

豆瓣电影 TOP250 数据分析与可视化

目 录

目 录.....	1
第 1 章 绪 论.....	2
1.1 选题背景与意义.....	2
1.2 国内外研究现状和相关工作	3
1.3 本文的研究内容及主要工作	4
1.4 应用场景	4
第 2 章 开发环境	6
2.1 开发环境的搭建.....	6
2.2 Python 简介及安装	6
2.3 Python 相关库及模块的安装	9
第 3 章 技术工具介绍	11
3.1 爬取数据	11
3.2 数据可视化.....	13
第 4 章 系统功能设计及实现	16
4.1 获取数据	16
4.2 数据可视化.....	23
第 5 章 项目总结与展望	28
5.1 总结	28
5.2 展望	28

第1章 绪论

1.1 选题背景与意义

根据《中国互联网络发展状况统计报告》，到 2020 年 12 月 为 止， 全 年 新 增 网 民 6653 万，网民规模达 8.84 亿，普及率为 63.6%，与 2019 年底相比提升 3.9%。互联网的普及使得 网上的信息资源呈现爆炸式增长，大 数据时代 的到来，对如何在短时间内从网页中找到用户 需要的信息提出了挑 战，无论是搜索引擎还是 个人或者组织，要获取目标数据，都要从公开 网站 爬取数据，在这样的需求之下，网络爬虫 技术应运而生。网络爬虫，又被称为 网页蜘蛛或者网络 机器人，是指按照某种规则从网络上自动爬取 用户所需内 容的脚本程序。通常情况下，每个网页包含其他网页的入口，网络爬虫可以通 过 一个网址，链接进入其他网址获取内容 ，最后 返还给广大用户所需要的信 息数据。目前最适合用来网络爬虫的编程语言是 Python，Python 语言整合了 针对网络爬虫所需要的一系列库。

随着万维网的快速发展，因特网上的信息已经爆炸式增长。结果，人们在 互联网上找到他们所需的信息变得越来越困难，这直接导致了搜索引擎的出 现。搜索引擎在 Internet 上收集了数亿个页面，如何有效，准确地从这些页面 获取信息成为一个问题，网络爬虫由此诞生。 Web 爬网程序（在 FOAF 社区 中也称为 Web 蜘蛛，Web 机器人，通常是 Web 追踪器）是根据某些方法和规 则自动在万维网上爬网信息的程序或脚本。其他不常见的名称包括蚂蚁，自动 索引，模仿物或蠕虫。网络爬虫是一个程序或者脚本，人为的编写规则，网络 爬虫程序根据规则对指定网址进行信息的获取。趋近完美的程序拥有高效、精 准、及时的效果。

目前，数据可视化在各个领域都有广泛的应用。Python 是近年逐渐流行的 计算机高级语言，在词云、数据可视化、数据仓库与数据挖掘、仿真系统等方 面都有很多应用。该语言在数据可视化方面有很多优秀的可视化库，在学术界

受到广泛的关注。通过 Python 基本库绘制柱状图、折线图 etc 对实例进行可视化，展现利用 Python 进行数据可视化的优势。

网络爬虫并非一件容易的问题，想要实现一般需要面对两个大的问题：

（一）爬虫本身程序的问题：高并发的实现，分布式的实现，数据的筛选及存储。（二）对应的拥有信息的网站为了减少信息被爬取的或者为了减轻服务器负载，各种反爬虫措施带来的问题使得信息不那么容易获取，或者获取的信息为加密信息，需要筛选或者进行反加密处理。

1.2 国内外研究现状和相关工作

简单来说，数据可视化就是图表。最早的数据可视化非地图莫属。随着时间的推移、人类知识的增长、更多的地区被探索以及地理和天文仪器的充分发展，人们开始进行地图的绘制。解析几何的出现使得当时的人们能够在若干个维度进行数据分析，成为数据可视化历史中重要的一步。1765 年，威廉·普莱费尔创造了时间线图、条形图、饼图及时序图等。这在当时是一种前沿的思想，利用新的图表形式直观的表达数据。1850 年开始，数据可视化领域全面发展。这离不开统计学理论的建立。随着统计学的社会影响力逐渐提高，政府及学者开始对数据可视化进行严格标准化。20 世纪初，随着数理统计的诞生，数据可视化逐渐被大众所熟知，并且开始用于解决其他学科。后来随着计算机的发展，凭借计算机的数据处理精度和速度，聚类图、树形图等可视化形式开始出现。计算机与数据可视化的结合，使得数据可视化迎来了新的全面发展阶段。数据可视化在这一阶段实现了动态、可交互变化，允许了对图像和相关统计的实时特征的操控。

如今，社会已经迈进了大数据时代，传统的数据可视化分析技术已经不足以表达庞大的数据信息，需要依赖更有效的数据处理算法。在互联网时代，数据总是实时更新，然而实时数据的展示依赖于数据可视化。实时数据的价值只有数据可视化处理才能够体现。因此，如今数据可视化研究的重点在于如何建立一种动态的、可交互的分析方法来表达大规模的数据。例如每年的“双十一”天猫购物节，阿里巴巴都会将一些关键数据集中展示在网络平台上，时刻变化的数字代

表实时的营业销售额。每时每刻跳动的数字，就是数据可视化技术带来的结果。数据可视化技术与人们生活并非遥不可及。此外，阿里巴巴发现，通过数据可视化技术除了能够发现异常的交易外，还能够更直观地理解和服务客户，成功定位忠诚度高地顾客，从而制定精准化商业计划。

1.3 本文的研究内容及主要工作

本文设计出了一个基于 Python 的爬虫，用来爬取豆瓣网电影排行榜 Top250，并对这些数据进行获取以及可视化的分析，实现用户对豆瓣电影评价数据采集信息数据进行定向爬取后，对爬取到的信息进行存储、数据清洗及可视化分析。可视化部分运用 WordCloud 词云库结合 matplotlib 库绘制词云图、饼图和条形图。利用饼图来分析评分等级，繁冗的文字数据就被图形替代，能更直观看到大众对电影的评分等级分布[1]。使用词云图分析，能在频数统计的基础上，更加美观地展示数据，对于重点词语有更重点的突出展示。

以豆瓣网电影模块为例，实现了 Python 网络爬虫的全过程，并将爬虫结果保存在本地。主要分四个步骤实现，寻找爬虫入口，使用 re 和 requests 库获得所有电影信息的 url 链接，通过爬虫程序实现对豆瓣电影 Top 榜的爬取，使用 BeautifulSoup 库解析电影数据、获取到排名、电影名称、评分、评价人数、概括、简介等数据，将爬取到的信息保存到本地，再进行数据分析，并对这些数据进行可视化的分析，使得数据以直观明了的图表等方式将数据呈现给用户，图形图表能够直观地表达数据包含的信息。

1.4 应用场景

实践是最好的学习。在学习网络爬虫时，许多知识点的掌握并不牢靠，通过自己动手编程能够快速且扎实的提高自身的能力。本文中的网络爬虫程序不仅实现对象信息的定量抓取、信息筛选，获取的信息对应届毕业生亦不失为极其有用的信息。电影，在当今社会，作为人们在日常生活中不可缺少的一种娱乐方式，已经发展出百花争鸣的局面，让我们欣赏各种各样的影视剧。但是，

