# ANALYZING EMOTIONAL CONTAGION IN COMMIT MESSAGES OF OPEN-SOURCE SOFTWARE REPOSITORIES

Rashmi Dhakad[1] and Luigi Benedicenti[2]

[1]Faculty of Computer Science, University of New Brunswick, Fredericton, Canada
[2] Dean, Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

***ABSTRACT***

*For more than a decade scientist have focused on the emotions of software developers in order to understand emotion's impact on their productivity, creativity, and quality of work. In modern time, there is a sharp rise in open-source software collaborations and software development models that are globally distributed. A crucial aspect of these collaborations is the affect of emotional contagion. Emotional contagion is a phenomenon of transfer of one's affective state to another. In this research study, we follow through previously established research and build on it how emotional contagion happens in large open-source software development. We further establish how emotional contagion happens during different time and how it affects the overall development process.*

***KEYWORDS***

*Emotional Contagion, Software development process, open-source repositories, OSS, sentiment analysis, commits*

## 1. INTRODUCTION

Software Development is a human-centered process involving close interactions[1]. It requires close human interactions. Software developers use different tools and channels like mailing lists, issue training tools, version control systems, etc. in order to complete various tasks, since the development process is a close collaborative activity [2]. Working in such an environment often leads to developer's expressing emotions while dealing with different tasks in the process.

Emotions have been proven to influence people's decision making skills and their ability to work with others in a team [3] [1]. These emotions can influence different skills required for successful development like productivity, decision-making, problem-solving skills, etc. Emotion or sentiment mining has been initially developed to gain insight into the sentiment polarity of consumer's feedback Recently it has gained more popularity within the software development community, as software development is a human centered process [4]. In order to understand different aspect of developer's skills like problem-solving, decision-making ability, productivity, sense-making etc. researchers are now working to understand the role of sentiment on the developer's skills during a development process.

David C. Wyld et al. (Eds): NLPML, AIAP, SIGL, CRIS, COSIT, DMA -2023
pp. 113-133, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130810

Using the method adopted by researchers to mine sentiments from communications amongst developers during the development process in order to better understand the impact of sentiments on the whole development process can give us better insights into building tools that can help leadership, management and developers themselves to build better environment for work that will lessen the gap of communication amongst management and development teams. Measuring emotions is an active research area. Researchers have used different methods and tools in identifying, categorizing and measuring the sentiments[4]–[8]. Though initially developed to provide feedback on online reviews by consumers, these tools were mainly trained on product reviews and their validity within the software engineering domain is still debatable [5], [6].

With globalization, more work is being done virtually including building software. Open-source software development provides the software community with reach to every developer around the world. Often these open-source software development teams operate in virtual environments, via virtual communication channels. These channels are often reflect the emotions displayed by team members during the development phases One drawback of working in virtual environments is the difficulty of keeping track of different time zones and thus different versions of developed tasks and assignments, which is why in such cases issue tracking systems come in handy. These tracking systems not only provide the developers with the ease of showcasing their work, but also help the team with feedback and communication on different pipelines of a project. Often the communication channel of these systems can see emotional outbursts from team members, leading to unfavorable outcomes. Understanding these emotional outbursts and display of emotions can help researchers better understand the team development processes and results of task assignments, and further their associativity with productivity and other skills in virtual environments and open-source software development.

Emotional contagion is the transfer of emotions amongst different individuals, either consciously or subconsciously. Group contagion also coexists alongside contagion affect of individuals amongst themselves [9]. As Barsade et Al. defined in their study, group emotions, which can be termed as group affective tone, can influence the work outcomes and we can expect different results in case of a group contagion.

Furthermore, emotional contagion is termed as a type of social influence, in which the emotional state or emotions of an individual or group of individuals (short term reaction towards an environment stimulus) can influence others, which results in the change in the emotional dynamic of the entire team or group. Researchers have previously focused on the productivity of developers, testers, engineers as teams and as individuals[4], [10]–[13], but there has been a research gap in the context of contagion of emotions amongst individuals in a team.

Hence, the objective of this study is to investigate the presence of emotional contagion patterns in open-source software development projects currently underway on Version Control systems like Git.

In this research paper we try to provide the results of analysis conducted to test the hypothesis presented our previous study[14]:

1. *"A negative response may evoke a negative contagion among the community."*
2. *"A positive response may evoke a positive contagion among the community."*
3. *"A neutral response can be deemed as neither positive or negative, and thus does not affect the curve, hence its contagion capability cannot be determined."*
4. *"The day of the week also affects the resulting sequence, as negative sequences or positive sequences seem to occur at different and recurring intervals of the week."*
5. *"A negative sequence leads to more commits with errors."*

6. *"A positive sequence leads to a smaller number of commits as well as fewer commits with errors."*

In this paper we try to build on the analysis and provide evidence through which we can test these hypotheses.

Based on the presented hypotheses we form the following research questions:

RQ1: What is the general sentiment of commit messages amongst different size projects? Do projects with a greater number of commits tend to have a greater number of emotional contagion episodes?
RQ2: Do projects with positive or negative sentimental polarity invoke positive or negative emotional contagion?
RQ3: Does having more members with neutral sentiment polarity result in a greater number of neutral contagion episodes?
RQ4: Do different days of the week have different numbers of contagion effects, i.e. are there more contagious episodes at the start of the week compared to the weekend?

We benefit from the open source repositories of GitHub as well as the archival collection of data at project GH Torrent [15] for the dataset required for this study. Keeping the objective in mind, we present a method workflow that analyzes the commit messages of authors in these open-source repositories for the detection and quantification of emotional contagion occurrences and allow further studies on the consequences of this phenomenon.

In this study, we analyzed commit messages of 4 different open-source projects on GitHub. We gathered more than 150K commit messages from different repositories, with more 10K authors. We extracted the sentiment polarity of these commit messages in order to find various episodes of emotional contagion, and determine how this contagion can affect the development process.
The paper is divided into further sections, with section 2 detailing the previous work done by researchers on emotions and measuring emotions in software engineering, particularly measuring developers' emotions. Section 2 further describes the concept of emotional contagion as described by psychological studies and previous studies done on the role of affect in software engineering. Section 3 outlines the method workflow of the analysis and the results. In the end we compile the results and discussion of the analysis and compare the analysis with previous research and discuss what future work can be done to further this study.

## 2. BACKGROUND WORK

### 2.1. Emotions in Software Engineering

Prioritizing human factors in software engineering has become increasingly popular in recent times, in order to increase productivity and other skills of developers. Emotions are essentially results of different activities, that affect one's decision-making ability and ability to work with others.

Researchers generally classify emotions with sentiments, which are classified into positive, negative and in some cases neutral. Measuring and capturing the presence of emotions have been done by researchers in earlier studies. Different studies have used different methods for capturing the emotions; for example, in some studies researchers have used surveys and self analysis questionnaires, other studies have followed the methodology of counting the keystrokes and mouse clicks, some used lexicon and keywords based classification method to identify the emotions in textual communication logs, etc. [16],[17],[2], [3], [5], other studies have used biometrics in order to detect the emotions of developers [18]. Capturing the emotions can be done using different methods as we have seen in earlier studies, but the reliability of these methods in terms of software engineering is still debatable. Since a lot of these tools and methodologies were initially developed for capturing the emotions of consumers of social media or marketing, their reliability for modern software development processes is controversial.

When it comes to the representation of emotions, there are various models which have been extensively studied both in the empirical software engineering community in context of developer's emotions, and in psychology. To say which is the best and most used model would be difficult, as each researcher uses their own principles for the selection of the model which represents the emotions best suited for their study. However, having stated this, the most common models, we came across during the literature review are the dimensional and discrete model of emotions. Researcher Paul Ekman identified 15 emotions that have distinctive expression [19]; some of these emotions are amusement, embarrassment, guilt, fear, pride, sadness, etc. Ekman's dimensional model includes valence, arousal and dominance (VAD). Valence describes the nature of the experience, and is associated with being pleasant and unpleasant, arousal is described as the stimulation level of emotion, for example joy gives high arousal level, so does anger, while tiredness will give low level of arousal, dominance is best described as the control of the emotional outburst [20].

