**目录**

[一、运行环境 1](#_Toc531704031)

[二、本地数据集上传到数据仓库Hive 4](#_Toc531704032)

[2.1 数据集的预处理 5](#_Toc531704033)

[2.1.1 删除文件第一行记录（字段名称) 5](#_Toc531704034)

[2.1.2 获取数据集中前10000条数据 5](#_Toc531704035)

[2.1.3 把small\_user\_log.csv中的数据导入数据仓库Hive 6](#_Toc531704036)

[三、Hive数据分析 9](#_Toc531704037)

[3.1 查看user\_log表 9](#_Toc531704038)

[3.2 查询条数及统计分析 10](#_Toc531704039)

[3.2.1 计算表内数据行数 10](#_Toc531704040)

[3.2.2 查询uid不重复的数据条数 10](#_Toc531704041)

[3.2.3 查询不重复数据条数 11](#_Toc531704042)

[3.2.4 统计男女买家购买商品数量 11](#_Toc531704043)

[3.2.5 查询某一天在网站购买该数量商品的用户id 12](#_Toc531704044)

[3.3 用户实时查询分析 12](#_Toc531704045)

[3.3.1 创建新数据表 12](#_Toc531704046)

[3.3.2 导入统计后的数据 12](#_Toc531704047)

[3.3.3 显示统计结果 13](#_Toc531704048)

[四、将数据从Hive导入到MySQL 13](#_Toc531704049)

[4.1 Hive预操作 13](#_Toc531704050)

[4.1.1 创建临时表 13](#_Toc531704051)

[4.1.2 数据插入 14](#_Toc531704052)

[4.1.3 查看是否成功执行 14](#_Toc531704053)

[4.2 使用Sqoop将数据从Hive导入MySQL 14](#_Toc531704054)

[4.2.1 创建数据库 14](#_Toc531704055)

[4.2.2 创建表 15](#_Toc531704056)

[4.2.3 导入数据 15](#_Toc531704057)

[4.2.4 查看MySQL中user\_log表中的数据 16](#_Toc531704058)

[五、利用Spark预测回头客 16](#_Toc531704059)

[5.1 预处理test.csv和train.csv数据集 16](#_Toc531704060)

[5.1.1 新建predeal\_test.sh脚本文件 16](#_Toc531704061)

[5.1.2 执行predeal\_test.sh脚本 17](#_Toc531704062)

[5.1.3 train.csv文件预处理 17](#_Toc531704063)

[5.1.4 新建predeal\_train.sh脚本文件 17](#_Toc531704064)

[5.1.5 执行predeal\_train.sh脚本文件 18](#_Toc531704065)

[5.2 预测回头客 18](#_Toc531704066)

[5.2.1 启用Hadoop 18](#_Toc531704067)

[5.2.2 进入MySQL Shell 18](#_Toc531704068)

[5.2.3 创建回头客预测表rebuy 18](#_Toc531704069)

[5.2.4 启动Spark Shell 18](#_Toc531704070)

[5.2.5 使用支持向量机SVM分类器预测回头客 19](#_Toc531704071)

[六、利用ECharts进行数据可视化分析 23](#_Toc531704072)

[6.1搭建Tomcat+MySQL+JSP开发环境 24](#_Toc531704073)

[6.1.1 下载Tomcat安装包 24](#_Toc531704074)

[6.1.2 启动MySQL 24](#_Toc531704075)

[6.2利用Eclipse 新建可视化Web应用 24](#_Toc531704076)

[6.2.1 在Eclipse中安装必要的Web组件 24](#_Toc531704077)

[6.2.2 新建Dynamic Web Project 24](#_Toc531704078)

[6.2.3 添加mysql-connector-java-5.1.47-bin.jar连接驱动 25](#_Toc531704079)

[6.2.4 利用Eclipse 开发Dynamic Web Project应用 25](#_Toc531704080)

[6.3启动项目 28](#_Toc531704081)

[6.3.1 启动Server 28](#_Toc531704082)

[6.3.2 在Web Browser浏览运行结果 28](#_Toc531704083)

