2.2 线性回归

这一步新用到的PAI组件有：

“机器学习”--“回归”--“线性回归”：可以建立线性回归模型；

“机器学习”--“预测”：根据机器学习模型与输入的数据进行预测；

“机器学习”--“评估”--“回归模型评估”：对回归模型进行评估，包含可视化回归结果与R^2、RMSE等指标；

**1.对销量进行线性回归**

从工作区左侧将：

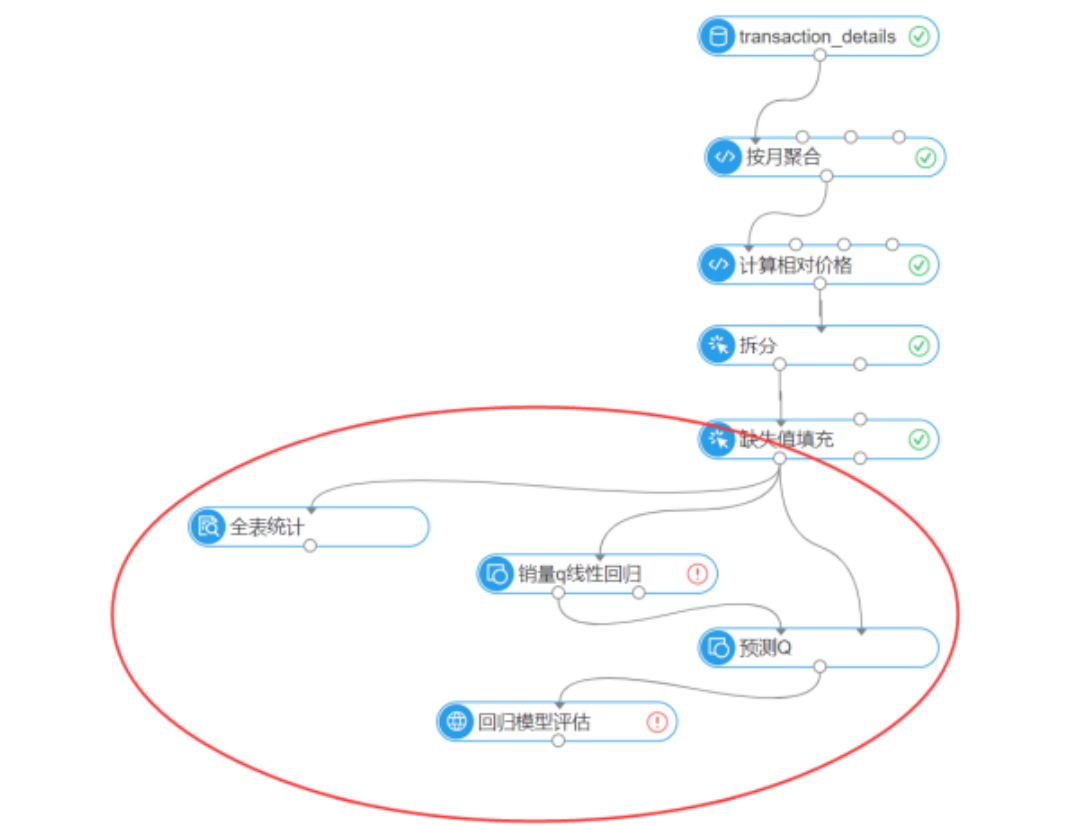
“机器学习”--“回归”--“线性回归”（重命名为“销量q线性回归”）；

“机器学习”--“预测”（重命名为“预测Q”）；

“机器学习”--“评估”--“回归模型评估”；

