Mining Reddit as a New Source for Software Requirements

Tahira Iqbal
University of Tartu
Tartu, Estonia
LMU
Munich, Germany
tahira.iqbal@ut.ee

Moniba Khan
NUST
Islamabad, Pakistan
moniba.khan@mcs.edu.pk

Kuldar Taveter
Univerity of Tartu
Tartu, Estonia
kuldar.taveter@ut.ee

Norbert Seyff
FHNW
Windisch, Switzerland
University of Zurich
Zurich, Switzerland
norbert.seyff@fhnw.ch

***Abstract*—Mining app stores and social media has proven to be a good source for collecting user feedback to foster requirements engineering and software evolution. Recent literature on mining software-related data from social platforms, such as Twitter and Facebook, shows that it complements app store mining. However, there are many other platforms where users discuss and provide feedback on software applications that are not thoroughly researched and analysed. One of such platforms is reddit. In this paper, we introduce reddit as a new potential data source and explore if and how requirements engineering and software evolution can benefit from obtaining user feedback from reddit. We also present an exploratory study in which we analysed the usage characteristics (i.e., frequency of posts, number of comments, and number of users for each subreddit) of reddit posts about software applications. Furthermore, we examined the content of the posts and the results reveal that almost 54% of posts contain useful information. Finally, we investigated the potential of automatic classification and applied machine learning algorithms to unstructured and noisy reddit data to perform automated classification into the categories of bug reports, feature related, and irrelevant. We found that the Support Vector Machine algorithm with the F1-score of 84% can be effective in categorizing reddit posts. Our results show that reddit posts provide useful feedback on software applications that can foster requirements engineering and software evolution.**

***Index Terms*—Requirements elicitation, app store mining, software feature mapping, crowd data, machine learning**

I. Introduction

In software development, user involvement is an important factor that contributes to system success [1, 2, 3]. User feedback can be employed as a starting point for identifying requirements [4] and can support software evolution. For example, it can provide input for release planning by helping practitioners to choose and prioritize the most desired features [5]. Considering users in requirements engineering and software evolution activities can be a challenging and time-consuming task due to their limited accessibility and availability [6]. Previous research has proposed mining of user feedback provided in multiple platforms and forums such as the app stores and Twitter [7, 8] as one of the solutions to reduce issues related to user involvement. These studies [7, 8] have shown that online user feedback forums provide a good understanding of user needs by integrating a wider range of user perspectives and help to improve software quality. Such user feedback contains valuable information regarding bug reports and feature requests [9, 10].

Several online forums contain a lot of user-generated data that includes valuable information about software applications and can be utilized to identify and update user requirements and facilitate software evolution. However, the existing literature focuses mainly on Twitter, Facebook and app stores. Therefore, it would be interesting to investigate and explore additional new data sources to obtain user feedback about software applications. One example of this kind of forum is reddit[1] – a network of a large community based on people's interests. At the end of 2020, reddit had more than 52 million daily active users[2] and was the 18th most popular website globally[3]. Reddit hosts online communities known as subreddits on user-generated topics. Members post in subreddits in formats, such as links, text posts, videos, and images, which are then commented, and voted up or down by other members. For example, the subreddit of Slack[4] helps to connect users of Slack, and users post regarding features, problems (bugs), and questions related to Slack. An example of a Slack subreddit post is: *"slack needs hyperlink support in markup mode?"*. From this post one can infer a feature request from the user concerning the hyperlink support in the markup mode of Slack. Another example of a post *"Can't open slack in Firefox on windows anymore"* can be interpreted as bug report. Members of the Slack subreddits can perform different actions on posts of this kind, such as comments, likes, dislikes, and sharing.

Reddit has been mined and analyzed for various purposes in different fields, e.g., to better understand health-related topics and identify symptoms of diseases and conditions like anxiety [11, 12]. Another research area where reddit has been applied is social sciences that have used reddit data for several tasks, including understanding community practices, misogyny

The research reported in this publication has been funded by the IT Academy program of the European Social Fund, H2020 Innovation Action "Pilots for Healthy and Active Ageing" funded by the EU Commission, and European Research Council (ERC AMPLIFY, no. 683008).

[1] https://www.reddit.com/
[2] https://redditblog.com/2020/12/08/reddits-2020-year-in-review/
[3] https://www.alexa.com/topsites
[4] https://www.reddit.com/r/Slack/

2332-6441/21/$31.00 ©2021 IEEE
DOI 10.1109/RE51729.2021.00019

detection, and measuring the impact of political propaganda [13, 14].

The large amount of information found on reddit made us curious about the relevance of this information for requirements engineering and software evolution. We first performed a quick search as a proof of concept, which revealed that reddit also contains software-related data. Software applications often own their official subreddits or might have subreddits generated by users providing relevant data. Furthermore, Kanchev et al. [15] report on software-related information found on reddit. For applications with a larger number of posts on reddit like Google Maps or applications having multiple subreddits it might not be possible to perform a manual analysis of posts. However, to the authors' knowledge, no study has yet been conducted that investigated the potential of automatic classification of reddit posts to support requirements engineering and software evolution.

In this paper, we present the results of an exploratory study that we performed to better understand reddit data on software applications. We examined the usefulness and relevance of the reddit data for requirements engineering and software evolution. For this purpose, we collected data about widely used software applications and analyzed data characteristics, such as frequency of posts, and number of comments and users for each subreddit. After determining that the reddit data contains a decent amount of relevant data, we continued with an in-depth analysis. We manually classified 1915 posts into the categories *feature related (e.g., a feature request), bug reports, and irrelevant* for requirements engineering and software evolution. Our manual analysis showed that 54% of the posts were informative and classified as *feature related* or *bug reports*. In a final step, we applied machine learning algorithms to classify reddit posts in order to investigate the possibility of automatic classification. We examined the Support Vector Machine (SVM) as one of the most suitable algorithms for the classification of reddit posts and achieved the 84% F1-score.

