

江苏农牧科技职业学院

毕业设计（论文）

题目 农产品质量追溯数据可视化子项目的设计与实现

姓名 姜 赵

学号 201710177

二级院（系）部 农业信息学院

班级 17 计算机应用技术 1

专业 计算机应用技术

指导教师 谢晓伟 职称 讲 师

指导教师 陈君艳 职称 副教授

二〇一九年十二月

江苏农牧科技职业学院学生毕业设计（论文）材料目录

序号	名 称	备 注
1	毕业设计（论文）诚信承诺书	
2	毕业设计（论文）正文全文	
3	毕业设计（论文）任务书	
4	毕业设计（论文）开题报告或实施方案	
5	毕业设计（论文）指导记录	
6	毕业设计（论文）中期检查记录表	
7	毕业设计（论文）指导教师意见表	
8	毕业设计（论文）评阅教师意见表	
9	毕业设计（论文）答辩记录表	
10	毕业设计（论文）成绩评定表	
11	毕业设计（论文）附件资料	

江苏农牧科技职业学院学生毕业设计（论文）诚信承诺书

论文题目	农产品质量追溯数据可视化子项目的设计与实现		
学生姓名	姜赵	学号	201710177
所属专业	计算机应用技术	班级	17 计算机应用技术 1
指导教师姓名	谢晓伟、陈君艳	职称	讲师、副教授
学生承诺			
<p>我承诺在毕业设计（论文）活动中，遵守学校有关规定，恪守学术规范，本人毕业设计（论文）内容除特别注明和引用外，均为本人观点，不存在剽窃、抄袭他人学术成果，伪造、篡改实验数据的情况，如果有违规行为，我愿意承担一切责任，接受学校的处理。</p>			
<p>学生(签名): 姜赵</p> <p>2019年 9 月 16 日</p>			
指导教师承诺			
<p>我承诺在毕业设计（论文）活动中，遵守学校有关规定，恪守学术规范，经过本人核查，该生毕业设计（论文）内容除特别注明和引用外，均为本人观点，不存在剽窃、抄袭他人学术成果，伪造、篡改实验数据的现象。</p>			
<p>指导教师(签名): 谢晓伟 陈君艳</p> <p>2019年 9 月 16 日</p>			

农产品质量追溯数据可视化子项目的设计与实现

学生：姜赵 指导老师：谢晓伟、陈君艳

【摘要】 江苏省农产品质量追溯作为我省农产品质量安全监管的重要手段之一，其在追溯、监管、监测、执法等各个环节采集了大量的农业生产主体、中小农户、合作社以及各类农产品的生产、检测等数据。本文就如何高效的应用这些数据，提出了基于农产品质量安全信用管理数据可视化系统的设计及实现方案，将这些数据通过处理，以图表对比的方式直观地呈现给用户，同时对这些数据进行分析、筛选，发现其内在的规律，挖掘其潜在的价值，为农产品的安全生产与监管提供科学的决策和预判，是助农举措有效的体现。

【关键词】 Django 架构；数据可视化；多源异构。

Design and implementation of agricultural product quality traceability data visualization sub project

【Abstract】 It is one of the important means of agricultural products quality monitoring and supervision in Jiangsu Province. However, the utilization rate of these massive data after collection is very low. In view of this problem, this paper proposes a design and implementation scheme based on the agricultural product quality and safety credit management data visualization system, which can intuitively show the massive data to the users in the form of chart through processing, and at the same time, the data are processed and analyzed to find its inherent rules and tap its potential value, To bring scientific decision-making and prediction for the production of agricultural products is an effective embodiment of measures to help agriculture.

【Keywords】 Django architecture; Multi-sourced heterogeneous; Data visualization

目 录

引 言	1
第一章 概述	2
1.1 项目背景	2
1.2 主要任务	2
1.3 系统架构	3
1.4 论文组织结构	3
第二章 相关技术介绍	5
2.1 应用框架概述	5
2.1.1 应用框架选择	5
2.1.2 Django 架构与 Spring 架构的比较	7
2.2 PYTHON	7
2.3 PANDAS 库	8
2.4 MATPLOTLIB 库	8
2.5 ECHARTS	8
第三章 系统需求分析	10
3.1 系统总体需求	10
3.2 数据预处理需求	10
3.3 数据存储需求	11
3.4 数据可视化需求	11
3.5 其他需求	12

第四章 系统设计	13
4.1 系统概要设计.....	13
4.2 系统架构设计.....	13
4.3 功能模块设计.....	14
4.3.1 系统管理模块.....	14
4.3.2 数据处理功能.....	14
4.3.3 数据存储模块设计.....	16
4.3.4 可视化模块设计.....	16
第五章 系统功能实现	18
5.1 地图的展现.....	18
5.2 柱状图的展现.....	19
5.3 堆叠图的展现.....	20
5.4 柱状图和折线图二合一的展现.....	20
5.5 仪表盘的展现.....	21
第六章 总结与展望	23
6.1 总结.....	23
6.2 展望.....	23
参考文献	25
致 谢	26

引言

随着我国经济社会的不断发展，人民生活水平稳步提高，然而食品安全问题却层出不穷，与我国全面建成小康社会的总体目标不符，严重侵害了老百姓的权益，危害了老百姓的生命安全。党中央对此要求从问题的源头进行有效治理，明确地指出农产品是食品安全问题的关键源头，其中农业起到基础性的作用，所以需要抓好我国的农产品质量问题。在这一领域中采取严格的问责及监管措施来保证我国的群众能够享受食品安全。针对前两个“最严”，江苏省通过信息技术手段，搭建了江苏省农产品质量追溯一揽子信息管理系统，采用最严谨的标准对农产品进行溯源管理，对每一份走上百姓餐桌的农产品进行最严格的监管，起到了良好的源头治理作用。

一方面，大力实现我国农产品领域的质量追溯目标，另一方面，还需对各种数据（包括农产品、农户、合作社和相关企业等）进行广泛的采集，除了可利用它们来进行溯源之外，也能够基于数据可视化途径来扩展其它各个领域里的深入应用，比如纳入监管范围的地区和企业、不同地区不同农产品的密集度、各类农产品的畅销度等，让系统使用者能够一目了然，提供有力的决策支持，甚至可以针对不同的农产品的生长因素数据，建立农产品生长模型，从而提高农产品的质量。

本论文基于农产品质量安全信用管理系统，依托江苏省大学生创新创业训练计划，数据范围覆盖江苏省农产品质量安全追溯平台、江苏省农产品质量安全平台、江苏省农产品质量安全监测信息系统、江苏省农业行政执法信息系统，采用大数据分析相关技术和数据可视化相关技术开展研究，实现基于农产品质量追溯的数据可视化原型系统设计。

第一章 概述

1.1 项目背景

近年来，食品安全的问题日益严重，而农产品作为食品的重要原料，其安全性也越来越受到人们的重视，如何贯彻落实党中央提出的“源头治理”的方针，众多的专家、学者以及行业监管部门将此提上了议事的日程，农产品质量的追溯作为一种手段，成为了破解这一命题的方法。基于上述观点，江苏省农业农村厅依托部分高校，共同搭建了以江苏省农产品质量追溯为背景的一揽子信息化应用系统，提供了追溯系统、信用管理、监管系统、监测系统、执法系统、溯源查询等功能，成为了农产品质量监管的一件利器，保障了全省百姓的农产品质量安全。

本文指导教师作为仪征市农产品质量安全信用管理系统项目负责人及团队成员，承担了农产品质量安全信用管理系统的开发任务，该系统通过数据交换的方式从江苏省农产品质量安全追溯平台、江苏省农产品质量安全管理平台、江苏省农产品质量安全监测信息系统、江苏省农业行政执法信息系统等系统中获取其用于信用认定的数据，给本文的研究和编写提供了极大的便利。

