分类号_____　　密级 _____
U D C _____　　编号 10741

兰州财经大学
LANZHOU UNIVERSITY OF FINANCE AND ECONOMICS

硕士学位论文

(专业学位)

论文题目 基于 GA-XGBoost 的量化投资策略研究

研 究 生 姓 名：徐罡

指导教师姓名、职称：杨世峰 教授

学 科、专 业 名 称：应用经济学 金融硕士

研 究 方 向：金融投资与理财实务

提 交 日 期：2023 年 6 月 7 日

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：徐罡 签字日期：2023.6.7

导师签名：[signature] 签字日期：2023.6.7

导师(校外)签名：_____ 签字日期：_____

关于论文使用授权的说明

本人完全了解学校关于保留、使用学位论文的各项规定，同意（选择“同意”/“不同意”）以下事项：

1.学校有权保留本论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文；

2.学校有权将本人的学位论文提交至清华大学“中国学术期刊（光盘版）电子杂志社”用于出版和编入 CNKI《中国知识资源总库》或其他同类数据库，传播本学位论文的全部或部分内容。

学位论文作者签名：徐罡 签字日期：2023.6.7

导师签名：[signature] 签字日期：2023.6.7

导师(校外)签名：_____ 签字日期：_____

Research on Quantitative Investment Strategy Based on Genetic Algorithm and Extreme Gradient Boosting

Candidate : Xu Gang

Supervisor : Yang Shifeng

摘 要

随着我国 A 股市场的不断完善与发展，传统的投资策略如基本面研究、技术面研究等已经很难使投资者在市场上获得超额收益。近年来，量化投资的概念进入人们视野，量化投资基金的整体规模逐渐增大，市场中量化交易的占比也越来越高。在未来这种新型投资方式将成为市场上主流的交易方式，同时也会成为金融机构争夺客户资源的主要工具。因此，研究量化投资策略对于我国量化投资行业的发展格外重要。

本文将遗传算法引入 XGBoost 模型的超参数调优过程，提高了模型的调参效率和预测准确率，在对策略进行历史数据回测后，获得了超过沪深 300 指数的收益。在选股模型的构建过程中，本文采用设置标签的方式对股票进行分类，设置得分评价方案对预测结果进行排序，同时使用滚动训练的方式来保证模型训练时使用到的是最新数据。在对遗传算法的设计上，使用 F1-score 作为算法的适应度函数。

通过因子重要性分析，本文发现在所使用的 31 个技术指标之中，波动率类指标如归一化的波动幅度均值（NATR）的分类表现最强，其次是重叠类指标如双指数移动平均线（DEMA）、考夫曼的自适应移动平均线（KAMA），周期类指标、成交量类指标等其他指标的分类效果不明显。

关键词：量化投资；机器学习；XGBoost；遗传算法

Abstract

With the continuous improvement and development of China's A-share market, traditional investment strategies such as fundamental research and technical research have made it difficult for investors to obtain excess returns in the market. In recent years, the overall scale of quantitative investment funds has gradually increased, and the proportion of quantitative trading in the market has also increased. In the future, quantitative investment this new investment method will become the mainstream trading method in the market, and become the main tool for financial institutions to compete for customer resources. Therefore, the study of new quantitative investment strategies is particularly important for the development of quantitative investment in China.

In this paper, for the first time in the field of financial quantification, the genetic algorithm is introduced into the hyperparameter tuning process of the XGBoost model, the stock selection model based on the XGBoost algorithm is optimized, and the return exceeding the CSI 300 index is finally obtained through backtesting of historical data. Compared with the traditional quantitative stock selection model, our model can significantly improve the computational efficiency. At the same time, due to the selection of suitable parameters, the prediction accuracy of the model and

the return rate of backtesting of historical data have been improved to varying degrees.

This paper also describes the construction process of the stock selection model in detail, and explains the setting of various aspects of the model. We use the method of setting labels to classify stocks, set score evaluation scheme to sort the prediction results, and select the rolling training method to ensure that the data used in model training is relatively new. In the design of genetic algorithm, F1-score is used as the fitness function of the algorithm.

Through factor importance analysis, we found that among the 31 technical indicators used, volatility indicators such as the normalized mean volatility (NATR) performed the strongest in the classification task, followed by overlapping indicators such as double-exponential moving average (DEMA), Kaufman's adaptive moving average (KAMA), and other indicators such as cycle indicators, volume indicators have less obvious classification effect.

Keywords： Quantitative investment; Machine learning; Genetic Algorithm

目 录

1 绪论

1.1 研究背景

随着我国 A 股市场的不断完善与发展，市场有效性的不断提高，传统的投资方式如基本面研究、技术面研究等很难使投资者在市场上继续获得超额收益。近年来，量化交易这种新兴模式的出现，为投资者的投资决策行为提供了另一种思路。量化投资基金的整体规模在逐年增加，市场中量化交易的占比也越来越高，可见在未来，量化投资将成为主流的交易方式，同时也将成为各金融机构研究的重点领域。

量化投资以市场上多种技术指标、财务指标等数据为基础建立数学模型，运用仿真分析及迭代算法不断修正数学模型，以达到预测未来股票波动的目的。近年来，计算机技术飞速发展，惠及金融领域后，各种与机器学习、深度学习算法相结合的量化投资策略逐渐兴起。

1.1.1 量化投资的发展

量化投资通过对标的资产的相关信息建立数字化模型，以模型为基础构建投资策略来获取稳定收益。它融合了基本面分析法和技术分析法两大投资分析方法，在海外的发展已有 30 多年的历史。量化投资的绩效稳定，近年来得到了越来越多投资者的认可，其市场规模和份额也在不断扩大。量化投资的概念最初由詹姆斯-西蒙斯（James Simons）创立的文艺复兴科技公司提出并引入大众的视野。文艺复兴科技公司于 1982 年成立，1989 年该公司推出旗下第一个量化投资基金大奖章基金（Medallion），在之后的 20 年间，该基金的实际年均回报率高达 60%，远高于同期标普 500 指数的年均回报率。大奖章基金的成功，吸引了更多金融机构着力发展对量化投资的研究。

我国的量化投资行业起步较晚，在 2010 年以前，量化投资的主要形式为量化择时、中高频交易等，而且投资者能选择的投资标的不多，主要是一些被动

指数基金，在这个时期，市场上真正从事量化投资研究的人很少。次贷危机在2008 年席卷美国，大量华尔街的金融机构面临崩溃而破产清算，而此时国内正计划推出股指期货产品来激发市场活力。在这双重因素的作用下，市场上出现海外量化从业人员归国潮，从此萌生了第一批中国量化私募管理人。证监会在2010 年 3 月宣布中信、国泰君安、国信、光大、海通和广发 6 家证券公司获得融资融券首批试点资格，随后沪深证券交易所开始接受证券公司融资融券交易申报，再加上同年 4 月沪深 300 股指期货推出，量化策略对冲工具的使用范围得到极大丰富，市场中性策略进而得到广泛的实践应用。在之后的五年间，各大投资管理人在探索中打磨策略框架、优化产品要素，各类型的指数增强策略、CTA 策略、套利策略和 FOF 基金等相继出现，进入市场和投资人的视野。2015 年，上证 50ETF 期权、上证 50 股指期货和中证 500 股指期货进入市场，各种量化对冲策略的选择余地被大大扩充。然而在 2015 年，A 股经历三轮暴跌，泡沫破灭，证监会为了维持市场稳定而限制股指期货交易，虽然投资标的受到制约，但是量化策略的研究没有停滞，市场上开始出现量化多策略、量化选股这类更具想象空间的新策略类型。直到 2019 年股指期货恢复常态化，市场中性策略再次以新的面貌出现在市场上。经过了多年的行业竞争，国内量化策略的平均年化超额收益从最初的 40%以上逐步压缩到现在的 25%左右，赛道愈发拥挤且策略趋于同质，使得形势实际上比往日更加严峻。

中国的量化行业发展较为迅速，2020 年后迎来规模增长浪潮，在投资者甚至整个金融业内得到了一些新的认可。截至 2021 年第四季度，证券类私募中量化产品规模接近 1.6 万亿，约占私募证券类产品规模的 25%。2022 年，传统的单一策略超额收益空间被不断压缩，为了寻求新的出路，一些先行的管理人开始更多地使用多框架、多策略、多品种、多市场的方式，整合各流派的策略风格，在引入新的思想的同时积极布局海外，这其中就包括与人工智能和机器学习相结合的量化策略。

1.1.2 量化投资与机器学习算法相结合

1992 年，芝加哥大学布斯商学院的两位经济学家 Eugene Fama 和 Kenneth French 提出了一种资产定价模型，即 Fama-French 三因子模型。该模型是量化投

资领域中的一个著名因子模型，为后来众多的多因子量化模型奠定了基础。随着时代发展，越来越多的因子被挖掘出来，从基本面因子到技术面因子，从市场情绪因子再到宏观分析因子，甚至未来的天气状况都可以作为量化分析的指标。众多的因子如何筛选成为了量化基金经理所面临的难题，机器学习技术的产生似乎为这个问题提供了解决方案。

在证券市场这个无数主体相互影响相互作用的混沌系统里，想通过一个模型把市场所有的经验和规则全部表达出来是不现实的。然而机器学习提供了一种方案：将已有的各种环境下产生的大量数据提供给计算机，由计算机来自主学习寻找规律，投资者根据计算机的计算结果来进行投资。如此一来，大量的因子不仅不是问题，反而能帮助投资者更好地完成投资决策。

机器学习算法从市场中全方位提取信息，并将其应用于交易策略中。在经过多年的对未来超额收益的预测研究中，大量的预测指标被挖掘出来，且被证明了这些指标具有预测回报的能力。然而，预测因子通常具有高度相关性，如果解释变量数过多而样本数过少，或者解释变量高度相关，传统的预测方法如回归分析等将会失去效果。机器学习算法的变量选择技术和降维技术（如主成分分析法），比较适合解决此类问题，它可以减少样本自由度并降低解释变量之间的复杂相关程度。如果进一步考虑高维变量输入来预测风险溢价的问题，传统的量化方法会变得更加吃力。变量间如何相互作用，变量与被解释变量之间是否是线性关系，模型需要经过非常复杂的计算来解决这些问题。机器学习的三个优势使其非常适合解决资产价格的预测问题。首先是它的多样性，包含了多种不同的方法，面对各种需求都可以采用匹配的方法。其次，无论是广义线性模型如线性回归和逻辑回归，还是以回归树为基础的决策树模型及其衍生模型，或者神经网络相关的种种方法，机器学习构建了复杂的非线性关联。最后，机器学习通过在损失函数中增加 L1 正则项和 L2 正则项等惩罚项来避免模型过于激进，这种方式使机器学习方法可以覆盖更广阔的功能形式，且尽量少的发生过拟合和错误发现（Gu，2020）。

1.2 研究意义

与欧美国家不同，中国的 A 股市场有着散户为主，市场有效性偏低的特点，

且散户追涨杀跌、操作活跃，使得证券市场的动荡加剧（方浩文，2012）。我国 A 股市场的个人投资者整体规模长期以来一直高于机构投资者，然而从 2015 年开始，机构投资者的持股和交易占比稳步上升，个人投资者占比呈现下降趋势，直到 2022 年第三季度，A 股个人投资者交易占比不足 60%。受全球资本市场的影响，A 股投资者结构从原本的散户主导正在向机构化转变，这种趋势是符合经济发展规律的。随着资本市场的完善与发展，未来机构化、专业化的程度将越来越高，市场的结构性变化，将导致传统的交易方式交易策略作用不再显著，更多科学、专业、有效的投资策略需要被研究和开发出来，帮助投资者获得超额收益。因此，量化投资这种符合市场需求的投资方式，需要人们给予更多的关注。

传统的量化投资模型分析角度往往局限于基本面分析或者技术分析两个维度，所使用的数据均为可再挖掘价值较低的结构化数据。而新兴的基于机器学习和深度学习算法的交易策略，可以从全新的视角来描绘特征与回报之间的复杂关系，这种方式将对金融行业产生巨大的影响。中国市场量化投资的起步较晚，量化策略的研究水平较低，而且与人工智能算法结合的研究深度还比较浅显。构建系统科学的量化投资策略，不仅可以使投资者快速有效地识别投资机会，为投资者提供更多的选择，同时也可以帮助投资者减少投资风险，对中国资本市场的发展有重要意义。

1.3 文献综述

1.3.1 传统量化投资的文献综述

Sharp（1964）提出了资本资产定价模型（CAPM），该模型揭示了股票的预期收益率与风险资产的关系，认为资产的预期回报率和资产的系统风险有关。但是，Banz（1981）的研究发现，公司的市面价值也可以在一定程度上解释公司股价的变化。后来 Fama and French（1992）提出 Fama-French 三因子模型，该模型解释了 CAPM 所不能解释的两个市场异象：一是小市值公司的股票平均收益率更高（size premium）；市净率（P/B）低的公司的股票平均收益更高（value premium）。针对这两个市场异象进行研究，结果表明上市公司的市值、

