医疗保险欺诈监测模型

详细设计报告

团队编号：2404505

团队名称：Amazing

日期：2024-4-10

目录

1. 引言

1.1 项目背景

在当今的医疗保险领域，欺诈行为已成为一个日益严峻的问题。这不仅给保险公司带来了巨大的经济损失，也影响了保险行业的信誉和运行效率。当前，医疗保险欺诈主要通过伪造、夸大医疗服务费用或者通过虚假保险索赔进行。传统的欺诈检测方法多依赖于手动审查和基本规则的系统，但这些方法不仅耗时且效率不高，还很难捕捉到复杂和变化多端的欺诈模式。随着数据科学和机器学习技术的飞速发展，开发一个高效、智能的医疗保险欺诈检测模型变得尤为迫切，这对于提高欺诈检测的效率和精确性，减少经济损失，以及保护客户权益具有重要意义。

1.2 项目目的和重要性

本项目的主要目标是开发一个基于机器学习技术的医疗保险欺诈检测模型。这个模型旨在利用复杂算法自动识别出欺诈行为，从而比传统方法更高效、更精确地进行检测。通过分析大量医疗保险数据，模型可以识别出潜在的异常模式和欺诈行为，帮助保险公司减少不必要的经济损失，同时提升处理索赔的效率。此外，这一模型的开发还具有社会意义，能够帮助维护医疗保险市场的公平性和正直性，保护消费者和保险公司的合法权益。

1.3 项目范围和界定

本项目将专注于开发和优化一个基于数据驱动的医疗保险欺诈监测模型。项目将使用官方给出的医疗保险数据集进行模型训练和测试，以确保模型能够在实际应用中达到

预期的检测效果。然而，需要明确的是，尽管模型的目标是提高检测效率和准确性，但任何自动化系统都无法完全替代人工审核。因此，本项目也会考虑模型的可解释性，确保其检测结果能够为专业人员提供有效信息，以便进行进一步的人工审查。

2. 问题分析

2.1 赛题理解

开发一套医疗保险欺诈检测模型，对医疗保险违规行为进行识别。对给定数据集进行分析处理、提取特征因子集合，并使其越少越好。利用特征因子集合和 AI 相关算法构建模型，强调模型准确性和可解释性。

2.2 数据理解及分析

2.2.1 数据理解

1. 一共一张数据表，共有 16000 条样本，该数据集经过额外处理且没有说明文档。

2. 根据我们对数据的理解，我们认为数据集中的 RES 列为目标列，其中 0 表示非欺诈，1 表示欺诈，且欺诈样本数为 793，占比 4.9%，数据样本分布不均衡。可以结合有监督学习算法作为二分类问题进行处理。

3. 除去‘个人编码’列和‘RES’目标列，共有 80 个特征。

4. 每条一行代表一个样本，也就是一个人的就诊行为，属于一对一关系。

2.2.2 数据分析

在本项目中做数据分析的主要目的是为了理解欺诈行为的特征，为特征工程做准备。

2.2.2.1 基本统计分析

总体分析：使用 PCA 将数据降低到三维空间，提取主要信息，对整体数据分布以及可分离性进行观察

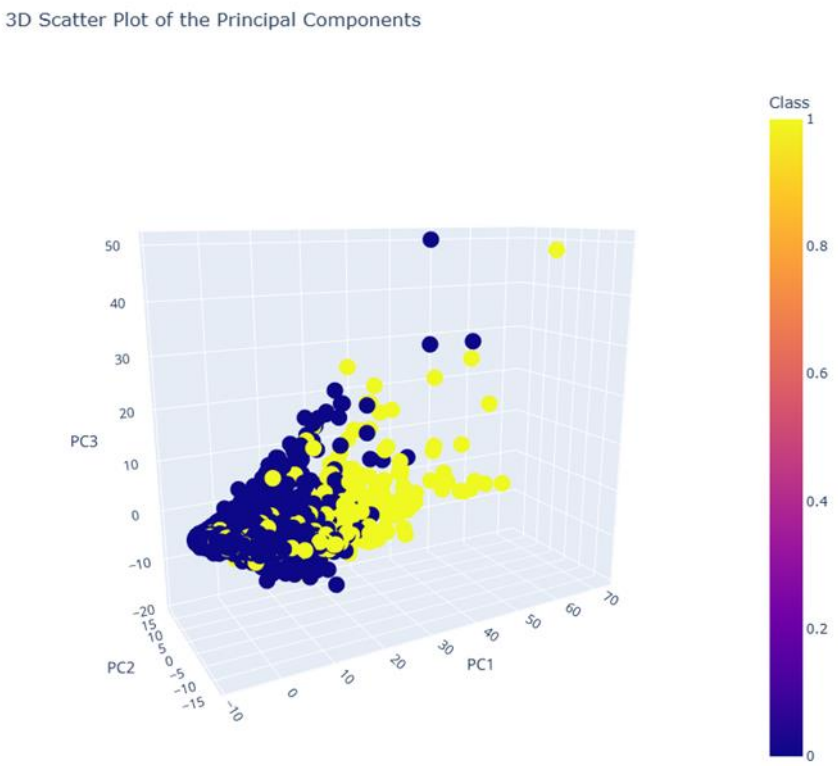

图 2-1 PCA 三维数据分布

从 PCA 可视化图形中可以得到以下结论：

1. 数据点在三维空间中的分布具有一定程度的聚类特性，但聚类的界限并不特别清晰。

2. 存在重叠区域，许多类别 1 样本与类别 0 重合。这意味着仅凭这三个主成分可能不足以完全区分两个类别。

3. 类别 1 数据中含有大量噪声，导致两个类别难以区分

就诊行为分析： 以‘月就诊次数_MAX’，‘月就诊天数为例’。

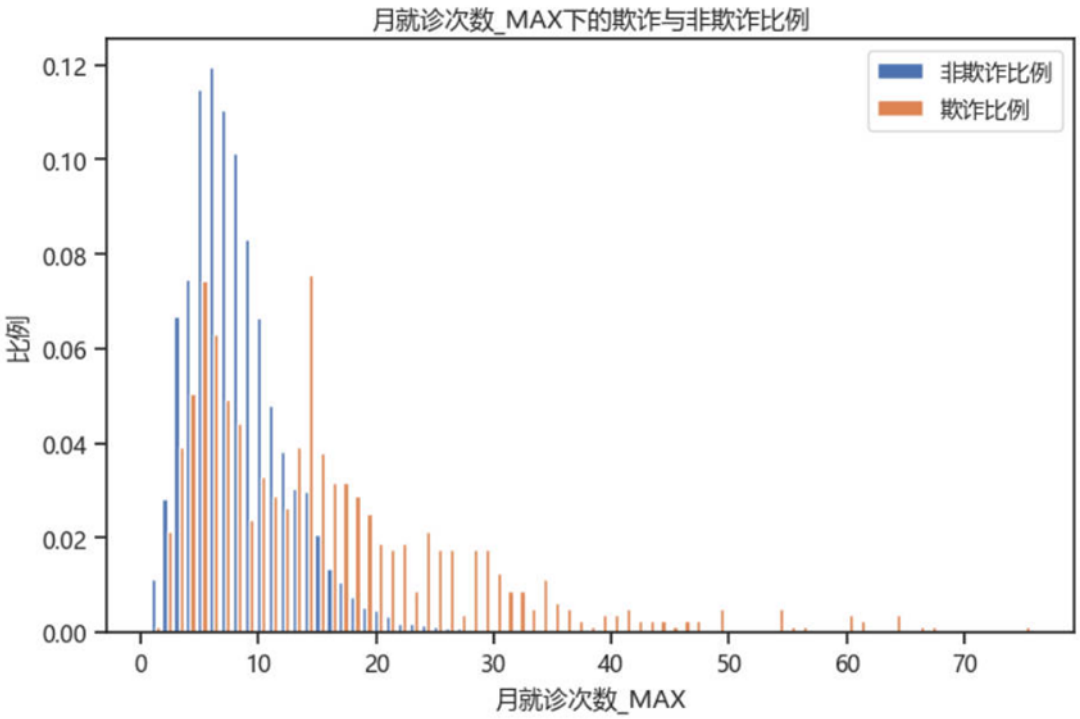

图 2-2 月就诊次数_MAX 下的欺诈与非欺诈比例

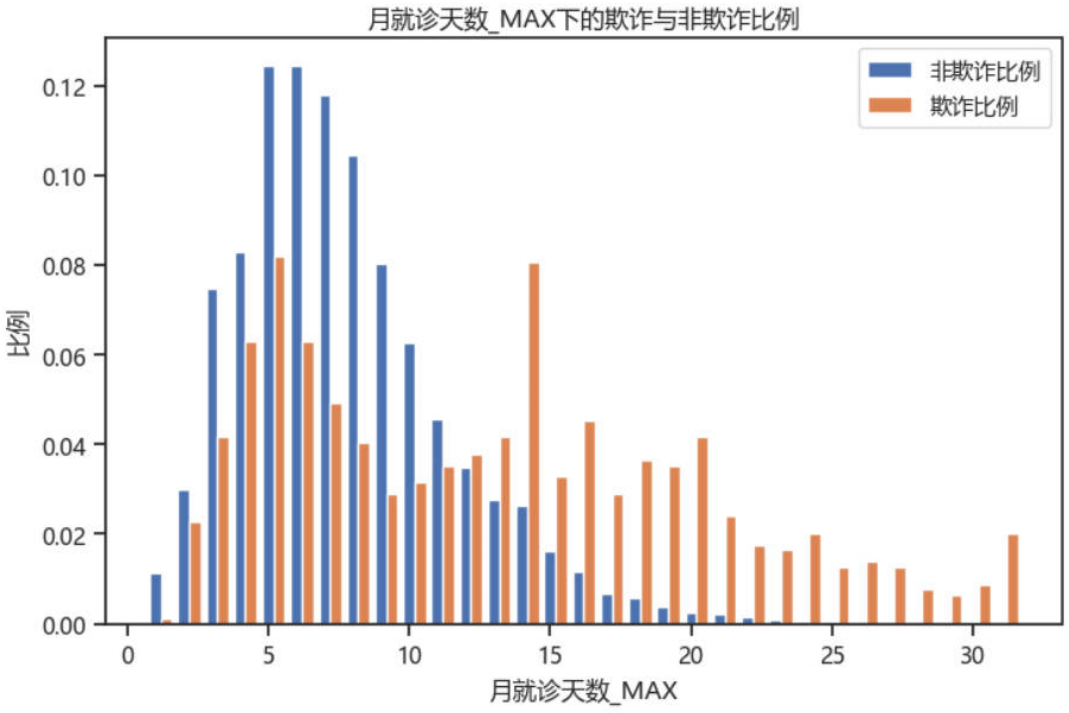

图 2-3 月就诊天数_MAX 下的欺诈与非欺诈比例

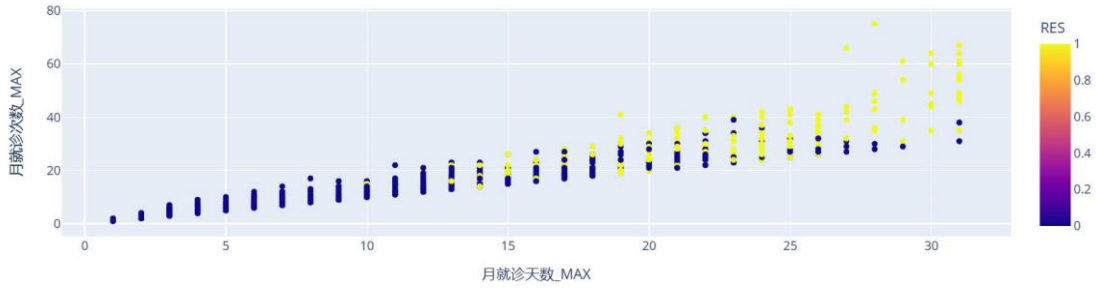

图 2-4 月就诊次数_MAX 与月就诊天数_MAX 数据分布

从两个表中可以看到，在一个月中，较高的就诊次数与就诊天数能够在一定程度上区分欺诈与非欺诈，但这种区分度并不高，其中一部分样本是重合的。

许多欺诈者的欺诈手段十分隐蔽，他们通过较少的就诊次数，在单个时间点进行高额诈骗；或者通过较多的就诊次数，在多个时间点进行诈骗。这也是样本分布重合的原因。仅仅通过两三个特征几乎不能有效识别是否存在欺诈行为。

基础费用分布分析：以药品金额与三目明细比例为例。

药品金额：

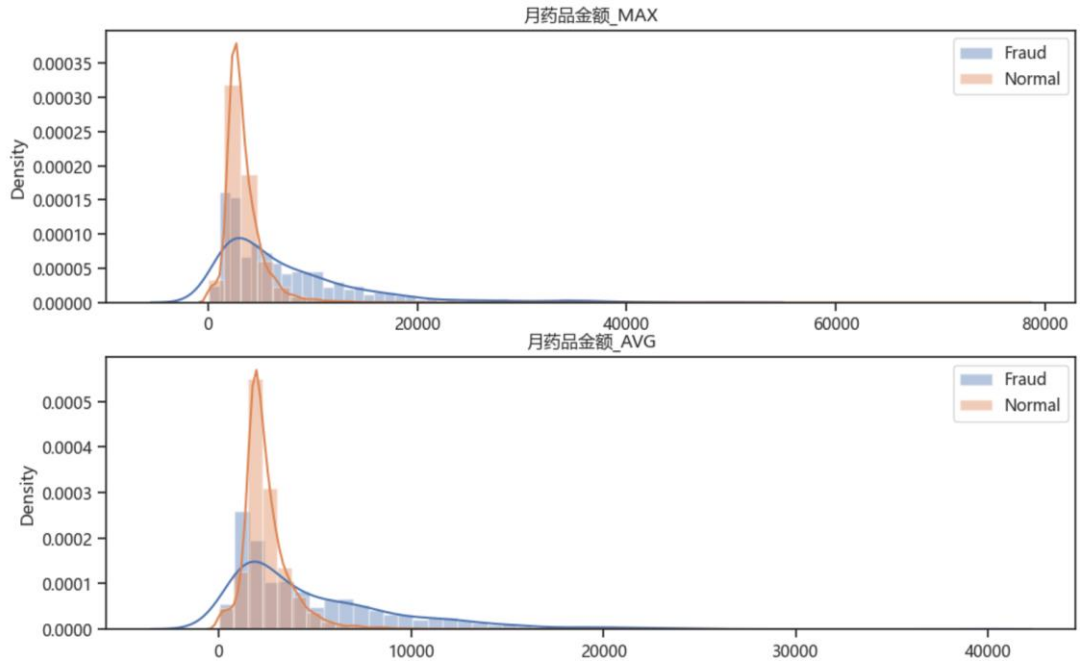

图 2-5 药品金额最大值与最小值分布

月药品金额最大值与平均值的分布并无太大差别，且两个类别的数据大部分都重叠在了一起。

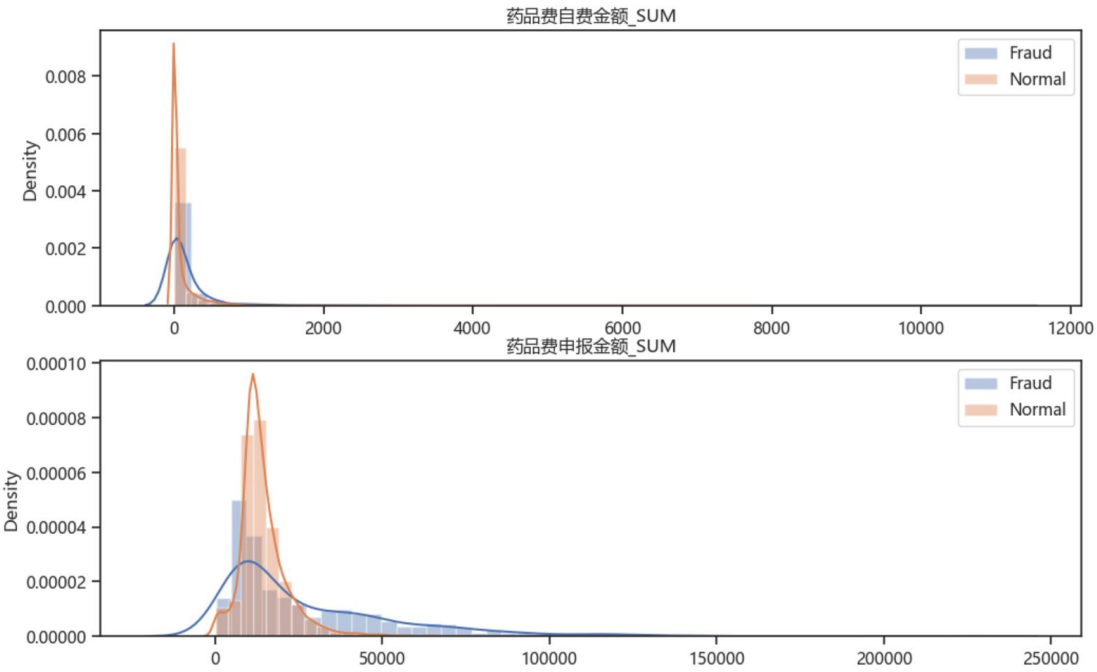

图 2-6 药品自费与申报金额分布

尽管药品费自费金额_SUM 的两个类别分布几乎重合，但药品费申报金额_SUM 的分布仍有一定的区别。那么通过计算申报金额所占自费金额的比例也能够在一定程度上区分欺诈与非欺诈。

