1. **引言**

重症监护室 (Intensive Care Unit，ICU) 是现代医院中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。对ICU患者多项生理指标进行挖掘与分析进而预测患者在ICU期间死亡率，可尽早的对患者病症针对性开展干预治疗，从而减少患者死亡的发生。随着病患数据的进一步积累与完善，传统的统计分析方法在面对高维度，高质量的重症监护数据库时，已逐渐呈现出效率不高，准确率较低等劣势。而医疗数据的专业性和许多难以量化的指标，也使得直接使用部分简单数据处理软件进行ICU患者死亡率预测的效果不尽如人意。

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性与复杂性，采用传统的评分系统无法有效的根据患者特征处理ICU患者死亡率预测问题【2】。主要问题表现在，一是，传统评分没有根据实际的医疗环境进行校准，一致性不高；二是多数评分模型使用的是逻辑回归，变量之间的关系式简单的线性叠加，无法代表真实病例的疾病情况，难以满足临床要求；三是，传统评分方法往往只考虑生理参数当前数值，只是从单一维度考虑，忽略了该参数在时间维度上的变化特征和统计学特征，丢失了很多有用的多维度信息。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】。

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，但该文章所使用的模型复杂度不高，算法之间的对比仍不够充分，且算法使用的参数中包含有有创参数，不易获取。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，但该方法使用的参数在非医院的情况下不易获得，且模型复杂度不高，难以做进一步的提升。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。但该文章使用的参数仅是特征最后的观察值与平均值，没有多维度的挖掘病例信息。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。但同样的，该方法并未考虑从多个维度处理数据。

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，但该方法并没有对缺失值进行填补，而删减带来的结果是维度的减少，这会在一定程度上影响模型性能。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。该方法缺少与传统评价方法之间的比较。

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构建反射状态神经网络（ESN），从而取得更加稳定的预测结果，但算法复杂度不高，且对模型性能的评价指标并不全面。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。但该方法使用的参数中包含有创参数，不易获取，对患者会有一定伤害。

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。但同样的，该方法使用的参数中包含有创参数，对患者有一定伤害，不易获取。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。但该方法仅限制在了在SVM上进行对比，缺少了其它机器学习算法与传统评分模型。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。但该方法并未考虑对缺失值的处理，归一化的过程也仅是去单位化，并未对数据中的噪声进行处理。

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，但并不能解决从实验到应用的可移植性问题。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。该研究并没有使用XGBoost与其它方法进行对比，且对模型性能的评价并不全面。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，机器学习方法优于传统的分析技术。但该文章使用的机器学习模型复杂度不高，难以做进一步的提升，且对模型的评价指标并不完善，难以从指标中推测模型的实际性能。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。但该方法的采用了多种有创参数，在非医院的状态下不易获取。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”。但该文章在类平衡问题上采用的是欠采样处理方法，数据的减少会在一定程度上降低模型性能，且确定类别最优比例的计算成本较高，对于其它算法可借鉴性不强。

ICU患者的生死预测研究在辅助进行临床决策，提高危重病人的生存几率方面具有十分重要的作用。然而，上述研究方法均为有创算法，具有院前急救情况下数据获取不便、获取成本高、获取时间长等问题，限制了其应用范围。另一方面，之前取得相对较好结果的研究，输入特征庞杂，算法计算成本过高，对硬件性能依赖过强，限制了其广泛推广。本文通过利用普通监护仪容易获取的各种无创生理参数训练了随机森林、XGBoost、lightGBM、Adaboost、逻辑回归、支持向量机、决策树、朴素贝叶斯等八种机器学习算法，用于在缺乏有创生理参数和生化指标的情况下对ICU患者生死预测的精确预测。本文通过调节分类阈值解决训练数据的类不平衡问题，通过组合特征选择方法与本文提出的一种新的特征子集选择模型确定最优和最小的特征子集，以实现在保证机器学习模型性能的基础上尽可能的降低计算成本的目的。同时，为了考察算法性能，我们利用Spasiii、Sofa、Apsiii、Mews与Oasis等五种传统评分系统对患者死亡率进行预测，并对十三种模型的性能进行综合对比。

1. **方法**

2.1 定义

在本研究中，仅对患者进入ICU后，30天内院内死亡的情况认定为患者死亡。特别地，对于进入ICU 30天后死亡的患者认定为存活。根据患者信息中以下17个输入生理参数：年龄，性别，BMI，ICU类型，入住与离开 ICU时间，是否死亡，SpO2，体温，心率，无创血压（收缩压，舒张压，平均压），GCS，排尿量，呼吸率，是否接入呼吸机，对患者死亡率进行预测。

学习窗口：定义为患者进入ICU后0~24小时。

预测窗口：定义为患者进入ICU后的1到30天。

非侵入式ICU患者生死预测：如图2.1所示，使用学习窗口中数据训练并判断预测窗口中患者生死情况。



图 2.1 患者生死预测定义

2.2 研究人群

本研究采用Medical Information Mart for Intensive Care III（MIMIC III）中的临床数据。MIMIC-III是一个庞大的、可公开获得的数据库，这些数据与2001年至2012年期间在the Beth lsrael Deaconess Medical Center的危重护理单元的患者有关。该数据库包括诸如人口统计学、在床边进行的生命体征测量(1数据点/小时)、实验室检测结果、程序、药物、护士和医生笔记、成像报告和院外死亡率等信息。MIMIC III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。

该数据库中共包含46520名患者的住院信息。在本研究中，为了简化数据处理过程，去除混杂因素干扰，选取只进入一次医院且仅进入一次ICU的成人患者共29466名，排除掉30天内院外死亡的患者854名，剩余的研究人群中，有9367名患者的17个输入生理参数中缺少超过一个以上的参数，予以排除。最终纳入研究的共19245名患者，其中有2693名患者在入ICU30天内发生院内死亡，其余16552名患者存活。

病例数据选择过程如图2.2所示。



图2.2 数据提取与数据预处理

数据提取具体流程如下：

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).对患者信息完整程度进行分析，人数变化最大的点如图2.3所示。最终选取至少包括88%个有效特征的患者信息，筛选出19245条患者信息；

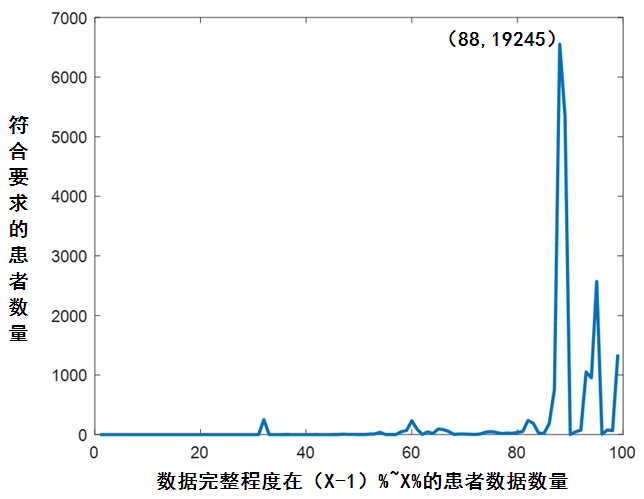


图2.3 患者有效数据量分布

2.3 数据预处理

2.3.1 异常值处理

如果患者信息特征值不在以下范围：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，则被视为异常数值。异常数据将被视为缺失数据，并在下一步中进行重新估算。

2.3.2 独热编码

独热编码即 One-Hot 编码，又称一位有效编码，其方法是使用N位状态寄存器来对N个状态进行编码，每个状态都有它独立的寄存器位，并且在任意时候，其中只有一位有效。

例如对六个状态进行编码：

自然顺序码为 000,001,010,011,100,101

独热编码则是 000001,000010,000100,001000,010000,100000。目前大部分算法都是基于向量空间（欧式空间）中的度量来进行计算的，为了使标量关系的变量取值不具有标量性质，并且在欧式空间中到圆点是等距的。将离散特征的取值扩展到了欧式空间，离散特征的某个取值就对应欧式空间的某个点。对离散型特征使用独热编码，会让特征之间的距离计算更加合理。独热编码中，不同特征之间的欧氏距离是相等的，这种特性经常运用到特征相似度的计算中，它可以让离散型变量的特征连续化。

2.3.3 数据补齐

关于数据补齐的处理，我们选择了5种补齐方法，分别是：使用平均值补齐，使用中位数补齐，使用众数补齐，随机森林与K近邻算法。对插补方法的测试为从原始数据集中选取特征数据完整的数据集，随后对需要进行补齐的特征随机产生1/3的缺失数据，对插补数据与原始数据进行相关评价指标的计算。

特别地，在对无创血压数据（舒张压，收缩压，平均压），FIO2与BMI进行插补时，是对原始数据进行插补。而对上述5种特征插补完成后，首先计算不完整数据集的衍生变量，随后对衍生变量中缺失的数据进行插补。这么处理是因为：无创血压（收缩压，舒张压，平均压）与有创血压（收缩压，舒张压，平均压）具有强相关性，FIO2与BMI更是设置人为规则的数据插补方法，这5种特征的插补结果造成的可能误差会更小。相比之下，在对其它特征进行插补时，若某一数据插补后具有较大的误差，则会造成其18个衍生变量数据均存在相应更多的计算误差。而直接对衍生变量进行插补则能尽可能多的减少这种后续的计算误差，且数据在计算衍生变量后增加了更多的维度，更加有利于模型训练与数据补齐。

2.3.3.1 无创血压数据补齐

针对不同特征数据缺失原因的不同，对缺失数据采取不同的数据补齐方法。对于患者无创血压（收缩压，舒张压，平均压）数据，其缺失数据采用随机森林模型进行数据补齐（测试时使用的方法有平均值，众数，中位数，随机森林与KNN），计算缺失值所用变量包括gender, icutype, age, vent与有创血压（收缩压，舒张压，平均压）。需要注意的是，无创血压（收缩压，舒张压，平均压）的插补在计算衍生变量之前进行，进行计算时需要对标量gender, icutype, vent等进行独热编码。

2.3.3.2 FIO2与BMI在人为规则下的数据补齐

对与FIO2，根据变量vent确定患者是否接入呼吸机。对于vent=0的患者，填补FIO2缺失值为21，即为空气中氧含量。对于进入ICU前12小时含有FIO2数据，12~24小时无数据的情况，由前12小时数据直接对12~24小时数据进行插补；对12~24小时有数据，前12小时无数据的情况，填补缺失值为21。对于BMI，由于该特征有效值缺失量较大，为了避免造成可能的多重共线性，使用平均值进行插补，特别地，根据bmi特征生成标量bmi\_label，来表示bmi数据是否为插补数据。