# 一、运行环境

实际配置环境结合了实际情况，没有和实验案例完全一致，不过整个功能正常实现。实际运行环境及版本如下所示。

Linux: Ubuntu14.04

JDK: Openjdk-1.7.0\_181

Hadoop: 2.7.6

MySQL: 5.7.24

Hive: 1.2.2

Sqoop: 1.4.7

Spark: 2.1.0

Eclipse: 4.5.0

Echarts: 3.8.4

配置过程中截图如下所示，由于步骤较多，仅截取部分关键步骤。

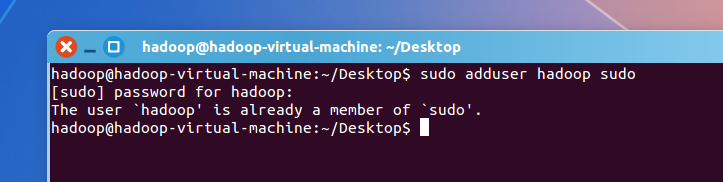


图1.1 创建hadoop用户,添加管理权限

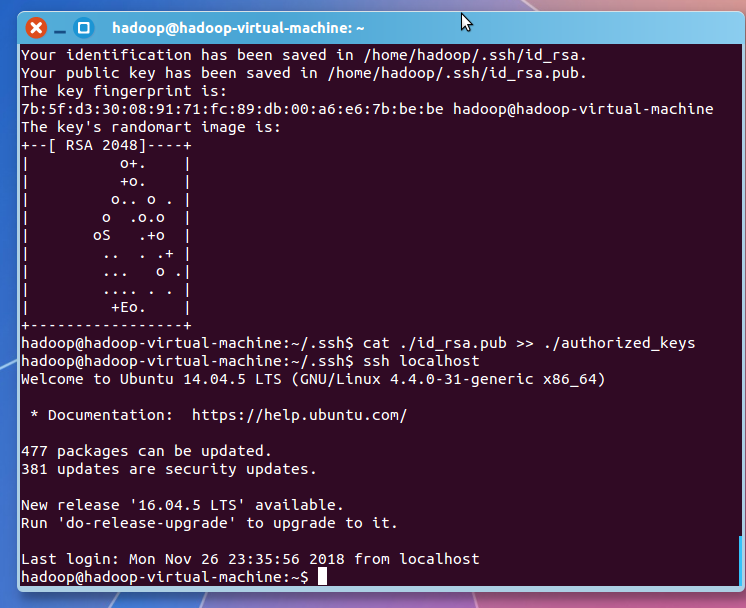


图1.2 安装配置SSH

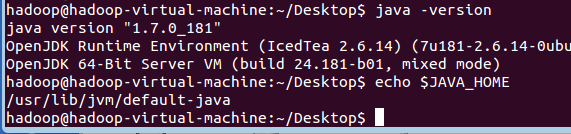


图1.3 配置Java

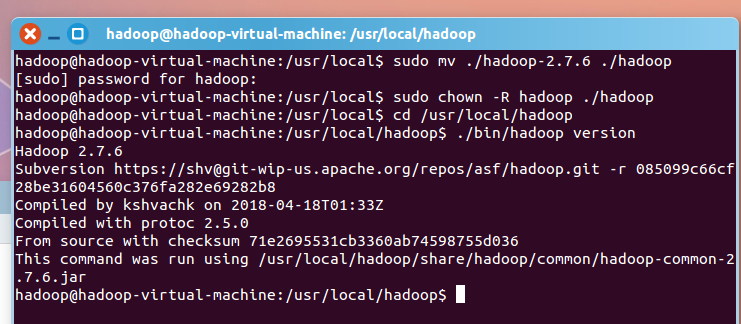


图1.4 配置Hadoop

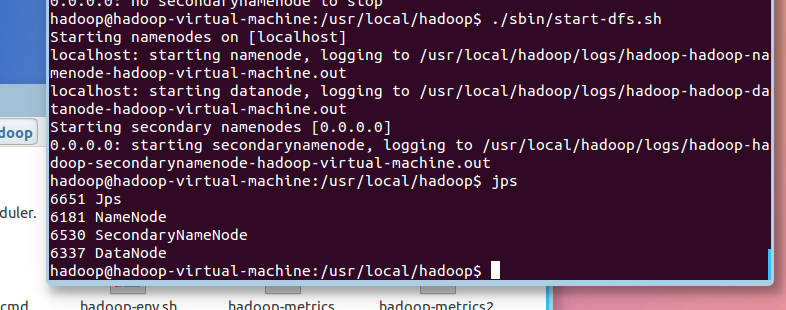


图1.5 运行Hadoop

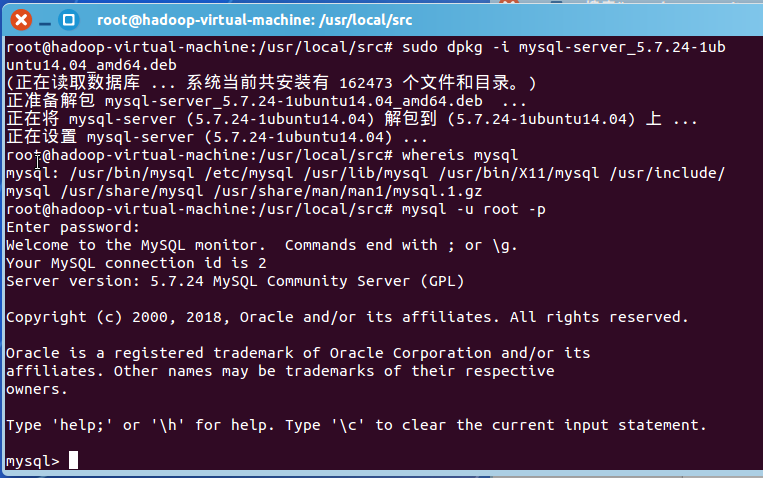


图1.6 配置运行MySQL

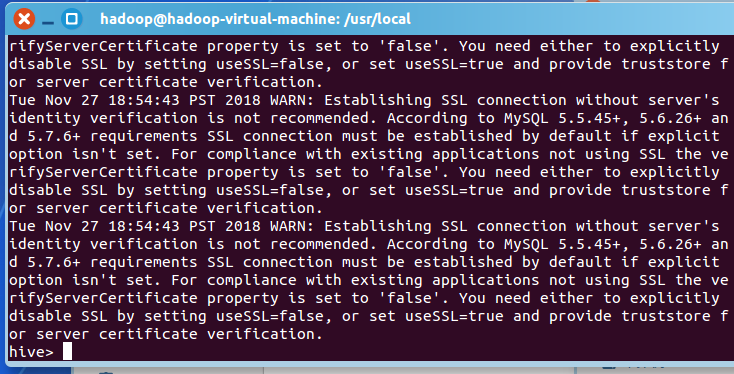


图1.7 配置运行Hive

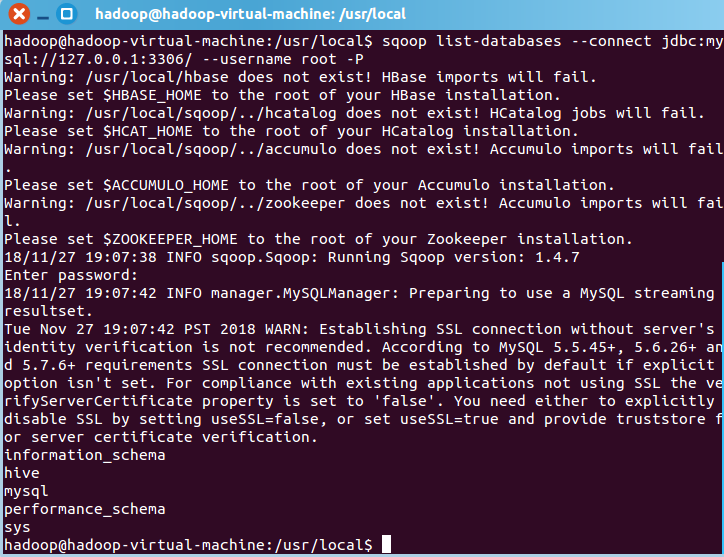


图1.8 配置运行Sqoop

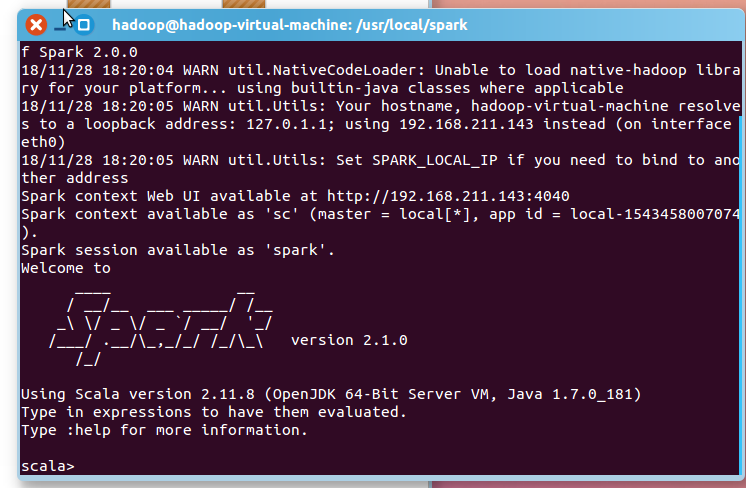


图1.9 配置运行Spark

至此“淘宝双11数据分析与预测课程案例”所需要的环境配置完成。另外实际操作中发现在案例教程中存在一些小问题，比如教程中Eclipse版本为3.8，但是在配置Tomcat Server时又要求配置v8.0版本，然而3.8版本的Eclipse最多仅支持到v7.0版本的Tomcat，所以实际操作时使用了更新的Eclipse版本。

# 二、本地数据集上传到数据仓库Hive

实验数据集有3个文件，分别是用户行为日志文件user\_log.csv、回头客训练集train.csv、回头客测试集test.csv，以下是三个文件的数据格式及说明。

表2.1 user\_log字段定义

|  |  |
| --- | --- |
| 字段名 | 字段含义 |
| user\_id | 买家id |
| item\_id | 商品id |
| cat\_id | 商品类别id |
| merchant\_id | 卖家id |
| brand\_id | 品牌id |
| month | 交易时间:月 |
| day | 交易事件:日 |
| action | 行为,取值范围{0,1,2,3}，0表示点击，1表示加入购物车，2表示购买，3表示关注商品 |
| age\_range | 买家年龄分段：1表示年龄<18，2表示年龄在[18,24]，3表示年龄在[25,29]，4表示年龄在[30,34]，5表示年龄在[35,39]，6表示年龄在[40,49]，7和8表示年龄>=50,0和NULL则表示未知 |
| gender | 性别:0表示女性，1表示男性，2和NULL表示未知 |
| province | 收获地址省份 |

回头客训练集train.csv和回头客测试集test.csv，训练集和测试集拥有相同的字段。

表2.2 user\_log字段定义

|  |  |
| --- | --- |
| 字段名 | 字段含义 |
| user\_id | 买家id |
| age\_range | 买家年龄分段：1表示年龄<18，2表示年龄在[18,24]，3表示年龄在[25,29]，4表示年龄在[30,34]，5表示年龄在[35,39]，6表示年龄在[40,49]，7和8表示年龄>=50,0和NULL则表示未知 |
| gender | 性别:0表示女性，1表示男性，2和NULL表示未知 |
| merchant\_id | 卖家id |
| label | 是否是回头客，0值表示不是回头客，1值表示回头客，-1值表示该用户已经超出我们所需要考虑的预测范围。NULL值只存在测试集，在测试集中表示需要预测的值。 |