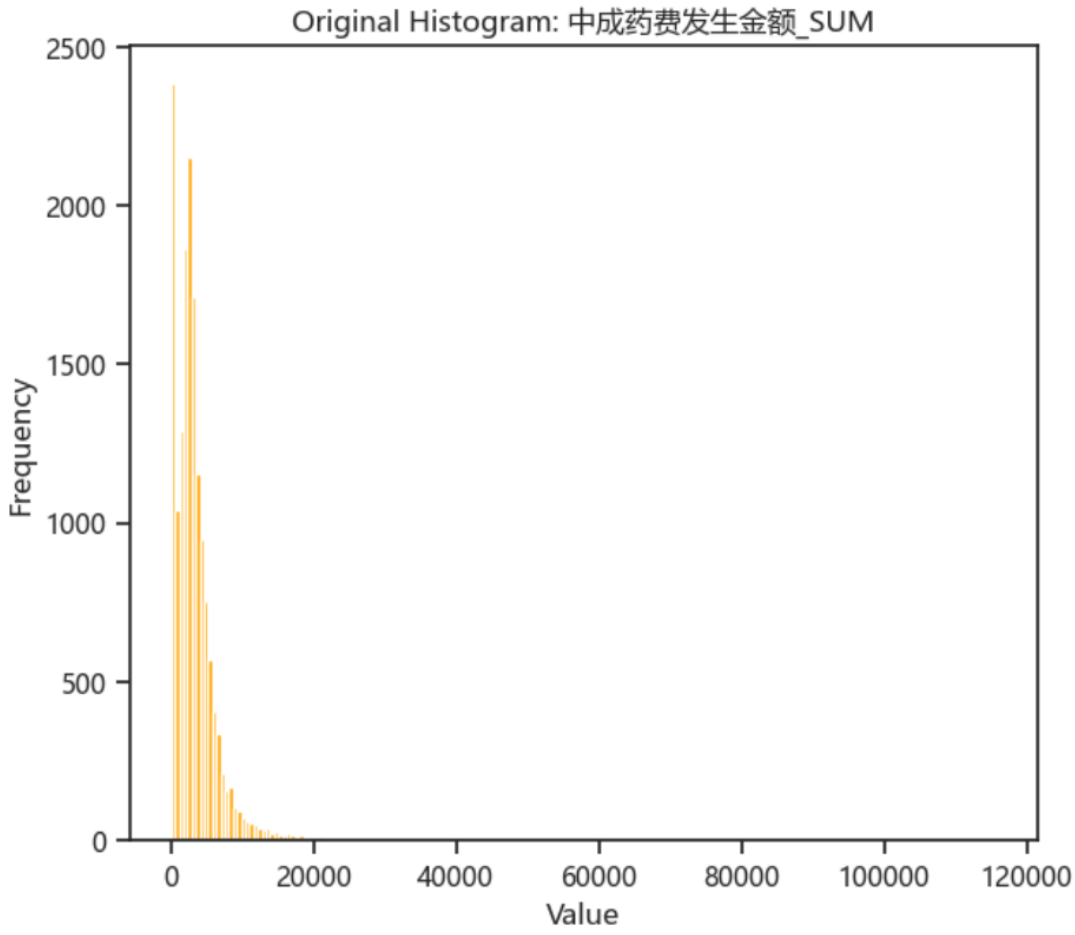

图 2-7 中成药费发生金额分布

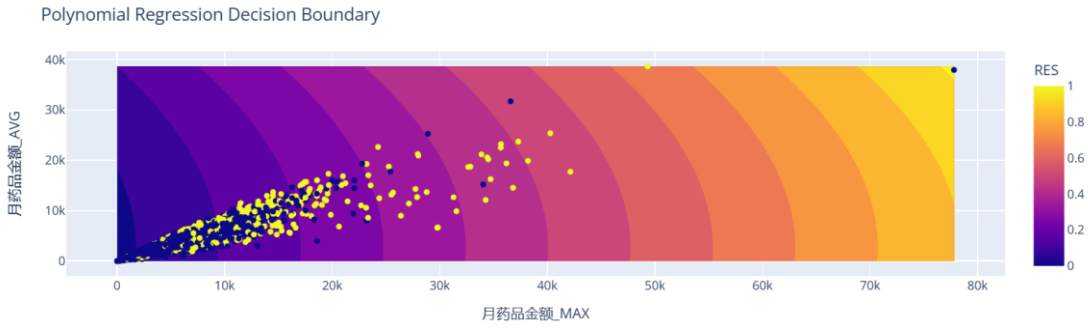

图 2-8 月药品金额最大值与平均值关系分布

部分特征如同中成药费发生金额一样，数据并不是正态分布。并且部分特征之间的关系呈非线性关系。在进行相关性分析时不再使用皮尔逊相关系数，而是使用斯皮尔曼等级相关系数。

三目明细比例：

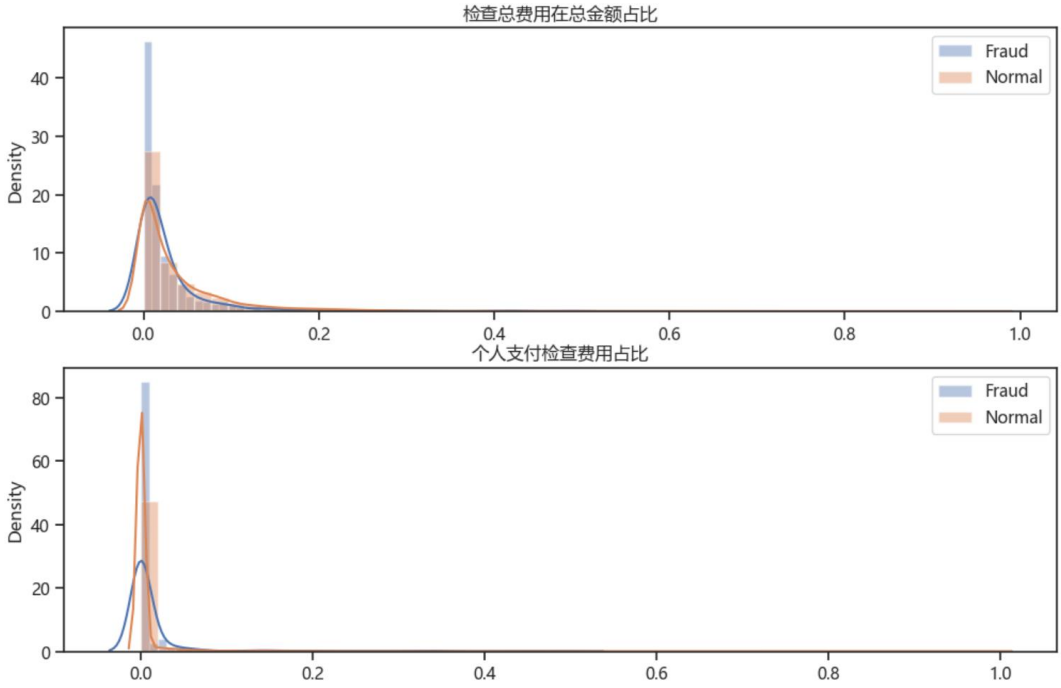

图 2-9 检查与个人支付金额占比分布

检查总费用占比与个人支付检查费占比几乎没有区分度。

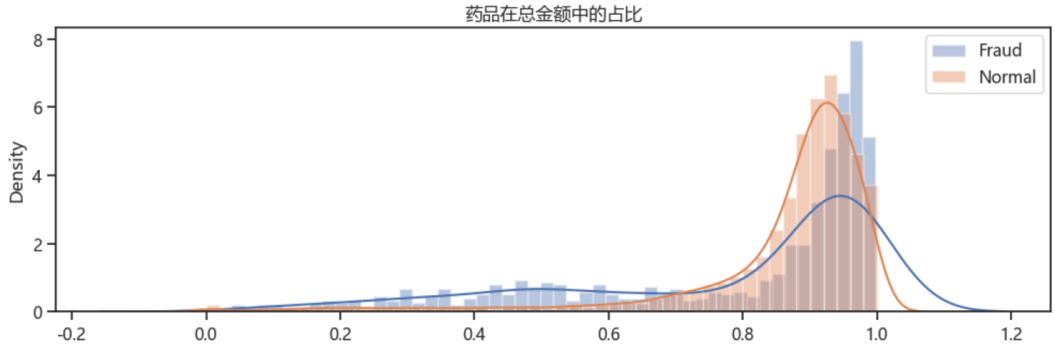

图 2-10 药品金额占比分布

药品在总金额中的占比具有一定区分度，尽管不明显，这表明与药品费用有关的特征

可以是特征提取的重点。

2.2.2.2 相关性分析

基于对数据分布以及变量间的关系的分析与观察，相关性分析使用斯皮尔曼等级相关系数。主要有以下考虑：

- 斯皮尔曼等级相关系数是一种非参数方法，它衡量的是变量之间的单调关系，而不仅仅是线性关系。这使得它能够检测出更广泛类型的关系，即使这些关系不是线性的。
- 斯皮尔曼系数对异常值不太敏感，因为它基于数据的等级（排名），而不是实际值。这意味着异常值对整体相关性的影响较小。
- 适用于定序（顺序）数据和定量（连续或离散）数据。
- 不要求数据遵循正态分布。

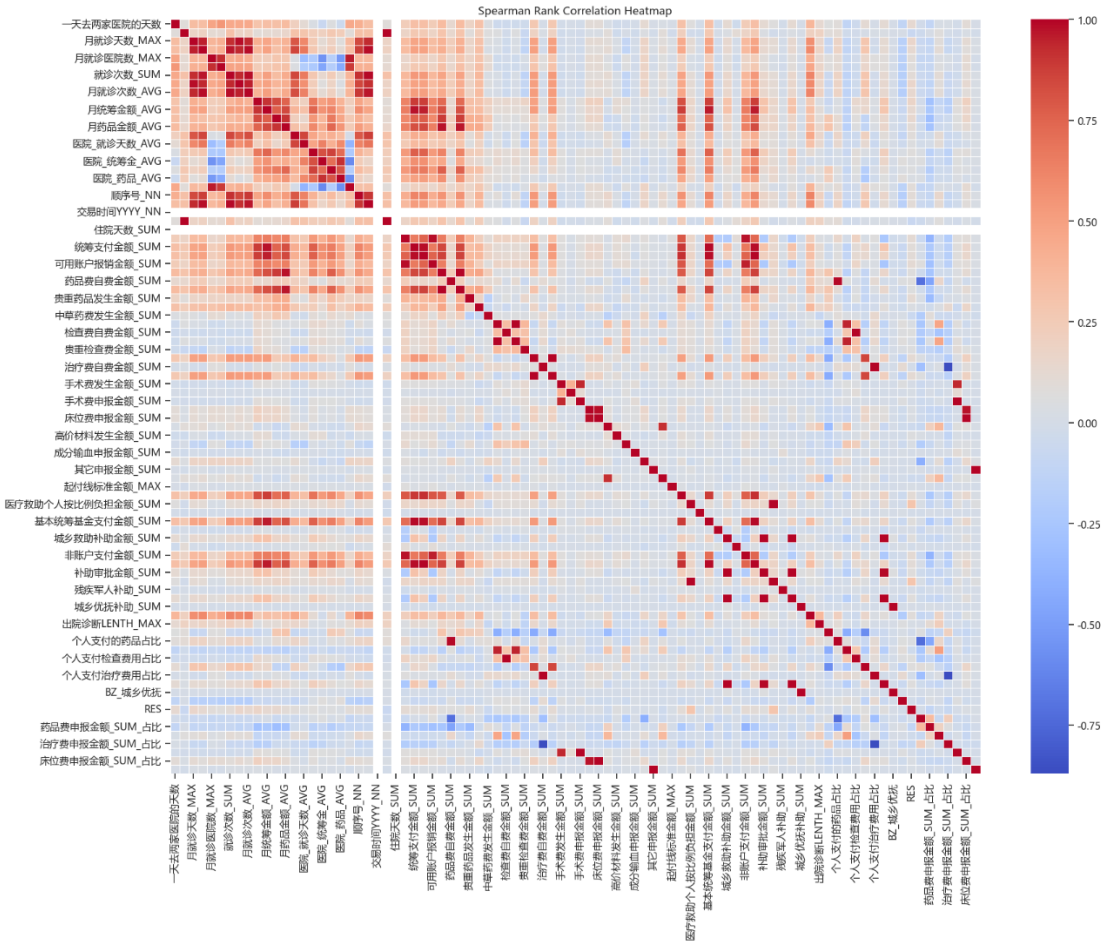

图 2-11 斯皮尔曼等级相关系数热力图

2.3 解决方法与核心思路

- 主要目标是要鉴别出欺诈行为，可以从具体的医保欺诈手段、欺诈者的内心活动、欺诈行为监管力度等角度为出发点进行分析，把实际发生的欺诈行为抽象为在数据上可能的表现形式。此外，医保欺诈核心目的是骗钱，各项费用应该是特征提取的重点。

- 少数类（欺诈）样本较少，仅有 5%，极端的类不平衡可能会使得模型倾向于多数类（非欺诈）。对于类不平衡处理，现有的技术主要包括过采样少数类（如 SMOT 及其变体)、欠采样多数类、少数类加权、代价敏感学习，较为前沿的技

术是利用 GAN（生成对抗网络）来合成样本。

- 由于存在极端的类不平衡问题，仅仅使用常规的模型评估方式不足以准确评估模型性能，例如，使用准确率评估模型性能时，模型仅仅全部预测多数类，也能够达到 95%的准确率，而欺诈样本全部未能预测出来，这显然不是我们想要的结果。本项目采用分类报告与 AUC 来评估模型。
- 在可解释性方面，目前已经有了比较成熟的工具，例如 shap 库能够提供各种模型的解释方法，在可解释性方面不再过多研究，使用成熟的工具能够使得项目在实际应用中更加稳定。
- 在实际应用中，会产生不断地数据，这部分数据能够帮助提升模型的性能，也能够识别新的欺诈模式。因此，快速迭代模型是至关重要的，我们在选择模型时使用了速度较快的算法（如随机森林和轻量型梯度提升机），以保证在不会明显降低性能的情况下保持能够快速迭代的优势。

3. 技术和方法论

3.1 数据预处理

3.1.1 空值和异常值

项目的最终阶段采用的是随机森林和轻量型梯度提升机，这两个算法对空值并不敏感，但是我们在最初测试了其他算法，所以对空值做了些处理。数据集中仅在‘出院诊断 LENTH_MAX’这一列，存在空值，经分析后将其填充为 0。对于异常值，由于本项目主要是做欺诈检测，异常值可以提供必要的信息，故不做处理。

