基于集成学习的上市公司高转送预测研究

摘要

在当前的市场背景下，股市的不确定性直接影响了众多股民和上市公司的收益，若能提前预知可能实施高转送的上市公司并在合适的时机购入则会有一个不错的收益。

首先我们对数据集进行了初步的探索，了解了三个数据集的基本内容，明确了数据集之间的联系。接着我们利用 Python 和 Spark 大数据平台对数据依次进行清洗，降噪，填补缺失值，处理异常值，降维，特征选择，从而得到一个规整的高质量数据集。

对于问题 1 我们首先将处理好的数据按照 7:3 划分成训练集和测试集，进行归一化处理后，依次将数据输入逻辑回归、决策树、KNN、随机森林、极度随机树、XGBoost 模型进行训练，再利用 AUC 指标进行模型评估并调整模型参数使结果达到最优。再用训练效果较好的模型计算每个特征的重要性权值选出重要性权值最大的前 30 个特征，再合并不同模型计算出来的结果，得到对实施高转送影响最大的 18 个特征因子。

结果是，营业收入同比增长(%)、高转送预案公告日_3、投资支出/折旧和摊销、最低价、最高价、收盘价、上市年限、总资产净利率、息税折旧摊销前利润/负债合计、基本每股收益、稀释每股收益、每股资本公积、每股收益(期末摊薄)、基本每股收益同比增长、总资产相对年初增长、最高价下半年变异系数、收盘价上半年变异系数、收盘价下半年变异系数。

对于问题 2，我们评估了六个模型后使用了 Stacking 集成学习的融合模型，将六个单独的模型整合到一起，其中逻辑回归、决策树、KNN、随机森林、极度随机树模型作为第一层基分类器，XGBoost 模型作为第二层分类器。Stacking 集成模型的预测效果要好于其他 6 种机器学习算法，而且 Stacking 集成学习融合模型的 AUC 达到了 83.32%，远比其他六种模型中的任意一种要好。因此我们将 Stacking 集成学习模型作为我们问题二的预测模型。我们预测出第八年有 10.27%的上市公司选择实施高送转，即有 356 个上市公司进行高送转，3110 家公司不进行高送转。

关键词：集成学习 特征工程 数据处理 重要性权值 高送转预测

1. 绪论

1.1 问题背景

近年来，我国证券市场的高速发展催生了一批题材股，根据重大事件的不同分类，可以分为资产重组板块、粤港澳板块、新能源板块等等。在这些题材中间，高送转这一题材无疑是中小投资者强烈追捧的对象。因为实施高送转后股价将做除权处理，投资者可以通过填权行情从二级市场的股票增值中获利。很多股票在公布派送预案的第二天直接涨停，而等除权后再买入可能会面临很大的回撤风险。如果我们能准确预测下一年可能实施高送转的上市公司并提前买入，这对我们投资的安全性具有很大的现实意义。

经过研究，影响上市公司实施高送转的因子主要有两类：一是基本因子，包括股价、总股本、上市年限等；二是成长因子，包括每股未分配利润、每股资本公积、每股现金流、每股收益等。除此之外，还有“未来 6 个月是否存在解禁”、“是否存在定增方案”等因子需要挖掘。

1.2 问题重述

问题1：针对附件给出的因子数据，根据经济学意义以及数理统计方法，筛选出对上市公司实施高送转方案有较大影响的因子。

问题2：利用问题1中确定的因子，建立模型来预测哪些上市公司可能会实施高送转，并对附件提供的数据，用所建立的模型来预测第8年上市公司实施高送转的情况。

1.3 我们的工作和方法

本文利用现有的股市数据集进行模型建立，预测出第八年实施高送转的上市公司。我们的完成的工作如下

1. 对数据进行初步探索，了解数据的结构和特征分布情况。
2. 对数据进行预处理，填补缺失值，消除噪声，对特征进行抽取完成数据降维减少计算提高模型效率和准确度。

3. 建立多种机器学习经典算法模型，评估不同模型的优劣，选出优良的模型来选择对上市公司实施高送转影响较大的因子。
4. 利用网格搜索进行调参，使用筛选过后的特征来预测第八年可能实施高转送的上市公司。

1.4 模型假设

1. 预测年分内不会出现影响股市出现剧烈波动的突发性偶然事件。
2. 所有数据都有实际参考价值
3. 假设股市的样本服从独立同分布

1.5 符号说明

表 1 符号说明表

符号	说明
w	激活函数权重
b	偏置值
x	样本数据
θ	权重和偏置值的替代矩阵

2. 理论介绍

2.1 概念介绍

高送转是指送股或者转增股票比例较大，一般 10 送或转 5 以上的才算高送转。举个例子，在牛市中某股 10 元一股，年报时 10 股送 10 股，而股价拦腰一半成了 5 元，视觉上价格较低容易吸引新股民跟风，就好像大白菜一块一公斤人们嫌贵，而五毛一斤人们就觉得便宜了的一样的无意义心理。填权后庄家获利丰厚，真实的复权价格已到天上，但跟风的并不感觉价格高。

在公司“高送转”方案的实施日，公司股价将作除权处理，也就是说，尽管“高送转”方案使得投资者手中的股票数量增加了，但股价也将进行相应的调整，投

资者持股比例不变，持有股票的总价值也未发生变化。“高送转”的实质是股东权益的内部结构调整，对净资产收益率没有影响，对公司的盈利能力也并没有任何实质性影响。而且，在净利润不变的情况下，由于股本扩大，资本公积金转增股本与送红股将摊薄每股收益。

实行高转送的公司，一方面表明公司对业绩的持续增长充满信心，公司正处于快速成长期，有助于保持良好的市场形象；

另一方面一些股价高、股票流动性较差的公司，也可以通过“高送转”降低股价，增强公司股票的流动性。

2.2 机器学习算法模型

2.2.1 逻辑回归模型

逻辑回归假设数据服从伯努利分布，通过极大化似然函数方法，运用梯度下降来求解参数，来达到将数据二分目的。

算法推导：

对数几率函数是一种 Sigmoid 函数，通过此函数来输出类别概率。

对数几率函数为：

$$y=\frac{1}{1+e^{-(w^T x+b)}} \tag{1}$$

其中 y 代表的是样本视为正样本的可能性，则$1-y$视为负样本的可能性。

对数几率定义为

$$\ln\frac{y}{1-y}=w^T x+b \tag{2}$$

其中$\frac{y}{1-y}$称为比率。

决策边界：作用在 n 维空间，将不同样本分开的平面或曲面，在逻辑回归中，决策边界对应

$$w^T x+b=0 \tag{3}$$

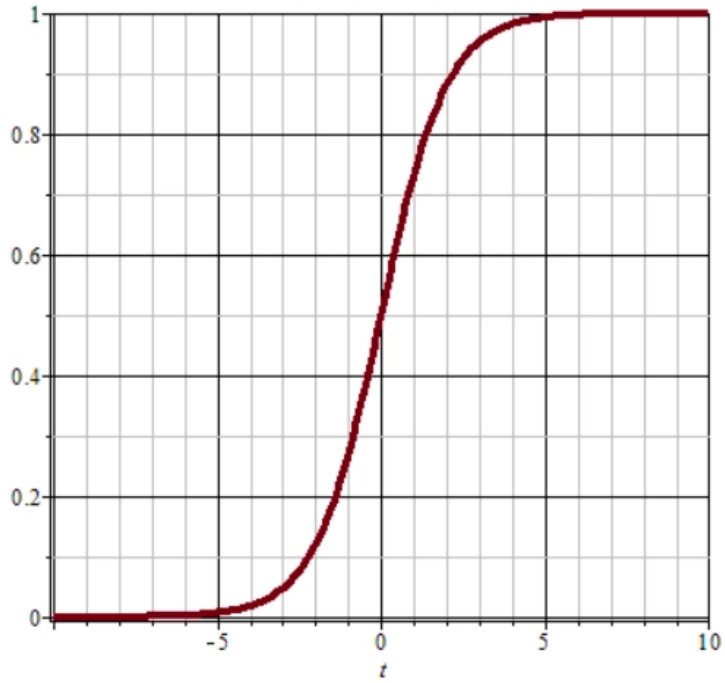

图 1　逻辑回归分布函数

使用极大似然法进行参数估计。由对数几率函数对应输出正样本的概率，可得对应关系：

$$P(\mathrm{y}=1\,|\,x)=\frac{e^{w^{T}x}+b}{1+e^{w^{T}x+b}}, \tag{4}$$

并令 $P(\mathrm{y}=1\,|\,x)=\pi(x)$，对应的

$$P(\mathrm{y}=0\,|\,x)=\frac{1}{1+e^{w^{T}x+b}} \tag{5}$$

则

$$P(\mathrm{y}=0\,|\,x)=1-\pi(x) \tag{6}$$

学习目标是对参数和 b 进行参数估计，使得逻辑回归模型能尽可能符合数据集分布。对于给定的数据集 $\{x^{(i)},y^{(i)}\}$ 其中 i 从 1 到数据集大小 m，来使得最大化对数似然。