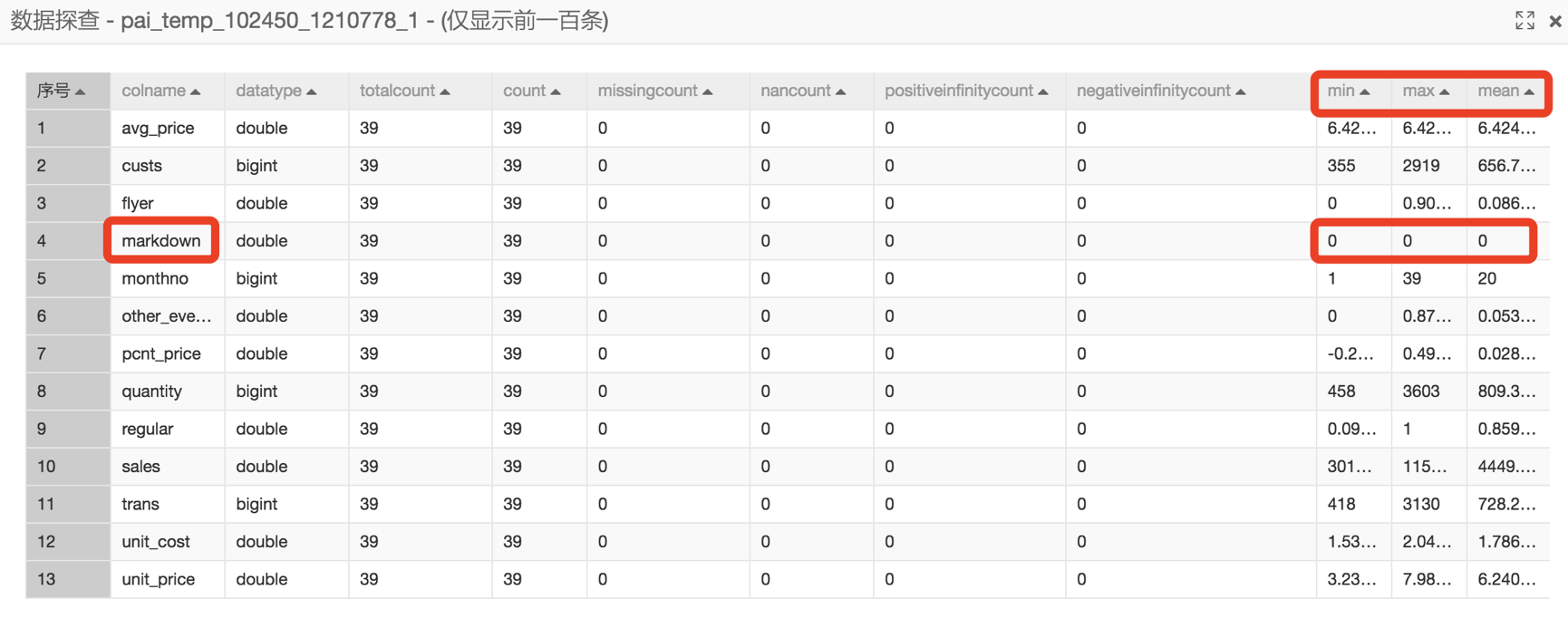
“统计分析”--“全表统计”。

拖至工作区按如下方式连接：



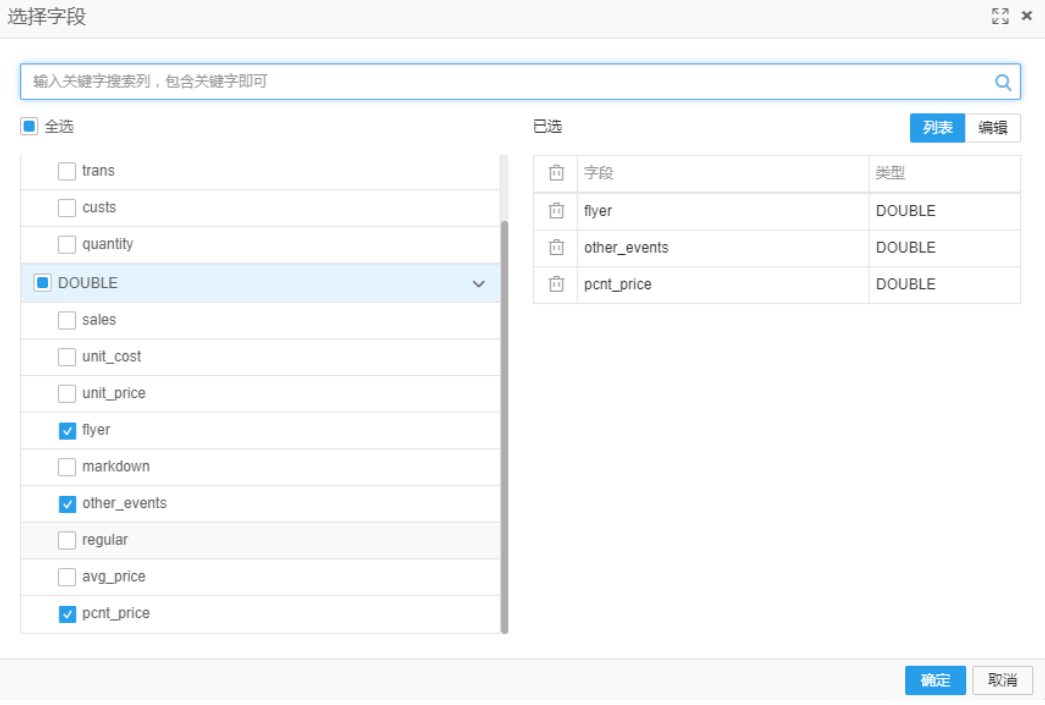
|  |  |  |  |
| --- | --- | --- | --- |
| **组件类型** | **重命名** | **输入端口** | **上游端口** |
| 全表统计 |  | 1 | 缺失值填充 |
| 线性回归 | 销量q线性回归 | 1 | 缺失值填充 |
| 预测 | 预测Q | 1 | 销量q线性回归 |
| 2 | 缺失值填充 |
| 回归模型评估 |  | 1 | 预测Q |