Using these models, different researchers have examined the nature of emotions found in communications in development processes [4], [10], [11], [13], [21]. A vast majority of these studies are done on online textual communication records present on platforms like GitHub and Jira. Whilst having the benefit of generalization, since the sample is tenfold in comparison to studies that are survey based or studies that used biometrics, etc. the underlaying problem with textual analysis is the unreliability in comparison to methodologies which use facial expressions, physical cues, and body language, etc.

## 2.2. Measuring Emotions in Software Engineering

As discussed earlier, while textual analysis is still debatable when compared to other methods for measuring emotions, it is still fruitful when it comes to providing generalized analysis of developer's emotions in textual communications across a development process. Many studies have been conducted on identification of developer's emotions, distinguish these emotions and how these emotions affect the developer's productivity, decision-making ability, and other skills[1], [11], [12], [22]–[25].

In [5], Jongeling et Al. presented a detailed analysis and evaluation metric which delved into using sentiment analysis tools for software engineering. However, tools like SentiStrength, NLTK, Stanford NLP, etc. were primarily built to analyze the emotions of social media users, or to study consumer's behaviour for different products. Very few to no studies were done that used these tools for the purpose of detecting emotional contagion in commit messages or issue comments by software developers.

## 2.3. Emotional Contagion

Emotional contagion is described as a phenomenon in which there is a mimicry or transfer of emotions from one individual to another [27]. Emotional contagion or the transfer of emotions is widespread in terms of organizational research interest. How emotions affect the very nature of the system is something that has been of interest of to researchers for past several decades. We can safely say that at the foundation of emotional contagion is emotions. Barsade et Al.

deconstructed emotional contagion into four different elements based on previous literature. First, emotions and the generalized term, mood, which has long been termed as an affective state rather than momentarily reaction. Next is the transpire of emotions from one individual to other on conscious level as well as subconscious level, where neither the inducer of the contagion, nor the recipient are aware of the contagion. Third and most important element is defined as emotional contagion being a process and hence can occur between any actors, be it a small group, organizational structure or large social circle, and it can be induced by one or more than one individual. The fourth element defined by the author which has our interest, is that emotional contagion not only defines the emotions of the recipient, i.e., it not only defines how an individual is feeling after the onslaught of contagion but the consequent action they take. This not only affects how individuals perceive the stimulation of emotions that they have been presented with but also their decision-making capabilities, which results in later consequential results.

Though conscious emotional contagion is mainly reported by oneself, in case of subconscious level, it is difficult to find instances. A great example of emotional contagion at a subconscious level facial mimicry. An individual during a conversation, assesses the physical traits of the other individual in order to mimic and respond appropriately. According to the evidence found by Hess et Al. [32], an individual when exposed to an emotional display tends to respond according to the emotional state that matches the one he/she is being exposed to. Facial mimicry or the imitation of physical cues displayed by one are often subconsciously mirrored by the other individual, as the individual is able to judge the situation accordingly. Another important aspect of emotional display has been described by author, decoding the emotional display can be done using varied methods where the sender's cues are used to draw implications about his state of emotions using patter matching approach. Presuming that the facial expressions are best for decoding the emotional state of an individual, it is difficult to assess the situation in case of textual communication where the physical cues are not displayed. In such cases it is up to the recipient to understand the underlying emotions of the text.

Hence, the facial mimicry and physical cues at large can be considered as conscious emotional contagion as the individual or dyad respond to the onslaught of the emotional display according to the situation present, but there is a vast gap in research when it comes to understanding and detecting, emotional contagion in textual communications or written communication.

Emotional contagion in groups where one individual can affect the emotional state of a group of people has been studied by author Barsade et Al. in their study [27]. According to the studies researched by author Barsade et Al. group emotional contagion stimulates behavioural consequences and teamwork, which in turn affects the overall team dynamics.

## 3. EXPERIMENTAL SETUP

As described in the earlier sections, there is a gap in the research on the effects of emotional contagion in software development environment. From past studies, we can safely say that the although the presence of emotional contagion and mimicry are thoroughly explored in other domains of industries, it is high time to look for evidence that can provide reasonable understanding of how emotional contagion affects the software engineering process. To conduct an analysis for finding the evidentiary proof of emotional contagion in software development process, we are working on some base principles and assumptions, so as to keep the scope of the study under control.

Although, as stated in the previous sections, emotional contagion and mimicry are assumed as interchangeable, and part of conscious contagion, in the case of non-verbal communication or textual communication, it is difficult to establish or find evidentiary proof of conscious emotional contagion. Keeping this in mind, in this study we are working under the assumption that the authors or contributors do not have any face-to-face interaction, where they are aware of the physical cues of the sender or recipient. It is important to stress on this aspect of the dataset, as the analysis is purely based on the dataset where the above assumption holds true.

For this purpose, the methodology and workflow described in this study cannot be repeated for a test sample where the above assumption does not hold true. For example, the methodology described in the coming section cannot be used for determining and establishing the evidence of emotional contagion among organizations where verbal communication and face-to-face interactions, including online meetings via sophisticated mediums like Microsoft Teams, Google Meet, Zoom, etc. is used. As mentioned earlier, in the case of such groups or dyads the affect of contagion is different as the participants report with emotional display according to the situation they are presented with.

The second principle of this study is based on recorded changes in a software development process. With the evolution of version control systems, where each step in the process of development is recorded for keeping track of changes, i.e., versions of the software in order to recall previous changes made or compare the different changes made during the entire development process [33]. For the purpose of this study, we will be using GitHub's open-source software projects repositories. GitHub is an online platform for version control system which allows contributors from across the globe to collaborate on projects [34].

The most important principle that we are basing this study on is the locality principle: date and time. For measuring contagion, it is assumed that the perceiver's emotional state prior to being exposed to the expresser's emotional display is neutral. From the time the participant sends a message which expresses their emotional state, to the time the recipient is exposed to this emotional state and subsequently gets affected with the sender's emotional state, hence changing their own state. Here we are working on the assumption that despite the time difference it takes for the recipient to be exposed to the sender's state, there are no other factors that can affect or change the perceiver's affective state. Thus, whatever sentiment polarity is detected from the message from the recipient's message is assumed to be caused by the sender's emotional display in the previous messages.

### 3.1. Mining GitHub's Open-Source Software Repositories for Commit Logs

In order to find and establish a generalized result it was important to consider a sample dataset which consisted of large number of participants. However, gaining access to an organization's communication records containing a large number of employees without in-person communication is difficult. Hence for the purpose of this study we collected our dataset from GitHub's open-source repositories, since open-source software development projects present on the platform provide us with a huge number of participants who collaborated into the development process.

For this study, we used the following four open-source repositories:

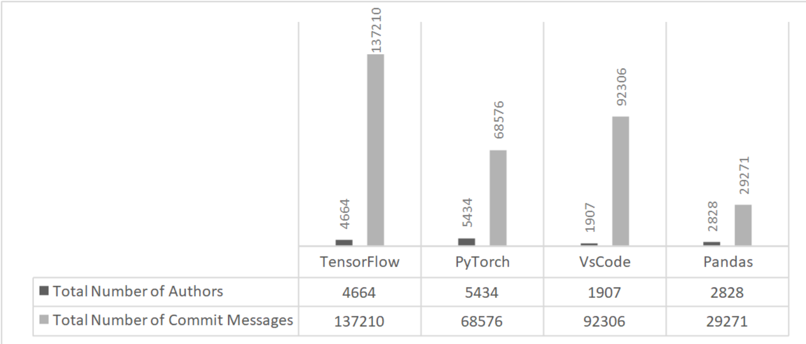



Figure 1: GitHub's open-source software repositories

The mentioned open-source software repositories have more than 14800 contributors from all around the planet. The data retrieval process for the above dataset was fairly straightforward: in order to gather a large amount whilst still keeping the sample size small but sizeable enough to provide generalized results, we made use of Git Bash. Aside from being open-source, GitHub has the added advantage of providing all the needed information, like total authors, total commits, merged issues, as well as the locality principle: date and time.

