YonghongTech
Talk with Data

深度分析手册

Yonghong Z-Suite — V7.5

版权声明

免责声明

通讯方式

北京永洪商智科技有限公司

北京市朝阳区光华路 9 号光华路 SOHO 二期 C 座 9 层

电话：(86-10)-58430919

邮箱：public@yonghongtech.com

网站：http://www.yonghongtech.com

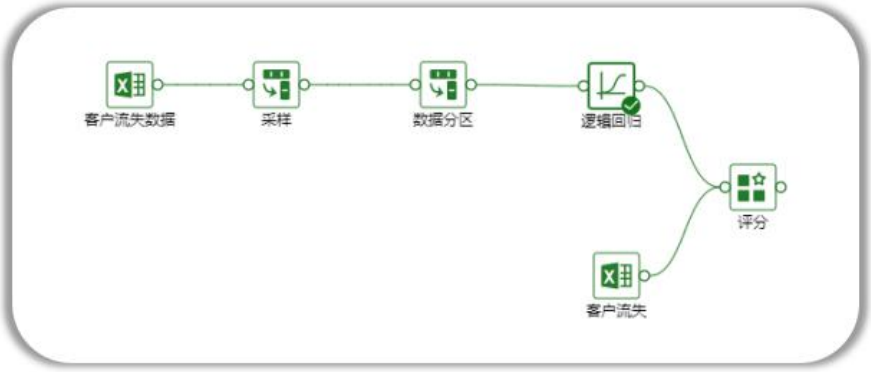
目录

随着计算机技术与网络技术的飞速发展，各企业都积累了大量的业务数据。这些数据，不仅如实反映了企业的经营状况，还潜藏着巨大的商业价值。面对海量数据，企业亟需一款功能强大的数据处理工具，准确高效地探索数据，反映数据事实，揭示数据规律，为企业的经营决策提供可靠信息。然而，传统的数据分析只局限于反映企业过去和现在的经营状况。管理者无法从报表展示中直接得出问题结论。企业的数据也无法发挥真正的价值。

永洪深度分析集成了复杂的统计算法和机器学习技术，能够从海量数据中，挖掘具有潜在价值的关系、模式和趋势，构建数据模型，做出预测分析。帮助企业及时了解自身问题，发现市场机会，做出科学的经营决策。

作为永洪一站式数据分析平台的重要组成部分，永洪深度分析从 7.5 版本开始，进行了功能模块化升级，实现了流程化操作分析。深度分析作为一个独立的功能模块，拥有全新的操作界面，实现更专业的机器学习功能。通过深度分析的可视化工作流，用户可将数据集拆分为测试集和训练集，选择特征列和目标列，选择分析算法、构建算法模型，对模型进行训练、得出模型参数，利用测试集对模型评分，调整参数/特征、使模型更具准确性，最终应用于数据集和可视化报告，得到预测分析结果。当然，永洪深度分析仍然包含传统的快速分析。快速分析是在数据集或者图表组件上快速的创建分析算法，快速的绑定到组件上以实现可视化结果。

永洪流程式深度分析既提供经典统计方法，如逻辑回归、K-Means 聚类、时序分析、关联规则、决策树，满足用户常用分析场景，简单可视化操作，即可轻松构建模型，完成预测分析；又支持 R 模型用户可调取 R 包函数，定制分析算法，充分发挥 R 中更丰富、更高级的统计分析与预测分析功能。



1. 进入深度分析

深度分析是 Yonghong Z-Suite 的重要功能块之一。既提供经典统计方法，如逻辑回归、K-Means 聚类、时序分析、关联规则、决策树，满足用户常用分析场景，简单可视化操作，即可轻松构建模型，完成预测分析；又支持 R 模型用户可调取 R 包函数，定制分析算法，充分发挥 R 中更丰富、更高级的统计分析与预测分析功能。

启动深度分析的步骤如下：

1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器， 然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是您的机器名，如果是本机访问，可以用 localhost。8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

4）点击深度分析按钮后，进入到深度分析的界面。

5）点击深度分析按钮后，进入到深度分析的界面。

6）运行建模节点，请先在管理系统模块中系统设置页面的 R 计算配置把 R 服务器配置好。

R 计算配置界面如下图所示：

配置信息之前，用户需要在本地上安装永洪产品，然后可以使用另一台电脑同时安装 R 和 Rserve，启动 RServe（Rserve 的安装和启动请参考“安装启动 Rserve”章节）。，然后在本地永洪产品 R 计算模块配置上远程电脑（即安装 R 和 Rserve 的电脑）的 IP，端口，以及系统，如果远程的电脑有用户名和密码，必须输入用户名和密码，就可以在本地使用远程的 R 做数据分析。

【系统】RServe 和 R 支持 Liunx 系统和 Windows。其中 Linux 并行计算；Windows 只支持单线程计算。

【服务器地址】R 安装的服务器 IP。

【端口】端口默认为 6311 。

【登录验证】当勾选时，用户名与密码输入框为可输入状态。

【用户名】输入用户名，用于验证 Rserve 连接权限。

【密码】输入用户名对应的密码。

【测试连接】测试 Rserve 是否连接成功。

【启动连接】保存当前设置，以连接 R 和 RServe。

2. 安装启动 RSERVE

❖ **推荐的安装方式**

（1）windows 安装产品，推荐用 linux 虚拟机的方式

安装 linux 虚拟机，里面有已经配置好的 R 和 Rserve，把 Rserve 启动即可。具体请参考“linux 虚拟机下安装启用 Rserve”的内容。

（2）linux 安装产品，推荐用 docker 镜像的方式

在 linux 上安装 docker，下载 R 的 docker 镜像（里面是 centos7，已经包含 R 和 Rserve），启动 docker，把 Rserve 启动即可。具体请参考“Docker 方式安装启用 Rserve”的内容。

以上两种方式为主要推荐的方式，其他情况包括：

（1）客户 linux 不支持 docker（docker 需要 linux 内核 3.10 以上）

需要安装 R（相关的依赖包也要装），再安装第三方的 packages。具体请参考“直接安装 R”的内容。

（2）windows 上，直接配置 R

只需要复制相关资源，配置好环境变量即可。具体请参考“Windows 下安装启用 Rserve”的内容。

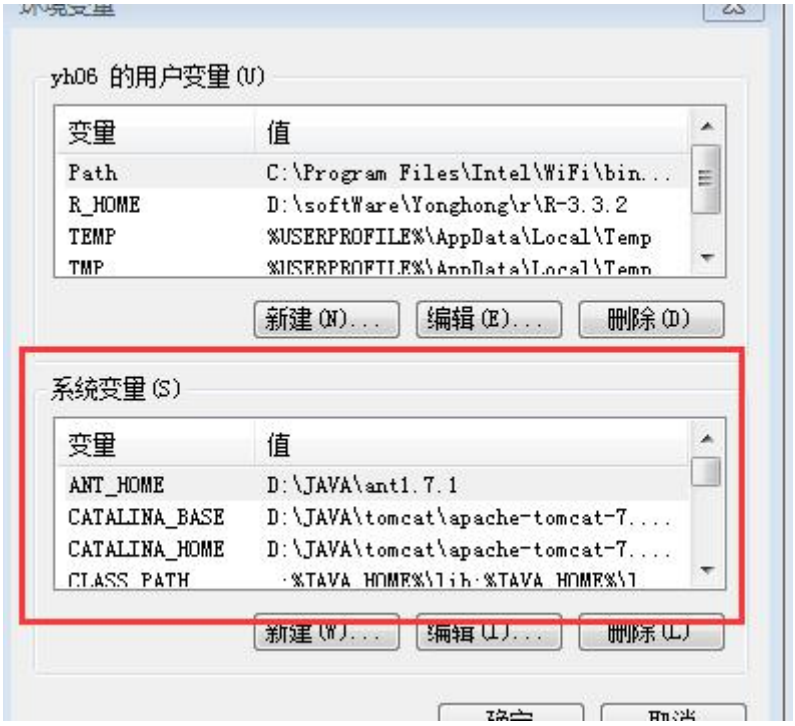
❖ Windows 下安装启用 Rserve

使用步骤：

1、将文件夹 r 拷贝到电脑上。

2、配置系统的环境变量

我的电脑->右键属性->高级系统设置->高级->环境变量->系统变量



新建,变量名为 R_HOME，值为 R-3.3.2 的文件夹路径

编辑变量 Path,在变量值后面添加;%R_HOME%\bin\x64;

3、后台运行 Rserve

控制台进入 rserve 文件夹下，运行 start /b Rserve.exe。

```
D:\softWare\Yonghong\r\rserve>start /b Rserve.exe

D:\softWare\Yonghong\r\rserve>WARNING: useplain=no, but this Rserve has no crypt
 support!
Set useplain=yes or compile with crypt support (if your system supports crypt).
Falling back to plain text password.
Rserve: Ok, ready to answer queries.
```

❖ linux 虚拟机下安装启用 Rserve

使用步骤：

1. 解压 centos7.zip。
2. 安装 VMware。
3. 进入操作系统的 BIOS，设置里面的 security 里的虚拟机那个选项设置为 enabled。
4. 然后启动 VMware，打开解压的那个虚拟机。
5. 关闭防火墙：systemctl stop firewalld.service
6. 远程模式启动 Rserve：R CMD Rserve --RS-enable-remote

❖ Docker 方式安装启用 Rserve

系统要求：centos7

linux 内核要求：3.10 以上

安装步骤：

1. 安装 docker：

   yum install -y docker

2. 关闭防火墙：

   systemctl stop firewalld.service

3. 启动 docker 并设置为开机自动启动：

systemctl start docker.service

systemctl enable docker.service

如果执行上面两条命令系统不支持，用下面的两条命令，如果成功，则直接进行第三步。

service docker start

chkconfig docker on

4. 将 centos_r.tar 放入 linux 机器中。

5. 导入镜像：

在同级目录执行 docker load --input centos_r

6. 运行镜像：

docker run -ti -p 192.168.20.191:5000:6311 yonghong/centos_r:1.0 /bin/bash

本条指令是将 docker 内 R 运行的 6311 端口映射到 linux 机器上的 5000 端口，并且进入 docker 容器内部，其中红色字体为本机的 ip 地址和端口号，即 ip 地址:端口号。

7. 使环境变量生效:

Source /etc/profile

8. 启动 R：

R CMD Rserve --RS-enable-remote

9. 按 Ctrl + P + Q 退出容器，再用 docker ps 进行查看运行状态，如果此刻还是运行的则正确。

❖ **直接安装 R**

系统要求：centos7,

Java 环境已经配置完成(包含 JAVA_HOME,PATH,CLASSPATH)

安装步骤：

1. 下载 R-3.3.2.tar.gz

2. 解压为 R-3.3.2，tar -xzvf R-3.3.2.tar.gz

3. 添加依赖包：

```
yum install -y gcc
yum install -y gcc-c++
yum install gcc-gfortran
yum install -y glibc-headers
yum install -y readline-devel
yum install -y wget libXt-devel
yum install -y fonts-chinese tcl tcl-devel tclx tk tk-devel
yum install -y mesa-libGLU mesa-libGLU-devel
yum install -y install bzip2-devel
yum install -y install xz-devel.x86_64
yum install -y install pcre-devel
```

yum install -y install libcurl

yum install -y install libcurl-devel

yum install -y texinfo.x86_64

yum install -y texlive-pdftex-doc.noarch

yum install -y texlive

yum install -y openssl-devel

yum install -y libxml2

yum install cairo-devel.x86_64

4. 进入 R-3.3.2 目录，然后安装

./configure --enable-R-shlib=yes --with-tcltk --prefix=/home/R

其中 --prefix 为准备安装 R 的地址，此时为 R 文件夹。

5. make 和安装

make

make install

6. 配置环境变量

编辑变量：vim /etc/profile

添加变量：export R_HOME=/usr/local/R/lib64/R

export PATH=$PATH:$R_HOME/bin

使环境变量生效：source /etc/profile

7. xshell 中输入 R 进入 R 的控制台。

8. 安装 R 语言包：

```
install.packages("flexclust")
install.packages("Rserve")
install.packages("rpart.plot")
install.packages("Cairo")
install.packages("arules")
install.packages("arulesViz", dependencies=TRUE, INSTALL_opts = c('--no-lock
'))
install.packages("glmnet")
install.packages("forecast")
```

9. 关闭防火墙：systemctl stop firewalld.service

10. 启动 RServe(允许远程)：

```
R CMD Rserve --RS-enable-remote
```

- **部分报错信息解决方案：**

1. make: *** [read.o] Error 1

ERROR: compilation failed for package ‘png’

解决：yum install libpng-devel

2. 如果 igraph 安装不上

解决：下载一个 igraph_1.0.1.tar.gz，离线安装,指令为 R INSTALL igraph_1.0.1.tar.gz

3. rJava 如果装不上

解决：在 shell 终端运行 R CMD javareconf,前提是系统的 java 环境配置正确，环境变量中的 JAVA_HOME,PATH,CLASSPATH。

4. zlib>1.2.5 no

1) 下载，解压 zlib1.2.8 到/usr/local/src/下

2) 查看一下 zlib 安装文件，rpm -ql zlib，可以看到库文件在 lib64 下

3) 配置 zlib，参数为 ./configure --prefix=/usr/local/zlib/。

4) 用 make 进行编译

5) 卸载 zlib，掌握这个顺序很有必要，如果卸载早了的话，上一步就会提示失败。卸载指令 rpm -e --nodeps zlib，卸载完成之后，发现/lib64/目录下，zlib 的库文件，libz.so*没有了。

6) 用 make install 安装 zlib，安装完成之后，可以看到/usr/local/zlib/目录下有个 lib 目录，里面存放的就是 zlib 的库文件。

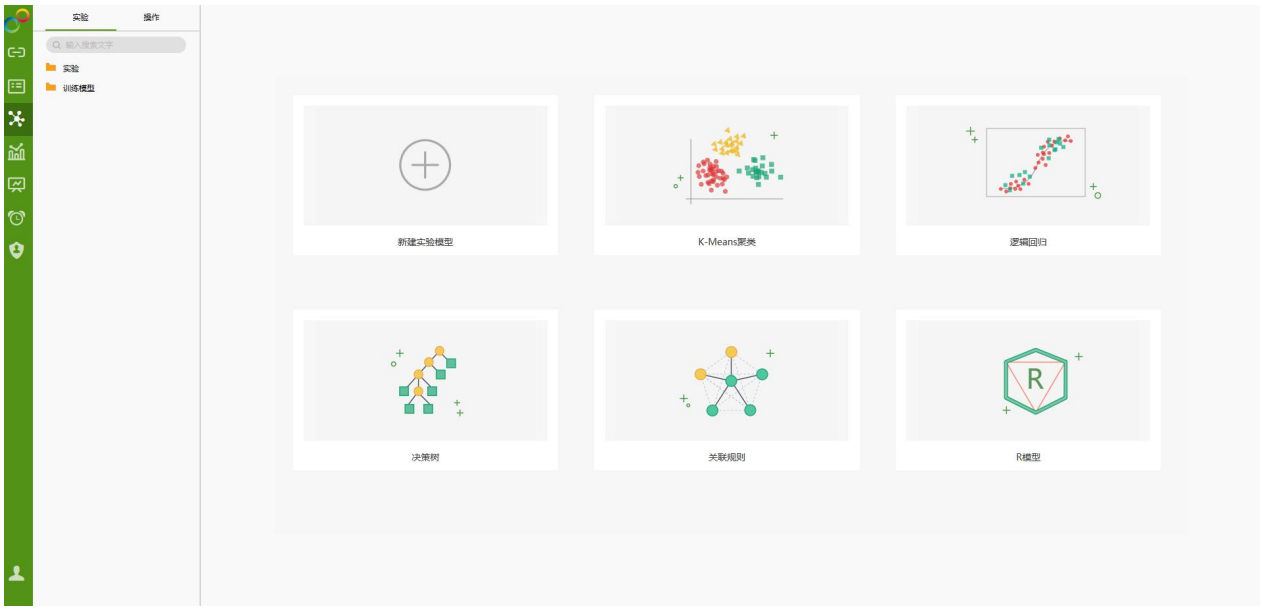
7) 这时候用 yum 等工具，会提示确实 libz.so*支持，所以必须把当前共享库文件注册到系统中，打开 /etc/ld.so.conf，在下面加入一行/usr/local/zlib/lib/,然后保存退出。

8) 用 ldconfig 重新更新一下 cache,这时候再用 yum 等工具，发现运行正常了，至此，zlib 就更新成功。

3. 深度分析界面

深度分析界面包含首页和编辑页面。

首页提供了深度分析的 5 个经典案例。用户可以从首页新建实验进入实验编辑页面。



在编辑页面中，用户可以探索数据特征，构建分析模型，查看预测结果。

具体界面内容请参看对应章节。

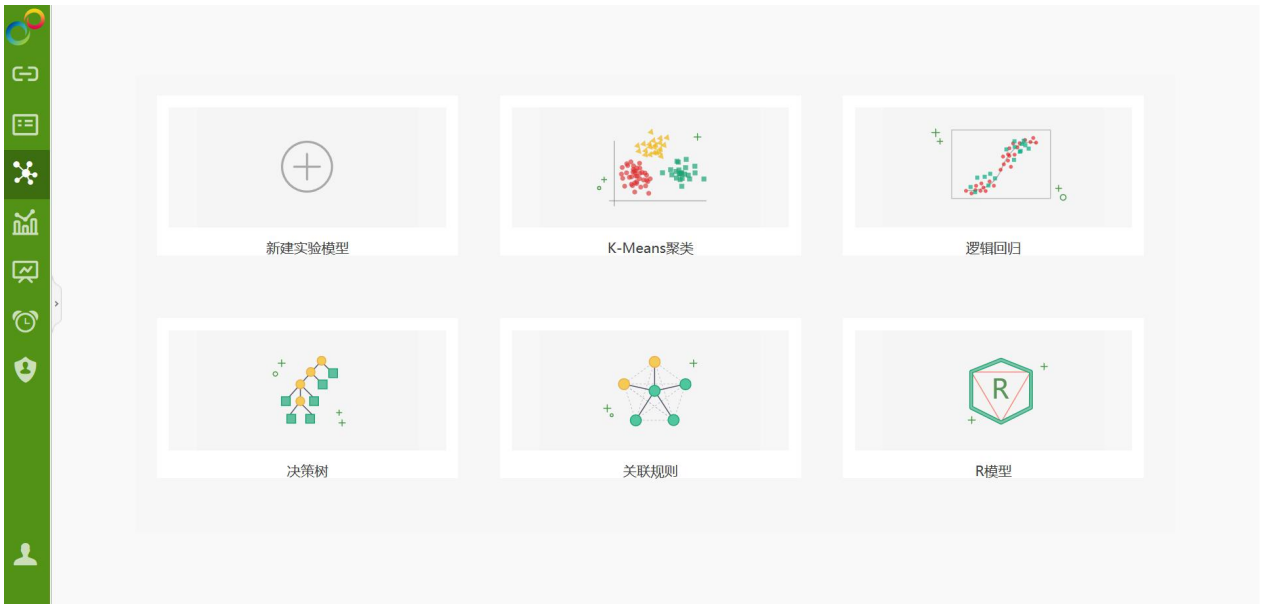
3.1.深度分析首页

深度分析首页包括新建实验模型按钮，5 个深度分析案例，点击可新建实验模型或者打开案例。

展示如下图：

鼠标移到案例处，会显示案例的背景介绍，如下图：

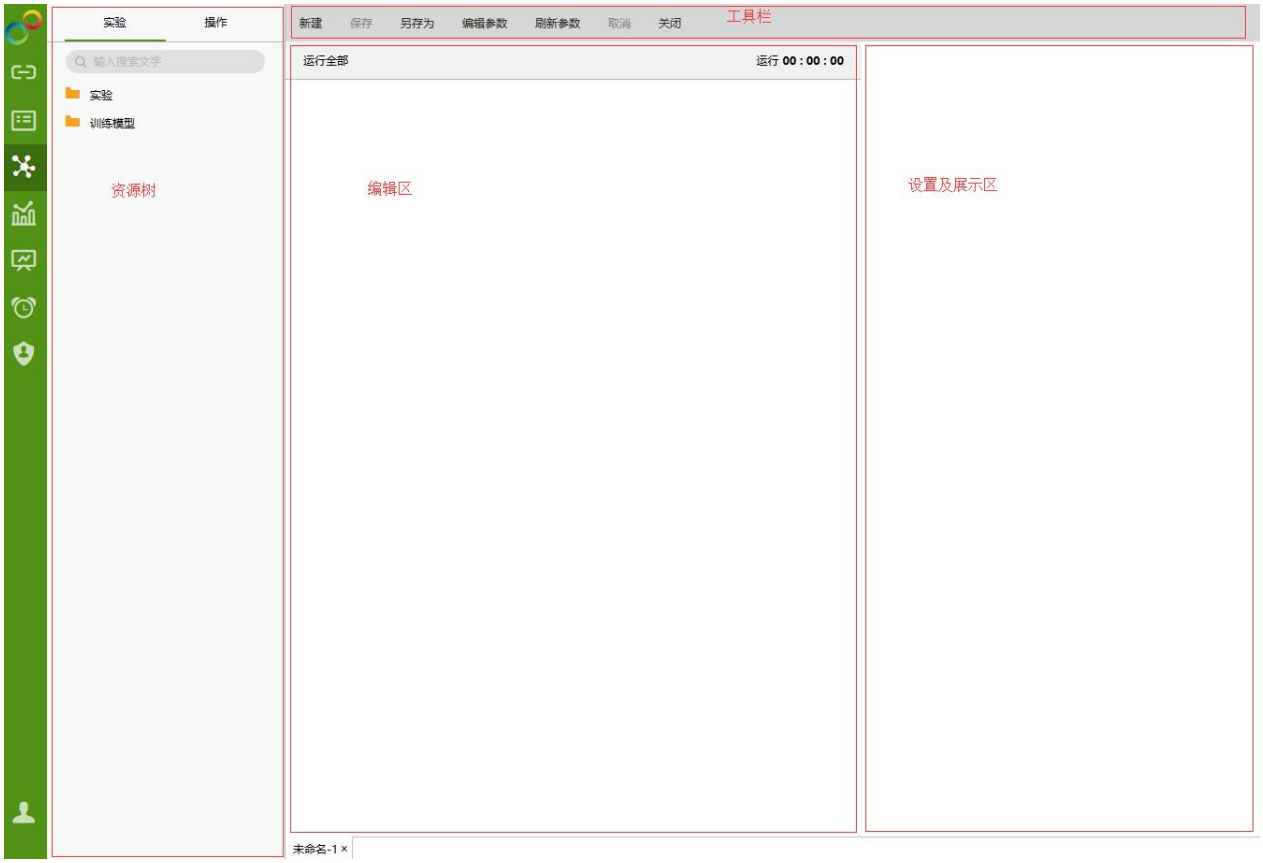
鼠标移到资源树和引导页交界线处，会出现一个弹入弹出按钮，默认资源树是显示的，点击一下此按钮资源树被隐藏，如下图：



再点击一下资源树展示。还可以通过拖拽交界线改变资源树的显示尺寸。

3.2.深度分析编辑页面

点击引导页的新建实验模型按钮，新建一个空实验，包括工具栏、资源树、编辑区、设置及展示区。展示如下图：



设置及展示区与编辑区密切相关，统一在编辑区页面进行介绍。其它区域请参看各自章节。

3.2.1. 工具栏

❖ 新建

点击创建新的实验。

❖ 保存

当用户保存新建的实验时，会弹出保存对话框，用户可设定保存路径以及实验的名称。当用户打开已经存在的实验后，对该实验进行修改，可点击保存按钮直接保存。

❖ 另存为

另保存已打开的实验。

❖ 编辑参数

可对当前实验添加参数，删除参数以及收集数据集节点中使用到的参数，编辑参数对话框如下图所示。

- **添加 / 删除**

用户点击添加按钮时，弹出名称编辑对话框，用户可设定参数的名称。在设定好参数后，不支持再对此参数进行重命名操作。倘若是收集到的参数，则在此对话框中不能删除，但可以对此参数进行编辑。当被收集的参数从所定义的地方被删除时，则在此参数对话框中该参数处于可被删除状态。

- **信息**

显示参数被引用的位置。

- **类型**

用户设定当前参数的数据类型。

- 默认

用户可设定参数的默认值。用户可设定单个值、或者空值，不支持定义多个值。

- 弹出

当勾选此项时，用户在点击刷新参数按钮或者在编辑器中再次打开此实验时，会弹出参数值输入对话框，如下图所示。在此对话框中存在设定的默认值。

- 参与报表 “ 参数过滤 ” 的过滤策略

只有自定义的参数才有这个选项。勾选 参与报表 “ 参数过滤 ” 的过滤策略 后，自定义的参数会受仪表盘属性里的参数未选值策略的控制。自定义的参数默认是非必选的。

- 可选值

深度分析里不支持设置可选值。

❖ 刷新参数

对设置了弹出状态的参数重新输入参数值。如存在一参数 param, 处于弹出状态，并且存在默认值 1，如下图所示。

当用户点击刷新参数按钮时，也将会把此参数的默认值刷出来，如下图所示。

❖ 取消

当某一操作长时间未响应时可点击此按钮取消当前操作。

❖ 关闭

关闭当前实验，倘若用户尚未保存对当前实验的修改，将弹出提示对话框。

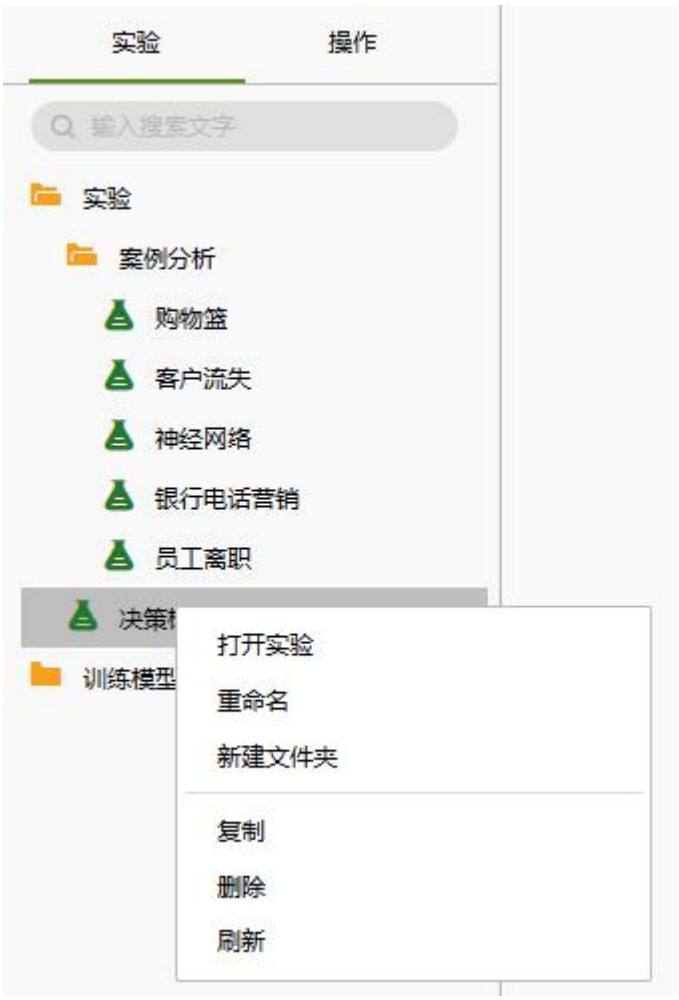
3.2.2. 资源树

资源树包含实验和操作两部分。

实验包括已保存的实验和训练模型，用户可对其进行管理。实验是数据挖掘的流程，训练模型是实验训练后的结果。

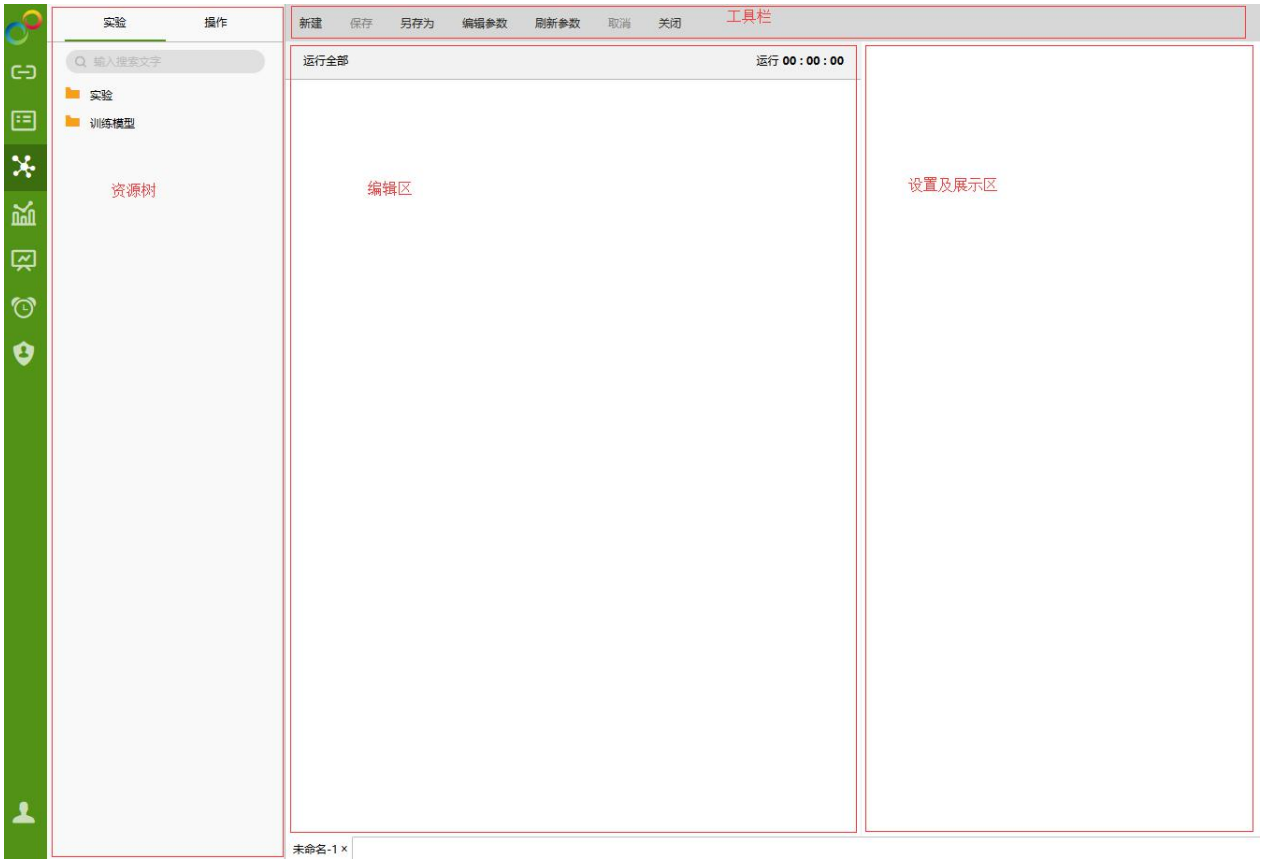
操作是制作分析流程所用的节点，主要应用在编辑区，具体介绍请看编辑区介绍。

在实验树上可以通过鼠标右键来创建文件夹，重命名文件夹及文件，移除文件夹及文件，以及刷新文件目录等操作。不能对已经打开的资源进行重命名、移除操作。还可搜索已经存在的资源。不同文件夹目录下支持重名，但同一个文件夹目录下不允许重名。文件夹内可以嵌套文件夹。



