Analysis of Community Friendliness of Programming Languages

Samarth Tambad
Courant Institute of Mathematical Sciences
New York University
New York, NY, USA
svt258@nyu.edu

Rohit Nandwani
Courant Institute of Mathematical Sciences
New York University
New York, NY, USA
rhn235@nyu.edu

Suzanne K.McIntosh
Courant Institute of Mathematical Sciences
New York University
New York, NY, USA
mcintosh@cs.nyu.edu

***Abstract*—**

We measure the popularity and the community friendliness of programming languages and estimate the availability and demand of developers proficient in them. We perform our analysis using data from Github and StackOverflow, two of the most popular programming communities. We get ongoing projects with interactions from Github and programming questions with answers and interactions from StackOverflow. We then combine the metrics on both the platforms to provide a holistic and robust picture of the communities for the most popular programming languages.

Keywords—analytics, big data, data visualization, apache spark, programming language, github, stackoverflow, tableu

I. INTRODUCTION

Software development and maintenance is a complex activity involving many important decisions that need to be made. The choice of programming language is one such decision. From the perspective of the managers of the software projects, this decision not only affects the performance of the software but also dictates the talent pool and community support available. From the perspective of the developer, it dictates the current job opportunities and their future career trajectory.

We analyse the popularity and the community friendliness of programming languages and estimate the availability and demand of developers proficient in them. For our analysis, we look at data from Github and StackOverflow, two of the most popular programming communities.

Github is a platform for collaborative software development. Data gathered from this platform is suitable for measuring the popularity of languages and availability/demand of developers. Particularly the information available about repositories such as languages used and contributions made by developers is useful.

StackOverflow is a popular online programming Q&A community providing its participants with rapid access to knowledge and expertise of their peers. The community support is a valuable tool for developers in any programming language. Therefore, a more open, welcoming and responsive (i.e. friendly) community is a good thing to have in order to be more productive as a developer. Data such as the questions asked and the quality and time-frame of the response is a good indicator of the "friendliness" of a particular programming community.

We combine the data gathered from the two sources to compute the metrics of popularity, community friendliness, availability and demand. These metrics provide a holistic view of the pros/cons of different languages. We then use these metrics to compare different languages and help answer questions such as: which is the first language I should learn?, which language is most in demand right now?, suggest an alternative language because I work with x language but the community support is bad, etc.

The remainder of this paper is organised as follows: we describe our motivation in Section II followed by a survey of related work in Section III. In Section IV, we provide a detailed description of the datasets used. We describe our analytic in Section V followed by application design and actuation/ remediation suggestion in Section VI and Section VII respectively. In Section VIII we describe our experimental setup and analysis of results. In Section IX we provide our conclusions and provide scope for future work in Section X.

II. MOTIVATION

Open source has been gaining popularity among the developer community. Increasingly, many companies are also realising the benefit of contributing to open source projects which may benefit their business directly or indirectly [11]. Also, developers are increasingly realising the benefit of contributing to open source. Therefore, analysis on the open source developer community is good proxy for the developer community in general.

While choosing a programming language to learn or build a project, it is important to understand the characteristics and strengths of the landscape of programming languages. At the same time, it is critical to have an active and cooperative community for the programming language under consideration to speed up the learning and building process. We find that there is a lack of research on the latter aspect, which combines data from multiple available sources.

The choice of language based on community has a massive impact on the levels of productivity for the developer and the company[4], performance of the applications[5], and the overall satisfaction of the development process[6]. It will also result in increased demand for the developers in the language with better community characteristics.

There are multiple studies mapping developer productivity and satisfaction [7][8], and to the profitability of the company[7]. Programming languages can also have a major impact in the career trajectories and overall satisfaction of developers [10].

III. Related Work

Analyzing the quality of the community on StackOverflow:

Jie Yang et al[12] study the characteristics of experts on StackOverflow. They give us important metrics such as the debatableness of a question and the utility of an answer.

Seyed Mehdi Nasehi et al[13] describe what makes a good code example on StackOverflow by analyzing the interactions with code examples.

Blerina Bazelli et al[14] describe the personality traits of successful contributors on StackOverflow including extroversion and negativity.

Gupta, R. et al[15] study reopened questions on StackOverflow, and suggest the editing questions / answers even after acceptance/closing is a good sign of expertise in the community.

Wang, S et al[16] study if the population on StackOverflow can be divided into givers or takers. They also model the types of questions asked using LDA.

Analyzing the quality of the community on Github:

Ray, B. et al[18], perform a large scale study on the quality of code with respect to programming languages using text mining and regression techniques. They find that there is a significant correlation between the two.

Kalliamvakou et al[17] describe the perils on mining data on GitHub. They point that inactive account, invisible merges on pull requests, public activity on repositories could cause problems in analysis and how to overcome them.

Combining data from StackOverflow and Github:

Lee, R. K. W[19] et al, compare the developer interests on Github and StackOverflow and suggest a high correlation between the two. This helps us know the differences in proportion of contribution on the two different collaboration platforms.

