基于机器学习的公司特有风险预测方法研究

周越[1] 王郑毅[2]

（坤元资产评估有限公司，浙江杭州 310007[1]；

湘财证券股份有限公司，上海 200120[2]）

【摘要】公司特有风险是企业价值评估特别是收益法评估的重点和难点。本文基于历史实践案例，从A股重大资产重组公开资料中提取可量化数据，并利用特征工程技术对这些数据进行特征筛选和增强，作为输入特征。经网格搜索方法对超参数进行调优后，笔者选取了多个学习器，对前述特征对应的数据进行集中学习，得出特有风险的预测模型。通过实践发现基于递归特征消除算法的特征选择方法能有效提高预测模型的正确率。希望以“特有风险预测”为案例的研究，展示机器学习的操作流程，为评估模型的构建提供新的尝试。

【关键词】评估 收益法评估 公司特有风险 机器学习

【中图分类号】F224 【文献标识码】A 【文章编号】

一、项目背景

折现率在企业价值评估中发挥着至关重要的作用，其可能直接影响并购重组的定价，进而影响交易的成败。遗憾的是，对于折现率的重要组成部分——特有风险，国内实证研究成果不甚理想，未能形成被广泛接受的测算方法。基于此，本文尝试多维度地提取样本的有效属性（特征），利用多种独立的模型捕获特征间关系，辅以网格搜索方法和递归特征消除特征法，以期生成一套较为高效、科学的特有风险测算模型。

二、机器学习概述

（一）机器学习简介

机器学习是一种利用计算机就已知数据构建概率统计模型，再运用该模型开展后续数据分析（学习）进而预测未知数据的数学建模方法。机器学习由模型、策略和算法构成，即学习器=模型+策略+算法，我们将学习器看作模型在给定数据和参数空间上根据给定算法实例化的结果。其中，模型指的是某种函数的集合；策略作为从函数的集合中选择最优函数的准则；而算法则是实现根据策略从模型中选择最优函数的具体计算方法。[1]为了便于区分，我们将学习器

[1] 李航，统计学习方法. 清华大学出版社，2012.

从数据中学习得到的最优函数称为预测模型。

（二）机器学习的一般流程

机器学习一般包含以下 5 个步骤：数据收集、数据准备、学习器选择、特征选择、预测模型评估，具体如下图所示：

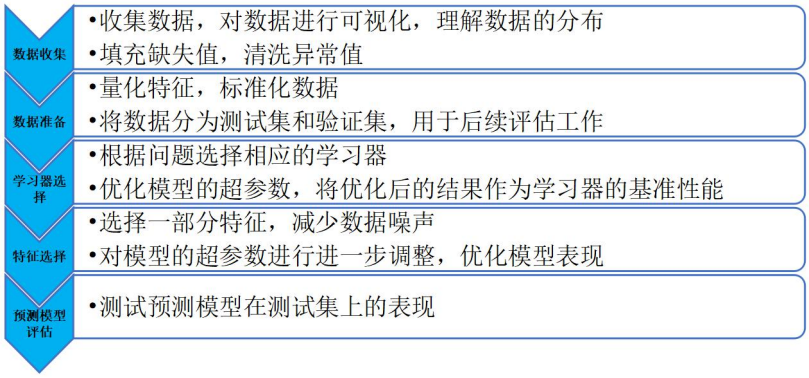

图 1 机器学习的一般流程

（三）机器学习的一般工具

基于 Python 等各类语言的数据分析和学习库成为当今机器学习的主流之一。其中，（简单高效的学习工具）Scikit-Learn、（科学和工程领域较常用的）SciPy、（用于存储和处理大量矩阵的）Numpy 和（常用于金融分析的）Pandas 较为常用。[2]用于进行深度学习的 TensorFlow、PyTorch、Keras，用于处理自然语言的 NLTK，用于机器视觉训练的 OpenCV 等学习库亦不乏应用场景。

通过将各种机器学习库进行整合和优化，结合大数据技术和用于发布服务的 API 发布工具，诸如阿里 PAI、星环 Sophon、百度 BML、4Paradigm Sage Studio 等全流程、低门槛 AI 应用开发与上线平台应运而生。这些平台通过可视化的开发界面，使用户通过选择并连接相应组件的方式，实现导入数据、训练模型、发布服务全流程的低代码开发，大大降低了机器学习的应用门槛。

本文基于 Jupyter Notebook 开发、展示窗口，以及 Scikit-Learn 开发工具开展研究。其中，开发工具可以通过安装 Anaconda3 获取，或在安装 Python 程序后通过“Pip3 install scikit-learn”等命令进行自定义安装。

三、数据收集

本文以通过证监会并购重组委审核的近年 128 宗案例数据为基础开展研究。数据来源包括交易报告书、审计报告、评估报告及说明、反馈意见、其他公开数据等。通过文献研究、财务评价体系参考、并购重组定价和风险影响因素描述性分析，笔者建立了研究的指标体系，由 28 项属性构成，涵盖行业、标的公司历史、股权、经营、技术、财务等风险驱动的重要属性。

[2] https://blog.csdn.net/a673519020/article/details/112471996

在 Jupyter NoteBook 界面，通过如下代码导入数据：

```
In [5]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

unique_risk = pd.read_csv('unique_risk.csv')
```

```
In [6]:
#将审核结果编码为0和1，作为一个可以进行机器学习的指标 （认为“发行股份购买资产获无条件通过”比“发行股份购买资产获有条件通过”好）
#删除公司名称列，该列无法作为特征学习的指标
unique_risk['审核结果']=unique_risk['审核结果'].map({'发行股份购买资产获无条件通过':1,'发行股份购买资产获有条件通过':0})
unique_risk.head(5)
```

Out[6]:

	Rc	审核结果	上一年归母净利润（万元）	承诺期业绩增长率	前三年承诺覆盖率	静态市盈率	动态市盈率	成立年限	大股东持股比例	最近一个完整会计年度对第一大客户的销售占比	...	固定资产周转率	流动比率	资产负债率	经营杠杆（EBITDA/EBIT）	（固定资产+土地）/归母权益	在建工程/归母权益	净资产收益率	毛利率	经营性现金流/收入	研发支出占比
0	0.020	1	15972.0	0.20	0.27	16.47	12.84	17.94	1.00	0.39	...	928.92	1.78	0.55	1.00	0.01	0.00	0.38	0.09	0.02	0.00
1	0.020	0	4443.0	0.27	0.33	15.37	11.68	9.04	0.35	0.23	...	17.93	1.64	0.56	1.03	0.23	0.00	0.69	0.45	0.11	0.04
2	0.010	1	5349.0	0.24	0.25	19.22	13.88	13.56	0.51	0.14	...	1087.77	1.37	0.73	1.02	0.01	0.00	0.23	0.06	0.00	0.00
3	0.005	1	14478.0	0.18	0.30	15.01	11.94	20.32	0.51	0.08	...	147.85	1.61	0.60	1.03	0.05	0.01	0.23	0.09	0.04	0.00
4	0.005	1	14798.0	0.20	0.32	14.19	10.77	16.07	0.51	0.05	...	620.62	1.23	0.81	1.02	0.02	0.00	0.43	0.07	0.04	0.00

5 rows × 29 columns

图 2 数据导入界面

通过 descrbie（）函数对各属性进行了统计描述并绘制直方图，具体如下：

```
In [50]:
unique_risk.describe()
```

Out[50]:

	Rc	审核结果	上一年归母净利润（万元）	承诺期业绩增长率	前三年承诺覆盖率	静态市盈率	动态市盈率	成立年限	大股东持股比例	最近一个完整会计年度对第一大客户的销售占比	...	固定资产周转率	流动比率
count	128.000000	128.000000	128.000000	128.000000	128.000000	128.000000	128.000000	128.000000	128.000000	128.000000	...	128.000000	128.000000
mean	0.023822	0.632812	7987.227656	0.441250	0.273828	37.858750	14.547344	12.834375	1.465391	0.466719	...	189.673125	2.922422
std	0.011094	0.483932	16949.153114	0.442664	0.065690	73.447075	4.293068	6.997562	6.005898	2.046034	...	672.974887	9.977001
min	0.005000	0.000000	-86262.000000	-0.080000	0.100000	-42.030000	5.680000	2.480000	0.090000	0.000000	...	0.000000	0.170000
25%	0.015000	0.000000	1719.250000	0.207500	0.230000	15.665000	11.650000	7.285000	0.365000	0.150000	...	3.950000	1.147500
50%	0.020000	1.000000	4451.500000	0.310000	0.270000	20.835000	13.960000	12.345000	0.510000	0.240000	...	15.770000	1.470000
75%	0.030000	1.000000	9718.500000	0.502500	0.310000	31.085000	16.070000	16.145000	0.820000	0.402500	...	127.072500	2.095000
max	0.050000	1.000000	94597.000000	2.970000	0.420000	609.820000	30.100000	37.590000	45.500000	23.320000	...	6898.070000	113.260000

8 rows × 29 columns

图 3 样本的统计描述

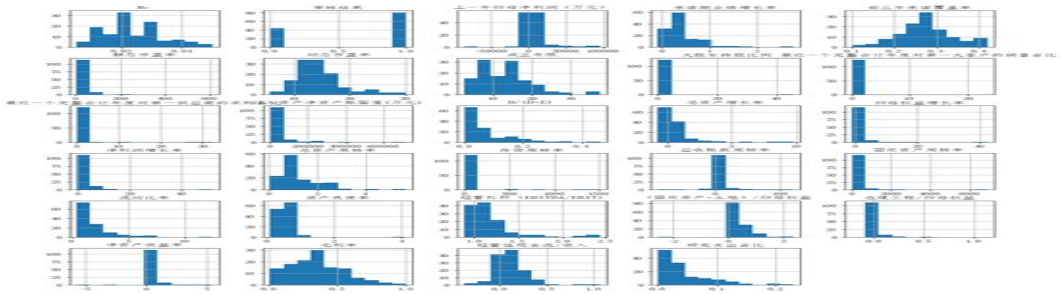

图 4 各属性直方图

由图显示，实践中特别风险的最大取值为 5%，最小为 0.5%，主要集中于[1.5%, 3%]区间内，数据集不存在缺失值，但部分属性离散程度较大。

由于共选择了 28 个属性进行研究，因此需通过降维的方法进行数据可视化处理。考虑到常用的 PCA（Principal Component Analysis）主成分分析算法为线性算法，难以解释属性间的复杂多项式关系，不能将相似数据点放置一起展示，因此笔者选取 t-SNE 算法进行数据降维。

t-SNE（t-distributed stochastic neighbor embedding，T-分布邻域嵌入算法）是一种用于挖掘高维数据的非线性降维算法，适用于将高维数据降维到二维或三维后进行可视化处理。该算法核心思想是将欧几里得距离转换为服从 t 分布

的条件概率来表达点与点间的相似度，能较好地描述点之间的相似度。

本次研究数据存在异常值，笔者将其定义为离群点（outlier），即远离具有相同分布的内点（inlier）的样本。由于离群点会影响模型拟合的效果，因此需要对其进行检测和剔除。常用的异常值检测方法如下：

算法	特点	应用场景
3Sigma	计算方便，需要先验知识	帮助识别哪些特征具有较强的解释性
One Class SVM	对异常的检测敏感，准确率较低	公司特征质量分析，新输入样本是否与已有样本群体相似
Local Outlier Factor	适用于不了解数据分布的情况	全数据集中寻找异常数据
Isolation Forest	适用于大数据高纬度，鲁棒性较好，效果较优	

