2023 年第四届“大湾区杯”粤港澳金融数学建模竞赛

题 目

基于中国特色估值体系的股票模型分析和投资策略

摘　　要:

本文立足于中国资本市场的独特性，尝试构建一个反映中国特色估值体系（中特估）的股票模型。这一模型旨在识别并量化那些符合中特估特征的股票，进而形成有效的投资策略。在现有的研究和实践基础上，我们综合了政策背景、市场定位和专家解析等多维度信息，以确定能够代表中特估的关键特征指标。

针对任务一，我们通过深入分析中国特色的资本市场定位和政策背景，以及从专家解析中抽取的智慧，构建了中特估股票的五个关键特征指标：政策支持度、自主创新能力、系统性风险度、市场重估度、和社会责任度。这些指标融合了定性和定量分析，确立了中特估特征指标体系。我们采用支持向量机（SVM）算法，对沪深 A 股的股票进行了的二元分类，以此量化地描绘中特估股票的特征画像，并解释什么是中特估股票。

针对任务二，利用上述特征指标，我们运用层次聚类（Hierarchical Classification）算法对沪深 A 股市场中的中特估股票进行了分类。这一过程揭示了不同分类股票的投资特点，为投资者提供了明确的投资方向。分类结果展现了各类股票在政策支持度、自主创新能力、系统性风险度、市场重估度、和社会责任度等方面的异同，帮助投资者把握投资重点，对短期和长期投资战略都具有指导意义。

针对任务三，我们结合市场热点，如价值投资、资产重组、国际环境和舆论影响等，设计了基于中特估的短期股票投资组合。通过应用马科维茨均值-方差模型，我们对每一支中特估股票进行了权重赋予，构建了一个旨在实现风险最小化的同时期望收益最大化的短期投资组合。在真实数据上的实测表明，我们的短期组合能够在不同市场条件下保持稳健，展现了优异的适应性和潜在的盈利能力。

针对任务四，我们设计了一个长期股票投资组合模型，再次采用马科维茨均值-方差模型，不仅考虑了股票的预期收益率和波动率，还兼顾了各股票间的相关性。长期投资组合的设计侧重于稳定增长和抵御市场波动的能力。通过对长期投资组合的回测分析，我们发现该策略在不同的市场周期中能有效分散风险，同时提供了与市场平均水平相比更为可观的回报。这些发现证明了中特估股票特征指标体系在实际投资中的应用价值和有效性。

关键词：支持向量机 层次聚类 马科维茨均值-方差模型 中特估模型

目录

1 问题背景与问题重述

1.1 问题背景

证券投资的核心目标是获取收益和规避风险，如何有效评估证券在市场交易中的价值，是进行证券投资的基本问题。在股市中，常用的估值模型各具特点，已经在实际的股票投资中得到了应用和检验。然而，投资者若想在中国股票市场有所收获，必须将股票估值模型和市场的背景结合起来进行综合研判，以获取客观准确的价格估值，形成有效的投资策略。2022 年下半年，证监会党委书记、主席易会满在《求是》杂志发表文章提出，努力建设中国特色现代资本市场。2022 年 11 月 21 日，在金融街论坛年会上，易会满表示，需要对中国特色现代资本市场的基本内涵、实现路径、重点任务深入系统思考。要把握好不同类型上市公司的估值逻辑，探索建立具有中国特色的估值体系，促进市场资源配置功能更好发挥。因此，本文将探索对于股票证券市场，中国特色估值体系的主要特色和核心内涵。

1.2 问题重述

任务一：现阶段中国特色估值体系虽然已经有比较明确的政策背景和清晰的资本市场定位，但急需构建中国特色估值体系的模型指标特征。我们需要基于中特估概念的政策背景，市场定位和专家解析这三个方面，构建出中特估股票的特征指标，然后用特征指标刻画出中特估股票的画像，从而回答中特估股票是什么的问题。

任务二：根据任务一中建立的模型特征指标，将沪深 A 股（或者限制一个范围，例如大湾区）证券市场的中特估股票进行分类，并分析每一类股票的投资特点。

任务三：证券市场周围的环境很大程度影响了市场的行为，其中经济环境的热点是影响股票走势的最敏感因素。针对中特估股票的模型特征，结合典型的市场热点，如：价值投资，资产重组，国际环境和舆论影响等热点，设计一个基于中特估的短期股票投资组合，并进行实测。

任务四：基于任务一构建的沪深 A 股中特估的股票特征指标，设计一个长期股票投资组合模型，并分析该投资组合的收益。

这些任务旨在探索和实现一个与中国股票市场环境相适应的股票估值和投资策略。

2 问题分析

我们面临的核心问题是如何构建一个能够反映中国特色估值体系的股票模型。这个模型需要捕捉到中特估股票的关键特征指标，并基于已有的数据将其量化为一系列可以计算和比较的指标。我们将可以公开查阅的政策背景，市场定位，专家解析，和成型单项指标进行综合研判，构建出中特估特征指标。

将这些定性和定量指标结合起来确立中特估特征指标后，我们将利用支持向量机（SVM）算法对沪深 A 股中所有股票进行二元分类，来量化描绘中特估股票的画像。我们将根据画像对中特估进行解释，有助于投资者理解什么是中特估股票，并为其投资决策提供依据。

在后续的任务中，我们将使用这些特征指标应用层次聚类（Hierarchical Classification）算法来分类沪深 A 股市场中的中特估股票，并分析归纳每一类股票的投资特点。在投资组合的选择上，我们将采用马科维茨均值-方差模型（Markowitz Model of Portfolio）赋予每一支股票的权重。随后，根据每一类股票的特点以及结合经济环境的热点信息，我们将分别设计出短期和长期投资组合，并在已有数据上进行实测。通过对收益的分析，我们将并评估短期和长期投资组合的表现。最后将对

我们建立的数学模型进行整体评价，挖掘优点和进步空间，以及分析模型在其他方面的可能推广应用。

此处，我们给出分析问题详细步骤的流程图。

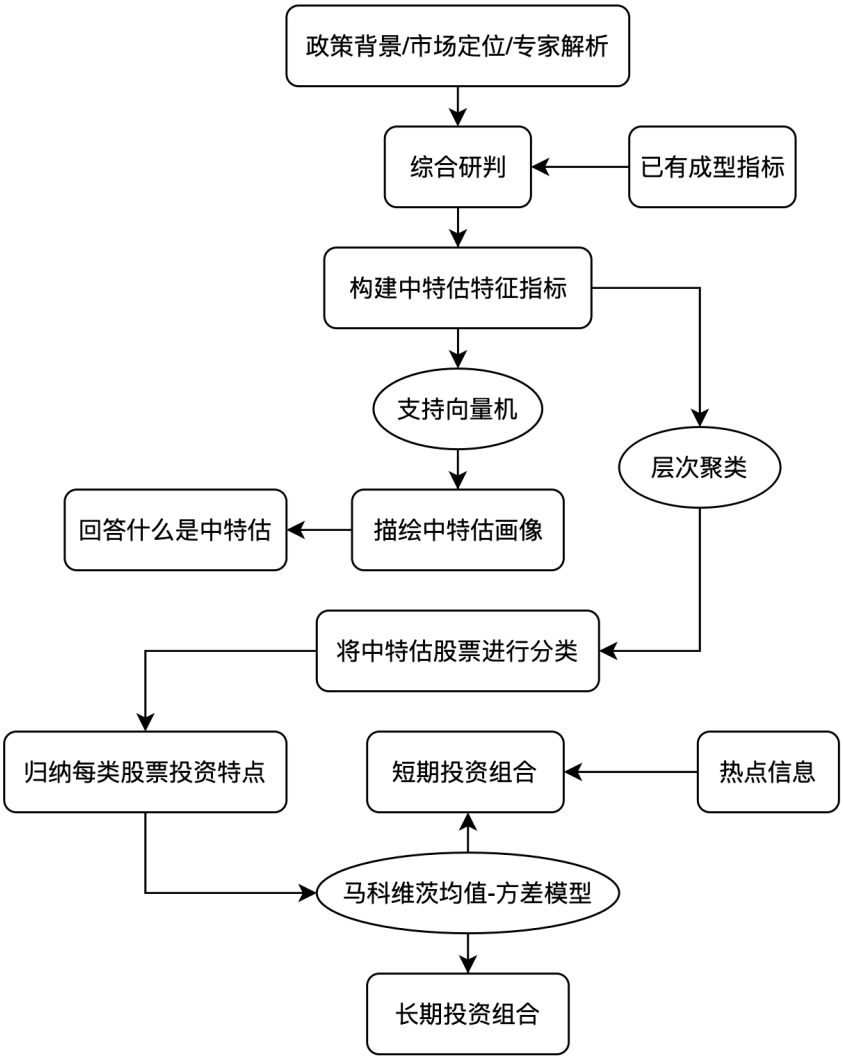

图 1: 问题分析流程图

3 模型假设

1. 股票交易默认 T+0 制度。
2. 所有交易下单即迅速全部成交。
3. 热点事件窗口期内的市场波动不受热点事件外的事件影响。
4. 忽略市场摩擦，如税收、交易费用、价格影响。
5. 资产收益服从正态分布，投资者的决策仅基于收益率的期望值和方差。
6. 资产之间的收益率是相关的，通过分散投资可以实现风险的降低。

