19ZO02-Social and Economic Network Analysis

Project Report

Team Members:

19Z210 DEEPTI RAVI KUMAR

19Z211 DIVYA DARSHINI R

19Z216 HARINI S

19Z230 PRETHIKA P

19Z265 SWATHI PRIYA M

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE AND ENGINEERING

Of Anna University

Problem Statement:

Version control and source code management are some of the most striking features of GitHub. Programmers use this site to network and exchange ideas. Employers and recruiters use it to evaluate software developers. In addition to gaining valuable insights into user behavior, the repositories on GitHub can also provide us with useful information on the current technologies that developers are using today and the technologies that make a repository popular.

Dataset Description:

It is derived from scraping top-starred GitHub repositories covering a wide range of topics. A Python library called BeautifulSoup was used to scrape the data. Its primary purpose is to analyze the most popular repositories on GitHub. Data-Science, Machine-Learning, Computer-Vision, etc, are among the topics covered in the dataset, along with repository commits, issues, forks, etc.

Tools used:

1. Gephi:
   A data visualization software package that is open-source and can be used to analyze and visualize networks.
2. NetworkX:
   To plot and understand the basic network graph
3. Pandas:
   A Statistical analysis and data manipulation software written for Python.
4. Matplotlib:
   Visualizations in Python can be created static, animated, and interactively using this library.
5. Seaborn.:
   Matplotlib-based Python data visualization library
6. AST:
   Represents the source code as a tree that conveys the structure of the source code.
7. Wordcloud:
   It is a visual representation of word frequency in a text where the size represents the frequency of each word.

Challenges Faced:

1. To find our topic, we combed the web. We took our time to study the subject and decide on the topic since there was a wide range of concepts available.
2. Our decision on how to perform the analysis took some time.
3. During implementation, we encountered some challenges with respect to the code, dataset, and collection process.

Contribution of Team Members:

Roll no	Name	Contribution
19Z210	Deepti Ravi Kumar	Understand contribution activities across the repositories
19Z211	Divya Darshini R	Read, clean, and structure data to make it suitable for analysis
19Z216	Harini S	Analysis of topic tags
19Z230	Preethika P	Analyze top repositories based on popularity
19Z265	Swathi Priya M	Analyze top repositories based on popularity

Annexure I: Code

Read, clean and structure data to make it suitable for analysis:

```
from google.colab import drive
drive.mount('/content/drive')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
github_data_df = pd.read_csv('/content/drive/MyDrive/Sena_Project/Github_data.csv')
github_data_df.head()
github_data_df = github_data_df.drop(['Unnamed: 0','Unnamed: 0.1'],axis=1)
github_df =
github_data_df[['topic','name','user','star','fork','watch','issue','pull_requests','topic_tag','commits','contributers']]
new_names = ['Topic','Repo_Name','User_Name','Star','Fork','Watch','Issues','Pull_Requests',
            'Topic_Tags','Commits','Contributors']
old_names = github_df.columns
github_df = github_df.rename(columns=dict(zip(old_names, new_names)))
github_df['Star'] = github_df['Star'].apply(lambda x: float(x.rstrip('k'))*1000 if x.endswith('k') else
float(x))
github_df['Fork'] = github_df['Fork'].apply(lambda x: float(x.rstrip('k'))*1000 if x.endswith('k') else
float(x))
github_df['Watch'] = github_df['Watch'].apply(lambda x: float(x.rstrip('k'))*1000 if 'k' in x else float(x))
cols = ['Issues','Pull_Requests','Commits','Contributors']
github_df[cols] = github_df[cols].apply(pd.to_numeric, errors='coerce', axis=1)
```

Analysis 1: Top repositories based on popularity

```
pop_mean_df = github_df.groupby('Topic').mean().reset_index()
```

1.1 Analysis of stars

```
fig, ax = plt.subplots(figsize=(6,4), dpi=100)
plt.rcParams['axes.edgecolor']='#333F4B'
```

```
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.tick_params(axis='both', which='both', labelsize=10, bottom=True, left=False)
ax.set_xlim(0,45000)
ax.grid(False)
ax.set_facecolor('white')
sns.barplot(data=pop_mean_df, x='Star', y='Topic');
ax.set_xlabel('Stars', fontsize=13, color = '#333F4B')
ax.set_ylabel('Topic', fontsize=13, color = '#333F4B')
fig.suptitle('Average stars on each topic',fontsize=18, color = '#333F4B');
github_df.nlargest(n=10, columns='Star')[['Repo_Name','Topic','Star']]
print('Most starred repository:')
print('Repository Name: ',github_df.iloc[github_df['Star'].idxmax()]['Repo_Name'])
print('Topic: ',github_df.iloc[github_df['Star'].idxmax()]['Topic'])
print('Star: ',github_df.iloc[github_df['Star'].idxmax()]['Star'])
```

