易企|僵尸企业分类系统

项目技术文档

开发团队：1903635

开发时间：2019.12.21 - 2020.05.27

目录

一、 项目简介

1. 项目背景

低生产率的企业由于竞争失败退出市场，从而把资源转配给高生产率的企业，这是一种健康的市场竞争状态，能够提高经济的资源配置效率。但现实是：存在自身经营无法存续却靠债权人或政府支持维持经营的“僵尸企业”。这些企业困而不变，僵而不死，阻碍生产率提高，降低资源配置效率，加剧产能过剩，甚至导致劣胜优汰。2013 年以来，中国经济进入了经济增长速度换挡器、结构调整阵期、前期刺激政策消化期“三期叠加”阶段。在此期间，经营不利乃至严重亏损的企业愈发增多。地方政府为了维持社会经济的稳定，给予了这些企业直接财政补贴或间接贷款贴息。但许多企业由于自身发展力不足，在接收补助后也无法力挽狂澜，变成了空有虚壳的“僵尸企业”[1]。僵尸企业无疑在妨碍我国的经济发展，如何清理僵尸企业，成为了我国经济结构转型时期的棘手问题。

清理僵尸企业首先要识别僵尸企业。僵尸企业的识别标准[2]主要有官方标准、CHK 标准，FN-CHK 标准以及各类 FN-CHK 修正标准等，官方标准定义为不符合国家能耗、环保、质量、安全等标准，持续亏损三年以上且不符合结构调整方向；已停产、半停产、连年亏损、资不抵债要靠政府补贴和银行续贷维持经营的企业。CHK 标准的核心是企业是否接受信贷补贴，FN-CHK 标准则包含“真实利润原则”以及“常青贷款原则”，相关的 FN-CHK 修正方法大部分都是对以上两类标准的修正，对企业利润与资产负债率等指标进行调整，将企业的经营管理费用、净资产水平、企业效率和创新等指标引入僵尸企业的识别标准体系中，力求从更加多维的层次反映僵尸企业的经营特征[3]。

通过各类标准分类识别僵尸企业均有其局限性。在互联网步入大数据时代后，“依托大数据识别、监管僵尸企业”越来越受到重视。在大数据时代，可以更便捷地获得企业的工商、司法、经营、上市、知识产权、舆论等多维度数据。通过对企业从业人数、成立年限、注册资本、营业收入、风险信息，行政处罚、纳税信用等级、黑名单、上市信息、电商信息等数据关联处理，按需进行权重分割，并对这些数据进行综合分析，能够构建企业全息画像，整体评估一个企业地综合价值，并更好地勾勒出企业的经营变化情况。

本项目通过综合分析企业的各项数据，构建企业全息画像，建立僵尸企业分类系统。市场监管部门可以通过该系统对中国企业进行监管，及时识别并处理僵尸企业。

2. 僵尸企业的表现形式

僵尸企业是指虽然可以产生现金流，但是扣除运营成本和固定成本之后，最多只能支付贷款利息，而无力偿还贷款本金的企业。这些企业依赖政府补贴或银行贷款来续存，阻碍生产率的提高，降低资源配置效率，加剧产能过剩，甚至导致劣胜优汰的现象。考虑到我国金融制度不健全，僵尸企业倾向于采用企业间商业信用的非正式融资方式进行融资，这不仅会损害合作企业，还会拖累银行成为“僵尸银行”。目前来看，僵尸企业正妨碍我国的经济发展。解决好僵尸企业才能促进经济动能转换以及高质量发展。

要清理僵尸企业先要识别僵尸企业。2015 年 12 月 9 日，李克强总理在国务院常务会议上首次对“僵尸企业”提出了具体的清理标准，即要对持续亏损 3 年以上且不符合结构调整方向的企业采取资产重组、产权转让、关闭破产等方式予以“清理”。

僵尸企业的官方标准被定义为：如果一个企业连续三年利润为负，则识别这家企业

为僵尸企业。识别僵尸企业的官方标准虽然直观易操作，但也存在问题。官方标准会把一些运转状况良好且发展潜力较大的企业错误识别为僵尸企业，比如美国亚马逊公司自创立以来连续 20 年都是亏损，直到 2015 年才实现盈利。

CHK 识别方法从信贷的角度来理解僵尸企业，如果一个企业为自己债务所支付的利息非常低，甚至低于采用市场最低利率所要支付的利息，那么这个企业与银行之间的借贷关系就是非正常的，该企业极有可能依靠银行贷款来生存。该识别方法也存在问题。现实情况中有一些企业由于经营状况良好，违约风险低，往往能从银行获得非常优惠的贷款；另外，政府也会向成长型企业发放低利息贷款。CHK 识别方法很有可能将这些企业识别为僵尸企业。CHK 方法还可能识别不出真正的僵尸企业。有些企业利率看似正常，但盈利水平实际很低，利润不足以支付贷款利息，全靠向银行“借信贷还旧息”，这些企业往往会成为 CHK 检测下的漏网之鱼。

通过对 CHK 方法的改进，又提出了 FN-CHK 检测标准。该检测标准不会漏掉那些盈利差还增加外部贷款的企业。

在 FN-CHK 检测标准基础上，我国也提出了“人大国发院标准”：如果一个企业在 t 年和 $t-1$ 年都被 FN-CHK 识别为僵尸企业，那么该企业在 t 年被识别为僵尸企业。

以上模型标准对识别我国僵尸企业既有一定的参考价值又存在局限性。在我国，政府干预对企业经营有不可忽视的作用，并且我国的可转债利率远低于其它债券利率，使用可转债利率这一单一指标来计算所有债券应支付的最低利息，可能会低估“利息支付下限”，进而低估僵尸企业数量。

3. 应用对象

2016 年 9 月，国家工商总局发布《工商总局关于新形势下推进监管方式改革创新的意见》（工商企监字〔2016〕185 号），指出要“依托大数据加强监管”，“充分发挥大数据在制定完善新型市场监管制度和政策中的作用，搜集掌握经营者、消费者和社会公众的反应，跟踪监测有关制度和政策的实施效果”，“在工商登记、企业监管、网络交易、竞争执法、消费维权等领域率先开展大数据示范应用”。

本项目通过对企业相关数据的关联处理，并对这些数据进行综合分析，构建出僵尸企业分类模型，整体评估企业的综合价值，勾勒企业的经营变化情况，对僵尸企业进行识别和分类。市场监管部门可以使用该模型识别僵尸企业，对僵尸企业进行及时的管理。

4. 应用环境

在互联网逐渐步入大数据时代后，可以更便捷的获得企业的工商、司法、经营、上市、知识产权、舆情等多维度数据。依托大数据对企业进行管理越来越受到市场监管部门的重视。识别和处理僵尸企业是市场监管部门面对的一大难题，本项目建立的僵尸企业分类模型，使政府部门能够及时识别与淘汰僵尸企业。

本系统以 C/S 架构模型设计，用户只需要使用浏览器访问本系统即可完成僵尸企业的分类与识别。由于算法运行速度很快，因此对服务器的性能也没有太高的要求，只需要安装 Python 3.7 和相应的第三方库即可。

二、 问题分析与解决思路

该项目主要分为分类算法模型和 Web 可视化界面两部分进行开发。

1. 分类算法模型

分类算法模型部分的总体开发流程图如下所示：

图 1：分类算法模型的总体开发流程图

首先我们对企业提供的原始数据集进行数据预处理，主要包括去除异常值、填充缺失值。在此基础上，我们进行特征工程，主要从业务和统计这两个角度构造和设计特征。对于数据集中的离散特征，我们进行 One-hot encoding。为了提升算法的运行效率，我们使用 RFECV（递归特征消除）对构造的特征进行初步的筛选。

在模型算法部分，我们首先选取在该数据集上表现较好的算法，并单独对其进行五折交叉验证网格搜索参数调优。在此基础上，我们对多个算法进行加权投票融合，以取得更好的效果。

考虑到原始数据集中存在一部分标签未知的数据，我们还需进行数据融合，我们利用在已知标签上训练的模型，预测未知标签的数据，每次选取置信度最大的数据，将预测标签作为真实标签加入训练数据，重新进行训练。反复迭代，直到所有数据都加入了训练数据集。通过该步骤充分利用企业给定的原始数据集。

2. Web 可视化界面

Web 可视化界面的总体开发流程图如下所示：

图 2：Web 可视化界面的总体开发流程图

首先我们根据企业的要求设计相应的页面，在此基础上进行页面的开发，编写网页模版。我们需要对 URL 进行配置，根据用户访问的 URL 路由到相应的视图处理函数。通过对视图函数的编写，正确处理用户请求，并向用户返回期望的页面。

3. 前后端交互流程

前后端交互流程如下图所示：

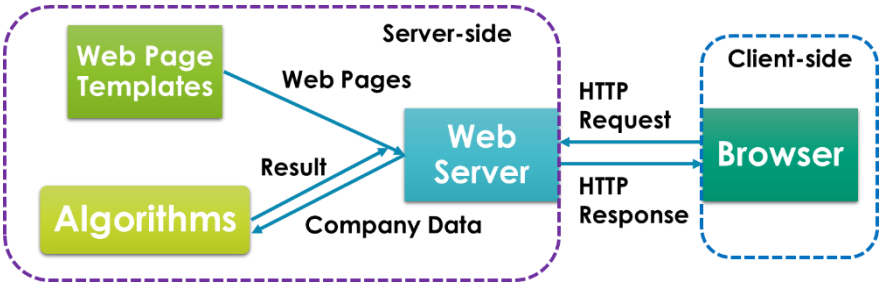

图 3：前后端总体交互流程图

当用户在浏览器端访问我们的分类系统，浏览器向服务器发送一个 HTTP 请求，服务器根据用户访问的 URL，选择合适的视图处理函数处理请求。视图函数将用户的请求数据发送给算法模型，并从算法模型得到分类结果。在此基础上，利用分类结果填充预先编写好的页面，生成最终页面，以 HTTP 响应的方式返回给用户。

三、 技术路线与开发工具

1. 分类算法

图 4：分类算法部分技术结构图

在数据预处理部分我们主要使用 Pandas 和 Numpy 对数据进行去除异常值、填充缺失值。

在特征工程部分，我们使用 Pandas 和 Numpy 构造业务和统计特征，并利用 Scikit-learn 中的 One-hot encoding 模块对离散特征进行编码，RFECV 模块进行特征的筛选。

在参数调优部分，我们主要使用 Scikit-learn 中的 GridSearchCV 模块进行最优参数的五折交叉验证网格搜索。

而在模型融合部分，我们使用 Scikit-learn 中的 VotingClassifier 对决策树、随机森林、XGBoost 三种模型进行加权投票融合。

在数据融合部分，我们利用在已知标签上训练的模型，预测未知标签的数据，每次选取置信度最大的数据，将预测标签作为真实标签加入训练数据，重新进行训练。反复迭代，直到所有数据都加入了训练数据集。通过该步骤充分利用企业给定的原始数据集。

