**山东科技大学**

**实验报告**

**课程名称： 大数据分析方法及应用实验**

**实验项目： 基于集成学习的酒店预订订单取消预测研究**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# 目录

[目录 2](#_Toc8247)

[引言​​ 3](#_Toc23353)

[文献综述 4](#_Toc8786)

[1. 算法介绍 5](#_Toc27796)

[1.1 基本概念 5](#_Toc22363)

[1.1.1 Adaboost 5](#_Toc468)

[1.1.2 Adaboost算法实现 5](#_Toc11181)

[1.1.3 随机森林 6](#_Toc13639)

[1.1.4 随机森林算法实现 6](#_Toc14907)

[2.实例分析 7](#_Toc24707)

[2.1 数据来源 7](#_Toc16111)

[2.2数据描述 7](#_Toc7635)

[2.3数据预处理 8](#_Toc2272)

[2.4数据分析 8](#_Toc3584)

[2.5集成学习 10](#_Toc21373)

[2.5.1随机森林 10](#_Toc19806)

[2.5.2 Adaboost​ 11](#_Toc26083)

[2.5.3综合比较 12](#_Toc6727)

[2.5.4结果分析 13](#_Toc4591)

[3.结论 14](#_Toc108)

[参 考 文 献 15](#_Toc14538)

[附录 16](#_Toc14238)

引言**​**​

随着旅游和酒店业的迅速发展，客户通过网络平台提前预订酒店房间已成为常态。然而，预订订单的高取消率给酒店的收益管理与运营规划带来了极大挑战。如何有效预测客户是否会取消订单，已成为提升酒店收益和优化资源配置的重要课题。近年来，随着大数据和机器学习技术的广泛应用，基于结构化业务数据的客户行为预随着数字化和互联网经济的深入发展，酒店行业的运营模式正在发生深刻变化。客户通过在线平台提前预订酒店房间已成为主要趋势，而随之而来的高比例预订取消现象也对酒店的收益管理和资源配置提出了新的挑战。订单取消不仅造成客房空置率上升，还会带来收益波动，影响服务质量与客户满意度。因此，如何基于历史预订数据和客户行为信息，有效预测订单的取消概率，成为现代酒店管理的重要研究方向之一。

近年来，随着大数据和人工智能技术的广泛应用，数据驱动的客户行为分析逐渐成为提升酒店核心竞争力的关键手段。结构化的酒店预订数据包含了丰富的客户属性、预订习惯、价格信息、渠道特征等多维度变量，为深入挖掘订单取消的影响因素和规律提供了数据基础。通过机器学习等智能算法对这些数据进行建模和分析，不仅能够显著提升取消预测的准确率，还能为酒店收益管理、动态定价和差异化营销策略制定提供科学决策依据。

Kaggle 平台发布的 Hotel Booking Demand Dataset 收录了来自两类酒店的大量真实预订记录，涵盖了从预订行为到实际入住的全过程信息。利用该数据集开展订单取消预测实验，能够在真实业务背景下考察不同算法的性能表现与实际适用性。集成学习方法如随机森林（Random Forest）和Adaboost 近年来在客户行为预测领域表现突出，具有较强的泛化能力和特征解释性。因此，本研究基于该数据集，结合典型业务特征，构建并比较两种集成学习模型，系统分析其在酒店预订取消预测任务中的优势与不足，为行业应用提供理论与实践参考。

文献综述

随着酒店行业竞争日趋激烈，基于数据挖掘和机器学习的客户行为分析成为研究和实践的热点。现有研究指出，订单取消行为与客户个体属性（如国籍、家庭结构）、预订行为（如预订提前期、特殊需求）、渠道特征（如直销、代理）以及定价策略等多种因素密切相关[1][2]。通过系统性分析和建模，不仅能够有效挖掘取消风险高的客户特征，还可以针对性地优化库存管理与收益控制流程[3]。

在酒店预订取消预测领域，传统方法多采用逻辑回归、决策树等基础统计和机器学习模型，但在应对高维、复杂和非线性特征时表现有限。近年来，集成学习方法如随机森林（Random Forest）和Adaboost 受到广泛关注。Breiman（2001）提出的随机森林算法通过集成多棵决策树并引入特征扰动机制，有效降低了单棵树的过拟合风险，提升了模型的稳定性和泛化能力[4]。Freund与Schapire（1997）提出的Adaboost 算法则通过动态调整样本权重，增强对难分类样本的关注，尤其在类别不平衡场景下表现出较强的识别能力[5]。

