

报名序号： 0158

赛题题目：光伏建筑一体化板块指数发展趋势分析及
预测

光伏建筑一体化板块指数发展趋势分析及预测

摘 要

本文运用了数据分析方法并结合股市实际,根据不同时间段内的流通股本建立光伏板块指数的适应性模型;随着对板块指数分析的不断深入,构建不同数学模型;利用 Python 对数学模型进行编程求解,对板块指标、个股相关性、风险度量等问题进行分析。

针对问题一,在考虑自由流通盘决定各股权重比例的情况下,利用派许加权公式求得光伏板块指数的移动平均线模型,由此求得的板块指数计算各时间段移动平均线。

针对问题二,根据问题 1 所建立的模型,使用“除数修正法”对模型进行修正以减小偏差。针对该板块未来发展趋势的预测,采用前 3 个月交易日的板块指数预测后一天的板块指数,搭建以 Tensorflow 为框架的 LSTM 时间序列预测模型,对后续时段的数据进行预测。

针对问题三,因各股所采用的总股本与流通股本在不同时间段有所变化,将光伏板块指数与上证指数分时间段进行相关性分析,利用肯德尔系数、皮尔逊相关系数、斯皮尔曼系数、最大信息系数等四种相关系数求得板块指数与上证指数的相关性。

针对问题四,采用熵风险度量方法对 37 家企业个股风险进行评估,并基于熵风险度量方法设计了一套 37 家企业个股股票投资选择方案,由此对 2021 年 6 月份光伏建筑一体化板块 37 家企业个股股票的投资风险进行由低到高排序和筛选。另外,得到了 100 万元人民币投资该板块股票的最优投资方案。

针对问题五,光伏建筑一体化发展报告主要从光伏建筑的技术、市场、政策支持以及光伏板块指数变化等几个方面分析来介绍未来光伏产业发展的前景。

本文综合考虑了多因素对板块指数影响,建立不同时间段板块指数的数学模型,采用相关智能算法进行计算,其结果符合题目要求,具备一定的参考价值以及实际应用前景。

关键词: 光伏板块指数; 移动平均线; 相关性分析; 数学分析方法

一、问题重述

1.1 问题背景

随着经济和科技的快速发展，对碳的排放量进一步增加，而碳的大量排放使得全球气候变暖，这也严重影响了生存的环境。因此绿色低碳已经成为未来人类生存与发展的主旋律。国家为了大力发展新能源，将 2030 年实现“碳达峰”与 2060 年实现“碳中和”作为我国应对全球气候变暖的一个重要远景目标。国家为了实现绿色低碳的发展，减少对化石燃料的依赖，发展绿色新能源成为必然的要求。光伏发电是新能源的重要组成部分，而利用工业建筑等资源实施光伏发电的光伏建设一体化（BIPV）成为完成节能减排的有效方法之一。通过对 BIPV 板块指数发展趋势的预测，了解板块未来的发展趋势，对该板块的发展具有重大的意义。

1.2 问题的提出

因我国的证券市场的不断发展，股票交易已经成为日常生活的一部分，在股票交易的过程中，不同类型的股票都要日 K 线（由开盘价、收盘价、最高价、最低价构成）以及平均移动线，通过对每只股票的移动平均线和日 K 线的分析，可以对每只股票的走势进行判断。股票市场中板块的指数是市场发展的综合反映，它可以反映板块的整体的走势，其作为投资者对市场趋势判断的重要指标。但板块指标作为一个多因素影响的参数，在利用光伏建筑一体化板块指数预测未来行业发展的趋势时，需要充分考虑板块各股票中的总股本、流通股本和股本价格等多影响因素，因此本文对利用板块指数走势来判断相对应行业发展趋势问题时，通过对股本价格、股本数量等因素建立不同的变化函数，通过层层递进分析，优化误差求解，实现对光伏建筑一体化板块指数走势做出预测，最终对该板块的行业发展给出分析。

1.3 本文待解决的问题

（1）利用附件 1 中数据，给出光伏建筑一体化板块指数的移动平均线（5 日、10 日、20 日等）模型，并绘制 2019 年 4 月 1 日至 2021 年 4 月 30 日该板块指数的移动平均线。

（2）利用 2021 年 5 月 6 日至 5 月 28 日数据，对所建立模型进行误差分析并修正模型，根据修正后的模型对该板块未来发展趋势做出预测，给出 5 月 28 日后 20 个交易日的日移动平均线、3 周的周移动平均线、2 个月的月移动平均线。

（3）利用 2019 年 4 月 1 日至 2021 年 5 月 28 日数据，以 2 个月为一个时间段，对上证指数和光伏建筑一体化板块指数进行相关性分析。

（4）对光伏建筑一体化板块个股投资风险进行评估，给出该板块 37 支股票 2021 年 6 月份投资风险由低到高的排序结果。假定投资者持有资金 100 万元人民币，欲全部用于该板块的投资，请给出 2021 年 6 月份的最优投资方案（当日可用于投资的资金为上一个交易日结束后投资者所持有的资金，且每日持股数不超过 5 支）。

(5) 请给相关部门写一份不少于 1500 字关于我国光伏建筑一体化行业未来发展趋势的报告。

二、问题分析

问题 1：在问题 1 的时间段内，仅考虑各股票的总股和流通盘数目和影响因子变化较小的前提下来计算板块的指数。由于自由流通盘决定各股权重比例，通过计算自由流通量比例赋予各股的权重大小，再利用派许加权公式求得该板块的板块指数，由求得的板块指数计算移动平均线模型。

问题 2：根据问题 1 所建立的模型，光伏建筑一体化板块指数会因为 37 家企业任意一家发生除息、停牌、除权、停市、汇率变动、新上市、摘牌、股本变动而产生误差，因此我们将使用“除数修正法”进行修正以减小偏差。针对该板块未来趋势发展的预测，我们考虑用前 3 个月的板块指数预测后一天的板块指数，时序预测模型的搭建利用 Tensorflow 框架，具体为 3 层 LSTM（长短期记忆神经网络）搭配 Dropout 层增强泛化能力，激活函数选择 Tanh。在 LSTM 后增加一个 Dense 层作为输出层，预测后一天的板块指数。

问题 3：在问题 3 中所给的时间段内，每隔 2 个月对上证指数与光伏板块指数进行相关性分析，由于所采用的总股与流通股在不同时间段的差异性，需要将光伏板块指数与上证指数分时间段进行相关性分析，板块相关性采用肯德尔系数、皮尔逊相关系数、斯皮尔曼系数、最大信息系数等四种相关系数进行相关性量化分析。

问题 4：针对光伏建筑一体化板块个股投资风险评估，首要解决的是如何选取 37 家企业个股的风险度量方法。我们采用熵风险度量方法对 37 家企业个股风险进行评估，并基于熵风险度量方法设计了一套 37 家企业个股股票投资选择方案，由此对 2021 年 6 月份光伏建筑一体化板块 37 家企业个股股票的投资风险进行由低到高排序和筛选。另外，需要得到 100 万元人民币投资该板块股票的最优投资方案，即需要根据各个股评估结果建立投资股票组合模型，从而获得最优投资方案。

问题 5：报告主要从光伏建筑的技术、市场、政策支持以及光伏板块指数的变化等几个方面来介绍未来光伏产业发展的前景。

三、模型假设

假设 1：光伏建筑一体化板块指数基点 A 为 1000 点；

假设 2：光伏建筑一体化板块指数得权重因子 c 以及汇率 s 均为 1；

假设 3：光伏建筑一体化板块 37 只股票的调整股本数保持不变；

四、符号定义

符号	定义
P	光伏建筑一体化板块指数
F_i	样本调整市值
F_i'	修正后样本调整市值
F	板块总调整市值
A	基点
r	除数
r'	新除数
n	自由流通量
N	样本总股本
λ	自由流通比例
v	加权比例
N_i	样本调整股本数
X_i	样本收盘价
c	权重因子
s	汇率
T	上证指数和光伏建筑一体化板块指数的肯德尔系数
C	上证指数和光伏建筑一体化板块指数一致性元素对数
D	上证指数和光伏建筑一体化板块指数不一致性元素对数
Y	上证指数和光伏建筑一体化板块指数的皮尔逊系数
MIC	上证指数和光伏建筑一体化板块指数的最大信息系数
SRC	上证指数和光伏建筑一体化板块指数的斯皮尔曼系数
r_{ij}	企业股票一段时间交易日的日收盘价
r_i	第 i 种股票的投资收益
W_i	投资第 i 种股票的比率
δ_{ij}^2	第 i 种和第 j 种股票间的风险关联性
V	第 i 种和第 j 种股票收益率的协方差矩阵

五、模型的建立与求解

5.1 问题 1

5.1.1 问题 1 模型思路分析

对于光伏建筑一体化板块指数的移动平均线模型建立，必然需要先将板块内的 37 只个股按照不同权重形式计算求得反映整个板块变化趋势的板块指数，进而通过算术运算求解 5 日、10 日、20 日板块指数的移动平均线。因此，首要解决 37 只个股各自的权重分配方式，目前股票板块对于个股权重分配类型有三种：一是根据总股本大小定义权重大小；二是通过流通盘来衡量权重大小；三是等权重来定义个股权重。显然，第一种会因为不能够自由流通的股本降低了板块指数反映整个板块实际运行的准确度，而第三种等权重分配更是没有考虑到各股的股本流通情况对板块指数的影响，故选择使用流通盘分配权重更接近于真实的股市运行情况，因此我们中证指数有限公司使用的分级靠档方法来定义个股股份的加权比例，以确保参与板块指数计算的样本股本保持相对稳定，同时我们也采用该公司使用的且当前世界上主流、适用性较强的派许加权方法计算光伏建筑一体化板块指数，计算该板块指数的公式为：

$$P = \frac{F}{r} \times A \quad (1)$$

式中， P 表示光伏建筑一体化板块指数， F 为该板块样本的总调整市值， A 、 r 分别为基点和除数。

5.1.2 问题 1 模型建立与求解

通过 5.1.1 对问题 1 所提模型的建立思路分析可得，针对该板块建立运用派许加权方法求得的指数目标函数和指数移动平均线模型，具体模型如下：

- (1) 以 2019 年 4 月 1 日为基日，利用 37 家企业股票为样本计算指数，给定基点 A 为 1000 点，并通过网上收集 37 家企业股票 2019 年 4 月 1 日的的总股本和自由流通股本计算自由流通比例 $M^{[1]}$ ，函数关系式如公式 2 所示：

$$\lambda = \frac{n}{N} \quad (2)$$

式中， n 、 N 分别为自由流通量和样本总股本。

- (2) 使用个股自由流通量占样本个股总股本的比例来定义个股股份的加权比例^[2]，即根据(1)中计算的自由流通比例对照分级靠档表得到一定的加权比例，具体加权比例如表 1 所示，然后通过样本总股本数 N 和加权比例 v 相乘得到各个样本调整股本数 N_i ，具体关系如公式 3 所示：

$$(3) \quad N_i = N \times v \quad i=1, 2, \dots, 37 \quad (3)$$

表 1 分级靠档表

自由流通 比例 (%)	≤15	(15, 20]	(20, 30]	(30, 40]	(40, 50]	(50, 60]	(60, 70]	(70, 80]	>80
加权比例 (%)	上调至最 接近整数	20	30	40	50	60	70	80	100

- (4) 得到各企业股票的样本调整股本数后，再将其与各股收盘价格 X_i 、权重因子 c 以及汇率 s 相乘得到各个样本的调整市值 F_i ，累加求得该板块的总调整市值 F ，计算公式为：

$$F = \sum_{i=1}^{37} X_i \times N_i \times c \times s \quad i=1, 2, \dots, 37 \quad (4)$$

