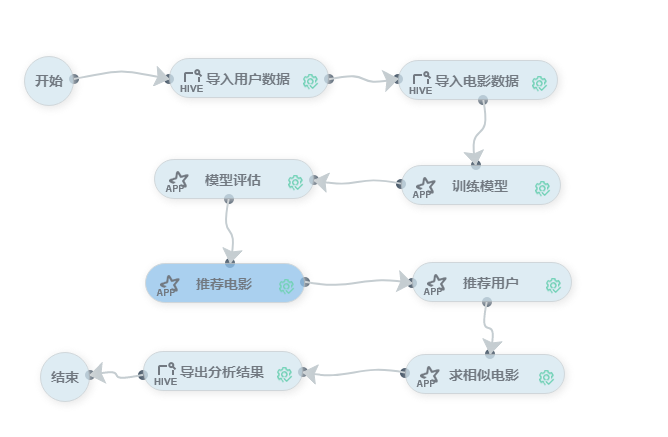
**工作流操作说明之电影推荐**

本案例数据量为100万条，主要是针对用户对观看电影的评分数据进行数据建模，用建立好的模型对测试集数据进行预测，为用户推荐那些未看过但可能会感兴趣的电影，并计算电影之间的相似度，取相似度最高的前10部电影进行推荐。

工作流需要有“开始”、“结束”组件，且各个组件用箭头连接。

本案例的总流程为：



1、创建新任务

在“工作流”中点击“创建新任务”，开始编辑任务。

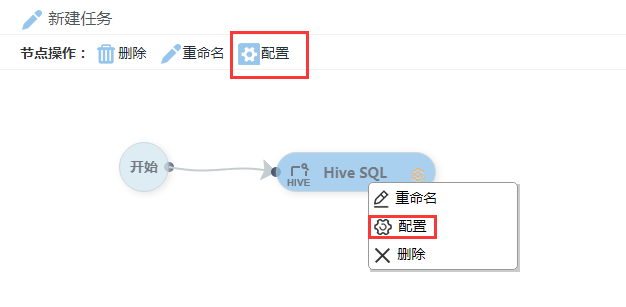


2、创建第一个节点Hive SQL：导入用户数据

将“开始”组件拖拽到工作界面，然后新建Hive SQL节点，从左侧将“Hive SQL”组件拖拽到工作界面。



配置组件的方法有三种：（1）双击组件直接进行配置（2）右击选择“配置”（3）选择工作栏中的配置



标题设置为：导入用户数据

编辑相应SQL语句，然后点击“确定”。

use courses;

drop table if exists user\_movie;

create table if not exists user\_movie(userid int, movieid int, rating float)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

load data local inpath '/home/coursedata/movie\_recommendation/ratings.csv' into table user\_movie;



3、创建第二个节点Hive SQL：导入电影数据



新建Hive SQL节点并进行配置

标题设置为：导入电影数据

编辑相应SQL语句，然后点击“确定”。

use courses;

drop table if exists movie;

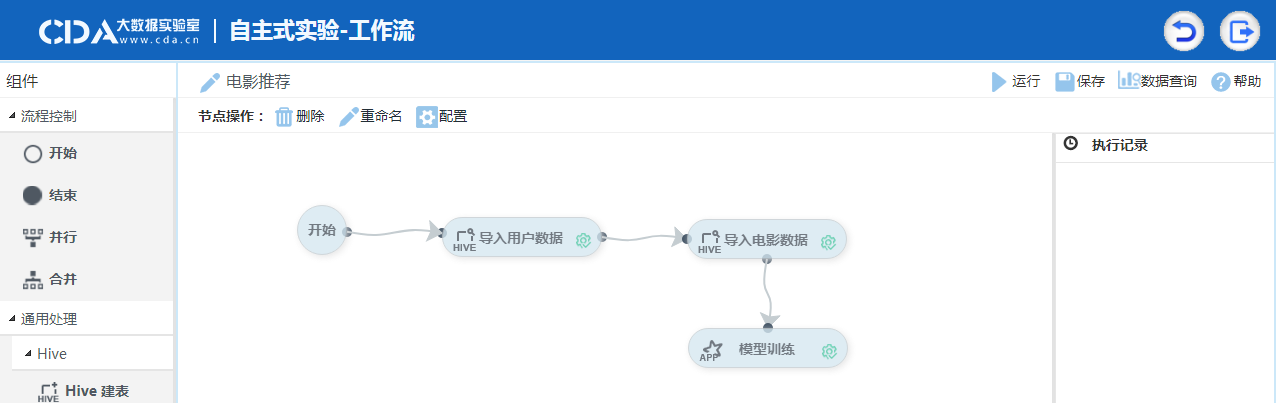
create table movie(movieid int, title string, year string)

row format delimited fields terminated by '|';

load data local inpath '/home/coursedata/movie\_recommendation/movies.csv' into table movie;



