

5、测试数据挖掘及其组件

- 1. 配置连接信息
- 2. 运行数据挖掘示例
 - 2.1. 测试数据挖掘执行引擎
 - 2.2. 测试数据挖掘服务引擎
 - 2.3. 测试数据挖掘的Python计算

本节介绍smartbi连接数据挖掘、Spark、Hadoop、Python以及测试服务是否正常运行。



文档环境

单机部署数据挖掘组件环境如下：

服务器IP	主机名	组件实例	部署目录
10.10.204.248	10-10-204-248	数据挖掘	/data
10.10.204.249	10-10-204-249	Spark, Hadoop	/data
10.10.204.250	10-10-204-250	Python	/data

请根据实际部署环境替换相关的配置信息。

1. 配置连接信息



前提条件

配置数据挖掘连接信息前，请确认数据挖掘已正常部署和启动。部署文档参考：[部署数据挖掘](#)

① 浏览器访问Smartbi，打开**系统运维 - 数据挖掘配置 - 引擎设置**，引擎地址和服务地址设置成正确的数据挖掘引擎地址，并**点击保存**

导航 | 数据挖掘配置 x

引擎设置 | 执行引擎 | 服务引擎 | 作业流

引擎设置

引擎地址: 示例(http://localhost:8899)

服务地址: 示例(https://localhost:8900)

代理地址: 示例(https://ip:port)

Python代理器启动命令:

全局词典文件地址:

自定义python节点包模板:

自定义python算法节点包模板:

上传自定义python节点包: ☐ 覆盖上传

替换成实际地址

点击保存



注意事项

修改引擎和服务地址后，需要点击保存，否则执行引擎和服务引擎可能会显示空白页

配置数据挖掘地址后，首次测试引擎地址，会提示“平台到引擎连接成功，引擎到平台连接失败，请检查引擎配置”

http://10.10.204.248

平台到引擎连接成功，引擎到平台连接失败，请检查引擎配置

测试

https://10.10.204.248:8900

示例(https://localhost:8900)

测试

示例(https://ip:port)

首次测试服务地址则会提示“平台到服务连接成功，服务到平台连接失败，请检查服务配置”

http://10.10.204.248

平台到服务连接成功，服务到平台连接失败，请检查服务配置

测试

https://10.10.204.248:8900

示例(https://localhost:8900)

测试

示例(https://ip:port)

② 打开系统运维 - 数据挖掘配置 - 执行引擎 - 引擎配置，参考下图修改smartbi地址、hadoop地址(根据实际环境修改)，修改完成后点击保存：



前提条件

配置Hadoop连接信息前，请确认Hadoop节点已正常部署和启动。部署文档参考：[部署Hadoop](#)

数据挖掘配置

引擎配置

引擎服务端口: 8899

最大运行流数: 8 初始值 (10) 恢复初始值

最大等待流数: 10 初始值 (10000) 恢复初始值

引擎高可用时连接zookeeper地址: 10.10.204.239:2181

引擎高可用设置,默认为不可用: false

引擎agent超时时间(单位:毫秒): 60000 初始值 (60000) 恢复初始值

系统api地址: http://10.10.204.248:18080/smartbi/smartbi/api/mc 初始值 (空白) 替换成实际的smartbi访问地址 恢复初始值

节点数据是否存储: true 初始值 (true) 恢复初始值

节点数据是否计数: true 初始值 (true) 恢复初始值

节点数据目录: /usr/local/smartbi-mining-engine-bin/data

节点日志目录: /usr/local/smartbi-mining-engine-bin/logs

节点数据存储行数: 100 初始值 (100) 恢复初始值

python插件存储目录: /usr/local/smartbi-mining-engine-bin/conf/plugins/

java插件jar包存储目录: /usr/local/smartbi-mining-engine-bin/conf/plugins/

节点数据hdfs存储目录: hdfs://10.10.204.249:9000/mining/ 初始值 (webhdfs://enginecluster/mining/) 替换成实际的Hadoop地址 恢复初始值

节点数据hdfs访问控制列表: mining

保存

③ 打开系统运维 - 数据挖掘配置 - 执行引擎 - 计算节点配置，参考下图设置，修改完成后点击保存



前提条件

配置Spark连接信息前，请确认Spark节点已正常部署和启动。部署文档参考：[部署Spark3.1](#)

配置spark计算节点：

引擎配置 计算节点配置

master(运行模式配置(1.单机模式:local[*], 2.集群模式:spark://ip:7077)): spark://10.10.204.249:7077 初始值 (local[*]) 恢复初始值

executor.instances(executor数量): 2 初始值 (3) 恢复初始值

executor.cores(executor分配cpu个数): 1 初始值 (2) 恢复初始值

cores.max(分配给引擎的最大cpu个数): 2 初始值 (6) 恢复初始值

submit.deployMode(提交模式): client 恢复初始值

driver.memory(driver内存使用量): 4096m 初始值 (4096m) 恢复初始值

executor.memory(executor内存使用量): 4096m 初始值 (8192m) 恢复初始值

driver.maxResultSize(driver能接收的最大数据集): 500m 初始值 (500m) 恢复初始值

executor.extraJavaOptions(executor启动的jvm参数): -XX:+UnlockExperimentalVMOptions -XX:+UseG1GC -XX: 初始值 (-XX:+UnlockExperimentalVMOptions -XX:+UseParallelOldGC) 恢复初始值