农产品质量安全信用管理系统采集了大量的农产品生产单位、个人信息，以及各类种植农产品生长数据，随着平台应用的不断深化，数据存储技术的不断成熟以及移动互联网等高速发展，平台数据呈现了大爆发的趋势，且在不同程度上向电子化转换，如何有效地利用这些采集到的数据成为了一个困扰研究者的问题。本文所设计的基于农产品质量追溯的数据可视化系统，根据当前数据利用率低的不足，结合了现有知识，使用了最新的各种数据可视化技术及一些简单的数据挖掘技术，将海量数据通过技术手段，以可视化的形式呈现在人们的眼前。

1.2 主要任务

在当前的工作中，系统地分析处理了来自农产品质量安全信用管理系统的一系列重要异构数据信息，着重地加工及挖掘与之相关的数据资源，实现一个 WEB 可视化系统，对内能够提供多样化的数据展示，对外能够提供数据共享服务。

本文的主要任务包括：

(1) 对数据资源进行 ETL 处理。基于用户和可视化方面的实际需求，结合各种来

源的数据集情况，清洗及处理各类原始数据，同时进行数据转换，并且实现数据的有效整合。

(2) 可视化系统设计与实现。基于 Django 的应用框架，设计并实现江苏省农产品质量溯源数据可视化系统，建立不同的可视化模型，将 ETL 后的数据资源映射其中，对内进行可视化图表的展示或者陈列展示，对外提供数据共享。

(3) 将一些数据模型构建起来。主要结合研究工作方面的具体需求，清洗所得到的海量数据信息，同时进行数据的挖掘，再将相应的数据模型有效地提炼出来，使各类用户决策可获得充分的支持。

1.3 系统架构

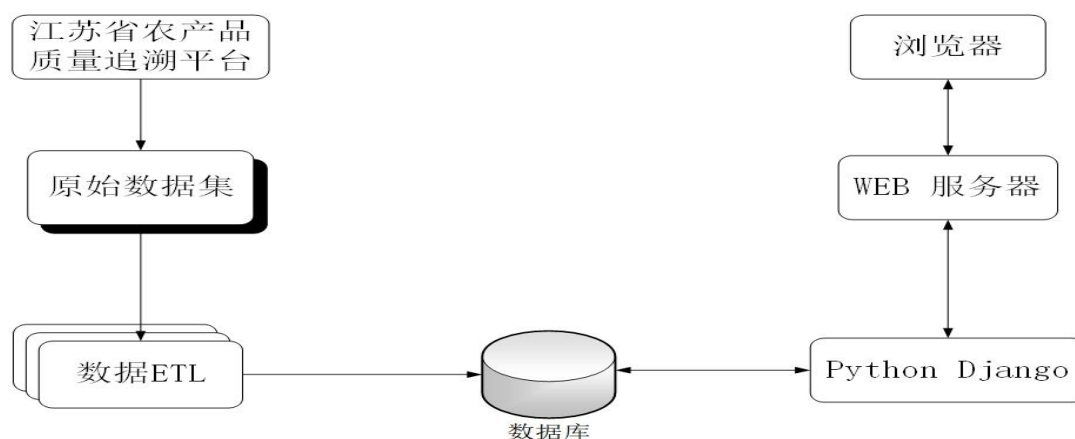


图 1-1 系统架构图

1.4 论文组织结构

本论文由六个章节组成，其组织结构具体如下：

第一章概述。主要介绍了基于农产品质量安全信用管理系统的数据可视化系统设计与实现课题的研究背景及其意义，提出了主要任务，简要介绍了本文设计系统的架构。

第二章相关技术介绍。介绍了本文应用的 Django 框架、设计语言 Python 和其在图形化方面应用比较成熟的各类库。

第三章系统需求分析。从系统的总体需求、数据预处理需求、数据存储需求、数据可视化需求和其他需求等五个方面，对农产品质量安全信用管理系统数据资源的处理需求和数据可视化展示需求进行了分析，确定了系统要实现的主要目标。

第四章系统详细设计。从系统概要设计、系统架构设计和功能模块设计三方面入手，对农产品质量安全信用管理的数据可视化系统进行总体架构设计，并对系统功能的详细

设计进行阐述。

第五章系统功能实现。介绍系统功能的实现方法和过程，详细介绍部分用例代码，如地图、柱状图、堆叠图和仪表盘等功能的部分实现代码以及实现效果。

第六章总结与展望。对本文的研究内容及研究方法等进行总结，并指出本系统设计实现的不足之处，以及下一步工作中需要的改进方向。

第二章 相关技术介绍

2.1 应用框架概述

2.1.1 应用框架选择

所谓的框架是指开发系统的时候，通过结构化方式来合理地描述某些领域所具有的层次逻辑，此外，还固化其中的某些通用性的模型、代码和技术文档等，以达到高度复用性的目标。目前基于 Python 的成熟 WEB 应用框架有很多，诸如 Django、Tornado、Flask 和 Twisted 等，本文选择的是 Django 框架，因为 Django 是基于 Python 开发的开源架构，具有完善的 ORM 关系映射、强大的路由映射功能、完善的视图模板的实现、健全的后台管理系统和强大的缓存支持。

其中，Django 架构作为一类软件架构模式，主要是以 Model - view - controller（其缩写为 MVC）作为基础，但为了各组件间保持松耦合关系，在 Django 中，由于框架自行处理了控制器接受用户输入的部分，因此，其中对于 Model（也就是模型）、Template（也就是模板）和 Views（也就是视图）尤为重视，所以也被称作所谓的 MTV 模式，具体见下面所列的信息：

表 2-1 MVT 各层次的职责

层次	职责
模型（也就是 Model），对应于数据存取层	主要是处理数据方面的各项事务，包括有效性的验证和数据的存取，此外还涉及到数据关系及行为等方面。
模板(也就是 Template)，对应于表现层	主要是处理和表现相关的各类决定，比如在不同的文档（及页面）中予以显示。
视图（也就是 View），对应于业务逻辑层	进行模型的有效存取，对模板逻辑进行调取。视图可起到沟通上述二者的功能。

由此可知，利用 Django，可以分解 MVC 里的相关视图，由此得到 Django 模板及视图，由此即可合理地确定“如何进行数据的展现”以及“展现时所需要采用的是哪一部分数据”等两个问题，因此，除了包括内置模板之外，还可基于真正的情况进行模板的替换。此外，也需利用 URL 分发器来分发各个 URL 页面请求至相应的 View 处理，

再由后者进行 Template 及 Medel 的调用，与之相关的响应模式见图 2-1:

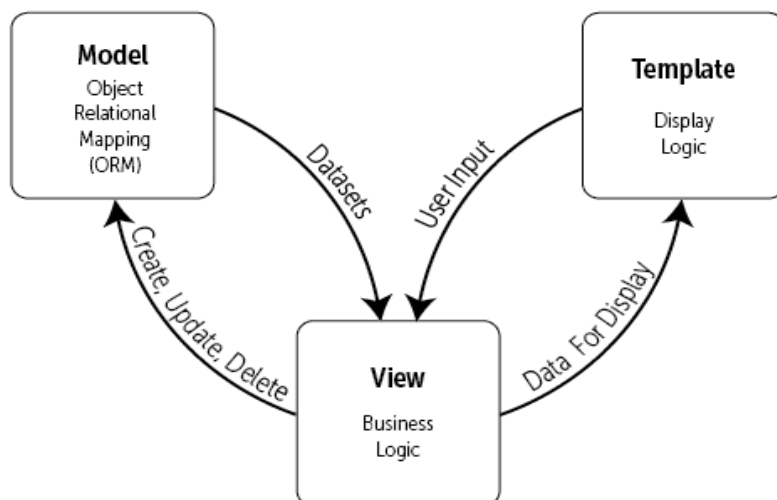


图 2-1 MVT 响应模式简易图

Django 的工作机制一般如下图所示:

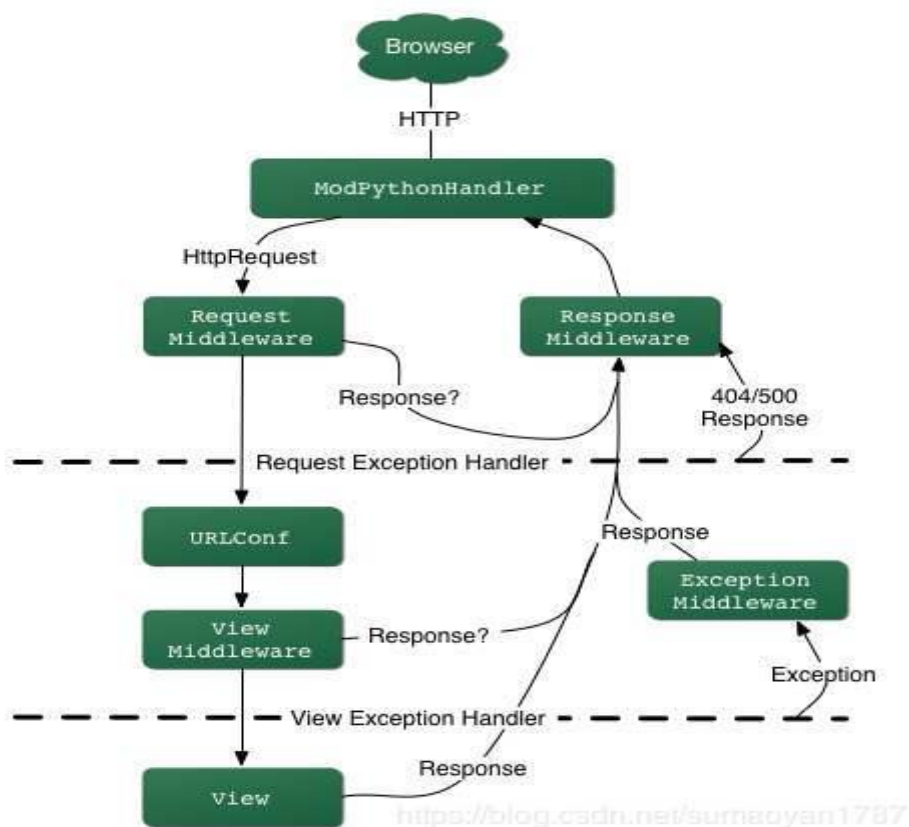


图 2-2 Django 的工作机制

首先通过 `manage.py runserver` 来启动 Django 服务器，这时就载入了在同一目录下的

settings.py，其中涵盖了与项目相关的各类配置信息，而 ROOT_URLCONF 就是最为关键的一个配置，其作用是，可使 Django 知道具体哪一个 Python 模块可被用来作为相应的 URL Conf，其中 urls.py 为默认的，在进行 url 访问时，将由 Django 基于实际的设置来进行 URL Conf 的装载，接着再根据具体的先后顺序来进行 URL patterns 的匹配。若能够找到，将进行视图函数的调用，同时其首个参数(一般为 request)为相应的 Http Request 对象，由 view 函数来进行 Http Response 相关对象的返回。

2.1.2 Django 架构与 Spring 架构的比较

Django 具有以下一些优点：所提供的可视化控制台可以方便用户操作；URL 的配置可以变得方便而且优雅；其 ORM 映射机制非常完善，甚至无需去写 SQL 语句；自带的安全验证系统可以方便进行用户管理和权限控制；提供多种网站防御措施，可以防止 SQL 注入 (SQL injection)、跨站请求伪造 (CSRF)、跨脚本攻击 (XSS) 和点击劫持 (ClickJacking) 等。其缺点是系统耦合度高，无法像 Spring 那样引入第三方库，且 ORM 没有 Mybatis 等持久层框架强大。

Spring 的优点也非常多：其 AOP 编程的支持可以方便进行非业务逻辑的控制；IOC 控制反转将对象间的依赖关系完全交给 Spring 负责，方便解耦、简化开发；可以集成诸如 Mybatis、Hibernate、Shiro、Dubbo 等优秀的框架；并支持声明式事务。其缺点就是框架太过笨重，应用时需要导入大量的 jar 包；而且配置 XML 文件时非常繁琐。

由于在数据可视化技术方面，Python 语言有着得天独厚的优势，因此我们在应用框架的选择上更加倾向于 Django+Python，而放弃了 Spring+Java 的方式。

2.2 Python

Python 是一门解释型高级动态编程语言，它具有跨平台、开源、免费的特点。Python 是面向对象的程序设计语言，它支持命令式编程、函数式编程等，拥有大量的成熟扩展库，这些扩展库可以支持所有领域的应用开发。还有人将 Python 比作是一种“胶水语言” (glue language)，因为它可以融合许多种不同语言编写的程序，实现了无缝拼接，这样可以充分发挥各类语言和工具的优势，满足各类不同应用场景下的需求。它源自 ABC 语言，Modula-3 对其带来了比较大的影响，它也融合了 C 及 UnixShell 的一些主要的使用习惯特点。发展至今，Python 的功能包括了类、函数、异常处理等，数据类型则包含表和字典等，同时通过模块来拓展系统，它将许多及其层面上的细节隐藏，交给编译器处理，并凸显出逻辑层面的编程思想，程序员可以花更多的时间用于思考程序的逻辑，

而不是具体的实现细节，这一特征吸引了广大的程序员。

2.3 Pandas 库

作为一个 BSD 许可且为开源性的库，Pandas 可为 Python 提供易用且高性能的数据分析工具以及有关的结构。实际上，Pandas 为一个以 NumPy 作为其重要基础的 Python 数据分析包，数据分析是它最为关键的应用目的，此外，还可进行数据的清洗。它可提供一系列比较高级的处理数据及针对数据结构的重要方法。其中的数据结构主要有 Data Frame 及 Series。Series 有点像一个一维数组，其组成包括了一组数据和一组与之相关的数据标签，简单的 Series 对象也可以只由一组数据产生。DataFrame 则好像一个表格，它由一组有序的列组成，每列的值的数据类型可以不同，可以使数值或者字符串亦或者布尔类型等。DataFrame 在行和列上均有索引，它实际上可以被看作是由 Series 对象组成的字典类型。

2.4 Matplotlib 库

Matplotlib 库是 Python 扩展的一种绘图库，通常与 NumPy 一同使用，它是开源的，可作为 MATLAB 的替代方案，它还可以和 PyQt、 wxPython 等图形工具包一起使用。Matplotlib 库作为 Python 中一个 2D 图形类库，它对于实现数据处理与展示是非常优秀的，提供了超过 100 多种的图像处理 and 实现方法，能够根据用户所需生成不同的图形，比如柱形图、散点图、直方图和三维图等等。Matplotlib 生成的图形可以通过鼠标点击操作实现界面交互，同时该 2D 图形库可以同时 Python 脚本和 Jupyter Notebook 中使用，实现可跨平台操作。另外 Matplotlib 图形库可以方便被其他平台使用，而且生成图形质量非常高，足以应对日常各类应用。

