

Issue 净积压预警可视化大屏报告

——基于 OpenDigger Top300 指标的开源项目

内容摘要

本报告面向 OpenSODA 作品类 W2 赛道，基于 Top300 开源项目的 OpenDigger 指标数据，完成了数据清洗、结构化建模与可视化展示。Top300 日志数据是对 GitHub Top300 项目在 2020-01 至 2023-03 期间的协作行为事件，如 issue/PR 创建、关闭、评论、提交等的原始记录。

在完整处理流程中，我们使用 python 进行数据处理，将原始日志数据转化为易于处理的 csv 表格，产出 5 张核心分析表：openrank_long.csv、issue_ops_long.csv、pr_ops_long.csv、health_score_end.csv、alerts_end.csv。其中，由于部分治理效率指标在当前数据中缺失，综合可解释性与可视化落地性，本次大屏最终选择 alerts_end.csv 作为核心展示数据，聚焦“Issue 净积压（新增-关闭）上升”风险预警，并在 DataEase 中制作大屏完成展示。

关键词：开源社区治理、OpenRank、Issue 净积压、预警、DataEase

目录

一、 问题分析	3
1.1 赛题背景	3
1.2 研究问题定义	3
二、 模块化设计	3
2.1 读取数据	3
2.2 数据建模	4
2.3 数据可视化	5
三、 问题求解	5
3.1 数据来源	5
3.2 数据清洗	5
1. 结构清洗:	5
2. 类型清洗:	6
3. 对齐清洗:	6
3.3 处理结果	6
3.4 可视化图表说明	8
四、 创新点	10
五、 结论与反思	10
5.1 研究结论	10
5.2 局限性	10
5.3 后续改进方向	10
六、 参考文献	11
七、 附录	11

一、 问题分析

1.1 赛题背景

开源数字生态由大量分布式协作活动构成：贡献者在 GitHub 等平台持续进行 Issue 讨论、Pull Request 协作、代码提交与评审迭代，形成复杂且高度动态的社区网络。随着开源项目规模扩大，单靠经验已难以支撑高效治理与运营，开源社区迫切需要以数据为基础的洞察工具与决策系统，用于持续监控社区状态、识别风险并指导资源投入。

因此，本次 OpenRank/OpenSODA 相关赛题以开源生态数据为基础，鼓励参赛者围绕“数据洞察、治理、运营与预测”等方向开展创新实践。官方推荐的作品主题覆盖多个关键场景：包括开源项目与社区的数据洞察与可视化作品、开源社区治理与运营工具、开源社区数据的时序分析与可视化、开源项目健康度监控与指标设计、活跃度与趋势预测、贡献者行为分析与协作网络图分析等。参赛作品不仅要呈现数据结果，更强调对治理与运营问题的解释能力与可行动性输出。

在这一背景下，Top300 项目作为开源生态的核心样本，天然适合构建“治理与运营驾驶舱”：一方面通过 OpenRank 等指标反映项目影响力与生态格局变化；另一方面通过 Issue/PR 等运营指标刻画协作效率与治理风险（例如新增与关闭不平衡导致的积压压力）。基于这些指标构建可视化大屏，可以将复杂的时序指标转化为直观的风险分布、重点项目排行与预警提示，帮助治理者快速定位问题、制定治理优先级，并为后后续引入健康度指标设计、趋势预测与图分析模型提供数据基础。

1.2 研究问题定义

本研究聚焦的主题是如何在 Top300 项目中快速识别“当月 Issue 净积压显著上升”的项目，并形成可视化预警看板，辅助社区治理与资源调度。

原因如下：

选择“当月 Issue 净积压显著上升”作为预警核心指标，是因为它直接反映项目当期需求压力与处理能力之间的缺口，具有覆盖面广、口径清晰、可解释、可行动的特点，能够作为开源社区治理的低成本早期预警信号，为资源调度与治理介入提供明确依据。

二、 模块化设计

整体方案共分为以下 4 部分：选取数据，数据建模，数据可视化，数据解读。

2.1 读取数据

本次大赛共提供了三份数据供参赛者探索，第一份是 GitHub 2020 年 1 月份的日志数据，第二份是 Top 300 仓库的从 2020 年 1 月到 2023 年 3 月的日志数据，第三份是 Top 300 仓库的 OpenDigger 的指标数据，包括 OpenRank 值。我们下载了这三份数据，简单梳理了这三份数据的关系。第一份 2020_01_log 数据大小为 51G，涵盖了