2. Web 可视化界面

Web 可视化部分我们主要使用 Django 框架搭建。Django 框架的基本结构如下所示：

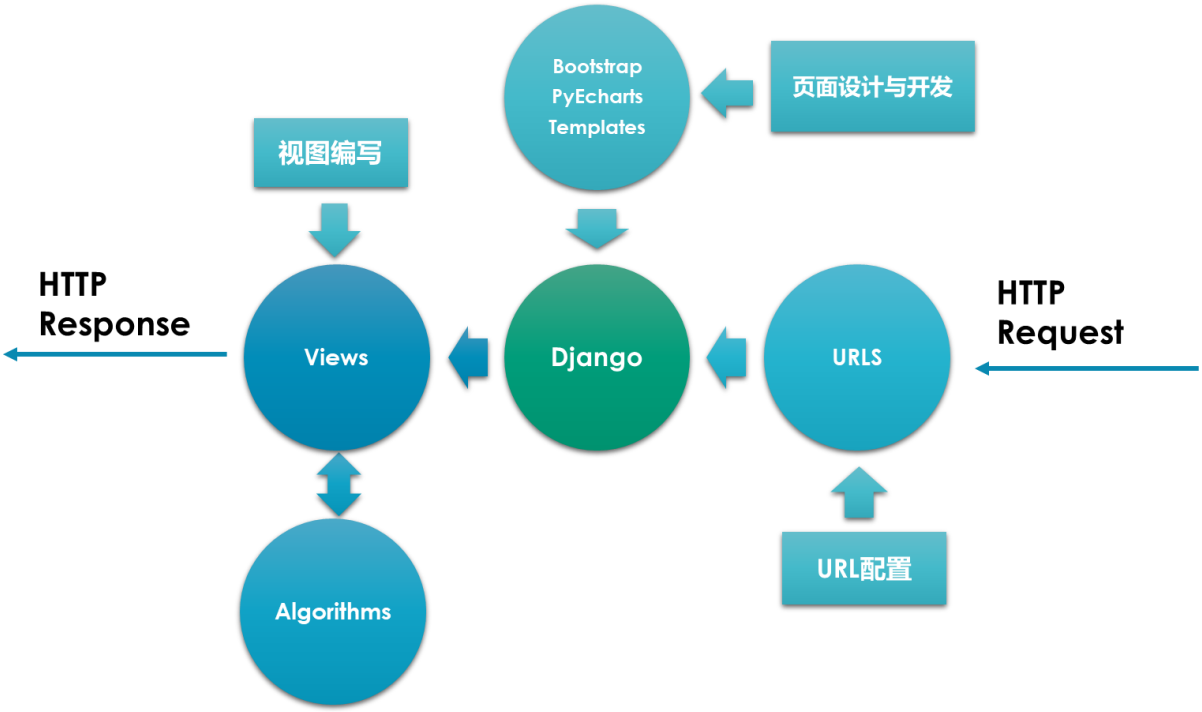

图 5：Django 框架基本结构

当 Django 服务器收到一个 HTTP 请求，将对其请求的 URL 进行路由，选择合适的视图处理函数处理请求。视图函数会调用预先训练好的算法模型，并将用户数据输入模型当中得到输出结果。

我们使用 Bootstrap 搭建网页基本模版，PyEcharts 绘制模版中的可视化图表部分。在模版中插入占位符，当 Django 从算法模型得到输出结果后，会对模版中的占位符进行填充，得到最终的页面返回给用户。

3. 项目的整体结构

项目的整体结构和交互流程如下图所示：

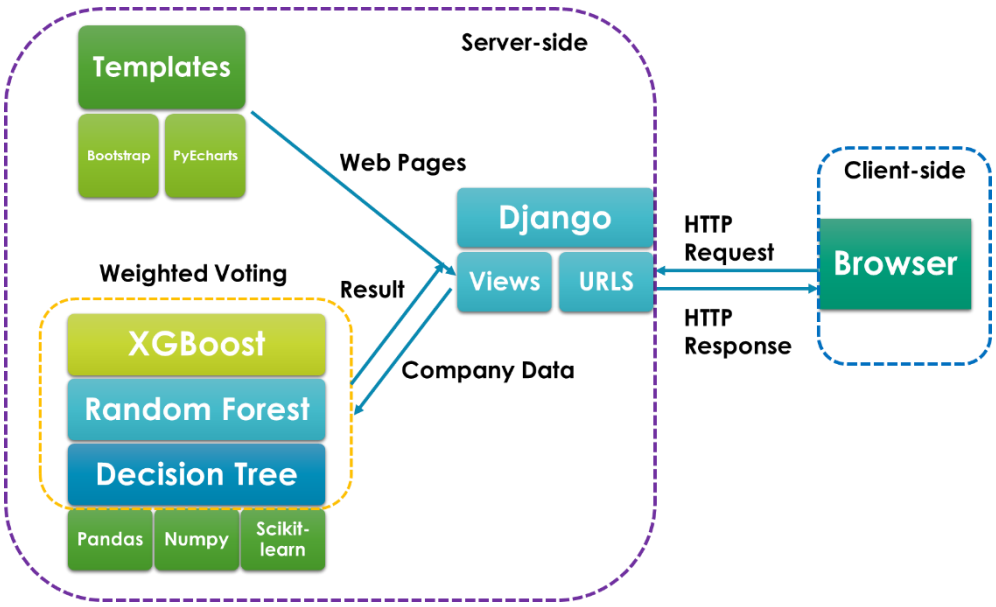

图 6：项目的整体结构和交互流程图

当用户在浏览器端访问我们的分类系统，浏览器向 Django 服务器发送一个 HTTP 请求，Django 服务器根据用户访问的 URL，选择合适的视图处理函数处理请求。视图函

数将用户的请求数据发送给训练好的加权投票融合算法模型，并从算法模型得到分类结果。在此基础上，利用分类结果填充预先编写好的页面模版，生成最终页面，以 HTTP 响应的方式返回给用户。

4. 开发工具与技术

- **主要开发工具**

➢ Jupyter Notebook：基于网页的交互式代码编辑器，本项目中主要用于数据处理和机器学习算法开发。

➢ PyCharm：功能强大的 Python IDE，本项目中主要用于 Web 可视化部分代码的编写与调试。

- **算法部分**

➢ Pandas：强大的分析结构化数据的 Python 第三方库，本项目中主要用于读取数据集，特征工程。

➢ Numpy：对高维数组有良好支持的 Python 第三方库，本项目中主要用于数据处理中的相关计算。

➢ Matplotlib：Python 中的绘图库，本项目中主要用于绘制各类图表。

➢ Scikit-learn：Python 下的机器学习库，本项目中主要用于算法实现。

➢ XGBoost：极端梯度提升算法库，本项目中作为多模型之一。

- **Web 可视化部分**

➢ Bootstrap：来自 Twitter 的前端框架，简洁灵活，本项目中主要用于搭建 Web 可视化网站的基本界面。

➢ PyEcharts：强大的数据可视化库，本项目中主要用于 Web 可视化分析部分图表的绘制。

➢ Django：基于 Python 的服务端网站框架，本项目中主要用于快速高效的开发 Web 可视化界面。

四、 分类算法

1. 数据预处理与特征工程

数据预处理与特征工程的总体流程图如下所示：

图 7：数据预处理与特征工程的总体流程图

接下来将对每个步骤进行简单的说明。

- **数据合并**

我们将企业提供的原始数据集中的训练集和验证集合并。对于验证集中企业基本信息表中的“控制人 ID”字段，由于在训练集和最终的测试集中并不出现，所以我们予以剔除。

- **会计恒等式处理**

我们利用会计恒等式：*资产总额 = 负债总额 + 所有者权益合计*，对数据进行了初步的筛查，发现并不存在不符合该式的数据。在此基础上，我们利用该式进行数据的初步填充。如果这三个字段中有任一字段缺失，我们就用其余两个字段计算出缺失的字段。

- **缺失填充**

对于数据集中的每个字段，我们计算其缺失比例，结果如下图所示：

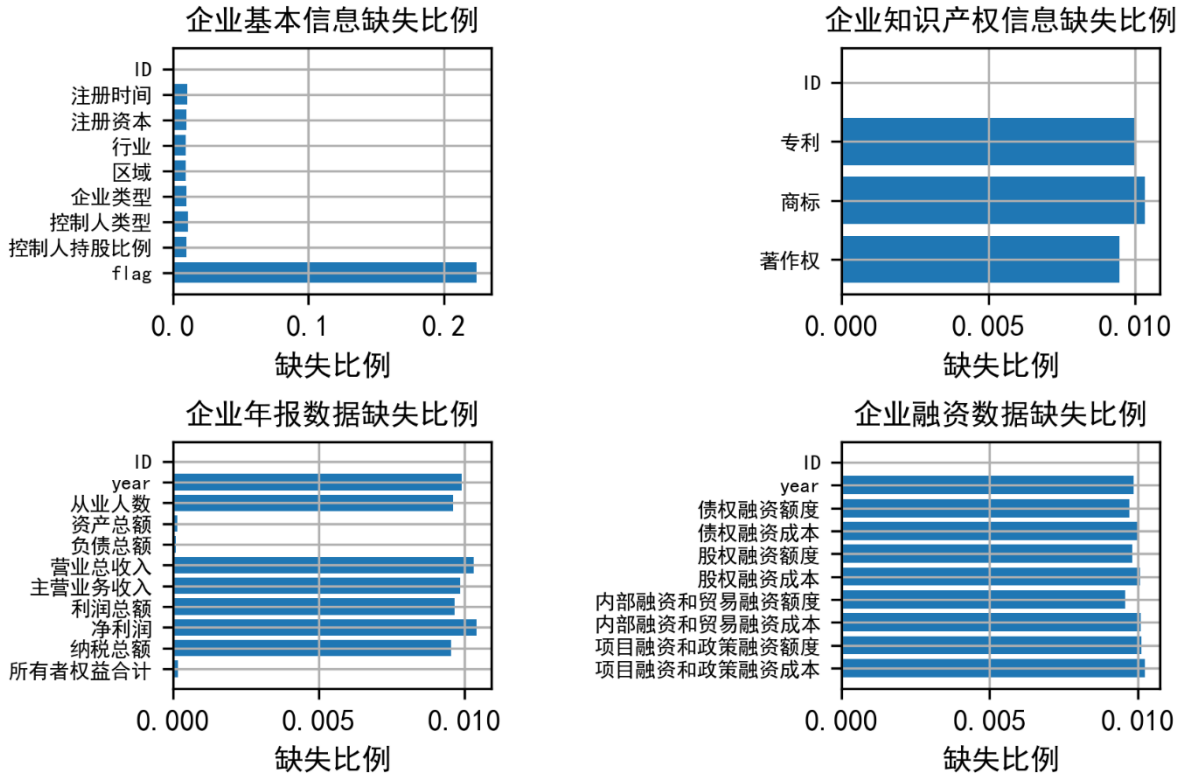

图 8：数据集字段缺失比例

可以发现，各字段的缺失比例都不高，因此我们考虑缺失填充。对于连续特征量，我们使用平均值进行填充，对于离散特征量，我们使用众数进行填充。特别地，对于 flag 缺失，我们使用-1 做标记。