driver.allowMultipleContexts(是否允许多个sparkcontext): true 初始值 (true) 恢复初始值

sql.broadCastTimeout(广播超时时间(单位:秒)): 3600 初始值 (3600) 恢复初始值

sql.autoBroadcastJoinThreshold(broadcastjoin大小): 10485760 初始值 (10485760) 恢复初始值

sql.shuffle.partitions(shuffle的并行度): 200 初始值 (200) 恢复初始值

shuffle.file.buffer(shuffle的缓存大小): 32K 初始值 (32K) 恢复初始值

local.dir(executor缓存目录): spark-local 初始值 (spark-local) 恢复初始值

driver.port(driver监听端口): 7777 初始值 (7777) 恢复初始值

一键推荐 保存

配置Spark节点资源，点击一键推荐，系统会根据Spark work节点的服务器资源，生成推荐的配置(如果使用推荐值，记得点击保存，否则配置不生效)：

引擎配置 计算节点配置

master(运行模式配置(1.单机模式:local[*], 2.集群模式:spark://ip:7077)): spark://10.10.204.249:7077 初始值 (local[*]) 恢复初始值

executor.instances(executor数量): 2 初始值 (3) 恢复初始值

executor.cores(executor分配cpu个数): 1 初始值 (2) 恢复初始值

cores.max(分配给引擎的最大cpu个数): 2 初始值 (6) 恢复初始值

submit.deployMode(提交模式): client 恢复初始值

driver.memory(driver内存使用量): 4096m 初始值 (4096m) 恢复初始值

executor.memory(executor内存使用量): 4096m 初始值 (8192m) 恢复初始值

driver.maxResultSize(driver能接收的最大数据集): 500m 初始值 (500m) 恢复初始值

executor.extraJavaOptions(executor启动的jvm参数): -XX:+UnlockExperimentalVMOptions -XX:+UseG1GC -XX: 初始值 (-XX:+UnlockExperimentalVMOptions -XX:+UseParallelOldGC) 恢复初始值

driver.allowMultipleContexts(是否允许多个sparkcontext): true 初始值 (true) 恢复初始值

sql.broadCastTimeout(广播超时时间(单位:秒)): 3600 初始值 (3600) 恢复初始值

sql.autoBroadcastJoinThreshold(broadcastjoin大小): 10485760 初始值 (10485760) 恢复初始值

sql.shuffle.partitions(shuffle的并行度): 200 初始值 (200) 恢复初始值

shuffle.file.buffer(shuffle的缓存大小): 32K 初始值 (32K) 恢复初始值

local.dir(executor缓存目录): spark-local 初始值 (spark-local) 恢复初始值

driver.port(driver监听端口): 7777 初始值 (7777) 恢复初始值

推荐配置项【点击保存才生效】 根据实际Spark work节点资源给出推荐值

配置项	当前值	推荐值
executor数量	1	2
executor分配cpu个数	1	1
executor内存使用量	4096m	4096m
分配给引擎的最大cpu个数	1	2

1.点击一键推荐 2.点击保存

一键推荐 保存

④ 打开系统运维 - 数据挖掘配置 - 服务引擎，参考下图修改smartbi地址(根据实际环境修改)，修改完成后点击保存：

引擎配置

引擎服务端口:	8900	
引擎高可用时连接zookeeper地址:	smartbi-zookeeper2181	
引擎高可用设置,默认为不可用:	false	
系统api地址:	http://10.10.204.248:18080/smartbi/smartbix/api/mc	初始值 (空白)
节点数据是否存储:	true	初始值 (true)
节点数据是否计数:	true	初始值 (true)
节点数据目录:	/usr/local/engine/data	
节点日志目录:	/usr/local/engine/logs	
节点数据存储行数:	100	初始值 (100)
python插件存储目录:	/usr/local/engine/conf/plugins/python	
java插件jar包存储目录:	/usr/local/engine/conf/plugins/java	
节点数据hdfs存储目录:	webhdfs://enginecluster/mining/	初始值 (webhdfs://enginecluster/mining/)
节点数据hdfs访问控制列表:	mining	

