

菜鸟-需求预测与分仓规划

成员:

崔绿叶 2120150981

陈帅 2120150979

贺辉 2120150991

目前进展:

对给定数据建立相应数据库并进行分析处理,提取需要的数据特征,完成数据的可视化,对数据使用预测模型进行初步预测。

目前成果:

关于数据特征,我们使用了商品在全国和地区的“非聚划算支付件数”(qty_alipay_njhs)这一特征;对数据进行处理时,我们首先进行了数据库的建立、链接操作,这样对于数据中所存在的一些问题,比如重复数据等,我们直接使用 SQL 语言对其进行处理;经过这样的处理后,可以保证我们得到的数据更加有效,其清理具体过程如下:

(1) 创建 tianchi 数据库:

```
create database tianchi;
```

(2) 将测试数据导入数据库:

两张表: 表 1.item_feature1

表 2.item_store_feature1

(3) 使用 compare 函数比较导出出来的数据是否和原始数据相同,尤其是 Double 类型的字段,有可能从小数点之后就已被截断。

(4) 查看是否有重复记录,确保记录没有重复导入:

```
select distinct * from item_store_feature1;
```

```
select distinct * from item_feature1;
```

(5) 查询商品的种类(1000),用 item_id 字段唯一确认一个商品:

```
select distinct item_id from item_feature1;
```

```
select distinct item_id, cate_id from item_feature1;
```

```
select distinct item_id, cate_id, cate_level_id, brand_id from  
item_feature1;
```

```
select distinct
```

```
item_id, cate_id, cate_level_id, brand_id, supplier_id from  
item_feature1;
```

这四个查询结果相同,再确认一下:

```
create tmp select
```

```
item_id, cate_id, cate_level_id, brand_id, supplier_id from  
item_feature1;
```

```
create tmp1 select item_id, cate_id, cate_level_id, brand_id  
from item_feature1;
```

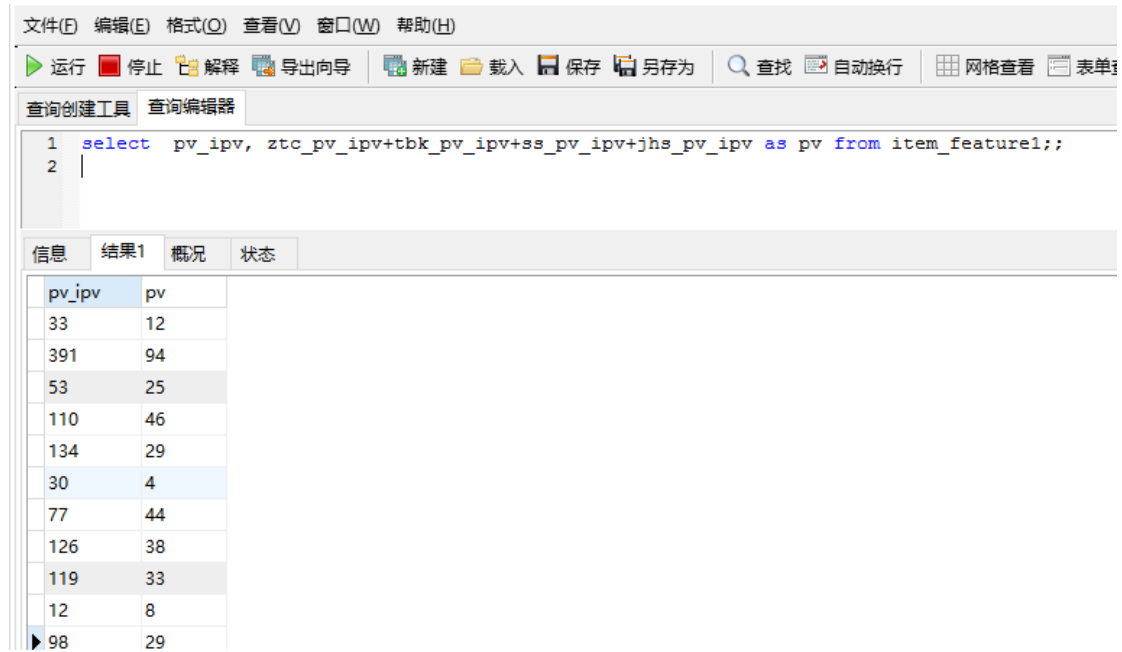
```
select DISTINCT * from tmp;
```

```
select DISTINCT * from tmp1;
```