账面市值比、市盈率可以解释股票回报率的差异。在 Fama-French 三因子模型的基础上，Carhart（1995）加入一年期收益动量异常因素，构造了四因素模型。Joseph（2001）构建了一个简单的基于财务指标的基本面分析策略，研究将高账面市值比的公司加入投资组合，是否能改变投资者所获得的收益分布。研究发现，加入实力雄厚的高账面市值比公司后，投资者获得的年平均回报至少可以增加 7.5%，回报的整体分布则向右移动。总体而言，市场价格并未及时充分反映公司的历史财务信息。A. Khodadadi 等（2006）简要概述了构建成功的量化投资组合策略需要考虑的关键问题，如集成数据集模块、调仓模块、回测模块以及业绩归因等，其中组合优化是此类策略的核心要素，除了讨论针对交易成本和税收进行调整的标准均值-方差优化模型外，还讨论了多期投资组合选择和稳健优化方法等问题。潘莉（2011）使用股票的贝塔系数、市盈率、市净率等指标作为解释变量研究股票的回报率，发现市场回报率变化的 90%以上都可以由股票市值、平均回报率和市盈率这三个因子解释，以这三个因子为基础建立模型，结果表明，股票市值背后既有风险也有特征因素，而市盈率对回报率的影响只与股票特征有关。周亮（2017）为了研究基本面因子对股票收益率的影响，使用每股盈余、净资产收益率等财务指标和股价、换手率等技术指标作为解释变量，以中小板上市公司为研究对象建立实证模型。结果表明，公司规模、股价、股东人数变动、换手率及毛利率五个指标对股票收益有显著影响。胡熠（2018）检验了巴菲特价值投资策略在中国股票市场的适用性，为了刻画巴菲特的价值投资风格，构造了综合性指标 B-score，从安全性、便宜性以及质量 3 个维度考虑。研究发现，基于不同市场状态和其他截面指标状态，B-score 可以很好地预测股票的未来回报。另外， B-score 策略在极端市场环境下（2015-2016）仍然可以获得正回报。考虑到中国和美国资本市场的巨大差异，Liu（2019）根据中国股票市场的特点构建了中国版的规模因子和价值因子。该研究认为盈利市值比（EP）是更好的价值因子指标，且剔除市值最小的 30%股票以减少壳公司的影响，在 Fama-French 三因子模型的基础上加以改进，加入换手率因子，解释了 CAPM 所不能解释的市场异象。

1.3.2 量化投资结合机器学习算法的文献综述

随着学术理论研究的不断深入和应用实践的不断探索，基于金融大数据的机器学习量化模型成为近年来备受关注的研究对象。Huseyin Ince（2000）将支持向量机算法与 BP 神经网络和径向基函数网络进行比较，并应用于财务预测领域。研究发现，SVM 的训练会导致二次规划问题。全林（2009）分析了中国 A 股市场的股票选择问题，通过稳健的改进主成分分析结合支持向量机算法构建 PCA-SVM 模型对股票进行特征提取。结果表明，运用 PCA-SVM 算法得到的组合回报率超过了市场基准。李斌（2016）以技术指标作为输入变量，结合不同的机器学习算法构建量化投资算法 ML-TEA，研究预测股票数日之后的涨跌方向。结果显示，三种模型的表现都超过了基准策略和现有策略。黄卿（2018）研究了股指期货的预测问题，通过收集沪深 300 股指期货 1 分钟高频数据，构建量化投资模型；研究还对比分析了神经网络、SVM 和 XGBoost 对股指期货 1 分钟价格变动方向的预测能力。研究结果表明，三种机器学习方法的预测能力都很强，XGBoost 的预测能要优于其他两种机器学习方法。王重仁（2019）将贝叶斯参数优化方法引入 XGBoost 算法来进行信用评估研究互联网信贷行业的个人信用风险评估问题。结果表明，该方法的预测效果优于支持向量机、Logistic 回归、神经网络和随机森林等对比算法，同时贝叶斯参数优化方法优于网格搜索法和随机搜索法。Jiang（2019）收集 75 个公司因子来预测中国股票市场的收益率，不仅使用了 Fama-Macbeth 回归模型，而且采用了主成分分析法以及偏最小二乘法来对特征降维，以解决因子过多导致的多重共线性问题。研究发现交易摩擦类、动量类和盈利性因子无论在统计意义还是经济意义上都可以更有效地预测未来股票收益。李斌（2019）以中国 A 股市场上的 96 个异象因子为指标，系统性地对比分析 12 种机器学习方法的股价预测能力，发现线性机器学习算法的表现整体优于单因子和线性回归模型，而非线性机器学习算法的表现总体优于线性机器学习算法；另外，在非线性算法中，除神经网络算法外，XGBoost 算法的表现最为显著。任君（2019）将 Lasso 回归方法与支持向量机（SVM）和改进的长短期记忆网络（LSTM）相结合，构建量化投资模型。使用改进的网格搜索法和指数衰减法对 SVM 和 LSTM 进行参数调优，输入技术指标作为自变量，构建 GSVM-L 和 ELSTM-L 量化投资模型。结果表明：该模型

具有较好的投资收益和较强的抗风险能力，且 ELSTM-L 模型对交易成本的容忍度更高。张虎（2020）以过去 60 个交易日的因子数据作为模型自变量，使用自注意力神经网络算法构建投资模型，研究结果表明，该投资策略相比于沪深 300 指数具有更高的收益和较低的风险。Shihao Gu（2020）使用机器学习的方法进行实证资产定价，发现使用决策树（decision tree）和神经网络（neural networks）这两种机器学习模型的策略表现最佳，而且动量（momentum）、流动性(liquidity)和波动性(volatility)这三个因子在所有的方法中都承担着主要的预测信号。Chen 等（2022）将卷积神经网络和深度学习技术相结合用于量化金融投资。结果表明，卷积神经网络和深度学习算法可以获得相对准确的投资策略，从而确保投资回报，降低投资风险。

1.3.3 遗传算法的文献综述

近年来，遗传算法凭借其简单、易实现、适应度高等特点，在各个领域都得到了广泛的研究和应用，在金融领域，遗传算法被应用于人工智能算法的参数优化过程，进而预测证券价格。Kim（2000）将遗传算法用于特征离散化，结合人工神经网络算法构建股价指数预测模型。结果表明，基于遗传算法改进人工神经网络算法，可以降低特征空间的复杂性。赵健（2011）采用遗传算法的思想对众多数量化模型进行参数优化，改进多种策略组合，以及通过进化产生新的策略，通过动态调整交易策略来进行仿真交易。结果表明该方法相较于大多数单一策略回报更高，稳定性更好。王德明（2012）通过自相关性分析寻找对预测值影响最大自变量作为输入变量，将遗传算法引入反向传播神经网络算法的参数优化过程。实验结果表明，该方法较传统的反向传播神经网络预测精度更高、收敛速度更快。齐岳（2015）通过遗传算法分别求解不同投资组合的有效边界，将其有效边界的精确解进行比较，精确解由参数二次规划法获得，研究遗传算法在不同约束条件下的效率问题。结果表明，遗传算法的有效率随着股票数目的上升呈下降的趋势。Chung H（2018）提出了一种结合长短期记忆（LSTM）网络和遗传算法（GA）的混合方法。以往基于启发式的试错法通常用于估计 LSTM 网络的时间窗口大小和架构因素。该研究通过提出一种使用遗传算法确定 LSTM 网络时间窗口大小和拓扑结构的系统方法来研究股市数据的

时间特性。为了评估所提出的混合方法，选择了每日韩国股票价格指数（KOSPI）数据。实验结果表明，LSTM 网络和 GA 的混合模型优于基准模型。邓翔（2020）在时间序列预测领域将遗传算法引入 Prophet 模型，以宏观时间序列 CPI 为研究对象，构建了新的 Prophet 模型。研究表明，该模型不仅可以简化超参数的优化过程，还能够更好地拟合和预测含有季节性和突发事件冲击的时间序列。李晓寒（2022）改进了传统的遗传算法，将其与图神经网络相结合预测股价未来波动情况，针对支持向量机、长短期记忆网络等智能算法在股市波动预测过程中特征选择困难及时序关系维度特征缺失的问题，构建了 IGA-GNN 模型。结果表明，该模型可以更好的提取有效股票特征，进而预测股价未来波动。

1.3.4 文献述评

通过对以上一些文献的回顾我们可以发现，传统的量化投资方法往往基于一些假设和前提，进而获得特定变量与投资回报的相关性，并且认为这些相关性将会永远存在下去。比如资本资产定价模型假设所有投资者都按 Markowize 的资产选择理论进行投资，对期望收益、方差和协方差等变量的估计完全相同，投资者可以以无风险利率自由借贷等。然而在实际的市场中，这些假设并不成立，市场风格在不断变化，投资者结构也在不断变化，这些相关性很难在实际操作中得到有效的验证。另外，传统的量化投资研究往往假设解释变量和被解释变量存在线性关系，建立的模型大多基于广义线性回归模型，这样的方法忽略了某些变量与目标之间的非线性关联，同时模型也就不能完整的体现二者之间的关系。

相比之下，与机器学习方法结合的量化投资策略则是基于“学习”的思想，突破了传统的思维框架，减少对模型经济意义的关注，而更多集中在统计意义上。首先，机器学习方法具有高度多样性，从最基本的广义线性回归模型到 Bagging 模型（随机森林）、Boosting 模型（梯度提升决策树），再到神经网络算法等等，研究者可以根据需要选择适配的算法模型。其次，机器学习模型可以解释变量间的非线性关系，即使样本过少或变量高度相关，仍然可以训练出行之有效的模型，而此时传统的量化方法可能不会有太好的效果。最后，机器学

习算法的自由度很高，使用者可以通过调整具体参数来适配不同的应用场景。另外，很多与机器学习算法相结合的策略都会采取滚动训练的模式，使得模型在面对各种市场变化时都能够有效地做出反应，且在实际应用时有着较好的表现。

在众多机器学习方法中，XGBoost 作为一种梯度提升树框架下的算法，对于股票定价具有较强的解释力（李斌，2019）。以神经网络算法为代表的深度学习算法虽然在图像、文本等非结构数据的预测问题中有较好的表现，但是在股票定价方面，过高的模型复杂度，可能带来过拟合的风险。XGBoost 善于捕捉复杂数据间的依赖关系，且可以通过提前将特征重要性排序来降低计算量，提高计算效率，从大规模数据集中获取有效模型。

不同于其他算法，XGBoost 算法的特点之一是它的超参数较多。超参数是在开始学习过程之前设置的参数，如学习率、最大叶子个数、最大深度等。超参数较多意味着它的调参过程将非常复杂，任何参数的微小变化都可能对结果产生巨大影响，所以参数寻优的过程尤为重要。在以往的研究中，采取的方法一般为网格搜索，网格搜索即全局搜索，本质是以一定间隔遍历全部参数组合的方法，在有限的运算能力下这种方式效率较低；另一种方法为贪心算法，然而贪心算法容易使调参过程陷入局部最优，难以获得全局最优的参数组合。遗传算法的独特优势可以解决这两种方法的弊端。遗传算法凭借其群体搜索的策略和遗传算子产生的随机性，可以实现解空间上的分布式信息采集和探索，从而大大提高寻求最优解的效率。

1.4 研究内容与研究方法

1.4.1 研究内容

本文以沪深 300 指数成分股作为研究对象，收集了从 2017 年 1 月 1 日至 2022 年 1 月 7 日的真实股票数据，使用 XGBoost 算法构建选股模型，并引入遗传算法对模型的超参数寻优，使用滚动训练的方式，在五年的数据周期内进行历史数据回测并分析其结果。全文分为五个部分，具体内容如下：

第一章，绪论。本章首先介绍了文章的研究背景，论述研究意义，其次回

顾以往国内外有关量化投资、机器学习和遗传算法的文献，最后介绍了本文的研究内容和方法以及可能存在的创新点。

第二章，相关理论基础。本章着重介绍文章所使用模型的相关理论，包括 XGBoost 算法，遗传算法等

第三章，基于 GA-XGBoost 的选股模型构建。本章具体阐述了整个选股模型的构建过程，从原始数据的获取及处理开始，介绍该模型所使用的因子情况、以及介绍标签设置和得分设置方式、模型训练模式以及遗传算法的相关设置等，最后在以上步骤的基础上，构建基于 GA-XGBoost 的选股模型。

第四章，策略评价与分析。本章主要从多个方面对选股策略进行评价与分析，一是从模型的角度评价预测准确率，二是通过历史数据回测，检验策略在实际选股过程中的绩效水平，三是根据因子重要性排序，分析重要因子的作用机理，四是将本模型与传统 XGBoost 模型以及支持向量机模型做对比分析，比较在相同条件下以不同模型为基础的选股策略绩效水平。

第五章，总结与展望。对全文进行总结，提出文章的不足之处，并对未来研究方向提供思路。

1.4.2 研究方法

本文所使用的研究方法有：文献研究法和实证分析法。

（1） 文献研究法

本文主要通过访问中国知网、CSDN 技术社区和谷歌学术等网站，搜集有关资本定价、量化投资和人工智能算法等相关文献和研究成果，并对其进行提炼和整理，为文章后续研究提供基础。

（2） 实证分析法

本文的原始数据来源于证券宝（Baostock.com）证券数据平台，基于计算机编程语言 Python 进行实证分析。Python 提供了丰富而强大的第三方库供使用者直接调用。比如 Scikit-Learn 就是一个针对 Python 的免费软件机器学习库，与数据科学库 Pandas 和 Numpy 联合使用，可以提高实证分析的效率。

1.4.3 研究路线图

图 1.1 研究路线图

1.5 本文可能的创新点

本文试图构建一个科学、有效的投资策略，使用 Ta-Lib 数据库中 31 个技术因子，基于 GA-XGBoost 构建选股模型，最终取得了超过沪深 300 指数的收益率。本文的主要贡献有：

（1）改进传统的基于 XGBoost 算法的选股策略，丰富了模型的细节和相关条件设置，其中滚动训练的模式不仅保证每期训练都使用了最新数据，而且模型超参数也会根据数据集的变化而实时更新。

（2）将遗传算法引入 XGBoost 调参过程，构建选股策略。一方面有效缩短了调参过程，提高模型的计算效率；另一方面，在选取了合适的参数以后，模

型在实际选股过程中的表现也有显著提高。

（3）从模型的视角分析因子重要性，筛选出分类效果较突出的重要因子，并提供经济意义和算法意义上的解释。

2 相关理论基础

2.1 有效市场假说

Eugene Fama 提出的有效市场假说认为，资本市场是否有效取决于证券价格能否充分而准确地反映全部相关信息。换句话说，如果证券市场的所有参与者接受了公开信息后，证券价格不受任何影响，说明市场对信息的反映是有效率的，即市场是有效的。市场有效意味着以信息作为参考的证券交易策略不可能获取超额利润。

有效市场假说基于三大基本假设：

（1）理性投资者假设。市场上的每个参与者都是理性的经济人，他们每时每刻都在分析公司的基本面，利用所有的可获得信息来评价公司的盈利能力和未来发展前景，进而预测股票价格。除此之外，他们还预估可能发生的风险事件，并谨慎地在风险与收益之间进行权衡取舍。

