# 房价估值大数据平台

**作者**：李坤朋，宁浩鉴，钱致远，池培德，冯永

**单位**：重庆大学，计算机学院

**案例版权**：该案例归重庆大学计算机学院所有

**涉及的知识点**：Python数据挖掘、数据预处理、决策树、数据可视化

**案例来源及案例真实性情况**：该案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。

**摘要** 人们在购买房屋时，对房价的预估是极其重要的一环。但是传统数据收集及分析方法效率低、不够全面等，个人难以准确地对市场进行定位和房屋估值。随着大数据时代的来临，数据挖掘和信息采集技术不仅能给研究人员提供足够的样本量和数据信息，还能够建立基于大数据数学模型对未来市场进行预测以及对房屋进行精确估值。基于以上，本项目是一个房屋数据分析与房屋估值平台，采集了全国所有省市的房价和相关数据并进行清洗，设计了对应房屋估值算法，为用户提供房屋估值功能。为提高预测准确性，本项目选取了小区、二手房、成交房的相关数据进行可视化，旨在采用大数据辅助房屋估值分析，精准地考虑面积、装修、建造时间、是否有电梯、小区均价等各方面因素。在具体实现上，我们首先通过爬虫爬取链家等网站的大规模交易数据，然后通过决策树等算法构建机器学习模型，最后对输入交易数据进行训练和预测。特别的，本案例还通过多种大数据可视化策略对模型性能进行了直观展示和分析。

**关键词**：房价预测，数据挖掘，机器学习，数据可视化

## **1 引言**

该教学案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。该案例的关键问题为房价预测，以根据房屋的各方面属性，如面积、住址、装修等，为考量提供房屋价格的预测估值功能为主要目的。基于以上，需引导学生进行的主要内容有：（1）利用Python相关工具获取各地区房屋的各项属性及价格等数据；（2）对数据进行筛选、清洗等数据预处理操作；（3）选取适当的机器学习方法对房屋价格进行建模并预测；（4）分析数据结果并进行数据可视化。

## **2 背景介绍**

房产，已是当今社会难以避开的一个话题。而房产的价格与众多因素存在着千丝万缕的联系，数据挖掘在这个领域中发挥着重要的作用。无论是针对购房者的住房需求、销售者的营销策略、生产者的市场定位，深层次的数据挖掘都会给出更合适的指导方向。在传统情况下，分析数据的收集主要来自于行业管理部门数据、相关市场调查、相关行业报告、行业专家意见和统计年鉴等。这些数据大多存在样本量不足，时间滞后和准确度低等缺点，研究人员能够获得的信息量十分有限，这使得准确的市场定位和房屋估值存在着一定瓶颈。随着大数据时代的来临，数据挖掘和信息采集技术不仅能给研究人员提供足够的样本量和数据信息，还能够建立基于大数据数学模型对未来市场进行预测以及对房屋进行精确估值。

本案例首先使用Python爬虫爬取链家网的房屋价格及其相关属性，再对获取的数据进行预处理操作，以使得数据更易于分析处理。之后，本案例应用机器学习中的决策树算法对处理后的数据建模，随后预测不同房屋的房价，最后使用相关的可视化工具将处理结果直观地展示给用户。

本案例是一个房屋数据分析与房屋估值平台，使用了有关数据挖掘、机器学习算法以及数据可视化等方法，与社会生活高度契合，具有相当的实际意义，是均衡培养各方面人才的有效案例。

## **3 内容**

该案例的主要内容主要分为三个小节，分别为数据集获取与预处理、房屋价格预测，以及数据可视化。

3.1 数据集获取与预处理

本案例通过对链家的web页面进行分析，运用requests，beautifulsoup等库和http代理等技术，实现不间断地从链家网页采集信息，保存为json格式。导出到外部数据库时，需先进行数据清洗，转化为数据库能够存储的格式。从外部数据库导入数据时无需进行清洗，直接解析即可。爬取的信息主要包括了小区、二手房和成交房相关信息，爬取的具体字段及含义如表4.1所示：

