# 引言

重症监护室 (Intensive Care Unit, ICU) 是现代医院中对抢救患有危重病情病人的重要单元，而ICU患者死亡率是评价ICU救治水平的一个重要评价指标。目前临床上已经有很多评分系统用于患者病情评估和生死预测。由于ICU患者生理数据具有高维型与复杂性，采用传统的评分系统处理ICU患者生死预测问题无法有效处理case-mix【2】问题，对不同疾病的预测效果也不稳定，不能满足ICU的临床要求。且传统评分方法常是侵入性的，在非住院情况下难以获得，容易受伪影影响，对患者伤害较大。如何对病人的海量生理数据进行分析，并从中提取隐含的有价值的、有意义的信息，同时降低预测成本，成为目前医学领域和数据挖掘领域的热点和难点【1,4】。

ICU患者的生死预测研究具有十分重要的意义。首先，准确地对ICU患者进行生死预测能够提高危重病人的生存几率；其次，ICU患者的早期生死预测作为衡量医疗水平的有用指标，能够帮助医生评估患者的病情，客观的制定和修正医疗护理计划，确定最佳的出院时间，减少对患者的过度治疗，从而降低患者的救治费用；再次，准确预测ICU患者的生死有利于评价医治措施和医生工作的效果，提高医疗质量；最后还对合理配置ICU医疗资源，营造良好的医患关系发挥着重要作用。

目前，已经有一些国家专门建立大型ICU数据库，用于收集来自各类ICU患者的信息，例如MIMIC【2，3】、ICNARC【2】、ANZIC【2】等。这些每年以成前上万规模增长的数据使得ICU死亡预测的问题的研究重点从数据收集变为数据分析。由于ICU患者先验治疗知识的不完备性和ICU时序检测数据的海量性，使得在ICU早期预警领域，数据挖掘方法受到越来越广泛的关注。仅几年来，许多学者尝试通过采用不同的机器学习方法，取得了接近或超越传统预测方法的效果【2】。2012年，麻省理工学院计算生理实验室举办了基于机器学习的ICU患者生死预测算法的国际竞赛【1,2】，他们提供了包括4种不同的ICU类型的4000例患者的ICU生理参数数据 (Physionet数据集)，并给出SAPS评分系统的预测结果作为对比。数据集包括6个描述性参数（性别，年龄等）和36个生理参数（体温，血压等），之后很多学者采用该数据集进行基于机器学习的ICU患者生死预测算法研究和测试。Hamilton等人【1】采用逐步回归法从Physionet数据集中筛选出40个变量作为逻辑回归的自变量，病人结果（生或死）作为因变量进行预测，结果表明逻辑回归的预测结果比判别分析，先验概率分析和柯尔莫哥洛夫-斯米尔诺夫 (KS) 检验分析方法更准确。Bera等人【1】根据患者生理参数，计算每个参数时间序列的最大值，最小值，平均值等衍生参数作为逻辑回归的模型的输入变量，预测准确率高于SAPS II评分系统的结果。同时作者提出性别，相对危险度等参数作为自变量可能会获得更好的预测结果。Lee等人【1】认为临床采用的评分系统严重依赖入院时对患者所做的诊断，并且不能处理缺失数据。他们采用6种方法补充缺失的生理参数，再使用逻辑回归模型进行预测。结果发现用与该病例相同性别和年龄层的病人的平均值作为缺失值的数据补充进行预测得到的结果更好。同时，在特征选择过程中，对于绝大多数病人都缺失的时间序列变量根据该参数是否被记录将其标记为一个二值属性。这样得到的预测结果和模型校准方面比SAPS I评分系统准确程度更高。由于隐马尔科夫模型能够更好的对时间序列的隐藏状态进行估计，Vairavan等人【1】提出了将马尔科夫模型与Logistics回归相结合的预测方法，与SAPS I评分系统结果相比该方法更为准确。

ICU预警模型除了对准确率上的要求，在提升模型鲁棒性，降低模型计算复杂度的问题上仍有大量尚未解决的难题。贝叶斯 (Bayes) 模型作为一种经典的分类方法，具有坚实的数学基础及稳定的分类效率，由于所需要的参数较少，且对缺失数据不太敏感，算法也相对简单，成为各个领域重要的分类工具。Martin等人【1】提出了利用线性贝叶斯分类器研究ICU患者生死预测问题。在特征选择阶段他们选取患者的高异常参数作为特征，并通过预处理和相关性分析将一些无法估计的参数（如缺失值）和相关系数高于0.6的参数去除，大大降低了特征维度，再用降维后的特征向量训练线性贝叶斯分类器。Johnson等人【1】提出了一种由500个弱学习器组成的贝叶斯集合方案对ICU患者生死进行预测。每一个弱分类器是一个二深度的决策树，能为随机选择的单个特征随机分配一个截距和斜率，同时学习器集合的参数由马尔可夫蒙特卡洛采样决定。

在机器学习算法中，人工神经网络由于具有类似非线性学习、多维映射和噪声容限等特点而被广泛应用。它为处理模糊、数据不完整、模拟、不精确的模式识别提供了全新的途径。Verplancke等人【4】通过把所有病人的选定生理指标放入一个“动态池”中并假设以最大似然为基础俩构建反射状态神经网络（ESN），从而取得更加稳定的预测结果。Xia等人【1】提出采用人工神经网络方法研究ICU患者的生死预测问题。通过计算参数的平均值、最小值、最大值、最终值和通过线性回归构建的这些序列的趋势参数等衍生参数共同构成了模型的特征向量。在神经网络训练过程中，为了克服神经网络容易出现的局部最小问题，实验采用了投票决策和模糊阈值的方法提高分类准确率。

支持向量机因为在分类与回归问题上的精确性和对样本维度的不敏感性得到越来越多的关注。Verplancke等人【4】在2008年首先验证了SVM处理具有恶性血液肿瘤疾病的ICU病人生死预测的可行性，发现SVM模型可以减少对病人生理数据的采样的同时取得类似的效果。Michael等人【9】利用GRID搜索算法和进化算法等多种算法对支持向量机模型参数进行优化，并将对percutaneous coronary intervention (PCI) 的病例的预测结果与逻辑回归模型结果进行比较，结果表明使用相同的评价指标进行优化，优化参数的选择对SVM核的性能都有显著影响。Citi等人【1】采用级联支持向量机-广义线性模型（SVM-GLM）范式研究ICU患者的生死预测问题。首先对缺失值插入零值，并对这些变量进行归一化。由于SVM在处理大型数据上具有良好的性能，所以在分类阶段采用了V-SVM方法进行分类，同时采用广义线性模型将分类结果做回归拟合预测患者的死亡率。实验结果证明SVM在预测患者死亡率问题上具有良好的表现。

对于其它机器学习方法，Marafino等人【8】开发了两种随机梯度下降的分类器，均使用了弹性网正则化的方法，并实现了在生物医学文本分类上的应用，其中表现最佳的分类器在对数损失函数和完全L2正则化下预测结果AUC为0.897，而完全L1正则化的AUC为0.889。虽然该文中使用的正则化方法能够增强模型的泛化能力，提高了可移植性，但正则化方法并不能完美解决从实验到应用的可移植性问题，可以考虑使用多模型的整合学习方法【8】。Delahanty等人【7】开发并评估了一种回顾性风险调整算法，并根据53所医院ICU病房患者数据进行学习与预测，其使用的XGBoost算法表现出了优异的辨别能力，AUC达到0.94，BER得分为52.8%，RIPD\_reduced模型AUC达0.91，BER得分0.49。Weissman等人【6】评估了逻辑回归，提升树模型，随机森林和弹性网络回归模型的性能，结果表明提升树模型具有最高的辨别能力，而逻辑回归的表现最差。但模型在患者年龄、ICU类型、机械通气方面对数据的要求很高。Awad等人【5】采用包括随机森林，决策树，朴素贝叶斯与基于规则的投影自适应共振理论模型的集成学习方法构建ICU患者早期死亡率预测框架，其预测结果要优于SOFA、SAPS I、APACHE II和MEWS等传统评分系统。但其使用的SMOTE算法虽然能够在整体上提升算法预测结果的AUC指标，但当原始数据类的规模差距较大时，容易造成在某一类上的过拟合情况，该文章中并未考虑到这一点。此外其只使用了患者入院6小时的数据，缺少在患者入院不同阶段的考虑。Cooper等人【18】使用了一种超学习（Super learner）的机器学习模型，该模型基本思想为：在一组候选算法的所的加权组合中选择最优回归方法。将超学习模型死亡率预测结果与标准逻辑回归，基于AIC的逻辑回归，广义可加模型，代惩罚最大似然的广义线性模型，多元自适应多项式样条回归，贝叶斯广义线性模型，广义回归模型，神经网络，决策树，随机森林，贝叶斯加性回归树等机器学习方法，SAPS II，SOFA等传统评分系统进行比较。结果显示超学习模型预测结果始终优于其它模型，特别地，当原始数据是时序数据时，超学习模型预测结果会有较为明显的提升，AUC达0.91。但在超学习方法中单个变量对结果的影响并不容易辨别，且模型中参数众多，计算复杂度高，往往需要数天的计算才能得到预测结果，这大大影响了在临床上的使用。类似的，Rose【19】也使用用Super learner 方法对老年人死亡率进行预测，结果显示Super learner方法相较于神经网络，随机森林，贝叶斯估计等多种方法预测结果均有明显提高。但Rose同样未能解决Super learner不适用于临床的问题。

Taylor等人【1】从患者的电子健康档案中提取查过500个临床数据变量构建随机森林模型，并采用该模型对患者病死率进行预测。然后将实验结果与分类与回归树模型，逻辑回归模型进行对比。通过对比证实，预测患者病死率的问题上，机器学习方法由于传统的分析技术。Churpek等人【1】采用利用一个大型的、多中心数据库的数据研究患者临床恶化预测的准确性，通过比较多种机器学习方法的实验结果，证明随机森林模型比逻辑回归模型的预测效果更加准确。Leite等人【4】借助生理指标特征对于集合的模糊隶属实现了类似人类思维的模糊分析，最终利用解模糊算子实现ICU病人的生死预测。Thongkam等人【4】提出了将温和Adaboost和随机森林算法结合的方法来研究乳腺癌症患者的存活几率。Dan Li等人【4】在对ICU数据形式化表示的基础上，提出了一种使用KNN算法来“投票”决定患者更倾向于“存活”还是“死亡”。Chih-wen等人【4】利用关联规则开发了针对ICU医疗干预的决策支持系统，该系统可以实时分析病人的状态并发现了需要重点关注的人群以及容易导致并发症的关键生理指标。

本文从MIMIC III数据库挖掘符合要求的患者特征信息，特征包括患者年龄、前24小时排尿量、BMI、昏迷指数、吸入氧浓度、心率、ICU类型、呼吸频率、体温、血氧饱和度、无创收缩压、无创舒张压与无创平均压，并对时序特征分别计算进入ICU前12小时与12~24小时的最大值、最小值、平均值、四分位数、方差、标准差与极差等衍生参数。在训练模型过程中使用欠采样，调节分类阈值等方法对数据进行预处理，并通过随机森林、XGBoost、lightGBM、Adaboost、逻辑回归、支持向量机、决策树、朴素贝叶斯等8种机器学习方法分别对特征降维前后的数据集进行学习、预测，使用组合特征选择的方法（a combinational feature selection method）确定最小和最佳特征子集。同时计算Spasiii、Sofa、Apsiii、Mews与Oasis等五种传统评分系统对患者生死预测的结果作为机器学习预测结果的对照。从预测结果可以看到，机器学习的预测结果能够超越传统评分系统的预测结果。计算传统评分往往需要患者大量的生理指标，或需要具有专业医学知识背景的医护人员对患者情况作出评价，需耗费大量人力和财力【25】，且传统评分方法常是侵入性的，不仅在非住院情况下难以获得，且容易受伪影影响，对患者伤害较大。相比之下，机器学习有着快捷、高效、低成本的优势，且需要的特征易获取，更稳定，本文中的无创参数相比于有创参数更是大大降低了临床使用要求，减少对患者的伤害。这种新颖的机器学习方法在缺乏侵入性生理参数和生化指标的情况下能够准确的进行ICU患者生死预测，有望在更广泛的场景中进行简单快速的临床应用。

# 2模型

## 2.1 数据提取与数据补齐

根据患者特征，我们对患者提出以下选择标准：

1. 仅进入过一次ICU；
2. 仅进入过一次医院；
3. 患者年龄超过16岁且无30天内非院内死亡的病例。

本文主要提取了患者的非侵入性生理参数：人口统计学信息（年龄，性别，BMI）; ICU信息（ICU类型，ICU住院时间，是否死亡）;临床信息（SpO2，体温，心率，无创血压，有创血压，GCS，排尿量）;呼吸系统信息（呼吸频率，是否接入呼吸机）。

### 2.1.1 数据异常值处理

选取正常值如下HR：0~300、Resprate：0~70、TempF：70~120、TempC：10~50、Spo2:0~100、Fio2:0~100、Nisysbp:0~400、Nidiasbp:0~300、Nimeanbp:0~300、GCS:0~15。

异常的数据将被置空。

### 2.1.2 无创血压插补

测量血压( Blood Pressure, BP)是临床和家庭保健中监测生理状态最普遍的方式。根据血压测量的方式不同,可以把血压分成直接血压测量( Invasive blood pressure,IBP)和间接血压测量( Non-invasive Blood Pressure. NIBP)。

直接血压方法测量的方式是将一导管插入到患者需要检测部位(一般是左上臂)的动脉血管中,再通过与之连接的传感器获得血压数据。这种方法能够非常确切的反应当下患者的血压情况,因为这种测量方法是实时检测变化血压并且测量结果十分精确,直接测量方法就常被当作其他测量方法的“金标准”,即作为诊断疾病的最可靠、最准确的方法,在医疗领域有着及其重要的应用:

(1)用在病情危险的患者、循环作用不完整、大血管相关的手术等可能导致产生大失血的患者中；

(2)用在紧急低血压、昏迷和其它供血不稳患者的血压测量中；

(3)用在重大高血压患者的血压测量中；

(4)必须进行重复抽取血液,以完成血液成分分析时。

直接测量血压方法可以比较准确、实时的监测用户血压,在特殊医疗领域中具有十分重要地位。但是,有创法也有很大的局限性:该方法是侵入性的方法,只有在病房的重症患者以及较大手术中的患者会被采用该法测量血压,而无特殊状况外,一般不采用此方法,而是采用更为简便易操作的间接测量方法，通过间接测量方法获得的数据即为无创数据。

