基于机器学习的发债主体违约风险预测

目　　录

一、 问题定义.....1
二、 问题背景.....1
三、 主要方法.....1
（一） LassoCV.....2
（二） LightGBM.....2
四、 数据收集.....5
（一） 数据获取.....5
（二） 数据概述.....6
（三） 重要数据描述性统计.....6
五、 数据预处理.....11
（一） 删除无用字段.....11
（二） 类型特征处理.....11
（三） 空值数据处理.....11
（四） 数据不平衡问题.....12
（五） 特征工程.....12
六、 模型初步选择.....12
七、 特征选择方案对比.....14
（一） 财务逻辑特征选择.....14
（二） 技术手段特征选择.....16
八、 最终模型.....18
（一） 特征选择.....19
（二） 参数调整.....19
（三） 模型训练及测试.....19
九、 总结.....20

基于机器学习的发债主体违约风险预测

一、问题定义

改革开放后，随着市场经济的蓬勃发展，我国债券市场经历了从无到有再到发展壮大的过程。在债券市场中，企业债的规模和占比都不断上升。债券在帮助企业融资、促进实体经济发展的同时也在客观上带来了违约风险，为投资者乃至整个金融市场增加不确定性。

在这样的现实问题面前，对企业债券市场进行研究，识别发债主体的违约风险就具备了重要的现实意义，可以对市场在债券的买入、评级、定价等环节发挥帮助作用。具体来说，本文尝试以发债企业作为研究对象，基于债券发行主体是否发生违约行为作为企业违约风险变量，同时利用财务逻辑和相关系数、Lasso 等技术手段对特征指标进行有效筛选，并构建基于多种机器学习算法的发债企业违约风险概率预测模型。

二、问题背景

近十年来，随着实体经济的不断增长，我国债券市场高速发展，在国内的战略地位与国际的影响力都进一步提高，当前债券规模位居世界第二。债券市场快速发展的同时也出现债券违约事件频发的问题，违约风险较以往大幅提高。从 2014 年第一只“11 超日债”违约到 2021 年 2 月，企业债券违约总数达 737 只，违约债券发行总额 6826.37 亿元，债券余额高达 5909.03 亿元。此外，我国高信用评级债券违约明显增多，AA 级以上债券违约占比逐年提高。2018 年，AAA 评级的沪华信发行的多只债券发生违约行为拉开了高评级主体违约的序幕，随后，民生投资、青海盐湖、北大方正和中融新大等 AAA 发行主体也接连发生了违约行为。有效地构建出企业债券违约风险预警模型，以防范和应对潜在的企业债券违约风险，维护中国债券市场的稳定健康发展，更好地为实体经济的发展服务，就显得十分重要。

三、主要方法

（一）LassoCV

Lasso 是利用压缩估计的思想，将参数估计与变量选择同时进行的一种正则化的方法。Lasso 参数估计被定义如下：

$$\hat{\beta}(Lasso) = argmin^2 |y - \sum_{j=1}^{p} x_i \beta_j|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

其中，λ为非负正则参数，$\lambda \sum_{j=1}^{p} |\beta_j|$被称为惩罚项。

LassoCV 的损失函数及损失函数的优化方法与 Lasso 相同，区别在于验证方法。LassoCV 对超参数 a 使用交叉验证，寻找最合适的 a。LassoCV 是进行 Lasso 回归的首选，尤其是从高维特征中寻找主要特征时，LassoCV 优势明显。

（二）LightGBM

LightGBM（Light Gradient Boosting Machine）是 20l6 年微软亚洲研究院公布的一个开源、快速、高效的基于决策树算法的提升（GBDT、GBRT、GBM 和 MART）框架，被用于排序、分类、回归等多种机器学习的任务，支持高效率的并行训练。

LightGBM 的相关理论基础：

(1) Gradient Boosting。

Boosting 是用一系列子模型的线性组合来完成学习任务的，它分为两种类型：AdaBoost 和 Gradient Boosting，LightGBM 属于 Gradient Boosting 的一种。

Gradient Boosting 的思想是：一次性迭代变量，迭代过程中，逐一增加子模型，并且保证损失函数不断减小。假设∫：（X）为子模型，复合模型为：

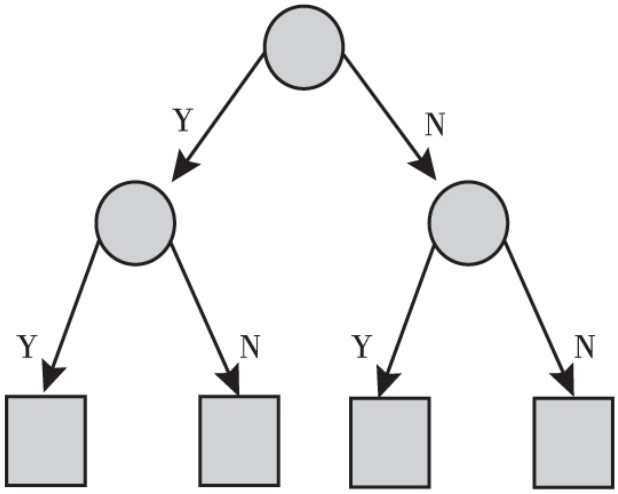
$$F_m(X) = \partial_0 f_0(X) + \partial_1 f_1(X) + \cdots + \partial_m f_m(X)$$

损失函数为 L［F_m(X)，Y］，每一次加入新的子模型后，使得损失函数不断朝着信息含量次高的变量的梯度减小：

$$L[F_m(X), Y] < L[F_{m-1}(X), Y]$$

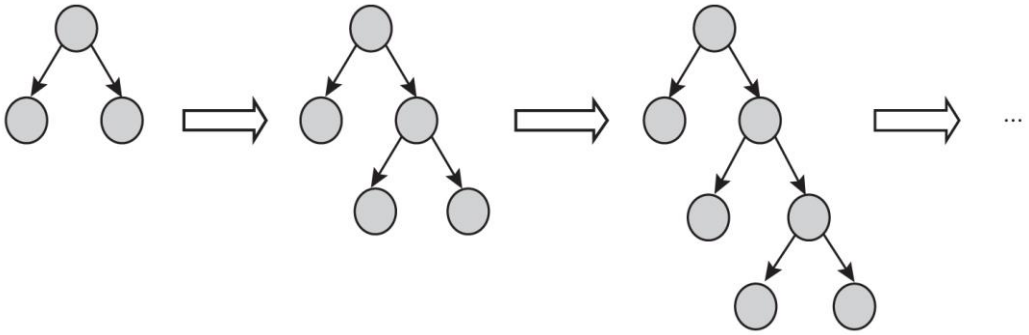
（2）决策树。决策树（Decision Tree）是一种分类和回归的方法，实际研究中大多用于分类。决策树的结构呈树形结构，大多运用的是二权树，在每一个叶子节点上，根据某一判断条件，输出“符合条件”和“不符合条件”两类，不断重复向下输出。可以把决

