|  |
| --- |
| **学士学位论文** |
| **基于WEB的数据可视化分析交互工具开发**  **--以石油和天然气为例** |
| |  |  | | --- | --- | | 学 号： | **20141002675** | | 姓　 名： | **张宣祺** | | 学 科 专 业： | **信息工程** | | 指 导 教 师： | **徐战亚 讲师** | | 培 养 单 位： | **信息工程学院** | |
| 二○一八年六月 |

中国地质大学（武汉）学士学位论文原创性声明

本人郑重声明：本人所呈交的学士学位论文《基于WEB的数据可视化分析交互工具开发--以石油和天然气为例》，是本人在指导老师的指导下，在中国地质大学（武汉）攻读学士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果，对论文的完成提供过帮助的有关人员已在文中说明并致以谢意。

本人所呈交的学士学位论文没有违反学术道德和学术规范，没有侵权行为，并愿意承担由此而产生的法律责任和法律后果。

学位论文作者签名：

日 期： 年 月 日

**摘 要**

移动互联网和智能终端技术的飞速发展，LBS正深入生活。据统计新浪微博的5.56亿用户产生了大约6亿条签到记录，其中旅游景点约占20%。Geo-tagged data 的爆发式增长为城市与地理学研究提供了一个前所未有的研究的新数据空间。然而不同类型的群体在社交媒体上的行为也有着很大的不同，本文基于在武汉的约43万用户，在2012年1月1日到2017年12月31日的1200余万次的微博签到数据，结合随机森林提出了一种有效区分游客与市民签到行为的方法。并通过分类结果分析比较武汉典型景区的游客与市民的行为区别，结合k-means聚类算法在时间上进行聚类，以及利用协同过滤和社区发现的算法在空间上进行聚类，寻找不同景区之间的相关性和差异。结合LDA主题分布对微博文本进行聚类处理，找出不同景区的“意象”与特征。通过社交媒体的签到数据全方位考察武汉这座城市的“城市意象”。

**关键词**：城市意象；景区推荐；LDA；社交媒体；大数据

**ABSTRACT**

**Keywords：**

**目 录**

[第一章 绪论 1](#_Toc10017)

[§1.1 研究背景与意义 1](#_Toc27110)

[§1.2 发展现状 1](#_Toc14012)

[§1.3 主要研究工作 2](#_Toc11769)

[§1.4 主要目标 2](#_Toc6005)

[§1.5 论文结构 3](#_Toc29722)

[第二章 本课题涉及的关键技术 4](#_Toc16078)

[§2.1 开发语言Python&JavaScript 4](#_Toc30973)

[§2.2 Flask框架 4](#_Toc1789)

[§2.3 MySQL数据库 6](#_Toc17673)

[§2.4 Ajax 简介 6](#_Toc14766)

[§2.5 可视化工具Highchart&Highmaps 7](#_Toc23189)

[§2.6本章小结 7](#_Toc6609)

[第三章 可行性与需求分析 8](#_Toc23288)

[§3.1 可行性分析 8](#_Toc11487)

[3.1.1 技术可行性 8](#_Toc323)

[3.1.2 资源可行性 8](#_Toc18234)

[第四章 系统设计与实现 13](#_Toc25491)

[§4.1 系统设计目标 13](#_Toc14353)

[§4.2系统设计思想 13](#_Toc26216)

[§4.3 系统总体设计 14](#_Toc8803)

[§ 4.4 数据库设计与实现 15](#_Toc31264)

[4.4.1 概念结构设计 15](#_Toc28109)

[4.4.2 数据库实现 17](#_Toc2123)

[§4.5 系统模块设计 19](#_Toc26036)

[§4.6 系统模块实现 24](#_Toc18485)

[§4.7本章小结 30](#_Toc1400)

[第5章 总结与展望 31](#_Toc28159)

[§5.1总结 31](#_Toc15673)

[§5.2展望 31](#_Toc4557)

[致 谢 32](#_Toc30523)

[参考文献 33](#_Toc24481)

# 第一章 绪论

## 研究背景及意义

“大数据时代”（Big Data Age）最早由全球知名的咨询公司麦肯锡提出，麦肯锡称：“数据，已经渗透到当今每一个行业和业务职能领域，成为重要的生产因素。人们对于海量数据的挖掘和运用，预示着新一波生产率增长和消费者盈余浪潮的到来。”互联网、云计算、物联网等五彩缤纷的信息技术在飞速发展，人类世界的各个方面与信息技术正在进行越来越密切地深入融合，前所未有数量的数据被产生并记录下来。对海量数据进行统计、分析，并挖掘出其中蕴含的潜在价值，是我们一直在深入研究的。数据可视化技术将会在数据被合理的采集、解读之后以一种更加容易被人理解，更具有亲切感的方式呈现出来。

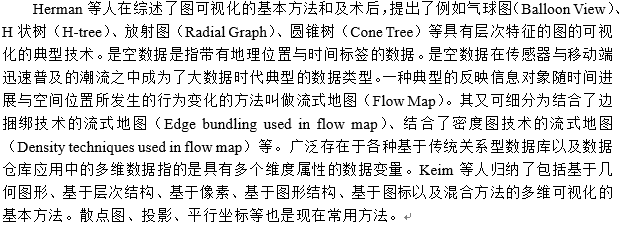
众所周知，随着工业化的发展，日益严重的能源危机已经成为了各国政府，各地民间资本的关注重点。各国政府也都做出了大力发展新能源产业的共同选择，中国的政策也随之倾向于加大对新能源产业扶持力度的方向上。因而可知，新能源将会越来越多地代替传统能源在提供和利用中的地位。2016年由中国产业调研网发布的中国石油天然气市场现状调研与发展趋势分析报告指出，石油在未来的几十年里在一次能源消费中的比重将会下降，但就总体而言，其消费总量依然会有比较大幅度的增加，继续保持第一大能源的地位。作为传统化石能源中绿色能源的天然气获得了世界各国普遍而同一的正视，呈现出了飞速的发展趋势。根据数据分析报告，预计到2035年，相比与2010年，全球天然气消费总量将同比增长50%左右，占到了一次能源消费的比重会超过25%。

以石油和天然气为例进行基于web端的数据可视化分析交互工具开发可以将这两种最具有代表性的一次能源的发展趋势以一种更加直观更加便捷的方式展现出来，便于掌握其发展规律并进行预测，可将其作为投资等方向的一块参考意见。

## 1.2 国内外研究现状

在2008年时，Michael提出了将数据可视化分别为两部分：主题图和统计图形。发展至今，随着数据可视化技术应用范围和影响的逐渐扩大，各种可视化方向也处于方兴未艾时期。简单可分为以下几种：文本可视化、网络（图）可视化、时空可视化、多维数据可视化等。大数据时代非结构化的数据类型的典型代表之一就是文本信息，同时它也是当今互联网生态系统中最为首要的信息类型，也是各类物联网传感器对信息进行采集之后生成的最重的要信息类型。将包含在文本中的语义特点直观地呈现出来时文本可视化的意义与重心。譬如互联网与社交网络的形式，大数据中最常见的关系就是网络关联关系。

Web是现在触及受众最快、最为广泛的选择。具有统一的网络标准与跨各类操作系统特性的Web端可以确保工具平台的广泛性与普适性。基于web端的数据可视化技术与方法也已发展出许多，如IBM的Many Eyes、分免费和商业版的iCharts、一个致力于与谷歌竞争的计算知识引擎Wolfram Alpha、可以上传自己数据集的托管工具Visualize Free以及Stanford University可视化团队开创的Data Wrangler还有基于Javascript的D3.js框架等。在web端数据可视化框架与工具百花齐放的环境中，选择基于web开发此分析交互工具是一件符合技术潮流且能为社会做出意义的事。

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴桯溱뻉지꦳쓌퓊릵쏋ﲲꊷ쟋陸킵쓌풺춹ꛄ벼꾳즵뷓풺쯐쒡ꏔ祴桯滌ꦵ쒸횸醴믖솼ꮵ쑁偉횧돖쿂ꎬ돌탲풱쓜릻쟡쯉뗘쪹폃䎡ꉃ⬫ꆢ䍹瑨潮살뇠킴삩돤쒣뿩ꆣ평폚偹瑨潮뇠틫웷놾짭튲뿉틔놻벯돉떽웤쯼탨튪뷅놾폯퇔뗄돌탲쓚ꎬ偹瑨潮폖

