酒店预订需求预测

——Python 大数据分析原理与应用期末大作业报告

哲学系 21 硕 李彤 2101210943

摘要： 报告用逻辑回归、决策树和随机森林对酒店客人预定的房型进行预测。在建模过程中，选择最优的参数。根据准确率评分和 F1 评分，比较这三种算法的优劣并分析原因。

一、 背景

对于旅游行业的酒店业来说，客人将预定何种房型，是关系到营业额的重要因素。如果酒店可以及时得知客人将预定何种房型，这将有助于酒店动态调整房间分配情况，提高满房率，进而提高营业额。报告使用的数据集来自 Kaggle[1]。

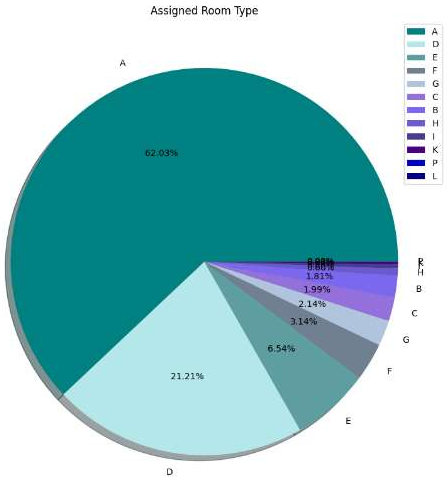
二、 数据集

1、数据集概览

该数据集共有 119390 个样本，32 个特征，二元属性有 1 个，类别属性有 10 个，数值属性有 21 个，我们选取“登记的房型”（“assigned_room_type”）作为预测值，其余 31 个特征作为特征值。预测客人将预定何种房型属于多分类任务。

31 个特征值存在不同程度的缺失值，其中：“Children”有 4 个缺失值，“Country”缺失比例为 0.4%，“Agent”缺失比例为 13.7%，“Company”缺失比例为 94.3%。

预测值“登记的房型”共 12 个类别，其中最多的 A 类占 62%，D 类占 21%，最少的 P 类和 L 类只 0.01%占和 0.0008%。以“登记的房型”为预测值的数据集是非常不平衡的数据集。