右键单击“全表统计”，“字段设置”可以不用修改，默认为全选。左键单击“执行该节点”，待其运行结束，再右键单击“全表统计”，查看数据：



可以发现，markdown的最小值（min），最大值（max）与均值（mean）全部为0，也就是该产品的销售记录中没有markdown类型，所以不需要再将其加入回归模型中。

左键单击“销量q线性回归”在“字段设置”--“选择特征列”选择如下几个字段（由于销售类型所占比例之和为1，则在没有markdown的情况下，regular=1-flyer-other\_events，为了防止出现多重共线性，只需要从这三种销售类型中取两个加入回归模型即可。一般来说，我们分析的目的是找到哪种促销手段对销量的提升帮助最大，所以可以不将regular加入回归模型）：



在“选择标签列”选择‘quantity’

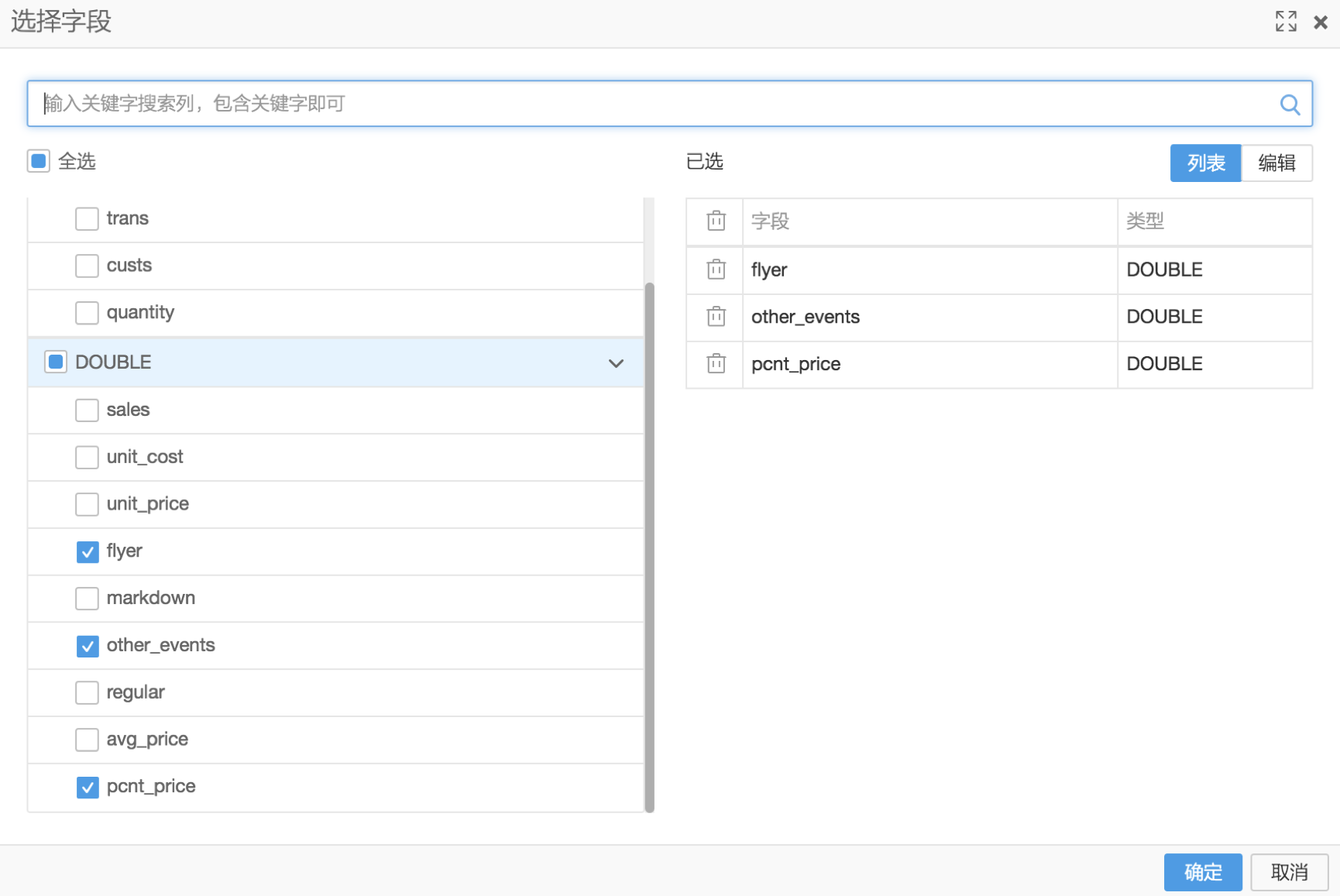


在“参数设置”里面勾选“生成模型评估表”与“回归系数评估”