假定式中权重因子 c 以及汇率 s 均为 1，得到板块总调整市值 F 简化公式如下：

$$F = \sum_{i=1}^{37} X_i \times N_i \quad i=1, 2, \dots, 37 \quad (5)$$

- (5) 根据 (3) 求得的该板块总调整市值 F 确定除数 r ，通过基日 37 家企业各自的样本调整市值和除数即可得到光伏建筑一体化板块的流通盘，见图 1 所示，然后再根据式 (1) 计算出基日收盘指数。另外，后续若有企业因公司事件影响证券价格或股本变动等时需要进行修正，具体修正方法见 5.2.1。

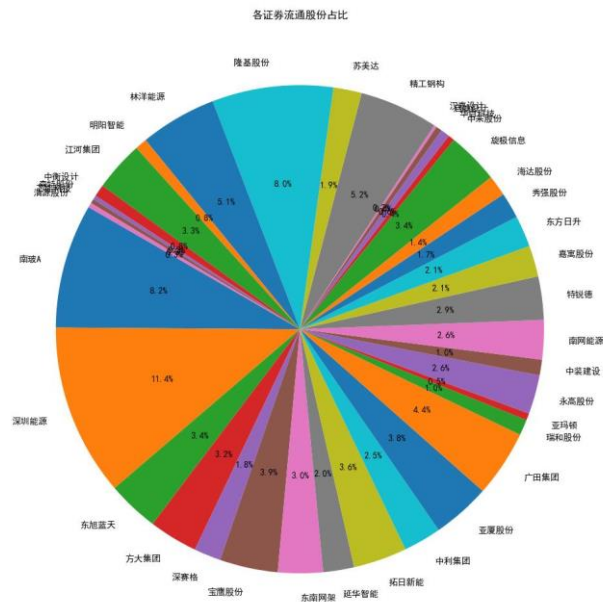


图 1 各股票流通股份占比分布图

- (6) 假定 2019 年 4 月 1 日至 2021 年 4 月 30 日无需进行修正，根据建立的模型和题目附件 1 给出数据即可计算出 2019 年 4 月 1 日至 2021 年 4 月 30 日光

伏建筑一体化板块指数，如图 2 所示。

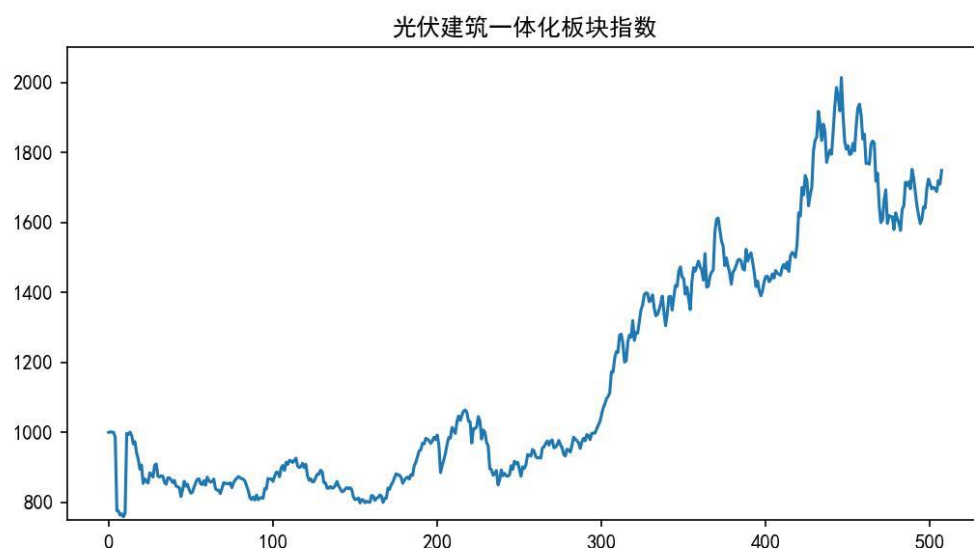


图 2 光伏建筑一体化板块指数图

由图 2 可知，该光伏建筑一体化板块指数整体呈现上升趋势，发展前景较好。根据模型计算出的 2019 年 4 月 1 日至 2021 年 4 月 30 日的光伏建筑一体化板块指数可直接通过数值求解方法得到 2019 年 4 月 1 日至 2021 年 4 月 30 日该板块指数的 5 日、10 日、20 日移动平均线，如图 3 所示。

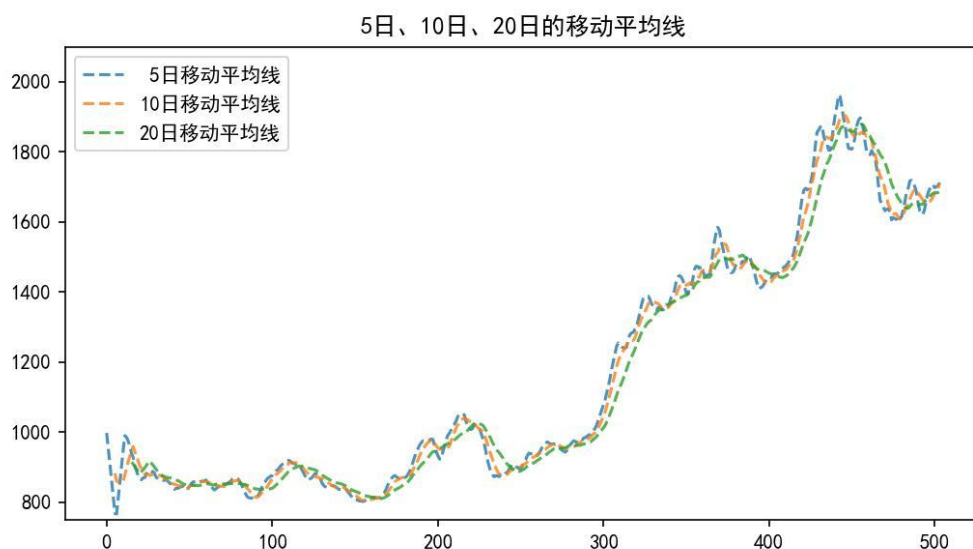


图 3 光伏建筑一体化板块指数 5 日、10 日、20 日的移动平均线

5.2 问题 2

5.2.1 问题 2 模型思路分析

首先，针对 5.1.2 建立的模型进行误差分析，自 2019 年 4 月 1 日到 2021 年 4 月 30 日以来，37 家光伏建筑一体化相关企业出现以下几种情况都将引起所建模型出现偏差：

- (1) 当 37 家相关企业任意一家发生可能影响证券价格变动的公司事件时；
- (2) 当 37 家相关企业任意一家公司发生引起股本变动的其他公司事件时；

(3) 当 37 家相关企业任意一家公司定期调整或临时调整生效时。

若任意一家企业出现以上情况且未进行修正的话均会导致模型产生误差。经调查发现，光伏建筑一体化板块 37 家企业自 2019 年 4 月 1 日以来均出现以上几种情况，因此需要对 5.1.2 建立的模型进行误差修正，具体修正方法见 5.2.2。

前文已用各证券的的总股本与流通股本对 2019 年 4 月 1 日的模型进行了修正得到了新的板块指数模型。相对于修正前的模型，该模型更能反映市场变化。通过对《东方财富网》调研发现：37 家证券的总股本和流通股本在近一年内的变化不大^[1]。因此，本文考虑以修正后的板块指数模型与近一年内的个股收盘价相结合，构建时序预测模型。

5.2.2 问题 2 模型建立与求解

针对 5.1.2 建立的模型因除息、停牌、除权、停市、汇率变动、新上市、摘牌、股本变动所产生的误差，我们一律根据“除数修正法”进行修正，以保证光伏建筑一体化板块指数的连续性，具体的修正公式为：

$$\frac{F_i}{r} = \frac{F'_i}{r'} \quad (6)$$

式中， F_i 、 F'_i 为该板块样本调整前后的总调整市值， r 、 r' 为修正前后的除数。

5.2.3 问题 3 时序预测模型

考虑用前 3 个月的板块指数预测后一天的板块指数，3 个月内的工作日约为 50 天，即用前 50 个数据预测后一个数据。时序预测模型的搭建利用 Tensorflow 框架，由 3 层 LSTM（长短期记忆神经网络）搭配 Dropout 层增强泛化能力，激活函数利用 Tanh 提高收敛速度，最后利用 Dense 神经网络作为输出层，预测后一天的板块指数。时序预测模型如图 4 所示。

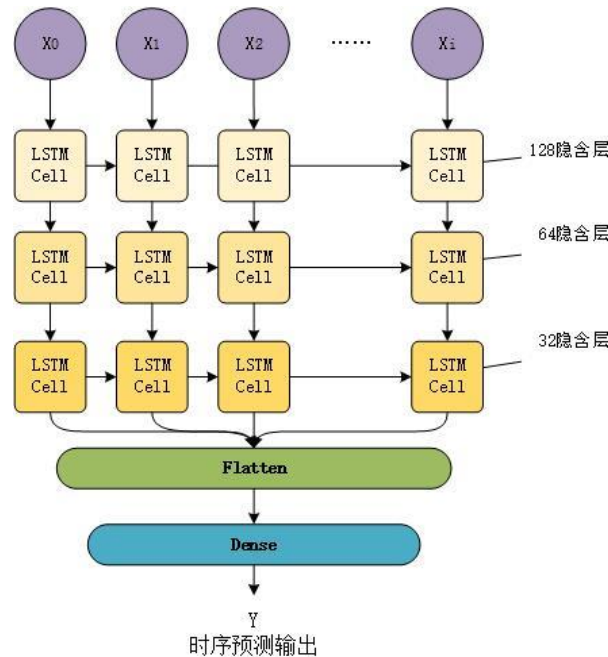


图 4 时序预测模型

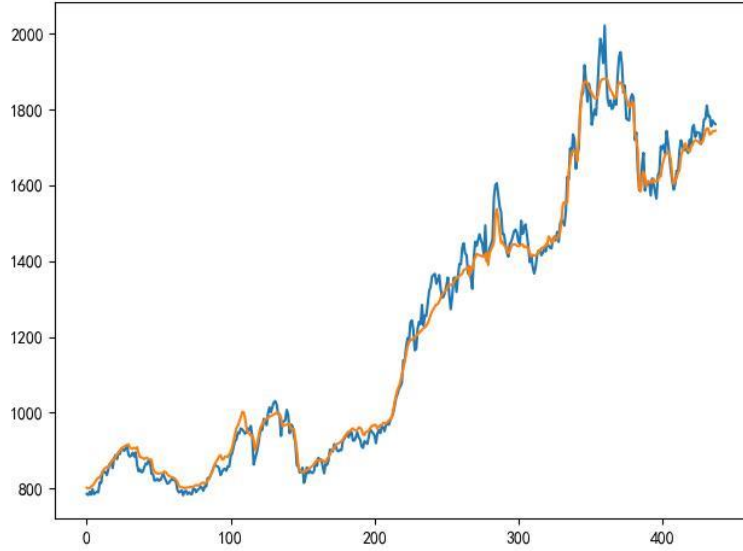


图 5 模型预测效果

模型训练时，优化器选择 RMSprop，损失函数选择 MAE 平均绝对误差，学习率设置为 $1 \times e^{-4}$ ，对所选数据进行训练 4000 次，最终预测的效果良好。模型预测效果如图 5 所示，

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_i'| \quad (7)$$

式中， y_i 为真实值， y_i' 为预测值。

5.2.4 未来发展趋势的预测

所建立的时序预测模型对数据已有较好的预测精度。因此，根据最新 50 个交易日的板块指数继续对后续交易日的板块指数进行预测。预测流程图如图 6 所示。将已知最新的 50 个交易日信息读取，并对下一个交易日进行预测，预测结果插入到下一步预测的末尾，并删除排在数据最前的交易日数据，得到新数据。再将新数据用于对下一个交易日板块指数的预测。重复上述操作，获得往后 50 个交易日（约两个月）的信息。5 月 28 日后 20 个交易日的日移动平均线、3 周的周移动平均线、2 个月的月移动平均线分别如图 7(a)、(b)、(c)所示。

