

高考志愿查询预测与推荐系统

项目开发总结报告

项目名称：高考志愿查询预测与推荐系统

项目组：高考志愿查询预测与推荐系统开发组

项目成员：陈思翰（项目经理）、李响、薛安宇、胡周峰、田如召

2021 年 6 月 2 日

目 录

1. 引言.....	1
1.1 文档编写目的.....	1
1.2 项目开发背景.....	1
1.3 相关定义.....	1
1.4 参考资料.....	2
2. 实际开发结果.....	2
2.1 软件产品描述.....	2
2.1.1 版本标识.....	2
2.1.2 同上一次发行版本的区别.....	2
2.1.3 软件模块清单.....	2
2.1.4 文档清单.....	2
2.1.5 已知缺陷.....	3
2.2 主要功能和性能.....	3
2.2.1 高校信息展示.....	3
2.2.2 分数线查询.....	3
2.2.3 专业推荐.....	3
2.2.4 模拟填报志愿.....	3
2.3 开发进度.....	4
2.4 开发成员.....	4
2.5 开发成本.....	4
2.5.1 工时.....	4
2.5.2 经费.....	4
3. 开发工作评价.....	5
3.1 项目开发过程评价.....	5
3.2 技术方法评价.....	5
3.3 产品质量评价.....	5
3.3.1 开发过程.....	5
3.3.2 开发技术.....	5
3.3.3 成员素质.....	5
3.3.4 时间进度.....	5
3.3.5 成本.....	5
3.3.6 总体产品质量.....	6
4. 测试工作评价.....	6
4.1 检查数据完整度.....	6
4.1.1 发现的问题.....	6
4.1.2 解决情况.....	6
4.2 检查数据清洗结果和数据库存储情况.....	6
4.2.1 发现的问题.....	6
4.2.2 解决情况.....	6

4.3	检查前后端数据传输状况	6
4.3.1	发现的问题	6
4.3.2	解决情况.....	6
4.4	检查机器学习算法效果和数据库存储情况	7
4.4.1	发现的问题	7
4.4.2	解决情况.....	7
4.5	检查所有功能的运行情况	7
4.5.1	发现的问题	7
4.5.2	解决情况.....	7
4.6	维护工作评价.....	7
5.	技术积累总结.....	7
6.	经验与教训	8

文档信息	
标题	项目开发总结报告 Project Development Summary Report
作者	陈思翰
创建日期	2021.05.24
更新日期	2021.06.02
版本	V1.6
部门名称	高考志愿查询预测与推荐系统项目开发组-华中农业大学实训第五组

文档更新记录			
版本号	更新内容	操作日期	操作人员
1.0	初步拟定文档架构，填入相关内容	2021.05.24	陈思翰
1.1	补充 4.1 节 检查数据完整度	2021.05.26	陈思翰
1.2	补充 4.2 节 检查数据清洗结果和数据库存储情况 更新第 3 部分 开发工作评价	2021.05.27	陈思翰
1.3	补充 2.1.2 节 同上一次发行版本的区别 更新 2.3 节 开发进度	2021.05.28	陈思翰
1.4	更新第 3 部分 开发工作评价 补充 4.3 节 检查前后端数据传输状况	2021.05.31	陈思翰
1.5	更新 2.3 节 开发进度 补充 4.4 节 检查机器学习算法效果和数据库存储情况 补充 4.5 节 检查所有功能的运行情况 补充 4.6 节 维护工作评价	2021.05.31	陈思翰
1.6	更新第 3 部分 开发工作评价 补充第 5 部分 技术积累总结 补充第 6 部分 经验与教训	2021.06.01	陈思翰

项目开发总结报告

Project Development Summary Report

1. 引言

1.1 文档编写目的

制定本文档的目的,是为了给本次项目开发经历作总结。本文档涵盖了项目概况介绍、项目开发历程、项目测试历程和项目实际开发结果。旨在有逻辑地、合理地思考并总结此次项目开发经历的所学所用、技术积累和经验教训,并做出自我评价。

1.2 项目开发背景

高考一直以来都是我国传统应试教育中不可或缺的一部分。这场考试是对每个学生 12 年学习生涯的一次检验,是最后一次,也是最重要的一次测试。因此,无论是学生还是家长都对高考有着极高的关注度。本次开发的软件旨在基于当前网络中各式各样的高考数据网站,对各高校相关信息及其录取分数线等各项数据进行汇总,并将相关功能精简化,同时引入机器学习相关算法对数据进行预测和分类。达成以上目标后,用户就能轻便地达到数据查询的目标,并在合适的范围内参考本系统预测和推荐的结果。

项目名称	高考志愿查询预测与推荐系统
项目委托方	华迪公司
用户	华迪公司
开发者	高考志愿查询预测与推荐系统开发组
交付的产品	高考志愿查询预测与推荐系统前端网站 V1.1

