小学期项目文档

指导老师：李宇

组长：王昱

组员：赵康辉 张正阳

武熠彬 林小倩

**目录**

[1 ARIMA模型 3](#_Toc45485088)

[1.1 任务介绍 3](#_Toc45485089)

[1.2 技术要点 3](#_Toc45485090)

[1.3 理论研究 3](#_Toc45485091)

[1.4 实例分析 5](#_Toc45485092)

[2 数据清洗 9](#_Toc45485093)

[2.1 任务介绍 9](#_Toc45485094)

[2.2 技术要点 9](#_Toc45485095)

[2.3 工作流程 9](#_Toc45485096)

[2.4 成果展示 10](#_Toc45485097)

[3 天气预测接口 11](#_Toc45485098)

[3.1 数据前期准备 11](#_Toc45485099)

[3.2 数据平稳性处理 11](#_Toc45485100)

[3.3 模型选择 12](#_Toc45485101)

[3.4 预测一周天气 12](#_Toc45485102)

[4 数据传输 13](#_Toc45485103)

[4.1 任务介绍 13](#_Toc45485104)

[4.2 技术要点 14](#_Toc45485105)

[4.3 工作流程 14](#_Toc45485106)

[4.4 成果展示 15](#_Toc45485107)

[5 前端界面设计 15](#_Toc45485108)

[5.1 任务介绍 15](#_Toc45485109)

[5.2 技术要点 15](#_Toc45485110)

[5.3 工作流程 15](#_Toc45485111)

[6 服务器端设计 16](#_Toc45485112)

[6.1 任务介绍 16](#_Toc45485113)

[6.2 技术要点 16](#_Toc45485114)

[6.3 环境配置 16](#_Toc45485115)

[6.4 工作流程 17](#_Toc45485116)

# 1 ARIMA模型

## 1.1 任务介绍

了解AR、MA、ARMA、ARIMA模型的原理和实现思路

## 1.2 技术要点

1. 完成以ARIMA模型为核心的数学知识理论的学习与理解，为实际操作提供理论支撑；
2. 通过pip配置安装pandas、Numpy等相关包；
3. 学习依赖包相关使用方法
4. 运用python语言对数据进行平稳性处理；

## 1.3 理论研究

1. **截尾和拖尾**

所谓截尾指的是从某阶开始截尾是指时间序列的自相关函数（ACF）或偏自相关函数（PACF）在某阶后均为0的性质（比如AR的PACF）；拖尾是ACF或PACF并不在某阶后均为0的性质（比如AR的ACF）。

总结：

截尾：在大于某个常数k后快速趋于0为k阶截尾

拖尾：始终有非零取值，不会在k大于某个常数后就恒等于零(或在0附近随机波动)

1. **自回归模型（AR）**

自回归模型（英语：Autoregressive model，简称AR模型），是统计上一种处理时间序列的方法，用同一变数例如x的之前各期，亦即x1至xt-1来预测本期xt的表现，并假设它们为一线性关系。因为这是从回归分析中的线性回归发展而来，只是不用x预测y，而是用x预测 x（自己）；所以叫做自回归。

AR模型思想很简单，该模型认为通过时间序列过去时点的线性组合加上白噪声即可预测当前时点，它是随机游走的一个简单扩展。AR模型对偏自相关函数（PACF）截尾，对自相关函数(ACF)拖尾。

时间序列自相关与概率论中的相关定义本质是一致的，它衡量的是序列自身在不同时刻随机变量的相关性；偏自相关系数则剔除了两时刻之间其他随机变量的干扰，是更加纯粹的相关。

自回归模型被广泛运用在经济学、信息学、自然现象的预测上。p阶自回归过程的公式定义：【如图1.3.1】

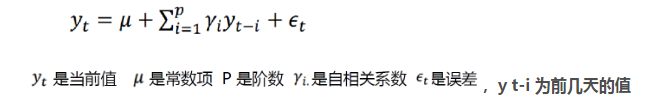


图1.3.1

1. **移动平均模型（MA）**

MA模型(moving average model)滑动平均模型，模型参量法谱分析方法之一，也是现代谱估中常用的模型。q阶移动平均模型的自相关系数q阶截尾，偏自相关系数拖尾。

MA模型和AR大同小异，它并非是历史时序值的线性组合而是历史白噪声的线性组合。与AR最大的不同之处在于，AR模型中历史白噪声的影响是间接影响当前预测值的（通过影响历史时序值）。

MA模型对偏自相关函数（PACF）拖尾，对自相关函数(ACF)截尾。在金融模型中，MA常用来刻画冲击效应，例如预期之外的事件。

关于它的最重要的关注点是自回归模型中的误差项的累加，移动平均法能有效地消除预测中的随机波动，这就是MA模型的作用及意义。

q阶自回归过程的公式定义：【如图1.3.2】



图1.3.2

1. **自回归滑动平均模型**

自回归滑动平均模型（英语：Autoregressive moving average model，简称：ARMA模型）。是研究时间序列的重要方法，由自回归模型（AR模型）与移动平均模型（MA模型）为基础“混合”构成。在市场研究中常用于长期追踪资料的研究。

ARMA模型可以表示为：【如图1.3.3】

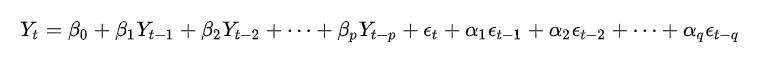


图1.3.4

1. **差分整合移动平均自回归模型（ARIMA）**

又称整合移动平均自回归模型（移动也可称作滑动），是时间序列预测分析方法之一。ARIMA模型是在ARMA模型的基础上解决非平稳序列的模型。 ARIMA(p，d，q)中，AR是“自回归”，p为自回归项数；MA为“滑动平均”，q为滑动平均项数，d为使之成为平稳序列所做的差分次数（阶数）。对时间序列数据进行分析和预测比较完善和精确的算法是博克思-詹金斯方法，其常用模型包括：自回归模型（AR模型）、滑动平均模型（MA模型）、（自回归-滑动平均混合模型）ARMA模型、（差分整合移动平均自回归模型）ARIMA模型。

ARIMA(p，d，q)模型是ARMA(p，q)模型的扩展。ARIMA(p，d，q)模型可以表示为：【如图1.3.4】

IMG_256

图1.3.4

其中*L*是滞后算子（Lag operator）, d∈Z，d>0.