## 2.1 数据集的预处理

### 2.1.1 删除文件第一行记录（字段名称)

1）获取数据集并解压，可以看到dataset下有三个文件：test.csv、train.csv、user\_log.csv。

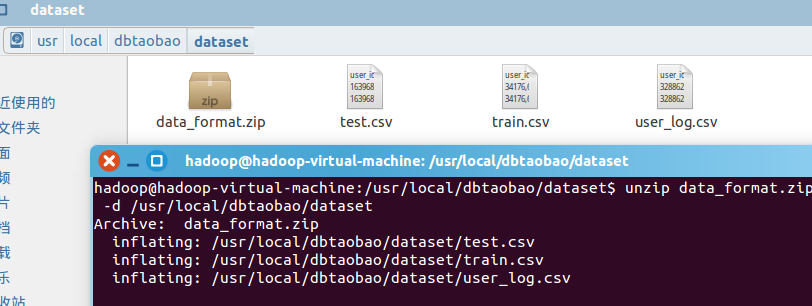


图 2.1

2）查看前五条记录

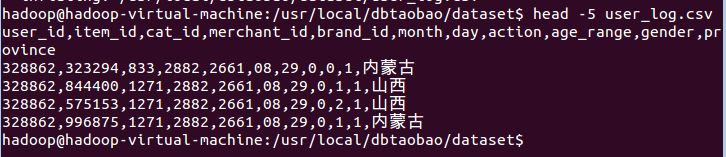


图 2.2

3）删除第一行的字段名称，并确认是否删除

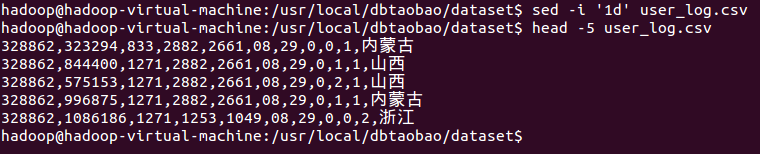


图 2.3

### 2.1.2 获取数据集中前10000条数据

1）建立脚本，编辑predeal.sh脚本文件



图 2.4

2）修改执行权限，执行脚本，输出small\_user\_log.csv

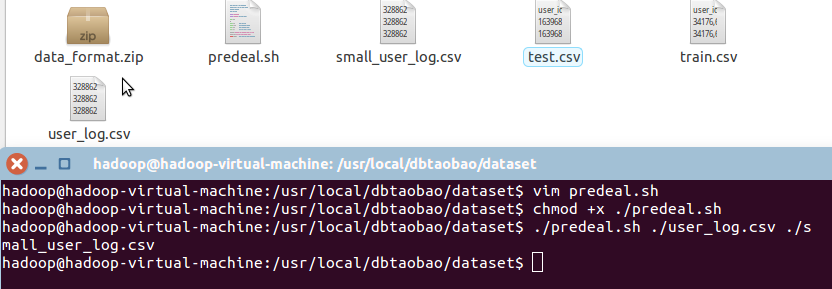


图 2.5

### 2.1.3 把small\_user\_log.csv中的数据导入数据仓库Hive

为此首先把这个文件上传到分布式文件系统HDFS中，然后在Hive中创建两个个外部表，完成数据导入

1）启动Hadoop

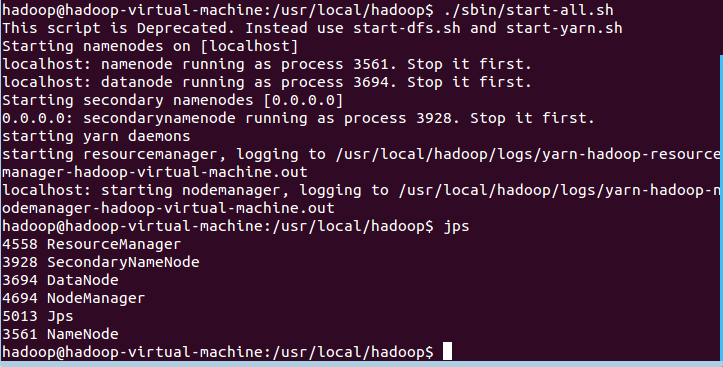


图 2.6

2）把user\_log.csv上传到HDFS中

a.在HDFS根目录下创建子目录



图 2.7

b.上传文件small\_user\_log.csv



图 2.8

c.查看HDFS中的small\_user\_log.csv的前10条记录

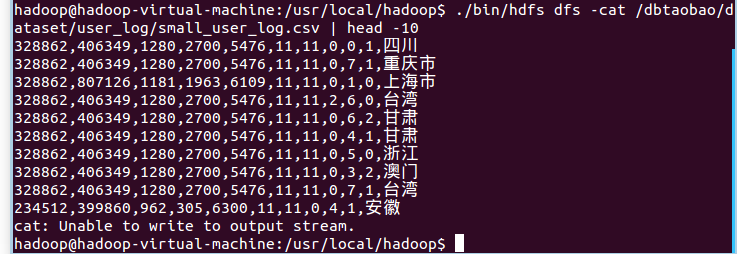


图 2.9

3）在Hive上创建数据库

a.启动MySQL

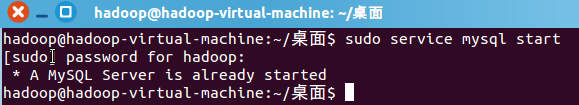


图 2.10

b.启动Hive



图 2.11

这里启动时出现了一个小bug，提示找不到spark-assembly集成包：



图 2.12

这是由于hive版本升级到2.0以上之后，spark-assembly集成包被拆成分散的小jar包导致的，只需要编辑bin/hive，将这个spark-assembly-\*.jar`替换成jars/\*.jar即可。

