Novatos: Analisando os Efeitos Causados Pelas Primeiras Rejeições em Projetos Open-source

Kalebe dos Santos Silva

[1]Universidade Federal do Acre (UFAC)
Programa de Pos-Graduacão em Ciência da Computacão (PPGC)
Caixa Postal 69920900 – Rodovia BR 364 – Distrito Industrial – Rio Branco - AC – Brazil

kalebeadvestudos@gmail.com

Abstract. *Open source communities have gained prominence over time, attracting programmers from all over the world. These people, called volunteers, are willing to contribute to the growth of open-source projects. In order for a project to survive, there needs to be a flow of new contributors, and this work focuses on analyzing the first contributions of these newcomers, seeking to understand the impacts caused by the first rejections. To do this, we used a data mining technique called association rules and found that newcomers tend to abandon projects when their contributions are rejected and even when they are accepted, only a small proportion return to contribute in the future.*

Resumo. *As comunidades de código livre tem ganhado destaque com o passar do tempo, atraindo programadores de todo o mundo. Essas pessoas, chamadas de voluntários, se prontificam em contribuir com o crescimento dos projetos open-source. Para que um projeto possa sobreviver é necessário existir um fluxo de novos contribuidores, este trabalho é voltado em analisar as primeiras contribuições destes novatos, buscando compreender os impactos causados pelas primeiras rejeições. Para isso, usamos uma técnica de mineração de dados chamada regras de associação e descobrimos que os novatos tendem a abandonar os projetos quando suas contribuições são rejeitadas e mesmo quando aceitas, apenas uma pequena parcela volta a contribuir no futuro.*

1. Introdução

As comunidades de software livre vem crescido cada vez mais com o passar dos anos [Pinto et al. 2016], dentre os fatores que influenciam no crescimento destas comunidades, encontram-se os novatos, pois eles possuem um papel crítico na sobrevivência dessas comunidades [Steinmacher et al. 2019a]. Não é novidade que esses projetos atraiam a atenção de voluntários do mundo inteiro. Desenvolvedores participam contribuem, pois auxiliar com o crescimento destes repositórios open-source, os ajuda a aprimorar seus conhecimentos [David and Shapiro 2008], possuir visibilidade na comunidade [Russo et al. 2008], possíveis benefícios sociais [Parra et al. 2016] e até mesmo conseguir um emprego [Steinmacher et al. 2019b].

De modo geral, o sucesso a longo prazo, a continuidade, a relevância do projeto gira em torno do fluxo de novatos, eles fazem parte do combustível que mantém a sobrevivência dos projetos de código aberto [Qureshi and Fang 2011]. Este trabalho é voltado em analisar as primeiras contribuições dos novatos em projetos open-source, buscando compreender os impactos causados pelas primeiras rejeições.

2. Método

Neste trabalho, foi adotado a técnica de mineração de dados intitulada regras de associação. Essa técnica pode revelar informações ocultas, que passariam despercebidas em uma análise exploratória. O cenário analisado gira em torno das pull requests, mais especificamente nas causas que a rejeição dos first pulls pode causar. Estudo foi empregado o processo de descoberta de conhecimento em banco de dados, também conhecido como *Knowledge Discovery in Databases (KDD)*, utilizando os seguintes passos [Fayyad et al. 1996]: (1) Seleção dos dados, (2) Pré-processamento dos dados, (3) Transformação e enriquecimento da base de dados, (4) Regras de associação, (5) Interpretação e avaliação dos resultados.

O primeiro passo foi feito utilizando a ferramenta GHTorrent para a coleta de dados; no segundo passo os dados coletados serão agrupados em forma de tabela e carregados na ferramente Weka para a remoção de atributos inúteis, esta ferramenta será utilizada para realizar algumas etapas dos passos posteriores; no terceiro passo a base de dados será carregada em um sistema de gerenciamento de banco de dados (SGBD), onde será incrementado um novo atributo; no quarto passo, a ferramenta Weka será utilizada para realizar a mineração de dados, por meio do algoritmo Apriori; no quinto passo, será utilizado o plugin WekaPar[da Silva 2016] para uma melhor análise dos resultados. Esse processo foi pensando visando encontrar a resposta para as seguintes questões de pesquisa.

- **RQ1:** Quais os efeitos colaterais causados pela primeira rejeição?

2.1. Seleção dos dados

A coleta das bases de dados GHtorrent [Gousios and Spinellis 2012]. Os dados foram extraídos entre os períodos de novembro de 2015 e fevereiro de 2016. As bases analisadas foram coletadas entre janeiro de 2011 e outubro de 2015, contendo 14.526 pull requests.[Moreira Soares et al. 2021] O cenário analisado é o de pull requests.