## 1.4 实例分析

1. **选取原始数据集**

选取原始数据集（这里用的是北京7.10的数据）

导入python工程包，确定具体的数据对象（dta），确定数据图像始末横坐标，调用plot与show展示原始折线图：【如图1.4.1】

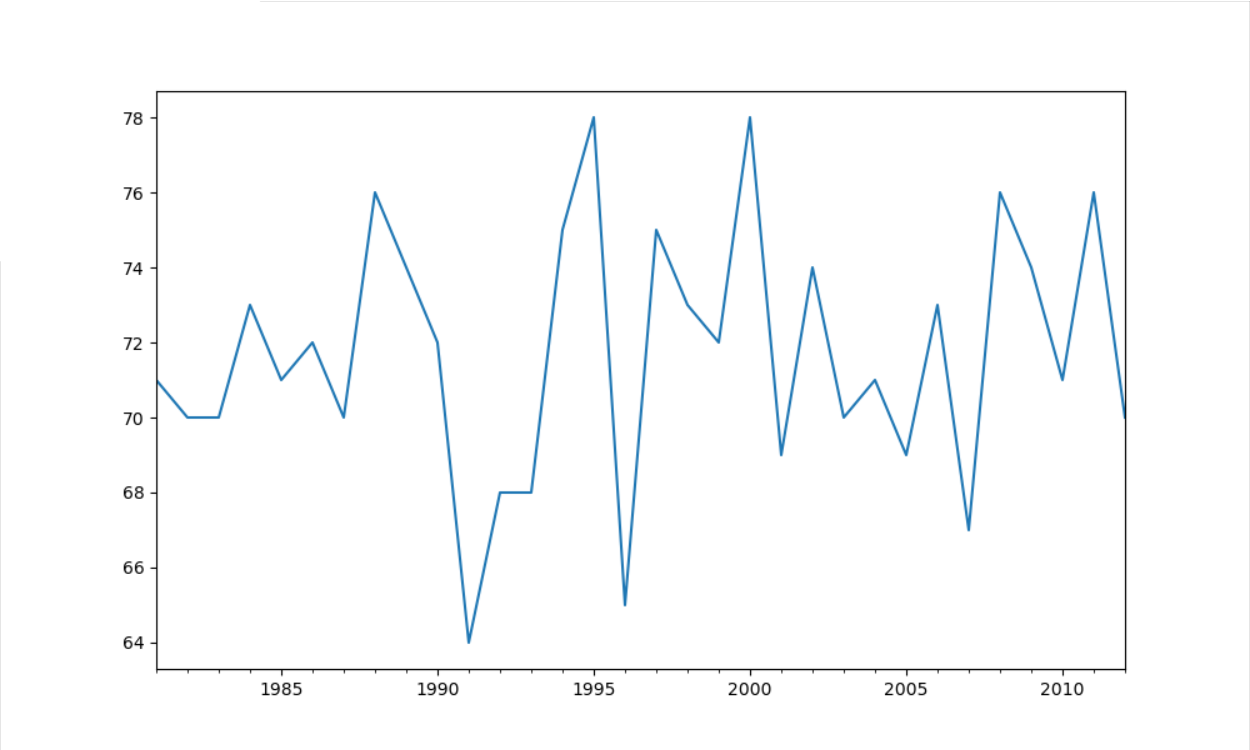


图1.4.1

1. **单位根检验**

在展示原始折线图后，案例中使用了单位根检验的方法对已知的数据进行平稳性判断，根据输出参数判定是否进行差分及进行几阶差分。

下面展示单位根检验方法【如图1.4.2】与如何判断进行差分：【如图1.4.3】

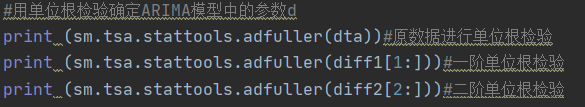


图1.4.2

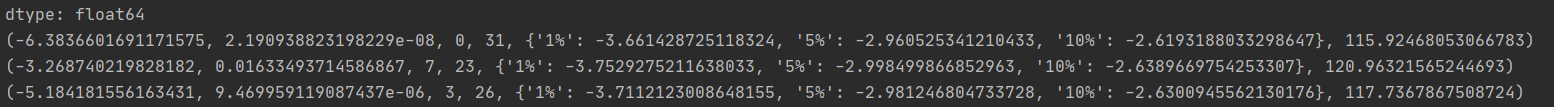


图1.4.3

该函数可以返回一个数组，包含五个数据。第一个是adt检验的结果，也就是t统计量的值。第二个是t统计量的P值。第三个是计算过程中用到的延迟阶数。第四个是用于ADF回归和计算的观测值的个数。第五个是配合第一个一起看的，是在99%，95%，90%置信区间下的临界的ADF检验的值。如果第一个值比第五个值小证明平稳，反正证明不平稳。根据这个图的结果来看，原始数据已经足够平稳，可以进入下一阶段。