The rest of this paper is organized as follows: In Section II, we present our research questions and approach. Section III discusses the reddit platform and data collection from reddit. It furthermore presents the results of a quantitative analysis of the collected data. Section IV presents the qualitative analysis of 1915 reddit posts and discusses their usefulness for requirements engineering and software evolution. Section V presents our work on automated classification of informative posts and evaluates the performance of various machine learning algorithms in different settings. Section VI provides and explains answers to the research questions and discusses associated threats to validity. Section VII outlines related work and Section VIII draws the conclusions and presents future work.

II. Research Questions and Approach

The main research goal of our study is to investigate whether and how requirements engineering and software evolution can benefit from obtaining user feedback data from reddit. Our initial scanning of subreddits dealing with software applications indicated that there is a decent amount of data, but not every post could be useful for requirements engineering and software evolution. Motivated by this insight, we collected posts from software related subreddits. Our research work has addressed the following research questions:

RQ1: What are the characteristics of data present in software related subreddits?
We aim to investigate characteristics of reddit posts in software related subreddits. For this purpose, we performed a quantitative evaluation of software related subreddits and analyzed different factors, such as the total numbers of users, frequencies of posts, and user responses to such posts. The analysis helps to know user engagements and data characteristics in software related subreddits.

RQ2: Does reddit data provide meaningful information for requirements engineering and software evolution?
Our initial scanning of subreddits related to software applications indicated that there is a decent amount of data available, but each post might not be useful for the software evolution process. There can be posts that have no software related information but have been posted in the subreddits of software applications. As the reddit data has proven to be a useful information source in other domains, we investigated if it contains valuable information about software applications. Therefore, we performed a qualitative evaluation and analysed reddit posts and categorized them into meaningful categories including *feature related*, *bug reports*, and *irrelevant*. We have chosen these categories because this classification has proven to be helpful for software requirements engineering and evolution processes [9, 10].

RQ3: To what extent can informative software-relevant reddit posts be automatically classified?
Assuming software-relevant reddit posts contain useful information, manual filtration of informative posts from a massive amount of non-informative reddit posts is challenging, error prone, and time-consuming. A practical solution can be an automated approach with a decent level of accuracy that can filter informative and non-informative reddit posts. Therefore, we investigated how the existing machine learning (ML) algorithms can help to build an automated approach. We applied different ML algorithms and discussed data preparation, ML model training, and model evaluation.

III. Data Collection and Quantitative Analysis

In this section, we first briefly discuss reddit platform and about the data available on reddit. This is followed by describing how we collected data from reddit. Finally, the results of the quantitative analysis of the reddit data are presented. This analysis was conducted to provide answers to RQ1.

TABLE I: Details of subreddits on software applications

Subreddit Name	Domain	Members (in thousands)	Number of posts
Adobe illustrator	Photography	73.9	599
Amazon	Shopping	148	2038
Microsoft office	Productivity	1.0	2285
Google Chrome	Communication	78.8	102
Microsoft Word	Productivity	1.2	2285
Netflix	Entertainment	516	965
Dropbox	Productivity	4.8	1216
Adobe Photoshop	Photography	166	368
Excel	Productivity	203	858
Power point	Productivity	7.2	1764
Facebook	Social	36	246
Adobe Premiere	Video Players & Editors	44.2	579
Google Maps	Travel & Local	28.9	3904
Safari	Communication	1.8	817
In Design	Art & Design	16.3	2008
Slack	Business	11.2	1868
Instagram	Social	135	595
Spotify	Music & Audio	232	1644
WhatsApp	Communication	14.1	1241
YouTube	Video Players & Editors	447	412

A. Reddit

Almost 38% of reddit users consider themselves as tech savvy, which is more than users of other social platforms like Facebook, Instagram, and Twitter[5]. This is a positive indication of useful insights for software-related products. There are millions of subreddits covering a wide range of topics. These subreddits can be either public or private, and can be run by moderators or representatives of companies. Users can write posts a.k.a. submissions in the form of text, image, video, or link submissions. Each post is characterised by the user ID, date, title, and selftext. Selftext is a detailed description that supports the title of the post. It is mandatory to have some text in the title of the post; selftext is optional and can be left empty. The character limits for the title and selftext are 300 and 40,000. Users can perform several actions on a post such as an upvote, downvote, share, comment, save, and report. Another parameter named as score is calculated for each post based on the ratio between upvotes and downvotes. Additionally, subreddit posts can be sorted in the best/hot, new, and top categories. The category best has the highest upvote to downvote ratio, the category top has the largest number of votes (upvotes – downvotes), and the category hot has the largest number of recent upvotes on posts.

B. Data collection

For our study, we collected reddit posts from March 2019 March 2020 for the top ten mobile and desktop applications. For mobile applications, we included applications from the App Annie list[6], and for desktop applications from the Microsoft store list[7]. As the first step in creating our dataset, we manually searched for top software applications from the lists in reddit to check if a list had a subreddit or not. We excluded software applications from the later analysis for the following reasons: 1) there was no subreddit for the application 2) the subreddit had less than one thousand members. We chose 1000 members as a threshold because based on our initial analysis we observed that subreddits with fewer than one thousand members often were not active and had a negligible number of posts. Furthermore, we did not see a need to investigate such subbredits in detail as other subbredits had a much higher number of members. As a result of applying these constraints, we excluded Tiktok, Share It, Facebook Messenger, Snapchat, and iTunes. The excluded applications were replaced with subsequent applications from the obtained lists.