表3.1 爬取字段说明

|  |  |
| --- | --- |
| **属性名** | **说明** |
| **areaUrl** | areaUrl\_1/2/3是二手房所属地区的链家web网页链接 |
| **selfURL** | 二手房的链家web网页链接 |
| **Title** | 二手房挂牌的标题 |
| **TotalPrice** | 总价（万元） |
| **UnitPrice** | 单价（元/m²） |
| **Room\_mainInfo** | 户型（几室几厅的） |
| **Room\_subInfo** | 楼层 |
| **Type\_mainInfo** | 房屋朝向 |
| **Type\_subInfo** | 是否平层/跃层 |
| **Area\_mainInfo** | 房屋面积 |
| **Area\_subInfo** | 房屋建成信息 |
| **resblockPosition** | 经纬度（用于在地图上显示） |
| **communityName** | 小区名 |
| **AreaName** | 社区街道名 |
| **baseInfo** | 额外信息（有数项，不固定） |
| **Feature** | 房屋卖点描述（周边配套，交通出行等） |

在获取数据后，本案例使用pyspark进行数据清洗。数据清洗使用到了spark API：主要是rdd和dataframe。使用lambda表达式对每一项进行解析，函数transformXiaoqu的作用就是解析并返回解析后的数据，主要针对确实项、重复项等数据进行清洗。

对于数据的保存格式，数据按照城市进行归类存放，并进一步按照省份进行归类存放。如果想要查找某城市的数据，比如查找四川成都的数据，那么需要依次进入数据集文件夹下的“四川”，“成都”，“二手房”文件夹，找到其中以entries开头的json文件。形象地表示如下：

------------------------------

|------------四川--------

|-----------成都----------

|--------二手房--------

|-------entries-\*\*.json

对于存储在数据库的数据字段如表3.2、表3.3、表3.4所示。

表3.2 小区数据字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 数据长度 | 说明 | 主键 |
| XiaoquID | Text | 200 | 小区ID | 是 |
| Name | Text | 200 | 小区名字 | 否 |
| Location | Text | 200 | 小区位置描述 | 否 |
| Longtitude | Float | 20 | 经度 | 否 |
| Latitude | Float | 20 | 纬度 | 否 |
| AvgUnitPrice | Float | 20 | 平均每平方米价格 | 否 |
| AreaID | Text | 200 | 所属地区ID | 否，外键 |
| BuildTime | Text | 200 | 建造时间 | 否 |
| BuildType | Text | 200 | 建筑类型 | 否 |
| ManagementFee | Float | 20 | 物业费用 | 否 |
| ManagementCompany | Text | 200 | 物业公司 | 否 |
| BuildConmapy | Text | 200 | 开发商 | 否 |
| BuildingCount | Int | 20 | 楼栋总数 | 否 |
| HouseCount | Int | 20 | 房屋总数 | 否 |
| NearShops | Text | 200 | 附近门店 | 否 |

表3.3 二手房数据字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 数据长度 | 说明 | 主键 |
| HouseID | Text | 200 | 二手房ID | 是 |
| Title | Text | 200 | 房源信息 | 否 |
| TotalPrice | Int | 20 | 总价 | 否 |
| UnitPrice | Int | 20 | 平方价格 | 否 |
| HouseType | Text | 20 | 房屋户型 | 否 |
| HouseFloor | Char | 30 | 所在楼层 | 否 |
| HouseDirection | Char | 20 | 房屋朝向 | 否 |
| HouseFitment | Char | 30 | 装修情况 | 否 |
| HouseAreaValue | Float | 30 | 建造面积 | 否 |
| HouseBuildingType | Char | 20 | 建筑类型 | 否 |
| XiaoquID | Text | 200 | 小区名称 | 否，外键 |
| AreaInfo | Char | 40 | 区域信息 | 否 |

表3.4 成交房数据字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 数据长度 | 说明 | 主键 |
| TransHouseID | Int | 10 | 成交房ID | 是 |
| HouseTransCycle | Int | 10 | 房屋成交周期 | 否 |
| ClaimedPrice | Int | 10 | 挂牌价 | 否 |
| DonePrice | Int | 10 | 成交价 | 否 |
| AreaHouseNum | Int | 10 | 地区在售房源数目 | 否 |
| HouseUsage | Char | 64 | 房屋用途 | 否 |
| HouseLayout | Char | 64 | 房屋户型 | 否 |
| HouseOrientation | Char | 64 | 房屋朝向 | 否 |
| BuildTime | Char | 64 | 建造时间 | 否 |
| FloorRatio | Float | 10 | 楼层占比 | 否 |
| PropertyInfo | Char | 128 | 房源信息 | 否 |
| TotalPriceInterval | Int | 10 | 总价区间 | 否 |
| IsNearSubway | Int | 1 | 是否近地铁 | 否 |
| HouseArea | Int | 10 | 房子面积 | 否 |
| HouseLabel | Char | 64 | 房源标签 | 否 |
| XiaoquID | Text | 200 | 小区ID | 否，外键 |