在现实医疗救治过程中，虽然有创的测量方式能够更加精准的测量出患者的血压数值，但相比无创血压数据却不易获得，且对患者有着一定的伤害。本文中采用易获得，对患者危害较小的无创血压参数，而无创血压数据量较少，而有创血压因为持续测量产生了很多数据，因此使用数据补齐的方法对无创数据进行补充。

所采用的数据补齐方法有：

1). 平均值

计算需要进行插补的特征的有效值平均值，并使用平均值对缺失数据进行插补。

2). 中位数

计算需要进行插补的特征的有效值中位数，并使用中位数对缺失数据进行插补。

3). 众数

计算需要进行插补的特征的有效值众数，并使用众数对缺失数据进行插补。

4). K近邻方法

K近邻 (K-Nearest Neighbor, KNN) 算法是机器学习中最古老也是最简单的算法之一，既可用于分类也可应用于回归问题。其基本思想是，对于一个给定的测试样例，寻找其在训练集中最近的K近邻点，并用它们来预测测试点的输出。虽然KNN算法的思想很简单，但是在许多实际应用中表现出很好的性能，如文本分类【12】、蛋白质定位【12】、无线室内定位【12】等，该算法基本思想如下：

对于给定的一组样本，传统的参数回归模型是通过拟合寻找参数的具体表达式：



其中是参数，假定各种变量的关系已知，拟合过程就是优化参数使得样本参数最小化。而非参数回归模型的表达形式为：



非参数回归模型假定各种变量关系未知，要对整个回归函数进行估计，这是一种基于范例推理的启发式建模技术，不对建模数据添加任何限制性假设，依靠数据潜在的关系对新问题做出估计，而无需具有关于模拟过程的先验知识，也不需要建立具体的模型参数，它的模型隐含在系统输入和输出状态数据中。因而，非参数回归模型是较线性和非线性回归模型更符合实际需要，更能反映变量之间的实际关系，具有很好的拟合效果和推广能力。近邻非参数回归是构造一系列最优近邻子集，在该子集上估计回归函数，对于给定的样本离其最近的个观测值组成近邻子集，由下式确定：



近邻中心为：



回归函数由下式进行估计：



其中为核函数。

5). 随机森林方法

随机森林是机器学习中一种集成式的分类、回归学习算法，最早是由Leo Breiman【13,14】通过将Bagging理论与随机子空间方法项结合提出的。随机建立多棵相互独立的决策树组成森林，可以防止过拟合现象。决策树之间各自生长、互不影响，因此可以并行的构建多棵决策树。与其它分类算法相比，随机森林的优势在于：

(1)、随机森林可以处理具有大量属性特征的数据，即适合在高纬度数据下运行，训练结果不易出现过拟合现象；

(2)、对数据的形式要求不高，且无需进行数据规范化，处理对象比较灵活；

(3)、能够判断出属性的重要性，并对个属性重要性进行排序；

(4)、预测速度相对较快。

随机森林回归基本思想如下：

随机森林回归算法是由一组回归决策子树构成的组合模型。其中是服从独立同分布的随机变量，表示自变量，表示决策树的个数。利用集成学习的思想取各决策子树的均值作为回归预测结果：



其中，为基于和的输出。

为了克服决策树模型精度不高、易出现过拟合的问题，随机森林引入了bagging【14】和随机子空间【14】的思想。

(1)、bagging思想

从原始样本中有放回地随机抽取多个训练样本，且每个训练样本分贝构建回归决策子树，最后取各棵的平均值作为最终预测结果。

假设为原始样本，为中的样本数，则中的每个样本没有被抽取的概率为，当时：



即，每次约有36.8%的样本未被抽取，称为OOB (out-of-bag)。Bagging思想不仅可以随机化建立更多的回归决策子树，同时还保证了子树与子树之间的独立性。

(2)、随机子空间思想

在构建回归决策子树的过程中，每个分裂节点都从总的特征空间中随机抽取特征子空间作为节点的候选特征集，并从中选取最优特征进行分裂。该方法既保证了树与树之间的节点以及每棵树节点之间特征子集都不同，又保证了树的多样性，进而提高了随机森林节点分裂的随机性。

在随机森林中，回归决策子树和特征子空间中的特征个数决定着模型最终的预测性能。

(3)、泛化误差

泛化误差反映了模型对训练集以外的数据的预测能力，是判断模型好坏的重要指标。

假设从独立同分布的随机向量中抽取训练集，形成的训练集之间各自独立。则输出的均方泛化误差收敛于：



其中，为第棵回归决策子树的随机变量，为数学期望。上式中即为随机森林的泛化误差。随机森林随着回归决策子树的增加会逐步收敛，不会出现过拟合的问题，但最终会趋于一个稳定值。

将每一颗回归决策树的平均泛化误差定义为：



假设对于所有的随机变量，回归决策子树都是无偏的，即，则有：



其中，为残差和的相关系数，和相互独立。

构建随机森林回归算法流程如下：

步骤1、bagging思想，随机产生样本子集；

步骤2、利用随机子空间思想，随机抽取个特征，进行节点分裂，构建单棵回归决策树；

步骤3、重复步骤1、步骤2，构建棵回归决策子树，每棵树自由生长，不进行剪枝，形成森林；

步骤4、计算OOB误差，调整参数优化模型，重复步骤1、步骤2、步骤3，直至OOB误差不再减小；

步骤5、棵决策子树的预测值取平均，作为最终结果。

### 2.1.3 衍生变量的计算与独热编码

在获取了患者数据补齐后的无创血压数据之后，为充分挖掘患者信息，对具有时序性的特征计算衍生变量。

计算衍生变量的特征有：吸入氧浓度(spo2)，心率(hr)，呼吸频率(resp)，体温(temp)，血氧饱和度(spo2) ，无创收缩压(nisysbp)、舒张压(nidiasbp)、平均压(nimeanbp)。

对上述特征首先筛选出患者进ICU前12小时数据，12~24小时数据，对这两段数据分别计算数据的最大值、最小值、平均值、方差、标准差、极差和四分位数等衍生变量。

对性别(gender)，是否接通呼吸机(vent)，是否死亡(death)，ICU类型(icutype)等非数值特征则进行独热编码。

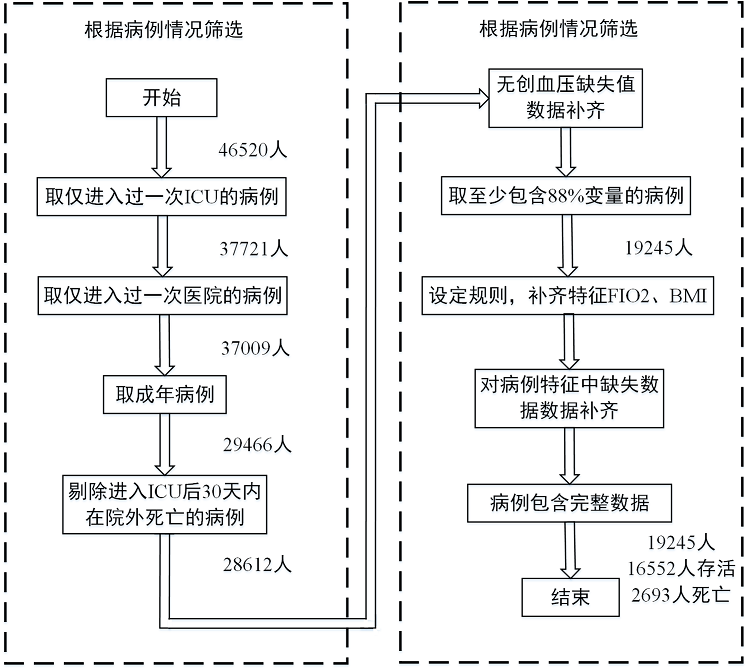
### 2.1.4 患者特征数值插补

因为患者数据记录完整程度参差不齐，对缺失数据过多的患者数据进行插补会增加模型的不稳定性。为此，在对患者特征数值进行插补之前首先根据患者数据缺失情况，剔除数据缺失过多的患者。

在对患者进行筛选之后，仍然会有某些特征缺失比例较大。对于这些特征，首先需要考虑根据患者入院实际情况人为提供数据进行插补。不能插补的考虑删除该特征。

对其它特征缺失数据，仍使用上述提及的插补模型进行插补。

### 2.1.5 数据提取与补齐流程



图x. 数据提取与数据预处理

1).根据患者信息逐次筛选出仅进入一次ICU的病例，仅进入一次医院的病例，成年病例，剔除进入ICU后30内在院外死亡的病例。最后提取出满足上述要求的病例28612人；

2).从原始数据集构建一个完整的数据集，根据五种插补方法对随机缺失的无创血压数值进行插补，根据插补结果评价指标与模型计算成本，最后选择随机森林插补方法对原始数据中缺失的无创血压数据进行插补；

3).根据患者信息完整程度，选择每增加1%人数减少最大的点，选取至少包括88%个有效特征的患者信息，筛选出19245条患者信息；

4).对bmi特征及fio2衍生变量根据实际情况进行人为插补，特别地，根据bmi特征生成非数值意义的变量bmi\_label，来表示bmi数据是都为插补数据，并进行独热编码；

5).由于缺失特征数量较大，在选择插补模型过程中，只对qfio2mean， hfio2mean， qtempmean， htempmean四个特征进行插补测试。从原始数据集构建一个完整的数据集，根据五种插补方法对随机缺失的特征数值进行插补，根据插补结果评价指标与模型计算成本，最后选择随机森林插补方法对原始数据中缺失的特征数值进行插补；

## 2.2 特征选择

考虑把维空间内的样本投影到条直线上，形成一维空间，通过寻找一个最佳的投影方向，使得不同类内的维样本在该方向上的投影可以被较好的区分开来，这就是特征选择的本质。

本研究提取了患者的各种信息和生理参数，从直观上不能区分哪些特征与ICU早期预警的相关性。 另一方面，监督学习算法的性能与输入的特征集合和结果变量之间的相关性有很大关系。 特征选择的目的是找到与原始特征集相比优化算法性能的特征子集。 通常有三种类型的特征选择算法：Filter，Wrapper，Embedded。Filter方法首先选择数据集的特征，然后训练分类器。 特征选择过程独立于后续分类器。 与Filter方法不同，Wrapper特征选择直接将分类器的性能用作特征子集的评估标准，对于大多数情况，该方法的计算成本通常比Filter大得多。 Embedded方法将评估特征重要性与模型算法相结合，从而导致特征值选择结果与评估算法之间的高度相关性。

本文中使用Relieff、Fisher、Gini指数三种方法挖掘特征权重信息。Relieff算法根据各个特征和类别的相关性赋予特征不同的权重，属于Filter方法；Fisher算法寻找最优的转化矩阵使类内间距尽可能小，类间间距尽可能大，运用距离度量计算特征权重，属于Filter方法；Gini指数计算样本集合中一个随机选中的样本被分错的概率（两个划分的一致性），根据信息混乱度进行特征排序，属于Embebded方法。三种算法分别从不同的角度对特征重要性进行评价，有效避免了因单一算法的模型缺陷造成所提取信息的结构缺失，充分挖掘样本特征信息。

本文中特征最终的权值由三种算法归一化后的结果线性组合构成：

FSscore = Relieff\_score + fisher\_score – Gini\_score

### 2.2.1 显著性检验

显著性检验是对整体所作的假设性检验，其原理就是“小概率事件实际不可能性原理”来接受或否定假设。所谓“显著”，就是指两种或多种处理试验结果之间，本身确实存在差异。如果是“不显著”，就说明它们之间的差异是由抽样或偶然的因素引起的，不是真正有实际差异存在。

当比较两种或多种处理的试验结果的平均数时，通常先假定它们是从同一总体内抽取的多个样本，它们之间没什么差异(即平均数之差等于零)。如果检验后所得的差数是由于抽样误差所引起的概率少于或等于5%时，则称这差数与假设不符合，即它们之间存在的差异是显著的。

### 2.2.2 reliefF

最早由Kira提出的 Relief算法局限于两类问题，随后Kononenko提出 Relieff算法，解决了多类问题和回归问题，以及数据缺失问题。 Relieff算法的基本思想是给特征集中每个特征赋予权值，迭代更新权值，然后根据权值大小选择特征子集，使得好的特征聚集同类样本，而离散异类样本。

首先从数据集中随机挑选个样本实例，对于每一个样本实例都有两个最近邻样本：与样本同类的最近邻样本NearHit，简称为，与样本不同类的最近邻样本NearMiss，简记为。分别从两个最近邻样本中选择个最近邻样本实例，然后计算样本与最近邻样本关于特征之间的距离和样本与之间的距离。如果，说明特征对于分类有利，权重增加；如果，则权重减小。最后将以上过程重复次，得到每一个特征次迭代结果的平均值。权重值越大，该特征对于分类越有利；反之，不利于分类。特征权重计算公式如下：



其中，代表特征序号，代表特征数目，代表当前特征的特征计算数值，特别地的初始值为0；代表选择的特征样本的数目；代表与样本不同类别的中的第个近邻样本；代表当前特征样本在当前大分类分布概率；代表当前选择的特征样本，表示所属类别；为计算样本实例和关于某个特征之间的距离大小，定义如下：



### 2.2.3 Fishier

Fisher判别分析（FDA）【16】的基本思想是将原来高维的模式样本投影到最佳鉴别向量空间，在此方向上的类内散布接近于正交，投影后的样本在新的特征子空间上有最大的类间距和最小的类内间距，使得该模式在该子空间中有最佳的可分离性。利用下式的思想，FDA可变成求矩阵最优化的形式：







其中表示满足时所有数字的总和，表示在类中样本的平均值。降维的目标：寻找一个最优化的转化矩阵，尽可能使样本数据潜在空间的类间距离最大，同时类内的距离最小，当考虑到总矩阵时，可得到公式。结合上式，PDA中的可分别用数据对权值思想表示成：





