二手房房价数据采集与分析

泰迪智能科技（武汉）有限公司

目 录

contents

第一部分

研究背景与目标

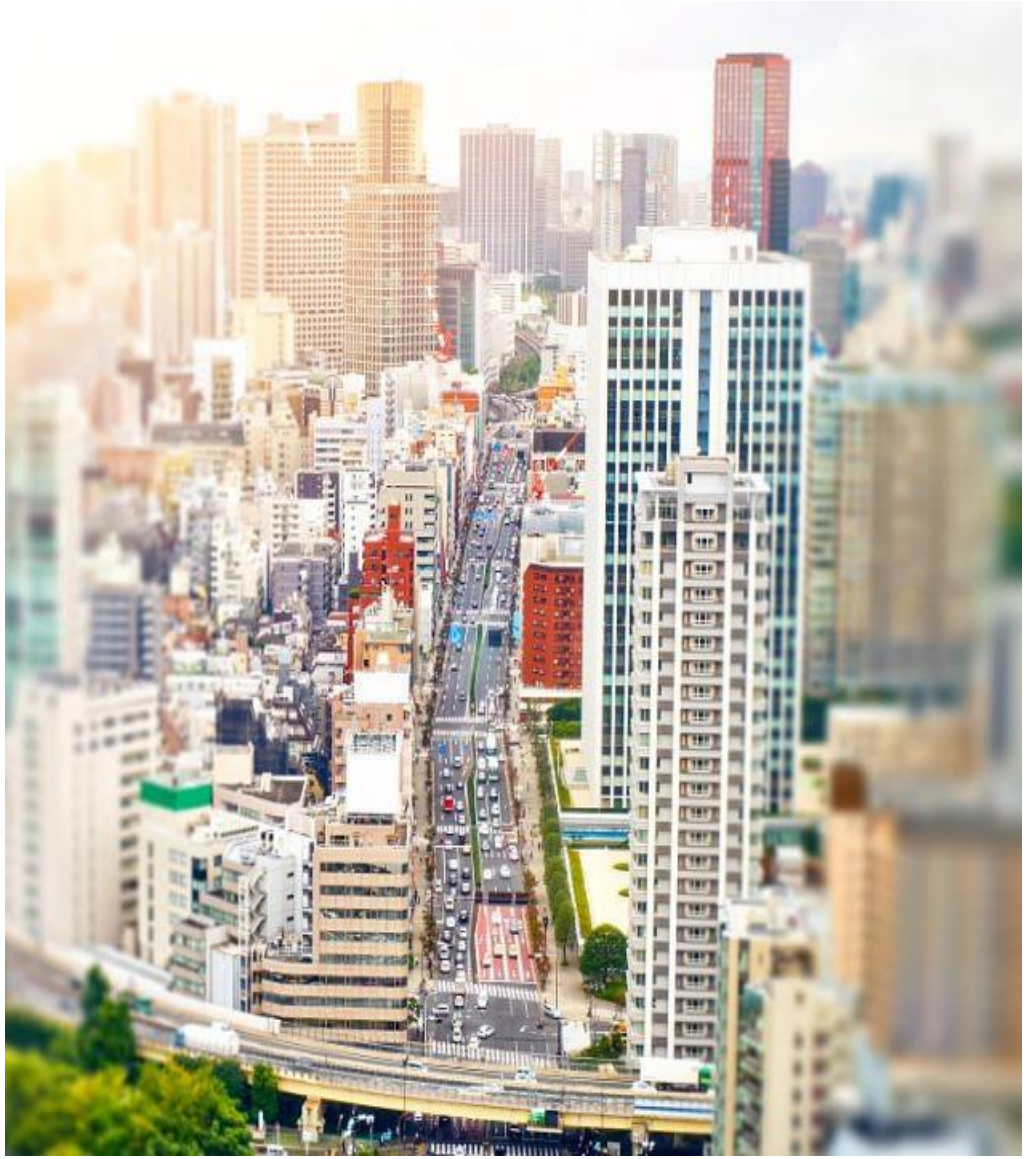
- 研究背景
- 项目目标
- 数据说明

研究背景

研究背景基础介绍

房地产市场作为国民经济的重要组成部分，其价格波动不仅关系到国家经济的稳定发展，也直接影响到居民的生活质量和社会福利。近年来，随着经济的快速发展和城市化进程的加快，房价问题已经成为社会关注的焦点。

房价的波动受多种因素影响，包括宏观经济状况、人口迁移、土地供应政策、金融政策、市场预期等。因此，对房价进行深入的数据分析与预测，对于政府制定合理的房地产政策、投资者做出明智的投资决策以及普通居民规划购房计划都具有重要意义。



项目目标

爬虫目标

使用爬虫访问链家指定地区的二手房数据，并按照要求存储指定字段的数据。
指定字段名称如下： 标题,小区名称,地点,总价,每平米价,房型,面积,朝向,风格,楼层,结构

数据分析目标

1. 数据预处理：在进行分析之前对爬取后的数据需要进行去重、文本提取等相关操作。
2. 数据分析：挖掘当前房屋售价与各方面要素的相关情况。
3. 数据可视化：可以绘制柱状图、折线图、热力图等来帮助我们更直观的看到房屋售价情况。
4. 房屋售价预测：通过之前数据分析得到的相关要素，用它们来实现对房屋价格的回归预测。

数据说明

爬虫需要耗费一定的时间，本案例配套已经爬取的数据。准备数据与爬取数据字段略有差异，具体说明如下

字段名称	解释	字段名称	解释
房源标题	卖方者提供的一些有价值的信息	户型	房源属于几室几厅
城市	房源所属城市	面积	房源面积
城市等级	房源所属城市等级	楼层位置	房源所属楼层位置
小区	房源所在小区	朝向	房源朝向
建房时间	房源修建时间	单价	房源每平方米单价
楼房位置	房源所属小区	总价	房源售卖总价

