利用python对上海二手房交易价格的分析报告

材料学院陆子良516051910046

March 26, 2019

1 数据清洗

爬虫得到的特征处理过后共有10项，分别为：

- 房屋价格（单价）；
- 所处楼层；
- 房屋面积；
- 房龄；
- 装修状况；
- 卧室数；
- 客厅数；
- 卫生间数；
- 厨房数；
- 所处位置；（包含经纬度）

共9061rows*9columns

1.1 填充房龄的缺失值

房龄中的缺失值用未缺失值的平均值填充，代码如下：

```
def AgeWash(self):
     self.df = self.df.replace(['未知 '], [np.nan])
     self.df['age'] = 2019 - np.array(self.df['age'], dtype=float)
     self.df = self.df.replace([-1], [np.mean(self.df['age'])])
     return self.df
```

1.2 将楼层和装修文字信息数字化

特征值中的楼层分为高、中、低，分别用2，1，0替代；

特征值中的装修情况为未知，毛坯，简装，精装，分别用-1，0，1，2替代。

代码如下：

```
def StoreyWash(self):
    self.df = self.df.replace(['高', '中', '低'], [2, 1, 0])
    return self.df

 def DecorateWash(self):
        self.df = self.df.replace(['毛坯      ', '精装      ', None, '简
装      ', '其他      '], [0, 2, -1, 1,-1])
        return self.df
```

1.3 计算距离值

在房屋交易过程中，房屋地理位置也是极其关键的因素，然而精确量化位置信息却不容易。本文通过房屋距离市中心距离来量化房屋的位置信息，这与人们直觉中的“市区的房屋比郊区的房屋总体上来说更昂贵”相符合。