首先，写出似然函数：

$$l=\prod_{i=1}^{m}[\pi x^{(i)}]^{y^{(i)}}[1-\pi x^{(i)}]^{1-y^{(i)}} \tag{7}$$

对数似然函数就是

$$L(w)=\sum_{i=1}^{m}[y^{(i)}\log\pi x^{(i)}+(1-y^{(i)})\log(1-\pi x^{(i)})] \tag{8}$$

化简得到

$$\sum_{i=1}^{m}[y^{(i)}\log\frac{\pi x^{(i)}}{1-\pi x^{(i)}}+\log(1-\pi x^{(i)})]=\sum_{i=1}^{m}[y^{(i)}(w\cdot x^{(i)})-\log(1+e^{w\cdot x^{(i)}})] \tag{9}$$

后面式子是带入 $\pi(x)$ 后化简得到。现在，即对对数似然函数求极大值，以对数似然函数为目标的最优化问题。$L(w)$ 是关于 w 的高阶连续可导凸函数，根据凸优化理论，可采用梯度下降法，牛顿法等优化方法求解。逻辑回归的损失函数是交叉熵损失函数，交叉熵主要用于度量分布的差异性。

令

$$h_\theta(x)=\frac{1}{1+e^{-\theta^T x}} \tag{10}$$

即用 θ 代替了 $[b,w_0,w_1,...,w_n]$。

交叉熵损失函数：

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log h_\theta(x)+(1-y^{(i)})\log(1-h_\theta(x))] \tag{11}$$

使用交叉熵损失函数的原因：$J(\theta)$ 中去掉 $-\frac{1}{m}$ 便是上述的对数似然函数，对 $J(\theta)$ 求最小，即对对数似然函数求极大。

逻辑回归的梯度下降

$$\frac{\partial J(\theta)}{\partial\theta_i}=\sum_{i=1}^{m}\frac{1}{m}(\frac{e^{\theta^T x^{(i)}}}{1+e^{-\theta^T x}}x_j^{(i)}-y^{(i)}x_j^{(i)})=\sum_{i=1}^{m}\frac{1}{m}(h_\theta x^{(i)}-y^{(i)})x^{(i)},(j=0,1,2,...,n) \tag{12}$$

所以，参数迭代更新式为：

$$\theta_j=\theta_j-\alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta x^{(i)}-y^{(i)})x_j^{(i)},(j=0,1,2,...,n) \tag{13}$$

普通逻辑回归是一个二分类模型，可推广至多分类。

2.2.2 决策树模型

决策树是通过一系列规则对数据进行分类的过程。它提供一种在什么条件下

会得到什么值的类似规则的方法。决策树分为分类树和回归树两种，分类树对离散变量做决策树，回归树对连续变量做决策树。

实际上，样本所有特征中有一些特征在分类时起到决定性作用，决策树的构造过程就是找到这些具有决定性作用的特征，根据其决定性程度来构造一个倒立的树--决定性作用最大的那个特征作为根节点，然后递归找到各分支下子数据集中次大的决定性特征，直至子数据集中所有数据都属于同一类。所以，构造决策树的过程本质上就是根据数据特征将数据集分类的递归过程。

信息、熵以及信息增益是决策树的根本，是决策树利用特征来分类时，确定特征选取顺序的依据。

引用香农的话来说，“信息是用来消除随机不确定性的东西”。对于机器学习中的决策树而言，如果带分类的事物集合可以划分为多个类别当中，则某个类(x_i)的信息可以定义如下:

$$l(X = x_i) = -\log_2 p(x_i) \qquad (14)$$

$l(x)$用来表示随机变量信息，$p(x_i)$指是当x_i发生时的概率。熵是用来度量不确定性的。当熵越大，$X = x_i$的不确定性越大，反之越小对于机器学习中的分类问题而言，熵越大即这个类别的不确定性更大，反之越小。

信息增益在决策树算法中是用来选择特征的指标，信息增益越大，则这个特征的选择性越好。

一棵决策树的生成过程主要分为以下 3 个部分:

特征选择: 特征选择是指从训练数据中众多的特征中选择一个特征作为当前节点的分裂标准，如何选择特征有着很多不同量化评估标准，从而衍生出不同的决策树算法。

决策树生成： 根据选择的特征评估标准，从上至下递归地生成子节点，直到数据集不可分则停止决策树停止生长。 树结构来说，递归结构是最容易理解的方式。

剪枝： 决策树容易过拟合，一般来需要剪枝，缩小树结构规模、缓解过拟合。剪枝技术有预剪枝和后剪枝两种。

2.2.3 K 近邻模型

KNN（K-Nearest Neighbor）法即 K 最邻近法，是一种分类算法。该方法的思路非常简单直观：如果一个样本在特征空间中的 K 个最相似（即特征空间中最邻近）的样本中的大多数属于某一个类别，则该样本也属于这个类别。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别

总体来说，KNN 分类算法包括以下 4 个步骤：

①准备数据，对数据进行预处理。

②计算测试样本点（也就是待分类点）到其他每个样本点的距离。

③对每个距离进行排序，然后选择出距离最小的 K 个点。

④对 K 个点所属的类别进行比较，根据少数服从多数的原则，将测试样本点归入在 K 个点中占比最高的那一类。

2.2.4 随机森林模型

随机森林就是通过集成学习的思想将多棵树集成的一种算法，它的基本单元是决策树，而它的本质属于机器学习的一大分支——集成学习（Ensemble Learning）方法。随机森林本质是是由成百上千棵树组成。这也是随机森林的主要思想--集成思想的体现。

其实从直观角度来解释，每棵决策树都是一个分类器（假设现在针对的是分类问题），那么对于一个输入样本，N 棵树会有 N 个分类结果。而随机森林集成了所有的分类投票结果，将投票次数最多的类别指定为最终的输出，这就是一种最简单的 Bagging 思想。

随机森林以如下的理论为基础。

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测试，每个分支代表一个测试输出，每个叶节点代表一种类别。

集成学习

集成学习通过建立几个模型组合的来解决单一预测问题。它的工作原理是生

成多个分类器/模型，各自独立地学习和做出预测。这些预测最后结合成单预测，因此优于任何一个单分类的做出预测。

随机森林是集成学习的一个子类，它依靠于决策树的投票选择来决定最后的分类结果。

随机森林的生成

随机森林中有许多的分类树。我们要将一个输入样本进行分类，我们需要将输入样本输入到每棵树中进行分类。在对一个样本进行分类时每棵树都要独立地进行计算得出一个结果，最终该样本属于哪一个类别需要要依据每一棵树的计算结果来确定，计算结果最多的类别就是森林的分类结果。森林中的每棵树都是独立的，99.9%不相关的树做出的预测结果涵盖所有的情况，这些预测结果将会彼此抵消。少数优秀的树的预测结果会脱颖而出，做出一个好的预测。将若干个弱分类器的分类结果进行投票选择，从而组成一个强分类器，这就是随机森林 bagging 的思想(关于 bagging 的一个有必要提及的问题：bagging 的代价是不用单棵决策树来做预测，具体哪个变量起到重要作用变得未知，所以 bagging 改进了预测准确率但损失了解释性。)下图可以形象地描述这个情况：

图 2 随机森林示意图

有了树我们就可以分类了,每棵树按照如下规则生成：

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训练样本（这种采样方式称为 Bootstrap sample 方法），作为该树的训练集；

从这里我们可以知道：每棵树的训练集都是不同的，而且里面包含重复的训练样本。

对训练集还要进行随机抽样，如果不进行随机抽样，每棵树的训练集都一样，那么最终训练出的树分类结果也是完全一样的，这样的话完全没有 Bagging 的必要；

随机抽样还要是有放回的抽样，如果不是有放回的抽样，那么每棵树的训练样本都是不同的，都是没有交集的，这样每棵树都是"有偏的"，都是绝对"片面的"（当然这样说可能不对），也就是说每棵树训练出来都是有很大的差异的；而随机森林最后分类取决于多棵树（弱分类器）的投票表决，这种表决应该是"求同"，因此使用完全不同的训练集来训练每棵树这样对最终分类结果是没有帮助的，这样无异于是"盲人摸象"。