策树理解成众多 fthe 规则的集合，也可以认为是定义在特定空间与类空间上的条件概率分布。决策树的创建包括 3 个主要步骤：特征选择、决策树的生成和决策树的修剪，该方法具有可读性高、分类速度快的优点。



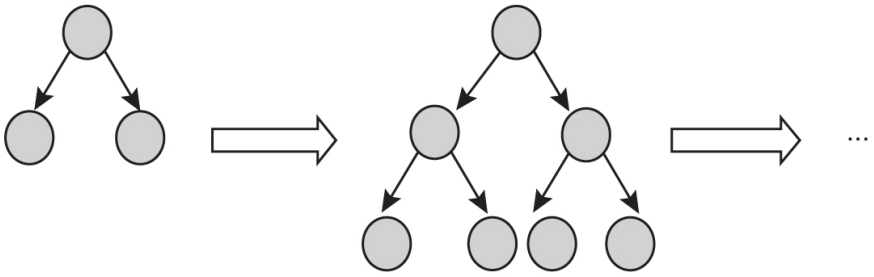
决策树结构

决策树的分裂方法分为两类，一类是按叶子分裂的学习方法（Leaf－wise Learning）；另一类是按层分裂的学习方法（Level－wise Learning）。按叶子分裂的学习方法是指在分裂的过程中要不断地寻找分裂后收益最大的节点，对其进行进一步的分裂，其他非收益最大化的结点不再继续分裂，以这样的规则生长这棵树。这样做的优点是可以使算法更加快速有效；缺点是会忽略掉那些被舍弃的叶子上的信息，导致分裂结果不够细化。



按叶子分裂的决策树学习过程

按层分裂的学习方法与按叶子分裂的学习方法不同，它不需要挑选收益最大化的节点，每一层的每一个结点都要进行分裂，也就是说每次迭代都要遍历整个训练数据的所有数据。优点是每一层的叶子可以并行完成，具有天然的并行性；缺点是这样会产生很多没有必要的分裂，需要更多的计算成本，同时，也会占用较大的运行内存。



按层分裂的决策树学习过程

（3）GBDT。GBDT（Gradient Boosting Decision Tree）是机器学习中一个长盛不衰的模型，事实上：

GBDT＝Gradient Boosting＋Decision Tree

即若 Gradient Boosting 中的每一个子模型都是一个 Decision Tree，这样的模型就是 GBDT。

GBDT 拥有着 Gradient Boosting 和 Decision Tree 的功能共同特性，具有训练效果好、不易过拟合等优点。GBDT 的工具主要包括 XGBoost、Pgbrt、Sklearn、RGBM 等。GBDT 在工业界应用广泛，通常被用于点击率预测，搜索排序等任务。GBDT 也是各种数据挖掘竞赛的致命武器，据统计 Kaggle 上的比赛有一半以上的冠军方案都是基于 GBDT。

2．LightGBM 应用

LightGBM 是 GBDT 的一种，被提出的主要原因是为了解决 GBDT 在海量数据遇到的问题，让 GBDT 可以更好更快地用于实践。

LightGBM 中的决策树子模型是采用按叶子分裂的方法分裂节点的，因此它的计算代价比较小，也正是因为选择了这种分裂方式，需要控制树的深度和每个叶子节点的最小数据量，从而避免过拟合现象的发生。LightGBM 选择了基于 Histogram 的决策树算法，将特征值分为很多个小“桶”，进而在这些“桶”上寻找分裂，这样可以成小储存成本和计算成本。另外，类别特征的处理，也使得 LightGBM 在特定数据下有比较好的提升。

LightGBM 分为三类：特征并行、数据并行和投票并行。特征并行运用在特征较多的场景，数据并行应用在数据量较大的场景，投票并行应用在特征和投票都比较多的场景。

LightGBM 通过以下几个主要的参数实现算法控制与优化：

Num_leaves：每棵数的叶子数量

Learning_rate：学习率

max_depth：最大学习深度。限制树模型的最大深度，用于控制过拟合现象。当 max depth＜0 时，没有学习深度的限制

min_data：一片叶子中数据的最小数量，可以用来控制过拟合现象

feature_fraction：选择特征占总特征数的比例，取值在 0 到 1 之间。当 feature_fraction ＜0 时，LightGBM 在每一次迭代时会随机选择部分特征，feature_fraction 用于控制选择总特征数的比例。该参数可以用于加快训练速度，并且控制过拟合现象

bagging fraction：选择数据占总数据量的比例，取值在 0 到 1 之间。与 feature_fraction 类似，但是随机并且不重复选择的是相应比例的观测，注意要将其设置成大于 0 的比例。该参数可以用于加快训练速度，并且辅助控制过拟合现象

LightGBM 算法自 2016 年发布以来，已经广泛运用于大数据机器学习领域，与之前的 XGBoost 并称为当今机器学习的“倚天屠龙”。公开数据的实验表明 LightGBM 能在学习效率和准确率上都表现出比其他已有 Boosting 工具更好的表现，相比 XGBoost 速度更快，内存占用更少，准确率更高。此外，实验也表明 LightGBM 通过使用多台机器进行特定设定的训练，它能取得线性加速效果。因此，总结该算法的优点显著体现在如下五个方面：①更快的训练速度；②更低的内存消耗；③更好的模型精度；④支持并行学习；⑤可以快速处理海量数据。将性能优良的 LightGBM 算法运用于信用评级系统，其可靠性和灵活性将大大促进相关领域的长足发展。

四、数据收集

（一）数据获取

本项目收集了发债企业 2019-2020 年之间的违约数据与 2018-2020 年的财务指标数据与企业基本信息数据，在债券市场信用风险加速暴露、违约事件发生趋于常态化的背景下，我们将研究如何利用这些数据构建模型对发债企业违约风险进行预测。

数据获取的具体方法为先根据企业所发行的信用债数据寻找 2019 年-2020 年这两

年时间发行过信用债的样本公司，再根据信用债从 2019 到 2020 年底是否违约得到企业是否违约数据作为标签数据，最后获取每家公司 2018 年-2020 年每个季度的所有财务数据和基本信息数据。