- **业务层特征**

在业务层特征方面，我们首先比较了僵尸企业和非僵尸企业在各个字段上的差别，结果发现，僵尸企业和非僵尸企业在纳税总额和净利润这两个字段上有显著的差异。具体差异如下图所示：

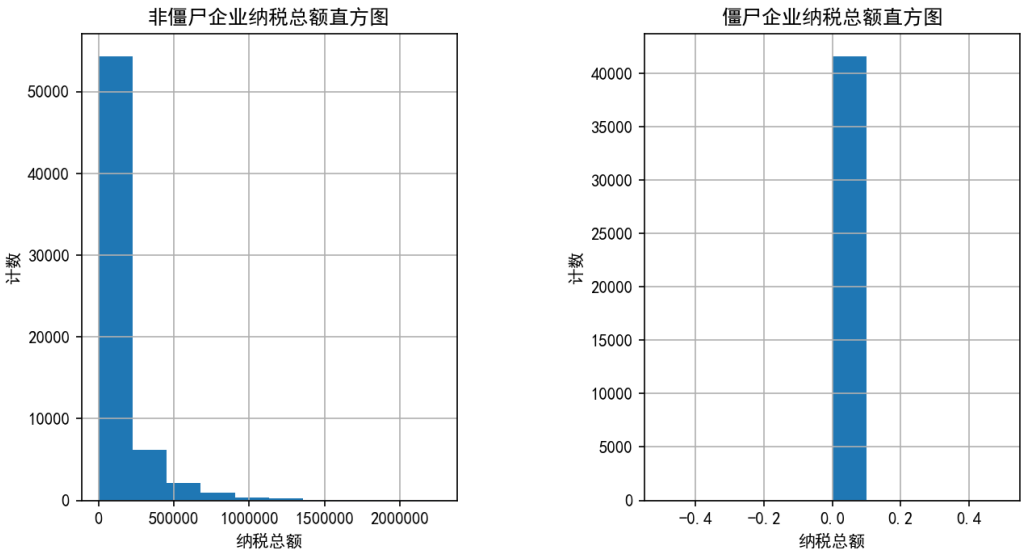

图 9：僵尸企业和非僵尸企业在纳税总额上的差别

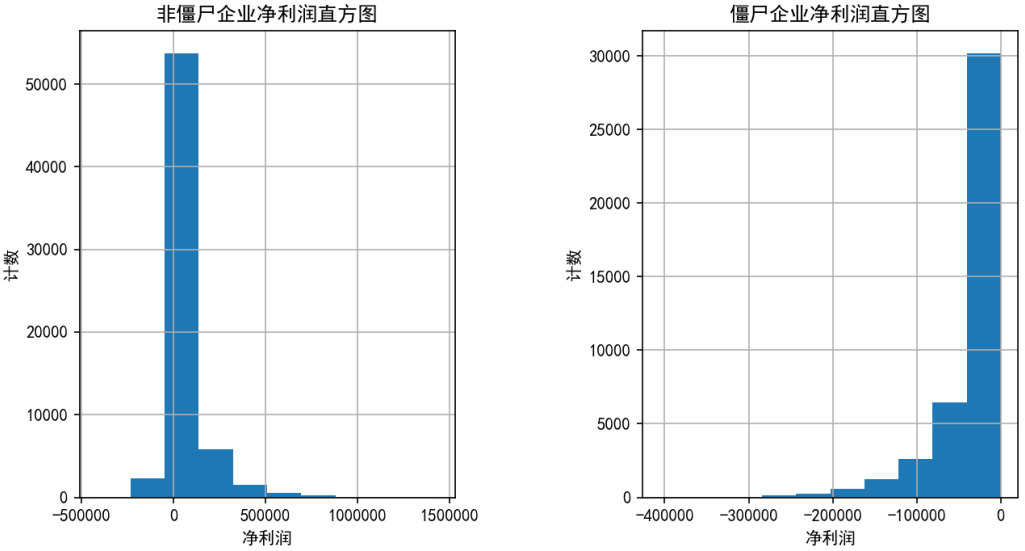

图 10：僵尸企业和非僵尸企业在净利润上的差别

从图中可以看到，僵尸企业的净利润基本都为零或者小于零，纳税总额均为零，而非僵尸企业的净利润则基本为零或者大于零，纳税总额基本大于等于零。

在此基础上，我们借鉴会计中对企业财务数据分析、企业运营状况分析的指标[4],[5]，最终构造了如下特征：

表 1：构造特征表

特征名	计算方法
纳税 / 净利	纳税总额 /(净利润 + 1)
资产负债率	负债总额 /(资产总额 + 1)
主收 / 营收	主营业务收入 /(营业总收入 + 1)

净利 / 资产	净利润 / (资产总额 + 1)
净利 / 营收	净利润 / (营业总收入 + 1)
纳税 / 营收	纳税总额 / (营业总收入 + 1)
额度 / 成本	(债权融资额度 + 股权融资额度 + 内部融资和贸易融资额度 + 项目融资和政策融资额度) / (债权融资成本 + 股权融资成本 + 内部融资和贸易融资成本 + 项目融资和政策融资成本 + 1)
债+股额度 / 内+项额度	(债权融资额度 + 股权融资额度) / (内部融资和贸易融资额度 + 项目融资和政策融资额度 + 1)
债+股成本 / 内+项成本	(债权融资成本 + 股权融资成本) / (内部融资和贸易融资成本 + 项目融资和政策融资成本 + 1)
融资额度	债权融资额度 + 股权融资额度 + 内部融资和贸易融资额度 + 项目融资和政策融资额度
融资成本	债权融资成本 + 股权融资成本 + 内部融资和贸易融资成本 + 项目融资和政策融资成本
利润 / 营收	利润总额 / (营业总收入 + 1)
净利 / 利润	净利润 / (利润总额 + 1)
所得税	利润总额 - 净利润
所得税 / 纳税	所得税 / (纳税总额 + 1)
净利 / 负债	净利润 / (负债总额 + 1)
纳税 / 负债	纳税总额 / (负债总额 + 1)
产权比例	负债总额 / (所有者权益合计 + 1)
费用	营业总收入 - 利润总额
费用 / 营收	费用 / (营业总收入 + 1)
利润 / 费用	利润总额 / (费用 + 1)

净利 / 融资额度	净利润 / (融资额度 + 1)
纳税 / 融资额度	纳税总额 / (融资额度 + 1)
总资产周转率	营业总收入 / (资产总额 + 1)
所有者权益构成率	所有者权益合计 / (资产总额 + 1)
净资产收益率	净利润 / (所有者权益合计 + 1)
综合资金成本	(债权融资成本*债权融资额度 + 股权融资成本*股权融资额度 + 内部融资和贸易融资成本*内部融资和贸易融资额度 + 项目融资和政策融资成本*项目融资和政策融资额度) / (融资额度 + 1)
政府补贴依赖程度	项目融资和政策融资额度 / (净利润 + 1)
融资/负债	融资额度 / (负债总额 + 1)
政策/负债	项目融资和政策融资额度 / (负债总额 + 1)
接受政府补助	政策 / 净利 < -0.5 || 政策 / 净利 > 1 && 负债/资产 > 0.5
净利润连续三年小于零	净利润 $_{i-2}$ < 0 && 净利润 $_{i-1}$ < 0 && 净利润 $_{i}$ < 0
经营风险	净利润平均差 / 净利润平均数
经营杠杆	利润变化 / 营业收入变化
连续三年纳税总额小于等于 0	纳税总额 $_{i-2}$ < 0 && 纳税总额 $_{i-1}$ < 0 && 纳税总额 $_{i}$ < 0
三年净利润和小于零	$\sum_{k=i-2}^{i} 净利润_k < 0$
连续三年净资产持续增加	所有者权益合计 $_{i-2}$ < 所有者权益合计 $_{i-1}$ < 0 < 所有者权益合计 $_{i}$

- **统计特征**

由于企业年报数据和企业融资数据包含了近三年的数据，因此我们考虑利用统计量对其进行汇总处理。我们使用了均值、最小值、最大值、方差、增长率这五个统计量。汇总处理之后，我们以企业 ID 为联结字段，对企业基本信息表、企业知识产权表、企

业年报数据表和企业融资数据表进行联结，得到最终用于训练的数据表。

- **One-hot encoding**

在训练之前，对于数据集中的离散特征，我们对其进行 One-hot encoding，具体的编码方案如下表所示。

表 2：行业 One-hot 编码结果

行业	One-hot 编码值
交通运输业	100000
商业服务业	010000
工业	001000
服务业	000100
社区服务	000010
零售业	000001

表 3：区域 One-hot 编码结果

区域	One-hot 编码值
山东	1000000
广东	0100000
广西	0010000
江西	0001000
湖北	0000100
湖南	0000010
福建	0000001

表 4：企业类型 One-hot 编码结果

企业类型	One-hot 编码值
农民专业合作社	10000
合伙企业	01000

有限责任公司	00100
股份有限公司	00010
集体所有制企业	00001

表 5：控制人类型 One-hot 编码结果

控制人类型	**One-hot 编码值**
企业法人	10
自然人	01

- **生成两份数据**

对于 flag 为-1 的数据，我们单独提取，作为未知企业数据，在数据融合阶段使用。而对于标签已经确定的企业数据，我们将其作为接下来的特征筛选、参数调优和模型融合的训练数据。

- **RFECV 特征筛选**

为了提高之后模型训练和预测的效率，我们使用 RFECV（递归特征消除）法对特征做了初步的筛选。

RFECV 的基本思想是采用交叉验证不断训练模型，每次训练完毕后记录评估分数，删除 k 个重要性最低的特征，然后用保留下来的特征再次进行训练。重复以上步骤，直到删除特征后评估分数降低，则结束特征筛选过程，得到最优特征组合。

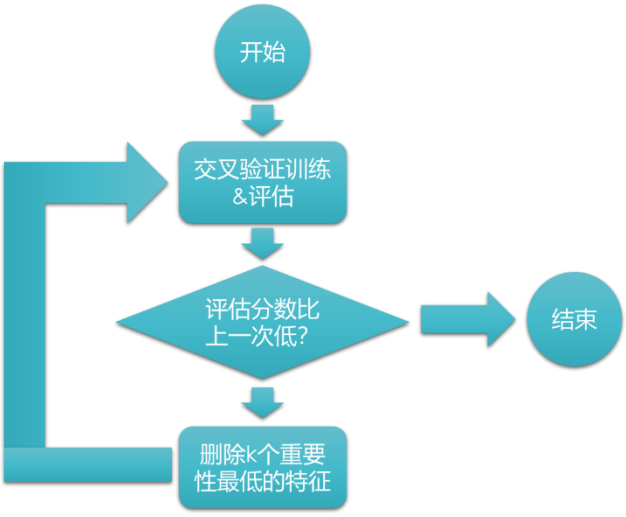

图 11：RFECV 基本流程

对于特征重要性的评价，采用属性 Gini 系数减少量作为依据，即计算采用该属性进行划分前后的 Gini 系数减少量。具体计算特征重要性的公式为：

$$FI(t) = \frac{N_t}{N} \times \left(Gini_{old} - \frac{N_{tR}}{N_t} \times Gini(tR) - \frac{N_{tL}}{N_t} \times Gini(tL) \right)$$