For the selected 20 applications with an official subreddit[8] (see Table I), we also searched for user-created subbreddits. For this, we searched reddit with the application name. We examined the top three search results to understand if these subreddits are active and the content is relevant. As a result, for 6 out of the 20 applications, we included one user-generated subreddit based on our analysis. These six software applications were Netflix, Google Maps, Amazon, Excel, Spotify, and Whatsapp.

For importing posts from the chosen subreddits, we used the Python library PRAW[9] to accesses the reddit's API.

C. Results

In total, we retrieved 25,794 reddit posts along with 891,066 comments from the 20 software applications' subreddits, shown in Table I. Our dataset covered more than 10 application domains, such as social, productivity, and entertainment. The average frequency of posts per software application in our dataset is 1,289.7 (median = 1,090.5). On average, each software subreddit in our dataset has 108,320 members (median= 40,100). The highest number of posts is 3904 for the Google Maps, and the lowest number of posts is 102 for the Google Chrome subreddit. In our dataset, Netflix subreddit is

[5] https://foundationinc.co/lab/reddit-statistics/

[6] https://www.appannie.com/

[7] https://www.microsoft.com/en-us/store/most-popular/apps/pc

[8] We consider the subreddit official if the software application name and subreddit name are identical.

[9] https://praw.readthedocs.io/en/latest/

TABLE II: Examples of reddit posts with assigned categories

Application	Title	SelfText	Category
Whatsapp	Message Scheduling	I am unable to understand why WhatsApp (the most popular messaging app) still doesn't have a scheduling feature which is such a basic and handy and useful feature. There are a lot of times when we have to message someone at a specific time	feature related
Whatsapp	Dear WhatsApp, These descriptions are supposed to give us info about the new update. Not a same sentence over and over again		feature related
Whatsapp	WhatsApp Desktop Failing?	Earlier today my desktop app started to tell me to update manually. But despite my efforts, it doesn't go back to normal. Only shows that screen. I even reinstalled the app. Has it happened to any of you?	bug report
Whatsapp	WHO Health Alert brings COVID-19 facts to billions via WhatsApp		irrelevant
Spotify	Spotify shuffle algorithm is not good	Whenever I shuffle my playlist I get like the same artist 5 songs in a row or it seems to just keep going in order from when I added the songs to the playlist with minor differences. My playlist has over 500 songs so it seems unlikely that this would continue to happen but I consistently get shuffles like this	feature related
Spotify	Dear Spotify, your shuffle songs algorithm is hot garbage		feature related
Spotify	Spotify needs to add a queue playlist feature	I don't know if anyone else feels this way but it would be convenient if Spotify made it where we can just queue a playlist. I have a few playlists that are just an hour-long so it's not uncommon for me to listen through 1-2 playlists at a time just be easier if I could make them roll into each other. Edit: I didn't clarify that I use an android, so if this isn't an issue for you apparently Spotify provides that feature for some other devices outside of android.	feature related
Spotify	Anyone having issues loading artist pages right now?	Currently having troubles on desktop, web browser and mobile versions	bug report

the largest subreddit with 516,000 members, while Microsoft Word subreddit is the smallest subreddit with 1,000 members.

Table I shows that there is no direct relationship between the number of members of a subreddit and the number of posts in the subreddit. For example, the Adobe Illustrator and Youtube subreddits have the highest numbers of members but do not have comparable numbers of posts. On the contrary, the subreddits for Microsoft Word and PowerPoint have more posts in comparison to their members. Fig.1 shows the calculated frequencies[10] for calculating the numbers of posts per day and comments per day for all subreddits, including categories for each post (top, hot, and new).

IV. QUALITATIVE ANALYSIS

In this section, we describe the manual analysis of reddit posts that we performed to understand their potential value for requirements engineering and software evolution. We discuss the protocol and steps followed for the manual analysis and report the main findings from the qualitative analysis. This analysis was conducted to provide answers to RQ2.

A. Data analysis

We manually analyzed the content of posts to learn the information related to software requirements and evolution processes. The first two authors of this paper manually labeled 2215 randomly selected reddit posts on WhatsApp, Slack, Premier and Excel.

We ran a two-step process to avoid misunderstandings and disagreements. As the first step, 300 out of 2,215 posts were analyzed together by both annotators. For each post, topics and keywords were identified, whereby the annotators were not restricted to any prior list of topics. The analysis included keywords such as 'bug', 'down', 'issues', 'failing', 'problem', 'unavailable', 'stuck', 'ugly', 'annoying', 'feature', 'new', 'need', 'help', 'wish', 'suggestion', 'question', 'advice', and 'new release'. During this analysis, the prominent topics identified were promotions and marketing of new features, asking for help to use specific features, feedback on features (like-dislike),