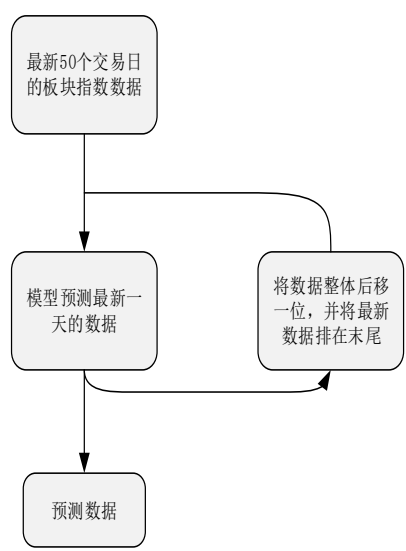
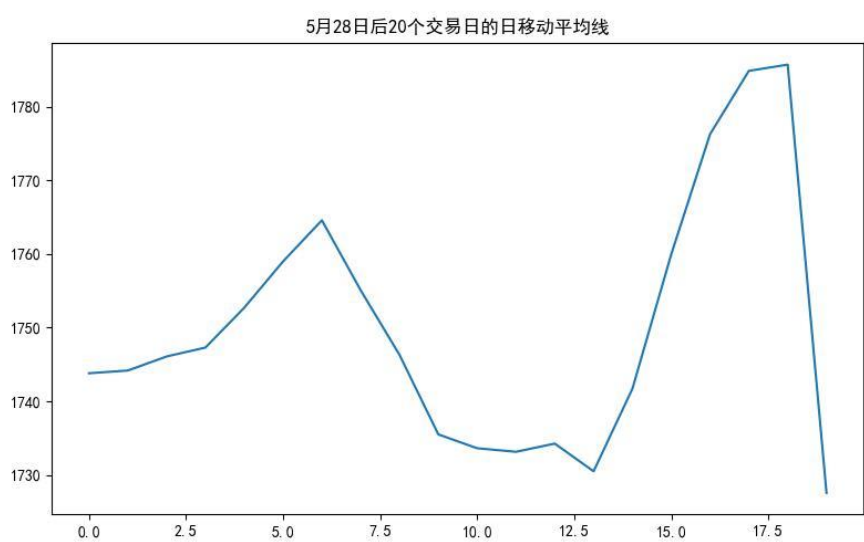
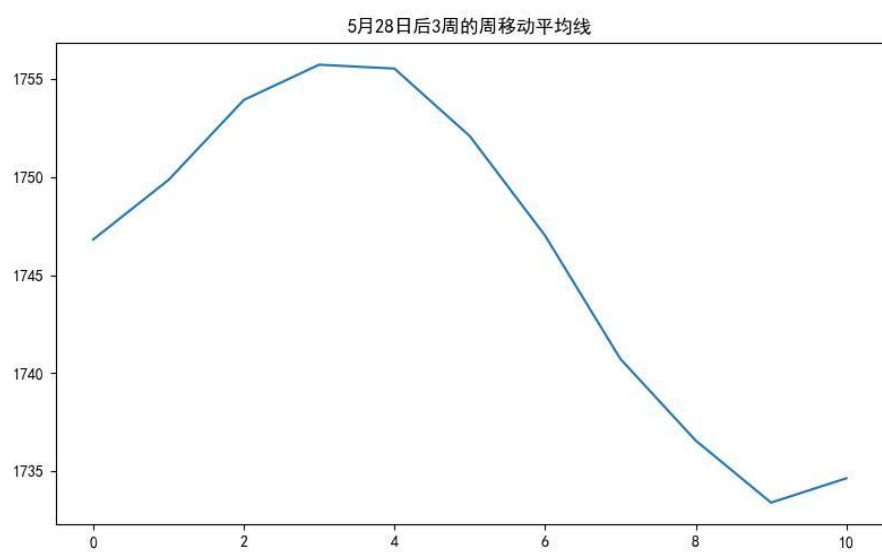


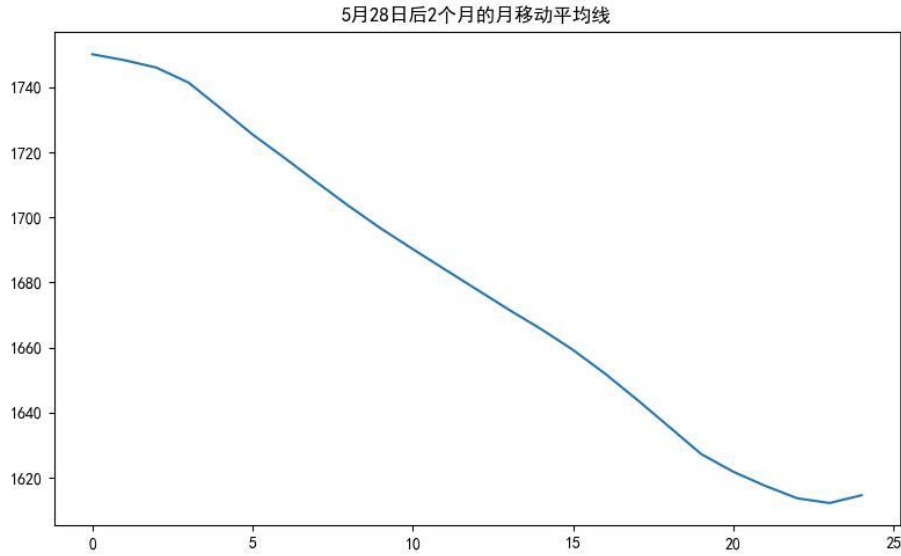
图 6 预测流程图



(a) 5 月 28 日后 20 个交易日的日移动平均线



(b) 5 月 28 日后 3 周的周移动平均线



(c) 5月28日后2个月的月移动平均线

图 7 2021 年 5 月 28 日后移动平均线

5.3 问题 3

5.3.1 问题 3 模型分析

由于问题 3 需要利用 2019 年 5 月 6 号至 2021 年 5 月 28 号数据,以 2 个月为一个时间段,对上证指数和光伏指数建筑一体化板块指数进行相关性分析^[3]。因此我们首先需要利用问题 1 中板块指数方法求解得到该时间段内的光伏建筑一体的板块的指数。由于此时间段内采用各股票的总股本以及流通股本的数目在 2019 年 4 月 1 日至 2020 年 4 月 30 日时间段内与 2020 年 5 月 6 日至 2020 年 5 月 27 日时间段内有一定不同,因此对于这两段时间内的板块指数采取分段求取。

$$\lambda = \frac{n}{N} \quad (8)$$

式中, n 、 N 分别为自由流通量和样本总股本。

当 N 和 n 变化时,从而可能对所得到的自由流通比例产生差异,因此对于这两段时间内的板块指数采取分段求取。

为了在不同时间段内的上证指数与光伏建筑一体化板块指数的相关性分析,采用多种不同类型的相关性分析方法对两者指数进行量化比较,以此更好分析以及检验指数的相关性,减少因单一相关性计算方法而造成对结果分析。

5.3.2 问题 3 模型建立与求解

(1) 通过板块指数计算的方法得到两段不同时间内的光伏板块指数,由于所得到的光伏板块的指数与上证指数的板块不在统一比较区间之内,这样会使得二者之间的相关性较弱,因此通过对二者的板块指数做标准化处理,使之规定在一定的标准之内,主要是对板块的指数减去均值之后再除以方差。具体如下:最后得到的不同的时间段内的

标准的板块指数对比图，如图 8（a）和图 8（b）。

$$\bar{P} = \frac{P_1 + P_2 + \cdots + P_n}{n} \quad (9)$$

$$P_N^2 = \frac{1}{N} \sum_{i=1}^N (P_i - \bar{P})^2 \quad (10)$$

$$P_i^1 = \frac{P_i - \bar{P}}{P_N^2} \quad (11)$$

式中， P 为板块指数； \bar{P} 为计算所得的板块指数平均值； P_N^2 为计算所得的板块方差值； P_i^1 为标准化的板块指数。



图 8（a）2019-04-01 至 2020-04-30 内上证指数与光伏板块指数对比图



图 8（b）2020-05-06 至 2021-05-27 内上证指数与光伏板块指数对比图

通过图 8 (a) 可以看出在前一段时间之内上证指数与光伏板块指数在经过时间段内的比较时，可以直观的看出该时间段二者的相关指数整体的趋势上呈现正相关的相关性；通过图 8 (b) 可以看出上证指数与光伏板块指数的相关性与前半段有所差异，后半段时间内二者呈现负相关性。采用分段时间的方式对二者的指数比较可以更加精确比较二者相关性。

(2) 通过计算得到标准化的上证指数与光伏板块指数，再以 2 个月为时间段对板块指数相关性分析，利用相关性分析的计算公式主要有肯德尔系数、皮尔逊相关系数、斯皮尔曼系数、最大信息系数，具体的相关性计算公式如下所示：

$$T = \frac{C - D}{\sqrt{(N_3 - N_1)(N_3 - N_2)}} \quad (12)$$

式中， T 为上证指数和光伏建筑一体化板块指数的肯德尔系数肯德尔系数； C 为上证指数和光伏建筑一体化板块指数一致性元素对数； D 为上证指数和光伏建筑一体化板块指数不一致性元素对数； N_1 、 N_2 、 N_3 分别为针对上证指数、光伏建筑一体化板块指数及两者计算的分段小集合，具体函数关系为：

$$\begin{cases} N_1 = \sum_{i=1}^s \frac{1}{2} U_i (U_i - 1) \\ N_2 = \sum_{i=1}^t \frac{1}{2} V_i (V_i - 1) \\ N_3 = \frac{N'(N' - 1)}{2} \end{cases} \quad (13)$$

式中， N' 为上证指数或光伏建筑一体化板块指数元素个数， s 、 t 分别为上证指数或光伏建筑一体化板块指数拥有的小集合数， U_i 、 V_i 分别为上证指数或光伏建筑一体化板块指数第 i 个小集合所包含的元素数。

$$Y = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (14)$$

式中， n 表示上证指数或光伏建筑一体化板块指数元素个数； x_i 、 y_i 分别为上证指数或光伏建筑一体化板块指数的第 i 个点指数值； \bar{x} 、 \bar{y} 分别为上证指数或光伏建筑一体化板块指数的平均值。

$$MIC[X;Y] = \max_{|X||Y| < B} \frac{I[X;Y]}{\log_2(\min(|X|, |Y|))} \quad (15)$$

式中， $MIC[X;Y]$ 为上证指数和光伏建筑一体化板块指数的最大信息系数；其中 X 、 Y 分别为上证指数或光伏建筑一体化板块指数； $I[X;Y]$ 为上证指数和光伏建筑一体化板块指数的相互信息，具体函数关系式为：

$$I[X;Y]=\sum_{X,Y}\rho(X,Y)\log_2\frac{\rho(X,Y)}{\rho(X)\rho(Y)} \quad (16)$$

式中， $\rho(X,Y)$ 为上证指数和光伏建筑一体化板块指数的联合概率密度分布函数。

$$SRC=1-\frac{\sum d_i}{n^3-n} \quad (17)$$

式中， n 为上证指数或光伏建筑一体化板块指数元素个数， d_i 为上证指数和光伏建筑一体化板块指数每一对样本的等级之差。

利用四个相关性公式计算所得的相关性结果如表 2 和表 3 所示

表 2 2019-04-01 至 2020-4-30 板块指数相关性计算结果

	皮尔逊 相关系数	最大 信息系数	斯皮 尔曼系数	肯德 尔系数	取绝对值 后平均值
2019 年 4 月 -2019 年 6 月	-0.2119	0.4387	-0.3460	-0.1607	0.2893
2019 年 6 月 -2019 年 8 月	-0.2552	0.5048	-0.4083	-0.1987	0.3418
2019 年 8 月 -2019 年 10 月	-0.2728	0.5956	-0.4406	-0.2262	0.3838
2019 年 10 月 -2019 年 12 月	-0.2714	0.5252	-0.4444	-0.2283	0.3673
2019 年 12 月 -2020 年 2 月	-0.2776	0.4854	-0.4610	-0.2431	0.3668
2020 年 2 月 -2020 年 4 月	-0.2749	0.5159	-0.4739	-0.2516	0.3791

