The 14th International Conference on Advanced Computational Intelligence (ICACI)
July 15-17 2022, Wuhan, China

Sentiment Analysis of Developers' Comments on GitHub Repository: A Study

Aman Kumar
DA-IICT Gandhinagar, India
201911018@daiict.ac.in

Manish Khare
DA-IICT Gandhinagar, India
manish_khare@daiict.ac.in

Saurabh Tiwari
DA-IICT Gandhinagar, India
saurabh_t@daiict.ac.in

2022 14th International Conference on Advanced Computational Intelligence (ICACI) | 978-1-6654-7045-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICACI55529.2022.9837754

***Abstract*—The sentiments of developers play a major role in productivity, code quality, and satisfaction. The workload of the developers and their interest in a specific programming language affect the overall quality of the development process. Open source projects, where developers (or contributors) work based on their interest in contributing to the project apart of their routine work. In this paper, we are analysing the sentiments of the developers on GitHub while working on different open source projects. Our study mainly focuses on three aspects: (1) analysing the day of the week in which the comment was made by the developer, (2) emotions of the developer throughout the course of a project, and (3) emotions with different programming languages. The analysis was done by looking into the developer comments on issues, pull requests, and comments for the repository. Our results show that projects developed on Monday's tend to more negative emotion. Additionally, comments written in issues have higher negative polarity in their sentimental content, and projects developed in Java and Python have more positive comments as compared to C and C++.**

***Index Terms*—Sentiment Analysis, GitHub, Issues, Pull Request, Commit, Comments, Developers**

I. Introduction

Software development is a collaborative activity that involves humans (or developers) in different stages of the development cycle. GitHub[1] is a popular source code hosting and collaboration platform. The collaboration among developers requires communication channels to express their emotions w.r.to issues, discussions and logs. Their emotions with the development may affect productivity and the quality of the project [1]. The project managers need to be aware of the developers' emotions while working on the project to ensure that emotions would not affect project quality. In this paper, we investigate developer emotions working on open-source projects and collaborations through GitHub Repositories. Additionally, we study the relationship between their emotions, the day of the week comments were written, the time duration the project was created and the programming language used to develop the project [2].

The process of analysing emotions from texts is called Sentiment Analysis [3]. In this paper, we are analysing the sentiments of the developers on GitHub repositories while working on different open source projects. The research questions (RQs) that guide our work are:

$RQ1$ *Sentiment of Python Developer related to the day of the week.*

$RQ2$ *Sentiment of Python Project Contributors throughout the course of the project.*

$RQ3$ *Sentiment of Developers across different programming languages in which project is developed.*

To answer the formulated RQs, we have analysed issue comments, pull request comments, and commit comments posted by developers in the repositories w.r.to their emotions to the day of the week and throughout the course of the project. To determine the emotion of developers related to the day of the week and throughout the course of the project, we have selected the top 10 Python GitHub repositories fetched based on the number of stars. The comments posted by the developers on the day of the week are analysed using Sentistrength [4][2]. The Sentistrength is a lexical extraction tool that estimates the strength of sentiments in texts. The analysis of the issue comments, pull request comments and commit comments posted by the developers w.r.to the day of the week revealed that comments written on Monday have more negative polarity emotions as compared to other days of the week.

Next, we analyse the emotion of developers throughout the course of the project by grouping the comments posted by the developers. The comments were then analysed based on time duration (i.e., days/months/years) from the date of project creation. We have used the Sentistrength tool to determine the sentiments and found that comments written in issues and pull requests are more negative in polarity as compared to comments written in commit comments. To answer RQ3, i.e., emotions of the developers based on the programming languages used for development. Here, we have used 3 GitHub repositories for each of the programming languages (such as Python, Java, C, C++ and Dot Net) based on the number of stars. We observed that comments written in Python and Java programming languages have positive sentiments compared with other languages.

The organisation of the paper is as follows. Section II contains the discussion about the research papers related to the analysis of sentiments in the GitHub repository. Section III presents the proposed methodology adopted to perform sentiment analysis. Here, we discuss dataset creation, data analysis, and tool selection for sentiment analysis. Section IV highlight the answers for RQs. Section V discuss and presents

[1]https://github.com/

[2]http://sentistrength.wlv.ac.uk

978-1-6654-7045-2/22/$31.00 ©2022 IEEE

the results of the analysis. Section VI presents the conclusion and future work of our study.

II. Related Work

In this section, we discuss about the various work done by several authors on sentiment analysis and sentiment analysis on GitHub repositories.

A. Sentiment Analysis

Several studies have been conducted by several authors on sentiment analysis. An important aspect in approaching Sentiment analysis is represented by the work of Feldman [5]. The author discussed the techniques to perform sentiment analysis and the various applications of sentiment analysis that help us in solving real-world problems. The author discussed how sentiment analysis techniques can be used to solve complex problems which some industries face and how they can simplify their problems using sentiment analysis.

Beigi et al. [6] discussed the application of sentiment analysis on disaster relief and the overview of sentiment analysis in social media. The author talks about how sentiment analysis is not limited to politics, business intelligence, etc. The author studied the reaction of the local crowd and used such information to improve disaster management.