[10] we used an online tool: https://subredditstats.com/r/

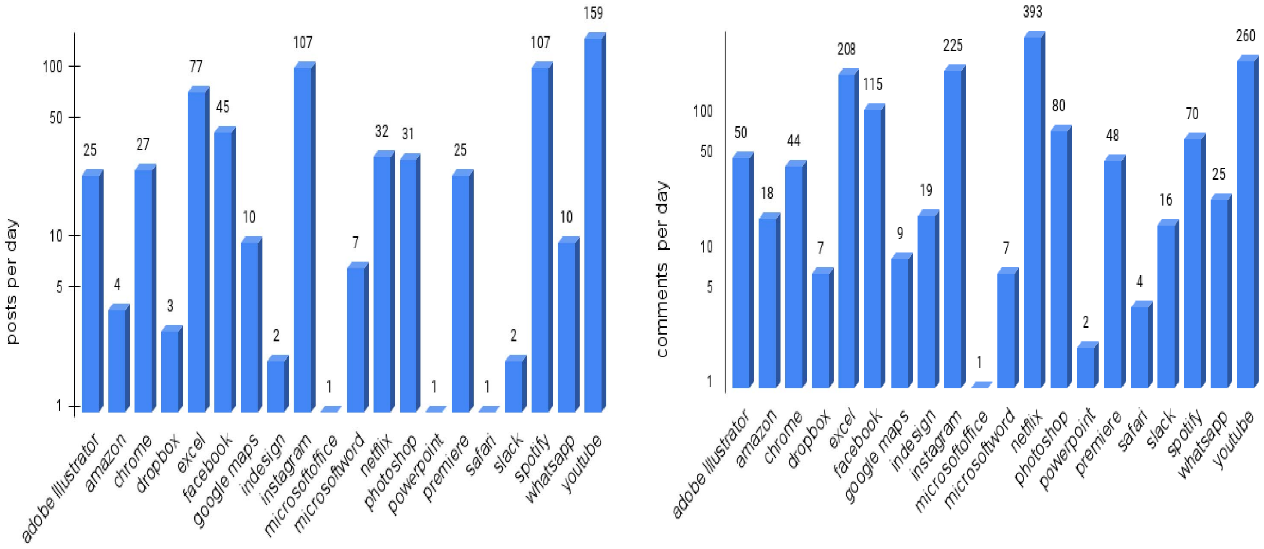

Fig. 1: Right: Comments per day for each subbreddit, Left: Posts per day for each subbreddit (graphs shown in logarithmic scale)

requests for new features, and reporting errors and bugs. Please note that the 300 post that were jointly labelled at this first step are not included in the further data analysis.

Based on this analysis, we adopted the standard categorization scheme that has been typically used to classify user reviews and tweets into the bug reports, feature related, and irrelevant categories [8, 9, 10]. The scope and definition of each category were agreed on the basis of the joint analysis of three hundred posts performed at the first step by both annotators. The categories were defined as follows:

- Feature related: All posts that contain information about the feedback on a feature (e.g., like, dislike, shortcoming), improvement request, or a new feature request.
- Bug report: All posts that report on bugs and errors of a software application.
- Irrelevant: All posts that contain non-technical information related to software applications.

As the second main step of the analysis, each annotator manually labeled a total of 1915 posts independently. Some examples of manually labeled posts from the WhatsApp and Spotify subreddits are shown in Table II. The subreddit posts in Table II reveal that user posts contain helpful information for the RE process. For example, the first post on WhatsApp illustrates that users are requesting a missing feature in the application, i.e., a message scheduling feature. The third shows that users are not happy with the new update feature because the displayed message on the update does not provide any information to users. Posts in the Spotify subreddit about the shuffle feature illustrate that users are not satisfied with this feature. Practitioners could derive a requirement or improvement request based on such posts.

For assessing the agreement rate between the annotators for manual tagging, the standard agreement rate of 84% was calculated. We also performed the Cohen's kappa analysis [16] as it is a more robust measure to assess the agreement between two annotators than a standard agreement rate [17]. We calculated the kappa value k = 0.74 with the 93% confidence interval. Our result showed a "good" strength of agreement between the annotators based on the Cohen's kappa analysis and standard agreement rate. Furthermore, each post was analyzed and discussed among the annotators for resolving conflicts. By means of negotiations and discussions, both annotators eventually agreed upon the final category for the post.

Fig. 2 shows the word cloud for the whole dataset. Notable in the word cloud are the words like "help", "new", "change", and "bug", which also supports the relevance of reddit data for software applications. While labeling, we had the impression that feature related posts can be helpful to derive user stories either for a new feature or for improving an existing one.

B. Results

In summary, our qualitative manual analysis performed on the reddit dataset[11] demonstrated that out of the 1,915 reddit posts examined, 54% of the posts were technically informative containing 35% feature related, and 19% bug reports. While the remaining 46% were categorized as irrelevant, containing invites, ads, and promotions.

V. AUTOMATED CLASSIFICATION

As the next step after the quantitative and qualitative analyses, we explored the potential of automatic classification of reddit posts using machine learning. Similar to the manual

[11] Our manually labelled dataset is available at: https://tinyurl.com/2n5fh7en

Fig. 2: Word Cloud from our dataset

analysis explained in Section IV, the goal of automated classification is to classify the data into bug reports, feature related, and irrelevant categories. To find the most suitable classifier for our dataset, we implemented the Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes (NB) algorithms. Our dataset was not balanced, and to overcome this issue, we used the oversampling technique Synthetic Minority Over-sampling Technique (SMOTE). We applied the following steps on our manually annotated reddit dataset (1915 posts) to train the classifiers and report our results.

A. Data pre-processing

For pre-processing of the reddit data, we used the Natural Language Toolkit (NLTK)[12]. With the help of the toolkit, we performed the data preprocessing by removing stop words, i.e., common words of the English language that have no specific meaning (e.g., "that", "the", "it", "am"), and numerical characters, such as #, @, and emojis. Additionally, we removed unnecessary information, such as URLs and posts in languages other than English. After that, we applied the stemming technique to reduce inflected words to their root forms or stems. Finally, we merged the title and selftext fields of reddit posts into one string and converted it into tokens.