2020 年一月 GitHub 上所有项目的行为日志，完整记录了 2020 年 1 月 GitHub 上发生了什么事。第二份 top300_log 数据大小为 78G，进一步聚焦影响力排名前 300 的项目，记录了这 300 个项目 2020-2023 年期间的项目行为日志。第三份数据则是基于第二份日志数据得到的初步处理结果，较前两份详尽的记录数据而言，这份 top300 的指标数据体量较小，仅 48M，但是相较于前两份数据而言，使用价值更高。因此，综合这三份数据集的内容、大赛要求、参赛主题，我们最终决定选择 top300_metrics 数据作为我们的分析对象。

2.2 数据建模

Top300 metrics 的数据可分为以下四层结构：组织、仓库、指标、指标数据。第一层结构包含了 300 个文件夹，每个文件夹以组织命名。第二层结构是仓库文件夹，以每个组织的 GitHub 仓库命名。第三层是指标 json 文件，内容是每一个仓库的若干指标数据，包括核心影响力指标：openrank.json、stars.json 关注度指标等指标，具体指标见图 1。第四层是指标数据，json 文件的核心结构是月度时间序列，即以具体的年月为 key，当月对应的指标值为 value。

Top300 metrics 数据结构清晰，数据完整，支持自动化数据分析。所以对于 top300 metrics 的数据建模，我们使用 python 编写脚本，遍历每个组织对应的仓库，获取项目对应的指标数据，并将所得数据导出为 5 张 csv 图表，便于下一步使用 DataEase 软件对数据进行深入分析。