其中，若，则；否则，，代表标签数据在类中的个数，在这种情况下，。FDA能够利用标号数据的信息，通过监督思想较好的发现样本数的潜在的低维表示。

### 2.2.4 Gini指数

基尼指数是一种非纯度的属性分裂方法，适用于类别、二进制、连续数值等类型的字段，是Breiman等人【10】于1984年提出的，被广泛应用于CART算法、SLIQ算法、SPRINT算法和Intelligent Miner的决策树算法中，算法思想描述如下：

假设是个数据样本的集合，其类标号属性具有个不同值，定义个不同类根据类标号属性值的不同可将划分为个子集，设是属于类的样本集，是集合中的样本数，则集合的Gini指标为:



其中是任意样本属于的概率，用估计。当最小为0时，在此集合处所有记录都属于同一类别，表示能得到最大的有用信息;当此集合中的所有样本对于类别字段来讲均匀分布时， 最大，表示能得到最小的有用信息。

## 2.3 最小特征子集确定方法

(我们的目的是获取最为优秀的一个关系，这个关系是 BER与特征量的映射关系。传统方法的错误在于忽视了这种映射，而是单纯的根据BER数据集合进行考虑，与最小特征量的选取过程是分离的，这就导致了传统方法选择出的最小特征量在很大程度上对应的性价比不是可能的特征子集中最优秀的)

鉴于在ICU早期预警装置中使用的微处理器的计算能力的限制，识别最小特征子集在使用算法进行预测的过程中是非常重要的。为此，我们提出一种新的最小特征子集选择方法，用以和传统方法效果进行比较。

1. 传统方法为，将特征排序后的特征全集，逐次增加的使用机器学习模型进行十折交叉验证，记录每次预测结果的平均值与标准差。在最小平均值的基础上加以对应的标准差，随后寻找与该数值最接近的平均值，则其对应的特征量即为最小特征子集的特征量。

2．我们提出的方法具体步骤如下：

1) 将特征排序后的特征全集，逐次增加的使用机器学习模型进行十折交叉验证，记录每次预测结果的平均值;

2) 首先在集合中，以为因变量，对应的特征量为自变量。在这个完整的映射中，寻找一个最大的严格递减的子映射。以为因变量，特征子集个数为自变量进行一阶求导。寻找这样一个关键点，该点后的一阶右导数值均小于该点处的右导数值，且该点后的右导数值均小于一个阈值;

3) 寻找到点后对该点后的严格下降数据的前进行二次求导寻找其中二阶导数最接近0的点;

4) 选取以为终点，长度为  的原始平均值的数据段，计算这段数据中最小与次小之间的一阶导数绝对值，并记录对应的特征子集数量从小到大将子集记录为。如果对应的小于则认为为最小特征子集。如果对应的大于，且导数值小于，认为为最小特征子集，反之导数值大于，认为为最小特征子集。

在本文中，被设置为0.005；被设置为0.001（细选，为的1/5到1/10）；被设置为关键点与关键点之间的距离；被设置为4。的取值越小（最小为1），则越可能使得计算得到的最小特征子集数量偏大。其中第二步与第三步是在大范围中进行筛选，是粗调，第四步为在小范围中确定最小子集，是细调。该方法适用于特征数量较多的情况。

## 2.4 传统评分系统与机器学习最优分类阈值选择模型【27】

这里使用Cost-Sensitive方法实现传统评分系统与机器学习的阈值选择。

### 2.4.1 Cost-Sensitive中Cost的定义与表示

Cost- sensitive分类实际研究的是在具有不同cost值的数据集上的分类问题。由于这类问题需要在既有的数据集上添加了cost信息，因此我们要给出cost的定义和具体描述方法。通常情况下，我们使用 cost matrix描述待分类数据集上的cost信息。 Cost matrix 中，是一个的矩阵，其中代表待分类数据集中的类别的数量。中的条目表示分类算法将实际属于类得到实例分为类所造成的cost。易见当时代表分类算法正确预測了实例的类别，而的条目对应于不正确的分类结果。

利用 cost matrix，我们就可以通过为其中的每一项赋值来定义待分类数据集上的cost。一般情况下，需要进行cost- sensitive分类的数据集会提供一个cost matrix供分类算法使用。由于现实世界中的cost信息很难准确获取，同时又不可能保证其在不同实例间的平均分布，因此 cost matrix往往是由数据集提供者通过自己的经验给出的值。在对进行赋值的过程中，一般会遵循合理性原则.该原则是在建立下述判断的基础上，即错误分类的造成的cost肯定要大于正确分类造成的cost，因此我们要求对的赋值应该满足。其中

|  |  |  |
| --- | --- | --- |
|  | **actual negative** | **actual positive** |
| **predict negative** | **c00** | **c01** |
| **predict positive** | **c10** | **c11** |

Cost的实际意义是分类算法的分类结果相对于真实分类给被分类数据造成的影响。可以看出，这种影响可以是负面的损失，也可以使正面的收益。在考虑损失的情况下，我们认为只有错误的分类结果会造成损失也就是说在 cost matrix中的， when 对于其他的值，即错误的分类结果对应的条目，我们赋其值为正数，用于表示其代价。同时我们应该保证，即做出FN判断的代价要高于做出FP的代价。在考虑收益的情况下，正确的分类结果会给使用者带来一定的收益。所以我们一般将， when 的值赋为正值，而将其他的条目，即错误的判断的cost条目赋值为负值。

### 2.4.2 Cost-Sensitive分类的目标

利用 cost matrix，我们可以将cost- sensitive分类问题转化一个优化问题，即对于某一分类算法，为实例寻找到一个类别，使得下式中的达到最小值：



其中是一个实例， 表示将其分为类， 表示在当前算法中获得的属于类别的概率。

对于每个类别，都是所有可能的类别的cost值的概率和。这样的目标使得我们不再关注如何获得的最大值，而是要关注何种分类能够最小化分类结果对应的cost值。这就使得cost- sensitive分类算法在追寻cost最优的前提下，可能会放弃可能性最大的分类结果。这种判断虽然有一定的反直觉成分，但是在实际应用中却有比较重要的现实意义。

在二分类的情况下，从表达式中可以知道，cost-sensltive分类算法将实例分为1类的条件是当且仅当将分为1类所造成的预期cost要不大于将分为0类的损失。即



令可以得到



即我们在二分类中所需要考虑具体问题。当式子取等号时，也就是说分类算法使得Class1和Class0的分类结果达到平衡时，即为我们需要的最优分类决定。

## 2.5 机器学习模型

### 2.5.1 XGBOOST

对于一个复杂的分类任务，给定一组训练样本集后，求比较粗糙的分类规则（弱分类器）要比求精确的分类规则（强分类器）容易的多，提升算法就是从弱分类器学习算法出发，反复学习，得到一系列弱分类器（称为基本分类器），然后组合这些弱分类器，构成强分类器。大多数的提升方法都是改变悬链数据的概率分布，针对不同的训练数据分布调用弱学习算法学习一系列弱分类器。因此，提升算法实际上就是采用加法模型（即基函数的线性组合）与前向分布算法将多个弱分类器按不同的比例组合成一个强分类器【15】。

当提升算法在学习弱分类器时，如果弱分类器采用决策树做基函数，由此得到的强分类器的算法称为提升树算法，提升树模型可以表示为决策树的加法模型:



其中表示决策树，表示决策树的参数；为树的个数。

提升树算法采用前向分布算法。首先确定初始提升树，第步的模型是：



其中是当前模型，通过经验风险极小化确定下一棵决策树的参数：



其中，表示损失函数。由于提升树算法利用加法模型与前向分布算法实现学习的优化过程，当损失函数是平方损失和指数损失时，每一步优化是很简单的。但对于其它损失函数，往往每一步的优化都并不容易。针对这一问题，Frediman【15】提出了梯度提升 (Gradient Boosting) 算法。这是利用梯度下降法的近似方法，其关键是利用损失函数的负梯度在当前模型的值，即：



Xgboost算法对GBDT算法的改进在于，一是求解优化函数过程中，GBDT采用一阶导数，而 Xgboost算法计算目标函数的二阶导数；二是 Xgboost算法在目标函数中加入了正则项，有利于防止算法在训练过程中产生过拟合现象，导致训练的得到模型泛化能力变差；三是 Xgboost算法在训练过程中借鉴了随机森林的思想，毎次迭代过程中不使用全部样本，并且不使用每个样本的所有特征，而是对样本进行抽样，采用部分样本的部分特征去训练，有效提高了模型的泛化能力:四是 XGBOOSL算法支持并行计算，提高了运行速度。首先定义目标函数：



其中是损失函数，常见的损失函数包括平方损失、Hinge损失、交叉信息熵损失等。是正则项，常见的正则项包括L1和L2，正则项的作用是防止模型在训练时发生过拟合现象，导致模型的泛化能力变差。

由于 XGBOOST算法采用决策树作为基分类器，因此正则项表示为：



其中，表示为当前决策树的叶子节点数，是每个叶子节点的权重。由于模型预测的结果可以表示为所有决策树预测结果的加权和：



其中，表示决策树空间，即所有决策树的集合。因此，目标函数可表示为：



对于第轮迭代，目标函数表示为：



即对的预测值是是第棵树的预测值和前棵树预测值的和。对上式求一阶导数和二阶导数并移除常数项可得：







其中是一阶导数值，是二阶导数值，是决策树上子节点的权重，是决策树的叶子数目，是惩罚值。

### 2.5.2 lightGBM

Lightgbm (Light Gradient Boosting Machine) 是2016年微软亚洲研究院公布的一个开源决速、高效的基于决策树算法的提升 (GBDT、GBRT、GBM和MART) 框架，被用于排序分类、回归等多种机器学习的任务，支持高效率的并行训练。

#### 2.5.2.1 Gradient Boosting

是用一系列子模型的线性组合来完成学习任务的,它分为两种类型：Adaboost和 Gradient boosting，LightGBM属于Gradient boosting的一种。

Gradient Boosting的思想是：一次性迭代变量，迭代过程中，逐一增加子模型，并且保证损失函数的值不断减小。

Gradient Boosting的思想是：在迭代过程中，一次性迭代变量，并逐一增加子模型，保证损失函数的数值不断减小。

假设为子模型，复合模型为：



损失函数为，每加入一次新的子模型后，使得损失函数不断朝着信息含量次高的变量的梯度减小：



#### 2.5.2.2 决策树

决策树 (Decision Tree) 是一种分类和回归的方法，实际研究中大多用于分类。决策树的结构呈树形结构，大多运用的是二杈树，在每一个叶子节点上，根据某一判断条件，输出“符合条件”和“不符合条件”两类，不断重复的输出。可以把决策树理解成众多规则的集合，也可以认为是定义在特定空间与类空间上的条件概率分布。决策树的创建包括3个主要步骤:特征选择、决策树的生成和决策树的修剪，该方法具有可读性高、分类速度快的优点。

决策树的分裂方法分为两类，一类是按叶子分裂的学习方法 (Leaf-wise Learning)；另类是按层分裂的学习方法 (Level-wise Learning)。

按叶子分裂的学习方法是指在分裂的过程中要不断地寻找分裂后收益最大的节点，对其进行进一步的分裂，其他非收益最大化的结点不再继续分裂，以这样的规则生长这棵树。这样做的优点是可以使算法更加快速有效;缺点是会忽略掉那些被舍弃的叶子上的信息，导致分裂结果不够细化。

按层分裂的学习方法与按叶子分裂的学习方法不同，它不需要挑选收益最大化的节点，每一层的每一个结点都要进行分裂，也就是说每次送代都要遍历整个训练数据的所有数据。优点是每一层的叶子可以并行完成，具有天然的并行性;缺点是这样会产生很多没有必要的分裂，需要更多的计算成本，同时，也会占用较大的运行内存。

#### 2.5.2.3 GBDT

GBDT (Gradient Boosting Decision Tree) 是机器学习中的一个长盛不衰的模型，事实上：



即若Gradient Boosting中的每一个子模型都是Decision Tree，这样的模型就是GBDT。

GBDT拥有着 Gradient Boosting和 Decision Tree的功能共同特性，具有训练效果好、不易过拟合等优点。GBDT的工具主要包括 Xgboost、Pgbrt、Sklearn、RGBM等。GBDT在工业界应用广泛，通常被用于点击率预测，搜索排序等任务。GBDT也是各种数据挖掘竟赛的致命武器，据统计 Kaggle上的比赛有一半以上的冠军方案都是基于GBDT

LightGBM是GBDT的一种，被提出的主要原因是为了解决GBDT在海量数据遇到的问题，让GBDT可以更好更快地用于实践。

LightGBM中的决策树子模型是采用按叶子分裂的方法分裂节点的，因此它的计算代价比较小，也正是因为选择了这种分裂方式，需要控制树的深度和每个叶子节点的最小数据量，从而避免过拟合现象的发生。 Lightgbm选择了基于 Histogram的决策树算法，将特征值分为很多个小“桶”，进而在这些“桶”上寻找分裂，这样可以减小储存成本和计算成本。另外，类别特征的处理，也使得 LightGBM在特定数据下有比较好的提升。

Lightgbm分为三类:特征并行、数据并行和投票并行。特征并行运用在特征较多的场景，数据并行应用在数据量较大的场景，投票并行应用在特征和投票都比较多的场景。

### 2.5.3 随机森林

随机森林是一个由一组决策树分类器组成的集成分类器，其中是服从独立同分布的随机向量，表示随机森林中决策树的个数，在给自定义下，每个决策树分类器通过投票来决定最优的分类结果。

随机森林是许多决策树集成在一起的分类器，生成随机森林的步骤如下：

1. 从原始训练数据集中，应用boostrap方法有放回的随机抽取个新的自助样本集，并由此构建棵分类回归树(CART)，每次未被抽到的样本组成了个袋外数据(out of bag，OOB)
2. 设有个特征，则在每一棵树的每个节点处随机抽取m个特征(m<=n)，通过计算每个特征蕴含的信息量，在m个特征中选择一个最具有分类能力的特征进行节点分裂。
3. 每棵树最大限度生长，不做任何裁剪。
4. 将生成的多棵树组成随机森林，用随机森林对新的数据进行分类，分类结果按树分类器的投票多少而定。

