|  |
| --- |
| 组长：王磊 |
| 总结文档 |
| 指导老师：李宇 |

|  |
| --- |
| 组员：申淳元、燕怡楠、韦永剑、吴明昊  2020-7-13 |

目录

[1 项目开启 1](#_Toc45536268)

[2项目完成情况 1](#_Toc45536269)

[3组员任及工作量总结 2](#_Toc45536270)

[4项目进度 2](#_Toc45536271)

[5 项目用例图 2](#_Toc45536272)

[6 项目流程图 3](#_Toc45536273)

[7 成员感受 3](#_Toc45536274)

[7.1 项目组长王磊 3](#_Toc45536275)

[7.2项目组员申淳元 4](#_Toc45536276)

[7.3项目组员燕怡楠 4](#_Toc45536277)

[7.4项目组员韦永剑 5](#_Toc45536278)

[7.5项目组员吴明昊 5](#_Toc45536279)

[8 项目总结 6](#_Toc45536280)

[8.1 总体总结 6](#_Toc45536281)

[8.2 难点、困难总结 6](#_Toc45536282)

## 1 项目开启

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | 天气预测系统 | | |
| 计划开始时间 | 2020年6月29日 | 计划结束时间 | 2020年7月12日 |
| 项目说明 | 该项目使用了java web、flask、python进行了开发工作，使用Github进行项目版本控制 | | |
| 项目目标 | 基于时间序列模型对历史气象数据进行分析，数据采集 NCDC 气象官方网站。主要功能是对北京地区 1980 年至 2020 年的气象数据进行 ETL 清洗后，筛选出某年中每一天的最低气温，最高气温，然后使用 Python 对数据进行时间序列分析，预测一周的气温走势。最后，使用 web 终端对分析的结果进行可视化展示。 | | |

## 2项目完成情况

|  |  |
| --- | --- |
| 模块 | 完成度 |
| 天气预测 | 100% |
| 展示页面 | 100% |
| 页面间的跳转与联系 | 100% |

## 3组员任及工作量总结

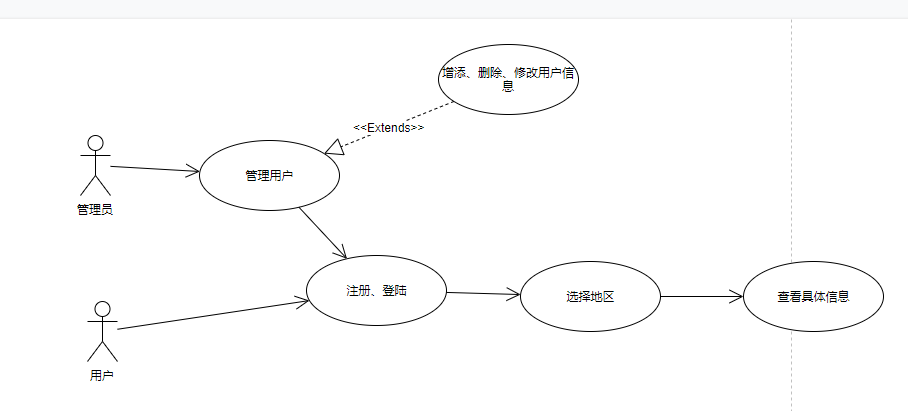
|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 负责部分 |
| 申淳元 | 18301105 | 研究特征清洗，虚拟机、服务器环境搭建，项目部署 |
| 王磊 | 18301107 | Prophet和ARIMA模型，前端H5网页，网络传输 |
| 韦永剑 | 18301109 | Java spring 学习,java编写，前端页面编写 |
| 吴明昊 | 18301110 | Flask学习，flask框架搭建，项目测试 |
| 燕怡楠 | 18301113 | 研究时间序列预测模型，包括LSTM、GRU、 SARIMA等模型 |

