Analyzing Open Source GitHub Repositories Towards Technology Acceptance Model

Dhruvil Gandhi
Seidenberg School of Computer Science and Information Systems
Pace University, New York, USA
Email: dgandhi@pace.edu

***Abstract*—Open Source is a gift and effort of the technology community, over years, creating and supporting platforms, projects, etc. This paper analyzes data from GitHub public repositories available on Google's BigQuery and observes relations, trends and anomalies. To set a baseline, public repositories for 22 different languages are analyzed, correlations, anomalies and trends are studied.**

***Index Terms*—Technology Acceptance Model, Data Analytics, Time Series Anomalies, Data Exploration**

I. Introduction

Open Source allows community from novice to advanced, everyone to collaborate and contribute to projects, platforms, programming languages and frameworks. Over time the advancement of languages and platforms; languages from Assembly Language to C to Python to TypeScript and Kotlin, for platforms - from Batch systems to multiprocessing and modern distributed systems. With the evolution of services and platforms, IaaS, Saas and PaaS, lead to the growth of open-source community and projects overtime.

Various studies have been conducted overtime to study the impact and growth of these. This study [1] studies the influence of software by studying GitHub. Evaluation [2] and [3] analysis observes and introduces Technology Acceptance Model, and develops a hypothesis for it. Another such study [4] was conducted to study UTAUT (Unified Theory of Acceptance and Use of Technology) [4] where a study was done on 78 undergraduate students for acceptance and impact of GitHub on their education. Various studies [5] [6] have been performed on commit messages, coding style, pull requests, coding times and other factors available for the dataset from GitHub [7]

The trends of open source repositories on GitHub was studied and trends were established and anomalies were studied using ELK stack and BigQuery. Initially, correlation of trends and data with StackOverflow, a technology Q/A platform was proposed, which however is moved as next steps and future actions for this study.

Methodology section talks about the technology used, dataset and analysis proposed. System and Experiment sections elaborate on the system setup used for the study and present the steps, algorithms, and data analytics methods used in the study. Results, dataset snippets, small data frames, and anomalies are discussed in the observation and result section. The paper ends with a preliminary conclusion and states the next steps and opportunities for analysis.

II. Methodology

A public dataset of nearly 3.8 million GitHub open source repositories [8] is available on Google Cloud Platform's BigQuery service. This dataset was used in the study. BigQuery is Google's columnar storage service for high performance and high throughput needs.

To store and analyze data, ElasticSearch and Kibana were used. ElasticSearch is based on Lucene which is a full-text search based search engine. ElasticSearch has advantage of indexing JSON documents without reindexing, only adding or updating the new or updated documents. To analyze and visualize, Kibana was used, Kibana is a web-based UI for ElasticSearch, which provides analysis and other services for ElasticSearch.

To perform the study, data about repositories for following languages was loaded in ElasticSearch from BigQuery, using Python. The repositories for which the repositories were loaded are highlighted in table I. The fields that were loaded from the BigQuery dataset are as in table II.

Language Name	Total Repositories
C	5,451,310
Python	878,748
Go	734,652
C++	689,476
Java	626,740
Dockerfile	264,042
Objective-C	144,206
Haskell	52,430
Clojure	43,686
Rust	34,519
R	29,813
Jupyter Notebook	26,891
Erlang	25,694
TypeScript	14,372
Julia	13,529
Kotlin	12,915
Swift	9,379
Objective-C++	60
Vue	38
SQL	20

TABLE I
LANGUAGES AND REPOSITORIES

Field Name	Data Type
Repository Name	String
Language Name	String
License Name	String
Timestamp of creation	Epoch Time in seconds
Year of creation	Integer
Month of creation	Integer

TABLE II
SCHEMA OF DATA USED

To analyze, word cloud, average repositories overtime for all languages, different visualizations, time-series analysis and timeseries anomaly detection and forecast were carried out. The result is described in section V and additional materials supplied with the paper.

One limitation while conducting the study is limit of availability of data on percentage of language for a particular repository, for example a simple webpage might have HTML and CSS, but it does not have data on what percent of each language is present in a particular repository.

III. SYSTEM AND EXPERIMENT

The project was first setup to load data into Azure cloud, but due to network limit because of the tier limit and different networks/datacenters, the data load was taking a long time. To speed up data loading, virtual machines that hosted ELK stack or ElasticSearch, LogStash and Kibana, was set up on Google Cloud Platform's Virtual Machine service.