2.5 Echarts

作为一个可视化的开源库，它主要是通过 Java Script 来得以最终实现，是由百度公司创建的，可以在 PC 或移动设备上流畅运行，兼容性极高，当前绝大部分浏览器都可以使用。其底层主要是基于 ZRender（一类矢量图形库），它所提供的数据可视化图表具有良好的可个性化定制性、交互性以及直观性等。它还可实现各类 K 线图、饼图、散点图、柱状图以及折线图等的提供，此外，还包括盒形图，漏斗图、旭日图，关系图、线图以及地图等，而且对于不同的图相互间混搭也予以支持。本文中主要应用的是

Pyecharts 库，它是 Python 中一个成熟的扩展库，主要作用是用于生成 Echarts 图表的类库，从本质上来讲是 Echarts 与 Python 的对接，使用 Pyecharts 既可以独立生成网页，也可以在 Flask、Django 等应用架构中集成使用。

第三章 系统需求分析

3.1 系统总体需求

目前,农产品质量安全信用管理系统只是对收集或采集的信息进行简单的统计分析,数据可视化水平不高。此外,在不同的子系统之中,也有着并不一致的数据结构特征,此外,在存储数据方面也有较高的分散性,所以难以关联分析数据。本论文旨在对农产品质量安全信用管理系统中的各类数据进行加工和处理,依据用户需要实现多种多样的数据可视化,针对应用要求、数据特点等因素的不同,提供多样化的数据展示方式,让用户能够更便捷地发现农产品质量安全信用管理系统数据中存在的潜在规律和附加价值。图 3-1 详细地展示了系统需求示意图,其中关于不同部分的深入分析将在这一章里展开。

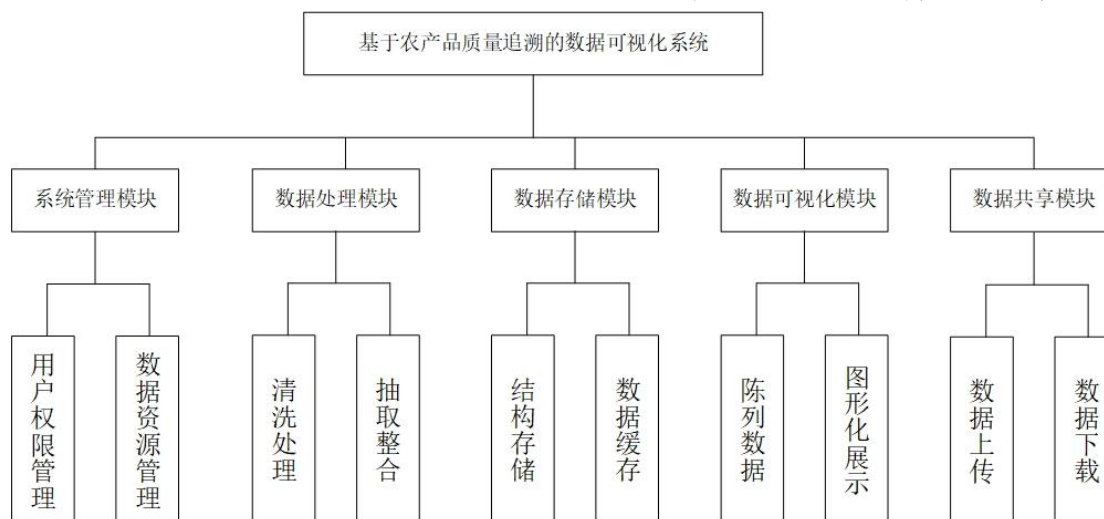


图 3-1 系统需求设计图

3.2 数据预处理需求

本论文中数据来源为农产品质量安全信用管理系统,数据由该平台抽取,抽取后会进行数据的处理,在处理数据的过程中需要格外注意数据质量,我们通常通过数据清洗来解决数据质量问题。对于农产品质量安全信用管理系统的原始数据,由于系统应用的方向不同,这些数据中往往会有一些冗余数据,甚至有些数据因为采集录入时的疏忽,本身就是一些无效数据,所以必须对这类数据先行过滤,再存储到数据库中。对于来自农产品质量安全信用管理系统不同子系统的数据而言,由于设计标准的不同,它们可能存在数据类型、数据含义、命名方法、数据单元甚至数据字符集等属性不同的情形,针对这种高度异构性的特点,我们在处理此类数据时,为了能够使其融合在具有相同数据类型的集合中,就需要通过数据清洗、精简和转换等高效数据处理方式,将其整合成

为属性同一的数据。

3.3 数据存储需求

在数据库选型方面，由于农产品质量安全信用管理系统中的大部分数据在使用过程中存在着事务一致性、使用方便、易于维护等要求，因此最为合适的选择还是关系型数据库。基于农产品质量安全信用管理系统的数据库特点，结合数据可视化的需求，在对数据存储之前要进行 ETL 处理和关系表设计，建议遵循“第三范式”，以降低数据存储开销。但是，在系统实际应用过程中，我们发现，有一些数据会被反复多次地进行查询，而且其数据的变化又不是很大，这类数据如果被存储到关系型数据库中，容易造成系统资源的浪费。我们假设一个动态网站，在每次请求页面时，Web 服务器都会进行各种计算，从“数据库查询——模板呈现——业务逻辑”，这类数据存储的模式定义不佳且更改频繁。再看请求页面的开销处理，这种频繁的请求操作一定会比读取一个现成的标准文件的代价要昂贵的多，所以对于这种类型的数据来说，一般不宜采用关系型的相关数据库，相应地，最佳选择就是 Not-Only-SQL，该数据库可符合一系列的要求（包括高度可用性、高度扩展性、高效率的访问及数据存储，以及高并发读写等方面），此外，对于需使用大量资源来开展重复性的计算操作的结果相关，它可进行有效的保持，由此将低了后续的资源重复消耗现象的出现几率。

3.4 数据可视化需求

其核心功能为：利用直观且合理的图表来促进有关的用户能够高效率地得到所需的数据信息以及主要的细节。采用数据可视化手段，建立起系统与用户之间的交互接口，用户只需要在系统中选择自己所需要的数据集，系统则通事先定义好的运算规则进行运算，将用户所需要的的数据以其所要的图表方式进行展现。

例如：在对某些单一维度下进行统计分析的数据展现时，我们可以对数据进行简单的加工，先按统计维度来进行分类汇总，在获得传统的统计表格后，可以根据表格的字段和值，结合其应用展示的需求，选择其最适合的图形来进行展示，包括柱状图、饼图、折线图、热力图等等；对于某些需要通过地域展示的数据，我们还可以在数据处理获得不同地域数值后，采用地图展示的方式来直观表示，还可以通过不同颜色值，来标记我们所需要的关注的数值高低，让用户可以一目了然；而在对于需要通过多个维度展现的数据来展现的效果时，我们可以对不同的维度采用不同的图形展示，并将其叠加在

同一个图表中，以获取我们需要关注的数据库特点。当然，上述只是我们简单设计的一些应用场景以及实现的思路，对于各类数据的统计处理，我们可以通过调用功能时实时去数据库获取数据并通过运算来获取最新的数据进行展现。

3.5 其他需求

为了适应数据的多源性，在数据共享模块还提供了共享数据的上传与下载的操作，用于非农产品质量安全信用管理系统的数据进行数据可视化操作，通过将不同格式的数据集合（如 Word、Excel、CSV、PDF 等）上传至系统后，经过处理、分析和可视化转换，同样能够将数据进行可视化呈现，同时对所有经过处理后的数据提供下载功能，提高资源的利用效率。

第四章 系统设计

由需求分析可以得知，基于农产品质量安全信用管理系统的可视化系统实现的主要目标是让用户可以更有效和直观地查看纳入农产品质量追溯的各类数据，发现其内在的规律，挖掘其更多的实用价值。本章介绍了系统的概要设计和架构设计，同时对系统的功能模块进行了具体的设计。