B. Feature Vectorization

After pre-processing of the reddit data, we converted the data into vectors to make it suitable as input for machine learning classifiers. For vector conversion, two different approaches can be used: Bag of Words (BoW) and Term Frequency and Inverse Document Frequency (TF-IDF).

BoW is a simpler approach that produces vectors containing the counts of word occurrences in the document. TF-IDF is a more comprehensive approach enabling to measure relevance rather than frequencies. By TF-IDF, the counts of word occurrences are replaced with the TF-IDF scores across the whole dataset. Therefore, we applied the TF-IDF algorithm to convert our textual data consisting of reddit posts into vectors.

The mathematical formulation of the TF-IDF algorithm is as follows [18]:

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i}) \tag{1}$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

C. Dealing with Imbalanced Data

Class distribution is defined as the number of examples that belong to each class. In classification, an unequal class distribution is referred to as imbalanced data. The manual data analysis in Section IV exhibited our class distribution for the feature related, bug report, and irrelevant categories as 35%, 19%, and 46%, respectively. This unequal class distribution makes our data imbalanced. The challenge of imbalanced data is that most machine learning algorithms will ignore and give poor performance results for the minority class (i.e., the class with fewer examples). One possible solution to balance data is oversampling the minority classes. The most famous oversampling technique is synthetic minority oversampling technique (SMOTE) [19, 20]. We used SMOTE, and its extension – the Support Vector Machine (SVM) SMOTE – for balancing our dataset. SMOTE produces artificial minority samples based on the randomly chosen minority samples and their k-nearest neighbors. In the SVM SMOTE, the SVM classifier is trained on the original training set. The new examples are randomly created along the lines of joining support vectors of all minority classes. The main difference between the SMOTE and SVM SMOTE is the algorithm used to generate synthetic examples, i.e., k-nearest neighbors and support vectors.

D. Classifiers

As the next step, we explored the suitability of different classification algorithms for our dataset. We chose to implement the Support Vector Machine (SVM), Naive Bayes (NB), and Random Forest (RF) algorithms. We chose these classification algorithms because previous studies [8, 21, 22] have found them to perform well in text classification tasks.

We trained our classifiers on the manually labeled dataset discussed in Section IV. For evaluating classifier accuracy, we calculated three parameters typically used in supervised machine learning algorithms – Precision, Recall, and F1-score – in the following way:

$$Precision = \frac{TP_i}{TP_i + FP_i},$$

$$Recall = \frac{TP_i}{TP_i + FN_i},$$

$$F1-score = \frac{2 * Precision * Recall}{Precision + Recall}$$

For example, in the classification of bug reports, TP_i is the total number of posts correctly classified as bug reports,

[12] https://www.nltk.org/

TABLE III: Classification results

	Feature Related			**Bug Report**			**Irrelevant**		
configuration	P	R	F1	P	R	F1	P	R	F1
NB	0.51	0.57	0.54	0.45	0.50	0.47	0.65	0.56	0.60
NB + SMOTE	0.73	**0.80**	0.76	0.82	**0.97**	0.89	**0.80**	0.57	0.67
NB + SVMSMOTE	0.62	0.67	0.65	0.72	0.88	0.79	0.71	0.55	0.62
SVM	0.64	0.52	0.58	0.85	0.33	0.47	0.63	0.87	0.73
SVM + SMOTE	0.81	0.77	**0.79**	0.95	0.93	**0.94**	0.76	0.81	**0.78**
SVM + SVMSMOTE	0.76	0.71	0.74	0.94	0.81	0.88	0.71	0.83	0.77
RF	0.72	0.34	0.46	0.90	0.25	0.39	0.56	**0.95**	0.71
RF + SMOTE	**0.85**	0.61	0.71	**0.97**	0.85	0.90	0.64	0.90	0.75
RF + SVMSMOTE	0.82	0.54	0.65	0.96	0.73	0.83	0.62	0.91	0.74

* We used P for precision, R for recall and F1 for F1-score.

FP_i is the total numbers of posts incorrectly classified as bug reports, and FN_i is the total number of posts that have been incorrectly classified as not belonging to the bug report group. The F1-score is the harmonic mean of precision and recall.

We trained and tested our classifier with a 10-fold cross validation on our manually labeled dataset. This method creates 10 partitions of the dataset such that each partition has 90% of the instances in a training set and 10% in a testing set. The advantage of the cross-fold technique is that it uses all of the data for building models and shows significantly less variance in results. We implemented each classifier using SMOTE and SVM SMOTE, and with the original dataset (imbalanced data).

E. Results

The results for the three classifiers with and without SMOTE and SVM SMOTE are presented in Table III. The classifiers are compared based on precision, recall, and F1-score for each classification category: bug report, feature related, and irrelevant. Our classifiers show poor performance for the original dataset (i.e., imbalanced dataset). Overall, all classifiers demonstrate improvements in results when the oversampling approach balances the data set. The minority class 'bug report' has shown a tremendous improvement with maximum F1-scores for all classifiers using SMOTE and SVM SMOTE. The majority class 'irrelevant' has shown the minimal F1-score of 75% and 67% for RF and NB classifiers, respectively, with the application of SMOTE.

As Table III reflects, SVM outperformed the other classifiers with the F1-score of 84% against 79% and 63% in all three settings: SMOTE, SVM SMOTE, and imbalanced dataset. In terms of accuracy of classifiers, both SVM and RF have achieved competitive results. The reddit posts contain informal language, such as slang, acronyms, and abbreviations that increase the number of features and make the vector representation sparse. Since machine learning algorithms tend to overlearn when the dimensionality is high, SVM is preferred in cases like ours with sparse features. The reason is that SVM works better in case of high-dimensional data spaces because SVM has an over-fitting avoidance behavior that scales up more easily [23].

