

毕业设计（论文）

**基于机器学习的脑卒中预测**

姓 名 袁留威

学 号 190440227

学 院 数理科学与工程学院

专 业 数据科学与大数据技术

指导教师 吕艳芬

二〇二三 年 六 月 十 日

学位论文原创性声明

本人所提交的学位论文《基于机器学习的脑卒中预测》，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

论文作者（签名）： 指导教师确认（签名）：

年 月 日 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解河北工程大学有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权河北工程大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

保密的学位论文在\_\_\_\_\_\_\_年解密后适用本授权书。

论文作者（签名）： 指导教师（签名）：

年 月 日 年 月 日

摘 要

脑卒中发病率较高，不同类型的脑卒中，其治疗方法不同。此外，脑卒中的诊断受医生主观因素的影响，为此本文使用多种机器学习模型来预测患者患上脑卒中的概率，以帮助医生评估和决策。

本文选取了kaggle网站上的脑卒中公开数据，共5110条数据，因数据量较少，本文采用生成对抗网络方法，利用真实的脑卒中样本生成更多的伪样本。先对脑卒中数据进行数据预处理，包括填补缺失值、处理异常值、数据规约，应用SMOTE方法解决脑卒中数据不平衡问题。随后，利用随机森林、XGBoost、CatBoost等三种机器学习模型对脑卒中数据进行预测，将10个特征作为脑卒中预测模型输入。实验结果表明，随机森林模型最优，其预测结果的F1-Score为0.978，AUC值为0.985。为了进一步提高脑卒中预测的准确率，对模型进行参数调优，模型精度有所提高。此时，在测试集上的准确率为0.965，F1-Score为0.981，AUC值为0.988。

关键词：脑卒中；生成对抗网络；随机森林；XGBoost；CatBoost

ABSTRACT

The incidence of stroke is high, and different types of stroke are treated differently. In addition, the diagnosis of stroke is influenced by the subjective factors of doctors, so this article uses a variety of machine learning models to predict the probability of a patient developing a stroke to help doctors evaluate and make decisions.

This paper selects the stroke public data on the Kaggle website, a total of 5110 data, because the amount of data is small, this paper adopts the generative adversarial network method to generate more pseudo samples using real stroke samples. First, the stroke data is preprocessed, including data cleaning, data transformation, and data protocol, and the SMOTE method is applied to solve the problem of stroke data imbalance. Subsequently, three machine learning models such as random forest, XGBoost, and CatBoost were used to predict stroke data, and 10 features were determined as input to the stroke prediction model. The experimental results show that the random forest model is optimal, and its prediction result has an F1-Score of 0.978 and an AUC value of 0.985. In order to further improve the accuracy of stroke prediction, the model was tuned and the model accuracy was improved. At this point, the accuracy on the test set is 0.965, the F1-Score is 0.981, and the AUC value is 0.988.

**Key words:** Stroke；CTGAN；Random forest；XGBoost；CatBoost

目 录

[第1章 绪 论 1](#_Toc24801)

[1.1 研究背景及意义 1](#_Toc27939)

[1.2 国内外研究现状 1](#_Toc7017)

[1.3 研究内容 2](#_Toc13605)

[1.4 章节安排 3](#_Toc24888)

[第2章 脑卒中预测模型相关理论基础 4](#_Toc31829)

[2.1 生成对抗网络 4](#_Toc27504)

[2.2 随机森林算法 5](#_Toc16762)

[2.3 XGBoost算法 7](#_Toc26425)

[2.4 Cat Boost算法 8](#_Toc2197)

[2.5 分类模型评价方法 10](#_Toc3621)

[第3章 基于随机森林构建脑卒中预测模型 12](#_Toc7228)

[3.1 预测模型的构建 12](#_Toc17042)

[3.1.1 数据预处理 12](#_Toc5867)

[3.1.2 SMTOE解决数据集不平衡问题 14](#_Toc22594)

[3.1.3 随机森林模型的构建 14](#_Toc23497)

[3.2 优化随机森林预测模型 15](#_Toc12926)

[第4章 基于XGBoost构建脑卒中预测模型 17](#_Toc8885)

[4.1 XGBoost预测模型的构建 17](#_Toc27772)

[4.2 优化XGBoost预测模型 18](#_Toc6550)

[第5章 基于Cat Boost构建脑卒中预测模型 19](#_Toc9705)

[5.1 CatBoost预测模型的构建 19](#_Toc27966)

[5.2 优化CatBoost预测模型 20](#_Toc7360)

[5.3 所有模型结果对比分析 20](#_Toc24168)

[结论 22](#_Toc18133)

[参考文献 23](#_Toc27538)

[致 谢 24](#_Toc28861)

[附录A 25](#_Toc7879)

# 第1章 绪 论

## 1.1 研究背景及意义

"脑卒中"(cerebral stroke)也称为中风或脑血管意外(CVA),这是一种急性脑血管疾病。因为人脑中的毛细血管突然断裂或阻塞,血流无法进入脑部,造成了脑组织严重受损,所以该疾病可以分为缺血性中风和出血性中风**[1]**。早在十几年前,中风就是全世界国民第一位的自杀因素,也是六十岁以上患者身体伤残的第一因素,严重威胁了人们的身体健康。

据统计，在中国，脑卒中的患病率和致死率持续上升。每年发生脑卒中事件的人数高达230万人以上，是全球最高的国家之一。由于中国的人口众多，并且中国人的饮食习惯不良，走路不够规律，容易导致高血压、高血脂、动脉硬化等疾病，这些因素是中国脑卒中高发的主要原因。同时，医疗资源的不平衡和缺乏也导致一些病人错过最佳治疗时机，延误了病情的治疗。因此，早期发现脑卒中的症状至关重要，症状被发现的时间越晚，治疗效果就会越差，康复时间就越长。

随着科技的不断发展，机器学习在医疗领域的应用越来越广泛，可以帮助处理脑卒中问题。医生可以利用机器学习模型分析出大量的脑部图像数据和病人的其他相关信息，帮助他们做出更精准的诊断以及预测病情的发展状况。此外，机器学习还可以用于开发智能辅助工具，帮助医生分析症状和制定治疗计划。虽然机器学习在医疗领域的应用还面临着一些挑战，但是随着技术的不断进步和应用场景的拓展，相信它一定会为我们带来更多的帮助。

本文旨在利用机器学习模型预测脑卒中，以达到提高脑卒中的预测能力的效果。通过机器学习模型的分析和诊断，可以更加准确地判断患者的脑卒中类型和程度，从而为医生治疗患者提供更加精准和个性化的建议。此外，机器学习还可以通过对大量的医学数据进行深度分析和挖掘，发现更深入的脑卒中的机制和治疗方法。因此，利用机器学习处理脑卒中具有广泛的应用前景和深远的社会意义。

## 1.2 国内外研究现状

心脑血管疾病已经成为全球致死率前列的疾病，其中脑卒中也成为很多国家重点研究的对象，对于脑卒中预测模型的研究是脑卒中疾病研究的重要分支，随着互联网信息的迅速发展，在临床上已经积累了大量复杂的医学数据，而机器学习拥有强大的数据处理能力，尤其是可以为疾病预测提供有价值的研究工具**[2]**。

在国内，一些研究团队尝试运用机器学习算法来预测脑卒中的风险和检测病变，并取得了一些初步的成果。例如，南昌大学的刘建模等人通过研究急性缺血性脑卒中患者诊疗数据，构建了基于机器学习的急性缺血性脑卒中感染预测模型，为临床治疗早期干预提供支持**[3]**。北京航空航天大学的常文兵等人通过研究脑卒中医疗数据，构建了基于支持向量机的脑卒中复发预测模型**[4]**。宋伟才等人通过构建人工神经网络预测模型，分析得出中风与年龄、季节和职业之间有着密切关系**[5]**。清华大学和北京大学等机构合作研究发现，基于计算机视觉以及深度学习的方法可以有效地自动诊断脑卒中。此外，南开大学医学院的研究结果表明，利用机器学习模型可以提高对脑卒中患者的预测准确性。

在国外，也有一些研究团队利用机器学习算法，从各种医学图像和数据中发现脑卒中的病变模式，评估脑卒中的严重程度，以及预测脑卒中的发生和恢复情况。例如，美国斯坦福大学的研究团队已经开发了一个名为Hemispheres的AI系统，该系统可以快速地检测脑部血管卒中造成的损伤部位。SN Min等人结合脑卒中和健康老年人的计算概率，通过构建贝叶斯预测模型进行脑卒中预测**[6]**。此外，Chun Matthew等人比较了Cox模型、机器学习模型以及两种方法结合的集成模型，分析出了中国成年人的前瞻性研究中的中风风险情况**[7]**。Ype de Jong系统评价和外部评估模型对透析患者缺血性卒中的预测性能,研究设计系统地搜索和选择了缺血性中风模型，采用PROBAST进行评估偏倚风险**[8]**。同时，一些公司和组织也在开发了基于机器学习的脑卒中诊断和治疗辅助系统，以帮助医生更好地治疗脑卒中患者。

基于机器学习的脑卒中预测具有较好的临床应用前景，但目前仍存在部分问题。首先，如今使用的机器学习算法与深度学习算法有着一定的联系，而深度学习算法有其自身的局限性，如参数不能自主优化，容易受到噪声影响。此外，在特征提取时往往只采用单一特征。因此，针对目前机器学习算法在脑卒中预测方面存在的问题，应结合多种机器学习算法构建深度学习模型。其次，现有研究均使用有监督学习进行预测，而无监督学习还处于起步阶段。因此，未来可考虑使用无监督学习方法进行机器学习。最后，机器学习的应用需考虑其可解释性。未来应考虑将机器学习与人工神经网络相结合来构建脑卒中预测模型。另外，可进一步探索新的生物标志物等以提高预测模型的预测性能。