如果每个样本的特征维度为M，指定一个常数$m << M$，随机地从M个特征中选取m个特征子集，每次树进行分裂时，从这m个特征中选择最优的；

每棵树都尽最大程度的生长，并且没有剪枝过程。

一开始我们提到的随机森林中的“随机”就是指的这里的两个随机性。两个随机性的引入对随机森林的分类性能至关重要。由于它们的引入，使得随机森林不容易陷入过拟合，并且具有很好得抗噪能力（比如：对缺省值不敏感）。

随机森林分类效果（错误率）与两个因素有关：

1. 森林中任意两棵树的相关性：相关性越大，错误率越大；
2. 森林中每棵树的分类能力：每棵树的分类能力越强，整个森林的错误率越低。

减小特征选择个数 m，树的相关性和分类能力也会相应的降低；增大 m，两者也会随之增大。所以关键问题是如何选择最优的 m（或者是范围），这也是随机森林唯一的一个参数。

2.2.5 极度随机树模型

Extra-Trees（Extremely randomized trees，极端随机树）算法与随机森林算法十分相似，都是由许多决策树构成。极限树与随机森林的主要区别：

1、RandomForest 应用的是 Bagging 模型，ExtraTree 使用的所有的样本，只是特征是随机选取的，因为分裂是随机的，所以在某种程度上比随机森林得到的结果更加好

2、随机森林是在一个随机子集内得到最佳分叉属性，而 ET 是完全随机的得到分叉值，从而实现对决策树进行分叉的。

对于第 2 点的不同，我们再做详细的介绍。我们仅以二叉树为例，当特征属性是类别的形式时，随机选择具有某些类别的样本为左分支，而把具有其他类别的样本作为右分支；当特征属性是数值的形式时，随机选择一个处于该特征属性的最大值和最小值之间的任意数，当样本的该特征属性值大于该值时，作为左分支，当小于该值时，作为右分支。这样就实现了在该特征属性下把样本随机分配到两个分支上的目的。然后计算此时的分叉值（如果特征属性是类别的形式，可以应用基尼指数；如果特征属性是数值的形式，可以应用均方误差）。遍历节点内的所有特征属性，按上述方法得到所有特征属性的分叉值，我们选择分叉值最大的那种形式实现对该节点的分叉。从上面的介绍可以看出，这种方法比随机森林的随机性更强。

对于某棵决策树，由于它的最佳分叉属性是随机选择的，因此用它的预测结果往往是不准确的，但多棵决策树组合在一起，就可以达到很好的预测效果。

当 ET 构建好了以后，我们也可以应用全部的训练样本来得到该 ET 的预测误差。这是因为尽管构建决策树和预测应用的是同一个训练样本集，但由于最佳分叉属性是随机选择的，所以我们仍然会得到完全不同的预测结果，用该预测结果就可以与样本的真实响应值比较，从而得到预测误差。如果与随机森林相类比的话，在 ET 中，全部训练样本都是 OOB 样本，所以计算 ET 的预测误差，也就是计算这个 OOB 误差。

2.2.6 XGBoost

XGBoost 为“Extreme Gradient Boosting”的缩写它是一个 Boosting 集成算法，它的主要思路是将成百上千个树模型组合起来成为一个准确率很高的模型，此模型通过不断迭代生成新的树。XGBoost 我们常用于监督学习，即建立一个数据模型，输入相关特征从而预测出目标，而这一过程，需要我们找到训练数据最好的

参数，所以我们需要定义一个目标函数，通常由训练损失（Traning loss）和正则项（Regularization term）组成。训练损失评估了预测模型的效果，正则项则是控制着模型的复杂度，避免模型不被过度拟合。这两个互相博弈（tradeoff）的指标保证了模型的预测效果以及简洁程度。

目标函数 $L(x)$ 由误差函数 $F\left(x\right)$ 和复杂度函数 $\Omega\left(x\right)$ 组成:

$$L(x)=F(x)+\Omega(x) \tag{15}$$

$$L(x)=\sum_{i} l(y_i,\hat{y}_i)+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}\|W_j\|^2 \tag{16}$$

其中 l 是用来衡量 $\hat{y}$ 与 y 的相近程度的可导且凸的损失函数，通过每一步增加一个基分类器，贪婪地去优化目标函数，使得每次增加都使得损失变小。然后让后一次迭代的基分类器去学习前一次遗留下来的误差。这样可以得到用于评价当前分类器性能的评价函数,如下：

$$L_m(x)=\sum_{i=1} l[y_i,\hat{y}_i^{(m-1)}+f_m(x_i)]+\Omega(f_m) \tag{17}$$

这个算法又可以成为前向分步优化。为了更好更快的优化此函数，可以在 $f_m=0$ 附近二阶泰勒展开，泰勒展开的形式为公式

$$f(x+\Delta x)=f(x)+f^{'}(x)\Delta x+\frac{1}{2}f^{''}(x)\Delta x^2 \tag{18}$$

令

$$g_i=\frac{\partial l(y_i,\hat{y}^{(t-1)})}{\partial \hat{y}^{(t-1)}},h_i=\frac{\partial^2 l(y_i,\hat{y}^{(t-1)})}{\partial(\hat{y}^{(t-1)})^2} \tag{19}$$

最后可得到目标函数，在剔除常数项后可以得到最终的表达式，如公式所示：

$$L_m(x)=\sum_{i=1}^{n}[g_i+f_m(x_i)+\frac{1}{2}h_i f_m^{\,2}(x_i)]+\Omega(f_m) \tag{20}$$

3. 数据预处理及特征选择

3.1 数据探索

我们首先利用 Python 读取了基础数据、日数据、年数据，对每一个数据集的数据内容有了基本的认识，基础数据有 4 个特征，分别是股票编号、上市年限、所属行业、所属概念板块，该数据集的 3466 条样本对应 3466 个上市公司的相关信息。