给定一组分类器，每个分类器的训练集都是从原始的服从随机分布的数据集中随机抽样所得，余量函数(margin function)定义为：



其中为示性函数。

余量函数用于度量平均正确分类数超过平均错误分类数的程度，余量值越大，分类预测越可靠。

泛化误差定义为



其中下标概率覆盖空间。

在随机森林中，当决策树分类器足够多，服从强大数定律。

随着随机森林中决策树数量的增加，对于所有序列，几乎处处收敛于



上式表明随机森林不会随着决策树的增加产生过拟合问题，但可能会产生一定限度内的泛化误差。

### 2.5.4 AdaBosot

AdaBoost算法是一种分类器算法，由Yoav Freund和Robert E.Schapire在1995年提出。其基本思想是利用分类能力简单的分类器( weaker classifier)通过一定方法叠加( boost)起来，构成具有更强分类能力的强分类器。理论证明，只要每个简单分类器分类能力比随机猜测要好，当简单分类器个数趋向于无穷时，强分类器的错误率将趋于零。算法描述如下：

已知有个训练样本的训练集，其中对应样本的假和真。在训练样本中共有个假样本，个真样本，待分类物体有个简单特征，表示为，其中。对于第个样本，它的特征为，对每一个输入特征有一个简单的二值分类器。第个特征的弱分类器由一个阈值，一个特征和一个指示不等式方向的偏置（只有两种情况）构成



训练目标是通过对正负样本进行分析，选择分类错误率最低的个弱分类器，最终优化组合成一个强分类器。

训练过程如下：

(1)、设为第次循环中第个样本的误差权重，对训练样本重的误差权重按照如下公式初始化：对于的样本，；对于的样本，。

(2)、

(I) 权重归一化，使得为；

(II) 对于每个特征，训练出来其弱分类器，也就是确定阈值和偏置，使其目标函数达到最小；

(III) 从(II)确定的弱分类器中找出一个具有最小错误的弱分类器；

(IV) 更新每个样本所对应的权重。确定的方法为：若第i个样本被正确分类，则；反之，。

(3)、形成的强分类器为



其中：。

以上训练过程的意义可以表述为，当已经提取的分类器对于某些样本分类正确，那么减少这些样本的权重。当分类错误，增加这些样本的权重。这样，后面的训练提取的若干分类器就会更加强化对这些分类错误样本的训练。

### 2.5.5 支持向量机

统计学习理论是在传统统计学基础上发展起来的一种机器学习方法。在二类线性可分的情况下，有很多可能的线性分类器可以把这组数据分割开，但是只有一个使两类的分割间隔margin最大，这个线性分类器就是最优分类超平面，与其它分类器相比，具有更好的泛化性。

若样本集是线性可分的。则存在分类超平面，对样本集中任一都满足：



在空间中样本到超平面的距离，其中。当存在使得，则超平面分类间隔。使分类间隔margin最大的超平面即为最优分类超平面。寻找最优分类超平面的问题将转化为求如下一个二次规划问题：



满足约束条件

采用Lagrange乘子转换为一个对偶问题，形式如下;



满足约束条件



其中为每一个样本对应的Lagrange乘子，根据Kuhn-Tucker条件，这个优化必须满足：



因此多数样本对应将为0，少部分不为0的对应的样本就是支持向量。最后得到分类判别函数为：



其中是分类的阈值，可以通过两类中任意一对支持向量取中值求得。

由上述易知，对于空间中任意样本，当时，表示此时在超平面的分类间隔内，越趋于0，则当前分类超平面对于的区分能力越差。而时能被超平面正确分类。

### 2.5.6 逻辑回归

Logisitic regression是比较常用的机器学习方法，用于估计某种事物的可能性。逻辑回归延伸了多元线性回归思想，即因变量是二值的情况，自变量为。逻辑回归是用来测量分类结果与因变量之间的关系。逻辑回归模型的最终结果为0,1分类结果。其中1表示属于该类，0表示不属于该类。逻辑回归方程表示如下：



其中表示个特征属性值，表示特征属性参数值。逻辑回归方程表示某一样本属于该分类的概率值。

构建逻辑回归模型的主要目的是求解逻辑回归方程中的参数。下面概要介绍求解逻辑回归方程参数的方法。

首先，知道一个样本属于正类的概率值为，那么，该样本属于负类的概率值为。所以，在样本和概率已知的条件下，该样本属于正类的概率可以表示为:



从上面的表达式可以推断出在参数逻辑回归模型已知的条件下，训练集发生的可能性以及对数可能性可以表示为：



其中，表示训练集的第个样本，表示第个样本的分类结果，表示样本的数目。当训练样本以及样本的分类结果已知的条件下，可以通过计算最大对数可能性求解逻辑回归模型参数。

### 2.5.7 朴素贝叶斯

朴素贝叶斯算法是基于一个简单假设所建立的一种贝叶斯 (Bayesian) 方法，朴素贝叶斯假定样本的不同特征属性对样本的归类是相互独立的。朴素贝叶斯分类器 (Naïve Bayesian Classifier, NBC) 是以贝叶斯定理为基础的一种应用，目前被广泛使用在信息领域，如垃圾电子邮件的过滤、模式识别、入侵检测等【11】，其算法思想描述如下：

设置是具有优先数量实例的训练集，是个有限的属性。一个实例用向量描述，其中是属性的当前取值。类别属性用表示。函数取得属性的定义域集合。则当



成立时，即预测实例的类别是在属性给定的条件下后验概率最大的类别时，预测的正确率最大。但是，后验概率难以计算，因此朴素贝叶斯引入了下列假设：

在给定类别的条件下，所有的属性相互独立：



上式即为朴素贝叶斯假设。

在朴素贝叶斯算法中，可以独立的学习每个属性在类别标签下的条件概率。然后，分类器应用贝叶斯公式计算特定数据在给定属性值下类别的后验概率：



并预测该数据属于后验概率最大的类别。

### 2.5.8 决策树

#### 2.5.8.1 决策树算法概述

决策树分类算法是一种机器学习算法，是从训练样本中自动挖掘出一套对训练样本以外的数据同样有效的规则模型的过程。决策树分类算法所构建的决策树是一种分类模型，模型中每一个分支都代表着对象的一个属性与属性的某一取值或取值类型的映射关系。在决策树中，每一个非叶子节点表示一个判断条件，每个判断条件对应一个对象属性，而每个分支路径代表满足判断条件的属性值。树中每个叶节点表示一个取值集合，集合中的每一个取值都满足从根节点到叶节点所经路径中的每一个判断条件。决策树从根节点开始构造，首先选择合适的属性将样本集合划分为若干子集合，每个子集合组成一个分直节点，然后再对每一个分支节点进行划分，直到节点中所有样本的类型一致，或者满足某种结束条件。

决策树构造一般包括以下两个步骤：1) 决策树的生成，即利用训练样本集生成决策树的过程；2) 决策树的剪枝，在生成决策树之后需要对其进行校验、校正与修改。决策树构造算法的输入样本集合形式如下：



其中表示集合中第条样本的第个属性的取值，第条样本的类型标记。决策树构造的结果是一颗二叉树或多叉树，二叉树一般用于属性都是布尔逻辑判断的数据集合。

决策树构造过程如下：

1. 首先构造决策树的根节点，根节点包含了所有的样本集合和属性集合。
2. 选择当前树中一个叶子结节点，判断其包含的所有样本是否属于同一类型，如果是则为该叶子节点标记类型；然后选择其它叶子节点进行判断，直到所有叶子结点被标记，结束决策树的构建过程。
3. 假如一个叶子节点，中包括了属于多个类型的样本集合；首先计算中样本对每一个属性的信息增益或者信息增益率，选择拥有最大信息增益或信息增益率的属性作为当前节点的分裂属性。
4. 根据分裂属性的取值对节点样本进行划分，得到多个分支节点。
5. 重复2~5步骤。

#### 2.5.8.2 分裂属性选择

不同的决策树分类算法使用不同的判断条件来选择分裂属性，最主要的两种判断条件就是信息增益与信息增益率。本文中使用的CART决策树分类算法使用信息增益作为判断条件，以下介绍根据信息增益进行分裂属性选择方法。

假设训练样本集合为,属性集合为，类型标记集合为,中总的样本量为。使用属性划分得到的第类的样本集合为，集合中的样本个数为；则样本数据集合中属于第类的样本所占的比例为



此时样本数据集合的信息熵为



假设样本数据集合中，属性对应的值域为，表示属性取值为的样本子集。则样本集合对属性的信息增益为：



计算集合每一个属性的信息增益值。假如最大的信息增益值为，则选择属性作为当前样本集合的分裂属性。

#### 2.5.8.3 CART决策树分类算法

CART分类算法是由Breiman等人提出的一种新型算法，它使用最小指数作为属性选择的判定条件。指数常用于度量数据划分或者训练样本集的不纯度，其计算公式如下：





上式中表示样本合计，样本集合中有条样本数据，表示第个样本在样本集合中所占的比例，表示样本集合的指数。后一个公式中和是由划分得到的两个子集，为划分后的两个样本子集的指数加权和。指数常常考虑属性的二元划分，比如样本集合的一个离散型属性，首先将属性的所有取值划分为两个子集和,通过这两个子集对样本集合进行划分；将集合中属性的取值在子集内的样本划分到一个节点中，不在内的样本划分到另一个节点中。

CART属于二分递归分类算法的原因就是指数的这种二元划分特性。CART每个分支节点的属性都只有两种取值：真和假；构建时首先对属性进行布尔测试，条件为真的划分为左子树，条件为假的划分到右子树。

在构建CART决策树时，不管节点是否被划分，CART都为该节点做类型标记，类型标记的判断公式如下:



其中表示将类样本错误分为类的代价；表示类型的先验概率值，表示样本集合中属于类型的样本数量；表示节点内属于类型的样本数量。类型标记判断方法是：如果节点的样本集合中所有样本对于类型以外的所有类型都能使上述公式成立，就将标记为类。

CART在节点剪枝中使用代价-复杂性剪枝算法。当决策树复杂度达到一定程度后，决策树的分类精确度会随着复杂度的再次提高而降低，因此需要在决策树构建之后对其复杂度进行计算并判断是否需要做剪枝处理。剪枝是从树的底部开始，计算以每个内节点为根的子树的代价复杂度和子树剪枝后节点的代价复杂度；比较这两个代价复杂度值，如果减去子树导致的代价复杂度更低，则实施剪枝。

## 2.6 传统评价系统

### 2.6.1 SAPSII评分【20】

Le Gall等自1984年提出SAPSI后，又对其进行了不断更新、不重合完善。他们在对12个欧美国家的137个ICU内的12997例患者进行了研究之后，与1993年提出了SAPSII。

SAPSII由两部分组成：SAPSII评分和PHM计算。SAPSII评分由17项变量（生理学变量12项、年龄、住院类型及3种慢性疾病：获得性免疫缺陷综合征（AIDS）、转移癌和血液恶性肿瘤）构成，每项变量分值不等，最低0分，最高26分，总分0~163分，生理学变量仍取患者入住ICU后第一个24小时内的最差值（即得分最高者），缺失项视为正常，总分越高，表示病情越重。

### 2.6.2 SOFA评分【21】

SOFA之前的许多评分系统经常将器官衰竭描述成“有”或“无”，并且仅将死亡率作为临床试验评价指标。而在ICU多源性的患病人群中，较难证实某新疗法对死亡率的重大影响，但此新疗法可能对个别器官功能大有益处，如将发病率作为考察指标便可显现该疗法的意义。发病率的危险因素不同于死亡率。

基于以上的认识，1994年 ESICM( EuropeanSociety of Intensive Care Medicine)的学者们在巴黎提出了SOFA。SOFA创建的原则在于:寻找一个客观而简单的方法并能以连续的形式描述单个器官的功能障碍或衰竭，同时能评价从轻微的功能障碍到重度衰竭的程度，能在临床研究中反复计量单个或全体器官功能障碍的发生发展，由此确定描述器官功能障碍或衰竭的特征。

SOFA评分的目的是描述MODS的发生、发展并评价发病率，它所采用的变量均为持续变量，具有客观、简单、容易获得及可靠的特点，对所评价的器官功能有特异性，每个医疗机构都能以常规的方法每天检测，并避免了有创性的操作。这些变量与病人来源、病种、人口统计学特征等因素无关，与治疗措施无关。它能区分单个器官功能障碍或衰竭的程度，这样可使评分能广泛应用，促进不同来源的病人之间的比较。这并非与之前的评分系统相背离，而是对它们的补充和完善。

### 2.6.3 APSIII评分【23】

急性生理学评分(APS)是国内常用的评价危重症监护室患者的病情危重程度和预后评估的评分系统。APS是以血液化验和患者的生命体征为基础，它主要适用于急诊危重症监护室(EICU)患者群体进行比较检测，准确率较高，已经广泛应用于临床治疗。不少学者指出，APS分值越高，患者的病情越重，病死率越高。

该评分来自17个生理参数，包括生命体征，动脉血气测量，格拉斯哥昏迷量表(GCS)，血液标本的实验室结果和尿量。

### 2.6.4 MEWS评分【22】

制定MEMS评分系统根据患者生命体征参数及意识状态情况制定MEWS评分系统。。此系统前3项参考指标为心率、收缩压、呼吸频率，每顶参数的分值范围为0-3分，最高分为3分，最低分为0分。分值越高说明病情越危重(排除假阳性)，护士应该积极应对，提前做好抢救准备。特别地，MEWS评分只适用于年龄大于14岁的患者。

运用MEMS预警评分标准量表对急诊手术患者进行评估接到急诊手术通知单后，巡回护士立即前往病区进行术前访视，综合判断，同时给予MEWS预警评分与相应护理。分值4.1-5.0分，表示病情有潜在变化的风险，应立即通知主管医生及麻醉师:分值5.1-7.9分，病情较重，潜在危险增大，应立即通知主管医生及麻醉师、通知配台的器械护士及护士长、联系重症医学科，做好抢救准备；分值≥8分，病情危重，应立即建立至少2条静脉留置针通路，通知手术医生做好手术准备，病房护士快速做好备皮、备血、抽血等，麻醉师及器械护土做好气管插管及手术器械等抢救准备。