## 1.3 研究内容

首先分析了关于脑卒中的国内外研究现状，获取脑卒中公开数据。因数据量较少，本文使用生成对抗网络方法，利用真实的脑卒中样本生成更多的伪样本。对脑卒中数据进行数据预处理，包括填补缺失值、处理异常值、数据规约，应用SMOTE方法解决脑卒中数据不平衡问题。

然后，本文使用随机森林算法构建脑卒中预测模型，将10个特征作为随机森林脑卒中预测模型输入，通过调整决策树个数和树的最大深度等参数，提高预测准确率，研究结果表明模型精度虽高，但存在过拟合问题，所以又使用了10折交叉验证以防止过拟合。为了与随机森林模型对比并找到最优模型，又构建了XGBoost模型对脑卒中数据预测，经过调整参数之后，模型精度有明显提高，但仍有一定的过拟合问题。因此，又构建了CatBoost模型，该模型可以防止模型过拟合，并且有助于提高算法的准确性。

结合准确率、精确率和F1等评估指标，随机森林模型最优，可用于脑卒中预测。

## 1.4 章节安排

第一章，绪论。简要的介绍了脑卒中研究的背景，可以通过机器学习模型的分析和诊断，更加准确地判断患者的脑卒中类型和程度，从而为治疗和康复提供更加精准和个性化的建议。最后，简述了基于机器学习的脑卒中预测模型的国内外研究现状。

第二章，脑卒中预测模型的相关理论基础。分别介绍了生成对抗网络、随机森林、XGBoost以及CatBoost的基本原理和模型的评价指标。

第三章，构建基于随机森林的脑卒中预测模型。首先对脑卒中数据进行包括填补缺失值、处理异常值、数据规约的预处理，然后，应用SMOTE方法解决脑卒中数据不平衡问题。最后，将10个特征作为随机森林脑卒中预测模型输入，调整了决策树个数和树的最大深度等参数，并对实验结果进行了分析。

第四章，构建基于XGBoost的脑卒中预测模型。首先对脑卒中数据进行预处理。然后，构建了基于XGBoost的脑卒中预测模型，经过调优参数提高模型精度。最后，对实验结果进行了分析。

第五章，构建基于CatBoost的脑卒中预测模型。首先对脑卒中数据进行预处理，然后，构建了基于CatBoost的脑卒中预测模型，调整参数并分析优化后的模型。最后，结合准确率、精确率和F-score等评估指标，对三种模型进行了对比分析。

# 第2章 脑卒中预测模型相关理论基础

在临床实践中，医生常常会对脑卒中患者的病史、身体检查和影像学检查进行评估，并从中筛选出与脑卒中发病相关的因素。这些因素通常包括年龄、性别、种族、体重指数(BMI)、糖尿病、高血压、心房颤动、房颤、吸烟史及饮酒史等。目前，临床上常用的预测脑卒中方法分为常规方法和机器学习方法，常规方法包括体格检查、实验室检查和影像学检查等。机器学习方法包括随机森林(Random forest)和支持向量机(Support Vector Machines, SVM)，以及XGBoost和决策树(Decision Tree)等算法。此外，有研究发现，机器学习方法在预测脑卒中中的应用价值可能超过其他预测方法。因此，本文将展开基于机器学习的脑卒中预测的研究。

## 2.1 生成对抗网络

本文采用的是条件表格生成对抗网络(Conditional Tabular Generative Adversarial Network)，是一种基于生成对抗网络(Generate adversarial networks)的方法，用于对表格数据分布建模并从分布中采样。CTGAN提出了一种面向模型的标准化方法，该方法能处理非高斯性和多模态的数据。针对非均衡离散序列，提出一种基于样本学习的有条件生成方法。利用完全连通的网络以及一些先进的技术来进行高品质的建模。通过在生成器和判别器中设置条件，可以生成符合特定条件的结构化数据。CTGAN还使用了一种名为“成分分解”的技术来处理输入数据的多个列之间的相关性，从而更好地模拟实际数据分布。

由图2.1可知，生成对抗网络(GAN)通过生成器网络和判别器网络之间互相竞争生成新的样本，生成器希望得到类似于实际数据分布的新样本,而判定器则希望识别生成器所产生的新样本和实际结果**[9]**。生成器网络进一步改善了生产数据的效率,并尽可能贴近于实际信息的状态。

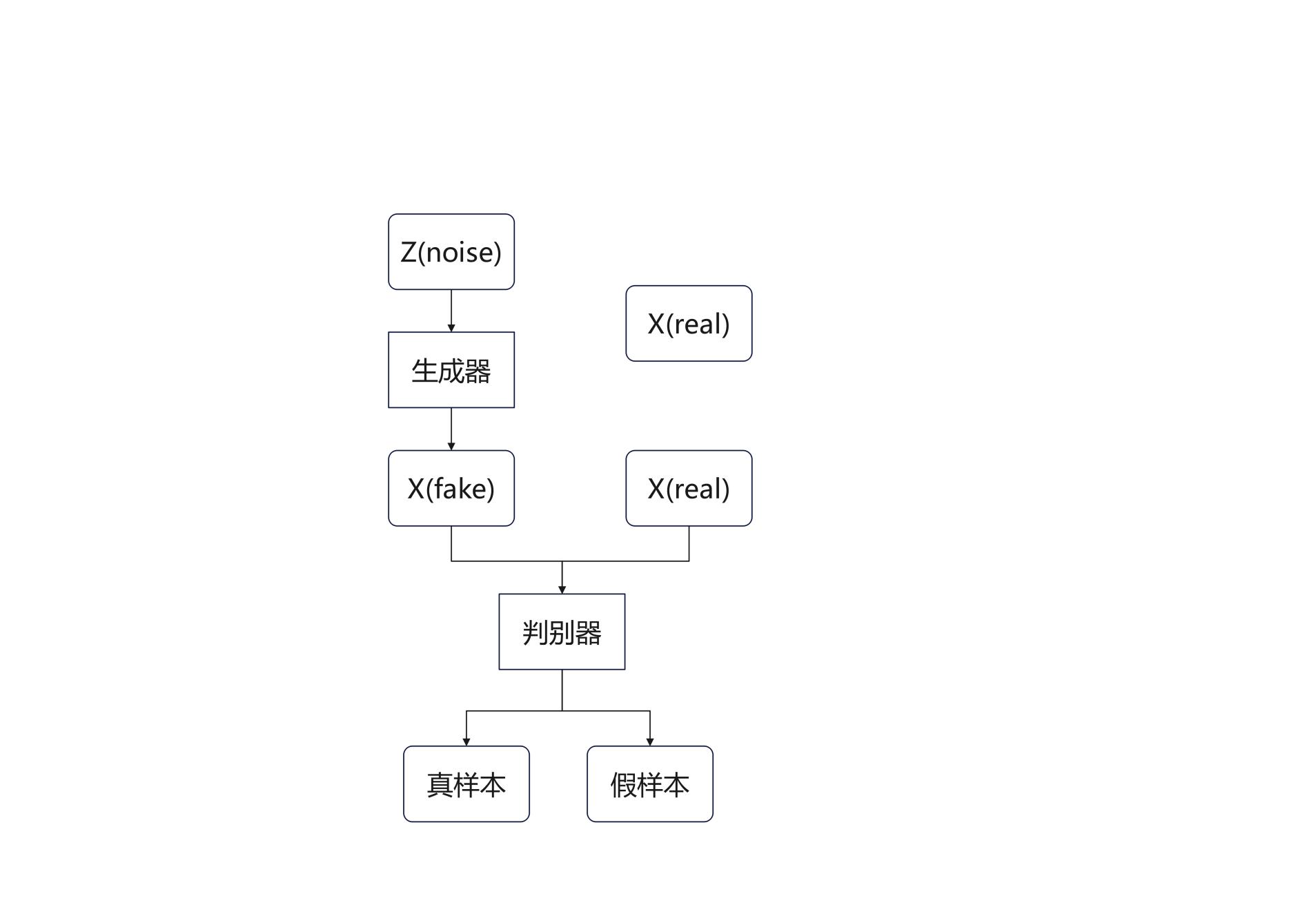


图2.1 GAN模型说明

## 2.2 随机森林算法

随机森林是利用集成学习的思想把多棵树整合的一个算法,也可用来分析和回归问题。随机森林算法通过构建多个决策树来形成一个“森林”，每棵树都是以随机方式从样本中选择的一部分特征进行训练的。随机森林可以通过组合多个决策树的预测结果来获得较高的准确度，并且通常具有很好的泛化能力。它适用于高维数据和非线性关系的建模，并且可以通过调整其超参数来进行优化。

由图2.2可知，随机森林算法的具体流程是：先用N表示培训样本的数目，用M表示特征数目。然后，对于在决策树的某个节点处确定一个决策的结论，在此m应当远小于 M。对 N个训练样本进行反复取样，取样 N次，形成一个训练集合(即Bootstrap样本)，利用没有取样的样本来进行预测，并评价其误差。最后，针对每一结点，随机选取 m个特征点，并根据其特征点，在决策树中对每一结点进行判定。在此基础上，求出 m个特征的最优分割方法。在没有修剪的情况下，所有的树木都会完全生长，这是在建立一个标准的树形分类器，建立完成之后就能使用该分类器**[10]**。

