

关于学术开源项目生命力的数据分析报告

团队：艾雅队 队长：于雅 队员：邓艾迪

W1 赛道：可视化作品或数据报告

一、 数据来源与获取

(1) 数据来源：

论文平台 + Github

我们选择了学术论文端 **Journal of Open Source Software (JOSS)** 作为我们的数据来源，它是一个经过同行评审、专门发表开源软件论文的期刊。其优势在于每一篇被接受的论文都必须提供一个关联的、可公开访问的代码仓库链接（绝大多数为 **GitHub**），这为我们的“论文-仓库”关联分析提供了高质量、高准确性的基准数据集。

开源仓库端：**GitHub REST API v3**，通过该 API 获取 JOSS 论文关联仓库的详细静态属性与动态活动数据。

(2) 数据获取方式（分层爬取策略）：

论文数据 -> Github 仓库数据

第一步：获取论文列表及仓库链接

我们首先爬取了 JOSS 网站上从 2023 年 1 月 27 日至 2025 年 12 月 23 日期间发表的所有论文，提取其元数据，包括：总序号、标题、GitHub 链接、上传时间、发表时间、论文地址、论文标签和论文语言。在爬取技术

上，采用 `Selenium` 自动化框架模拟真实浏览器操作，确保能够完整获取动态加载的页面内容。爬取过程分为两层：外层循环负责翻页导航，内层循环负责逐篇提取论文信息。对于每篇论文，我们获取其标题、发表时间、论文地址、`GitHub` 仓库链接等信息。为了提高爬取效率和稳定性，我们实施了多项优化措施：启用无头模式减少资源消耗，设置显式等待确保元素加载完成，添加异常处理机制应对网络波动，并采用增量滚动方式完整加载页面内容。

具体的数据获取采用“论文-仓库”双向联动策略：以 `JOSS` 为权威论文源，通过其强制关联的 `GitHub` 仓库链接，定向获取代码库的完整生态数据。具体而言，在获取 `JOSS` 论文列表及对应 `GitHub` 链接后，我们借助 `GitHub` REST API v3，系统爬取了每个关联仓库的详细静态属性与动态活动数据，构建了可追溯、可量化的“学术发表-代码演进”关联数据集。

