

Airbnb 短租房源数据分析

<第二组>期末报告

目录

1. 问题的提出.....	2
2. 数据来源与数据说明.....	2
2.1 原始数据.....	2
2.2 初步整理.....	3
3. 数据清洗.....	4
4. 探索性数据分析.....	5
4.1 所用工具.....	5
4.2 总体情况.....	6
4.2.1 房源区域密度.....	6
4.2.2 房源上线时间.....	7
4.2.3 房源地址准确性.....	7
4.2.4 房源取消规则.....	8
4.3 什么样的房子热度高——房源热度分析.....	8
4.3.1 评论数趋势（消费周期）.....	8
4.3.2 行政区域×热度.....	9
4.3.3 房间×热度.....	10
4.3.4 房型×热度.....	10
4.3.5 价格×热度.....	12
4.3.6 房源名.....	14
4.3 房东分析——谁是房东一哥.....	15
4.3.1 整体情况.....	15
4.3.2 房东-房源.....	16
4.3.3 房东特性.....	16
4.3.4 入住规则.....	17
5. 如何找到适合用户的短租房——房源推荐挖掘.....	17
5.1 推荐实例（一）.....	19
5.2 推荐实例（二）.....	24
6. 什么样的房子更贵——价格预测分析.....	27
6.1 影响价格的因素探索.....	27
6.2 价格预测模型.....	32
7. 用户对房源的评价如何——评论分析.....	35
7.1 各项评分情况.....	35
7.2 评论情感分析.....	35
8. 结论.....	37
8.1 结论及建议.....	37
8.2 不足与展望.....	38

1. 问题的提出

共享，通过让渡闲置资源的使用权，在有限增加边际成本的前提下，提高了资源利用效率。随着信息的透明化，越来越多的共享行为发生在陌生人之间。短租是伴随协同消费模式的兴起而出现的一种新兴的房屋租赁形式，主要房源集中在旅游热点地区。它可被视作共享空间的一种模式，不论是否有过入住陌生人家中的体验，我们认为人们都可以从短租的数据里挖掘出有趣的信息。爱彼迎（Airbnb）作为一个典型的旅行房屋短租社区，覆盖范围广（全球191个国家，超3万座城市），市场份额占比大且增长迅猛，实力雄厚（D轮融资，估值约300亿美元），它的数据具有较高的代表性和实际价值，故被我们选定为研究对象。此次数据挖掘的任务具体如下：

- 探索性分析：基于清洗后的数据对房源总体情况、房源热度、房东基本情况等进行探索性分析，此部分具有有一定的发散性。
- 房源推荐数据挖掘：主要是根据用户需求筛选房源，再通过评论数等指标推荐热门房源。
- 价格预测分析：
- 评论文本情感分析：

2. 数据来源与数据说明

2.1 原始数据

我们的数据来自阿里云计算主办的“Tianchi Data Hero Cup —— 短租数据集分析赛”。该活动采用了Airbnb官方网站于2019年4月17日公开的北京地区短租房源相关数据，包括了结构化的表格数据、非结构化的文本和地图数据，方便参与者从统计分析、时间序列、关系网络分析、中英文自然语言处理、数据可视化以及数据应用等多个维度展开探索。数据来源网址具体为：

<https://tianchi.aliyun.com/competition/entrance/231715/introduction>

原始数据集共5个文档，以下为原始数据的具体说明：

文档	格式
calendar_detail.zip	.zip(31MB)
listings.csv	.csv(4MB)
listings_detail.zip	.zip(27MB)
neighbourhoods.csv	.csv(315B)
neighbourhoods.geojson	.geojson(183KB)
reviews.csv	.csv(3MB)
reviews_detail.zip	.zip(19MB)

图 原始数据

listings 数据为短租房源基础信息，包括房源、房东、位置、类型、价格、评论数量和可租时间等等。listingsdetail中包含更多房源相关细节。calendar 数据为短租房源时间表信息，包括房源、时间、是否可租、租金和可租天数等等。reviews 数据为短租房源的评论信息。汇总版中仅包括房源 listingid 和评论日期。reviews_detail 还包括评论相关的内容和作者信息。neighbourhoods 数据为北京的行政区划。

2.2 初步整理

初步整理形成以下 5 个表，每个表约 28444 条记录，数据来源于北京地区，时间范围：2010/8/25–2019 年 4/17 房源表 (listingid) 房东表 (listingid, hostid) 评论表 (listingid) 评论时间表 (listingid) 评论文本 (listingid, review_id)。

字段名称	表	字段名称	表	字段名称	表
# Id	清洗2_房源表.csv	# id	房东表.csv	# id	评论表.csv
Abc Summary	清洗2_房源表.csv	# host_id	房东表.csv	# number_of_reviews	评论表.csv
Abc Description	清洗2_房源表.csv	Abc host_name	房东表.csv	Abc first_review	评论表.csv
Abc Notes	清洗2_房源表.csv	Abc host_since	房东表.csv	Abc last_review	评论表.csv
Abc Transit	清洗2_房源表.csv	Abc host_location	房东表.csv	Abc review_scores_rating	评论表.csv
Abc Access	清洗2_房源表.csv	Abc host_about	房东表.csv	Abc review_scores_accuracy	评论表.csv
Abc House Rules	清洗2_房源表.csv	Abc host_response_time	房东表.csv	Abc review_scores_cleanliness	评论表.csv
Abc Latitude	清洗2_房源表.csv	Abc host_response_rate	房东表.csv	Abc review_scores_checkin	评论表.csv
Abc Longitude	清洗2_房源表.csv	Abc host_acceptance_rate	房东表.csv	Abc review_scores_communication	评论表.csv
T/F Is Location Exact	清洗2_房源表.csv	Abc host_is_superhost	房东表.csv	Abc review_scores_location	评论表.csv
Abc Property Type	清洗2_房源表.csv	# host_listings_count	房东表.csv	Abc review_scores_value	评论表.csv
Abc Bed Type	清洗2_房源表.csv	# host_total_listings_count	房东表.csv	# reviews_per_month	评论表.csv
Abc Amenities	清洗2_房源表.csv				
Abc Accommodates	清洗2_房源表.csv				
T/F Instant Bookable	清洗2_房源表.csv				
T/F Is Business Travel Ready	清洗2_房源表.csv				
Abc Cancellation Policy	清洗2_房源表.csv				
Abc 房源名称	清洗2_房源表.csv	# listing_id	评论文本.csv	字段名称	表
Abc 房源类型	清洗2_房源表.csv	# id	评论文本.csv	# listing_id	评论时间表.csv
Abc 价格	清洗2_房源表.csv	Abc date	评论文本.csv	Abc date	评论时间表.csv
Abc 人均床位	清洗2_房源表.csv	Abc comments	评论文本.csv		
# 人均卫生间	清洗2_房源表.csv				
Abc Neighbourhood	清洗2_房源表.csv				

图 字段展示

3. 数据清洗

观察各表数据后，发现存在重复值、缺失值与异常值，格式以及字段多余等问题。此外，一些字段的合并也不利于进一步的数据分析，需要将所对应字段进行拆分。下图详细地阐述了各表清洗的过程。



图 数据清洗主要工作梳理

对于评论时间表，数据清洗的过程主要是删除2017年以前的数据、修改时间格式；对于评论文本数据，数据清洗的过程主要是去重、筛选时间、删除无关字段如review_name；对于房东表，数据清洗的过程主要是删除缺失值和增加字段。房源表的数据清理则较为麻烦一点，主要包括去重、删除缺失值、检查异常值以及增加新字段等工作。

4. 探索性数据分析

4.1 所用工具

利用对xlsx和csv文件格式友好的spss数据分析软件和tableau数据可视化软件进行一些随意但有趣的、较为基础的描述统计探索（包括房东一览、价格探索、房源分布等），并针对有可能的实际应用情况，在预设故事背景下实现简单的案例实践功能。

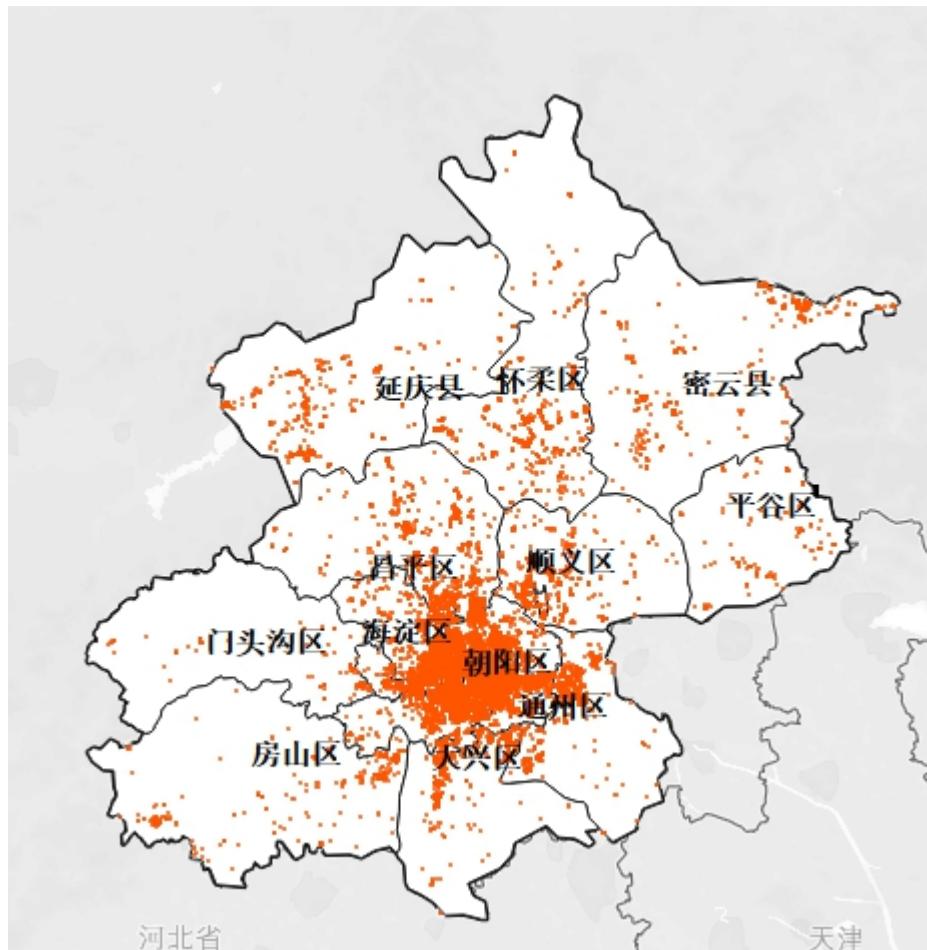
SPSS SPSS (Statistical Product and Service Solutions, 统计产品与服务解决方案) 为 IBM 公司推出的一系列用于统计学分析运算、数据挖掘、预测分析和决策支持任务的软件产品，社会科学领域常用。