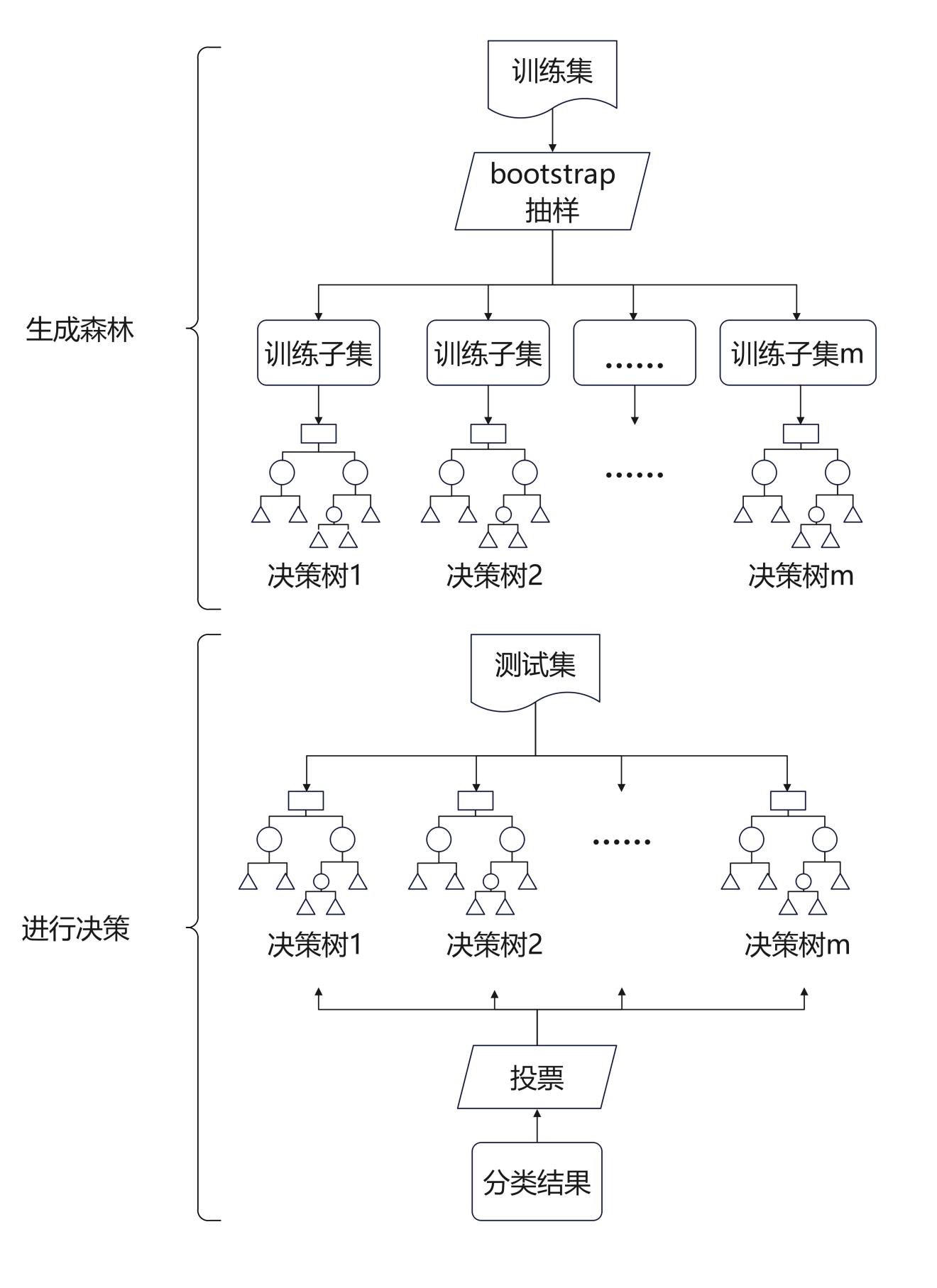


图2.2 随机森林流程图

## 2.3 XGBoost算法

XGBoost(eXtreme Gradient Boosting)是一种高效的开源机器学习算法，常用于解决分类、回归和排序等问题**[11]**。它基于梯度提升决策树(Gradient Boosting Decision Tree)算法，使用决策树作为基础分类器，可以学习多个弱分类器的线性组合来提高整体性能。

Xgboost将使用额外的训练。也就是说，需要确定第一棵树是否是最好的。首先，找出t-1树是否最优。以此类推，我们得到了最好的树。目标函数见公式2-1。

|  |  |  |
| --- | --- | --- |
|  |  | (2-1) |

从目标函数出发，通过对叶片结点进行权重分析，得到了叶片结点划分后所获得的信息增量，并给出了叶片结点的重要程度。xgboost训练过程见表2.1，在目前的决策树构建过程中，第一步是以贪心算法为基础，对其进行划分，并对其进行计算，从而对其所使用的特征进行选择。在选择好特征之后，需要确定分左右子树的条件。为了提高算法效率，使用“加权分位法”，计算分裂点（这里由”特征值重要性排序函数”得出分裂点）。并且对应叶子节点的权值就由上述的“每个叶子节点应该赋予的权值”给出。不断进行上述算法，直至所有特征都被使用或者已经达到限定的层数，则完整的决策树构建完成。

表2.1 xgboost训练过程

|  |
| --- |
| Alogrithm1:Exact Greedy Algorithm for Split Finding |
| Input:*I*,instance set of current node  Input:*d*,feature dimension  *gain*  0  *G*  for k = 1 to m do    for *j in sorted(I,)* do        end  end  Output:Split with max score |

## 2.4 Cat Boost算法

CatBoost, XGBoost, LightGBM是目前最常用的三种基于GBDT的梯度增长方法，它们均是对 GBDT的一种改良。相比XGBoost算法，CatBoost在处理类型变量和缺失值方面更具优势，并且其准确率更高。

Cat Boost算法的机器学习参数少，支持类别变量且精度高，在处理类别数据方面具有显著优势，与其他算法相比，准确性更好，并且泛化能力更高。在脑卒中的预测过程中，可能会出现预测偏移和梯度偏差，以至于出现过拟合问题，而Cat Boost恰好能够有效解决该问题。此外，Cat Boost算法减少了对广泛的超参数调整的需求，具有很高的鲁棒性**[12]**。

CatBoost利用OrderedTS建立了OrderedTS模型，通过排序提升的方式对其进行了改进，排序算法流程见表2.2。

表2.2 排序提升算法流程

|  |
| --- |
| Alogrithm1:Ordered boosting |
| Input:,*I*;  andom permutation of [1,n];  0 for i = 1..n;  for t 1 to I do  for i 1 to n do  ;  for i 1 to n do    ;  *)*;  *;*  return |

在训练数据中,进行排序提升后会出现一次随机排序,而随机排序是对后续的模拟训练,也是在排序中的前几个数据在使用第一个模型前就完成的训练。在迭代的过程中,通过对第一个样本的估计可以获得该样本的残留误差估计值。

对称树被CatBoost用作基本学习器，此处的对称指的是在树的同一层，它的分解标准均是一样的。但是，这样的方法不仅会带来大量的存储空间，还会带来大量的时间复杂性，而且在实际应用中并不是很好。为此，CatBoost提出了一种基于梯度的决策树分类提升方法。

CatBoost提供了两种提升方法，一种是Ordered模式，而另一种是Plain模式。Plain模式是一种常规的提升方法，即它是以标准的 GBDT算法为基础进行分类TS操作的。然而Ordered模式，则是以提升排序水平进行改进的**[13]**。

完全Ordered模式是这样描述的：CatBoost在一个训练集中生成一个独立的随机序列，用它来定义并评价一个分解后的树状结构，并计算出所生成的叶子结点的数值。对称树具有平衡性好、不易过度拟合等优点，可极大地缩短试验周期。CatBoost构造树的算法流程见表2.3。

表2.3 CatBoost构建树的算法流程

|  |
| --- |
| Alogrithm2:Building a tree in Cat Boost |
| Input:  *grad CalcGradient(L,M,y)*;  *r randam(1,s)*;  if *Mode = Plain* then  G ;  if *Mode = Ordered* then  G ;  T empty tree;  foreach *step of top-down procedure* do  foreach *candidate split c* do  add split c to T;  if *Mode = Plain* then  for  *p:* for  if *Mode = Ordered* then  for  *p:*  *for i =1..n;*  *Loss(*  *T*  if *Mode = Plain* then  for  *p:* for = 1..s,i=1..n;  if *Mode = Ordered* then  for  *p:* for = 1..s,i=1..n;  i = 1..n,;  return T,M |

Cat Boost主要创新点是，采用了类型特征的 Ordered Target Statistics数值编码方法、基于贪婪的特征合成方法和Ordered Boosting方法。

相比于XGBoost，Cat Boost算法模型精度更好，且一般无需调参即可获取很好的结果。

## 2.5 分类模型评价方法

在监督学习中，混淆矩阵是一种用于衡量预测性能的工具，可用于比较分类结果和实际测得值。在非监督的情况下，混淆矩阵通常被称作匹配矩阵。在影像准确度评估中，利用混淆矩阵来反映影像的准确度。在测试集进入模型之后，我们可以使用混淆矩阵来评估模型的表现，见表2.4。

表2.4 混淆矩阵

|  |  |  |
| --- | --- | --- |
|  | 实际正类 | 实际负类 |
| 预测正类  预测负类 | TP  FP | FN  TN |

准确率(Accuracy, ACC)是以样品总量为基础对样品进行了正确排列的样品比例，其计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-2) |