c.创建数据库dbtaobao

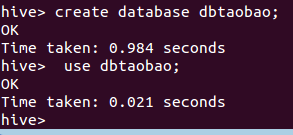


图 2.13

4）创建外部表

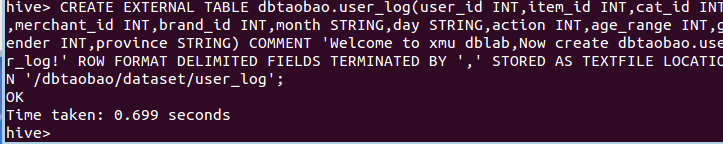


图 2.14

5）查询数据

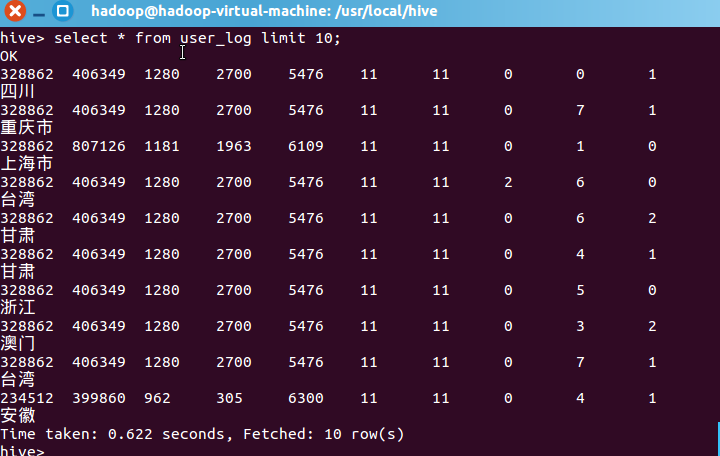


图 2.15

# 三、Hive数据分析

## 3.1 查看user\_log表

进入hive shell，通过show指令查看表的属性和结构。如图3.1。

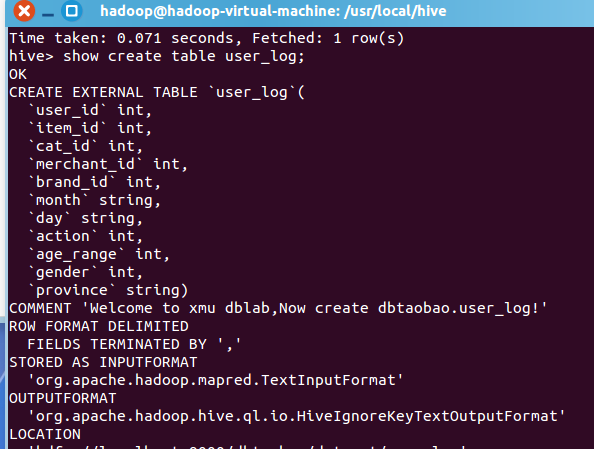


图 3.1

显示结果如图3.2。

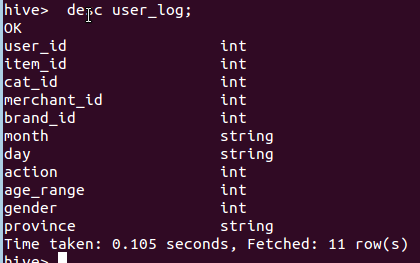


图 3.2

## 3.2 查询条数及统计分析

### 3.2.1 计算表内数据行数

通过select指令结合count指令可以统计记录行数。

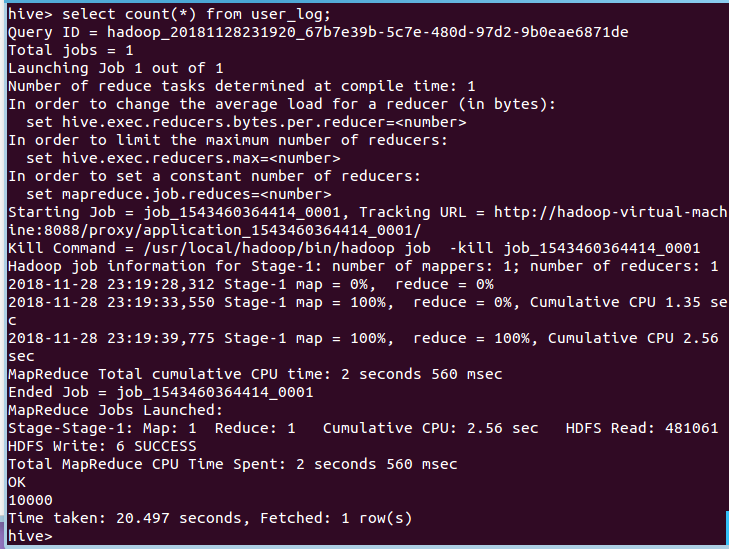


图 3.3

### 3.2.2 查询uid不重复的数据条数

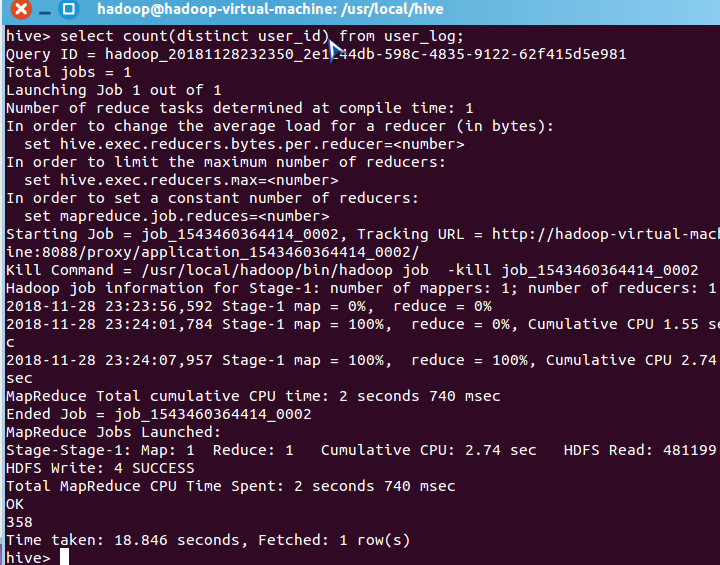


图 3.4

### 3.2.3 查询不重复数据条数

主要为了排除客户刷单情况。

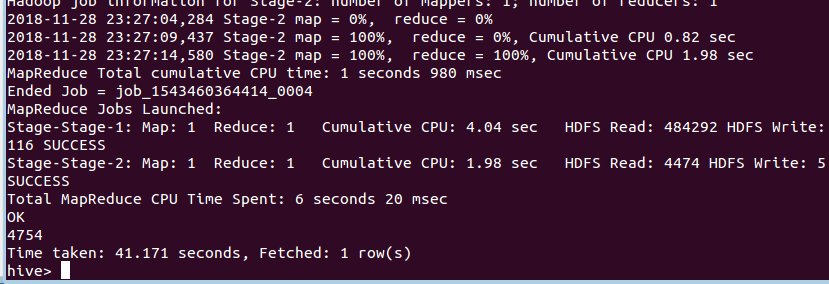


图 3.5

可以看出，排除掉重复信息以后，只有4754条记录。

### 3.2.4 统计男女买家购买商品数量

统计双十一当天男女购买数量，方便后面计算比例。

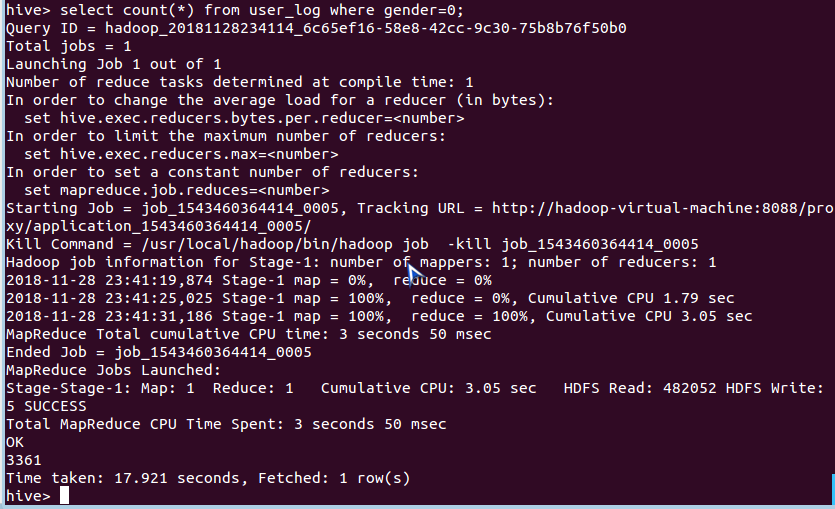


图 3.6 统计男买家

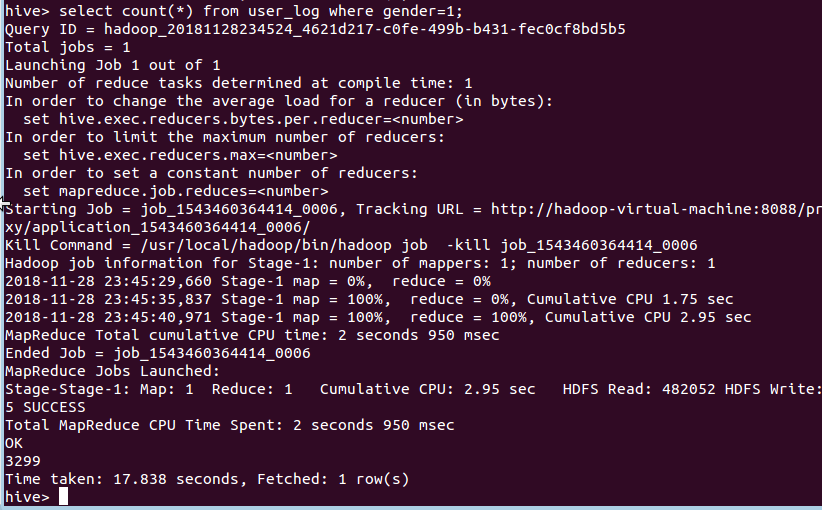


图 3.7 统计女买家

### 3.2.5 查询某一天在网站购买该数量商品的用户id

给定购买商品的数量范围，查询某一天在该网站的购买该数量商品的用户id.

通过group by和having结合，如查询购买商品超过5次的用户id指令：

select user\_id from user\_log where action='2' group by user\_id having count(action='2')>5;

查询结果较长，此处不一一列出。

## 3.3 用户实时查询分析

### 3.3.1 创建新数据表

创建新的数据表scan进行存储，执行结果如图3.8所示。

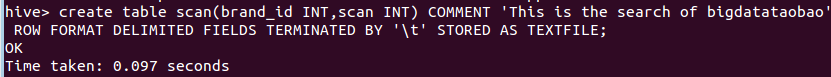


图 3.8

### 3.3.2 导入统计后的数据

通过insert指令导入数据：

hive>insert overwrite table scan select brand\_id,count(action) from user\_log where action='2' group by brand\_id;

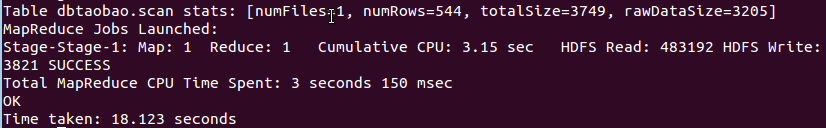


图 3.9

### 3.3.3 显示统计结果

执行select指令：

Hive>select \* from scan;

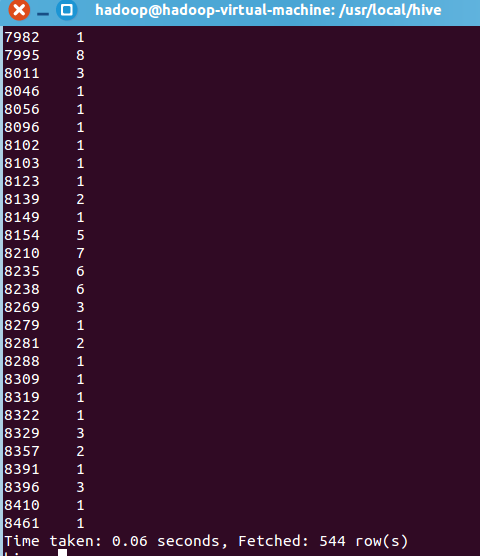


图 3.10

# 四、将数据从Hive导入到MySQL

## 4.1 Hive预操作

### 4.1.1 创建临时表

创建临时表inner\_user\_log和inner\_user\_info.