4.1 系统概要设计

在该系统中，所包含的功能模块如下：数据的共享、可视化、存储及处理，此外还包括系统管理。其中，系统管理的功能是有用地管理相关的数据资源以及日志和用户的权限等方面；而数据处理的功能则是针对该平台所得到的各类原始性数据展开集成整合、转换以及清洗等操作；数据存储的目标为：在数据库里实现非结构化数据的临时缓存以及针对结构化数据所进行的永久存储等；数据可视化实现将清洗和整合后的数据按照分析要求，通过数据陈列或者图表形式在前端界面中进行展现；数据共享对实现可视化的数据提供了相应的下载服务，同时对于非农产品质量安全信用管理系统的数据库提供上传服务，上传后可以通过清洗、整合、分析等操作，再采用数据可视化技术将其呈现出来。

4.2 系统架构设计

本系统采用的是 Django 的 WEB 开发架构，前面介绍过 Django 架构采用的是 MVT 模式，针对于本系统的架构设计，本文从模型（Model）、视图（View）和模板（Template）三个层面进行阐述。

其中的模型层也就是所谓的数据存取层，主要是处理数据方面的各项事务，包括有效性的验证和数据的存取，此外还涉及到数据关系及行为等方面，此外，还有各类数据相互间所存在的内在关联性等方面。在 Django 框架中，采用了 ORM 关系映射来映射数据到有关的对象上，当前端用户发出相关的请求时，利用业务逻辑处理方法来获得，同时，通过关系对象模型来有效地转化关系型数据（来自于数据库）以及对象型数据（来自于前台），由此，就无需对数据库实施直接性的操作。对于当前的系统而言，其模型层主要是用来访问相关的数据库（文本文档数据映射的有关对象以及系统数据等），由此使业务逻辑获得所需的数据服务。

视图层不同于传统意义上“MVC”模式中的 V，主要是处理和表现相关的各类决定，比如用户需要对哪些数据进行展现，以及通过渲染模型层返回的处理结果，生成响应页

面。

对于模板层而言，其功能主要是进行模型的有效存取，对模板逻辑进行调取，根据实际的需要，通过 html 文件来显示数据。当前的系统是以 B/S 架构系统作为基础，而系统的用户则是利用浏览器来和可视化平台实现交互的目标。

4.3 功能模块设计

4.3.1 系统管理模块

系统管理模块主要通过后台管理功能，对系统用户权限、数据库数据及文件以及系统运行日志进行管理，从而实现对系统、数据和图表的管理，具体功能图见图 4-1 所示。

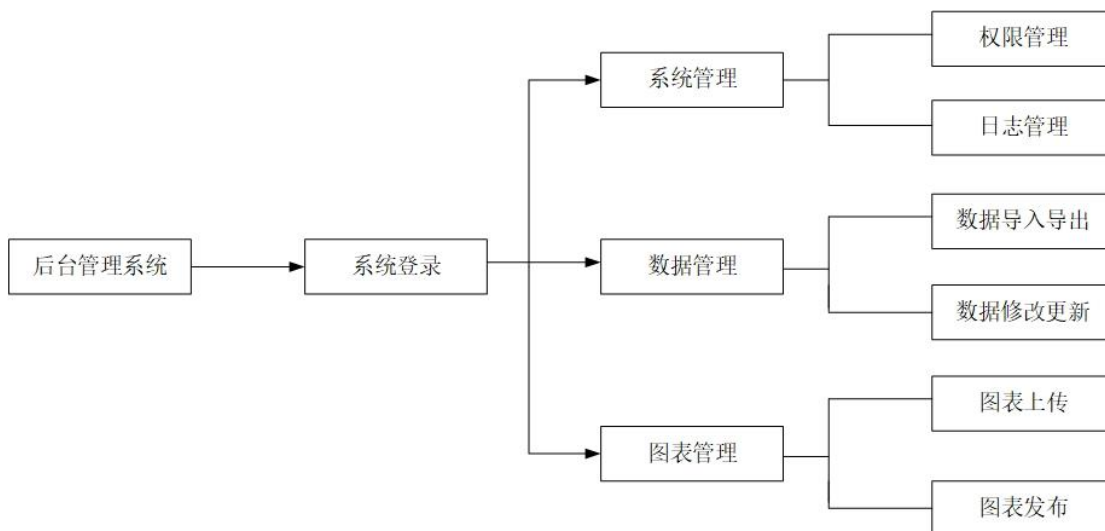


图 4-1 后台管理功能图

很明显，其后台管理主要包括了图表资源管理、数据资源管理以及系统设置等三个方面。对系统设置而言，其包含了运行日志查询、用户权限及用户管理等；对数据资源管理而言，其包含了对系统运行产生的数据进行管理，主要是对前端显示数据的增删改操作以及对后台抽取的数据资源的增删改操作，另外还提供了批量导入功能，用户可以批量导入规定格式的数据文件到后台系统，以作为对数据来源的补充，后台系统将数据转换并存入数据库，并实现存在相互关联的不同的数据资源可以通过后台管理系统建立关联关系，经过相应的可视化处理后一起展现；对图表资源管理而言，系统可以对图表进行上传操作或者发布、更新生成后的可视化分析图（表）。

4.3.2 数据处理功能

该模块主要功能如下：针对平台数据以及各类由外部所导入进来的相关数据开展处

理，其中所涉及的数据又包括了非结构化以及结构化等数据。结构化数据可以轻易地将其根据不同的加工要求写入数据库相应的表中，但非结构化数据，如文件类数据，包括 Word 文件、Excel 文件、PDF 文件及一些图片格式文件，其无法继续再被分割，在数据处理过程中，则是将这些文件一定的规律存储在对应的文件目录中，而数据库表中仅仅是存储了这些文件存储的文件目录路径，以供后续方便查找。部分文件类数据中也存在可抽取的数据信息，我们需要对这一类的文件数据进行分析，通过挖掘或图文识别等手段，将其转换为结构化数据，经过处理后存入数据库中。数据处理流程图见图 4-2。

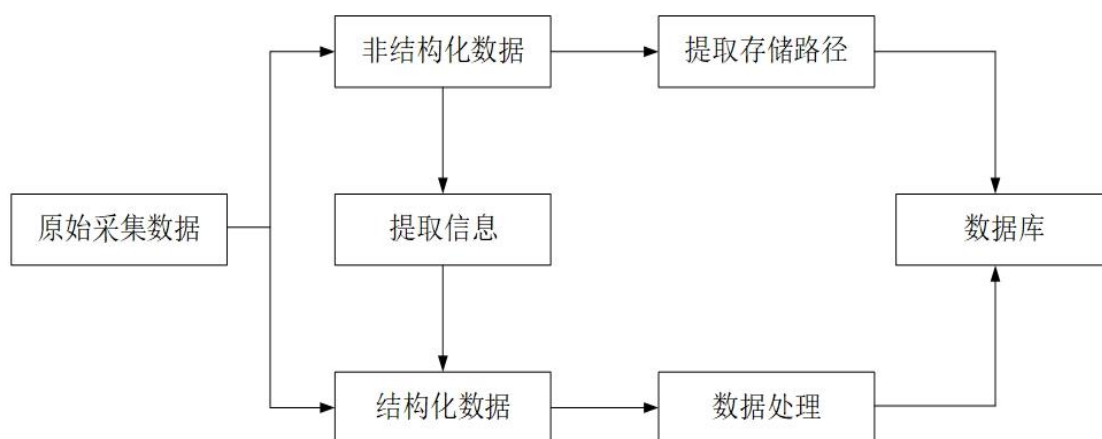


图 4-2 数据处理流程图

不同数据的处理方式是不同的，具体来说：

1. 结构化数据处理

它又具体涵盖了在数据集的内部转换其类型以及进行空值处理，还包括数据的去重处理，此外还有在不同的数据集之间进行集成和整合等处理。例如一些数值型数据需要按需转换成整型或浮点型等数据，以利于数据存储、数据运算和后期图表的生成；而一些字段信息不完整的数据，需要对空缺字段进行处理，通常是在写入到数据库之前，通过 SQL 语句将空缺字段设置为 Null 或者默认值，以确保数据符合存储规范；对于有些数据，其间可能存在某种关联关系，则可以通过字段关联，获取某些缺失的字段，来进行添加，也可以整合成新的数据集。