3.1.2 数据分割与采样

- 数据集将分为训练集和测试集，其比例为 0.75:0.25，我们希望有更多的数据被用于训练，同时又需要确保测试集能够较为准确的反映模型的性能。
- 在超参数调优时，我们使用交叉验证来评估模型，对训练集采用 5 折交叉验证。交叉验证能够减少过拟合风险、提供更可靠的性能评估和更好的超参数泛化。
- 评估了各种采样方法后，我们最终使用手动欠采样多数类来平衡数据集，其样本数目分别为 2000 与 595。过采样少数类会严重增加过拟合风险，SMOTE 及其变体会引入大量噪声，实测效果不佳。

3.2 特征工程

3.2.1 特征提取

特征提取主要是从统计的角度出发，结合对于业务的理解。主要包含以下几个方面：

- 组合特征并编码：

组合特征能够进一步揭示数据中存在的复杂关系，例如“医院编码_NN、出院诊断病种名称_NN”，可能只有部分医院能够诊断特定的病种，而欺诈人员会选择这些病种进行欺诈。

其中‘交易时间_NN’由交易时间 DD_NN、交易时间 YYYY_NN、交易时间 YYYYMM_NN 组合而成。

表 3-1 组合特征

交易时间 DD_NN、交易时间 YYYY_NN、交易时间 YYYYMM_NN
医院编码_NN、出院诊断病种名称_NN、顺序号_NN
医院编码_NN、顺序号_NN
医院编码_NN、出院诊断病种名称_NN

医院编码_NN、是否挂号
出院诊断 LENTH_MAX、是否挂号
出院诊断病种名称_NN、是否挂号
交易时间_NN、就诊的月数

- 比例特征：

表 3-2 比例特征及其说明

就诊欺诈系数 诊断欺诈系数	以就诊欺诈系数为例：某医院编号为 1，该医院在非欺诈类别中出现 30 次，在欺诈类别中出现 10 次，那么欺诈系数就为 10/(30+10)=0.25。 欺诈系数能够在一定程度上揭示欺诈人员对特定医院和特定病种的偏好，这些医院或者病种可能更加容易骗保。
自费比例 申报比例	药品费发生金额_SUM、检查费发生金额_SUM、治疗费发生金额_SUM、手术费发生金额_SUM、床位费发生金额_SUM 与其它发生金额_SUM 中各自的自费金额与申报金额占发生金额的比例。 部分欺诈者可能为了骗保，会虚构发生金额，而其自费与申报所占比例差距过大
平均费用波动	使用所有费用特征除以就诊次数_SUM 得到费用波动特征。 部分欺诈行为及其隐蔽，尽管每次的骗保金额不大，且就诊次数较少。 过高的平均费用可能表明过度诊疗或虚报费用。一些欺诈者可能会在少数几次就诊中索取高额费用，以避免频繁的小额欺诈容易被发现，或者频繁就诊平摊风险。

- 基本统计特征：

表 3-3 基本统计特征及其说明

特征差值	部分特征有最大值与平均值，计算二者的差得到每个特征的差值。如月就诊天数差、月统筹金额差等 较大的差值可能表明个体在某些日子里就诊次数异常多、频繁更换医院；一些欺诈者可能会在少数几次就诊中索取高额费用
标准差差值	计算费用特征的标准差，之后减去标准差。 如果一个数据点远低于这个特征定义的标准值，它可能被视为异常或离群值。离群值很可能存在欺诈行为。
病种费用标准差差值 医院费用标准差差值	对病种和医院进行分组，计算每个病种和医院对应的费用的标准差，并减去标准差得到差值特征 不同病种和医院所花费的费用可能是不同的，比较个别案例与医院整体平均水平的差异，从而识别异常或不寻常的行为 例如，如果某个病人的就诊费用远高于该医院的平均水平，这可能表明存在欺诈或滥用的风险。
病种平均费用差值 医院平均费用差值	对病种和医院进行分组，计算每个病种和医院对应的费用的平均费用，并减去均值得到平均费用差值特征 费用明显高于或低于平均水平的病例可能表明欺诈、过度治疗或不必要的医疗服务，可以更容易地识别出偏离常规治疗模式的案例

- 分箱特征：

该特征主要用于构建逻辑回归，因为随机森林和 lightGBM 能够很好的处理非线性关系，这种转换可能有助于改善逻辑回归的性能。其方法是将所有连续特征按照十分位数进行分箱。

- 多项式特征：

同分箱特征类似，对于基于树的模型（如随机森林、梯度提升机等）不是很需要多项式特征，因为它们能够很好的扑捉非线性关系，但对于线性模型或某些基于距离的模

型，这些特征可能有助于改善性能。其方法是构建多项式特征转换器，利用二阶多项式构建特征。

3.2.2 特征选择

特征选择核心在于迭代地优化特征集，结合贪婪思想，通过自动删除较不重要或高度相关的特征，缩减特征因子集合，提高模型的泛化能力和性能。

这种方法有效地结合了特征重要性评分和相关性分析，以确保最终模型只包含对预测目标变量最有影响力的特征。通过多次迭代，模型逐渐去除不必要的特征，使模型更加精简高效。

根据使用的数据集的特征数量不同，可以更改迭代次数，在第一次迭代时根据特征重要性评分和相关性分析删除大量特征，之后逐个删除特征。

传统的特征选择逐个添加特征，在添加过程中训练并评估模型，这种方法不能扑捉特征之间的潜在关联，如果所舍弃的特征和后来添加的特征有关联，该方法就不能判断出来。另一种流行方式是递归特征消除法（RFE），我们借鉴了这种思路，但做了进一步的改进，我们在第一次迭代时删除了相关性较高的特征，而传统的方式仅仅是依照特征重要性逐个删除特征，没有考虑相关性。

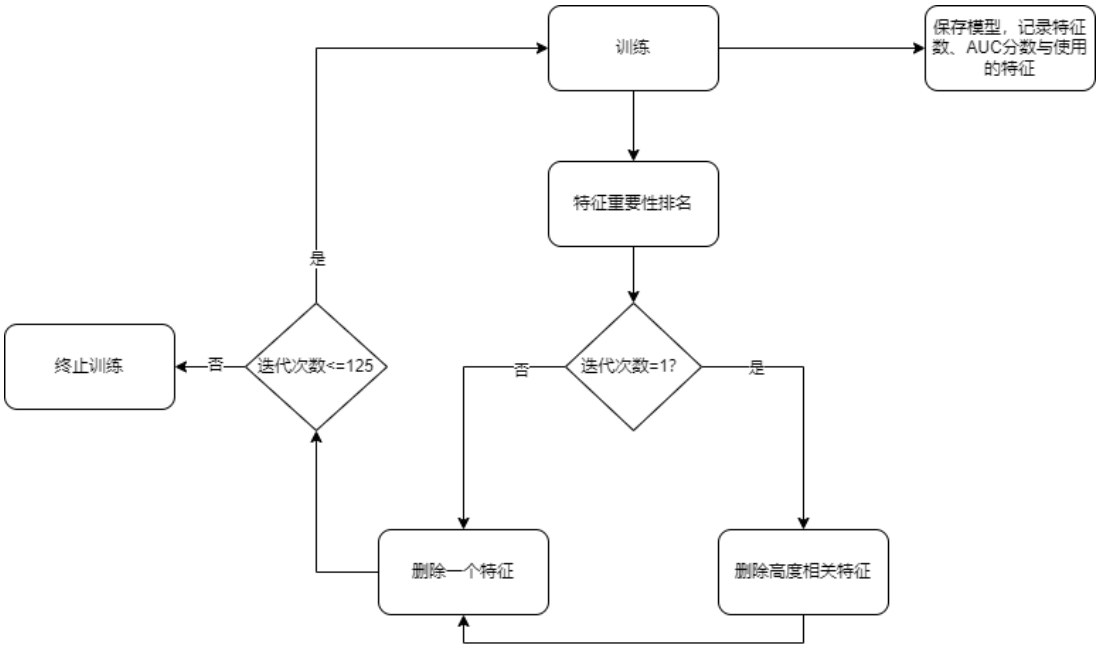

图 3-1 特征选择简易流程

3.3 模型构建与方法选择

3.3.1 算法评估与方法选择

1. 初步评估，使用原始数据集，构建多个常见的分类器，如决策树、逻辑回归、随机森林、梯度提升机、等。数据集划分为训练集和测试集，比例为 0.75:0.25，随机数种子均为 42。

以下分类器除了多层感知机与隔离森林，其余均使用默认参数。

多层感知机共5个全连接层，每层神经元数量分别为128、128、128、64、32、16，使用 Relu 激活函数与 Adam 优化器，迭代 300 次。

表 3-4 分类器原始数据集评估

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试

Decision Tree	1.00	0.93	1.00	0.68	1.00	0.94	1.00	0.702
Logistic	0.96	0.96	0.71	0.71	0.95	0.95	0.845	0.853
RF	1.00	0.96	1.00	0.74	1.00	0.96	1.00	**0.917**
GBDT	0.97	0.96	0.80	**0.76**	0.97	0.96	0.960	0.912
LightGBM	1.00	0.96	0.99	0.74	1.00	0.96	0.999	**0.935**
XGBoost	1.00	0.96	1.00	0.74	1.00	0.96	1.00	**0.926**
MLP	1.00	0.95	0.98	0.71	1.00	0.95	0.999	0.884
ISO_Forest	0.96	0.96	0.65	0.65	0.93	0.93	NAN	

注意：由于类别 1 的样本数量稀少，因此训练集和验证集的划分比例很重要，如果将比例划分为 0.8:0.2，会发现模型评分会略微上升，这并不是模型的性能有所提高，而是用于评估的样本变少了。部分分类器重复训练可能有不同的结果。

2. 从初步评估的结果表中不难发现大多数分类器在训练集上表现良好，而在测试集上表现不佳。有两个方面会造成这种结果，欺诈类样本数量过少，模型偏向于多数类或者模型过分拟合少数类样本。下面将使用手动随机欠采样多数类、SMOTE 过采样、少数类加权/代价敏感学习三种方式来解决少数类样本过少的问题。

多层感知机在之后的评估中将被舍弃，原因在于多层感知机相比于逻辑回归与基于树的模型效率低且表现不佳。

- 手动欠采样：

表 3-5 样本数量分别为 595:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.77	1.00	0.56	1.00	0.84	1.00	0.774
Logistic	0.78	0.84	0.78	0.61	0.78	0.88	0.880	0.860
RF	1.00	0.82	1.00	0.60	1.00	0.87	0.999	0.909
GBDT	0.95	0.83	0.95	0.62	0.95	0.88	0.993	0.920
LightGBM	1.00	0.83	1.00	0.62	1.00	0.87	1.00	0.926
XGBoost	1.00	0.84	1.00	0.62	1.00	0.88	1.00	0.923

表 3-6 样本数量分别为：1000:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.83	1.00	0.60	1.00	0.87	1.00	0.791
Logistic	0.81	0.90	0.77	0.66	0.80	0.92	0.871	0.861
RF	1.00	0.89	1.00	0.67	1.00	0.91	1.00	0.916
GBDT	0.94	0.88	0.93	0.66	0.94	0.91	0.988	0.917
LightGBM	1.00	0.88	1.00	0.66	1.00	0.90	1.00	0.923
XGBoost	1.00	0.87	1.00	0.65	1.00	0.90	1.00	0.923

表 3- 7 样本数量分别为：1500:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.87	1.00	0.63	1.00	0.90	1.00	0.778
Logistic	0.84	0.93	0.78	0.69	0.83	0.93	0.864	0.870
RF	1.00	0.92	1.00	0.71	1.00	0.93	1.00	0.917
GBDT	0.93	0.92	0.91	0.71	0.93	0.93	0.984	0.922
LightGBM	1.00	0.91	1.00	0.70	1.00	0.93	1.00	0.930
XGBoost	1.00	0.91	1.00	0.69	1.00	0.92	1.00	0.927

表 3-8 样本数量分别为：2000:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.89	1.00	0.65	1.00	0.91	1.00	0.762
Logistic	0.88	0.95	0.77	0.72	0.87	0.95	0.858	0.871
RF	1.00	0.95	1.00	0.77	1.00	0.96	1.00	0.919
GBDT	0.94	0.95	0.89	0.75	0.93	0.95	0.977	0.925
LightGBM	1.00	0.94	1.00	0.75	1.00	0.95	1.00	0.931
XGBoost	1.00	0.93	1.00	0.73	1.00	0.94	1.00	0.929

表 3-9 样本数量分别为：2500:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.89	1.00	0.65	1.00	0.91	1.00	0.762
Logistic	0.88	0.95	0.77	0.72	0.87	0.95	0.858	0.871
RF	1.00	0.95	1.00	0.77	1.00	0.96	1.00	0.919
GBDT	0.94	0.95	0.89	0.75	0.93	0.95	0.977	0.925
LightGBM	1.00	0.94	1.00	0.75	1.00	0.95	1.00	0.931
XGBoost	1.00	0.93	1.00	0.73	1.00	0.94	1.00	0.929

表 3-10 样本数量分别为：3000:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.89	1.00	0.64	1.00	0.91	1.00	0.734
Logistic	0.89	0.95	0.76	0.71	0.88	0.95	0.857	0.870
RF	1.00	0.95	1.00	0.76	1.00	0.95	1.00	0.925
GBDT	0.94	0.95	0.88	0.76	0.93	0.95	0.977	0.924
LightGBM	1.00	0.95	1.00	0.75	1.00	0.95	1.00	0.933
XGBoost	1.00	0.94	1.00	0.75	1.00	0.95	1.00	0.935

- 少数类加权/代价敏感：

表 3-11 原始数据集样本数

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.94	1.00	0.69	1.00	0.94	1.00	0.689
Logistic	0.85	0.85	0.61	0.61	0.89	0.89	0.869	0.871
RF	1.00	0.96	1.00	0.71	1.00	0.95	1.00	0.914
GBDT	0.90	0.89	0.72	0.67	0.92	0.91	0.973	0.925
LightGBM	0.98	0.94	0.92	0.74	0.98	0.95	0.999	0.928
XGBoost	1.00	0.95	1.00	0.74	1.00	0.95	1.00	0.924

表 3-12 样本数分别为 1000:500

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.83	1.00	0.60	1.00	0.87	1.00	0.786
Logistic	0.79	0.84	0.77	0.60	0.79	0.88	0.874	0.862
RF	1.00	0.89	1.00	0.67	1.00	0.91	1.00	0.914
GBDT	0.94	0.85	0.94	0.64	0.95	0.89	0.989	0.919
LightGBM	1.00	0.86	1.00	0.65	1.00	0.90	1.00	0.923
XGBoost	1.00	0.86	1.00	0.65	1.00	0.65	1.00	0.920

表 3-13 样本数分别为 1500:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.86	1.00	0.63	1.00	0.89	1.00	0.791
Logistic	0.81	0.85	0.77	0.61	0.81	0.89	0.871	0.870
RF	1.00	0.92	1.00	0.71	1.00	0.93	0.999	0.917
GBDT	0.93	0.86	0.92	0.64	0.93	0.89	0.984	0.925
LightGBM	1.00	0.89	1.00	0.68	1.00	0.92	1.00	0.930
XGBoost	1.00	0.89	1.00	0.68	1.00	0.92	1.00	0.925

表 3-14 样本数分别为 2000:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.88	1.00	0.64	1.00	0.90	1.00	0.773
Logistic	0.82	0.85	0.76	0.61	0.83	0.89	0.869	0.869
RF	1.00	0.94	1.00	0.74	1.00	0.95	0.999	0.922
GBDT	0.92	0.87	0.90	0.66	0.92	0.90	0.979	0.926
LightGBM	1.00	0.91	1.00	0.71	1.00	0.93	1.00	0.931
XGBoost	1.00	0.91	1.00	0.70	1.00	0.93	1.00	0.933

表 3-15 样本数分别为 2500:595

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.90	1.00	0.66	1.00	0.92	1.00	0.758
Logistic	0.83	0.85	0.75	0.62	0.84	0.89	0.868	0.872
RF	1.00	0.95	1.00	0.77	1.00	0.96	1.00	0.919
GBDT	0.91	0.87	0.88	0.65	0.92	0.90	0.976	0.924
LightGBM	1.00	0.92	1.00	0.71	1.00	0.93	1.00	0.931