As laid out in the diagram, the data for languages of different public repositories as specified in table I was loaded into BigQuery. This was done to take subset of data and not cross processing threshold of the GCP. Then using BigQuery's API and Python script, data as laid out in table II was loaded into ElasticSearch's index. All the fields were set to searchable to enable them for analysis. This helped in analyzing on different factors. Once the data was loaded, different visualizations were made as seen in section IV, and a time-series anomaly machine learning [9] job was started using Kibana for ElasticSearch. The analysis was done for repositories from 2009 to 2019. All the services, ElasticSearch, Kibana and Python scripts are currently hosted on 2.5GiB RAM, 4 vCPUs virtual instance on GCP.

After anomalies were highlighted, causes were searched on internet for possible causes. For three instances, for three different languages, they were linked with a conference or a major version release. All the analysis, the code used and the reports obtained are stored on GitHub.

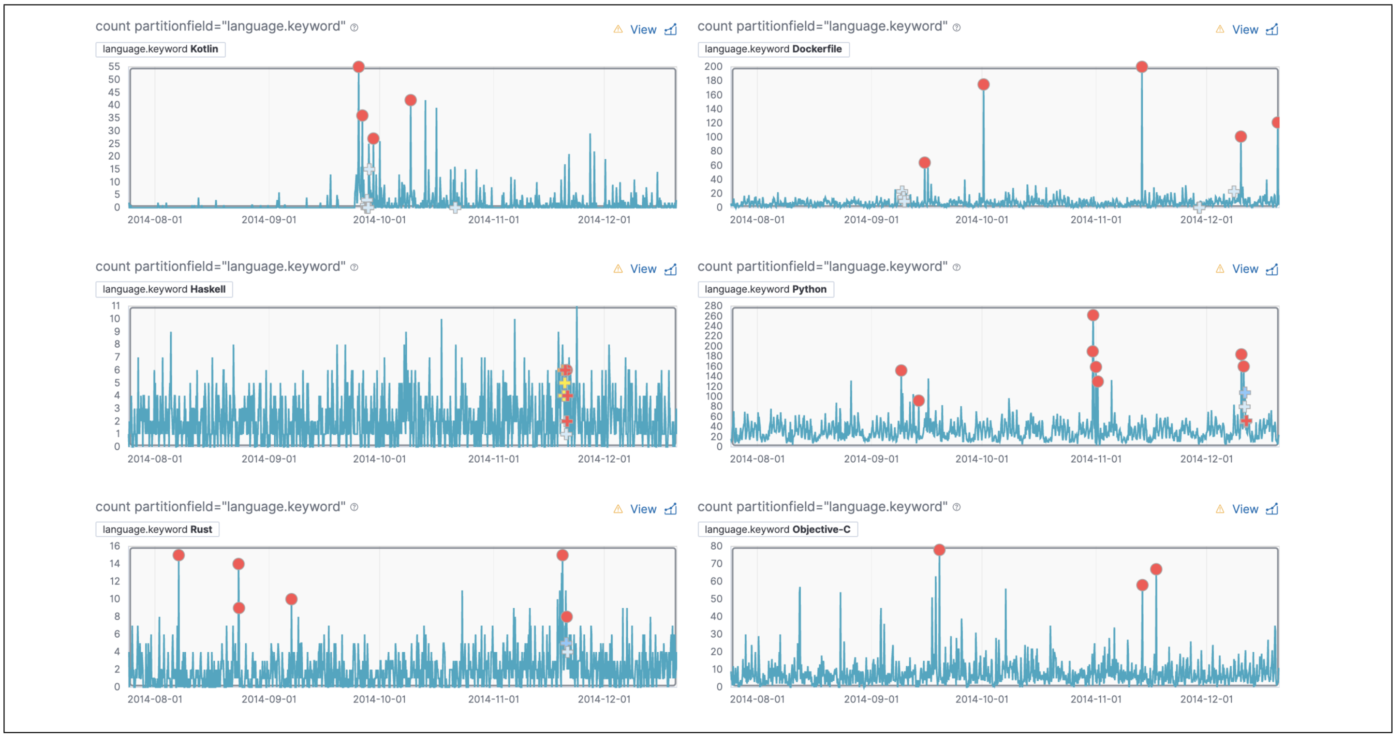
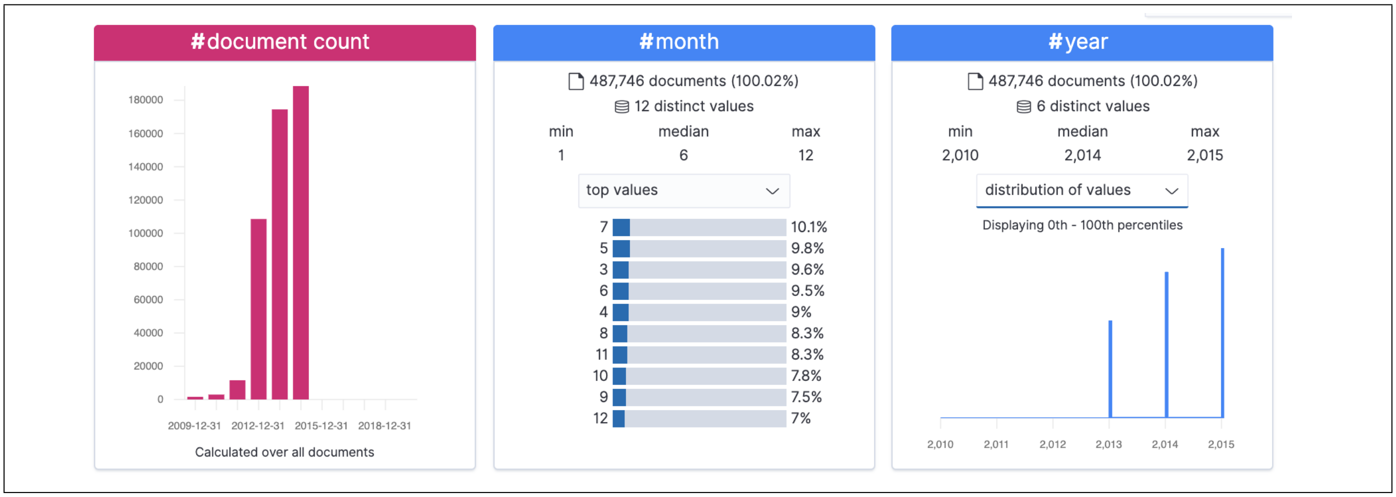