表 2 异常值检测的常用方法

本次使用 Local Outlier Factor 和 Isolation Forest 方法对离群点进行检测，结果如下：

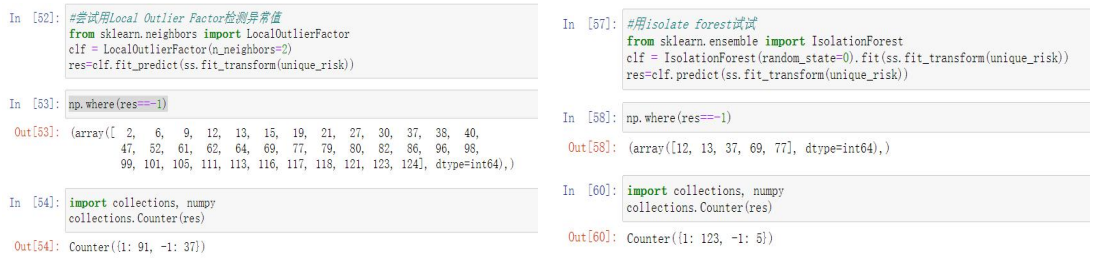

图 5 Local Outlier Factor（左）及 Isolation Forest 离群点检测过程

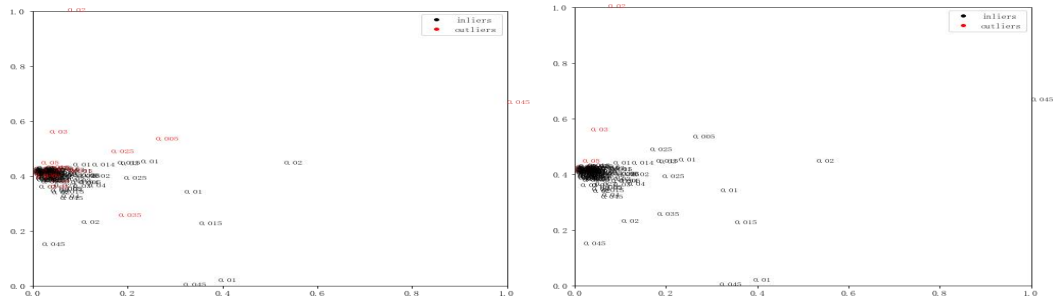

图 6 Local Outlier Factor（左）及 Isolation Forest 离群点检测图

由图显示，Local Outlier Factor 在离群点的检测上更敏感，可视化后也更符合直观感受，因此，笔者选择其结果作为后续训练的数据集。

四、数据准备

由于原始数据集内存在不可量化数据，且各属性口径存在较大差异，因此需要对数据进行编码和标准化，以增强数据的可用性。此外，为便于对生成的预测模型进行评估，本次将数据分为训练集和测试集两部分。

（一）特征构建

对于“审核结果”属性，原始数据集内表现为“发行股份购买资产获无条件通过”和“发行股份购买资产获有条件通过”两个字符串，鉴于无法直接输入到模型中，考虑到“发行股份购买资产获无条件通过”比“发行股份购买资产获有条件通过”更优，故分别将“发行股份购买资产获有条件通过”“发行股份购买资产获无条件通过”编码为 0 和 1。

（二）数据标准化处理

由于不同变量间的量级存在较大差异，学习器的算法往往会被数值大的属性所主导，因此需对数据进行标准化处理。考虑到原始数据不完全符合正态分布，本次选用 Min-Max 标准化方法进行数据处理。Min-Max 标准化公式如下：

$$m=\frac{x-x_{min}}{x_{max}-x_{min}}$$

式中： m 是新值；x 是单元格原始值；x_{min} 及 x_{max} 分别是该列的最小和最大值。

标准化后各属性取值位于[0,1]区间内，规避了算法受数据尺度影响的不利情形，结果更为精准。

（三）分离训练集和测试集

由于机器学习的复杂性，预测模型有时会过于紧密或精确地匹配已知数据集，以致缺乏泛化能力，无法很好地预测未来的观察结果，即出现过拟合。因此，笔者将数据集的 90%部分用于生成预测模型，其余部分作为测试集，用于后续评估预测模型对于未知数据的预测能力。

```
In [12]: #抽取 90% 的数据作为训练集，10%的数据作为测试集，用来展示模型的结果。
         #实际是在完成特征选择、调参等一系列过程后，用所有已知数据训练得到最终模型，保存模型文件，进行使用。
         #并定期重新更新数据集（一般会设定一个窗口），定期重新训练、更新模型。
         #由于抽样是随机的，为了使结果具有可复现性，这里额外设定了一个随机数种子 random_state=22
         from sklearn.model_selection import train_test_split
         train, test = train_test_split(unique_risk, test_size = 0.1,random_state=22)

In [13]: #分解成为特征矩阵和结果矩阵
         train_X = train.drop(labels=["Rc"], axis=1)
         train_y=train['Rc']

         test_X= test.drop(labels=["Rc"], axis=1)
         test_y =test['Rc']

In [14]: from sklearn.preprocessing import MinMaxScaler
         # 采用 Min_Max 规范化数据，保证每个特征维度的数据均值为 0，方差为 1
         ss = MinMaxScaler()
         train_X = ss.fit_transform(train_X)
         test_X = ss.transform(test_X)
```

图 7 对数据进行标准化并拆分为训练集和测试集

五、学习器选择

由于数据集输出数据的连续性，因此特有风险的预测属于机器学习中的回归问题。机器学习的回归模型可以分为广义线性、树、支持向量机、K 近邻、Bagging 集成、Boosting 集成、多层感知机（神经网络）回归等七类模型。此次研究中，我们选择最小化模型在训练集上的 R^2 作为学习策略，使用 Scikit-Learn 中的算法选择最优模型。