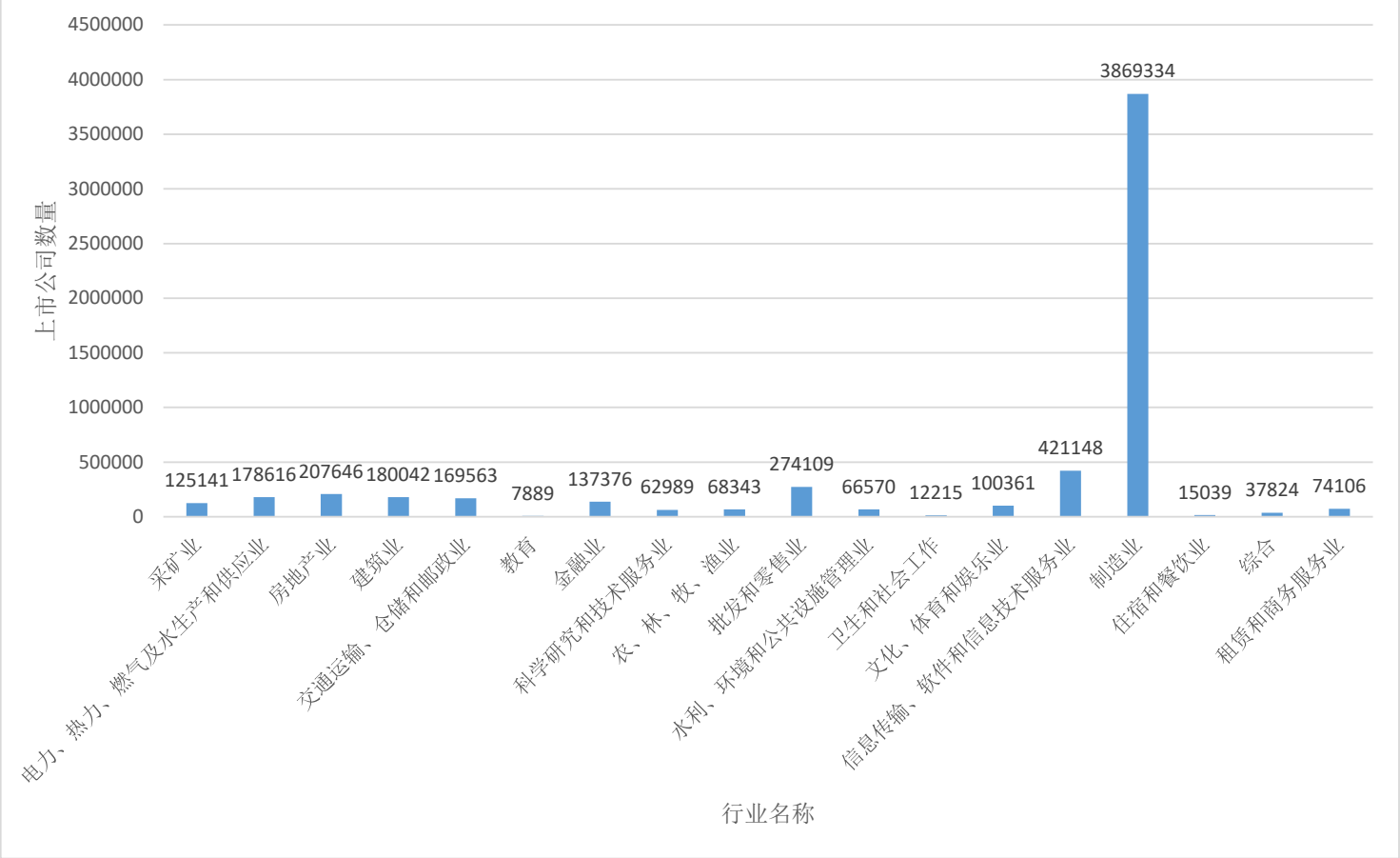

图 3 各行业上市公司数量

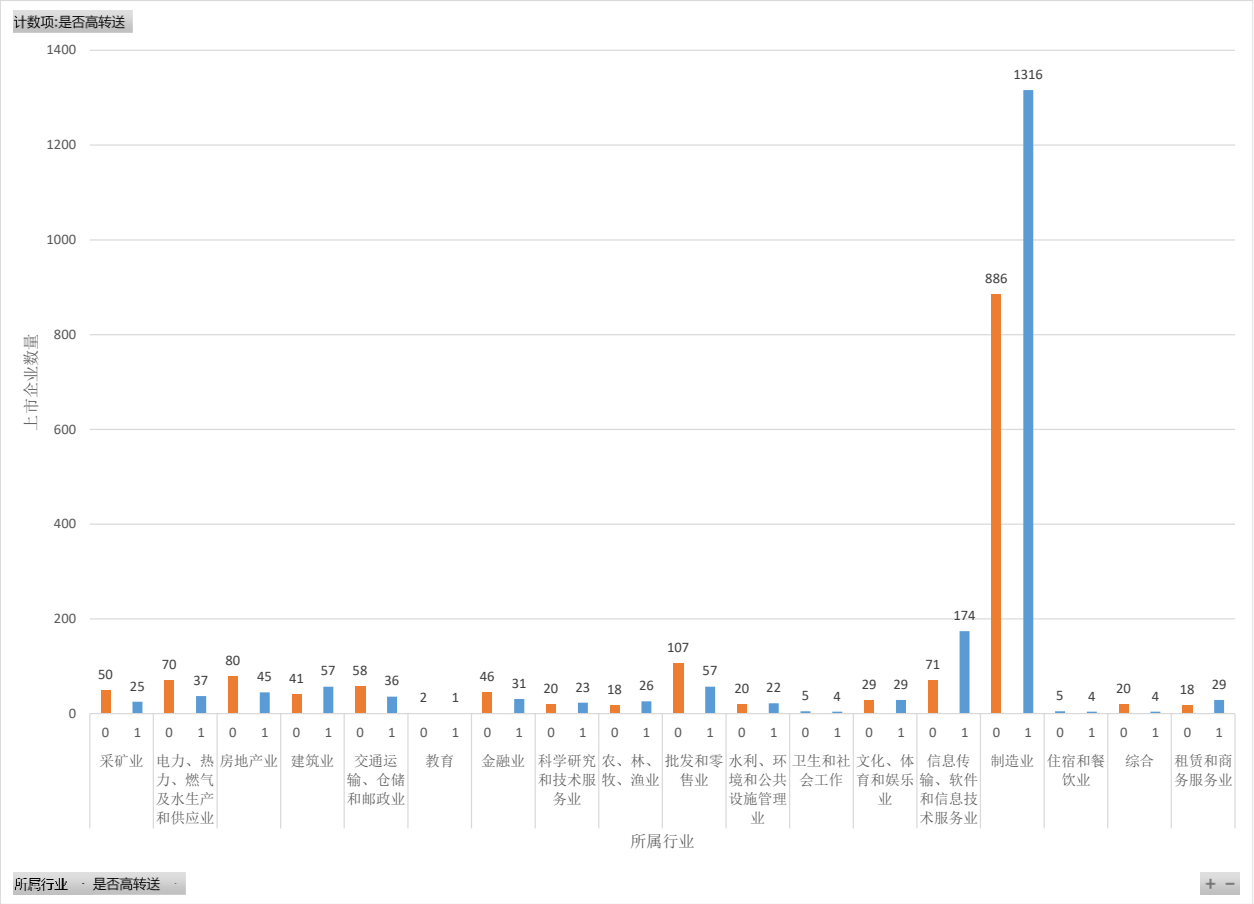

图 4 各行业实施高转送上市公司数量图

如上图 3 展示了数据集中各行业的上市公司数量，图 4 展示了每个行业的实施高转送上市公司的数量分布情况。可以明显看出制造业的上市公司数量远远多于其他行业，说明原始数据集的上市公司行业分布不均匀，因此上市公司所属行业在本文的研究中参考意义不大。

年数据共有 362 个特征，其中是否高转送可以作为数据标签，样本数量 24262。年数据是三个数据集的核心，其他两个数据集都需要合并到年数据中，作为我们预测第八年实施高转送上市公司的依据。

日数据有 61 个特征，没有特征可以作为标签，样本数量为 5899132。三个数据集都包含了股票编号，因此我们利用股票编号来将三个数据集外连接，由于数据量过于庞大，所以我们利用 Spark 读取流式数据。并将该问题转化成二分类问题来解决。我们直接将标签“是否高转送”作为因变量，再通过特征工程来寻找适量的特征作自变量。

3.2 数据处理

3.2.1 填补缺失值

因为我们在后面的问题求解过程中需要利用现有模型进行预测，采用机器学习相关的算法模型是不能让数据中存在缺失值的，而原始数据很脏存在大量的缺失值，所以我们需要采用数据填充的方式来填补缺失值。

数值型特征我们采用插值填补法，遍历所有的特征，利用现有的数据进行多项式拟合，将存在缺失值的样本的其他有值的特征代入拟合多项式中算出拟合值作为填充数。

类别型特征我们采用众数填补法，我们直接调用 Python 中 Pandas 库的 mean 函数来找出每一个数值特征的众数，然后再将众数作为填充值填入缺失处，若存在一个特征有多个众数的情况则随机选取一个众数作为填充数。

3.2.2 数据归一化

把数据变成（-1,1）之间的小数[1]。主要是为了数据处理方便提出来的，把数据映射到 0～1 范围之内处理，更加便捷快速。

把有量纲表达式变成无量纲表达式，便于不同单位或量级的指标能够进行比较和加权。归一化是一种简化计算的方式，即将有量纲的表达式，经过变换，化为无量纲的表达式，成为纯量。

归一化实质是一种线性变换，线性变换有很多良好的性质，这些性质决定了对数据改变后不会造成“失效”，反而能提高数据的表现，这些性质是归一化的前提。

3.3 特征选择

特征选择(Feature Selection)也称特征子集选择(Feature Subset Selection , FSS)，或属性选择(Attribute Selection)。是指从已有的 M 个特征(Feature)中选择 N 个特征使得系统的特定指标最优化，是从原始特征中选择出一些最有效特征以降低数据集维度的过程,是提高学习算法性能的一个重要手段,也是模式识别中

关键的数据预处理步骤。对于一个学习算法来说,好的学习样本是训练模型的关键。

特征选择的经验总结有三种方法：Fliter、Wrapper、Embedded。

Filter 故名思议就是过滤式方法，对过滤之后的特征进行训练，特征选择的过程与后序学习器无关。其评估手段是判断单维特征与目标变量之间的关系，常用的手段包括 Pearson 相关系数、Gini 系数、信息增益、方差校验和相似度度量等。

Wrapper 即封装法，可理解为将特征选择的过程与分类器封装在一起，以一个分类器的交叉验证结果，作为特征子集是否优秀的评估方式。而特征子集的产生方式分为前向搜索与后向搜索两种方法，在实现上，常见的方法有稳定性选择(Stability Selection)和递归特征消除(Recursive Feature Elimination)两种方式。

