Analyzing Trends in Open Source Software Contributions

Andrew Casner
University of Colorado, Boulder
andrew.casner@colorado.edu

Oliver Collins
University of Colorado, Boulder
oliver.collins@colorado.edu

Carl Cortright
University of Colorado, Boulder
carl.cortright@colorado.edu

Shubha Swamy
University of Colorado, Boulder
shubha.swamy@colorado.edu

1. Problem Statement

Our primary goal of this project is to gain insights about open source software. We will accomplish this goal by mining data from a dataset about open source projects. We will focus our work on four main areas: tracking trends in programming languages, analyzing how popular repositories have changed over time, how contributions to those repositories have changed over time, and also monitoring repository life cycles.

2. Literature Survey

There has been a significant body of research done on patterns in open source contributions. Every year Stack Overflow, a popular site for asking and answering coding questions, does a poll of the developers on their platform [1]. This research is used to track popular languages and libraries developers are using on the platform. Other research focuses on the potential impacts of research on open source software on the broader computer science community [2]. This research can help act as a guide as we decide what areas of open source development are important to focus on in our data mining project. The majority of the research in this area has focused specifically on Github repositories, attempting to determine the influence of any given project [3]. In our work, with the dataset we have access to, we will attempt to extend this research to all open source projects independent of where they are hosted or built. Other research by the Apache foundation has shown that the majority of open source contributions are made in a way that is not collaborative but instead driven by individual developers [4].

3. Proposed Work

Since this dataset is already in excellent condition, not much will have to be performed on the dataset. However, some necessary data cleansing will need to be completed such as scrubbing the dataset to remove null values and synchronizing time zones. Once these simple tasks are complete, our team will perform more advanced data preprocessing techniques such as matching a unique user across multiple package managers.

Once our team has appropriately cleaned our data, we will then decide which patterns answer the questions asked of the dataset. Those will be used to create visualizations, such as bar plots and graphs to show the trends found in the given data. Using the data available, we will construct a dependency graph of all of the dependencies in the entire open source ecosystem. We will then conduct an eigenvector analysis on this graph to find the most influential projects. This same analysis can then be done on language-specific sub-graphs to find the most influential projects in each language.

Finally, we will create a write up of what we have learned and possibly publish it online in a way that is consistent with the open source mentality that we are studying. It is essential to our team that what we learn about the open source community can be freely and openly shared. Our write up is also aligned with the license that the libraries.io dataset is under.

4. Dataset

The dataset we will be using for our project comes from a company called libraries.io. The purpose of libraries.io is to monitor open source projects to help developers better understand dependencies for their projects.

The dataset itself contains 311 million data points from 34 package managers and three source code repositories, including npm, GitHub, PyPi, RubyGems, Maven, Bower, and other large, language-specific package managers. With this breadth, libraries.io can track over 2.7 million unique open source packages, spanning 31 million repositories, tracking 161 million dependencies between them [5]. With this large of a dataset, we expect to have many different variables of interest that we can mine.

The dataset comes packaged in several large CSV formatted files. The main file that we will be mining is the projects database. This file contains all of the individual projects libraries.io is tracking. This file also has vital attributes for each project including language, status (active, depreciated, etc.), dependent projects count and more. Other important datasets include the dependencies CSV, which has detailed information on interdependencies between projects. Using this module we can build a dependencies graph, and mine information about which projects are most important in the open source ecosystem.

5. Evaluation Methods

We will be gathering various conclusions from our dataset. Our project focuses on four areas: tracking trends in programming languages, analyzing how popular repositories have changed over time, how contributions to those repositories have changed over time, and also tracking repository life cycles. We will need to utilize various evaluation methods to draw insights about those topics. To track trends in programming languages over time, we will analyze the frequency of popular programming languages used in various open source projects and how that has shifted over time. To analyze how popular repositories have changed over time, we will investigate how project dependencies of various open source projects have changed over time. To analyze how contributions to repositories have changed over time, we will first narrow down our data range to top contributed repositories, and then we will analyze the rate at which the contributions to those repositories have changed over time. Through this, we will also be able to conclude information about repository life cycles. By looking at trends and dependencies of various repositories over time, we will be able to conclude the repository's life cycle.