Tabela 1. Informações sobre os repositórios

Nome da base	#Atributos	#Pull request	Linguagem
Xbcm	61	3816	C++
Docker	61	6979	Go
Django	61	3731	Python

2.2. Pré-processamento dos dados

O primeiro critério foi avaliar se os atributos poderiam ser necessários ou não para o escopo dos problemas de pesquisa. Foram removidos todos os atributos que a primeiro momento, não iram contribuir para as próximas fases do estudo, a tabela 2 informa os atributos obtidos após a filtragem.

2.3. Transformação e enriquecimento da base de dados

Nesta etapa foi utilizado o gerenciador de banco de dados mysql workbench. O motivo para optar por utilizar um SGBD (sistema gerenciador de banco de dados), foi por conta do comprometimento que os bancos de dados relacionais possuem com os dados

Tabela 2. Resultados após a filtragem dos dados

Atributos	Descrição
idPull	Referência a ID única de cada pull request
login	Indica o nome do usuário que fez o pull request
owner_repo	Informa o nome do repositório
language	Informa qual linguagem de programação está sendo utilizada no projeto
first_Pull	Indica se é a primeira contribuição do desenvolvedor no repositório
status	Indica se o pull request foi aceito ou rejeitado
requester_experience_project	Rotula o nível de experiência que o contribuidor possui no projeto

armazenados, além disso, a linguagem sql é poderosa e possui uma sintaxe de simples compreensão[Melton and Simon 1993]. Primeiramente foi necessário criar duas tabelas, a primeira chamada *user* e a segunda chamada *dataset*.

Tabela 3. User

Campo	Características
Devname	Varchar(255) Primary key

A tabela 5 contem apenas o campo *devname*. Esta coluna contem o nome de todos os desenvolvedores que contribuíram com o projeto até o momento da coleta dos dados, ele é único, ou seja, não possui nomes duplicados.

Tabela 4. Dataset

Campo	Característica
IdPull	Varchar (255) primary key
Login	Varchar (255) foreign key
Owner_repo	Varchar (255)
Language	Varchar (255)
first_Pull	Varchar (255)
status	Varchar (255)
requester_experience_project	Varchar (255)

Esta tabela 4 contém todos os dados filtrados (disponibilizados na tabela 2). O campo *idPull* é responsável por identificar cada instância do dataset por meio de um valor único. Campo login indica qual desenvolvedor fez o pull request, ele é utilizado para linkar a entidade *user* com a entidade *dataset* no banco de dados.

Após realizar a criação das duas entidades no banco de dados, inserimos os registros coletados e realizados uma operação com o intuito de enriquecer a base de dados. Esse query consiste em buscar campos que contem apenas um *login* único em todo o repositório de software e adicionar o valor ”FALSE”, informando que este desenvolvedor nunca mais voltou a contribuir com o projeto; em contrapartida, todos os desenvolvedores que possuem mais de um pull request enviado serão rotulados como ”TRUE”, esse rótulo é válido apenas para o primeiro registro desse desenvolvedor, ou seja, o rótulo não se repete para o mesmo desenvolvedor, não importando a quantidade de contribuições que ele tenha feito.

```
WITH user_repo_counts AS (SELECT owner_repo,login, COUNT(*) AS repo_count FROM dataset GROUP BY owner_repo, login ), marked_records AS ( SELECT d.*, CASE WHEN urc.repo_count = 1 THEN 'FALSE' WHEN ROW_NUMBER() OVER (PARTITION BY d.owner_repo, d.login ORDER BY d.idPull) = 1 THEN 'TRUE' ELSE '' END AS returnProject FROM dataset d JOIN user_repo_counts urc ON d.owner_repo = urc.owner_repo AND d.login = urc.login ) SELECT * FROM marked_records;
```

Tabela 5. Query utilizada para enriquecimento da base

User_repo_counts conta o número de registros (repo_count) para cada login em owner_repo; marked_records adiciona o campo record_mark à tabela e então este campo irá definir "FALSE"se a quantidade de repo_count for 1 e "TRUE"para todos os registros com repo_count maior que 1. Os campos posteriores ficam vazios.

2.4. Regras de associação

As regras de associação funcionam de forma em que o antecedente (X) implica no consequente (Y), a relevância (peso) entre $X \rightarrow Y$ é definida pelo lift, o suporte é métrica que define aquela ocasião em relação à base de dados e a confiança é métrica que mede a probabilidade de que o consequente (Y) seja verdadeiro[Witten and Frank 2002, Han et al. 2022].

2.5. RQ1 - Quais os efeitos colaterais causados pela primeira rejeição?

Esta analise, busca extrair informações que indiquem o impacto que as primeiras rejeições possuem para os novos colaboradores.