1. **参数定阶（p，q）**

方法一：原始方法，使用工具包输出一阶差分后的ACF图与PACF图【如图1.4.4】，并且以相关图判断法则大致确定参数p，q。

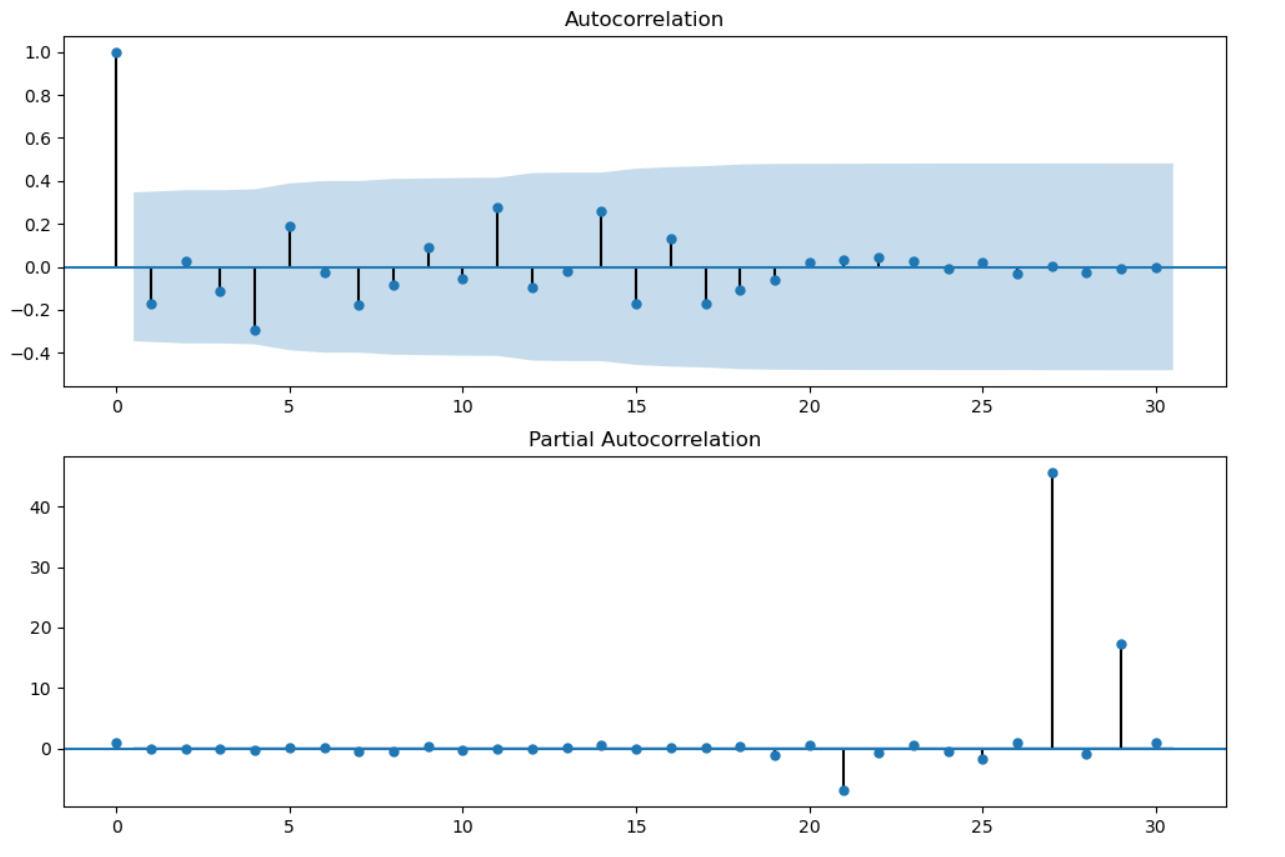


图1.4.4

此为ARMA模型参数确定法则：【如图1.4.5】



图1.4.5

方法二： AIC（BIC）法，给定范围，输出AIC（BIC）的热力图【如图1.4.6】，然后工作人员自行寻找最佳参数。颜色最深、数值最小的便是最佳pq取值。

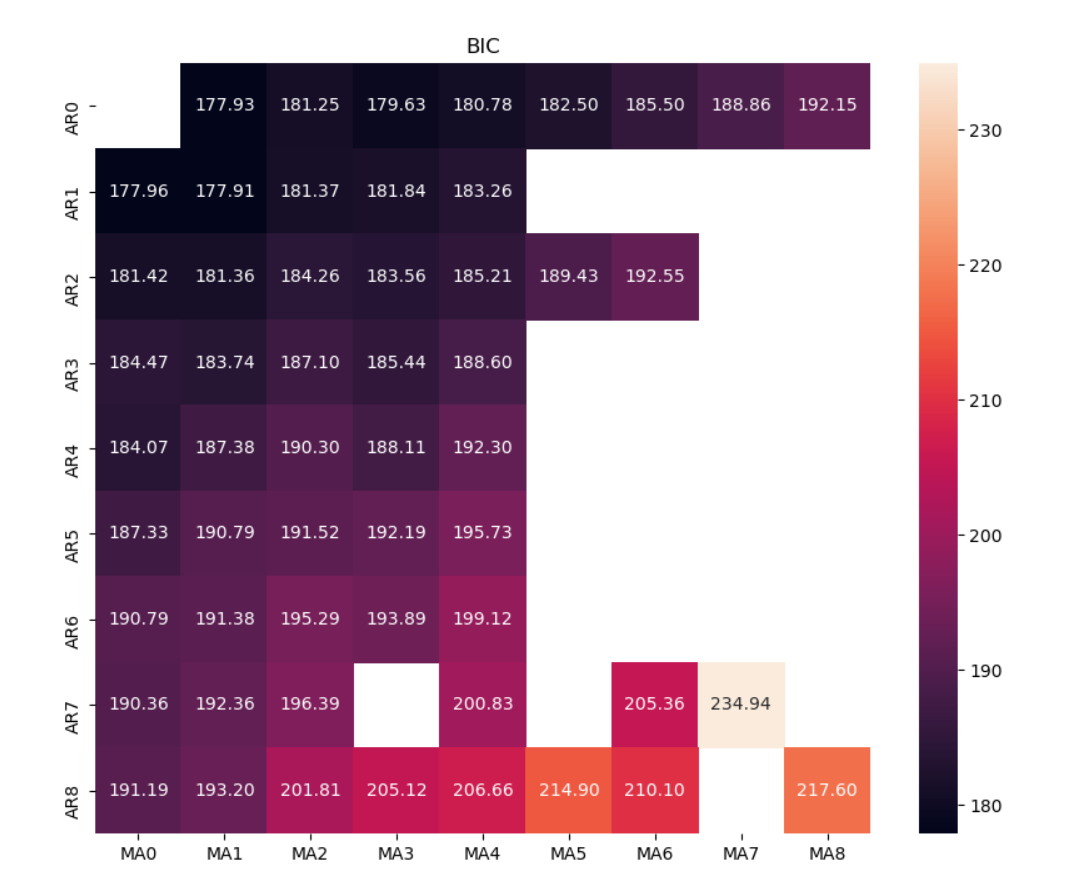


图1.4.6

方法三，使用auto\_arima()函数，输出最佳模型参数取值。如图1.4.7model便是取（0,0,4）

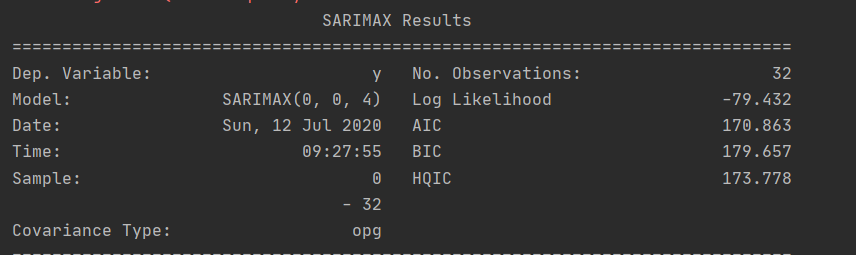


图1.4.7

方法四，自己编写函数，对pq从0到7取值，根据模型的bic值确定最佳pq。这样可以达到自动化确定，适合后续编写。

1. **训练模型**

本阶段旨在寻找最佳参数进行数据预测。实例在最初给定原始数据时，隐藏了最后5个数据，然后分别调用了不同的参数对，再进行QQplot检验与数据预测观察（与原始隐藏的5个数据进行对比），最终确定最优参数对，训练得到最优模型。

1. **数据预测**

确定最佳参数对后，首先再次输出此时的QQplot，从而更为直观明显地呈现数据预测拟合效果【如图1.4.8】。接着就进行n年份的数据预测，并输出预测图【如图1.4.9】，以直观表现预测效果。