## 1.3 论文组织结构

本文在接下来的章节中，主要做了以下工作：

第2章，主要介绍本项目所用到的各种核心技术及框架、工具的分析与搭建。

第3章，分别对数据的获取、处理、需求以及各种实现进行可行性分析。

第4章，分析整个工具项目的设计与实现。

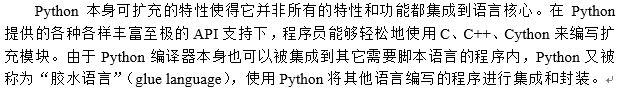
第5章，分析讨论展示与处理结果，做出总结与展望。

# 第二章 本课题涉及的关键技术

## 2.1 开发语言Python & JavaScript

### 2.1.1 Python

Python，由荷兰程序员Guido van Rossum发布于1991年，是一种广泛使用的通用型编程语言。其特点在于融合了多种编程语言的特点，如面向对象等。Python是一种解释性语言，其设计哲学强调代码的可读性和简介的语法（如使用空格缩进划分代码块替代大括号或关键词）。在最初设计Python时，Guido van Rossum的目的是设计出一种强大而优美，提供给非专业程序设计师使用的语言，同时采取开放策略，使Python能够完美结合如C 、 C++和Java等其他语言。

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐

在Web开发方面，Python的一些功能优异的框架如Django、Tornado、Flask、Pyramid使得开发者在开发Web程序遇到复杂的管理功能时节省了大量的时间与精力。国内外许多知名大型公司都在使用Python如Google、Youtube、Quora、知乎、豆瓣等。由于Python对各种网络协议的完善支持，它又经常被用来编写网络爬虫、服务器软件等。常用爬虫框架有Scrapy、Crawle、Newspaper、Python-goose、Portia等。

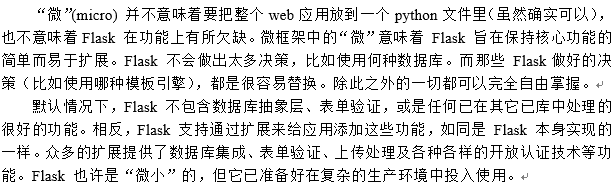
### 2.1.2 JavaScript

JavaScript是由美国程序员Brendan Eich于1995年开发出的一种通过解释执行的高级编程语言。它是一门面向对象（基于原型），动态类型的解释型语言。至今为止，它被世界上的绝大多数网站所使用，也被世界主流浏览器（Chrome、IE、Firefox、Safari、Opera）支持。主要被作为客户端脚本语言的JavaScript相比于服务端的脚本语言，如PHP、ASP，在用户的浏览器上运行，不需要服务器的支持。后来随着Ajax技术、V8引擎和Node.js的发展，及其事件驱动及异步IO等特性，JavaScript逐渐被用来编写服务器端程序。

## 2.2 Flask框架

### 2.2.1 Flask框架简介

Flask是一个用Python编写的基于Werkzeug工具包和Jinja2模板引擎的微型Web框架。Flask使用BSD许可。Flask出自于来自Pocoo Team的Armin Ronacher。以其出色的扩展弹性、简单的核心而出名，被称为“微框架”（microframework）。

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴桯溱뻉지꦳쓌퓊릵쏋ﲲꊷ쟋陸킵쓌풺춹ꛄ벼꾳즵뷓풺쯐쒡ꏔ祴桯滌ꦵ쒸횸醴믖솼ꮵ쑁偉횧돖쿂ꎬ돌탲풱쓜릻쟡쯉뗘쪹폃䎡ꉃ⬫ꆢ䍹瑨潮살뇠킴삩돤쒣뿩ꆣ평폚偹瑨潮

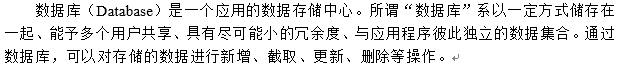
### 2.2.2 技术评估

在Github上star数最多，实际运用中最为广泛的两个Python web框架就是Django / Flask。该如何选用适合本工具的技术栈成为了一个必须要面对的问题。

Django和Flask各有优劣，我的对比如下：

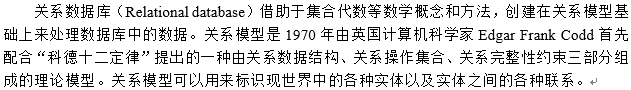
1. Flask
   1. Flask自由、灵活、可拓展性强，在开发时可供选择的第三方库覆盖面更广。
   2. Flask入门更加简便易上手，学习曲线平滑，适合此项目的开发。
   3. Flask确实很“轻”，不像Django框架一般各方面功能都有所涉及，全面而笨重。
   4. Flask更加适合开发小型网站。
   5. 各方面性能均等于或优于Django。
   6. Flask与关系型数据库的配合使用不弱于Django，而其与NoSQL数据库的配合远远优于Django
2. Django
   1. Django除web框架外，自带 ORM 和模板引擎，导致框架稍微有些笨重，自由度和灵活度方面略微有些欠缺。
   2. 使用Django来开发小型应用，会出现许多功能和性能的冗余。
   3. Django更适合于需要快速、稳定、靠谱的企业级网站的开发。
   4. Django成熟、稳定、完善，但相比于Flask，Django的整体生态相对封闭，不太适合后续的继续深入研究。

## 2.3 数据库管理系统



数据库主要可分为关系型数据库和非关系型数据库，根据需求的科学地不同选用正确适合的数据库会极大地降低开发难度，将应用与数据库适当地解耦。

### 2.3.1关系数据库



풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ

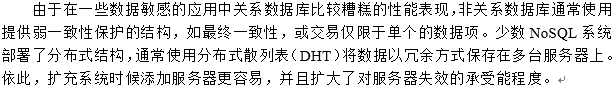
关系型数据库最广为人知的“标志性”技术当属标准数据查询语言—SQL。这是一种基于关系型数据库的语言，可以执行对关系数据库中数据的检索和操作。SQL语句通常操作于表（关系Relation）。表（关系Relation）是以列（属性Attribute）和行（值组Tuple）的形式组织起来的数据的集合。一个数据库包括一个或多个表（关系Relation）。

目前被最广泛使用的关系数据库主要有Oracle、MySQL、SQLite、PostgreSQL等。

### 2.3.2 非关系数据库

非关系数据库（NoSQL）是对不同于传统的关系数据库的数据库管理系统的统称。

NoSQL最早是出现于1998年，由Carlo Strozzi开发的一个轻量、开源、不提供SQL功能的关系数据库，而后逐渐演变并被解释为“非关联性的”， 强调Key-Value Stores和文档数据库的优点的数据库管理系统。

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴

目前被最广泛使用的非关系数据库主要有MongoDB、Apache Cassandra和Redis。

### 2.3.3 技术评估

1. 关系数据库的优缺点
   1. 容易理解。非常贴近逻辑世界的二维表结构配合相对其他模型更容易理解的关系模型使得关系数据库更容易被人接受和理解。
   2. 使用方便。通用的SQL语言使得操作关系型数据库非常方便。
   3. 易于维护。具有丰富的完整性的关系数据库（实体完整性、参照完整性和用户定义完整性）使得数据冗余和数据不一致的概率大大降低，更易于维护。
   4. 高并发都写需求。如往往达到每秒上万次读写请求的网站用户并发行对于传统关系数据库来说，硬盘I/O是一个很大的瓶颈。
   5. 海量数据的高效率读写。对于关系型数据库来说，在一张包含海量数据的表中查询，效率是非常低的。
   6. 高扩展性和可用性。在基于web的结构当中，数据库是最难进行横向扩展的。
2. 非关系数据库的优缺点
   1. 性能。非关系数据库基于键值对，可以想象成表中的主键和值的对应关系，而且不需要经过SQL层的解析，所以性能非常高。
   2. 可扩展性。同样因为非关系数据库基于键值对，数据之间没有耦合性，所以非常容易水平扩展。
   3. 复杂查询。复杂查询是SQL语句的强项，在这个方面非关系数据库的查询效率远低于关系数据库。
   4. 事务支持。对于安全性能很高的数据访问要求是关系数据库的强项。
3. 技术栈选择

