1. 项目背景与目标

本项目旨在构建一个开源项目健康度监控与评价系统，通过对多角色（创建者、维护者、贡献者、用户）的主观与客观指标的分析，量化开源项目的健康状况，并通过可视化展示结果，从而促进开源生态的可持续发展。

二、数据采集与处理

1. 数据来源与工具

数据来源于Opendigger工具，采集了GitHub上前300个流行开源项目的14个核心指标，包括关注度（attention）、参与人数（participants）、PR 响应时间（change\_request\_response\_time）等。使用IoTDB作为存储数据库，适用于高效管理和查询时序数据。

2. 数据分类与转换

定义classify函数，分类处理JSON数据，并将其转化为CSV格式，用于后续数据库存储和处理。

三、数据存储与导入

1. 数据库存储

使用IoTDB数据库作为核心存储工具，支持高效的时序数据查询。编写自动化脚本通过load\_index函数批量导入300个项目的14个指标数据至IoTDB数据库中。

2. 数据缺失处理

定义process\_matrix函数处理决策矩阵中的缺失值：非NaN值以该列均值替代；全NaN列则用0填充，确保数据完整性。

3. 数据清理

通过Python-SQL语句计算数据库中的指标均值，生成初始决策矩阵，用于后续评价与分析。

四、指标评价与算法实现

1. 决策矩阵构建

使用cal\_matrix函数生成300个项目×14个指标的矩阵，作为健康度分析的核心数据结构。

2. 使用CRITIC 算法计算指标权重

定义critic\_weight函数：①标准化数据，消除量纲影响②计算每列标准差，确定指标区分能力③构造相关性矩阵，评估指标间的信息冗余④最终生成权重向量，权重越高表明指标对健康度评价的影响越大

3. 角色满意度计算

基于不同角色的关注点，提取对应指标：

①创建者：项目参与者人数（participants）、每月的关注度（attention）、fork数量（technical\_fork）和openrank值（openrank）

②维护者：项目每月的巴士系统（bus\_factor）、代码总变更数（code\_change\_lines\_sum）、PR 数量（change\_requests）和issue与request的数量（issues\_and\_change\_request\_active）

③贡献者：PR请求从创建到首次响应的时长（change\_request\_response\_time）、PR审阅者的数量（change\_requests\_reviews）、issue从开始到首次响应的时长（issue\_response\_time）和openrank值（openrank）

④用户：start的数量（stars）、项目每天的活跃度（active\_dates\_and\_times）、issue的评论数量（issue\_comments）和openrank值（openrank）

计算角色满意度：定义to\_critic\_and\_to\_final函数计算每个角色的满意度，并输出300个项目×4个角色的满意度总表。

4. 项目健康度评分

使用自创的关联权重算法：①计算角色间的相关性矩阵，减少高相关性对结果的干扰②根据角色相关性调整权重，确保评价更科学合理③生成健康度评分表，用于反映项目整体健康状态

五、数据可视化

1. 结果导出

使用Pandas导出数据至CSV文件，供DataEase可视化，导出的数据包括：14 张指标表、4 张角色满意度表和 1 张健康度总表

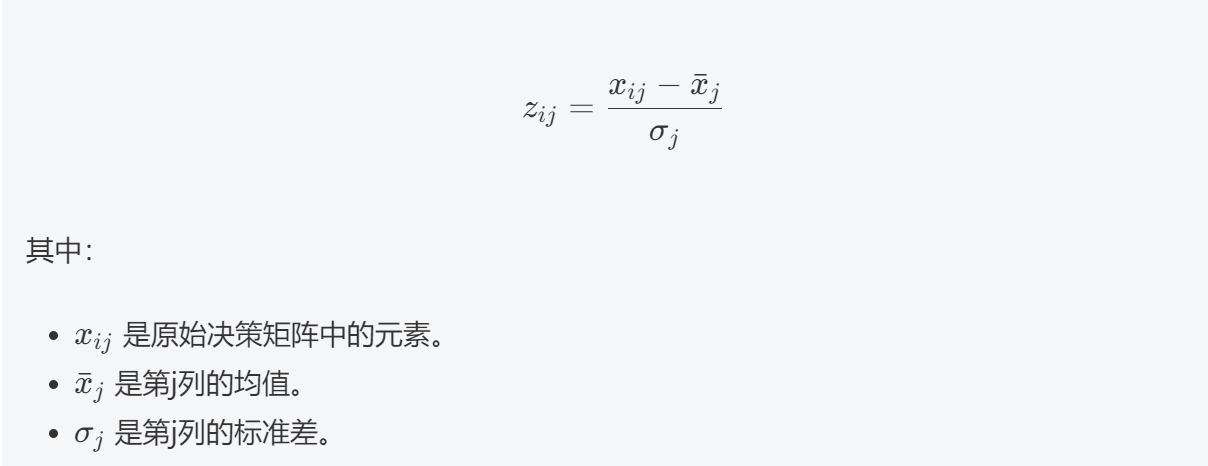
定义process\_data函数，将导出的用于可视化的数据的时间格式统一为 “YYYY-MM-DD”，使可视化图标更加美观。