Tableau Tableau Software 将数据运算与美观的图表结合，致力于帮助人们查看并理解数据，利用它可以实现快速分析、可视化并分享信息的目的。官网提供非常详细的使用手册（教程：Tableau Desktop 入门指南），对初学者友好，地址附后
(<https://help.tableau.com/current/guides/get-started-tutorial/zh-cn/get-started-tutorial-home.htm>)

(所用工具简介)

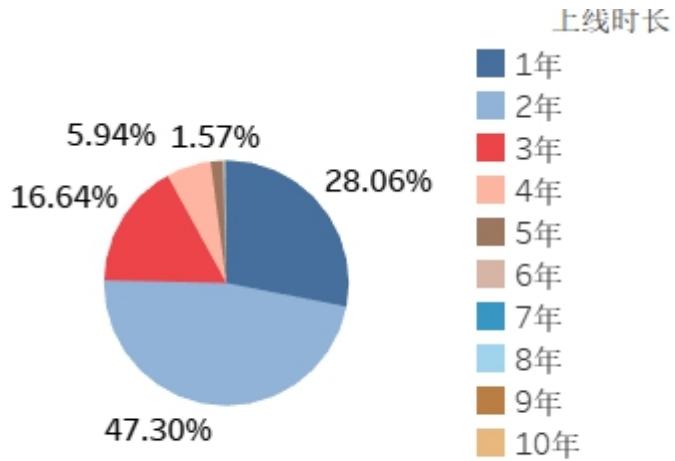
4.2 总体情况

4.2.1 房源区域密度



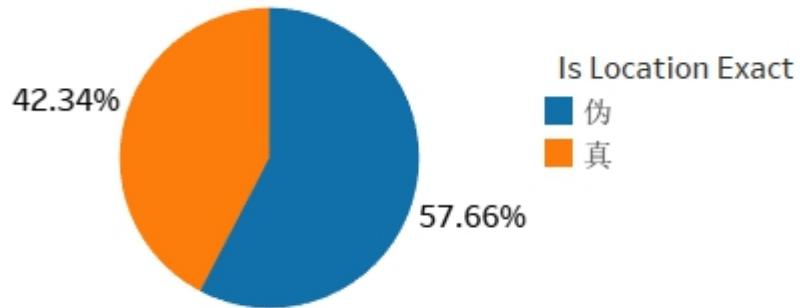
房源区域分布：- 房源集中在海淀、朝阳等中心地区，向外逐渐稀疏；基本沿着地铁线分布，房源密集度跟行政区景点密度成正比。

4.2.2 房源上线时间



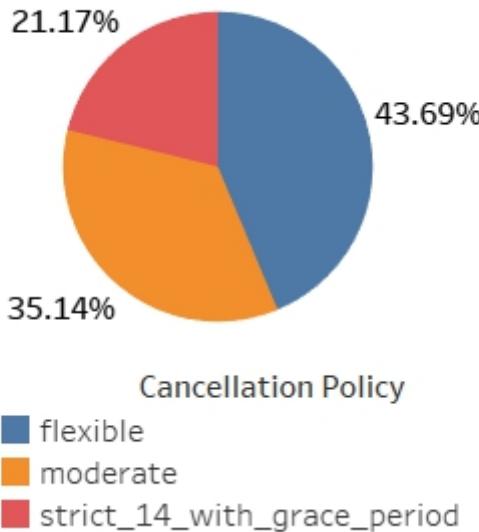
房源上线时间——根据最早的评论时间来看,房源上线时长集中在2年左右,极少有上线超过4年的老房源。

4.2.3 房源地址准确性



整体房源位置的准确度不高,可能与房东发布房源无需房源方面的认证有关,建议增加房源真实性的核实。

4.2.4 房源取消规则



整体取消政策比较宽松，若取消时距离入住时间至少还有 14 天，可免费取消预订。

4.3 什么样的房子热度高——房源热度分析

因无法获取房源的预订次数，由于评论数可以一定程度反映订单数和房源的热度，在此将房源的每月评论数量视为用户的需求。

4.3.1 评论数趋势（消费周期）

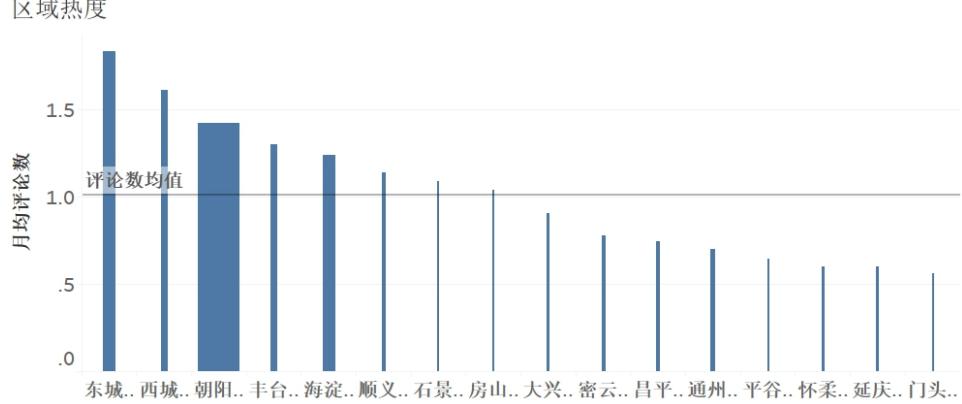


从 2017 年 1 月至 2019 年 4 月，airbnb 消费热度逐渐上升。



可以看出，订房热度呈现轻微的季节性，1-4月订房较多，5-7月相对淡季。

4.3.2 行政区域×热度



从房源的供给上看，朝阳区、东城区、海淀区三个区域的房源占了60%以上。

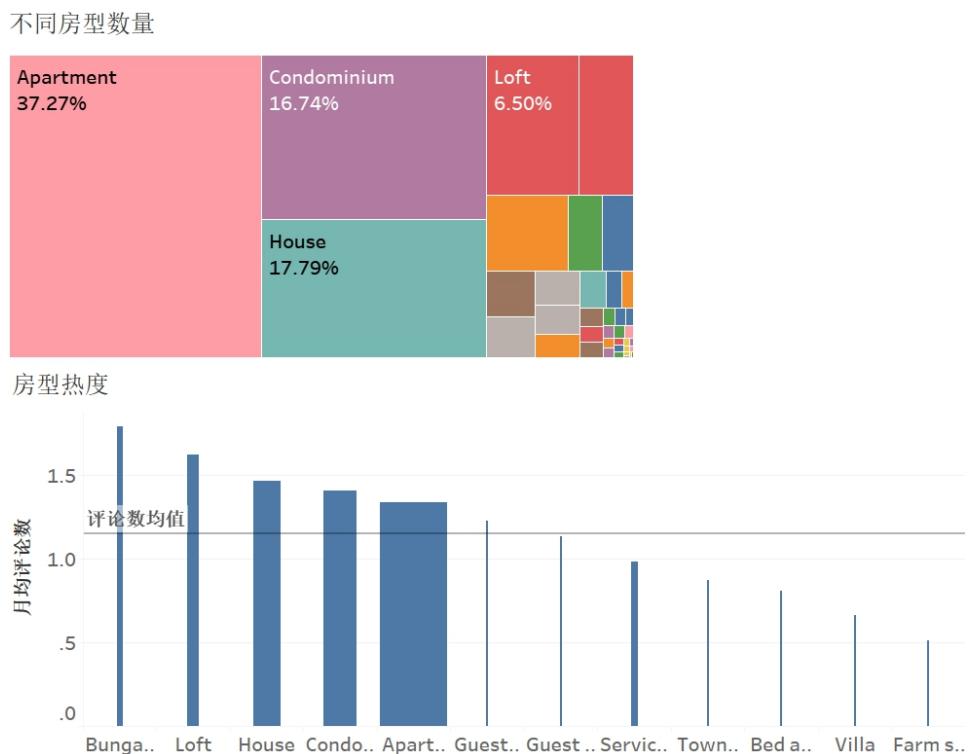
从需求上看，朝阳区、东城区、海淀区、丰台区、西城区、顺义区、房山区、石景山区热度高于均值。区域上整体供需比较匹配，西城区远高于整体水平，但房源供给低于平均水平，宜通过引导的方式，鼓励加大西城区的房源发布。

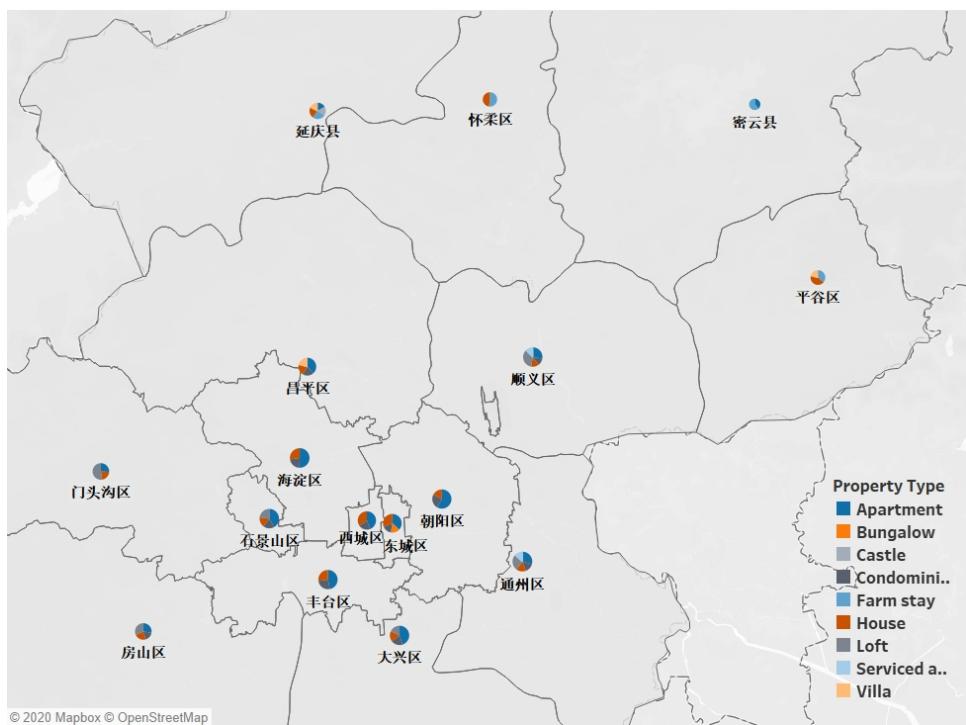
4.3.3 房间×热度

从总体来看，房间类型为整间的房源占比超过 60%，共享间（合住间）供给较少，租客对于房源的喜好顺序为：整间>单间>共享（合住）间。房源供需匹配。分区域看，各个行政区域的房间类型供给大多和整体一致。朝阳区 3 种房型的均远高于其他区域。