第二部分

数据获取

- 页面分析
- 数据解析

页面分析

https://zh.lianjia.com/ershoufang/pg3/

翠湖苑 精装两房 中间楼层 自住出租皆可... 必看好房

翠湖苑 - 井岸镇

2室2厅 | 83.7平米 | 南 | 精装 | 中楼层(共24层) | 板塔结合

1人关注 / 4个月以前发布

VR看装修 房本满五年

65万

7,766元/平

前山河畔,电梯高层看河景三房，配套完... 必看好房

宏华花园 - 前山

3室2厅 | 94.6平米 | 东南 | 简装 | 高楼层(共9层) | 板楼

2人关注 / 一年前发布

VR看装修 房本满五年 随时看房

138万

14,588元/平

天庆晋海岸 精装三房两卫 中间楼层 必看好房

天庆晋海岸 - 三灶镇

3室2厅 | 91平米 | 北 | 简装 | 中楼层(共30层) | 板楼

105万

11,539元/平

试试地图找房

下载链家APP

扫描上方二维码

随时查看新房源

了解更多 >

卖房

客服

在线咨询

网页源码解析示例

```
data-housecode="105117158288" data-is_focus=""data-sl=""><!-- 热推标签、埋点 --><img
src="https://s1.ljcdn.com/feroot/pc/asset/img/vr/vrgold.png?_v=20240731114856" class="vr_item"><img class="lj-lazy"
src="https://s1.ljcdn.com/feroot/pc/asset/img/blank.gif?_v=20240731114856" data-original="https://image1.ljcdn.com/110000-
inspection/8406fa27-9cec-470c-ac14-58dfdf1c3de2_1000.jpg.296x216.jpg" alt="珠海香洲区南湾"></a><div class="info clear"><div
class="title"><a class="" href="https://zh.lianjia.com/ershoufang/105117158288.html" target="_blank" data-log_index="3"  data-
el="ershoufang" data-housecode="105117158288" data-is_focus="" data-sl="">中信红树湾高楼层大三房 看河景</a><!-- 拆分标签 只留一
个优先级最高的标签--><span class="goodhouse_tag tagBlock">必看好房</span></div><div class="flood"><div class="positionInfo">
<span class="positionIcon"></span><a href="https://zh.lianjia.com/xiaoqu/6320022386470129/" target="_blank" data-log_index="3"
data-el="region">中信红树湾北区 </a>  - <a href="https://zh.lianjia.com/ershoufang/nanwan/" target="_blank">南湾</a> </div>
</div><div class="address"><div class="houseInfo"><span class="houseIcon"></span>3室2厅 | 126.13平米 | 东北 | 精装 | 32层 | 板
楼</div></div><div class="followInfo"><span class="starIcon"></span>2人关注 / 1个月以前发布</div><div class="tag"><span
class="isVrFutureHome">VR看装修</span></div><div class="priceInfo"><div class="totalPrice totalPrice2"><i> </i><span
class="">426</span><i>万</i></div><div class="unitPrice" data-hid="105117158288" data-rid="6320022386470129" data-price="33775">
<span>33,775元/平</span></div></div></div><div class="listButtonContainer"><div class="btn-follow followBtn" data-
hid="105117158288"><span class="follow-text">关注</span></div><div class="compareBtn LOGCLICK" data-hid="105117158288" log-
mod="105117158288" data-log_evtid="10230">加入对比</div></div></li><li class="clear LOGVIEWDATA LOGCLICKDATA" data-
lj_view_evtid="21625" data-lj_evtid="21624" data-lj_view_event="ItemExpo" data-lj_click_event="SearchClick"  data-
lj_action_source_type="链家_PC_二手列表页卡片" data-lj_action_click_position="3" data-lj_action_fb_expo_id='871426973665583167'
data-lj_action_fb_query_id='871426973149683712' data-lj_action_resblock_id="6315185584360439" data-
lj_action_housedel_id="105115457016" ><a class="noresultRecommend img LOGCLICKDATA"
href="https://zh.lianjia.com/ershoufang/105115457016.html" target="_blank" data-log_index="4"data-el="ershoufang" data-
```

解析过程

由于本案例要爬取的网页是静态展示的，可以采用request访问网页，用lxml中的etree修改数据格式并利用Xpath解析内容。具体说明如下：

01

网页请求

使用Requests库向目标网页发送请求，获取网页内容。

02

数据格式修改

利用lxml库的etree模块对获取到的网页数据进行格式上的修改，使其更易于解析。

03

内容解析

使用XPath语法对已修改格式的数据进行解析，提取需要的信息。

第三部分

数据预处理

- 缺失值处理
- 重复值处理
- 文本处理

数据预处理

缺失值处理

代码：

```
1 data.isnull().sum()
```

运行结果：

```
[4]: 房源标题      0
     城市          0
     城市等级      0
     小区          0
     户型          0
     面积          0
     楼层位置      0
     朝向          0
     建房时间      0
     楼房位置      0
     单价          0
     总价          0
     dtype: int64
```

结论：数据不存在缺失值。

重复值处理

房屋销售数据在爬取过程中可能出现重复爬取的情况，以下对重复数据进行处理。

检查重复值代码：

```
print(data.duplicated().sum())
data[data.duplicated(keep=False)].sort_values(by='房源标题')
```

删除重复值

```
print(data.shape)
# 删除重复的行，并保留第一次出现的
data.drop_duplicates(inplace=True)
print(data.shape)
```

数据预处理

重复值处理

重复数据显示：

1568

	房源标题	城市	城市等级	小区	户型	面积	楼层位置	朝向	建房时间	楼房位置	单价	总价
29703	(京基山海御园)地铁口,赠送精装 ,新盘在售,非常的繁华。	深圳	一线	京基智农·山海御园	3室2厅	98㎡	底层 (共1层)	南向	2018年建	西乡	64700元/㎡	634.1万
29176	(京基山海御园)地铁口,赠送精装 ,新盘在售,非常的繁华。	深圳	一线	京基智农·山海御园	3室2厅	98㎡	底层 (共1层)	南向	2018年建	西乡	64700元/㎡	634.1万
11821	(价格可谈)金碧世纪花园 精装修三房 满五年 少税	广州	一线	金碧世纪花园	3室2厅	90.89㎡	中层 (共32层)	南北向	2005年建	大沙地	44779元/㎡	407万
13754	(价格可谈)金碧世纪花园 精装修三房 满五年 少税	广州	一线	金碧世纪花园	3室2厅	90.89㎡	中层 (共32层)	南北向	2005年建	大沙地	44779元/㎡	407万
12963	(内转)五矿招商鹭山府 一手价格在售 鱼珠港国际金融城	广州	一线	五矿招商鹭山府	3室2厅	96.5㎡	低层 (共32层)	南向	2019年建	大沙地	38860元/㎡	375万

删除前后的数据大小：

```
(32533, 12)
(30965, 12)
```

文本处理

提取文本中有用的数据

比如将原始“面积”数据的形式“103.87m²”→103.87；
比如将原始“楼层位置”数据的形式“低层（共36层）”→36.

思路：利用正则表达式处理，删除数字前后的文字。

```
for i in ['面积','建房时间','单价','总价','楼层位置']:
    data[i]=data[i].apply(lambda x: float(re.findall(r'-?\d+\.\d+|-?\d+',x)[0]))
```

部分结果展示：

面积	楼层位置	朝向	建房时间	楼房位置	单价	总价
103.87	36.0	南向	2000.0	新华路	142485.0	1480.0
108.91	27.0	南向	1999.0	打浦桥	127628.0	1390.0
60.73	6.0	南向	1986.0	淞宝	33591.0	204.0
105.88	15.0	南向	2006.0	天山	112391.0	1190.0
62.28	24.0	南向	1991.0	徐家汇	118497.0	738.0

处理户型

在房屋销售中，房子是几室几厅是决定房价至关重要的一个因素，因此我们需要单独把房屋的“室”和“厅”提取出来。

思路：利用正则表达式处理，提取数字。

```python
def fun(x):
    try:
        rooms = re.findall(r'\d', x)
        return (rooms[0], rooms[1])
    except:
        return 0,0
data["室"], data["厅"] = zip(*data['户型'].apply(fun))
```

部分结果展示：

户型	面积	楼层位置	朝向	建房时间	楼房位置	单价	总价	室	厅
2室2厅	103.87	36.0	南向	2000.0	新华路	142485.0	1480.0	2	2
2室2厅	108.91	27.0	南向	1999.0	打浦桥	127628.0	1390.0	2	2
2室1厅	60.73	6.0	南向	1986.0	淞宝	33591.0	204.0	2	1
3室2厅	105.88	15.0	南向	2006.0	天山	112391.0	1190.0	3	2
2室1厅	62.28	24.0	南向	1991.0	徐家汇	118497.0	738.0	2	1