通过表 2 看出利用皮尔逊相关系数与肯德尔系数对上证指数与光伏板块指数进行相关性分析时，二者的在每隔 2 个月内的相关性较弱；利用最大信息系数与斯皮尔曼系数对上证指数与光伏板块指数进行相关性分析时，其中最大信息系数计算所得 的相关性较强，而斯皮尔曼系数计算所得相关性有一定的关联。

表 3 2020-05-06 至 2021-5-28 板块指数相关性计算结果

	皮尔逊 相关系数	最大 信息系数	斯皮 尔曼系数	肯德 尔系数	取绝对值 后平均值
2020 年 5 月 -2020 年 7 月	-0.4307	0.4369	-0.3037	-0.2204	0.3479

2020 年 7 月 -2020 年 9 月	-0.4487	0.4369	-0.3288	-0.2315	0.3614
2020 年 9 月 -2020 年 11 月	-0.4289	0.4713	-0.3147	-0.2204	0.3588
2020 年 11 月 -2021 年 1 月	-0.4785	0.4831	-0.3464	-0.2425	0.3876
2021 年 1 月 -2021 年 3 月	-0.4005	0.4053	-0.3010	-0.2071	0.3285
2021 年 3 月 -2021 年 5 月	-0.3687	0.3873	-0.2792	-0.1938	0.3073

通过表 3 看出利用 s 斯皮尔曼相关系数与肯德尔系数对上证指数与光伏板块指数进行相关性分析时，二者的在每隔 2 个月内的相关性较弱；利用最大信息系数与皮尔逊相关系数对上证指数与光伏板块指数进行相关性分析时，二者所计算得到的相关性较强。

由表 2 和表 3 可知，当在不同段时间之内上证指数与光伏板块指数相关性呈现不同的规律，同时利用不同的相关性系数进行量化比较时，不同的相关性系数所得到的相关性不同，这也说明不同的相关性系数所使用数据有不同的适应性。

5.4 问题 4

5.4.1 问题 4 模型分析

针对光伏建筑一体化板块个股投资风险评估，首要解决的是如何选取 37 家企业个股的风险度量方法。我们采用熵风险度量方法对 37 家企业个股风险进行评估，并基于熵风险度量方法设计了一套 37 家企业个股股票投资选择方案，由此对 2021 年 6 月份光伏建筑一体化板块 37 家企业个股股票的投资风险进行由低到高排序和筛选。另外，需要得到 100 万元人民币投资该板块股票的最优投资方案，即需要根据各个股评估结果建立投资股票组合模型，从而获得最优投资方案。

任何股票或证券的风险度量方法最好的检验就是应用到实践当中去，因此风险度量方法的设计必须满足以下三个条件^[4]：

- 1) 可行性，股票的风险度量方法必然需要在投资实战当中接受考验，若经受不起实操检验则是没有意义的。
- 2) 实用性，度量方法应用到实战中时应通俗易懂、便捷可行，这样才能被广大投资者接受。
- 3) 准确性，风险度量是投资者进行股票投资的参考，如果度量方法的准确度不高，可能会给广大投资者带来难以估量的损失。

5.4.2 股票风险度量熵模型建立

基于熵风险度量方法的股票风险度量熵模型实施步骤如下：

(1) 初始熵值求解

假设 37 家企业股票一段时间交易日的日收盘价为 r_{ij} ($i=1,2,\dots,s, j=1,2,\dots,d$), i 表示该板块的股票支数, j 表示连续采集的天数, 则第 i 支股票日收益率记为 $R_{ij} = \frac{r_{ij} - r_{ij-1}}{r_{ij-1}}$ 。

依据我国证券管理部门规定: 中国股票在单日内涨跌幅度被限定在 10% 以内, 若股票在一天内涨跌幅度超过限度, 将自动停牌, 也即股票实际收益率区间为 $[-10\%, +10\%]$ 。我们将股票收益率区间 $[-10\%, +10\%]$ 均分成

$$q = 10 \times 2^k \quad (k = 0, 1, 2, 3, 4, 5; q = 10, 20, 40, 80, 160, 320) \quad (18)$$

个子区间, 即:

$$[-10\%, +10\%] = \bigcup_{t=1}^q \Delta_t \quad (t = 1, 2, \dots, q) \quad (19)$$

取步长为 $l = \frac{0.2}{q}$, 则子区间组为:

$$\Delta_i = \begin{cases} [-10\% + (i-1)l, -10\% + il] & (t = 1, 2, \dots, q-1) \\ [-10\% + (i-1)l, -10\% + il] & (t = q) \end{cases} \quad (20)$$

然后计算第 i 支股票分别落在第 t 个子区间的频数 n_{it} ($t = 1, 2, \dots, q$);

假若 $R_{ij} \leq -10\%$, 那么假定它落在 $[-10\%, -10\% + l]$;

假若 $R_{ij} \geq 10\%$, 那么假定它落在 $[+10\% - l, +10\%]$;

频率值为 $\rho_{it} = \frac{n_{it}}{d}$, 假定频率值与该股票状态概率相等, 即 $\rho_{it} = p_{it}$;

那么, 第 i 支股票的初始熵风险值:

$$H(s_i) = - \sum_{t=1}^q \rho_{it} \ln \rho_{it} \quad (21)$$

(2) 确定调节因子

因为初始熵值一定会随着收益区间密度增加而增长, 而通常情况下, 度量得到的风险值在一个恒定的范围较好, 即熵风险值不应该随度量标准的改变而变化。

因此, 我们利用引入调节因子来调解, 具体实施方法是: 首先确定区间密度为 q ($q = 10, 20, 40, 80, 160, 320$) 的六组初始熵值 $H(s_i) = - \sum_{t=1}^q \rho_{it} \ln \rho_{it}$, 并分别求解初始熵值的均值 $\overline{H_q(s)}$, 然后再将初始熵均值与 k 做线性相关性检验, 如果通过检验则做线性拟

合，建立变量为 k 的调节因子 $y = (A + Bk)$ ；如果不通过，则对初始熵值拟合出的非线性方程作为调节因子。通过验证，初始熵均值 $\overline{H_q(s)}$ 与 k 线性相关且效果良好，故选线性调节因子 $y = (A + Bk)$ 。

(3) 股票熵风险值

通过新调节因子重新代入熵函数式计算出股票的熵风险值：

$$H(s_i) = - \sum_{it=1}^q \rho_{it} \ln \rho_{it} (A + Bk) \quad (22)$$

5.4.3 Markowitz 股票组合投资模型的建立

马柯威茨组合理论的出发点是综合考虑“期望收益最大”和“风险最小”，通过投资亏损、收益概率与预期收益的偏离程度进行分散投资。记光伏建筑一体化板块 37 只股票的收益率向量记为： $R = (r_1, r_2, r_3, \dots, r_{37})^T$ ，投资组合向量为： $W = (w_1, w_2, w_3, \dots, w_{37})^T$ ，两种股票收益率的协方差矩阵为 $V = (\sigma_{ij})_{37 \times 37}$ ，则该股票组合的收益率为：

$$R_p = W^T R = \sum_{i=1}^{37} w_i r_i, \text{ 总风险记为 } \sigma_p^2 = W^T V W, \text{ 因此，股票组合投资的优化决策模型可以简化为等式约束二次规划问题：}$$

$$\min \frac{1}{2} \sigma_p^2 = \frac{1}{2} W^T V W \quad (23)$$

$$s.t. \begin{cases} W^T e = 1 \\ E(X_p) = W^T E(X) \end{cases} (e = (1, 1, \dots, 1)^T) \quad (24)$$

假定求得的收益方差即为投资风险。现有资金 100 万元人民币用于该板块的投资，当日可用于投资的资金为上一个交易日结束后投资者所持有的资金，且每日持股数不超过 5 支，用 r_i 表示第 i 种股票的投资收益，用 w_i 表示投资第 i 种股票的比率。用 δ_{ij}^2 表示第 i 种和第 j 种股票间的风险关联性，第 i 种和第 j 种股票收益率的协方差矩阵为

$V = (\sigma_{ij})_{37 \times 37}$ ，根据收益最大化和投资风险最小化的组合原则，建立如下多目标规划的投资组合模型：

目标函数：

$$\begin{aligned} \max E &= \sum_{i=1}^{37} w_i r_i \\ \min V &= \sum_{i=1}^{37} \sum_{j=1}^{37} \delta_{ij}^2 w_i w_j \end{aligned} \quad (25)$$

约束条件：

$$s.t \begin{cases} \sum_{i=1}^{37} w_i \leq 1 \\ w_i \geq 0 \\ \delta^2_{ij} = W^T V W \end{cases} \quad (26)$$

因为每日持股数不能超过 5 支，因而在此引入变量 ξ_i ，且满足 0-1 分布，即：

$$\xi_i = \begin{cases} 1, & \text{第 } i \text{ 种项目被投资} \\ 0, & \text{第 } i \text{ 种项目不被投资} \end{cases} \quad (27)$$

因此，在该模型基础上建立基于 Markowitz 的组合投资股票决策模型的目标函数为：

$$\begin{aligned} \max E &= \sum_{i=1}^{37} \xi_i w_i r_i \\ \min V &= \sum_{i=1}^{37} \sum_{j=1}^{37} \delta^2_{ij} \xi_i w_i \xi_j w_j \end{aligned} \quad (28)$$

其约束条件：

$$s.t \begin{cases} \sum_{i=1}^{37} \xi_i w_i \leq 1 \\ 0 \leq \sum_{i=1}^{37} \xi_i \leq 5 \\ \sum_{i=1}^{37} w_i \leq 1 \\ w_i \geq 0 \\ \delta^2_{ij} = W^T V W \end{cases} \quad (29)$$

5.4.4 模型的求解

(1) 通过 MATLAB 编写建立的股票风险度量熵模型程序对光伏建筑一体化板块 37 家企业个股投资风险进行评估，得出该板块 37 支股票 2021 年 6 月份投资风险由低到高排序结果如下：

表 4 该板块 37 支股票 2021 年 6 月份投资风险排序表

个股名称	风险指数 (%)	个股名称	风险指数 (%)	个股名称	风险指数 (%)
南玻 A	1.1751	明阳智能	0.8059	旋极信息	0.7329
江河集团	0.9685	延华智能	0.7904	拓日新能	0.7151

深圳能源	0.9164	东方日升	0.7904	广田集团	0.7151
华自科技	0.8887	启迪设计	0.7892	清源股份	0.7136
海达股份	0.8730	嘉寓股份	0.7792	东旭蓝天	0.7032
深赛格	0.8684	永高股份	0.7693	亚厦股份	0.7032
中利集团	0.8541	方大集团	0.7643	林洋能源	0.7032
隆基股份	0.8447	汉嘉设计	0.7580	中装建设	0.6931
芯能科技	0.8193	东南网架	0.7560	精工钢构	0.6890
森特股份	0.8158	秀强股份	0.7530	中衡设计	0.6890
中来股份	0.8142	瑞和股份	0.7397	宝鹰股份	0.6828
南网能源	0.8081	特锐德	0.7397		
苏美达	0.8059	亚玛顿	0.7383		

(2) 通过 MATLAB 编写建立的股票组合投资模型程序实现 37 家企业个股综合评估结果, 对持有资金 100 万元人民币的投资者拟定 2021 年 6 月份的最优投资方案如下:

表 5 2021 年 6 月份投资该板块股票分布表

天数	企业 1	企业 2	企业 3	企业 4	企业 5
第 1 天	深赛格	东南网架	东方日升	旋极信息	林洋能源
第 2 天	南玻 A	深赛格	拓日新能	江河集团	清源股份
第 3 天	东南网架	永高股份	华自科技	明阳智能	江河集团
第 4 天	宝鹰股份	东南网架	延华智能	特锐德	隆基股份
第 5 天	南玻 A	深圳能源	东南网架	亚玛顿	旋极信息
第 6 天	南玻 A	深圳能源	广田集团	中装建设	嘉寓股份
第 7 天	南玻 A	深圳能源	宝鹰股份	东南网架	精工钢构
第 8 天	亚厦股份	瑞和股份	旋极信息	中来股份	隆基股份
第 9 天	延华智能	中利集团	启迪设计	汉嘉设计	隆基股份
第 10 天	南玻 A	方大集团	中利集团	特锐德	旋极信息
第 11 天	南玻 A	亚玛顿	海达股份	苏美达	中衡设计
第 12 天	东南网架	中装建设	特锐德	精工钢构	林洋能源
第 13 天	东旭蓝天	延华智能	特锐德	华自科技	隆基股份
第 14 天	延华智能	亚玛顿	南网能源	苏美达	清源股份
第 15 天	南玻 A	亚厦股份	广田集团	嘉寓股份	林洋能源
第 16 天	深圳能源	中利集团	秀强股份	苏美达	明阳智能
第 17 天	东南网架	拓日新能	亚厦股份	中来股份	隆基股份
第 18 天	亚玛顿	海达股份	启迪设计	隆基股份	芯能科技

第 19 天	南玻 A	亚厦股份	广田集团	中装建设	嘉寓股份
第 20 天	南玻 A	秀强股份	隆基股份	明阳智能	江河集团
第 21 天	深赛格	东南网架	瑞和股份	明阳智能	森特股份
第 22 天	延华智能	中来股份	启迪设计	隆基股份	清源股份

表 6 2021 年 6 月份投资该板块股票资金分布表

	企业 1	企业 2	企业 3	企业 4	企业 5
第 1 天	0.004562	0.324846	0.264798	0.292772	0.113022
第 2 天	0.478437	0.156078	0.001132	0.090202	0.274151
第 3 天	0.000837	0.061082	0.605798	0.271601	0.060682
第 4 天	0.022309	0.321241	0.098077	0.272533	0.28584
第 5 天	0.466411	0.046486	0.156504	0.198783	0.131816
第 6 天	0.46934	0.245127	0.208341	0.010343	0.066849
第 7 天	0.398156	0.243298	0.030359	0.271763	0.056424
第 8 天	0.165875	0.083831	0.036313	0.287369	0.426611
第 9 天	0.319922	0.020068	0.15836	0.178363	0.323286
第 10 天	0.607363	0.010322	0.060167	0.085392	0.236755
第 11 天	0.456287	0.009919	0.355642	0.13564	0.042512
第 12 天	0.571266	0.156357	0.034776	0.18516	0.052441
第 13 天	0.060626	0.261887	0.194783	0.062507	0.420197
第 14 天	0.065229	0.121665	0.043056	0.498629	0.271421
第 15 天	0.481733	0.156455	0.091686	0.181929	0.088198
第 16 天	0.318832	0.177216	0.053956	0.024549	0.425447
第 17 天	0.069806	0.154709	0.02284	0.353138	0.399507
第 18 天	0.595952	0.013004	0.090308	0.265395	0.035342
第 19 天	0.695995	0.093358	0.087571	0.112095	0.01098
第 20 天	0.406911	0.14057	0.015012	0.253587	0.18392
第 21 天	0.019365	0.345008	0.124588	0.296392	0.214647
第 22 天	0.197793	0.436897	0.168481	0.126444	0.070385
组合收益期望（百万）	0.0142635435126756				

5.5 问题 5

因光伏技术和节能建筑的快速发展，由光伏与建筑组合形成新的一种建筑形式—光

电建筑，“光电建筑”一词最开始是在 2009 年国家有关文件中出现，当中国提出力争在 2030 年前实现“碳达峰”，在 2060 年前实现“碳中和”这一伟大目标，同时在 2021 年全国两会上提出实现低碳绿色发展规划的路线图时，光电建筑成为节能减排的有效方法，同时其成为建设绿色中国的方法之一。而光电建筑未来的发展形势应该是光明的，具体的主要表现在以下几个方面：

光伏建筑的发展首先在光伏发电的市场规模，光伏发电的不断发展对促进光伏建筑的未來应用具有现实的意义^[5]。目前在风电的领域，中国的风电装机和发电量都位居世界第一，整体的产业体系较为完整，同时国内的风电企业也具有国际竞争力，例如在 2020 年，全球最知名的前 15 大风电生产商中，中国占据了 10 位，整体的份额达到了 54.2%。在国家大力提倡核心竞争力的背景之下，光伏发电的核心技术也得到了较大的发展，其中储能技术以及电池转化效率技术也处于世界领先的位置，例如我国的龙头企业商业化电池最高效率提升至 23%，这也不断在改写光伏电池转换效率的世界记录，技术水平处于国际领先，这也进一步降低了光伏发电使用成本，另外用于建筑的光伏组件发展态势良好，国内的多家公司也推出多款适用于建筑的光伏组件，相关的产品也符合建筑的使用特性，可以完全满足“经济，美观，绿色”的建筑理念。我国每年新增的建筑面积达到 20 亿 m²，a 但对建筑进行光电建筑改造的面积只占 1%，因此对于建筑的面积的利用还有很大的空间。光电建筑属于节能建筑范畴，节能建筑提倡的是所建建筑的能耗由零消耗向正能源建筑发展，我国的建筑也是逐步从普通建筑到光电建筑，由消耗型建筑到节能型建筑过渡发展。通过图 9 可以看出我国未来节能建筑发展的历程。

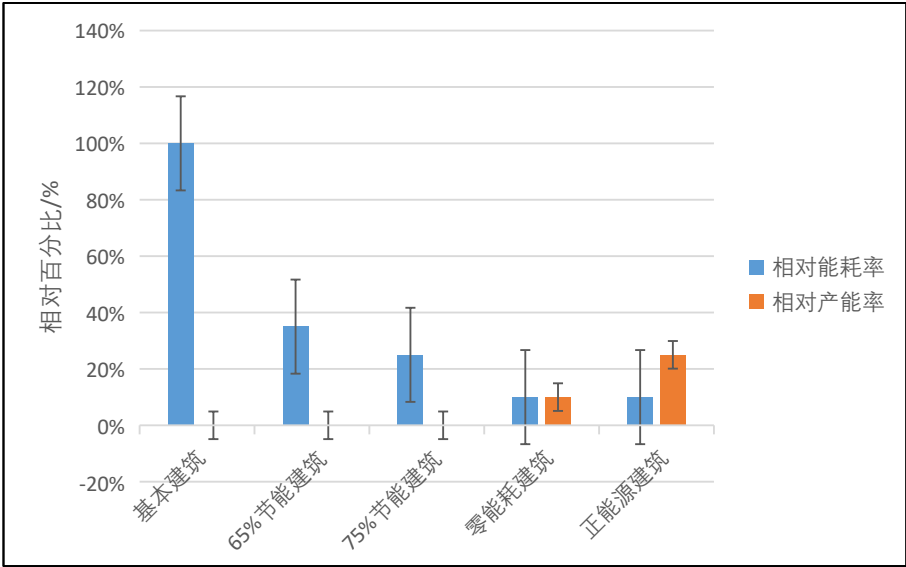


图 9 我国节能建筑的发展历程

而要实现零能耗和正能源的建筑最可能实现是增加新能源在建筑中的供能，光伏与建筑的结合是目前最好的一种应用方式，而充分利用建筑面积进行光伏发电最有效的方法。通过图 10 与图 11 比对可以清晰的发现，光电建筑与普通建筑的随着使用的时间年

限，建筑整体的造价是随之降低的，因光电建筑利用光伏发电可以节省大量的电费，同时可以获得相应的补贴，当建筑达到一定的使用年限，光电建筑依靠光伏发电收入使造价成本不断的下降，最后当建筑使用年限达到 30 至 50 周期寿命时，由于建筑本身带来一定的经济效益，使得造价成本为负数或盈利。

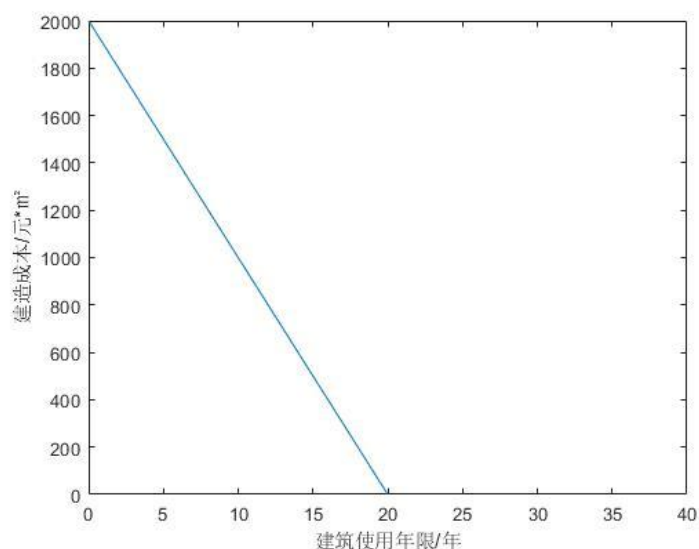


图 10 光电建筑的造价成本

普通建筑的造价成本呈现一条直线，其只能依靠折旧来回收成本，这也进一步的突出了光电建筑改变了建筑的造价特性，这样一种环保的建筑方式会在未来得到大力的推广。

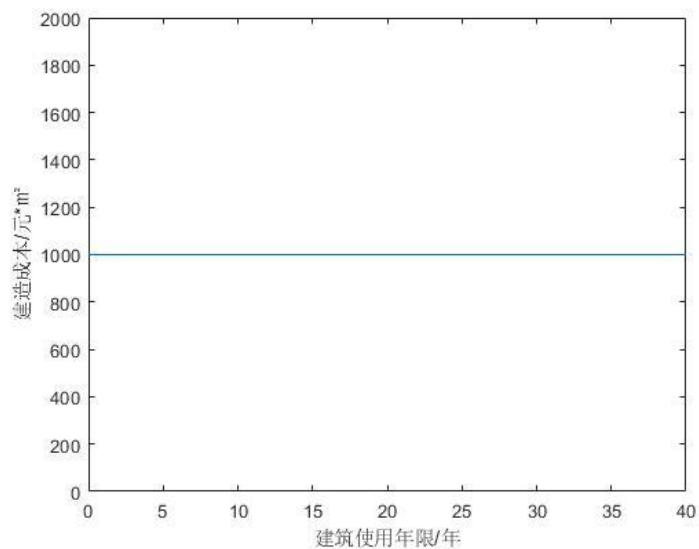


图 11 普通建筑的造价成本

新兴的产业发展需要技术、市场、政策等协同促进，缺少任何一方都失去发展的前景，产业的发展需要制度的保障，市场的需求推动，政策的扶持，技术创新引领企业发展的内生动力。目前随着各产业相互竞争，进一步促进产业技术的不断更新换代，这也为未来光电建筑的发展打下坚实基础，目前国内的光伏组件相关技术还是处于国际领先

的地位，这为进一步打开国际市场奠定基础。光电产业的需求是其发展的内在驱动，由前面对于光伏板块指数的计算可知，整个光伏板块从 2019 年开始，呈现一定的上升趋势，投资资金的大量涌入，这也预示着市场前景的广阔，投资者对于这一产业未来发展向好的信心。由于保护生态环境以及应对气候变化等问题日益突出，发展新能源的意义更加重要。目前，中国是世界上最大的风电生产与消费市场，截止 2020 年前，我国的能源发电中风能发电占 6.3%，而欧洲的风电占比已经达到了 15%，由此可以看出，我国的新能源发电占比过少，对化石燃料的发电过于依赖，为了改变这种能源利用结构，国家在 2020 前已经出台相关的促进新能源发展的各项通知，由表 7 可以看出，这一系列的通知为光电建筑的发展提供了有力的保证，这也预示着风力发电的市场具有广阔的前景^[6]。