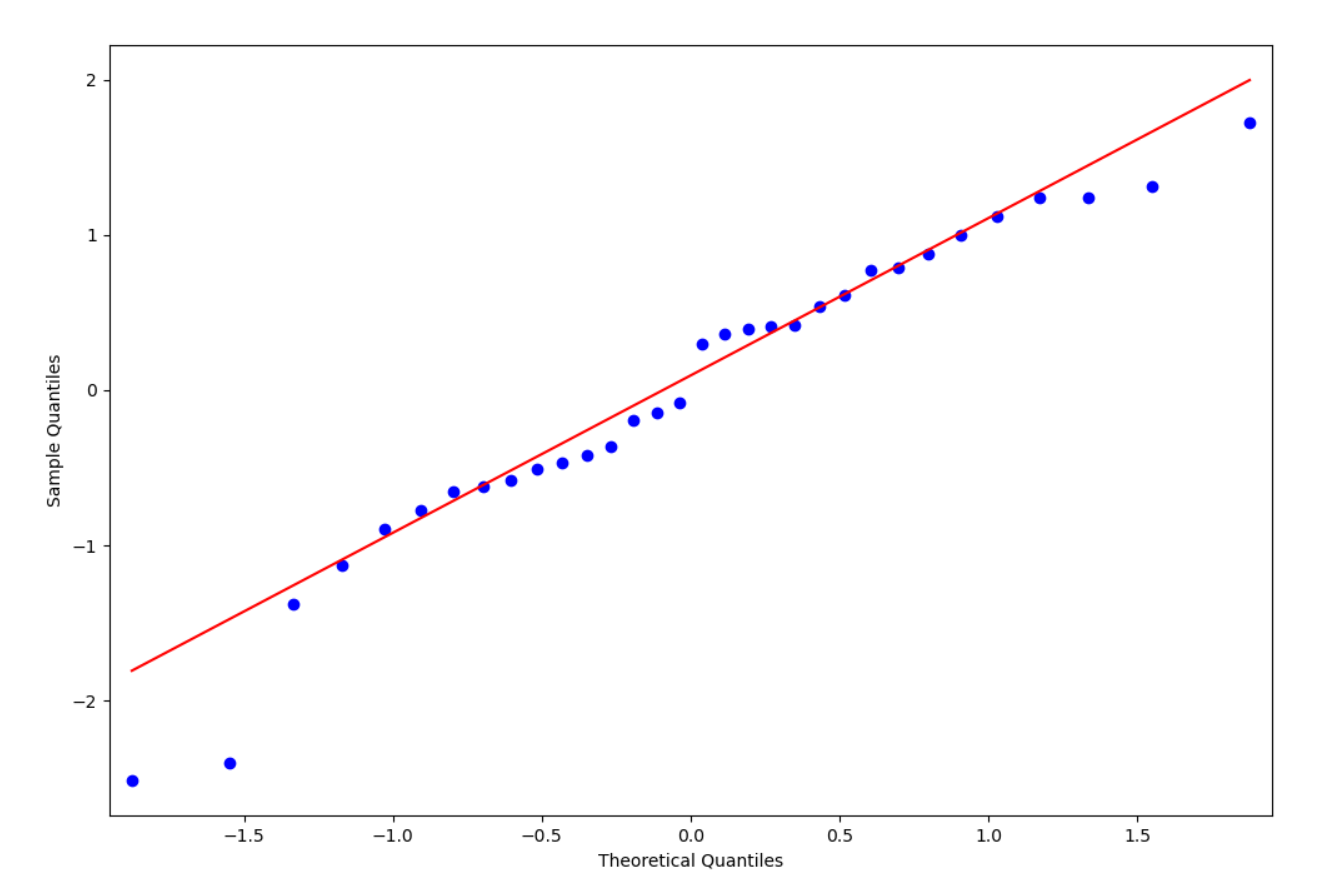


图1.4.8

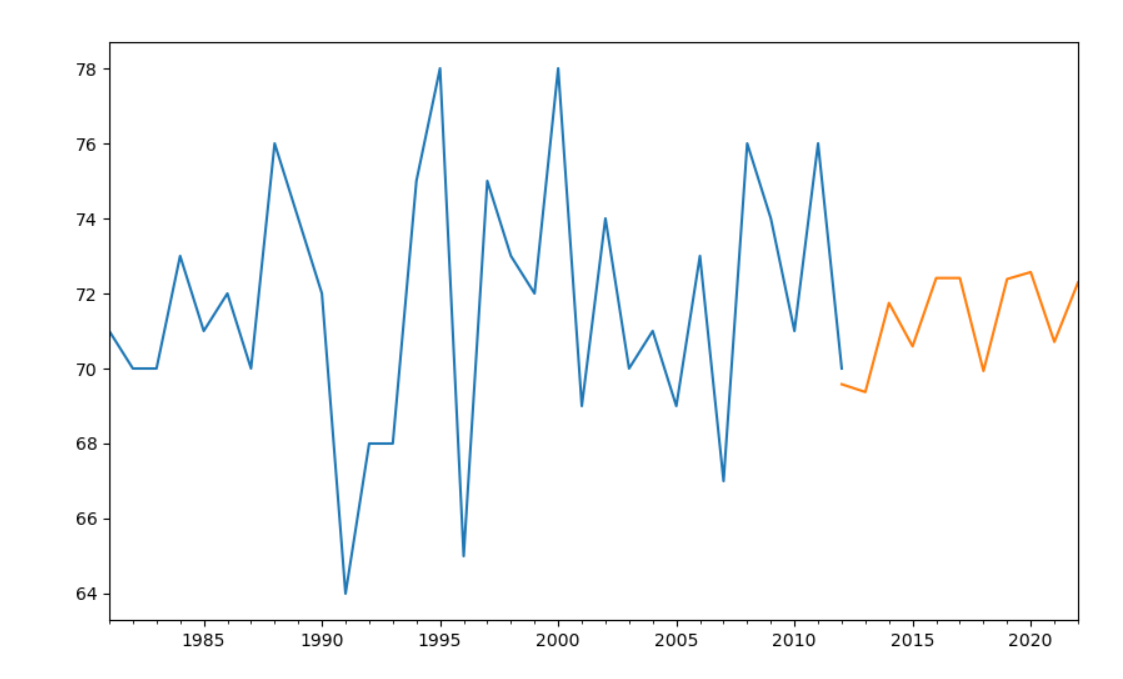


图1.4.9

# 2 数据清洗

## 2.1 任务介绍

1. 获取原始数据集；
2. 根据原始数据集筛选出项目所需要的数据并且过滤掉空值，生成清洗后的数据集。

## 2.2 技术要点

1. 配置Spark环境；
2. 下载并导入pandas依赖包；
3. 学习python关于筛选数据，读取和生成csv文件，过滤数据的相关知识。
4. 虚拟机配置
5. Spark集群配置

## 2.3 工作流程

1. **获取数据集**
   1. 进入美国NOAA网站输入地区名称及所需数据信息类型，点击下载即可完成；

由于需要翻墙，所以还有一个学习翻墙的过程。

1. **下载并导入pandas依赖包**
   1. 在Pycharm中import相关项目包，再点击下载即可导入。出现问题时，使用控制台终端手动下载。
2. **进行数据清洗**
   1. 进入D:\Spark\bin 目录，运行pySpark,进入Spark，导入pandas,datetime,parser依赖包；
   2. 加载历史数据文件，筛选数据，例如天气预测需要日期，气温的最高值和最低值，筛选‘DATE’, ‘TMAX’, ‘TMIN’三列数据配置数据属性，得到需要的列，过滤空值；
   3. 根据上一步的过滤条件，对日期数据进行限制，进一步清洗数据，将清洗后的数据写入csv文件，生成新的数据集。
3. **虚拟机与Spark集群配置**

配置完成后发现电脑无法运行3台虚拟机，另外再了解到虚拟机和spark集群是为了处理大量数据，故放弃使用虚拟机，直接使用windows下单节点清洗数据。

## 2.4 成果展示

1980-2012（因为12年后数据缺失过多，故只采用到12年）年每年7月10日的最高气温与最低气温数据集（部分）：【如图2.4.1】

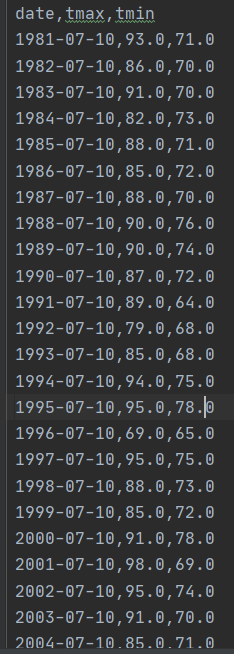


图2.4.1

# 3 天气预测接口

## 3.1 数据前期准备

1. **任务介绍**
   1. 下载全国各省省会城市天气数据
   2. 将数据建模相关包导入项目
   3. 将清洗数据函数导入，实现动态清洗，即需要哪个城市清洗哪个数据
2. **技术要点**
   1. 相关包的导入与配置；
   2. 相关数据集的导入。
3. **任务流程**
   1. 导入数据建模相关包，导入结果【如图3.1.1】；