In order to get the required data from the GitHub repository, we first cloned the repository onto the local machine for further analysis. After cloning the repositories on the local machine, we used Git commands for mining the necessary information we require, the author of the commit messages, date and time of the commit message, body and subject of the message and the commit hash [35]. In particular, we used the Git Log documentation for specific commands that can be used in order to fetch all the necessary details about the commits from the oldest to the newest.

```
rashm@Rashmi-1 MINGW64 ~/2 (main)
$  git clone https://github.com/pandas-dev/pandas
```

Figure 2: Cloning GitHub Pandas repository on local machine

After we had the whole repository cloned on the local machine, we are interested to get the desired information for our dataset, i.e., commit messages by the contributors of each individual repository. For this we used the Git Log command [35]; with git log command we were able to get only the necessary information required for the analysis, for example, git log provides an ample number of options for fetching all the commit messages in different chronological orders with respect to date, authors' date (no subsequent commit will be fetched prior to the main commit, i.e. the parent commit), reverse order, etc. Along with this, the most important feature that came handy while fetching the commit comments were the commit formatting option. Git log provides various formatting option to fetch the contents of the commit logs; in our case we made use of the —pretty format in mining the required information.

```
rashm@Rashmi-1 MINGW64 ~/2/pandas (main)
$ git log --date=format:'%Y-%m-%d %H:%M:%S' --pretty=format:'"%h","%an","%ad","%s","%b"'>pandas.csv
```

Figure 3: Fetching contents for dataset from the commit logs of Pandas

For easier analysis we formatted the data into a readable comma separated file that can be processed by the analytical tool we have used for this study.

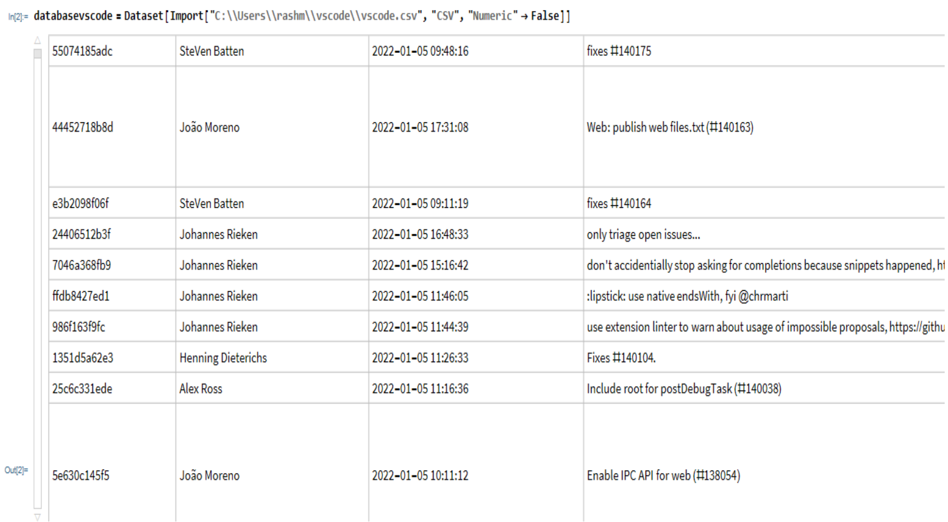
### 3.2. Evaluating The Commit Messages

For the analysis of the sentiments, for this research we adopted Wolfram Mathematica software. Mathematica is a modern-day computing system which supports all the necessary computing area required for this research, as well as image processing, machine learning, visualization, data science and analysis, for a much larger domain of research area from bio science to industry marketing analytics [36]. In comparison to some other software and tools because of the added advantage of having built in functionalities, which has automized and systematic process for all the analytics needs, Mathematica stands out because of its vast knowledge domain application. Mathematica provides several pretrained classifiers for specific classification task in almost every domain possible, for instance natural language processing. These classifiers have been trained and designed for copious amount of data in a variety of fields.

To achieve our task of classifying the sentiments commit messages into positive, negative, neutral we use the Classify function. The classify function works by figuring out the best possible model that has the respectable probability on the test dataset. Initially, possible candidates for the carefully chosen based on the testing dataset for the model, then different models are cross – validated against other models; subsequently, the finest model is selected in order to identify the sentiment of the context. Usually, the input is several lines of text, and the output is one sentiment: positive, negative, neutral.


Figure 4: Classify function

From the following graphs, we can see the total number of commit messages in each repositories. Keeping in mind that the initially fetched dataset is unclean, and hence we removed the log entries where any of the locality principle, in this case the time and commit message body are empty or redundant.



In[2]:= databasevscode = Dataset[Import["C:\\Users\\rashm\\vscode\\vscode.csv", "CSV", "Numeric" → False]]

Out[2]=

| | | | |
|---|---|---|---|
| 55074185adc | SteVen Batten | 2022-01-05 09:48:16 | fixes #140175 |
| 44452718b8d | João Moreno | 2022-01-05 17:31:08 | Web: publish web files.txt (#140163) |
| e3b2098f06f | SteVen Batten | 2022-01-05 09:11:19 | fixes #140164 |
| 24406512b3f | Johannes Rieken | 2022-01-05 16:48:33 | only triage open issues... |
| 7046a368fb9 | Johannes Rieken | 2022-01-05 15:16:42 | don't accidentially stop asking for completions because snippets happened, ht |
| ffdb8427ed1 | Johannes Rieken | 2022-01-05 11:46:05 | :lipstick: use native endsWith, fyi @chrmarti |
| 986f163f9fc | Johannes Rieken | 2022-01-05 11:44:39 | use extension linter to warn about usage of impossible proposals, https://githu |
| 1351d5a62e3 | Henning Dieterichs | 2022-01-05 11:26:33 | Fixes #140104. |
| 25c6c331ede | Alex Ross | 2022-01-05 11:16:36 | Include root for postDebugTask (#140038) |
| 5e630c145f5 | João Moreno | 2022-01-05 10:11:12 | Enable IPC API for web (#138054) |

Figure 5: Fetching the dataset from cloned repository to Mathematica

```
In[6]:= databasevscode = databasevscode[Select[#Message ≠ "" &], All]
```

| Commit | Author | Date | Subject |
|---|---|---|---|
| 44452718b8d | João Moreno | 2022-01-05 17:31:08 | Web: publish web files.txt (#140163) |
| 25c6c331ede | Alex Ross | 2022-01-05 11:16:36 | Include root for postDebugTask (#140038) |
| 5e630c145f5 | João Moreno | 2022-01-05 10:11:12 | Enable IPC API for web (#138054) |

Figure 6: Removing commit logs with anomalies in commit message body

Upon removing these anomalies, we found that there was a huge shift in the total length of these dataset. For instance in case of VsCode repository, the total number of commit logs fetched initially were more than 92k, but upon removing redundant dataset that did not contain any one variable or the variable value was not desirable, or empty subject of body, we got a clean dataset of more than 14k, which is a vast difference from the initial commit log lengths. One of the assumptions behind this difference can be that a majority of commit messages have a subject whilst not a body; these commit logs used to revert back to other commit logs, hence the body of the message often remains empty.

