中国A股AI量化交易系统构建与实施深度调研报告

1. 概述与系统架构设计

1.1 研究背景与目标

随着人工智能技术的飞速发展，金融市场的交易模式正经历着深刻的变革。对于个人投资者而言，利用机器学习算法挖掘市场非线性规律，结合自然语言处理（NLP）技术解析海量财经新闻，已成为构建超额收益策略的重要途径。本报告旨在响应构建一个针对中国A股市场的AI自动交易决策系统的需求，具体目标是利用前一交易日的收盘数据，预测未来一周（5个交易日）的股价走势，并结合持仓股票的每日新闻舆情进行风险监控。

中国A股市场具有独特的微观结构，如T+1交易制度、涨跌停板限制以及散户占比较高等特征，这使得直接套用美股的量化模型往往失效。因此，本系统设计将严格遵循A股的市场规则，采用模块化设计思想，整合数据采集（**ETL**）、特征工程（**Alpha Factor**）、预测建模（**Model Training**）、**情感分析（Sentiment Analysis）以及策略回测（Backtesting）**五大核心模块。

1.2 系统总体架构

为了实现从数据获取 to 交易信号生成的全流程自动化，本研究推荐采用“松耦合、高内聚”的架构设计。

系统逻辑流如下：

1. 数据层 **(Data Layer)**：利用开源库 **AkShare** 作为核心数据源，每日定时抓取A股全市场行情数据（OHLCV）及个股新闻数据。数据需经过清洗、复权处理后存入本地数据库（如PostgreSQL或HDF5文件系统）。
2. 特征层 **(Feature Layer)**：依托微软开源的 **Qlib** 量化平台，构建基于量价的Alpha因子库（如Alpha158）。针对“未来一周上涨”的目标，设计跨度为5天的收益率标签（Label），并进行标准化处理以消除个股绝对价格差异。
3. 模型层 **(Model Layer)**：采用 **LightGBM**（梯度提升决策树）作为主预测模型，因其在处理表格型金融数据时表现出的鲁棒性和抗噪能力优于普通深度学习模型。同时，集成 **FinBERT** 预训练模型处理中文财经新闻，输出情感评分。
4. 策略层 **(Strategy Layer)**：基于模型预测的收益率排名（Ranking），结合 **Top-K Dropout** 策略进行周频换仓。在此过程中，融入NLP情感分值作为过滤机制，剔除舆情恶化的标的。
5. 执行与回测层 **(Execution Layer)**：在Qlib或Backtrader回测引擎中模拟真实交易环境，严格执行T+1规则，扣除印花税与佣金，验证策略的夏普比率（Sharpe Ratio）与最大回撤（Max Drawdown）。

1.3 技术选型论证

- 开发语言：Python。作为数据科学与量化金融的事实标准语言，Python拥有最丰富的生态支

持[1]。

- 量化框架：**Qlib**。相较于Backtrader仅侧重于回测，Qlib提供了一整套从数据处理到AI模型训练的完整工作流（Workflow），特别适合机器学习驱动的策略开发。其内置的RollingTrainer能有效应对金融市场的数据漂移问题[3]。
- 数据接口：**AkShare**。作为Tushare的免费替代方案，AkShare接口丰富，涵盖了东方财富、新浪财经等主流财经门户的数据，适合个人开发者低成本获取实时与历史数据[5]。
- **NLP**模型：**FinBERT**。金融文本具有特殊的语境（例如"负增长"在特定语境下可能并非完全负面），通用BERT模型往往效果不佳。FinBERT在金融语料上进行了微调，能更准确地捕捉市场情绪[7]。

---

2. A股市场微观结构与数据工程

在构建模型之前，必须深入理解A股的数据特性与交易规则，这是模型能否落地的关键。

2.1 T+1 交易制度对数据标签的影响

中国A股市场实行 T+1 交易制度，即当日（T日）买入的股票，必须等到下一个交易日（T+1日）才能卖出[9]。这一规则对短线策略（如日内高频）限制极大，但对于本研究设定的"未来一周"预测周期而言，反而是天然适配的。