其中，$FI(t)$ 为属性 t 的特征重要性，$Gini_{old}$ 为采用该属性进行划分前的 Gini 系数，N_t 为该节点下的样本数，N 为样本总数，N_{tR} 为右子树下的样本数，N_{tL} 为左子树下的样本数，$Gini(tR)$ 为右子树的 Gini 系数，$Gini(tL)$ 为左子树的 Gini 系数。

筛选前，我们去除 ID 字段，一共有 250 个特征参与筛选。

在筛选时，我们分别选择随机森林和决策树作为筛选时的基模型，使用 F1-Score 作为评价指标，采用五折交叉验证，k为 1。筛选后共有 131 个特征被保留。

2. 算法比较选择

我们选取了六种常见的分类算法，在处理之后的训练集上进行初步的测试，采用五折交叉验证，各模型的 F1-Score、训练时间和预测时间如下图所示：

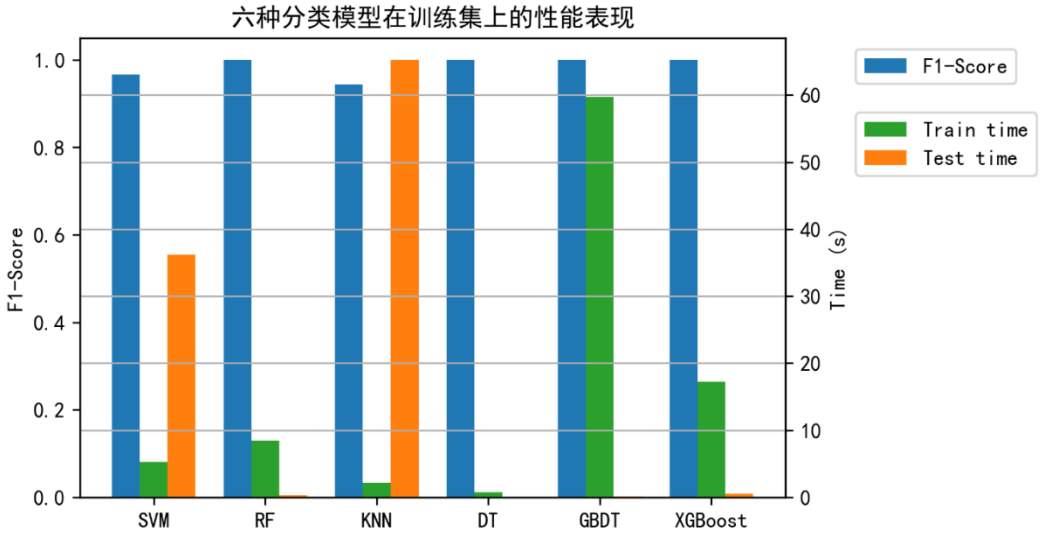

图 12：六种分类模型在训练集上的性能表现

通过上图可知，决策树、随机森林与 XGBoost 在 F1-Score、训练时间和预测时间上表现良好，因此我们选择决策树、随机森林与 XGBoost 作为模型融合的基模型。

3. 算法原理概述

- **决策树**

决策树（Decision Tree）是一类常见的机器学习分类算法，思想十分朴素，类似于我们平时利用选择做决策的过程。

决策树在逻辑上以树的形式存在。一般的，一棵决策树包含一个根结点、若干个内部结点和若干个叶子结点；叶子结点对应决策结果，其他每个结点则对应一个属性测试。每个结点包含的样本集合根据属性测试的结果被划分到子结点中，根结点包含样本全集。从根结点到每个叶结点的路径对应一个判定测试序列。

决策树学习的关键步骤在于从候选属性中选取最优的属性 a，其属性选择的质量决定了决策树的预测准确度。一般而言，随着划分过程的不断进行，我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”越来越高。而对于节点的纯度，我们可以使用 Gini 系数来度量。定义节点 B 的 Gini 系数为：

$$Gini(B) = 1 - \sum_{i=0}^{k} p_i^2$$

其中，p_i 表示节点 B 中属于第 i 类的概率，k 为数据集中的类别总数。显然，Gini 系数越小，节点的纯度越高。

对于每个候选属性 c 来说，选择该属性进行划分后分支节点的 Gini 系数加权平均和就是该属性的 Gini 系数：

$$Gini(c) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v)$$

其中，D 是所有分支节点组成的集合，D^v 是第 v 个分支节点，$|D|$ 表示所有分支节点中的样本总数，$|D^v|$ 表示第 v 个分支节点中的样本总数。

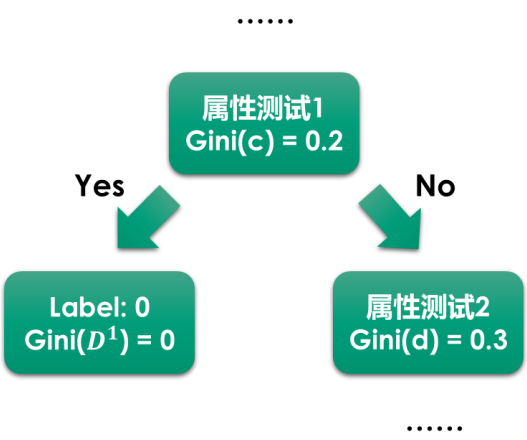

图 13：决策树的部部分示意图

每次从候选属性中选取最优的属性 a 时，我们总是选择候选属性中 Gini 系数最小的那个作为当前对数据集进行划分的属性。重复这个过程，就可以构建出一棵决策树。而在预测时，我们对于输入数据，从根节点出发，根据属性测试选择下一条路径，直到到达叶子节点，即完成了预测。

- **随机森林**

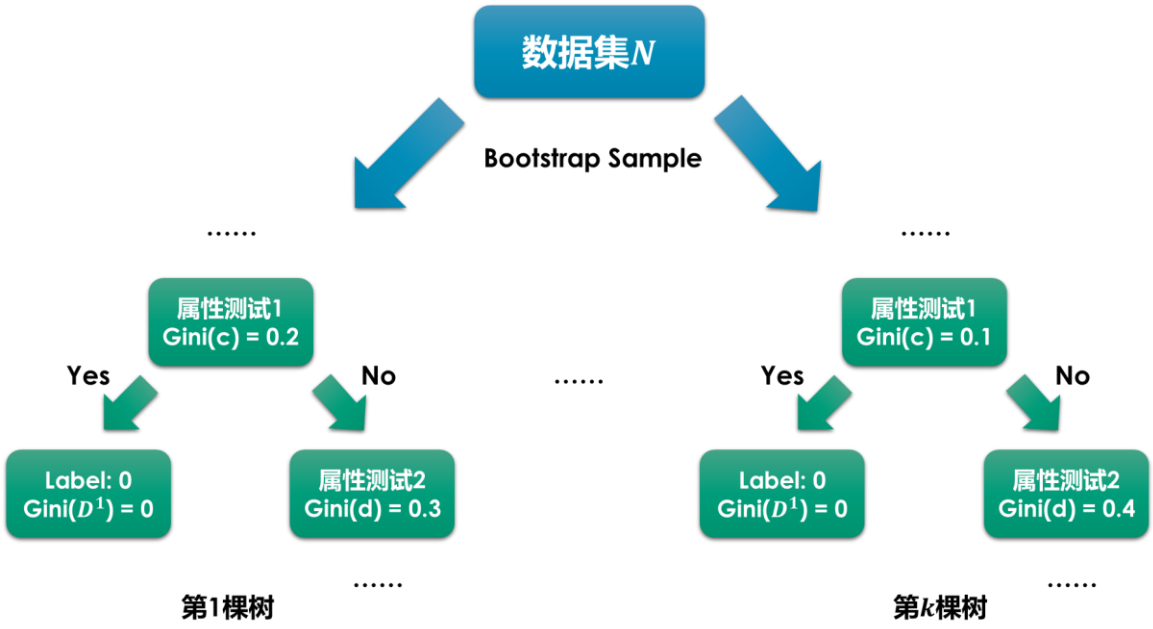

图 14：随机森林部分原理示意图

随机森林就是通过集成学习的思想将多棵决策树集成的一种算法。所谓森林，即表明有多棵决策树，最终的输出结果是所有决策树输出结果的多数派。

而对于随机，其定义了每棵决策树的生成方式。如果训练集大小为N，对于每棵树而言，随机且有放回地从训练集中的抽取 N 个训练样本（bootstrap sample），作为该树的训练集。由此可以知道，每棵树的训练集都是不同的，且里面包含重复的训练样本。

- **XGBoost**

XGBoost 模型中使用的是决策树集成，即由一系列决策树组成。其基本想法是单一决策树的预测能力是有限的，使用多棵决策树能够提升预测性能。其最终的预测结果就是 k 棵树的结果求和：

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

其中，K 是树的棵数，f_k 表示第 k 棵决策树，$f_k(x_i)$ 就是第 k 棵决策树对第 i 个样本 x_i 的预测结果，$\mathcal{F}$ 是树的集合。

在训练时，XGBoost 采用了一种残差思想，第 t 轮的预测结果，是在 $t-1$ 轮的预测结果基础上，添加一棵新树产生的，即新树拟合的是前一轮预测值与真实值间的误差。

$$\hat{y}_i^0 = 0$$
$$\hat{y}_i^1 = f_1(x_i) = \hat{y}_i^0 + f_1(x_i)$$
$$\hat{y}_i^2 = f_1(x_i) + f_2(x_i) = \hat{y}_i^1 + f_2(x_i)$$
$$\cdots\cdots$$
$$\hat{y}_i^t = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{t-1} + f_t(x_i)$$

上一轮预测结果　　残差修正

图 15：XGBoost 原理简单示意

4. **参数调优**

我们使用网格搜索五折交叉验证进行参数调优。

对于决策树，我们主要调整的参数以及每个参数的搜索范围如下表所示：

表 6：决策树参数搜索范围

参数名称	搜索范围
class_weight	None, balanced
max_depth	[3, 9]
min_samples_split	[2, 9]

min_samples_leaf	[2, 9]

部分参数搜索过程如下图所示：

图 16：决策树部分参数调优过程

最终的参数取值如下表所示：

表 7：决策树参数搜索结果

参数名称	**最终取值**
class_weight	None, balanced
max_depth	3
min_samples_split	2
min_samples_leaf	4