（一）模型介绍

1.岭回归

岭回归在最小二乘法的基础上，利用添加 L2 范数对系数进行惩罚的方法，

对属性间具有完全共线性或高度相关性的情形进行优化。

2.K 近邻回归

K 近邻回归属于懒惰模型，其并不从训练数据中生成判别函数，而是基于某种距离度量找出训练集中最靠近待预测样本的 k 个训练样本，将这 k 个邻居的输出值的平均值标记为待预测样本的预测结果。

3.多层感知机回归

多层感知机回归即按照每层感知机与下一层感知机的全连接，感知机间以不存在同层或者跨层连接的方式构建一个多层前馈神经网络，通过误差逆传播算法不断调整感知机之间的权值，最终获得一个复杂的非线性预测模型。

4.Boosting 类回归

Boosting 类回归的核心思想是通过增大错误样本的权重，将多个弱预测模型组合成一个强预测模型进而实现回归。其中比较常用的模型有 AdaBoost（Adaptive Boosting，自适应增强模型）、XGBoost（eXtreme Gradient Boosting，极限梯度提升模型）、LGBM（Light Gradient Boosting Machine，轻量级的高效梯度提升模型），其主要区别在于如何识别模型和权重的调整方法上。

前述模型的优劣势可以通过下表进行描述：

回归模型	特点	优势	劣势
线性回归-最小二乘	最小化均方误差	简单，结果直观； 可人为修正学习结果	对离群点敏感； 属性较多时易过拟合； 无法处理非线性问题
岭回归	最小化添加了 L2 范数的损失函数	简单，结果直观； 可人为修正学习结果； 能一定程度避免过拟合	属性较少时易欠拟合； 无法处理非线性问题
K 近邻回归	寻找离样本点最近的 K 个邻居	简单，结果直观； 可人为修正学习结果； 新数据可直接加入数据集而不必重新进行训练； 可以处理非线性问题	样本不平衡时，预测偏差比较大； 需要占用大量内存； 对 K 值的选择比较敏感
多层感知机回归	通过多个感知机构建多层前馈神经网络，通过误差逆传播算法计算每层的权重	具有不错的容错能力，对离群点不敏感； 可处理高维非线性问题； 模型精度高	模型依赖众多超参数； 不能观察并修正学习过程，可解释性最差； 训练耗时； 极易过拟合
Boosting 类回归	多个弱分类器通过投票或加权的方法组成一个强分类器	不易过拟合； 可以处理非线性问题； 对特征选择结果无强依赖； 模型精度高	对离群点敏感； 训练耗时

表 3 各类回归模型对比

（二） 模型超参数选择

机器学习中，超参数系在学习开始之前需要为模型设置值的参数。与之对应，其他参数的值是通过训练得出的。超参数配置不同，学习器的性能往往会

有显著区别。实际应用中，Scikit-Learn 中各模型默认的超参数组合（详见下表）往往并不适合训练集的数据模式，需要进行优化。

模型	普通参数举例	超参数举例
岭回归	各项的系数和截距	L2 范数系数
Boosting 类回归	各弱分类器的权重	弱分类器种类，弱分类器个数，学习速率等
多层感知机回归	每层的权重	隐藏层数量，每层神经元数量，训练的 epoch 等
K 近邻回归		近邻 K 的选择，距离函数的选择，初始化选择等

表 4 各模型普通参数和超参数对比

参数网格搜索是一项超参数优化技术，常用于三个及以下超参数的优化，其本质属穷举法范畴。对于每个超参数，使用者创建一个较小的有限集合，作为该超参数的备选项。然后，从各项超参数备选项的笛卡尔乘积中得到若干组不同的超参数组合。网格搜索使用每组超参数训练模型，挑选验证集误差最小的超参数组合作为模型最好的超参数组合。

由于样本容量较小，按照常规 70%的训练集，20%验证集和 10%测试集的样本划分方法，从数据集中取出 20%的验证集对超参数进行选择的话，会存在训练集样本容量和验证集样本容量都不够大的问题。而通过交叉验证的方法对训练集上的数据进行循环使用，可以使预测模型在训练集的多个而非单个子数据集上实现优异表现，增强预测模型的泛化能力。由此，笔者选择了 5 折交叉验证后各预测模型在验证集上 R^2 的平均值对模型进行评价，并作为该模型的基础性能。

K 折交叉验证（K-fold cross validation）的核心思想如图 8 所示，把训练数据 D 分为 K 份，其中（K-1）份用于训练模型，剩余 1 份用于评估预测模型的准确率。前述过程在 K 份数据依次循环，最终得到 K 个评估结果。[3]

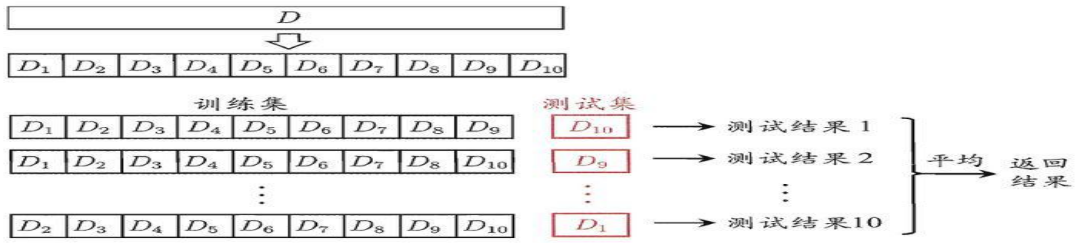

图 8 10 折交叉验证

通过如下代码构建基础的学习器：

[3] 周志华，机器学习. 清华大学出版社，2016.

```
#线性回归
from sklearn import linear_model
model_LinearRegression = linear_model.LinearRegression()

# 岭回归
from sklearn.linear_model import RidgeCV
model_RidgeCV= RidgeCV()

#MLPRegressor神经网络回归
from sklearn.neural_network import MLPRegressor
model_MLPRegressor=MLPRegressor(random_state=23)

#Adaboost回归
from sklearn import ensemble
model_AdaBoostRegressor = ensemble.AdaBoostRegressor(n_estimators=50,random_state=23)#这里使用50个决策树

#LGBM回归
import lightgbm as lgb
model_LGBMRegressor= lgb.LGBMRegressor()

#XGBoost回归
import xgboost as xgb
model_XGBRegressor = xgb.XGBRegressor()

#KNN回归
from sklearn import neighbors
model_KNeighborsRegressor = neighbors.KNeighborsRegressor()
```