2.3.3.3 衍生变量中缺失数据的补齐

在对前两部分数据进行数据补齐之后，对不完整的数据集计算衍生变量。由于不完整特征数量较多，在对数据补齐模型进行测试时仅选取了数据缺失量在1%以上的2个变量fio2与Temp的进入ICU前12小时数据的平均值，进入ICU12~24小时数据的平均值共四个变量进行数据插补模型性能的评价，在选取最优数据补齐模型后对其它缺失变量进行数据补齐。

值得注意的是，与无创血压数据补齐使用变量不同，衍生变量的数据补齐使用除每次需要补齐缺失数据的特征变量以外的其它所有变量，即159个插补变量进行数据补齐。特别的，对159个插补变量中的缺失数值使用平均值进行插补。

从图2.4可见，缺失特征数量较大，但仅有fio2与temp的衍生变量缺失比例在1%以上，共计36个衍生变量。考虑到衍生变量中平均值的特性，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，使用五种数据补齐方法对随机缺失的特征数值进行补齐，根据补齐结果评价指标与模型计算成本，最后选择随机森林对原始数据中缺失的特征数值进行补齐；

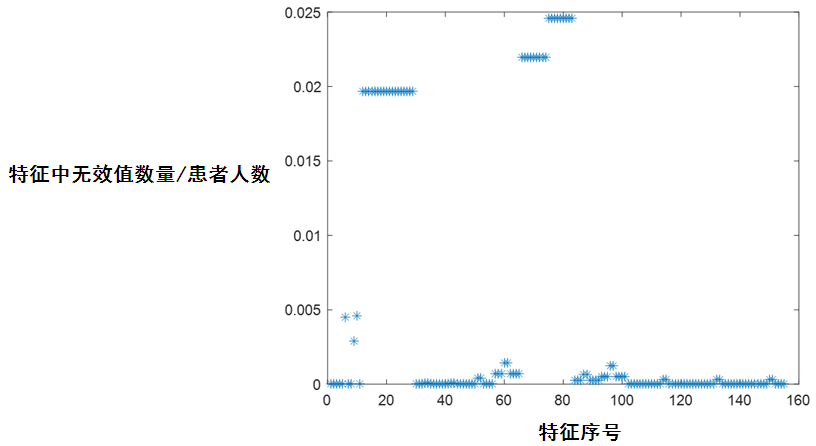


图2.4缺失数据分布情况

上图中，纵坐标在0.01以上的共36个点，对应横坐标为fio2与temp的衍生变量序号。

2.3.4 数据标准化

Z-score是将原始分数与其平均值之差除以标准差所得的商数，其使得数据标准统一化，相比于0~1标准化提高了数据的可比性。Z -score以标准差为单位，度量原始数据距离均值的距离长短。Z-score是一个抽象值，不受原始测量单位的影响，是一种具有相等单位的量数，可接受进一步的统计处理。经过标准化后的数值服从均值为0，方差为1的标准正态分布。

2.4 数据补齐方法

2.4.1 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。

2.4.2 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。

2.4.3 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。

2.4.4 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。

2.4.5 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。

2.5 衍生变量的计算

计算衍生变量的参数有：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)、格拉斯哥昏迷指数(GCS)与吸入氧浓度(FIO2)。

为了捕捉趋势记录的各种统计特征，从多个维度挖掘数据特性，将患者入ICU后24小时的数据记录分为0～12小时与12～24小时两部分，分别计算各部分内各特征的最大值，最小值，平均值，方差，标准差，极差，四分位数，共得到18个统计参数作为衍生变量。同时对gender, icutype, vent, bmi\_label等四个变量进行独热编码。

各个特征衍生变量的计算如表2.1所示，共有160个变量在本研究中被考虑。

Table 2.1. Overview about non-invasive physiological parameters and related feature variables

|  |  |  |
| --- | --- | --- |
| **NIPP type** | **NIPP** | **160 feature variables** |
| **Clinical variables** | **Demographic data** | Age, BMI, urineoutput, gender1, gender2, icutype1, icutype2, icutype3, icutype4, icutype5, vent1, vent2, bmi\_label1, bmi\_label2; |
| **GCS** | qGCS, hGCS; |
| **FIO2** | f\_fio2\_min, f\_fio2\_max, f\_fio2\_mean, f\_fio2\_var, f\_fio2\_stddev, f\_fio2\_range, f\_fio2\_perc25, f\_fio2\_perc50, f\_fio2\_perc75, l\_fio2\_min, l\_fio2\_max, l\_fio2\_mean, l\_fio2\_var, l\_fio2\_stddev, l\_fio2\_range, l\_fio2\_perc25, l\_fio2\_perc50, l\_fio2\_perc75; |
| **Trends records** | **HR** | f\_HR\_min, f\_HR\_max, f\_HR\_mean, f\_HR\_var, f\_HR\_stddev, f\_HR\_range, f\_HR\_perc25, f\_HR\_perc50, f\_HR\_perc75, l\_HR\_min, l\_HR\_max, l\_HR\_mean, l\_HR\_var, l\_HR\_stddev, l\_HR\_range, l\_HR\_perc25, l\_HR\_perc50, l\_HR\_perc75; |
| **RESP** | f\_resp\_min, f\_resp\_max, f\_resp\_mean, f\_resp\_var, f\_resp\_stddev, f\_resp\_range, f\_resp\_perc25, f\_resp\_perc50, f\_resp\_perc75, l\_resp\_min, l\_resp\_max, l\_resp\_mean, l\_resp\_var, l\_resp\_stddev, l\_resp\_range, l\_resp\_perc25, l\_resp\_perc50, l\_resp\_perc75; |
| **TEMP** | f\_temp\_min, f\_temp\_max, f\_temp\_mean, f\_temp\_var, f\_temp\_stddev, f\_temp\_range, f\_temp\_perc25, f\_temp\_perc50, f\_temp\_perc75, l\_temp\_min, l\_temp\_max, l\_temp\_mean, l\_temp\_var, l\_temp\_stddev, l\_temp\_range, l\_temp\_perc25, l\_temp\_perc50, l\_temp\_perc75; |
| **SPO2** | f\_spo2\_min, f\_spo2\_max, f\_spo2\_mean, f\_spo2\_var, f\_spo2\_stddev, f\_spo2\_range, f\_spo2\_perc25, f\_spo2\_perc50, f\_spo2\_perc75, l\_spo2\_min, l\_spo2\_max, l\_spo2\_mean, l\_spo2\_var, l\_spo2\_stddev, l\_spo2\_range, l\_spo2\_perc25, l\_spo2\_perc50, l\_spo2\_perc75; |
| **NISYSBP** | f\_nisysbp\_min, f\_nisysbp\_max, f\_nisysbp\_mean, f\_nisysbp\_var, f\_nisysbp\_stddev, f\_nisysbp\_range, f\_nisysbp\_perc25, f\_nisysbp\_perc50, f\_nisysbp\_perc75, l\_nisysbp\_min, l\_nisysbp\_max, l\_nisysbp\_mean, l\_nisysbp\_var, l\_nisysbp\_stddev, l\_nisysbp\_range, l\_nisysbp\_perc25, l\_nisysbp\_perc50, l\_nisysbp\_perc75; |
| **NIDIASBP** | f\_nidiasbp\_min, f\_nidiasbp\_max, f\_nidiasbp\_mean, f\_nidiasbp\_var, f\_nidiasbp\_stddev, f\_nidiasbp\_range, f\_nidiasbp\_perc25, f\_nidiasbp\_perc50, f\_nidiasbp\_perc75, l\_nidiasbp\_min, l\_nidiasbp\_max, l\_nidiasbp\_mean, l\_nidiasbp\_var, l\_nidiasbp\_stddev, l\_nidiasbp\_range, l\_nidiasbp\_perc25, l\_nidiasbp\_perc50, l\_nidiasbp\_perc75; |
| **NIMEANBP** | f\_nimeanbp\_min, f\_nimeanbp\_max, f\_nimeanbp\_mean, f\_nimeanbp\_var, f\_nimeanbp\_stddev, f\_nimeanbp\_range, f\_nimeanbp\_perc25, f\_nimeanbp\_perc50, f\_nimeanbp\_perc75, l\_nimeanbp\_min, l\_nimeanbp\_max, l\_nimeanbp\_mean, l\_nimeanbp\_var, l\_nimeanbp\_stddev, l\_nimeanbp\_range, l\_nimeanbp\_perc25, l\_nimeanbp\_perc50, l\_nimeanbp\_perc75; |

f\_HR\_min是患者进入ICU 0~12小时HR数据的最小值，l\_HR\_min是患者进入ICU 12~24小时HR数据的最小值。其它缩写规则与f\_HR\_min相近。全部缩写规则如下：f (first 0~12 hours), l (later, 12~24 hours), min (min), max (max), mean (mean), var (variance), stddev (standard deviation), range(range), perc25, perc50, perc75 (quartile).

