重庆市租房价格相关因素分析

Team 4

目录

1 背景介绍 4

2 数据预处理 5

2.1 文字数据的处理 5
2.2 缺失数据的处理 5
2.3 哑变量 5

3 描述性分析 6

3.1 变量表 6
3.2 饼图分析 6
3.2.1 因变量：房源平台 6
3.2.2 因变量：是否近地铁 7
3.2.3 因变量：房源优势 7
3.3 房租区域性在地图上的展示 10
3.4 上线时间对应的房租情况 11

4 模型构建 12

4.1 逻辑回归 12
4.1.1 定义 12
4.1.2 筛选变量 12
4.1.3 模型结果 12
4.1.4 变量重要性 14
4.2 k-邻 14
4.2.1 定义 14
4.2.2 k 的确定 15
4.2.3 变量重要性 16
4.3 决策树 16
4.3.1 定义 16
4.3.2 模型结果 17
4.3.3 变量重要性 18
4.4 朴素贝叶斯 18
4.4.1 定义 18
4.4.2 处理数据与分类过程 19

4.4.3 贝叶斯 19
4.5 LDA 模型 20
4.5.1 定义 20
4.5.2 LDA 系数 20
4.6 随机森林 20
4.6.1 定义 20
4.6.2 错误率与“树”数的选择 21
4.6.3 变量重要性 23

5 模型评估 **23**
5.1 混淆矩阵对比 24
5.2 ROC 对比 25
5.3 准确率对比 26
5.4 AUC 对比 27
5.5 F-score 对比 28

6 结论与建议 **29**
6.1 案例分类讨论 29
6.2 建议 29

7 Appendix: 决策树 **30**

8 参考资料 **33**

1 背景介绍

重庆，简称渝，为中华人民共和国省级行政区，中西部唯一的直辖市、国家中心城市、超大城市、国际大都市，长江上游地区的经济、金融、科创、航运和商贸物流中心，西部大开发重要的战略支点、“一带一路”和长江经济带重要联结点以及内陆开放高地；既以江城、雾都、桥都著称，又以山城扬名。

重庆地处中国内陆西南部，东邻湖北、湖南，南靠贵州，西接四川，北连陕西。总面积 8.24 万平方千米，辖 38 个区县（自治县）；2018 年，重庆常住人口 3101.79 万，地区生产总值 20363.19 亿元；有中国火锅之都、中国会展名城、世界温泉之都之称。

重庆是中国西南地区融贯东西，汇通南北的综合交通枢纽。其江北机场居中国内陆“十大”空港之一，果园港为渝新欧大通道的起点。重庆地处盆地东部，地形由南北向长江河谷倾斜，地貌以丘陵、山地为主，其山地占 76%；长江自西向东横贯境内，流程 691 千米。

重庆是西南地区最大的工商业城市，国家重要的现代制造业基地，有国家级重点实验室 8 个、国家级工程技术研究中心 10 个、高校 67 所，还有中国（重庆）自由贸易试验区、中新（重庆）战略性互联互通示范项目、两江新区、渝新欧国际铁路等战略项目。

基于重庆的优越地理位置和快速发展现状，许多外地人都想要在这个魅力独特的“山城”站稳脚跟，但是这座城市的房价却高居不下，大多数的工薪阶级都买不起房，只能依靠租房来继续自己的生活，租什么样的房比价实惠？租怎样的房比较舒服？怎么样可以租到一个好房等一系列租房问题成了大多数人需要思考的问题，同时这些问题也在困扰着还在找房租的租客，我们利用 2010 条房源数据，试图分析影响重庆租房价格的因素。

2 数据预处理

2.1 文字数据的处理

由于租房价非常依赖房子的基本情况，因此我们可以通过文字处理，针对基本信息列提取出房子几室几厅几居等重要信息；
由于房子出租信息的发布时间包含四十余种类型，为此我们可以根据距离现在的时间划分出九个时间结点，分别是发布不到半个月，发布不到一个月，在 1、2、3、4、5、6 或 8 个月前发布，并以此更新变量值，便于我们进行后期处理；
和发布时间类似的是，房源平台同样包含有非常多不同的值，并且少部分如“链家”等平台的房源要远多于其他的平台，因此可以自行化为一类，剩下的平台均可划为“其他”类别中；
对于仅有是、否两值的列，我们将设置 1、0 变量。

2.2 缺失数据的处理

1. 从基本信息中提取的几室几厅几居的信息会存在比较严重的缺失问题。据常识而言，一般的房屋室的数量等于居的数量，因此，对于缺失“室”数的行，若该行存在“居”的数目，可认为它的“室”数等于“居”数，据此可以对缺失“室”数的变量进行填充。另外，因为室的数据已经可以覆盖居的数据，我们就可以把“居”列从模型中删除。
2. 另外，我们将另起一列命名为“缺失数据”，若室、厅、居数的信息均存在缺失的时候，我们把该行的该值设为 1，其他情况设为 0。这样的操作是为了弥补由于缺失数据而对模型造成影响，并且可以起到与原有的两列类似的效果。

2.3 哑变量

在完成基本的数据处理之后，我们可以进一步进行哑变量的设置，该操作将针对“房源地段”、“房源平台”、“发布时间”、“房源优势”这几列。

3 描述性分析

3.1 变量表

重庆房源变量说明表

变量类型	变量名	详细说明	取值范围
因变量	房租	定性变量 共2个水平	高、低
自变量	卧室数	单位：个	0~9
	厅数	单位：个	0~3
	是否整租	定性变量 共2个水平	1代表整租 0代表非整租
	房源地段	定性变量 共8个水平	大渡口、江北、九龙坡、南岸、沙坪坝、渝北、渝中、其它
	房源平台	定性变量 共3个水平	链家、途家盛捷服务公寓、其它
	发布时间	定性变量 共9个水平	不到半个月前发布、不到一个月前发布、2个月前发布、3个月前发布、4个月前发布、5个月前发布、6个月前发布、8个月前分布、其它
	是否近地铁	定性变量 共2个水平	1代表近地铁 0代表不近地铁
	是否免中介费	定性变量 共2个水平	1代表免中介费 0代表不免中介费
	房源优势	定性变量 共7个水平	精装、拎包入住、双卫生间、随时看房、新上、月租、其它

图 1: 变量表

3.2 饼图分析

3.2.1 因变量：房源平台

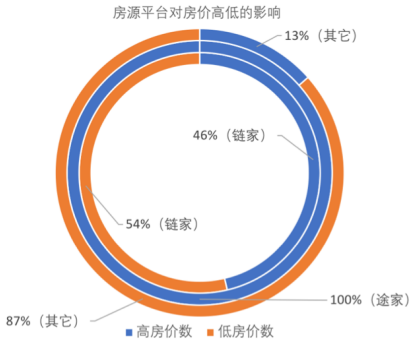


图 2: 房源平台与房租

房源主要来自于链家和途家两个平台，所有非上述两个平台的房源全部归入“其它”类别。有饼图可以看出，途家的房价普遍都较高，链家高低房价分布较均匀，其它平台的房源则低房价数较多。由此可得：大型平台的房源价格普遍较高，可见房源平台是影响房价高低的一大重要因素。

