

Polymarket 套利全面研究与系统化实践

前言

预测市场自上世纪末诞生以来一直被视为检验“群众智慧”的实验场，在加密时代诞生的 **Polymarket** 更是凭借链上透明度、低交易成本和去中心化结算机制迅速崛起。作为预测市场参与者，我们既可以通过押注事件结果获取收益，也可以利用价格偏差获取**风险可控甚至 风险为零的套利收益**。这种套利行为并不会损害市场公平，相反，套利者在纠正价格错误、补充流动性方面发挥重要作用，使市场更接近真实概率。

本文面向具有编程背景的投资者，尤其是熟悉 Python 的工程师，系统整理在 Polymarket 上开展套利的**策略框架、优化路径、实现方法与风险控制**。内容涵盖从基本二元市场套利、尾盘扫货策略、多选市场组合套利、跨平台套利、时间型触碰合约定价套利、市场再平衡套利及组合套利等多种路径，并结合公开研究论文、社区文章、开源项目等文献，为读者提供完整的学习、实践与深度优化路线图。文章大量引用权威资料中的数据与观点，并给出 Python 工具链和代码实现建议，力求构建一篇超万字的知识库，以便读者在掌握原理的基础上快速上手。请注意，套利并非无风险套利的代名词，每一种策略都有特定的前提条件和潜在风险，本报告会在后文详细阐述风险管理方法。

第一部分：Polymarket 基本机制与套利框架

1.1 Polymarket 的市场结构

Polymarket 是基于 Polygon 公链的去中心化预测市场平台。用户通过买卖代币形式的合约来下注某个事件是否发生或者多个事件结果中哪一项会发生。Polymarket 每一份合约最终结算时都可以兑换 1 美元或 0 美元，因此合约价格被视为市场参与者对事件发生概率的共识估¹。例如，某个事件的 Yes 合约价格为 0.6 美元，隐含着市场认为该事件发生的概率约为 60%。由于 Polymarket 强调所有互斥结果的合约数量之和应等于 1，当市场出现概率总和不等于 1 的情况时会产生套利空间²。Polymarket 的订单簿去中心化，买卖双方通过链上撮合，无额外做市商费用，只收取极少的矿工费，且平台对提供流动性的用户给予奖励，这些特点使得价格偶尔出现偏差并为套利提供机会。

Polymarket 合约分为以下几类：

1. **二元市场 (Binary Market)**：只有两个可能的结果，例如“拜登是否赢得总统选举”。Yes 与 No 两个代币加起来固定价值 1 美元，用户购买任意一侧即可押注，结算时若对应结果发生则可兑换 1 美元，否则价值归零。
2. **多选市场 (Multiple-Choice Market)**：存在多个互斥结果，例如“美联储在 7 月会议会将利率调整到哪个区间”，各选项中只有一个会发生。每个选项的代币价格反映该选项发生的概率。
3. **时间型触碰市场 (Touch-style Market)**：在给定时间窗口内标的资产价格是否触及某阈值，例如“11 月前 BTC 是否触及 50,000 美元”。这种合约类似于金融衍生品中的数字触碰期权。
4. **组合/条件市场**：某些市场之间存在逻辑关系，例如一个市场的结果会影响另一个市场概率。这类市场可以通过组合套利或条件套利进行挖掘。

1.2 套利基本概念

套利是指利用市场的不合理定价构建 **无风险或低风险的交易组合**，在 未来无论事件结果如何都能获得确定或高度确定的利润。在 Polymarket 环境中，由于价格由用户自行出价，且每个市场的订单簿独立，使得 价格偶尔不能完全反映概率公理，从而出现套利机会。与传统金融市场一样，预测市场的套利分为以下几类：

1. **单市场均衡套利 (Market Rebalancing Arbitrage)**：在单个市场内 “Yes + No” 价格之和偏离 1 美元；套利者可以同时买入或卖出两边锁定无风险收益。研究论文对这一策略进行了形式化定义，并给出了盈利公式³。
2. **多市场组合套利 (Combinatorial Arbitrage)**：涉及两个或以上相关市场，通过构建组合头寸确保至少有一个头寸会兑现，理论上可以在不同事件组合之间发现概率矛盾并套利⁴。
3. **跨平台套利 (Cross-Platform Arbitrage)**：在不同预测市场或衍生品交易所之间存在价格差异时，通过在价格低的平台买入、在价格高的平台卖出实现收益。例如同一事件在 Polymarket 和 Kalshi 上价格不同，或者利用外部衍生品市场对价格进行校准⁵。
4. **市场制造与再平衡套利 (Market-Making & Rebalancing)**：通过提供流动性或重调仓位赚取买卖差价和平台奖励。这种策略兼具套利和收益稳定的特点⁶。

套利不仅限于买低卖高，套利者还会扮演流动性提供者、信息发现者和市场纠偏者的角色。以下各章节将逐一介绍每种策略的原理、优化方法及实践建议。

第二部分：二元市场套利策略

二元市场的 Yes 和 No 两个代币加起来应该等于 1 美元，如果出现 总和 < 1 或 > 1 ，则存在无风险套利机会。这是 Polymarket 上最基础的套利形式，也是许多自动化套利机器人的核心逻辑。我们从初级入门到高级优化逐步展开。

2.1 基本 “Yes+No” 套利 (Binary Dutch Book)

在二元市场中，Yes 代币表示事件发生的概率，No 代币表示事件不发生的概率。理论上 $Yes + No = 1$ ，如果市场中挂单价格导致 $Yes + No$ 之和 **小于 1**，则可以同时买入 Yes 和 No 代币各 1 份，以低于 1 美元的成本换取结算时 1 美元的收益；反之如果 $Yes + No > 1$ ，则可以借助 USDC 拆分代币、卖出 Yes 和 No 各 1 份套现再合并为 1 美元获利⁷。这个策略又称 **Dutch Book Arbitrage**，其利润率 $= |Yes + No - 1|$ 。

示例：某市场 Yes 价格 0.45 美元，No 价格 0.50 美元，总和 0.95 美元。如果买入一份 Yes 和一份 No，总成本 0.95 美元，结算时无论哪个事件发生，都可兑付 1 美元，因此净赚 0.05 美元，收益率约 5%。