### 2.6.5 OASIS评分【24】

疾病严重程度评分因使用预测死亡率和住院时间等结果备受关注。复杂的评分系统需要收集大量的生理测量值，这使得在实际应用中变得难以实现。Johnson AE 首先通过遗传算法模型对特征进行降维，挑选变量与结果间非线性相关性较大的特征子集，根据变量观察数值计算变量的十分位数，随后使用粒子群算法优化的机器学习算法确定每个十分位数的权重。将权值线性组合并应用于内部验证数据集以确保足够的模型准确性。Johnson AE将这一技术命名为牛津急性疾病严重程度评分(Oxford acture severity of illness score, OASIS)。报道称OASIS对患者的ICU病死率和住院期间病死率有较好的判断效果，与 APACHE IV的急性生理评分预测结果大致相同，且能够有效地解释生理参数与结果之间的非线性关系。特别地，OASIS评分系统使用的变量中包含患者第一天进入ICU的最差值，这导致OASIS评分仅在进入ICU一天后才可以使用，这可能会延误病情变化迅速的患者的治疗。

## 2.7 模型实现流程



图x.机器学习与传统评价系统预测流程图

1). 数据提取并进行数据补齐；

2). 以death作为标签，对数值特征进行显著性检验，对无显著性差异的9个特征进行删除，剩余151个特征；

3).对数据集进行标准化处理。以death为标签，根据ReliefF、Fisher、Gini模型进行特征值排序。对每个模型排序结果进行0-1标准化。并根据下式计算数据对特征进行排序，FSscore = ReliefF\_score + Fisher\_score – Gini\_score。

4).根据排序后的特征逐次增加每一个机器学习模型所使用的特征数量，并进行十折交叉验证并计算评价指标BER。选择BER最小时对应的特征数量作为最佳的特征数。使用两种方法分别计算最小特征子集用于后续比较；

5).分别对降维前后的4种数据集（特征全集、最优特征子集，最小特征子集一、最小特征子集二）根据不同分类器进行十折交叉验证，并根据验证集预测结果评价指标对不同的机器学习模型进行调参；在特征全集上对使用传统评分系统的预测结果进行评价。其中（训练集+验证集）：测试集=8:2。特别的，对机器学习训练集使用欠采样与非欠采样，默认分类阈值与最优分类阈值多种方法进行患者生死预测。随后将机器学习结果与传统评分系统预测结果进行比较。

# 3结果

## 3.1 实验材料

为了评价算法性能，采用MIMIC III数据库数据对模型进行评估。算法性能评分方法是使用算法模型对患者生死情况进行预测，再将得到的结果与患者真实的生死结果进行比较。MIMIC-III（Medical Information Mart for Intensive Care）是一个大型的单中心数据库，包含有关在大型三级医院接受重症监护病房入院的患者的信息。 数据包括生命体征，药物使用情况，临床测量数据，护理人员提供的观察和记录，程序代码，诊断代码，成像报告，住院时间，生存数据等。该数据库被广泛应用于学术和工业研究，质量改进计划和高等教育课程【26】。数据库可以从网站[https://mimic.physionet.org/tutorials/install-mimic-locally-windows/](ICU预警_（3改）_增加引言+修改部分初稿.docx)下载。

仿真实验环境如下Intel (R) Core (TM) i5-7200 CPU @2.50GHz 2.70GHz 8GB内存，相关程序由MATLAB 2014b，python3.6.0 与R 3.5.1编程实现。

## 3.2 评价指标

为了更加直观的定量分析算法性能，此处使用一些常用的性能指标对插补结果及机器学习与传统评分系统预测结果进行评价。

### 3.2.1 数据补齐结果评价指标

本文评价插补结果采用以下指标：

平均绝对误差 (Mean Absolute Error, MAE)，定义为绝对误差的平均值，计算公式为：



其中表示预测值，表示真实值。

均方误差 (Mean Squared Error, MSE)，定义为参数估计值与参数真值之差平方的期望值，计算公式为：



均方根误差 (Root Mean Square Error, RMSE)，定义为均方误差的算术平方根，计算公式为：



平均绝对百分误差 (Mean Absolute Percentage Error, MAPE) 计算公式为：



对于一个给定的离散变量，测量变量的平均不确定性的变量信息熵 (Entropy) 可以定义为



其中是离散变量为值时的概率。

互信息MI的定义式为



在显著性检验中，P值定义为拒绝域的面积或概率，计算公式为：



其中，常被设置为0.05。

### 3.2.2 预测结果评价指标

对于一个机器学习算法而言，常用到的评价标准有：

真阳性 (The Positive, TP) :正类数据中被预测为正类的数据；

假阳性 (False Positive, FP) :负类数据中被预测为正类的数据；

真阴性 (True Negative, TN) :负类数据中被预测为负类的数据；

假阴性 (False Negative, FN) :正类数据中被预测为负类的数据。

利用这几个指标通过额外计算还可以得到如敏感度 (TPR)，特异性 (TNR), 贝叶斯错误率 (BER)，F1分数 (f1score)，精确度 (ACC) 及Matthews相关系数 (MCC) 等重要指标，它们的定义及计算公式描述如下：

敏感性 (The Positive Rate, TPR)，定义为正类点被算法模型正确预测为正类点的比例，计算公式为



特异性 (True Negative Rate, TNR)，定义为负类点被算法模型正确预测为负类点的比例，计算公式为



贝叶斯错误率 (Bayes Error Rate, BER)，定义为任意一个分类器在一个数据集上能取得的最低的错误率，计算公式为



F1分数 (F1score)，定义为准确率与召回率的调和平均数，计算公式为



精确度 (Accuracy, ACC)，定义为算法模型正确预测的点占总样本点 (包括正类点与负类点) 的比值，计算公式为



Matthews相关系数 (Matthews Correlation Coefficient,MCC)，计算公式为



其中TP， FP， TN和FN分别表示真阳性，假阳性，真阴性和假阴性。

AUC (Area Under Curve) 定义为受试者工作特征曲线 (Receiver Operating Characteristic Curve, ROC) 下的面积，常常被用来表示算法的优劣。计算公式为



其中是ROC曲线上的点数，为ROC曲线纵坐标，为横坐标。

## 3.3 插补结果

### 3.3.1 无创血压插补结果

再对插补结果进行比较时除了使用mae、mse、rmse、mape等传统评价指标，特别地，为了评价数据补全模型插补结果与原数据在数据结构与分布上的变化，又增加了互信息，T检验等评价方法以检测模型插补结果的非线性相关性与显著性差异的强弱。

对于KNN，与随机森林数据补全方法，使用尽可能少而有效的变量对缺失数据进行预测，在此我们使用的辨别有，年龄，性别，是否接入呼吸机，进入ICU类型，在进行模型学习时，对性别，是否接入呼吸机，进入ICU类型等变量进行独热编码。

数据补全模型在患者无创血压上的插补结果如下：

表1. 无创收缩压插补结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mae | mse | rmse | mape | MI | P值 |
| 中位数 | 0.154 | 0.201 | 0.449 | 0.489 | 0 | 0 |
| 平均值 | 0.156 | 0.198 | 0.446 | 0.209 | 0 | 0.0644 |
| 众数 | 0.16 | 0.229 | 0.478 | 1.138 | 0 | 0 |
| KNN | 0.071 | 0.046 | 0.215 | 0.521 | 0.8767 | 0.6395 |
| 随机森林 | 0.097 | 0.084 | 0.29 | 0.879 | 0.7589 | 0.2531 |

表2.无创平均压插补结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mae | mse | rmse | mape | MI | P值 |
| 中位数 | 0.153 | 0.205 | 0.453 | 0.269 | 0 | 0 |
| 平均值 | 0.153 | 0.2 | 0.447 | 0.2 | 0 | 0.0414 |
| 众数 | 0.165 | 0.249 | 0.499 | 0.531 | 0 | 0 |
| KNN | 0.046 | 0.023 | 0.151 | 0.185 | 0.944 | 0.7631 |
| 随机森林 | 0.097 | 0.091 | 0.301 | 0.251 | 0.7388 | 0.6922 |

表3.无创舒张压插补结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mae | mse | rmse | mape | MI | P值 |
| 中位数 | 0.155 | 0.205 | 0.453 | 0.217 | 0 | 0 |
| 平均值 | 0.155 | 0.203 | 0.451 | 0.2 | 0 | 0.064 |
| 众数 | 0.166 | 0.241 | 0.491 | 0.401 | 0 | 0 |
| KNN | 0.065 | 0.043 | 0.207 | 0.205 | 0.8926 | 0.7554 |
| 随机森林 | 0.11 | 0.112 | 0.334 | 0.318 | 0.6691 | 0.4457 |

从以上三个表中可以看到平均值，中位数，众数的插补结果在MI与T检验上的表现并不理想，这表明平均值，中位数，众数等方法对缺数数据进行补全改变了原本的数据结构与数据分布情况。虽然KNN数据补全方法插补结果较优，虽然KNN使用的特征较少，但仍然不可避免的会对数据加入多重共线性的影响。且KNN方法在特征数量大的情况下，计算成本高。在一台普通的计算机上，对于较大的数据往往需要十几小时甚至数天才能得到结果，而随机森林仅仅需要几分钟。基于上述原因，最终选取随机森林作为无创血压插补方法。

### 3.3.2 特征数值插补结果

介于缺失数据的变量的数量较多，但往往缺失比例都小于1%，最大缺失比例为2.5%，在此选取缺失比例大于1%的四个变量对数据补全算法进行检测，选择出最优的插补方法，随后再对每一个缺失数据的变量进行数据补全。

表4.特征qfio2temp插补结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mae | mse | rmse | mape | MI | P值 |
| 中位数 | 3.0896 | 69.0047 | 8.3069 | 0.1206 | 0 | 0 |
| 平均值 | 3.1053 | 68.1562 | 8.2557 | 0.0985 | 0 | 0.594 |
| 众数 | 4.0532 | 150.2971 | 12.2596 | 0.0737 | 0 | 0 |
| KNN | 0.7595 | 8.2674 | 2.8753 | 0.0157 | 0.9441 | 0.0036 |
| 随机森林 | 0.1449 | 0.5726 | 0.7567 | 0.0029 | 0.9958 | 0.9212 |

表5.特征hfio2temp插补结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mae | mse | rmse | mape | MI | P值 |
| 中位数 | 2.6467 | 53.4176 | 7.3087 | 0.0888 | 0 | 0 |
| 平均值 | 2.6926 | 53.0777 | 7.2854 | 0.0873 | 0 | 0.7274 |
| 众数 | 3.5384 | 115.6797 | 10.7555 | 0.0696 | 0 | 0 |
| KNN | 0.6922 | 6.4926 | 2.5481 | 0.0155 | 0.9413 | 0.0484 |
| 随机森林 | 0.1399 | 0.7747 | 0.8801 | 0.0025 | 0.9972 | 0.9692 |

表6.特征qtemptemp插补结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mae | mse | rmse | mape | MI | P值 |
| 中位数 | 0.1091 | 0.1013 | 0.3183 | 0.2954 | 0 | 0.0006 |
| 平均值 | 0.1094 | 0.101 | 0.3178 | 0.003 | 0 | 0.6335 |
| 众数 | 0.11 | 0.1044 | 0.3232 | 0.003 | -0.5575 | 0 |
| KNN | 0.0612 | 0.0334 | 0.1828 | 0.0017 | 0.8524 | 0.8327 |
| 随机森林 | 0.0137 | 0.0038 | 0.0616 | 0.0004 | 0.9811 | 0.7907 |

表7.特征htemptemp插补结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | mae | mse | rmse | mape | MI | P值 |
| 中位数 | 0.1041 | 0.0873 | 0.2954 | 0.0028 | 0 | 0.5745 |
| 平均值 | 0.1041 | 0.0873 | 0.2954 | 0.0028 | 0 | 0.5502 |
| 众数 | 0.1241 | 0.123 | 0.3508 | 0.0033 | 0.9424 | 0 |
| KNN | 0.0565 | 0.0287 | 0.1695 | 0.0015 | 0.8385 | 0.4727 |
| 随机森林 | 0.0092 | 0.0022 | 0.0474 | 0.0002 | 0.9871 | 0.9867 |

与对无创血压数据补全结果类似，从对MI，P值的评价指标结果上来看，平均值，中位数，众数等方法对缺失数据进行补全改变了原本的数据结构与数据分布情况。对比特征缺失数值上的插补结果，随机森林插补效果要明显优于其它算法，此处选择随机森林作为原始数据特征缺失数值的插补方法。

## 3.4 特征选择结果

由于空间限制，所有151个变量在AHE预测中的重要性排序见附录。 作为一个例子，按组合得分排列的39个最重要的特征变量显示在下表中。由于三种特征排序方法是从三种角度对数据进行分析，所以不同方法对特征的评价有可能不同