1.2 Analysis of watch

```
fig, ax = plt.subplots(figsize=(6,4), dpi=100)
plt.rcParams['axes.edgecolor']='#333F4B'
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.tick_params(axis='both', which='both', labelsize=10, bottom=True, left=False)
ax.set_xlim(0,1600)
ax.grid(False)
ax.set_facecolor('white')
sns.barplot(data=pop_mean_df, x='Watch', y='Topic');
ax.set_xlabel('Watchers', fontsize=13, color = '#333F4B')
ax.set_ylabel('Topic', fontsize=13, color = '#333F4B')
fig.suptitle('Average watchers on each topic',fontsize=18, color = '#333F4B');
github_df.nlargest(n=10, columns='Watch')[['Repo_Name','Topic','Watch']]
print('Most watched repository:')
print('Repository Name: ',github_df.iloc[github_df['Watch'].idxmax()]['Repo_Name'])
print('Topic: ',github_df.iloc[github_df['Watch'].idxmax()]['Topic'])
print('Watch: ',github_df.iloc[github_df['Watch'].idxmax()]['Watch'])
```

1.3 Analysis of fork

```
fig, ax = plt.subplots(figsize=(6,4), dpi=100)
plt.rcParams['axes.edgecolor']='#333F4B'
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.tick_params(axis='both', which='both', labelsize=10, bottom=True, left=False)
ax.set_xlim(0,8000)
ax.grid(False)
ax.set_facecolor('white')
sns.barplot(data=pop_mean_df, x='Fork', y='Topic');
ax.set_xlabel('Forks', fontsize=13, color = '#333F4B')
ax.set_ylabel('Topic', fontsize=13, color = '#333F4B')
fig.suptitle('Average forks on each topic',fontsize=18, color = '#333F4B');
```

```
github_df.nlargest(n=10, columns='Fork')[['Repo_Name','Topic','Fork']]
print('Most forked repository:')
print('Repository Name: ',github_df.iloc[github_df['Fork'].idxmax()]['Repo_Name'])
print('Topic: ',github_df.iloc[github_df['Fork'].idxmax()]['Topic'])
print('Fork: ',github_df.iloc[github_df['Fork'].idxmax()]['Fork'])
```

Relationship between Star, Fork and Watch

```
fig, ax = plt.subplots(figsize=(8,4), dpi=100)
sns.set_theme('paper')
sns.regplot(data=github_df, x='Star', y='Fork', color='purple');
ax.set_xlabel('Star', fontsize=13, color = '#333F4B')
ax.set_ylabel('Fork', fontsize=13, color = '#333F4B')
fig.suptitle('Relationship between Star and Fork',fontsize=18, color = '#333F4B');
fig, ax = plt.subplots(figsize=(8,4), dpi=100)
sns.set_theme('paper')
sns.regplot(data=github_df, x='Watch', y='Fork', color='purple');
ax.set_xlabel('Watch', fontsize=13, color = '#333F4B')
ax.set_ylabel('Fork', fontsize=13, color = '#333F4B')
fig.suptitle('Relationship between Watch and Fork',fontsize=18, color = '#333F4B');
```

Analysis 2: Contribution activities using issues, pull requests, commits, and contributors across the repositories

```
corr_df = github_df.dropna(axis=0, subset =
['Issues','Pull_Requests','Commits','Contributors'])[['Issues','Pull_Requests','Commits','Contributors']]
fig, ax = plt.subplots(figsize=(6,4), dpi=100)
sns.heatmap(corr_df.corr(), linewidths=0.1, vmax=1.0, square=True, linecolor='white', annot=True,
cmap='winter');
fig.suptitle('Correlation between the contribution columns',fontsize=16, color = '#333F4B');
popular_df =
github_df.nlargest(n=100,columns=['Star'])[['Issues','Pull_Requests','Commits','Contributors']]
fig, ax = plt.subplots(figsize=(6,4), dpi=100)
sns.heatmap(popular_df.corr(), linewidths=0.1, vmax=1.0, square=True, linecolor='white', annot=True,
cmap='winter');
fig.suptitle('Correlation of contributions in Top 100 popular repositories',fontsize=16, color = '#333F4B');
users_with_more_repos =
github_df.groupby('User_Name').size().nlargest(n=10).reset_index(name='Count')['User_Name'].to_list()
more_repos_users_df =
github_df[github_df['User_Name'].isin(users_with_more_repos)][['Issues','Pull_Requests','Commits','Con
tributors']]
fig, ax = plt.subplots(figsize=(6,4), dpi=100)
sns.heatmap(more_repos_users_df.corr(), linewidths=0.1, vmax=1.0, square=True, linecolor='white',
annot=True, cmap='summer');
fig.suptitle('Correlation of contributions among users with more repositories',fontsize=16, color =
'#333F4B');
```