然而这种简单套利机会在 Polymarket 上存在的时间极短，通常只持续数秒或更短时间。原因在于**自动化高频套利机器人**遍布这一市场，会实时监听订单簿并在出现少量差价时立即执行交易。ChainCatcher 报道指出，多选市场的订单簿独立导致概率和可能小于 1，此类套利窗口可能只持续几秒，且竞争激烈⁸。因此，如果你打算手动执行这种策略，可能会发现机会已经被机器人捕获。

2.1.1 Python 实现示例

以下代码展示如何使用 Python 调用 Polymarket Gamma API 扫描所有二元市场的 Yes/No 价格并计算是否存在套利空间。示例简化了查询，仅演示概念，实际应用需要考虑网络延迟、API 速率限制和链上交易执行。

```
import requests
import time

def fetch_markets():
```

```

# 调用 Polymarket Gamma API 获取当前活跃市场列表
response = requests.get('https://gamma.polymarket.com/markets')
return response.json()['markets']

def check_binary_arbitrage(market):
    # 假设 market 包含 yesPrice 和 noPrice 字段
    yes_price = float(market['yesPrice'])
    no_price = float(market['noPrice'])
    total = yes_price + no_price
    if total < 1:
        return 'Long', 1 - total # 买入 YES 和 NO
    elif total > 1:
        return 'Short', total - 1 # 拆分 1 USDC 卖出 YES+NO
    else:
        return None, 0

def run_scanner():
    markets = fetch_markets()
    for market in markets:
        if market['outcomeType'] == 'binary':
            status, profit = check_binary_arbitrage(market)
            if status:
                print(f"发现套利机会：市场 {market['title']}, 类型 {status}, 利润率 {profit:.2%}")
# 实际应考虑循环调用并设置延迟

```

实际运行时需要不断循环检查，并在确认机会出现时立即构造链上交易。此外，若 Yes+No 总和 > 1 时，需要调用 Polymarket 的“拆分”功能将 1 USDC 拆分为 Yes 和 No 代币并在订单簿中卖出；反之当总和 < 1 时则需要合并或等待结算。

2.1.2 高级优化：速度与交易执行

由于简单价格偏差持续时间很短，执行成功率取决于数据获取速度和交易提交速度。以下建议可用于优化策略：

1. **使用 WebSocket 订阅实时订单簿：**Gamma API 提供 WebSocket 通道推送订单簿变化，相比 HTTP 轮询延迟更低。可以利用 Python 的异步库 `websockets` 或 `aiohttp` 订阅实时数据。
2. **靠近 Polygon 节点部署：**部分专业套利者将服务器部署在靠近 Polygon RPC 节点或使用专用高速节点，以减少交易提交与确认的延迟⁹。
3. **签名与交易逻辑预先准备：**对于需要拆分或合并的交易，可预先准备好签名模板，监听到机会时只更新价格字段并立即提交。
4. **合并多个市场扫描：**可同时监控上百个市场，但要合理配置并发掘优先级，例如集中监控交易量大的市场或接近结算时间的市场。
5. **考虑手续费与滑点：**虽然 Polymarket 手续费较低，但交易中可能存在滑点，需要在利润率计算中扣除预期成本。通常只有当利润率超过几乎忽略不计的费用时才值得执行。

通过以上优化，可以显著提高捕捉单市场二元套利的成功率。但由于竞争激烈，这种策略适合作为入门学习，不宜依赖其长期收益。

2.2 尾盘扫货 (Endgame Sweep) 策略

当事件结果基本确定但市场尚未正式结算时，Yes 代币价格会接近 1 美元但不到 1。这时买入 Yes 并等待结算，可以在几乎无风险的情况下赚取 0.1%–1% 的收益，这被称为 **尾盘扫货** 或 **扫尾单**。链上 社区文章指出，在 Polymarket 上约 90% 的 1 万美元以上大额订单 都发生在价格高于 0.95 时¹⁰；套利者通过在结算前“扫尾”锁定最后一点利润。

核心原理：市场参与者希望尽快释放资本或锁定收益，因此在结算 前会将 Yes 代币以略低于 1 的价格售出；而套利者则愿意用资金换取时间等待结算，赚取这笔时间价值差¹¹。

风险因素：

1. **黑天鹅事件：**极小概率的反转事件或官方信息变化可能导致看似确定的结果推翻，从而损失全部投入¹²。例如体育赛事在结算前被裁判改判等。
2. **鲸鱼操纵：**大户可能通过砸盘制造恐慌，让零售投资者抛售 0.99 的代币，再低价买入实现套利。链上文章指出鲸鱼会利用评论区散播逆转谣言并借机扫货¹³。

风险控制与优化：

1. **控制仓位：**不在单一市场投入过多资金，建议将资本分散到多个即将结算的市场¹⁴。
2. **选择即将结算的市场：**选择距离结算时间较近且价格高于 0.997 的市场，缩短黑天鹅风险暴露期¹⁵。
3. **关注官方信息源：**确认证据后再介入，例如关注官方通告、赛事裁判公告等，避免情绪化评论误导。
4. **检测鲸鱼行为：**监控大额卖单和评论区活动，避免在操纵阶段追高。高胜率套利者会根据历史经验识别鲸鱼模式并谨慎跟随。

尾盘扫货虽然单次收益不高，但在对冲风险和复利运用下可实现稳定回报，适合作为长线资金管理中的辅助策略。

2.3 多选市场组合套利 (Sum < 100% / > 100%)

在多选市场中，只有一个选项会获胜。如果所有选项价格之和小于 1 美元，那么同时买入所有选项在结算时必定获得 1 美元，锁定利润；如果总和大于 1 美元，则可以拆分等值代币卖出所有选项并收取差价。链上报道指出，有人利用该策略将 1 万美元资本在 6 个月内翻了十倍，参与上万场市场¹⁶。实例中以美联储会议四个选项为例，其价格之和只有 0.995 美元，买入所有选项即可确保以 0.5% 的收益结算¹⁷。