替换成实际的smartbi访问地址

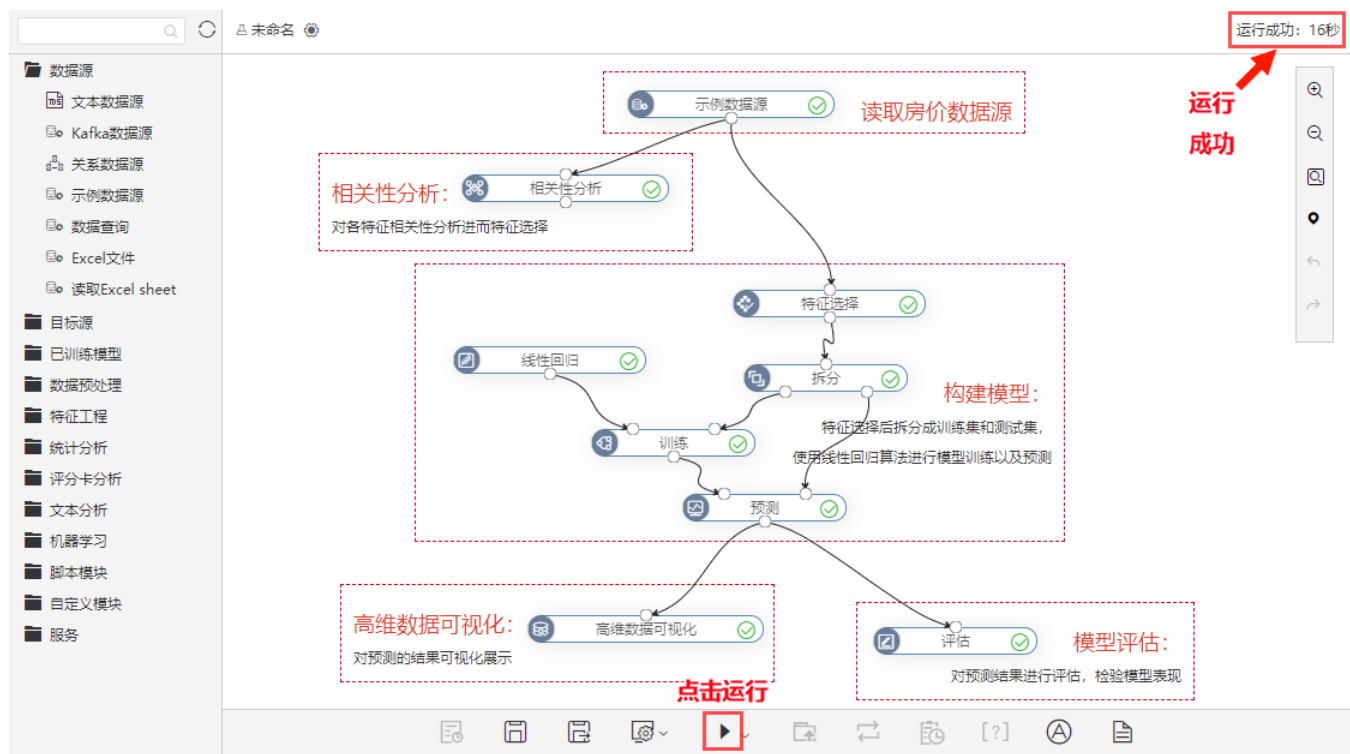
⑤ 执行引擎和服务引擎完成设置、保存后,返回 系统运维 - 数据挖掘配置 - 引擎设置,重新点击测试,如下提示表示配置成功:

http://10.10.204.248:	✓ 平台和引擎双向连通	✕	测试
https://10.10.204.248:8900	示例(https://localhost:8900)		测试
http://10.10.204.248:	✓ 平台和服务双向连通	✕	测试
https://10.10.204.248:8900	示例(https://localhost:8900)		测试