XGBoost	1.00	0.92	1.00	0.70	1.00	0.93	1.00	0.931

结合了手动欠采样与少数类加权/代价敏感的分类器在测试集上的性能并没有太多的提升。该方法在之后将被舍弃。

- SMOTE 合成少数类样本

SMOTE（Synthetic Minority Over-sampling Technique）是一种流行的数据过采样技术，用于解决分类问题中的类别不平衡问题。它通过创建合成的少数类样本来增加少数类别的代表性，从而帮助改善模型对少数类的识别能力。

以下是 6 种常见的 smote 合成少数类样本的方法，合成样本数量为多数类样本的 50%。

表 3-16 Borderline-SMOTE

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.92	1.00	0.63	1.00	0.92	1.00	0.663
Logistic	0.83	0.91	0.80	0.66	0.83	0.92	0.919	0.811
RF	1.00	0.95	1.00	0.73	1.00	0.95	1.00	0.913
GBDT	0.95	0.94	0.94	0.71	0.95	0.94	0.988	0.893
LightGBM	0.99	0.95	0.99	0.73	0.99	0.95	0.999	0.915
XGBoost	1.00	0.95	1.00	0.73	1.00	0.95	0.999	0.913

表 3-17 ADASYN

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.91	1.00	0.66	1.00	0.92	1.00	0.731
Logistic	0.82	0.90	0.78	0.64	0.81	0.92	0.888	0.799
RF	1.00	0.95	1.00	0.73	1.00	0.95	1.00	0.918
GBDT	0.94	0.94	0.93	0.70	0.94	0.94	0.983	0.876
LightGBM	0.99	0.95	0.99	0.72	0.99	0.95	0.999	0.915
XGBoost	1.00	0.95	1.00	0.72	1.00	0.95	1.00	0.911

表 3-18 SVMSMOTE

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.93	1.00	0.67	1.00	0.93	1.00	0.702
Logistic	0.87	0.93	0.85	0.68	0.87	0.93	0.935	0.815
RF	1.00	0.96	1.00	0.74	1.00	0.95	1.00	0.914
GBDT	0.95	0.94	0.95	0.74	0.95	0.95	0.988	0.899
LightGBM	0.99	0.95	0.99	0.74	0.99	0.95	0.999	**0.923**
XGBoost	1.00	0.96	1.00	0.75	1.00	0.95	1.00	**0.923**

表 3-19 K-Means SMOTE

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.93	1.00	0.68	1.00	0.94	1.00	0.696
Logistic	0.97	0.96	0.97	0.71	0.97	0.95	0.977	0.800
RF	1.00	0.96	1.00	0.73	1.00	0.96	1.00	0.907
GBDT	0.98	0.96	0.97	0.73	0.98	0.96	0.994	0.910
LightGBM	1.00	0.96	1.00	0.74	1.00	0.96	0.999	**0.928**
XGBoost	1.00	0.96	1.00	0.75	1.00	0.96	1.00	0.925

表 3-20 SMOTE-ENN

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.90	1.00	0.64	1.00	0.91	1.00	0.726
Logistic	0.86	0.90	0.84	0.65	0.86	0.91	0.930	0.820
RF	1.00	0.94	1.00	0.71	1.00	0.94	1.00	**0.922**
GBDT	0.96	0.92	0.95	0.69	0.96	0.93	0.992	0.899
LightGBM	1.00	0.94	1.00	0.72	1.00	0.94	0.999	0.919
XGBoost	1.00	0.94	1.00	0.73	1.00	0.94	0.999	0.917

表 3-21 SMOTE-Tomek

分类器/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.92	1.00	0.66	1.00	0.93	1.00	0.715
Logistic	0.84	0.92	0.81	0.67	0.83	0.93	0.909	0.805

RF	1.00	0.95	1.00	0.74	1.00	0.95	1.00	0.919
GBDT	0.94	0.94	0.94	0.72	0.94	0.94	0.986	0.890
LightGBM	0.99	0.95	0.99	0.73	0.99	0.95	0.999	0.918
XGBoost	1.00	0.95	1.00	0.74	1.00	0.95	1.00	0.909

以上评估了 6 种 smote 变体合成少数类样本的方法，从中可以看到大多数方法并不能明显提升模型性能，仅有 SMOTE-ENN 方法中的 RF 分类器有着些许提高，大约为 0.005。模型性能没有明显提高的原因很可能在于 smote 合成样本会引入大量噪声。

3. 构建特征，使用新的数据集重新评估每个分类器。逻辑回归和隔离森林额外使用带有数据分箱和多项式的数据集。本项目构建特征主要是为了降低模型复杂度，使用更少的特征构建模型；提高模型在未知数据集上的稳定性。

表 3-22 新数据集分类器评估

分类器/评估	**准确率**		**宏平均 F1**		**加权平均 F1**		**AUC 分数**	
	训练	测试	训练	测试	训练	测试	训练	测试
Decision Tree	1.00	0.94	1.00	0.69	1.00	0.94	1.00	0.697
Logistic	0.97	0.96	0.81	0.73	0.97	0.95	0.947	0.850
RF	1.00	0.96	1.00	0.74	1.00	0.96	1.00	0.909
GBDT	0.97	0.96	0.82	0.73	0.97	0.95	0.964	0.915
LightGBM	1.00	0.96	1.00	0.71	1.00	0.95	1.00	0.927
XGBoost	1.00	0.96	1.00	0.71	1.00	0.95	1.00	0.925
ISO_Forest	0.94	0.94	0.68	0.68	0.94	0.94	NAN	

注意：逻辑回归（Logistic）在本次实验中没有收敛。

在使用构建的特征的新数据集后，各个分类器的模型的性能只有 GBDT 与 Logistic 有些许的提升。

4. 模型训练时长

分类器较快的训练速度能够保持迭代更新的优势，在较大的数据集上该优势会进一步增加。

表 3-23 模型效率

分类器/时长	测试一	测试二	测试三	平均时长
Decision Tree	2.87s	2.66s	2.84s	2.79s
Logistic	0.51s	0.57s	0.60s	0.56s
RF	11.27s	17.07s	18.42	15.58
GBDT	46.18s	47.57s	51.56s	48.43s
LightGBM	1.50s	1.43s	1.50s	1.47s
XGBoost	1.97s	1.87s	1.94s	1.92s

5. 模型选择与方法选择

表 3-24 模型选择

分类器	使用/弃用原因
Decision Tree	使用：单颗决策树在测试集上表现效果不佳，但训练集上效果较好，后续通过集成多个决策树来增强泛化能力；效率较高
Logistic	弃用：在训练集和测试集上表现不佳，在使用构建的非线性特征后仍表现不足
RF	使用：随机森林的每棵树独立训练，并且使用随机的样本，这能够有效降低方差。后续能够考虑与基于 boosting 的算法相结合。
GBDT	弃用：传统的梯度提升方法，重点在于减少模型的偏差，但在大规模数据集上表现不佳且且效率很低
LightGBM	使用：在训练集和测试集上表现最好，能够有效降低模型偏差；使用基于梯度的单边采样和互斥特征捆绑，效率高并降低了内存的使用；后续能够考虑与基于 bagging 的算法相结合
XGBoost	弃用：与 LightGBM 性能表现类似，但始终略低于 LightGBM 且效率相较于 LightGBM 略差。考虑到二者均使用了梯度提升与 boosting 方式，故不再使用。
MLP	弃用：相较于基于树的分类器表现不佳、调整困难、效率低、可解释性不高
ISO_Forest	弃用：无监督学习算法、在训练集和测试集上均表现不佳

表 3-25 类不平衡处理方式选择

类不平衡处理方法	使用/弃用原因
手动欠采样	欠采样比例为 2000:595 时效果较好，AUC 分数较高，其余比例不再使用。保留使用方法，以便在增加数据后进一步评估
过采样-SMOTE	引入大量噪声，导致模型在未知数据集上表现不佳，保留使用方法，以便在增加少数类数据后进一步评估
少数类加权/代价敏感	模型性能没有提升，不再使用，但保留使用方法，以便在增加数据后能够进一步评估

3.3.2 模型融合策略

1. 模型融合的优势：

1.）性能提升: 融合多个模型可以改善预测性能，通常能够达到比任何单一模型都要好的结果。

2）降低过拟合风险: 由于模型融合结合了多个模型的预测，它可以减少过拟合的风险，因为不太可能所有模型都会在相同的数据点上过拟合。

3）增加稳定性: 融合多个模型可以增加预测的稳定性和鲁棒性，尤其是在数据可能有噪声或变动较大的情况下。

4）利用不同模型的优势: 不同的模型可能会在数据的不同方面表现出优势。通过融合，可以结合它们的优点，达到更全面的预测效果。

2. 模型融合方式介绍：

1）简单平均融合 (Simple Averaging):

这是最直接的模型融合方法之一，它简单地取多个模型的预测结果的算术平均值作为最终预测。对于回归问题，这意味着对于每个实例，所有模型预测的目标值取平均；对于分类问题，可以对概率预测取平均。

2）加权平均融合 (Weighted Averaging):

与简单平均类似，不同之处在于每个模型的预测被赋予不同的权重。权重通常基于模型的性能（如交叉验证得分），性能更好的模型会有更高的权重。这可以提高融合模型的预测准确度，尤其是当某些模型明显优于其他模型时。本例中 RF 与 LGBM 权重为 0.4:0.6。

3）软投票 (Soft Voting):

在分类问题中，软投票是一种融合方法，其中每个模型为每个可能的输出类别提供概率预测。这些概率预测然后被平均（简单或加权），最终预测是具有最高平均概率的类别。

4）硬投票 (Hard Voting):

硬投票也是用于分类问题的融合策略。在这种方法中，每个模型做出一个最终预测（而不是概率），然后最终预测是根据“多数投票”确定的。即被大多数模型选择的类别成为最终预测结果。

5）多层感知机-堆叠 (Multilayer Perceptron - Stacking):

堆叠是一种更复杂的融合方法，它首先训练多个不同的模型，然后再训练一个新的模型（称为元学习器或堆叠器）来综合前一层模型的预测。多层感知机是一种用于堆叠的神经网络模型。它不是直接使用原始输入特征，而是使用前一层模型的输出作为输入。本例中一共四个隐藏层，每层神经元数目分别为 100、50、30、10。

3. 不同融合策略的结果：

此处选用 RF 的特征数量为 19，LightGBM 特征数量为 12。选用不同模型所得到的融合模型有着不同的性能，本例中尽可能兼顾特征数量与模型性能。

表 3-26 融合策略评估

融合策略		AUC 分数/宏平均 F1
原始模型	RF	0.92255
	LightGBM	0.92814
简单平均融合		**0.92917/0.74**
加权平均融合		**0.92910/0.73**
软投票		0.91768/0.74
硬投票		无 AUC/0.72
多层感知机-堆叠		0.91443/0.75

由测试结果可以看出，简单平均融合后的模型结合了随机森林与轻量型梯度提升机的优势，最终的性能有了略微的提高。本例中仅结合了两个模型，在日后的改进与迭代中可以尝试融合更多模型增强其表现。

3.4 参数优化

3.4.1 超参数调优方法

遗传算法（GA）常用于求解优化问题中的最优解，属于启发式算法。作为超参数调优方法，与其他常用方法（如网格搜索、随机搜索或贝叶斯优化）相比有着独特的优势：

1. 全局搜索能力：遗传算法通过模拟自然进化机制，能有效探索整个参数空间，有助于找到全局最优解，尤其在复杂或非线性的参数空间中表现突出。

2. 避免局部最优：通过交叉和变异操作，遗传算法能探索新的参数区域，从而减少陷入局部最优解的风险。

3. 并行性：遗传算法可以轻松地并行化处理，因为每个个体（参数组合）的评估是相互独立的。

4. 适应性强：适合处理有多种类型参数（如离散、连续）的复杂问题，以及当参数空间大或参数之间有复杂交互时。

对于医疗保险欺诈领域这种需要更好的超参数的情况下，遗传算法进行超参数调优相比常规的超参数调优方法效果要更好。

本项目中遗传算法主要流程：

1. 定义超参数空间：选择需要进行优化的超参数，为每一个超参数定义一个列表或范

围。

2. 初始化种群：每个个体代表一组超参数的组合。这些个体可以随机生成，其中每个超参数的值从其范围内随机选择（本项目中使用 50 个）。确定种群大小（即超参数组合）的数量。

3. 定义适应度函数：适应度函数用于评估每个个体（超参数组合）的性能。通过使用这些超参数在验证集上训练模型并评估其性能（如准确率、ROC AUC 等）来完成。采用 5 折分层交叉验证。

4. 应用遗传算法操作：

选择（Selection）：从当前种群中选择个体作为下一代的“父代”。选择通常基于适应度，适应度高的个体被选中的机会更大。

交叉（Crossover）：“父代”个体配对并交换它们的部分超参数，产生“子代”个体。这个过程模拟了生物遗传中的基因重组。交叉率为 0.8.

变异（Mutation）：以一定概率随机改变个体中的某些超参数值。变异引入新的遗传变异，有助于探索搜索空间。变异率为 0.2.

5. 精英保留策略：确保最优秀的个体被保留到下一代，避免优秀解决方案被随机操作破坏。最优选择个体使用锦标赛制度，保留最好的 5 个个体。

6. 迭代进化：重复选择、交叉和变异步骤多个代（迭代），种群逐渐演化，适应度通常会增加。迭代次数为 30。

7. 确定最佳超参数组合：经过多代迭代后，选择性能最好的个体，其代表的超参数组合被认为是最优的。

8. 在测试集上验证：使用最优超参数组合在测试集上进行最终验证。必要时进行调整：如果测试结果不理想，可能需要调整超参数空间或算法设置，并重新运行遗传算法

9. 将交叉变异、参数选择等信息保存在文件中。

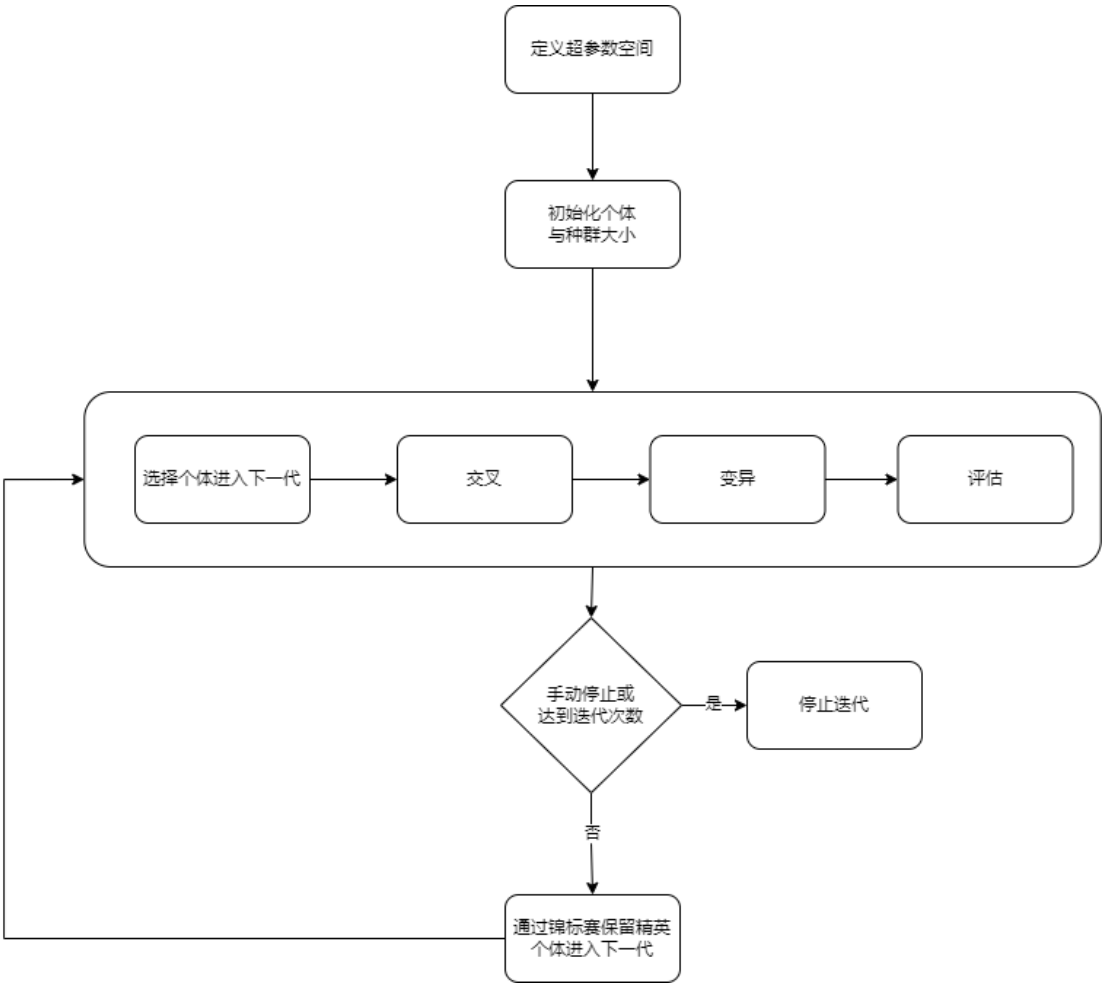

图 3-2 遗传算法简易流程图

决策树超参数调优示例：调整其最大树深、最小分割节点、最小分割样本

表 3-27 决策树超参数调优

超参数	取值/范围
max_depth	3, 5, 7, 9, 12, 15, 17, 25

min_samples_leaf	2, 5, 10, 15, 20
min_samples_split	1, 2, 5, 10

表 3-28 决策树超参数调优前后评估

调整/评估	准确率		宏平均 F1		加权平均 F1		AUC 分数	
	训练	测试	训练	测试	训练	测试	训练	测试
调整前	1.00	0.94	1.00	**0.69**	1.00	0.94	1.00	**0.710**
调整后	0.97	0.96	0.76	**0.74**	0.96	0.96	0.914	**0.838**