第四部分

统计分析与可视化

- 房屋内部因素可视化
- 房屋外部因素可视化

房屋内部因素可视化

绘制卖家在描述房屋特点的词云图

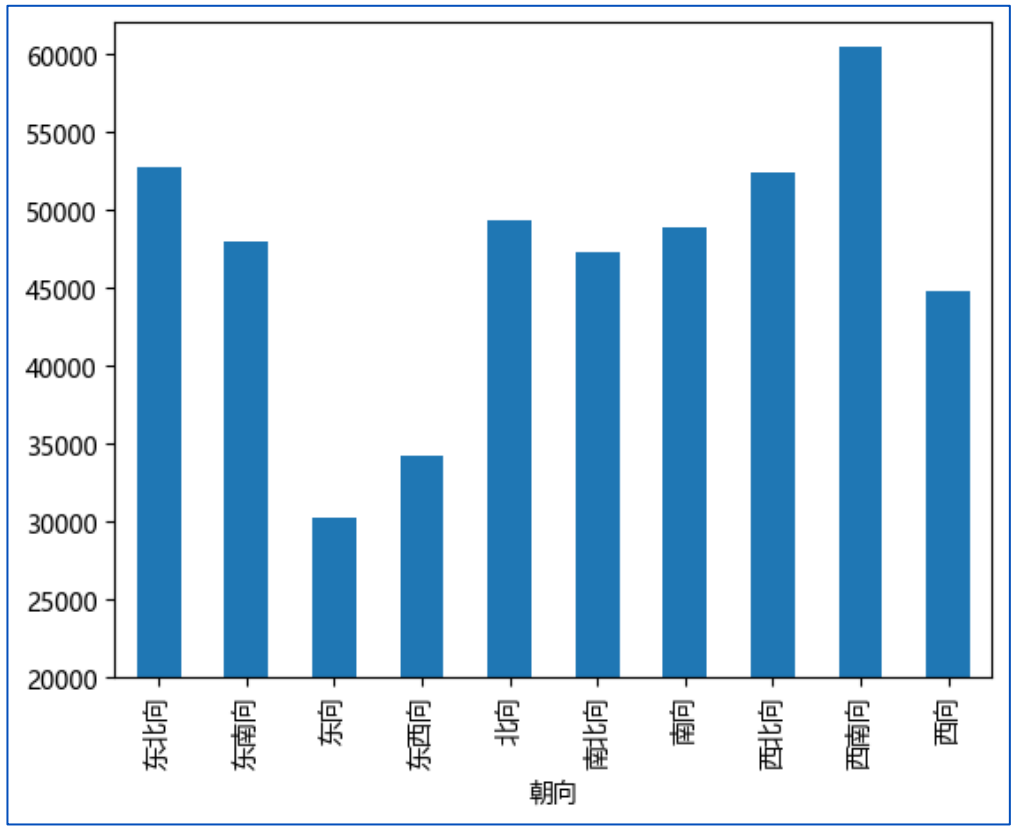
```
[price_dict[key] for key in ['院中','耗资','广粤']]
```

```
[193702.5, 202663.0, 185711.66666666666]
```

