# 项目总结报告

豆瓣电影数据可视化

小组名称： 第五组

学员姓名： 郑钰淇

目录

[项目总结报告 1](#_Toc17289904)

[1. 项目基本信息 2](#_Toc17289905)

[2. 项目成果 2](#_Toc17289906)

[3. 开发工具与环境 2](#_Toc17289907)

[4. 项目需求变更 3](#_Toc17289908)

[5. 项目计划和进度实施 3](#_Toc17289909)

[6. 经验总结 3](#_Toc17289910)

[7. 教训总结 4](#_Toc17289911)

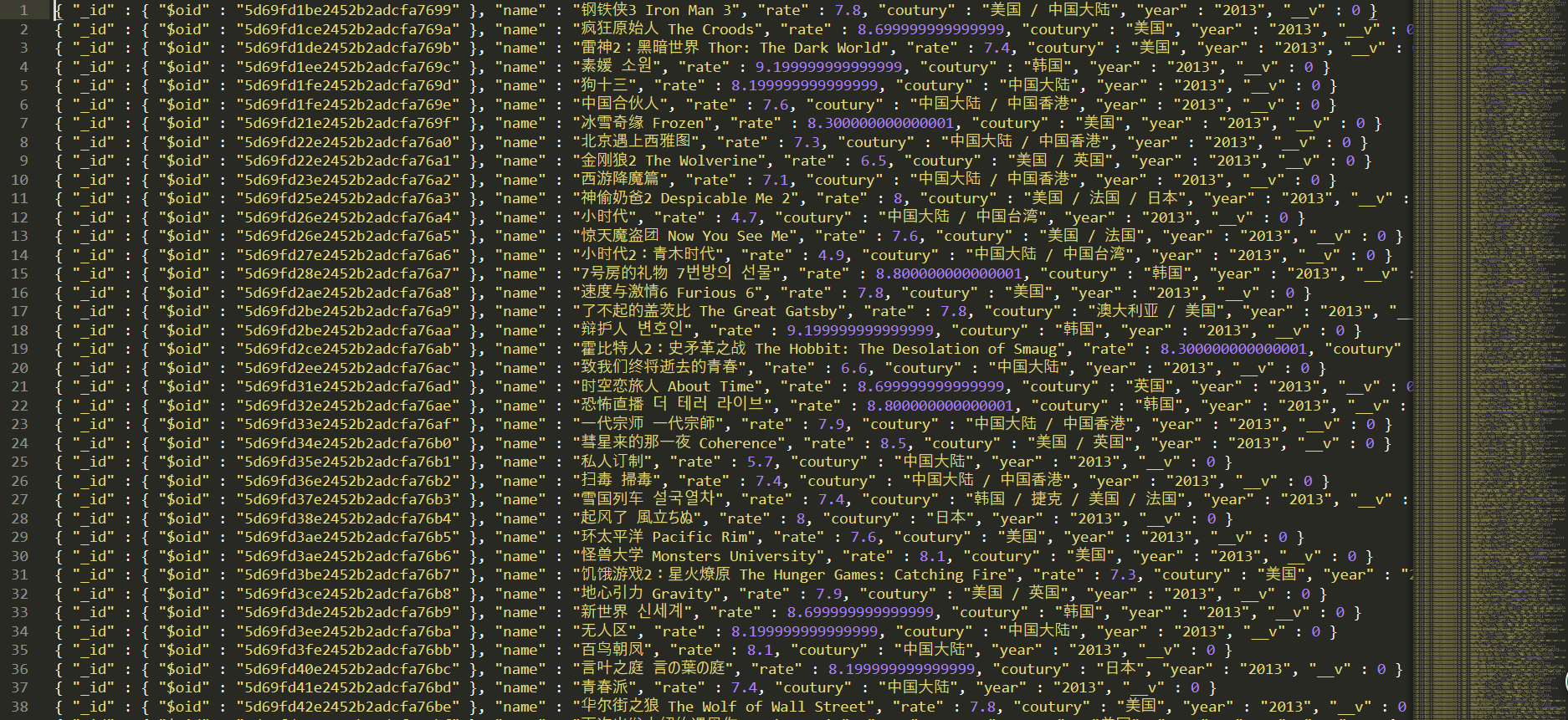
[8. 其他改进意见 4](#_Toc17289912)

# 项目基本信息

基于Node.js从豆瓣网上爬取2009-2019的电影数据，筛选清洗后将电影名，评分，上映国家和上映日期存入mongo数据库，通过echarts渲染展示出影评分和电影数量分析、近十年来每年发行的电影分析、2019各个国家发行的电影数量、评分在5分一下的电影名称等数据可视化的展示，最终实现豆瓣电影数据可视化分析平台的搭建。

# 项目成果

爬取2013-2014两年的电影所需数据



# 开发工具与环境

Node.js：前端应用开发方法

Cheerio：实现HTML数据解析

MongoDB：对需要的数据进行存储

eCharts：数据可视化展示

superagent：客户端请求代理模块（支持HTTPS）

# 项目需求变更

数据变更：

爬取名字、评分、国家、年份、评论人数

第一次变更为 爬取名字、评分、国家、年份、导演

第二次变更为 爬取名字、评分、国家、年份

变更原因：电影子页面没有评论人数、图表中没有需要导演数据的内容

导致的结果和影响：修改爬虫代码cheerio部分以及数据库存储部分

数据库变更：

存储所有电影

第一次变更为 存储有电影名称，评分，国家和上映日期的电影

# 项目计划和进度实施

一共十年数据，每个人爬两年数据，剩余一人完成数据可视化部分内容

# 经验总结

项目初期的分工进行有利于项目的总体实现

豆瓣网设有反爬虫机制，为了降低ip被封的问题，首先使用代理，在请求头携带Cookie等信息来模拟浏览器访问，另外设置较大请求间隔，不触发封ip的机制。

# 教训总结

计划赶不上变化，想法过于理想化，从而项目开展不是很顺利。程序不是一蹴而就的，要经过反复调试和完善。

# 其他改进意见

(对项目的开发过程有其他可以优化和改进的地方)