第二步：获取仓库详细数据

针对第一步中获取的每一个 `GitHub` 仓库链接，我们系统性地调用 `GitHub` API，爬取了两大类数据：静态属性：星标数、分支数、创建时间、主题标签；动态历史：总提交次数、贡献者列表、最近的更新记录、`Issue` 和 `Pull Request` 信息等，其初步结构如下：

```

1 {"总序号": 1, "github链接": "https://github.com/vortex-exoplanet/VIP", "owner": "vortex-exoplanet", "repo_name": "VIP", "stargazers_count": 77, 0, "forks_count": 17
2 {"总序号": 2, "github链接": "https://github.com/szhorvat/IGraphM", "owner": "szhorvat", "repo_name": "IGraphM", "stargazers_count": 99, 0, "forks_count": 17
3 {"总序号": 3, "github链接": "https://github.com/ComputationalPhysiology/simcardems", "owner": "ComputationalPhysiology", "repo_name": "simcardems", "stargazers_count": 100, 0, "forks_count": 17
4 {"总序号": 4, "github链接": "https://github.com/Pizzutti92/MG-MAMPOSSt", "owner": "Pizzutti92", "repo_name": "MG-MAMPOSSt", "stargazers_count": 3, 0, "forks_count": 1
5 {"总序号": 6, "github链接": "https://github.com/tkonolige/city2ba", "owner": "tkonolige", "repo_name": "city2ba", "stargazers_count": 20, 0, "forks_count": 1
6 {"总序号": 8, "github链接": "https://github.com/MartianColonist/POSEIDON", "owner": "MartianColonist", "repo_name": "POSEIDON", "stargazers_count": 37, 0, "forks_count": 1
7 {"总序号": 9, "github链接": "https://github.com/Jackson0J/PyFPT", "owner": "Jackson0J", "repo_name": "PyFPT", "stargazers_count": 11, 0, "forks_count": 1, 0, "forks_count": 1
8 {"总序号": 10, "github链接": "https://github.com/ianfhunter/GNOLL", "owner": "ianfhunter", "repo_name": "GNOLL", "stargazers_count": 48, 0, "forks_count": 2
9 {"总序号": 11, "github链接": "https://github.com/ICESat2-SlideRule/paper", "owner": "ICESat2-SlideRule", "repo_name": "paper", "stargazers_count": 0, 0, "forks_count": 0
10 {"总序号": 12, "github链接": "https://github.com/N-Wouda/ALNS", "owner": "N-wouda", "repo_name": "ALNS", "stargazers_count": 590, 0, "forks_count": 142, 0
11 {"总序号": 13, "github链接": "https://github.com/mgmb/pyngham", "owner": "mgmb", "repo_name": "pyngham", "stargazers_count": 3, 0, "forks_count": 0, "open_issues_count": 1
12 {"总序号": 14, "github链接": "https://github.com/sjbeckett/localcovid19now", "owner": "sjbeckett", "repo_name": "localcovid19now", "stargazers_count": 5, 0, "forks_count": 1
13 {"总序号": 15, "github链接": "https://github.com/lorenzo-rovigatti/oxDNA", "owner": "lorenzo-rovigatti", "repo_name": "oxDNA", "stargazers_count": 63, 0, "forks_count": 1
14 {"总序号": 17, "github链接": "https://github.com/BiomedSciAI/fuse-med-m1", "owner": "BiomedSciAI", "repo_name": "fuse-med-m1", "stargazers_count": 152, 0, "forks_count": 1
15 {"总序号": 18, "github链接": "https://github.com/ropensci/spiro", "owner": "ropensci", "repo_name": "spiro", "stargazers_count": 15, 0, "forks_count": 0, "forks_count": 1
16 {"总序号": 19, "github链接": "https://github.com/Jammy2211/PyAutoGalaxy", "owner": "Jammy2211", "repo_name": "PyAutoGalaxy", "stargazers_count": 31, 0, "forks_count": 1
17 {"总序号": 20, "github链接": "https://github.com/martinlackner/abcvoting", "owner": "martinlackner", "repo_name": "abcvoting", "stargazers_count": 42, 0, "forks_count": 1
18 {"总序号": 21, "github链接": "https://github.com/annajenul/UBayFS", "owner": "annajenul", "repo_name": "UBayFS", "stargazers_count": 5, 0, "forks_count": 1
19 {"总序号": 22, "github链接": "https://github.com/SuperKogito/space", "owner": "SuperKogito", "repo_name": "space", "stargazers_count": 479, 0, "forks_count": 1
20 {"总序号": 23, "github链接": "https://github.com/schochastics/signnet", "owner": "schochastics", "repo_name": "signnet", "stargazers_count": 23, 0, "forks_count": 1
21 {"总序号": 24, "github链接": "https://github.com/rhdtownsend/msg", "owner": "rhdtownsend", "repo_name": "msg", "stargazers_count": 12, 0, "forks_count": 3, 0
22 {"总序号": 25, "github链接": "https://github.com/nickelnine37/pykronencker", "owner": "nickelnine37", "repo_name": "pykronencker", "stargazers_count": 17, 0, "forks_count": 1
23 {"总序号": 26, "github链接": "https://github.com/kungfugo/BellDiagonalQudits.jl", "owner": "kungfugo", "repo_name": "BellDiagonalQudits.jl", "stargazers_count": 1, 0, "forks_count": 1
24 {"总序号": 27, "github链接": "https://github.com/Rocsig/FijiRelax", "owner": "Rocsig", "repo_name": "FijiRelax", "stargazers_count": 8, 0, "forks_count": 4, 0
25 {"总序号": 29, "github链接": "https://github.com/mpack/mpack", "owner": "mpack", "repo_name": "mpack", "stargazers_count": 5571, 0, "forks_count": 1689
26 {"总序号": 30, "github链接": "https://github.com/tud-hri/joan", "owner": "tud-hri", "repo_name": "joan", "stargazers_count": 17, 0, "forks_count": 10, 0, "open_issues_count": 1
27 {"总序号": 31, "github链接": "https://github.com/hydrosolutions/riversCentralAsia/", "owner": "hydrosolutions", "repo_name": "riversCentralAsia", "stargazers_count": 1, 0, "forks_count": 1
28 {"总序号": 32, "github链接": "https://github.com/parmoos/parmo", "owner": "parmoos", "repo_name": "parmo", "stargazers_count": 87, 0, "forks_count": 11, 0, "forks_count": 1
29 {"总序号": 33, "github链接": "https://github.com/bjmorgan/py-sc-Fermi", "owner": "bjmorgan", "repo_name": "py-sc-Fermi", "stargazers_count": 29, 0, "forks_count": 1
30 {"总序号": 34, "github链接": "https://github.com/inrae/RCalModel/", "owner": "inrae", "repo_name": "RCalModel", "stargazers_count": 6, 0, "forks_count": 2, 0
31 {"总序号": 35, "github链接": "https://github.com/ttarkowski/quila", "owner": "ttarkowski", "repo_name": "quila", "stargazers_count": 15, 0, "forks_count": 4, 0
32 {"总序号": 36, "github链接": "https://github.com/CLOVER-energy/CLOVER", "owner": "CLOVER-energy", "repo_name": "CLOVER", "stargazers_count": 17, 0, "forks_count": 1
33 {"总序号": 37, "github链接": "https://github.com/olivecha/guitarsounds", "owner": "olivecha", "repo_name": "guitarsounds", "stargazers_count": 8, 0, "forks_count": 1
34 {"总序号": 38, "github链接": "https://github.com/ThomasMBury/ewstools", "owner": "ThomasMBury", "repo_name": "ewstools", "stargazers_count": 94, 0, "forks_count": 1
35 {"总序号": 39, "github链接": "https://github.com/mad-lab-fau/tppc/", "owner": "mad-lab-fau", "repo_name": "tppc", "stargazers_count": 18, 0, "forks_count": 1

```

二、 初步数据处理

(1) 初步评分计算:

经整理之后，共获得 1358 个论文的相关数据，我们设计评分模型并且进行如下处理：

首先，由于我们想评估论文的生命力，而可用数据包括贡献者总数、总提交数、核心贡献者及其提交占比、近期提交数、最后更新时间、Star 数量等。

那么我们就从“开发深度”和“持续状态”两个核心维度量化开源项目的生命力、采用加权计分制，总分为 100 分，分别由贡献度（50 分）和活跃度（50 分）两个维度构成，每个维度下又细分为多个具体指标。