Dolianiti et al. [7] presented the applications of sentiment analysis in education. The author talked about how cognition and emotions are involved in every learning process and how student profiles can be enhanced with information regarding the effective state. The author explored many different ways in which sentiment analysis can be applied in the educational domain.

Das et al. [8] discussed real-time prediction of stock by analysing sentiment of Twitter streaming data in their study. The author attempts to make decisions related to finance such as stock market prediction, to predict the prices of a company's stock. To perform this, Twitter streaming data has been considered for scoring the impression that is carried for a particular firm.

B. Sentiment Analysis on GitHub

Sentiment Analysis on GitHub has many significant studies. One of them is done by Pletea et al. [9]. The authors detected the relationship between security-related topics and sentiment analysis on comments. As application security is becoming increasingly prevalent during software development, the authors studied that the majority of security-related comments have a negative sentiment. These findings helps in gaining the importance of proper training of developers.

Huq et al. [10] discussed the relationship between developer sentiments and their relationship to software bugs. The author correlated several patterns by conducting sentiment analysis commits in GitHub repositories to find the relation between developer sentiment and software bugs. It is statistically observed by the author that commits related to bug fixes are more negative than regular commit comments.

Skriptsova et al. [11] presented the activity of newcomers in communicative posts on GitHub. The author aims to investigate relationship of newcomers through sentiment analysis of issue and pull request comments in GitHub repositories. Author uses sentiment analysis to find reactions of 'old' and 'new' developers contributions. The author observed that majority of comments are neutral but negativity is higher for new comers.

Huq et al. [12] described how developer sentiments are related to bugs, by analysing sentiments between regular and Fix-Inducing Changes (FIC).FIC are the changes to code that can introduce bugs to system. The author analysed pull requests from GitHub repositories that contain contributor discussions and analyzed the sentiment of those discussions.

III. Methodology

This study performs sentiment analysis on GitHub comments posted by the developers for the issues, pull requests and commits to assess the developer sentiments throughout the project duration, on the specific day of the week, and with a programming language. The overall methodology adopted for analysis is divided into four parts. First, the data is collected from GitHub repositories. Second, the obtained data is classified on the three formulated RQs mentioned above. Third, the sentiment analysis is conducted on the artefacts. Finally, the tool is selected for performing sentiment analysis. Fig. 1 shows the overview of the methodology and following subsections provides their descriptions.

A. Dataset Creation

The data collection from GitHub is done using pyGitHub[3], a library in python. pyGitHub is a library written in python to make use of GitHub API. Using pyGitHub, we have retrieved issue comments, pull request comments and commit comments posted by developers in the GitHub repositories. The pyGitHub contains functions and variables to communicate with the GitHub API and fetch the desired results from GitHub repositories.

To answer RQ1 and RQ2, we have considered the top 10 python repositories from GitHub based on the number of stars. In RQ1, we grouped the developer comments w.r.to the date, comments are written to determine the day of the week for getting the date of the creation of the comment. In RQ2, we have considered only comments posted by the contributors. Then, we sorted those comments based on the date of posting those comments. Next, we grouped those comments based on time duration from the date of the creation of the comment (e.g., 1 month, 2 months, 3 months, 6 months, 1 year, 2 year and 5 years).

For RQ3, we used the comments posted on the top 3 GitHub repositories for each of the programming languages (such as Python, Java, C, C++ and Dot Net) based on the number of stars. After extracting issue, pull request and commit comments, we grouped each of these comments based on the programming languages used to develop the projects.

[3]https://pygithub.readthedocs.io/en/latest/

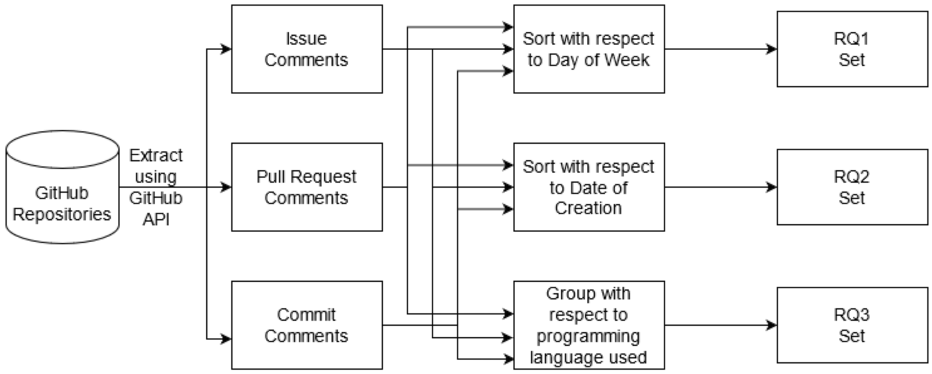

Fig. 1. Proposed Methodology for Sentiment Analysis

B. Dataset

The repositories used for answering RQ1 and RQ2 is shown in Table I. Table I show ten repositories that contain approximately 230,000 issue with comments, 120,000 pull requests with comments and 14,613 commit with comments. The repositories used for answering RQ3 is shown in Table II. It shows the set of 3 repositories considered for each of the programming languages. The dataset has approximately 135,000 issues with comments, 95,000 pull requests with comments and 23,000 commits with comments. The dataset contains five fields:

1) repository: the name of repository.
2) user: the user that written the comment.
3) text: the text of the comment.
4) cdate: the date of creation of comment.
5) edate: the date of closing of issue (only in closed issues).