2. Excel 文件数据处理

我们在对 Excel 表进行存储是，其实质是通过二维表来进行数据的存储，然后通过数据抽取的方式来获取信息，抽取本质上就是将这个二维表格转换成一维数据记录。实际操作中，假设一个 Excel 文件中每条数据分别记录着地区、年份、业务项和业务值，由于采集或录入的原因，有的数据中部分地区或年份中没有相关的业务项和业务值，那

么数据要先进行空值判断，删除缺失值的记录，保留字段信息完整的数据，将其存储在数据库中。

3. 其他文件类数据处理

对于其他的一些文件，我们则要依赖于 Python 强大的第三方库来实现数据的抽取。如视频文件，可以导入 MoviePy 库，先将音频从视频文件中抽取出来，再调用讯飞开放平台的语音转写功能，将语音转为文本后按照相应的方式进行存储；而 PDF 文件，可以导入 pdfminer3k 库，将 PDF 文件中的内容提取成文本后按照相应的方式进行存储；还有 Word 文档文件，可以导入 python-docx 库，提取 Word 文档中的文本内容后按照相应的方式进行存储；另外 PPT 幻灯片文件，可以导入 python-pptx 库，对 PPT 中的内容提取成文本内容后按相应的方式进行存储。

4.3.3 数据存储模块设计

数据存储模块主要用于存储原始数据经过清洗之后形成的合规数据、数据分析运算过程中产生的临时性数据以及分析处理后形成的结果型数据，它们是系统的底层数据基础。在需求分析时已经分析过，本系统对于结构型统一的规整数据，采用 MySQL 关系型数据库存储，对于系统临时数据信息存储在 Redis 缓存数据库中，利用 key-value 存储系统来对关系型数据库起到一个很好的补充作用。

对于 WEB 应用来说，最佳 RDBMS 应用软件为 MySQL，通过有效的数据库设计，它并不是将所有数据都放在一个大仓库里，而是根据各种业务需要，在各类表进行保存，由此使得访问速度得以提升，同时具有更高的灵活性，可通过 MySQL 来进行大型数据的处理，所用的是标准型的 SQL 语言，因此可在若干各系统中进行运行，此外，对于各种编程语言均予以支持。作为一类开源的 Key-Value 数据库，Redis 是基于内存的，可对网络予以支持，遵循 BSD 协议，可通过 ANSI C 语言来进行编写，可进行一系列语言 API 的支持。一般情况下，它也被称作数据结构服务器。Redis 是现在最受欢迎的 No-only-SQL 数据库之一，它基于内存运行，性能高效，可以进行持久化、高并发的读写，对于本系统中后台视图生成计算量的减少、请求度高的分析图表的缓存以及“降开销、提速度”方面都有很大的帮助。

4.3.4 可视化模块设计

可视化模块是本系统的核心模块，它通过可视化的技术手段将上述处理完成的数据

以数据陈列或者图表的方式在 web 前端展示。本系统主要的可视化类库是基于 Python 的 Pyecharts，其本质是百度的 Echarts，它提供了饼图、柱状图、K 线图、折线图以及散点图等一系列的图，也提供了如盒形图一类的统计类图形，还提供了地理数据可视化的地图和热力图，也有平行坐标或极坐标（多维数据可视化）、仪表盘和漏斗图（二者被用在 BI 上）、treemap 和关系图（二者为关系数据可视化），此外，对于混搭不同的图也提供支持。

系统对于整体宏观方面的数据可视化，采用了地图展现的形式，针对于江苏省不同地市之间纳入生产经营管理的主体，通过直观的地图模式进行了展现；对于一些农产品的生长性指标数据，采用了折线图、柱状图等方式展现，也尝试了使用二者合二为一的方式；对于监管性的指标类数据，我们可以通过仪表盘的方式来直观展现。具体实现方式参见第五章内容。

第五章 系统功能实现

本系统在 Windows10 的操作系统下，采用 Django 的 Web 架构，使用 MySQL、Redis 数据库，运用 Python 编程语言，依托其内置类库，完成了相应的开发工作。本章将以部分实例的方式进行介绍。

5.1 地图的展现

为展现全省各地区 2020 年追溯企业的分布，我们从农产品质量安全信用管理系统获取全省各市 2020 年 1-8 月份登记的追溯企业数据，并采用地图模式展现，在展现的地图中，通过地区颜色从蓝到红，展示数据的高低值，直观反映各地区开展农产品质量追溯的情况。

部分实现代码如图 5-1 所示：

```
from pyecharts import Map
map2 = Map("江苏地图", '江苏', width=1200, height=600)
city = ['南京市', '无锡市', '徐州市', '常州市', '苏州市', '南通市', '连云港市', '淮安市', '盐城市', '扬州市', '镇江市', '泰州市']
values2 = [583, 192, 629, 2511, 166, 697, 204, 310, 1010, 556, 106, 1054, 1570]
map2.add('全省2020年1-8月追溯企业分布', city, values2, visual_range=[100, 3000], maptype='江苏', is_visualmap=True,
map2.render(path="江苏地图.html")
```

