上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

School of Cyber Science and Engineering

人人车二手车价格预测模型
数据挖掘课程设计

个人信息：

Renrenche Used car price prediction model

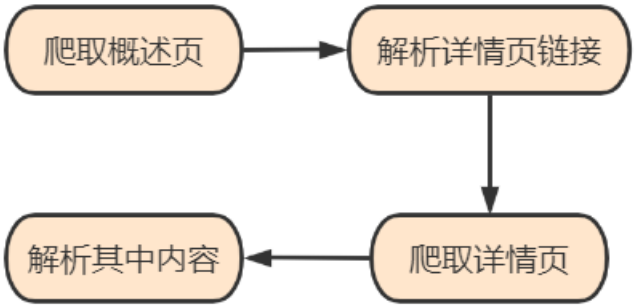
Course Code: IS303

2022 年 6 月 19 日

一、互联网数据爬取

1. 流程的整体概述

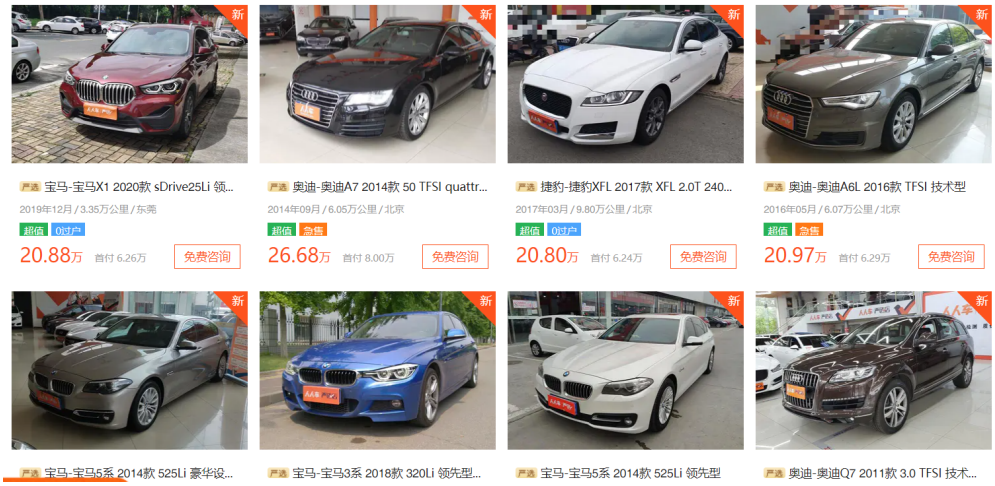
在获取详情页内容的过程中，采用如下流程图中的步骤进行，首先进入人人车的“概述页”（每页包含可以点进去的40个详情页链接），当爬取该页后，可以解析出约为40个的详情页链接。通过获取的详情页链接进入详情页并爬取后，再解析详情页信息。



2. 概述页的爬取

2.1 页面介绍

概述页如图所示，每页可以解析出约为40个的详情页链接。



2.2 url构造

经分析，该网页为html静态网页，requests.get(url)即可实现爬取网页html信息的功能。并且该网页无需登录，因此不需要加入cookie信息，对于爬取工作来说较为友好。该网页的具体结构如下：

https://www.renrenche.com/cn/ershouche/pr-20-30/p2

pr-20-30代表价格区间（价格区间可以自行设置）、p2表示第二页。因此可以通过更改价格区间、页数来访问到不同的页面。,页数范围为1-50，构造的价格区间范围：

```
#价格区间范围
price_range = ['1-2', '2-3', '3-4', '4-5', '5-6', '6-7', '7-8', '8-9', '9-10',
'10-11', '11-12', '12-13', '13-14', '14-15', '15-16', '16-17', '17-18', '18-19',
'19-20', '20-21', '21-22', '22-23', '23-24', '24-25', '25-26', '26-27', '27-28',
'28-29', '29-30''30-31', '31-32', '32-33', '33-34', '34-35', '35-36', '36-37',
'37-38', '38-39', '39-40']

#price表示价格范围，str(i)为页数
url = r"https://www.renrenche.com/cn/ershouche/pr-" + price + "/p" + str(i) +
r"/"
```