从结果中可以看到，经过超参数调优的决策树性能有着显著的提升。

3.5 模型评估与验证

3.5.1 性能评估指标

在类不平衡的数据集中，选择合适的评估指标尤为重要，因为传统的指标（如准确率）可能会产生误导性的结论。分类报告和 AUC 在这种情况下具有特定的优势：

1. 分类报告的优势

1）精细化的性能评估：分类报告提供了准确率、召回率和 F1 分数等指标，这些指标可以分别为每个类别计算。这在类不平衡的情况下特别有用，因为它可以揭示哪些类别被准确预测，哪些则没有。

2）平衡查全率和查准率：F1 分数作为查全率和查准率的调和平均数，有助于平衡这两个方面。在类不平衡的数据集中，某些类别的预测可能会倾向于多数类，但 F1 分数能够揭示出对少数类别的预测性能。

3）精确率（Precision）：对于给定的类别，准确率是指正确预测为该类别的样本数占预测为该类别总样本数的比例。公式为：

$$\text{精确率} = \frac{\text{真正例(TP)}}{\text{真正例(TP)} + \text{假正例(FP)}}$$

4）召回率（Recall)：也称为灵敏度，是正确预测为该类别的样本数占实际为该类别总样本数的比例。公式为：

$$\text{召回率} = \frac{\text{真正例(TP)}}{\text{真正例(TP)} + \text{假负例(FN)}}$$

5）F1 分数（F1 Score)：是准确率和召回率的调和平均数，用于综合考虑这两个指标。公式为：

$$\text{F1 分数} = 2 \times \frac{\text{精确率} \times \text{召回率}}{\text{精确率} + \text{召回率}}$$

2. AUC 的优势

1）阈值独立性：AUC 衡量的是分类器对于排序样本的能力，与阈值选择无关。这意味着无论是哪种分类阈值，AUC 提供了一个综合的性能评估。

2）对类不平衡敏感：AUC 通过考虑所有可能的分类阈值，提供了对于类不平衡敏感的性能度量。它能很好地展示分类器在区分不同类别时的整体能力。

3. 与其他常见评估方式的对比

1）准确率：仅仅度量了分类正确的样本比例，对于类不平衡的数据集来说，这可能产生误导性的高值。例如，如果一个类占总数据的 95%，一个始终预测这个多数类的简单模型也会有 95%的准确率。

2）混淆矩阵：提供了一个更细致的性能视图，包括真正例（TP)、假正例（FP)、真负例（TN）和假负例（FN)。然而，混淆矩阵本身并不提供一个综合的性能评估指标，需要进一步转化为如准确率、召回率等指标。

3）Kappa 统计量：尝试量化分类性能，超出了仅凭偶然性所期望的水平。但 Kappa 统计量在类不平衡的数据集中也可能受到限制，因为它依赖于所有类别的预测频率。

3.5.2 交叉验证与模型稳定性

交叉验证是一种评估机器学习模型泛化能力的方法，尤其适用于数据量不大的情况。我们使用交叉验证来评估模型在不同子集上的性能，从而确保模型稳定、不会过拟合且保证能够找到最好的超参数。

本项目中的医疗保险欺诈数据集少数类样本仅占 5%，故使用 5 折分层交叉验证，该方法将保持每个折叠中欺诈和非欺诈样本的比例，从而提供更可靠的模型性能评估。

3.6 模型的可解释性

在医疗保险欺诈检测项目中，模型的可解释性不仅增强了决策者对模型预测的理解，也是保障透明度和公正性的关键。通过详细解释模型的工作机制和决策依据，我们能够建立用户和管理者的信任，同时为模型的法律和道德合理性提供支持。

3.6.1 模型可解释性

在本项目中，我们使用了随机森林和轻量型梯度提升机作为主要的预测模型。为了提高这些模型的可解释性，我们采用了如 LIME（局部可解释模型-不透明预测）与 SHAP（SHapley Additive exPlanations）这类工具。SHAP 通过计算每个特征对每个预测的贡献值，提供了一个直观且强有力的解释框架。这样，我们不仅能够理解模型整体上是如何工作的，还能够解析出特定预测背后的具体驱动因素。

1. LIME 的核心思想是在模型做出预测的局部区域，近似这个复杂模型。简而言之，LIME 创建一个简单的模型（如线性模型），该模型只在模型预测附近的小区域内有效，

但足以解释此区域内的预测行为。

2. 部分依赖图（Partial Dependence Plot，PDP）展示了当保持其他特征固定时，单一特征值的变化如何影响模型的预测输出。

3. shap 值是一种理论上公平的分配方式，可以解释每个特征在模型预测中的贡献。对于一个机器学习模型，它考虑了所有可能的特征组合来评估每个特征对模型预测的贡献。换句话说，SHAP 值度量了如果没有这个特征，模型输出会发生多大的变化。

3.6.2 案例分析

3.6.2.1. LIME

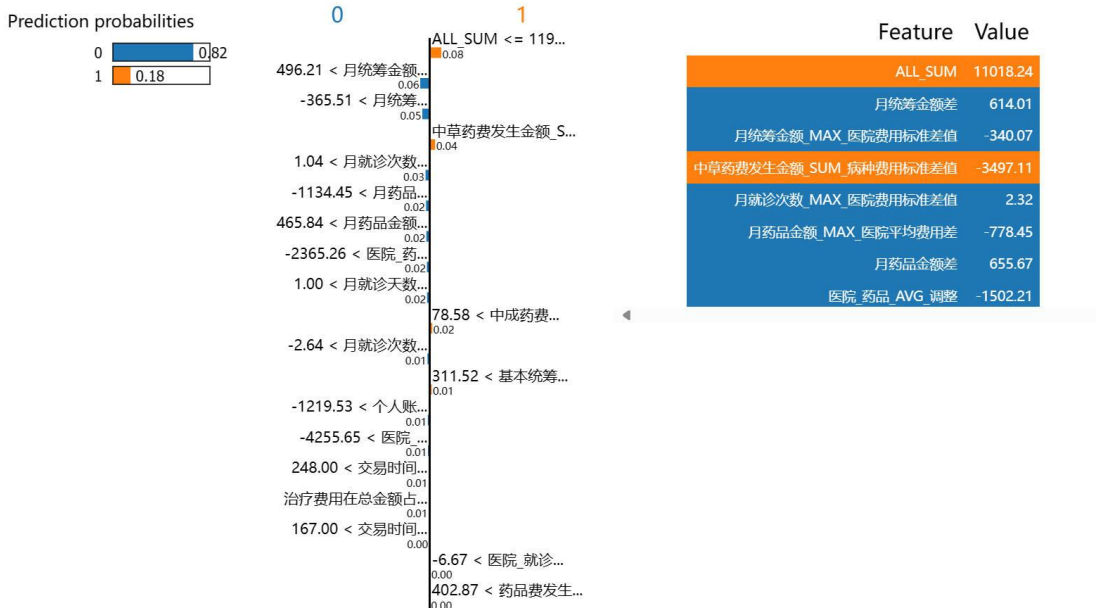

图 3-3 LIME 对样本 1 的预测结果解释

1）预测概率：图像左侧显示了模型为两个类别（标签 0 和标签 1）预测的概率。在这个案例中，模型预测实例属于类别 0 的概率为 0.82，属于类别 1 的概率为 0.18。这意味着根据模型，此实例被归类为类别 0（模型阈值为 0.5）。

2）特征贡献的可视化解释：图像右侧是一个条形图，展示了各个特征对预测结果的贡献度。在这个条形图中：

- 橙色条形代表推动模型预测向类别 1 倾斜的特征。
- 蓝色条形代表推动模型预测向类别 0 倾斜的特征。
- 每个条形的长度表示该特征贡献的绝对值大小。
- 条形右侧的数值代表该特征的实际数值。
- 例如，某特征“ALL SUM <= 119...”的橙色条形较长，意味着这个特征的数值将预测结果向类别 1 推进了较多，即对类别 1 预测的贡献比其他特征要大。而某些蓝色条形表示这些特征的值将预测结果向类别 0 推进，相对减少了模型预测实例为类别 1 的概率。

3）条形图中的具体特征和它们的值（例如，“月统筹金额_MAX_医院费用标准差值 <= -340.07”，“中草药费发生金额_SUM_病种费用标准差值 <= -3497.11”等）是模型输入的一部分。每个特征的贡献（显示为橙色或蓝色的长度）是通过比较有和没有该特征时模型输出的变化计算出来的。

4）从条形图可以看出，特征“ALL SUM”对类别 1 的预测有很大的正向贡献（推动概率更偏向类别 1），而“月统筹金额_MAX_医院费用标准差值”（降低了类别 1 的预测概率，增加了类别 0 的预测概率）。

3.6.2.2. PDP

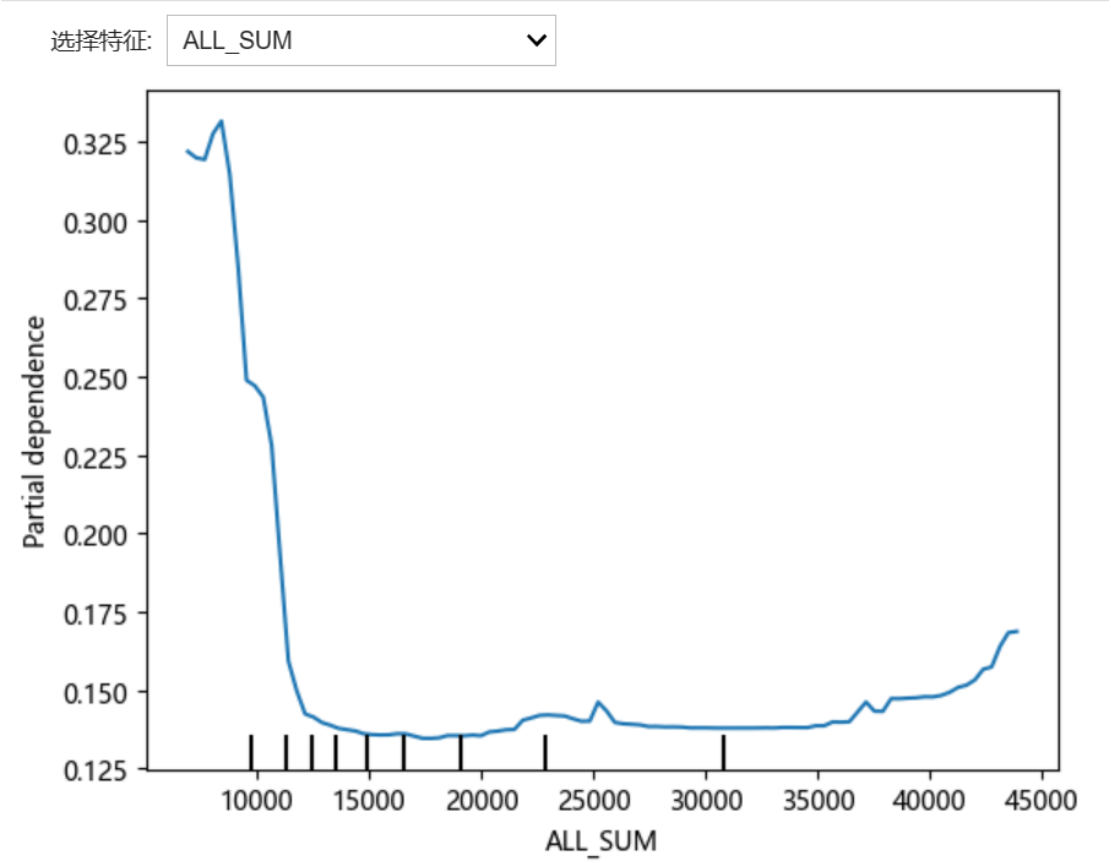

图 3-4 ALL_SUM 对模型的影响

1）在这个图中，横轴表示特征“ALL_SUM”的不同值，纵轴表示模型输出的部分依赖值，也就是预测值的平均变化。图中的曲线表明了“ALL_SUM”这个特征对模型预测结果的影响：

2）在“ALL_SUM”较低的值区间，部分依赖值相对较高。这表明在“ALL_SUM”较小时，模型预测输出会更高。这个高值可能代表了某个特定的事件或类别的概率增加。

3）随着“ALL_SUM”的增加，部分依赖值开始急剧下降，特别是在大约 10000 的值处，之后它逐渐趋于平稳。这表明在“ALL_SUM”增加到某个点后，它对模型输出的影响减弱，并且在某个范围内，进一步增加“ALL_SUM”对模型预测影响不大。

4）在“ALL_SUM”接近 40000 的值时，部分依赖值再次上升，这可能表明在这个特征值区间，模型预测输出又开始增加。

5）底部的垂直线表示样本分布，即“ALL_SUM”的某个值有多少数据点。垂直线越多，表示该值的数据点越密集。从图中可见，大多数数据集中在“ALL_SUM”较低的区域。

