

数据挖掘项目报告

崔绿叶 2120150981

贺辉 2120150991

陈帅 2120150979

July 14, 2016

简介

项目要求以历史一年海量买家和卖家的数据为依据，预测某商品在未来二周全国和区域性需求量。我们需要用数据挖掘技术和方法精准刻画商品需求的变动规律，对未来的全国和区域性需求量进行预测，同时考虑到未来的不确定性对物流成本的影响，做到全局的最优化。更精确的需求预测，能够大大地优化运营成本，降低收货时效，提升整个社会的供应链物流效率。问题的具体目标是根据现有的从20141010到20151227 的全国和区域分仓数据，预测给出后面两周（20151228-20160110）的全国和区域分仓目标的库存量。

该问题属于典型的时间序列分析问题，时间序列分析是指某一指标在一定周期内各个时间点上的数值，按时间先后顺序排列成数列，并对此数列进行平稳化检验、差分、建模、诊断等过程后预测该指标在未来的发展趋势。我们组利用所给数据建立相应数据库，分析数据变化趋势并对其进行清理，最后应用自动回归移动平均混合模型（ARIMA 模型）预测之后两周的全国和区域分仓目标库存量。

1 问题陈述

1.1 问题提出

高质量的商品需求预测是供应链管理的基础和核心功能。本赛题以历史一年海量买家和卖家的数据为依据，要求参赛者预测某商品在未来二周全国和区域性需求量。选手们需要用数据挖掘技术和方法精准刻画商品需求的变动规律，对未来的全国和区域性需求量进行预测，同时考虑到未来的不确定性对物流成本的影响，做到全局的最优化。更精确的需求预测，能够大大地优化运营成本，降低收货时效，提升整个社会的供应链物流效率。

1.2 问题目标

根据现有的从20141010到20151227的全国和区域分仓数据，预测给出后面两周（20151228-20160110）的全国和区域分仓目标的库存量。

1.3 数据说明

赛题数据：提供商品从20141010到20151227的全国和区域分仓数据（Table1-Table3）。商品在全国的特征包括商品的本身的一些分类：类目、品牌等，还有历史用户行为特征：浏览人数、加购物车人数，购买人数。注意要预测的未来需求是“非聚划算支付件数” (*qty_alipay_njhs*)。

1.4 预测目标表

参赛者需要提供每个商品的全国和分仓区域的未来两周(20151228-20160110)目标库存（如Table 4所示）。 *cn_submit*: 每个商品在全国和分仓区域的目标库存。

1.5 评测指标

在本赛题中，参赛者需要提供对于每个商品在未来两周的全国最优目标库存和分仓区域最优目标库存的预测。我们会提供每一个商品的补少成本(A)和补多成本(B)，然后根据用户预测的目标库存值跟实际的需求的差异来计算总的成本。参赛者的目标是让总的成本最低。

我们定义以下变量：

T_i : 商品*i* 的全国目标库存（参赛者提供） T_{ia} : 商品*i* 在分仓区域 a 的目标库存（参赛者提供） D_i : 商品*i* 的未来全国实际销量（不提供给参赛者） T_{ia} : 商品*i* 的未来在分仓区域 a 的实际销量（不提供给参赛者） A_i : 商品*i* 的全国补少货的成本 A_{ia} : 商品*i* 在分仓区域 a 的补少货的成本 B_i : 商品*i* 的全国补多货的成本 B_{ia} : 商品*i* 在分仓区域 a 的补多货的成本

全国范围内的成本计算如下：

$$C_N = \sum_i [A_i * \max(D_i - T_i, 0) + B_i * \max(T_i - D_i, 0)] \quad (1)$$

分仓区域内的成本计算如下：

$$C_R = \sum_{ia} [A_{ia} * \max(D_{ia} - T_{ia}, 0) + B_{ia} * \max(T_{ia} - D_{ia}, 0)] \quad (2)$$