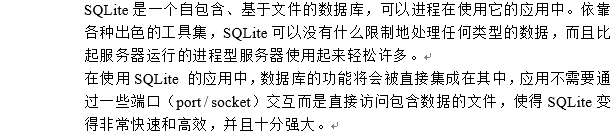
由于本工具所要存储与处理的数据主要为石油与天然气的各种相关数据，数据量并不算海量，没有很大的并发请求与处理。在将数据进行可视化展示时，会涉及到一些交叉查询的请求，表的结构不算复杂但是会有很多数据项，即会有很多列，容易使人对其数据存储结构理解不到位，造成使用时的解读数据库的时间耗费程度以及复用的复杂度非常严重。关系数据库的二维表结构可以帮助我们直观简便地理解其构造，更加便于我们对数据进行存取。

综上所述，决定选取关系数据库作为本工具的数据库管理系统。

### 2.3.4 数据库管理系统的选取

确定以关系数据库作为本工具的数据库管理系统后，在三大开源关系数据库中选取一种作为最终决定：SQLite、MySQL、PostgreSQL。

1. SQLite



풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴桯溱뻉지꦳쓌퓊릵쏋ﲲꊷ쟋陸

SQLite的优缺点：

* + - 1. 基于文件。整个数据库都有很好的迁移性，因为它全部包含在磁盘的一个文件中。
      2. 标准化。SQLite同样支持SQL语句。
      3. 灵活且丰富的功能。SQLite基础功能十分丰富，通过简单地调用一个文件和一个C链接库，就能得到超乎开发所需的灵活且并发的解决方案。
      4. 用户管理。SQLite因为其产生的目的和本身的性质（没有多用户并发的高层设计），并没有支持用户系统。
      5. 灵活性仅局限于使用中。SQLite缺乏额外优化性能的灵活性。SQLite配置、使用都十分容易，并没有使性能更加优化的技巧了。

1. MySQL

MySQL是大型数据库服务器中最流行的。丰富的特性和开源的性质使得它在线上的大量网站和应用程序中大显身手。并且由于其强大而广泛的普及型，大量的第三方应用、工具和集成库对于操作这个RDBCMS的方方面面大有帮助。

与SQLite不同的是，MySQL作为一个独立的数据库系统，应用程序同Mysql守护进程的交互，告诉它去访问数据库自身。

MySQL的优缺点：

* + - 1. 容易使用。MySQL的安装与使用都十分容易，包括可视化的库。
      2. 功能丰富。MySQL 支持大部分关系型数据库应该有的 SQL 功能——有些直接支持，有些间接支持。
      3. 安全性高。MySQL的安全系数非常高。
      4. 快速。MySQL效率较高，速度较快。
      5. 开发停滞。更新缓慢。

1. PostgreSQL

PostgreSQL是一个以实现标准和可扩展性为主要目标的一个先进开源的关系数据库管理系统。对于事务可靠性，例如隔离性和持久性（ACID）、原子性、一致性的完全支持使得PostgreSQL在与其他RDBMS比较时脱颖而出，彰显出了它的独特性。

PostgreSQL拥有高度可编程性，因而可以使用被称作“存储过程”的自定义程序进行扩展. 这些功能可以被创建用来简化一个写重复、复杂并且常常需要数据库操作的任务的执行。

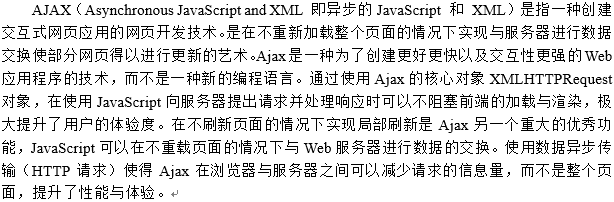
PostgreSQL的优缺点：

* + - 1. 可扩展性。PostgreSQL一如一个高级的关系数据库的标配，可以用预先存储的流程来进行程序性的扩展。
      2. 面向对象。PostgreSQL不仅仅只是一个关系数据库，更出彩的特性是它是一个支持嵌套及一些其他功能的面向对象数据库。
      3. 性能。PostgreSQL在面对简单而繁重的读取操作时可能会出现小题大做的问题而造成性能不如类似于MySQL的其他同行。

(4) 技术评估

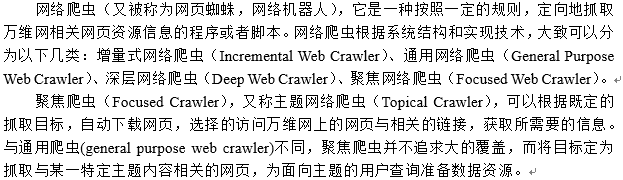
综上所述，在我们需要的是一个具有普适性，功能比较强大，整体稳重而高效的数据库管理系统时，最终选取了MySQL作为本工具的数据存储中心。

## 2.4 Ajax

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴桯溱뻉지꦳쓌퓊릵쏋ﲲꊷ쟋陸킵쓌풺춹ꛄ벼꾳즵뷓풺쯐쒡ꏔ祴桯滌ꦵ쒸횸醴믖솼ꮵ쑁偉횧돖쿂ꎬ돌탲풱쓜릻쟡쯉뗘쪹폃䎡ꉃ⬫ꆢ䍹瑨潮살뇠킴삩돤쒣뿩ꆣ평폚偹瑨潮

## 2.5 爬虫

本工具使用Scrapy框架爬取相关的天然气和石油数据，用于分析和展示。Scrapy是一个为了爬取网站数据，提取结构性数据而编写的应用框架。 可以应用在包括数据挖掘，信息处理或存储历史数据等一系列的程序中。

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴桯溱뻉지꦳쓌퓊릵쏋ﲲꊷ쟋陸킵쓌풺춹ꛄ벼꾳즵뷓풺쯐쒡ꏔ祴桯滌ꦵ쒸횸醴믖솼ꮵ쑁偉횧돖쿂ꎬ돌탲풱쓜릻쟡쯉뗘쪹폃䎡ꉃ⬫ꆢ䍹瑨潮살뇠킴삩돤쒣뿩ꆣ평폚偹瑨潮

## 2.6 可视化工具Highcharts & Highmaps

Highcharts是一个基于纯JavaScript的交互性图表。它兼容性强大，甚至可以向下兼容至IE6，这为我们工具的跨浏览器版本的普适性提供了技术保障与支持。它方便快捷，Highcharts能够简单而有效地为web应用程序添加有交互性的图表，不需要像Flash和Java那样依靠插件才能运行。它功能强大，提供了各种各样对图表配置细致入微的接口API，且图表类型多种多样，可以满足各种需求。适合本工具对各种数据的处理与展示，可以直观有效地将数据更深层的含义呈现出来。

Highmaps是一款基于HTML5的地图组件，继承了Highcharts简单易用的特性，拥有着极其丰富的地图数据集。使用此组件可以简单直观地将石油、天然气在世界，在我国各省的储量、使用量等情况展示出来，便于更加简单地了解目前的情势。

## 2.7 多元线性回归分析模型与GM(1,1)灰色预测模型

### 2.7.1多元线性回归分析模型

天然气消费量预测的研究从20世纪80年代开始，经历了从线性系统预测到非线性系统预测。目前，预测方法主要分为两种：基于时间序列 和基于相关关系的预测。由于我国的社会、经济、政策等随时间变化较大，故本文选择基于相关关系的预测——多元线性回归分析。

多元线性回归预测是研究统计规律的方法之一。在回归分析中，用户把 所关心的指标称为因变量，通常用Y表示；影响因变量的变量称为自变量，用X1、X2.....Xp表示。 回归分析研究的主要问题是:确定Y与X1、X2.....Xp间的定量关系表达式，这种表达式称为回归方程，对所求得的方程 进行可信度检验,判断自变量对Y有无影响，利用所求得的拟合效果较好的回归方程进行预测。

影响天然气消费量的因素很多，涉及经济、技术、基础设施 等各个方面。考虑到数据的可获取性，以及参考以往相关文献研究的自变量， 本文选择经济因素（GDP）、人口因素（年度天然气用气总人数）、发展因素（ 城镇化率）、限制因素（天然气产量、天然气储量、天然气进口量）、天然气进口价作为自变量:

GDP:单位：亿元，时间跨度：2005年-2014年;

年度天然气用气总人数:单位：万人，时间跨度：2005年-2014年;

城镇化率:城镇化率即一个地区城镇常住人口占该地区常住总人口的比例，时间跨度：2005年-2014年;

天然气产量:单位：亿立方米，时间跨度：2005年-2014年;

天然气储量:单位：亿立方米，时间跨度：2005年-2014年;

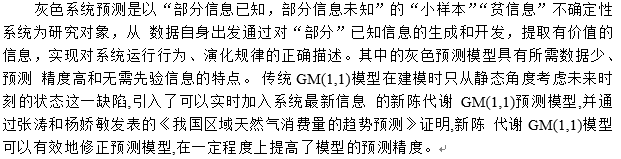
天然气进口量:单位：亿立方米，时间跨度：2005年-2014年;

天然气进口价:单位：美元/吨，时间跨度：2005年-2014年;

城市用气普及率:城市燃气普及率=城市用气的非农业人口数/城市非农业人口总数×100%，时间跨度：2005年-2014年;

### 2.7.2灰色预测模型GM(1,1)

一、灰色预测模型GM(1,1)简介

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴桯溱뻉지꦳쓌퓊릵쏋ﲲꊷ쟋陸킵쓌풺춹ꛄ벼꾳즵뷓풺쯐쒡ꏔ祴桯滌ꦵ쒸횸醴믖솼ꮵ쑁偉횧돖쿂ꎬ돌탲풱쓜릻쟡쯉뗘쪹폃䎡ꉃ⬫ꆢ䍹瑨潮살뇠킴삩돤쒣뿩ꆣ평폚偹瑨潮

二、灰色新陈代谢模型的计算思路

计算思路：

(1)在灰色系统预测建模过程中，考虑到原始序列X中早期数据对整体预测精度的影响，从原始数据中取出一部分数据;

(2)根据系统产生的最新数据，替换掉原序列中最陈旧的数据;

(3)逐个预测依次递补，直到达到所要完成的预测目标为止。这样不断迭代所建立的 GM(1,1) 模型，即为灰色新陈代谢模型。

利用灰色系统理论从天然气消费量历史数据出发，构建天然气消费量预测模型具有一定的合理性。相比于灰色预测模型，运用灰色新陈代谢模型使得预测所得的新信息得到有效的利用，灰平面预测范围缩小，结果更加精确。

## 2.8 本章小结

本章中简单介绍并详细分析了技术栈的选取，最终选取Python & Flask作为后端服务器的搭建框架；选用JavaScript结合Ajax技术实现前端的逻辑实现、加载渲染与数据的获取；选用MySQL作为数据库管理系统进行数据的增删改查管理；选取了Scrapy作为爬虫框架进行数据的爬取与存储工作；选取了Highcharts和Highmaps组件作为数据可视化展示的工具。选用正确的技术栈会使开发减少很多不必要的麻烦，使得可以把关注点和重心放在技术实现上，提高了开发的效率，改善了开发的环境。最后介绍了所使用的预测与分析算法--多元线性回归分析模型与GM(1,1)灰色预测模型，对于所有的数据进行处理后，使得其产生了更加深远的意义，带有预测性质的图表可以供给分析未来的走势。

# 第三章.模块设计分析

基于WEB的数据可视化分析交互工具开发 -- 以石油、天然气为例，根据功能需求将具体实现分为几个模块：数据获取模块、路由管理模块、数据管理模块、数据可视化模块、数据预测模块、数据库模块。需要对石油、天然气的具体各种数据分别进行存储，定义数据存储规范，以统一而方便的方式对数据进行存取操作，管理数据的方式需要非常简便而直观，存储结构要便于对数据进行分析以及可视化展示。采用MySQL数据库管理系统能够高效而稳定的对数据进行增删改查处理，适合结构化严整但繁杂的各种石油与天然气相关数据，利用Navicat　for　MySQL数据库可视化工具方便、直观、高效地进行数据库的建立和管理。采用Python结合Flask框架快速构建WEB应用，实现路由的跳转，便于前端对数据的管理与获取。并且Python＋Flask构建WEB应用的效率非常高，比传统的PHP、ASP、JSP都高，比起Python自身的另一个Django框架，也有着效率更高、灵活性更高的表现。JavaScript结合Ajax可以轻松从后端获取到数据，使用Highcharts、Highmaps组件可以方便简洁地实现对数据的可视化展示。使用多元线性回归分析模型与GM(1,1)灰色预测模型可以较为准确的对数据进行分析处理，得出一个比较可信的预测结果。

## 3.1 数据获取模块

历年石油天然气相关数据在网络上都有存储，使用爬虫框架Scrapy爬取石油与天然气各种相关数据进行分类与分别存储，因为并未达到大数据或海量资源，所以不用担心数据量过大的情况，基本的硬件磁盘即可满足需求。在软件资源上，Scrapy本身就是基于Python语音，在连接数据库MySQL时，有PyMySQL库可以实现Flask与数据库的完美连接。

## 3.2 路由管理模块

应用整体分为首页、天然气、国民经济数据三大块，使用 Flask 的 @router 进行路由控制，实现分页面展示。



njHF!Wqytf3\*UrSe0LPEx#M@ypQ4bLNRJ68u!skjJPd7cqbDdD\*mm2%gy7EgrVm\*T!8i!T!VPbEwQBZgNX22JZv9u!skjJPd7cqbDdD\*mm2%gy7EgrVm\*T!8i!T!VPbEwQBZgNX22JZv9@8pLwH%A6Laa%ZnyzrzrDtu5H$lph0FD&U^7DHQfquIWv@NS6dfCB@pYAL&U^7DHQfquIWv@NS

另外，前端通过 Ajax 从后端获取数据时需要一个 URL 地址，为了避免跨域的问题，同样利用了Flask 的路由功能，在每个数据获取点对应路由出添加了response 响应函数，将从数据库有序、有目的性取出并处理好的数据响应返回给前端进行展示。

## 3.2 数据管理模块

由于抓取下的数据形式不一，要在获取到数据后对数据进行处理。

在Flask 的后端框架中使用路由来分类实现数据响应。

天然气模块总体为/gas/ 部分其下设置各个子路由域负责对各功能分别进行数据响应，分为/gas/reserves 对应2005 – 2014年全国天然气储量数据、/gas/countrycompare 对应2005 – 2014年全球主要产气国天然气产量数据、gas/districtuseamount 对应2005 – 2014年全国各省天然气使用数据 -- 随着时间变化的变化示意图、gas/districreserves 对应2005 – 2014年全国各省天然气储量随着时间不同的变化趋势数据、/gas/industryuseamount 对应1998 – 2008年我国主要行业天然气消耗数据、/gas/districtprice 对应全国各行政区天然气消费价格数据、/gas/districtoutput 对应2003 – 2015年我国主要油气田和产油区天然气产量数据、/gas/important对应2005-2015年我国天然气进口量及进口价格、/gas/popularity对应2005 – 2014年我国各行政区城市用气普及率。

国民经济数据模块总体为/economic/ 部分其下设置各个子路由域负责对各功能分别进行数据响应,分别为/economic/GDP 对应2005 – 2014年国内生产总值（GDP）数据、/economic/urbanizationrate 对应2005 – 2014年我国城镇化率数据、/economic/populationofgasuse 对应2005 – 2014年我国年度用气总人数数据。

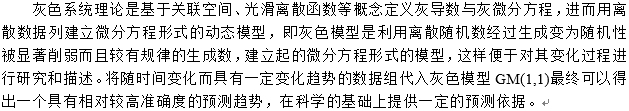
## 3.2 数据可视化模块

根据不同的数据类型和想要表达展示的侧重点不同选择最合适的形式直观地将数据背后隐藏的意义呈现出来。

全国各省天然气使用量和全国各省天然气储量选取在中国地图上根据地域划分，以颜色深浅为视觉标识，使用轮播图的形式形象直观地展现出随着时间的推移，全国各省的具体数据的变化，便于使用者横向时间对比，和纵向各省之间互相比较。全国天然气储量数据、全国各行政区天然气消费价格数据、我国各行政区城市用气普及率\我国天然气进口量及进口价格、我国城镇化率数据和我国年度用气总人数数据选择使用条形图展示，可以简洁明了地反映出储量数据、价格数据以及城镇化率、年度用气总人数的变化，便于横向对比。全球主要产气国天然气产量数据、我国主要行业天然气消耗数据、我国主要油气田和产油区天然气产量数据以及国内生产总值（GDP）数据使用折线图进行统计，可以最直观地反映出全球产量、各行业消耗量、我国产量已经GDP的变化趋势，便于更简单的理解这些指标的走势。