4.3.4 房型×热度

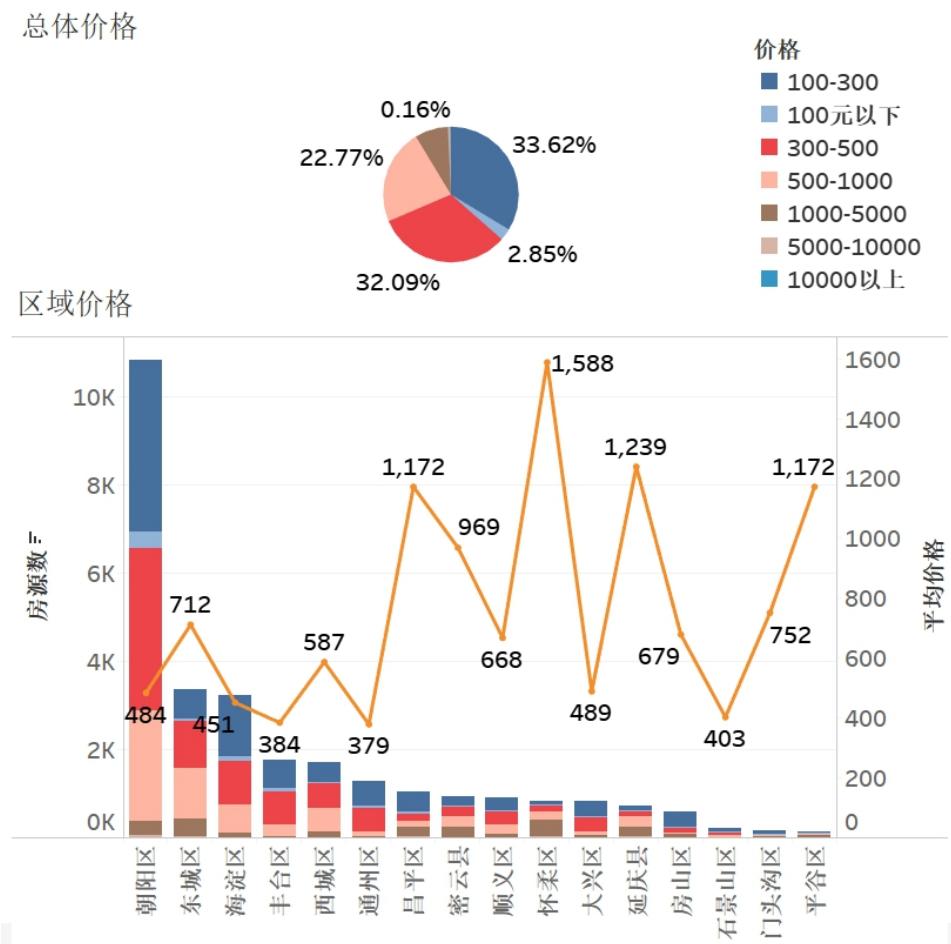
公寓，高档公寓，独栋房三种房型供给占了超过 70%。平房，Loft，独栋房，高档公寓，公寓，家庭旅馆等房型的热度远高于整体水平。平房、Loft、家庭旅馆房源供给少，热度高，建议鼓励增加这几类房源的发布。



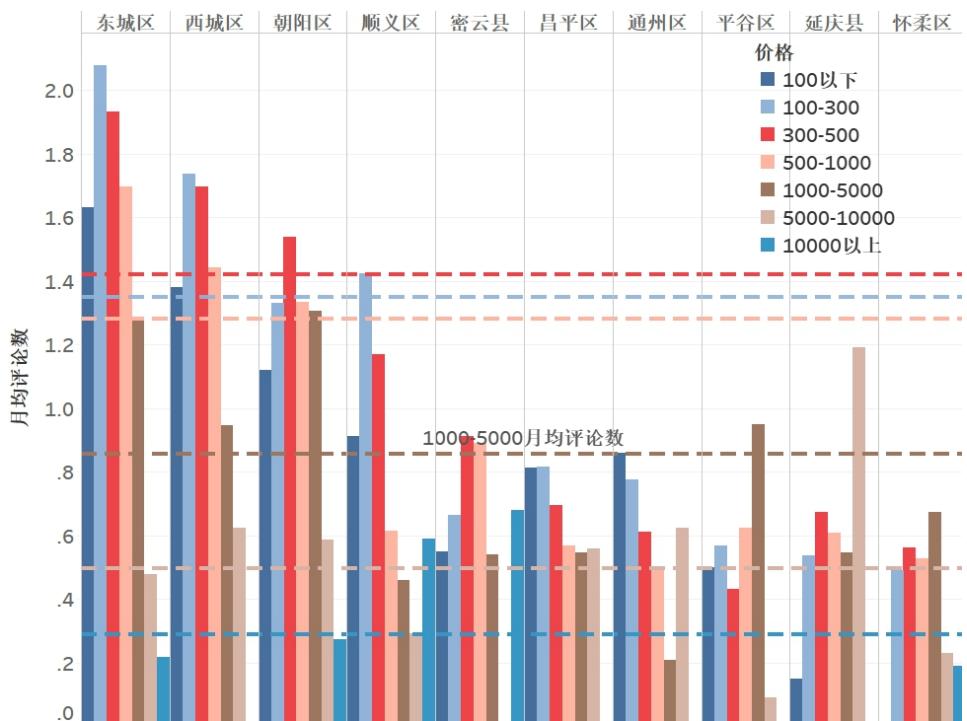


中心区域（海淀、朝阳、东城区等）主要以公寓，高档公寓，独栋房三种房型为主。周边区域（怀柔区、密云县、平谷区等）以农家乐和独栋房为主，带有一些别墅、城堡。

4.3.5 价格×热度



100-300 元和 300-500 元的房源占比超一半，1000 元以上房源占比仅为 8.7%，
500 元以下房子为 Airbnb 主力房源；昌平区、怀柔区、延庆县、平谷县等非中
心区域高价房源占比大，区域均价超过 1000



整体需求来看，300-500 元价位的房源热度最高，其他 100 元以上的房源，随着价格的升高房源需求降低。1000 元是房源热度的分水岭，性价比是租客选择房源的重要因素，高价房源不受青睐。

分片区看，东城区、西城区、朝阳区 1000 元以下的房源热度高。平谷，怀柔，延庆等地区高价房源热度较高。结合房源类型可以推测，中心区域租客是出于商务、购物、求医等场景，非中心片区租客更多是为了旅游、团建。平台可以在搜索房源页面增加租房目的选项（如：商旅出差、团建、旅游）更精准地为用户推荐房源

4.3.6 房源名



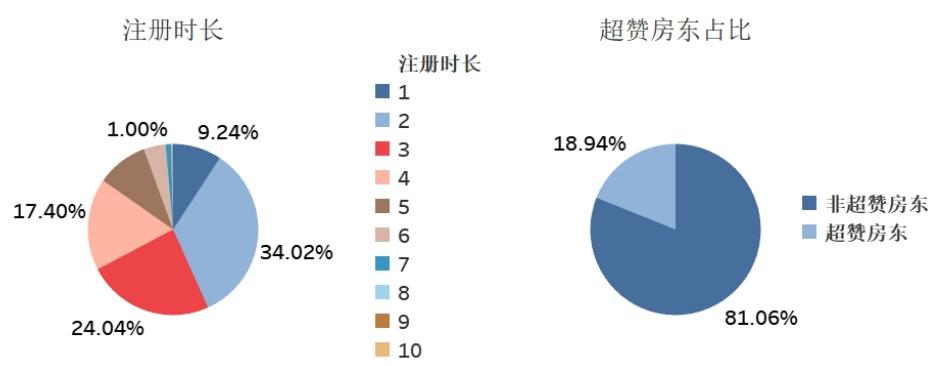
总的来看，房东设置房源名时倾向于标明：周边配套设施（尤其是地铁）、周边景区、房子类型、装修风格。



而月均评论较高房源，名称更多描述地点而非房源本身，其中，天安门，南锣鼓巷，三里屯和故宫热度远高于其他地方。租客倾向于能一眼看出地点的房源，建议根据房东提交的地理位置，添加智能标题建议。并且在租客房源检索筛选页面增加“1km 内是否有地铁”选项。

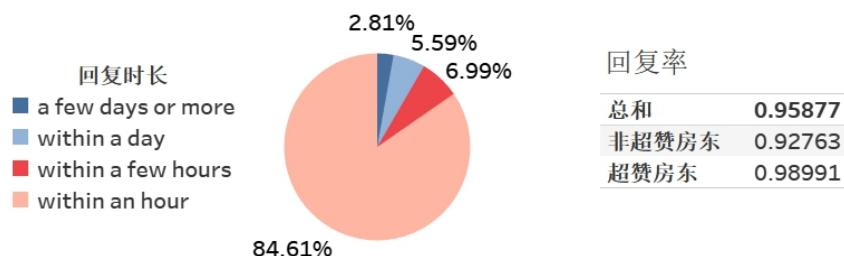
4.3 房东分析——谁是房东一哥

4.3.1 整体情况



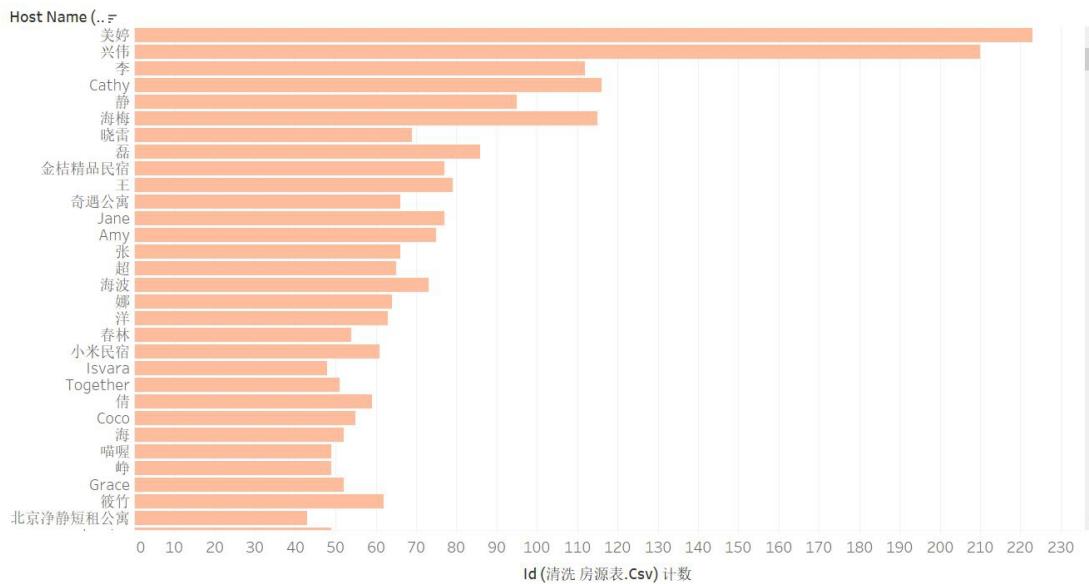
有发布房源的房东中，70%以上注册时长不超过 4 年，和房源上线时间一致。
注册时长 1 年的仅 9.24%，2-3 年占 50%以上，可以推测房东的存留情况不错。