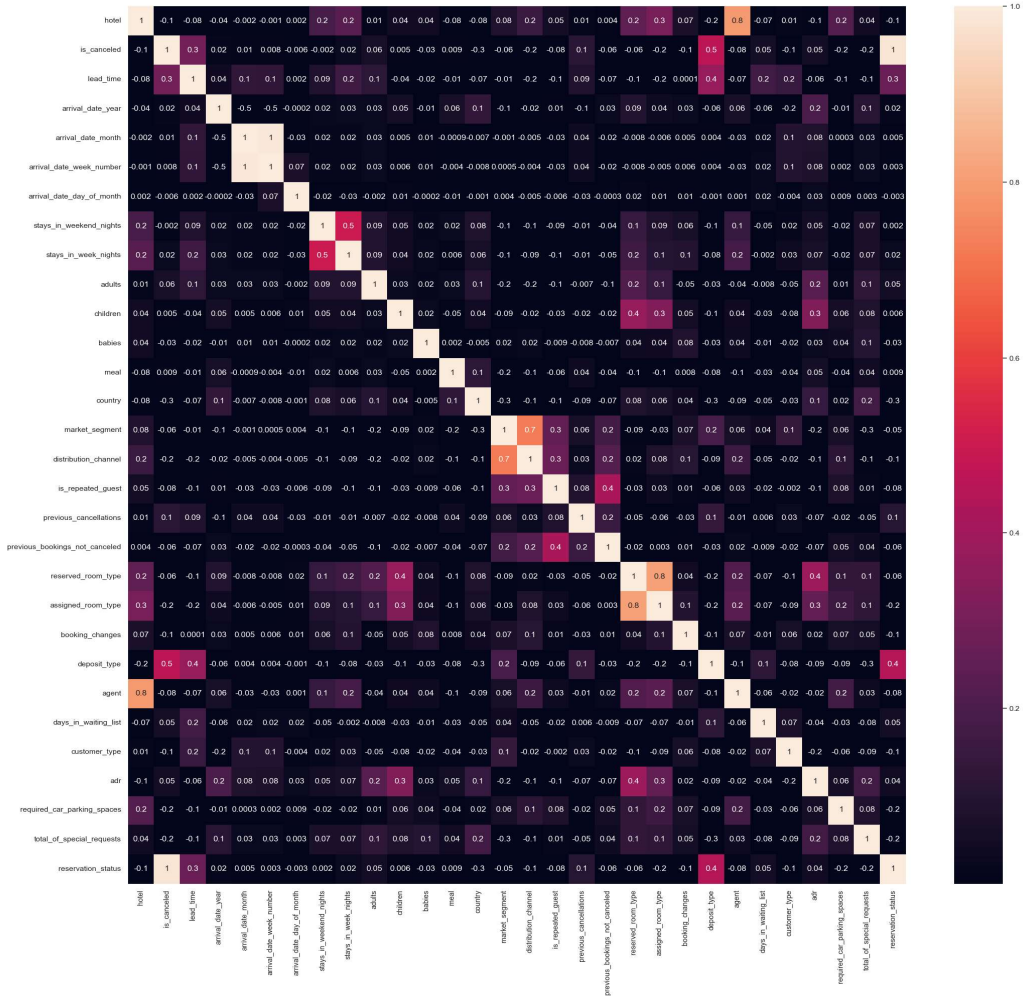


图一

2、特征工程

对 32 个变量做相关性分析，相关性热力图如下（原图在附件中给出）：

[1] 数据集来源：https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand



图二

从图二可以看出，“hotel”、“stay_in_weekend_nights”、“stay_in_week_nights”、“children”、“reserved_room_type”这几个字段与登记何种房型相关性较大。其中，“预定的房型”和“登记的房型”相关性达到 80%，可见大部分客人在预定房型后，不会再更改房型。

三、 数据预处理

1、 缺失值填充

由于“Children”、“Country”和“Agent”都是类别属性，数据类型为字符型，所以选择“根据上下条数据填充”的方法填充，而“Company”缺失比例过高，无法为预测提供足够多的有效信息，故删去。

2、 数据编码

对于二元属性“hotel”，用“0”表示城市酒店，“1”表示郊区酒店。对于“到达日期所在月份”，由于月份是有时间前后关系的类别，故用数值映射的方法，用“1~12”表示 1-12 月。对于取值之间没有大小的意义的类别属性，用 one-hot 编码将其特征数字化。特别地，“reservation_status_date”是用“2017-01-01”的形式表示“预定状态所在日期”，需要转化成单独的年、月、日三列。

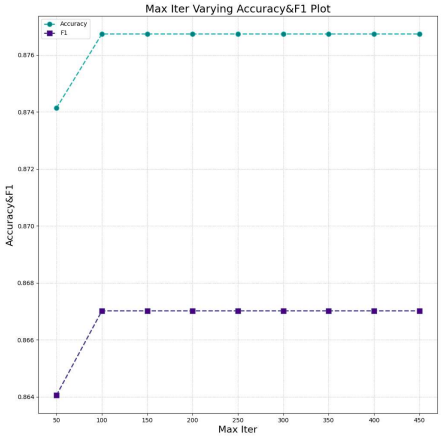
四、 建模

在机器学习的算法选择方面，选择逻辑回归算法、决策树算法，和随机森林算法（作为集成学习的代表）来预测客人登记的房间类型，并讨论算法的各个参数对算法性能的影响。

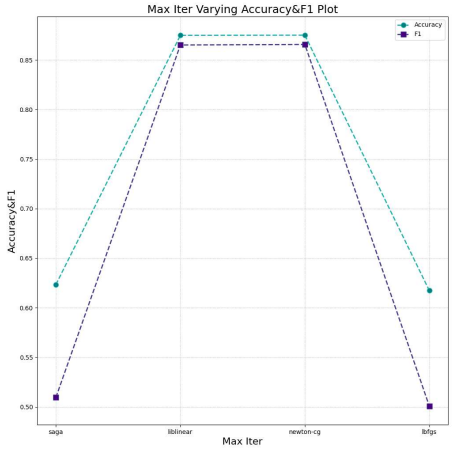
1、 逻辑回归（Logistic Regression）

（1） 模型迭代次数（max_iter）

控制正则化参数（penalty）使用 12 正则化，损失函数优化器使用“liblinear”，探究随着模型迭代次数的增加，模型性能的变化。由图三可见，迭代次数在 100 次后，Accuracy Score 稳定在 0.876，F1 Score 稳定在 0.874。



