**开源软件基础大作业**

**分析GitHub上热门项目的提交情况**

学 院（系）： 软件学院

专 业： 软件工程

组 长 姓 名： 高康博

学 号： 20212241442

组 员 姓 名： 刘扬河

学 号： 20212241178

评 阅 教 师： 任志磊

完 成 日 期： 2024/1/29

大连理工大学

Dalian University of Technolog

目录

[1 需求分析 3](#_Toc157446939)

[1.1 项目概述 3](#_Toc157446940)

[1.2 功能需求 3](#_Toc157446941)

[2 设计程序 5](#_Toc157446942)

[2.1 技术选择 5](#_Toc157446943)

[2.2 模块设计 5](#_Toc157446944)

[3 实现程序 6](#_Toc157446945)

[3.1 github\_api模块 6](#_Toc157446946)

[3.2 data\_processing模块 7](#_Toc157446947)

[3.3 main模块： 10](#_Toc157446948)

[4 测试： 11](#_Toc157446949)

[4.1 github\_api模块测试： 11](#_Toc157446950)

[4.2 data\_processing模块测试 12](#_Toc157446951)

[5 小组总结 14](#_Toc157446952)

# 需求分析

## 项目概述

### 项目名称：分析GitHub上热门开源项目spotube的提交情况

### 目标：分析GitHub上热门开源项目spotube的提交情况，通过统计图、折线图和词云来可视化展示相关信息。

## 功能需求

### 数据获取：通过GitHub上的个人令牌获取该项目的提交情况，从而获得数据

### 数据分析：对获得的大量数据进行数据清洗，挑选出相关数据，例如提交次数、提交日期、提交类型等等。再对提取出来的数据进行函数处理。

### 可视化展示：利用python库中的Matplotlib、WordCloud等，通过输入数据来绘图，进行可视化展示。根据提交次数和提交类型生成统计图，根据提交日期来生成提交趋势的折线图和根据提交信息来生成词云。

# 设计程序

## 技术选择

### 编程语言：python

### 数据获取：使用GitHub API来获取相关数据

### 数据清洗：利用JSON在线解释工具来解释所获取的数据包括什么，提取相关有用的数据。

### 数据分析：通过上网查询来弄清楚JSON数据中的键值对分别代表什么，利用变量来进行封装有用数据。

### 可视化展示：利用Matplotlib、WordCloud等python库来可视化展示相关数据，来生成条状统计图、折线图和词云。

## 模块设计

### github\_api模块：此模块用来和GitHub进行交互，进行数据获取工作，需要将交互用到的数据，比如个人令牌、仓库名和仓库拥有者等信息在本模块中存储。

### data\_processing模块：此模块用来进行数据处理和可视化展示，在此模块中需要实现四个函数分别对应四个图表，例如plot\_commit\_counts、plot\_commit\_trend、generate\_wordcloud和commit\_type。

### main模块：本模块是程序的主入口，首先通过调用github\_api模块来进行和GitHub交互来获取数据,再调用data\_processing模块来进行数据处理和可视化展示。

# 实现程序

## github\_api模块

### 程序代码：

1. import requests
2. import json
3. def get\_commit\_data(repo, owner, token):
4. url = f"https://api.github.com/repos/{owner}/{repo}/commits"
5. headers = {'Authorization': f'token {token}'}
6. response = requests.get(url, headers=headers)
7. if response.status\_code == 200:
8. return json.loads(response.text)
9. else:
10. print("Error:", response.status\_code)
11. return []

### 分析代码：

首先定义函数get\_commit\_data(repo,owner,token)，来进行和GitHub交互，需要接收仓库名（repo）、仓库拥有者（owner）和令牌（token），通过发送请求request,其中包括url和令牌，如果回复的状态码为200，则接收成功，返回json的数据格式。

## data\_processing模块

### 程序代码：

1. def plot*\_commit\_*counts(data):
2. authors = [commit['commit']['author']['name'] for commit in data]
3. # 使用Counter来统计每个作者的提交次数
4. author*\_commit\_*counts = Counter(authors)
5. # 创建图表
6. plt.bar(author*\_commit\_*counts.keys(), author*\_commit\_*counts.values())
7. plt.xlabel('Author')
8. plt.ylabel('Number of Commits')
9. plt.title('Commit Counts by Author')
10. plt.xticks(rotation=45)  # 旋转x轴标签，使其更容易阅读
11. plt.show()

### 分析代码：

函数plot\_commit\_counts(data)接收json数据data，并且遍历data中的commit项，使用Counter函数来获取每个作者的提交次数，并通过matplotlib来创建图表，x轴为作者名，y轴为提交次数