精确率(Precision,PPV)用于预测准确的传统数据和预期的传统数据之间的关系，精确率计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-3) |

灵敏度(Sensitivity)，又称召回率，它是指对精确的数据和真实数据的比的预测，灵敏度越高，模型越好，灵敏度计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-4) |

特异度(Specificity)，即真阴性率，它是指所预测的负面类别与真实负面类别的比率，特异度计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-5) |

F1值(F1-score)是一个衡量一个类别划分问题的指标。在某些关于各种问题的竞赛中，F1值常常被用作最后比赛结果的评价方式。将准确度和调用速度调整为平均最大值1和最小值0。F1-score可以通过公式2-6计算。

|  |  |  |
| --- | --- | --- |
|  |  | (2-6) |

ROC(Receiver Operating Characteristic)曲线,即接收者操作特征曲线，使用实际阳性率(TPR)和假阳性率(FPR)作为横轴，TPR作为纵轴。ROC曲线下面的区域，一般为0.5到1.0。AUC值越接近1，模型评估的准确性就越高。医学数据的预测需要更高的AUC值。

# 第3章 基于随机森林构建脑卒中预测模型

## 3.1 预测模型的构建

数据集的数据是公开脑卒中数据集，数据集含有5110个样本，虽然可以基本满足训练模型，但由于目前的数据量较少，所以本文使用了生成对抗网络(CTGAN)方法，以实际的脑卒中样本为基础，产生更多的伪样本，将数据集扩充至10110个样本。为了检验样本的可用性本文使用了Synthetic Data Vault(SDV)库中用于评估合成数据质量的模块,该模块提供了几种常见的质量评估指标，可以用于比较合成数据与原始数据之间的相似性。经检验生成的样本数据的总体质量得分可到94%，可用于训练和测试模型。

### 3.1.1 数据预处理

数据集有12个变量，分别为ID、性别、年龄、高血压、心脏病、婚姻情况、工作类别、居住所在地、平均血糖水平、体重指数、吸烟状况、中风**[14]**，见表3.1。其中，ID是为了保护病人隐私，对患者是否中风没有影响，在后续构建模型中会进行删除。其中，中风属于模型要预测的类别。

表3.1 脑卒中数据的特征展示

|  |  |
| --- | --- |
| 特征 | 特征字段名 |
| 性别 | Gender |
| 年龄 | Age |
| 高血压 | Hypertension |
| 心脏病 | Heart\_disease |
| 婚姻状况 | Ever\_married |
| 工作类型 | Work\_type |
| 居住地 | residence |
| 平均血糖水平 | Avg\_glucose\_level |
| 体重指数 | BMI |
| 吸烟状况 | Smoking\_status |
| 中风 | stroke |

首先，将数据集中的缺失值进行填补，并去除异常值以达到数据清洗的目的。本文所使用的数据集中体重指数有少量数据缺失，采用均值填充缺失值。对于性别这一变量，数据中有三类，分别为male、female、other，在现实生活中人分为男性和女性，一般不存在other，故将other作为异常值删除。

脑卒中数据集的数据格式、数据标识不尽相同，甚至可能存在错误的数据，因此，需要将数据的格式统一。为了将具有差异的脑卒中数据标准化，本文将性别、婚姻状况、居住地、吸烟状况和中风等信息转换为统一格式。可以根据医学标准将平均血糖水平、体重指数、年龄等连续值变量分为多个类别，以便于统计和分析，然后使用数字表示其类别**[15]**。规范后的数据见表3.2。

表3.2 脑卒中数据预处理

|  |  |  |  |
| --- | --- | --- | --- |
| 特征 | 特征字段名 | 种类划分 | 数据变换 |
| 性别 | Gender | Male  Female | 0  1 |
| 高血压 | Hypertension | No  Yes | 0  1 |
| 心脏病 | Heart\_disease | No  Yes | 0  1 |
| 婚姻状况 | Ever\_married | No  Yes | 0  1 |
| 工作类型 | Work\_type | Govt\_job  Never\_worked  Private  Self\_employed  children | 0  1  2  3  4 |
| 居住地 | residence | rural  urban | 0  1 |
| 平均血糖水平 | Avg\_glucose\_level | X<=100  100<X<=125  X>125 | 0  1  2 |
| 吸烟状况 | Smoking\_status | Unknown  Formerly smoked  Never smoked  smokes | 0  1  2  3 |
| 中风 | stroke | No  Yes | 0  1 |

### 3.1.2 SMTOE解决数据集不平衡问题

在医疗领域中，常常会出现数据不平衡的情况，脑卒中数据集也存在这类问题。为了解决此类问题，出现了许多处理数据不平衡问题的方法，常见的处理方法有过采样和欠采样。其中过采样中的SMOTE、Borderline SMOTE和ADASYN是实现简单且常见的处理方法**[16]**。由图3.1知，脑卒中数据中正常的人数为9446，中风的人数为664，数据集存在不平衡问题。故本文采用SMOTE算法处理数据集不平衡的问题。

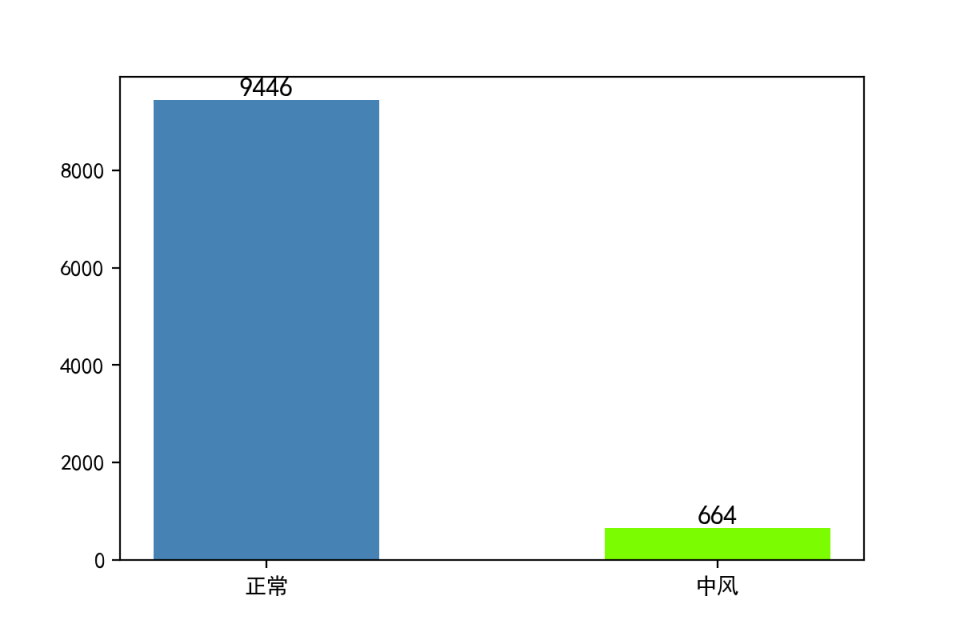


图3.1 中风人数与正常人数对比

SMOTE算法主要是先对小型化样本进行分析，然后综合出新的小型化样本，从而使小型化的样本数目达到一定程度。它需要构造输入和输出数据集，输入不平衡数据集D，近邻数K，采样率N。输出平衡数据集**[16]**。对于每一个少数样本D，通过欧式距离计算得到少数类样本中的所有样本，得到近邻数K。采用输入的采样率N，针对每个少数样本D，从K中任意选取作为近邻数。计算少数样本D与的差值，将差值与随机数(0,1)相乘，再将D与之相加，得到平衡数据集。计算公式见3-1。

|  |  |  |
| --- | --- | --- |
|  |  | (3-1) |

### 3.1.3 随机森林模型的构建

在经过数据预处理之后，将脑卒中数据集按照8：2的比例划分为训练集和和测试集，其中训练集8088条，测试集2022条。通过sklearn库调用随机森林模型，新建模型对象randomforest=RandomForestClassifier()。首先通过fit方法对训练集合X\_train,Y\_train进行模拟训练,随后再对测试集合X\_test进行预测,最终获得模型的得分和预测结果。

随机森林方法对脑卒中数据预测的结果见表3.3，其中，随机森林模型的F1值为0.978，精确率为0.985。由图3.2可知，随机森林模型的AUC值为0.98，综合这几项指标，该模型的预测效果不错。然而，该模型虽然在训练集上的精度为100%，但是在测试集上的精度只有95%。在训练集上的性能很不错，但在测试集上的性能却低于训练集。如果出现这种情况，通常是产生了过拟合现象。在这种情况下，模型可能过度适应了训练数据的噪声和特定特征，而造成了在测试集上表现不佳的问题。因此，需要进一步优化模型，可以增加样本数、减少特征数或者使用其他正则化手段来降低模型的复杂度。

表3.3 随机森林模型结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 混淆矩阵 | 测试集准确率 | 训练集准确率 | F1 | 精确率 |
| [1882 26]  [141 1730] | 0.956 | 1.0 | 0.978 | 0.985 |

