

算子平台快速入门



深圳力维智联技术有限公司

www.znv.com 400-886-6755

前言

版权说明

本资料著作权属深圳力维智联技术有限公司所有。未经著作权人书面许可，任何单位或个人不得以任何方式摘录、复制或翻译。

本手册将随产品进行不断更新，更新的内容将在新版本中加入，恕不另行专门通知。

版本更新说明

修订履历				
版号	日期	内容	责任人	审批人
V1.0	2020/03/09	初次编写	薛鹏	

目 录

智能算子平台使用手册.....	1
1 创建你的第一个算子流.....	4
2 对数据进行处理.....	8
3 数据融合.....	13
4 对数据进行统计分析.....	16
5 对数据进行可视化.....	18
6 用逻辑回归对鸢尾花进行分类.....	23



1 创建你的第一个算子流

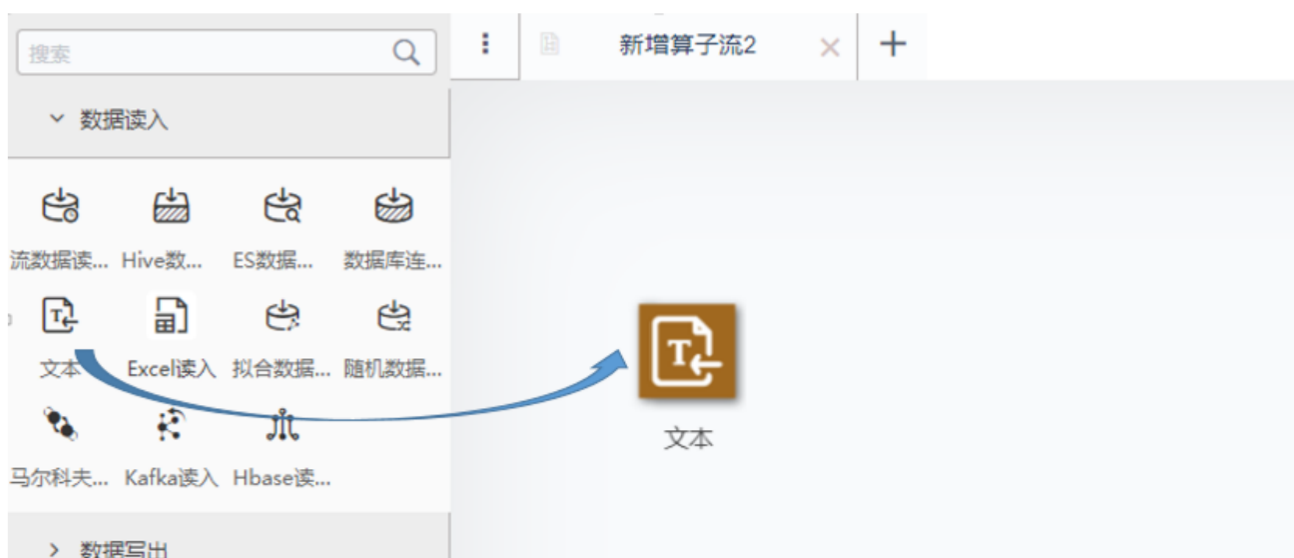
通过本例，你将学会如何：

1. 如何创建一个算子流
2. 如何编辑算子的属性
3. 如何连接两个算子
4. 如何运行一个算子流
5. 如何查看算子运行的结果

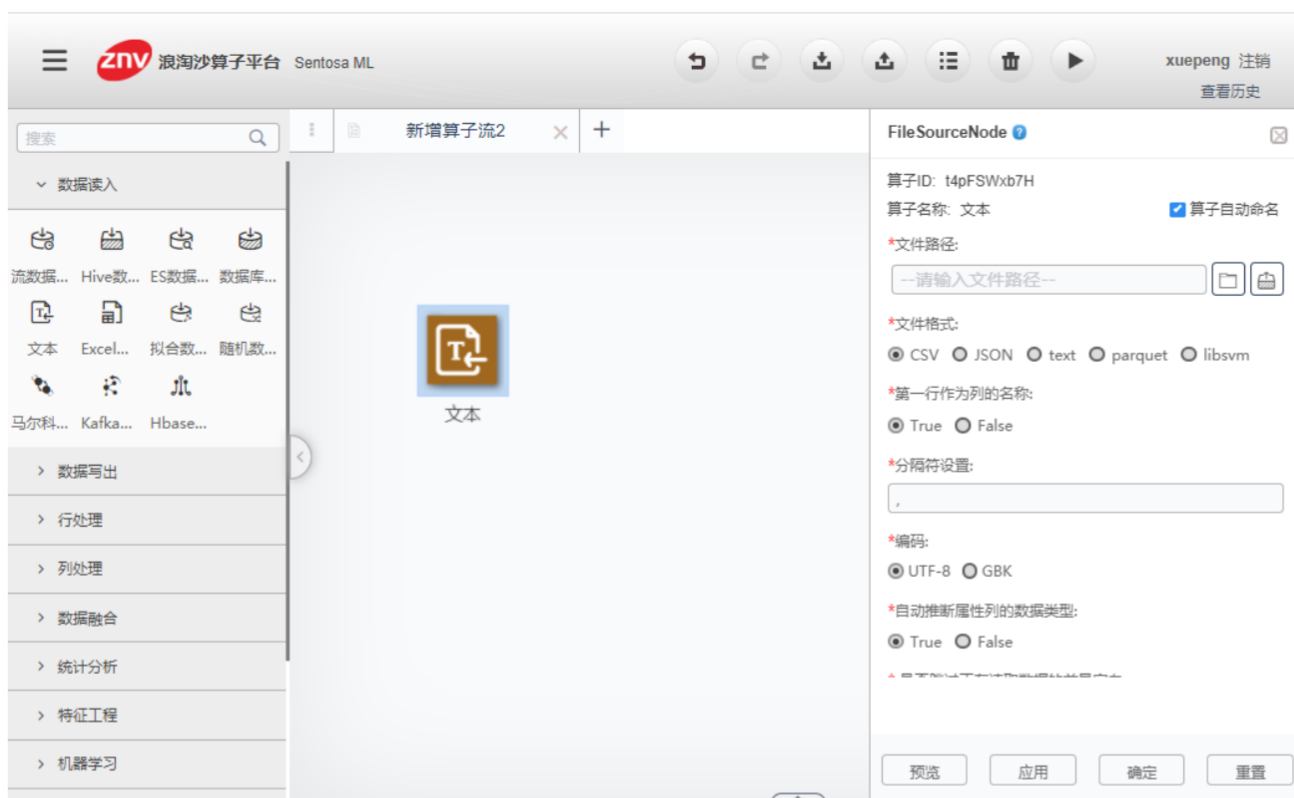
第一步，用管理员提供的用户名密码登录算子平台。



第二步，从数据读入算子里鼠标左键点选并拖拽文本算子到画布上。



第三步，双击文本算子，打开算子属性编辑页面。



第四步，上传文件。通过点击上传文件按钮来上传文件。



在弹出的文件上传对话框里选择 Sentosa 服务器目录，本地文件上传后在服务器存放的目录。

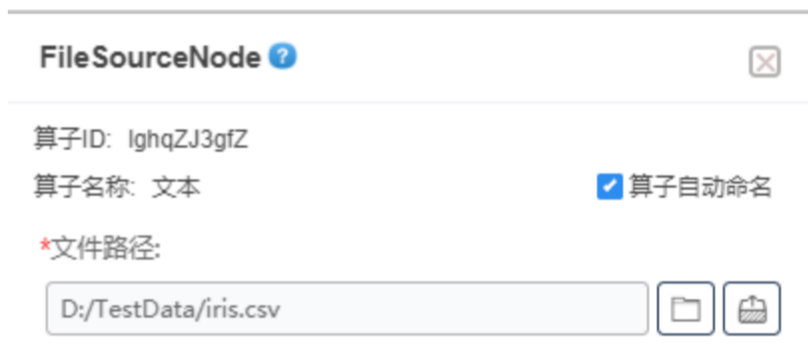


点击选择本地测试文件 iris.csv:



最后点击确定上传文件。

第五步，通过文本路径控件选择刚才上传的文件：iris.csv



第六步，点击文本算子属性页面的确定按钮。完成对文本算子的配置。

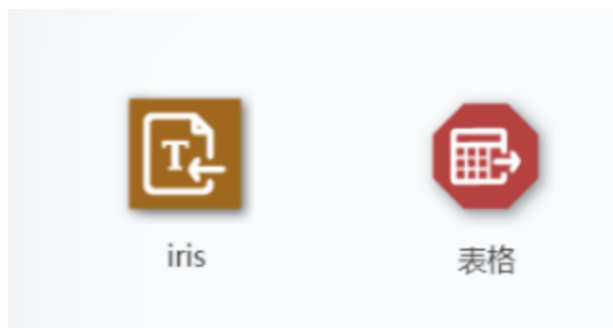
第七步，预览数据。在文本算子上点击右键，选择预览



通过预览，可以以表格的形式预览数据。

iris算子预览					
iris	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
	5.100	3.500	1.400	0.200	setosa
	4.900	3	1.400	0.200	setosa
	4.700	3.200	1.300	0.200	setosa
	4.600	3.100	1.500	0.200	setosa
	5	3.600	1.400	0.200	setosa
	5.400	3.900	1.700	0.400	setosa
	4.600	3.400	1.400	0.300	setosa
	5	3.400	1.500	0.200	setosa
	4.400	2.900	1.400	0.200	setosa
	4.900	3.100	1.500	0.100	setosa