（2）股价均衡假设。股票的价格处在供求均衡的水平上，在这些理性的投资者们当中，认为股价被高估的人与认为股价被低估的人正好相等。假如股价脱离均衡水平，即存在套利的可能性，就立刻会有套利者买进或者卖出股票来使股价回归到原本的状态，使供求关系恢复平衡。

（3）信息有效假设。市场上的股票价格是所有理性投资者的共同结果，已经充分反映该资产的所有可获得的信息，如果新的信息出现，股价就会立刻发生相应变动。实际上，股价的异动在利好或利空消息刚传出时就已经开始，当信息真正落地时，股票的价格也已经涨或跌到适当的价位了。

以往的很多研究都证明了我国的 A 股市场是一个非有效市场，其中的原因是多方面的，比如我们的市场交易机制还存在不合理的现象，金融产品种类不够丰富，以及信息披露质量不高等等。我国 A 股市场的非有效，说明当前市场对资产的定价并未充分反应其历史信息，股票的价值和价格存在背离现象，未来的价格变化将进一步对过去的价格信息作出反映。在这种情况下，投资者可以利用技术手段对历史数据进行分析，预测未来价格的变化，从而获得超额收

益。因此，本文的研究以有效市场假说中关于非有效市场的推论为理论引导，以中国 A 股市场非有效的结论为基础，将技术指标输入机器学习算法，构建一个可以获得超额收益的投资策略是可行的。

2.2 证券技术分析理论

证券技术分析是证券投资分析方法的一种，不同于基本面分析，技术分析的主要研究对象是金融市场交易行为。技术分析通过计算各种交易数据，分析各类图形形状用以判断市场供求关系的变化，进而指导交易决策。技术分析对市场趋势的预测集中于分析资产价格的波动规律。

证券技术分析存在三大假设：市场行为涵盖一切信息、证券价格沿趋势运动和历史会重演。

（1）市场行为包涵一切信息，即能够影响某种证券价格的任何因素——基础的、政治的、心理的或任何其他方面的——实际上都反映在其价格中。

（2）证券价格沿趋势运动，价格的变动按一定规律进行，价格有保持原来方向运动的惯性。技术分析法是从供求关系的角度来解释价格的惯性的，认为供求关系是一种理性和非理性力量的综合，价格运动反映了一定时期内供求关系的变化。如果供求关系保持不变，证券价格就会按照当前趋势一直持续下去。一旦价格走势发生反转，必然导致均衡被打破，资产的供求关系发生了彻底的改变。

（3）历史会重演，在相同的条件下，市场必然会出现相同的结果。在技术分析里，“历史会重演”的假设是从投资者的心理角度来理解和分析的，当市场满足上涨的条件价格就会继续上涨，满足下跌的条件价格就会继续下跌。市场出现和过去相同或相似的情况时，投资者会根据过去的成功经验或失败教训来做出目前的投资选择，市场行为和证券价格走势会出现历史重演。因此，技术分析法认为，根据历史资料概括出来的规律已经包含了未来证券市场一切变动的趋势，所以可以根据历史预测未来。

然而，由于技术分析的第一假设市场涵盖一切信息难以得到完美证实。信息损失是必然的，因此市场行为包括一切信息也只能是理想状态。在第一假设受到挑战时，技术分析的有效性将受到冲击。从历史经验来看，当价格进入长

期低波动的横向状态时，技术分析的有效性将大打折扣，由于市场关注度下降，进而导致信息损失变得更大。相反，在市场高度关注的市场中，价格往往大幅波动，而在此过程中技术分析的有效性将大大提高。

2.3 行为金融学理论

有效市场假说假设人们行为都是理性的，理性的人总是能够最大化其预期效用，并能掌握处理所有可得信息，形成均衡预期收益。然而在现实的市场上存在着大量难以解释的金融异象，经过很多观察和研究证明了这些异象是真实存在的。为了能够得到合理的解释，人们开始从另一个角度入手，暂时忽略有效市场理论的完全理性人假设，从心理学的角度出发，结合相关学科的理论成果，试图揭示金融市场规律中的非理性因素，行为金融学这一学术流派由此逐渐形成。

行为金融学是金融学、心理学、行为学、社会学等学科相交叉产生的学科，它认为证券的市场价格还取决于投资主体的心理因素和个体行为，而不仅仅取决于证券的内在价值，因此研究投资者的心理和行为可以在一定程度上解释证券价格的变动。行为金融学不相信随机漫步理论，股价并不是随机游走的，而是在一群人的共同作用下变动的。基于心理学、社会学和人类学的社会科学学科所研究建立的相关模型，运用在金融建模上，可以预测市场投资者的整体行为进而预测证券价格变化。

人不可能如有效市场假说所假设的一样时时刻刻保持理性，人在做决策的时候往往会受到非理性因素的影响，如一些固有偏见和情绪，这些都会使投资者的决策脱离客观。比如面对一次成功的投资，人们往往会归功于自己的判断能力，而忽略了运气因素在其中的作用，而如果是一次失败的投资，人们又会认为是自己运气不好而非判断出现失误，这种心理在行为金融学中被称作“过度乐观”。诸如此类的行为金融学概念揭示了证券市场投资主体的非理性和市场的非有效性，同时也赋予了技术分析理论和量化投资理论存在的意义。人本身具有主观性，这种主观性带来的不确定性是很多投资方法失败的原因，而量化投资模型基于客观的数据和算法输出，其结果是客观而确定的，作为投资者需要完善的是方法而不是克服某种情绪。

2.4 XGBoost 算法

2.4.1 XGBoost 算法简介

XGBoost 的全称是 eXtreme Gradient Boosting，中文含义为极限梯度提升，XGBoost 算法起源于陈天奇博士（Tianqi Chen） 在分布式机器学习社区的一个研究项目，之后他与 Carlos Guestrin 合作将算法深化并在 2016 年的 SIGKDD 大会上发表，该算法一经问世便在整个机器学习领域引起轰动。XGBoost 是梯度提升树（GBDT）的改进版本，在算法层面和系统设计层面都有不同程度的创新性的改进。XGBoost 中的 X 代表的就是 eXtreme（极致），XGBoost 能够更快的、更高效率的训练模型。XGBoost 在许多机器学习以及数据挖掘的任务中表现惊艳，非常适合处理结构性数据，而对于图像、文本等非结构性数据的预测问题，现在普遍认为人工神经网络等深度学习算法更为适合。

2.4.2 XGBoost 算法基本原理

XGBoost 通过对原函数做二阶泰勒展开，作为新的损失函数，然后求解损失函数的最小值来构建最优模型。此外，XGBoost 为了避免模型过拟合，在损失函数中加入了 L2 正则项，L2 正则项包括了子树数量和子树叶节点数等表示模型复杂度的信息，这些数值过高会引起损失函数的相应惩罚，在实际建模中会将低重要性的解释变量权重降至接近 0。在效率上，XGBoost 训练时可以在特征维度上并行，将每个特征按特征值大小对样本进行预排序并存储为 Block 结构，在下次查找时可以重复使用，这种方法相比建模效率较一般的 GBDT 有了大幅的提升。

从本质上来讲，XGBoost 模型由多个基分类器组成，最终结果由基分类器共同决定。基分类器一般采用弱分类器，比如决策树和逻辑回归甚至 SVM 都可以作为 XGBoost 的基分类器来使用。模型将原始数据分配给基分类器进行训练，将得到的结果按照一定权重加起来得到最终的结果，公式如下：

$$\hat{y}_i^{\,t} = \sum_{k=1}^{t} f_k\,(x_i) = \hat{y}_i^{t-1} + f_t(x_i)$$

其中，$\hat{y}_i^{\,t}$表示经过 t 次迭代后对样本 i 的预测情况，$\hat{y}_i^{t-1}$表示前 t-1 个基分

类器的预测结果，$f_t(x_i)$表示第 t 个基分类器模型。

一般来讲模型的损失函数可以通过比较预测值与真实值的差别来得出：

$$Loss = \sum_{i=1}^{n} l(y_i, \hat{y_i})$$

模型的偏差和方差衡量了模型整体的预测效果，偏差体现了模型对已知结果的数据集的拟合程度，由损失函数表示，而方差体现了模型的泛化能力，过高的方差往往意味着模型过拟合，对新的数据预测效果不好，为了减小模型的方差，在目标函数中加入正则项来控制模型复杂程度，目标函数定义如下：

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y_i}) + \sum_{i=1}^{t} \Omega(f_i)$$

上式的右半部分表示将 t 棵树中每棵树的复杂度求和。

然后，将目标函数进行二阶泰勒公式展开，得到最终的目标函数近似值：

$$Obj^t \simeq \sum_{t=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

对每一次迭代的损失函数求一阶导和二阶导（由于前一次迭代的损失是已知的，所以这两个值就是常数），求解目标函数最小值得到每次迭代的目标值，相加得到一个整体模型。

关于决策树的构造，我们定义一颗棵树：叶子结点的权重向量 w；样本到叶子节点的映射关系 q：

$$f_t(x) = \omega_q(x), \omega \in R^T, q: R^d \to \{1,2,\cdots,T\}$$

叶子数表示了决策树的复杂度，叶子节点越少模型越简单，另外为了避免欠拟合，叶子节点的权重不宜过高，所以综合考虑目标函数的正则项由生成的所有决策树的叶子节点数量，和所有节点权重所组成的向量的范式共同决定：

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

代入 XGBoost 的目标函数，最终目标函数为：

$$Obj^t = \sum_{j=1}^{T} [G_j \omega_j + \frac{1}{2}(H_j + \lambda)\omega_j^2] + \gamma T$$

要注意G_j和H_j是前 t-1 步得到的结果，其值已知可视为常数，只有最后一棵树的叶子节点ω_j不确定：

$$G_j = \sum_{i \in I_j} g_i$$

$$H_j = \sum_{i \in I_j} h_i$$

2.4.3 XGBoost 算法的优势

XGBoost 算法的优势在于：

（1）XGBoost 损失函数中的正则项包含了树节点的个数和子树叶节点数值，将这两个参数加入损失函数可以控制模型的复杂度，避免过拟合：

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

（2）支持并行处理。XGBoost 可以在训练过程中在特征维度上并行，而不是在不同树之间并行。XGBoost 本质上仍然采用 Boosting 思想，在一棵树训练完成后才开始下一颗树的训练。而所谓特征维度的并行是指在模型训练之前，将每个特征按特征值大小对样本进行预排序并存储为 Block 结构，这种 Block 结构可以在查找特征分割点时重复使用，以此避免多次计算。而且特征已经被存储为一个个 Block 结构，那么在寻找每个特征的最佳分割点时，可以利用多线程对每个 Block 并行计算。

（3）XGBoost 算法允许存在缺失值。在训练过程中，如果特征出现了缺失值，处理步骤如下：

a）首先对于特征非缺失的数据，计算出分裂损失并比较大小，选出最大的分裂损失，确定其为分裂节点（即选取某个特征的某个阈值）；

b）然后对于特征缺失的数据，将缺失值分别划分到左子树和右子树，分别计算出左子树和右子树的分裂损失，选出更大的 Lsplit，将该方向作为缺失值的分裂方向（记录下来，预测阶段将会使用）。公式如下：

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\left(\sum_{i \in I_L} h_i\right) + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\left(\sum_{i \in I_R} h_i\right) + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\left(\sum_{i \in I} h_i\right) + \lambda}\right] - \gamma$$

在预测阶段，如果特征出现了缺失值，则可以分为以下两种情况：

a）如果训练过程中，出现过缺失值，则按照训练过程中缺失值划分的方向（left or right），进行划分；

b）如果训练过程中，没有出现过缺失值，将缺失值划分到默认方向（左子树）。

（4）可并行的近似算法。树节点在进行分裂时，以往的做法是使用贪心法

枚举所有可能的分割点以计算每个特征的每个分割点上对应的信息增益，当数据量过大或者数据以分布式存储的情况下，贪心算法效率就会变得很低。所以 XGBoost 还提出了一种可并行的近似算法，用于高效地生成候选的分割点。

2.5 支持向量机

2.5.1 支持向量机简介

支持向量机（Support Vector Machine）是一种经典的机器学习算法，在本文中，将其作为对比算法与本文的模型相比较。1995 年 Vapnik V 提出了支持向量机模型，该模型是一种广义线性分类器，通过求解分类的最优超平面，将问题转化为一个求解凸二次规划的问题，对数据进行二元分类，是一种监督学习的方法。不同于逻辑回归等传统线性回归模型，支持向量机以一种更为清晰，更加强大的方式求解复杂的非线性方程。

支持向量机的本质实际上是对线性可分的数据集空间寻找可以将样本完美分类的最优分类超平面。如果数据集线性不可分，通过加入松弛变量的方式，将低维度空间样本非线性映射到高维度空间使其变为线性可分，这样就可以在该特征空间中寻找最优分类超平面。

2.5.2 支持向量机的基本原理

支持向量机的最终目标是寻找最有分类超平面，首先假设存在一个超平面：

$$w^T x + b = 0$$

基于超平面的表达式，计算某个数据样本点到超平面的距离。假设样本点的坐标为$P(x_1, x_2, \dots, x_n)$，其中x_i表示第 i 个特征变量，那么该样本点到超平面的距离为：

$$d = \frac{|w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b|}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}} = \frac{W^T X + b}{\|W\|}$$

其中$\|W\|$为超平面的范数，常数 b 类似于直线方程中的截距。

支持向量即样本中距离超平面最近的一些点，在超平面确定的情况下，可以按一定条件筛选出支持向量。计算支持向量与超平面之间的平均距离。求解

最优超平面的问题等价于求解平均距离最大的超平面，建立目标：

$$\arg\max_{w,b}\{\min(y(w^T x+b)\cdot\frac{1}{\|W\|}\}$$

其中 y 表示数据点的标签，其为-1 或 1。

为了求解超平面更方便，应用拉格朗日对偶性，可以得到原始问题的对偶问题：

$$min\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}a_i a_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}a_i$$

