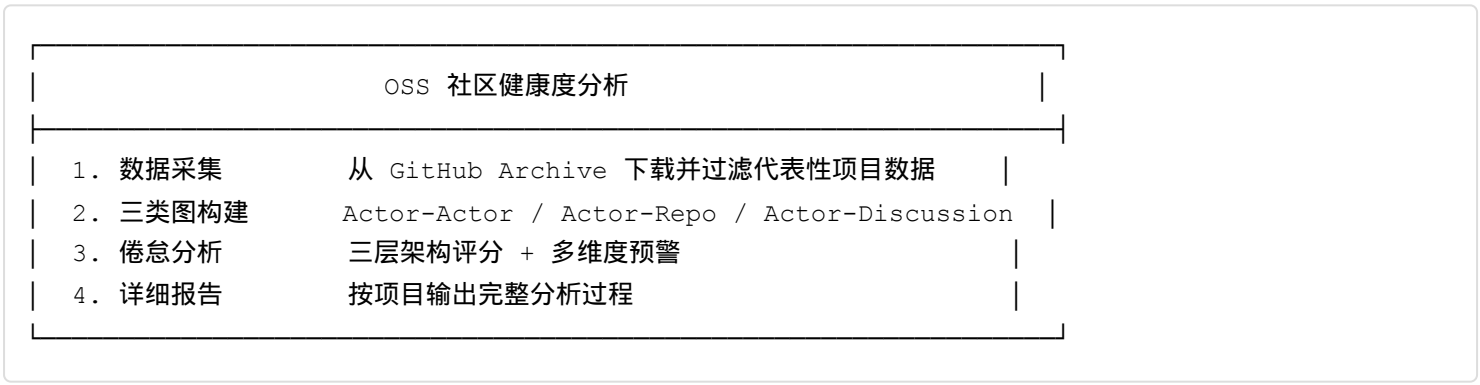
OSS 社区健康度分析工具

基于 GitHub Archive 事件数据，构建多类型时序图并进行维护者倦怠（Burnout）分析的工具集。

功能概览



快速开始

1. 安装依赖

```
python -m venv venv
source venv/bin/activate  # Linux/macOS
pip install -r requirements.txt
```

2. 数据采集（可选）

如果需要采集新数据：

```
# 下载 2023-2025 年数据（月采样模式，约 36GB）
python -m src.data_collection.gharchive_collector \
  --start-date 2023-01-01 \
  --end-date 2025-12-31 \
  --sample-mode monthly \
  --output-dir data/filtered
```

3. 构建月度图

```
python -m src.analysis.monthly_graph_builder \
  --data-dir data/filtered \
  --output-dir output/monthly-graphs \
  --workers 4
```

4. 运行倦怠分析

```
python -m src.analysis.burnout_analyzer \
  --graphs-dir output/monthly-graphs \
  --output-dir output/burnout-analysis
```

5. 查看详细报告

```
# 查看前 10 个高风险项目
python -m src.analysis.detailed_report --top 10

# 查看指定项目
python -m src.analysis.detailed_report --repo "kubernetes/kubernetes"
```

三类图构建

系统为每个项目的每个月构建三类图：

图类型	节点	边	用途
Actor-Actor	开发者	协作/评审/回复关系	倦怠分析（核心成员识别、协作网络）
Actor-Repo	开发者 + 仓库	贡献关系	贡献者分析、项目热度
Actor-Discussion	开发者 + Issue/PR	参与讨论关系	社区互动、新人融入

构建流程：

```
GitHub Archive 事件数据
    ↓
按月聚合（2023-01, 2023-02, ...）
    ↓
按项目分组
    ↓
为每个项目的每个月构建三类图
    ↓
导出为 GraphML 格式
```

Actor-Actor 图边的产生规则：

- `ISSUE_INTERACTION`：Actor A 在 Actor B 创建的 Issue 中发表评论
- `PR_REVIEW`：Actor A 在 Actor B 创建的 PR 中发表代码审查评论
- `PR_MERGE`：Actor A 合并了 Actor B 创建的 PR
- `ISSUE_CO_PARTICIPANT`：两个 Actor 都参与了同一个 Issue 的讨论