TABLE I
GITHUB PYTHON REPOSITORIES FOR RQ1 AND RQ2.

Repository	Issues	Pull Requests	Commits
pytorch/fairseq	2382	46	1752
keras-team/keras	10414	11	5398
scrapy/scrapy	2372	307	8604
AdguardTeam/AdguardFilters	70788	43	79988
freeCodeCamp/freeCodeCamp	15097	31	26899
scikit-learn/scikit-learn	8249	738	26502
home-assistant/core	18842	318	33036
ageitgey/face_recognition	1095	12	211
apache/superset	5653	87	6732
tensorflow/tensorflow	32931	188	112600

C. Sentiment Analysis

In this section, we describe the steps for performing sentiment analysis. The first step of sentiment analysis is data pre-processing. The data obtained from GitHub is in raw format. As the texts with special characters such as emoticons, punctuations, etc do not contribute to Sentiment Analysis, hence need to remove those before processing the data. The data pre-processing can be done in steps as shown in Fig. 2.

1) Tokenization - Tokenization is converting a piece of text into tokens. It is the task of chopping it up into pieces, called tokens. Tokens are the building blocks of the Natural Language. The common ways of processing raw text happen at the token level. We have used functions of the list to tokenise the text [13].
2) Lower Casing - Converting the tokens into lower case letters. We converted all texts into lowercase so that the processing algorithm does not recognize capital letters and lowercase letters separately. We have used the lower method of string in python for lower casing the text [14].
3) Removing Punctuation - It helps us to optimize text so that the processing algorithm works more efficiently because punctuation's do not add more value to sentiment data. We have removed punctuation from the text by using a punctuation variable from the string in python [14].
4) Removing Stop Words - Stop words are the words that do not contribute to sentiment analysis. These stop words are a set of commonly used words that carry very little information. Stop words such as I, am, the, etc. We have used nltk.corpus[4] library to remove stop words from text [14].
5) Stemming - It is a process of gaining the root word of a word by removing affixes and suffixes. Some examples of stemming are like for word 'confirmed' the stemmed word is 'confirm'. We have used WordNetLemmatizer

TABLE II
GITHUB PYTHON REPOSITORIES FOR RQ3

Language	Repository	Issues	Pull Requests	Commits
Python	pytorch/fairseq	2382	46	1752
Python	keras-team/keras	10414	11	5398
Python	scrapy/scrapy	2372	307	8604
Java	ReactiveX/RxJava	70788	43	79988
Java	google/guava	15097	31	26899
Java	spring-projects/spring-boot	8249	738	26502
C	Genymobile/scrcpy	18842	318	33036
C	obsproject/obs-studio	1095	12	211
C	git/git	5653	87	6732
C++	x64dbg/x64dbg	1974	629	4651
C++	fffaraz/awesome-cpp	484	593	1393
C++	microsoft/calculator	728	755	649
DotNet	SeleniumHQ/selenium	7179	2108	26814
DotNet	dotnet/core	4110	1350	3186
DotNet	quozd/awesome-dotnet	75	923	1450

[4]https://www.nltk.org/api/nltk.corpus.html

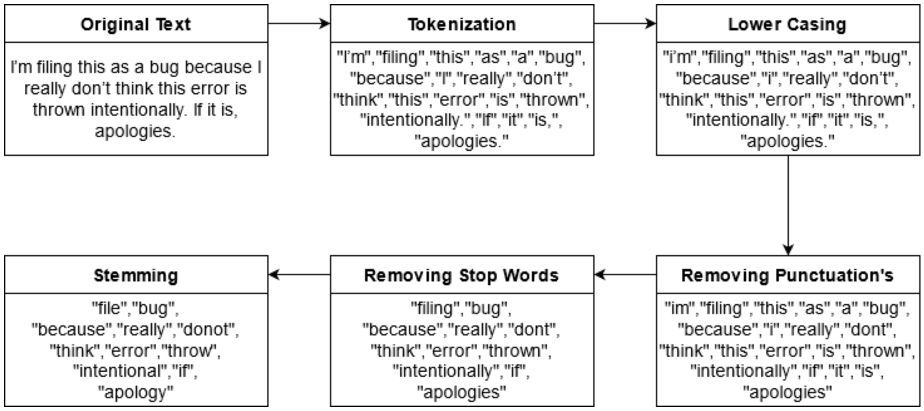

Fig. 2. Steps of Data Pre-Processing for Sentiment Analysis

from nltk.stem[5] library to perform word stemming [14].

D. Tool Selection

After pre-processing the data and making it fit for our sentiment analysis algorithm, we have used different classifiers and an algorithm to classify our text into sentiments.

Classification of text into sentiments is done using *SentiStrength*[6] a lexical extraction tool implemented in python to classify our text into sentiment scores. *SentiStrength* is a sentiment analysis tool used to convert texts into sentiment scores. We have chosen this tool because it has high accuracy as reported by previous studies [2] [15] [16]. *SentiStrength* assign a sentiment score to every word. Words which convey negative emotions are assigned a sentiment score between -5 and -1, whereas words which contains positive emotions are assigned a sentiment score between 1 and 5. After assigning score to word *SentiStrength* generates senti-score. Senti-score contains two values which signifies positive and negative score. By taking the sum of both values we can find the final sentiment score. If the sum is positive then it signifies that the sentiment of word is positive, else if sum is zero then the word is neutral i.e. it does not reflect any emotion and if sum is negative then the sentiment of word is negative.