人们在看完电影之后，往往会有一些发自内心的感触，或许是同情主人公的悲欢离合，或许是对于故事的情节触目惊心，或许是对电影特效的与众不同，总之，人们在看完一部电影后或多或少都会将自己内心的所思所想告诉他人，或者是想了解他人是否同自己一样感同身受，因此，为了让更多的人可以方便地通过互联网相互之间交流对于电影的看法或是发布一些影评，或者可以从他人的影评中了解这部电影是否值得去看，于是一个对于影评排行信息的可视化分析就很有必要了。

本次对豆瓣网站电影排行的爬取实现，证明了网络规则越来越规范，便利了爬虫获取数据，爬虫可以简单快速地实现多种任务，它拥有更加广阔的应用前景。若目标是更全面地分析大众对该类影片的看法，可根据本文使用的爬虫方法扩大爬取广度，采集同类标签影片的评论数据和评分数据，以及加深爬取深度，分析受众人群特征，实现更全面的分析。而基于 Python 实现网络爬虫豆瓣电影模块的数据信息，可以根据爬虫得到的信息进行相关的市场分析，具有一定的商业价值。

第 2 章 开发环境

2.1 开发环境的搭建

本项目需要用到的开发环境包括：

(1) **PyCharm**：第三方开发工具，由著名的 JetBrains 公司开发，带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具，比如调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该 IDE 提供了一些高级功能，以用于支持 Django 框架下的专业 Web 开发。

(2) **IDLE**：Python 内置 IDE (随 python 安装包提供)，在安装 Python 后，会自动安装一个 IDLE。它是一个 Python Shell (可以在打开的 IDLE 窗口的标题栏上看到)，也就是一个通过输入文本与程序交互的途径，程序开发人员可以利用 Python Shell 与 Python 交互。初学者建议一开始可以使用 IDLE 来编写代码。

(3) **Python3.9.7**：我们在 Python 官网中可以下载到 Python 安装包，在这个安装包里有 Python 解释器 (负责执行运行 python 的程序)、Python 运行需要的基础库，一个命令行交互环境，以及交互式运行工具—Python Shell。

(4) **Anaconda5.3.1**：是一个开源的 Python 发行版本，其包含了 conda、Python 等 180 多个科学包及其依赖项。因为包含了大量的科学包，Anaconda 的下载文件比较大 (约 531 MB)，如果只需要某些包，或者需要节省带宽或存储空间，也可以使用 Miniconda 这个较小的发行版 (仅包含 conda 和 Python)。Anaconda 包括 Conda、Python 以及一大堆安装好的工具包，比如：numpy、pandas 等。

2.2 Python 简介及安装

2.2.1 Python 简介

Python 由荷兰数学和计算机科学研究学会的 Guido van Rossum 于 1990 年代初设计，作为一门叫做 ABC 语言的替代品。Python 提供了高效的高级数据结

构，还能简单有效地面向对象编程。Python 语法和动态类型，以及解释型语言的本质，使它成为多数平台上写脚本和快速开发应用的编程语言，随着版本的不断更新和语言新功能的添加，逐渐被用于独立的、大型项目的开发。

Python 解释器易于扩展，可以使用 C 或 C++(或者其他可以通过 C 调用的语言)扩展新的功能和数据类型。Python 也可用于可定制化软件中的扩展程序语言。Python 丰富的标准库，提供了适用于各个主要系统平台的源码或机器码。2021 年 10 月，语言流行指数的编译器 Tiobe 将 Python 加冕为最受欢迎的编程语言，20 年来首次将其置于 Java、C 和 JavaScript 之上。



图 2.2.1 Python3.9.7 标识

由于 Python 语言的简洁性、易读性以及可扩展性，在国外用 Python 做科学计算的研究机构日益增多，一些知名大学已经采用 Python 来教授程序设计课程。例如卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用 Python 语言讲授。众多开源的科学计算软件包都提供了 Python 的调用接口，例如著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK。而 Python 专用的科学计算扩展库就更多了，例如如下 3 个十分经典的科学计算扩展库:NumPy、SciPy 和 matplotlib，它们分别为 Python 提供了快速数组处理、数值运算以及绘图功能。因此 Python 语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表，甚至开发科学计算应用程序。

2.2.2 安装 Python

要进行 Python 开发，需要先安装 Python 解释器。因为 Python 是解释型编程语言，所以需要有一个解释器，这样才能运行我们编写的代码。

(1) 以 Windows 系统为例，打开 [python 官网](https://www.python.org/)，如果是 win10 则把鼠标

放到 Downloads 菜单上选择 python 3.9.7 进行下载，如果你的系统为 win7 请选择蓝色箭头所指的 Windows 菜单，在菜单中选择 3.9 以下的版本即可。

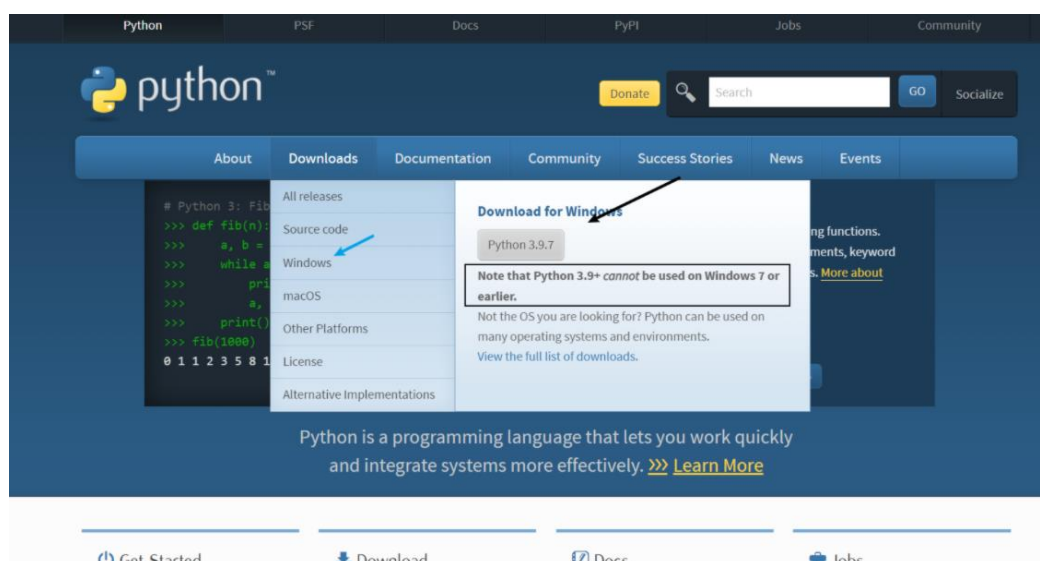


图 2.2.1 官网下载图示

(2) 在下载完成后，就可以双击安装包进行安装了，在安装界面中选中复选框 Add Python 3.x to PATH，勾选后可以将 Python 的安装路径，添加到环境变量 PATH 中，这样就可以在任意文件夹下使用 Python 命令了，单击 Install Now 按钮就可以开始安装了。