Actor-Repo 图边的产生规则：

- 所有 GitHub 事件（Push、PR、Issue、Comment、Review、Star、Fork 等）都会在 Actor 和 Repo 之间创建边

Actor-Discussion 图边的产生规则：

- Issue 相关：创建、关闭、评论 Issue
- PR 相关：创建、合并、关闭、审查 PR

倦怠分析算法详解

四个核心指标

对每个月的 Actor-Actor 图，提取以下四个指标：

1. **事件数（total_events）**：该月的 GitHub 事件总数，反映项目活跃度
2. **贡献者（unique_actors）**：该月参与项目的不同开发者数量，反映社区规模
3. **核心成员（core_actors）**：通过算法识别的核心维护者，反映项目维护能力
4. **协作质量（clustering_coefficient）**：平均聚类系数，反映协作网络的紧密程度（值域 [0, 1]）

核心成员识别

核心成员通过以下方法识别：

1. **计算加权度数**：根据边类型权重（PR_MERGE=3.0, PR_REVIEW=1.5, ISSUE_INTERACTION=0.5等）计算每个节点的加权贡献
2. **计算 k-core 值**：识别节点在网络结构中的核心位置
3. **综合得分**：`得分 = 0.5 × 归一化加权度数 + 0.5 × 归一化k-core值`
4. **动态筛选**：按得分排序，累计加权贡献达到 50% 时停止，确保核心成员覆盖主要贡献

三层评分架构

对每个指标的时间序列，使用三层分析计算得分（每个维度 0-25 分）：

1. 长期趋势（40%权重，满分 10 分）

- **计算方法**：使用线性回归拟合整个时间序列的斜率
- **公式**：$slope = \Sigma(x_i-\bar{x})(y_i-\bar{y}) / \Sigma(x_i-\bar{x})^2$
- **得分**：`得分 = -slope × 100`（斜率 < 0 时），最高 10 分
- **意义**：捕捉整体下降趋势

2. 近期状态（40%权重，满分 10 分）

- **计算方法**：对比最近 3 个月均值 vs 最早 3 个月均值
- **公式**：`变化率 =（最近均值 - 最早均值）/ 最早均值`
- **得分**：`得分 = -变化率 × 10`（变化率 < 0 时），最高 10 分
- **意义**：关注最近状态，捕捉突发下降

3. 稳定性（20%权重，满分 5 分）

- **计算方法**：计算月度变化率的标准差（波动率）
- **公式**： `波动率 = √(Σ(变化率ᵢ - 平均变化率)² / (n-1))`
- **得分**： `得分 = (波动率 - 0.3) × 25`（波动率 > 0.3 时），最高 5 分
- **意义**：惩罚高波动性，反映不稳定性

维度总分 = 长期趋势得分 + 近期状态得分 + 稳定性得分 = 0-25 分

综合倦怠评分

对四个维度分别应用三层分析，得到：

- **活跃度得分**（0-25分）：事件数趋势分析
- **贡献者得分**（0-25分）：贡献者数量趋势分析
- **核心成员得分**（0-25分）：核心成员留存率趋势分析
- **协作质量得分**（0-25分）：聚类系数趋势分析

综合倦怠评分 = 活跃度得分 + 贡献者得分 + 核心成员得分 + 协作质量得分 = **0-100 分**

风险等级：

- ≥60 分： 高倦怠风险
- 40-59 分： 中等风险
- 20-39 分： 低风险
- <20 分： 健康

预警信号检测

系统会逐月对比相邻两个月的数据，检测以下预警信号：

1. ACTIVITY_DROP（活跃度下降）

检测方法：

1. 对比相邻两个月的事件总数
2. 计算变化率： `变化率 = (本月事件数 - 上月事件数) / 上月事件数`
3. 触发条件：变化率 < -50%（下降超过 50%）
4. 严重程度：
    - `high`：变化率 < -70%（下降超过 70%）
    - `medium`：-70% ≤ 变化率 < -50%

示例：

- 上月：1000 事件
- 本月：400 事件
- 变化率： `(400-1000)/1000 = -60%`

- 触发： `medium` 级别预警

---

2. CORE_MEMBER_LOSS（核心成员流失）

检测方法：

1. 获取上月和本月的核心成员 ID 集合
2. 计算流失成员： `流失成员 = 上月核心成员 - 本月核心成员`
3. 计算流失率： `流失率 = 流失成员数 / 上月核心成员数`
4. 触发条件：
    - 流失率 ≥ 30%，或
    - 流失绝对数量 ≥ 2 人
5. 严重程度：
    - `high` ：流失率 ≥ 50% 或流失 ≥ 3 人
    - `medium` ：30% ≤ 流失率 < 50% 且流失 2 人