Subject to:

$$\sum_{i=1}^{N}a_i y_i=0$$

$$a_i\geq 0, i=1,2,\dots,N$$

求解该函数的最小值，便可得到最终的分类超平面。

2.6 遗传算法

2.6.1 遗传算法简介

1859 年，英国生物学家达尔文的《物种起源》出版，其中进化论的思想震惊了整个学术界和宗教界。几年后，奥地利植物学家孟德尔从豌豆的杂交实验中发现生物遗传的正确规律，建立遗传学说。后来到了 20 世纪 40 年代计算机问世后，学者们利用计算机模拟生物系统的遗传进化过程。1975 年，美国的 J.H.Holland 教授发表专著《自然界和人工系统的适应性》，在该著作中，他提出了遗传算法（Genetic Algorithm）的概念，首次将自然界生物遗传的规律转化为人工智能算法。

遗传算法借鉴了达尔文的进化论和孟德尔的遗传学说，其本质是一种随机全局搜索的优化方法，可以高效、并行地搜索目标空间，并且在搜索过程中积累历史经验，以自适应的方式求得最优解。遗传算法以其高效、全局的特点，如今被广泛地应用于组合优化、机器学习等领域，是现代有关智能计算中的关键技术之一。

2.6.2 遗传算法与传统算法的差异

遗传算法与传统的搜索和优化算法间存在一些重要区别。

（1）基于种群。遗传算法在搜索过程中是以并行的方式进行的，即搜索迭代以种群的形式而不是某个个体的形式进行。而大多数其他搜索算法则相反，网格搜索、贪心算法等方法都是针对单个解决方案的搜索，只对某一个方案组合进行迭代修改以寻找最佳解决方案。基于种群的搜索方式使遗传算法不仅仅局限于当前的搜索空间，而是在整个解空间上的进行的分布式信息采集和探索，这样的好处一是大幅提高搜索效率，二是能够有效避免搜索陷入局部最优。

（2）遗传表征。遗传算法不是直接在候选解上运行，而是在它们的表示（或编码）上运行。染色体能够利用交叉和突变的遗传操作。使用遗传表示的弊端是使搜索过程与原始问题域分离。遗传算法不知道染色体代表什么，也不试图解释它们。

（3）适应度函数。适应度函数衡量个体的优劣程度，决定什么样的个体可以保留下来。遗传算法的目的是找到利用适应度函数求得的得分最高的个体。与许多传统的搜索算法不同，遗传算法以适应度函数计算得到的值作为评价标准，而不依赖于导数或任何其他信息。面对一些不可求导或者非线性的优化问题时，常规的求极值方法会束手无策，而遗传算法非常适合解决这类问题。

（4）概率行为。大多数传统算法本质上是确定性的，而遗传算法的交叉、突变算子是一种概率行为，交叉基因的位置，突变的位置和变化幅度这些因素都是以一定概率出现的，这种概率行为赋予了遗传算法积极的随机性。例如，选择的个体将被用来创建下一代，选择个体的概率随着个体的适应度得分增加，但仍有可能选择一个得分较低的个体。尽管此过程具有概率性，但基于遗传算法的搜索并不是随机的；取而代之的是，它利用随机性将搜索引向搜索空间中有更好机会改善结果的区域。

2.6.3 遗传算法的基本流程

遗传算法的具体步骤如下：

（1）初始化种群。设置迭代次数，变异概率等基本参数，在范围内随机生成 n 个个体作为初始种群。

（2）计算个体适应度。确定适应度函数后，对种群中的每个个体计算其适应度。

（3）选择。根据上一步计算出的个体的适应度，按照一定的规则或方法（一般为适应度高的个体），选择优良个体遗传到下一代群体。

（4）交叉。将交叉算子作用于群体，对选中的成对个体，以某一概率交换它们之间的部分染色体，产生新的个体。

（5）变异。随机选择群体中的某个个体，对选中的个体，以某一概率改变某一个或某一些基因值为其他的等位基因。

（6）终止条件判断。达到预先设置的迭代次数后，此进化过程中所得到的具有最大适应度的个体作为最优解输出，终止计算。

下图显示了遗传算法流程的主要阶段：

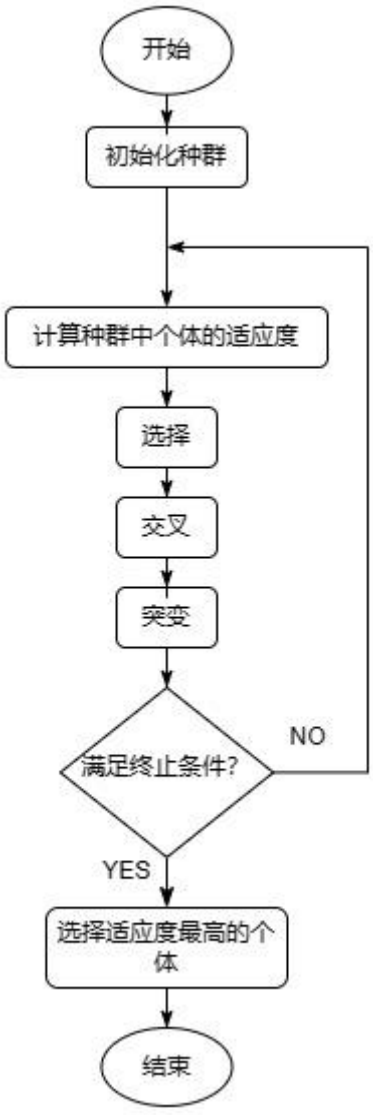

图 2.1 遗传算法流程图

3 基于 GA-XGBoost 算法的选股策略构建

3.1 数据获取

本文所使用的原始数据来源于 Baostock（证券宝）证券数据平台，分析对象为沪深 300 指数成分股，数据涉及的时间区间为 2017 年 1 月 1 日至 2022 年 1 月 7 日，包含期间全部交易日的研究对象 K 线数据，频率为日频。

沪深 300 指数规定了严格的成分股入选条件，不仅剔除了 ST 股票、暂停上市股票，以及经营异常或严重亏损的公司，而且只允许上市超过三个月的公司股票入选成分股，因此在本文的研究中，研究对象不包含这类股票。另外沪深 300 指数在编制过程中对企业流动性赋予较大的权重，反映了规模较大且流动性强的行业代表性公司的股价综合变动。这意味着公司的股价不易受个别投资主体操纵，是多方博弈、供需平衡的结果。本文是关于投资策略的研究，基于策略可行性的角度，研究集中于这类流通市值大、交易相对活跃的公司，是有积极意义的。

此外，沪深 300 指数成分股每年 6 月和 12 月调整一次，策略的研究对象需要根据当前日期的沪深 300 指数成分股进行调整。

3.2 数据处理

3.2.1 异常值处理

在机器学习任务中，要密切注意异常值。异常值是指数据集中远远超出数据整体特征的值。异常值的存在会使数据发生偏移，增加误差差异，从而降低模型整体的准确性。本文将未来 30 日收益率高于 100%的股票样本定义为异常样本，在处理时直接删除该样本。

为了方便说明，以某一期对异常数据的处理为例。在本期中，首先观察收益率的整体分布情况：

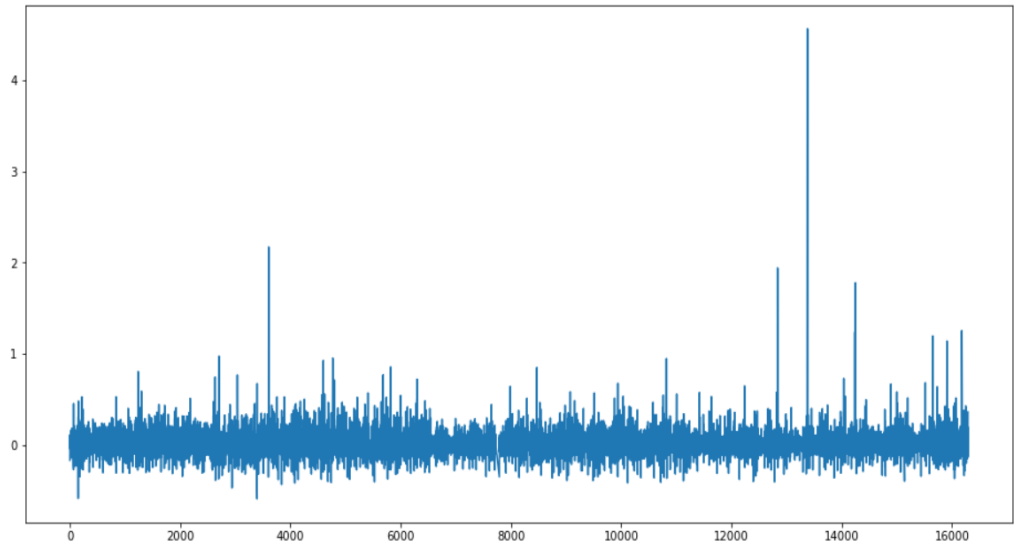

图 3.1 原始数据分布

根据图3.1所示，大部分的股票收益率数据集中在-100%到100%的区间内，仅有少部分数据超过了 100%，为了避免这类数据影响模型训练效果，我们选择删除这部分数据，删除后的结果如下图所示：

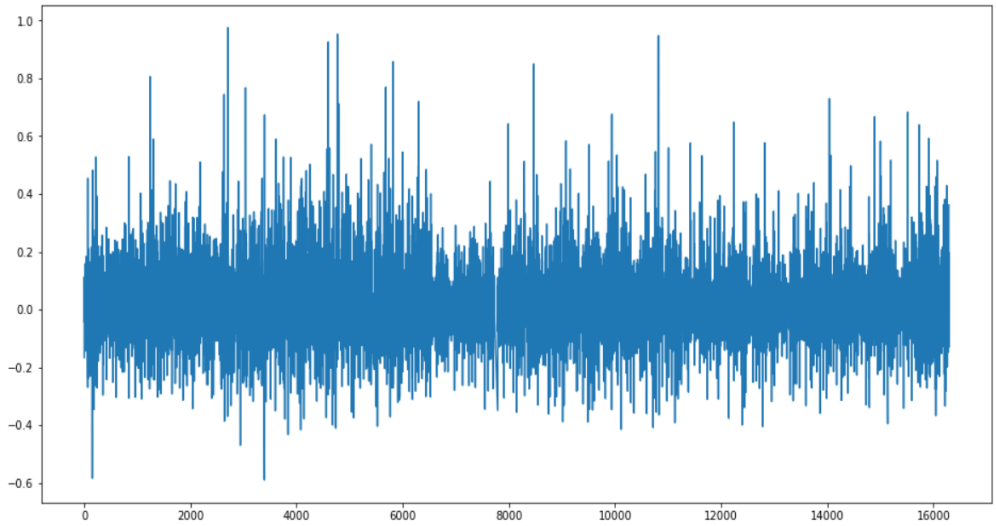

图 3.2 处理后数据分布

3.2.2 标准化处理

在对数据的处理过程中，标准化处理是统计学上常用的处理方法，可以消除不同数据指标之间量纲的差异，减小样本内数据的方差。在对股票因子的分析中，具有不同量纲的因子之间的差别可能很大，例如交易量和交易额指标的数据规模可能是千万级或亿级，KDJ 指标的取值范围在 0 至 100 之间，如果将

这两个因子放在一起构建模型，它们之间的巨大差距将会使结果产生难以预估的偏移。因此，在模型训练前需要将数据进行标准化处理，以消除不同指标间量纲不同影响。本文所使用的标准化处理方法有：

（1）离差标准化

离差标准化是对原始数据的线性变换，将数据映射到[0,1]之间。公式如下：

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

离差标准化保留了原来数据中存在的关系，是消除量纲和数据取值范围影响的最简单方法。

（2）Z-Score 标准化

Z-Score 标准化是数据处理的一种常用方法，基于原始数据的均值和标准差进行数据的标准化。通过它能够将不同量级的数据转化为统一量度的 Z-Score 分值进行比较。公式如下：

$$x^* = \frac{x - \mu}{\delta}$$

其中 μ 为 x 的平均值，δ 为 x 的标准差。

Z-Score 标准化消除了数据量纲的影响，使衡量不同单位的因子可以在同一范围内进行比较。

3.2.3 缺失值处理

在原始数据中，数据缺失的现象是很常见的，比如股票的短期停牌会导致当日的交易数据缺失，或者在获取数据时由于人的主观失误导致的数据缺失。一般情况下，对于缺失的数据会选择直接删除、均值插补、极大似然估计插补等方法进行处理。本文所使用的 XGBoost 算法其选取的基分类器是树模型，因此可以自行处理缺失值。

XGBoost 算法在处理特征缺失的数据时，会将缺失值分别划分到树模型的左子树和右子树，分别计算出左子树和右子树的分裂损失，最后选出损失更大的方向作为分裂方向，这种方法在训练和预测过程中均适用。

3.3 构建因子池

量化选股模型的超额收益主要来源于因子，因子的选择和构建极为重要，直接决定了最终策略的优劣。影响股价的因素（因子）是多种多样的，在以往的研究中已有数以万计的因子被挖掘出来，其数字至今还在不断增加，而其中大部分因子都可以归为以下三类：财务因子、技术指标因子和宏观因子。

财务因子反映公司的实际经营情况，比如负债状况、盈利能力、现金流状况等，财务因子包括净资产收益率（ROE）、资产负债率、市净率等指标，主要通过对公司财务报表公布的相关信息计算获得。虽然这些指标在一定程度上影响了公司股价，但是它们有着更新频率较低的特点，多为季度更新，而本文的原始数据频率为日频，如果将财务因子纳入因子池，可能会出现因子数值在大部分日期都没有变化的问题，可能导致模型的最终结果不显著。另外，财务因子的数据大都来源于会计财务报告，包含了会计报表、会计报表附录以及财务说明书等，这些资料往往具有一定的局限性。首先，财务报表的真实性难以保证，财务报表由会计事务所的会计师审计后得到， 由于审计方法的不同，往往结果也不尽相同，所以，会计事务所出具的审计报告质量高低很大程度上取决于其是否经过了有效的监督；其次，财务指标具有滞后性。企业的财务会计报告通常是在一个会计年度结束后的 3-4 个月公示，所以这就造成了企业的会计报告与企业实际经营的财务状况发生了时间的偏差。这些问题的存在，导致我们不能及时获取真实的财务数据，因此本文的因子池中不包括财务因子。