After getting the clean dataset from each repository we used the Classify function to perform sentiment analysis on each commit message body text, the results of which are discussed in the next section.

Upon the sentiment analysis, each value obtained belonged to one class: positive, negative, or neutral. We then converted these values into numeric values of +1 for positive, -1 for negative and 0 for neutral.

```
In[8]:= sentimentvscode = Classify["Sentiment", databasevscode[All, "Message"]]

Out[8]= {Negative, Neutral, Positive, Positive, Negative, Negative, Positive, Negative, Negative, Negative, Neutral, Negative, Positive, Positive, Neutral, Neutral, Neutral,
Neutral, Neutral, Neutral, Positive, Neutral, Neutral, Positive, Negative, Neutral, Neutral, Positive, Neutral, Neutral, Neutral, Neutral, Neutral, ... 14785 ...,
Neutral, Negative, Neutral, Neutral, Neutral, Negative, Negative, Negative, Neutral, Negative, Neutral, Neutral, Neutral, Neutral, Neutral, Neutral, Positive,
Neutral, Positive, Neutral, Neutral, Negative, Negative, Negative, Negative, Neutral, Negative, Neutral, Positive, Positive, Neutral, Neutral, Neutral}

large output | show less | show more | show all | set size limit...
```

Figure 7: Sentiment analysis on the commit messages body

# 4. RESULTS

## 4.1. Evaluating the Presence of Contagion

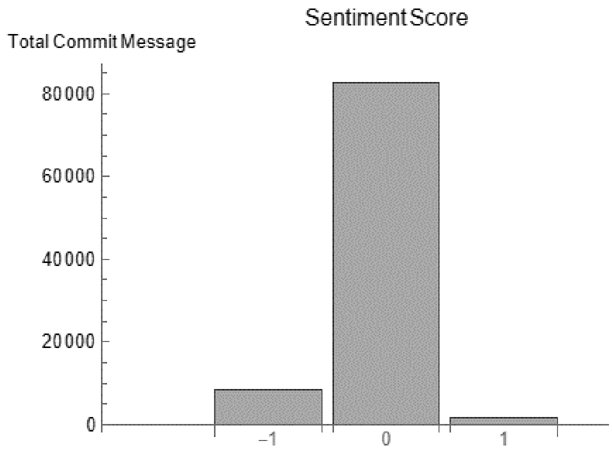



Figure 8(a): Total number of positive, negative, neutral messages in TensorFlow

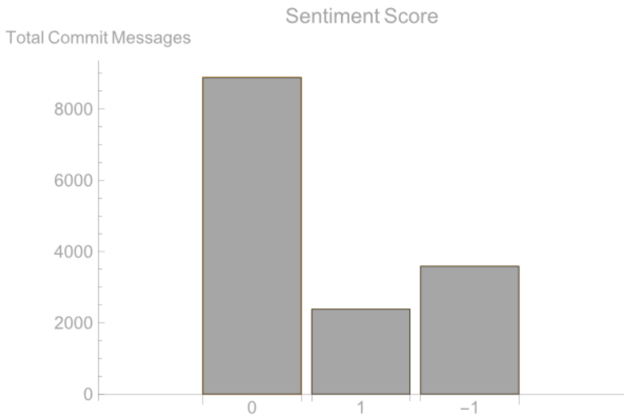



Figure 8(b): Total number of positive, negative, neutral messages in VsCode

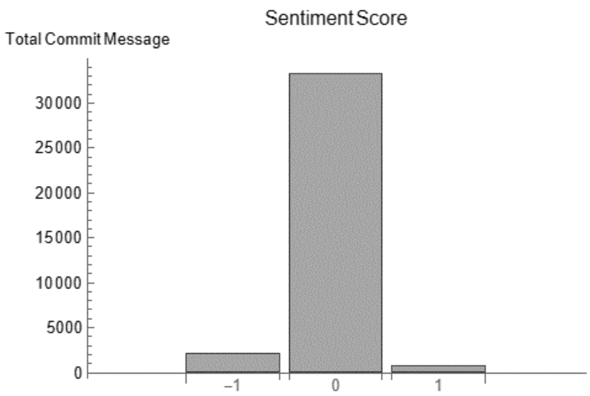



Figure 8(c): Total number of positive, negative, neutral messages in PyTorch

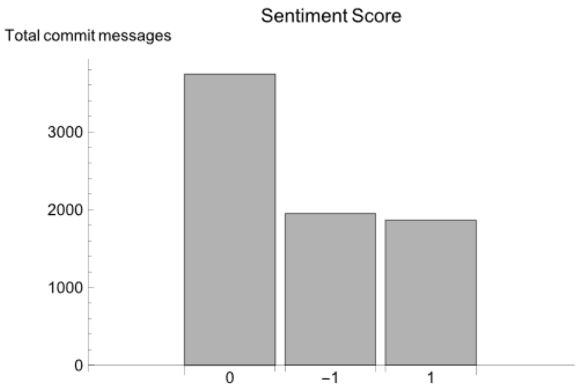



Figure 8(d): Total number of positive, negative, neutral messages in Pandas

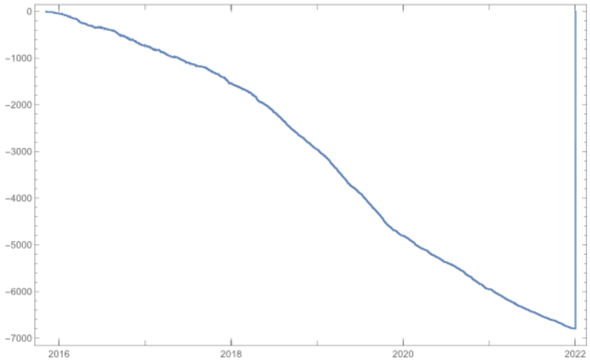



Figure 9(a): Overall Project Affect Graph for TensorFlow

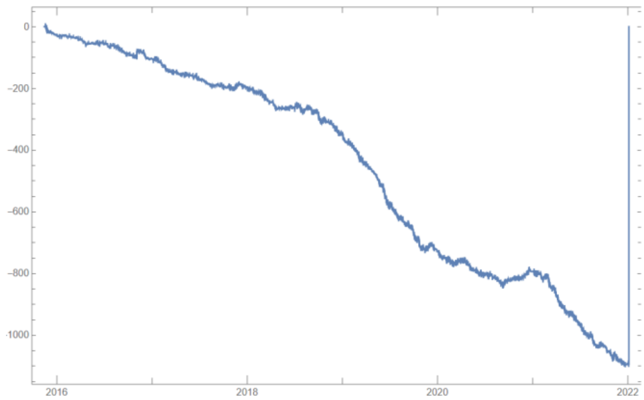



Figure 9(b): Overall Project Affect Graph for VsCode

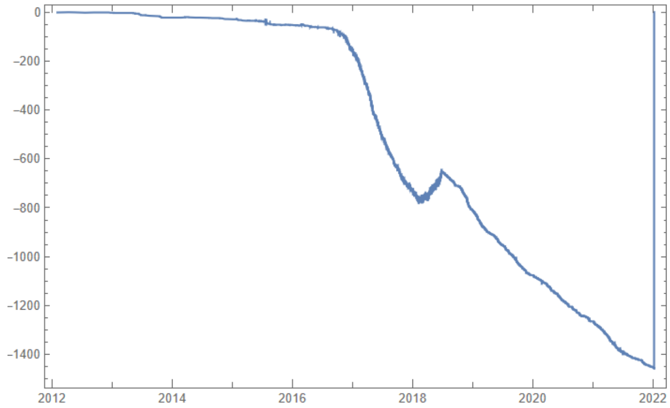



Figure 9(c): Overall Project Affect Graph for PyTorch

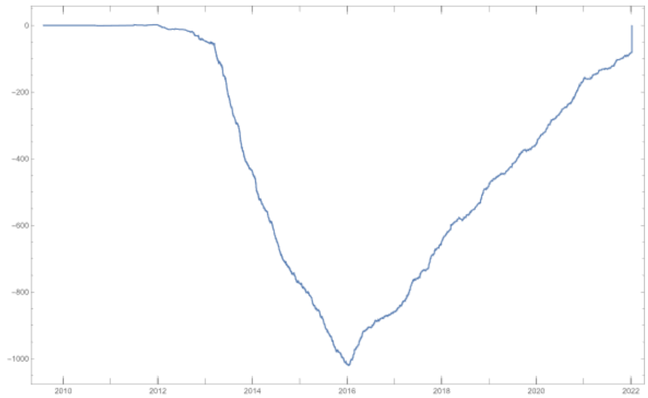


Figure 9(d): Overall Project Affect Graph for Pandas

RQ1: What is the general sentiment of commit messages amongst different size projects? Do projects with a greater number of commits tend to have a greater number of emotional contagion episodes?