4 符号说明

符号	说明	单位
X_i	表示第 i 个特征指标的值	-
ω_i	表示第 i 个特征指标的指标系数	-
b	表示支持向量机模型训练后的偏置	-
FR_i	表示第 i 支股票政策关键词出现频次	-
M_i	表示第 i 支股票研发费用	元
β_i	表示第 i 支股票 BETA 值	-
ROE_i	表示第 i 支股票净资产收益率	-
PB_i	表示第 i 支股票市净率	-
ESG_i	表示第 i 支股票同花顺 ESG 综合评分	-
R_i	表示第 i 类股票的年收益率	-
k_i	表示第 i 类股票的投资权重	-

表 1: 符号说明

5 任务一

5.1 中特估股票特征指标

中国证监会主席易会满强调，要针对不同类型的上市公司制定估值体系，暗示国家政策在市场估值中扮演重要角色。这反映出政策制定者意图引导资本流向与国家战略一致的行业和公司，这是政策支持指标的基本组成部分。

此外，创新常被中国政策突出为国家经济转型和竞争力的关键驱动力。这与中国证监会的号召相一致，鼓励上市公司，特别是国有公司，“提高内功”和战略性整合，以增强核心竞争力。这可以构成自主创新能力指标的基础，因为与国家战略一致的公司可能会获得更有利的估值。

再者，在讨论中国估值体系时，系统性风险是重要的，尤其是关于传统上被低估的国有企业和银行。这种低估可能反映了感知到的系统性风险以及它对通过资本市场融资能力的限制。因此，系统性风险指标需要考虑更广泛的经济政策及其对不同行业的影响。

另外，“对‘中国优势资产’进行重估”的概念表明需要识别 A 股市场中被低估的部分，如国企和银行，并在它们对国家经济贡献和政策目标一致性的背景下重新评估它们的价值。这与市场重估度指标相一致，该指标将评估市场根据公司的战略重要性和政策一致性调整其估值的程度。

最后，企业社会责任在全球范围内、特别是在中国日益被认识为公司估值的一个重要因素。这在中国推动更“规范、透明、开放、有活力和有韧性”的资本市场的背景下尤为相关。展现出更高社会责任程度的公司可能会获得更有利的估值，尤其是在强调可持续发展的环境下。

基于中特估概念的政策背景，市场定位和专家解析，我们在此构建中特估股票的五大特征指标：

政策支持度	**自主创新能力**	**系统性风险度**	**市场重估度**	**社会责任度**
FR	$\log(M+1)$	β	ROE − PB	ESG

表 2: 中特估股票五大特征指标

这五大方面融合了中国市场的多面性，其中政策指导、创新、系统稳定性、市场认知和社会责任都相互联系，形成了一个反映中国经济和治理模式独特特征的估值体系。

1. **政策支持度：**政策关键词出现频次（FR）。

   股票的政策导向程度与涉足重点行业领域数量和所参与的重要政策支持计划数量成正相关关系。所以我们在此统计了政策关键词在股票所属概念，所属 GICS 行业，所属国民经济行业，所属中信行业 4 项指标中的出现频次，来衡量股票的政策支持度。

2. **自主创新能力：**对研发费用取 log 函数。

   研发费用的投入直接影响了企业的自主创新能力，投入的研发投入越多，所带来的创新成果也越多。对研发费用取 log 函数是因为我们需要减小企业体量的影响。大型企业的研发投入的值可能很高，但所占总投入比例很小，而小型企业可能研发投入的值不高，但是所占总投入比例很高，这种情况下可以说明小型企业依然是重视创新发展的。所以对研发费用取 log 函数削弱了上述可能性的影响。

3. **系统性风险度：**BETA 值。BETA 值反映了个股的变动与大盘的联动性，是一个用来衡量系统性风险，即不可分散风险的主要指标。通常 BETA 值越大，代表股票或投资组合波动性越高。BETA 值以 1 为分界，BETA 值等于 1 时，代表大盘上涨（下跌）1% 时，个股或投资组合也上涨（下跌）1%。

4. **市场重估度：**净资产收益率（ROE）与市净率（PB）的差值。

   研报显示央企和国企 ROE 优于大多数上市企业，但同时 PB 估值长期处于低位，修复空间大。同时历史数据表明，长周期下 A 股 PB 有向 ROE 回归的趋势。故设置市场重估度为 ROE 与 PB 的差值。

5. **社会责任度：**同花顺 ESG 综合总评分。

   ESG 评分指环境、社会及管治评分，可有效衡量企业的 ESG 风险和表现因子，以财务重要性作为主要关注点。未来，关注经济、社会、环境的可持续发展，推动实现国家经济高质量发展和生态建设目标将成为全球和我国未来经济发展的重要战略内容。减缓气候变化相关等 ESG 产品将为实现经济低碳转型提供重要支持，ESG 投资将面临重要的发展机遇。

5.2 数据收集

在构建好中特估五大特征指标后，我们开始数据收集工作。在同花顺 iFinD 数据终端中，我们首先找出了在沪深 A 股中发行的所有股票，共 5074 支。然后我们从数据库中选取了 9 项指标并提取了数据，其中所属概念，所属 GICS 行业，所属国民经济行业，所属中信行业这 4 项指标为文字描述，其余研发费用，BETA 值，净资产收益率 ROE，市净率 PB，同花顺 ESG 总评分 5 项为量化数据。股票代码和股票名称与 9 项指标共同组成了我们最初收集到的数据集。

5.3 数据预处理

对于第一项特征指标政策支持度，由于所属概念，所属 GICS 行业，所属国民经济行业，所属中信行业这 4 项指标为文字描述，我们需要将他们进行量化以便后续进行分析，在此我们以公司所涉及的重点行业领域数量和所参与的重要政策支持计划数量来量化。我们选取的政策关键词有：金融，石油，电力，通信，钢铁，国防，交通，医药，芯片，新能源，高铁，核电，航空，环保，互联

网，人工智能，大数据，国企改革，一带一路。随后通过所属概念，所属 GICS 行业，所属国民经济行业，所属中信行业这 4 项文字描述中所涉及的政策关键词频次来进行统计来反映政策支持度。

对于第二项特征指标自主创新能力，我们需要采取对数变换的预处理手段。由于不同股票的呀发费用之间的数量级差距比较大，我们对所有股票的研发费用进行取 log 函数操作。变化后的研发费用 $\widehat{M_i}$ 为

$$\widehat{M_i} = \log(M_i + 1)$$

对于第三项特征指标系统性风险度，我们直接采用提取出的 BETA 值数据。但是我们需要进行一步去除异常值的操作。设 P_k 代表 BETA 值的第 k 个百分位数。如果某只股票的 BETA 值小于 P_1 时，则令该股票的 BETA 值等于 P_1，而如果某只股票的 BETA 值大于 P_{99} 时，则令该股票的 BETA 值等于 P_{99}。

对于第四项特征指标市场重估度，我们将提取出的净资产收益率 ROE 与市净率 PB 作差。这里我们同样需要进行去除异常值的操作。设 P_k 代表 ROE−PB 的第 k 个百分位数。如果某只股票 ROE−PB 的值小于 P_1 时，则令该股票 ROE−PB 的值等于 P_1，而如果某只股票 ROE−PB 的值大于 P_{99} 时，则令该股票 ROE−PB 的值等于 P_{99}。

对于第五项特征指标社会责任度，我们直接采用提取出的同花顺 ESG 总评分数据。同样，我们需要进行异常值处理操作。设 P_k 代表 ESG 评分的第 k 个百分位数。如果某只股票的 ESG 评分小于 P_1 时，则令该股票的 BETA 值等于 P_1，而如果某只股票的 ESG 评分大于 P_{99} 时，则令该股票的 ESG 评分等于 P_{99}。

此外，我们需要标注好中特估的数据，我们选用同花顺中特估 100，里面包含了 100 支已经被标注好的沪深 A 股中的中特估股票。将中特估标签添加到数据集最后一栏中，其中 0 代表不是中特估股票，1 代表是中特估股票。

以下是我们处理好的（部分）数据展现：

证券代码	证券名称	政策支持度	自主创新能力	系统性风险度	市场重估度	社会责任度	标签
600000.SH	浦发银行	1	0	0.6306	6.023475	68.7987	0
600004.SH	白云机场	3	7.9016	0.7333	-2.4136	63.2242	1
600006.SH	东风汽车	2	8.6027	1.3922	1.158	69.9393	0
600007.SH	中国国贸	1	0	1.141	11.0616	66.4681	1
600008.SH	首创环保	3	8.3269	0.8888	5.1385	65.4513	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
301548.SZ	崇德科技	1	7.3911	1.2588	5.1905	0	0
301550.SZ	斯菱股份	1	7.5102	1.2582	4.9224	0	0
301555.SZ	惠柏新材	1	7.5704	0.055625	6.3903	0	0
301558.SZ	三态股份	0	7.6175	1.7588	2.4361	0	0
301559.SZ	中集环科	2	8.2211	2.2476	10.8526	0	0