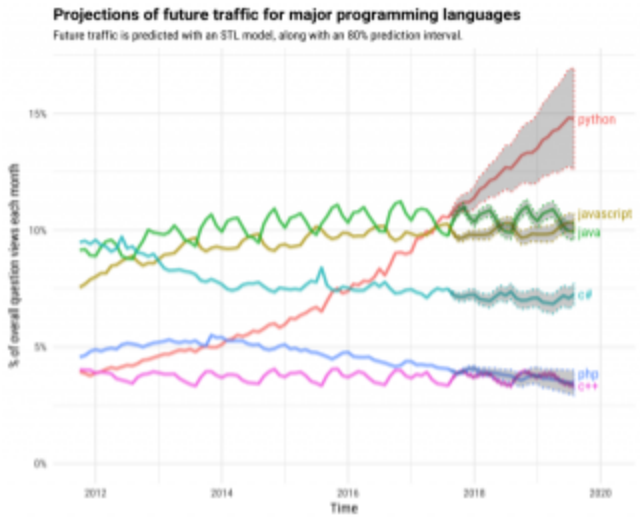


Fig 1. Predictions of future traffic for major programming languages. Graph from StackOverflow, Web. 8 Sept. 2017. n. pag.

In addition to analyzing our data in regards to the four main areas we will be focusing on, we will also compare our results to conclusions found in the Literature Survey section (2).

6. Tools

We will be using several tools to mine and analyze the dataset to reach our conclusions about open source software. The primary programming language in this project will be Python. We will be using various Python libraries to analyze, parse, and present our data. One of the tools we aim to use, Pandas, will be

used for most of our data analysis. As for our computational and statistical analysis, we will be using SciPy and NumPy. Since our team will be presenting our findings through data visualizations, we will be using a plethora of different libraries including Matplotlib, Bokeh, graph-tool, Seaborn, and several others. Moreover, lastly, for a lot of our numerical simulations, statistical modeling, data visualization, as well as a way to keep our code clean and organized we will be encapsulating our code in Jupyter Notebooks.

In addition, we will also use AWS Cloud 9 as a collaborative working environment. The Cloud 9 is a collaborative IDE that allows us to all contribute to the code base and run intensive data mining tasks on a much more powerful cloud computer.

7. Milestones

We will be watching the same timeline as outlined in the project description in regards to our milestones but we will mostly be following the milestone table our team has devised below.

No.	Milestone	Due Date
1	Proposal Presentation*	February 27th
2	Proposal Paper*	March 6th
3	Data Hosting on AWS	March 20th
4	Data Cleaning & Preprocessing	April 1st
5	Create Test Data	April 7th
6	Progress Report*	April 10th
7	Data Visualizations	April 17th
8	Data Analysis	April 17th
9	Analyzing Results	April 20th
10	Application	April 20th
11	Interactive Site with Visualizations	April 23rd
12	Final Presentation*	April 23rd
13	Final Paper*	May 1
* indicates milestones (due dates) set by professor		

7.1 Milestones Completed

Data Hosting on AWS

To collaboratively work on this project, we used a shared platform to work collaboratively on this project. We settled on using Amazon Web Services Cloud 9 as our IDE, cloud computing resource, and storage solution for our project. A significant amount of time involved figuring out and setting up the proper EC2 Instance type and mounting an EBS volume to the Instance. Once we finally loaded and unzipped the dataset to the volume on AWS, we began cleaning and preprocessing the data.

Data Cleaning and Preprocessing

We parsed and cleaned the data to ensure the quality of the dataset even though it came pre-processed. We performed several processes to further enhance the quality of our dataset including data cleaning, data integration, data transformation, and data reduction. The primary purpose of these pre-processing methods was to be confident that the dataset was clean of incomplete, noisy, and inconsistent data. The quality of data is significant when it comes to concluding the information because higher quality data yields more accurate results [6].