在数据标注（Labeling）阶段，如果我们的目标是预测 $T+1$ 到 $T+5$ 的收益，我们需要构建跨日标签。
假设当前时间为 T 日收盘后：

- 决策点：T 日晚间。
- 执行点：$T+1$ 日开盘（Open）或收盘（Close）。考虑到流动性，通常假设以 $T+1$ 收盘价或加权均价（VWAP）成交。
- 持有期：5个交易日。
- 标签定义：

  $$Label_T = \frac{Price_{close, T+6}}{Price_{close, T+1}} - 1$$

  或者，在Qlib的表达式中，若以 T 日为基准，标签通常写作 Ref($close, -5) / $close - 1（注：Qlib中负数代表未来偏移）11。这种设计确保了模型预测的是"买入后持有5天的收益率"，完全避开了T+1的卖出限制。

2.2 数据获取方案：AkShare 接口深度解析

AkShare 通过爬虫技术封装了各大财经网站的API。针对本系统的需求，主要涉及以下两类数据接口。

2.2.1 历史行情数据采集

使用 stock_zh_a_hist 接口获取个股的历史行情。该接口源自东方财富网，数据质量较高，且支持

复权操作。

参数	描述	示例值
symbol	股票代码	"000001"
period	数据周期	"daily" (日线)
start_date	开始日期	"20100101"
end_date	结束日期	"20231231"
adjust	复权方式	"qfq" (前复权)

关键实现步骤：

1. 复权的重要性：A股上市公司常有分红、送股等行为，导致股价出现断崖式下跌（除权）。如果不进行复权处理，模型会错误地将除权缺口识别为暴跌信号。必须使用 前复权（**qfq**）数据，使价格序列具有连续性，以此计算出的收益率才是真实的投资回报[12]。
2. 数据清洗：A股存在停牌机制。停牌期间价格不变或数据缺失。在数据预处理阶段，需剔除停牌期间的数据，或使用前值填充（Forward Fill），但在计算收益率标签时，若未来5天内包含停牌日，需特别处理以避免标签失真。

2.2.2 个股新闻数据采集

为了实现每日监控，需要获取指定股票的新闻。AkShare 提供了 stock_news_em 接口（东方财富个股新闻）。

代码实现逻辑：

Python

```
import akshare as ak

# 获取个股新闻列表
# symbol: 股票代码
news_df = ak.stock_news_em(symbol="600519")

# 返回字段通常包括：
```

```
# title: 新闻标题
# content: 新闻内容（部分接口可能仅包含摘要）
# public_time: 发布时间
# url: 原文链接
```

需要注意的是，爬虫类接口可能受到反爬策略的限制（如IP封禁）。在生产环境中，建议设置请求间隔（time.sleep），或构建IP代理池。同时，数据采集应设定在每日收盘后（15:30）和次日开盘前（09:00）分批进行 [12]。

2.3 数据存储与管理

对于全市场5000多只股票，每日产生的数据量虽然不大，但累积数年数据量可观。建议放弃CSV文件，转而使用 **SQLite**（轻量级）或 **PostgreSQL**（生产级）数据库，或者使用 Qlib 专用的二进制格式（.bin）。

Qlib 数据转换：
Qlib 要求数据遵循特定的目录结构和二进制格式以优化加载速度。用户需编写脚本将 AkShare 下载的 Pandas DataFrame 转换为 Qlib 格式：

1. 将每个股票的数据保存为独立的 CSV，包含 date, open, close, high, low, volume, factor（复权因子）列。
2. 使用 Qlib 提供的 dump_bin.py 脚本将 CSV 转换为 .bin 文件。
   Bash
   ```
   python scripts/dump_bin.py dump_all --csv_path ~/.qlib/csv_data --qlib_dir ~/.qlib/qlib_data/cn_data
   ```

这一步是系统跑通的关键，因为后续的模型训练将直接读取这些二进制文件 [14]。

---

3. 基于Qlib的量化特征工程与标签体系

特征工程（Feature Engineering）是将原始数据转化为模型可理解的信号的过程。在量化领域，这通常意味着构建“因子”（Factors）。

3.1 预测目标的设计：周频收益率标签

用户的核心需求是“选出未来1周内可能上涨的股票”。在监督学习框架下，这定义了一个回归（Regression）或排序（Ranking）问题。

标签定义（Label Definition）：
在 Qlib 的配置文件（YAML）中，我们定义标签为未来5日的收益率。为了消除市场整体波动的影响（例如大盘整体上涨带来的个股普涨），通常会采用 超额收益（Excess Return） 或 收益率排名（Rank） 作为目标。