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rank | Features | Relief | Fisher | gini | combinational score |
| 1 | qrespperc25 | 0.31 | 1.00 | 0.62 | 1.92 |
| 2 | qrespmean | 0.28 | 0.94 | 0.63 | 1.85 |
| 3 | icutype2 | 0.55 | 0.69 | 0.56 | 1.81 |
| 4 | age | 1.00 | 0.42 | 0.35 | 1.77 |
| 5 | qrespperc50 | 0.27 | 0.90 | 0.56 | 1.74 |
| 6 | icutype3 | 0.86 | 0.38 | 0.32 | 1.56 |
| 7 | qrespperc75 | 0.23 | 0.74 | 0.52 | 1.49 |
| 8 | urineoutput | 0.07 | 0.32 | 1.00 | 1.39 |
| 9 | qrespmin | 0.23 | 0.62 | 0.39 | 1.25 |
| 10 | hspo2min | 0.14 | 0.55 | 0.56 | 1.25 |
| 11 | qfio2min | 0.24 | 0.41 | 0.52 | 1.17 |
| 12 | hspo2stddev | 0.07 | 0.55 | 0.55 | 1.17 |
| 13 | hrespperc25 | 0.32 | 0.44 | 0.39 | 1.15 |
| 14 | hspo2range | 0.14 | 0.52 | 0.48 | 1.14 |
| 15 | qnimeanbp\_max | 0.24 | 0.00 | 0.89 | 1.14 |
| 16 | hfio2min | 0.25 | 0.37 | 0.50 | 1.12 |
| 17 | qnidiasbp\_min | 0.22 | 0.41 | 0.44 | 1.08 |
| 18 | qfio2perc\_25 | 0.22 | 0.39 | 0.45 | 1.07 |
| 19 | hgcs | 0.24 | 0.44 | 0.37 | 1.05 |
| 20 | hrespmean | 0.32 | 0.37 | 0.33 | 1.01 |
| 21 | hfio2perc\_25 | 0.22 | 0.35 | 0.44 | 1.01 |
| 22 | hrespperc50 | 0.32 | 0.35 | 0.32 | 0.99 |
| 23 | hnidiasbp\_min | 0.25 | 0.37 | 0.34 | 0.96 |
| 24 | qfio2perc\_50 | 0.25 | 0.35 | 0.35 | 0.95 |
| 25 | qnisysbp\_min | 0.16 | 0.33 | 0.46 | 0.94 |
| 26 | qnisysbp\_range | 0.19 | 0.43 | 0.29 | 0.92 |
| 27 | hnimeanbp\_min | 0.25 | 0.32 | 0.31 | 0.88 |
| 28 | hnisysbp\_min | 0.15 | 0.29 | 0.43 | 0.87 |
| 29 | hrespperc75 | 0.31 | 0.27 | 0.29 | 0.87 |
| 30 | qfio2max | 0.69 | 0.08 | 0.08 | 0.86 |
| 31 | hrespmin | 0.30 | 0.27 | 0.28 | 0.85 |
| 32 | hfio2perc\_50 | 0.23 | 0.29 | 0.33 | 0.85 |
| 33 | bmi\_label1 | 0.24 | 0.33 | 0.27 | 0.85 |
| 34 | bmi\_label2 | 0.24 | 0.33 | 0.27 | 0.85 |
| 35 | qrespmax | 0.20 | 0.33 | 0.30 | 0.83 |
| 36 | hspo2var | 0.00 | 0.27 | 0.55 | 0.83 |
| 37 | hfio2max | 0.66 | 0.08 | 0.08 | 0.81 |
| 38 | qnisysbp\_stddev | 0.16 | 0.39 | 0.26 | 0.81 |
| 39 | hnidiasbp\_perc25 | 0.24 | 0.29 | 0.28 | 0.81 |
| 40 | qhrmax | 0.20 | 0.31 | 0.29 | 0.80 |
| 41 | hfio2mean | 0.26 | 0.24 | 0.30 | 0.80 |

如2.5部分所述，在训练阶段将这些特征按权重降序排序相继添加到机器学习模型的输入集中。如下图所示，基于八个机器学习模型分别计算每次特征加法迭代的BER值（平均值和标准误差）。最佳特征子集由最小BER确定，并且选择最小数量的特征。下图中给出了最佳和最小特征子集的特征数。对于八种机器学习算法，随着添加更多特征，BER不会持续减少，因此可以寻找到更小的最佳特征子集。特征选择方法将所有特征的数量减少了一半以上。所有最小特征子集的特征数也大大减少到不超过39。这39个特征显示在上中。



Regr XGB



GBM Adb



SVM Log



GBN DT

Figure X. 橙色圆点为最佳特征子集，黄色五角星为最小特征子集一，紫色五角星为最小特征子集二，由新模型计算得出。

Figure X.

图X. 特征子集比例图

上图为各个模型特全集，最优特征子集与两种最小特征子集的比例图形，从图中可以明显看到我们所使用的最小特征子集所包含特征的数量远小于传统方法所得到的结果。下面从AUC的角度进一步比较两种模型在验证集上结果的优劣。

表X. 新模型特征子集预测AUC与传统模型的比较

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | regr | XGB | GBM | Adb | SVM | Log | GBN | DT |
| **最小特征子集1 AUC** | 0.879 | 0.903 | 0.909 | 0.863 | 0.864 | 0.841 | 0.798 | 0.803 |
| **最小特征子集2 AUC** | 0.879 | 0.903 | 0.899 | 0.852 | 0.848 | 0.834 | 0.811 | 0.789 |

相比于传统模型，新模型在大大降低特征子集特征数量的同时，在很大程度上防止出现预测准确率大幅降低的情况。其中因为朴素贝叶斯模型对特征数量较为敏感，所以在两种特征子集上准确率变化较大，但这主要是因为该机器学习模型的缺陷引起的，并非是特征子集选择模型的泛化能力低。

为了更加直观的体现两个特征子集选择模型的优劣，两种子集在验证集上的结果将上表绘制为比例图形的形式。

图X. 新模型特征子集预测结果AUC与传统模型比较比例图

从上图可以很直观的看出，除朴素贝叶斯模型本身缺陷引起的准确度大幅度变化的情况外，其它7个模型预测准确率基本保持不变，均在80%以上，特别地XGB与GBM均在90%以上。基于上述结果，我们所开发的最小特征子集选择模型能够很好地完成寻找最下特征子集这一任务，在大幅减少特征数量的同时，能够保证模型性能基本不变。

下文中提及的最小特征子集均是由新模型选取的。

## 3.5 机器学习模型与传统评分系统预测结果

利用特征选择方法获得的最优特征子集和最小特征子集，分别使用机器学习方法：随机森林，XGB，GBM，Adb，SVM，Log，GNB，DT与传统评分系统：Sapsiii，Sofa，apsiii，mews，oasis，以（训练集+验证集）：测试集=8:2，训练集：验证集=9:1的形式在测试集上随机重复10次交叉验证。以上预测方法的性能在下表中给出。

表X. **Identification results of algorithms on test sets under different feature subsets**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subset** | **Methods** | **ACC** | **AUC** | **BER** | **F1\_score** | **MCC** | **TNR** | **TPR** | **阈值** | **Characteristic quantity** |
| **All** | **regr** | **0.795791115** | **0.903409479** | **0.1843234** | **0.531585221** | **0.475036924** | **0.84310019** | **0.788253012** | **0.5** | **151** |
| **XGB** | **0.831124968** | **0.926293643** | **0.15510112** | **0.584398977** | **0.535656566** | **0.86389414** | **0.825903614** | **0.5** | **151** |
| **GBM** | **0.838139777** | **0.927963665** | **0.15103486** | **0.594661932** | **0.546649317** | **0.86389414** | **0.834036145** | **0.5** | **151** |
| **Adboost** | **0.808002078** | **0.881588927** | **0.19710952** | **0.53257432** | **0.468013521** | **0.79584121** | **0.809939759** | **0.5** | **151** |
| **SVM** | **0.782800727** | **0.886343009** | **0.18788063** | **0.518987342** | **0.463054891** | **0.85255198** | **0.771686747** | **0.5** | **151** |
| **log** | **0.779683035** | **0.86103184** | **0.22067637** | **0.492822967** | **0.420966499** | **0.77882798** | **0.779819277** | **0.5** | **151** |
| **GBN** | **0.806183424** | **0.814729713** | **0.27841603** | **0.461760462** | **0.366770175** | **0.60491493** | **0.838253012** | **0.5** | **151** |
| **DT** | **0.740452065** | **0.827228289** | **0.24897938** | **0.447761194** | **0.366901291** | **0.76559546** | **0.736445783** | **0.5** | **151** |
| **Optimal** | **regr** | **0.779942842** | **0.898488282** | **0.19271557** | **0.51349799** | **0.455576229** | **0.84499055** | **0.769578313** | **0.5** | **124** |
| **XGB** | **0.833463237** | **0.927421596** | **0.15295112** | **0.588310854** | **0.540156262** | **0.8657845** | **0.828313253** | **0.5** | **121** |
| **GBM** | **0.83424266** | **0.934441547** | **0.14932101** | **0.591549296** | **0.544966915** | **0.87334594** | **0.828012048** | **0.5** | **149** |
| **Adboost** | **0.795791115** | **0.881487576** | **0.20577698** | **0.516009852** | **0.449070462** | **0.79206049** | **0.796385542** | **0.5** | **123** |
| **SVM** | **0.787737075** | **0.892198852** | **0.18184088** | **0.526925304** | **0.473192637** | **0.86011342** | **0.776204819** | **0.5** | **125** |
| **log** | **0.775266303** | **0.859106179** | **0.22164746** | **0.489072652** | **0.417324064** | **0.7826087** | **0.774096386** | **0.5** | **150** |
| **GBN** | **0.727201871** | **0.819490059** | **0.26222214** | **0.4312026** | **0.345122719** | **0.75236295** | **0.723192771** | **0.5** | **35** |
| **DT** | **0.717069369** | **0.823391487** | **0.24425832** | **0.440102828** | **0.365195745** | **0.80907372** | **0.702409639** | **0.5** | **121** |
| **Smallest** | **regr** | **0.778643804** | **0.891668754** | **0.18313908** | **0.519187359** | **0.466630138** | **0.86956522** | **0.764156627** | **0.5** | **27** |
| **XGB** | **0.817614965** | **0.918183319** | **0.16134329** | **0.566666667** | **0.517460314** | **0.86767486** | **0.809638554** | **0.5** | **29** |
| **GBM** | **0.82047285** | **0.91816396** | **0.15571378** | **0.573193329** | **0.52620796** | **0.87712665** | **0.811445783** | **0.5** | **29** |
| **Adboost** | **0.797090153** | **0.869438814** | **0.20979143** | **0.514001245** | **0.445050668** | **0.78071834** | **0.799698795** | **0.5** | **16** |
| **SVM** | **0.760457262** | **0.861255039** | **0.21672398** | **0.483183857** | **0.416130769** | **0.8147448** | **0.751807229** | **0.5** | **17** |
| **log** | **0.763574955** | **0.850081422** | **0.23001372** | **0.475201845** | **0.400907625** | **0.77882798** | **0.761144578** | **0.5** | **41** |
| **GBN** | **0.708755521** | **0.820001367** | **0.26179083** | **0.423650386** | **0.340159549** | **0.77882798** | **0.697590361** | **0.5** | **27** |
| **DT** | **0.723304754** | **0.79431298** | **0.28037272** | **0.415156507** | **0.320270204** | **0.71455577** | **0.724698795** | **0.5** | **12** |
| **Traditional scoring system** | **Sapsiii** | **0.721226** | **0.81561** | **0.266481** | **0.425281** | **0.337675** | **0.750473** | **0.716566** | **40** | **-** |
| **Sofa** | **0.645103** | **0.721177** | **0.341596** | **0.3439** | **0.222342** | **0.676749** | **0.64006** | **5** | **-** |
| **apsiii** | **0.727462** | **0.791199** | **0.272401** | **0.42331** | **0.332022** | **0.727788** | **0.72741** | **45.8** | **-** |
| **mews** | **0.736035** | **0.657535** | **0.360397** | **0.345361** | **0.21762** | **0.506616** | **0.77259** | **5** | **-** |
| **oasis** | **0.733437** | **0.806638** | **0.272115** | **0.426174** | **0.334487** | **0.720227** | **0.735542** | **36.5** | **-** |

在所有分析的指标中，八种机器学习算法要显著优于传统评分系统，虽然在ACC指标上有部分机器学习模型未能超过传统评分系统，但ACC模型对模型评价并不全面。BER，MCC，F1score等指标从整体出发对模型进行评价，能够准确评估模型整体性能。从这三个指标的角度出发，机器学习模型是显著优于传统评分系统的，且从敏感性及特异性两个指标可以看出，传统评分系统的Sofa与Mews出现了较为明显的过拟合情况，这在机器学习中并未出现。

当使用ACC和AUC作为性能指标时，使用最佳特征子集的XGB在所有机器学习算法中具有最佳性能，相较于传统评分系统中最优的Sapsiii的预测结果，ACC平均值和AUC均值显着提高了5.27和11.88个百分点。同样从ACC和AUC出发，regr，XGB，GBM的效果相近。对于四种机器学习算法，将特征数量减少到不超过41​​仍然可以提供相对高的性能，ACC平均值高于77％，AUC平均值高于86％。特别的，GBM在特征量下降80.54%的情况下，ACC平均值和AUC平均值分别仅下降1.38％和1.63％。

此外，在对训练集数据欠采样的情况下，对机器学习预测结果的默认阈值进行修改。通过训练集与验证集上预测结果获取使得敏感性与特异性最接近的最优分类阈值，并在测试集上计算最终的预测结果。结果各项指标见下表：