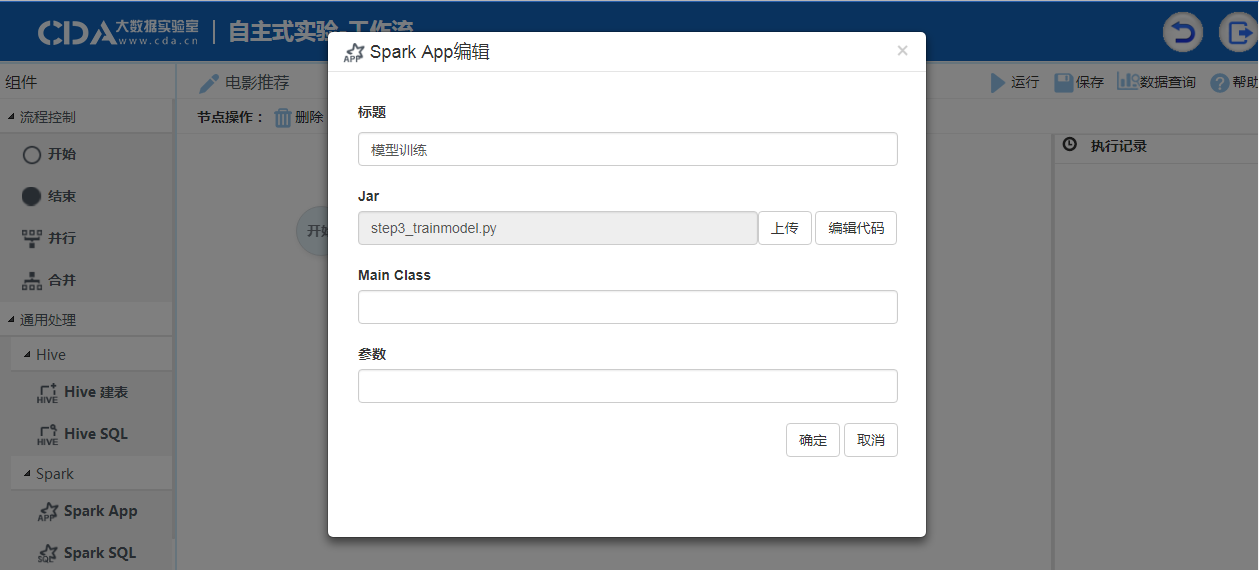
4、创建第三个节点Spark App：模型训练



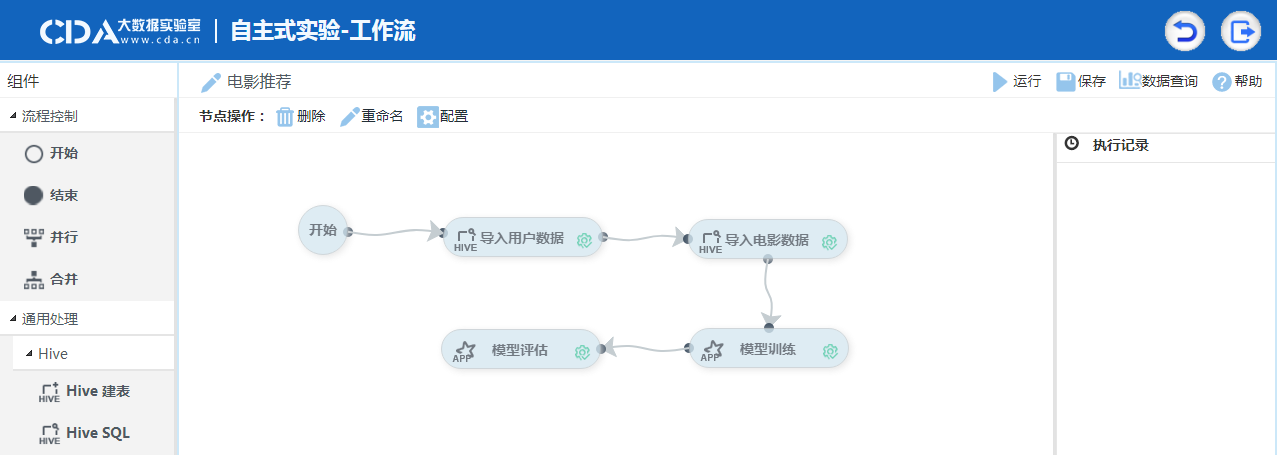
新建Spark App节点并进行配置

标题设置为：模型训练

上传所需文件：step3\_trainmodel.py，然后点击“确定”。



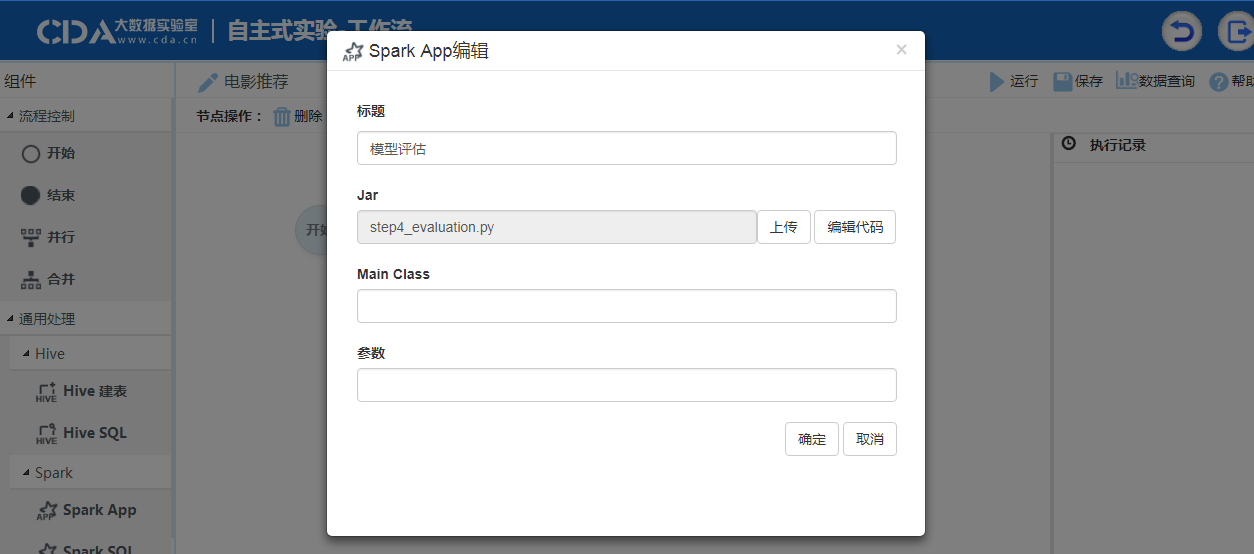
5、创建第四个节点Spark App：模型评估



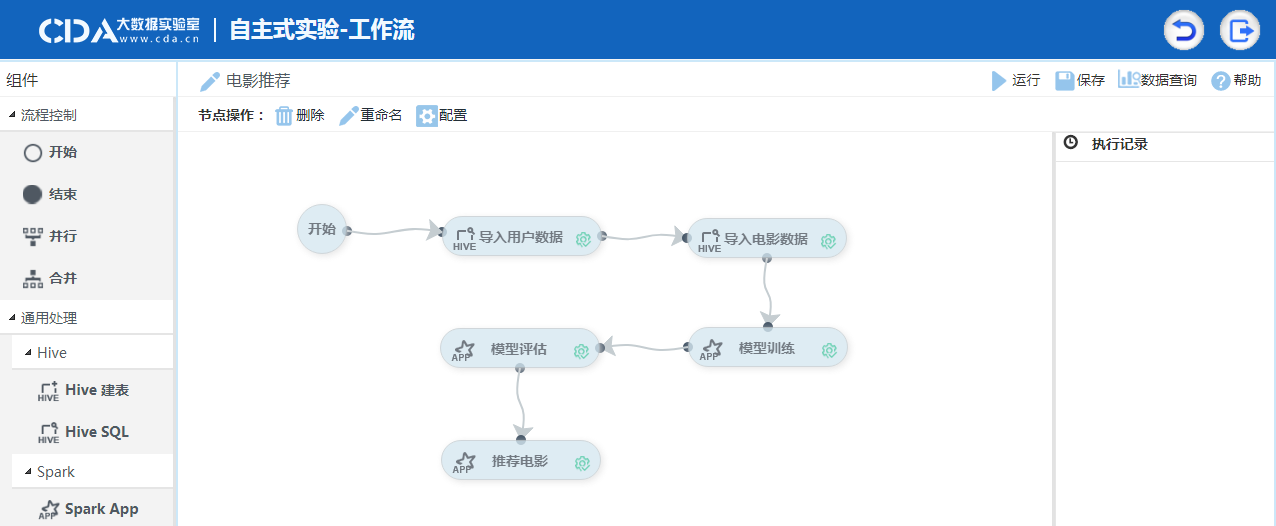
新建Spark App节点并进行配置

标题设置为：模型评估