总的衡量标准是上面两者的相加：

$$C = C_N + C_R \quad (3)$$

字段	类型	含义	示例
<i>date</i>	bigint	日期	20150912
<i>item_id</i>	bigint	商品ID	132
<i>cate_id</i>	bigint	叶子类目ID	18
<i>cate_level_id</i>	bigint	大类目ID	12
<i>brand_id</i>	bigint	品牌ID	203
<i>supplier_id</i>	bigint	供应商ID	1976
<i>pv_ipv</i>	bigint	浏览次数	2
<i>pv_uv</i>	bigint	流量UV	2
<i>cart_ipv</i>	bigint	被加购次数	0
<i>cart_uv</i>	bigint	加购人次	0
<i>collect_uv</i>	bigint	收藏夹人次	0
<i>num_gmv</i>	bigint	拍下笔数	0
<i>amt_gmv</i>	Double	拍下金额	0
<i>qty_gmv</i>	bigint	拍下件数	0
<i>unum_bigint</i>	bigint	拍下UV	0
<i>amt_alipay</i>	Double	成交金额	0
<i>num_alipay</i>	bigint	成交笔数	0
<i>qty_alipay</i>	bigint	成交件数	0
<i>unum_alipay</i>	bigint	成交人次	0
<i>ztc_pv_ipv</i>	bigint	直通车引导浏览次数	0
<i>tbk_pv_ipv</i>	bigint	淘宝客引导浏览次数	0
<i>ss_pv_ipv</i>	bigint	搜索引导浏览次数	0
<i>jhs_pv_ipv</i>	bigint	聚划算引导浏览次数	0
<i>ztc_pv_uv</i>	bigint	直通车引导浏览人次	0
<i>tbk_pv_uv</i>	bigint	淘宝客引导浏览人次	0
<i>ss_pv_uv</i>	bigint	搜索引导浏览人次	0
<i>jhs_pv_uv</i>	bigint	聚划算引导浏览人次	0
<i>num_alipay_njhs</i>	bigint	非聚划算支付笔数	0
<i>amt_alipay_njhs</i>	Double	非聚划算支付金额	0
<i>qty_alipay_njhs</i>	bigint	非聚划算支付件数	0
<i>unum_alipay_njhs</i>	bigint	非聚划算支付人次	0

Table 1: *item_feature*:商品粒度相关特征

字段	类型	含义	示例
<code>date</code>	bigint	日期	20150912
<code>item_id</code>	bigint	商品ID	132
<code>store_code</code>	String	仓库CODE	1
<code>cate_id</code>	bigint	叶子类目ID	18
<code>cate_level_id</code>	bigint	大类目ID	12
<code>brand_id</code>	bigint	品牌ID	203
<code>pv_ipv</code>	bigint	浏览次数	2
<code>pv_uv</code>	bigint	流量UV	2
<code>cart_ipv</code>	bigint	被加购次数	0
<code>cart_uv</code>	bigint	加购人次	0
<code>collect_uv</code>	bigint	收藏夹人次	0
<code>num_gmv</code>	Double	拍下笔数	0
<code>amt_gmv</code>	Double	拍下金额	0
<code>qty_gmv</code>	bigint	拍下件数	0
<code>unum_bigint</code>	bigint	拍下UV	0
<code>amt_alipay</code>	Double	成交金额	0
<code>num_alipay</code>	bigint	成交笔数	0
<code>qty_alipay</code>	bigint	成交件数	0
<code>unum_alipay</code>	bigint	成交人次	0
<code>ztc_pv_ipv</code>	bigint	直通车引导浏览次数	0
<code>tbk_pv_ipv</code>	bigint	淘宝客引导浏览次数	0
<code>ss_pv_ipv</code>	bigint	搜索引导浏览次数	0
<code>jhs_pv_ipv</code>	bigint	聚划算引导浏览次数	0
<code>ztc_pv_uv</code>	bigint	直通车引导浏览人次	0
<code>tbk_pv_uv</code>	bigint	淘宝客引导浏览人次	0
<code>ss_pv_uv</code>	bigint	搜索引导浏览人次	0
<code>jhs_pv_uv</code>	bigint	聚划算引导浏览人次	0
<code>num_alipay_njhs</code>	bigint	非聚划算支付笔数	0
<code>amt_alipay_njhs</code>	Double	非聚划算支付金额	0
<code>qty_alipay_njhs</code>	bigint	非聚划算支付件数	0
<code>unum_alipay_njhs</code>	bigint	非聚划算支付人次	0