Embedded 可以理解为嵌入法，即使用分类器本身的特性进行特征选择。这种方法速度快，也易出效果，但需较深厚的先验知识调节模型。

在数据探索阶段我们浏览了所有的特征，结合相关的金融背景知识以及 Wrapper 的方法进行特征选择。

4. 问题一

4.1 模型建立

4.1.1 样本划分

从前面的数据探索阶段我们就将三个数据集的内容做了分析，可绘制出表 2 如下所示。

表 2 数据集规模表

	特征数量	样本数量
年数据	362	24262
日数据	61	5899132
基础数据	4	3466

我们将三个数据集以“股票编号”为主键进行左外连接形成一份新的数据集，

再利用 Python 第三方库 Sklearn 中的数据划分函数 train_test_split 来将新的数据集，按照 7:3 换分成训练集和测试集两部分。

4.1.2 模型评价

我们一共使用了六种经典的机器学习算法模型，分别是逻辑回归、决策树、K 近邻、随机森林、极度随机树
模型、XGBoost。在机器学习算法的模型评估指标中，精确率、召回率、F1-Score、AUC 等经常被作为评价指标。

精确率是针对我们预测结果而言的，它表示的是预测为正的样本中有多少是真正的正样本。那么预测为正就有两种可能了，一种就是把正类预测为正类(TP)，另一种就是把负类预测为正类(FP)，也就是

$$P = \frac{TP}{TP + FP} \tag{21}$$

召回率是针对我们原来的样本而言的，它表示的是样本中的正例有多少被预测正确了。那也有两种可能，一种是把原来的正类预测成正类(TP)，另一种就是把原来的正类预测为负类(FN)。

$$R = \frac{TP}{TP + FN} \tag{22}$$

F1 分数（F1-Score），是统计学中用来衡量二分类模型精确度的一种指标。它同时兼顾了分类模型的精确率和召回率。F1 分数可以看作是模型精确率和召回率的一种调和平均，它的最大值是 1，最小值是 0。

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{23}$$

ROC 的全称是 Receiver Operating Characteristic Curve，中文名字叫“受试者工作特征曲线”，顾名思义，其主要的分析方法就是画出特征曲线。该曲线的横坐标为假阳性率（False Positive Rate, FPR），N 是真实负样本的个数，FP 是 N 个负样本中被分类器预测为正样本的个数。纵坐标为真阳性率（True Positive Rate, TPR），

$$TPR = \frac{TP}{P} \tag{24}$$

$$FPR = \frac{FP}{F} \tag{25}$$

P 是真实正样本的个数，TP 是 P 个正样本中被分类器预测为正样本的个数。在本文的研究中我们使用 AUC 作为模型的评估指标。

4.2 模型参数优化与模型特征

4.2.1 参数优化概念及方法

模型参数优化是通过极小化目标函数使得模型输出和实际观测数据之间达到最佳的拟合程度，由于环境模型本身的复杂性，常规优化算法难以达到参数空间上的全局最优。近年来，随着计算机运算效率的快速提高，直接优化方法得到了进一步开发与广泛应用。

在模型调参中常用的三种方法分别是贪心算法、网格调参、贝叶斯调参。

贪心算法是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，它所做出的仅仅是在某种意义上的局部最优解。

贪心算法没有固定的算法框架，算法设计的关键是贪心策略的选择。必须注意的是，贪心算法不是对所有问题都能得到整体最优解，选择的贪心策略必须具备无后效性（即某个状态以后的过程不会影响以前的状态，只与当前状态有关。）

网格调参通过循环遍历，尝试每一种参数组合，返回最好的得分值的参数组合。每个参数都能组合在一起，循环过程就像是在网格中遍历，所以叫网格搜索。

贝叶斯优化通过基于目标函数的过去评估结果建立替代函数（概率模型），来找到最小化目标函数的值。贝叶斯方法与随机或网格搜索的不同之处在于，它在尝试下一组超参数时，会参考之前的评估结果，因此可以省去很多无用功。

超参数的评估代价很大，因为它要求使用待评估的超参数训练一遍模型，而许多深度学习模型动则几个小时几天才能完成训练，并评估模型，因此耗费巨大。贝叶斯调参发使用不断更新的概率模型，通过推断过去的结果来“集中”有希望的超参数。

我们选用网格调参方法来调整集成学习模型中的各个参数。

4.2.2 模型参数调优

我们在建立每一个模型的时候都自行设定了一个初始值，以便于让模型可以

正常的进行训练，之后再设定一个调整参数的范围进行网格搜索和交叉验证找到最优结果下的参数值。我们评价模型的指标为曲线下面积[2]。

逻辑回归模型的调参结果如下表 3 所示。

表 3 逻辑回归模型调参结果表

参数名称	初始值	调参范围	调参结果	调参后 AUC
penalty	*l*1	*l*1、*l*2	*l*1	80.57%
C	0.5	[1,4]	4	81.23%

逻辑回归模型的 *penalty* 表示正则化的方式，设定其初始值为 *l*1；*C* 表示正则化强度的倒数，设定其初始值为 0.5；从 评价指标 AUC 的结果来看，逻辑回归模型的效果不佳。

表 4 决策树模型调参结果表

参数名称	初始值	调参范围	调参结果	调参后 AUC
max_ *depth*	5	8~50	28	80.62%
min_ *samples _ leaf*	1	5~10	10	81.18%

决策树模型的 max_ *depth* 表示树的最大深度，设定其初始值为 5；min_ *samples _ leaf* 表示每个叶子结点包含的最小样本数，设定其初始值为 1；从评价指标 AUC 的结果来看，决策树模型的训练效果不佳。

表 5 KNN 模型调参结果表

参数名称	初始值	调参范围	调参结果	调参后 AUC
n _ neighbors	5	6~20	15	81.02%
weights	uniform	uniform、distance	distance	81.06%

KNN 模型的 *n _ neighbors* 表示选择最近的几个点，其初始值默认为 5；*weights* 表示距离权重，默认值为 uniform；从评价指标 AUC 的结果来看，KNN 模型的训练效果不佳。

表 6 随机森林模型调参结果表

参数名称	初始值	调参范围	调参结果	调参后 AUC
n_estinators	300	200~800	700	81.9%
max_*depth*	5	6~15	10	82.1%
min_*samples_split*	50	10~100	60	82.2%
min_*samples_leaf*	10	10~100	10	82.2%

随机森林模型的*n_estinators*表示随机森林中树模型的数量，设定其初始值为 300；max_*depth*表示树模型的最大深度，设定其初始值为 5；min_*samples_split*表示中间节点要分枝所需要的最小样本数，设定其初始值为 50；min_*samples_leaf*表示每个叶子结点包含的最小样本数，设定其初始值为 10。从评价指标 AUC 的结果来看，决策树模型的训练效果较好。

极度随机树模型的调参过程类似决策树模型的调参过程，从调参之后的 AUC 值来看极度随机树模型的训练效果不佳，此处就不再赘述。

表 7 XGBoost 模型调参结果表

参数名称	初始值	调参范围	调参结果	调参后 AUC
n_estinators	600	700~850	720	81.2%
Min_child_weight	1	1~5	2	81.3%
Max_depth	5	4~10	8	81.4%
gamma	0.5	0.1~1	0.6	81.5%
subsample	0.8	0.5~1	0.8	81.5%
Colsample_bytree	0.8	0.5~1	0.8	81.5%
Reg_alpha	1	0~3	0	82.1%
Reg_lambda	1	0.01~2	1	82.1%
learning_rate	0.1	0.001~0.2	0.01	82.5%

XGBoost 模型的*n_estinators*越大，模型的学习能力也就越强，由于该模型

的参数较多，仅介绍几个最为重要的模型。 *subsample* 表示随机抽样的时候抽取的样本比例，范围是(0,1]，参数 *learning_rate* 表示集成模型中的学习率，又称为步长以控制迭代速率，常常用于防止过拟合。默认值是 0.1，取值范围(0,1]。从调参结果来看，其 AUC 值再多次调参后有明显的提升，是六种机器学习模型中效果最好的。

