

Практическая работа 1. Начало работы с API

Цель работы: изучение работы с API для сбора и анализа данных, связанных с большими данными, с использованием Python.

Необходимое ПО: Python 3.x, Библиотеки: requests, pandas, matplotlib, seaborn

Задача

Проанализировать активность пользователей и репозиторий на GitHub, связанных с большими данными, с использованием языка Python.

Шаги решения задачи

1. Регистрация и получение токена доступа.

Зарегистрируйтесь на GitHub и создайте персональный токен доступа для работы с API. Сгенерировать персональный токен на 7 дней

<https://github.com/settings/tokens/new>

2. **Настройка окружения:** убедитесь, что у вас установлены необходимые библиотеки: requests, pandas, matplotlib, seaborn.

3. Подключение к GitHub API:

```
import requests
GITHUB_API_URL = "https://api.github.com"
ACCESS_TOKEN = "ваш_токен_доступа"
headers = {
    "Authorization": f"token {ACCESS_TOKEN}"
}
```

4. **Получение данных о репозиториях:** запросите информацию о репозиториях, используя определенный поисковый запрос, например, big data.

```
query = "big data"
response = requests.get(f"{GITHUB_API_URL}/search/repositories", params={"q":
query}, headers=headers)
data = response.json()
repos = data['items']
```

5. **Анализ данных.** извлеките и проанализируйте данные, такие как количество звезд, форков, количество открытых issues и т.д.

```
import pandas as pd
repo_data = pd.DataFrame(repos, columns=['name', 'stargazers_count',
'forks_count', 'open_issues_count'])
print(repo_data.describe())
```

6. **Визуализация данных:** постройте графики для визуализации активности репозитория.

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.barplot(x='name', y='stargazers_count', data=repo_data.head(10))
plt.xticks(rotation=45)
plt.title('Top 10 Repositories by Stars')
plt.show()
```

7. **Анализ активности пользователей:** проанализируйте активность пользователей, например, по количеству коммитов или участия в pull requests. Для этого можно получить данные о коммитах или pull requests для каждого репозитория.

```
repo_name = repos[0]['full_name']
commit_response =
requests.get(f"{GITHUB_API_URL}/repos/{repo_name}/commits", headers=headers)
commits = commit_response.json()
print(f"Number of commits in {repo_name}: {len(commits)}")
```

8. **Отчет и выводы:** напишите отчет, в котором представьте результаты анализа и сделайте выводы о текущих трендах в разработке проектов, связанных с большими данными. Предоставить код на Python и краткий отчет с графическим представлением результатов анализа.

Варианты заданий

Работа с Kaggle API

1. **Анализ популярных датасетов по большим данным:** Изучение скачиваний и использования датасетов.
2. **Темы конкурсов по большим данным:** Анализ тематики конкурсов, связанных с большими данными.
3. **Анализ участников конкурсов:** Географический анализ участников конкурсов по большим данным.
4. **Тренды в больших данных:** Изучение изменений в темах конкурсов и датасетов.
5. **Анализ инструментов:** Исследование популярных инструментов для работы с большими данными на Kaggle.
6. **Взаимодействие сообщества:** Анализ комментариев и обсуждений, связанных с большими данными.
7. **История конкурсов:** Анализ динамики проведения конкурсов по большим данным.
8. **Победители конкурсов:** Изучение профилей победителей конкурсов по большим данным.
9. **Анализ тегов:** Популярные теги, связанные с большими данными.
10. **Визуализация данных:** Изучение подходов к визуализации больших данных.

Работа с GitHub API

11. **Популярные репозитории по большим данным:** Анализ репозиторий и их тематики.
12. **Языки программирования:** Изучение языков, используемых в проектах по большим данным.
13. **Активность разработчиков:** Анализ активности разработчиков в проектах по большим данным.
14. **Изучение форков:** Анализ репозиторий с наибольшим количеством форков.
15. **Pull requests:** Анализ pull requests в репозиториях по большим данным.
16. **География разработчиков:** Изучение географического распределения разработчиков.
17. **Анализ issues:** Изучение открытых и закрытых issues в проектах.
18. **Социальная активность:** Анализ обсуждений в репозиториях по большим данным.
19. **История создания проектов:** Анализ трендов создания проектов по большим данным.
20. **Лицензии проектов:** Изучение типов лицензий, используемых в проектах по большим данным.

21. **Анализ README файлов:** Исследование содержания и структуры README в репозиториях по большим данным.
22. **Тренды в больших данных:** Определение трендов на основе анализа обновлений репозиторияев.
23. **Сравнение популярных библиотек:** Исследование популярности и использования библиотек для работы с большими данными.
24. **Анализ коммитов:** Изучение активности коммитов в проектах по большим данным.
25. **Влияние крупных корпораций:** Анализ вклада крупных корпораций в проекты по большим данным.