1. 实验概述

股票自动交易程序能够提高执行力，减少时间成本，提高工作效率。并且能在一定程度上提高操盘精准度，瞬间下单锁定最佳买卖点。本实验主要通过移动平均线来构建股票交易策略，使用了浦发银行和山东机场两支股票的交易数据来构建模型，验证模型的有效性。

1. 实验目的

1.了解常见的股票交易策略

2.能根据目的选择合适的股票交易策略

3.能够使用MaxCompute建立简单的股票交易模型

1. 实验架构

阿里云MaxCompute

1. 实现步骤
   1. 数据预处理

4.1.1数据来源

目前阿里云大数据产品已经免费向全部用户开放了多种公用数据集。目前开放的数据类别包括：股票价格数据，房产信息，影视及其票房数据。本实验需要使用到股票价格数据。该数据集的数据表结构如下：



4.1.2新建脚本

该数据在阿里云数加平台是以分区表的形式存储的，因此为了方便接下来的实验，将需要使用到的数据切取下来保留在本地个人项目中。

进入MaxCompute项目工作区，点击新建脚本按钮，创建一个新脚本

4.1.3构建SQL脚本

根据阿里云的官方文档，存放该数据的数据表位于public\_data项目中，表名为ods\_enterprise\_share\_trade\_h。

为了做分析，我们需要选取一直股票作为实验样本，在这里选择股票代号为600000的浦发银行，并选取2014年8月22日至2017年8月22日之间的数据进行分析，因此构建下列SQL语句

create table pufa as

select trde\_date,open,high,close,low,volume

from public\_data.ods\_enterprise\_share\_trade\_h

where code='600000'

and trde\_date >= '2014-08-22'

and trde\_date <= '2017-08-22'

and ds='20170824'

order by trde\_date

limit 1000;

将股票的交易日期、开盘价、最高价、收盘价、收盘价与成交量选取出来，并按照交易日期升序排列，将其保存在MaxCompute个人项目中的pufa表里。

点击运行按钮，当显示如下界面时即可点击于运行按钮，执行该语句

等待一分钟左右该语句执行完毕。点击表查询即可看到刚刚新建的表

4.1.4查看数据

构造SQL语句，将数据从表中读取出来进行数据探查。

select \* from pufa

用鼠标选中上述语句，如下。切记一定要选中该段语句才能运行，否则将会出错。

等待片刻运行完毕后，在结果对话框中可以看到这些信息

可以点击页面顶部的按钮保存写好的脚本，以免意外丢失

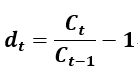
* 1. 对股票数据进行基本分析

4.2.1 机器学习PAI平台

本项目需要使用机器学习PAI平台来进行更详细的处理，首先先进入机器学习平台。在管理控制台界面点击机器学习标签

4.2.2 计算收盘价的每日价格变化率

收盘价的每日变化率ⅆt 用下列公式计算



其中Ct为t日的收盘价

首先点击数据源标签，在搜索框中搜索广告导入的数据表pufa，按下回车搜索出该数据框，将其拖动至主界面

点击组件标签，在上部的搜索栏中搜索sql，将sql脚本这一组件拖动至主界面中。

为了区分方便，右键点击该节点，选择重命名，将其命名为每日价格变化率

用鼠标按照下图所示将线连接两个节点，这就代表了数据流。

点击每日价格变化率标签，在右侧的编辑框中输入下列代码

select a.trde\_date,a.shift1\_close/a.close-1 daily\_pct

from(

select lead(trde\_date,1) over(partition by 1 order by trde\_date) as trde\_date

,close

,lead(close,1) over(partition by 1 order by trde\_date) as shift1\_close

from ${t1}

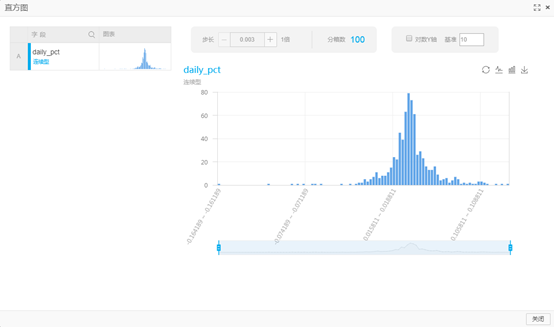
) a

where a.trde\_date is not null;

