mean square error。
	树数量	配置为5。

4. 配置统一预测组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	特征列	默认全选，多余列不影响预测结果。
	原样输出列	选择为label列。
	稀疏矩阵	选中稀疏矩阵复选框。
	key与value分隔符	配置为英文冒号（:）。
	kv对间的分隔符	配置为空格或\u0020。

5. 配置PS-SMART预测组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	特征列	默认全选，多余列不影响预测结果。
	原样输出列	选择label列。
	稀疏矩阵	选中稀疏矩阵复选框。

页签	参数	描述
	key与value分隔符	配置为英文冒号（:）。
	kv对间的分隔符	配置为空格或\u0020。

6. 运行实验，查看统一预测组件的预测结果。

序号	label	features	prediction_result	prediction_score	prediction_detail
1	2.0	1:0.55 2:-0....	1.467519998550415	1.467519998550415	{"label": 1.467519998550415}
2	1.0	1:-1.26 2:1....	0.8999134302139282	0.8999134302139282	{"label": 0.8999134302139282}
3	1.0	1:-0.77 2:0....	0.8999134302139282	0.8999134302139282	{"label": 0.8999134302139282}
4	2.0	1:0.86 2:-0....	1.467519998550415	1.467519998550415	{"label": 1.467519998550415}
5	1.0	1:-0.76 2:0....	0.8999134302139282	0.8999134302139282	{"label": 0.8999134302139282}
6	1.0	1:2.22 2:-0....	0.8771200180053711	0.8771200180053711	{"label": 0.8771200180053711}
7	0.0	1:-1.21 2:0....	0.16383999586105347	0.16383999586105347	{"label": 0.1638399958610535}
8	1.0	1:2.17 2:-0....	0.8771200180053711	0.8771200180053711	{"label": 0.8771200180053711}
9	0.0	1:-0.40 2:0....	0.16383999586105347	0.16383999586105347	{"label": 0.1638399958610535}
10	1.0	1:0.17 2:0....	0.8999134302139282	0.8999134302139282	{"label": 0.8999134302139282}

7. 查看PS-Smart预测组件的预测结果。

序号	original_label	prediction_score	leaf_index
1	2	1.467519998550415	1 5 5 5 5
2	1	0.8999134302139282	1 3 3 3 3
3	1	0.8999134302139282	1 3 3 3 3
4	2	1.467519998550415	1 5 5 5 5
5	1	0.8999134302139282	1 3 3 3 3
6	1	0.8771200180053711	1 6 6 6 6
7	0	0.16383999586105347	2 2 2 2 2
8	1	0.8771200180053711	1 6 6 6 6
9	0	0.16383999586105347	2 2 2 2 2
10	1	0.8999134302139282	1 3 3 3 3

其中prediction_score列表示预测值，**leaf_index**列表示预测的叶子节点编号。

8. 右键单击**PS-SMART回归**组件，在快捷菜单，选择**查看数据 > 查看输出桩3**，查看特征重要性。

序号	id	value
1	1	0.14059734344482422
2	4	0.8594027161598206

其中**id**列表示传入的特征序号。因为该示例的输入数据是KV格式，所以**id**列表示KV对中的key。该特性重要性表中仅有2个特性，表示树在分裂过程中仅使用了这两个特性，可以认为其他特性的特征重要性为0。**value**列表示特征重要性类型，默认为gain，即该特征对模型带来的信息增益之和。

相关说明：

- PS-SMART回归组件的目标列仅支持数值类型。如果MaxCompute表数据是STRING类型，则需要进行类型转换。

- 如果数据是KV格式，则特征ID必须为正整数，特征值必须为实数。如果特征ID为字符串类型，则需要使用序列化组件进行序列化。如果特征值为类别型字符串，需要进行特征离散化等特征工程处理。
- 虽然PS-SMART回归组件支持数十万特征任务，但是消耗资源大且运行速度慢，可以使用GBDT类算法进行训练。GBDT类算法适合直接使用连续特征进行训练，除需要对类别特征进行One-Hot编码（筛除低频特征）外，不建议对其他连续型数值特征进行离散化。
- PS-SMART算法会引入随机性。例如，data_sample_ratio及fea_sample_ratio表示的数据和特征采样、算法使用的直方图近似优化及局部Sketch归并为全局Sketch的顺序随机性。虽然多个Worker分布式执行时，树结构不同，但是从理论上可以保证模型效果相近。如果您在训练过程中，使用相同数据和参数，多次得到的结果不一致，属于正常现象。
- 如果需要加速训练，可以增大**计算核心数**。因为PS-SMART算法需要所有服务器获得资源后，才能开始训练，所以集群忙碌时，申请较多资源会增加等待时间。

PS线性回归

线性回归（Linear Regression）是分析因变量和多个自变量之间的线性关系模型，参数服务器PS（Parameter Server）致力于解决大规模的离线及在线训练任务。PS线性回归支持千亿样本、十亿特征的大规模线性训练任务。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**选择特征列**	输入数据源中，参与训练的特征列。
	选择标签列	支持DOUBLE及BIGINT类型。
	是否稀疏格式	使用KV格式表示稀疏格式。
	kv间的分隔符	默认使用空格分隔。
	key与value分隔符	默认使用英文冒号（:）分隔。
参数设置	**L1 weight**	L1正则化系数。该参数值越大，表示模型非零元素越少。如果过拟合，则增大该参数值。
	L2 weight	L2正则化系数。该参数值越大，表示模型参数绝对值越小。如果过拟合，则增大该参数值。
	最大迭代次数	算法进行的最大迭代次数。如果**最大迭代次数**为0，则算法迭代次数无限制。
	最小收敛误差	优化算法终止条件。
	最大特征ID	最大的特征ID或特征维度，该参数取值可以大于实际值。如果未配置该参数，则系统启动SQL任务自动计算。
执行调优	**核心数**	默认为系统自动分配。
	每个核的内存大小	默认为系统自动分配。

- PAI命令方式

```
#训练。
PAI -name ps_linearregression
  -project algo_public
  -DinputTableName="lm_test_input"
  -DmodelName="linear_regression_model"
  -DlabelColName="label"
  -DfeatureColNames="features"
  -Dl1Weight=1.0
  -Dl2Weight=0.0
  -DmaxIter=100
  -Depsilon=1e-6
  -DenableSparse=true
#预测。
drop table if exists logistic_regression_predict;
PAI -name prediction
  -DmodelName="linear_regression_model"
  -DoutputTableName="linear_regression_predict"
  -DinputTableName="lm_test_input"
  -DappendColNames="label,features"
  -DfeatureColNames="features"
  -DenableSparse=true
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
modelName	是	输出模型的名称。	无
outputTableName	否	输出的模型评估表名称。如果enableFitGoodness为true，则该参数必选。	无
labelColName	是	输入表的标签列名，支持DOUBLE及BIGINT类型。	无
featureColNames	是	输入表中，用于训练的特征列名。如果输入数据为稠密格式，则支持DOUBLE及BIGINT类型。如果输入数据为稀疏格式，则支持STRING类型。	无
inputTablePartitions	否	输入表的分区。	无
enableSparse	否	输入数据是否为稀疏格式，取值范围为{true,false}。	false
itemDelimiter	否	KV对之间的分隔符。如果enableSparse为true，则该参数生效。	空格
kvDelimiter	否	keyvalue之间的分隔符。如果enableSparse为true，则该参数生效。	英文冒号（:）

参数	是否必选	描述	默认值
enableModelIo	否	是否输出到Offline Model。如果enableModelIo为false，则将模型输出到MaxCompute表。取值范围为{true,false}。	true
maxIter	否	算法进行的最大迭代次数，取值范围为非负整数。	100
epsilon	否	优化算法终止条件，取值范围为[0,1]。	0.000001
l1Weight	否	L1正则化系数。该参数值越大，模型非零元素越少。如果过拟合，则增大该参数值。	1.0
l2Weight	否	L2正则化系数。该参数值越大，模型参数绝对值越小。如果过拟合，则增大该参数值。	0
modelSize	否	最大的特征ID或特征维度，该参数取值可以大于实际值。如果未配置该参数，则系统启动SQL任务自动计算。取值范围为非负整数。	0
coreNum	否	计算的核心数量。	系统自动分配
memSizePerCore	否	每个核心的内存，单位为MB。	系统自动分配

PS线性回归示例

1. 使用如下SQL语句，生成输入数据（以KV格式数据为例）。

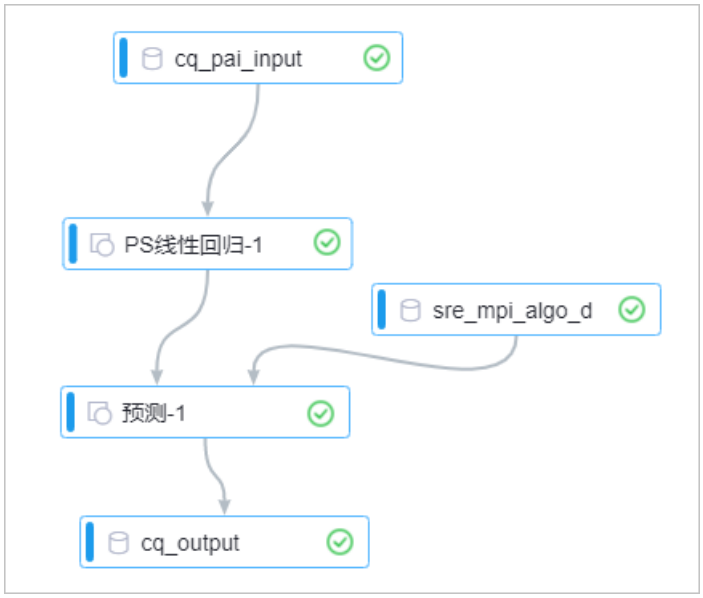
```
drop table if exists lm_test_input;
create table lm_test_input as
select
*
from
(
select 2 as label, '1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17' as features from dual
  union all
select 1 as label, '1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41' as features from dual
  union all
select 1 as label, '1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91' as features from dual
  union all
select 2 as label, '1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60' as features from dual
  union all
select 1 as label, '1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86' as features from dual
  union all
select 1 as label, '1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84' as features from dual
  union all
select 0 as label, '1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30' as features from dual
  union all
select 1 as label, '1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41' as features from dual
  union all
select 0 as label, '1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44' as features from dual
  union all
select 1 as label, '1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35' as features from dual
) tmp;
```

生成的数据如下。

序号	label	features
1	2	1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17
2	1	1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41
3	1	1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91
4	2	1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60
5	1	1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86
6	1	1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84
7	0	1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30
8	1	1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41
9	0	1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44
10	1	1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35

> **说明** KV格式数据的特征ID必须为正整数，特征值必须为实数。如果特征ID为字符串，则需要进行序列化操作。如果特征值为类别型字符串，则需要进行特征离散化操作。

2. 构建实验，详情请参见算法建模。



3. 配置PS线性回归组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**是否稀疏格式**	选择true。
	选择特征列	选择features列。
	选择标签列	选择label列。
执行调优	**核心数**	配置为3。
	每个核的内存大小	配置为1024 MB。

4. 配置预测组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**特征列**	默认全选，多余列不影响预测结果。
	原样输出列	选择label列。
	稀疏矩阵	选中**稀疏矩阵**复选框。
	key与value分隔符	配置为英文冒号（:）。
	kv对间的分隔符	配置为\u0020。

5. 运行实验，查看预测结果。

序号	label	features	prediction_result	prediction_score	prediction_detail
1	2	1:0.55 2:-0.15 3:0.82 4:-0.99 5:0.17	0.9950045235424285	0.9950045235424285	{"1": 0.9950045235424285}
2	1	1:-1.26 2:1.36 3:-0.13 4:-2.82 5:-0.41	0.9022041049173464	0.9022041049173464	{"1": 0.9022041049173464}
3	1	1:-0.77 2:0.91 3:-0.23 4:-4.46 5:0.91	1.228147671448144	1.228147671448144	{"1": 1.228147671448144}
4	2	1:0.86 2:-0.22 3:-0.46 4:0.08 5:-0.60	0.9043180917054381	0.9043180917054381	{"1": 0.9043180917054381}
5	1	1:-0.76 2:0.89 3:1.02 4:-0.78 5:-0.86	0.7116744886195954	0.7116744886195954	{"1": 0.7116744886195954}
6	1	1:2.22 2:-0.46 3:0.49 4:0.31 5:-1.84	1.135349294653747	1.135349294653747	{"1": 1.135349294653747}
7	0	1:-1.21 2:0.09 3:0.23 4:2.04 5:0.30	0.22724956563861654	0.22724956563861654	{"1": 0.2272495656386165}
8	1	1:2.17 2:-0.45 3:-1.22 4:-0.48 5:-1.41	1.2369534428175184	1.2369534428175184	{"1": 1.236953442817518}
9	0	1:-0.40 2:0.63 3:0.56 4:0.74 5:-1.44	0.5672805014883875	0.5672805014883875	{"1": 0.5672805014883875}
10	1	1:0.17 2:0.49 3:-1.50 4:-2.20 5:-0.35	1.0918540901263776	1.0918540901263776	{"1": 1.091854090126378}

3.5.5. 评估

本文为您介绍PAI-Studio提供的评估算法，包括二分类评估、回归模型评估、聚类模型评估、混淆矩阵及多分类评估。

二分类评估

二分类评估通过计算AUC、KS及F1 Score指标，输出KS曲线、PR曲线、ROC曲线、LIFT Chart及Gain Chart。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

参数	描述
原始标签列列名	目标列的名称。
分数列列名	预测分数列，通常为prediction_score列。
正样本的标签值	正样本的分类。
计算KS、PR等指标时按等频分成多少个桶	将数据按照等频划分为桶的数量。
分组列列名	分组ID列。对各分组的数据分别计算评估指标，适用于分组评估场景。
高级选项	如果选中**高级选项**复选框，则**预测结果详细列**、**预测目标与评估目标是否一致**及**保存性能指标**参数生效。
预测结果详细列	预测结果详细列的名称。
预测目标与评估目标是否一致	例如，在金融场景中，训练程序预测坏人的概率，其值越大，表示样本越坏，相关指标（例如LIFT）评估的是抓坏率，此时预测目标与评估目标一致。在信用评分场景中，训练程序预测好人的概率，其值越大，表示样本越好，而相关指标评估的是抓坏率，此时预测目标与评估目标不一致。
保存性能指标	保存性能指标的开关。

- PAI命令方式

```
PAI -name=evaluate -project=algo_public
  -DoutputMetricTableName=output_metric_table
  -DoutputDetailTableName=output_detail_table
  -DinputTableName=input_data_table
  -DlabelColName=label
  -DscoreColName=score
```

参数	是否必选	参数描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表的分区。	全表
labelColName	是	目标列的名称。	无
scoreColName	是	分数列的名称。	无
groupColName	否	分组列的名称，用于分组评估场景。	无
binCount	否	计算KS及PR等指标时，按照等频将数据分成的桶数量。	1000
outputMetricTableName	是	输出的指标表，包括AUC、KS及F1 Score指标。	无
outputDetailTableName	否	用于画图的详细数据表。	无
positiveLabel	否	正样本的分类。	1
lifecycle	否	输出表的生命周期。	无
coreNum	否	核心数量。	系统自动计算
memSizePerCore	否	每个核心的内存。	系统自动计算

回归模型评估

回归模型评估是指基于预测结果和原始结果，评估回归算法模型的优劣性，从而输出评估指标及残差直方图。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**原回归值**	支持数值类型。
	预测回归值	支持数值类型。
执行调优	**节点个数**	与参数**单个节点内存大小**搭配使用，取值范围为1~9999。

页签	参数	描述
	单个节点内存大小	取值范围为1024 MB~64*1024 MB。

- PAI命令方式

```
PAI -name regression_evaluation -project algo_public
  -DinputTableName=input_table
  -DyColName=y_col
  -DpredictionColName=prediction_col
  -DindexOutputTableName=index_output_table
  -DresidualOutputTableName=residual_output_table;
```

参数	是否必选	参数描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表中，参与计算的分区。	全表
yColName	是	输入表中，原始因变量的列名，支持数值类型。	无
predictionColName	是	预测结果中，因变量的列名，支持数值类型。	无
indexOutputTableName	是	回归指标输出表的名称。	无
residualOutputTableName	是	残差直方图输出表的名称。	无
intervalNum	否	直方图区间数量。	100
lifecycle	否	输出表的生命周期，取值范围为正整数。	无
coreNum	否	Instance数量，取值范围为1~9999。	系统自动设置
memSizePerCore	否	每个核心的内存，取值范围为1024 MB~64*1024 MB。	系统自动设置

回归模型评估的输出结果

回归指标输出表的输出结果为JSON格式，包括以下参数。

参数	描述
SST	总平方和
SSE	误差平方和
SSR	回归平方和

参数	描述
R2	判定系数
R	多重相关系数
MSE	均方误差
RMSE	均方根误差
MAE	平均绝对误差
MAD	平均误差
MAPE	平均绝对百分误差
count	行数
yMean	原始因变量的均值
predictionMean	预测结果的均值

聚类模型评估

基于原始数据和聚类结果，评估聚类模型的优劣性，从而输出评估指标。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**参与评估列**	参与评估的列名，该参数必须与模型存储的特征列保持一致。
	输入为稀疏格式	使用KV格式表示稀疏数据。
	kv键间分隔符	默认为英文逗号（,）。
	kv键内分隔符	默认为英文冒号（:）。
执行调优	**核心数**	与参数**每个核的内存大小**搭配使用，取值范围为正整数。
	每个核的内存大小	与参数**核心数**搭配使用，单位为MB。

- PAI命令方式

```
PAI -name cluster_evaluation
  -project algo_public
  -DinputTableName=pai_cluster_evaluation_test_input
  -DselectedColNames=f0,f3
  -DmodelName=pai_kmeans_test_model
  -DoutputTableName=pai_ft_cluster_evaluation_out;
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
selectedColNames	否	输入表中，参与评估的列名，多个列以英文逗号（,）分隔。该参数必须与模型存储的特征列保持一致。	所有列
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	全表
enableSparse	否	输入数据是否为稀疏格式，取值范围为{true,false}。	false
itemDelimiter	否	稀疏格式KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	稀疏格式key和value之间的分隔符。	英文冒号（:）
modelName	是	输入的聚类模型。	无
outputTableName	是	输出表。	无
lifecycle	否	输出表的生命周期。	无

评估指标Calinski-Harabasz又称VRC（Variance Ratio Criterion），其计算公式如下。

$$VRC_k = \frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)}$$

参数	描述
SS_B	聚类之间的方差，定义如下。 $$SS_B = \sum_{i=1}^{k} n_i \|\|m_i - m\|\|^2$$ 其中： • k：聚类中心点的数量。 • m_i：聚类的中心点。 • m：输入数据的均值。

参数	描述
SS_W	聚类内的方差，定义如下。 $SS_W = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - m_i\|^2$ 其中： • k：聚类中心点的数量。 • x：数据点。 • c_i：第i个聚类。 • m_i：聚类的中心点。
N	记录的总数量。
k	聚类中心点的数量。

聚类模型评估示例

1. 使用SQL语句，生成测试数据。

```
create table if not exists pai_cluster_evaluation_test_input as
select * from
(
 select 1 as id, 1 as f0,2 as f3 from dual
 union all
 select 2 as id, 1 as f0,3 as f3 from dual
 union all
 select 3 as id, 1 as f0,4 as f3 from dual
 union all
 select 4 as id, 0 as f0,3 as f3 from dual
 union all
 select 5 as id, 0 as f0,4 as f3 from dual
)tmp;
```

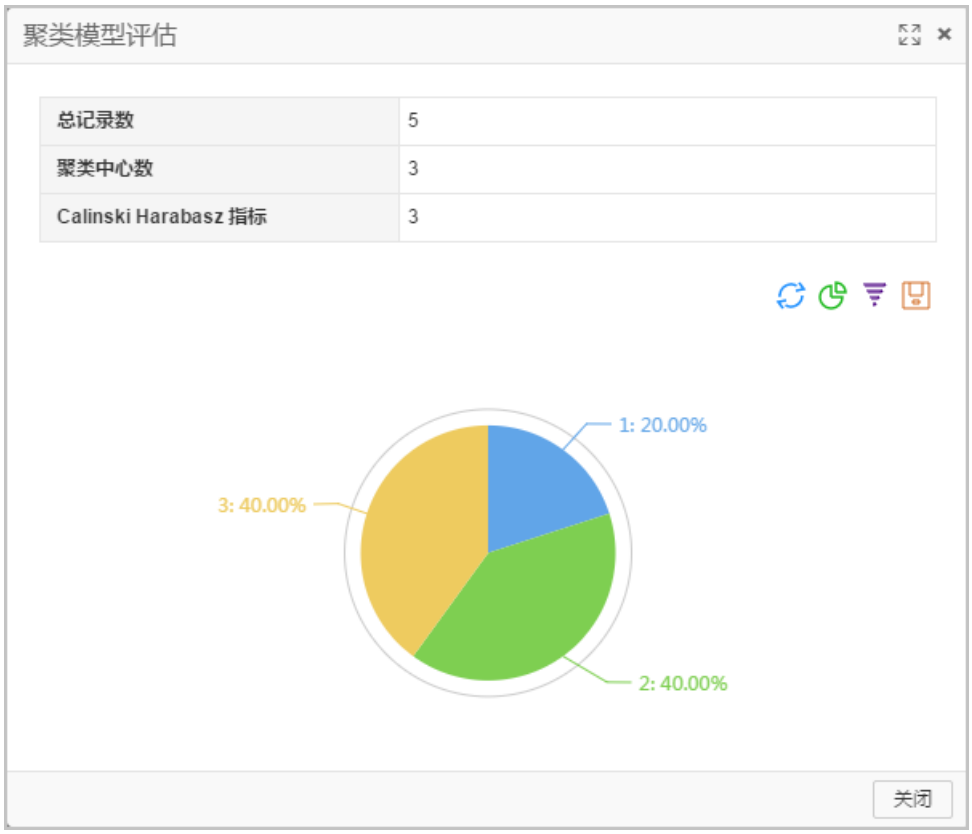
2. 使用PAI命令，构建聚类模型（以K均值聚类为例）。

```
PAI -name kmeans
  -project algo_public
  -DinputTableName=pai_cluster_evaluation_test_input
  -DselectedColNames=f0,f3
  -DcenterCount=3
  -Dloop=10
  -Daccuracy=0.00001
  -DdistanceType=euclidean
  -DinitCenterMethod=random
  -Dseed=1
  -DmodelName=pai_kmeans_test_model
  -DidxTableName=pai_kmeans_test_idx
```

3. 使用PAI命令，提交聚类模型评估组件的参数。

```
PAI -name cluster_evaluation
  -project algo_public
  -DinputTableName=pai_cluster_evaluation_test_input
  -DselectedColNames=f0,f3
  -DmodelName=pai_kmeans_test_model
  -DoutputTableName=pai_ft_cluster_evaluation_out;
```

4. 查看评估输出表pai_ft_cluster_evaluation_out，其可视化报告如下图所示。



该图表与pai_ft_cluster_evaluation_out表中字段对应关系如下。

表字段	可视化图表
count	总记录数
centerCount	聚类中心数
calinhara	Calinski Harabasz指标
clusterCounts	各聚类包含的点数目

混淆矩阵

混淆矩阵（Confusion Matrix）适用于监督学习，与无监督学习中的匹配矩阵对应。在精度评价中，混淆矩阵主要用于比较分类结果和实际测量值，可以将分类结果的精度显示在一个矩阵中。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

参数	描述
原数据的标签列列名	支持数值类型。
预测结果的标签列列名	如果未配置**阈值**，则该参数必选。
阈值	大于该参数值的样本为正样本。
预测结果的详细列列名	与**预测结果的标签列列名**不能共存。如果已配置**阈值**，则该参数必选。
正样本的标签值	如果已配置**阈值**，则该参数必选。

- PAI命令方式
  - 未指定阈值

```
pai -name confusionmatrix -project algo_public
  -DinputTableName=wpbc_pred
  -DoutputTableName=wpbc_confu
  -DlabelColName=label
  -DpredictionColName=prediction_result;
```

  - 指定阈值

```
pai -name confusionmatrix -project algo_public
  -DinputTableName=wpbc_pred
  -DoutputTableName=wpbc_confu
  -DlabelColName=label
  -DpredictionDetailColName=prediction_detail
  -Dthreshold=0.8
  -DgoodValue=N;
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称，即预测输出表。	无
inputTablePartition	否	输入表的分区。	全表
outputTableName	是	输出表的名称，用于存储混淆矩阵。	无
labelColName	是	原始标签列的名称。	无
predictionColName	否	预测结果列的名称。如果未配置threshold，则该参数必选。	无
predictionDetailColName	否	预测结果详细列的名称。如果已配置threshold，则该参数必选。	无
threshold	否	划分正样本的阈值。	0.5
goodValue	否	二分类时，指定训练系数对应的标签值。如果已配置threshold，则该参数必选。	无

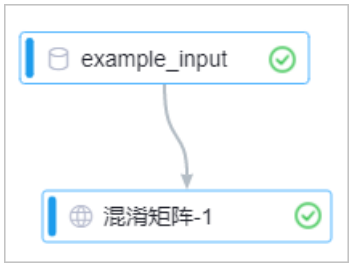
参数	是否必选	描述	默认值
coreNum	否	计算的核心数量。	系统自动分配
memSizePerCore	否	每个核心的内存，单位为MB。	系统自动分配
lifecycle	否	输出表的生命周期。	无

混淆矩阵示例

1. 导入如下测试数据。

id	label	prediction_result
0	A	A
1	A	B
2	A	A
3	A	A
4	B	B
5	B	B
6	B	A
7	B	B
8	B	A
9	A	A

2. 构建如下实验，详情请参见算法建模。

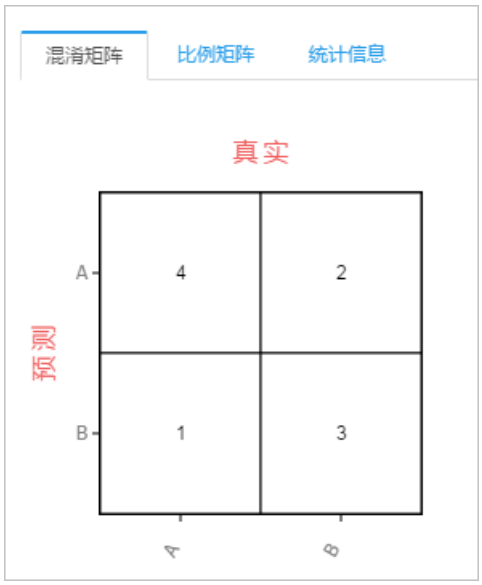


3. 配置混淆矩阵的参数（配置如下表格中的参数，其余参数使用默认值）。

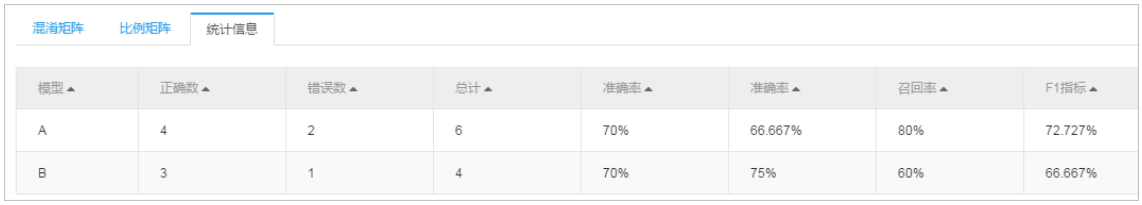
参数	描述
原数据的标签列列名	选择label列。
预测结果的标签列列名	选择prediction_result列。

4. 运行实验，查看混淆矩阵组件的输出结果：

- 单击**混淆矩阵**页签，查看输出的混淆矩阵。



- 单击**比例矩阵**页签，查看比例矩阵。
- 单击**统计信息**页签，查看模型统计信息。



多分类评估

多分类评估是指基于分类模型的预测结果和原始结果，评估多分类算法模型的优劣性，从而输出评估指标（例如Accuracy、Kappa及F1-Score）。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**原分类结果列**	可以选择原始标签列，分类数量不能大于1000。
	预测分类结果列	预测分类列，通常为prediction_result列。
	高级选项	如果选中**高级选项**复选框，则**预测结果概率列**参数生效。
	预测结果概率列	用于计算模型的Logloss，通常为prediction_detail列，且仅对随机森林模型有效。如果对其他模型设置该参数，则系统可能报错。
执行调优	**核心数**	与**核内存分配**搭配使用，默认为系统自动分配。
	核内存分配	每个核心的内存，单位为MB，默认为系统自动分配。

- PAI命令方式

```
PAI -name MultiClassEvaluation -project algo_public \
  –DinputTableName="test_input" \
  -DoutputTableName="test_output" \
  -DlabelColName="label" \
  -DpredictionColName="prediction_result" \
  -Dlifecycle=30;
```

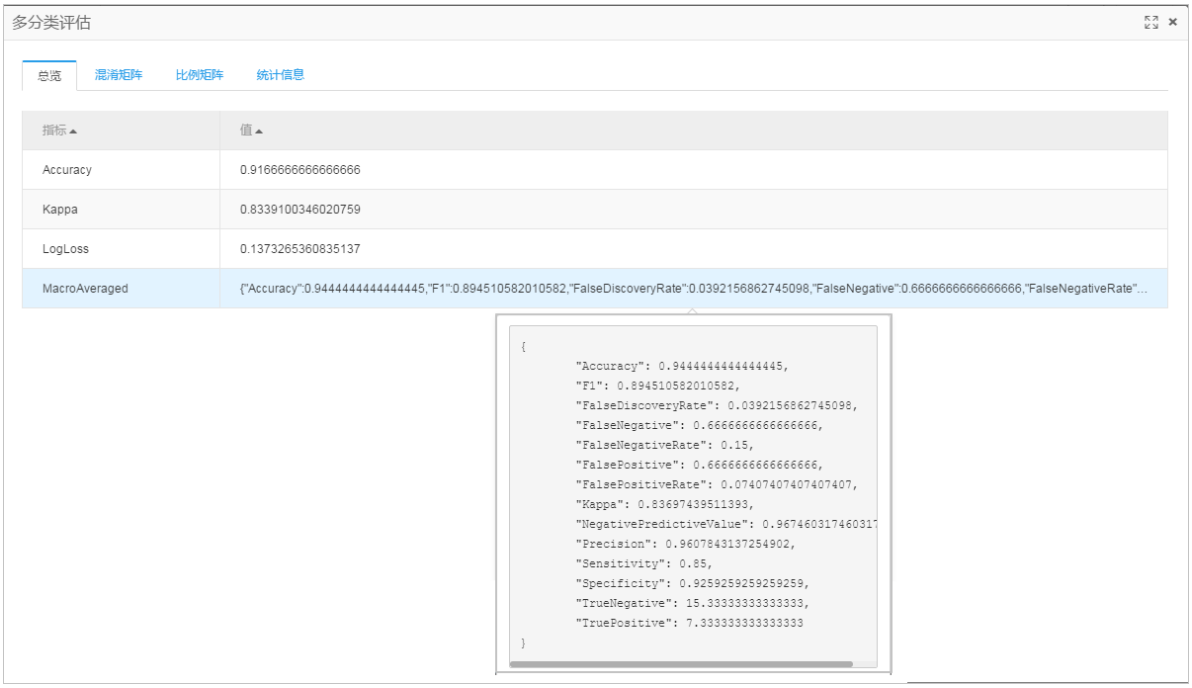
参数	是否必选	参数描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表的分区。	全表
outputTableName	是	输出表的名称。	无
labelColName	是	输入表原始标签列名。	无
predictionColName	是	预测结果的标签列名。	无
predictionDetailColName	否	预测结果的概率列，例如{“A”:0.2,“B”:0.3,“C”:0.5}。	空
lifecycle	否	输出表的生命周期。	无
coreNum	否	核心数量。	系统自动计算
memSizePerCore	否	每个核心的内存。	系统自动计算

多分类评估的输出说明

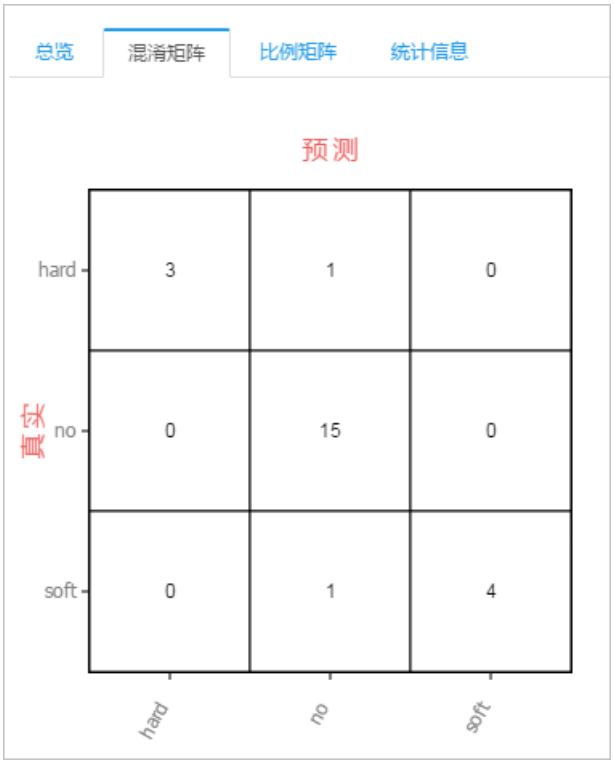
多分类评估组件输出的评估报告包括：

- 总览

汇总指标，其中MacroAveraged为每个标签指标的平均值。



- 混淆矩阵



- 比例矩阵
- 统计信息

按照One-VS-All的方式，计算每个标签的指标。

总览 混淆矩阵 比例矩阵 统计信息

模型	TruePositive	TrueNegative	FalsePositive	FalseNegative	Sensitivity	Specificity	Precision	Accuracy	F1	Kappa
A	4	3	2	1	0.8	0.6	0.6666666...	0.7	0.727...	0.3999...
B	3	4	1	2	0.6	0.8	0.75	0.7	0.666...	0.3999...

多分类评估组件输出结果表的JSON格式如下。

```
{
  "LabelNumber": 3,
  "LabelList": ["A", "B", "C"],
  "ConfusionMatrix": [ // 混淆矩阵[actual][predict]。
    [100, 10, 20],
    [30, 50, 9],
    [7, 40, 90] ],
  "ProportionMatrix": [ // 比例矩阵（按行占比）[actual][predict]。
    [0.6, 0.2, 0.2],
    [0.3, 0.6, 0.1],
    [0.1, 0.4, 0.5] ],
  "ActualLabelFrequencyList": [ // 每个标签的真实数量。
    200, 300, 600],
  "ActualLabelProportionList": [ // 每个标签的真实占比。
    0.1, 0.2, 0.7],
  "PredictedLabelFrequencyList": [ // 预测的每个标签数量。
    300, 400, 400],
  "PredictedLabelProportionList": [ // 预测的每个标签占比。
    0.2, 0.1, 0.7],
  "OverallMeasures": {     // 汇总指标。
    "Accuracy": 0.70,
    "Kappa" : 0.3,
    "MacroList": {     // 每个标签的指标平均值。
      "Sensitivity": 0.4,
      "Specificity": 0.3,
    },
    "MicroList": {     // 根据每个标签的TP、TN、FP及FN之和，计算该指标。
      "Sensitivity": 0.4,
      "Specificity": 0.3,
    },
    "LabelFrequencyBasedMicro": { // 按照频率，计算每个标签指标的加权平均值。
      "Sensitivity": 0.4,
      "Specificity": 0.3,
    },
  },
  "LabelMeasuresList": [     // 每个标签的指标。
    {
      "Accuracy": 0.6,
      "Sensitivity": 0.4,
      "Specificity": 0.3,
      "Kappa": 0.3
    },
    {
      "Accuracy": 0.6,
      "Sensitivity": 0.4,
      "Specificity": 0.3,
      "Kappa": 0.3
    },
  ]
}
```

多分类评估示例

1. 导入如下测试数据。

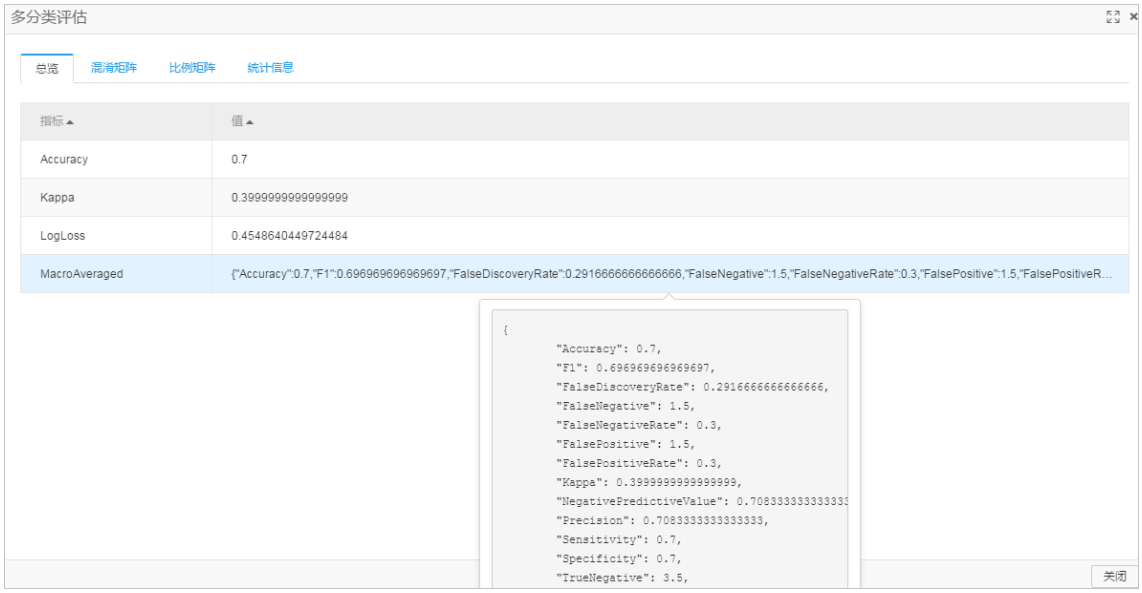
id	label	prediction	detail
0	A	A	{“A”：0.6，“B”：0.4}
1	A	B	{“A”：0.45，“B”：0.55}
2	A	A	{“A”：0.7，“B”：0.3}
3	A	A	{“A”：0.9，“B”：0.1}
4	B	B	{“A”：0.2，“B”：0.8}
5	B	B	{“A”：0.1，“B”：0.9}
6	B	A	{“A”：0.52，“B”：0.48}
7	B	B	{“A”：0.4，“B”：0.6}
8	B	A	{“A”：0.6，“B”：0.4}
9	A	A	{“A”：0.75，“B”：0.25}

2. 构建实验，详情请参见算法建模。

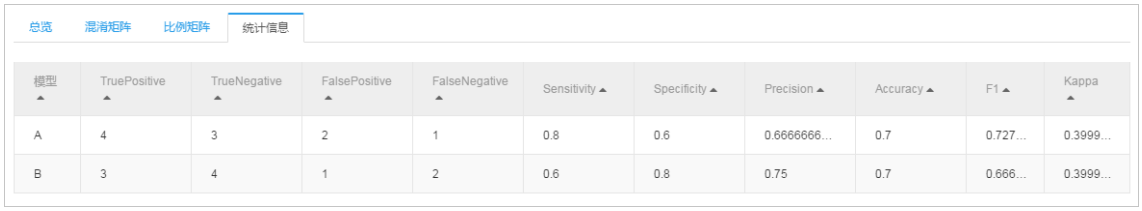
3. 配置多分类组件的参数（配置如下表格中的参数，其余参数使用默认值）。

页签	参数	描述
字段设置	**原分类结果列**	选择label列。
	预测分类结果列	输入prediction。
	高级选项	选中**高级选项**复选框。
	预测结果概率列	输入detail。

4. 运行实验，查看多分类评估组件输出的评估报告：
   - 单击**总览**页签，查看总览信息。



- 单击**混淆矩阵**页签，查看混淆矩阵。
- 单击**比例矩阵**页签，查看比例矩阵。
- 单击**统计信息**页签，查看模型统计信息。



该评估报告对应的JSON文件如下。


```
{
  "ActualLabelFrequencyList": [5,
    5],
  "ActualLabelProportionList": [0.5,
    0.5],
  "ConfusionMatrix": [[4,
      1],
    [2,
      3]],
  "LabelList": ["A",
    "B"],
  "LabelMeasureList": [{
      "Accuracy": 0.7,
      "Auc": 0.9,
      "F1": 0.7272727272727273,
      "FalseDiscoveryRate": 0.3333333333333333,
      "FalseNegative": 1,
      "FalseNegativeRate": 0.2,
      "FalsePositive": 2,
      "FalsePositiveRate": 0.4,
      "Kappa": 0.3999999999999999,
      "NegativePredictiveValue": 0.75,
      "Precision": 0.6666666666666666,
      "Sensitivity": 0.8
```