2.3 多线程获取网页信息

在具体的爬取过程中，由于要与服务器进行三次握手连接，所以在访问量较大时比较消耗时间。因此采用多线程技术，开启50个线程并发访问，进行调度。能够有效减少爬取时间。经试验，原来爬取50个网页需要50-60秒，现在仅需10-15秒，提高了近5倍的效率。

```
#经测试，发现是静态html文件，这就比较好办了
#获得html文件所有内容
car_detail_url = []

thread_list = []

price_range = ['1-2', '2-3', '3-4', '4-5', '5-6', '6-7', '7-8', '8-9', '9-10',
               '10-11', '11-12', '12-13', '13-14', '14-15', '15-16', '16-17', '17-18', '18-19', '19-20',
               '20-21', '21-22', '22-23', '23-24', '24-25', '25-26', '26-27', '27-28', '28-29', '29-30'
               '30-31', '31-32', '32-33', '33-34', '34-35', '35-36', '36-37', '37-38', '38-39', '39-40']


for price in price_range:
    start = time.time()
    for i in range(1,51,1):
        url = r"https://www.renrenche.com/cn/ershouche/pr-" + price + "/p" + str(i) + r"/"
        thread = threading.Thread(target=get_detail_url, args=(url,))
        thread_list.append(thread)
        thread.start()

    for thread in thread_list:
        thread.join()

    end = time.time()
    print(f'访问50次网页，耗时：{end - start}', price)
```

```
访问50次网页，耗时：13.937147855758667 1-2
访问50次网页，耗时：15.097347736358643 2-3
访问50次网页，耗时：15.937325954437256 3-4
访问50次网页，耗时：14.769612550735474 4-5
访问50次网页，耗时：14.616761922836304 5-6
```

2.4 详情页链接解析：

通过对于网页中Html信息的分析，其中详情页链接位于 `class_="span6 list-item car-item"`，或`class_="span6 list-item car-item margin-left-0"` 其中的 `href` 元素之中。通过构建：https://www.renrenche.com/href 的方式，进入各个车的详细介绍页面。

```
<li class="span6 list-item car-item " data-is-near="0" style>
  <a rrc-event-name="position2" rrc-event-param="search" href="/bj/car/ede630d7025c6b7d" data-param-r="c
  M2M0NTE5MzYyNmJiKysrJTI2LjY4JTE2NTU2MDIxOTY=" target="_blank" class="thumbnail" id="list_item_href/ede
```

2.5 链接数据处理

如图，这种链接是无效的，可以在获得了数据集后，遍历一遍删去。最终，得到了52000行详情页的链接（请见 `data/car_url_new` ）。

	car_url
0	/sz/car/aedfb4f98c685149
1	/dl/car/3d4e46fee2001d9f
2	/hf/car/9b7211c1b4412c9b
3	/hf/car/80569ef33385119f
4	/hf/car/01b3e467e93b7770
5	/anqing/car/c3567961ed6b9f57
6	/wh/car/164ba63295f2cd86
7	/sz/car/f53884f603d9271d

工作簿	工作表	名称	单元格	值
car_url.csv	car_url		B294	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B5526	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B6487	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B9558	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B10559	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B13340	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B15091	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B16432	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B18433	/cn/sales?fromSource=pc-carlist-banner
car_url.csv	car_url		B20244	/cn/sales?fromSource=pc-carlist-banner

3. 详情页内容爬取

3.1 页面介绍

页面中，有价值的信息主要位于如图两个区域中。

3.2 url构造

采用上述获得的 `data/car_url_new` 中的数据构造访问链接：https://www.renrenche.com/href 即可访问到各个详情页。

```
import pandas as pd
import numpy as np
data = "data/car_url_new.csv"
pd_table = pd.read_csv(data, usecols=[1], header=None) # 没有表头
array_table = np.array(pd_table)
print(array_table[0:6])
```

```
[['car_url']
 ['/sz/car/aedfb4f98c685149']
 ['/dl/car/3d4e46fee2001d9f']
 ['/hf/car/9b7211c1b4412c9b']
 ['/hf/car/80569ef33385119f']
 ['/hf/car/01b3e467e93b7770']]
```

3.3 网页内容的获取

同上，采用多线程的方法获取详情页Html信息

```
#多线程解析详情页情况
car_detail_all = []
thread_list = []


for k in range(0,1056,1):
    start = time.time()

    for i in range(1,51,1):
        url = r"https://www.renrenche.com" + array_table[50*k+i][0]
        thread = threading.Thread(target=get_detail_all, args=(url,))
        thread_list.append(thread)
        thread.start()

    for thread in thread_list:
        thread.join()

    thread_list = []

    end = time.time()
    print(50*k, f'访问50次网页，耗时：{end - start}')
```

3.4 网页内容解析

首先进行的，是较为粗粒度的解析。分别有如下几个方面：车辆名称、二手车售价、新车价格、具体信息、标注信息、检测报告。

初步爬到的单条信息如下：包含文本描述的“检测报告”。

需要处理的中文：车辆名称、上牌时间、车辆所在地、外迁查询、自动/手动变速箱、上架时间、检测报告详情

['现代-索纳塔 2015款 1.6T GLS智能型', '7.98万', '新车含税20.31万', '10.01万公里行驶里程/5年7个月上牌/苏州车辆所在地/国五外迁查询/自动变速箱/1次 过户记录', '超值/急售', '该车原厂原漆，外观无瑕疵，外观无更换；灯光系统正常；内饰整洁；电子系统正常；发动机、变速箱工况正常，怠速规律无抖动，转向灵活；综合车况优秀。无重大事故无泡水事故无火烧事故重大事故排查水泡车排查12项火烧车排查5项重大撞击排查17项轻微碰撞排查21项易损易耗部件23项功能部件检测内部配置15项动力、传动系统9项安全系统5项灯光、空调系统8项外观内饰检测内饰检测24项外观检测49项启动检测4项']