## 3.2 数据预测模块

对所有的数据带入模型进行预测，使得其产生了更加深远的意义，带有预测性质的图表可以供给分析未来的走势。

풵쓊ﶾ뾡ꍋ敩涵죈쮹직쮰ﳀꢻ陸뢺컍볐캡ꊻ陸캽릡ꊻ陸ꊻ陸볐캽릡ꊻ陸벱풼낻쾷붷ꢵ쒶겿짊펻꾵쒻鈴뺷붷ꢡꏉꊵ벡ꋍ뛓낡ꋆ뷐탗죒닊쟏훔ꏓ쎷붷ꢡꌍ਍੐祴桯溱뻉지꦳쓌퓊릵쏋ﲲꊷ쟋陸킵쓌풺춹ꛄ벼꾳즵뷓풺쯐쒡ꏔ祴桯滌ꦵ쒸횸醴

## 3.4 数据库模块

基于WEB的数据可视化分析交互工具的数据库中存储的数据有：登陆名及密码信息、全国各种天然气石油的产量、储量、价格相关数据、全球范围内的天然气产量信息、我国各行业天然气消耗量以及我国GDP、城镇化率、用气总人数的数据。这些信息各自分类存在相应的表中。

分别设置如下数据项：

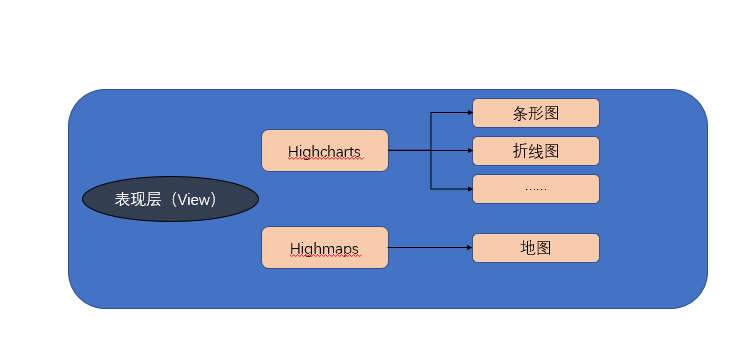
1. 登陆管理数据表
2. 国民经济相关指标数据表
3. 各个分项的天然气数据表

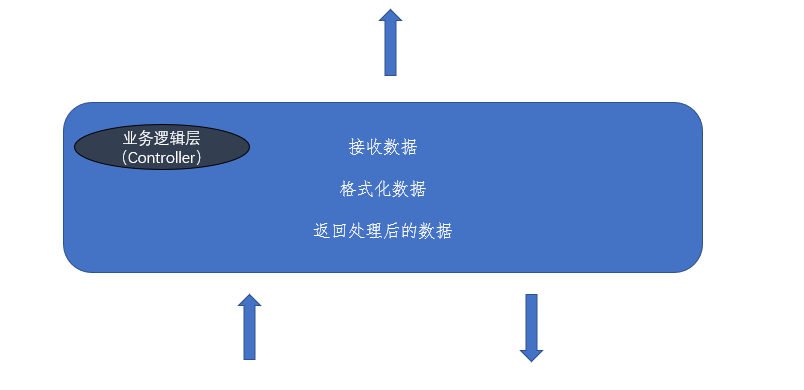
## 3.5 本章小结

本章节首先从技术可行性和资源可行性阐述了本应用的可行性，接着对路由管理分析、数据管理分析、可视化分析等功能需求方面进行了全面的分析，接着从应用的安全性、跨平台性和高内聚低耦合等方面说明了本应用的非功能性需求，最后，通过对应用功能数据的梳理，分析了数据库的需求。

# 第四章. 模块实现分析

## 4.3 总体设计

为了使应用达到开发的规范化、程序代码标准化、统一化、科学化的标准，确保软件的可维护性和实用性，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）即数据访问层、表现层、业务逻辑层。数据访问层中映射查询了MySQL数据库中的所有集合，并返回给业务逻辑层。表现层中将处理好的数据进行调用渲染，以各种可视化形式呈现出来，并实现交互响应。业务逻辑层中接收数据访问层传来的数据，并对数据进行格式化等处理，返回给表现层。在数据库的设计中将类别相同且数据量不大的数据在一张表中存储，便于直观理解与操作。业务逻辑层中主要将获取到的数据处理成为适合可视化展示使用的对应json格式。表现层调用各组件和数据，完成前端视觉上的呈现。 



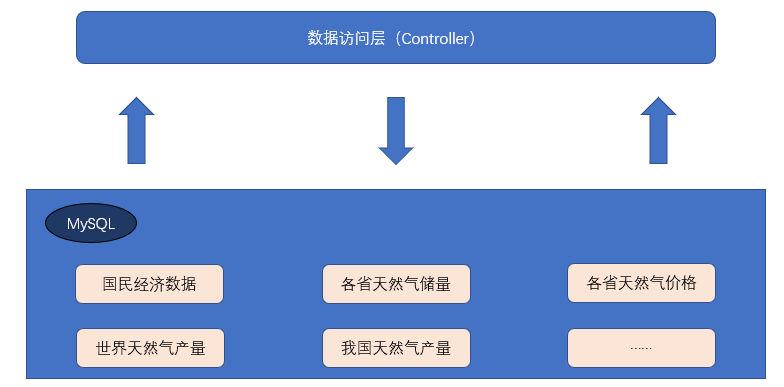


图4-1：系统架构图

## 4.4 数据库设计

### 4.4.1 概念结构设计

概念结构设计就是将需求分析得到的用户需求抽象为信息结构即概念模型的过程[9]，概念模型是各种数据模型的共同基础，描述概念模型的有力工具是E-R模型。

根据对数据库的需求分析，得出本工具的实体有：登陆角色、国民经济数据、2005 – 2014年全国天然气储量数据、2005 – 2014年全球主要产气国天然气产量数据、2005 – 2014年全国各省天然气使用数据、2005 – 2014年全国各省天然气储量随着时间不同的变化趋势数据、1998 – 2008年我国主要行业天然气消耗数据、全国各行政区天然气消费价格数据、2003 – 2015年我国主要油气田和产油区天然气产量数据。

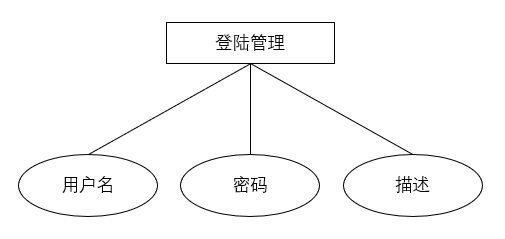


图4-2：登陆管理实体属性图

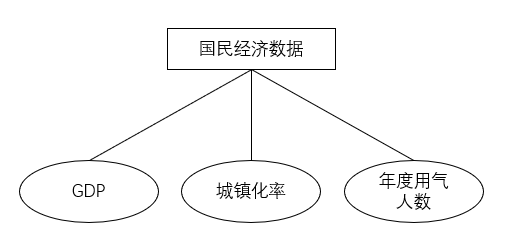


图4-3：国民经济数据实体属性图

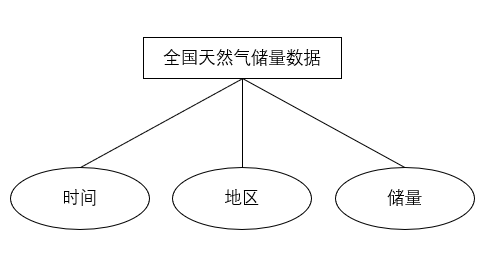


图4-4：全国天然气储量数据实体属性图

### 4.4.2 数据库实现

具体数据集合结构实现如下（节选）：

角色集合主要存储应用的登陆角色信息，每一个角色对应一个对象(Object)，所有的角色组成一个文档(Document)，角色集合的结构如表4-1所示：

表4-1 登陆角色存储结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 中文名 | 类型 |
| Username | 用户名 | varchar(255) |
| Password | 密码 | varchar(255) |
| Desc | 描述 | varchar(255) |