```
      "Specificity": 0.6,
      "TrueNegative": 3,
      "TruePositive": 4},
    {
      "Accuracy": 0.7,
      "Auc": 0.9,
      "F1": 0.6666666666666666,
      "FalseDiscoveryRate": 0.25,
      "FalseNegative": 2,
      "FalseNegativeRate": 0.4,
      "FalsePositive": 1,
      "FalsePositiveRate": 0.2,
      "Kappa": 0.3999999999999999,
      "NegativePredictiveValue": 0.6666666666666666,
      "Precision": 0.75,
      "Sensitivity": 0.6,
      "Specificity": 0.8,
      "TrueNegative": 4,
      "TruePositive": 3}],
  "LabelNumber": 2,
  "OverallMeasures": {
    "Accuracy": 0.7,
    "Kappa": 0.3999999999999999,
    "LabelFrequencyBasedMicro": {
      "Accuracy": 0.7,
      "F1": 0.696969696969697,
      "FalseDiscoveryRate": 0.2916666666666666,
      "FalseNegative": 1.5,
      "FalseNegativeRate": 0.3,
      "FalsePositive": 1.5,
      "FalsePositiveRate": 0.3,
      "Kappa": 0.3999999999999999,
      "NegativePredictiveValue": 0.7083333333333333,
      "Precision": 0.7083333333333333,
      "Sensitivity": 0.7,
      "Specificity": 0.7,
      "TrueNegative": 3.5,
      "TruePositive": 3.5},
    "LogLoss": 0.4548640449724484,
    "MacroAveraged": {
      "Accuracy": 0.7,
      "F1": 0.696969696969697,
      "FalseDiscoveryRate": 0.2916666666666666,
      "FalseNegative": 1.5,
      "FalseNegativeRate": 0.3,
      "FalsePositive": 1.5,
      "FalsePositiveRate": 0.3,
      "Kappa": 0.3999999999999999,
      "NegativePredictiveValue": 0.7083333333333333,
      "Precision": 0.7083333333333333,
      "Sensitivity": 0.7,
      "Specificity": 0.7,
      "TrueNegative": 3.5,
      "TruePositive": 3.5},
```

```
    "MicroAveraged": {
      "Accuracy": 0.7,
      "F1": 0.7,
      "FalseDiscoveryRate": 0.3,
      "FalseNegative": 3,
      "FalseNegativeRate": 0.3,
      "FalsePositive": 3,
      "FalsePositiveRate": 0.3,
      "Kappa": 0.3999999999999999,
      "NegativePredictiveValue": 0.7,
      "Precision": 0.7,
      "Sensitivity": 0.7,
      "Specificity": 0.7,
      "TrueNegative": 7,
      "TruePositive": 7}},
  "PredictedLabelFrequencyList": [6,
    4],
  "PredictedLabelProportionList": [0.6,
    0.4],
  "ProportionMatrix": [[0.8,
      0.2],
    [0.4,
      0.6]]}
```


3.5.6. 推荐算法

本文为您介绍PAI-Studio提供的推荐算法，包括协同过滤etrec、FM训练、FM预测及ALS矩阵分解。

协同过滤etrec

etrec是基于item的协同过滤算法，PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**用户列名**	user列名。
	物品列名	item列名。
参数设置	**相似度类型**	支持**wbcosine**、**asymcosine**及**jaccard**类型。
	TopN	输出结果中最多保留的相似物品数量。
	计算行为	如果同一user出现相同的item，则payload进行的计算行为。系统支持**add**、**mul**、**min**及**max**行为。 **说明** 该参数即将下线，目前对训练效果无影响。
	最小物品值	如果某user的item数小于该值，则忽略该user的行为。

页签	参数	描述
	最大物品值	如果某user的item数大于该值，则忽略该user的行为。
	平滑因子	仅**相似度类型**为**asymcosine**时，该参数生效。
	权重系数	仅**相似度类型**为**asymcosine**时，该参数生效。

- PAI命令方式

```
PAI -name pai_etrec
  -project algo_public
  -DsimilarityType="wbcosine"
  -Dweight="1"
  -DminUserBehavior="2"
  -Dlifecycle="28"
  -DtopN="2000"
  -Dalpha="0.5"
  -DoutputTableName="etrec_test_result"
  -DmaxUserBehavior="500"
  -DinputTableName="etrec_test_input"
  -Doperator="add"
  -DuserColName="user"
  -DitemColName="item"
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
userColName	是	输入表中的user列名。	无
itemColName	是	输入表中的item列名。	无
inputTablePartitions	否	输入表中，参与训练的分区名称。	全表
outputTableName	是	输出表的名称。	无
outputTablePartition	否	输出表的Partition。	无
similarityType	否	相似度类型。取值范围为{wbcosine,asymcosine,jaccard}。	wbcosine
topN	否	相似度最大的N个item。取值范围为1~10000。	2000
minUserBehavior	否	最小用户行为。	2
maxUserBehavior	否	最大用户行为。	500
itemDelimiter	否	输出表中item之间的分隔符。	空格
kvDelimiter	否	输出表中key与与value之间的分隔符。	英文冒号（:）

参数	是否必选	描述	默认值
alpha	否	Asymcosine的平滑因子。	0.5
weight	否	Asymcosine的权重指数。	1.0
operator	否	如果同一user出现相同的item，则payload进行的计算行为。取值范围为{add,mul,min,max}。	add
lifecycle	否	输出结果表的生命周期。	1
coreNum	否	核心数。	默认自动分配
memSizePerCore	否	单个核心的内存，单位为MB。	默认自动分配

协同过滤etrec示例

1. 使用SQL语句，生成训练数据。

```
drop table if exists etrec_test_input;
create table etrec_test_input
as
select
  *
from
(
  select
    cast(0 as string) as user,
    cast(0 as string) as item
  from dual
  union all
    select
      cast(0 as string) as user,
      cast(1 as string) as item
    from dual
  union all
    select
      cast(1 as string) as user,
      cast(0 as string) as item
    from dual
  union all
    select
      cast(1 as string) as user,
      cast(1 as string) as item
    from dual
) a;
```

生成的训练数据表etrec_test_input如下。

user	item
0	0

user	item
0	1
1	0
1	1

2. 使用PAI命令，提交训练参数。

```
drop table if exists etrec_test_result;
PAI -name pai_etrec
  -project algo_public
  -DsimilarityType="wbcosine"
  -Dweight="1"
  -DminUserBehavior="2"
  -Dlifecycle="28"
  -DtopN="2000"
  -Dalpha="0.5"
  -DoutputTableName="etrec_test_result"
  -DmaxUserBehavior="500"
  -DinputTableName="etrec_test_input"
  -Doperator="add"
  -DuserColName="user"
  -DitemColName="item";
```

3. 查看结果输出表etrec_test_result。

itemid	similarity
0	1:1
1	0:1

FM算法

FM（Factorization Machine）算法兼顾特征之间的相互作用，是一种非线性模型，适用于电商、广告及直播的推荐场景。PAI-Studio提供的FM算法模板包括FM训练和FM预测组件，您可以在PAI-Studio控制台首页的**FM算法实现推荐模型**区域，单击**从模板创建**，快速构建FM实验。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

组件	页签	参数	描述
	字段设置	**特征列**	数据格式为key:value，多个特征以英文逗号（,）分隔。
		标签列	输入列。仅支持DOUBLE类型。
		任务类型	支持regression及binary_classification类型。

组件	页签	参数	描述
FM训练	**参数设置**	**迭代数**	无。
		正则化系数	使用英文逗号（,）分隔的三个浮点数，分别表示0阶项、1阶项及2阶项的正则化系数。
		学习率	如果训练发散，则降低该参数取值。
		参数初始化标准差	无。
		维度	使用英文逗号（,）分隔的三个正整数，分别表示0阶项、1阶项、2阶项长度。
		数据块大小	性能参数。
		输出表生命周期	无。
	执行调优	**节点个数**	与**单个节点内存大小**配合使用。取值范围为1~9999。
		单个节点内存大小	与**节点个数**配合使用。取值范围为1024 MB~64*1024 MB。
FM预测	**参数设置**	**预测结果列名**	无。
		预测得分列名	无。
		详细预测信息列名	无。
		保持列	保存至输出结果表的列。
	执行调优	**节点个数**	与**单个节点内存大小**配合使用。取值范围为1~9999。
		单个节点内存大小	与**节点个数**配合使用。取值范围为1024 MB~64*1024 MB。

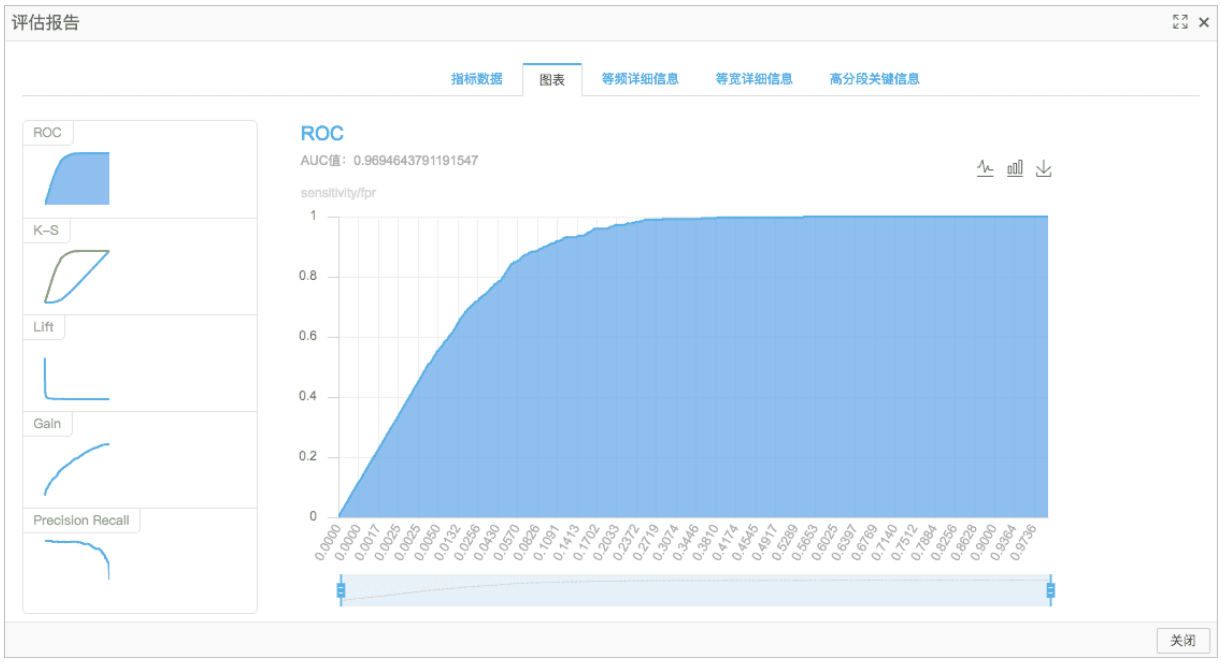
- PAI命令方式

组件	参数	是否必选	描述	默认值
	tensorColName	是	特征列名称。 数据格式为key:value，多个特征使用英文逗号（,）分隔。例如1:1.0,3:1.0。	无
	labelColName	是	label列名。数据必须是数值类型。如果task取值为binary_classification，则label只能取0或1。	无

组件	参数	是否必选	描述	默认值
FM训练	task	是	任务类型。取值范围为{regression,binary_classification}。	regression
	numEpochs	否	迭代数。	10
	dim	否	使用英文逗号（,）分隔的三个整数，分别表示0次项、线性项及二次项的长度。	1,1,10
	learnRate	否	学习率。 **说明** 如果训练发散，则降低学习率。	0.01
	lambda	否	使用英文逗号（,）分隔的三个浮点数，分别表示0次项、线性项及二次项的正则化系数。	0.01,0.01,0.01
	initStdev	否	参数初始化标准差。	0.05
FM预测	predResultColName	否	预测结果列名。	prediction_result
	predScoreColName	否	预测得分列名。	prediction_score
	predDetailColName	否	详细预测信息列名。	prediction_detail
	keepColNames	否	保存至输出结果表的列。	所有列

例如，使用如下数据作为FM算法模板的输入数据，训练生成的模型AUC约为0.97。

序号	label	features
1	0	3:1,11:1,14:1,19:1,39:1,42:1,55:1,64:1,67:1,73:1,75:1,76:1,80:1,83:1
2	0	3:1,6:1,17:1,27:1,35:1,40:1,57:1,63:1,69:1,73:1,74:1,76:1,81:1,103:1
3	0	4:1,6:1,15:1,21:1,35:1,40:1,57:1,63:1,67:1,73:1,74:1,77:1,80:1,83:1
4	0	5:1,6:1,15:1,22:1,36:1,41:1,47:1,66:1,67:1,72:1,74:1,76:1,80:1,83:1
5	0	2:1,6:1,16:1,22:1,36:1,40:1,54:1,63:1,67:1,73:1,75:1,76:1,80:1,83:1
6	0	2:1,6:1,14:1,20:1,37:1,41:1,47:1,64:1,67:1,73:1,74:1,76:1,82:1,83:1
7	0	1:1,6:1,14:1,22:1,36:1,42:1,49:1,64:1,67:1,72:1,74:1,77:1,80:1,83:1
8	0	1:1,6:1,17:1,19:1,39:1,42:1,53:1,64:1,67:1,73:1,74:1,76:1,80:1,83:1
9	0	2:1,6:1,18:1,20:1,37:1,42:1,48:1,64:1,71:1,73:1,74:1,76:1,81:1,83:1
10	1	5:1,11:1,15:1,32:1,39:1,40:1,52:1,63:1,67:1,73:1,74:1,76:1,78:1,83:1



ALS矩阵分解

交替最小二乘ALS（Alternating Least Squares）算法的原理是对稀疏矩阵进行模型分解，评估缺失项的值，从而得到基本的训练模型。在协同过滤分类方面，ALS算法属于User-Item CF（Collaborative Filtering），兼顾User和Item项，也称为混合CF。

PAI-Studio提供ALS算法模板，您可以在PAI-Studio控制台首页的**ALS实现音乐推荐**区域，单击**从模板创建**，快速构建ALS实验。因为系统不支持使用PAI命令配置ALS矩阵分解组件的参数，所以只能通过可视化方式配置该组件参数。

页签	参数	描述
字段设置	**user列名**	输入数据源中，用户ID列的名称。该列数据必须是BIGINT类型。
	item列名	输入数据源中，item项的列名。该列数据必须是BIGINT类型。

页签	参数	描述
	打分列名	输入数据源中，用户对item项的打分所在的列名。该列数据必须是数值型。
参数设置	**因子数**	默认值为10，取值范围为(0,+∞)。
	迭代数	默认值为10，取值范围为(0,+∞)。
	正则化系数	默认值为0.1，取值范围为(0,+∞)。
	是否采用隐式偏好模型	隐式偏好模型开关。
	隐式偏好系数	默认值为40，取值范围为(0,+∞)。
执行调优	**节点个数**	取值范围为1~9999。
	单个节点的内存大小	取值范围为1024 MB~64*1024 MB。

例如，使用如下数据作为ALS算法模板的输入数据，可以获得输出的X矩阵和Y矩阵：

- 输入数据源

id	user	score	item
5	3249	1	978245916
5	3176	2	978243085
5	1719	3	978244205
5	2806	2	978243085
5	2734	2	978242788
5	1649	4	978244667
5	321	3	978245863

- 输出的X矩阵

user	factors
1	[0.5775652527809143,0.6737191677093506,0.9059759974479675,0.9866708517074585,0.15602371096611023,0.2735472023487091,0.4610620439052582,0.53126531839…
2	[0.8297491073608398,0.9742560982704163,0.20942062139511108,0.701496422290802,0.3298608958721161,0.2637876868247986,0.2463243007659912,1.072137832641…
3	[0.41283249855041504,0.05786990746855736,0.09279601275920868,0.17055073380470276,0.28811654448509216,0.520247757434845,0.6457393169403076,0.55301398…
4	[0.4568990468978882,0.706784725189209,0.27782294154167175,0.4249696731567383,0.2610655725002289,0.8090317845344543,0.38613465428352356,0.73447245359…
5	[0.27422088384628296,1.4948835372924805,0.7701016664505005,0.09342949837446213,0.0758112445473671,0.6870493888854981,1.7983382940292358,0.56345194578…
6	[0.5954760313034058,0.3251837193965912,0.6079537868499756,0.24247819185256958,0.48808714747428894,0.29382362961769104,0.9361608624458313,0.706079840…
7	[-0.03147859871387482,0.15576674044132233,0.14959987998008728,0.6218625903129578,0.665594756603241,0.8321151733398438,1.3799917697906494,0.372453927…
10	[0.9043779969215393,1.0728607177734375,0.3242603838443756,0.3242107033729553,0.8776293992996216,0.4835788905620575,0.5912584662437439,0.761821210384…
11	[0.5306817889213562,0.2584091229652405,0.3185936808586120 6,0.4959831237792969,0.12244139611721039,0.9162191152572632,0.2533270120620727 5,0.74489533…
14	[0.5447552800178528,0.7779694199562073,0.523705005645752,0.8368392586708069,0.492429256439209,0.4384959638118744,0.4921191334724426,0.90973246097564…
16	[0.2452736794948578,0.8825525045394897,0.5863954424858093,0.366982638835907,0.5939876437187195,0.3587069809436798,0.6400734186172485,1.3128618001937…
17	[0.9096112847328186,0.7072085738182068,0.3837902545928955,0.21780475974082947,0.5543192028999329,0.6474085450172424,1.1136257648468018,0.26731985807…
21	[1.0358597040176392,0.40937551856040955,-0.1093253344297409,0.12899069488048553,1.085453987121582,1.1062564849853516,0.48638203740119934,0.351608008…
24	[0.5609134435653687,0.8832410573959351,0.8289874792098999,0.1667052060365677,-0.0539841428399086,0.2634010314941406,0.8972880244255066,-0.06595267355…
25	[0.8964614868164062,1.0373890399932861,1.121972680091858,0.4297601282596588,0.9643515348434448,0.3594096601009369,0.8026714324951172,0.5237404108047…
26	[0.8279176950454712,0.06267868727445602,0.6535260081291199,0.7451098561286926,0.6139190793037415,0.9026572704315186,0.6037695407867432,0.82576447725…
29	[0.4705789387226105,0.8837192058563232,0.4688251912593841 6,0.2622719407081604,0.3242076635360718,-0.3560419082641601 6,1.171418547630 31,0.35865238308…

- 输出的Y矩阵

item	factors
978130429	[0.2431642860174179,0.6019538044929504,0.4035401940345764,0.254305899143219,0.4056856632232666,0.46871861815452576,0.3701469600200653,0.37579229...
978130465	[0.12887848913669586,0.5140372514724731,0.7569844126701355,0.43304914236068726,0.5548039078712463,0.1846630722284317,0.8375828862190247,0.922644...
978130485	[0.5187233090400696,0.01644902676343918,0.30217915773391724,0.7121358513832092,0.3559131622314453,0.6373451352119446,0.6356949210166931,0.7720878...
978130612	[0.6717739701271057,0.07011739909648895,-0.14576716721057892,-0.5848092436790466,0.6463701128959656,1.4211229085922241,0.978522777557373,0.0279569...
978130633	[0.76789790391922,0.7038623094558716,0.5854156613349915,0.6497761607170105,0.5391698479652405,0.6755023002624512,0.7179682850837708,0.5982030034...
978130670	[0.2403375208377838,0.5461633801460266,0.9200704097747803,1.064564824104309,0.6212608218193054,1.267152190208435,0.5108421444892883,0.4369839131...
978130703	[0.28001174330711365,0.7113327383995056,0.5033919811248779,0.10984116047620773,0.7204229831695557,0.4389004111289978,0.7448101043701172,0.36075645...
978130711	[0.7834188342094421,0.6742879748344421,0.5627444386482239,0.5531031489372253,0.9128396511077881,0.5433675050735474,0.5437983870506287,0.662769436...
978130715	[0.4402806758880615,0.5298744440078735,0.44332155585289,0.6304010152816772,0.45532888174057007,0.37801969051361084,0.8687032461166382,0.801231682...
978130729	[-0.07635237276554108,0.7233363389968872,0.4513658583164215,0.687857985496521,0.7565320134162903,0.8551117777824402,0.6841593980789185,0.27169799...
978130782	[0.15669465065002441,0.30549323558807373,0.8395602107048035,0.5658870339393616,0.4259960353374481,0.19400911033153534,4.0137249743565917E-4,0.672...
978130855	[0.4147326648235321,0.3227925300598144 5,0.5249242186546326,0.24316798150539398,0.5092757344245911,-0.018865084275603294,0.6107094883918762,0.4126...
978130895	[0.4914812445640564,0.037220340222120285,0.43273890018463135,0.499946266412735,0.5319150686264038,0.39401403069496155,-0.05503401905298233,0.4671...
978130919	[1.032468557357788,1.0190776586532593,0.4246324300765991,0.5157247185707092,0.28222957253456116,0.664626955986023,0.9744172096252441,0.8370118737...
978130991	[0.6485525965690613,0.6814789772033691,0.30417549610137 94,0.2130739688873291,0.9877475500106812,0.8997684717178345,0.6870101094245911,0.316646069...
978131048	[0.6646682620048523,0.18477798998355865,0.9337624311447144,0.7535161375999451,0.6630390882492065,0.1786441206932068,0.76865154504776,0.4035094082...
978131094	[0.45458555221557617,1.3109282803582764,0.0458631291985 5118,0.3885376453399658,0.6060879230499268,0.06554857641458511,0.5884337425231934,1.107482...

如果预测user1对994556636项的评分，则将如下两个向量相乘即可。

```
#向量x。
[-0.14220297,0.8327106,0.5352268,0.6336995,1.2326205,0.7112976,0.9794858,0.8489773,0.330319,0.7426911]
#向量y。
[0.71699333,0.5847747,0.96564907,0.36637592,0.77271074,0.52454436,0.69028413,0.2341857,0.73444265,0.8352135]
```

3.5.7. 预测

本文为您介绍PAI-Studio提供的预测算法。

预测

通常，传统的数据挖掘算法都可以使用预测组件进行模型预测。该组件的输入为训练模型和预测数据，输出为预测结果。

PAI-Studio支持通过可视化或PAI命令的方式，配置该组件参数：

- 可视化方式

页签	参数	描述
字段设置	**特征列**	参与预测的特征列，默认选择所有列。
	原样输出列	建议添加标签列，便于评估。
	输出结果列名	输出表的结果列。
	输出分数列名	输出i表的分数列。
	输出详细列名	输出表的详细列。
	稀疏矩阵	使用KV格式表示稀疏数据。
	key与value分隔符	默认为英文冒号（:）。

页签	参数	描述
	kv对间的分隔符	默认为英文逗号（,）。
执行调优	**核心数**	与参数**每个核的内存大小**搭配使用，取值范围为正整数。
	每个核的内存大小	与参数**核心数**搭配使用，单位为MB。

- PAI命令方式

```
pai -name prediction
  -DmodelName=nb_model
  -DinputTableName=wpbc
  -DoutputTableName=wpbc_pred
  -DappendColNames=label;
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
featureColNames	否	输入表中，参与预测的特征列，多个列以英文逗号（,）分隔。	所有列
appendColNames	否	输入表中，附加至输出表的预测列。	无
inputTablePartitions	否	输入表中，参与训练的分区。支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级格式 **说明** 如果指定多个分区，则使用英文逗号（,）分隔。	全表
outputTablePartition	否	结果输出至输出表的分区。	无
resultColName	否	输出表的结果列。	prediction_result
scoreColName	否	输出表的分数列。	prediction_score
detailColName	否	输出表的详细列。	prediction_detail
enableSparse	否	输入数据是否为稀疏格式，取值范围为{true,false}。	false
itemDelimiter	否	稀疏格式KV对之间的分隔符。	英文逗号（,）
kvDelimiter	否	稀疏格式key和value之间的分隔符。	英文冒号（:）
modelName	是	输入的聚类模型。	无

参数	是否必选	描述	默认值
outputTableName	是	输出表。	无
lifecycle	否	输出表的生命周期。	无
coreNum	否	节点个数。	系统自动分配
memSizePerCore	否	每个节点的内存，单位为MB。	系统自动分配

预测示例

1. 使用SQL语句，生成测试数据。

```
create table pai_rf_test_input as
select * from
(
select 1 as f0,2 as f1, "good" as class from dual
union all
select 1 as f0,3 as f1, "good" as class from dual
union all
select 1 as f0,4 as f1, "bad" as class from dual
union all
select 0 as f0,3 as f1, "good" as class from dual
union all
select 0 as f0,4 as f1, "bad" as class from dual
)tmp;
```

2. 使用PAI命令，构建模型（以随机森林算法为例）。

```
PAI -name randomforests
  -project algo_public
  -DinputTableName="pai_rf_test_input"
  -DmodelName="pai_rf_test_model"
  -DforceCategorical="f1"
  -DlabelColName="class"
  -DfeatureColNames="f0,f1"
  -DmaxRecordSize="100000"
  -DminNumPer="0"
  -DminNumObj="2"
  -DtreeNum="3";
```

3. 使用PAI命令，提交预测组件参数。

```
PAI -name prediction
  -project algo_public
  -DinputTableName=pai_rf_test_input
  -DmodelName=pai_rf_test_model
  -DresultColName=prediction_result
  -DscoreColName=prediction_score
  -DdetailColName=prediction_detail
  -DoutputTableName=pai_temp_2283_76333_1
```

4. 查看输出结果表pai_temp_2283_76333_1。

prediction_result ▲	prediction_score ▲	prediction_detail ▲
bad	0.7777777777777778	{"bad": 0.7777777777777778, "good": 0.2222222222222222}
bad	0.8333333333333334	{"bad": 0.8333333333333334, "good": 0.1666666666666667}
good	0.8333333333333334	{"bad": 0.1666666666666667, "good": 0.8333333333333334}
good	0.8888888888888888	{"bad": 0.1111111111111111, "good": 0.8888888888888888}
good	0.8888888888888888	{"bad": 0.1111111111111111, "good": 0.8888888888888888}

3.6. 时间序列

3.6.1. x13_arima

13-arima是基于开源X-13ARIMA-SEATS封装的针对季节性调整的Arima算法。

背景信息

Arima全称为自回归积分滑动平均模型（Autoregressive Integrated Moving Average Model），是由博克思（Box）和詹金斯（Jenkins）于70年代初提出的著名时间序列预测方法，所以又称为box-jenkins模型、博克思-詹金斯法。

可视化

下面为您介绍可视化配置x13_arima时间序列组件的参数：

页签	参数	描述
字段设置	**时序列**	必选，仅用来对数值列排序，具体数值与计算无关。
	数值列	必选
	分组列	可选，多列以半角逗号（,）分隔，例如col0,col1，每个分组会构建一个时间序列。
参数设置	**格式**	支持输入的格式为p,d,q。 p、d和q均为非负整数，取值范围为[0, 36]。 • p：自回归系数 • d：差分 • q：滑动回归系数
	开始日期	支持输入的格式为year.seasonal。例如1986.1。
	series频率	支持输入正整数，取值范围为12。

参数设置 页签	参数	描述
	格式	支持输入的格式为sp,sd,sq。sp、sd和sq均为非负整数，取值范围为[0, 36]。 • sp：季节性自回归系数 • sd：季节性差分 • sq：季节性滑动回归系数
	seasonal周期	支持输入数字，取值范围为(0，12]。默认值为12。
	预测条数	支持输入数字，取值范围为(0，120]。默认值为12。
	预测置信水平	支持输入数字，取值范围为(0, 1)默认值为0.95。
执行调优	**核数目**	节点个数，默认自动计算。
	内存数	单个节点内存大小，单位为MB。

命令方式

```
PAI -name x13_arima
  -project algo_public
  -DinputTableName=pai_ft_x13_arima_input
  -DseqColName=id
  -DvalueColName=number
  -Dorder=3,1,1
  -Dstart=1949.1
  -Dfrequency=12
  -Dseasonal=0,1,1
  -Dperiod=12
  -DpredictStep=12
  -DoutputPredictTableName=pai_ft_x13_arima_out_predict
  -DoutputDetailTableName=pai_ft_x13_arima_out_detail
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表中，用于训练的特征列名。	默认选择所有分区
seqColName	是	时序列。仅用来对valueColName排序。	无
valueColName	是	数值列。	无

参数	是否必选	描述	默认值
groupColNames	否	分组列，多列用逗号分隔，例如col0,col1。每个分组会构建一个时间序列。	无
order	是	p、d和q分别表示自回归系数、差分、滑动回归系数。取值均为非负整数，范围为[0, 36]。	无
start	否	时序开始日期。字符串类型，格式为year.seasonal，例如1986.1。请参见时序格式介绍。	1.1
frequency	否	时序频率。正整数类型，范围为(0, 12]。请参见时序格式介绍。	12 **说明** 12表示12月/年。
seasonal	否	sp、sd和sq分别表示季节性自回归系数、季节性差分、季节性滑动回归系数。取值均为非负整数，范围为[0, 36]。	无seasonal
period	否	seasonal周期。数字类型，取值范围为(0, 100]。	frequency
maxiter	否	最大迭代次数。正整数类型。	1500
tol	否	容忍度，DOUBLE类型。	1e-5
predictStep	否	预测条数。数字类型，取值范围为(0, 365]。	12
confidenceLevel	否	预测置信水平。数字类型，取值范围为(0, 1)。	0.95
outputPredictTableName	是	预测输出表。	无
outputDetailTableName	是	详细信息表。	无
outputTablePartition	否	输出分区，分区名。	默认不输出到分区
coreNum	否	节点个数，与参数memSizePerCore配对使用，正整数。	默认自动计算

参数	是否必选	描述	默认值
memSizePerCore	否	单个节点内存大小，单位为MB。正整数，取值范围为[1024, 64 *1024]。	默认自动计算
lifecycle	否	指定输出表的生命周期。	默认没有生命周期

时序格式介绍

参数start和frequency规定了数据（valueColName）的两个时间维度ts1、ts2：

- frequency：表示单位周期内数据的频率，即单位ts1中ts2的频率。
- start：格式为 `n1.n2` ，表示开始日期是第n1个ts1中的第n2个ts2。

单位时间	ts1	ts2	frequency	start
12月/年	年	月	12	1949.2 表示第1949年中的第2个月
4季/年	年	季	4	1949.2 表示第1949年中的第2个季度
7天/周	周	天	7	1949.2 表示第1949周中的第2天
1	任何时间单位	1	1	1949.1 表示第1949（年、天、时等）

例如value=[1,2,3,5,6,7,8,9,10,11,12,13,14,15]

- `start=1949.3，frequency=12` 表示数据是12月/年，预测开始日期是1950.06。

year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct		
1949			1	2	3	4	5	6	7	8	9	10
1950	11	12	13	14	15							

- `start=1949.3，frequency=4` 表示数据是4季/年，预测开始的日期是1953.02。

year	Qtr1	Qtr2	Qtr3	Qtr4
1949			1	2
1950	3	4	5	6
1951	7	8	9	10
1952	11	12	13	14

year	Qtr1	Qtr2	Qtr3	Qtr4
1953	14	15		

- `start=1949.3，frequency=7` 表示数据是7天/周，预测开始的日期是1951.04。

week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1949			1	2	3	4	5
1950	6	7	8	9	10	11	12
1951	13	14	15				

- `start=1949.1，frequency=1` 表示任何时间单位，预测开始日期是1963.00。

cycle	p1
1949	1
1950	2
1951	3
1952	4
1953	5
1954	6
1955	7
1956	8
1957	9
1958	10
1959	11
1960	12
1961	13
1962	14
1963	15

具体示例

测试数据

使用的数据集为 AirPassengers

id	number
1	112
2	118
3	132
4	129
5	121
...	...

使用tunnel命令行工具上传数据，命令如下。

```
create table pai_ft_x13_arima_input(id bigint,number bigint);
tunnel upload data/airpassengers.csv pai_ft_x13_arima_input -h true;
```

PAI 命令

```
PAI -name x13_arima
  -project algo_public
  -DinputTableName=pai_ft_x13_arima_input
  -DseqColName=id
  -DvalueColName=number
  -Dorder=3,1,1
  -Dseasonal=0,1,1
  -Dstart=1949.1
  -Dfrequency=12
  -Dperiod=12
  -DpredictStep=12
  -DoutputPredictTableName=pai_ft_x13_arima_out_predict
  -DoutputDetailTableName=pai_ft_x13_arima_out_detail
```

输出说明

- 输出表outputPredictTableName

column name	comment
pdate	预测日期。
forecast	预测结论。
lower	置信度为 confidenceLevel（默认0.95）时，预测结论下界。
upper	置信度为 confidenceLevel（默认0.95）时，预测结论上界。

数据展示如下。

pdate	forecast	lower	upper
1	338.137741861791	153.024741712748	523.250742010834
2	312.051016375618	111.626955045557	512.475077705679
3	320.733042630875	105.372184116693	536.093901145057
4	331.738858260638	105.061267261819	558.416449259457
5	334.436605040811	99.2217920238739	569.651418057748
6	320.48563667084	77.9180949329431	563.053178408737
7	315.77614369154	66.7270692190054	564.825218164075
8	324.593521277343	69.629334007144	579.557708547541
9	335.095482316107	74.6003000456564	595.590664586558
10	317.960431620407	52.2122280976782	583.708635143136
11	330.473685507076	59.6790629665311	601.268308047621
12	344.912483954059	69.2318322886809	620.593135619437

- 输出表outputDetailTableName

  字段说明

column name	comment
key	○ model：表示模型 ○ evaluation：表示评估结果 ○ parameters：表示训练参数 ○ log：表示训练日志
summary	存储具体信息。

数据展示

序号	key	summary
1	model	{"comment":{"ma":"arima estimate","mr":"regress...
2	evaluation	{"comment":{"aic":"AIC","aicc":"AICC (F-correcte...
3	paramters	{"arima":{"d":1,"isSeasonal":true,"p":3,"period":...
4	log	1 Log for X-13ARIMA-SEATS program (Version 1.1...

PaiWeb展示-模型系数（key=model）

operator	factor	period	lag	estimate	standard error
AR	Nonseasonal	1	1	0.6135	0.0928
AR	Nonseasonal	1	2	0.2403	0.1035
AR	Nonseasonal	1	3	-0.0732	0.0906
MA	Nonseasonal	1	1	0.9737	0.0376
MA	Seasonal	12	12	0.1051	0.1031

PaiWeb展示-评估指标（key=evaluation）

Name	Indicator
AIC	1019.6973
BIC	1036.9485
Hannan Quinn	1026.7072
Log likelihood	-503.8487
Effective number of observations	131
Number of observations	144
variance	127.0384

算法规模

- 支持规模
  - 行：单Group数据最大1200条
  - 列：1数值列
- 资源计算方式
  - 不设置groupColNames，默认计算方式

    ```
    coreNum=1
    memSizePerCore=4096
    ```

  - 设置groupColNames，默认计算方式

    ```
    coreNum=floor(总数据行数/12万)
    memSizePerCore=4096
    ```

常见问题

为什么预测结果都一样？

在模型训练异常时，会调用均值模型，则所有预测结果都是训练数据的均值。

常见的异常包括**时序差分diff后不稳定**、**训练没有收敛**、**方差为0**等，可以在logview中查看单独节点的stderr文件，获取具体的异常信息。

3.6.2. x13_auto_arima

x13-auto-arima包括自动ARIMA模型选择程序，主要基于TRMO（1996）及后续修订中实施的Gomez和Maravall（1998）的程序。

背景信息

x13_auto_arima选择过程如下：

- **default model esitmation**

  当 `frequency = 1` 时，默认模型是 `(0,1,1)` 。

当 frequency > 1 时，默认模型是 (0,1,1)(0,1,1) 。

- identication of dierencing orders

  如果设置了diff 和 seasonalDiff，则跳过此步骤。

  使用 Unit root test (wiki) 确定差分d，和季节性差分D。

- identication of ARMA model orders

  根据 BIC(wiki) 准则选择最合适的模型， 其参数maxOrder、maxSeasonalOrder在此步骤起作用。

- comparison of identified model with default model

  使用Ljung-Box Q statistic(wiki) 比较模型，如果两个模型均是不可接受的，则使用 (3,d,1)(0,D,1) 模型。

- final model checks

Arima详细介绍请参见wiki。

可视化

下面为您介绍可视化配置x13-auto-arima时间序列组件的参数：

页签	参数	描述
字段设置	时序列	必选，仅用来对数值列排序，具体数值与计算无关。
	数值列	必选
	分组列	可选，多列以逗号（,）分隔，例如col0,col1，每个分组会构建一个时间序列。
参数设置	开始日期	支持输入的格式为year.seasonal。例如1986.1。
	series频率	支持输入正整数，取值范围为(0,12]。
	p,q最大值	支持输入正整数，取值范围为(0,4]。
	季节性p，q最大值	支持输入数字，取值范围为(0，2]。
	差分d最大值	支持输入数字，正整数，取值范围为(0，2]。
	季节性差分d最大值	支持输入数字，正整数。取值范围为(0, 1]。
	差分d	支持输入数字，正整数。取值范围为(0, 2]。 diff与maxDiff同时设置时，maxDiff被忽略。 diff与seasonalDiff要同时设置。

页签	参数	描述
	季节性差分d	支持输入数字，正整数。取值范围为(0, 1]。 seasonalDiff与maxSeasonalDiff同时设置时，maxSeasonalDiff被忽略。
	预测条数	支持输入数字，正整数。取值范围为(0, 120]。
	预测置信区间	默认值为0.95。
	容忍度	可选，默认值为1e-5。
	最大迭代次数	支持输入数字，正整数。默认值为1500。
执行调优	**核数目**	节点个数，默认自动计算。
	内存数	单个节点内存大小，单位为MB。

命令方式

```
PAI -name x13_auto_arima
  -project algo_public
  -DinputTableName=pai_ft_x13_arima_input
  -DseqColName=id
  -DvalueColName=number
  -Dstart=1949.1
  -Dfrequency=12
  -DpredictStep=12
  -DoutputPredictTableName=pai_ft_x13_arima_out_predict2
  -DoutputDetailTableName=pai_ft_x13_arima_out_detail2
```

参数	是否必选	描述	默认值
inputTableName	是	输入表的名称	无
inputTablePartitions	否	输入表中，用于训练的特征列名	默认选择所有分区
seqColName	是	时序列。仅用来对valueColName排序。	无
valueColName	是	数值列	无
groupColNames	否	分组列，多列用逗号分隔，如“col0,col1”。每个分组会构建一个时间序列	无

参数	是否必选	描述	默认值
start	否	时序开始日期，字符串，格式为 `vear.seasonal` ，例如 `1986.1` 。请参见时序格式介绍。	1.1
frequency	否	时序频率，正整数，范围为(0, 12]。请参见时序格式介绍。	12 **说明** 12表示12月/年。
maxOrder	否	p，q最大值，正整数，范围为[0,4]。	2
maxSeasonalOrder	否	季节性p，q最大值，正整数，范围为[0,2]。	1
maxDiff	否	差分d最大值。正整数，范围为[0,2]。	2
maxSeasonalDiff	否	季节性差分d最大值。正整数，范围为[0,1]。	1
diff	否	差分d，正整数，范围为[0,2]。 diff与maxDiff同时设置时，maxDiff被忽略。 diff与seasonalDiff要同时设置。	-1 **说明** 取值为-1表示不指定diff。
seasonalDiff	否	季节性差分d。正整数，范围为[0,1]。 seasonalDiff与maxSeasonalDiff同时设置时，maxSeasonalDiff被忽略。	-1 **说明** 取值为-1表示不指定seasonalDiff。
maxiter	否	最大迭代次数，正整数	1500
tol	否	容忍度，DOUBLE类型。	1e-5
predictStep	否	预测条数，数字，范围为(0, 365]。	12
confidenceLevel	否	预测置信水平，数字，范围为(0, 1)。	0.95

参数	是否必选	描述	默认值
outputPredictTableName	是	预测输出表	无
outputDetailTableName	是	详细信息表	无
outputTablePartition	否	输出分区，分区名。	默认不输出到分区
coreNum	否	节点个数，与参数memSizePerCore配对使用，正整数。	默认自动计算
memSizePerCore	否	单个节点内存大小，单位为MB。正整数，范围为[1024, 64 *1024]。	默认自动计算
lifecycle	否	指定输出表的生命周期。	默认没有生命周期

时序格式介绍

参数start和frequency规定了数据（valueColName）的两个时间维度ts1、ts2：

- frequency：表示单位周期内数据的频率，即单位ts1中ts2的频率。
- start：格式为 `n1.n2` ，表示开始日期是第n1个ts1中的第n2个ts2。

单位时间	ts1	ts2	frequency	start
12月/年	年	月	12	1949.2表示第1949年中的第2个月
4季/年	年	季	4	1949.2表示第1949年中的第2个季度
7天/周	周	天	7	1949.2表示第1949周中的第2天
1	任何时间单位	1	1	1949.1表示第1949（年、天、时等）

例如value=[1,2,3,5,6,7,8,9,10,11,12,13,14,15]

- `start=1949.3，frequency=12` 表示数据是12月/年，预测开始日期是1950.06。

year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1949			1	2	3	4	5	6	7	8	9	10
1950	11	12	13	14	15							

- `start=1949.3，frequency=4` 表示数据是4季/年，预测开始的日期是1953.02。

year	Qtr1	Qtr2	Qtr3	Qtr4
1949			1	2
1950	3	4	5	6
1951	7	8	9	10
1952	11	12	13	14
1953	15			

- `start=1949.3，frequency=7` 表示数据是7天/周，预测开始的日期是1951.04。

week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1949			1	2	3	4	5
1950	6	7	8	9	10	11	12
1951	13	14	15				

- `start=1949.1，frequency=1` 表示任何时间单位，预测开始日期是1963.00。

cycle	p1
1949	1
1950	2
1951	3
1952	4
1953	5
1954	6
1955	7
1956	8
1957	9
1958	10
1959	11
1960	12
1961	13
1962	14

cycle	p1
1963	15

具体示例

测试数据

使用的数据集为 AirPassengers，是1949～1960年每个月国际航空的乘客数量，如下表所示。

id	number
1	112
2	118
3	132
4	129
5	121
...	...