图三



图四

（2） 损失函数优化器（solver）

控制正则化参数使用 12 正则化，模型迭代次数在 100 次，探究不同的损失函数优化器对模型性能的影响。

优化器选择的猜想：

本数据集的样本数量在 10 万以上，属于非常大的数据集，所以可以选择在大数据集上速度更快的快速梯度下降法（saga），但是 saga 对不平衡数据集的鲁棒性不强；

从预测值的类别比例上看，本数据集非常不平衡，所以可选择对不平衡数据集的鲁棒性强的坐标轴下降法（liblinear）、牛顿法（newton-cg）和拟牛顿法（lbfgs），但是三者的速度不如 saga。

实验结果如下：

	saga	liblinear	newton-cg	lbfgs
Accuracy Score	0.661	0.8748	0.8750	0.617
F1 Score	0.572	0.8650	0.8656	0.501
Faster	206.0s	54.5s	1847.5	19.4s
Robust to unscaled datasets	No	Yes	Yes	No

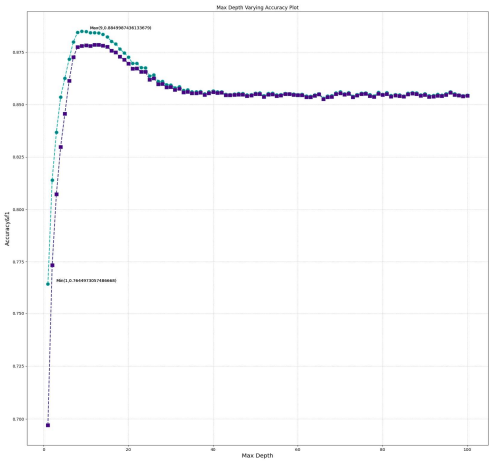
表一

由图四和表一可见，liblinear 和 newton-cg 的运行时间虽长，但评分 saga 和 lbfgs 高 20~30%，鲁棒性高，而 newton-cg 的运行时间大约是 liblinear 的 34 倍，所以应选 liblinear 作为模型的参数。

2、 决策树（Decision Tree Classifier）

（1） 最大深度（max_depth）

最大深度是影响决策树模型性能的重要指标。控制其他因变量不变，探究随着决策树最大深度的增加，模型性能的变化。结果如下：



图五

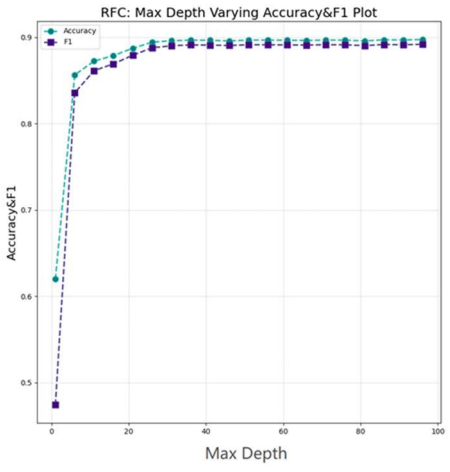
可见最大深度在 1~7 左右时，模型性能上升但不佳，原因是欠拟合；最大深度在 8~15 左右时，模型的性能最好，Accuracy Score 可达到 0.885，F1 Score 可达到 0.878。大于 15 之后，模型的性能下降，原因是过拟合。