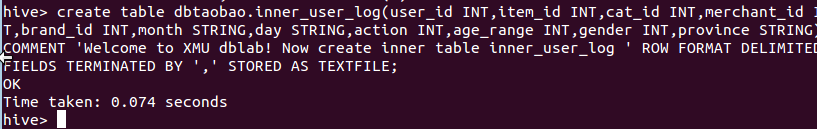


图 4.1

### 4.1.2 数据插入

将user\_log表中的数据插入到inner\_user\_log。

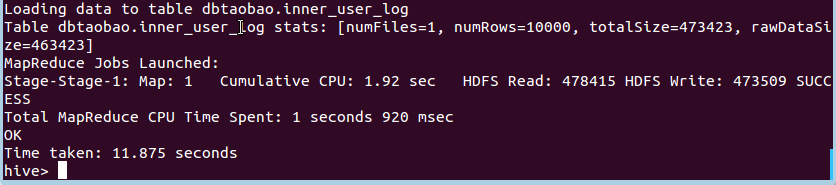


图 4.2

### 4.1.3 查看是否成功执行

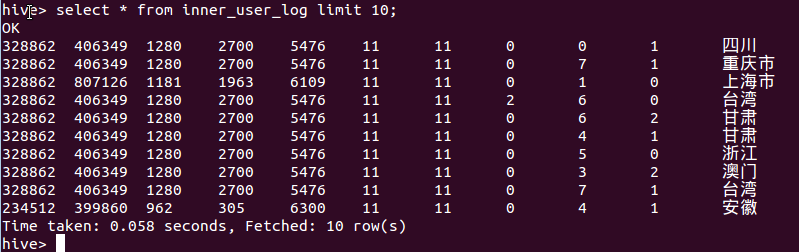


图 4.3

## 4.2 使用Sqoop将数据从Hive导入MySQL

### 4.2.1 创建数据库

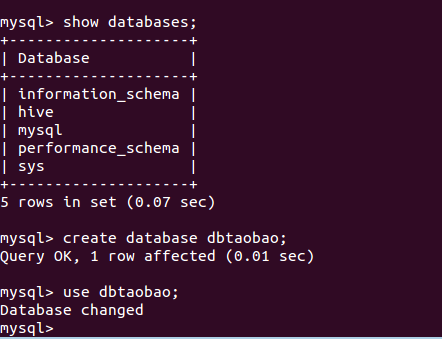


图 4.4

### 4.2.2 创建表

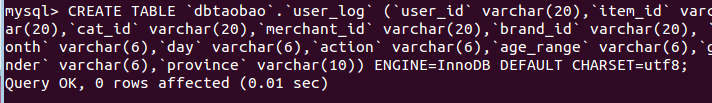


图 4.5

### 4.2.3 导入数据

输入指令如图4.6：

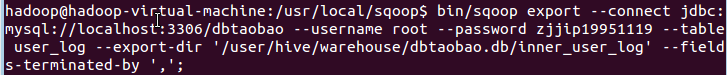


图 4.6

结果输出如图4.7：

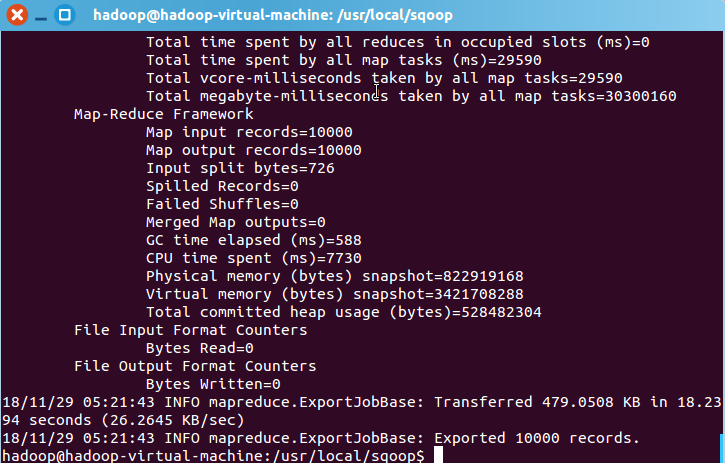


图 4.7

### 4.2.4 查看MySQL中user\_log表中的数据

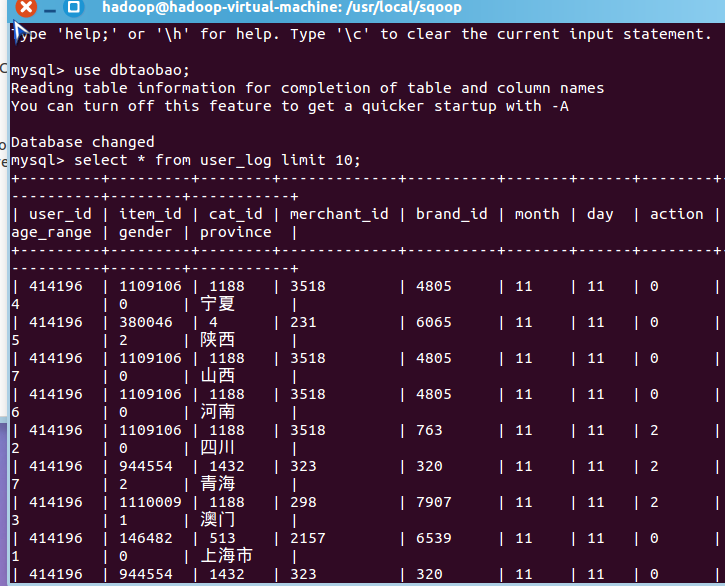


图 4.8

可以看到数据已经成功从Hive导入到MySQL中了。

# 五、利用Spark预测回头客

## 5.1 预处理test.csv和train.csv数据集

### 5.1.1 新建predeal\_test.sh脚本文件

预先处理test.csv数据集，把这test.csv数据集里label字段表示-1值剔除掉,保留需要预测的数据.并假设需要预测的数据中label字段均为1，也就是对应回头客的数据。

其中if($1 && $2 && $3 && $4 && !$5)对应test.csv中的五个字段值。

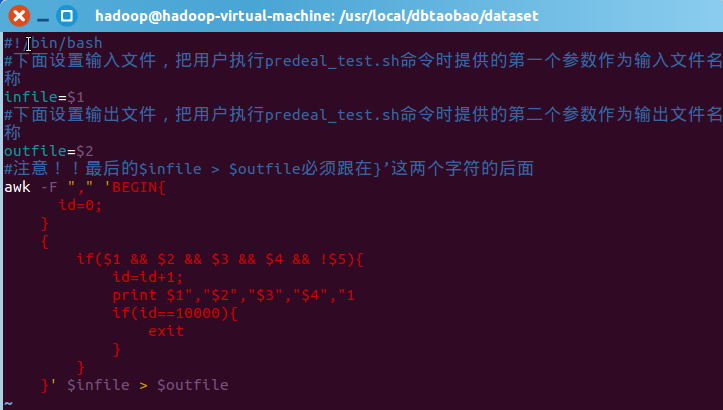


图 5.1

### 5.1.2 执行predeal\_test.sh脚本

执行predeal\_test.sh脚本文件，截取测试数据集需要预测的数据到test\_after.csv文件。



图 5.2

### 5.1.3 train.csv文件预处理

train.csv的第一行都是字段名称，不需要第一行字段名称,这里在对train.csv做数据预处理时，删除第一行



图 5.3

### 5.1.4 新建predeal\_train.sh脚本文件

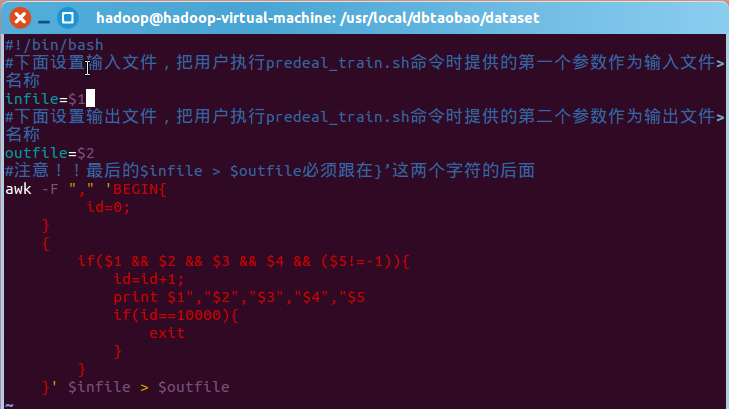


图 5.4

### 5.1.5 执行predeal\_train.sh脚本文件

执行脚本，截取测试数据集需要预测的数据到train\_after.csv文件。

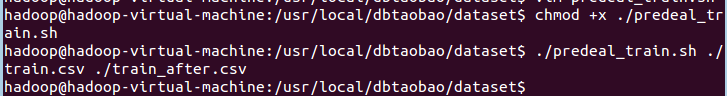


图 5.5

## 5.2 预测回头客

### 5.2.1 启用Hadoop

通过Hadoop将train\_after.csv和test\_after.csv两个数据集分别存取到HDFS中。

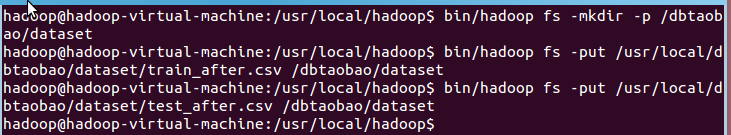


图 5.6

### 5.2.2 进入MySQL Shell

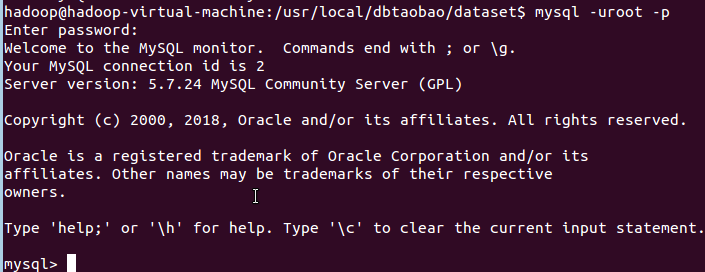


图 5.7

### 5.2.3 创建回头客预测表rebuy

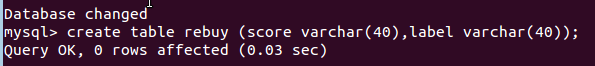


图 5.8

### 5.2.4 启动Spark Shell

通过JDBC方式连接到MySQL数据库获取数据生成DataFrame。

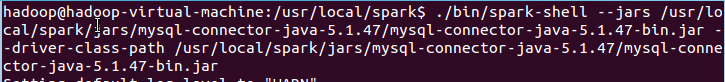


图 5.9

Spark启动成功：

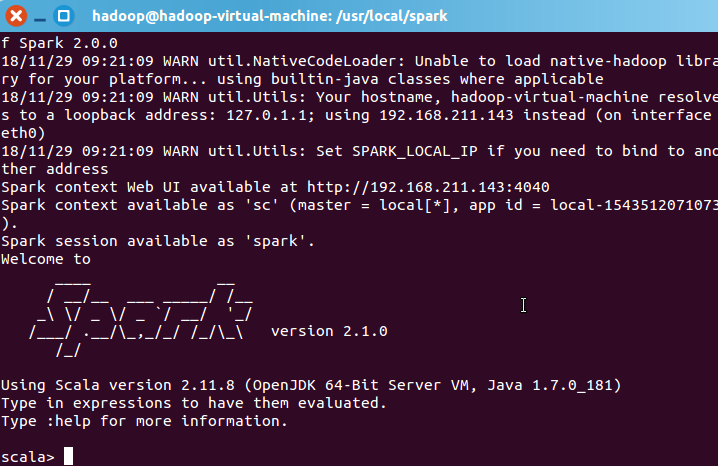


图 5.10

### 5.2.5 使用支持向量机SVM分类器预测回头客

1）导入所需的包

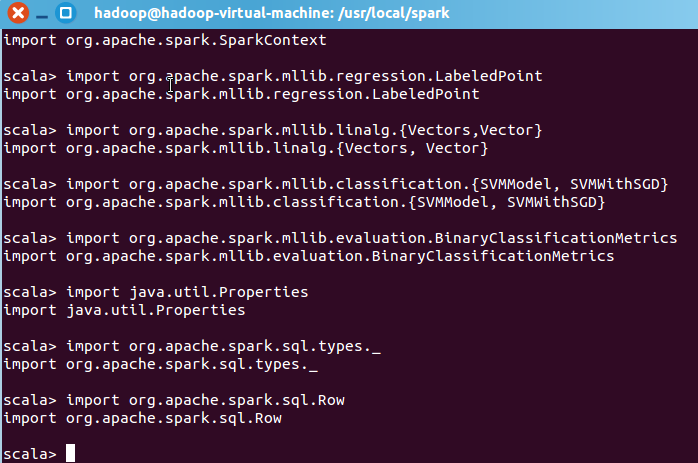


图 5.11

2）读取训练文本文件和测试文本文件

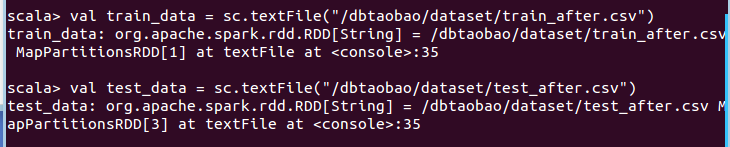


图 5.12

3）构建模型

a.构建数据集

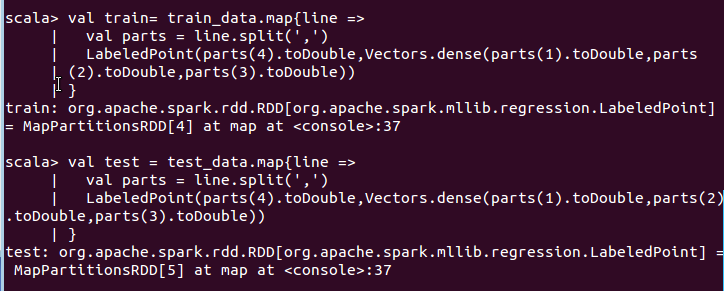


图 5.13

这里通过map将每行的数据用“,”隔开，在数据集中，每行被分成了5部分，前4部分是用户交易的3个特征(age\_range,gender,merchant\_id)，最后一部分是用户交易的分类(label)。把这里我们用LabeledPoint来存储标签列和特征列。LabeledPoint在监督学习中常用来存储标签和特征，其中要求标签的类型是double，特征的类型是Vector。