可以看出带有“院中”、“耗资”、“广粤”等词语的房屋的均价就会比较贵。

房屋内部因素可视化

绘制价格与朝向的变化关系柱状图

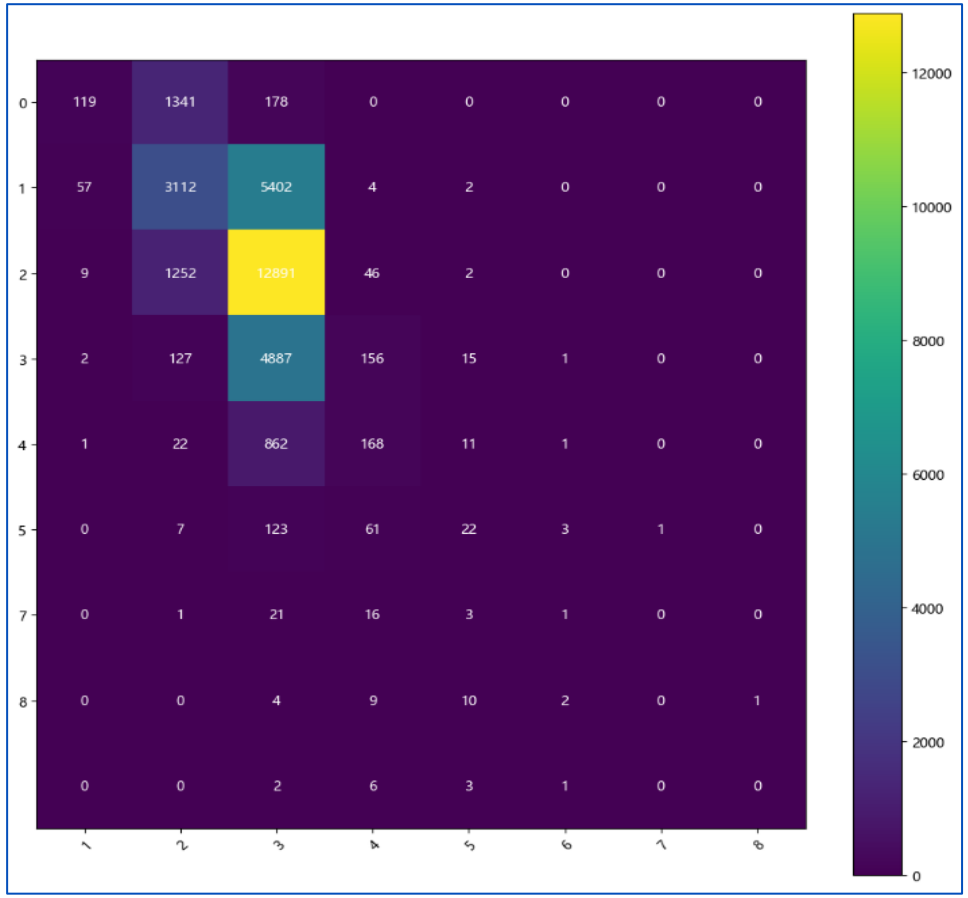


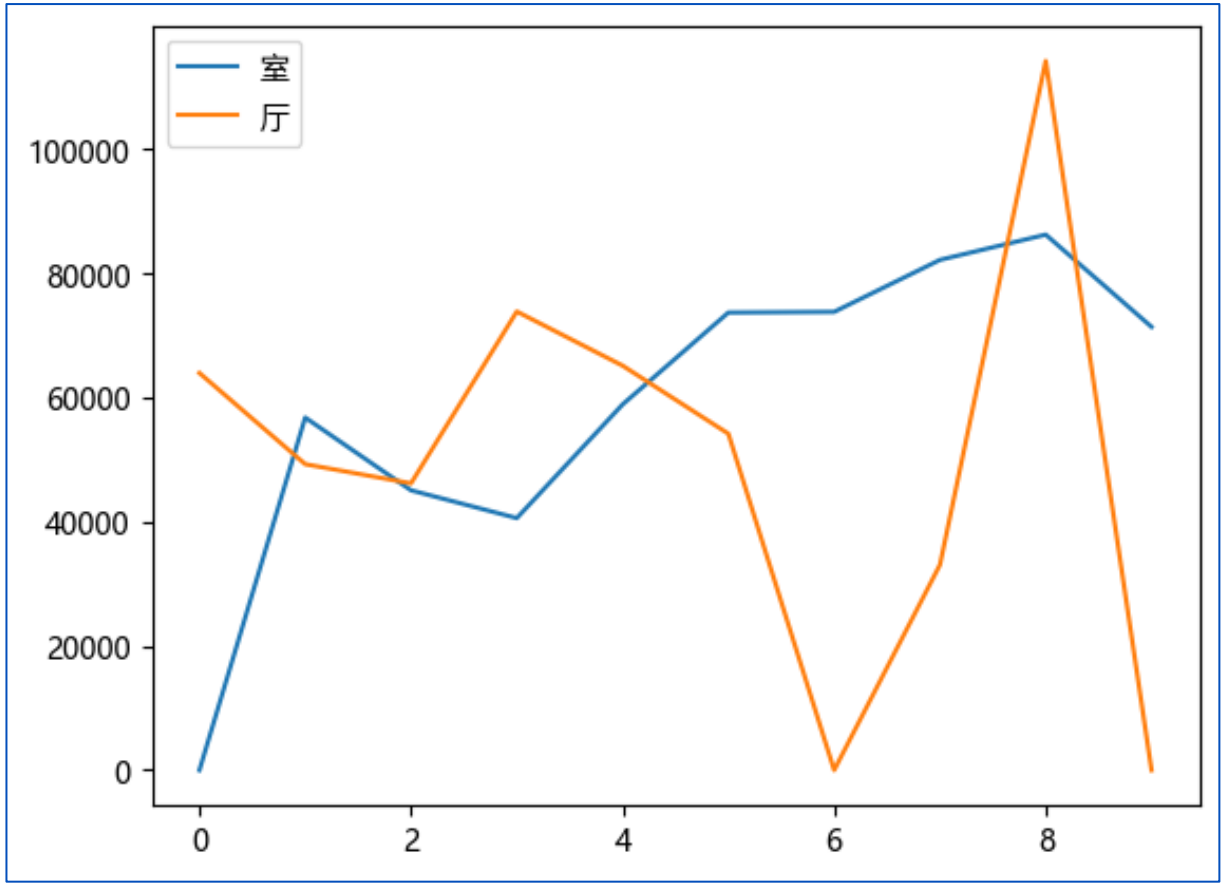
结论：在房屋的朝向方面，“东向”和“东西向”的价格是最低的，这也比较容易理解，朝东向的房屋采光不是很好。