示例：

- 上月核心成员：10 人（ID: [1,2,3,4,5,6,7,8,9,10]）
- 本月核心成员：6 人（ID: [1,2,3,4,5,6]）
- 流失成员：4 人（ID: [7,8,9,10]）
- 流失率： `4/10 = 40%`
- 触发： `medium` 级别预警

---

3. COLLABORATION_DECLINE（协作质量下降）

检测方法：

1. 对比相邻两个月的聚类系数
2. 计算变化率： `变化率 =（本月聚类系数 - 上月聚类系数）/ 上月聚类系数`
3. 触发条件：变化率 < -30%（下降超过 30%）
4. 严重程度：固定为 `medium`

聚类系数说明：

- 聚类系数衡量节点的邻居之间相互连接的程度
- 值域 [0, 1]，值越高说明协作网络越紧密
- 聚类系数下降可能意味着协作质量下降、社区分裂

示例：

- 上月聚类系数：0.15
- 本月聚类系数：0.09
- 变化率： `(0.09-0.15)/0.15 = -40%`
- 触发： `medium` 级别预警

---

4. CONTRIBUTOR_DROP（贡献者下降）

检测方法：

1. 对比相邻两个月的活跃贡献者数量
2. 计算变化率： `变化率 = （本月贡献者数 - 上月贡献者数） / 上月贡献者数`
3. 触发条件：变化率 < -40%（下降超过 40%）
4. 严重程度：固定为 `medium`

示例：

- 上月贡献者：50 人
- 本月贡献者：25 人
- 变化率： `(25-50)/50 = -50%`
- 触发： `medium` 级别预警

---

5. SUSTAINED_DECLINE（持续下降）

检测方法：

1. 检查最近 3 个月的事件数序列
2. 判断是否连续下降： `events[0] > events[1] > events[2]`
3. 如果连续下降，计算累计下降率：
    - `累计下降率 = （最近1月 - 3个月前） / 3个月前`
4. 触发条件：累计下降率 < -30%（累计下降超过 30%）
5. 严重程度：固定为 `high`

示例：

- 3 个月前：1000 事件
- 2 个月前：800 事件（环比 -20%）
- 1 个月前：600 事件（环比 -25%）
- 本月：400 事件（环比 -33%）
- 累计下降率： `(400-1000)/1000 = -60%`
- 触发： `high` 级别预警

注意：此预警需要至少 3 个月的数据才能检测。

综合倦怠评分计算

计算流程

步骤 1：提取四个维度的时间序列

1. **活跃度序列**： `[month1.total_events, month2.total_events, ..., monthN.total_events]`
2. **贡献者序列**： `[month1.unique_actors, month2.unique_actors, ..., monthN.unique_actors]`

3. **核心成员流失率序列**：
   - 以首月核心成员为基准
   - 每月流失率 = `1 - （该月核心成员 ∩ 首月核心成员） / 首月核心成员数`
   - 序列：`[month1流失率, month2流失率, ..., monthN流失率]`
4. **协作质量序列**：`[month1.clustering_coefficient, month2.clustering_coefficient, ..., monthN.clustering_coefficient]`

步骤 2：对每个维度应用三层分析

对每个时间序列分别计算：

- 长期趋势得分（0-10分）
- 近期状态得分（0-10分）
- 稳定性得分（0-5分）

得到每个维度的总分（0-25分）。

步骤 3：综合评分

首先计算各维度的风险得分总和（越高越差）：

```
风险得分总和 = 活跃度风险得分 + 贡献者风险得分 + 核心成员风险得分 + 协作质量风险得分
             = (0-25) + (0-25) + (0-25) + (0-25)
             = 0-100分
```

然后转换为健康度得分（越高越好）：

```
健康度得分 = 100 - 风险得分总和
           = 0-100分（得分越高表示项目越健康）
```

评分示例

假设某项目经过 12 个月的分析，四个维度的风险得分如下：

维度	长期趋势	近期状态	稳定性	维度风险得分
活跃度	8.5分	7.2分	4.3分	**20.0分**
贡献者	7.8分	6.5分	3.4分	**17.7分**
核心成员	1.2分	2.1分	1.7分	**5.0分**
协作质量	4.5分	3.8分	2.0分	**10.3分**
风险得分总和				**53.0分**
健康度得分				**47.0分** (100 - 53.0)

解读：

- **活跃度风险（20.0分）**：事件总数明显下降，长期和近期都有下降趋势
- **贡献者风险（17.7分）**：活跃人数下降，但波动相对较小
- **核心成员风险（5.0分）**：核心成员相对稳定，流失率较低