b.构建模型SVMWithSGD

SGD即随机梯度下降算法，设置迭代次数为1000。生成的模型用model变量保存。

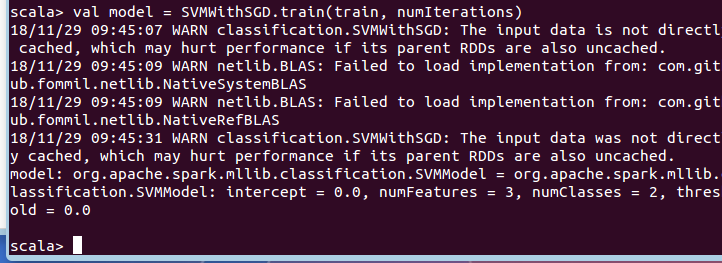


图 5.14

4）评估模型

清除默认阈值，这样会输出原始的预测评分，即带有确信度的结果。

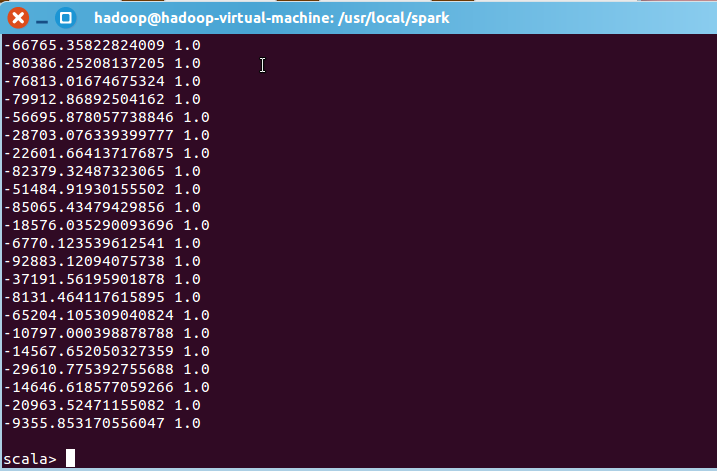


图 5.15

如果我们设定了阀值，则会把大于阈值的结果当成正预测，小于阈值的结果当成负预测。

比如我们设置阈值为0.0：

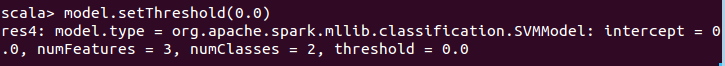


图 5.16

此时输出结果：



图 5.17

5）将结果写入MySQL数据库

在不设置阈值的情况下将测试集结果存入到MySQL数据表rebuy中。通过以下步骤逐步完成。

a.清除阈值



图 5.18

b.将测试数据通过map使用model模型进行预测并存储在scoreAndLabels中

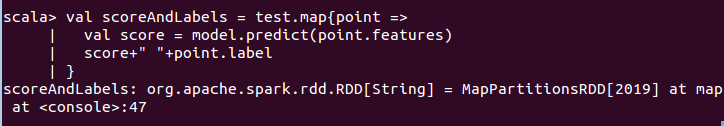


图 5.19

c.设置回头客数据，将数据集用空格分开

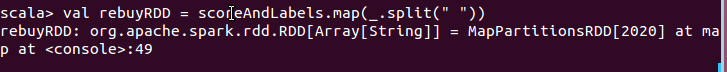


图 5.20

d.设置模式信息

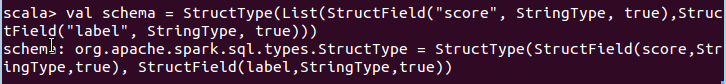


图 5.21

e.创建Row对象，每个Row对象都是rowRDD中的一行



图 5.22

f.建立起Row对象和模式之间的对应关系，也就是把数据rowRDD和模式schema对应起来



图 5.23

g.创建一个prop变量用来保存JDBC连接参数，并设置驱动程序

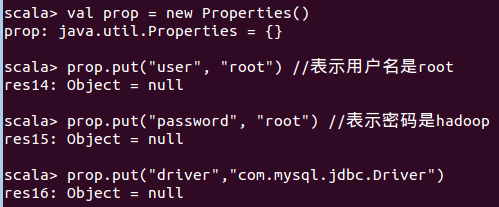


图 5.24

h.连接数据库，采用append模式，表示追加记录到数据库dbtaobao的rebuy表中



图 5.25

i.执行结果

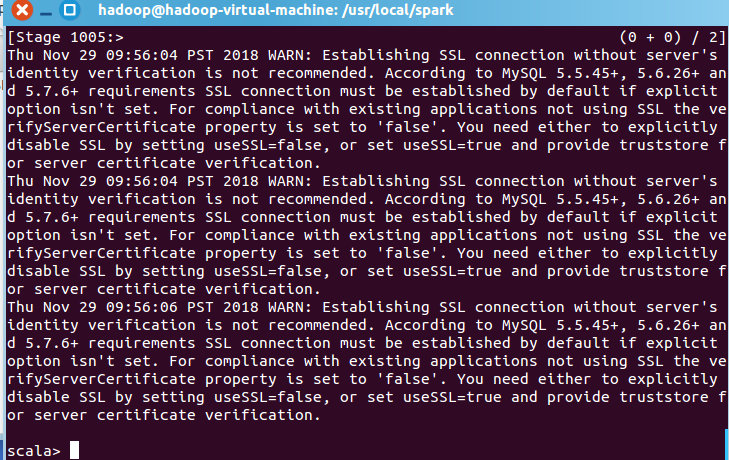


图 5.26

# 六、利用ECharts进行数据可视化分析

ECharts是一个基于Javascript的数据可视化图表库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等），提供直观，生动，可交互，可高度个性化定制的数据可视化图表。