另外，gender1, gender2, icutype1, icutype2, icutype3, icutype4, icutype5, vent1, vent2, bmi\_label1, bmi\_label2分别为gender, icutype, vent, bmi\_label进行独热编码后的结果。

2.6 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。

本文首先使用显著性检验方法对无显著性的变量进行删除，随后使用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。

特别的，为了保障模型在对测试集进行预测时的可信度，特征排序时仅使用训练集数据。

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 = Normalization\_Relieff\_score + Normalization\_fisher\_score –Normalization\_Gini\_score

2.6.1 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小于或等于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。

本文首先使用显著性检验方法对无显著性的变量进行删除，设定当时拒绝原假设。共删除变量9个，包括q\_fio2\_var, q\_fio2\_stddev, q\_fio2\_range, q\_temp\_range, h\_temp\_max, h\_spo2\_mean, h\_spo2\_perc25, h\_nisysbp\_max, h\_nimeanb­p\_max等。变量对应的P值详见附表。

2.6.2 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：



2.6.3 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。

2.6.4 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。

2.6.5最小特征子集确定方法

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，用以和传统方法进行比较，新的特征子集选择方法被命名为A feature selection framework based on Bayes Error Rate，简记为FSFBER。

值得注意的是，在进行特征子集选择时，使用的是训练集数据进行十折交叉验证所得到的评价指标BER。

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．FSFBER方法具体步骤如下：

1) 按特征排序后的特征顺序，逐一增加特征数量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，选择平均值最小点位置对应的特征数作为最优特征子集的特征数;

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。在上述子映射中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 （即点后的递减下降均没有点之前的递减下降剧烈，点为局部下降最快段的末端，阈值设定了点后的下降程度小于阈值）；

3) 寻找到点后对该点后的严格递减映射的前进行二次求导寻找其中二阶导数最接近0的点()（即为点之后前1/t数据内BER变化最缓慢的点）；

4) 选取以为终点，长度为（对于子映射中变量数量较多的情况,对于数量较少的情况）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。如果对应的小于对应的则认为为最小特征子集。如果对应的大于对应的，且导数值小于（一般设置为），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。



为了更好的理解上述模型，下面以Adaboost模型在不同特征子集下预测结果的BER为例，对最小特征子集选择过程进行描述：

1. 按特征排序后的特征顺序，逐一增加特征数量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，图2.5将预测结果一并列出。其中紫色五角星表示新模型选择出的特征量，黄色五角星为传统模型选择出的特征量。

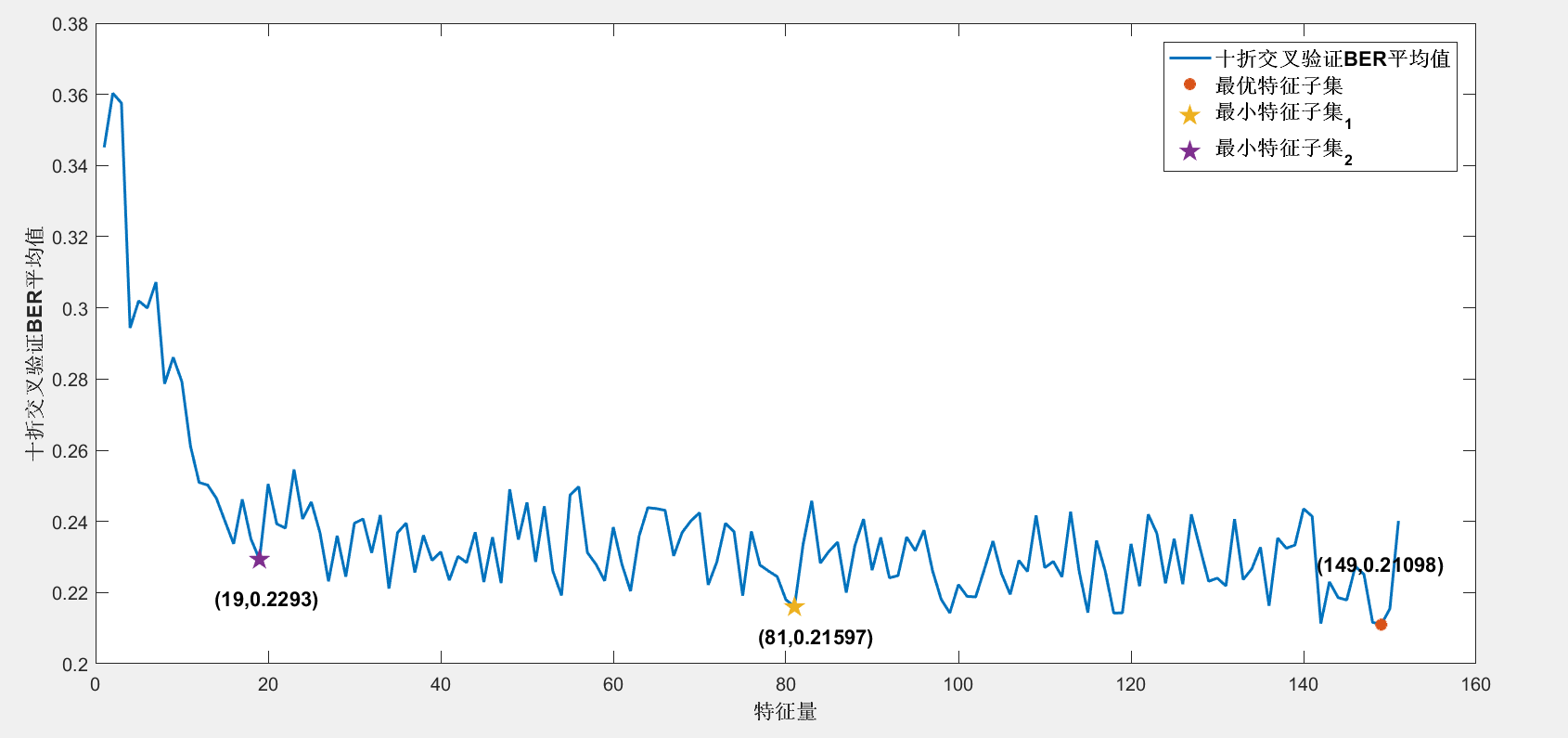


图2.5 不同特征集合下十折交叉验证结果BER平均值

1. 在预测结果BER的全集集合中，以BER为因变量，对应的特征量为自变量。在这个映射中寻找一个最大的严格单调递减的子映射。如图2.6所示

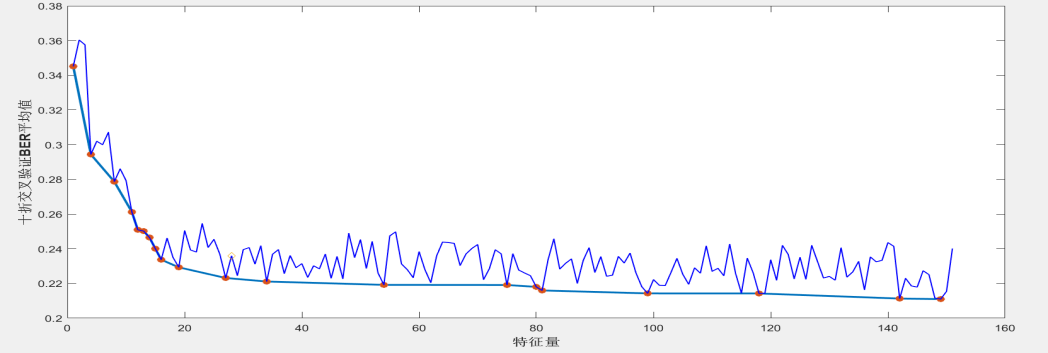


图2.6 最大的严格单调递减的子映射集合

图中由橘红色标注出的点即为最大的严格单调递减的子映射集合。

在上述子映射的基础上对数据进行一阶求导，并寻找这样一个关键点，使得该点以后的一节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（此处设置为0.005）。

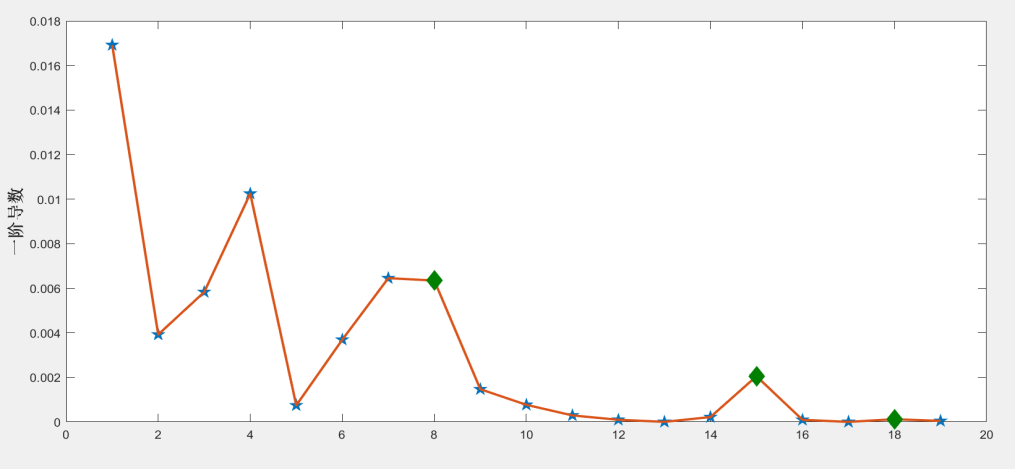


图2.7 子映射的一阶导数图

图2.7中绿色菱形位置即为满足要求的关键点位置，其在原图中对应位置如图2.8。根据模型需要，选取折线下降的末尾位置，在此选择第一个关键点(16, 023371)作为。

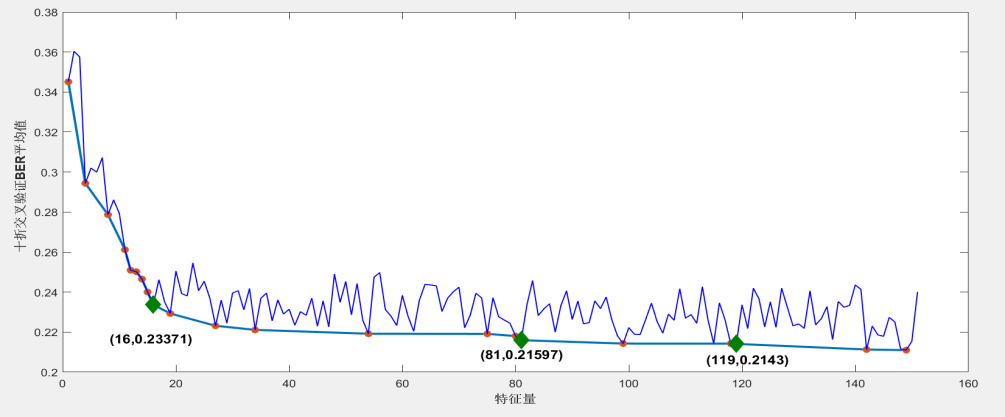


图2.8 在原始映射中的位置

1. 选取第一个关键点后的数据，对该点后的严格下降数据的前(此处取4)进行二次求导寻找其中二阶导数最接近0的点，如图2.9所示;