We used several libraries in python to help us organize and clean the data. The primary library that we used to accomplish this was Python Pandas. Pandas is a python library used for data manipulation and analysis.Since our data set consisted of millions of data points, we used the head function is Pandas to help us preview the data. Previewing various parts of

our dataset allowed us to not only understand the organization of our dataset better but also helped us determine what steps we needed to take to ensure a high-quality dataset. Using the describe function in pandas gave us a better insight into our data set. We were able to gain quick summaries of our overall dataset using the describe function in pandas for various columns. For example, we were able to conclude the average dependencies for open source projects. This conclusion was crucial for us to recognize the significant trends in our dataset. We used other functions in Pandas to delve deeper into our data. Once we were able to find unusual patterns and had a greater understanding of our data, we continued through the cleaning process. By analyzing the count for each attribute, we concluded that individual data values were missing. Missing values contribute significantly to data quality problems. It is imperative to handle these missing values to ensure our data analysis will yield high-quality results. To fix this issue, we filled in numerical data with the attribute mean and categorical data with a global constant relating to its corresponding attribute. The data comes in six different packages-- projects, versions, tags, dependencies, repositories, repository dependencies, and projects with related repository fields. Since we will be concluding all these six packages, we had to ensure that we applied this cleaning process on all six packages.

After we completed our initial step of the data preprocessing method, data cleaning, we continued the process to ensure a data set of high standard. The next process involved data integration. We handled the issue of redundancy in this process. Since all our data is from a single database, we did not encounter the problem of running into data duplicates. Our data reduction was a significant step in our data preprocessing. This process eliminated irrelevant features and reduced noise which helped us crucially since we are working with a massive dataset. In turn, this will speed up the mining and allow for more straightforward visualizations. Our goal for data reduction was to use a dataset much smaller in volume representative of our whole dataset which would produce almost the same mining results as it would for our entire data set (see section 'Create Test Data' below). Our final step of data transformation included using the Pandas library in python to perform operations such as data discretization and normalization.

Create Test Data

We extracted a smaller subset of data from our data set to create our test data set. The test data allowed us to practically run code on a sample before applying it to the original dataset with over 397 million rows of data. Even though we are running an EC2 Instance through AWS that can process through all of our data quickly and seamlessly, we still think it would be beneficial to work through a smaller dataset on our machines as it would still take a substantial time running through our EC2 Instance. Moreover, since the AWS Instance we are running on is significantly more costly than running through the dataset on our computer, it would save us not only valuable time but money as well.

7.2 Milestones Todo

Data Visualizations

By the 17th of April, we plan on producing a set of stimulating visuals that we have obtained from mining through our dataset. This milestone will allow us to not only understand our dataset but also enable us to display our findings succinctly. We plan on visualizing what our team has stated in our Problem Statement (1) including finding trends within programming languages, popularities of open source repositories over time as well as their life cycles. We also plan on visualizing additional topics we might find interesting while mining the dataset. This additional knowledge will allow us to mine the data adequately given our other domain experience.

Data Analysis

In addition to our data visualization milestone, our data analysis milestone is in place so that we can start to mine and pull real results from our dataset. While all the previous milestones were in place to set us up

to do our data analysis, we can finally begin to extract valuable knowledge from our dataset. Our team plans on utilizing many different clustering methods on the data, and across many different dimensions to identify and find undiscovered patterns about open source development.

We plan on applying most of the skills and techniques we are learning in this class. A rough exhaustive list includes clustering through k-means, k-medoids, DBSCAN; confusion matrices; contingency tables; naïve Bayesian classification; and linear, multiple, and log-linear regression models.

Analyzing Results

Similar to the previous milestone, analyzing our data is in place to allow us to examine our current results and attempt to comprehend all of our findings. While the previous milestone is our team mining the data, in analyzing the data we get to see what kinds of applications or implications that insight can provide. We will attempt to understand our findings through a few different metrics. First, we will evaluate our visualizations and find anything interesting from them. Next, from our data analysis using the tools stated from our Tools section (6), we will see if we can find anything that seems to provide any sort of information our team might find intriguing or unique. Moreover, once we have dug through our findings, we can further proceed to understand the direction our team would love to move forward with on this project.

Application

Once we complete analyzing our data, we will conclude the applications from our results. This process will help us answer our questions that we defined in Evaluation Methods (5). We initially stated that our data mining would focus on four main areas: tracking trends in programming languages, analyzing how popular repositories have changed over time, how contributions to those repositories have changed over time, and repository life cycles. We will conclude applications and apply the results we will gain through our data analysis to these topics. In our application, we will focus on how our conclusions will help us make better and more useful decisions regarding open source projects. Open source projects play a significant role in the development of software. Tens of thousands of open source projects run worldwide, and millions of users rely on open source software [2]. Concluding applications based on our data analysis will help us better understand open source software that influences millions across the world.