评估维度	具体指标	满分	评分逻辑	设计理念	示例
贡献度 (50分)	提交历史	15分	实际提交数 / 10000 × 15	衡量项目开发深度和积累	5000次提交 → 7.5分
	社区规模	20分	贡献者数 / 200 × 20	评估社区广度和参与度	50名贡献者 → 5分
	贡献分布	15分	前10人提交占比: ≤20% = 15分, ≥80% = 0分, 中间线性	鼓励去中心化, 避免个人依赖	前10人占30% → 11.25分
活跃度 (50分)	近期活动	25分	近期提交数 / 50 × 25	反映当前开发活跃状态	40次提交 → 20分
	更新时效	15分	最后更新时间: 1天内=15分, 180天=0分, 中间线性	识别“僵尸仓库”, 鼓励持续维护	90天前更新 → 7.5分
	社区关注	10分	Star数 / 50000 × 10	衡量项目影响力和受欢迎度	10000个Star → 2分

总分 = 贡献度得分 + 活跃度得分 | 总分范围: 0-100分

1. 在贡献度维度, 具体通过三项指标加权计算:

提交历史 (**15 分**) : 直接量化项目的代码开发总量, 以实际提交数按比例折算。

社区规模 (**20 分**) : 用于评估项目吸引和容纳协作者的广度, 以贡献者数量为基准。

贡献分布 (**15 分**) : 重点关注社区结构的去中心化程度, 通过计算前**10**名贡献者的提交占比来评分, 以此识别并规避“巴士因子”风险, 鼓励形成健康的、非个人依赖型的协作网络。

2. 在活跃度维度, 同样包含三项指标:

近期活动 (**25 分**) : 反映项目最新的开发强度, 以特定时间段内的提交数为准。

更新时效 (**15 分**) : 根据最后更新时间计算其时效, 通过惩罚长时间未更新的项目, 激励持续的维护行为。

社区关注 (**10 分**) : 以 Star 数量作为项目流行度和社区影响力的代理指标, 将“表面活跃”(如偶尔的依赖更新)与“实质性持续开发”区

分开来。

按照这样的标准我们就把原始的数据简单转换成了一个已经打分之后的 `scored_data.jsonl` 文件。

(2) 转换成.csv 文件

由于.jsonl 后缀的文件并不适合用于可视化，团队的设想即将其转化为更好阅读和操作的.csv 文件，这个操作要写一个 python 代码、读取原来的.jsonl 文件并完成转化：

```
87     # 分析贡献者
88     def analyze_contributors(contrib_list):
89         """分析贡献者数据"""
90         if not contrib_list or not isinstance(contrib_list, list):
91             return 0, 0, 0, "无"
92
93         try:
94             # 计算贡献者数量
95             num_contributors = len(contrib_list)
96
97             # 计算总提交数
98             total_commits = sum(contrib[1] for contrib in contrib_list)
99
100            # 计算主要贡献者
101            if total_commits > 0 and contrib_list:
102                main_contributor = max(contrib_list, key=lambda x: x[1])
103                main_ratio = round(main_contributor[1] / total_commits * 100, 1) # 也修正这里
104                main_name = main_contributor[0]
105            else:
106                main_ratio = 0
107                main_name = "无"
108
109            # 活跃贡献者（提交>5）
110            active_contributors = len([c for c in contrib_list if c[1] > 5])
111
112            return num_contributors, active_contributors, main_ratio, main_name
113
```

(全代码约 200 行，上图展示部分核心代码)

这样我们就转换成了一个.csv 文件，每一列分别是：总序号、github 链接、owner、repo_name、stargazers_count、forks_count、open_issues_count、created_at、pushed_at、项目年龄月、最近活动间隔天、项目状态、贡献者总数、活跃贡献者数、核心贡献者、主要贡献者占比_%、recent_commits_count、total_commits_count、贡献度得分、活跃度得分、总分，大致如下图所示：