应用层面，Moreno-Izquierdo等（2019）综合评估了多种机器学习模型在旅游订单流失预测中的表现，证实了集成算法在提升预测精度、增强业务可解释性方面的优势[6]。Zhang等（2021）利用大规模酒店订单数据，对比分析了随机森林、Adaboost 等集成模型与单一模型的性能差异，发现集成方法在特征丰富和样本异质性强的场景下具有更优表现[7]。此外，一些研究结合大数据平台与实时建模，实现了酒店预订取消风险的在线预测与动态管理，有效支持了酒店的收益优化和资源调度[8]。

总体来看，集成学习模型凭借其在高维特征空间中的鲁棒性、对复杂非线性关系的刻画能力以及较好的可扩展性，已成为酒店预订取消预测及客户流失分析中的主流技术路线。结合行业实际数据进行定制化建模，不仅能够提升酒店经营的科学化水平，也为智能旅游和服务个性化带来新机遇。

# 1.算法介绍

1.1 基本概念

集成学习通过建立几个模型组合的来解决单一预测问题。它的工作原理是生成多个分类器，各自独立地学习和作出预测。这些预测最后结合成单预测，因此优于任何一个单分类的做出预测。

　　装袋、提升、随机森林是集成学习的子类，它依靠于决策树的投票选择来决定最后的分类结果。

1.1.1 Adaboost

给定数据集D，它包含d个类标记的元组(，)，(, )，..，. (, )，其中是元组的类标号。

首先，Adaboost 对每个训练元组赋予相等的权重1/d。为组合分类器产生k个基分类器需要执行算法的其余部分k轮。在第i轮，从D中有放回抽取大小为d的训练集Di。同一个元组可能被选中多次。每个元组被选中的机会由它的权重决定。从训练集Di导出分类器Mi。然后使用Di作为检验集计算Mi的误差。训练元组的权重根据它们的分类情况调整。

为了计算模型Mi的错误率，求Mi误分类Di中的每个元组的加权和。即，其中，是元组j的误分类误差:如果被误分类，则为1;否则，它为0。

如果分类器的性能太差，错误率超过0.5，则丢弃它，并重新产生新的训练集，由它导出新的。

1.1.2 Adaboost算法实现

输入： D：类标记的训练元组集。

k：轮数（每轮产生一个分类器）。

一种分类学习方案。

输出：一个复合模型。

(1) 将D中每个元组的权重初始化为1/d；  
(2) for i = 1 to k do // 对于每一轮  
(3) 根据元组的权重从D中有放回抽样，得到Di；  
(4) 使用训练集Di导出模型Mi；  
(5) 计算Mi的错误率error(Mi)（8.34式）  
(6) if error(Mi) > 0.5 then  
(7) 转步骤(3)重试；  
(8) endif  
(9) for Di的每个被正确分类的元组do  
(10) 元组的权重乘以error(Mi)/(1-error(Mi))； // 更新权重  
(11) 规范化每个元组的权重；  
(12) end for

使用组合分类器对元组x分类：

(1) 将每个类的权重初始化为0；

(2) for i = 1 to k do // 对于每个分类器

(3) wi = log((1-error(Mi))/error(Mi))； // 分类器的投票权重

(4) c = Mi(x)； // 从Mi得到x的类预测

(5) 将wi加到类c的权重；

(6) end for

(7) 返回具有最大权重的类；

1.1.3 随机森林

a. 给定一个训练样本集，数量为N，使用有放回采样到N个样本，构成一个新的训练集。注意这里是有放回的采样，所以会采样到重复的样本。

b.从总量为M的特征向量中，随机选择m个特征（m<M），其中m可以等于 sqrt(M)，然后计算m个特征的信息增益，选择最优特征（属性）。注意，这里的随机选择特征是无放回的选择。（在整个森林的生长过程中，m的值一般维持不变）

c.有了上面随机产生的样本集，就可以使用一般决策树的构建方法，得到一棵分类（或者预测）的决策树。需要注意的是，在计算节点最优分类特征的时候，要使用b中的随机选择特征方法。

d.通过以上三步，可以得到一棵决策树，重复这样的过程H次，就得到了H棵决策树。然后来了一个测试样本，就可以用每一棵决策树都对它分类一遍，得到了H个分类结果。这时，使用简单的投票机制，或者该测试样本的最终分类结果来判别该样本的所属类别。