Fig. 1: Language Word Cloud by frequency



Calculated over all documents

487,746 documents (100.02%)
12 distinct values

min	median	max
1	6	12

top values

7	10.1%
5	9.8%
3	9.6%
6	9.5%
4	9%
8	8.3%
11	8.3%
10	7.8%
9	7.5%
12	7%

487,746 documents (100.02%)
6 distinct values

min	median	max
2,010	2,014	2,015

distribution of values

Displaying 0th - 100th percentiles

Fig. 2: Go language launch year stats

IV. Observation and Result

Figure 1 shows anomalies of 6 languages in red, when compared to news or significant event, it maps around the month of occurence of that event. For instance, Kotlin has a high spike, which when searched online, maps to beginning of involvement of a company called Instil, which consults and trains for Kotlin. Another example is for Python, we see a surge around 2014, a spike is observed which can be related to the announcement of Python 2.7 support end date which is end of 2020.

Figure 2 shows stats from 2015, which can be mapped to GopherCon in July, which is the reason for spiked activity in 7th month or July in 2015.

Basic statistics about the data used in analyzing all the repositories is presented in table I and II. Figure 5 shows the timestamp range of all the repositories loaded.

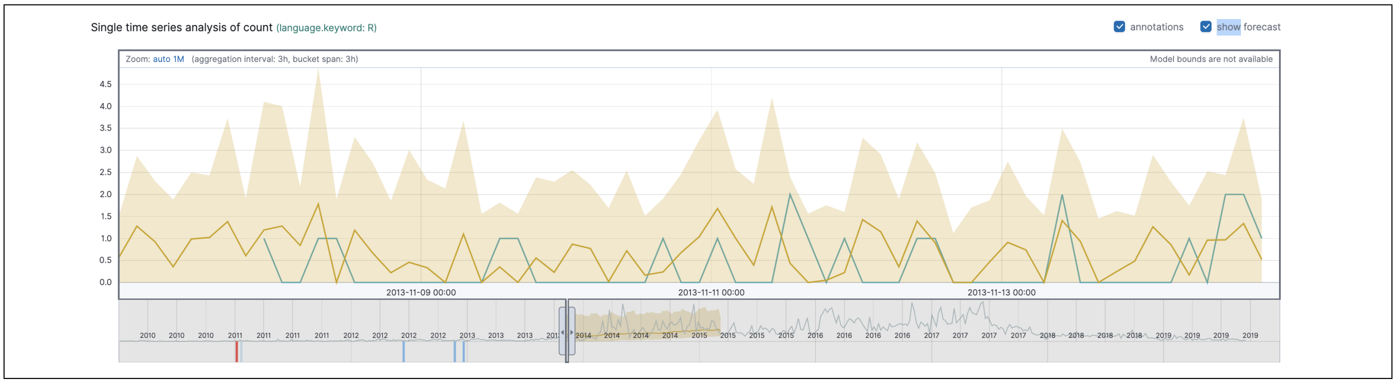


Fig. 3: R forecast and anomaly prediction

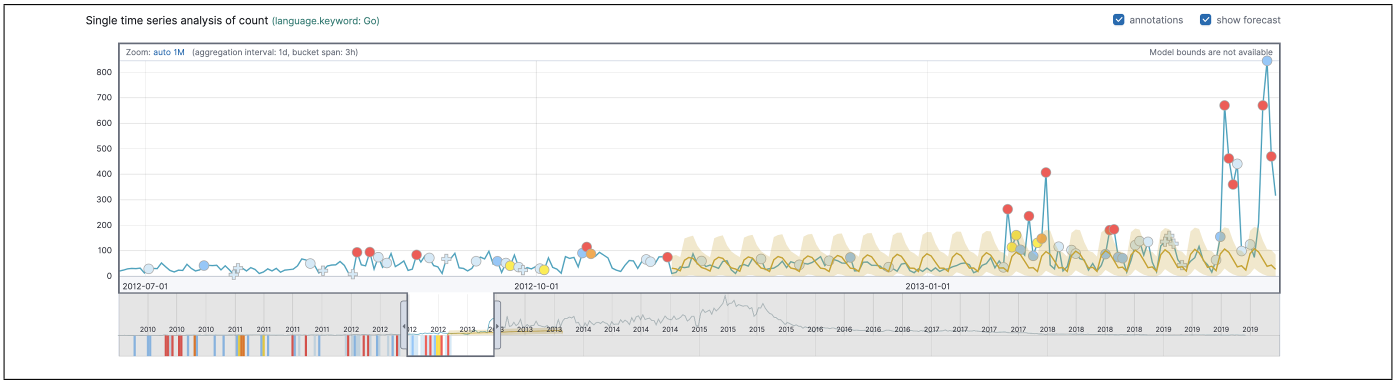
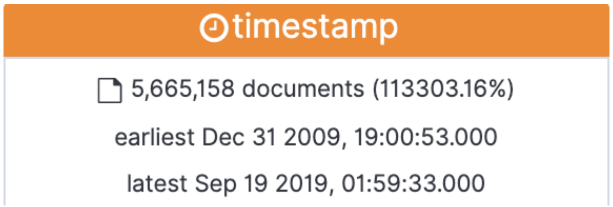


Fig. 4: Go forecast and anomaly prediction



timestamp

5,665,158 documents (113303.16%)

earliest Dec 31 2009, 19:00:53.000

latest Sep 19 2019, 01:59:33.000

Fig. 5: Timestamp of repositories loaded

Figure 6 and figure 7 shows word cloud for number of repositories by language frequency and licenses frequency.

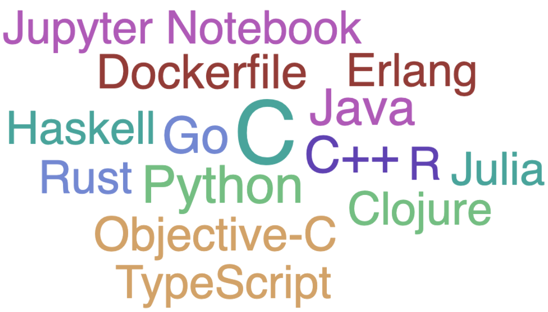


Fig. 6: Wordcloud of repositories loaded

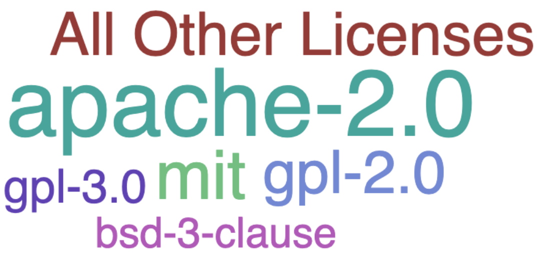
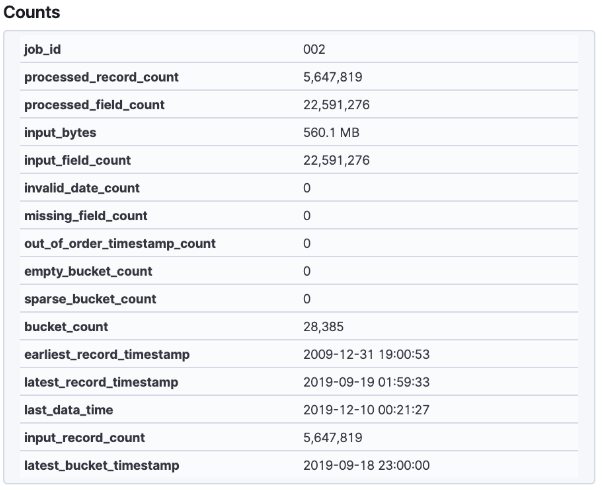