原理分析：多选市场的每个选项使用独立的订单簿，交易一个选项不会立即影响其他选项的价格¹⁸。当零售用户集中交易某个选项时，其他选项价格暂时不变，这会导致概率和偏离 1，为套利者提供机会。由于套利者的介入时间极短（仅几秒），对技术要求很高⁸。

Python 实现思路：

1. 获取所有多选市场列表及每个选项的价格，计算价格总和。
2. 如果总和 < 1，则构建买入各选项的交易；如果总和 > 1，则拆分 USDC 并卖出所有选项。为了效率，可设置阈值（如差值 > 0.003）以覆盖手续费。
3. 监控每个选项的订单簿深度，计算能否买入/卖出足够数量不造成滑点。
4. 交易时需要快速签名并发送链上交易，所有动作应在一个事务内完成以避免被夹击。

优化技巧

- **并行监控：**同时对上千个多选市场执行计算，可使用协程或多线程提高效率。

- **预测市场波动**：通过分析历史波动，挑选更容易出现概率总和 偏差的市场，例如极受关注或流动性差的市场。
- **限价挂单**：在总和 > 1 情况下，可以提前挂单卖出某些选项，等待被市场吃掉，利用被动填单赚取手续费和价差。

需要注意的是，大部分无风险组合套利已被专业机器人垄断¹⁹。若自建机器人难以与专业套利者竞争，可考虑参与这些机器人背后的流动性池或使用更多创新策略（如多市场相关性套利）。

2.4 跨时间 & 条件套利

Polymarket 上有很多嵌套型时间事件，例如“BTC 在 6 月前是否触及 50,000 美元”与“BTC 在 12 月前是否触及 50,000 美元”。前者是后者的充分条件：如果 6 月前触及，则 12 月前必然触及。因此，长期事件的 Yes 价格理应不低于短期事件的价格。如果发现短期事件价格高于长期事件，则构建组合可以实现**确定性套利**。

示例：6 月合约价格 0.60 美元，12 月合约价格 0.50 美元。套利者可以卖出 6 月 Yes 并买入 12 月 Yes：1. **若 6 月前触及**：6 月合约需要支付 1 美元，但 12 月合约也触发支付 1 美元，两腿对冲，多余收益为 0.50 美元（买入成本）。2. **若 6 月前未触及但 12 月前触及**：6 月合约不需要支付，12 月合约支付 1 美元，净收益 = $1 - 0.50 = 0.50$ 美元，加上卖出 6 月合约获得的 0.60 美元，收益更高。3. **若全年都不触及**：两合约均不支付，卖出 6 月合约赚 0.60 美元，买入 12 月合约损失 0.50 美元，净赚 0.10 美元。

总之，当短期概率与长期概率违反递增关系时，简单的对冲头寸就能够锁定无风险收益。类似的逻辑可应用于概率逻辑关系中，例如“民主党赢总统”必然意味着“某州得票率高于某数”，若市场价格存在矛盾，则可以在两市场之间构建套利组合²⁰。

这类策略常被称作**概率单调性套利**或**条件概率套利**，需要对事件之间的逻辑关系有深刻理解，并实时监控跨市场价格。

2.5 市场再平衡套利与市场制造

市场再平衡套利关注市场价格被短期冲击后偏离均衡的情况。例如某位鲸鱼突然买入大量 Yes 导致价格从 0.5 飙升至 0.8，后续可能回落至合理水平。套利者可以在价格过冲时建立反向仓位（卖出 Yes 或买入 No），等待回调后平仓获利。论文中将此分类为市场再平衡套利³。这是统计套利的一种，需要判断市场超调的程度和回归速度。

与再平衡相似的还有**市场制造策略**。链上分析指出，这些套利机器人实质上充当市场做市商，为市场提供流动性并通过不断调整 Yes/No 持仓赚取差价²¹。在 Polymarket 中，用户可以将 USDC 存入市场池，系统自动将其拆分成等价的 Yes 和 No 份额，并将其添加到订单簿中作为买卖对手。随着市场价格变化，Yes/No 份额比会偏离 50:50；聪明的市场制造者会主动买入或卖出调整仓位，从而维持组合平衡并锁定利润⁶。

市场再平衡套利与市场制造要求：

1. 具备对订单簿和成交流的实时监控能力，识别鲸鱼冲击或短期情绪波动。可结合智能预警系统识别异常交易量。
2. 拥有快速下单能力以捕捉价格回归的时间窗口。需要高性能服务器、低延迟链上接口。
3. 需要注意**行情反转风险**：如果事件的基本面信息变化导致价格跳升并维持，不应盲目做空，需要甄别价格暴涨背后是否有新信息。通过外部新闻或社交媒体辅助判断。

通过市场制造或再平衡方式套利，虽然单次收益较小，但可将策略扩展至多个市场，累积稳定盈利，且不会与专业机器人硬拼“单次机会”，适合具备一定技术和资金实力的用户。

第三部分：跨平台套利与高级策略

随着预测市场生态丰富，单纯在 Polymarket 内部的无风险套利机会 逐渐减少，套利者开始将目光放到不同平台之间的价格差异，例如 Kalshi、Deribit、Betfair 等合法或灰色预测市场以及传统衍生品交易所。跨平台套利的核心逻辑是利用同一事件或同类事件在不同市场的定价差异，通过买低卖高锁定收益。该部分也包括触碰期权型 套利、组合套利等高级策略。

3.1 跨平台差价套利

跨平台套利的思路是在 Polymarket 等预测市场与传统预测平台或衍生品市场（如 Kalshi、Deribit、Betfair）之间比较价格。Alinvest 的报道总结了 Polymarket 上套利的三种形式：1. **单市场套利**：利用 Yes/No 加总不等于 1 的市场价格⁵。2. **跨平台套利**：当 Polymarket 与竞争对手（Kalshi 等）对同一事件的价格存在显著差异时，买入低价平台的 Yes 或对应合约，并在高价平台上卖出或做空²²。3. **自动化高频套利**：利用机器人扫描政治事件等高波动市场的不合理定价并快速下单，这些机器人在 2024–2025 年间通过执行 10,200 次交易赚取了 420 万美元²³。