使用tunnel命令行工具上传数据，命令如下。

```
create table pai_ft_x13_arima_input(id bigint,number bigint);
tunnel upload data/airpassengers.csv pai_ft_x13_arima_input -h true;
```

PAI 命令

```
PAI -name x13_auto_arima
  -project algo_public
  -DinputTableName=pai_ft_x13_arima_input
  -DseqColName=id
  -DvalueColName=number
  -Dstart=1949.1
  -Dfrequency=12
  -DmaxOrder=4
  -DmaxSeasonalOrder=2
  -DmaxDiff=2
  -DmaxSeasonalDiff=1
  -DpredictStep=12
  -DoutputPredictTableName=pai_ft_x13_arima_auto_out_predict
  -DoutputDetailTableName=pai_ft_x13_arima_auto_out_detail
```

输出说明

- 输出表outputPredictTableName

column name	comment
pdate	预测日期。

column name	comment
forecast	预测结论。
lower	置信度为confidenceLevel（默认0.95）时，预测结论下界。
upper	置信度为confidenceLevel（默认0.95）时，预测结论上界。

数据展示

pdate	forecast	lower	upper
1	338.137741861791	153.024741712748	523.250742010834
2	312.051016375618	111.626955045557	512.475077705679
3	320.733042630875	105.372184116693	536.093901145057
4	331.738858260638	105.061267261819	558.416449259457
5	334.436605040811	99.2217920238739	569.651418057748
6	320.48563667084	77.9180949329431	563.053178408737
7	315.77614369154	66.7270692190054	564.825218164075
8	324.593521277343	69.629334007144	579.557708547541
9	335.095482316107	74.6003000456564	595.590664586558
10	317.960431620407	52.2122280976782	583.708635143136
11	330.473685507076	59.6790629665311	601.268308047621
12	344.912483954059	69.2318322886809	620.593135619437

- 输出表outputDetailTableName

  字段说明

column name	comment
key	○ model：表示模型 ○ evaluation：表示评估结果 ○ parameters：表示训练参数 ○ log：表示训练日志
summary	存储具体信息。

数据展示

序号	key	summary
1	model	{"comment": {"ma": "arima estimate", "mr": "regress...
2	evaluation	{"comment": {"aic": "AIC", "aicc": "AICC (F-correcte...
3	paramters	{"arima": {"d": 1, "isSeasonal": true, "p": 3, "period":...
4	log	1 Log for X-13ARIMA-SEATS program (Version 1.1...

PaiWeb展示-模型系数（key=model）

operator	factor	period	lag	estimate	standard error
AR	Nonseasonal	1	1	0.6135	0.0928
AR	Nonseasonal	1	2	0.2403	0.1035
AR	Nonseasonal	1	3	-0.0732	0.0906
MA	Nonseasonal	1	1	0.9737	0.0376
MA	Seasonal	12	12	0.1051	0.1031

PaiWeb展示-评估指标（key=evaluation）

Name	Indicator
AIC	1019.6973
BIC	1036.9485
Hannan Quinn	1026.7072
Log likelihood	-503.8487
Effective number of observations	131
Number of observations	144
variance	127.0384

算法规模

- 支持规模
  - 行：单group数据最大1200条
  - 列：1数值列
- 资源计算方式
  - 不设置groupColNames，默认计算方式

    ```
    coreNum=1
    memSizePerCore=4096
    ```

  - 设置groupColNames，默认计算方式

    ```
    coreNum = floor(总数据行数/12万)
    memSizePerCore = 4096
    ```

常见问题

为什么预测结果都一样？

在模型训练异常时，会调用均值模型，则所有预测结果都是训练数据的均值。

常见的异常包括**时序差分diff后不稳定**、**训练没有收敛**、**方差为0**等，可以在logview中查看单独节点的stderr文件，获取具体的异常信息。

3.7. 文本分析

3.7.1. Split Word

本文为您介绍PAI-Studio提供的Split Word算法组件。

Split Word算法组件基于AliWS（Alibaba Word Segmenter）词法分析系统，对指定列的内容进行分词，分词后的各个词语之间以空格分隔。如果您配置了词性标注或语义标注相关参数，则系统会将分词、词性标注和语义标注结果一同输出，其中词性标注分隔符为正斜线（/），语义标注分隔符为竖线（|）。

Split Word仅支持中文淘宝分词（TAOBAO_CHN）和互联网分词（INTERNET_CHN）。

PAI-Studio支持通过可视化或PAI命令方式，配置Split Word算法组件相关参数。

配置组件

- 可视化方式

页签	参数	描述
字段设置	**列名**	用来进行分词的列名。
参数设置	**识别选项**	识别内容类型。取值范围为： ○ 识别简单实体 ○ 识别人名 ○ 识别机构名 ○ 识别电话号码 ○ 识别时间 ○ 识别日期 ○ 识别数字字母 默认值为：识别简单实体、识别电话号码、识别时间、识别日期和识别数字字母。
	合并选项	合并内容类型。取值范围为： ○ 合并中文数字 ○ 合并阿拉伯数字 ○ 合并中文日期 ○ 合并中文时间 默认值为：合并阿拉伯数字。
	过滤器	过滤器的类型。取值范围为TAOBAO_CHN和INTERNET_CHN。默认值为TAOBAO_CHN。
	Pos Tagger	是否进行词性标注。默认进行词性标注。
	Semantic Tagger	是否进行语义标注。默认不进行语义标注。
	过滤分词结果为数字的词	是否过滤分词结果为数字的词。默认不过滤。

页签	参数	描述
	过滤分词结果为全英文的词	是否过滤分词结果为全英文的词。默认不过滤。
	过滤分词结果为标点符号的词	是否过滤分词结果为标点符号的词。默认不过滤。
执行调优	**核心数**	默认为系统自动分配。
	每个核的内存数	默认为系统自动分配。

- PAI命令方式

```
pai -name split_word_model
  -project algo_public
  -DoutputModelName=aliws_model
  -DcolName=content
  -Dtokenizer=TAOBAO_CHN
  -DenableDfa=true
  -DenablePersonNameTagger=false
  -DenableOrgnizationTagger=false
  -DenablePosTagger=false
  -DenableTelephoneRetrievalUnit=true
  -DenableTimeRetrievalUnit=true
  -DenableDateRetrievalUnit=true
  -DenableNumberLetterRetrievalUnit=true
  -DenableChnNumMerge=false
  -DenableNumMerge=true
  -DenableChnTimeMerge=false
  -DenableChnDateMerge=false
  -DenableSemanticTagger=true
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
inputTablePartitions	否	输入表中参与分词的分区名。格式为 `partition name=value` 。多级分区格式为 `name1=value1/name2=value2` 。如果指定多个分区，用英文逗号（,）分隔。	所有分区
selectedColNames	是	输入表中用于分词的列名。如果指定多列，用英文逗号（,）分隔。	无
dictTableName	否	是否使用自定义词典表。自定义词典表只有一列，每一行是一个词。	无
tokenizer	否	过滤器类型。取值为TAOBAO_CHN或INTERNET_CHN。	TAOBAO_CHN
enableDfa	否	是否识别简单实体。取值为True或False。	True

参数名称	是否必选	描述	默认值
enablePersonNameTagger	否	是否识别人名。取值为True或False。	False
enableOrgnizationTagger	否	是否识别机构名。取值为True或False。	False
enablePosTagger	否	是否进行词性标注。取值为True或False。	False
enableTelephoneRetrievalUnit	否	是否识别电话号码。取值为True或False。	True
enableTimeRetrievalUnit	否	是否识别时间号码。取值为True或False。	True
enableDateRetrievalUnit	否	是否识别日期号码。取值为True或False。	True
enableNumberLetterRetrievalUnit	否	是否识别数字字母。取值为True或False。	True
enableChnNumMerge	否	是否将中文数字合并为一个检索单元。取值为True或False。	False
enableNumMerge	否	是否将普通数字合并为一个检索单元。取值为True或False。	True
enableChnTimeMerge	否	是否将中文时间合并为一个语义单元。取值为True或False。	False
enableChnDateMerge	否	指定是否将中文日期合并为一个语义单元。取值为True或False。	False
enableSemanticTagger	否	是否进行语义标注。取值为True或False。	False
outputTableName	是	输出表的名称。	无
outputTablePartition	否	输出表的分区名称。	无
coreNum	否	节点个数，需要与memSizePerCore参数同时设置才生效。取值为正整数，范围为[1,9999]。	系统自动分配
memSizePerCore	否	单个节点内存大小，单位为MB。取值为正整数，范围为[1024,64×1024]。	系统自动分配
lifecycle	否	输出表的生命周期。取值为正整数。	无

如果表为普通表，不建议您设置coreNum和memSizePerCore，Split Word算法组件会自动计算。

在资源有限的情况下，您可以使用如下代码计算节点个数和单个节点内存。

```
def CalcCoreNumAndMem(row, col, kOneCoreDataSize=1024):
  """计算节点个数和单个节点内存
    Args:
      row：输入表行数
      col：输入表列数
      kOneCoreDataSize：单个节点计算的数据量，单位MB，正整数，默认为1024
    Return:
      coreNum, memSizePerCore
    Example:
      coreNum, memSizePerCore = CalcCoreNumAndMem(1000,99, 100, kOneCoreDataSize=2048)
  """
  kMBytes = 1024.0 * 1024.0
  #按数量划分，计算节点个数
  coreNum = max(1, int(row * col * 1000/ kMBytes / kOneCoreDataSize))
  #单个节点内存 = 数据量大小
  memSizePerCore = max(1024, int(kOneCoreDataSize*2))
  return coreNum, memSizePerCore
```

示例

- 生成数据

```
create table pai_aliws_test
as select
  1 as id,
  '今天是个好日子，天气好晴朗。' as content
from dual;
```

- PAI命令行

```
pai -name split_word
  -project algo_public
  -DinputTableName=pai_aliws_test
  -DselectedColNames=content
  -DoutputTableName=doc_test_split_word
```

- 输入说明

  输入包含两列的表，第一列是文档ID，第二列是文档内容。

```
+------------+------------+
| id     | content   |
+------------+------------+
| 1      | 今天是个好日子，天气好晴朗。 |
```

- 输出说明
  - 输出原表中的分词列的分词，其余列原样输出。
  - 使用自定义词典时，系统会按照自定义词典和上下文来分词，不会完全按照自定义词典分词。

3.7.2. 三元组转kv

本文为您介绍PAI-Studio提供的三元组转kv算法组件。

三元组转kv算法组件用于将三元组表（row,col,value）转换为kv表（row,[col_id:value]）。

三元组（row,col,value）表的数据类型为“XXD”或“XXL”，“X”表示任意数据类型，“D”表示DOUBLE数据类型，“L”表示BIGINT数据类型。转换为kv表后，row和value的数据类型与原始输入数据类型一致，col_id是BIGINT数据类型，并输出col的索引表映射到col_id。

转换示例如下。

- 三元组表如下。

id	word	count
01	a	10
01	b	20
01	c	30

- 输出kv表如下。

id	key_value
01	1:10;2:20;3:30

> **说明** key_value中key和value、kv对之间的分隔符可自定义。

- 输出索引表如下。

key	key_id
a	1
b	2
c	3

PAI-Studio支持通过可视化或PAI命令方式，配置三元组转kv算法组件相关参数。

配置组件

- 可视化方式

页签	参数	描述
字段设置	**转成kv表时保持不变的列名**	指定转换为kv表时，保持不变的列名称。
	输出kv中的key	kv表中的key。
	输出kv中的value	kv表中key的值。
	输入索引表key的列名	索引表中key的列名。
	输入索引表key索引号的列名	索引表中索引号的列名。

页签	参数	描述
	key和value之间分隔符	kv表中key和value之间的分隔符。默认值为冒号（:）。
	kv对之间分隔符	kv对之间的分隔符。默认值为英文逗号（,）。
执行调优	**指定实例总数**	指定Instance总数。取值范围为正整数。系统默认会根据输入数据量大小计算。
	指定内存（单位MB）	指定内存总数。取值范围为正整数。系统默认会根据输入数据量大小计算。

- PAI命令方式

```
PAI -name triple_to_kv
  -project algo_public
  -DinputTableName=test_data
  -DoutputTableName=test_kv_out
  -DindexOutputTableName=test_index_out
  -DidColName=id
  -DkeyColName=word
  -DvalueColName=count
  -DinputTablePartitions=ds=test1
  -DindexInputTableName=test_index_input
  -DindexInputKeyColName=word
  -DindexInputKeyIdColName=word_id
  -DkvDelimiter=:
  -DpairDelimiter=;
  -Dlifecycle=3
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
idColName	是	指定转换为kv表时，保持不变的列名称。	无
keyColName	是	kv表中的key。	无
valueColName	是	kv表中key的值。	无
outputTableName	是	输出kv表的名称。	无
indexOutputTableName	是	输出索引表的名称。	无
indexInputTableName	否	输入已有的索引表的名称。不能为空表。	无
indexInputKeyColName	否	索引表中key的列名。配置indexInputTableName后，必须配置该参数。	无

参数名称	是否必选	描述	默认值
indexInputKeyIdColName	否	索引表中索引号的列名。配置indexInputTableName后，必须配置该参数。	无
inputTablePartitions	否	输入表的分区名称，只能输入单个分区。	无
kvDelimiter	否	kv表中key和value之间的分隔符。	冒号（:）
pairDelimiter	否	kv对之间的分隔符。	英文逗号（,）
lifecycle	否	输出表的生命周期。	无
coreNum	否	指定Instance总数。取值为正整数。	系统默认会根据输入数据量大小计算
memSizePerCore	否	指定内存总数。取值为正整数。	系统默认会根据输入数据量大小计算

示例

- 输入数据

```
drop table if exists triple2kv_test_input;
create table triple2kv_test_input as
select
 *
from
(
 select '01' as id, 'a' as word, 10 as count from dual
  union all
   select '01' as id, 'b' as word, 20 as count from dual
  union all
   select '01' as id, 'c' as word, 30 as count from dual
  union all
   select '02' as id, 'a' as word, 100 as count from dual
  union all
   select '02' as id, 'd' as word, 200 as count from dual
  union all
   select '02' as id, 'e' as word, 300 as count from dual
) tmp;
```

- 运行PAI命令

```
PAI -name triple_to_kv
  -project algo_public
  -DinputTableName=triple2kv_test_input
  -DoutputTableName=triple2kv_test_input_out
  -DindexOutputTableName=triple2kv_test_input_index_out
  -DidColName=id
  -DkeyColName=word
  -DvalueColName=count
  -Dlifecycle=1;
```

- 运行结果
  - 输出triple2kv_test_input_out kv表

```
+------------+------------+
| id      | key_value |
+------------+------------+
| 02      | 1:100;4:200;5:300 |
| 01      | 1:10;2:20;3:30 |
+------------+------------+
```

  - 输出triple2kv_test_input_index_out索引表

```
+------------+------------+
| key     | key_id   |
+------------+------------+
| a      | 1      |
| b      | 2      |
| c      | 3      |
| d      | 4      |
| e      | 5      |
+------------+------------+
```

3.7.3. 字符串相似度

本文为您介绍PAI-Studio提供的字符串相似度算法组件。

计算字符串相似度是机器学习领域的一个基本操作，主要用于信息检索、自然语言处理和生物信息学等领域。字符串相似度算法组件支持Levenshtein（Levenshtein Distance）、lCS（Longest Common SubString）、SSK（String Subsequence Kernel）、Cosine（Cosine）和Simhash_Hamming五种相似度计算方式，支持两两计算。

- Levenshtein支持计算距离和相似度。
  - 距离在参数中表示为**levenshtein**。
  - 相似度=1-距离。在参数中表示为**levenshtein_sim**。
- lCS支持计算距离和相似度。
  - 距离在参数中表示为**lcs**。
  - 相似度=1-距离。相似度在参数中表示为**lcs_sim**。
- SSK支持计算相似度，在参数中表示为**ssk**。
- Cosine支持计算相似度，在参数中表示为**cosine**。
- Simhash_Hamming，其中SimHash算法是把原始的文本映射为64位的二进制指纹，Hamming Distance则

是计算二进制指纹在相同位置上不同字符的个数，支持计算距离和相似度。

- 距离在参数中表示为simhash_hamming。
- 相似度=1-距离/64.0。相似度在参数中表示为simhash_hamming_sim。

PAI-Studio支持通过可视化或PAI命令方式，配置字符串相似度算法组件相关参数。

配置组件

- 可视化方式

页签	参数	描述
字段设置	输出表追加的列名	指定输出表中追加的列名。
	相似度计算中第一列的列名	默认为表中第一个STRING类型的列名。
	相似度计算中第二列的列名	默认为表中第二个STRING类型的列名。
	输出表中相似度列的列名	指定输出表中相似度列的列名。
参数设置	相似度计算方法	指定相似度计算方法类型。取值范围为： - levenshtein - levenshtein_sim - lcs - lcs_sim - ssk - cosine - simhash_hamming - simhash_hamming_sim 默认值为levenshtein_sim。
	子串的长度	仅当相似度计算方法取值为ssk、cosine、simhash_hamming或simhash_hamming_sim时，才需要配置该参数。取值范围为(0,100)。默认值为2。
	匹配字符串的权重	仅当相似度计算方法取值为ssk时，才需要配置该参数。取值范围为(0,1)。默认值为0.5。
执行调优	计算的核心数	默认为系统自动分配。
	每个核心的内存（MB）	默认为系统自动分配。

- PAI命令方式

```
PAI -name string_similarity
  -project algo_public
  -DinputTableName="pai_test_string_similarity"
  -DoutputTableName="pai_test_string_similarity_output"
  -DinputSelectedColName1="col0"
  -DinputSelectedColName2="col1";
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
outputTableName	是	输出表的名称。	无
inputSelectedColName1	否	相似度计算中第一列的列名。	表中第一个STRING类型的列名
inputSelectedColName2	否	相似度计算中第二列的列名。	表中第二个STRING类型的列名
inputAppendColNames	否	输出表追加的列名。	无
inputTablePartitions	否	输入表的分区名称。	所有分区
outputColName	否	输出表中相似度列的列名。列名中不能有特殊字符，只能使用英文a~z，A~Z、数字或下划线（_），且以字母开头，长度不超过128字节。	output
method	否	相似度计算方法类型。取值范围为： ○ levenshtein ○ levenshtein_sim ○ lcs ○ lcs_sim ○ ssk ○ cosine ○ simhash_hamming ○ simhash_hamming_sim	levenshtein_sim
lambda	否	仅当**相似度计算方法**取值为**ssk**时，才需要配置该参数。取值范围为(0,1)。	0.5
k	否	仅当**相似度计算方法**取值为**ssk**、**cosine**、**simhash_hamming**或**simhash_hamming_sim**时，才需要配置该参数。取值范围为(0,100)。	2
lifecycle	否	输出表的生命周期。取值范围为正整数。	无
coreNum	否	计算的核心数。	系统自动分配

参数名称	是否必选	描述	默认值
memSizePerCore	否	每个核心的内存数。	系统自动分配

示例

- 输入数据

```
create table pai_ft_string_similarity_input as select * from
(select 0 as id, "北京" as col0, "北京" as col1 from dual
union all
select 1 as id, "北京" as col0, "北京上海" as col1 from dual
union all
select 2 as id, "北京" as col0, "北京上海香港" as col1 from dual
)tmp;
```

- 运行PAI命令

```
PAI -name string_similarity
  -project sre_mpi_algo_dev
  -DinputTableName=pai_ft_string_similarity_input
  -DoutputTableName=pai_ft_string_similarity_output
  -DinputSelectedColName1=col0
  -DinputSelectedColName2=col1
  -Dmethod=simhash_hamming
  -DinputAppendColNames=col0,col1;
```

- 输出结果
  - 使用simhash_hamming计算方法的输出结果如下。

col0	col1	output
北京	北京	0
北京	北京上海	6
北京	北京上海香港	13

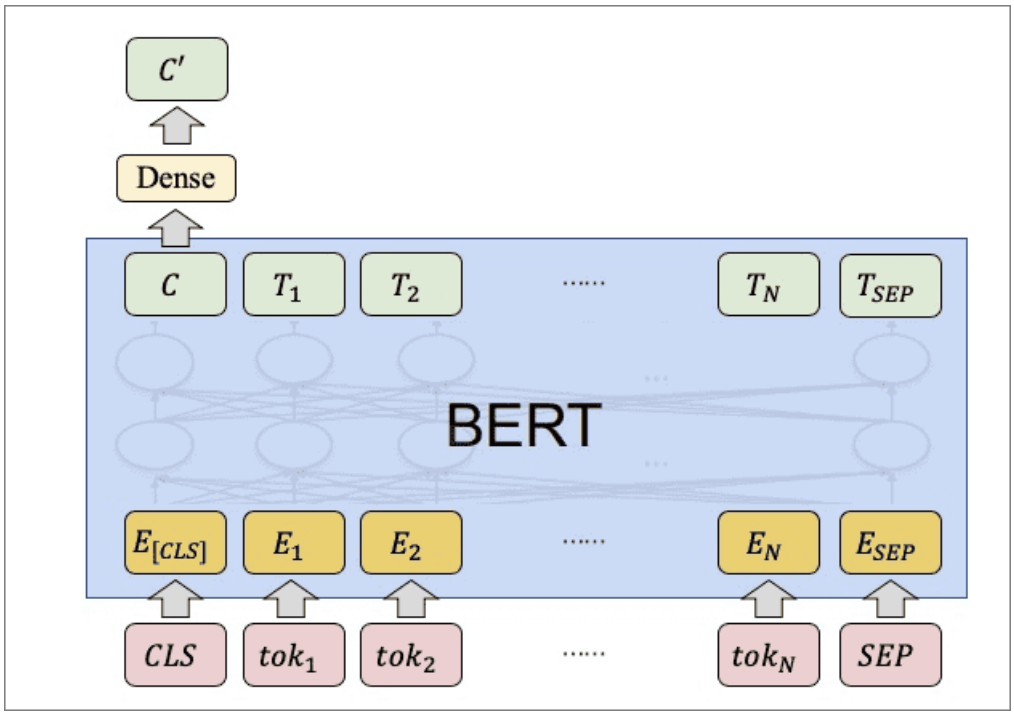
  - 使用simhash_hamming_sim计算方法的输出结果如下。

col0	col1	output
北京	北京	1
北京	北京上海	0.90625
北京	北京上海香港	0.796875

3.7.4. BERT文本向量化

BERT文本向量化是以原始文本作为输入，系统提取特征后输出一个向量序列。您还可以将CLS输出的向量经过Dense后的向量作为整个句子的句向量。本文为您介绍PAI-Studio提供的BERT文本向量化。

BERT文本向量化组件以原始文本作为输入，端到端输出经过BERT后的向量。



- pool_output ：图中的C'，即对句子进行编码后的向量。
- first_token_output ：图中的C。
- all_hidden_outputs ：图中的[C, T, T, ..., TN, TSEP]。

BERT文本向量化组件拥有以下特性：

- 命令简单，最短只需要4个PAI命令参数。
- MaxCompute表端到端输出，输入原始数据，输出向量，仅需指定输出表名。
- 支持对输入表中字段添加到输出表中。

价格说明

本组件调用GPU计算资源，收费标准与其它文本分析组件不同。其它文本分析组件1.7元每计算时。BERT组件北京集群（P100卡）12元每计算时，上海集群（M40卡）8.4元每计算时。

配置组件

您可以通过以下任意一种方式，配置BERT文本向量化参数：

- 可视化方式

页签	参数	描述
字段设置	第一文本列选择	选择第一列文本的输出字段。
	第二列文本列选择	选择第二列文本的输出字段。
	附加列	选择附加列文本的输出字段。
	batchSize	默认值为256。

页签	参数	描述
参数设置	sequenceLength	默认值为128。
	输出向量选项	支持pool_output、first_token_output和all_hidden_outputs。
	模型选择	支持pai-bert-base-zh、pai-bert-small-zh和pai-bert-large-zh。
执行调优	指定Worker数	支持输入1、2、3和4。
	指定Worker的GPU卡数	支持输入1和2。
	指定Worker的CPU卡数	默认值为1。

- PAI命令方式

```
PAI -name ez_bert_feat_ext
  -DinputTable=odps://{project}/tables/{表名}
  -DoutputTable=odps://{project}/tables/{表名}
  -DfirstSequence=col0
  -DsecondSequence=''
  -DappendCols=co1,col2,col0
  -DoutputSchema=pool_output,first_token_output,all_hidden_outputs
  -DsequenceLength=128
    -DmodelName=pai-bert-base-zh
  -DbatchSize=100
  -DworkerCount=1
  -DworkerCPU=1
  -DworkerGPU=1
  -Dbuckets=oss://atp-modelzoo/tmp/?role_arn=${role_arn}&host=oss-cn-hangzhou.aliyuncs.com
```

参数名称	是否必选	描述	默认值
inputTable	是	输入待特征提取文本表格。STRING类型，格式为project.table。	无
outputTable	是	输出特征表格。STRING类型，格式为`project.table`。	无
firstSequence	是	第一个文本序列在输入格式中对应的列名。STRING类型。	无
secondSequence	否	第二个文本序列在输入格式中对应的列名。STRING类型。	默认为空。
appendCols	否	用户输入表中添加到输出的列。STRING类型。	默认为空。

参数名称	是否必选	描述	默认值
outputSchema	否	选择输出数据中需要哪几个特征。STRING类型。	'pool_output','pool_output,first_token_output,all_hidden_outputs'（支持选择多个特征）
sequenceLength	否	序列整体最大长度。INT类型。范围为1~512。	128
modelName	否	预训练模型名。STRING类型。	pai-bert-base-zh。
batchSize	否	特征提取批大小。STRING类型。	256
workerCount	否	指定Worker数。INT类型。	1个Worker
workerGPU	否	指定Worker的GPU卡数，标识是否使用GPU。INT类型。	1张卡
workerCPU	否	指定Worker的CPU卡数，标识是否使用CPU。INT类型。	1张卡

3.7.5. 字符串相似度-topN

本文为您介绍PAI-Studio提供的字符串相似度-topN算法组件。

字符串相似度-topN算法组件用于计算字符串相似度并筛选出最相似的Top N个数据。

PAI-Studio支持通过可视化或PAI命令方式，配置字符串相似度-topN算法组件相关参数。

配置组件

- 可视化方式

页签	参数	描述
字段设置	**输入表在输出表追加的列名**	输入表中需要在输出表中追加的列名。
	映射表在输出表追加的列名	映射表中需要在输出表中追加的列名。
	相似度计算中左表的列名	在相似度计算中，左表的列名。
	相似度计算中映射表的列名	在相似度计算中，映射表的列名。左表中的每一行都会和映射表中所有的字符串计算出相似度，并最终以Top N的方式输出计算结果。

页签	参数	描述
	输出表中相似度列的列名	输出表中相似度列的列名。列名中不能有特殊字符，只能使用英文a~z、A~Z、数字或下划线（_），且以字母开头，长度不超过128字节。默认值为output。
参数设置	**最终给出的相似度最大值的个数**	输出Top相似度的个数。取值范围为正整数，默认值为10。
	相似度计算方法	指定相似度计算方法类型。取值范围为： ○ levenshtein_sim ○ lcs_sim ○ ssk ○ cosine ○ simhash_hamming_sim 默认值为levenshtein_sim。
	子串的长度	仅当**相似度计算方法**取值为ssk、cosine或simhash_hamming_sim时，才需要配置该参数。取值范围为(0,100)。默认值为2。
	匹配字符串的权重	仅当**相似度计算方法**取值为ssk时，才需要配置该参数。取值范围为(0,1)。默认值为0.5。
执行调优	**计算的核心数**	默认为系统自动分配。
	每个核心的内存（MB）	默认为系统自动分配。

- PAI命令方式

```
PAI -name string_similarity_topn
  -project algo_public
  -DinputTableName="pai_test_string_similarity_topn"
  -DoutputTableName="pai_test_string_similarity_topn_output"
  -DmapTableName="pai_test_string_similarity_map_topn"
  -DinputSelectedColName="col0"
  -DmapSelectedColName="col1";
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的名称。	无
mapTableName	是	映射表的名称。	无
outputTableName	是	输出表的名称。	无
inputSelectedColName1	否	在相似度计算中，左表的列名。	表中第一个STRING类型的列名

参数名称	是否必选	描述	默认值
inputSelectedColName2	否	在相似度计算中，映射表的列名。	表中第一个STRING类型的列名
inputAppendColNames	否	输入表中需要在输出表追加的列名。	无
inputAppendRenameColNames	否	输入表中需要在输出表追加的列名的别名。	无
mapSelectedColName	是	相似度计算中字典表的列名。	无
mapAppendColNames	否	映射表中需要在输出表追加的列名。	无
mapAppendRenameColNames	否	映射表中需要在输出表追加的列名的别名。	无
inputTablePartitions	否	输入表的分区名称。	所有分区
mapTablePartitions	否	映射表的分区名称。	所有分区
outputColName	否	输出表中相似度列的列名。列名中不能有特殊字符，只能使用英文a~z，A~Z、数字或下划线（_），且以字母开头，长度不超过128字节。	output
method	否	相似度计算方法类型。取值范围为： ○ levenshtein_sim ○ lcs_sim ○ ssk ○ cosine ○ simhash_hamming_sim	levenshtein_sim
lambda	否	仅当**相似度计算方法**取值为ssk时，才需要配置该参数。取值范围为(0,1)。	0.5
k	否	仅当**相似度计算方法**取值为ssk、cosine或simhash_hamming_sim时，才需要配置该参数。取值范围为(0,100)。	2
lifecycle	否	输出表的生命周期。取值范围为正整数。	无
coreNum	否	计算的核心数。	系统自动分配
memSizePerCore	否	每个核心的内存数。	系统自动分配

示例

- 输入数据

```
create table pai_ft_string_similarity_topn_input as select * from
(select 0 as id, "北京" as col0 from dual
union all
select 1 as id, "北京上海" as col0 from dual
union all
select 2 as id, "北京上海香港" as col0 from dual
)tmp
```

- 运行PAI命令

```
PAI -name string_similarity_topn
  -project sre_mpi_algo_dev
  -DinputTableName=pai_ft_string_similarity_topn_input
  -DmapTableName=pai_ft_string_similarity_topn_input
  -DoutputTableName=pai_ft_string_similarity_topn_output
  -DinputSelectedColName=col0
  -DmapSelectedColName=col0
  -DinputAppendColNames=col0
  -DinputAppendRenameColNames=input_col0
  -DmapAppendColNames=col0
  -DmapAppendRenameColNames=map_col0
  -Dmethod=simhash_hamming_sim
```

- 输出结果

input_col0	map_col0	output
北京	北京	1
北京	北京上海	0.90625
北京	北京上海香港	0.796875
北京上海	北京上海	1
北京上海	北京	0.90625
北京上海	北京上海香港	0.828125
北京上海香港	北京上海香港	1
北京上海香港	北京上海	0.828125
北京上海香港	北京	0.796875

3.7.6. 停用词过滤

本文为您介绍PAI-Studio提供的停用词过滤算法组件。

停用词过滤算法组件是文本分析中的一个预处理方法，用于过滤分词结果中的噪声（例如的、是或啊）。

停用词过滤算法组件的两个输入桩为输入表和停用词表。输入表为需要过滤停用词的表。停用词表的格式为一列，每行对应一个停用词。

PAI-Studio支持通过可视化或PAI命令方式，配置停用词过滤算法组件相关参数。

配置组件

- 可视化方式

页签	参数	描述
字段设置	**待过滤列**	指定待过滤的列，多列以英文逗号（,）分隔。
执行调优	**核心数**	默认为系统自动分配。
	内存数	默认为系统自动分配。

- PAI命令方式

```
PAI -name FilterNoise -project algo_public \
  -DinputTableName=”test_input” -DnoiseTableName=”noise_input” \
  -DoutputTableName=”test_output” \
  -DselectedColNames=”words_seg1,words_seg2” \
  -Dlifecycle=30
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入分词表的名称。	无
inputTablePartitions	否	输入分词表的分区名称。	所有分区
noiseTableName	是	停用词表的名称。	无
noiseTablePartitions	否	停用词表的分区名称。	所有分区
outputTableName	是	输出表的名称。	无
selectedColNames	是	指定待过滤的列，多列以英文逗号（,）分隔。	无
lifecycle	否	输出表的生命周期。取值范围为正整数。	无
coreNum	否	计算的核心数。	系统自动分配
memSizePerCore	否	每个核心的内存数。	系统自动分配

示例

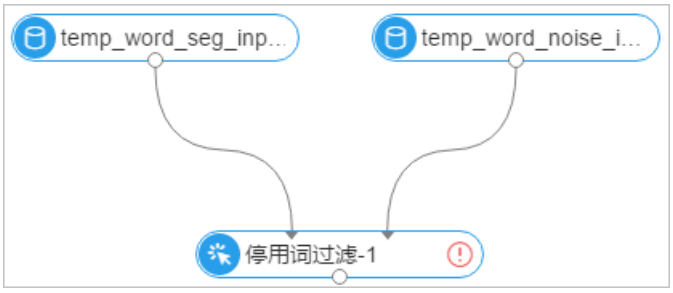
- 输入数据
  - 输入分词表temp_word_seg_input示例如下。

doc_idx	s_idx	seg
1	1	在 金融 危机 最 严重 时 , 在 股市 持续 暴跌 期间 , 2008年 10月 17日 在 《 时报 》 发表 文章 罕见 地 公开 宣称 “ 我 正在 买入 股票 ”
1	2	股市 暴跌 , 别人 都 恐惧 得 要 死 , 纷纷 抛售
1	3	的 公司 股票 投资 超过 600亿 美元 , 市值 也 是 大幅 下跌
1	4	他 不但 不 抛 反而 大量 买入 加上 收购 累计 投资 超过 500亿 美元
1	5	为什么 呢 ? 这 就是 近 50 年 来 长期 业绩 远远 战胜 市场 的 秘诀 : 反向 思考 , 反向 投资 , 而且 长期 坚持 如此
2	1	娱乐圈 有 很 多 明星 是 一夜 走红 , 比如 , 她 尚 是 大 二 学生 时 就 在 1987年 拍摄 的 《 》 而 一举 成为 国际 巨星
2	2	再 比如 很 多 明星 们 家境 很 好 , 出来 打拼 娱乐圈 无非 是 为了 兴趣 , 比如 tvb 的 , , 以及 内地 的
2	3	然而 娱乐圈 也 有 很 多 明星 她们 却 并非 是 一夜 成名 , 也 非 家境 殷实 , 而是 一 步步 的 打拼 到 现在

- 停用词表temp_word_noise_input示例如下。

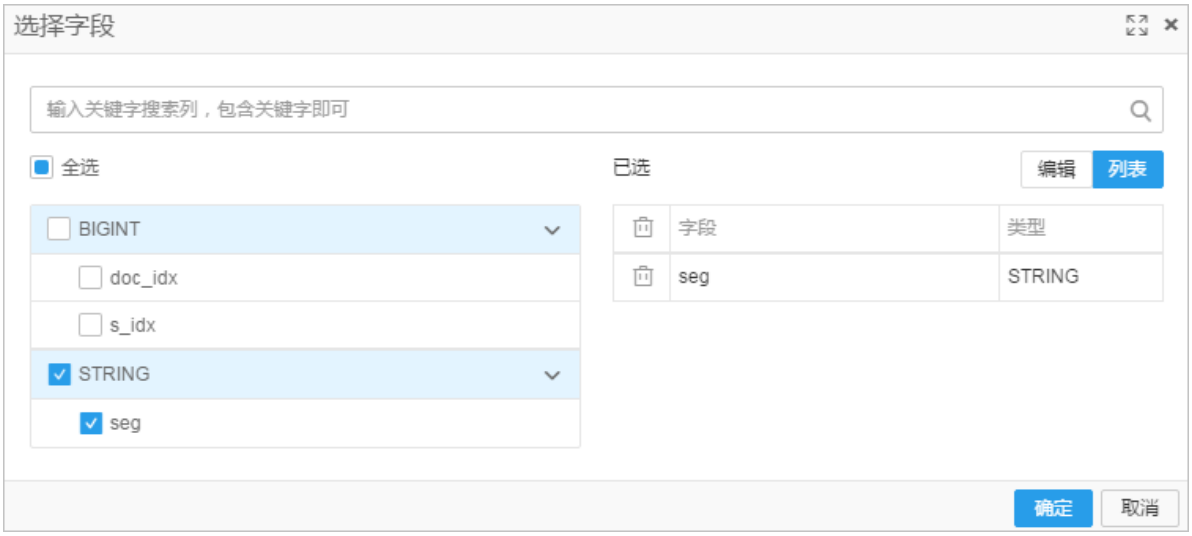
noise_word
在
地
时
他
是
比如
而是
的
《
》
“
”
，
、
？
也

- 创建实验



- 选择待过滤列

  选择seg字段为待过滤列。



- 运行结果

doc_idx	s_idx	seg
1	1	金融 危机 最 严重 股市 持续 暴跌 期间 2008年 10月 17日 时报 发表 文章 罕见 公开 宣称 我 正在 买入 股票
1	2	股市 暴跌 别人 都 恐惧 得 要 死 纷纷 抛售
1	3	公司 股票 投资 超过 600亿 美元 市值 大幅 下跌
1	4	不但 不 抛 反而 大量 买入 加上 收购 累计 投资 超过 500亿 美元
1	5	为什么 呢 ? 这 就是 近 50 年 来 长期 业绩 远 远 战胜 市场 秘诀 : 反向 思考 反向 投资 而且 长期 坚持 如此
2	1	娱乐圈 有 很 多 明星 一夜 走红 她 尚 大 二 学生 就 1987年 拍摄 而 一举 成为 国际 巨星
2	2	再 很 多 明星 们 家境 很 好 出来 打拼 娱乐圈 无非 为了 兴趣 tvb 以及 内地
2	3	然而 娱乐圈 有 很 多 明星 她们 却 并非 一夜 成名 非 家境 殷实 一 步步 打拼 到 现在

3.7.7. ngram-count

本文为您介绍PAI-Studio提供的ngram-count算法组件。

ngram-count是语言模型训练其中一个步骤。在词的基础上生成n-gram，并统计在全部语料集上，对应n-gram的个数。其结果是全局的个数，而不是单个文档的个数。详情请参见ngram-count。

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	输入表中句子所在的列	输入表中句子所在列字段。
	词袋中词所在的列名	选择词袋中词所在的列名。
	输入的count结果表的words列	输入的count结果表的words列。
	输入的count结果表的count列	输入的count结果表的count列。
	句子的权重列	输入句子的权重列。
参数设置	N-grams的最大长度	设置N-grams的最大长度，默认为3。
执行调优	可选，核心个数	默认自动选择。
	可选，单个核心使用的内存数	默认自动选择。

- PAI命令方式