超赞房东是指 “至少完成 10 次行程接待，或完成 3 次预订且总晚数达到 100 晚；回复率保持在 90% 或以上；预订取消率不超过 1%（即每 100 笔预订最多取消 1 次）；总体评分保持在 4.8 分或以上” 的优质房东。有发布房源的房东中，超赞房东和非超赞房东的比例约为 2:8，符合实际，可见超赞房东的评价标准比较科学合理。



超过 80% 的房东能在一小时内回复租客，整体回复率高达 95%，即使不是超赞房东回复率也有 92%。房东整体比较积极活跃。

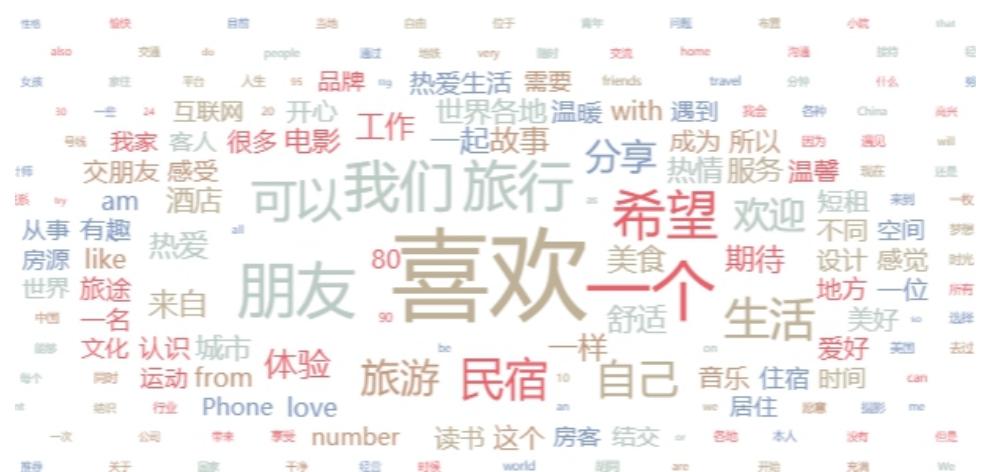
4.3.2 房东-房源



因为清洗后的房源表每个 ID 对应着一条数据，故将其用来计数。在 tableau 中将房东名作为行，数量作为列，并按从大到小的顺序显示；表中有的房东值为空（即不知道谁是房东），在显示时我们筛去 null 值，得到如图所示的结果。

可见，美婷和兴伟是当之无愧的北京市爱彼迎房哥房姐，两人拥有的房源数量都在 200 以上，甩开第三名一倍的距离。

4.3.3 房东特性



从图中可以看出，爱彼迎的房东兴趣爱好主要集中在：旅行、美食、运动、电影、音乐、读书。他们大都有自己的职业和爱好，喜欢和不同的文化背景的人交朋友，乐于分享自己的生活。身兼多职的房东，个人爱好和职业为房间附加一

定文化属性。在旅游新模式之下，房东对于房客不再是简单接待，除了优质的住宿环境，房东还能为房客带来什么，特色服务就显得尤为重要。每一间房都有一个故事，不管是房东个人经历故事还是房间打造过程的故事，都在一定程度上代表着房间的文化。平台宜鼓励有文化特色的房源发布，引导房东更多分享自己的故事，赋予民宿更多的文化属性，进而吸引房客预订。

4.3.4 入住规则



多数房东会对租客提出以下要求： 1. 保持屋内/厨房干净整洁。比如：退房时要倒垃圾/进屋时换鞋等 2. 限制抽烟/带客人/带宠物/限制出入时间 3. 爱护物品。造成物品损坏要赔偿或者扣除押金等 4. 不能打扰邻居/喧哗。建议平台在发布房源页面添加“是否能带客人”，“押金金额”等选项，方便房东发布房源。为部分房东/租客提供清理房间增值服务、继续完善邻里调解服务。

5. 如何找到适合用户的短租房——房源推荐挖掘

Airbnb发展至今天已经成为一个非常成熟的公寓平台，房源十分丰富，这给用户的房源选择带来了更多可能性，但同时也加剧了用户选择上的负担。为了做到效率的提升，同时尽可能满足用户的个性化体验，许多平台都引入了推荐算法。目前电商平台中主流的推荐算法包括协同过滤推荐算法，通过相同用户群体的不同选择进行商品的推荐，内容推荐算法，相似性算法等。在Airbnb的房源排序中应用到了嵌入的技术，在基于嵌入的解决方案中，相似房源是通过在房源嵌入空

间中找到K近邻来生成的。之后通过对用户历史性行为数据（用户过去点击的房源ID和用户过去跳过的房源ID）的分析达到个性化搜索的目的。

由于现有的数据并不包括用户的行为数据，因此无法通过用户的行为进行房源的分析，所以考虑通过关键词的提取，找出哪些符合用户需求的关键词，以此确定推荐房源。具有评价性信息的字段包括房东对房源的描述信息和过往往客对房源的评价，考虑通过这两方面进行推荐。

由此我们假定两个需求差异较大的用户小王和小埋，小王是一名底层的上班族，工资不高，但为人热情，希望结识各种朋友，某天被领导要求去北京出差三天（每日房补150元），出差的地点主要集中在北京朝阳区；小埋是一名家境优渥的留学生，艺术专业出身的她拥有着极高的审美，在美好的东西面前价格对她只是一些数字，十分看重生活的品质，放春假回国希望在北京玩15天，顺便可以见见高中时要好的朋友们（多数学校位于海淀区）。在对两位用户的基本信息有了了解之后可以确定二者的需求，根据过往经验，绝大多数用户在Airbnb中搜索时会设定的限制有价格、地区、房间类型和人数，这些限制条件也是前期做数据筛选时所需要的条件。可以简单总结二者的关键性需求特点，小王是热闹，而小埋是品质。

整个分析通过Python进行实现，代码在jupyter notebook中书写，以截图的方式展示。用到的包如下：

包	功能
pandas	csv 的读写以及科学计算
numpy	科学计算
seaborn	可视化
Matplotlib.pyplot	可视化
warnings	忽视报错信息
jieba	中文分词
jieba.analyse	TF-IDF 关键词的提取

分析用到了四个数据集，其中涉及到的数据如下：

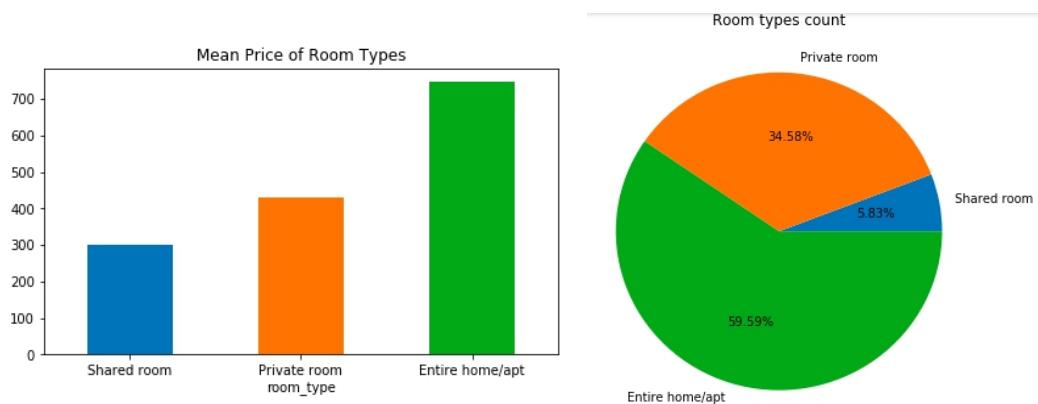
数据表	含义	代码缩写	字段
-----	----	------	----

listings	房源列表信息	listings	Id, neighbourhood, room_type...
listings_detail	房源列表详细版	ld	room_id, url, summary...
reviews_detail	评论文本	rd	listings_id, comments
reviews	评论时间	c	listings_id, date

整个分析的思路就是根据基本的信息，作出房源的筛选，包括地区、价格、房型等。在符合条件的房源中根据每月的评论数量、最近评论日期和总评论数等判断房源是否为热门房源。最后读取这些房源的文本信息，利用TF-IDF计算其关键词，根据关键词作出第二次筛选，观察其他信息和房源图片，作出最终选择。

5.1 推荐实例（一）

通过小埋的信息，在数据的描述性统计中已经了解到均价最贵的房间类型是 Entire home/apt，且其在数量上也是最多的，而且一般独享整个房源会有更好的居住体验，这也是小埋所追求的，因此限制房间类型为 Entire home/apt。



```

data1 = listings.groupby(['neighbourhood','room_type']).agg({'id':'size','price':'mean'})
data1 = data1.rename(columns={'id':'number'})
data_1 = data1.unstack()['number']
data_1 = data_1.sum(0).sort_values()
plt.title('Room types count')
data_1.plot.pie(figsize=(6,6), autopct='%.2f%%')

```

房源质量与价格成正比，如果想兼顾美学与体验，必须选择足够贵的房子，因此选定价格为所有价格的90%分位数，该数字十分接近于1000，因此设定价格为1000元。同时，为了方便与朋友聚会，设置房源所在地区为海淀区。

小埋要在北京待15天，这就要求房源最小可居住天数小于15，最大可居住天数大于15。以上是第一次筛选的条件，代码如下：

```
data = listings[listings['neighbourhood']=='海淀区']
data = data[data['price']>=1000]
data = data[(data['availability_365']>=15)&(data['minimum_nights']<=15)]
data = data[data['room_type']=='Entire room/apt']
data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 116 entries, 23 to 28298
Data columns (total 16 columns):
```

共筛选出116条数据，对于房源推荐来说仍然有点多，以总评论数做为房源热度考量的标准，选出评论数最多的前十名房源，作为关键词选择的对象。

```
data10 = data.sort_values(by='number_of_reviews',ascending=False)[:10]
import jieba.analyse as analyse
```

遍历筛选出的十条数据的id，从listings_detail表中提取出这十条数据，从房东对房源的描述出发进行关键词提取。对summary字段分词，并串接字符串，通过analyse包进行基于TF-IDF算法的关键词提取。在实验中分别选择了基于extract_tags和TextRank的主题词，其中sentence为待提取的文本，topK为返回n个TF-IDF权重最大的关键词，withWeight为是否同时返回关键词权重值，此处并不需要，allowPOS为仅包括指定词性的词，为了方便筛选，设定关键词词性为形容词。

```
for i in data10['id']:
    print('Id:',i)
    b = Id[Id['id'] == i]
    lines = b['summary'].values.tolist() # 得文本分词列表
    content = ''.join(lines) # 串接字符串

    words1 = ' '.join(analyse.extract_tags(content, topK=20, withWeight=False, allowPOS=('adj')))
    print('主题词：\n',words1)# 提取主题词
```

```

id: 12654145
主题词:
 仍为 巨大 有效 就是
id: 15209963
主题词:
 宽敞 繁多 不少
id: 7852881
主题词:
 大自然 自然
id: 13873843
主题词:
 保洁 分别
id: 21110486
主题词:
 当然
id: 22537483
主题词:

id: 11990102
主题词:

id: 25115794
主题词:
 简约 惊喜 愉快 优秀 特别
id: 20049185
主题词:
 舒心 幽静