表 7 各部委出台有关可再生能源开发利用的规章及规范文件

颁布部门	出台时间	政策名称	主要规定
国家发展改革委、 国家能源局	2017.01.18	关于试行可再生能源电力证书核发及资源认购交易制度的通知	目录内的项目可申请绿证权属资格
国家发展改革委、 国家能源局	2019.01.07	关于积极推进风电、光伏发电无补贴平价上网有关工作的通知	鼓励平价和低价上网项目通过绿证交易获得合理收益补偿
国家发展改革委、 国家能源局	2019.05.10	关于建立健全可再生能源电力消纳保障机制的通知	设定可再生能源电力消纳责任权重，认购绿证可记为消纳量
国家发展改革委、 国家能源局	2020.05.18	关于各省行政区域 2020 年可再生能源电力消纳责任权重的通知	明确政府、国网分部、省电力公司、电力交易机制以及承担消纳责任的市场主体（含自备电厂）各层面的职责定位及主要工作内容
财政部，国家发展改革委、 国家能源局	2020.09.29	关于促进非水可再生能源发电健康发展的若干意见	目录制改为清单制；新增海上风电和光热项目不再纳入补贴范围；全面推广绿色电力证书交易

而长期稳定的政策制度是推动产业发展的重要一环，只有当政府出台一系列惠民政策时，可以大力发展光伏建筑的基地，同时进一步减低其成本价格，企业以优惠的政策、完善的机制实现市场的快速发展，对核心技术不断革新基础下，从而支撑整个产业的良好发展。

目前正处在十四五的开端之际，新能源的发展正处在一个重要的战略窗口期，虽目

前的国际环境不稳定因素增多，但是国内经济稳中向好的趋势依然没有改变，国内所具有的优势和环境也没有变化，随着国家将低碳发展作为接下来一段时期的发展战略，进一步促进经济社会全面向绿色、高质量发展。在建立的新能源发展的大背景之下，光电建筑产业也会乘势而上，依靠国内的超级市场，完备的政策支持与保障，将战略机遇转化为产业发展的前进动力，实现高质量快速的发展。

参考文献

- [1] 东方财富网官网, <https://www.eastmoney.com/> , 2021 年 5 月 30 日.
- [2] 中证指数有限公司, <http://www.csindex.com.cn/zh-CN/indices/index-rules>, 2021 年 5 月 30 日.
- [3] 网易财经上证指数, [https:// quotes. money. 163. com/trade/lsjysj_zhishu_000001. html?year=2019&season=2](https://quotes.money.163.com/trade/lsjysj_zhishu_000001.html?year=2019&season=2) , 2021 年 5 月 30 日.
- [4] 郭正权, 王中魁, 王欣欣. 股票风险度量熵方法与方差法一致性实证研究[J]. 合作经济与科技, 2007, 000(013):54-55.
- [5]黄栋, 杨子杰, 王文倩. 新发展格局下新能源产业发展历程、内生逻辑与展望[J/OL].新疆师范大学学报(哲学社会科学版), 2021(06): 1-11[2021-05-31]. <https://doi.org/10.14100/j.cnki.65-1039/g4.20210526.001>.
- [6]徐伟, 边萌萌, 张昕宇等. 光电建筑应用发展的现状[J]. 太阳能, 2021(04): 6-15.

附录

附录 1

```
import pandas as pd
import json
import numpy as np
#%%
'''    读取 37 个公司的股价数据
df = []
for i in range(1,38):
    tmp = pd.read_excel('附件 1.xlsx',sheet_name='Sheet0 {}'.format(i))
    df.append(tmp)
'''

dict_z = {'南玻 A': 286327.72, '深圳能源': 396449.16, '东旭蓝天': 148687.39, '方大集团': 112338.42, '深
赛格': 123565.62, '宝鹰股份': 134129.69, '东南网架': 103440.22, '延华智能': 71215.30, '拓日新能':
123634.21, '中利集团': 87178.71, '亚厦股份': 133999.65, '广田集团': 153727.97, '瑞和股份': 36250.00, '亚
玛顿': 16000.00, '永高股份': 112320.00, '中装建设': 60000.00, '南网能源': 303030.30, '特锐德': 99757.01, '嘉
寓股份': 71676.00, '东方日升': 90430.19, '秀强股份': 59295.24, '海达股份': 60123.42, '旋极信息': 171080.26, '
中来股份': 24099.47, '华自科技': 26193.98, '启迪设计': 13422.35, '汉嘉设计': 21040.00, '精工钢构
': 181044.52, '苏美达': 130674.94, '隆基股份': 279079.52, '林洋能源': 176541.78, '明阳智能': 137972.24, '江河
集团': 115405, '中衡设计': 27517.87, '森特股份': 48001.2, '芯能科技': 50000, '清源股份': 27380,
}

dict_lt = {'南玻 A': 278564.87, '深圳能源': 396449.16, '东旭蓝天': 106034.06, '方大集团': 112195.26, '深
赛格': 53802.49, '宝鹰股份': 126306.31, '东南网架': 95851.10, '延华智能': 70996.81, '拓日新能':
121571.73, '中利集团': 70189.95, '亚厦股份': 124272.89, '广田集团': 152929.58, '瑞和股份': 30135.89, '亚
玛顿': 15957.36, '永高股份': 89592.45, '中装建设': 30606.00, '南网能源': 75757.58, '特锐德': 92472.11, '嘉
寓股份': 71428.50,
'东方日升': 68254.06, '秀强股份': 57471.52, '海达股份': 45533.35, '旋极信息': 105480.98, '中来股份
': 12057.77, '华自科技': 19668.31, '启迪设计': 11825.84, '汉嘉设计': 5260, '精工钢构': 151044.52, '苏美达
': 64028.40, '隆基股份': 278173.69, '林洋能源': 174887.68, '明阳智能': 27590, '江河集团': 115405, '中衡设计
': 26992.69, '森特股份': 7501.2, '芯能科技': 8800, '清源股份': 8880,
}

with open('./网上搜得的信息/各公司总股本.json', 'w') as f:
    json.dump(dict_z, f, ensure_ascii=False)
with open('./网上搜得的信息/各公司流通股本.json', 'w') as f:
    json.dump(dict_lt, f, ensure_ascii=False)

# 定义分级靠档的函数求得加权比例
def get_rate(rate):
    rate = rate*100
    if rate<15:
        if rate%1>0:
            rate = int(rate)+1
```

```

        else:
            rate = int(rate)
    elif 15<rate<=20:
        rate = 20
    elif 20<rate<=30:
        rate = 30
    elif 30<rate<=40:
        rate = 40
    elif 40<rate<=50:
        rate = 50
    elif 50<rate<=60:
        rate = 60
    elif 60<rate<=70:
        rate = 70
    elif 70<rate<=80:
        rate = 80
    elif rate>80:
        rate = 100
    return rate/100
# 自由流通比例

'''各公司调整股本'''
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值

# 饼图
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签

plt.figure(figsize=(10,10))
ans = [i for i in fluent.values()]
label = [i for i in fluent.keys()]
plt.pie(ans,labels=label,autopct='%1.1f%%',shadow=False,startangle=150)
plt.title('各证券流通股份占比')
plt.show()

### 板块指数
date = {}
start_index = df[0][df[0]['交易时间'] == '2019-04-01'].index.values[0]
end_index = df[0][df[0]['交易时间'] == '2021-04-30'].index.values[0]
for i in range(start_index,end_index+1):    #先获取日期
    ans = str(df[0]['交易时间'][i])[0:10]
    date[ans] = []

```

下面这段程序获取了问题 1 时间段内的收盘价，缺失的以 0 进行填充

```
for dat in date.keys():
    each_day = []
    for i in range(37):
        try:
            spj = df[i][df[i]['交易时间']==dat]['收盘价'].values[0]
        except:
            spj = 0
        each_day.append(spj)
    date[dat] = each_day
with open('37 家公司在 2019-04-01_2021-04-30 的收盘价.json','w') as f: #以字典保存 37 个公司的
    收盘价
    json.dump(date,f,ensure_ascii=False)
```

假设基期指数为 1000

```
base = 1000
tend = {}
# 求基数
'''基数'''
base_num=0
for i,key in enumerate(fluent.keys()):
    base_num += date['2019-04-01'][i] * fluent[key]
for dat, lis in date.items():
    each_day_score = 0
    for i,coffe in enumerate(fluent.values()):
        each_day_score += lis[i]*coffe/base_num
    tend[dat] = each_day_score*base
tend_array = np.array([i for i in tend.values()])
def avg_line(days):
    arr = []
    for i in range(len(tend_array)-days+1):
        arr.append(sum(tend_array[i:i+days])/days)
    arr = np.array(arr)
    return arr
''' 5 日移动平均线'''
d5 = avg_line(5)
'''10 日移动平均线'''
d10 = avg_line(10)
'''20 日移动平均线'''
d20 = avg_line(20)
```

作图

```
import matplotlib.pyplot as plt
```

```

plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.plot(tend_array);plt.show()
plt.plot(d5);plt.show()
plt.plot(d10);plt.show()
plt.plot(d20);plt.show()

#作图
plt.figure(figsize=(7,4),dpi = 150)
plt.plot(tend_array);plt.ylim((750,2100))
plt.title('光伏建筑一体化板块指数');plt.show()
plt.figure(figsize=(7,4),dpi = 150)
plt.plot(range(504),d5,'--',lw = 1.5,alpha = 0.8,label= '5 日移动平均线')
plt.plot(range(5,504),d10,'--',lw = 1.5,alpha = 0.8,label= '10 日移动平均线')
plt.plot(range(15,504),d20,'--',lw = 1.5,alpha = 0.8,label= '20 日移动平均线')
plt.title('5 日、10 日、20 日的移动平均线')
plt.ylim((750,2100));plt.legend();plt.show()

```

附录 2

```

import json
from utils import get_rate
import pandas as pd
import numpy as np
#'''    读取 37 个公司的股价数据
df = []
for i in range(1,38):
    tmp = pd.read_excel('附件 1.xlsx',sheet_name='Sheet0 {}'.format(i))
    df.append(tmp)
#'''

dict_z = {'南玻 A': 307069.21,'深圳能源': 475738.99,'东旭蓝天': 148687.39,'方大集团': 107387.42,'
深赛格': 123565.62,'宝鹰股份': 134129.69,'东南网架': 103440.22,'延华智能': 71215.30,'拓日新能':
123634.21,'中利集团': 87178.71,'亚厦股份': 133999.65,'广田集团': 153727.97,'瑞和股份': 37829.20,'亚
玛顿': 16000.00,'永高股份': 123538.39,'中装建设': 72144.58,'南网能源': 378787.88,'特锐德':
104071.07,'嘉寓股份':71676.00,'东方日升':90135.99,'秀强股份':61850.24,'海达股份':60123.42,'旋极信
息':172759.06,'中来股份':108962.74,'华自科技':25617.15,'启迪设计':17448.02,'汉嘉设计':22573.83,'精
工钢构':201287.43,'苏美达':130674.94,'隆基股份':386639.48,'林洋能源':174888.93,'明阳智能
':195092.87,'江河集团':115405.00,'中衡设计':27680.77,'森特股份':53880.00,'芯能科技':50000.00,'清源
股份':27380,}

dict_lt = {'南玻 A': 306736.81,'深圳能源': 475738.99,'东旭蓝天': 106034.07,'方大集团': 107157.21,'深
赛格': 78466.18,'宝鹰股份': 133561.95,'东南网架': 95851.10,'延华智能': 71125.66,'拓日新能':
121571.98,'中利集团': 69827.66,'亚厦股份': 133099.12,'广田集团': 152929.43,'瑞和股份': 30983.80,'亚
玛顿': 15957.36,'永高股份': 112973.09,'中装建设': 63666.45,'南网能源': 75757.58,'特锐德': 94472.35,'嘉
寓股份': 71676.00,'东方日升':89944.59,
'秀强股份':59313.41,'海达股份':48815.19,'旋极信息':125369.03,'中来股份':77977.89,'华自科技