上传所需文件：step4\_evaluation.py，然后点击“确定”。



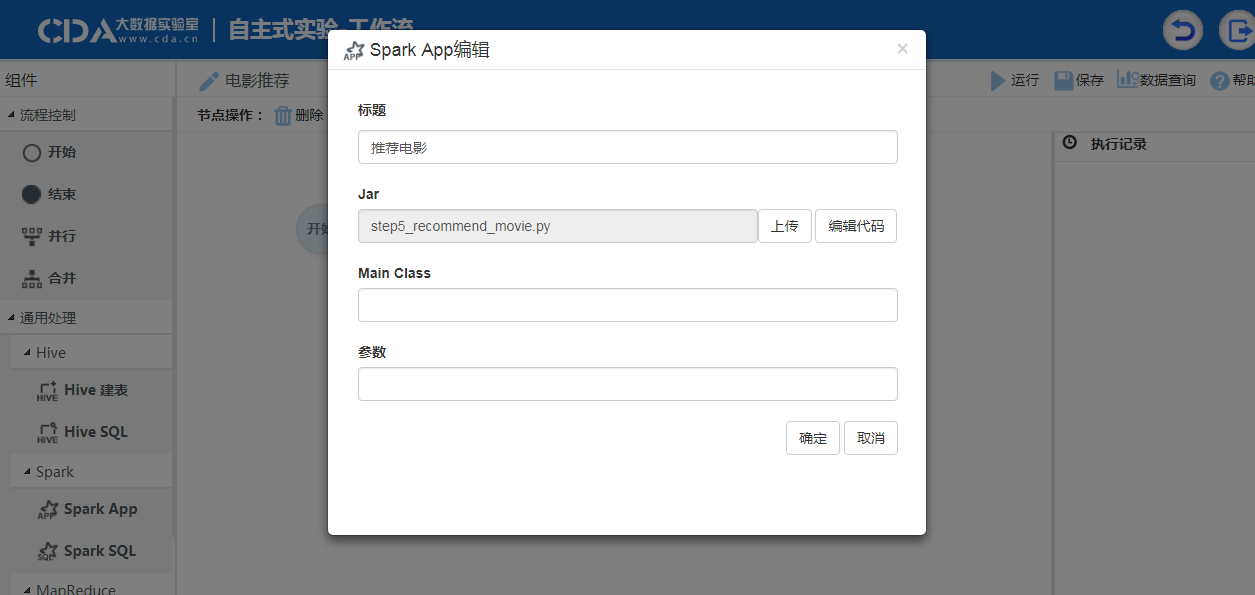
6、创建第五个节点Spark App：推荐电影



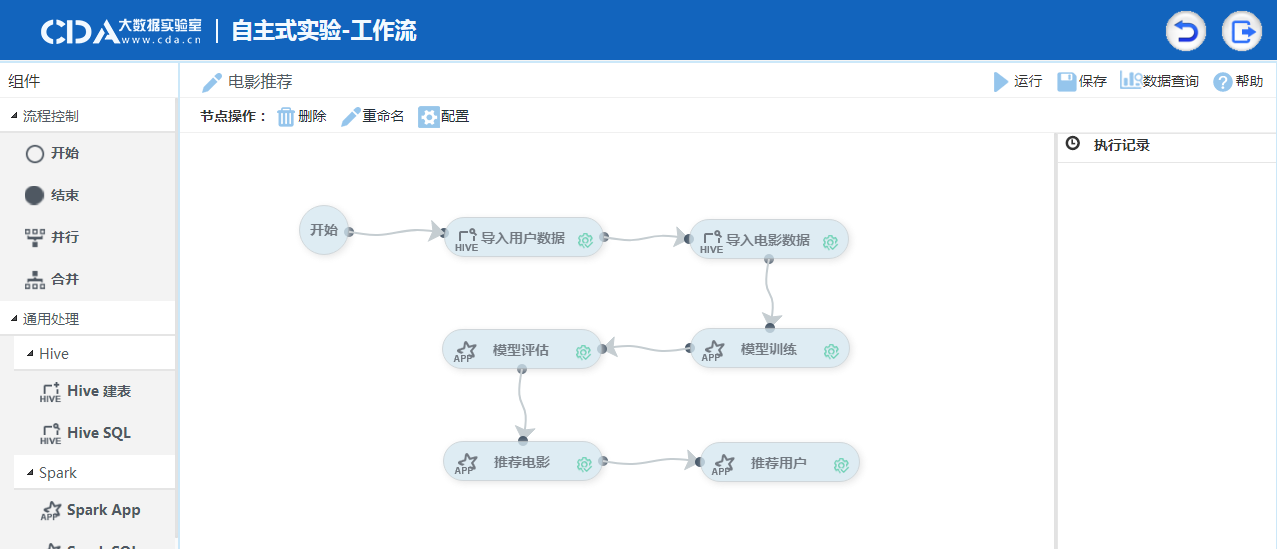
新建Spark App节点并进行配置

标题设置为：推荐电影（**将某个电影推荐给不同的用户**）

上传所需文件：step5\_recommend\_movie.py，然后点击“确定”。



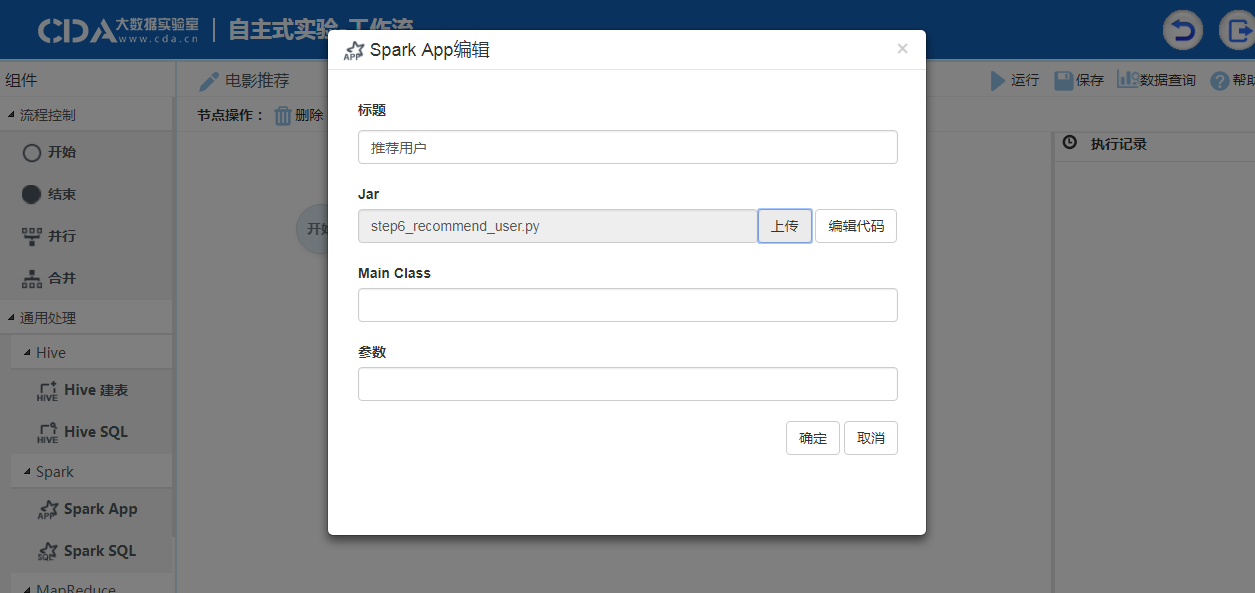
7、创建第六个节点Spark App：推荐用户



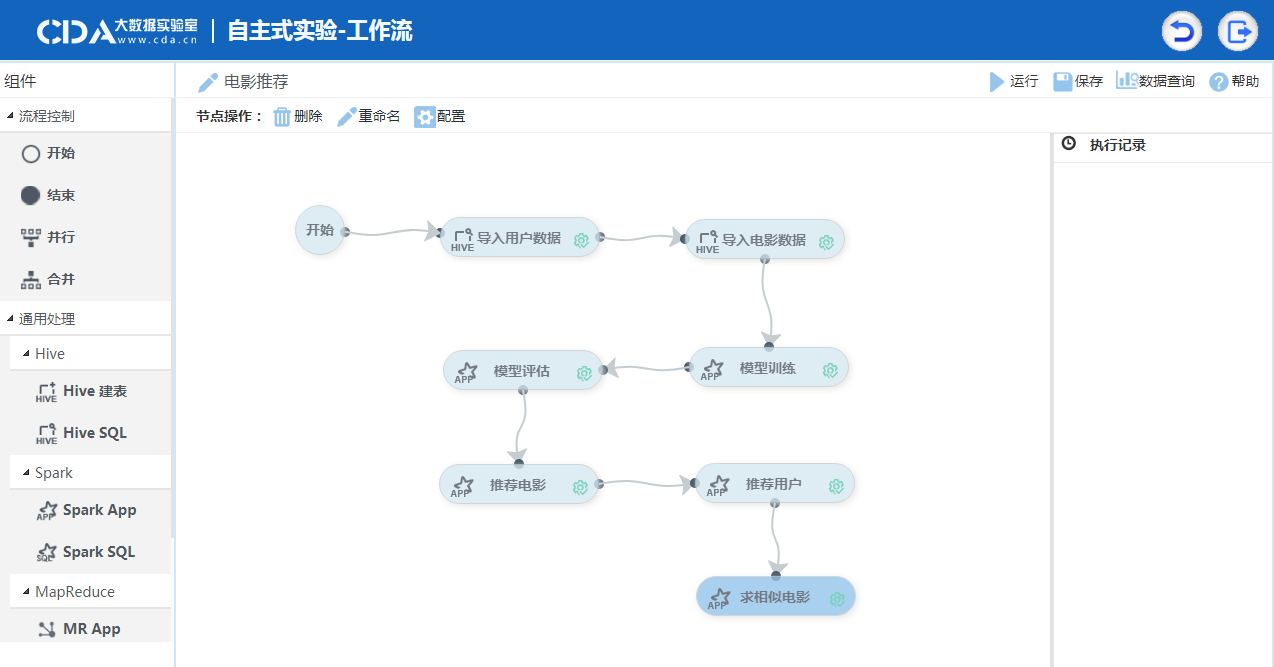