表 3: 预处理好的数据（部分）展现

我们现在通过直方图对我们的特征指标值分布进行可视化，进一步分析一下我们构建的五大指标是否合理。

以下为沪深 A 股股票五大特征指标分布的直方图：

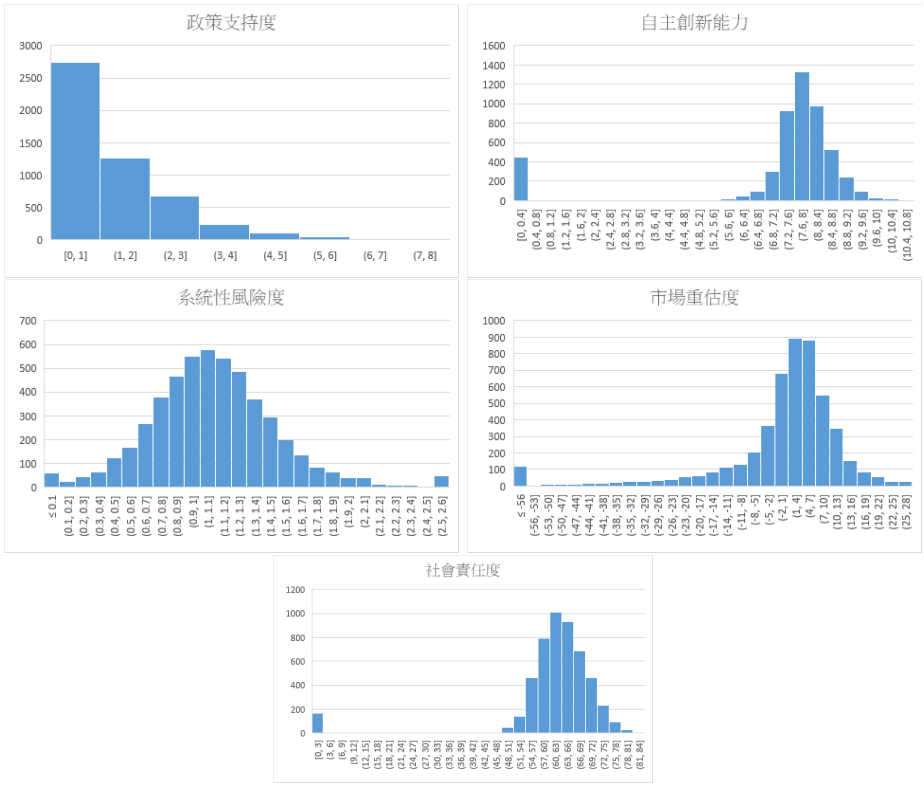

图 2: 沪深 A 股特征指标分布直方图

最后我们要对五项特征指标全部进行标准化，我们以政策支持度为例: μ 和 sd 分别代表所有股票的政策支持度的均值和标准差，则第 i 支股票经过标准化后的政策支持度 $\widehat{FR_i}$ 为

$$\widehat{FR_i} = \frac{FR_i - \mu}{sd}.$$

5.4 中特估股票画像

在收集，处理好数据集之后，我们开始用模型对数据集进行训练。

我们把 5074 支股票划分为训练集与测试集，并使用了支持向量机（SVM）模型和不同的随机种子进行了 10 次训练与测试，得到了不同特征指标的系数 ω_i、偏置 b 和准确率的均值与标准误差。

指标名称	均值	标准误差
政策支持度系数 ω_1	0.832	0.042
自主创新能力系数 ω_2	-0.214	0.052
系统性风险度系数 ω_3	-0.047	0.044
市场重估度系数 ω_4	0.725	0.228
社会责任度系数 ω_5	1.076	0.197
偏置 b	-0.866	0.058
准确率	0.738	0.015

表 4: 不同特征指标的系数 ω_i、偏置 b 和准确率的均值与标准误差

为进一步验证指标系数的正确性，我们分别计算了沪深 A 股特征指标分布，和其中中特估与非中特估的各项平均指标系数，下图是绘制出的直方图:

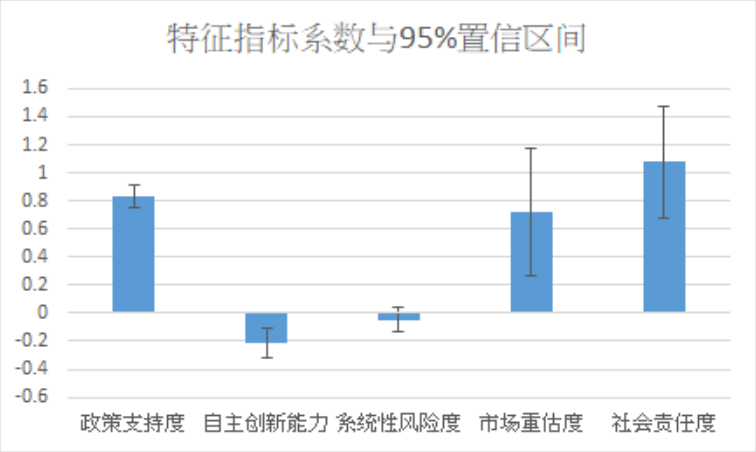

图 3: 沪深 A 股特征指标分布直方图

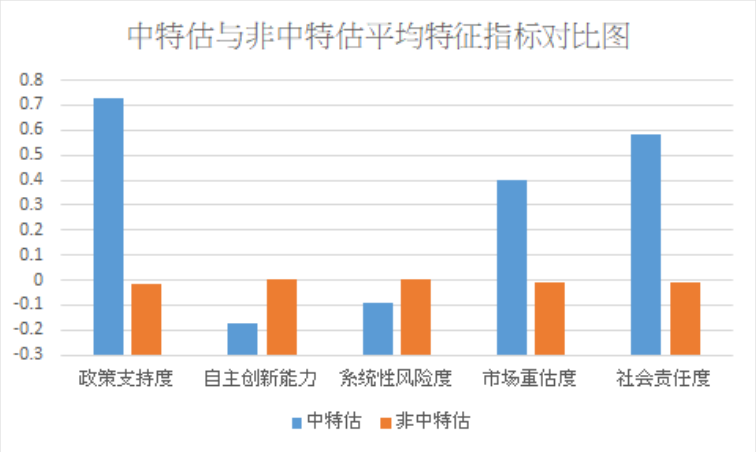

图 4: 中特估与非中特估的五大特征指标平均系数

可以看到，中特估的平均政策支持度、市场重估度和社会责任度均高于非中特估，因此政策支持度、市场重估度和社会责任度高的股票更有可能为中特估股票，这与对应政策支持度、市场重估度和社会责任度的特徵指标系数均大于 0 的结果吻合。

从上表中，我们可以直观看出对于中特估，五大特征指标的影响时非常显著的。因为我们可以用各项平均特征指标系数对中特估画像进行刻画。在此，我们构建中特估股票画像 $f(X)$:

$$\begin{aligned} f(X) &= \sum_{n=1}^{5} \omega_n X_n + b \\ &= \omega_1 X_1 + \omega_2 X_2 + \omega_3 X_3 + \omega_4 X_4 + \omega_5 X_5 + b \\ &= 0.832 X_1 - 0.214 X_2 - 0.047 X_3 + 0.725 X_4 + 1.076 X_5 - 0.866 \end{aligned}$$

因此，每一支股票 (X) 都可以算出对应的 $f(X)$ 的值。当 $f(X) \geq 0$ 时，这支股票将被划分为中特估股票，而当 $f(X) < 0$ 时，这支股票将被归纳为非中特估股票。

5.5 什么是中特估股票

根据训练的 SVM 模型结果，我们已经得到刻画中特估股票画像的 $f(X)$ 值，我们可以将中特估股票概括为一个符合特定社会和市场评价标准的投资目标。这揭示了五个关键的特征指标和它们

对于股票是否被认定为“中特估”类别的重要性。

首先，社会责任度对模型来说是最重要的特征，具有最高的正权重。这表明具有高社会责任感的公司，即那些在环境、社会和公司治理（ESG）方面表现出色的公司，更可能被认定为中特估股票。换句话说，这类股票在其经营活动中不仅追求财务回报，还注重对社会的积极贡献。

其次，市场重估度和政策支持度同样有较高的正权重，说明市场对这些股票的重新评估以及政府政策的支持对它们的中特估属性有着显著的影响。市场重估度高意味着股票的潜在价值可能被市场低估，而政策支持度高反映了公司能够从政府政策中获得优势，这两者都使得股票更有可能获得中特估的标签。

相反，自主创新能力具有负权重，虽然这看起来与直觉不符，但它可能表明在中特估股票的定义中，自主创新能力不是最核心的因素，或者在特定的市场和政策环境中，创新能力并不总是第一位的考量因素。最后，系统性风险度对模型的影响相对较小，表明它在中特估股票的识别中不是一个主要的决定性因素。

综上所述，中特估股票是那些在社会责任上表现出色，受益于市场重估和政策支持的公司，而这些公司的自主创新能力和系统性风险度对其中特估属性的影响相对较小。这样的股票可能代表了一种均衡的投资选择，它们不仅财务表现稳健，还符合社会伦理和政策导向的要求。