Interactive Site With Visuals

While the course does not require this milestone, we think it would be a great way to showcase the data we have mined. We plan on hosting a website through GitHub pages, as a way to view and interact with the findings our team has mined. We aim to provide all our visuals, analyses we encountered, and how we mined and found all of our findings on our site so others can try and explore and understand how we came to this point. Creating this website is an essential step for us as we want to show off all the hard work we put into this project, as well as allow others to use the knowledge and data we mined to hopefully better the Open Source Software Community.

Final Paper

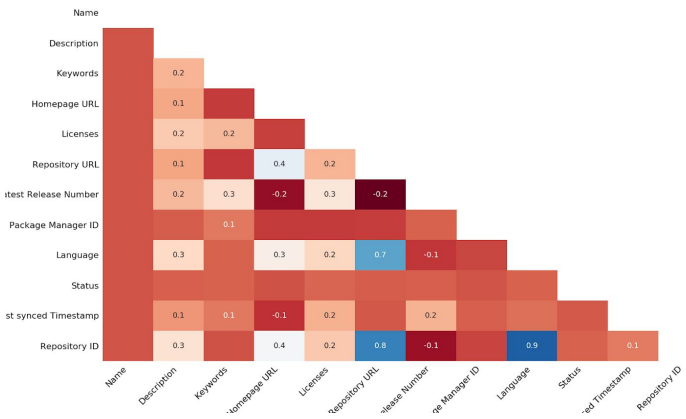
While our course requires this final paper, we still think it is a great way to portray the culmination of work our team has put into the project over the past few months. The final paper will be in the ACM SIG paper format with 11 point font and 1.1 line spacing just as all of our other progress reports are and will include our abstract, all of our related works and findings, as well as the tools we used. Moreover, if our findings are significant, we plan on releasing our paper for others to learn from and extend on if desired.

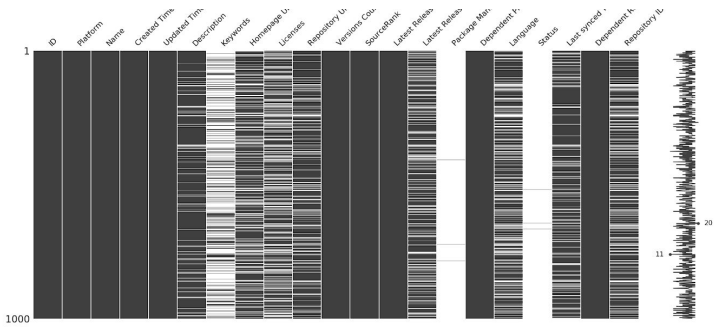
8. Results So Far

Most of the results we have compiled so far have been pretty rudimentary, so we can get a better grasp of how the AWS EC2 Instance operates. Once we dive

deeper into our project, we will be able to produce much more meaningful visualizations.

At this moment we have a few graphs that parse through the entire dataset and provide us with a cluster map and a frequency table that hopefully visualizes the data in a meaningful way. Most of the other visualizations we have pulled so far have either provided no visually appealing data or did not correlate attributes. As the week's progress we are planning to add more and more visualizations that speak more about the data and any correlations we find.





9. References

[1] “Stack Overflow Developer Survey 2017.” Stack Overflow, insights.stackoverflow.com/survey/2017.

[2] Scacchi, Walt. “The Future of Research in Free/Open Source Software Development .” 2010.

[3] Hu, Yan, et al. “Influence Analysis of Github Repositories.” SpringerPlus, Springer International Publishing, 5 Aug. 2016, www.ncbi.nlm.nih.gov/pmc/articles/PMC4975729/.

[4] Chełkowski, Tadeusz, et al. “Inequalities in Open Source Software Development: Analysis of Contributor's Commits in Apache Software Foundation Projects.” PLOS ONE, Public Library of Science, 20 Apr. 2016, journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0152976.

[5] libraries.io/data.

[6] David Hand, Heikki Mannila and Padhraic Smyth. 2001. “Principles of Data Mining”, (34-37). MIT Press.