结果证明可以用 item_id 字段唯一确认一个商品。

(6) 浏览次数字段不是其他四个浏览次数字段的简单相加，即 pv_ipv 和 pv 并不相等

```
select pv_ipv, ztc_pv_ipv+tbk_pv_ipv+ss_pv_ipv+jhs_pv_ipv as pv
from item_feature1;
```



The screenshot shows a SQL query editor with the following query:

```
1 select pv_ipv, ztc_pv_ipv+tbk_pv_ipv+ss_pv_ipv+jhs_pv_ipv as pv from item_feature1;;
2 |
```

The results table is as follows:

pv_ipv	pv
33	12
391	94
53	25
110	46
134	29
30	4
77	44
126	38
119	33
12	8
98	29

(7) 非聚划算的浏览次数计算公式为：njhs_pv_ipv==pv_ipv-jhs_pv_ipv（浏览次数-聚划算浏览次数）

```
alter table item_feature1 add njhs_pv_ipv Double;
alter table item_store_feature1 add njhs_pv_ipv Double;
update item_feature1 set where
njhs_pv_ipv=item_feature1.pv_ipv&item_feature1.jhs_pv_ipv;
update item_store_feature1 set where
njhs_pv_ipv=item_store_feature1.pv_ipv-
item_store_feature1.jhs_pv_ipv.
```

(8) 探索拍下件数、成交件数和非聚划算成交件数的关系

```
select qty_alipay, qty_alipay_njhs from item_feature1 where
qty_alipay!= qty_alipay_njhs
```

查询结果不为 None，所以成交件数和非聚划算成交件数这两个字段不相同

运行 停止 解释 导出向导 新建 载入 保存 另存为 查找 自动换行 网络查看 表单查看 备注

查询创建工具 查询编辑器

```
1 select qty_alipay, qty_alipay_njhs from item_feature1 where qty_alipay!= qty_alipay_njhs
```

信息 结果1 概况 状态

qty_alipay	qty_alipay_njhs
393	60
130	86
88	67
2528	2122
90	76
150	108
173	80
117	96
265	216
306	88

select qty_gmv, qty_alipay, qty_alipay_njhs from item_feature1
 从结果看：拍下件数>成交件数

运行 停止 解释 导出向导 新建 载入 保存 另存为 查找 自动换行

查询创建工具 查询编辑器

```
1 select qty_gmv, qty_alipay, qty_alipay_njhs from item_feature1
```

信息 结果1 概况 状态

qty_gmv	qty_alipay	qty_alipay_njhs
0	0	0
40	33	33
0	1	1
8	4	4
13	12	12
2	2	2
4	1	1
10	8	8
5	5	5
1	1	1
9	9	9
2	2	2
6	5	5
5	4	4
3	2	2
6	6	6
2	1	1

(9) 提取特征建立新表，方便处理：

```
create table 01_item_feature select
date, item_id, qty_alipay, qty_alipay_njhs, pv_ipv, jhs_pv_ipv, njhs_pv_ipv
, collect_uv
from item_feature1;
```

```
create table 01_item_store_feature select
date, item_id, store_code, qty_alipay, qty_alipay_njhs, pv_ipv, jhs_pv_ipv,
njhs_pv_ipv, collect_uv
from item_store_feature1.
```

(10) 检查数据的一致性:

```
select sum(qty_alipay_njhs) from 01_item_feature;
select sum(qty_alipay_njhs) from 01_item_store_feature;
```