6 任务二

6.1 中特估股票分类

基于我们构建的中特估五大特征指标，我们将对中特估股票进行分类。

我们选用层次分类（Hierarchical Classification）模型对 100 只中特估股票进行分类。我们需要确定一个合理的聚类数目。我们先观察层次聚类算法拟合出的树状图：

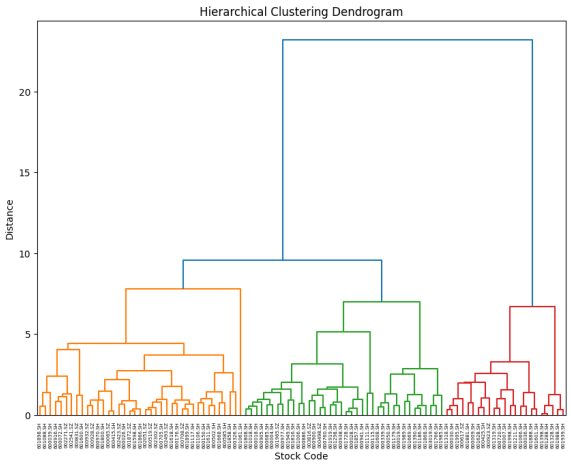

图 5: 层次分类树状图

树状图显示了不同股票之间的距离和关联。在这个图中，每个点代表一个股票，而每个垂直线的合并代表了股票之间的一次聚类。树状图的高度表示聚类合并时的距离，这个距离可以解释为聚类间的不相似性。为了更精确地确定聚类数，接下来我们计算了不同聚类数目的类内平方和（WCSS），并绘制它关于聚类数目的变化图，通常称为“肘部法则”图。这将帮助我们估计聚类数目的合理范围。以下是绘制出的“肘部法则”图：

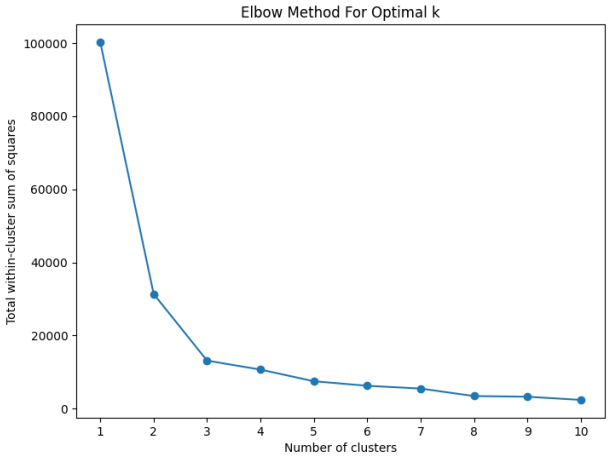

图 6: 聚类数目的类内平方和图

现在我们有了类内平方和（WCSS）相对于聚类数目的变化图。根据“肘部法则”，我们寻找 WCSS 图的肘部点，这个点通常表示增加聚类数目不再显著降低 WCSS 的地方。在这个点之后，增加聚类数量对模型的改进不大，可能导致过度拟合。从图中我们可以看出，在聚类数目为 2 到 4 之间，WCSS 的下降开始变得缓慢，这意味着 2-4 可能是合理的聚类数目范围。在此，我们选择 3 个聚类，这看起来是一个在模型复杂度和解释能力之间取得平衡的好选择。

因此，我们指令层次聚类算法将 100 支中特估股票分成三大类：

聚类 1：包括中国国贸、上海机场、中信证券等，共有 23 只股票。

聚类 2：包括白云机场、上港集团、宝钢股份等，共有 35 只股票。

聚类 3：包括中远海能、四川路桥、中国船舶等，共有 42 只股票。

6.2 中特估股票投资特点

我们根据五大特征指标用箱形图画出中特估股票三大分类的特征指标平均值与 95% 置信区间：

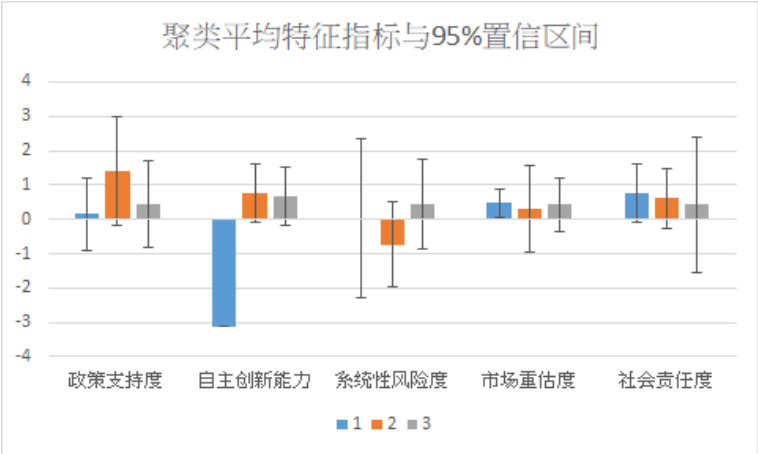

图 7: 聚类特征指标平均值与 95% 置信区间

在分析聚类结果时，我们将会寻找每个聚类内部的共性和不同聚类之间的差异性。我们的分析会基于聚类内股票的数量和五大特征指标来进行。以下是对三个聚类结果所反映出的每一类股票的

投资特点进行归纳分析：

- **潜力型股票**（聚类 1）：这个聚类包括了 23 只股票，主要为银行业和保险业公司，这些公司有较高的市场重估度和社会责任度，因为这些行业通常受益于政策优惠和市场重视。同样，这类公司的自主创新能力可能不如高科技行业，但可能在社会责任度上表现较好，尤其是那些大型的、受到社会监督的金融机构。
- **稳定型股票**（聚类 2）：这个聚类包括了 35 只股票，包括基础设施、通信和部分工业公司，这些公司有较高的政策支持度和自主创新能力，同时有较低的系统性风险度，但市场重估度较低。
- **机遇型股票**（聚类 3）：这个聚类包括了 42 只股票，主要是能源、材料和建筑行业的公司，这些公司有较高的自主创新能力和系统性风险度，但社会责任度较低。

对于潜力型股票，自主创新能力的平均值是最低的，且标准差为 0，表明这个聚类中所有股票在这个指标上的值是相同的，显示出这个聚类的股票在创新能力上非常一致，但较低。系统性风险度的平均值接近 0，但标准差较大，表明这个聚类中股票的市场风险表现差异较大。市场重估度和社会责任度的平均值较高，但社会责任度的标准差较大，说明聚类中的股票在这些方面的表现比较好，但社会责任度的一致性不如市场重估度。政策支持度的平均值相对较低，且标准差表明在这个聚类中股票在政策支持上的表现有一定的波动。

对于稳定型股票，自主创新能力的平均值较高，标准差表明在这个聚类中股票的创新能力有一定的差异。系统性风险度的平均值为负，这可能表明这些股票比市场平均风险低，但标准差相对较大，表明聚类内部在风险度上的差异也较大。市场重估度和社会责任度的平均值适中，标准差表明有一定程度的差异。政策支持度的平均值是所有聚类中最高的，但也伴随着较大的标准差，表明虽然这个聚类中的公司普遍受到较多的政策支持，但这种支持的程度不尽相同。

对于机遇型股票，自主创新能力的平均值略低于稳定型股票，但与稳定型股票相似，标准差表明聚类内部存在差异。系统性风险度的平均值最高，但仍然有较大的标准差，表明这个聚类的股票在市场风险上的表现相对较高，且不一致性较大。市场重估度的平均值略高于稳定型股票，且标准差较小，这表明这个聚类的股票在市场重估度上较为一致。社会责任度的平均值最低，且标准差最大，这可能表明这个聚类的公司在履行社会责任方面存在很大的不一致性。政策支持度的平均值和标准差都介于潜力型股票和稳定性股票之间。

总的来说，这些分类揭示了股票之间在关键指标上的内在相似性和差异性，为投资者提供了根据个人投资策略进行投资选择的依据。在实际的投资决策中，投资者应该考虑到每种股票类型的特点，并结合其他财务数据和市场动态来做出全面的分析和决策。在下面的任务中，我们将具体实践。

7 任务三

7.1 选择热点事件

在对 100 支中特估股票根据特征进行好分类之后，我们现在要结合典型的市场热点，设计一个基于中特估的短期股票投资组合。我们选择了几个具有代表性的事件：

1. 2023 年 10 月 7 日，巴勒斯坦伊斯兰抵抗运动 (哈马斯) 向以色列发射超 5000 枚火箭弹，随后以方宣布进入“战争状态”并展开大规模报复行动。我们选取事件发生后 15 个交易日的数据。

2. 2023 年 7 月 6 日，世界人工智能大会在上海举行。我们选取事件发生后 15 个交易日的数据。我们选取事件发生后 15 个交易日的数据。

3. 2008 年的 9 月 19 日，财政部、国家税务总局决定证券（股票）交易印花税调整为单边征税。我们选取事件发生后 11 个交易日的数据。

7.2 数据收集与预处理

我们收集中特估股票每个交易日的收益率（即当日涨跌幅/昨日收盘价）并剔除数据缺失的股票，并根据问题二中对中特估股票的分类，令每一类内股票的权重相同（即取每一类内股票当日收益率的均值作为这一类的收益率）。