对于随机森林，我们主要调整的参数以及每个参数的搜索范围如下表所示：

表 8：随机森林参数搜索范围

参数名称	**搜索范围**
class_weight	None, balanced, balanced_subsample
n_estimators	[10, 200]

部分参数搜索过程如下图所示：

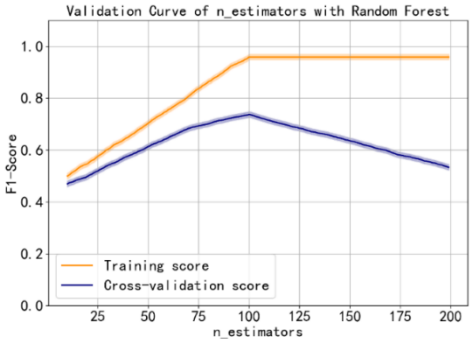

图 17：随机森林部分参数调优过程

最终的参数取值如下表所示：

表 9：随机森林参数搜索结果

参数名称	最终取值
class_weight	balanced
n_estimators	100

对于 XGBoost，我们主要调整的参数以及每个参数的搜索范围如下表所示：

表 10：XGBoost 参数搜索范围

参数名称	搜索范围
max_depth	[3, 10]
gamma	[1, 10]
reg_alpha	[1, 10]
reg_lambda	[1, 10]

参数搜索过程如下图所示：

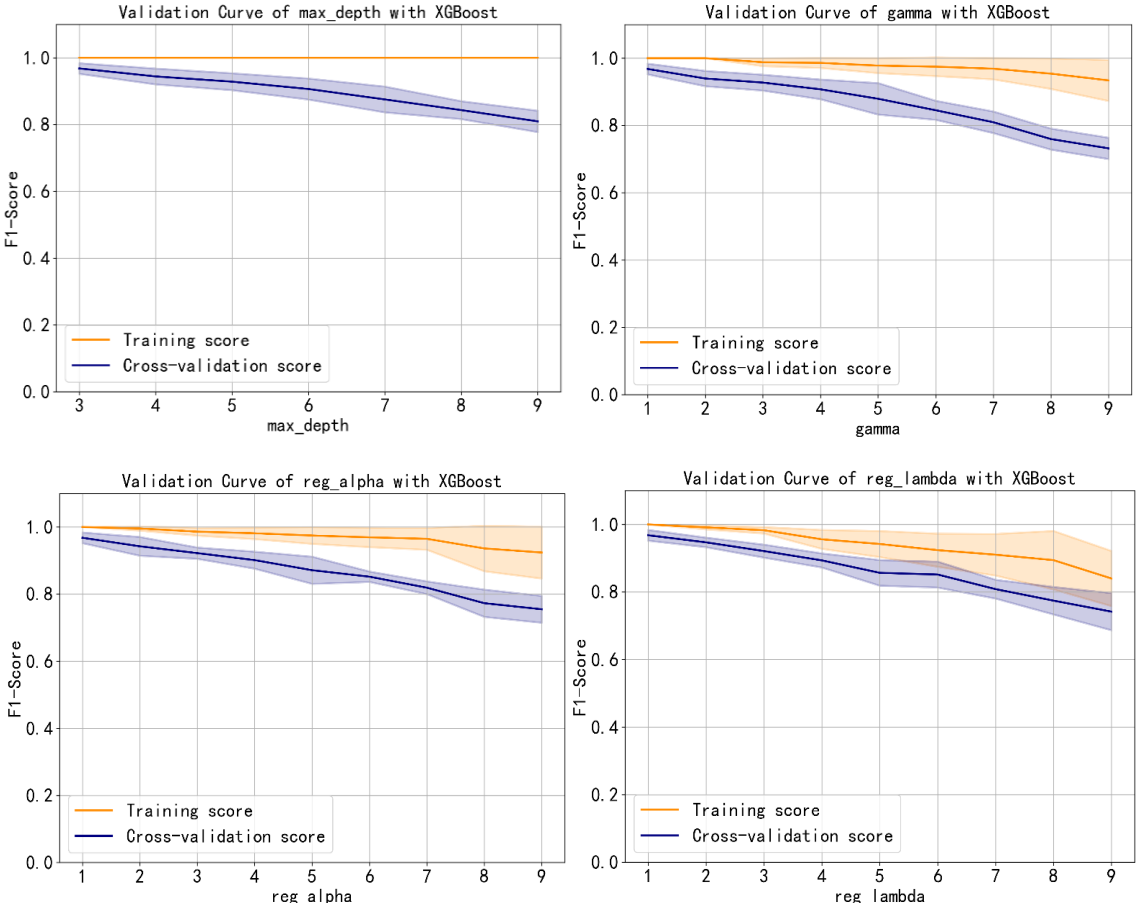

图 18：XGBoost 参数调优过程

最终的参数取值如下表所示：

表 11：XGBoost 参数搜索结果

参数名称	最终取值
max_depth	3
gamma	1
reg_alpha	1
reg_lambda	1

5. 模型融合

- **加权投票融合机制**

为了进一步提升模型的预测表现，我们将经过参数调优后的决策树、随机森林与 XGBoost 进行模型融合，采用的是加权投票融合机制。

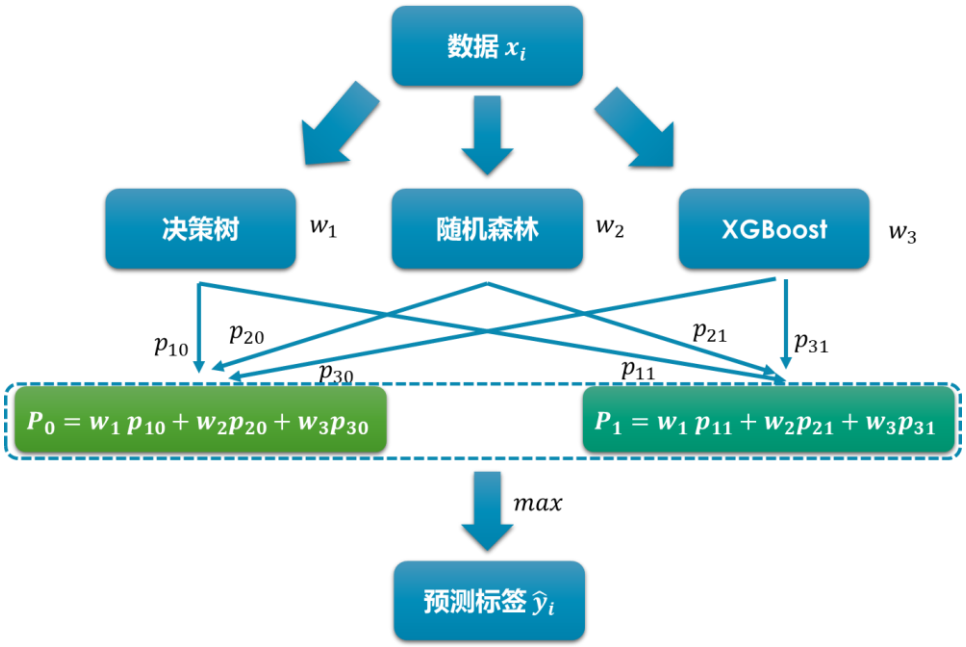

图 19：加权投票融合机制示意图

加权投票融合机制即将模型对各类别的预测概率进行加权平均。记模型 i 对第 j 个类别的预测概率为 p_{ij}，模型 i 的权重为 w_i，则最终所有模型对每个类别的加权预测概率为：

$$P_j = \sum_{i=1}^{3} w_i p_{ij}$$

在本项目中，只有僵尸企业和非僵尸企业两种类别，所以 $j = 0, 1$。最终的预测标签输出即取所有类别的加权预测概率的最大值即可。

- **遗传算法确定权重**

加权投票融合机制的关键是要确定各模型的权重。由于权重的搜索空间过大，依靠遍历搜索并不可取。因此，我们采取随机优化方法——遗传算法来确定权重。

遗传算法的基本流程如下图所示：

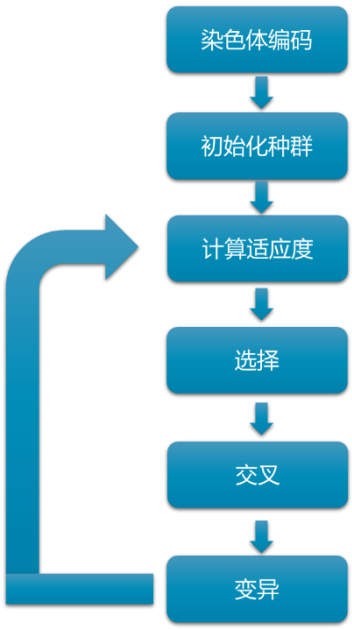

图 20：遗传算法基本流程图

➢ **染色体编码**

我们采用实数直接编码，权重精度取10^{-3}，因此每个权重占三位，染色体总长为九位。

$$w_1 = 0.267$$
$$w_2 = 0.947$$
$$w_3 = 0.708$$

图 21：染色体编码示意图

➢ **适应度函数**

适应度函数我们选取 F1-Score，其计算公式为：

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

其中，Precision 和 Recall 分别为精确率和召回率。

➢ **选择**

对于个体的选择，我们采用轮盘赌算法。为此，需要计算每个个体的归一化适应度函数值：

$$F1 - Score_i = \frac{F1 - Score_i}{\sum_{k=1}^{K} F1 - Score_k}$$

其中，$F1-Score_i$ 为第 i 个个体的适应度函数值，K 为种群大小。

- **交叉**

对于染色体的交叉，我们产生一个[0, 8]之间的随机数 rand，表示需要进行交叉的起始位置，然后以此位置为起始进行交叉。

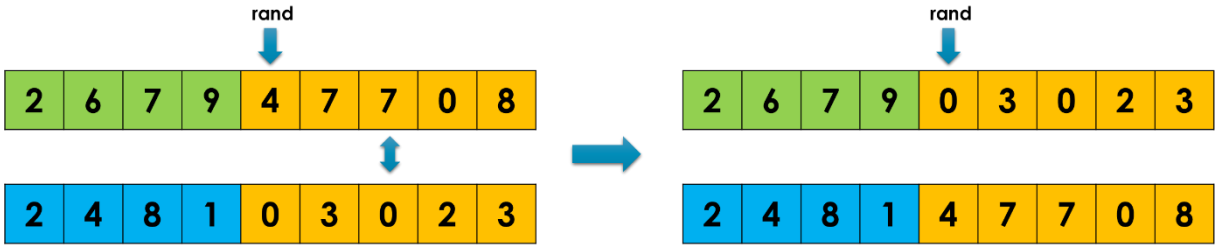

图 22：染色体交叉示意图

- **变异**

对于染色体的变异，我们产生一个[0, 8]之间的随机数 rand1，表示需要进行变异的起始位置，然后再产生一个[0, 9]之间的随机数 rand2，表示该位置变异后的值。