全组人员均积极参与项目，按照分工按时完成预期任务，每个人都为项目完成做出了巨大的贡献。

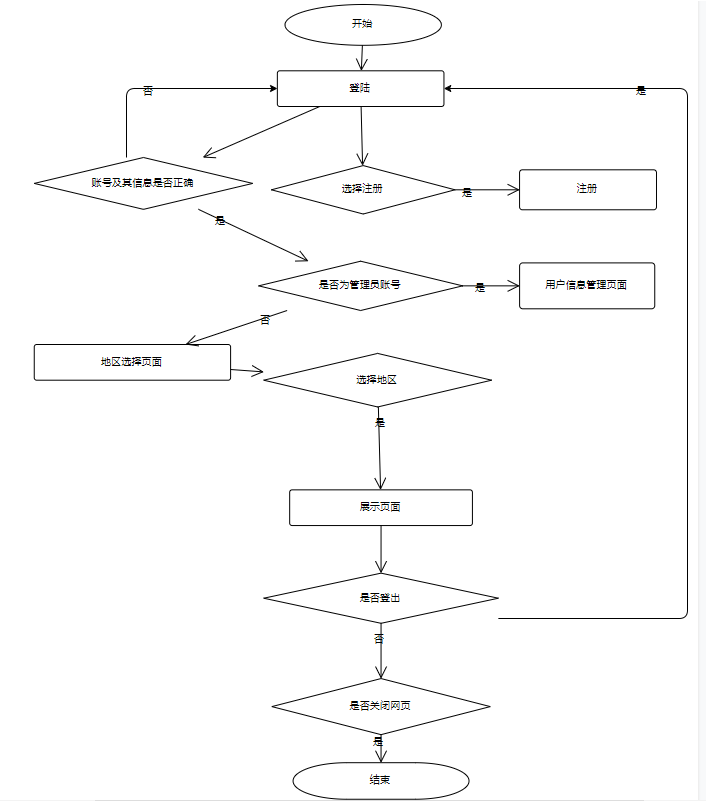
## 4项目进度

|  |  |  |
| --- | --- | --- |
| 项目进度 | 开始时间 | 结束时间 |
| 环境搭建 | 2020.06.29 | 2020.07.01 |
| 模型构建 | 2020.07.01 | 2020.07.12 |
| 前端开发 | 2020.07.01 | 2020.07.11 |
| 项目整合 | 2020.07.11 | 2020.07.12 |

## 5 项目用例图



## 6 项目流程图



## 7 项目创新与拓展

### 7.1 虚拟机利用

我们在小学期刚开始时，按照任务指导在本机配置Ubuntu系统的虚拟机，成功安装并实现了虚拟机的联网，但是由于不知道后续操作，因此暂时搁置了一段时间。在暂时放弃虚拟机一段时间后，我们进行了第二次迭代，利用虚拟机将项目部署至云端服务器。我们在云端安装了Ubuntu系统，利用x-shell进行本机和虚拟机间的文件传输。并且在虚拟机中配置了项目中所用到的许多包和库文件然后在虚拟机中运行程序，将项目部署至服务器，让网页可以从公网访问，实现天气预测网站制作的真正意义。

### 7.2 Javaweb作为前端框架

我们在第一次迭代时因java spring学习成本较高，入门相对困难，因此我们首先选择以flask作为前后端框架，加快项目进度，采用flask搭建路由，实现网页间跳转。经过调试运行，纯python版本功能正常，且也已经部署至服务器，可以实现公网访问。

在第二次迭代时，我们负责Java开发的韦永剑取得了一定进展，开始进行前端从flask到Java的迁移，是循序渐进的努力学习的结果。 最后，我们采用Java web框架，以html,css,js等网页元素进行网页编写，利用tomcat作为项目服务器，将java前端部署至服务器，最终实现了既定目标。在该过程中，通过先对flask的学习，为后续java版本的成功实现积累了经验基础。

### 7.3 云端服务器部署

当主要目标完成后就需要将项目部署到服务器上，以便在任何时间地点我们都可以使用它。首先服务器选择了购买阿里云ecs，省去了申请公网ipv4，做ddns和穿透以及转发的步骤，且有成熟的控制后台、快照系统以及防火墙，相对简单便捷。为了隔离不同的服务并能够更快更稳定，使用了虚拟环境+web server+niginx反代+进程监视的体系。

初次迭代使用了anaconda+uWGSI+ningx+Supervisor。Flask前端项目可以运行及跳转，但涉及到pandas的后端由于未知原因的numpy问题导致无法运行。

第二次迭代对第一次进行了大改，使用virtualenv + virtualenvwrapper + Gunicorn + ningx + Supervisor，配置难度有所降低，但是后端可以正常运行。同时加入了java前端，正确配置了jdk及tomcat以及解决跨域问题后前端可以正常运行并连接后端，实现全天候的外网访问。