2. 运行数据挖掘示例

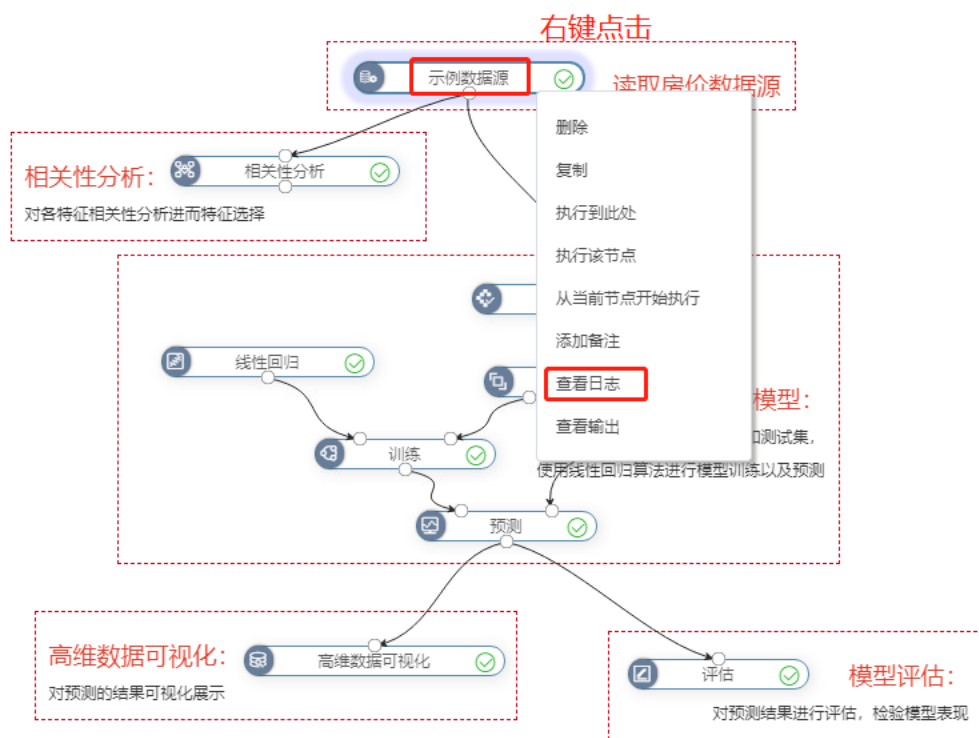
2.1. 测试数据挖掘执行引擎

打开数据挖掘 - 案例一波士顿房价预测,点击运行,如下图显示运行成功即可



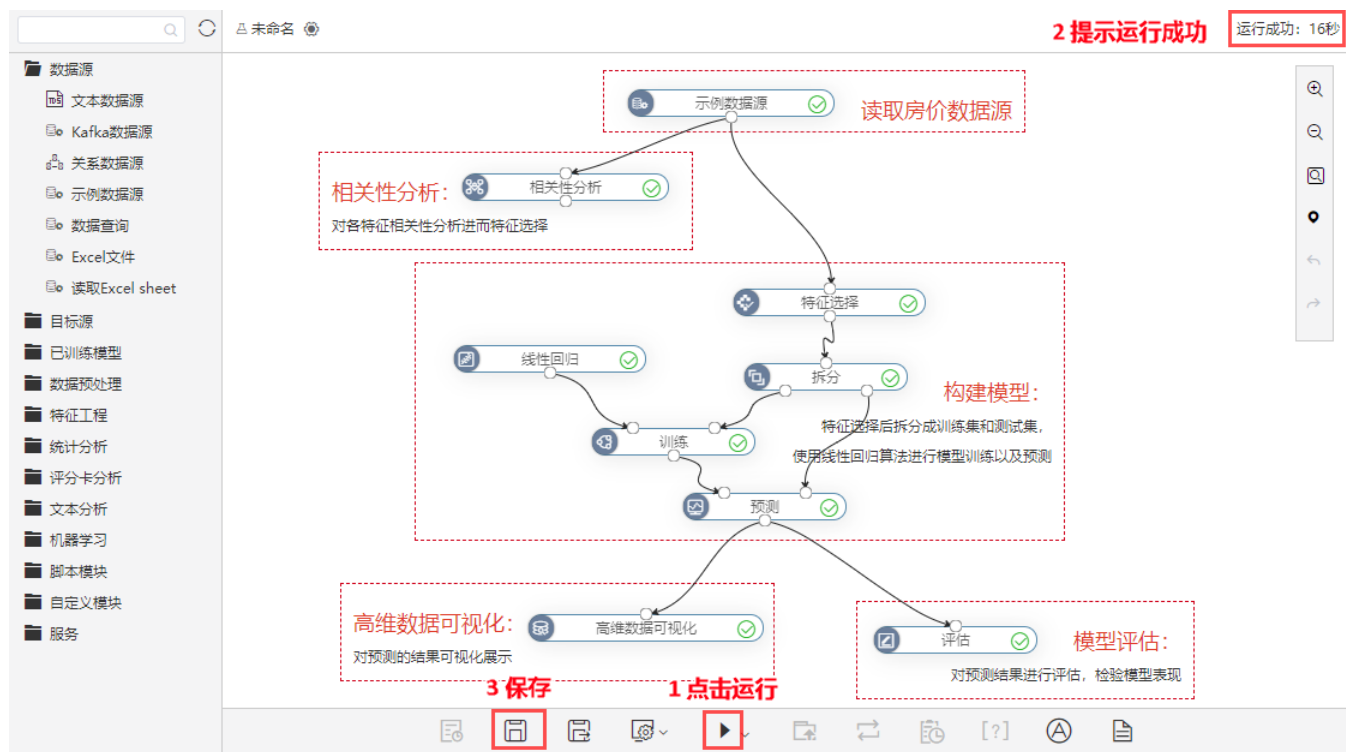
如果某个节点运行失败, 可以右键点击节点, 选择查看日志分析错误原因:

未命名 运行成

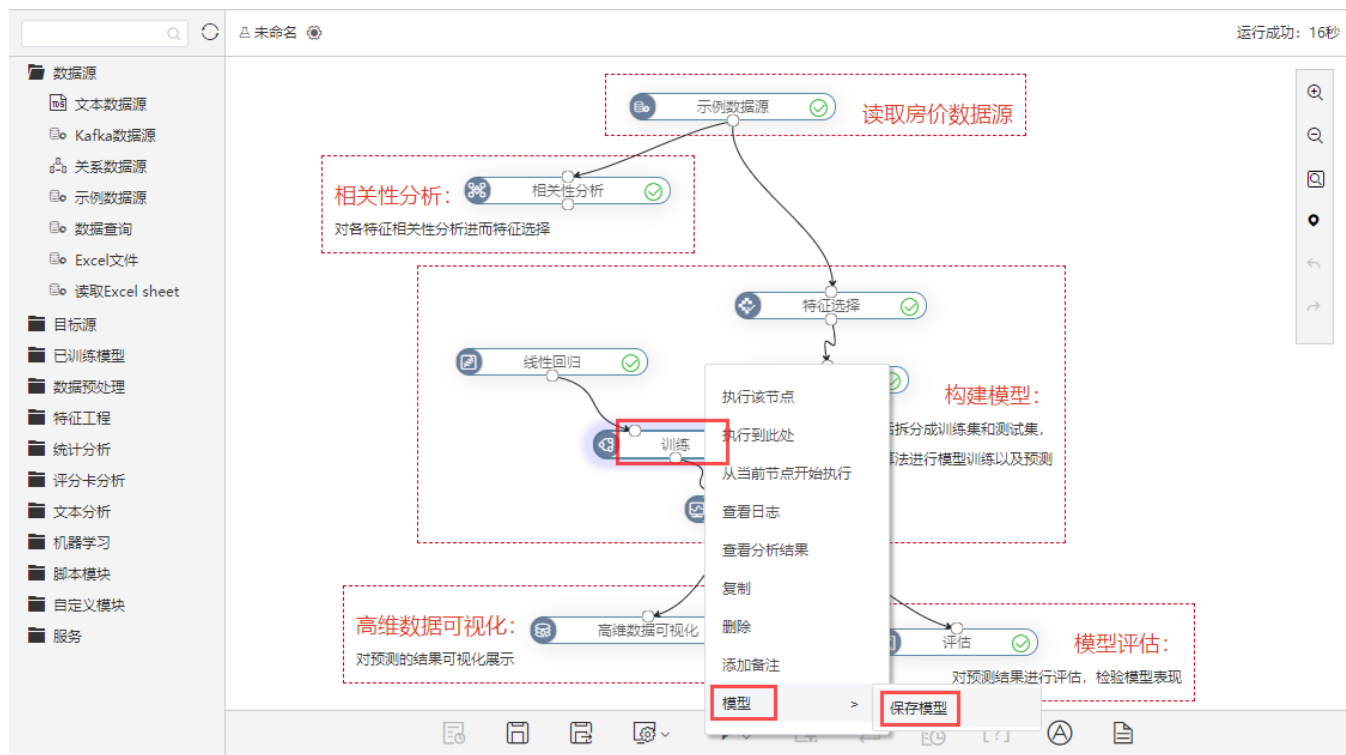


2.2. 测试数据挖掘服务引擎

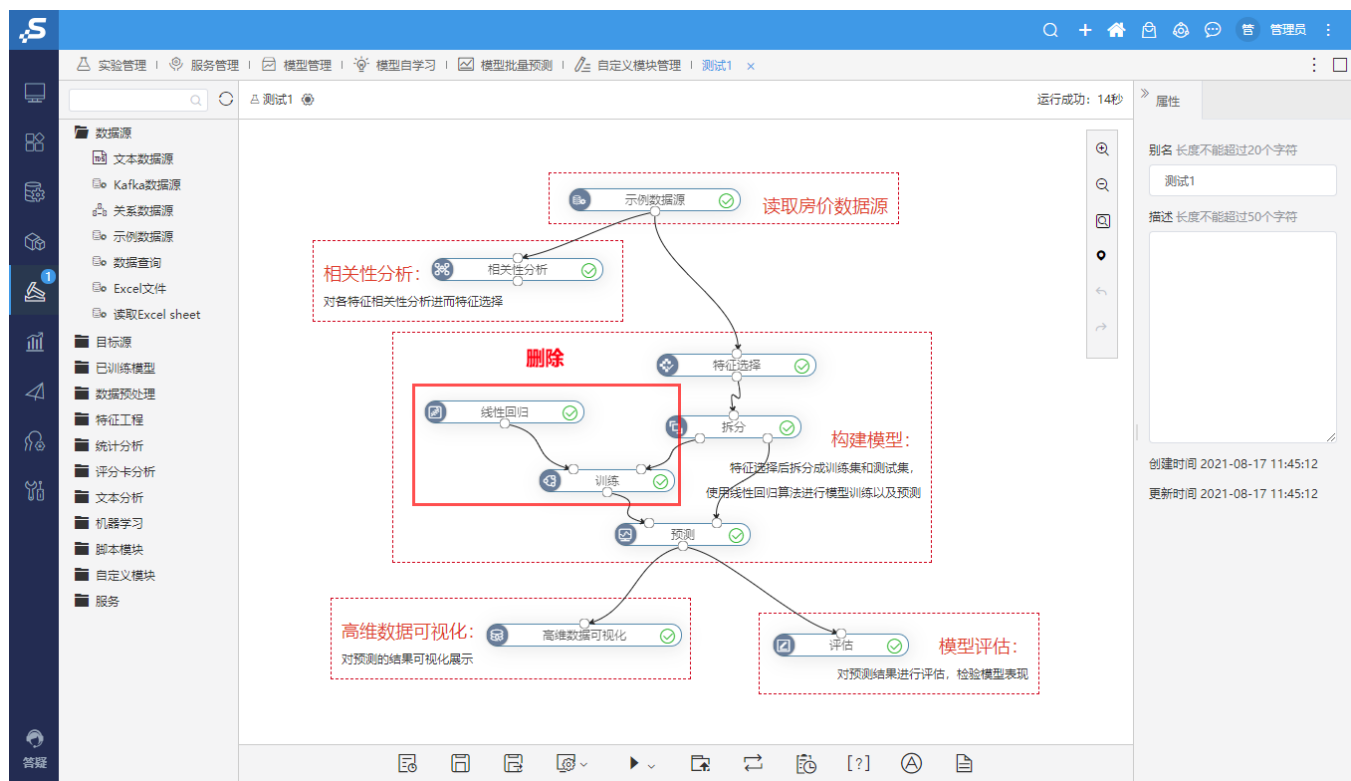
① 打开数据挖掘 - 案例一波士顿房价预测, 点击运行, 如下图显示运行成功, 并点击保存案例