第八步，从图表分析算子里拖拽表格算子到画布上。和预览类似，表格算子可以将数据以表格的形式在结果里展现。而且表格算子的属性编辑页面里可以指定你要展现的数据条数。



第九步，鼠标移动到文本算子上，会出现一个小圆圈，点击拖拽这个小圆圈到表格算子上，然后松开鼠标左键。文本算子会和表格算子连接起来。



第十步，点击运行按钮，执行整个算子流：



第十一步，查看算子流执行结果：

通过点击[查看历史](#)来查看历史执行记录：



点击查看最近一次执行结果:

新增算子流3						全屏切换 X
表格	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species	
日志	5.100	3.500	1.400	0.200	setosa	
	4.900	3	1.400	0.200	setosa	
	4.700	3.200	1.300	0.200	setosa	
	4.600	3.100	1.500	0.200	setosa	
	5	3.600	1.400	0.200	setosa	
	5.400	3.900	1.700	0.400	setosa	
	4.600	3.400	1.400	0.300	setosa	
	5	3.400	1.500	0.200	setosa	
	4.400	2.900	1.400	0.200	setosa	
	4.900	3.100	1.500	0.100	setosa	

2 对数据进行处理

第一步，在画布上添加文本算子，并配置 iris.csv 文件为文本路径。

第二步，从行处理算子里拖拽填充算子到画布，并连接文本算子和填充算子。（填充算子可以将数据某一列里满足某种条件的值进行替换。）



第三步，双击填充算子，进入填充算子属性配置列。选择填充列为 Sepal.Length，填充条件为 `Sepal.Length < 5`，填充值为 5。（在进行填充条件和填充值属性设置时，通过表达式编辑器进行设置。）

表达式输入

构建表达式:

表达式校验

`Sepal\_Length`<5

C

+

-

*

/

>

<

>=

<=

==

(

)

'

%

or

and

like

rlike

as

函数

搜索

一般函数

列

函数	返回类型
coalesce(expr1, expr2, ...)	不定
greatest(expr1, expr2, ...)	不定
isnotnull(expr)	Boolean
least(expr1, expr2, ...)	不定
nanvl(expr1, expr2)	不定
nvl2(expr1, expr2, expr3)	不定
rand([seed])	Double

字段	存储类型
Sepal_Length	Double
Sepal_Width	Double
Petal_Length	Double
Petal_Width	Double
Species	String

取消

确定

FillNode ?



算子ID: qfufNtUhpE

算子名称: 填充

☒ 算子自动命名

*填充列

Sepal_Length

*填充条件

`Sepal\_Length`<5

*填充值

5

第四步，点击填充算子的确定按钮。

第五步，在填充算子上点击数据右键，选择预览。可以发现 Sepal.Length 里小于 5 的值都被填充为 5。

填充算子				
填充	Sepal_Length		Sepal_Width	Peta
	5.100		3.500	1.40
	5		3	1.40
	5		3.200	1.30
	5		3.100	1.50
	5		3.600	1.40
	5.400		3.900	1.70
	5		3.400	1.40
	5		3.400	1.50
	5		2.900	1.40
	5		3.100	1.50

第六步，关闭预览窗口，从行处理算子里拖拽出排序算子，并连接填充算子和排序算子。



第七步，双击排序算子，点击“+”增加一个排序列，并指定在排序列上进行升序 (asc) 或者进行降序 (desc) 排列。我们分别增加 Petal_Length 升序和对 Petal_Width 降序排列。

SortNode ?

✕

算子ID: hEEJPEnV75

算子名称: 排序 ☒ 算子自动命名

*列名

*排序方式

Petal_Length

asc

—

Petal_Width

desc

—

+

第八步，点击排序算子属性编辑页的确定，并在排序算子上右键选择预览。可以发现数据已经按我们设置进行排序，并且保留了之前填充算子对 Sepal_Length 列的填充处理。

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
5	3.600	1	0.200	setosa
5	3	1.100	0.100	setosa
5	3.200	1.200	0.200	setosa
5.800	4	1.200	0.200	setosa
5.400	3.900	1.300	0.400	setosa
5	2.300	1.300	0.300	setosa
5	3.500	1.300	0.300	setosa
5	3	1.300	0.200	setosa
5	3.200	1.300	0.200	setosa
5	3.200	1.300	0.200	setosa

第九步，从行处理算子里拖拽过滤算子，并连接排序算子和过滤算子。



第十步，双击过滤算子，进入过滤算子属性编辑页面。我们期望选择“Species”里以字母“V”开头的数据条目。

第十一步，在函数选择控件里，首先过滤“字符串处理函数”



第十二步，单击选择 instr 函数，会展现这个函数的说明和示例。

函数

搜索

字符串处...

函数

返回类型

instr(str, substr)

Integer

length(expr)

Integer

levenshtein(str1, str2)

Integer

lower(str)

String

octet_length(expr)

Integer

字段

存储类型

Sepal_Length

Double

Sepal_Width

Double

Petal_Length

Double

Petal_Width

Double

Species

String

函数说明: 传入两个参数，第一个是字符串参数，第二个是要查询的字符串，查找第二个字符串在第一个字符串中首次出现的位数

参数说明: 参数str:基准str 参数substr:要查找的字符串

返回值: 返回index值

示例: instr('SparkSQL', 'SQL')

结果: 6

第十三步，双击 **instr** 这个函数，它会自动填充到构建表达式输入框，然后双击右侧 **Species** 列名，它会填充到 **instr** 的第一个参数，然后我们手动补全整个表达式：**instr('Species','v')=1**。

算子ID: fGZpC1ox5h
算子名称: 过滤

☒ 算子自动命名

*是否保留过滤项:
☒ 是 ☐ 否

*构建表达式:

表达式校验

instr('Species','v')=1

第十四步，在**过滤算子**上预览，我们发现数据流过该节点，只留下满足我们设定条件 **Species** 字段以 **V** 开头的记录。

第十五步，从行处理算子里选取**聚合算子**，并连接**过滤算子**和**聚合算子**。

第十六步，双击**聚合算子**，打开**聚合算子**的属性页面，进行编辑。

从可用的列里选择 **Species** 作为用来聚合的列。对不同的 **Species** 分别求以下的值：

Speal_Length 的平均值，

Speal_width 的最大值

Petal_length 的最小值

petal_Width 的标准差

AggregateNode ?

算子ID: 7UQzvUob6U
算子名称: 聚合

☒ 算子自动命名

可用的列4

请输入搜索内容

☐ Sepal_Length
☐ Sepal_Width
☐ Petal_Length
☐ Petal_Width

用来聚合的列1

请输入搜索内容

☐ Species

是否生成count列: ☒ 是 ☐ 否

count列列名: count

字段	近似去重总数	平均值	数量	第一个	峰度	最后一个	最大值	最小值	偏度	标准差	总体标准差
Sepal_Length	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐
Sepal_Width	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐
Petal_Length	☐	☐	☐	☐	☐	☐	☐	☒	☐	☐	☐
Petal_Width	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☐

<本文中的所有信息归深圳力维智联技术有限公司所有，未经允许，不得外传>

第 12 / 44 页



第十七步，在聚合算子后边连接表格算子，并执行整个算子流。在查看历史里，单击打开最后一次执行的结果。

Species	avg(Sepal_Length)	max(Sepal_Width)	min(Petal_Length)	stddev(Petal_Width)	count
virginica	6.590	3.800	4.500	0.275	50
versicolor	5.938	3.400	3	0.198	50

3 数据融合

第一步，通过文本算子上传测试数据里的以下数据：

基站 A 电流.txt

基站 A 电压.txt

基站 B 电流.txt

基站 B 电压.txt

第二步，拖拽两个文本算子到画布，并设置它们的文本路径分别为“基站 A 电压”和“基站 A 电流”。同时拖拽一个合并算子到画布。并将两个文本算子都连接到合并算子上。



第三步，双击打开合并算子的属性编辑页面，从可选关键字里选择“设备 ID”，移动到右边用于合并的关键字。

MergeNode

算子ID: H02TKRTouS

算子名称: 合并

☒ 算子自动命名

	1	ij0tgi7lxJ	基站A电压
	2	K8QrDsjsy6T	基站A电流

可用的关键字

0

请输入搜索内容

列表为空

<

>

用于合并的关键

请输入搜索内容

设备id

列名

字段	节点标识	节点名	新列名

连接方式:

☒ 内连接(inner)

☐ 左连接(left)

☐ 右连接(right)

☐ 全外连接(full)

☐ 左半连接(left_semi)

☐ 左反连接(left_anti)

预览

应用

确定

重置

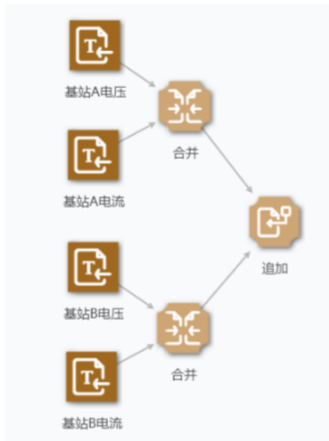
第四步，同样步骤添加“基站 B 电压”和“基站 B 电流”文本算子以及合并算子。