由于 Kalshi 等平台受到美国监管限制，某些事件在 Kalshi 上许可但在 Polymarket 上不可用或定价不同；反之亦然。例如，某些货币政策或就业数据预测在 Kalshi 上有法定合约，而 Polymarket 上则需通过转化逻辑构建相似事件。这种差异导致同类事件的合约供需不一致，从而形成价格差。这些差价通常存在于事件早期或交易量较低时，一旦被市场发现会迅速收敛。

实践建议：

1. **对比同名事件**：列出 Polymarket 与 Kalshi 或 Betfair 上标的相同或高度相关的事件，实时抓取其成交价格 and 盘口深度，比较其概率差异。例如比较美国总统选举、加息决议、体育赛事等重大事件。
2. **考虑结算机制差异**：不同平台的结算标准和时间可能不同，例如 Kalshi 由 CFTC 监管，有明确时间和新闻来源；Polymarket 采用 UMA 预言机，用户可用 UMA 代币发起争议²⁴。因此跨平台套利需确认结算逻辑一致，才能建立对冲。
3. **计算交易成本**：跨平台套利涉及在两个或多个平台分别下单，需要支付平台费、链上手续费、汇兑手续费等，要确保利润超过成本。
4. **监控汇率波动**：如果使用稳定币不同（如 USDC 与 USD），需考虑汇率及入金/出金时间差导致的资金机会成本。
5. **融合外部数据**：结合衍生品市场的期权价格或期货价格来判断某事件的公允概率。例如使用 Deribit 的隐含波动率推断 BTC 触碰某阈值的概率，再与 Polymarket 相应市场比较偏差。

3.2 时间型触碰合约定价套利（Touch-Style Options）

Polymarket 上的时间触碰市场其实是一种 **数字触碰期权**（one-touch digital option）——在未来某一时间窗口内标的资产是否达到某个价格阈值。VertoxQuant 的文章将其与常见的触碰期权进行类比，指出此类合约可以通过外部期权市场估值。文章举例称，当 Polymarket 上存在“11 月 BTC 是否在任何时刻高于 130,000 美元”的市场时，这相当于支付 1 美元购买一张一触即付的期权²⁵。要计算这一合约的公平价值，可以利用 Deribit 等期权市场的隐含波动率曲线和现货价格，通过布朗运动或布朗桥模型模拟未来价格路径²⁶。

定价步骤：

1. **获取数据**：从 Deribit API 获取标的资产（如 BTC 或 ETH）的当前价格、期货价格以及隐含波动率曲面²⁶。同时获取无风险利率（可用 ETH 或 BTC 债券收益率估计）以及期货拆借利率。
2. **设置模型参数**：确定时间窗口、障碍价（Barrier）、步数和模拟路径数量。例如设置 100,000 条蒙特卡洛路径，每条路径 365 步表示一年，每步模拟一个交易日的价格变化。

3. **模拟价格路径**：采用几何布朗运动或带布朗桥的蒙特卡洛模拟生成价格路径，并在每条路径上判断是否在到期前达到障碍价。记录触及概率。
4. **计算理论价格**：触及概率乘以预期支付（1 美元）并贴现，得到理论价格，再与 Polymarket 上对应合约实际价格比较。如果市场价格明显低于理论价格，则可买入此合约，在 Deribit 上做空或购买相反期权对冲；如果市场价格明显高于理论价格，则相反操作。
5. **校验与阈值**：文章建议只有当偏差超过绝对门槛（如 3%）并在置信区间之外时才认定为显著偏差²⁷。这样可以减少由波动率估算误差或交易噪音带来的假信号。

实践建议：

- 使用 Python 库（如 `requests`、`quantlib`、`numpy`）实现期权定价模型。可借助 open-source 项目中的代码模板，快速构建测算脚本。
- 自动化监控：编写脚本定时从 Polymarket 获取触碰市场价格，从 Deribit 获取隐含波动率并计算理论值。当发现偏差超过阈值时，发送通知或自动下单。
- 风险控制：注意 Deribit 期权与 Polymarket 合约的结算时间、结算逻辑可能不完全一致，需要谨慎匹配。对冲交易不能完全覆盖所有风险，需设定止损或头寸控制策略。

此策略利用了外部衍生品市场的深度与定价透明度，可以帮助发现 Polymarket 市场定价不合理之处，但需要较高的金融工程知识和编程能力。

3.3 复杂组合套利与 LLM 智能关联

除了单纯的概率相加套利，一些研究者利用大型语言模型（LLM）探索多个市场之间的语义关联，以发现隐含的逻辑依赖并构建组合套利。IMDEA Networks Institute 的研究分析了 2024 年美国大选前后共 86 万笔 OrderFilled 事件，利用 LLM 嵌入识别出了 13 对逻辑上相关的市场组合，但只有 5 对组合产生了 9.5 万美元的套利利润²⁸。这表明虽然理论上组合套利可以在多个市场之间构建无风险组合，但实际可行性受到执行障碍、资金消耗和时滞的限制²⁸。

研究指出组合套利执行率低的原因包括：

- 多市场组合涉及更多合约和更高的交易成本，尤其是市场数量多时很难在所有市场同时建立或清算头寸。
- 事件之间的逻辑关系虽然存在，但因结算时间不同或市场深度差异导致套利空间无法在现实交易中兑现。
- LLM 方法可能发现语义关联但未充分考虑统计相关性，导致依赖关系不足以形成套利。

因此，虽然 LLM 助力的组合套利是一条研究路径，实际操作需要谨慎评估事件间关联度、实操成本及清算风险。

3.4 跨资产相关性套利

跨资产套利是指通过不同资产或指数的相关性来预测某一市场的走向。例如**加密货币与利率决策、股票指数与宏观事件**。套利者可观察传统金融市场对某一事件的反应，快速在 Polymarket 上布局。例如美国通胀数据公布前，债券或期货市场可能提前反映预期，而 Polymarket 上“通胀率是否超 X%”的市场调整滞后，利用这一时间差进行套利。这需要接入更多数据源，包括新闻、宏观经济指标和金融市场行情，以及建立快速反应系统。