数据集（`data/car_detail`）样子：

	car_name	two_price	new_price	summary	tag	Chinese_description
0	江淮-和悦A30 2013款 1.5L CVT豪华型	1.99万	新车含税6.98万	14.44万公里行驶里程/7年上牌/重庆车辆所在地/国五外迁查询/自动变速箱/0次 过户	超值/0过户	该车外观未发现色差，但局
1	众泰-众泰Z300 2014款 1.5L 都市版手动豪华型	1.88万	新车含税6.99万	7.41万公里行驶里程/6年8个月上牌/成都车辆所在地/国三外迁查询/手动变速箱/0次	超值/急售/0过户	该车原厂原漆，外观无瑕疵，
2	东风-帅客 2011款 1.6L 手动舒适型7座（改装天然气）	2.00万	新车含税8.69万	7.61万公里行驶里程/8年上牌/郑州车辆所在地/国五外迁查询/手动变速箱/0次 过户	0过户	该车原厂原漆，外观有少量瑕
3	大众-宝来 2012款 1.6L 自动舒适型	3.00万	新车含税13.64万	11.06万公里行驶里程/9年8个月上牌/武汉车辆所在地/国四外迁查询/自动变速箱/0次	超值/0过户	该车原厂原漆，外观有少量瑕
4	长安跨越-跨越王 2017款 1.5L后双轮单排DK15	2.98万	新车含税5.06万	7.07万公里行驶里程/2年11个月上牌/深圳车辆所在地/国五外迁查询/手动变速箱/1次	超值	该车原厂原漆，外观无更换；
5	大众-朗逸 2011款 1.6L 自动品雅版	3.00万	新车含税14.06万	2.60万公里行驶里程/10年1个月上牌/吉安车辆所在地/国四外迁查询/自动变速箱/1次	过户记录/7天内上	该车外观未发现钣金、色差，
6	野马汽车-野马T70 2016款 升级版 1.8L 手动舒适型	2.18万	新车含税7.58万	4.83万公里行驶里程/5年9个月上牌/重庆车辆所在地/国五外迁查询/手动变速箱/1次	超值/急售	该车原厂原漆，外观无瑕疵，
7	大众-甲壳虫 2008款 1.8T AT 豪华型	2.70万	新车含税24.67万	9.93万公里行驶里程/14年8个月上牌/潍坊车辆所在地/国三外迁查询/自动变速箱/5次	超值	该车外观未发现钣金、色差，
8	比亚迪-比亚迪S6 2013款 白金版 2.4L 自动精英型	2.00万	新车含税11.93万	9.14万公里行驶里程/9年3个月上牌/成都车辆所在地/国四外迁查询/自动变速箱/2次	超值	该车外观未发现色差，但局
9	大众-宝来 2008款 1.6L 自动舒适型	1.80万	新车含税13.64万	12.78万公里行驶里程/12年7个月上牌/深圳车辆所在地/国三外迁查询/自动变速箱/1	急售	该车外观未发现钣金、色差，
10	大众-朗逸 2008款 1.6L 手动品悠版	1.80万	新车含税11.76万	13.51万公里行驶里程/13年10个月上牌/北京车辆所在地/国三外迁查询/手动变速箱/0次	过户	该车外观未发现钣金、色差，
11	五菱汽车-五菱之光 2015款 1.2L 实用型LSI	2.00万	新车含税3.21万	2.96万公里行驶里程/4年3个月上牌/石家庄车辆所在地/国五外迁查询/手动变速箱/1次	过户记录/7天内	该车原厂原漆，外观无更换；

二、数据预处理

1. 数据集的中文文本信息处理

在获得的数据集`data/car_detail`中，车辆名称、具体信息、标注信息、检测报告都需要进行中文文本信息的处理。

1.1 车辆名称

车辆信息如图所示，据调查，在二手车市场上，主流品牌的车辆相对于非主流车辆来说会更加保值一些，首先选取“-”前的品牌名称，以人人车主要显示的品牌为“主流品牌”，主流品牌为1，非主流品牌为0。