```

```

':24500.19,'启迪设计':15838.22,'汉嘉设计':20936.17,'精工钢构':201287.43,'苏美达':130674.94,'隆基股份':386630.92,'林洋能源':174888.93,'明阳智能':147260.83,'江河集团':115405.00,'中衡设计':27479.82,'森特股份':53880.00,'芯能科技':30830.00,'清源股份':27380,
}

```

```

with open('./网上搜得的信息/各公司总股本（最新）.json','w') as f:
    json.dump(dict_z,f,ensure_ascii=False)
with open('./网上搜得的信息/各公司流通股本（最新）.json','w') as f:
    json.dump(dict_lt,f,ensure_ascii=False)

```

```

#%%%

```

```

# 2019-04-01 调整市值计算

```

```

with open('./网上搜得的信息/各公司总股本.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本.json') as f:
    dict_lt = json.load(f)
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
"""求取每个公司的调整市值（2019-04-01）"""
for i,key in enumerate(fluent.keys()):
    fluent[key] = df[i][df[i]['交易时间'] == '2019-04-01']['收盘价'].values[0] * fluent[key]
"""总调整市值（2019-04-01）"""
total_fluent_3_ = sum([i for i in fluent.values()])

```

```

# 2021-05-05 调整市值计算

```

```

with open('./网上搜得的信息/各公司总股本（最新）.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本（最新）.json') as f:
    dict_lt = json.load(f)
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
"""求取每个公司的调整市值（2021-05-06）"""
for i,key in enumerate(fluent.keys()):
    fluent[key] = df[i][df[i]['交易时间'] == '2021-05-06']['收盘价'].values[0] * fluent[key]
"""总调整市值（2021-05-06）"""
total_fluent_2_ = sum([i for i in fluent.values()])

```

```

# 以 2019-04-01 的股本计算 2021-05-06

```

```

with open('./网上搜得的信息/各公司总股本.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本.json') as f:
    dict_lt = json.load(f)
fluent = {}
for key in dict_z.keys():

```

```

        fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
'''求取每个公司的调整市值（2021-05-06）'''
for i,key in enumerate(fluent.keys()):
    fluent[key] = df[i][df[i]['交易时间'] == '2021-05-06']['收盘价'].values[0] * fluent[key]
'''总调整市值（2019-05-06）'''
total_fluent_1_ = sum([i for i in fluent.values()])

'''计算新除数 new_div'''
new_div = total_fluent_3_*total_fluent_2_/total_fluent_1_
# 修正后的板块指数计算方法：    total_fluent_2_/new_div*1000
#
'''2021-05-06 到 2021-05-27 的收盘价'''
date = {}
start_index = df[0][df[0]['交易时间'] == '2021-05-06'].index.values[0]
end_index = df[0][df[0]['交易时间'] == '2021-05-27'].index.values[0]
for i in range(start_index,end_index+1):    # 先获取日期
    ans = str(df[0]['交易时间'][i])[0:10]
    date[ans] = []
# 下面这段程序获取了问题1 时间段内的收盘价，缺失的以0 进行填充
for dat in date.keys():
    each_day = []
    for i in range(37):
        try:
            spj = df[i][df[i]['交易时间']==dat]['收盘价'].values[0]
        except:
            spj = 0
        each_day.append(spj)
    date[dat] = each_day
with open('37 家公司在 2021-05-06_2021-05-27 的收盘价.json','w') as f:    # 以字典保存 37 个公司的
    收盘价
    json.dump(date,f,ensure_ascii=False)
### 以 2019-04-01 的股本数据做的数据
with open('./网上搜得的信息/各公司总股本.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本.json') as f:
    dict_lt = json.load(f)
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
'''求取每个公司的调整市值（2021-05-06）'''
final_score = []
fluent_value = {}
for dat in date.keys():
    for i,key in enumerate(fluent.keys()):

```



```

        fluent_value[key] = df[i][df[i]['交易时间'] == dat]['收盘价'].values[0] * fluent[key]
    total = sum([i for i in fluent_value.values()])
    final_score.append(total/new_div * 1000)
'''板块指数 old'''
final_score_old = final_score

# 以 2021-05-27 的最新股本数据修正后的数据
with open('./网上搜得的信息/各公司总股本（最新）.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本（最新）.json') as f:
    dict_lt = json.load(f)
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
'''求取每个公司的调整市值（2021-05-06）'''
final_score = []
fluent_value = {}
for dat in date.keys():
    for i, key in enumerate(fluent.keys()):
        fluent_value[key] = df[i][df[i]['交易时间'] == dat]['收盘价'].values[0] * fluent[key]
    total = sum([i for i in fluent_value.values()])
    final_score.append(total/new_div * 1000)
'''板块指数 new'''
final_score_new = final_score

### 作图
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei'] # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False # 设置支持负号显示

plt.plot(final_score_old, label = '待修正数据')
plt.plot(final_score_new, label = '修正后数据')
plt.title('模型修正前后板块指数对比')
plt.legend()
plt.show()

final_score_old = np.array(final_score_old)
final_score_new = np.array(final_score_new)

print('原模型得到数据的\n 均值： {} \t 方差： {} \t 标准差： {} \t 最值差： {}'.format(
    final_score_old.mean(), final_score_old.var(), final_score_old.std(),
    final_score_old.max() - final_score_old.min()))
print('修正模型得到数据的\n 均值： {} \t 方差： {} \t 标准差： {} \t 最值差： {}'.format(
    final_score_new.mean(), final_score_new.var(), final_score_new.std(),

```

```

final_score_new.max() - final_score_new.min()))

#####
zf_old = [final_score_old[i+1] - final_score_old[i] for i in range(len(final_score_old)-1)]
zf_new = [final_score_new[i+1] - final_score_new[i] for i in range(len(final_score_new)-1)]

plt.bar(range(1,45,3),zf_old,label = '待修正')
plt.bar(range(2,46,3),zf_new,label = '已修正')
plt.title('模型修正前后板块指数涨幅对比')
plt.legend();plt.show()

### 预测问题： 后续数据准备

'''2020-05-27 到 2021-05-27 的收盘价'''
date = {}
start_index = df[0][df[0]['交易时间'] == '2020-05-27'].index.values[0]
end_index = df[0][df[0]['交易时间'] == '2021-05-27'].index.values[0]
for i in range(start_index,end_index+1):    #先获取日期
    ans = str(df[0]['交易时间'][i])[0:10]
    date[ans] = []

# 下面这段程序获取了问题1 时间段内的收盘价，缺失的以0 进行填充
for dat in date.keys():
    each_day = []
    for i in range(37):
        try:
            spj = df[i][df[i]['交易时间']==dat]['收盘价'].values[0]
        except:
            spj = 0
        each_day.append(spj)
    date[dat] = each_day
with open('37 家公司在 2020-05-27_2021-05-27 的收盘价.json','w') as f:    #以字典保存37 个公司的
    收盘价
    json.dump(date,f,ensure_ascii=False)

# 该时间段内板块指数变化情况（修正模型下）
with open('./网上搜得的信息/各公司总股本（最新）.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本（最新）.json') as f:
    dict_lt = json.load(f)
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    #每个股份的调整市值
'''时间段内的变化'''
final_score = []

```

```

for dat, arr in date.items():
    total = 0
    for i, key in enumerate(fluent.keys()):
        total += arr[i]*fluent[key]
    final_score.append(total/new_div * 1000)
'''板块指数 new'''
data_train = np.array(final_score)

plt.figure(figsize=(8,3))
plt.plot(data_train)
plt.title('以修正模型评估的近一年时间内的板块指数');plt.show()
# 时序预测
import torch
from torch import nn
from torch.utils.data import TensorDataset, DataLoader
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
step = 50      #20 个数据差不多为1 个月的量，拿前一个月的数据预测后一个结果
X,Y = [],[]
for i in range(len(data_train)-step):
    X.append(data_train[i:i+step])
    Y.append(data_train[i+step])
X,Y = torch.from_numpy(np.array(X)),torch.from_numpy(np.array(Y))
batch_size = len(X)
train = TensorDataset(X,Y)
train = DataLoader(train,batch_size=batch_size,shuffle=True,drop_last=True)
import torch.nn.functional as F
# 第二个模型
class Model2(nn.Module):
    def __init__(self):
        super(Model2, self).__init__()
        self.gru1 = nn.GRU(1,128,bidirectional=True,num_layers=3)
        self.out = nn.Sequential(nn.Flatten(),
                                   nn.Linear(128*2*step,640),
                                   nn.Tanh(),
                                   nn.Linear(640,128),
                                   nn.Tanh(),
                                   nn.Linear(128,1))

    def forward(self,x):
        gru_out = self.gru1(x)[0]
        out = F.tanh(self.out(gru_out))
        return out

self = Model2().to(device)
self(x)

```

```

model = Model2().to(device)
optim = torch.optim.RMSprop(model.parameters(), lr=1e-2)
loss_fn = nn.MSELoss()
epochs = 500

torch.autograd.set_detect_anomaly(True)
LOSS = []
for epo in range(epochs):
    l=0
    for x,y in train:
        x = x.view(batch_size,-1,1).type(torch.FloatTensor).to(device)
        y = y.view(batch_size).type(torch.FloatTensor).to(device)
        y_pred = model(x)
        loss = loss_fn(y_pred,y)
        loss.backward(retain_graph=True)
        optim.step()
        optim.zero_grad()
        with torch.no_grad():
            l += loss.cpu().numpy()/len(X)
    with torch.no_grad():
        LOSS.append(loss.cpu().numpy())
    print('epochs:{}\t MAE_loss:{:.2f}'.format(epo,l))

```

损失函数曲线

```

plt.figure(figsize=(10,3))
plt.plot(LOSS)
plt.show()

```

预测值

```

pred = model(X.type(torch.FloatTensor).unsqueeze(-1).to(device))
with torch.no_grad():
    pred = pred.cpu().numpy()
    #Y = Y.cpu().numpy()
plt.plot(pred)
plt.plot(Y)
plt.show()

```

tensorflow 实现

```

import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

```

```

from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout, GRU
from tensorflow.keras.optimizers import SGD

step = 50      #20 个数据差不多为 1 个月的量，拿前一个月的数据预测后一个结果
X,Y = [],[]
for i in range(len(data_train)-step):
    X.append(data_train[i:i+step])
    Y.append(data_train[i+step])
X,Y = np.array(X),np.array(Y)
batch_size = len(X)

X = X.reshape(batch_size,step,1)
Y = Y.reshape(-1)

model = Sequential()
# LSTM 第一层
model.add(LSTM(128, return_sequences=True, input_shape=(X.shape[1], 1)))
model.add(Dropout(0.2))
# LSTM 第二层
model.add(LSTM(128, return_sequences=True))
model.add(Dropout(0.2))
# LSTM 第三层
model.add(LSTM(128))
model.add(Dropout(0.2))
# Dense 层
model.add(Dense(units=1))

model.summary()

# 编译训练
model.compile(optimizer='rmsprop', loss='mae', learning_rate = 1e-4)
# 模型训练
history = model.fit(X,Y, epochs=4000)
model.save('model.h5')

pred = model(X)
plt.plot(Y)
plt.plot(tf.reshape(pred,[-1]));
plt.show()

### 预测
res = []