3.2.2 因变量：是否近地铁

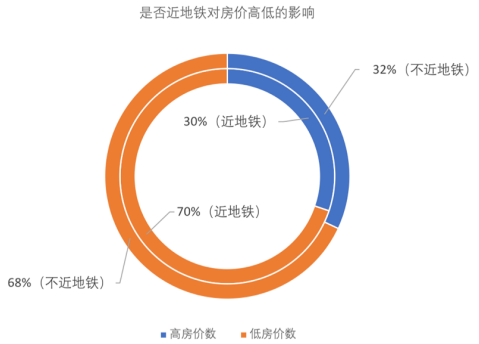


图 3: 是否近地铁与房租

近地铁与不近地铁的高低房价比率基本相同，可见是否近地铁不是影响房价高低的一大因素。

3.2.3 因变量：房源优势

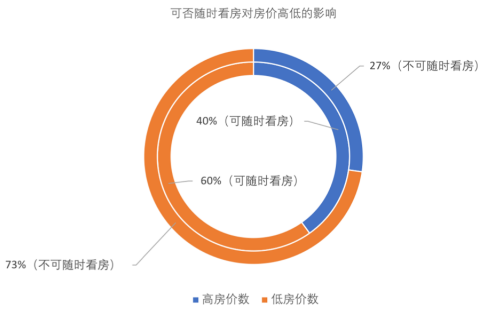


图 4: 可否随时看房与房租

可随时看房的房源相对于不可随时看房的房源高房价数占比更高，可见是否可以随时看房是影响房价高低的一大因素。

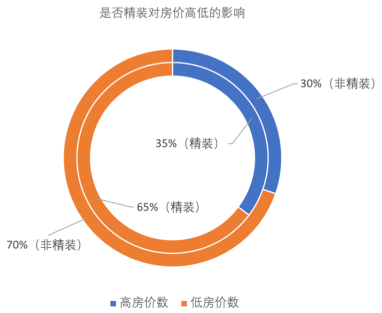


图 5: 是否精装与房租

精装房源相对于非精装的房源高房价数占比更高，可见是否精装是影响房价高低的一大因素。

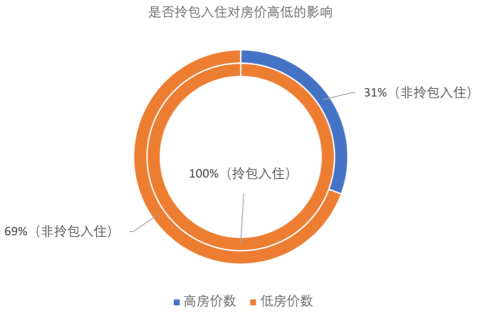


图 6: 是否拎包入住与房租

非拎包入住相对于拎包入住的房源高房价数占比更高，可见是否拎包入住是影响房价高低的一大因素。

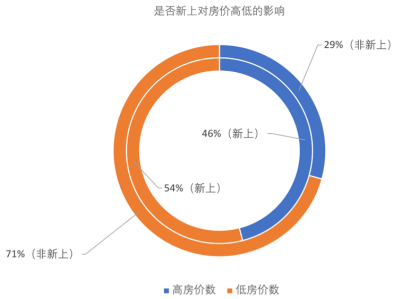


图 7: 是否新上与房租

新上相对于非新上的房源高房价数占比更高，可见是否新上是影响房价高低的一大因素。

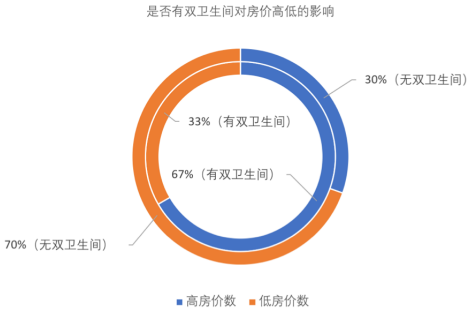


图 8: 有无双卫生间与房租

有双卫生间相对于无双卫生间的房源高房价数占比更高，可见有无双卫生间是影响房价高低的一大因素。

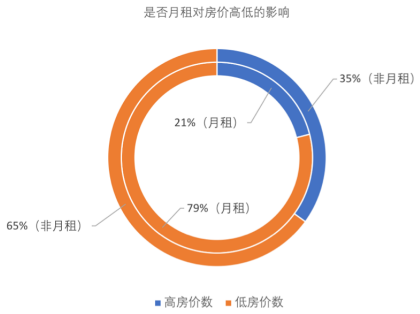


图 9: 是否月租与房租

非月租相对于月租的房源高房价数占比更高，可见是否月租是影响房价高低的一大因素。

3.3 房租区域性在地图上的展示

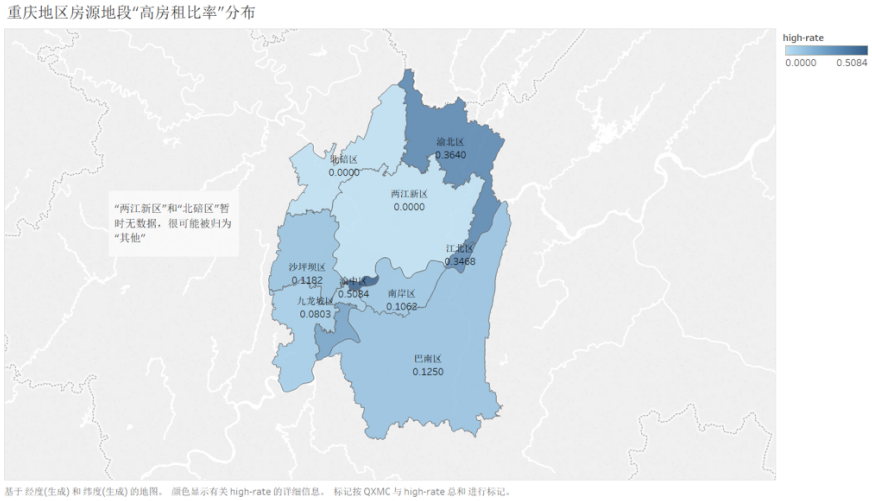


图 10: 重庆各区高房租比例

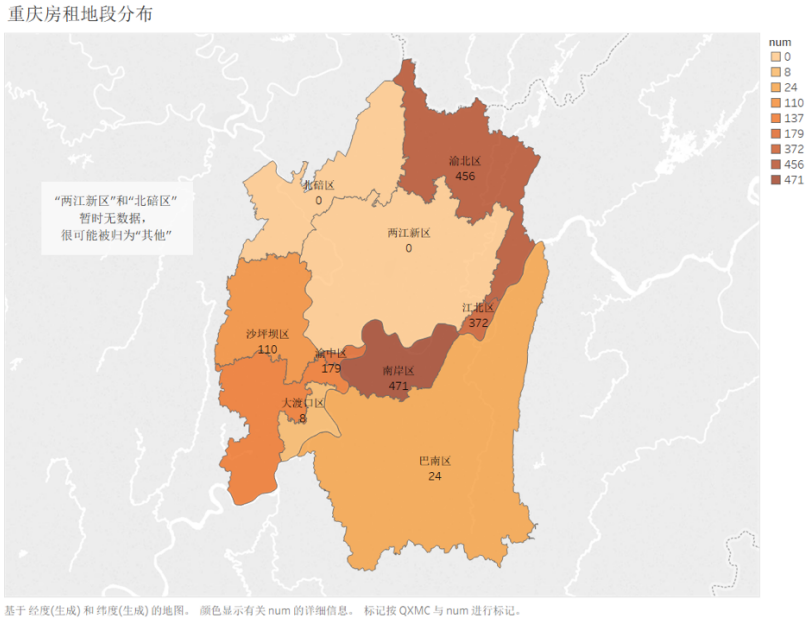


图 11: 重庆房源地段分布

“两江地区”和“北碚区”暂时无数据，很可能是因为被归在了“其他”里面，这给地图分析带来了一些困扰。利用现有数据，可以看到重庆租房集中在“南岸区”、“渝北区”、“江北区”，且这三个区域高房租比率最高，其中以“江北区”和“南岸区”人口密度最大，这也符合了好地段、市中心、沿江区域房价高的道理。

3.4 上线时间对应的房租情况

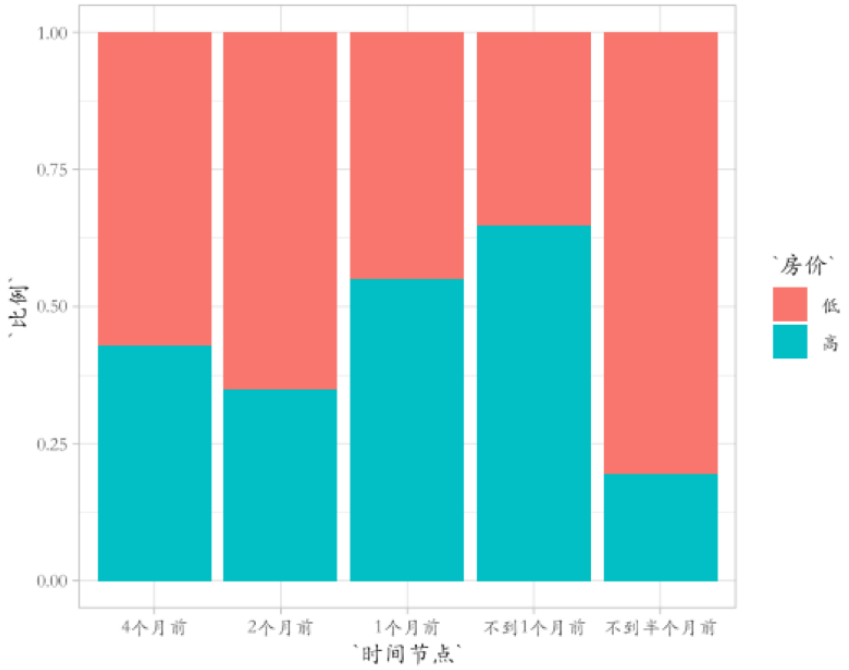


图 12: 上线时间与房租

依据各个时间结点房源数量的分步情况，我们将其继续划分为如图五个时间结点。从时间轴可以看出，不到半个月前刚刚发布的房源中，低房价数比例较高。随着发布时间的增长，高低房价的分布逐渐靠近 1:1 的趋势。