Tabela 6. RQ1 - Atributos utilizados

Atributo	Descrição
returnProject	rotulado em *"false"* indicando que o desenvolvedor não voltou a contribuir com o projeto; *"true"*, indicando se o desenvolvedor voltou a contribuir em algum determinado momento.
requester_experience_project	rotulado em *"no contributions", "some contributions", "many contributions"*. Indica o quão familiarizado o desenvolvedor está com o projeto
status	retulado em *"merged", "closed"*, indicando se o pull request foi aceito ou rejeitado

3. Resultados e Discussão

Nesta sessão, discutiremos sobre os resultados encontrados nas questões de pesquisa definição na sessão anterior. O algoritmo Apriori foi configurado com suporte mínimo de 0.01 e confiança de 0.01, gerando regras até a exaustão.

3.1. RQ1 - Quais os efeitos colaterais causados pela primeira rejeição?

Nesta etapa, para gerar as regras de associação, foram utilizados os seguintes atributos: *returnProject, status, requester_experience_project*.

Ao analisarmos as regras, podemos observar que:

Tabela 7. RQ1 - Resultados

Id	Antecedente (X)	Consequente (Y)	Suporte	Confiança	lift
1	status=closed requester_experience_project='no contribution'	returnProject=FALSE	0.08	0.68	4.52
2	status=merged requester_experience_project='no contribution'	returnProject=FALSE	0.06	0.6	3.98
3	status=merged requester_experience_project='no contribution'	returnProject=TRUE	0.03	0.33	3.61
4	status=closed requester_experience_project='no contribution'	returnProject=TRUE	0.03	0.25	2.82

1. Na regra 1, quando um novato envia sua primeira contribuição (requester_experience_project = 'no contribution'), caso ela seja rejeitada (status=closed), 68% tendem a desistir de realizar novas contribuições com o projeto (lift=4.52).
2. Na regra 2, caso a primeira contribuição seja aceita (status=merged), 60% tendem a não voltar a contribuir com o projeto (lift=3.98).
3. Na regra 3, observamos que existem outros casos mais específicos (suporte = 0.03), em que, caso a primeira contribuição seja aceita, 33% dos contribuidores, realizaram novas contribuições (lift=3.61).
4. Na regra 4, observamos que em casos mais específicos (suporte = 0.03), mesmo que a primeira contribuição seja rejeitada (status=closed), 25% dos novatos voltam a realizar uma nova contribuição(lift=2.82).

RQ1 - Com base nos dados extraídos, compreendemos que os efeitos colaterais, causados pelas primeiras rejeições, normalmente, implicam no abandono dos novos colaboradores (68%), porém é possível notar que, caso, a primeira contribuição seja aceita, ainda há um alto índice de evasão (60%). Em casos mais raros, a aceitação influenciou positivamente no retorno dos colaboradores e mesmo havendo rejeição, ainda tentaram novamente. Possibilitando compreender que apenas uma pequena parcela dos novatos que realmente desejam contribuir com o crescimento dos projetos open-source.

4. Trabalhos Relacionados

[Soares et al. 2018] em seu estudo sobre os fatores que influenciam a designação de revisores em pulls request, examina causas que contribuem para a aceitação ou rejeição de certos pulls requests, em um determinado momento, é descoberto que a familiaridade do contribuidor para com o projeto, tende a implicar no fator da aceitação de pull requests, ou seja, novos contribuidores, tendem a ter maiores chances de adquirir uma rejeição em sua primeira contribuição. Neste estudo, buscamos compreender quais os efeitos colaterais causados pela primeira rejeição.

[Steinmacher et al. 2013], em seu estudo, visando compreender as causas que levam novos contribuidores a abandonar projetos open-source, conduziu uma investigação em um projeto de sucesso, chamado *Hadoop Common project*. O estudo consistia na coleta e análise de dados referentes a emails e comentários jira, em um espaçamento de 60 meses. Ao observar o comportamento dos novatos, pode-se notar que, quando estes, realizam alguma pergunta e esta é prontamente respondida, eles agradecem e saem da comunidade, demonstrando nestes casos, que não possuíam intensão de contribuir com o projeto. Neste estudo, buscamos descobrir certos fatores que influenciam na evasão em projetos de software livre.

[Fronchetti et al. 2019] em seu estudo, busca entender quais fatores influenciam normalmente a chegada de novos contribuidores em projetos de código aberto, para compreender tais causas, foi condizida uma investigação em 450 repositórios de software,

observando, fatores, como, linguagem de programação, número de estrelas em um projeto, entre outros. Ao realizar a análise destes repositórios, utilizando o algoritmo de clusterização *K-Spectral Centroid (KSC)*, descobriu-se que a popularidade do projeto, o time de revisão de pull requests, a idade do projeto e a linguagem de programação utilizada, são as causas que melhor indicam a adesão de novatos em projetos open-source. Neste estudo, buscamos encontrar fatores que influenciam a perma dos novatos nos projeto open-source.