接下来需要确定“市中心”的具体位置。

先利用python可视化部分画出房屋价格和房屋位置的散点图：

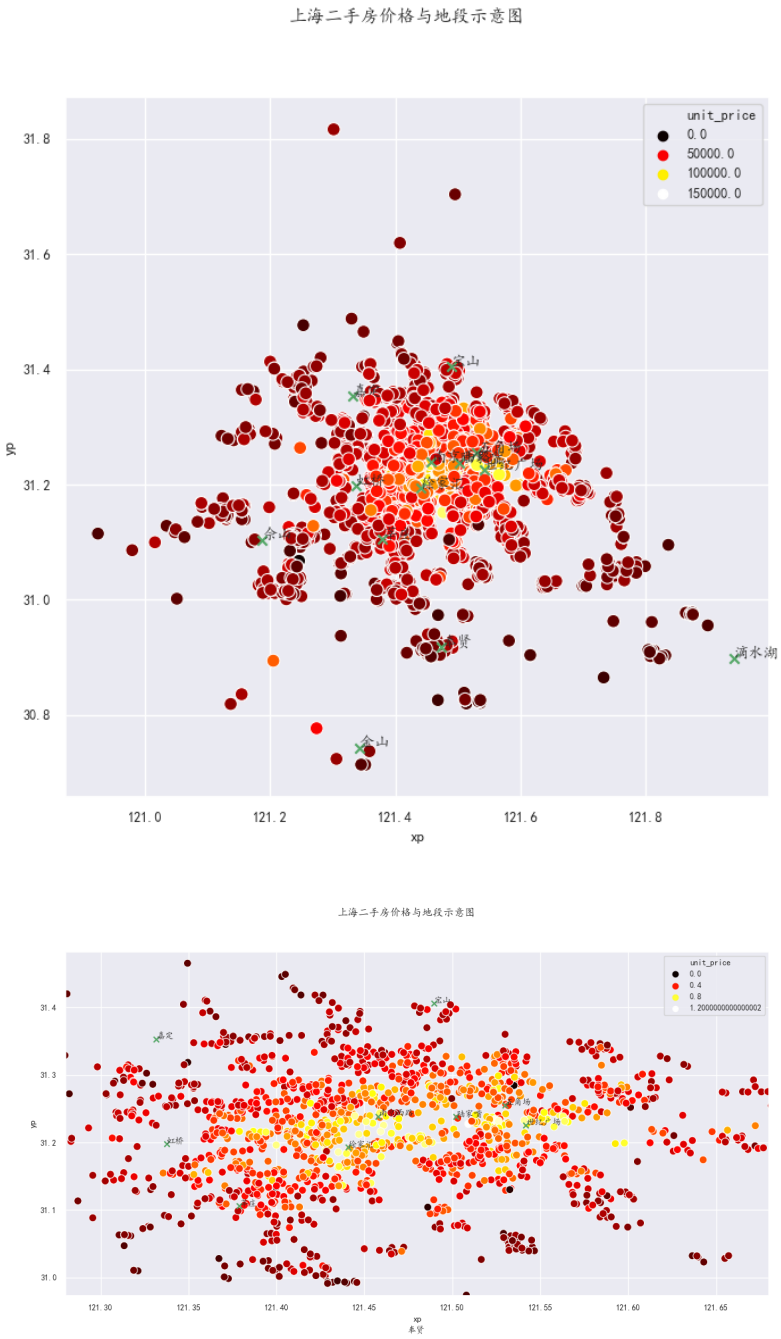

Figure 1: 上海二手房价格与地段散点图及其局部放大图

从图中可以看出，房屋单价从上海市中心向外围呈椭圆形递减。从放大图看，房屋单价最高的区域大致是从徐家汇到世纪广场的长条形区域。由于房屋价格的等高线大致呈现椭圆形，故我们假定房屋质心的位置，及所谓“市中心”就大致处于椭圆的两个焦点处，如此可保证同一等高线上的各个点距这两个焦点的距离的平均值大致相等。

为了进一步确定两个焦点的位置，我们此处采用K-means聚类，以经纬度为坐标，房屋单价为权重，计算出两个焦点的坐标。并据此换算得到图中各点距此的平均距离。这样我们就量化了房屋的位置信息。代码如下：

```
def GetDistance(self,n=2):
    data = self.df[['xp', 'yp', 'unit_price']]
    k = KMeans(n_clusters=n).fit_predict(data[['xp', 'yp']], data['unit_price'])
    data['label'] = k
    xc, yc = [], []
    for i in range(n):
        d = data[data['label'] == i]
        xe = sum(d['xp'] * d['unit_price']) / sum(d['unit_price'])
        ye = sum(d['yp'] * d['unit_price']) / sum(d['unit_price'])
        xc.append(xe)
        yc.append(ye)
    distance=np.zeros_like(data['xp'])
    for x, y in zip(xc, yc):
        dx = np.array(data['xp'] - x,dtype=float)
        dy = np.array(data['yp'] - y,dtype=float)
        distance+=np.hypot(96*dx,57*dy)
    self.df['distance']=distance
    return self.df
```

2 归一化处理

由于各特征值的大小差别很大，为了给之后的拟合提供便利，数据的归一化是很有必要的。这里比较了max-min，对数，zscore归一化方法的效果。首先先考察原数据房屋价格、面积、房龄、距离的分布。