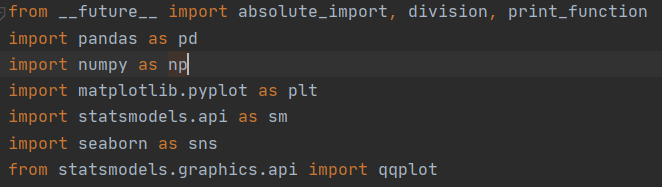


图3.1.1

* 1. 导入相关数据集；
  2. 导入数据清洗函数所在包
  3. 改写函数，实现指定城市和日期清洗

## 3.2 数据平稳性处理

1. **任务介绍**
   1. 平稳性要求经由时间序列所得到的拟合曲线在未来的一段时间内仍能顺着现有的形态“惯性”地延续下去，平稳性要求序列的均值和方差不发生明显变化；
   2. ARIMA 模型对时间序列的要求是平稳型。当数据差异特别大时，为使数据变得平稳，应当对时间序列进行差分，知道得到一个平稳时间序列。
2. **技术要点**
   1. 差分法：时间序列在t和t-1时刻的差值；
   2. 单位根检验：根据在不同差分情况下得到的单位根值进行比较，选择最小值作为最优差分次数。
3. **任务流程**
   1. 对原数据分别进行一阶差分和二阶差分；
   2. 分别对原数据，一阶差分数据，二阶差分数据进行单位根检验；
   3. 选择单位根检验最小值作为最终模型参数；
   4. 编写函数实现自动化确定。

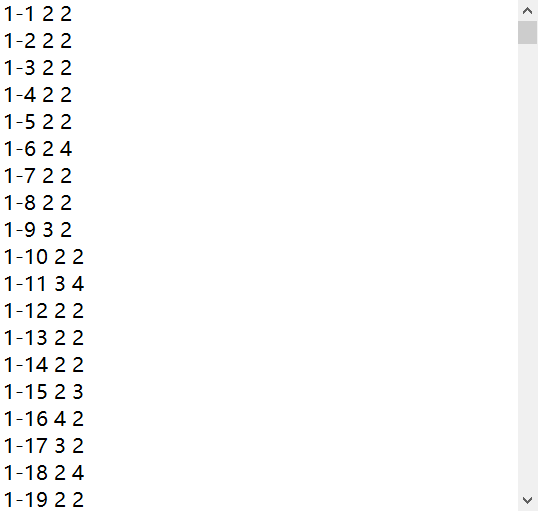
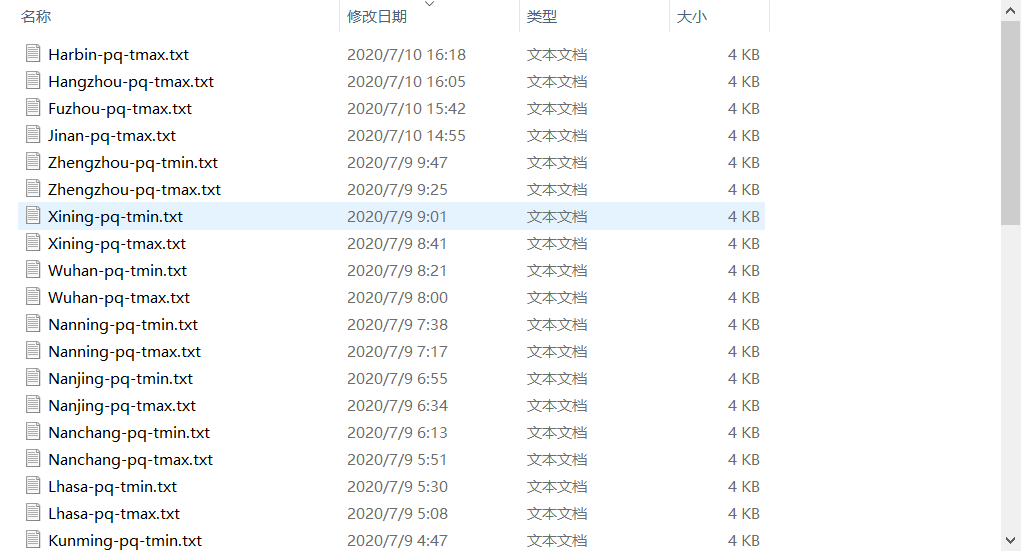
## 3.3 模型选择

1. **任务介绍**

编写函数自动确定pq值，达到自动化。

1. **技术要点**
   1. ARIMA模型参数p, q的判断；
2. **任务流程**
   1. 编写函数指定pq从0至7共64组情况进行运算；
   2. 确定指定数据集的最佳pq；
   3. 写入文件；
   4. 重复以上步骤，确定所有数据集的pq。
3. **成果展示**

将各个城市每一天的pq值都写入txt文件，预测天气时直接读取文件即可，避免实际使用时花大量时间确定pq。见图



## 3.4 预测一周天气

1. **任务介绍**

编写接口函数，获得指定城市和日期后，计算获得指定日期之后一周的天气数据。

1. **技术要点**

Python语法的基础使用；读写文件；获得一周的预测数据；转化为json数据

1. **任务流程**
   1. 读取相应文件获得pq
   2. 根据所选择的p, q值建立模型
   3. 预测一周数据
   4. 生成json数据
2. **成果展示**

生成的预测图：【如图3.5.1】json文件【如图3.5.2】

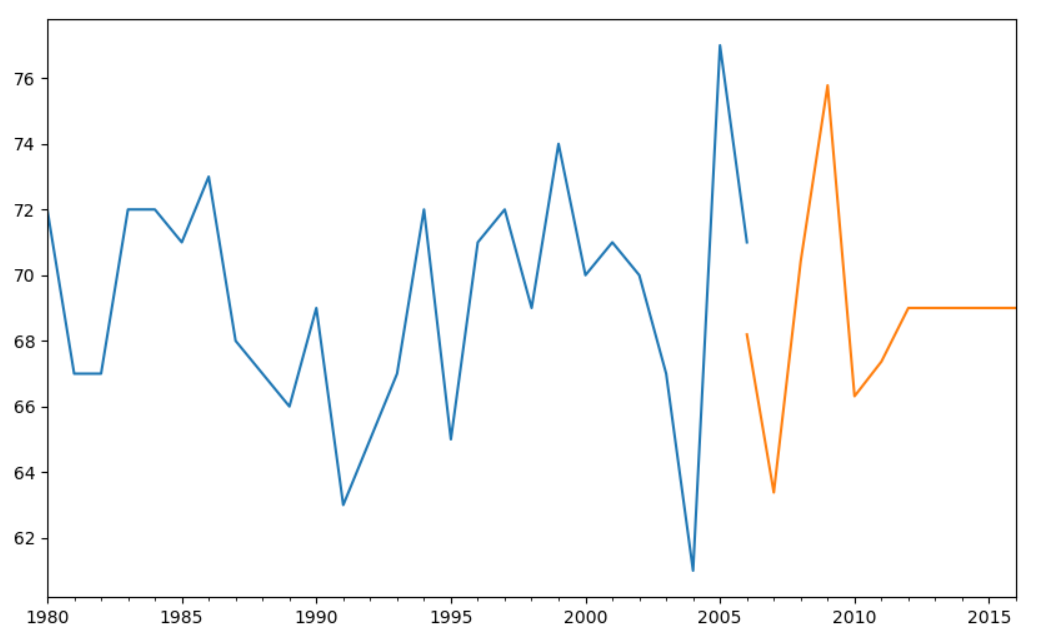


图3.5.1

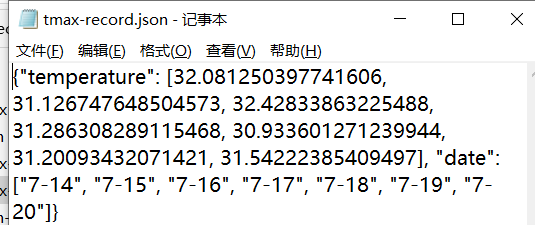


图3.5.