<本文中的所有信息归深圳力维智联技术有限公司所有，未经允许，不得外传>

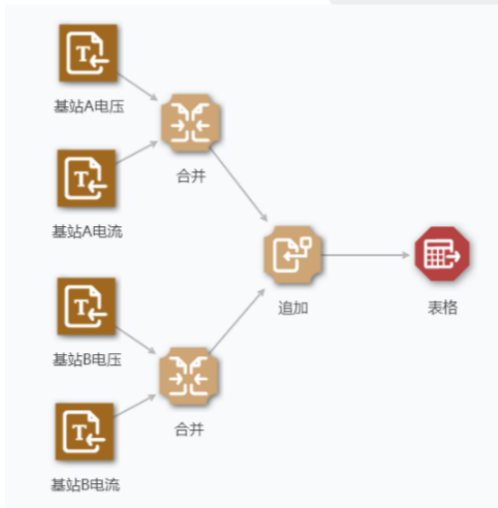
第 14 / 44 页



第五步，拖拽一个追加算子到画布，并将两个合并算子连接到追加算子上。



第六步，在追加算子后边连接一个表格算子。然后执行整个算子流。得到结果如下：



设备id	电压	电流
1	239	23
2	218	22
3	220	21
4	229	13
5	228	12
6	210	11

4 对数据进行统计分析

通过本例你将学到：

1. 如何判断数据中离散型变量之间的相关性。
2. 如何判断数据中连续型变量之间的相关性。

我们将对一份销售数据进行分析，探索离散型变量：顾客性别，产品种类，以及顾客是否为 vip 之间是否有相关性。以及探索连续型变量：年龄，消费金额，注册年限之间是否有相关性。

第一步，拖拽一个文本算子，上传测试数据里的“销售数据.txt”，并设置“销售数据.txt”为文件路径。并在文本算子上进行预览。

性别	年龄	金额	种类	是否VIP	注册年限
男	15	10	电子产品	是	2
女	20	100	化妆品	否	1
男	20	800	电子产品	是	3
女	30	2000	化妆品	否	1
男	23	1000	玩具	是	2
女	20	200	零食	否	4
女	18	140	零食	是	1
男	28	1800	电子产品	否	2

第二步，从统计分析算子里拖拽一个卡方检验算子到画布，并连接文本算子到卡方检验算子。



第三步，双击卡方检验算子，对卡方检验算子属性进行配置。这里我们选择离散型变量：“性别”，“种类”，“是否VIP”三列来进行卡方检验。

ChiSquareNode ?



算子ID: MEDzLeSd3m

算子名称: 卡方检验

☒ 算子自动命名

*输入列表

性别 × 种类 × 是否VIP ×



第四步，拖拽一个表格算子到画布，并连接卡方检验算子与表格算子。然后运行算子流。



新增算子流1					全屏切换 X	
表格	ChiSquare_cols/pValues	性别	种类	是否VIP		
日志	性别	0.005	0.046	0.157		
	种类	0.046	0.004	0.343		
	是否VIP	0.157	0.343	0.005		

第五步，结果分析，如果我们设定 p 值为 0.05,则可发现“种类”和“性别”是有关联性，而“种类”与“是否为VIP”，“性别”与“是否VIP”都没有统计学差异。

第六步，从统计分析算子里拖拽皮尔森相关系数算子，并连接文本算子与皮尔森相关系数算子。



第七步，双击皮尔森相关系数算子，进入属性编辑页面，选择连续型变量“年龄”，“金额”，“注册年限”并保存设置。

PearsonCorrelationNode ? ✕

算子ID: 7Ak6U3tKrs

算子名称: 皮尔森相关性系数 ☒ 算子自动命名

*输入列表

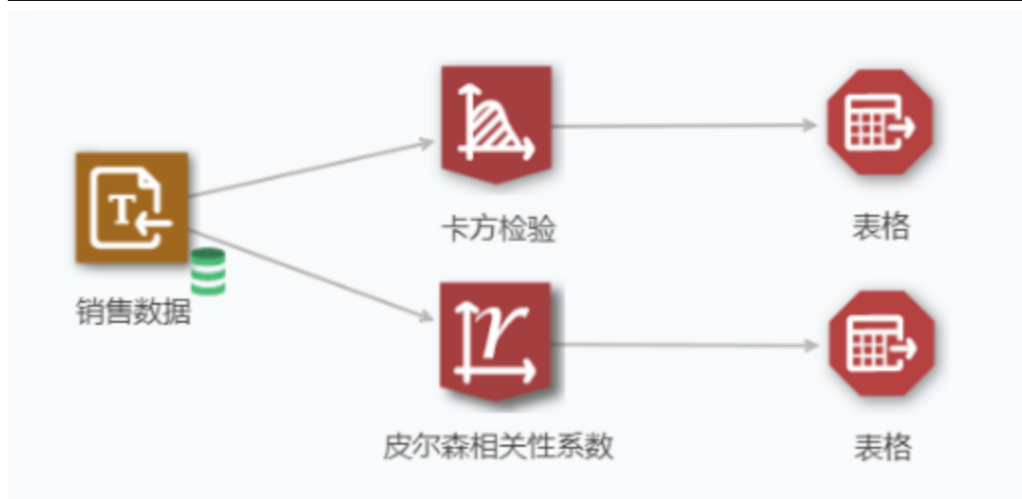
年龄 ✕

金额 ✕

注册年限 ✕

▼

第八步，拖拽一个表格算子，并连接皮尔森相关系数算子与表格算子。



第九步，运行算子流，查看结果。可以发现有两个表格结果。一个为卡方检验结果，一个为皮尔森相关系数结果。我们选择到皮尔森相关系数结果页面。我们可以发现“金额”和年龄有强相关性。

新增算子流1					全屏切换 X
表格	Correlation_cols/Coe_pearson	年龄	金额	注册年限	
表格	年龄	1	0.950	-0.212	
	金额	0.950	1	-0.176	
日志	注册年限	-0.212	-0.176	1	

5 对数据进行可视化

在本例中，你将学会如何利用算子平台来对数据进行可视化。

第一步，拖拽一个文本算子，上传测试数据里的“iris.csv”，并设置“iris.csv”为文件路径。并在文本算子上进行预览。

第二步，从图表算子里拖拽一个散点图算子到画布上，并连接文本算子与散点图算子。



第三步，双击散点图算子，进入散点图算子属性编辑页面。按照如下配置设置散点图属性。并点击保存。

ScattergramNode ?



算子ID: fj4oaRLiR3

算子名称: 散点图

☒ 算子自动命名

*标题:

散点图

*子标题:

散点图

抽样因子:

1

数据提取行数:

2000

*X轴名:

Sepal_Length

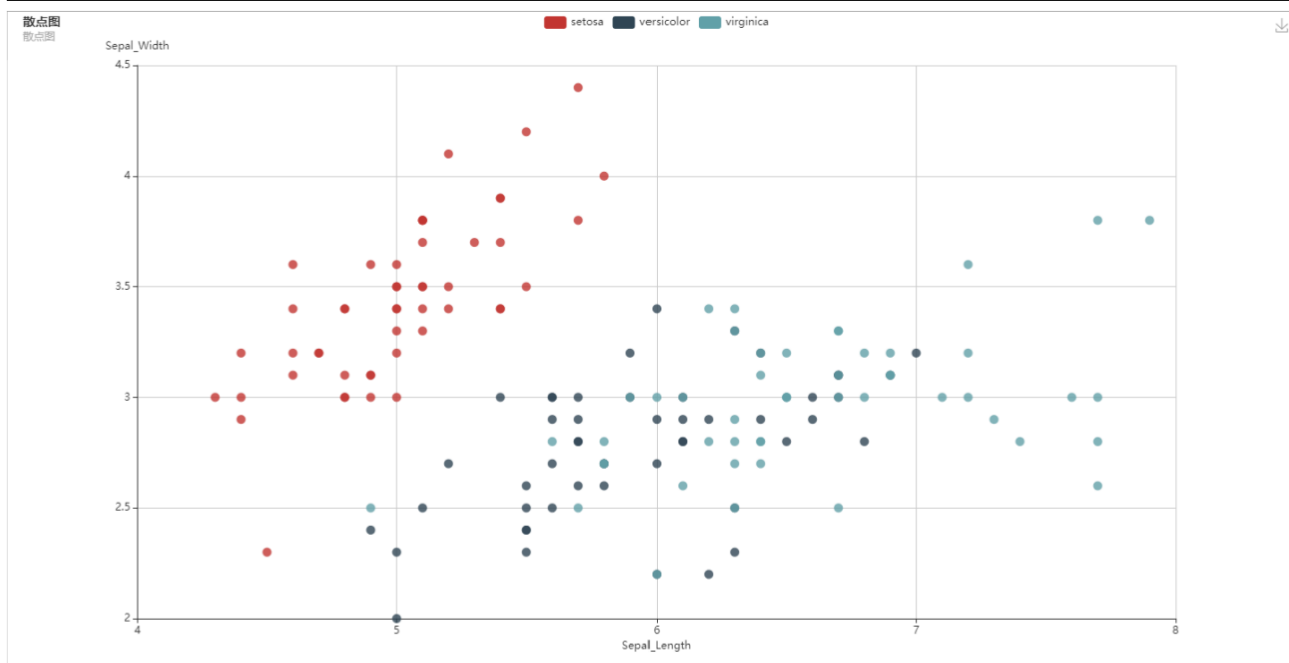
*Y轴名:

Sepal_Width

分组列名:

Species

第四步，运行算子流。从结果页面里打开最后一次运行结果。



第五步，从图标分析算子里拖拽柱状图算子，并连接文本算子与柱状图算子。并按照如下配置柱状图属性：

BarNode ?