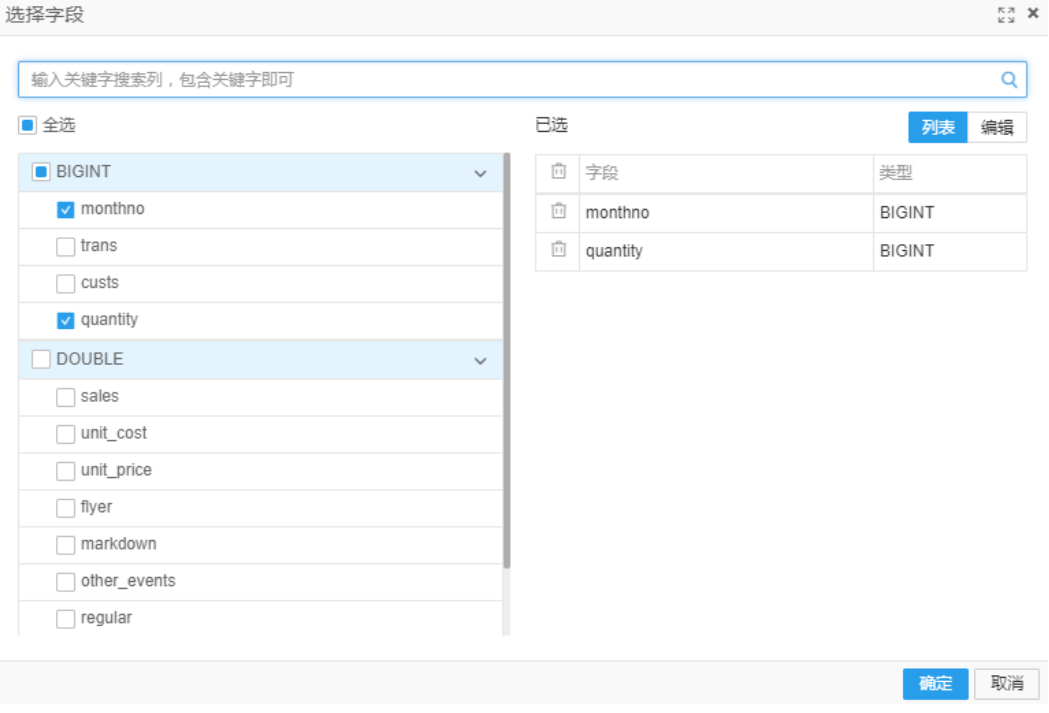


线性回归组件是构建一个特征列为自变量，标签列为因变量的线性回归模型。

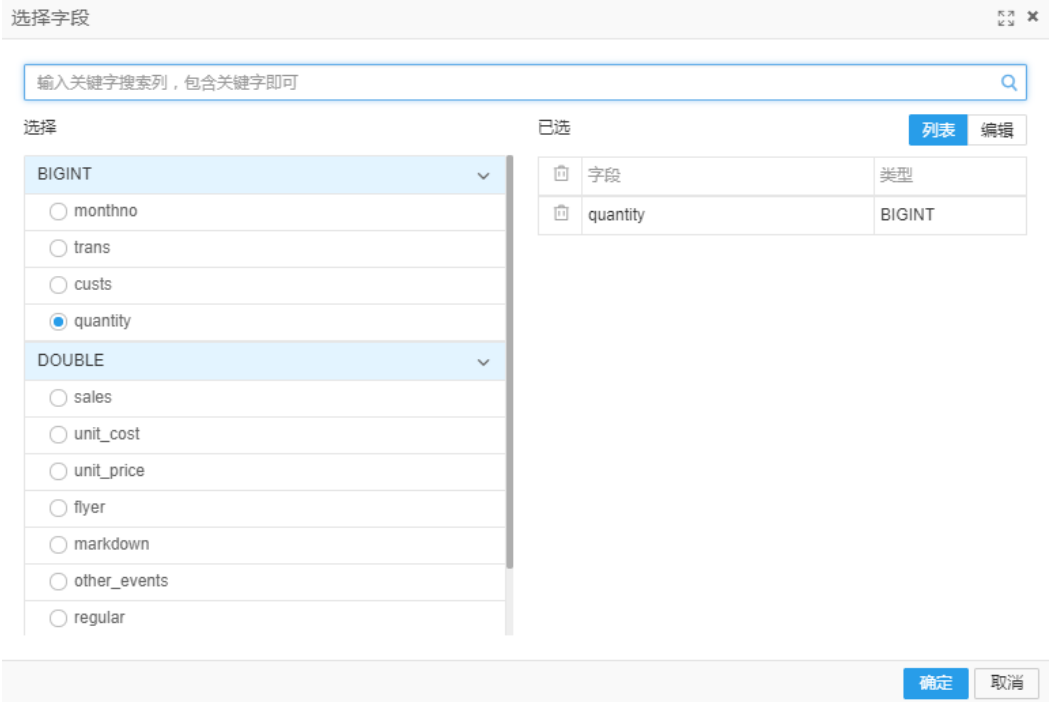
左键单击“预测Q”，在右侧“选择字段”选择：”flyer”,”oyher\_event”,”pcnt\_price”这三个字段。



在“原样输出列”选择‘monthno’与‘quantity’



左键单击“回归模型评估”，原回归值选择‘quantity’：



右键单击“销量q线性回归”，左键单击“从此处开始执行”。

再右键单击“全表统计”，查看数据：

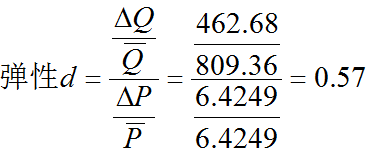


可以看出，产品的均价约为0.0286，月均销量809.3590。

右键单击“销量q线性回归”，左键单击“查看分析报告”



其中，pcnt\_price的回归系数为462.68。

从上面两张图可以看出，产品的平均销量（quantity的mean）为809.36，产品的平均价格（avg\_price）为6.42，pcnt\_price的回归系数为462.68，这意味着当pcnt\_price=100%=1 时，比如价格从12.84变动到6.42时，价格的变动除以平均价格就是（12.84-6.42）/ 6.42 = 1。此时，销量的变化是462.68，也就是说价格变动了100%的时候，销量会变动462.68。所以根据公式：，它的弹性为0.57，属于缺乏弹性的产品。它的销量对价格是不太敏感的，可以考虑减少促销甚至提升价格来提高该产品的利润。