图 5-1 地图展现的实现代码

效果如图 5-2 所示：

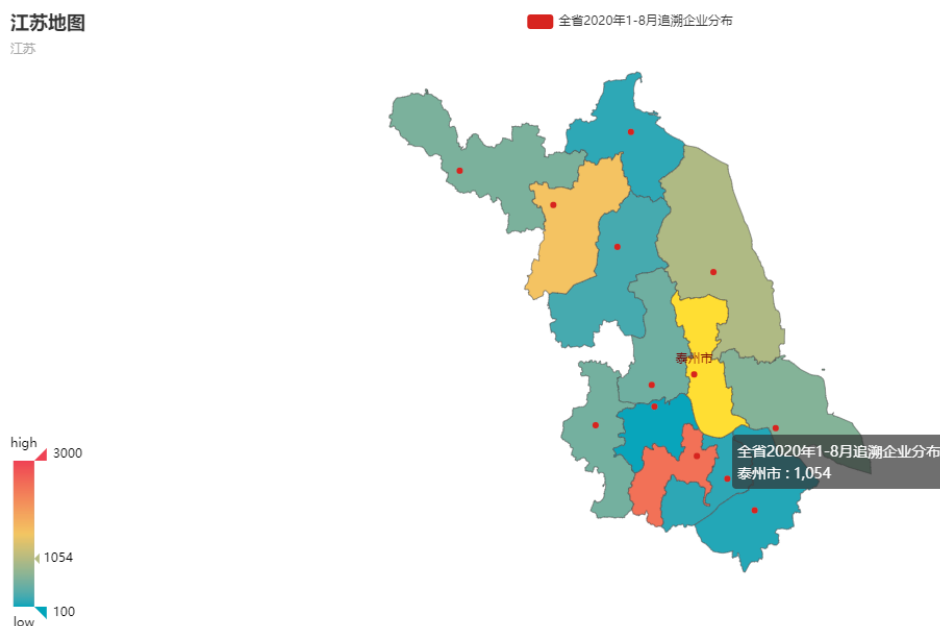


图 5-2 地图展现效果图

5.2 柱状图的展现

针对于 2018 年和 2019 年分月份江苏省农产品安全质量追溯企业的新增量，我们可以通过柱状图的形式将其展现出来。

部分实现代码如图 5-3 所示：

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams["font.sans-serif"]=["SimHei"]
#建立一个坐标系
plt.subplot(1,1,1)
#指明x和y1,y2值
x=np.array([1,2,3,4,5,6,7,8,9,10,11,12])
y1=np.array([64,0,0,0,0,0,0,34,85,690,824,201])
y2=np.array([155,50,172,590,75,286,247,159,270,829,1362,1998])
#绘图
plt.bar(x,y1,width=0.2,label="2018年追溯企业值")
plt.bar(x+0.2,y2,width=0.2,label="2019年追溯企业值")
#设置标题及x、y轴名称
plt.title("全省2018年、2019年追溯情况对比图")
plt.xlabel("月份")
plt.ylabel("追溯企业值")
plt.legend()#设置图例
#保存图表到本地
plt.savefig("c:/bar.jpg")
```

图 5-2 柱状图实现代码

效果如图 5-4 所示：

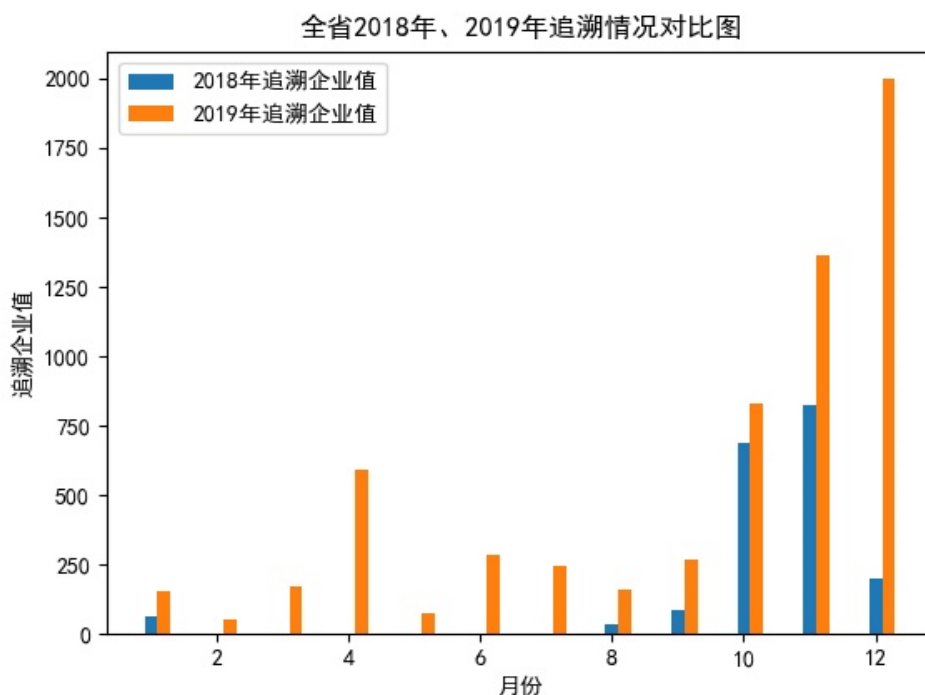


图 5-4 柱状图展现效果图

5.3 堆叠图的展现

针对于上述柱状图，还可以用堆叠图的形式展现，部分实现代码如图 5-5 所示：

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams["font.sans-serif"]=["SimHei"]
#建立一个坐标系
plt.subplot(1,1,1)
#指明x, y1, y2的值
x=np.array([1,2,3,4,5,6,7,8,9,10,11,12])
y1=np.array([64,0,0,0,0,0,0,0,34,85,690,824,201])
y2=np.array([155,50,172,590,75,286,247,159,270,829,1362,1998])
#绘图
plt.plot([],[],color="b",label="2018年追溯企业值",linewidth=5)
plt.plot([],[],color="y",label="2019年追溯企业值",linewidth=5)
plt.stackplot(x,y1,y2,colors=["b","y"])
#设置标题及x、y轴名称
plt.title("全省2018年、2019年追溯情况对比图")
plt.xlabel("月份")
plt.ylabel("追溯企业值")
plt.legend(loc="upper center")
#保存
plt.savefig("c:/stackplot.jpg")
```

图 5-5 堆叠图实现代码

效果如图 5-6 所示：

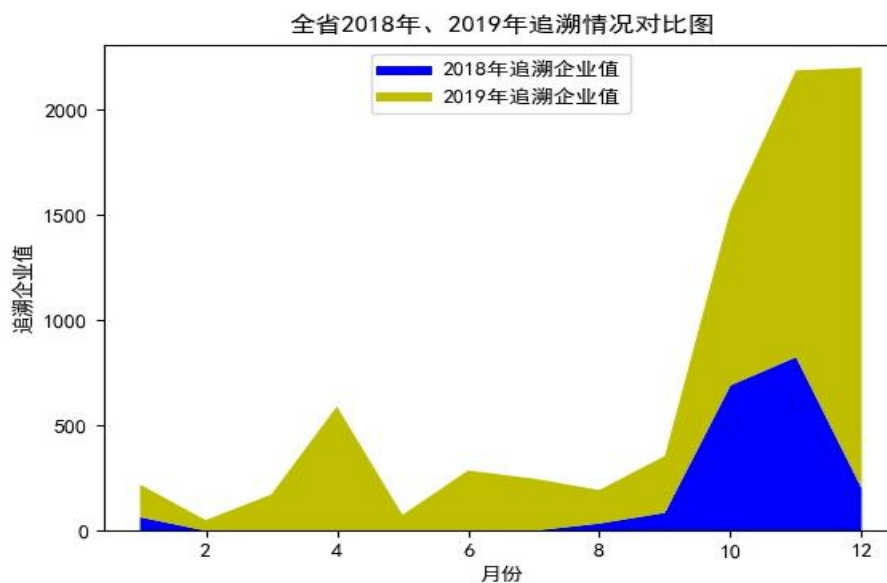


图 5-6 堆叠图展示效果图

5.4 柱状图和折线图二合一的展现

农产品的生长，我们关注其浇灌度的影响，降雨和蒸发量是决定其土壤水分的重要因素，我们对于降雨量和蒸发量采用柱状与折线图二合一的方式展现，部分实现代码如

图 5-7 所示：

```

from pyecharts import Bar
#设置行名
columns = ["1月", "2月", "3月", "4月", "5月", "6月", "7月", "8月", "9月", "10月", "11月", "12月"]
#设置数据
data1 = [2.0, 4.9, 7.0, 23.2, 25.6, 76.7, 135.6, 162.2, 32.6, 20.0, 6.4, 3.3]
data2 = [2.6, 5.9, 9.0, 26.4, 28.7, 70.7, 175.6, 182.2, 48.7, 18.8, 6.0, 2.3]
#设置柱状图的主标题与副标题
bar = Bar("柱状图", "一年的降水量与蒸发量")
#添加柱状图的数据及配置项
bar.add("降水量", columns, data1, mark_line=["average"], mark_point=["max", "min"])
bar.add("蒸发量", columns, data2, mark_line=["average"], mark_point=["max", "min"])
#生成本地文件（默认为.html文件）
bar.render()

from pyecharts import Line
line = Line("折线图", "一年的降水量与蒸发量")
#is_label_show是设置上方数据是否显示
line.add("降水量", columns, data1, is_label_show=True)
line.add("蒸发量", columns, data2, is_label_show=True)
line.render()