4 模型构建

4.1 逻辑回归

4.1.1 定义

逻辑回归也被称为广义线性回归模型，它与线性回归模型的形式基本上相同，是一种有监督的统计学习方法，主要用于对样本进行分类。通过逻辑回归模型，我们将在整个实数范围上的 x 映射到了有限个点上，这样就实现了对 x 的分类。

4.1.2 筛选变量

未筛选变量时，逻辑回归得到的模型在各项指标上的表现比较差。综合向前向后选择的方法后对变量进行了选择。根据 AIC 判断准则，我们通过筛选得到了以下的模型。通过模型的系数可以发现一些有趣的结论。

4.1.3 模型结果

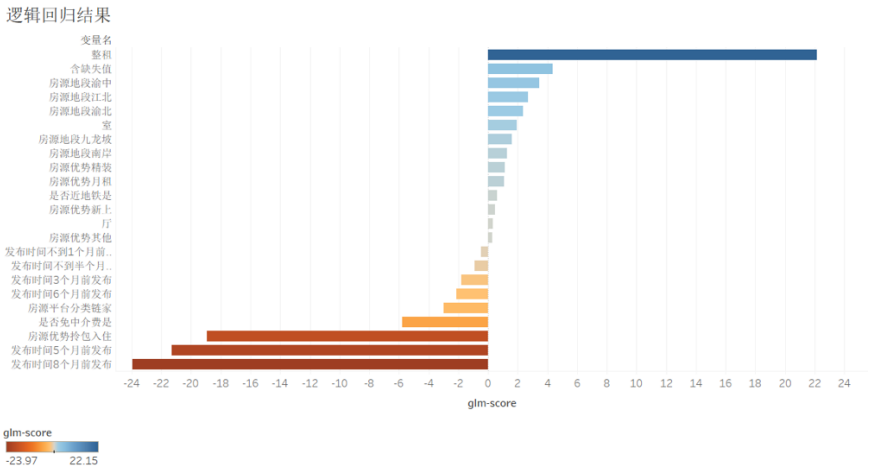


图 13: Logistic Coefficients

室	厅	整租	含缺失值	房源地段江北	房源地段九龙坡	房源地段南岸
1.915545	0.308159	22.14911	4.355474	2.682174	1.596229	1.258096

房源地段其他	房源地段渝北	房源地段渝中	房源平台分类链家
5.462241	2.383196	3.451209	-3.00787

房源平台分类其他	房源平台分类途家盛捷服务公寓	3 个月前发布	5 个月前发布
-1.05268	38.22584	-1.81086	-21.3106

6 个月前发布	8 个月前发布	不到 1 个月前发布	不到半个月前发布	发布时间其他
-2.13594	-23.9735	-0.48973	-0.89632	-2.70236

是否近地铁是	是否免中介费是	房源优势精装	房源优势拎包入住	房源优势其他
0.612112	-5.77234	1.126182	-18.9223	0.2998

房源优势新上	房源优势月租
0.459399	1.100366

表 1: Coefficients Table

逻辑回归模型的分析如下：

(1) 室和厅的数量显然对房价有正的影响，数量越多则房价越可能高。

(2) 整租的房子更有可能属于高租金房子，并且其系数相当大。

(3) 室和厅数据是否缺失也对模型有不小的影响，可能是这些房子体现出在其他方面的优势，更加吸引房客。

(4) 江北、渝中和未列出的其他地区的房子的租金可能要高一些。

(5) 一般平台挂出的房子大部分是低价房，途家盛捷服务公寓等特定房源挂出的房子中更有可能是高价房，这也许和平台定位相关。

(6) 从发布时间来看，发布时间长的房子更有可能是低价房。

(7) 从房源优势上看，近地铁、新上、精装、月租的特性更容易带来高租金房。

4.1.4 变量重要性

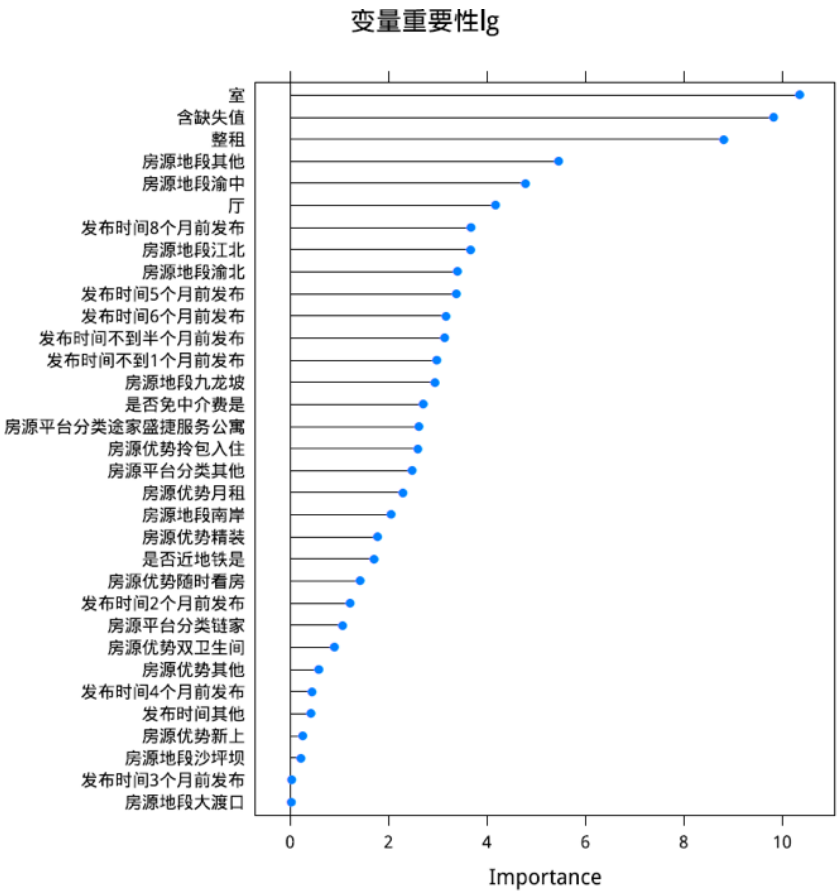


图 14: 逻辑斯蒂回归对应的变量重要性

可以看出，本例的逻辑回归模型中，“室”“厅”的数目、是否整租以及房源地段都很重要。

4.2 k-邻

4.2.1 定义

KNN 是通过测量不同特征值之间的距离进行分类。它的思路是：如果一个样本在特征空间中的 k 个最相似 (即特征空间中最邻近) 的样本中的大多数属于某一个类别，则该样本也属于这个类别，其中 K 通常是不大于 20