```

对summary字段提取关键词并不顺利，关键词价值不高，且数量过少，还混杂了包括“当然”、“分别”在内的无法帮助分析的词。这可能是因为summary字段包含的文本量不多，导致没有可分析的东西，抛弃对房源描述信息文本挖掘的思路，选择文本量更大的评论数据。

与上述步骤类似，但在分词之前需要对评论数据中空缺值采用dropna()语句进行删除。之后选择评论信息数据comments字段，仍用extract_tags提取关键词：

```

for i in data10['id']:
    print('id:',i)
    a = rd[rd['listing_id'] == i]
    lines = a['comments'].values.tolist() # 得文本分词列表
    content = ''.join(lines) # 串接成字符串

    # 提取主题词
    words1 = ''.join(analyse.extract_tags(content, topK=20, withWeight=False, allowPOS=('adj'))))
    print('主题词: \n',words1)

```

图 代码过程

id: 762579
 主题词:
 非常 很大 方便 舒适 宽敞 不错 便利 相當 干净 整洁 美好 愉快 就是 特別 親切 寬敞 舒適 不錯 很棒 再次
 id: 12654145
 主题词:
 非常 干净 特別 舒适 愉快 很棒 舒服 慨意 不错 很好 挺好 很大 整洁 踏实 友好 遗憾 便利 到来 微冷 常乐
 id: 15209963
 主题词:
 非常 方便 特別 真的 很棒 干净 很大 不错 十足 惊喜 不远 意外 很爽 親切 巨大 真正 随处 私密 有福 比较
 id: 7852881
 主题词:
 非常 干净 整洁 方便 舒适 愉快 慨意 不错 真的 不远 充足 特別 宽敞 真是 精致 惊喜 耐心 整齐 舒服 完美
 id: 13873843
 主题词:
 舒適 方便 干淨 不錯 要好看 非常 寬敞 浪漫 完美 便利 就是 真是 很大 一般
 id: 21110486
 主题词:
 不錯 干淨 不冷 舒暢 整洁 惊喜 舒适 整齐 舒服 亲切 就是 不远 准確 正好 方便 向上 刚刚 不足 很快 真的
 id: 22537483
 主题词:
 非常 不錯 很大 方便 干淨 真的 很棒 特別 顺畅 精致 一共 舒服 完美 便利 快速 比較
 id: 11990102
 主题词:
 非常 很棒 简约 整洁 真心 友好 精致 耐心 有趣 舒服 日常 干净 幸福 好好 方便 快速 真的 容易
 id: 25115794
 主题词:
 非常 干净 方便 不错 真的 很好 很棒 很大 有幸 宽敞 细致 稍微 有趣 明亮 遗憾 赶上 舒服 愉快 亲切 便利
 id: 20049185
 主题词:
 非常 特別 被套 舒服 小次 妥妥 多定 宽亮 简约 乖巧 清爽 出奇 幸运 刚好 舒适 一共 遗憾 高低 清洁 热闹

在对评论文本进行关键词权重的计算后，关键词选择的结果非常好，逐行查看关键词：第二组含有“遗憾”“微冷”，第五组包含“一般”，第六组包含“不足”，第九组包含“遗憾”，第十组包含“小次”“遗憾”，删除这些数据。选择id为762579, 15209963, 7852881, 22537483, 11990102，查看剩余数据的url，name, space, transit等信息：

	listing_url	name	space	summary	transit	accommodates	bathrooms	bed_type	price	reviews_per_month
23	https://www.airbnb.com/rooms/762579	3BR Luxury Apt in Downtown Beijing	Hi there, I am ShenJie, nice to "meet" you! I ...	I am sweet home (in Beijing)	Buses: 16, 26, 27, 30, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099, 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1109, 1110, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1129, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158, 1159, 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1168, 1169, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1198, 1199, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1229, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1239, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1248, 1249, 1249, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1268, 1269, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1278, 1279, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1288, 1289, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1297, 1298, 1298, 1299, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318, 1319, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1358, 1359, 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1398, 1399, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1418, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1478, 1479, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1488, 1489, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1498, 1498, 1499, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 1559, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1577, 1578, 1579, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1588, 1589, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597, 1598, 1598, 1599, 1599, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1609, 1609, 1610, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1618, 1618, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626, 1627, 1628, 1629, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1639, 1639, 1640, 1641, 1642, 1643, 1644, 1645, 1646, 1647, 1648, 1649, 1649, 1650, 1651, 1652, 1653, 1654, 1655, 1656, 1657, 1658, 1659, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667, 1668, 1669, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1679, 1680, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1697, 1698, 1698, 1699, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 170					

1899	https://www.airbnb.com/rooms/15209963	到谷仓酒店去体验悠闲的生活	180 square meters LOFT, 6 meters high, the spa...	民宿空间宽敞,几颗小树木和盆栽绿意盎然,书柜上的书籍玲珑满目,种类繁多,小巧别致的泥模型也有...	Can also provide Airport Transportation and ot...	6	1.0	Real Bed	\$1,899.00	1.10
7240	https://www.airbnb.com/rooms/22537483	『多床房』魏公村/中关村/颐和园/动物园/什刹海/香山/电视台/地铁4号线	户型-复式四居室风格-创意设计,每个房间设计风格不同,结合整套房屋设计理念,分别进行细...	公交:地铁线路4.9号线,在魏公村或国家图书馆下车,步行至韦伯豪家园大约1500米打车:直...	云来~云往~云宿~	12	2.0	Real Bed	\$1,798.00	0.64

Id 为 11990102 的房源可住人数仅为 3，且月均评论数为 0.4，删除该数据，通过所给 url 查看剩余数据的照片。最后选择人气最高，且观感最优的房源，id 为 15209963。该房源评分达到了满分 5.0，同时房东也是“超赞房东”，如果租期超过一周，房东还可以在原价的基础上给予 35% 的价格减免。



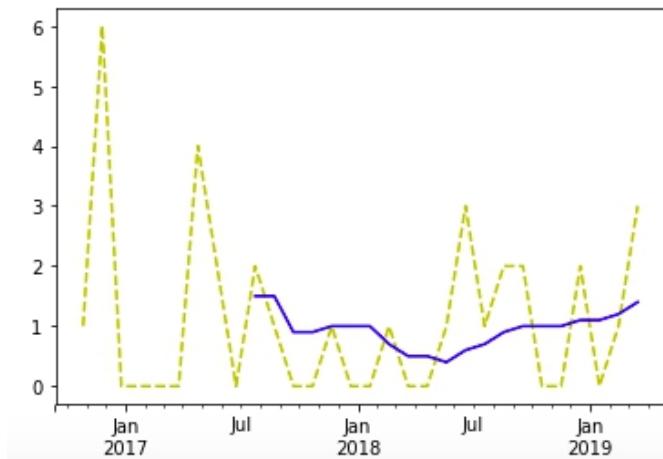
图 推荐房源图

为了加深对该房源热度的判断，引入评论时间序列，绘制每月评论数量曲线。在高价房源中，该房源月均评论数总体较高，但还是随着季节的更替而波动，在秋冬季节游人不多时，房源评论较少，夏季房源评论相比冬季高出很多。总体趋势比较平稳，有上升的迹象。

```

c=pd.read_csv('/Users/mac/Desktop/Practise/Python/reviews.csv')
choose=c[c['listing_id']==15209963]
new_date=pd.to_datetime(choose.date.values)
new=pd.Series([1]*choose.shape[0],index=new_date)
count_review=new.resample('30D').sum()
count_review.plot(style='y--')
count_review_window=count_review.rolling(window=10).mean()
count_review_window.plot(style='b')
plt.show()

```



在该房源下方的相似推荐中，第一条便是上述选择结果中一条，这也能从侧面印证关键词的挖掘结果与 Airbnb 官方嵌入技术 k 近邻获得的相似房源十分接近。



5.2 推荐实例（二）

小王的居住地区需要设定在朝阳区，接受最高的价格为 150，因为只住三天，同时需要年可住天数大于等于 3，最低可住天数小于等于 3。考虑到小王是个喜欢热闹，交朋友的人，共享房源更有这样的机会，同时该房源的平均价格也较低，因此设置房型为 shared room。最终筛选出 187 条数据。

```
data = listings[listings['neighbourhood']=='朝阳区 / Chaoyang']
data = data[data['price']<=150]
data = data[(data['availability_365']>=3)&(data['minimum_nights']<=3)]
data = data[data['room_type']=='Shared room']
data = data[data['last_review']>='2019-02-01']
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 187 entries, 278 to 28119
```

图 代码过程

重复推荐实例 1 中的步骤，筛选 10 条最热门房源，并删除空缺评论信息。

对评论文本分词，并基于 textrank 计算关键词权重。筛选出的关键词如下：

主题词：
非常 不错 房客 就是 有趣 特别 方便 很好 很棒 真的 干净 不远 比较 果然 愉快 重要 整洁 难搞 很酷 一小
id: 16035699
主题词：
非常 不错 方便 比较 有趣 便利 舒适 干净 愉快 极其 简约 深刻 配套 当然 唯美 更加 很大 就是 稳重 成熟
id: 16694287
主题词：
非常 干净 真的 就是 很棒 不错 舒服 合适 比较 便利 最好 方便 惊喜 纤丽 配套 很大 本当 不难 其实 特别
id: 17813168
主题词：
非常 方便 不错 舒服 有趣 干净 一幅 特别 很好 整洁 柔软 相当 首先 青旅 很棒 比较 很大 房客 温馨 便利
id: 15897421
主题词：
真的 特别 非常 央美 方便 有趣 比较 干净 很棒 最好 就是 房客 漂亮 友善 热闹 热闹 乐观 不错 毕竟 完美
id: 17290581
主题词：
非常 就是 真的 漂亮 有趣 特别 比较 舒服 相处 自由 惊喜 方便 友好 善良 房客 温柔 整洁 反正 幽默 简直
id: 14792340
主题词：
非常 方便 真的 特别 有趣 就是 房客 很棒 比较 不错 便利 绝对 整洁 央美 随机 毕竟 敏感 随意 激动 已经
id: 22296264
主题词：
非常 干净 特别 真的 四惠 方便 舒服 就是 很棒 不错 最好 舒适 便利 漂亮 首先 整洁 友善 优异 有幸 终点
id: 21255132
主题词：
非常 干净 舒适 特别 不错 便利 青旅 友善 一定 分别 方便 房客 暖和 整洁 绝对 遗憾 早点 快乐 一小 很棒
id: 16695169
主题词：
特别 干净 比较 舒适 非常 方便 親切 友善 爱笑 越来越 精明 親為 就是 日常 勤劳 愉快 舒服 整洁 真的 青旅