Badashian, A. S et al[20] provide methods and metrics to measure core contributions, editorial activities and influence on Github and StackOverflow.

Vasilescu, B. et al[21] show how activity on StackOverflow impacts the activity on Github and vice versa.

Tian, Y et al[22] measure the quality of individual contributors on the two platforms and combine the measures across both platforms for each individual.

IV. Datasets

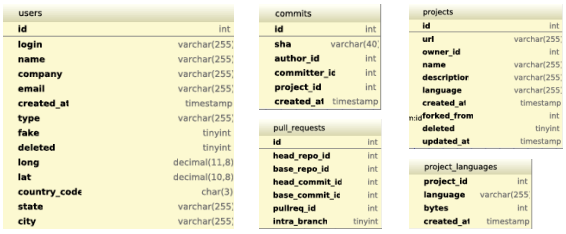
There are two main datasets used in our analysis. Here we describe the datasets in detail along with their schema.

A. Github[1]

Github is a collaborative software development platform that allows code sharing and version control. Developers can perform various activities such as creating, forking or committing to a repository, opening issues or submitting pull requests to contribute someone else's repository. The programming language used is tagged for each repository which is very helpful.

We collected the data from the GH Torrent project [2]. It is a dump of Github usage data over the period ranging from October 2013 to June 2019. The data is separated into multiple tables and stored as csv files. The complete schema is available here.

We are interested primarily in the following tables:



B. StackOverflow[2]

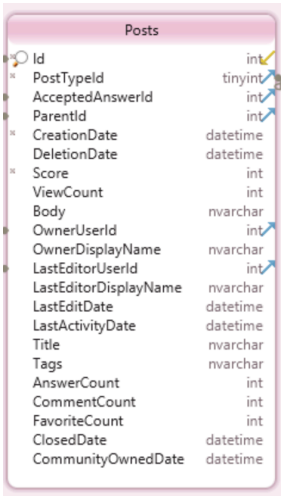
StackOverflow is the largest peer reviewed Q/A system for computer programming. All the data is open source and it is available on here. It has data from the year 2008 to 2020. We are primarily interested in the 'Posts' dataset which is all the questions, answers and the interactions with them. There are more than 15M posts with total size of 15 GB.

V. Description of Analytic

(Describe the analytic, which is the back-end of your application. What are the findings? What actionable insights does it provide?)

[1] https://ghtorrent.org/

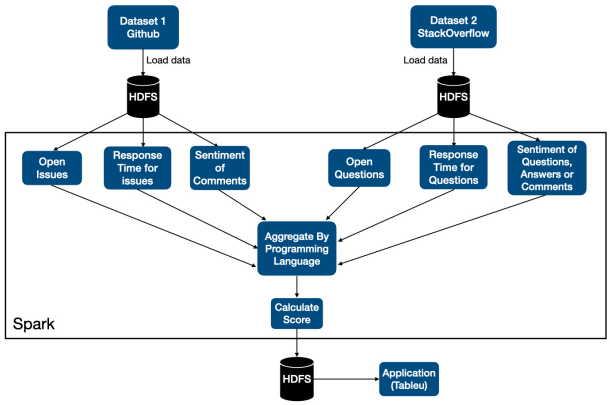
[2] https://archive.org/details/stackexchange



VI. Application Design

The overall architecture of out analytics development process is described in this section.

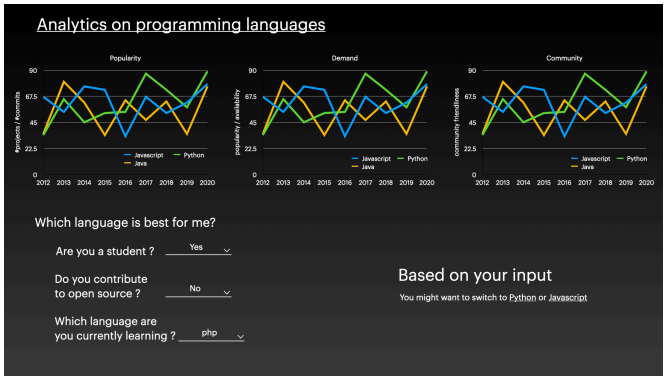
We begin by ingesting the two datasets into HDFS to be able to perform distributed processing of our big data in a scalable and efficient manner. Once we have our data ready, we perform ETL step to clean the data, remove unwanted columns and store it back. After this, we profile the data to get a rough idea about the data we are dealing with.



We then process this cleaned data to derive data relevant to our analytic. Using the Github data, we count the number of active users, repositories, commits, pull requests and pending issues grouped by programming languages spread over time (yearly). Using StackOverflow data, we count the number of users, questions, answers, unanswered questions and calculate the average response time for a question. Again, we group all this data by programming languages spread over time.

Once we have the gathered the relevant data from each dataset, we combine the data by assigning different weights to each data source. We then store this data into a database and query this data for our front-end visualisation.