图 9 利用 Scikit-Learn 生成学习器

之后，通过下述代码对每个学习器中模型的超参数予以调优。

对MLP神经网络进行调参

默认只有一层(200,)的效果很差。调参时神经网络的必要步骤。

```
parameter_space = {
    'hidden_layer_sizes': [(5,5,5),(3,3,3),(3,3,3,3)],
    'activation': ['tanh', 'relu','logistic'],
    'solver': ['sgd', 'adam','lbfgs'],
    'alpha': [0.0001, 0.0002,0.0005],
    'learning_rate': ['constant','adaptive'],
}

grid = GridSearchCV(MLPRegressor(random_state=23), parameter_space, cv=5)
grid.fit(train_X,train_y)
print("The best parameters are %s with a score of %0.2f" %(grid.best_params_, grid.best_score_))
```

```
The best parameters are {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (3, 3, 3), 'learning_rate': 'constant', 'solver': 'lbf
gs'} with a score of -0.05
```

```
model_MLPRegressor=MLPRegressor(activation='tanh', alpha=0.0001, hidden_layer_sizes=(3,3,3), learning_rate='constant', solver='lbfgs',random_s
```

```
score=[]
score=cross_val_score(model_MLPRegressor,train_X,train_y,cv=5,scoring='r2')
print(score)
score.mean()
```

```
[-0.0657214   0.01119044 -0.0106349  -0.23491783  0.03798471]
-0.05241979719935137
```

图 10 网格搜索最优超参数组合

```
In [13]: #导入交叉验证包
         from sklearn.model_selection import cross_val_score
```

线性回归

```
In [14]: #交叉验证 计算平均值
         score=[]
         score=cross_val_score(model_LinearRegression,train_X,train_y,cv=5,scoring='r2')
         print(score)
         score.mean()
```

```
[-4.31859267 -2.20635012 -1.73417002 -0.39247642 -0.18406846]
Out[14]: -1.7671315362442512
```

图 11 通过交叉验证评估模型的基础性能

多轮迭代后，各模型在超参数优化前后验证集上的 R^2 平均值如表 5 所示：

模型	超参数优化前 R^2	超参数优化后 R^2
Adaboost 回归	0.0789	0.1403
XGBOOST 回归	-0.2189	0.0603
LGBM 回归	-0.0217	0.1457
多层感知机回归	-61.7015	-0.0524
K 近邻回归	-0.1619	-0.0649

表 5 模型在原始测试集上调参前后对比

六、特征选择

（一）特征选择概述

数据集内的各属性对预测结果提供的信息增益各异。鉴于此，往往需通过特征选择，于给定的所有属性中选取相关属性作为样本的特征，去除掉无关和冗余属性，从而达到降低拟合风险，提高训练速度的目的。[4]

特征选择方法可分为过滤法、包裹法、嵌入法三类。

过滤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，因而最终准确率可能不高。此外，由于采取的是单变量统计检验手段，故未考虑特征间的相互关系。

包裹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的准确率，寻找最优的特征子集。包裹法因为有模型的直接参与，因而准确性较高，但是计算成本高，容易出现过拟合。

嵌入法利用了模型本身的特性，将特征选择嵌入到模型构建过程中。典型的如 Lasso 和树模型等。其准确率较高，计算复杂度介于过滤式和包裹式方法之间，但缺点是仅部分模型适用此方法。[5]

方法类别	方法举例
过滤法	皮尔逊相关系数法，方差分析法，Relief (Relevant Features)
包裹法	前向选择法，迭代剃除法，Las Vegas Wrapper
嵌入法	L1、L2 范数法，C4.5 剪枝算法

表 6 特征选择方法列举

（二）过滤法

过滤法最常用的方法是 SelectKBest（）。顾名思义，该方法就是根据传入的评分函数，从所有特征中挑选出最好的 K 个特征组成新的特征集。由于本次研究的问题属于回归问题范畴，因此选择了 f-regression 方法对各属性进行线性相关分析，并根据得到的 F 值计算出相应的 p 值。本次研究过程及结果如下：

```
#ANOVA测试
from sklearn.feature_selection import f_regression
from sklearn.feature_selection import SelectKBest

#保留一半，14个特征
k_best=SelectKBest(f_regression, k=14)

k_best.fit_transform(train_X, train_y);

p_values=pd.DataFrame({'Column':columns,'p_value':k_best.pvalues_}).sort_values('p_value')

p_values[p_values['p_value']<.05]
```

	Column	p_value
3	前三年承诺覆盖率	0.000220
6	成立年限	0.000546
14	净利润增长率	0.001216
18	固定资产周转率	0.002355
5	动态市盈率	0.004184
15	总资产周转率	0.040629
13	归母权益增长率	0.046219

```
train_new_data=train_data[["前三年承诺覆盖率", "成立年限", "净利润增长率", "固定资产周转率", "动态市盈率", "总资产周转率", "归母权益增长率"]]
train_new_X=scaler.fit_transform(train_new_data)
```

图 12 通过 f-regression 进行特征选择

属性名称	p_value
前三年承诺覆盖率	0.00022
成立年限	0.00055

[4] Ozdemir, S. Susarla, D. Feature engineering made easy. Birmingham, UK: Packt Publishing. 2018.
[5] A review of feature selection techniques in bioinformatics[J]. Bioinformatics(19):2507-2517.