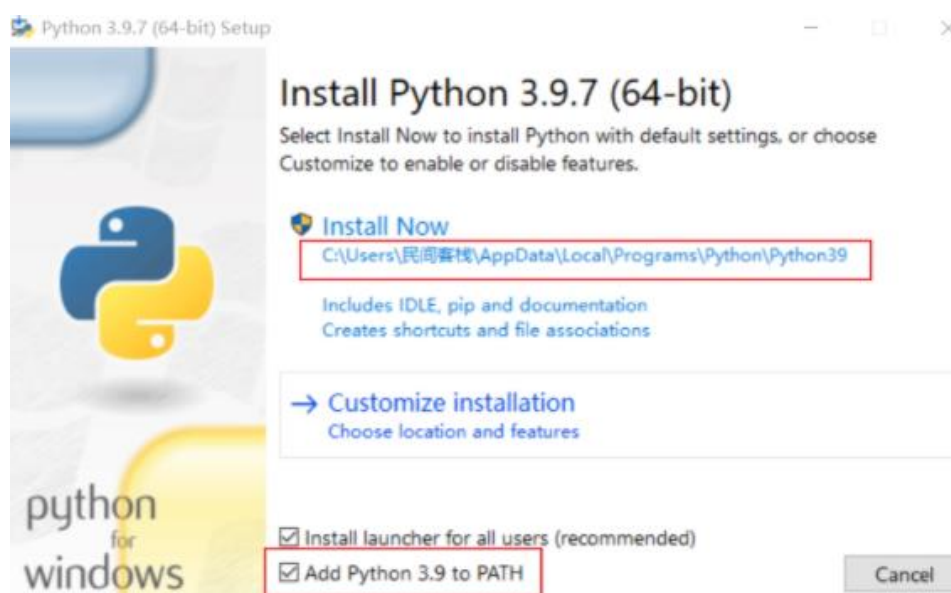


图 2.2.2 安装界面选项图示

(3) 打开命令提示符 (win + r) 输入 cmd，输入 python 后如下图出现版

本信息就说明安装完成。

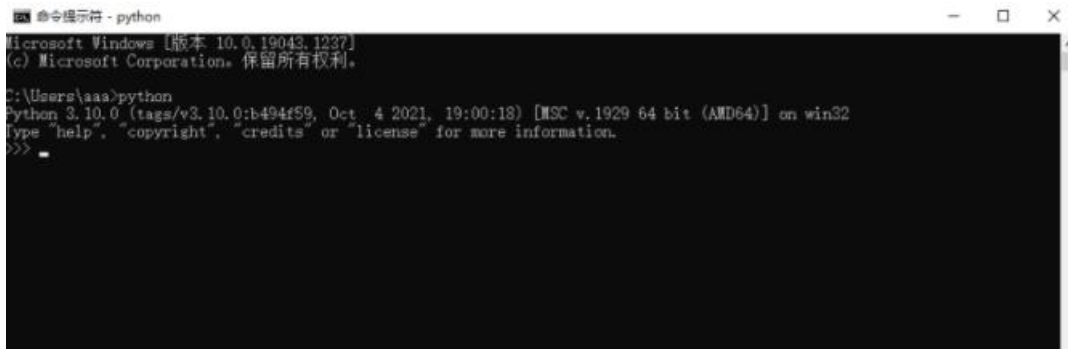


图 2.2.3 命令行窗口运行界面图

2.3 Python 相关库及模块的安装

Python 常用模块库下载及安装：

(1) 单文件模块：直接把文件拷贝到 `$python_dir/Lib`。

(2) 多文件模块：带 `setup.py` 后缀的下载模块包（压缩文件 `zip` 或 `tar.gz`），进行解压，键盘上按住 `win+R` 键输入 `cmd->cd` 进入模块文件夹，执行：`python setup.py install`。

(3) `easy_install` 方式：先下载 `ez_setup.py`，运行 `python ez_setup` 进行 `easy_install` 工具的安装，之后就可以使用 `easy_install` 进行安装 `package`（文件名称、资源的 URL、`.egg` 文件（python egg 文件）来下载安装文件）

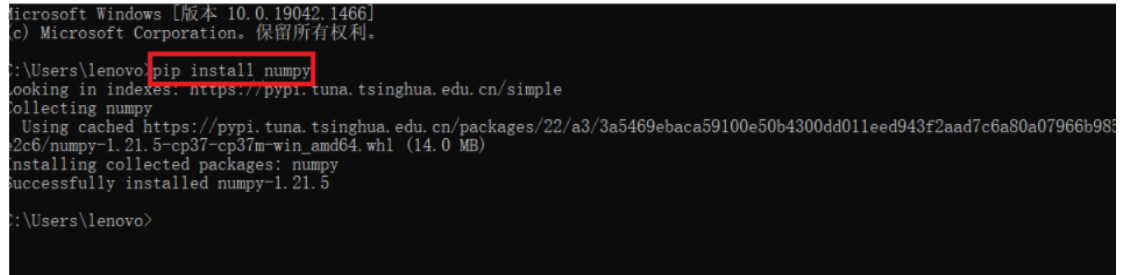
```
easy_install packageName
```

```
easy_install package.egg
```

(4) `pip` 网络搜索自动安装：先进行 `pip` 工具的安装：`easy_install pip`（`pip` 可以通过 `easy_install` 安装，而且也会装到 `Scripts` 文件夹下 `D:\Python2.7\Lib\site-packages`）

安装：`pip install PackageName`

例如：以 numpy 为例，直接打开 cmd 输入 `pip install numpy` 回车后得到如下图所示即表示安装成功。



```
Microsoft Windows [版本 10.0.19042.1466]
(c) Microsoft Corporation。保留所有权利。

C:\Users\lenovo>pip install numpy
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting numpy
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/22/a3/3a5469ebaca59100e50b4300dd011eed943f2aad7c6a80a07966b9832c6/numpy-1.21.5-cp37-cp37m-win_amd64.whl (14.0 MB)
Installing collected packages: numpy
Successfully installed numpy-1.21.5

C:\Users\lenovo>
```

图 2.3.1 命令行窗口安装界面图

pip 常用命令列出安装的 packages: `pip freeze`

安装特定版本的 package:通过使用 `==`, `>=`, `<=`, `>`, `<` 来指定一个版本号

`$ pip install 'Markdown<2.0'`

`$ pip install 'Markdown>2.0,<2.0.3'`

升级包到当前最新的版本，可以使用 `-U` 或者 `--upgrade`:

升级包: `pip install -U Markdown`

卸载包: `pip uninstall Markdown`

查询包: `pip search "Markdown"`

(5) .whl 文件 pip 方式: 下载对应模块.whl 文件，在 CMD->`cd` 命令下进入到.whl 文件所在目录，如果 pip 目录未添加到环境变量，最好把.whl 文件放置到 pip.exe 所在目录

安装: `pip install 包名.whl`

第3章 技术工具介绍

3.1 爬取数据

爬取数据是指利用一种装置，将来自各种数据源的数据自动收集到一个装置中，本文基于去哪儿网站，通过 Python 的爬虫技术获取相应数据，数据获取的过程为：发起请求、获取地址、分析网页、提取数据、存储数据。