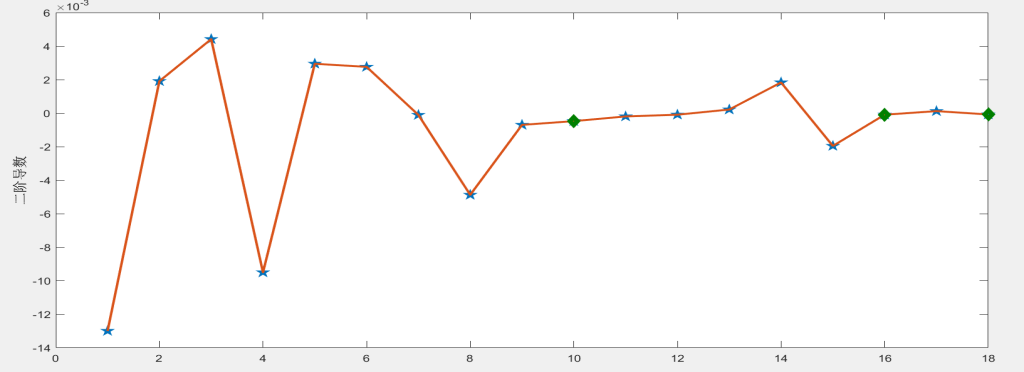


图2.9子映射的二阶导数图

图2.9中绿色菱形位置即为满足要求的关键点位置，其在原图中对应位置如图2.10。根据模型需要，选取折线平缓的开始位置，在此选择第一个关键点(27, 022316)作为（若出现第一个点与重合的情况，选取第二个点作为关键点）。

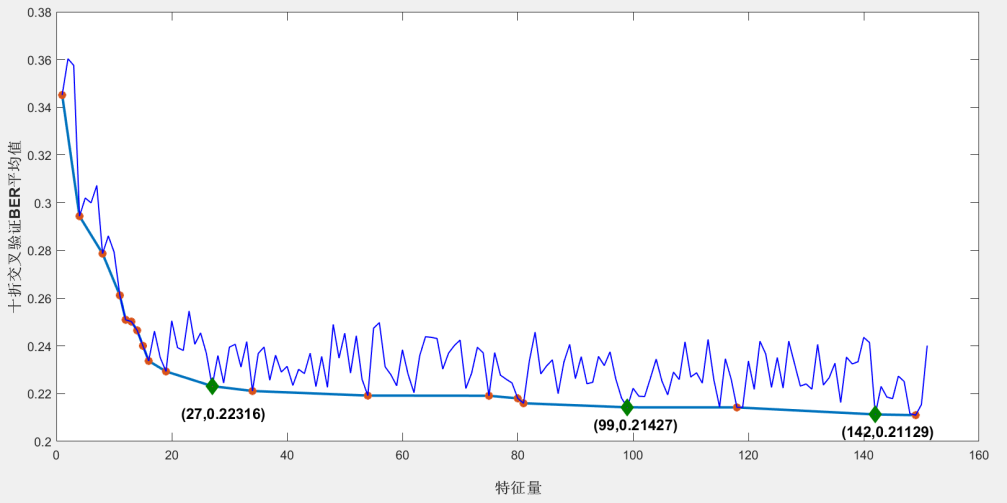


图2.10在原始映射中的位置

1. 选取出这样一点数据，以为终点，以与距离为单边长度。选取出原始数据中对应的数据段，如图2.11中黄色折线标注部分，绿色菱形标注部分分别为选取数据段中的最小点与次小点。

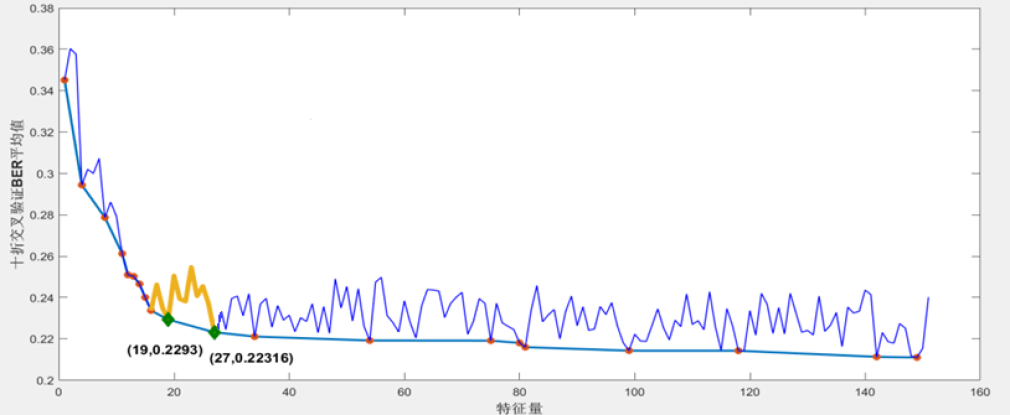


图2.11 候选数据段在原始映射中的位置

设置阈值 ()。计算最小点与次小点之间的一阶导数绝对值。因为该图中最小点位于次小点之后，且两点之间一阶导数0.0061/8<0.001，故选择次小点对应的特征量作为结果输出。

2.7 机器学习方法

特别地，机器学习模型通过训练集进行十折交叉验证完成调参并确定初始最优分类阈值。

2.7.1 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。

在本研究中，XGBOOST关键参数被设置为：学习速率为0.01，森林中树的个数为5000，树的最大深度为5，子节点中最小的样本权重和为5，使用的树构造算法为精确的贪婪算法，指定学习任务和相应的学习目标为通过最小化成对损失来执行排序任务，构造每个树时的子样本比率为0.8，L1正则化项权重为0.005，L2正则化项权重为1，使用树模型作为每次迭代的模型，对于有效数据的度量方法为error。

2.7.2 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。

2.7.2.1 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：



2.7.2.2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。

2.7.2.3 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。

在本研究中，LightGBM关键参数被设置为：树上的叶子数为1000，学习速率为0.005，森林中树的个数为7000，L1正则化项权重为0.008，L2正则化项权重为0，树的最大深度为10，子节点中最小的样本权重和为10，使用传统的梯度提升决策树作为每次迭代的模型，boosting的迭代次数为100。

2.7.3 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。

在本研究中，随机森林关键参数被设置为：森林中树的个数为500，使用信息增益来衡量分类质量，估计泛化误差时使用out-of-bag samples袋外样本，分裂一个内部节点所需的最小样本数为6，在新创建的树叶中最少的样本数为6，随机森林划分时考虑的最大特征数为sqrt(n\_feature)，不限制决策树最大深度，最小样本划分数目为2，叶子节点最小的样本权重和为0，不限制最大叶子节点数，。

2.7.4 AdaBoost

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。

在本研究中，AdaBoost关键参数被设置为：学习速率为0.1，森林中树的个数为500，使用决策树作为基分类器，使用SAMME.R作为模型提升准则。

2.7.5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。

在本研究中，支持向量机关键参数被设置为：错误项的惩罚参数为1，启用概率估计。

2.7.6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。

在本研究中，逻辑回归关键参数被设置为：使用saga作为逻辑回归损失函数的优化方法，使用热启动方法，即每次训练都使用上一次的调用作为初始化。

2.7.7 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。

在本研究中，朴素贝叶斯模型关键参数被设置为：类的先验概率为0.2,0.8。

2.7.8决策树

2.7.8.1 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。

2.7.8.2 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。

2.7.8.3 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。

在本研究中，决策树模型关键参数被设置为：将选择最好的随机分类作为在节点中选择分类的策略，区分一个内部节点需要的最少的样本数为40，叶子节点所需要的最小样本数为80。

2.8 传统评价系统

特别地，传统评分模型通过训练集进行十折交叉验证来确定最优分类阈值。

2.8.1 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评分由年龄，心率，收缩压，体温，格拉斯哥昏迷指数等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。

2.8.2 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。该评分基于六个不同的评分，分别针对呼吸、心血管、肝脏、凝血、肾脏和神经系统。

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。

2.8.3 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。

2.8.4 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。

2.8.5 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。

2.9参数调节

本文所使用的数据集中数据类别共有两类，数据量之比为16552:2693=6.15:1。虽然有数据类别严重不平衡的情况，但本文并未在训练集上对数据进行类平衡处理。这是因为在数据集上操作的类平衡方法主要分为两类，欠采样与过采样。欠采样的方法会减少大量数据，在一定程度上降低模型性能；而过采样的方法无论是重复采样或是生成新的数据均容易使得模型出现过拟合的情况出现。为了避免上述问题，本文使用调节分类阈值的方法来对病态的模型进行调校，故在此引入Cost-Sensitive的概念。

2.9.1最优分类阈值选择模型Cost-Sensitive【27】

2.9.1.1 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

表2.2 混淆矩阵各个数据标识

|  |  |  |
| --- | --- | --- |
|  | **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。

2.9.1.2 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。

* + 1. 调参（与确定最优分类阈值）流程
       1. 机器学习模型的调参流程

Step1. 通过训练集特征全集进行十折交叉验证调节机器学习参数；

Step2. 根据每次交叉验证预测结果设置分类阈值使得敏感性与特异性达到平衡。对十次计算得出的分类阈值取平均值，保存为当前参数与数据下的最优分类阈值

Step3. 重复Step1、2直至机器学习模型性能不再提升；

Step4. 通过组合特征选择方法与本文提出的一种新的特征子集选择模型确定最优和最小的特征子集。

Step5. 通过训练集进行十折交叉验证，三种不同特征集合下对机器学习模型调参（本文中不同特征集合下尝试调参后，参数不变），并获得相应的最优分类阈值；

Step6 通过训练集下的三种不同特征集合训练机器学习模型，设置参数与分类阈值为Step5中获得的数值。对测试集进行模型性能测试。

* + - 1. 传统评分模型的调参流程

Step1. 以预测结果的敏感性与特异性平衡为目的，在训练集进行十折交叉验证调节分类阈值，对十次计算得出的分类阈值取平均值，保存为当前传统评分模型的最优分类阈值。

Step2. 根据Step1中获取的不同模型的最优分类阈值，使用测试集对不同传统评分模型性能进行测试。

2.10 Performance Evaluation

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。

2.10.1 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。

不同于相关系数，互信息并不局限与实值随机变量，它更加一般且决定着联合分布和分解的边缘分布的乘积的相似程度，是度量两个事件集合之间的相关性。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。

2.10.2 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。

2.11 模型实现流程



图 2.12 死亡率预测流程图. 其中RF 是Random Forest的简称，XGB是XGBoost的简称，LGB是LightGBM的简称，AdaBoost是Adaptive Boosting的简称，SVM是Support Vector Machine的简称，LR是Logistic Regression简称，NB 是Naive Bayesian的简称，DT是Decision Tree Model的简称。