From the graph of the sentiment scores of different project repository we can establish that the majority of the commit messages have been neutral in all the cases we have taken into account. In larger project repositories like Tensorflow and VsCode we can see that there is a huge difference in positive and neutral commit messages. Again, we can also see that project repositories where the total number of commits are relatively less, although the neutral commit messages exceed there as well, there is a very slight difference in the number of commit messages which are positive and those which are negative. Again, we can infer from the graph that the project repositories like Tensorflow and VsCode with a relatively large amount of authors have a significant difference in the number of commits which are positive and those which are negative.
From the above graphs we can tell the overall affective graph curve of the repository. For each repository the affect graph is different; for instance, the overall mood in the Tensorflow and VsCode seems to be declining from the start of the project whereas if we talk about Pandas, we can see around the mid term in the year 2016, a positive breakthrough which supports our second hypothesis: a positive comments generates a series of positive comments in the community.

RQ2: Do projects with positive or negative sentimental polarity invoke positive or negative emotional contagion?

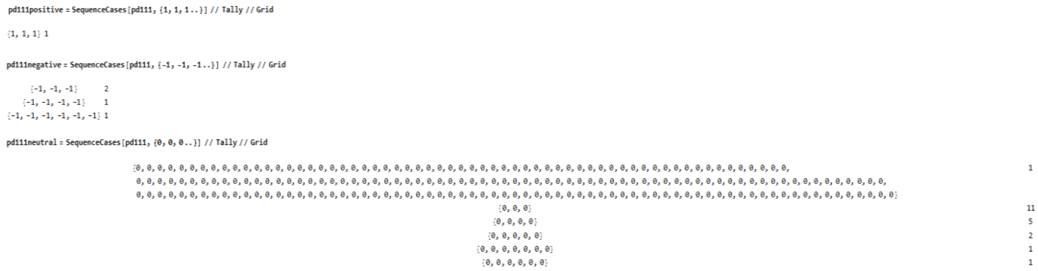



Figure 10: Sequence cases for 500 datapoints in Pandas

In the following graph we see the affect curve for each individual repository where the 500 data points are chosen at random. In order to capture minute fluctuations in the contagion affect we have chosen a small set of datapoints.

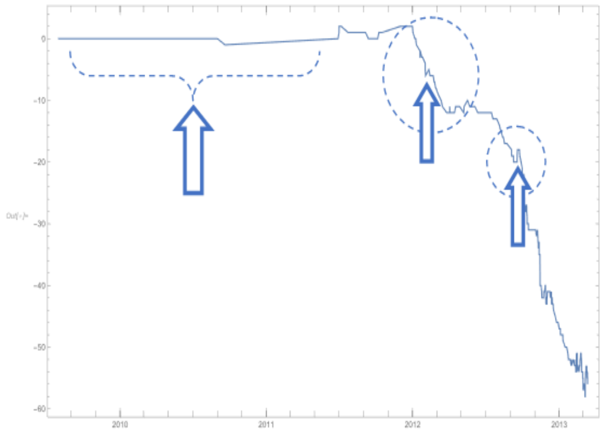



Figure 11(a): Affect Graph of 500 datapoints for Pandas

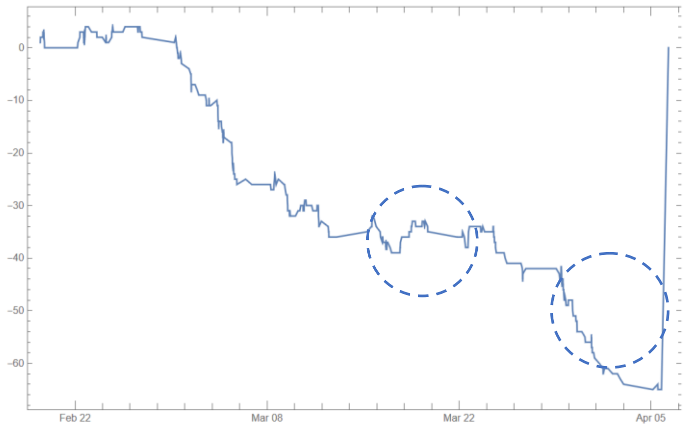



Figure 11(b): Affect Graph of 500 datapoints for VsCode

```
q1ts = TimeSeries[q12, {Normal@q11}]
```

```
TimeSeries[ Time: 30 Oct 2019 to 08 Nov 2019
            Data points: 512 ]
```

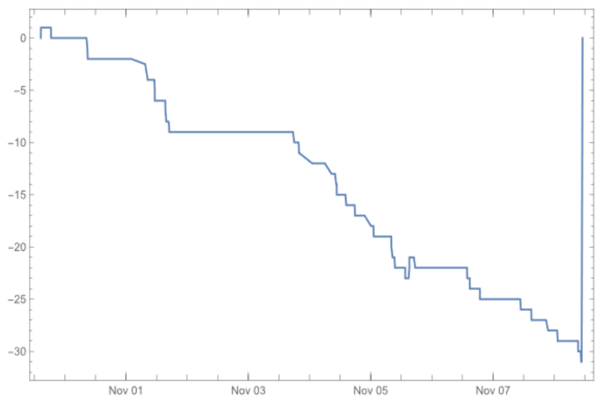



Figure 11(c): Affect Graph of 500 datapoints for TensorFlow

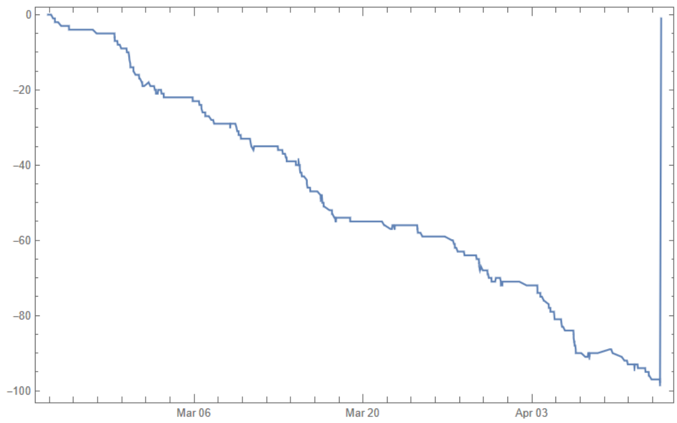



Figure 11(d): Affect Graph of 500 datapoints for PyTorch

From the above graphs we can see the affect curve for randomly chosen 500 data points. It is important to keep in mind the x axis represents the time period for the series whereas the y axis is growth level for the sentiment. From the highlighted area, we can determine that a series of datapoints have a negative accumulation over a period of time; this indicates towards our first hypothesis: a negative comment generates a series of negative contagion in the community.