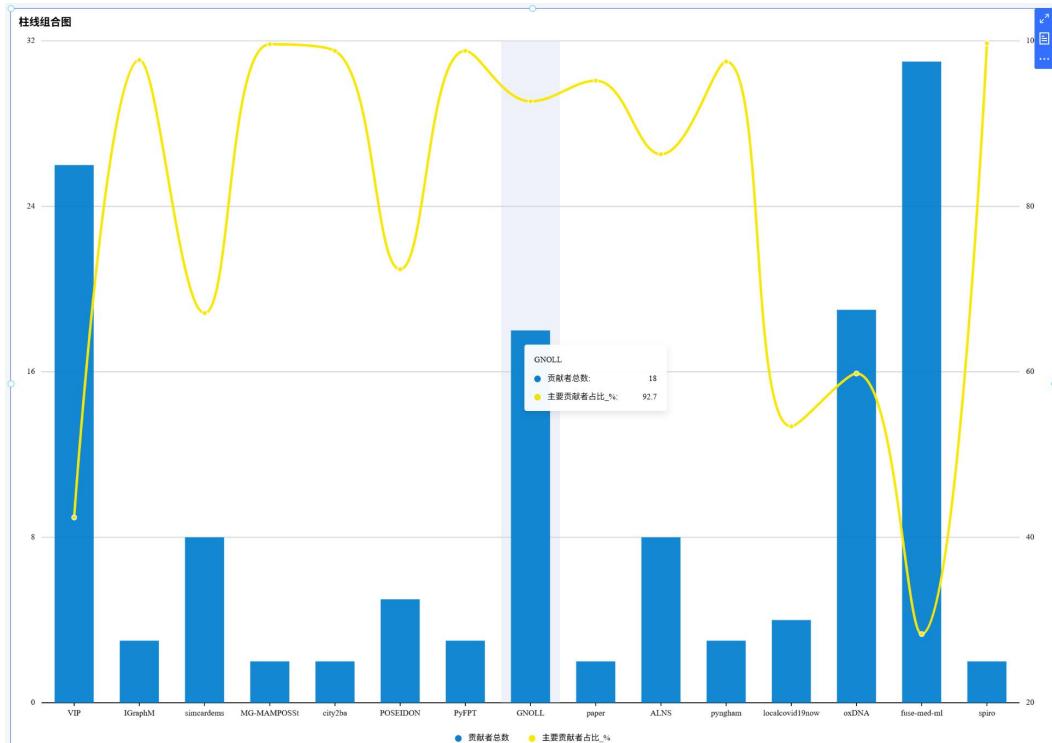
总序号	github链接	owner	repo_name	stargazer	forks	couopen	issue	created	apushed_at	项目年龄	最近活动	项目状态	贡献者数	总活跃贡献者数	核心贡献者数	recent_cot	total_com	贡献度得分	活跃度得分	总分
1	https://github.com/vortex-ex/VIP		VIP	77	63	13	129	3	53	一般	26	13	VChristiaen	42.4	0	3090	7.23	10.93	18.17	
2	https://github.com/szhorvat/IGraphM		IGraphM	99	17	34	126	406	停滞	3	2	szhorvat	97.7	0	1680	2.82	0.02	2.84		
3	https://github.com/Computatisimcardem/simcardem		simcardem	13	12	16	52.2	14	活跃	8	6	finberg	67.1	10	1128	2.49	19.17	21.66		
4	https://github.com/Pizziut92/MG-MAMPOSSt		MG-MAMPOSSt	3	3	3	58.9	859	停滞	2	1	Pizziut92	99.6	0	450	0.88	0	0.88		
6	https://github.com/tkmonlige/city2ba		city2ba	20	1	1	71.7	1091	停滞	2	1	tkmonlige	98.8	0	84	0.33	0	0.33		
8	https://github.com/MartianCo/POSEIDON		POSEIDON	37	10	8	47	33	一般	5	3	MartianCo	72.4	0	1203	2.3	12.59	14.9		
9	https://github.com/JacksonOn/PyPPT		PyPPT	11	1	4	46.9	461	停滞	3	1	JacksonOnJ	98.8	0	248	0.67	0	0.67		
10	https://github.com/ianfhunter/GNOLL		GNOLL	48	24	57	81.3	177	一般	18	5	ianfhunter	92.7	0	1987	4.78	0.59	5.37		
11	https://github.com/ICESat-2-S-paper/paper		paper	0	4	0	40.8	1084	停滞	2	1	dshean	95.2	0	21	0.23	0	0.23		
12	https://github.com/N-Wouda/ALNE		ALNE	590	142	0	80.3	315	停滞	8	2	N-Wouda	86.3	0	197	1.1	0.12	1.21		
13	https://github.com/mgmn8/pyngm8		pyngm8	3	0	0	53.3	967	停滞	3	1	mgmn8	97.5	0	163	0.54	0	0.55		
14	https://github.com/sjbeckett/localcovit		localcovit	5	6	0	51.8	1078	停滞	4	3	sjbeckett	53.4	0	299	0.85	0	0.85		
15	https://github.com/glorenzo-rc/oxDNA		oxDNA	63	37	13	76.8	14	活跃	19	9	glorenzo-rc	59.8	1	1768	4.55	14.68	19.23		
17	https://github.com/BiomedSci/fuse-med-1		fuse-med-1	152	37	35	55.3	43	一般	31	9	mosherabot	28.3	0	546	5.42	11.86	17.28		
18	https://github.com/gropensci/spiro		spiro	15	0	0	58	120	一般	2	1	smnult	99.7	0	397	0.8	5.42	6.22		
19	https://github.com/Jammy2211/PyAutoGalaxy		PyAutoGalaxy	31	14	19	75.7	15	活跃	12	5	Jammy2211	91.1	14	3348	6.22	21.09	27.31		
20	https://github.com/martinlaclab/voting		voting	42	23	11	79.8	40	一般	11	4	martinlaclab	79.2	0	736	2.2	12.01	14.21		
21	https://github.com/annajenul/BarFS		BarFS	5	1	0	62.6	908	停滞	3	2	annajenul	94.9	0	360	0.84	0	0.84		
22	https://github.com/SuperKogi/spafe		spafe	479	78	0	76.8	291	停滞	10	2	SuperKogi	94.4	0	377	1.57	0.1	1.66		
23	https://github.com/schochast/signnet		signnet	23	7	1	77.9	61	一般	3	2	schochast	95.8	0	166	0.55	10.25	10.8		
24	https://github.com/rhdtownsemsg/localkovid19now		localkovid19now	12	3	4	60.5	62	一般	2	1	rhdtownsemsg	99.7	0	947	1.62	10.17	11.79		
25	https://github.com/nickelining/pykronckeck		pykronckeck	17	3	2	41.2	653	停滞	2	1	nickelining	96.5	0	85	0.33	0	0.33		
26	https://github.com/kungfugo/BellDifago		BellDifago	10	2	0	43.1	385	停滞	2	1	kungfugo	95.2	0	81	0.32	0	0.32		
27	https://github.com/Roessg/FijiRelax		FijiRelax	8	4	6	60	279	停滞	4	1	Roessg	98.8	0	245	0.77	0	0.77		
29	https://github.com/mlpack/mlpack		mlpack	5571	1689	25	134.6	17	活跃	264	159	rurcint	32.6	10	30714	41.44	20.03	61.47		
30	https://github.com/tud-hri/joint		joint	17	10	18	60	336	停滞	6	4	niekbecker	56.2	0	1701	3.15	0	3.15		
31	https://github.com/hydrosolutions/riverCentrifuge		riverCentrifuge	2	1	0	63.4	609	停滞	3	2	mabesa	68.7	0	434	0.95	0	0.95		
32	https://github.com/gparmo/parmoo		parmoo	87	11	15	48.3	5	活跃	2	2	thchang	65.5	0	703	1.25	15.02	16.27		
33	https://github.com/bjmorgan/py-sc-ferr		py-sc-ferr	29	9	6	77.5	209	停滞	4	2	alexquires	78.9	0	503	1.15	0.01	1.16		