的整数。KNN 算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

4.2.2 k 的确定

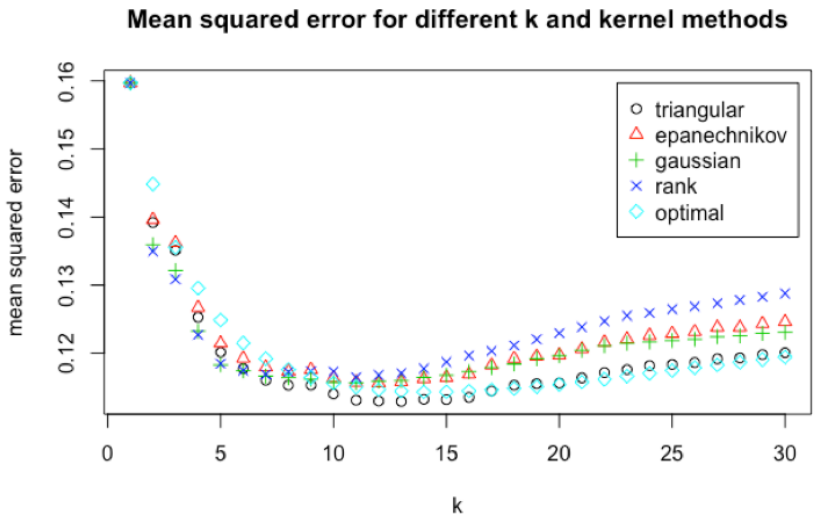


图 15: 不同 k 和 kernel 方法对应误差

从图中可以看到，当 k 为 13 时，大部分 kernel 方法对应的 kNN 模型的误差均取到最小值，因此如果把 k 值取为 13，我们得到的模型的误差将达到最小。另外，从图中可以看出，当 k 在 12 到 15 的范围内，optimal 和 triangular 两种 kernel 方法的误差值最小，因此 kernel 取两种方法都可；相比照下，我们决定选择 optimal。

4.2.3 变量重要性

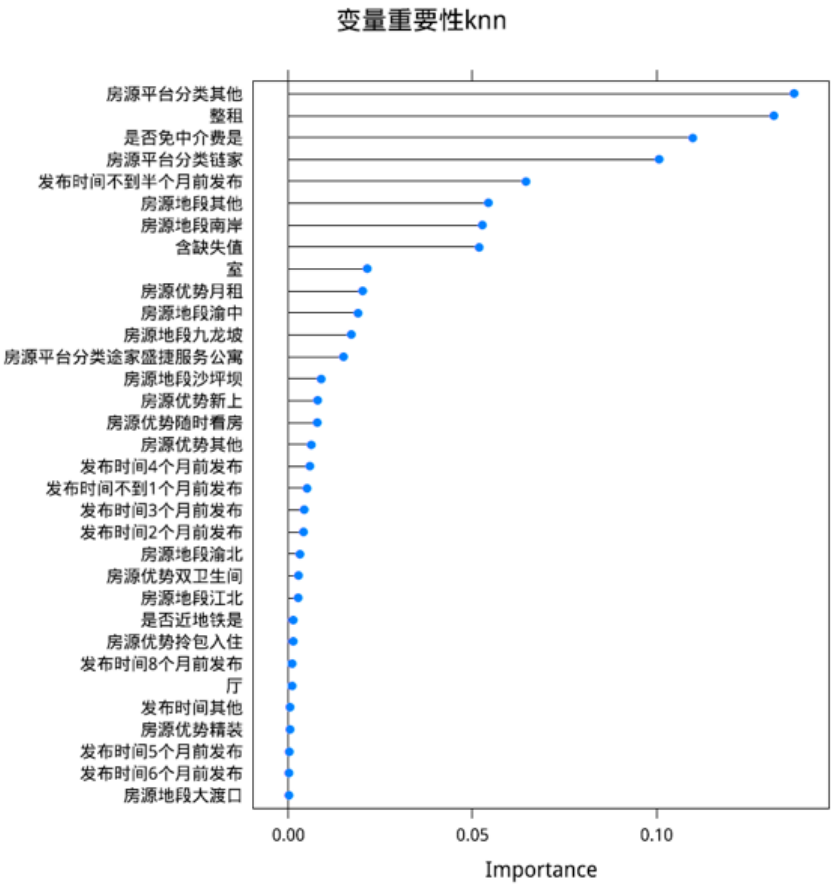


图 16: kNN 对应的变量重要性

从变量重要性图中可以看出，knn 模型里，房源平台、整租、是否免中介费占很大因素。

4.3 决策树

4.3.1 定义

在已知各种情况发生概率的基础上，通过构成决策树来求取净现值的期望值大于等于零的概率，评价项目风险，判断其可行性的决策分析方法，是直观运用概率分析的一种图解法。由于这种决策分支画成图形很像一棵

树的枝干，故称决策树。在机器学习中，决策树是一个预测模型，他代表的是对象属性与对象值之间的一种映射关系。Entropy = 系统的凌乱程度，使用算法 ID3, C4.5 和 C5.0 生成树算法使用熵。
决策点，是对几种可能方案的选择，即最后选择的最佳方案。如果决策属于多级决策，则决策树的中间可以有多个决策点，以决策树根部的决策点为最终决策方案。
状态节点，代表备选方案的经济效果（期望值），通过各状态节点的经济效果的对比，按照一定的决策标准就可以选出最佳方案。由状态节点引出的分支称为概率枝，概率枝的数目表示可能出现的自然状态数目每个分枝上要注明该状态出现的概率。
结果节点，将每个方案在各种自然状态下取得的损益值标注于结果节点的右端。

4.3.2 模型结果

由决策树可以发现，信息增益最大的是“室”这个变量，因此以室作为决策树的根节点。
房源平台对于房价的高低也有一定的影响因素，在变量重要性中排序第二。
[全部结果见 Appendix]

4.3.3 变量重要性

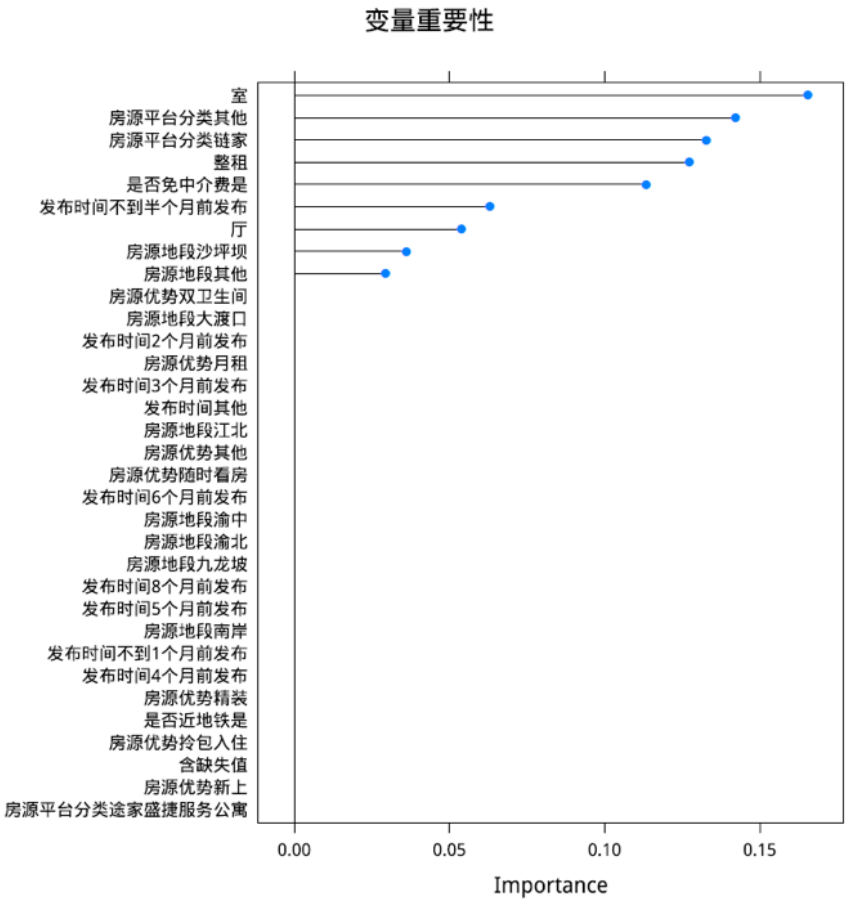


图 17: 决策树对应的变量重要性

4.4 朴素贝叶斯

4.4.1 定义

朴素贝叶斯最核心的部分是贝叶斯法则，而贝叶斯法则的基石是条件概率。贝叶斯法则如下：

$$P(c_i|x,y) = \frac{P(x,y|c_i)P(c_i)}{P(x,y)} \tag{1}$$

这里的 C 表示类别，输入待判断数据，式子给出要求解的某一类的概率。我们的最终目的是比较各类别的概率值大小。这里我们假设 X 的特征之间是

独立的，互相不影响，这就是朴素贝叶斯中“朴素”的由来。

4.4.2 处理数据与分类过程

朴素贝叶斯分类实现的三阶段：
第一阶段，准备工作。根据具体情况确定特征属性，并对每一特征属性进行划分，然后人工对一些待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征属性和训练样本。唯一需要人工处理的阶段，质量要求较高。
在处理数据时需要删掉“零方差”（non-variance）的数据。
第二阶段，分类器训练阶段（生成分类器）。计算每个类别在训练样本中出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。其输入是特征属性和训练样本，输出是分类器。
第三阶段，应用阶段。使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。