品牌: 不限 大众 丰田 宝马 本田 奔驰 奥迪 别克 日产 福特 现代 雪佛兰 吉利汽车 哈弗 马自达 比亚迪 长安 起亚 凯迪拉克 标致

	car_name	品牌
0	宝骏-宝骏730 2014款 1.5L 手动舒适ESP版 7座	0
1	五菱汽车-五菱荣光 2011款 1.2L标准型	0
2	标致-标致301 2014款 1.6L 自动舒适版	1
3	马自达-马自达6 2008款 2.0L 自动时尚型	1
4	雪佛兰-科鲁兹 2013款 1.6L SE MT	1
5	福特-福克斯 2012款 两厢 1.6L 自动舒适型	1
6	起亚-秀尔 2010款 1.6L MT GL	1
7	五菱汽车-五菱之光V 2017款 1.2L基本型 LMH	0
8	本田-思迪 2007款 1.5L 自动豪华版	1
9	大众-捷达 2015款 1.4L 手动时尚型	1
10	江淮-和悦 2012款 1.5L 手动豪华运动型	0
11	比亚迪-比亚迪F6 2008款 财富版 2.0L 手动舒适型	0
12	五菱汽车-五菱宏光 2014款 1.5L S豪华型	0
13	荣威-荣威350 2011款 350S 1.5L 自动迅悦版	0

1.2 具体信息

具体信息部分可以拆分出多项数据：行驶里程、上牌时间、车牌所在地、外迁查询、变速箱、过户记录、上架时间。

summary
11.65万公里行驶里程/7年4个月上牌/惠州车辆所在地/国五外迁查询/手动变速箱/0次 过户记录/3天内上架时间
9.97万公里行驶里程/8年9个月上牌/合肥车辆所在地/国四外迁查询/手动变速箱/1次 过户记录/3天内上架时间
8.95万公里行驶里程/8年上牌/武汉车辆所在地/国五外迁查询/自动变速箱/0次 过户记录/3天内上架时间
13.21万公里行驶里程/12年9个月上牌/北京车辆所在地/国四外迁查询/自动变速箱/0次 过户记录/3天内上架时间
10.81万公里行驶里程/8年10个月上牌/沈阳车辆所在地/国四外迁查询/手动变速箱/0次 过户记录/3天内上架时间
18.00万公里行驶里程/9年2个月上牌/合肥车辆所在地/国四外迁查询/自动变速箱/1次 过户记录/3天内上架时间
11.65万公里行驶里程/11年7个月上牌/大连车辆所在地/国四外迁查询/手动变速箱/0次 过户记录/3天内上架时间
5.63万公里行驶里程/4年3个月上牌/东莞车辆所在地/国五外迁查询/手动变速箱/1次 过户记录/超过120天上架时间

（1）其中，行驶里程、上牌时间、过户记录、上架时间可以采用正则表达式的方法提取出其中的数字。如果没有上架时间项，说明上架时间很近，设置为1。

（2）变速箱只有手动自动两种：手动设置为0，自动设置为1。

（3）车牌所在地：从上牌难度来看，城市不同同样二手车的价格也会有不同。一般来说一线城市上牌最为困难，新一线城市较为困难，其他城市一般，分为三档：2、1、0。

（4）外迁查询：主要有国三、国四、国五三种。其中国三的上路受到限制，无法迁入外地。国四迁入外地受限，国五最佳。设置成一个二维向量，国三[1,1]、国四[0,1]、国五[0,0]。

使用时间	车牌所在地	外迁标准1	外迁标准2	变速箱	过户次数	上架时间
88	0	0	0	0	0	3
105	1	0	1	0	1	3
96	1	0	0	1	0	3
153	2	0	1	1	0	3
106	1	0	1	0	0	3
110	1	0	1	1	1	3
139	0	0	1	0	0	3
51	1	0	0	0	1	120
172	2	1	1	1	0	3

1.3 标注信息

标注信息如图所示，分为 超值/急售/0过户/准新车 四项。设置一个1*4的向量来进行表示

tag	超值	急售	0过户	准新车
超值/0过户	1	0	1	0
超值	1	0	0	0
超值/急售/0过户	1	1	1	0
超值/0过户	1	0	1	0
0过户	0	0	1	0
超值	1	0	0	0

1.4 检测报告

在该部分详细描述了车辆的检测情况，具体来看，可以提取两个信息：综合车况、外观检测。

综合车况分为：优秀/良好/较好。分别为3、2、1。

外观检测以瑕疵数目记。

检测报告： 该车原厂原漆，外观无瑕疵，外观无更换；灯光系统正常；内饰整洁；电子系统正常；发动机、变速箱工况正常，怠速规律无抖动，转向灵活；综合车况优秀。无重大事故无泡水事故无火烧事故重大事故排查水泡车排查12项火烧车排查5项重大撞击排查17项轻微碰撞排查21项易损易耗部件23项功能部件检测内部配置15项动力、传动系统9项安全系统5项灯光、空调系统8项外观内饰检测内饰检测24项外观检测49项启动检测4项

Chinese_description
该车原厂原漆，外观无瑕疵，外观无更换；灯光系统正常；内饰整洁；电子系统正常；发动机、变速箱工况正常，怠速规律
该车外观未发现钣金、色差，但局部喷漆；外观有少量瑕疵，外观无更换；灯光系统正常；内饰整洁；电子系统正常；发动
该车外观未发现钣金、色差，但局部喷漆；外观有少量瑕疵，外观无更换；灯光系统正常；内饰整洁；电子系统正常；发动
该车外观未发现色差，但局部钣金、多处喷漆；外观多处有瑕疵，外观少量更换；灯光系统正常；内饰整洁；电子系统正常
该车外观未发现色差，但局部钣金、局部喷漆；外观有少量瑕疵，外观无更换；灯光系统正常；内饰整洁；电子系统正常；