如图2.12所示，本研究的开展流程如下

1). 数据提取并进行数据预处理；

2). 计算衍生变量，得到共160个变量。在训练集中以death作为标签，对衍生变量中的数值变量进行显著性检验，对无显著性差异的9个衍生变量进行删除，剩余151个衍生变量；

3). 以death为标签，在训练集中根据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，并通过线性组合的方式计算衍生变量权重并排序。

4).根据排序后的衍生变量逐次增加每一个机器学习模型所使用的衍生变量数量，并根据训练集进行十折交叉验证计算评价指标BER。根据FSFBER模型选择最优与最小特征子集，并通过十折交叉验证获得特征全集，最优、最小特征子集下的模型参数与最优分类阈值。随后使用不同特征集合的训练集对模型进行训练。

5). 使用相应特征集合的测试集对8种机器学习模型与5种传统评分模型进行验证。

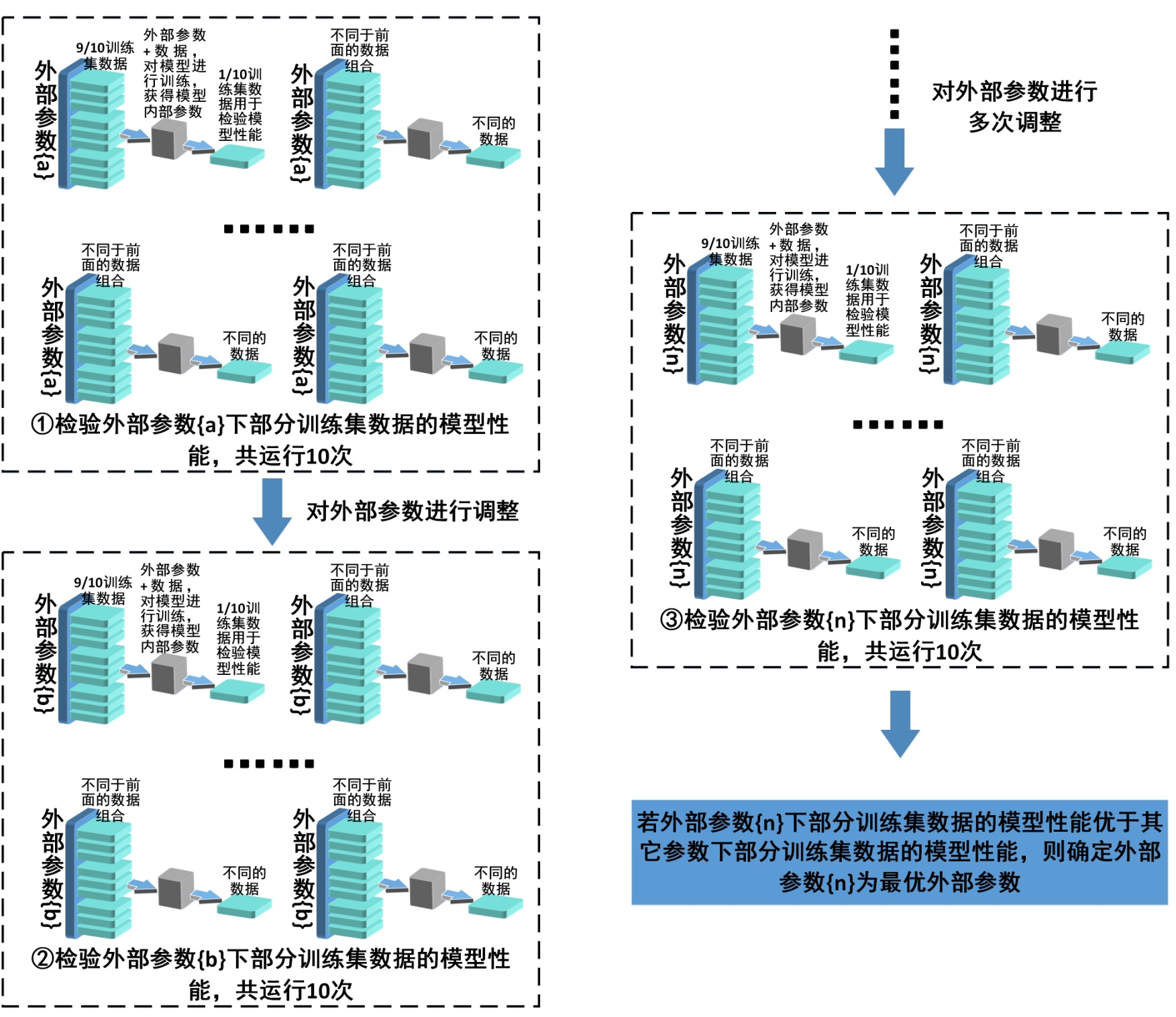
在对机器学习模型进行调参时，仅能够对模型外部参数进行参数调整，模型内部参数由外部参数与训练数据共同得出

图 2.13 机器学习调参流程图

如图2.13所示，本研究的开展流程如下：

(1). 将训练集分为10份，10份数据之间无交集，10份数据之和为训练集数据全集。输入初始外部参数{a}，将9份数据与参数{a}用于模型训练，获得模型的内部参数。用剩余的1份数据对模型性能进行检验；

(2). 选取不同于之前的9份数据的组合与参数{a}对模型进行训练，选取剩余的1份数据对模型性能进行检验。如此重复10次，获得外部参数{a}下部分训练集数据稳定的模型性能；

(3). 对初始外部参数{a}进行调整，得到外部参数{b}，重复步骤1,2，得到外部参数{b}下部分训练集数据稳定的模型性能；

(4). 重复步骤3，通过多次调整获得外部参数{n}，若外部参数{n}下部分训练集数据的模型性能优于其它参数下部分训练集数据的模型性能，则将外部参数{n}确定为最优的外部参数，进而停止参数调整。

1. **Results**

仿真实验环境如下：Intel (R) Xeon E5 V3 CPU @3.50GHz 3.70GHz 8GB内存，相关程序由MATLAB 2016b，python3.6.0 与R 3.5.1编程实现。

3.1 Feature Selection Result

3.1.1 无创血压插补结果

在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，非参数检验等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，在此我们使用的变量有：有创血压（收缩压，舒张压，平均压）数据，年龄，性别，是否接入呼吸机，入住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（收缩压，舒张压，平均压）上的插补结果如下：

表3.1 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3.2 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3.3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，且使用的特征较少，但仍然不可避免的会对数据加入多重共线性的影响。且KNN方法在样本数量大的情况下，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才能得到结果，而随机森林仅仅需要几分钟，且其在4种误差评价指标的表现仅次于KNN模型。MI数值达到0.67，可认为补齐数据与原始数据具有中度非线性相关性，而P值达到0.45，可认为补齐数据与原始数据并无显著差距。基于上述原因，最终选取随机森林作为无创血压插补方法。

3.1.2 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%的两个变量FIO2与temp的部分衍生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衍生变量进行数据补全。在此选取两变量进入ICU前12小时数据的平均值，进入ICU12~24小时数据的平均值共四个变量进行数据插补模型性能的评价，即q\_fio2\_mean, h\_fio2\_mean, q\_temp\_mean, h\_temp\_mean。

表3.4特征qfio2mean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3.5特征hfio2mean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3.6特征qtempmean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3.7特征htempmean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。而随机森林

在四个误差项上的表现明显优于其它四种模型，除在qtempmean特征上的插补结果，随机森林的误差仅不足次优的KNN模型的1/3。其补齐数据与原始数据之间的互信息在各个特征上高于其它模型且数值均在0.98以上，表明了随机森林补齐数据与原始数据之间具有非常强的非线性相关性，原始数据的取值在很大程度上能够确定随机森林补齐数据的取值。而随机森林模型的P值指标更是达到了0.79以上，表明补齐数据在分布上与原始数据并未有显著性差异。这正是我们在进行数据补齐时想实现的结果。综合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。

3.2 Feature Selection Result

由于篇幅限制，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28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

表3.8特征权重部分结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | Features | Relief | Fisher | gini | combinational score | 原假设 | p value | 决策者 |
| 1 | qrespperc25 | 0.31 | 1 | 0.62 | 1.92 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 2 | qrespmean | 0.28 | 0.94 | 0.63 | 1.85 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 3 | icutype2 | 0.55 | 0.69 | 0.56 | 1.81 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 4 | age | 1 | 0.42 | 0.35 | 1.77 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 5 | qrespperc50 | 0.27 | 0.9 | 0.56 | 1.74 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 6 | icutype3 | 0.86 | 0.38 | 0.32 | 1.56 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 7 | qrespperc75 | 0.23 | 0.74 | 0.52 | 1.49 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 8 | urineoutput | 0.07 | 0.32 | 1 | 1.39 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 9 | qrespmin | 0.23 | 0.62 | 0.39 | 1.25 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 10 | hspo2min | 0.14 | 0.55 | 0.56 | 1.25 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 11 | qfio2min | 0.24 | 0.41 | 0.52 | 1.17 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 12 | hspo2stddev | 0.07 | 0.55 | 0.55 | 1.17 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 13 | hrespperc25 | 0.32 | 0.44 | 0.39 | 1.15 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 14 | hspo2range | 0.14 | 0.52 | 0.48 | 1.14 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 15 | qnimeanbp\_max | 0.24 | 0 | 0.89 | 1.14 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 16 | hfio2min | 0.25 | 0.37 | 0.5 | 1.12 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 17 | qnidiasbp\_min | 0.22 | 0.41 | 0.44 | 1.08 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 18 | qfio2perc\_25 | 0.22 | 0.39 | 0.45 | 1.07 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 19 | hgcs | 0.24 | 0.44 | 0.37 | 1.05 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 20 | hrespmean | 0.32 | 0.37 | 0.33 | 1.01 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 21 | hfio2perc\_25 | 0.22 | 0.35 | 0.44 | 1.01 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 22 | hrespperc50 | 0.32 | 0.35 | 0.32 | 0.99 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 23 | hnidiasbp\_min | 0.25 | 0.37 | 0.34 | 0.96 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 24 | qfio2perc\_50 | 0.25 | 0.35 | 0.35 | 0.95 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 25 | qnisysbp\_min | 0.16 | 0.33 | 0.46 | 0.94 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 26 | qnisysbp\_range | 0.19 | 0.43 | 0.29 | 0.92 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 27 | hnimeanbp\_min | 0.25 | 0.32 | 0.31 | 0.88 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 28 | hnisysbp\_min | 0.15 | 0.29 | 0.43 | 0.87 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |

基于八个机器学习模型分别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）计算不同特征子集下的BER值（十折交叉验证的平均值），并求解出最佳和最小特征子集的特征数。所有最小特征子集的特征数从151个大大减少到不超过28个，降幅达81.5%。图3.1描述了在BER与特征数构成的映射中，不同模型下最优子集与最小子集的位置。



Regr XGB



GBM Adb



SVM Log



GBN DT

Figure 3.1. 橙色圆点为最佳特征子集，黄色五角星为最小特征子集一，由传统方法计算得出；紫色五角星为最小特征子集二，由FSFBER方法计算得出。

其中regr 是Random Forest的简称，XGB是XGBoost的简称，GBM是LightGBM的简称，Adb是Adaboost的简称，SVM是Support Vector Machine的简称，log是Logistic Regression简称，BN 是Naive Bayesian的简称，DT是Decision Tree Model的检测。

在选取最小特征子集时，Regr，XGB，GBM，Adb，BN 与DT设置参数为与；SVM与Log设置参数为与.