第四部分：智能监控与高胜率钱包分析

在了解各种策略的基本原理后，我们需要建立**智能监控系统**，实时发现套利机会，并跟踪所谓的“聪明钱”（smart money）。聪明钱通常是指那些交易规模大、胜率高、信息获取效率高的玩家。以下内容介绍如何构建监控系统并识别高胜率钱包。

4.1 数据来源与监控工具

要构建智能监控系统，需要整合链上数据与链下数据：

1. **Polymarket Gamma API & Subgraph**：Gamma API 提供市场列表、订单簿、成交记录查询¹。Subgraph 提供更细致的持仓、盈亏、用户活动信息²¹。通过 GraphQL 可以按用户查询持仓与交易记录。
2. **链上事件日志**：使用 Polygon ETL 或 Web3.py 可以直接监听区块链上的交易事件，如 `OrderFilled` 和 `Transfer` 事件，实时捕获大额成交。
3. **外部衍生品行情**：Deribit API 提供期权价格和隐含波动率，帮助估算触碰合约公平价格²⁶。
4. **新闻与社交媒体**：关注事件相关的官方公告、新闻报道和 Twitter/Reddit 信息，辅助判断市场信息差。
5. **链上分析工具**：Dune Analytics 或 Nansen 可查询大额地址的历史盈亏，用于识别盈利高手。

4.2 鲸鱼与高胜率钱包识别

链上文章指出，Polymarket 上大多数用户并未实现持续盈利，而在获利者中“鲸鱼”即大额资金参与者占比很高²⁹。识别聪明钱需要从以下维度综合评估：

1. **交易规模**：统计每个地址的交易量和持仓规模，挑选出单笔成交额或累计持仓超过某阈值的地址。例如设置单笔交易>1 万美元或累计交易量>5 万美元作为鲸鱼识别标准。
2. **盈亏记录**：通过 Subgraph 查询地址在过往已结算市场的 PNL（Profit and Loss），计算其胜率和收益率。胜率=盈利市场数量/参与市场数量；收益率=累计盈利/投入资本。
3. **交易频率**：识别高频交易者。频繁进出多场市场且胜率高，可能说明其对信息敏感。
4. **内幕交易迹象**：观察某些地址是否在重大事件公布前短时间内大额下注且结果正确，可推测其掌握内幕信息¹³。
5. **交叉平台关联**：某些地址同时活跃在 Kalshi、Betfair 等平台，可借助链上跨链分析或第三方数据推断其身份关联。

通过计算上述指标，构建**综合评分模型**，高分地址可视为聪明钱名单。系统应重点监控这些地址的实时交易，当他们在某市场大额买入或卖出时视为可能存在信息优势或套利机会，并向用户发出提示。

4.3 监控系统设计

4.3.1 架构概述

监控系统可分为三个层次：

1. **数据采集层**：通过 API/WebSocket 拉取 Polymarket 市场数据、订单簿和用户交易事件；订阅 Deribit 等行情接口；抓取新闻 RSS；监听链上事件。
2. **数据处理与分析层**：包括套利检测算法（Yes+No 套利、多选组合套利、触碰定价套利）、异常交易监控、高胜率地址评分等模块。采用 Python 的 Pandas/Numpy 等进行数据处理，利用异步框架实现并发处理。可以将策略封装为策略类，便于扩展。
3. **通知与交互层**：检测到套利机会后通过 Webhook、邮件或 IM 推送通知，包含市场名称、当前价格、理论价格差、建议操作。也可以提供命令行输出或 Web 仪表盘显示当前市场的套利机会和聪明钱动态。建议设置通知节流，以避免频繁报警。

4.3.2 策略组合与优先级

系统可以同时运行多种套利策略，但受限于交易成本和资金规模，需要分配优先级。一般来说：

1. **高确定性策略优先**：无风险组合套利（Yes+No、Sum<1）因无风险而优先执行。若检测到这样的机会并满足手续费覆盖，立即构建或拆分头寸。
2. **触碰定价套利次之**：利用外部市场定价对比，盈利空间通常更大，但需在衍生品市场开仓对冲，操作复杂，资金占用较高。
3. **市场再平衡套利再次**：依赖价格回归，收益不确定但机会较多，可以用较少资本试水；如能与市场制造策略结合，则收益更稳定。
4. **跨时间/条件套利与组合套利**：需要深入分析事件逻辑和市场结构，实际收益比率较低，但作为策略多样化可纳入体系。

每种策略可设置不同的报警阈值和头寸限制，以避免资金集中到某一策略，导致风险过高。比如设置单个策略最大投入资金比例，或对特定策略单笔最大下单量。系统应及时汇总各策略的仓位和风险暴露，进行动态调整。

4.4 与外界信息同步

套利机会往往与信息传递速度紧密相关。系统除了监控链上数据，还需引入外部数据源：

- **宏观新闻和实时推送**：例如美联储决议、公司公告、选举结果公布等信息对预测市场有立即影响。可接入新闻 API 或 RSS 订阅，甚至直接爬取 Twitter 搜索关键字，以实时捕捉事件。
- **数据科学与 LLM 辅助**：利用自然语言处理模型分析新闻文档与社交媒体文章，提取情绪和事实，辅助判断是否存在新的信息导致价格合理上升或下降。这可以减少因盲目做空而被新信息打脸的风险。
- **高频数据同步**：对于需要秒级反应的策略，如多选组合套利和市场再平衡套利，要确保系统能够以极低延迟获取数据，可以使用异步 IO 或多线程技术，并与 Chainlink、Web3 节点及 Polymarket RPC 紧密相连。

第五部分：风险管理与合规性

套利并非百分之百的“免费午餐”。任何策略都有风险，尤其在去中心化预测市场这一相对新兴的领域，政策和技术风险同样不可忽视。本部分总结最常见的风险并提出应对措施。

5.1 流动性风险与滑点

Polymarket 部分市场交易量较小，订单簿深度有限。当套利者同时买入/卖出多份代币时可能造成价格滑点，使得实际成本高于理论价值，导致套利收益下降甚至转负。因此在计算套利利润时要考虑实际可成交的数量和价格，并尽量采取 **分批下单** 或 **限价挂单** 等方式减少滑点。对于多选组合套利，需检查每个选项的挂单数量是否足够支撑组合头寸。