2. 数据集的所有特征

数据集整理后的表格项如图所示：

项目名称	数据类型	取值情况	意义解释
Output:			
二手车价格	float	单位：万元	输出项
Input:			
品牌	int	主流品牌1/其他0	主流品牌的二手车相对保值
新车购入价	float	单位：万元	当时购入的价格，具有参照意义
行驶里程	float	单位：万公里	可以体现车辆的使用、损耗情况
使用时间	int	单位：月	使用时间久的车相对不保值
车牌所在地	int	范围：2/1/0	一线/新一线/普通城市，反映了经济状况，供给/需求市场
外迁标准1	int	范围：1/0	针对排放标准，国三上路受限，价格受到影响
外迁标准2	int	范围：1/0	针对排放标准，国三国四外迁受限，价格受到影响
变速箱	int	范围：1/0	手动档0，自动档1。反映车况，需求
过户次数	int	单位：次	过户越多，车况更复杂，相对越不保值
上架时间	int	单位：天	上架时间长，卖主可能会相对降价等
超值	int	范围：1/0	超值标记，意味着价格可能偏低
急售	int	范围：1/0	急售标记，意味着价格可能偏低
0过户	int	范围：1/0	0过户标记，意味着过户少
准新车	int	范围：1/0	准新车标记，意味着车更新，价格可能相对高
综合车况	int	范围：3/2/1	人人车车检情况总结，数值高车况好
外观检测	int	单位：个	外观检查，瑕疵的数目，影响价格

	二手车价格	品牌	新车购入价格	行驶里程	使用时间	车牌所在地	外迁标准1	外迁标准2	变速箱	过户次数	上架时间	超值	急售	0过户	准新车	综合车况	外观检测
0	2.6	0	7.90	11.65	88	0	0	0	0	0	3	1	0	1	0	3	0
1	1.0	0	7.59	9.97	105	1	0	1	0	1	3	1	0	0	0	3	1
2	2.5	1	9.87	8.95	96	1	0	0	1	0	3	1	1	1	0	3	2
3	2.6	1	19.52	13.21	153	2	0	1	1	0	3	1	0	1	0	1	11
4	2.3	1	12.50	10.81	106	1	0	1	0	0	3	0	0	1	0	3	3

3. 数据集的性质展示

处理后的数据集性质：

3.1 数据行数、列数：

```
Number of rows    : 51199
Number of columns : 17
```

3.2 数据统计特征：

	二手车价格	品牌	新车购入价格	行驶里程	使用时间	车牌所在地	外迁标准1	外迁标准2
count	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000
mean	15.091028	0.759546	31.079437	7.474026	68.169984	0.823805	0.012110	0.265103
std	9.209838	0.427363	17.613882	4.857677	35.621234	0.725870	0.109376	0.441392
min	1.000000	0.000000	2.610000	0.010000	12.000000	0.000000	0.000000	0.000000
25%	7.600000	1.000000	17.130000	3.820000	41.000000	0.000000	0.000000	0.000000
50%	14.000000	1.000000	28.200000	6.720000	62.000000	1.000000	0.000000	0.000000
75%	21.000000	1.000000	41.640000	10.190000	92.000000	1.000000	0.000000	1.000000
max	40.000000	1.000000	99.750000	52.720000	185.000000	2.000000	1.000000	1.000000

变速箱	过户次数	上架时间	超值	急售	0过户	准新车	综合车况	外观检测
51199.000000	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000	51199.000000
0.916151	0.452567	33.536456	0.805973	0.275767	0.702846	0.038184	2.587277	2.464716
0.277164	0.879743	32.929163	0.395454	0.446904	0.457009	0.191643	0.751988	3.272230
0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
1.000000	0.000000	7.000000	1.000000	0.000000	0.000000	0.000000	2.000000	0.000000
1.000000	0.000000	30.000000	1.000000	0.000000	1.000000	0.000000	3.000000	1.000000
1.000000	1.000000	60.000000	1.000000	1.000000	1.000000	0.000000	3.000000	4.000000
1.000000	10.000000	120.000000	1.000000	1.000000	1.000000	1.000000	3.000000	28.000000

3.3 数据性质：

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51199 entries, 0 to 51198
Data columns (total 17 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   二手车价格   51199 non-null  float64
 1   品牌      51199 non-null  int64
 2   新车购入价格  51199 non-null  float64
 3   行驶里程    51199 non-null  float64
 4   使用时间    51199 non-null  int64
 5   车牌所在地   51199 non-null  int64
 6   外迁标准1   51199 non-null  int64
 7   外迁标准2   51199 non-null  int64
 8   变速箱     51199 non-null  int64
 9   过户次数    51199 non-null  int64
 10  上架时间    51199 non-null  int64
 11  超值      51199 non-null  int64
 12  急售      51199 non-null  int64
 13  0过户     51199 non-null  int64
 14  准新车     51199 non-null  int64
 15  综合车况    51199 non-null  int64
 16  外观检测    51199 non-null  int64
dtypes: float64(3), int64(14)
memory usage: 6.6 MB
None
```

3.4 值缺失情况：