新建Spark App节点并进行配置

标题设置为：推荐用户（**为某个用户推荐不同的电影**）

上传所需文件：step6\_recommend\_user.py，然后点击“确定”。



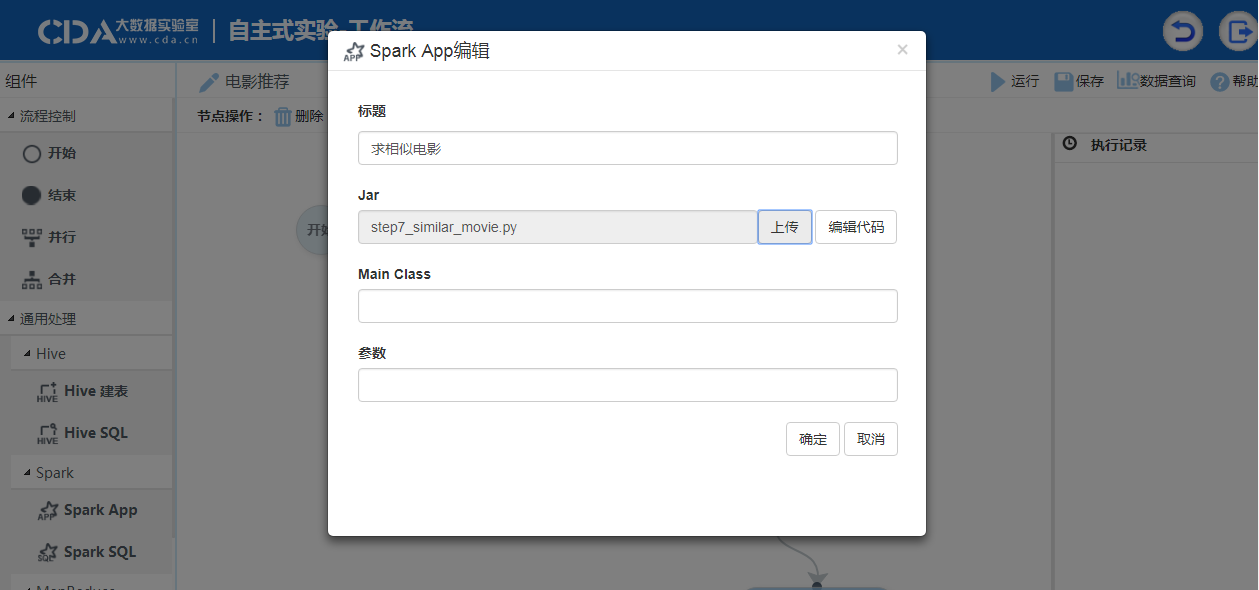
8、创建第七个节点Spark App：求相似电影



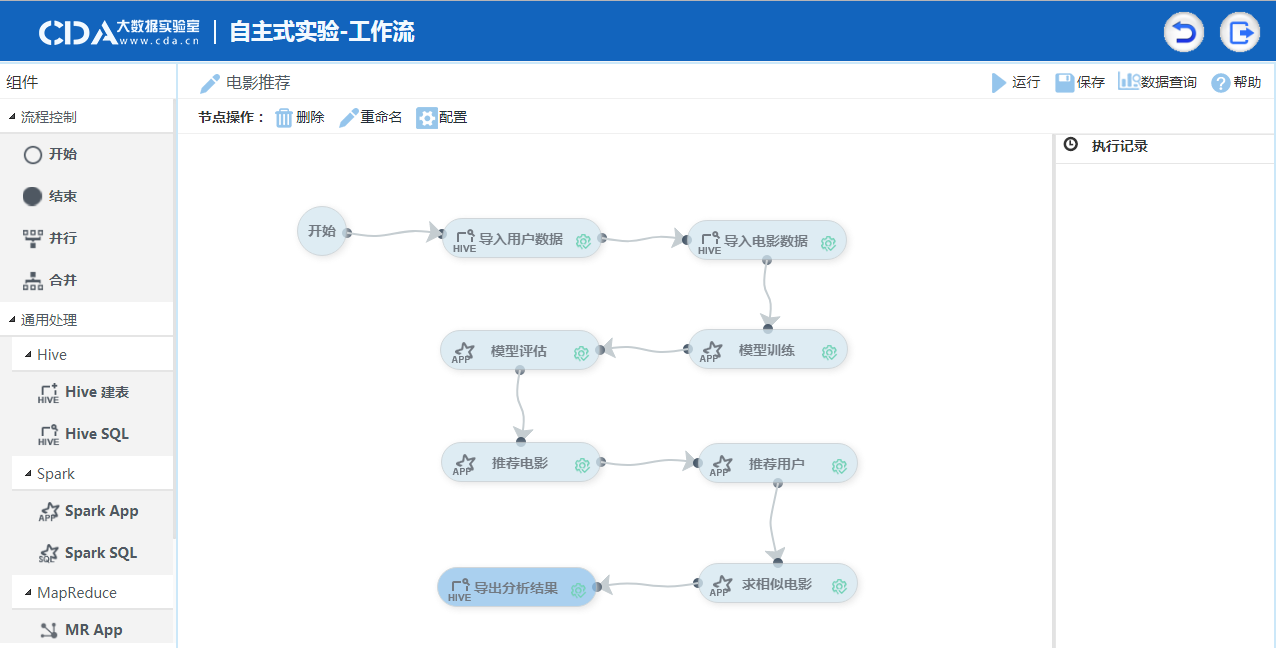
新建Spark App节点并进行配置

标题设置为：求相似电影

上传所需文件：step7\_similar\_movie.py，然后点击“确定”。



9、创建第八个节点Hive SQL：导出分析结果



新建Hive SQL节点并进行配置

标题设置为：导出分析结果