（二）数据概述

经测算，由于一共有 3 年时间 12 个季度的数据，因此最终数据一共有 62700 条，因子一共有 178 类，构成一个 62700*178 的矩阵，包含了企业所有的财务数据和基本信息数据。

数据集中一共有 7811 家独立企业，其中涉及 317 条违约记录，本项目将出现违约记录的企业均标记为违约公司，最终处理后的结果在 62700 条数据中有 766 条数据被标注为违约数据。

（三）重要数据描述性统计

根据金融经济逻辑，在 178 类因子中存在某些因子可能会更加有效，因此先预先对这些因子进行简单的验证。

（1）重要财务指标

根据金融逻辑，提取一些直观上认为有效的偿债能力指标，对违约企业和未违约企业数据分别取均值进行对比，得到结果如下表。

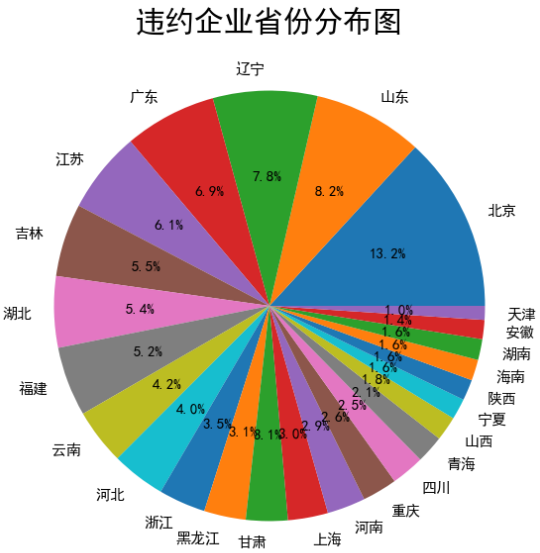
类型	因子名	违约企业	未违约企业
短期偿债能力	流动比率	1.222264874	2.589189578
	速动比率	0.822166715	1.455936401
	保守速动比率	0.645962609	1.226635392
	货币资金／流动负债	0.202060894	0.446762195
长期偿债能力	资产负债率	77.70589816	58.98493826
	已获利息倍数(EBIT/利息费用)	0.837441733	10.00282245
	有形资产/净债务	0.546540629	2.396915918
	有形资产/带息债务	0.365623654	1.820119039
盈利能力	销售毛利率	14.26037442	23.77843175
	净利润/营业总收入	-90.62120764	10.77671049
	净资产收益率	-17.26924639	3.413194243
	总资产报酬率	-0.784615217	2.2068182
营运能力	存货周转率	2.771852212	4.521155079
	应收账款周转率	5.490943021	12.54474573
成长能力	营业收入同比增长率(%)	-15.02870804	16.41122521
	同比增长率-利润总额(%)	-159.5523076	17.18191813
	净资产(同比增长率)	-17.13311972	11.94789362
	同比增长率-净资产收益率(摊薄)(%)	-517.5668421	4.1903428
现金流量能力	经营活动产生的现金流量净额/流动负债	0.015952106	0.007111116
	经营活动产生的现金流量净额/负债合计	0.007676927	0.013822367

上表根据企业的短期偿债能力、长期偿债能力、盈利能力、营运能力、成长能力、现金流量能力得到六个维度，一共找了 20 个指标。

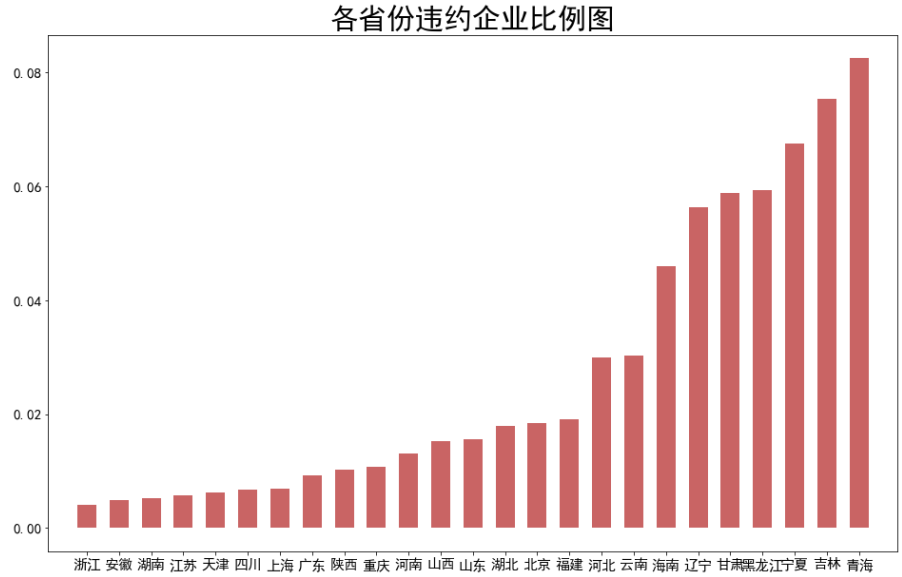
由表可以看出，违约企业与未违约企业对比来看，各项指标均值违约企业都会更差。可以明显看出，违约企业具有更低的短期偿债能、长期债能力、盈利能力、营运能力、成长能力、现金流量能力。并且有些指标上甚至有比较大的差距，因此我们可以初步认为上述指标有一定效果，可以用于企业违约预测。