4.3 对高转送影响较大的特征

为了得出对上市公司实施高转送影响较大的特征，我们挑选了几训练效果较好的模型，利用这些模型分别计算每个特征的重要性权值。

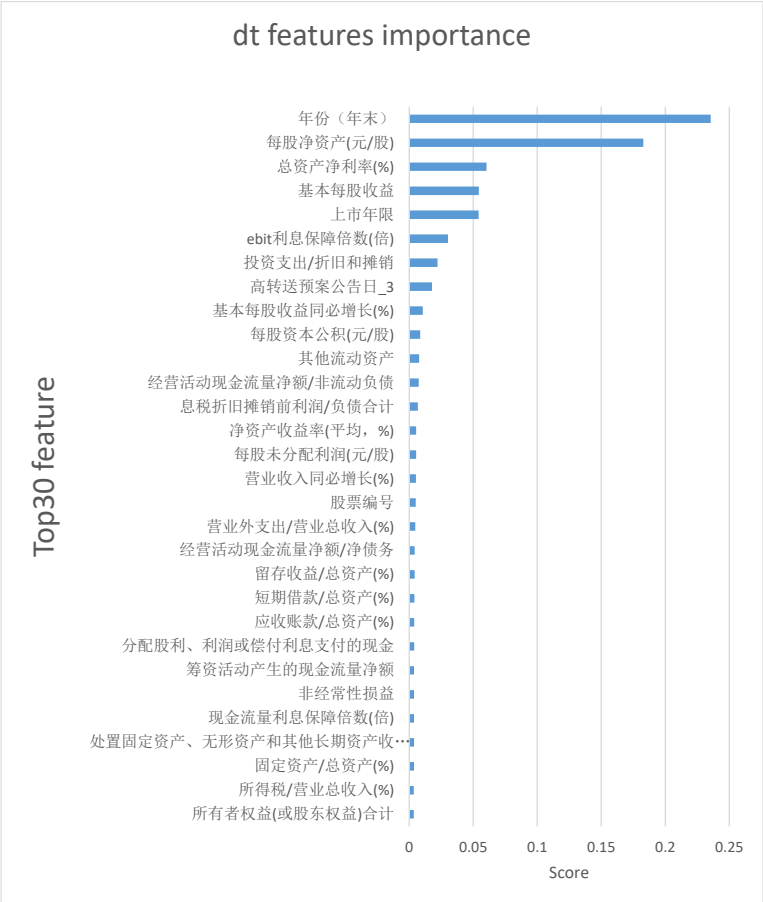

图 5 决策树模型特征抽取结果

图 5 是决策树模型计算得到的重要性权值最大的前 15 个特征，其中最重要的特征是“年份(年末)”。

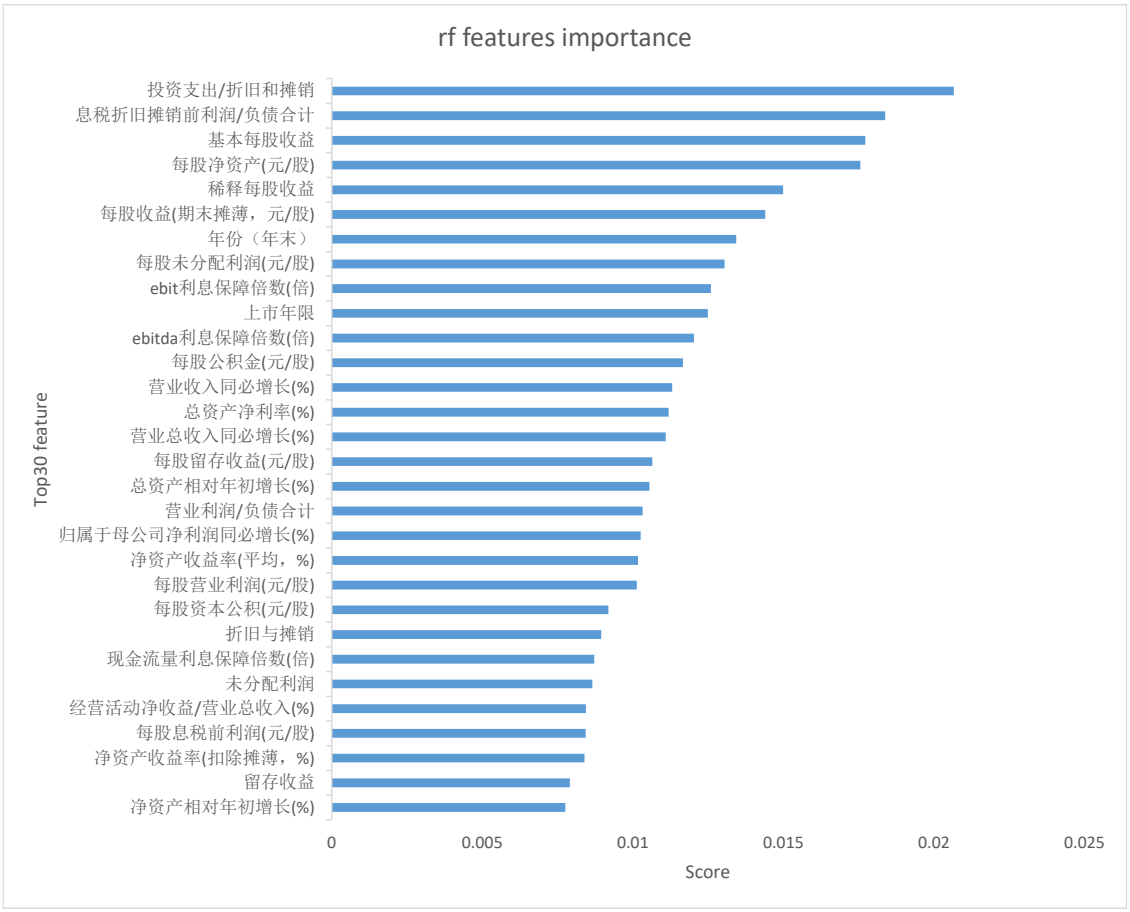

图 6 随机森林模型特征抽取结果

图 6 是随机森林模型计算得到的重要性权值最大的前 15 个特征，其中最重要的特征是“投资支出/折旧和摊销”。

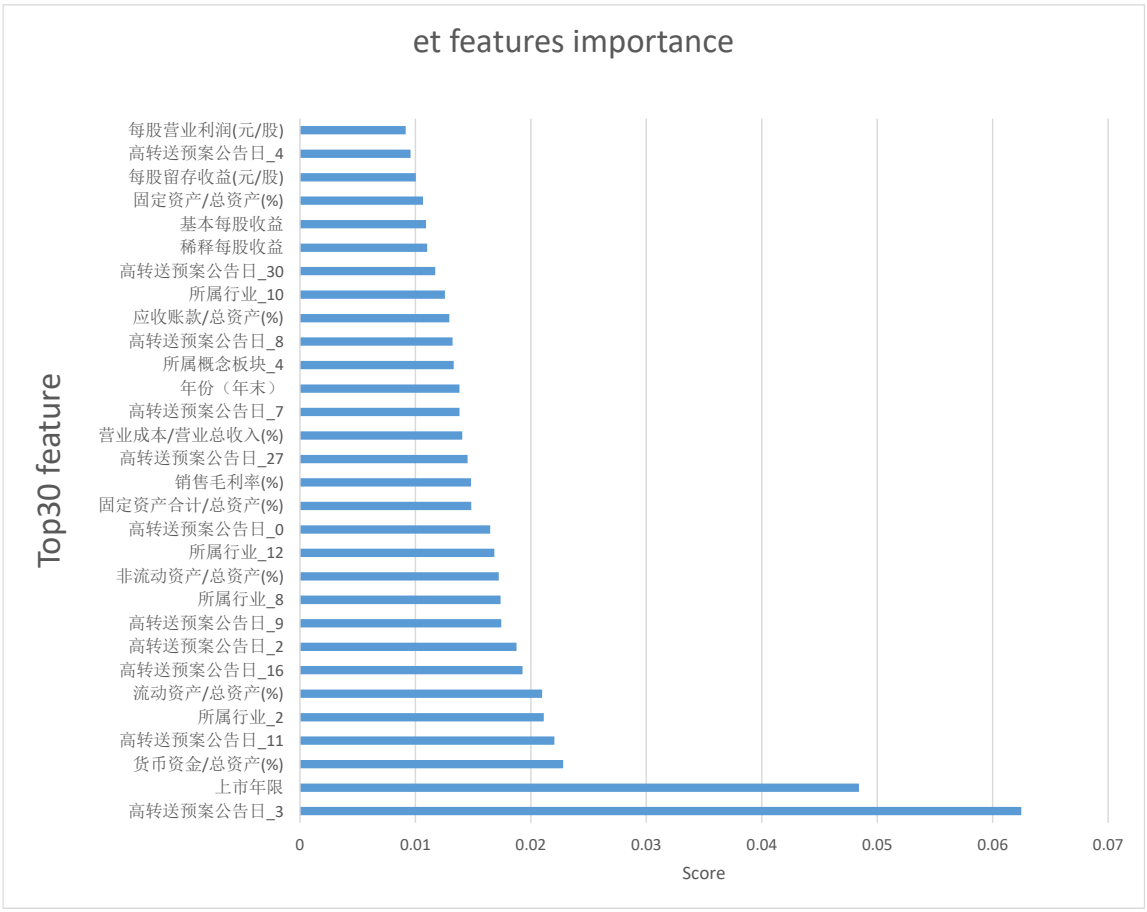

图 7 极度随机树模型特征抽取结果

图 7 是随机森林模型计算得到的重要性权值最大的前 15 个特征，其中最重要的特征是“高转送预案公告日_3”。

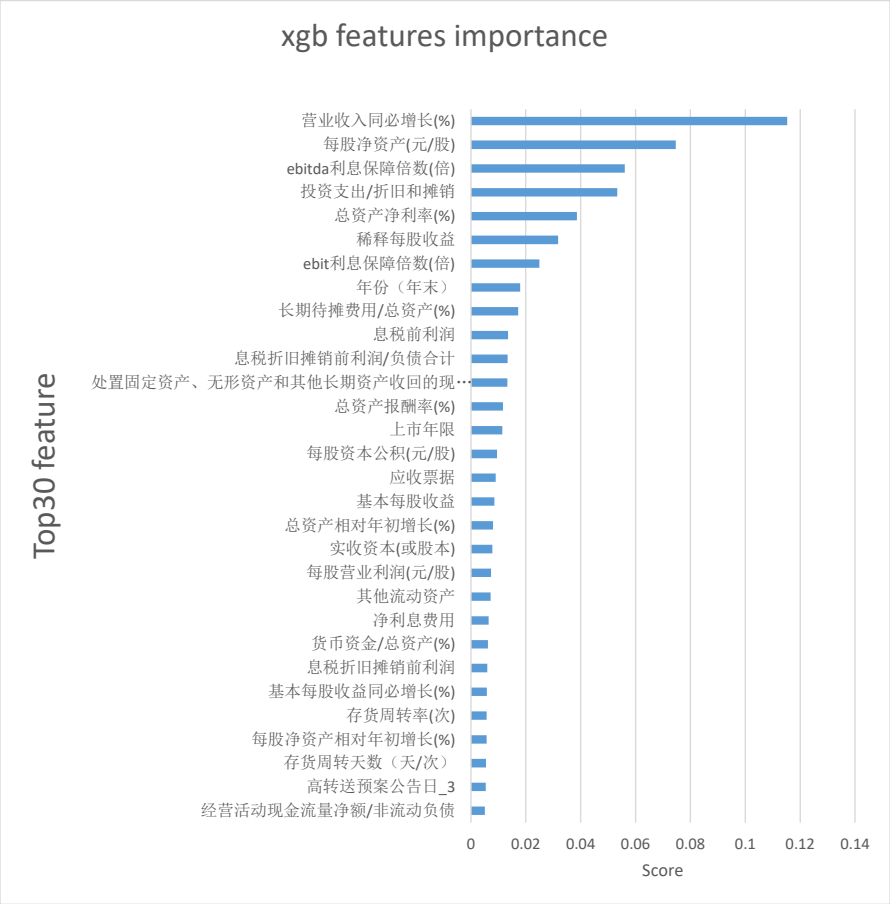

图 8 XGBoost 模型特征抽取结果

图 8 是随机森林模型计算得到的重要性权值最大的前 15 个特征，其中最重要的特征是“营业收入同比增长(%)”。

结合上面四种模型挑选出的特征我们汇总得到对上市公司实施高转送影响较大的特征因子结果如下：

营业收入同比增长(%)
高转送预案公告日_3
投资支出/折旧和摊销
最低价
最高价
收盘价
上市年限
总资产净利率
息税折旧摊销前利润/负债合计
基本每股收益

稀释每股收益
每股资本公积
每股收益(期末摊薄)
基本每股收益同比增长
总资产相对年初增长
最高价下半年变异系数
收盘价上半年变异系数
收盘价下半年变异系数

5. 问题二

5.1 模型选择

在前面对 6 种机器学习模型的评估中我们记录了每个模型调参的过程，总结可得出下表 8

表 8 各模型 AUC 变化对比表

算法	调参前	调参后
逻辑回归	79.92%	80.57%
决策树	80.79%	81.18%
KNN	81.03%	81.06
随机森林	81.82%	82.2%
极度随机树	81.13%	81.21%
XGBoost	81.27%	82.5%
Stacking	83.32%	

从前文 6 种机器学习模型来看单独进行训练的情况下 AUC 值最高也仅为 82.5%，并不算太突出。所以我们使用 Stacking。

Stacking[3]先从初始数据集训练出初级学习器,然后“生成”一个新数据集用于训练次级学习器。在这个新数据集中,初级学习器的输出被当作样例输入特征,而初始样本的标记仍被当作样例标记. Stacking 的算法描述如下表所示。这里我们假定初级学习器使用不同学习算法产生,即初级集成是异质的.

表 9 Stacking 算法

Stacking 算法