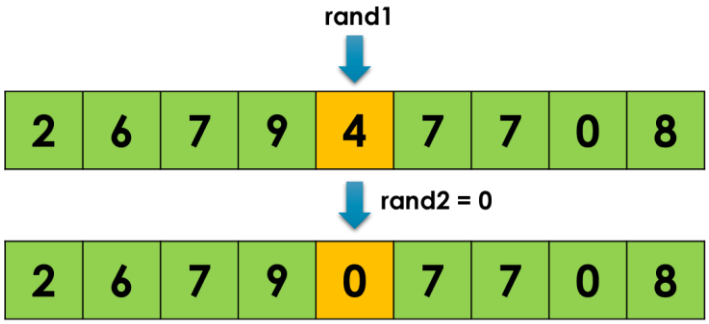

图 23：染色体变异示意图

在权重搜索时，我们取种群大小 200，迭代次数 100，交叉概率 0.8，变异概率 0.1. 最终得到的最佳模型权重如下表所示：

表 12：最佳模型权重

w_1	w_2	w_3
0.416	0.198	0.386

6. 数据融合

由于原始数据集中存在大量的未知标签数据，我们采用数据融合的方式利用这部分数据。数据融合的基本过程如下图所示：

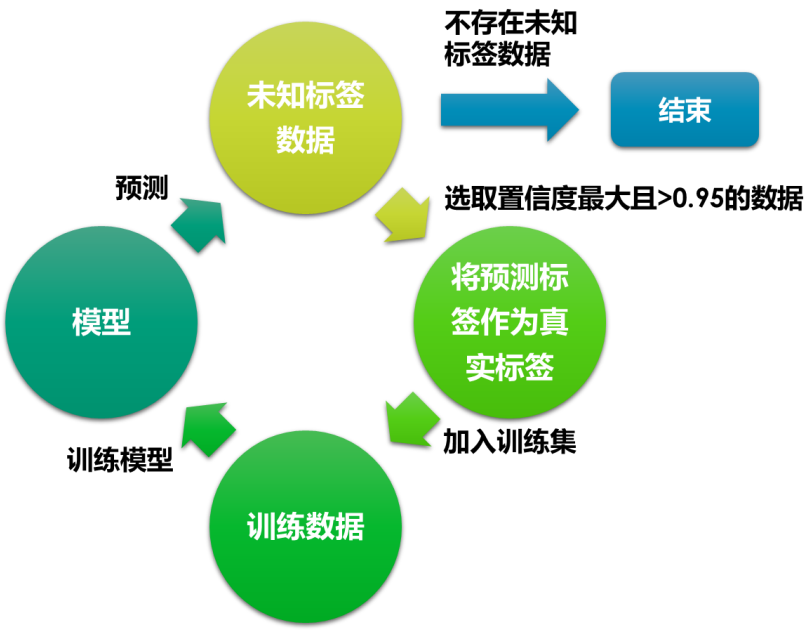

图 24：数据融合流程

我们利用在已知标签上训练的模型，预测未知标签的数据，每次选取置信度最大且大于 0.95 的数据，将预测标签作为真实标签加入训练数据，重新进行训练。反复迭代，直到所有数据都加入了训练数据集。迭代过程中剩余的未知标签数据数量变化如下图所示：

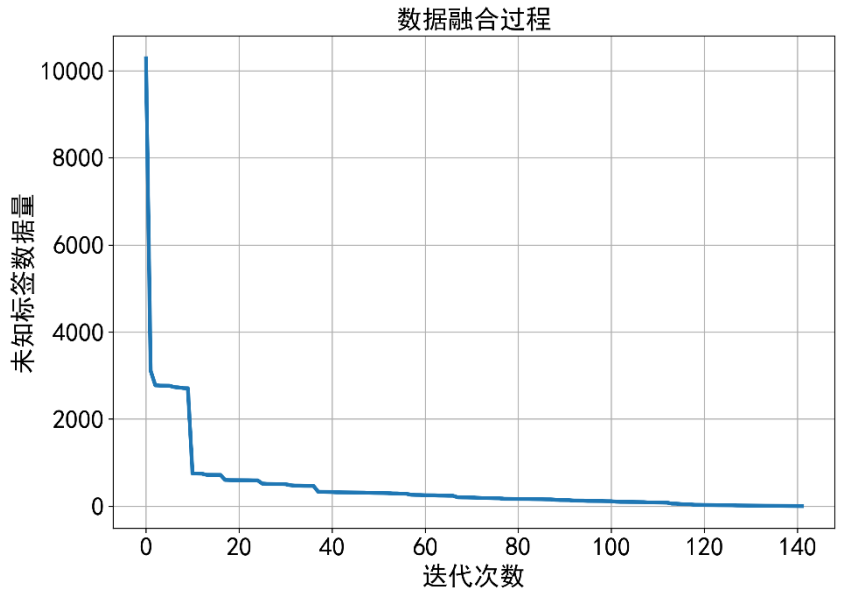

图 25：数据融合过程

可以看到，随着不断地迭代，最终所有未知标签数据数据都以较高地置信度加入了训练数据集中。

五、 Web 可视化界面

1. 页面设计

根据企业要求，我们需要开发 Web 可视化界面，支持单个和批量企业分类，数据输入使用 csv 文件导入，输出结果同样为一个 csv 文件，包含 ID 和 flag 两列。在此基础上，我们还增加了单个企业输入时可以手动输入所有特征的功能，以及单个企业信息可