表X. 8种机器学习方法在最优分类阈值下各项指标

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subset** | **Methods** | **ACC** | **AUC** | **BER** | **F1\_score** | **MCC** | **TNR** | **TPR** | **阈值** | **Characteristic quantity** |
| **All** | **regr** | **0.81423746** | **0.90062974** | **0.18634386** | **0.54603175** | **0.48551569** | **0.81285444** | **0.81445783** | **0.53** | **151** |
| **XGB** | **0.84593401** | **0.92696723** | **0.14492763** | **0.60754467** | **0.56088543** | **0.86767486** | **0.84246988** | **0.6** | **151** |
| **GBM** | **0.8355417** | **0.92621507** | **0.15095173** | **0.59187621** | **0.54428205** | **0.86767486** | **0.83042169** | **0.51** | **151** |
| **Adboost** | **0.80072746** | **0.8829583** | **0.20450469** | **0.52092442** | **0.45393339** | **0.78827977** | **0.80271084** | **0.5** | **151** |
| **SVM** | **0.8155365** | **0.88865073** | **0.18479627** | **0.54834606** | **0.48835117** | **0.8147448** | **0.81566265** | **0.56** | **151** |
| **log** | **0.79423227** | **0.85857153** | **0.22257214** | **0.50188679** | **0.42753617** | **0.75425331** | **0.80060241** | **0.53** | **151** |
| **GBN** | **0.78098207** | **0.81949689** | **0.2604468** | **0.46134185** | **0.37193249** | **0.68241966** | **0.79668675** | **1** | **151** |
| **DT** | **0.71057418** | **0.82177614** | **0.24802338** | **0.43451777** | **0.35853115** | **0.80907372** | **0.69487952** | **0.49** | **151** |
| **Optimal** | **regr** | **0.82177189** | **0.90435067** | **0.18277097** | **0.55569948** | **0.49575471** | **0.81096408** | **0.82349398** | **0.54** | **124** |
| **XGB** | **0.83632112** | **0.92191849** | **0.16003485** | **0.58661417** | **0.53471547** | **0.84499055** | **0.83493976** | **0.62** | **121** |
| **GBM** | **0.83632112** | **0.92758956** | **0.14970534** | **0.59354839** | **0.54639638** | **0.86956522** | **0.8310241** | **0.55** | **149** |
| **Adboost** | **0.80618342** | **0.89128727** | **0.18306677** | **0.54120541** | **0.48346732** | **0.83175803** | **0.80210843** | **0.5** | **123** |
| **SVM** | **0.81839439** | **0.89096272** | **0.18393422** | **0.55163566** | **0.49162979** | **0.81285444** | **0.81927711** | **0.59** | **125** |
| **log** | **0.80514419** | **0.86111042** | **0.22896007** | **0.50527704** | **0.42743172** | **0.72400756** | **0.81807229** | **0.55** | **150** |
| **GBN** | **0.73759418** | **0.8182656** | **0.26652755** | **0.43258427** | **0.34325296** | **0.72778828** | **0.73915663** | **0.57** | **35** |
| **DT** | **0.73343726** | **0.82859624** | **0.24351072** | **0.4483871** | **0.37134639** | **0.78827977** | **0.7246988** | **0.49** | **121** |
| **Smallest** | **regr** | **0.79812938** | **0.88946922** | **0.18376255** | **0.53389322** | **0.47720422** | **0.84120983** | **0.79126506** | **0.53** | **27** |
| **XGB** | **0.81293843** | **0.91533981** | **0.16484871** | **0.5599022** | **0.5098709** | **0.8657845** | **0.80451807** | **0.53** | **29** |
| **GBM** | **0.81735516** | **0.91383037** | **0.1718234** | **0.55924765** | **0.50500919** | **0.84310019** | **0.81325301** | **0.54** | **29** |
| **Adboost** | **0.78098207** | **0.86293416** | **0.22310167** | **0.49186257** | **0.41870041** | **0.77126654** | **0.78253012** | **0.5** | **16** |
| **SVM** | **0.77292803** | **0.86442509** | **0.21744084** | **0.49067599** | **0.42131496** | **0.79584121** | **0.76927711** | **0.55** | **17** |
| **log** | **0.77162899** | **0.84943972** | **0.22931793** | **0.48080331** | **0.405824** | **0.76937618** | **0.77198795** | **0.52** | **41** |
| **GBN** | **0.72018706** | **0.822292** | **0.25436975** | **0.4340515** | **0.35304313** | **0.78071834** | **0.71054217** | **0.57** | **27** |
| **DT** | **0.71499091** | **0.81564272** | **0.25340891** | **0.43248836** | **0.35276157** | **0.79017013** | **0.70301205** | **0.49** | **12** |

由于在使用最优分类阈值之前，机器学习模型已经尽量实现了敏感性与特异性的平衡，故在使用最优分类阈值之后模型预测结果并未有明显变化。

特别地，考虑到训练集样本类别失衡情况严重，生存患者人数与死亡患者人数之比接近6/1，而使用SMOTE算法生成新数据或随机重复采样的过采样方法均会造成机器学习模型严重的过拟合情况。最后实验中使用的欠采样方法使训练集数据量损失了将近5/7，如此大的数据损失势必会影响训练出的机器学习性能。在此，我们在对训练集样本不进行类平衡的情况下，通过上述方法使得验证集上敏感性与特异性差异达到最小，并将对应的最优分类阈值作为参数应用到对测试集的分类判断上。且对各个机器学习模型重新调参。

表X. 8种机器学习方法未进行类平衡时在最优分类阈值下预测结果各项指标

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subset | Methods | ACC | AUC | BER | F1\_score | MCC | TNR | TPR | 阈值 | Characteristic quantity |
| All | regr | 0.839958431 | 0.917296217 | 0.168255916 | 0.58490566 | 0.528906752 | 0.820415879 | 0.843072289 | 0.2 | 151 |
| XGB | **0.886464017** | 0.944434259 | 0.129379427 | **0.6726592** | **0.627246** | 0.848771267 | 0.89246988 | 0 | 151 |
| GBM | 0.870355937 | **0.9498132** | **0.1172632** | 0.656099242 | 0.617518939 | 0.899810964 | 0.865662651 | 0.07 | 151 |
| Adboost | 0.803065731 | 0.90044469 | 0.181695686 | 0.539489672 | 0.482950326 | 0.839319471 | 0.797289157 | 0.49 | 151 |
| SVM | 0.8020265 | 0.886009065 | 0.192627599 | 0.530788177 | 0.469140272 | 0.814744802 | 0.8 | 0.12 | 151 |
| log | 0.797090153 | 0.863600337 | 0.227272132 | 0.500319898 | 0.423698496 | 0.739130435 | 0.806325301 | 0.16 | 151 |
| GBN | **0.731618602** | **0.797551074** | **0.273169711** | **0.424512535** | **0.332467244** | 0.720226843 | 0.733433735 | 0.01 | 151 |
| DT | 0.740711873 | 0.84117282 | 0.226580614 | 0.464592275 | 0.395578709 | 0.81852552 | 0.728313253 | 0.15 | 151 |
| Optimal | regr | 0.825409197 | 0.917706744 | 0.169538456 | 0.568677792 | 0.514180449 | 0.837429112 | 0.823493976 | 0.19 | 124 |
| XGB | **0.885944401** | 0.943958253 | 0.123324014 | **0.6755358** | **0.632366** | 0.86389414 | 0.889457831 | 0 | 121 |
| GBM | 0.866718628 | **0.9501304** | **0.1201662** | 0.649350649 | 0.610137559 | 0.897920605 | 0.861746988 | 0.07 | 149 |
| Adboost | 0.802286308 | 0.900634865 | 0.180558339 | 0.539624924 | 0.483784578 | 0.843100189 | 0.795783133 | 0.49 | 123 |
| SVM | 0.810340348 | 0.887564056 | 0.190986631 | 0.539141414 | 0.477050892 | 0.807183365 | 0.810843373 | 0.12 | 125 |
| log | 0.796570538 | 0.86353315 | 0.227573337 | 0.499680511 | 0.422972119 | 0.739130435 | 0.805722892 | 0.16 | 150 |
| GBN | **0.7181086** | **0.807820507** | **0.27702815** | **0.415724286** | **0.322991765** | **0.729678639** | **0.71626506** | 0.1 | 35 |
| DT | 0.736295142 | 0.830713781 | 0.243443244 | 0.449864499 | 0.372447266 | 0.784499055 | 0.728614458 | 0.16 | 121 |
| Smallest | regr | 0.816575734 | 0.908733801 | 0.168302321 | 0.560945274 | 0.508532483 | 0.852551985 | 0.810843373 | 0.18 | 27 |
| XGB | **0.848532086** | **0.9331769** | 0.141037876 | **0.6131387** | **0.5677923** | 0.873345936 | 0.844578313 | 0 | 29 |
| GBM | 0.841257469 | 0.932587628 | **0.1404873** | 0.605042017 | 0.561302932 | 0.884688091 | 0.834337349 | 0.1 | 29 |
| Adboost | 0.770589764 | 0.877582732 | 0.204493873 | 0.498580352 | 0.436200489 | 0.829867675 | 0.761144578 | 0.49 | 16 |
| SVM | 0.729799948 | 0.819951261 | 0.248002881 | 0.443254818 | 0.364330307 | 0.782608696 | 0.721385542 | 0.11 | 17 |
| log | 0.772668226 | 0.850368392 | 0.230304678 | 0.480712166 | 0.405138199 | 0.765595463 | 0.773795181 | 0.15 | 41 |
| GBN | **0.725123409** | **0.810726934** | **0.269783577** | **0.42437432** | **0.334638954** | 0.737240076 | 0.723192771 | 0.1 | 27 |
| DT | 0.769290725 | 0.821854146 | 0.252921516 | 0.460510328 | 0.374882241 | 0.716446125 | 0.777710843 | 0.17 | 12 |

从结果上看，在敏感性与特异性接近的情况下，随机森林，XGB，XGB，GBM等算法预测结果有了明显的提升。最为明显的为XGB在特征全集上的预测结果，ACC比默认分类阈值时高出5.53%，达到88.65%；AUC比默认分类阈值时高出1.81%，达到94.44%。但在SVM，Log，GBN，DT上的表现并不理想，这可能是受模型学习能力限制导致的结果。

# 4 讨论

ICU患者的早期生死预测作为衡量医院医疗水平的重要指标，近年来称为重症医学研究的热点问题，并得到了国内外研究人员的广泛关注。由于ICU患者的临床检测数据具有高性和复杂性，而目前临床上使采用的传统评分方法需要医护人员手工操作，耗费大量人力与时间，不能满足ICU临床的需求。且传统评分方法多是侵入性的，在非入院情况下难以获取，且容易被伪影所干扰，对患者伤害较大，使用传统评分系统所需要的高成本使其在临床应用中有着很大限制。

为了在缺乏侵入性生理参数和生化指标的情况下准确进行ICU患者生死预测，本文开发了一系列使用针对无创特征的机器学习方法与综合特征选择方法。从MIMIC III数据库中选择19245个患者的记录，包括19个无创变量，随后通过计算衍生变量与进行显著性检验确定151个特征变量。通过ReliefF方法，Fisher准则，GINI指数三种特征权值计算方法对数据特征进行多角度的评分与综合排序。并通过不同子集的预测结果确定最佳特征子集与最小特征子集。对在最优、最小特征子集上的8种机器学习算法与传统评分系统进行了全面的比较。

特别的，鉴于最小特征子集在临床中的重要作用，我们开发了一种新的特征子集选择方法，该方法能够实现在几乎不影响模型性能的同时，最大程度的降低特征量，且泛化能力强，结果易被解释，灵活性高。但可惜的是该方法不适用于特征量较小的特征集合。

本文通过对8个具有时序性质的变量分别计算18个衍生参数，进而充分挖掘非侵入式特征数据的数据信息，并结合三种特征权值计算择方法多角度的进一步分析提取特征变量的辨别特性。通过滤波器类型特征选择方法的组合策略（combination strategy of filter-type feature selection method）我们能够对特征进行综合排序。之后使用子集选择模型挑选出每个机器学习算法的最优和最小特征子集，进而更好地分析特征与预测结果之间的关系。

我们的分析表明，最优特征子集消除了冗余的特征信息并构建了具有更好预测结果的机器学习算法。相比之下，只保留了1/4特征量的最小特征子集在大幅减少计算复杂度的同时，仍保证了较高的算法预测性能。鉴于在ICU早期预警装置中使用的微处理器的计算能力的限制，识别这些较小的特征子集在ICU早期预警算法的实现中是非常重要的。

在这项研究中，已经证实了机器学习算法与传统评分系统相比在性能与可操作性上有了很大的提升。且传统评分系统易出现敏感性与特异性不平衡的情况，通过MCC指标可以看出在模型预测性能上，即便是简单的机器学习算法在ICU患者生死预测的分类性能上也要优于传统评分系统。

特别的，XGB与GBM的模型性能明显优于其它模型，这主要得益于这两个模型的优点。XGBoost借鉴RF的做法，支持列抽样，这样不仅能防止过拟合，还能降低计算；XGBoost的代价函数引入正则化项，控制了模型的复杂度，正则化项包含全部叶子节点的个数，每个叶子节点输出的score的L2模的平方和。从贝叶斯方差角度考虑，正则项降低了模型的方差，防止模型过拟合；XGBoost在每次迭代之后，为叶子结点分配学习速率，降低每棵树的权重，减少每棵树的影响，为后面提供更好的学习空间；XGBoost工具支持并行,但并不是tree粒度上的，而是特征粒度，决策树最耗时的步骤是对特征的值排序，XGBoost在迭代之前，先进行预排序，存为block结构，每次迭代，重复使用该结构，降低了模型的计算；block结构也为模型提供了并行可能，在进行结点的分裂时，计算每个特征的增益，选增益最大的特征进行下一步分裂，那么各个特征的增益可以开多线程进行；可并行的近似直方图算法，树结点在进行分裂时，需要计算每个节点的增益，若数据量较大，对所有节点的特征进行排序，遍历的得到最优分割点，这种贪心法异常耗时，这时引进近似直方图算法，用于生成高效的分割点，即用分裂后的某种值减去分裂前的某种值，获得增益，为了限制树的增长，引入阈值，当增益大于阈值时，进行分裂。相比于XGBoost，LightGBM拥有更快的训练速度和更高的学习效率。LightGBM使用基于直方图的算法，例如，它将连续的特征值分桶(buckets)装进离散的箱子(bins)，这使得训练过程中变得更快；使用离散的箱子(bins)保存并替换连续值导致更少的内存占用；它通过leaf-wise分裂方法产生比level-wise分裂方法更复杂的树，这就是实现更高准确率的主要因素。然而，它有时候或导致过拟合，但是可以通过设置 \*\*max-depth\*\* 参数来防止过拟合的发生；相比于XGBoost，由于它在训练时间上的缩减，它同样能够具有处理大数据的能力；支持并行学习。

这项研究有几个局限性，首先，由于数据类不平衡情况严重，生存患者人数与死亡患者人数之比接近6/1，而使用SMOTE算法生成新数据或随机重复采样的过采样方法均会造成机器学习模型严重的过拟合情况。最后实验中使用的欠采样方法使训练集数据量损失了将近5/7，如此大的数据损失势必会影响训练出的机器学习性能。下图是使用随机重复采样的过采样方法与随机抽样的欠采样方法在同一数据集上的预测结果的TPR，即对其中生还患者预测的准确率。虽然过采样的方法改变了原始的数据结构，这两种准确率不能进行直接的比较，但仍可以看出增加训练集数据量能够有效提升随机森林，SVM，XGB与GBM模型性能。

图X. 两种类平衡方法在数据集上预测结果TPR的比较

虽然通过修改最优分类阈值的方式能够在一定程度上减少甚至消除机器学习模型的过拟合问题。但最优分类阈值的调控能力对模型来说是有限的。

其次，由于本研究中的数据集数据多是通过人为筛选的方法以保障数据的高质量，所以文中并未使用数据去噪方法，这提高了本文方法在临床使用中的条件。为了实现在恶劣医疗条件下仍能使用本文方法，应该将开发新的数据去噪算法纳入下一步计划中。

最后，鉴于本文所讨论的数据集仅来自一家医院的患者数据，为了证明模型的泛化能力与用于临床的可能性，需要通过大规模和多中心的临床试验进一步验证这些预测算法，以探索其在实际临床实践中的应用价值。