### 7.4 算法模型

我们尝试了ARIMA，fbprophet，SARIMA等多种模型，并将各个模型算法的误差值进行横向比较，最终选择了SARIMA算法作为最终的模型，进行未来一年的数据预测。

在具体选择SARIMA模型进行预测时，仅采用了单变——气温来预测，因此并没有拓展升级为SARIMAX模型，原因在于数据的建模会因为多变量的不断修正而耗费过多时间，毕竟家用电脑实在是算力有限。

然而，即便是单变量的SARIMA模型，在进行季节性周期模型学习建立的时候，会因为一个周期中数据过多但是拥挤，使得周期与周期之间不够融洽，模型虽然不断修正学习，但是因为一个周期中数据量大，需要修改的要素会过多，于是会因为学习内容过多导致拥挤，模型修正虽然可以进行且预测准确，但是在时间要求下，需要改进该模型，必须适应小学期的节奏，否则模型就不是一个好模型，而且存在闰年情况，周期不能恒定为365，需要增加额外处理，损失更多时间。

针对以上问题，考虑通过牺牲小部分准确度来获取更高速的效率。于是在阅读大量资料案例以及自主尝试之后，发现固定频率可以由一天更改为三十天。同样的，因为月份之间的相同日期之间间隔会因为每个月的长短不同而有所差异，为了解决该问题，学习了关于INDEX时间序列间隔频率的相关知识，得知将freq赋值为预设值MS后，会将频率固定为每月第一天，那这在一定程度上就代表频率自动符合了每个月的长短甚至闰年变化，恰好符合我的要求。因此在对数据的处理时，我修改了清洗方法，将所需时间段的时间全部抽出，然后根据末尾的日期来分开存储，并且将日期存储后抹除，只保存年月信息，同时利用循环将日期存储为文件名，则一个城市的信息被我拆分成31天的信息，没有则为空，不影响后续运算。而值得注意的是，在上述过程中，虽然简化了模型，但是由于具体的日期之间相互分割，因此建立的模型不可避免出现理论漏洞，那就是日期与日期之间的印象变化被削弱。不过因为历史数据上日期与日期虽然分隔，但是实际上对彼此的影响已经造成，所以训练学习的过程中不会出现具体偏差，反而会更加准确。然而在预测结果中，因为可能存在的偶然因素，会导致日期模型之间出现轻微错位，但是可能温差变化增大，出现误差，这一点是牺牲效率后无法规避的潜在风险。而相应带来的好处就是，原本需要四五小时才能建立的预测模型，甚至不能在这漫长时间里完成预测，仅仅是学习完毕，优化后的模型中，只需要数十秒就可以轻松跑完多个城市的模型训练、学习与预测，并且整合输出到指定文件，效率几乎是成倍的提升，大大节约了不必要的计算资源，也实现了模型优化的意义，早早为前端提供好需求数据。

在调用以上处理完的数据时，我就可以根据文件名（即日期）来直接放入建模预测，因为MS自动实现了月份之间频率不同的间隔，因此可以根据此顺利建模，则得到历年每月某一天的气温预测值。由于MS会自动补全年月日信息，且日期设置为01，因此简单存储后读取，将’date’时间序列单独取出，通过抹去01，写入文件名获得真实日期信息，重新存储到跳板模块（类似于缓冲区域，为自行定义），最后写入目标城市气温预测文件，通过sort\_values中对index时间序列作为排序目标实现所有无序预测值的整合，得到可以输出到前端的文件，命令举例：北京\_max.csv，通过前端的文件读取，可以直接展示到天气预测界面。

### 7.5 利用RBAC2模型进行用户角色设管理

RBAC 是基于角色的访问控制（Role-Based Access Control ）在 RBAC 中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。这就极大地简化了权限的管理。这样管理都是层级相互依赖的，权限赋予给角色，而把角色又赋予用户，这样的权限设计很清楚，管理起来很方便。

我们将用户角色分为user,vip,admin,root四种，使用RBAC2模型。RBAC2是RBAC0模型的扩展，在RBAC0的基础上添加了约束。主要约束有：

1. 互斥角色，同一个用户不能授予互斥关系的角色，如：不能同时是vip和admin。
2. 基数约束，一个用户拥有的角色是有限的，在本项目一个用户只能拥有一个角色。
3. 先决条件约束，用户想得到高级权利，必须先拥有低级权利，用户必须先拥有user角色才能通过管理员调整提升为其他角色。