编辑相应SQL语句，然后点击“确定”。

drop table if exists results.movie\_similarmovies;

drop table if exists results.movie\_evaluateresult;

drop table if exists results.movie\_movies\_recommend;

drop table if exists results.movie\_user\_recommend;

create table if not exists results.movie\_similarmovies as select \* from courses.similarmovies;

create table if not exists results.movie\_evaluateresult as select \* from courses.evaluateresult;

create table if not exists results.movie\_movies\_recommend as select \* from courses.movie\_recommend;

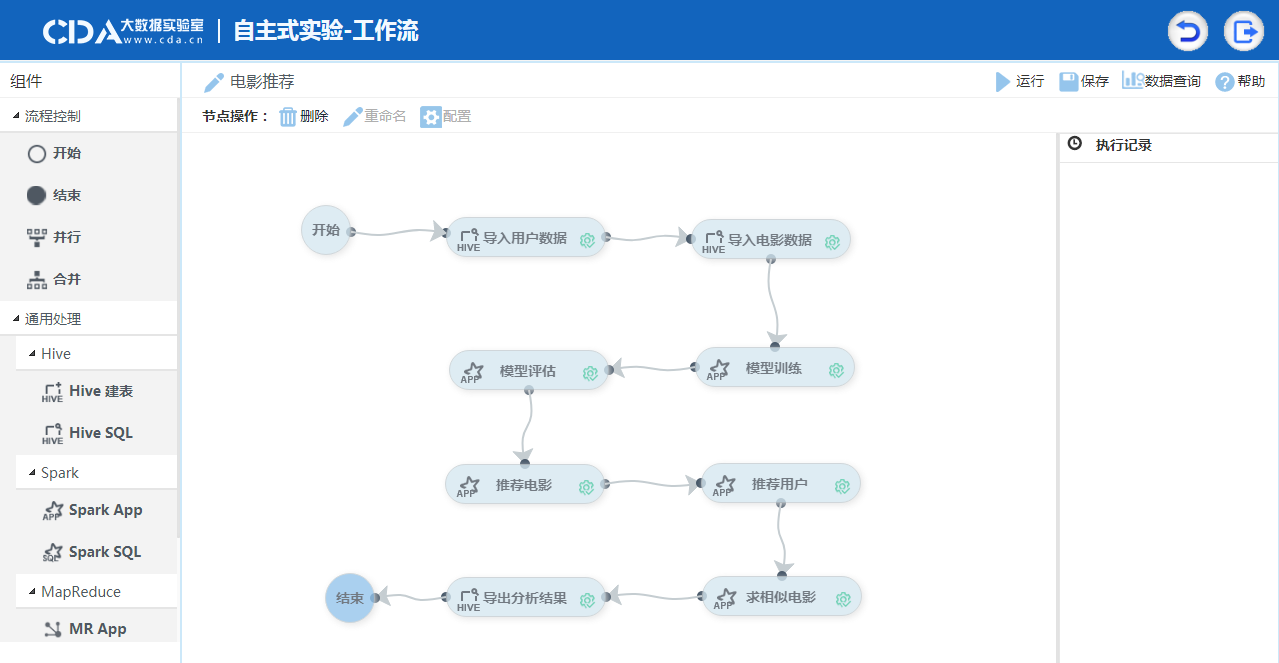
create table if not exists results.movie\_user\_recommend as select \* from courses.user\_recommend;