5.2 执行风险与交易延迟

区块链上的交易需要等待确认，尤其在网络拥堵时会延迟几秒甚至几分钟。在此期间，价格可能发生变化，导致原本存在的套利空间消失。解决办法包括：

- 预签名交易，发现机会后立即发送；
- 使用优先级较高的 Gas 费用以加快确认速度；
- 在有条件的平台采用闪电贷款或闪电结算的方式实现原子交易（需具备智能合约开发能力）。

5.3 隐私与治理风险

Polymarket 采用 UMA 预言机决议市场结果，社区可以通过质押 UMA 代币发起争议并表决。然而这一机制存在被大户操纵的风险，Coin360 报道一名鲸鱼利用 500 万 UMA 代币（25% 投票权）操纵市场结果，从 “Too Early” 改为 “Yes”，引发广泛争议³⁰。结果证明在投票参与率低的情况下，大户可扭曲结果，伤害了普通交易者的权益³¹。此外，发起争议需要缴纳 750 USDC 保证金，这对一般用户构成门槛³²。

因此，在选择事件参与时要考虑预言机的可靠性，关注社区关于治理漏洞的讨论，避开可能被操纵的市场或降低持仓比例。同时应关注 Polymarket 平台对治理机制的改进，例如延长争议期、引入更多数据源和增加惩罚力度³³。

5.4 合规与法律风险

不同国家/地区对预测市场的监管态度差异较大。Kalshi 等平台受美国 CFTC 监管，需要遵守严格的合约范围和交易限额；而 Polymarket 当前面向全球用户，尚处于灰色地带。跨平台套利时必须了解各平台的合规要求、KYC/AML 政策以及税务影响；某些国家可能禁止参与预测市场或将其视为赌博，需要谨慎评估。建议咨询专业法律顾问并关注相关法规变化。

5.5 信息风险与黑天鹅

预测市场的价格反映参与者对未来事件的预期，但突发新闻、政治事件、市场黑天鹅等会导致短时间内价格剧烈波动甚至逆转。如果在尾盘扫货策略中投入过大资金，一旦出现黑天鹅，可能导致全额亏损¹²。应通过分散投资、止盈止损设置以及动态调整仓位来降低风险。实时关注官方公告和可信新闻可以降低误判风险。

第六部分：开源项目和工具链

Polymarket 社区有大量开源项目和工具，可帮助交易者快速搭建机器人或监控系统。下面列举一些常用项目，并结合其功能讨论如何利用它们构建自定义套利程序。

6.1 Polymarket-Kalshi BTC Arbitrage Bot

该项目旨在监控 Polymarket 上的 “Bitcoin Up or Down” 市场与 Kalshi 上的相应市场，通过计算两腿组合成本发现风险中性套利机会。它使用 Python 后端（FastAPI）和前端仪表盘。核心逻辑包括：

- 数据采集：抓取 Polymarket 最新的 Bitcoin Up/Down 市场及 Kalshi 上的对应市场³⁴。
- 价格标准化：将不同平台的价格转换为相同单位或概率，使其可以比较。
- 成本计算：计算组合构建两腿仓位的成本，例如 Polymarket “Up” + Kalshi “No” 与 Polymarket “Down” + Kalshi “Yes” 两种策略³⁵。
- 监控与报警：若组合成本 < 1 美元，则表示存在无风险套利，提示用户³⁶。

项目提供了完整的代码示例，适合参考如何处理跨平台数据、建立前端仪表盘、以及自动报警逻辑。使用者可以根据需要调整监控频率、增加更多市场组合或对接其他平台。

6.2 Polymarket Binary Arbitrage Bot

另一个开源项目专注于单市场二元套利和多市场组合套利。其文档强调：

- 通过 Gamma API 获取所有市场元数据、价格、未结算列表等³⁷。
- 当 Yes+No 总和 < 1 时，构建多头头寸并等待结算获利；当总和

1 时，拆分 USDC 卖出并合并。³⁸

- 支持多市场套利：在互斥的多个市场中，如果 Yes 价格之和 <1 ，则买入每个市场的 Yes；如果 >1 ，则卖空或者买 No 并拆分合并。³⁸
- 该程序会将市场和利润信息输出，用户可根据提示下单³⁷。

从该项目可以学习如何使用 Gamma API 查询所有市场并循环计算 总和，以及如何在发现机会时输出或记录。虽然实际竞争激烈，但 作为学习工具非常有价值。

6.3 find-related-markets-arbitrage

这个 Python 包用于找出 Polymarket 中相关事件的套利机会，文档中 详细解释了 **underpriced** 和 **overpriced** 两种情况：当多个市场 Yes 价格之和 <1 为 **underpriced** 情形； >1 为 **overpriced**。文档还 强调了真套利与资本效率的区别，注意在执行多市场套利时计算最 优资产配置³⁹。这个项目还提供了基于 NLP 的事件相似 度识别模型，可帮助筛选互相关联的市场。

6.4 其他资源

- **Polymarket Agents 框架**：官方提供的智能代理框架，封装了 Gamma API 调用、下单签名、策略执行等组件，可用来快速 开发自己的策略。
- **Polymarket API Python SDK**：封装了 REST API 和 GraphQL 查询，提供了同步和异步接口，方便从 Python 脚本中调用。
- **Deribit API Python 客户端**：可获取期权价格、波动率指数、深度等数据，支持 WebSocket 订阅实时行情，是触碰定价策略的数据源。
- **定价库与金融工具**：QuantLib、py_vollib 等可用于构建期权 定价模型，对触碰合约和组合期权进行复杂计算。

第七部分：策略优化路径与实战指南

对于初入预测市场套利的新手和希望进一步优化策略的专业者，本 部分提供分阶段实战指南，涵盖从小资金试水到构建完整监控系统 的建议。

7.1 阶段一：基础策略学习与手工实践

1. **了解平台规则**：仔细阅读 Polymarket 的结算规则、预言机机制和市场创建条款。了解 Yes/No 合约拆分与合并、市场类型区分、争议流程等。
2. **手动识别机会**：使用 Polymarket 官方网站或第三方分析仪表盘，观察哪些市场的 Yes+No 总和偏离 1，或多选市场的价格之和偏离 1。尝试少量资金进行尾盘扫货或组合套利，感受结算流程。
3. **跟踪聪明钱**：观察一些历史高胜率钱包的交易行为，理解他们偏好的市场类型和下注时机，验证其逻辑。