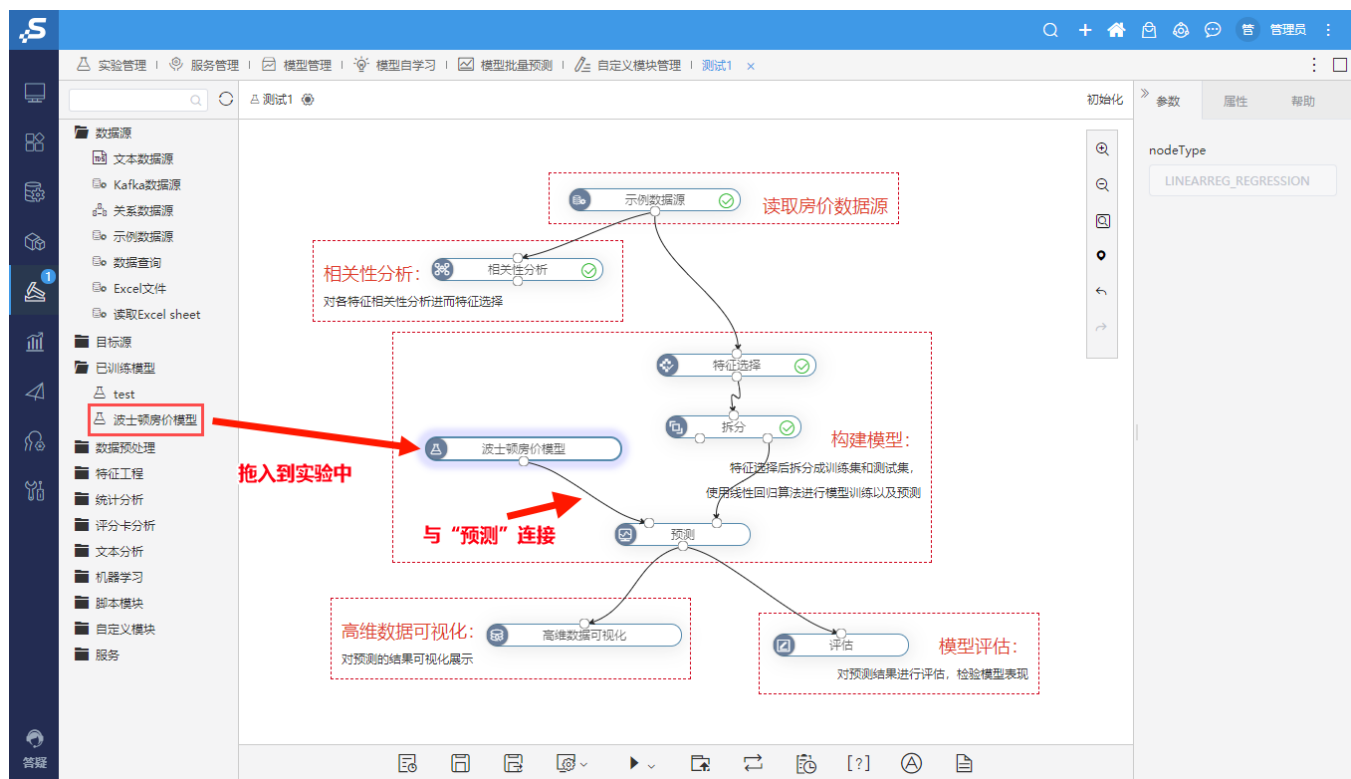
② 右键点击 **训练—模型—保存模型**，如下图所示



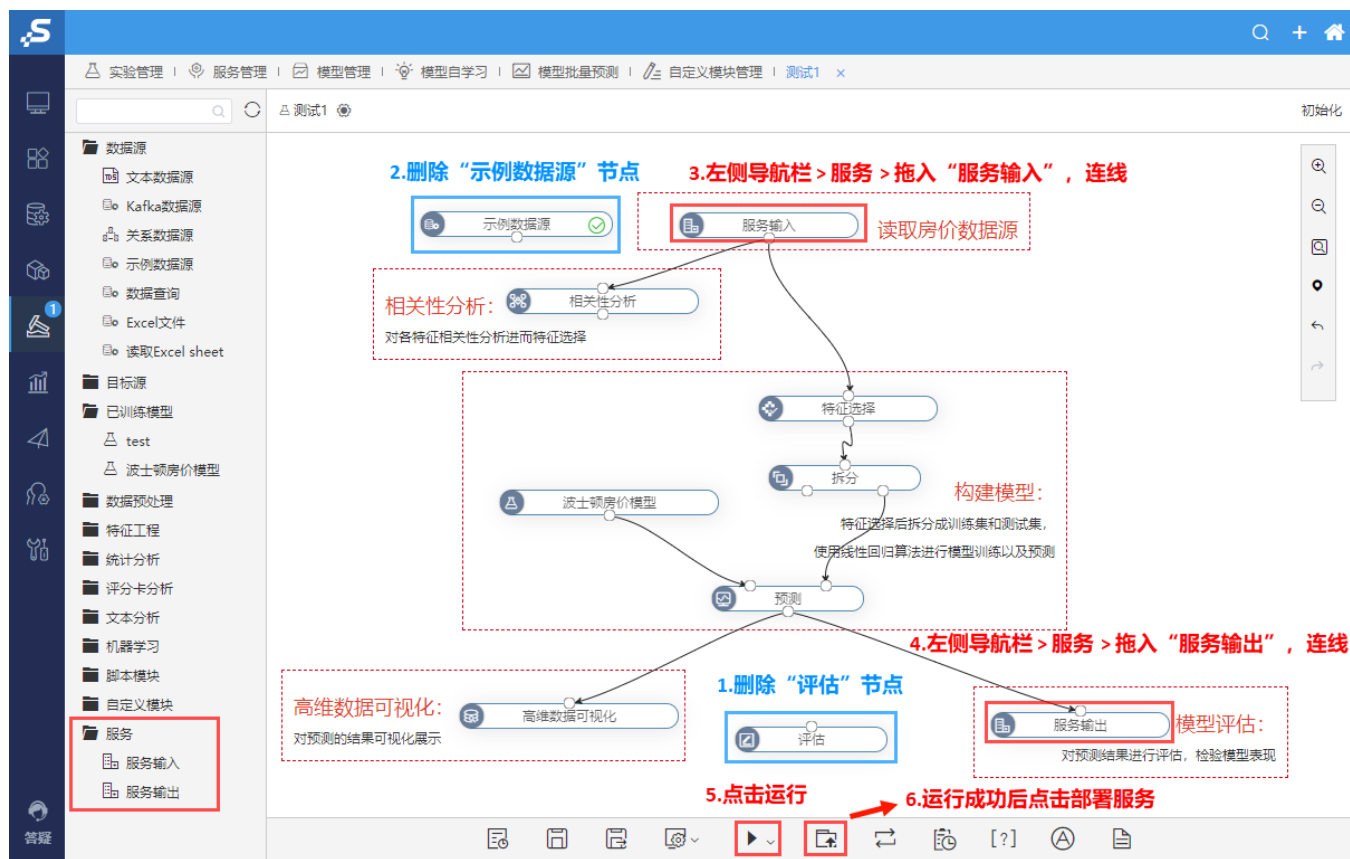
③ 保存的模型可以在左侧导航栏的“已训练模型”中查看。右键删除“线性回归”，“训练”。