1.1.4 随机森林算法实现

输入： D：训练样本集（N个样本）  
 H：决策树数量（森林规模）  
 M：特征总数  
 m：每棵树分裂时随机选择的特征数（一般为 sqrt(M)）

输出：一个由多棵决策树组成的复合模型。

(1) for h = 1 to H do // 生成H棵决策树  
(2) 从训练集D中有放回随机采样N个样本，构成新训练集Dh；  
(3) 在每个决策树节点分裂时，从M个特征中无放回随机选择m个特征（一般 m = sqrt(M)）；  
(4) 在m个特征中，计算各自的信息增益（或Gini系数），选择最佳特征进行分裂；  
(5) 使用一般决策树算法递归构建决策树，直到满足停止条件；  
(6) end for  
(7) 得到H棵决策树，组成随机森林模型。

预测阶段：  
(1) 对于测试样本x，将x输入到每棵决策树中，得到H个分类结果；  
(2) 统计所有决策树的分类结果，采用投票法，票数最多的类别作为最终输出类别。

# 2.实例分析

# 2.1数据来源

本研究所使用的数据集来自于Kaggle平台上的 Hotel Booking Demand Dataset，由数据贡献者 Jesse Mostipak 提供。该数据集详细收录了两类酒店（Resort Hotel和City Hotel）自2015年至2017年期间的真实预订记录，涵盖了客户基本信息（如国籍、家庭成员数量、预订渠道等）、预订行为（提前预订天数、入住天数、是否回头客等）、订单经济信息（每日房价、押金类型、特殊需求等）以及订单结果（是否取消、实际入住或爽约等）等丰富内容。数据集共包含超过11万条预订数据，为订单取消行为分析、客户细分与预测建模等数据挖掘任务提供了全面、可靠的数据基础。该数据集被广泛应用于旅游管理、酒店收益优化、客户行为建模和机器学习算法实证等多个领域。

2.2数据描述

表1：数据变量描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **变量名称** | **含义** | **类型** | **单位** | **示例值** |
| **hotel** | 酒店类型 | 分类型 | - | Resort Hotel |
| **is\_canceled** | 是否取消订单 | 数值型 | 0/1 | 0 |
| **lead\_time** | 提前预订天数 | 数值型 | 天 | 342 |
| **arrival\_date\_year** | 到店年份 | 数值型 | 年 | 2015 |
| **arrival\_date\_month** | 到店月份 | 分类型 | - | July |
| **arrival\_date\_week\_number** | 到店周数 | 数值型 | 周 | 27 |
| **arrival\_date\_day\_of\_month** | 到店日 | 数值型 | 日 | 1 |
| **stays\_in\_weekend\_nights** | 周末入住夜数 | 数值型 | 晚 | 0 |
| **stays\_in\_week\_nights** | 工作日入住夜数 | 数值型 | 晚 | 0 |
| **adults** | 成人数量 | 数值型 | 人 | 2 |
| **children** | 儿童数量 | 数值型 | 人 | 0 |
| **babies** | 婴儿数量 | 数值型 | 人 | 0 |
| **meal** | 订餐类型 | 分类型 | - | BB |
| **country** | 客户国籍 | 分类型 | - | PRT |
| **market\_segment** | 市场渠道 | 分类型 | - | Direct |
| **distribution\_channel** | 分销渠道 | 分类型 | - | Direct |
| **is\_repeated\_guest** | 是否回头客 | 数值型 | 0/1 | 0 |
| **previous\_cancellations** | 先前取消订单次数 | 数值型 | 次 | 0 |
| **previous\_bookings\_not\_canceled** | 先前未取消订单次数 | 数值型 | 次 | 0 |
| **reserved\_room\_type** | 预订房型 | 分类型 | - | C |
| **assigned\_room\_type** | 实际分配房型 | 分类型 | - | C |
| **booking\_changes** | 订单变更次数 | 数值型 | 次 | 3 |
| **deposit\_type** | 押金类型 | 分类型 | - | No Deposit |
| **agent** | 代理人编号 | 分类型 | - | NULL/304 |
| **company** | 公司编号 | 分类型 | - | NULL |
| **days\_in\_waiting\_list** | 等待名单天数 | 数值型 | 天 | 0 |
| **customer\_type** | 客户类型 | 分类型 | - | Transient |
| **adr** | 平均每日房价 | 数值型 | 欧元 | 0 / 75 |
| **required\_car\_parking\_spaces** | 预订停车位数 | 数值型 | 个 | 0 |
| **total\_of\_special\_requests** | 特殊需求数量 | 数值型 | 个 | 0 |
| **reservation\_status** | 订单最终状态 | 分类型 | - | Check-Out |
| **reservation\_status\_date** | 状态变更日期 | 分类型 | 日期 | 2015/7/1 |