VI. Discussion

In this section, we revisit our RQs and discuss results. Furthermore, we also describe threats to validity of our study.

A. Answers to the Research Questions (RQs)

Overall we can conclude that reddit is a useful additional data source of user feedback for requirements engineering and software evolution and contains important information about software applications. As all of the authors of this paper have a software engineering background, we can also claim that the information about software applications that can be extracted from reddit is helpful for software practitioners to improve their products as shown in the Table II.

In summary, we answer the research questions (RQs) that were posed in Section II as follows:

- (RQ1) For several of the top mobile and desktop apps there are subreddits where users post frequently.
- (RQ2) The content of reddit posts provides meaningful input for requirements engineering and software evolution. However, the large number of posts found for some applications might not allow for manual analysis in real-world settings.
- (RQ3) Machine learning algorithms (in our case SVM with the application of data balancing methods) provide reasonable accuracy and yield high F1-score for the automated classification of reddit posts. This shows the potential for automatic retrieval of relevant information. The automated analysis of reddit posts can make it valuable and simpler to integrate this information in requirements engineering and software evolution processes.

In the following, we will discuss the answers to the RQs in more detail.

With respect to the research question RQ1 on characteristics of reddit posts, we mined subreddits of top mobile and desktop applications. The statistics of reddit data, including frequencies of posts, numbers of comments, and the total number of users, demonstrated a decent volume of available data and user engagement for software applications. We observed that most big software companies, such as Microsoft, Google maps, and Spotify own their subreddits, where they get posts about their respective software applications. In addition to official subreddits, user-generated software subreddits also exist.

With respect to the research question RQ2 on the significance of the reddit data for requirements engineering and software evolution, we manually analyzed reddit posts. We found that users post their opinions about the software applications, discuss particular features (such as praise, dislike, and shortcomings), and post queries about the usage of a feature, bug reports, and new feature requests. Our qualitative manual analysis demonstrated that out of the 1,915 reddit posts, 35% were feature related and 19% were bug reports. For example, let us consider the following reddit post: *"I am unable to understand why WhatsApp (the most popular messaging app) still does not have a scheduling feature which is such a basic and handy and useful feature. There are a lot of times when we have to message someone at a specific time."* If the software company is working in an agile developing environment, one possible way to analyze these posts is to derive user stories. Consequently, a requirements engineer or product owner can infer a time scheduling feature and derive the corresponding user story *"As an app user, I want to schedule my messages so that I can send a message to my contact at a specific time"*. Let us consider another example of a reddit post: *"WhatsApp battery drain, Whatsapp has recently started using a crazy amount of battery on my OnePlus 6. I've reported it but am curious if the same has happened to anyone else?"*. This post hints to the developer about a bug report that the application might consume too much energy. The Spotify examples in Table II inferred that the same topic could be posted in multiple posts. In addition to the interests of company, other factors related to reddit posts can also help practitioners to prioritize the requirements. For example, the Spotify post *"Spotify needs a sleep timer."* has 45 comments, and 537 upvotes, which can help to prioritize the sleep timer requirement compared to another requirement with a fewer number of upvotes or a higher number of downvotes. At this point, we have not utilized these parameters, but we plan to analyze them in our future work. Regarding the significance of reddit posts, the quantitative analysis of reddit posts performed by us in Section III indicated that 54% of the reddit posts were informative (35% feature related and 19% being bug reports). In comparison to work on Twitter data analysis, our study demonstrated improved results. In [8, 9], the numbers of informative tweets were 42% out of 1350 posts and 50% out of 4000 posts, respectively.

With respect to the research question RQ3 on automated classification, as the manual classification of reddit posts is time consuming and laborious task, we examined three machine learning algorithms with different settings to investigate the scope of automatic classification. Our results show that reddit data has the potential for automatic classification. Since the training dataset used by us was imbalanced between different categories, we used the SMOTE method for balancing our dataset that also improved the results. The SVM classifier with SMOTE demonstrated the F1-score of 84%, and outperformed the NB and RF classifiers. In comparison to twitter data classification results the average classification F1 score in [9] was 76% with SVM for the improvement requests, i.e., feature requests and bug reports combined. In [8], the F1-scores for bug reports and user requests using SVM were 78% and 66%, respectively. For the classification of reddit posts, we obtained higher F1-scores 79% and 94% for feature-related and bug reports using the SVM algorithm. However, we can not directly compare our results with the results obtained in the studies [8, 9] because of the different nature of the Twitter data. In particular, the structure of a tweet and reddit post are different because the limit on the number of characters in a tweet is shorter than the limit on the size of the selftext field of a reddit post. Furthermore, these studies analyzed data for different software applications retrieved at different times.

In a nutshell, reddit data hold valuable information for RE; extracting such information helps to understand the challenges stakeholders face using the software manifested by bug reports or improvement requests on existing features and new feature requests by the users. Overall, this can lead to the formulation of new requirements or to the statement that some of the existing requirements are not met. This, in turn, helps to prioritize development tasks for the existing software applications. On the other hand, this analysis can also be helpful to build a new similar product by selecting features to be included in them based on preferences by the users.

B. Threats to Validity

Our study has several limitations that might affect the validity of the results [24].