5. Ameaças a Validade

Este estudo possui vários riscos a sua validade. O primeiro é referente aos atributos tratados, pois podem ter havido excluído atributos importantes que auxiliariam na extração de dados. O segundo está atrelado a confiabilidade dos dados, pois em algum determinado ponto do tratamento dos dados, podem ter ocorrido alguma perda de dados ocasionando imprecisão nos resultados obtidos. O terceiro é em relação à coleta de dados, pois ela foi feita a bastante tempo e por conta disso, então os resultados encontrados podem estar desatualizados ou até mesmo enviesados. Por fim, há riscos relacionados a extensão dos resultados obtidos, pois eles não valem todos os projetos (incluindo projetos fechados e até mesmos projetos de código aberto), pois essa é uma análise isolada com apenas três projetos, então, as regras extraídas aqui encontradas podem não remeter a realidade da grande parte dos projetos de software de código aberto.

6. Conclusões

Este estudo busca descobrir a existência de fatores que contribuem ou não com a adesão de novos usuários para projetos open-source. Então foi descoberto, por meio da mineração de dados, utilizando a técnica de extração de regras de associação, que, o novo contribuidor, ao receber sua primeira rejeição, tendem a não voltar a contribuir no futuro, também foram identificados que desenvolvedores que possuem os primeiros pull requests aceitos, tendem a não voltar. Além disso, foram encontrados, que, em proporções menores, existem novatos que independente da rejeição ou aceitação, tendem a voltar a contribuir. Com base nos dados encontrados, podemos observar que grande parte dos novatos não possuíam quaisquer interesse em contribuir verdadeiramente com projetos open-source. Em estudos futuros, estudar maneiras de identificar e compreender fatores que cativem os novatos a se tornarem contribuidores em projetos, mesmo após as primeiras rejeições em seus pull requests.

Referências

da Silva, D. A. N. (2016). Wekapar: uma extensão da ferramenta weka para auxiliar o pós-processamento de regras de associação.

David, P. A. and Shapiro, J. S. (2008). Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy*, 20(4):364–398. Empirical Issues in Open Source Software.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). Advances in knowledge discovery and data mining. In *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence.

Fronchetti, F., Wiese, I., Pinto, G., and Steinmacher, I. (2019). What attracts newcomers to onboard on oss projects? tl;dr: Popularity. In Bordeleau, F., Sillitti, A., Meirelles, P., and Lenarduzzi, V., editors, *Open Source Systems*, pages 91–103, Cham. Springer International Publishing.

Gousios, G. and Spinellis, D. (2012). Ghtorrent: Github's data from a firehose. In *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*, pages 12–21.

Han, J., Pei, J., and Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

Melton, J. and Simon, A. R. (1993). *Understanding the new SQL: a complete guide*. Morgan Kaufmann.

Moreira Soares, D., de Lima Júnior, M. L., Murta, L., and Plastino, A. (2021). What factors influence the lifetime of pull requests? *Software: Practice and Experience*, 51(6):1173–1193.

Parra, E., Haiduc, S., and James, R. (2016). Making a difference: an overview of humanitarian free open source systems. In *Proceedings of the 38th International Conference on Software Engineering Companion*, ICSE '16, page 731–733, New York, NY, USA. Association for Computing Machinery.

Pinto, G., Steinmacher, I., and Gerosa, M. A. (2016). More common than you think: An in-depth study of casual contributors. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, volume 1, pages 112–123.

Qureshi, I. and Fang, Y. (2011). Socialization in open source software projects: A growth mixture modeling approach. *Organizational Research Methods*, 14(1):208–238.

Russo, B., Damiani, E., Hissam, S., Lundell, B., and Succi, G. (2008). *Open Source Development, Communities and Quality: IFIP 20th World Computer Congress, Working Group 2.3 on Open Source Software, September 7-10, 2008, Milano, Italy*, volume 275. Springer Science & Business Media.

Soares, D. M., de Lima Júnior, M. L., Plastino, A., and Murta, L. (2018). What factors influence the reviewer assignment to pull requests? *Information and Software Technology*, 98:32–43.

Steinmacher, I., Gerosa, M., Conte, T. U., and Redmiles, D. F. (2019a). Overcoming social barriers when contributing to open source software projects. *Computer Supported Cooperative Work (CSCW)*, 28(1):247–290.

Steinmacher, I., Gerosa, M., Conte, T. U., and Redmiles, D. F. (2019b). Overcoming social barriers when contributing to open source software projects. *Computer Supported Cooperative Work (CSCW)*, 28:247–290.

Steinmacher, I., Wiese, I., Chaves, A. P., and Gerosa, M. A. (2013). Why do newcomers abandon open source software projects? In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 25–32.

Witten, I. H. and Frank, E. (2002). Data mining: practical machine learning tools and techniques with java implementations. *Acm Sigmod Record*, 31(1):76–77.