3.2.3. 编辑区

编辑区是用来制作流程式分析的容器，可以拖拽不同的节点进来，建立不同的分析流程，将其结果以可视化的形式展示出来。具体介绍请参看以下章节。



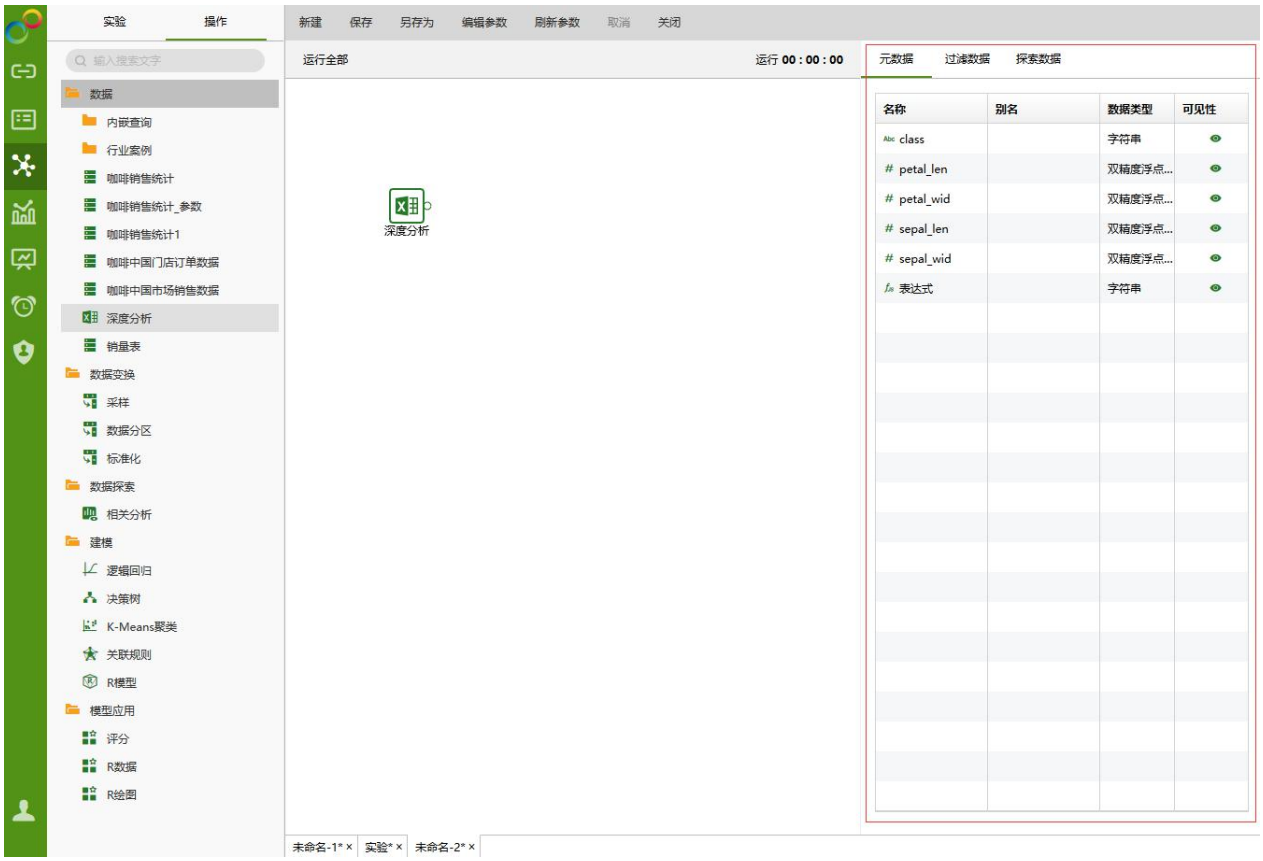
3.2.3.1. 节点类型

节点包含操作树上的数据、数据变换、数据探索、建模、模型应用。以下章节将分别介绍每种节点的参数信息。

3.2.3.1.1. 数据

❖ 数据

数据包含创建数据集模块里创建的所有数据集。数据集节点是深度分析流程的输入节点。拖拽一个数据集节点到编辑区。设置及展示区包含三个页面：元数据、过滤数据、探索数据。



- 元数据

【名称】数据集字段名称。

【别名】可以给新建的表达式字段设置别名，数据集里的原始字段不允许设置别名。

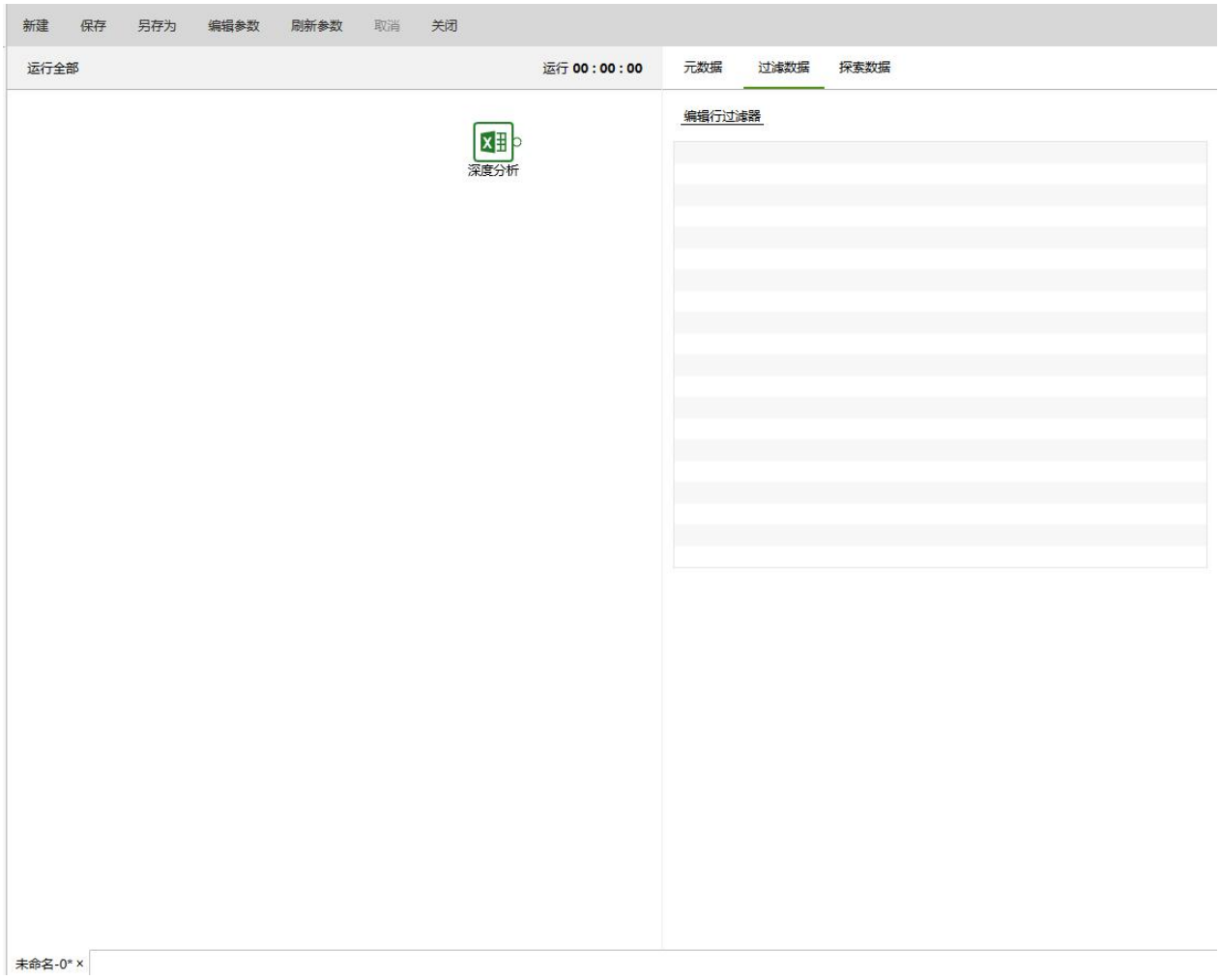
【数据类型】数据集节点中各字段的数据类型，不可以修改。

【可见性】设置字段在探索数据页面中的显示和隐藏。

【表达式】元数据区域可以通过右键菜单创建 JS 表达式，数据范围，缺失值填充，拆分列，去空格，值映射，转换为日期列，转换为数字列。此处不做过多介绍，详细用法请参照创建数据集模块中的数据类型和字段类型章节。

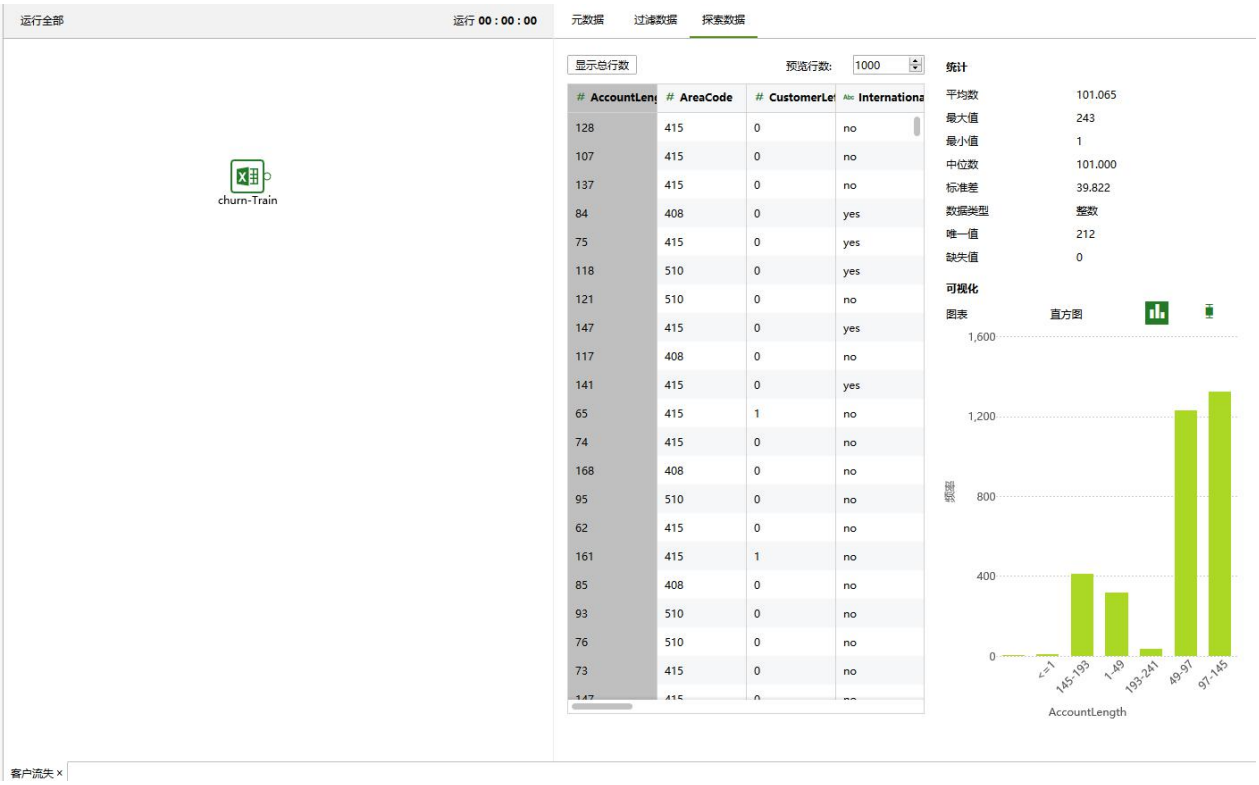
- 过滤数据

可以增加行过滤器来筛选数据集的行数据，详细用法请参照创建数据集模块中的虚拟权限控制中行控制章节。



- 探索数据

数据探索中对数据进行初步研究，以便更好地解释它的特殊性质。有助于选择合适的数据预处理和数据分析技术。它甚至可以处理一些通常由数据挖掘解决的问题，例如，有时可以通过对数据进行直观检查来发现模式。此外，数据探索中使用可视化界面用于更好的理解和解释数据挖掘结果。数据探索界面如下：



【显示总行数】点击”显示总行数”按钮后，会在按钮的后面显示出所预览数据集节点的总行数。

【预览行数】数据集节点的默认显示行数。默认值为 1000 行。预览行数可以修改，修改后，点击空白处预览行数发生变化。

【统计】统计区域展示所选列的特征值。在左侧表内选择不同的列来显示不同列的特征值。

【可视化】可视化区域通过两种图表展示所选列的数据分析结果：直方图展示所选列的数据分布；盒须图展示所选列的数据范围和异常数据分布情况。当选择的是非数据类型的列时，不画图表。

直方图：

元数据 过滤数据 探索数据

显示总行数 预览行数: 1000

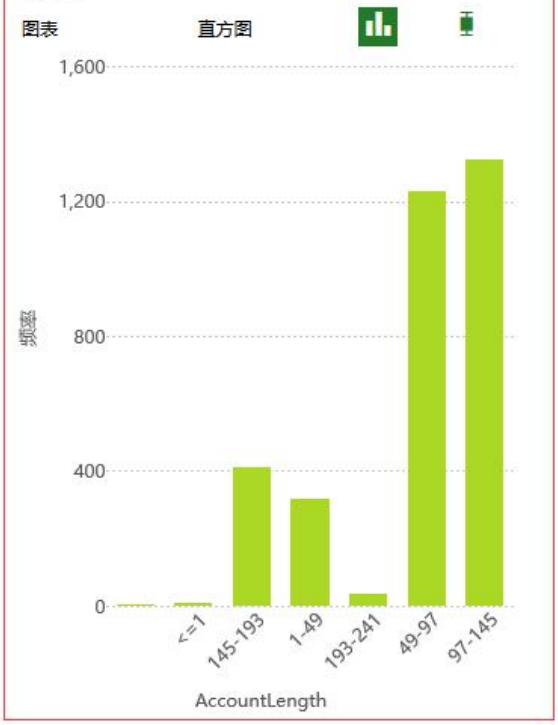
# AccountLeng	# AreaCode	# CustomerLe	Abc Internationa
128	415	0	no
107	415	0	no
137	415	0	no
84	408	0	yes
75	415	0	yes
118	510	0	yes
121	510	0	no
147	415	0	yes
117	408	0	no
141	415	0	yes
65	415	1	no
74	415	0	no
168	408	0	no
95	510	0	no
62	415	0	no
161	415	1	no
85	408	0	no
93	510	0	no
76	510	0	no
73	415	0	no
147	415	0	no

统计

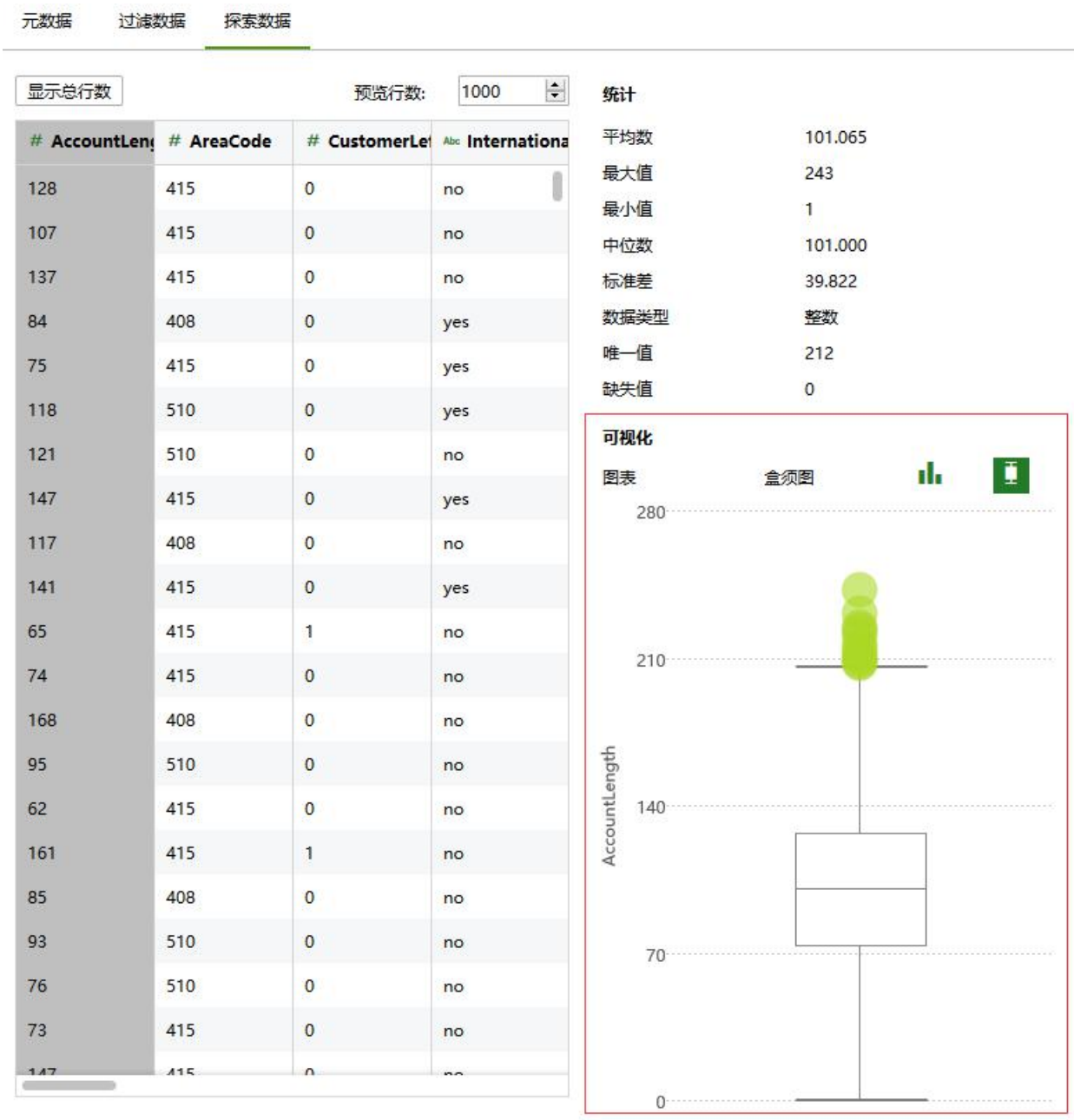
平均数	101.065
最大值	243
最小值	1
中位数	101.000
标准差	39.822
数据类型	整数
唯一值	212
缺失值	0

可视化

图表 直方图



盒须图：



3.2.3.1.2. 数据变换

❖ 数据变换

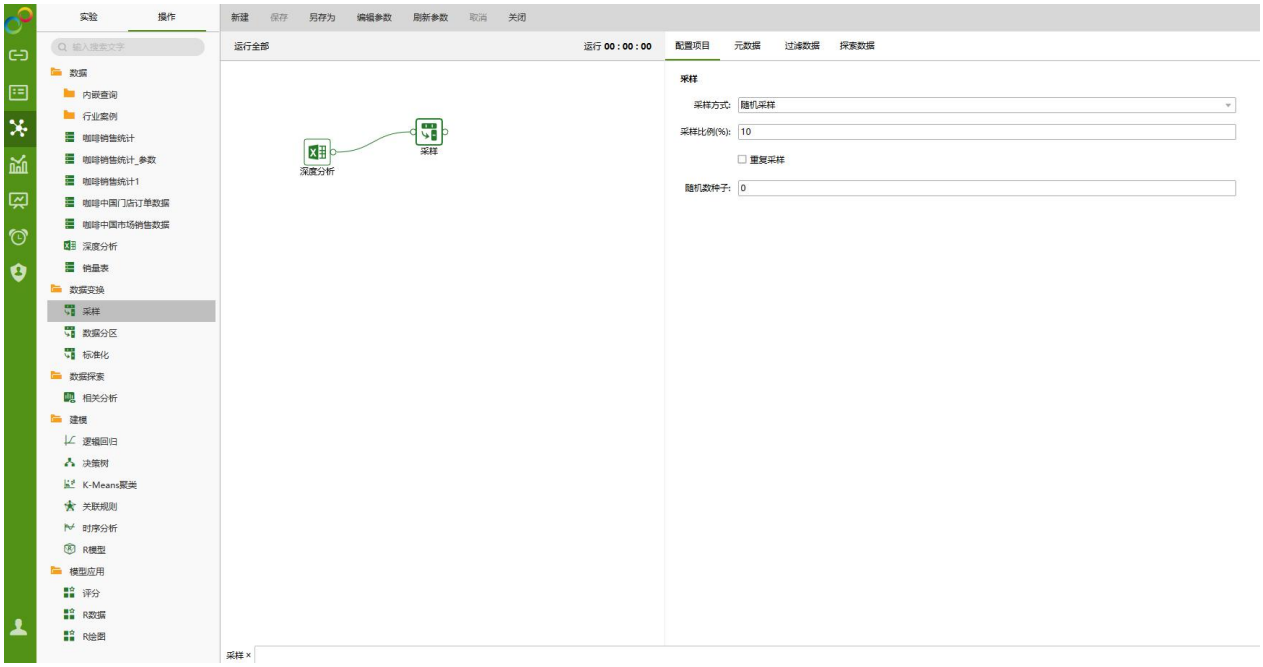
数据变换包含采样、数据分区、标准化。

• 采样

采样是一种选择数据对象子集进行分析的常用方法。在统计学中，采样长期用于数据的实现调查和最终的数据分析。在数据挖掘中，采样也非常有用。然而，在统计学和数据挖掘中，采样

的动机并不相同。统计学使用采样是因为得到感兴趣的整个数据集的费用太高、太费时间，而数据挖掘使用采样是因为处理所有的数据的费用太高，太费时间。在某些情况 下，使用采样的算法可以压缩数据量，以便可以使用更好但开销较大的数据挖掘算法。

拖拽一个数据集和一个采样节点到编辑区，连接数据集和采样节点。选中采样节点设置及展示区包含四个页面：配置项目、元数据、过滤数据、探索数据。



○ 配置项目

采样方式有三种：随机采样、按序采样、分层采样。

随机采样：

随机采样是按照随机的原则，即保证总体中每一个对象都有已知的、非零的概率被选入作为研究的对象，从数据集节点里抽取采样比例的样本行数，保证样本的代表性。

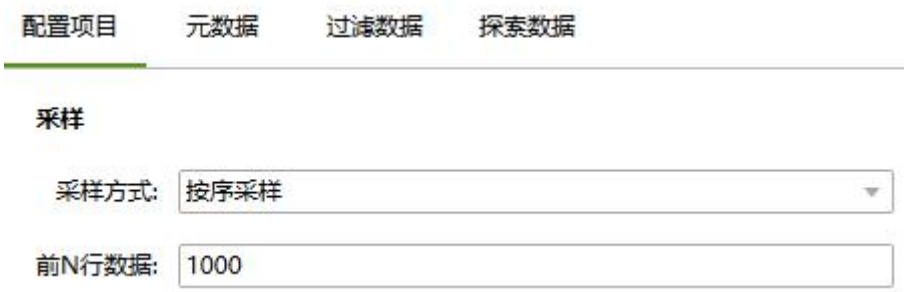
【采样比例】抽取样本的比例。

【重复采样】当不选中时，每个选中项立即从构成总体的所有对象集中删除。当选中时，对象被选中时不从总体中删除。当重复采样时，相同的对象可能被多次抽出。默认未选中。

【随机数种子】生成随机数的种子。默认值是 0。

按序采样：

按序采样是取数据集的前 N 行作为结果集。



【前 N 行数据】按序采样抽取样本的前多少行。默认值是 1000。

分层采样：

分层采样是从预先指定的组（即选择的列的不同值）开始抽样。每组按采样比例抽取。

【选择列】分层列，以它的不同值作为组按采样比例抽取样本行数。

其它参数请参看随机采样。

○ 元数据

请参考数据节点里的介绍。

○ 过滤数据

请参考数据节点里的介绍。

○ 探索数据

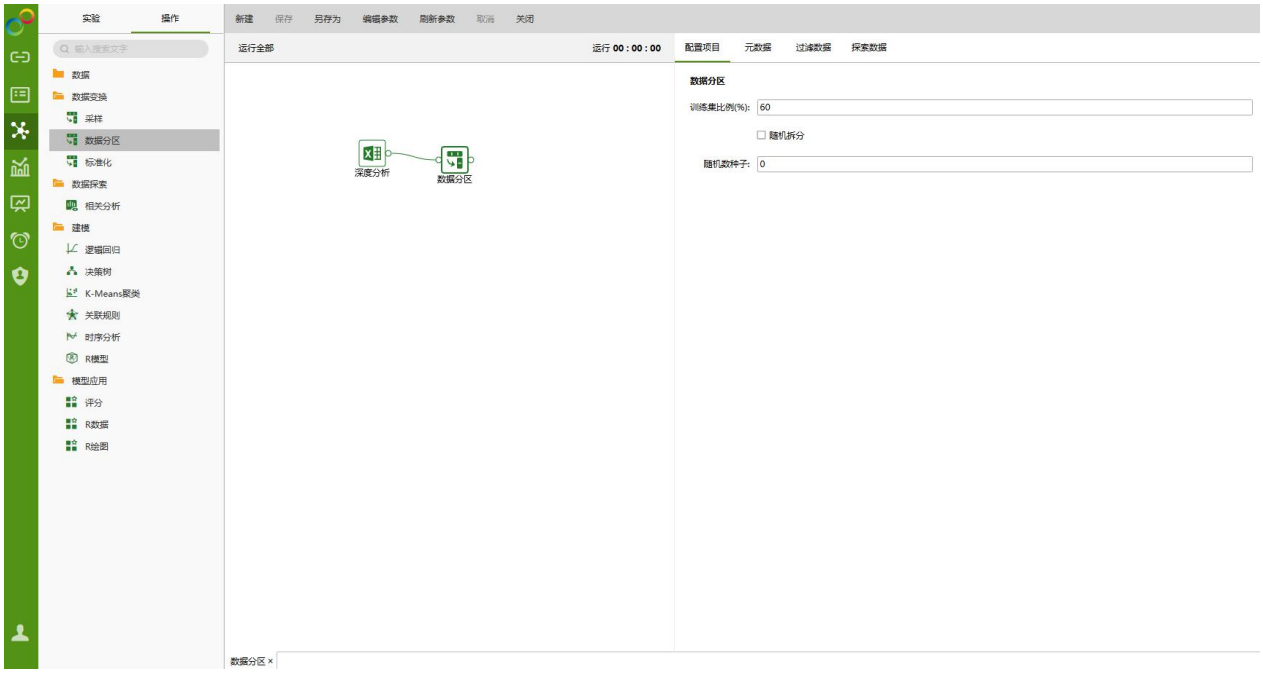
采样节点的全部数据是抽取的样本个数，其它详细信息请参考数据节点里的介绍。

• **数据分区**

一般做预测分析时，会将数据分为两大部分。一部分是训练数据，用于构建模型，一部分是测试数据，用于检验模型。数据分区就是把数据集节点的数据分为验证集和训练集。

拖拽一个数据集和一个数据分区节点到编辑区，连接数据集和数据分区节点。选中数据分区节

点设置及展示区包含四个页面：配置项目、元数据、过滤数据、探索数据。



○ 配置项目

【训练集比例(%)】训练集占总样本数的比例，默认值是 60。

【随机拆分】当不选中时按顺序抽取训练集。当选中时随机抽取训练集。默认未选中。

【随机数种子】生成随机数的种子。默认值是 0。

○ 元数据

请参考数据节点里的介绍。

○ 过滤数据

请参考数据节点里的介绍。

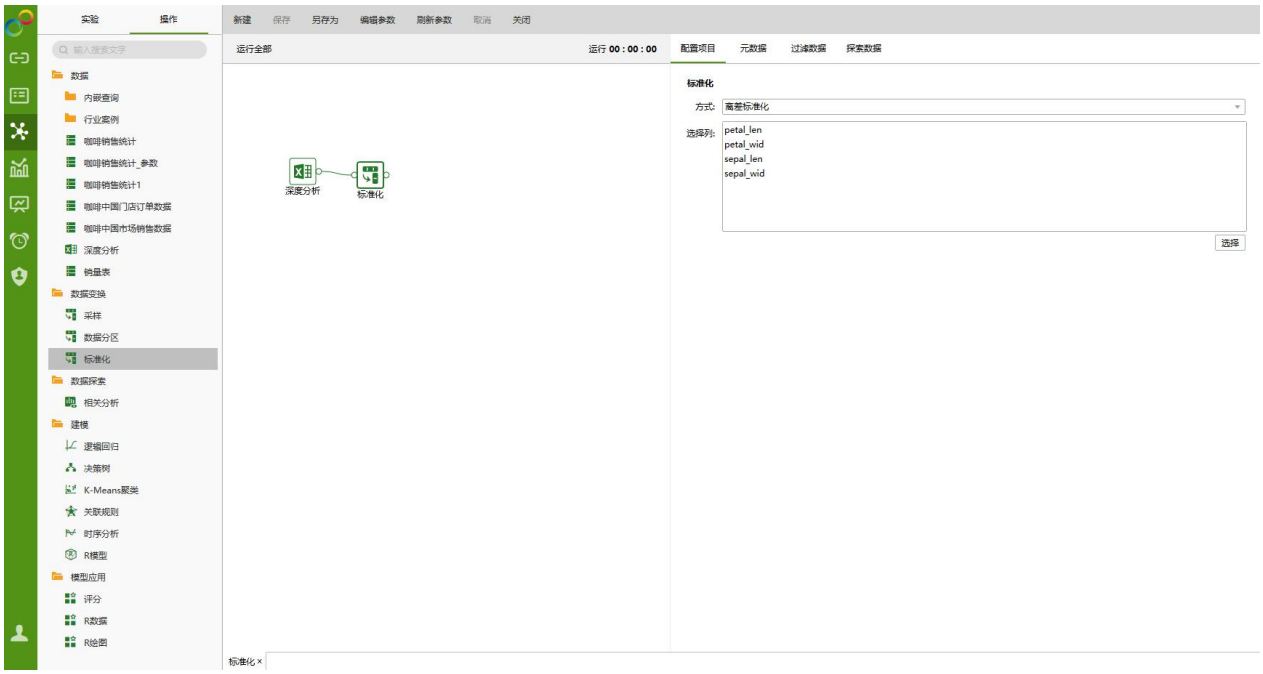
○ 探索数据

数据分区的探索数据中，可查看训练集和验证集的数据特征。其它详细信息请参考数据节点里的介绍。。

- 标准化

数据的标准化（normalization）是将数据按比例缩放，使之落入一个小的特定区间。在某些比较和评价的指标处理中经常会用到，去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。