最后对获取的数据进行预处理。其大体的思路如下：把房价离散化，因为决策树的输出结果都是离散值，连续值不能作为决策树的分类结果。定义房屋的“价格指数”=房屋单价/城市平均房屋单价。从“价格指数”=0.5开始，以0.01为步长打标签，到“价格指数”=1.5为止。一共得到100种标签，这些标签是取值为[0,99]的整数。通过这种方式，把连续的房价变为了离散值，将“房屋估值”这个回归问题近似地转换为了分类问题。

我们对数据集进一步处理，对每个房屋提取1个标签和6种属性，如表3.5所示。

表3.5 提取的标签和属性

|  |  |  |
| --- | --- | --- |
| **属性名** | **意义** | **取值范围** |
| 标签 | 根据房屋“价格指数”打的标签 | [0,99]，整数 |
| 面积 | 房屋面积 | 浮点数 |
| 是否有电梯 | 是否有电梯 | 0或1 |
| 是否近地铁 | 是否近地铁 | 0或1 |
| 建造时间 | 建造时间 | 整数 |
| 装修 | 装修的程度 | [0,4]，整数，分别代表“无信息”，“其它”，“毛坯”，“简装”，“精装” |
| 小区均价 | 小区内所有房屋的平均“价格指数” | 浮点数 |

3.2 房屋价格预测

本案例选择机器学习中的决策树算法来预测房价。其算法流程是，根据输入的标签和属性集，寻找一个最优划分属性，然后根据这个最优划分属性，将数据集划分为左子集和右子集。由左子集构建左子树，由右子集构建右子树，然后对左右子树进行递归的建立决策树的过程，直到达成递归结束条件。递归结束条件是输入数据集的标签全部都为同一类，或者输入数据集的数量小于阈值。

运行算法进行测试，输入测试数据各项信息，输出预测总价以及对应误差。例如，测试案例如图3.1所示，单个误差为4.1%。

文本

描述已自动生成

图3.1 测试案例

为了测试算法的准确性，在“北京”，“上海”，“成都”，“重庆”四座城市的数据上的评估结果，如表5.2所示，算法的平均误差在10%左右，符合预期目标。

表3.6 算法评估结果

|  |  |  |  |
| --- | --- | --- | --- |
| **城市** | **训练集数量** | **测试集数量** | **平均偏差** |
| 北京 | 35590 | 35590 | 9.54% |
| 上海 | 22455 | 22455 | 7.32% |
| 成都 | 52058 | 52058 | 12.16% |
| 重庆 | 57024 | 57023 | 10.61% |

3.3 数据可视化

本案例基于Python的Flask实现大数据可视化，其强大的插件库可以让用户实现个性化的网站定制，开发出功能强大的网站。由于可视化涉及大量的图表展示，这一部分采用Echarts来实现。

主页展示对数据库所有二手房信息的总体统计结果，从左到右，从上到下依次为：“城市房价排名”，“其它数据统计”，“房源地域分布”，“房屋用途统计”，和“户型分类统计”。从“城市房价排名”可以看出，北京、深圳、上海的房价高居全国前三，与我们在生活中的经验相符合，这表明本项目统计的信息是可信的和有价值的。如图3.2所示。