（2）省份因子

通过绘制违约企业的省份分布饼状图，可以看到不同省份企业违约的风险具有显著差距。较多违约企业集中在北京、山东、辽宁。

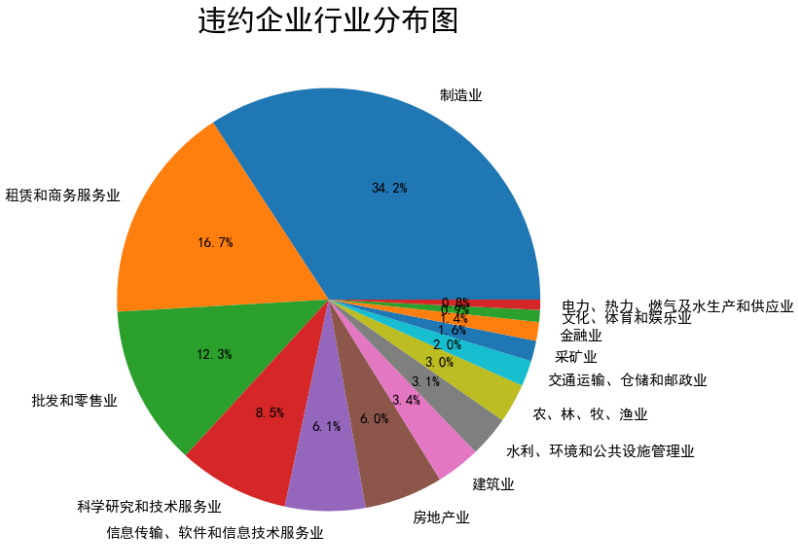


计算各个省份企业中违约比例，得到如下图。可以发现青海、宁夏、吉林等地的企业具有更高的违约比例。因此初步认为企业属地也是一个有效因子。

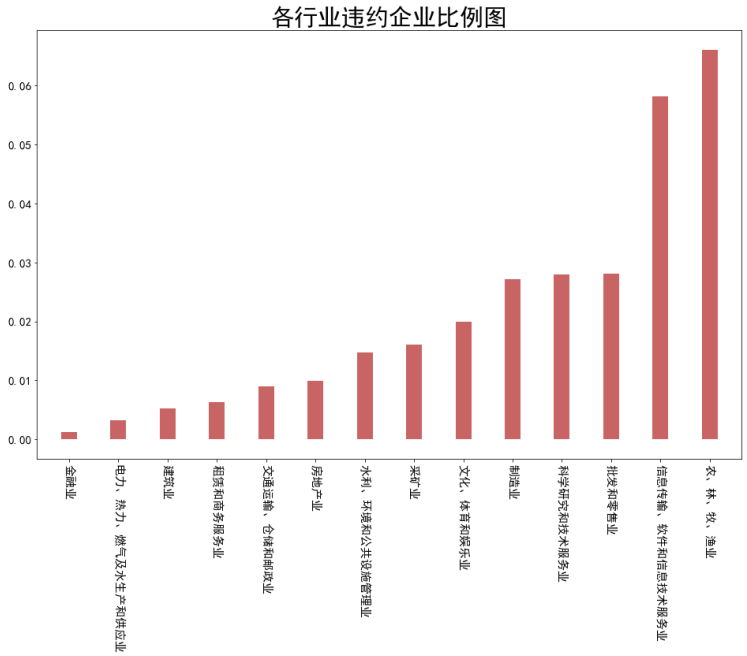


（3）行业因子

通过绘制违约企业行业分布饼状图，可以发现较多违约企业集中在制造业、租赁和商务服务业、批发和零售业，各行业所占比例具有显著差距。

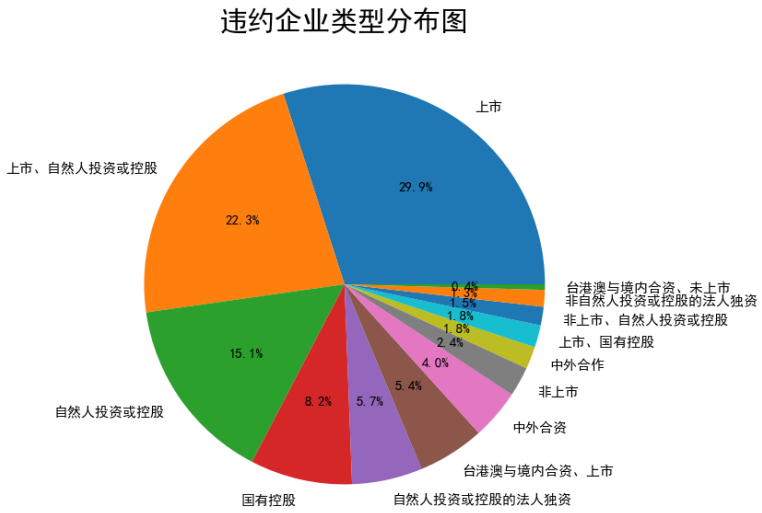


计算各个行业违约比例得到如下图，可以明显发现农、林、牧、渔业和信息传输、软件和信息技术服务业这两个行业具有更高的违约企业比例，均超过 5%，行业间违约比例存在显著差距，因此可以初步认为行业属性是一个有效因子。

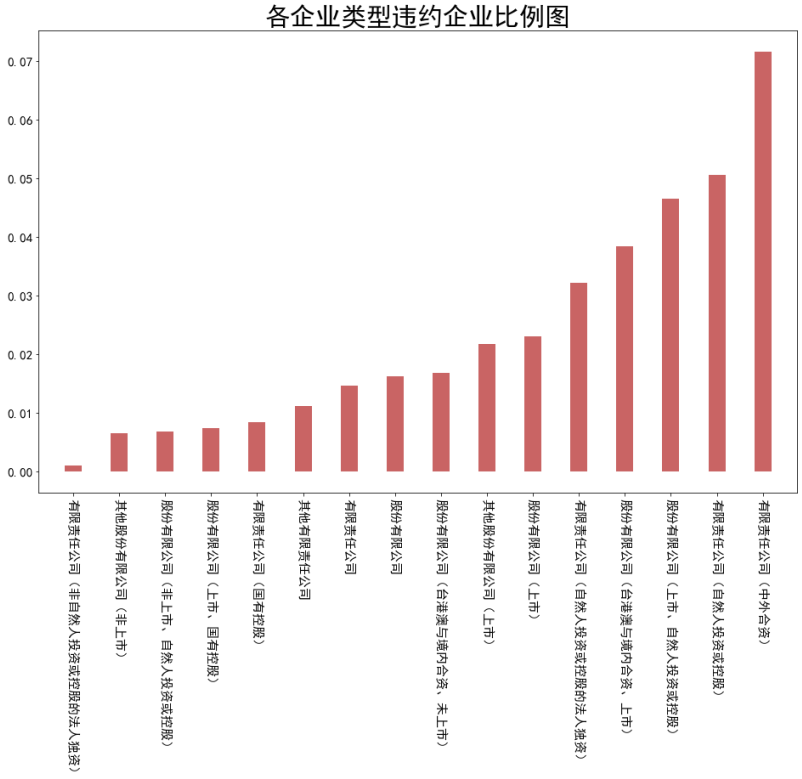


（4）企业类型因子

通过绘制违约企业类型分布饼状图，可以发现较多违约企业集中上市、上市、自然人投资或控股，各类型所占比例具有显著差距。



计算各个行业违约比例得到如下图，可以明显发现有限责任公司（中外合资）、有限责任公司（自然人投资或控股）这两个企业类型的企业具有更高的违约企业比例，这样的偏外资和偏私有制的企业更容易违约，而国资背景的企业由于国资的担保具有更低的违约率，这与一般认识也相符。不同类型企业间违约比例存在显著差距，因此可以初步认为企业类型属性是一个有效因子。