拖拽一个数据集和一个标准化节点到编辑区，连接数据集和标准化节点。选中标准化节点设置及展示区包含四个页面：配置项目、元数据、过滤数据、探索数据。



○ 配置项目

【方式】标准化方式有两种：离差标准化，Z-Score 标准化。离差标准化是对原始数据的线性变换，使结果落到[0,1]区间。Z-Score 标准化处理的数据符合标准正态分布，即均值为 0，标准差为 1。

【选择列】需要被标准化的数据类型的列。

○ 元数据

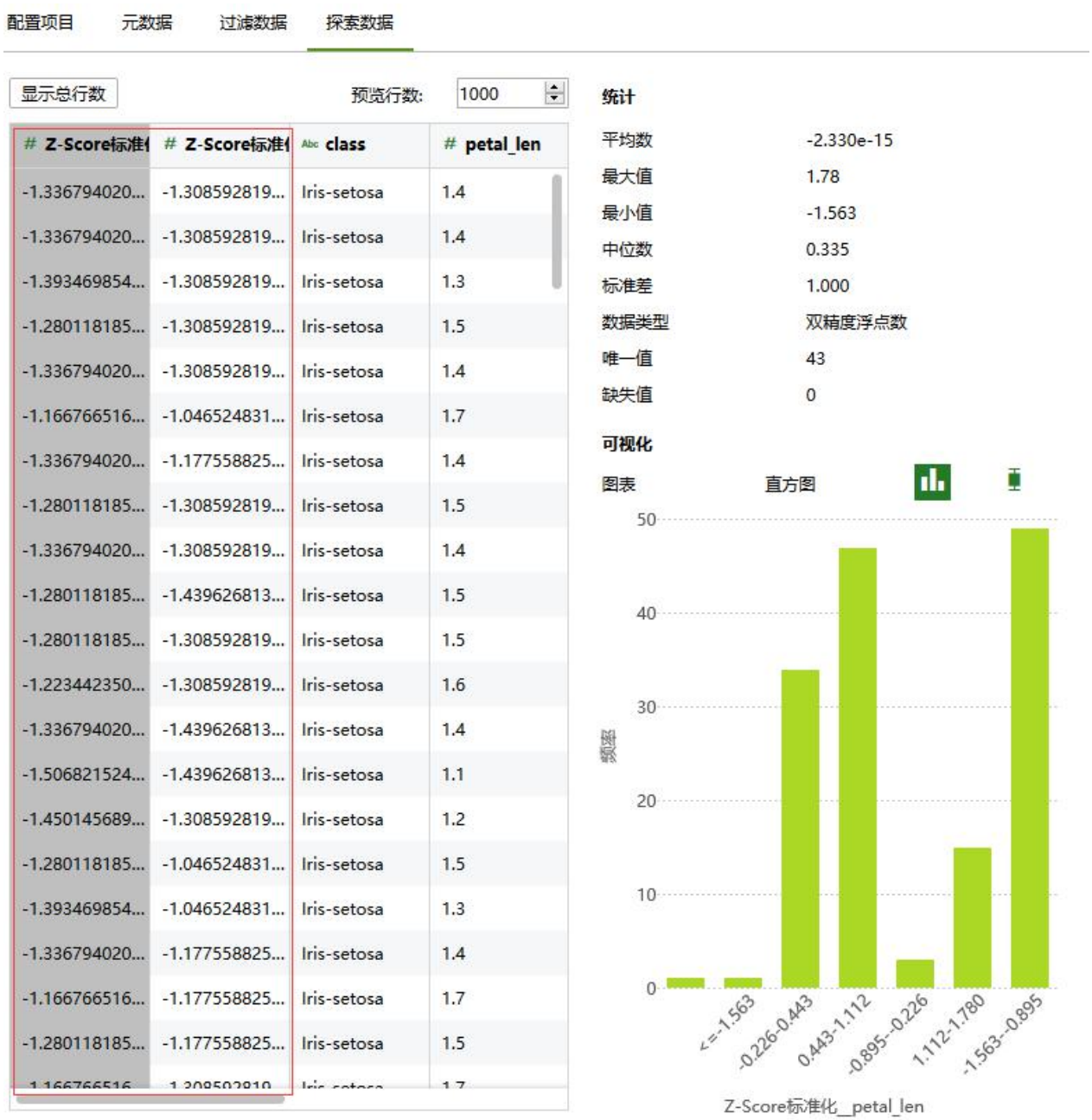
请参考数据节点里的介绍。

○ 过滤数据

请参考数据节点里的介绍。

○ 探索数据

数据预览区增加显示标准化后的列。其它详细信息请参考数据节点里的介绍。



3.2.3.1.3. 数据探索

❖ 数据探索

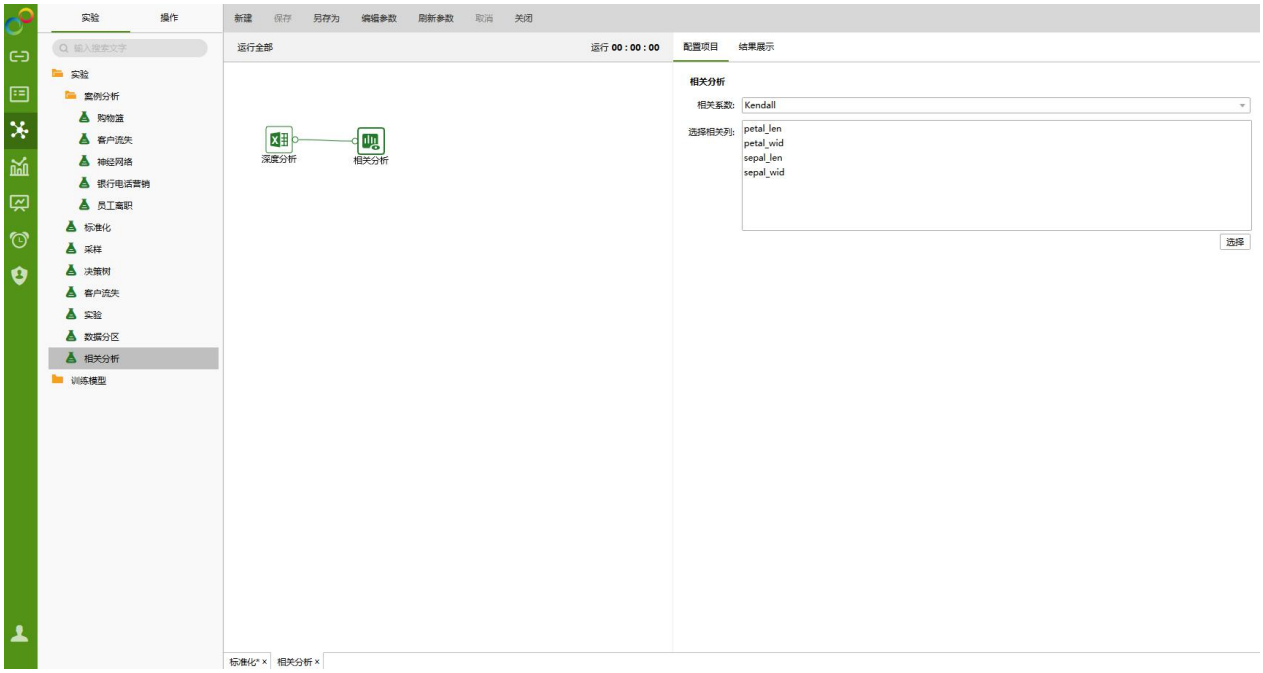
数据探索包含相关分析。

- 相关分析

相关分析（correlation analysis），相关分析是研究现象之间是否存在某种依存关系，并对具体有依存关系的现象探讨其相关方向以及相关程度，是研究随机变量之间的相关关系的一种统

计方法。

拖拽一个数据集和一个相关分析节点到编辑区，连接数据集和相关分析节点。选中相关分析节点设置及展示区包含两个页面：配置项目、结果展示。



○ 配置项目

【相关系数】相关系数有三种：Pearson，Kendall，Spearman。

- Pearson 相关系数（Pearson Correlation Coefficient）是用来衡量两个数据集合是否在一条线上面，它用来衡量定距变量间的线性关系。
- Kendall 系数是 n 个同类的统计对象按特定属性排序，其他属性通常是乱序的。同序对（concordant pairs）和异序对（discordant pairs）之差与总对数（n*(n-1)/2)的比值。
- Spearman 相关系数是衡量分级定序变量之间的相关程度的统计量。

【选择相关列】点击选择按钮打开选择列对话框，从左边拖拽列到右边，添加相关列。

○ 结果展示

两个变量之间的相关程度通过相关系数 r 来表示。相关系数 r 的值在-1 和 1 之间，可以是此范围内的任何值。正相关时，r 值在 0 和 1 之间；负相关时，r 值在-1 和 0 之间。r 的绝对值越接近 1，两变量的关联程度越强，r 的绝对值越接近 0，两变量的关联程度越弱。

相关分析的结果以矩阵形式展示出来。如下图，蓝色表示正相关，粉色表示负相关，颜色越深相关性越强。

	petal_len	petal_wid	sepal_len	sepal_wid
petal_len	1	0.803	0.718	-0.182
petal_wid	0.803	1	0.655	-0.147
sepal_len	0.718	0.655	1	-0.072
sepal_wid	-0.182	-0.147	-0.072	1

注：相关系数的大小说明：|r|>0.95 存在显著性相关；|r|≥0.8 高度相关；0.5≤|r|<0.8 中度相关；0.3≤|r|<0.5 低度相关；|r|<0.3 关系极弱，认为不相关。

3.2.3.1.4. 建模

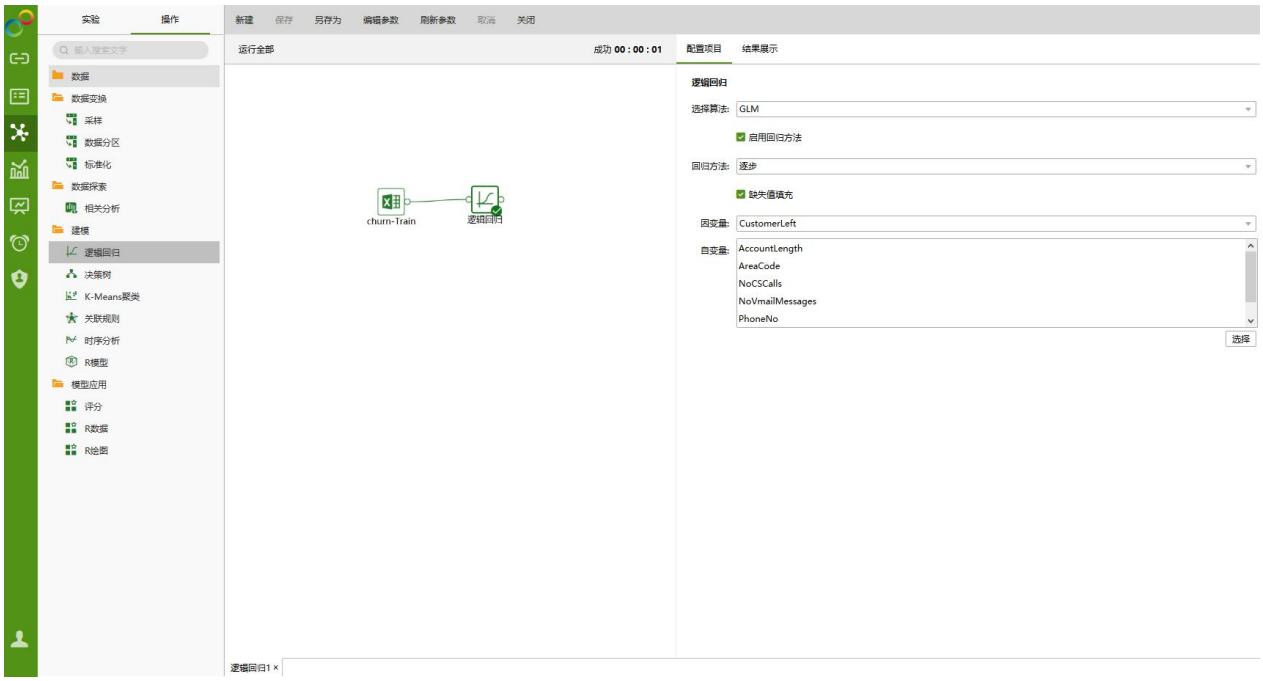
❖ 建模

建模包含逻辑回归、决策树、K-Means 聚类、关联规则、时序分析、R 模型。

• 逻辑回归

逻辑回归（Logistic Regression）是机器学习中的一种分类模型，由于算法的简单和高效，在实际中应用非常广泛。

拖拽一个数据集和一个逻辑回归节点到编辑区，连接数据集和逻辑回归节点。选中逻辑回归节点设置及展示区包含两个页面：配置项目、结果展示。

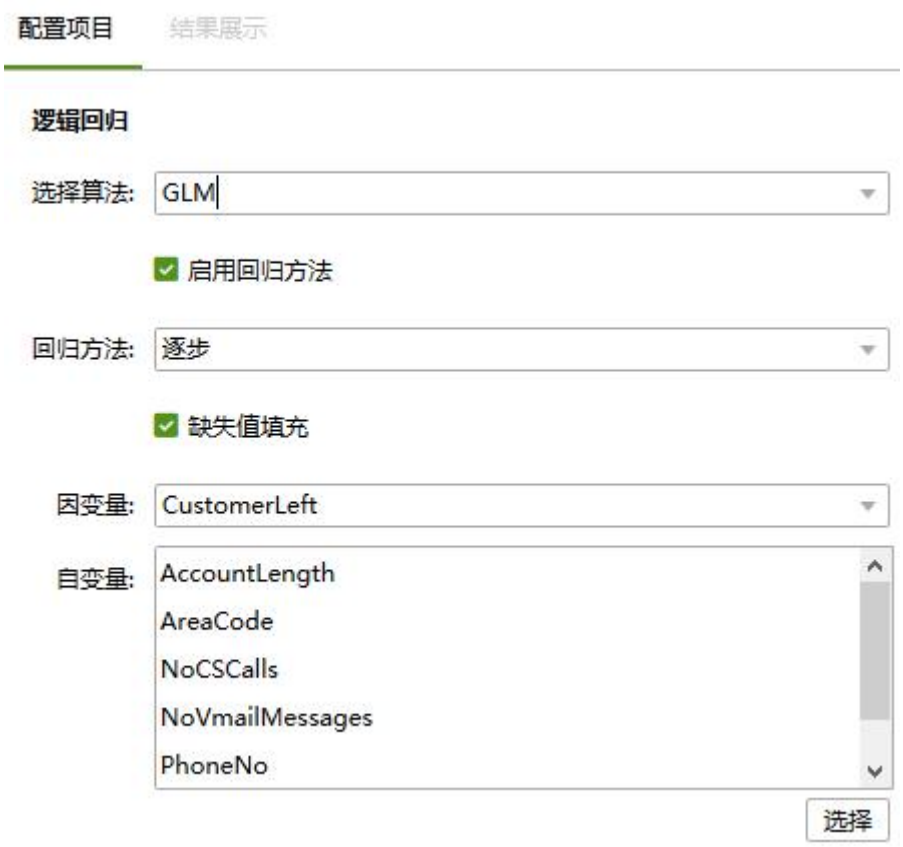


○ 配置项目

逻辑回归有两种算法：GLM，GLMNET，默认是 GLM。

GLM：

广义线性模型（GLM），这种模型是把自变量的线性预测函数当作因变量的估计值。常用于逻辑回归中。



【启用回归方法】控制是否使用回归方法。默认选中。

【回归方法】包含逐步、向前、向后。默认选中逐步。

逐步：就是分步构建方程式。初始模型是尽可能简单的模型，其方程式中不含任何输入字段。在每个步骤中，对尚未添加到模型的输入字段进行评估，如果其中的最佳输入字段能够显著增加模型预测能力，那么将该字段添加到模型中。此外，还会重新评估当前包含在模型中的输入字段，以确定能否在不对模型功能造成重大减损的情况下删除其中任何字段。如果可以，则会将其删除。然后重复此过程，添加或删除其他字段。当无法再添加任何字段来改进模型、且无法再删除任何字段而不对模型功能造成减损时，最终模型便已生成。

向后：与分步构建模型的逐步法类似。但是，采用这种方法时，初始模型包含作为预测变量的所有输入字段，只能从模型中删除字段。对模型影响较小的输入字段将被逐一删除，直到无法再删除任何字段而不对模型功能造成重大损害，从而生成最终模型。

向前：向前与向后是相反的回归方法。采用这种方法，初始模型是最简单的模型，不包含任何输入字段，只能向模型中添加字段。每个步骤会对尚未纳入到模型中的输入字段进行检验，看它们对模型的改进起多大作用，然后将其中的最佳字段添加到模型中。当无法再添加任何字

段、或最佳备选字段无法对模型产生足够的改进时，最终模型便已生成。

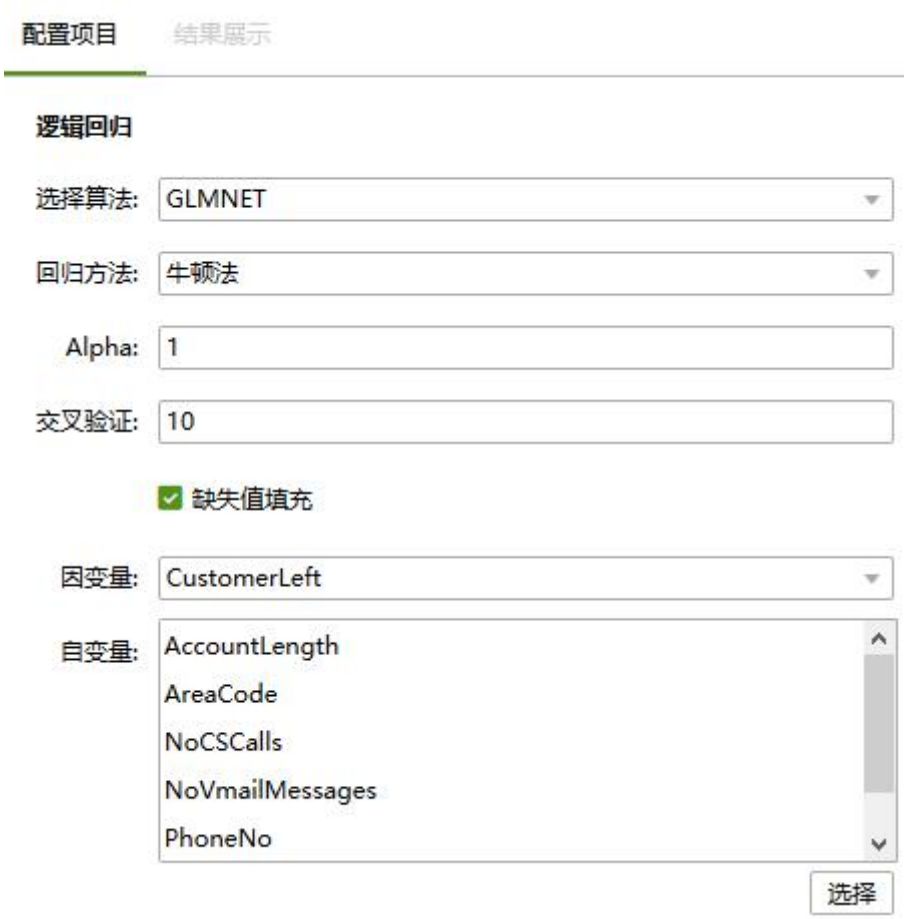
【缺失值填充】用自变量列平均值填充此列的空值。默认是填充的。

【因变量】从下拉列表中选出需要作为因变量的字段。任何一个系统（或模型）都是由各种变量构成的，当我们分析这些系统（或模型）时，可以选择研究其中一些变量对另一些变量的影响，那么我们选择的这些变量就称为自变量，而被影响的量就被称为因变量。

【自变量】从选择列对话框中选出需要作为自变量的字段。

GLMNET：

GLMNET 使用 Lasso 、 Elastic-Net 等正则化方式来实现逻辑回归。



【回归方法】包含牛顿法、拟牛顿法。默认选中牛顿法。

牛顿法：数值优化的一种方法，利用函数在当前点的一阶导数，以及二阶导数，寻找搜寻方向。

拟牛顿法： 牛顿法的变形，用近似矩阵替代牛顿法中的海森矩阵。

【Alpha】Elastic-Net 混合参数。范围是 0 到 1。等于 1 时，惩罚项采用 L1 范数；等于 0 时，惩罚项采用 L2 范数。默认值为 1。

【交叉验证】通过交叉验证可以得到最优的方程。默认值是 10。

○ 结果展示

1. 模型系数

GLM 算法通过模型系数可以得到逻辑回归方程的系数，包括截距项，各自变量的系数以及它们的 P 值，标准误差。还可以看到模型训练后的准确率和均方误差。如果做了数据分区还能看到基于验证集的模型准确率和均方误差。以下是 GLM 是算法的结果。如果系数为 0，结果就不会显示在模型系数表格里。

2. ROC 曲线

绘制 ROC 曲线计算 AUC 值，AUC 值越大模型分类效果越好。如果做了数据分区还可以比较验证集的 AUC 值。

配置项目　结果展示

1.模型系数

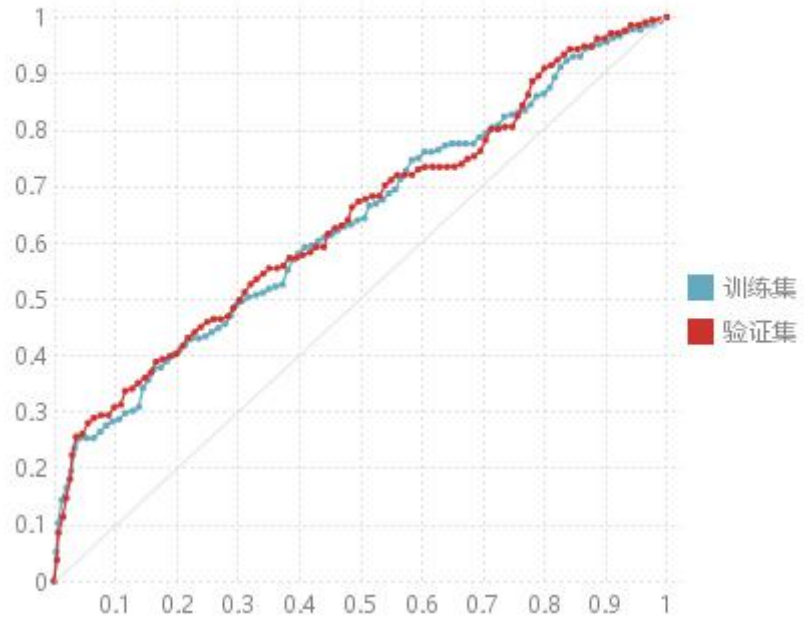
训练集： 准确率: 0.866　　均方误差: 219.725

验证集： 准确率: 0.843　　均方误差: 165.678

	截距项	NoCSCalls	NoVmailMessa
回归系数	-2.440	0.406	-0.020
标准误差	0.118	0.046	0.005
P值	1.507e-95	5.083e-19	3.670e-4

2.ROC曲线

训练集的AUC值: 0.635　　验证集的AUC值: 0.641



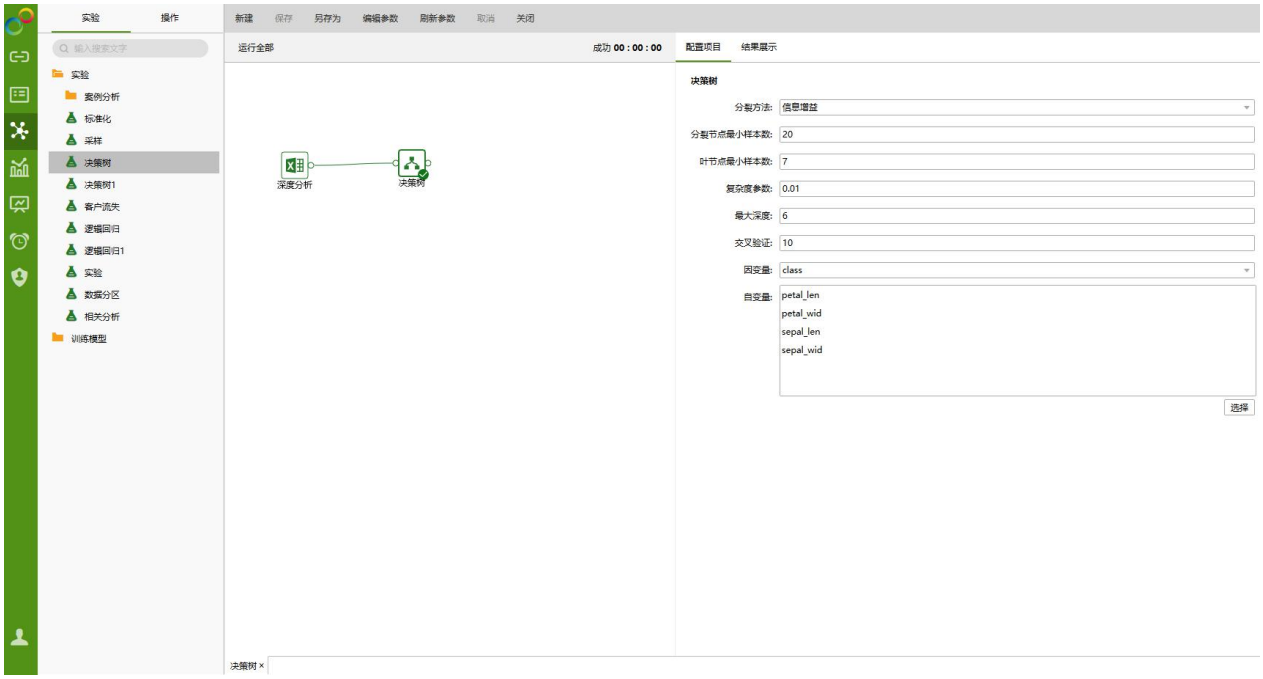
- 决策树

决策树是一种用于对实例进行分类的树形结构。决策树由节点（node）和有向边（directed edge）组成。节点的类型有两种：内部节点和叶子节点。其中，内部节点表示一个特征或属

性的测试条件（用于分开具有不同特性的记录），叶子节点表示一个分类。

一旦我们构造了一个决策树模型，以它为基础来进行分类将是非常容易的。具体做法是，从根节点开始，用实例的某一特征进行测试，根据测试结构将实例分配到其子节点（也就是选择适当的分支）；沿着该分支可能达到叶子节点或者到达另一个内部节点时，那么就使用新的测试条件递归执行下去，直到抵达一个叶子节点。当到达叶子节点时，我们便得到了最终的分类结果。

拖拽一个数据集和一个决策树节点到编辑区，连接数据集和决策树节点。选中决策树节点设置及展示区包含两个页面：配置项目、结果展示。



○ 配置项目

【分裂方法】包含信息增益和 Gini 系数。

信息增益：划分前后，子集纯度的提升值。

Gini 系数：在样本集合中一个随机选中的样本被分错的概率。

【分裂节点最小样本数】样本数小于该值该节点将不进行分裂。默认值 20。

【叶节点最小样本数】样本数小于该值，该分支将被剪除。默认值 7。

【复杂度参数】每次分裂会计算一个复杂度，当复杂度大于该参数值不再进行分裂。默认值 0.01。

【最大深度】决策树的最层次数。默认值 6。

【交叉验证】通过交叉验证可以得到最优的方程。默认值是 10。

【因变量】从下拉列表中选出需要作为因变量的字段。任何一个系统（或模型）都是由各种变量构成的，当我们分析这些系统（或模型）时，可以选择研究其中一些变量对另一些变量的影响，那么我们选择的这些变量就称为自变量，而被影响的量就被称为因变量。

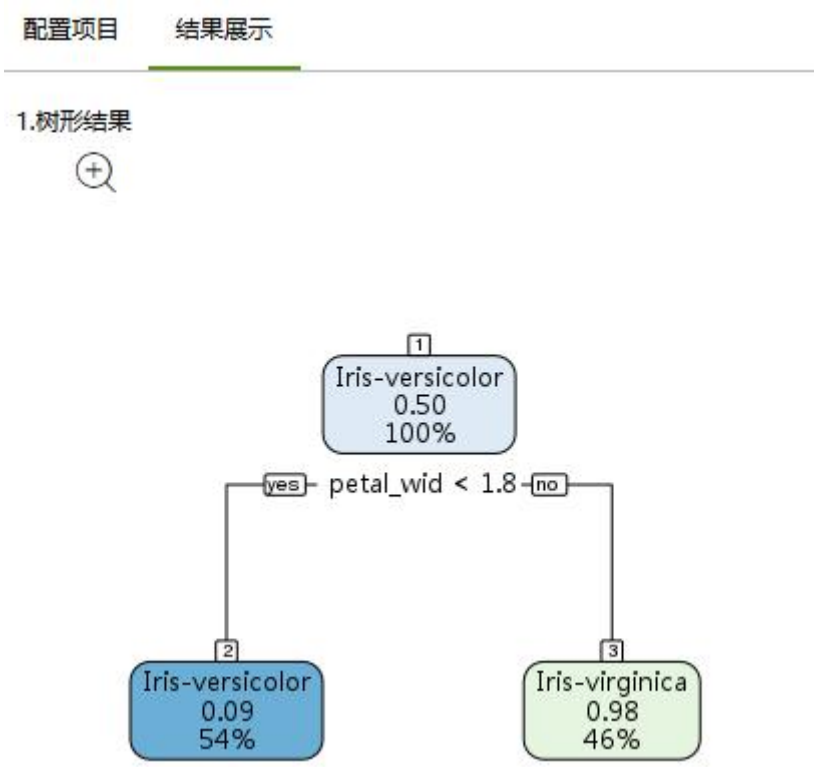
【自变量】从选择列对话框中选出需要作为自变量的字段。

○ 结果展示

1. 树形结果

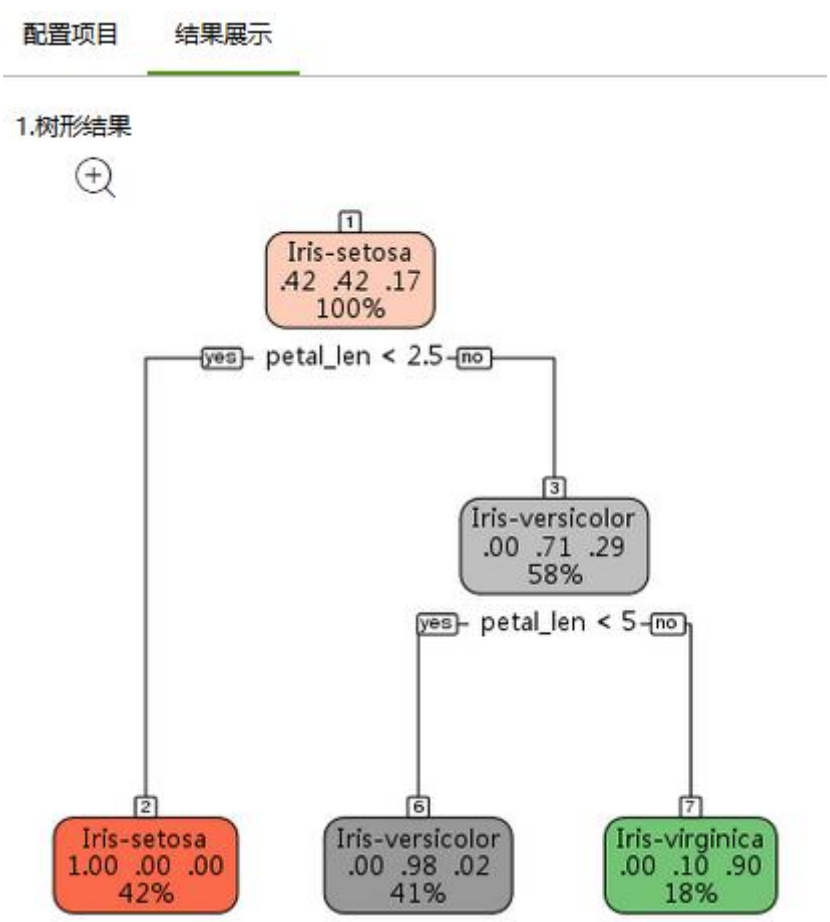
分类结果有两种，二分类和多分类。当因变量只有两个不同值时是二分类；当因变量有多于两个不同值时是多分类。节点最上方是节点编号。节点内有三行，第一行是该节点最终分类；当为多分类时第二行是该节点各种分类的概率值，当为二分类时第二行是该节点是主分类的概率值；第三行是样本行数的占比。yes 和 no 代表是否满足条件，来确定分支方向。节点颜色代表纯度。