查看价格变化率的分布及基本的统计描述

左键点击直方图节点，在右侧出现的字段设置中选择字段，将daily\_pct作为绘制直方图的字段

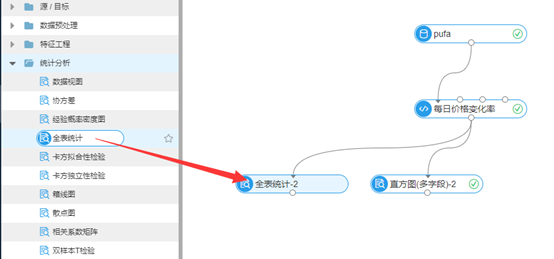
点击确定设置完成后，运行该节点，稍等片刻其运行完毕后，在该节点上点击右键，选择查看分析报告

运行结果如下

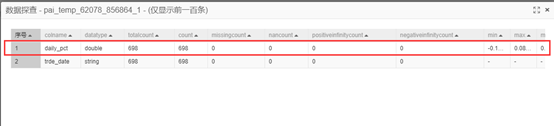
结果说明

通过直方图发现价格每日变化率主要分布在-0.05（-5%）到0.05（5%）之间。

为了能更好的对每日价格变化率进行数据探查，从组件库中拖动一个全表统计节点到主界面上，如图所示连线



连线完成后直接右键运行该节点。运行完成后点击右键选择查看数据

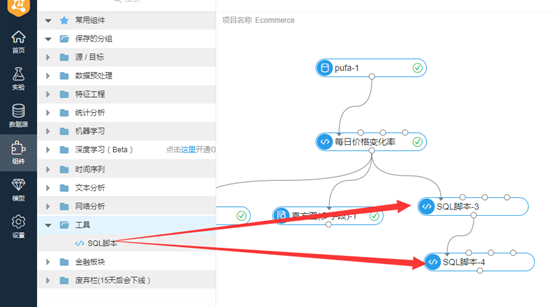


左右滑动可以查看更多的数据项，从数据中可以分析得出，股票每日波动最小值为-0.16，最大值为0.08，均值接近0。可见股票价格每日的波动还是挺大的。

4.1.4计算波动率

由于受到机器学习PAI平台的限制，在这里计算波动率需要使用两个SQL脚本组件才能完成，波动率的计算方式是依次取五个交易日为一组，计算其样本标准差。

从组件库中拖动两个SQL脚本组件，按照如图所示连线，第一个SQL节点负责聚合数据，第二个脚本负责计算每日的价格波动率。为了方便区分，也可将点击右键将该脚本重命名一下。



将刚拖进来的第一个节点重命名为“聚合1”，点击该节点，在右侧将下列代码复制进去

select \* from

( select trde\_date,a.daily\_pct from ${t1} a

union all

select trde\_date,lead(b.daily\_pct,1) over (partition by 1 order by trde\_date) as daily\_pct from ${t1} b

union all

select trde\_date,lead(c.daily\_pct,2) over (partition by 1 order by trde\_date) as daily\_pct from ${t1} c

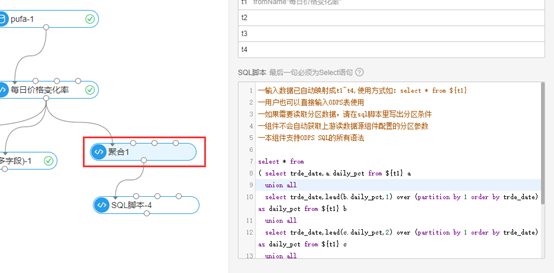
union all

select trde\_date,lead(d.daily\_pct,3) over (partition by 1 order by trde\_date) as daily\_pct from ${t1} d

union all

select trde\_date,lead(e.daily\_pct,4) over (partition by 1 order by trde\_date) as daily\_pct from ${t1} e

) t1;