### 程序代码：

1. def plot\_commit\_trend(data):
2. date\_commit\_counts = defaultdict(int)
3. for commit in data:
4. commit\_date = commit['commit']['author']['date']
5. *# 将日期字符串转换为日期对象*
6. date\_obj = datetime.fromisoformat(commit\_date)
7. *# 只保留年月日*
8. simple\_date = date\_obj.date()
9. date\_commit\_counts[simple\_date] += 1
10. *# 对日期进行排序*
11. sorted\_dates = sorted(date\_commit\_counts.keys())
12. *# 准备绘图数据*
13. dates = []
14. counts = []
15. for date in sorted\_dates:
16. dates.append(date)
17. counts.append(date\_commit\_counts[date])
18. *# 绘制趋势图*
19. plt.figure(figsize=(10, 5))
20. plt.plot(dates, counts, marker='o')
21. plt.xlabel('Date')
22. plt.ylabel('Number of Commits')
23. plt.title('Commit Trend Over Time')
24. plt.xticks(rotation=45)
25. plt.tight\_layout()
26. plt.show()

### 分析代码：

函数plot\_commit\_tread(data)接收json数据data，提取data中有关作者和日期的数据，并且利用sort函数来对日期排序，利用matplotlib来生成折线图，也就是提交趋势图。

### 程序代码：

1. def generate\_wordcloud(data):
2. commit\_messages = [commit['commit']['message'] for commit in data]
3. *# 将 commit messages 转换为文本*
4. text = ' '.join(commit\_messages)
5. *# 生成词云*
6. wordcloud = WordCloud(width=800, height=400, background\_color='white').generate(text)
7. *# 显示词云图*
8. plt.figure(figsize=(10, 5))
9. plt.imshow(wordcloud, interpolation='bilinear')
10. plt.axis('off')
11. plt.title('Commit Messages Word Cloud')
12. plt.show()

### 分析代码：

首先将data中的commit项中的message提取出来，将其转化为文本格式，利用WordCloud生成词云，再通过matplotlib来进行绘图，显示词云图。

### 程序代码：

1. def commit\_type(data):
2. *# 提取提交消息*
3. messages = [commit["commit"]["message"].lower() for commit in data]
4. *# 定义不同类型的提交及其关键词*
5. commit\_types = {
6. "feature": ["feature", "add", "new"],
7. "bugfix": ["fix", "bugfix", "issue"],
8. "docs": ["docs", "documentation", "readme"],
9. *# 您可以根据需要添加更多类型*
10. }
11. *# 分类计数*
12. type\_counts = Counter()
13. for message in messages:
14. for commit\_type, keywords in commit\_types.items():
15. if any(keyword in message for keyword in keywords):
16. type\_counts[commit\_type] += 1
17. break
18. else:
19. type\_counts["other"] += 1  *# 对于不匹配任何关键词的提交，归类为"other"*
20. *# 转换为Pandas DataFrame以便于可视化*
21. df = pd.DataFrame(type\_counts.items(), columns=["Type", "Count"])
22. *# 绘制条形图*
23. df.plot(kind="bar", x="Type", y="Count", legend=False, color="skyblue")
24. plt.ylabel("Number of Commits")
25. plt.title("Commit Type Analysis")
26. plt.xticks(rotation=45)
27. plt.show()

### 代码分析：

此函数用来绘制提交类型的统计图，先获取提交消息，然后定义不同的提交及其关键词，利用Counter来进行分类计数，对于什么也不匹配的归类为other，然后转换为Pandas DataFrame以便于可视化，最后绘制条形图。

## main模块：

### 程序代码

1. import github\_api
2. import data\_processing
3. def main():
4. *# 获取 GitHub 数据*
5. data = github\_api.get\_commit\_data(repo="spotube", owner="KRTirtho", token="ghp\_5lJlJfn0oKpYIdMGYrBQEVti1vN2TQ2mGI23")
6. *# 处理数据并生成图表*
7. data\_processing.plot\_commit\_counts(data)
8. data\_processing.plot\_commit\_trend(data)
9. data\_processing.generate\_wordcloud(data)
10. data\_processing.commit\_type(data)
11. *# 您可以在这里添加更多的数据处理和分析功能*
12. if \_\_name\_\_ == "\_\_main\_\_":
13. main()