以巴以冲突为例以下是我们收集并处理好的历史数据（部分）展现：

潜力型	稳定型	机遇型
-1.161473913	-0.246994737	-0.470410256
-1.042708696	-1.471126316	-3.073664103
0.230378261	-0.492557895	-0.522751282
2.046669565	0.895126316	0.966015385
-0.67573913	-0.894518421	-1.707715385
⋮	⋮	⋮
-1.116586957	-1.520771053	-2.787476923
0.691886957	1.024986842	1.483512821
0.481182609	0.595718421	1.953235897
0.292256522	0.736034211	0.057282051
0.186895652	0.549355263	0.817920513

表 5: 巴以冲突中三大类别平均收益率数据（部分）展现

7.3 模型的建立

我们已经将 100 支股票分为三大类，每一类分别对应的时间段内的收益率为 R_1, R_2, R_3。决策变量 k_1, k_2, k_3 分别表示每一类别股票对应的投资权重，由于 A 股不允许做空，因此我们将权重的约束条件设为

$$k_1, k_2, k_3 \geq 0, \quad k_1 + k_2 + k_3 = 1.$$

我们可以得到投资的期望收益率

$$E(R) = E(k_1R_1 + k_2R_2 + k_3R_3) = k_1E(R_1) + k_2E(R_2) + k_3E(R_3).$$

投资的方差为

$$\begin{aligned}
V &= var(R) \\
&= var(k_1R_1 + k_2R_2 + k_3R_3) \\
&= var(k_1R_1) + var(k_2R_2) + var(k_3R_3) \\
&\quad + 2cov(k_1R_1, k_2R_2) + 2cov(k_1R_1, k_3R_3) + 2cov(k_2R_2, k_3R_3) \\
&= k_1^2var(R_1) + k_2^2var(R_2) + k_3^2var(R_3) \\
&\quad + 2k_1k_2cov(R_1, R_2) + 2k_1k_3cov(R_1, R_3) + 2k_2k_3cov(R_2, R_3) \\
&= \sum_{j=1}^{3}\sum_{i=1}^{3} k_ik_jcov(R_i, R_j).
\end{aligned}$$

假设约束期望收益率为 α，我们构建出了二次规划问题进行求解

$$\begin{cases}
\min V = \sum\limits_{j=1}^{3}\sum\limits_{i=1}^{3} k_ik_jcov(R_i, R_j) \\
k_1E(R_1) + k_2E(R_2) + k_3E(R_3) \geq \alpha \\
k_1, k_2, k_3 \geq 0 \\
k_1 + k_2 + k_3 = 1
\end{cases}$$

我们利用 Python 进行求解。

7.4 短期股票投资组合

对于短期投资组合的建立，我们基于热点事件，寻找最优策略。我们应用上述马科维茨均值-方差模型进行投资组合的选择。下表是基于每个热点时期，每个类别股票分别对应的权重以及每个事件时期的收益率及风险：

	潜力型	**稳定型**	**机遇型**	收益率	风险
巴以冲突	1	0	0	-0.166	1
人工智能	0.639	0.361	0	0.2	1.069
印花税	0	0	1	13.052	0.623

表 6: 针对热点事件的短期股票投资组合

7.5 短期股票投资组合实测

通过在数据集上的实测，我们绘制出了时间窗口开始时起每一天的优化前后的累计收益率 $\tilde{R}$。假设 $\tilde{R}_t$ 是累积到第 t 天累计收益率，则

$$\tilde{R}_t = (1 + \tilde{R}_{t-1})(1 + R_t) - 1.$$

经过马科维茨均值-方差模型优化后的权重为潜力型 1，稳定型 0，机遇型 0。可以看到巴以战争期间两种策略的持续持有收益基本都在 0 以下，这可能是由于巴以冲突造成了较大的市场恐慌，整体行情向下。

但我们仍能看到即便在行情下行的情况下，我们的投资组合表现仍强于平均投资。这可能是因为潜力型股票多属于银行和银行和保险业，业务多面向国内，且其营收来源较为稳定，不易受外部

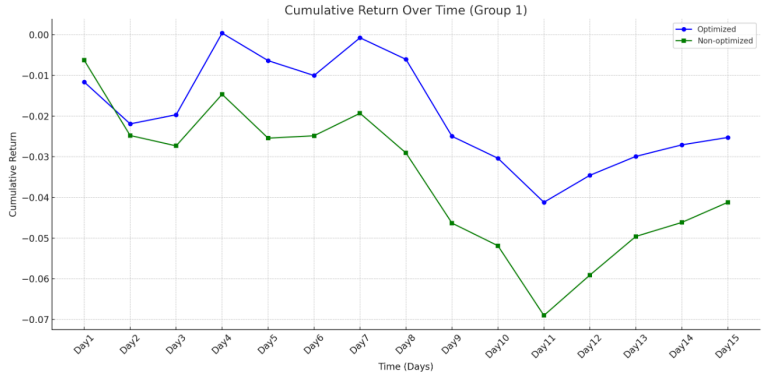

图 8: 2023 年巴以冲突后 15 个交易日累计收益率

战争等影响。且潜力型股票本身估值较低因此在行情下行时较难产生太大的波动。可以大致推断，潜力型股票具有一定的避险能力。

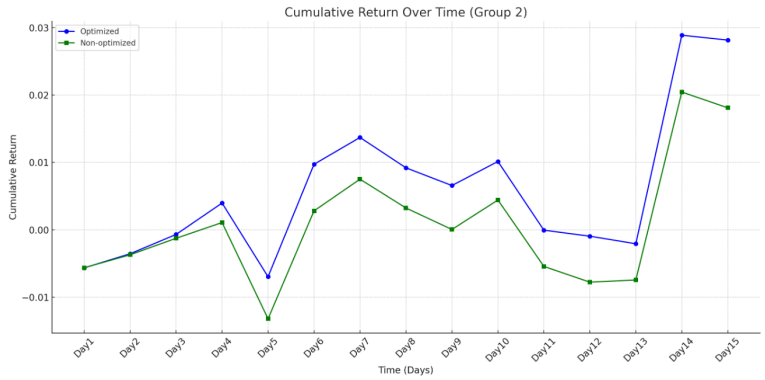

图 9: 2023 年人工智能大会后 15 个交易日累计收益率

经过马科维茨均值-方差模型优化后的权重为潜力型 0.604，稳定型 0.396，机遇型 0。2023 世界人工智能大会在 7 月 6 日至 8 日期间举行，可以看到在会议期间，市场行情略有提振，但随后便开始波动。可能是因为今年沪深 A 股总体行情下行，市场对经济状态的预测始终不乐观，故导致世界人工智能大会对市场的作用有限。

我们的投资组合表现仍强于平均投资。由于市场不稳定，潜力型的股票仍然占最高比重。稳定型也在投资组合中有相当的权重，可能是由于稳定型中有一定数量涉及通信行业的公司，世界人工智能大会的开展对这些公司而言是利好消息。可以预测，稳定型股票对于一些行业政策导向性的热点事件会有较好的市场表现。

经过马科维茨均值-方差模型优化后的权重为潜力型 0，稳定型 0，机遇型 1。2008 年印花税改革，印花税改为单边征收，释放巨大市场利好，市场整体行情向上展开反攻行情。可以看到在利好行情下，优化后的投资组合更能把握市场机会。优化后的权重显示应全部投入机遇型。机遇型虽然系统性风险度较高，但也说明其更能反应整体大盘的趋势。当大盘整体十分利好时，投资机遇型可以更大化收益。

图 10: 2008 年印花税下调后 11 个交易日累计收益率

8 任务四

8.1 长期股票投资组合模型

马科维茨均值-方差模型是一种能够在风险和收益之间取得平衡的策略投资模型。当投资组合内的各成分相关程度较低时，马科维茨均值-方差模型可以利用协方差矩阵量化相关性通过算法为不同股票分配权重，而相关性较低的投资组合成分可以为模型提供更丰富的策略选择，效果也就更好。而当投资组合内的各成分相关程度较高时，各成分间协同变动的趋势强，模型便没有足够的优化选择，效果也就更差。因此为了验证分类的合理性，我们计算三个类别股票的相关性，并作出相关性热力图：

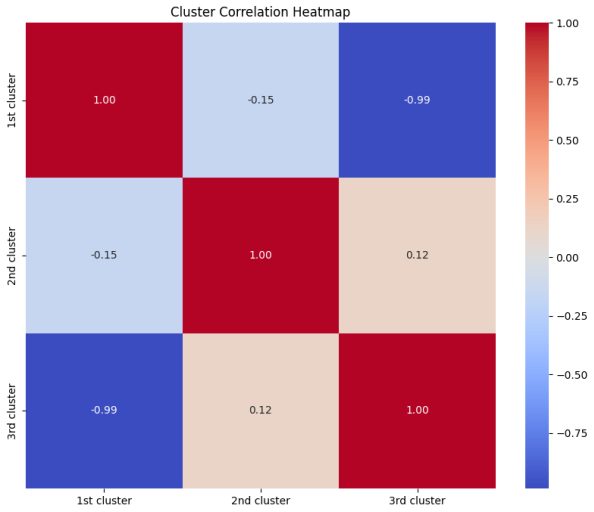

图 11: 中特估股票三大类别相关性热力图

可以看到潜力型（1st cluster）和稳定型（2nd cluster）呈负相关性为-0.15，这意味着两者变化方向相反；潜力型（1st cluster）和机遇型（3nd cluster）呈负相关性为-0.99，这意味着两者变化方向相反，且互斥性较强；稳定型（2st cluster）和机遇型（3nd cluster）呈正相关性为 0.12，但两者间的正相关性不强，处于可接受范围。简而言之，采用三种类型的分类进行投资可以做出较为灵活的投资组合。

