第九届“泰迪杯”数据挖掘挑战赛

作品单位：西南石油大学

作品成员：黄裕萌

指导老师：朱云华

基于数据挖掘的上市公司财务数据分析

摘要

由于信息不对称，隐蔽的上市公司财务数据造假及爆雷很难被预防和预测，一旦雷爆，有可能产生多米诺骨牌效应，严重损害投资者利益。本文尝试用机器学习，寻找各行业与财务数据造假相关的数据指标，并且使用挑选出来的指标进行机器学习预测得到第六年可能会发生造假的公司。

针对问题一，采用了基于惩罚项和基于树模型两种不同的嵌入法，前一种用了 LR、LASSO、SVM 三种模型，后一种用了 RF、GBDT、XGBoost、LightGBM 模型共七种来机器学习，求出对上市公司发生造假有相关性的特征。再根据特征重要性程度，得到每个算法排名的前 30 个指标，挑选出在所有算法中出现次数大于或等于 3 次的指标，作为与财务数据造假相关的数据指标。并以制造业和房地产业为例，对挑选出来的指标进行共性分析和差异性分析。

针对问题二，通过 SMOTE 采样解决不同行业的数据不平衡问题之后，利用 F1-score 和 AUC 指标，并基于 k 折交叉验证和网格搜索给第一问中效果较好的 LR、RF、GBDT、XGBoost、LightGBM 模型进行超参数调优。在调试好的机器模型的基础上，本文选择了基于 Stacking 集成学习把模型融合成一个集成分类预测模型，第 1 层基学习器选择 RF、GBDT、XGBoost、LightGBM 模型，第 2 层元学习器选择了 LR 模型，从而确定了最优的 Stacking 集成学习分类预测模型。Stacking 集成模型在测试集上的 F1-score 得分为 0.96，AUC 为 0.79，高于所有的基学习器，不存在过拟合现象并且预测效果良好，并求出第 6 年制造业的预测结果为有 20 家上市公司发生财务数据造假。

针对问题三，采用了第二问的方法对其他行业进行预测，对没有发生造假情况的租赁和商务服务业使用 OneClass SVM 进行异常点检测，得到所有行业（除制造业）在第 6 年共有 27 家上市公司发生财务造假。本文旨在寻求一个能包含全行业的集成学习机器分类模型，但是在超参数调优的过程中，发现由于 LR 模型的局限性，限制了集成学习机器模型的能力，暂时不能得到此机器模型。

本文利用机器学习算法，充分使用上市公司历史数据，融合了多种算法，且建立的 Stacking 集成学习预测模型较为稳定，能够帮助企业及时发现财务问题以采取补救措施；为投资者、企业合作伙伴等利益相关者提供更多财务信息来优化投资决策；为监管层提供有效方法，降低人力、物力成本，完善市场监管的作用，具有较大的参考价值和现实意义。

关键词：财务造假 嵌入法 机器学习 Stacking

Abstract

Due to the information asymmetry, it is difficult to prevent and predict the hidden financial data fraud and explosion of listed companies. Once the explosion occurs, it may produce domino effect and seriously damage the interests of investors. In this paper, machine learning is tried to use to find out the factors that have a great influence on the occurrence of fraud in each industry, as well as use the selected factors for machine learning to predict the companies that may have fraud in the sixth year.

Aiming at problem 1, two different embedding methods based on penalty terms and tree models are used. The former uses three models of LR, LASSO, and SVM, and the latter uses seven models of RF, GBDT, XGBoost, and LightGBM. Come to machine learning to find the features that are relevant to the fraud of listed companies. According to the importance of the features, get the top 30 indicators of each algorithm ranking, and then select the indicators that appear more than or equal to 3 times in all algorithms As a factor in each industry that has a greater impact on whether listed companies are fraudulent, and taking the manufacturing and real estate industries as examples, the selected indicators are analyzed for commonality and difference.

For question 2, after solving the problem of data imbalance in different industries through smote sampling, based on k-fold cross validation and grid search to optimize the super parameters of LR, RF, GBDT, XGBoost and LightGBM models with better effect in the first question with an evaluating indicator of F1 score and AUC. In this paper, Base models are mixed up to form an ensemble classification prediction model based on stacking ensemble learning theory. The first level basic learner selects RF, GBDT, XGBoost, LightGBM models, and the second level meta learner selects LR model, as to determine the optimal stacking ensemble learning classification prediction model. The F1 score and AUC of stacking integrated model in the test set are 0.96 and 0.79, which are higher than all the basic learners. There is no over fitting and the prediction effect is good. The prediction results of manufacturing industry in the sixth year are obtained: 20 listed companies have financial fraud.

One proposed solution for question 3, the second question method is used to predict other industries, and One Class SVM is used to detect outliers in the leasing and business service industries without fraud. It is found that 27 listed companies in all industries (except manufacturing) have financial fraud in the sixth year. This paper aims to find an ensemble learning machine classification model that can include the whole industry. However, in the process of super parameter optimization, it is found that due to the limitations of LR model, the ability of ensemble learning machine model is

limited, and this machine model can not be obtained temporarily.

This paper uses machine learning algorithm, makes full use of the historical data of listed companies, integrates a variety of algorithms, and establishes the stacking integrated learning prediction model, which is relatively stable, and can help enterprises find financial problems in time to take remedial measures; Provide more financial information for investors, business partners and other stakeholders to optimize investment decisions; It has great reference value and practical significance to provide effective methods for regulators, reduce the cost of human and material resources, and improve the role of market supervision.

Key Words: Financial Fraud, Embedded Methods, Machine Learning, Stacking

目录

1 问题重述

1.1 问题背景

这是一个信息增长速度飞快的时代，人们获取信息的方式也更加多样化。随着网络和计算机技术的快速发展，如何对各种重要资料进行数据分析是应对变化发展的主要途径。公司在经营过程中积累了大量的数据，股份持有者需要对企业财务数据进行有效分析；很多金融网站每天都发布各上市公司的信息，所有公司的管理人员分析这些数据决定各种策略，投资者也分析这些数据进行有效合理的投资。现代的信息管理系统在公司的运营管理活动中广泛使用，但传统的公司财务统计表分析等方法已经难以满足管理者对信息发现的需要。

目前的财务分析是基于公司或部门收集的公司经营状况的各项指标数据或统计资料，这些信息可以形成财务报表的数据或报告，以便向财务部门的工作人员提供分析和研究。比较指标数据和往年的数据，可以分析现在公司的运营是否有异常，如有异常，具体分析哪方面有异常，根据分析的报告书来改善公司的经营战略。但是，传统的财务分析手段无法准确发现数据记录和报告指标之间的内在联系，无法充分利用这些资源的有效信息。作为一个专业投资者，要研究上市公司的财务数据是否可靠，可以对上市公司多年的财务数据报告进行数据挖掘，选定数据指标进行跟踪分析和研究，识别真伪，避免投资踩雷。

1.2 问题提出

1. 根据附件 1 的行业分类，利用附件 2 所提供的相关上市公司的财务数据，确定出各行业与财务数据造假相关的数据指标，并分析比较不同行业上市公司相关数据指标的异同。

2. 根据附件 2 中制造业各上市公司的财务数据，确定出第 6 年财务数据造假的上市公司。

3. 根据附件 2 中其他（除制造业外）各行业上市公司的财务数据，确定出第 6 年财务数据造假的上市公司。

2 模型假设

（1） 假设所获取的数据是真实可靠的；

（2） 假设第 6 年未发生重大事件和灾难或国家未推行重要政策影响公司运营。

3 机器模型介绍

3.1 LASSO 模型

LASSO 是由 1996 年 Robert Tibshirani 首次提出，全称 Least absolute shrinkage and selection operator。该方法是一种压缩估计，它通过构造一个惩罚函数得到一个较为精炼的模型，使得它压缩一些回归系数，即强制系数绝对值之和小于某个固定值，同时设定一些回归系数为零。因此保留了子集收缩的优点，是一种处理具有复共线性数据的有偏估计。

LASSO 是在 RSS 最小化的计算中加入一个 l_P 范数作为惩罚约束。l_P 范数的好处是当 λ 充分大时，可以把某些待估系数精确地收缩到零。通过交叉验证法：对 λ 的给定值，进行交叉验证，选取交叉验证误差最小的 λ 值，然后按照得到的 λ 值，用全部数据重新拟合模型即可。

3.2 Logistic 模型

3.2.1 模型介绍

Logistic 回归即对数概率回归，它的名字虽然叫“回归”，但却是一种用于二分类问题的分类算法，它用 sigmoid 函数估计出样本属于某一类的概率。概率的值为 0~1，如果有这样一个函数：对于一个样本的特征向量，这个函数可以输出样本属于每一类的概率值，那么这个函数就可以用来作为分类函数，sigmoid 函数（也称 logistic 函数）就具有这种性质，他的定义为：

$$h(z)=\frac{1}{1+\exp(-z)} \tag{1}$$

这个函数的定义域为整个实数域，值域为（0，1），并且是一个单调的增函

数。根据对分布函数的要求，这个函数可以用来作为随机变量 x 的分布函数，即：

$$p(x \le z) = h(z) \tag{2}$$

直接将这个函数用于分类有问题，它是一个一元函数，在实际应用中特征向量一般是多维的。先用一个线性函数将输入向量 x 映射成一个实数 z 即可，这样就得到如下预测函数：

$$h(x) = \frac{1}{1+\exp(-w^{\mathrm{T}}x)} \tag{3}$$

其中，w 为线性映射权向量，由训练算法决定。

样本属于正样本的概率为：$p(y=1|x) = h(x)$；属于负样本的概率为：$p(y=0|x) = 1-h(x)$。

其中，y 为类别标签，取值为 1 或者 0，分别对应正负样本。样本属于正样本和负样本概率值比的对数称为对数似然比：

$$\ln\frac{p(y=1|x)}{p(y=0|x)} = \ln\frac{\dfrac{1}{1+\exp(-w^{\mathrm{T}}x)}}{1-\dfrac{1}{1+\exp(-w^{\mathrm{T}}x)}} = w^{\mathrm{T}}x \tag{4}$$

分类规则：如果正样本的概率大于负样本的概率，即：$h(x) > 0.5$。则样本被判定为正样本，否则被判定为负样本。这等价于：

$$\frac{h(x)}{1-h(x)} = \frac{p(y=1|x)}{p(y=0|x)} > 1 \tag{5}$$

也就是下面的线性不等式：$w^{\mathrm{T}}x > 0$，因此，logistic 回归是一个线性模型。

3.2.2 正则化

在构造机器学习模型时，最终目的是让模型在面对新数据的时候，可以有很好的表现。机器学习中，如果参数过多，模型过于复杂，容易造成过拟合（overfit）。即模型在训练样本数据上表现的很好，但在实际测试样本上表现的较差，不具备良好的泛化能力。为了避免过拟合，最常用的一种方法是使用正则化，例如 L1 和 L2 正则化。简单来说，正则化是一种为了减小测试误差的行为。

L1 正则化时，对应惩罚项为 L1 范数：

$$\Omega(\omega)=\|\omega\|_1=\sum_i|\omega_i| \tag{6}$$

L2 正则化时，对应惩罚项为 L2 范数：

$$\Omega(\omega)=\|\omega\|_2^2=\sum_i\omega_i^2 \tag{7}$$

从上式可以看出，L1 正则化通过让原目标函数加上了所有特征系数绝对值的和来实现正则化，而 L2 正则化通过让原目标函数加上了所有特征系数的平方和来实现正则化。

两者都是通过加上一个和项来限制参数大小，却有不同的效果：L1 正则化更适用于特征选择，而 L2 正则化更适用于防止模型过拟合。

3.3 支持向量机

3.3.1 模型介绍

支持向量机的目标是寻找一个分类超平面，它不仅能正确地分类每一个样本，并且要使得每一类样本中距离超平面最近的样本到超平面的距离尽可能远。假设训练样本有 l 个样本，特征向量 x_i 是 n 维向量，类别标签 y_i 取值为+1 或-1，分别对应正样本和负样本。支持向量机为这些样本寻找一个最优分类超平面，首先要保证每个样本能正确分类，因此，分类超平面的约束为：

$$y_i\left(w^{\mathrm{T}}x_i+b\right)\geq 1 \tag{8}$$

目标函数是超平面离两类样本的距离要足够大，可以写成：

$$\frac{1}{2}\|w\|^2 \tag{9}$$

再加上松弛变量 ξ_i 和惩罚因子 C 对违反不等式的样本进行惩罚，可以得到如下最优化问题：

$$\begin{aligned}&\min\frac{1}{2}w^{\mathrm{T}}w+C\sum_{i=1}^{l}\xi_i\\&y_i\left(w^{\mathrm{T}}x_i+b\right)\geq 1-\xi_i\\&\xi_i\geq 0,i=1,2,\cdots,l\end{aligned} \tag{10}$$

3.3.2 核映射和核函数

如果样本线性不可分，可以对特征向量进行映射将它转换到更高维的空间，使得在该空间中线性可分，这种方法在机器学习中被称为核技巧。核映射ϕ将特征向量变换到更高维的空间：

$$z=\phi(x) \tag{11}$$

常用的核函数如下表 1 所示。

表 1 常用的核函数

核函数	计算公式
线性核	$K\left(x_i,x_j\right)=x_i^T x_j$
多项式核	$K\left(x_i,x_j\right)=\left(\gamma x_i^T x_j+b\right)^d$
径向基函数核/高斯核	$K\left(x_i,x_j\right)=\exp\left(-\gamma\left\|x_i-x_j\right\|^2\right)$
sigmoid 核	$K\left(x_i,x_j\right)=\tanh\left(\gamma x_i^T x_j+b\right)$

3.4 One Class SVM

3.4.1 OCSVM

基本上将所有的数据点与零点在特征空间F分离开，并且最大化分离超平面到零点的距离。这产生一个*binary*函数能够获取特征空间中数据的概率密度区域。当处于训练数据点区域时，返回+1，处于其他区域返回-1。