Fig. 7: Wordcloud of repositories loaded

Figure 8 shows the stats for anomaly machine learning job created and processed on elasticsearch. This job detected the results shows in 1. This job as stated earlier, was expanded to forecast, on multiple different languages and different time periods, the limit being 365 days due to memory limit for JVM and the virtual machine on GCP. This job predicted the trend in number of repositories over time, absolute accuracy was not calculated. However, by visual observation, it is closely following the historical trend for the examined languages.

Figure 3 shows running R language single time-series anomaly detection job which was scheduled to predict the trend over a period of 5 years, and from the graph, blue being actual and yellow being forecasted, the trend follows similar line. Figure 4 show the same for Go language, however, this was on a shorter period of time, due to memory and resource restrictions on GCP's virtual instance.



Counts

job_id	002
processed_record_count	5,647,819
processed_field_count	22,591,276
input_bytes	560.1 MB
input_field_count	22,591,276
invalid_date_count	0
missing_field_count	0
out_of_order_timestamp_count	0
empty_bucket_count	0
sparse_bucket_count	0
bucket_count	28,385
earliest_record_timestamp	2009-12-31 19:00:53
latest_record_timestamp	2019-09-19 01:59:33
last_data_time	2019-12-10 00:21:27
input_record_count	5,647,819
latest_bucket_timestamp	2019-09-18 23:00:00

Fig. 8: Anomaly machine learning job

V. Future Work

Due to limited time and resources, full analysis and StackOverflow's data loading and its analysis was not performed. Further extension of this research can be performed using data for following points:

- StackOverflow data for language questions and answers, its time and sentiment analysis.
- GitHub commit messages, time, releases, pull requests, code comments and forking analysis.
- Correlating both datasets with mentions of repositories, issues and links.
- Getting or creating a dataset for significant events for a subset of programming language.

more parameters and data from stack and mention of repositories in stack

VI. Conclusion

Throughout the paper, a pattern and trend has been observed for languages and how different significant event impact quantity of open source GitHub repositories. The time-series analysis helps to confirm the same. An algorithm to correlate trend with significant event occurence has not been devised. The anomalies, however, help detect outliers and relate them to a version release, support announcement or end of support announcement.

Acknowledgment

I would like to thank Dr. Charles Tappert, Dr. Ronald Frank and Professor Stephan Barabasi for mentoring and guidance. I would also like to thank the Elastic.co team to extend my license twice which allowed me to use all their features and perform analysis.

References

[1] A. S. Badashian and E. Stroulia, "Measuring user influence in github: The million follower fallacy," in *Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering*, ser. CSI-SE '16. New York, NY, USA: ACM, 2016, pp. 15–21. [Online]. Available: http://doi.acm.org/10.1145/2897659.2897663

[2] B. Szajna, "Empirical evaluation of the revised technology acceptance model," *Management Science*, vol. 42, no. 1, pp. 85–92, 1996. [Online]. Available: https://doi.org/10.1287/mnsc.42.1.85

[3] W. R. King and J. He, "A meta-analysis of the technology acceptance model," *Information and Management*, vol. 43, no. 6, pp. 740 – 755, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378720606000528

[4] A. Čižmešija and Z. Stapić, "Github as backbone in software engineering course: Technology acceptance analysis," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2019, pp. 742–746.

[5] F. Chatziasimidis and I. Stamelos, "Data collection and analysis of github repositories and users," in *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 2015, pp. 1–6.

[6] A. Nandi, A. Mandal, S. Atreja, G. B. Dasgupta, and S. Bhattacharya, "Anomaly detection using program control flow graph mining from execution logs," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 215–224.

[7] G. Gousios, “The ghtorrent dataset and tool suite,” in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR ’13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 233–236. [Online]. Available: http://dl.acm.org/citation.cfm?id=2487085.2487132

[8] “Github activity data.” [Online]. Available: https://console.cloud.google.com/marketplace/details/github/github-repos?filter=solution-type:dataset&id=46ee22ab-2ca4-4750-81a7-3ee0f0150dcb

[9] “Machine learning anomaly detectionedit.” [Online]. Available: https://www.elastic.co/guide/en/elastic-stack-overview/current/xpack-ml.html