

《近三年（2023-2025）北京市房价变化分析》

项目计划书

团队成员：苏崇博，王俊涛，侯钧译，王梓玥，郝竟妍

2025 年 11 月 14 日

目录

1 项目概述	3
1.1 项目背景	3
1.2 项目目标	3
1.3 项目范围	3
1.4 预期成果	4
2 技术架构与路线	4
2.1 技术栈清单	4
2.2 数据处理与交互方案	4
3 团队成员及分工	5
4 项目进度规划	5

1 项目概述

1.1 项目背景

近年来，中国房地产市场经历了深刻的变革和调整。房价不仅是衡量宏观经济的重要指标，也直接关系到民生福祉。本项目旨在利用 Python 数据分析技术，对北京市 2023 年至 2025 年间的房价数据进行系统性分析，并通过一个交互式的 Web 前端界面进行成果展示。

1.2 项目目标

- **目标一 (数据获取与处理):** 从主流房地产数据网站（如【链家、安居客等】）获取北京市近三年的二手房月度平均价格数据，并进行数据清洗、整合，构建一个干净、规范的数据集。
- **目标二 (多维度分析):** 对数据进行多维度分析，包括时间序列分析、各行政区横向对比等，挖掘数据背后的核心规律。
- **目标三 (可视化呈现):** 利用数据可视化技术，将分析结果以直观、清晰的图表形式呈现。
- **目标四 (Web 前端开发):** 开发一个简洁、友好的 Web 前端应用，以交互式图表的形式动态展示数据分析结果，提升用户体验。
- **目标五 (机器学习回归与特征解释):** 在清洗后的数据基础上，引入随机森林回归（Random Forest）与 XGBoost 回归模型，对房价走势进行建模，并输出特征重要性，分析影响房价变化的关键驱动因素。并交叉验证合理性。

1.3 项目范围

- **分析城市:** 北京市
- **时间范围:** 2023 年 1 月 至 2025 年 11 月
- **数据范围:** 重点分析二手房挂牌均价数据，数据采集粒度为月度。
- **特征选择:** 本项目在获得月度均价数据后，将进一步构造多维度特征，包括：
 - **区域特征:** 北京市各行政区（如海淀、朝阳、丰台、东城等）采用 One-hot 编码；同时加入环线位置（1 – 6 环）的序号型特征，用于刻画城市空间差异。
 - **时间类特征:** 月份、季度、是否为传统交易旺季（如 3 – 5 月、9 – 11 月）等采用哑变量处理，以捕捉季节性变化。
 - **趋势类特征:** 包括房价环比（Month-over-Month）涨跌幅、同比（Year-over-Year）涨跌幅，以及过去数月的移动平均值，用于刻画时间序列趋势。
 - **政策与宏观特征:** 在关键政策节点（如限购、限售政策调整）加入政策哑变量，以反映政策冲击。
 - **通勤与地段特征:** 地铁站等通勤距离，以及是否属于主城区等使用 0 – 1 变量编码，用于区分快速增长区与刚需集中区。

1.4 预期成果

1. 可运行的后端 Python 源代码：包含数据爬取、数据清洗和数据分析的完整代码。
2. 可运行的前端项目代码：包含 HTML, CSS, JavaScript 的 Web 前端项目文件。
3. 清洗后的数据集 (JSON/CSV)：用于前后端交互的、格式干净的数据文件。
4. 一个可访问的 Web 应用：部署后的网页链接或本地运行指南。
5. 最终分析报告 (PDF)：一份详细阐述项目全过程、核心发现及 Web 应用介绍的综合报告。
6. 机器学习建模结果：包含随机森林与 XGBoost 回归模型的训练结果、性能指标（如 RMSE）以及特征重要性图表，用于补充房价变化的结构化解释。

2 技术架构与路线

2.1 技术栈清单

- 后端技术 (数据处理)
 - 编程语言: Python 3.9+
 - 数据获取: requests, BeautifulSoup4 / Scrapy
 - 数据处理与分析: Pandas, NumPy
 - 回归处理与预测: scikit-learn, XGBoost
- 前端技术 (数据展示)
 - 核心技术: HTML5, CSS3, JavaScript (ES6+)
 - 可视化库: ECharts.js (推荐) 或 D3.js
 - 前端框架: Vue.js 或 React.js
- 通用工具
 - 版本控制: Git
 - 开发环境: VS Code, Jupyter Notebook, PyCharm

2.2 数据处理与交互方案

本项目采用前后端分离的开发模式。

1. 后端：Python 脚本负责完成数据的获取、清洗和分析。
2. 数据导出：后端分析完成后，将前端可视化所需的数据处理成简洁的 JSON 格式文件。
3. 前端：前端 Web 页面通过 JavaScript 异步加载这些 JSON 数据文件，然后调用 ECharts.js 库将数据渲染成交互式图表。这种方式无需搭建复杂的后端服务器，简化了项目部署。

3 团队成员及分工

本项目的分工采用以**模块负责人**为主，团队成员协同参与的模式，鼓励跨模块协作和学习。

主要负责人	职责与协作方式
苏崇博	负责整体进度规划，组织团队会议，整体架构的修改，数据处理，核心代码实现，前端优化，管理 Git 仓库。确保项目方向正确、沟通顺畅。
王俊涛	主要负责： 前端整体架构、核心可视化图表的开发与数据接入。 协同参与： 数据清洗和预处理，爬虫核心代码实现，撰写最终的分析报告、项目文档及部署说明。
侯钧译	主要负责： 数据清洗、预处理和核心分析逻辑实现，爬虫框架设计与核心代码实现。 协同参与： 前端优化，撰写最终的分析报告、项目文档及部署说明。
王梓玥	主要负责： 数据清洗、预处理和核心分析逻辑实现。 协同参与： 前端优化，爬虫核心代码实现，撰写最终的分析报告、项目文档及部署说明。
郝竞妍	主要负责： 爬虫框架设计与核心代码实现，整合所有模块的成果，撰写最终的分析报告。 协同参与： 同时负责前端整体架构，数据清洗和预处理。

表 1: 团队分工表

注：每位成员在完成自己主要负责的模块任务外，会积极参与其他模块的讨论、代码审查 (*Code Review*) 和测试工作，共同对项目最终质量负责。

4 项目进度规划

项目进度围绕核心产出物进行规划，鼓励前后端并行开发和快速迭代。

阶段	教学周次	核心目标	主要交付成果
1. 启动与设计	第 1-2 周	完成项目规划, 搭建开发环境, 定义前后端数据接口标准。	项目计划书, Git 仓库, 前后端环境配置文档, 数据接口定义文档 (JSON 格式草案)。
2. 后端开发	第 3-5 周	完成数据的获取、清洗与分析, 并能生成标准格式的 JSON 数据。	可运行的数据处理全流程 Python 脚本, 清洗后的数据集 (CSV), 符合接口定义的 JSON 数据文件。
3. 前端开发	第 3-6 周	完成页面 UI 设计与主体框架, 使用模拟数据或后端产出的初步数据开发核心可视化图表。	Web 页面静态原型, 使用 ECharts 实现的核心交互图表组件。
4. 集成与测试	第 7 周	将后端生成的真实数据接入前端, 进行全面的功能测试和 UI 优化。	功能完整、数据驱动的 Web 应用初版。
5. 报告与完善	第 8-9 周	撰写最终分析报告, 完善项目文档, 进行项目部署并准备最终答辩。	最终分析报告 (PDF), 完整的项目代码库, Web 应用部署链接或指南。

表 2: 项目进度规划表