三、可视化展示

可视化方面我们团队主要从两个方面着手，第一是利用 DataEase，选择数据源、创建数据集，最后制作仪表板并发布；第二是设计一个 html 网页以便使用者直接筛选合适的项目并且通过链接跳转。

(1) DataEase 制作的仪表板：

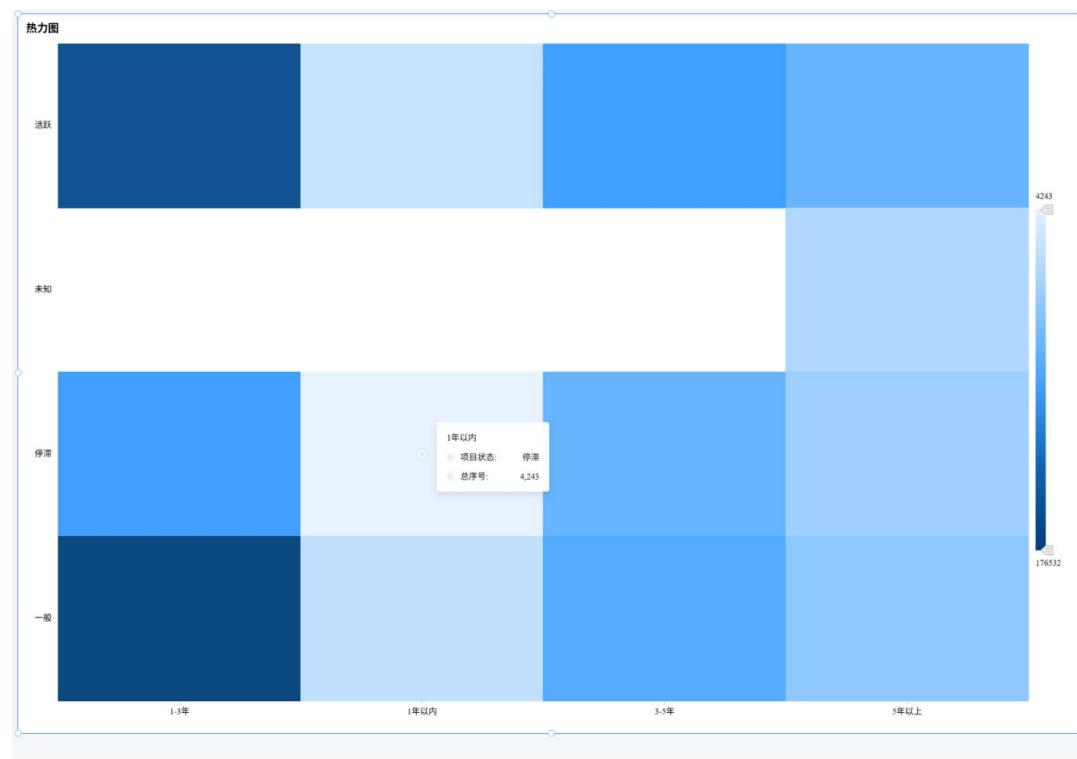
1. 贡献度分析视图：诊断社区协作结构与健康度



内容与设计目的：解构项目的社区协作模式，以评估其可持续发展的内在风险。我们采用柱线组合图，用蓝色柱状图表示论文项目中的贡献者总数，黄色折线图表示核心贡献者占比，将社区的参与规模与贡献集中度置于同一维度进行对比。

此设计使我们能够超越单一的“贡献者数量”指标，精准识别“巴士因子”风险：对于一个贡献者总数可观但核心贡献者占比极高的项目，其柱状图虽高，但折线图也处于高位，那么这就揭示了此项目过度依赖个别核心成员的脆弱性，是一种隐含的“广度假象”；反之，一个贡献者数量适中且贡献分布均匀（也就是折线相对较低）的项目，则表现出更健康的去中心化协作结构。该视图的核心目的是为评估者提供一个直观的工具，以快速判断项目的协作模式属于“健康型”、“高风险型”还是“潜力型”，为评估其长期可持续性提供关键依据。