```
PAI -name ngram_count
  -project algo_public
  -DinputTableName=pai_ngram_input
  -DoutputTableName=pai_ngram_output
  -DinputSelectedColNames=col0
  -DweightColName=weight
  -DcoreNum=2
  -DmemSizePerCore=1000;
```

参数名称	是否必选	默认值	描述
inputTableName	是	无	输入表
outputTableName	是	无	输出表
inputSelectedColNames	否	第一个字符类型的列	输入表选择列
weightColName	否	1	权重列名
inputTablePartitions	否	选择全表	输入表指定分区
countTableName	否	无	ngram-count以往的输出表，最终结果将合并这张表。
countWordColName	否	选择第二列	count表中词所在的列名
countCountColName	否	选择第三列	count表中count所在的列
countTablePartitions	否	无	count表指定分区
vocabTableName	否	无	词袋表，不在词袋中的词在结果中会被标识为\<unk\>。
vocabSelectedColName	否	选择第一个字符类型的列	词袋所在的列名
vocabTablePartitions	否	无	词袋表指定分区
order	否	3	N-grams的最大长度
lifecycle	否	无	输出表的生命周期
coreNum	否	无	核心个数
memSizePerCore	否	无	单个核心使用的内存数

3.7.8. 文本摘要

本文为您介绍PAI-Studio提供的文本摘要算法组件。

文本摘要是文献中简单连贯的短文，能够全面准确地反映该文献的中心思想。自动文摘利用计算机自动从原始文献中提取摘要内容。

本算法基于TextRank，通过提取文档中已存在的句子形成摘要。详情请参见TextRank: Bringing Order into Texts。

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	标识文章ID的列名	输入标识文档ID的列名。
	句子列	可指定一列。
参数设置	输出前的关键句个数	默认是3。
	句子相似度的计算方法	句子相似度计算方法： ○ lcs_sim ○ leveshtein_sim ○ ssk ○ cosine
	匹配字符串的权重	句子相似度的计算方法取值ssk时，该参数生效。默认为0.5。
	子串的长度	句子相似度的计算方法取值ssk/cosine时，该参数生效。默认为2。
	阻尼系数	默认为0.85。
	最大迭代次数	默认为100。
	收敛系数	默认为0.000001。
执行调优	核心数	默认自动分配。
	单个核心的内存	默认自动分配。

- PAI命令方式

```
PAI -name TextSummarization
  -project algo_public
  -DinputTableName="test_input"
  -DoutputTableName="test_output"
  -DdocIdCol="doc_id"
  -DsentenceCol="sentence"
  -DtopN=2
  -Dlifecycle=30;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表名	无
inputTablePartitions	否	输入表中指定参与计算的分区	输入表所有分区
outputTableName	是	输出表名	无

参数名称	是否必选	描述	默认值
docIdCol	是	标识文章ID的列名	无
sentenceCol	是	句子列，仅可指定一列。	无
topN	否	输出前几个关键句	3
similarityType	否	句子相似度计算方法： ○ lcs_sim ○ leveshtein_sim ○ ssk ○ cosine	lcs_sim
lambda	否	匹配字符串的权重，ssk中可用。	0.5
k	否	子串的长度，ssk和cosine中可用。	2
dampingFactor	否	阻尼系数	0.85
maxIter	否	最大迭代次数	100
epsilon	否	收敛系数	0.000001
lifecycle	否	输入出表的生命周期	无
coreNum	否	参与计算的核心数	系统自动分配
memSizePerCore	否	每个核心需要的内存	系统自动分配

示例

输出表为两列，分别是doc_id和abstract。

doc_id	abstract
1000894	早在2008年，上交所便发布了上市公司社会责任披露相关指引，强制要求三类公司披露社会责任报告，同时鼓励其他有条件的上市公司进行自愿披露。统计显示，2012年，沪市上市公司共计379家披露社会责任报告，包括强制披露公司305家和自愿披露公司74家，合计占沪市全部上市公司的40%。胡汝银表示，下一步上交所将探索扩大社会责任报告的披露范围，修订细化有关社会责任报告披露的指引，并鼓励更多的机构推进社会责任产品创新。

3.7.9. 关键词抽取

本文为您介绍PAI-Studio提供的关键词抽取算法组件。

关键词抽取是自然语言处理中的重要技术之一，具体是指从文本中将与这篇文章意义相关性最强的一些词抽取出来。该算法基于TextRank，根据PageRank算法思想，利用局部词汇之间关系（共现窗口）构建网络，并计算单词的重要性，最终选取权重大的作为关键词。

常用流程如下：

1. 原始语料
2. 分词
3. 使用词过滤
4. 关键词抽取

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	**标识文章id的列名**	输入标识文章ID的列名。
	标识文章内容分完词结果	输入标识文章内容分完词结果名称。
参数设置	**输出前多少个关键词**	整数，默认值为5。
	窗口大小	整数，默认值为2。
	阻尼系数	默认值为0.85。
	最大迭代数	默认值为100。
	收敛系数	默认值为0.000001。
执行调优	**核心数，默认自动分配**	默认自动选择。
	每个核心的内存，默认自动分配	默认自动选择。

- PAI命令方式

```
PAI -name KeywordsExtraction
  -DinputTableName=maple_test_keywords_basic_input
  -DdocIdCol=docid -DdocContent=word
  -DoutputTableName=maple_test_keywords_basic_output
  -DtopN=19;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表。	无

参数名称	是否必选	描述	默认值
inputTablePartitions	否	输入表中指定哪些分区参与训练，格式为“Partition_name=value”。如果是多级格式为“name1=value1/name2=value2”。如果指定多个分区，中间用半角逗号（,）分隔。	选择所有分区
outputTableName	是	输出表名。	无
docIdCol	是	标识文章ID的列名，仅可指定一列。	无
docContent	是	Word列，仅可指定一列。	无
topN	否	输出前多少个关键词，当关键词个数小于全部词个数时，全部输出。	5
windowSize	否	TextRank算法的窗口大小。	2
dumpingFactor	否	TextRank算法的阻尼系数。	0.85
maxIter	否	TextRank算法的最大迭代次数。	100
epsilon	否	TextRank算法的收敛残差阈值。	0.000001
lifecycle	否	指定输出表的生命周期。	无
coreNum	否	节点个数。	自动计算
memSizePerCore	否	单个节点内存大小，单位为MB。	自动计算

示例

1. 数据生成

   输入表需采用空格分词，并过滤掉停用词（如“的”、“地”、“得”、“了”、“个”）和所有标点符号。

docid:string	word:string

docid:string	word:string
doc0	翼身融合 飞机 是 未来 航空 领域 发展 一个 新 方向 国内外 诸多 研究 机构 已经 开展 对翼身融合 飞机 研究 而 其 全自动 外形 优化 算法 已 成为 新 研究 热点 国内 外 现有 成果 基础 之上 分析 比较 常用 建模 求解 平台 使用 方式 及 特点 设计 编写 翼身融合 飞机 外形 优化 几何 建模 网格 划分 流场 求解 外形 优化 模块 比较 不同 算法 间 优劣 实现 翼身融合 飞机 概念设计 中 外形 优化 几何 建模 及 网格 生成 模块 实现 基于 超限 插值 网格 生成 算法 基于 样条 曲线 建模 方法 流场 求解 模块 包括 有限 差分 求解器 有限元 求解器和面元法 求解器 其中 有限 差分 求解器 主要 包括 基于 有限 差分法 势流 数学 建模 基于 笛卡尔 网格 变 步长 差分格式 推导 笛卡尔 网格 生成 索引 算法 基于 笛卡尔 网格 诺 依曼 边界条件 表达 形式 推导 实现 基于 有限 差分 求解器 二维 翼型 气动 参数 计算 算例 有限元 求解器 主要 包括 基于 变分 原理 势流 有限元 理论 建模 二维 有限元 库塔 条件 表达式 推导 基于 最小 二乘 速度 求解 算法 设计 基于 Gmsh 二维 带尾迹 翼型 空间 网格 生成器 开发 实现 基于 有限元 求解器 二维 翼型 气动 参数 计算 算例 面元法 求解器 主要 包括 基于 面元法 势流 理论 建模 自动 尾迹 生成 算法 设计 基于 面元法 三维 翼身融合 体 流场 求解器 开发 基于 布拉 修斯 平板 解 阻力 估算 算法 设计 求解器 Fortran 语言 上移 植 Python 和 Fortran 代码 混编 基于 OpenMP 和 CUDA 并行 加速 算法 设计 与 开发 实现 基于 面元法 求解器 三维 翼身融合 体 气动 参数 计算 算例 外形 优化 模块 实 现了 基于 自由 形状 变形 网格 变形 算法 遗传算法 差分 进化 算法 飞机 表面积 计算 算法 基于 矩 积分 飞 机 体积 计算 算法 开发 基于 VTK 数据 可视化 格式 工具

2. PAI命令

```
PAI -name KeywordsExtraction
  -DinputTableName=maple_test_keywords_basic_input
  -DdocIdCol=docid -DdocContent=word
  -DoutputTableName=maple_test_keywords_basic_output
  -DtopN=19;
```

3. 输出说明

docid	keywords	weight
doc0	基于	0.041306752223538405
doc0	算法	0.03089845626854151
doc0	建模	0.021782865850562882
doc0	网格	0.020669749212693957
doc0	求解器	0.020245609506360847

docid	keywords	weight
doc0	飞机	0.019850761705313365
doc0	研究	0.014193732541852615
doc0	有限元	0.013831122054200538
doc0	求解	0.012924593244133104
doc0	模块	0.01280216562287212
doc0	推导	0.011907588923852495
doc0	外形	0.011505456605632607
doc0	差分	0.011477831662367547
doc0	势流	0.010969269350293957
doc0	设计	0.010830986516637251
doc0	实现	0.010747536556701583
doc0	二维	0.010695570768457084
doc0	开发	0.010527342662670088
doc0	新	0.010096978306668461

3.7.10. 句子拆分

本文为您介绍PAI-Studio提供的句子拆分算法组件。

将一段文本按标点进行句子拆分。该组件主要用于文本摘要前的预处理，将一段文本拆分成一句一行的形式。

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	**标识文章ID的列名**	输入标识文章ID的列名
	标示文章内容的列名	输入标示文章内容的列名
	句子的间隔字符集合	默认“。！？”
执行调优	**核心数**	默认自动分配
	每个核心的内容	默认自动分配

- PAI命令方式

```
PAI -name SplitSentences
  -project algo_public
  -DinputTableName="test_input"
  -DoutputTableName="test_output"
  -DdocIdCol="doc_id"
  -DdocContent="content"
  -Dlifecycle=30
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表名	无
inputTablePartitions	否	输入表中指定参与计算的分区	输入表的所有分区
outputTableName	是	输出表名	无
docIdCol	是	标识文章ID的列名	无
docContent	是	标示文章内容的列名，仅可指定一列。	无
delimiter	否	句子的间隔字符集合	“。！？”
lifecycle	否	输入出表的生命周期	无
coreNum	否	参与计算的核心数	系统自动计算
memSizePerCore	否	每个核心需要的内存	系统自动计算

示例

输出表为两列，分别是doc_id和sentence。

doc_id	sentence
1000894	早在2008年，上交所便发布了上市公司社会责任披露相关指引，强制要求三类公司披露社会责任报告，同时鼓励其他有条件的上市公司进行自愿披露。
1000894	统计显示，2012年，沪市上市公司共计379家披露社会责任报告，包括强制披露公司305家和自愿披露公司74家，合计占沪市全部上市公司的40%。

3.7.11. 语义向量距离

本文为您介绍PAI-Studio提供的语义向量距离算法组件。

基于算法语义向量结果（如Word2Vec生成的词向量），计算给定的词（或者句子）的扩展词（或者扩展句），即计算其中某一向量距离最近的向量集合。其中一个用法是，基于Word2Vec生成的词向量结果，根据输入的词返回最为相似的词列表。

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	**id所在列名**	输入其唯一标识列。
	向量的列名列表	如f1，f2。
参数设置	**输出的距离最近的向量的数目**	默认值为5。
	距离的计算方式	支持如下计算方式： ○ euclidean ○ cosine ○ manhattan 默认值为euclidean。
	距离的阈值	当两个向量的距离小于此值时输出，默认值为+∞。
执行调优	**计算的核心数**	默认自动分配。
	每个核心的内存（MB）	默认自动分配。

- PAI命令方式

```
PAI -name SemanticVectorDistance
  -project algo_public
  -DinputTableName="test_input"
  -DoutputTableName="test_output"
  -DidColName="word"
  -DvectorColNames="f0,f1,f2,f3,f4,f5"
  -Dlifecycle=30
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表名	无
inputTablePartitions	否	输入表中指定参与计算的分区	输入表的所有分区
outputTableName	是	输出表名	无
idTableName	否	需要计算相近向量的ID的列表所在表名。格式为一列，每一行一个ID。默认为空，即输入表中的所有向量参与计算。	无

参数名称	是否必选	描述	默认值
idTablePartitions	否	ID表中参与计算的分区列表，默认为所有分区。	无
idColName	是	ID所在列名	3
vectorColNames	否	向量的列名列表，如f1，f2。	无
topN	否	输出的距离最近的向量的数目。取值范围[1,+∞]。	5
distanceType	否	距离的计算方式	euclidean
distanceThreshold	否	距离的阈值。当两个向量的距离小于此值时输出。取值范围(0,+∞)。	+∞
lifecycle	否	输入出表的生命周期，取值为正整数。	无
coreNum	否	参与计算的核心数，取值为正整数。	系统自动计算
memSizePerCore	否	每个核心需要的内存，取值为正整数。	系统自动计算

示例

输出表为四列，分别是original_id、near_id、distance、rank。

original_id	near_id	distance	rank
hello	hi	0.2	1
hello	xxx	xx	2
Man	Woman	0.3	1
Man	xx	xx	2

3.7.12. 词频统计

词频统计是指输入一些字符串（手动输入或者从指定的文件读取），用程序来统计这些字符串中总共有多少个单词，每个单词出现的次数。单词的总数（即为Total）为不重复的单词数总和。本文为您介绍PAI-Studio提供的词频统计。

词频指词的频率，即词在一定的语料中出现的次数。请在对文档进行分词的基础上，按行保序输出对应文档ID列（docId）对应的词，统计指定文档ID列对应文档内容（docContent）的词频。

配置组件

您可以通过以下任意一种方式，配置词频统计参数：

- 可视化方式

页签	参数	描述
字段设置	选择文档ID列	选择文档ID列。
	选择文档内容列	选择文档内容列。
执行调优	核心数	节点数量。
	每个核心的内存	单个节点内存大小，单位为MB。

- PAI命令方式

```
pai -name doc_word_stat
  -project algo_public
  -DinputTableName=tdl_doc_test_split_word
  -DdocId=docid
  -DdocContent=content
  -DoutputTableNameMulti=doc_test_stat_multi
  -DoutputTableNameTriple=doc_test_stat_triple
  -Dlifecycle=7
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表名称。	无
docId	是	标识文档ID的列名，仅可指定一列。	无
docContent	是	标识文档内容的列名，仅可指定一列。	无
outputTableNameMulti	是	输出保序词语表名。	无
outputTableNameTriple	否	输出词频统计表名。	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区之间使用英文逗号（,）分隔。	选择所有分区
lifecycle	否	输出表生命周期。正整数。	-1

示例

采用阿里分词实例数据中，将分别将输出表的两个列作为词频统计的输入参数：

- 选择文档ID列：ID。
- 选择文档内容列：text经过词频统计运算后，生成的结果见本组件中第一个输出参数展示图。

PAI命令行

```
pai -name doc_word_stat
  -project algo_public
  -DinputTableName=tdl_doc_test_split_word
  -DdocId=id
  -DdocContent=content
  -DoutputTableNameMulti=doc_test_stat_multi
  -DoutputTableNameTriple=doc_test_stat_triple
  -DinputTablePartitions="region=cctv_news"
  -Dlifecycle=7
```

输入参数：经过分词组件生成两列—文档ID列和分词后的文档内容列。

两个输出参数：

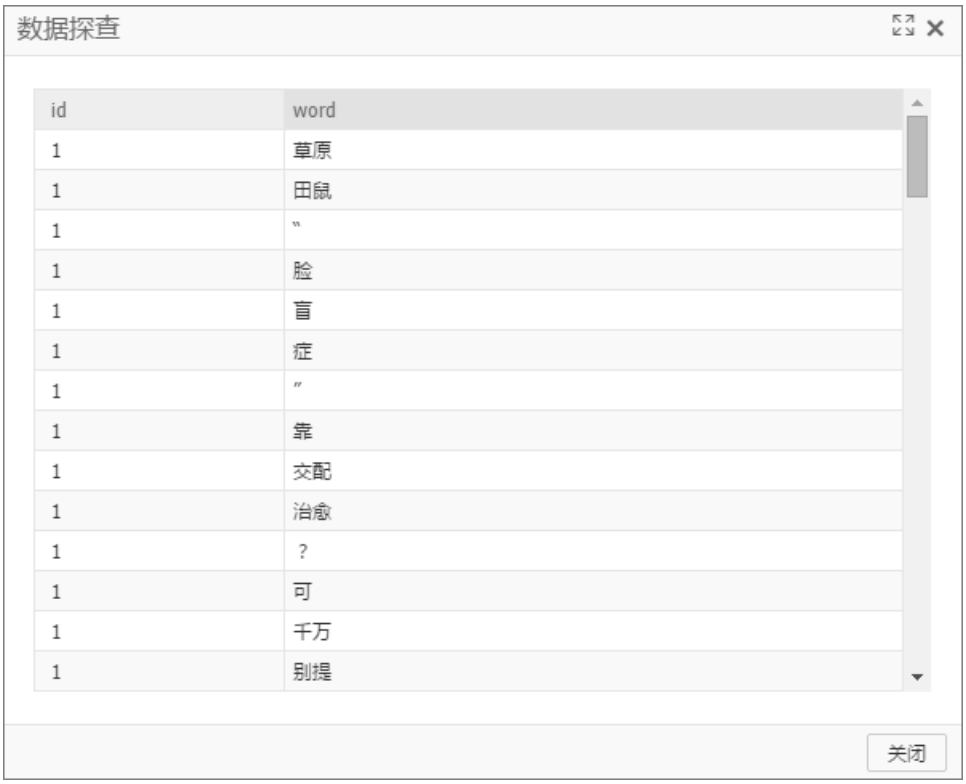
- 第一个输出端：输出表包含id、word和count三个字段。

id	word	count
1	"	1
1	”	1
1	。	2
1	不	1
1	不知道	1
1	不过	1
1	之前	1
1	之后	1
1	了	1
1	交配	3
1	什么	1
1	会	1
1	伴侣	2
1	分	1

参数	描述
id	文档ID列。
word	单词列。
count	统计每个文档中，对应word词汇出现的次数。

> **说明** 该输出表的列可以分别作为TF-IDF组件的输入。

- 第二个输出端：输出包含id和word两个字段。



本端口输出表按词语在文档中出现的顺序依次输出，没有统计词语的出现次数，因此同一文档中某个词汇可能出现多条记录。 包输出表格式主要用于兼容Word2Vec组件。

常见问题

- 参数outputTableNameMulti指定的表是docId列及docId列对应的文档内容（docContent）完成分词后，按各个词语在文档中出现的顺序依次输出。
- 参数outputTableNameTriple指定的表输出docId列及docId列对应的文档内容（docContent。

3.7.13. TF-IDF

TF-IDF（Term Frequency-Inverse Document Frequency）是一种用于资讯检索与文本挖掘的常用加权技术。通常在搜索引擎中应用，可以作为文件与用户查询之间相关程度的度量或评级。

TF词频（Term Frequency）是指某一个给定的词语在该文件中出现的次数。IDF反文档频率（Inverse Document Frequency）是指如果包含词条的文档越少，IDF越大，则说明词条的类别区分能力越强。

TF-IDF是一种统计方法，用于评估字词或文件的重要程度。例如：

- 在文件集中的字词会随着出现次数的增加呈正比增加趋势。
- 在语料库中的文件会随着出现频率的增加呈反比下降趋势。

TF-IDF组件基于**词频统计**算法的输出结果（而不是基于原始文档），计算各词语对于各文章的TF-IDF值。

配置组件

您可以通过以下任意一种方式，配置TF-IDF参数：

- 可视化方式

页签	参数	描述
字段设置	选择文档ID列	您可以直接选择**词频统计**组件输出的文档ID列（**id**列）或自行将原始文档处理为相应格式，详情请参见词频统计示例部分的输出介绍。
	选择单词列	您可以直接选择**词频统计**组件输出的单词列（**word**列）或自行将原始文档处理为相应格式，详情请参见词频统计示例部分的输出介绍。
	选择单词计数列	您可以直接选择**词频统计**组件输出的单词计数列（**count**列）或自行将原始文档处理为相应格式，详情请参见词频统计示例部分的输出介绍。
执行调优	计算核心数	节点个数，默认自动计算。
	每个核心内存	单个节点内存大小，单位为MB。

- PAI命令方式

```
PAI -name tfidf
  -project algo_public
  -DinputTableName=rgdoc_split_triple_out
  -DdocIdCol=id
  -DwordCol=word
  -DcountCol=count
  -DoutputTableName=rg_tfidf_out;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表名称。	无
inputTablePartitions	否	输入表中，参与训练的分区。 格式为 `partition name=value` 。如果是多级格式为 `name1=value1/name2=value2` 。如果是指定多个分区，中间用英文逗号分开。	输入表的所有分区
docIdCol	是	标识文章ID的列名，仅可指定一列。	无
wordCol	是	Word列名，仅可指定一列。	无
countCol	是	Count列名，仅可指定一列。	无
outputTableName	是	输出表名称。	无
lifecycle	否	输出表生命周期。正整数。单位：天	无
coreNum	否	核心数，与memSizePerCore同时设置才生效。	自动计算

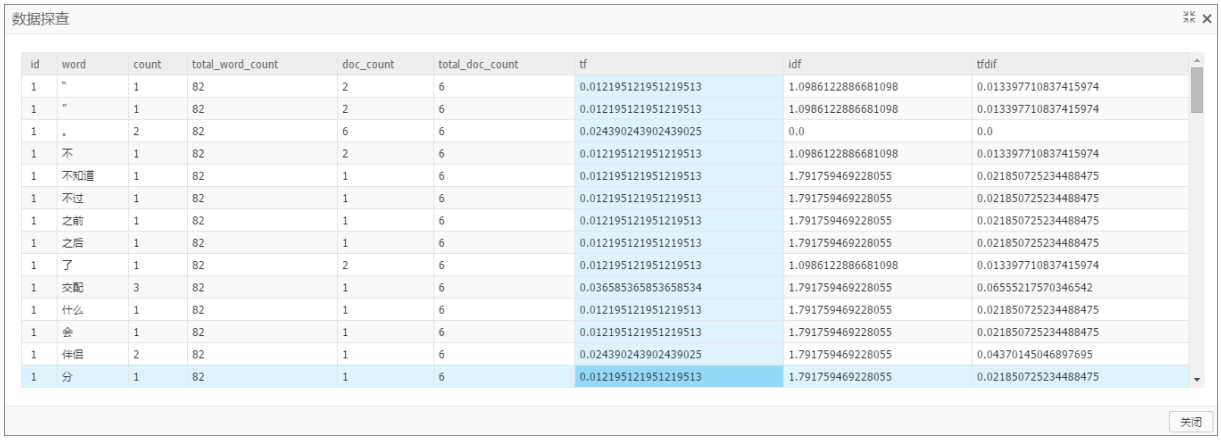
参数名称	是否必选	描述	默认值
memSizePerCore	否	内存数，与coreNum同时设置才生效。	自动计算

示例

以TF-IDF组件实例中的输出表作为TF-IDF组件的输入表，对应的参数设置如下：

- 选择文档ID列： id
- 选择单词列：word
- 选择单词计数列：count

输出表有9列：**docid**、**word**、**word_count**（当前word在当前doc中出现次数）、**total_word_count**（当前doc中总word数）、**doc_count**（当前word的总doc数）、**total_doc_count**（全部doc数）、**tf**、**idf**和**tfidf**。



id	word	count	total_word_count	doc_count	total_doc_count	tf	idf	tfdif
1	“	1	82	2	6	0.012195121951219513	1.0986122886681098	0.013397710837415974
1	”	1	82	2	6	0.012195121951219513	1.0986122886681098	0.013397710837415974
1	。	2	82	6	6	0.024390243902439025	0.0	0.0
1	不	1	82	2	6	0.012195121951219513	1.0986122886681098	0.013397710837415974
1	不知道	1	82	1	6	0.012195121951219513	1.791759469228055	0.021850725234488475
1	不过	1	82	1	6	0.012195121951219513	1.791759469228055	0.021850725234488475
1	之前	1	82	1	6	0.012195121951219513	1.791759469228055	0.021850725234488475
1	之后	1	82	1	6	0.012195121951219513	1.791759469228055	0.021850725234488475
1	了	1	82	2	6	0.012195121951219513	1.0986122886681098	0.013397710837415974
1	交配	3	82	1	6	0.036585365853658534	1.791759469228055	0.06555217570346542
1	什么	1	82	1	6	0.012195121951219513	1.791759469228055	0.021850725234488475
1	会	1	82	1	6	0.012195121951219513	1.791759469228055	0.021850725234488475
1	伴侣	2	82	1	6	0.024390243902439025	1.791759469228055	0.04370145046897695
1	分	1	82	1	6	0.012195121951219513	1.791759469228055	0.021850725234488475

3.7.14. PLDA

主题模型用于在一系列文档中发现抽象主题（topic）的一种统计模型，在机器学习PAI平台，您可以通过给PLDA组件设置topic参数值，从而让每篇文档抽象出不同主题。

LDA（Latent Dirichlet allocation）是一种主题模型，它可以按照概率分布的形式给出文档集中每篇文档的主题。LDA也是一种无监督学习算法，在训练时您无需手工标注的训练集，仅需要在文档集中指定主题的数量K即可（K即为PLDA参数topic）。

LDA首先由David M. Blei、Andrew Y. Ng和Michael I. Jordan于2003年提出，在文本挖掘领域应用于文本主题识别、文本分类和文本相似度计算等方面。

配置组件

您可以通过以下任意一种方式，配置PLDA参数：

- 可视化方式

参数说明

页签	参数	参数描述
字段设置	选择特征列	选择参与训练的特征列。
	主题个数	设置LDA的输出的主题个数。

页签	参数	参数描述
参数设置	Alpha	P(z/d) 的先验狄利克雷分布的参数。
	beta	P(w/z) 的先验狄利克雷分布的参数。
	burn In 迭代次数	Burn In 迭代次数，必须小于总迭代次数，默认值为100。
	总迭代次数	正整数，非必选，默认值为150。

- PAI命令方式

```
pai -name PLDA
  -project algo_public
  -DinputTableName=lda_input
  –DtopicNum=10
  -topicWordTableName=lda_output;
```

参数名称	是否必选	描述	类型	默认值
inputTableName	是	输入表的名称。	STRING	无
inputTablePartitions	否	输入表中，参与训练的分区。系统支持以下格式： ○ Partition_name=value ○ name1=value1/name2=value2：多级分区 **说明** 指定多个分区时，分区之间使用英文逗号（,）分隔。	STRING	输入表的所有分区
selectedColNames	否	输入表中用于LDA的列名。	STRING	输入表中所有的列名
topicNum	是	Topic的数量，取值范围为[2, 500]。	正整数	无
kvDelimiter	否	Key和Value间的分隔符。取值： ○ 空格 ○ 英文逗号（,） ○ 英文冒号（:）	STRING	英文冒号（:）
itemDelimiter	否	Key和Key间的分隔符。取值： ○ 空格 ○ 英文逗号（,） ○ 英文冒号（:）	STRING	空格

参数名称	是否必选	描述	类型	默认值
alpha	否	P(z/d) 的先验狄利克雷分布的参数。取值为(0, ∞)。	FLOAT	0.1
beta	否	P(w/z) 的先验狄利克雷分布的参数。取值为(0, ∞)。	FLOAT	0.01
topicWordTableName	是	topic-word频率贡献表。	STRING	无
pwzTableName	否	P(w/z) 输出表。	STRING	无，即不输出 P(w/z) 表
pzwTableName	否	P(z/w) 输出表。	STRING	无，即不输出 P(z/w) 表
pdzTableName	否	P(d/z) 输出表。	STRING	无，即不输出 P(d/z) 表
pzdTableName	否	P(z/d) 输出表。	STRING	无，即不输出 P(z/d) 表
pzTableName	否	P(z) 输出表。	STRING	无，即不输出 P(z) 表
burnInIterations	否	Burn In迭代次数，且取值必须小于totalIterations。	正整数	100
totalIterations	否	迭代次数。 **说明** z是主题，w是词，d是文档。	正整数	150
enableSparse	否	是否是kv输入，选择kv输入或分词结果。取值范围如下： ○ true：kv输入 ○ false：非kv输入	BOOL	true
coreNum	否	与参数memSizePerCore配对使用，默认系统会根据输入数据量计算所起Instance的数量，即取值为-1。	正整数	-1
memSizePerCore	否	单个节点内存大小，单位MB。范围为[1024, 64*1024]。默认系统自动计算每个节点的内存大小，即取值为-1。	正整数	-1

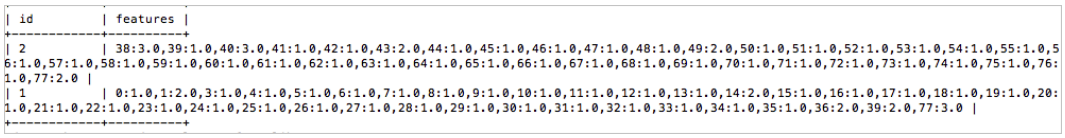
输入和输出设置

- 输入

数据必须为稀疏矩阵的格式。可以通过三元组转KV组件进行转换。

输入格式如输入格式所示。

输入格式



- 第一列：docid。
- 第二列：单词及词频的kv数据。

- 输出：

  输出依次为：topic-word频率贡献表、单词|主题输出表、主题|单词输出表、文档|主题输出表、主题|文档输出表、主题输出表。

  topic-word频率贡献表的输出格式如输出格式所示。

  输出格式

wordid	topic_0	topic_1
0	1	0
1	2	0
2	0	0
3	1	0
4	1	0
5	1	0
6	1	0
7	1	0
8	0	1
9	1	0
10	1	0
11	1	0

3.7.15. Split Word（生成模型）

本文为您介绍PAI-Studio提供的Split Word（生成模型）算法组件。

Split Word（生成模型）算法组件基于AliWS（Alibaba Word Segmenter）词法分析系统，根据参数和自定义词典生成分词模型。

Split Word（生成模型）算法组件仅支持中文淘宝分词和互联网分词。

与Split Word的区别：

- Split Word是直接将输入的文本分词。
- Split Word（生成模型）用于生成分词的模型。如果您需要对文本分词，您需要先部署模型，再进行预测或调用在线API。

配置组件

- 可视化方式

页签	参数	描述
字段设置	**选择字段列**	用来生成模型的字段列。

页签	参数	描述
参数设置	识别选项	识别内容类型。取值范围为： ○ 识别简单实体 ○ 识别人名 ○ 识别机构名 ○ 识别电话号码 ○ 识别时间 ○ 识别日期 ○ 识别数字字母 默认值为：识别简单实体、识别电话号码、识别时间、识别日期和识别数字字母。
	合并选项	合并内容类型。取值范围为： ○ 合并中文数字 ○ 合并阿拉伯数字 ○ 合并中文日期 ○ 合并中文时间 默认值为：合并阿拉伯数字。
	过滤器	过滤器的类型。取值范围为TAOBAO_CHN和INTERNET_CHN。默认值为TAOBAO_CHN。
	Pos Tagger	是否进行词性标注。默认不进行词性标注。
	Semantic Tagger	是否进行语义标注。默认不进行语义标注。
	过滤分词结果为数字的词	是否过滤分词结果为数字的词。默认不过滤。
	过滤分词结果为全英文的词	是否过滤分词结果为全英文的词。默认不过滤。
	过滤分词结果为标点符号的词	是否过滤分词结果为标点符号的词。默认不过滤。
执行调优	核心数	默认为系统自动分配。
	每个核的内存数	默认为系统自动分配。

- PAI命令方式

```
pai -name split_word_model
  -project algo_public
  -DoutputModelName=aliws_model
  -DcolName=content
  -Dtokenizer=TAOBAO_CHN
  -DenableDfa=true
  -DenablePersonNameTagger=false
  -DenableOrgnizationTagger=false
  -DenablePosTagger=false
  -DenableTelephoneRetrievalUnit=true
  -DenableTimeRetrievalUnit=true
  -DenableDateRetrievalUnit=true
  -DenableNumberLetterRetrievalUnit=true
  -DenableChnNumMerge=false
  -DenableNumMerge=true
  -DenableChnTimeMerge=false
  -DenableChnDateMerge=false
  -DenableSemanticTagger=true
```

参数名称	是否必选	描述	默认值
userDictTableName	否	是否使用自定义词典表。自定义词典表只有一列，每一行是一个词。	无
outputModelName	是	输出模型的名称。	无
colName	否	预测文本的列名。	context
dictTableName	否	是否使用自定义词典表。自定义词典表只有一列，每一行是一个词。	无
tokenizer	否	过滤器类型。取值范围为TAOBAO_CHN和INTERNET_CHN。	TAOBAO_CHN
enableDfa	否	是否识别简单实体。取值为True或False。	True
enablePersonNameTagger	否	是否识别人名。取值为True或False。	False
enableOrgnizationTagger	否	是否识别机构名。取值为True或False。	False
enablePosTagger	否	是否进行词性标注。取值为True或False。	False
enableTelephoneRetrievalUnit	否	是否识别电话号码。取值为True或False。	True
enableTimeRetrievalUnit	否	是否识别时间号码。取值为True或False。	True
enableDateRetrievalUnit	否	是否识别日期号码。取值为True或False。	True
enableNumberLetterRetrievalUnit	否	是否识别数字字母。取值为True或False。	True

参数名称	是否必选	描述	默认值
enableChnNumMerge	否	是否将中文数字合并为一个检索单元。取值为True或False。	False
enableNumMerge	否	是否将普通数字合并为一个检索单元。取值为True或False。	True
enableChnTimeMerge	否	是否将中文时间合并为一个语义单元。取值为True或False。	False
enableChnDateMerge	否	是否将中文日期合并为一个语义单元。取值为True或False。	False
enableSemanticTagger	否	是否进行语义标注。取值为True或False	False

示例

- PAI命令行

```
pai -name split_word_model
  -project algo_public
  -DoutputModelName=aliws_model
```

- 部署

```
create onlinemodel ning_test_aliws_model_2 -offlinemodelName ning_test_aliws_model -instanceNum 1
-cpu 100 -memory 4096;
```

- 在线分词

```
KVJsonRequest request = new KVJsonRequest();
Map<String, JsonFeatureValue> row = request.addRow();
row.put(col_name, new JsonFeatureValue("大数据算法平台是个新的平台"));
KVJsonResponse res = predictClient.syncPredict(new JsonPredictRequest(project_name, model_name, r
equest));
List<ResponseItem> ri = res.getOutputs();
for (ResponseItem item : ri) {
    System.out.println(item.getOutputLabel());
}
```

- 离线分词

```
pai -name prediction
  -DmodelName=ning_test_aliws_model
  -DinputTableName=ning_test_aliws
  -DoutputTableName=ning_test_aliws_offline_predict;
```

3.7.16. Word2Vec

本文为您介绍PAI-Studio提供的Word2Vec算法组件。

Word2Vec算法组件利用神经网络，通过训练，将词映射为K维度空间向量，且支持对表示词的向量进行操作并和语义相对应。输入为单词列或词汇表，输出为词向量表和词汇表。

配置组件

- 可视化方式

页签	参数	描述
字段设置	**选择单词列**	用来进行训练的单词列。
参数设置	**单词特征维度**	单词的特征维度数量。取值范围为0~1000，默认值为100。
	语言模型	训练使用的语言模型。取值范围为**skip-gram模型**和**cbow模型**，默认值为**skip-gram模型**。
	单词窗口大小	单词的窗口大小。取值范围为正整数，默认值为5。
	使用随机窗口	是否使用随机窗口。默认使用。
	截断最小词频	取值范围为正整数，默认值为5。
	采用hierarchical softmax	是否采用HIERARCHICAL SOFTMAX。默认采用。
	负采样	负采样的窗口大小。默认值为0，表示不可用。
	向下采样阈值	向下采样的阈值。默认值为0，表示不可用。
	起始学习速率	取值大于0，默认值为0.025。
	迭代次数	取值大于等于1，默认值为1。
执行调优	**核心数**	默认为系统自动分配。
	每个核的内存大小	默认为系统自动分配。

- PAI命令方式

```
pai -name Word2Vec
  -project algo_public
  -DinputTableName=w2v_input
  -DwordColName=word
  -DoutputTableName=w2v_output;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入词汇表的名称。	无
inputTablePartitions	否	输入词汇表中参与分词的分区名称。格式为 `partition name=value` 。多级分区格式为 `name1=value1/name2=value2` 。如果指定多个分区，用英文逗号（,）分隔。	无
wordColName	是	单词列名。单词列中每行为一个单词，换行符用</s>表示。	无

参数名称	是否必选	描述	默认值
inVocabularyTableName	否	对输入词汇表执行wordcount操作后的输出结果。	系统会对输出表执行wordcount操作
inVocabularyPartitions	否	对输入词汇表执行wordcount操作后的输出结果中的分区名称。	inVocabularyTableName对应表的所有分区
layerSize	否	单词的特征维度数量。取值范围为0~1000。	100
cbow	否	训练使用的语言模型。取值范围为0（skip-gram模型）和1（cbow模型）。	0
window	否	单词的窗口大小。取值范围为正整数。	5
minCount	否	截断的最小词频。取值范围为正整数。	5
hs	否	是否采用HIERARCHICAL SOFTMAX。取值范围为0（不采用）和1（采用）。	1
negative	否	负采样的窗口大小。取值范围为正整数，0表示不可用。	0
sample	否	向下采样的阈值。取值范围为1e-3～1e-5。0表示不可用。	0
alpha	否	取值大于0。	0.025
iterTrain	否	取值大于等于1。	1
randomWindow	否	指定单词窗口的展示方式。取值范围为0（不随机，其值由window参数指定）和1（大小在1～5间随机）。	1
outVocabularyTableName	否	输出词汇表的名称。	无
outputTableName	是	输出词向量表的名称。	无
lifecycle	否	输出表的生命周期。取值范围为正整数。	无
coreNum	否	核心数，需要与memSizePerCore参数同时设置才生效。取值范围为正整数。	系统自动分配
memSizePerCore	否	内存数，需要与coreNum参数同时设置才生效。取值范围为正整数。	系统自动分配

示例

- 输入词汇表如下。

> **说明** word2vec只能接收词频统计的右输出节点作为输入。

- 输出结果

第一列为对应的词输出，第二列之后为对应的embedding结果，维度由layerSize参数指定。

常见问题

报错“Vocab size is zero! vocab_size: 0”，表示词典为空，调小minCount参数值，即可解决。

3.7.17. Doc2Vec

本文为您介绍PAI-Studio提供的Doc2Vec算法组件。

您可以通过Doc2Vec算法组件将文章映射为向量。输入为词汇表，输出为文档向量表、词向量表或词汇表。

PAI-Studio支持通过可视化或PAI命令方式，配置Doc2Vec算法组件相关参数。

可视化方式

页签	参数	描述
IO/字段设置	**文档ID列名**	用来进行训练的文档列名。
	文档内容	用来进行训练的词汇，以空格分隔。
	单词的特征维度	单词的特征维度数量。取值范围为0~1000，默认值为100。

页签	参数	描述
参数设置	语言模型	训练使用的语言模型。取值范围为**skip-gram模型**和**cbow模型**，默认值为**skip-gram模型**。
	单词窗口大小	单词的窗口大小。取值范围为正整数，默认值为5。
	截断的最小词频	取值范围为正整数，默认值为5。
	Hierarchical Softmax	是否采用HIERARCHICAL SOFTMAX。默认采用。
	Negative Sampling	负采样的窗口大小。取值范围为正整数，默认值为5，0表示不可用。
	向下采样阈值	向下采样的阈值。取值范围为1e-3～1e-5，默认值为1e-3，0表示不可用。
	开始学习速率	取值大于0，默认值为0.025。
	训练的迭代次数	取值大于等于1，默认值为1。
	Window是否随机	指定单词窗口的展示方式。取值范围为**大小在1～5间随机**和**不随机，其值由window参数指定**，默认值为**不随机，其值由window参数指定**。
执行调优	计算的核心数	默认为系统自动分配。
	每个核心的内存（MB）	默认为系统自动分配。

