

SOP（每日运行流程）

目标：统一使用本地 MySQL（见 `config.ini`），每日增量更新 `stock_daily/indicators/levels/score_v3`，并将 LAOWANG/FHKQ 的每个交易日结果写入 MySQL（`model_*` 表）。bingwu 仍输出到 `outputs/`。

0. 一句话流程

收盘后（建议交易日 15:05）运行：

```
python everyday.py --config config.ini
```

1. 环境准备

- Python/Conda：示例 `conda activate p312`
- 依赖：`pip install -r requirements.txt`
- 需要联网：数据源来自 AkShare（含东财/同花顺等）

2. MySQL 配置

1. 启动本地 MySQL（建议 8.x）
2. 配置 `config.ini`（推荐写 `db_url`；也可用 `[mysql]` 段自动拼接）

连接优先级（高 → 低）：

1. CLI `--db-url`
2. 环境变量 `ASTOCK_DB_URL`
3. CLI `--db`（SQLite 路径）
4. `config.ini`
5. 默认 SQLite：`data/stock.db`（不推荐）

3. 初始化（只需一次）

推荐：

```
python init.py --config config.ini
```

等价（核心表）：

```
python astock_analyzer.py --config config.ini init-db
```

如需直接用 SQL 落库：

- 执行 `sql/schema_mysql.sql`
- 库表说明见 `sql/README.md`

4. 首次跑数据（可能较慢）

建议直接跑一次 `everyday.py`，它会自动拉取最近约 1500 天的日线（仅对“从未拉过的股票”生效），足够支撑指标/评分计算：

```
python everyday.py --config config.ini --workers 32 --models-workers 32
```

如你确实要全历史回补（非常慢）：

```
python astock_analyzer.py --config config.ini run --start-date 20000101 --end-date 20260119 --workers 32
```

5. 评分表（LAOWANG/FHKQ）初始化、增量、全量

这里的“评分表”指用于 UI/BI 查询的物化结果表（`model_*`）：

- `model_runs`：每个模型、每个交易日的运行状态（ok/error/row_count/message）
- `model_laowang_pool`：LAOWANG 每日 TopN 池
- `model_fhkq`：FHKQ 每日信号明细（可能为 0 行，但依然会记录 run）

5.1 初始化（只需一次）

通常 `python init.py --config config.ini` 已包含；也可单独执行：

```
python models_update.py --config config.ini init-tables
```

5.2 智能增量更新（推荐日常用）

根据 `stock_daily` 最新交易日 + `model_runs.last_ok_date` 自动补齐缺失日期：

```
python models_update.py --config config.ini --only both --workers 64 --laowang-top 200 --laowang-min-score 0 update
```

只更新单一模型：

```
python models_update.py --config config.ini --only laowang --workers 64 update  
python models_update.py --config config.ini --only fhkq --workers 64 update
```

5.3 全量重算（慢）

对 `stock_daily` 中所有交易日重算并覆盖写入 `model_*` :

```
python models_update.py --config config.ini --only both --workers 128 --
laowang-top 200 --laowang-min-score 0 full
```

说明：`models_update.py` 允许 `full/update/init-tables` 写在参数任意位置（例如 `... full --workers 64` 也能跑）。

5.4 强制重算某一天（不改代码的办法）

如果某日结果明显异常（例如行数很少），通常是当天 `stock_daily/score_v3` 尚未完整。可删除该日 `run` 记录后再跑增量：

```
DELETE FROM model_runs WHERE model_name='laowang' AND trade_date='2026-01-13';
DELETE FROM model_laowang_pool WHERE trade_date='2026-01-13';

DELETE FROM model_runs WHERE model_name='fhkq' AND trade_date='2026-01-13';
DELETE FROM model_fhkq WHERE trade_date='2026-01-13';
```

然后：

```
python models_update.py --config config.ini --only both --workers 64 update
```

6. 每日增量（收盘后）

```
python everyday.py --config config.ini
```

它会做：

1. 校验 DB 必须是 MySQL
2. 初始化核心表 + 模型表（幂等）
3. 增量更新日线 + 指标/压力支撑 + `score_v3`（多线程）
4. 将 LAOWANG/FHKQ 写入 MySQL（多线程）
5. 导出 `outputs/bingwu_YYYYMMDD.csv`

7. 多线程与性能调参（重点）

7.1 K 线 + `score_v3`（pipeline）

- 入口：`everyday.py --workers N` 或 `astock_analyzer.py run --workers N`

- 并发粒度：按股票并发（每个线程处理一个股票：拉日线 → 算指标/算 levels → 写入 DB）
- 注意：线程不是越多越好；网络源/数据库都会成为瓶颈。一般 16~64 比较稳。

7.2 模型评分物化（LAOWANG/FHKQ）

- 入口：`everyday.py --models-workers N` 或 `models_update.py --workers N`
- 并发结构：外层按交易日并发（增量 cap=4；全量 cap=32），内层把 `model_workers` 传给模型 `compute`（内部可能再用线程）。
- FHKQ：先用一条 SQL 批量拉取候选股最近 N 日，再多线程计算（所以 `--workers` 能明显加速）。
- LAOWANG：优先使用 `stock_scores_v3`（快）；若 `score_v3` 覆盖率明显不足，会自动回退为从 `stock_daily` 现算（慢，但保证不会出现“只有几行”的假结果）。

7.3 推荐配置（经验值）

- 单机本地 MySQL：`--workers 32`、`--models-workers 32`
- MySQL 默认 `max_connections` 常见为 151：不建议把并发开到几百（会变慢甚至报错）。

8. 常见问题排查

8.1 `no such table: stock_daily`

含义：连到了一个“空库”（最常见是误连 SQLite 或 MySQL 库没建表）。

处理：

1. 检查你最终使用的 DB（优先级见第 2 节）
2. 跑 `python init.py --config config.ini`

8.2 模型结果行数很少 / 明显不对

优先排查“数据覆盖率”是否不足（常见于首次跑数没跑完/中断）：

```
SELECT COUNT(DISTINCT stock_code) FROM stock_daily WHERE date='2026-01-13';
SELECT COUNT(*) FROM stock_scores_v3 WHERE score_date='2026-01-13';
SELECT COUNT(*) FROM model_laowang_pool WHERE trade_date='2026-01-13';
```

经验判断：

- 近年的交易日，`stock_daily` 的股票数通常应是几千；如果只有几十/几百，说明日线库不完整。
- `stock_scores_v3` 应接近 `stock_daily`（覆盖率低会导致 LAOWANG 只产出少量行）。

处理建议：

1. 先跑 `python everyday.py --config config.ini --workers 32` 补齐日线与 `score_v3`
2. 再跑 `python models_update.py --config config.ini --only both --workers 64 update`

8.3 `--workers` 怎么改都一样慢

常见原因：

- 任务本身是 DB/网络瓶颈（不是 CPU）
- MySQL 连接数/磁盘 IO 成为瓶颈
- 你跑的是 “LAOWANG 快路径”（主要是一条 SQL + 排序），线程不会带来线性加速