宏观因子是影响公司企业生产经营的外部因素，如国民生产总值（GDP）、物价水平（CPI）、货币供应量、信贷规模状况等。宏观因子作为整体变量，在研究预测经济周期、宏观资产配置等方面比较重要，虽然宏观因子的变动会对股票的股价有不同程度的影响，但其影响通常是同向且存在时滞的。本文研究的是选股分类模型，宏观因子同样不适合本文的研究。

因此模型聚焦于技术指标因子，这类因子来源于股票交易中的真实数据，根据市场行情中股价、交易量、换手率等指标计算获得，可以充分反映市场行为，用以推断价格变动趋势。技术指标因子的更新频率高，数据来源真实准确，数据获取相对较为容易，比较适合本文的研究。另外，李斌（2019）在对异象因子应用于机器学习算法的研究发现，各类学习算法均对交易摩擦类因子存在

明显偏好，基于财务报表数据构建的因子预测能力较弱，这进一步佐证了在因子池中只纳入技术指标因子的构想。

本文所使用的全部因子均来自于 TA-LIB 金融量化分析库。TA-LIB 全称（Technical Analysis Library）是 Python 金融量化的高级库，涵盖了 8 个大类共 150 多种技术指标，包括股票、期货交易软件中常用的技术分析指标。本文选取了其中 6 大类共 31 个技术指标作为因子池，具体因子如下表所示：

表 3.1 模型因子池

因子分类	因子代码	因子全称	因子释义
重叠指标	BBANDS	Bollinger Bands	布林带
	MA	Moving Average	移动平均线
	WMA	Weighted Moving Average	加权移动平均线
	EMA	Exponential Moving Average	指数移动平均线
	T3	Triple Exponential Moving Average	三重指数移动平均线
	SMA	Simple Moving Average	简单移动平均线
	KAMA	Kaufman Adaptive Moving Average	考夫曼的自适应移动平均线
	SAR	Stop and Reverse	抛物线指标
	DEMA	Double Exponential Moving Average	双移动平均线
动量指标	WILLR	Williams' %R	威廉指标
	RSI	Relative Strength Index	相对强弱指标
	MOM	Momentum	上升动量值
	PPO	Percentage Price Oscillator	价格震荡百分比指数
	ULTOSC	Ultimate Oscillator	终极波动指标
	MFI	Money Flow Index	资金流量指标
	MACD	Moving Average Convergence Divergence	异同移动平均线
	DX	Directional Movement Index	趋向指标
	CMO	Chande Momentum Oscillator	钱德动量摆动指标
	BOP	Balance Of Power	均势指标
	ADX	Average Directional	平均趋向指数

		Movement Index	
	CCI	Commodity Channel Index	顺势指标
	STOCH	Stochastic	随机指标
波动率指标	ATR	Average True Range	真实波动幅度均值
	NATR	Normalized Average True Range	归一化波动幅度均值
统计函数指标	VAR	Variance	方差
	STDDEV	Standard Deviation	标准偏差
	BETA	Beta	贝塔系数
周期指标	HT_DCPERIOD	Dominant Cycle Period	主导周期
	HT_DCPHASE	Dominant Cycle Phase	主导循环阶段
成交量指标	AD	Chaikin A/D Line	累积线
	ADOSC	Chaikin A/D Oscillator	震荡指标
	OBV	On Balance Volume	能量潮

3.4 模型超参数优化

3.4.1 超参数优化机理分析

超参数是指模型在开始学习之前设置的参数，它们不能直接从数据中得到。超参数的设置直接影响了模型的拟合能力和泛化能力。

XGBoost 算法的超参数主要分为三类：通用参数、提升器参数和任务参数。通用参数宏观控制模型使用的提升器种类，使用的线程数等。提升器参数控制每一步中提升器的迭代次数、学习率、随机采样比例等，总体控制提升器的复杂程度。本文所涉及的调参，很大程度上都是在调整提升器参数。任务参数控制模型的学习目标，比如预测问题是分类问题还是回归问题，如果是分类问题，目的是二分类还是多分类，均由目标参数所控制。超参数的具体介绍如下表所示：

表 3.2 XGBoost 算法参数

参数类别	参数名	释义
通用参数	Booster	决定模型基于哪种 booster，gbtree 是采用树的结构来运行数据，而 gblinear 是基于线性模型。

	Nthread	使用线程数
提升器参数	N_estimator	生成的最大树的数目，也是最大的迭代次数
	Learning_rate	学习率，表示每一步的迭代步长
	Gamma	指定了节点分裂所需的最小损失函数下降值。
	Subsample	控制对于每棵树，随机采样的比例
	Colsample_bytree	用来控制每棵随机采样的列数的占比
	Min_child_weight	子节点中最小的样本权重和
	Max_depth	树的最大深度，用来控制过拟合
	Lambda	权重的 L2 正则化项
	Alpha	权重的 L1 正则化项
任务函数	Objective	定义学习任务及相应的学习目标

XGBoost算法的超参数有很多，本文的调参过程集中在提升器参数中的7个参数上，在超参数和预测目标之间的关系比较复杂或者是无法显示表达的时候，研究往往需要选取不同的超参数的取值，最终选择使模型预测效果最好的超参数。在以往的研究中，基于不同的参数选取策略，模型调参主要使用两种方法：网格搜索法和贪心算法。

网格搜索法（Grid Search）本质是一种穷举的调参方式，通过循环遍历每个超参数的每一种可能取值，得到每组参数值对应的模型，从而选择拥有最佳性能的参数组合。之所以叫作网格搜索，是因为超参数的取值是离散的，画在图像上就表现为一张二维甚至高维的网，每一组测试的超参数对应着一个个的网格。网格搜索的本质是穷举，穷举便意味着它比较合适较少的参数组合，一旦参数过多，参数寻优的过程将非常耗时耗力。同时，网格搜索法的每一次搜索过程相对独立，无法将已经搜索过的参数有效转化为先验信息，并与后续的最优参数选择（即后验）建立联系。

贪心算法（Greedy Algorithm）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。使用贪心算法得到的是在某种意义上的局部最优解。在参数调优过程中，贪心算法首先令其他参数不变，调整某个参数的取值，找到令模型效果最好的参数值后固定该参数，然后寻找下一个参数的最优值，以此类推，最后获得所有参数的最优组合。通过这种方式，虽然一定程度上提高了模型调参效率，但其解必然是局部最优的，这也是贪心算法的缺点。

对于本文的模型来说，以上两种调参方式都不太适合。一方面 XGBoost 算法的参数很多，网格搜索法仅适用于参数组合较少的情况，若使用该方法对本模型调参，所需要的计算量是难以估计的，效率过低；另一方面，贪心算法虽然计算复杂度低于网格搜索法，同时也考虑了先验信息，但其最终得到的解不一定是最优解，这种方法适合于不需要全局考虑的简单问题。而本文所使用的 XGBoost 模型内部充斥着因子之间复杂的非线性关联，必须以全局视角考虑才能获得最优解，因此本文在调参过程中引入了遗传算法的思想。遗传算法的计算效率很高，不需要遍历每个参数在范围内的所有取值。另外它是一种基于生物遗传规则的随机全局搜索的优化方法，其搜索方式更为灵活，仅需随机初始化若干个种群，并使其在迭代过程中选择、交叉、突变，最终结果将会收敛至最优解。

3.4.2 遗传算法的初始群体设置

在遗传算法中，遗传操作是由众多个体同时进行的，这些个体组成了群体。为了使遗传算法高效运行，尽快搜索出最优的种群个体，需要提高遗传算法的收敛速度，其中一种方式就是设置高质量的初始群体，高质量的初始种群个体可以大幅提高遗传算法的搜索能力。以往的研究中，会使用随机生成的方式初始化种群，比如基于通过“种子值”得到的伪随机数（PRNG）的方法和基于混沌发生器的方法等。而这种随机生成的策略产生出的个体质量也是随机的，难以全部保持在高水平，而且全部的种群个体也不能保证可以覆盖到全部的搜索区域，其结果是算法需要更多迭代次数、更多搜索时间才能像最优方向收敛，降低了搜索效率。

考虑到上述关于随机生成种群的问题，参考以往的研究，结合 XGBoost 模型参数自身的特点和目标问题的特点，本文决定对于不同的参数采取不同的方法。比如，参数基分类器的数量（N_Estimators）取值最小为 10，没有上限，但是该值越大模型也就越容易过拟合。根据以往的经验来看，基分类器的数量的取值往往与数据量的大小有关。在本模型中，平均每期训练所获取的数据条数大概为 6 万条左右，因此在设置初始种群时，可以将其限制在 600-800 的区间内随机生成，这样更有利于遗传算法的收敛。再比如参数 Gamma，代表模型复

杂度的惩罚项，是用来防止过拟合的重要参数。在以往的研究中，发现该参数的最优取值受多方面影响，因此在构建初始种群时，这类参数则是在其取值范围内随机生成。

表 3.3 初始参数生成设置

参数名	初始随机生成区间
N_estimator	600-800
Learning_rate	0.3-0.7
Gamma	0.01-10
Subsample	0.01-1
Colsample_bytree	0.01-1
Max_depth	1-10
Min_child_weight	0.01-10

3.4.3 遗传算法的适应度函数选择

在生物的进化过程中，适应能力强的个体往往会在恶劣的环境中存活下去并将优良基因保留下来，而适应能力弱的个体正好相反，个体对环境适应能力的强弱由适应度来体现。在遗传算法中，通过适应度函数（Fitness Function）来评价群体中个体的优劣程度，因此适应度也叫评价函数。适应度函数运用在迭代中的选择过程，经过选择，适应度较低的个体被淘汰，适应度较高的个体被保留，然后被保留的个体进行交叉、变异等操作产生子代。

选取合适的适应度函数，可以提高遗传算法的收敛速度，增加搜索效率，同时也能保证算法能够求得最优解。由于遗传算法在进化搜索中基本不利用外部信息，仅以适应度函数为依据，利用种群每个个体的适应度来进行搜索，因此适应度函数在设计上应尽量简单，计算复杂度小。

本文所设计的投资策略是基于机器学习算法的预测模型，显然预测的准确率是一个非常重要的指标，直接体现了模型的预测能力强弱。在信息检索和统计学分类领域，为了评价预测结果的质量，往往采用准确率（Precision Rate）和召回率（Recall Rate）这两个指标。准确率表示预测正确的样本占所有样本的比例，召回率表示在所有正确样本中，被正确预测的比例。然而，准确率和召回率在一定条件下会出现矛盾的情况，一方的增加会导致另一方的降低。

为了综合考虑，比较常见的方法是使用 F1 分数。F1 分数（F1-Score），又称平衡 F 分数，被定义为准确率和召回率的调和平均数，公式如下：

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

本文选择 F1 分数作为遗传算法的适应度函数，F1 分数越接近 1，表明模型拟合效果越好。在多分类任务中，准确率的计算方式和二分类任务类似，召回率则可以使用加权平均的方式计算，权重为各类别样本占总样本的比例。

3.5 标签设置和评分设置

在一个机器学习任务中，标签表示要预测的目标，标签如何设置同样影响着模型的预测能力。作为一个量化选股模型，其首要任务是筛选出上涨概率最大和上涨幅度最大的股票，而并不关心未来股价的具体数值。从以往的经验来看，对于股价的预测往往是一个从不确定性中寻找确定性的过程，通过将收益率分组来设置标签恰好符合这一理念。

因此，本文将股票未来 30 日的收益率按一定条件分组，训练模型得到的预测结果为某只股票未来收益可能处在的区间。具体方案如表 3.4 所示：

表 3.4 标签设置方案

未来 30 日收益率（R）	标签
R＜-10%	1
-10%＜R≤10%	2
10%＜R≤30%	3
R≥30%	4

如果我们预测某两只股票都属于标签 4，即未来 30 日收益均在 30%以上，那么如何区分哪只股票更好更值得买入呢？在 XGBoost 算法中有一个参数 Objective，它决定了模型的学习目标，将其设置为 Objective：“multi：softprob”后，模型将会输出一个概率矩阵，表示目标分类为某个标签的概率值。我们以概率作为权重，乘上每个标签赋予的分值再求和，便能得到最终的评价分数。分数更高则表示该股票 30 天后在我们的股票池中有更大的概率表现更好。

评分设置方案如表 3.5 所示：

表 3.5 评分设置方案

所属标签	分值
1	0
2	1
3	2
4	3

3.6 模型训练方式

以往有关机器学习在量化投资领域的研究中，一部分研究员会采用大量学习的方式来训练模型，认为使用的历史数据越久远越全面，模型的预测效果便越好，事实上这二者并无直接关系。一方面，庞大的数据在实际过程中处理起来费时费力，这种处理对设备条件要求很高，如果是在普通的计算机上处理，一次训练可能持续几个小时之久；另一方面，股票市场风格频繁切换，原始数据本身就在不断变化，那么通过对数据解析而获得的预测模型也应该是不断更新并与时俱进的。

没有一种模型能够在市场中屹立不倒并持续有效，其背后存在着“概念漂移（Concept Drift）”的技术原因，简单来说就是模型要预测的目标变量，随着时间的推移，可能会发生模型无法预知的改变。比如 2019 年之前的任何模型都不可能准确预测口罩销量的曲线。为了尽量避免策略失效，我们可以定期更新训练集数据，也就是通过滚动训练的方式更新预测模型以适应最新市场行情的变化。相较于其他训练方法，滚动训练不仅保留了数据中的时间序列特征，同时也帮助模型捕获当前市场风格，这种方式更符合现实的投资策略逻辑。

因此，本文将训练区间设置为目标日期前 210 天至前 30 天，共计 180 天的数据周期，训练区间随调仓日期的变化而变化。在沪深 300 指数成分股的 300 支股票中，每支股票都有至少 150 个交易日的交易数据，每期总计 40000 个以上的样本，足够一次训练的样本数量要求。训练区间终点设置为目标日期前 30 天是因为模型的预测目标是未来 30 日的收益率，可以避免在训练过程中使用到未来数据。