```
值缺失情况：

二手车价格      0
品牌         0
新车购入价格     0
行驶里程        0
使用时间        0
车牌所在地       0
外迁标准1       0
外迁标准2       0
变速箱         0
过户次数        0
上架时间        0
超值          0
急售          0
0过户         0
准新车         0
综合车况        0
外观检测        0
dtype: int64
```

三、数据建模、调优、测试

1. 数据集划分：

首先，采用7：3的比例划分训练集与测试集：

```
Number of train    : 35839
Number of test     : 15360

输入数据：
[[ 0.     9.26   1.68 ...   1.     0.     3.   ]
 [ 0.    27.12   6.3  ...   1.     0.     3.   ]
 [ 1.    54.23  12.95 ...   0.     0.     2.   ]
 ...
 [ 1.    14.06  12.41 ...   1.     0.     3.   ]
 [ 1.    59.77  10.22 ...   1.     0.     3.   ]
 [ 1.    15.07   4.11 ...   1.     0.     3.   ]]

输出数据：
[ 5.76 13.5  22.8  ...  2.2  22.1   6.  ]
```

2. 数据建模训练

2.1 简单的回归算法

首先，采用了几种简单的回归算法：线性回归，岭回归，Lasso回归。效果不太理想。

以线性回归为例，$R2$ 值在0.766左右，均方误差19.72，平均绝对误差：3.28，平均相对误差34.6%。在效果不太理想。

```
The value of default measurement of LinearRegression is 0.7656556644426913
The mean squared error of LinearRegression is 19.725482473721062
The mean absoluate error of LinearRegression is 3.2817077608899243
The mean absoluate percentage error of LinearRegression is 34.57563331493244%

The value of R-squared of LinearRegression is 0.7656556644426913

真实值
[22.8  16.88 22.   ... 18.    1.7  23.8 ]
预测值
[19.51092565 15.26058353 25.70679184 ... 16.32159304 -1.1956383
 16.66371176]
```

2.2 SVM高斯核函数

采用SVM支持向量机，并采用核函数：rbf（其他几个核函数经测试效果没有这个好）。

$R2$ 值在0.843，均方误差13.3，平均绝对误差：2.43，平均相对误差19.4%。

```
The value of default measurement of rbf_svr is 0.8425616521816595
The mean squared error of rbf_svr is 13.25206928171225
The mean absoluate error of rbf_svr is 2.4343747687704815
The mean absoluate percentage error of rbf_svr is 19.392440164095877

The value of R-squared of rbf_svr is 0.8425616521816595

真实值
[22.8  16.88 22.   ... 18.    1.7  23.8 ]
预测值
[24.05792225 14.04939    26.5538342  ... 14.80379905  2.21866567
 16.46117863]
```

2.3 KNN回归算法

采用KNN算法，参数：`n_neighbors=2`

$R2$ 值在0.860，均方误差11.8，平均绝对误差：2.28，平均相对误差19.2%。

```
The value of default measurement of neigh is 0.8592272968851322
The mean squared error of neigh is 11.84927078125
The mean absoluate error of neigh is 2.2754316406250004
The mean absoluate percentage error of neigh is 19.218032056348395

The value of R-squared of neigh is 0.8592272968851322

真实值
[22.8  16.88 22.   ... 18.    1.7  23.8 ]
预测值
[25.5  10.8  25.5  ... 17.25  1.7  18.4 ]
```

2.4 回归树算法

采用回归树算法，参数：`criterion = "mse",min_samples_leaf = 5`

$R2$ 值在0.901，均方误差8.37，平均绝对误差：1.89，平均相对误差16.1%。

```
The value of default measurement of reg_tree is 0.900565308844123
The mean squared error of reg_tree is 8.36972335179598
The mean absoluate error of reg_tree is 1.8927947219122025
The mean absoluate percentage error of reg_tree is 16.070449517246068%

The value of R-squared of reg_tree is 0.900565308844123

真实值
[22.8  16.88 22.   ... 18.    1.7  23.8 ]
预测值
[23.34666667 11.885      24.18333333 ... 17.28666667  2.18
 20.772      ]
```

2.5 随机森林回归

采用随机森林的回归算法，默认参数。

$R2$ 值在0.939，均方误差5.16，平均绝对误差：1.45，平均相对误差12.8%。

```
The value of default measurement of RandomForest is 0.9387017725650916
The mean squared error of RandomForest is 5.1596600705621105
The mean absoluate error of RandomForest is 1.445085228949653
The mean absoluate percentage error of RandomForest is 12.800837660937253%