Consider the following example of a commit log: "I'm filing this as a bug because I really don't think this error is thrown intentionally. If it is, apologies". *SentiStrength* provides us sentiscore for each word and sentence as:

```
``file[0] bug[-1] because[0] really[0]
donot[0] think[0] error[-1] throw[0]
intentional[0] If[0] apology[0] [[Sentence
= -3,1 = word max 1-5]]''.
```

The positive rating is 1 on the scale of 1 to 5 and the negative rating is -2 on the scale of -1 to -5. So the sentence comes out to be negative as when we take sum 1 and -2, it comes out as -1, which signifies negative sentiment. Here, we have taken the maximum scores. This signifies the polarity of the commit log. The steps for analysing data using *SentiStrength* is shown in Fig. 3 [2] [15] [17].

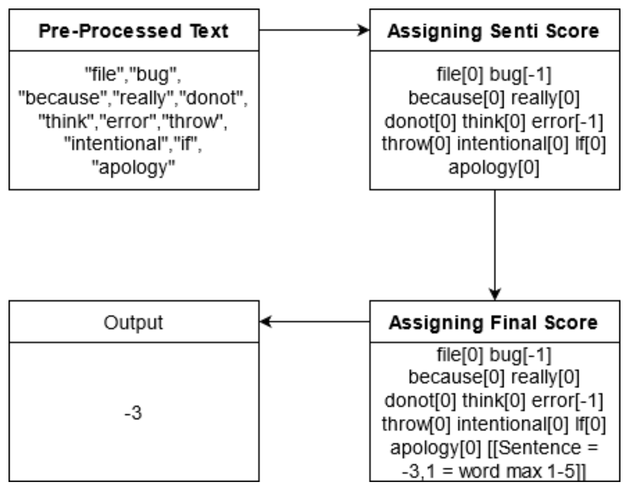

Fig. 3. Process of analysing Sentiments using *SentiStrength*

IV. Answers for RQs

In the following sections, we present an analysis of the emotions expressed in issue comments, pull request comments and commit comments with additional factors such as day of the week, throughout the course of a project and, for different programming languages.

A. Sentiment of Python Developer related to day of week (RQ1)

We have analyzed emotions present in issue comments, pull request comments as well as commit comments. We have analyzed a total of 10 projects which are written in python. The average score tended between -1 to 1 as most of the comments only describe technical aspects and no emotions or have a slight positive or negative factor in them. Fig. 4 and Table III shows the sentiment of a python developer related to the day of the week on which they write the comment. Most comments, 85%, were written during the week, while 15% during the weekends. Developer's shows more negative sentiments at the starting of the week as we can see that comments written

[5] https://www.nltk.org/api/nltk.stem.html
[6] http://sentistrength.wlv.ac.uk/

on Monday are 42.58% negative. As the week progresses the sentiment of developers shifts from negative to positive but as the weekend approaches the sentiment of developers shifts to negative as we can see the sentiment on Friday and Saturday are 45.75 % and 43.86% negative. The reason behind this can be as Monday is the first day of the week, developers experience more negative emotions as they have to work after the weekend also like the weekend approaches developers reflects more negative emotions [18] [19].

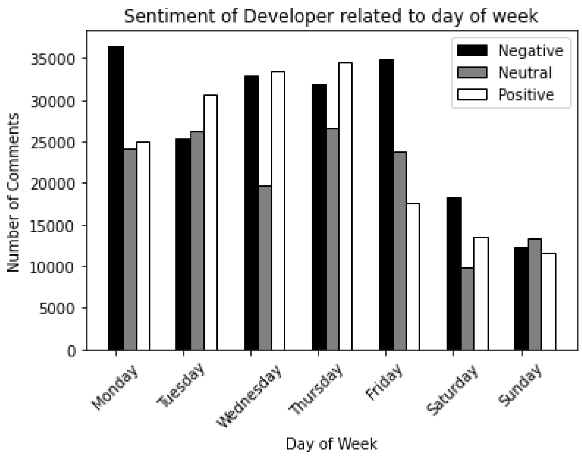

Fig. 4. Sentiment of Developer with respect to Week Days.

TABLE III
SENTIMENT OF DEVELOPER WITH RESPECT TO WEEK DAYS.

Day of Week	Negative	Neutral	Positive
Monday	42.58	28.23	29.19
Tuesday	30.84	31.85	37.31
Wednesday	38.23	22.94	38.83
Thursday	34.29	28.60	37.11
Friday	45.78	31.09	23.13
Saturday	43.86	23.58	32.56
Sunday	33.03	35.99	30.98

We observe the same trend in the case of issue comments, the comments were written on Monday, Friday and Saturday reflect more negative sentiments of developers as compared to other weekdays as shown in Fig. 5 and Table IV. Whereas comments written in pull requests have negative sentiments consistently throughout the weekdays as shown in Fig. 6 and Table V. But in the case of comments written in commits, the sentiment of developers are positive throughout the week as shown in Fig. 7 and Table VI. The reason behind this can be the comments written in issues and pull requests signify errors or changes in source code, so comments written in issues and pull requests are more negative compared to comments written in commits.