二分类树形结果展示如下：



注意：主分类是绿色节点，蓝色节点的 0.09 表示 Iris-versicolor 是 Iris-virginica 的概率值。

多分类树形结果展示如下：



点击放大按钮，可放大图片以更清晰的查看图片。

2. 节点分类情况

列出所有叶子节点的信息。

2.节点分类情况

节点编号	类别	判别条件	节点样本数
2	Iris-setosa	petal_len < 2....	50
6	Iris-versicolor	petal_len >= ... petal_len < 4....	49
7	Iris-virginica	petal_len >= ... petal_len >= ...	21

【节点编号】叶子节点的编号。

【类别】叶子节点的类别。

【判别条件】从根部到该叶子节点的判别条件。

【节点样本数】叶子节点的样本数。

【误分样本数】错误分类的样本数。

【复杂度参数】每个叶子节点的复杂度参数。

3. 混淆矩阵

预测结果的情形分析表，当有数据分区时还可查看验证集的分析表。

横向表头是真实值，纵向表头是预测值。整数值代表样本个数，百分数是样本个数占总样本个数的比例。精确度为真实值和预测值相同的比例总和。

3.混淆矩阵

训练集的精确度: 0.975

预测值 \ 真实值	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	50(41.67%)	0(0.00%)	0(0.00%)
Iris-versicolor	0(0.00%)	48(40.00%)	1(0.83%)
Iris-virginica	0(0.00%)	2(1.67%)	19(15.83%)

验证集的精确度: 0.833

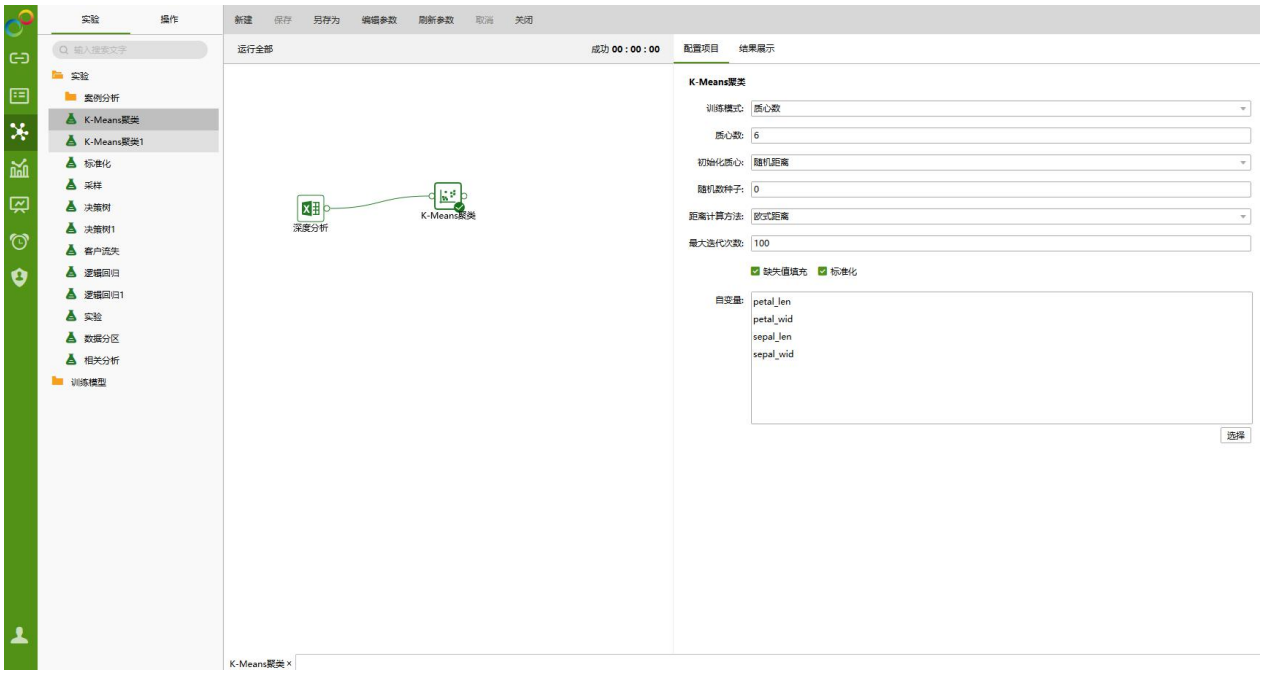
预测值 \ 真实值	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	0(0.00%)	0(0.00%)	0(0.00%)
Iris-versicolor	0(0.00%)	0(0.00%)	5(16.67%)
Iris-virginica	0(0.00%)	0(0.00%)	25(83.33%)

- **K-Means 聚类**

K-Means 是聚类算法中的一种，其中 K 表示类别数，Means 表示均值。顾名思义 K-Means 是一种通过均值对数据点进行聚类的算法。K-Means 算法通过预先设定的 K 值及每个类别的初始质心对相似的数据点进行划分，并通过划分后的均值迭代优化获得最优的聚类结果。

为了提升 K-Means 聚类的计算效率，产品支持分布式系统计算 K-Means。当输入节点数据集是“数据集市数据集”时就是分布式计算的。

拖拽一个数据集和一个 K-Means 聚类节点到编辑区，连接数据集和 K-Means 聚类节点。选中 K-Means 聚类节点设置及展示区包含两个页面：配置项目、结果展示。



○ 配置项目

【训练模式】包含质心数、质心数范围。

【质心数】质心的个数。

【质心数范围】质心个数的范围。

【初始化质心】初始化质心的方法包括：随机距离、Kmeans++。随机距离是所有质心都是随机选取的。Kmean++是第一个质心是随机选取，其它质心按距离选取，距离其它质心越远被选中的概率越大。

【随机数种子】生成随机数的种子。默认值是 0。

【距离计算方法】包括两张方法：欧式距离、余弦距离。欧式距离是两个数据点的实际距离。余弦距离是用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。

【最大迭代次数】迭代计算的最大次数，最终算出稳定的质心数。默认值 100。

【缺失值填充】用自变量列平均值填充此列的空值。默认是填充的。

【标准化】对自变量标准化，默认标准化方式是 Z-Score 标准化。

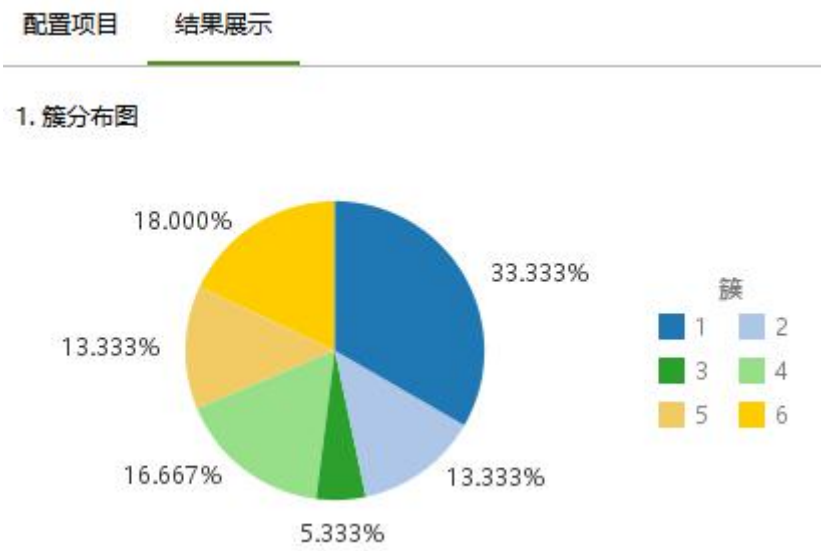
【自变量】从选择列对话框中选出需要作为自变量的字段。

○ 结果展示

质心数为 6，样本个数 150，K-Means 聚类展示结果如下：

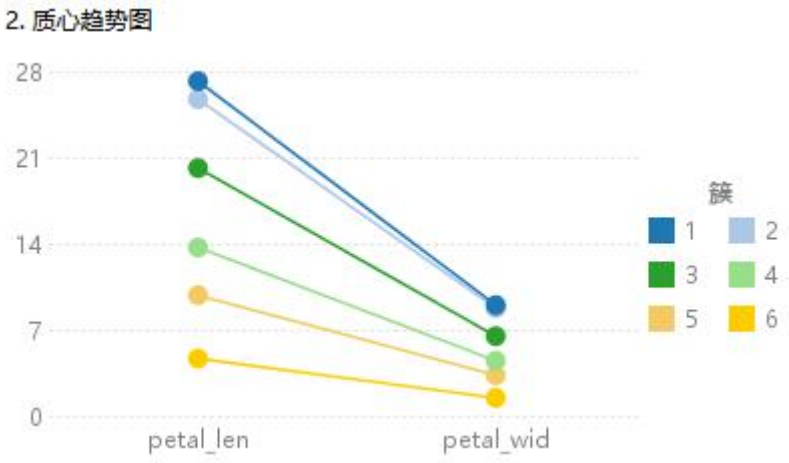
1. 簇分布图

簇内的样本个数占总样本个数的比例。



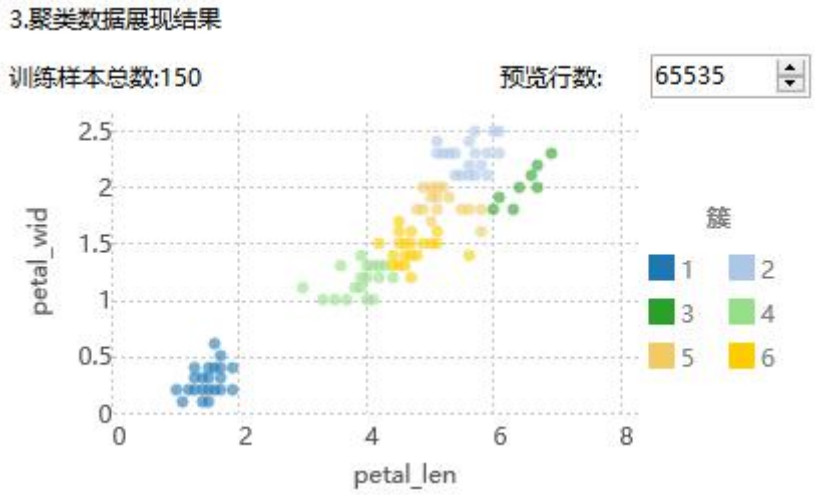
2. 质心趋势图

各个质心在自变量上的变化趋势。



3. 聚类数据展现结果

根据前两个列绘制的聚类之后的散点图。



【预览行数】图表默认展示 65535 行数据，可修改此值改变预览行数。

4. k 均值聚类质心

质心在自变量上的取值。

4.k均值聚类质心

质心数:6

簇编号	簇内样本数	petal_len	petal_wid
1	50	1.464	0.244
2	20	5.615	2.285
3	8	6.463	2.013
4	25	3.892	1.180
5	20	5.160	1.845
6	27	4.670	1.459

5. 簇成员

样本分别属于哪个簇，距离质心的距离。

5.簇成员

训练样本总数:150　　预览行数: 1000

petal_len	petal_wid	簇	距离
1.4	0.2	1	0.068
1.4	0.2	1	0.068
1.3	0.2	1	0.109
1.5	0.2	1	0.061
1.4	0.2	1	0.068
1.7	0.4	1	0.244
1.4	0.3	1	0.082
1.5	0.2	1	0.061
1.4	0.2	1	0.068
1.5	0.1	1	0.190

【预览行数】默认预览行数是 1000，可修改预览行数。

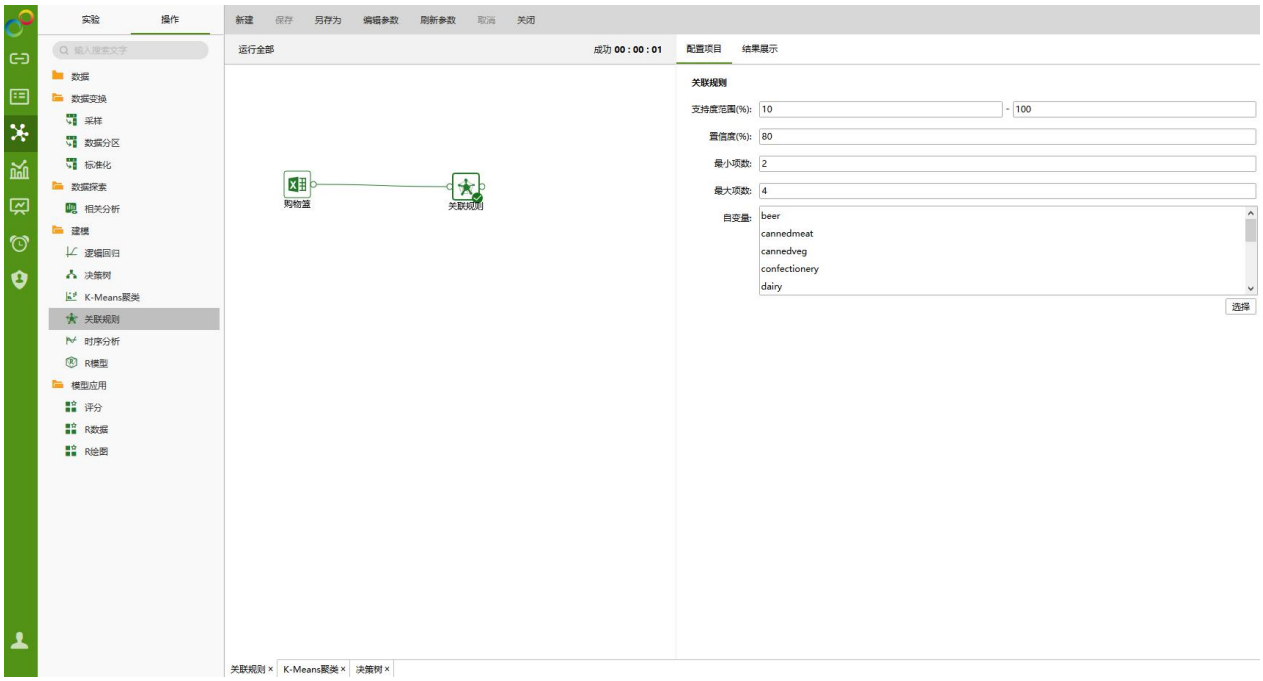
【簇】分类编号。

【距离】不同距离计算方法计算出的每条样本到最近质心的距离值。

- 关联规则

关联规则（Association Rules）是无监督的机器学习方法，从数据背后发现事物之间可能存在的关联或者联系，用于知识发现，而非预测。这种事物之间的关联或者联系就叫规则。

拖拽一个数据集和一个关联规则节点到编辑区，连接数据集和关联规则节点。选中关联规则节点设置及展示区包含两个页面：配置项目、结果展示。



○ 配置项目

【支持度范围(%)】模型所生成的规则的支持度级别的百分比值范围。如果不在此范围内规则将被废弃。

【置信度(%)】模型所生成的规则的置信度级别的最小百分比值。如果模型所生成的规则的置信度级别小于此数量，那么该规则将被废弃。

【最小项数】模型所生成的规则的最小项数，小于此值将被废弃。

【最大项数】模型所生成的规则的最大项数，大于此值将被废弃。

【自变量】从选择列对话框中选出需要作为自变量的字段。

○ 结果展示

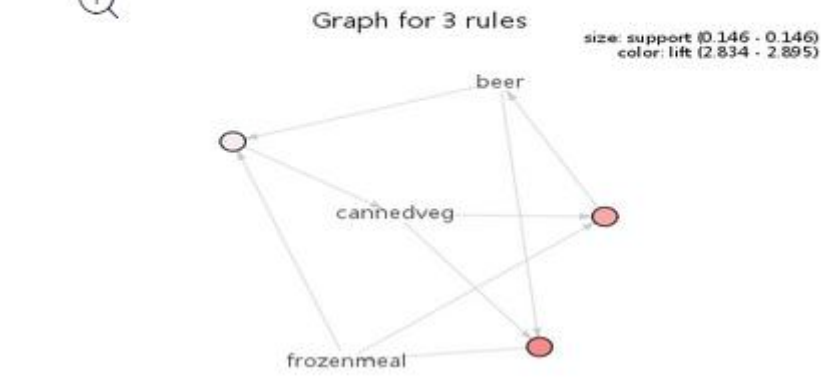
1. 关联规则图

各项的关联关系图，每个圆圈代表一条规则，指向圆圈的是左项，圆圈指向的是右项；圆圈大小代表支持度大小，圆圈越大支持度越大，圆圈颜色代表提升度，颜色越深提升度越大。

点击放大按钮，可放大图片以更清晰的查看图片。

配置项目　结果展示

1.关联规则图



2. 关联规则

2.关联规则

左项	右项	支持度	置信度
[beer, frozen...	[cannedveg]	0.146	0.859
[beer, canned...	[frozenmeal]	0.146	0.874
[cannedveg, f...	[beer]	0.146	0.844
[beer, frozen...	[confectioner...	0.136	0.800
[beer, canned...	[confectioner...	0.137	0.820
[cannedveg, f...	[confectioner...	0.143	0.827
[beer, canned...	[confectioner...	0.120	0.822
[beer, confect...	[cannedveg]	0.120	0.882
[beer, canned...	[frozenmeal]	0.120	0.876

【左项】规则的先导项集。

【右项】规则的结论项集。

【支持度】项集出现的次数除以总的记录数。

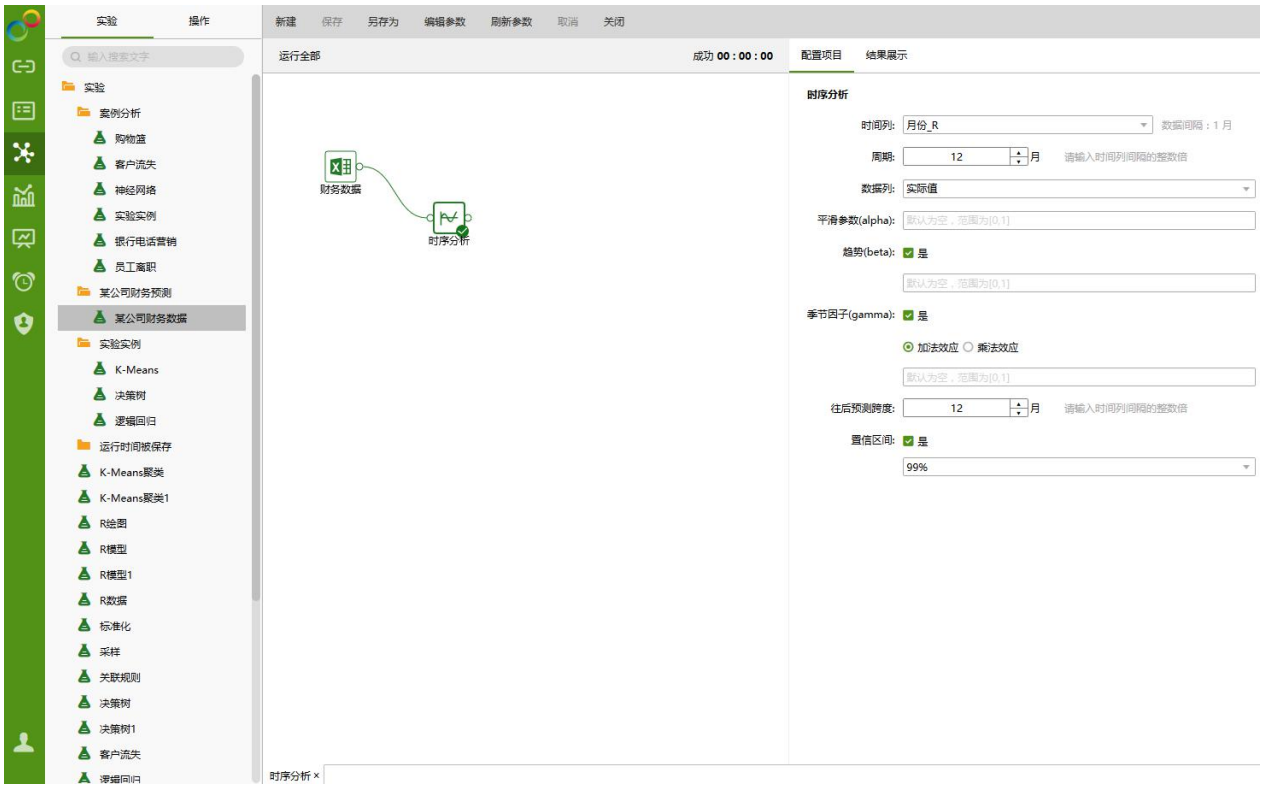
【置信度】项集{X，Y}同时出现的次数占项集{X}出现次数的比例。

【提升度】度量项集{X}和项集{Y}的独立性。数值越大模型越好。

- 时序分析

时序分析通过考虑水平趋势和季节性趋势，对一段时间内、等时间间隔的采样数据进行分析，以预测未来一段时间的数据。即根据已知的历史数据，预测未来的数据。

拖拽一个数据集和一个时序分析节点到编辑区，连接数据集和时序分析节点。选中时序分析节点设置及展示区包含两个页面：配置项目、结果展示。



○ 配置项目

【时间列】选择时间字段。根据选择的时间字段的数据，自动算出时间间隔。

【周期】需填入时间间隔的整数倍，根据周期和时间间隔（周期/ 时间间隔）算出频率，即单位时间内的观测数。根据时间间隔，系统会自动往周期填入一个合理的数值，此数值也可手动修改。

【数据列】选择数据字段。

【平滑参数(alpha)】α越接近 1，平滑后的值越接近当前时间的数据值。

【趋势(beta)】是否考虑纵向趋势。默认是被勾选，表示按纵向趋势拟合。

【季节因子(gamma)】是否考虑季节性趋势。如果设置为不勾选（FALSE），则非季节性模型拟合。如果设置为勾选，则进行季节性模型拟合。季节性模式可以是加法效应（additive）和乘法效应（multiplicative）。加法效应默认勾选，表示按季节性加法的趋势增长。乘法效应被勾选时，表示按季节性乘法趋势增长。季节性模型拟合时，需满足一个周期内至少有两个数据点，即频率大于等于 2，且时间序列至少包含 2 个周期。