房屋内部因素可视化

绘制不同户不同厅对应数量分布的热力图和折线图

结论：热力图表明大多数房屋为3室2厅；折线图表示8厅、8室房屋的均价都是最高，当然这可能和数据样本有一定关系。

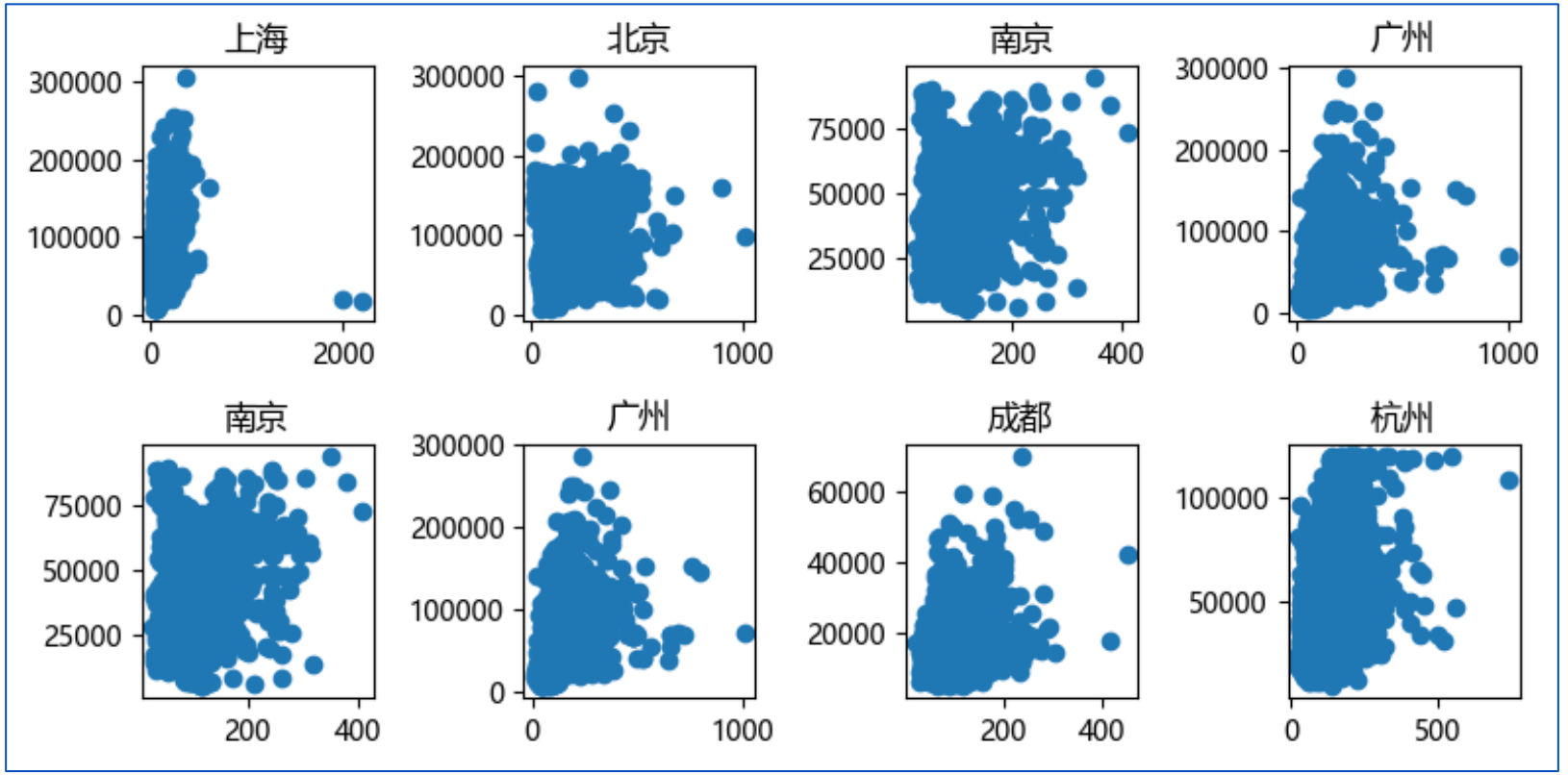




房屋内部因素可视化

绘制房屋面积与房价对应的散点图

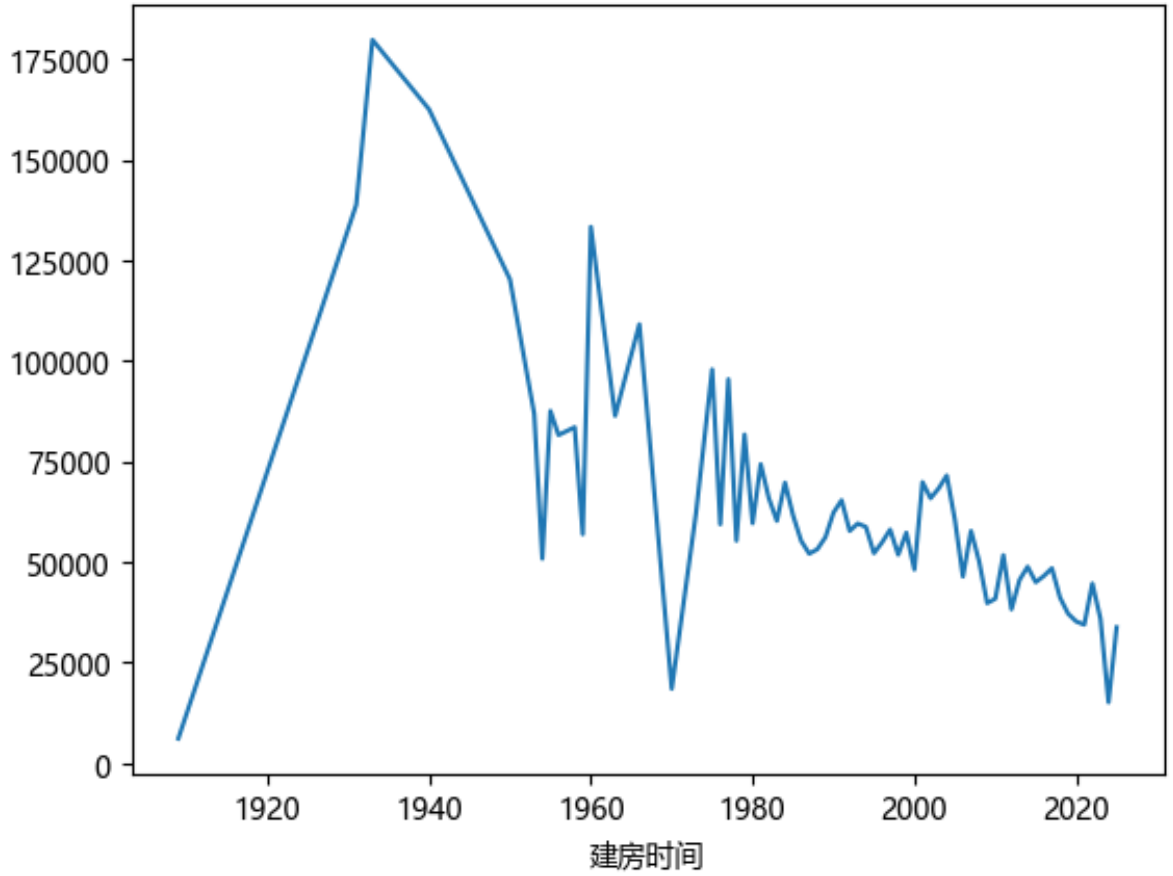
- 首先对每个城市所在的房屋，依据不同的面积，对“单价”求平均；
- 绘制每个城市的散点图。



结论：上海的每平方米的房价随着房屋面积的增加增加的最快，平均每平方米的房价最高。

房屋内部因素可视化

绘制价格与房龄的变化关系折线图

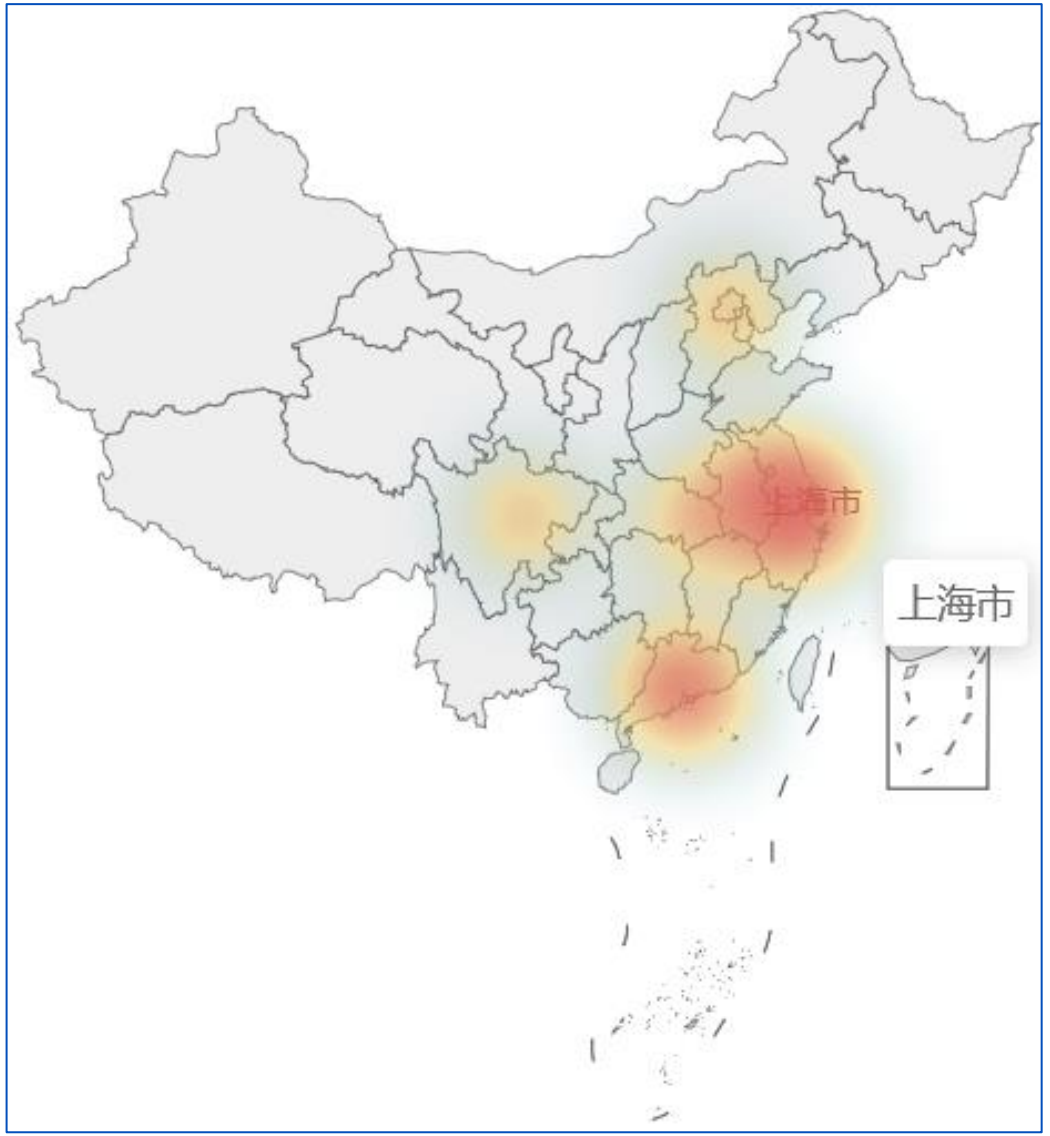


结论：在二手房中，每平方米的房屋价格随着时间是逐年递减的，这可能与环境和地理位置有很大关系（比如北京四合院虽然比较旧，但是房价比全国大多数地方都要高）。

房屋外部因素可视化

绘制不同城市房屋均价的地图热力图

```
[('上海', 74109),
 ('北京', 91263),
 ('南京', 36656),
 ('广州', 54108),
 ('成都', 16036),
 ('杭州', 45654),
 ('武汉', 18617),
 ('深圳', 70603)]
```