算子ID: yamoRnhJSN

算子名称: Sepal_Length-Sepal_Wid... ☒ 算子自动命名

***标题**
柱状图

***子标题**
柱状图

***抽样因子**
1

***数据提取行数**
2000

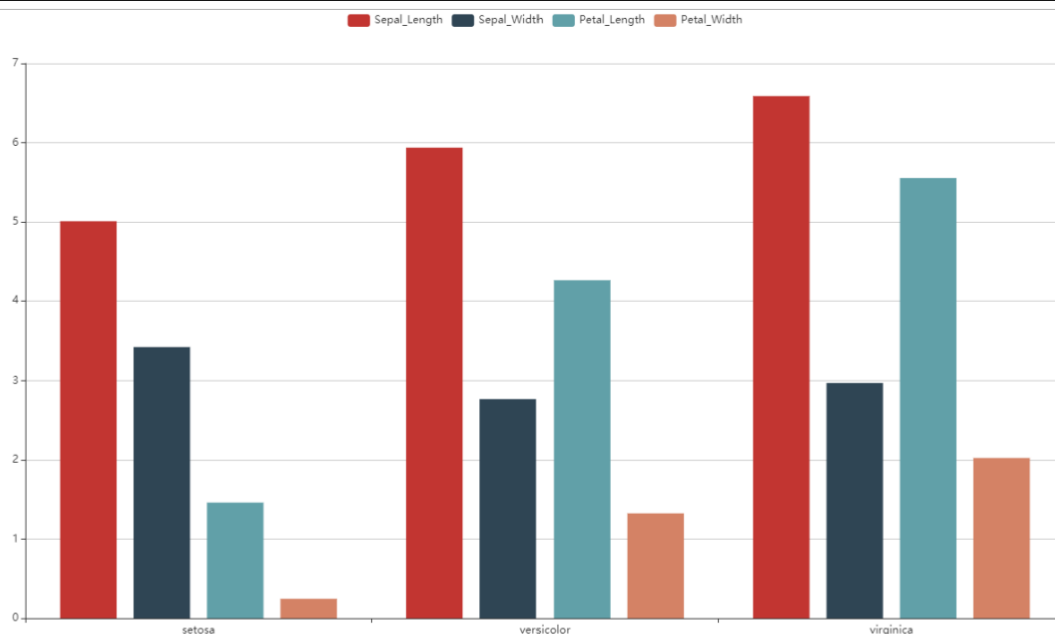
分组列
Species

***统计列名**
Sepal_Length x Sepal_Width x Petal_Length x
Petal_Width x

***统计方法**
avg

第六步，运行算子流。得到柱状图如下：

柱状图
柱状图



第七步，从图标分析算子里拖拽二维雷达图到画布，并连接文本算子与二维雷达图。并按照如下配置设置雷达图属性：

RadarNode ?

算子ID: CaxcHJNfC1

算子名称: Sepal_Length,Sepal_Widt...

☒ 算子自动命名

*标题:

二维雷达图

*子标题:

二维雷达图

抽样因子:

1

数据提取行数:

2000

*顶点列:

Sepal_Length ×

Sepal_Width ×

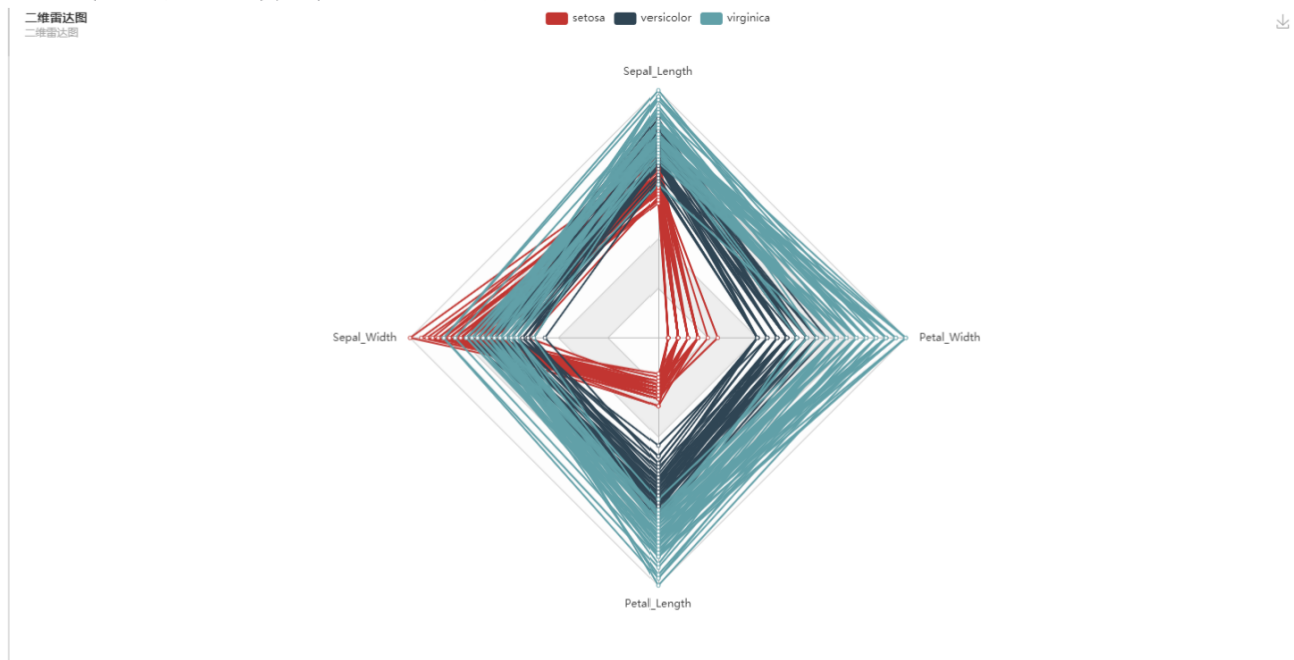
Petal_Length ×

Petal_Width ×

分组列名:

Species

第八步，运行雷达图算子，得到雷达图如下：



第九步，从图标分析算子里拖拽平行关系图到画布，并连接文本算子与平行关系图，配置平行关系图属性如下：

ParallelgramNode ?

算子ID: yJPRUe4hQp

算子名称: Sepal_Length, Sepal_Width... ☒ 算子自动命名

*标题:

平行关系图

*子标题:

平行关系图

抽样因子:

1

数据提取行数:

2000

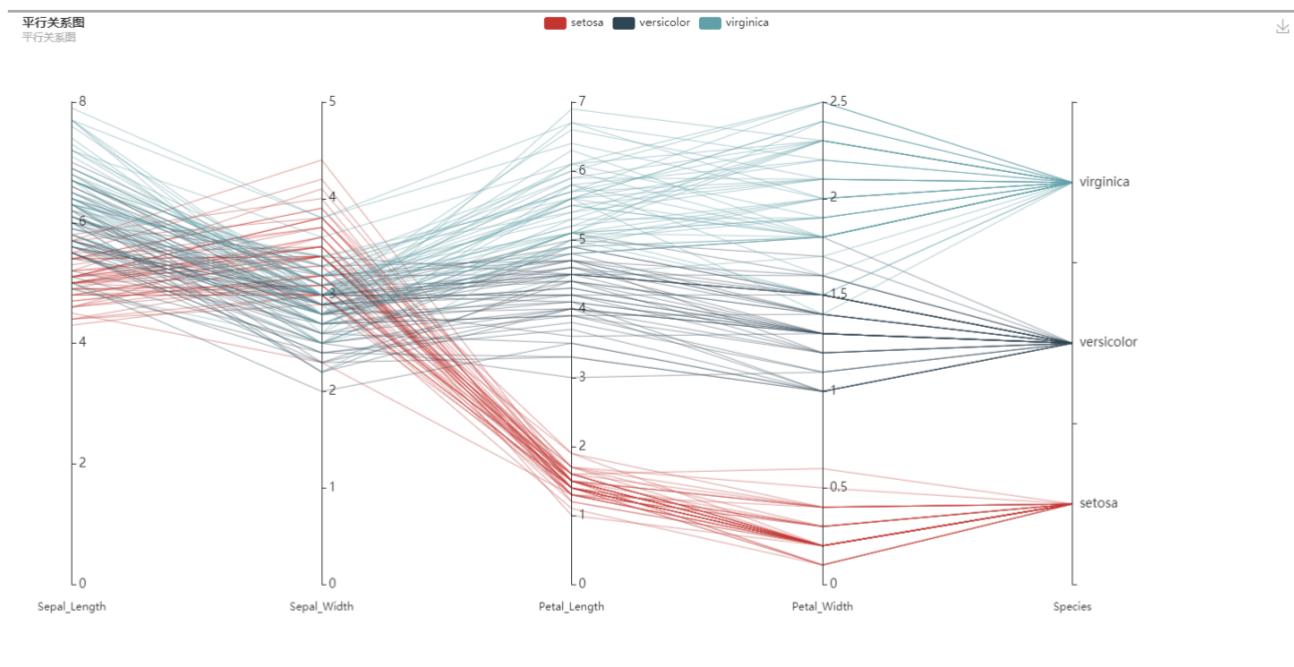
*统计列名:

Sepal_Length × Sepal_Width × Petal_Length ×
Petal_Width × Species ×

分组列名:

Species

第十步，运行平行关系图算子，得到平行关系图如下：



6 用逻辑回归对鸢尾花进行分类

在本例中，我们将训练一个模型，利用样本数据建立鸢尾花类型与其花瓣长度，花瓣宽度，花萼长度，花萼宽度的关系，从而通过花瓣，花萼的长度，宽度来预测一个鸢尾花的具体类型。

第一步，拖拽一个文本算子，上传测试数据里的“iris.csv”，并设置“iris.csv”为文件路径。并在文本算子上进行预览。我们准备用前 4 列作为 feature 列，用最后一列作为 label 列。

iris算子预览					全屏切换 X
iris	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
	5.100	3.500	1.400	0.200	setosa
	4.900	3	1.400	0.200	setosa
	4.700	3.200	1.300	0.200	setosa
	4.600	3.100	1.500	0.200	setosa
	5	3.600	1.400	0.200	setosa
	5.400	3.900	1.700	0.400	setosa
	4.600	3.400	1.400	0.300	setosa
	5	3.400	1.500	0.200	setosa
	4.400	2.900	1.400	0.200	setosa
	4.900	3.100	1.500	0.100	setosa

第二步，用归一化算子对 feature 列进行归一化。从特征工程算子里拖拽归一化算子到画布，并连接文本算子与归一化算子。



第三步，双击打开归一化算子，按照如下属性配置归一化算子：

BuildRescaleNode ?