The first possible threat to construct validity is that human-labeled data was used as a ground set for the classification of posts. The labeling might reflect an experimental bias as humans tend to be subjective in their judgments. On the other hand, labeling by humans is common in text classification tasks. We have tried to mitigate the possible threat by having two expert annotators, and resolving conflicts by negotiations and mutual agreements. The Cohen's kappa value and confidence value calculated by us also supported our manual data labelling performance. However, the labelling could be further improved by adding more expert annotators to the labeling process. Furthermore, our manual annotations contain data from different domains that enhances the validity of the results achieved by us. Lastly, we have minimized threats to construct validity by using standard evaluation parameters – precision, recall, and F1-score – in our experiments.

A threat to internal validity involves the categorization of our dataset into the feature requests, bug reports, and irrelevant categories. For mitigation, we have very closely followed the same list of categories that is reported in the literature as having been used for mining app reviews and Twitter posts [8, 9, 10]. In our study, the annotation of relevance of posts was determined by the authors of this paper rather than by actual software companies. There could be a different understanding of which content is relevant for software companies.

A possible threat to external validity is the generalizeability of our results due to the limited size of our dataset. We have tried to mitigate this threat by collecting and analyzing data from a wide range of domains. We randomly selected 1915 posts about different software applications and analyzed them without judging the different characteristics of software subreddits. As our dataset incorporated the top software applications, we cannot generalize our results to smaller and less popular applications.

VII. Related work

Existing work on reddit and software engineering is limited and, to the authors' knowledge, reddit has not yet been analyzed using machine learning to improve requirements engineering and software evolution processes. Therefore, we focus the discussion of related work on the mining of software requirements mainly from Twitter and app stores. Our focus is on how these platforms opened up the opportunity to improve requirements engineering and software evolution processes, and how ML and NLP can support in gathering relevant information.

Mobile platforms are the most widely researched platforms for obtaining user feedback. The survey presented in [25] demonstrated a positive impact of app store analysis on the software engineering processes including the phases of requirements engineering, release planning, software design, security, and testing. The study [10] classified app reviews by applying NLP and ML algorithms into bug reports, feature requests, user experiences, and ratings. Another similar study [26] performed automated analysis of app store reviews and classified them to support the software evolution process. The study [27] extracted features and provided opinion summaries from the mobile distribution platforms.

Another popular research area is further sub-categorization of non-functional requirements [28, 29, 30]. In addition to classification problems, also other topics have been discussed in the literature. For example, the article [31] predicted the response by the developers based on the features extracted from user reviews. The study [32] improved the app release planning process, calculating the estimated values of features and cohesiveness between the new and existing features by mining descriptions of the existing applications from app stores.

Another valuable data source for extracting and analyzing user requirements is Twitter [33] that complements app store mining [34]. An exploratory study [35] analyzed the usage characteristics, content, and automated classification potential of tweets about software applications by means of machine learning. This study was extended in [9], where the ALERTme application was proposed to classify, group, and rank tweets about software applications using the techniques of machine learning, topic modeling, and sentiment analysis. A similar study with improved results was conducted in [8] for the classification of tweets into the categories of user requirements, bug reports, and spam. A method named MAPFEAT was proposed in [36] for extracting user needs from Twitter and mapping them into features of already existing apps in app stores. In [37], Facebook, as another social media platform, was analyzed along with app stores and Twitter for automated identification of similarities between feedback provided on multiple platforms and in different languages.

The research presented in [15] analyzed reddit data using high-level query language for extracting information. However, the proposed solution required manual analysis of reddit data for every search query result. Differently, in our work, we have provided an automated approach to categorize useful information from reddit that can support requirements engineering and software evolution processes.

VIII. Conclusions and Future Work

In this paper, we performed an exploratory study that investigated reddit as an additional source of user feedback about software applications. We analyzed the content of reddit posts and the automation potential to classify related posts for requirements engineering and software evolution.

Our results demonstrate that reddit is a useful additional platform providing user feedback with valuable information that can complement user feedback from other data sources, such as app stores, Twitter, and Facebook. This finding is especially valuable in the light of the research result indicating that practitioners are interested in obtaining and analyzing user requirements from multiple platforms [5].

The manual analysis of reddit posts demonstrated that 54% of the reddit posts were informative and discussed bug reports or feature related topics. The remaining 46% posts were non-informative, such as invites, ads, and promotions. The results of our manual analysis supported the need for automation for filtering irrelevant information. We also achieved the 84% F1-score for the Support Vector Machine (SVM) algorithm. These results indicate that classification of reddit data can help software engineers to take quick actions, such as fixing a bug or deriving a user story from a reddit post.

In the future, we are planning to validate our results with software engineers and practitioners. We intend to build a larger dataset by adding more software applications and perform a more in-depth analysis. We want to analyze how posts from a particular software application subreddit can support requirements engineering and software evolution. Furthermore, we intend to add reddit comments and conduct sentiment analysis. To better understand the priority of posts, we also aim at analyzing related parameters, e.g., the numbers of upvotes and downvotes.

References

[1] D. Zowghi. Affects of user involvement in software development. In *2018 1st International Workshop on Affective Computing for Requirements Engineering (AffectRE)*, pages 13–13, 2018.

[2] Sari Kujala. User involvement: a review of the benefits and challenges. *Behaviour & information technology*, 22(1):1–16, 2003.

[3] Muneera Bano and Didar Zowghi. A systematic review on the relationship between user involvement and system success. *Information and Software Technology*, 58:148–169, 2015.

[4] Marc Oriol, Melanie Stade, Farnaz Fotrousi, Sergi Nadal, Jovan Varga, Norbert Seyff, Alberto Abello, Xavier Franch, Jordi Marco, and Oleg Schmidt. Fame: supporting continuous requirements elicitation by combining user feedback and monitoring. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 217–227. IEEE, 2018.