此外，“min_samples_leaf”和“min_samples_split”对决策树的性能影响较微，对随机森林的性能影响较明显，故放到随机森林部分讨论。

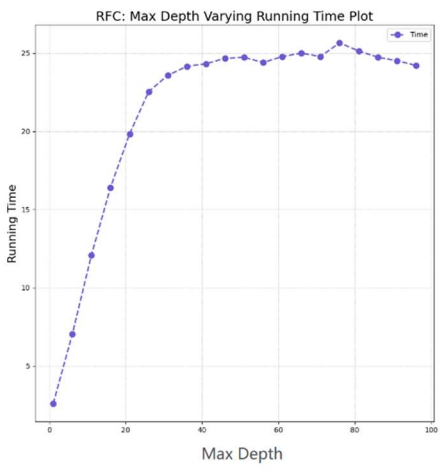
3、随机森林

（1） 最大深度（max_depth）

首先探究最大深度如何影响随机森林模型的性能，设置最大深度的范围从 1~100，步长为 5：



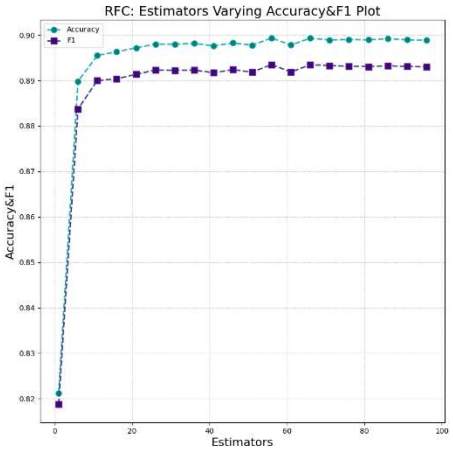
图六



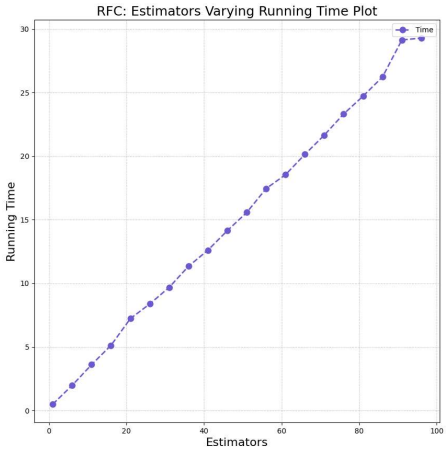
图七

由上图可见，不同于决策树在最大深度为 8~15 时 Accuracy Score 和 F1 Score 最大，随机森林在最大深度>30 才出现评分的最大值，而时间也在最大深度>30 后稳定在 25s 左右。

（2）分类器个数（n_estimators）



图八



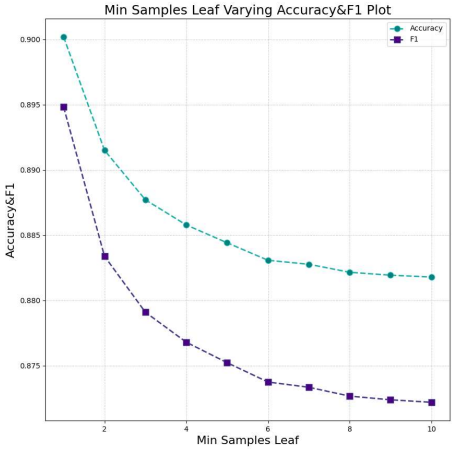
图九

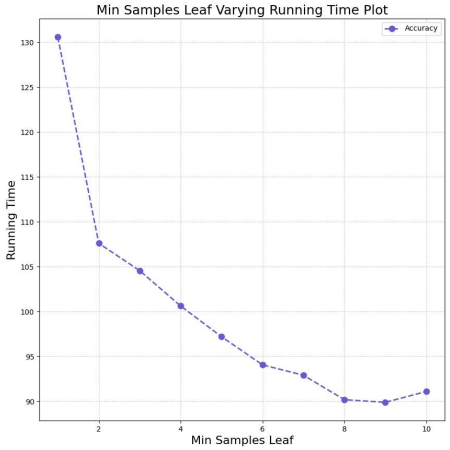
分类器个数是影响随机森林模型性能的主要指标。根据上一部分的结果，控制每个分类器的最大深度为 35，探究随着分类器个数的增加，模型性能指标的变化,设置分类器个数在 5~100 个，步长为 5。结果显示，当分类器个数在 80 个以上时，算法的 accuracy score 基本稳定在 0.899 左右，f1 score 基本稳定在 0.899 左右。当分类器个数达到 80 个以上时，效果没有显著提升。而随着分类器个数的增多，占用的内存与训练/预测时间也会增多。右图所示，训练/预测时间随分类器个数的增加呈线性增长。为了节省资源，本数据集的随机森林算法的分类器个数可选择在 80 个左右。

（3） “min_samples_leaf”& “min_samples_split”