根据“热闹”、“青旅”等关键词进行选择，最终选择出

17813168, 15897421, 16694287, 16695169 这四条符合用户需求的记录，其中
16694287 和 16695169 两处房源其实是一栋房子中的两间，均属于同一房东。同

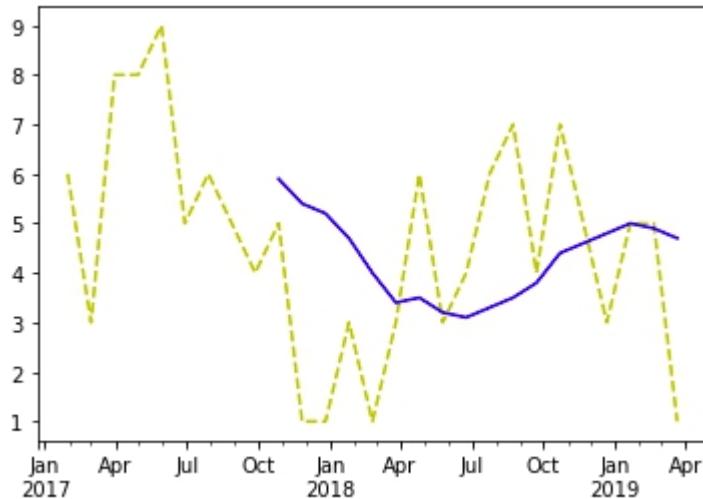
时查看其他房源的其他信息：

	listing_url	name	space	summary	transit	accommodes	bathrooms	bed_type	price	reviews_per_month
2180	https://www.airbnb.com/rooms/15897421	Dark room	【优势一】交通非常便利,出门就是地铁站14号线望京南,40分钟直达北京南火车站,去798,中...	(近期都有房哦,长租的宝宝看过来~具体的可以直接问我,日历的床位显示信息不一定准确哈)...	望京南地铁站14号线a口出,穿过马路即使我家,北站,14号线直达40分钟。而且家...	5	2.0	Real Bed	\$87.00	3.72
2552	https://www.airbnb.com/rooms/16694287	Superman Room for 4 person	This room charged bed by bed. The price of one...	2 bunk beds for 4 persons. We offer our guests...	交通指南:从机场坐地铁到五号线和平西桥站或者打车直接来我们这里都行,地铁花费一小时左右30元...	4	2.0	Real Bed	\$87.00	4.60
2555	https://www.airbnb.com/rooms/16695169	French Style Room for 6 person with balcony	We provide to our guests; Free high speed Wi-Fi...	3 bunk beds for 6 person. We provide to our gu...	共享单车、自行车出行方便	6	2.0	Real Bed	\$87.00	3.46

16694287 号房源交通便利，方便小王进行工作，同时也是这四处房源中月评论数量、总评论数量最多的，价格目前只要 80 元一晚，非常符合小王的要求。网页中展示的房源照片也能看出来房间比较舒适、干净。



同样绘制该房源的评论时间序列曲线，该房源评论呈明显的季节性波动，夏季为高峰，冬季为低谷。房源月均评论数表现突出，高峰时可达到 9 条，尽管最近月均评论数较过去有所下滑，但仍然维持在较高的水平。



6. 什么样的房子更贵——价格预测分析

6.1 影响价格的因素探索

在这一部分我们主要想弄清楚的问题是——什么样的房子更贵？不妨从房型、大小、内置家具，地理位置、地址准确度，是否立刻可订与退订政策这三个方面进行探讨。

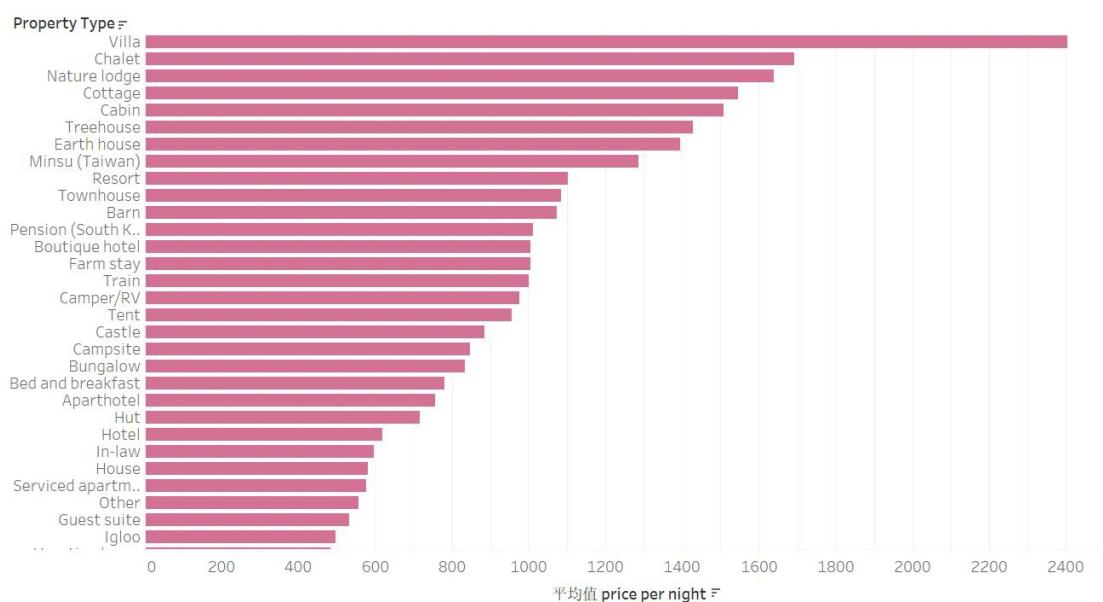
浏览官网提供的数据，我们发现部分房源存在最小入住天数的问题，即，有的房子的租赁政策是顾客至少住够指定天数，此时它标注的价格一栏实际上是每晚单价*最小入住天数后得出的总价，而为了让价格处于统一跑线上，在开始之前，我们需要新建一个计算字段，写一个简单的除法公式算出每晚的价格（注：右侧可见函数导览）。

price per night ×

[Price] / [Minimum Nights]

计算有效。 2 依赖项 应用 确定

设置好后便可开始我们的价格探索环节。

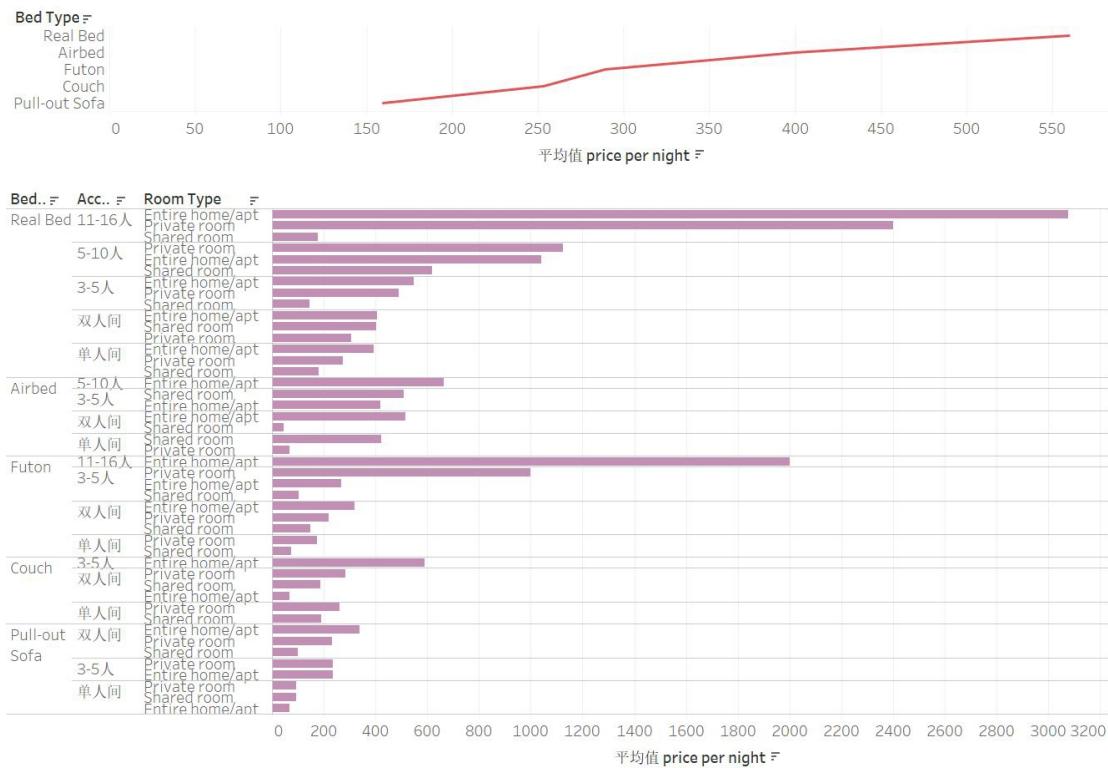


对每种房型的每晚价格求平均值，发现最贵的是度假别墅（Villa）、度假用木屋（Chalet）、自然旅社（Nature lodge）等，住一晚可能需要花上千元，这些房子常坐落在乡间或山野的空旷处，与自然相亲近且占地面积大；而酒店式公寓（Serviced apartment）、客房套房（Guest suite）的价格则比较便宜，主城区每晚均价在 600 元以下。



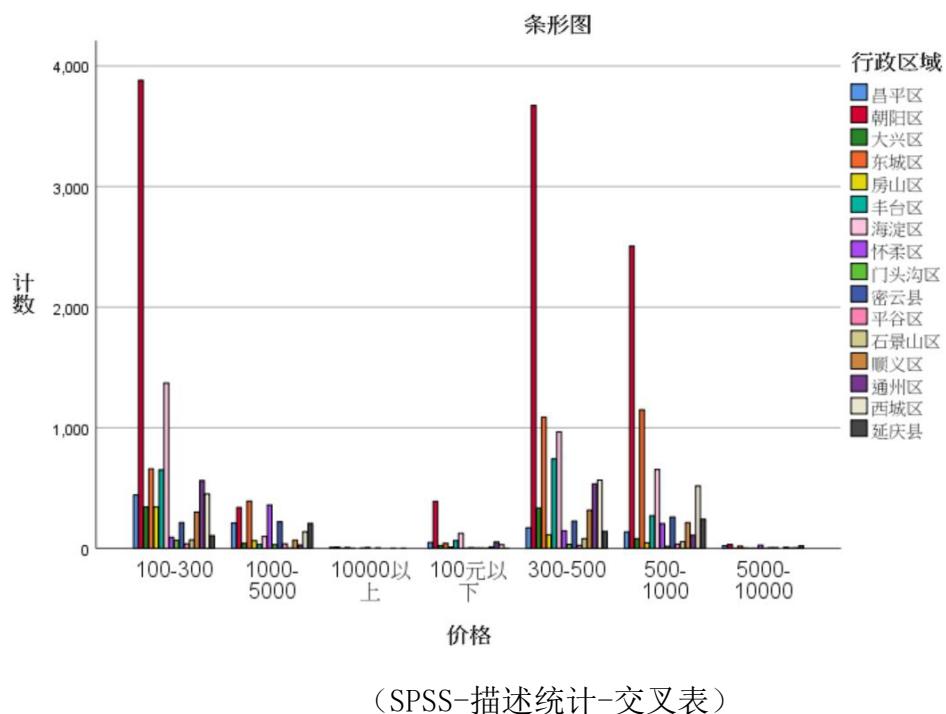
(nature lodge)