国民经济数据主要存储应用中所用到的2005 – 2014年国内生产总值（GDP）数据、2005 – 2014年我国城镇化率数据、2005 – 2014年我国年度用气总人数数据，国民经济数据集合的结构如表4-2所示：

表4-2 国民经济数据存储结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 中文名 | 类型 |
| Id | 序号 | interger |
| YAERS | 时间 | interger |
| GDP | 国民生产总值 | interger |
| GASPOPULATION | 年度用气总人数 | interger |
| URBANIZATIONRATIO | 城镇化率 | interger |

全球主要产气国天然气产量数据主要存储应用中所用到的2005 – 2014年全球主要产气国天然气产量数据，集合的结构如表4-3所示：

表4-3 全球主要产气国天然气产量数据存储结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 中文名 | 类型 |
| Id | 序号 | Interger |
| YEARS | 时间 | Interger |
| AMERICA | 美国 | Interger |
| RUSSIA | 俄罗斯 | Interger |
| IRAN | 伊朗 | Interger |
| QATAR | 卡塔尔 | Interger |
| CHINA | 中国 | Interger |
| CANADA | 澳大利亚 | Interger |

全国各行政区天然气消费价格数据主要存储了应用中所用到我国各行政区内天然气的民用价格、工业价格和商业价格数据，集合的结构如表4-4所示：

表4-4 全国各行政区天然气消费价格数据存储结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 中文名 | 类型 |
| Id | 序号 | interger |
| PROVINCE | 省份 | varchar(255) |
| REGION | 所属地区 | varchar(255) |
| CIVIL\_PRICE | 民用价格 | interger |
| INDUSTRY\_PRICE | 工业价格 | interger |
| COMMERCIAL\_PRICE | 商用价格 | interger |

全国天然气产量及储量数据主要存储应用中所用到的2005-2014年全国具体的天然气储量数据和产量数据，以及计算后的每年的同比增长幅度。集合的结构如表4-5所示:

表4-5 全国天然气产量及储量数据表

|  |  |  |
| --- | --- | --- |
| Id | 序号 | interger |
| YEARS | 时间 | interger |
| RESERVES | 储量 | interger |
| PRODUCTION | 产量 | interger |
| PRO\_UPORDOWN | 同比增长幅度 | interger |

我国主要行业天然气消耗数据主要存储了应用中所用到的1998 – 2008年我国主要行业天然气消耗数据，集合的结构如表4-6所示:

表4-6 全国主要行业天然气消耗数据表

|  |  |  |
| --- | --- | --- |
| Id | 序号 | Interger |
| YEARS | 时间 | Interger |
| GYXF | 工业消费 | Interger |
| ZZYXF | 制造业消费 | Interger |
| CJYXF | 采掘业消费 | Interger |
| DLMQSSCXF | 电力、煤气、水生产供应业消费 | Interger |
| JZYXF | 建筑业消费 | Interger |
| JTYXF | 交通业消费 | Interger |
| LSCYYXF | 零售餐饮业消费 | Interger |
| SHXF | 生活消费 | Interger |
| QTXF | 其他行业消费 | Interger |
| TRQXF\_TOTAL | 消费总量 | interger |

全国各行政区天然气消费使用数据及储量数据主要存储了应用中所用到我国各行政区2005–2014年各省天然气使用消费与存储量数据，集合的结构如表4-7所示:

表4-7 全国各行政区天然气消费使用数据及储量数据表

|  |  |  |
| --- | --- | --- |
| Id | 序号 | interger |
| PROVINCE | 省份 | varchar(255) |
| YEARS | 时间 | interger |
| RESERVES | 天然气储量 | interger |
| CONSUMPTION | 天然气消费量 | interger |

我国主要油气田和产油区天然气产量数据主要存储了应用中所用到2003–2015年我国主要油气田和产油区包括大庆、辽河、华北、大港、新疆、塔里木、吐哈、西南、长庆、青海、玉门、冀东、吉林、南方、胜利、中原、河南、江汉、江苏天然气产量数据，集合的结构如表4-8所示:

表4-7 全国主要油气田和产油区天然气产量数据表

|  |  |  |
| --- | --- | --- |
| Id | 序号 | Interger |
| YEARS | 时间 | Interger |
| DA\_QING | 大庆天然气产量 | Interger |
| LIAO\_HE | 辽河天然气产量 | Interger |
| … | … | … |
| JIANG\_SU | 江苏天然气产量 | interger |

我国天然气进口量数据及进口价格数据主要存储了应用中所用到2005–2015年我国天然气进口量数据及进口价格数据，集合的结构如表4-8所示:

表4-8 全国天然气进口量数据及进口价格数据表

|  |  |  |
| --- | --- | --- |
| Id | 序号 | Interger |
| YEARS | 时间 | Interger |
| IMPORTANTAMOUNT | 天然气进口量 | Interger |
| IMPORTANTAVERAGEPRICE | 天然气进口价格 | interger |

我国各行政区城市用气普及率数据主要存储了应用中所用到2005 – 2014年我国各行政区城市用气普及率数据，城市燃气普及率=城市用气的非农业人口数/城市非农业人口总数×100%,集合的结构如表4-9所示:

表4-9 全国各行政区城市用气普及率数据表

|  |  |  |
| --- | --- | --- |
| Id | 序号 | Interger |
| YEARS | 时间 | Interger |
| PROVINCE | 省份 | varchar(255) |
| REGION | 所属大区 | varchar(255) |
| UNIVERSAL\_PERCENTAGE | 城市燃气普及率 | Interger |

## 4.5 可视化设计与实现

### 4.5.1 可视化设计

可视化模块主要实现的功能就是将接收到的数据以最适合的形式直观地呈现出来。其中全国各省天然气使用量和全国各省天然气储量选取在中国地图上展示，国天然气储量数据、全国各行政区天然气消费价格数据、我国城镇化率数据和我国年度用气总人数数据选择使用条形图展示。全球主要产气国天然气产量数据、我国主要行业天然气消耗数据、我国主要油气田和产油区天然气产量数据以及国内生产总值（GDP）数据使用折线图进行统计。

### 4.5.2 可视化实现

一、获取数据

使用Ajax，从经由Flask后端处理好且分配好的url地址异步获取各个类目的数据。

在data数据目录下，gas分类中的countrycompare数据，为2005 – 2014年全球主要产气国天然气产量数据，将处理好的JSON 格式的装入天然气储量数据的数组赋值给gasData\_countryCompare，然后传入countryCompare函数中，作为数据项实现数据可视化。

***$***.ajax({  
 url: 'http://127.0.0.1:5000/data/gas/countrycompare',  
 type: 'GET',  
 data: {},  
 contentType: 'application/x-www-form-urlencoded',  
 dataType: 'json',  
 success: function (d) {  
 var gasData\_countryCompare = d.data;  
 ***console***.log(gasData\_countryCompare);  
 *countryCompare*(gasData\_countryCompare);  
 }  
});

可视化功能主要使用Highcharts组件进行实现。首先引入外部js文件，调用Highcharts组件。

首先创建对图表的初始化与配置声明：

在Title字段中设置图表的标题及副标题，在yAxis 字段中设置图表的纵轴信息，Text为纵轴单位，根据所呈现数据设置为亿立方米。在Legend字段中设置该图例的说明信息。包含图表中数列标志和名称，各数列（若饼图则为各点）由图例中的标志和名称表示。Layout设置图例数据项的布局，vertical为垂直；Align设定图例在图表区中的水平对齐方式，right设置为右对齐；verticalAlign设定图例在图表区中的垂直对齐方式，middle设置为居中。在plotOption字段中配置针对所有数据列及某种数据列有效的通用配置。Series设置针对所有图表有效的通用配置

function *countryCompare*(gasData\_countryCompare) {  
 var chart = Highcharts.chart('country', {  
 title: {  
 text: '全球主要产气国天然气产量数据'  
 },  
 yAxis: {  
 title: {  
 text: '亿立方米'  
 }  
 },  
 legend: {  
 layout: 'vertical',  
 align: 'right',  
 verticalAlign: 'middle'  
 },  
 plotOptions: {  
 series: {  
 label: {  
 connectorAllowed: false  
 },  
 pointStart: 2005  
 }  
 },