右键点击该节点运行，同时将刚刚拖进来的另一个SQL脚本节点重命名为“计算波动率”，同样点击该节点在右侧将下列代码复制进去

select trde\_date,stddev\_samp(daily\_pct) vol

from ${t1}

group by trde\_date

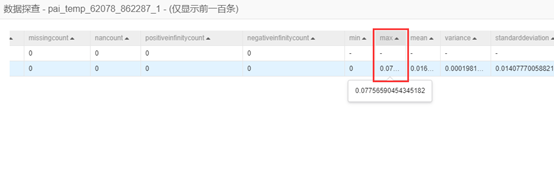


在等待上一个节点运行结束后，右键点击运行该节点，稍等片刻运行完毕

为了更好的分析结果，从组件库中拖动一个全表统计节点至计算波动率节点下方，连线如图。右键点击该节点选择执行该节点。



在运行结束后在该节点点击右键，选择查看数据

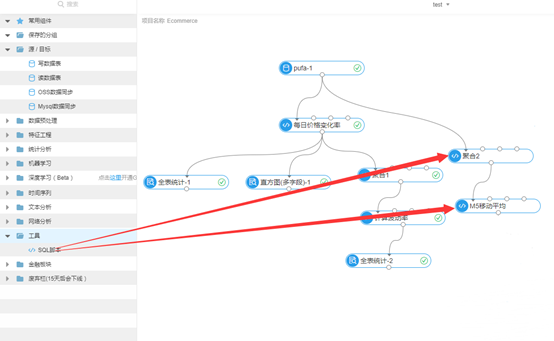
  
通过计算移动标准差得到最大波动率为7.7%，这个波动率还是较大的。

* 1. 构建股票交易策略

在这一步中将使用比较长短期移动平均值的方法来构建股票交易策略，本实验中短期移动平均值为5日的移动平均值，长期移动平均值为20日的移动平均值。

1.     计算短期移动平均值

由于使用机器学习PAI平台计算移动平均值的步骤较为复杂，这里需要使用两个SQL脚本组件完成。因此从组件库中拖动两个SQL组件至主面板，如下



将刚拖进来的第一个SQL组件重命名为“聚合2”，点击该节点，在右侧将下列代码复制进编辑窗口，编辑完成后右键点击该节点，选择执行该节点来执行代码。

select \* from

( select trde\_date,a.close from ${t1} a

union all

select trde\_date,lag(b.close,1) over (partition by 1 order by trde\_date) as close from ${t1} b

union all

select trde\_date,lag(c.close,2) over (partition by 1 order by trde\_date) as close from ${t1} c

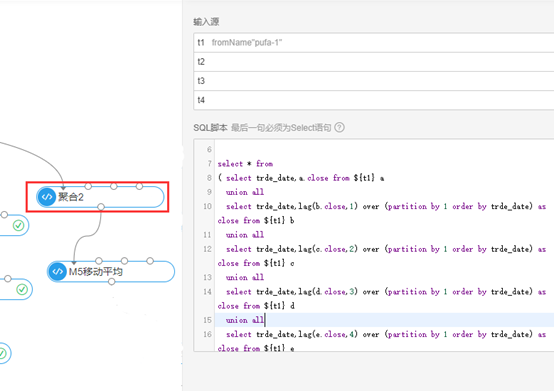
union all

select trde\_date,lag(d.close,3) over (partition by 1 order by trde\_date) as close from ${t1} d

union all

select trde\_date,lag(e.close,4) over (partition by 1 order by trde\_date) as close from ${t1} e

) t1;

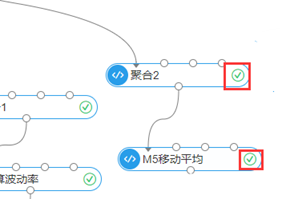


同时可以点击刚拖进来的另一个SQL脚本节点，将其重命名为“M5移动平均”，点击该标签，在右侧编辑栏输入下列代码

select trde\_date,avg(close) M5from ${t1} group by trde\_date

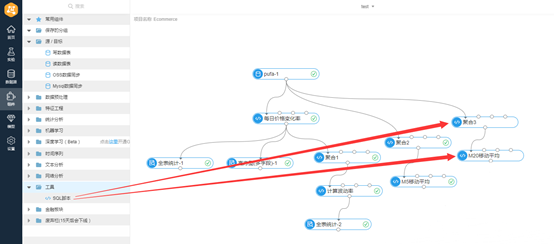


 编辑完成后等待上一节点运行完毕后右键执行该节点，等待这两个节点运行完毕。



2.     计算长期移动平均值

和上一步一样，计算长期移动平均值也需要使用两个SQL脚本组件，因此拖动两个SQL脚本组件至主面板中，连线和上一步类似



将第一个节点更名为聚合3，复制下列代码至其编辑窗口

select \* from

( select trde\_date,a.close from ${t1} a

union all

select trde\_date,lag(b.close,1) over (partition by 1 order by trde\_date) as close from ${t1} b

union all

select trde\_date,lag(c.close,2) over (partition by 1 order by trde\_date) as close from ${t1} c

union all

select trde\_date,lag(d.close,3) over (partition by 1 order by trde\_date) as close from ${t1} d