The value of R-squared of RandomForest is 0.9387017725650916

真实值
[22.8  16.88 22.   ... 18.    1.7  23.8 ]
预测值
[22.4598 13.7602 23.2243 ... 17.171   1.7607 18.2706]
```

2.6 XGBoost 算法

经过调查得知，基本所有的机器学习比赛的冠军方案都使用了XGBoost算法。

采用默认参数下XGBoost算法：
$R2$ 值在0.915，均方误差7.14，平均绝对误差：1.86，平均相对误差16.0%。

```
The mean squared error of xgb is 7.145584375132968
The mean absoluate error of xgb is 1.8629960373346306
The mean absoluate percentage error of xgb is 15.961696361462527

The value of R-squared of xgb is 0.9151084276498644

真实值
[22.8  16.88 22.   ... 18.    1.7  23.8 ]
预测值
[21.208641  14.859972  24.62674   ... 15.428477   2.3888865 16.35744  ]
```

3. XGBoost 算法参数调优

经过调查得知，基本所有的机器学习比赛的冠军方案都使用了XGBoost算法。

所以针对XGBoost算法尝试进行参数调优。

参数	默认值	可选值	推荐值	含义
n_estimators	100	int	100	弱评估器的数量
booster	‘gbtree’	[‘gbtree’, ‘gblinear’ , ‘dart’]	‘gbtree’	弱学习器的类型
learning_rate	0	0~1	0.1,0.015…	防止过拟合
gamma	0	0~1	1,0.9,0.7…	叶节点分支所需损失减少的最小值
reg_alpha	0	0~1	1,0.1,0.01…	L1正则化权重项
reg_lambda	0	0~1	1,0.1,0.5…	L2正则化权重项
max_depth	6	int	9,12,15,17…	树的最大深度
min_child_weight	1	int	1,3,5,7…	孩子节点中最小的样本权重和
subsample	1	0~1	1,0.9…	弱学习器训练比例，防止过拟合
colsample_bytree	1	0~1	1,0.9…	特征的随机采样比例

参数	默认值	可选值	推荐值	含义
objective	“reg:linear”	“reg:linear”：线性回归 “reg:logistic”：逻辑回归 “binary:logistic”：二分类输出为概率 “multi:softmax”：多分类问题	“reg:linear”	指定学习任务及学习目标
eval_metric	‘rmse’	‘rmse’：用于回归任务 ‘mlogloss’：用于多分类任务 ‘error’：用于二分类任务 ‘auc’：用于二分类任务	‘rmse’	评估方法

贪心算法逐个调参寻找局部最优：

```
import xgboost as xgb
xgb_model = xgb.XGBRegressor(booster = 'gbtree',
                             learning_rate = 0.1,
                             gamma = 0,
                             reg_alpha = 1,
                             reg_lambda = 1,
                             max_depth = 20,
                             min_child_weight = 3,
                             subsample = 0.95,
                             colsample_bytree = 0.9,
                             objective = 'reg:linear',
                             n_estimators = 300,
                             n_jobs = -1)

xgb_model.fit(x_train, y_train,
              eval_set=[(x_train, y_train)],
              eval_metric='logloss',
              verbose=100)

xgb_y_pred = xgb_model.predict(x_test)
```

调参后的结果：

$R2$ 值在0.947，均方误差4.49，平均绝对误差：1.25，平均相对误差11.8%。

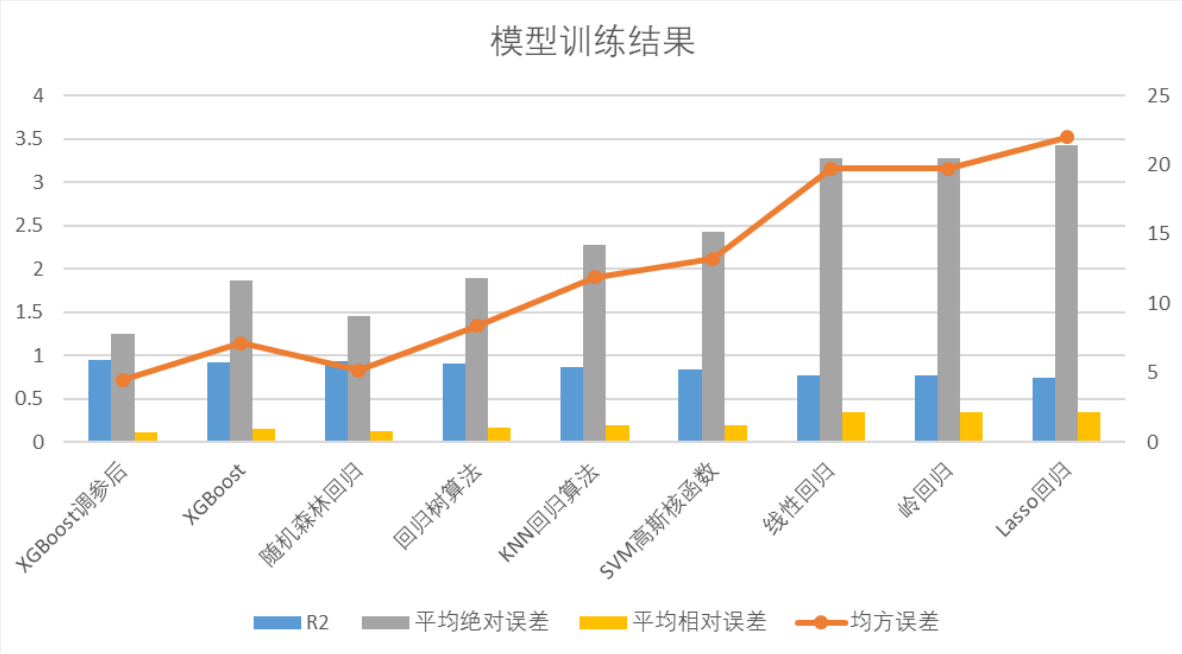
```
The mean squared error of xgb is 4.494070255303872
The mean absoluate error of xgb is 1.2452496707684668
The mean absoluate percentage error of xgb is 11.84320521318842