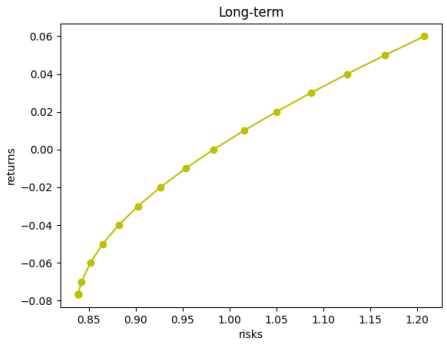

图 12: 长期投资组合马科维茨模型曲线

上图是马科维茨模型的二次规划的图像解，图中横轴为投资组合的风险，纵轴是对应风险下的期望日收益率，可以看到随着风险增长收益也随之增长。经过对比不同的收益风险组合，我们选择了期望日收益率为百分之 0.05 的投资组合。

下表是对于长期投资，每个类别股票分别对应的权重以及每个事件时期的收益率及风险：

	潜力型	**稳定型**	**机遇型**	收益率	风险
长期投资	0.878	0.122	0	0.05	1.166

表 7: 长期股票投资组合

8.2 长期股票投资组合收益分析

将 7 月至 10 月期间三种类型股票的日收益率带入马科维茨均值-方差模型内，添加约束条件令期望日收益率大于等于百分之 0.05 进行优化，得到三个月内交易日的三种类型股票的优化权重为潜力型 0.878，稳定型 0.122，机遇型 0。可以看到量化投资策略长期下更重视潜力型资产的投资，由于潜力型有更高的市场重估度，ROE 较高且稳定，且 PB 估值较低有进一步拉升的空间，所以更适合长期持有价值投资。

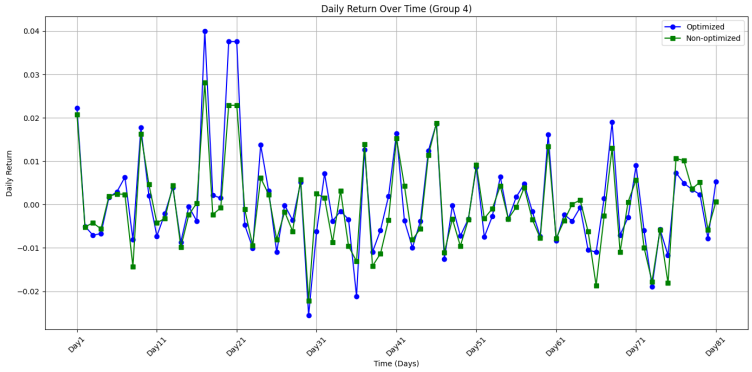

图 13: 2023 年 7 月-10 月 81 天交易日每日收益率

结合每日收益率和累计收益率的图像可以看到。七月初的行情走势较好，而后期可能因为巴以冲突爆发的原因造成市场变动。因此可以总结八月至十月的行情属于动荡行情。

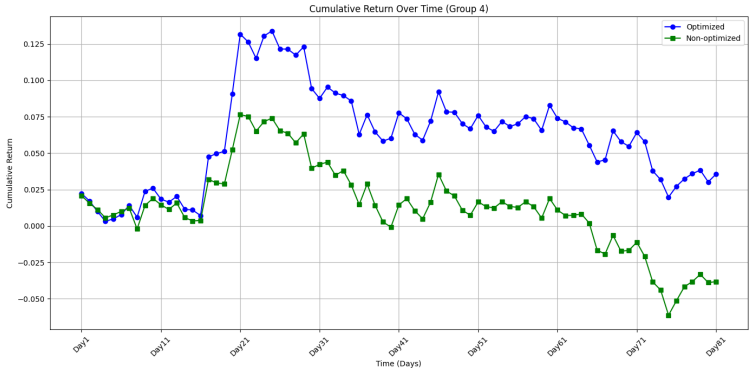

图 14: 2023 年 7-10 月 81 天交易日累计收益率

可以看到，在长期动荡的行情下，将潜力型股票作为主力确定价值投资稳定风险的主基调配合稳定型股票捕捉市场机会可以更好地积累收益优势。

9 模型评价与推广

9.1 模型优点

本模型的主要优点在于其综合了中国特色的资本市场定位和政策背景，通过明确的特征指标体系，准确捕捉中特估股票的特性。首先，模型将政策支持度作为关键评价因素，紧密贴合中国市场的实际情况，增加了模型的实用性和准确度。其次，通过结合净资产收益率与市净率两个指标，模型能够全面评估股票的综合价值，而非单一依赖传统指标。再者，模型的应用范围广泛，适用于短期和长期的投资决策，为投资者提供了量化的决策工具。最后，该模型的策略输出具有较高的灵活性，可以根据市场热点和宏观环境的变化进行动态调整，提高了投资组合的适应性和盈利潜力。

9.2 模型不足

模型存在的不足主要体现在几个方面。首先，模型的准确性在很大程度上依赖于数据的质量和完整性，对于数据的采集和处理要求较高。数据的获取难度、时效性以及处理的主观性都可能影响模型的输出。其次，模型虽然考虑了多个维度，但可能仍有遗漏重要的市场或非市场因素，例如股票的流动性、投资者情绪、政策变动的不确定性等。此外，模型采用的算法，如支持向量机和层次聚类，尽管在分类上表现出色，但可能存在过拟合的风险，特别是在样本量较小或特征维度较高时。最后，模型对于不同投资者的风险偏好适配不足，可能需要进一步定制化以满足个体化的投资需求。

9.3 模型推广

模型的推广潜力巨大，尤其是在中国特色资本市场背景下。首先，该模型提供了一个结构化的评估框架，帮助投资者理解和利用政策导向对投资决策的影响，这在中国市场尤为重要。模型的推广可以帮助国内投资者更好地理解市场动态，提升投资效率。其次，模型的多维度评价体系可应用于投资教育和资本市场分析工具的开发，有助于提升投资者的整体财经素养。再者，该模型的框架和方法论可以被用于其他新兴市场，尤其是那些政府政策对市场影响较大的市场。最后，随着金融科技的发展，该模型可以整合到智能投顾服务中，为广大投资者提供个性化的投资建议和解决方案。然而，模型的推广需要考虑地区差异性，确保模型参数和策略能够适应不同市场环境的变化。

参考文献

[1] 王军, 肖塞, 李昕澎. 加快探索中国特色估值体系 [J]. 中国金融, 2023, No.997(07): 69-71.

[2] 周道洪. 建设中国特色估值体系是国资国企改革的大考 [J]. 上海国资, 2023, No.271(04): 12.

[3] 赵文荣, 张若海, 宋广超. ESG 融入中国特色估值体系, 新窗口期助力国企可持续发展 [J]. 冶金财会,2023,42(5):16-21. DOI:10.3969/j.issn.1004-7336.2023.05.003.

[4] 余婧. 均值—方差—近似偏度投资组合模型与实证分析 [D]. 上海: 复旦大学,2010.