Also taking into account, for repository VsCode, there is a lot of varying points at which the curve changes; this indicates that instead of having long episode of contagions, we can see small, miniscule set of contagion activity. Another way to theorize this hypothesis is to look at the sequence of sentiment polarities.

| | | | | | |
|---|---|---|---|---|---|
| {0,0,0} | 439 | | | | |
| {0,0,0,0,0} | 160 | | | | |
| {0,0} | 815 | | | | |
| {0} | 1370 | | | | |
| {0,0,0,0,0,0} | 98 | | | | |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 1 | | | | |
| {0,0,0,0,0,0,0,0,0,0} | 16 | | | | |
| {0,0,0,0} | 281 | {1,1,1} | 69 | {−1} | 1897 |
| {0,0,0,0,0,0,0} | 68 | {1} | 1503 | {−1, −1, −1, −1} | 49 |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 1 | {1,1,1,1} | 25 | {−1, −1, −1} | 142 |
| {0,0,0,0,0,0,0,0} | 44 | {1,1} | 312 | {−1, −1} | 475 |
| {0,0,0,0,0,0,0,0,0,0,0} | 13 | {1,1,1,1,1,1,1} | 5 | {−1, −1, −1, −1, −1} | 13 |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 7 | {1,1,1,1,1,1,1,1,1,1,1,1} | 1 | {−1, −1, −1, −1, −1, −1} | 8 |
| {0,0,0,0,0,0,0,0,0} | 41 | {1,1,1,1,1,1} | 2 | {−1, −1, −1, −1, −1, −1, −1, −1} | 2 |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 1 | {1,1,1,1,1,1,1,1,1} | 1 | {−1, −1, −1, −1, −1, −1, −1, −1, −1, −1} | 1 |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 1 | {1,1,1,1,1} | 4 | {−1, −1, −1, −1, −1, −1, −1} | 1 |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 1 | | | | |
| {0,0,0,0,0,0,0,0,0,0,0,0} | 8 | | | | |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 10 | | | | |
| {0,0,0,0,0,0,0,0,0,0,0,0,0} | 4 | | | | |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 1 | | | | |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 2 | | | | |
| {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} | 1 | | | | |

Figure 12: Sequence Cases for 500 random data points in VsCode

Through the help of sequence cases, we can see the general length of contagion for each commit messages. In order to capture the contagion, we can use multiple instances through which we can deduce the presence of contagion, for instance, using pattern recognition to find the patterns where there is a shift in the polarity in the sentiments of the commit messages because of one initial message. Though this process would also provide us with results which are more reliable in constructing our result, the process was not time feasible. For example, for even a small dataset of 20K to 90K, the time complexity to apply the functionality on the dataset and run it on the machine is more than two or three orders of magnitude. Hence instead of this, we chose small set of random datapoints from a repository and examined different sequence cases.

For example, to test our first three hypothesis, we tried to find sequence cases where there is a sequence of positive contagion, i.e., the sequence generated a series of +1 datapoints, similarly for negative and finally for neutral. From the above figure 12 we can see for neutral contagion, there are long sequences, every sequence where the number of sub list elements are more than 2; for example a sequence like: {0,0,0} or {0,0,0,0,0,0,0,0,0,0,0,0,0,0} can be termed as part of a neutral contagion, since a sequence less than 3 elements can merely be result of individual sentiment polarity. We do not consider those cases. Similar to this, if we look for cases of positive contagion, we see total positive contagion episodes are a little over 100, whereas if we talk about negative contagion, we see that there are over 200 cases of sequences. This suggests that negative contagion episodes are far more prevalent in comparison to positive. This not only gives rise to conflicts among the development environment but also leads to merge issues in case of open-source software development.

RQ3: Does having a greater number of neutral sentiment polarity instances predict a greater number of neutral contagion episodes?

The contagion affect of neutral commit messages is still a bit unclear, as we can see from the graphs. In the case of instances where there are a number of neutral datapoints, the curve of the graph seems unchanged, despite having huge number of sequences where neutral contagion is seen. This rather supports the third hypothesis: a neutral response does not affect the curve at large.

In the coming section we will take a look at the affect curve for the most committed author in PyTorch repository. In case of the neutral contagion, one of the key inferences that we came across concerns an author with a smaller number of opposite polarity, i.e. negative or positive polarity. We infer that despite being exposed to a contagion episode the affective curve remains unchanged at mass. If we look at figure 13(a), we can see in case of the most committed author in PyTorch the majority of the commit messages are neutral.

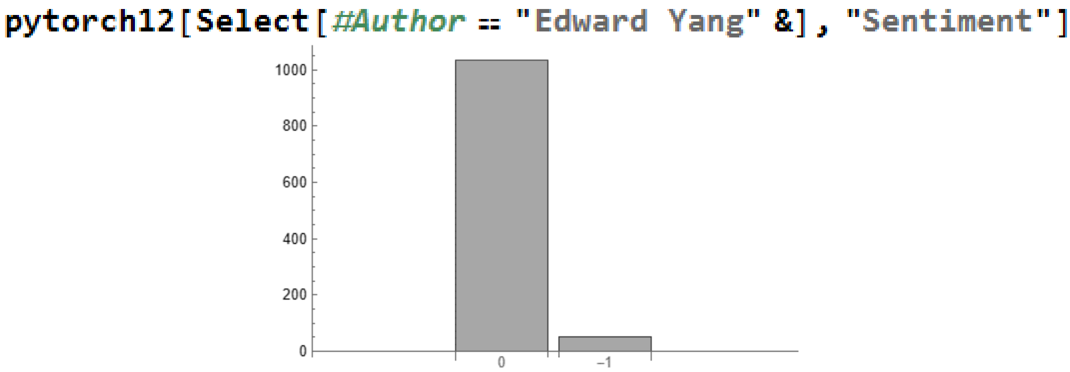



Figure 13(a): Sentiment Polarity of Most Committed Author in Pytorch

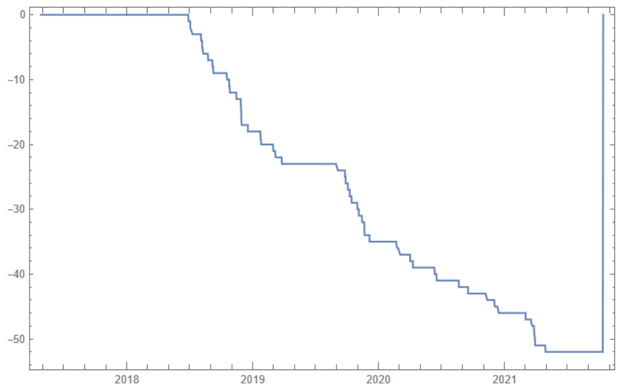



Figure 13(b): Affective curve for the most committed author in Pytorch

RQ4: Does different day of the week have different number of contagion effect, i.e. are there more number of contagious episodes at the start of the week compared to the weekend?

Another inference that we made regarding the fourth hypothesis is based on the day of the week. We found that the number of commits posted are differ from one repository to another; for example, for Pytorch, the greatest number of commits were observed on Wednesday, where as for VsCode, most commits are on Monday. Hence to infer if the day of the week has a significance in the change in affective curve is a bit difficult at this stage, since the data are scattered according to the time frame, and we are working under the impression that each commit message's contagious affect is linked to the next. We established the affective graph of PyTorch in the Figure 14 (c), we infer that since Wednesday has the greatest number of commit messages their emotional contagion episodes are observed the most during this day. We can also infer that despite being the day in the mid of the week, we observed a lot more commit messages which we initially hypothesized to be related to the start or the end of the week. Though from the observed statistics, we can see that on weekends we observe the least number of commits, and least commits with polarity episodes.

