基于ARIMA模型的股票价格预测问题研究

# 问题描述

时间序列预测方法是比较常用的预测方法，它有一系列完善的理论基础，随着人们对股市的追捧，不少学者也尝试将时间序列预测应用于股票的价格预测，具体的讲时间序列预测股价是将股票价格或者价格指数看作变化的时间序列，通过建立合理的时间序列模型以预测未来发展变化的规律和趋势。本文选取“长安汽车”的股票收盘价作为时间序列实证分析数据，通过建立ARIMA模型对创业板市场股票价格变动的规律和趋势进行预测。需要指出的是，股票市场的行情是千变万化的，时间序列分析法只是利用历史数据，期望从中获取有用信息来预测将来走势，而并没有考虑影响股价变动的原因，故一般只是直观分析，仅做短时间内的预测。[1]

# 数据爬取

## 爬取流程

### 导入所需模块

首先用到了selenium库，优点是能够很好地模拟浏览器的使用情况，能用来简单地编写出爬虫程序且不容易被反爬；其次用到了time库，用来暂停一段时间模仿浏览器的真实行为同时等待网页加载；最后用到了re库对网页的url进行分析。[2]

### 分析url

在网易财经中爬取股票的历史数据，观察出股票信息的存储地址为<http://quotes.money.163.com/trade/lsjysj_xxxxxx.html>的格式，其中xxxxxx为股票代码，在此地址中点击“下载数据”按钮，再点击“下载”按钮即可下载该股票的所有历史数据。

### 获取数据

由于本次只需获取代码000625股票的历史数据，故存储地址可唯一确定。关键代码如下：

1. url = 'http://quotes.money.163.com/trade/lsjysj\_000625.html#01b07'
2. browser.get(url)
3. time.sleep(2)
4. browser.find\_element(
5. By.XPATH, '//\*[@id="downloadData"]').click()   # press button
6. time.sleep(2)
7. browser.find\_element(
8. By.XPATH, '/html/body/div[2]/div[5]/div[2]/form/div[3]/a[1]').click()
9. time.sleep(2)

# 数据预处理

## 导入数据

直接使用pandas库的read\_csv方法，代码如下

1. df = pd.read\_csv('000625\_source.csv', sep=',')

## 缺失值检查

经检查，数据完整、无缺失值，代码如下

1. na\_cols = df.isnull().any(axis=0)

# 时间序列预处理

若选取数据过少则无法充分提取历史数据中的信息，数据选取过多又会因间隔较长时的股价会对后期的预测股价影响较小造成不必要的误差，所以取最近150天的数据进行分析。为方便起见，只取每天的收盘价作为当日股价。

关键代码如下

1. data = data.head(150)
2. dataUtil = data.iloc[:, ‘日期’]