算子ID: Te1yCMCkcf

算子名称: 归一化

☒ 算子自动命名

*归一化列

Sepal_Length ×

Sepal_Width ×

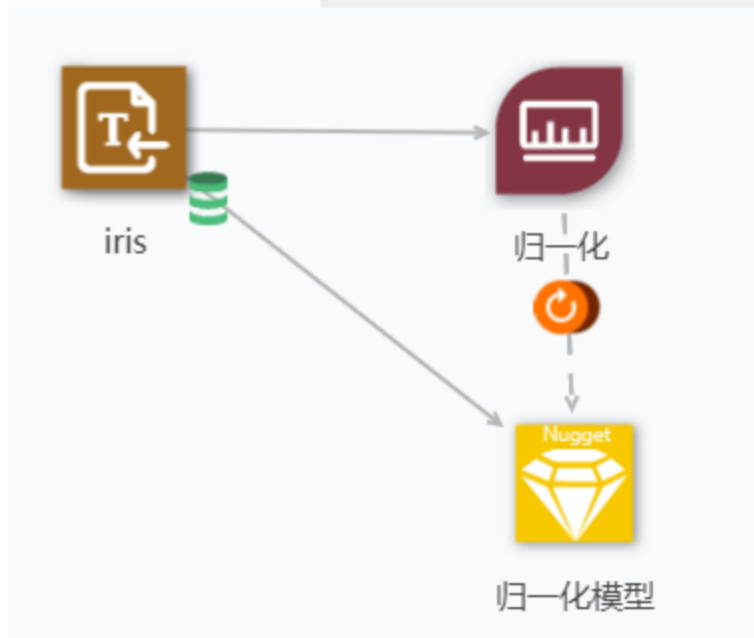
Petal_Length ×

Petal_Width ×

*算法类型:

☒ 绝对值最大标准化
☐ 最大最小值标准化

第四步，在归一化算子上点击数据右键，选择运行，得到归一化模型。



第五步，在归一化模型上右键选择预览，可以看到数据已经被归一化。

归一化模型算子预览					全屏切换 X
归一化模型	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
	0.646	0.795	0.203	0.080	setosa
	0.620	0.682	0.203	0.080	setosa
	0.595	0.727	0.188	0.080	setosa
	0.582	0.705	0.217	0.080	setosa
	0.633	0.818	0.203	0.080	setosa
	0.684	0.886	0.246	0.160	setosa
	0.582	0.773	0.203	0.120	setosa
	0.633	0.773	0.217	0.080	setosa
	0.557	0.659	0.203	0.080	setosa
	0.620	0.705	0.217	0.040	setosa

第六步，从列处理算子里拖拽类型算子到画布，并连接归一化模型算子与类型算子。双击打开类型算子，将 Species 列的模型类型设置为 Label:

TypeNode ?



算子ID: HEWNLc6TRY

算子名称: 类型

☒ 算子自动命名

字段	存储类型	测量类型	模型类型
Sepal_Length	Double	Continuous ▾	Feature ▾
Sepal_Width	Double	Continuous ▾	Feature ▾
Petal_Length	Double	Continuous ▾	Feature ▾
Petal_Width	Double	Continuous ▾	Feature ▾
Species	String	Categorical ▾	Label ▾