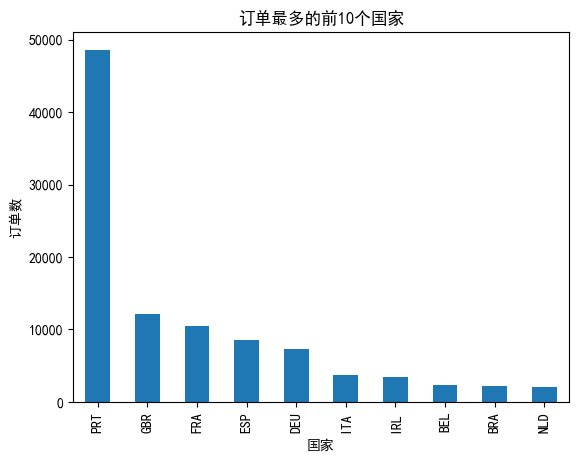
2.3数据预处理

在数据预处理阶段，首先筛选出与订单取消预测密切相关的核心特征，包括酒店类型、预订提前天数、到店日期、入住天数、客户结构、订餐类型、客户国籍、市场渠道、分销渠道、是否回头客、历史订单情况、房型信息、订单变更次数、押金类型、客户类型、平均每日房价、预订停车位数量以及特殊需求数量，并以订单是否取消作为目标变量。针对上述特征，逐一检测并剔除所有存在缺失值的记录，确保后续分析和建模样本的完整性。

本次数据集在预处理前共包含119390条样本记录。经缺失值处理后，剩余118898条完整样本，共剔除492条存在缺失值的无效记录。最终获得无缺失值的标准化建模数据集，为随机森林和Adaboost等集成学习方法的训练与评估提供了统一、可靠的数据基础。

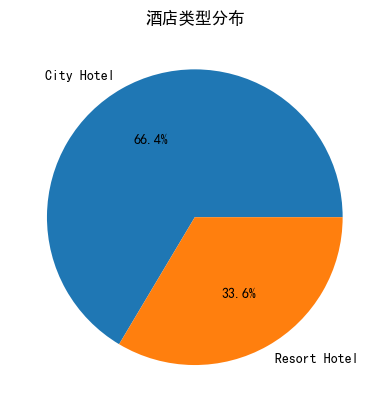
2.4数据分析

图1：订单来源国家



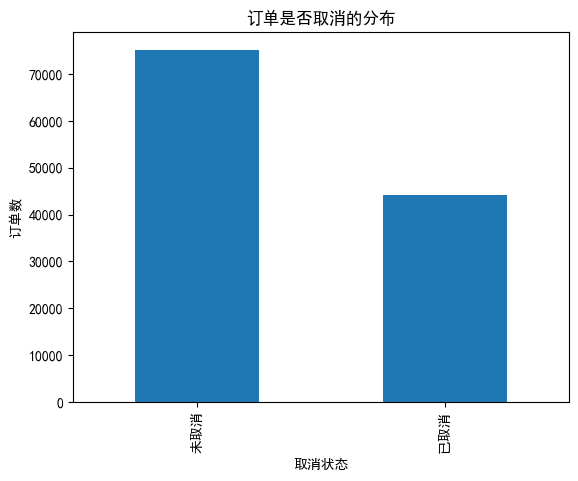
从上图可以看出，酒店预订订单主要集中在少数几个国家。其中，葡萄牙（PRT）客户的订单量遥遥领先，超过4.8万单，远高于其他国家。英国（GBR）和法国（FRA）分列第二、三位，订单量分别在1万单左右。其余如西班牙（ESP）、德国（DEU）、意大利（ITA）、爱尔兰（IRL）、比利时（BEL）、巴西（BRA）、荷兰（NLD）等国家，订单量也较为可观，但与葡萄牙相比差距较大。这一分布表明，酒店的主要客源国集中度较高，尤其是本地（葡萄牙）客户占据了绝对优势，这对于后续的客户结构分析和市场定位具有重要参考意义。

图2：酒店类型扇形图



从上图可以看出，本数据集中的订单主要来自两类酒店：城市酒店（City Hotel）和度假酒店（Resort Hotel）。其中，城市酒店订单占比为66.4%，度假酒店订单占比为33.6%。这说明，样本中城市酒店的订单量约为度假酒店的两倍，城市酒店业务在整个数据集中占据主导地位。不同类型酒店的客源结构和业务特点可能存在显著差异，为后续深入分析不同酒店类型的订单取消行为和特征模式提供了基础。