该问题的优化目标与二分类 SVM 略微不同，但依然很相似：

$$\begin{aligned}&\min_{\omega,\varsigma_i,\rho}\frac{1}{2}\|\omega\|^2+\frac{1}{\nu n}\sum_{i=1}^{n}\varsigma_i-\rho\\&s.t.\left(\omega^T\phi\left(x_i\right)\right)>\rho-\varsigma_i,i=1,\cdots,n\\&\varsigma_i>0\end{aligned} \tag{12}$$

其中ς_i表示松弛变量，ν类似于二分类 SVM 中的C，同时：

1，ν为异常值的分数设置了一个上限（训练数据集里面被认为是异常的）；

2，ν是训练数据集里面作为支持向量的样例数量的下界。

因为这个参数的重要性，这种方法也被称为$\nu-SVM$。采用*Lagrange*技术并且采用$dot-product\ calculation$，确定函数变为：

$$f(x)=\mathrm{sgn}\left(\left(\omega^{T}\phi(x_i)\right)-\rho\right)=\mathrm{sgn}\left(\sum_{i=1}^{nn}\alpha_i K(x,x_i)-\rho\right) \tag{13}$$

这个方法创建了一个参数为ω,ρ的超平面，该超平面与特征空间中的零点距离最大，并且将零点与所有的数据点分隔开。

3.4.2 SVDD

SVDD 采用一个球形而不是平面的方法，该算法在特征空间中获得数据周围的球形边界，这个超球体的体积是最小化的，从而最小化异常点的影响。

产生的超球体参数为中心a和半径$R>0$，体积R^2被最小化，中心a是支持向量的线性组合；跟传统 SVM 方法相似，可以要求所有数据点x_i到中心的距离严格小于R，但同时构造一个惩罚系数为C的松弛变量ς_i，优化问题如下所示：

$$\begin{aligned}&\min_{R,a} R^2+C\sum_{i=1}^{n}\varsigma_i\\&s.t.\left\|x_i-a\right\|^2\le R^2+\varsigma_i, i=1,\cdots,n\\&\varsigma_i\ge 0, i=1,\cdots,n\end{aligned} \tag{14}$$

在采用拉格朗日算子求解之后，可以判断新的数据点z是否在类内，如果z到中心的距离小于或者等于半径。采用*Gaussian Kernel*作为两个数据点的距离函数：

$$\left\|z-x\right\|^2=\sum_{i=1}^{n}a_i\exp\left(\frac{-\left\|z-x_i\right\|^2}{\sigma^2}\right)\ge -R^2/2+C_R \tag{15}$$

3.5 随机森林

3.5.1 决策树

决策树是一种树形结构，其中每个内部节点表示一个属性上的判断，每个分支代表一个判断结果的输出，最后每个叶节点代表一种分类情况，本质是一颗由

多个判断节点组成的树，是通过训练数据并根据基尼系数的增益统计而来。

基尼系数表示数据集中样本的差异程度，基尼系数越大，表示数据集的种类越多样，即表示有多中的分类结果，表示数据集当前特征的越多样，即越不纯，即可能是一个多分类的问题。具体公式如下：

$$Gini(D)=1-\sum_{k=1}^{|y|}p_k^2 \tag{16}$$

一般选择基尼系数最小的特征作为最初的根节点。在实际项目中根据定义的分类结果进行根节点的选择，通过统计训练样本的特征与类别，根据特征相对初始类别的基尼系数的增益决定之后每一步根节点的选择。

在计算基尼系数增益之前需要知道当对每个特征作为判断依据时的基尼指数，具体公式如下：

$$Gini\_index(D,\alpha)=\sum_{v=1}^{v}\frac{|D^v|}{|D|}Gini(D^v) \tag{17}$$

基尼增益为：

$$Gini(D,\alpha)=Gini(D)-Gini\_index(D,\alpha) \tag{18}$$

不断地对内部节点进行分类直至显示最终的分类结果为止。此时对样本训练完毕，当有新的样本来进行测试时我们根据当前模型进行预测。在实际工程中需要控制决策树的深度防止过拟合的情形，即在训练样本可以表现很好的性能，但是并不能保证测试样本的准确度，这在训练集很大的时候容易出现。

3.5.2 Bagging 算法原理

Bagging 基于自助采样法（bootstrap sampling）。给定包含 N 个样本的训练数据集 D，自助采样法是这样进行的：先从 D 中随机取出一个样本放入采样集 D_s 中，再把该样本放回 D 中（有放回的重复独立采样）。经过 N 次随机采样操作，得到包含 N 个样本的采样集 D_s。

注意：数据集 D 中可能有的样本在采样集 D_s 中多次出现，但是 D 中也可能有样本在 D_s 中从未出现。一个样本始终不在采样集中出现的概率是 $\left(1-\frac{1}{N}\right)^N$。

根据：

$$\lim_{N\to\infty}\left(1-\frac{1}{N}\right)^{N}=\frac{1}{e}\approx 0.368 \tag{19}$$

因此D中约有63.2%的样本出现在了D_s中。

Bagging首先采用M轮自助采样法，获得M个包含N个训练样本的采样集。然后，基于这些采样集训练出一个基学习器。最后将这M个基学习器进行组合。训练流程如下：

循环，对$i=1,2,\cdots,T$**，**

对训练样本集进行Bootstrap抽样，得到抽样后的训练样本集，用抽样得到的样本集训练一个模型$h_i(x)$。

结束循环，输出模型组合$h_1(x),\cdots,h_T(x)$。

其中，T为弱学习器的数量。如果弱学习器是决策树，这种方法就是随机森林。

3.5.3 随机森林

随机森林（Random Forest，RF）是一种以决策树为基学习器的Bagging算法，传统决策树在选择划分属性时，每次选择一个最优属性，但是在BF在决策树的训练过程中引入了随机属性选择。

在RF中构建决策树，选择节点的划分属性时，首先从该节点的属性集合中随机选择一个包含k个属性的子集，然后再从这个子集中选择一个最优属性用于划分。

如果$k=n$（其中n为当前节点的属性的数量），则RF中决策树的构建与传统决策树相同。如果$k=1$，则随机选择一个属性用于划分。通常建议$k=\log_2 n$。

RF的训练效率较高，因为RF使用的决策树只需要考虑一个属性的子集。另外，RF简单并且容易实现，计算开销小，而且它在很多现实任务中展现出强大的性能。

在随机森林中，每棵树构建时的样本都是由训练集经过有放回抽样得来的（例如，自助采样法(bootstrap sample)）。此外，在树的构造过程中拆分每个节点

时，可以从所有输入特征或大小为 max_features 的随机子集中找到最佳拆分。

这两个随机性的目的是减少森林估计器的方差。实际上，单个决策树通常表现出较高的方差并且倾向于过拟合。在森林中注入随机性来产生决策树，让其预测误差有些解耦。通过取这些预测结果的平均值，可以减少一些误差。随机森林通过组合不同的树来减少方差，有时会以略微增加偏差(bias)为代价。在实践中，由于方差的减小通常是很明显的，因此在总体上产生了更好的模型。

与上述相比，scikit-learn 的实现是取每个分类器预测出的概率的平均，而不是让每个分类器对单个类别进行投票。

3.6 梯度提升树

梯度提升树模型是一种基于回归树的集成学习方法，它通过构造多个弱的回归树作为基学习器，并把这些树的结果累加起来作为最终的预测输出。因为其特征处理的简单性和优异的效果，梯度提升树模型被认为是传统统计学习中效果最好的方法之一。

梯度提升树算法是集成学习 Boosting 家族的一员，它在训练时采用前向分步算法，首先确定第一棵树拟合的值，然后基于之前所有树的误差来更新训练并训练下一棵树，一步一步迭代下去直到梯度提升树模型构建完毕。所以我们在训练时，首先确定初始提升树 $f(x)=0$，然后在后续训练时第 m 步的模型是：

$$f_m(x)=f_{m-1}(x)+T(x;\Theta_m) \tag{20}$$

其中，$f_{m-1}(x)$ 是当前模型，通过经验风险最小化确定下一棵数的参数 Θ_m。

$$\Theta_m=\arg\min\sum_{i=1}^{N}L\left(y_i,f_{m-1}(x)+T(x;\Theta_m)\right) \tag{21}$$

那么我们可以使用梯度下降法对以上式子进行求解，并得到梯度提升树的算法，下面是梯度提升树的算法：

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\},x_1\in\chi\subseteq R^n,y_1\in Y\subseteq R^n$；**损失函数** $Cost(y,f(x))$。

过程：（1）**初始化：** $f_0(x)=\arg\min\sum_{i=1}^{N}L(y_i,c)$；

（2）对 $m=1,2,\cdots,M$ ；

（a）对 $i=1,2,\cdots,N$ ，计算： $\gamma_{mi}=-\left[\frac{\partial L\left(y_i,f\left(x_i\right)\right)}{\partial f\left(x_i\right)}\right]f\left(x\right)=f_{m-1}\left(x\right)$ ；

（b）对 γ_{mi} 拟合一个回归树，得到第 m 棵树的叶节点区域 $R_{mj},j=1,2,\cdots,J$ ；

（c）对 $j=1,2,\cdots,J$ ，计算： $c_{mj}=\arg\min_{c}\sum_{x_i\in R_{mj}}L\left(y_i,f_{m-1}\left(x_i\right)+c\right)$ ；

（d）更新 $f_m\left(x\right)=f_{m-1}\left(x\right)+\sum_{j=1}^{J}c_{mj}I\left(x\in R_{mj}\right)$ ；

（3）得到回归树 $\hat{f}\left(x\right)=f_M\left(x\right)=\sum_{m=1}^{M}\sum_{j=1}^{J}c_{mj}I\left(x\in R_{mj}\right)$ 。

输出：回归树 $\hat{f}\left(x\right)$ 。

3.7 XGBoost

XGBoost 是 Boosting Tree 的一种实现，它对损失函数进行了二阶泰勒展开，同时用到了一阶和二阶导数，并引入了适用于树模型的正则项用于控制模型的复杂度。XGBoost 的正则项里包含了树的叶子节点个数、每个叶子节点输出分数的 $L2$ 平方和。

$$\tilde{L}^{(t)}=\sum_{i=1}^{n}l\left(y_i,\hat{y}_i^{(t-1)}+f_t\left(X_i\right)\right)+\Omega\left(f_t\right)+\text{常数} \tag{22}$$

对上式用泰勒展开可以将目标函数转化为：

$$\begin{aligned}\tilde{L}^{(t)}&\simeq\sum_{i=1}^{n}l\left(y_i,\hat{y}_i^{(t-1)}+f_t\left(X_i\right)\right)+\Omega\left(f_t\right)+\text{常数}\\&=\sum_{i=1}^{n}\left[g_iw_q\left(x_i\right)+\frac{1}{2}h_iw_q^2\left(x_i\right)\right]+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2\\&=\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j}g_i\right)w_j+\frac{1}{2}\left(\sum_{i\in I_j}h_i+\lambda\right)w_j^2\right]+\gamma T\end{aligned} \tag{23}$$

对上式求 ω 的偏导，然后令偏导等于零，可以求解到：

$$\omega^*=-\frac{\sum_{i\in I_j}g_i}{\sum_{i\in I_j}h_i+\lambda} \tag{24}$$

然后代入目标函数得到：

$$\tilde{L}^{(t)}(q)=-\frac{1}{2}\sum_{j=1}^{T}\frac{\sum_{i\in I_j}g_i}{\sum_{i\in I_j}h_i+\lambda}+\gamma T \tag{25}$$

同样在选取分裂点的时候， 也以最小化目标函数为目标。假设在某次选取分裂点进行划分后，I_L和I_R分别是划分后的左右节点，且$I=I_L\cup I_R$，那么在该划分后损失函数减小的值为：

$$L_{split}=\frac{1}{2}\left[\frac{\left(\sum_{i\in I_L}g_i\right)^2}{\sum_{i\in I_L}h_i+\lambda}+\frac{\left(\sum_{i\in I_R}g_i\right)^2}{\sum_{i\in I_R}h_i+\lambda}-\frac{\left(\sum_{i\in I_j}g_i\right)^2}{\sum_{i\in I_j}h_i+\lambda}\right]-\gamma \tag{26}$$

3.8 LightGBM

LightGBM 是一个梯度 Boosting 框架，使用基于决策树的学习算法。它可以说是分布式的，高效的，有以下优势：

（1） 更快的训练效率；

（2） 低内存使用；

（3） 更高的准确率；

（4） 支持并行化学习。

3.8.1 GOSS

GOSS（Gradient-based One-Side Sampling）是通过区分不同梯度的实例，保留较大梯度实例同时对较小梯度随机采样的方式减少计算量，从而达到提升效率的目的，在提升树训练过程中目标函数学习的就是负梯度（近似残差）。

GOSS 的计算步骤如下：

选取前$a\%$个较大梯度的值作为大梯度值的训练样本，从剩余的$1-a\%$个较小梯度的值中，我们随机选取其中的$b\%$个作为小梯度值的训练样本，对于较小梯度的样本，也就是$b\%\times(1-1\%)\times samples$，我们在计算信息增益时将其放大$(1-a)/b$倍。总的来说就是$a\%\times samples+\ b\%\times(1-a\%)\times samples$个样本作为训

练样本。而这样的构造是为了尽可能保持与总的数据分布一致，并且保证小梯度值的样本得到训练。

3.8.2 独立特征合并

EFB（Exclusive Feature Bundling）是通过特征捆绑的方式减少特征维度的方式来提升计算效率。通常被捆绑的特征都是互斥的（一个特征值为零，一个特征值不为零），这样两个特征捆绑起来才不会丢失信息。如果两个特征并不是完全互斥（部分情况下两个特征都是非零值），可以用一个指标对特征不互斥程度进行衡量，称之为冲突比率，当这个值较小时，可以选择把不完全互斥的两个特征捆绑，而不影响最后的精度。