附录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rank | Features | Relief | Fisher | gini | combinational score |
| 1 | qrespperc25 | 0.31 | 1.00 | 0.62 | 1.92 |
| 2 | qrespmean | 0.28 | 0.94 | 0.63 | 1.85 |
| 3 | icutype2 | 0.55 | 0.69 | 0.56 | 1.81 |
| 4 | age | 1.00 | 0.42 | 0.35 | 1.77 |
| 5 | qrespperc50 | 0.27 | 0.90 | 0.56 | 1.74 |
| 6 | icutype3 | 0.86 | 0.38 | 0.32 | 1.56 |
| 7 | qrespperc75 | 0.23 | 0.74 | 0.52 | 1.49 |
| 8 | urineoutput | 0.07 | 0.32 | 1.00 | 1.39 |
| 9 | qrespmin | 0.23 | 0.62 | 0.39 | 1.25 |
| 10 | hspo2min | 0.14 | 0.55 | 0.56 | 1.25 |
| 11 | qfio2min | 0.24 | 0.41 | 0.52 | 1.17 |
| 12 | hspo2stddev | 0.07 | 0.55 | 0.55 | 1.17 |
| 13 | hrespperc25 | 0.32 | 0.44 | 0.39 | 1.15 |
| 14 | hspo2range | 0.14 | 0.52 | 0.48 | 1.14 |
| 15 | qnimeanbp\_max | 0.24 | 0.00 | 0.89 | 1.14 |
| 16 | hfio2min | 0.25 | 0.37 | 0.50 | 1.12 |
| 17 | qnidiasbp\_min | 0.22 | 0.41 | 0.44 | 1.08 |
| 18 | qfio2perc\_25 | 0.22 | 0.39 | 0.45 | 1.07 |
| 19 | hgcs | 0.24 | 0.44 | 0.37 | 1.05 |
| 20 | hrespmean | 0.32 | 0.37 | 0.33 | 1.01 |
| 21 | hfio2perc\_25 | 0.22 | 0.35 | 0.44 | 1.01 |
| 22 | hrespperc50 | 0.32 | 0.35 | 0.32 | 0.99 |
| 23 | hnidiasbp\_min | 0.25 | 0.37 | 0.34 | 0.96 |
| 24 | qfio2perc\_50 | 0.25 | 0.35 | 0.35 | 0.95 |
| 25 | qnisysbp\_min | 0.16 | 0.33 | 0.46 | 0.94 |
| 26 | qnisysbp\_range | 0.19 | 0.43 | 0.29 | 0.92 |
| 27 | hnimeanbp\_min | 0.25 | 0.32 | 0.31 | 0.88 |
| 28 | hnisysbp\_min | 0.15 | 0.29 | 0.43 | 0.87 |
| 29 | hrespperc75 | 0.31 | 0.27 | 0.29 | 0.87 |
| 30 | qfio2max | 0.69 | 0.08 | 0.08 | 0.86 |
| 31 | hrespmin | 0.30 | 0.27 | 0.28 | 0.85 |
| 32 | hfio2perc\_50 | 0.23 | 0.29 | 0.33 | 0.85 |
| 33 | bmi\_label1 | 0.24 | 0.33 | 0.27 | 0.85 |
| 34 | bmi\_label2 | 0.24 | 0.33 | 0.27 | 0.85 |
| 35 | qrespmax | 0.20 | 0.33 | 0.30 | 0.83 |
| 36 | hspo2var | 0.00 | 0.27 | 0.55 | 0.83 |
| 37 | hfio2max | 0.66 | 0.08 | 0.08 | 0.81 |
| 38 | qnisysbp\_stddev | 0.16 | 0.39 | 0.26 | 0.81 |
| 39 | hnidiasbp\_perc25 | 0.24 | 0.29 | 0.28 | 0.81 |
| 40 | qhrmax | 0.20 | 0.31 | 0.29 | 0.80 |
| 41 | hfio2mean | 0.26 | 0.24 | 0.30 | 0.80 |
| 42 | qfio2mean | 0.25 | 0.27 | 0.25 | 0.77 |
| 43 | icutype4 | 0.66 | 0.05 | 0.04 | 0.75 |
| 44 | qnidiasbp\_perc25 | 0.26 | 0.22 | 0.26 | 0.74 |
| 45 | hhrmax | 0.22 | 0.25 | 0.27 | 0.74 |
| 46 | qhrmean | 0.26 | 0.21 | 0.26 | 0.73 |
| 47 | hfio2range | 0.66 | 0.03 | 0.03 | 0.72 |
| 48 | qhrperc75 | 0.20 | 0.23 | 0.27 | 0.70 |
| 49 | hnidiasbp\_mean | 0.25 | 0.22 | 0.23 | 0.70 |
| 50 | qfio2perc\_75 | 0.28 | 0.23 | 0.18 | 0.70 |
| 51 | hhrrange | 0.14 | 0.29 | 0.26 | 0.70 |
| 52 | qnisysbp\_perc25 | 0.24 | 0.08 | 0.38 | 0.69 |
| 53 | qnisysbp\_var | 0.06 | 0.37 | 0.26 | 0.69 |
| 54 | qgcs | 0.46 | 0.10 | 0.12 | 0.68 |
| 55 | qnidiasbp\_range | 0.10 | 0.32 | 0.25 | 0.68 |
| 56 | hfio2perc\_75 | 0.29 | 0.19 | 0.19 | 0.67 |
| 57 | hhrperc75 | 0.23 | 0.18 | 0.25 | 0.66 |
| 58 | qspo2min | 0.10 | 0.27 | 0.29 | 0.66 |
| 59 | hnisysbp\_perc25 | 0.21 | 0.12 | 0.33 | 0.66 |
| 60 | hnidiasbp\_perc50 | 0.24 | 0.20 | 0.21 | 0.65 |
| 61 | hhrperc50 | 0.29 | 0.15 | 0.21 | 0.65 |
| 62 | qhrperc50 | 0.23 | 0.19 | 0.24 | 0.65 |
| 63 | hhrmean | 0.29 | 0.15 | 0.21 | 0.65 |
| 64 | qnimeanbp\_min | 0.31 | 0.12 | 0.21 | 0.64 |
| 65 | qspo2stddev | 0.07 | 0.27 | 0.29 | 0.63 |
| 66 | qnimeanbp\_perc75 | 0.15 | 0.04 | 0.44 | 0.63 |
| 67 | hnisysbp\_range | 0.15 | 0.27 | 0.20 | 0.63 |
| 68 | hnimeanbp\_range | 0.20 | 0.22 | 0.20 | 0.62 |
| 69 | qhrperc25 | 0.27 | 0.15 | 0.20 | 0.62 |
| 70 | qspo2range | 0.10 | 0.24 | 0.28 | 0.62 |
| 71 | qspo2perc25 | 0.07 | 0.29 | 0.25 | 0.61 |
| 72 | qnidiasbp\_perc50 | 0.27 | 0.13 | 0.21 | 0.61 |
| 73 | icutype5 | 0.60 | 0.00 | 0.00 | 0.61 |
| 74 | qspo2mean | 0.05 | 0.32 | 0.23 | 0.61 |
| 75 | hnimeanbp\_perc25 | 0.24 | 0.14 | 0.21 | 0.60 |
| 76 | hnimeanbp\_stddev | 0.15 | 0.22 | 0.22 | 0.59 |
| 77 | htempperc25 | 0.23 | 0.07 | 0.29 | 0.59 |
| 78 | qnidiasbp\_stddev | 0.11 | 0.28 | 0.20 | 0.59 |
| 79 | hnisysbp\_mean | 0.22 | 0.06 | 0.29 | 0.57 |
| 80 | hnisysbp\_stddev | 0.14 | 0.25 | 0.18 | 0.57 |
| 81 | qnidiasbp\_mean | 0.26 | 0.11 | 0.20 | 0.57 |
| 82 | hhrperc25 | 0.27 | 0.11 | 0.19 | 0.57 |
| 83 | qnisysbp\_mean | 0.25 | 0.02 | 0.29 | 0.56 |
| 84 | hrespmax | 0.22 | 0.14 | 0.19 | 0.56 |
| 85 | gender1 | 0.46 | 0.05 | 0.04 | 0.55 |
| 86 | gender2 | 0.46 | 0.05 | 0.04 | 0.55 |
| 87 | hnisysbp\_perc50 | 0.23 | 0.05 | 0.26 | 0.55 |
| 88 | qhrmin | 0.27 | 0.11 | 0.17 | 0.55 |
| 89 | hhrstddev | 0.09 | 0.24 | 0.21 | 0.54 |
| 90 | hnidiasbp\_perc75 | 0.24 | 0.14 | 0.15 | 0.53 |
| 91 | icutype1 | 0.53 | 0.00 | 0.00 | 0.53 |
| 92 | qnisysbp\_perc50 | 0.23 | 0.02 | 0.27 | 0.52 |
| 93 | qnimeanbp\_mean | 0.26 | 0.08 | 0.18 | 0.52 |
| 94 | hnisysbp\_var | 0.05 | 0.30 | 0.18 | 0.52 |
| 95 | qnimeanbp\_perc25 | 0.25 | 0.09 | 0.17 | 0.51 |
| 96 | htempmean | 0.23 | 0.03 | 0.25 | 0.50 |
| 97 | hfio2stddev | 0.45 | 0.03 | 0.02 | 0.50 |
| 98 | htempperc50 | 0.23 | 0.04 | 0.24 | 0.50 |
| 99 | qnimeanbp\_perc50 | 0.25 | 0.08 | 0.17 | 0.49 |
| 100 | htempmin | 0.09 | 0.10 | 0.29 | 0.48 |
| 101 | hhrvar | 0.01 | 0.26 | 0.21 | 0.48 |
| 102 | hspo2perc50 | 0.02 | 0.10 | 0.34 | 0.47 |
| 103 | qhrrange | 0.14 | 0.16 | 0.16 | 0.46 |
| 104 | qtempperc75 | 0.20 | 0.01 | 0.25 | 0.46 |
| 105 | qtempmean | 0.19 | 0.03 | 0.24 | 0.46 |
| 106 | qhrstddev | 0.13 | 0.16 | 0.15 | 0.44 |
| 107 | qnidiasbp\_perc75 | 0.22 | 0.05 | 0.17 | 0.44 |
| 108 | htempperc75 | 0.21 | 0.01 | 0.21 | 0.43 |
| 109 | hnimeanbp\_mean | 0.18 | 0.05 | 0.19 | 0.42 |
| 110 | qnidiasbp\_var | 0.03 | 0.19 | 0.20 | 0.42 |
| 111 | vent1 | 0.10 | 0.17 | 0.14 | 0.42 |
| 112 | vent2 | 0.10 | 0.17 | 0.14 | 0.42 |
| 113 | qspo2perc50 | 0.05 | 0.22 | 0.15 | 0.42 |
| 114 | hnisysbp\_perc75 | 0.22 | 0.02 | 0.18 | 0.42 |
| 115 | qtempmax | 0.19 | 0.01 | 0.22 | 0.41 |
| 116 | qspo2var | 0.01 | 0.10 | 0.29 | 0.40 |
| 117 | qtempperc50 | 0.17 | 0.02 | 0.21 | 0.40 |
| 118 | qnisysbp\_perc75 | 0.24 | 0.00 | 0.15 | 0.40 |
| 119 | hhrmin | 0.23 | 0.02 | 0.15 | 0.39 |
| 120 | hrespstddev | 0.20 | 0.00 | 0.18 | 0.38 |
| 121 | hresprange | 0.23 | 0.00 | 0.14 | 0.38 |
| 122 | qnisysbp\_max | 0.22 | 0.05 | 0.09 | 0.37 |
| 123 | qtempmin | 0.10 | 0.06 | 0.19 | 0.36 |
| 124 | qhrvar | 0.03 | 0.17 | 0.15 | 0.36 |
| 125 | hnimeanbp\_var | 0.02 | 0.11 | 0.22 | 0.35 |
| 126 | hnimeanbp\_perc50 | 0.14 | 0.06 | 0.15 | 0.35 |
| 127 | hspo2perc75 | 0.01 | 0.05 | 0.28 | 0.34 |
| 128 | qtempperc25 | 0.15 | 0.02 | 0.16 | 0.34 |
| 129 | hnidiasbp\_max | 0.20 | 0.04 | 0.09 | 0.33 |
| 130 | qtempstddev | 0.12 | 0.08 | 0.14 | 0.33 |
| 131 | htemprange | 0.04 | 0.16 | 0.12 | 0.31 |
| 132 | qtempvar | 0.02 | 0.15 | 0.14 | 0.31 |
| 133 | hnimeanbp\_perc75 | 0.15 | 0.01 | 0.16 | 0.31 |
| 134 | htempstddev | 0.05 | 0.16 | 0.10 | 0.31 |
| 135 | qnimeanbp\_range | 0.20 | 0.03 | 0.06 | 0.29 |
| 136 | qnimeanbp\_stddev | 0.17 | 0.04 | 0.07 | 0.27 |
| 137 | qresprange | 0.23 | 0.01 | 0.03 | 0.26 |
| 138 | qspo2perc75 | 0.04 | 0.14 | 0.08 | 0.26 |
| 139 | hrespvar | 0.05 | 0.02 | 0.18 | 0.24 |
| 140 | hnidiasbp\_range | 0.15 | 0.05 | 0.04 | 0.23 |
| 141 | qrespstddev | 0.19 | 0.01 | 0.03 | 0.23 |
| 142 | qnidiasbp\_max | 0.13 | 0.02 | 0.06 | 0.21 |
| 143 | hfio2var | 0.17 | 0.02 | 0.02 | 0.21 |
| 144 | hspo2max | 0.00 | 0.04 | 0.17 | 0.21 |
| 145 | htempvar | 0.01 | 0.10 | 0.10 | 0.20 |
| 146 | hnidiasbp\_stddev | 0.11 | 0.04 | 0.03 | 0.18 |
| 147 | bmi | 0.03 | 0.01 | 0.10 | 0.15 |
| 148 | qnimeanbp\_var | 0.02 | 0.05 | 0.07 | 0.15 |
| 149 | qspo2max | 0.01 | 0.05 | 0.05 | 0.10 |
| 150 | qrespvar | 0.05 | 0.01 | 0.03 | 0.09 |
| 151 | hnidiasbp\_var | 0.03 | 0.03 | 0.03 | 0.09 |