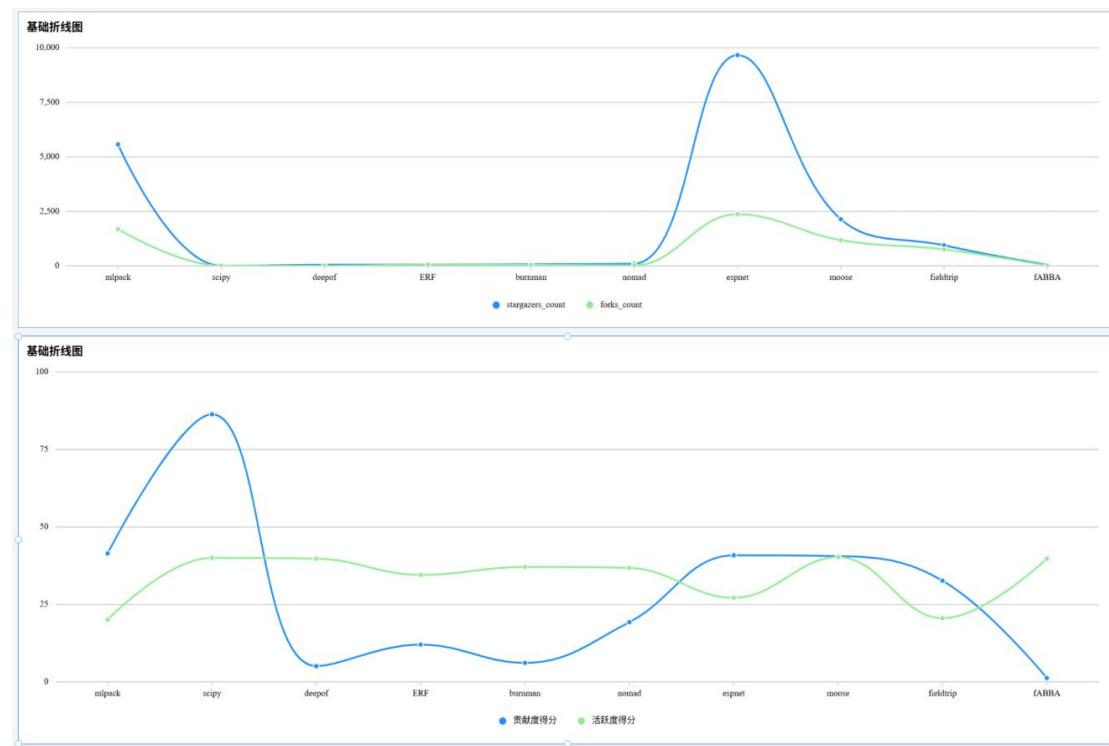
2. 活跃度热力图：追踪项目生命周期与状态分布

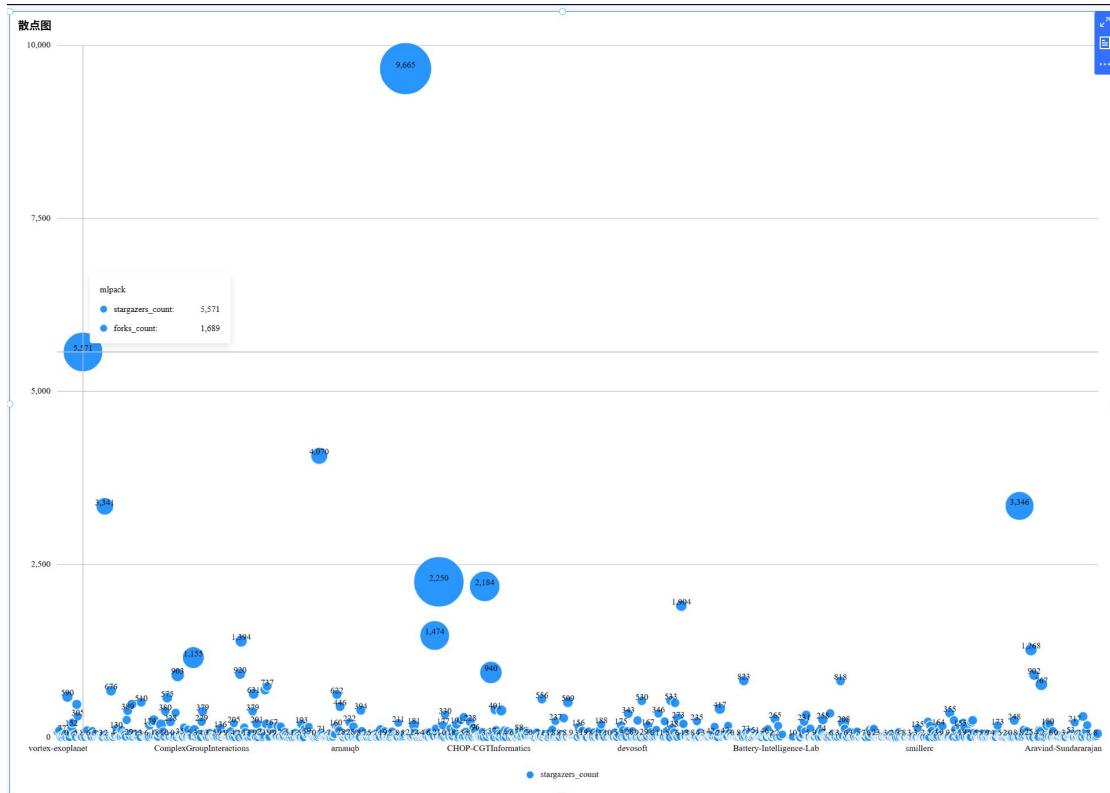


内容与设计目的：刻画项目群体在时间维度上的活跃状态演变规律，其设计采用矩阵布局，X 轴为项目年龄分组（1 年以内、1-3 年、3-5 年、5 年以上），Y 轴为项目健康状态（按活跃度降序排列：活跃、一般、停滞），单元格颜色深度编码处于该“年龄段-状态”的项目数量，越深则代表处于对应状态的项目数量越多。