3.8.3 特征融合

构建一个以特征为图中的点(V)，以特征之间的总冲突为图中的边(E)，这里说的冲突值应该是特征之间*cos*夹角值，因为要尽可能保证特征之间非 0 元素不在同一个行上。而寻找合并特征且使得合并的簇个数最小，这是一个图着色问题。所以这个找出合并的特征且使得簇个数最小的问题需要使用近似的贪心算法来完成。

首先按照度来对每个点（特征）做降序排序（度数越大与其他点的冲突越大），然后将特征合并到冲突数小于 K 的簇或者新建另外一个簇。

3.8.4 特征合并

将这些簇中的特征合并起来。由于每一个簇当中，特征的范围都是不一样，所以我们需要重新构建合并后簇特征的范围。在第一个 for 循环当中，我们记录每个特征与之前特征累积总范围。在第二个 for 循环当中，根据之前的 bin Ranges 重新计算出新的 bin value 保证特征之间的值不会冲突。这是针对于稀疏矩阵进行优化。由于之前 Greedy Bundling 算法对特征进行冲突检查确保簇内特征冲突尽可能少，所以特征之间的非零元素不会有太多的冲突。

EBF 的算法步骤如下：

（1） 将特征按照非零值的个数进行排序；

（2） 计算不同特征之间的冲突比率；

（3） 遍历每个特征并尝试合并特征，使冲突比率最小化。

我们称使用 GOSS 算法和 EFB 算法的梯度提升树（GBDT）称之为 LightGBM。

4 相关理论介绍

4.1 缺失值处理[1]

4.1.1 缺失值来源

数据的缺失是无法避免的，可能的原因有很多种，有以下三大类：

（1） 无意的：信息被遗漏，比如由于工作人员的疏忽，忘记而缺失；或者由于数据采集器等故障等原因造成的缺失，比如系统实时性要求较高的时候，机器来不及判断和决策而造成缺失；

（2） 有意的：有些数据集在特征描述中会规定将缺失值也作为一种特征值，这时候缺失值就可以看作是一种特殊的特征值；

（3） 不存在：有些特征属性根本就是不存在的，比如一个未婚者的配偶名字就没法填写，再如一个孩子的收入状况也无法填写；

总而言之，对于造成缺失值的原因，我们需要明确：是因为疏忽或遗漏无意而造成的，还是说故意造成的，或者说根本不存在。只有知道了它的来源，才能对症下药，做相应的处理。

4.1.2 数据缺失类型

在对缺失数据进行处理前，了解数据缺失的机制和形式是十分必要的。将数据集中不含缺失值的变量称为完全变量，数据集中含有缺失值的变量称为不完全变量。而从缺失的分布来将缺失可以分为完全随机缺失，随机缺失和完全非随机缺失。

（1） 完全随机缺失（missing completely at random，MCAR）：指的是数据的缺失是完全随机的，不依赖于任何不完全变量或完全变量，不影响样本的无偏性，如家庭地址缺失；

（2） 随机缺失(missing at random，MAR)：指的是数据的缺失不是完全随机的，即该类数据的缺失依赖于其他完全变量，如财务数据缺失情况与

企业的大小有关；

（3） 非随机缺失(missing not at random, MNAR)：指的是数据的缺失与不完全变量自身的取值有关，如高收入人群不原意提供家庭收入；

对于随机缺失和非随机缺失，直接删除记录是不合适的，原因上面已经给出。随机缺失可以通过已知变量对缺失值进行估计，而非随机缺失的非随机性还没有很好的解决办法。题目中的数据缺失属于随机缺失，所以选择估计缺失值的方法对缺失值进行填充。

4.1.3 缺失值处理

以下是处理缺失值的三种方法：删除记录，数据填补，和不处理。

（1） 删除记录

优点：最简单粗暴；缺点：牺牲了大量的数据，通过减少历史数据换取完整的信息，这样可能丢失了很多隐藏的重要信息；当缺失数据比例较大时，特别是缺失数据非随机分布时，直接删除可能会导致数据发生偏离，比如原本的正态分布变为非正态分布；这种方法在样本数据量十分大且缺失值不多的情况下非常有效，但如果样本量本身不大且缺失也不少，那么不建议使用。

（2） 数据填补

对缺失值的插补大体可分为两种：替换缺失值，拟合缺失值，虚拟变量。替换是通过数据中非缺失数据的相似性来填补，其核心思想是发现相同群体的共同特征，拟合是通过其他特征建模来填补，虚拟变量是衍生的新变量代替缺失值。

替换缺失值

均值插补：对于定类数据：使用众数（mode）填补；对于定量（定比）数据：使用平均数（mean）或中位数（median）填补。一般如果特征分布为正态分布时，使用平均值效果比较好，而当分布由于异常值存在而不是正态分布的情况下，使用中位数效果比较好。此方法虽然简单，但是不够精准，可能会引入噪声，或者会改变特征原有的分布。

热卡填补（Hot deck imputation）：热卡填充法是在完整数据中找到一个与它最相似的对象，然后用这个相似对象的值来进行填充。通常会找到超出一个的相似对象，在所有匹配对象中没有最好的，而是从中随机的挑选一个作为填充值。这个问题关键是不同的问题可能会选用不同的标准来对相似进行判定，以及如何

制定这个判定标准。该方法概念上很简单，且利用了数据间的关系来进行空值估计，但缺点在于难以定义相似标准，主观因素较多。

K 最近距离邻法（K-means clustering）：另外一种方法就是利用无监督机器学习的聚类方法。通过 K 均值的聚类方法将所有样本进行聚类划分，然后再通过划分的种类的均值对各自类中的缺失值进行填补。归其本质还是通过找相似来填补缺失值。

拟合缺失值

拟合就是利用其它变量做模型的输入进行缺失变量的预测，与我们正常建模的方法一样，只是目标变量变为了缺失值。如果其它特征变量与缺失变量无关，则预测的结果毫无意义。如果预测结果相当准确，则又说明这个变量完全没有必要进行预测，因为这必然是与特征变量间存在重复信息。一般情况下，会介于两者之间效果为最好，若强行填补缺失值之后引入了自相关，这会给后续分析造成障碍。

回归预测：基于完整的数据集，建立回归方程。对于有缺失值的特征值，将已知特征值代入模型来估计未知特征值，以此估计值来进行填充。

极大似然估计（Maximum likelyhood）：在缺失类型为随机缺失的条件下，假设模型对于完整的样本是正确的，那么通过观测数据的边际分布可以对未知参数进行极大似然估计（Little and Rubin）。这种方法也被称为忽略缺失值的极大似然估计，对于极大似然的参数估计实际中常采用的计算方法是期望值最大化(Expectation Maximization，EM)。该方法比删除个案和单值插补更有吸引力，它一个重要前提：适用于大样本。有效样本的数量足够以保证 ML 估计值是渐近无偏的并服从正态分布。但是这种方法可能会陷入局部极值，收敛速度也不是很快，并且计算很复杂，且仅限于线性模型。

多重插补（Mutiple imputation）：多值插补的思想来源于贝叶斯估计，认为待插补的值是随机的，它的值来自于已观测到的值。具体实践上通常是估计出待插补的值，然后再加上不同的噪声，形成多组可选插补值。根据某种选择依据，选取最合适的插补值。

根据数据缺失机制、模式以及变量类型，可分别采用回归、预测均数匹配(predictive mean matching, PMM)、趋势得分(propensity score, PS)、Logistic 回

归、判别分析以及马尔可夫链蒙特卡罗(Markov Chain Monte Carlo, MCMC) 等不同的方法进行填补。

（3） 不处理

补齐处理只是将未知值补以我们的主观估计值，不一定完全符合客观事实，在对不完备信息进行补齐处理的同时，我们或多或少地改变了原始的信息系统。而且，对空值不正确的填充往往将新的噪声引入数据中，使挖掘任务产生错误的结果。因此，在许多情况下，我们还是希望在保持原始信息不发生变化的前提下对信息系统进行处理。

在实际应用中，一些模型无法应对具有缺失值的数据，因此要对缺失值进行处理。然而还有一些模型本身就可以应对具有缺失值的数据，此时无需对数据进行处理，比如 XGBoost，RF 等高级模型。

4.2 数据不平衡[2]

数据不平衡也可称作数据倾斜。在实际应用中，数据集的样本特别是分类问题上，不同标签的样本比例很可能是不均衡的。因此，如果直接使用算法训练进行分类，训练效果可能会很差，本文主要是用以下三种方法对数据进行采样，消除不平衡。

4.2.1 过采样

过采样（RandomOverSampler）是主动获取更多的比例少的样本数据。由于样本比例不均衡，在条件允许的情况下可以尝试获取占比少的类型的样本数据，也可以通过使用重复、自举或合成少数类过采样等方法（SMOTE）来生成新的稀有样品。

直接简单复制重复的话，如果特征少，会导致过拟合的问题。经过改进的过抽样方法通过在少数类中加入随机噪声、干扰数据或通过一定规则产生新的合成样本(数据增强)。

4.2.2 欠采样

数据量足够时，通过欠采样（RandomUnderSampler）保留比例小的样本数据和减少比例大的样本数据来平衡数据集。缺点是会丢失多数类中的一些重要信息。

4.2.3 SMOTE 采样

SMOTE 采样的原理是在少数类样本之间进行插值来产生额外的样本。具体地，对于一个少数类样本使用 k 近邻法(k 值需要提前指定)，求出离 x_i 距离最近的 k 个少数类样本，其中距离定义为样本之间 n 维特征空间的欧氏距离。然后从 k 个近邻点中随机选取一个，使用下列公式生成新样本：

$$x_{new} = x_i + (\hat{x}_i - x_i) \times \delta \tag{27}$$

其中 $\hat{x}_i$ 为选出的 k 近邻点， $\delta \in [0,1]$ 是一个随机数。

SMOTE 会随机选取少数类样本用以合成新样本，而不考虑周边样本的情况，这样容易带来两个问题：

（1） 如果选取的少数类样本周围也都是少数类样本，则新合成的样本不会提供太多有用信息；

（2） 如果选取的少数类样本周围都是多数类样本，这类的样本可能是噪音，则新合成的样本会与周围的多数类样本产生大部分重叠，致使分类困难。

4.3 特征选择[3]

与特征提取是从原始数据中构造新的特征不同，特征选择是从这些特征集合中选出一个子集。特征选择对于机器学习应用来说非常重要。特征选择也称为属性选择或变量选择，是指为了构建模型而选择相关特征子集的过程。

使用特征选择的前提是：训练数据中包含许多冗余或者无关的特征，移除这些特征并不会导致丢失信息。冗余和无关是两个概念。如果一个特征本身有用，但这个特征与另外一个有用的特征强相关，则这个特征可能就变得冗余。特征选择常用于特征很多但样本相对较少的情况。

特征选择一般包括产生过程、评价函数、停止准则、验证过程。为了进行特征选择，我们首先需要产生特征或特征子集候选集合，其次需要衡量特征或特征子集的重要性或者好坏程度，因此需要量化特征变量和目标变量之间的联系以及特征之间的相互联系。为了避免过拟合，我们一般采用交叉验证的方式来评估特征的好坏；为了减少计算复杂度，我们可能还需要设定一个阈值，当评价函数值

达到阈值后搜索停止；最后，我们需要再验证数据集上验证选出来的特征子集的有效性。

特征选择的方法主要分为三大类：过滤式方法 (Filter Methods)，包裹式方法 (Wrapper Methods) 和嵌入式方法 (Embedded Methods)。

过滤式方法运用统计指标来为每个特征打分并筛选特征，其聚焦于数据本身的特点。其优点是计算快，不依赖于具体的模型，缺点是选择的统计指标不是为特定模型定制的，因而最后的准确率可能不高。而且因为进行的是单变量统计检验，没有考虑特征间的相互关系。

包裹式方法使用模型来筛选特征，通过不断地增加或删除特征，在验证集上测试模型准确率，寻找最优的特征子集。包裹式方法因为有模型的直接参与，因而通常准确性较高，但是因为每变动一个特征都要重新训练模型，因而计算开销大，其另一个缺点是容易过拟合。

嵌入式方法利用了模型本身的特性，将特征选择嵌入到模型的构建过程中。典型的如 LASSO 和树模型等。准确率较高，计算复杂度介于过滤式和包裹式方法之间，但缺点是只有部分模型有这个功能。

4.3.1 常用方法比较

对常用的特征选择算法进行比较，如下表 2 所示。

表 2 常用的特征选择算法

特征选择方法		优点	缺点	举例
过滤方法	单变量	速度快、可扩展、跟机器学习模型独立	忽略特征之间的关系、忽略了特征和模型之间的关系	卡方检验、信息增益、相关系数
	多变量	考虑了特征之间的相关性、跟机器学习模型独立、计算复杂度优于封装方法	计算速度和可扩展性低于单变量方法、忽略了特征和模型之间的关系	基于相关性的特征选择（CFS）、MBF、FCBF
封装法	确定性算法	简单、跟机器学习模型相关、考虑特征之间的相互作用、计算密集	容易过拟合、相比于随机算法容易卡在局部最优子集（贪心搜索）、	序列向前特征选择（SFS）、序列向后特征删减（SBE）、增 q 删 r

		程度低于随机算法	依赖机器学习模型	
	随机算法	不容易达到局部极小点、跟机器学习模型相关、考虑特征之间的相互作用	计算密集型、依赖机器学习模型、相比确定系算法过拟合的风险较高	模拟退火、随机爬山、基因算法
嵌入法	与模型相关、计算复杂度优于封装方法、考虑特征之间的相互作用、		依赖机器学习模型	决策树、随机森林、梯度提升树、SVM、LASSO

4.4 超参数寻优

机器学习模型参数众多，参数选择不恰当，就会出现欠拟合或者过拟合的问题。为了提高模型的精度，同时提升模型的泛化能力，调参过程不可缺少。而在选择超参数的时候，有两个途径，一个是凭经验微调，另一个就是选择不同大小的参数，带入模型中，挑选表现最好的参数。