Table 2: *item_store_feature*:商品粒度相关特征

字段	类型	含义	示例
<code>item_id</code>	bigint	商品ID	333442
<code>store_code</code>	String	仓库CODE	1
<code>a_b</code>	String	商品补少补多cost，用”.”联接起来	10.44_20.88

字段	类型	含义	示例
<i>item_id</i>	bigint	商品ID	333442
<i>store_code</i>	String	仓库CODE	1
<i>target</i>	Double	商品补少补多cost，用”„联接起来	30.0

2 技术方案

2.1 解决思路

该问题属于典型的时间序列分析问题，时间序列分析是指某一指标在一定周期内各个时间点上的数值，按时间先后顺序排列成数列，并对此数列进行平稳化检验、差分、建模、诊断等过程后预测该指标在未来的发展趋势。我们组利用所给数据建立相应数据库，分析数据变化趋势并对其进行清理，最后应用自动回归移动平均混合模型（ARIMA 模型）预测之后两周的全国和区域分仓目标库存量。

2.2 具体方案

数据方面：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。

模型构建：ARIMA模型是自回归移动平均模型（AutoRegressive Moving Average Models）的简称，作为一种经典时间序列预测方法，其主要思想是将非平稳时间序列转化为平稳时间序列，然后将因变量仅对它的滞后值以及随机误差项的现值和滞后值进行回归。其预测过程如下：

（1）将收集到的数据以某个时间点为界限分割为训练集和验证集。对训练集绘制时序图，以判别其是否具有平稳性和季节性特征；

（2）对非平稳序列进行平稳化处理，一般采用差分处理，且处理后的数据能够通过单位根检验；

（3）根据时间序列模型的特征进行模型识别和参数估计，建立相应的模型；

（4）对模型进行假设检验，采用Ljung-Box检验残差序列是否为白噪声；

（5）利用已通过检验的模型进行预测模型分析并和验证集进行对比，评估模型的拟合精度。

3 实现和实验结果

3.1 具体实现

3.1.1 数据清理

数据清理的具体过程如下:

(1) 创建tianchi数据库:

```
createdatabasetianchi;
```

(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;
```

(7) 非聚划算的浏览次数计算公式为:

$$njhs_pv_ipv == pv_ipv - jhs_pv_ipvAg - yAg$$

```
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。

(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')`:

其次, 统计所有商品在时间轴上的分布:

```
create table statfigure1
```

```
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;
```

(13) 分仓的数据表处理:

首先检查看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),  
sum(jhs_pv_ipv), sum(njhs_pv_ipv), sum(collect_uv)
```

```
from 01_item_store_feature
```

```
where item_id = 30378
```

```
group by store_code;
```

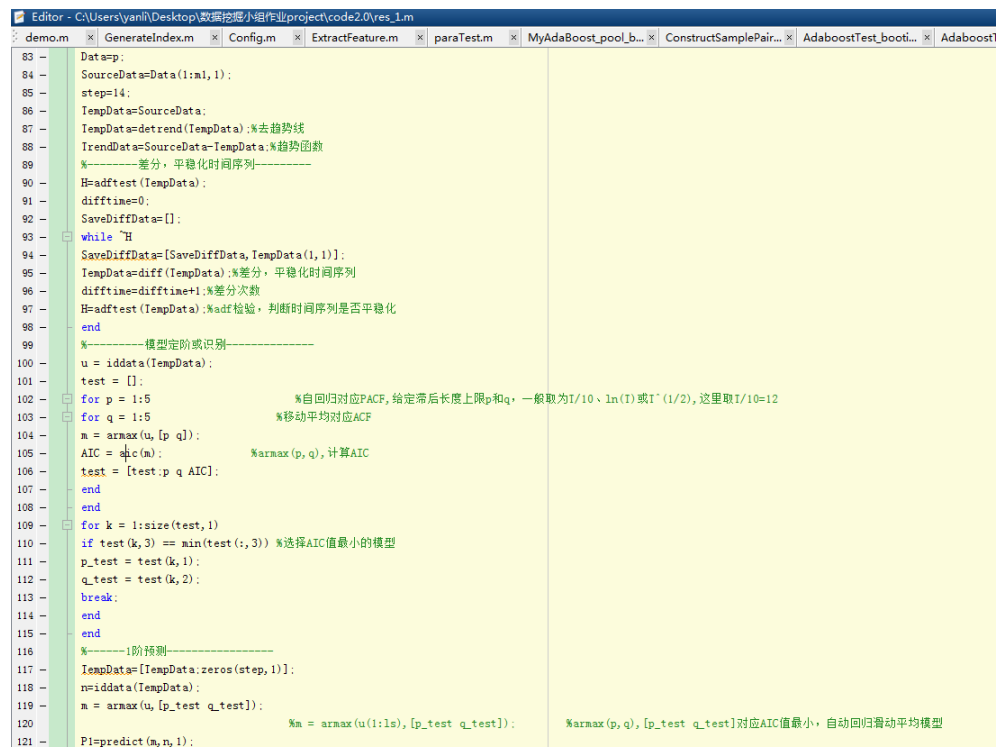
最后，查看某一个商品在售时间段（有浏览记录的日期，并不是销售量的日期）

