1．实验目的

（1）掌握sklearn中决策树方法的调用。

（2）掌握决策树方法基本调参技术。

（3）掌握基于决策树的组合方法，包括Adaboost和Random Forest，了解XGBoost和LightGBM

（4）能够根据准确率比较、评价不同的分类模型。

2．任务描述

2.1勒索软件识别

勒索软件（Ransomware）是一类恶意软件，其主要目的是通过加密受害者的文件或系统，然后要求受害者支付赎金以获取解密密钥，从而恢复其被锁定的数据和系统。勒索软件在当今数字时代威胁着个人、企业和组织的信息安全，因为它们可能导致严重的数据损失、业务中断以及隐私侵犯。而针对勒索软件的识别，能够使得受害者保护个人数据的安全、避免金钱损失、防止业务中断以及保护个人隐私。因此在勒索软件持续威胁信息安全的背景下，能够实现对勒索软件的准确识别具有十分重要的意义。

2.2数据集

本次使用的数据集为勒索软件数据集Ransomware.csv，该数据集中包含138047条数据，每条数据包含56个特征以及一个名为“legitimate”的标签，用于指示是否为合法（legitimate）的样本（其中1为合法，0为非法）。利用该数据集能够训练勒索软件识别的模型。

以下是关于数据集中特征的介绍：

表1 数据集特征介绍

|  |  |
| --- | --- |
| 特征 | 说明 |
| Name | 文件名，表示样本的文件名称 |
| md5 | 文件的MD5散列值，用于唯一标识文件 |
| Machine | 目标机器类型，表示为一个数字代码，指示文件可以在哪种类型的机器上运行 |
| SizeOfOptionalHeader | 可选头的大小，表示PE文件头中可选头的字节数 |
| Characteristics | 文件特性，表示为一个数字代码，指示文件的一些特性 |
| MajorLinkerVersion | 主链接器版本，用于标识文件的主要链接器版本号 |
| MinorLinkerVersion | 次链接器版本，用于标识文件的次要链接器版本号 |
| SizeOfCode | 代码段大小，表示PE文件中代码段的大小 |
| SizeOfInitializedData | 已初始化数据段大小，表示PE文件中已初始化数据段的大小 |
| SizeOfUninitializedData | 未初始化数据段大小，表示PE文件中未初始化数据段的大小 |
| AddressOfEntryPoint | 入口点地址，表示PE文件的程序入口点在内存中的地址 |
| BaseOfCode | 代码段基址，表示PE文件中代码段在内存中的起始地址 |
| BaseOfData | 数据段基址，表示PE文件中数据段在内存中的起始地址 |
| ImageBase | 映像基址，表示PE文件在内存中的基址 |
| SectionAlignment | 区段对齐，表示PE文件在内存中的区段对齐方式 |
| FileAlignment | 文件对齐，表示PE文件在磁盘上的文件对齐方式 |
| MajorOperatingSystemVersion | 主要操作系统版本，表示文件期望运行的主要操作系统版本 |
| MinorOperatingSystemVersion | 次要操作系统版本，表示文件期望运行的次要操作系统版本 |
| MajorImageVersion | 主要映像版本，表示文件的主要映像版本号 |
| MinorImageVersion | 次要映像版本，表示文件的次要映像版本号 |
| MajorSubsystemVersion | 主要子系统版本，表示文件期望运行的主要子系统版本 |
| MinorSubsystemVersion | 次要子系统版本，表示文件期望运行的次要子系统版本 |
| SizeOfImage | 映像大小，表示PE文件在内存中的映像大小 |
| SizeOfHeaders | 文件头大小，表示PE文件头的大小 |
| CheckSum | 校验和，表示PE文件的校验和 |
| Subsystem | 子系统，表示文件期望运行的子系统类型 |
| DllCharacteristics | DLL特性，表示动态链接库（DLL）的一些特性 |
| SizeOfStackReserve | 堆栈保留大小，表示程序堆栈的保留大小 |
| SizeOfStackCommit | 堆栈提交大小，表示程序堆栈的提交大小 |
| SizeOfHeapReserve | 堆保留大小，表示程序堆的保留大小 |
| SizeOfHeapCommit | 堆提交大小，表示程序堆的提交大小 |
| LoaderFlags | 装载器标志，表示PE文件的装载器标志 |
| NumberOfRvaAndSizes | 数据目录数目，表示PE文件中数据目录的数量 |
| SectionsNb | 区段数量，表示PE文件中的区段数量 |
| SectionsMeanEntropy | 区段熵均值，表示PE文件各区段熵的平均值 |
| SectionsMinEntropy | 区段熵最小值，表示PE文件各区段熵的最小值 |
| SectionsMaxEntropy | 区段熵最大值，表示PE文件各区段熵的最大值 |
| SectionsMeanRawsize | 区段原始大小均值，表示PE文件各区段原始大小的平均值 |
| SectionsMinRawsize | 区段原始大小最小值，表示PE文件各区段原始大小的最小值 |
| SectionMaxRawsize | 区段原始大小最大值，表示PE文件各区段原始大小的最大值 |
| SectionsMeanVirtualsize | 区段虚拟大小均值，表示PE文件各区段虚拟大小的平均值 |
| SectionsMinVirtualsize | 区段虚拟大小最小值，表示PE文件各区段虚拟大小的最小值 |
| SectionMaxVirtualsize | 区段虚拟大小最大值，表示PE文件各区段虚拟大小的最大值 |
| ImportsNbDLL | 导入的DLL数量，表示PE文件导入的DLL数量 |
| ImportsNb | 导入数量，表示PE文件的导入数量 |
| ImportsNbOrdinal | 导入的序数数量，表示PE文件导入的序数数量 |
| ExportNb | 导出数量，表示PE文件的导出数量 |
| ResourcesNb | 资源数量，表示PE文件中的资源数量 |
| ResourcesMeanEntropy | 资源熵均值，表示PE文件各资源熵的平均值 |
| ResourcesMinEntropy | 资源熵最小值，表示PE文件各资源熵的最小值 |
| ResourcesMaxEntropy | 资源熵最大值，表示PE文件各资源熵的最大值 |
| ResourcesMeanSize | 资源大小均值，表示PE文件各资源大小的平均值 |
| ResourcesMinSize | 资源大小最小值，表示PE文件各资源大小的最小值 |
| ResourcesMaxSize | 资源大小最大值，表示PE文件各资源大小的最大值 |
| LoadConfigurationSize | 装载配置大小，表示PE文件的装载配置大小 |
| VersionInformationSize | 版本信息大小，表示PE文件的版本信息大小 |