由于ECharts是运行在网页前端，选用JSP作为服务端语言，读取MySQL中的数据，然后渲染到前端页面，因此还需要Tomcat作为Web服务器。

## 6.1搭建Tomcat+MySQL+JSP开发环境

### 6.1.1 下载Tomcat安装包

案例中使用的是v8.0.47，实际操作时选择的Tomcat版本为8.0.53。

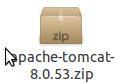


图 6.1

### 6.1.2 启动MySQL



图 6.2

## 6.2利用Eclipse 新建可视化Web应用

### 6.2.1 在Eclipse中安装必要的Web组件

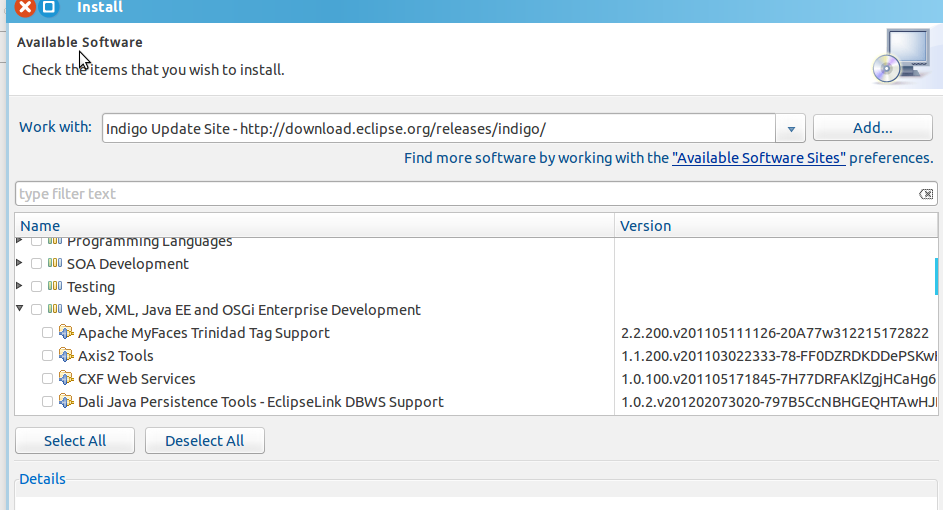


图 6.3

### 6.2.2 新建Dynamic Web Project

配置runtime为Tomcat v8.0。

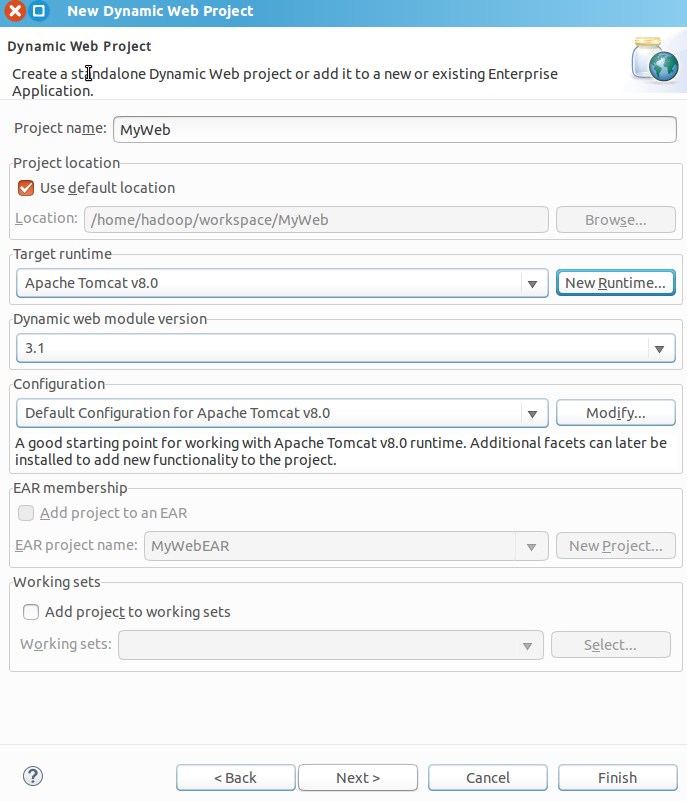


图 6.4

### 6.2.3 添加mysql-connector-java-5.1.47-bin.jar连接驱动



图 6.5

### 6.2.4 利用Eclipse 开发Dynamic Web Project应用

1）构建项目文件和代码，注意MySQL连接参数配置正确

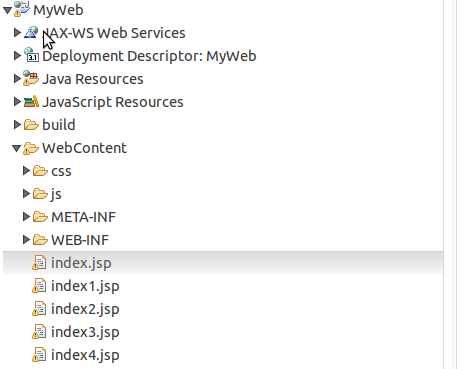


图 6.6

2）关键功能代码：

数据库连接通过jdbc，获取需要的数据则通过select指令配合group by完成。

a.连接MySQL数据库

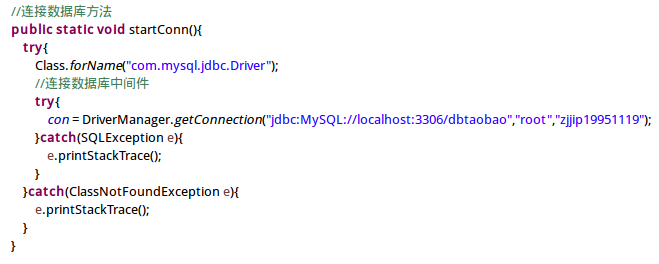


图 6.7

b.所有买家不同消费行为比例

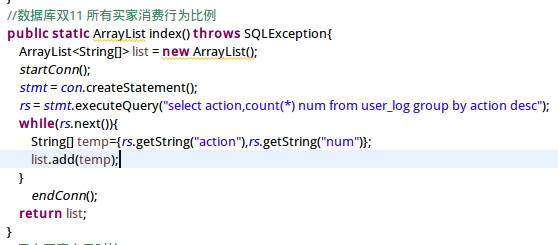


图 6.8

c.男女买家交易量对比

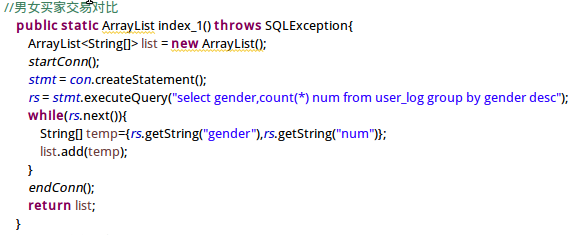


图 6.9

d.男女买家各个年龄段交易对比

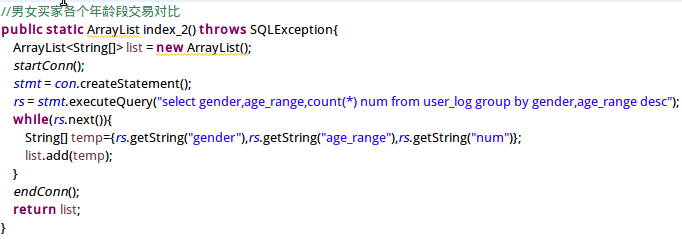


图 6.10

e.销量前十的商品类别对比

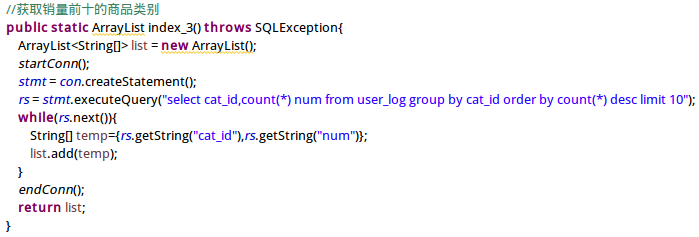


图 6.11

f.各年龄段交易比例

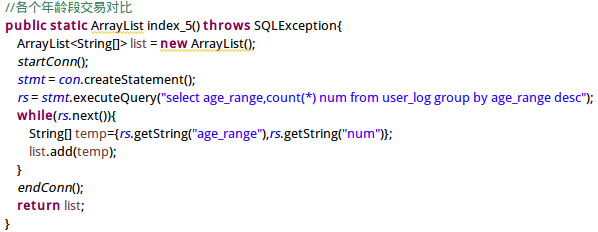


图 6.12

g.各个省份的的总成交量对比

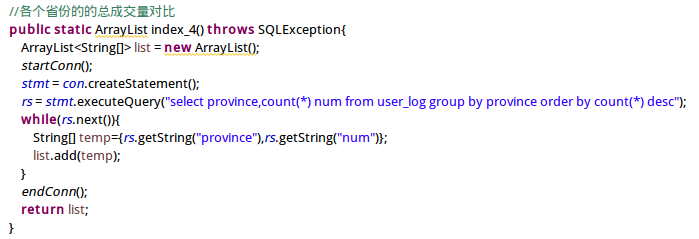


图 6.13

## 6.3启动项目

### 6.3.1 启动Server

双击index.jsp，选择run as->run on server开始运行项目。

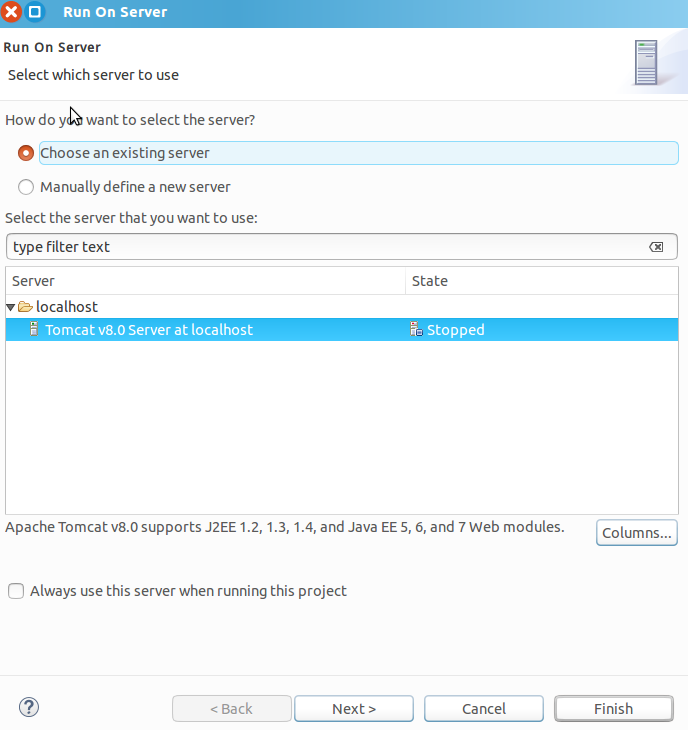


图 6.14

### 6.3.2 在Web Browser浏览运行结果

这里进行了五种统计，分别展示了所有买家消费比例、男女买家交易量比例、男女买家各年龄段交易对比、各年龄段交易比例、销量前十商品对比、全国各省成交量分布。