- **协作质量风险（10.3分）**：聚类系数下降，协作网络紧密程度下降
- **总体评价**：健康度得分 47.0 分，中等风险，主要问题是活跃度和贡献者下降

风险等级划分（基于健康度得分）

健康度得分	等级	含义	建议
<40	high	高倦怠风险，需要关注	立即调查原因，考虑干预措施
40-59	medium	中等风险，有下降趋势	持续监控，准备应对方案
60-79	low	低风险，基本健康	正常监控即可
≥80	healthy	健康，无明显问题	保持现状

说明：健康度得分 = 100 - 风险得分总和，得分越高表示项目越健康。

项目结构

```
oss_graph_construction/
├── src/
│   ├── analysis/                    # 分析模块
│   │   ├── monthly_graph_builder.py # 月度图构建
│   │   ├── burnout_analyzer.py      # 倦怠分析
│   │   └── detailed_report.py       # 详细报告生成
│   ├── data_collection/             # 数据采集
│   │   ├── gharchive_collector.py   # GitHub Archive 下载器
│   │   └── representative_projects.py # 代表性项目列表
│   ├── models/                      # 数据模型
│   ├── services/                    # 核心服务
│   │   └── temporal_semantic_graph/ # 图构建服务
│   ├── cli/                         # 命令行接口
│   └── utils/                       # 工具函数
├── data/
│   └── filtered/                    # 过滤后的事件数据
├── output/
│   ├── monthly-graphs/              # 月度图文件
│   │   └── {owner}-{repo}/
│   │       ├── actor-actor/
│   │       ├── actor-repo/
│   │       └── actor-discussion/
│   └── burnout-analysis/            # 分析结果
│       ├── summary.json             # 评分排名
│       ├── all_alerts.json          # 预警列表
│       ├── full_analysis.json       # 完整分析数据
│       └── detailed_report.txt      # 可读报告
├── scripts/
│   ├── collect_data.sh              # 数据采集脚本
│   └── analyze_burnout.sh           # 分析运行脚本
└── requirements.txt
```

命令行参考

数据采集

```
python -m src.data_collection.gharchive_collector \
  --start-date 2023-01-01 \
  --end-date 2025-12-31 \
  --sample-mode monthly \      # daily/weekly/monthly
  --output-dir data/filtered \
  --resume                     # 断点续传
```

月度图构建

```
python -m src.analysis.monthly_graph_builder \
  --data-dir data/filtered \
  --output-dir output/monthly-graphs \
  --workers 4                  # 并行进程数
```

倦怠分析

```
python -m src.analysis.burnout_analyzer \
  --graphs-dir output/monthly-graphs \
  --output-dir output/burnout-analysis
```

详细报告

```
# 查看前 N 个高风险项目
python -m src.analysis.detailed_report --top 10

# 查看指定项目
python -m src.analysis.detailed_report --repo "kubernetes/kubernetes,facebook/react"

# 只看高风险项目（评分 ≥ 60）
python -m src.analysis.detailed_report --min-score 60

# 指定输出文件
python -m src.analysis.detailed_report --output my_report.txt
```

输出示例

详细报告片段

```
================================================================================
📊 项目：kubernetes/kubernetes
================================================================================

  综合倦怠评分：35.42 / 100
   风险等级：    低风险
   分析周期：2023-01 to 2025-12 (36 个月)

--------------------------------------------------------------------------------
📈 各因子详细分析（三层架构：长期趋势40% + 近期状态40% + 稳定性20%）
--------------------------------------------------------------------------------

【1. 活跃度】(0-25分)
   📊 数据概览：
      首月：1250.00 事件  →  末月：890.00 事件
   📉 长期趋势 (40%权重)：
      线性回归斜率：-2.15%/月
      ⚠ 每月平均下降 2.2%
      → 趋势得分：2.15
     近期状态 (40%权重)：
      早期3月均值：1180.33  →  近期3月均值：920.67
      变化率：-22.0%
      → 近期得分：2.20
   📊 稳定性 (20%权重)：
      月度波动率：18.5%
        波动可控 (≤30%)
      → 稳定性扣分：0.00
   ➡ 维度总分：4.35 / 25
```

技术栈

- **Python 3.8+**
- **NetworkX**: 图构建与算法
- **tqdm**: 进度显示
- **multiprocessing**: 并行处理

许可证