3.6.2.3. shap

1. 依赖图

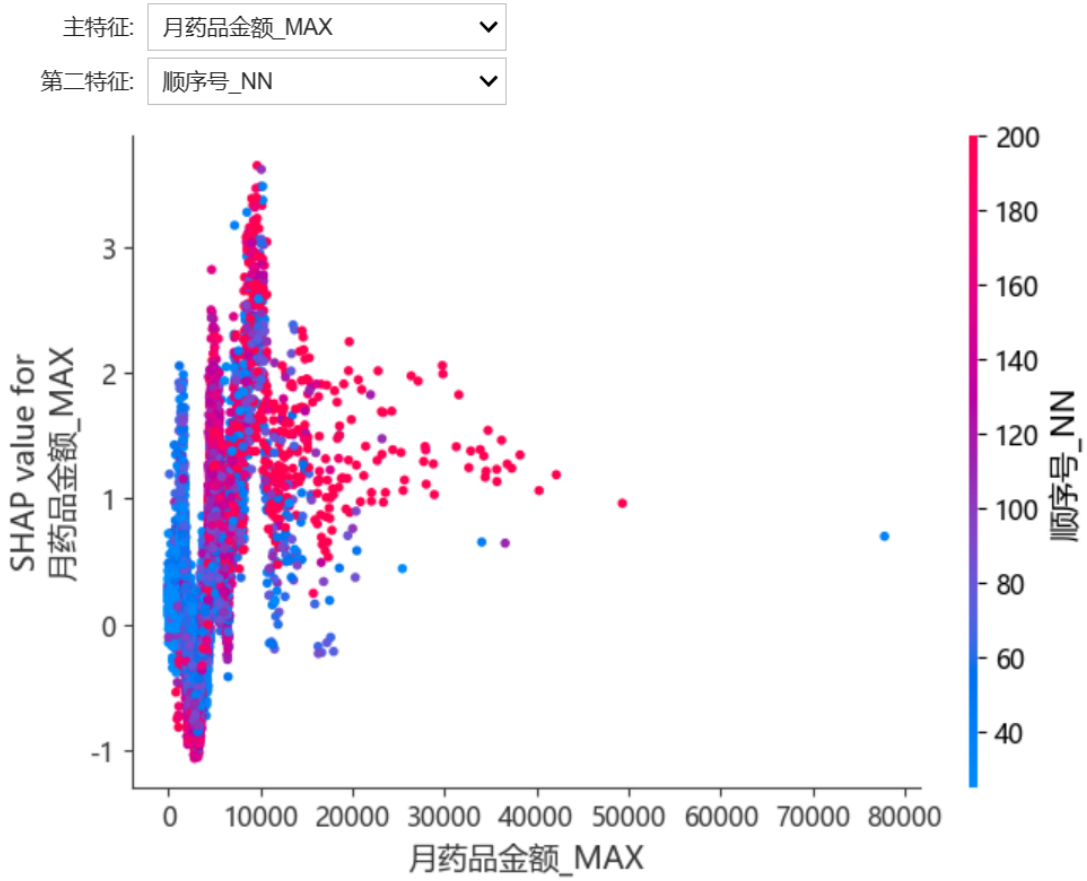

图 3-5 月药品金额_MAX 对模型影响与顺序号_NN 交互

1）图中显示了单个特征（在这个案例中为“月药品金额_MAX”）的值如何影响模型预测的 SHAP 值，即该特征对模型输出影响的大小和方向。在这张图中，颜色编码表

示了另一个特征（“顺序号_NN”）的值，提供了关于双特征之间相互作用的附加信息。

2）解释 SHAP 依赖图：

- 横轴：表示“月药品金额_MAX”特征的值。
- 纵轴：表示该特征对模型预测的 SHAP 值。
- 颜色：图中的点根据颜色的深浅表示“顺序号_NN”特征的值，颜色越深（蓝色到红色）表示“顺序号_NN”的值越高。

3）从这张图可以观察到以下几点：

- 对于“月药品金额_MAX”值较低的数据点，它们的 SHAP 值主要集中在 0 附近，这意味着这些值对模型输出的影响相对较小或者是中性的。
- 当“月药品金额_MAX”值增加时，SHAP 值也呈现出一定的增加趋势，表明这个特征值的增加对模型预测有正向的影响，即可能增加了模型预测为某个特定类别的概率。
- 在某些高“月药品金额_MAX”值的点上，SHAP 值有很大的波动，表明在这些值上模型输出的变化更加复杂，这可能是由于模型在这个区域捕捉到了更多的交互作用或者其他特征的影响。
- 可以观察到颜色的变化，颜色变化表示随着“顺序号_NN”值的增加，其对 SHAP 值的影响也在变化。如果颜色变化与 SHAP 值变化呈现出某种趋势，这可能表示这两个特征在模型中相互作用。

2. 总结图

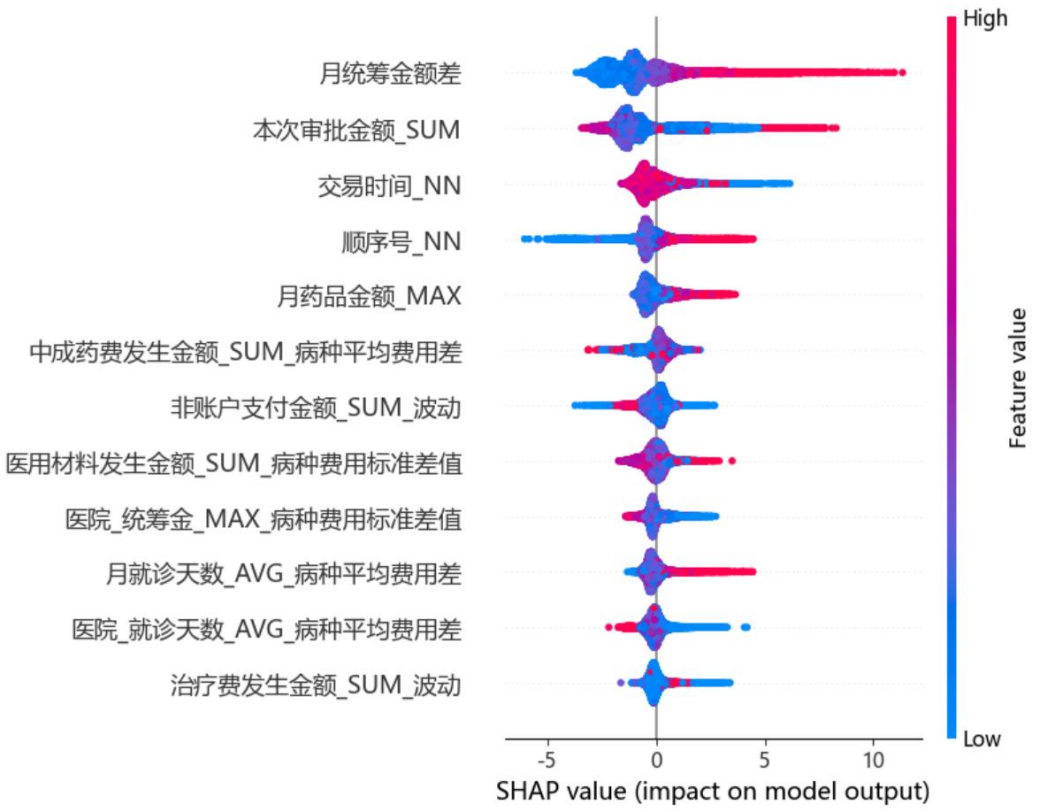

图 3-6 不同特征对模型类别影响

1）每个点代表一个数据点的 SHAP 值，它反映了该数据点的该特征对模型预测的影响。这些点的颜色代表特征的实际数值，从低（蓝色）到高（红色）。特征的实际数值是沿着颜色条的右侧垂直线显示的。

2）解释 SHAP 总结图：

- 横轴：表示 SHAP 值，即该特征如何影响模型输出。值可以是正的（推动模型预测向正类），也可以是负的（推动模型预测向负类）。
- 纵轴：列出了数据集中的各个特征，通常按照影响力大小排序。
- 例如，最上面的特征“月药品金额_MAX”有许多点在 SHAP 值正值区域，这意味着对于这个特征的高数值，它们推动了模型的预测向正类方向。相反，如果许多点在负值区域，就表明对于这个特征的低数值，它们将模型的预测推向

了负类方向。

3）分析总结图的关键点：

- 如果一个特征的点主要集中在 SHAP 值的正（或负）区域，表明这个特征通常增加（或减少）模型预测正类的概率。
- 如果一个特征的点在 SHAP 值 0 附近分布比较广，意味着它对模型预测的影响较小或有正有负，可能依赖于与其他特征的交互。
- 特征点在横轴上分布的宽度显示了这个特征对模型输出的影响的变异性。分布越宽，表明不同数据点上该特征的影响更加多变。
- 从图中可以看出，一些特征（如“顺序号_NN”和“月统筹金额差”）对模型输出的影响，它们的 SHAP 值分布宽度较大，并且多数集中在正值区域。

3. 力量图

图 3-7 模型对样本 1 的类别判断影响

1）力量图提供了对一个特定预测的解释，显示每个特征如何推动模型的输出从基线值变动到最终预测值。

2）解释 shap 力量图：

- 基线值（base value）：这是模型在没有任何特征输入时的平均预测输出，也就是预测的起点。
- f(x)：表示模型对特定实例的实际预测输出。

- 红色条：表示推动预测向较高值移动的特征贡献，即这些特征的存在提高了预测结果。
- 蓝色条：表示推动预测向较低值移动的特征贡献，即这些特征的存在降低了预测结果。
- 条的长度：表示特征贡献的大小。较长的条表示较大的影响力，较短的条表示较小的影响力。
- 条的标签：显示了特征的名称以及它们的实际值。

3）在图像中，可以看到“月药品金额_SUM”的高值显著推高了模型预测，而“顺序号_NN”和“本次审批金额_SUM”的一些值对预测的影响是负面的，即它们降低了模型的预测输出。图表显示了从模型预测的基线值到实际预测值的过程中，各个特征的正面或负面贡献。

4. 集合力量图

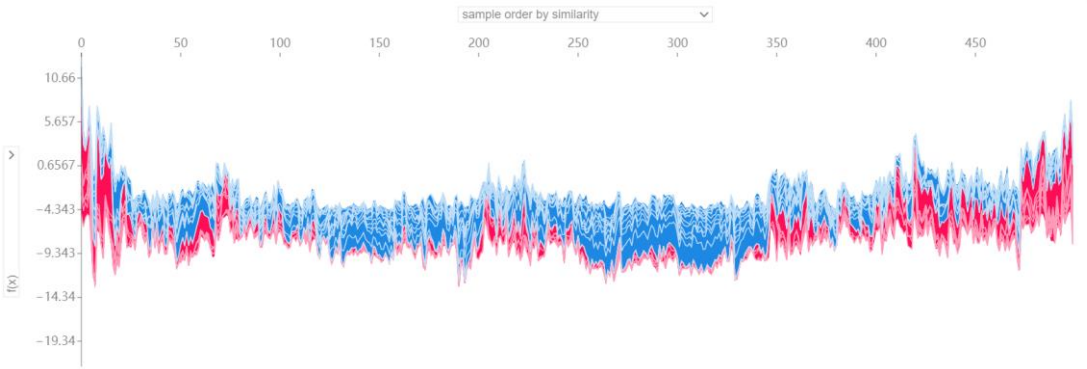

图 3-8 模型对前 500 样本预测结果堆叠

1）将多个单独的 SHAP 力量图进行堆叠与旋转，以便同时展示多个预测的特征影响。在这个图中，纵轴不再表示单一预测的特征贡献，而是所有样本的 SHAP 值在特定特征影响力度上的分布。这张图中展示了前 500 个样本

2）解释 shap 堆叠力量图：

- 横轴：表示模型预测的变化量（SHAP 值）。它显示了特征对模型预测输出的影响程度。
- 纵轴：现在代表的是样本的索引或类似性排序，可视为不同样本的集合。
- 颜色：通常表示 SHAP 值的密度；图中，红色代表更高的密度，而蓝色代表较低的密度。
- 显示所有样本中特征影响的变异性。揭示了哪些特征在推动模型预测时表现出更大的变异性（颜色条带宽度较大的区域），以及哪些特征对预测的影响比较稳定（颜色条带宽度较窄的区域）。

3）在这个具体的图中，可以看到 SHAP 值的变化区间相当广泛，表明特征对预测的影响在不同的样本之间有显著的差异。大量样本的 SHAP 值聚集在零附近，这可能表示对于这些样本，特征影响是中性的。同时，有些样本的特征影响力度很大，要么非常正（推动预测输出更高），要么非常负（推动预测输出更低）。

5. 决策图

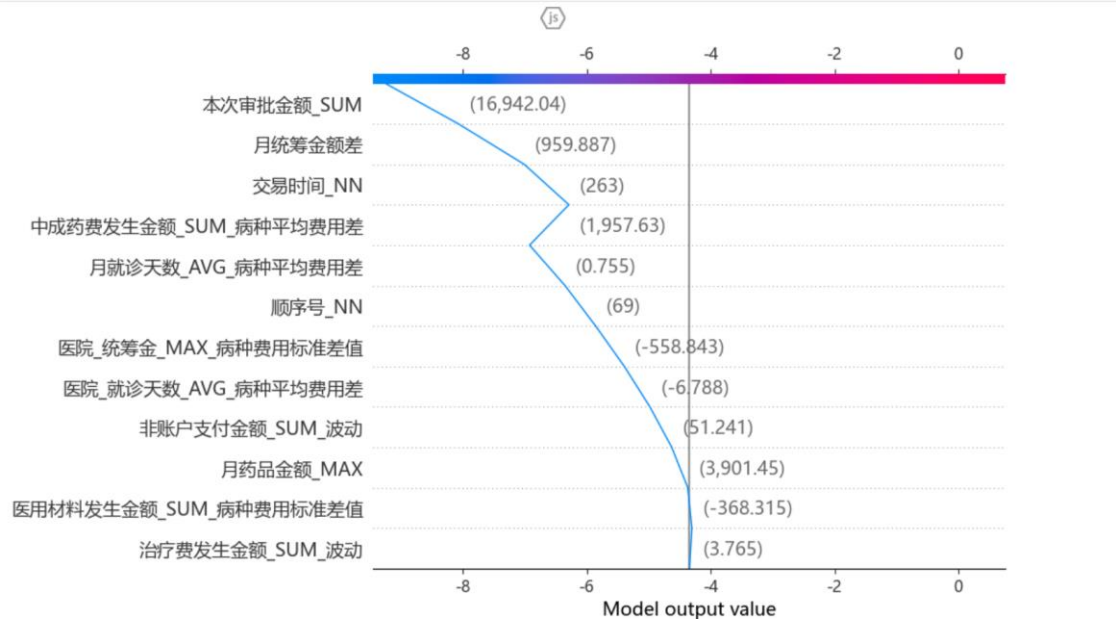

图 3-9 模型对样本 1 决策方式

1）决策图展示了一个单一预测的详细特征贡献。这种图展示如何从一个基线值（通常是数据集的平均预测值）通过各个特征的影响来达到最终的预测值。

2）解释 shap 决策图：

- 横轴：代表了模型的输出值。在这个例子中，向左的蓝色线表示负面贡献（即这些特征的值导致预测值降低），而向右的红色线表示正面贡献（即这些特征的值导致预测值升高）。
- 纵轴：列出了模型的特征。通常，这些特征会按照它们对预测结果的影响程度进行排序，最顶部的特征对预测结果有最大的影响。
- 线的长度：表示每个特征对预测值的实际贡献大小。线越长，该特征的影响越大。
- 点的位置：表示最终的预测输出值。

3）在图像中，可以看出“本次审批金额_SUM”特征对模型输出有很大的正向影响，

其余特征的贡献量较小，有的正向有的负向，最终这些贡献累积起来形成了最终的模型输出。

6. 条形图

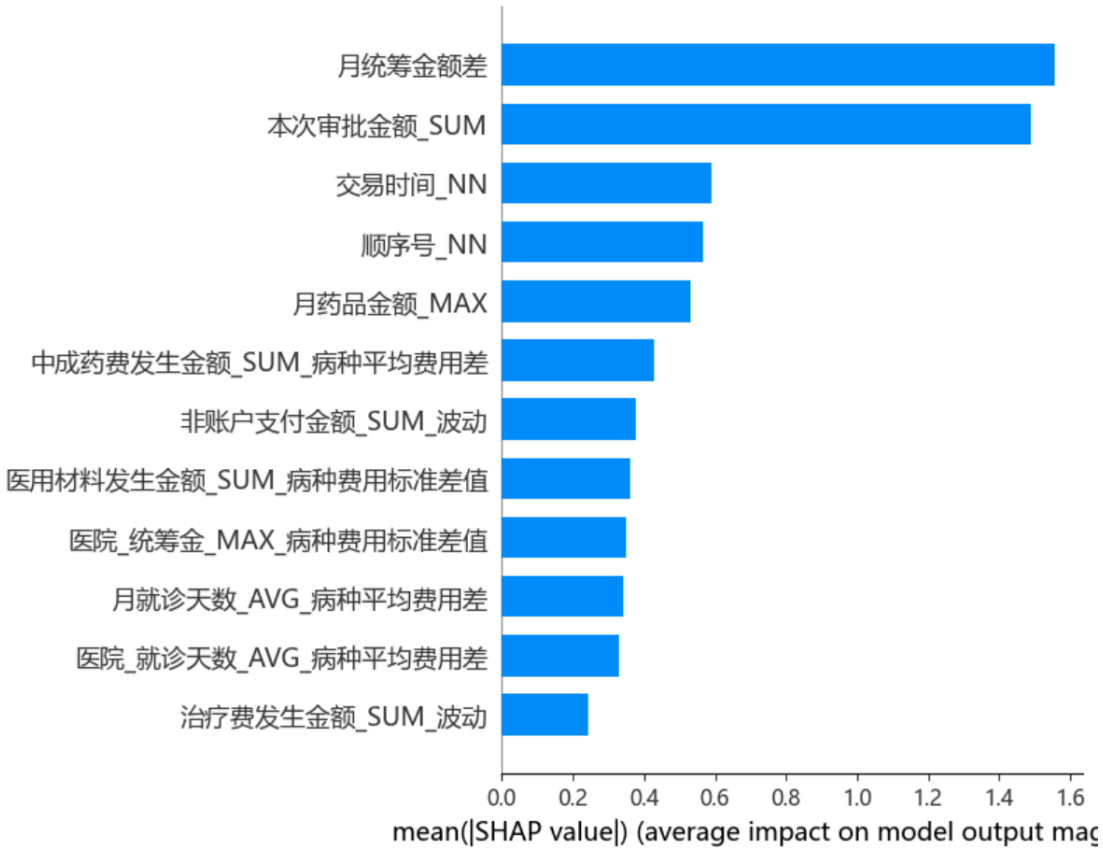

图 3-10 不同特征对模型影响直观显示

1）这张图用于表示特征在模型中的平均影响力。SHAP 值的条形图是理解模型决策最关键因素的一种直观方式。

2）解释 shap 条形图：

- 横轴：代表 SHAP 值的平均绝对值（或平均影响力），即特征对模型输出的平均影响大小。
- 纵轴：列出了特征名称。

- 在此图中，每个条形的长度代表了该特征对模型输出影响的平均大小。较长的条形表示该特征对模型输出有较大的平均影响，而较短的条形表示影响较小。

3）图中最长的条形代表“月统筹金额差”特征，这意味着它在所有特征中对模型输出的平均影响最大。其次是“本次审批金额_SUM”特征，以及“交易时间_NN”，依此类推。