共 5 条

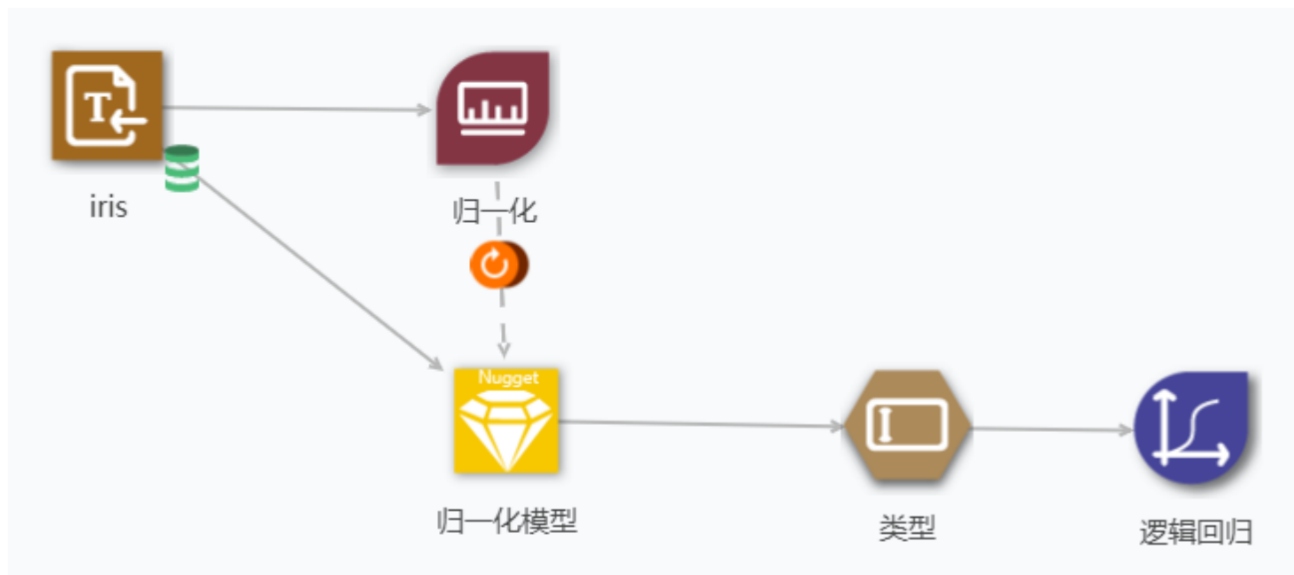


1

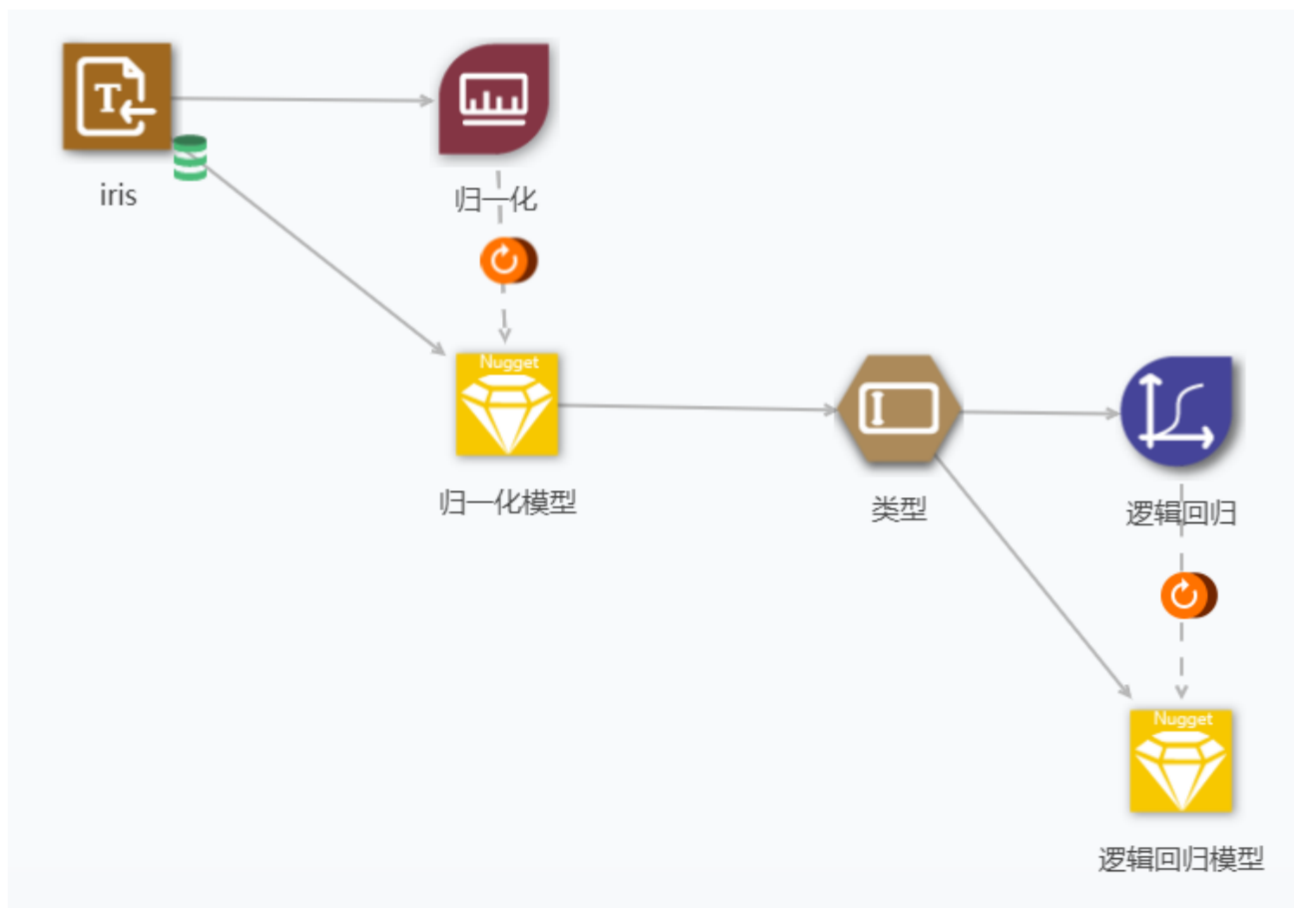


10 条/页 ▾

第七步，从机器学习/分类算子里拖拽逻辑回归算子到画布，连接类型算子到逻辑回归算子。



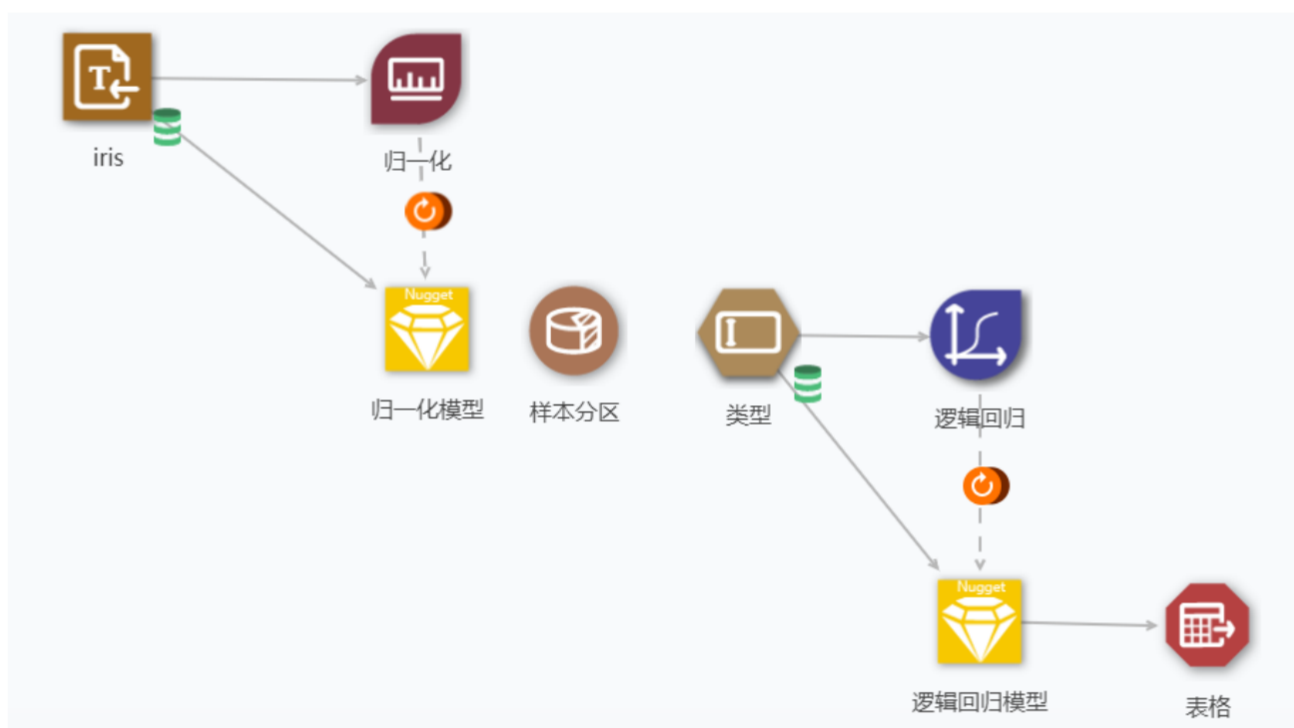
第八步，鼠标在逻辑回归算子上单击右键，运行。得到逻辑回归模型。



第九步，在逻辑回归模型算子后连接一个表格算子，并运行表格算子。得到如下预测结果：

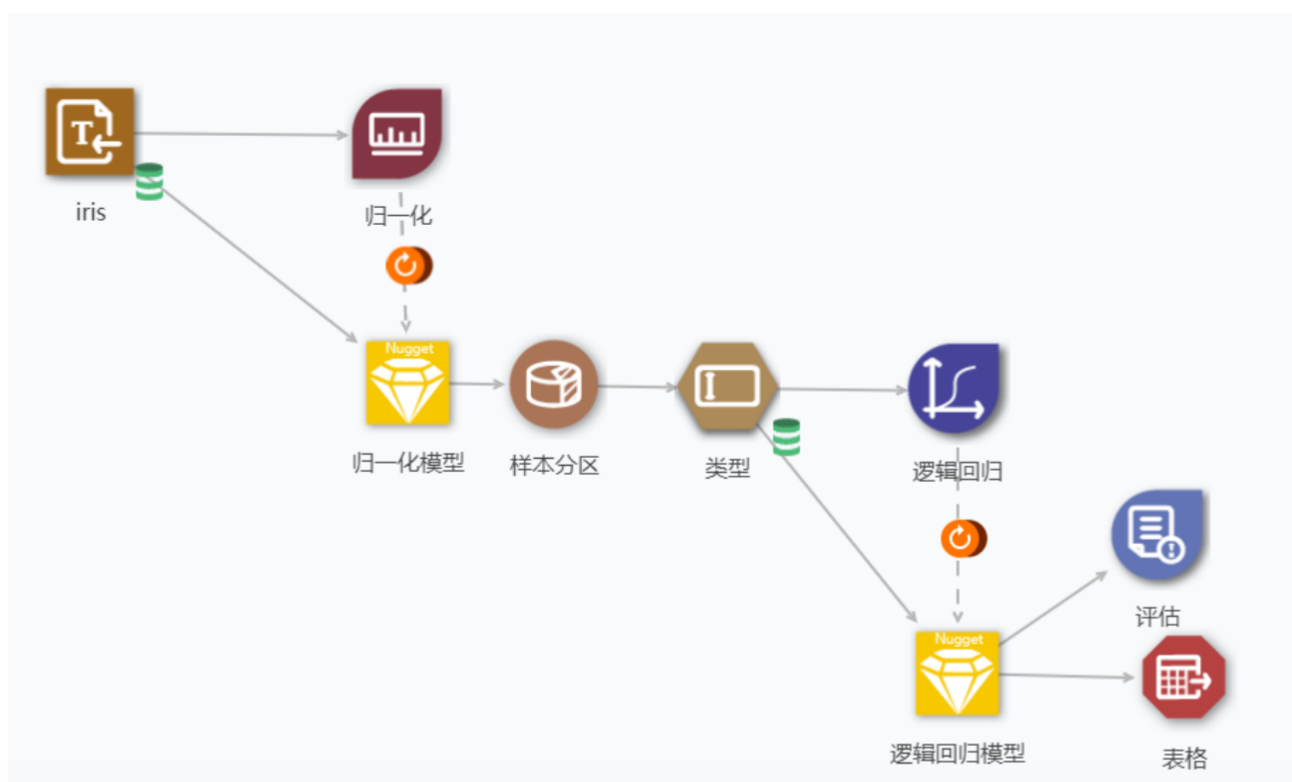
新增算子流4									
表格	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species	virginica_Probability	versicolor_Probability	setosa_Probability	Predicted_Species
日志	0.646	0.795	0.203	0.080	setosa	-66.530	-4.795	71.325	setosa
	0.620	0.682	0.203	0.080	setosa	-55.767	2.134	53.634	setosa
	0.595	0.727	0.188	0.080	setosa	-62.033	-2.346	64.379	setosa
	0.582	0.705	0.217	0.080	setosa	-57.999	-1.112	59.111	setosa
	0.633	0.818	0.203	0.080	setosa	-69.143	-6.987	76.129	setosa
	0.684	0.886	0.246	0.160	setosa	-67.172	-8.511	75.683	setosa
	0.582	0.773	0.203	0.120	setosa	-63.696	-5.691	69.388	setosa
	0.633	0.773	0.217	0.080	setosa	-63.535	-3.658	67.193	setosa
	0.557	0.659	0.203	0.080	setosa	-55.128	0.872	54.256	setosa
	0.620	0.705	0.217	0.040	setosa	-59.206	0.249	58.957	setosa

第十步，将数据集划分为训练集和测试集。将鼠标移动到归一化模型算子与类型算子之间的连线上。并点击断开它们之间的连接。并从行处理算子中拖拽一个样本分区算子到画布上。



第十一步，连接归一化模型算子->样本分区算子->类型算子。

第十二步，从机器学习/模型评估算子里拖拽一个评估算子，并连接在逻辑回归模型算子后。



第十三步，重新运行整个算子流。我们可以看到两个评估结果，分别为训练集评估结果和训练集评估结果。

index	评估算子: 训练集评估结果
accuracy	1
weightedPrecision	1
f1	1
weightedRecall	1

index	评估算子: 测试集评估结果
accuracy	0.875
weightedPrecision	0.920
f1	0.883
weightedRecall	0.875