【往后预测跨度】往后预测的时间跨度，需填入时间间隔的整数倍。选择时间列后，系统会自动填入一个合理的数值，此数值也可进行手动修改。

【置信区间】根据 Level 算出估计值的上界和下界，默认 Level 是 99%。

○ 结果展示

1. 季节模型预测曲线

【历史拟合图】历史数据变化曲线和拟合数据变化曲线。绿色线是历史数据变化曲线，黑色线是拟合数据变化曲线。两条线重合度越高拟合的越好。

【预测图】对未来指定时间段内的预测。绿色线是历史数据变化曲线，黑色线是对指定时间段内预测的变化曲线。由此可以看出未来数据的变化趋势。

【预测图+置信区间图】对未来指定时间段内的预测。深绿色线是历史数据变化曲线，黑色线指定时间段内预测的变化曲线。浅绿色线是置信区间的上界，橙色线是置信区间的下界。

2. 模型统计量

【MSE】平方误差的均值。越小越好。

【LBQ 检验】时间序列是否存在滞后相关的一种统计检验。显著性小于 0.05 时说明误差存在明显的自相关性，表示模型拟合不良。显著性值越接近 1 模型拟合越好。

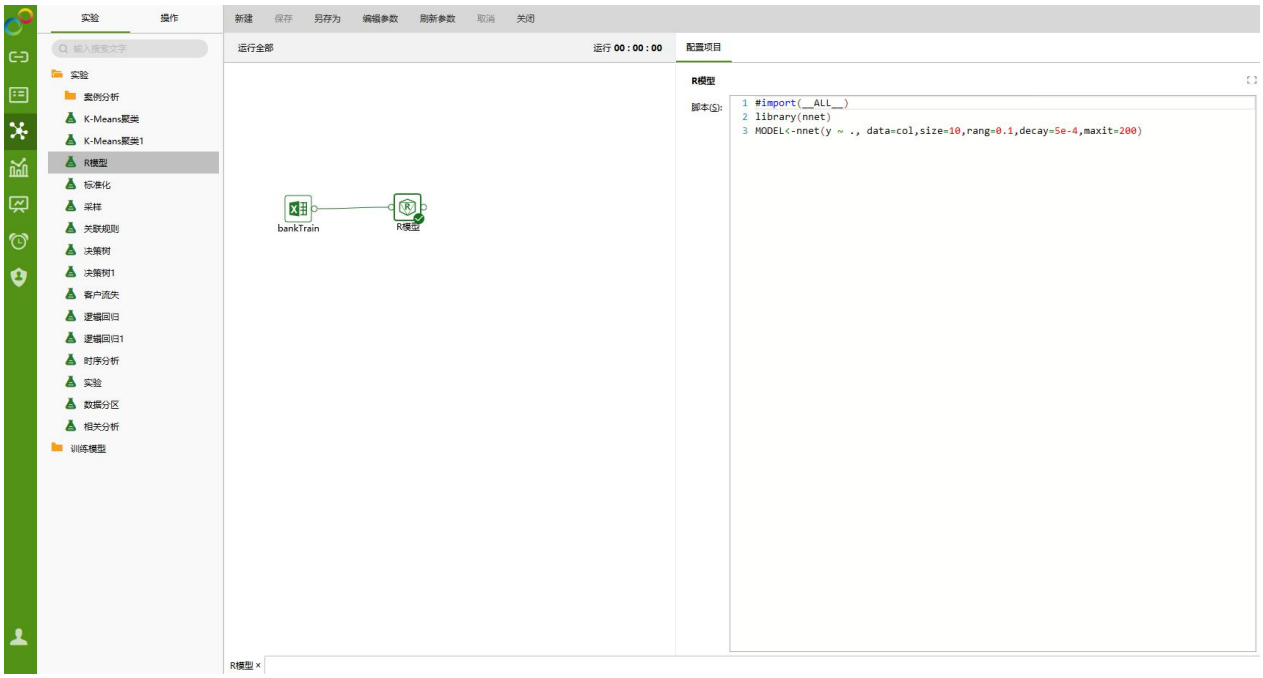
2.模型统计量

MSE	LBQ检验		
	统计量	显著性	自由度
26718743914.824	20.173	0.323	18

- **R 模型**

用户可以使用 R 模型节点引用任意 R package 来连接 R 数据或者 R 绘图进行分析。

拖拽一个数据集和一个 R 模型节点到编辑区，连接数据集和 R 模型节点。选中 R 模型节点设置及展示区包含一个页面：配置项目。



○ 配置项目

【脚本】引用 R package 的脚本，必须将最终结果存入 MODEL 变量中作为返回值。

点击右上按钮，可弹出脚本输入框更方便的输入。

3.2.3.1.5. 模型应用

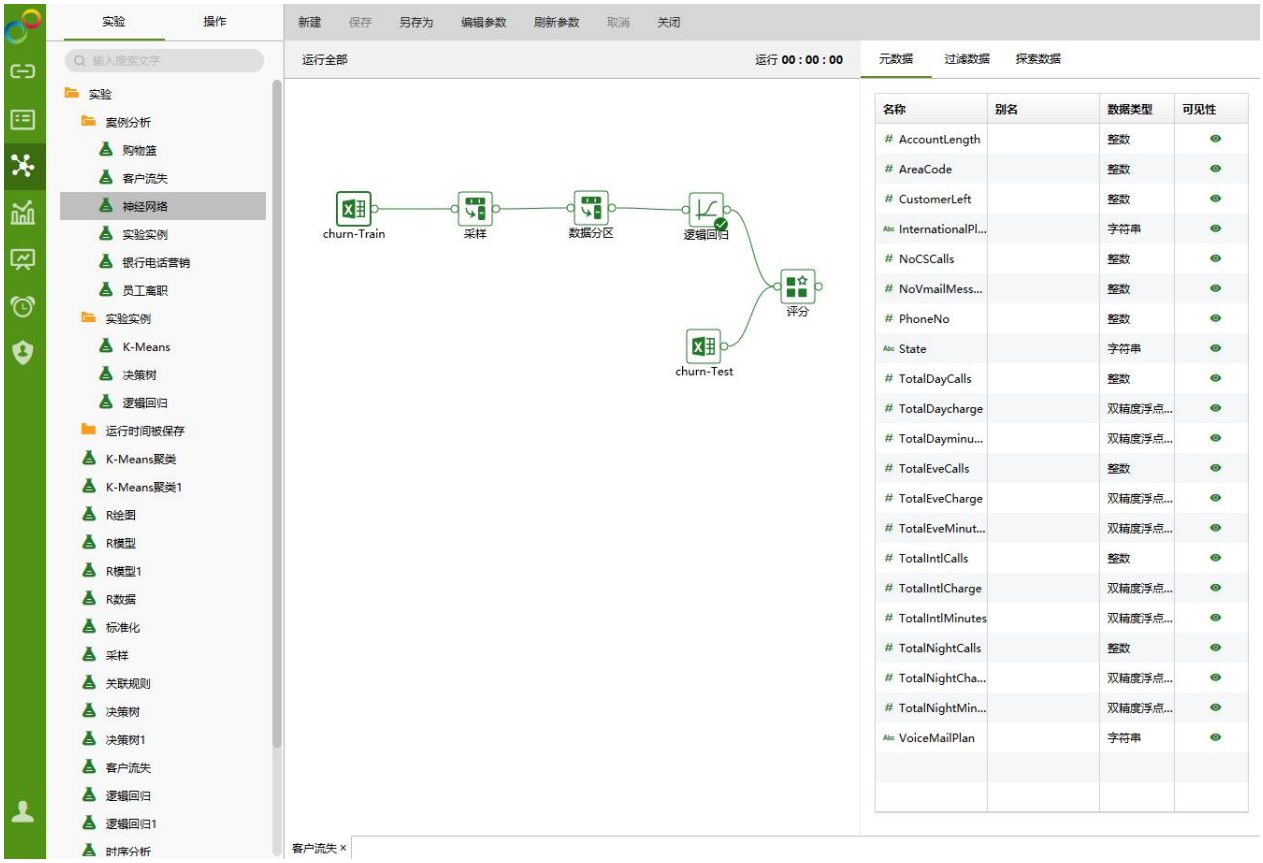
❖ 模型应用

模型应用包含应用、R 数据、R 绘图。

• 评分

将已训练的模型应用于新数据。评分节点输入节点必须有两个：一个是模型节点；一个是数据节点，该数据集节点要包含和模型中使用的列相同列名和类型的字段。

评分节点连接一个模型节点和一个数据集节点，选择评分节点，设置及展示区包含三个页面：元数据、过滤数据、探索数据。



○ 元数据

请参考数据节点里的介绍。

○ 过滤数据

请参考数据节点里的介绍。

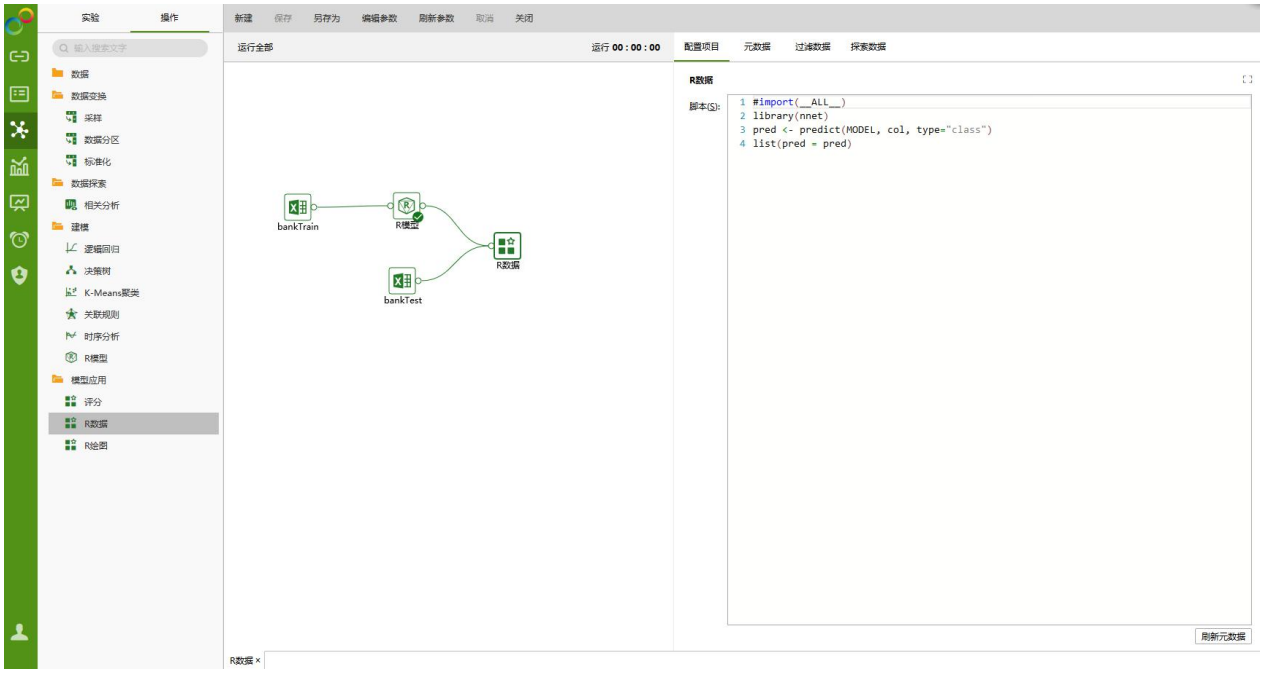
○ 探索数据

请参考数据节点里的介绍。

- **R 数据**

R 数据节点只能连接 R 模型节点，输入脚本把 R 模型的数据以表格的形式展示出来。

连接一个 R 模型节点，输入脚本后，选中 R 数据节点设置及展示区包含四个页面：配置项目、元数据、过滤数据、探索数据。



○ 配置项目

R 数据可以直接引用 MODEL 变量。也可以通过 col[[“xxx”]] 来传入数据集中对应列的值,xxx 为列的名称；也可以通过 param[[“xxx”]]来传入参数值，xxx 为参数名称。对于定制脚本，R 将最后执行的代码行的结果作为返回值返回。产品中要求返回值必须是 list 对象，包含若干返回值列，如 list(out1=a, out2=b)，其中 out1,out2 为返回值列的名称，而 a，b 为相应返回值列的取值，可以是常数或向量。

○ 元数据

请参考数据节点里的介绍。

○ 过滤数据

请参考数据节点里的介绍。

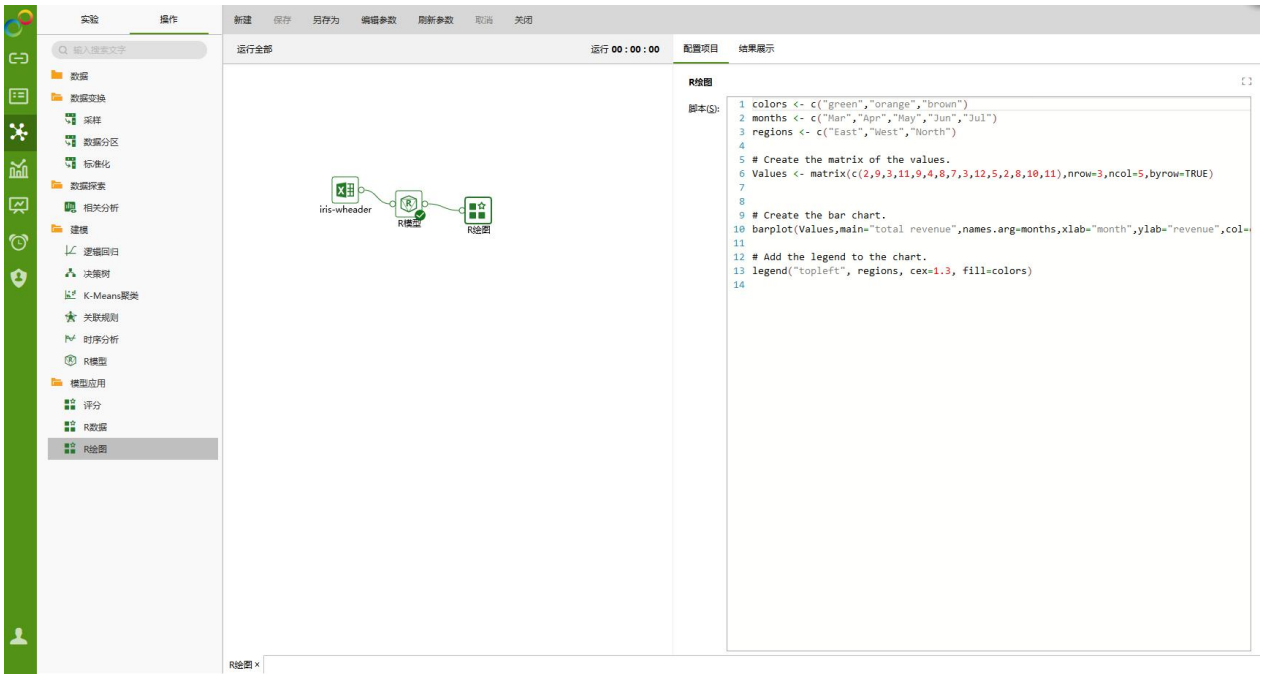
○ 探索数据

请参考数据节点里的介绍。

- **R 绘图**

R 绘图节点只能连接 R 模型节点，输入脚本把 R 模型的数据以图的形式展示出来。

连接一个 R 模型节点，输入脚本后，选中 R 绘图节点设置及展示区包含两个页面：配置项目、结果展示。

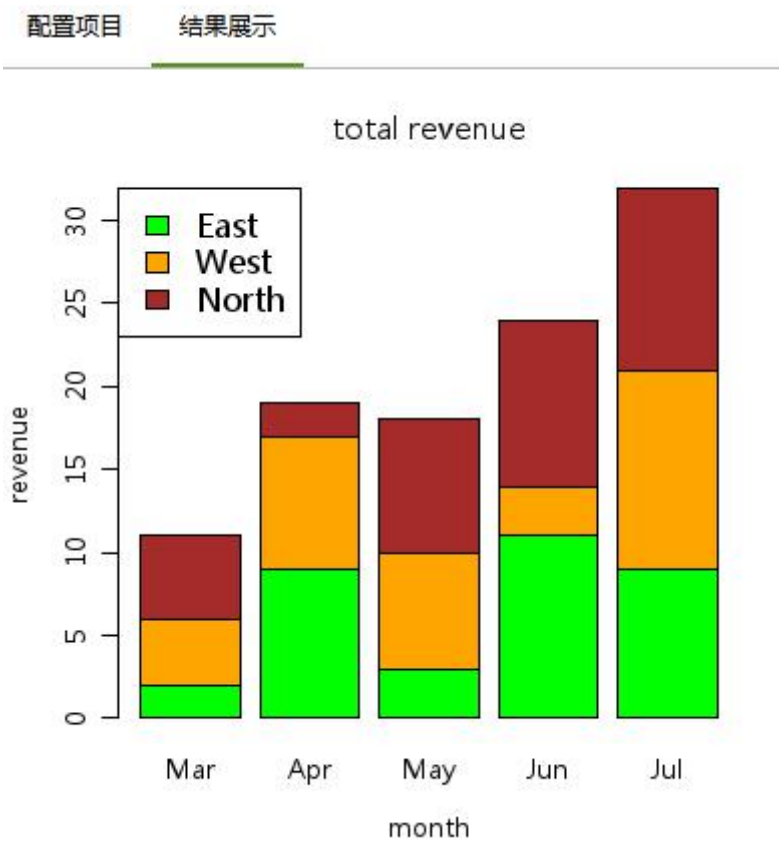


○ 配置项目

输入脚本引用 R 模型。R 绘图可以直接引用 MODEL 变量。还可引用 R 脚本来画图。

○ 结果展示

以图的形式展示 R 模型的结果。



3.2.3.2. 节点编辑

❖ **全局编辑**

【选中】单击可以选中单个节点，也可以按 Ctrl 键对节点进行多选。

【重命名】节点右键菜单点击重命名，可以对节点进行重命名。

【删除】点击节点右键菜单点击删除，或者点击键盘 delete 键进行删除，能够删除节点以及节点的输入、输出连线。

【刷新】点击节点右键刷新，可以更新同步数据或者参数信息。

❖ **数据节点**

【打开数据集】在数据节点上右键选择打开数据集，可以将数据集在创建数据集模块中打开。

【显示/隐藏所有列】将数据节点元数据中所有的列都隐藏。反之将所有的列都显示。

【复制】选中数据节点可以复制。

【粘贴】选择复制后，画布空白处右键可以粘贴，把数据节点复制一份。

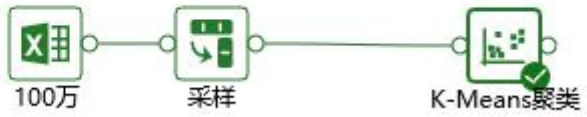
❖ 建模&模型应用节点

【运行】建模节点信息配置好后，点击右键菜单的运行来运行建模节点。

【运行全部】编辑区可以创建多个建模节点，点击运行全部，会按从上到下，从左到右的顺序来运行建模节点，当有失败时就全部停止。

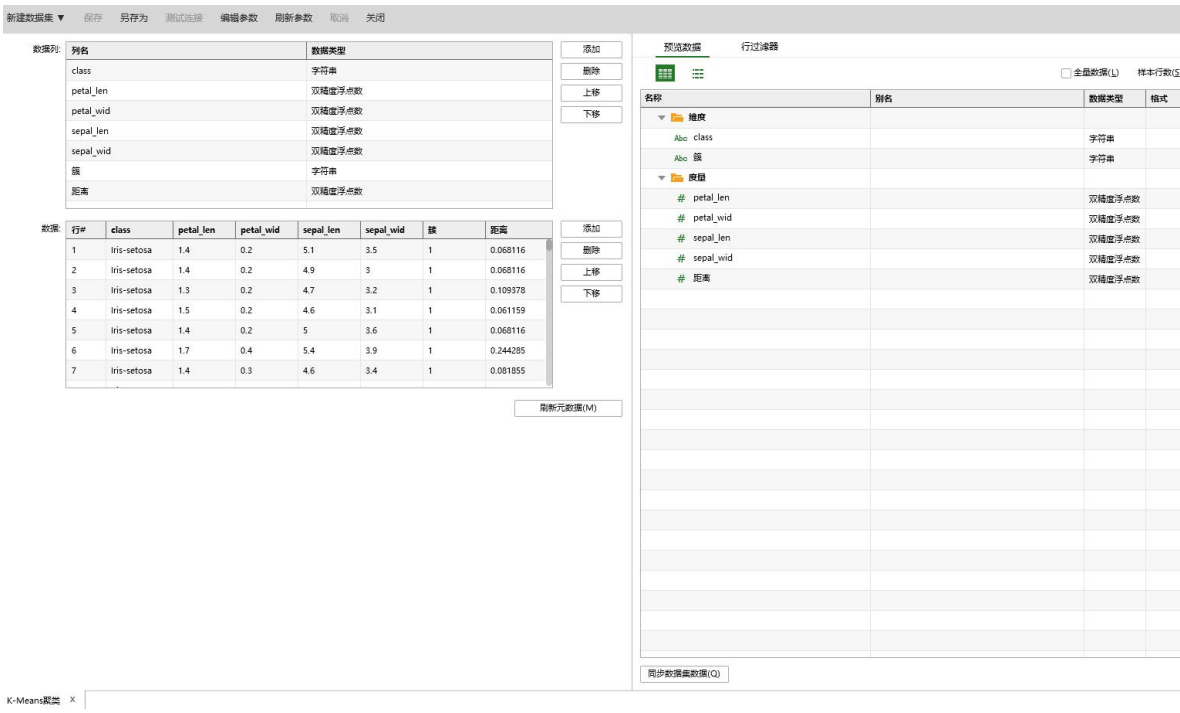
【运行状态】未运行，运行成功，运行失败。

【运行时间】单个节点运行显示的是此节点的运行时间；运行全部显示的是运行的总时间。



【保存为数据集】K-Means 聚类、时序分析、评分、R 数据四个节点支持保存为数据集。10 万以内的数据支持保存为内嵌数据集，超过 10 万不允许保存为内嵌数据集。保存的数据集可以在创建数据集模块中查看。点击此处可查看“保存为数据集”的示例。

例如，K-Means 聚类节点保存为数据集后，元数据如下：


新建数据集 ▼
保存
另存为
测试连接
编辑参数
刷新参数
取消
关闭
数据列:
列名
数据类型
class
字符串
petal_len
双精度浮点数
petal_wid
双精度浮点数
sepal_len
双精度浮点数
sepal_wid
双精度浮点数
簇
字符串
距离
双精度浮点数
添加
删除
上移
下移
数据:
行#
class
petal_len
petal_wid
sepal_len
sepal_wid
簇
距离
1 Iris-setosa 1.4 0.2 5.1 3.5 1 0.068116
2 Iris-setosa 1.4 0.2 4.9 3 1 0.068116
3 Iris-setosa 1.3 0.2 4.7 3.2 1 0.109378
4 Iris-setosa 1.5 0.2 4.6 3.1 1 0.061159
5 Iris-setosa 1.4 0.2 5 3.6 1 0.068116
6 Iris-setosa 1.7 0.4 5.4 3.9 1 0.244285
7 Iris-setosa 1.4 0.3 4.6 3.4 1 0.081855
刷新元数据(M)
预览数据
行过滤器
全量数据(L)
名称
别名
数据类型
格式
维度
class
字符串
簇
字符串
度量
petal_len
双精度浮点数
petal_wid
双精度浮点数
sepal_len
双精度浮点数
sepal_wid
双精度浮点数
距离
双精度浮点数
同步数据集数据(Q)
K-Means聚类

预览后数据如下：

预览数据　行过滤器

显示总行数(G)　预览行数: 1000

Abc class	# petal_len	# petal_wid	# sepal_len	# sepal_wid	Abc 簇	# 距离
Iris-setosa	1.4	0.2	5.1	3.5	1	0.068116
Iris-setosa	1.4	0.2	4.9	3.0	1	0.068116
Iris-setosa	1.3	0.2	4.7	3.2	1	0.109378
Iris-setosa	1.5	0.2	4.6	3.1	1	0.061159
Iris-setosa	1.4	0.2	5.0	3.6	1	0.068116
Iris-setosa	1.7	0.4	5.4	3.9	1	0.244285
Iris-setosa	1.4	0.3	4.6	3.4	1	0.081855
Iris-setosa	1.5	0.2	5.0	3.4	1	0.061159
Iris-setosa	1.4	0.2	4.4	2.9	1	0.068116
Iris-setosa	1.5	0.1	4.9	3.1	1	0.189789
Iris-setosa	1.5	0.2	5.4	3.7	1	0.061159
Iris-setosa	1.6	0.2	4.8	3.4	1	0.096256
Iris-setosa	1.4	0.1	4.8	3.0	1	0.192144
Iris-setosa	1.1	0.1	4.3	3.0	1	0.279577
Iris-setosa	1.2	0.2	5.8	4.0	1	0.160348
Iris-setosa	1.5	0.4	5.7	4.4	1	0.205429
Iris-setosa	1.3	0.4	5.4	3.9	1	0.224553
Iris-setosa	1.4	0.3	5.1	3.5	1	0.081855
Iris-setosa	1.7	0.3	5.7	3.8	1	0.152561
Iris-setosa	1.5	0.3	5.1	3.8	1	0.076163
Iris-setosa	1.7	0.2	5.4	3.4	1	0.145652
Iris-setosa	1.5	0.4	5.1	3.7	1	0.205429
Iris-setosa	1.0	0.2	4.6	3.6	1	0.269222
Iris-setosa	1.7	0.5	5.1	3.3	1	0.36113
Iris-setosa	1.9	0.2	4.8	3.4	1	0.253744
Iris-setosa	1.6	0.2	5.0	3.0	1	0.096256
Iris-setosa	1.6	0.4	5.0	3.4	1	0.218463

【导出到数据库】K-Means 聚类、时序分析、评分、R 数据四个节点支持导出数据到数据库。将节点数据导入所选数据库指定的表中。

K-Means 聚类、时序分析、评分、R 数据节点的数据可以快速回填到数据库。请点击查看“导出到数据库”具体方法。

以实验“案例分析/客户流失”为例，选中评分节点右键选择导出到数据库，如下图所示：

1.模型节点：导出到数据库的节点路径和名称。

2.数据源：用户根据需求选择已存在的数据源。目前支持的数据库类型为 "Mysql", "Oracle", "SQLServer", "DB2", "PostgreSQL", "Derby"。

3.数据库：所选数据源的默认数据库。支持的数据库为"Mysql"，"SQLServer"。

4.表结构模式：所选数据源的表结构模式。支持的数据库为 "PostgreSQL"，"SQLServer"，"DB2" ，"Oracle"，"Derby"。

5.表名：指定数据库的表名，数据集结果会插入到该表中。

6.过滤：可对数据集进行过滤条件的设置。

7.追加：当用户勾选追加时，表中原先的数据仍然存在，数据集的结果直接插入表中。当用户不勾选追加时，会先删除表中已存在的数据，然后再插入数据集结果到表中。

【保存为训练模型】K-Means 聚类、逻辑回归、决策树三个节点支持保存为训练模型。运行成功后可保存为训练模型。训练模型应用在制作报告模块的仪表盘里，可以参考深度分析实验

实验及应用章节里的 K-Means 聚类案例介绍。

1. K-Means 聚类运行成功后保存为训练模型

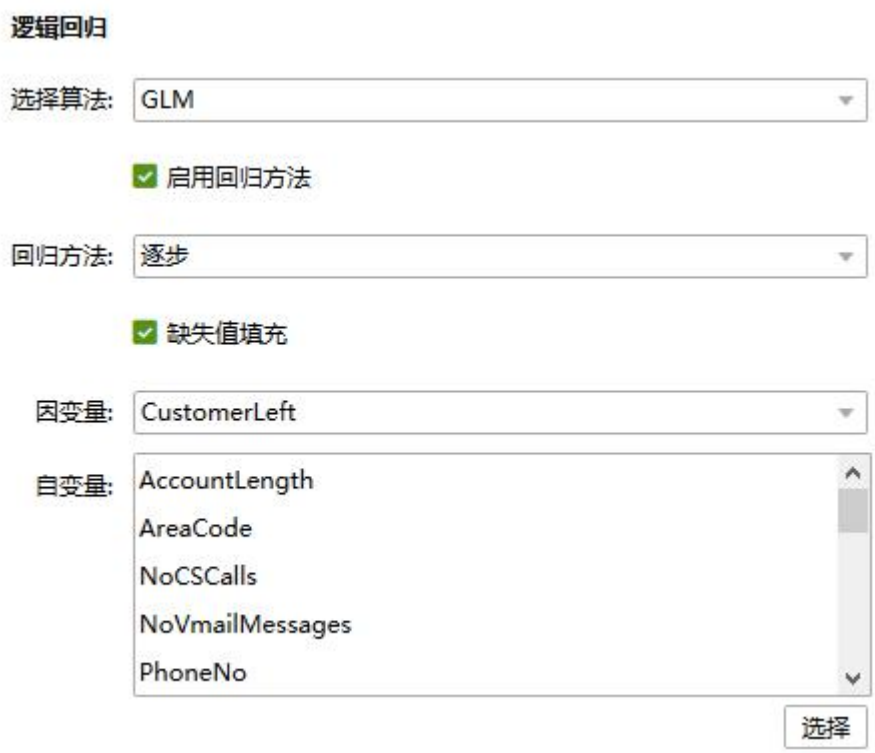
以“案例分析/员工离职”为例，选中 K-Means 聚类节点，节点的配置项目如下图：

右键选择保存为训练模型，在弹出的保存为训练模型的对话框中，可选择路径，名字默认是节点名称也可修改名字，点击确定后将模型保存到资源树上的训练模型文件夹下。

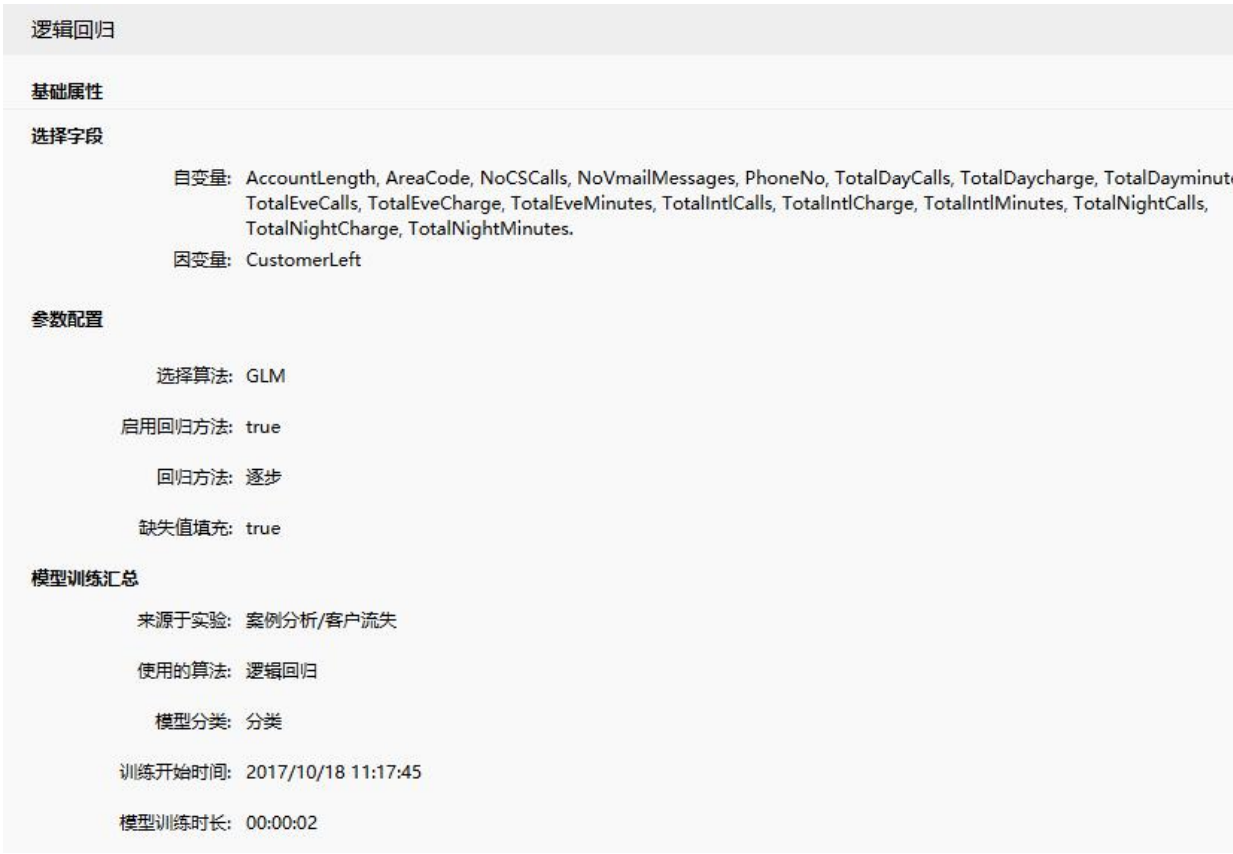
打开保存好的训练模型“K-Means 聚类”，展示信息包含三部分：标题，基础属性，模型训练汇总。标题为训练模型的名称；基础属性是配置项目里所有属性的值；模型训练汇总部分显示实验节点的来源，算法，类别，时间因素。具体展示如下：

2. 逻辑回归运行成功后保存为训练模型

以“案例分析/客户流失”为例，选中逻辑回归节点，节点的配置项目如下图：



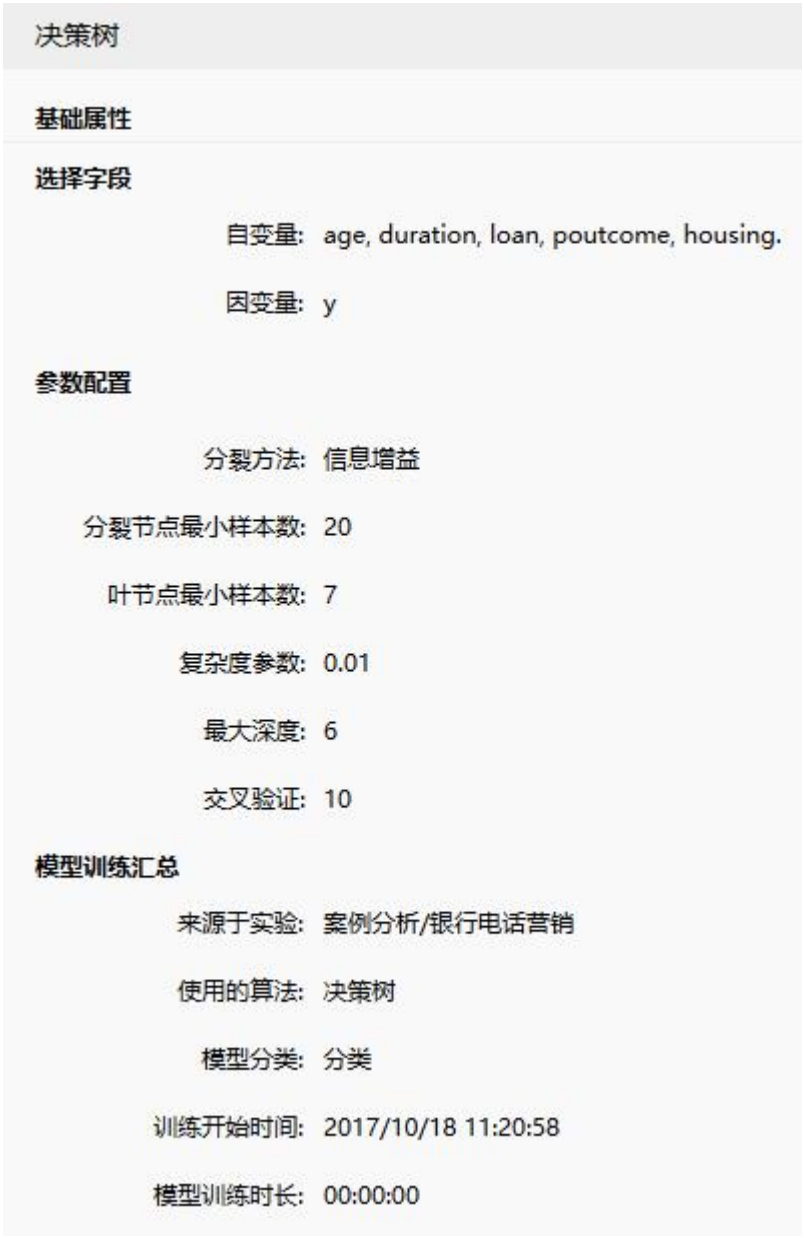
保存为训练模型，并且打开保存的训练模型，信息展示如下：



3. 决策树运行成功后保存为训练模型

以“案例分析/银行电话营销”为例，选中决策树节点，节点的配置项目如下图：

保存为训练模型，并且打开保存的训练模型，信息展示如下：



3.2.3.3. 节点连线

❖ 自动连线

以每个节点输入端或输出端所在边框的中点为圆心，在半径为 75px 半圆内区域会触发和其它节点的自动连线。

❖ 手动连线

不能自动连线的部分，都可以进行手动连线。

手工连线时，输入输出反馈，当鼠标移至输出端时，空心原点变为实心，鼠标为十字同时会出现提示框；此时点按鼠标并移至下一节点的输入端时，下一节点的输入端原点也会变为实心；如移动到输出端则该原点没有反馈。