[5] 如何深刻理解“中国特色的估值体系”？-知乎
https://zhuanlan.zhihu.com/p/586152537

附录

代码清单

```
""" 任务一 SVM训练代码 """

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.metrics import classification_report, accuracy_score

# Load the data
file_path = '/Users/jinqigong/Desktop/unnormalized data with label.xlsx'
data = pd.read_excel(file_path)

# Separate features and target variable
X = data.iloc[:, 2:-1]  # Exclude the first two columns and the last column
y = data.iloc[:, -1]  # Target variable is the last column

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize a pipeline with Standard Scaler and SVM with a linear kernel
# We use class_weight='balanced' to address the class imbalance issue
pipeline = make_pipeline(StandardScaler(), SVC(kernel='linear', class_weight='balanced'))

# Train the SVM model
pipeline.fit(X_train, y_train)

# Predict on the test set
y_pred = pipeline.predict(X_test)

# Calculate accuracy and classification report
accuracy = accuracy_score(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

# Output the results
print(f'Accuracy: {accuracy}')
print(f'Classification Report: \n{class_report}')

# Retrieve the SVM model from the pipeline
svm_model = pipeline.named_steps['svc']

# Get the weights of the features
feature_weights = svm_model.coef_[0]
# Get the bias term from the SVM model
bias = svm_model.intercept_[0]

# Display the weights and bias
weights_df = pd.DataFrame({'Feature': X.columns, 'Weight': feature_weights})
print(weights_df)
print(f'Bias: {bias}')
```

```
""" 任务二 层次聚类树状图和肘部法则图代码 """

import pandas as pd
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
```

```
from sklearn.metrics import pairwise_distances
import matplotlib.pyplot as plt
import numpy as np

# Load the Excel file
file_path = 'E:\\大四第一段\\數學建模比賽\\支撐材料\\normalized data only csi.xlsx'
data = pd.read_excel(file_path)

# Selecting relevant columns for clustering (excluding '证券代码' and '证券名称')
clustering_data = data.iloc[:, 2:]

# Using the hierarchical clustering method
linked = linkage(clustering_data, method='ward')

# Creating a dendrogram to visualize the hierarchical clustering
plt.figure(figsize=(10, 7.5))
dendrogram(linked, orientation='top', labels=data['证券代码'].values, distance_sort='descending',
                                                show_leaf_counts=True, color_threshold=8)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Stock Code')
plt.ylabel('Distance')
plt.show()

# Function to calculate the total within-cluster sum of square (WSS)
def calculate_wss(points, kmax):
    sse = []
    for k in range(1, kmax+1):
        # Determine the clusters at the given level k
        labels = fcluster(linked, k, criterion='maxclust')
        # Calculate pairwise distance matrix for the dataset
        dist_matrix = pairwise_distances(points)
        # Sum of squares within each cluster
        total_within_ss = 0
        for i in range(1, k+1):
            cluster_points = points[labels == i]
            if cluster_points.shape[0] > 0:  # Check if the cluster is not empty
                cluster_distance_matrix = pairwise_distances(cluster_points)
                total_within_ss += np.sum(cluster_distance_matrix**2)
        sse.append(total_within_ss)
    return sse

# Calculate WSS for a range of number of clusters
kmax = 10  # maximum number of clusters to consider
wss = calculate_wss(clustering_data, kmax)

# Plot the Elbow Method Graph
plt.figure(figsize=(8, 6))
plt.plot(range(1, kmax+1), wss, marker='o')
plt.title('Elbow Method For Optimal k')
plt.xlabel('Number of clusters')
plt.ylabel('Total within-cluster sum of squares')
plt.xticks(range(1, kmax+1))
plt.show()
```

```
""" 任务二 层次聚类分类代码 """

# Assign clusters with the updated number of clusters
num_clusters = 3
clusters = fcluster(linked, num_clusters, criterion='maxclust')
```

```
# Add the cluster information to the original dataframe
data_with_clusters = data.copy()
data_with_clusters['Cluster'] = clusters

# Display the results
# Grouping the data by cluster and display the stocks in each cluster
clustered_data = data_with_clusters.groupby('Cluster')
cluster_results = {}
for cluster_num, cluster_data in clustered_data:
    cluster_results[cluster_num] = cluster_data[['证券代码', '证券名称']]
    print(f"Cluster {cluster_num}:")
    print(cluster_data[['证券代码', '证券名称']], "\n")

# Calculate the mean and standard deviation for each cluster

cluster_mean_std = data_with_clusters.groupby('Cluster').agg(['mean', 'std'])
print("cluster_mean_std")
```

```
""" 任务三 分类相关性热力图 """

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the Excel file
file_path = '/Users/jinqigong/Desktop/formap.xlsx'
data = pd.read_excel(file_path)

# Select only the columns with the features, ordered as specified
features_ordered = ['1st cluster', '2nd cluster', '3rd cluster']

# Calculate the correlation matrix
correlation_matrix = data[features_ordered].corr()

# English labels for the features, in the order specified
feature_names_english_ordered = ['1st cluster', '2nd cluster', '3rd cluster']

# Rename the columns and index of the correlation matrix to English
correlation_matrix_english = correlation_matrix.rename(columns=dict(zip(features_ordered,
                                                    feature_names_english_ordered)), index=dict(zip(
                                                    features_ordered, feature_names_english_ordered)))

# Plot the heatmap with English labels
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix_english, annot=True, fmt=".2f", cmap='coolwarm', cbar=True)
plt.title('Cluster Correlation Heatmap')
plt.show()
```

```
""" 任务三 短期投资组合 马科维茨均值-方差模型 巴以冲突 """


import numpy as np
import matplotlib.pyplot as plt
import cvxopt as opt
from cvxopt import blas, solvers
import pandas as pd

np.random.seed(123)
```

```
# 关掉进度展示，进度展示是运行过程进度的一个打印输出，可以通过其查看代码运行进度
solvers.options['show_progress'] = False

# 加载数据
df = pd.read_excel('E:\\大四第一段\\數學建模比賽\\bayi average return.xlsx')

# 仅保留收益率数据
return_vec = df.iloc[:, 1:]

def optimal_portfolio(returns):
    n = len(returns)
    returns = np.asmatrix(returns)

    N = 25
    mus = [-0.41+0.01*i for i in range(N)]

    # 转化为cvxopt matrices
    P = opt.matrix(np.cov(returns))
    q = opt.matrix(np.zeros((n, 1)))
    pbar = opt.matrix(np.mean(returns, axis=1))

    # 约束条件
    G = opt.matrix(np.concatenate((-np.array(pbar).T,-np.eye(n)),0))   # opt默认是求最大值，因此要求
                                                                        最小化问题，还得乘以一个负号
    A = opt.matrix(1.0, (1, n))
    b = opt.matrix(1.0)

    # 使用凸优化计算有效前沿
    portfolios = [solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * mu, np.zeros((n,
                                                                1))), 0)), A, b)['x']
                  for mu in mus]
    ## 计算有效前沿的收益率和风险
    returns = [blas.dot(pbar, x) for x in portfolios]
    risks = [np.sqrt(blas.dot(x, P*x)) for x in portfolios]
    #m1 = np.polyfit(returns, risks, 2)
    #x1 = np.sqrt(m1[2] / m1[0])
    #print(x1)
    x1=-0.17
    # 计算最优组合
    opt_weights = solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * x1, np.zeros((n,
                                                                1))), 0)), A, b)['x']
    opt_returns = blas.dot(pbar,opt_weights)
    opt_risks = np.sqrt(blas.dot(opt_weights,P*opt_weights))
    return opt_weights, opt_returns, opt_risks, returns, risks

opt_weights, opt_returns, opt_risks, returns, risks = optimal_portfolio(return_vec)

print(opt_weights)
print(opt_returns)
print(opt_risks)
#print(returns)
#print(risks)

plt.plot(risks, returns, 'y-o')
plt.title('war')
plt.ylabel('returns')
plt.xlabel('risks')
plt.ticklabel_format(useOffset=False,style='plain')
```

```
plt.show()
```

```
""" 任务三 短期投资组合 马科维茨均值-方差模型 人工智能 """


import numpy as np
import matplotlib.pyplot as plt
import cvxopt as opt
from cvxopt import blas, solvers
import pandas as pd

np.random.seed(123)

# 关掉进度展示，进度展示是运行过程进度的一个打印输出，可以通过其查看代码运行进度
solvers.options['show_progress'] = False

# 加载数据
df = pd.read_excel('E:\\大四第一段\\數學建模比賽\\rengongzhineng average return.xlsx')

# 仅保留收益率数据
return_vec = df.iloc[:, 1:]

def optimal_portfolio(returns):
    n = len(returns)
    returns = np.asmatrix(returns)

    N = 28
    mus = [0.02+0.01*i for i in range(N)]

    # 转化为cvxopt matrices
    P = opt.matrix(np.cov(returns))
    q = opt.matrix(np.zeros((n, 1)))
    pbar = opt.matrix(np.mean(returns, axis=1))

    # 约束条件
    G = opt.matrix(np.concatenate((-np.array(pbar).T,-np.eye(n)),0))   # opt默认是求最大值，因此要求
                                                                        最小化问题，还得乘以一个负号
    A = opt.matrix(1.0, (1, n))
    b = opt.matrix(1.0)

    # 使用凸优化计算有效前沿
    portfolios = [solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * mu, np.zeros((n,
                                                                1))), 0)), A, b)['x']
                  for mu in mus]
    ## 计算有效前沿的收益率和风险
    returns = [blas.dot(pbar, x) for x in portfolios]
    risks = [np.sqrt(blas.dot(x, P*x)) for x in portfolios]