Qlib 标准配置代码如下：

YAML

```yaml
data_handler_config:
    # 标签：(未来5日收盘价 / 明日收盘价) - 1
    # Ref($close, -5) 表示未来第5天的收盘价
    # Ref($close, -1) 表示未来第1天（即T+1日）的收盘价
    # 含义：在T+1日收盘买入，持有至T+5日收盘卖出的收益率
    label:
```

这里使用 Ref($close, -1) 作为分母，是因为在 T 日收盘后运行模型，最早的买入机会是 $T+1$ 日。如果假设以 $T+1$ 开盘价买入，分母应改为 $open 的相应偏移 [11]。

3.2 Alpha158 因子体系

对于输入特征（Features），我们不需要手动计算每一个技术指标。Qlib 内置了 **Alpha158** 因子库，这是一套经典的量价因子集合，包含 158 个基于每日量价数据生成的因子。

Alpha158 涵盖了以下几类信息：

1. 趋势类 **(Trend)**：不同周期的移动平均线（MA5, MA10, MA60）及其偏离度。例如 ($close - Mean($close, 5)) / Mean($close, 5)。
2. 动量类 **(Momentum)**：如 ROC（变动率）、RSI（相对强弱指标）。捕捉股价上涨的加速度。
3. 波动率 **(Volatility)**：如 STD（标准差）、ATR（平均真实波幅）。用于衡量风险。
4. 量价配合 **(Volume-Price)**：如 VWAP（成交量加权平均价）与收盘价的关系。例如 ($close - $vwap) / $vwap。

使用 Alpha158 的优势在于它是一个经过验证的、覆盖面广的特征集，能有效捕捉A股市场的技术面特征。用户无需手动编写 talib 代码，只需在 Qlib 配置中调用 qlib.contrib.data.handler.Alpha158 即可 [3]。

3.3 数据预处理与标准化

A股市场个股价格差异巨大（如贵州茅台2000元 vs 银行股3元），直接将价格输入模型会导致严重的偏差。必须进行标准化处理。

推荐的处理链（Processor Pipeline）：

1. **DropnaLabel**：剔除标签缺失的数据（例如刚上市或停牌的股票）。
2. CSZScoreNorm (Cross-Sectional Z-Score Normalization)：横截面标准化。这是量化模型最关键的一步。它计算每一天所有股票因子的均值和标准差，将每个股票的因子值转化为

Z-Score（标准分数）。

$$z_{i,t} = \frac{x_{i,t} - \mu_t}{\sigma_t}$$

这样模型学习到的是该股票在全市场中的相对地位，而非绝对数值。这对于捕捉“未来一周表现优于大盘”的股票至关重要 17。

3. **Fillna**：填充缺失值（通常填0或均值），保证数据完整性。

---

4. 预测模型构建：LightGBM与周频Alpha生成

4.1 模型选择：为何是LightGBM？

尽管深度学习（如LSTM、Transformer）在学术界很火，但在处理量化表格数据（Tabular Data）时，梯度提升决策树（**GBDT**）依然是工业界的霸主。

针对本系统，推荐使用 **LightGBM**，原因如下：

1. 处理非线性关系：股价与因子之间往往不是线性关系（例如RSI过高可能意味着反转而非继续上涨），树模型能天然捕捉这种非线性分割。
2. 抗噪能力强：A股日线数据充满噪声，LightGBM 通过集成多棵树，能有效降低过拟合风险。
3. 训练速度快：相比 XGBoost，LightGBM 的直方图算法大大加快了训练速度，适合全市场5000只股票的大规模训练 [3]。

4.2 滚动训练（Rolling Training）机制

金融市场存在 概念漂移（**Concept Drift**）现象，即市场规律会随时间变化（例如2015年的杠杆牛市规律不适用于2018年的熊市）。因此，不能简单地用过去10年的数据训练一个模型就一直使用。

本系统采用 滚动窗口（**Rolling Window**）训练模式：

- 训练集：过去 N 年（例如 2015-2019）。
- 验证集：随后 1 年（例如 2020）。
- 测试集：再下 1 年（例如 2021）。
- 滚动：完成测试后，将时间窗口整体向后平移一年，重新训练模型。

Qlib 的 RollingTrainer 类原生支持这种模式，确保模型始终适应最近的市场风格 [17]。

4.3 模型输出

模型对于每一只股票，会输出一个 预测分数（**Prediction Score**）。这个分数并不直接等于预测的涨跌幅数值，而是代表该股票在未来一周相对于其他股票的排序。分数越高，预期该股票在未来一周的表现越好（即排名越靠前）。这一排名是后续选股策略的核心依据。