（1）os 库

os 模块是 Python 标准库中的一个用于访问操作系统相关功能的模块，os 模块提供了一种可移植的使用操作系统功能的方法。使用 os 模块中提供的接口，可以实现跨平台访问。但是，并不是所有的 os 模块中的接口在全平台都通用，有些接口的实现是依赖特定平台的，比如 linux 相关的文件权限管理和进程管理。在使用 os 模块的时候，如果出现了问题，会抛出 OSError 异常，表明无效的路径名或文件名，或者路径名 (文件名)无法访问，或者当前操作系统不支持该操作。

（2）re 库

正则表达式 (regular expression) (regex)(RE)，用来简洁表达一组字符的表达式，由字符和操作符构成。正则表达式是对字符串操作的一种逻辑公式，可以通过一个特定的组合，从复杂烦琐的字符串中快速提取符合条件的字符子串。针对字符串表达“简洁”和“特征”思想的工具，判断某字符串的特征归属。正则表达式常用于文本处理，表达文本类型的特征（病毒、入侵等），同时查找或替换一组字符串，匹配字符串的全部或部分正则表达式的使用。

（3）time 库

time 库是 Python 中处理时间的标准库计算机时间的表达，提供获取系统时间并格式化输出功能，提供系统级精确计时功能，用于程序性能分析 time 库概述。time 库包括三类函数：时间获取：time () ctime () gmtime () ，时间格式化：strftime () strptime () ，程序计时：sleep (), perf_counter ()，时间获取. time ()获取当前时

间戳，即计算机内部时间值，浮点数。

（4）requests 库

Requests 是唯一的一个非转基因的 Python HTTP 库，可以安全享用，Requests 继承了 urllib2 的所有特性。Requests 支持 HTTP 连接保持和连接池，支持使用 cookie 保持会话，支持文件上传，支持自动确定响应内容的编码，支持国际化的 URL 和 POST 数据自动编码。Requests 库作为 Python 的第三方数据库，能自动解码来自服务器的内容并基于 HTTP 头部对相应的编码做出有根据性的推测，它对指定的 URL 发起请求，获取想要爬取的数据页面的响应信息。爬取网页数据最常使用的是通用爬取框架，它最大的作用就是让用户有效、稳定、可靠地爬取网页上的内容。

（5）BeautifulSoup 库

Beautiful Soup 是 python 的一个库，最主要的功能是从网页抓取数据。官方解释如下：

Beautiful Soup 提供一些简单的、python 式的函数用来处理导航、搜索、修改分析树等功能。它是一个工具箱，通过解析文档为用户提供需要抓取的数据，因为简单，所以不需要多少代码就可以写出一个完整的应用程序。Beautiful Soup 自动将输入文档转换为 Unicode 编码，输出文档转换为 utf-8 编码。你不需要考虑编码方式，除非文档没有指定一个编码方式，这时，Beautiful Soup 就不能自动识别编码方式了。然后，你仅仅需要说明一下原始编码方式就可以了。Beautiful Soup 已成为和 lxml、html6lib 一样出色的 python 解释器，为用户灵活地提供不同的解析策略或强劲的速度，可以更加方便地通过 css 选择器对元素进行定位，通过树形结构依次访问目标数据标签，具有方便、快捷等特性，因此它是爬虫程序中定位标签位置的首选。

（6）User-Agent 库

User-Agents 是 Python 的 UserAgent 解析库，通过解析浏览器或 HTTP 的 UserAgent 字符串，检测访问设备如手机、平板电脑及是否具备触摸能力。爬虫时，经常会需要提供 User-Agent，如果不提供 User-Agent，会导致爬虫在请求网

页时，请求失败，所以需要大量 User-Agent。

(7) openpyxl 库

Openpyxl 库是一个读写 Excel 2010 文档的 Python 库，如果要处理更早格式的 Excel 文档，需要用到额外的库，openpyxl 是一个比较综合的工具，是专门用来操作 Excel 的，使得两者可以无缝衔接，同时因为数据较少，用 Excel 来进行管理远比数据库来进行管理更加高效，

能够同时读取和修改 Excel 文档，其他很多的与 Excel 相关的项目基本只支持读或者写 Excel 一种功能。

3.2 数据可视化

数据可视化首先对数据进行一定的处理，包括统计、去重、整理等，然后以图形化的形式呈现，便于我们更好对其进行识别、判断[6]。数据可视化过程包括数据处理、可视化、词云分析等等。

(1) pandas 库

Pandas 是 Python 的一个数据分析包，该工具为解决数据分析任务而创建。Pandas 纳入大量库和标准数据模型，提供高效的操作数据集所需的工具，提供大量能使我们快速便捷地处理数据的函数和方法。Pandas 是字典形式，基于 NumPy 创建，让 NumPy 为中心的应用变得更加简单。

(2) pyecharts 库

pyecharts 是一个用于生成 Echarts 图表的类库。Echarts 是一个由百度开源的数据可视化工具，凭借着良好的交互性，精巧的图表设计，用 Echarts 生成的图可视化效果非常棒，为了与 Python 进行对接，方便在 Python 中直接使用数据生成图，得到了众多开发者的认可。而 python 是一门富有表达力的语言，很适合用于数据处理，当数据分析遇上了数据可视化时，pyecharts 诞生了。pyecharts 可以展示动态图，在线报告使用比较美观，并且展示数据方便，鼠标悬停在图上，即可显示数值、标签等。pyecharts 分为 v0.5 和 v1 两个大版本，v0.5 和 v1 两个版本不兼容，v1 是一个全新的版本，因此我们的学习尽量都是基于 v1 版本进行

操作。pyecharts 是由一个中国人开发的，也存在一个中文网站，这样学习起来就方便多了。

（3）options 函数

可以通过 Options 设置组件的属性，从而控制组件的各种状态。比如：宽度、高度、颜色、位置等等。

（4）bar 函数

bar 函数用于绘制条形图，每个条按给定的对齐方式定位在参数指定的位置。它们的宽、高尺寸由参数 height 和 width 决定。垂直基线由参数 bottom 指定，默认为 0。常用 bar(left, height, width, color, align, yerr)函数：绘制柱形图。left 为 x 轴的位置序列，一般采用 arange 函数产生一个序列；height 为 y 轴的数值序列，也就是柱形图的高度，一般就是我们需要展示的数据；width 为柱形图的宽度，一般这是为 1 即可；color 为柱形图填充的颜色；align 设置 plt.xticks()函数中的标签的位置；yerr 让柱形图的顶端空出一部分。

（5）numpy 库

Numpy 库是 Python 进行科学计算的基础库，它是一个由多维数组对象组成，包含数学运算、逻辑运算、形状操作、排序、选择、I/O、离散傅里叶变换、基本线性代数、基本统计运算、随机模拟等功能。

（6）matplotlib 库

matplotlib 是一款命令式、较底层、可定制性强、图表资源丰富、简单易用、出版质量级别的 python 2D 绘图库。它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形。本次课程设计我们基于 matplotlib 库绘制了按照年份来划分的柱状图，通过柱状图的描述，可以清晰的反映对比情况。