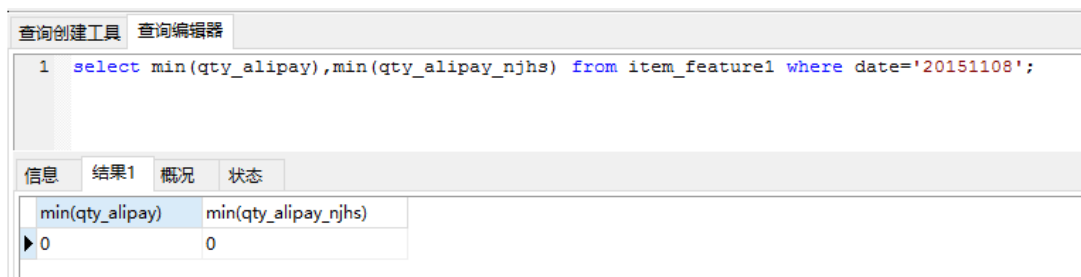
结果表明检查结果一致

(11) 潜在问题: 有些商品在当天没有行为记录, 既没有浏览记录也没有交易记录: 有些商品可能中途上架; 可能中途上架, 上架后又下架, 下架之后又上架, 比如衣服有季节性销售的倾向, 导致数据比较稀疏。

```
select min(qty_alipay), min(qty_alipay_njhs) from item_feature1
where date='20151108';
```

(12) 数据可视化查看:

首先, 计算相隔天数为 443, 总共 “444” 天数据



```
select DATEDIFF('20151227', '20141010')
```

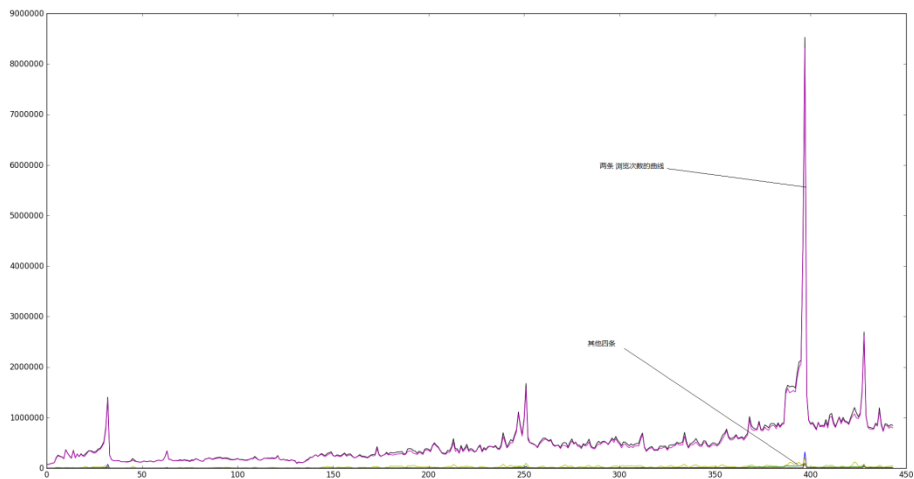
运行 停止 解释 导出向导 新建 载入 保存 另存



其次, 统计所有商品在时间轴上的分布

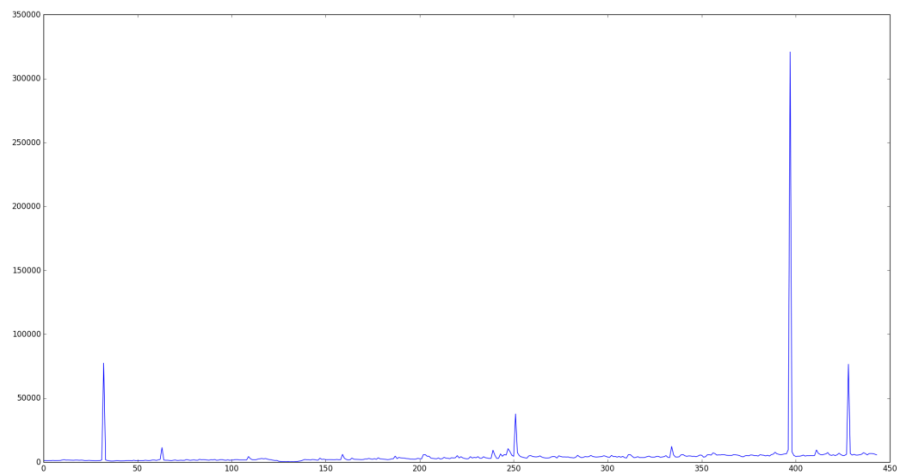
```
create table stat_figure1 select
date, sum(qty_alipay), sum(qty_alipay_njhs), sum(pv_ipv), sum(jhs_pv_ipv),
sum(njhs_pv_ipv), sum(collect_uv) from 01_item_feature group by
date;
```