PAI命令方式

```
PAI -name pai_doc2vec
  -project algo_public
  -DinputTableName=d2v_input
  -DdocIdColName=docid
  -DdocColName=text_seg
  -DoutputWordTableName=d2v_word_output
  -DoutputDocTableName=d2v_doc_output;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入词汇表的名称。	无

参数名称	是否必选	描述	默认值
inputTablePartitions	否	输入词汇表中参与分词的分区名称。格式为 partition_name=value 。多级分区格式为 name1=value1/name2=value2 。如果指定多个分区，用英文逗号（,）分隔。	无
docIdColName	是	用来进行训练的文档列名。	无
docColName	是	用来进行训练的词汇，以空格分隔。	无
layerSize	否	单词的特征维度数量。取值范围为0~1000。	100
cbow	否	训练使用的语言模型。取值范围为0（skip-gram模型）和1（cbow模型）。	0
window	否	单词的窗口大小。取值范围为正整数。	5
minCount	否	截断的最小词频。取值范围为正整数。	5
hs	否	是否采用HIERARCHICAL SOFTMAX。取值范围为0（不采用）和1（采用）。	1
negative	否	负采样的窗口大小。取值范围为正整数，0表示不可用。	5
sample	否	向下采样的阈值。取值范围为1e-3～1e-5，默认值为1e-3，0表示不可用。	1e-3
alpha	否	取值大于0。	0.025
iterTrain	否	取值大于等于1。	1
randomWindow	否	指定单词窗口的展示方式。取值范围为0（不随机，其值由window参数指定）和1（大小在1～5间随机）。	1

参数名称	是否必选	描述	默认值
outVocabularyTableName	否	输出词汇表名称。	无
outputWordTableName	是	输出词向量表名称。	无
outputDocTableName	是	输出文档向量表名称。	无
lifecycle	否	输出表的生命周期。取值范围为正整数。	无
coreNum	否	核心数，需要与memSizePerCore参数同时设置才生效。取值范围为正整数。	系统自动分配
memSizePerCore	否	内存数，需要与coreNum参数同时设置才生效。取值范围为正整数。	系统自动分配

3.7.18. 条件随机场

本文为您介绍PAI-Studio提供的条件随机场算法组件。

条件随机场CRF（conditional random field）是给定一组输入随机变量条件下，另一组输出随机变量条件的概率分布模型，其特点是假设输出随机变量构成马尔可夫随机场。条件随机场可用于不同的预测问题，主要应用于标注问题中，其中最典型的是线性链（linear chain）。详情请参见wiki。

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	**请选择ID列**	样本以N元组的形式存储，ID列为一条样本的唯一ID。
	请选择特征列	要进行标注的单词，以及该单词所对应的特征（如果有）。
	请选择目标列	选择目标列。

页签	参数	描述
参数设置	特征生成模板	默认值为 [-2:0],[-1:0],[0:0],[1:0],[2:0],[-1:0]/[0:0], [0:0]/[1:0],[-2:1],[-1:1],[0:1], [1:1],[2:1], [-2:1]/[-1:1],[-1:1]/[0:1], [0:1]/[1:1],[1:1]/[2:1], [-2:1]/[-1:1]/[0:1],[-1:1]/[0:1]/[1:1], [0:1]/[1:1]/[2:1] 。
	低频词过滤阈值	默认值为1。
	L1正则项系数	默认值为1。
	L2正则项系数	默认值为0。
	最大迭代次数	默认值为100。
	收敛阈值	默认值为0.00001。
执行调优	核心数	默认自动调整。
	每个核心的内容	默认自动调整。

- PAI命令方式

```
PAI -name=linearcrf
  -project=algo_public
  -DinputTableName=crf_input_table
  -DidColName=sentence_id
  -DfeatureColNames=word,f1
  -DlabelColName=label
  -DoutputTableName=crf_model
  -Dlifecycle=28
  -DcoreNum=10
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入特征数据表	无
inputTablePartitions	否	输入特征表选择的分区	全表
featureColNames	否	输入表选择的特征列	默认选择全部，自动排除label列。
labelColName	是	目标列	无
idColName	是	样本标号列	无

参数名称	是否必选	描述	默认值
outputTableName	是	输出模型表	无
outputTablePartitions	否	输出模型表选择的分区	全表
template	否	算法特征生成的模板	◦ 定义 <template .=. <template_item, <template_item, ..., <template_item <template_item .=. [row_offset:col_index]/[row_offset:col_index]/.../[row_offset:col_index] row_offset .=. integer col_index .=. integer ◦ 默认值 [-2:0],[-1:0],[0:0],[1:0],[2:0],[-1:0]/[0:0],[0:0]/[1:0],[-2:1],[-1:1],[0:1],[1:1],[2:1],[-2:1]/[-1:1],[-1:1]/[0:1],[0:1]/[1:1],[1:1]/[2:1],[-2:1]/[-1:1]/[0:1],[-1:1]/[0:1]/[1:1],[0:1]/[1:1]/[2:1]
freq	否	过滤特征的参数，算法只保留出现次数大于等于freq的特征。	1
iterations	否	优化的最大迭代次数	100

参数名称	是否必选	描述	默认值
l1Weight	否	L1正则的参数权重	1.0
l2Weight	否	L2正则的参数权重	1.0
epsilon	否	收敛误差。L-BFGS的终止条件，即两次迭代之间log-likelihood的差。	0.0001
lbfgsStep	否	lbfgs优化过程中的历史长度，仅对lbfgs有效。	10
threadNum	否	模型训练时并行启动线程的数量	3
lifecycle	否	输出表的生命周期	无
coreNum	否	核心数	自动计算
memSizePerCore	否	内存数	自动计算

示例

- 输入数据

sentence_id	word	f1	label
1	Rockwell	NNP	B-NP
1	International	NNP	I-NP
1	Corp	NNP	I-NP
1	‘s	POS	B-NP
…	…	…	…
823	Ohio	NNP	B-NP
823	grew	VBD	B-VP
823	3.8	CD	B-NP
823	%	NN	I-NP
823	.	.	O

- 预测算法PAI命令

```
PAI -name=crf_predict
  -project=algo_public
  -DinputTableName=crf_test_input_table
  -DmodelTableName=crf_model
  -DidColName=sentence_id
  -DfeatureColNames=word,f1
  -DlabelColName=label
  -DoutputTableName=crf_predict_result
  -DdetailColName=prediction_detail
  -Dlifecycle=28
  -DcoreNum=10
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入特征数据表	无
inputTablePartitions	否	输入特征表选择的分区	全表
featureColNames	否	输入表选择的特征列	默认选择全部，自动排除label列。
labelColName	否	目标列	无
IdColName	是	样本标号列	无
resultColName	否	输出表中result列名	prediction_result
scoreColName	否	输出表中score列名	prediction_score
detailColName	否	输出表中detail列名	无
outputTableName	是	输出预测结果表	无
outputTablePartitions	否	输出预测结果表选择的分区	全表
modelTableName	是	算法模型表	无
modelTablePartitions	否	算法模型表选择的分区	全表
lifecycle	否	输出表的生命周期	无
coreNum	否	核心数	自动计算
memSizePerCore	否	内存数	自动计算

- 输出数据

sentence_id	word	f1	label
1	Confidence	NN	B-NP
1	in	IN	B-PP

sentence_id	word	f1	label
1	the	DT	B-NP
1	pound	NN	I-NP
...	...	...	...
77	have	VBP	B-VP
77	announced	VBN	I-VP
77	similar	JJ	B-NP
77	increases	NNS	I-NP
77	.	.	O

> **说明** label列为可选列。

3.7.19. 文章相似度

本文为您介绍PAI-Studio提供的文章相似度算法组件。

文章相似度是在字符串相似度的基础上，基于词，计算两两文章或者句子之间的相似度。文章或句子需要以空格分割，计算方式和字符串相似度类似，支持Levenshtein（Levenshtein Distance）、lCS（Longest Common SubString）、SSK（String Subsequence Kernel）、Cosine（Cosine）和Simhash_Hamming五种相似度计算方式。

- Levenshtein支持计算距离和相似度。
  - 距离在参数中表示为**levenshtein**。
  - 相似度=1-距离。在参数中表示为**levenshtein_sim**。
- lCS支持计算距离和相似度。
  - 距离在参数中表示为**lcs**。
  - 相似度=1-距离。相似度在参数中表示为**lcs_sim**。
- SSK支持计算相似度，在参数中表示为**ssk**。
- Cosine支持计算相似度，在参数中表示为**cosine**。
- Simhash_Hamming，其中SimHash算法是把原始的文本映射为64位的二进制指纹，Hamming Distance则是计算二进制指纹在相同位置上不同字符的个数，支持计算距离和相似度。
  - 距离在参数中表示为**simhash_hamming**。
  - 相似度=1-距离/64.0。相似度在参数中表示为**simhash_hamming_sim**。

> **说明**
> - 关于SimHash详细介绍请参见Similarity Estimation Techniques from Rounding Algorithms。
> - 关于HammingDistance详细介绍请参见wiki。

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	相似度计算中第一列的列名	默认为表中第一个string类型的列名。
	相似度计算中第二列的列名	默认为表中第二个string类型的列名。
	输出表追加的列名	指定输出表中追加的列名。
	输出表中相似度列的列名	指定输出表中相似度列的列名，默认值为output。 说明 列名中不能有特殊字符，只能用英文的a-z，A-Z及数字和下划线_，且以字母开头，名称的长度不超过128字节。
参数设置	相似度计算方法	指定相似度计算方法类型。取值范围为： ○ levenshtein ○ levenshtein_sim ○ lcs ○ lcs_sim ○ ssk ○ cosine ○ simhash_hamming ○ simhash_hamming_sim 默认值为levenshtein_sim。
	子串的长度，ssk和cosine中可用	仅当**相似度计算方法**取值为**levenshtein**、**ssk**或**cosine**时，该参数生效。取值范围为(0,100)。默认值为2。
	匹配词组合的权重，ssk中可用	仅当**相似度计算方法**取值为**ssk**时，该参数生效。取值范围为(0,1)。默认值为0.5。
执行调优	计算的核心数	系统自动选择。
	每个核心的内存（MB）	系统自动选择。

- PAI命令方式

```
PAI -name doc_similarity
  -project algo_public
  -DinputTableName="pai_test_doc_similarity"
  -DoutputTableName="pai_test_doc_similarity_output"
  -DinputSelectedColName1="col0"
  -DinputSelectedColName2="col1"
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表的表名	无
outputTableName	是	输出表的表名	无
inputSelectedColName1	否	相似度计算中第一列的列名	表中第一个类型为string的列名
inputSelectedColName2	否	相似度计算中第二列的列名	表中第二个类型为string的列名
inputAppendColNames	否	输出表追加的列名	不追加
inputTablePartitions	否	输入表选中的分区	选择全表
outputColName	否	输出表中相似度列的列名。 **说明** 列名中不能有特殊字符，只能用英文的a-z，A-Z及数字和下划线_，且以字母开头，名称的长度不超过128字节。	output
method	否	相似度计算方法，取值范围： ○ levenshtein ○ levenshtein_sim ○ lcs ○ lcs_sim ○ ssk ○ cosine ○ simhash_hamming ○ simhash_hamming_sim	levenshtein_sim
lambda	否	匹配词组合的权重，ssk中可用，取值范围为(0,1)。	0.5

参数名称	是否必选	描述	默认值
k	否	子串的长度，ssk和cosine中可用。取值范围为(0,100)。	2
lifecycle	否	指定输出表的生命周期	无
coreNum	否	计算的核心数	系统自动分配
memSizePerCore	否	每个核心的内存，单位为MB。	系统自动分配

示例

- 生成数据

```
drop table if exists pai_doc_similarity_input;
create table pai_doc_similarity_input as
select * from
(
select 0 as id, "北京 上海" as col0, "北京 上海" as col1 from dual
union all
select 1 as id, "北京 上海" as col0, "北京 上海 香港" as col1 from dual
)tmp
```

输入表为pai_doc_similarity_input，如下所示。

id	col0	col1
1	北京 上海	北京 上海 香港
0	北京 上海	北京 上海

- 运行PAI命令

```
drop table if exists pai_doc_similarity_output;
PAI -name doc_similarity
  -project algo_public
  -DinputTableName=pai_doc_similarity_input
  -DoutputTableName=pai_doc_similarity_output
  -DinputSelectedColName1=col0
  -DinputSelectedColName2=col1
  -Dmethod=levenshtein_sim
  -DinputAppendColNames=id,col0,col1;
```

- 输出结果

输出表为pai_doc_similarity_output，如下所示。

id	col0	col1	output
1	北京 上海	北京 上海 香港	0.6666666666666667

id	col0	col1	output
0	北京 上海	北京 上海	1.0

常见问题

- 相似度计算是基于分词的结果，即以空格分割的每个词作为相似度计算的一个单位。如果是以字符串整体输入，需要使用字符串相似度方法。
- 参数method中，levenshtein、lcs、simhash_hamming为计算距离。levenshtein_sim、lcs_sim、ssk、cosine、simhash_hamming_sim为计算相似度。距离=1.0-相似度。
- 相似度计算方法为cosine或ssk时，存在参数k，表示以k个词作为一个组合，进行相似度计算。如果k大于词的个数，即是两个相同的字符串，相似度输出也为0。此时需要调小k的值，使其小于或等于最小词个数。

3.7.20. PMI

本文为您介绍PAI-Studio提供的PMI算法组件。

互信息（Mutual Information）是信息论里一种有用的信息度量，它可以看成是一个随机变量中包含的另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不确定性。

本算法统计若干文章中所有词的共现情况，计算两两之间的PMI（point mutual information）。其定义为：**PMI(x,y)=ln(p(x,y)/(p(x)p(y)))=ln(#(x,y)D/(#x#y))**。其中，#(x,y)为pair(x,y)的count数，D为pair的总数。若x、y在同一个窗口出现，那么#x+=1; #y+=1;#(x,y)+=1。了解更多PMI的信息，请参见PMI。

配置组件

PAI-Studio支持通过可视化或PAI命令的方式，配置组件参数：

- 可视化方式

页签	参数	描述
字段设置	**分词好的文档列名，分词用空格隔开**	无
参数设置	**截断的最小词频**	出现次数少于该值的词会被过滤掉。默认值为5。
	窗口大小	例如5指当前词右边相邻的5个词（不包含当前词）。在窗口中出现的词被被认为与当前词相关。
执行调优	**计算的核心数**	系统自动选择。
	每个核心的内存（MB）	系统自动选择。

- PAI命令方式

```
PAI -name PointwiseMutualInformation
  -project algo_public
  -DinputTableName=maple_test_pmi_basic_input
  -DdocColName=doc
  -DoutputTableName=maple_test_pmi_basic_output
  -DminCount=0
  -DwindowSize=2
  -DcoreNum=1
  -DmemSizePerCore=110;
```

参数名称	是否必选	描述	默认值
inputTableName	是	输入表	无
outputTableName	是	输出表	无
docColName	是	分词好的文档列名，分词用空格隔开。	无
windowSize	否	窗口大小。例如5指当前词右边相邻的5个词（不包含当前词）。在窗口中出现的词被被认为与当前词相关。	默认整行内容
minCount	否	截断的最小词频，出现次数少于该值的词会被过滤掉。	5
inputTablePartitions	否	输入表中指定哪些分区参与训练，格式为：Partition_name=value。如果是多级格式为name1=value1/name2=value2。如果指定多个分区，中间用“,”隔开。	选择所有分区
lifecycle	否	指定输出表的生命周期	无
coreNum	否	节点个数，取值范围为[1,9999]。	自动计算
memSizePerCore	否	单个节点内存大小，单位为MB，取值范围为[1024,64*1024]。	自动计算

示例

- 生成数据

```
create table maple_test_pmi_basic_input as
select * from
(
  select "w1 w2 w3 w4 w5 w6 w7 w8 w8 w9" as doc from dual
  union all
  select "w1 w3 w5 w6 w9" as doc from dual
  union all  select "w0" as doc from dual
  union all
  select "w0 w0" as doc from dual
  union all
  select "w9 w1 w9 w1 w9" as doc from dual
)tmp;
```

doc:string
w1 w2 w3 w4 w5 w6 w7 w8 w8 w9
w1 w3 w5 w6 w9
w0
w0 w0
w9 w1 w9 w1 w9

- 运行PAI命令

```
PAI -name PointwiseMutualInformation
  -project algo_public
  -DinputTableName=maple_test_pmi_basic_input
  -DdocColName=doc
  -DoutputTableName=maple_test_pmi_basic_output
  -DminCount=0
  -DwindowSize=2
  -DcoreNum=1
  -DmemSizePerCore=110;
```

- 输出结果

word1	word2	word1_count	word2_count	co_occurrences_count	pmi
w0	w0	2	2	1	2.0794415416798357
w1	w1	10	10	1	-1.1394342831883648
w1	w2	10	3	1	0.06453852113757116

word1	word2	word1_count	word2_count	co_occurrences_count	pmi
w1	w3	10	7	2	-0.089612158689687 04
w1	w5	10	8	1	-0.916290731874155
w1	w9	10	12	4	0.064538521137577 16
w2	w3	3	7	1	0.421213465076303 5
w2	w4	3	4	1	0.980829253011726 2
w3	w4	7	4	1	0.133531392624522 57
w3	w5	7	8	2	0.133531392624522 57
w3	w6	7	7	1	-0.426084395310900 14
w4	w5	4	8	1	0
w4	w6	4	7	1	0.133531392624522 57
w5	w6	8	7	2	0.133531392624522 57
w5	w7	8	4	1	0
w5	w9	8	12	1	-1.098612288668109 8
w6	w7	7	4	1	0.133531392624522 57
w6	w8	7	7	1	-0.426084395310900 14
w6	w9	7	12	1	-0.965080896043587 2

word1	word2	word1_count	word2_count	co_occurrences_count	pmi
w7	w8	4	7	2	0.8266785731844679
w8	w8	7	7	1	-0.42608439531090014
w8	w9	7	12	2	-0.27193371548364 18
w9	w9	12	12	2	-0.81093021621632 88

3.8. 网络分析

3.8.1. k-Core

k-Core算法用于在图中找出符合指定核心度的紧密关联的子图结构，节点核数的最大值被称为图的核数。本文为您介绍PAI-Studio提供的k-Core组件。

PAI-Studio支持通过可视化或PAI命令方式，配置k-Core组件的参数。

可视化方式

页签	参数	描述
字段设置	**选择源顶点列**	边表的起点所在列。
	选择目标顶点列	边表的终点所在列。
参数设置	**k 核数**	核数的值，必填，默认为3。
执行调优	**进程数**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。

PAI命令方式

```
PAI -name KCore
  -project algo_public
  -DinputEdgeTableName=KCore_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=KCore_func_test_result
  -Dk=2;
```

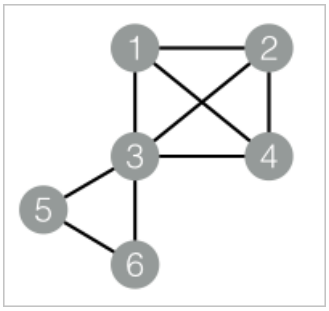
参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64
k	是	核数。	3

使用示例

1. 生成训练数据。

```
drop table if exists KCore_func_test_edge;
create table KCore_func_test_edge as
select * from
(
 select '1' as flow_out_id,'2' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'3' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '2' as flow_out_id,'3' as flow_in_id from dual
 union all
 select '2' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '3' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '3' as flow_out_id,'5' as flow_in_id from dual
 union all
 select '3' as flow_out_id,'6' as flow_in_id from dual
 union all
 select '5' as flow_out_id,'6' as flow_in_id from dual
)tmp;
```

对应的图结构如下所示。



2. 设定k=2，查看训练结果。

```
+-------+-------+
| node1 | node2 |
+-------+-------+
| 1     | 2     |
| 1     | 3     |
| 1     | 4     |
| 2     | 1     |
| 2     | 3     |
| 2     | 4     |
| 3     | 1     |
| 3     | 2     |
| 3     | 4     |
| 4     | 1     |
| 4     | 2     |
| 4     | 3     |
+-------+-------+
```

3.8.2. 单源最短路径

单源最短路径使用Dijkstra算法，给定起点，输出该点和其他所有节点的最短路径。本文为您介绍PAI-Studio提供的单源最短路径组件。

PAI-Studio支持通过可视化或PAI命令方式，配置单源最短路径组件的参数。

可视化方式

页签	参数	描述
字段设置	**选择源顶点列**	边表的起点所在列。
	选择目标顶点列	边表的终点所在列。
	选择边权值列	边表边的权重所在列。
参数设置	**起始节点 ID**	用于计算最短路径的起始节点，必填。
执行调优	**进程数**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。

PAI命令方式

```
PAI -name SSSP
  -project algo_public
  -DinputEdgeTableName=SSSP_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=SSSP_func_test_result
  -DhasEdgeWeight=true
  -DedgeWeightCol=edge_weight
  -DstartVertex=a;
```

参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无

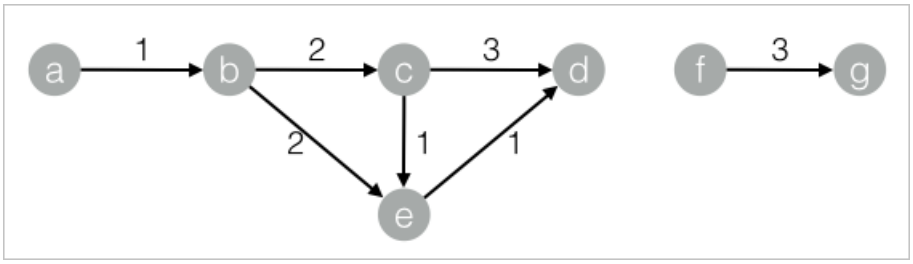
参数	是否必选	描述	默认值
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64
startVertex	是	起始节点ID。	无
hasEdgeWeight	否	输入边表的边是否有权重。	false
edgeWeightCol	否	输入边表边的权重所在列。	无

使用示例

1. 生成训练数据。

```
drop table if exists SSSP_func_test_edge;
create table SSSP_func_test_edge as
select
  flow_out_id,flow_in_id,edge_weight
from
(
  select "a" as flow_out_id,"b" as flow_in_id,1.0 as edge_weight from dual
  union all
  select "b" as flow_out_id,"c" as flow_in_id,2.0 as edge_weight from dual
  union all
  select "c" as flow_out_id,"d" as flow_in_id,1.0 as edge_weight from dual
  union all
  select "b" as flow_out_id,"e" as flow_in_id,2.0 as edge_weight from dual
  union all
  select "e" as flow_out_id,"d" as flow_in_id,1.0 as edge_weight from dual
  union all
  select "c" as flow_out_id,"e" as flow_in_id,1.0 as edge_weight from dual
  union all
  select "f" as flow_out_id,"g" as flow_in_id,3.0 as edge_weight from dual
  union all
  select "a" as flow_out_id,"d" as flow_in_id,4.0 as edge_weight from dual
) tmp;
```

对应的图结构如下所示。



2. 查看训练结果。

```
+------------+------------+------------+--------------+
|start_node|dest_node |distance  |distance_cnt|
+------------+------------+------------+--------------+
|a     |b     |1.0    |1      |
|a     |c     |3.0    |1      |
|a     |d     |4.0    |3      |
|a     |a     |0.0    |0      |
|a     |e     |3.0    |1      |
+------------+------------+------------+--------------+
```

3.8.3. PageRank

PageRank起源于网页的搜索排序，即使用网页的链接结构计算每个网页的等级排名。本文为您介绍PAI-Studio提供的PageRank组件。

背景信息

PageRank的基本思路是：

- 如果一个网页被其他多个网页指向，这说明该网页比较重要或者质量较高。
- 除考虑网页的链接数量，还考虑网页本身的权重级别，以及该网页有多少链接链到其它网页。
- 对于用户构成的人际网络，除了用户本身的影响力之外，边的权重也是重要因素之一。

例如新浪微博的某个用户，会更容易影响粉丝中关系比较亲密的家人、同学、同事等，而对陌生的弱关系粉丝影响较小。在人际网络中，边的权重等价于用户与用户之间的关系强弱指数。

带链接权重的PageRank公式为：

$$W(A) = (1-d) + d * (\sum_{i} W(i) * C(Ai))$$

- W(i)：节点i的权重。
- C(Ai)：链接权重。
- d：阻尼系数。
- W(A)：算法迭代稳定后的节点权重，即每个用户的影响力指数。

PAI-Studio支持通过可视化或PAI命令方式，配置PageRank组件的参数。

可视化方式

页签	参数	描述
字段设置	**选择源顶点列**	边表的起点所在列。
	选择目标顶点列	边表的终点所在列。
	选择边权值列	边表边的权重所在列。
参数设置	**最大迭代次数**	算法自身会收敛并停止迭代，可选，默认为30。
	阻尼系数	在任意时刻，用户到达某网页后并继续向后浏览的概率。
执行调优	**进程数**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。

PAI命令方式

```
PAI -name PageRankWithWeight
  -project algo_public
  -DinputEdgeTableName=PageRankWithWeight_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=PageRankWithWeight_func_test_result
  -DhasEdgeWeight=true
  -DedgeWeightCol=weight
  -DmaxIter 100;
```

参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置

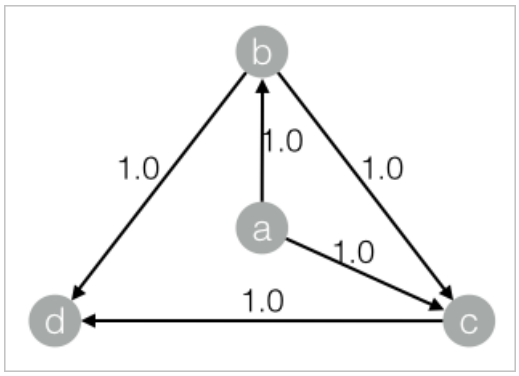
参数	是否必选	描述	默认值
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64
hasEdgeWeight	否	输入边表的边是否有权重。	false
edgeWeightCol	否	输入边表边的权重所在列。	无
maxIter	否	最大迭代次数。	30

使用示例

1. 生成训练数据。

```
drop table if exists PageRankWithWeight_func_test_edge;
create table PageRankWithWeight_func_test_edge as
select * from
(
  select 'a' as flow_out_id,'b' as flow_in_id,1.0 as weight from dual
  union all
  select 'a' as flow_out_id,'c' as flow_in_id,1.0 as weight from dual
  union all
  select 'b' as flow_out_id,'c' as flow_in_id,1.0 as weight from dual
  union all
  select 'b' as flow_out_id,'d' as flow_in_id,1.0 as weight from dual
  union all
  select 'c' as flow_out_id,'d' as flow_in_id,1.0 as weight from dual
)tmp;
```

对应的图结构如下所示。



2. 查看训练结果。

```
+------+------------+
|node|weight    |
+------+------------+
|a   |0.0375    |
|b   |0.06938   |
|c   |0.12834   |
|d   |0.20556   |
+------+------------+
```

3.8.4. 标签传播聚类

标签传播算法LPA（Label Propagation Algorithm）是基于图的半监督学习方法，其基本思路是节点的标签（community）依赖其相邻节点的标签信息，影响程度由节点相似度决定，并通过传播迭代更新达到稳定。本文为您介绍PAI-Studio提供的标签传播聚类组件。

背景信息

图聚类是根据图的拓扑结构，进行子图的划分，使得子图内部节点的连接较多，子图之间的连接较少。

PAI-Studio支持通过可视化或PAI命令方式，配置标签传播聚类组件的参数。

可视化方式

页签	参数	描述
字段设置	**顶点表：选择顶点列**	点表的点所在列。
	顶点表：选择权值列	点表的点的权重所在列。
	边表：选择源顶点列	边表的起点所在列。
	边表：选择目标顶点列	边表的终点所在列。
	边表：选择权值列	边表边的权重所在列。
参数设置	**最大迭代次数**	可选，默认为30。
执行调优	**进程数**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。

PAI命令方式

```
PAI -name LabelPropagationClustering
  -project algo_public
  -DinputEdgeTableName=LabelPropagationClustering_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DinputVertexTableName=LabelPropagationClustering_func_test_node
  -DvertexCol=node
  -DoutputTableName=LabelPropagationClustering_func_test_result
  -DhasEdgeWeight=true
  -DedgeWeightCol=edge_weight
  -DhasVertexWeight=true
  -DvertexWeightCol=node_weight
  -DrandSelect=true
  -DmaxIter=100;
```

参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
inputVertexTableName	是	输入点表名称。	无
inputVertexTablePartitions	否	输入点表的分区。	全表读入
vertexCol	是	输入点表的点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64

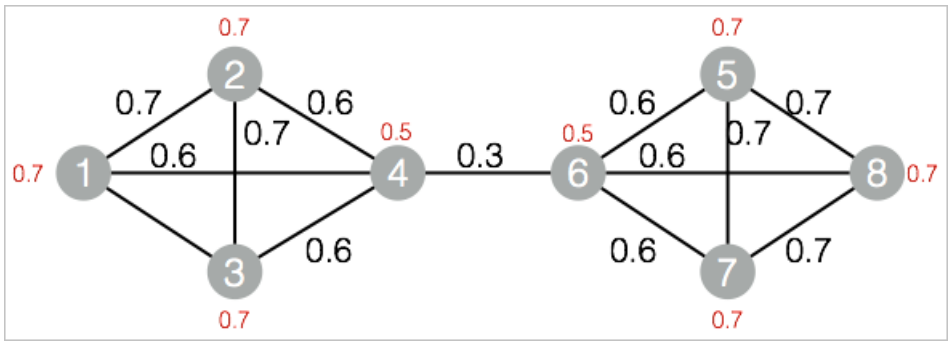
参数	是否必选	描述	默认值
hasEdgeWeight	否	输入边表的边是否有权重。	false
edgeWeightCol	否	输入边表边的权重所在列。	无
hasVertexWeight	否	输入点表的点是否有权重。	false
vertexWeightCol	否	输入点表的点的权重所在列。	无
randSelect	否	是否随机选择最大标签。	false
maxIter	否	最大迭代次数。	30

使用示例

1. 生成训练数据。

```
drop table if exists LabelPropagationClustering_func_test_edge;
create table LabelPropagationClustering_func_test_edge as
select * from
(
  select '1' as flow_out_id,'2' as flow_in_id,0.7 as edge_weight from dual
  union all
  select '1' as flow_out_id,'3' as flow_in_id,0.7 as edge_weight from dual
  union all
  select '1' as flow_out_id,'4' as flow_in_id,0.6 as edge_weight from dual
  union all
  select '2' as flow_out_id,'3' as flow_in_id,0.7 as edge_weight from dual
  union all
  select '2' as flow_out_id,'4' as flow_in_id,0.6 as edge_weight from dual
  union all
  select '3' as flow_out_id,'4' as flow_in_id,0.6 as edge_weight from dual
  union all
  select '4' as flow_out_id,'6' as flow_in_id,0.3 as edge_weight from dual
  union all
  select '5' as flow_out_id,'6' as flow_in_id,0.6 as edge_weight from dual
  union all
  select '5' as flow_out_id,'7' as flow_in_id,0.7 as edge_weight from dual
  union all
  select '5' as flow_out_id,'8' as flow_in_id,0.7 as edge_weight from dual
  union all
  select '6' as flow_out_id,'7' as flow_in_id,0.6 as edge_weight from dual
  union all
  select '6' as flow_out_id,'8' as flow_in_id,0.6 as edge_weight from dual
  union all
  select '7' as flow_out_id,'8' as flow_in_id,0.7 as edge_weight from dual
)tmp
;
drop table if exists LabelPropagationClustering_func_test_node;
create table LabelPropagationClustering_func_test_node as
select * from
(
  select '1' as node,0.7 as node_weight from dual
  union all
  select '2' as node,0.7 as node_weight from dual
  union all
  select '3' as node,0.7 as node_weight from dual
  union all
  select '4' as node,0.5 as node_weight from dual
  union all
  select '5' as node,0.7 as node_weight from dual
  union all
  select '6' as node,0.5 as node_weight from dual
  union all
  select '7' as node,0.7 as node_weight from dual
  union all
  select '8' as node,0.7 as node_weight from dual
)tmp;
```

对应的图结构如下所示。



2. 查看训练结果。

```
+------+------------+
|node|group_id  |
+------+------------+
|1  |1     |
|2  |1     |
|3  |1     |
|4  |1     |
|5  |5     |
|6  |5     |
|7  |5     |
|8  |5     |
+------+------------+
```

3.8.5. 标签传播分类

标签传播分类为半监督的分类算法，原理为用已标记节点的标签信息去预测未标记节点的标签信息。本文为您介绍PAI-Studio提供的标签传播分类组件。

背景信息

标签传播分类算法的基本思路：在算法执行过程中，每个节点的标签按相似度传播给相邻节点，在节点传播的每一步，每个节点根据相邻节点的标签来更新自己的标签。与该节点相似度越大，其相邻节点对其标注的影响权值越大，相似节点的标签越趋于一致，其标签就越容易传播。在标签传播过程中，保持已标注数据的标签不变，使其像一个源头把标签传向未标注数据。最终，当迭代过程结束时，相似节点的概率分布也趋于相似，可以划分到同一个类别中，从而完成标签传播过程。

PAI-Studio支持通过可视化或PAI命令方式，配置标签传播分类组件的参数。

可视化方式

页签	参数	描述
字段设置	**顶点表：选择顶点列**	点表的点所在列。
	顶点表：选择标签列	点表的点的权重所在列。
	顶点表：选择权值列	边表的起点所在列。
	边表：选择源顶点列	边表的终点所在列。

页签	参数	描述
	边表：选择目标顶点列	边表的终点所在列。
	边表：选择权值列	边表边的权重所在列。
参数设置	**最大迭代次数**	可选，默认为30。
	阻尼系数	默认为0.8。
	收敛系数	默认为0.000001。
执行调优	**进程数**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。

PAI命令方式

```
PAI -name LabelPropagationClassification
  -project algo_public
  -DinputEdgeTableName=LabelPropagationClassification_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DinputVertexTableName=LabelPropagationClassification_func_test_node
  -DvertexCol=node
  -DvertexLabelCol=label
  -DoutputTableName=LabelPropagationClassification_func_test_result
  -DhasEdgeWeight=true
  -DedgeWeightCol=edge_weight
  -DhasVertexWeight=true
  -DvertexWeightCol=label_weight
  -Dalpha=0.8
  -Depsilon=0.000001;
```

参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
inputVertexTableName	是	输入点表名称。	无
inputVertexTablePartitions	否	输入点表的分区。	全表读入

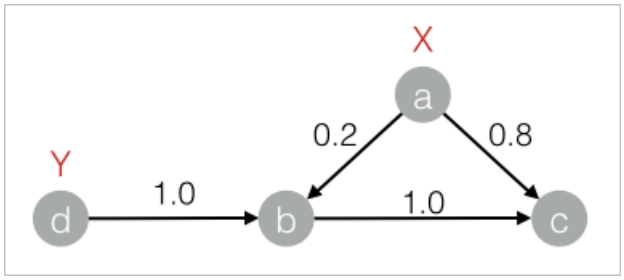
参数	是否必选	描述	默认值
vertexCol	是	输入点表的点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64
hasEdgeWeight	否	输入边表的边是否有权重。	false
edgeWeightCol	否	输入边表边的权重所在列。	无
hasVertexWeight	否	输入点表的点是否有权重。	false
vertexWeightCol	否	输入点表的点的权重所在列。	无
alpha	否	阻尼系数。	0.8
epsilon	否	收敛系数。	0.000001
maxIter	否	最大迭代次数。	30

使用示例

1. 生成训练数据。

```
drop table if exists LabelPropagationClassification_func_test_edge;
create table LabelPropagationClassification_func_test_edge as
select * from
(
  select 'a' as flow_out_id, 'b' as flow_in_id, 0.2 as edge_weight from dual
  union all
  select 'a' as flow_out_id, 'c' as flow_in_id, 0.8 as edge_weight from dual
  union all
  select 'b' as flow_out_id, 'c' as flow_in_id, 1.0 as edge_weight from dual
  union all
  select 'd' as flow_out_id, 'b' as flow_in_id, 1.0 as edge_weight from dual
)tmp
;
drop table if exists LabelPropagationClassification_func_test_node;
create table LabelPropagationClassification_func_test_node as
select * from
(
  select 'a' as node,'X' as label, 1.0 as label_weight from dual
  union all
  select 'd' as node,'Y' as label, 1.0 as label_weight from dual
)tmp;
```

对应的图结构如下所示。



2. 查看训练结果。

```
+------+-----+------------+
| node | tag | weight     |
+------+-----+------------+
| a    | X   | 1.0        |
| b    | X   | 0.16667    |
| b    | Y   | 0.83333    |
| c    | X   | 0.53704    |
| c    | Y   | 0.46296    |
| d    | Y   | 1.0        |
+------+-----+------------+
```

3.8.6. Modularity

Modularity是一种评估社区网络结构的指标，用来评估网络结构中划分出来社区的紧密程度，通常0.3以上是比较明显的社区结构。本文为您介绍PAI-Studio提供的Modularity组件。

PAI-Studio支持通过可视化或PAI命令方式，配置Modularity组件的参数。

可视化方式

页签	参数	描述
字段设置	**源顶点列**	边表的起点所在列。
	起始点标签列	边表起点的群组。
	目标顶点列	边表的终点所在列。
	目标点标签列	边表终点的群组。
执行调优	**进程数**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。

PAI命令方式

```
PAI -name Modularity
  -project algo_public
  -DinputEdgeTableName=Modularity_func_test_edge
  -DfromVertexCol=flow_out_id
  -DfromGroupCol=group_out_id
  -DtoVertexCol=flow_in_id
  -DtoGroupCol=group_in_id
  -DoutputTableName=Modularity_func_test_result;
```

参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
fromGroupCol	是	输入边表起点的群组。	无
toVertexCol	是	输入边表的终点所在列。	无
toGroupCol	是	输入边表终点的群组。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置

参数	是否必选	描述	默认值
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64

使用示例

1. 生成训练数据。

   与标签传播聚类算法的数据类似，详情请参见标签传播聚类。

2. 查看训练结果。

```
+--------------+
| val      |
+--------------+
| 0.4230769   |
+--------------+
```

3.8.7. 最大连通子图

在无向图G中，若从顶点A到顶点B有路径相连，则称A和B是连通的。在图G中存在若干子图，如果其中每个子图中所有顶点之间都是连通的，但在不同子图间不存在顶点连通，那么称图G的这些子图为最大连通子图。本文为您介绍PAI-Studio提供的最大连通子图组件。

PAI-Studio支持通过可视化或PAI命令方式，配置最大连通子图组件的参数。

可视化方式

页签	参数	描述
IO/字段设置	**起始节点**	边表的起点所在列。
	结束节点	边表的终点所在列。
执行调优	**进程数量**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。
	数据切分大小	数据切分的大小，默认为64。

PAI命令方式

```
PAI -name MaximalConnectedComponent
  -project algo_public
  -DinputEdgeTableName=MaximalConnectedComponent_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=MaximalConnectedComponent_func_test_result;
```

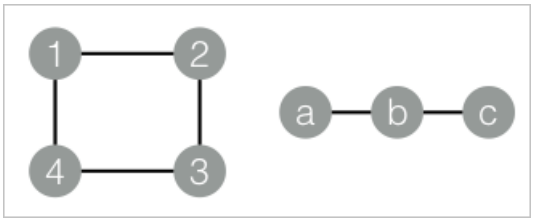
参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64

使用示例

1. 生成训练数据。

```
drop table if exists MaximalConnectedComponent_func_test_edge;
create table MaximalConnectedComponent_func_test_edge as
select * from
(
 select '1' as flow_out_id,'2' as flow_in_id from dual
 union all
 select '2' as flow_out_id,'3' as flow_in_id from dual
 union all
 select '3' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'4' as flow_in_id from dual
 union all
 select 'a' as flow_out_id,'b' as flow_in_id from dual
 union all
 select 'b' as flow_out_id,'c' as flow_in_id from dual
)tmp;
drop table if exists MaximalConnectedComponent_func_test_result;
create table MaximalConnectedComponent_func_test_result
(
 node string,
 grp_id string
);
```

对应的图结构如下图所示。



2. 查看训练结果。

```
+-------+-------+
| node | grp_id|
+-------+-------+
|1   |4   |
|2   |4   |
|3   |4   |
|4   |4   |
|a   |c   |
|b   |c   |
|c   |c   |
+-------+-------+
```

3.8.8. 点聚类系数

在无向图G中，计算每一个节点周围的稠密度，星状网络稠密度为0，全联通网络稠密度为1。本文为您介绍PAI-Studio提供的点聚类系数组件。

PAI-Studio支持通过可视化或PAI命令方式，配置点聚类系数的参数。

可视化方式

页签	参数	描述
IO/字段设置	**起始节点**	边表的起点所在列。
	终止节点	边表的终点所在列。
参数设置	**最大节点度**	如果节点度大于该值，则进行抽样。默认为500，选填。
执行调优	**进程数量**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。
	数据切分大小	数据切分的大小，默认为64。

PAI命令方式

```
PAI -name NodeDensity
  -project algo_public
  -DinputEdgeTableName=NodeDensity_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=NodeDensity_func_test_result
  -DmaxEdgeCnt=500;
```