下面这张图展示了 top300_metrics 数据的四层结构：

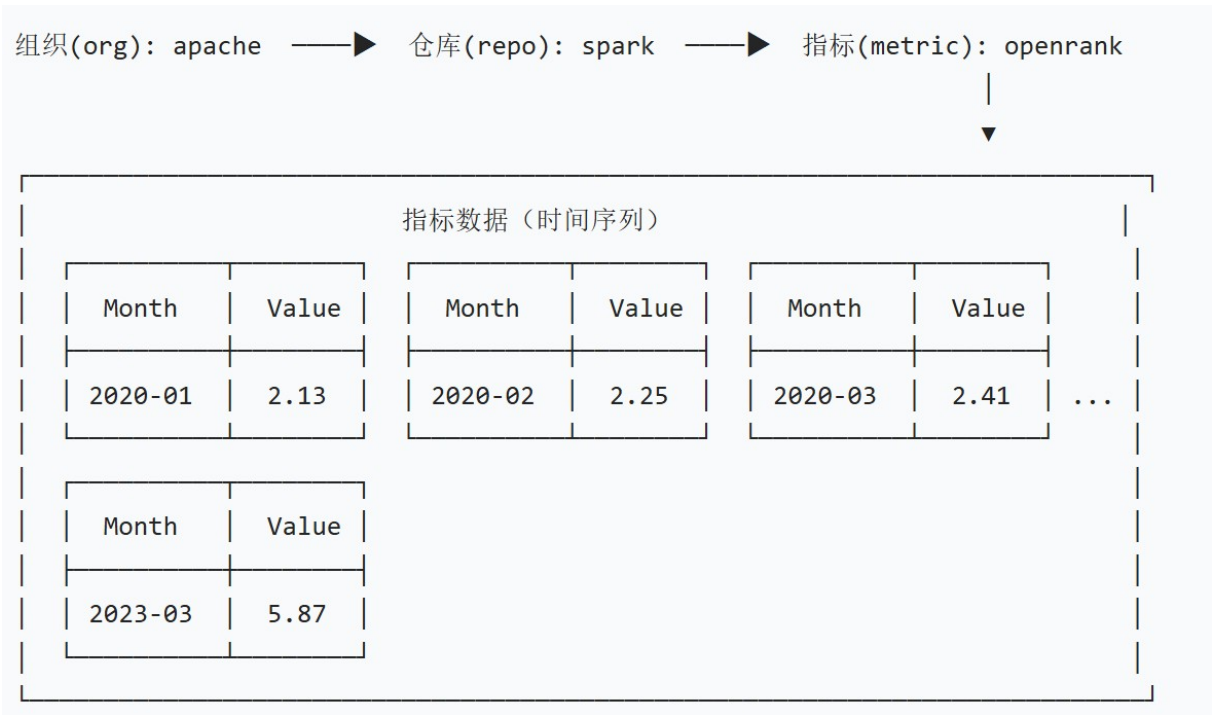


图 1：Top300_metrics 数据结构图解

2.3 数据可视化

OpenSODA 鼓励围绕开源数字生态做数据洞察、治理与运营类创新应用。针对大赛要求，我们的可视化治理大屏主要从两方面展示数据。

首先从影响力视角，我们的大屏计划展示排名前列的项目，帮助治理者准确把握热点项目方向。其次从社区治理角度，我们使用根据得出的预警规则，统计出最危险的项目，为治理者的资源调度和提前介入提供参考。

三、 问题求解

3.1 数据来源

使用 OpenSODA 提供的数据集 Top300_metrics 数据。Top300_metrics 中 json 文件的含义如下图所示。

文件名称	描述		
		contributor_email_suffixes.json	贡献者邮箱后缀
active_dates_and_times.json	项目每天的活跃度	inactive_contributors.json	不活跃开发者数
activity_details.json	每人每天的活跃度	issue_age.json	issue的生命时长（没有关闭的PR默认到23年3月份）
activity.json	项目每月的活跃度	issue_comments.json	issue的评论数量
attention.json	每月的关注度	issue_resolution_duration.json	issue的开启到关闭的时长
bus_factor_detail.json	每人每月的巴士系数	issue_response_time.json	issue从开始到首次响应的时长
bus_factor.json	项目每月的巴士系统	issues_and_change_request_active.json	issue和request的数量
change_request_age.json	PR请求的生命时长（没有关闭的PR默认到23年3月份）	issues_closed.json	issue关闭的数量
change_request_resolution_duration.json	PR请求从创建到结束的	issues_new.json	issue创建的数量
change_request_response_time.json	PR请求从创建到首次响应的时长	new_contributors_detail.json	新增加的贡献者名单
change_requests_reviews.json	PR审阅者的数量	new_contributors.json	新增加的贡献者数量
change_requests.json	PR的数量	openrank.json	openrank值
code_change_lines_add.json	代码添加的行数	participants.json	项目参与者人数
code_change_lines_remove.json	代码减少的行数	stars.json	star的数量
code_change_lines_sum.json	代码总变更数	technical_fork.json	fork数量

脚本读取以下数据，并依次进行数据清洗。 读取的数据有：

openrank.json

issues_new.json

issues_closed.json

issue_response_time.json

issue_resolution_duration.json

change_requests.json

change_request_response_time.json

change_request_resolution_duration.json

1. 结构清洗：

原始 top_300_metrics 数据结构是这样的：项目在文件夹名里，指标在文件名里，月份在 JSON 的 key 里，这三层全是隐式的，无法直接拿来分析。结构化清洗的目的是把数据变成标准的表格形式，把文件夹名变为表格的 repo_full 列，把 JSON key 变为表格

month 列，把 JSON value 变为表格的指标列。这一步是后面所有分析的基础。

2. 类型清洗:

原始数据中存在部分缺失现象，有些数值数据是以字符串的形式存储。为了确保后续分析数据的正确性，需要将数值字段转为数值型，并且将缺失的数据保存为 NA 空值。

3. 对齐清洗:

脚本读取到的很多指标如 issues_new,issues_closed 等，这些指标不一定每个月都有值，也不一定每个项目这些指标都齐全。脚本以仓库名 repo_full 和 month 为唯一坐标，把同一个项目、同一个月的不同指标并排放在同一行，便于后续导入 DataEase 进行图表绘制。