create table if not exists results.movie\_table\_info as select count(\*) as row\_count from courses.user\_movie;



10、保存并提交任务

添加“结束”节点，点击“保存”，修改任务名称为：电影推荐。

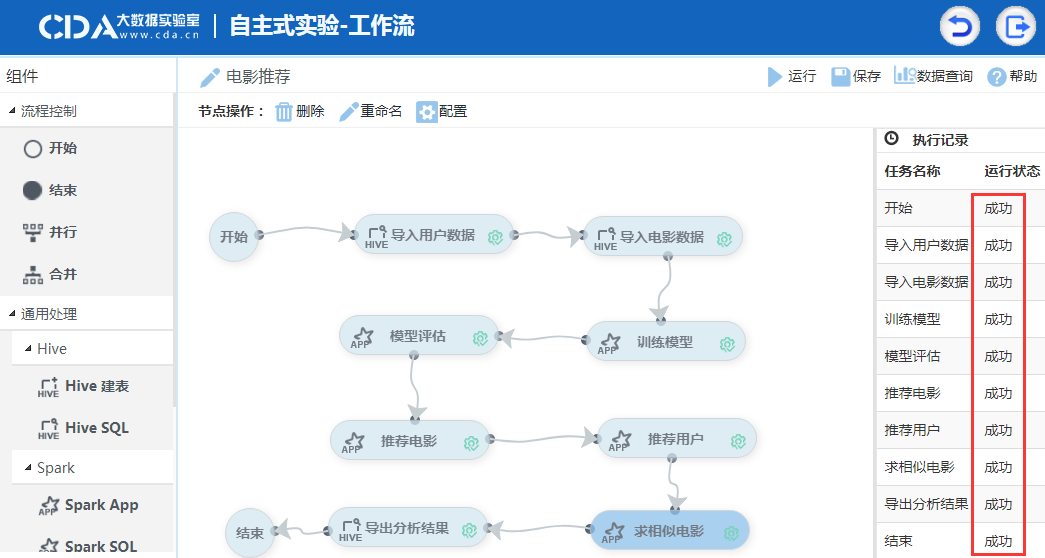




然后点击右上方“运行”按钮，选择“立即执行”，即可在右边“执行记录”中看到任务运行状态。



在右侧“执行记录”中，可以清晰的看到各节点的运行情况，如果遇到失败的节点，可返回上一级界面中的“历史任务”中查看具体出错原因。

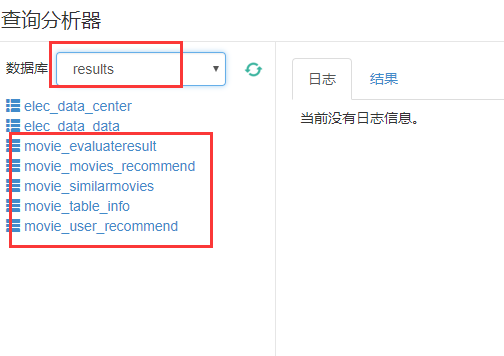


11、查询结果

（1）点击本界面的“数据查询”会跳转到“查询分析器”界面。



（2）选择保存数据的数据库：results，点击其中的结果表即可查看结果。



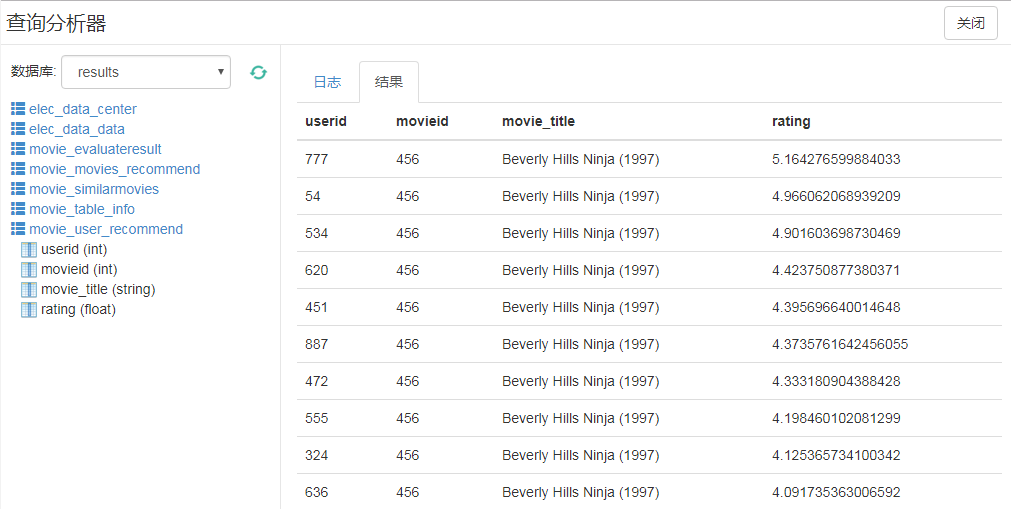
（3）点击movie\_evaluateresult，此为第4个节点“模型评估”结果表，参数信息：