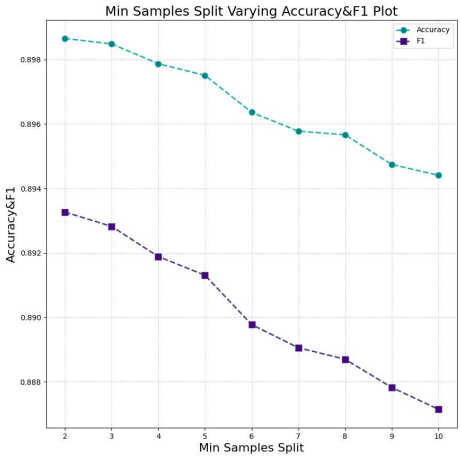
“生成的叶节点最小样本数”指分枝后的每个子节点都必须包含至少 x 个训练样本，因此分枝会朝着满足每个子节点都包含 x 个样本的方向去发生。而“被分枝的节点最小样本数”指一个节点必须要包含至少 x 个训练样本，这个节点才允许被分枝。“min sample leaf”最小值为 1，“min sample split”最小值为 2，此种情况模型有最高复杂度。对于特征量大的数据来说，为了防止过拟合，必须适当增大两个数值。

一般来说，“min_samples_leaf”和“min_samples_split”的数值越小，决策树辨别越精细，运行时间越长，这点在单个决策树上体现的不明显，但是在随机森林算法中，由于随机森林由若干个决策树组成，运行时间成倍叠加，效果较为明显。实验结果如下：

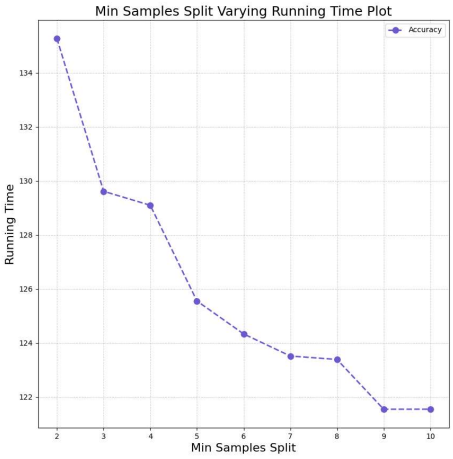




图十



图十一



图十二　　图十三

结果显示，生成的叶节点最小样本数为 1，被分枝的节点最小样本数为 2 时，模型性能最好，说明本数据集特征数量不大（31 个），不容易发生过拟合的情况。而生成的叶节点最小样本数为 1 时时间最长，约为 130s，样本数为 9 时时间最短，为 90s，最长时间是最短时间的 1.4 倍。

综上所述，我们选择参数为{n_estimators=80,max_depth=35,min_samples_split=2,min_samples_leaf=1,max_features='sqrt',bootstrap=False}的随机森林分类器，Accuracy Score 可达到 0.900，F1 Score 可达到 0.899，单次运行时间为 24.8s。

五、 结果评价及分析

用准确率评分（Accuracy Score）和 F1 分数（F1 Score）作为评价指标，对比三个算法的优劣，并分析原因。结果如下：

	逻辑回归	决策树	随机森林	SVM（对照实验）
Accuracy Score	0.875	0.887	0.900	0.619
F1 Score	0.866	0.878	0.899	0.474
Time	54.5s	2.5s	24.8s	1800+s
Robust to unscaled datasets	Yes	Yes	Yes	No

表二

由表二可见，评分由高到低分别是随机森林＞决策树＞逻辑回归，时间的由快到慢分别是决策树＜随机森林＜逻辑回归。可能的原因是：（1）比较逻辑回归和决策树，逻辑回归对极值比较敏感，容易受极端值的影响，而决策树在这方面表现较好，而本数据集的各类样本分布极不均匀，有的类只有几个或十几个样本，因此逻辑回归算法可能受此影响。此外，逻辑回归较为擅长处理线性特征，决策树较为擅长处理非线性特征，在本数据集的 32 个特征中，线性特征有 13 个，非线性特征有 19 个，所以决策树可能比较适合本任务。（2）比较决策树和随机森林，由于随机森林采用集成算法，由每棵树投票选取最终结果，所以精度上优于单模型的决策树。从训练时间上说，决策树的时间复杂度是 O（n*log(n)*d），随机森林的是 O（n*log(n)*d*k）（k 是分类器个数），所以随机森林的运行时间比决策树稍长。