3.3 处理结果

运行脚本，得到五张 csv 数据表:

issue_ops_long.csv / pr_ops_long.csv: 提供治理效率时序(用于折线趋势、下钻分析)。

health_score_end.csv: 希望将多个维度归一化后形成综合健康评分(红/黄/绿)。

alerts_end.csv: 将“治理风险”转成可直接展示与行动的预警清单。

Openrank_long.csv: 提供影响力榜单与增长榜单(用于“生态影响力侧”展示)。

由于后续只选用 alerts_end.csv 进行详细的分析，所以下面只对针对这五张表格读取的数据、囊括的指标与使用价值进行简单分析:

1.issue_ops_long.csv 表格。这张表格读取了每个项目目录中 issue 相关的 JSON 文件，每月新建的 issue 数、每月关闭的 issue 数、issue 首次响应时间、issue 解决时长。这是 issue 治理分析的基础时序表，可用于回答项目每个月的 Issue 压力是变大还是变小，新增和关闭是否平衡以及响应与解决效率是否存在长期恶化趋势。表格前十行数据如所示。

1	repo_full	org	repo	month	issues_new	issues_closed
2	AUTOMATIC	AUTOMATIC	stable-di	Aug-22	26	14
3	AUTOMATIC	AUTOMATIC	stable-di	Sep-22	922	615
4	AUTOMATIC	AUTOMATIC	stable-di	Oct-22	1243	611
5	AUTOMATIC	AUTOMATIC	stable-di	Nov-22	459	187
6	AUTOMATIC	AUTOMATIC	stable-di	Dec-22	327	202
7	AUTOMATIC	AUTOMATIC	stable-di	Jan-23	576	762
8	AUTOMATIC	AUTOMATIC	stable-di	Feb-23	318	138
9	AUTOMATIC	AUTOMATIC	stable-di	Mar-23	434	185
10	AdguardTe	AdguardTe	AdguardFi	Jan-15	60	57
11	AdguardTe	AdguardTe	AdguardFi	Feb-15	45	46

图 2: issue_ops_long 表格前十项数据预览

2.pr_ops_long.csv 表格。这张表格读取了来自 每个项目目录中的 PR (Change Request) 相关 JSON 文件，囊括的指标有每月新建 PR 数、PR 首次响应时间、PR 合并/关闭时长。可用于分析: 项目是否存在 PR 堆积? Review / 合并是否变慢? 项目协作是否因规模扩大而效率下降? 表格前十行数据如图示。

1	repo_full	org	repo	month	change_requests
2	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Aug-22	12
3	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Sep-22	266
4	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Oct-22	517
5	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Nov-22	200
6	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Dec-22	123
7	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Jan-23	191
8	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Feb-23	102
9	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Mar-23	141
10	AdguardTeam/AdguardFilters	AdguardTeam	AdguardFilters	Feb-16	1
11	AdguardTeam/AdguardFilters	AdguardTeam	AdguardFilters	Aug-16	1

图 3: pr_ops_long 表格前十项数据预览

3.openrank_long.csv 表格。这张表格读取了项目的 openrank 值。Openrank 值是 OpenDigger 基于 GitHub 行为网络计算得到的项目影响力指标。这张开原生态影响力的核心时序表可用于分析项目影响力的长期变化趋势，对比不同项目的影响力轨迹和计算 Top300 的整体影响力走势。表格前十行数据如图所示。

1	repo_full	org	repo	month	openrank
2	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Aug-22	6.09
3	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Sep-22	335.01
4	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Oct-22	858.02
5	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Nov-22	763.82
6	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Dec-22	662.23
7	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Jan-23	736.33
8	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Feb-23	650.07
9	AUTOMATIC1111/stable-diffusion-webui	AUTOMATIC1111	stable-diffusion-webui	Mar-23	736.73
10	AdguardTeam/AdguardFilters	AdguardTeam	AdguardFilters	Jan-15	2.07
11	AdguardTeam/AdguardFilters	AdguardTeam	AdguardFilters	Feb-15	2.78

图 4: openrank_long 表格前十项数据预览

4.health_score_end.csv 表格。这张表格并不是直接来自某个 JSON，而是综合多张表格最后一个月份的指标数据，包括 issue_ops_long.csv（Issue 相关指标）、pr_ops_long.csv（PR 相关指标）、openrank_long.csv（影响力指标）。这张表格设计的初衷是尝试把多个治理与影响力维度合成为一个项目健康度评分，但是由于这批数据中存在部分指标缺失，导致健康分数无法稳定计算。