# 4 数据传输

## 4.1 任务介绍

将数据建模小组根据清洗后的数据集所生成的预测数据发送给前端，实现后端到前端的数据传输。用户在前端页面进行操作时，前端将用户的登录信息和用户选中的城市日期等信息发送给后端。

## 4.2 技术要点

1. Sqlite数据库的建立；
2. 对数据库中数据信息的存储和管理；
3. 对数据库中数据信息的查找和调用。

## 4.3 工作流程

1. SQLite3数据库是一款非常小巧轻量级的嵌入式开源数据库软件，也就是说没有独立的维护进程，所有的维护都来自于程序本身。由于其方便快捷，并且是Python中自带的数据库接口类模块，适合用户开发小型数据库系统。结合项目实际，我们所需要存储的数据量较小，因此选择Sqlite数据库；
2. 创建一个数据库并建立表格，使用sql语言设置表格属性：【如图4.3.1】

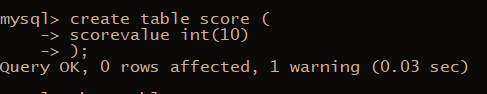


图4.3.1

将每个城市的气温预测数据依次建立一个表单【如图4.3.2】，存储至数据库中。表单名由城市名称组成，当用户在页面选择城市时，前端将用户的选择信息发送到后端，后端根据接收到的城市打开相应表单，然后读取数据并发送到前端显示在页面上。

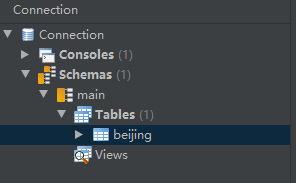


图4.3.2

1. 编写js文件对用户的操作进行判断，将用户的操作信息传回到后端，创建String对象和stmt对象，拼接SQL查询语句存到新建的String对象中，调用prepare函数，查询所得数据存到stmt对象中，循环读取stmt中的数据即可用于前端显示。