视化分析的功能。

最终的页面构成与交互流程如下图所示：

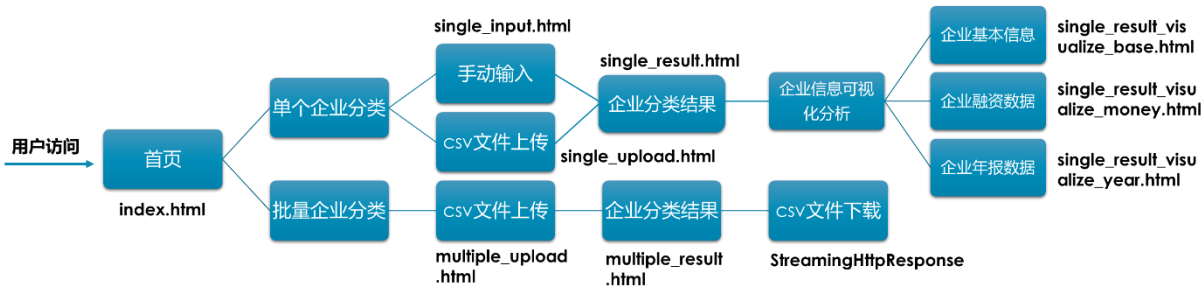

图 26：页面构成与用户交互流程

2. 页面开发

- **首页**

首页主要用于引导用户进行单个企业分类和批量企业分类。页面如下所示：

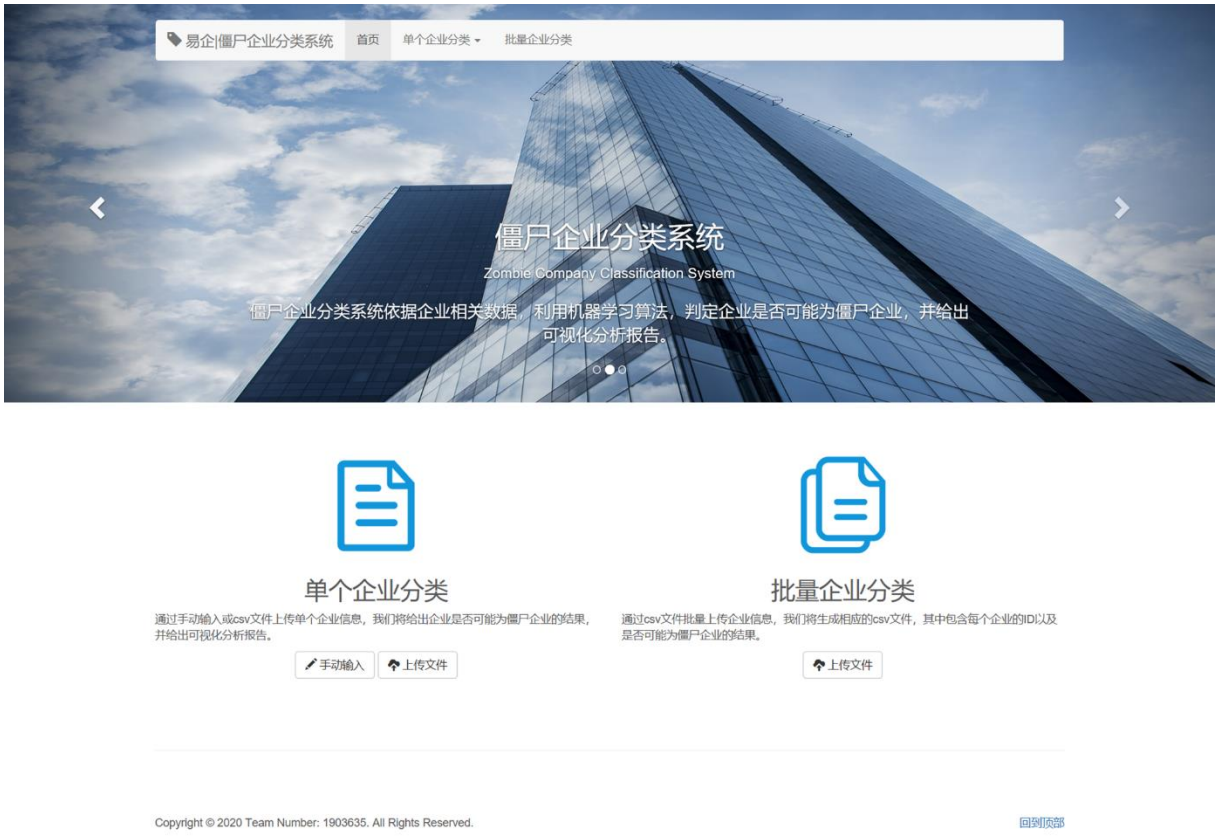

图 27：首页

- **单个企业分类-手动输入**

该页面用于用户想要手动输入单个企业的所有特征。由于页面较长，这里只展示部分。部分页面如下所示：

图 28：单个企业分类-手动输入（部分）

- **单个企业分类-csv 文件上传**

该页面用于用户上传单个企业信息 csv 文件。页面如下所示：

图 29：单个企业分类-csv 文件上传

- **单个企业分类-企业分类结果**

该页面展示用户上传的企业信息的分类结果，是僵尸企业还是非僵尸企业。页面如下所示：

图 30：单个企业分类-企业分类结果

单击可视化分析按钮，则可以跳转到可视化分析页面。

- **单个企业分类-可视化分析-企业基本信息**

该页面可视化用户上传的企业信息中的基本信息。部分页面如下所示：

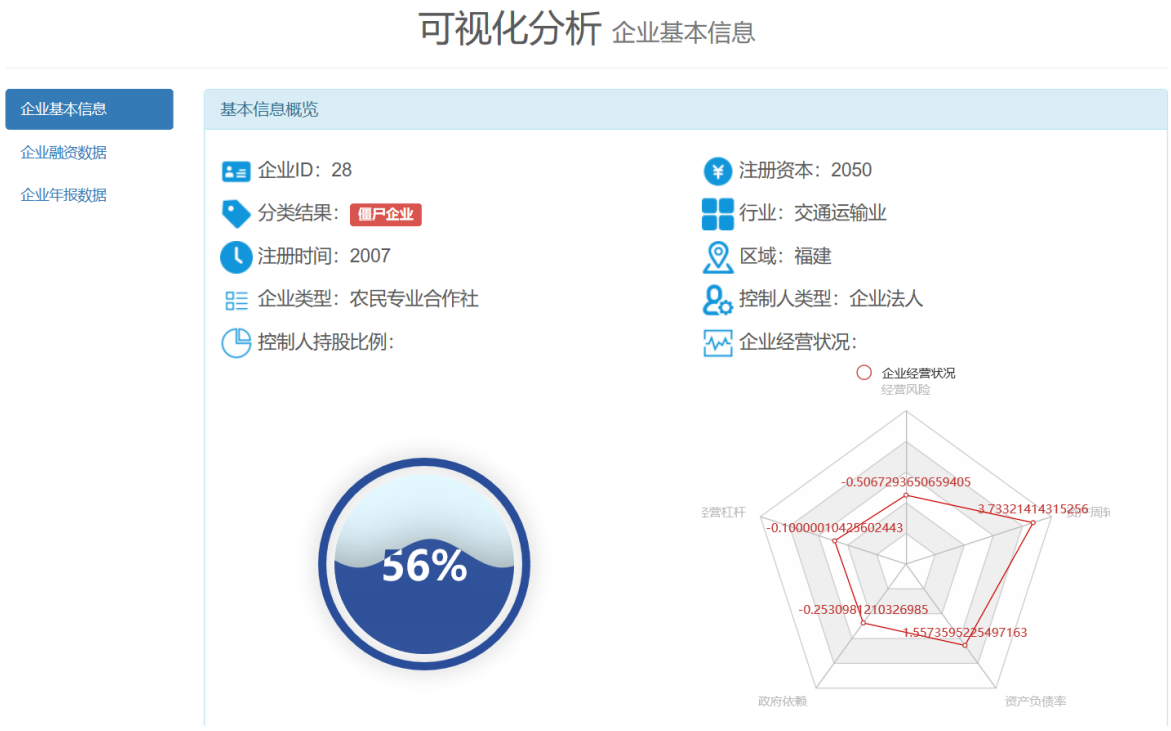

图 31：单个企业分类-可视化分析-企业基本信息（部分）

- **单个企业分类-可视化分析-企业融资数据**

该页面可视化用户上传的企业信息中的融资数据，帮助用户分析为何该企业是僵尸企业或非僵尸企业。部分页面如下所示：

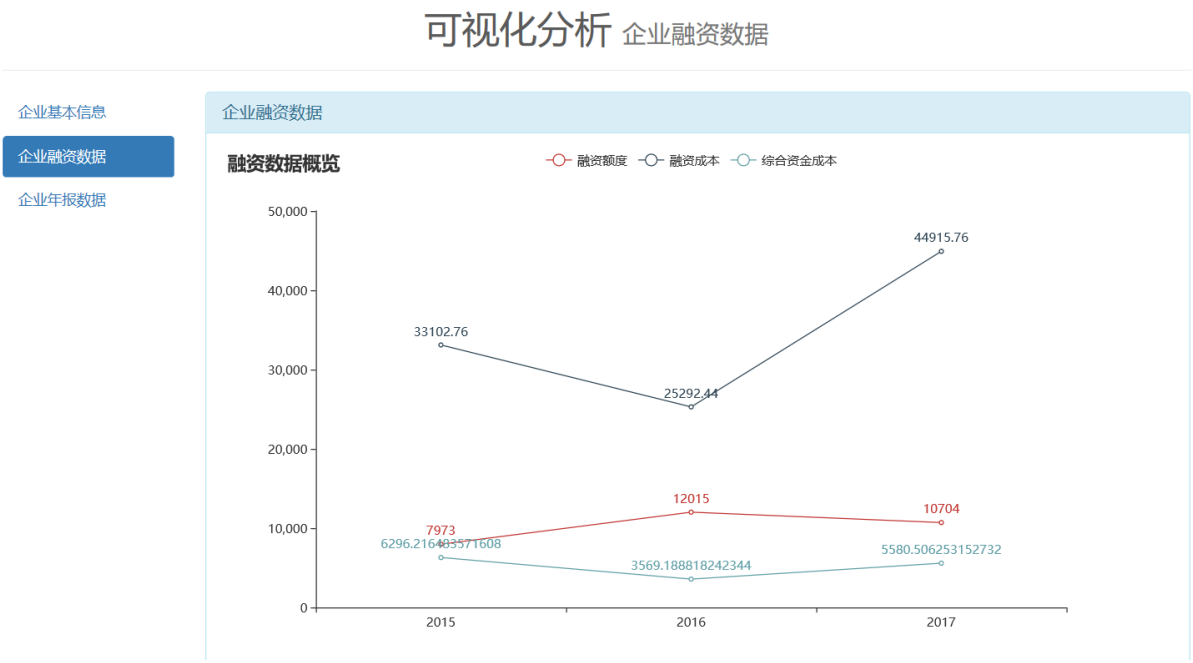

图 32：单个企业分类-可视化分析-企业融资数据（部分）

- **单个企业分类-可视化分析-企业年报数据**

该页面可视化用户上传的企业信息中的年报数据，帮助用户分析为何该企业是僵尸企业或非僵尸企业。页面如下所示：

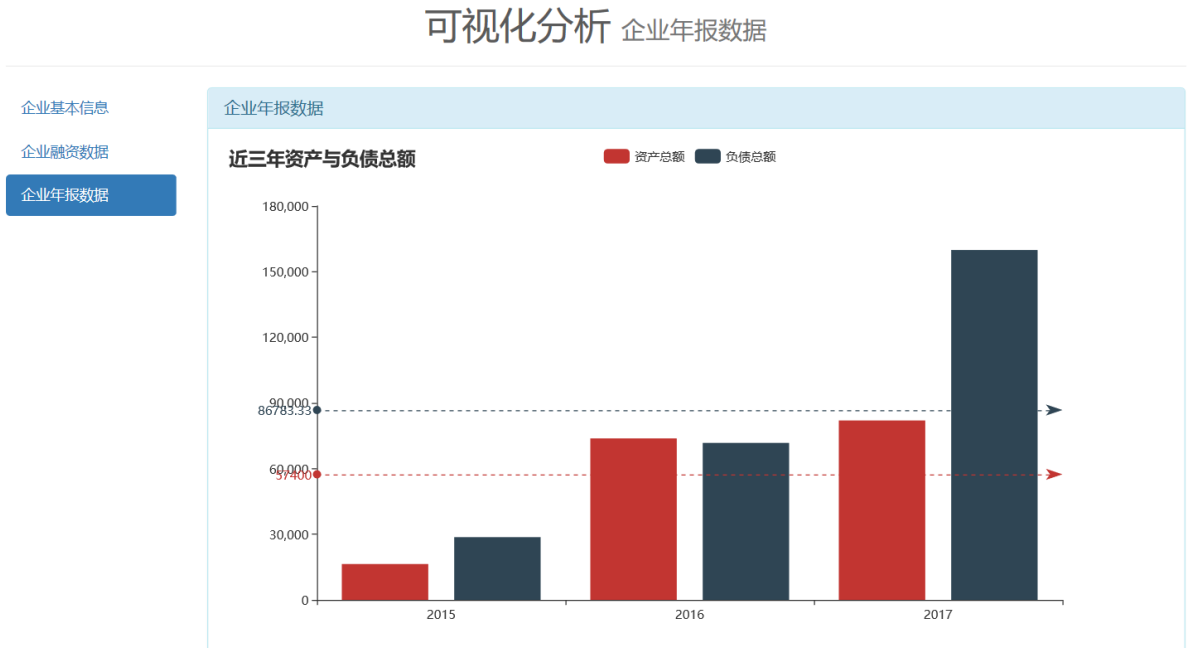

图 33：单个企业分类-可视化分析-企业年报数据（部分）

- **批量企业分类-csv 文件上传**

该页面用于用户上传批量企业信息 csv 文件。页面如下所示：

图 34：批量企业分类-csv 文件上传

- **批量企业分类-企业分类结果**

该页面提供用户上传的批量企业信息的分类结果下载。页面如下所示：

图 35：批量企业分类-企业分类结果

单击下载结果文件按钮即可得到最终的包含企业 ID 和 flag 标签的结果 csv 文件。

3. URL 配置

为了使用户能够访问到每个页面，我们需要为每个页面配置 URL。具体配置的 URL 如下表所示，此处已省略域名和公共前缀 classifier.

表 13：URL 配置表

URL	对应页面
/	首页
/single/input/	单个企业分类-手动输入
/single/upload/	单个企业分类-csv 文件上传
/single/result/	单个企业分类-企业分类结果
/single/result/visualize/base/	单个企业分类-可视化分析-企业基本信息
/single/result/visualize/money/	单个企业分类-可视化分析-企业融资数据
/single/result/visualize/year/	单个企业分类-可视化分析-企业年报数据
/multiple/upload/	批量企业分类- csv 文件上传
/multiple/result/	批量企业分类-企业分类结果
/multiple/download/	下载结果文件（StreamingHttpResponse）

4. **视图函数编写**

对于单个或批量企业分类时的文件上传或手动输入，我们首先需要编写相应的表单类。文件上传使用 UploadForm 表单类，包含四个文件字段，分别对应企业基本信息、企业知识产权数据、企业融资数据和企业年报数据。手动输入使用 SingleDataInputForm 表单类，包含所有特征字段。在视图函数中，实例化表单类，然后需要处理表单请求，其具体流程如下图所示：

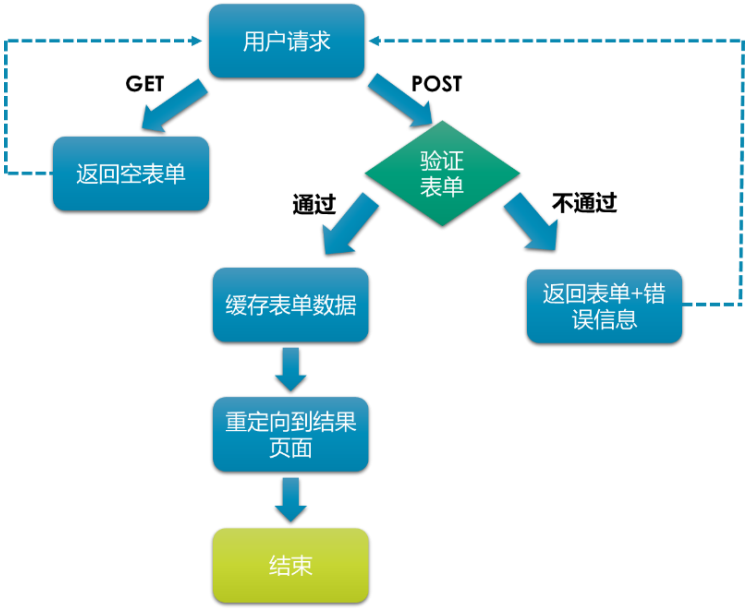

图 36：表单处理流程

对于首页，只需要直接返回编写好的首页页面即可。

对于企业分类结果，根据表单请求处理后得到的文件，调用预训练好的模型 CompanyClassifier 进行分类，将分类结果返回给用户。如果是批量企业分类，则使用 StreamingHttpResponse 方法返回流式 csv 结果文件下载。