4.4.3 贝叶斯

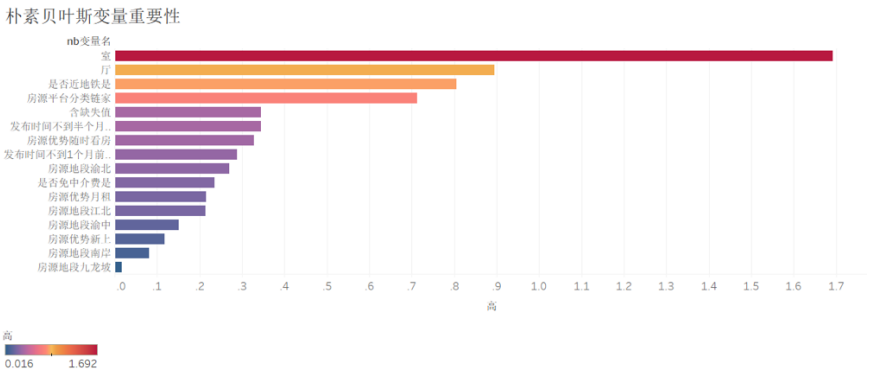


图 18: NB 对应的变量重要性

从变量重要性可以看出：室厅仍然占主导地位，在 NB 模型中，是否靠近地铁也很重要。

4.5 LDA 模型

4.5.1 定义

线性判别分析是一种经典的线性学习方法，在二分类问题上最早由 Fisher 在 1936 年提出，亦称 Fisher 线性判别。线性判别的思想比较朴素：设法将训练样例投影到一条直线上，使得同类样例的投影点尽可能接近，异样样例的投影点尽可能远离；在对新样本进行分类时，将其投影到同样的直线上，再根据投影点的位置来确定新样本的类别。

4.5.2 LDA 系数

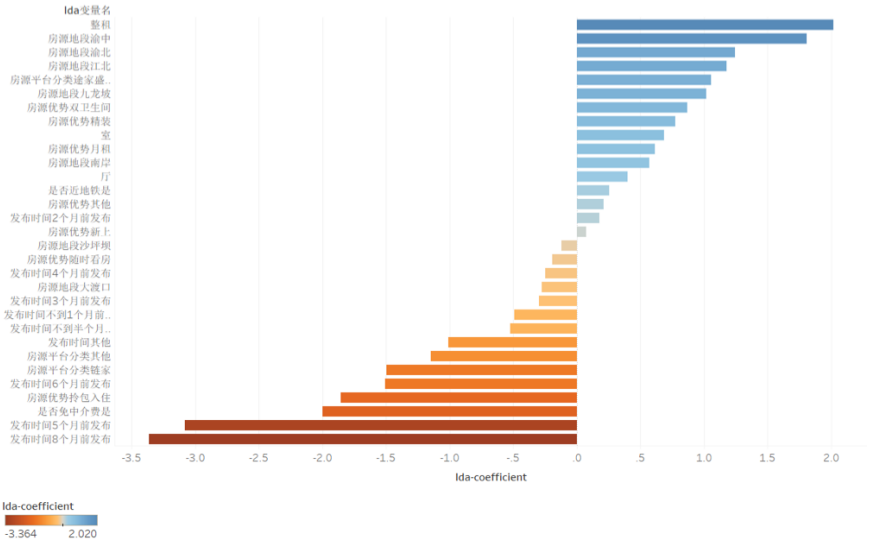


图 19: Coefficients of linear discriminants analysis

从系数可以看出，整租、渝中、渝北、江北等都对房价产生正向的影响。而发布时间很长或者是收取中介费都会产生负影响。

4.6 随机森林

4.6.1 定义

随机森林就是通过集成学习的思想将多棵树集成的一种算法，它的基本单元是决策树。每棵树的按照如下规则生成：

1) 如果训练集大小为 N，对于每棵树而言，随机且有放回地从训练集中的抽取 N 个训练样本（这种采样方式称为 bootstrap sample 方法），作为该树的训练集；

2) 如果每个样本的特征维度为 M，指定一个常数 m«M，随机地从 M 个特征中选取 m 个特征子集，每次树进行分裂时，从这 m 个特征中选择最优的；

3) 每棵树都尽最大程度的生长，并且没有剪枝过程。

一开始我们提到的随机森林中的"随机"就是指的这里的两个随机性。两个随机性的引入对随机森林的分类性能至关重要。由于它们的引入，使得随机森林不容易陷入过拟合，并且具有很好得抗噪能力（比如：对缺省值不敏感）。另，由于随机森林是在决策树基础上将变量数目也进行了随机化选择，因此，在后面的评估分析中，我们将只考虑随机森林模型。

4.6.2 错误率与"树"数的选择

随机森林分类效果（错误率）与两个因素有关：
森林中任意两棵树的相关性：相关性越大，错误率越大；
森林中每棵树的分类能力：每棵树的分类能力越强，整个森林的错误率越低。
减小特征选择个数 m，树的相关性和分类能力也会相应的降低；增大 m，两者也会随之增大。所以关键问题是如何选择最优的 m（或者是范围），这也是随机森林唯一的一个参数。
绘制随机森林 error 图：