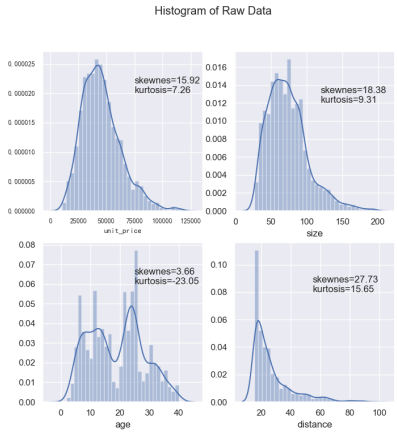

Figure 2: 未归一化的数据直方图

从图中可以看出，单价、面积、距离都呈现明显的负偏。

经过三种归一化后的直方图分别如下所示：

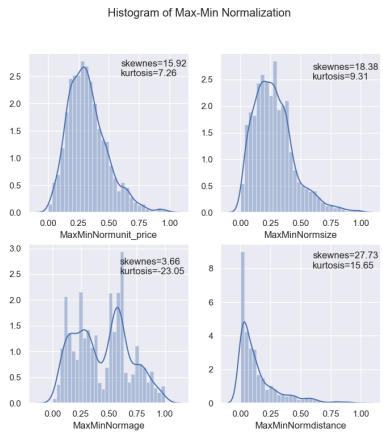

Figure 3: max-min归一化的直方图

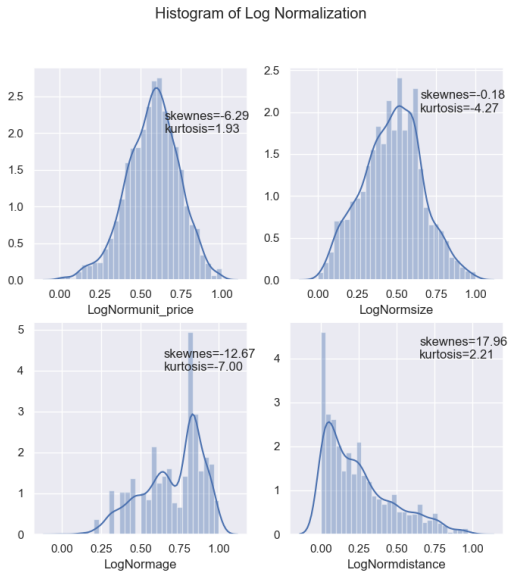

Figure 4: log归一化的直方图

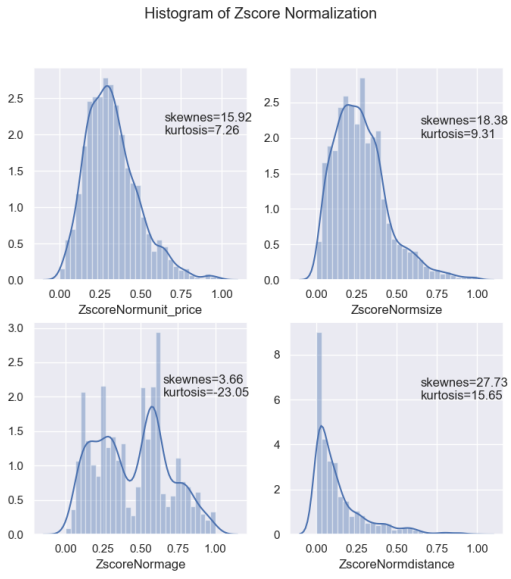

Figure 5: Zscore归一化的直方图

可以看出对数归一化方法得出的数据偏度和峰度明显减小，故采用对数归一化，实现代码如下：

```
def LogNorm(self,columns):
    self.df['LogNorm'+columns] = np.log(self.df[columns])
    max=np.max(self.df['LogNorm'+columns])
    min=np.min(self.df['LogNorm'+columns])
    self.df['LogNorm'+columns]=(self.df['LogNorm'+columns]-min)/(max-min)
    return self.df
```