---

5. 每日新闻舆情监控：NLP情感分析模块

单纯的量价模型无法感知突发新闻（如公司暴雷、政策利好）。因此，必须引入 NLP 模块作为“风险过滤器”。

5.1 模型选择：FinBERT-Tone-Chinese

通用 NLP 模型（如普通的 BERT）在金融语境下往往失效。例如，“成本大幅下降”中包含“下降”（通常为负面词），但在金融语境下是重大利好。

本系统采用 **FinBERT** 的中文衍生版本（如 yiyanghkust/finbert-tone-chinese）。该模型是在大规模中文金融语料（研报、财经新闻、公告）上进行预训练和微调的，能够准确识别金融文本的情感倾向 [7]。

5.2 情感评分计算流程

1. 文本输入：每日通过 AkShare 获取持仓股票的最新新闻标题和摘要。
2. 分词与编码：使用 Hugging Face 的 BertTokenizer 将文本转换为模型可读的 Tensor。
3. 推理（**Inference**）：将 Tensor 输入 FinBERT 模型，获取 Positive（积极）、Negative（消极）、Neutral（中性）三个类别的概率（Logits）。
4. 合成得分：

   $$\text{Sentiment Score} = P(\text{Positive}) - P(\text{Negative})$$

   得分范围在 $[-1, 1]$ 之间。正值代表利好，负值代表利空。

5.3 自动化脚本实现逻辑

这部分需要编写一个独立的 Python 脚本，每日定时运行：

Python

```
# 伪代码逻辑展示
from transformers import BertTokenizer, BertForSequenceClassification
import torch

# 加载模型
tokenizer = BertTokenizer.from_pretrained("yiyanghkust/finbert-tone-chinese")
model = BertForSequenceClassification.from_pretrained("yiyanghkust/finbert-tone-chinese")

def analyze_sentiment(text):
```

```
    inputs = tokenizer(text, return_tensors="pt", max_length=512, truncation=True)
    outputs = model(**inputs)
    probs = torch.nn.functional.softmax(outputs.logits, dim=-1)
    # 假设索引：0=中性, 1=积极, 2=消极 (需根据具体模型config确认)
    score = probs.item() - probs.item()
    return score

# 每日循环检查持仓
for stock in my_portfolio:
    news_list = ak.stock_news_em(symbol=stock)
    daily_score = 0
    for news in news_list:
        daily_score += analyze_sentiment(news['title'])

    if daily_score < -0.5: # 设定阈值
        print(f"警告：{stock} 出现重大利空舆情，建议人工复核或卖出。")
```

20

---

6. 策略回测与执行：应对T+1与涨跌停限制

6.1 Top-K Dropout 策略逻辑

针对“选出未来1周可能上涨股票”的需求，最适合的策略是 **Top-K Dropout**。这是一个典型的定期换仓策略。

- 逻辑：
  1. 每日根据 LightGBM 模型预测的全市场股票分数进行降序排列。
  2. 选取分数最高的 K 只股票（例如 Top 50）。
  3. 换仓规则：检查当前持仓。如果持仓股票跌出了预测排名的 Top N（例如跌出前100名），则卖出（Dropout）。
  4. 买入规则：资金释放后，按排名从高到低买入尚未持有的 Top K 股票。
  5. 周频控制：虽然模型每日更新分数，但强制设定每隔 5 个交易日进行一次大规模调仓，或者仅当排名发生剧烈变化时调仓，以降低换手率。

6.2 回测环境配置（Configuring the Backtester）

在 Qlib 中进行回测时，必须通过配置文件严格模拟 A股环境。

关键配置参数：

- **limit_threshold**: 设置为 0.095 (9.5%)。这意味着如果一只股票当日涨幅超过 9.5%（接近涨停），回测引擎将禁止买入；跌幅超过 9.5%（接近跌停），禁止卖出。这是模拟 A股流动性风险的关键 [22]。

- **deal_price**: 设置为 close 或 vwap。由于我们是在收盘后通过模型做决策，实际交易是在次日。严格的回测应使用 Ref($open, -1) 或 Ref($vwap, -1) 作为成交价。但在 Qlib 的标准日线回测中，通常配置为当日收盘价成交（假设尾盘操作），或者次日开盘价。为了保守估计，建议设置滑点（Slippage）和交易成本。
- **cost**: A股印花税为卖出方单边征收（目前为0.05%），佣金双边征收（通常万分之二）。

```yaml
backtest:
    limit_threshold: 0.095
    account: 100000000
    benchmark: SH000300
    deal_price: close  # 或建议设为 next_open
    open_cost: 0.0002 # 买入佣金
    close_cost: 0.0007 # 卖出佣金+印花税
```