```

```

data = X[-1].reshape(1,-1,1)
for i in range(60):
    out = model(data).numpy()[0,0]
    data = np.concatenate((data[:,1:,:], out.reshape(1,1,1)), axis=1)
    print(data[0,0,0])
    res.append(out)

```

20 个交易日的日移动平均线

```

plt.figure(figsize=(8,5))
plt.plot(res[:20])
plt.title('5 月 28 日后 20 个交易日的日移动平均线');plt.show()

```

3 周的周移动平均线

```

res7=[]
for i in range(15-5+1):
    res7.append(sum(res[i:i+5])/5)
plt.figure(figsize=(8,5))
plt.plot(res7)
plt.title('5 月 28 日后 3 周的周移动平均线');plt.show()

```

2 个月的月移动平均线

```

res30=[]
for i in range(44-20+1):
    res30.append(sum(res[i:i+20])/20)
plt.figure(figsize=(8,5))
plt.plot(res30)
plt.title('5 月 28 日后 2 个月的月移动平均线');plt.show()

```

附录 3

```

import json
from utils import get_rate
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False # 设置支持负号显示

#'''    读取 37 个公司的股价数据
df = []
for i in range(1,38):
    tmp = pd.read_excel('附件 1.xlsx',sheet_name='Sheet0 {}'.format(i))
    df.append(tmp)

```

```

#'''

### 前一模型，前段时间（2019-04-01 —— 2020-04-30）
with open('./网上搜得的信息/各公司总股本.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本.json') as f:
    dict_lt = json.load(f)

'''各公司调整股本'''
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值

date = {}
start_index = df[0][df[0]['交易时间'] == '2019-04-01'].index.values[0]
end_index = df[0][df[0]['交易时间'] == '2020-04-30'].index.values[0]
for i in range(start_index,end_index+1):    # 先获取日期
    ans = str(df[0]['交易时间'][i])[0:10]
    date[ans] = []

for dat in date.keys():
    each_day = []
    for i in range(37):
        try:
            spj = df[i][df[i]['交易时间']==dat]['收盘价'].values[0]
        except:
            spj = 0
        each_day.append(spj)
    date[dat] = each_day

base = 1000
tend = {}
# 求基数
'''基数'''
base_num=0
for i,key in enumerate(fluent.keys()):
    base_num += date['2019-04-01'][i] * fluent[key]

for dat, lis in date.items():
    each_day_score = 0
    for i,coffe in enumerate(fluent.values()):
        each_day_score += lis[i]*coffe/base_num
    tend[dat] = each_day_score*base

```

"""2019-04-01 —— 2020-04-30 时间段内的板块指数"""

```
tend_array = np.array([i for i in tend.values()])
```

后一模型，后段时间（2020-05-01 —— 2021-05-27）

计算 new_div

2019-04-01 调整市值计算

```
with open('./网上搜得的信息/各公司总股本.json') as f:
```

```
    dict_z = json.load(f)
```

```
with open('./网上搜得的信息/各公司流通股本.json') as f:
```

```
    dict_lt = json.load(f)
```

```
fluent = {}
```

```
for key in dict_z.keys():
```

```
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
```

"""求取每个公司的调整市值（2019-04-01）"""

```
for i, key in enumerate(fluent.keys()):
```

```
    fluent[key] = df[i][df[i]['交易时间'] == '2019-04-01']['收盘价'].values[0] * fluent[key]
```

"""总调整市值（2019-04-01）"""

```
total_fluent_3_ = sum([i for i in fluent.values()])
```

2021-05-05 调整市值计算

```
with open('./网上搜得的信息/各公司总股本（最新）.json') as f:
```

```
    dict_z = json.load(f)
```

```
with open('./网上搜得的信息/各公司流通股本（最新）.json') as f:
```

```
    dict_lt = json.load(f)
```

```
fluent = {}
```

```
for key in dict_z.keys():
```

```
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
```

"""求取每个公司的调整市值（2021-05-06）"""

```
for i, key in enumerate(fluent.keys()):
```

```
    fluent[key] = df[i][df[i]['交易时间'] == '2021-05-06']['收盘价'].values[0] * fluent[key]
```

"""总调整市值（2021-05-06）"""

```
total_fluent_2_ = sum([i for i in fluent.values()])
```

以 2019-04-01 的股本计算 2021-05-06

```
with open('./网上搜得的信息/各公司总股本.json') as f:
```

```
    dict_z = json.load(f)
```

```
with open('./网上搜得的信息/各公司流通股本.json') as f:
```

```
    dict_lt = json.load(f)
```

```
fluent = {}
```

```
for key in dict_z.keys():
```

```
    fluent[key] = dict_z[key] * get_rate(dict_lt[key]/dict_z[key])    # 每个股份的调整市值
```

"""求取每个公司的调整市值（2021-05-06）"""

```
for i, key in enumerate(fluent.keys()):
```



```

        fluent[key] = df[i][df[i]['交易时间'] == '2021-05-06']['收盘价'].values[0] * fluent[key]
'''总调整市值（2019-05-06）'''
total_fluent_1_ = sum([i for i in fluent.values()])

'''计算新除数 new_div'''
new_div = total_fluent_3 * total_fluent_2_ / total_fluent_1_
# 修正后的板块指数计算方法: total_fluent_2_ / new_div * 1000

'''2020-05-06 —— 2021-05-27'''
date = {}
start_index = df[0][df[0]['交易时间'] == '2020-05-06'].index.values[0]
end_index = df[0][df[0]['交易时间'] == '2021-05-27'].index.values[0]
for i in range(start_index, end_index + 1):    # 先获取日期
    ans = str(df[0]['交易时间'][i])[0:10]
    date[ans] = []

# 下面这段程序获取了问题1 时间段内的收盘价，缺失的以0 进行填充
for dat in date.keys():
    each_day = []
    for i in range(37):
        try:
            spj = df[i][df[i]['交易时间'] == dat]['收盘价'].values[0]
        except:
            spj = 0
        each_day.append(spj)
    date[dat] = each_day
with open('./网上搜得的信息/各公司总股本（最新）.json') as f:
    dict_z = json.load(f)
with open('./网上搜得的信息/各公司流通股本（最新）.json') as f:
    dict_lt = json.load(f)
fluent = {}
for key in dict_z.keys():
    fluent[key] = dict_z[key] * get_rate(dict_lt[key] / dict_z[key])    # 每个股份的调整市值
'''求取每个公司的调整市值（2021-05-06）'''
final_score = []
fluent_value = {}

for dat in date.keys():
    each_day = []
    for i in range(37):
        try:
            spj = df[i][df[i]['交易时间'] == dat]['收盘价'].values[0]
        except:
            spj = 0

```

```

        each_day.append(spi)
    date[dat] = each_day
mat = np.array([i for i in fluent.values()])
for dat in date.values():
    fluent_value = 0
    for i,key in enumerate(fluent.keys()):
        fluent_value = np.array(dat) * mat
    fluent_value = sum(fluent_value)
    final_score.append(fluent_value/new_div*1000)

'''板块指数 new'''
final_score_new = final_score
#%% 读取上证指数(要注意这个时间的序号是反着的)
dataframe = pd.read_excel('上证指数.xlsx')
score = dataframe['收盘价'].values
# 前一时间段
tend_array = np.array(tend_array)
tend_array_real = np.array(score[-len(tend_array):].values)
# 后一时间段
final_score_new = np.array(final_score_new)
final_score_new_real = np.array(score[1:len(final_score_new)+1].values)

from scipy.stats import pearsonr
from minepy import MINE
from sklearn.preprocessing import MinMaxScaler

# 标准化
tend_array_std = (tend_array-tend_array.mean())/tend_array.std()
tend_array_real_std = (tend_array_real - tend_array_real.mean())/tend_array_real.std()

final_score_new_std = (final_score_new - final_score_new.mean())/final_score_new.std()
final_score_new_real_std = (final_score_new_real - final_score_new_real.mean())/final_score_new_real.std()

# 作图
def get_corr(tend_array_std,tend_array_real_std,final_score_new_std,final_score_new_real_std):
    """
    plt.figure(figsize=(8,5))
    plt.plot(tend_array_std,label='光伏板块指数');plt.plot(tend_array_real_std,label='上证指数')
    plt.legend()
    plt.title('标准化后 2019-04-01 至 2020-04-30 的上证指数与光伏板块指数走势对比');plt.show()
    plt.figure(figsize=(8,5))
    plt.plot(final_score_new_std,label='光伏板块指数');plt.plot(final_score_new_real_std,label='上证指数')
    plt.legend()
    plt.title('标准化后 2020-05-06 至 2021-05-27 的上证指数与光伏板块指数走势对比');plt.show()

```

```

'''
# 皮尔逊相关系数
p1 = pearsonr(tend_array_std,tend_array_real_std)[0]
p2 = pearsonr(final_score_new_std,final_score_new_real_std)[0]
print('前段时间皮尔逊相关系数{:.4f}'.format(p1))
print('后段时间皮尔逊相关系数{:.4f}'.format(p2))

# 互信息和最大信息系数

m=MINE()
m.compute_score(tend_array_std,tend_array_real_std)
m1 = m.mic()
print('前段时间最大信息系数{:.4f}'.format(m.mic()))
m=MINE()
m.compute_score(final_score_new_std,final_score_new_real_std)
m2 = m.mic()
print('后段时间最大信息系数{:.4f}'.format(m.mic()))

# 斯皮尔曼
x1 = pd.Series(tend_array_std)
y1 = pd.Series(tend_array_real_std)
x2 = pd.Series(final_score_new_std)
y2 = pd.Series(final_score_new_real_std)
rsep1 = x1.corr(y1,method='spearman')
rsep2 = x2.corr(y2,method='spearman')
print('前段时间斯皮尔曼系数{:.4f}'.format(rsep1))
print('后段时间斯皮尔曼系数{:.4f}'.format(rsep2))

rkend1 = x1.corr(y1,method='kendall')
rkend2 = x2.corr(y2,method='kendall')
print('前段时间肯德尔系数{:.4f}'.format(rkend1))
print('后段时间肯德尔系数{:.4f}'.format(rkend2))

# 绝对值后平均
avg1 = abs(p1)+abs(m1)+abs(rsep1)+abs(rkend1)
avg2 = abs(p2)+abs(m2)+abs(rsep2)+abs(rkend2)
print('前段时间取绝对值后平均值{:.4f}'.format(avg1/4))
print('后段时间取绝对值后平均值{:.4f}'.format(avg2/4))
print('\n')

# 数据
l1 = len(tend_array_std)
l2 = len(final_score_new)
for i in range(6):

```

```

print('{} 月 - {} 月'.format(2*i,2*i+2))
get_corr(tend_array_std[i:i+11//6], tend_array_real_std[i:i+11//6],
         final_score_new_std[i:i+12//6], final_score_new_real_std[i:i+12//6])

```

附录 4

```

clear, clc
d=[10 20 40 80 160 320];
H=[];
for i=1:6
    h=pg(d(i));
    H=[hh;h];
end
for j=1:37
    f(j)=mean(H(:,j));
end
f

function h=pg(q)
load('SHUJU.mat')
r=yy;
[n,m]=size(r);
R=zeros(n,m);
for j=2:size(r,2)
    R(:,j)=(r(:,j)-r(:,j-1))./r(:,j-1);
end
l=0.2/q;
p=zeros(size(r,1),q);
for i=1:size(r,1)
    for j=1:size(r,2)
        for k=1:q
            if R(i,j)<(-0.1+(k-1)*l)
                p(i,k)=p(i,k)+1;
                break
            end
            %p(i,k)=p(i,k)+1;
        end
    end
    p(i,10)=p(i,10)+size(r,2)-sum(p(i,:));
end
p=p/size(r,2);
for i=1:size(r,1)
    for x=1:q
        if p(i,x)==0

```

```
        v(x)=0;
    else
        v(x)=log(p(i,x));
    end
end
h(i)=-sum(p(i,:)*v');
end
```