横坐标为查询结果的第一列 (即日期), 其他列各对应一条曲线

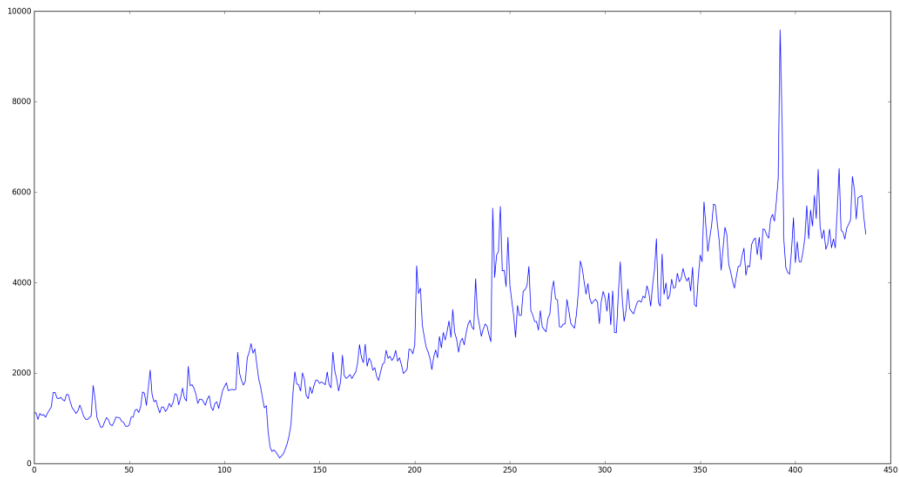


根据图标发现，在 2014. 11. 11, 2014. 12. 12, 2015. 6. 14, 2015. 6. 18, 2015. 11. 10, 2015. 11. 11, 2015. 12. 12 这几个时间段因为促销活动而有很大波动。我们可以去掉这几个点，或者用前后订单的平均值代替这些天地额订单数而做一个平滑处理或者拿出来单独处理：

单独拿出 `sum(qty_alipay_njhs)`，如下图：



去掉 2014. 11. 11, 2014. 12. 12, 2015. 6. 14, 2015. 6. 18, 2015. 11. 10, 2015. 11. 11, 2015. 12. 12 这几个点，画出 `sum(qty_alipay_njhs)`，如下图：



最后，统计每个商品的数量分布，结果发现数据很稀疏：

```
select
item_id, sum(qty_alipay), sum(qty_alipay_njhs), sum(pv_ipv), sum(jhs_pv_ipv), sum(njhs_pv_ipv), sum(collect_uv) from 01_item_feature group by
item_id;
```

1 `select item_id, sum(qty_alipay), sum(qty_alipay_njhs), sum(pv_ipv), sum(jhs_pv_ipv), sum(njhs_pv_ipv), sum(collect_uv) from 01_item_feature group by item_id;`