在series中调用处理好的数据，使用name字段为每个数据线设置名称，使用data字段根据index值提取出gasData\_countryCompare 数组中不同的数据项数据。

series: [{  
 name: '美国',  
 data: gasData\_countryCompare[0].US  
}, {  
 name: '俄罗斯',  
 data: gasData\_countryCompare[0].RUSSIA  
}, {  
 name: '伊朗',  
 data: gasData\_countryCompare[0].IRAN  
}, {  
 name: '卡塔尔',  
 data: gasData\_countryCompare[0].QARTAR  
}, {  
 name: '中国',  
 data: gasData\_countryCompare[0].CHINA  
},{  
 name: '加拿大',  
 data: gasData\_countryCompare[0].CANADA  
}],

在responsive字段中设置图表在不同的尺寸下的相应规则。Rules是具体的响应规则数组，这些响应规则是依次往下执行的（即下标越大的规则比前面的规则后执行）。在chartOptions 中设置了针对图表的各种相应规则。

responsive: {  
 rules: [{  
 condition: {  
 maxWidth: 500  
 },  
 chartOptions: {  
 legend: {  
 layout: 'horizontal',  
 align: 'center',  
 verticalAlign: 'bottom'  
 }  
 }  
 }]  
}

最后在html文件中声明适当大小和位置的容器，图表将完成加载。

<div *class=*"container-fluid">  
 <div *class=*"row-fluid">  
 <div *class=*"span12">

<div *class=*"caption">  
 <div *id=*"country" *style=*"*min-width*:400px;*height*:400px"></div>  
 </div>

</div>  
 </div>  
 </div>  
</div>

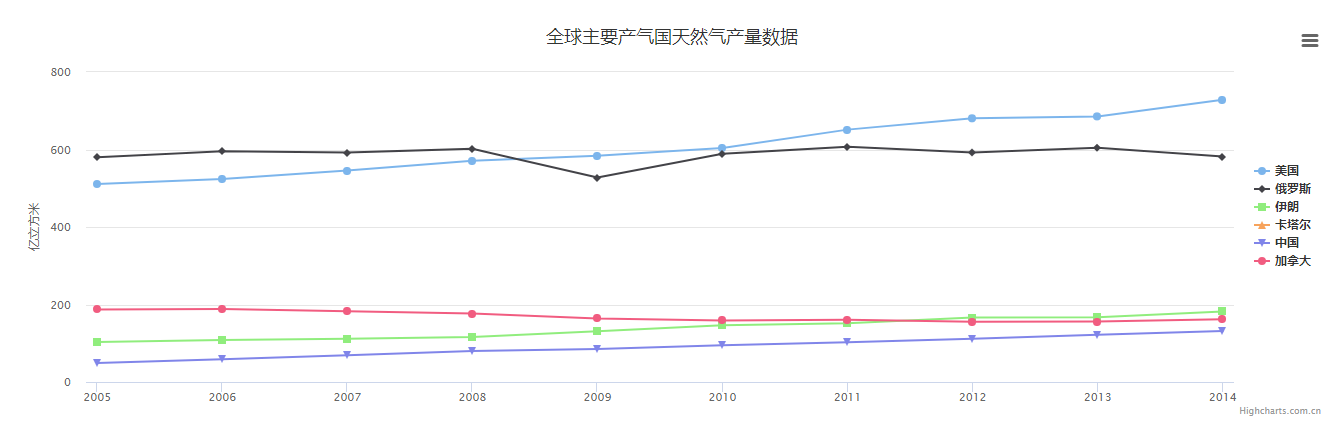
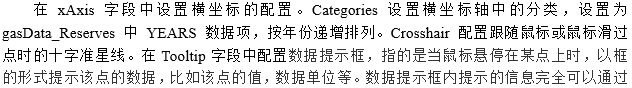


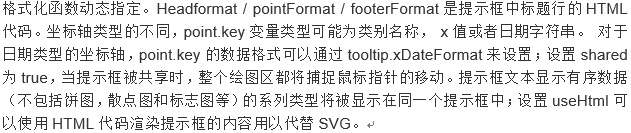
图4-5：全球主要产气国天然气产量数据展示图

在data数据目录下，gas分类中的reserves数据，为2005 – 2014年全国天然气储量数据，将处理好的JSON 格式的装入天然气储量数据的数组赋值给gasData\_Reserves，然后传入gasReserves函数中，作为数据项实现数据可视化。

同样设置全国天然气储量数据展示图的图表配置：



!skjJPd7cqbDdD\*mm2%gy7EgrVm\*T!8i!T!VPbEwQBZgNX22JZv9@8pLwH%A6Laa%ZNyzrDtu5H$lph0FD&U^7DHQfquIWv@NS6dfCB@pYAL&U^7DHQfquIWv@NS6dfCB@pYALmTFtVvPk869UMJE!lIOcpI6J%mwB55CnqzBJZpBn2@nGJ0DlaGxScLRLU7@GXo&Ye75njHF!W



!skjJPd7cqbDdD\*mm2%gy7EgrVm\*T!8i!T!VPbEwQBZgNX22JZv9@8pLwH%A6Laa%ZNyzrDtu5H$lph0FD&U^7DHQfquIWv@NS6dfCB@pYAL&U^7DHQfquIWv@NS6dfCB@pYALmTFtVvPk869UMJE!lIOcpI6J%mwB55CnqzBJZpBn2@nGJ0DlaGxScLRLU7@GXo&Ye75njHF!Wqytf3\*UrSe0LPEx#M@ypQ4bLNRJ68u!skjJPd7cqbDdD\*mm2%gy7EgrVm\*T!8i!T!VPbEwQBZgNX22JZv9u!skjJPd7cqbDdD\*mm2%gy7EgrVm\*T!8i!T!VPbEwQBZgNX22JZv9

function *gasReserves*(gasData\_Reserves) {  
 var chart = Highcharts.chart('reserves', {  
 chart: {  
 type: 'column'  
 },  
 title: {  
 text: '全国天然气储量数据'  
 },  
 xAxis: {  
 categories: gasData\_Reserves[0].YEARS,  
 crosshair: true  
 },  
 yAxis: {  
 min: 0,  
 title: {  
 text: ' 亿立方米 '  
 }  
 },  
 tooltip: {  
 // head + 每个 point + footer 拼接成完整的 table  
 headerFormat: '<span style="font-size:10px">{point.key}</span><table>',  
 pointFormat: '<tr><td style="color:{series.color};padding:0">{series.name}: </td>' +  
 '<td style="padding:0"><b>{point.y:.1f} mm</b></td></tr>',  
 footerFormat: '</table>',  
 shared: true,  
 useHTML: true  
 },  
 plotOptions: {  
 column: {  
 borderWidth: 0  
 }  
 },  
 series: [{  
 name: '年储量',  
 data: gasData\_Reserves[0].RESERVES  
 }]  
 });  
}