本文主要选择网格搜索，网格搜索是一种重要调参手段，即穷举搜索：在所有候选的参数选择中，通过循环遍历，尝试每一种可能性，表现最好的参数就是最终的结果。其原理就像是在数组里找到最大值，网格搜索可以保证在指定的参数范围内找到精度最高的参数。

4.5 K 折交叉验证

K 折交叉验证（K-fold Cross Validation）将数据集 D 划分成 K 份互斥数据集 D_K，满足 $D = D_1 \cup \cdots \cup D_k$，一般是平均分配使每份数据量接近并且数据分布尽可能一致。每次用一份数据测试，其余 $K-1$ 份数据训练，需要迭代 K 轮得到 K 个模型；最后再将 K 份测试结果汇总到一起评估一个离线指标。

$$cv\_score = \frac{1}{K}\sum_{k=1}^{K} L\left(P_k, Y_k\right) \tag{28}$$

K 折交叉验证的稳定性与 K 取值有很大关系。K 值太小实验稳定性依然偏低， K 值太大又可能导致实验成本高，K 最常用的取值是 5 和 10。K 折交叉验证能够更好地避免过拟合和欠拟合，得到的结论也更有说服力。

4.6 模型评估

评估指标用于反映模型效果。在预测问题中，要评估模型的效果，就需要将模型预测结果 $f(X)$ 和真实标注 Y 进行比较。

评估指标定义为 $f(X)$ 和 Y 的函数：

$$score = metric\left(f(X), Y\right) \tag{29}$$

模型的好坏是相对的，在对比不同的模型效果时，使用不同评估指标往往会导致不同的结论。

4.6.1 精确率和召回率

精确率和召回率多用于二分类问题，可结合混淆矩阵介绍，如表 3 所示。

表 3 混淆矩阵

	预测结果		
真实结果		正（P）	负（N）
	正（P）	TP	FN
	负（N）	FP	TN

其中，TP（真正，True Positive）表示真实结果为正例，预测结果也是正例；FP（假正，False Positive ）表示真实结果为负例，预测结果却是正例；TN（真负，True Negative）表示真实结果为正例，预测结果却是负例；FN（假负，False Negative）表示真实结果为负例，预测结果也是负例。显然，TP+FP+FN+TN＝样本总数。

精确率 P 和召回率 R 的定义为：

$$\begin{aligned} \text{精确率}(P) &= \frac{TP}{TP+FP} \\ \text{召回率}(R) &= \frac{TP}{TP+FN} \end{aligned} \tag{30}$$

理想情况下，精确率和召回率两者都越高越好。然而事实上这两者在某些情况下是矛盾的：精确率高时，召回率低；而精确率低时，召回率高。

4.6.2 F1-score

F_1值是精确率和召回率的调和平均值：

$$\frac{2}{F_1}=\frac{1}{P}+\frac{1}{R} \tag{31}$$

F值可泛化为对精确率和召回率赋予不同权重进行加权调和：

$$F_\alpha=\frac{\left(1+\alpha^2\right)\cdot P\cdot R}{\alpha^2\cdot P+R} \tag{32}$$

此外，准确率和错误率也是常用的评估指标。

$$\begin{aligned}&\text{准确率}\left(accuracy\right)=\frac{TP+TN}{TP+FP+FN+TN}\\&\text{错误率}\left(error\ rate\right)=\frac{FP+FN}{TP+FP+FN+TN}\end{aligned} \tag{33}$$

精确率和准确率是比较容易混淆的两个评估指标，两者是有区别的。精确率是一个二分类指标，而准确率能应用于多分类，其计算公式为：

$$\text{准确率}\left(accuracy\right)=\frac{1}{n}\sum_{i=1}^{n}I\left(f\left(x_i\right)=y_i\right) \tag{34}$$

4.6.3 ROC 与 AUC

在众多的机器学习模型中，很多模型输出是预测概率。而使用精确率、召回率这类指标进行模型评估时，还需要对预测概率设分类阈值，比如预测概率大于阈值为正例，反之为负例。这使得模型多了一个超参数，并且这个超参数会影响模型的泛化能力。

接收者操作特征（Receiver Operating Characteristic，ROC）曲线不需要设定这样的阈值。ROC 曲线纵坐标是真正率，横坐标是假正率，其对应的计算公式为：

$$\begin{aligned}&\text{真正率}\left(TPR\right)=\frac{TP}{TP+FN}\\&\text{假正率}\left(FPR\right)=\frac{FP}{TP+TN}\end{aligned} \tag{35}$$

ROC 曲线越靠近左上角性能越好。左上角坐标为(0,1)，即 $FPR=0$， $TPR=1$，

根据 FPR 和 TPR 公式可以得知，此时 $FN=0$，$FP=0$，模型对所有样本分类正确。

绘制 ROC 曲线很简单，首先对所有样本按预测概率排序，以每条样本的预测概率为阈值，计算对应的 FPR 和 TPR，然后用线段连接。当数据量少时，绘制的 ROC 曲线不平滑；当数据量大时，绘制的 ROC 曲线会趋于平滑。

AUC(Area Under Roc Curve）即 ROC 曲线下的面积，取值越大说明模型越可能将正样本排在负样本前面。AUC 还有一些统计特性：AUC 等于随机挑选一个正样本（P）和负样本（N）时，分类器将正样本排前面的概率；AUC 和 Wilcoxon Test of Ranks 等价；AUC 还和基尼（Gini）系数有联系，满足等式 $Gini+I=2\cdot AUC$。

AUC 的计算方法有多种，从物理意义角度理解，AUC 计算的是 ROC 曲线下的面积：

$$AUC=\sum_{i\in(P+N)}\frac{\left(TPR_i+TPR_{i-1}\right)\cdot\left(FPR_i-FPR_{i-1}\right)}{2} \tag{36}$$

AUC 计算主要与排序有关，所以它对排序敏感，而对预测分数没那么敏感。

4.7 模型融合

Stacking 是另一种更强大的模型融合方法，于 1992 年被 Wolpert©提出，其基本思路是，通过一个模型来融合若干单模型的预测结果，目的是降低单模型的泛化误差。在这里，这些单模型被称为一级模型，Stacking 融合模型被称为二级模型或元模型。

Stacking 先从初始的训练集训练出若干单模型，然后把单模型的输出结果作为样本特征进行整合，并把原始样本标记作为新数据样本标记，生成新的训练集。再根据新训练集训练一个新的模型，最后用新的模型对样本进行预测。Stacking 融合模型本质上是一种分层结构，每一层都是若干模型。简单起见，这里先讨论二级 Stacking。同时，我们假设所有的单模型都是使用不同的学习算法产生，即异质模型（当然，Stacking 的一级模型也可以是同质模型）。

在 Stacking 的模型训练阶段，二级模型的训练集是利用一级模型产生的。如果直接使用一级模型对初始的训练集样本进行预测来产生二级训练集，这样会有极大的过拟合的风险。因此，一般是用训练一级模型未使用的样本来产生二级模

型的训练集。交叉验证法或留一法是比较常用的方法。下面以 K 折交叉验证为例，来介绍二级模型的训练集是如何生成的。

输入：训练集 $D=\{(x_1,y_1),(x_2,y_2),\cdots(x_m,y_m)\}$；**单模型学习算法** $\xi_1,\xi_2,\cdots\xi_T$；**融合模型学习算法** ξ。

过程：

for $t=1,2,\cdots,T$ *do*
　$h_t=\xi_t(D)$
end for
$D'=\phi$　　//基于原始数据训练个单模型
for $i=1,2,\cdots,m$ *do*
　for $t=1,2,\cdots,T$ *do*
　　$z_{it}=h_t(x_i)$　　//单模型输出作为样本新特征
　end for
　$D'=D'\cup((z_{i1},z_{i2},\cdots,z_{iT}),y_i)$ //单模型输出和样本标记构建新训练集
end for
$h'=\xi(D')$　　//在新训练集上训练二级模型

输出： $H(x)=h'(h_1(x),h_2(x),\cdots,h_T(x))$ //二级模型输出作为*Stacking*输出

我们把初始训练集 D 随机划分成 k 个大小相似的集合 $D_1,D_2,\cdots,D_k$。令 D_j 和 $\overline{D}_j$（它等于 $D\setminus D_j$）分别表示第 j 折的测试集和训练集。给定 T 个初级学习算法，初级学习器 $h_t^{(j)}$ 通过使用第 t 个学习算法而得。对于 $\overline{D}_j$ 中的每个样本 x_i，定义第 t 个模型的预测结果为 $z_{jt}=h_t^{(j)}(x_i)$，那么由样本 x_i 产生的二级模型训练样本特征为 $z_i=(z_{i1};z_{i2};\cdots z_{iT})$，样本标记还是原始的样本标记 y_i。于是，在经过 $k\times T$ 次模型训练和预测后，得到二级训练集 $D'=\{(z_i,y_i)\}_{i=1}^{m}$，然后 D' 用来训练二级模型。

在这里，二级模型 h' 是 $(z_1,z_2,\cdots,z_T)$ 关于 y 的函数。

5 问题一：基于嵌入法的特征选择

5.1 数据预处理

5.1.1 数据描述

题目已公布三个附件，每个附件都包含不同的数据。

附件 1：上市公司的行业分类；

附件 2：上市公司财务数据；

附件 3：附件 2 中数据字段的说明。

根据统计，附件 1 共有 4163 家企业，涵盖了 19 个行业，附件二包含了每个上市公司多年的财务数据报告数据，共 22214 条记录，每条记录有 364 个特征。前 5 年共 18061 条标记了 FLAG 的数据，第六年包含了 4154 条不标记 FLAG 的数据。

在 364 个财务数据因子中，特征 FLAC 是目标列，其中 1 表示上市公司在当年出现财务造假；0 表示没有造假。剩下的 363 个特征因子中，包含 10 个上市公司身份特征因子和 353 个上市公司财务特征因子，如图 1 所示。

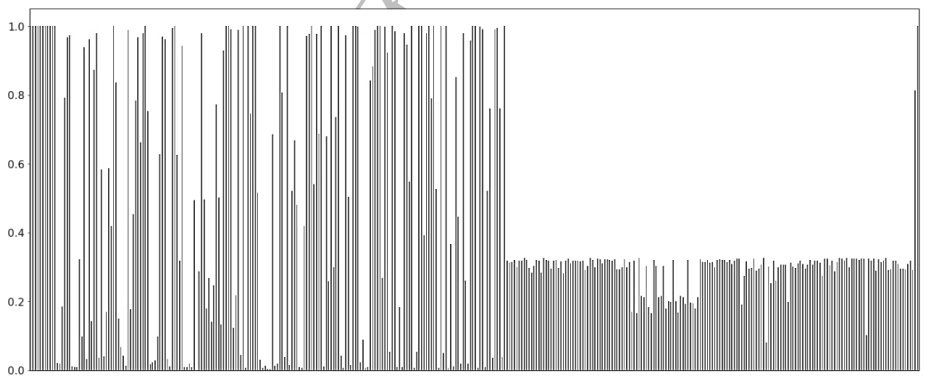

图 1 附件二的缺失程度示意图

5.2 无关信息处理

在上市公司身份特征因子中股票代码是每个股票的唯一特征，没有实际意义，需要删除。其他的，报告类型、会计区间、合并标志、会计准则、货币代码这些没用的身份特征因子与是否会造假没有关系，需要先将它们删除。

表 4 剔除的特征示意表

TICKER_SYMBOL	股票代码
ACT_PUBTIME	实际披露时间
PUBLISH_DATE	发布时间
END_DATE_REP	报告截止日期
REPORT_TYPE	报告类型
FISCAL_PERIOD	会计区间
MERGED_FLAG	合并标志：1-合并，2-母公司
ACCOUTING_STANDARDS	会计准则
CURRENCY_CD	货币代码

附件二中截止日期特征代表该公司造假的年份，其他年份无研究意义，将其删除，最后得到 354 个特征，如图 2 所示。

图 2 删除无关信息后的缺失程度示意图

5.2.1 缺失值处理

提出无关信息后，每个特征因子或多或少都存在缺失，确实太多的特征不利于机器学习，缺失程度大于 70%的特征已经没有应用意义，所以我们对缺失程度为大于 70%的特征进行删除。

缺失程度在 40%到 70%之间的也属于缺失严重，因为后面会采用高级机器模型进行训练，所以填充过多的缺失值会给模型带来噪音而干扰到模型的准确性，因此采用 0 值对缺失值进行填充。

缺失程度在 40%以下，采用 KNN 算法进行填充。处理缺失值的数学模型如下：

$$\begin{cases} 70\% < \Gamma \le 100\%, \text{舍弃该特征} \\ 40\% < \Gamma \le 70\% \ , \text{用 } 0 \text{ 值填充} \\ 0\ \% < \Gamma \le 40\% \ , \text{用} KNN \text{算法填充} \end{cases} \tag{37}$$

经过筛选，留下 239 个特征，并且所有特征已经全部补全，没有缺失值，如

图 3 所示。

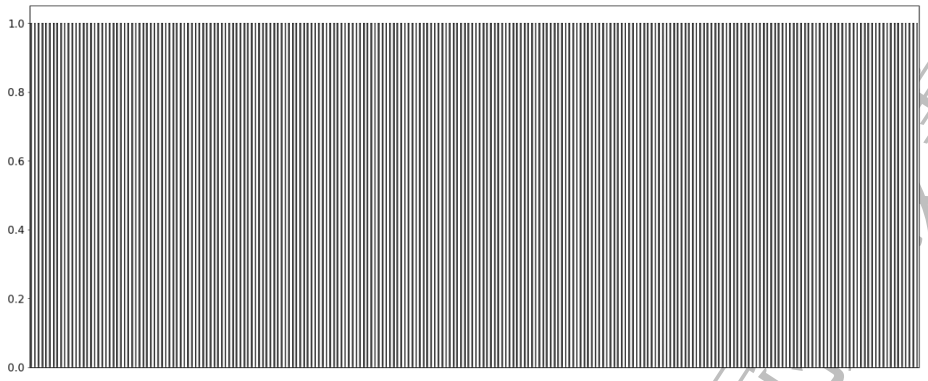

图 3 缺失值处理后的特征缺失程度示意图

5.2.2 标准化处理

数据的标准化是指处理数据时统一不同指标之间的量纲。具有不同量纲的数据的差别可能会很大，并且其中有些数据是负值，会影响到后面的特征选择和机器学习，所以，为了解决量纲不统一所带来的问题，需要对数据进行标准化处理，通常采用的方法是把每个指标按照一定比例地缩放到 0 与 1 之间。