3 特征值选择

诸特征值之间的相关系数热力图如图所示：

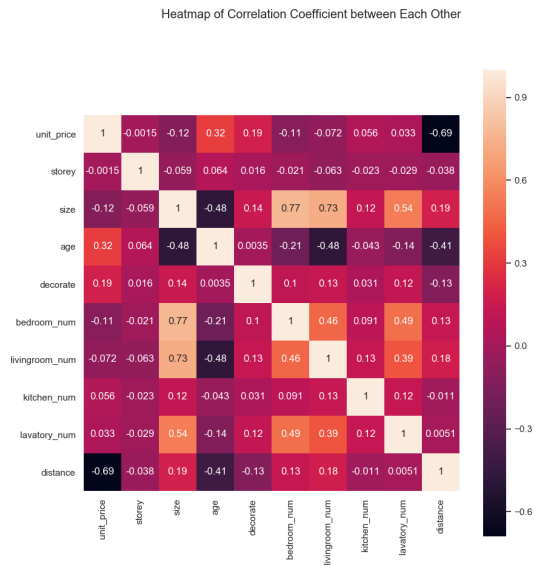

Figure 6: 诸特征之间相关系数热力图

可以看出，房屋价格与距市中心的距离相关性比较高，而其余特征由于较为分散相关性普遍不高，即使如此，项目中仍然选取了几组弱相关的特征值，比如房龄和装修状况。

3.1 房屋价格与距市中心的距离的关系

绘制出房屋价格与距市中心距离的散点图：

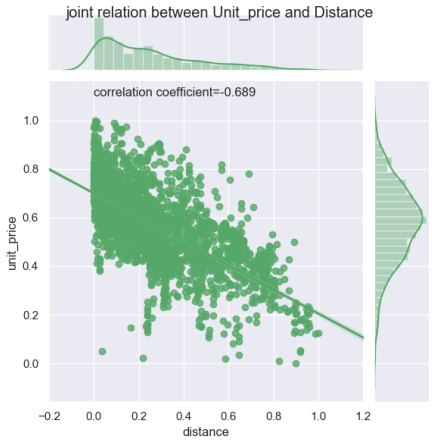

Figure 7: 房屋单价及距市中心距离关系图

可以看出距离市中心的距离是影响房屋价格的重要参数，说明之前通过K-means聚类方法得出的中心点和距离计算方式是有效的。

3.2 房屋价格与房龄的关系

绘制出房屋价格与距市中心距离的散点图：

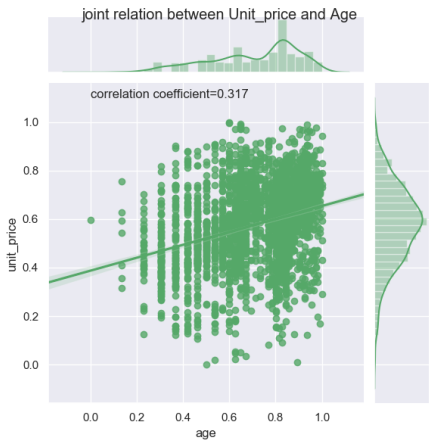

Figure 8: 房屋单价及房龄关系图

关系图反应出的现象是房屋价格与房龄呈现弱正相关，与直觉中“新房的价格更加昂贵”相反。由于房龄和房屋面积、客厅、卧室数呈现负相关，房龄越大的房屋呈现的舒适度不及较新的房屋，然而房龄大的房屋大致聚集在市中心

附近，房屋附近的配套设施带来的便利性更能吸引消费者的瞩目。故房屋价格与房龄呈现弱正相关。

3.3 房屋价格与房屋装修情况的关系

绘制出房屋价格与房屋装修情况的散点图：

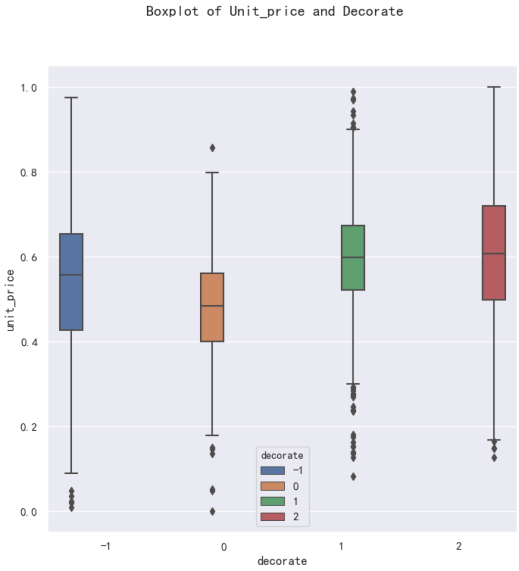

Figure 9: 房屋单价及装修情况箱型图

由箱型图可以看出毛坯房的价格在整体上低于装修过的房屋，然而对于简装和精装房而言，在高价房源上，无疑精装修会增加房屋的价值，然而在中低价位上，精装与简装的区别没有很大的体现。