union all

select trde\_date,lag(e.close,4) over (partition by 1 order by trde\_date) as close from ${t1} e

union all

select trde\_date,lag(f.close,5) over (partition by 1 order by trde\_date) as close from ${t1} f

union all

select trde\_date,lag(g.close,6) over (partition by 1 order by trde\_date) as close from ${t1} g

union all

select trde\_date,lag(h.close,7) over (partition by 1 order by trde\_date) as close from ${t1} h

union all

select trde\_date,lag(i.close,8) over (partition by 1 order by trde\_date) as close from ${t1} i

union all

select trde\_date,lag(j.close,9) over (partition by 1 order by trde\_date) as close from ${t1} j

union all

select trde\_date,lag(k.close,10) over (partition by 1 order by trde\_date) as close from ${t1} k

union all

select trde\_date,lag(l.close,11) over (partition by 1 order by trde\_date) as close from ${t1} l

union all

select trde\_date,lag(m.close,12) over (partition by 1 order by trde\_date) as close from ${t1} m

union all

select trde\_date,lag(n.close,13) over (partition by 1 order by trde\_date) as close from ${t1} n

union all

select trde\_date,lag(o.close,14) over (partition by 1 order by trde\_date) as close from ${t1} o

union all

select trde\_date,lag(p.close,15) over (partition by 1 order by trde\_date) as close from ${t1} p

union all

select trde\_date,lag(q.close,16) over (partition by 1 order by trde\_date) as close from ${t1} q

union all

select trde\_date,lag(r.close,17) over (partition by 1 order by trde\_date) as close from ${t1} r

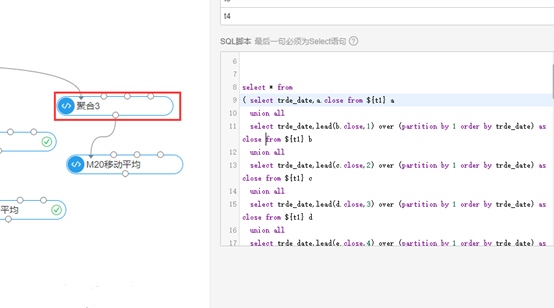
union all

select trde\_date,lag(s.close,18) over (partition by 1 order by trde\_date) as close from ${t1} s

union all

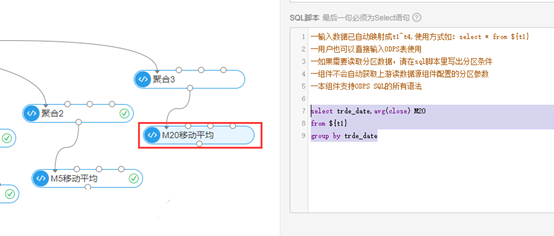
select trde\_date,lag(t.close,19) over (partition by 1 order by trde\_date) as close from ${t1} t

) t1;

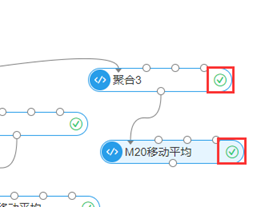


将刚拖进来的第二个节点重命名为“M20移动平均”，在其代码编辑窗口输入下列语句

select trde\_date,avg(close) M20from ${t1}group by trde\_date

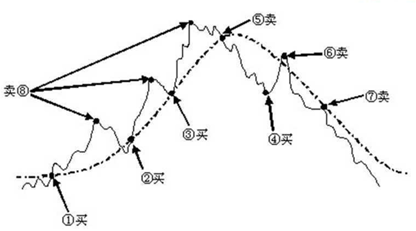


编辑好后依次运行这两个节点，等待其运行成功。



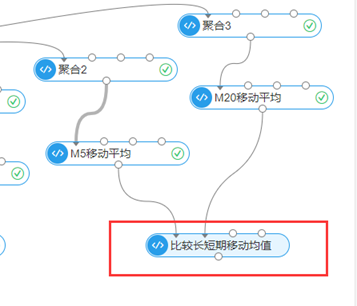
3.     比较长短期移动平均值

根据格兰维尔法则产生交易订单，短期移动平均线向上突破长期移动平均线，为买入信号，短期移动平均线向下突破长期移动平均线为卖出信号。



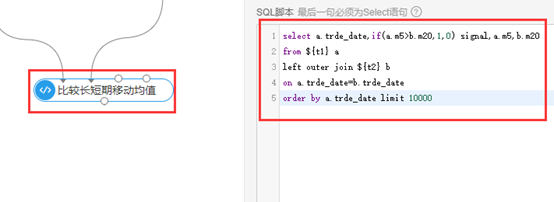
因此需要构建SQL语句，新生成一个signal字段来判断短期移动均值和长期移动均值的大小，当短期移动均值大于长期移动均值，将signal设为1，当短期移动均值小于长期移动均值时，将signal设为0。