Matplotlib Pyplot Pyplot 是 Matplotlib 的子库，提供了和 MATLAB 类似的绘图 API。matplotlib.pyplot 是一些命令行风格函数的集合，使 matplotlib 以类似于 MATLAB 的方式工作。每个 pyplot 函数对一幅图片(figure)做一些改动：比如创建新图片，在图片创建一个新的作图区域(plotting area)，在一个作图区域内画直线，给图添加标签(label)等。matplotlib.pyplot 是有状态的，亦即它会保存当前图

片和作图区域的状态，新的作图函数会作用在当前图片的状态基础之上。

（7）wordcloud 库

wordcloud 库，一款展示词云图的第三方库，以词语为基本单位，用于生成各种漂亮的词云图。词云是一种典型的文本可视化技术。词云对文本中出现频率较高的“关键词”予以视觉上的突出，从而形成“关键词云层”或“关键词渲染”，词条在词云中所占区域的大小代表了它出现频率的高低，从视觉上达到更加直观的效果。

（8）jieba 库

jieba 是一个 python 实现的分词库，对中文有着很强大的分词能力，是一款优秀的 Python 第三方中文分词库，jieba 支持三种分词模式：精确模式、全模式和搜索引擎模式，下面是三种模式的特点。精确模式：试图将语句最精确的切分，不存在冗余数据，适合做文本分析；全模式：将语句中所有可能是词的词语都切分出来，速度很快，但是存在冗余数据；搜索引擎模式：在精确模式的基础上，对长词再次进行切分。

第4章 系统功能设计及实现

本章是项目的核心内容，主要是详细完成整体课程设计的过程。（按照设计要求完成并包含源代码）

4.1 获取数据

数据的作用主要表现在数据的分析使用。通过对信息数据进行分析，不仅能把隐藏的数据挖掘出来，还能通过这些隐藏的讯息，挖掘出有利于决策的信息。

4.1.1 获取爬虫所需的 header 和 cookie

（1）首先进入到豆瓣电影 TOP250 页面，[豆瓣电影 Top 250 \(douban.com\)](https://douban.com/top250)



图 4.1.1 豆瓣 top250 电影页面

（2）按下 F12，就会出现网页的 js 语言设计部分，在开发者工具网页中找到 network 部分，先清空后再刷新，浏览名字（name）部分，找到第一个 TOP250 文件，鼠标右键，选择复制，复制下当前网页的 URL（bash），如下图所示：

cURL command

Examples: GET - POST - Basic Auth

```
curl 'https://movie.douban.com/top250' \
-H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9' \
-H 'Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6' \
-H 'Cache-Control: max-age=0' \
-H 'Connection: keep-alive' \
```

Ansible CFML Dart Elixir Go Java JavaScript JSON Node.js Python R Rust Strest

```
import requests

cookies = {
    'll': '"118284"',
    'bid': '"Xj0Itg5gDOM"',
    '_vwo_uuid_v2': '"D3F65F081728CABE635B6C3B3D4A84703|09f26d8f468468453243db53d24e3cf9"',
    '__gads': '"ID=99d040e8ec36e924-221e2ac15cd400b2:T=1655126776:RT=1655126776:S=ALNI_MaHQQ04j9IKLX0EP5RxiPH_O1Luxg"',
    '_pk_ref.100001.4cf6': '"%5B%22%22%22%22%2C1655257549%2C%22https%3A%2F%2Fwww.baidu.com%2Flink%3Furl%3D40Vxz07qyhMgE2q09ZT3pssMr33AaE"',
    'ap_v': '"0.6.0"',
    '_utma': '"30149280.1704592545.1655124507.1655126668.1655257549.3"',
    '_utmc': '"30149280"',
    '_utmz': '"30149280.1655257549.3.3.utmcsr=baidu|utmccn=(organic)|utmcmd=organic"',
    '_utma': '"223695111.250833532.1655124507.1655126668.1655257549.3"',
    '_utmc': '"223695111"',
    '_utmz': '"223695111.1655257549.3.3.utmcsr=baidu|utmccn=(organic)|utmcmd=organic"',
    '__gpi': '"UID=0000069bbd1484a5:T=1655126776:RT=1655257551:S=ALNI_MbfDLLBG0vb8wBxUvREiOL7biWdQQ"',
    '_pk_id.100001.4cf6': '"fe527758b4762aca.1655124508.3.1655257814.1655126789."'
}
```

图 4.1.4 cookie 页面

```
headers = {
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9',
    'Accept-Language': 'zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6',
    'Cache-Control': 'max-age=0',
    'Connection': 'keep-alive',
    # Requests sorts cookies= alphabetically
    # 'Cookie': '"ll=118284"; bid=Xj0Itg5gDOM; _vwo_uuid_v2=D3F65F081728CABE635B6C3B3D4A84703|09f26d8f468468453243db53d24e3cf9; __gads=ID=99d040e8ec36e924-221e2ac15cd400b2:T=1655126776:RT=1655126776:S=ALNI_MaHQQ04j9IKLX0EP5RxiPH_O1Luxg; _pk_ref.100001.4cf6=%5B%22%22%22%22%2C1655257549%2C%22https%3A%2F%2Fwww.baidu.com%2Flink%3Furl%3D40Vxz07qyhMgE2q09ZT3pssMr33AaE"',
    'Referer': 'https://www.baidu.com/link?url=40Vxz07qyhMgE2q09ZT3pssMr33AaE',
    'Sec-Fetch-Dest': 'document',
    'Sec-Fetch-Mode': 'navigate',
    'Sec-Fetch-Site': 'cross-site',
    'Sec-Fetch-User': '?1',
    'Upgrade-Insecure-Requests': '1',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36 Edg',
    'sec-ch-ua': '"Not A:Brand";v=99", "Chromium";v=102", "Microsoft Edge";v=102"',
    'sec-ch-ua-mobile': '?0',
    'sec-ch-ua-platform': '"Windows"',
}

response = requests.get('https://movie.douban.com/top250', cookies=cookies, headers=headers)
```

图 4.1.5 header 页面

4.1.2 导入以下的模块

我们在 jupyter 中编写代码，首先导入我们需要的模块，Python 中导入库直接用“import+库名”，需要用库里的某种方法用“from+库名+import+方法名”。

```
import os
import re
import time
import requests
from bs4 import BeautifulSoup
from fake_useragent import UserAgent
from openpyxl import Workbook, load_workbook
```

4.1.3 获取每页电影链接

我们要爬取的是豆瓣电影 TOP250，该网站地址是豆瓣电影 TOP250 页面，[豆瓣电影 Top 250 \(douban.com\)](https://movie.douban.com/top250)，通过给定的 url 并用 `requests.get()` 方法来获取页面的 text。

```
def getonepagelist(url,headers):
    try:
        r = requests.get(url, headers=headers, timeout=10)
        r.raise_for_status()
        r.encoding = 'utf-8'
        soup = BeautifulSoup(r.text, 'html.parser')
        lsts = soup.find_all(attrs={'class': 'hd'})
        for lst in lsts:
            href = lst.a['href']
            time.sleep(0.5)
            getfilminfo(href, headers)
    except:
        print('getonepagelist error!')
```