五、数据预处理

原先得到的数据为 62700*178 的矩阵，包含 62700 条数据和 178 个因子，但该数据为原始数据，如需代入模型使用需经过预处理加工后使用，下面是我们的预处理流程。顺序为删除无用字段、类型特征处理、空值数据处理、数据不平衡问题处理。

（一）删除无用字段

将企业 ID、是否发债（均为 Y）、成立日期，存续日期、经营期限、核准日期、所属城市、所属县等明显无效字段删除。

（二）类型特征处理

数据有几个特征为类型特征，如企业类型、所属行业、所属省份，由于数据为非数字型不可直接用于模型，因此首先进行编码操作。其中行业一共有 20 个、省份为 31 个、企业类型有 35 个，每种类型对应一个编码，选取前几条数据得到如下图所示。

	industryphy	enttype	prov
0	17	18	14
1	15	26	29
2	15	26	11
3	17	34	18
4	17	34	18

（三）空值数据处理

第一步：首先进行空值比例统计，观察各个因子下的数据空值比例，前几条因子如图所示。

```
s_fa_roe_avg                    0.839490
rd_expense                      0.791148
s_qfa_deductedprofittoprofit    0.778038
s_fa_deductedprofittoprofit     0.750638
s_qfa_roe_deducted              0.742408
s_fa_yoyeps_diluted             0.740797
s_qfa_deductedprofit            0.740287
s_fa_eps_diluted                0.735391
s_fa_yoynetprofit_deducted      0.728469
s_fa_yoyeps_basic               0.725136
dtype: float64
```

第二步：删除空值比例多于 30%的特征，下图即为被删除的因子。

```
['rd_expense', 's_fa_capitalizedtoda', 's_fa_deductedprofit', 's_fa_deductedprofittoprofit', 's_fa_ebitda', 's_fa_ebitdatodebt', 's_fa_eps_basic', 's_fa_eps_diluted', 's_fa_extraordinary',
's_fa_longdebttoworkingcapital', 's_fa_ocftooperateincome', 's_fa_roe_avg', 's_fa_roe_deducted', 's_fa_yoyeps_basic', 's_fa_yoyeps_diluted', 's_fa_yoynetprofit_deducted', 's_qfa_deductedprofit',
's_qfa_deductedprofittoprofit', 's_qfa_expensetosales', 's_qfa_finaexpensetogr', 's_qfa_gctogr', 's_qfa_grossprofitmargin', 's_qfa_impairtogr_ttm', 's_qfa_investincome',
's_qfa_investincometoebt', 's_qfa_ocftoor', 's_qfa_operateincometoebt', 's_qfa_roe_deducted', 's_qfa_saleexpensetogr', 's_qfa_salescashintoor', 's_qfa_yoygr', 's_qfa_yoynetprofit',
's_qfa_yoyop', 's_qfa_yoyprofit', 's_qfa_yoysales', 's_stm_is', 's_stmnote_finexp']
```

第三步：用均值填充空值

（四）数据不平衡问题

经计算，样本中仅有 1.2%违约样本，因此存在严重的数据不平衡问题，因此本项目根据相关文献的建议，针对数据不平衡现象采用对未违约样本欠采样的处理方法。重新随机采取 1000 条未违约样本，加上本已存在的 766 条违约样本数据，一共有 1766 条数据。经前文所述因子处理后，仍余下 130 个因子，最终所用样本数据为 1766*130 的矩阵。

（五）特征工程

经过数据预处理后，我们仍有 130 个特征因子，但这些因子肯定存在一定量的无效因子，通过选取更有效的因子能大幅提高模型效果，因此接下来将进行特征工程的研究。特征工程部分，本项目采用两条思路，一条是纯技术的手段提取有效因子，另一条是金融逻辑。

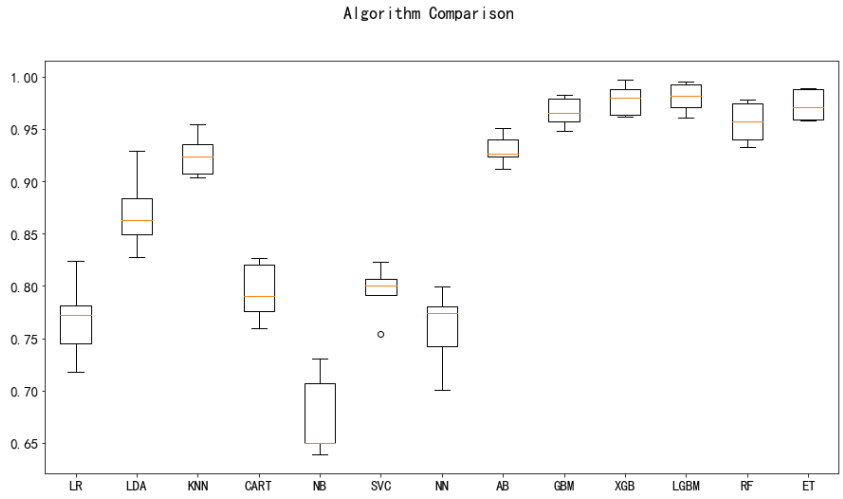
六、模型初步选择

本部分将在未经特征选择的数据上对不同模型进行比较，初步选取表现良好模型，待选模型包括：LR、LDA、KNN、CART、NB、SVC、NN、AB、GBM、XGB、LGBM、