```

图 5-7 柱状-折线二合一代码实现

效果如图 5-8 所示：

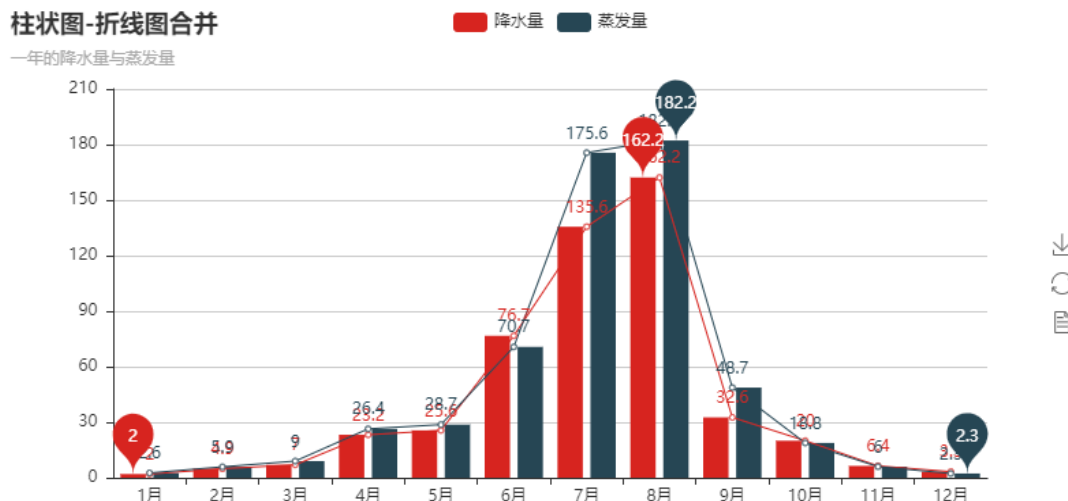


图 5-8 柱状-折线二合一展现效果图

5.5 仪表盘的展现

对于纳入监管的主体，其抽检的结果也可以通过仪表盘的方式展现，可以让监管部门和监管人员一目了然看到该企业的农产品抽检情况，从而进一步加强监管的力度、提高监管的手段。部分实现代码如图 5-9 所示：

```
# coding=utf-8
from pyecharts import Gauge
import random
sjs = 10000
hgs = 9994
gauge = Gauge("仪表盘显示")
gauge.add("抽检指标", "抽检合格率", hgs/sjs*100)
gauge.render('gauge.html')
```

图 5-9 仪表盘代码实现

效果如图 5-10 所示：



图 5-10 仪表盘效果图

以上是本系统中部分数据可视化功能的实现代码和效果图展示，Echarts 提供了强大的数据可视化技术，可以支持多种形式的图表生成，在此不再进行赘述。

第六章 总结与展望

6.1 总结

随着农产品质量追溯工作的不断推进，质量监管工作积累了大量的数据，如何有效地整合和利用此类异构数据，发现其内在规律，挖掘其潜在价值，成为了今后一段时间农产品质量监管的重要方向。在当前的工作中，针对农产品质量安全信用管理系统相关数据可视化系统展开了一系列的设计工作，并实现了该可视化系统，同时探索了其主要的技术。当前的工作设计了一系列的功能模块，包括数据的共享、存储和处理，以及实现了可视化的功能以及系统管理等，着重地阐述了实现数据可视化的主要途径。针对目前的农产品质量监管的现状，给出了合理的解决方案。本系统不仅提供了数据可视化功能，同时也在平台实现了数据的共享功能。具体所做的工作如下：

1. 开展了相关数据资源的 ETL 处理的研究工作。结合用户的实际需求，充分地考虑各类数据集的情况（其主要的来源）以及数据可视化方面的具体需求，有效地清洗及处理了相关的原始数据，同时进行了数据转换以及整合，对处理非结构化以及结构化数据的主要方法进行了深入的阐述。

2. 可视化系统设计与实现。基于 Django 的应用框架，设计并实现江苏省农产品质量溯源数据可视化系统，建立不同的可视化模型，将 ETL 后的数据资源映射其中，对内进行可视化图表的展示或者陈列展示，对外提供数据共享。

6.2 展望

虽然本次的研究与设计基本实现课题最初的设想，但是受制于个人能力水平有限以及课题研究时间较短等原因，本系统还有许多不尽如人意的地方，在下一步的研究过程中需要加以改进，具体内容如下：

1. 基于数据的模型建立有待加强。对于平台中的数据，特别是农产品生长环境等第一手的数据资料，在系统中并未有效建立起其生长预测模型，为农业企业或农户提供更多的数据支撑；

2. 实现更多的可视化图表。由于部分数据主要文字类数据，无法开展数值分析，且图形化展示较难，系统中可用于图形化展示的结构化数据较少，导致展示的分析图表数量不丰富。同时由于时间仓促，Echarts 提供的大量可以实现的图表形式未有时间进行进一

步的尝试。

3.系统性能有待提高。当模型层有大量数据需要在视图层进行加载渲染时，系统处理的效率较慢，需要在下一步研究过程中对系统进行优化，提高系统运行效率。

参考文献

- [1] 周志化, 任玉玲, 陆树芬. Python 编程基础 (双色版) [M]. 上海: 上海交通大学出版社, 2019
- [2] 毛俊, 彭宏, 孟利民. 基于 Django 框架的车载定位系统 PDF 报表下载软件的设计与实现[J]. 浙江工业大学学报, 2012.
- [3] 许文鹏. 数据可视化系统架构的设计与实现[D]. 北京: 北京交通大学, 2015.
- [4] 冯毅. 微博数据的可视分析[D]. 杭州: 杭州电子科技大学, 2016.
- [5] 杜丽娟. 关系型数据库与 NoSQL 数据库的性能对比[J]. 智能计算机与应用, 2017.
- [6] 陈为. 大数据可视化与可视分析[J]. 金融电子化, 2015.
- [7] 张若愚. Python 科学计算: 第 2 版 [M]. 北京: 清华大学出版社, 2016.
- [8] 贾利娟, 刘娟, 等. 基于 PyEcharts 的全球玉米贸易数据可视化系统建设及应用展望[J]. 农业展望, 2019.
- [9] 刘大成. Python 数据可视化之 matplotlib 实践[M]. 北京: 电子工业出版社, 2018.
- [10] Igor Milovanovic. Python 数据可视化编程实战[M]. 北京: 人民邮电出版社, 2015.
- [11] 向怡宁. 就这么简单: Web 开发中的可用性和用户体验[M]. 北京: 清华大学出版社, 2008.
- [12] 薛耀伟. 基于 Django 框架管理界面自动生成模块的设计与实现[D]. 哈尔滨: 哈尔滨工业大学, 2014.

致 谢

时光如白驹过隙，我的大学生涯即将结束，在即将完成本文的撰写工作之前，我要向三年校园生活中所有帮助、关心、和支持过我的老师、同学表达深深的谢意。

首先，我要感谢我的两位论文指导老师——谢晓伟老师和陈君艳老师，在系统设计与论文撰写的过程中，是她们的指导与督促、帮助与批评才让我能够坚持下来。本系统的设计，是对我大学所学知识的一次全面检验，也是让我将所学知识融会贯通的一次机会，让我学到了许多课堂上没有的新知识；另外，两位老师对于学术研究的严谨精神，也令我受益匪浅。

其次，我要感谢我的同学和朋友，感谢他们在我学校生活中给予我的照顾和帮助，感谢他们论文研究陷入困境时的关心和鼓励，正是由于他们的鼓励，我才得以顺利完成毕业设计和论文编写工作。

最后，我要感谢我的家人们，使他们用无私的爱将我抚养成人，取得今天的成就。

最后，我再次向指导我论文工作的专家和老师们表示衷心的感谢！