如果还要进行促销的话，从促销活动的类型来看，flyer的回归系数明显大于other\_events，所以flyer对销量的提升是最有效的。

**2.计算回归的差值dQ**

使用线性回归方程预测出来的是与时间无关因素对产品销量的影响，本实验假设用产品实际销量减去回归模型预测销量就可以得到随时间波动的产品销量。

从左侧拖出一个“SQL脚本”，将其重命名为“计算dQ”，连接至“预测Q”，并输入如下代码：

select monthno

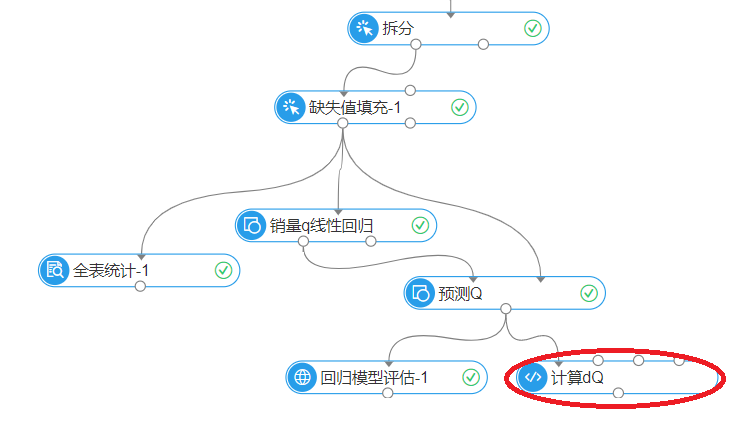
,quantity

,prediction\_result

,(quantity - prediction\_result) as dQ

from ${t1}

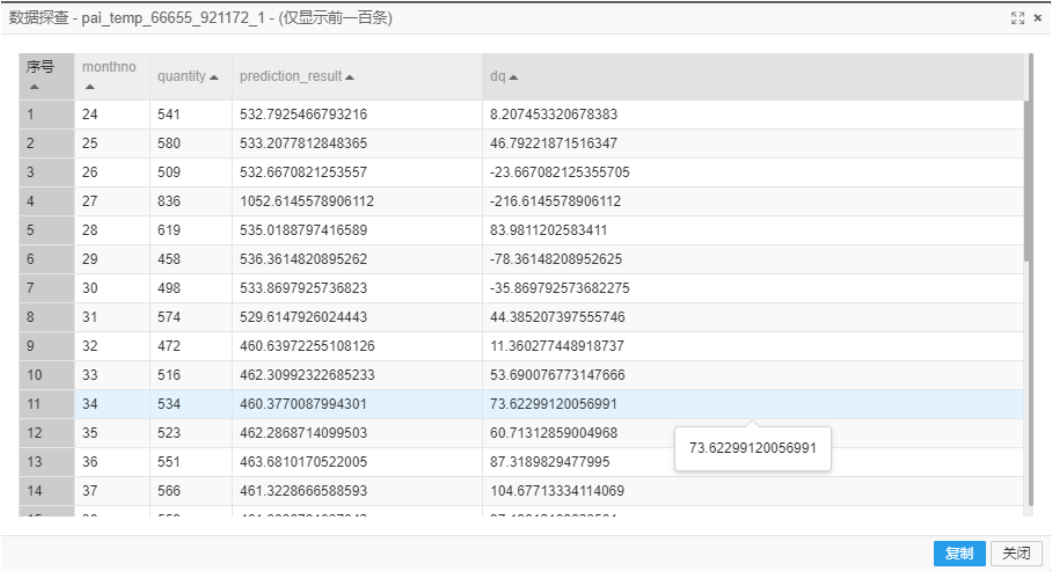
这段代码计算了该产品真实销量q与线性回归模型预测销量Q的差值dQ，本模型假设dQ可用时间序列分解的方法进行预测。