最小—最大标准化方法是对原始数据的线性变换。设 $minA$ 和 $maxA$ 分别为属性 A 的最小值和最大值，将 A 的一个原始值 x 通过 $min-max$ 标准化映射在区间 $[new\_minA, new\_maxA]$ 中的值 x，转换公式如下：

$$x^{*}=\frac{x-\min}{\max-\min} \tag{38}$$

其中，max 为样本数据的最大值，min 为样本数据的最小值。$max-min$ 为极差。这种方法的优势在于能够提升模型的归一化程度。

5.3 企业合并

对前 5 年造价公司的行业分布情况进行统计得到表 5，有一些行业的数量很多，有一些行业的数量很少，对于数量太少的行业进行分析是不合理的，因此我们将教育、居民服务、修理和其他服务业、卫生和社会工作、住宿和餐饮业和综合这几个行业与普通居民的公共生活服务息息相关，所以将这几个行业进行合并成一个新的行业——公共服务业。

而一些行业内公司足够多但是却没有发生过造假的行业，无法通过采样处理使数据平衡，也不能进行机器学习训练，因此把它看做一分类（One Class Learning）

或异常检测（Novelty Detection）问题。这类方法的重点不在于捕捉类间的差别，而是为其中一类进行建模，经典的模型包括 One-class SVM 等，此类的行业有租赁和商务服务业。

表 5 前 5 年造假公司的行业分布情况统计表

FLAG	0						1						
次数 / 行业	1	2	3	4	5	总计	1	2	3	4	5	总计	总计
采矿业	72	72	72	74	74	364	0	1	1	0	1	3	367
电力、热力、燃气及水生产和供应业	109	113	113	117	118	570	0	0	2	1	1	4	574
房地产业	111	110	110	111	115	557	0	3	3	3	1	10	567
建筑业	84	90	93	95	97	459	0	0	0	1	4	5	464
交通运输、仓储和邮政业	84	91	95	99	104	473	0	0	2	1	1	4	477
教育	9	8	9	9	9	44	0	1	0	0	0	1	45
金融业	91	95	106	113	121	526	1	1	2	5	2	11	537
居民服务、修理和其他服务业	0	1	1	1	1	4	0	0	0	0	0	0	4
科学研究和技术服务业	31	43	47	52	62	235	0	1	1	0	1	3	238
农、林、牧、渔业	39	39	40	39	40	197	1	1	1	2	2	7	204
批发和零售业	152	161	161	166	171	811	2	2	4	2	4	14	825
水利、环境和公共设施管理业	44	52	51	56	68	271	0	0	2	1	1	4	275
卫生和社会工作	11	12	12	12	12	59	0	0	0	0	0	0	59
文化、体育和娱乐业	48	60	59	62	62	291	0	0	1	0	1	2	293
信息传输、软件和信息技术服务业	246	263	265	298	330	1402	0	3	7	5	4	19	1421
制造业	1880	2143	2193	2363	2633	11212	6	14	29	17	25	91	11303
住宿和餐饮业	9	9	9	9	10	46	0	0	0	0	0	0	46
综合	16	16	17	16	17	82	0	1	0	1	0	2	84
租赁和商务服务业	48	55	55	59	60	277	0	0	0	0	0	0	277
总计	**3084**	**3433**	**3508**	**3751**	**4104**	**17880**	**10**	**28**	**55**	**39**	**48**	**180**	**18060**

5.4 特征选择

针对本文对上市公司的财务数据进行特征选取，以制造业为例，我们采用嵌入法来得到每个行业是否造假的有效特征。

常用的嵌入法有两种：

一是基于惩罚项的特征选择法，即用正则 L_1 范数作为惩罚项。L_1 范数不但可以降低过拟合风险，还可以使求得的 ω 有较多分量为 0。所以当希望减少特征的维度以用于其他分类器时，可以选择不为 0 的系数所对应的特征．本文采用基于 LASSO 回归、逻辑回归、SVM 模型的惩罚项特征选择法来选择特征，计算结果如表 6 所示。

表 6 基于惩罚项的特征选择表

LASSO 回归模型	逻辑回归模型	SVM 模型
CAP_FIX_RATIO	CASH_C_EQUIV	T_COMPR_INCOME
INTAN_A_TA	PAID_IN_CAPITAL	PAID_IN_CAPITAL
WORK_CAPITAL	'DEFER_TAX_ASSETS'	OPA_PROFIT
IC	INT_FREE_NCL	AR
VAL_CHG_PROFIT	ROE_YOY	NI_CUT_NI
EQUITY_RATIO	C_PAID_FOR_TAXES	COMPR_INC_ATTR_M_S
ROE_CUT	INCOME_TAX	CAP_FIX_RATIO
TL_TEAP	OPA_PROFIT	EBIT
ROE_A	C_FR_OTH_OPERATE_A	REV_PS
N_CF_OPA_OP	OPER_PROFIT_YOY	N_CF_OPA_OP
COGS_TR	OTH_COMPR_INCOME	INT_FREE_CL
NI_ATTR_P_CUT	N_CF_OPER_A_PS	S_RESER_PS
EBIT	NA_PS_YTD	N_WORK_CAPITAL
T_REV_PS	N_C_IN_CASH_PS	FOREX_EFFECTS
PAID_IN_CAPITAL	N_CF_FR_FINAN_A	SELL_EXP
REV_PS	FINAN_EXP	N_INCOME_ATTR_P
EBIAT	OTH_COMPRE_INCOME	C_FR_OTH_OPERATE_A
INT_FREE_NCL	NI_YOY	EBIAT
INT_DEBT	TL_TEAP	T_FIXED_ASSETS
N_WORK_CAPITAL	S_RESER_PS	INT_RECEIV
OTH_COMPRE_INCOME	N_CF_OPA_OP	FINAN_EXP
NOPERATE_EXP	COMPR_INC_ ATTR_P	ADMIN_EXP
INT_CL	ROE_CUT	SQUICK_RATIO
ID_IC	BTAX_SURCHG_TR	ROE_CUT
OTH_COMPR_INCOME	T_PROFIT_YOY	N_CF_FR_FINAN_A
R_TR	ROE_A	OTH_RECEIV

续上表

FIXED_A_TA	N_WORK_CAPITAL	NOPERATE_EXP
INT_FREE_CL	NOPERATE_EXP	INT_PAYABLE
T_FIXED_A_TA	INT_FREE_CL	BTAX_SURCHG_TR
INVEN_TA	T_FIXED_ASSETS	NI_YOY

二是基于树模型的特征选择法，这种方法能够用来计算特征的重要程度，因此可以用来去除不相关的特征．本文采用随机森林、GBDT 模型，XGBoost 模型和 LightGBM 的树模型特征选择法来选择特征，计算结果如表 7 所示。

表 7 基于树模型的特征选择表

随机森林模型	GBDT 模型	XGboost 模型	LightGBM 模型
T_COGS	OTH_COMPRE_INCOME	GROSS_PROFIT	GAIN_INVEST'
NOTES_PAYABLE	AP	AIL_TR	AR'
N_CF_OPERATE_A	C_RESER_PS	MINORITY_INT	INT_PAYABLE
ASSETS_IMPAIR_LOSS	NCL_TA	C_OUTF_FR_INVEST_A	RETAINED_EARNINGS
N_CF_FR_INVEST_A	INTAN_A_TA	NOPERATE_EXP	MINORITY_INT
N_INCOME	INVENTORIES	T_FIXED_A_TA	T_NCL
SURPLUS_RESER	INT_FREE_NCL	N_CE_BEG_BAL	INVEST_INCOME
CIP	NI_YOY	WORK_CAPITAL	A_J_INVEST_INCOME
DILUTED_EPS	MINORITY_INT	T_COMPR_INCOME	C_FR_CAP_CONTR
C_PAID_TO_FOR_EMPL	N_CE_BEG_BAL	REVENUE_YOY	GOODWILL
PAID_IN_CAPITAL	T_PROFIT	ROE_YOY	N_CE_END_BAL
NOPERATE_EXP	BASIC_EPS	C_INF_FR_OPERATE_A	OTH_RECEIV
C_PAID_DIV_PROF_INT	REFUND_OF_TAX	DEFER_REVENUE	C_FR_OTH_OPERATE_A
N_CF_OPA_OP	BIZ_TAX_SURCHG	N_CF_OPA_YOY	C_OUTF_FR_INVEST_A
T_LIAB_EQUITY	BTAX_SURCHG_TR	GOODWILL	N_CF_FR_FINAN_A
T_REVENUE	T_CA	ASSETS_DISP_GAIN	NOPERATE_INCOME
DEFER_TAX_ASSETS	FCFF	FCFF_PS	NOTES_PAYABLE
MINORITY_GAIN	FOREX_EFFECTS	NOPL_TR	TAXES_PAYABLE
T_SH_EQUITY	C_FR_OTH_FINAN_A	T_SH_EQUITY	REFUND_OF_TAX
C_FR_OTH_OPERATE_A	T_COGS	N_CF_OPER_A_PS	OTH_PAYABLE
C_PAID_FOR_TAXES	NOTES_PAYABLE	DAYS_AP	C_PAID_INVEST
LT_AMOR_EXP	INT_PAYABLE	COGS	DEFER_TAX_ASSETS
CASH_C_EQUIV	C_PAID_FOR_TAXES	C_FR_OTH_INVEST_A	ADVANCE_RECEIPTS
ST_BORR	PAYROLL_PAYABLE	N_CF_OPA_PROPT	PREPAYMENT
T_CA	ASSETS_IMPAIR_LOSS	DILUTED_EPS	LT_AMOR_EXP
N_CE_BEG_BAL	T_NCL	FCFF	DISP_FIX_ASSETS_OTH
INCOME_TAX	CASH_C_EQUIV	ADV_R_R	FOREX_EFFECTS
MINORITY_INT	T_SH_EQUITY	T_LIAB	PAID_IN_CAPITAL
N_CE_END_BAL	DILUTED_EPS	T_ASSETS	ASSETS_IMPAIR_LOSS
RETAINED_EARNINGS	RETAINED_EARNINGS	ASSET_LIAB_RATIO	NOPERATE_EXP

5.4.1 确定与财务数据造假相关的数据指标

通过特征选择发现，各种模型选择的前 30 个指标都有一定的重合度。在此基础上，挑选出 7 个模型共同确定的数据指标，并进行分类，得到结果如下：

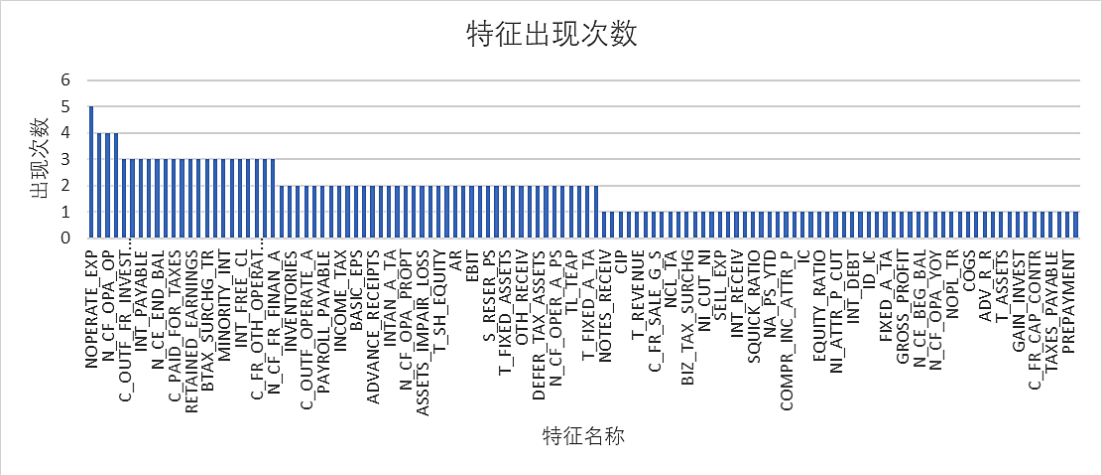

图 4 特征因子出现次数

统计所有方法获取的特征，将出现次数大于等于 3 次的特征挑选为最终的指标，共 23 个，如表 8 所示。

表 8 制造业的特征选择示意表

NOPERATE_EXP	T_COMPR_INCOME	N_CF_OPA_OP	PAID_IN_CAPITAL
OTH_COMPR_INCOME	INT_PAYABLE	NOTES_PAYABLE	N_CE_END_BAL
C_PAID_FOR_TAXES	DILUTED_EPS	RETAINED_EARNINGS	WORK_CAPITAL
INT_FREE_NCL	MINORITY_INT	FOREX_EFFECTS	INT_FREE_CL
C_FR_OTH_OPERATE_A	ROE_CUT	N_CF_FR_FINAN_A	N_WORK_CAPITAL
C_OUTF_FR_INVEST_A	REFUND_OF_TAX	BTAX_SURCHG_TR	

除了租赁和商务服务业需要另外分析外，对其他行业重复上述操作，就可以选择出每个行业的数据指标。对租赁和商务服务业单独进行 OneClass SVM 异常点分析，再根据每个指标的相关系数进行排序，得到前 25 个特征，将其作为该行业的数据指标，如表 9 所示。