拖动一个SQL脚本至主界面，将其重命名为“比较长短期移动均值”。连线如图，将M5和M20作为输入表

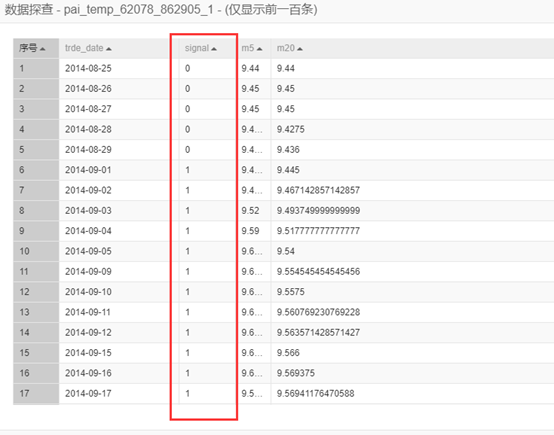


在该节点的编辑窗口复制下列代码

select a.trde\_date,if(a.m5>b.m20,1,0) signal,a.m5,b.m20from ${t1} aleft outer join ${t2} bon a.trde\_date=b.trde\_dateorder by a.trde\_date limit 10000



右键执行该节点，等待节点运行完毕后右键点击该节点查看数据



可以看到在数据表中新生成了一列signal，得到了预想的结果。

4.     产生交易订单

在上一步中生成了signal列，现在需要通过signal列构造交易策略，当signal由0变为1时，是买入信号，当signal由1变为0时是卖出信号，当signal维持0或1不变是则不需要进行交易。单纯的判断signal的数值不好分析买入卖出的时机，因此对signal列进行一次差分，即后一个数值减去前一个。因此当差分后的序列为0时无需发生交易，当差分后的序列为1时表示买入，为-1时表示卖出。

构建SQL语句如下

select a.trde\_date,a.signal,a.shift1\_signal-a.signal difffrom(

select trde\_date

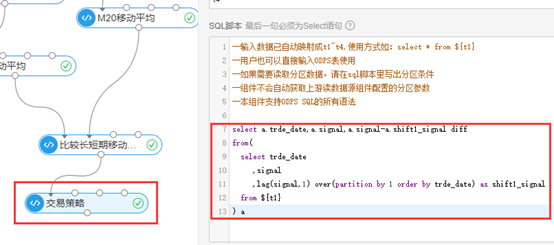
,signal

,lead(signal,1) over(partition by 1 order by trde\_date) as shift1\_signal

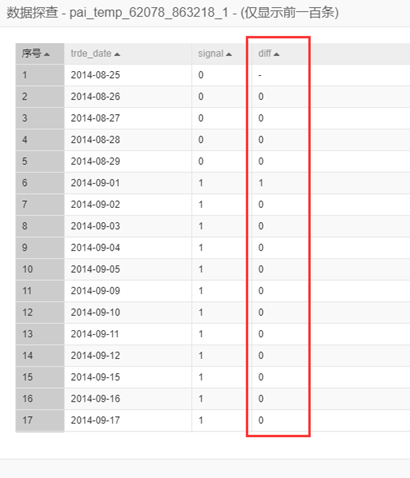
from ${t1}

) a

拖动一个SQL脚本至主界面，重命名为交易策略，连线如图，将上述代码复制进其代码编辑框



在编辑完成后运行该节点，当运行完成后右键点击该节点，查看数据



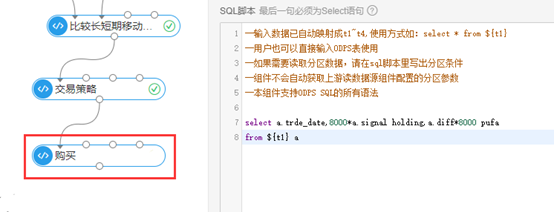
从数据中可以看出，当diff列为0时表示不交易，当diff为1时表示买入，当diff为-1时表示卖出。