采用滚动训练方式的另一个原因和本文因子库的构建方式有关，本文所采

用的全部因子均为技术性因子，这类因子的重要特点是有效期短，因此选择时间跨度相对较小的数据集作为训练数据比较合理。

3.7 GA-XGBoost 选股模型整体流程

GA-XGBoost 选股模型的整体思路为：通过遗传算法处理为原始 XGBoost 模型提供参数寻优支持，将原始的股票交易数据导入训练后得到最终预测模型，然后根据新的股票交易数据预测其未来价格走势，最后通过评分系统选择评分最优的股票加入股票池。

下图为 GA-XGBoost 选股模型技术路线图：

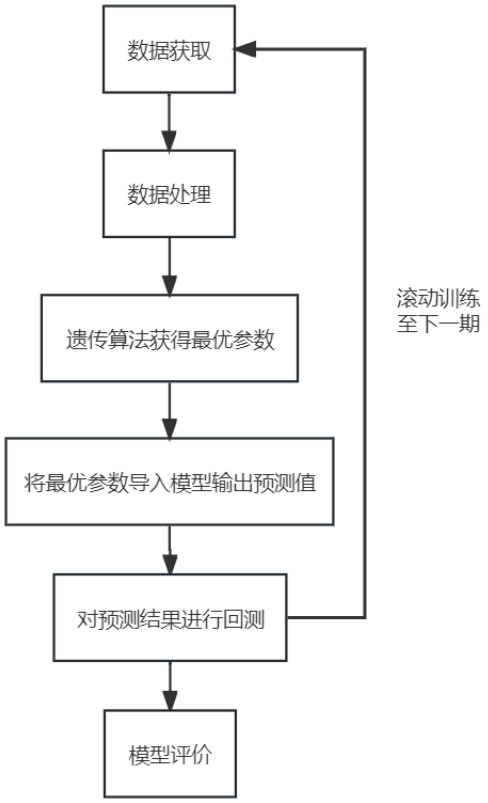

图 3.3 GA-XGBoost 模型技术路线图

以某一期的训练过程为例：

（1）首先，向模型输入当天日期，模型根据日期获取当前沪深 300 指数成分股列表，再根据列表获取原始 K 线数据；

（2）然后对数据进行异常值处理和标准化处理，同时计算模型所需因子；

（3）完成数据获取和处理后，使用遗传算法进行多次迭代直至到达初始设

置的迭代次数，输出迭代过程中表现最优的个体（即 F1 分数最高），获取个体的特征（即最优参数组合），结果如下表所示：

表 3.6 某期训练参数结果

参数	最优参数值
Learning_rate	0.22
N_estimators	211
Max_depth	8
Min_child_weight	5.13
Gamma	0.28
Subsample	0.84
Colsample_bytree	0.62

（4）将最优参数导入 XGBoost 模型，输出每只股票对应标签的概率矩阵，再计算股票评价得分，然后将得分排序；

（5）选择得分排名前十的 10 支股票，等权重构建投资组合。由于 A 股市场为单边市场，缺乏健全的做空机制，本模型不对排名靠后的股票做空，重点关注多头部分。计算投资组合未来 30 日回报率作为回测指标记录。本期训练结束，模型进入下一期。

4 基于 GA-XGBoost 的选股策略评价与分析

4.1 模型准确率评价

4.1.1 混淆矩阵和 ROC 曲线

在机器学习中，混淆矩阵是一个误差矩阵，常用来可视化地评估监督机器学习算法的性能。混淆矩阵的一个轴为预测类别，另一个轴为真实类别。将预测为正例的样本称为 Positive，预测为反例的那些样本称为 Negative。而预测为正例的那些样本中如果预测正确了我们称之为 True Positive，如果预测错了我们称之为 False Positive。Negative 同理。如此便可得到一个混淆矩阵：

		真实类别：正	真实类别：负
预测类别	正	TP （预测为正中预测对的数量）	FP （预测为正中预测错的数量）
	负	FN （预测为负中预测错的数量）	TN （预测为负中预测对的数量）

图 4.1 混淆矩阵

混淆矩阵的四个元素分别是 TP（预测正类中预测对的数量）、FP（预测正类中预测错的数量）、FN（预测负类中预测错的数量）、TN（预测负类中预测对的数量）。

ROC 曲线全称为接受者操作特性曲线，ROC 曲线上的不同点反映了在不同阈值下的感受性。在机器学习领域，ROC 曲线被用来评估模型的预测效果。ROC 曲线的纵轴是“真正例率”（TPR），横轴是“假正例率”（FPR），两者分别定义为：

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

其中 TP、FP、FN、TN 分别表示混淆矩阵中的真正例、假正例、假反例、

真反例。

将模型的分类结果和实际值相比较，计算不同阈值下的 TPR 和 FPR，并以他们为横纵坐标画出曲线，就得到了“ROC 曲线”。ROC 曲线越高，表示模型的预测效果越好，如果 ROC 曲线接近 45°线，表示模型的预测效果和随机分类类似。AUC 值表示 ROC 曲线下方的面积，反映了预测模型对样本的排序能力，即预测正例排在负例前的概率。同样，AUC 值越接近 1，表示模型的预测能力越强。

在本模型中，所有股票按照收益率大小被分配至四个类别，对于这样的多分类任务，ROC 的绘制方式为分别对每个类别计算 TPR 和 FPR 后，绘制多条曲线。对所有分类下的 AUC 分数取平均，得到模型总体的 AUC 分数。由于模型是多期滚动训练的，所以我们以某期训练过程绘制的 ROC 曲线为例：

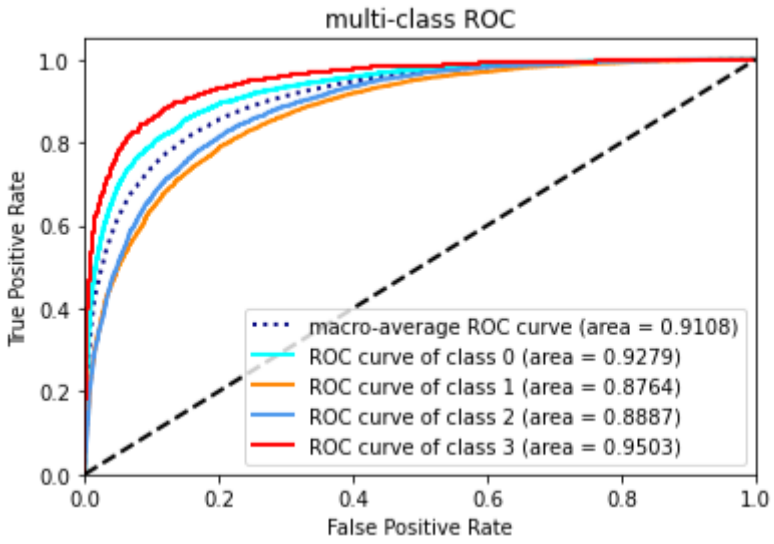

图 4.2 ROC 曲线

如图 4.2 所示，图中每条实线代表该类别下的 ROC 曲线，虚线代表“macro”方法计算出的总 ROC 曲线。可以看到，该期模型的 ROC 曲线总体处在 45°线的上方，证明模型的拟合效果比较好，在数据集内有很高的预测准确率。另外，模型的 AUC 分数为 0.9108，远大于 0.5 的阈值，同样证明了模型的分类效果十分优秀。另外，图中表示对标签 3 分类的 ROC 曲线处在所有曲线的最上方，这表明相较于其他类别，当期模型可以更好的识别出未来 30 天收益大于 30%的股票。

在完整训练 60 期后，获得了 60 个 ROC 曲线和 AUC 分数，将这 60 个 AUC 分数做简单算数平均。结果表明，模型整体的平均 AUC 分数为 0.8323，是一个比较准确的分类模型。

4.1.2 F1-Score 评价

在第三章中，由于准确率和召回率在一定条件下会出现矛盾，为了能够综合考虑这两个指标，本文确定了遗传算法的适应度函数为 F1-Score，即准确率和召回率的调和平均数。F1-Score 作为每次迭代中，子代的适应程度评判标准，那么其也在一定程度上反映了模型的准确率。

表 4.1 多期模型 F1-Score

日期	2017-01-01	2017-01-31	……	2021-12-07	2022-01-07
F1-Score	0.7865	0.7664	……	0.8453	0.8077

上表显示了多期模型训练后计算出的 F1-Score，其中每一个数据都是经过遗传算法多起迭代后保留下来的最优个体所对应的适应度值，即参数组合在当期数据环境下的 F1-Score。F1-Score 越接近 1，证明模型预测越准确。我们对模型 60 期的 F1-Score 做算数平均，得到结果为 0.7906，证明模型的准确率较高，预测效果好。

4.2 策略回测评价

4.2.1 策略回测基本设置

本文所选择的历史回测区间为 2017 年 1 月 1 日至 2022 年 1 月 7 日，调仓周期为每 30 天调仓一次，这么设置的目的是为了与标签设置相对应。另外设置初始本金为 3310.08，即回测第一期时的沪深 300 指数，便于后续比较。

4.2.2 策略回测结果

通过滚动训练的模式，在五年的数据周期内，训练出了 60 个模型，生成了 60 个目标投资组合。我们的研究标的是沪深 300 指数成分股，因此使用沪深

300 指数作为参照基准，回测结果如图 4.3 所示：

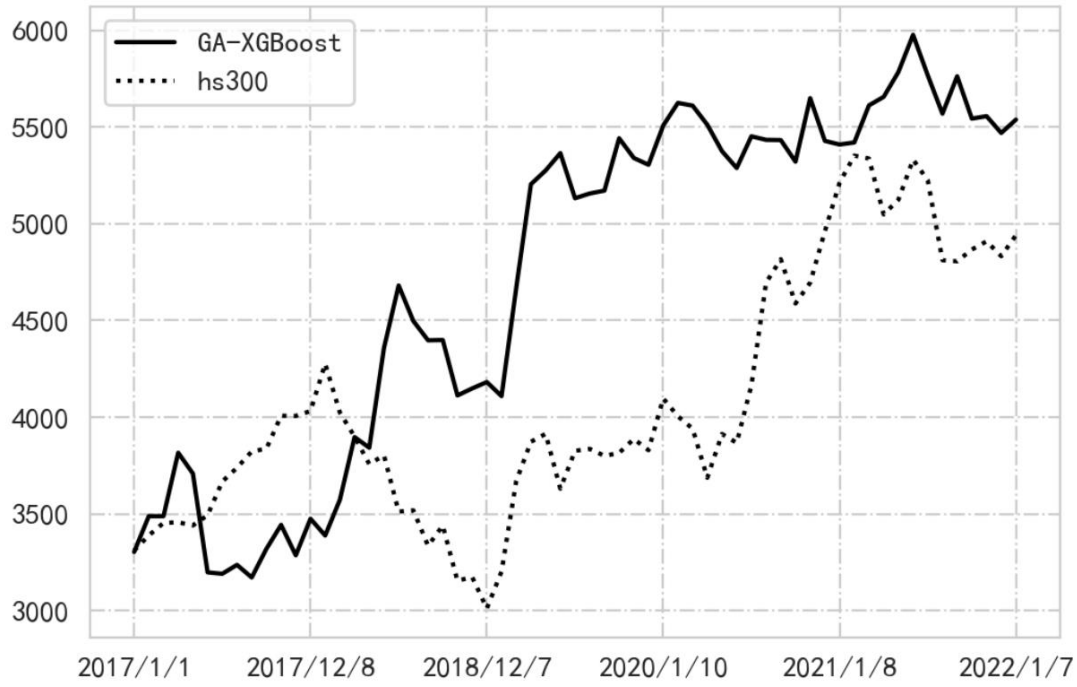

图 4.3 GA-XGBoost 模型回测结果

具体回测数值如表 4.2：

表 4.2 GA-XGBoost 模型回测结果

模型	总收益率	年化收益率	标准差	夏普比率	最大回撤幅度
GA-XGBoost	67.68%	10.89%	4.75%	1.45	12.43%
沪深 300 指数	49.25%	8.33%	4.64%	0.93	31.89%

从表 4.2 可以看到，从 2017 年 1 月 1 日到 2022 年 1 月 7 日，在这 5 年的回测区间内，基于 GA-XGBoost 模型的选股策略的总收益率为 67.68%，年化收益率为 10.89%，同期沪深 300 指数的总收益率为 49.25%，年化收益率为 8.33%，仅从收益率的角度比较，本策略的超额收益为 18.43%。

从风险度量的角度来看，策略多期收益的标准差为 4.75%，最大回撤发生在 2018 年 6 月至 2019 年 1 月，幅度为 12.43%，而沪深 300 指数的最大回撤幅度达到了 31.89%。夏普比率衡量每承担一单位风险能得到的回报，策略的夏普比率为 1.45。索提诺比率与夏普比率类似，衡量投资者每承担一单位下行风险所能得到的超额收益，策略的索提诺比率为 1.78。总体而言，策略的风险水平较低，收益相对稳定。

因此可以认为基于 GA-XGBoost 模型的选股策略基本达到了战胜指数的效

果。首先，策略的收益率是高于指数的，从图 4.3 可以看出，在 2018 年 3 月之后，策略的净值一直处在沪深 300 指数的上方。另外在 60 个投资组合中，有 55%的组合获得了正的收益，有 52%的组合跑赢了沪深 300 指数。其次，策略的稳定性也高于指数。在回测期间，沪深 300 指数的波动率为 25.46%，最大回撤幅度达到了 31.89%，这两项数据都不如我们的选股策略。

策略战胜指数的结果并不使我们感到意外，这说明模型本身具备一定的筛选优质股票的能力，然而在 5 年的时间里最终仅仅收获 18%左右的超额收益，这个成绩低于我们的预期，况且这还是在不考虑投资成本和交易成本的条件下，如果是在现实生活中，这样的结果恐怕难以令投资者满意。分析其原因，本文所构建的投资组合均来自沪深 300 指数成分股，可以说本文的选股是经过沪深 300指数纳入标准筛选后的第二道筛选，然而指数成分股公司中大多都来自传统行业、周期行业，这些公司市值较大、波动率较低，业务发展已经进入成熟期，其业绩相对稳定且增长能力有限，市场预期鲜有分歧，因此他们的股价也很少有剧烈变化。在这样的基础上构建选股策略，实际上是在公认优质的股票中筛选更优质的股票，其难度是很大的，结果就是策略净值难以呈现爆发式增长，只能在风险水平较低的基础上，以稳定趋势小幅增长。