```
select min(date), max(date) from 01_item_store_feature
```

```
where item_id = 30378
```

3.1.2 模型构建

利用Arima模型进行预测的代码如Figure 1 和Figure 2 所示：



```
83 Data=p;
84 SourceData=Data(1:m,1);
85 step=14;
86 TempData=SourceData;
87 TempData=detrend(TempData); %去趋势线
88 TrendData=SourceData-TempData; %趋势函数
89 %-----差分，平稳化时间序列-----
90 H=adftest(TempData);
91 diffTime=0;
92 SaveDiffData=[];
93 while ~H
94     SaveDiffData=[SaveDiffData, TempData(1,1)];
95     TempData=diff(TempData); %差分，平稳化时间序列
96     diffTime=diffTime+1; %差分次数
97     H=adftest(TempData); %adf检验，判断时间序列是否平稳化
98 end
99 %-----模型定阶或识别-----
100 u = iddata(TempData);
101 test = [];
102 for p = 1:5 %自回归对应PACF，给定滞后长度上限p和q，一般取为T/10、ln(T)或T^(1/2)，这里取T/10=12
103     for q = 1:5 %移动平均对应ACF
104         n = armax(u, [p q]);
105         AIC = aic(n); %armax(p,q)，计算AIC
106         test = [test; p q AIC];
107     end
108 end
109 for k = 1:size(test,1)
110     if test(k,3) == min(test(:,3)) %选择AIC值最小的模型
111         p_test = test(k,1);
112         q_test = test(k,2);
113         break;
114     end
115 end
116 %-----1阶预测-----
117 TempData=[TempData; zeros(step,1)];
118 n=iddata(TempData);
119 m = armax(u, [p_test q_test]); %m = armax(u(1:ls), [p_test q_test]); %armax(p,q), [p_test q_test]对应AIC值最小，自动回归滑动平均模型
120 Pl=predict(m, n, 1);
```

Figure 1: Arima代码1

3.2 实验结果

根据这个预测模型，我们可以得到相应的预测结果。由于Arima 模型要求输入数据满足平顺性，在试验中我们使用adftest函数对输入数据进行平顺性检测，如Figure 3展示的是一个ID300 的商品不平滑的数据。