4.1.4 获取每部电影具体信息

```
def getfilminfo(url,headers):
    filminfo = []
    r = requests.get(url, headers=headers, timeout=10)
    r.raise_for_status()
    r.encoding = 'utf-8'
    soup = BeautifulSoup(r.text, 'html.parser')
    # 片名
    name = soup.find(attrs={'property': 'v:itemreviewed'}).text.split(' ')[0]
    # 上映年份
    year = soup.find(attrs={'class': 'year'}).text.replace('(', '').replace(')', '')
    # 评分
    score = soup.find(attrs={'property': 'v:average'}).text
    # 评价人数
    votes = soup.find(attrs={'property': 'v:votes'}).text
    infos = soup.find(attrs={'id': 'info'}).text.split('\n')[1:11]
    # 导演
    director = infos[0].split(':')[1]
    # 编剧
    scriptwriter = infos[1].split(':')[1]
```

```

# 主演
actor = infos[2].split(':')[1]
# 类型
filmtype = infos[3].split(':')[1]
# 国家/地区
area = infos[4].split(':')[1]
if '.' in area:
    area = infos[5].split(':')[1].split('/')[0]
# 语言
language = infos[6].split(':')[1].split('/')[0]
else:
    area = infos[4].split(':')[1].split('/')[0]
# 语言
language = infos[5].split(':')[1].split('/')[0]
if '大陆' in area or '香港' in area or '台湾' in area:
    area = '中国'
if '戛纳' in area:
    area = '法国'
# 时长
times0 = soup.find(attrs={'property': 'v:runtime'}).text
times = re.findall('\d+', times0)[0]
filminfo.append(name)
filminfo.append(year)
filminfo.append(score)
filminfo.append(votes)
filminfo.append(director)
filminfo.append(scriptwriter)
filminfo.append(actor)
filminfo.append(filmtype)
filminfo.append(area)
filminfo.append(language)
filminfo.append(times)
filepath = 'TOP250.xlsx'
insert2excel(filepath, filminfo)

```

4.1.5 保存数据

```

def insert2excel(filepath, allinfo):
    try:
        if not os.path.exists(filepath):
            tableTitle = ['片名', '上映年份', '评分', '评价人数', '导演', '编剧', '主演', '类型', '国家/地区', '语言', '时长(分钟)']

```

```

        wb = Workbook()
        ws = wb.active
        ws.title = 'sheet1'
        ws.append(tableTitle)
        wb.save(filepath)
        time.sleep(3)
    wb = load_workbook(filepath)
    ws = wb.active
    ws.title = 'sheet1'
    ws.append(allinfo)
    wb.save(filepath)
    return True
except:
    return False

```

4.1.6 主函数入口

```

if __name__ == '__main__':
    ua = UserAgent(verify_ssl=False)
    headers = {
        'User-Agent': ua.random,}
    cookies = {
        'll': '"118284"',
        'bid': 'Xj0Itg5gDOM',
        'ap_v': '0,6.0',
        '__utmc': '30149280',
        '__utmc': '223695111',
        '_vwo_uuid_v2':
'D3F65F081728CABE635B6C3B3D4A84703|09f26d8f468468453243db53d24e3cf9',
        '_pk_ref.100001.4cf6':
'%5B%22%22%2C%22%22%2C1655126668%2C%22https%3A%2F%2Fcn.bing.com%2F%22
%5D',
        '_pk_ses.100001.4cf6': '*',
        '__utma': '30149280.1704592545.1655124507.1655124507.1655126668.2',
        '__utmb': '30149280.0.10.1655126668',
        '__utmz':
'30149280.1655126668.2.2.utmcsr=cn.bing.com|utmccn=(referral)|utmcmd=referral|utmcct=/',
        '__utma': '223695111.250833532.1655124507.1655124507.1655126668.2',
        '__utmb': '223695111.0.10.1655126668',
        '__utmz':
'223695111.1655126668.2.2.utmcsr=cn.bing.com|utmccn=(referral)|utmcmd=referral|utmcct=/',
        '_pk_id.100001.4cf6': 'fe527758b4762aca.1655124508.2.1655126773.1655124508.',

```

```
print(f'第{i}页爬取完成')
```

```

正在爬取第0页, 请稍候...
第0页爬取完成
正在爬取第1页, 请稍候...
第1页爬取完成
正在爬取第2页, 请稍候...
getonepage2ist error!
第2页爬取完成
正在爬取第3页, 请稍候...
getonepage2ist error!
第3页爬取完成
正在爬取第4页, 请稍候...
第4页爬取完成
正在爬取第5页, 请稍候...
第5页爬取完成
正在爬取第6页, 请稍候...
第6页爬取完成
正在爬取第7页, 请稍候...
第7页爬取完成
正在爬取第8页, 请稍候...
getonepage2ist error!
第8页爬取完成
正在爬取第9页, 请稍候...
getonepage2ist error!
第9页爬取完成
正在爬取第10页, 请稍候...
第10页爬取完成

```

爬取结果:

2	电影链接/影片链接	影片中英文名影片外国名称	评价数	概况	相关信息
3	https://movhttps://img	肖申克的救赎 The Shaw's 9.7	2633695	希望让人自由导演: 弗兰克·德拉邦特 Frank Darabont 主演: 蒂姆·罗宾斯 Tim Robbins / ... 1994 / 美国 / 犯罪 剧情	
4	https://movhttps://img	卧虎藏龙 9.6	1956209	一部美国大片 导演: 陈凯歌 Kaige Chen 主演: 张震张 Leslie Cheung / 张丰毅 Fong Yee Zha ... 1993 / 中国 / 中国香港 / 剧情 爱情 同性	
5	https://movhttps://img	朗读者 9.6	1797914	东德美国导演: 罗伯特·泽尔伯格 Robert Zemeckis 主演: 汤姆·汉克斯 Tom Hanks / ... 1994 / 美国 / 剧情 爱情	
6	https://movhttps://img	费城故事 9.4	1940400	失去的女主角导演: 詹姆斯·梅隆 James Cameron 主演: 桑岛阿诺拉·达普里奥 Leonardo ... 1997 / 美国 墨西哥 澳大利亚 加拿大 / 剧情 爱情	
7	https://movhttps://img	这个杀手不太冷 9.4	2130308	惊悚泰和导演: 吕克·贝松 Luc Besson 主演: 让·雷诺 Jean Reno / 娜塔莉·波特曼 ... 1994 / 法国 / 美国 动作 犯罪	
8	https://movhttps://img	美丽人生 9.6	2167872	最动人的导演: 罗伯托·贝尼尼 Roberto Benigni 主演: 罗伯托·贝尼尼 Roberto Beni ... 1997 / 意大利 / 剧情 喜剧 爱情 战争	
9	https://movhttps://img	千与千寻 9.4	2058287	最棒的动画导演: 宫崎骏 Hayao Miyazaki 主演: 柊瑠美 Rumi Hagi / 入野自由 Yūki ... 2001 / 日本 / 剧情 动画 奇幻	
10	https://movhttps://img	辛德勒的名单 9.6	1015262	拯救一个导演: 史蒂文·索德伯格 Steven Spielberg 主演: 连姆·尼森 Liam Neeson ... 1993 / 美国 / 剧情 历史 战争	
11	https://movhttps://img	星际迷航 Inception 9.4	1897047	诺兰导演: 克里斯托弗·诺兰 Christopher Nolan 主演: 莱昂纳多·迪卡普里奥 Leo ... 2010 / 美国 / 美国 / 科幻 悬疑 惊悚	
12	https://movhttps://img	星际穿越 Interstella 9.4	1588782	永恒是种幻觉导演: 克里斯托弗·诺兰 Christopher Nolan 主演: 马修·麦康纳 Matthew Mc ... 2014 / 美国 / 美国 / 科幻 剧情 冒险	
13	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
14	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
15	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
16	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
17	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
18	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
19	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
20	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
21	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
22	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
23	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
24	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
25	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
26	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
27	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
28	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
29	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	
30	https://movhttps://img	当哈利遇到当妮 Tru 9.3	1525558	永远都是真爱导演: 彼得·威尔 Peter Weir 主演: 休·杰克曼 Hugh Jackman / 瑞 ... 1995 / 美国 / 美国 / 爱情	