B. Sentiment of Contributors throughout the course of the project (RQ2)

To study the relationship between the sentiments expressed by contributors in comments written in issues, pull requests and commits, and the time spent on the project we analyzed the data of 10 python projects. We manually sorted the comments

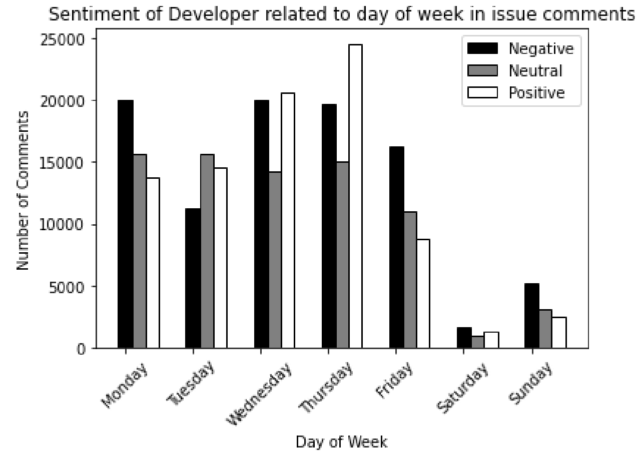

Fig. 5. Sentiment of Developer with respect to Week Days in Issue Comments.

TABLE IV
SENTIMENT OF DEVELOPER WITH RESPECT TO WEEK DAYS IN ISSUE COMMENTS.

Day of Week	Negative	Neutral	Positive
Monday	40.46	31.68	27.86
Tuesday	27.12	37.81	35.07
Wednesday	36.58	25.92	37.50
Thursday	33.16	25.36	41.48
Friday	44.99	30.52	24.49
Saturday	42.00	24.90	33.10
Sunday	48.62	28.65	22.73

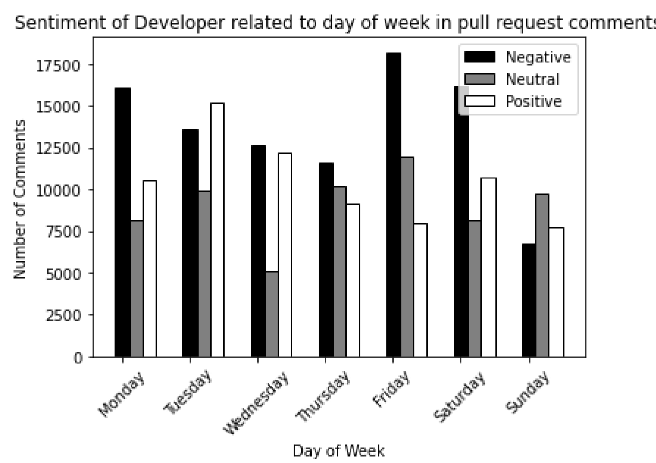

Fig. 6. Sentiment of Developer with respect to Week Days in Pull Request Comments.

by the date on which the comment is posted and grouped the data with respect to time spent from the creation of a project. We have analyzed the data for 1 month, 3 months, 6 months, 9 months, 1 year, 2 years and 5 years from the creation of the project. As shown in Fig. 8 and Table VII the sentiment of the developer is mostly negative throughout the project. The overall sentiment becomes negative as there are more issue comments and pull request comments which are negative in nature as compared to commit comments.

We observed that the sentiment of the contributor in issue

TABLE V
SENTIMENT OF DEVELOPER WITH RESPECT TO WEEK DAYS IN PULL REQUEST COMMENTS.

Day of Week	Negative	Neutral	Positive
Monday	46.29	23.52	30.19
Tuesday	35.11	25.67	39.22
Wednesday	42.22	16.92	40.86
Thursday	37.53	32.99	29.48
Friday	47.76	31.25	20.99
Saturday	46.11	23.32	30.57
Sunday	27.70	40.31	31.99

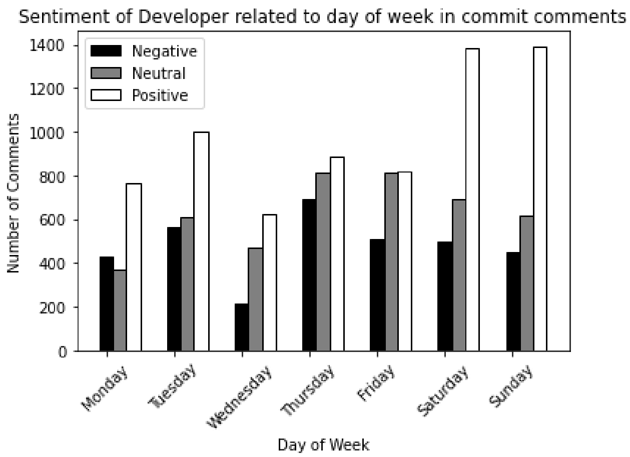

Fig. 7. Sentiment of Developer with respect to Week Days in Commit Comments.