随后右键单击“计算dQ”，左键单击“执行该节点”。

**3.dQ归一化**

右键单击“计算dQ”，左键单击“查看数据”。

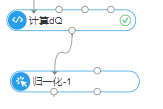


由于dQ是回归模型预测值与真实值的差，实际上是模型的误差项，所以dQ较为均匀地分布在x轴两侧，均值接近于0。

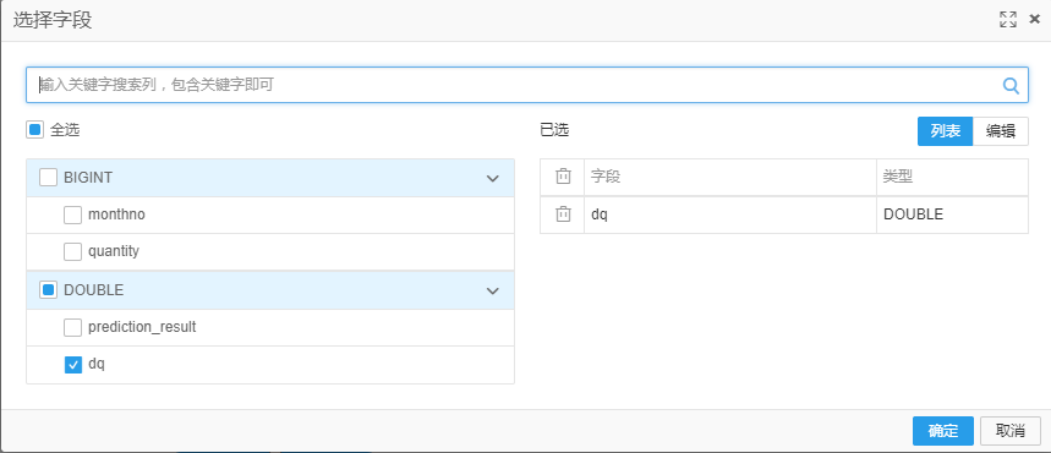
由于在乘法分解模型中dQ的均值需要做分母，可能会放大模型的误差，因此需要对dQ进行归一化。

归一化是一种用来消除量纲的数据预处理方法。通常来说，不同的指标有着不同的量纲，对不同指标进行归一化可以消除他们之间的量纲差异。在本次实验中，归一化的目的不是消除量纲，通过将dQ的值域都映射到[0,1]区间上的归一化方法可以消除dQ可能会放大模型误差的问题。

从左侧拖出“数据预处理”--“归一化”，连接至“计算dQ”。

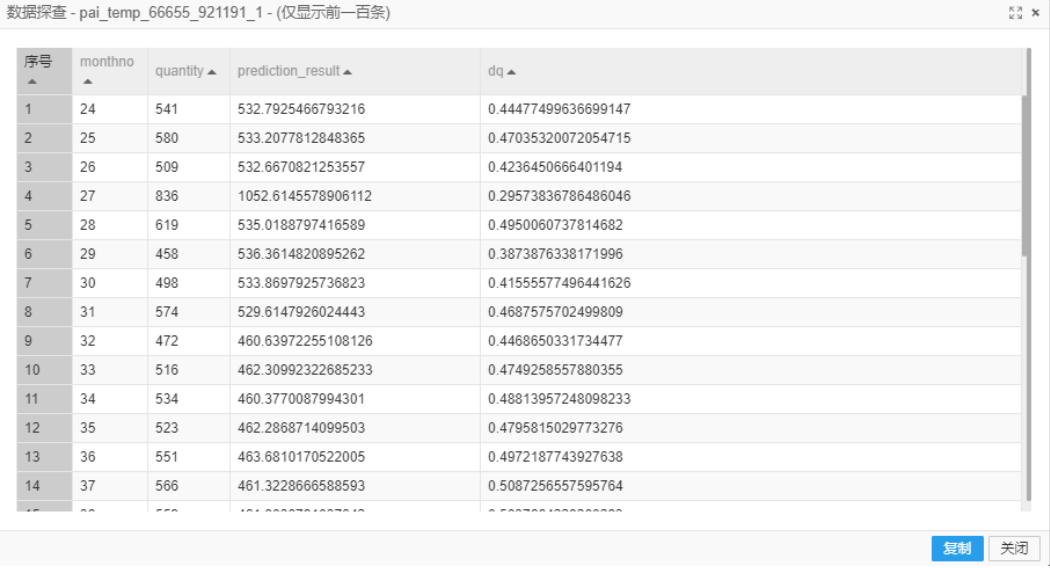


左键单击“归一化”，在右侧“选择字段”选择‘dQ’。



右键单击“归一化”，左键单击“执行该节点”。

待其运行结束，右键单击“归一化”--“查看数据”--“查看输出桩1”



可以看出，归一化之后，dQ已经被投影到了[0,1]的区间上。