图3：订单是否取消条形图



从上图可以看出，未取消的订单数量显著高于已取消的订单，约占全部订单的三分之二左右。尽管大多数客户最终完成了入住，但仍有较大比例的订单出现了取消情况。这说明在实际酒店运营过程中，订单取消现象较为普遍，酒店管理者需要关注和应对这一问题。订单是否取消这一目标变量具有明显的类别分布差异，为后续的分类建模和影响因素分析提供了坚实的数据基础。

2.5集成学习

2.5.1随机森林

**准确率​**​: 88.51%

**分类报告​**​:

表2：随机森林分类性能指标表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类别** | **精确率** | **召回率** | **F1分数** | **支持数** |
| **0** | 0.89 | 0.93 | 0.91 | 14,851 |
| **1** | 0.88 | 0.80 | 0.84 | 8,929 |
| **​**​准确率​**​** | - | - | 0.88 | 23,780 |
| **​**​宏平均​**​** | 0.88 | 0.87 | 0.88 | - |
| **​**​加权平均​**​** | 0.88 | 0.89 | 0.88 | - |

**混淆矩阵​**​:、

图4：随机森林混淆矩阵

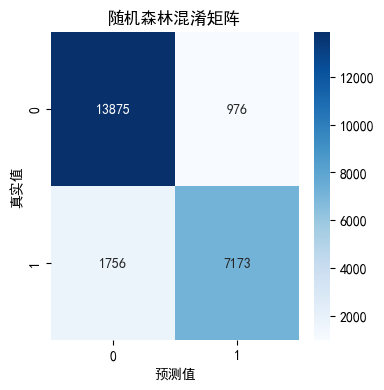
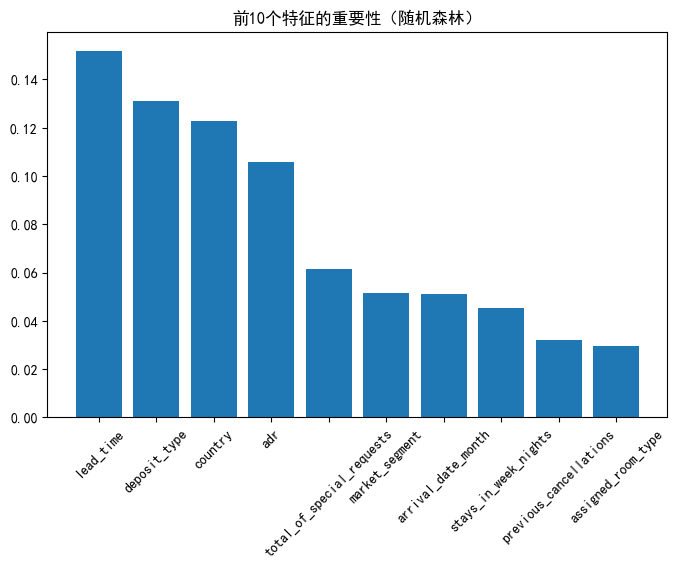


图5：随机森林变量重要性条形图



2.5.2 Adaboost**​**

**准确率​**​: 81.51%

**分类报告​**​:

表3：Adaboost分类性能指标表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类别** | **精确率** | **召回率** | **F1分数** | **支持数** |
| **0** | 0.81 | 0.92 | 0.86 | 14,851 |
| **1** | 0.83 | 0.64 | 0.72 | 8,929 |
| **​**​准确率​**​** | - | - | 0.82 | 23,780 |
| **​**​宏平均​**​** | 0.82 | 0.79 | 0.81 | - |
| **​**​加权平均​**​** | 0.82 | 0.82 | 0.81 | - |