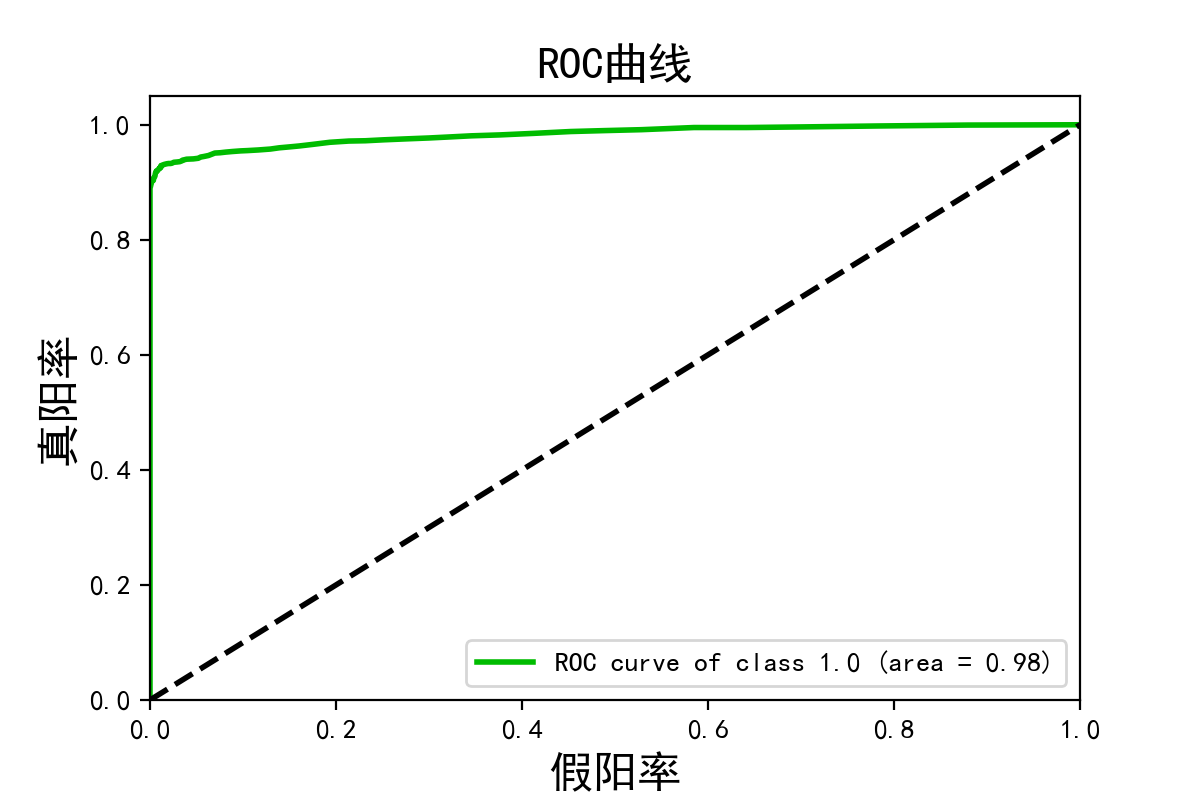


图3.2 随机森林模型ROC曲线

## 3.2 优化随机森林预测模型

过拟合是机器学习中的一个错误，它会降低模型的性能，但是，我们可以使用多种方式防止过拟合。防止过拟合的方法有很多，例如，可以使用更多数据进行训练、特征选择、交叉验证、数据加强等。本文利用交叉验证方法用于防止过拟合。

交叉验证是一种通过“交叉”（将原始数据分割成多个数据组合）来“验证”模型的方法。K折交叉检验也是在限定原始数据前提下,最普遍的泛化错误统计技术之一。它把数据集中分割为K个子集,然后每次都通过其中K负一的子集来训练建模,把剩下的子集都用作数据集中以测试建模功能。再通过反复上述步骤K次,每个数据集中都不同,最后再把K个结果的平均数用作泛化误差的计算**[17]**。这种方法简单易行，并且可以充分利用有限数据进行模型性能评估。

对于K值的选择，一般需大于2，通过手动调整K的值，10折交叉验证效果较为明显。由表3.4可知，随机森林模型的F1为0.981，精确率为0.988。由图3.3可知AUC值也没有明显变化，仍为0.98。测试集准确率有所提高，过拟合程度减少，可用于脑卒中预测。

表3.4 优化后的随机森林模型结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 混淆矩阵 | 测试集准确率 | 训练集准确率 | F1 | 精确率 |
| [1888 20]  [142 1729] | 0.965 | 1.0 | 0.981 | 0.988 |

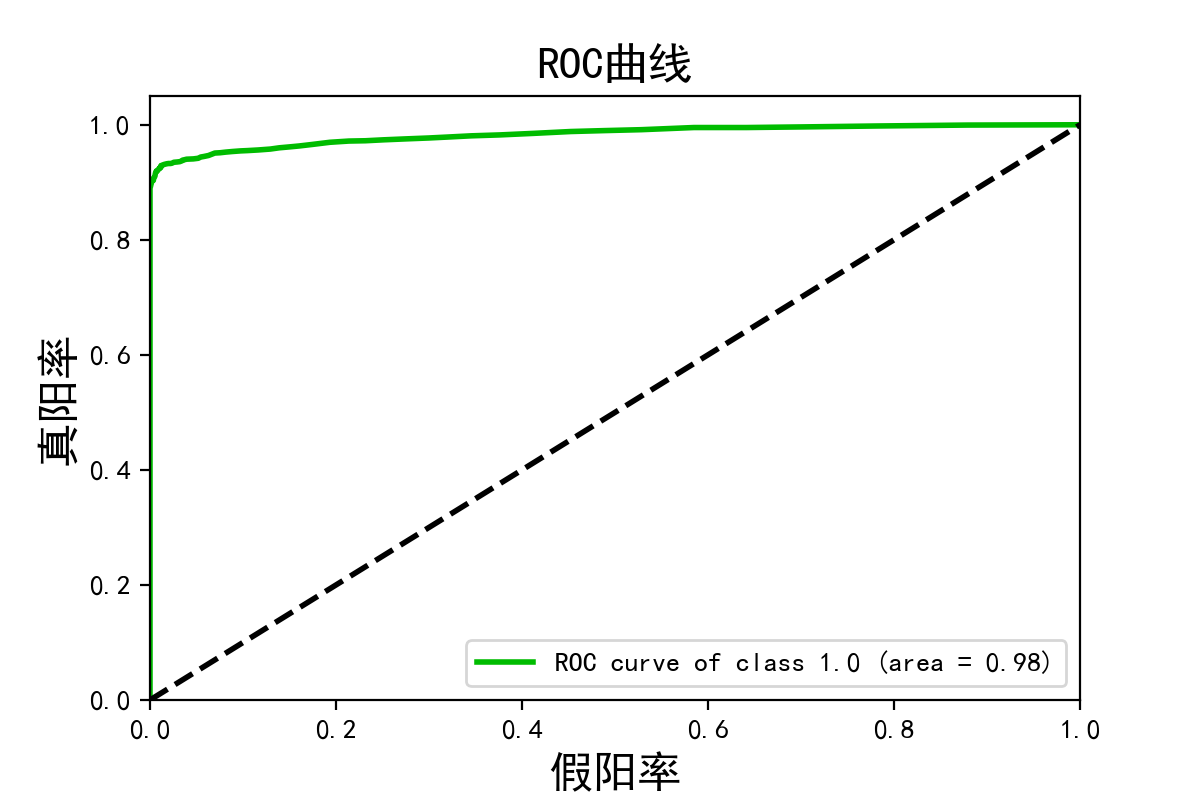


图3.3 优化后的随机森林模型ROC曲线

# 第4章 基于XGBoost构建脑卒中预测模型

## 4.1 XGBoost预测模型的构建

在建立 XGBoost模型前，需要对脑卒中数据集进行预处理，采用上述的方式，将处理后的脑卒中数据集以8:2的比例分成了训练集和测试集，其中训练集8088条，测试集2022条。然后，通过sklearn库调用模型，新建模型对象XGBoost = XGBClassifier()。先使用fit方法对训练集X\_train, Y\_train进行模型训练，之后对测试集x\_test进行预测，最终获得模型的得分和预测结果。

XGBoost方法对脑卒中数据处理的预测的结果见表4.1，其中，XGBoost模型的F1值为0.962，精确率为0.980。由图4.1可知，XGBoost的 AUC值为0.98。此外，这个模型在训练集上的精确度为98%，在测试集上的精确度为95%，说明该模式的整体表现良好。与随机森林模型相比，XGBoost模型的过拟合现象更小，且模型整体效果与随机森林相差不大，但模型的精度仍有待提高。故下面对该模型进行一些参数调整，以达到提高模型精度的效果。

表4.1 XG Boost模型结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 混淆矩阵 | 测试集准确率 | 训练集准确率 | F1 | 精确率 |
| [1873 35]  [150 1721] | 0.951 | 0.985 | 0.962 | 0.980 |