1. 工具与方法

将代码中导出的数据导入DataEase创建交互式仪表盘，通过条件筛选设置，实现与用户的交互，允许查找指定项目的指标数据

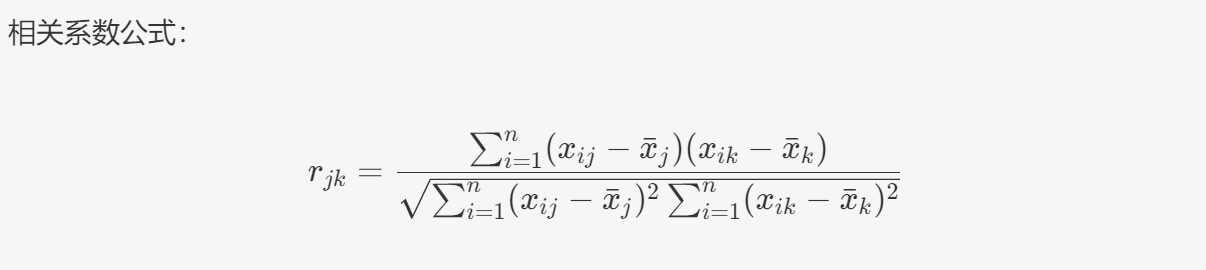
1. 算法具体细节
2. Critic算法：

①标准化数据：对于之前计算出来的决策矩阵总表进行数据标准化，CRITIC方法首先要求对原始数据进行标准化处理，目的是消除量纲的影响。标准化的方法通常是通过将每个元素减去该列的均值，并除以该列的标准差（Z-score标准化）。具体公式如下：

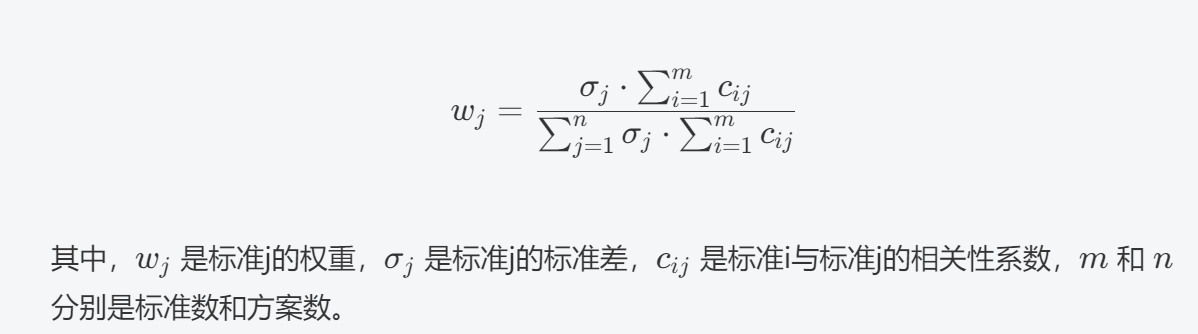


②计算标准差：对于标准化后的决策矩阵总表的每一列（代表一个指标）进行计算标准差

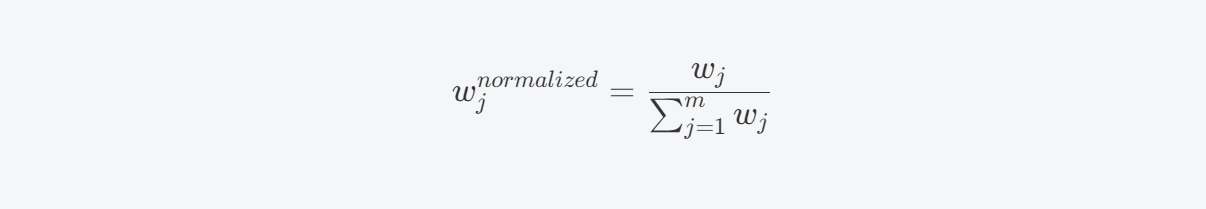
③计算相关性矩阵：对于标准化后的决策矩阵总表的每一列（代表一个指标）和其他列都进行相关系数的计算，从而得到相关系数矩阵计算标准之间的相关性（即标准之间的皮尔逊相关系数）。相关性越强，意味着这些标准之间的信息冗余越高，因此它们的重要性会被调整。具体公式如下：



④计算权重：根据标准差和相关系数，结合公式计算每个标准的最终权重。CRITIC方法通过以下公式来综合权重：



⑤归一化权重：由于CRITIC方法计算出的权重可能和标准化结果不一致，需要进行归一化处理，使得所有标准的权重之和为1。归一化后的权重计算公式为：



1. 关联权重算法

设共有个数据组，为数据组与数据组的相关系数，为数据组与数据组的相关系数，





计算步骤：

①计算数据组之间的相关系数：计算数据组两两之间的相关系数

②计算权重系数

③计算关联权重

④将关联权重作为求和权重计算项目健康度

1. 更多实现细节

更多实现细节见项目代码的注释