1.3 相关定义

①PyCharm: 是一种 Python IDE,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;

②requests: 用Python语言编写,基于 urllib,采用 Apache2 Licensed 开源协议的 HTTP 库;

③selenium: 是一个用于 Web 应用程序测试的工具,可以模拟用户操作浏览器;

④sklearn: sklearn 是一个 Python 第三方提供的非常强力的机器学习库,它包含了从数据预处理到训练模型的各个方面;

⑤MongoDB: 一个基于分布式文件存储的数据库,由 C++语言编写。旨在为 WEB 应用提供可扩展性的高性能数据存储解决方案;

⑥Apache Spark: 是专为大规模数据处理而设计的快速通用的计算引擎。

1.4 参考资料

无。

2. 实际开发结果

2.1 软件产品描述

2.1.1 版本标识

版本号：V1.1

2.1.2 同上一次发行版本的区别

更新整个前端界面，使其整体使用感更直观、更美观。

2.1.3 软件模块清单

序号	模块名称	子系统名称	代码量大小
1	数据获取		约 500 行
2	数据清洗		约 100 行
3	机器学习		约 100 行
4	数据存储	高考志愿查询系统 高考分数线预测系统 高考志愿推荐系统	约 500 行
5	前端界面		约 1300 行
代码总量		2500 行以上	

2.1.4 文档清单

序号	文档名称	须提交用户
1	《0. 项目需求分析文档》	华迪公司
2	《1. 用例实现规约》	华迪公司
3	《2. 项目开发计划文档》	华迪公司
4	《3. 系统架构设计文档》	华迪公司
5	《4. 原型设计文档》	华迪公司
6	《5. 数据库设计文档》	华迪公司
7	《6. 测试计划文档》	华迪公司

8	《7. 测试计划文档》	华迪公司
9	《8. 测试报告》	华迪公司
10	《9. 项目开发总结报告》	华迪公司

2.1.5 已知缺陷

暂无

2.2 主要功能和性能

2.2.1 高校信息展示

用户可以在页面中查看中国各地区开设的高校数量（精确到市级行政区），并能够查询到各高校的具体信息，包括高校名称、地址、建校时间、占地面积、官网网址、联系电话等。

该功能已实现。

2.2.2 分数线查询

用户能够根据自己所在省份、高校类型、高校名称、专业名称等关键字查询到历年来该省的批次线、以及该专业在该省录取的最低分数线和最低排名。同时系统也会展示出当年批次线和分数线的预测结果供用户参考。

该功能已实现。

2.2.3 专业推荐

用户可以在页面中查看中国所有高校开设的所有专业的相关信息，包括专业门类、专业大类、专业层次、专业简介、就业情况等。同时系统也能根据专业推荐合适的高校。

该功能已实现。

2.2.4 模拟填报志愿

用户需填入自己所在省份、文理科和高考分数。系统会根据用户填入的信息预测该省当年的高考批次线，并向用户推荐 30 所高校及每所高校对应的若干专业作为高考志愿。其中 10 所高校为“录取难度较大，可以尝试冲击”、另 10 所高校为“录取难度不大，基本可以录取”、最后 10 所高校为“没有录取难度，可以作为保底”。

该功能未完全实现。

2.3 开发进度

阶段	计划工作日	实际工作日	对比	原因
项目启动和立项	3	3	按时完成	无
各功能应开发完成，并完成整合	8	10	延迟 2 天	前端开发遇见难题；机器学习算法难以下手
项目测试，对存在 bug 和不足的地方进行更新	9	10	延迟 1 天	上一阶段延迟
更新完成，进行二轮测试	4	尚未结束	尚未结束	尚未结束
准备答辩工作，相关文档需编写完毕	2	尚未开始	尚未开始	尚未开始

2.4 开发成员

序号	角色	成员姓名	职责
1	项目经理、数据分析师	陈思翰	负责整个项目中大部分文档的撰写和核查，协调各个成员之间的工作。同时负责采用机器学习相关算法进行数据预测。
2	爬虫工程师	李响	负责整个项目的文档中，涉及爬虫部分的编写。负责编写爬虫程序获取数据。
3	数据库工程师	田如召	负责整个项目的文档中，涉及数据清洗和数据库部分的编写。负责进行数据清洗和数据存储。负责建立和维护数据库。
4	前端工程师	薛安宇	负责整个项目的文档中，涉及前后端接口部分的编写。负责建立前后端相关接口，将项目的各个部分结合起来。
5	前端工程师	胡周峰	负责整个项目的文档中，涉及数据可视化和前端架构部分的编写。负责前端部分的架构，并实现数据可视化功能。