```

Editor - C:\Users\yani\Desktop\数据挖掘小组作业\project\code2.0\res_1.m
demo.m  GenerateIndex.m  Config.m  ExtractFeature.m  paraTest.m  MyAdaBoost_pool_b...  ConstructSamplePair...  AdaboostTest_booti...  A
116 %-----1阶预测-----
117 TempData=[TempData;zeros(step,1)];
118 n=iddata(TempData);
119 m = armax(u,[p_test q_test]);
120 %m = armax(u(1:ls),[p_test q_test]); %armax(p,q),[p_test q_test]对应AIC值最小，自动回归滑动平均模型
121 Pl=predict(n,n,1);
122 PreR=Pl.OutputData;
123 PreR=PreR';
124 %-----还原差分-----
125 if size(SaveDiffData,2)~=0
126 for index=size(SaveDiffData,2):-1:1
127 PreR=cumsum([SaveDiffData(index),PreR]);
128 end
129 end
130 %-----预测趋势并返回结果-----
131 mpl=polyfit([1:size(TrendData',2)],TrendData',1);
132 xt=[];
133 for j=1:step
134 xt=[xt,size(TrendData',2)+j];
135 end
136 TrendResult=polyval(mpl,xt);
137 PreData=TrendResult+PreR(size(SourceData',2)+1:size(PreR,2));
138 tempx=[TrendData',TrendResult]+PreR; % tempx为预测结果

```

Figure 2: Arima代码2

对于不满足平顺性的数据我们需要做一阶差分，使数据满足模型的要求。如Figure 4为数据做一阶差分处理后的结果。满足了数据要求之后，我们使用代码自动调整Armax的参数P、Q，并选择出最优的参数。并用此模型进行进一步的数据预测。预测结果如Figure 5所示，可见该模型对于数据量较小且不满足平顺性的数据而言表现出较差的性能。

为做进一步的对比，我们使用id 400的商品数据进行预测分析，该商品数据满足平顺性，且具有一定的规模。原始数据如Figure 6所示。