### 代码分析

在main函数中，先是调用get\_commit\_data来和GitHub交互获取数据，然后用获得的数据分别调用处理函数来处理数据并生成图表。

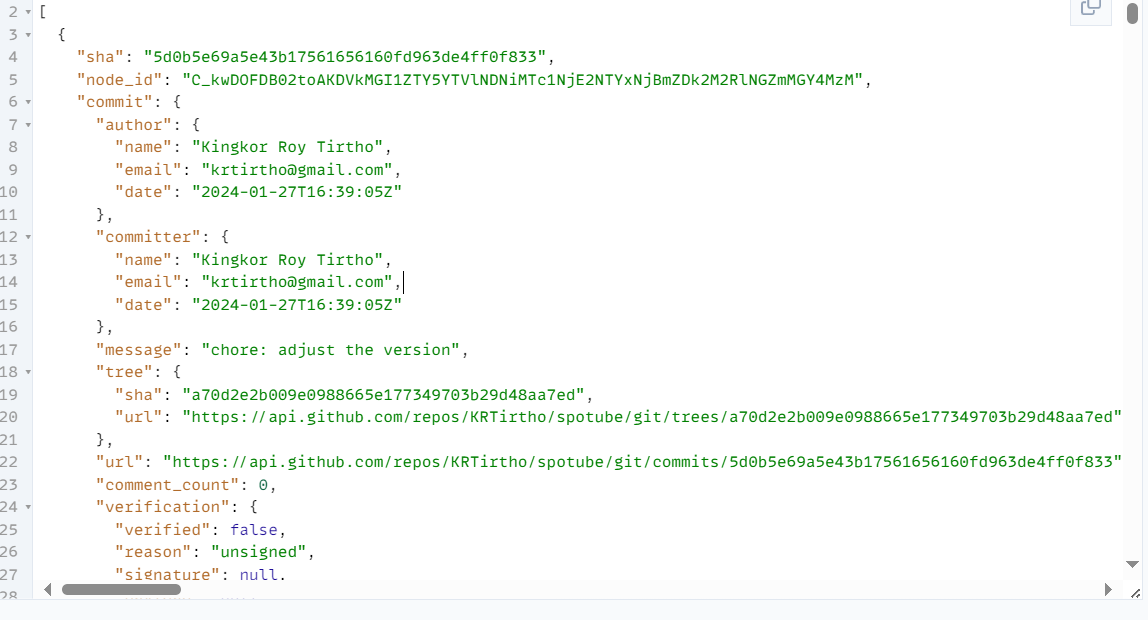
# 测试：

## github\_api模块测试：

### 获取数据：

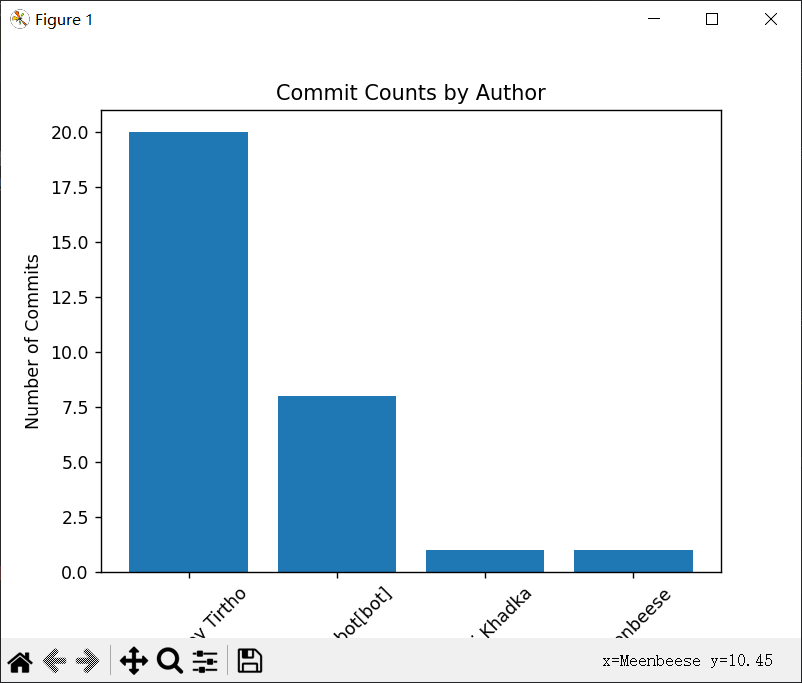
1. *# 使用 json.dumps 对 JSON 数据进行格式化*
2. formatted\_data = json.dumps(data, indent=2)
3. *# 打印格式化后的 JSON 数据*
4. print(formatted\_data)