图3.2 特征子集比例图

图3.2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FSFBER方法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训练集上结果的优劣。

表3.9 FSFBER方法特征子集预测AUC与传统模型特征子集预测AUC的比较

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RF | XGB | LGB | Adaboost | SVM | LR | NB | DT |
| **最小特征子集1 AUC** | 0.894 | 0.921 | 0.930 | 0.818 | 0.848 | 0.842 | 0.791 | 0.818 |
| **最小特征子集2 AUC** | 0.895 | 0.921 | 0.919 | 0.825 | 0.821 | 0.829 | 0.797 | 0.813 |

相比于传统模型，FSFBER方法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模型性能大幅降低的情况。

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

图3.3 新模型特征子集预测结果AUC与传统模型比较比例图

从图3.3可以很直观的看出，8个模型模型性能基本保持不变，均在79%以上，特别地XGB与GBM均在91%以上。基于上述结果，FSFBER方法能够很好地完成寻找最小特征子集这一任务。在特征数量降幅超过80%以上的同时，保证模型性能基本不变。

为了更清楚地表示不同模型的特征子集选择结果，表3.10描述了不同模型对应的最优与FSFBER方法的最小特征集合数量与BER值。

表3.10 BER values and feature numbers for optimal and smallest feature subsets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Feature numbers for OPT\_subset | BER for OPT\_subset | Feature numbers for SMA\_subset | BER for SMA\_subset |
| RF | 28 | 0.196 | 112 | 0.183 |
| XGB | 27 | 0.162 | 129 | 0.156 |
| LGB | 28 | 0.162 | 111 | 0.144 |
| Adaboost | 26 | 0.355 | 104 | 0.350 |
| SVM | 27 | 0.247 | 149 | 0.225 |
| LR | 27 | 0.234 | 149 | 0.202 |
| NB | 27 | 0.277 | 100 | 0.273 |
| DT | 16 | 0.257 | 132 | 0.241 |
| OPT\_subset is short for optimal feature subset, SMA\_subset is short for smallest feature subset | | | | |

下文中提及的最小特征子集均是由FSFBER方法选取的。

3.3 Validation Results

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，Adaboost，SVM，Log，GNB，DT与传统评分系统：Sapsiii，Sofa，apsiii，mews，oasis。使用训练集训练模型，使用测试集对模型性能进行测试。通过调节分类阈值的方法来调节由于训练数据类不平衡造成的结果，以上方法的性能均在表3.11中给出。

表3.11 Identification results of algorithms on test sets under different feature subsets

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subset** | **Methods** | **ACC** | **AUC** | **BER** | **F1\_score** | **MCC** | **TNR** | **TPR** | **Threshold** | **Characteristic quantity** |
| **All** | **regr** | **0.83995843** | **0.91729622** | **0.16825592** | **0.58490566** | **0.52890675** | **0.82041588** | **0.84307229** | **0.2** | **151** |
| **XGB** | **0.886464** | **0.94443426** | **0.12937943** | **0.6726592** | **0.627246** | **0.84877127** | **0.89246988** | **0** | **151** |
| **GBM** | **0.87035594** | **0.9498132** | **0.1172632** | **0.65609924** | **0.61751894** | **0.89981096** | **0.86566265** | **0.07** | **151** |
| **Adaboost** | **0.80306573** | **0.90044469** | **0.18169569** | **0.53948967** | **0.48295033** | **0.83931947** | **0.79728916** | **0.49** | **151** |
| **SVM** | **0.8020265** | **0.88600907** | **0.1926276** | **0.53078818** | **0.46914027** | **0.8147448** | **0.8** | **0.12** | **151** |
| **log** | **0.79709015** | **0.86360034** | **0.22727213** | **0.5003199** | **0.4236985** | **0.73913044** | **0.8063253** | **0.16** | **151** |
| **BN** | **0.7316186** | **0.7975511** | **0.2731697** | **0.4245125** | **0.3324672** | **0.72022684** | **0.73343374** | **0.01** | **151** |
| **DT** | **0.74071187** | **0.84117282** | **0.22658061** | **0.46459228** | **0.39557871** | **0.81852552** | **0.72831325** | **0.15** | **151** |
| **Optimal** | **regr** | **0.82748766** | **0.91696882** | **0.16594991** | **0.57326478** | **0.5200855** | **0.84310019** | **0.825** | **0.19** | **112** |
| **XGB** | **0.8854248** | **0.94535325** | **0.12759811** | **0.672119** | **0.6274187** | **0.85444234** | **0.89036145** | **0** | **129** |
| **GBM** | **0.85632632** | **0.9467471** | **0.1261903** | **0.6320692** | **0.59207815** | **0.89792061** | **0.8496988** | **0.07** | **111** |
| **Adaboost** | **0.75526111** | **0.8429977** | **0.23880589** | **0.46355353** | **0.3859683** | **0.76937618** | **0.75301205** | **0.47** | **104** |
| **SVM** | **0.80046765** | **0.88676464** | **0.19432579** | **0.52825553** | **0.46603709** | **0.81285444** | **0.79849398** | **0.12** | **149** |
| **log** | **0.79631073** | **0.86354454** | **0.22772394** | **0.49936143** | **0.42260954** | **0.73913044** | **0.80542169** | **0.16** | **149** |
| **BN** | **0.7160301** | **0.810867** | **0.2663143** | **0.423219** | **0.3363143** | **0.75803403** | **0.70933735** | **0.03** | **100** |
| **DT** | **0.76981034** | **0.84194775** | **0.24308539** | **0.46882494** | **0.38758698** | **0.73913044** | **0.7746988** | **0.17** | **132** |
| **Smallest** | **regr** | **0.82359054** | **0.9092639** | **0.17138725** | **0.56557902** | **0.51051889** | **0.83553875** | **0.82168675** | **0.19** | **28** |
| **XGB** | **0.8477527** | **0.9325358** | **0.1375168** | **0.6144737** | **0.5708826** | **0.88279773** | **0.84216868** | **0** | **27** |
| **GBM** | **0.84307612** | **0.9325142** | **0.1418168** | **0.60625815** | **0.56154461** | **0.87901701** | **0.8373494** | **0.1** | **28** |
| **Adaboost** | **0.77370746** | **0.83597376** | **0.26386909** | **0.4539185** | **0.36336619** | **0.68431002** | **0.78795181** | **0.47** | **26** |
| **SVM** | **0.79137438** | **0.84304838** | **0.21310469** | **0.50705955** | **0.43728705** | **0.78071834** | **0.79307229** | **0.11** | **27** |
| **log** | **0.75188361** | **0.84223814** | **0.23520168** | **0.46438587** | **0.38906723** | **0.7826087** | **0.74698795** | **0.14** | **27** |
| **BN** | **0.7251234** | **0.8107269** | **0.2697836** | **0.4243743** | **0.334639** | **0.73724008** | **0.72319277** | **0.1** | **27** |
| **DT** | **0.76071707** | **0.8073115** | **0.26265886** | **0.4475105** | **0.35814922** | **0.70510397** | **0.76957831** | **0.17** | **16** |
| **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** | **-** |

从评价指标的角度看，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，能够更准确评估模型整体性能。从这三个指标的角度出发，除去在最小特征子集下的NB模型外，机器学习模型均要显著优于传统评分系统，且从敏感性及特异性两个指标可以看出，传统评分系统Mews出现了较为明显的过拟合情况，而在机器学习模型中并未出现。

明显的，XGB在使用的八种机器学习算法中具有最佳性能。XGB即便在最小特征子集上，相较于传统评分系统中最优的Spasiii的预测结果，ACC和AUC仍能显著提高12.7和11.7个百分点。同样从ACC和AUC出发，RF，XGB，LGB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28仍然可以提供相对高的性能，ACC均高于82％，AUC均高于90％。特别的，XGB在特征量下降82%的情况下，ACC和AUC分别仅下降3.87％和1.19％。

为了更清楚地表示不同特征子集对模型性能的影响，表3.12描述了FSFBER方法的最小特征子集，最优特征子集相较于最小特征子集，特征全集相较于最优特征子集的模型性能的AUC指标变化。

表3.12 不同特征子集下模型AUC指标变化

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RF** | **XGB** | **LGB** | **Adaboost** | **SVM** | **LR** | **NB** | **DT** |
| 最小特征子集上AUC值 | **0.909** | **0.933** | **0.933** | **0.836** | **0.843** | **0.842** | **0.811** | **0.807** |
| 最优相较于最小变化的AUC值(增加的特征数) | **0.008 (84)** | **0.013 (102)** | **0.014 (83)** | **0.007 (78)** | **0.044 (122)** | **0.021 (122)** | **0.000 (73)** | **0.035 (116)** |
| 全集相较于最优变化的AUC值(增加的特征数) | **0.000 (39)** | **-0.001 (22)** | **0.003 (40)** | **0.057 (47)** | **-0.001 (2)** | **0.000 (2)** | **-0.013 (51)** | **-0.001 (19)** |