用该数据进行预测的结果Figure 7所示，可见该模型在这种数据的预测中表现出了良好的性能。

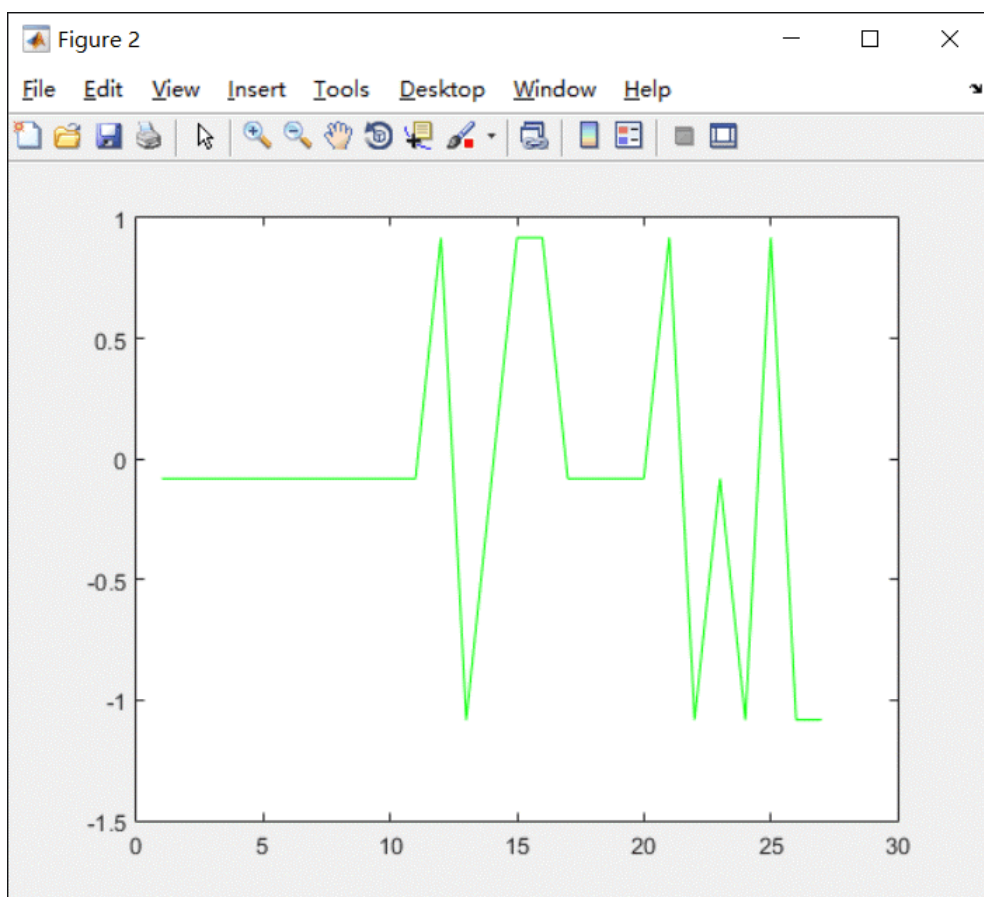


Figure 3: 数据未差分结果

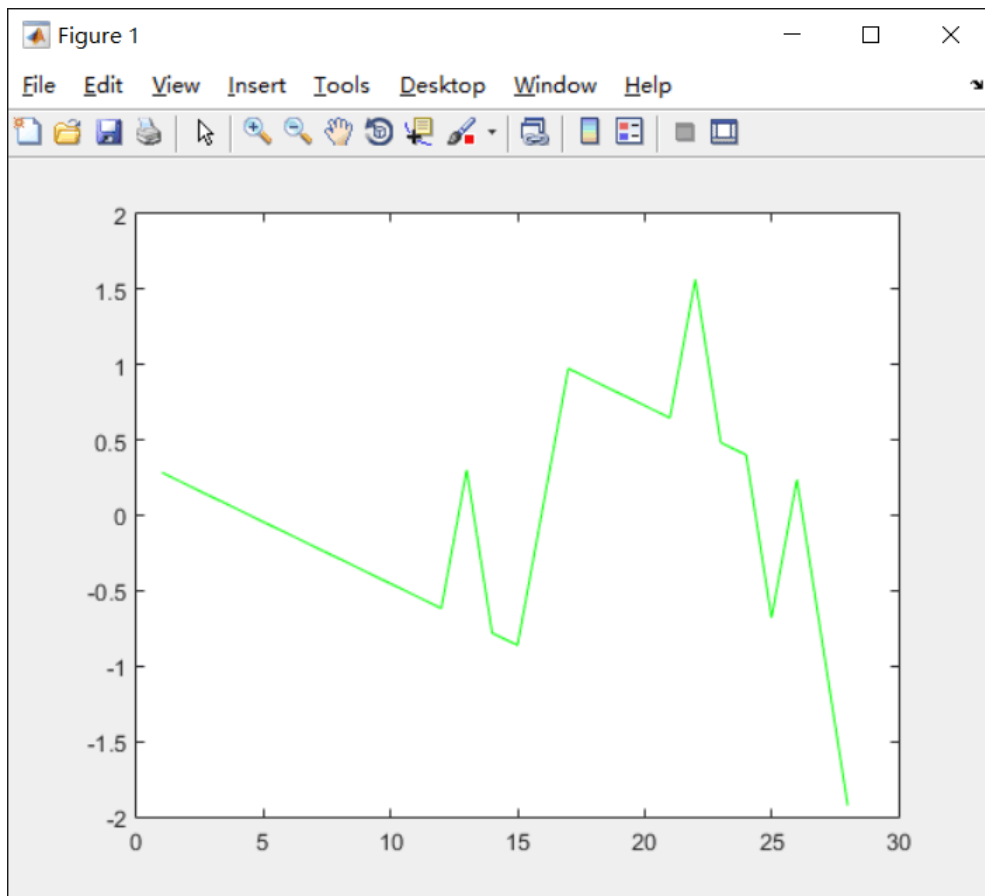


Figure 4: 数据差分结果

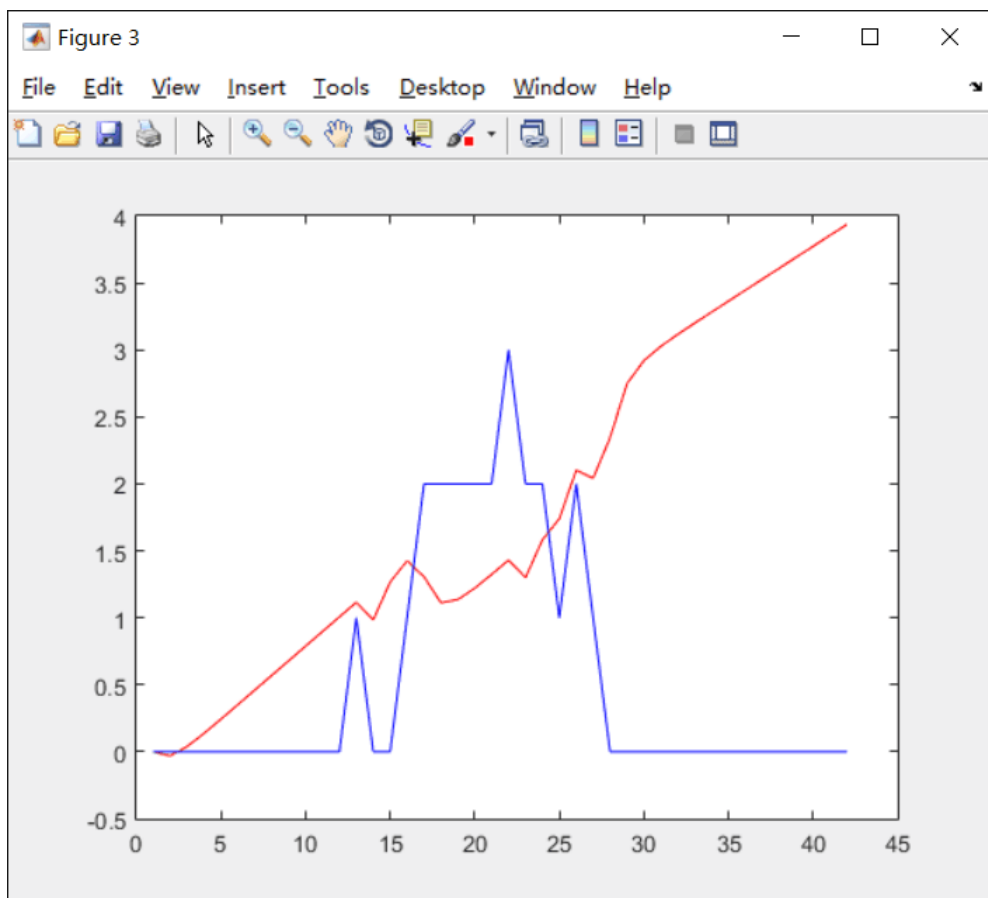


Figure 5: 数据预测结果:其中蓝色线表示实际数据，红色线表示实际数据。

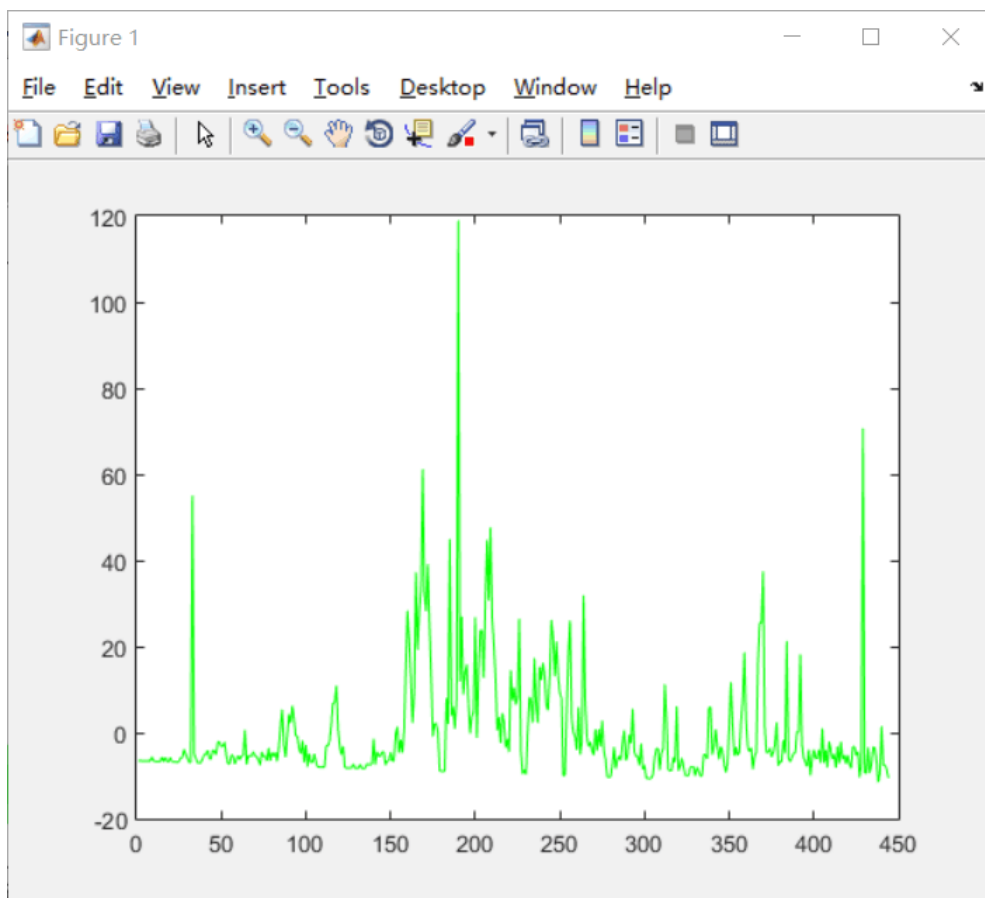


Figure 6: 满足数据平顺性的原始数据

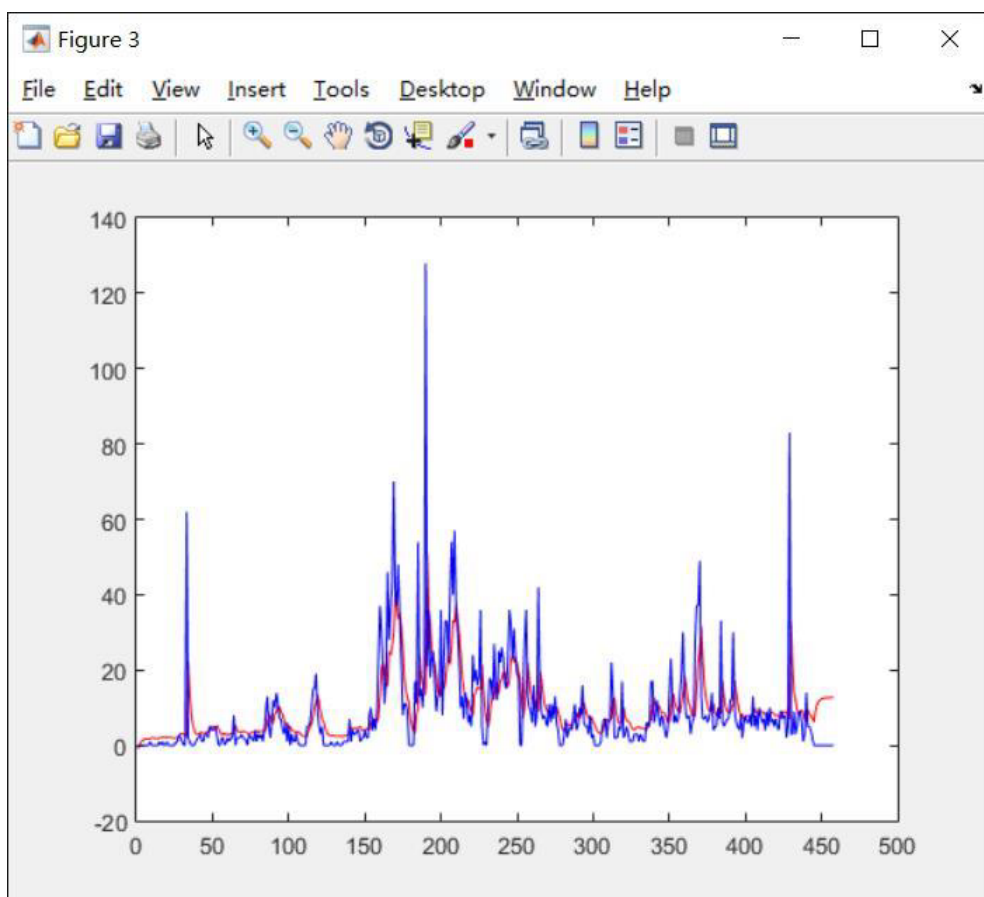


Figure 7: 数据最终预测结果