4.3 因子重要性分析

本文所使用的 31 个技术因子，根据其不同的特点可以被分成六大类，分别为重叠指标、动量指标、波动率指标、统计函数指标、周期指标和成交量指标。不同的指标在模型中的实际作用是不同的，XGBoost 算法在训练过程中并不是等权重地使用每个因子，而是根据因子不同的分类效果赋予其不同的权重。

XGBoost 算法提供了三种方法衡量因子重要性：

“Weight”：因子在所有树中作为划分节点的次数。例如，因子在第一棵树用来划分了 3 次，第二棵树用来划分了 4 次等等，那么这个因子的的重要性就是（3+4+……..）。在分裂树的时候，模型选择分裂特征是基于一些如信息增益，信息增益率，gini 系数的评判标准。如果一个特征被选择用来分裂的次数越多，证明它是在树里面是比较优的因子。

“Gain”：因子作为划分节点后的平均增益。Gain 代表了分裂后的信息增益，

Gain 越大，代表着对于该节点来说，下一轮的损失函数越小，如此类推每一个节点都进行相同的操作，全局的损失就会更小。计算信息增益的过程如下：

$$Gain = \frac{1}{2} \cdot \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G_L^2 + G_R^2}{H_L + H_R + \lambda} \right] - \gamma$$

其中，L 和 R 代表的是左子树和右子树，γ代表了模型复杂度。

“Coverage”：因子作为划分节点后对样本的覆盖度。表示在所有树中，某特征在每次分裂节点时处理（覆盖）的所有样例的数量。

在本模型中，由于模型的基分类器是树结构，使用 Gain 增益来衡量因子重要程度是比较合适的。Weight 方法只统计因子作为划分节点的次数，也就是说，默认了每次划分的效果是相同的，所以不能完全体现因子的重要程度，Weight 给予数值特征更高的值，因为它的变数越多，树分裂时可切割的空间越大。所以这个指标，会掩盖掉重要的枚举特征。。而 Coverage 方法，Gain 用到了熵增的概念，它可以方便的找出最直接的特征，因此本模型采用 Gain 的方法来对因子排序。

在 60 期中，对每一期的每个因子的信息增益值取平均值，得到了如图 4.4 所示的结果：

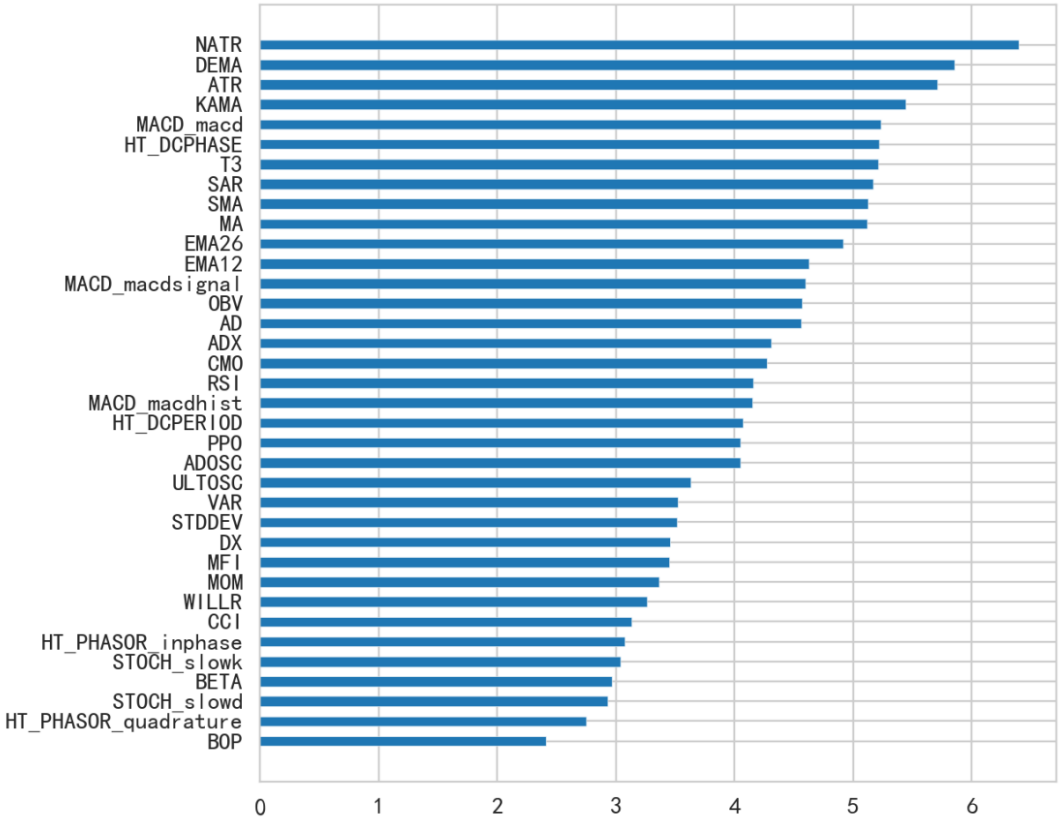

图 4.4 GA-XGBoost 因子重要性

图4.4表明，在31个因子之中，平均因子重要性排名前五的因子为NATR、DEMA、ATR、KAMA、MACD，中文含义为归一化波动幅度均值、双移动平均线、真实波动幅度均值、考夫曼自适应移动平均线和异同移动平均线。

其中排名第一和第三的因子为归一化的波动幅度均值（NATR）和波动幅度均值（ATR），这两个指标都反映了一支股票在一定周期内的真实波动幅度，是 31 个因子中唯二的波动率指标。波动幅度可以显示出交易者的热情，大幅的增加波动幅度表示交易者在当天可能持续买进或卖出股票，波动幅度的减少意味着交易者对该股没有太大的兴趣。波动幅度不仅反映市场的活络程度，还能够帮助交易者预判价格在未来的波动区间，对于设置止损和止盈有很大帮助。波动幅度反映了股票未来上涨或下跌可能的区间，虽然它与股票未来的涨跌幅并非线性关系，但从模型的角度来看，这两个指标包含了更多的市场信息，能够更好的反映市场的短期预期。

双指数移动平均线（DEMA）被赋予如此高的重要性评分，或许和数据本身的特点有关，不过不能否认的是均线类指标在模型分类中充当了重要的角色。从以往的经验来看，均线指标的构造原理决定了它具有反映价格运行趋势的特性，因此其指标可以对价格运行起到趋势跟踪的作用，或许这就是模型所看重的一点。其次均线指标有稳定和滞后的特点，由于均线是对价格收盘价进行算术平均以后产生的新的价格点连线，因此均线相对于价格的变化来说就更为稳定。而正是因为均线系统的稳定和对趋势所具有的这种跟踪特性，相对于价格趋势的变化来说，均线指标又具有一定的滞后性。即当价格趋势已经出现变化的时候，均线指标还会按照惯性再继续维持原来的方向运行一段时间，而不是立即改变运行的方向，这是均线指标的不足之处。而就本文的模型来说，预测目标为 30 日后的收益率，均线指标的稳定滞后性符合模型的偏好。

考夫曼的自适应移动平均线（KAMA）也属于均线类，也就是本文因子分类的重叠指标类。与其他传统的移动平均线不同，KAMA 除了考虑价格波动外，还考虑了市场波动，或许这就是它因子排名靠前的原因。KAMA 在设计之初的目标是解决小而微不足道的价格上涨，即“市场噪音”带来的影响。考夫曼的自适应移动平均线既有均线指标滞后性的特点，又在一定程度上反映了股票的波

动性，可以有效提高模型的泛化能力。

异同移动平均线（MACD）是市场中的经典指标，它从双指数移动平均线（DEMA）发展而来，在本文中被分类成了动量指标。MACD 是由快的指数移动平均线（EMA12）减去慢的指数移动平均线（EMA26）得到快线 DIF，再用 2×（快线 DIF-DIF 的 9 日加权移动均线 DEA）得到 MACD 柱。MACD 的意义和双移动平均线基本相同，即由快、慢均线的离散、聚合表征当前的多空状态和股价可能的发展变化趋势。MACD 的变化代表着市场趋势的变化，不同 K 线级别的 MACD 代表当前级别周期中的买卖趋势。

总体而言，波动率指标归一化的波动幅度均值和真实波动幅度均值在模型分类中有最大贡献，重叠指标双指数移动平均线和考夫曼自适应移动平均线其次，其他类型如周期指标、动量指标、成交量指标和统计函数指标等，重要性不显著。

4.4 策略对比分析

为了证明遗传算法超参数调优后的 XGBoost 模型相较于普通的 XGBoost 模型在选股分类上具有优越性，本文将 GA-XGBoost、XGBoost 和支持向量机（SVM）三个模型基于同样的数据集和时间跨度进行比较，并从超参数调优效率角度、模型准确率角度和历史数据回测表现角度进行分析。

4.4.1 超参数调优效率

首先，从超参数调优效率的角度来看，普通的 XGBoost 模型如果通过网格搜索来获得最优参数，以 7 个参数作为标准，假如每个参数区间有 10 个取值，那么遍历全部组合需要的次数为 7 的 10 次方，况且参数区间的取值往往不只 10 个，有些甚至高达几百个，最终所需要的时间和计算量是难以估计的。对于这种现象，目前大多数研究采用的方式是贪心算法（Greedy Algorithm）。

而遗传算法的参数调优过程则是建立在众多种群并行的基础上并多次迭代的，每次迭代过程中的“交叉”和“变异”赋予了调优的随机性，而“选择”过程又保证了模型整体参数调优方向收敛至最优组合。虽然遗传算法也受计算效率的限制，想要获得最优的参数组合要经过多次迭代，但其收敛速度是远远大于其

他算法的。

由于贪心算法得到的是局部最优解，遗传算法的目标是全局最优解，这二者的计算效率难以在同一标准上比较，本文仅从理论上对二者的计算效率进行分析。贪心算法和遗传算法相较网格搜索而言，均可以大幅提高模型计算效率，但贪心算法本身的局限性，使其在提升效率的同时牺牲了获得最优参数组合可能性。而遗传算法可以说既保证了最优参数组合向最优参数组合收敛，又兼顾了模型的计算效率。

4.4.2 模型准确率

评价模型准确率的指标有 ROC 曲线的 AUC 分数和 F1-Score，现在基于这两个指标对 GA-XGBoost、XGBoost 和支持向量机三个模型进行准确率比较。原始的 XGBoost 和 SVM 均采用网格调参的方式，且只对第一期模型进行调参，后续模型基于第一期的最优参数进行训练。将相同的数据导入上述三个模型，滚动训练 60 期后，计算平均 AUC 分数和平均 F1-Score，结果如表 4.3 所示：

表 4.3 模型准确率比较

模型	平均 AUC 分数	平均 F1-Score
GA-XGBoost	0.8323	0.7906
XGBoost	0.7223	0.7480
支持向量机	0.7561	0.6944

结果显示，GA-XGBoost 的平均 AUC 分数和平均 F1-Score 均高于其他两个模型，说明该模型在测试集上的预测准确率好于网格调参下的原始 XGBoost 模型和支持向量机模型。

4.4.3 历史数据回测表现

下面从历史数据回测的角度来比较这三个模型，同样，为了避免其他因素干扰，本文将相同的原始数据和因子数据导入 GA-XGBoost、XGBoost 和支持向量机三个模型之中，且均采用滚动训练的方式连续训练 60 期，基于相同的回测设置条件，最终得到了如下表所示的回测结果：

表 4.4 回测结果比较

模型	总收益率	年化收益率	标准差	夏普比率	最大回撤率
GA-XGBoost	67.68%	10.89%	4.75%	1.45	12.43%
XGBoost	47.35%	8.06%	4.95%	0.82	11.71%
支持向量机	40.77%	7.07%	4.66%	0.66	26.58%

从表 4.4 可以看到，本文所构建的基于 GA-XGBoost 的选股模型在相同条件下，收益率高于原始 XGBoost 模型和支持向量机模型。在风险控制角度上，三个模型的收益率方差相差不多，最大回撤率虽然略高于传统 XGBoost 模型，但也处于较低水平，总体而言风险控制较好。综合来看，GA-XGBoost 的夏普比率最高，为 0.45，XGBoost 的和支持向量机的夏普比率分别为 0.39 和 0.27。由此可见，遗传算法超参数寻优可以显著提高 XGBoost 模型的分类能力，同时也能提高基于该模型的选股策略绩效。

5 总结与展望

5.1 总结

随着我国 A 股市场的不断完善与发展，传统的投资策略如基本面研究、技术面研究等已经很难使投资者在市场上获得超额收益。近年来，量化投资的概念进入人们视野，量化投资基金的整体规模逐渐增大，市场中量化交易的占比也越来越高。在未来这种新型投资方式将成为市场上主流的交易方式，同时也会成为金融机构争夺客户资源的主要工具。因此，研究量化投资策略对于我国量化投资行业的发展格外重要。

本文在金融量化领域，将遗传算法引入 XGBoost 模型的超参数调优过程，提高了模型的调参效率和预测准确率，通过对历史数据回测，最终获得了超过沪深 300 指数的收益。在选股模型的构建过程中，本文采用设置标签的方式对股票进行分类，设置得分评价方案对预测结果进行排序，同时使用滚动训练的方式来保证模型训练时使用到的是最新数据。在对遗传算法的设计上，使用 F1-score 作为算法的适应度函数。

通过因子重要性分析，本文发现在所使用的 31 个技术指标之中，波动率类指标如归一化的波动幅度均值（NATR）的分类表现最强，其次是重叠类指标如双指数移动平均线（DEMA）、考夫曼的自适应移动平均线（KAMA），周期类指标、成交量类指标等其他指标的分类效果不明显。

5.2 论文存在的不足

（1）在对因子重要性排序，得到重要因子后，未能对因子进行单因子测试，区别本模型筛选出的重要因子是否区别于单因子检验得到的重要因子，同时也未能验证在绩效上是否本模型更优于单因子模型。

（2）标签设置和得分设置方式仅仅根据经验判断，未能比较在不同设置条件下模型的最终效果。

（3）受制于技术条件，本论文所使用的代码仍存在重复、冗余、占用内存

高等不足，还需要对代码继续优化。

5.3 展望

（1）研究对象继续扩展。目前本文的研究对象为沪深 300 指数成分股，希望未来可以扩展至全 A 股的所有股票乃至国外市场的股票，进一步验证模型的预测能力和稳定性。

（2）丰富因子池，增加更多有效因子。做量化研究，因子的选择尤为关键。本文仅选取了部分技术因子，是出于对因子质量的担忧。如果未来可以收集到更多来源准确、更新及时的新因子，引入到模型当中来，也许可以提高模型的预测能力。

（3）与更多机器学习和深度学习算法相结合，构建新型量化投资策略。虽然 XGBoost 算法在以往的研究中已被验证是较为优异的算法，尤其是在量化金融领域，但是随着计算机技术和人工智能技术的发展，新的学习算法如 LightGBM、CatBoost 等相继被提出，它们虽然都是基于 GBDT 框架的改进实现，但仍在效率和准确率上较 XGBoost 有不同程度的提升。如果将其应用至量化投资选股模型，是否会优于以往的策略，以及与遗传算法结合后的效果如何，还需要继续验证。

参考文献

[1] 周志华. 2016. 机器学习[M].清华大学出版社.