在最小特征子集上，所有机器学习模型AUC值均能达到0.8以上，最优达0.933。所有模型在最优特征子集上相较于最小特征子集，模型性能均有提升。但八种机器学习模型最小特征子集相较于最优特征子集AUC下降不超过0.05，说明了最小特征子集在幅度达到80%以上的同时，仍保证了模型性能。此外，在特征全集上，除AdaBoost外，其他模型性能提升不明显甚至有部分下降，也在一定程度上表明了特征降维的必要性。

1. **Disscution**

ICU患者的死亡率预测具有十分重要的意义。进行ICU患者死亡率预测可提前对危重患者进行介入，提高危重病人的生存几率，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，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。近年来，对ICU患者死亡率进行预测成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。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，但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ICU患者的生死预测方法使用的数据多是侵入式的，在院前急救情况下数据获取不便，传统评分需要根据实际的医疗环境进行校准，一致性不高，多数评分模型使用的是逻辑回归，变量之间的关系式简单的线性叠加，无法代表真实疾病的病例情况，难以满足临床要求，限制了其应用范围。

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普通监护仪容易获取的各种非侵入性生理参数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，每一条记录包括10种NIPP参数。随后通过数据预处理，对每个时序参数计算18个衍生变量并进行显著性检验，最终确定151个衍生变量。通过ReliefF方法，Fisher准则，GINI指数三种特征权值计算方法从多维度充分挖掘非侵入式时序特征的数据信息，对数据特征进行全面的评分与综合排序，并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）与本文提出的一种新的特征子集选择模型(A feature selection framework based on Bayes Error Rate, FSFBER)确定最佳特征子集与最小特征子集。随后对在最优、最小特征子集与特征全集上的8种机器学习算法与5种传统评分系统进行全面的比较。

我们的分析表明，机器学习算法相比于传统方法具有更加优秀的表现。更高的性能与更宽泛的应用范围：几乎所有机器学习在AUC方面均优于传统方法，尤其是XGBoost和LightGBM算法。在最优子集下，XGBoost算法在ACC和AUC参数方面分别优于预测效率最高的传统方法Spasiii 16.4%和12.9%。此外，本算法所采用生理参数的均为普通监护仪容易获取的各种非侵入性生理参数，可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，相比于需要生化和有创参数支持的传统评分方法更是大大降低了临床使用要求，减少对患者的伤害；降维的优势：本研究建立的基于FSFBER模型的降维方法消除了冗余的特征信息，构建了最小特征子集，在特征数量降幅80%以上的同时，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早期预警算法的临床实现中是非常重要的。这种新颖的机器学习方法最小特征子集的使用更是在保证模型性能的前提下大大降低了计算成本，使其有望在广泛移植于多样化的硬件平台中，得到广泛的临床应用。类不平衡问题的处理：此外，本文所使用的数据集中阳性和隐性两类数据类别数据量之比为16552:2693=6.15:1。虽然有数据类别严重不平衡的情况，但该研究并未在训练集上对数据进行类平衡处理。为了避免使用欠采样与过采样带来的数据量下降与模型过拟合问题，本文使用优化调节分类阈值的方法来对病态的模型进行调校，结果中各模型性能指标的敏感性与特异性均较为平衡，表明该方法能够有效调节病态机器学习模型的过拟合问题。

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。

虽然本文在MIMIC III数据库上取得了较为不错的预测结果，但本研究仍然存在着一些局限性。首先，分类阈值对病态的机器学习模型的调节能力是有限的，为了能够取得整体较优的模型，有时不得不降低模型的学习能力以平衡敏感性与特异性。为了提高模型性能，在考虑数据量与各个模型学习能力关系的基础上，可适当采用双采样与调节分类阈值相结合的方法。其次，由于本研究中的数据集数据多是通过人为设定条件进行筛选，且文中并未使用数据去噪方法，这在一定程度上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，介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。

1. **Conclusion**

本研究提出的机器学习模型的最优预测结果ACC达88.6%，AUC达95.0%，证实了机器学习算法与传统评分系统相比，无需专业医务人员支持，可基于患者无创参数之准确自动生成评分结果，在性能与可操作性上有了很大的提升。在临床实践中，通过应用本工作中介绍的方法，相信可以容易，迅速地识别出死亡率较高的患者，以更早地采取必要的治疗干预措施来提高危重病人的生存几率。希望本研究能鼓励和加快该领域的进一步发展，为ICU患者死亡率预测方面提供有价值的参考和建议指导。