2.5 开发成本

2.5.1 工时

原定计划：15 个工作日，每个工作日最低标准为 8 个工时。
实际成本：15 个工作日，每个工作日 12 个工时。

2.5.2 经费

原定计划：无。
实际成本：无

3. 开发工作评价

3.1 项目开发过程评价

在项目开发前期：

①前端架构阶段中，前端页面视觉效果差，功能少，前后端接口不能正常工作。原因在于没有前端开发经验，需要加快学习进度，并多多实践；

②数据获取阶段中，数据不满足要求。原因在于前期需求分析不明确，且采用的 `python` 模块不对，需要尽快确定所需要的数据，并更改相关代码；

③数据预测阶段中，机器学习算法不好切入。原因是数据项少，数据量大，需要寻找更加契合本次项目数据的机器学习算法。

其余阶段没有产生问题。

以上问题都在项目开发后期得到解决，本次项目开发完成情况良好。

3.2 技术方法评价

除了开发前后端接口采取的技术方法不合理之外，其他技术方法都能达到项目开发要求。

5月31日：现已采用 `Flask` 框架开发前后端接口，效果良好。

3.3 产品质量评价

3.3.1 开发过程

开发过程总体顺利，前端部分出现的问题较多，开发过程较为困难。目前已经解决大多数问题。

3.3.2 开发技术

开发采取的技术大部分合理，但前后端接口难以实现（已于5月31日解决）。

3.3.3 成员素质

开发成员素质良好，除了客观因素影响以外，都可以按时完成任务。

3.3.4 时间进度

时间进度略微拖沓，原因是前端开发受限，目前已经追上进度。

3.3.5 成本

时间成本分配合理，没有经费需求。

3.3.6 总体产品质量

产品质量良好（截至 6 月 2 日已完成三个功能，高考志愿推荐系统正在开发中）。

4. 测试工作评价

4.1 检查数据完整度

4.1.1 发现的问题

大部分数据存在缺失，部分数据项未能获取，部分数据被网页加密，无法获取。

4.1.2 解决情况

采取措施：更改爬虫代码，延长网页访问时间，增加 `try.....except` 语句抛出异常，采用 `selenium` 模块代替 `requests` 模块编写爬虫。

解决情况：顺利解决。

4.2 检查数据清洗结果和数据库存储情况

4.2.1 发现的问题

清洗后的数据格式错误，缺失值填充后依然无法使用，数据编码不统一。

4.2.2 解决情况

采取措施：重写代码，更改缺失值填充方式，统一数据编码。

解决情况：顺利解决

4.3 检查前后端数据传输状况

4.3.1 发现的问题

数据未能在前端界面上展示，前后端数据无法实现动态交互。

4.3.2 解决情况

修改源码，改用 `Flask` 框架开发前后端接口。

解决情况：顺利解决。

4.4 检查机器学习算法效果和数据库存储情况

4.4.1 发现的问题

无

4.4.2 解决情况

无

4.5 检查所有功能的运行情况

4.5.1 发现的问题

测试尚未开始

4.5.2 解决情况

无

4.6 维护工作评价

维护工作组织及时且到位，能够快速高效地实施，并能有效解决问题。

5. 技术积累总结

①数据获取阶段：加强了对 Python 的 requests 模块和 selenium 模块的理解和运用。今后编写爬虫遇见数据加密问题时，可以通过此方法解决。

②数据清洗阶段：加强了 Spark 的运用技巧，并可以在 Spark 和 MongoDB 之间灵活切换。

③数据存储阶段：加强了 MongoDB 的运用技巧，且能通过 Python 使用 MongoDB，加深了对 pymongo 模块的理解。

④机器学习阶段：加强了对 Python 的 sklearn 模块的理解和运用。已经掌握线性回归分析和聚类分析的相关技巧，今后可以用这些方法分析其他数据。

⑤前端开发阶段：学会使用 Flask 框架整合各组件。学会使用 JavaScript 和 H5C3 开发前端界面。

6. 经验与教训

首先，本次项目开发过程中，各成员都不同程度的加深了对大数据综合项目的理解，也具有了一定程度的开发能力，并因此对爬虫、Spark、MongoDB、机器学习有了更加深入的理解。同时，本次项目开发经历让各成员首次接触前端架构及其相关内容，因此有很广阔的学习空间。

其次，由于此次项目开发为团队项目，各成员也体验到了团队协作的魅力，并逐步摸索到团队开发的工作素养和工作要领。唯一的一点遗憾是，由于立项时对相关领域不够熟悉，导致人员分配不合理。前端部分分配人员过少，前端工程师工作压力大；而数据获取、分析、清洗部分分配人员过多，出现部分成员无事可做的状态。这种情况明显影响到了项目开发的初期阶段，并拖慢了开发进度，好在后期及时更改了人员分配情况，追回前期丢失的进度。

最后，本次项目开发相较之前更加正规，需要撰写相当数量的文档，这对各成员的写作能力而言是一次学习和考验。另外本次采取线下答辩，更加需要各成员拥有纯熟的语言能力和随机应变能力。