3．相关知识

评价指标

模型的评价指标是用来衡量机器学习或数据分析模型在特定任务上表现如何的一种方法。这些指标对于评估模型的性能、选择合适的模型、调整参数以及优化模型在现实世界中的应用都起着关键作用。

在分类问题常用的评价指标如下：

（1）准确率（Accuracy）：正确预测的样本数占总样本数的比例。

（2）精确率（Precision）：预测为正样本中实际为正样本的比例，衡量模型的预测准确性。

（3）召回率（Recall）：实际为正样本中被正确预测为正样本的比例，衡量模型对正样本的捕获能力。

（4）F1分数（F1 Score）：精确率和召回率的调和平均，综合考虑了预测准确性和正样本捕获能力。

（5）混淆矩阵（Confusion Matrix）：显示了模型预测结果与真实标签之间的关系，可用于计算上述指标。

4．实验要求和步骤

（1）查阅资料，撰写一段关于LightGBM算法的简介文字，列于实验结果中。要求图文并茂，注明引用的文献来源，不少于800字。

（2）完成头歌平台的实训作业2 【挑战】梯度提升：高性能的LightGBM，要求模型准确率高于预设值0.995.

（3）尝试调试LightGBM的参数，观察并记录参数变化对性能的影响。要求记录至少2个参数的调试结果，记录不同数值下获得的准确率，并进行简要分析。

5．实验结果

（1）LightGBM算法的简介

LightGBM 是光梯度增强机的缩写，是一个免费开源的分布式机器学习梯度增强框架，最初由 Microsoft 开发。它基于决策树算法，用于排序、分类和其他机器学习任务。开发重点是性能和可扩展性。

GBDT(Gradient Boosting Decision Tree)梯度提升决策树

弱分类器迭代训练得到最优模型，训练效果好，不易过拟合

XGBoost eXtreme GB

对所有特征都按照特征的数值进行预排序

花O(#data)找最佳分割点，将数据分裂

优点:精确找到分割点

缺点:慢，保存数据特征值，排序结果，消耗空间也大；在遍历每一个分割点时，都需要进行分裂增益计算

LightGBM(Light Gradient Boosting Machine)轻量梯度提升器

一个实现GBDT算法的框架，支持高效率的并行训练，并且具有更快的训练速度，更低的内存消耗，更好的准确率，支持分布式，快速处理海量数据等优点

1. 基于Histogram的决策树算法

离散化特征，存储方便，计算代价小

1. 单边梯度采样 GOSS
2. 互斥特征捆绑
3. 带深度限制的Leaf-wise的叶子生长策略

只找好叶子往下延申，缺点深度容易很深，过拟合

1. 直接支持类别特征(Categorical Feature)
2. 支持高效并行

数据并行

传统的数据并行策略主要为水平划分数据，让不同的机器先在本地构造直方图，然后进行全局的合并，最后在合并的直方图上面寻找最优分割点。这种数据划分有一个很大的缺点：通讯开销过大。

LightGBM在数据并行中使用分散规约 (Reduce scatter) 把直方图合并的任务分摊到不同的机器，降低通信和计算，并利用直方图做差，进一步减少了一半的通信量。

投票并行

基于投票的数据并行则进一步优化数据并行中的通信代价，使通信代价变成常数级别。在数据量很大的时候，使用投票并行的方式只合并部分特征的直方图从而达到降低通信量的目的，可以得到非常好的加速效果。

1. Cache命中率优化

首先，所有的特征都采用相同的方式获得梯度（区别于XGBoost的不同特征通过不同的索引获得梯度），只需要对梯度进行排序并可实现连续访问，大大提高了缓存命中率；

其次，因为不需要存储行索引到叶子索引的数组，降低了存储消耗，而且也不存在 Cache Miss的问题。

（2）LightGBM模型训练

（实现代码，及通过平台测试的截图）

n\_estimators=300

（3）LightGBM模型调参结果

（用表格、折线图，记录、比较至少2个参数，在不同取值下对性能的影响）

6．分析与讨论

（1）除实验中使用的参数外，LightGBM还有哪些参数，可以进行调节？试简单介绍这些参数。

（2）这个数据集中有哪些字段是相对重要的？如果剔除不重要的字段，是否可改善分类性能？