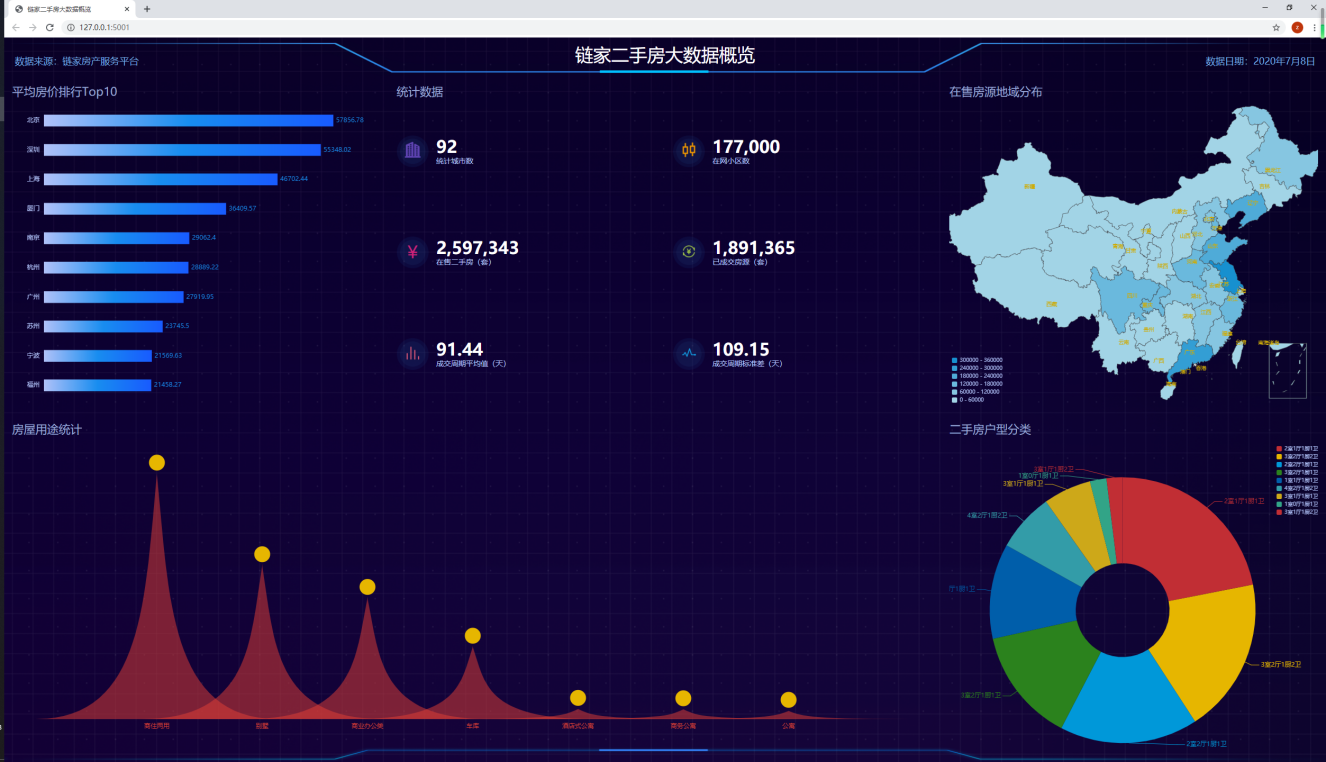


图3.2 主页展示图

在主页选择任意城市，可进入其详情页。图3.3和3.4分别展示“北京”和“重庆”的详情页，从左到右，从上到下依次为：“房屋朝向统计”，“交易热点地区显示”，“楼层占比与装修信息统计”，“总体数值统计”，“房源标签统计”和“总价区间统计”。从“交易热点地区显示”可以看出，二手房交易的热点地区基本上都避开了热门景点和重要商圈。结合生活经验，我们认为合理的解释是，热门景点和重要商圈范围内本身就很少存在居民楼，建造的建筑基本上都是商业性建筑，因而很少有二手房交易。