此视图通过时空分布的视角，揭示了学术开源项目典型的生命周期模式与风险拐点。图中通常呈现几个关键模式：在“1 年以内”与“停滞”的交叉区域出现深色块，这量化了“发表即巅峰”现象的普遍性；而在“5 年以上”与“活跃”的交叉区域若存在深色块，则指示了历经时间考验的“长青项目”。该热力图的设计目的，是提供一种宏观的生态诊断工具，让使用者不仅能一眼看清整个生态的“健康高地”与“风险洼地”，还能预警项目在特定发展阶段的失活风险，从而理解项目生命周期的整体态势。

3. 进化度散点图：审视项目影响力与实质使用的均衡性





内容与设计目的：用于评估项目影响力与实质使用之间的匹配关系，从而衡量其“进化”的实质性。第一张折线图通过对 Top 10 项目的双指标对比，量化了项目影响力的真实构成；第二张散点图呈现了完整项目生态的进化梯度分布。

两张图表形成了从微观对比到宏观定位的完整分析链条。折线图的个体指标分离现象（如 **mlpack** 高星标但非最高复刻）在散点图中找到了生态位解释——**mlpack** 虽非复刻数最高，但其综合指标仍使其占据头部位置。这种关联表明：真正的优质项目需要在星标（影响力）与复刻（使用深度）间取得平衡。散点图的分布梯度进一步印证了折线图的发现——仅有少数项目能同时在两个维度上表现优异，这正是“进化度”评估要捕捉的核心特征：可持续的实质成长而非单点爆发。

(2) html 网页设计：

首先读取论文元数据 **JSONL** 文件和 **GitHub** 仓库评分数据，通过智能清洗验证机制剔除缺失关键信息的记录，确保数据质量。在特征工程环节，将发表时间转换为标准化时间维度特征（发表周、月份、年份），为时间序列分析提供基础。同时计算项目活跃度占比，并采用平方根缩放算法生成气泡大小参数，使可视化既能反映得分差异又保持美观。这些预处理步骤将原始数据转化为结构化、标准化的分析数据集。

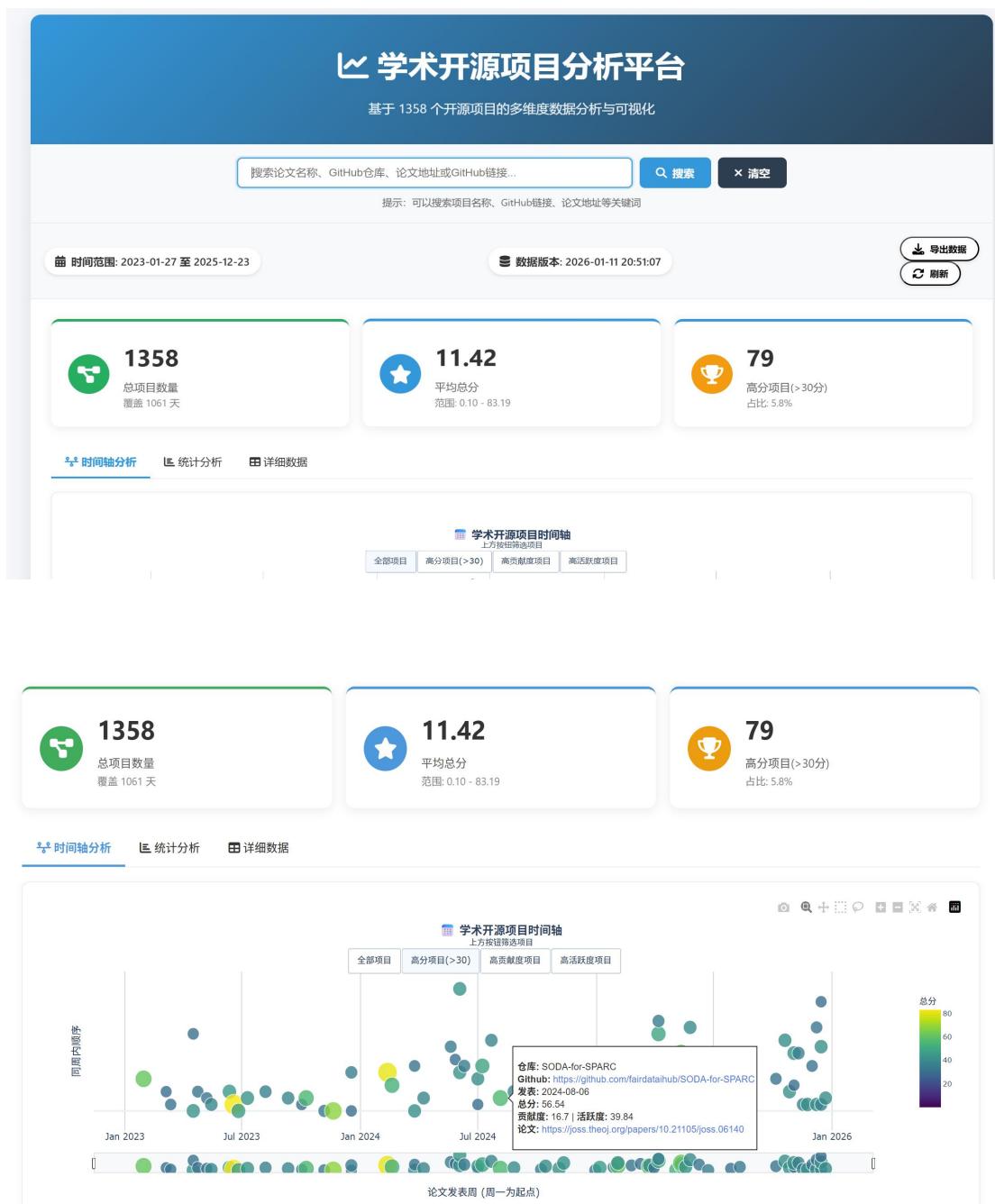
其次基于 **Plotly** 图表库构建多维度分析视图。核心时间轴采用气泡图形式，每个气泡代表一个开源项目，其大小和颜色深浅对应项目总分，支持鼠标悬停查看详细信息；此外生成四种辅助图表：直方图展示得分分布规律，散点图揭示贡献度与活跃度相关性，趋势线图呈现月度发表动态，饼图直观显示项目类型占比。所有图表均经过精心设计，确保信息传达直观清晰。