表 9 租赁和商务服务业特征选择示意表

OP_TR	GROSS_MARGIN	NI_CUT_NI	EPS	OP_PS
ROE	CA_TURNOVER	N_CF_FR_INVEST_A	INVENTORIES	TRE_TA
OPA_P_TR	N_CF_IA_PROPT	T_RE_PS	ROA_EBIT	AP
ROA	C_RCVRY_A	RE_PS	CURRENT_RATIO	NOTES_PAYABLE
T_FIXED_A_TA	FIXED_A_TA	QUICK_RATIO	SQUICK_RATIO	CASH_CL

5.4.2 数据指标共性分析

统计所有行业的数据指标出现次数，如图 5 所示，我们挑选出现次数大于和等于 4 次的指标，并将挑选出来的指标进行分类，分成三大类。

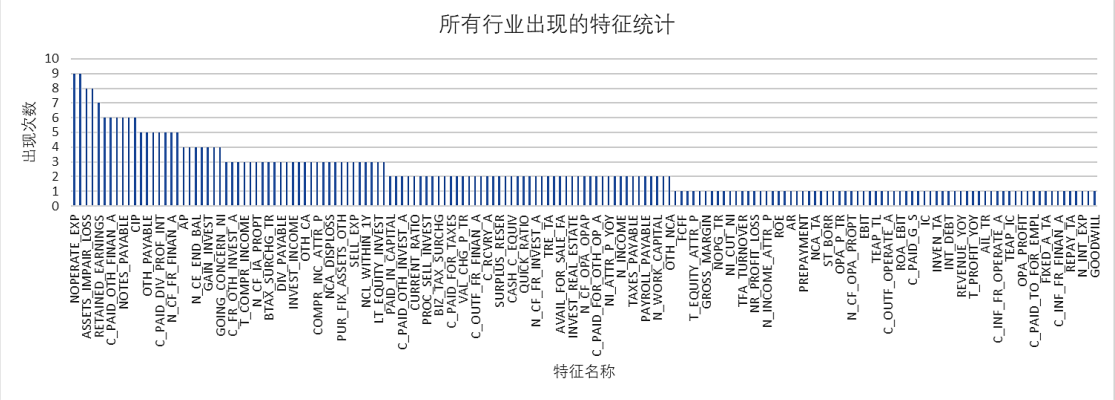

图 5 所有行业的数据指标出现次数统计图

现金流量是评价投资方案经济效益的必备资料，包括：（1）现金流出：现金流出是投资项目的全部资金支出。（2）现金流入：现金流入是投资项目所发生的全部资金收入。运营能力是指企业基于外部市场环境的约束，通过内部人力资源和生产资料的配置组合而对财务目标实现所产生作用的大小。盈利能力是指企业获取利润的能力，也称为企业的资金或资本增值能力，通常表现为一定时期内企业收益数额的多少及其水平的高低。

表 10 指标分类示意表

特征类别	变量名	含义
现金流量	NOPERATE_EXP	营业外支出
	INCOME_TAX	所得税费用
	C_PAID_OTH_FINAN_A	支付其他与筹资活动有关的现金
	C_OUTF_FR_INVEST_A	投资活动现金流出小计
现金流量	OTH_PAYABLE	其他应付款
	INT_PAYABLE	应付利息
	AP	应付账款
	FOREX_EFFECTS	汇率变动对现金及现金等价物的影响
	N_CF_FR_FINAN_A	筹资活动产生的现金流量净额
	C_PAID_DIV_PROF_INT	分配股利、利润或偿付利息支付的现金
	C_FR_OTH_OPERATE_A	收到其他与经营活动有关的现金
	N_CE_END_BAL	期末现金及现金等价物余额
	GAIN_INVEST	取得投资收益收到的现金
	INVENTORIES	存货
营运能力	ASSETS_IMPAIR_LOSS	资产减值损失
	T_PROFIT	利润总额

续上表

营运能力	NOTES_PAYABLE	应付票据
	CIP	在建工程
	MINORITY_INT	少数股东权益
盈利能力	DILUTED_EPS	稀释每股收益
	RETAINED_EARNINGS	未分配利润
	OPERATE_PROFIT	营业利润
	ASSETS_DISP_GAIN	资产处置收益
	REFUND_OF_TAX	收到的税费返还
	GOING_CONCERN_NI	持续经营净利润

5.4.3 数据指标差异性分析

以制造业和房地产业为例，不在上述的因子如表 11 和表 12 所示。

影响制造业的三大主要因子：外部融资条件、企业盈利和投资回报预期，前两者代表企业投资能力，后者代表企业投资意愿。选出的特征出现差异的原因主要是因为传统制造业生产方式。企业以生产大批量、单一产品为手段达到实现规模经济的目的，从而形成一种“大批量——低成本”的循环模式。传统制造业主要以固定资产、存货等实体资产产生盈利，业绩稳定，增长趋势可预测性强，企业价值估值相对容易，主要侧重于债务融资，其中选出来的因子更多地反映企业的负债能力。

表 11 制造业中相较表 10 有区别的指标

变量名	含义
T_COMPR_INCOME	综合收益总额
N_CF_OPA_OP	经营活动产生的现金流量净额/营业利润
PAID_IN_CAPITAL	实收资本(或股本)
OTH_COMPR_INCOME	其他综合收益
N_CE_END_BAL	期末现金及现金等价物余额
WORK_CAPITAL	营运资本
BTAX_SURCHG_TR	营业税金及附加/营业总收入
INT_FREE_NCL	无息非流动负债
INT_FREE_CL	无息流动负债
N_WORK_CAPITAL	净营运资本
ROE_CUT	净资产收益率(扣除摊薄)

房地产业选出的特征出现差异的原因主要是因为以下几点：

（1）房地产企业在建设过程中要投入大量的建设资金，因此，建设阶段最主要的支出就是工程款支出，工程款支出中归属于持有物业的部分计入在建工程项目，归属于销售物业的部分计人存货项目；

（2）由于预售制度的存在，房地产企业一般在建设阶段采用期房销售，因此，此阶段房地产企业会取得大量的预售房款，这是房地产企业最主要的现金流入项目，而预售房款应计入预收账款项目；

（3）在建设阶段，由于开发产品尚未竣工，企业不能结转收入和成本，而大量管理费用和销售费用的支出，造成此阶段企业呈现亏损状态；

（4）随着建设阶段工程进度的加快，房地产企业取得的开发贷款在此阶段将陆续到位，企业贷款规模不断加大。

表 12 房地产业中相较表 10 有区别的指标

变量名	含义
N_TAN_ASSETS	有形净资产
CURRENT_RATIO	流动比率
VAL_CHG_PROFIT	价值变动净收益
SELL_EXP_TR	销售费用/营业总收入
C_PAID_FOR_TAXES	支付的各项税费
QUICK_RATIO	速动比率
BIZ_TAX_SURCHG	营业税金及附加
VAL_CHG_P_TR	价值变动净收益/营业总收入
N_CF_OPA_PROPT	经营活动产生的现金流量净额占比
TEAP_TL	归属于母公司的股东权益/负债合计
T_RE	留存收益
N_CF_FR_FINAN_A	筹资活动产生的现金流量净额
LT_EQUITY_INVEST	长期股权投资
GAIN_INVEST	取得投资收益收到的现金
C_PAID_FOR_DEBTS	偿还债务支付的现金
NCA_DISPLOSS	非流动资产处置损失
N_CE_BEG_BAL	加:期初现金及现金等价物余额

通过以上分析可以发现，各个行业挑选的特征基本上能看出该行业的资金使用特点，其他行业将不再赘述。

6 问题二：基于 Stacking 模型的造假公司筛选

根据上文筛选出来的数据指标，用 F1-score 和 AUC 作为评价指标，使用各种机器模型算法进行训练，如表 13 所示。

表 13 各个机器模型初步训练效果

模型	F1-score	AUC
逻辑回归模型	0.99	0.50
支持向量机模型	0.99	0.50
随机森林模型	0.99	0.50

续上表

GBDT 模型	0.98	0.54
XGBoost	0.99	0.50
LightGBM	0.99	0.50

从上表可以看出，每个模型的 F1-score 都很高，但是 AUC 只有一半的水平，AUC 是 ROC 曲线下覆盖的总面积，数值范围为[0.5,1]。分类器性能越好，ROC 曲线越接近左上角，AUC 的值越接近于 1；分类器性能越差，ROC 曲线越接近对角线，AUC 的值越接近于 0.5。

这是由于 AUC 希望训练一个尽量不误报的模型，也就是知识外推的时候倾向保守估计，而 F1-score 希望训练一个不放过任何可能的模型，即知识外推的时候倾向激进，这就是这两个指标的核心区别，样本不平衡往往会导致机器模型误判而导致 AUC 偏低。

6.1 数据采样

通过应用一些欠采样或过采样技术来处理失衡样本。欠采样就是对多数类进行抽样，保留少数类的全量，使得两类的数量相当，过采样就是对少数类进行多次重复采样，保留多数类的全量，使得两类的数量相当。但是，这类做法也有弊端，欠采样会导致我们丢失一部分的信息，可能包含了一些重要的信息，过采样则会导致分类器容易过拟合。当然，也可以是两种技术的相互结合。

通过采样共获得 16818 个过采样样本、136 个欠采样样本、16818 个 SMOTE 采样样本，并且对每一种采样方法进行机器学习，评价每一种采样方法效果，评价方法为 F1-score 和 AUC，如表 14~表 19 所示。

表 14 逻辑回归模型采样效果

采样方法	F1-score	AUC
不采样	0.99	0.50
过采样	0.65	0.50
欠采样	0.50	0.50
SMOTE 采样	0.62	0.50

表 15 支持向量机模型采样效果

采样方法	F1-score	AUC
不采样	0.99	0.50
过采样	0.60	0.50
欠采样	0.01	0.50
SMOTE 采样	0.55	0.50

表 16 随机森林模型采样效果

采样方法	F1-score	AUC
不采样	0.99	0.50
过采样	0.99	0.50
欠采样	0.65	0.50
SMOTE 采样	0.98	0.50

表 17 GBDT 模型采样效果

采样方法	F1-score	AUC
不采样	0.98	0.51
过采样	0.94	0.52
欠采样	0.66	0.52
SMOTE 采样	0.89	0.66

表 18 XGBoost 模型采样效果

采样方法	F1-score	AUC
不采样	0.99	0.50
过采样	0.99	0.50
欠采样	0.60	0.50
SMOTE 采样	0.98	0.53

表 19 LightGBM 模型采样效果

采样方法	F1-score	AUC
不采样	0.99	0.50
过采样	0.99	0.50
欠采样	0.64	0.50
SMOTE 采样	0.98	0.54

经过对比分析，发现采样处理之后数据发现，经过 SMOTE 采样可以提升 AUC，并且保证 F1-score 不会骤降，初步推断是由于模型还未调整超参数，所以先用 SMOTE 采样处理数据。

6.2 寻找最优超参数

建模时先固定每个参数的初始值，再设定其调参范围，进行网格搜索和交叉验证寻找最优化结果。其中设置的初始值、范围和调参结果见各算法框架参数结果详情表，本文模型优化评价指标设为 F1-score 和 AUC。

在经过 SMOTE 采样之后，共获得 16818 个样本，但是通过测试发现，SVM 对于大数据样本的运行速度太慢，不适合进行调参等工作。

这是由于以下原因：（1）SVM 在样本量比较少的时候，容易抓住数据和特征之间的非线性关系（相比线性分类方法如 logistic regression，或者 linear SVM）。但是，在样本量比较多的时候，线性分类方法的劣势就要小了很多，例如可以通

过手工拆分/离散化特征来模拟非线性关系。（2）计算复杂度高。主流的算法是O(n^2)的，这样对大规模数据就显得很无力了。不仅如此，由于其存在两个对结果影响相当大的超参数（如果用 RBF 核，是核函数的参数 gamma 以及惩罚项 C），这两个超参数无法通过概率方法进行计算，只能通过穷举试验来求出，计算时间要远高于不少类似的非线性分类器。

所以本文暂不考虑支持向量机。

6.2.1 逻辑回归模型调参过程

图 6 为逻辑回归模型的学习曲线。

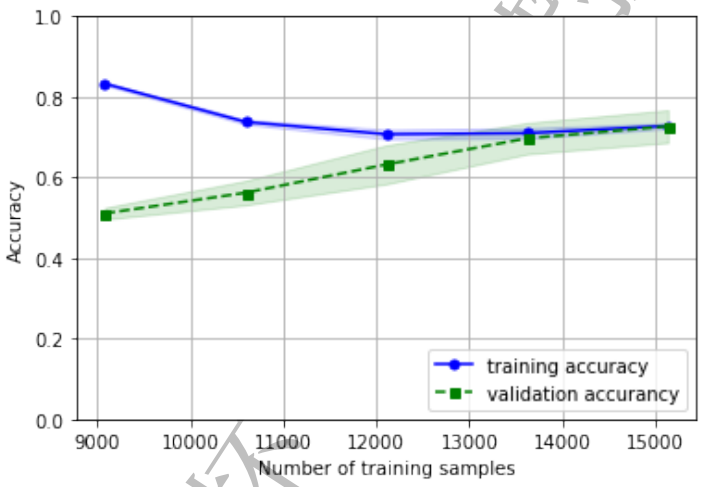

图 6 LR 模型的学习曲线

经过调参发现，无论如何调整惩罚系数 C，AUC 的最高值保持在 0.75，由此可以看出逻辑回归模型的效果只是一般，但是难以将其作为最优算法预测，继续探索其他模型。