4. 风险评估与管理

4.1 误识别风险

在医疗保险欺诈检测模型中，误识别风险主要表现为两种情况：误判为欺诈（假阳性）和漏判欺诈（假阴性）。这两种误识别对保险公司和参保者都会产生不同程度的负面影响。假阳性可能导致诚实参保者受到不公正对待，损害保险公司的声誉和客户关系；假阴性则会导致欺诈行为未能及时查处，造成经济损失。

4.2 模型的技术演变适应性

技术演变是不断进行的过程，新的算法和方法论不断出现。本项目的模型需要适应这种变化，以维持其准确性和效率。并定期评估是否有必要对模型进行更新或改进。例如探索并使用改进版的贪婪随机森林等等。

4.3 持续监控与性能评估

持续监控模型的运行状况和性能表现是重要的。使用实时监控工具来跟踪模型的输入输出数据质量、预测准确率、处理时间等关键性能指标。由于时间限制与数据集限制，我们未能开发出完整的监控系统。

4.4 风险缓解策略

- 数据质量管理：确保数据的准确性和时效性。定期清理和更新数据集，消除数据质量问题。进一步收集数据、建立更高效的模型。前期模型的快速迭代至关重要。
- 模型透明度与可解释性：进一步提高模型的透明度和可解释性，以便专业人员可以理解和验证模型的预测结果。
- 反馈机制：建立有效的反馈机制，以收集用户和专业人员对模型性能的反馈，及时发现并修正问题。
- 应急预案：针对可能的误识别和技术问题，制定应急预案和快速响应机制。

5. 项目成果与预期影响

5.1 模型性能与效果

1. 本项目最终通过简单平均融合随机森林与轻量型梯度提升机，模型的表现有了略微的提高。通过调节模型的阈值可以平衡精确率与召回率。这取决于在实践应用中倾向于识别出更多的欺诈者还是识别的欺诈者都准确。

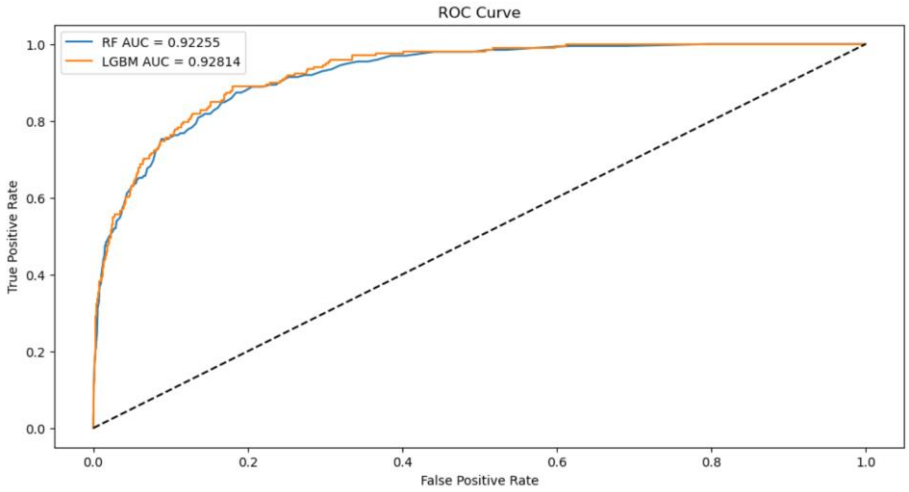

图 6-1 原始模型 AUC-ROC 变化曲线

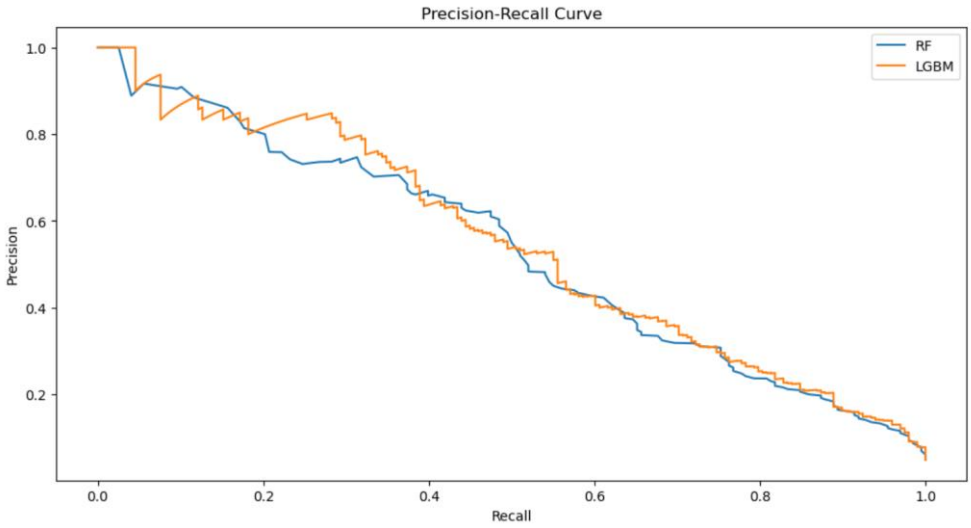

图 6-2 原始模型精确率与召回率变化曲线

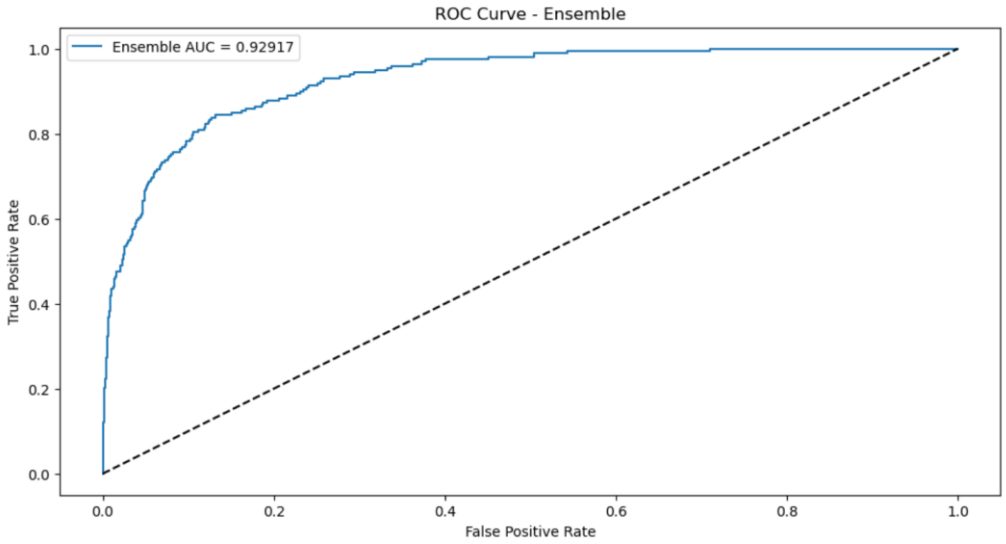

图 6-3 简单平均后模型 AUC-ROC 变化曲线

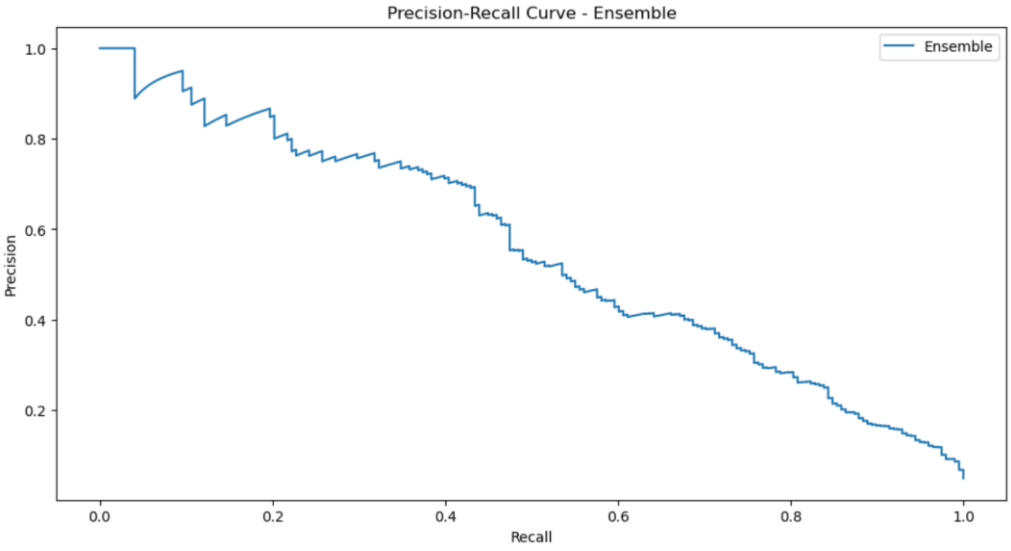

图 6-4 简单平均后模型精确率与召回率变化曲线

2. 在进行特征提取后，模型能够利用更少的特征因子来达到更好的性能。

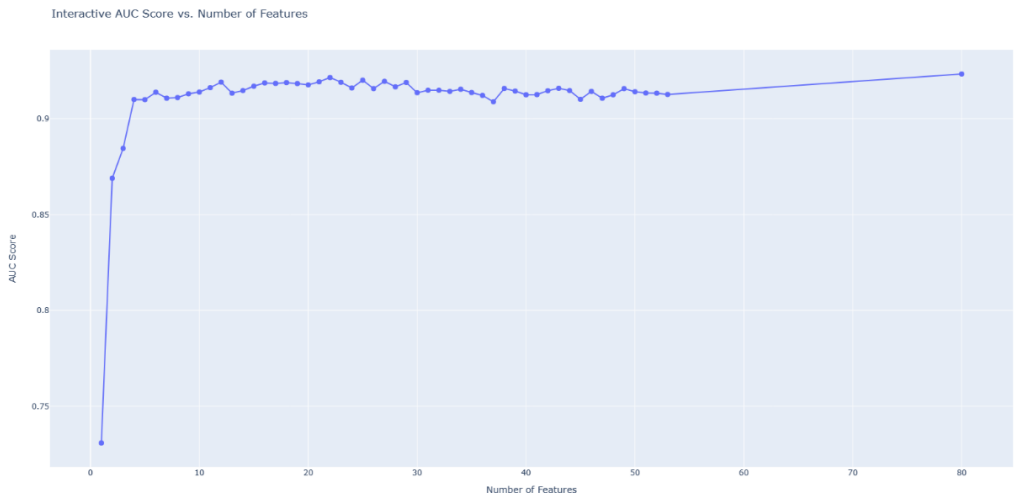

图 6-5 RF 原始数据集特征数量与 AUC 变化

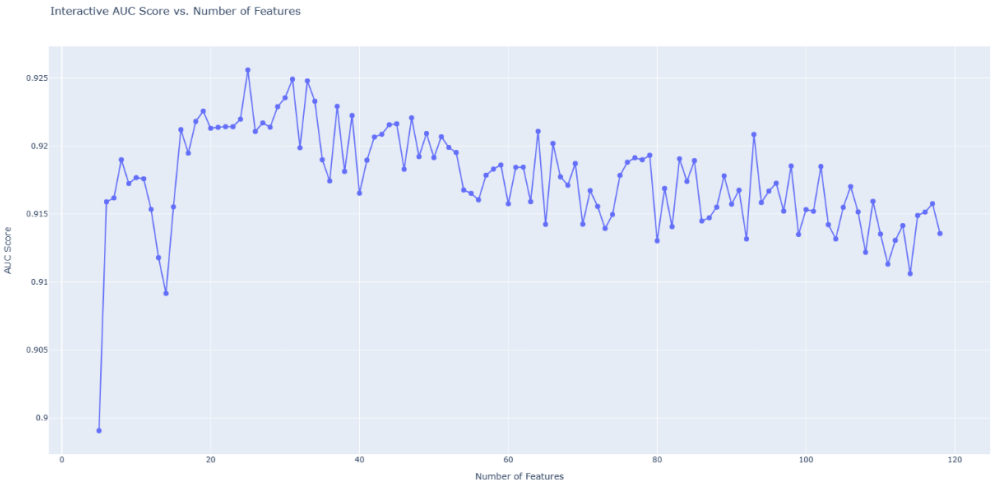

图 6-6 RF 新数据集特征数量与 AUC 变化

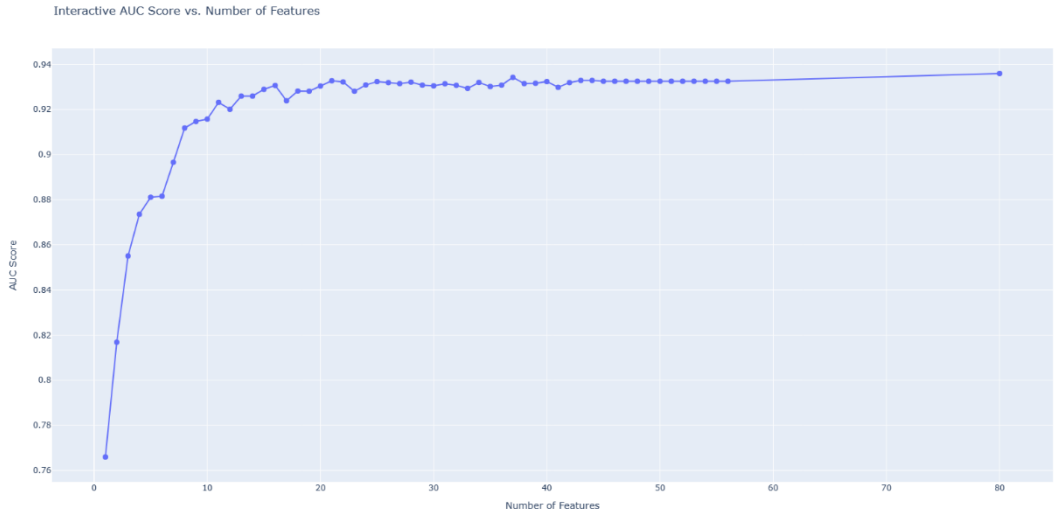

图 6-7 Lgb 原始数据集特征数量与 AUC 变化

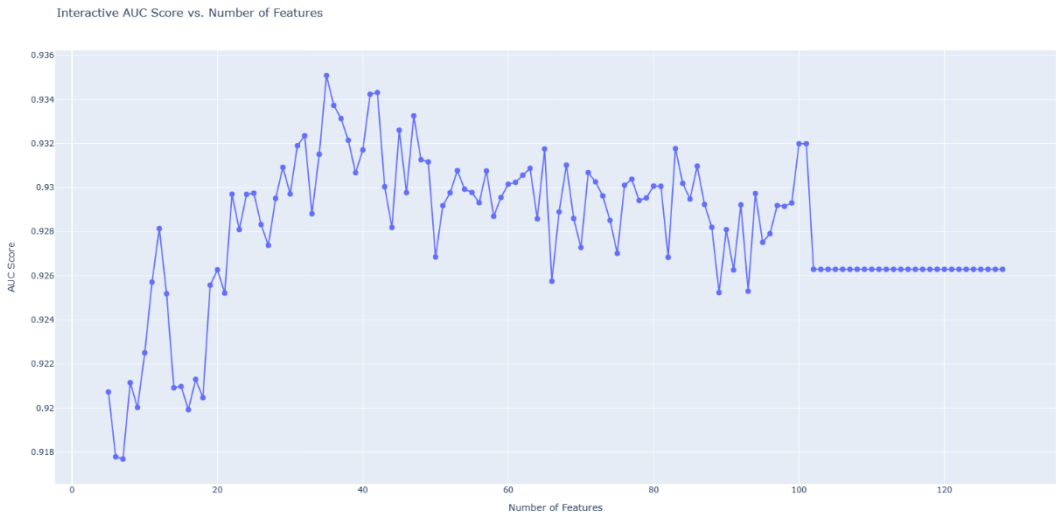

图 6-8 Lgb 新数据集特征数量与 AUC 变化

从结果中可以看到，通过特征选择能够使用更少的特征数量达到较好的模型性能。例如：

表 6-1 原始数据与新数据评估示例

分类器/评估	原始数据集		新数据集	
	特征数量	AUC 分数	特征数量	AUC 分数
RF	22	0.9215	19	0.9225
LGB	37	0.9342	35	0.9351