The visualisation of out analytic is done in Tableu. The UI design is given below:



There are three main metrics that we track over time for each programming language. They are popularity, demand and community. Based on these metrics, we are able to help answer different types of questions about choice of a programming language.

VII. Actuation or Remediation

(Describe the actuation or remediation response to the actionable insight. This is basically the action that can be initiated in response to the actionable insight produced by the analytic - the back-end of your application.)

VIII. Analysis

(In this section, describe: Your experimental setup (tools, platforms), problems (with data, performance, tools, platforms, etc.). Describe what you learned. Discuss limitations of the application. Make recommendations for others, e.g. best practices.)

IX. Conclusion

(One paragraph about the value, results, usefulness of your application.)

X. Future Work

(Discuss possible future work for extending this project. Discuss how would you improve it, etc.)

Acknowledgment

(This section is optional. Use it to thank the people/ companies/organizations who made data available to you, for example. You can list HPC people who were particularly helpful. List Amazon if you used an Amazon voucher. Cloudera for CDH.)

References

1. Chambers, B., & Zaharia, M. (2018). Spark: The definitive guide: Big data processing made simple. " O'Reilly Media, Inc.".

2. Gousios, G. (2013, May). The GHTorent dataset and tool suite. In 2013 10th Working Conference on Mining Software Repositories (MSR) (pp. 233-236). IEEE.
3. Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2016). An in-depth study of the promises and perils of mining GitHub. *Empirical Software Engineering*, *21*(5), 2035-2071.
4. D. P. Delorey, C. D. Knutson and S. Chun, "Do Programming Languages Affect Productivity? A Case Study Using Data from Open Source Projects," First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS'07: ICSE Workshops 2007), Minneapolis, MN, 2007, pp. 8-8.
5. Amy Jo Kim. 2000. Community Building on the Web: Secret Strategies for Successful Online Communities (1st. ed.). Addison-Wesley Longman Publishing Co., Inc., USA.
6. Edwards, Kate; Weststar, Johanna; Meloni, Wanda; Pearce, Celia, & Legault, Marie-Josée (2014). Developer satisfaction survey 2014. Summary report. International Game Developers Association (IGDA).
7. Impact Of Employee Participation On Job Satisfaction, Employee Commitment And Employee Productivity Komal Khalid Bhatti* and Tahir Masood Qureshi.
8. StackOverflow Developer Survey 2019.
9. Baishakhi Ray, Daryl Posnett, Premkumar Devanbu, Vladimir Filkov
10. Communications of the ACM, October 2017, Vol. 60 No. 10, Pages 91-100. A Large-Scale Study of Programming Languages and Code Quality in Github.
11. Lerner, J., & Tirole, J. (2005). The economics of technology sharing: Open source and beyond. *Journal of Economic Perspectives*, *19*(2), 99-120.
12. Jie Yang, Ke Tao, Alessandro Bozzon, and Geert-Jan Houben Sparrows and Owls: Characterisation of Expert Behaviour in StackOverflow.
13. Seyed Mehdi Nasehi, Jonathan Sillito , Frank Maurer, and Chris Burns What Makes a Good Code Example 2012 28th IEEE International Conference on Software Maintenance (ICSM).
14. Blerina Bazelli, Abram Hindle, Eleni Stroulia On the Personality Traits of StackOverflow Users. 2013 IEEE International Conference on Software Maintenance.
15. Gupta, R., & Reddy, P. K. (2016, March). Learning from gurus: Analysis and modeling of reopened questions on stack overflow. In Proceedings of the 3rd IKDD Conference on Data Science, 2016 (pp. 1-2).
16. Wang, S., Lo, D., & Jiang, L. (2013, March). An empirical study on developer interactions in StackOverflow. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (pp. 1019-1024).
17. Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2016). An in-depth study of the promises and perils of mining GitHub. Empirical Software Engineering, 21(5), 2035-2071.
18. Ray, B., Posnett, D., Filkov, V., & Devanbu, P. (2014, November). A large scale study of programming languages and code quality in github. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (pp. 155-165).
19. Lee, R. K. W., & Lo, D. (2017, September). GitHub and Stack Overflow: Analyzing developer interests across multiple social collaborative platforms. In International Conference on Social Informatics (pp. 245-256). Springer, Cham.
20. Badashian, A. S., Esteki, A., Gholipour, A., Hindle, A., & Stroulia, E. (2014, November). Involvement, contribution and influence in GitHub and stack overflow. In CASCON (Vol. 14, pp. 19-33).
21. Vasilescu, B., Filkov, V., & Serebrenik, A. (2013, September). Stackoverflow and github: Associations between software development and crowdsourced knowledge. In 2013 International Conference on Social Computing (pp. 188-195). IEEE.
22. Tian, Y., Ng, W., Cao, J., & McIntosh, S. (2019). Geek Talents: Who are the Top Experts on GitHub and Stack Overflow?