上面说的是整体房子的类型，接着让我们细化来看看具体的房间类型、容纳人数与家具情况和价格的关系。



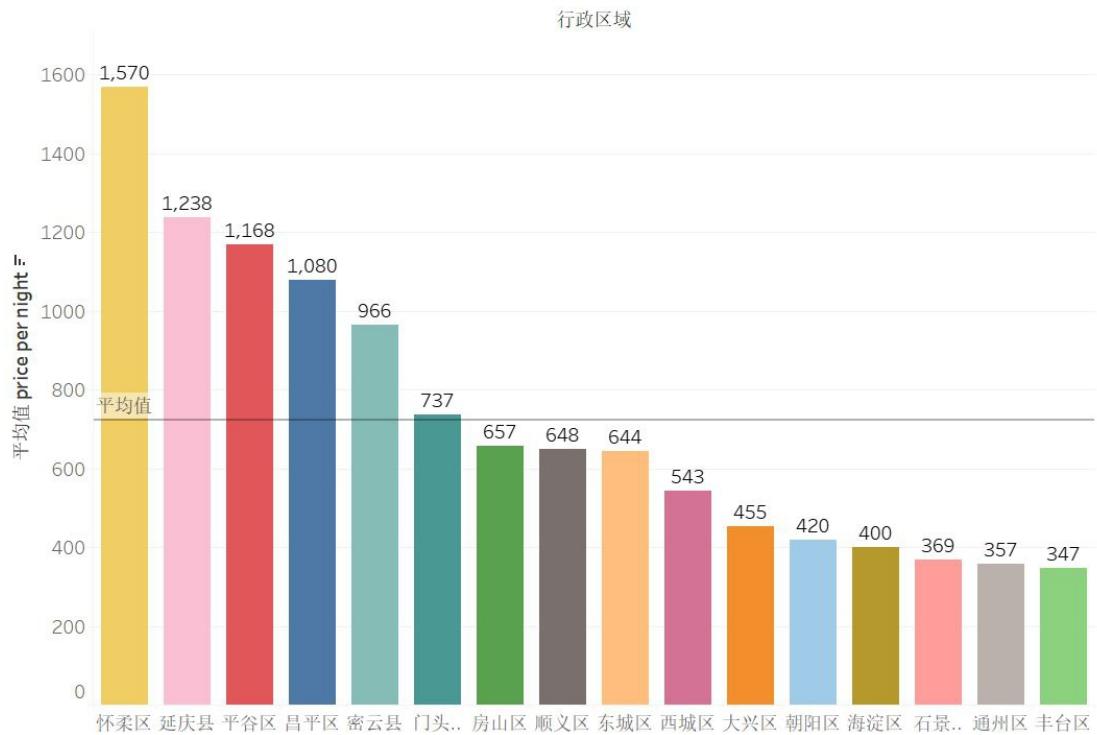
我们发现，在床的类型上，真正的床、气垫床、日本床垫（折叠时可坐，铺开时可卧）、沙发、沙发床（可将沙发垫翻出用作床）价格依次递减；总体而言房间是可供入住的人数越多越贵（想想看轰趴别墅便不难理解）；整间房贵于单间房贵于共享间。但也不乏异常值，例如配置有日本床垫的、可住 11-16 人的整间房，床的类型固然是重要的一环，被官方单独作为一列考虑，但其影响并不绝对，上述的异常可能是由装配着蒲团作为特色的日式别墅贡献的。

在地理位置与地址准确度方面，我们也发现了一些有趣的现象。



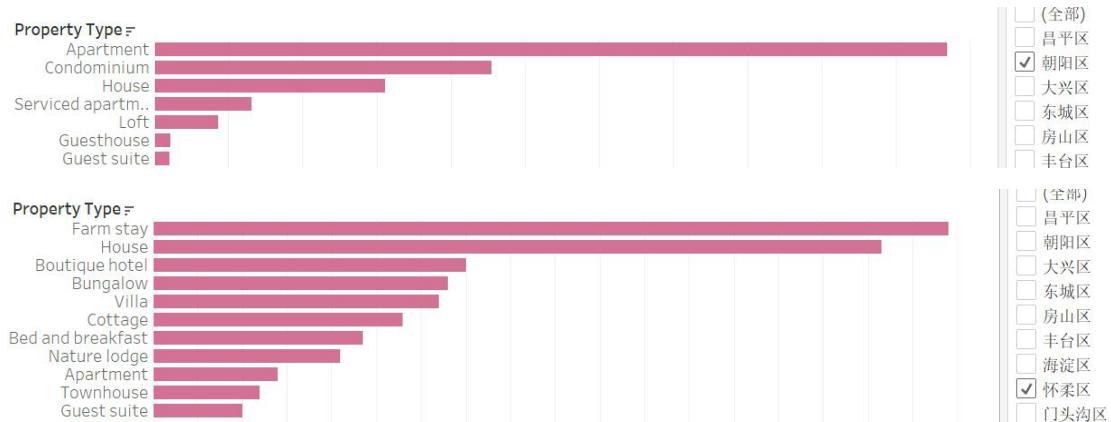
(SPSS-描述统计-交叉表)

不难看出，朝阳区的房源是最为充裕的，且价格区间集中在 100-300 元，300-500 元和 500-1000 元之前，总体而言比较低廉；东城区的房价则是以 500-1000 元为多。这张图本身包含了很多信息，但是对于反映各行政区的房源均价差异还是不够直观，为此我们做了下表。



(将行政区域拖到“颜色”中，并在分析页面添加平均线)

处在房价平均线处的是门头沟区，而房价均值最高的是怀柔区。这引起了我们的关注，因为怀柔区地处北京东北六环以外，地租其实相对市区/北京城中心是比较便宜的，为何房源价格却居高不下呢？我们把区域作为筛选器，将怀柔区与上文所述均价较低房源较为充裕的朝阳区进行对比。



(二者的纵轴计数不相同，前者是千后者是百；条状的长度只代表在本区内数量的相对多少)

朝阳区的房型以公寓、住宅公寓（Condominium）为主，而怀柔区则以收费昂贵（见前文房型统计表）的农场住宿（Farm stay）、精品酒店（Boutique hotel）为主。或许正是因为怀柔区地处偏远，才能有地方并以合适的地租发展农家乐产

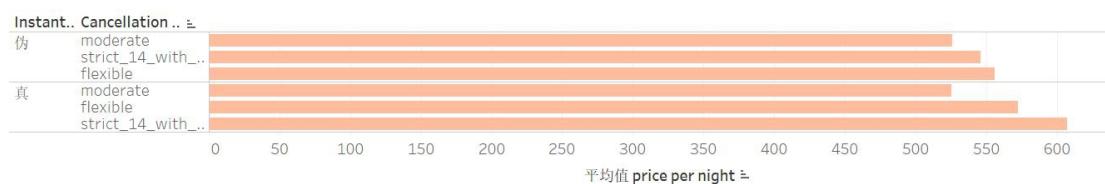
业，这其实帮助我们窥见了不同城中-城郊的不同租房趋势、导向与生态。另一方面，怀柔区房源较为稀疏，小的样本量加上部分房型过高的房价可能会导致算出的数据出现极端化倾向。

在地址填写准确度方面，大部分房源都是不准确的，这与我们现实生活中的现象和体悟是大致重合的（笔者便曾多次有找不到房子在哪里的经历）。且地址精准度与价格之间关系不明显，如图所示，准确度最高的是 300-500 元价格区间的房源。



（蓝色代表 f/false 即不准确， 橙色代表 t/true 即准确）

最后大致看一退订政策的内容。即刻可订、严格遵循 14 天退订宽限期的房源每晚均价略高，但整体看来影响力度较小，订房相关内容对房价的左右有限。



6.2 价格预测模型

该部分的主要思路是计算各因素与价格的相关性，进而通过逻辑回归求出相关因素的影响程度系数，本来想采用SPSS这个之前熟悉的工具进行分析，但是由

于数据量过大，SPSS过于卡顿，故选择Python作为主要分析工具。首先是相关性分析，此处主要使用了pandas统计函数[corr]，从下方结果可以明显发现房源所属城区、房间类型影响着房源的价格。

```
corr_matrix=df_list.corr()  
corr_matrix['price'].sort_values(ascending=False)
```

price	1.000000
Entire home/apt	0.469738
latitude	0.237862
怀柔区	0.194351
延庆县	0.137348
东城区	0.126049
密云县	0.092883
availability_365	0.058505
host_id	0.046889
西城区	0.045494
calculated_host_listings_count	0.040781
平谷区	0.036569
昌平区	0.015875
longitude	0.009830
门头沟区	0.007755
顺义区	0.007408
id	-0.001202
石景山区	-0.009798
房山区	-0.038302
大兴区	-0.041452
minimum_nights	-0.042272
丰台区	-0.065966
number_of_reviews	-0.077208
海淀区	-0.081479
reviews_per_month	-0.083035
通州区	-0.083052
朝阳区	-0.118807
Private room	-0.299270
Shared room	-0.381356

Name: price, dtype: float64

进一步进行逻辑回归分析，准备建模。此处主要是通过Xgboost建模，该算法的思想是不断地添加树，不断进行特征分类来生长一棵树，每添加一个树就是

学习一个新函数，去你和上次预测的残差。首先调用板块并准备建模数据，设y为价格，其他因素为x。

```
import xgboost as xgb
from sklearn import svm
from sklearn.linear_model import LogisticRegression # 逻辑回归
from sklearn.metrics import mean_squared_error,r2_score
# 准备建模数据
drop_list = ['last_review']
df_list=df_list.drop('last_review',axis=1)
y=df_list['price']
x=df_list.iloc[:,7:]
x=x.values
y=y.values
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=1/3,random_state=10)
df_list.iloc[:,7:].head()
```