表 20 逻辑回归模型调参

参数名称	调参范围	调参结果	调参后 F1	调参后 AUC
惩罚系数（C）	[0.1,10]	0.001	0.66	0.76

6.2.2 随机森林模型调参过程

图 7 是随机森林模型的学习曲线。

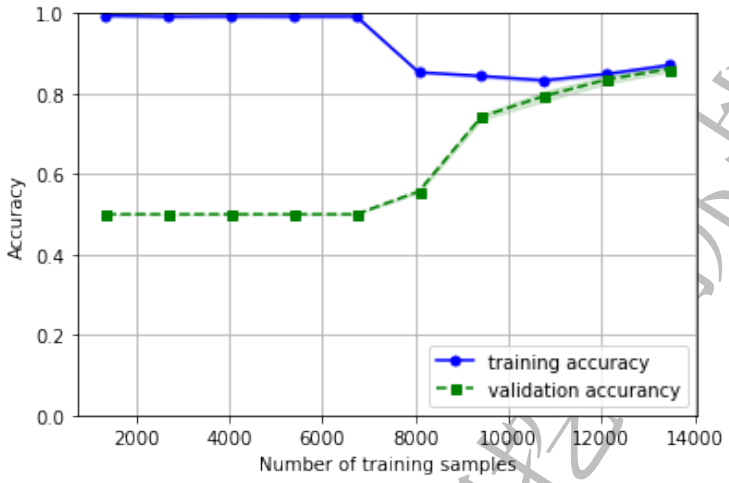

图 7 RF 模型的学习曲线

先通过网格搜索确定决策树数目（n_estimators）和决策树最大深度（max_depth），再确定分裂所需最少样本数（min_samples_split）和叶节点最少承载样本数（min_samples_leaf），可以很直观地看到随机森林模型的表现性能更好，与比线性模型相比有着很大优势。

并且在运行随机森林的时候发现代码的运行速度快，这是由于随机森林容易做成并行化方法，随机森林在训练时，树与树之间是相互独立的。

表 21 随机森林模型调参

参数名称	调参范围	调参结果	调参后 F1	调参后 AUC
决策树最大深度（max_depth）	[1,10]	211	0.85	0.80
决策树数目（n_estimators）	[10,200]	60		
分裂所需最少样本数（min_samples_split）	[10,150]	150		
叶节点最少承载样本数（min_samples_leaf）	[10,100]	250		

6.2.3 GBDT 模型调参过程

图 8 是 GBDT 模型的学习曲线。

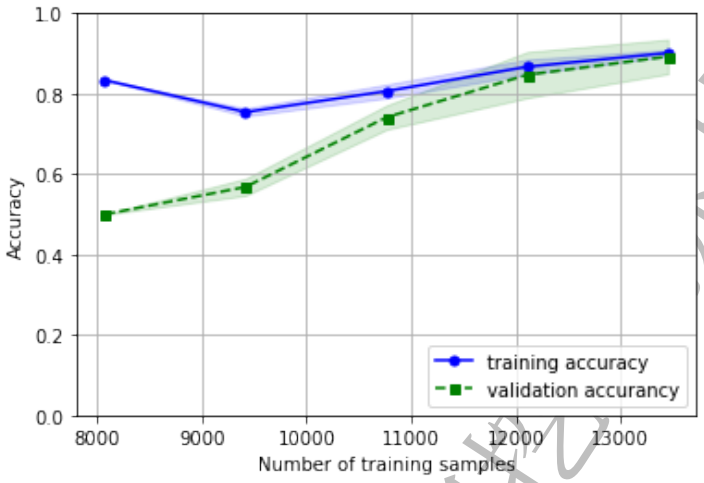

图 8 GBDT 模型的学习曲线

先确定学习率（learning_rate）为 0.1，再用网格搜索对决策树数目（n_estimators）和决策树最大数目（max_dapth）的最优值进行搜索，得到最优参数后对分类所需最少样本数（min_samples_split）和叶节点最少承载样本数（min_samples_leaf）进行调整，最后根据 AUC 对子样本占总样本的比例（subsample）进行寻优。为了防止模型过拟合，需要调高分类所需最少样本数（min_samples_split）来使学习率（learning_rate）降到 0.01。

表 22 GBDT 模型调参

参数名称	调参范围	调参结果	调参后 F1	调参后 AUC
决策树最大数目（max_dapth）	[1,15]	12	0.86	0.78
决策树数目（n_estimators）	[10,200]	60		
子样本占总样本的比例（subsample）	[0.5,1]	0.6		
学习率（learning_rate）	[0.01,1]	0.01		
分裂所需最少样本数（min_samples_split）	[10,150]	150		
叶节点最少承载样本数（min_samples_leaf）	[10,100]	200		

6.2.4 XGBoost 模型调参过程

图 9 是 XGBoost 模型的学习曲线。

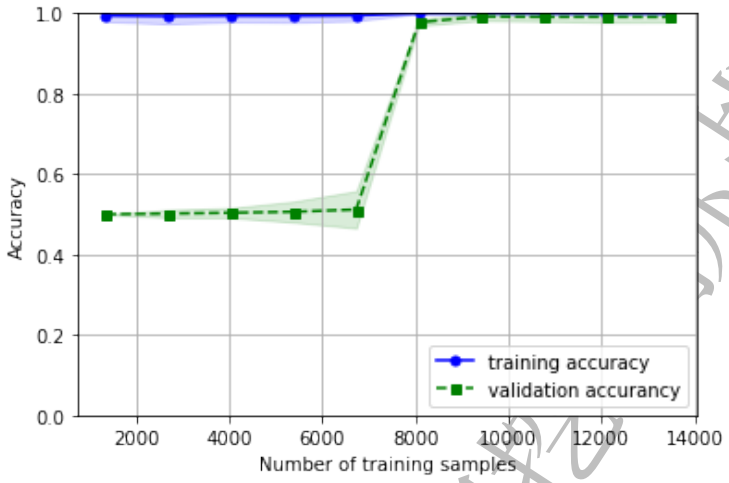

图 9 XGBoost 模型的学习曲线

跟 RF 模型和 GBDT 模型一样，先确定 0.1 为 XGBoost 模型的学习率，再调整决策树数目（n_estmators），再确定模型的决策树的最大深度（max_depth），最后调整子样本占总样本的比例（subsample），最后根据 AUC 略调整模型的学习率（learning_rate）。

表 23 XGBoost 模型调参

参数名称	调参范围	调参结果	调参后 F1	调参后 AUC
决策树最大深度（max_depth）	[1,15]	8	0.97	0.76
决策树数目（n_estmators）	[10,200]	150		
学习率（learning_rate）	[0.01,1]	0.1		
子样本占总样本的比例（subsample）	[0,2]	0.9		

6.2.5 LightGBM 模型调参过程

图 10 是 LightGBM 的学习曲线。

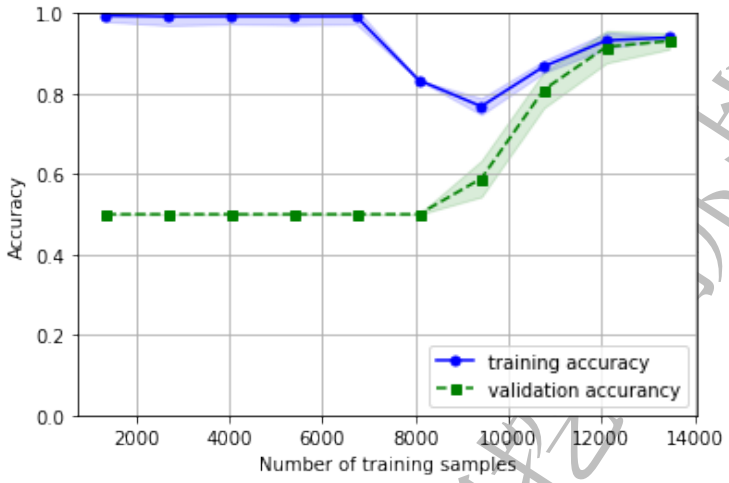

图 10 LightGBM 模型学习曲线

LightGBM 模型的参数较多，但是调参规则和每一个基于决策树的模型是差不多的，先确定 0.1 为初始学习率（Learning_rata），可以使模型有更快的收敛速度，然后通过网格搜索确定决策树数目（n_estimators）和决策树最大深度（max_depth），再确定最大叶子节点数（num_leaves）和叶节点最小样本数（min_data_in_leaf），最后调节特征子采样比例（feature_fraction）和样本子采样频率（bagging_fraction），回过头再提升叶节点最小样本数来降低学习率防止模型过拟合。

表 24 LightGBM 模型调参

参数名称	调参范围	调参结果	调参后 F1	调参后 AUC
决策树最大深度（max_depth）	[1,15]	13	0.89	0.77
决策树数目（n_estimators）	[10,200]	50		
最大叶子节点数（num_leaves）	[20,100]	90		
学习率（Learning_rata）	[0.01,1]	0.01		
叶节点最小样本数（min_data_in_leaf）	[1,80]	150		
特征子采样比例（feature_fraction）	[0.5,1]	0.7		
样本子采样频率（bagging_fraction）	[0.5,1]	0.7		

6.2.6 Stacking 模型融合

在模型融合中，第 1 层学习器使用 RF 模型、GBDT 模型、XGBoost 模型和 LightGBM 模型。

为了得到第 2 层的学习器所需的数据，采取 k 折交叉验证来划分训练集的数据，本文使用 5 折交叉验证，得到第 2 层的训练集之后，第 2 层使用逻辑回归模型进行训练，并计算融合模型的 F1-score 和 AUC，如表 25 所示。发现 Stacking 模型有更高的评分并且保持了良好的 AUC，说明模型的预测效果良好。

表 25 Stacking 模型融合

Stacking 模型融合			
第一层	第二层	F1-score	AUC
RF 模型	LR 模型	0.96	0.79
GBDT 模型			
XGBoost 模型			
LightGBM 模型			

绘制 5 个模型和 Stacking 模型的 ROC 曲线进行比对，如图 11 所示：

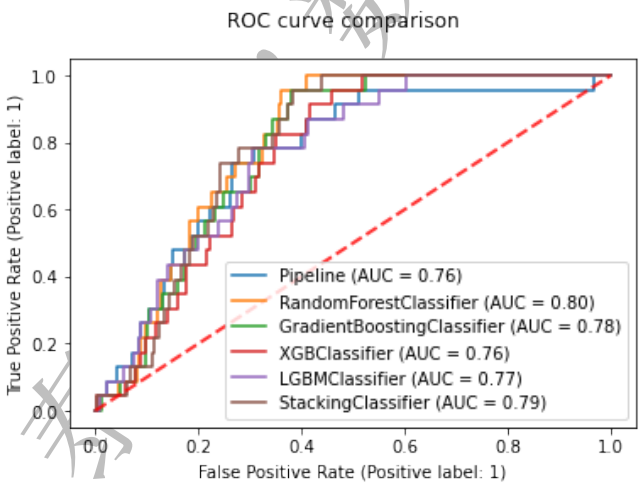

图 11 5 个模型和融合模型的 ROC 曲线

对第六年的制造业的公司进行预测，预测得到 20 家公司的财务数据造假，每个企业的编号如表 26 所示：

表 26 制造业中财务数据造假的企业

166659	376089	993505	1001540	1347287
1644007	1802653	1868870	2694256	2974487
3285030	3646698	3916009	4174154	4564768
4578172	4720778	4735999	4789867	4968617

7 问题三：寻找发生造假的公司

7.1 Stacking 集成模型进行预测

根据问题二对制造业进行预测的思路，通过 Stacking 模型，利用各个行业挑选出来的数据指标，并对模型重新进行调参后进行机器学习，预测得到每个行业发生造假的公司。

表 27 每个行业出现造假的公司个数和公司代号

行业	个数	公司代码
采矿业	5	1760451，1801564，2285062，3065911，4669213
电力、热力、燃气及水生产和供应业	4	2631311，2633527，2769232，4986359
房地产业	2	981402，3279001
建筑业	3	1892808、2893562、3889707
交通运输、仓储和邮政业	2	903243、3872511
金融业	3	79573、1857999、4763536
科学研究和技术服务业	1	3182805
农、林、牧、渔业	1	657586
批发和零售业	1	2213558
水利、环境和公共设施管理业	1	3360110
文化、体育和娱乐业	1	3262635
信息传输、软件和信息技术服务业	3	426125、2344580、3276265
公共服务	0	没有

7.2 OneClass SVM 异常点检测

对于无法分类的行业——租赁和商务服务业，采用 OneClass SVM 进行异常点检测，预测得到如表 28 的公司发生造假。

表 28 租赁和商务服务业出现造假的公司个数和公司代号

行业	个数	公司代码
租赁和商务服务业	0	没有

7.3 模型的思考和尝试

通过调参发现，每一个行业机器模型的超参数相差不大，所以提出一个猜想：是否可以使用一个融合模型将所有行业统一起来，并进行预测。假如这个猜想成立，将可以增强模型的鲁棒性和增加模型的使用范围。

特征选取的是第一问统计出来每个行业出现次数大于等于 4 次的数据指标，但是，得到的每个机器模型和融合模型的 F1-score 和 AUC 如表 29 所示，同时绘出了 ROC 曲线（图 12）。

表 29 合并行业的集成学习模型的 F1-score 和 AUC

模型	F1-score	AUC
LR 模型	0.99	0.75
RF 模型	0.99	0.77
GBDT 模型	0.99	0.78
XGBoost 模型	0.99	0.78
LightGBM 模型	0.99	0.77
Stacking 模型	0.99	0.75

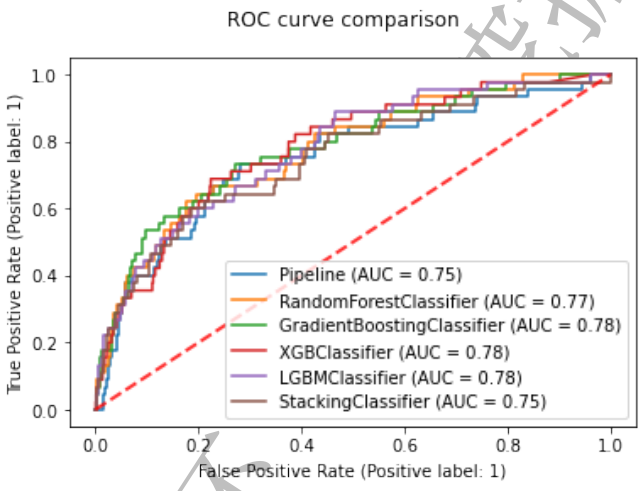

图 12 合并行业的集成学习模型的 ROC 曲线

无论怎么调整 LR 模型的惩罚系数 C，都无法再将 LR 模型的 AUC 提高，原因猜测是由于 LR 模型的预测结果呈“S”型，因此从 log(odds)向概率转化的过程是非线性的，在两端随着 log(odds)值的变化，概率变化很小，边际值太小，slope 太小，而中间概率的变化很大，很敏感。导致很多区间的变量变化对目标概率的影响没有区分度，无法确定阈值。

8 总结

本文的研究结果有以下几点：

（1） 机器学习适用于预测上市公司是否会发生财务造假。传统的预测方法可能“力不从心”的现实情况下，可以将本文的经验运用到企业财务管理的实践中。

（2） 本文详细介绍了如何选择数据因子、训练和检验机器学习模型，通过

F1-score 和 AUC 指标综合考量了各学习模型的预测准确度和稳定性，并且将众多学习模型中进行融合，从而构建一套适用的、有效的基于集成学习的机器学习分类的财务造假预测方法。

（3） 由于 LR 模型的局限性，暂时还没有办法得到一个能涵盖所有行业的统一的机器模型，或许以后可以从别的基学习器出发，寻求构建一个涵盖所有行业的机器学习，本文可以进行更多的后续研究。

参考文献

[1]东哥起飞,【Python 数据分析基础】.数据缺失值处理, https://zhuanlan.zhihu.com/p/40775756[EB/OL]. 2021-3-15.