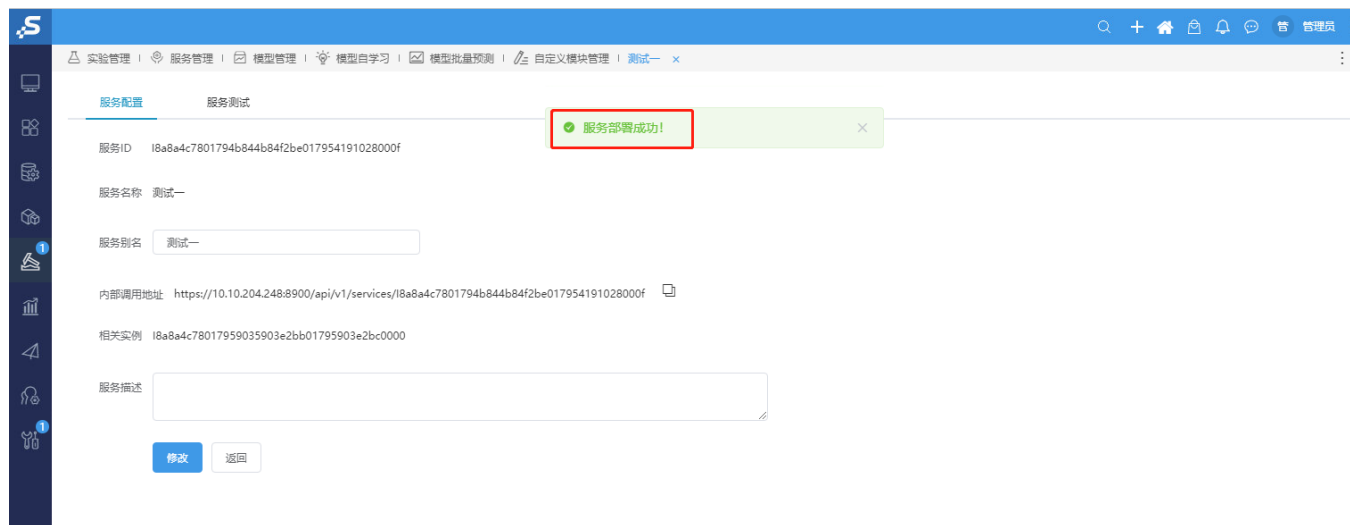
④ 将保存的模型拖入，并与“预测”连线。



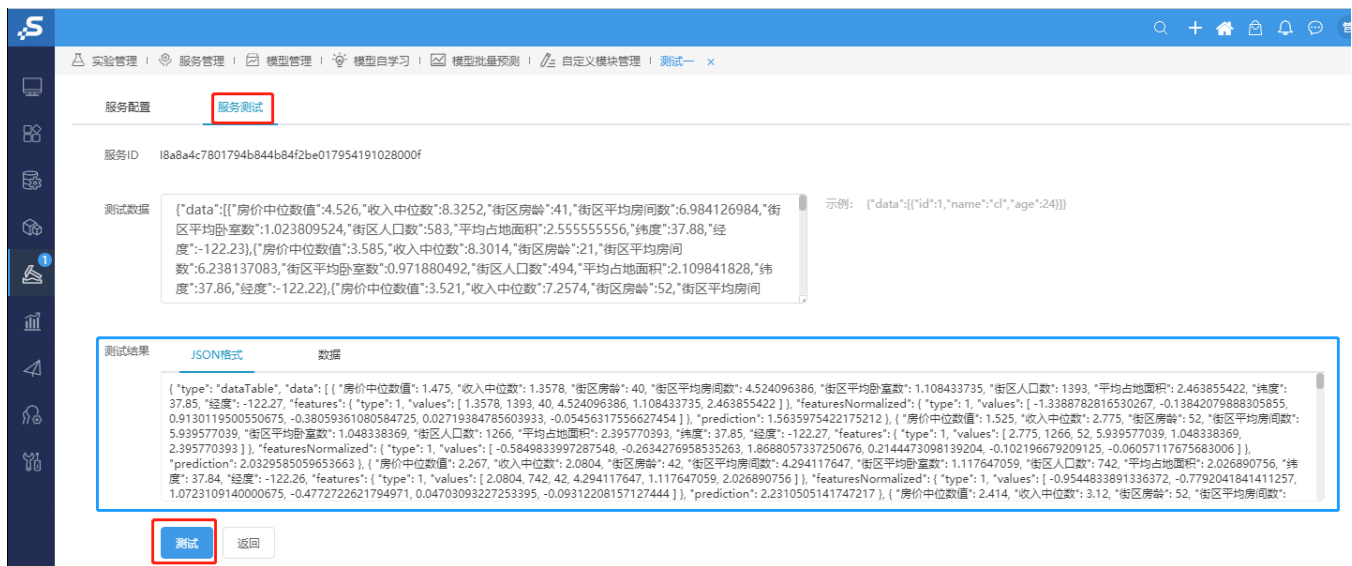
⑤ 导航栏-服务-拖入 服务输入、服务输出，并连线，运行成功后点击部署服务



⑥ 显示部署成功



⑦ 点击服务测试，会显示测试结果



2.3. 测试数据挖掘的Python计算



前提条件