❖ **删除连线**

可以点击连线进行删除；删除节点也会自动删除左右的连线。

4. 深度分析实验及应用

本章节通过典型案例介绍 K-Means 聚类、逻辑回归、决策树、关联规则、时序分析、R 模型六种模型的实际应用。创建流程，训练以及模型应用具体细节请参看各个章节。

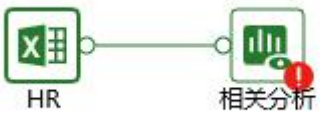
4.1.K-MEANS 聚类

❖ **员工离职率分析**

企业通过构建 K-Means 聚类分析模型，对员工进行分类，尽早识别可能离职的员工。针对性采取相关措施，降低员工离职给公司带来的损失。

• **数据准备及相关分析**

拖拽数集据节点“HR”到编辑区，添加相关分析节点去连接数据集节点。



配置相关性分析节点，相关系数选择 Pearson，在相关列中，添加入职年限，员工水平，完成项目数，平均每月工作时长，是否失误，是否离职，最近 5 年是否升职，满意度。

点击结果展示，是否离职与员工满意度的相关系数为-0.388，员工水平与完成的项目数的相关系数为 0.349，完成项目数和平均每月工作时长相关系数为 0.417，都在低相关的范围内。

	入职年限	员工水平	完成项目数	平均每月工作	是否失误	是否离职	最近5年是否升	满意度
入职年限	1	0.132	0.197	0.128	0.002	0.145	0.067	-0.101
员工水平	0.132	1	0.349	0.34	-0.007	0.007	-0.009	0.105
完成项目数	0.197	0.349	1	0.417	-0.005	0.024	-0.006	-0.143
平均每月工作	0.128	0.34	0.417	1	-0.01	0.071	-0.004	-0.02
是否失误	0.002	-0.007	-0.005	-0.01	1	-0.155	0.039	0.059
是否离职	0.145	0.007	0.024	0.071	-0.155	1	-0.062	-0.388
最近5年是否升	0.067	-0.009	-0.006	-0.004	0.039	-0.062	1	0.026
满意度	-0.101	0.105	-0.143	-0.02	0.059	-0.388	0.026	1

➢相关系数的大小说明：|r|>0.95 存在显著性相关；|r|≥0.8 高度相关；0.5≤|r|<0.8 中度相关；0.3≤|r|<0.5 低度相关；|r|<0.3 关系极弱，认为不相关。

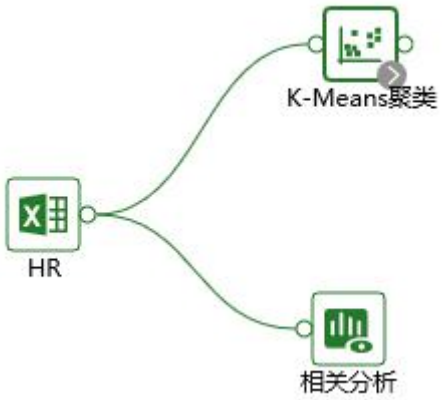
- **K-Means 聚类**

○ 配置项目

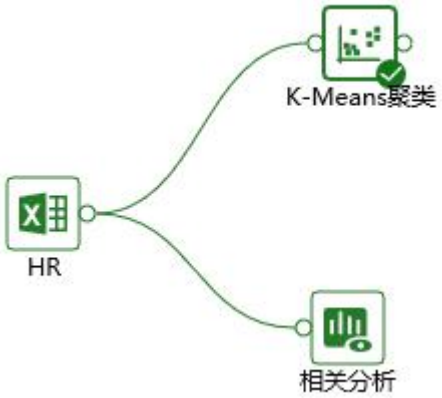
添加 K-Means 聚类节点，连接数据集节点；配置参数，选择质心数，设置为 3，初始化质心选择随机距离，随机数种子设置为 0，距离计算方法选择欧式距离。默认选择缺失值填充，不选择标准化；添加满意度和员工水平作为自变量。

○ 运行

配置完参数后，K-Means 聚类节点处于未运行状态。

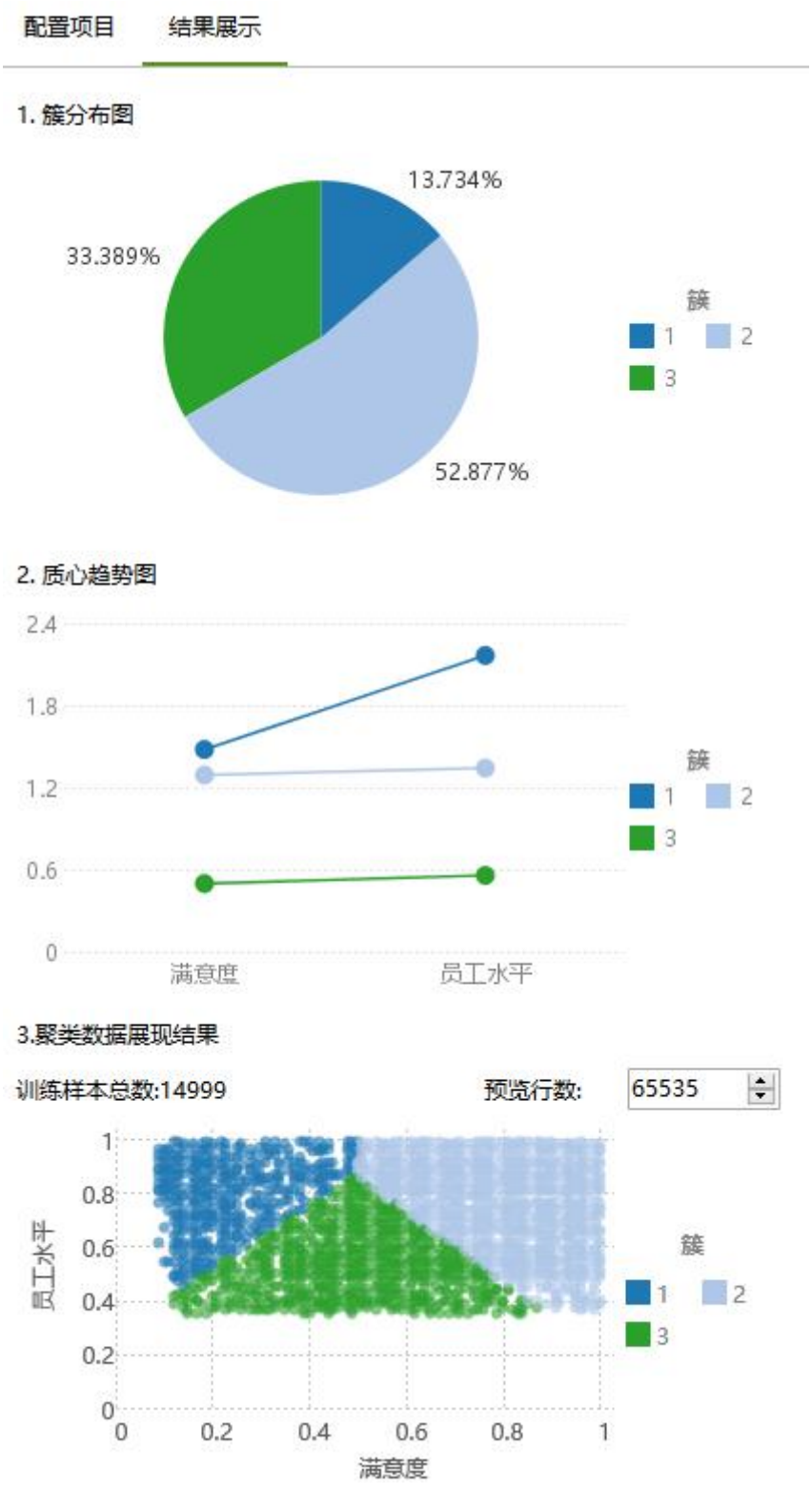


在 K-Means 聚类节点右键选择运行，运行成功后，节点展示如下：



○ 结果展示

点击结果展示，通过结果我们可以看到，按照满意度和员工水平员工明显分为 3 类，分类情况如下图：



通过 k 均值聚类质心表格可以看出 3 个质心的具体值和每个簇内的样本总数；通过簇成员表格可以看出每个簇内样本的细节数据和计算出的距离质心的距离。

4.k均值聚类质心

质心数:3

簇编号	簇内样本数	满意度	员工水平
1	2060	0.187	0.825
2	7931	0.796	0.787
3	5008	0.498	0.559

5.簇成员

训练样本总数:14999　　预览行数: 1000

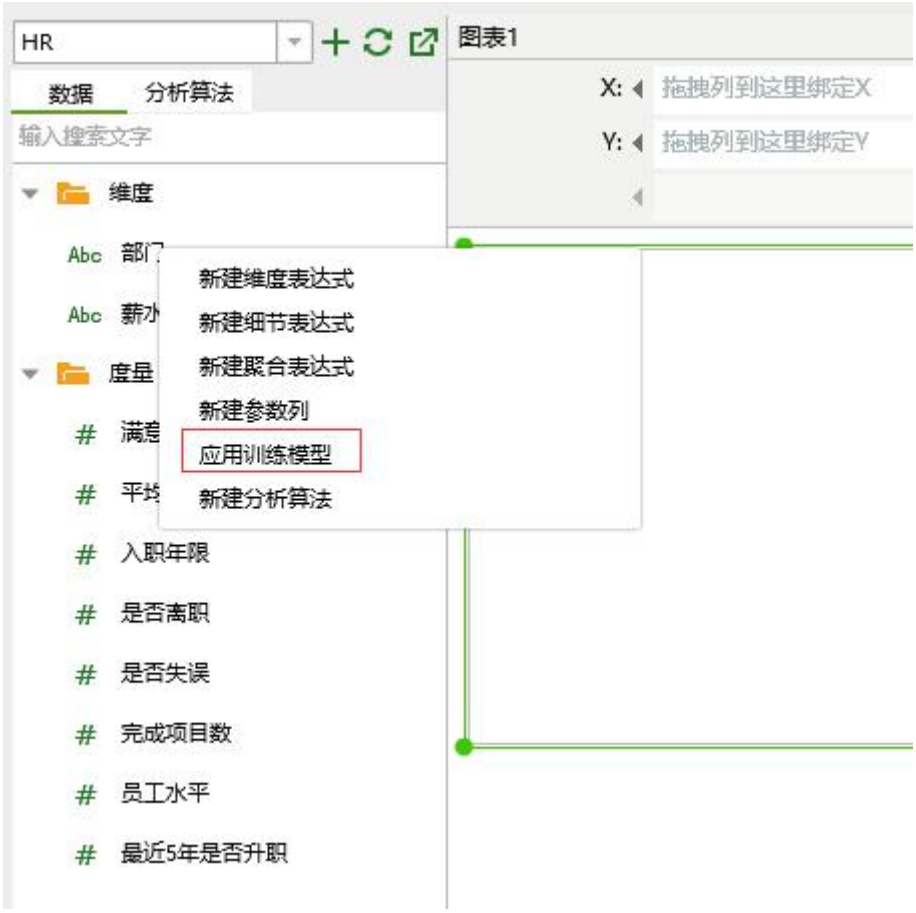
员工水平	满意度	簇	距离
0.53	0.38	3	0.122
0.86	0.8	2	0.073
0.88	0.11	1	0.094
0.87	0.72	2	0.113
0.52	0.37	3	0.134
0.5	0.41	3	0.106
0.77	0.1	1	0.103
0.85	0.92	2	0.139
1.0	0.89	2	0.233
0.53	0.42	3	0.083

- 模型应用

  - 模型在仪表盘中的应用

选择 K-Means 聚类节点，保存为训练模型，保存的模型可以应用在制作报告仪表盘的组件绑定的数据集上。

1. 在制作报告处选择数据集新建报表，右键选择应用训练模型。在数据集树上右键选择应用训练模型：

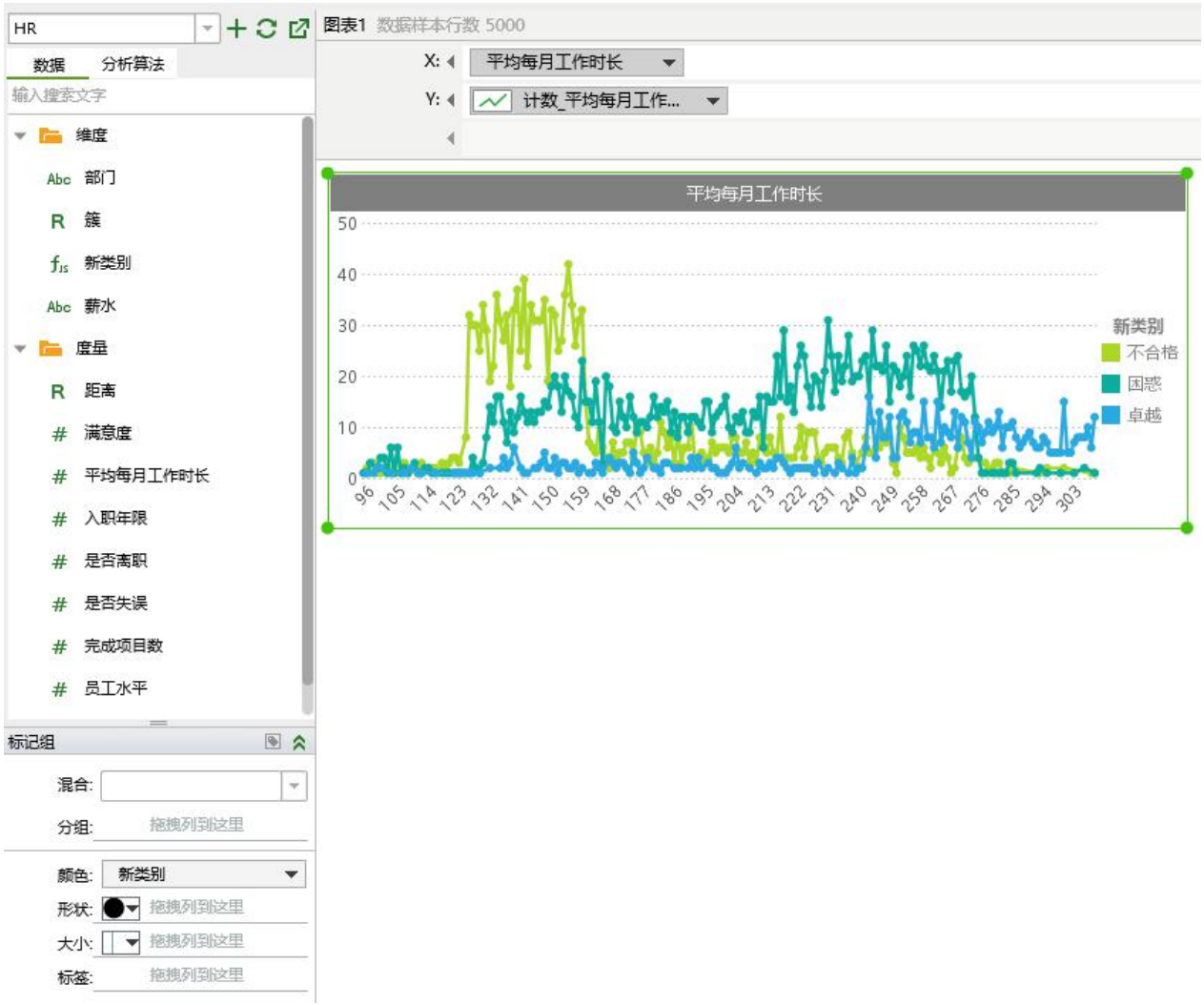


2. 打开应用训练模型对话框，此对话框内只显示可应用在此绑定的数据集上的模型。

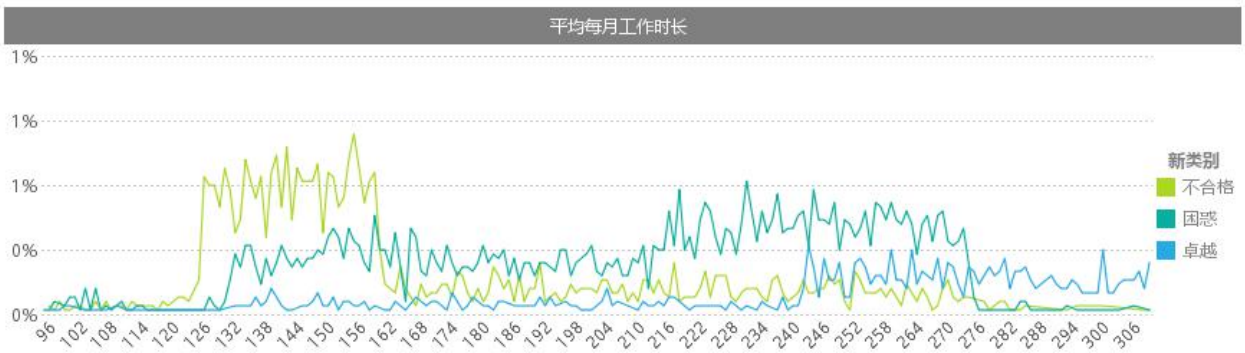
3. 选择保存的训练模型“K-Means 聚类“，点击确定，会新生成 2 列，簇和距离。簇：聚类分类的结果；距离：该点和质心的距离。

4. 基于应用模型的结果，新建维度表达式，将这三类重新命名如下：

5. 将数据如下绑定到图表组件上，选择线图。



6. 在 y 轴的字段上设置动态计算器：汇总百分比；去掉标记上的点，修改大小得出以下结果图；从上图可以看出，不合格者每月工作时长集中于左侧（偏低），困惑者工作时长最长，卓越者次之。



○ 评分

具体用法请参考深度分析实验及应用的逻辑回归章节。

4.2.逻辑回归

❖ 电信类客户流失

通信类服务商可以通过逻辑回归模型，尽早地发现可能流失的客户，并针对这些客户的行为进行干预，通过定制更多的服务来挽留客户。

- **数据准备**

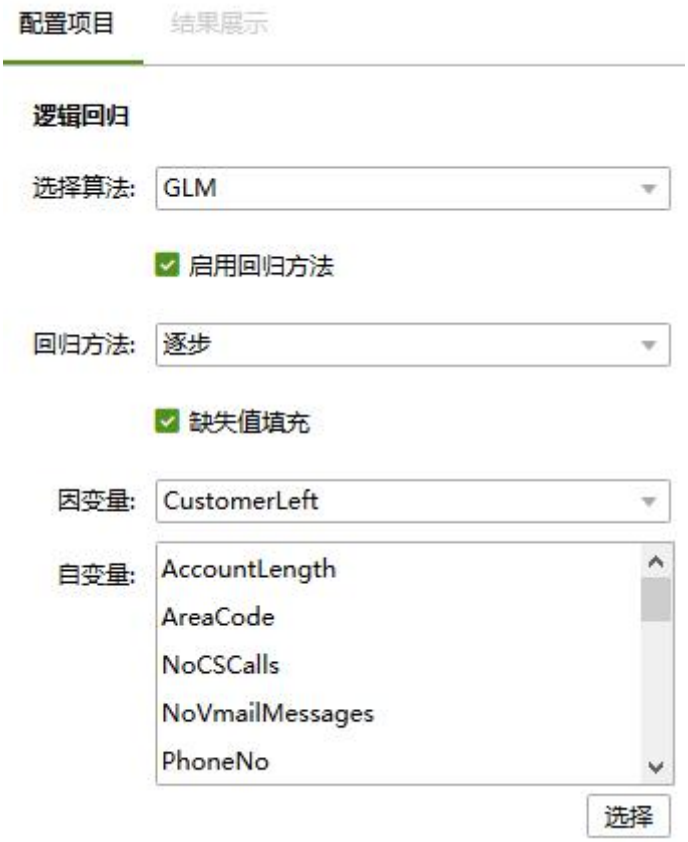
拖拽数据集节点“churn-Train”到编辑区，添加采样和数据分区节点，采样节点连接数据集，数据分区连接采样。

采样和数据分区节点配置项目如下：

- 逻辑回归

○ 配置项目

添加逻辑回归节点，连接数据分区；配置参数，算法为 GLM，启用回归方法，选择逐步，缺失值默认填充，因变量为 CustomerLeft，其它列做为自变量。



○ 运行

配置完参数后，逻辑回归节点处于未运行状态。

在逻辑回归节点右键选择运行，运行成功后，节点展示如下：

○ 结果展示

通过模型系数可以得到逻辑回归方程的系数，包括截距项，各自变量的系数以及它们的 P 值，标准误差。还可以看到模型训练集合验证集的准确率和均方误差。P 值都较小，P 值越小结果越好。

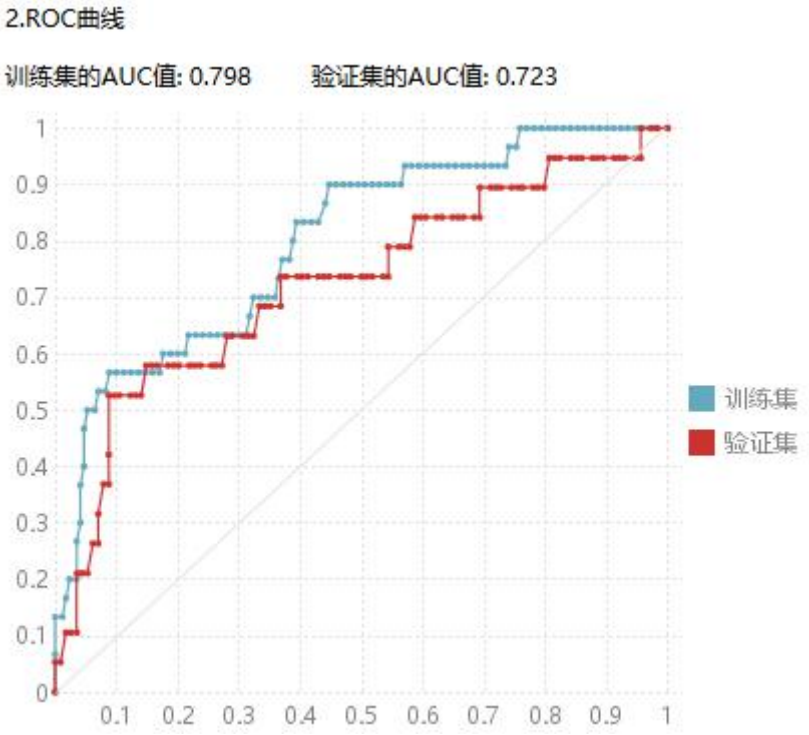
1.模型系数

训练集： 准确率: 0.855　　均方误差: 20.216

验证集： 准确率: 0.850　　均方误差: 14.514

	截距项	NoCSCalls	NoVmailMessa
回归系数	-8.203	0.553	-0.063
标准误差	1.961	0.181	0.027
P值	2.884e-5	0.002	0.020

通过 ROC 曲线看出训练集 AUC 值 0.798；验证集 AUC 值 0.723。AUC 值越大模型分类效果越好。



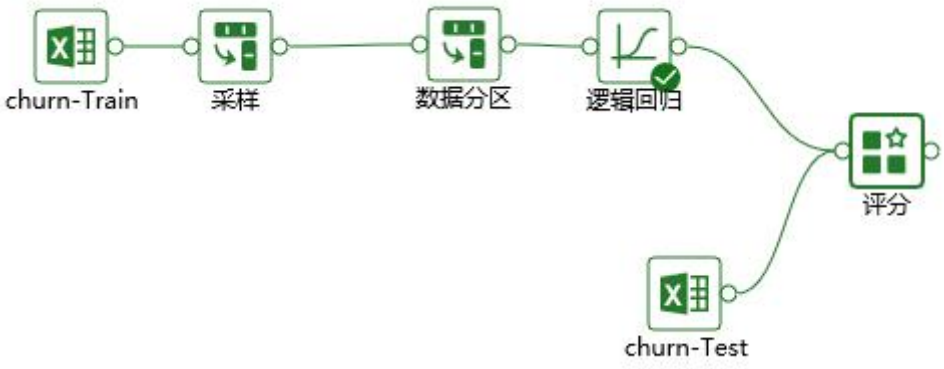
- 模型应用

○ 保存为训练模型

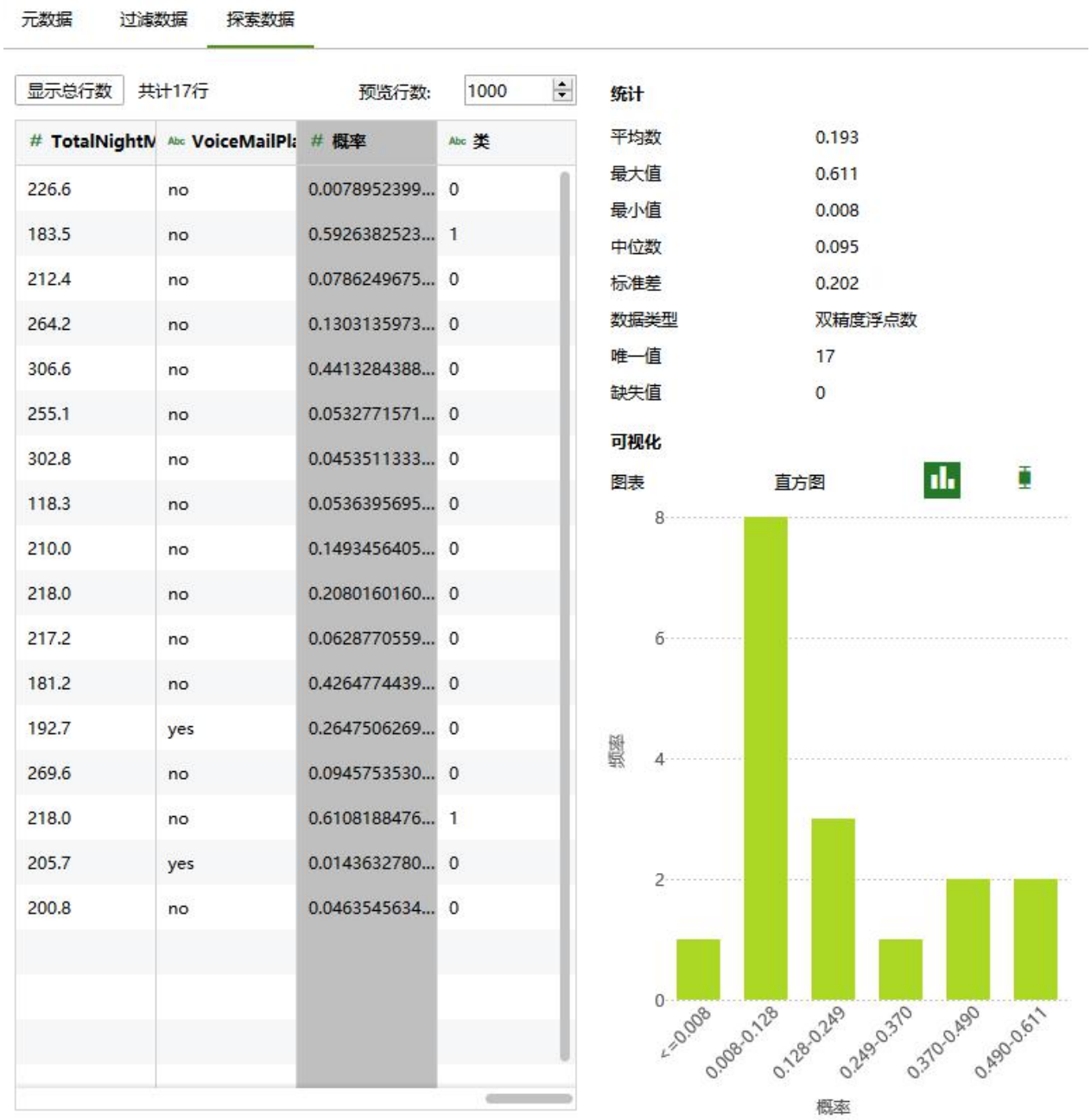
选择逻辑回归节点，保存为训练模型，保存的模型可以应用在制作报告仪表盘的组件绑定的数据集上。根据实际的分析需求来使用，类似用法请参 K-Means 聚类的介绍。