附录

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | Features | Relief | Fisher | gini | combinational score | 原假设 | p value | 决策者 |
| 1 | qrespperc25 | 0.31 | 1 | 0.62 | 1.92 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 2 | qrespmean | 0.28 | 0.94 | 0.63 | 1.85 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 3 | icutype2 | 0.55 | 0.69 | 0.56 | 1.81 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 4 | age | 1 | 0.42 | 0.35 | 1.77 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 5 | qrespperc50 | 0.27 | 0.9 | 0.56 | 1.74 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 6 | icutype3 | 0.86 | 0.38 | 0.32 | 1.56 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 7 | qrespperc75 | 0.23 | 0.74 | 0.52 | 1.49 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 8 | urineoutput | 0.07 | 0.32 | 1 | 1.39 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 9 | qrespmin | 0.23 | 0.62 | 0.39 | 1.25 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 10 | hspo2min | 0.14 | 0.55 | 0.56 | 1.25 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 11 | qfio2min | 0.24 | 0.41 | 0.52 | 1.17 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 12 | hspo2stddev | 0.07 | 0.55 | 0.55 | 1.17 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 13 | hrespperc25 | 0.32 | 0.44 | 0.39 | 1.15 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 14 | hspo2range | 0.14 | 0.52 | 0.48 | 1.14 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 15 | qnimeanbp\_max | 0.24 | 0 | 0.89 | 1.14 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 16 | hfio2min | 0.25 | 0.37 | 0.5 | 1.12 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 17 | qnidiasbp\_min | 0.22 | 0.41 | 0.44 | 1.08 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 18 | qfio2perc\_25 | 0.22 | 0.39 | 0.45 | 1.07 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 19 | hgcs | 0.24 | 0.44 | 0.37 | 1.05 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 20 | hrespmean | 0.32 | 0.37 | 0.33 | 1.01 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 21 | hfio2perc\_25 | 0.22 | 0.35 | 0.44 | 1.01 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 22 | hrespperc50 | 0.32 | 0.35 | 0.32 | 0.99 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 23 | hnidiasbp\_min | 0.25 | 0.37 | 0.34 | 0.96 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 24 | qfio2perc\_50 | 0.25 | 0.35 | 0.35 | 0.95 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 25 | qnisysbp\_min | 0.16 | 0.33 | 0.46 | 0.94 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 26 | qnisysbp\_range | 0.19 | 0.43 | 0.29 | 0.92 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 27 | hnimeanbp\_min | 0.25 | 0.32 | 0.31 | 0.88 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 28 | hnisysbp\_min | 0.15 | 0.29 | 0.43 | 0.87 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 29 | hrespperc75 | 0.31 | 0.27 | 0.29 | 0.87 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 30 | qfio2max | 0.69 | 0.08 | 0.08 | 0.86 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 31 | hrespmin | 0.3 | 0.27 | 0.28 | 0.85 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 32 | hfio2perc\_50 | 0.23 | 0.29 | 0.33 | 0.85 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 33 | bmi\_label1 | 0.24 | 0.33 | 0.27 | 0.85 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 34 | bmi\_label2 | 0.24 | 0.33 | 0.27 | 0.85 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 35 | qrespmax | 0.2 | 0.33 | 0.3 | 0.83 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 36 | hspo2var | 0 | 0.27 | 0.55 | 0.83 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 37 | hfio2max | 0.66 | 0.08 | 0.08 | 0.81 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 38 | qnisysbp\_stddev | 0.16 | 0.39 | 0.26 | 0.81 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 39 | hnidiasbp\_perc25 | 0.24 | 0.29 | 0.28 | 0.81 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 40 | qhrmax | 0.2 | 0.31 | 0.29 | 0.8 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 41 | hfio2mean | 0.26 | 0.24 | 0.3 | 0.8 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 42 | qfio2mean | 0.25 | 0.27 | 0.25 | 0.77 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 43 | icutype4 | 0.66 | 0.05 | 0.04 | 0.75 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 44 | qnidiasbp\_perc25 | 0.26 | 0.22 | 0.26 | 0.74 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 45 | hhrmax | 0.22 | 0.25 | 0.27 | 0.74 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 46 | qhrmean | 0.26 | 0.21 | 0.26 | 0.73 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 47 | hfio2range | 0.66 | 0.03 | 0.03 | 0.72 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 48 | qhrperc75 | 0.2 | 0.23 | 0.27 | 0.7 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 49 | hnidiasbp\_mean | 0.25 | 0.22 | 0.23 | 0.7 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 50 | qfio2perc\_75 | 0.28 | 0.23 | 0.18 | 0.7 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 51 | hhrrange | 0.14 | 0.29 | 0.26 | 0.7 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 52 | qnisysbp\_perc25 | 0.24 | 0.08 | 0.38 | 0.69 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 53 | qnisysbp\_var | 0.06 | 0.37 | 0.26 | 0.69 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 54 | qgcs | 0.46 | 0.1 | 0.12 | 0.68 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 55 | qnidiasbp\_range | 0.1 | 0.32 | 0.25 | 0.68 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 56 | hfio2perc\_75 | 0.29 | 0.19 | 0.19 | 0.67 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 57 | hhrperc75 | 0.23 | 0.18 | 0.25 | 0.66 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 58 | qspo2min | 0.1 | 0.27 | 0.29 | 0.66 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 59 | hnisysbp\_perc25 | 0.21 | 0.12 | 0.33 | 0.66 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 60 | hnidiasbp\_perc50 | 0.24 | 0.2 | 0.21 | 0.65 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 61 | hhrperc50 | 0.29 | 0.15 | 0.21 | 0.65 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 62 | qhrperc50 | 0.23 | 0.19 | 0.24 | 0.65 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 63 | hhrmean | 0.29 | 0.15 | 0.21 | 0.65 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 64 | qnimeanbp\_min | 0.31 | 0.12 | 0.21 | 0.64 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 65 | qspo2stddev | 0.07 | 0.27 | 0.29 | 0.63 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 66 | qnimeanbp\_perc75 | 0.15 | 0.04 | 0.44 | 0.63 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 67 | hnisysbp\_range | 0.15 | 0.27 | 0.2 | 0.63 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 68 | hnimeanbp\_range | 0.2 | 0.22 | 0.2 | 0.62 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 69 | qhrperc25 | 0.27 | 0.15 | 0.2 | 0.62 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 70 | qspo2range | 0.1 | 0.24 | 0.28 | 0.62 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 71 | qspo2perc25 | 0.07 | 0.29 | 0.25 | 0.61 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 72 | qnidiasbp\_perc50 | 0.27 | 0.13 | 0.21 | 0.61 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 73 | icutype5 | 0.6 | 0 | 0 | 0.61 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 74 | qspo2mean | 0.05 | 0.32 | 0.23 | 0.61 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 75 | hnimeanbp\_perc25 | 0.24 | 0.14 | 0.21 | 0.6 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 76 | hnimeanbp\_stddev | 0.15 | 0.22 | 0.22 | 0.59 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 77 | htempperc25 | 0.23 | 0.07 | 0.29 | 0.59 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 78 | qnidiasbp\_stddev | 0.11 | 0.28 | 0.2 | 0.59 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 79 | hnisysbp\_mean | 0.22 | 0.06 | 0.29 | 0.57 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 80 | hnisysbp\_stddev | 0.14 | 0.25 | 0.18 | 0.57 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 81 | qnidiasbp\_mean | 0.26 | 0.11 | 0.2 | 0.57 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 82 | hhrperc25 | 0.27 | 0.11 | 0.19 | 0.57 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 83 | qnisysbp\_mean | 0.25 | 0.02 | 0.29 | 0.56 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 84 | hrespmax | 0.22 | 0.14 | 0.19 | 0.56 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 85 | gender1 | 0.46 | 0.05 | 0.04 | 0.55 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 86 | gender2 | 0.46 | 0.05 | 0.04 | 0.55 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 87 | hnisysbp\_perc50 | 0.23 | 0.05 | 0.26 | 0.55 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 88 | qhrmin | 0.27 | 0.11 | 0.17 | 0.55 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 89 | hhrstddev | 0.09 | 0.24 | 0.21 | 0.54 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 90 | hnidiasbp\_perc75 | 0.24 | 0.14 | 0.15 | 0.53 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 91 | icutype1 | 0.53 | 0 | 0 | 0.53 | 该特征的分布在death类别上相同 | 0.047 | 拒绝原假设 |
| 92 | qnisysbp\_perc50 | 0.23 | 0.02 | 0.27 | 0.52 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 93 | qnimeanbp\_mean | 0.26 | 0.08 | 0.18 | 0.52 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 94 | hnisysbp\_var | 0.05 | 0.3 | 0.18 | 0.52 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 95 | qnimeanbp\_perc25 | 0.25 | 0.09 | 0.17 | 0.51 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 96 | htempmean | 0.23 | 0.03 | 0.25 | 0.5 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 97 | hfio2stddev | 0.45 | 0.03 | 0.02 | 0.5 | 该特征的分布在death类别上相同 | 0.002 | 拒绝原假设 |
| 98 | htempperc50 | 0.23 | 0.04 | 0.24 | 0.5 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 99 | qnimeanbp\_perc50 | 0.25 | 0.08 | 0.17 | 0.49 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 100 | htempmin | 0.09 | 0.1 | 0.29 | 0.48 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 101 | hhrvar | 0.01 | 0.26 | 0.21 | 0.48 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 102 | hspo2perc50 | 0.02 | 0.1 | 0.34 | 0.47 | 该特征的分布在death类别上相同 | 0.030 | 拒绝原假设 |
| 103 | qhrrange | 0.14 | 0.16 | 0.16 | 0.46 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 104 | qtempperc75 | 0.2 | 0.01 | 0.25 | 0.46 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 105 | qtempmean | 0.19 | 0.03 | 0.24 | 0.46 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 106 | qhrstddev | 0.13 | 0.16 | 0.15 | 0.44 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 107 | qnidiasbp\_perc75 | 0.22 | 0.05 | 0.17 | 0.44 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 108 | htempperc75 | 0.21 | 0.01 | 0.21 | 0.43 | 该特征的分布在death类别上相同 | 0.001 | 拒绝原假设 |
| 109 | hnimeanbp\_mean | 0.18 | 0.05 | 0.19 | 0.42 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 110 | qnidiasbp\_var | 0.03 | 0.19 | 0.2 | 0.42 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 111 | vent1 | 0.1 | 0.17 | 0.14 | 0.42 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 112 | vent2 | 0.1 | 0.17 | 0.14 | 0.42 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 113 | qspo2perc50 | 0.05 | 0.22 | 0.15 | 0.42 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 114 | hnisysbp\_perc75 | 0.22 | 0.02 | 0.18 | 0.42 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 115 | qtempmax | 0.19 | 0.01 | 0.22 | 0.41 | 该特征的分布在death类别上相同 | 0.008 | 拒绝原假设 |
| 116 | qspo2var | 0.01 | 0.1 | 0.29 | 0.4 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 117 | qtempperc50 | 0.17 | 0.02 | 0.21 | 0.4 | 该特征的分布在death类别上相同 | 0.001 | 拒绝原假设 |
| 118 | qnisysbp\_perc75 | 0.24 | 0 | 0.15 | 0.4 | 该特征的分布在death类别上相同 | 0.004 | 拒绝原假设 |
| 119 | hhrmin | 0.23 | 0.02 | 0.15 | 0.39 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 120 | hrespstddev | 0.2 | 0 | 0.18 | 0.38 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 121 | hresprange | 0.23 | 0 | 0.14 | 0.38 | 该特征的分布在death类别上相同 | 0.037 | 拒绝原假设 |
| 122 | qnisysbp\_max | 0.22 | 0.05 | 0.09 | 0.37 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 123 | qtempmin | 0.1 | 0.06 | 0.19 | 0.36 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 124 | qhrvar | 0.03 | 0.17 | 0.15 | 0.36 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 125 | hnimeanbp\_var | 0.02 | 0.11 | 0.22 | 0.35 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 126 | hnimeanbp\_perc50 | 0.14 | 0.06 | 0.15 | 0.35 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 127 | hspo2perc75 | 0.01 | 0.05 | 0.28 | 0.34 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 128 | qtempperc25 | 0.15 | 0.02 | 0.16 | 0.34 | 该特征的分布在death类别上相同 | 0.004 | 拒绝原假设 |
| 129 | hnidiasbp\_max | 0.2 | 0.04 | 0.09 | 0.33 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 130 | qtempstddev | 0.12 | 0.08 | 0.14 | 0.33 | 该特征的分布在death类别上相同 | 0.001 | 拒绝原假设 |
| 131 | htemprange | 0.04 | 0.16 | 0.12 | 0.31 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 132 | qtempvar | 0.02 | 0.15 | 0.14 | 0.31 | 该特征的分布在death类别上相同 | 0.001 | 拒绝原假设 |
| 133 | hnimeanbp\_perc75 | 0.15 | 0.01 | 0.16 | 0.31 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 134 | htempstddev | 0.05 | 0.16 | 0.1 | 0.31 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 135 | qnimeanbp\_range | 0.2 | 0.03 | 0.06 | 0.29 | 该特征的分布在death类别上相同 | 0.007 | 拒绝原假设 |
| 136 | qnimeanbp\_stddev | 0.17 | 0.04 | 0.07 | 0.27 | 该特征的分布在death类别上相同 | 0.001 | 拒绝原假设 |
| 137 | qresprange | 0.23 | 0.01 | 0.03 | 0.26 | 该特征的分布在death类别上相同 | 0.001 | 拒绝原假设 |
| 138 | qspo2perc75 | 0.04 | 0.14 | 0.08 | 0.26 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 139 | hrespvar | 0.05 | 0.02 | 0.18 | 0.24 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 140 | hnidiasbp\_range | 0.15 | 0.05 | 0.04 | 0.23 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 141 | qrespstddev | 0.19 | 0.01 | 0.03 | 0.23 | 该特征的分布在death类别上相同 | 0.003 | 拒绝原假设 |
| 142 | qnidiasbp\_max | 0.13 | 0.02 | 0.06 | 0.21 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 143 | hfio2var | 0.17 | 0.02 | 0.02 | 0.21 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 144 | hspo2max | 0 | 0.04 | 0.17 | 0.21 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 145 | htempvar | 0.01 | 0.1 | 0.1 | 0.2 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 146 | hnidiasbp\_stddev | 0.11 | 0.04 | 0.03 | 0.18 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 147 | bmi | 0.03 | 0.01 | 0.1 | 0.15 | 该特征的分布在death类别上相同 | 0.005 | 拒绝原假设 |
| 148 | qnimeanbp\_var | 0.02 | 0.05 | 0.07 | 0.15 | 该特征的分布在death类别上相同 | 0.001 | 拒绝原假设 |
| 149 | qspo2max | 0.01 | 0.05 | 0.05 | 0.1 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 150 | qrespvar | 0.05 | 0.01 | 0.03 | 0.09 | 该特征的分布在death类别上相同 | 0.003 | 拒绝原假设 |
| 151 | hnidiasbp\_var | 0.03 | 0.03 | 0.03 | 0.09 | 该特征的分布在death类别上相同 | 0.000 | 拒绝原假设 |
| 152 | qfio2var | - | - | - | - | 该特征的分布在death类别上相同 | 0.152 | 保留原假设 |
| 153 | qfio2stddev | - | - | - | - | 该特征的分布在death类别上相同 | 0.152 | 保留原假设 |
| 154 | qfio2range | - | - | - | - | 该特征的分布在death类别上相同 | 0.084 | 保留原假设 |
| 155 | qtemprange | - | - | - | - | 该特征的分布在death类别上相同 | 0.490 | 保留原假设 |
| 156 | htempmax | - | - | - | - | 该特征的分布在death类别上相同 | 0.403 | 保留原假设 |
| 157 | hspo2mean | - | - | - | - | 该特征的分布在death类别上相同 | 0.056 | 保留原假设 |
| 158 | hspo2perc25 | - | - | - | - | 该特征的分布在death类别上相同 | 0.181 | 保留原假设 |
| 159 | hnisysbp\_max | - | - | - | - | 该特征的分布在death类别上相同 | 0.612 | 保留原假设 |
| 160 | hnimeanbp | - | - | - | - | 该特征的分布在death类别上相同 | 0.070 | 保留原假设 |