24

---

7. 系统自动化部署与风险控制

7.1 每日工作流自动化

系统部署需通过脚本串联各个模块，并利用操作系统的定时任务（如 Linux 的 crontab）实现自动化。

时间表（交易日）：

- **16:00**：触发数据更新脚本。调用 AkShare 抓取当日全市场日线数据，并更新至 Qlib 数据库。
- **16:30**：触发模型推理脚本。加载训练好的 LightGBM 模型，读取最新数据，生成全市场股票的预测分数，并输出 Next_Week_Buy_List.csv。
- **17:00**：触发舆情监控脚本。读取当前持仓及 Buy_List 中的股票代码，抓取当日新闻，计算情感得分。
- **17:30**：生成决策日报。将量化预测排名高、且情感得分为正的股票生成最终买入建议；将持仓中排名下降或情感得分为负的股票生成卖出建议。通过邮件或消息推送给用户 [21]。

7.2 风险控制机制

在模型之外，必须人为设定硬性风控规则：

1. 分散投资：限制单只股票的最大仓位（例如不超过 10%），避免黑天鹅事件导致账户归零。
2. 止损策略：无论模型预测如何，一旦个股亏损超过特定阈值（如 -8%），强制止损。
3. 指数熔断：监控大盘指数（沪深300）。若大盘处于明显下行趋势（如跌破20日均线），强制降低整体仓位（Model-Down-Market-Down）。

---

8. 总结与建议

本报告详细阐述了基于 Python 构建 A股 AI 量化交易系统的全过程。通过整合 **Qlib** 的工业级流程、**AkShare** 的数据能力以及 **FinBERT** 的语义理解能力，我们构建了一个具备“量价选股”与“舆情排雷”双重能力的自动化系统。

核心结论：

1. 可行性：利用开源工具完全可以实现机构级别的量化流程。
2. 关键点：T+1 规则下的标签设计（预测 T+1 到 T+5 的收益）是模型有效的核心。
3. 局限性：机器学习模型基于历史数据归纳规律，无法预测从未发生过的极端市场事件。

实施建议：
建议用户按照“数据落地 -> 回测验证 -> 模拟盘运行 -> 实盘接入”的节奏推进。切勿在回测结果完美的情况下直接全仓实盘，因为滑点、冲击成本和市场微观结构的变动在回测中往往被低估。初期应重点关注模型在 2022-2024 年震荡市中的表现，而非 2019-2020 年牛市中的表现。

数据表格：常用**Python**量化库对比

库名称	主要功能	**A**股支持度	适用场景	本系统角色
Qlib	AI模型训练、回测、数据管理	高（原生支持）	机器学习策略研发	核心引擎 (建模/回测)
AkShare	数据采集 (API)	极高	实时/历史数据获取	数据源 (ETL)
Backtrader	策略回测	中 (需自定义数据)	传统技术指标策略	备选回测引擎
Transformers	NLP模型调用	通用	文本情感分析	舆情分析模块
Pandas	数据清洗、分析	通用	数据预处理	基础工具

Citations:

1

引用的著作

1. Top 31 Python Libraries for Data Science in 2026 | DataCamp, 访问时间为 十二月 26, 2025, https://www.datacamp.com/blog/top-python-libraries-for-data-science
2. Best Python Libraries for Algorithmic Trading and Financial Analysis - QuantInsti Blog, 访问时间为 十二月 26, 2025, https://blog.quantinsti.com/python-trading-library/
3. Python + Qlib AI: Algorithmic Trading - Modbus & Embedded Systems, 访问时间为 十二月 26, 2025, https://modbus.pl/2025/02/07/python-qlib-ai-algorithmic-trading/
4. Demystifying Qlib: Your Guide to Microsoft's AI-Driven Quantitative Investment Platform, 访问时间为 十二月 26, 2025, https://grepix.medium.com/demystifying-qlib-your-guide-to-microsofts-ai-driven-quantitative-investment-platform-c530fd632995
5. Awesome-data shows most interesting data-source around the financial world! - GitHub, 访问时间为 十二月 26, 2025, https://github.com/akfamily/awesome-data
6. Unlocking Chinese Financial Data: A Deep Dive into the AKShare MCP Server - Skywork.ai, 访问时间为 十二月 26, 2025, https://skywork.ai/skypage/en/chinese-financial-data-akshare/1980819556317073408
7. wangy8989/Chinese-Financial-News-Sentiment-Analysis - GitHub, 访问时间为 十二月 26, 2025, https://github.com/wangy8989/Chinese-Financial-News-Sentiment-Analysis
8. ProsusAI/finBERT: Financial Sentiment Analysis with BERT - GitHub, 访问时间为 十二月 26, 2025, https://github.com/ProsusAI/finBERT
9. Understanding T+1, T+2, T+3: Securities Settlement Dates Explained - Investopedia, 访问时间为 十二月 26, 2025, https://www.investopedia.com/terms/t/tplus1.asp
10. Understanding Settlement Cycles: What Does T+1 Mean for You? | FINRA.org, 访问时间为 十二月 26, 2025, https://www.finra.org/investors/insights/understanding-settlement-cycles
11. Label calculation · Issue #1060 · microsoft/qlib - GitHub, 访问时间为 十二月 26, 2025, https://github.com/microsoft/qlib/issues/1060
12. AKShare is an elegant and simple financial data interface library for Python, built for human beings! 开源财经数据接口库 - GitHub, 访问时间为 十二月 26, 2025, https://github.com/akfamily/akshare
13. 5 akshare获取股市新闻，snownlp判断情绪 - 量化交易与投资社区- CSDN, 访问时间为 十二月 26, 2025, https://quant.csdn.net/6874aaebbb9d8e0ecec22cb5.html
14. qlib/examples/tutorial/detailed_workflow.ipynb at main · microsoft/qlib - GitHub, 访问时间为 十二月 26, 2025, https://github.com/microsoft/qlib/blob/main/examples/tutorial/detailed_workflow.ipynb
15. Why label is calculated on the day after tomorrow? · Issue #682 · microsoft/qlib - GitHub, 访问时间为 十二月 26, 2025, https://github.com/microsoft/qlib/issues/682
16. Workflow Management — QLib 0.9.8.dev11 documentation, 访问时间为 十二月 26, 2025, https://qlib.readthedocs.io/en/latest/component/workflow.html

17. Estimator: Workflow Management — QLib 0.5.1 documentation, 访问时间为 十二月 26, 2025，https://qlib.readthedocs.io/en/v0.5.1/component/estimator.html
18. microsoft/qlib. Continue this conversation at http://localhost:3000?gist=7a2da9a3b46893e06f2ba4681988cb85 · GitHub, 访问时间为 十二月 26, 2025, https://gist.github.com/m0o0scar/7a2da9a3b46893e06f2ba4681988cb85
19. Harnessing AI for Quantitative Finance with Qlib and LightGBM - Vadim's blog, 访问时间为 十二月 26, 2025，https://vadim.blog/qlib-ai-quant-workflow-lightgbm
20. yiyanghkust/finbert-tone-chinese - Hugging Face, 访问时间为 十二月 26, 2025, https://huggingface.co/yiyanghkust/finbert-tone-chinese
21. Building A Stock Automation To Predict The Daily Stock Analysis Using News API - Medium, 访问时间为 十二月 26, 2025, https://medium.com/@honneyykhatter/building-a-stock-automation-to-tell-the-daily-stock-analysis-using-news-api-f3716b7d41c7
22. qlib/qlib/backtest/exchange.py at main · microsoft/qlib - GitHub, 访问时间为 十二月 26, 2025, https://github.com/microsoft/qlib/blob/main/qlib/backtest/exchange.py
23. Portfolio Strategy: Portfolio Management — QLib 0.8.0.99 documentation, 访问时间为 十二月 26, 2025, https://qlib.readthedocs.io/en/v0.8.1/component/strategy.html
24. Workflow Management — QLib 0.9.7 documentation, 访问时间为 十二月 26, 2025，https://qlib.readthedocs.io/en/stable/component/workflow.html
25. Portfolio Strategy: Portfolio Management — QLib 0.9.8.dev11 documentation, 访问时间为 十二月 26, 2025, https://qlib.readthedocs.io/en/latest/component/strategy.html
26. 19 Super-Useful Python Scripts to Automate Your Daily Tasks - Index.dev, 访问时间为 十二月 26, 2025，https://www.index.dev/blog/python-automation-scripts