* 1. 验证股票模型的有效性

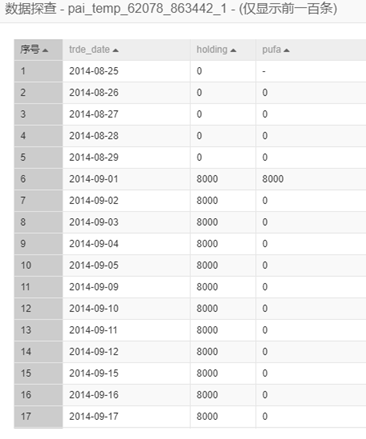
1.     购买股票

设置每次股票购买和卖出都是8000股。从组件库中拖动一个SQL脚本组件至主面板，将其更名为购买，连线如下图，在右侧面板中输入下列代码。由于在上一步中讨论了关于8月27日这天卖出异常的问题，因此在此步骤中将8月26日之前的数据都去掉，因为在有卖出交易之前一定要有一个买入交易。因此语句如下

select a.trde\_date,8000\*a.signal holding,a.diff\*8000 pufafrom ${t1} a



点击右键执行该节点，等待片刻，右键单击该节点，选择查看数据查看运行结果。



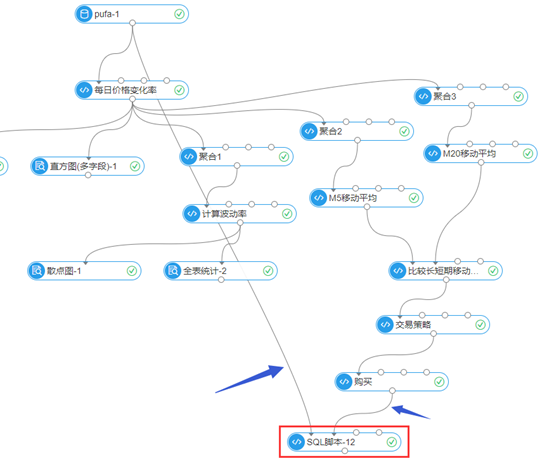
结果说明

使用SQL语句创建一个买入数据集，并且在短期移动平均线大于长期平均线的时候，购入8000股，另外输出的Holding列是指当前持有多少股。Pufa这一列指交易了多少股，8000代表买入8000股，-8000代表卖出8000股。

2.     计算持有金额数和每次的交易金额

【注】持有股票金额数=当期持有股票数\*当期股票收盘价格

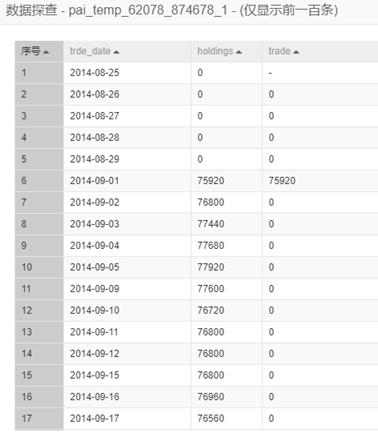
从组件库中拖动一个SQL脚本文件至主窗口，因为此处需要当期股票收盘价格，因此在这里需要将联合原始的数据表，连线如下：



将其重命名为计算持有金额数。根据上述公式，构建下列SQL语句

select b.trde\_date,b.holding\*a.close holdings,b.pufa\*a.close tradefrom ${t2} bleft join ${t1} aon a.trde\_date=b.trde\_date

运行结果如下



将上述语句输入SQL脚本的编辑窗口

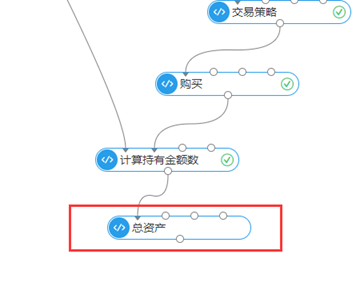
代码说明

* 根据之前构建的交易策略，如果短期移动平均线向上突破长期移动平均线则买入股票，如果短期移动平均线向下突破长期移动平均线则卖出股票。
* Holdings列记录了当前持有股票的金额数
* trade列记录了每次交易产生的资金变动，正数表示买入，负数表述卖出

3.     计算经过交易后的总资产

【注】总资产=初始资金-sum(交易的现金数)

为了计算出最终通过该交易策略获得的总资产，直接对所有者三年中的交易金额求和，用初始资产减去这个交易金额和便得到了在这一期间通过交易的总资产，拖动一个SQL语句组件，将其重命名为总资产，连线如下