○ 评分

添加数据集节点“churn-Test”和评分节点，评分节点连接逻辑回归和 churn-Test 两个节点。



选择评分节点，在数据探索里可查看已训练模型逻辑回归应用于数据集“churn-Test“的结果，展示结果如下：



4.3.决策树

❖ 银行电话营销

银行应用决策树分类算法对具有不同金融产品消费特征的客户进行分类，从而针对不同类别的

客户采取对应的营销策略，增加电话营销成功的概率。

- 数据准备

拖拽数据集节点“bankTrain”到编辑区，添加采样和数据分区节点，采样节点连接数据集，数据分区连接采样。

采样和数据分区节点配置项目如下：

- 决策树

○ 配置项目

添加决策树节点，连接数据分区；配置参数如下图：

○ 运行

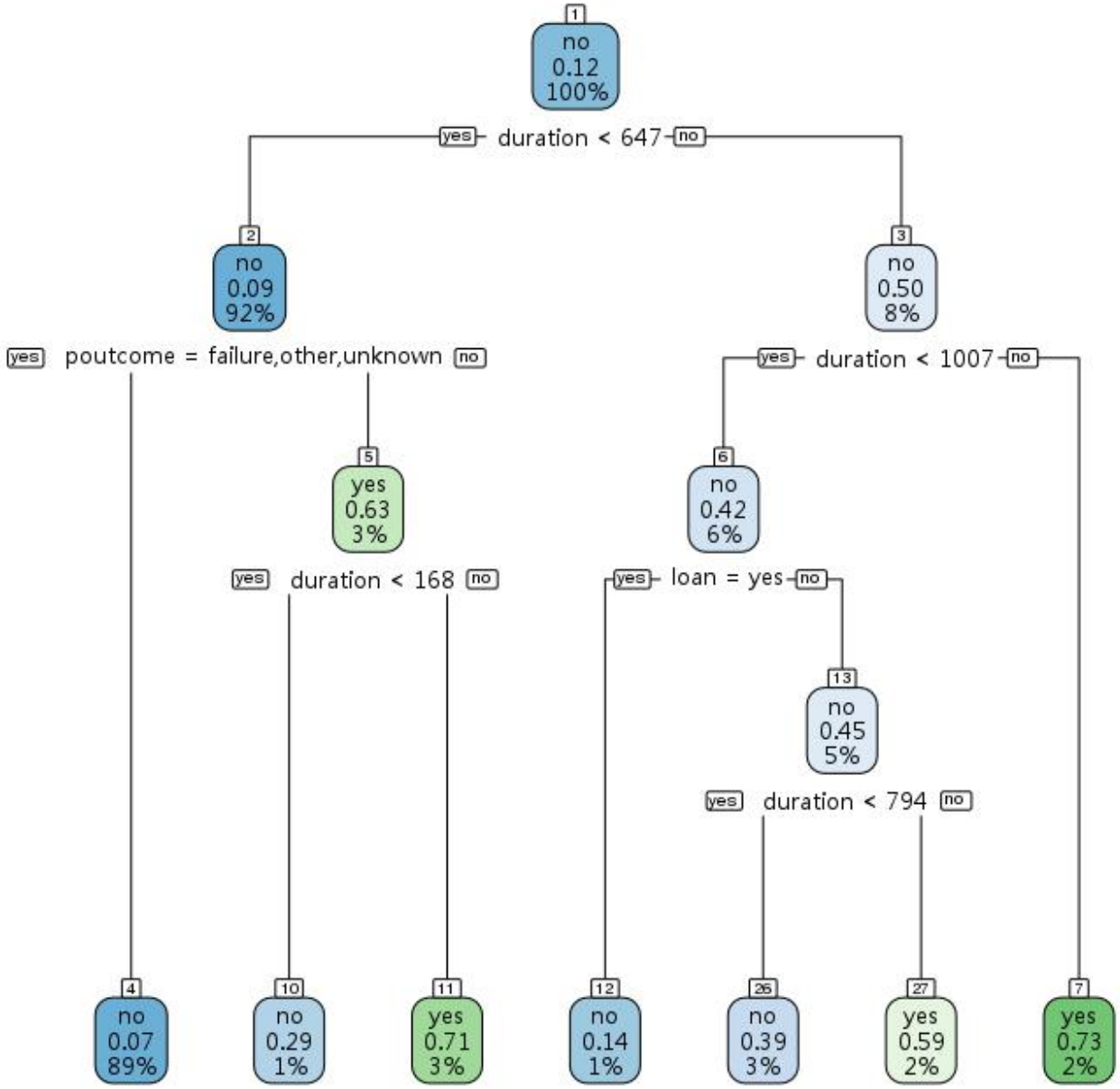
点击运行全部按钮，运行成功后，节点展示如下：

○ 结果展示

1. 叶子节点分析：以节点 11 为例，绿色底色的规则为「当 duration>=167.500 and duration<647，poutcome 不在[“failure” “other”，“unknown”]，也就其是 success 时

其预测结果是 yes」，节点样本数为 28，其中 71% 被预测为 yes，占总数的 3%。

树形结果：



节点分类情况：

2.节点分类情况

节点编号	类别	判别条件	节点样本数
11	yes	duration < 647 poutcome in [success] duration >= 167.500	28
12	no	duration >= 647 duration < 1007 [illegible]	7

2. 混淆矩阵分析

通过混淆矩阵可以看出训练集预测的精确度为 90.4%，验证集的精确度为 89.4%。

3.混淆矩阵

训练集的精确度: 0.904

真实值 / 预测值	no	yes	
no	930(86.11%)	83(7.69%)	
yes	21(1.94%)	46(4.26%)	

验证集的精确度: 0.894

真实值 / 预测值	no	yes	
no	624(86.67%)	61(8.47%)	
yes	15(2.08%)	20(2.78%)	

3. 总体分解结果

通过决策树的结果明显可以看出通话时长越长，这次营销成功的概率越高，通话时长小于 647s 的营销失败的案例占比为 89%。

- 模型应用

○ 保存为训练模型

选择决策树节点，保存为训练模型，保存的模型可以应用在制作报告仪表盘的组件绑定的数据集上。根据实际的分析需求来使用，类似用法请参 K-Means 聚类的介绍。

○ 评分

具体用法请参考深度分析实验的逻辑回归章节。

4.4.关联规则

❖ 购物篮分析

关联规则是一个典型的购物篮分析模型。分析人员可以通过关联规则模型发现交易数据中不同的商品项的关联，从而找出客户的购买行为模式。

- **数据准备**

进行关联分析时，我们抽取了某商场的客户购买数据，并从中筛选出和消费行为相关属性的数据，主要是相关的商品列，设置成 bool 类型的变量。

拖拽数据集节点“购物篮”到编辑区，添加关联规则节点，关联规则连接数据集。

- **关联规则**

○ 配置项目

选择关联规则节点配置参数如下图：


配置项目
结果展示
关联规则
支持度范围(%):
10
-
100
置信度(%):
80
最小项数:
2
最大项数:
8
自变量:
frozenmeal
fruitveg
pmethod
softdrink
wine
选择

选择的自变量如下：


选择列
数据列:
homeown
sex
选择的列:
beer
cannedmeat
cannedveg
confectionery
dairy
fish
freshmeat
frozenmeal
fruitveg
pmethod
softdrink
wine
确定(O)
取消(C)

○ 运行

点击运行全部按钮，运行成功后，节点展示如下：

○ 结果展示

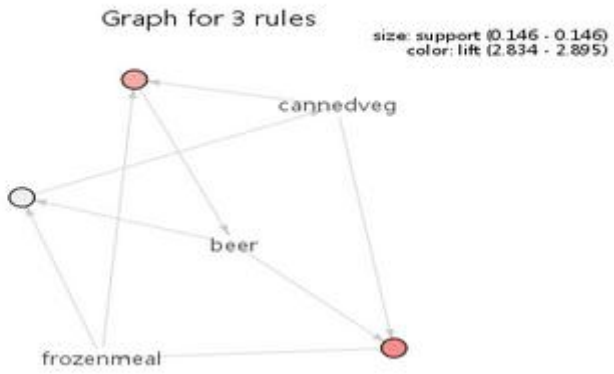
以表中第一条数据为例，bear（啤酒）和 frozenmeal（冻肉）是左项，cannedveg（罐装蔬菜）是右项。支持度（Support）表示购买左项的记录数占所有的购买记录数的百分比，置信度（Confidence）表示同时购买左项和右项的记录数占购买右项记录数的百分比，提升度（Lift）表示置信度与已知右项的百分比的比值，提升度大于 1 的规则才是有意义的。

根据结果可以看出购买了啤酒，冻肉，的顾客有 85.9%同时购买了罐头蔬菜，而且提升度为 2.834，表明此规则的相关性很强。由此可知，啤酒和冻肉、罐装蔬菜放在一起销售比较合理。

关联规则图显示了前三项关联性较强的项目。

关联规则图：

1.关联规则图



关联规则信息表：

2.关联规则

左项	右项	支持度	置信度	提升度
[beer, frozenmeal]	[cannedveg]	0.146	0.859	2.834
[beer, cannedveg]	[frozenmeal]	0.146	0.874	2.895
[cannedveg, frozenmeal]	[beer]	0.146	0.844	2.880
[beer, frozenmeal]	[confectioner...	0.136	0.800	1.105
[beer, cannedveg]	[confectioner...	0.137	0.820	1.133
[cannedveg, frozenmeal]	[confectioner...	0.143	0.827	1.142
[beer, cannedveg, frozenmeal]	[confectioner...	0.120	0.822	1.135

4.5. 时序分析

❖ 财务预测

时序分析就是根据时间序列所反映出来的发展过程、方向和趋势，进行类推，预测下一个阶段的发展规律。企业可以通过时序分析模型来及时有效的监控企业的财务风险。

- **数据探索**

拖拽数据集节点“财务数据”到编辑区，在探索数据页面选择实际数值，通过直方图可以分析2010-2016 每月的财务收入分布情况。

元数据 过滤数据 探索数据

显示总行数 预览行数: 1000

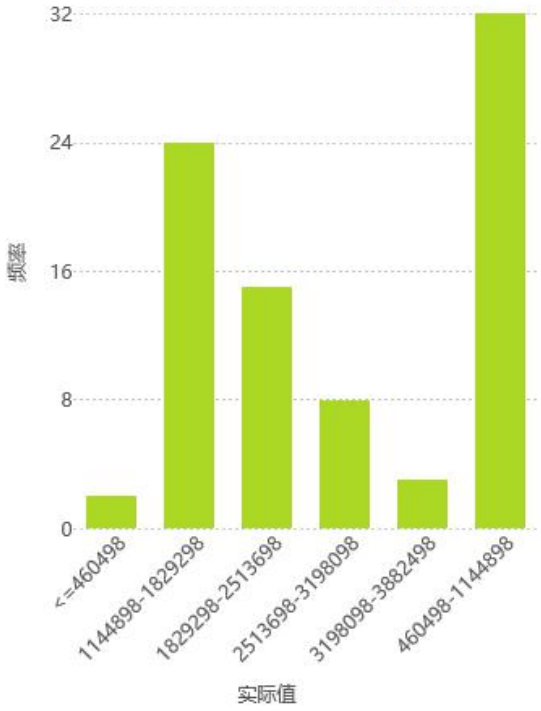
# 实际值	# 月份_R
460498	2010-01-01
460498	2010-02-01
698942	2010-03-01
635317	2010-04-01
662832	2010-05-01
899792	2010-06-01
553362	2010-07-01
603177	2010-08-01
880225	2010-09-01
815538	2010-10-01
867434	2010-11-01
1380870	2010-12-01
670387	2011-01-01
670386	2011-02-01
955024	2011-03-01
752910	2011-04-01
851678	2011-05-01
1106011	2011-06-01
781866	2011-07-01
765519	2011-08-01
1022129	2011-09-01

统计

平均数	1516177.964
最大值	3882496
最小值	460498
中位数	1293701.000
标准差	784048.483
数据类型	整数
唯一值	80
缺失值	0

可视化

图表 直方图



- 时序分析

拖拽时序分析节点，连接数据集节点。

- 配置项目

选择时序分析节点配置参数如下图：

○ 运行

点击运行全部按钮。

- 结果分析

○ 拟合程度分析

图中绿色部分为实际值，灰色线为拟合值。从下图可以看出拟合曲线和实际曲线重合度很高，说明模型预测的比较成功，其中峰值预测也比较准确。

○ 分布情况说明

从图中可以看到整体趋势在逐年递增，每季度末都会有指标冲高的情况，以年底冲指标尤为明显。

但是 2016 年 12 月发生了比较大的波动，实际收入明显低于同期水平，可能是受到政策或宏观经济环境的影响。

○ 模型统计量说明

2.模型统计量

MSE	LBQ检验		
	统计量	显著性	自由度
26718743914.824	20.173	0.323	18

【MSE】平方误差的均值。越小越好。

【LBQ 检验】时间序列是否存在滞后相关的一种统计检验。显著性小于 0.05 时说明误差存在明显的自相关性，表示模型拟合不良。显著性值越接近 1 模型拟合越好。

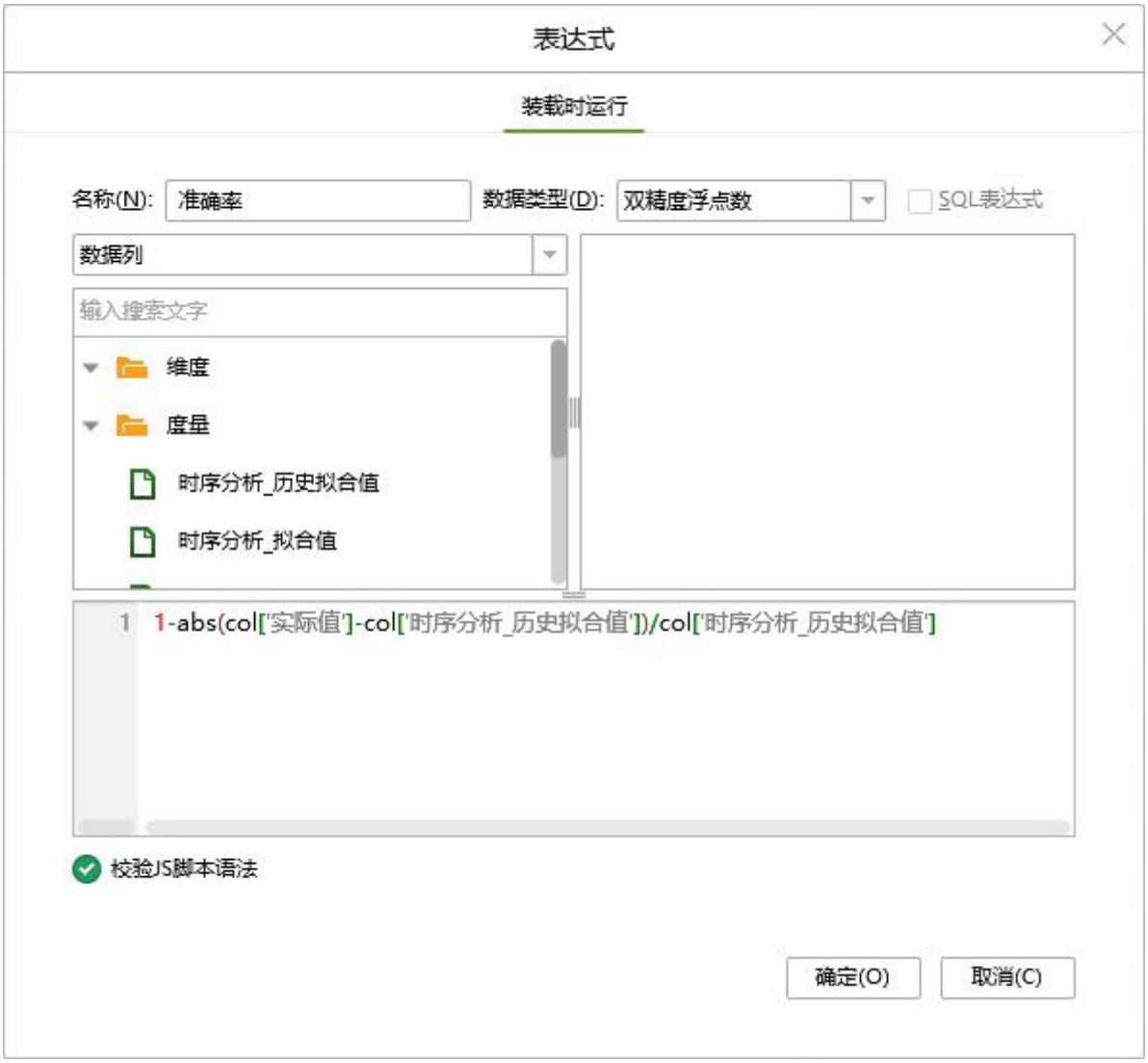
○ 预测值分析

- 分析数据拟合准确率

1. 选择时序分析节点，右键选择保存为数据集，保存成内嵌数据集。

2. 在创建数据集模块，打开保存的内嵌数据集，在元数据区域右键新建表达式“准确率”。

输入脚本：1-abs(col['实际值']-col['时序分析_历史拟合值'])/col['时序分析_历史拟合值']



3. 对比 2015，2016 财务数字的实际值和历史拟合值，发现预测的准确率很好，平均准确率为 91.34%, 2016 年的预测的平均准确率为 93%。

# 时序分析_历史拟合值	# 时序分析_拟合值	# 实际值	# 月份_R	f_{1S} 准确率
1397499.756696		1301241	2015-01-01	0.9311207345584251
1401540.147892		1301240	2015-02-01	0.9284357654378597
1848096.356826		2011350	2015-03-01	0.9116638899422004
1834731.356106		1453572	2015-04-01	0.7922533155399025
1779225.461584		1828353	2015-05-01	0.9723882445048515
2201788.35309		2522724	2015-06-01	0.8542386481154751
1871702.447384		2054402	2015-07-01	0.9023885698972337
1892148.496982		2171734	2015-08-01	0.8522391326769848
2365872.306814		2364585	2015-09-01	0.9994558849138677
2329632.19349		2632152	2015-10-01	0.8701426743005307
2729871.927226		2775542	2015-11-01	0.9832702507694534
3745463.02985		3882496	2015-12-01	0.9634136102644997
1813979.523923		1625462	2016-01-01	0.8960751643352054
1800833.86809		1625462	2016-02-01	0.9026162983729293
2444479.754507		2507823	2016-03-01	0.9740872284271486
1984731.545026		2232231	2016-04-01	0.8752982711469133
2409872.207187		2258815	2016-05-01	0.9373173371033949
3011867.277197		3029292	2016-06-01	0.9942146445379837
2513811.183893		2230977	2016-07-01	0.8874878965829923
2514507.750705		2368854	2016-08-01	0.9420746463540776
2676025.479418		2896538	2016-09-01	0.9175970026152522
2926907.448055		2747611	2016-10-01	0.9387420165355256
3020210.984524		3036174	2016-11-01	0.9947146025367775
4098790.72728		3683372	2016-12-01	0.8986484661158399

- 模型预测值

2017 年的预测值如下图：

分析_下界	# 时序分析_历史拟合值	# 时序分析_拟合值	# 实际值	# 月份_R	f_{1S} 准确率
	2926907.448055		2747611	2016-10-01	0.93874201
	3020210.984524		3036174	2016-11-01	0.99471460
	4098790.72728		3683372	2016-12-01	0.89864846
2.660334		1787335.133776		2017-01-01	
3.933592		1819027.782172		2017-02-01	
2.548898		2683917.549715		2017-03-01	
4.02466		2351587.02852		2017-04-01	
5.922531		2403821.673154		2017-05-01	
2.110755		3165434.14698		2017-06-01	
7.411628		2421464.041287		2017-07-01	
.094592		2586661.429364		2017-08-01	
5.973875		3065421.015196		2017-09-01	
2.167912		2950404.934913		2017-10-01	
3.894723		3232906.58607		2017-11-01	
.172141		3964917.354592		2017-12-01	

4.6. R 模型

❖ 神经网络分析

用户可以使用 R 模型节点引用任意 R package 来连接 R 数据或者 R 绘图进行分析，本示例就是通过引用神经网络的 R package 去分析银行的电话营销效果。

1. 拖拽数据集节点“bankTrain”到编辑区，添加 R 模型节点，R 模型连接数据集。

2. 在 R 模型中输入脚本

```
#import(__ALL__)
                                    //引用数据集中所有的列
library(nnet)
                                    //引用的 package：nnet 是神经网络
```

MODEL<-nnet(y ~ ., data=col,size=10,rang=0.1,decay=5e-4,maxit=200) //建模。

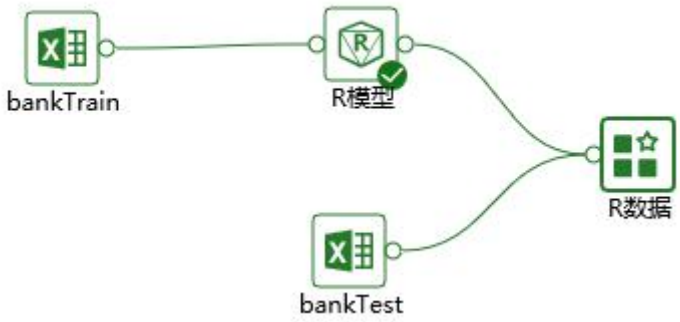
Size： 隐藏层的层数；Rang：参数的范围；Decay：衰减的参数；Maxit 迭代次数。

3. 右键运行 R 模型，运行成功。

4. 拖拽数据集节点“bankTest”到编辑区。

5. 拖拽 R 数据节点到编辑区。

6. R 数据连接“bankTest”和 R 模型。



7. 在 R 数据节点输入脚本

#import(__ALL__)

//引用数据集中所有的列

library(nnet)

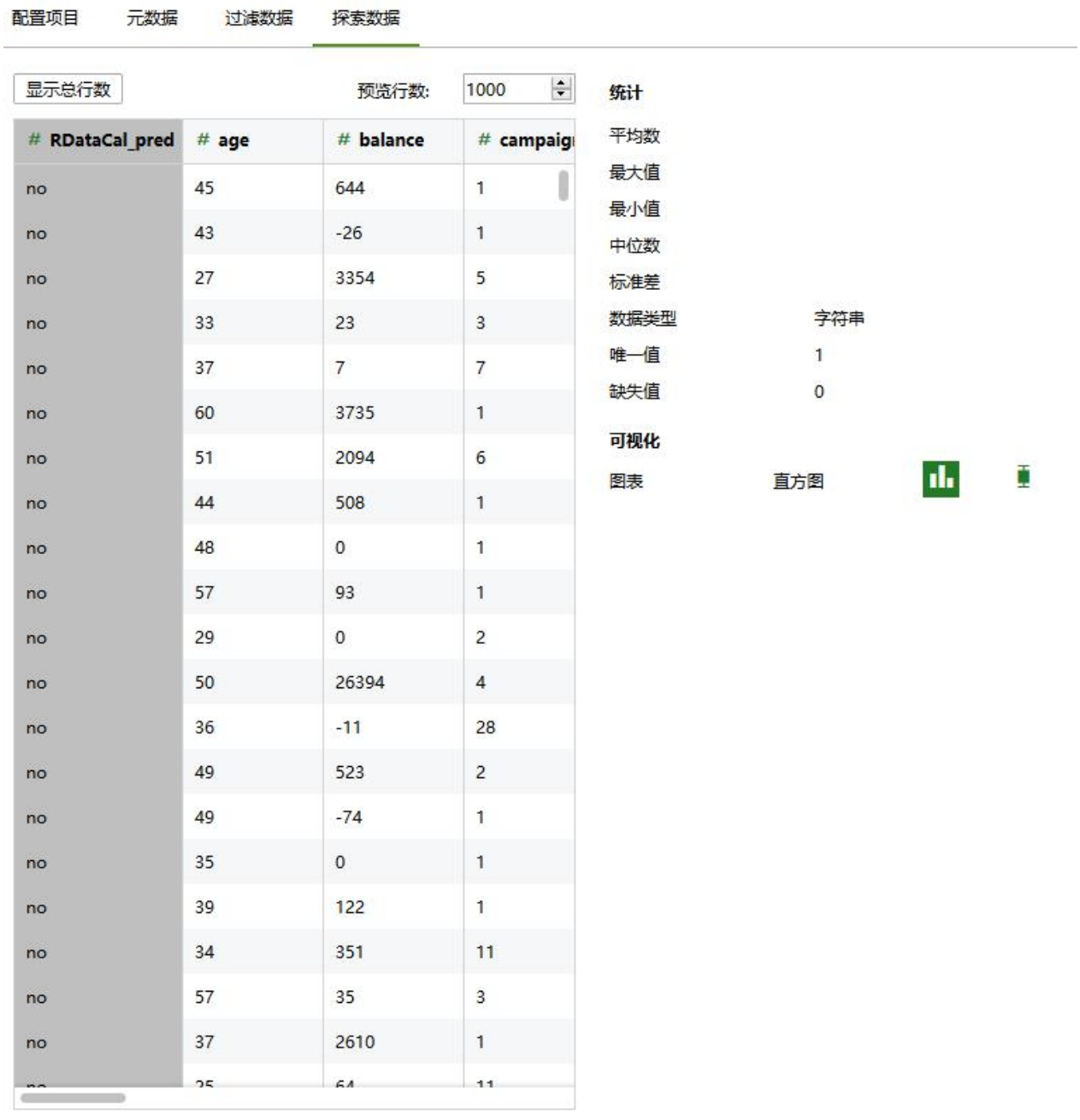
//引用的 package：nnet 是神经网络

pred <- predict(MODEL, col, type="class")

//引用 MODEL 变量，col 是包含数据集所有列的数据框

```
list(pred = pred)
                    //返回 list 结果
```

8. 选择 R 数据的探索数据，增加预测列 RDataCal_pred。



# RDataCal_pred	# age	# balance	# campaig
no	45	644	1
no	43	-26	1
no	27	3354	5
no	33	23	3
no	37	7	7
no	60	3735	1
no	51	2094	6
no	44	508	1
no	48	0	1
no	57	93	1
no	29	0	2
no	50	26394	4
no	36	-11	28
no	49	523	2
no	49	-74	1
no	35	0	1
no	39	122	1
no	34	351	11
no	57	35	3
no	37	2610	1

5. 其它分析算法

产品中的传统快速分析算法，包含以下三种创建方式：

1. 在数据集上新建分析算法。在数据集的元数据界面上，右键选择“ 新建分析算法”，在弹出的“ 分析算法” 窗口中输入名称，选择分析算法类型，选择并设置算法需要的数据列或属性值，或自定义 R 脚本，返回新的 R 字段（输出值）。根据分析算法生成 R 字段，作用域是当前数据集，所有使用该数据集的报表都能使用此数据集上的 R 字段。

2. 在报表组件绑定的数据集上新建分析算法。新生成的 R 字段作用域是当前报表，当前报表上的所有组件都可以使用此数据集上的 R 字段。

3. 图表组件上可以用四种分析算法进行快速绘图。支持一元线性回归、LDA 线性分类、K-means 聚类、HoltWinters 时序分析。把分析算法拖入图表区域，在弹出的“ 分析算法”窗口中选择并设置算法需要的数据列或属性值，点击“ 确定” 之后，R 字段会自动绑定到图表上，绘制出用户期望的图形。

5.1. 一元线性回归

❖ 一元线性回归

回归分析是一种应用非常广泛的统计工具，主要用来建立两个变量之间的关系模型。其中一个变量被称为自变量，其值是通过实验收集的。另一变量称为因变量，其值是根据自变量计算而得。 线性回归这两个变量满足一个等式，其中这两个变量是指数(幂) 相关。 在数学上，一元一次线性关系的图形表示为直线。一元 N 次线性关系的图形表示为曲线。 一元一次线性回归的一般数学方程为 y = ax+b，其中 y 是因变量，x 是自变量，a 和 b 称为系数常数。

【自变量】x，从下拉列表中选出需要作为自变量的字段。

【因变量】y，从下拉列表中选出需要作为因变量的字段。

【多项式次方】表示自变量与因变量是一个几次的线性方程关系，默认为 1，如果是 2 则表示一元二次方程。

【输出值】【拟合值】：被勾选时，会得到一个拟合值字段。其结果是得到相应的线性回归模型后，对给定的样本值(x1,x2,...,xn) 做预测，也就是(y1,y2,...,yn) 的估计值。

【输出值】【残差】：被勾选上时，会得到一个残差字段。其结果是 y 的实际值减去拟合值。

【输出值】【置信区间】：被勾选上时，根据 Level 算出估计值得到上界和下界，默认 Level 是 95%。

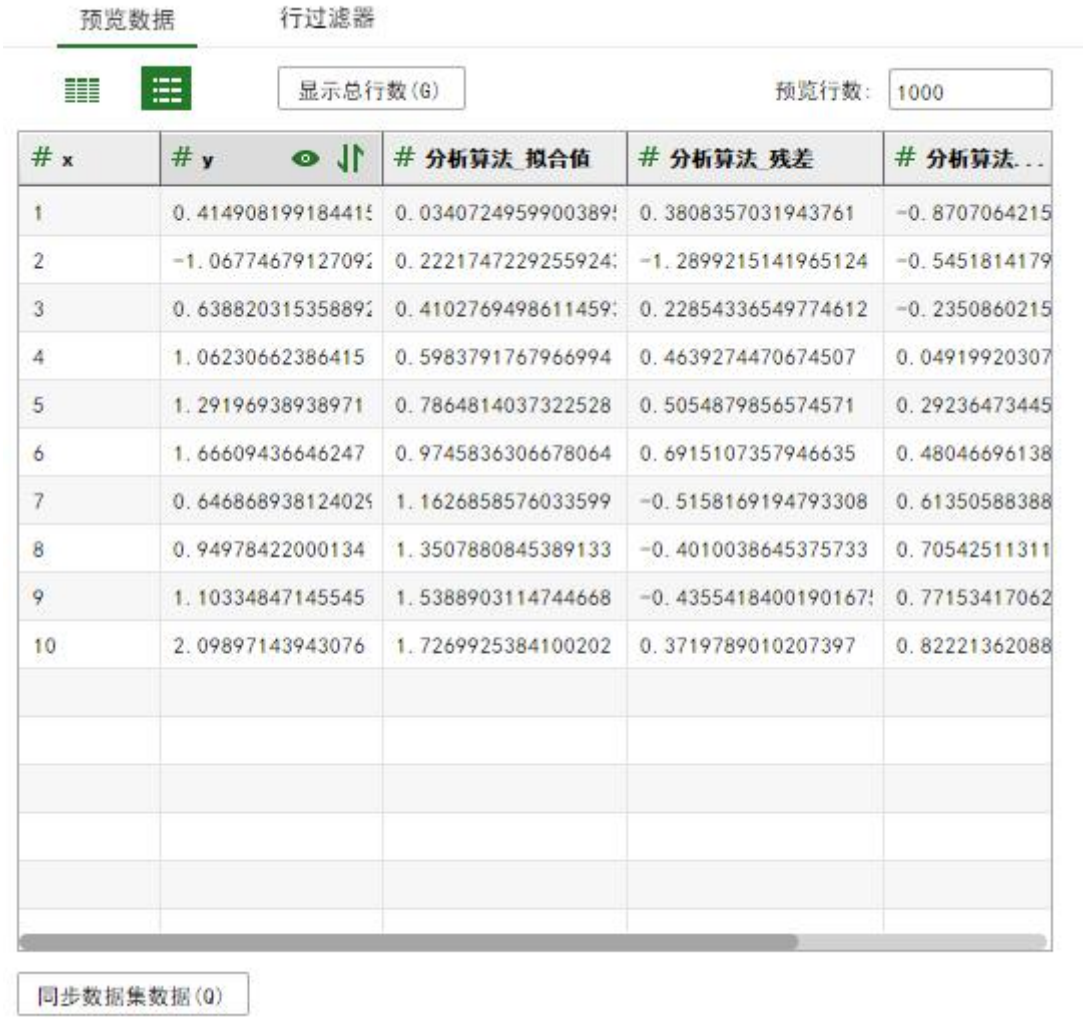
- **举例说明**

对一系列身高和体重的值进行线性回归分析。收集一系列身高和体重的值，身高为自变量，体重为因变量，使用一元线性回归分析算法找出所创建模型的数学方程。根据数学方程，计算体重的拟合值。

如图在数据集上创建分析算法：


分析算法
名称(N)：
分析算法
分析算法(A)：
一元线性回归
自变量(V)：
x
因变量(D)：
y
多项式次方(P)：
1
输出值：
拟合值(F)
残差(R)
置信区间(U)
95%
确定(O)
取消(C)

输出结果如图：



5.2. LDA 线性分类

❖ 一元线性回归

LDA 线性分类(Linear Discriminant Analysis, LDA) 的基本思想是将高维的模式样本投影到最佳鉴别矢量空间，以达到抽取分类信息和压缩特征空间维数的效果。投影后保证模式样本在新的子空间有最大的类间距离和最小的类内距离，即模式在该空间中有最佳的可分离性。