测试数据挖掘的Python计算时，请确认Python节点已正常部署和启动。部署文档参考：[部署Python节点](#)

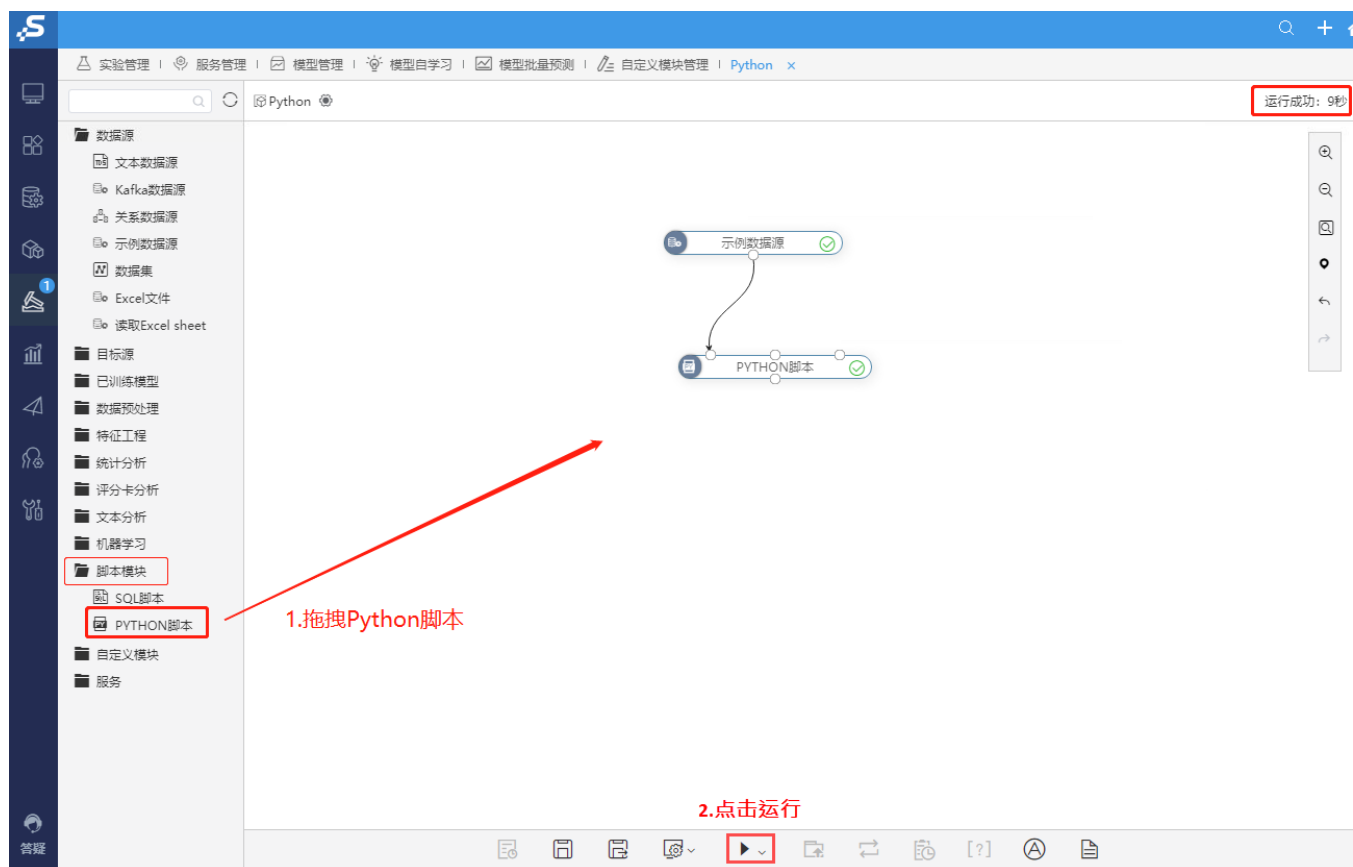
① 打开数据挖掘—新建实验



② 拖拽“示例数据源”，选择数据源 - 保存 - 运行 - 运行成功—保存



③ 拖拽PYTHON脚本，与示例数据源连线，点击运行，如果显示运行成功则表示Python计算节点正常



数据挖掘及其组件测试完成。