### 7.6 利用Mysql数据库进行信息存储

我们利用Mysql数据库作为用户信息存储数据库。在flask中利用flask\_sqlalchemy 包将数据库操作对象化，从而可以以对象的操控方法，简易地对数据库进行增删改查等操作。

数据库中存储用户的username，password，kind三种信息。

### 7.7 利用CORS(跨域资源共享)进行flask与java间信息传递

我们在将Java前端部署至服务器后发现了前端和后端不能成功握手的问题。为此我们查阅了许多资料，并尝试过很多的方法，如：修改tomcat域名和端口，修改服务器端口配置，防火墙等。

最后采用HTTP CORS跨域资源共享的方式，在flask中导入flask\_cors，调整配置为可以接受前端域名访问，让flask后端和Java web 前端成功进行握手，可以进行彼此间的数据传递。

## 8 成员感受

### 8.1 项目组长王磊

十四天的小学期，时间不长，但是经历和学习了很多。从最初始的配置环境，到ARIMA模型的学习，参数调整，转用fbprophet，到后面前端网页的搭建，编写，网络和数据库连接。这次小学期让我们大多数人处于从零开始的懵逼境地，很多东西都是从头开始学，中间踩过无数的坑，挠过无数次头，熬了14个夜。压力确实很大，不过在压力之下，大家也有了令人惊喜的进步。分给每一个成员的艰巨任务都得到了解决，组员们总是在意想不到的地方给你惊喜。最终的结果是我们每一个人努力的结果，虽然还有许多可优化的地方，还有许多未完成的功能，但我认为这是一个值得我们骄傲的小学期项目，也是我们现阶段尽力交出的最好答卷。

### 8.2项目组员申淳元

为期两周的小学期已经接近尾声，两周时间内我们分工合作，一同完成了基于大数据的天气预测。两周之内我遇到了很多困难，但同样也有很多收获。

首先是合作本次是5人一组，与之前课程中大部分都要独自完成的作业、或最多二三人的小组可以说完全不同，再加上疫情原因，在分工，合作与讨论上带来了很大挑战。小学期中让我更好的适应了多人间的合作。接着是python和机器学习，作为后端的一员，python算是新接触的语言，机器学习更是完全不懂，在第一次迭代的过程中我对它们进行了了解，虽然最后出现了很多问题并没有使用我的方案，但是还是学到了很多东西。在后端代码任务重点转移后，我对项目的部署进行了接触，ubuntu server属实对新手非常不友好，这点和windows server没法比，在部署中遇到了大量的问题，从依赖到镜像来源到不知原因的bug，换了数种虚拟环境软件和包管理器，web服务器和反代软件，以及自动监控等等，接触了让人头疼的uwgsi配置文件，了解了运行及命令的相关知识，且环境正常之后的跨域访问问题。通过14天的学习生活，我感受到了英语材料检索、去读官方文档的重要性，以及算是首次感觉到了日志的作用。一个小学期下来，甚至给了我一种以后不用选修linux与网络编程这门课的错觉。

小学期中我除了获得了知识，也开拓了眼界。从未开课时的资料中我看到了很多以前不知道的东西，开课后老师也经常会花大量的时间来给我们将一些东西，其中好多都是没有听说过的，包括课上大佬们所讨论的东西，对我来说都是全新的。

总体而言，14天的小学期给我带来了巨量的信息，我还应该进一步消化，提升自己。

### 8.3项目组员燕怡楠

在本学期的学习过程中，很高兴能在李宇老师的指导下学习python并实现对数据的清洗处理与建模分析，也很高兴能够与其他四位同学组成小组、分工合作，最终完成任务目标。因为之前许多文档的介绍着重SARIMA的改进历程，在这里主要突出描写我整个的数据预测经历。

在最初的分工之中，我首先被安排为前端开发，负责网页、服务器、微服务框架的，最开始很顺利的配置了环境，安装了各种所需工具，不过由于一位同学电脑环境复杂，长时间不能正常安装运行，最终决定我与其更换岗位，因此开始了后端开发旅程，内心颇有些对未知的忐忑，但也有一番求知的欲望在蠢蠢欲动。

果然由于对任务目标的不够了解，在虚拟机方面耽误了过多时间，也出现了注册表杂乱等问题，同时，在尝试为虚拟机配置网络使其与真实机或其他虚拟机联机建立集群的过程中，因为虚拟机占用过多内存，导致电脑常常死机，着实费了一番功夫。