    #m1 = np.polyfit(returns, risks, 2)
    #x1 = np.sqrt(m1[2] / m1[0])
    #print(x1)
    x1=0.2
    # 计算最优组合
    opt_weights = solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * x1, np.zeros((n,
                                                                1))), 0)), A, b)['x']
    opt_returns = blas.dot(pbar,opt_weights)
    opt_risks = np.sqrt(blas.dot(opt_weights,P*opt_weights))
    return opt_weights, opt_returns, opt_risks, returns, risks

opt_weights, opt_returns, opt_risks, returns, risks = optimal_portfolio(return_vec)
```

```
print(opt_weights)
print(opt_returns)
print(opt_risks)
#print(returns)
#print(risks)

plt.plot(risks, returns, 'y-o')
plt.title('ai')
plt.ylabel('returns')
plt.xlabel('risks')
plt.ticklabel_format(useOffset=False,style='plain')
plt.show()
```

```
""" 任务三 短期投资组合 马科维茨均值-方差模型 印花税 """


import numpy as np
import matplotlib.pyplot as plt
import cvxopt as opt
from cvxopt import blas, solvers
import pandas as pd

np.random.seed(123)

# 关掉进度展示，进度展示是运行过程进度的一个打印输出，可以通过其查看代码运行进度
solvers.options['show_progress'] = False

# 加载数据
df = pd.read_excel('E:\\大四第一段\\數學建模比賽\\yinhua average return.xlsx')

# 仅保留收益率数据
return_vec = df.iloc[:, 1:]

def optimal_portfolio(returns):
    n = len(returns)
    returns = np.asmatrix(returns)

    N = 47
    mus = [8.4+0.1*i for i in range(N)]

    # 转化为cvxopt matrices
    P = opt.matrix(np.cov(returns))
    q = opt.matrix(np.zeros((n, 1)))
    pbar = opt.matrix(np.mean(returns, axis=1))

    # 约束条件
    G = opt.matrix(np.concatenate((-np.array(pbar).T,-np.eye(n)),0))   # opt默认是求最大值，因此要求
                                                                       最小化问题，还得乘以一个负号
    A = opt.matrix(1.0, (1, n))
    b = opt.matrix(1.0)

    # 使用凸优化计算有效前沿
    portfolios = [solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * mu, np.zeros((n,
                                                                  1))), 0)), A, b)['x']
                  for mu in mus]
    ## 计算有效前沿的收益率和风险
    returns = [blas.dot(pbar, x) for x in portfolios]
    risks = [np.sqrt(blas.dot(x, P*x)) for x in portfolios]
```

```
    #m1 = np.polyfit(returns, risks, 2)
    #x1 = np.sqrt(m1[2] / m1[0])
    #print(x1)
    x1=13
    # 计算最优组合
    opt_weights = solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * x1, np.zeros((n,
                                                        1))), 0)), A, b)['x']
    opt_returns = blas.dot(pbar,opt_weights)
    opt_risks = np.sqrt(blas.dot(opt_weights,P*opt_weights))
    return opt_weights, opt_returns, opt_risks, returns, risks

opt_weights, opt_returns, opt_risks, returns, risks = optimal_portfolio(return_vec)

print(opt_weights)
print(opt_returns)
print(opt_risks)
#print(returns)
#print(risks)

plt.plot(risks, returns, 'y-o')
plt.title('tax')
plt.ylabel('returns')
plt.xlabel('risks')
plt.ticklabel_format(useOffset=False,style='plain')
plt.show()
```

```
""" 任务三 优化前后累计收益率对比折线图 """

import pandas as pd
import matplotlib.pyplot as plt

# Load the Excel file
file_path = '/Users/jinqigong/Desktop/Research/大湾区杯/yin.xlsx'
data_full = pd.read_excel(file_path)

# Extract the specific rows for the four groups and clean the data
groups_data = [
    data_full.iloc[0:3, :].dropna(axis=1, how='all'),  # 1st group: rows 1-3
    data_full.iloc[4:7, :].dropna(axis=1, how='all'),  # 2nd group: rows 5-7
    data_full.iloc[8:11, :].dropna(axis=1, how='all'), # 3rd group: rows 9-11
    data_full.iloc[12:15, :].dropna(axis=1, how='all'), # 4th group: rows 13-15
]

# Function to plot each group
def plot_group(data, group_idx, title):
    # Set the plot size
    plt.figure(figsize=(14, 7))

    # Extract data for plotting
    dates = data.columns[1:]  # Skipping the 'Cumulative Return' column
    optimized_returns = data.iloc[0, 1:].values  # Optimized data
    non_optimized_returns = data.iloc[1, 1:].values  # Non-optimized data

    # Plotting the data
    plt.plot(dates, optimized_returns, label='Optimized', marker='o', color='blue')
    plt.plot(dates, non_optimized_returns, label='Non-optimized', marker='s', color='green')

    # Adding titles and labels
    plt.title(f'{title} (Group {group_idx})')
```

```
    plt.xlabel('Time (Days)')
    plt.ylabel('Cumulative Return')
    plt.legend()

    # Improving display
    plt.grid(True)
    plt.xticks(rotation=45)
    plt.tight_layout()  # Adjusts the plot to ensure everything fits without overlapping

    # Special case for the 4th group
    if group_idx == 4:
        xticks = [day for day in dates if day in ['Day1', 'Day11', 'Day21', 'Day31', 'Day41', 'Day51
                                                  ', 'Day61', 'Day71', 'Day81']]
        plt.xticks(ticks=xticks, labels=xticks, rotation=45)

    # Show the plot
    plt.show()

# Loop through each group and plot the data
for idx, group_data in enumerate(groups_data, start=1):
    plot_group(group_data, idx, "Cumulative Return Over Time")
```

```
""" 任务四 长期投资组合 马科维茨均值-方差模型 """

import numpy as np
import matplotlib.pyplot as plt
import cvxopt as opt
from cvxopt import blas, solvers
import pandas as pd

np.random.seed(123)

# 关掉进度展示，进度展示是运行过程进度的一个打印输出，可以通过其查看代码运行进度
solvers.options['show_progress'] = False

# 加载数据
df = pd.read_excel('E:\\大四第一段\\數學建模比賽\\longterm average return.xlsx')

# 仅保留收益率数据
return_vec = df.iloc[:, 1:]

def optimal_portfolio(returns):
    n = len(returns)
    returns = np.asmatrix(returns)

    N = 19
    mus = [-0.12+0.01*i for i in range(N)]

    # 转化为cvxopt matrices
    P = opt.matrix(np.cov(returns))
    q = opt.matrix(np.zeros((n, 1)))
    pbar = opt.matrix(np.mean(returns, axis=1))

    # 约束条件
    G = opt.matrix(np.concatenate((-np.array(pbar).T,-np.eye(n)),0))   # opt默认是求最大值，因此要求
                                                                       最小化问题，还得乘以一个负号
    A = opt.matrix(1.0, (1, n))
    b = opt.matrix(1.0)
```

```
    # 使用凸优化计算有效前沿
    portfolios = [solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * mu, np.zeros((n,
                                                    1))), 0)), A, b)['x']
                  for mu in mus]
    ## 计算有效前沿的收益率和风险
    returns = [blas.dot(pbar, x) for x in portfolios]
    risks = [np.sqrt(blas.dot(x, P*x)) for x in portfolios]
    #m1 = np.polyfit(returns, risks, 2)
    #x1 = np.sqrt(m1[2] / m1[0])
    #print(x1)
    x1=0.05
    # 计算最优组合
    opt_weights = solvers.qp(P, q, G, opt.matrix(np.concatenate((-np.ones((1, 1)) * x1, np.zeros((n,
                                                    1))), 0)), A, b)['x']
    opt_returns = blas.dot(pbar,opt_weights)
    opt_risks = np.sqrt(blas.dot(opt_weights,P*opt_weights))
    return opt_weights, opt_returns, opt_risks, returns, risks

opt_weights, opt_returns, opt_risks, returns, risks = optimal_portfolio(return_vec)

print(opt_weights)
print(opt_returns)
print(opt_risks)
#print(returns)
#print(risks)

plt.plot(risks, returns, 'y-o')
plt.title('long-term')
plt.ylabel('returns')
plt.xlabel('risks')
plt.ticklabel_format(useOffset=False,style='plain')
plt.show()
```

```
""" 任务四 长期投资组合累计收益率 """

import pandas as pd
import matplotlib.pyplot as plt

# Reload the Excel file to extract the specific rows for the fourth group
file_path = '/Users/jinqigong/Desktop/yin.xlsx'
data_full = pd.read_excel(file_path)

# Reload the Excel file to extract the specific rows for the fourth group
data_group_4 = pd.read_excel(file_path, skiprows=12, nrows=3)

# Clean the data to remove any NaN values that may affect the plotting
# Dropping the 'Cumulative Return' column to have only the days and returns
data_group_4_cleaned = data_group_4.dropna(axis=1).drop('Cumulative Return', axis=1)

# Getting the days for the x-axis (which are the column names of the dataframe)
# We filter out the columns where the 'Optimized' row (which is now the first row after skipping
                                                    rows) is NaN
days_group_4 = [day for day in data_group_4_cleaned.columns if not pd.isna(data_group_4_cleaned.iloc
                                                    [0][day])]

# Extracting the 'Optimized' and 'Non-optimized' returns for plotting
optimized_returns_group_4 = data_group_4_cleaned.iloc[0].dropna()
non_optimized_returns_group_4 = data_group_4_cleaned.iloc[1].dropna()
```

```
# Plotting the data for the fourth group
plt.figure(figsize=(14, 7))
plt.plot(days_group_4, optimized_returns_group_4, label='Optimized', marker='o', color='blue')
plt.plot(days_group_4, non_optimized_returns_group_4, label='Non-optimized', marker='s', color='
                                                    green')

# Adding titles and labels
plt.title('Cumulative Return Over Time (Group 4)')
plt.xlabel('Time (Days)')
plt.ylabel('Cumulative Return')
plt.legend()
plt.grid(True)

# We only label specific days on the x-axis as per the instruction
xticks = [day for day in days_group_4 if day in ['Day1', 'Day11', 'Day21', 'Day31', 'Day41', 'Day51'
                                                    , 'Day61', 'Day71', 'Day81']]
plt.xticks(ticks=xticks, labels=xticks, rotation=45)

plt.tight_layout()
plt.show()
```