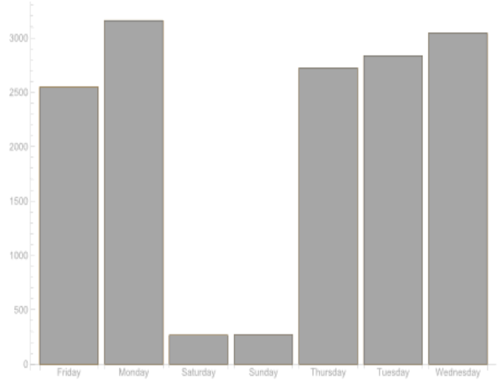



Figure 14 (a): Most number of commits on different days for VsCode

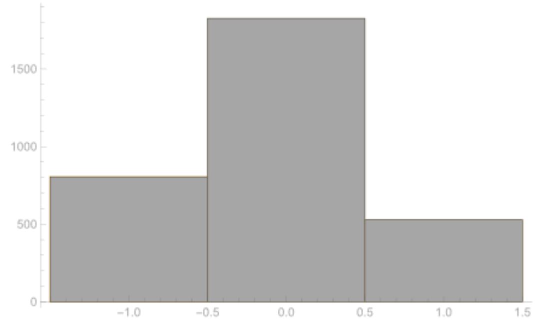



Figure 14(b): Polarity index of all the commits on the most committed day

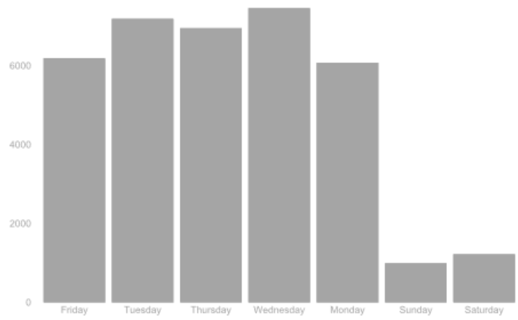



Figure 14(c): Most number of commits for Pytorch on different days

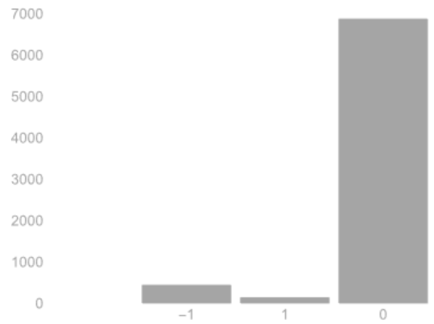



Figure 14(d): Polarity index of all the commits on the most committed day

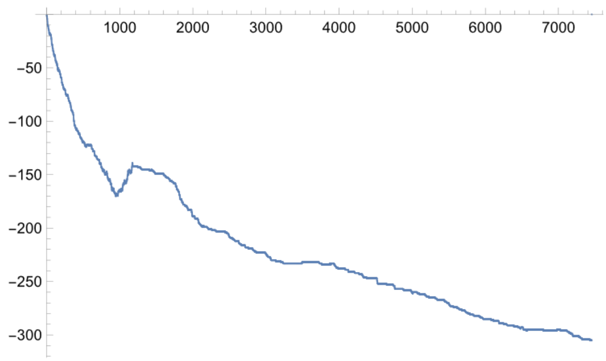



Figure 14(e): Affective graph of all commit messages on Wednesday in PyTorch

## 5. DISCUSSION AND THREAT TO VALIDITY

From the studies conducted earlier, it is clear that emotional contagion is a phenomenon that can affect the individual subconsciously or consciously. Moreover, the expresser of an emotional state often can be considered as a generator for the contagion, who could be aware of the phenomenon that he is generating or could not be, as in the case of verbal or face-to-face communication. In such scenarios it has been observed by researchers that individuals often tend to express their emotions in keeping with the nature of the in the social situation they are exposed to. The problem occurs in nonverbal communication where the perceiver is not aware of the intended emotional expression of the message. Often what the initial intent to the expresser was in the message gets lost because of not having any other surrounding scenario that helps the perceiver understand the intent of the message. Other factors also affect the sentiment of the subsequent messages. As we mentioned earlier, in this study, we assumed that the perceiver's emotional state at any given point is neutral, i.e., the only change affecting the response of the perceiver is that of the preceding response from the expresser.

Regarding the hypothesis presented in this study, we found multiple affect cases that supports the following conclusions:

1. A negative response or a positive response generates a series of negative or positive response.
2. Neutral sentiment polarity does not affect the overall dynamics of the affective curve though it is still considered a contagion, as it indicates not deflecting from the initial emotional state.
3. Contributors tend to have a major shift in their affective state in the beginning of the week and thus leads to a greater number of commits in comparison to the weekend, when the response is negligibly small.
4. Contributors with a greater number of negative responses tend to have more commits in comparison to the contributors whose affective curve lies on the positive side.

Coming towards the affect of emotional contagion, we have studied various other studies by numerous researchers who talk about the ill affects of having negative emotions in the development environment. Graziotin et Al. have stressed through their studies[4], [11]–[13], that negative affective state not only results in poor mental state, productivity but also affects the performance and cognitive skills of the developers. A developer with negative affective state tends to take easier methods for critical problem solving which leads to poor code quality.

Also, a positive environment is required not only from the contributors' side but from the overall management. A poor environment results in more severe consequences and poor results.
From the aspect of validity, this study is heavily dependent on the size of the dataset available and the locality principle of time and contributors' commit logs. For each dataset it was vital to have commit logs where there were no anomalies and some inferences about the emotional display can be extracted.

We were reliant on the use of the software Wolfram Mathematica for the initial analysis; though useful, the results strongly are dependent on the software's algorithms. In order to get a generalized and higher validity we can replicate the study to achieve more success.

## 6. CONCLUSION

This provided a methodology to determine the presence of emotional contagion in open-source software development. The goal was to test out the contagion hypothesis and provide evidentiary results that can be explained and used for the consequential results in future. The instances and illustrations provided in this study were limited to the scope of the assumptions and principles we first described.

Emotional contagion is considered as a conscious and subconscious phenomenon which requires deeper understanding in the field of software development. This study mainly focused on the nonverbal aspect of communication in the development environment. Though repeatable, this methodology is required to be used for verbal communication and environments where there are physical cues are observable by the contributors. By combining facial expressions, pattern matching and machine learning model, we will be able to create a more sophisticated system that is able to provide results with less complexity. By developing such a model, we will be able to detect the contagion in a much shorter time frame and provide the developers with better tools to work with.

## REFERENCES

[1] D. Graziotin, X. Wang, and P. Abrahamsson, "Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering," *J. Softw. Evol. Process*, vol. 27, no. 7, pp. 467–487, 2015, doi: 10.1002/smr.1673.

[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*, Hyderabad, India, May 2014, pp. 352–355. doi: 10.1145/2597073.2597118.

[3] J. Ding, H. Sun, X. Wang, and X. Liu, "Entity-level sentiment analysis of issue comments," in *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, New York, NY, USA, Jun. 2018, pp. 7–13. doi: 10.1145/3194932.3194935.

[4] M. V. Mäntylä, D. Graziotin, and M. Kuutila, "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers," *Computer Science Review*, vol. 27, pp. 16–32, Feb. 2018, doi: 10.1016/j.cosrev.2017.10.002.

[5] R. Jongeling, S. Datta, and A. Serebrenik, "Choosing your weapons: On sentiment analysis tools for software engineering research," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Sep. 2015, pp. 531–535. doi: 10.1109/ICSM.2015.7332508.

[6] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, "On negative results when using sentiment analysis tools for software engineering research," *Empir Software Eng*, vol. 22, no. 5, pp. 2543–2584, Oct. 2017, doi: 10.1007/s10664-016-9493-x.