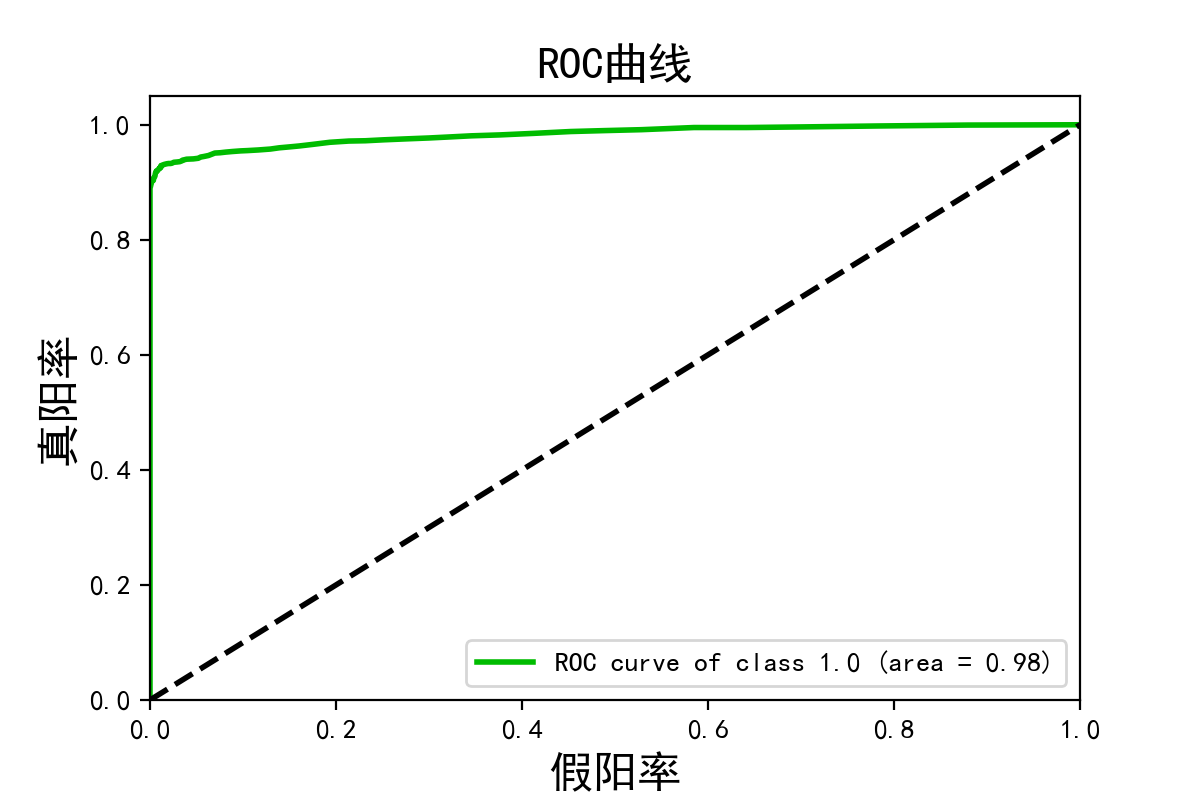


图4.1 XG Boost模型ROC曲线

## 4.2 优化XGBoost预测模型

XGBoost模型使用的初始值参数，见表4.2。各参数含义分别是：Silent为0时，静默模式开启，不会输出任何信息；反之，输出信息；min\_child\_weight表示最小叶子节点样本权重，默认为1；n\_estimators表示总迭代次数；max\_depth:树的最大深度，值越大，越容易过拟合；值越小，越容易欠拟合；Learning\_rate为控制每次根据估计误差对模型权重更新的多少。

表4.2 XG Boost模型参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| silent | min\_child\_weight | n\_estimators | Max\_depth | Learning\_rate |
| 1 | 1 | 100 | 7 | 0.05 |

因模型存在一定的过拟合问题，故本文将max\_depth=7,n\_estimators=100调整为max\_depth=6,n\_estimators=300。通过减少树的最大深度，增加迭代次数，可以提高模型的精度。优化后的XGBoost模型结果见表4.3。由表4.3可知，XGBoost模型的F1值为0.976，精确率为0.988，且过拟合现象明显减弱，模型的精度有所提升。由图4.2可知，ROC值仍为0.98。与调优前相比，XGBoost预测模型的精度有所提高。

表4.3 优化后的XG Boost模型结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 混淆矩阵 | 测试集准确率 | 训练集准确率 | F1 | 精确率 |
| [1873 35]  [148 1723] | 0.960 | 0.992 | 0.976 | 0.988 |

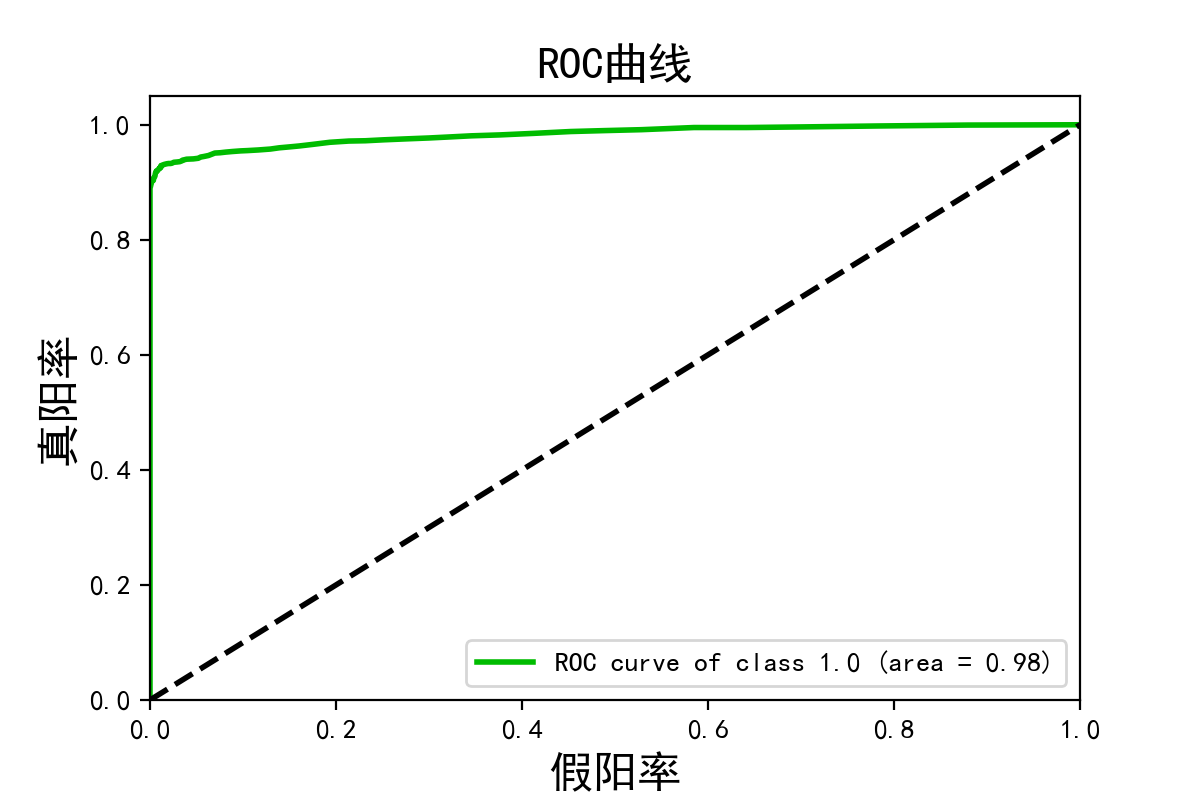


图4.2 优化后的 XG Boost模型ROC曲线

# 第5章 基于Cat Boost构建脑卒中预测模型

## 5.1 CatBoost预测模型的构建

在经过数据预处理之后，将处理后的脑卒中数据集按照8：2的比例分为训练集和测试集，其中训练集8088条，测试集2022条。然后，通过sklearn库调用模型，新建模型对象CatBoost = CatClassifier()。然后使用fit方法对训练集X\_train, Y\_train进行模型训练，并且对测试集x\_test进行预测，最终获得模型的得分和预测结果。

Cat Boost方法对脑卒中数据处理的预测的结果见表5.1，其中，Cat Boost模型的F1值为0.951，精确率为0.988。由图5.1可知，Cat Boost的AUC值也达到0.97，此外，它在训练集上的准确率为97%，测试集上的准确率接近95%，该模型的效果总体不错。为了进一步提高脑卒中预测的准确率，下面对该模型的参数进行调整。

表5.1 Cat Boost模型结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 混淆矩阵 | 测试集准确率 | 训练集准确率 | F1 | 精确率 |
| [1888 20]  [172 1699] | 0.949 | 0.970 | 0.951 | 0.988 |

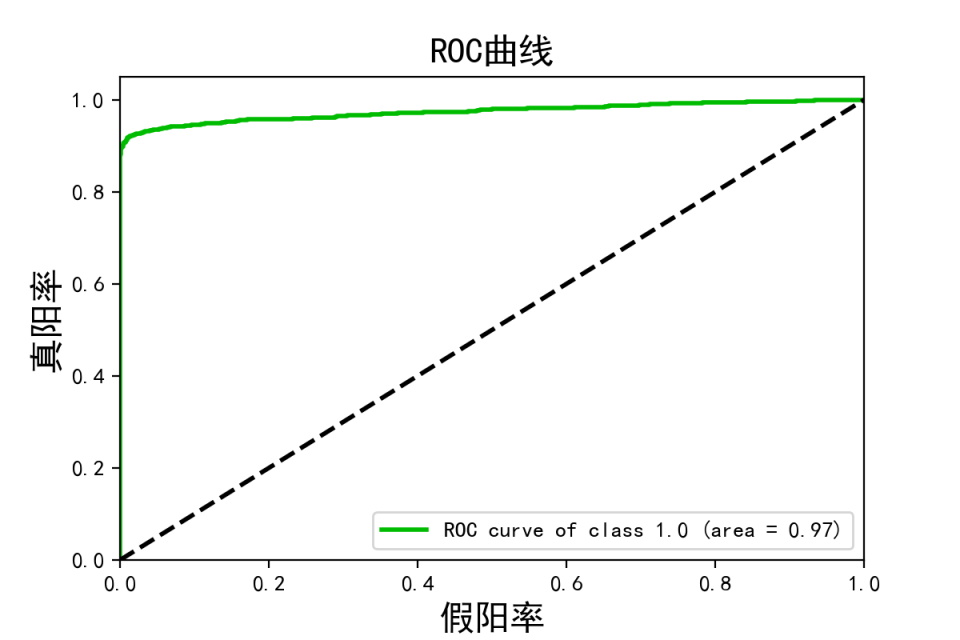


图5.1 Cat Boost模型ROC曲线

## 5.2 优化CatBoost预测模型

Cat Boost模型的参数与XG Boost模型相似，将max\_depth=6调整max\_depth=7,并且增加迭代次数之后，Cat Boost模型预测的结果见表5.2，其中，Cat Boost模型的F1值为0.956，精确率为0.986。由图5.2可知，Cat Boost模型的AUC值为0.98，在优化参数之后，模型的整体性能有所提高。