3. 通过 shap 库增加模型的可解释性，观察模型的决策方式，为监管人员提供搜集证据与判断的思路。

5.2 业务影响分析

增强风险控制：本项目实施的医疗保险欺诈检测模型将大幅提升保险公司在风险控制

方面的能力。通过精准识别欺诈行为，公司能够减少由欺诈引起的经济损失，增强风险预防和控制机制。

提升处理效率和客户满意度：自动化和智能化的检测流程将显著提高索赔审批的效率。这不仅减少了人工审核的工作量，也加速了合法索赔的处理速度，从而提升客户满意度。

改善决策制定：数据驱动的欺诈检测为公司提供了更深入的业务洞察，辅助管理层在产品设计、定价策略和风险管理等方面做出更为科学和精准的决策。

财务表现改善：减少欺诈损失和提升运营效率将直接反映在公司的财务表现上，提升利润率和市场竞争力。

5.3 社会价值与未来应用

1. 提升公平性和正义：有效的欺诈检测模型有助于保持医疗保险市场的公平性，确保诚实的参保者不会因欺诈者的行为而受到不公正的待遇或额外的财务负担。

2. 促进健康产业发展：减少医疗保险欺诈能够为整个健康产业创造一个更加健康和可持续的发展环境，有助于资源的合理配置和医疗服务质量的提升。

3. 社会教育和意识提升：此项目的成功实施和推广可作为提升公众对医疗保险欺诈危害性认知的有效工具，增强社会公众的法律意识和道德标准。

4. 未来应用前景：随着人工智能和大数据技术的不断进步，医疗保险欺诈检测模型的应用范围有望进一步扩大。未来，该技术可应用于其他类型的保险欺诈检测，甚至跨入医疗服务质量监控、药品监管等领域。

5. 促进科技创新：本项目的实施也将激励科技领域的创新。医疗保险欺诈检测领域的

成功案例可推动相似技术在其他领域的研究和应用，如金融欺诈监测、网络安全等。

6. 持续改进与未来方向

6.1 模型迭代与更新

1. 本项目最终选择的算法为随机森林(RF)与轻量型梯度提升机(LightGBM)，相比于神经网络，这两个模型能够进行快速调整与迭代，并且保持较高的性能与可解释性。

2. 在超参数调优方面，由于超参数调优需要大量的计算资源，尤其是在较高的超参数空间中，故本项目未对随机森林与轻量型梯度提升机进行超参数调优，而是选择使用了常见的超参数。但毫无疑问的是，使用本项目中给出的遗传算法进行超参数调优能够进一步提高模型的性能。

3. 在实际应用中，欺诈者的欺诈手段是不断变化的，保持较新的模型十分有必要。

6.2 数据集的扩展与多样化

1. 从最开始的数据分析中也不难发现，少数类样本稀少，通过手动欠采样来平衡数据集会使得多数类样本的中的信息丢失，但不平衡数据集会使得模型倾向于预测多数类。收集更多的欺诈样本将进一步提高模型性能。

2. 更加详细的数据同样有助于提高模型性能，例如将就诊人每一次的就诊行为记录在列、包括交易记录、金额费用等等。

3. 多样化的数据可以更好地识别欺诈行为，例如每个人的索赔单、就诊表、药品目录等文字数据。

6.3 新技术的探索与集成

1.本项目在最终的模型融合阶段，使用两个模型来进行融合。如果条件允许，可以尝试将更多的模型进行融合，例如使用原数据集训练的随机森林与使用新数据集训练的随机森林。，以此来观察模型的性能是否有所提升，并进一步增强模型的稳定性。

2. 使用分层交叉验证来评估每个算法，而不是仅仅使用测试集。这能够进一步评估模型在未知数据集上的泛化性能。也能够更加全面评估模型在类不平衡上的性能。

3. 尝试更高级的模型融合方式，一种可能的思路是，将原先的随机森林与轻量型梯度提升机所得的预测结果连同数据集一同放进分类器中进行训练，并将所得结果再次放入分类器中，循环训练并观察模型性能。

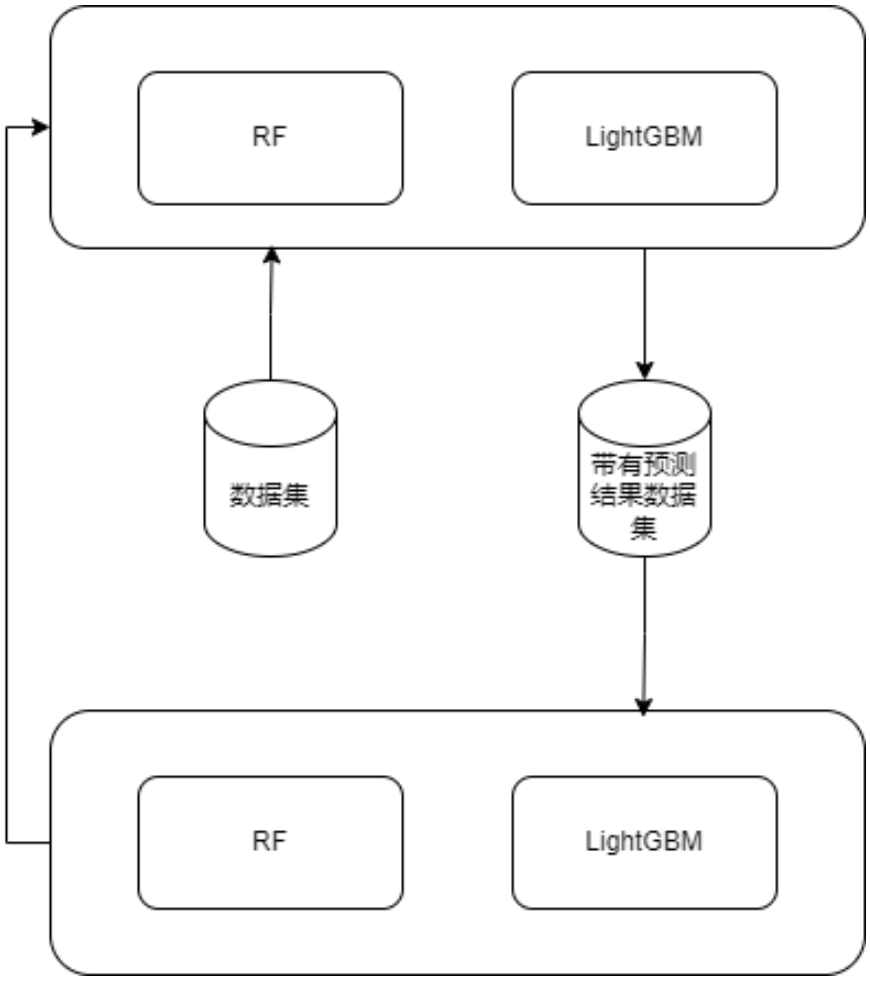

图 7-1 高级模型融合简易思路

4. 将项目由监督学习转为无监督学习。例如使用隔离森林与自编码器等算法。将无监督学习与有监督学习进行融合。

5. 我们在额外的资料中了解到，使用 GAN（对抗生成网络）合成结构化数据是一种新的思路，该方法通过编码器与解码器学习数据分布来生成与原始数据类似的数据点。该方式或许可以进一步解决类不平衡问题。

7. 总结与建议

7.1 项目总结

本项目成功开发了一个基于机器学习技术的医疗保险欺诈检测模型。通过深入分析医疗保险数据，结合先进的数据处理和机器学习方法，模型能有效识别出潜在的异常模式和欺诈行为。最终，选定的模型为随机森林(RF)和轻量级梯度提升机(LightGBM)，它们展现出了高效、稳定且具有较强可解释性的特点。对于其中存在的不足之处，在未来也将会进一步得到改进。

7.2 关键学习点

1. 数据处理与特征工程：成功处理不平衡数据集，通过精心设计的特征工程提取出关键信息，增强模型的性能与稳定性。

2. 模型选择和评估：深入比较多种算法，采用 AUC 分数与分类报告等方法全面评估模型性能，确保选出最适合的模型。

3. 特征选择：优化特征选择的方式，提高选择效率。降低特征因子集合能够进一步缩

小模型，提高预测速度，兼顾模型的性能与速度同样十分重要。值得注意的是，如果使用 PCA 等降维技术，将在很大程度上丧失模型的可解释性。

4. 模型的可解释性：重视模型的可解释性，确保其检测结果可以为专业人员提供有效信息，促进决策制定。本项目采用了较为成熟的方法来增加模型的可解释性。

5. 超参数调优：使用遗传算法及逆行超参数调优，确保能够找到更适合模型的超参数。

6. 模型融合：模型融合应遵循好而不同的原则，否则并不能增强模型的稳定性与泛化性。本项目最终采用简单平均进行融合，随机森林与轻量型梯度提升机通过不同的学习方式（bagging 集成与 boosting 集成）与学习不同的特征最终达到了类似的性能。

7.3 后续步骤与建议

1. 增加数据的多样性，例如就诊者每次的就诊时间、交易时间等等。

2. 在获取多样化的数据后（例如就诊时间、交易时间），可以尝试使用更强大的模型，例如使用 Transform 等神经网络，这类神经网络能够处理更好地处理时间序列，尽管这会降低一定程度上的迭代效率和可解释性，但如果追求高精确率与高召回率，使用神经网络等算法或许能够提供更好的帮助。

3. 混合有监督与无监督学习，通过收集反馈数据，训练强化学习模型，构建更强大的混合专家模型。

4. 部署模型，构建实时监管系统。由于时间限制，我们未能构建出完整的实时监测系统。我们尝试构建的系统中主要包含以下方面：

1）数据分析模块：用于查看当前数据相对于之前的数据其分布有无明显的变化，这将帮助做出是否更新模型的决定。

2）实时监控：能够利用就诊人当前的数据进行预测，判断是否存在欺诈。其中包含模型的决策路径。

3）反馈页面：模型预测结果与调查结果相比较，存储到数据库中，用于更新模型。

8. 附录

8.1 主要数据和代码文件说明

1. 数据文件：

process_data.csv：经处理后的原始数据集文件，仅填充了空值。

new_features.csv：经过特征提取后的数据集文件，包含原始特征与构建特征。

new_features_data_2.csv：在 new_features.csv 的基础上额外添加了多项式特征与分箱特征。

2. 代码脚本文件：

Lgbm_undersampling.py：轻量型梯度提升机模型脚本，采用手动欠采样、减少特征数量并保存模型。

Rf_undersampling.py：与 Lgbm_undersampling.py 类似，不过模型换为了随机森林模型。

Xgboost_undersampling.py：与 Lgbm_undersampling.py 类似，不过模型换为了极限梯度提升模型。

Auc_feature.py：打印所保存的模型的文件夹下的特征数目与 AUC 变化曲线

GA.py：使用遗传算法来选取超参数。

3. 代码笔记本文件：

Feature_Engineering.ipynb：特征工程，用于特征提取。

ensemble-decision-tree.ipynb：采用简单平均法集成单颗决策树，思路来源于随机森林。

explain.ipynb：对随机森林与轻量型梯度提升机进行模型解释。

LightGBM.ipynb：功能大致与脚本文件类似。

RF.ipynb：功能大致与脚本文件类似。

model_demo_1_3.ipynb：模型评估与选择，方法选择。测试不同分类器与采样策略效果。

model_demo_2.ipynb：与 model_demo_1_3.ipynb 文件类似。

model_demo_3.ipynb：与 model_demo_1_3.ipynb 文件类似。

model_demo_time.ipynb：与 model_demo_1_3.ipynb 文件类似，用于评估模型效率。

RF_LGB.ipynb：随机森林与轻量型梯度提升机融合策略。

数据分析.ipynb：用于初步分析数据。

数据分析_2.ipynb：用于初步分析数据，增加了交互窗口与多项式决策边界等。

8.2 参考资料

1. Kate, P., Ravi, V., & Gangwar, A. (2024 年). 利用机器学习检测医疗保险索赔中的欺诈行为. 《银行与保险分析中心研究》.

2. Patience Chew Yee Cheah, Yue Yang, 和 Boon Giin Lee。2023。《通过混合 SMOTE-GAN 技术提高金融欺诈检测》。《国际金融研究杂志》11: 110。网址：https://doi.org/10.3390/ijfs11030110。

3. 《利用数据分析检测医疗行业欺诈》。讨论白皮书。

4.作者未知。《IEEE TSMC 部分 B 探索性欠采样用于类不平衡学习》。IEEE Transactions on Systems, Man, and Cybernetics, Part B。

5. Jing Li & Kuei-Ying Huang & Jionghua Jin & Jianjun Shi。2007 年。《关于医疗保健行业欺诈检测的统计方法调查》。医疗保健行业欺诈检测调查报告。

6. Satyendra Singh Rawat 和 Amit Kumar Mishra。《处理分类问题中类别不平衡的方法综述》。Amity University, Gwalior, India。电子邮件：satyendra.rawat@s.amity.edu; akmishra1@gwa.amity.edu。

7. Prateek Kate, Vadlamani Ravi, 和 Akhilesh Gangwar。《利用机器学习检测医疗保险索赔中的欺诈行为》。《分析客户关系管理在银行和保险业的应用》。Hyderabad, India: Institute for Development and Research in Banking Technology。

8. Lin-Lin Wang, Na Hyun Jo , Brinda Kuthanazhi, Yun Wu, Robert J. McQueeney, Adam Kaminski, and Paul C. Canfield Ames Laboratory, U.S. Department of Energy, Ames, IA 50011, USA Department of Physics and Astronomy, Iowa State University, Ames, IA 50011, USA《FinGAN: 用于银行和保险业客户关系分析管理的生成对抗网络》。

9. Abed Mutemi 和 Fernando Bacao。2023 年《用于检测零售欺诈的基于数字的机器学习设计》。地址：https://doi.org/10.1038/s41598-023-38304-5

10. Justin M. Johnson 和 Taghi M. Khoshgoftaar。2019-6-27。《类别不平衡的深度学习》。地址：https://doi.org/10.1186/s40537-019-0192-5

11. Lei Xu 和 Kalyan Veeramachaneni。2018 年。《使用生成对抗网络合成表格数据》。MIT LIDS, Cambridge, MA。地址：arXiv:1811.11264v1 [cs.LG]。