5.alerts_end.csv 表格。这张表格主要来源于 issues_ops_long.csv 的期末月份数据，依赖两个最稳定最完整的指标 issues_new 和 issues_closed。这张表的指标包含 repo_full: 仓库全名、alert_type: 预警类型（如 Issue 积压上升 / 严重）、value: 当月 Issue 净积压变化量、threshold: 触发该预警的阈值和 reason: 规则说明文本。其中当月 issue 净积压变化量是 issue_new 与 issue_closed 之差。表格前十行数据如图所示。

	repo_full	month	alert_type	severity	metric	value	threshold	reason
1	AUTOMATIC1111/stable-diffusion-webui	Mar-23	Issue积压严重	红	backlog_end	249	50	新增>关闭>50, 积压上升
2	ArduPilot/ardupilot	Mar-23	Issue积压上升	黄	backlog_end	36	10	新增>关闭, 积压开始累积
3	Automattic/jetpack	Mar-23	Issue积压严重	红	backlog_end	53	50	新增>关闭>50, 积压上升
4	Automattic/wp-calypso	Mar-23	Issue积压上升	黄	backlog_end	34	10	新增>关闭, 积压开始累积
5	Azure/azure-cli	Mar-23	Issue积压严重	红	backlog_end	52	50	新增>关闭>50, 积压上升
6	Azure/azure-powershell	Mar-23	Issue积压上升	黄	backlog_end	41	10	新增>关闭, 积压开始累积
7	Azure/azure-sdk-for-java	Mar-23	Issue积压上升	黄	backlog_end	15	10	新增>关闭, 积压开始累积
8	ClickHouse/ClickHouse	Mar-23	Issue积压严重	红	backlog_end	113	50	新增>关闭>50, 积压上升
9	Expensify/App	Mar-23	Issue积压严重	红	backlog_end	322	50	新增>关闭>50, 积压上升
10	Expensify/App	Mar-23	Issue积压严重	红	backlog_end	322	50	新增>关闭>50, 积压上升
11	JuliaLang/julia	Mar-23	Issue积压上升	黄	backlog_end	31	10	新增>关闭, 积压开始累积

图 5: alerts_end 表格前十项数据预览

本研究只选择 alerts_end 的原因:

issue_ops_long.csv 与 pr_ops_long.csv 的 响应时长/解决时长相关字段确实较严重, 直接导致 health_score_end.csv 的 health_score 无法计算(大量为 NA), 健康度分层(红/黄/绿)失真, 无法支撑可靠的大屏展示。

因此, 本次大屏优先选择 数据最完整、规则最清晰、可解释性最强的 alerts_end.csv:

alerts 的核心计算只依赖 issues_new 与 issues_closed (在数据中存在且可用), 能稳定生成预警。

3.4 可视化图表说明

绘图方法: 利用开源 DataEase 工具, 使用 alerts_end.csv 数据绘制了 4 张图。

我们的可视化治理大屏由以下四个板块组成: 告警数量类型分析、项目净积压变化 Top10、value 和 threshold 对比、规则触发条件占比可视化。

板块 1: 告警类型数量分析

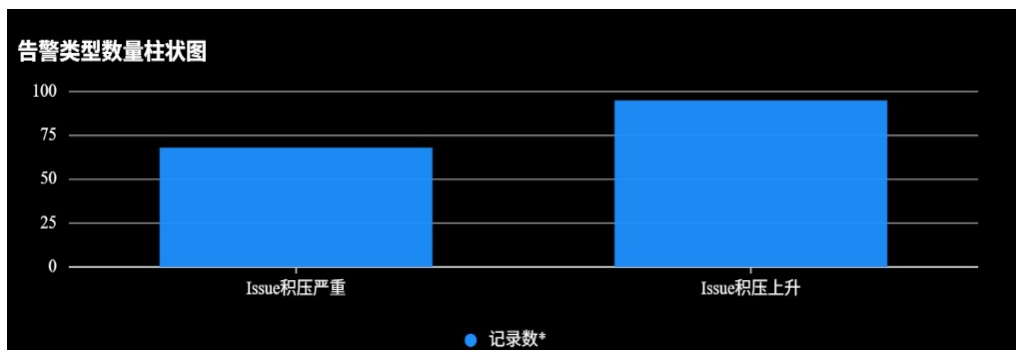


图 5: 板块 1 预览

该板块对预警类型数据进行分析, 用可视化的方式展示了项目的预警情况:

95 个项目 “Issue 净积压上升”, 占比为 58.3%, 68 个项目 “Issue 净积压严重”, 占比为 41.7%。这张图对预警风险进行分层盘点, 发现多数项目已出现净积压上升, 近四成达到更高阈值, 需要优先治理。

板块 2: 项目净积压变化 TOP10

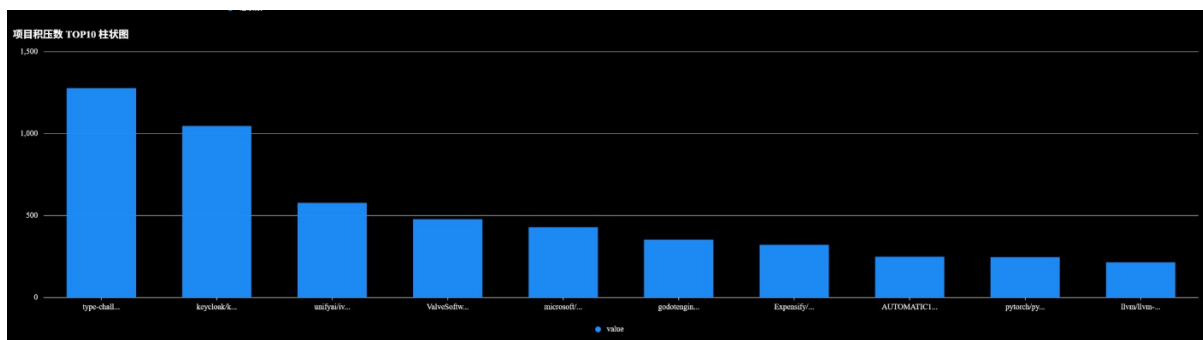


图 6：板块 2 预览

该板块分析了 2023-03 单月净积压变化量的极端项目。以图中排名 Top1 的项目为例：该项目当月 $\text{issues_new} - \text{issues_closed} = 1278$ ，新增远超关闭，说明这个项目存在明显处理能力不足或者协作拥堵风险，社区治理时需要重点关注。项目净积压变化 TOP10 榜单可以帮助治理者快速锁定最需要协助治理的项目，为专项治理与资源倾斜提供治理建议。

板块 3：value vs threshold 对比板块

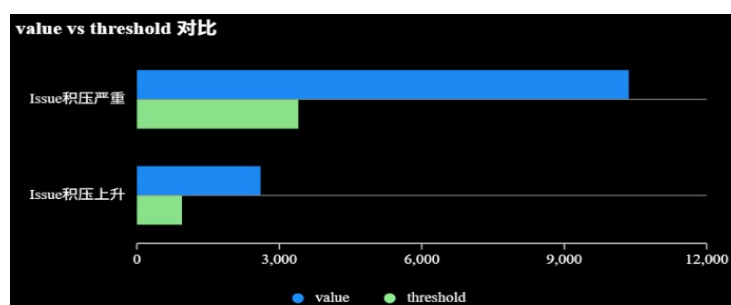


图 7：板块 3 预览

我们将项目的风险划分为两档，分别是严重档和上升档。为验证我们设置的预警规则确实能够起到较好的划分作用，在这张图中，我们析了不同预警档位下，平均净积压变化量与阈值的差距，并且发现严重档和上升档的数值确实存在显著差异。从图上数据可知：严重档：平均 $\text{value} \approx 152$ ，阈值=50，上升档：平均 $\text{value} \approx 27$ ，阈值=10。这张图证明了预警规则能把风险“拉开层级”，严重档确实显著高于阈值，提示更紧急的治理优先级。