[2]qq_24591139, imblearn 算法详解及实例, https://blog.csdn.net/qq_24591139/article/details/100518532[EB/OL]. 2021-4-25.

[3]美团算法团队, 美团机器学习实践[M], 北京, 人民邮电出版社, 2018, 27-33.

[4]王嘉欣. 机器学习方法在上市公司财务舞弊预测问题中的应用[D]. 山东大学, 2019.

[5]李星辰, 王青龙, 林国庆. 基于机器学习方法的上市公司财务预警模型对比研究[J]. 商场现代化, 2020, No.916(07):156-158.

[6]黄志刚、刘佳进、林朝颖. 基于机器学习的上市公司财报舞弊识别前沿方法比较研究[J]. 系统科学与数学, 2020, v.40(10):185-203.

[7]蒋盛益, 汪珊, 蔡余冲. 基于机器学习的上市公司财务预警模型的构建[J]. 统计与决策, 2010(09):166-167.

[8]jimofanhua0000, RF,GBDT,xgboost 调参方法整理, https://blog.csdn.net/u014465639/article/details/74351982[EB/OL], 2021-5-2.

附录

下面各表是问题一挑选出来的各个行业对公司是否造假有重要影响的因子。

附表 1 重要因子表（一）

制造业	采矿业	电力、热力、燃气及水生产和供应业	房地产业
NOPERATE_EXP	DILUTED_EPS	ASSETS_IMPAIR_LOSS	N_TAN_ASSETS
T_COMPR_INCOME	DAYS_AP	C_INF_FR_INVEST_A	CURRENT_RATIO
N_CF_OPA_OP	NOTES_RECEIV	PROC_SELL_INVEST	VAL_CHG_PROFIT
PAID_IN_CAPITAL	NOTES_PAYABLE	A_J_INVEST_INCOME	SELL_EXP_TR
C_OUTF_FR_INVEST_A	N_INCOME_ATTR_P	AR_TA	C_PAID_FOR_TAXES
OTH_COMPR_INCOME	ADVANCE_RECEIPTS	INVEST_INCOME	FOREX_EFFECTS
INT_PAYABLE	FOREX_EFFECTS	C_FR_OTH_INVEST_A	RETAINED_EARNINGS
NOTES_PAYABLE	C_PAID_OTH_FINAN_A	C_PAID_INVEST	QUICK_RATIO
N_CE_END_BAL	SELL_EXP	CAPITAL_RESER	BIZ_TAX_SURCHG
REFUND_OF_TAX	OPERATE_PROFIT	AVAIL_FOR_SALE_FA	NOPERATE_EXP
C_PAID_FOR_TAXES	C_FR_OTH_OPERATE_A	DIV_PAYABLE	VAL_CHG_P_TR
DILUTED_EPS	GAIN_INVEST	C_FR_OTH_FINAN_A	INVENTORIES
RETAINED_EARNINGS	N_CHANGE_IN_CASH	NOPERATE_EXP	N_CF_OPA_PROPT
WORK_CAPITAL	N_INCOME	COMPR_INC_ATTR_P	TEAP_TL
BTAX_SURCHG_TR	CASH_C_EQUIV	OTH_COMPR_INCOME	T_RE
INT_FREE_NCL	OTH_CA	RETAINED_EARNINGS	N_CF_FR_FINAN_A
MINORITY_INT	C_PAID_DIV_PROF_INT	OTH_COMPRE_INCOME	NOTES_PAYABLE
FOREX_EFFECTS	LT_EQUITY_INVEST	C_PAID_FOR_OTH_OP_A	LT_EQUITY_INVEST
INT_FREE_CL	INTAN_A_TA	MINORITY_GAIN	GAIN_INVEST
N_WORK_CAPITAL	AIL_TR	AP	C_PAID_FOR_DEBTS
C_FR_OTH_OPERATE_A	MINORITY_INT	SURPLUS_RESER	NCA_DISPLOSS
ROE_CUT	C_PAID_TO_FOR_EMPL	ASSETS_DISP_GAIN	N_CE_BEG_BAL
N_CF_FR_FINAN_A	OTH_NCA	T_COMPR_INCOME	
	T_REVENUE	TAXES_PAYABLE	
	C_INF_FR_FINAN_A	C_PAID_DIV_PROF_INT	
	INT_RECEIV	N_CE_END_BAL	
	T_PROFIT	INCOME_TAX	
	N_CF_IA_PROPT	ADVANCE_RECEIPTS	
	OTH_PAYABLE	C_OUTF_FR_INVEST_A	
	PAYROLL_PAYABLE	T_PROFIT	

附表 2 重要因子表（二）

公共服务业	建筑业	交通运输、仓储和邮政业	金融业
C_FR_OTH_OPERATE_A	NOPERATE_EXP	CASH_CL	C_FR_OTH_INVEST_A

续上表

N_CF_IA_PROPT	OTH_COMPR_INCOME	BTAX_SURCHG_TR	NOPERATE_EXP
AP	DISP_FIX_ASSETS_OTH	DIV_PAYABLE	NOPG_TR
INCOME_TAX	ASSETS_DISP_GAIN	TFA_TURNOVER	C_OUTF_FR_INVEST_A
OTH_PAYABLE	C_FR_OTH_OPERATE_A	C_TA	VAL_CHG_P_TR
T_RE_PS	N_CE_END_BAL	C_RCVRY_A	NOPERATE_INCOME
LT_AMOR_EXP	CASH_C_EQUIV	CA_TURNOVER	ASSETS_DISP_GAIN
GAIN_INVEST	RETAINED_EARNINGS	C_INF_FR_INVEST_A	FIXED_ASSETS
TRE_TA	DILUTED_EPS	REFUND_OF_TAX	T_COGS
CASH_CL	NOPERATE_INCOME	FA_TURNOVER	NCA_TA
SELL_EXP	C_PAID_DIV_PROF_INT	GAIN_INVEST	INTAN_A_TA
INVEST_REAL_ESTATE	N_CE_BEG_BAL	C_PAID_OTH_FINAN_A	INT_PAYABLE
T_PROFIT	T_PROFIT	NI_ATTR_P_YOY	OTH_PAYABLE
N_CF_OPERATE_A	OPERATE_PROFIT	SQUICK_RATIO	C_PAID_OTH_INVEST_A
INVENTORIES	OP_TR	NCL_WITHIN_1Y	T_PROFIT
OTH_COMPRE_INCOME	CIP	LT_BORR	OPERATE_PROFIT
OTH_CA	REFUND_OF_TAX	AP	DILUTED_EPS
PAYROLL_PAYABLE	GOING_CONCERN_NI	AR_TA	RETAINED_EARNINGS
C_OUTF_FR_INVEST_A	ASSETS_IMPAIR_LOSS	T_PROFIT_YOY	GOING_CONCERN_NI
NOPERATE_EXP	OTH_PAYABLE	TA_TURNOVER	C_INF_FR_INVEST_A
NOTES_RECEIV	MINORITY_INT	TEAP_IC	FOREX_EFFECTS
RE_PS	OTH_COMPRE_INCOME		CA_TURNOVER
T_FIXED_A_TA	DEFER_REVENUE		C_FR_OTH_OPERATE_A
OTH_NCA			C_PAID_FOR_DEBTS
NOTES_PAYABLE			

附表 3 重要因子表（三）

科学研究和技术服务业	农、林、牧、渔业	批发和零售业	水利、环境和公共设施管理业
PAID_IN_CAPITAL	T_FIXED_ASSETS	ASSETS_IMPAIR_LOSS	DISP_FIX_ASSETS_OTH
T_EQUITY_ATTR_P	T_LIAB	C_PAID_OTH_FINAN_A	ASSETS_DISP_GAIN
NOTES_PAYABLE	C_PAID_OTH_FINAN_A	OPERATE_PROFIT	INT_FREE_NCL
BTAX_SURCHG_TR	ADVANCE_RECEIPTS	BIZ_TAX_SURCHG	ID_IC
C_PAID_INVEST	FIXED_ASSETS	DILUTED_EPS	ASSETS_IMPAIR_LOSS
SURPLUS_RESER	COMPR_INC_ATTR_P	T_PROFIT	INTAN_ASSETS
PUR_FIX_ASSETS_OTH	N_CF_OPA_OP	CIP	C_PAID_DIV_PROF_INT
AR	INVEST_INCOME	BASIC_EPS	RETAINED_EARNINGS
PREPAYMENT	LT_BORR	RETAINED_EARNINGS	INCOME_TAX
DAYS_INVEN	C_OUTF_FR_INVEST_A	ST_BORR	PUR_FIX_ASSETS_OTH
PROC_SELL_INVEST	CIP	SELL_EXP	REVENUE
REVENUE	N_CF_OPA_OPAP	COMPR_INC_ATTR_P	VAL_CHG_P_TP

续上表

COGS	NOTES_RECEIV	C_OUTF_FR_INVEST_A	DEFER_REVENUE
C_OUTF_OPERATE_A	BASIC_EPS_YOY	INT_PAYABLE	TAXES_PAYABLE
T_REVENUE	REFUND_OF_TAX	C_OUTF_FR_FINAN_A	C_PAID_OTH_FINAN_A
C_PAID_G_S	DILUTED_EPS_YOY	NOPERATE_EXP	C_PAID_INVEST
INVEN_TA	DEFER_TAX_ASSETS	DIV_PAYABLE	NCA_DISPLOSS
IT_TP	REVENUE_YOY	PUR_FIX_ASSETS_OTH	INVEST_REAL_ESTATE
OPER_CYCLE	MINORITY_INT	N_CE_END_BAL	N_CF_FR_INVEST_A
C_INF_FR_OPERATE_A	C_TA	INTAN_ASSETS	INT_FREE_CL
C_FR_SALE_G_S	NI_ATTR_P_YOY	GOING_CONCERN_NI	N_CF_FR_FINAN_A
ASSETS_IMPAIR_LOSS	FINAN_EXP	MINORITY_INT	NCL_WITHIN_1Y
LT_EQUITY_INVEST	NCL_TA	N_CF_OPA_OPAP	CIP
NOPERATE_EXP	INVENTORIES	N_CF_FR_FINAN_A	N_CHANGE_IN_CASH
CIP	ASSETS_IMPAIR_LOSS	A_J_INVEST_INCOME	ADMIN_EXP
N_WORK_CAPITAL			FOREX_EFFECTS
BASIC_EPS			DILUTED_EPS
DILUTED_EPS			

附表 4 重要因子表（四）

文化、体育和娱乐业	信息传输、软件和信息技术服务业	租赁和商务服务业
FCFF	C_PAID_OTH_INVEST_A	OP_TR
T_FIXED_ASSETS	OPERATE_PROFIT	GROSS_MARGIN
EBIAT	INVEST_INCOME	NI_CUT_NI
NR_PROFIT_LOSS	OTH_CA	EPS
C_OUTF_FR_FINAN_A	C_FR_OTH_OPERATE_A	OP_PS
C_PAID_FOR_DEBTS	NCA_DISPLOSS	ROE
T_RE	CIP	CA_TURNOVER
C_FR_OTH_FINAN_A	T_PROFIT	N_CF_FR_INVEST_A
DISP_FIX_ASSETS_OTH	ASSETS_IMPAIR_LOSS	INVENTORIES
ASSETS_IMPAIR_LOSS	NOPERATE_EXP	TRE_TA
N_CF_OPER_A_PS	GOING_CONCERN_NI	OPA_P_TR
GROSS_PROFIT	C_PAID_FOR_OTH_OP_A	N_CF_IA_PROPT
EBIT	DILUTED_EPS	T_RE_PS
INT_FREE_NCL	N_INCOME	ROA_EBIT
NI_ATTR_P_CUT	ASSETS_DISP_GAIN	AP
IC	C_PAID_OTH_FINAN_A	ROA
INT_DEBT	REFUND_OF_TAX	C_RCVRY_A
DILUTED_EPS	INT_PAYABLE	RE_PS
OTH_PAYABLE	REVENUE	CURRENT_RATIO
FCFF_PS	CAPITAL_RESER	NOTES_PAYABLE
OPA_PROFIT	T_COMPR_INCOME	T_FIXED_A_TA
BASIC_EPS	C_PAID_DIV_PROF_INT	FIXED_A_TA

续上表

OPERATE_PROFIT	INCOME_TAX	QUICK_RATIO
T_PROFIT	NCL_WITHIN_1Y	SQUICK_RATIO
INT_FREE_CL	REPAY_TA	CASH_CL
C_FR_OTH_INVEST_A	GROSS_PROFIT	
N_INT_EXP	AVAIL_FOR_SALE_FA	
N_CF_FR_FINAN_A	S_RESER_PS	
	GOODWILL	