LDA 线性分类通过分类标签列和训练集数据列，调用 lda 函数得到 lda 模型，根据模型对新样本数据进行预测分类。

【分类标签列】样本数据标签列，每个样本数据都有对应的标签值。

【训练集】样本数据列。从左侧的可配置列中选择需要作为训练集的字段直接拖入到训练集框中。

【新样本】新样本数据列。从左侧的可配置列中选择需要作为新样本的字段直接拖入到新样本框中。新样本的数据行数可以和训练集数据行数不一致。

【输出值】【分类结果】新样本数据的预测标签值。

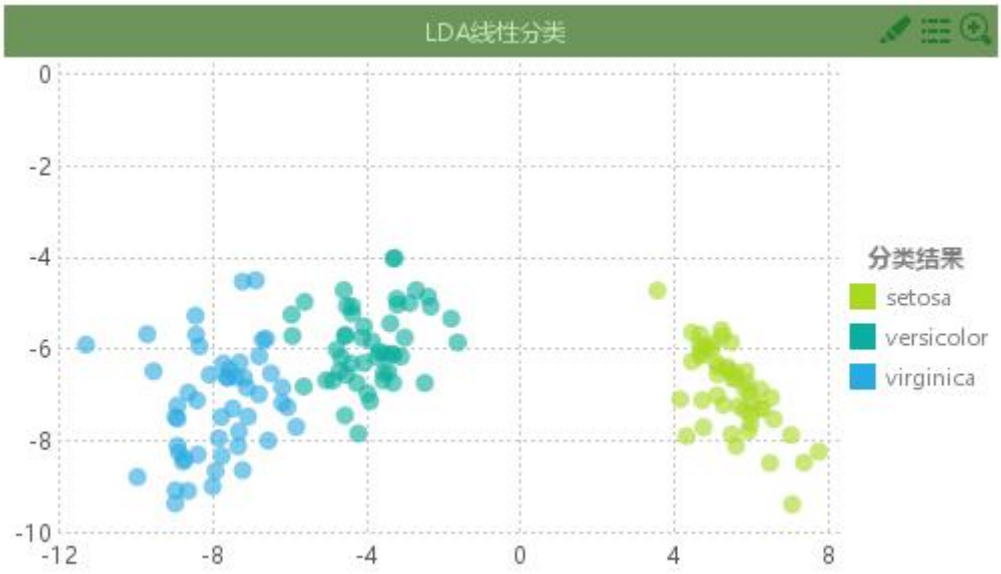
【输出值】【主成分】对分类的特征做主成分分析，取最重要的两个成分。

- **举例说明**

已知三种花的样本数据，根据样本数据（花瓣和萼片的长宽值）建立合适的线性分类函数，根据线性分类函数对新样本进行预测分类。

在图表上创建 LDA 线性分类分析，如图：

分类结果如图：



5.3. K-MEANS 聚类

❖ K-Means 聚类

K-Means 聚类要指定聚类的分类个数 N，随机取 N 个样本作为初始类的中心，计算各样本与类中心的距离并进行归类，所有样本划分完成后重新计算类中心，重复这个过程直到类中心不再变化。在 R 中使用 kmeans 函数进行 K-means 聚类。

kmeans(data,centers=3,nstart=10)，其中 centers 参数用来设置分类个数，nstart 参数用来设置取随机初始中心的次数，即运行 kmeans 方法的次数，我们在用 kmeans 函数时，默认取 10。

【聚类维度】聚类的样本集。从左侧的可配置列中选择需要作为聚类维度的字段直接拖入到聚类维度框中。

【设置 K 值】分类的个数。可以手动输入分类个数，也可以输入最大 K 值，系统根据轮廓系数计算出最佳的 K 值。

【输出值】【聚类标签】每个样本所属的类别。

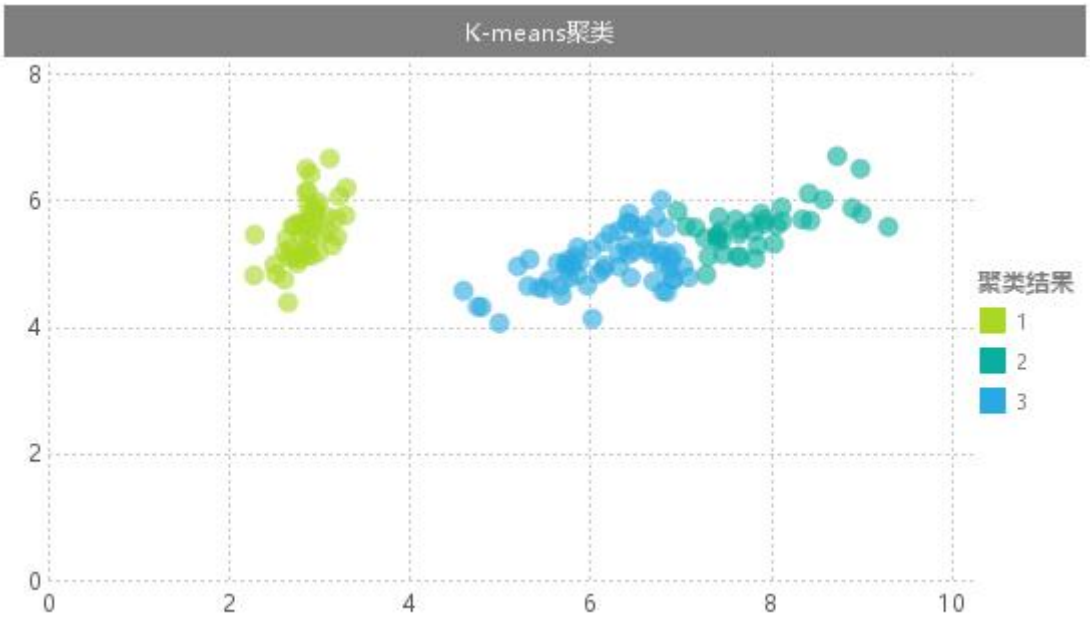
【输出值】【主成分】对聚类的维度做主成分分析，取最重要的两个成分。

- 举例说明

假设去掉类别列，根据四种属性对三种花进行分类。

在图表上创建 K-means 聚类分析，如图：

聚类结果如图：



5.4. HOLTWINTERS 时序分析

❖ HoltWinters 时序分析

HoltWinters 时序分析通过考虑水平趋势和季节性趋势，对一段时间内、等时间间隔的采样数据进行分析，以预测未来一段时间的数据。即根据已知的历史数据，预测未来的数据。

【时间列】选择时间字段。根据选择的时间字段的数据，自动算出时间间隔。

【数据列】选择数据字段。在报表组件绑定的数据集上新建分析算法，或使用图表的快速分析算法时，需要选择聚合函数，这样将按时间列进行分组，数据列进行聚合，在此分组聚合之后的数据上进行时序分析。

【周期】需填入时间间隔的整数倍，根据周期和时间间隔（周期/ 时间间隔）算出频率，即单位时间内的观测数。根据时间间隔，系统会自动往周期填入一个合理的数值，此数值也可手动修改。

【往后预测跨度】往后预测的时间跨度，需填入时间间隔的整数倍。选择时间列后，系统会自动填入一个合理的数值，此数值也可进行手动修改。

【趋势(beta)】是否考虑纵向趋势。默认是被勾选，表示按纵向趋势拟合。

【输出值】预测值被勾选上时，表示会得出一个拟合值字段。置信区间被勾选上时，表示会得出一个上界和一个下界字段。

【季节因子(gamma)】是否考虑季节性趋势。如果设置为不勾选（FALSE），则非季节性模型拟合。如果设置为勾选，则进行季节性模型拟合。季节性模式可以是加法效应（additive）和乘法效应（multiplicative）。加法效应默认勾选，表示按季节性加法的趋势增长。当乘法效应被勾选时，表示按季节性乘法趋势增长。季节性模型拟合时，需满足一个周期内至少有两个数据点，即频率大于等于 2，且时间序列至少包含 2 个周期。

【输出值】【预测值】被勾选时，会得到一个拟合值字段。其结果是根据得到的模型，对往后预测的时间跨度做预测，算出预测值。

【输出值】【置信区间】被勾选上时，根据 Level 算出估计值的上界和下界，默认 Level 是 95%。

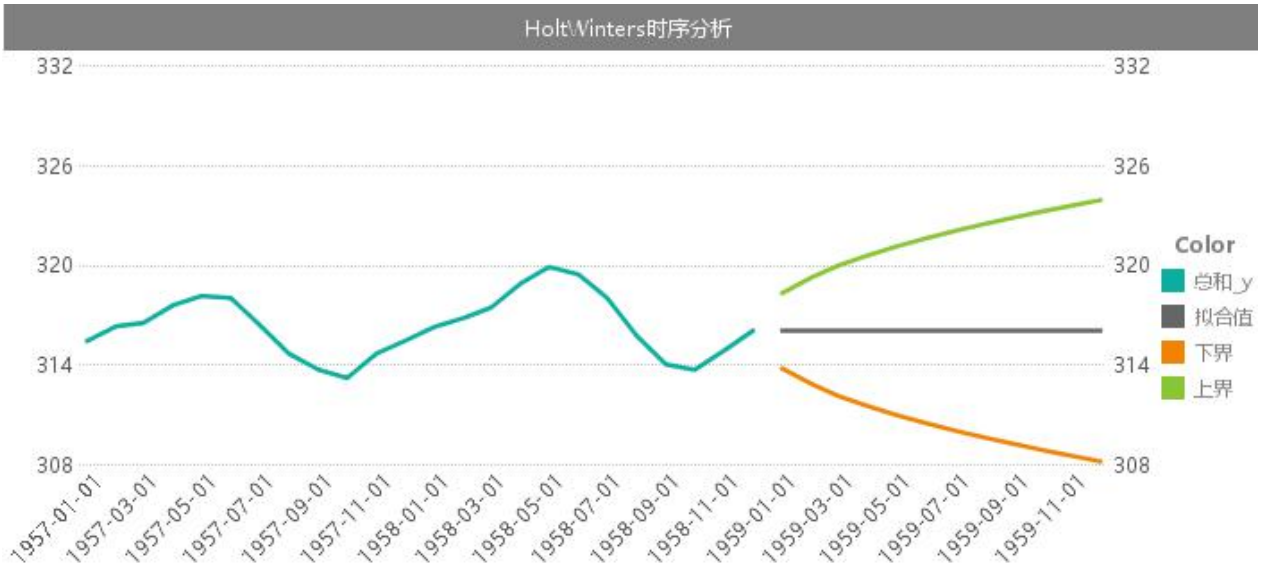
- **举例说明**

假设数据是从 1957 年 1 月到 1958 年 12 月的数据，时间间隔为 1 月，用 HoltWinters 时序分析，选择周期 6 个月，往后预测跨度 12 个月。

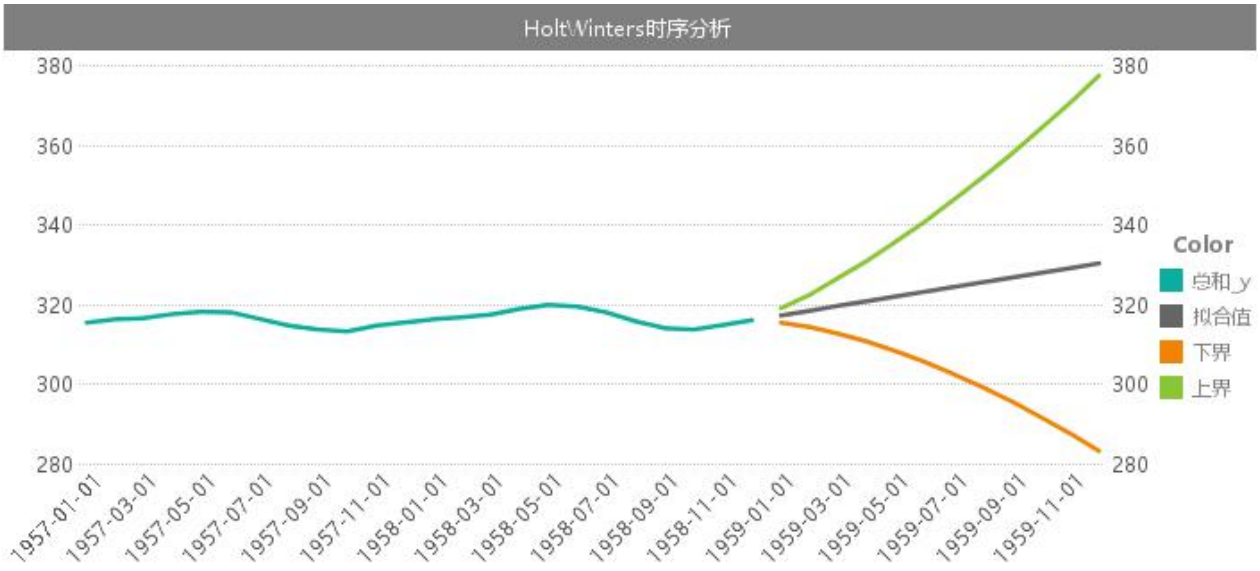
如果趋势（beta）选择否，季节因子（gamma）不勾选：


分析算法
名称(N):
分析算法(A):
HoltWinters时序分析
时间列(T):
year
数据间隔：1 年
周期(U):
2
年
请输入时间列间隔的整数倍
数据列(D):
GDP
空
往后预测跨度(B):
2
年
请输入时间列间隔的整数倍
趋势(beta):
是(R)
否(F)
季节因子(gamma):
是(Y)
加法效应(P)
乘法效应(M)
图表类型:
置信区间(U)
95%
确定(O)
取消(C)

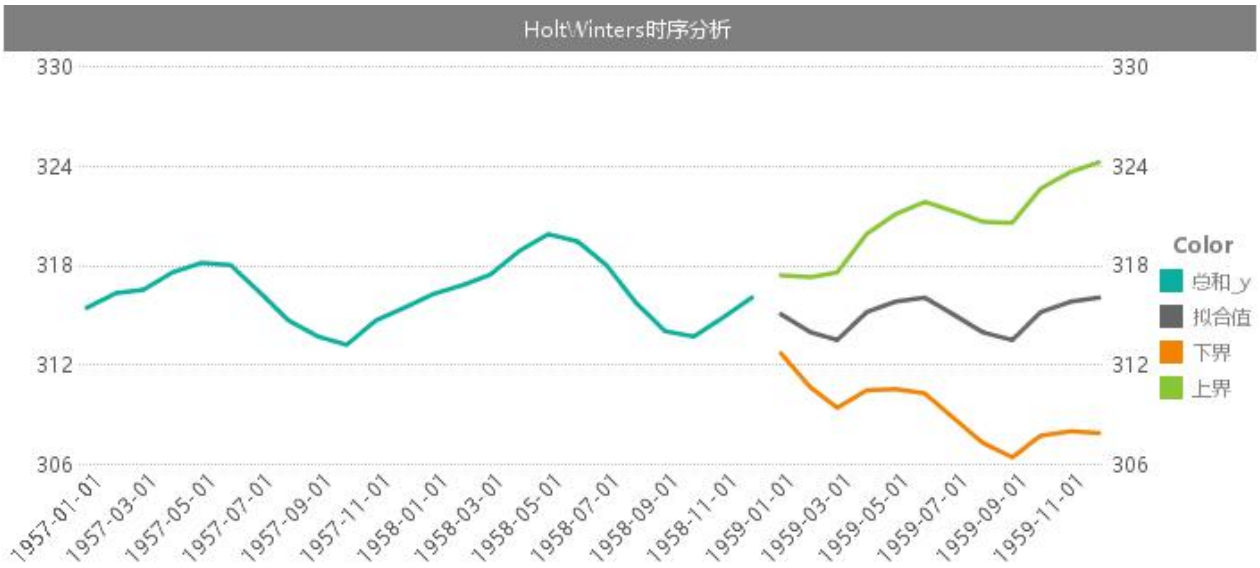
预测结果如图：


HoltWinters时序分析
332
326
320
314
308
Color
总和_y
拟合值
下界
上界
1957-01-01
1957-03-01
1957-05-01
1957-07-01
1957-09-01
1957-11-01
1958-01-01
1958-03-01
1958-05-01
1958-07-01
1958-09-01
1958-11-01
1959-01-01
1959-03-01
1959-05-01
1959-07-01
1959-09-01
1959-11-01

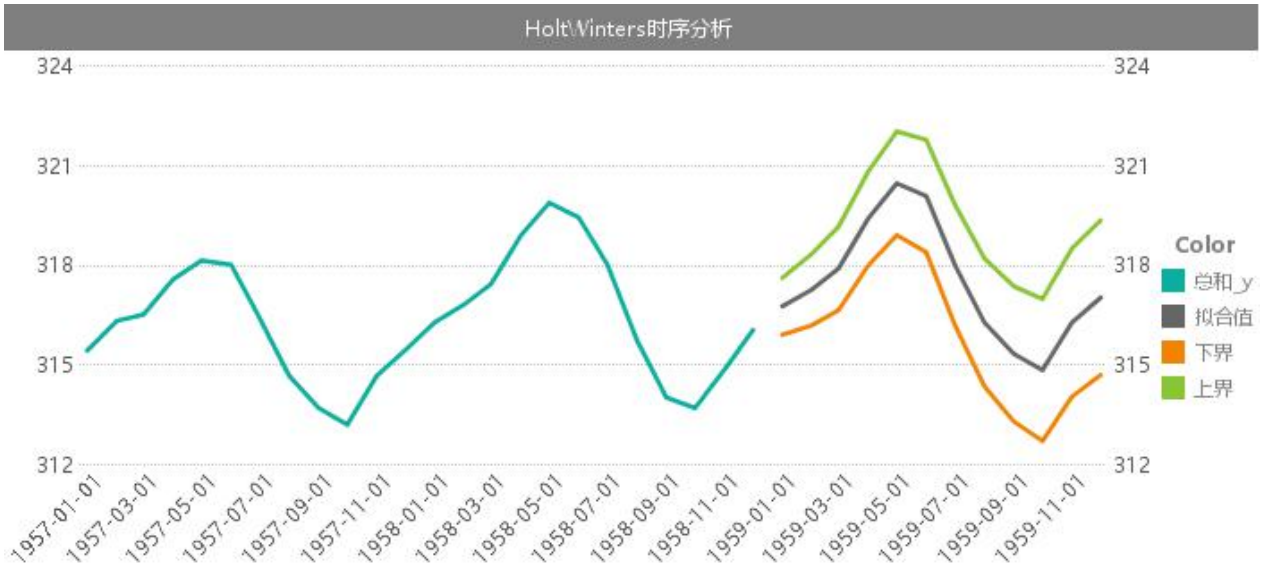
如果趋势（beta）选择是，季节因子（gamma）不勾选：



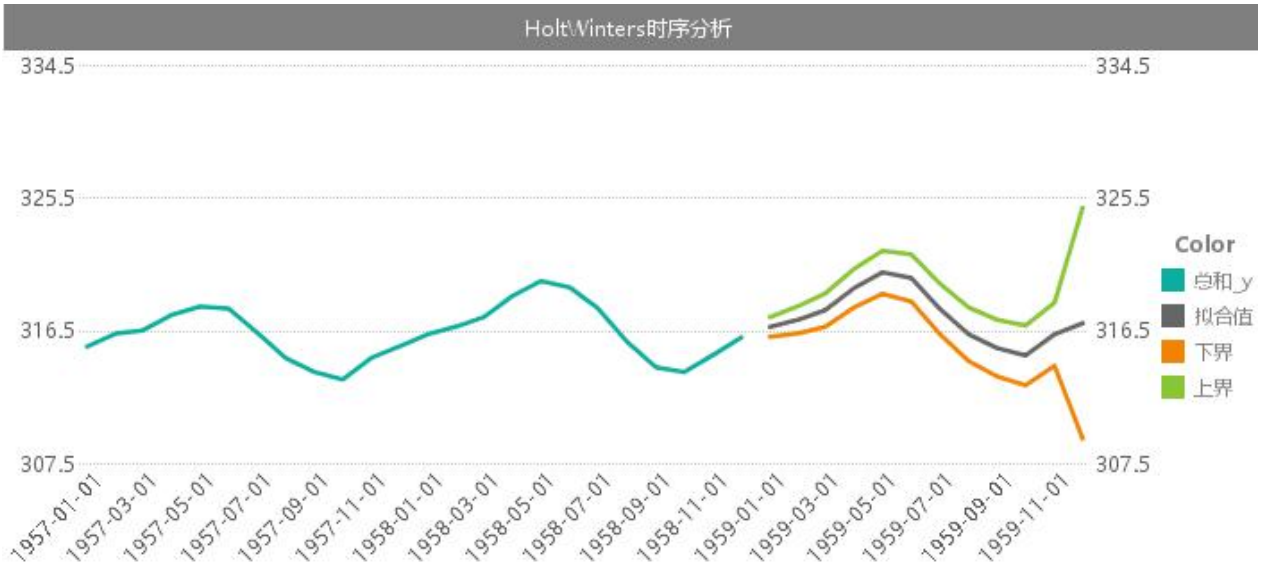
如果趋势（beta）选择否，季节因子（gamma）勾选，选择加法效应：



如果趋势（beta）选择是，季节因子（gamma）勾选，选择乘法效应：



如果趋势（beta）选择是，季节因子（gamma）勾选，选择加法效应：



5.5. 定制

❖ 定制

【计算类型】在连接数据的元数据区，默认细节计算处于置灰状态。在组件绑定数据集，默认细节计算处于选中状态。其下拉框中有细节计算和聚合计算，细节计算和聚合计算的区别是：聚合计算出来的 R 字段为聚合字段。

【脚本】输入脚本内容。可以通过 col[[“xxx”]] 来传入数据集中对应列的值,xxx 为列的名称；也可以通过 param[[“xxx”]]来传入参数值，xxx 为参数名称。对于定制脚本，R 将最后执行的代码行的结果作为返回值返回。Yonghong 产品中要求返回值必须是 list 对象，包含若干返回值列，如 list(out1=a, out2=b)，其中 out1,out2 为返回值列的名称，而 a，b 为相应返回值列的取值，可以是常数或向量。

其他 R 脚本请参考 R 官网。

- **举例说明**

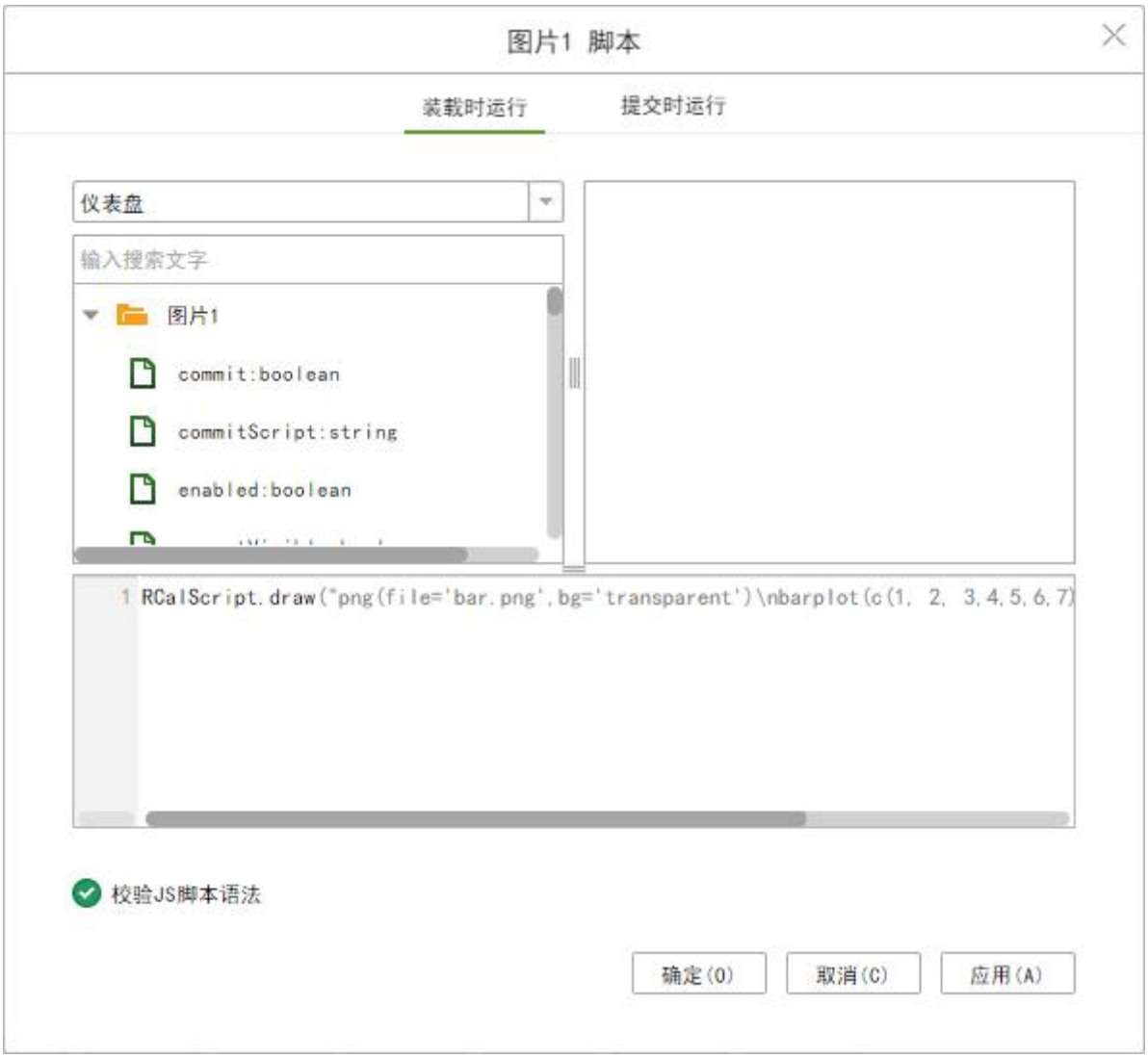
5.6. 动态更新 R 脚本生成图片

R 语言有丰富的绘图函数，系统为用户提供使用 R 脚本绘图的接口。通过 js 脚本在后台运行 R 脚本以生成图片，用户可通过 Image 组件动态加载生成的图片。产品目前支持三种类型：png 格式， jpg 格式， bmp 格式。

操作步骤如下：

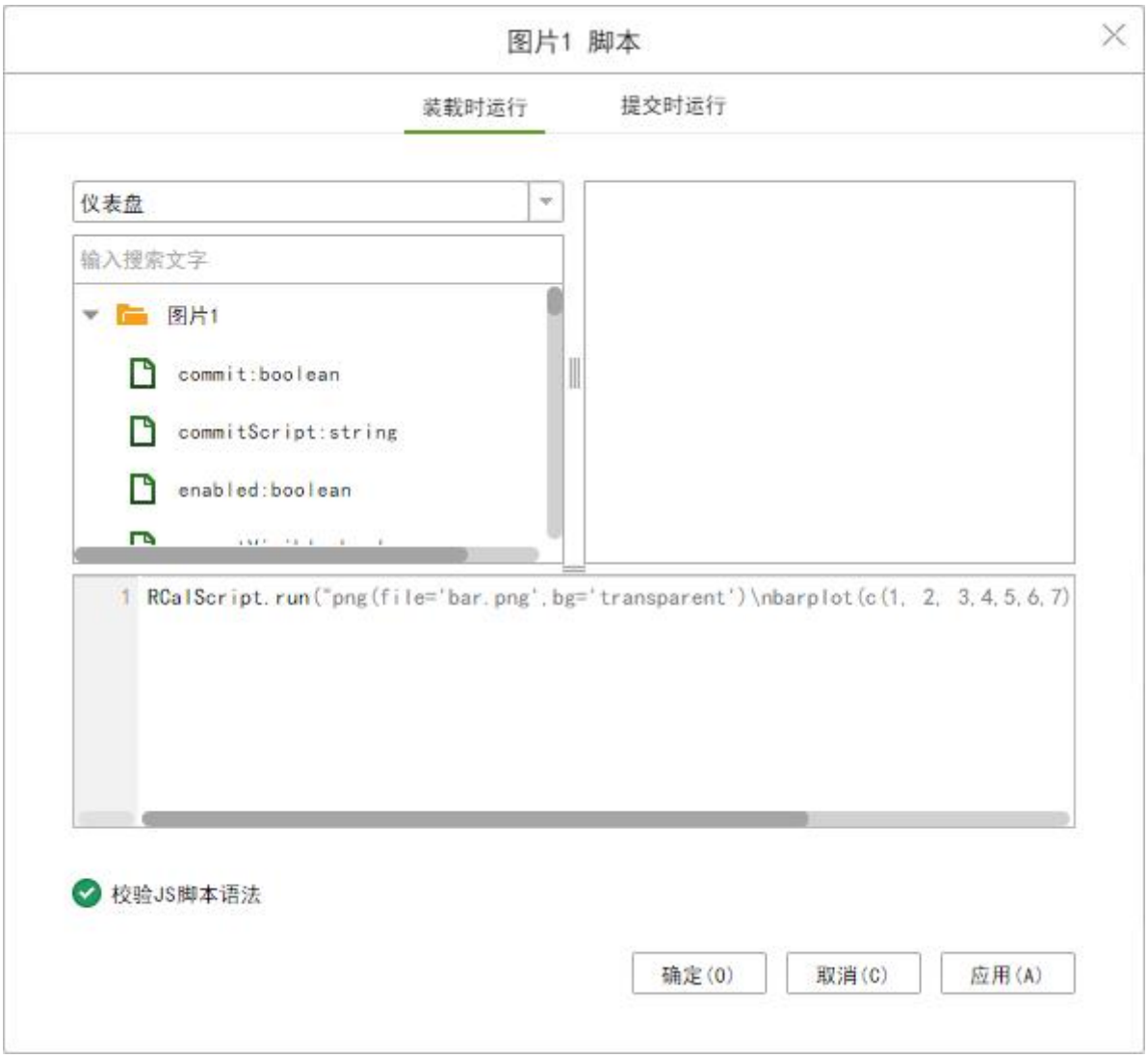
1. 使用 RCalScript 定义 Scriptable 对象，有如下方法：

（1）draw(String script, String path, boolean refresh):其中 script 指的是 R 脚本； path 指的是图片路径（包括名称、后缀，图片默认保存在 bihome/image 下）； refresh 指的是是否刷新，如果是则每次都生成新的图片，如果否则只在图片不存在的情况下生成图片。如果 R 脚本绘制失败，前台返回出错原因。



（2）run(String script)：

其中 script 指的是 R 脚本。如果 R 脚本绘制失败，前台返回出错原因。



2. 执行完绘图方法后，图片保存路径下就有相应的图片资源，在图片组件的属性对话框中刷新导入即可。


图片1 属性
属性
图片
高级
操作
图片
导入图片
输入搜索文字
bar.png
pie.jpg
plot.jpg
rodbc.png
填充(F)
缩放(S)
九宫格缩放(9)
顶部(T):
px
底部(B):
px
左(L):
px
右(I):
px
确定(O)
取消(C)
应用(A)