3.4 房屋价格与卧室、客厅、卫生间数量的关系

对于房屋的居住舒适度，卧室数、客厅与卫生间数量是一个重要的指标，国人一般注重的是卧室与客厅数，可反映在房屋的面积上；而国外更加注重卧室和卫生间的配套情况，一般两者越配套，则使用的舒适度更加高，这里采用卧室数与卫生间数之差与卧室数的比值来量化两者的配套关系。通过图形比较两者对房屋价格的影响程度：

通过计算两者的相关系数和分布，房屋价格和面积的数据体现出的更加分散均匀，且房屋价格与卧室、卫生间数量的相关系数更高，两者呈现负相关也更加符合常理，故此处选用卧室数与卫生间数之差与卧室数的比值作为最终的特征。

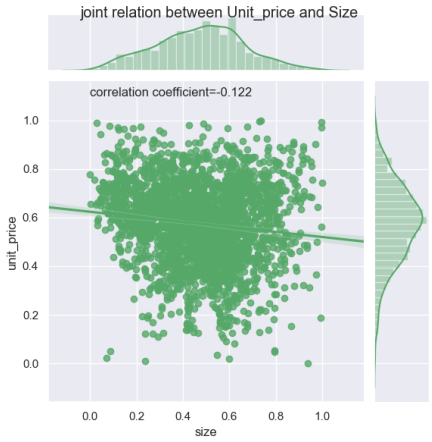

Figure 10: 房屋单价及面积关系图

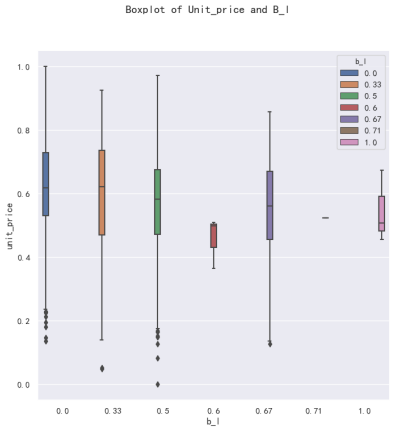

Figure 11: 房屋单价及卧室卫生间比率箱型图

4 数据拟合

下图显示了最终选取特征值之间的相关系数：

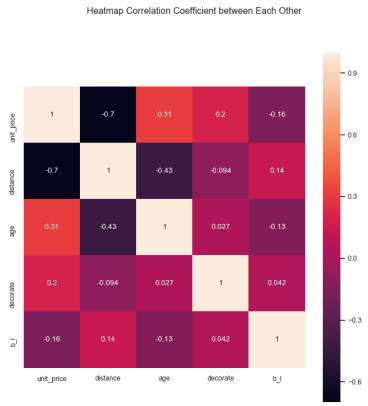

Figure 12: 最终选取特征值之间相关系数热力图

本文利用运用最小平方线性回归和SVM回归来进行多元线性拟合。此外，由于变量之间也存在一定的相关性，故也采用岭回归和Lasso回归算法来优化。在比较预测值与真实值时，由于不可能完全准确，故此处采用相差在10%之内时,判定为准确。在进行十折相差验证后取平均准确率，并引入完全随机数的准确率进行比较，具体结果如下：

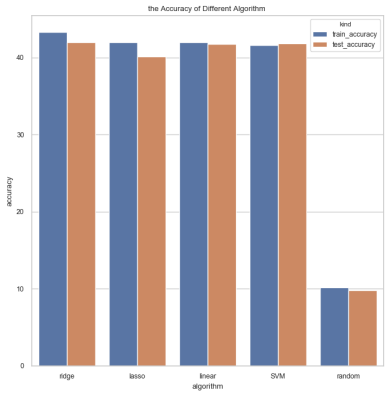

Figure 13: 各种回归算法及随机数预测准确率比较柱状图

可以看出，各种算法的准确率都在40%以上，而完全随机数则在10%以下，说明特征的选择和拟合的结果相对有效。但是还远远没有达到很高的准确率，说明在数据处理和算法的选择上还不够精准，这里由于作者学识有限，敬请读者指教。