在main模块中添加如上代码，可以打印接收的JSON数据，如下：

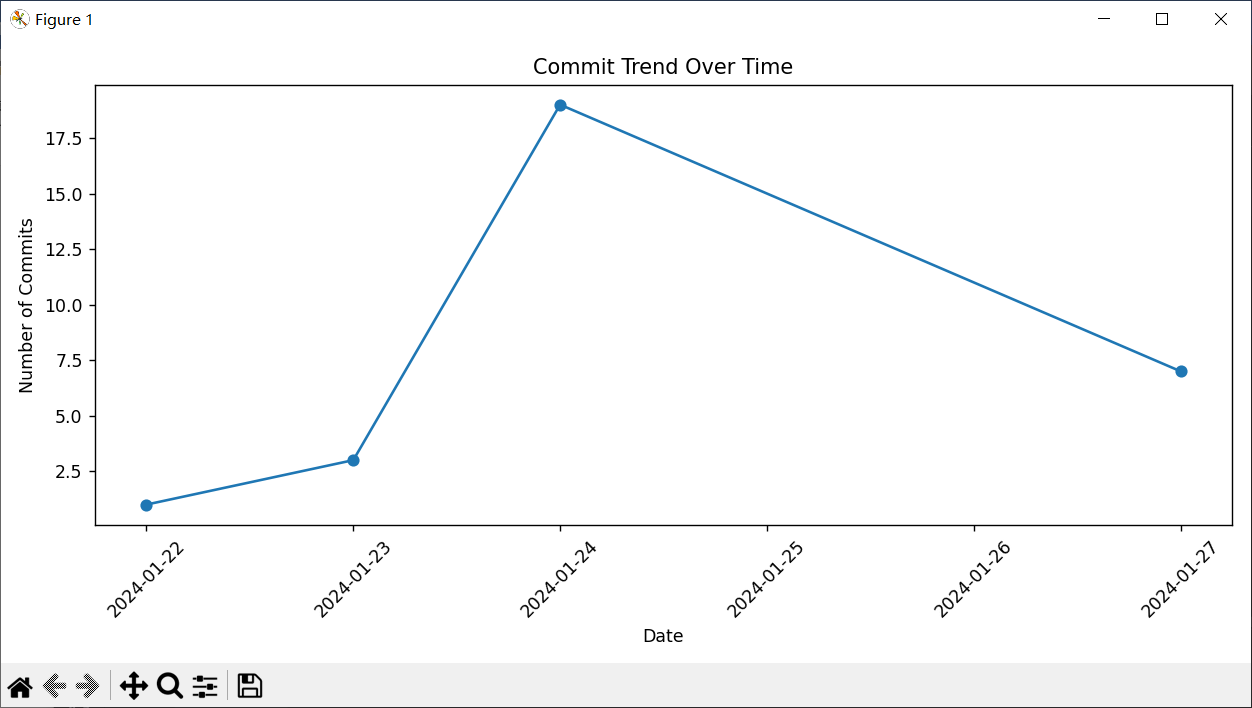


## data\_processing模块测试

### 生成提交次数统计图



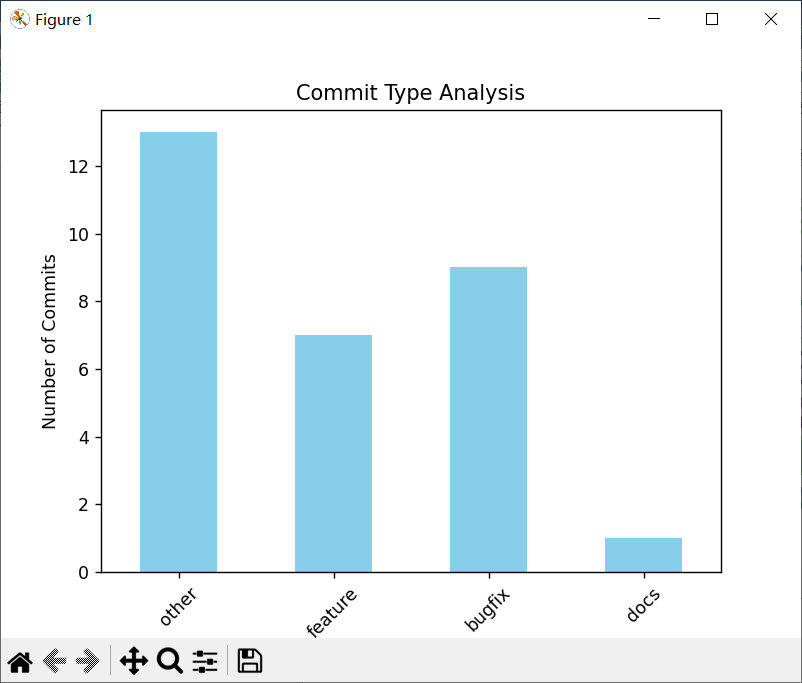
### 生成提交趋势图



### 生成词云



### 生成提交类型统计图



# 小组总结

编写代码由高康博负责，写报告由刘扬河负责，贡献度都为50%，两人共同查阅相关资料，学习相关技巧。