至此，我们完成了爬取豆瓣电影 TOP250 中信息的代码编写，并且可以在

Jupyter 中顺利运行。

4.2 数据可视化

数据可视化首先对数据进行一定的处理，包括统计、去重、整理等，然后以图形化的形式呈现，便于我们更好对其进行识别、判断。数据可视化过程包括数据处理、可视化、词云分析、热度分析。

（1）可视化实现

当数据处理完毕后，需要对这些数据进行可视化分析，通过图表的形式将数据中隐藏的信息展现出来。数据可视化分析可以用 Excel 和数据分析器 Pandas、。

（2）可视化呈现

本文分别采用了 matplotlib 库, pyecharts 库可直接生成图像, 其中爬取豆瓣 TOP250 后生成了词云图、各年份上映电影数量柱状图、各地区上映电影数量前十柱状图、电影评价人数前二十柱状图、豆瓣 Top 榜上电影年产量图与豆瓣 Top250 榜电影平均分图

4.2.1 导入以下的模块

```
import pandas as pd
from pyecharts import options as opts
from pyecharts.charts import Bar
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

4.2.2 用 pandas 模块读取数据

```
ata = pd.read_excel('TOP250.xlsx')
```

4.2.3 各年份上映电影数量柱状图（纵向）

```
def getzoombar(data):
    year_counts = data['上映年份'].value_counts()
    year_counts.columns = ['上映年份', '数量']
    year_counts = year_counts.sort_index()
    c = (
        Bar()
```



```

.add_xaxis(list(year_counts.index))
.add_yaxis('上映数量', year_counts.values.tolist())
.set_global_opts(
    title_opts=opts.TitleOpts(title='各年份上映电影数量'),
    yaxis_opts=opts.AxisOpts(name='上映数量'),
    xaxis_opts=opts.AxisOpts(name='上映年份'),
    datazoom_opts=[opts.DataZoomOpts(), opts.DataZoomOpts(type_='inside')],
).render('各年份上映电影数量.html')
)

```

效果：

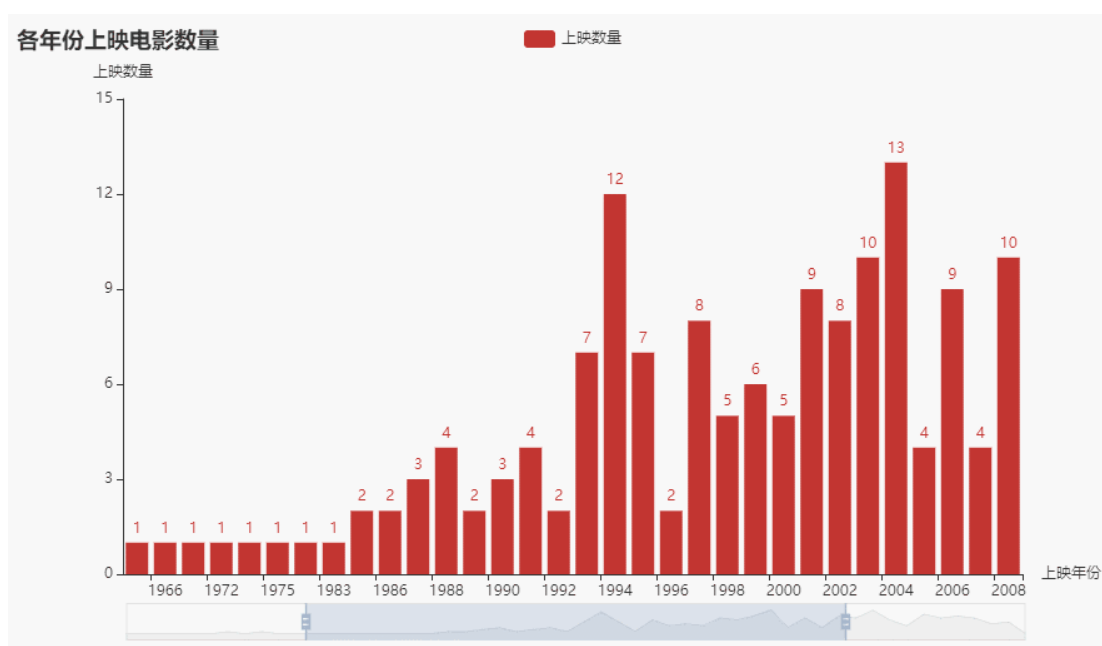


图 4.2.1 各年份上映电影数量柱状图（纵向）

从图中我们可以看到，250 部电影中 1985 年之前产量的增速比较缓慢；而在 1985 年之后，电影产量迅速提升，在 2019 年产量约达到 100 部。从图中我们可以看到，250 部电影中美国的出产量最多。

4.2.4 各地区上映电影数量前十柱状图（横向）

```

def getcountrybar(data):
    country_counts = data['国家/地区'].value_counts()
    country_counts.columns = ['国家/地区', '数量']
    country_counts = country_counts.sort_values(ascending=True)
    c = (

```