净利润增长率	0.00122
固定资产周转率	0.00236
动态市盈率	0.00418
总资产周转率	0.04063
归母权益增长率	0.04622

表 7 各属性显著性分析

表格显示，前三年承诺覆盖率、成立年限两个指标对特有风险有非常显著的影响；净利润增长率、固定资产周转率、动态市盈率、总资产周转率、归母权益增长率对特有风险有显著影响。但是由于 F 检验属于线性回归测试，因此存在部分和特有风险呈非线性关系的特征未被选取的可能。

（三）递归式

递归特征消除法（RFE, Recursive feature elimination）是一种常用的包装法特征选择方法。其核心思想系通过不断地迭代训练模型，每次删除若干重要性较低的特征，直到最新删除特征造成总体性能损失时结束。

通过如下代码调用 RFECV 函数，对每种模型进行特征选择。过程如下：

```
In [94]:
from sklearn.feature_selection import RFECV
estimator =lgb.LGBMRegressor()
# 5折交叉
selector = RFECV(estimator, step=1, cv=5,scoring='r2')
selector = selector.fit(train_X, train_y)
# 输出特征入选最后特征，true表示入选
#print(selector.support_)
# 每个特征的得分排名，特征得分越低（1最好），表示特征越好
#print(selector.ranking_)
#  选选了几个特征
print(selector.n_features_)
16

In [95]:
selected_feature=[]
for i in range(len(selector.support_)):
    if selector.support_[i]:
        selected_feature.append(feature_dict[i])

In [96]:
feature_importances_value=pd.DataFrame({'Feature': selected_feature,'feature_importances':selector.estimator_.feature_importances_}).sort_value

In [97]:
feature_importances_value
Out[97]:
```

	Feature	feature_importances
7	D/(D+E)	26
9	归母权益增长率	25
13	资产负债率	24
3	动态市盈率	23
5	大股东持股比例	23
6	最近一个完整会计年度对第一供应商的采购占比	23
11	总资产周转率	23
15	毛利率	23
1	承诺期业绩增长率	22
2	前三年承诺覆盖率	21
8	总资产增长率	18
0	上一年归母净利润（万元）	17
10	净利润增长率	17
14	净资产收益率	16
4	成立年限	14
12	存货周转率	11

图 13 通过 RFECV 进行特征选择

```
In [98]:
train_new_data=train_data[selected_feature]
train_new_X=scaler.fit_transform(train_new_data)

In [99]:
from sklearn.model_selection import GridSearchCV
parameter_space = {
    'max_depth': [1,4,6],
    'num_leaves': [3,5,8],
    'min_child_samples': [15,18,19],
    'min_child_weight': [0.001,0.002],
    'cat_smooth': [0,10]
}

grid = GridSearchCV(lgb.LGBMRegressor(), parameter_space, cv=5)
grid.fit(train_new_X, train_y)
print("The best parameters are %s with a score of %0.2f" %(grid.best_params_, grid.best_score_))
The best parameters are {'cat_smooth': 0, 'max_depth': 4, 'min_child_samples': 18, 'min_child_weight': 0.001, 'num_leaves': 5} with a score
of 0.20

In [100]:
score=[]
model_LGBM=lgb.LGBMRegressor(cat_smooth=0, max_depth=4, min_child_samples=18, min_child_weight=0.001, num_leaves=5)
score=cross_val_score(model_LGBM,train_new_X,train_y,cv=5,scoring='r2')
score.mean()
Out[100]: 0.2040889447046296
```

图 14 根据特征选择结果重新训练模型

（四）结果对比

对各模型通过 5 折交叉验证法，生成的预测模型在原始数据验证集及通过不同特征选择方法生成的验证集上的 R^2 的平均值如表 8 所示：

模型	原始数据集	过滤法	RFECV
线性回归	-1.7671	0.1288	-0.2222
岭回归	-0.0921	0.1552	0.1834
AdaBoost回归	0.1403	0.161	0.1747
XGBOOST回归	0.0603	0.0718	0.1311
LGBM回归	0.1457	0.0791	0.2088
多层感知机回归	-0.0524	0.2182	\
K近邻回归	-0.0649	0.1872	\

表 8 预测模型在不同特征方法上的表现

由上表显示，除了多层感知机回归和 K 近邻回归没有权值系数属性（coef 或 feature_importances）而无法进行迭代外，递归特征消除法相比过滤法在验证集上能有更好的表现。

七、预测模型评估与分析

（一）预测模型评估

根据上述特征选择和超参数调优的结果，笔者选出各模型在 5 折交叉验证中具有最高 R^2 的特征及超参数组合，在之前分离的 90%的训练集上根据选出的特征生成新的训练数据，利用超参数组合输入到模型中，得到最终的预测模型。

```
In  [75]: train_X, test_X, train_y, test_y = train_test_split(new_X, y,  test_size = 0.1,  random_state=2)

In  [76]: model_MLPRegressor=MLPRegressor(activation= 'relu',  alpha=0.0001,  hidden_layer_sizes= (5,  5,  5,  5),  learning_rate='constant',  solver= 'lbf

In  [77]: model_MLPRegressor.fit(train_X,train_y)
          result =model_MLPRegressor.predict(test_X)
          result_frame=pd.DataFrame({'true_value': test_y,'predict_value':result})
          result_frame

Out[77]:
```

	true_value	predict_value
92	0.005	0.017163
36	0.010	0.021354
12	0.050	0.054342
3	0.010	0.024101
30	0.010	0.024555
24	0.010	0.020562
64	0.030	0.024713
61	0.010	0.021012
44	0.028	0.025157
71	0.015	0.015404
2	0.045	0.029481
112	0.030	0.025052
28	0.025	0.025338