以下为所得到的结果展示：

minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365	东城区	丰台区	大兴区	密云区	平谷区	.朝阳区	海淀区	石景山区	西城区	通州区	门头沟区	顺义区	Entire home /apt	Private room	Shared room
0	0.000000	0.276398	0.0425	0.036199	0.934 066	0	0	0	0	0	..	1	0	0	0	0	1	0	
2	0.005495	0.804348	0.1350	0.000000	0.252 747	1	0	0	0	0	..	0	0	0	0	0	1	0	
3	0.000000	0.080745	0.0140	0.018100	0.793 956	1	0	0	0	0	..	0	0	0	0	0	1	0	
4	0.000000	0.114907	0.0200	0.018100	0.964 286	0	0	0	0	0	..	1	0	0	0	0	1	0	

进一步测试模型，首先分割训练数据和测试数据，输出R方和MSE均方误差，具体代码如下。

```
def select_SVR(name,number_c,number_gamma):
    svr = svm.SVR(kernel=name, C=number_c, gamma=number_gamma)
    svr.fit(x_train, y_train)

    y_train_pred = svr.predict(x_train)
    y_test_pred = svr.predict(x_test)

    print('MSE train: %.3f, test: %.3f' % (mean_squared_error(y_train,y_train_pred),
                                            mean_squared_error(y_test,y_test_pred)))

    print('R方 train: %.3f, test: %.3f' % (r2_score(y_train,y_train_pred),
                                            r2_score(y_test,y_test_pred)))
select_SVR('rbf',100,0.1) # rbf 径向基
```

- R 方 train: 0.459, test: 0.439
- MSE train: 0.012, test: 0.013

但是输出结果中，我们发现训练集的R方值并不高，说明数据集特征并不能很好地预测价格，所以继续尝试参数调优。在尝试过程中发现，gamma小于等于0.1，C大于等于10的效果都很好。

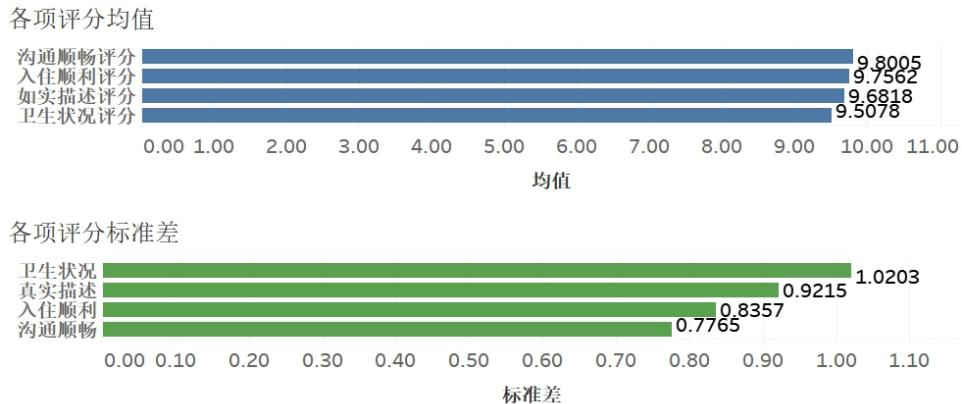
```
model=xgb.XGBRegressor()
model.fit(x_train,y_train)
y_train_pred = model.predict(x_train)
y_test_pred = model.predict(x_test)
print('MSE train: %.3f, test: %.3f' % (mean_squared_error(y_train,y_train_pred),
                                         mean_squared_error(y_test,y_test_pred)))

print('R方 train: %.3f, test: %.3f' % (r2_score(y_train,y_train_pred),
                                         r2_score(y_test,y_test_pred)))
```

MSE train: 0.012, test: 0.012
R 方 train: 0.470, test: 0.471

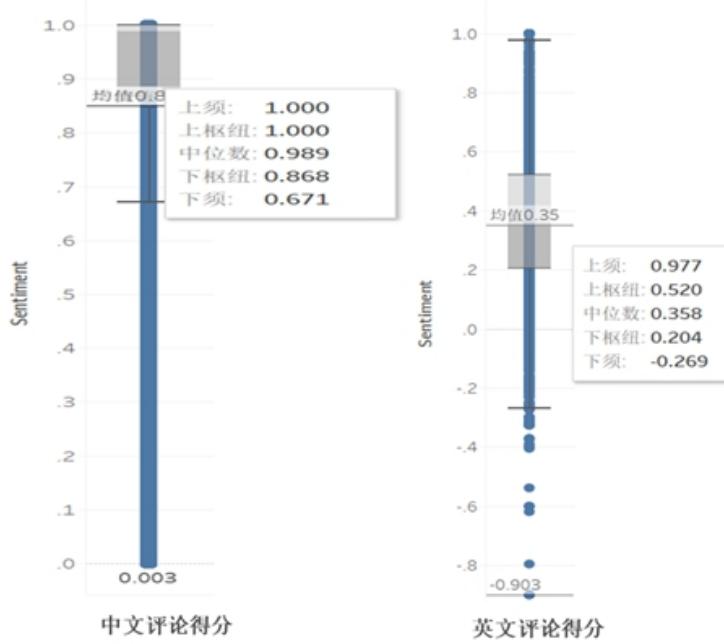
7. 用户对房源的评价如何——评论分析

7.1 各项评分情况



从整体分数来看，消费者对于卫生状况满意度最低。从评价指标的标准差来看，卫生状况分数差异最大。平台有必要督促房东做改善房源的卫生状况，长期干净卫生分数低的房源，设定限制发布房源。

7.2 评论情感分析



分离中英文评论，用 SnowNLP 快速进行评论数据情感分析。情感极性的变化范围是 $[-1, 1]$ ，-1 代表完全负面，1 代表完全正面。中文评论整体情感正面，租客满意度高。中英文评论得分都较为分散，有较多异常点。



从整体上看，游客的体验感知评价维度主要包括：房源质量、周围环境、房东评价及个性化体验四个方面。

游客对在线短租房源的硬件设施、装修风格、卫生和安全方面评价较多。在周围环境方面，游客尤其注重房源选址附近的交通可达性及周边生活的便捷性。关于房东评价方面，房东友好程度会影响游客心情及旅游归属感。个性化体验、温馨舒适与自由自在的住宿氛围是游客在体验短租住宿过程中获得的重要感知。

分别选取情感得分高于均分的正面评论和情感得分异常低的负面评论。评论较为正面的原因是多方面的(房源质量、周围环境、房东): 房东热情友好、反应及时; 房间干净; 位置好、交通便利、靠近地铁; 给租客以温馨舒适的体验。评论情感负面主要原因集中在房源质量本身和房东: 房间或卫生间不干净、味道不好; 没有 wifi、空调、热水、暖气、等基本设施; 房东冷淡、回复慢; 整体体验差, 性价比不高。

8. 结论

8.1 结论及建议

1. 房源供需匹配

- 数量上, 房源集中在海淀、朝阳等中心地区, 向外逐渐减少。整体供需比较匹配, 西城区供不应求, 宜通过引导的方式, 鼓励加大西城区的房源发布。
- 时间上, 订房热度呈现轻微的季节性, 1-4 月订房较多, 5-7 月相对淡季。
- 价格上, 整体来看 1000 元以上高价房源不受青睐。中心区域 500 元以下房型更受欢迎, 主要以公寓、高档公寓、独栋房为主。周边区域有高价房源更有市场, 以农家乐和独栋房为主。平台可以在搜索房源页面增加租房目的选项(如: 商旅出差、团建、旅游)更精准地为用户推荐房源。
- 房间类型上, 租客对于喜好顺序为: 整间>单间>共享间, 供需比较匹配。
- 游客的体验感知评价维度主要包括: 房源质量、周围环境、房东评价及个性化体验四个方面。周边环境方面, 房源名称带有周边位置描述的热度更高, 尤其关注周边是否有地铁。建议根据房东提交的地理位置, 添加智能标题建议。并且在租客房源检索筛选页面增加“1km 内是否有地铁”选项。房源质量上, 卫生情况评分相对较低, 平台有必要督促房东做改善房源的卫生状况。

- 取消规则整体宽松，但房源位置的准确度不高，建议增加房源真实性的核实。

2. 房东

- 70%以上房东注册时长不超过4年，注册时长2-3年居多，和房源发布时长一致。房东回复率高，回复时长大约在一个小时内，整体比较活跃。超赞房东数量符合二八定律，评价标准合理。一半以上房东拥有超过5套房源，63%的房东发布了北京的房源但所在地不在北京。建议平台为拥有较多房源的房东开发更专业化的房源管理功能，为房源和所在地不一致的房东提供完善的自助入住、房屋托管服务。
- 许多房东会对租客在带客、押金、邻居、卫生方面提出要求。建议平台在发布房源页面添加“是否能带客人”，“押金金额”等选项，方便房东发布房源。为部分房东/租客提供清理房间增值服务、继续完善邻里调解服务。
- 平台宜鼓励房东发布特色房源，分享个人故事，赋予民宿更多的文化属性，进而吸引房客预订。
- 用户在房屋选择上比较关注的是房源的质量、周围环境、房东评价及个性化体验四个方面，房东可以在此基础上作出优化。

3. 用户

- 房屋类型和房源所在区域对房子的价格影响较大，若用户追求性价比，可以考虑价格相对偏低的海淀区、丰台区等。

8.2 不足与展望

在探索性分析方面，可以让我们对北京市爱彼迎房源的整体情况有一个大致了解。但是也有些字段没有用到，例如是否支持商务旅行（is business travel ready），房源数据显示的都是不支持（f/false），没有体现出任何差异，因而也不适合拿来绘制简单图表。

在房源推荐挖掘方面，尽管上述分析通过条件筛选、评论排序、关键词查找和其余信息呈现的流程选出了相应的推荐房源，但是该过程仍有许多比较主观并可以改进的方向。首先，在推荐房源挖掘过程中，为了关键词筛选方便，仅仅选

出了10条评论数最多的房源信息。评论数多并不能与房源优质划等，在现有的数据中只有总评论数属性，而并没有提供好评数量和差评数量，如果可以搜集到这些数据，可以从好评数多的房源中进行筛选（为保证不存在好评数和差评数都很高的现象，还需要对差评数多的房源进行删除），同时以月均评论数量为辅助排序方式，进一步提高优质房源筛选的准确性。此外在关键词的选择中，仅使用机器自动计算出关键词，人为地筛选出符合用户需求的关键词。这一步骤其实是相当具有主观性的，而且在实际业务过程中人为地进行筛选会造成效率极其低下。可以通过情感分析，建立词库，给积极的关键词赋正分，消极的关键词负分，计算每条数据的情感均分并进行排列。或者采用分类算法，在用户给出需求关键词后建立近似词库，将每条数据中的每个关键词与所建词库进行比对，如果比对上一条则加一分，最后按照总得分进行排序，选择得分最高的房源。最后同样是人为地查看房源图片，并选出合适的房源。在实际业务中可以通过机器学习的方式，对图像进行识别，并对房源环境给出评分，减少人的主观性。

在价格预测和评论情感分析方面，由于技术、能力等限制，没能很好的优化所得模型，所得预测结果和评论情感分析的结果与真实情况具有一定的误差，这也是日后可以继续优化和改进的地方。