然后使用 **Jinja2** 模板引擎将分析结果嵌入 **HTML** 页面，采用响应式布局适配多终端设备。页面顶部设有智能搜索框，支持按项目名称、**GitHub** 链接等关键词的全文检索，结果实时显示并含直接访问链接。统计摘要区通过卡片式设计展示关键指标（项目总数、时间跨度、平均得分等）。选项卡导航使用户在时间轴、统计图表与详细数据间流畅切换。页面还提供数据导出等前端交互功能，为用户提供完整分析体验。

最后使用 **Pandas** 进行数据处理与统计分析，**Plotly** 生成交互式图表，前端基于 **HTML5**、**CSS3** 和 **JavaScript** 构建用户界面。架构采用模块化设计，评分算法、图表生成和网页模板相互独立，便于功能扩展和维护。最终生

成的网页不仅是静态报告，更是动态可交互的分析工具，帮助研究者快速发现高质量学术开源项目，评估项目生命力，为学术研究和开源生态建设提供数据支持。

生成网页效果如下：



其大致功能如下：

1. 多维度可视化分析

时间轴气泡图通过空间布局与视觉编码清晰呈现项目随时间演进的分布规律。允许使用者按要求进行搜索，例如“高分”、“高活跃度”、“高贡献度”，快速发掘网站上较好的论文项目。

2. 交互式探索能力

使用者可通过搜索框快速定位目标项目，检索结果直接关联至 GitHub 仓库或论文原文，鼠标悬停功能则允许无需跳转页面即可查看项目关键指标，提升信息获取效率。数据导出功能满足用户进一步离线分析的需求，增强了工具的实用性。

3. 可扩展性与学术意义

系统设计注重通用性，通过配置化调整可适配其他开源社区数据（如 GitLab 或 Bitbucket）。当前平台已为识别高质量学术开源项目提供量化依据，例如通过活跃度与贡献度的相关性分析，帮助评估项目可持续性。未来可扩展集成论文引用分析、社区协作网络图谱等功能，进一步深化对开源生态的解读。

四、项目未来展望

基于本项目的核心成果与发现，我们制定了系统性的未来发展蓝图，旨在从数据、智能、平台与生态四个维度进行深化与扩展，构建更完善、更智能的学术开源项目生命力评估体系：

1. 数据维度的扩展与深化

未来，我们将着力拓展数据源的广度与深度。横向层面，计划将数据采集范围从当前的 JOSS 平台延伸至 arXiv、ACL 等主流学术出版物平台，旨在构建一个跨学科、跨领域的论文-代码仓库关联网络，从而进行更宏观的生态对比研究；纵向层面，将引入 CI/CD 通过率、依赖项更新频率、安全漏洞修复时效等更精细的工程健康指标，使评估体系从“活跃度”表面观察，深入到“工程质量”内核诊断，全面刻画项目的长期维护能力。

2. 智能分析能力的升级

我们计划为平台注入更强的预测与动态评估能力。一方面，将基于项目的历史行为数据（如提交频率、Issue 响应时间、贡献者增长曲线）构建 LSTM 等时间序列预测模型，实现对项目“僵尸化”风险的早期预警，为用户选择依赖项目提供前瞻性洞察；另一方面，将建立动态的生命力指数排行榜，支持按研究领域、时间跨度等维度生成定制化的评估报告，从静态快照式评估转向持续性的健康度追踪。

3. 平台化部署与应用集成

为推动成果的广泛应用，我们将致力于打造轻量化工具与开放接口。计划开发浏览器插件，使用户在浏览 arXiv 等论文页面时，即可实时看到关联代码仓库的生命力评分，将评估能力无缝集成到科研工作流中；同时，平台将提供一套完整的 RESTful API，允许期刊评审系统、项目资助平台等第三方机构集成我们的评估数据，扩展其应用场景，赋能学术评审的数字化转型。

4. 社区化运营与生态激励

最终目标是推动学术开源生态的良性发展。我们计划设计一套“生命力

徽章”认证体系，对长期保持高健康度的优秀项目给予官方认证与展示，形成正向激励。同时，将建立“学术开源最佳实践”案例库，沉淀那些在工程规范、社区运营、可持续性方面表现卓越的项目的成功经验，为后来的研究者提供行动指南，从工具评估层面上升到知识沉淀与范式传播层面，最终促进整个学术开源文化的成熟与进步。