图 15 根据训练集训练数据，并对测试集上的数据进行预测

将之前分离出的 10%的数据集输入到预测模型中，对预测值和实际值进行对比，得到如下折线图：

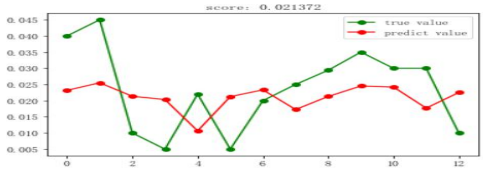

图 16-1：线性回归

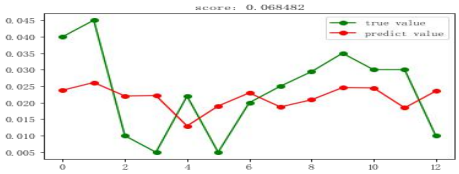

图 16-2：岭回归

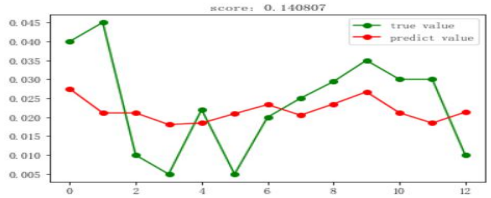

图 16-3：AdaBoost 回归

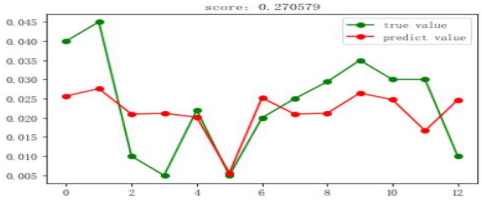

图 16-4：多层感知机回归

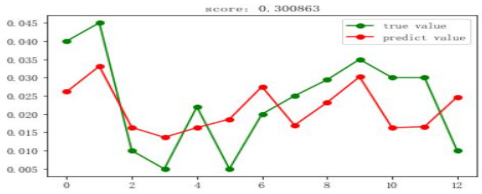

图 16-5：XGBoost 回归

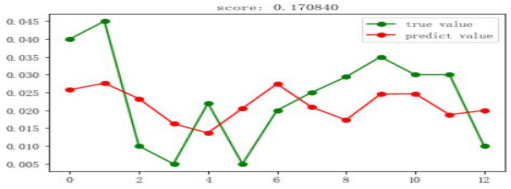

图 16-6：LGBM 回归

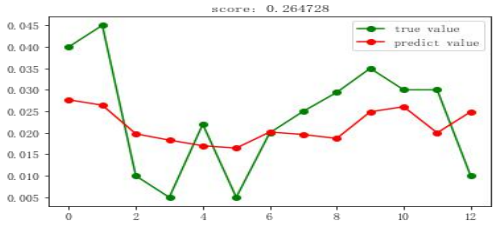

图 16-7：K 近邻回归

对比预测模型在 5 折交叉验证及在测试集上的 R^2，具体见下表：

预测模型	验证集表现	测试集表现
XGBOOST 回归	0.1311	0.3008
多层感知机回归	0.2182	0.2706
K 近邻回归	0.1872	0.2647
LGBM 回归	0.2088	0.1708
Adaboost 回归	0.1863	0.1408
岭回归	0.1834	0.0685
线性回归	0.1288	0.0214

表 9 预测模型在验证集和测试集表现

其中，每个模型选用的特征如下表所示：

模型	选用的特征
XGBOOST 回归	未进行特征选择
多层感知机回归	前三年承诺覆盖率、成立年限、净利润增长率、固定资产周转率、动态市盈率、总资产周转率、归母权益增长率
K 近邻回归	前三年承诺覆盖率、动态市盈率、成立年限、净利润增长率、固定资产周转率
LGBM 回归	承诺期业绩增长率、总资产增长率、净资产收益率、大股东持股比例、D/(D+E)、动态市盈率、上年归母净利润（万元）、前三年承诺覆盖率、净利润增长率、毛利率、成立年限、归母权益增长率、资产负债率
Adaboost 回归	前三年承诺覆盖率、动态市盈率、成立年限、净利润增长率、固定资产周转率
线性回归	前三年承诺覆盖率、成立年限、净利润增长率、固定资产周转率、动态市盈率、总资产周转率、归母权益增长率
岭回归	前三年承诺覆盖率、成立年限、净利润增长、固定资产周转率、动态市盈率、总资产周转率、归母权益增长率

表 10 各模型选用的特征

（二）模型结果与分析

从结果上看，模型在训练集和测试集上的多折交叉验证表现有一定差别。通过测试集确定模型的参数，验证集确定模型的超参数后，笔者通过独立的测试集来评估预测模型的最终性能，以决定预测模型的选择结果。

通过配对样本的 t 检验，笔者对不同预测模型的均方误差（MSE）是否存在显著差异进行了检验。通过检验发现，线性回归预测模型和岭回归预测模型的 MSE 相比 XGBoost 预测模型有显著差异；AdaBoost 预测模型和 LGBM 预测模型的 MSE 相比 XGBoost 预测模型有一定差异，但并不显著；K 近邻回归预测模型和多层感知机预测模型 MSE 相比 XGBoost 预测模型几乎不存在差异。

由于模型的复杂度差异，非线性模型预测准确率相对更高。XGBoost 回归模型具有最好的表现，但距离对特有风险进行精准定量测算的初衷仍有差距。

此外，应关注到，对于 XGBoost 回归和多层感知机回归模型而言，我们无法给出明确的解析解来说明预测值生成的原因。当部分预测值存在偏差时，我们也只能通过有倾向性地输入新的训练集来纠正偏差。而 K 近邻回归模型则可以通过构建 KD 树来明确每个样本的范围，并输出相应用于预测的 K 个最近邻居，使得模型具有更强的解释性。因此，就实践而言，K 近邻回归具有更大的应用价值，表现也更优秀。

在特征选择上，总体来看，各模型使用的频度较高的特征为前三年承诺覆盖率、成立年限、净利润增长率、动态市盈率、固定资产周转率、归母权益增长率等，且该等特征的影响程度逐项递减。