7 用知识图谱进行推测

注意：知识图谱中，不支持中文设置关系名，属性名，实体名称，且不能设置相同的名称

在本例中，我们将通过个人的身份信息和夫妻子女关系信息，构建一张关系图，来推测出爷孙，舅舅和外甥，姑姑和侄子关系

第一步，拖拽一个文本算子，上传测试数据里的“information.txt”，并设置“information.txt”为文件路

径。并在文本算子上进行预览

information算子预览							全屏切换 X
information	姓名	年龄	性别	身份证号码	出生年月	地址	
	周府海	32	女	110102198712042200	1987-12-04 00:00:00.0	北京市朝阳区	
	周地华	106	男	362427191305057300	1913-05-05 00:00:00.0	陕西省西安市莲湖区	
	赵越三	19	男	362429200002087100	2000-02-08 00:00:00.0	陕西省西安市长安区	
	郑控蜜	69	女	362421195006108350	1950-06-10 00:00:00.0	陕西省西安市碑林区	
	蒋章荆	118	女	362432190102029600	1901-02-02 00:00:00.0	江苏省南京市玄武区	
	赵杰星	6	女	362428201310202100	2013-10-20 00:00:00.0	陕西省西安市高陵区	
	蒋都故	59	男	362424196001159200	1960-01-15 00:00:00.0	江苏省南京市玄武区	
	孙列衡	90	女	110105192904141810	1929-04-14 00:00:00.0	浙江省杭州市拱墅区	
	李庐列	111	女	110104190810245120	1908-10-24 00:00:00.0	北京市房山区	
	杨华都	117	女	362427190212024400	1902-12-02 00:00:00.0	陕西省西安市莲湖区	

第二步，分析数据，通过观察数据，发现可以将每一行数据抽象成一个人，姓名，年龄，性别，身份证号码，出生年月，地址，都是这个人的属性

第三步，拖拽一个图生成算子，连接文本算子和图生成算子



第四步，双击打开图生成算子，首先点击设置图属性按钮，进去后，填写一个图名称，后点击保存按钮

实体模型构建

设置图属性

关系 实体 映射 规则

关系名

设置图属性

*图名:
people_relationship 导入图数据

第五步，增加实体和实体属性，左上角选择实体选项，点击+新实体，按照填写实体名称和实体属性，填写完成后，点击添加按钮



编辑实体

实体名: people

实体父类: 请选择

属性编辑

涉及角色

属性名称

name

age

gender

id_card

birthday

address

+

注意这些都不要使用中文名称

第六步，添加实体映射，点击映射按钮，再点击添加实体，在实体的下拉列表中，选择新增的实体名称，然后在点击右侧添加属性按钮，有几个属性，就点击几次按钮，按照属性名称和实际数据中的对应关系填写，选中唯一能标识数据属性，设置为 **key**，填写完成后，点击提交按钮

实体模型构建

设置图属性

关系 实体 映射 规则

实体映射 关系映射

映射规则

添加实体

映射规则	实体	操作
rule_0	people	删除

属性列名映射

添加属性

属性	列名	操作
name	姓名	删除 Key
age	年龄	删除 Key
gender	性别	删除 Key
id_card	身份证号码	删除 Key
birthday	出生年月	删除 Key
address	地址	删除 Key

第七步，点击运行按钮，将 information 数据插入到图中

第八步，新建一个算子流，拖拽一个文本算子，上传测试数据里的“relationship.txt”，并设置“relationship.txt”为文件路径。并在文本算子上进行预览

relationship算子预览						全屏切换 X
relationship	丈夫	妻子	儿子	女儿	身份证号码	
	蒋杰磊	周府湖	周地华	郑控蜜	362402198109130940	
	周地华	孙耀郡	蒋都放	孙列衡	362427191305057300	
	李庐列	郑控蜜	钱荆湖	杨华都	110104190810245120	
	卫露华	褚翼分	李庐列	沈三江	110109195404105420	

第九步，分析数据，relationship 数据不能再抽象出新的实体，所有数据都是之前创建的 people 实体的属性，每一行数据中，有夫妻关系，父子关系，父女关系，母子关系，母女关系，姐弟关系

第十步，拖拽一个图生成算子，连接文本算子和图生成算子，打开图生成算子，点击设置图属性按钮，图名选择之前创建的 people_relationship，点击右侧的导入图数据，即可在实体选项下面看到之前插入的 information 数据中建的 people 实体和属性，点击保存按钮。

设置图属性

×

*图名:

people_relationship

导入图数据

关系	实体	映射	规则	+ 新实体
实体名	父类名	属性	角色	操作
people		age, address, name, birthday, gender, id_card		编辑 删除

第十一步，创建关系和角色，点击关系选项，在右侧点击+新关系按钮，由于现有的 relationship 数据中，每一行存在五个关系，分别是夫妻关系，父子关系，父女关系，母子关系，母女关系，姐弟关系，所以要新增加五条关系，由于数据中，不存在关系上的属性，所以不需要在属性中添加任何信息，只需要给每个关系分配相应的角色就可以

关系

实体

映射

规则

关系名

型

操作

+ 新关系

关系定义

×

关系名:

husband_and_wife

关系父类:

请选择

属性编辑

涉及角色

角色名称

husband

wife

+

关系定义

×

关系名:

father_and_son

关系父类:

请选择

属性编辑

涉及角色

角色名称

father

son

+

编辑关系

×

关系名: father_and_daughter

关系父类: 请选择

属性编辑 涉及角色

角色名称

father

daughter

+

关系定义

×

关系名: mother_and_son

关系父类: 请选择

属性编辑 涉及角色

角色名称

mother

son

+

关系定义

×

关系名: mother_and_daughter

关系父类: 请选择

属性编辑 涉及角色

角色名称

mother

daughter

+

关系定义

×

关系名: brother_and_sister

关系父类: 请选择

属性编辑 涉及角色

角色名称

brother

sister

+

第十二步，给实体分配关系的角色，经分析，由于我们抽象出来的实体是一个 **people**，一个 **people** 是可以担任任何角色的，包括 **husband**，**wife**，**father**，**mother**，**son**，**daughter**，**brother**，**sister**，所以要将这些角色全部分配给 **people** 实体。选中实体 **people** 的编辑按钮，点击涉及角色，将所有角色全部分配给 **people**，点击更新按钮

+ 新实体

角色	操作
	编辑 删除

编辑实体

×

实体名:

people

实体父类:

请选择

属性编辑

涉及角色

选择角色

<

>

husb...

wife

father

son

daug...

mother

brother

sister

已选择:

husband x

wife x

father x

son x

daughter x

mother x

brother x

sister x

映射	规则		
实体名	父类名	属性	角色
people		address, name, gender, birthday, id_card, age	husband, wife, father, son, daughter, mother, brother, sister

第十三步，给关系添加实体映射，点击映射按钮，选中实体映射，添加实体，实体选中 **people**，再点击右侧的添加属性按钮，因为每一行数据除了丈夫有额外的 **id_card** 属性外，其余都是 **people** 实体的 **name** 属性，所以丈夫添加两个属性 **name** 和 **id_card**，其余实体添加一个属性 **name**，由于共有 4 个实体，所以相同步骤，执行四次

实体映射

关系映射

映射规则

添加实体

属性列名映射

添加属性

映射规则	实体	操作	属性	列名	操作
rule_0	people	删除	name	丈夫	删除 Key
			id_card	身份证号码	删除 Key

实体映射

关系映射

映射规则

添加实体

属性列名映射

添加属性

映射规则	实体	操作	属性	列名	操作
rule_0	people	删除	name	妻子	删除 Key
rule_1	people	删除			

实体映射

关系映射

映射规则

添加实体

属性列名映射

添加属性

映射规则	实体	操作	属性	列名	操作
rule_0	people	删除	name	儿子	删除 Key
rule_1	people	删除			
rule_2	people	删除			

实体映射

关系映射

映射规则

添加实体

属性列名映射

添加属性

映射规则	实体	操作	属性	列名	操作
rule_0	people	删除	name	女儿	删除 Key
rule_1	people	删除			
rule_2	people	删除			
rule_3	people	删除			

第十四步，给关系添加关系映射，选择映射下的关系映射，点击添加关系，选择之前创建的一个关系，再点击添加角色，由于一个关系有两个角色，所以需要点击两下添加角色，然后分别选中不同的角色和映射规则，关于映射规则的选择，可以点击到刚才添加的实体映射，查看对应实体的映射规则是哪一个，就在关系映射的角色添加中的映射规则选择哪一个，映射规则可以在实体映射中修改，本次我使用的自动生成的规则。由于关系中没有属性，所以可以不添加关系中的属性。有六个关系，相同步骤，执行六次，点击提交按钮

实体映射

关系映射

映射规则	实体	操作
rule_0	people → 丈夫	删除
rule_1	people → 妻子	删除
rule_2	people → 儿子	删除
rule_3	people → 女儿	删除

实体映射 关系映射				
关系 添加关系		关联实体角色 添加角色		
关系	操作	角色	映射规则	操作
husband_and_wife	删除	husband	rule_0	删除
		wife	rule_1	删除

实体映射 关系映射				
关系 添加关系		关联实体角色 添加角色		
关系	操作	角色	映射规则	操作
husband_and_wife	删除	father	rule_0	删除
father_and_son	删除	son	rule_2	删除

实体映射 关系映射				
关系 添加关系		关联实体角色 添加角色		
关系	操作	角色	映射规则	操作
husband_and_wife	删除	father	rule_0	删除
father_and_son	删除	daughter	rule_3	删除
father_and_daughter	删除			

实体映射 关系映射				
关系 添加关系		关联实体角色 添加角色		
关系	操作	角色	映射规则	操作
husband_and_wife	删除	mother	rule_1	删除
father_and_son	删除	son	rule_2	删除
father_and_daughter	删除			
mother_and_son	删除			

实体映射

关系映射

关系		添加关系	关联实体角色			添加角色
关系	操作		角色	映射规则	操作	
husband_and_wife ▾	删除		mother ▾	rule_1 ▾	删除	
father_and_son ▾	删除		daughter ▾	rule_3 ▾	删除	
father_and_daughter ▾	删除					
mother_and_son ▾	删除					
mother_and_daughter ▾	删除					