Input: 训练集 $D=\{(x_1,y_1),(x_1,y_1),...,(x_m,y_m)\}$;

初级学习算法 ε_1， $\varepsilon_1,...\varepsilon_1$;

次级学习算法 ε.

Process:

1: **for** $t=1,2,...,T$ **do**

2: $h_t=\varepsilon_t(D)$;

3: **end for**

4: $D^{'}=\Phi$;

5: **for** $i=1,2,...,m$ **do**

6: **for** $t=1,2,...,T$ **do**

7: $z_{it}=h_i(x_i)$;

8： **end for**

9： $D^{'}=D^{'}\cup((z_{i1},z_{i2},...,z_{iT}),y_i)$;

10: **end for**

11: $h_t=\varepsilon_t(D)$;

Output: $H(x)=h^{'}(h_1(x),h_2(x),...,h_\gamma(x))$

在训练阶段,次级训练集是利用初级学习器产生的,若直接用初级学习器的训练集来产生次级训练集，则过拟合风险会比较大;因此,一般是通过使用交叉验证或留一法这样的方式,用训练初级学习器未使用的样本来产生次级学习器的训练样本，以 k 折交叉验证为例,初始训练集 D 被随机划分为 k 个大小相似的集合 $D_1,D_2,...D_k$. 令 D_j 和 $\hat{D}_j=D\backslash D_j$ 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 $h_i^{(j)}$ 通过在 $\hat{D}_j$ 上使用第 t 个学习算法而得.对 D_j 中每个样本 x_i ,令 $z_{it}=h_i^{(j)}(x_i)$; ,则 x_i 由所产生的次级训练样例的示例部分为 $z_i=(z_{i1};z_{i2};...;z_{iT})$,标记部分为 y_i .于是,在整个交叉验证过程结束后,从这 T 个初级学习器产生的次级训练集是 $D^{'}=\{(z_i,y_i)\}_{i=1}^{m}$ 然后 $D^{'}$ 将用于训练次级学习器。

次级学习器的输入属性表示和次级学习算法对 Stacking 集成的泛化性能有很大影响.有研究表明,将初级学习器的输出类概率作为次级学习器的输入属性效

果更好。

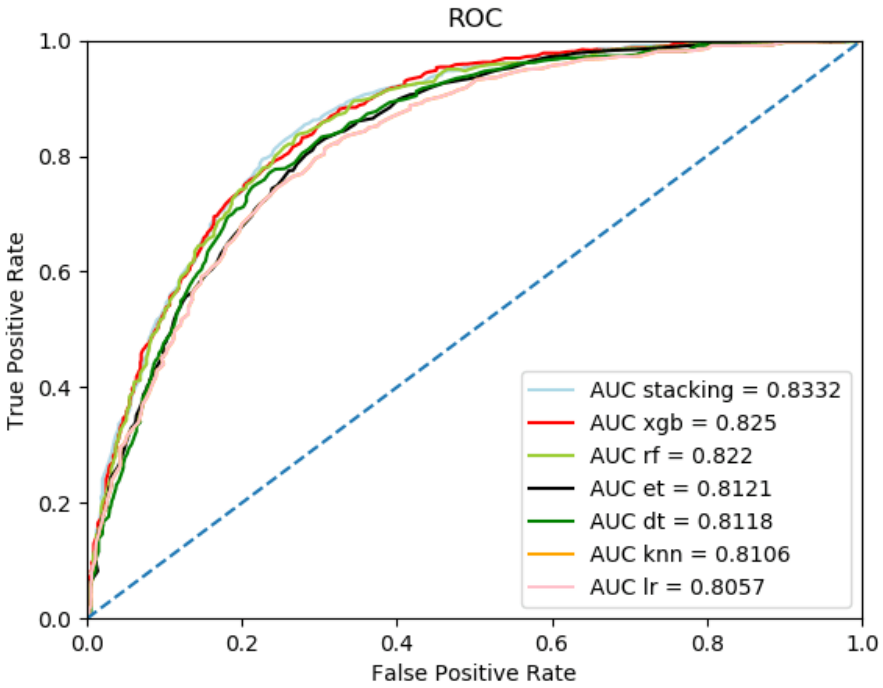

图 9 各种模型的 ROC 示意图

如图 9 所示，就是我们绘制的每个模型的特征曲线，其中 XGBoost 是 6 种模型种效果最好的，因此在将六种模型集成到一起形成一个新的模型时，把 XGBoost 模型作为第二层的模型，其余 5 个模型作为一底层的基学习器，通过整合六种模型得到的新的集成模型的 AUC 高达 83.32%。