```

Bar()
.add_xaxis(list(country_counts.index)[-10:])
.add_yaxis('地区上映数量', country_counts.values.tolist()[-10:])
.reversal_axis()
.set_global_opts(
    title_opts=opts.TitleOpts(title='地区上映电影数量'),
    yaxis_opts=opts.AxisOpts(name='国家/地区'),
    xaxis_opts=opts.AxisOpts(name='上映数量'),
)
.set_series_opts(label_opts=opts.LabelOpts(position="right"))
.render('各地区上映电影数量前十.html')
)

```

效果:

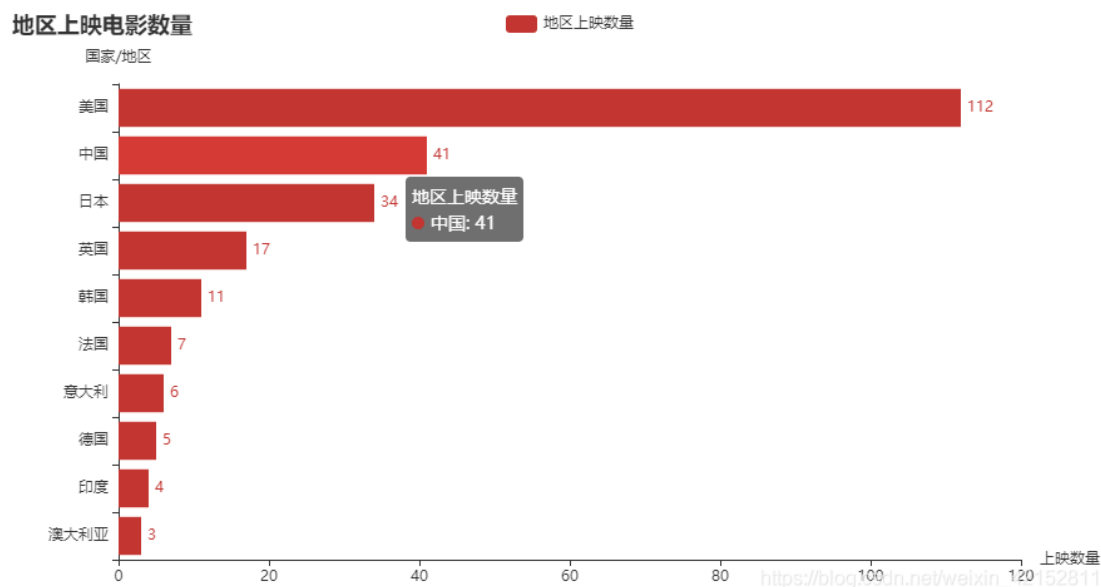


图 4.2.2 各地区上映电影数量前十柱状图（横向）

从图中我们可以看到，250 部电影中美国的出产量最多。

4.2.5 电影评价人数前二十柱状图（横向）

```

def getscorebar(data):
    df = data.sort_values(by='评价人数', ascending=True)
    c = (
        Bar()
        .add_xaxis(df['片名'].values.tolist()[-20:])
        .add_yaxis('评价人数', df['评价人数'].values.tolist()[-20:])
    )

```

```

.reversal_axis()
.set_global_opts(
    title_opts=opts.TitleOpts(title='电影评价人数'),
    yaxis_opts=opts.AxisOpts(name='片名'),
    xaxis_opts=opts.AxisOpts(name='人数'),
    datazoom_opts=opts.DataZoomOpts(type_='inside'),
)
.set_series_opts(label_opts=opts.LabelOpts(position="right"))
.render('电影评价人数前二十.html')
)

```

效果:

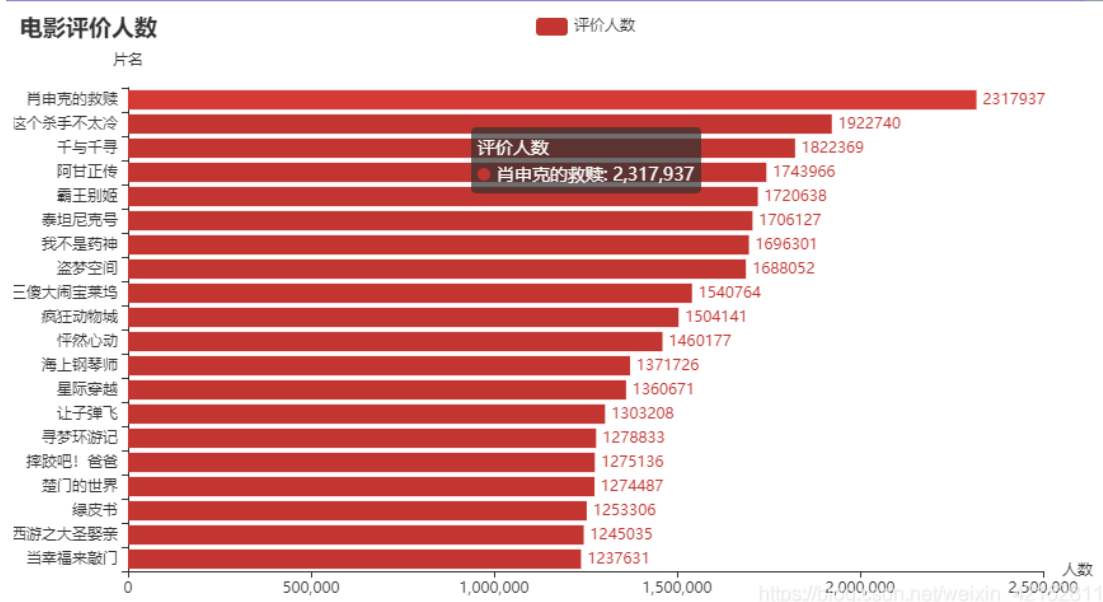


图 4.2.3 电影评价人数前二十柱状图（横向）

从图中我们可以看出，电影排名与评价人数不相关。

4.2.6 豆瓣 Top250 榜电影平均分图

```

score_mean = df.groupby('year')['score'].mean()
plt.figure(figsize = (12,6))
plt.plot(score_mean)
plt.title('豆瓣 Top250 榜电影平均分')
plt.show()

```


第 5 章 项目总结与展望

5.1 总结

通过这次实验设计，不仅仅体会到技术的世界日新月异，也在感受各种技术的出现和不断发展促进了各行各业的发展。虽然也滋生了一些并不总是积极的东西，但此消彼长，技术对抗技术也一定会值得讨论的话题。相信爬虫的编写和可塑性会越来越好，也能够更加智能地开展。同时这次设计的项目不仅仅让我感受到了编程语言，尤其是 Python 的魅力，也通过解决实际问题的方式，了解更多此前仅从书面上不曾学会的东西。Python 的优雅，易于实现以及丰富的库都给我留下了深刻的印象。尤其是开源社区和开源文档的出线，让我有理由相信未来的程序发展会越来越多的成为平常人而不只是程序员所需要具备的特权

5.2 展望

虽然本次实验设计的最终成果基本可以满足需求，但是离一开始的设想仍然有不少的距离。主要的遗憾以及希望继续改进的地方有：

（1）URL 中对应的中文随机变化的问题。由于多数网站是使用自己的加密方式，中文输入后会对应转成随机的数字英文符号代码，这给批量抓取带来了巨大的麻烦。

（2）BeautifulSoup 页面抓取格式兼用性的问题。由于 BeautifulSoup 在抓取指定内容时依靠的是 Html 中的<tag>信息，所以每一个网站需要单独对内容进行一次匹配，无法兼容其他的网站。

（3）尝试其他方式占用了太多的时间。在实际开发中，我还尝试了使用标准的抓取框架 Scrapy 以及正则表达式匹配方式达成抓取的目的，但都失败了，也让我明白解决问题比理论最优解更值得投入时间。本次实验项目设计编写的爬虫程序相对而言维度仍比较单一，支持的格式和网站也有限，希望在后续的优化中可以更好的处理各种情况。