参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
maxEdgeCnt	否	如果节点度大于该值，则进行抽样。	500
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置

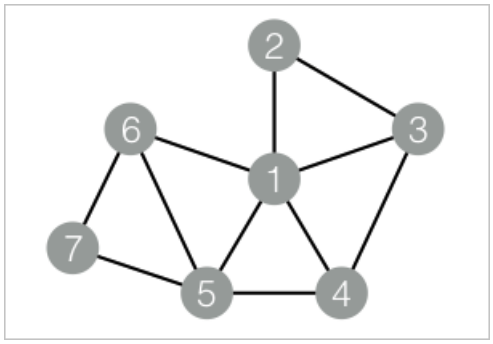
参数	是否必选	描述	默认值
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64

使用示例

1. 生成训练数据。

```
drop table if exists NodeDensity_func_test_edge;
create table NodeDensity_func_test_edge as
select * from
(
 select '1' as flow_out_id, '2' as flow_in_id from dual
 union all
 select '1' as flow_out_id, '3' as flow_in_id from dual
 union all
 select '1' as flow_out_id, '4' as flow_in_id from dual
 union all
 select '1' as flow_out_id, '5' as flow_in_id from dual
 union all
 select '1' as flow_out_id, '6' as flow_in_id from dual
 union all
 select '2' as flow_out_id, '3' as flow_in_id from dual
 union all
 select '3' as flow_out_id, '4' as flow_in_id from dual
 union all
 select '4' as flow_out_id, '5' as flow_in_id from dual
 union all
 select '5' as flow_out_id, '6' as flow_in_id from dual
 union all
 select '5' as flow_out_id, '7' as flow_in_id from dual
 union all
 select '6' as flow_out_id, '7' as flow_in_id from dual
)tmp;
drop table if exists NodeDensity_func_test_result;
create table NodeDensity_func_test_result
(
 node string,
 node_cnt bigint,
 edge_cnt bigint,
 density double,
 log_density double
);
```

对应的图结构如下图所示。



2. 查看训练结果。

```
1,5,4,0.4,1.45657
2,2,1,1.0,1.24696
3,3,2,0.66667,1.35204
4,3,2,0.66667,1.35204
5,4,3,0.5,1.41189
6,3,2,0.66667,1.35204
7,2,1,1.0,1.24696
```

3.8.9. 边聚类系数

边聚类系数算法是指在无向图G中，计算每一条边周围的稠密度。本文为您介绍PAI-Studio提供的边聚类系数组件。

PAI-Studio支持通过可视化或PAI命令方式，配置边聚类系数组件的参数。

可视化方式

页签	参数	描述
IO/字段设置	**起始节点**	边表的起点所在列。
	结束节点	边表的终点所在列。
执行调优	**进程数量**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。
	数据切分大小	数据切分的大小，默认为64。

PAI命令方式

```
PAI -name EdgeDensity
  -project algo_public
  -DinputEdgeTableName=EdgeDensity_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=EdgeDensity_func_test_result;
```

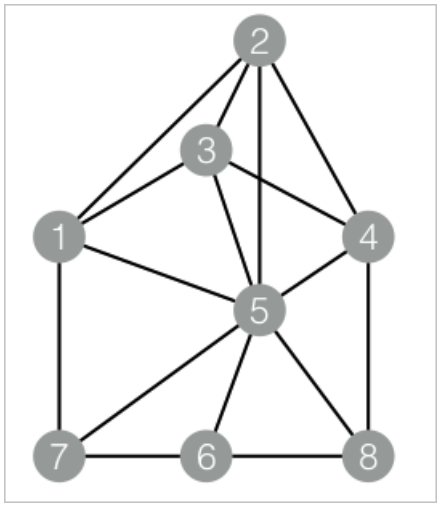
参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64

使用示例

1. 生成训练数据。

```
drop table if exists EdgeDensity_func_test_edge;
create table EdgeDensity_func_test_edge as
select * from
(
 select '1' as flow_out_id,'2' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'3' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'5' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'7' as flow_in_id from dual
 union all
 select '2' as flow_out_id,'5' as flow_in_id from dual
 union all
 select '2' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '2' as flow_out_id,'3' as flow_in_id from dual
 union all
 select '3' as flow_out_id,'5' as flow_in_id from dual
 union all
 select '3' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '4' as flow_out_id,'5' as flow_in_id from dual
 union all
 select '4' as flow_out_id,'8' as flow_in_id from dual
 union all
 select '5' as flow_out_id,'6' as flow_in_id from dual
 union all
 select '5' as flow_out_id,'7' as flow_in_id from dual
 union all
 select '5' as flow_out_id,'8' as flow_in_id from dual
 union all
 select '7' as flow_out_id,'6' as flow_in_id from dual
 union all
 select '6' as flow_out_id,'8' as flow_in_id from dual
)tmp;
drop table if exists EdgeDensity_func_test_result;
create table EdgeDensity_func_test_result
(
 node1 string,
 node2 string,
 node1_edge_cnt bigint,
 node2_edge_cnt bigint,
 triangle_cnt bigint,
 density double
);
```

对应的图结构如下图所示。



2. 查看训练结果。

```
1,2,4,4,2,0.5
2,3,4,4,3,0.75
2,5,4,7,3,0.75
3,1,4,4,2,0.5
3,4,4,4,2,0.5
4,2,4,4,2,0.5
4,5,4,7,3,0.75
5,1,7,4,3,0.75
5,3,7,4,3,0.75
5,6,7,3,2,0.66667
5,8,7,3,2,0.66667
6,7,3,3,1,0.33333
7,1,3,4,1,0.33333
7,5,3,7,2,0.66667
8,4,3,4,1,0.33333
8,6,3,3,1,0.33333
```

3.8.10. 计数三角形

计数三角形是指在无向图G中，输出所有三角形。本文为您介绍PAI-Studio提供的计数三角形组件。

PAI-Studio支持通过可视化或PAI命令方式，配置计数三角形组件的参数。

可视化方式

页签	参数	描述
IO/字段设置	**起始节点**	边表的起点所在列。
	结束节点	边表的终点所在列。
参数设置	**最大节点度**	如果节点度大于该值，则进行抽样。默认为500，选填。
	进程数量	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。

页签	参数	描述
执行调优		
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。
	数据切分大小	数据切分的大小，默认为64。

PAI命令方式

```
PAI -name TriangleCount
  -project algo_public
  -DinputEdgeTableName=TriangleCount_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=TriangleCount_func_test_result;
```

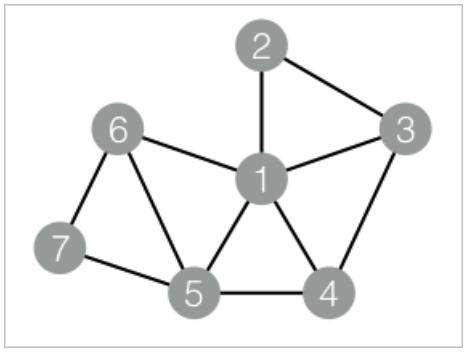
参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
maxEdgeCnt	否	如果节点度大于该值，则进行抽样。	500
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64

使用示例

1. 生成训练数据。

```
drop table if exists TriangleCount_func_test_edge;
create table TriangleCount_func_test_edge as
select * from
(
 select '1' as flow_out_id,'2' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'3' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'5' as flow_in_id from dual
 union all
 select '1' as flow_out_id,'6' as flow_in_id from dual
 union all
 select '2' as flow_out_id,'3' as flow_in_id from dual
 union all
 select '3' as flow_out_id,'4' as flow_in_id from dual
 union all
 select '4' as flow_out_id,'5' as flow_in_id from dual
 union all
 select '5' as flow_out_id,'6' as flow_in_id from dual
 union all
 select '5' as flow_out_id,'7' as flow_in_id from dual
 union all
 select '6' as flow_out_id,'7' as flow_in_id from dual
)tmp;
drop table if exists TriangleCount_func_test_result;
create table TriangleCount_func_test_result
(
 node1 string,
 node2 string,
 node3 string
);
```

对应的图结构如下图所示。



2. 查看训练结果。

```
1,2,3
1,3,4
1,4,5
1,5,6
5,6,7
```

3.8.11. 树深度

对于众多树状网络，树深度组件能够输出每个节点的所处深度和树ID。本文为您介绍PAI-Studio提供的树深度组件。

PAI-Studio支持通过可视化或PAI命令方式，配置树深度组件的参数。

可视化方式

页签	参数	描述
字段设置	**输入边表的起点所在列**	边表的起点所在列。
	输入边表的终点所在列	边表的终点所在列。
执行调优	**进程数量**	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。
	进程内存	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。
	数据切分大小	数据切分的大小，默认为64。

PAI命令方式

```
PAI -name TreeDepth
  -project algo_public
  -DinputEdgeTableName=TreeDepth_func_test_edge
  -DfromVertexCol=flow_out_id
  -DtoVertexCol=flow_in_id
  -DoutputTableName=TreeDepth_func_test_result;
```

参数	是否必选	描述	默认值
inputEdgeTableName	是	输入边表名。	无
inputEdgeTablePartitions	否	输入边表的分区。	全表读入
fromVertexCol	是	输入边表的起点所在列。	无
toVertexCol	是	输入边表的终点所在列。	无
outputTableName	是	输出表名。	无

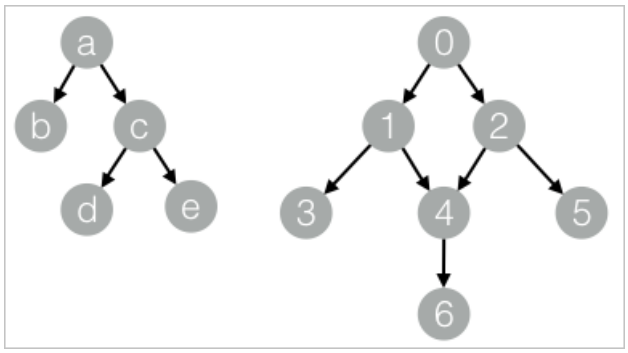
参数	是否必选	描述	默认值
outputTablePartitions	否	输出表的分区。	无
lifecycle	否	输出表的生命周期。	无
workerNum	否	作业并行执行的节点数。数字越大并行度越高，但是框架通讯开销会增大。	未设置
workerMem	否	单个作业可使用的最大内存量。系统默认为每个作业分配4096 MB内存，实际使用内存超过该值，会抛出OutOfMemory异常。	4096
splitSize	否	数据切分大小。	64

使用示例

1. 生成训练数据。

```
drop table if exists TreeDepth_func_test_edge;
create table TreeDepth_func_test_edge as
select * from
(
  select '0' as flow_out_id, '1' as flow_in_id from dual
  union all
  select '0' as flow_out_id, '2' as flow_in_id from dual
  union all
  select '1' as flow_out_id, '3' as flow_in_id from dual
  union all
  select '1' as flow_out_id, '4' as flow_in_id from dual
  union all
  select '2' as flow_out_id, '4' as flow_in_id from dual
  union all
  select '2' as flow_out_id, '5' as flow_in_id from dual
  union all
  select '4' as flow_out_id, '6' as flow_in_id from dual
  union all
  select 'a' as flow_out_id, 'b' as flow_in_id from dual
  union all
  select 'a' as flow_out_id, 'c' as flow_in_id from dual
  union all
  select 'c' as flow_out_id, 'd' as flow_in_id from dual
  union all
  select 'c' as flow_out_id, 'e' as flow_in_id from dual
)tmp;
drop table if exists TreeDepth_func_test_result;
create table TreeDepth_func_test_result
(
 node string,
 root string,
 depth bigint
);
```

对应的图结构如下图所示。



2. 查看训练结果。

```
0,0,0
1,0,1
2,0,1
3,0,2
4,0,2
5,0,2
6,0,3
a,a,0
b,a,1
c,a,1
d,a,2
e,a,2
```

3.9. 工具

本文为您介绍PAI-Studio提供的工具组件，包括SQL脚本和语义向量距离（双表）。

SQL脚本

您可以通过SQL脚本编辑器编写SQL语句，详细请参见SQL概述。

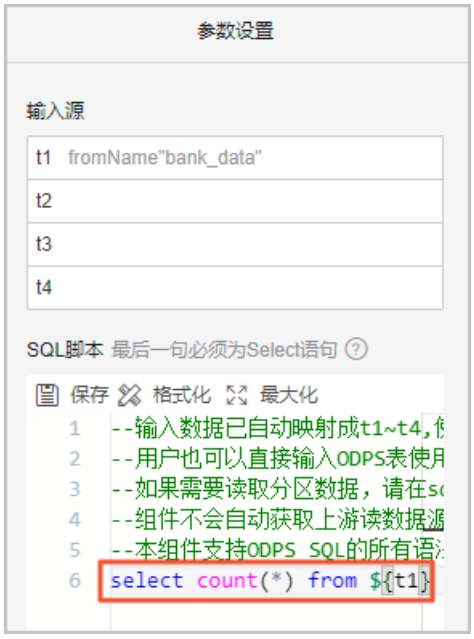
PAI-Studio仅支持通过可视化方式，配置SQL脚本组件的参数，页面参数如下所示。

参数	描述
输入源	展示上游输入的表名。
SQL脚本	待实现功能的SQL脚本。

SQL脚本实例

1. 将左侧组件栏中**源/目标**下的**读数据表**组件拖入画布中，并在右侧参数配置面板中配置具体的表名，如下图所示。

   - 如果输入表是分区表，则系统会自动选中**分区**复选框，用户可以选择或输入分区参数（系统仅支持输入单个分区）。如果未勾选分区复选框或勾选后未输入分区参数，均默认其输入为全表。
   - 如果输入表是非分区表，则分区复选框不可选中。
2. 向画布拖入**工具**下的**SQL脚本**组件，并与**读数据表**组件连接。
3. 单击画布中的**SQL脚本**，在页面右侧**参数设置**面板的**SQL脚本**区域，输入待实现功能的SQL脚本。

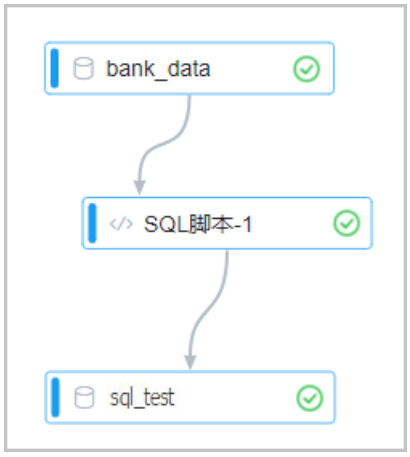


SQL脚本组件的说明如下：

- SQL脚本支持1～4个输入，1个输出。
- 输入数据表已自动映射成t1～t4，用户可以直接调用${t1}或${t2}，不用写入原表名。
- SQL脚本的中间可以执行任意的SQL语句，但是最后一句必须为Select语句。输出表内容为该Select语句的结果。
- 示例的SQL脚本用于统计输入表的行数。

4. 向画布中拖入**源/目标**下的**写数据表**组件，并在右侧的**表选择**面板输入**新表名**，系统会直接创建新表，如下图所示。如果需要写入分区表，则必须预先创建待写入的分区表。

5. 连接所有组件后，单击画布上方的**运行**，如下图所示。

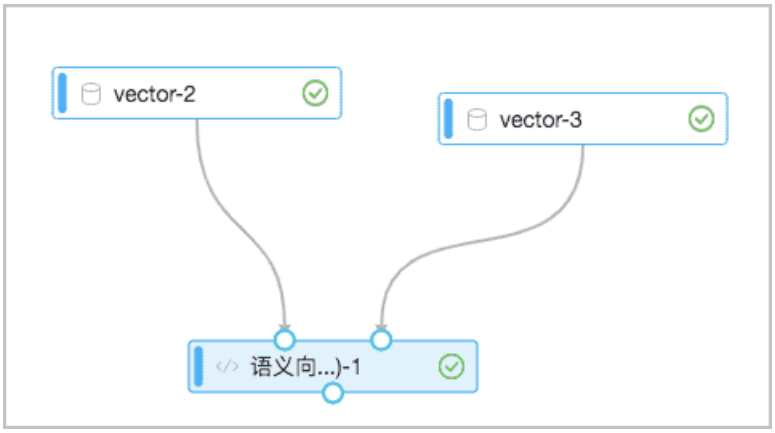


6. 运行完成后，右键单击画布中的**写数据表**组件，单击**查看数据**即可查看写入的数据。

语义向量距离（双表）

- 组件输入

  语意向量距离组件支持双表输入，两个输入桩分别为左侧的查询表和右侧的字典表，如下面所示。



  您可以通过可视化方式，配置语义向量距离（双表）组件的参数，页面参数如下所示。

页签	参数	描述
字段设置	**向量列**	向量数值，需要将整个向量写在一个字段中，每个数值以空格分割，如下图所示。 

页签	参数	描述
	ID列	作为每一列的主键。
参数设置	**距离计算方法**	支持**euclidean**和**cosine**距离计算方法。
	最终给出的相似度最大值的个数	取值为正整数。
执行调优	**计算的核心数**	计算使用的CPU Core数量，默认值为3。如果计算过程中出现OOM等情况，则适当增大**计算的核心数**和**每个核心的内存**。
	每个核心的内存	每个CPU Core的内存大小，单位为MB，默认值为2046 MB。如果计算过程中出现OOM等情况，则适当增大**计算的核心数**和**每个核心的内存**。

- 组件输出

  输出结果为查询表对应在字典表的TopN的距离和排序，如下图所示。

original_id ▲	near_id ▲	distance ▲	rank ▲
2	3	2.236067...	2
2	2	0	1
3	2	2.236067...	2
3	3	0	1
1	2	2.236067...	2
1	1	0	1

- 使用建议
  - 算法本质上是计算两个输入表的笛卡尔积距离并排序，因此建议数据量不超过千万样本。
  - **执行调优**中预设置的资源较小，如果出现OOM等情况，则需要手动调大资源配置。
  - 使用Cosine距离计算时，由于Double计算存在数据误差，因此可能出现极小的负数情况，属于正常现象。

3.10. 金融板块

本文为您介绍PAI-Studio提供的金融板块组件，包括分箱、数据转换模块、评分卡训练、评分卡预测及样本稳定指数（PSI）。

分箱

通过分箱组件可以进行特征离散化，即将连续的数据进行分段，使其变为多个离散化区间。分箱组件支持等频分箱、等宽分箱及自动分箱。PAI-Studio支持通过可视化或PAI命令方式，配置分箱组件的参数：

- 可视化方式

页签	参数	描述
	特征列	支持STRING、BIGINT及DOUBLE类型。
	标签列	仅支持二分类。

页签	参数	描述
字段设置		
	正例值	仅当**标签列**存在时才生效。
	选择分箱的参数来源	参数来源支持**“参数设置”中的参数**和**手动分箱或自定义JSON。**
	是否保留没有在“特征列”中选择的字段	使用自定义分箱时，如果该参数选择**是**，则未在**特征列**中选择的字段会原样保留，否则会删除未选择的字段。
	上传分箱和约束Json	当**选择分箱的参数来源**取值为**手动分箱或自定义JSON时，该参数生效。**
	追加分箱文件	如果此文件含有新的Feature，将会追加到原来的分箱结果中。如果是一样的Feature，将会以此文件中的为准。
参数设置	**分箱个数**	配置为10，表示将连续特征离散化至10个区间中。
	自定义列分箱个数	可以指定单个或多个字段的分箱数，会覆写总的分箱个数。如果自定义的列不在字段选择中，则多出的列也会进行计算。例如，字段选择为col0和col1， 自定义分箱为col0:3,col2:5，当nDivide为10时，那么按照col0:3,col1:10,col2:5进行计算。 取值格式为：字段名1:分箱个数, 字段名2:分箱个数。
	自定义离散值个数阈值	格式为col0:3。
	区间选择	支持**左开右闭**或**左闭右开**区间。
	分箱方式	支持**等频**、**等宽**及**自动分箱**。
	离散值个数阈值	如果小于该值，则分到“其它”分箱。
执行调优	**核心数**	默认系统自动分配。
	每个核分配的内存数	默认系统自动分配。

- PAI命令方式

```
PAI -name binning
  -project algo_public
  -DinputTableName=input
  -DoutputTableName=output
```

参数	描述	是否必选	默认值
inputTableName	输入表的名称。	是	无
outputTableName	输出表的名称。	是	无
selectedColNames	输入表选择分箱的列。	否	除Label外的其他列，如果无Label，则选择全部。
labelColumn	Label所在的列。	否	无
validTableName	表示binningMethod为auto时输入的验证表名。在auto模式下，该参数为必选。	否	空
validTablePartitions	验证表选择的分区。	否	全表
inputTablePartitions	输入表选择的分区。	否	全表
inputBinTableName	输入的分箱表。	否	无
selectedBinColNames	分箱表选择的列。	否	空
positiveLabel	输出正样本的分类。	否	1
nDivide	分箱的个数，取值为正整数。	否	10
colsNDivide	自定义列的分箱个数，例如col0:3,col2:5。如果colsNDivide中选中的列不在selectedColNames中，则多出的列也会进行计算。例如，selectedColNames为col0,col1，colsNDivide为col0:3,col2:5，nDivide为10时，则按照col0:3,col1:10,col2:5进行计算。	否	空
isLeftOpen	选择区间为左开右闭或左闭右开，取值包括为： ○ {true}：左开右闭。 ○ {false}：左闭右开。	否	true
stringThreshold	离散值为其他分箱的阈值。	否	无
colsStringThreshold	自定义列的阈值，同colsNDivide。	否	空
binningMethod	分箱类型，取值包括： ○ quantile：等频分箱。 ○ bucket：等宽分箱。 ○ auto：当在quantile模式时，自动选择单调性的分箱。	否	quantile
lifecycle	输出表的生命周期，取值为正整数。	否	无

参数	描述	是否必选	默认值
coreNum	核心数，取值为正整数。	否	系统自动计算
memSizePerCore	内存数，取值为正整数。	否	系统自动计算

分箱约束功能需要与评分卡训练组件配合使用。在评分卡训练过程中通过分箱进行特征工程，将特征离散化生成Dummy变量，并对训练过程中的每个Dummy变量的权重增加一定约束。各个约束项的含义如下：

- 顺序升序约束：该特征的各个Dummy变量按照Index从小到大添加权重上升的约束，即Index越大，权重越大。
- 顺序降序约束：该特征的各个Dummy变量按照Index从小到大添加权重下降的约束，即Index越大，权重越小。
- 相等权重值：该特征两个Dummy变量的权重值相等的约束。
- 权重值为0：该特征某个Dummy变量的权重值为0的约束。
- 等于固定权重值：该特征某个Dummy变量的权重值等于固定浮点数值的约束。
- WOE值顺序约束：该特征各个Dummy变量按照WOE值从小到大添加权重上升的约束，即WOE值越大，权重值越大。

数据转换模块

通过数据转换模块可以对数据进行归一化、离散化、Index化或WOE转换。PAI-Studio支持通过可视化或PAI命令方式，配置数据转换模块组件的参数：

- 可视化方式

页签	参数	描述
字段设置	**输入表选择的特征列**	输入的特征列，默认选择全表。
	不进行转换的数据列	选中的列会原样输出，可以在此指定Label。
	数据转换的类型	支持的转换类型包括**归一化**、**离散化**、**转换为WOE值**及**Index**。
	默认WOE值	仅当**数据转换的类型**取值为**转换为WOE值**时，该参数生效。 如果配置了该参数，当样本值落入无WOE值的分箱时，使用该值进行替换。如果未配置该参数，则当样本值落入无WOE值的分箱时，算法报错。
执行调优	**核心数**	使用的CPU Core数量，默认系统自动分配。
	每个核内存数	每个CPU Core所使用的内存大小，默认系统自动分配。

- PAI命令方式

```
PAI -name data_transform
-project algo_public
-DinputFeatureTableName=feature_table
-DinputBinTableName=bin_table
-DoutputTableName=output_table
-DmetaColNames=label
-DfeatureColNames=feaname1,feaname2
```

参数	描述	是否必选	默认值
inputFeatureTableName	输入特征数据表。	是	无
inputBinTableName	输入分箱结果表。	是	无
inputFeatureTablePartitions	输入特征表选择的分区。	否	全表
outputTableName	输出表。	是	无
featureColNames	输入表选择的特征列。	否	所有列
metaColNames	不进行转换的数据列，选中的列会原样输出。可以在此列中指定Label及sample_id等。	否	无
transformType	数据转换的类型，取值包括： ○ normalize：归一化。 ○ dummy：离散化。 ○ woe：转换为WOE值。	否	dummy
itemDelimiter	特征分隔符，仅在进行离散化时有效。	否	英文逗号（,）
kvDelimiter	KV分隔符，仅在进行离散化时有效。	否	英文冒号（:）
lifecycle	输出表的生命周期。	否	无
coreNum	使用的CPU Core数量。	否	系统自动计算
memSizePerCore	每个CPU Core所使用的内存大小，单位为MB。	否	系统自动计算

归一化是指根据输入的分箱信息，将变量值转换为0～1之间，缺失值填充为0。具体的算法如下。

```
if feature_raw_value == null or feature_raw_value == 0 then
  feature_norm_value = 0.0
else
  bin_index = FindBin(bin_table, feature_raw_value)
  bin_width = round(1.0 / bin_count * 1000) / 1000.0
  feature_norm_value = 1.0 - (bin_count - bin_index - 1) * bin_width
```

通过数据转换模块进行不同类型的数据转换，其输出格式不同：

- 归一化和WOE转换的输出为普通表。
- 离散化将数据转换成Dummy变量时，输出为KV格式的表，生成的变量格式为${feaname}]_bin_${bin_id}。以sns变量为例，生成的变量如下：
  - 如果sns落入第二个桶中，则生成的变量为[sns]_bin_2。
  - 如果sns为空，则落入空桶，生成的变量为[sns]_bin_null。
  - 如果sns不为空，且未落入任何一个已经定义的桶中，则落入else桶，生成的变量为[sns]_bin_else。

评分卡训练

评分卡是信用风险评估领域常用的建模工具，其原理是通过分箱输入将原始变量离散化后再使用线性模型（逻辑回归或线性回归等）进行模型训练，其中包含特征选择及分数转换等功能。同时也支持在训练过程中为变量添加约束条件。

> **说明** 如果未指定分箱输入，则评分卡训练过程完全等价于一般的逻辑回归或线性回归。

以下介绍评分卡训练过程中的相关概念：

- 特征工程

  评分卡与普通线性模型的最大区别在于进行线性模型训练之前会对数据进行一定的特征工程处理。本文中，评分卡提供了如下两种特征工程方法：

  - 先通过分箱组件将特征离散化，再将每个变量根据分箱结果进行One-Hot编码，分别生成N个Dummy变量（N为变量的分箱数量）。

    > **说明** 使用Dummy变量变换时，每个原始变量的Dummy变量之间可以设置相关的约束，详情请参见。

  - 先通过分箱组件将特征离散化，再进行WOE转换，即使用变量落入的分箱所对应的WOE值替换变量的原始值。

- 分数转换

  评分卡的信用评分等场景中，需要通过线性变换将预测得到的样本odds转换成分数，通常通过如下的线性变换实现。

  $$log(odds) = \sum_i w_i x_i = a * scaled\_score + b$$

  您可以通过如下三个参数指定线性变换关系：

  - scaledValue：给出一个分数的基准点。
  - odds：在给定的分数基准点处的odds值。
  - pdo（Point Double Odds）：表示分数增长多少分时，odds值增长到双倍。

  例如，scaledValue=800，odds=50，pdo=25，则表示指定了直线中的如下两点。

  ```
  log(50)=a×800+b
  log(100)=a×825+b
  ```

  解出a和b，对模型中的分数进行线性变换即可得到变换后的变量分。

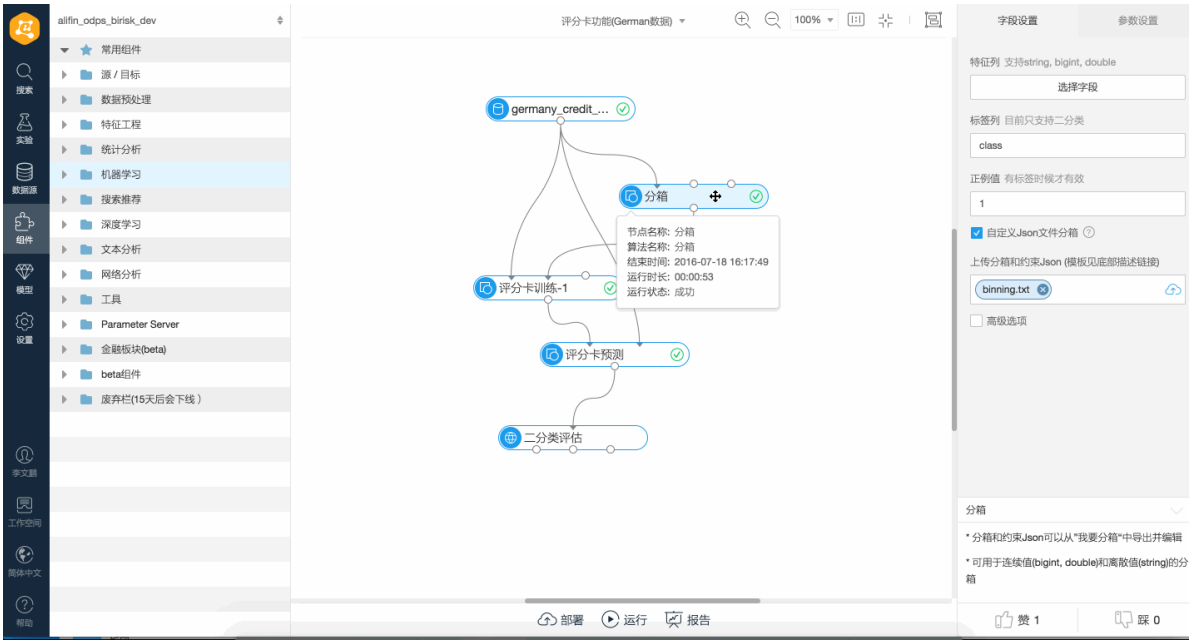
  Scaling信息由参数 `-Dscale` 指定，格式为JSON，示例如下。

```
{"scaledValue":800,"odds":50,"pdo":25}
```

当 `-Dscale` 参数不为空时，需要同时配置scaledValue、odds及pdo的值。

- 训练过程中增加约束

  评分卡训练过程支持对变量添加约束。例如指定某个bin所对应的分数为固定值，两个bin的分数满足一定比例，对bin之间的分数进行大小限制，或设置bin的分数按照bin的WOE值排序等。约束的实现依赖于底层带约束的优化算法，可以在分箱组件中通过可视化方式设置约束，设置完成后分箱组件会生成一个JSON格式的约束条件，并将其自动传递给下游连接的训练组件，详情请参见如下演示操作。



  系统支持如下六种JSON约束：

  - “<”：变量权重按照顺序满足升序的约束。
  - “>”：变量权重按照顺序满足降序的约束。
  - “=”：变量权重等于固定值。
  - “%”：变量之间的权重符定一定的比例关系。
  - “UP”：变量的权重约束上限。
  - “LO”：变量的权重约束下限。

  JSON约束以字符串的形式存储在单行单列（字符串类型）的表中，存储的JSON字符串示例如下。

```
{
  "name": "feature0",
  "<": [
    [0,1,2,3]
  ],
  ">": [
    [4,5,6]
  ],
  "=": [
    "3:0","4:0.25"
  ],
  "%": [
    ["6:1.0","7:1.0"]
  ]
}
```

- 内置约束

  每个原始变量都有一个隐含约束，无需用户指定，即单个变量人群的分数平均值为0。通过该约束，模型截距项的scaled_weight即为整个人群的平均分。

- 优化算法

  在高级选项中可以配置训练过程中使用的优化算法，系统支持如下四种优化算法：

  - L-BFGS：是一阶的优化算法，支持较大规模的特征数据级。该算法属于无约束的优化算法，会自动忽略约束条件。
  - Newton's Method：牛顿法是经典的二阶算法，收敛速度快，准确度高。但是由于需要计算二阶Hessian Matrix，因此不适用于较大特征规模。该算法属于无约束的优化算法，会自动忽略约束条件。
  - Barrier Method：二阶的优化算法，在没有约束条件的情况下完全等价于牛顿法。该算法的计算性能和准确性与SQP差别不大，通常建议选择SQP。
  - SQP

    二阶的优化算法，在没有约束条件的情况下完全等价于牛顿法。该算法的计算性能和准确性与Barrier Method差别不大，通常建议选择SQP。

  > **说明**
  > - L-BFGS和Newton's Method均属于无约束的优化算法，Barrier Method和SQP属于带约束的优化算法。
  > - 如果不了解优化算法，建议将优化算法配置为”自动选择“，系统会自动根据用户任务的数据规模和约束情况选择最合适的优化算法。

- 特征选择

  训练模块支持Stepwise特征选择功能。Stepwise是一种前向选择和后向选择的融合，即每次进行前向特征选择将一个新变量加入模型后，需要对已经进入模型的变量进行一次后向选择，以移除显著性不满足需求的变量。由于同时支持多种目标函数和多种特征变换方法，因此Stepwise特征选择过程支持如下多种选择标准：

- 边缘贡献（Marginal Contribution）：适用于所有目标函数和特征工程方法。

  模型A中不包含变量X，模型B包含所有A的变量，且包含变量X。两个训练模型最终收敛时所对应目标函数的差值，即为变量X在模型B中所有变量之间的边缘贡献度。在特征工程为Dummy变换的场景中，原始变量的X边缘贡献度定义为两个模型分别包含和不包含该变量的所有Dummy变量的目标函数之差。因此，使用边缘贡献度进行特征选择支持所有的特征工程方法。

  该方法的优点是比较灵活，不局限于某一种模型，直接选择使得目标函数更优的变量进入模型。缺点是边缘贡献度不同于统计显著性，统计显著性通常选择0.05为阈值，而边缘贡献度新用户没有一个绝对的概念阈值，建议将其设置为10E-5。

- 评分检验（Score Test）：仅支持WOE转换或无特征工程的逻辑回归选择。

  前向选择过程中，首先训练一个仅有截距项的模型，在之后的每一步迭代中，分别对未进入模型的变量计算其评分卡方统计量（Score Chi-Square），然后将评分卡方统计量最大的变量选入模型。同时，根据卡方分布计算该统计量所对应的显著性P Value。如果评分卡方统计量最大的变量其P Value大于用户指定的进入模型的最大显著性阈值（slentry），则不会将该变量纳入模型，并停止选择过程。

  完成一轮前向选择后，将对已经选中进入模型的变量进行一轮后向选择。后向选择过程中，对于已经进入模型中的变量分别计算其对应的沃尔德卡方统计量（Wald Chi-Square），并计算其对应的显著性P Value。如果P Value大于用户指定的移除模型的最大显著性阈值（slstay），则从模型中移除该变量，并继续进行下一轮迭代选择。

- F检验（F Test）：仅支持WOE转换或无特征工程的线性回归选择。

  前向选择过程中，首先训练一个仅有截距项的变量，在之后的每一步迭代中，分别对未进入模型的变量计算其F Value。F Value的计算与边缘贡献度的计算类似，需要训练两个模型以计算一个变量的F Value。F Value符合F分布，可以根据其F分布的概率密度函数求得其对应的显著性P Value。如果P Value大于用户指定的进入模型的最大显著性阈值（slentry），则不会将变量纳入模型，并停止选择过程。

  后向选择过程也是使用F Value计算显著性，其过程与评分检验类似。

- 强制选择加入模型的变量

  进行特征选择之前，可以设置强制进入模型的变量，被选中的变量不参与前向和后向的特征选择过程。无论选中的变量其显著性取值如何，都会直接进入模型。您可以在命令行中通过-Dselected参数指定迭代次数和显著性阈值，格式为JSON，示例如下。

  ```
  {"max_step":2, "slentry": 0.0001, "slstay": 0.0001}
  ```

  如果-Dselected参数为空或max_step为0，则表示正常的训练流程，不进行特征选择。

PAI-Studio支持通过可视化（详见评分卡训练示例）或PAI命令的方式配置评分卡训练组件的参数，使用PAI命令的方式如下。

```
pai -name=linear_model -project=algo_public
  -DinputTableName=input_data_table
  -DinputBinTableName=input_bin_table
  -DinputConstraintTableName=input_constraint_table
  -DoutputTableName=output_model_table
  -DlabelColName=label
  -DfeatureColNames=feaname1,feaname2
  -Doptimization=barrier_method
  -Dloss=logistic_regression
  -Dlifecycle=8
```

参数	描述	是否必选	默认值
inputTableName	输入特征数据表。	是	无
inputTablePartitions	输入特征表选择的分区。	否	全表
inputBinTableName	输入分箱结果表。如果该表指定，则先自动根据该表的分箱规则对原始特征进行离散化，再进行训练。	否	无
featureColNames	输入表选择的特征列。	否	选择全部，自动排除Label列。
labelColName	目标列。	是	无
outputTableName	输出模型表。	是	无
inputConstraintTableName	输入的JSON格式约束条件，存储在表的一个单元中。	否	无
optimization	优化类型，支持的类型包括： • lbfgs • newton • barrier_method • sqp • auto 仅sqp和barrier_method支持约束，auto即为根据用户数据和相关参数自动选择合适的优化算法。如果您对优化算法不太了解，建议使用auto。	否	auto
loss	Loss类型，支持logistic_regression和least_square类型。	否	logistic_regression
iterations	优化的最大迭代次数。	否	100
l1Weight	L1正则的参数权重，仅lbfgs优化算法支持L1 Weight。	否	0
l2Weight	L2正则的参数权重。	否	0
m	lbfgs优化过程中的历史长度，仅对lbfgs优化算法有效。	否	10
scale	评分卡对Weight进行Scale的信息。	否	空
selected	评分卡特征选择功能。	否	空
convergenceTolerance	收敛条件。	否	1e-6
positiveLabel	正样本的分类。	否	1
lifecycle	输出表的生命周期。	否	无

参数	描述	是否必选	默认值
coreNum	核心数。	否	系统自动计算
memSizePerCore	内存数，单位为MB。	否	系统自动计算

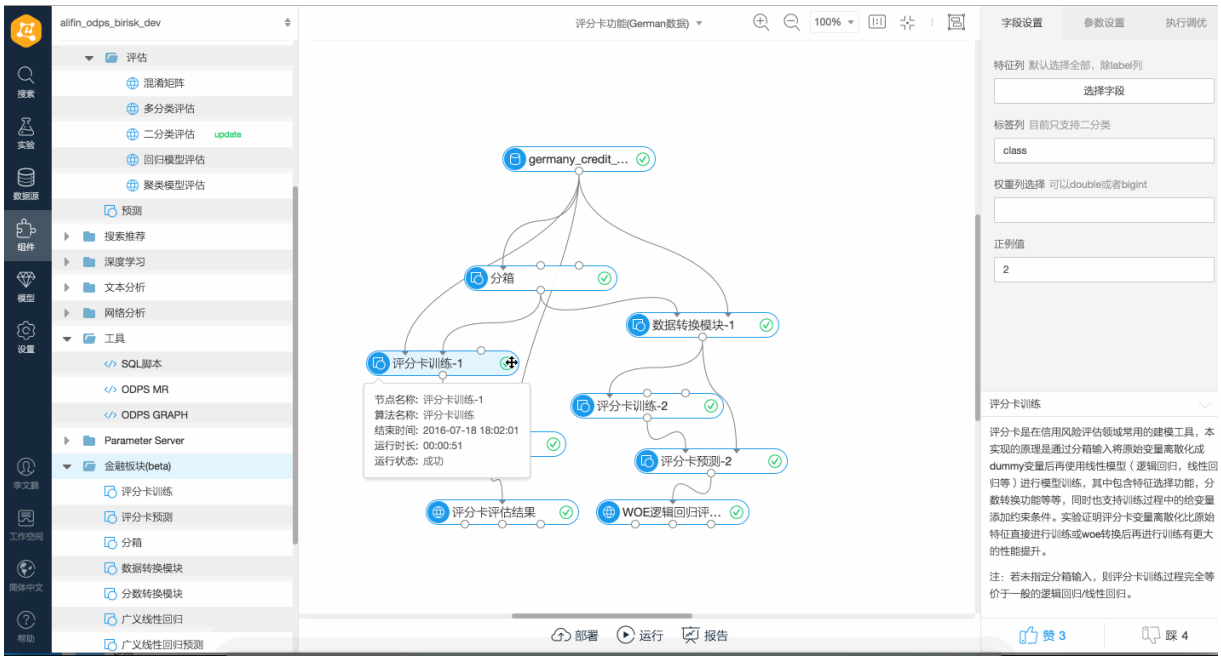
评分卡模型的输出为一个Model Report，其中包含了变量的分箱信息、分箱的约束信息、WOE及Marginal Contribution等基本的统计指标。PAI Web端展示的评分卡模型评估报告的相关列描述如下所示。