通过手工实践可以逐步了解预测市场的动态，积累风险意识。

7.2 阶段二：开发简单扫描脚本

1. **学习 Gamma API 与 Subgraph 使用**：使用 Python 调用 API 获取市场数据，学习 GraphQL 查询语法，获取订单簿、持仓等。
2. **编写二元套利扫描器**：实现代码定时拉取二元市场价格，判断 Yes+No 是否偏离 1，输出可能套利的市场。可设置阈值和 筛选规则，如排除流动性不足的市场。
3. **实现多选组合套利扫描**：扩展脚本获取多选市场所有选项价格，计算和是否小于 1 或大于 1，输出结果。
4. **日志和报表**：将扫描结果保存到文件或数据库，记录历史价格与机会，用于后续策略优化与回测。

该阶段目的是熟悉自动化数据采集与基本算法实现，为下一阶段打 基础。

7.3 阶段三：引入外部数据与高级策略

1. **Deribit 数据接入**：学习 Deribit 的 REST/WebSocket API，获取现货价格、期权隐含波动率。编写程序计算触碰合约理论价格，并与 Polymarket 市场价格对比，输出偏差。
2. **跨平台数据整合**：如果你有 Kalshi 或 Betfair 账户，可调用其 API 获取市场价格，与 Polymarket 对比。创建策略：当价格差超过阈值时执行跨平台套利。需要考虑资金跨平台的时效问题和法币兑换费。
3. **异常事件检测**：监控大额交易与市场异动；结合社交媒体和新闻，识别基本面变化。例如监测特定关键字在 Twitter 上突然爆发，用情绪分析模型判断是否会影响相关市场。
4. **引入统计模型**：对历史数据进行回测，评估各策略的收益分布、盈亏比、胜率等，使用统计学方法（如 t-test、均值恢复指数）挑选最有效策略。引入机器学习模型预测市场回归速度、识别异常点，为再平衡策略提供依据。

这一阶段会显著提升套利系统的复杂度，需要掌握数据科学和金融工程知识，但也是提高收益和竞争优势的重要手段。

7.4 阶段四：构建完整自动化系统

1. **系统架构设计**：基于前面介绍的数据采集、策略分析和告警模块，搭建异步或分布式系统。可使用 FastAPI/Flask 构建后端服务，通过 WebSocket 订阅实时数据；使用 Celery 或 APS Scheduler 定时执行策略；通过数据库（如 PostgreSQL）保存历史数据。
2. **交易自动化**：如果计划自动执行交易，需要集成 Polymarket 订单簿交易库（py-clob-client）以及钱包签名模块。构建交易引擎，支持根据策略计算仓位并发出链上交易。需要谨慎管理密钥，做好权限隔离。
3. **前端仪表盘**：使用 React/Next.js 等技术构建界面，显示实时市场价格、策略机会、历史盈亏、聪明钱动态等。方便自己或团队随时监控系统运行。也可以通过移动端推送实时消息。
4. **风险控制与日志**：引入仓位管理模块，根据策略风险等级限制最大投入；为每笔交易设置止盈止损；记录每次执行决策和结果，方便后续审计与改进。
5. **持续迭代与优化**：定期对策略回测，加入新的策略模块；使用测试环境仿真系统，确保部署前策略有效性；关注平台规则和市场变化，及时调整策略。

这一阶段需投入较多时间与资源，但可以打造竞争力强、反应迅速、灵活可扩展的套利系统，为长期稳定收益奠定基础。

7.5 阶段五：深度研究与创新

在系统稳定后，可以探索前沿研究与创新方向：

1. **利用大型语言模型（LLM）对市场描述和新闻进行语义分析**，自动识别相关市场并挖掘隐含关系；开发基于 LLM 的事件分类和情绪识别算法，以预测市场动向。
2. **组合优化**：构建包含多种策略的投资组合，使用现代投资组合理论（Mean-variance、Kelly Criterion）来动态分配资金，实现收益与风险的平衡。
3. **跨链与 Layer-2 扩展**：关注以太坊主网及其他预测市场平台的套利机会，例如 LayerZero 等跨链通信协议可能支持跨链套利；DeFi 组合代币或收益策略也可与预测市场结合。
4. **算法交易与机器学习**：开发强化学习或深度学习模型，用于预测短期价格变化并自动执行再平衡策略。这些模型可透过历史订单簿数据训练；同时需要防止过拟合并考虑市场环境变化。

第八部分：案例分析

为了更直观地理解套利策略的应用，本节通过几个真实案例和模拟实验展示如何构建头寸以及如何从中获利。

8.1 多选市场套利案例

背景：在美联储加息预期市场中，存在四个互斥选项：降息 50 基点、降息 25 基点、不变、加息 25 基点，交易价格分别为 0.001、0.008、0.985、0.001⁴⁰。价格总和 0.995，小于 1。

操作：买入每个选项 1 份，总成本 0.995 美元。待市场结算时 其中一个选项将支付 1 美元，其余归零。无论哪个结果发生，都会得到 1 美元，收益 0.005 美元（约 0.5%）。如果能迅速捕捉到 类似机会并多次执行，收益可累积⁴¹。

注意：由于机会短暂，需要自动化工具实时监控并下单，手动 操作成功率几乎为零。还需考虑交易费和滑点。

8.2 尾盘扫货案例

背景：某场体育比赛中，比分已经明朗但市场尚未结算，Yes 价格 0.997。

操作：买入 Yes 代币 10,000 美元成本，静待结算，收益可能约为 0.3%（相当于 30 美元）。

风险：若比赛出现黑天鹅导致结果反转，则这一头寸损失 10,000 美元全部本金；或鲸鱼故意砸盘造成暂时价格暴跌，引发恐慌 卖盘。为降低风险，可将资金拆分投放在多个接近结算的市场。