Mse：（Mean Squared Error）方差/均方差

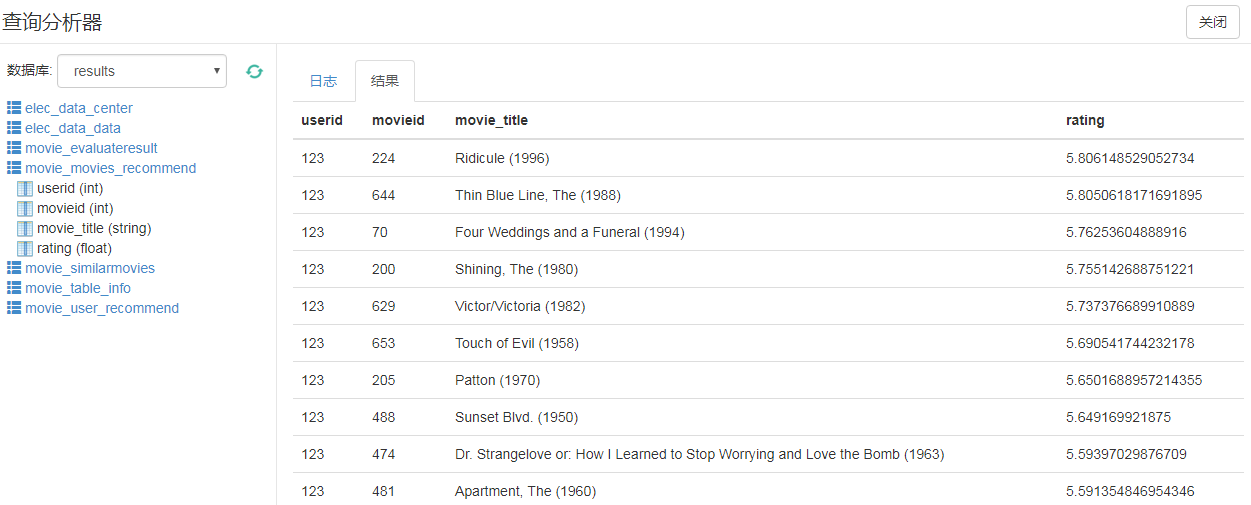
Rmse：（Root Mean Square Error）标准差/均方根

Mae：（Mean Absolute Error）平均绝对差值

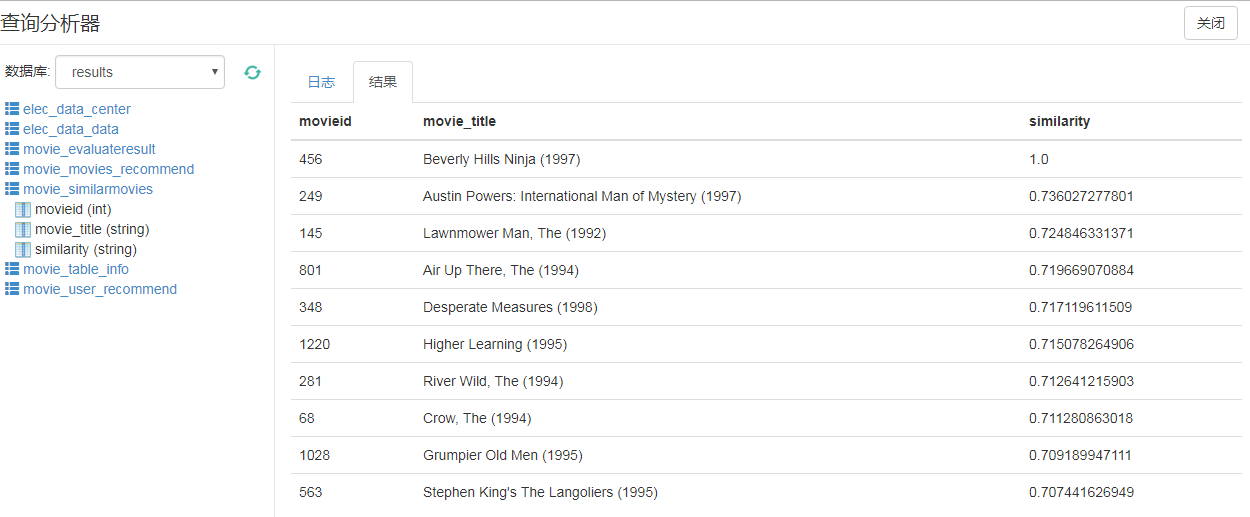
（4）点击movie\_user\_recommend，此为第5个节点“推荐电影”结果表，实现将电影推荐给的用户。



（5）点击movie\_movies\_recommend,此为第6个节点“推荐用户”结果表，实现了为用户推荐偏好值最高的电影。



（6）点击movie\_similarmovies，此为第7个节点“求相似电影”结果表。



（7）点击movie\_table\_info，此为电影数据总量表，可以看到数据总量为10万多条。