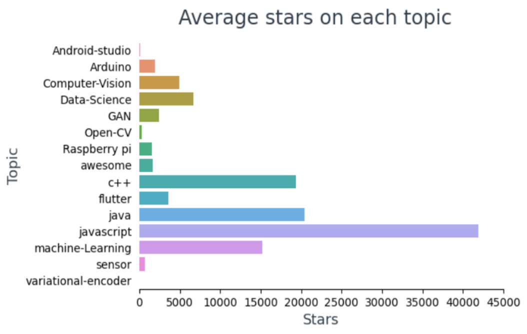
Analysis 3: Topic Tags

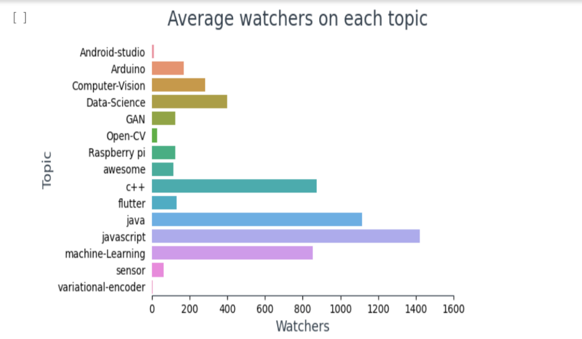
```
import ast
from collections import Counter
topic_tags = github_df['Topic_Tags'].apply(lambda x: ast.literal_eval(x)).tolist()
all_tags = [tag for topic in topic_tags for tag in topic]
```

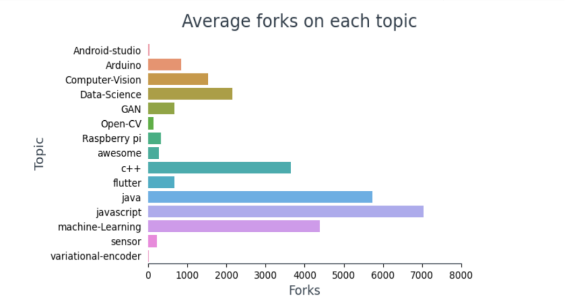
```
tags_dict = Counter(all_tags)
toptags_df = pd.DataFrame(tags_dict.most_common(15), columns=['Name of the Tag','Count'])
fig, ax = plt.subplots(figsize=(7,4), dpi=100)
plt.xticks(rotation=90)
ax.grid(False)
ax.set_facecolor('white')
sns.despine()
sns.barplot(data=toptags_df, x='Name of the Tag', y='Count', palette='twilight_shifted');
ax.set_xlabel('Topic Tags', fontsize=13, color = '#333F4B')
ax.set_ylabel('Count', fontsize=13, color = '#333F4B')
fig.suptitle('Most popular topic tags',fontsize=18, color = '#333F4B');
len_tags = [len(tag) for tag in topic_tags]
github_df['Total_Tags'] = len_tags
topic_wise_tags = github_df.groupby('Topic').sum()['Total_Tags'].reset_index(name='Total Tags')
fig, ax = plt.subplots(figsize=(7,4), dpi=100)
ax.grid(False)
ax.set_facecolor('white')
sns.despine()
sns.barplot(data=topic_wise_tags,x='Total Tags', y='Topic', ci=None, palette='gist_rainbow');
ax.set_xlabel('Total Tags', fontsize=13, color = '#333F4B')
ax.set_ylabel('Topic', fontsize=13, color = '#333F4B')
fig.suptitle('Tags distribution across topics',fontsize=18, color = '#333F4B');
```

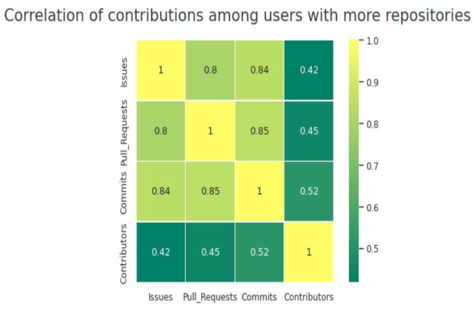
Annexure II:

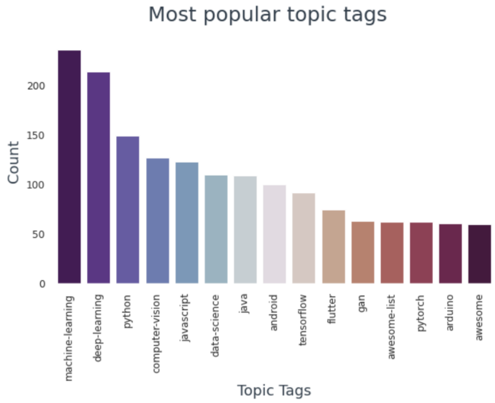
SCREENSHOTS:











REFERENCES:

1. https://www.analyticsvidhya.com/blog/2021/07/analyzing-popular-repositories
2. https://towardsdatascience.com/social-network-analysis-from-theory
3. https://gephi.org/
4. https://medium.appbase.io/analyzing-20k-github-repositories-af76de21c3fc
5. https://livablesoftware.com/tools-mine-analyze-github-git-software-data/
6. https://springerplus.springeropen.com/articles/10.1186/s40064-016-2897-7
7. https://stackoverflow.com/questions/71363111/how-to-filter-github-repositories-of-an-user-by-topics
8. https://stackoverflow.com/questions/69933548/getting-static-analysis-of-github-repositories
9. https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html
10. https://github.com/topics/social-network-analysis?l=r