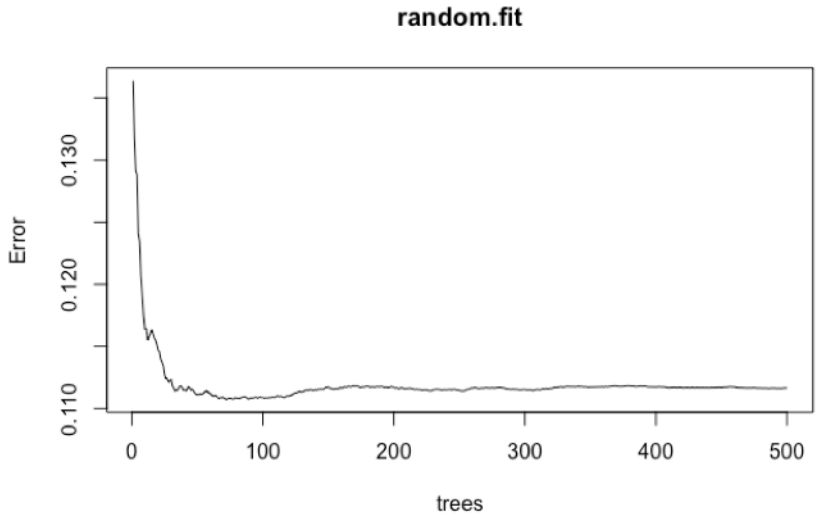


图 20: 随机森林的误差分析

从图中可以看出当 tree 的数量在 100 左右时模型误差最小，因此我们选择 tree 数 100 作为随机森里模型的参数。

4.6.3 变量重要性

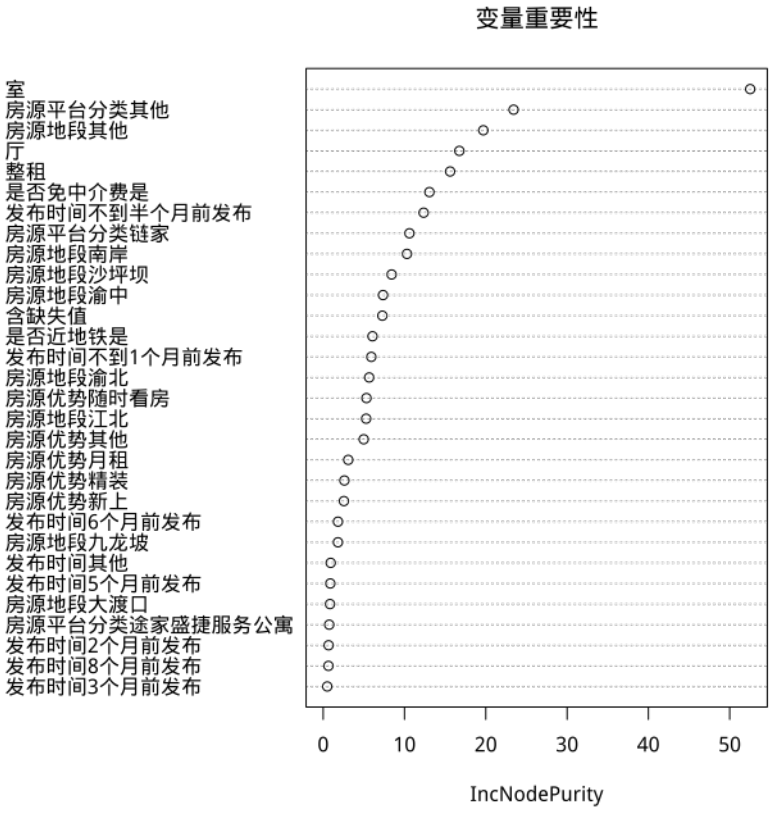


图 21: 随机森林对应的变量重要性

从变量重要性图可以看出，在随机森林模型中，“室”的数量是影响房价高低最大的因素。

5 模型评估

在接下来的分析中，我们采用 10-fold Cross Validation 的方法对各模型进行检验，得到 10 组数据针对各参数进行比对；另外，如前述所示，由于随机森林在决策树的基础上更近一步，因此我们仅对随机森林进行相关分析。

模型	准确度	Auc	F1-score
逻辑回归	0.832	**0.915**	0.723
LDA	0.82	0.902	0.704
随机森林	**0.843**	0.914	**0.751**
朴素贝叶斯	0.697	0.79	0.74
KNN	0.805	0.870	0.717

表 2: 评估数据表

5.1 混淆矩阵对比

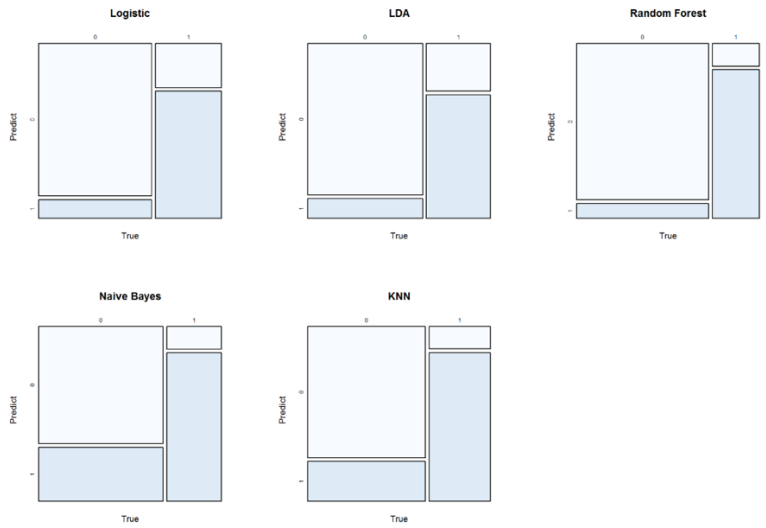


图 22: 各模型对应的混淆矩阵

可以看出，随机森林的真正例的比例明显比其他方法大，而其假正例的比率相当低，说明其准确率等一系列的值都很高。
朴素贝叶斯和 KNN 方法的 FN 很低，代表其查全率较高。
LDA 和逻辑回归方法的 FP 很低，代表其查准率较高。

