# S&P 500 High-Growth Scoring —— 完整算法流程说明

（兼顾我们一路迭代的优化点）

---

## 1. 数据源与抓取

| 步骤 | 说明 | 关键优化 |
| --- | --- | --- |
| **1-1 获取标的** | 通过维基百科实时爬取 S&P 500 成份列表（Ticker、公司名、Sector、Industry）。 | 任何时间运行都能拿到最新成份；行业信息后面做分位数时要用。 |
| **1-2 按 Ticker 下载财报** | *yfinance* 同时抓取 • **季度**：quarterly_income_stmt / balance_sheet / cashflow • **年度**：income_stmt / balance_sheet / cashflow | 失败重试 RETRIES=3，每次 SLEEP_SEC=0.2 s 间隔，降低被 Yahoo 限速风险。季度缺表也先保留收入表，其他列在后面用空值填。 |
| **1-3 增量 or 全量** | update_mode = incremental / full 可选：• **增量**时，只保存比 DB 中最新 report_date 更新的财报；• **全量**时删除旧 DB 重新下载。 | 避免重复 I/O；全量时自动重建 Engine，防止旧连接指向已删除文件。 |
| **1-4 DB 落库** | 先创建一个空的 raw_financials 表（含**全部**列）——这样任何 DataFrame append 都不会触发 “Item wrong length” 异常。每抓到一批（季度/年度）就即时 to_sql(append)。 | 彻底解决了之前 SQLite schema 不完整导致的写入失败。 |

---

## 2. 原始字段标准化 (

**save_raw_to_db**

**)**

- **字段匹配**：用 first_available() 做 3 层模糊匹配
    1. **完全匹配**列名
    2. 转小写去符号后匹配

3. “包含/被包含” 模糊匹配

- **补衍生列**
  - total_debt = Long Term Debt + Short Long Term Debt
  - free_cash_flow = operating_cash_flow + capital_expenditures
  - 最新一天 price、forwardEps 也一并快照，后面估值要用。

---

**3. 指标计算 (**

**compute_metrics**

**)**

| 模块 | 计算逻辑 | 关键配置 |
| --- | --- | --- |
| **3-1 FY 视角** | 读最近 fy_years 份年报（默认 2 期）。对 **收入 / EPS / FCF** 和 **毛利率**：- **average**：逐年 YoY 取均值 (fy_calc=average)- **cagr**：起始→终点做 CAGR (fy_calc=cagr) | fy_years, fy_calc |
| **3-2 Q-Seq 视角** | 最近 4 个季度，计算连续 **3 个环比增速** 均值：g = (Q_t / Q_{t-1} – 1) → mean | 与财报发布频率无关，能捕捉最新加速或减速趋势。 |
| **3-3 盈利质量/效率/安全** | ROIC、ROE、OCF/Revenue、Asset Turnover、Net Debt/EBITDA、利息覆盖倍数、流动比率等 | 直接取最新一个季度/年报数。 |
| **3-4 估值** | PEG = (Price / Forward EPS) / Q-Seq EPS 增速 FCF Yield = FCF / 市值 | _ |

---

**4. 指标预处理**

1. **缺失值填充**：先用行业中位数填，再 Winsorize。
2. **Winsorize**：配置 winsor_min / winsor_max (默认 5 % / 95 %)。
3. **分位数打分**
   - percentile_scope = industry / all　→ 在行业内或全市场做 0-100 分位。
   - 当所属行业样本数 < min_industry_size（默认 5）时自动 fallback 到全市场分位，保证稳定。

---

**5. FY + Q-Seq 融合**

- 每个维度（收入 / EPS / FCF / 毛利率）先得到 **FY-score** 与 **Q-Seq-score**。
- 按权重线性融合：

```
combo_score = (fy_weight * FY + qseq_weight * QSeq) / (fy_weight +
qseq_weight)
```

- - 默认 fy = 0.40, qseq = 0.60，可在 [combo_weights] 自由调。
- 四个融合分再平均 → **growth_score**。

---

**6. 五大维度 & 总分**

| 维度 | 子指标 | 默认权重 ([weights]) |
|---|---|---|
| Growth | 上述 4 个融合分 | **0.45** |
| Quality | ROIC, ROE | 0.20 |
| Efficiency | OCF Ratio, Asset Turnover | 0.10 |
| Safety | Net Debt/EBITDA, Interest Coverage, Current Ratio | 0.15 |
| Valuation | PEG, FCF Yield | 0.10 |

total_score = Σ (weight_i × score_i) → 0-100。

---

**7. 星级阈值 (**

**[rating_thresholds]**

**)**

- ≥85：★★★★★
- 70-85：★★★★
- 55-70：★★★
- 40-55：★★
- ＜40：★

阈值也在 INI 可调。

---

**8. 输出**

- **数据库：**
  - raw_financials → 原始财务表
  - derived_metrics → 计算后指标
  - scores → 加权得分
- **Excel 报表：**high_growth_scoring_YYYYMMDD.xlsx
  - 已按 total_score 降序，前三列是 Ticker / Company / Total Score。

---

## 这套算法的核心优势

1. **双时空视角：**FY 捕捉长期趋势，Q-Seq 捕捉短期动量；权重灵活。
2. **行业分位+样本数回退：**既有可比性又避免小行业噪声。
3. **Winsorize + 中位数填补：**对极端值与缺失值都稳健。
4. **配置化：**所有权重、维度、分位范围、FY 年数、CAGR/平均等都可一键修改。
5. **增量刷新：**每天跑也只写入新财报，加快速度、节省 I/O。
6. **完整落库：**原始→指标→评分三表分层，便于审计与二次分析。

---

## 可探讨的改进点

| 方向 | 思路 |
| --- | --- |
| 行业对标更细化 | 用 GICS Sub-Industry 或自定义 peer group。 |
| 因子加权 | 通过历史回测，用 IR/IC 调整 weights 而非手工设定。 |
| 审计追溯 | 给 Excel 报表加 “钻取链接”跳到单家公司原始数据。 |
| 估值维度 | 加 EV/EBITDA、EV/FCF 等多估值因子，做分位后 PCA。 |

---

这样，金融背景的同事可以从 **数据链路 → 指标 → 加工 → 打分 → 输出** 全流程理解模型，进而评估其稳健性、可解释性以及对投资决策的实际支持度。