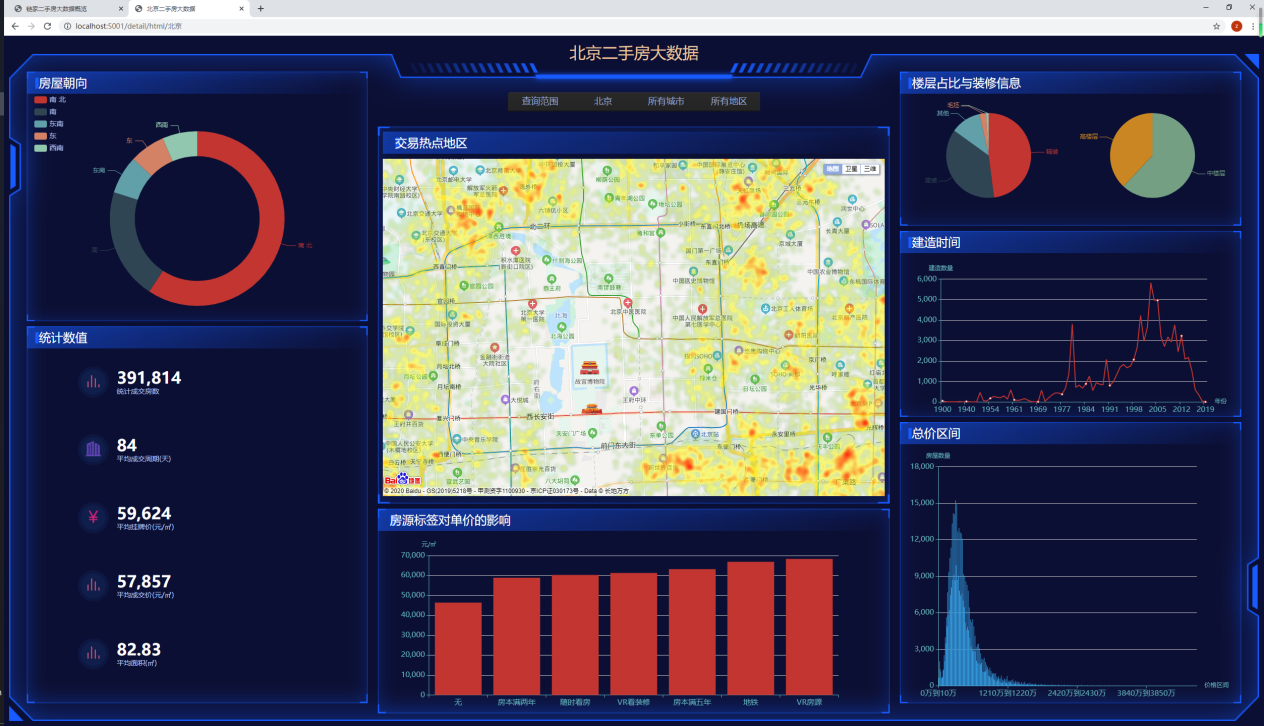


图3.3“北京”的详情页



图3.4 “重庆”的详情页

## **4 小结**

本案例的主要研究内容是大数据背景下房屋估值问题，研究的目的是掌握大数据的搜集方法，数据挖掘方法以及数据可视化方法，同时使用了Python爬虫技术，并以机器学习中的决策树算法作为预测的主要工具。该案例主题贴合实际，将现实生活与多种大数据技术相结合，具有一定的经济和社会效益，容易引起学生的学习兴趣，能够锻炼学生在处理实际问题时的独立思考和解决问题的能力。同时，本案例具有可拓展性，在教学过程中，可以引导学生学习其他的数据挖掘、处理工具和机器学习算法，做到举一反三。

## **附录**

1. 本案例提供配套的PPT、视频、数据集与代码等，发布于Github，链接为：https://github.com/Wanghui-Huang/CQU\_bigdata。

2. 本案例涉及到数据预处理以及多种机器学习算法，建议使用python语言进行编写，推荐的工具包有pandas（数据读取与预处理库），scikit-learn（机器学习算法库），Flask和ECharts（可视化绘图库）。

3. 本案例参考文献如下：

[1] SHVACHKO K, KUANG H, RADIA S, et al. The Hadoop Distributed File System; proceedings of the IEEE Symposium on Mass Storage Systems & Technologies, F, 2010 [C].

[2] DEAN J, GHEMAWAT S. MapReduce: Simplified Data Processing on Large Clusters [J]. USENIX Association, 2004.

[3] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing; proceedings of the Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, F, 2012 [C].

[4] TAYLOR R C. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics [J]. Bmc Bioinformatics, 2010, 11 Suppl 12(S12): S1.

[5] 刘小虎, 李生. 决策树的优化算法 [J]. 软件学报, 1998, 9(10): 4.

[6] 格林贝格. Flask Web开发 [M]. Flask Web开发, 2015.

[7] 王子毅, 张春海. 基于ECharts的数据可视化分析组件设计实现 [J]. 微型机与应用, 2016, 35(014): 46-8.