板块 4：规则触发条件占比可视化

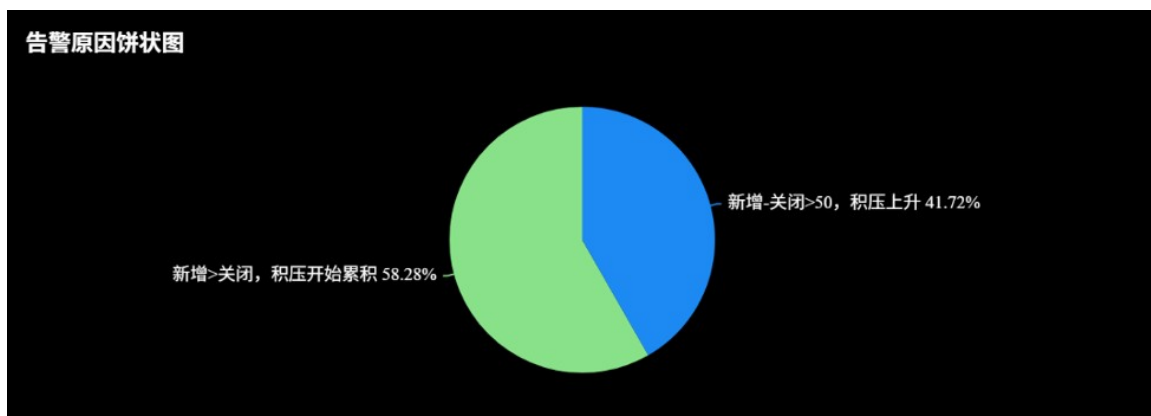


图 8：板块 4 预览

我们使用了 alerts 表里的 reason 字段，该字段属于规则标签。需要注意的是，这个字段并不是触发预警的真实原因归因，而是触发预警的规则分布。本次预警目前只覆盖“净积压”维度，后续可以补齐响应慢/解决慢/PR 慢等维度，探索完整的预警原因。

四、 创新点

很多作品停留在展示指标曲线，而本作品将 issue 新增/关闭转为可行动的预警清单 (alerts)，直接回答“先治理谁”。

通过 Python 脚本将 JSON 时序摊平成 CSV，统一键 (repo_full、month)，使 DataEase 可直接聚合、筛选与做大屏组件复用。

五、 结论与反思

5.1 研究结论

在 2023-03 的 Top300 项目中，有 163 个项目出现当月 Issue 净积压显著上升 (alerts 记录数)。

其中 41.7% 达到更高阈值档位，属于更紧急的治理风险，需要优先关注与资源投入。

预警 TOP10 项目呈现出“新增远超关闭”的典型治理压力特征，适合做专项治理对象。

5.2 局限性

由于 response, resolution 等指标的缺失，当前大屏只覆盖“净积压”单一维度，暂时无法回答为什么处理慢。

reason 字段只是规则标签，不能当作真实原因归因。

由于读取数据时未引入“存量 open issues”字段，本报告不讨论总积压规模，仅讨论“当月增量压力”。

5.3 后续改进方向

补齐响应/解决时长指标后，完善健康分数与多维预警（响应慢/解决慢/PR 慢/积压

上升)。

引入全局趋势 (Top300 SUM/AVG 的 OpenRank 趋势折线) 与项目下钻页, 形成 “影响力 + 治理” 的完整驾驶舱闭环。

六、 参考文献

- [1] OpenSODA 数据集说明 (Top300、时间范围、指标数据构成)。
- [2] OpenDigger 指标文档: OpenRank 指标说明。
- [3] OpenDigger 指标文档: Issue Response Time 指标说明 (用于治理效率维度定义参考)。

七、 附录

附录 A: 核心产出

openrank_top_end.csv / openrank_top_end.png: 2023-03 OpenRank Top1

openrank_top_growth.csv / openrank_top_growth.png: 区间增长 Top15

alerts_end.csv: 2023-03 净积压预警清单

issue_ops_long.csv: Issue 运营时序表 (99 个月)

pr_ops_long.csv: PR 运营时序表 (99 个月)

health_score_end.csv: end 月健康度尝试

build_long_tables.py 和 build_health_and_alerts.py: 数据处理脚本

附录 B: 核心计算口径

净积压变化量: issues_new - issues_closed (按 repo、按 month)

预警分层:

上升: 净积压变化量 > 10

严重: 净积压变化量 > 50