# ARIMA模型预处理

## 时序图检验

首先用直观的时序图观察该时间序列是否平稳，结果如图1

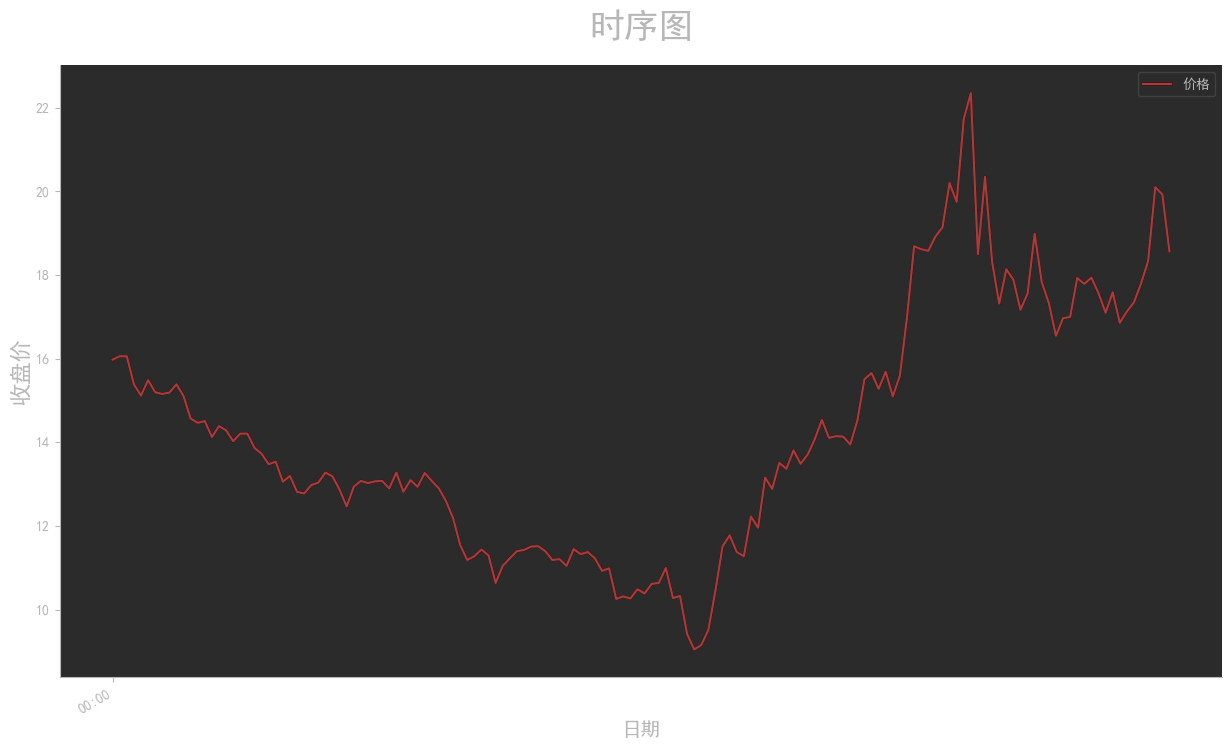


图 1

显然不是平稳时间序列，需要做平稳化处理。

## 差分平稳化

对原始数据做一阶差分处理，相关值如表1所示，处理后，可以看出 dy 的 ADF 统计值的绝对值为 3.9615702003502933，大于显著性水平为0.05时的3.8746401679188405，故接受存在一个单位根的原假设，说明一阶差分序列是平稳的，因此d=1。

**表 1:一阶差分后adf值显著性比较**

|  |  |
| --- | --- |
| **分类** | **adf值** |
| 结果 | -3.9615702003502933 |
| 1% | -4.454630091433345 |
| 5% | -3.8746401679188405 |
| 10% | -3.579316526412991 |

## 自相关图和偏自相关图

自相关图和偏自相关图如图2、3所示

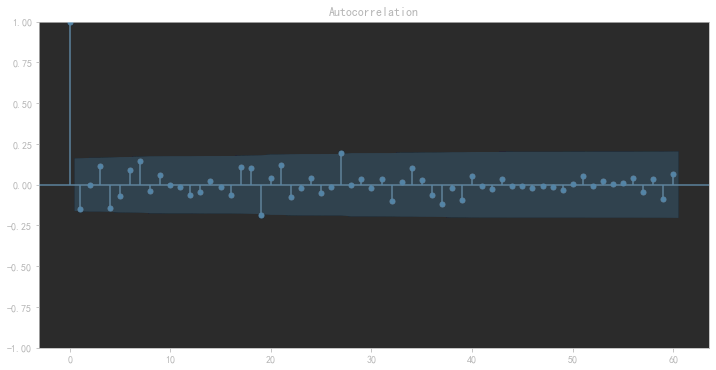


图 2

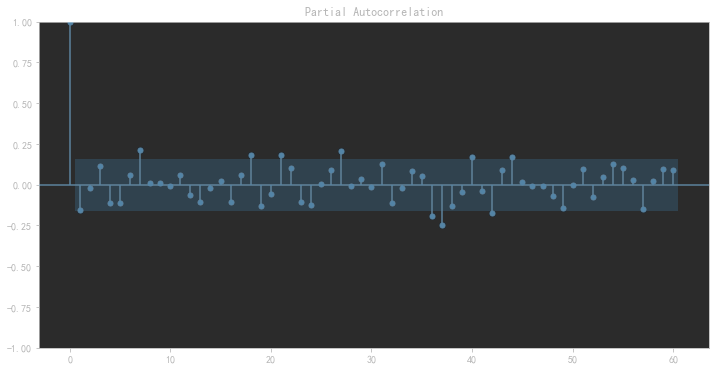


图 3

观察出自相关图截尾，偏自相关图拖尾，按理应使用MA模型，在之后的网格搜索中包含。

## 白噪声检验

白噪声检验值如表2所示

**表2:** 白噪声检验值

|  |  |
| --- | --- |
| **lb\_stat** | **lb\_pvalue** |
| 3.485312 | 0.061916 |
| 3.485558 | 0.175033 |
| 5.420166 | 0.143492 |
| 8.508024 | 0.074644 |
| 9.200742 | 0.101320 |
| 10.421292 | 0.107995 |
| 13.678446 | 0.057205 |
| 13.909369 | 0.084158 |

lag=1，4，7时，p值接近5%，依然当作通过检验处理。认为数据为非白噪声序列，存在相关性，有一定规律可循。

# 模型的建立与预测

## 划分训练集和测试集

差分后还剩余149个数据，将前144个作为训练集，最后5个作为测试集。

## 网格搜索最优参数

对于库函数ARIMA(train, order(p, d, q))，使用网格搜索的方式寻找order的最佳匹配值，定义了函数evaluate\_arima\_model和evaluate\_models，关键代码如下