[5] Tahira Iqbal, Norbert Seyff, and Daniel Mendez. Generating requirements out of thin air: Towards automated feature identification for new apps. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 193–199. IEEE, 2019.

[6] Irina Todoran, Norbert Seyff, and Martin Glinz. How cloud providers elicit consumer requirements: An exploratory study of nineteen companies. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 105–114. IEEE, 2013.

[7] Mark Harman, Yue Jia, and Yuanyuan Zhang. App store mining and analysis: Msr for app stores. In *2012 9th IEEE working conference on mining software repositories (MSR)*, pages 108–111. IEEE, 2012.

[8] Grant Williams and Anas Mahmoud. Mining twitter feeds for software user requirements. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 1–10. IEEE, 2017.

[9] Emitza Guzman, Mohamed Ibrahim, and Martin Glinz. A little bird told me: Mining tweets for requirements and software evolution. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 11–20. IEEE, 2017.

[10] Walid Maalej and Hadeer Nabil. Bug report, feature request, or simply praise? on automatically classifying app reviews. In *2015 IEEE 23rd international requirements engineering conference (RE)*, pages 116–125. IEEE, 2015.

[11] Michael M Tadesse, Hongfei Lin, Bo Xu, and Liang Yang. Detection of depression-related posts in reddit social media forum. *IEEE Access*, 7:44883–44893, 2019.

[12] Albert Park and Mike Conway. Tracking health related discussions on reddit for public health applications. In *AMIA Annual Symposium Proceedings*, volume 2017, page 1362. American Medical Informatics Association, 2017.

[13] Alexey N Medvedev, Renaud Lambiotte, and Jean-Charles Delvenne. The anatomy of reddit: An overview of academic research. In *Dynamics on and of Complex Networks*, pages 183–204. Springer, 2017.

[14] Alexander James Richardson. *Estimating the Impact of Political Propaganda on Reddit Users' Political Opinions*. PhD thesis, Georgetown University, 2020.

[15] Georgi M Kanchev, Pradeep K Murukannaiah, Amit K Chopra, and Pete Sawyer. Canary: an interactive and query-based approach to extract requirements from online forums. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 470–471. IEEE, 2017.

[16] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

[17] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[18] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. New Jersey, USA, 2003.

[19] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[20] Jiawen Kong, Wojtek Kowalczyk, Stefan Menzel, and Thomas Bäck. Improving imbalanced classification by anomaly detection. In *International Conference on Parallel Problem Solving from Nature*, pages 512–523. Springer, 2020.

[21] Charu C. Aggarwal and ChengXiang Zhai. *A Survey of Text Classification Algorithms*, pages 163–222. Springer US, Boston, MA, 2012.

[22] Manal Binkhonain and Liping Zhao. A review of machine learning algorithms for identification and classification of non-functional requirements. *Expert Systems with Applications: X*, 1:100001, 2019.

[23] Peter Brusilovski, Alfred Kobsa, and Wolfgang Nejdl. *The adaptive web: methods and strategies of web personalization*, volume 4321. Springer Science & Business Media, 2007.

[24] Angela Dean, Daniel Voss, and Danel Draguljić. *Design and analysis of experiments*, volume 1. Springer, 1999.

[25] William Martin, Federica Sarro, Yue Jia, Yuanyuan Zhang, and Mark Harman. *IEEE transactions on software engineering*, 43(9):817–847, 2016.

[26] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A Visaggio, Gerardo Canfora, and Harald C Gall. How can i improve my app? classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 281–290. IEEE, 2015.

[27] Emitza Guzman and Walid Maalej. How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd international requirements engineering conference (RE)*, pages 153–162. IEEE, 2014.

[28] Tianlu Wang, Peng Liang, and Mengmeng Lu. What aspects do non-functional requirements in app user reviews describe? an exploratory and comparative study. In *2018 25th Asia-Pacific Software Engineering Conference*

(APSEC), pages 494–503. IEEE, 2018.

[29] Mengmeng Lu and Peng Liang. Automatic classification of non-functional requirements from augmented app user reviews. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 344–353, 2017.

[30] Nishant Jha and Anas Mahmoud. Mining non-functional requirements from app store reviews. *Empirical Software Engineering*, 24(6):3659–3695, 2019.

[31] Kamonphop Srisopha, Daniel Link, Devendra Swami, and Barry Boehm. Learning features that predict developer responses for ios app store reviews. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–11, 2020.

[32] Maleknaz Nayebi and Guenther Ruhe. Optimized functionality for super mobile apps. In *2017 IEEE 25th international requirements engineering conference (RE)*, pages 388–393. IEEE, 2017.

[33] Tahira Iqbal, Parisa Elahidoost, and Levi Lúcio. A bird's eye view on requirements engineering and machine learning. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pages 11–20, 2018.

[34] Maleknaz Nayebi, Henry Cho, and Guenther Ruhe. App store mining is not enough for app improvement. *Empirical Software Engineering*, 23(5):2764–2794, 2018.

[35] Emitza Guzman, Rana Alkadhi, and Norbert Seyff. A needle in a haystack: What do twitter users say about software? In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, pages 96–105. IEEE, 2016.

[36] Maleknaz Nayebi, Mahshid Marbouti, Rache Quapp, Frank Maurer, and Guenther Ruhe. Crowdsourced exploration of mobile app features: A case study of the fort mcmurray wildfire. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*, pages 57–66. IEEE, 2017.

[37] Emanuel Oehri and Emitza Guzman. Same same but different: Finding similar user feedback across multiple platforms and languages. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 44–54. IEEE, 2020.