## 4.4 成果展示

前端根据后端发送的天气气温预测数据，生成相应的预测表格和气温预测图。【如图4.4.1】

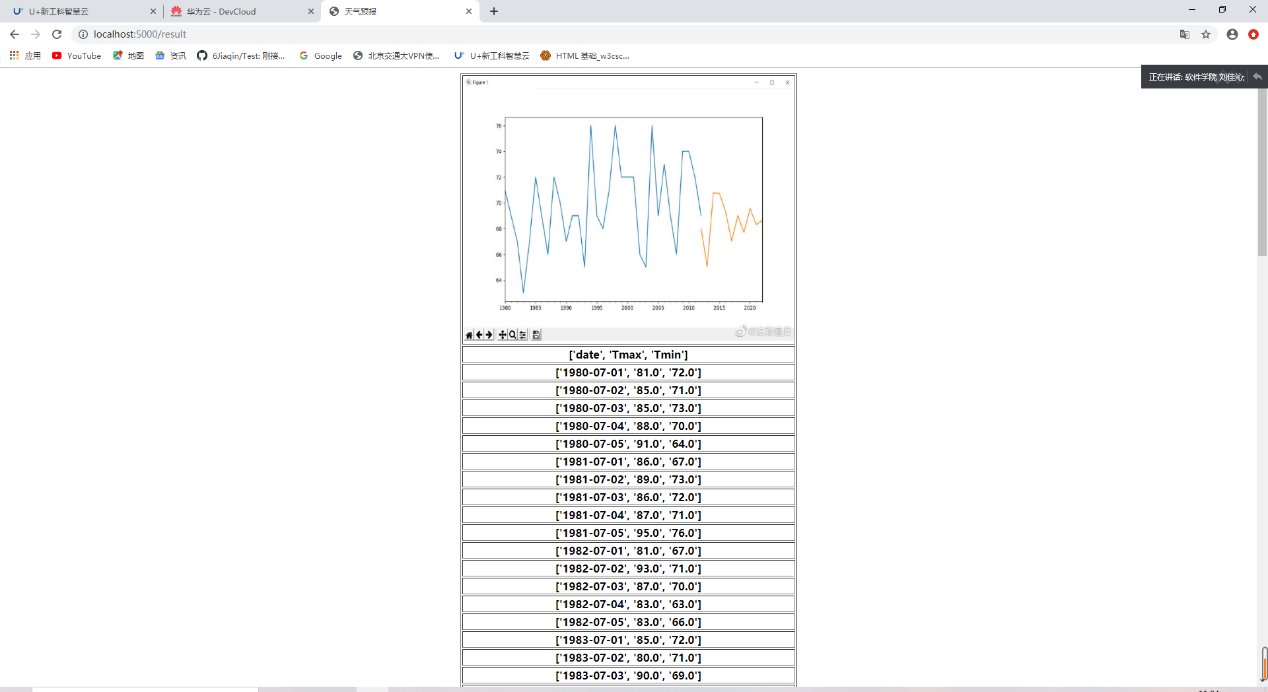


图4.4.1

# 5 前端界面设计

## 5.1 任务介绍

制作较为精美的用户注册登录界面，展示预测天气数据界面，以及用户管理界面，并实现前台页面与后台数据的交互。

## 5.2 技术要点

（1）使用Html制作简陋页面

（2）使用Css将页面进行美化

（3）使用JavaSpring给页面添加行为

（4）使用Vue+Springboot实现前后端分离

（5）使用Element让页面更精美

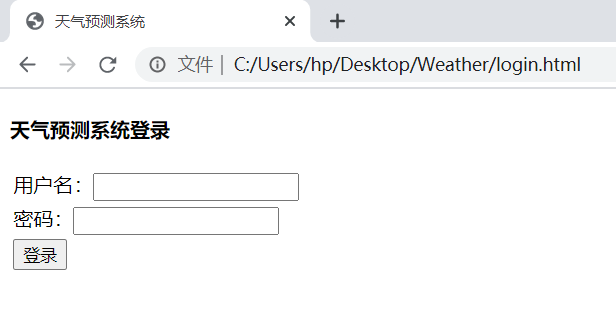
## 5.3 工作流程

**1. 学习Html知识**

**学习时间较长，基本掌握基础语法。**

**2. 制作简单的登陆注册界面**



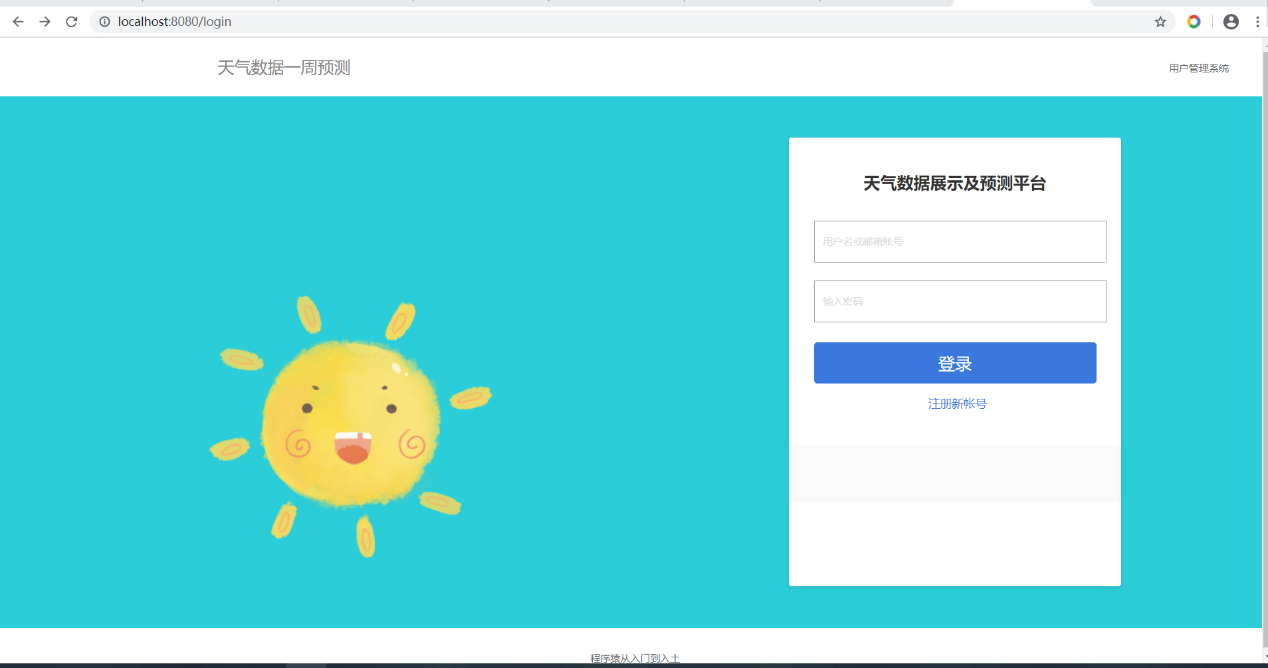


**3. 学习Css、js知识**

**学习时间较长，基本掌握基础语法。**

**4. 对登陆注册界面的美化**

****

****

**5. 学习Vue知识实现前后端分离**

**Vue-axios前后端分离跨域访问的实现**

**一、基本环境：**

**前端：**

**Vue；**

**axios；**

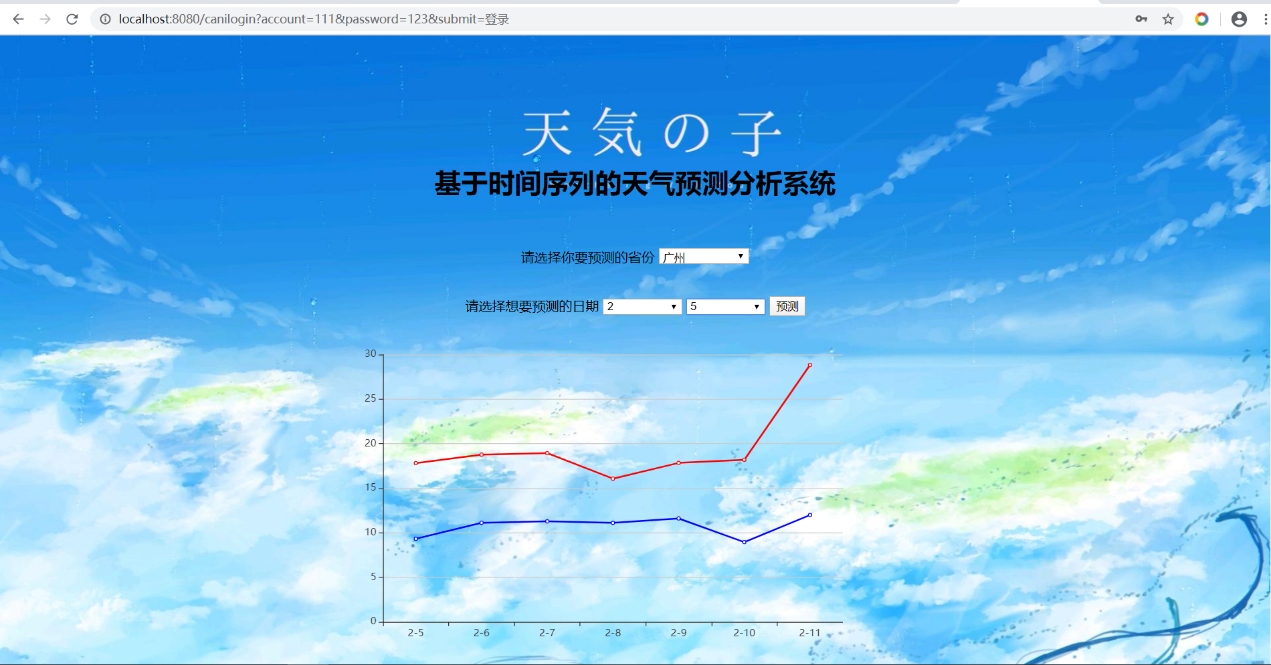
**使用脚手架构建项目**

**前后端数据采用json格式传输**

**二、前端配置**

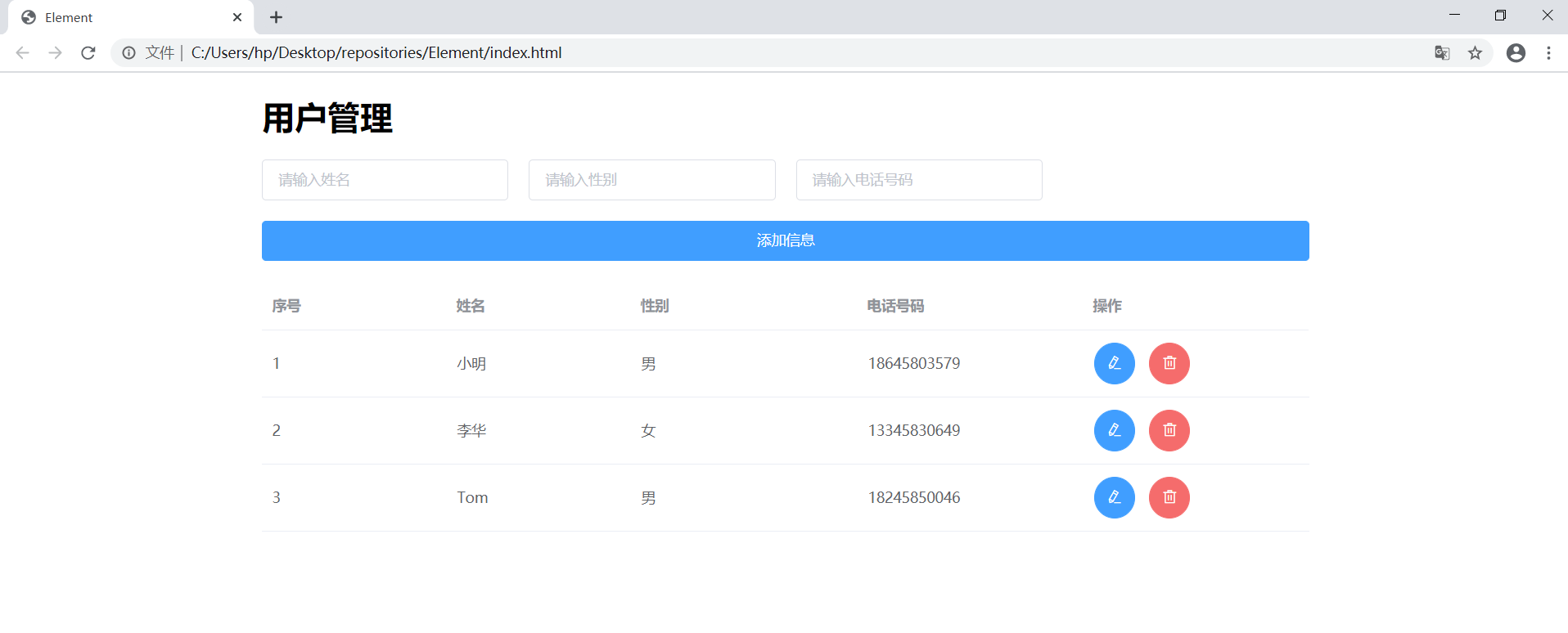
* **安装：npm install axios –save（切换至项目根目录，运行安装命令。因为之前安装过cnpm,我们使用cnpm使用淘宝镜像安装，速度会更快）**
* **在所需组件中引入axios: import axios from ‘axios’;**
* **在<script>中配置axios参数**
* **代理配置：打开config/index.js文件，在dev里的proxyTable对象修改内容，target的参数就是要访问的服务器地址。**
* **使用axios发送请求**
* **成功从接口取出数据**

