GitHub Recommender System

Aman Kumar Chaursiya (2019CSB1067)
Harshwardhan Kumar (2019CSB1089)
Krithika Goyal (2019CSB1094)

22nd, November 2020

Abstract

Created a web app where any GitHub user can come and look for repositories he can contribute to based on his interest. We have also added recommendations of people he can follow based on his interest and current activities in GitHub. We collected the GitHub data using selenium package of python and then tried to find out various ways in which we can recommend the repositories based on the interest of the user. Assigned a similarity index between every two users. Also we tried to get the interest of the user by looking at the repositories of the people he follows. Based on the similarity index and the people he follows, we recommended him the repositories. The interface we made also tells why we recommended that along with the recommendations.

1 Introduction

We collected data about repositories user have and repositories he forked, we also collected data of his followers and followee. We used data of languages used in repository to calculate similarity index between 2 users and used that similarity index for recommendations. Similarly used followers data to recommend repositories and people. And finally made a web app with a personalised page rendered for every user using Django framework.

1.1 Problem

Can there be a recommendation system for GitHub like various other sites so that people can look for the repositories of there interest easily and also find people of same interest?

1.2 Literature

We didn't refer any papers but we found slides of Dr.Tanmoy Chakraborty on **Link prediction** a nice inspiration for our project. Slides were provided to CS522 students.

1.3 New idea

We are proposing a web application where GitHub users can come and find repositories that match their interest. Earlier there was no such platform where people can get such recommendations. We also help users to find other GitHub users with common interests.

2 Method

2.1 Motivation

GitHub is a very popular websites but none of our team members ever saw any recommend system of repository, which makes GitHub difficult for new users.
We decided to make a recommendation system, for users and started this project to see what are the difficulties, so we started with a few users and made a recommendation system with it.

Later we found that GitHub indeed have a recommendation system in explore tab which they added recently (we saw this November for the first time)

2.2 Data Collection

We started with collecting data from GitHub using selenium package of python. We started by finding some repositories with at least 100 contributors and then stored those users in a text file, this way we got around 2000 users. We then collected the repositories of each of those 2000 users and the users they follow. Sometimes running a code for the data took 2 days to run.

2.3 Implementation

1. After working on collecting data we started working on how we will recommend. Then in our discussions we came up with many ideas like if two users have some number of repositories in common then repositories of one user can be recommended to the other user like applying BFS on a graph made with users as nodes and an edge between them if they have some repositories in common.

2. After going through a lot of such ideas we finally converged to the following implementation details.

3. Similarity index Calculation for every two users: We picked two users from our database, let us denote first user as α and second user as β. Let $L1, L2, L3.....LN$ be the languages considered. Further let us denote Li_α and Li_β as the number of repositories of α and β respectively in Li.

$$SimilarityIndex = \sum_{i=1}^{n} \left(\frac{min(Li_\alpha, Li_\beta)}{Li_\alpha + Li_\beta} \right)$$

4. A graph was made with nodes as users and edges were decided based on the similarity index. Then for each user we picked three users with maximum similarity with the given user and connected them. We filtered out the repositories of those three users and recommended them to the concerned user. We filtered out the repositories based on the interest of the user.

5. The second major idea for recommending repositories to a GitHub user was through his/her followers. We explored all the followers of a user and filtered out their repositories based on the user's interests. Then for all the users in our database we stored the recommended repositories which were thereafter used for recommendations.

6. Recommending people of same interest: For every user we found GitHub users who were similar to him/her based on the similarity index value. We filtered out top 20 users and recommended them to the user (If they were not already in his followee list.)

7. We explored followee of the people the concerned user follows and then filtered them on the basis of the user's interest. Those users who were found to be similar were recommended to him/her.

2.4 Web Application Development

1. We made a web application using Django (python framework), HTML, CSS/Bootstrap, SQLite and JavaScript. We have also made our interface to look similar to GitHub.

2. We made 4 pages:

   (a) Enter-Page: In this page any GitHub user can come and enter his username to get details of his contributions and recommendations too.

   (b) Home-Page: In this page we plotted a pie chart conveying his contribution details in different languages.

   (c) Repo-Recommendation-Page: In this page we showed the final repositories that we recommended to him along with the information of why we recommended that repository and also the link to that repository.

   (d) Followee-Recommendation-Page: This page shows GitHub users that the user can follow. These users are selected based on the ideas we discussed above. This page contains information regarding why that user is selected and also the link to that user.

3 Results

3.1 Experiment findings

1. More than 175 types of files are used by only 2000 users.

2. There are very few (almost none) isolated contributors, ie. the contributors with no followers or followee.

3. A very few languages are contributed in a very high percentage, and Rich-gets-richer phenomenon is visible in languages and in followers data of GitHub.