[2] 何龙. 2020. 深入理解 XGBoost：高效机器学习算法与进阶[M]. 机械工业出版社.

[3] 李航. 2019. 统计学习方法[M]. 第 2 版.清华大学出版社.

[4] 李斌，邵新月，李玥阳. 2019. 机器学习驱动的基本面量化投资研究[J]. 中国工业经济，8：61-79.

[5] Eric Matthes. 2016. Python 编程：从入门到实践[M]. 人民邮电出版社.

[6] 黄卿. 2018. 机器学习方法在股指期货预测中的应用研究-基于 BP 神经网络、SVM 和 XGBoost 的比较分析[J]. 数学的实践与认识，48(8)：297-307.

[7] 李斌. 2017. ML_TEA_一套基于机器学习和技术分析的量化投资算法 [J]. 系统工程理论与实践，5：1090-1100.

[8] 胡熠，顾明. 2018. 巴菲特的阿尔法：来自中国股票市场的实证研究[J]. 管理世界，2018（8）：41-54.

[9] 王重仁，韩冬梅. 基于超参数优化和集成学习的互联网信贷个人信用评估[J]. 统计与决策，2019，35(01): 87-91.

[10] 方浩文. 2012. 量化投资发展趋势及其对中国的启示[J]. 管理现代化. 2012(05): 3-5.

[11] 赵建. 2011. 基于遗传算法的量化投资策略的优化和决策[J]. 上海管理科学. 2011(10).

[12] Aurélien Géron. 机器学习实战:基于 Scikit-Learn 和 TensorFlow[M]. 机械工业出版社,2018,9.

[13] 邓翔. 2020. 基于量子遗传算法优化的新 Prophet 模型及其验证[J]. 系统工程. 38(5).

[14] 马永杰. 2012. 遗传算法研究进展[J]. 计算机应用研究. 29(4).

[15] 齐岳. 2015. 大数据背景下遗传算法在投资组合优化中的效果研究[J]. 中国管

理科学. 2015: 23.

[16] 李晓寒. 2022. 基于改进遗传算法和图神经网络的股市波动预测方法[J]. 计算机应用. 42(5): 1624-1633.

[17] 全林，姜秀珍，赵俊和，汪东. 2009. 基于 SVM 分类算法的选股研究[J]. 上海交通大学学报，2009，43(09): 1412-1416.

[18] 任君，王建华，王传美，王建祥. 基于 ELSTM-L 模型的股票预测系统[J]. 统计与决策. 2019，35(21).

[19] 潘莉，徐建国. A 股市场的风险与特征因子[J]. 金融研究，2011，(10): 140-154.

[20] 周亮. 影响股票收益的基本面因子略探——基于中小板上市公司的实证分析[J]. 金融理论与实践，2017，(02)，93-98.

[21] 张虎，沈寒蕾，刘晔诚. 基于自注意力神经网络的多因子量化选股问题研究[J]. 数理统计与管理，2020，39(03): 556-570.

[22] 王德明，王莉，张广明. 基于遗传 BP 神经网络的短期风速预测模型[J]. 浙江大学学报（工学版），2012，46(05): 837-841+904.

[23] Fischer，T.，and C. Krauss. Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions[J]. European Journal of Operational Research，2018，270(2): 654-669.

[24] Hou，K.，C. Xue, and L. Zhang. Replicating Anomalies [EB/OL].The Review of Financial Studies，2019，https://doi.org/10.1093/rfs/hhy131.

[25] Jiang，F.，G. Tang，and G. Zhou. Firm Characteristics and Chinese Stocks[J]. Journal of Management Science and Engineering，2019，3（4）：259-84.

[26] Krauss，C.，X. A. Do，and N. Huck. Deep Neural Networks，Gradient-boosted Trees，Random Forests：Statistical Arbitrage on the S&P 500[J]. European Journal of Operational Research，2017，259（2）：689-702.

[27] Gu S, Kelly B, Xiu D, 2020. Empirical asset pricing via machine learning[J]. Review of Financial Studies，33(5): 2223-2273.

[28] William F. Sharpe. 1964. Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk[J]. The Journal of Finance, Vol. 19, No. 3. (Sep.,1964), pp. 425-442.

[29] Rolf W. BANZ. 1981. The Relationship between Return and Market Value of Common Stocks[J]. Journal of Financial Economics 9 (1981) 3-18.

[30] Eugene F. Fama and Kenneth R. French. 1992. Common risk factors in the returns on stocks and bonds[J]. Journal of Financial Economics 33 (1993) 3-56.

[31] Carhart M M, 1997. On persistence in mutual fund performance[J]. Journal of Finance, 52(1): 57-82.

[32] Liu J, Stambaugh R F, Yu Y, 2019. Size and value in China[J]. Journal of Financial Economics, 134(1): 48-69.

[33] C.W.Hsu;C. J. Lin. A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks, 2002, 13(2).

[34] Torlay L ;PerroneBertolotti M ,Thomas E ,Baciu M. Machine learning–XGBoost analysis of language networks to classify patients with epilepsy. Brain Informatics.2017.1.

[35] Kyoung-jae Kim, Ingoo Han. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index[J]. Expert Systems with Applications. 2000. 19(2): 125-132.

[36] McLean，R. D.，and J. Pontiff. Does Academic Research Destroy Stock Return Predictability[J]. The Journal of Finance，2016，71(1): 5-32.

[37] Joseph D P. Value investing: The use of historical financial statement information to separate winners from losers[J]. Journal of Accounting Research，2001，38(2): 1-41.

[38] Huseyin Ince. Support vector machine for regression and applications to financial forecasting[J]. IEEE Computer Society，2000: 6348-6348.

[39] Tianqi Chen, Carlos Guestrin. XGBoost: A Scalable Tree Boosting System[C]. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining，2016.

[40] A. Khodadadi，R. Tütüncü，& P. Zangari. Optimisation and quantitative investment management[J]. Journal of Asset Management，2006，7: 83–92.

[41] Chunchun Chen, Pu Zhang, Yuan Liu, Jun Liu. Financial Quantitative Investment using Convolutional Neural Network and Deep Learning Technology[J].

Neurocomputing，Volume 390，2020，Pages 384-390，ISSN 0925-2312.

[42] Chung H，Shin K S. Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction[J]. Sustainability，2018，10(10).

附 录

部分程序代码：

```
import baostock as bs
import pandas as pd
import numpy as np
import datetime
import talib
from talib import MA_Type
##获取沪深 300##
def get_hs300(target_date):
    rs = bs.query_hs300_stocks(target_date)
    # 打印结果集
    hs300_stocks = []
    while (rs.error_code == '0') & rs.next():
        # 获取一条记录，将记录合并在一起
        hs300_stocks.append(rs.get_row_data())
    result = pd.DataFrame(hs300_stocks, columns=rs.fields)
    return result


def get_start_date(target_date):
    target_date = datetime.datetime.strptime(target_date,"%Y-%m-%d")
    start_date = target_date - datetime.timedelta(days=365)
    start_date = start_date.strftime("%Y-%m-%d")
    return start_date


def get_end_date(target_date):
```

```
    target_date = datetime.datetime.strptime(target_date,"%Y-%m-%d")
    end_date = target_date + datetime.timedelta(days=160)
    end_date = end_date.strftime("%Y-%m-%d")
    return end_date

def get_plus1_date(target_date):
    target_date = datetime.datetime.strptime(target_date,"%Y-%m-%d")
    plus1_date = target_date + datetime.timedelta(days=1)
    plus1_date = plus1_date.strftime("%Y-%m-%d")
    return plus1_date

def get_k_data(stock_code,start_date,end_date):#获取 k 线因子数据
    rs = bs.query_history_k_data_plus(stock_code,
        "date,code,open,high,low,close,volume,amount,turn,pctChg",
        start_date=start_date, end_date=end_date,
        frequency="d", adjustflag="2")
    #print('query_history_k_data_plus respond error_code:'+rs.error_code)
    #print('query_history_k_data_plus respond  error_msg:'+rs.error_msg)

    #### 打印结果集 ####
    data_list = []
    while (rs.error_code == '0') & rs.next():
        # 获取一条记录，将记录合并在一起
        data_list.append(rs.get_row_data())
    result = pd.DataFrame(data_list, columns=rs.fields)
    return result

def get_label(data,period):
    data.pctChg = data.pctChg.astype(float)
    arr = data.pctChg.values
```

```
    pctChg_30 = np.empty([len(arr),1])
    for i in range(pctChg_30.shape[0]):
        if i < (pctChg_30.shape[0]-period):
            q = 0
            for j in range(period): ##30 日后的收益率
                q = ((q/100) + 1)* ((arr[i+j+1] / 100) + 1) * 100 - 100
                j+=1
            pctChg_30[i] = q
        else:
            q = 0
            pctChg_30[i] = q
    data['pctChg_30'] = pctChg_30
    condition = [
    (data['pctChg_30'] <=-10 ),
    (data['pctChg_30'] > -10) & (data['pctChg_30'] <=10 ),
    (data['pctChg_30'] > 10) & (data['pctChg_30'] <= 30),
    (data['pctChg_30'] > 30)
    ]
    value = ['0','1','2','3']
    data['label'] = np.select(condition,value)
    return data
def get_next_date(target_date):
    target_date = datetime.datetime.strptime(target_date,"%Y-%m-%d")
    next_date = target_date + datetime.timedelta(days=30)
    next_date = next_date.strftime("%Y-%m-%d")
    return next_date


def update_cash(cash,LongSet):
    r_yield = LongSet['r_yield'].mean()
    cash = cash * (r_yield/100 + 1)
```

```
    return cash


##### main ######
cash_series = []
date_series = []
fitness_series = []
features = pd.DataFrame()
j = 0
for j in range(20):
    cash_series.append(cash)
    date_series.append(target_date)
    ##print('login respond error_code:'+lg.error_code)
    ##print('login respond  error_msg:'+lg.error_msg)



    train_set = pd.DataFrame(columns = ['date','code','open','high','low','close','volume','amount','turn','pctChg','pctChg_30'])
    X_pred = pd.DataFrame(columns = ['date','code','open','high','low','close','volume','amount','turn','pctChg','pctChg_30'])
    y_ture = pd.DataFrame(columns = ['date','code','open','high','low','close','volume','amount','turn','pctChg','pctChg_30'])
    #hs300_k = get_label(hs300_k,30)
    lg = bs.login()
    hs300_list = get_hs300(target_date)
    end_date = get_end_date(target_date)
    start_date = get_start_date(target_date)


    test_data = get_k_data('sh.000001',start_date,end_date)
    judge_date = test_data['date'].values.tolist()
```

```
        if target_date in judge_date:
            i=1
            for stock_code in hs300_list.code:
                raw_data = get_k_data(stock_code,start_date,end_date)
                #RawData.dropna(inplace=True)
                raw_data = get_factors(raw_data)

                #X_pred =
pd.concat([X_pred,RawData[RawData['date']==target_date]],axis=0)

                target_row = raw_data.set_index('date').index.get_loc(target_date)
                raw_data_30 = get_label(raw_data,30)
                X_pred =
pd.concat([X_pred,raw_data_30.iloc[target_row:(target_row+1)]],axis=0)
                y_ture =
pd.concat([y_ture,raw_data_30.iloc[(target_row+29):(target_row+30)]],axis=0)

                train_set = pd.concat([train_set,raw_data_30.iloc[0:(target_row-
30)]],axis=0)


                #print(f'正在获取第{i}个股票数据')
                i += 1
            bs.logout()
            train_set.to_csv(f'G:\\data\\train_set.csv',index=False)
            X_pred.to_csv(f'G:\\data\\X_pred.csv', index=False)
            hs300_list.to_csv(f'G:\\data\\hs300_list.csv', index=False)
            %run GA.py
            cash = update_cash(cash,LongSet)
            target_date = get_next_date(target_date)
```

```
        fitness_series.append(fitness[bestFitnessIndex])
        features = pd.concat([features,features_score],axis=0)

        j += 1
        print(f'第{j}期训练完毕')
        print(cash,target_date)

    else:
        target_date = get_plus1_date(target_date)
```

致 谢

时光荏苒，研究生的学习生涯已进入尾声。衷心感谢我的导师，对我的论文提供耐心的指导。非常感谢我的同学和同门，每个人都心怀善意且有求必应。当然还要感谢我的父母，永远给予我最大的支持让我轻松面对生活。毕业只是开始，人生还将继续，请鼓起勇气过好每一个当下，积极拥抱未来。

愿每个人都能找到自己的理想并为之奋斗一生。