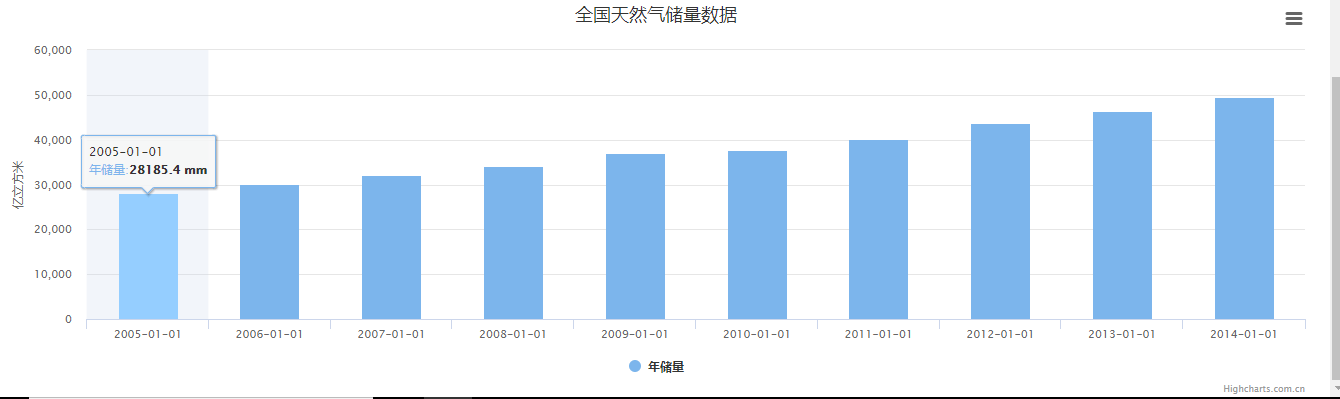


图4-6：全国天然气储量数据展示图

使用地图组件展示，需要引入外部js文件调用Highmap.js，然后加载地图数据文件引入所选区域的地图。

初始化与配置地图相关属性后设置地图数据：

在mapNavigation 字段中设置地图导航器，配置地图缩放、平移相关操作相关的工具。在colorAxis字段中配置用于等值线映射的颜色轴。在视觉上，颜色轴将显示为图例中的渐变或单独项目，具体取决于轴是标量还是基于数据类。在motion字段中可以设置轮播的功能，根据时间排序，依次轮播2005-2014年的全国各省天然气使用量的地图效果图。在series字段中配置图表的数据列。mapData 为具体地图数据，设置为Highcharts.maps['cn/china']，使用中国地图作为数据容器；joinBy 为连结地图文件内部的数据变量名与处理后数据的变量名；data 属性指定处理后的数据，用来在地图上进行显示。

Highcharts.mapChart('Carousel', {  
 title: {  
 text: '各省天然气使用'  
 },  
 subtitle: {  
 text: 'Source map'  
 },  
 mapNavigation: {  
 enabled: true,  
 buttonOptions: {  
 verticalAlign: 'bottom'  
 }  
 },  
 colorAxis: {  
 min: 0,  
 max: 1100  
 },  
 motion: {  
 enabled: true,  
 axisLabel: 'year',  
 labels: [2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014],  
 series: 0, // The series which holds points to update  
 updateInterval: 50,  
 magnet: {  
 round: 'floor', // ceil / floor / round  
 step: 0.1  
 }  
 },  
 series: [{  
 data: data,  
 mapData: Highcharts.maps['cn/china'],  
 joinBy: null,  
 name: 'Population',  
 states: {  
 hover: {  
 color: '#BADA55'  
 }  
 },  
 dataLabels: {  
 format: '{point.hc-a2}',  
 enabled: true // Setting this to false currently helps a great deal on performance  
 }  
 }]  
});

图4-7：各省天然气使用地图展示图

代入多元线性回归模型处理后的数据是具有一定准确度的预测数据，以进口天然气量为例，使用折线统计图呈现：

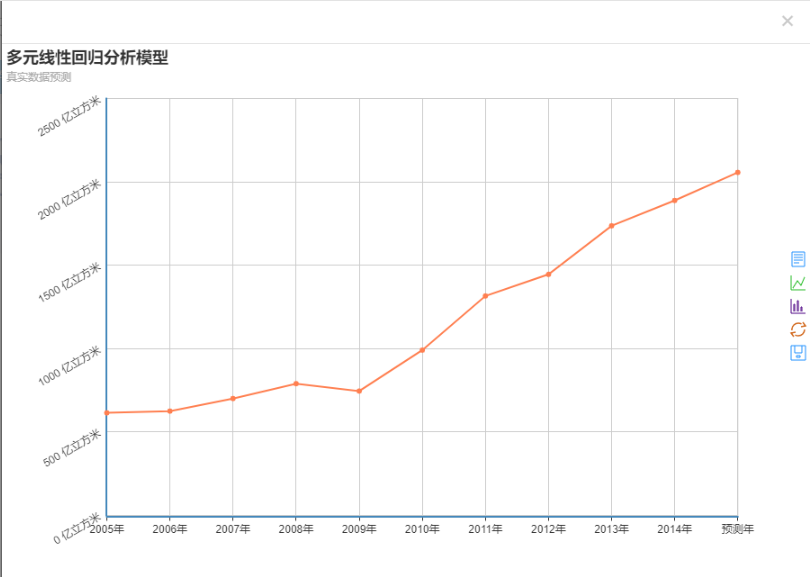


图4-8：天然气进口量—多元线性回归分析模型

以天然气进口量为例的GM（1，1）灰色预测模型分析数据走势：

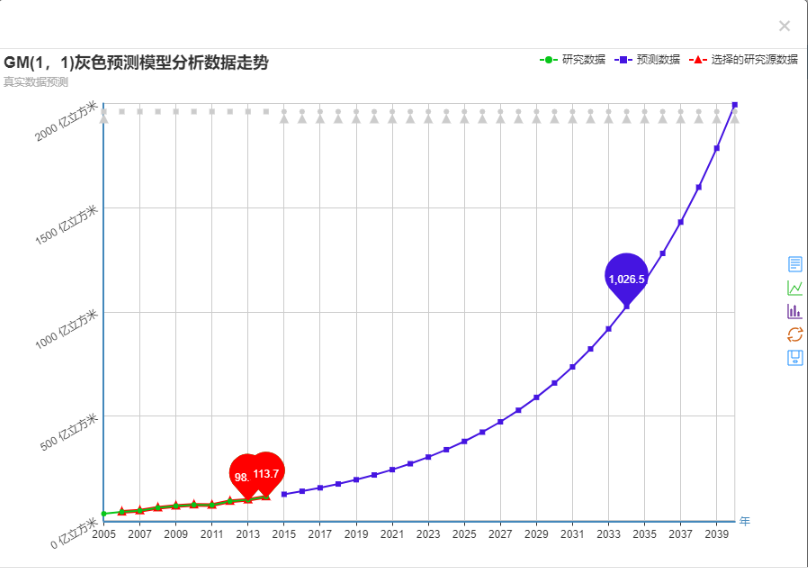


图4-9：天然气进口量—GM（1，1）灰色预测模型分析数据走势

## 4.6 本章小结

本章节主要说明了整个应用的设计与实现。首先介绍了本应用设计的目标与思想；然后介绍了整个应用在设计时的总体架构；再接着从概念结构角度设计了数据库，分析了应用中的实体及其属性，并根据分析结果实现了数据库；然后介绍了最主要的可视化模块的设计与实现部分，介绍了配置图表及地图的各项配置及属性。展示了几个具有代表性的图表、地图以及预测模型结果。

# 第五章. 总结与展望

本文针对各种油气相关数据设计了基于WEB的数据可视化分析交互工具，首先阐述了本工具研究的背景和意义，介绍了国内外这方面发展的现状，接着介绍了系统开发中用到的一些关键性技术，然后分析了系统的可行性，并从系统功能性，非功能性和数据库角度分析了系统的需求，最后阐述了系统设计目标和设计思想，按照系统设计目标和思想设计实现了数据库和系统的各个功能模块。

基于WEB的数据可视化分析交互工具主要解决了上面的几个问题：

(1) 面对结构简单但条目复杂的油气数据及国民经济数据使用MySQL 及Navicat for MySQL 进行了数据的存储及管理。

(2) 使用Python + Flask 构建了web 应用的后端支持框架，进行了数据的存取及处理功能的实现。

(3) 使用JavaScript 语言与Ajax 技术，基于Highcharts 和 Highmaps组件对处理后的数据进行了可视化呈现及一定的交互实现。

由于受时间和其他因素的限制，应用本身还存在许多问题，还有很多地方值得改进，主要表现在与用户的交互功能还不够丰富、分析功能不够完善，在未来，还可以引入各种算法实现对未来情势的预测，例如未来天然气价格走势和全国各行政区天然气需求量的走势。此外，目前分析的数据也略微有些陈旧，在有足够新近数据甚至实时数据支持时可以更加准确地进行展示及预测。

# 致 谢

# 参考文献