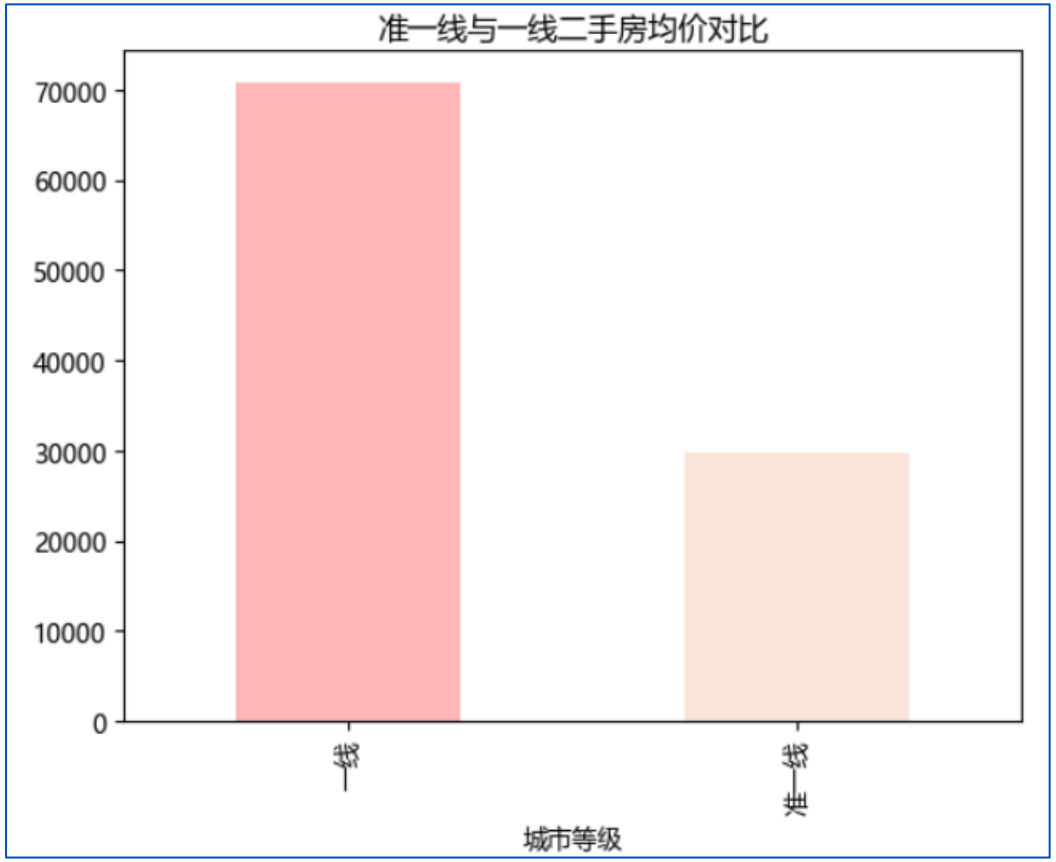
可以看到在全国范围内，省会、中心城市的房价还是较高，特别是江浙一带、珠三角一带房价特别高。