列名	列类型	描述
feaname	STRING	特征名称。
binid	BIGINT	分箱ID。
bin	STRING	分箱描述，用于表明该分箱的值域。
constraint	STRING	训练时增加到该分箱的约束条件。
weight	DOUBLE	训练完成后所对应的分箱变量权重，或未指定分箱输入的非评分卡模型，该项直接对应模型变量权重。
scaled_weight	DOUBLE	评分卡模型训练过程中指定分数转换信息后，将分箱变量权重经过线性变换得到的分数值。
woe	DOUBLE	统计指标：训练集上该分箱的WOE值。
contribution	DOUBLE	统计指标：训练集上该分箱的Marginal Contribution值。
total	BIGINT	统计指标：训练集上该分箱的总样本数。
positive	BIGINT	统计指标：训练集上该分箱的正样本数。
negative	BIGINT	统计指标：训练集上该分箱的负样本数。
percentage_pos	DOUBLE	统计指标：训练集上该分箱的正样本数占总正样本的比例。
percentage_neg	DOUBLE	统计指标：训练集上该分箱的负样本数占总负样本的比例。
test_woe	DOUBLE	统计指标：测试集上该分箱的WOE值。
test_contribution	DOUBLE	统计指标：测试集上该分箱的Marginal Contribution值。
test_total	BIGINT	统计指标：测试集上该分箱的总样本数。
test_positive	BIGINT	统计指标：测试集上该分箱的正样本数。
test_negative	BIGINT	统计指标：测试集上该分箱的负样本数。

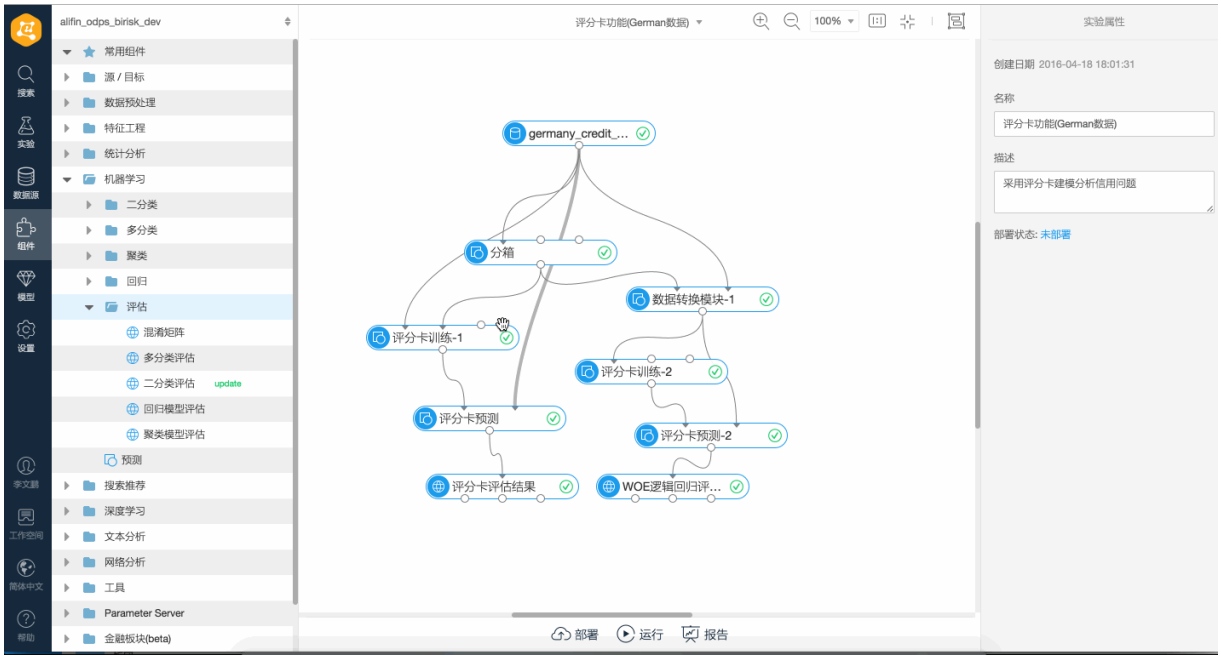
列名	列类型	描述
test_percentage_pos	DOUBLE	统计指标：测试集上该分箱的正样本数占总正样本的比例。
test_percentage_neg	DOUBLE	统计指标：测试集上该分箱的负样本数占总负样本的比例。

评分卡训练示例

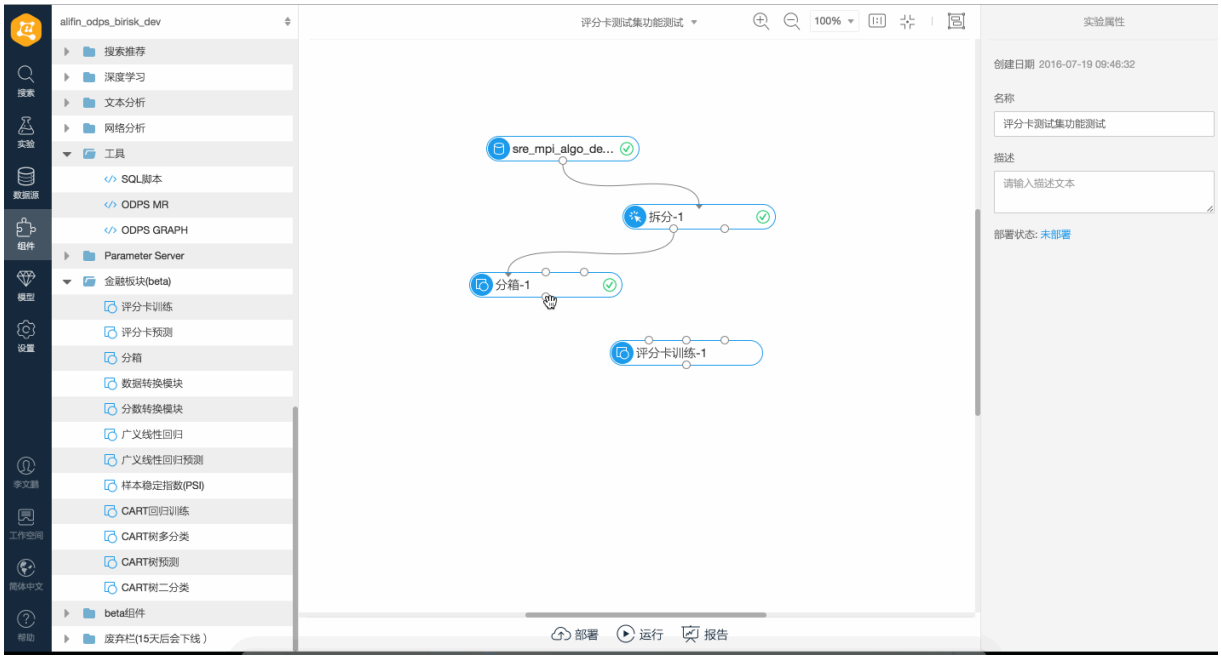
推荐通过PAI Web（可视化方式）使用评分卡训练组件，如下是一个简单的评分卡Stepwise特征选择、特征WOE变换及逻辑回归Stepwise特征选择的对比演示。



如下是一个简单的评分卡训练、特征WOE变换及逻辑回归的对比演示。



如果输入训练组件中连接测试集，则输出的模型报告中会同时输出模型在测试集上的统计指标，例如WOE及MC等。如下是一个简单的带测试集的训练演示。



评分卡预测

评分卡预测组件对原始数据根据评分卡训练组件产出的模型结果进行预测打分，PAI-Studio支持通过可视化或PAI命令配置评分卡预测组件的参数：

- 可视化方式

页签	参数	描述
字段设置	**特征列**	选择用于预测的原始特征列，默认选择全部。
	原样添加到结果表	选择不进行任何处理，直接附加到预测结果表中的列。例如ID列和目标列等。
	输出变量分	是否输出每个特征变量所对应的分数，最终的预测总得分为截距项的得分加所有的变量分。
执行调优	**核数目**	使用的CPU Core数量，默认系统自动分配。
	每个核的内存大小	每个CPU Core所用的内存大小，默认系统自动分配。

- PAI命令方式

```
pai -name=lm_predict
  -project=algo_public
  -DinputFeatureTableName=input_data_table
  -DinputModelTableName=input_model_table
  -DmetaColNames=sample_key,label
  -DfeatureColNames=fea1,fea2
  -DoutputTableName=output_score_table
```

参数	描述	是否必选	默认值
inputFeatureTableName	输入特征数据表。	是	无
inputFeatureTablePartitions	输入特征表选择的分区。	否	全表
inputModelTableName	输入的模型表。	是	无
featureColNames	输入表选择的特征列。	否	所有列
metaColNames	不进行转换的数据列，选中的列会原样输出。可以在此指定Lable和sample_id等。	否	无
outputFeatureScore	预测结果中是否输出变量分，取值包括： ○ true：输出变量分。 ○ false：不输出变量分。	否	false
outputTableName	输出预测结果表。	是	无
lifecycle	输出表的生命周期。	否	无
coreNum	核心数。	否	默认自动计算
memSizePerCore	内存大小，单位为MB。	否	默认自动计算

评分卡预测组件输出的打分表示例如下。

churn ▲	prediction_score ▲	prediction_prob ▲	prediction_detail ▲
1	-2.152581613419945	0.10409022740823...	{"0":0.8959097726,"1":0.1040902274}
1	0.40321295914989297	0.599459362718963	{"0":0.4005406373,"1":0.5994593627}
0	-5.9448781609701316	0.00261237905306...	{"0":0.9973876209,"1":0.0026123791}
1	-1.4235254136279643	0.19410950481015...	{"0":0.8058904952,"1":0.1941095048}
0	-0.4354127766052662	0.3928345539282477	{"0":0.6071654461,"1":0.3928345539}
0	-2.322642905577369	0.08926496638122...	{"0":0.9107350336,"1":0.0892649664}
1	-1.8095060182152187	0.14069783849895...	{"0":0.8593021615,"1":0.1406978385}
0	0.09435077042706097	0.5235702098997645	{"0":0.4764297901,"1":0.5235702099}
0	-2.460605112010114	0.0786664686456529	{"0":0.9213335314,"1":0.0786664686}

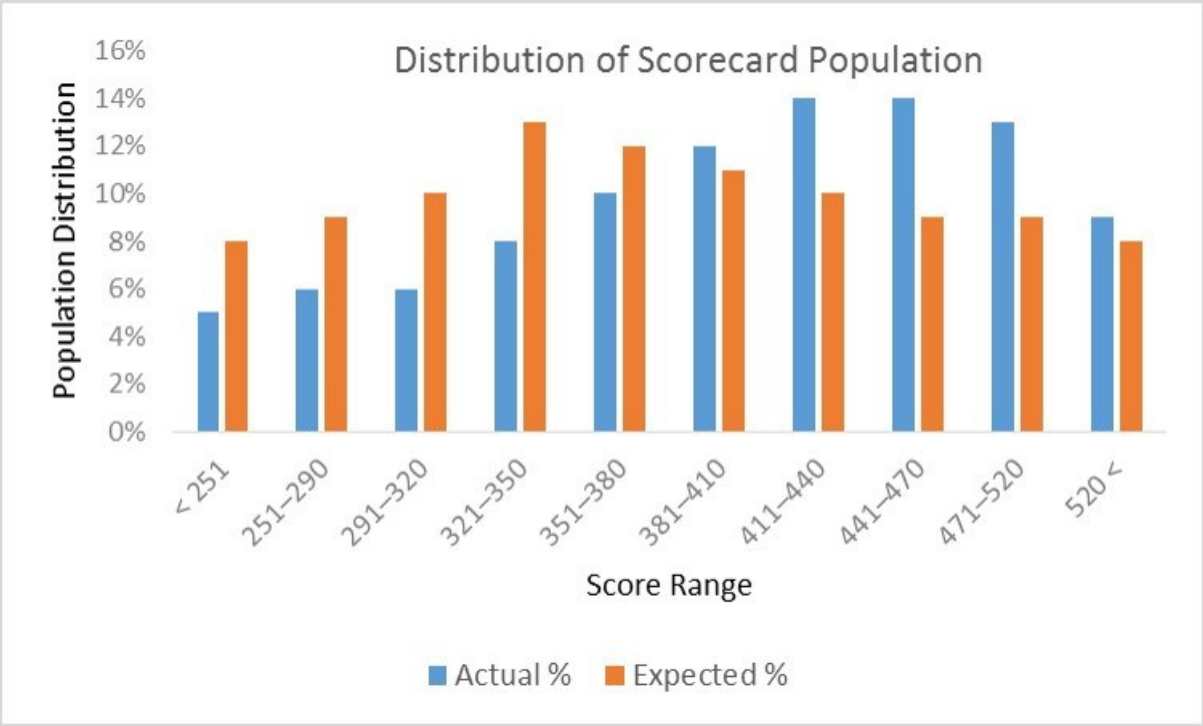
其中churn列为用户选择的原样添加到结果表中的列，与预测结果无关。其它三列为预测结果列，其含义如下表所示。

列名	列类型	列描述
prediction_score	DOUBLE	预测分数列。线性模型中特征值和模型权重值直接相乘相加的结果，对应到评分卡模型中，如果模型进行了分数转换，则该分数输出转换后的得分。
prediction_prob	DOUBLE	二分类场景中预测得到的正例概率值，原始得分（未经分数转换）经过Sigmoid变换后得到该值。
prediction_detail	STRING	用JSON格式描述的各类别概率值，其中0表示负类，1表示正类。例如{“0”:0.1813110520,”1”:0.8186889480}。

样本稳定指数（PSI）

样本稳定指数（PSI）是衡量样本变化所产生的偏移量的一种重要指标，通常用于衡量样本的稳定程度。例如样本在两个月份之间的变化是否稳定，如果变量的PSI值小于0.1，则表示变化不太显著。如果PSI值在0.1到0.25之间，则表示有比较显著的变化。如果PSI值大于0.25，则表示变量变化比较剧烈，需要特殊关注。

通过画图的方法可以衡量样本在不同时刻的稳定性，即将待比较的变量离散化成N个分箱，然后计算样本分别在各个分箱中的数量及比例，并以柱状图的形式呈现出来，如下图所示。



该方法可以直观地查看某个变量在两批样本上是否有剧烈的变化，但是无法量化，从而无法实现对样本稳定性的自动监控。因此PSI就显得尤为重要，PSI的计算公式如下。

$$PSI = \sum((Actual\ \% - Expected\ \%) \times (ln(\frac{Actual\ \%}{Expected\ \%}))$$

PAI-Studio支持通过可视化或PAI命令方式，配置样本稳定指数（PSI）组件的参数：

- 可视化方式

页签	参数	描述
字段设置	**要计算PSI指标的特征**	需要进行PSI指标计算的特征列。
执行调优	**核心数**	使用的CPU Core数量，默认系统自动分配。
	内存数	每个CPU Core使用的内存大小，默认系统自动分配。

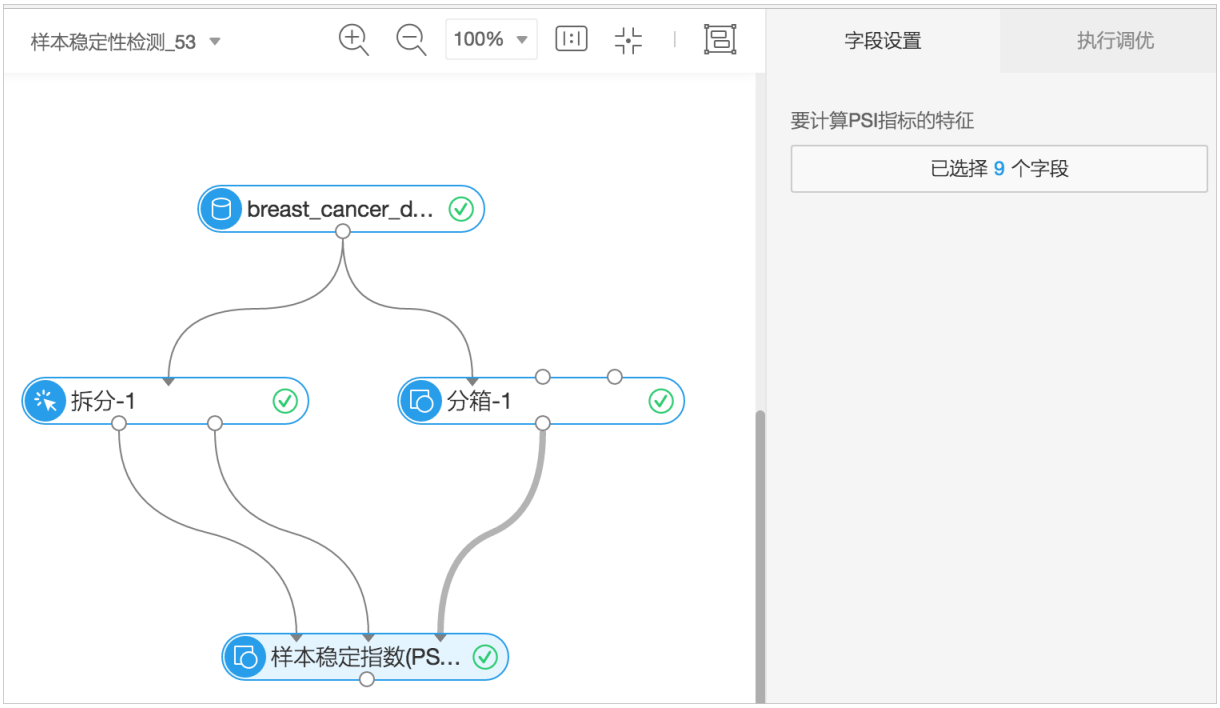
- PAI命令方式

```
PAI -name psi
-project algo_public
-DinputBaseTableName=psi_base_table
-DinputTestTableName=psi_test_table
-DoutputTableName=psi_bin_table
-DinputBinTableName=pai_index_table
-DfeatureColNames=fea1,fea2,fea3
-Dlifecycle=7
```

参数	描述	是否必选	默认值
inputBaseTableName	输入基础表表名，计算测试表在基础表的基础上产生的偏移量。	是	无
inputBaseTablePartitions	输入基础表分区。	否	全表
inputTestTableName	输入测试表的名称，计算测试表在基础表的基础上产生的偏移量。	是	无
inputTestTablePartitions	输入测试表分区。	否	全表
inputBinTableName	输入分箱结果表的名称。	是	无
featureColNames	需要计算PSI指标的特征列。	否	全表
outputTableName	输出的指标表。	是	无
lifecycle	输出表的生命周期。	否	无
coreNum	使用的CPU Core数量。	否	系统自动分配
memSizePerCore	每个CPU Core使用的内存大小，单位为MB。	否	系统自动分配

样本稳定指数（PSI）示例

使用PSI之前需要对特征数据进行分箱，因此需要一个分箱组件。如下图使用的示例，PSI组件分别连接待比较的两个样本数据集，再连接一个分箱组件。只需要配置**要计算PSI指标的特征**，即可进行PSI计算。



PSI计算的结果如下图所示。

查看模型报告 - pai_temp_10478_187051_1

收起 展开

Feature	Bin	Test %	Base %	Test - Base	ln(Test/Base)	PSI
age	-	-	-	-	-	0.0475
--------	(-inf,1]	21.35	19.35	1.99	0.0979	0.0019
--------	(1,3]	23.1	21.99	1.11	0.049	0.0005
--------	(3,4]	10.23	12.9	-2.67	-0.2318	0.0062
--------	(4,5]	16.37	21.11	-4.74	-0.2542	0.0121
--------	(5,7]	7.89	8.5	-0.61	-0.0744	0.0005
--------	(7,9]	10.82	6.16	4.66	0.5635	0.0263
--------	(9,+inf)	10.23	9.97	0.26	0.0261	0.0001
menopause	-	-	-	-	-	0.0588
--------	(-inf,1]	56.14	53.08	3.06	0.0561	0.0017
--------	(1,2]	4.68	8.5	-3.83	-0.5976	0.0229
--------	(2,4]	15.2	11.14	4.06	0.3107	0.0126
--------	(4,6]	9.06	7.04	2.03	0.253	0.0051
--------	(6,9]	6.73	8.8	-2.07	-0.2686	0.0056
--------	(9,+inf)	8.19	11.44	-3.25	-0.3343	0.0109
tumor_size	-	-	-	-	-	0.0409

4.AutoML自动机器学习

4.1. Auto ML自动调参指南

本文为您介绍Auto ML自动调参的算法介绍及操作流程。

操作步骤

1. 登录PAI控制台。
2. 单击左侧导航栏的**实验**并选择某个实验。

   本文以**雾霾天气预测**实验为例。
3. 在实验画布区，单击左上角的**Auto ML > 模型自动调参**。
4. 在**自动调参**配置页面，选择需要调参的算法，单击**下一步**。

> **说明** 一个实验中有多个算法时请单选一个算法。

5. 在**调参配置**模块，选择**调参方式**，完成后单击**下一步**。阿里云机器学习提供如下调参方式供您选择：
   - EVOLUTIONARY_OPTIMIZER
     a. 随机选定a个参数候选集（**探索样本数**a）。
     b. 取其中评估指标较高的n个参数候选集，作为下一轮迭代的参数候选集。
     c. 继续在这些参数周边的r倍（**收敛系数**r）标准差范围探索，以探索出新的参数集，来替代上一轮中评估指标靠后的a-n个参数集。
     d. 根据以上逻辑，迭代m轮（**探索次数**m），直到找到最优的参数集合。

   根据如上原理，最终产生的模型数目为**a+(a-n)*m** 。

> 注意　n的第一个值为a/2-1，在迭代过程中默认为n/2-1（小数向上取整）。

- 数据拆分比例：将输入数据源分为训练集和评估集。0.7表示70%的数据用于训练模型，30%用于评估。
- 探索样本数：每轮迭代的参数集个数，个数越多越准，计算量越大，取值范围为5～30。
- 探索次数：迭代次数，次数越多探索越准、计算量越大，取值范围为1～10。
- 收敛系数：调节探索范围，越小收敛越快，但是可能会错过适合的参数，取值范围为0.1～1。
- 自定义范围：输入每个参数的调节范围，如果未改变当前参数范围，则此参数按照默认值代入，不参与自动调参。

- RANDOM_SEARCH
  a. 每个参数在所在范围内随机选取一个值。
  b. 将这些值组成一组参数进行模型训练。
  c. 如此进行m轮（**迭代次数**），训练产生m个模型并进行排序。

- 迭代次数：表示在所配置的区间的搜索次数，取值范围为2～50。
- 数据拆分比例：将输入数据源分为训练集和评估集，0.7表示70%的数据用于训练模型，30%用于评估。
- 自定义范围：输入每个参数的调节范围，如果未改变当前参数范围，则此参数按照默认值代入，不参与自动调参。

- GRID_SEARCH
  a. 将每个参数的取值区间拆成n段（**网格拆分数**）。
  b. 在n段里面各随机取出一个随机值。假设有m个参数，就可以组合出n^m组参数。
  c. 根据n^m组参数训练生成n^m个模型并进行排序。

- 网格拆分数：表示拆分出的grid个数，取值2～10。
- 数据拆分比例：将输入数据源分为训练集和评估集，0.7表示70%的数据用于训练模型，30%用于评估。
- 自定义范围：输入每个参数的调节范围，如果未改变当前参数范围，则此参数按照默认值代入，不参与自动调参。

- UserDefine

自定义范围：系统对您枚举的参数取值范围进行全部组合尝试并打分，如果未输入按照默认参数执行。

说明 在2.0版本中调参算法种类从4个增加到7个，各算法详细说明如下：

算法名称	说明
GAUSE（高斯算法）	高斯过程，是一种非参数贝叶斯模型。作为经典算法，高斯过程已经广泛地被应用于超参优化领域。通过不断观测超参配置表现来拟合代理模型，再通过模型的预测能力强化决策，从而能在有限的尝试次数中更有目的地选出合适的超参结果。
SAMPLE（采样算法）	该算法是PAI团队与达摩院合作自研的算法。对于数据量巨大的实验，仅需要一部分数据，就可以对一组超参所能得到的最终结果作预估。采样算法利用这一特性，结合PBT算法的思想，在增加超参选取数量的同时，逐步提高采样比例，不仅能进行更广的探索也能获得更快的加速。
EVOLUTIONARY_OPTIMIZER（进化式调参方法）	该算法是PAI团队基于PBT理论自研的算法，EVOLUTIONARY_OPTIMIZER算法将调参问题看成一个多轮迭代循序渐进探索最优解的问题。其中”探索样本数“表示每轮迭代的样本，”探索次数“表示迭代轮数，”收敛系数“控制每次迭代的步长。在迭代过程中EVOLUTIONARY_OPTIMIZER会在每轮结束后抛弃效果不理想的探索样本，并在效果更优的探索样本集合中向外拓展更多探索样本，形成下一轮的计算探索样本集合。以此方式迭代，直到完成迭代轮数。
PBT（Population-based training）	PBT是一类基于种群概念的演化算法。它把超参配置看为一个种群，将搜索过程作为一个动态环境，在不断的迭代中对超参配置们进行优胜劣汰的筛选，最终得到表现更好的结果。这类算法概念简洁，可以适应不同的数据结构，在深度学习模型训练中取得过较好效果。
GRID_SEARCH	网格搜索调参法，将每个参与调参的参数按照比例等分切割，并且将切割后的参数随机组合生成参数数候选集进行计算和对比。
RANDOM_SEARCH	随机搜索调参法，在每个参数空间中随机采样并且组合形成参数候选集，并对候选集进行计算和对比。
UserDefine	用户自定义参数组合。

6. 在**调参模型输出选择**模块，配置模型输出参数，完成后单击**下一步**。

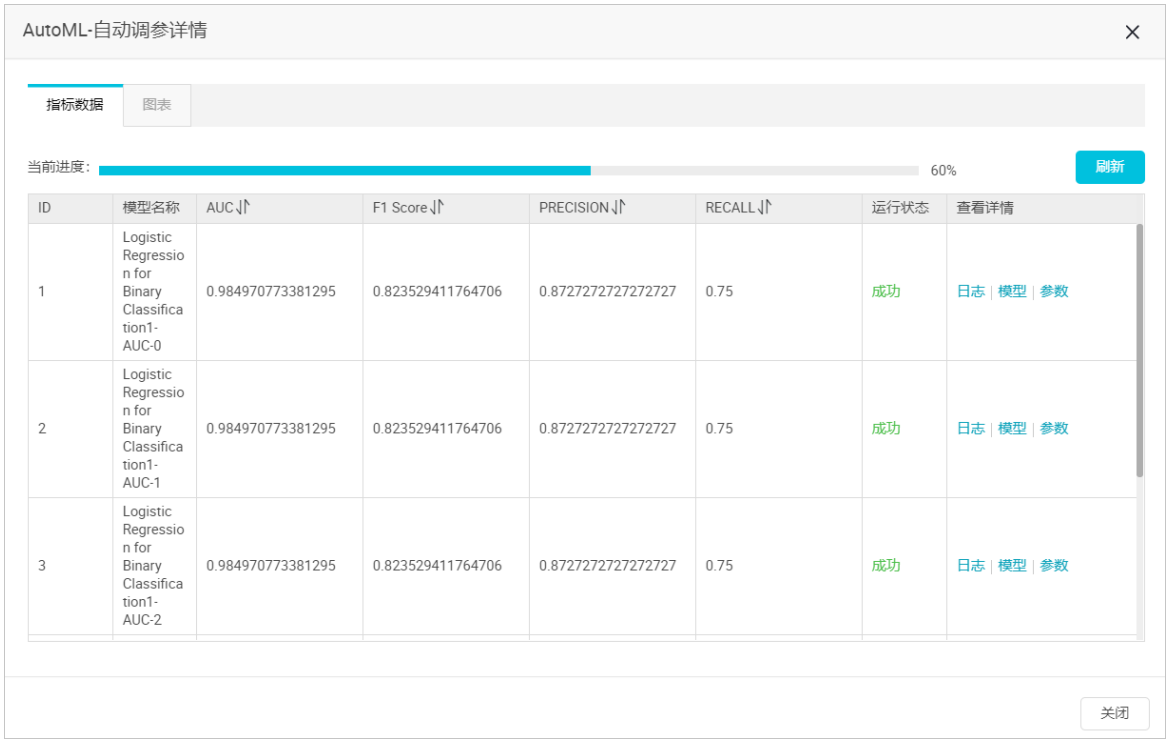
- 评估标准：可选择 **AUC**、**F1-score**、**Precision**、**Recall**四个维度中的一个作为评估标准。
- 保存模型数量：取值范围为1～5。根据所选择的评估标准，对模型进行排名，最终保存排名靠前的几个模型，数量对应您所选择的**保存模型数量**。
- 模型是否向下传导：默认打开。如果开关关闭，则将当前组件的默认参数生成的模型，向下传导至后续组件节点；如果开关打开，则将自动调参生成的最优模型，向下传导至后续组件节点。

7. 配置完成后，单击画布左上角的**运行**。此时画布上的对应算法已打开**Auto ML** 开关，后续也可以选择打开或关闭此开关。
8. （可选）鼠标右键单击画布模型组件，选择**编辑AutoML参数**，修改AutoML配置参数。

执行结果

输出模型：

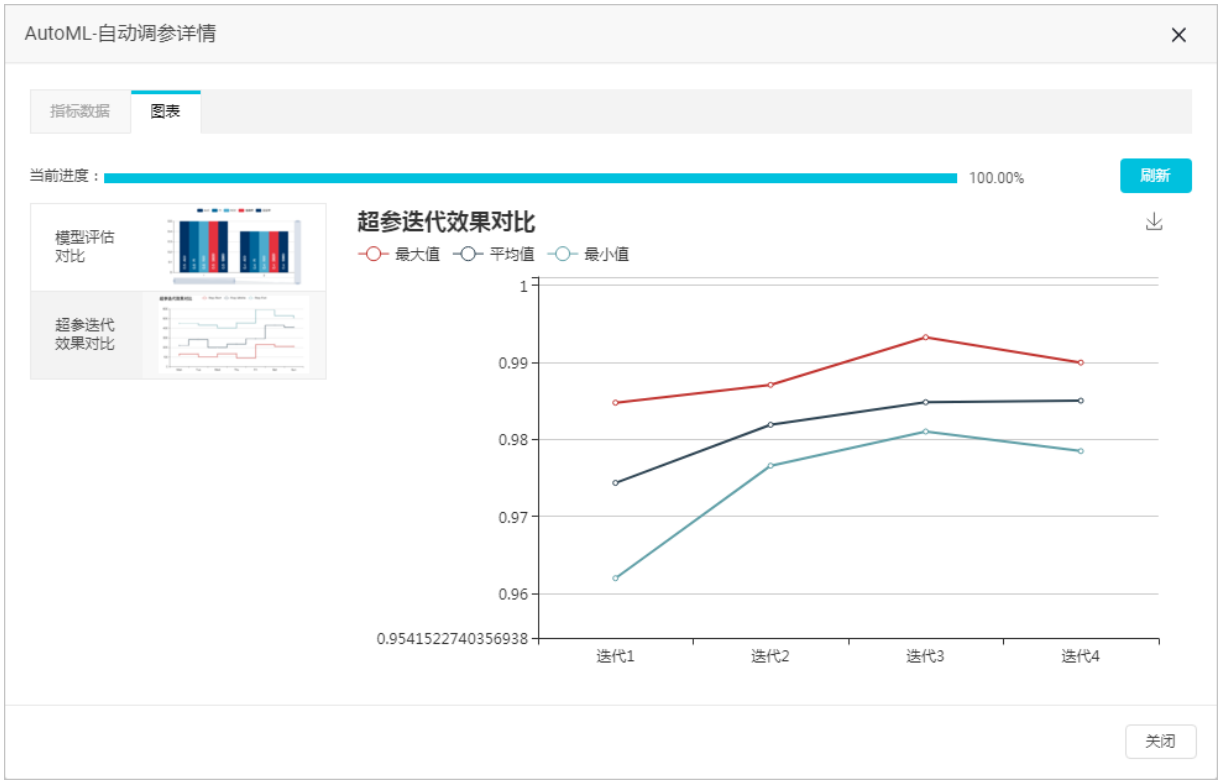
1. 在调参过程中，鼠标右键单击目标模型组件，选择**调参运行详情**。
2. 在 **AutoML-自动调参详情**页面，单击**指标数据**，查看当前调参的进度、各模型的运行状态等信息。



3. 根据候选模型的指标列表（**AUC**、**F1-score**、**准确率**、**召回率**）进行排序。
4. 在**查看详情**列单击**日志**或**参数**，查看每一个候选模型的日志及参数。

调参效果展示：

您可以通过**超参迭代效果对比**，查看每一轮参数更新后评估指标增长的趋势。



模型存储：

1. 选择左侧导航栏的**模型**。
2. 单击**实验模型**，打开**实验模型**文件夹。
3. 单击打开对应实验文件夹，查看Auto ML保存的模型。

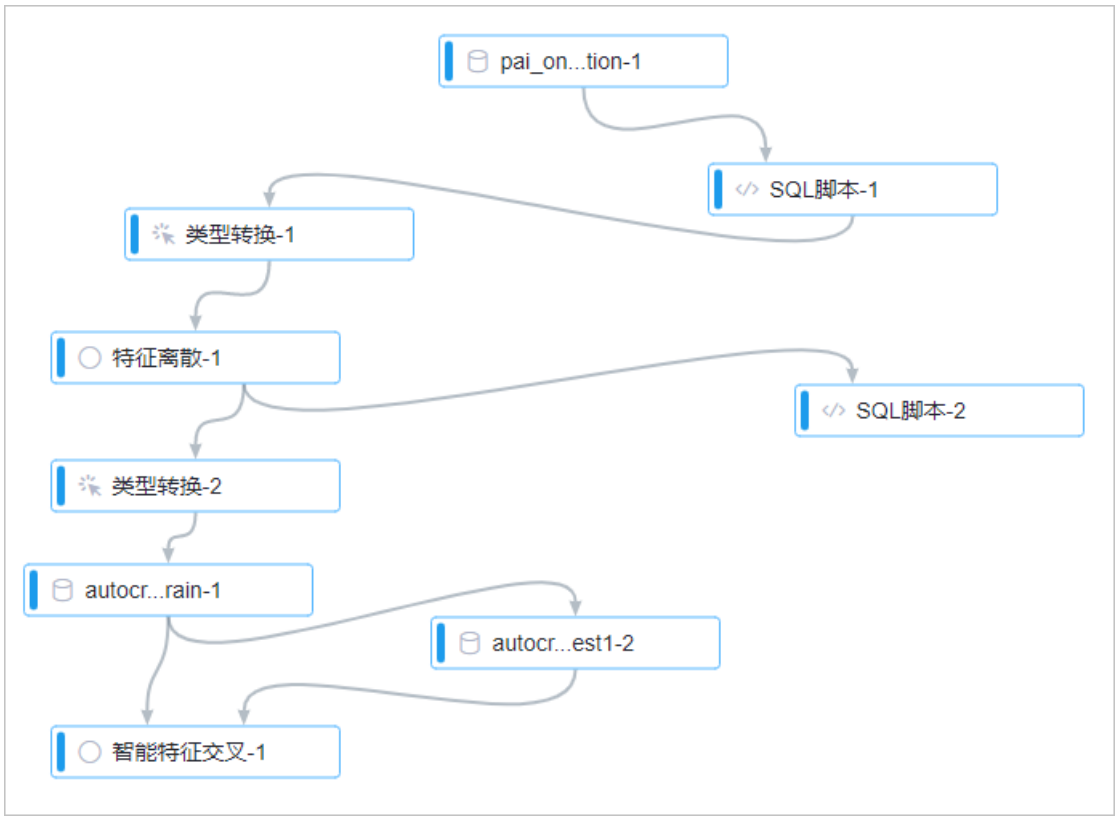
4.2. Auto ML自动特征工程使用说明

特征工程是在做机器学习训练的过程中必不可少的环节，特征工程就是找出对模型结果有益的特征交叉关系，通常特征工程需要耗费算法工程师大量的精力去尝试。针对这样的场景，PAI推出智能特征交叉组件，基于该组件可以帮助您锁定哪些特征的交叉是有意义的。本文介绍智能特征交叉组件的使用方法。

流程图

智能特征交叉基于深度学习框架TensorFlow开发，底层有大量并行化计算的工作，需要使用GPU。目前只有北京和上海两个区域支持该功能。

总流程图：



说明　使用首页的模板列表创建项目时，需要修改智能特征交叉组件的模型输出路径为您自己账号的OSS地址。

1.开通GPU和OSS访问权限

1. 登录PAI控制台。
2. 单击左侧导航栏设置，在基本设置处开通GPU和OSS访问权限。

2.数据分桶

目前智能特征交叉组件只支持BIGINT型的数据交叉，考虑到平时业务中的原始数据通常是如下图所示的Double类型：

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slop	ca	thal	status	style
63.0	male	an...	145.0	233.0	true	hyp	150.0	fal	2.3	down	0.0	fix	buff	H
67.0	male	as...	160.0	286.0	fal	hyp	108.0	true	1.5	flat	3.0	norm	sick	S2
67.0	male	as...	120.0	229.0	fal	hyp	129.0	true	2.6	flat	2.0	rev	sick	S1
37.0	male	not...	130.0	250.0	fal	norm	187.0	fal	3.5	down	0.0	norm	buff	H
41.0	fem	ab...	130.0	204.0	fal	hyp	172.0	fal	1.4	up	0.0	norm	buff	H
56.0	male	ab...	120.0	236.0	fal	norm	178.0	fal	0.8	up	0.0	norm	buff	H
62.0	fem	as...	140.0	268.0	fal	hyp	160.0	fal	3.6	down	2.0	norm	sick	S3
57.0	fem	as...	120.0	354.0	fal	norm	163.0	true	0.6	up	0.0	norm	buff	H
63.0	male	as...	130.0	254.0	fal	hyp	147.0	fal	1.4	flat	1.0	rev	sick	S2
53.0	male	as...	140.0	203.0	true	hyp	155.0	true	3.1	down	0.0	rev	sick	S1
57.0	male	as...	140.0	192.0	fal	norm	148.0	fal	0.4	flat	0.0	fix	buff	H

所以使用SQL组件或Onehot组件将字符型数据转为BIGINT型，另外需要使用特征离散组件进行特征分桶，将不同区间的特征按照分布的不同划分到不同的数据桶之中。数据经过分桶之后变为如下形态：

sex	cp	fbs	restecg	exang	slop	thal	age	trestbps	chol	thalach	oldpeak	ca	ifhealth
4	1	4	4	1	4	3	4	3	2	4	2	1	0
4	4	1	4	4	3	1	4	4	2	2	2	4	1
4	4	1	4	4	3	4	4	2	2	3	3	4	1
4	3	1	1	1	4	1	1	2	2	4	3	1	0
1	4	1	4	1	1	1	2	2	1	4	2	1	0
4	4	1	1	1	1	1	3	2	2	4	1	1	0
1	4	1	4	1	4	1	4	3	2	4	3	4	1
1	4	1	1	4	1	1	3	2	3	4	1	1	0
4	4	1	4	1	3	4	4	2	2	3	2	2	1
4	4	4	4	4	4	4	3	3	1	4	3	1	1
4	4	1	1	1	3	3	3	3	1	3	1	1	0
1	4	1	4	1	3	1	3	3	2	4	2	1	0

3.确定特征范围

特征交叉的基本原理是将特征先按照向量空间展开，然后做特征间的相互交叉验证，最终挑选出合理的特征组合方式。在计算之前需要知道每个特征的空间的最大值，如下面这组数据：