笔者认为，这些特征对于特有风险的确定，确实存在很强的因果关联。

一是，通常来说，前三年承诺覆盖率、净利润增长率、动态市盈率、归母权益增长率越高，则企业特有风险越大。当企业处于高速增长阶段之时，表明其需要更多的资源予以支持，很可能在人员、技术、产能及营运资金等方面存在大量缺口；高速增长期间，企业面临的自身文化、组织结构、外部认同及管理者能力不足问题尤为突出，资金流、人力资源、营销部门及管理能力等往往相对薄弱。另外，不排除部分企业为凸显高速增长的市场形象，进行不恰当的会计估计、会计政策改变甚至财务造假的可能，易造成后续风险的集中爆发。净利润增长率、归母权益增长率为历史静态性特征，而前三年承诺覆盖率、动态市盈率属于预测期动态性特征，均属于增长率范畴，由前文可知，该等指标与特别风险成正比，且动态特征影响力大于静态特征。

二是成立年限越长，企业特有风险越小。企业成立年限越长，表明其极有可能占领市场先机，掌握更为充足的原材料、技术、渠道等关键资源，通过多年的经营和多轮优胜劣汰，拥有更丰富的经验，具备一定的竞争优势，赢得长期的市场优势。另外，企业经营多年，也表明其所处行业存续时间较久，产业普遍较为成熟，行业的不确定性较小。

三是固定资产周转率越高，则企业特有风险越大。这个结论与增长率结论相似。高周转的企业往往处于某一个爆发式发展阶段，但难以长期维持，就像一台高速运转的机器、一根紧绷的弦，需要外部资源的不断支持。而纵观国内

外市场和企业发展历程，从中、长期而言，良性的发展大部分是细火慢炖的，符合市场整体发展趋势的周转率更为适宜。

鉴于机器学习方法生成的预测模型往往具有“黑盒”特征，因此，本研究未能生成定量公式。但我们可以将相关数据输入预测模型得出结论，随着数据数量与质量的不断提升，预测数据将更为准确。

八、总结与展望

本文通过回归分析和递归特征消除法，建立了一套涵盖 6 个主要指标的特有风险评价体系，并在体系基础上利用机器学习方法对特有风险进行了预测。

几种预测模型的表现总体差强人意，R^2 均不甚理想，笔者认为原因有以下几点：一是本次研究的数据来源于过往实践，鉴于实务中评估专业人员过度依赖主观判断致结果偏差，甚至根据结果导向确定特别风险，因此实证数据质量欠佳；二是样本容量相对较小，无法完全满足机器学习对数据规模的要求。

笔者曾采用传统统计学路径研究了同样的样本数据，生成了回归预测模型，认为资产负债率、研发支出占比、归母权益增长率、净利润增长率、总资产周转率、应收账款周转率及上一年归母净利润与特有风险呈正相关关系；经营性现金流/收入、成立年限呈负相关关系。前次与本次研究结果有一定的共同点，均认为特别风险与归母权益增长率、净利润增长率呈正比，与成立年限成反比，且两次研究分别提及的总资产周转率与固定资产周转率有共通之处。但笔者以为，本研究中多个模型都一致认可了前三年承诺覆盖率、动态市盈率的重要性，这与并购重组定价逻辑及博弈重点不谋而合，具有合理性。从定性角度来看，本次研究结果更具温度。此外，前次研究的拟合优度 R^2（分别为 0.210 和 0.189）亦较低（略低于本次研究），两次研究均表明自变量对因变量的解释力度不足。

虽然在现有实证数据基础上，特别风险的准确厘定较难实现。但对数据进行特征工程并构建非线性模型的思路，具有一定的借鉴价值。

首先，由于机器学习模型的复杂性，一方面其在处理海量、多属性的数据集时具有不错表现，另一方面也会产生黑盒的可解释性问题。因此，可以从获取数据的数量与质量，以及结论的可解释性需求两个角度综合分评估是否要引入机器学习。一般而言，当可获取的数据多且全面，能对预测结果给出充分反馈，且无需做出完整解时，则适合引入机器学习方法。

其次，在具体模型的选择上，虽然机器学习构建的非线性模型的预测准确率往往好于线性模型，但没有一个模型能在所有问题上都优于其他模型。如果两模型表现接近，那么选择相对简单的线性回归模型，不失为明智的选择。

再次，需要关注的是，机器学习方法的实质是对变量和因变量的相关性进行数理分析并得出答案。其分析结果只能说明因变量和变量间存在一定的相关性，并不能说明两者间是否存在因果关系。因此，在机器学习的同时，不可忽

视专业知识的重要性。

最后，对于本次研究，还可以在特征选择方法上做进一步探索。由于特征选择不是具有贪心选择性质的组合最优化问题，无法在多项式时间内直接计算得到最优解。因此除了通过用如递归特征消除法这种贪心算法得到近似解外，也可以考虑采取诸如模拟退火、遗传算法、蚁群算法等启发式算法予以优化。

The company-specific risk prediction method research based on machine learning

Yue Zhou[1],Zhengyi Wang[2]

（CanWin Appraisal Co.,Ltd.,Hangzhou,310007[1];

Xiangcai Securities Co., Ltd., Shanghai, 200120[2])

Abstract：The company-specific risk is the key and difficult point of corporate value appraisal, especially the income capitalization approach. Based on historical practical cases, this paper extracts quantifiable data from the public information of A-share capital market major mergers and acquisitions, and uses feature engineering technology to screen and enhance these data as input features. After tuning the hyperparameters by the grid search method, the authors selected multiple learners to carry out machine learning on the data corresponding to the aforementioned characteristics, and derive a unique risk prediction model. Through practice, it is found that the feature selection method based on the recursive feature elimination algorithm can effectively improve the accuracy of the prediction model. It is hoped that the "specific risk prediction" will be used as a case study to demonstrate the operation process of machine learning and provide a new attempt for the construction of evaluation models.

Keywords：Appraisal; Income capitalization approach; Company-specific risk; Machine learning.