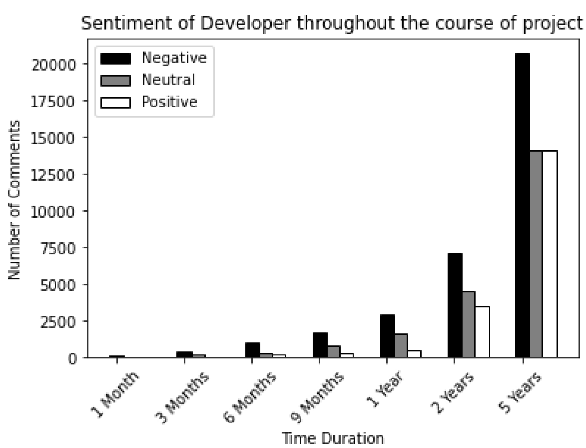

Fig. 8. Sentiment of Developer throughout the course of the project.

TABLE VI
SENTIMENT OF DEVELOPER WITH RESPECT TO WEEK DAYS IN COMMIT COMMENTS.

Day of Week	Negative	Neutral	Positive
Monday	27.48	23.66	48.86
Tuesday	25.99	28.15	45.86
Wednesday	16.29	36.03	47.68
Thursday	28.91	33.97	37.12
Friday	23.88	37.82	38.30
Saturday	19.40	26.87	53.73
Sunday	18.32	25.0	56.68

TABLE VII
SENTIMENT OF DEVELOPER THROUGHOUT THE COURSE OF THE PROJECT.

Day of Week	Negative	Neutral	Positive
1 Month	52.94	36.02	11.04
3 Months	65.92	28.02	6.06
6 Months	66.49	21.31	12.20
9 Months	59.74	28.42	11.84
1 Year	57.49	31.53	10.98
2 Years	47.22	29.71	23.07
5 Years	37.17	31.32	31.51

comments are negative throughout the project as shown in fig 9 and Table VIII. As comments written in issues represent bugs or errors in code, hence these comments are more negative in nature. Also, comments written in pull requests have negative sentiment throughout the project as shown in Fig 10 and Table IX as they signify code changes, other developers suggest changes in code through pull requests. But comments written in commits are positive in nature throughout the project as shown in Fig 11 and Table X. Commit comments are written when a user commits the repository, i.e., they save the state of the repository so that a user can get versions of the project. So it consists of more positive comments as compared to issues and pulls requests.

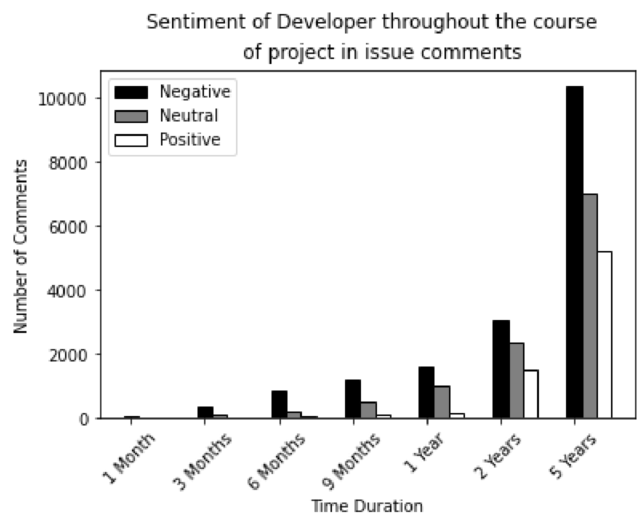

Fig. 9. Sentiment of Developer throughout the course of the project in Issue Comments.

TABLE VIII
SENTIMENT OF DEVELOPER THROUGHOUT THE COURSE OF THE PROJECT IN ISSUE COMMENTS.

Day of Week	Negative	Neutral	Positive
1 Month	57.95	38.63	3.42
3 Months	73.60	23.59	2.81
6 Months	77.07	18.95	3.98
9 Months	64.89	28.86	6.25
1 Year	57.69	36.28	6.03
2 Years	43.84	34.19	21.97
5 Years	45.82	31.09	23.09

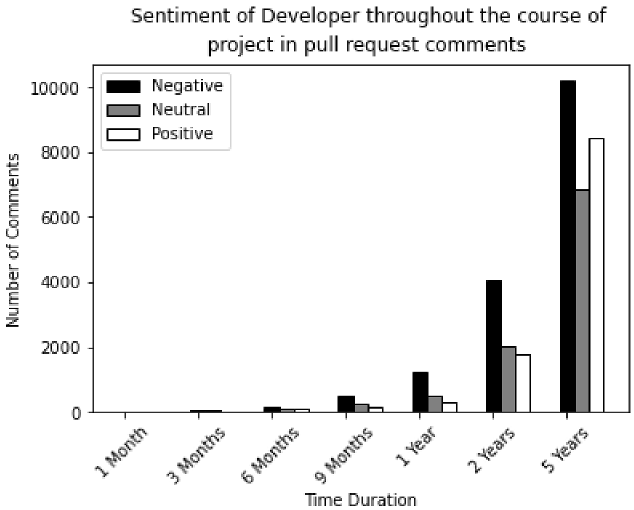

Fig. 10. Sentiment of Developer throughout the course of the project in Pull Request Comments.