房屋外部因素可视化

绘制“一线”与“准一线”城市均价柱状图

研究城市等级与其对应房屋均价的关系还是很有必要的，通过“一线”与“新一线”房屋均价对比就可以看出其是否具有明显差距。

- 首先对“城市等级”进行分组，对“单价”求平均后绘制柱状图。



结论：“一线”与“新一线”城市房子平均价格差距明显。

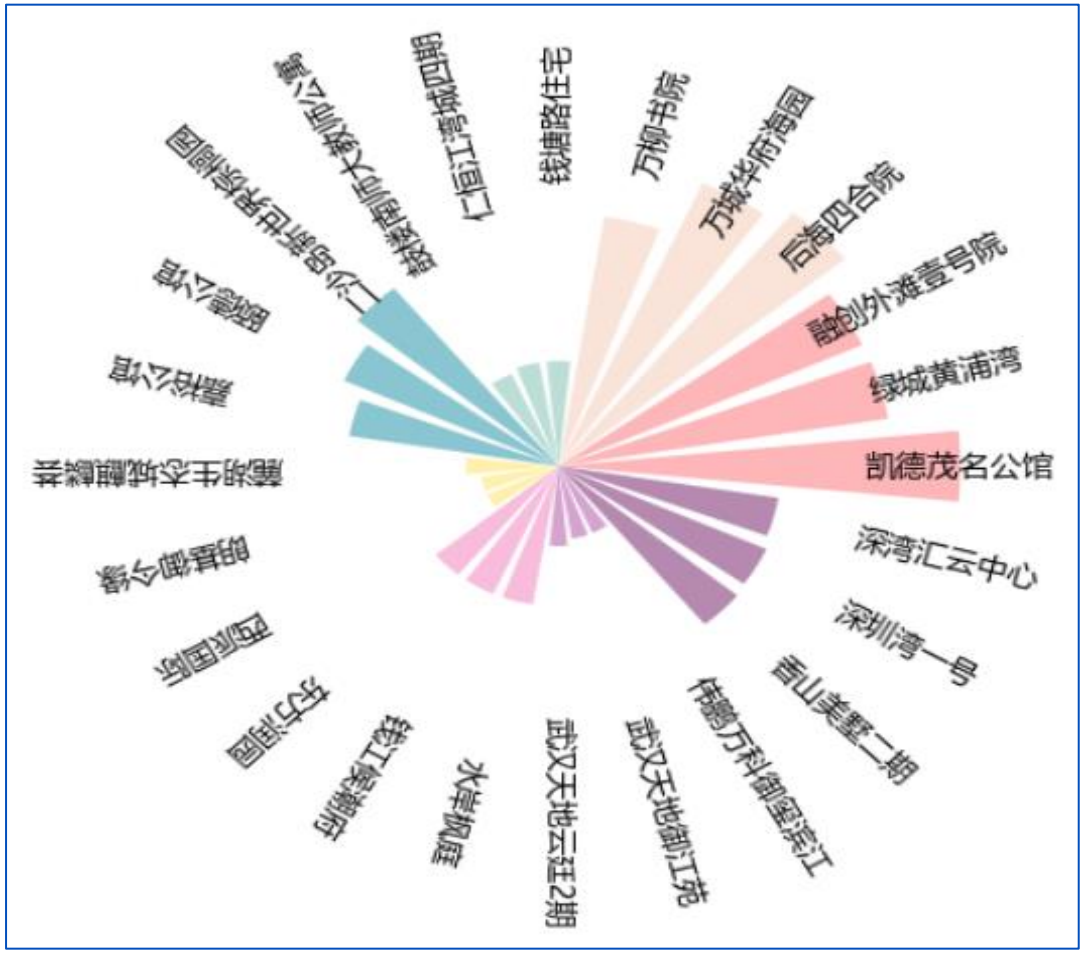
房屋外部因素可视化

绘制各城市房价前三的小区的玫瑰图

通过此极坐标图，我们就能知道在此数据中，各城市前三名的小区是哪几个。同时可以明显对比看个各城市顶尖小区平均价格的差距大小。

思路：

- 首先对“小区”进行分组，对“单价”求平均；
- 选取各地区平均价格前三的小区；
- 绘制极坐标图。

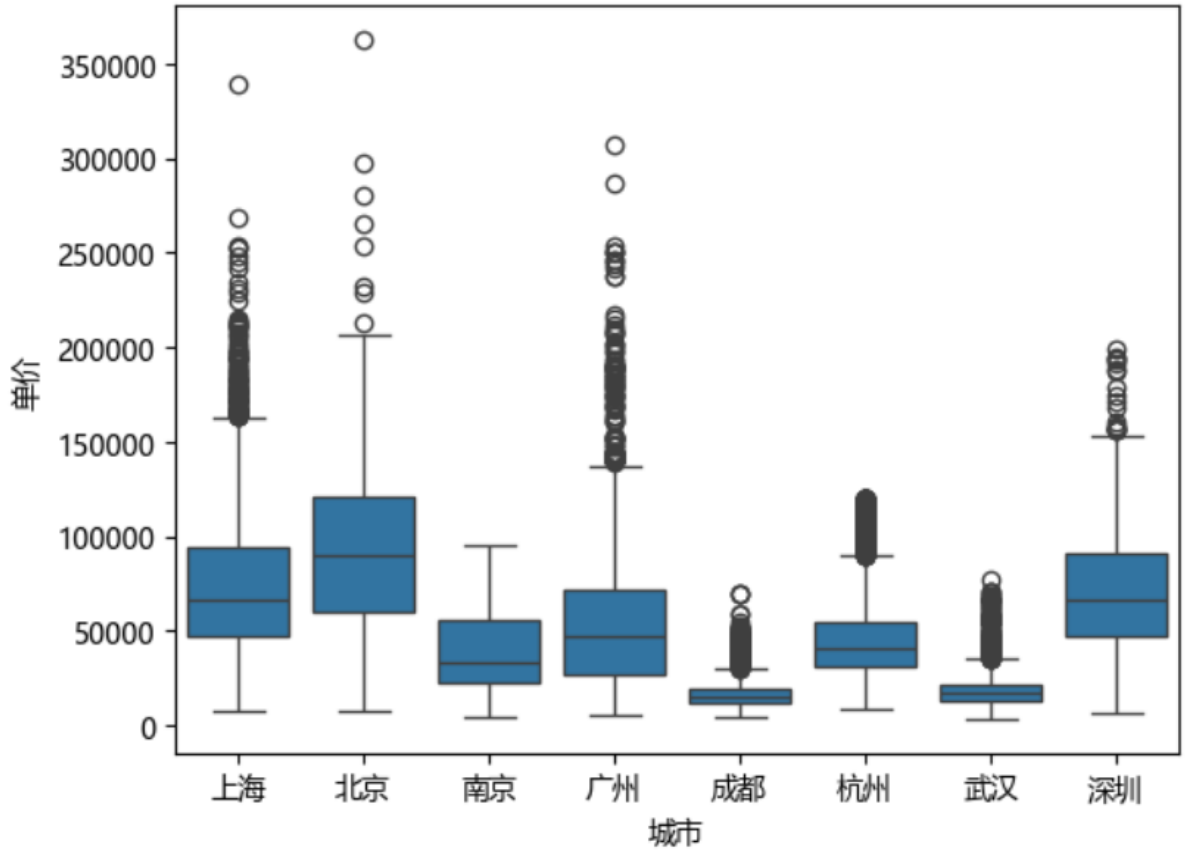


结论：在本数据中，上海和北京的房价相对来说较高，其中“凯德茂名公馆”小区的房价最高。

房屋外部因素可视化

不同城市的房价分布箱线图

对每个城市的房屋单价进行箱线图绘制，可以发现，北上广深的房价分布最高，其次是南京、杭州。选择的八个城市中，成都和武汉的房价分布区间是最低的。



第五部分

房屋价格预测

- 数据准备
- 回归模型的建立

数据准备

“城市”、“方向”、“城市级别”的映射

我们首先需要把文本类型的数据映射为数值类型的数据。

```
# 城市处理
city_ans.sort_values(inplace=True)
city_dict={item:i for i,item in enumerate(city_ans.index)}
# 方向处理
direction_data.sort_values(inplace=True)
direction_dict={item:i for i,item in enumerate(direction_data.index)}
# 城市级别处理
level_dict={'一线':1,'准一线':0}
data['city']=data['城市'].map(city_dict)
data['dir']=data['朝向'].map(direction_dict)
data['level']=data['城市等级'].map(level_dict)
```

结果：

city	dir	level
6	5	1
6	5	1
6	5	1
6	5	1
6	5	1
...	...	...
5	8	1
5	9	1
5	5	1
5	4	1
5	5	1

数据准备

提取训练所需的数据列

并不是所有数据都是我们训练所需要的，我们只需要将【city, dir, level,室, 厅,面积,楼层位置,建房时间】提取出来即可，并将所有的数据列都设为浮点类型。

```
X,Y=data[['city', 'dir', 'level','室', '厅','面积',
          '楼层位置','建房时间']],data['单价']
X=X.astype(float)
```

特征X：

	city	dir	level	室	厅	面积	楼层位置	建房时间
0	6.0	5.0	1.0	2.0	2.0	103.87	36.0	2000.0
1	6.0	5.0	1.0	2.0	2.0	108.91	27.0	1999.0
2	6.0	5.0	1.0	2.0	1.0	60.73	6.0	1986.0
3	6.0	5.0	1.0	3.0	2.0	105.88	15.0	2006.0
4	6.0	5.0	1.0	2.0	1.0	62.28	24.0	1991.0
...	...	...	...	...	...	...	...	...

标签Y：

```
0    142485.0
1    127628.0
2     33591.0
3    112391.0
4    118497.0
```

回归模型的建立

建立简单的回归模型

一般线性回归模型$\hat{y} = \beta_0 + \beta_1 x_1 + \ldots + \beta_8 x_8$

```
# 划分数据集，通常使用 70-30 或 80-20 的比例
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42)
# 创建线性回归模型实例
model = LinearRegression()
# 训练模型
model.fit(X_train, y_train)
# 在测试集上进行预测
y_pred = model.predict(X_test)
# 计算预测的均方误差
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
# 打印模型参数
print(f"Coefficients: {model.coef_}")
# 打印截距
print(f"Intercept: {model.intercept_}")
plt.scatter(y_test,y_pred,alpha=0.1)
```

回归结果：

误差平方和	600898148
β_0	1188495.16
β_1	11022.63
β_2	437.72
β_3	-5182.71
β_4	131.32
β_5	-1214.30
β_6	142.13
β_7	457.00
β_8	-596.65

XGBoost模型的建立

XGBoost回归模型简介

一般线性回归模型的误差平方和MSE较大，以下建立XGBoost回归模型。先首先对xgboost模型进行简单介绍。

XGBoost（极限梯度提升）算法是由陈天奇博士在2016年发表的论文《XGBoost：一种可扩展的树提升系统》中正式提出的。这一算法在梯度提升决策树（GBDT）算法的基础上进行了多项创新性的优化。

具体来说，XGBoost在损失函数的计算过程中引入了二阶导数，这有助于更准确地估计模型参数。同时，它还增加了正则化项，以防止模型过拟合，提高泛化能力。此外，XGBoost还实现了一定程度的并行计算，这显著提高了算法的训练速度和处理大规模数据集的能力。通过这些优化，XGBoost在多个机器学习竞赛中表现出色，成为了一种广泛应用的集成学习算法。

XGBoost的核心是梯度提升算法（Gradient Boosting），它通过集成多个弱学习器（通常是决策树）来构建一个强大的集成模型。与传统的梯度提升算法相比，XGBoost引入了一些创新，使其在性能和速度上都有所提升。