其中销量前十商品图相比案例中稍作了一点修改，各年龄段交易比例和全国各省成交量分布则是自己实现的，所以着重写一下这3个。

1）所有买家消费行为比例图

使用饼图进行展示，通过查询action字段获取，包含点击、添加购物车、特别关注、购买四种行为，其中点击比例达到八成，占绝大多数。



图 6.15

2）男女买家交易对比图

使用饼图展示，通过查询gender字段获取，包含男性、女性、未知三种分布。

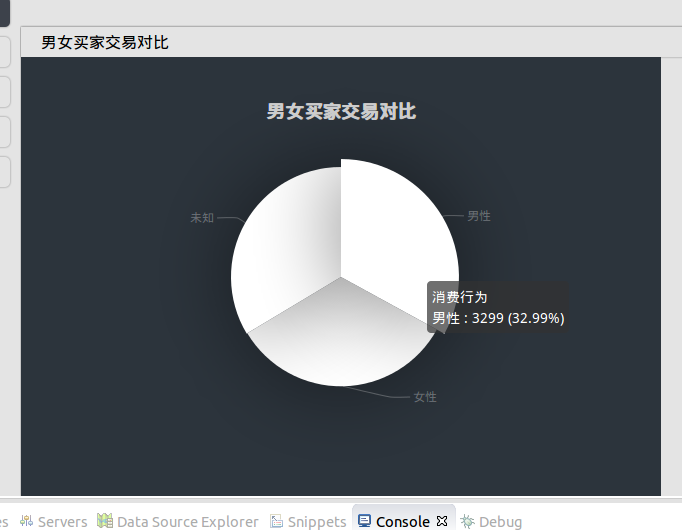


图 6.16

3）男女买家各个年龄段交易对比图

使用散点图展示，并将男性和女性用两种颜色的图例区分开，通过查询gender字段、age\_range字段获取，横轴是不同年龄段，纵轴是交易记录数。

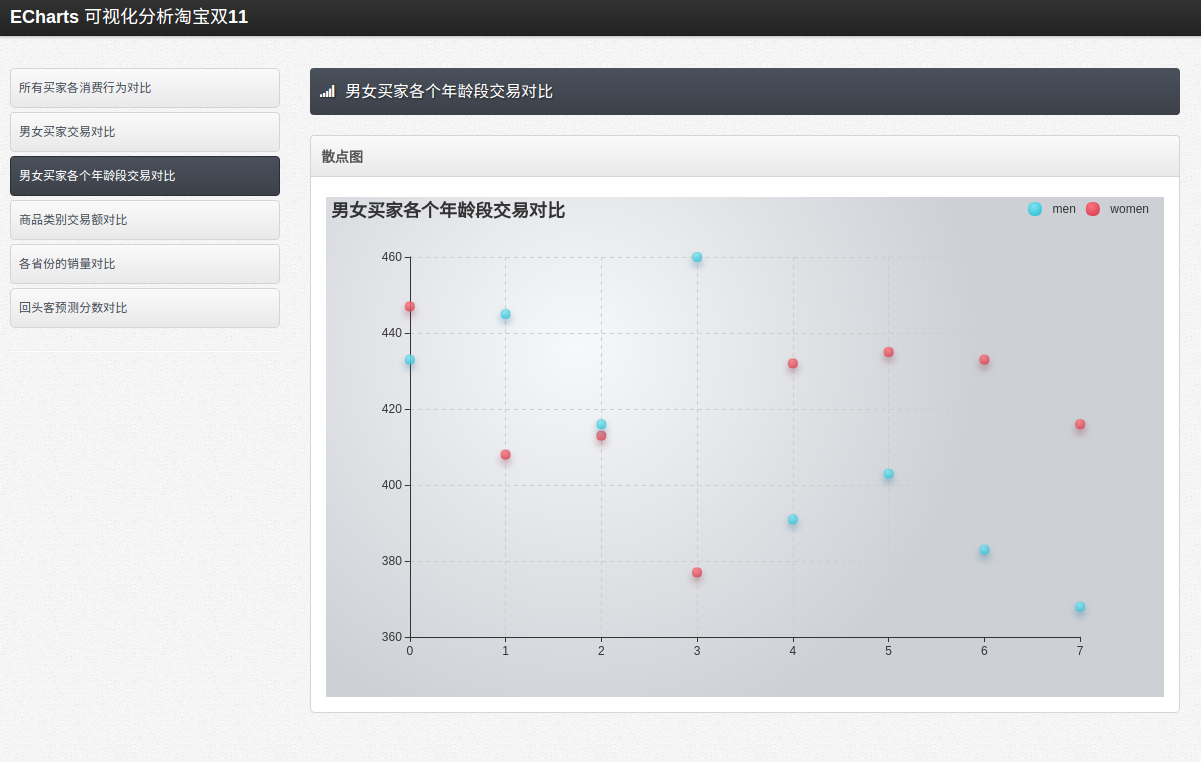


图 6.17

4）销售前十商品类别图

使用柱状图展示，案例教程中是只获取了销售前五的商品数据，这里为了对比更加清晰鲜明，获取的是前十的数据，在柱状图上方加了图例，并将数据按降序全部展示出来，数据通过cat\_id字段结合select、group by、limit等指令完成：

select cat\_id,count(\*) num from user\_log group by cat\_id order by count(\*) desc limit 10；

横轴展示的是商品id，纵轴是销售量。可以看到第一的商品和第十的商品销量差距接近一倍，还是差别挺大的。

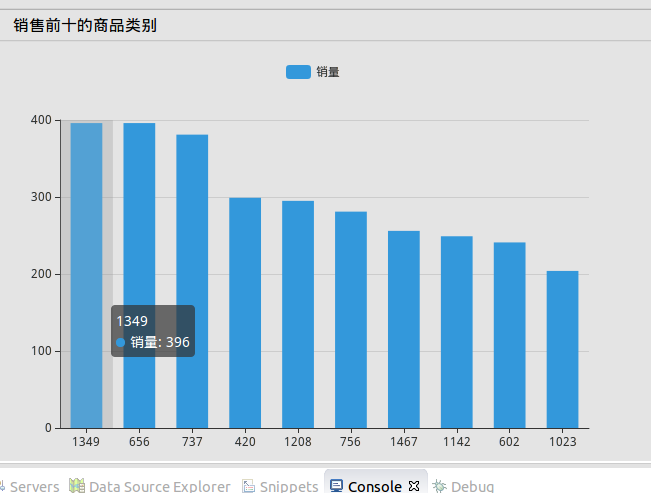


图 6.18

5）各年龄段交易比例

这个是自己根据需求实现的，案例教程中没有。主要目的是想看一下买家中什么年龄段在双十一那天最活跃，因而选择了环形图，比较清晰。获取的方式也是通过age\_range字段：

select age\_range,count(\*) num from user\_log group by age\_range desc;

由于截取的数据记录量只有10000条，所以差距不是很大，不过也可以看出18~24岁年龄段相对更活跃，这也和大学生消费需求较大有关；另外25~29岁年龄段也很活跃，这可能因为这部分人都工作了几年，有了一定的消费力，同时又比较年轻，接受能力强，所以消费力比较高。

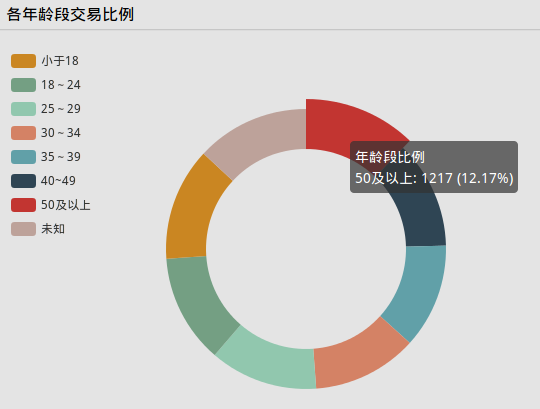


图 6.19

6）各省份成交量对比

案例教程中有提到这个分布图，不过没有给出代码，此处自己进行了实现。主要目的是看一下全国双十一活跃度分布情况，通过province字段获取数据。

首先左下角是一个连续型的visualMap视觉映射组件，代码中会实时根据数据的分布改变max值和min值。另外为了和china.js的地图数据匹配，代码中也将四个直辖市（上海市、重庆市、北京市、天津市）中的“市”截掉，从而与地图中的（上海、重庆、北京、天津）匹配。

关键代码如下：



图 6.20

通过分布看出，南方的活跃度还是相对高于北方，沿海活跃度相对高于内陆，这可能和省市的经济发展状况以及人们的平均生活水平有关。

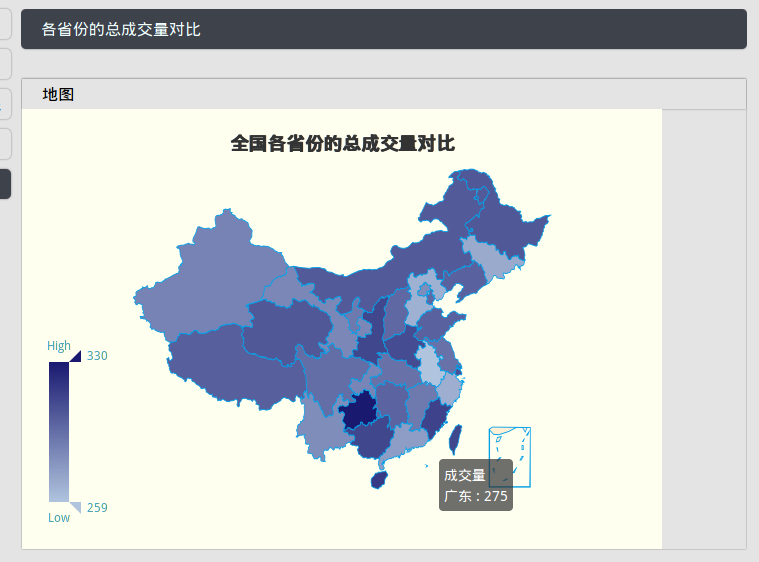


图 6.21