在该SQL脚本右侧的编辑栏中输入下列语句，计算总资产

select 100000-sum(trade)from ${t1}

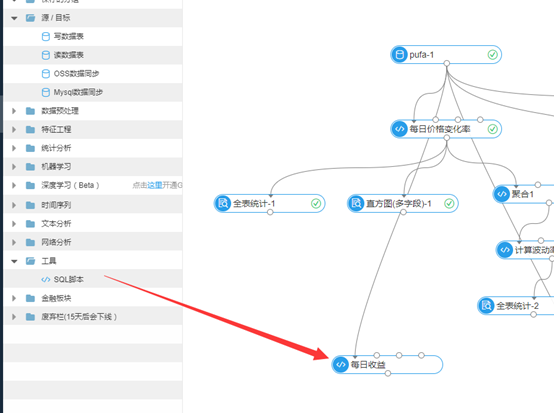
**结果说明**

编辑完成后运行该节点，稍等片刻运行完毕后点击右键可以查看数据。

从运行结果发现剩余总资产为193280，大于初始资金100000。收益为93280元。

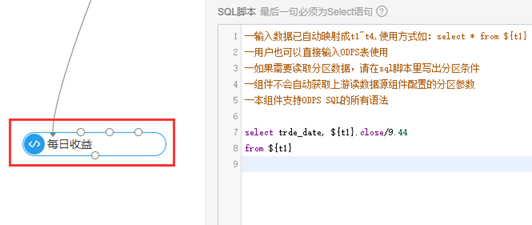
4.     假设交易

假设有这样一个模拟交易，在最初将资金投入，直到最后一天将所有股票卖出，忽略中间所有价格变动，只关心最初的买入日的收盘价与卖出日的收盘价，这样可以通过公式 卖出日的收盘价/买入日的收盘价 来计算每日的收益率。从组件库中拖动一个SQL组件到主面板，将其更名为每日收益

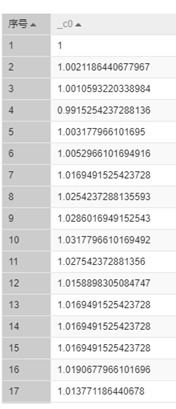


构建SQL代码，将其输入上述的SQL组件的编辑栏中，在最初的数据探查中，可以看到第一天(2014年8月25日)的股票收盘价为9.44元，因此在下列语句中将每日的收盘价close都去除以9.44。得到每日的收益

select trde\_date, ${t1}.close/9.44 from ${t1}



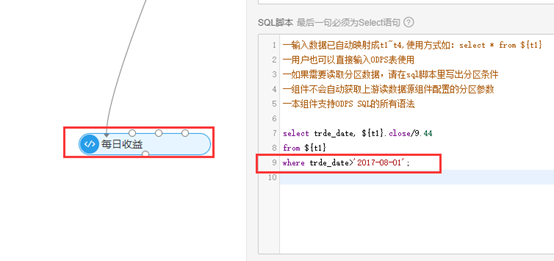
运行该节点，在运行完毕后点击右键查看数据



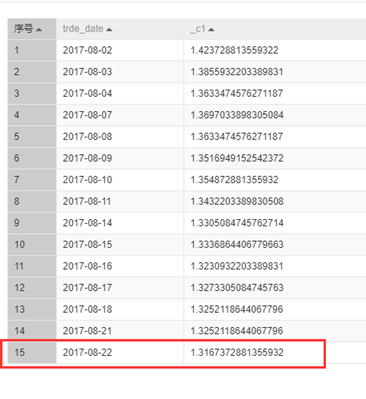
由于我们关注的是最后一天股票交易的数据变化，因此修改上述的SQL语句，对数据进行筛选，选取2017年8月以后的数据。

select trde\_date, ${t1}.close/9.44 from ${t1}where trde\_date>'2017-08-01';

 将代码输入SQL组建的编辑框



再次运行得到结果如下表：



从表中可以看出，如果从第一天开始就买入股票，期间不进行任何交易，直到最后一天，这支股票的总收益为1.31倍，也就是说投入10万元最终能有大致3万的回报。

5.     结论

从模拟交易来看，如果按照上述步骤确定的交易策略，最终的投资结束后的总资产为193280元，而对照买入后不卖出，三年总收益为13.1万元，比使用交易策略的回报少，所以通过移动均线构建的股票交易策略是可靠有效的。

1. 课后任务

计算OBV

计算OBV：

OBV能量潮又称为平衡交易量,是由美国投资分析家葛兰碧在1981年创立的,它的理论基础是“能量是因,股价是果”。 其计算公式为：当日OBV＝前一日OBV＋sgn\*今日成交量。如果当日收盘价高于前日收盘价sgn取+1，反之取-1。