8.3 触碰期权套利案例

背景：Polymarket 市场：“ETH 在 2025 年 6 月前是否触及 3,000 美元”。市场价格 Yes=0.40，No=0.60。

定价：通过 Deribit 获取 ETH 当前价格 2,600 美元，隐含波 动率 40%，无风险利率 5%。使用布朗运动模拟 100,000 条价格路径， 计算在 6 个月内触及 3,000 美元的概率为 48%，对应公平价格 0.48。比较市场价格 0.40，低估了 8 个百分点。

操作：在 Polymarket 上买入 Yes，同时在 Deribit 上买入 6 月到期、行权价约 3,000 美元的看涨期权或执行合适的 Delta 对冲（买入现货 / 卖出期货）以覆盖价格暴涨风险。若到期前 ETH 触及 3,000 美元，Polymarket 合约支付 1 美元，Deribit 期权利润 抵消成本；若未触及，则 Polymarket 合约归零，但期权到期亦可 保持小幅损失。由于理论概率高于市场定价，该策略预期获利。

8.4 跨时间套利案例

背景：市场 A：“BTC 在 2025 年 3 月前是否触及 50,000 美元”，Yes=0.55；市场 B：“BTC 在 2025 年 12 月前是否触及 50,000 美元”，Yes=0.50。短期概率竟然高于长期概率。

操作：卖出市场 A Yes（0.55 收入），买入市场 B Yes（0.50 成本）。

结局：* 若 3 月前触及，则 12 月前必然触及；卖出 A 赔付 1 美元，买入 B 获得 1 美元，合计 0 盈利，但额外赚得差价 0.05 美元；* 若 3 月前未触及但 12 月前触及，则 B 获得 1 美元，A 不需要付 费，加上卖出 A 获取的 0.55 美元，总计 1.55 美元，减去买入 B 的 0.50 成本，利润 1.05；* 若全年未触及，则 B 与 A 均为 0，收益仅来自卖出 A 的 0.55 净额减去买入 B 的 0.50，利润 0.05。无论哪种情况，组合都为正收益。这是概率单调性套利的典型案例。

第九部分：前景展望与结语

Polymarket 的兴起标志着去中心化预测市场开始进入主流。随着平 台的估值达到 90 亿美元⁴² 并获得大型投资机构支持，流动性 和用户群体急剧增长。这意味着市场会逐渐高效，简单套利机会会 越来越少；同时新的市

场类型和更复杂的事件组合会不断出现，带来更具挑战性的套利机会。根据研究，2024–2025 年间通过套利从 Polymarket 抽取的利润约 4,000 万美元⁴³。然而鲸鱼操纵和暗箱投票事件也暴露出平台的治理漏洞³¹。

未来的预测市场套利可能呈现以下趋势：

1. **技术化程度提升**：自动化高频交易和机器学习将成为常态，手工套利难以生存。投资者需要掌握编程、数据分析和金融工程技能，才能构建高效的套利系统。
2. **跨平台与跨链**：随着预测市场平台多样化，跨平台套利将更常见，甚至可能跨链（例如不同链上的预测市场或 DeFi 协议）。对资金调度和合规性的要求更高。
3. **监管规范化**：监管机构开始关注预测市场的合法性与投机风险，未来可能出台更严格的规定，限制某些事件的预测并加强 AML/KYC。套利者需要遵守法规，避免违规。
4. **更复杂的衍生品融合**：随着期权、期货等金融产品引入 Web3，预测市场可能与 DeFi 结合产生新的结构化产品（如保本基金、杠杆合约）。套利策略也会随之升级，例如利用期权对冲预测市场风险。

总之，套利不仅需要数学和编程技巧，更需要对市场逻辑、心理与制度环境的深刻理解。通过本文的系统梳理，相信读者对 Polymarket 及其套利策略有了全面认知，可以根据自己的风险偏好和技术条件制定合理的交易计划。在实践过程中，请谨记风险控制和合规要求，将套利视为市场效率提升的手段而非单纯的暴利工具。祝您在预测市场的旅程中好运。

1 6 7 8 9 10 11 12 13 14 15 16 17 18 19 21 29 40 41 42 People quietly making a fortune through arbitrage on Polymarket | Bitget News

<https://www.bitget.com/news/detail/12560605016888>

2 3 4 Unravelling the Probabilistic Forest: Arbitrage in Prediction Markets

<https://arxiv.org/pdf/2508.03474.pdf>

5 20 22 23 43 The Alchemy of Arbitrage: Exploiting Inefficiencies in Polymarket's Prediction Markets

<https://www.ainvest.com/news/alchemy-arbitrage-exploiting-inefficiencies-polymarket-prediction-markets-2509/>

24 30 31 32 33 Whale Rigged \$7M Polymarket Bet With 5M UMA Tokens

<https://coin360.com/news/polymarket-oracle-vote-manipulation-scandal>

25 26 27 How to Price Touch-Style Options (Polymarket) - VertoxQuant

<https://www.vertoxquant.com/p/how-to-price-touch-style-options>

28 Combinatorial Arbitrage in Prediction Markets: Why 62% of LLM-Detected Dependencies Fail to Generate Profit | by Navnoor Bawa | Nov, 2025 | Medium

<https://medium.com/@navnoorbawa/combinatorial-arbitrage-in-prediction-markets-why-62-of-llm-detected-dependencies-fail-to-26f614804e8d>

34 35 36 polymarket-kalshi-btc-arbitrage-bot/README.md at main · CarlosIbCu/polymarket-kalshi-btc-arbitrage-bot · GitHub

<https://github.com/CarlosIbCu/polymarket-kalshi-btc-arbitrage-bot/blob/main/README.md>

37 38 raw.githubusercontent.com

<https://raw.githubusercontent.com/P-x-J/Polymarket-Arbitrage-Bot/main/README.md>

39 find-related-markets-arbitrage command - github.com/ivanzzeth/polymarket-go-gamma-client/examples/find-related-markets-arbitrage - Go Packages

<https://pkg.go.dev/github.com/ivanzzeth/polymarket-go-gamma-client/examples/find-related-markets-arbitrage>