表5.2 调参后的Cat Boost模型结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 混淆矩阵 | 测试集准确率 | 训练集准确率 | F1 | 精确率 |
| [1885 23]  [153 1718] | 0.957 | 0.977 | 0.956 | 0.986 |

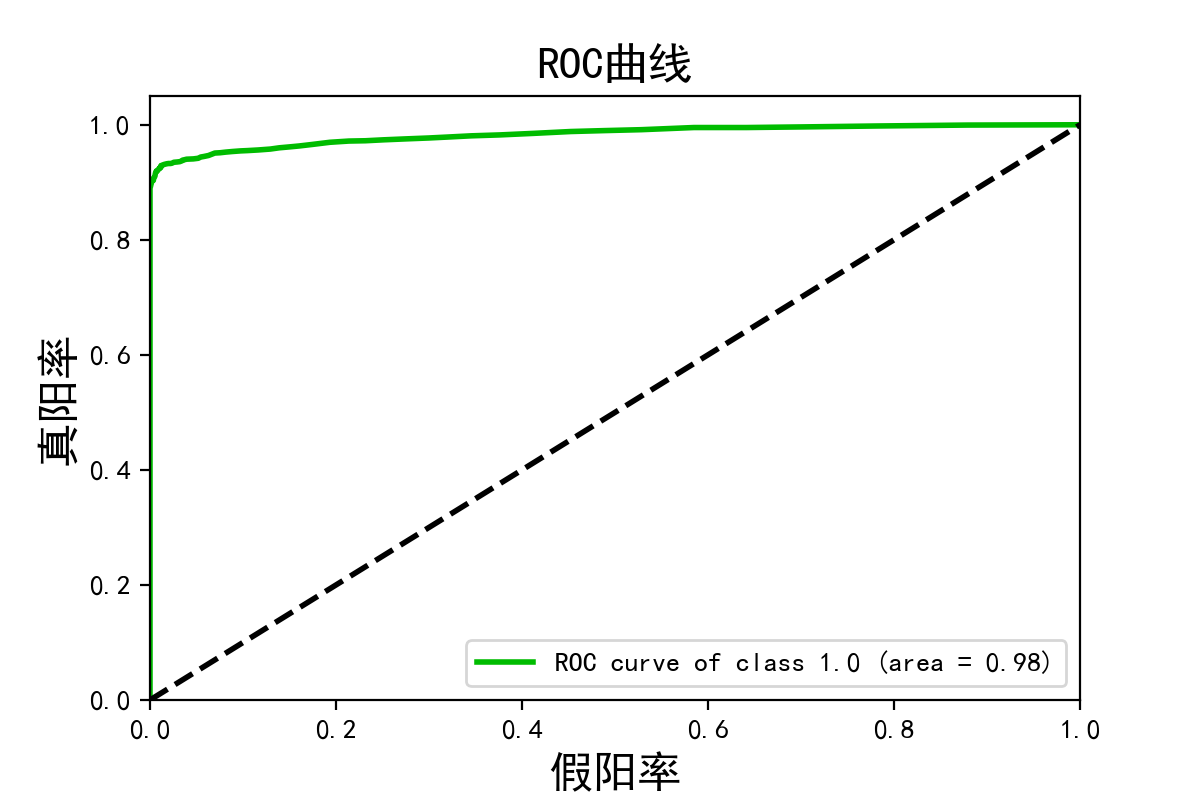


图5.2 调参后的Cat Boost模型ROC曲线

## 5.3 所有模型结果对比分析

将随机森林、XG Boost、Cat Boost三种模型的实验结果进行分析对比，见表5.3。由表5.3所知，随机森林模型的精确率为0.988，XG Boost模型的精确率为0.988，Cat Boost的精确率为0.986。从F1值的角度来看，随机森林模型的F1值为0.981，XG Boost模型的F1值为0.976，Cat Boost的F1值为0.956。由此可知，随机森林模型的效果最优，可用于脑卒中预测。

表5.3 模型结果对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 算法名称 | 测试集准确率 | 训练集准确率 | F1 | 精确率 |
| 随机森林 | 0.965 | 1.0 | 0.981 | 0.988 |
| XGBoost | 0.960 | 0.992 | 0.976 | 0.988 |
| CatBoost | 0.957 | 0.977 | 0.956 | 0.986 |

# 结论

脑卒中是一种患病率和死亡率很高的疾病，不同类型的脑卒中，其治疗方法不同。脑卒中的诊断结果受医生主观因素的影响，有效的预测脑卒中能为医生提供更加精准和有利的建议。随着科技的快速发展，机器学习被广泛应用于预测脑卒中。

本文选取了来源于kaggle网站的脑卒中数据，共计5110条。因数据量较少，故采用生成对抗网络(CTGAN)方法，利用真实的脑卒中样本生成更多的伪样本，以此将数据集扩充至10110条。然后对脑卒中数据进行数据预处理，包括填补缺失值、处理异常值、数据规约。因为脑卒中数据存在着数据不平衡的问题，所以使用了SMOTE方法解决脑卒中数据不平衡问题。在经过数据预处理之后，使用随机森林、XGBoost、CatBoost算法构建脑卒中预测模型，将10个特征作为脑卒中预测模型输入，通过调整决策树个数和树的最大深度等参数，提高预测准确率。从随机森林、XG Boost、Cat Boost三种模型的实验结果进行分析对比中发现，随机森林模型的F1分数为0.981，精确率为0.988，AUC值为0.98，在测试集上的准确率为96%。随机森林模型最优，可用于脑卒中预测。使用该模型对脑卒中数据进行预测分析，医生可以根据得出的预测结果为脑卒中患者进行治疗。

本文在基于机器学习的脑卒中预测研究中取得了一定的成果，但仍有许多不足之处。比如因脑卒中数据集的数据量较少，本文采用了生成对抗网络的方法生成了一部分数据，虽然解决了数据量少的问题，但生成的样本可能与真实样本存在着一些差异，可能会影响预测模型的精度。此外，虽然随机森林模型的精度很不错，但存在着一定的过拟合问题。在未来的研究中，希望能够减少生成的样本与真实样本的差异，进一步提高模型精度，为医生提供更加准确的建议。

# 参考文献

1. Maharjan Jenish, Ektefaie Yasha, Ryan Logan. Enriching the Study Population for IschemicStroke Therapeutic Trials Using a Machine Learning Algorithm[J]. Frontiers in Neurology, 2022, 12.
2. Aigner A, Gritter U, Rolfs A, Norrving, Bo，Siegerink, Bob，Busch, Markus A. Contribution of established stroke risk factors to the burden of stroke in young adults[J]. 2017, 48(7): 1744-1751.
3. 刘建模, 罗颢文, 俞鹏飞, 吴一帆, 韩梦琦, 贾伟杰, 易应萍. 基于机器学习的急性缺血性脑卒中医院感染预测模型建立与评价[J]. 中国感染控制杂志, 2023, 22(02): 129-135.
4. Chang W, Liu Y, Xu X. Prediction model of stroke recurrence based on support vector machine[J]. Journal of Physics Conference Series, 2019, 1324: 012095.
5. 宋伟才, 金国华, 陆路路, 江雨霆. 脑卒中发病率的统计分析及人工神经网络模型研究[J]. 科技广场, 2016(8): 4.
6. Min S N, Jin P, Im J N, M Subramaniyam. A Bayesian Model for Prediction of Stroke with Voice Onset Time[C]. ICAME2020.
7. Chun Matthew, Clarke Robert，Cairns Benjamin J，Clifton David. Stroke risk prediction using machine learning: a prospective cohort study of 0.5 million Chinese adults[J]. Journal of the American Medical Informatics Association : JAMIA, 2021, 28(8) : 1719-1727.
8. Ype de Jong, Chava L. Ramspek. A systematic review and external validation of stroke prediction models demonstrates poor performance in dialysis patients[J]. Journal of Clinical Epidemiology, 2020, 123(prepublish): 69-79.
9. 顾永涛, 徐泽禹, 盛庆博, 张卫, 李炜, 谭嘉, 李知艺. 基于生成对抗网络的光伏出力区间预测方法[J]. 哈尔滨理工大学学报, 2021, 26(02): 100-109.
10. 董师师, 黄哲学. 随机森林理论浅析[J]. 集成技术, 2013, 2(1): 1-7
11. 万安平, 龚志鹏, 张银龙, 纪云松, 马士东, 刘海南, 张运宁, 敖立争, 陈希, 彭晨. 基于XGBoost-KDE的风机主轴承故障预警方法[J]. 热力发电, 2022, 51(12):8.
12. 陈海宏, 易永利, 黄珅, 韩静怡. 基于CatBoost算法的短期光伏功率预测方法[J]. 浙江电力, 2023, 42(02): 67-75.
13. Prokhorenkova L, Gusev G, Vorobev A. CatBoost: unbiased boosting with categorical features[J]. 2017.
14. 刘洋. 基于机器学习的脑卒中预测模型的研究及其应用[D]. 哈尔滨: 东北林业大学, 2022.
15. 林振. 基于深度学习的卒中诊疗文本信息提取方法研究及应用[D]. 上海: 中国人民解放军海军军医大学, 2021.
16. 董永峰, 董彦琦, 张亚娟. 面向不平衡数据集的改进SMOTE算法[J]. 河北工业大学学报, 2022, 51(06): 40-46.
17. 杨柳, 王钰. 泛化误差的各种交叉验证估计方法综述[J]. 计算机应用研究, 2015, 32(05): 1287-1297.