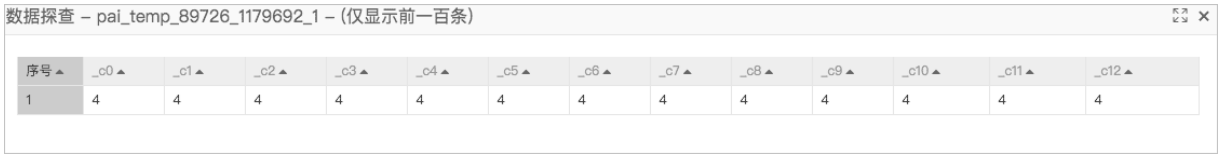
- thalach的特征最大值为4
- oldpeak的特征最大值为3
- ca的特征最大值为4

thalach	oldpeak	ca
4	2	1
2	2	4
3	3	4

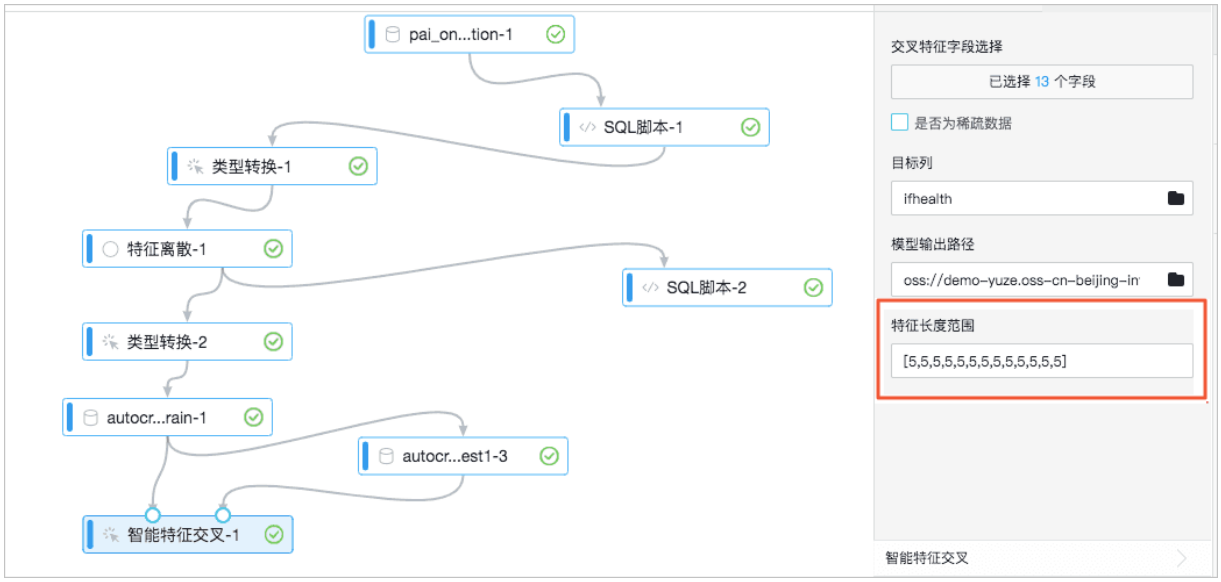
执行如下SQL语句获取最大值。

```
select max(feature) from table;
```

在本实验样例数据中，所有分桶完的特征的最大值均为4。



于是智能特征交叉的**特征长度范围**要写成如下图样式。其中5表示开区间[0,5)，包含4。



4.生成训练和测试数据

本实验使用的训练数据和测试数据是相同的表，实际使用中也可以把测试数据替换成跟训练数据字段相同的不同表。

5.智能特征交叉

- 字段设置

  输入桩左侧是训练数据，右侧是测试数据。

  - 交叉特征字段选择：选择需要交叉验证的特征字段。

- 是否为稀疏数据：默认不选中。表示稠密数据。
- 目标列：选择目标列字段。
- 模型输出路径：生成的模型存于您的OSS中。

- 参数设置

- 遍历次数：迭代次数。
- 特征阶数：指特征交叉阶数。如3，表示结果最多计算出3个特征之间的交叉。

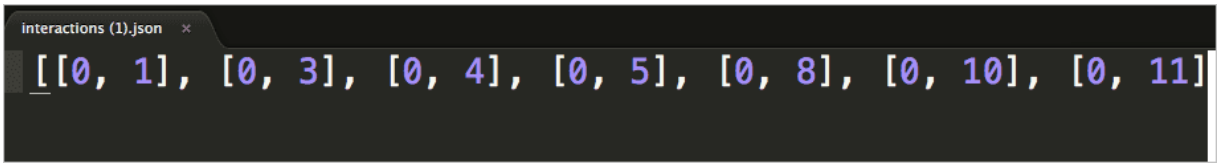
PAI命令：

```
PAI -name fives_ext -project algo_public
  -DlabelColName="ifhealth"  //目标列
  -Dmetric_file="metric_log.log" //日志
  -Dfeature_meta="[5,5,5,5,5,5,5,5,5,5,5,5,5]"
  -DtrainTable="odps://项目名/tables/表名"
  -Dbuckets="oss://{oss_bucket}/"
  -Dthreshold="0.5"
  -Dk="3"
  -DossHost="oss-cn-beijing-internal.aliyuncs.com" //区域
  -Demb_dims="16"
  -DenableSparse="0"
  -Dtemp_anneal_steps="30000"
  -DfeatureColName="sex,cp,fbs,restecg,exang,slop,thal,age,trestbps,chol,thalach,oldpeak,ca"  //特征
  -DtestTable="odps://项目名/tables/表名"
  -Darn="acs:ram::********:role/aliyunodpspaidefaultrole"  //rolearn
  -Depochs="1500"
  -DcheckpointDir="oss://{oss_bucket}/{path}/";
```

结果查询

在OSS根目录下，亦即Dbuckets路径下找到interactions.json文件。

文件显示的是各种组合的关系：



结果显示的是一些启发性的特征交叉组合方式，您可以按照该特征进行特征组合，举例如下：

- [0,1]代表着第一个特征和第二个特征组合会有效果，特征顺序跟输入表的特征顺序一致。
- [8, 6, 5]代表一个三阶特征组合关系。第七个、第五个、第四个这三个特征组合起来会有效果。

5.PAI Studio-Notebook使用文档

机器学习为您提供Notebook功能，支持在线编辑并调试TensorFlow代码。目前线上Notebook已经支持与OSS存储服务以及机器学习底层计算集群互通，轻松实现云端的代码调试工作。

背景信息

PAI Studio-Notebook的使用限制如下：

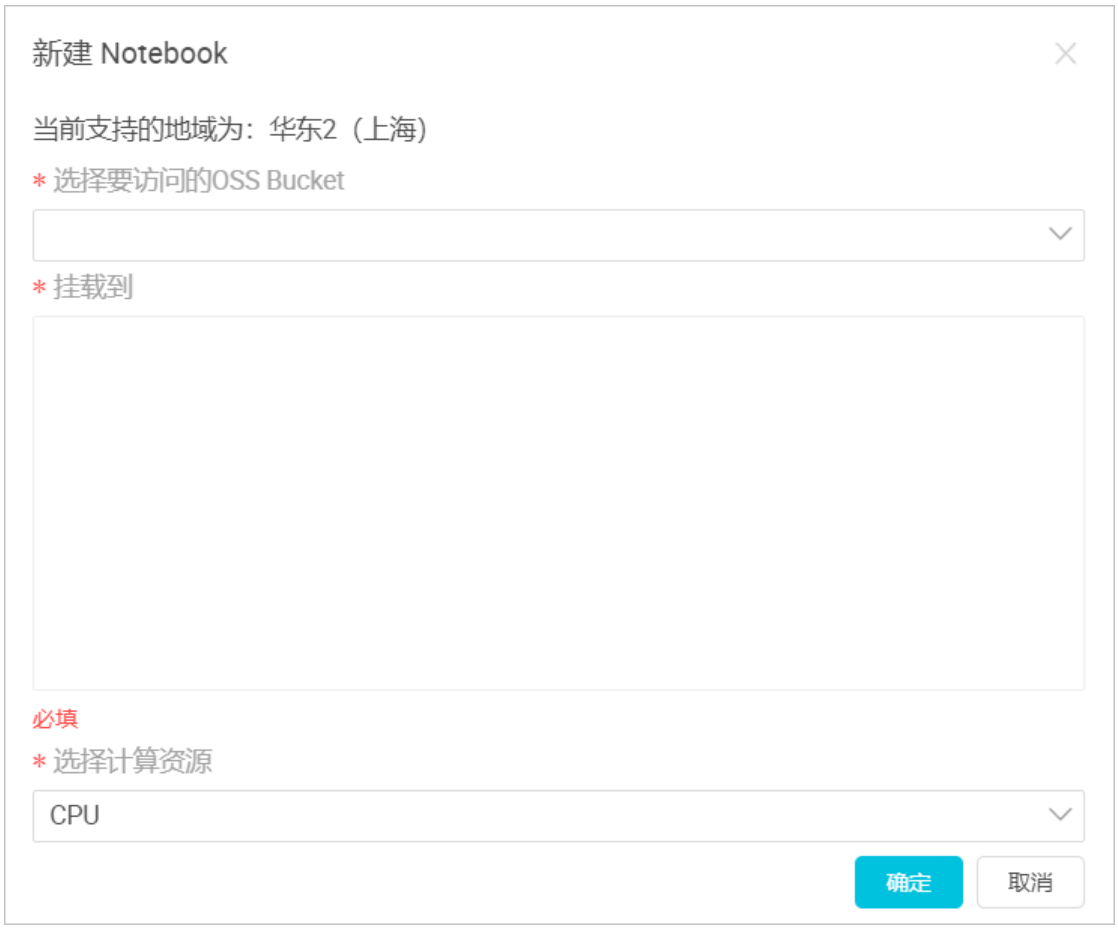
- 启动Notebook前，您选择的OSS路径下的文件（包含下级目录）的数据建议不超过80 MB。机器学习从OSS端读取数据时，超过2分钟会超时重试。
- 启动Notebook前，您选择的OSS路径下的.py，.tar.gz或.zip文件，以及包含的下级目录对应的总文件个数不可以超过500个。
- 启动Notebook后，如果超过20分钟未在前端Notebook页面操作，服务资源会被释放。如果需要再次使用，请重新启动Notebook，请注意随时保存内容。

操作步骤

1. 进入机器学习的**首页**。
   i. 登录PAI Console。
   ii. 在左侧导航栏，单击**模型开发和训练 > Studio-可视化建模**，进入**PAI可视化建模**页面。创建项目时，建议您使用按量付费模式（后付费），并开启GPU，PAI-TF任务只能在GPU资源中运行。

   iii. 单击目标项目**操作**列的**进入机器学习**。
   iv. 在左侧导航栏，单击**首页**。
2. 创建Notebook。
   i. 在**首页**页面，单击右上方的**新建实验 > 新建Notebook**。
   ii. 在**新建Notebook**对话框中，输入**名称**、**描述**，并选择**位置**。
   iii. 单击**下一步**。

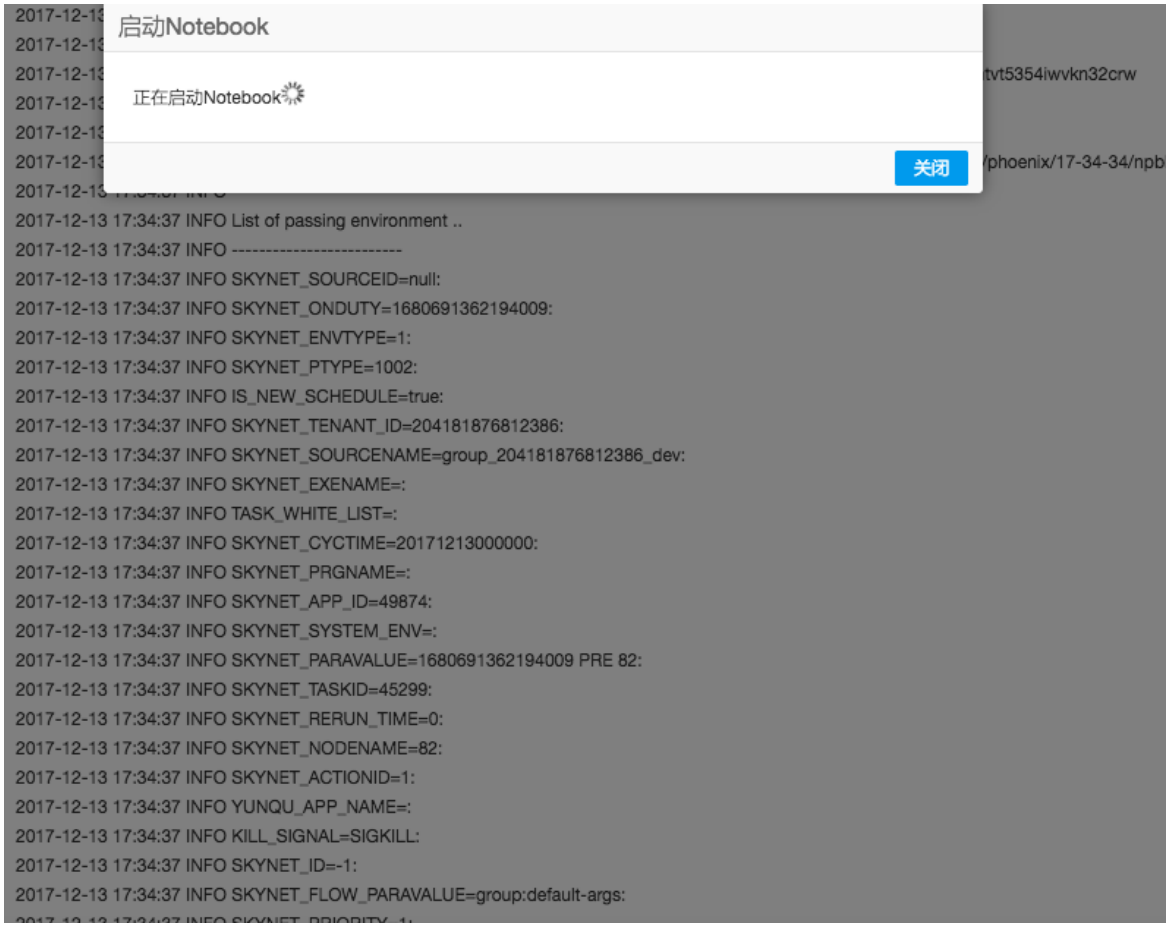
iv. 在对话框中，选择OSS Bucket下的文件夹以及对应的计算资源类型。



注意事项如下：

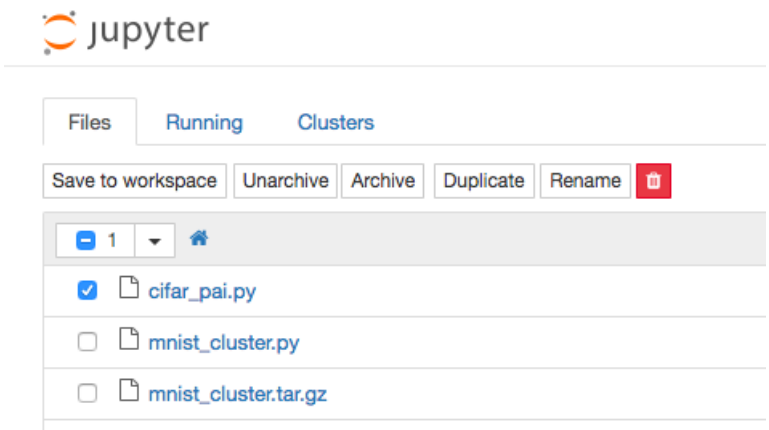
- 机器学习平台的Notebook只会拉取对应文件夹下的tar.gz、.py文件和.zip文件。
- 如果OSS Bucket与实验项目不在同一个地域下，会产生额外的跨地域费用。
- 您可以选择CPU和GPU两种计算资源，GPU可以选择所需卡数。

3. 在左侧导航栏，单击**Notebook**，查看已创建的Notebook。
4. 右键单击相应的Notebook名称，选择**打开Notebook**。您需要等待2～5分钟左右，即可启动Notebook云端服务。



5. 启动Notebook后，单击**点击查看**。

6. 编辑Notebook，其使用方式和开源版本基本一致。编辑Notebook时，请注意以下问题：
   - 对选择的文件提供了解压和压缩功能，主要针对机器学习集群项目工程需要打包为tar.gz文件的需求。
   - 只有单击了**Save to workspace**才会将文件存储到OSS对应目录下。



6.配置与使用全局变量

全局变量可以减轻您在使用PAI平台配置组件时，重复配置参数的工作量。支持创建和删除全局变量，并通过复制功能获取全局变量的引用方式。本文介绍如何配置与使用全局变量。

应用场景

- 实验内多个组件使用相同的参数。

  详情请参见示例1：实验内组件公用参数。

- 替换定时调度参数。

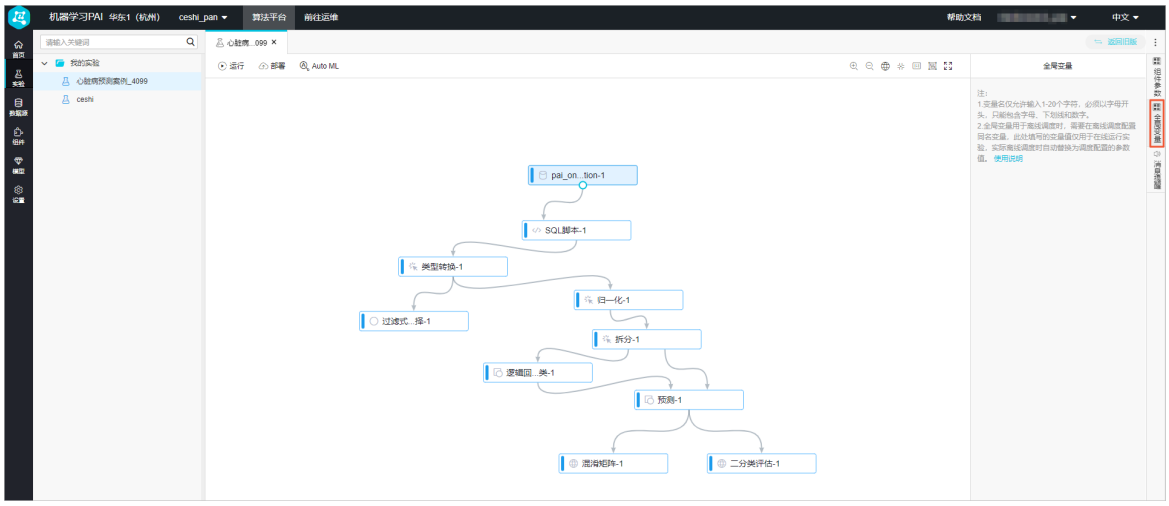
  详情请参见示例2：定时调度参数替换。

前提条件

已创建项目和实验，具体操作步骤请参见快速入门章节。

进入全局变量配置页面

1. 登录PAI控制台。
2. 在左侧导航栏，单击**模型开发和训练**下的Studio-可视化建模。
3. 在**PAI 可视化建模**页面，单击目标项目右侧的**进入机器学习**。
4. 在**实验**页面，单击目标实验。
5. 在画布中，单击选中一个组件。
6. 在右侧配置面板，单击**全局变量**，即可进入全局变量配置页面。



示例1：实验内组件公用参数

以心脏病预测模板实验为例，选取参数时直接使用全局变量替换，其他使用相同参数的地方均可使用该变量。

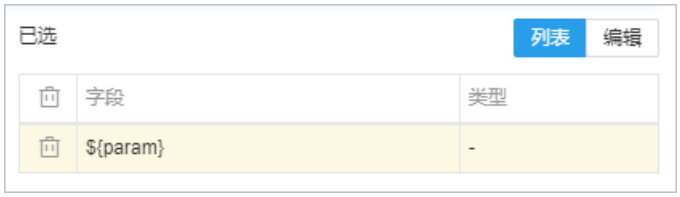
1. 配置全局变量。
   i. 进入全局变量配置页面。
   ii. 在页面下方，单击**新建全局变量**。

iii. 填写**变量名**和**变量值**。

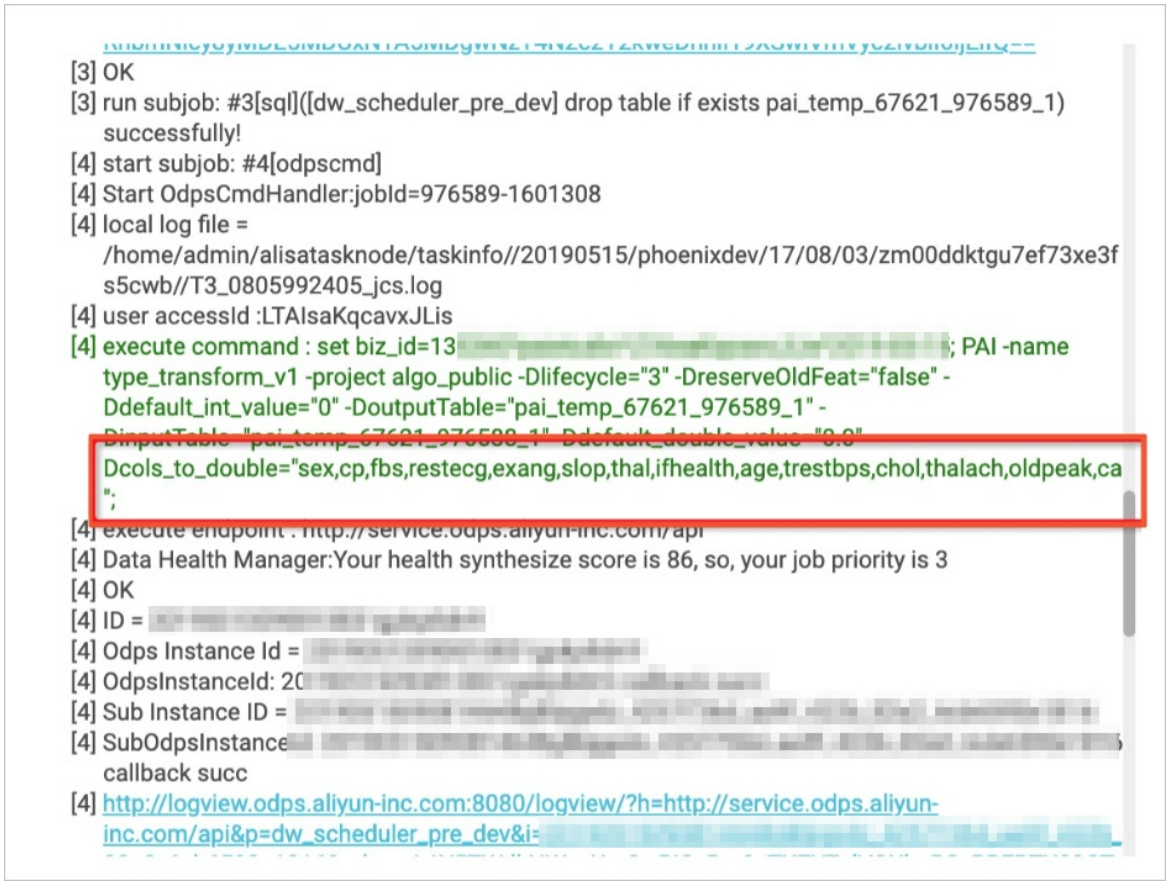
本示例填写**变量名**为param，填写完成后，系统会自动生成**引用方式**：${param}，在使用变量处直接输入${param}，即可引用该变量。**变量值**填写cp、fbs等。

2. 使用全局变量。

在使用变量处，输入该变量的**引用方式**。

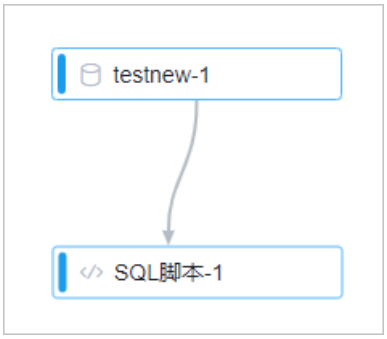


3. 运行试验，验证变量是否自动替换为填写的变量值。



示例2：定时调度参数替换

全局变量还可以用于定时调度的实验与日期关联的场景。在PAI实验里配置的全局变量值仅用于在线运行实验使用，离线调度时会使用调度中配置的参数值替换运行。本示例以以下实验为例，源数据包括两个分区的数据。



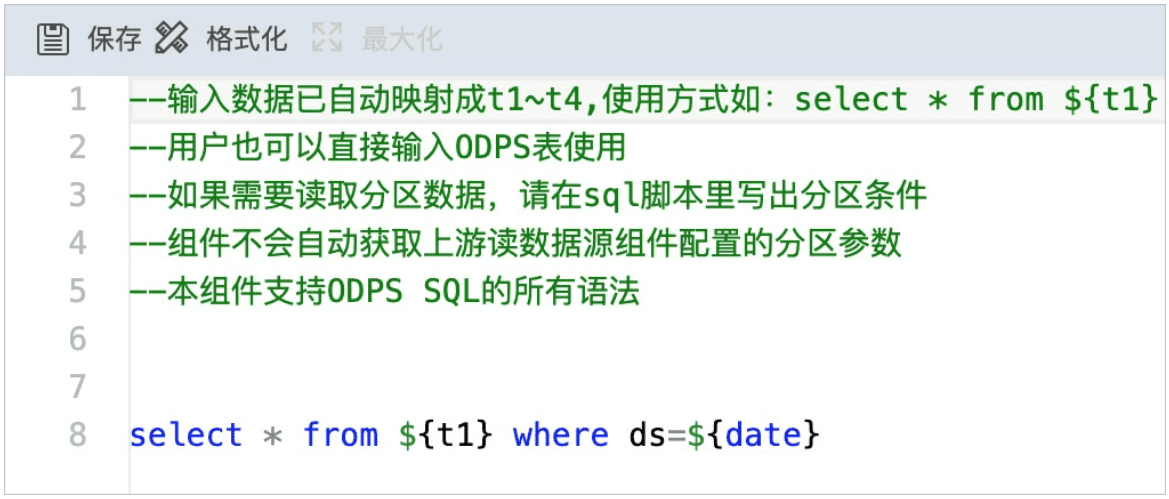
1. 准备源数据。

	A	B	C	D
1	t1	t2	label	ds
2	1.0	1.1	'A'	20190519
3	1.0	1.0	'A'	20190519
4	0.0	0.0	'B'	20190519
5	0.0	0.1	'B'	20190519
6	1.2	1.1	'A'	20190519
7	0.1	0.1	'C'	20190519
8	0.1	0.1	'C'	20190519
9	0.1	0.1	'D'	20190519
10	0.1	0.1	'E'	20190519
11	1.0	1.1	F	20190520

2. 配置全局变量。

   具体步骤请参见示例1：实验内组件公用参数。本示例使用的**变量名**为date，**变量值**为20190520。

3. 在SQL脚本组件中使用全局变量。



4. 在线运行实验，查看结果。

数据探查 – pai_temp_67897_980942_1 – (仅显示前一百条)

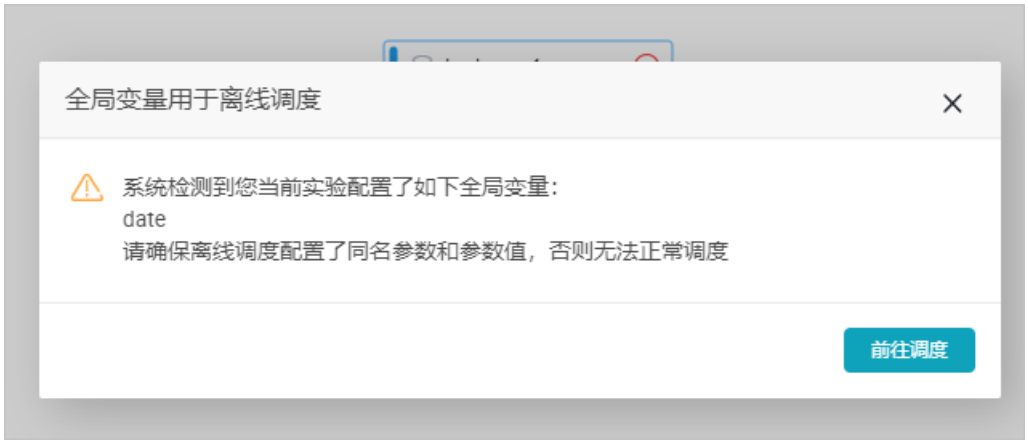
序号	t1	t2	label	ds
1	1	1.1	F	20190520

5. 配置离线调度，使用全局变量。

> 说明　以下仅介绍简单步骤，详细操作步骤请参见离线调度。

i. 进入离线调度页面。

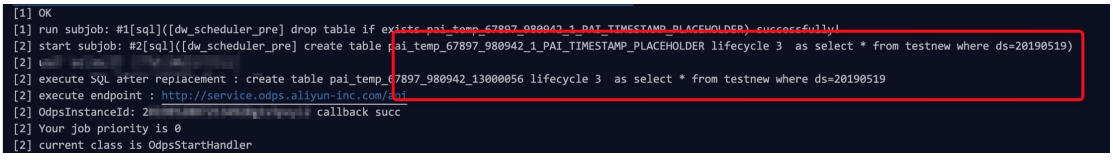
进入离线调度页面时，系统会提示配置同名参数。



ii. 配置离线调度任务，使用全局参数。

iii. 运行调度任务，查看结果。

由结果可以看到调度生成实例对变量完成替换，使用了业务日期20190519。



7.模型仓库（FastNN）

PAI模型仓库FastNN（Fast Neural Networks）是一个基于PAISoar的分布式神经网络仓库。目前FastNN已经支持了Inception、Resnet、VGG等经典算法，后续会逐步开放更多的先进模型。目前FastNN已经内置于PAI-Studio平台中，并且可以直接在该平台中使用。

准备数据源

为了方便在PAI控制台上试用FastNN，cifar10、mnist、flowers数据已下载并转换为tfrecord后存储在公开OSS上，可通过PAI的**读数据表**或**OSS数据同步**组件访问。存储OSS的路径如下。

数据集	分类数	训练集	测试集	存储路径
mnist	10	3320	350	• 北京：oss://pai-online-beijing.oss-cn-beijing-internal.aliyuncs.com/fastnn-data/mnist/ • 上海：oss://pai-online.oss-cn-shanghai-internal.aliyuncs.com/fastnn-data/mnist/
cifar10	10	50000	10000	• 北京：oss://pai-online-beijing.oss-cn-beijing-internal.aliyuncs.com/fastnn-data/cifar10/ • 上海：oss://pai-online.oss-cn-shanghai-internal.aliyuncs.com/fastnn-data/cifar10/
flowers	5	60000	10000	• 北京：oss://pai-online-beijing.oss-cn-beijing-internal.aliyuncs.com/fastnn-data/flowers/ • 上海：oss://pai-online.oss-cn-shanghai-internal.aliyuncs.com/fastnn-data/flowers/

FastNN库已支持读取tfrecord格式的数据，并基于TFRecordDataset接口实现dataset pipeline以供模型训练试用，几乎可掩盖数据预处理时间。另外，由于目前FastNN库在数据分片方面不够精细，建议您在准备数据时，尽量保证数据能平均分配到每台机器，即：

- 每个tfreocrd文件的样本数量基本一致。
- 每个worker处理的tfrecord文件数量基本一致。

如果数据格式同为tfrecord，可参考datasets目录下的cifar10、mnist和flowers等各文件实现dataset pipeline。以cifar10数据为例，实现方法如下。

假设cifar10数据的key_to_features格式为如下。

```
features={
    'image/encoded': tf.FixedLenFeature((), tf.string, default_value=''),
    'image/format': tf.FixedLenFeature((), tf.string, default_value='png'),
    'image/class/label': tf.FixedLenFeature(
     [], tf.int64, default_value=tf.zeros([], dtype=tf.int64)),
}
```

1. 在datasets目录下创建数据解析文件*cifar10.py*，并编辑内容。

```
"""Provides data for the Cifar10 dataset.
The dataset scripts used to create the dataset can be found at:
datasets/download_and_covert_data/download_and_convert_cifar10.py
"""
from __future__ import division
from __future__ import print_function
import tensorflow as tf
"""Expect func_name is  'parse_fn'
"""
def parse_fn(example):
 with tf.device("/cpu:0"):
  features = tf.parse_single_example(
   example,
   features={
    'image/encoded': tf.FixedLenFeature((), tf.string, default_value=''),
    'image/format': tf.FixedLenFeature((), tf.string, default_value='png'),
    'image/class/label': tf.FixedLenFeature(
     [], tf.int64, default_value=tf.zeros([], dtype=tf.int64)),
   }
  )
  image = tf.image.decode_jpeg(features['image/encoded'], channels=3)
  label = features['image/class/label']
  return image, label
```

2. 在*datasets/dataset_factory.py*中补足dataset_map。

```
from datasets import cifar10
datasets_map = {
  'cifar10': cifar10,
}
```

3. 执行任务脚本时，指定参数dataset_name=cifar10和train_files=cifar10_train.tfrecord，即可使用cifar10数据进行模型训练。

> **说明** 如果您需要读取其他的格式数据，需自行实现dataset pipeline构建逻辑（参考*utils/dataset_utils.py*）。

超参文件说明

PAI-FastNN支持以下类型的超参：

- 数据集参数：确定训练集的基本属性的参数，例如训练集存储路径dataset_dir。
- 数据预处理参数：数据预处理函数及dataset pipeline相关参数。
- 模型参数：模型训练基本参数，包括model_name、batch_size等。

- 学习率参数：学习率及其相关调优参数。
- 优化器参数：优化器及其相关参数。
- 日志参数：关于输出日志的参数。
- 性能调优参数：混合精度等其他调优参数。

超参文件的格式如下。

```
enable_paisora=True
batch_size=128
use_fp16=True
dataset_name=flowers
dataset_dir=oss://pai-online-beijing.oss-cn-beijing-internal.aliyuncs.com/astnn-data/flowers/
model_name=inception_resnet_v2
optimizer=sgd
num_classes=5
job_name=worker
```

- 数据集参数

名称	类型	描述
dataset_name	string	指定输入数据解析文件的名称。取值包括：mock、cifar10、mnist、flowers，取值说明请参见*images/datasets*目录下所有的数据解析文件。默认使用模拟数据mock。
dataset_dir	string	指定输入数据集的绝对路径，默认为None。
num_sample_per_epoch	integer	指定数据集总样本数，一般用来配合学习率的衰减。
num_classes	integer	指定样本分类数，默认为100。
train_files	string	指定所有训练数据的文件名，文件间分隔符为逗号，例如0.tfrecord,1.tfrecord。

- 数据预处理参数

名称	类型	描述
preprocessing_name	string	和model_name共同指定数据预处理的方法名，取值范围请参见*images/preprocessing*目录下的*preprocessing_factory*文件。默认设置为None，表示不进行数据预处理。
shuffle_buffer_size	integer	在生成数据流水线时，以样本为粒度进行shuffle的缓存池大小，默认为1024。
num_parallel_batches	integer	与batch_size乘积为map_and_batch的并行线程数，协助指定解析样本的并行粒度，默认为8。
prefetch_buffer_size	integer	指定数据流水线预取数据的批数，默认为32。

名称	类型	描述
num_preprocessing_threads	integer	指定数据流水线进行并行数据预取的线程数，默认为16。
datasets_use_caching	bool	是否打开以内存为开销，进行输入数据的压缩缓存。默认为False，表示不打开。

- 模型参数

名称	类型	描述
task_type	string	任务类型，取值包括： ○ pretrain：模型预训练，默认。 ○ finetune：模型调优
model_name	string	指定进行训练的模型，取值包括*images/models*下的所有模型。您可以参考*images/models/model_factory*文件中所有定义的模型设置model_name，默认为inception_resnet_v2。
num_epochs	integer	训练集训练轮数，默认为100。
weight_decay	float	模型训练时权重的衰减系数，默认为0.00004。
max_gradient_norm	float	是否根据全局归一化值进行梯度裁剪。默认为None，表示不进行梯度裁剪。
batch_size	integer	单卡一次迭代处理的数据量，默认为32。
model_dir	string	重载checkpoint的路径。默认为None，表示不进行模型调优。
ckpt_file_name	string	重载checkpoint的文件名，默认为None。

- 学习率参数

名称	类型	描述
warmup_steps	integer	逆衰减学习率的迭代数，默认为0。
warmup_scheme	string	学习率逆衰减的方式。取值为t2t（Tensor2Tensor），表示初始化为指定学习率的1/100，然后exponentiate逆衰减到指定学习率为止。

名称	类型	描述
decay_scheme	string	学习率衰减的方式。可选值： ○ luong234：在2/3的总迭代数之后，开始4次衰减，衰减系数为1/2。 ○ luong5：在1/2的总迭代数之后，开始5次衰减，衰减系数为1/2。 ○ luong10：在1/2的总迭代数之后，开始10次衰减，衰减系数为1/2。
learning_rate_decay_factor	float	指定学习率衰减系数，默认为0.94。
learning_rate_decay_type	string	指定学习率衰减类型，可选值：fixed、exponential（默认）和polynomial。
learning_rate	float	指定学习率初始值，默认为0.01。
end_learning_rate	float	指定衰减时学习率值的下限，默认为0.0001。

- 优化器参数

名称	类型	描述
optimizer	string	指定优化器名称。可选值：adadelta、adagrad、adam、ftrl、momentum、sgd、rmsprop、adamweightdecay， 默认为rmsprop。
adadelta_rho	float	adadelta的衰减系数，默认为0.95。
adagrad_initial_accumulator_value	float	AdaGrad积累器的起始值，默认为0.1。是Adagrada优化器专用参数。
adam_beta1	float	一次动量预测的指数衰减率，默认为0.9。是Adam优化器专用参数。
adam_beta2	float	二次动量预测的指数衰减率，默认为0.999。是Adam优化器专用参数。
opt_epsilon	float	优化器偏置值，默认为1.0。是Adam优化器专用参数。
ftrl_learning_rate_power	float	学习率参数的幂参数，默认为-0.5。是Ftrl优化器专用参数。
ftrl_initial_accumulator_value	float	FTRL积累器的起始，默认为0.1，是Ftrl优化器专用参数。
ftrl_l1	float	FTRL l1正则项，默认为0.0，是Ftrl优化器专用参数。
ftrl_l2	float	FTRL l2正则项，默认为0.0，是Ftrl优化器专用参数。
momentum	float	MomentumOptimizer的动量参数，默认为0.9，是Momentum优化器专用参数。

名称	类型	描述
rmsprop_momentum	float	RMSPropOptimizer的动量参数，默认为0.9。
rmsprop_decay	float	RMSProp的衰减系数，默认为0.9。

- 日志参数

名称	类型	描述
stop_at_step	integer	训练总迭代数，默认为100。
log_loss_every_n_iters	integer	打印loss信息的迭代频率，默认为10。
profile_every_n_iters	integer	打印timeline的迭代频率，默认为0。
profile_at_task	integer	输出timeline的机器对应索引，默认为0，对应chief worker。
log_device_placement	bool	是否输出device placement信息，默认为False。
print_model_statistics	bool	是否输出可训练变量信息，默认为false。
hooks	string	训练hooks，默认为StopAtStepHook,ProfilerHook,LoggingTensorHook,CheckpointSaverHook。

- 性能调优参数

名称	类型	描述
use_fp16	bool	是否进行半精度训练，默认为True。
loss_scale	float	训练中loss值scale的系数，默认为1.0。
enable_paisoar	bool	是否使用paisoar框架，默认True。
protocol	string	默认grpc.rdma集群可以使用grpc+verbs，提升数据存取效率。

开发主文件

如果已有模型无法满足您的需求，您可以通过继承dataset、models和preprocessing接口进一步开发。在此之前需要了解FastNN库的基本流程（以images为例，代码入口文件为*train_image_classifiers.py*），整体代码架构流程如下。

```
# 根据model_name初始化models中对应模型得到network_fn，并可能返回输入参数train_image_size。
  network_fn = nets_factory.get_network_fn(
      FLAGS.model_name,
      num_classes=FLAGS.num_classes,
      weight_decay=FLAGS.weight_decay,
      is_training=(FLAGS.task_type in ['pretrain', 'finetune']))
# 根据model_name或preprocessing_name初始化相应数据预处理函数得到preprocess_fn。
  preprocessing_fn = preprocessing_factory.get_preprocessing(
        FLAGS.model_name or FLAGS.preprocessing_name,
        is_training=(FLAGS.task_type in ['pretrain', 'finetune']))
# 根据dataset_name，选择正确的tfrecord格式，同步调用preprocess_fn解析数据集得到数据dataset_iterator。
  dataset_iterator = dataset_factory.get_dataset_iterator(FLAGS.dataset_name,
                                  train_image_size,
                                  preprocessing_fn,
                                  data_sources,
# 调用network_fn、dataset_iterator，定义计算loss的函数loss_fn。
  def loss_fn():
   with tf.device('/cpu:0'):
     images, labels = dataset_iterator.get_next()
    logits, end_points = network_fn(images)
    loss = tf.losses.sparse_softmax_cross_entropy(labels=labels, logits=tf.cast(logits, tf.float32), weights=1.
0)
    if 'AuxLogits' in end_points:
     loss += tf.losses.sparse_softmax_cross_entropy(labels=labels, logits=tf.cast(end_points['AuxLogits'], tf
.float32), weights=0.4)
    return loss
# 调用PAI-Soar API封装loss_fn、tf原生optimizer。
  opt = paisoar.ReplicatedVarsOptimizer(optimizer, clip_norm=FLAGS.max_gradient_norm)
  loss = optimizer.compute_loss(loss_fn, loss_scale=FLAGS.loss_scale)
# 根据opt和loss形式化定义training tensor。
  train_op = opt.minimize(loss, global_step=global_step)
```