TABLE IX
SENTIMENT OF DEVELOPER THROUGHOUT THE COURSE OF THE PROJECT IN PULL REQUEST COMMENTS.

Day of Week	Negative	Neutral	Positive
1 Month	47.50	32.50	20.00
3 Months	46.62	36.19	17.19
6 Months	46.49	26.49	27.02
9 Months	54.50	27.56	17.94
1 Year	60.57	25.68	13.75
2 Years	51.57	25.81	22.62
5 Years	40.01	26.90	33.09

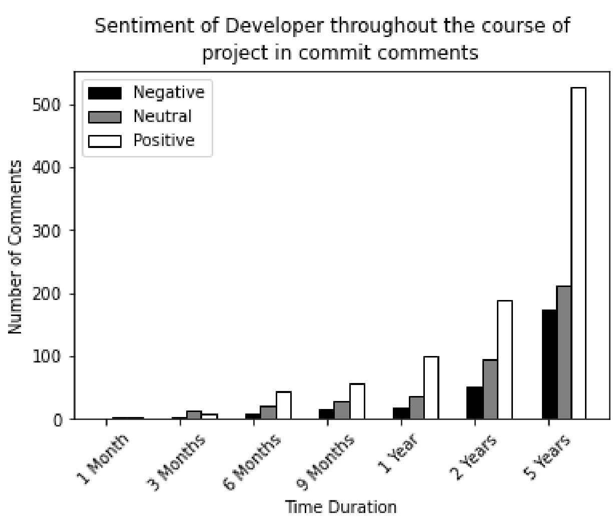

Fig. 11. Sentiment of Developer throughout the course of the project in Commit Comments.

C. *Sentiment of developers across different programming languages (RQ3)*

The repositories in this paper were implemented in the 5 most used programming languages. These are Python, Java, C, C++ and DotNet. We have chosen the top 3 repositories for each language. This study is performed on the issue comments, pull request comments and the commit comments

TABLE X
SENTIMENT OF DEVELOPER THROUGHOUT THE COURSE OF THE PROJECT IN COMMIT COMMENTS.

Day of Week	Negative	Neutral	Positive
1 Month	14.28	28.57	57.15
3 Months	12.00	52.00	36.00
6 Months	12.16	29.72	58.12
9 Months	15.68	28.43	55.89
1 Year	12.41	22.87	64.72
2 Years	15.52	28.05	56.43
5 Years	19.00	23.16	57.84

of the particular libraries. Fig. 12 and Table XI shows the comparison of different sentiment scores for each language. We have observed that the Sentiment of python and java developers have more positive sentiments 37.97% and 36.85% respectively as compared to C and C++ developers 29.8% and 27.55% respectively. Whereas DotNet developers have neutral sentiments. We see these trends as C and C++ are the base of languages and are harder to learn and require a deeper understanding to program and implement, whereas python is easy to learn and implement.

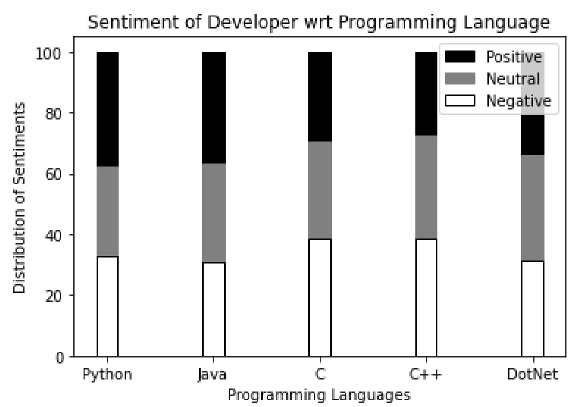

Fig. 12. Sentiment of Developer with respect to Programming Language.

TABLE XI
SENTIMENT OF DEVELOPER WITH RESPECT TO PROGRAMMING LANGUAGE.

Programming Language	Negative	Neutral	Positive
Python	32.72	29.31	37.97
Java	31.02	32.13	36.85
C	38.75	31.45	29.80
C++	38.68	33.76	27.56
DotNet	31.18	34.98	33.84

V. DISCUSSIONS

Our study suggests that comments logs that are submitted on Monday, Friday and Saturday have more negative sentiment compared to comments written on any other day of the week. This is also reported by E. Guzman, et al [20] in their study during sentiment analysis of commit logs. They also observed that the comments written in commits have more negative sentiment as compared to any other day of the week. From this analysis, one can help developers to get a more positive response from contributors. Our results also suggest that comments written in issue comments and pull request comments are more negative as compared to comments written in commit comments. Our results provide the relationship between the sentiment of developers and the course of the project. We also found that comments written in python and java repositories are more positive as compared to programming languages such as C and C++.

Our study was conducted on the top 3 GitHub repositories based on a number of stars. However, the projects were not related in terms of the function they are performing so we cannot say that particular programming language contains more positive sentiment comments.

VI. Conclusions and Future Work

This paper deals with the sentiment analysis of developers on various factors by analysing the open-source repositories based on their contributions. We have analysed developer sentiments for the days of the week, and the behaviour of developers throughout the course of the project. Our analysis results revealed that developer sentiments on the first day of the week, Monday, are more negative than on other days. We can also conclude that sentiment in reported issues are mostly negative as the developer reacts negatively to bugs whereas commit comments are mostly positive in nature. This trend can also be seen in the whole project's life cycle. The issue comments are more negative and commit comments are positive throughout the course of the project.