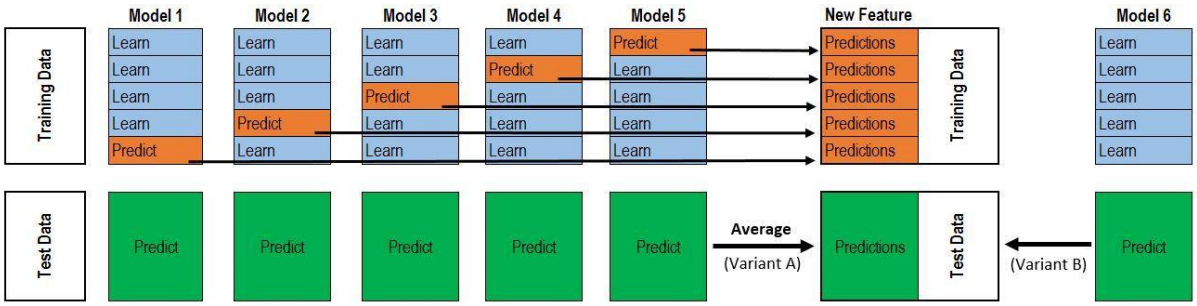

图 10 Stacking 模型融合过程图示

图 10 展示的是 Staking 融合六种经典的机器学习模型，这里分为了两层，第一层的基学习器主要是产生可供第二层元学习器使用的训练数据(train data)。

用一个基础模型进行 k 折交叉验证的结果。k 折交叉验证，就是先拿出 k-1 折作为训练数据，另外一折作为测试数据(testing data)。每一个交叉验证产生的预测结果组合起来，作为第 2 层模型的训练集数据。另外，还要对数据集原来

的整个训练数据进行预测，这个过程会生成 k 个预测数据集，对于这部分数据，本文将数据集各部分相加取平均值作为下一层模型的测试集数据。第 2 层学习模型采用非线性模型 XGBoost，通过将第 1 层模型输出的结果作为训练数据训练模型，得到新的预测结果。通过将新的预测结果和第 2 层模型的测试数据集进行对比，观察预测准确度。

5.2 上市公司高转送预测

我们将处理好的数据带入到建立好的 Stacking 集成学习融合模型中预测第八年中 3466 个上市公司实施高送转的数量。

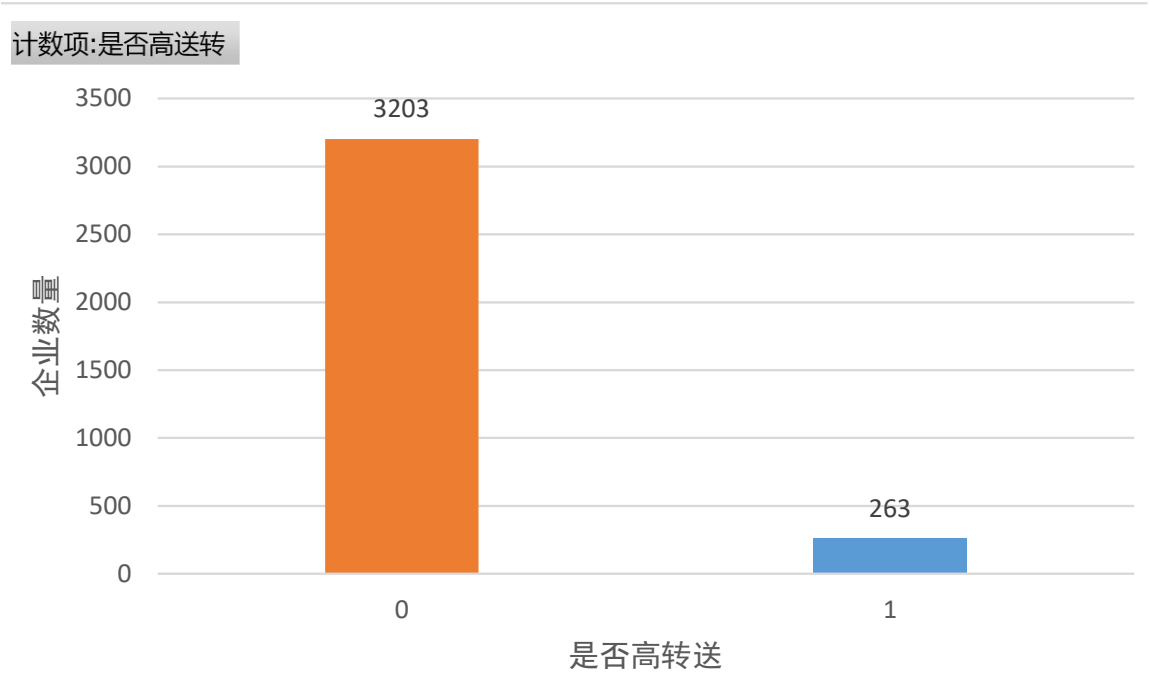

图 11 预测结果分布图

图 11 是我们绘制出的预测结果，0 表示不实施高送转，1 表示实施高送转。有 7.59%的上市公司将实施高送转，即有 263 家上市公司将实施高送转，3203 家上市公司不会实施高送转。

6. 总结

我们通过影响上市公司实施高转送的特征因子建立集成学习模型来预测第八年会实施高送转的上市公司数量。研究结论如下：

1. 我们对基础数据、日数据、年数据进行初步探索以后，以股票编号作为主键进行外连接，然后对数据进行处理得到规整的高质量数据，再后续的模型中预测效果优良，说明我们对于数据的合并和处理是正确可行的。
2. 在建立模型时，我们首先选用了 6 中经典的机器学习算法逻辑回归、决策树、KNN、随机森林、极度随机树、XGBoost，我们再结合 AUC 作为模型评估指标，发现 XGBoost 的预测效果最为突出，然后再用不同的模型计算了每个特征因子的重要性权值，找出最重要的 30 个特征，再汇总到一起最后得到 18 个影响上市公司实施高送转的重要特征因子。
3. 18 个特征因子：营业收入同比增长(%)、高转送预案公告日_3、投资支出/折旧和摊销、最低价、最高价、收盘价、上市年限、总资产净利率、息税折旧摊销前利润/负债合计、基本每股收益、稀释每股收益、每股资本公积、每股收益(期末摊薄)、基本每股收益同比增长、总资产相对年初增长、最高价下半年变异系数、收盘价上半年变异系数、收盘价下半年变异系数。
4. 在模型预测的选择上，我们并没有从 6 种基础的机器学习算法模型中挑选出一个，而是基于 Stacking 将逻辑回归、决策树、KNN、随机森林、极度随机树作为第一层基学习器，XGBoost 作为第二层元学习器，从而得到一个新的集成学习融合模型作为我们问题二的预测模型。Stacking 集成模型的预测效果要好于其他 6 种机器学习算法，AUC 值更是高达 83.32%。
5. 我们预测第八年上市公司实施高送转的情况。7.59%的上市公司将实施高送转，即 263 个上市公司决定高送转，3203 个上市公司不会高送转。

参考文献

[1]蔡景波,蔡志杰.基于机器学习模型预测 A 股市场高送转股票[J].数学建模及其应用,2020,9(04):74-84.
[2]胡宸. 基于集成学习的上市公司高送转预测模型及投资策略设计[D].上海师范大学,2019.
[3]史佳琪,张建华.基于多模型融合 Stacking 集成学习方式的负荷预测方法[J].中国电机工程学报,2019,39(14):4032-4042.