**混淆矩阵​**​:

图6：Adaboost混淆矩阵

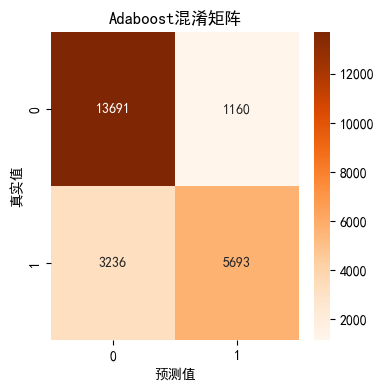
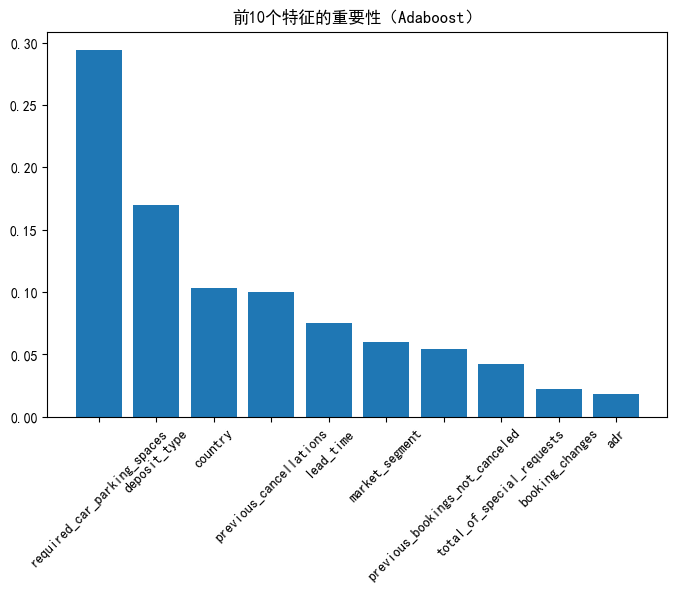
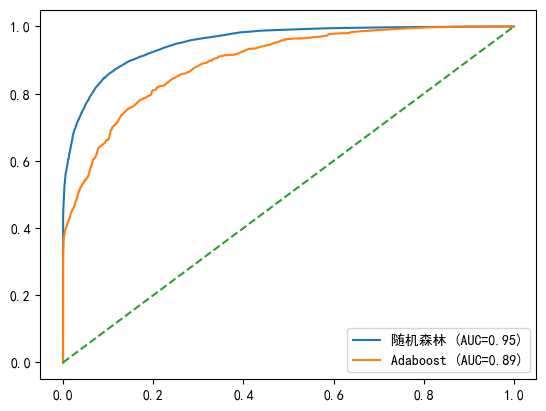


图7：Adaboost重要性条形图



2.5.3综合比较

图7：随机森林与AdaBoost模型ROC曲线对比图



上图为随机森林与Adaboost模型在酒店订单取消预测任务中的ROC曲线对比。从图中可以看出，随机森林模型的ROC曲线始终位于Adaboost曲线之上，并且二者均明显高于对角线，说明两种模型均具备较好的二分类判别能力。其中，随机森林模型的AUC值达到0.95，显著高于Adaboost的AUC值0.89。这表明，随机森林模型在所有分类阈值下对“已取消”与“未取消”订单的区分能力更强，整体预测性能更优。

综合AUC指标与前述准确率、召回率等结果，随机森林模型在本数据集上具有更高的应用价值，更适合用于实际的订单取消风险预警与管理场景。

2.5.4结果分析

首先，从特征重要性分析来看，随机森林模型中影响订单取消预测最显著的特征依次为预订提前天数、押金类型、客户国籍、平均每日房价、特殊需求数量等。其中，预订提前天数权重最高，说明客户越早预订，订单被取消的概率变化趋势越明显。押金类型和客户国籍也对客户的取消行为具有重要影响。Adaboost模型则将所需停车位数的重要性排在首位，押金类型、客户国籍和历史取消行为等特征同样占有较大比重。这表明两种模型均认为押金类型、客户国籍和订单历史行为是影响订单取消概率的核心因素，但随机森林更偏重价格和行为变量，Adaboost则更关注客户出行需求与历史行为。由此可见，酒店在制定管理策略时，既需关注价格和押金政策的调整，也应重视客户的出行实际需求和过往行为模式。

其次，从模型性能评估来看，随机森林在订单取消预测任务中表现最佳，准确率达到88.51%，对未取消订单的召回率为0.93，F1分数为0.91；对已取消订单的召回率为0.80，F1分数为0.84，整体表现优异。Adaboost准确率为81.51%，未取消订单的召回率为0.92，F1分数为0.86，但在已取消订单的识别能力上（召回率0.64，F1分数0.72）明显低于随机森林。混淆矩阵分析也进一步印证了这一结论，随机森林对两类样本识别更为均衡，误判率较低。

综合来看，基于集成学习的订单取消预测模型在酒店行业具有较高的应用价值。随机森林模型能够更全面、准确地识别高风险订单，为酒店提前预警和主动管理提供了可靠的技术支持。特征重要性分析结果也为酒店管理者优化业务策略、提升运营效率和客户体验提供了数据支撑。建议酒店管理者在制定收益管理和客户服务策略时，重点关注模型揭示的核心变量，有针对性地开展风险干预和资源调配，进一步提升整体经营水平。同时，不同算法可组合应用，根据实际业务场景灵活选用，不断完善订单管理与客户服务体系。