# 致 谢

大学的生活已经接近了尾声，在校这四年的学习和生活经历，学到了很多。在这四年的学习生涯里，衷心的感谢各位老师对我的谆谆教诲，使我学到了很多知识，也非常感谢各位同学们对我学习和生活中的帮助，在这四年的学习生活中，认识了很多优秀的老师和同学，这是我人生中宝贵的财富。毕业论文完成后，我要感谢父母养育我的养育之恩。没有他们，就没有今天的我。同时，我也要感谢我的家人在我学习期间对我的鼓励和支持。在这段时间里，我还要感谢我的室友对我的帮助。我们曾在一起学习、生活和玩耍，互相交流、互相学习和探讨问题，建立了深厚的友谊。正是因为有了他们，我们才能度过这段美好而难忘的时光。非常感谢吕艳芬老师对我的细心指导，不断的给我提出改进意见，在论文的选题、构思和资料等方面都给予我悉心的教导，使我的论文能够不断的完善。

# 附录A

核心代码：

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

import scikitplot as skplt

from sklearn.svm import LinearSVC

from sdv.metadata import SingleTableMetadata

from ctgan import CTGAN

from sdv.evaluation.single\_table import evaluate\_quality

from sklearn.model\_selection import train\_test\_split, cross\_val\_score,

StratifiedKFold, GridSearchCV

from sklearn.metrics import accuracy\_score, classification\_report, roc\_curve,

precision\_recall\_curve, auc, confusion\_matrix, roc\_auc\_score

from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler

from sklearn.compose import ColumnTransformer

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

from sklearn.svm import SVC

from sklearn.impute import KNNImputer

from xgboost import XGBClassifier

from catboost import CatBoostClassifier

import warnings

warnings.filterwarnings("ignore")

#导入脑卒中数据集

data = pd.read\_csv('healthcare-dataset-stroke-data.csv')

data.head()

#利用生成对抗网络生成新样本

metadata = SingleTableMetadata()

metadata.detect\_from\_dataframe(data=data)

discrete\_columns = ['id', 'gender', 'age', 'hypertension',

'heart\_disease', 'ever\_married', 'work\_type',

'Residence\_type','avg\_glucose\_level',

'bmi','smoking\_status', 'stroke']

ctgan = CTGAN(epochs=10)

ctgan.fit(data, discrete\_columns)

synthetic\_data = ctgan.sample(5000)

quality\_report = evaluate\_quality(data,synthetic\_data,metadata)

df=pd.concat([data,synthetic\_data])

df.reset\_index(drop=True,inplace=True)

#导出新的脑卒中数据集

outputpath='data.csv'

df.to\_csv(outputpath,sep=',',index=False,header=True)

#数据预处理

data = pd.read\_csv("data.csv")

df = data.copy()

df.head()

df.drop(["id"], axis = 1, inplace = True)

df.info()

df.isnull().sum()

gender = df.iloc[:,0:1].values

ever\_married = df.iloc[:,4].values

work\_type = df.iloc[:,5:6].values

residence\_type = df.iloc[:,6].values

smoking\_status = df.iloc[:,9:10].values

age = df[["age"]]

hypertension = df[["hypertension"]]

heart\_disease = df[["heart\_disease"]]

avg\_glucose\_level = df[["avg\_glucose\_level"]]

bmi = df[["bmi"]]

stroke = df[["stroke"]]

unique, counts = np.unique(ever\_married, return\_counts = True)

print(np.asarray((unique, counts)).T)

le = LabelEncoder()

ever\_married = le.fit\_transform(ever\_married)

ever\_married = pd.DataFrame(ever\_married, columns = ["ever\_married"])

print("Labels 0, 1:",le.classes\_)

residence\_type = le.fit\_transform(residence\_type)

residence\_type = pd.DataFrame(residence\_type, columns = ["residence\_type"])

print("Labels 0, 1:",le.classes\_)

ohe = OneHotEncoder(dtype = np.int64, sparse = False)

gender = ohe.fit\_transform(gender)

gender = pd.DataFrame(gender, columns = ['female', 'male', 'other'])

print("Gender dummies respectively 0, 1, 2:", ohe.categories\_)

work\_type = ohe.fit\_transform(work\_type)

work\_type = pd.DataFrame(work\_type, columns = ['govt\_job', 'never\_worked', 'private', 'self-employed', 'children'])

print("Work type dummies respectively 0, 1, 2, 3, 4:", ohe.categories\_)

smoking\_status = ohe.fit\_transform(smoking\_status)

smoking\_status= pd.DataFrame(smoking\_status, columns = ['unknown', 'formerly\_smoked', 'never\_smoked', 'smokes'])

print("Smoking status dummies respectively 0, 1, 2, 3:", ohe.categories\_)

imputer = SimpleImputer(strategy = 'mean')

df\_en = pd.DataFrame(imputer.fit\_transform(df\_en), columns = df\_en.columns)

df\_en.head()

int\_vars=["female","male","other","age","hypertension","heart\_disease","ever\_married","govt\_job","never\_worked","private","self-employed","children","residence\_type",

"unknown","formerly\_smoked","never\_smoked","smokes"]

df1 = df\_en[int\_vars].astype(np.int64)

num\_vars = ["avg\_glucose\_level","bmi"]

df2 = df\_en[num\_vars]

label = ["stroke"]

df3 = df\_en[label]

df\_en = pd.concat([df1, df2, df3], axis = 1)

df\_en

df\_en.isnull().sum()

y = df\_en.stroke

X = df\_en.drop(['stroke'], axis=1)

#使用SMOTE处理数据不平衡问题

oversample = SMOTE(random\_state=42)

X, y = oversample.fit\_resample(X, y)

print("After Oversampling, the counts of label 1: ", y.value\_counts()[1])

print("After Oversampling, the counts of label 0: ", y.value\_counts()[0])

#将数据集按照8：2划分

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

scaler = StandardScaler()

x\_train = scaler.fit\_transform(x\_train)

x\_test = scaler.fit\_transform(x\_test)

#构建随机森林、XGBoost、Catboost模型

models = [('Random Forest', RandomForestClassifier()),

('XG Boost', XGBClassifier()),

('Cat Boost', CatBoostClassifier(logging\_level = 'Silent'))]

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

plt.rcParams['axes.unicode\_minus'] = False # 用来正常显示负号

models\_score = []

for name, model in models:

model = model

model.fit(x\_train, y\_train)

model.predict(x\_test)

models\_score.append([name, accuracy\_score(y\_test, model.predict(x\_test))])

print("Model: ",name)

print('Validation Accuracy: ', accuracy\_score(y\_test, model.predict(x\_test)))

print('Training Accuracy: ', accuracy\_score(y\_train, model.predict(x\_train)))

plt.figure()

cf\_matrix = confusion\_matrix(y\_test, model.predict(x\_test))

plt.title('Confusion Matrix: {}'.format(name))

sns.heatmap(cf\_matrix,annot=True,fmt='g',

cmap=sns.cubehelix\_palette(as\_cmap=True))

plt.show()

skplt.metrics.plot\_roc(y\_test, model.predict\_proba(x\_test),

title='ROC曲线',classes\_to\_plot=True,

figsize=(6, 4),

plot\_micro=False, plot\_macro=False)

plt.xlabel('假阳率')

plt.ylabel('真阳率')

plt.show()

#参数调优

models = [('Random Forest', RandomForestClassifier(criterion = 'gini', n\_estimators = 100, random\_state = 42)),

('XG Boost', XGBClassifier(max\_depth = 7, min\_child\_weight = 1)),

('Cat Boost', CatBoostClassifier(max\_depth = 7, n\_estimators = 300, logging\_level = 'Silent'))]

models\_score = []

for name, model in models:

model = model

model.fit(x\_train, y\_train)

model.predict(x\_test)

models\_score.append([name, accuracy\_score(y\_test, model.predict(x\_test))])

print("Model: ",name)

print('Validation Accuracy: ', accuracy\_score(y\_test, model.predict(x\_test)))

print('Training Accuracy: ', accuracy\_score(y\_train, model.predict(x\_train)))

plt.figure()

cf\_matrix = confusion\_matrix(y\_test, model.predict(x\_test))

plt.title('Confusion Matrix: Tuned {}'.format(name))

sns.heatmap(cf\_matrix,annot=True,fmt='g',cmap=sns.cubehelix\_palette

(as\_cmap=True))

plt.show()

skplt.metrics.plot\_roc(y\_test, model.predict\_proba(x\_test),

title='ROC曲线',classes\_to\_plot=True,

figsize=(6, 4),

plot\_micro=False, plot\_macro=False)

plt.xlabel('假阳率')

plt.ylabel('真阳率')

importance = model.feature\_importances\_

# summarize feature importance

for i,v in enumerate(importance):

print('Feature: %0d, Score: %.5f' % (i,v))

# plot feature importance

plt.figure(figsize = (12, 5))

plt.bar([x for x in range(len(importance))], importance)

plt.title("{} Classification Feature Importance".format(name))

plt.xticks(range(0, 19))

plt.show()