RF、ET。采取的方法为 K 折交叉验证（K=5）。

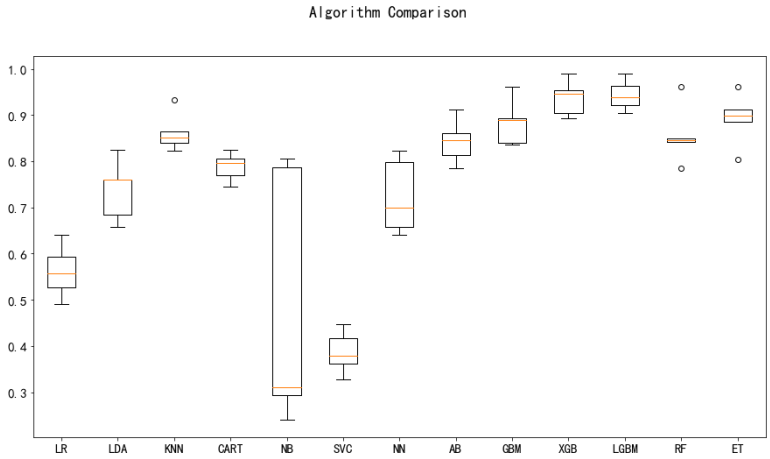
首先选用模型评估指标为 roc_auc，该指标为模型评估常用指标，模型评估结果如下图所示，第一张图为各个模型评价指标均值与括号内的标准差，第二张图为模型评价箱线图。由图可知，LGBM 模型具有最好的预测效果，GBM、XGB、RF、ET、AB 也有不错的效果。

```
LR: 0.768199 (0.035578)
LDA: 0.870703 (0.034595)
KNN: 0.924931 (0.018367)
CART: 0.794793 (0.025680)
NB: 0.675423 (0.036386)
SVC: 0.795406 (0.022929)
NN: 0.759348 (0.034543)
AB: 0.930654 (0.013499)
GBM: 0.966412 (0.012945)
XGB: 0.978062 (0.013760)
LGBM: 0.980171 (0.013042)
RF: 0.956453 (0.017816)
ET: 0.972783 (0.013469)
```



为了保证模型评估的稳健性，再选用模型评估指标为 recall。选用 recall 的原因是我们比较关注于将真正违约的企业识别出来，为此我们愿意放弃一定正确率，容忍把未违约企业判定为违约企业的情况，此时的 recall 反应了实际违约的企业有多少被我们的模型识别。模型评价如图所示，第一张图为各个模型评价指标均值与括号内的标准差，第二张图为模型评价箱线图。由图可知，LGBM 取得最优效果，XGB 也有不错的效果。

```
LR: 0.561873 (0.051976)
LDA: 0.737594 (0.059970)
KNN: 0.862703 (0.037285)
CART: 0.800090 (0.046104)
NB: 0.487240 (0.253207)
SVC: 0.386366 (0.041567)
NN: 0.699293 (0.106193)
AB: 0.843672 (0.043494)
GBM: 0.885815 (0.043209)
XGB: 0.937380 (0.035571)
LGBM: 0.943344 (0.030500)
RF: 0.856551 (0.052551)
ET: 0.899474 (0.055202)
```



综上所述，经过两次比较结果如下图所示，两次比较中 LGBM 模型均取得最优结果，因此我们将把 LGBM 作为我们的所选模型进一步调参优化。

	LR	LDA	KNN	CART	NB	SVC	NN	AB	GBM	XGB	LGBM	RF	ET
recall	0.56	0.74	0.86	0.79	0.49	0.39	0.72	0.84	0.88	0.94	0.94	0.86	0.89
roc_auc	0.77	0.87	0.92	0.79	0.68	0.80	0.76	0.93	0.97	0.98	0.98	0.95	0.97

七、特征选择方案对比

（一）财务逻辑特征选择

企业相关的原始特征指标有将近 130 个，数据比较繁杂、信息比较重复，且并非每个原始特征指标都有意义，因此按照财务分析逻辑从原始特征指标中筛选出一批相对重要的特征如下：

指标类型	变量名称	变量含义
基本信息	国民经济行业门类	企业主营业务所在行业
	国民经济行业	
	企业类型	企业所有制形式，如央企、国企、民企等
	所属省	企业所在省份
短期偿债能力	流动比率	流动资产/流动负债
	速动比率	（流动资产-存货）/流动负债
	保守速动比率	（货币资金+短期投资+应收账款及票据）/流动负债
	货币资金比率	货币资金/流动负债
长期偿债能力	资产负债率	负债总额/资产总额
	利息保障倍数	息税前利润/利息费用
	长期债务与营运资金比率	长期债务/（流动资产-流动负债）
	有形资产与净债务比率	有形资产/净债务
	有形资产与带息债务比率	有形资产/带息债务
盈利能力	销售毛利率	销售毛利润/销售收入
	净利润率	净利润/营业收入
	净资产收益率	净利润/净资产
	总资产收益率	净利润/总资产
营运能力	存货周转率	营业收入/平均存货余额
	应收账款周转率	营业成本/应收账款平均余额
成长能力	营业收入同比增长率	本年度营业收入/上年度营业收入-1
	利润总额同比增长率	本年度利润总额/上年度利润总额-1
	净资产同比增长率	本年末净资产/上年末净资产-1
	净资产收益率同比增长率	本年度净资产收益率/上年度净资产收益率-1
现金流量能力	经营净现金比率(短期债务)	经营性现金流量净额/流动负债
	经营净现金比率(全部债务)	经营性现金流量净额/负债总额

从以上七大方面相对全面地覆盖了企业的各方面能力。其中，短期偿债能力反映了企业流动资产等对短期债务的覆盖情况；长期偿债能力反映了企业总资产等对长期债务的覆盖能力；盈利能力反映了企业经营获取利润的情况；营运能力反映了企业正常经营运行的效率情况；成长能力反映了企业业绩增长情况；现金流量能力反映了现金收付覆盖债务的情况。将原始近 130 个特征按照以上七大方面进行分类、筛选，最后得到 25 个重要特征。

将选好的特征导入 LGBM 模型中，对训练集和测试集进行训练学习和测试。相关代码如下：

```
dataset_fin = dataset[not_drop_list_fin]
X = dataset_fin.drop("isDefault", axis=1)
Y = dataset_fin["isDefault"]
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=seed)
# 训练模型
model = lgb.LGBMClassifier()
model.fit(X_train, Y_train)
# 预测
Y_pred = model.predict(X_test)
```

（二）技术手段特征选择

A. 相关系数特征选择

皮尔森相关系数是一种最简单的，能帮助理解特征和响应变量之间关系的方法，该方法衡量的是变量之间的线性相关性，结果的取值区间为[-1，1]，-1 表示完全的负相关，+1 表示完全的正相关，0 表示没有线性相关。利用相关系数进行特征选择的优点是速度快、易于计算；缺点是作为特征排序机制，他只对线性关系敏感，如果关系是非线性的，即便两个变量具有一一对应的关系，Pearson 相关性也可能会接近 0。

可以利用 Pandas 中 DataFrame.corr()方法计算相关系数矩阵：

```
correlation = dataset.corr()
correlation_isDefault = abs(correlation['isDefault'])
correlation_isDefault.sort_values(ascending=False)
```

```
isDefault                  1.000000
s_fa_current               0.319152
industryphy                0.301979
s_fa_longdebtodebt         0.289156
s_fa_currentdebttodebt     0.286215
                             ...
s_fa_arturndays            0.007248
s_fa_cfps                  0.006385
s_fa_nonoperateprofittoebt 0.005522
s_fa_noptoebt              0.005227
s_fa_optoebt               0.005208
Name: isDefault, Length: 130, dtype: float64
```