3.2 Interpretation of findings

We used type of file as a measure of interest and finding similarity between 2 users.
We also used the connections of GitHub as a variable in our recommendations.

4 Conclusion

After trying so many approaches, finally this very simple approach gave us best working output. Go to the following links for final deliverable of this project

Link to YouTube video demonstration of web application: Click Here

Link to GitHub Repository of this project: Click Here

4.1 Team Work

Data-Collection was the most difficult task in this project. Sometimes it took days to store the whole data. So we divided the work of data collection between us uniformly.

We conducted frequent online Google meet sessions to discuss the idea and implementation details. We also discussed the project progress and doubt regarding the concepts were also addressed.
All team members played their roles with great team spirit to convert a vague idea into a nice project.

1. **Krithika Goyal**

    (a) **Data-Collection:** Implemented the code for collecting the users by looking into the contributors of repositories in urls.txt.
    Implemented the formula of measuring the similarity between two users (measureSimilarity.py)
    Calculated and stored the value of similarity index for every two user(values.txt), Recommended the repositories based on similarity index, Implemented the code for collecting Recommended-Repo.txt(33%), collected the follower data(userFollowingData.txt, 33%)

    (b) **Data analysis:** Plotted the graph of users as nodes and an edge between them if they have atleast one repository in common which helped us to know how much data was sufficient for the project.
    Made the graph based on similarity index(graphBasedOnSimilarity.py)

    (c) **Web Application:** Designed all the pages of the web app using HTML/CSS/Bootstrap. Made the cards for showing the recommendations of repositories and followee. For making the app look similar to GitHub, looked into chrome Dev tools of GitHub for getting the color of the nav-bar and background image GitHub uses.

    (d) **Others:** Proposed the idea of users as nodes of graph and edges as common contributing repo.
    Proposed the idea to calculate similarity index between users to recommend the repos which became the key of the project.
    Proposed the formula for the similarity index which was highly approved by all of the team members.

2. **Aman Kumar Chourasiya**

    (a) **Data Collection:** Implemented the code for finding and storing the repository type of each repo(lang.csv),
    Defined the language encoding to integers and stored it in languages.txt,
    Implemented the code for finding and storing the number of repositories each user has contributed in different languages considered which formed the basis for finding the interest of a user(userRepoTypeInfo.txt),
    Implemented the code for finding the followee of a GitHub user, collected the followee data(userFollowingData.txt, 33%).

    (b) **Web application:** Loaded the previously collected data into Django database system (sqlite3), Each of the different kind of data we collected were stored in a separated database instance(mygrs/recommender/models.py).
    Sent the data from the backend to the frontend whenever required(like sending data of recommended repositories and followee, and for pie chart generation for each user)
    Collected the data from the frontend and rendered it properly(like rendering the pie chart and providing data to the cards designed by Krithika Goyal).

    (c) **Analysis:** For each user, I made a pie chart in Django app itself, showing the number of his/her contributions in different languages.

The link to this pie chart is available in the video demonstration. This is useful as this gives a basic overview of a user's contributions in GitHub so far. Also this showed how many languages the user has contributed in currently.

(d) **Others:** Recorded the demonstration of the web application and uploaded it on YouTube. Played a key role in deciding the way for recommendation of followers and repositories to GitHub users. Like I proposed the idea of recommending repositories of one's followee after filtration which was readily approved by the team members.
I designed the data flow in the backend of the web application. Played a key role in distribution of work among the team members.

3. **Harshwardhan Kumar**

(a) **Data Collection:** Found the famous Repositories to get list of GitHub users on internet (manually) and stored them in urls.txt.
Implemented the first draft of code of finding and storing repository type in lang.csv and tokenising all languages to assign them integers.
Implemented the code for finding the users, a GitHub user follow and collected the folowee data (33 %).

(b) **Data Analysis:** Analysed and plotted the followers graph and plots on followers data and user's repo type data and stored my observations and plots in analysis.ipynb. Analysed all sorts of txt files to see what can be the best further approach to find recommendations, finally we settled in followers and repository types(languages) and discarded other ideas.

(c) **Others:** Edited the template to make this report.
Played my role in video making and editing for final demonstration.
Maintained the GitHub repository of this project by using issues, projects and other features of GitHub.
Proposed the idea of changing our final deliverable from a GUI to a web application and helped in implementation of our Django based web application which was approved by other members.
Proposed the idea and helped in implementation of using percentage of language showed in GitHub repository to be used in similarity index calculation which was later proved to be the Key of this project when krithika came with formula of similarity Index
Contributed majorly in planning the "How" of this project and with approval and valuable inputs of my team members, finalized the plan.