开源项目健康度监控与指标设计

创新科技　梦想未来

目 录

contents

1 项目背景及设计理念

2 健康度算法设计思路与实现

3 项目实施成果展示

0

contents

1

项目背景及设计理念

Project Background and Design Concept

项目背景及设计理念

项目背景

在应用开发，云计算，人工智能等技术不断迭代的数字生态中，开源社区的维护成为技术发展与创新的摇篮

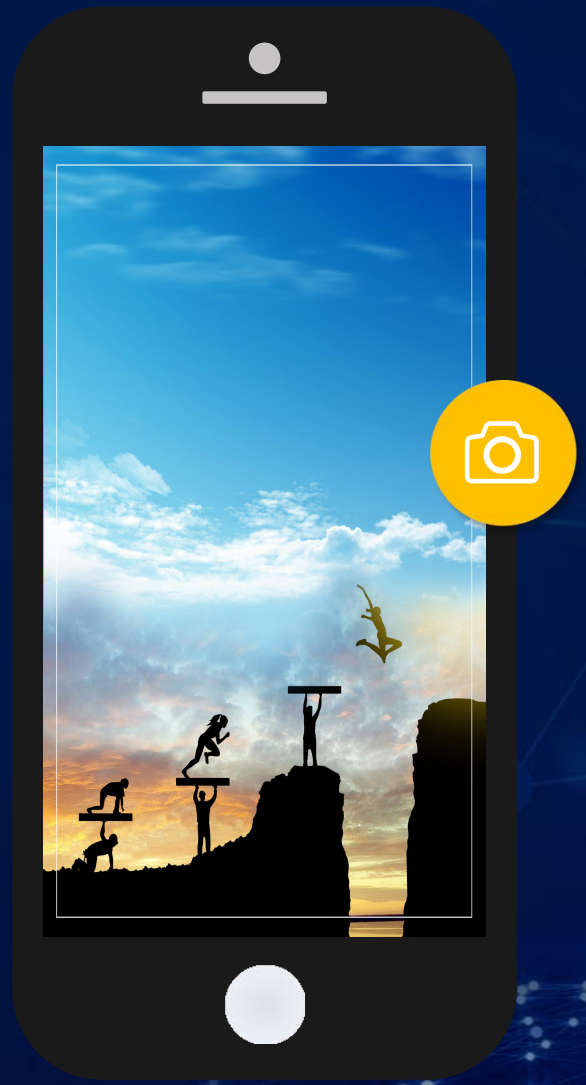
设计思路

如何进行有效的开源社区治理，使之不断促进协作，激发创新，并为技术爱好者，个人开发者以及大型企业决策提供有价值的参考是当下实现良好开发环境的前提和根本

设计理念

基于开源工具opendigger实现的指标，我们得以对开源项目的所有活动以及其与其他项目的关联程度进行数据分析，并依照过往经验及健康标签化数据的学习对开源项目的健康值进行较为科学的评估和预测

项目背景及设计理念



健康度预测

关键指标

设计理念

我们旨在通过良好的健康度可视化并通过合适的模型对未来健康度预测，为开发者提供直观的项目评估服务

简洁的交互接口便于实时得到健康度背后的关键指标，使得开发者能够基于本工具进行代码的本地部署与二次开发

02

contents

健康度算法设计思路与实现

Design Ideas and Implementation of Health Degree Algorithm

健康度算法设计思路与实现

变量列举

数据来源于top_300_metrics，综合考虑活跃度算法和价值流网络，我们最终采用以下字段：

1.Activity
2.Stars
3.Attentions
4.Issue_resolution_duration
5.Issue_response_time
6.Issue_comments
7.Change_request_resolution_duration
8.Change_request_response_time
9.Participants
10.Openrank
11.Technical_fork
12.New_contributors
13.Inactive_contributors

其中4，5，7，8均取时序数据中当月的平均值

健康度算法设计思路与实现

相关性的取定

4，5，7，8的response_time和resolution_duration，均为时间越短代表

活跃度越高越健康，所以呈负相关性

相关性的取定

13的inactive_contributors显然也是越多代表项目越不健康，所以也是负相关的

相关性的取定

1，2，3，6，9，12这些指标都代表着社区的活跃度和关注度，所以也是越高代表此项目越健康

相关性的取定

10和11比较特殊，是关于这个项目的技术分支和与其他开源项目的关联性，是开源项目具有稳定性和社区的依赖性相关性，也是越高代表着这个社区越健康，所以以上值取正相关

健康度算法设计思路与实现

第一步

在给到的所有数据量中，我们先看单个项目的单种数据的变化，第一个时间节点取0值，然后用当前时间下的数据减去之前所有数据的平均值，这个值显然能看出在某个时间下对于之前的数据的变化量

第二步

然后对其进行归一化，在同一个数据中，项目各自的值去除以绝对值的最大值，计算后的值范围在-1~1之间，这样就能直观地表示这个项目的这个数据相对于其他项目的的同种数据的差距

第三步

然后在算出来的所有数据后进行加权，然后以给出的所有项目的结果进行截取然后得到健康的标准，关于权值和健康度的标准的取定我们希望通过之后的计算比较之后再进行确定

03

contents

项目实现方案

Project Implementation Plan

项目实现方案

开发形式

综合考虑跨平台性和简洁易用的特性，

本工具采用web app的形式进行开发

构建和维护方式

我们小组现阶段拟定通过MySQL+React+Flask技术栈作为项目构建和维护的方式

项目实现方案

01 前端开发

- 在React项目中，使用组件化开发方式构建用户界面。
- 使用React的状态管理（如useState、useReducer）和生命周期方法（如useEffect）来处理用户交互和数据更新。
- 使用Axios或Fetch API等库来向后端发送HTTP请求，获取或提交数据。

02 后端开发

- 在Flask项目中，定义API接口来处理前端的请求。
- 使用Flask的路由装饰器来映射URL到视图函数。
- 在视图函数中，处理业务逻辑，与数据库交互，然后返回JSON格式的数据给前端。

03 前后端交互

- 前端通过API接口与后端进行通信，获取所需的数据或提交用户输入。
- 后端接收前端的请求，处理完毕后返回响应数据。

04 模型预测

- 使用MLP对项目健康度进行预测

使用React对结果进行可视化