# 3.结论

本研究基于Kaggle公开的葡萄牙酒店订单数据集，系统开展了订单取消预测的建模分析。通过对数据的预处理、特征筛选与编码、样本可视化与探索性分析，采用随机森林与Adaboost两种集成学习方法对订单取消进行了分类预测，并从模型性能、特征重要性和业务意义等角度进行了系统对比和讨论。

实验结果显示，酒店订单的取消行为受多种因素综合影响，其中预订提前天数、押金类型、客户国籍、历史订单行为、每日均价、特殊需求等变量在模型中具有较高的特征重要性。随机森林模型在预测准确率、F1分数等各项指标上均优于Adaboost，尤其在识别高风险（已取消）订单方面更具优势，能够为酒店运营中的风险控制和收益优化提供更为可靠的技术手段。Adaboost模型则在部分特征（如停车需求、押金类型、客户历史行为）上表现出较高敏感度，可以作为补充分析手段，辅助深入挖掘特定客户群体或订单特征。

通过本研究的系统分析，验证了集成学习方法在实际酒店运营数据中的有效性与应用价值。基于模型输出的特征重要性排序，酒店管理者可以重点关注那些对取消风险影响较大的变量，如优化预订政策、调整押金与价格策略、加强对重点客户群体的管理和服务，提升整体订单履约率和客户满意度。

未来展望

首先，可以尝试引入更多外部数据源（如天气、节假日、经济指标、疫情信息等），提升模型的业务场景适应性和预测精度。

其次，可以采用更为先进的深度学习模型或集成多模型融合方法，对复杂非线性特征和更大规模数据进行深入挖掘。

第三，在模型应用层面，可结合实时数据流分析和动态建模，实现对订单取消风险的动态预警和在线干预。

最后，建议结合可解释性AI等新技术，进一步提升模型的透明度和业务决策支持能力，助力酒店行业智能化、精细化管理的不断升级。

综上所述，基于集成学习的订单取消预测模型不仅为酒店管理实践提供了有效的数据分析工具，也为大数据与人工智能在旅游与服务业的深度融合探索了可行路径，具有良好的理论意义与现实应用前景。

**参 考 文 献**

[1] T. Kuo, S. Chen, Y. Lin. Application of machine learning techniques to predict cancellations of hotel bookings. Sustainability, 2019, 11(13): 3459.  
[2] Miguéis V. L., Camanho A. S., e Cunha J. F. Customer attrition in retailing: A comparison of forecasting methods. European Journal of Operational Research, 2012, 217(2): 275-285.  
[3] S. Pan, D. Yang, et al. Predicting and Understanding Hotel Booking Cancellations with Machine Learning. Journal of Hospitality & Tourism Research, 2021, 45(1): 119-140.  
[4] Breiman L. Random Forests. Machine Learning, 2001, 45(1): 5-32.  
[5] Freund Y, Schapire R E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 1997, 55(1): 119-139.  
[6] L. Moreno-Izquierdo, R. Perles-Ribes, J. Such-Devesa. Impact of big data and artificial intelligence on the tourism industry: A review of literature. International Journal of Contemporary Hospitality Management, 2019, 31(3): 1171-1190.  
[7] Zhang, W., Zhang, X., Hu, C. Predicting Hotel Booking Cancellation with Machine Learning Approaches. Information, 2021, 12(7): 273.  
[8] Lee J., Jang S. Real-time hotel booking cancellation prediction using streaming big data. International Journal of Hospitality Management, 2020, 89: 102525.

**附录**

import pandas as pd

import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

df = pd.read\_csv('hotel\_bookings.csv')

# 目标变量分布

cancel\_counts = df['is\_canceled'].value\_counts()

cancel\_counts.plot(kind='bar')

plt.xticks([0, 1], ['未取消', '已取消'])

plt.title('订单是否取消的分布')

plt.xlabel('取消状态')

plt.ylabel('订单数')

plt.show()

# 国家分布Top 10

df['country'].value\_counts().head(10).plot(kind='bar')

plt.title('订单最多的前10个国家')

plt.xlabel('国家')

plt.ylabel('订单数')

plt.show()

# 酒店类型分布

df['hotel'].value\_counts().plot(kind='pie', autopct='%1.1f%%')

plt.title('酒店类型分布')

plt.ylabel('')

plt.show()

import pandas as pd

df = pd.read\_csv('hotel\_bookings.csv')

features = [

'hotel', 'lead\_time', 'arrival\_date\_year', 'arrival\_date\_month', 'stays\_in\_weekend\_nights',

'stays\_in\_week\_nights', 'adults', 'children', 'babies', 'meal', 'country', 'market\_segment',