实体映射

关系映射

关系		添加关系	关联实体角色			添加角色
关系	操作		角色	映射规则	操作	
husband_and_wife ▾	删除		brother ▾	rule_2 ▾	删除	
father_and_son ▾	删除		sister ▾	rule_3 ▾	删除	
father_and_daughter ▾	删除					
mother_and_son ▾	删除					
mother_and_daughter ▾	删除					
brother_and_sister ▾	删除					

第十五步，点击运行按钮，将 **relationship** 数据插入到图中

第十六步，查看两个数据是否正常插入到图中，拖拽一个图可视化算子，双击打开，在输入框中选择刚才新建的图名，点击构建查询，在关系类型中随意选择一种关系，点击提交按钮，再点击补全按钮，再点击提交按钮，正常应该查询出满足符合条件的图谱

图查询语句构建

语法构造器

提交

清空

当前无语法帮助说明

谓词

isa

has

plays

relates

contains

==

!=

>

>=

<

<=

占位符

+ 新占位符

属性选择

辅助构造

开

实体:

实体类型

关系:

关系类型

属性:

属性类型

角色:

角色类型

输入语句

收藏夹

补全

清空

语句连接

and

or

(

)

提交

重置

取消

图查询语句构建

语法构造器

\$a

(mother:

\$b,

daughter:

\$c)

isa

mother_and_daughter

:

提交

清空

当前无语法帮助说明

输入语句

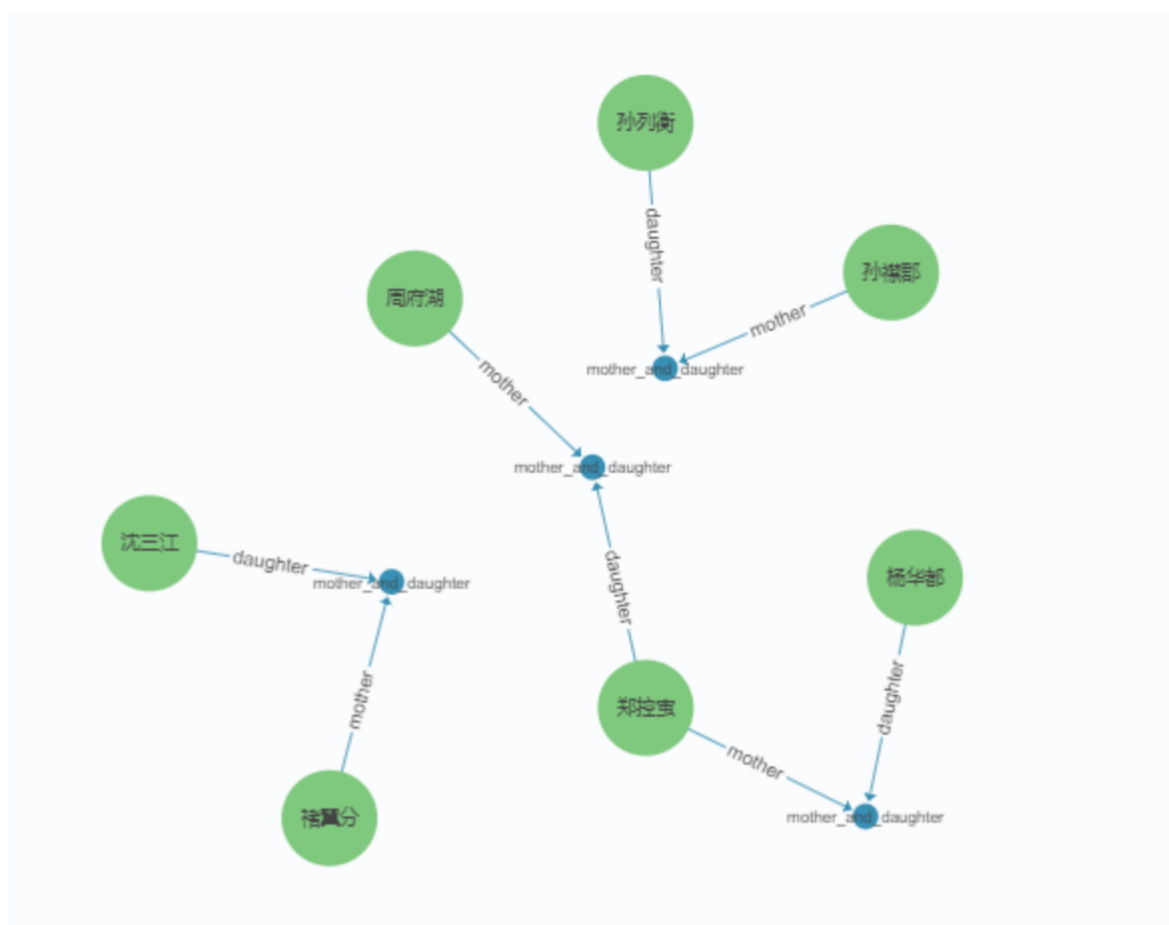
收藏夹

补全

清空

```
match $a (mother: $b, daughter: $c) isa mother_and_daughter ;get $a;limit 30;
```

语句连接



第十七步，添加推测关系，本例中，我们需要推测出爷孙，舅舅和外甥，姑姑和侄子等关系，新建一个算子流，拖一个文本和图生成算子连接，文本算子随意选择一个数据即可，因为是推测关系，肯定是不存在这样的数据的，我们只是将关系插入图中，双击打开图生成算子，设置图属性，选择 `people_relationship`，导入图数据，点击保存。点击新增关系，创建三个推测的关系。

编辑关系

关系名:

关系父类:

角色名称

-

-

+

关系定义

关系名:

关系父类:

角色名称

-

-

+

编辑关系

×

关系名:

关系父类:

请选择

属性编辑

涉及角色

角色名称

—

—

+

在实体中把新增的角色，赋给 **people** 实体，点击更新按钮，然后在点击提交按钮，运行算子流，将新增的三个关系添加到图中

编辑实体

×

实体名:

实体父类:

请选择

属性编辑

涉及角色

选择角色

<

>

wife

husb...

mother

son

brother

sister

father

daug...

gran...

gran...

uncle

neph...

aunt

已选择:

新增的角色

mother ×

son ×

wife ×

father ×

husband ×

brother ×

daughter ×

sister ×

grandpa ×

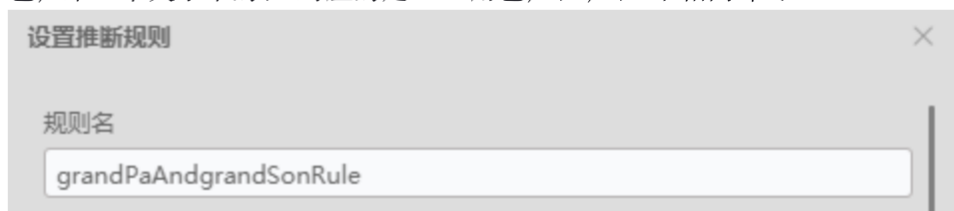
grandson ×

uncle ×

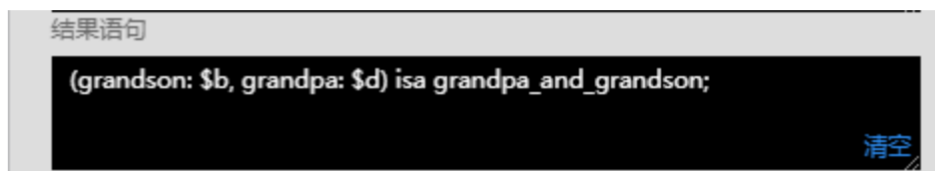
nephew ×

aunt ×

第十八步，构建推测规则，拖一个图可视化算子，双击打开，选中图名，点击设置推断规则，填写规则名，填写条件语句和结果语句。条件语句和结果语句都可以借助系统自带的辅助功能方便添加，例如，我想在条件语句中，填写两个父子关系，只需要点击关系类型中的 **father_and_son**，会自动把语句提交到语法构造器的输入窗口，在点击语法构建器下面的提交的条件语句，因为要有两个父子关系，所以点击两下提交到条件语句，之后要修改参数，因为推测关系中要有实体对应，例如第一个关系中的 **\$c** 对应的是 **father** 角色，第二个关系中的 **\$c** 对应的是 **son** 角色，**\$b**，**\$d** 不相同即可



点击语法构造器中的清空按钮，再次选择关系类型中的爷孙关系，在语法构造器中，提交到结果语句，将爷孙关系的实体角色改成之前设置的**\$b**，**\$d**，对应起来，然后点击提交



相同步骤，在把舅舅和外甥，姑姑和侄子的推测规则添加到图中

设置推断规则

×

规则名

uncleAndnephewRule

输入语句

规则列表 ?

条件语句

(brother: \$b, sister: \$c) isa brother_and_sister ;(wife: \$c, husband: \$d) isa husband_and_wife ;(mother: \$c, son: \$e) isa mother_and_son ;

清空

结果语句

(uncle: \$b, nephew: \$e) isa uncle_and_nephew ;

清空

设置推断规则

×

规则名

auntAndnephewRule

条件语句

(brother: \$b, sister: \$c) isa brother_and_sister ;(wife: \$c, husband: \$d) isa husband_and_wife ;(mother: \$c, son: \$e) isa mother_and_son ;

清空

结果语句

(aunt: \$c, nephew: \$e) isa aunt_and_nephew ;

清空

第十九步，查询推测关系，双击图可视化算子，选择图名，选择是否开启推断为是，再点击构建查询按钮，关系类型中分别选择爷孙关系，舅舅和外甥关系，姑姑和侄子关系，然后提交到输入语句，点击补全按钮，再次点击提交按钮，知识图谱会将推测出来的关系展示出来