当OBV线下降，股价上升，表示买盘无力为卖出信号，2．OBV线上升，股价下降时，表示有买盘逢低介入，为买进信号。现尝试使用SQL将OBV计算出来。以便后续的分析

参考答案：

在MaxCompute的IDE中新建一个脚本文件，键入下列代码，点击运行按钮，即可得到OBV值。另外读者也可尝试在机器学习PAI平台中将下列计算OBV功能实现。

create table t1as select trde\_date

,close

,lag(close,1) over(partition by 1 order by trde\_date)as shift1\_close

,volumefrom pufa;

create table t2as select trde\_date

,close

,shift1\_close

,case when close>shift1\_close then 1 else -1 end as sgn

,volumefrom t1;

create table t3as select trde\_date

,close

,shift1\_close

,sgn

,sgn\*volume as yester\_obv

,volumefrom t2;

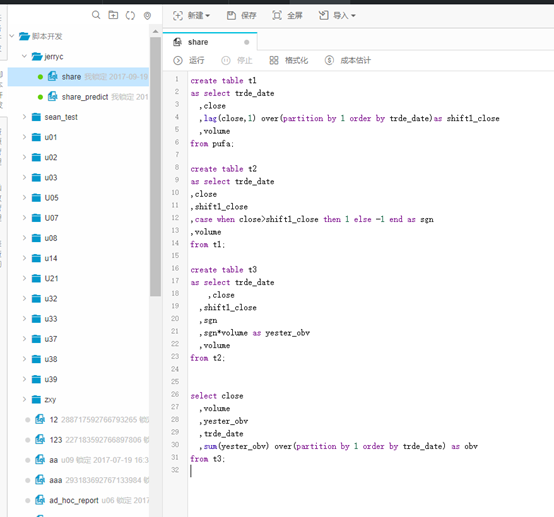
select close

,volume

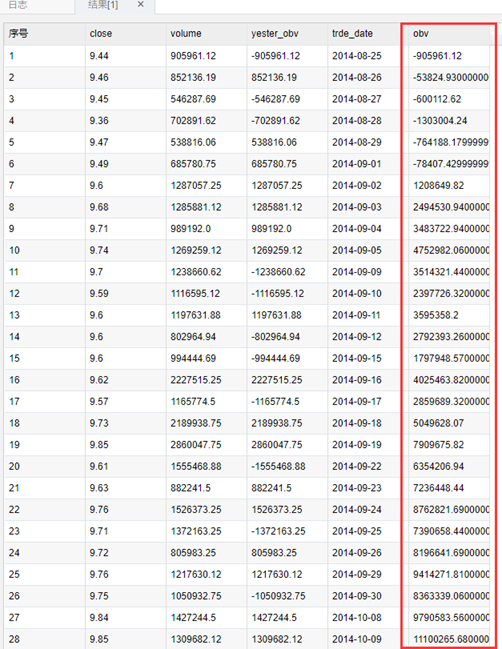
,yester\_obv

,trde\_date

,sum(yester\_obv) over(partition by 1 order by trde\_date) as obvfrom t3;



**运行结果：**



1. 思考与讨论

1.     在使用移动平均线构建股票交易策略的时候，如何选取移动平均的周期？

答案：

根据投资方式来选取移动平均的周期，如果是短期投资可以使用短期移动平均线和股票收盘价格之间的关系作出交易决策；如果是中长期投资可以使用中长期移动平均线和股票收盘价格之间的关系作出交易决策也可以通过中长期移动平均线和短期移动平均线之间的关系作出交易决策。但是长期投资不一定要使用移动平均这种趋势型策略进行投资，可以考虑使用价值型策略，即根据股票价格和企业内在价值之间的关系作出交易决策。

2.     除了使用移动平均线的方法构建交易策略模型外，还可以通过什么方式进行模型构建？

答案：

* 还有MACD指标，RSI指标等方式。MACD指标通过计算长短期指数移动平均线的离差值来作为交易信号。MACD指标在股票市场应用较为普遍。
* RSI指标根据一定时期内上涨点数和涨跌点数之和的比率来反映出市场在一定时期内的景气程度。它也分为长期和短期。RSI指标保持高于50表示为强势市场，反之低于50表示为弱势市场。一般情况下指标在0-20和50-80区间时可以考虑买入。但它通常是事后验证型的，事前很难看出，具有滞后性。