The value of R-squared of xgb is 0.9466091686563254

真实值
[22.8  16.88 22.   ... 18.    1.7  23.8 ]
预测值
[21.519754  14.207056  22.138641  ... 17.653316   2.5171142 14.399664 ]
```

四、总结

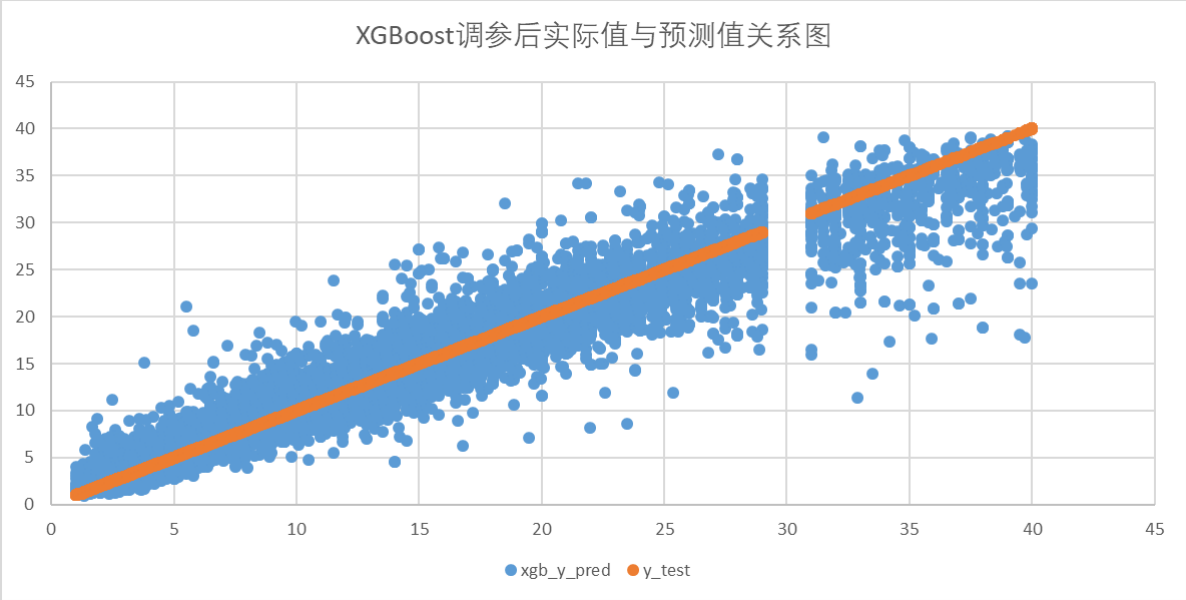
首先，给出各个训练结果的展示：

值在0.843，均方误差13.25，平均绝对误差：2.43，平均相对误差19.4%。

算法	R2	均方误差	平均绝对误差	平均相对误差
XGBoost调参后	0.947	4.49	1.25	11.8%
XGBoost	0.915	7.14	1.86	16.0%
随机森林回归	0.939	5.16	1.45	12.8%
回归树算法	0.901	8.37	1.89	16.1%
KNN回归算法	0.860	11.9	2.28	19.2%
SVM高斯核函数	0.843	13.2	2.43	19.4%
线性回归	0.766	19.7	3.28	34.6%
岭回归	0.766	19.7	3.28	34.6%
Lasso回归	0.739	22.0	3.43	35.0%



最后，在采用 $XGBoost$ 算法并进行参数调优后，获得的 $R2$：0.947，均方误差：4.49，平均绝对误差：1.25，平均相对误差：11.8%。


XGBoost调参后实际值与预测值关系图
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20
25
30
35
40
45
xgb_y_pred
y_test