[7] M. Obaidi and J. Klünder, "Development and Application of Sentiment Analysis Tools in Software Engineering: A Systematic Literature Review," in *Evaluation and Assessment in Software Engineering*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 80–89. http://doi.org/10.1145/3463274.3463328

[8] A. Qazi, R. G. Raj, G. Hardaker, and C. Standing, "A systematic literature review on opinion types and sentiment analysis techniques: Tasks and challenges," *Internet Research*, vol. 27, no. 3, pp. 608–630, Jan. 2017, doi: 10.1108/IntR-04-2016-0086.

[9] S. G. Barsade, "The Ripple Effect: Emotional Contagion in Groups," Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 250894, Oct. 2000. doi: 10.2139/ssrn.250894.

[10] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "Consequences of Unhappiness while Developing Software," in *2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, May 2017, pp. 42–47. doi: 10.1109/SEmotion.2017.5.

[11] D. Graziotin, X. Wang, and P. Abrahamsson, "Are Happy Developers More Productive?," in *Product-Focused Software Process Improvement*, Berlin, Heidelberg, 2013, pp. 50–64. doi: 10.1007/978-3-642-39259-7_7.

[12] D. Graziotin, X. Wang, and P. Abrahamsson, "Software Developers, Moods, Emotions, and Performance," *IEEE Softw.*, vol. 31, no. 4, pp. 24–27, Jul. 2014, doi: 10.1109/MS.2014.94.

[13] D. Graziotin, X. Wang, and P. Abrahamsson, "Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering," in *Proceedings of the 7th International Workshop on Social Software Engineering*, Bergamo, Italy, Sep. 2015, pp. 25–32. doi: 10.1145/2804381.2804386.

[14] R. Dhakad and L. Benedicenti, "Detecting Emotional Contagion in OSS Projects," presented at the The 7th World Congress on Electrical Engineering and Computer Systems and Science, Aug. 2021. doi: 10.11159/cist21.301.

[15] "The GHTorrent project." https://ghtorrent.org/

[16] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment Polarity Detection for Software Development," *Empir Software Eng*, vol. 23, no. 3, pp. 1352–1382, Jun. 2018, doi: 10.1007/s10664-017-9546-9.

[17] N. Novielli, F. Calefato, and F. Lanubile, "Towards discovering the role of emotions in stack overflow," in *Proceedings of the 6th International Workshop on Social Software Engineering*, Hong Kong, China, Nov. 2014, pp. 33–36. doi: 10.1145/2661685.2661689.

[18] D. Girardi, F. Lanubile, N. Novielli, L. Quaranta, and A. Serebrenik, "Towards recognizing the emotions of developers using biometrics: the design of a field study," in *Proceedings of the 4th International Workshop on Emotion Awareness in Software Engineering*, Montreal, Quebec, Canada, May 2019, pp. 13–16. doi: 10.1109/SEmotion.2019.00010.

[19] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3–4, pp. 169–200, May 1992, doi: 10.1080/02699939208411068.

[20] A. S. Cowen and D. Keltner, "Self-report captures 27 distinct categories of emotion bridged by continuous gradients," *PNAS*, vol. 114, no. 38, pp. E7900–E7909, Sep. 2017, doi: 10.1073/pnas.1702247114.

[21] V. Sinha, "SENTIMENT ANALYSIS ON JAVA SOURCE CODE IN LARGE SOFTWARE REPOSITORIES," p. 71.

[22] A. Alhubaishy and L. Benedicenti, "Toward a Model of Emotion Influences on Agile Decision Making," in *2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, May 2017, pp. 48–51. doi: 10.1109/SEmotion.2017.7.

[23] M. R. Anany, H. M. W. Hussein, and S. G. Aly, "A survey on the influence of developer emotions on software coding productivity," *IJSHC*, vol. 3, no. 3/4, p. 216, 2020, doi: 10.1504/IJSHC.2020.111166.

[24] B. Crawford, R. Soto, C. L. de la Barra, K. Crawford, and E. Olguín, "The Influence of Emotions on Productivity in Software Engineering," in *HCI International 2014 - Posters' Extended Abstracts*, Cham, 2014, pp. 307–310. doi: 10.1007/978-3-319-07857-1_54.

[25] I. A. Khan, W.-P. Brinkman, and R. M. Hierons, "Do moods affect programmers' debug performance?," *Cogn Tech Work*, vol. 13, no. 4, pp. 245–258, Nov. 2011, doi: 10.1007/s10111-010-0164-1.

[26] A. Alhubaishy and L. Benedicenti, "Toward a Model of Emotional Contagion Influence on Agile Development for Mission Critical Systems," in *2017 International Conference on High Performance Computing Simulation (HPCS)*, Jul. 2017, pp. 541–544. doi: 10.1109/HPCS.2017.86.

[27] S. G. Barsade, C. G. V. Coutifaris, and J. Pillemer, "Emotional contagion in organizational life," *Research in Organizational Behavior*, vol. 38, pp. 137–151, Jan. 2018, doi: 10.1016/j.riob.2018.11.005.

[28] J. Schneider, O. Müller, and K. Bösch, *EMOTIONAL CONTAGION THROUGH ONLINE NEWSPAPERS*. 2018.

[29] X. Wang, D. Jin, M. Liu, D. He, K. Musial, and J. Dang, "Emotional Contagion-Based Social Sentiment Mining in Social Networks by Introducing Network Communities," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, New York, NY, USA, Nov. 2019, pp. 1763–1772. doi: 10.1145/3357384.3357941.

[30] W. Xiao, L. Wen-zhong, and D. Jian-gang, "The development research of the emotional contagion theory," in *2010 IEEE International Conference on Software Engineering and Service Sciences*, Jul. 2010, pp. 628–632. doi: 10.1109/ICSESS.2010.5552269.

[31] A. Goldenberg and J. J. Gross, "Digital Emotion Contagion," *Trends Cogn. Sci.*, vol. 24, no. 4, pp. 316–328, Apr. 2020, doi: 10.1016/j.tics.2020.01.009.

[32] U. Hess and S. Blairy, "Facial mimicry and emotional contagion to dynamic emotional facial expressions and their influence on decoding accuracy," *International Journal of Psychophysiology*, vol. 40, no. 2, pp. 129–141, Mar. 2001, doi: 10.1016/S0167-8760(00)00161-6.

[33] "Git - About Version Control." https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

[34] "GitHub.com Help Documentation," *GitHub Docs*. https://docs.github.com/_next/data/5OaLy0Xft1a1F62xoVr4z/en.json

[35] "Git - git-log Documentation." https://git-scm.com/docs/git-log

[36] "Wolfram Mathematica: Modern Technical Computing." https://www.wolfram.com/mathematica/

## AUTHORS

**Dr. Luigi Benedicenti** is a full professor and dean in the Faculty of Computer Science at the University of New Brunswick. Benedicenti received his Laurea in Electrical Engineering and Ph.D. in Electrical and Computer Engineering from the University of Genoa, Italy. He is a Professional Engineer licensed in Saskatchewan and a licensed Italian Engineer. His collaborative network extends beyond Saskatchewan with TRLabs and IEEE, and Canada through collaborative work with colleagues in Europe, South East Asia, and North America.

Benedicenti's current research is in three areas: Software Agents, Software Process, and New Media Technology. He envisions the unification of platform, tools, and optimizations for the provision of persistent distributed digital services, regardless of people's location and delivery device.

**Rashmi Dhakad** is a full-time scholar at the Faculty of Computer Science at the University of New Brunswick. She received her bachelor's in technology majoring in Computer Engineering from Rajasthan Technical University, India. She has published previous work at CIST 2021, for which she was awarded best paper. Currently she is working with Dr. Benedicenti on determining the affects of emotional contagion on software development processes in large distributed open-source repositories.