对于可视化分析，根据表单请求处理后得到的文件，使用 PyEcharts 绘图并生成图表嵌入代码，插入到网页中返回给用户。

六、 功能测试

1. 分类算法

- **数据预处理以及分类模型效率**

我们测试了我们的数据预处理与特征工程方法以及分类模型在不同数据量规模情况下完成工作所需要的时间，得到的结果如下图所示：

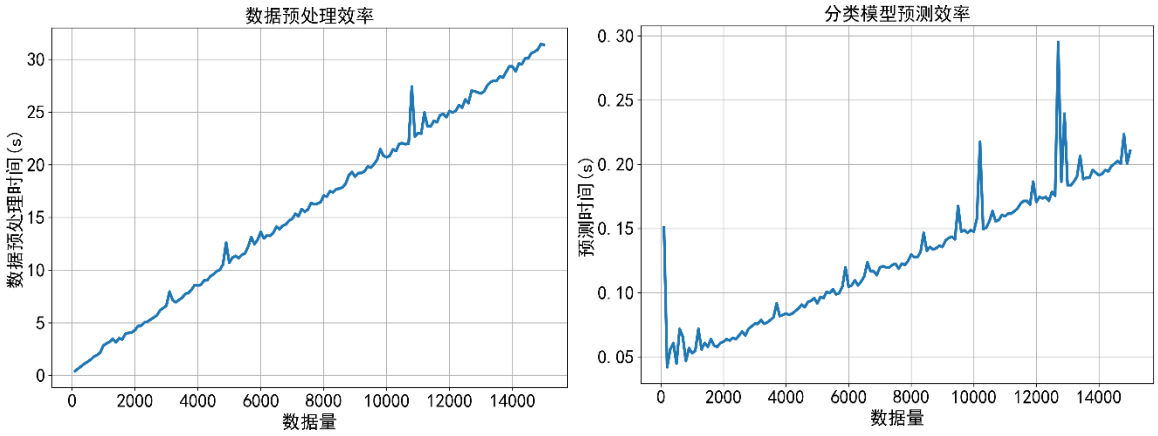

图 37：数据预处理以及分类模型效率

从图中可以看出，我们的数据预处理与特征工程方法完成工作所需要的时间随着数据量规模的增加线性增长，但总体上所需时间并不多。而模型预测所需要的时间虽然也随着数据量规模的增加线性增长，但增长幅度较小。总体上，我们的数据预处理与特征工程方法以及分类模型的效率与性能表现良好。

- **分类模型学习曲线**

我们画出分类模型的学习曲线，如下图所示：

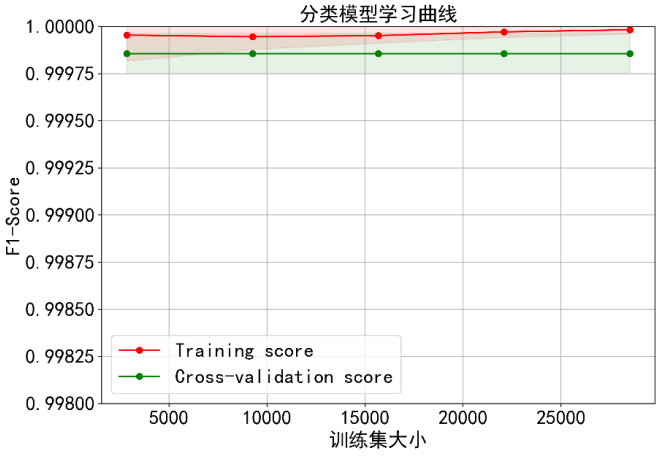

图 38：分类模型学习曲线

从图中可以看到，经过大量样本的训练，模型基本拟合了数据集，且没有发生过拟合和欠拟合。

模型的可扩展性如下图所示：

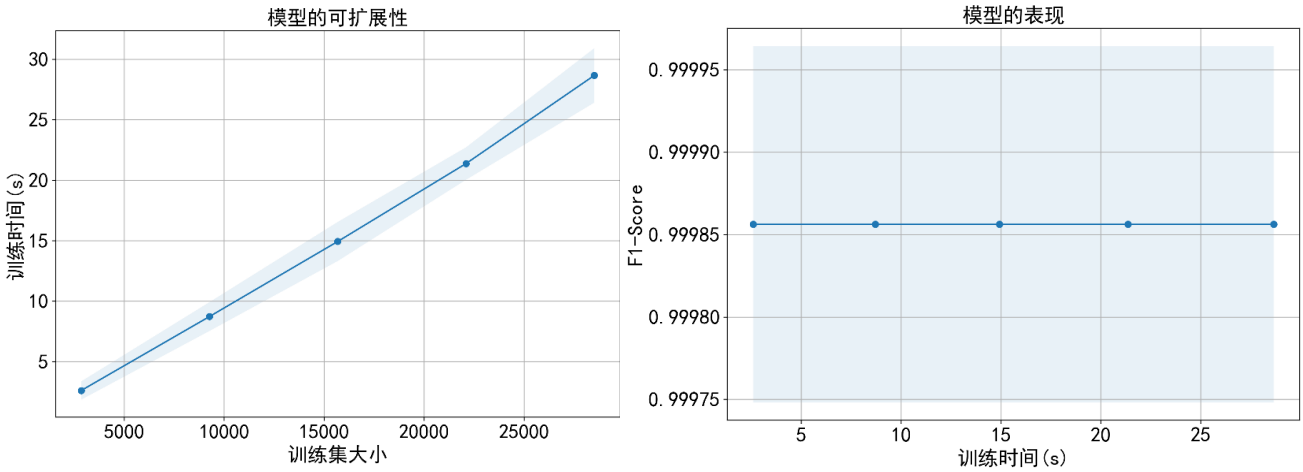

图 39：模型的可扩展性

从图中可以看出，模型在不同规模的数据集上的训练时间线性增长，但总体在可接受的范围内。在不同规模的数据集上模型的性能表现优异。总体来说，模型具有比较好的可扩展性。

2. Web 可视化端

- **测试内容**

对于 Web 可视化端的测试，我们主要测试了视图函数是否能够正确处理用户的请求。每个视图函数具体的测试内容如下图所示：

表 14：测试内容

视图函数	测试内容
index	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
multiple_upload	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
	测试文件能否被正确上传以及上传后的返回状态码是否为 302
multiple_result	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
	测试是否能得到分类结果

multiple_download	测试能否下载结果文件以及返回状态码是否为 200
single_upload	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
	测试文件能否被正确上以及上传后的返回状态码是否为 302
single_input	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
	测试表单提交后能否正确生成文件以及返回状态码是否为 302
single_result	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
	测试是否能得到分类结果
single_result_visualize_base	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
single_result_visualize_money	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版
single_result_visualize_year	测试用户访问的返回状态码是否为 200
	测试是否使用了正确的网页模版

- **测试结果**

一共有 24 个测试样例，测试结果显示 24 个测试样例均全部通过。

测试代码覆盖率如下表所示：

表 15：测试代码覆盖率

代码文件	代码总数	未测试数	测试覆盖率
classifier__init__.py	0	0	100%

classifier\admin.py	1	0	100%
classifier\algorithm.py	164	0	100%
classifier\apps.py	3	0	100%
classifier\forms.py	73	0	100%
classifier\migrations__init__.py	0	0	100%
classifier\models.py	1	0	100%
classifier\tests.py	125	0	100%
classifier\urls.py	3	0	100%
classifier\views.py	262	0	100%
manage.py	14	0	100%
webclassifier__init__.py	0	0	100%
webclassifier\settings.py	18	0	100%
webclassifier\urls.py	10	0	100%
总计	674	0	100%

从表中可以看到，测试用例覆盖了 Web 端的所有代码，具有良好的测试效果。

七、 项目总结

1. 僵尸企业的主要特征

我们在训练完毕分类模型后绘制了特征重要性图，以各特征在所有决策树中被用来作为属性测试节点的次数来衡量特征重要性，得到的结果如下所示：

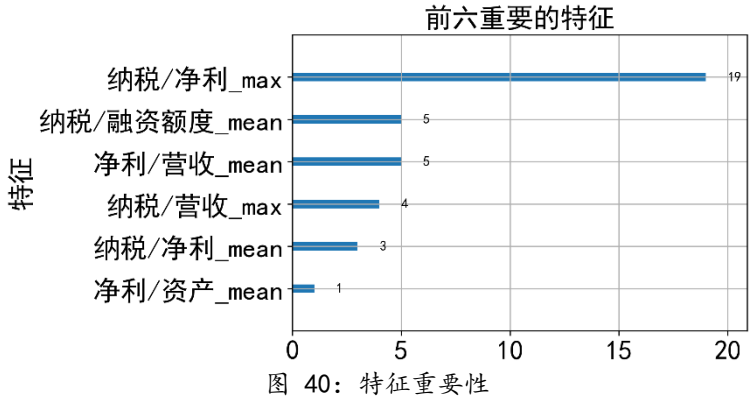

图 40：特征重要性

从图中可以看到，对于分类模型最重要的特征是纳税总额和净利润。结合特征工程阶段对数据集中僵尸企业和非僵尸企业的对比，我们认为僵尸企业最主要的特征就是纳税总额基本为 0，净利润基本为 0 或为负，表明长期处于亏损状态因而无法纳税。这也符合僵尸企业盈利性差、吸血性强的特征。

参考资料

[1] 周琎, 冼国明, 明秀南. 僵尸企业的识别与预警—来自中国上市公司的证据[J]. 财经研究, 2018, 44(4): 130-142.

[2] 栾甫贵, 刘梅. 僵尸企业僵尸指数的构建及应用研究[J]. 经济与管理研究, 2018 (6): 12.

[3] 邹蕴涵. 我国僵尸企业的判别, 影响及对策建议[J]. 中国物价, 2016 (7): 80-82.

[4] 李霄阳, 瞿强. 中国僵尸企业: 识别与分类[J]. 国际金融研究, 2017, 364(8): 3-13.

[5] 黄少卿, 陈彦. 中国僵尸企业的分布特征与分类处置[J]. 中国工业经济, 2017, 3: 24-43.