'distribution\_channel', 'is\_repeated\_guest', 'previous\_cancellations', 'previous\_bookings\_not\_canceled',

'reserved\_room\_type', 'assigned\_room\_type', 'booking\_changes', 'deposit\_type', 'customer\_type',

'adr', 'required\_car\_parking\_spaces', 'total\_of\_special\_requests'

]

label = 'is\_canceled'

# 原始数据行数

print("原始数据形状：", df[features + [label]].shape)

# 删除缺失

df\_clean = df[features + [label]].dropna(subset=features + [label])

print("删除缺失后数据形状：", df\_clean.shape)

print("共删除样本数：", df.shape[0] - df\_clean.shape[0])

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

# 分类特征编码

categorical\_cols = [

'hotel', 'arrival\_date\_month', 'meal', 'country', 'market\_segment',

'distribution\_channel', 'reserved\_room\_type', 'assigned\_room\_type',

'deposit\_type', 'customer\_type'

]

df\_model = df\_clean.copy()

for col in categorical\_cols:

df\_model[col] = LabelEncoder().fit\_transform(df\_model[col])

# 特征与标签

features = [

'hotel', 'lead\_time', 'arrival\_date\_year', 'arrival\_date\_month', 'stays\_in\_weekend\_nights',

'stays\_in\_week\_nights', 'adults', 'children', 'babies', 'meal', 'country', 'market\_segment',

'distribution\_channel', 'is\_repeated\_guest', 'previous\_cancellations', 'previous\_bookings\_not\_canceled',

'reserved\_room\_type', 'assigned\_room\_type', 'booking\_changes', 'deposit\_type', 'customer\_type',

'adr', 'required\_car\_parking\_spaces', 'total\_of\_special\_requests'

]

label = 'is\_canceled'

X = df\_model[features]

y = df\_model[label]

# 划分训练集和测试集

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 随机森林

rf = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf.fit(X\_train, y\_train)

y\_pred\_rf = rf.predict(X\_test)

print("随机森林结果：")

print("准确率：", accuracy\_score(y\_test, y\_pred\_rf))

print("分类报告：\n", classification\_report(y\_test, y\_pred\_rf))

print("混淆矩阵：\n", confusion\_matrix(y\_test, y\_pred\_rf))

# Adaboost

ab = AdaBoostClassifier(n\_estimators=100, random\_state=42)

ab.fit(X\_train, y\_train)

y\_pred\_ab = ab.predict(X\_test)

print("\nAdaboost结果：")

print("准确率：", accuracy\_score(y\_test, y\_pred\_ab))

print("分类报告：\n", classification\_report(y\_test, y\_pred\_ab))

print("混淆矩阵：\n", confusion\_matrix(y\_test, y\_pred\_ab))

import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

importances = rf.feature\_importances\_

feat\_names = X.columns

indices = importances.argsort()[::-1][:10]

plt.figure(figsize=(8,5))

plt.bar(range(len(indices)), importances[indices])

plt.xticks(range(len(indices)), [feat\_names[i] for i in indices], rotation=45)

plt.title("前10个特征的重要性（随机森林）")

plt.show()

import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

importances\_ab = ab.feature\_importances\_

indices\_ab = importances\_ab.argsort()[::-1][:10]

plt.figure(figsize=(8,5))

plt.bar(range(len(indices\_ab)), importances\_ab[indices\_ab])

plt.xticks(range(len(indices\_ab)), [feat\_names[i] for i in indices\_ab], rotation=45)

plt.title("前10个特征的重要性（Adaboost）")

plt.show()

plt.plot(fpr\_rf, tpr\_rf, label='随机森林 (AUC=%.2f)' % auc\_rf)

plt.plot(fpr\_ab, tpr\_ab, label='Adaboost (AUC=%.2f)' % auc\_ab)

plt.plot([0, 1], [0, 1], '--')

plt.legend()

plt.show()

import seaborn as sns

plt.figure(figsize=(4,4))

sns.heatmap(confusion\_matrix(y\_test, y\_pred\_rf), annot=True, fmt='d', cmap='Blues')

plt.title("随机森林混淆矩阵")

plt.xlabel("预测值")

plt.ylabel("真实值")

plt.show()

import seaborn as sns

plt.figure(figsize=(4,4))

sns.heatmap(confusion\_matrix(y\_test, y\_pred\_ab), annot=True, fmt='d', cmap='Oranges')

plt.title("Adaboost混淆矩阵")

plt.xlabel("预测值")

plt.ylabel("真实值")

plt.show()