

适用于 S&P 500 全体成分股的**量化趋势评分框架**，包括（1）所需技术指标、（2）各指标量化方法、（3）归一化与加权方案、（4）完整算法流程与伪代码示例。按照此框架即可计算出每只股票的 Trend Score，并据此生成强势趋势排行榜。

1 所用技术指标与设计逻辑

维度	指标	设计初衷	常用参数
价格均线结构	SMA 10/20/50/150 日多头排列分数	判断短中长期多头共振	10, 20, 50, 150
均线斜率	50 日 SMA 斜率（年化）	衡量中期趋势陡峭度	回归 20 个交易日
趋势强度	ADX	分离强趋势与震荡	14 日
动量	ROC（Rate of Change）	捕捉加速或放缓	20 日
反转压力	RSI	筛掉超买/超卖噪音	14 日
趋势延续	MACD Histogram 斜率	判断动量扩张	EMA(12, 26, 9)
相对强度	Relative Strength vs S&P 500	把个股与大盘对比	63 日（≈3 个月）
量价确认	OBV 斜率	用资金流量验证趋势	回归 20 日
高点突破	距 52 周新高百分比	趋势股往往接近新高	252 日高点
成交量增速	20 日平均量/120 日平均量	动能行情伴随放量	20 日、120 日

注：指标都可直接用 Yahoo Finance 等免费数据计算，无须专有订阅。

2 单指标评分方法

- 数值化**
对每个指标先计算原始值，例如 ADX=35，RSI=68，价格距离 52 周高点 = -4% 等。
- 区间映射**
将原始值映射到 0 - 100 分。常见办法是：
 - 线性映射：若 $ADX < 15$ 记 0 分， ≥ 40 记 100 分；中间线性插值。
 - 对数或双曲：对量价比等极端分布可取 $\log/\text{sigmoid}$ 。
- 方向一致化**
 - 越大代表越强（ADX、ROC、斜率、量比、相对强度）：直接用映射分。
 - 越小代表越强（距离新高、RSI 超买风险）：先取相反数或 $100 - \text{score}$ ，再映射。

4. 异常 Winsorize
限制 1–99 分，避免极端值主导。

3 综合权重与最终 Trend Score

一种常用、解释性强的权重示例（总分 = 100）：

类别	指标	权重
均线结构	SMA 多头排列分数	15
均线斜率	50 日 SMA 斜率	10
趋势强度	ADX	15
动量	ROC	10
反转压力	RSI	5
趋势延续	MACD Histogram 斜率	10
相对强度	RS vs S&P 500	10
量价确认	OBV 斜率	10
高点突破	距 52 周新高	10
成交量增速	20/120 日量比	5
合计		100

权重可根据策略风格微调：偏交易可提高动量、滑点、量价权重；偏长线可提高均线、相对强度权重。

4 算法流程

1. 拉取数据
2. `tickers = sp500_list()`
3. `price_df = download_prices(tickers, start, end, interval="1d")`
4. `volume_df = price_df["Volume"]`
5. 批量计算指标（可用 TA-Lib 或 pandas-ta）
6. `indicators = {}`
7. `for t in tickers:`
8. `p = price_df[t]`
9. `v = volume_df[t]`
10. `indicators[t] = {`
11. `"sma10": SMA(p, 10),`
12. `"sma20": SMA(p, 20),`
13. `...`

```

14.         "adx": ADX(p, 14),
15.         "roc": ROC(p, 20),
16.         ...
17.     }
18. 取最新数据点
19. latest = {t: {k: v.iloc[-1] for k, v in ind.items()} for t, ind in
    indicators.items()}
20. 单指标映射到 0-100
21. def map_score(name, value):
22.     if name == "adx":
23.         return np.clip((value-15)/(40-15)*100, 0, 100)
24.     elif name == "distance_high":
25.         return np.clip((10-abs(value))/10*100, 0, 100)
26.     ...
27. mapped = {t: {k: map_score(k, val) for k, val in d.items()} for t, d
    in latest.items()}
28. 加权求 Trend Score
29. weights = {...} # 上表
30. trend_score = {t: sum(mapped[t][k]*weights[k]/100 for k in weights)
    for t in tickers}
31. 排序输出
32. ranking = sorted(trend_score.items(), key=lambda x: x[1],
    reverse=True)
33. 可视化或落库
34. pd.DataFrame(ranking, columns=["Ticker",
    "TrendScore"]).to_csv("trend_ranking.csv", index=False)

```

5 伪代码—一步到位版本

```

INPUT  : list_of_sp500_tickers
OUTPUT : ranked_table(ticker, trend_score)

FOR each ticker IN list_of_sp500_tickers PARALLEL:
    prices, volumes ← get_price_volume(ticker)
    IND ← {
        sma10, sma20, sma50, sma150,
        sma50_slope, adx14, roc20,
        rs14, macd_hist_slope,
        rs_63d, obv_slope, pct_to_52w_high,
        vol_ratio_20_120
    }
    SCORE_raw ← map_to_0_100(IND)
    trend_score ←  $\sum$  (SCORE_raw_i × weight_i)
END FOR

RANK tickers BY trend_score DESC
RETURN DataFrame(ticker, trend_score, rank)

```

6 落地建议

1. 增量更新
 - 每日收盘后仅拉取当日数据，指标也只更新最新值，速度快。
2. 并行计算
 - Python + multiprocessing / Ray，或转移至 SQL + 窗口函数。
3. 监控与阈值
 - 设定 Trend Score ≥ 70 进入候选池；跌破 50 剔除。
4. 回测验证
 - 建议用 10 年历史回测，观察年化收益与回撤后，再上线实盘。
5. 可视化仪表板
 - 配合 Plotly Dash 或 Streamlit，实时展示排名、分项得分雷达图。

按照上述方案，即可快速生成一个透明、可解释且可扩展的 S&P 500 趋势排行榜，为选股、风控和市场监管提供量化依据。