根据相关系数，从中选择绝对值不低于 0.03 的特征。

```
# 与isDefault相关性低于0.03的特征
drop_list_corr = list(correlation_isDefault[correlation_isDefault<0.03].index)
```

B. LassoCV 特征选择

Lasso 算法可以使特征的系数进行压缩并且可以使某些回归系数为 0，即不选用该

特征，因此可以进行特征选择。

Lasso 回归方法的优点是可以弥补最小二乘法和逐步回归局部最优估计的不足，可以很好的进行特征选择，可以有效的解决各特征之间存在的多重共线性问题。缺点是如果存在一组高度相关的特征时，Lasso 回归倾向于选择其中一个特征，而忽视其他所有特征，这种情况会导致结果的不稳定性。

通过 sklearn 中的 LassoCV 模型，对数据进行拟合，得到 Lasso 模型。

```
X = dataset.drop("isDefault", axis=1)
Y = dataset["isDefault"]
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.3, random_state=9)
model_lasso = LassoCV(alphas=[0.1, 1, 0.001, 0.0005], cv=10).fit(X_train, Y_train)
coef = pd.Series(model_lasso.coef_, index=X_train.columns)
```

然后从中选择系数等于 0 的特征。

```
# 系数等于0的特征
drop_list_lasso = list(coef[coef==0].index)
```

C. XGBoost 特征选择

XGBoost 是 boosting 算法的一种实现方式，主要是降低偏差，也就是降低模型的误差。因此它是采用多个基学习器，每个基学习器都比较简单，避免过拟合，下一个学习器是学习前面基学习器的结果和实际值的差值，通过多个学习器的学习，不断降低模型值和实际值的差。

XGBoost 算法通过梯度提升的方法，我们可以根据提升之后的树获取每个特征的重要性。一般来说，特征的重要性表示这个特征在构建提升树的作用。如果一个特征在所有树中作为划分属性的次数越多，那么该特征就越重要。通过每个属性分割点改进性能度量的量来计算单个决策树的重要性，并由节点负责的观察数量加权。性能度量可以是用于选择分裂点的纯度（基尼指数）或另一个更具体的误差函数。最后，在模型中的所有决策树中对要素重要性进行平均。最终得到每个特征的重要性,之后可以特征排序或者进行相互比较。

通过 xgboost 软件包中的 XGBClassifier()类拟合数据建立模型，再通过该类提供的 feature_importances_属性得到特征重要性，从中选取排名前 40 的特征。

```
X = dataset.drop("isDefault", axis=1)
Y = dataset["isDefault"]
X_train, X_test, Y_train, Y_test = train_test_split(X,  Y, test_size=0.3, random_state=9)
model = xgb.XGBClassifier()
model.fit(X_train, Y_train)
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
```

```
# 选择模型特征重要性前40的特征
not_drop_list_xgb = list(feat_importances.nlargest(40).index)
not_drop_list_xgb.append('isDefault')
```

（3）效果分析

对未进行特征选择和不同方法特征选择后的模型结果进行对比分析，得到下表：

单位：%

Metrics	未进行特征选择	财务逻辑	相关系数	Lasso	相关性+Lasso	XGBoost
Accuracy	92.45	**94.52**	94.72	94.91	**96.42**	96.23
Recall	93.99	**94.49**	95.76	95.76	**97.03**	97.03
AUC	92.62	**94.52**	94.82	95.00	**96.48**	96.31

可以看到，按照财务逻辑对特征进行选择后，AUC 等指标均有明显提高，其中 Accuracy、Recall、AUC 三个指标分别提高 2.07%、0.50%、1.90%；按照技术手段对特征选择后，各项指标均有明显提高，其中利用相关性+Lasso 的特征选择方案效果最优，Accuracy、Recall、AUC 三个指标分别提高 3.97%、3.04%、3.86%。

从模型结果来看，有以下简单结论：

（1）相对于原始 130 个特征，进行财务逻辑选择后的 25 个特征不仅在数量上更加简化，同时在模型效果上也更优。

（2）发债企业的违约概率与财务数据强相关，利用重要的财务指标对企业违约概率进行预测是可行有效的。

（3）利用技术手段对原始特征进行选择是有效的，其中相关系数和 Lasso 两种方法均可从不同角度对结果进行优化，两种方法共同使用时的模型效果最佳。

八、最终模型

（一）特征选择

经上述多次对比，最终选定采用相关系数及 LassoCV 进行特征的选择，具体过程为：首先计算所有特征与目标变量的相关性，删掉相关系数绝对值小雨 0.03 的特征；其次通过建立 LassoCV 模型，提取模型拟合后各特征变量的系数参数，将系数等于 0 的特征删除，剩余的特征即为最终选定的特征。

最终选定的特征共 98 个：

```
print(list(dataset.columns))
✓ 0.1s                                                                 Python

['s_fa_operateincome', 's_fa_ebit', 's_fa_fcfe', 's_fa_tangibleasset', 's_fa_workingcapital', 's_fa_networkingcapital', 's_fa_retainedearnings', 's_fa_eps_diluted2', 's_fa_bps', 's_fa_surpluscapitalps',
's_fa_ebitps', 's_fa_fcffps', 's_fa_fcfeps', 's_fa_netprofitmargin', 's_fa_grossprofitmargin', 's_fa_expensetosales', 's_fa_profittogr', 's_fa_saleexpensetogr', 's_fa_adminexpensetogr', 's_fa_finaexpensetogr',
's_fa_impairtogr_ttm', 's_fa_gctogr', 's_fa_optogr', 's_fa_ebittogr', 's_fa_roe', 's_fa_roa2', 's_fa_roa', 's_fa_roic', 's_fa_roe_yearly', 's_fa_roa2_yearly', 's_fa_investincometoebt', 's_fa_taxtoebt',
's_fa_salescashintoor', 's_fa_ocftoor', 's_fa_debttoassets', 's_fa_assetstoequity', 's_fa_dupont_assetstoequity', 's_fa_catoassets', 's_fa_ncatoassets', 's_fa_tangibleassetstoassets',
's_fa_intdebttototalcap', 's_fa_equitytototalcapital', 's_fa_currentdebttodebt', 's_fa_longdebtodebt', 's_fa_current', 's_fa_quick', 's_fa_cashratio', 's_fa_debttoequity', 's_fa_equitytodebt',
's_fa_equitytointerestdebt', 's_fa_tangibleassettodebt', 's_fa_tangassettointdebt', 's_fa_tangibleassettonetdebt', 's_fa_ocftonetdebt', 's_fa_ebittointerest', 's_fa_turndays', 's_fa_invturndays', 's_fa_invturn',
's_fa_arturn', 's_fa_faturn', 's_fa_roa_yearly', 's_fa_dupont_roa', 's_stm_bs', 's_fa_prefinexpense_opprofit', 's_fa_nonopprofit', 's_fa_cashtoliqdebt', 's_fa_cashtoliqdebtwithinterest', 's_fa_optoliqdebt',
's_fa_optodebt', 's_fa_roic_yearly', 's_fa_tot_faturn', 's_fa_profittoop', 's_qfa_operateincome', 's_qfa_eps', 's_qfa_netprofitmargin', 's_qfa_profittogr', 's_qfa_adminexpensetogr', 's_qfa_optogr',
's_qfa_roe', 's_qfa_roa', 's_qfa_ocftosales', 's_fa_yoyop', 's_fa_yoyebt', 's_fa_yoynetprofit', 's_fa_yoyroe', 's_fa_yoybps', 's_fa_yoyassets', 's_fa_yoyequity', 's_fa_yoy_tr', 's_fa_yoy_or', 's_qfa_cgrgr',
's_qfa_cgrsales', 's_qfa_cgrop', 's_qfa_cgrnetprofit', 'isDefault', 'industryphy', 'enttype', 'prov']
```