数据集中 非聚划算支付总件数

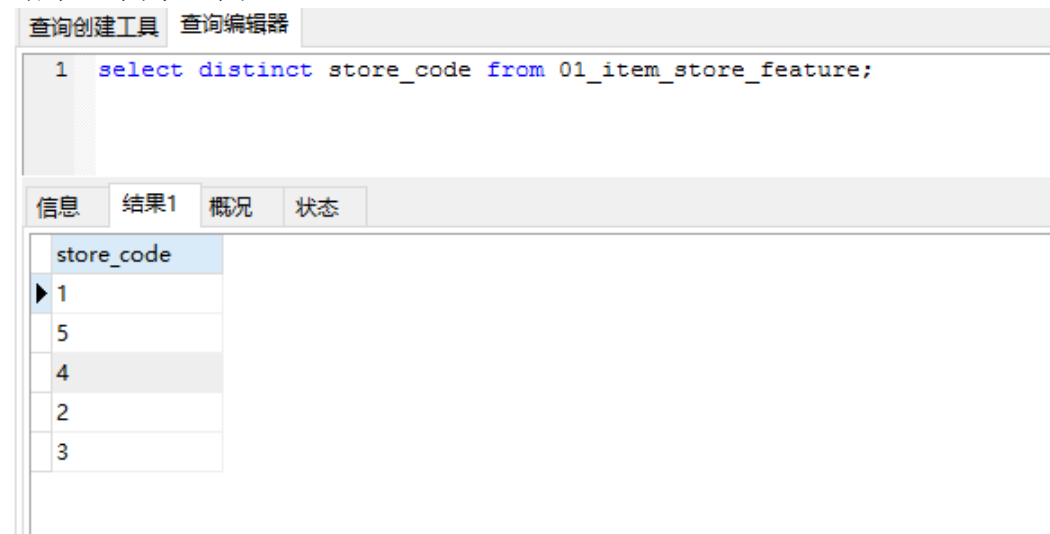
信息	结果1	概况	状态				
item_id	sum(qty_alipay)	sum(qty_alipay_njhs)	sum(pv_ipv)	sum(jhs_pv_ipv)	sum(njhs_pv_ipv)	sum(collect_uv)	
100015	57	57	4311	0	4311	293	
100060	29	29	1474	0	1474	81	
100126	26	26	1104	0	1104	40	
100267	1709	1709	78593	4	78589	3282	
100840	165	127	24353	5909	18444	399	
101615	8356	4262	958732	119033	839699	17714	
101896	111	111	52791	10	52781	669	
102094	503	497	30389	272	30117	1171	
102362	2006	2006	1284838	58	1284780	23637	
102476	3650	278	575006	23652	551354	9421	
102478	68	68	2560	0	2560	409	
102811	181	181	6895	0	6895	185	
102955	7777	3520	487655	18247	469408	7509	
103184	464	464	16775	0	16775	427	
103198	88	88	21598	0	21598	832	
10325	1471	1471	83742	11	83731	1773	
103261	16	16	324	0	324	11	
103267	390	390	15845	0	15845	937	
103330	811	811	233743	4	233739	9443	
103390	512	356	90147	1997	88150	2092	
103580	1919	1919	94976	0	94976	1854	
103675	79	79	47894	3	47891	858	
103855	542	542	7410	1	7409	446	

(13) 分仓的数据表处理：

首先检查一下全国一共有几个仓库

```
select distinct store_code from 01_item_store_feature;
```

结果显示为 5 个;



其次，对某个商品的销售分布情况进行统计：

```
select  
date, item_id, store_code, qty_alipay, qty_alipay_njhs, pv_ipv, jhs_pv_ipv,  
njhs_pv_ipv, collect_uv from 01_item_store_feature where  
item_id=30378 order by date;
```

然后，对某个商品的各仓库销售总量进行统计：

```
select  
item_id, store_code, sum(qty_alipay), sum(qty_alipay_njhs), sum(pv_ipv), s  
um(jhs_pv_ipv), sum(njhs_pv_ipv), sum(collect_uv) from  
01_item_store_feature where item_id=30378 group by store_code;
```

最后，查看某一个商品在售时间段（有浏览记录的日期，并不是销售量的日期）

```
select min(date), max(date) from 01_item_store_feature where  
item_id=30378 ;
```

下一步工作：

进一步分析处理数据，优化 ARIMA 预测模型进行预测。