Additionally, we have also analysed the sentiments of developers for the programming languages they are using. We found that the developer comments on python repositories are, on average, positive and comments on programming languages like C and C++ are comparatively more negative.

In the future, we plan to study the impact of sentiments on quality of code written in repositories and productivity of the software. We also plan to make our dataset more diverse and scaled. We can also explore more methods for sentiment analysis

References

[1] Z. Chen, Y. Cao, X. Lu, Q. Mei, and X. Liu, "Sentimoji: An emoji-powered learning approach for sentiment analysis in software engineering," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 841–852. [Online]. Available: https://doi.org/10.1145/3338906.3338977

[2] E. Guzman, D. Azócar, and Y. Li, "Sentiment analysis of commit comments in github: An empirical study," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 352–355. [Online]. Available: https://doi.org/10.1145/2597073.2597118

[3] V. G and D. Chandrasekaran, "Sentiment analysis and opinion mining: A survey," *Int J Adv Res Comput Sci Technol*, vol. 2, 06 2012.

[4] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 69–78. [Online]. Available: https://www.aclweb.org/anthology/C14-1008

[5] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, p. 82–89, Apr. 2013. [Online]. Available: https://doi.org/10.1145/2436256.2436274

[6] G. Beigi, X. Hu, R. Maciejewski, and H. Liu, *An overview of sentiment analysis in social media and its applications in disaster relief*, ser. Studies in Computational Intelligence. Springer Verlag, 2016, pp. 313–340, funding Information: This material is based upon thework supported by, or in part by,Office of Naval Research (ONR) under grant number N000141410095, the NSF under Grant Number 1350573, and the U.S. Department of Homeland Security?s VACCINE Center under Award Number 2009-ST-061-CI0001. Publisher Copyright: © Springer International Publishing Switzerland 2016.

[7] F. S. Dolianiti, D. Iakovakis, S. B. Dias, S. Hadjileontiadou, J. A. Diniz, and L. Hadjileontiadis, "Sentiment analysis techniques and applications in education: A survey," in *Technology and Innovation in Learning, Teaching and Education*, M. Tsitouridou, J. A. Diniz, and T. A. Mikropoulos, Eds. Cham: Springer International Publishing, 2019, pp. 412–427.

[8] S. Das, R. K. Behera, M. kumar, and S. K. Rath, "Real-time sentiment analysis of twitter streaming data for stock prediction," *Procedia Computer Science*, vol. 132, pp. 956–964, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918308433

[9] D. Pletea, B. Vasilescu, and A. Serebrenik, "Security and emotion: Sentiment analysis of security discussions on github," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 348–351. [Online]. Available: https://doi.org/10.1145/2597073.2597117

[10] S. F. Huq, A. Z. Sadiq, and K. Sakib, "Is developer sentiment related to software bugs: An exploratory study on github commits," in *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2020, pp. 527–531.

[11] E. Skriptsova, E. Voronova, E. Danilova, and A. Bakhitova, "Analysis of newcomers activity in communicative posts on github," in *Digital Transformation and Global Society*, D. A. Alexandrov, A. V. Boukhanovsky, A. V. Chugunov, Y. Kabanov, O. Koltsova, and I. Musabirov, Eds. Cham: Springer International Publishing, 2019, pp. 452–460.

[12] S. F. Huq, A. Z. Sadiq, and K. Sakib, "Understanding the effect of developer sentiment on fix-inducing changes: An exploratory study on github pull requests," in *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, 2019, pp. 514–521.

[13] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 55–60. [Online]. Available: https://www.aclweb.org/anthology/P14-5010

[14] D. Kawade, "Sentiment analysis: Machine learning approach," *International Journal of Engineering and Technology*, vol. 09, pp. 2183–2186, 06 2017.

[15] B. Yang, X. Wei, and C. Liu, "Sentiments analysis in github repositories: An empirical study," in *2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW)*, 2017, pp. 84–89.

[16] G. Destefanis, M. Ortu, D. Bowes, M. Marchesi, and R. Tonelli, "On measuring affects of github issues' commenters," in *2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, 2018, pp. 14–19.

[17] S. Vinayak, L. Alina, and S. Bonita, "Analyzing developer sentiment in commit logs," in *Proceedings of the 13th International Conference on Mining Software Repositories*, ser. MSR '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 520–523. [Online]. Available: https://doi.org/10.1145/2901739.2903501

[18] V. Sinha, A. Lazar, and B. Sharif, "Analyzing developer sentiment in commit logs," in *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*, 2016, pp. 520–523.

[19] N. Imtiaz, J. Middleton, P. Girouard, and E. Murphy-Hill, "Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people," in *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, ser. SEmotion '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 55–61. [Online]. Available: https://doi.org/10.1145/3194932.3194938

[20] E. Guzman, D. Azócar, and Y. Li, "Sentiment analysis of commit comments in github: An empirical study," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 352–355. [Online]. Available: https://doi.org/10.1145/2597073.2597118