回归模型的建立

XGBoost回归模型实例

XGBoost回归模型的建立如下：

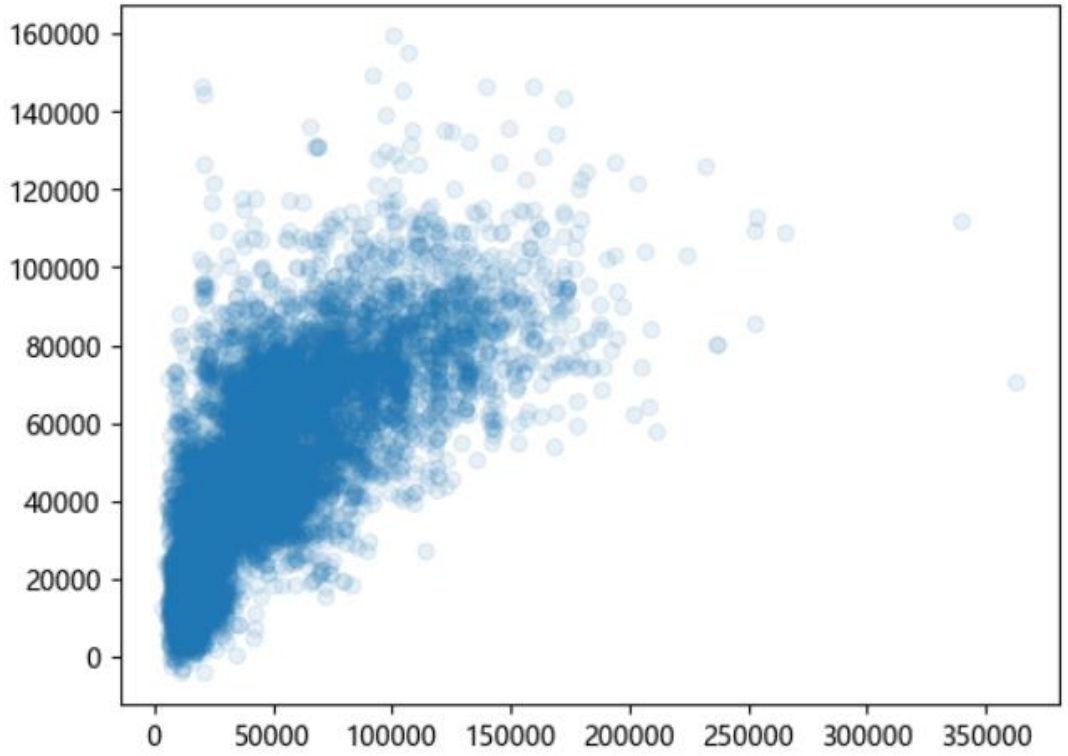
```
import xgboost as xgb
# 创建XGBoost回归模型实例
xgb_reg = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=100,
seed=42)
# 训练模型
xgb_reg.fit(X_train, y_train)
# 在测试集上进行预测
y_pred = xgb_reg.predict(X_test)
# 计算预测的均方误差
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
plt.scatter(y_test,y_pred,alpha=0.1)
```

XGBoost回归模型的误差平方和MSE：376230258.82。此模型的MSE相较于普通的线性回归减少了37.38%。

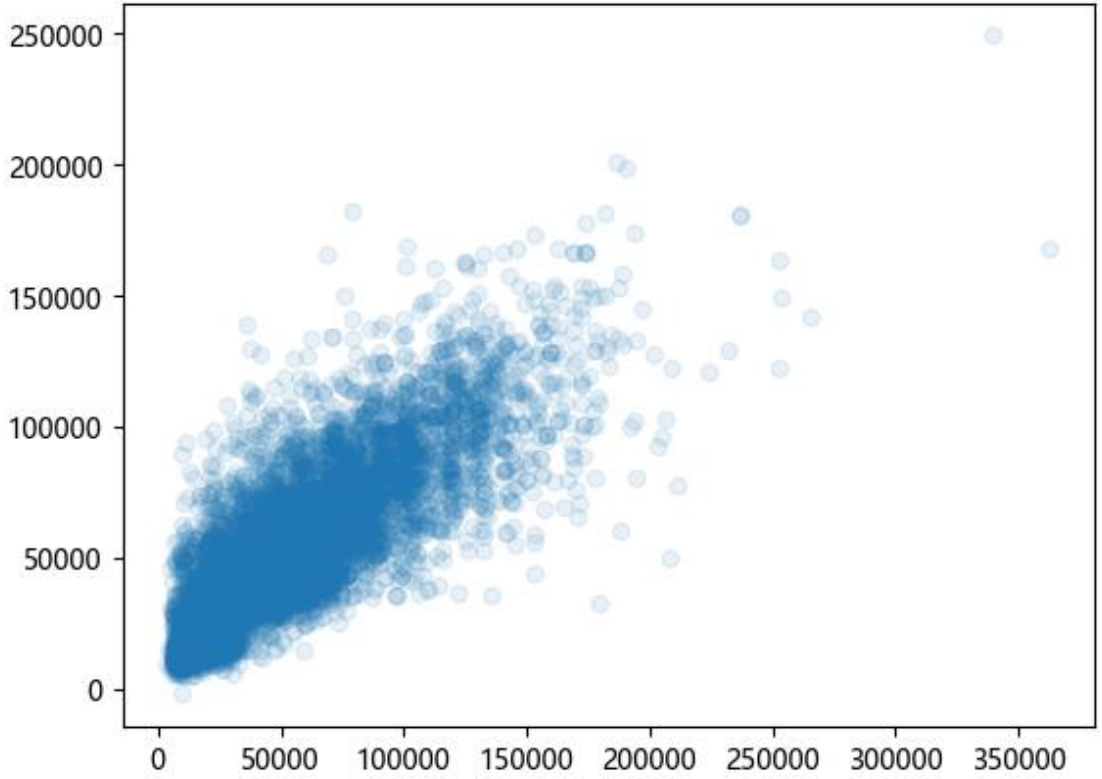
回归模型的建立

线性回归模型和XGBoost回归模型结果对比

一般线性回归模型MSE=600898148



xgboost回归模型MSE=376230258



结论：我们大致可以看到普通线性回归的模型并不好，回归效果比较分散；XGBoost回归模型预测的就较为集中，符合实际情况。

第六部分

小结

- 总结与分析
- 实际意义

总结与分析

本研究通过深入的数据分析与预测，为房地产市场的参与者提供了宝贵的信息。通过可视化手段，直观展示了影响房价的多种因素，并通过回归模型对房价进行了有效预测。

具体实施步骤如下：

1. 数据处理：导入数据，进行数据检查和预处理，提取数字信息。
2. 户型处理：对房屋户型进行分析和处理。
3. 房源特点可视化：展示房源的各类特点，如房屋内部因素和标题。
4. 房屋特征分析：包括朝向、户型、面积、年份等内部因素，以及城市、城市等级、小区等外部因素。
5. 房价分析：对比不同厅室的价格，绘制热力图和计算均值，分析不同城市的均价和房价分布情况。
6. 房价预测：通过映射字典处理数据，提取关键信息，建立模型进行房价预测。

实际意义

本案例的意义在于，通过对房产数据的全面分析与可视化展示，我们能够深入理解房屋的内部因素（如户型、朝向、面积、年份）和外部因素（如城市、城市等级、小区）对房价的影响，从而为房地产市场提供有价值的信息。通过数据预处理和特征提取，我们能够更准确地预测房价走势，帮助购房者、投资者和开发商做出更明智的决策，同时揭示不同城市间房价差异的原因，为城市规划和房地产市场调控提供数据支持。

谢谢