**6. 制作天气预测的主界面**

****

**7. 使用Vue+element框架制作用户管理界面并实现增删改查**

**Element框架网址：**[**https://element.eleme.io/#/**](https://element.eleme.io/#/)



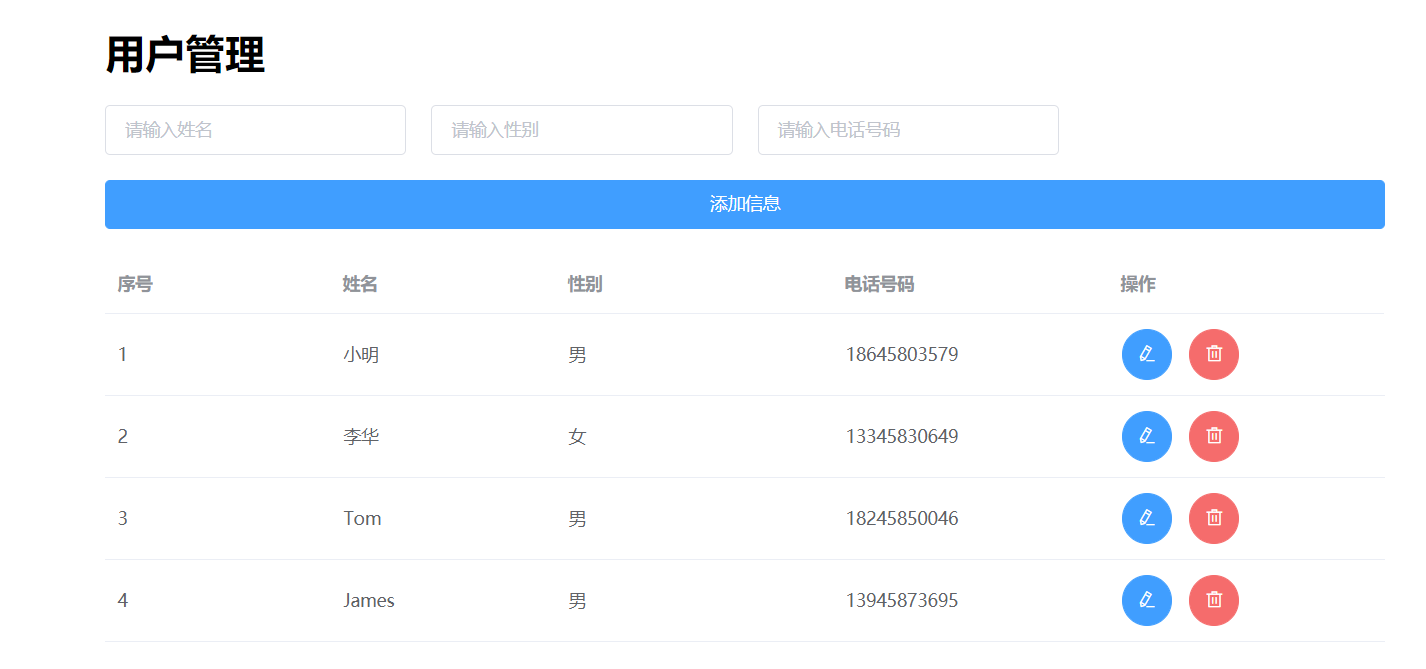
添加信息时需要输入姓名性别和正确格式的电话号码：



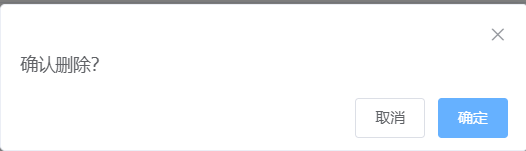




正确添加后：



删除某个用户（李华）：





修改信息（Tom）:





# 6 服务器端设计

## 6.1 任务介绍

接收前端界面输入的数据，调用python文件，利用其算法生成预测数据并传递到前端。编写整个项目的框架，以及数据库的连接和用户的增删改查等功能。

## 6.2 技术要点

1. 使用spring boot编写程序框架
2. 使用MySQL数据库进行用户管理操作
3. 通过路由来实现controller文件对html文件的控制
4. html前端通过axios来实现相关数据的传输
5. Controller后端通过@request以及httpservice来进行数据的接收

## 6.3 环境配置

1. Maven环境配置

从官网下载Maven文件，配置环境变量，并将其配置入IDEA中；

1. spring boot配置

在编译器中下载spring boot所需工具，即可新建spring boot项目。第一次创建项目时速度很慢，经研究发现是因为创建该项目需要从国外网站下载一些需要的配置，利用国内的镜像网站可以解决该问题。

1. MySQL配置

从官网下载MySQL软件，配置环境变量，并在命令行中初始化与创建数据库；下载navicat软件连接该数据库并对数据库进行操作；下载java连接数据库所需jar包并导入。

1. Jquery与Axios的引用

在html文件中加入<script src=””></script>中通过在src里输入相应的网址来引用封装好的Jquery与Axios代码包。

## 6.4 工作流程

1. 学习spring boot相关知识
2. spring boot调用python文件

使用Runtime.getRuntime().exec(args)调用python脚本，将python文件修改为对应的格式（含main函数），然后直接开启线程运行该文件，在该main函数中调用python函数生成预测数据与文件。利用BufferedReader接收生成的预测数据并转化成为java中的String数据，利用fastjson包将字符串转换为json格式的数据并传递到前端。

1. Springboot的前后端数据传输

当Ajax以application/x-www-form-urlencoded格式上传即使用JSON对象，后台只能使用@RequestParam 或者Servlet获取参数。 当Ajax以application/json格式上传即使用JSON字符串，后台可以使用@RquestBody或者@RequestParam获取。

当后端接收完前端的数据，响应一般也是返回json数据给前端，此时只需要在后端控制器Contoller类加上@ResponseBody即可。该注解用于将Controller的方法返回的对象，通过HttpMessageConverter接口转换为指定格式的数据如：json,xml等，通过Response响应给客户端。@Controller 与 @ResponseBody 结合使用返回json数据给前端，我们还可以使用@RestController替换他们，从而使代码更加的精简。

1. 学习MySQL相关知识

学习MySQL的创建与用户信息的增、删、改、查功能，并学习java与数据库的连接功能，连接数据库，从前端获取用户信息，如用户名和密码，并将其保存在数据库中；编写一系列容错功能，如将信息与数据库比对，向用户返回“用户名已存在”或“密码错误”等信息。

1. 学习Jquery相关知识

Jquery封装了一系列关于获取html控件内value与text值的函数，通过￥符号以及各种doucement.getElemengById.innerText可以获得各种需要的控件的文本值；通过msg可以实现后端的相应在前端的展示。