5.2 ROC 对比

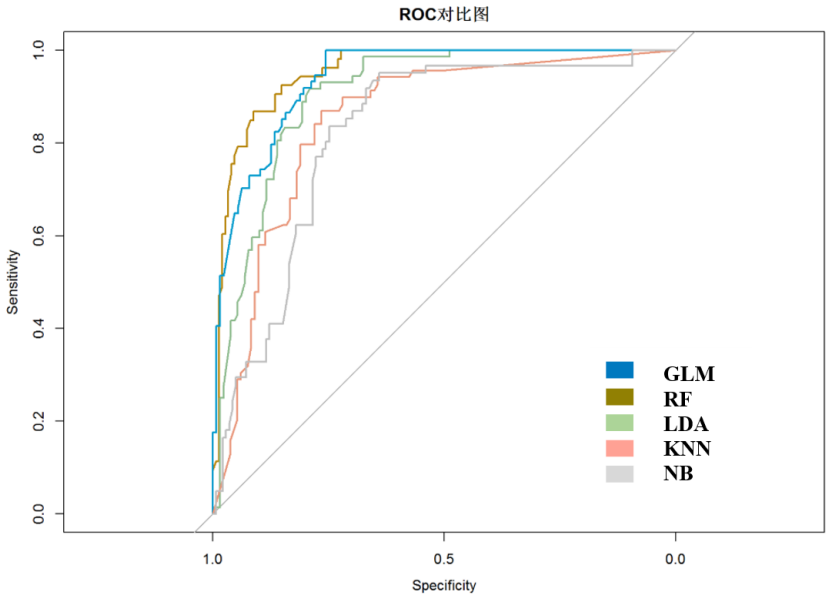


图 23: 各模型 ROC 对比图

可以看出，从 AUC 值上看出，随机森林的效果好于逻辑回归、LDA，最差的是朴素贝叶斯。

5.3 准确率对比

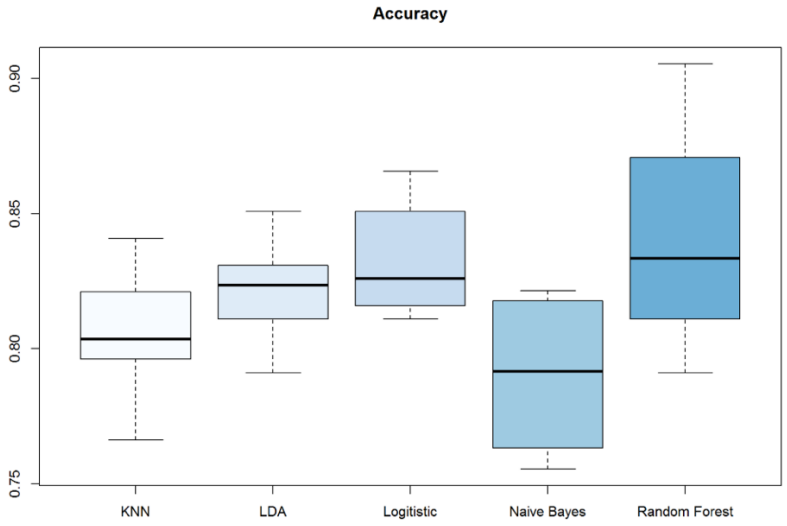


图 24: 各模型准确率对比图

5.4 AUC 对比

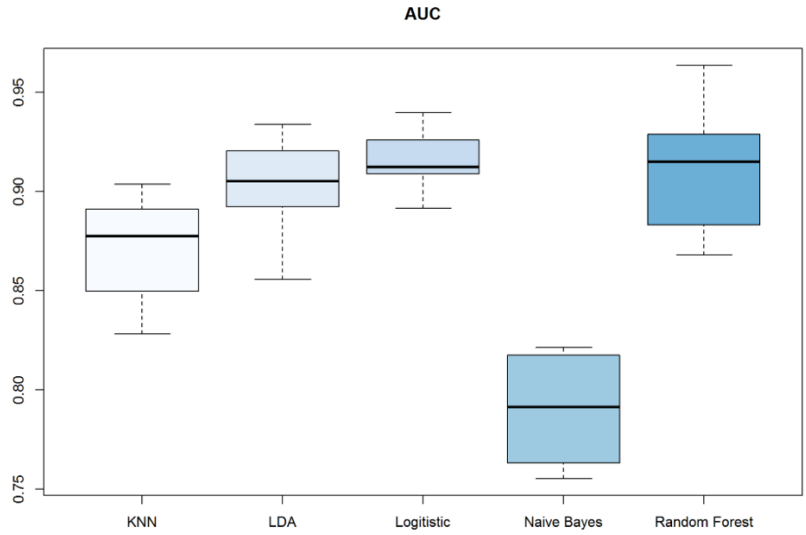


图 25: 各模型 AUC 值对比图

由上面的箱线图可以看出，本例中，rf 模型分类效果最好，它的准确度和 AUC 值都是其中最高的，尤其是准确度的最大值甚至可以达到 0.85 左右。

LDA 和 glm 效果其次，但是稳定性甚至好于随机森林。

朴素贝叶斯方法最差，推测是因为变量之间的相关性无法忽略，没有办法认可变量是相互独立这一理想情况。

5.5 F-score 对比

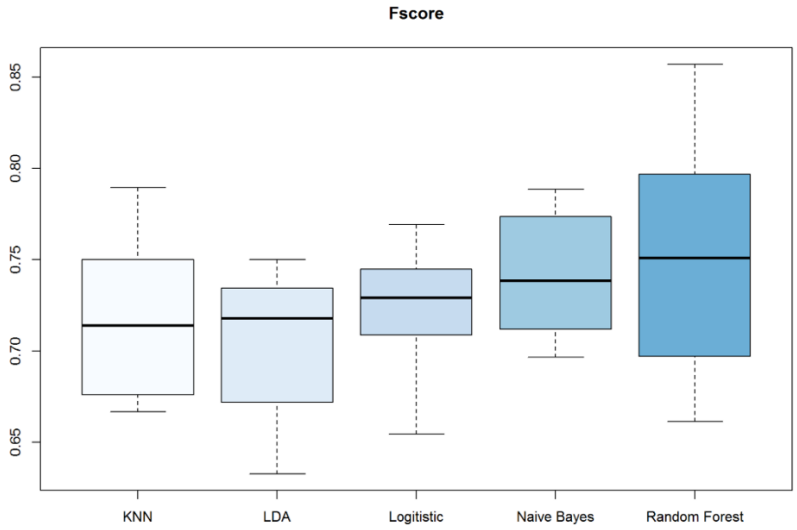


图 26: 各模型 F1 值对比图

与 AUC、accuracy 不太相同的是，朴素贝叶斯方法的 Fscore 值在五种分类方法中跃居到第二名，甚至超过了逻辑回归。推测是因为其 FN 较小，查全率较大。

6 结论与建议

6.1 案例分类讨论

(1) 本例中，分类器随机森林的效果好于逻辑回归、LDA，最差的是朴素贝叶斯。

(2) 在变量中，“室”、“厅”的数目、房源地段、是否整租、是否免中介费都是很重要的因素。这也与大众租房的需求与心理相符合，首先要考虑最基本的“室”、“厅”这样的房间大小客观因素，其次要考虑房源地段这样的问题，是否位于市中心、交通是否便利也是很关键的因素，在这些基础条件满足之后再去考虑“是否免中介费”等价格问题。

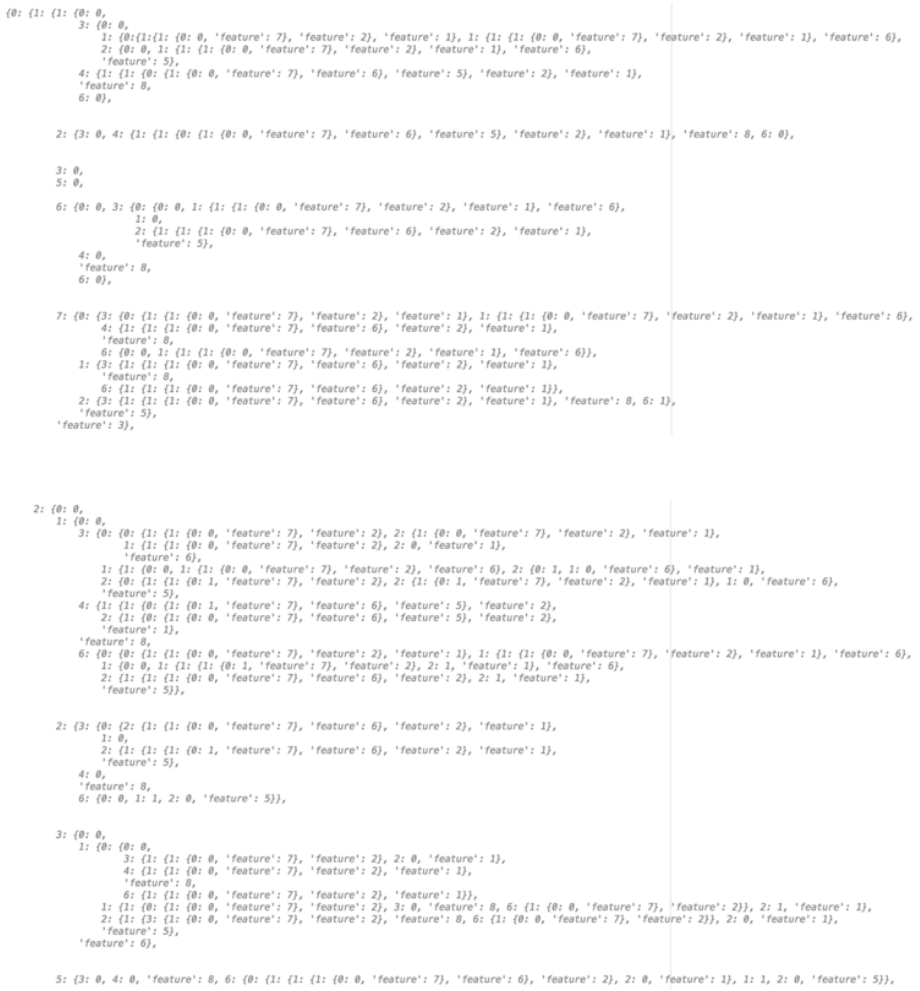
6.2 建议

(1) 在此，我们也尝试向想要尽可能租到便宜又好的租房者提出建议：
首先要充分考虑自己的实际需求，客观而全面地列出自己的目标条件。然后再选择租房平台时，在保证正规、安全的前提下，如果想要尽可能的节省价格，除了链家外，可以选择“若航寓”“小租乐”这样的平台，或许可以更加物超所值，租到令自己满意的房子。

(2) 对于那些想要把自己房子挂出去的出租者，我们也建议把房源优势写的越清楚越好，很多客观因素例如“房源地段”没有办法改变，但是是否整租、装修条件等等仍然有很大的提升空间。
可以考虑整租与将不同房间出租哪种更获益，对于出租者，整租更省心，但是不可避免的可能会有利益损失。
另外在房源平台的选择上除了链家，可以多多去探索“冠寓”“途家”这样的平台，里面的租房价格普遍都为“高”。