不过小组工作的进度不能被一件不甚重要的事情耽搁，于是我便继续进行后续任务，其中主要负责的是考察各类预测模型，比较其工作效果与效率，做到能够快速分析大量数据并建立模型分析预测。在此过程中，我先后接触了ARIMA、LSTM、RNN、GRU、SARIMA等不同模型，其中ARIMA与SARIMA属于同一类型，而LSTM、RNN以及GRU则关联较深，涉及时间序列的深度神经网络学习。而在这一切之前，我还了解了有关线性回归预测的模型，该模型采用一年中的每一天作为学习与预测对象，在处理时，需要对历年的某一天进行分析计算，但是由于线性模型很难预测其他天带来的关联影响，会产生理论漏洞，最终被我很快抛弃。

关于LSTM与GRU的学习花费时间最多，虽说最后并未采用，但是在学习的过程中使我获益匪浅，在LSTM及其变种GRU中，门与门的确十分难以操作，这使得我这个初学者在手动编译函数的过程中受到了极大地挑战，最终草草了解了几次实际内容后，主要使用tensorflow和keras等函数库进行调用预测，不过在keras中model所调用的简单函数里，设置完环境参数后，可以方便计算model. fit所产生的误差值，而且epoch时的多轮学习可以通过loss与val\_loss的相对变化来判断参数的设置是否合理，而问题出现在model. predict函数输出结果，因为在调用fit时采用了generator手动编译，导致在predict时长时间无法输出预测结果，甚至采用自带的fit也无法进一步预测，只能在现有范围内训练，经历过一晚上的数据挂机，predict依然不能产生结果，迫于暂时无法改进的效率不得不考虑更换模型。

最终采用的SARIMA是在ARIMA基础上进行的拓展，主要是改进在季节性方面，通过温度的季度性变化来建立模型预测变化，在这里最初选择了以一年三百六十五天为周期，进行一年的预测，只是原有的ARIMA模型如果添加了365的周期，会导致模型效率极其低下，在一次实际预测时，经过三小时计算依然未能完成一座城市预测值所需的一半计算，所以模型必须简化。

经过改进，最终我将365天拆开使用，每个月对应的那一天，进行12周期循环预测，对每月的那一天分别预测，然后将结果汇总排序，就使得运行效率大大提升，并利用两个小时处理完所有省会城市，最终传输到前端。

在这学期的漫长的数据处理工作中，熟悉了各种数据模型，将会使我对于以后数据处理更加得心应手。

### 8.4项目组员韦永剑

经过十四天的小学期的学习与实践，我感觉收获良多。

1. 更加理解了要如何进行团队的沟通，也懂得了如何在一个团队中更好的贡献自己的一份力量。在老师和组长的带领下，我们这个小团队齐心协力克服了项目中的一个个挑战，攻克了项目中一个个难关
2. 学会了如何更高效更准确的查找一些有用的知识。比如我之前从没留意过的学校图书馆，又比如老师所说的中国知网上的博士论文·············································等等。老师介绍的一个个学习网站都让我大开眼界。同时老师讲授的一些课上没学到过得知识也让我获益匪浅
3. 掌握了一些专业技能。像GitHub的分布式管理，python的使用，对java spring MVC 的知识有所了解，有关java中http请求的相关知识，myeclipse的使用，Tomcat的有关配置，对前后端分离式开发有所了解等等

### 8.5项目组员吴明昊

通过这次的项目实训，我学习了flask的相关知识，以及网页的简单开发，对我以后前端的开发提供了一定的帮助。这次实训过程中，我们经历了环境配置的失败、程序不断报错，等等一系列的问题，确实让人感到疲惫，再加上老师几乎没有给完整详细的教程，真的可以说整个项目是从0开始进行的，大家不断地在网络上进行着相关资料的查询以及学习，在某种程度上着实锻炼了我们的能力，也感谢我们的组长，对我们的帮助，没有组长的帮助以及领导，我们团队是不可能完成这次项目的！

## 8 项目总结

本次项目本小组成员认真分析任务指导，完成上面指定各项任务。并任务指导的基础上添加了大量的拓展，包括更换模型，网络连接，以精美的网页将数据进行可视化展示，将网页部署至服务器，将用户信息存储至，添加用户管理和权限管理功能，所有功能均已如预期完成。