1. **def** evaluate\_arima\_model(X, arima\_order):
2. size = len(X) - 5
3. train\_tmp, test\_tmp = X[:size], X[size:]
4. tmp\_model = ARIMA(train\_tmp, order=arima\_order).fit()
5. tmp\_pred = tmp\_model.forecast(5)
6. error = mean\_squared\_error(tmp\_pred, test\_tmp)
7. **return** error
8. **def** evaluate\_models(dataset, p\_values, d\_values, q\_values):
9. best\_score, best\_cfg = float("inf"), None
10. **for** p **in** p\_values:
11. **for** d **in** d\_values:
12. **for** q **in** q\_values:
13. order = (p,d,q)
14. **try**:
15. mse = evaluate\_arima\_model(dataset, order)
16. **if** mse < best\_score:
17. best\_score, best\_cfg = mse, order
18. **print**('ARIMA%s MSE=%.3f' % (order,mse))
19. **except**:
20. **continue**

通过搜索得到最佳参数为ARIMA(8, 1, 6) ，其MSE最小，MSE=0.102

## 运用模型进行预测

预测结果如表3所示

**表3:** 模型预测结果

|  |  |
| --- | --- |
| **预测值** | **真实值** |
| 0.117384 | 0.45 |
| -0.259985 | 0.54 |
| -0.035829 | 1.76 |
| 0.093502 | -0.17 |
| -0.139943 | -1.36 |

将预测值与真实值画在同一坐标系中，如图4所示

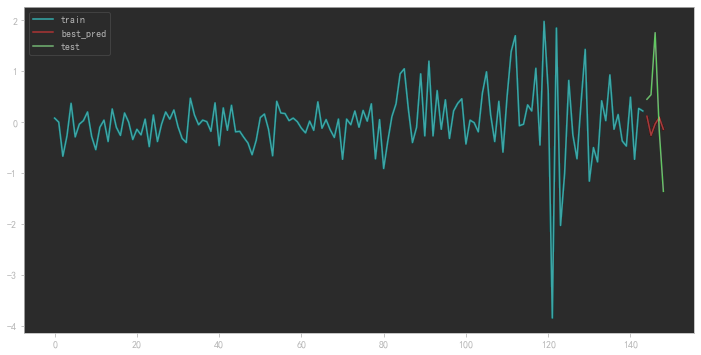


图 4

# 结果分析

## 由图4直观分析

结果与真实值在增减趋势上几乎相同，但是在数值大小方面还不够准确。

## 残差分析

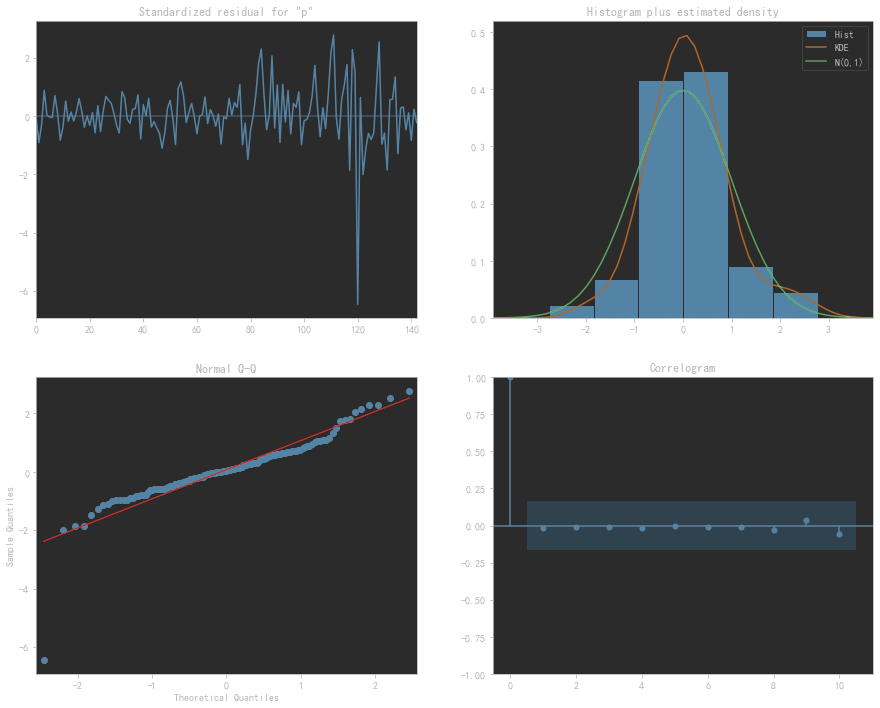


图 5

如图5所示，残差分布有较好的正态性，模型拟合较好。

# 参考文献

1. 吴玉霞 & 温欣.(2016).基于ARIMA模型的短期股票价格预测. 统计与决策(23),83-86. doi:10.13546/j.cnki.tjyjc.2016.23.051.
2. 知乎用户：黑木鸟.如何用爬虫抓取股市数据并生成分析报表[EB/OL]. https://www.finlab.tw/%E7%94%A8%E7%88%AC%E8%9F%B2%E7%88%AC%E5%85%A8%E4%B8%96%E7%95%8C%E8%82%A1%E5%83%B9/, 2017–02–22/2018–05–31.

附录