7 Appendix: 决策树

打印决策树结果如下:



```
    6: {0: 0,
        1: {0: {1: 1, 2: 0, 'feature': 1},
            3: {1: {0: {1: {0: 0, 'feature': 7}, 'feature': 2}, 1: {1: {0: 0, 'feature': 7}, 'feature': 2}, 2: {1: {0: 0, 'feature': 7}, 'feature': 2}, 'feature': 5},
                2: {0: 1, 1: {1: {0: 1, 'feature': 7}, 'feature': 2}, 2: {1: {0: 1, 'feature': 7}, 'feature': 2}, 'feature': 5},
                'feature': 1},
            4: {1: {1: {0: {0: 0, 'feature': 7}, 'feature': 5}, 'feature': 2}, 2: {1: {0: {0: 1, 'feature': 7}, 'feature': 5}, 'feature': 2}, 'feature': 1},
            'feature': 8,
            6: {0: {1: {1: {0: 1, 'feature': 7}, 'feature': 2}, 2: 1, 'feature': 1},
                1: {1: {1: {0: 1, 'feature': 7}, 'feature': 2}, 2: 1, 'feature': 1},
                2: {1: {1: {0: 0, 'feature': 7}, 'feature': 2}, 2: 0, 'feature': 1},
                'feature': 5}},
        'feature': 6},


    7: {0: {1: {0: {1: {1: {0: 0, 'feature': 7}, 'feature': 6}, 'feature': 2},
                3: {0: 0, 1: {1: {0: 0, 'feature': 7}, 'feature': 2}, 'feature': 6},
                4: {1: {1: {0: 1, 'feature': 7}, 'feature': 6}, 'feature': 2},
                'feature': 8,
                6: {1: {1: {0: 1, 'feature': 7}, 'feature': 6}, 'feature': 2}},
            2: 0,
            'feature': 1},
        1: {0: 1, 3: {1: {1: {0: 1, 'feature': 7}, 'feature': 6}, 'feature': 2}, 2: 0, 'feature': 1}, 'feature': 8, 6: {1: {1: {1: {0: 0, 'feature': 7}, 'feature': 6}, 'feature':
        2: 1,
        'feature': 5},
    'feature': 3},
```

```
3: {0: {1: {1: 1, 2: {1: {1: {0: {'feature': 8, 6: 0}, 'feature': 7}, 'feature': 6}, 'feature': 2}, 'feature': 5}, 2: 0, 'feature': 1},
    1: {0: 1,
        2: 1,
        3: {0: {0: {1: 1, 2: {1: {0: 1, 'feature': 7}, 'feature': 2}, 'feature': 1},
                1: {1: 0, 2: {1: {0: 1, 'feature': 7}, 'feature': 2}, 'feature': 1},
                2: {1: 1, 2: 0, 'feature': 1},
                'feature': 5},

            1: {1: {0: {1: {0: 1, 'feature': 7}, 'feature': 2}, 1: {1: {0: 1, 'feature': 7}, 'feature': 2}, 2: 1, 'feature': 5},
                2: {0: 1, 1: 1, 2: {1: {0: 1, 'feature': 7}, 'feature': 2}, 'feature': 5},
                'feature': 1},
            'feature': 6},
        4: 1,
        6: {1: {0: {1: {0: {0: 0, 'feature': 7}, 'feature': 6}, 'feature': 2}, 1: {0: 1, 1: {1: {0: 0, 'feature': 7}, 'feature': 2}, 'feature': 6}, 'feature': 5},
            2: 1,
            'feature': 1},
        'feature': 8},
    2: {0: {1: 1, 2: 0, 'feature': 1}, 1: 1, 2: 1, 'feature': 5},
    3: {0: 1,
        3: 1,
        4: {1: 1, 2: {1: {0: {1: {0: 0, 'feature': 7}, 'feature': 6}, 'feature': 5}, 'feature': 2}, 'feature': 1},
        'feature': 8,
        6: {0: {1: {1: {0: 1, 'feature': 7}, 'feature': 6}, 'feature': 2}, 2: 1, 'feature': 1}, 1: 1, 2: 1, 'feature': 5}},
    5: {0: 0, 3: {1: 0, 2: {0: 1, 1: 0, 'feature': 5}, 'feature': 1}, 4: 0, 'feature': 8},
    6: {0: {0: {2: 1, 3: {2: {1: {0: 0, 'feature': 7}, 'feature': 2}, 'feature': 1}, 'feature': 8},
            1: {3: 1, 4: {1: {1: {0: 1, 'feature': 7}, 'feature': 2}, 2: 1, 'feature': 1}, 'feature': 8, 6: {1: 1, 2: {1: {0: 1, 'feature': 7}, 'feature': 2}, 'feature': 1}},
            'feature': 6},
        1: {0: 1,
            1: {3: {1: {1: {0: 0, 'feature': 7}, 'feature': 2}, 2: 1, 'feature': 1}, 'feature': 8, 6: {1: 1, 2: {1: {0: 1, 'feature': 7}, 'feature': 2}, 'feature': 1}},
            'feature': 6},
        2: {1: {3: 0, 'feature': 8, 6: {1: {1: {0: 0, 'feature': 7}, 'feature': 6}, 'feature': 2}}, 2: 1, 'feature': 1},
        'feature': 5},
    7: {3: {0: {1: {1: {1: {0: 1, 'feature': 7}, 'feature': 6}, 'feature': 2}, 'feature': 1}, 1: 1, 2: 1, 'feature': 5}, 4: 1, 'feature': 8, 6: 1},
    'feature': 3},
```

```
    4: {0: 0,
        1: {1: 1,
            3: {1: 0, 2: 1, 'feature': 1},
            5: 1,
            6: 1,
            7: 1,
            'feature': 3},
        'feature': 2},
    5: 1,
    6: 1,
    'feature': 0},


1: 1,


2: {0: 0,


    1: {0: 0,
        1: {1: {0: {0: 0, 1: {1: {0: {1: 0, 'feature': 7}, 'feature': 5}, 'feature': 1}, 'feature': 6}, 3: 0, 'feature': 8, 6: 0, 5: 0},
            2: {3: 0, 'feature': 8, 6: 1, 5: 0},
            3: 1,
            4: 1,
            'feature': 0},
        'feature': 2},


    2: 0,
    3: 0,
    4: {0: {2: {0: {1: {1: {1: {'feature': 8, 5: 0}, 'feature': 7}, 'feature': 2}, 'feature': 1}, 1: 0, 'feature': 6},
            4: 1,
            'feature': 0},
        1: 1,
        2: {'feature': 8,
            6: 1,
            5: {1: {0: 1, 1: {1: {1: {1: 1, 'feature': 7}, 'feature': 2}, 'feature': 1}, 'feature': 6},
                2: {0: {1: {1: {1: 1, 'feature': 7}, 'feature': 2}, 'feature': 1}, 1: {1: {1: {1: 1, 'feature': 7}, 'feature': 2}, 'feature': 1}, 'feature': 6},
                'feature': 0}},
        3: 1,
        4: 1,
        5: 1,
```

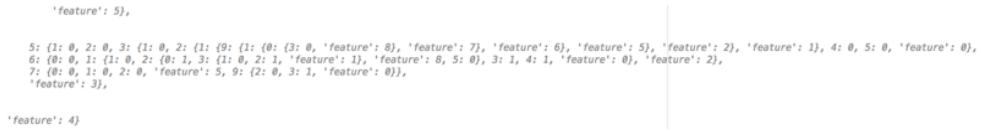


图 27: 打印决策树

图样解释：对齐表示同一结点生成的分支，例如 0:0:0,1:1,feature:1,1:1, feature:2 表示以 feature2 为根节点生成 0、1 两条分支，其中 0 分支通向根节点 feature:1，又生成 0、1 两条分支的子树。

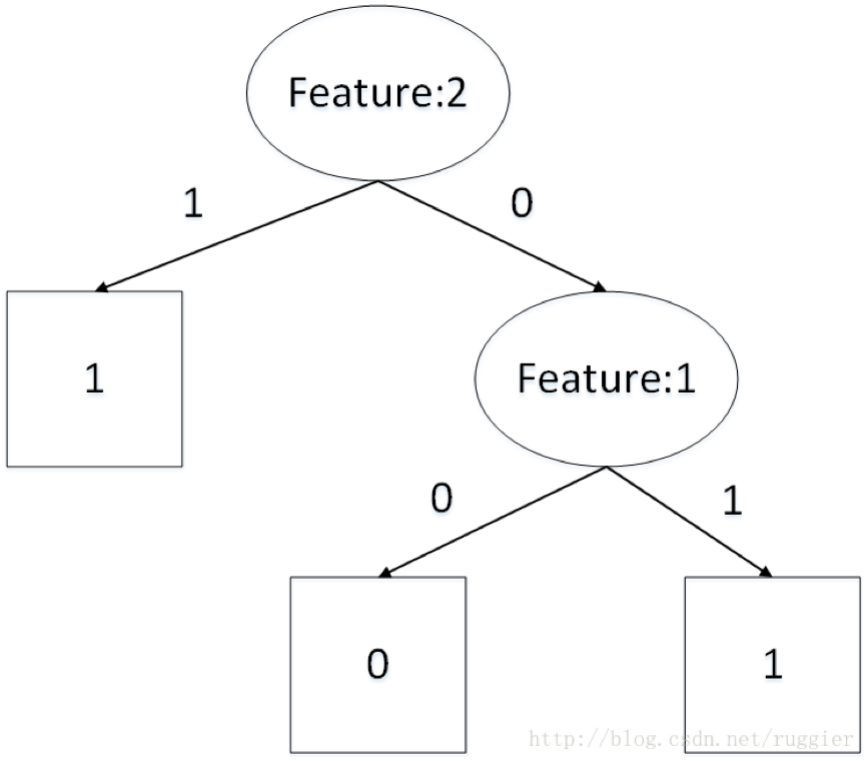


图 28: 决策树直观解释

8 参考资料

参考文献

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction Second Edition, Springer, Febrary 2009*, pages 106-108, 119

[2] CSDN