（二）参数调整

采用 sklearn 软件包提供的网格搜索方法 GridSearchCV()，进行网格搜索，以获取最优参数：

```python
model = lgb.LGBMClassifier()
param_dist = {
    "max_depth": [25, 50, 75],
    "learning_rate": [0.01, 0.1, 0.15],
    "num_leaves": [100, 300, 500],
    "n_estimators": [200]
}
kfold = KFold(n_splits=num_folds, random_state=seed, shuffle=True)
grid = GridSearchCV(estimator=model, param_grid=param_dist, cv=kfold, scoring=scoring, verbose=10, n_jobs=-1)
grid_result = grid.fit(X_train, Y_train)
print("Best: %f using %s" %(grid_result.best_score_, grid_result.best_params_))
✓ 1m 8.8s
```

（三）模型训练及测试

建立 LGBM 模型，模型参数设置为网格搜索获取的最优参数，将经特征选择后的数据集按 7:3 的比例划分为训练集和测试集，训练模型并在测试集上进行测试：

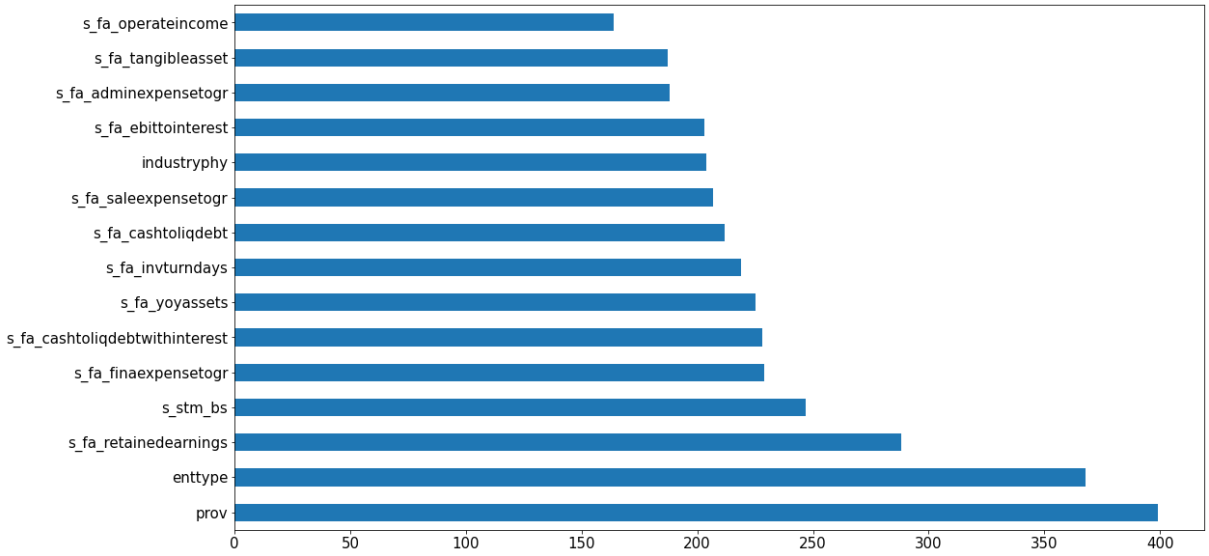
```python
# 训练模型
model = lgb.LGBMClassifier(learning_rate=0.1, max_depth=25, n_estimators=200, num_leaves=100)
model.fit(X_train, Y_train)
# 预测
Y_pred = model.predict(X_test)
```

最终测试结果如下：准确率为 0.964，召回率为 0.974，AUC 为 0.965，预测具有

较好的效果。

（四）模型特征重要性

最终模型训练完毕后，使用模型的 feature_importances_属性可得到每个特征的重要性，下图展示了重要性较高的前 15 个特征：



九、总结

本文以发债企业作为研究对象，基于债券发行主体是否发生违约行为作为企业违约风险变量，同时利用财务逻辑和相关系数、Lasso 等技术手段对特征指标进行有效筛选。进一步地，构建了基于 LGBM、XGBoost 等机器学习算法的发债企业违约风险概率预测模型。为了让模型结果更加准确可靠，本文运用 5 折交叉验证法对预测模型进行实证研究，并采用多种性能评估指标（Accuracy、Recall、AUC 等）和多个基准模型（AB、GBM、XGB、LGBM、RF、ET 等）进行比较研究，最后采用网格搜索法对 LGBM 发债企业违约风险概率预测模型进行参数优化，学习训练测试后得到最终模型的重要特征排序。

实证结果表明：

第一，本文使用的财务逻辑和技术手段特征选择方法能有效提高模型预测违约概率的准确度，其中相关系数+Lasso 方法对模型的优化效果最佳。

第二，发债企业的违约概率与财务数据强相关，利用重要的财务指标对企业违约概率进行预测是可行有效的。其中，企业偿债能力、现金流量能力对企业违约风险有相当

大的贡献。

第三，从技术手段特征选择来看，企业类型、省份对企业违约风险有着较大的重要性。国企、央企的违约风险明显低于民营企业；发达省份企业的违约概率明显低于欠发达省份企业。模型重要性排序和市场真实违约情况相吻合。