Avaliando a Importância do Arquivo *contributing.md* em Projetos de Código Aberto

Silvana de Andrade Gonçalves[1], **Alexandre Plastino**[2], **Daricélio Moreira Soares**[3]

[1]Instituto Federal do Acre (IFAC), Rio Branco – AC – Brasil

[2]Universidade Federal Fluminense (UFF), Niterói – RJ – Brasil

[3]Universidade Federal do Acre (UFAC), Rio Branco – AC – Brasil

silvana.goncalves@ifac.edu.br, platino@ic.uff.br, daricelio.soares@ufac.br

Abstract. *This article investigates whether the inclusion and updates on the contributing.md file in open source projects influence the participation of new contributors. Using a temporal association rules analysis methodology on pull request data, the study evaluates the impact of these creation and changes on the file, implemented by the project's core team. The results indicate that the updates and active use of the contributing.md file facilitate the participation of new collaborators, reducing the chances of their contributions being rejected. Furthermore, the analysis reveals significant temporal variations of these chances, providing a more detailed understanding of the factors that affect the acceptance and lifetime of pull requests over time.*

Resumo. *Este artigo investiga se a inclusão e atualização do arquivo contributing.md em projetos de código aberto influenciam a participação de novos colaboradores. Utilizando uma metodologia de análise temporal de regras de associação sobre dados de pull requests, o estudo avalia o impacto da criação e modificações no arquivo, implementadas pela equipe principal do projeto. Os resultados indicam que a utilização ativa do arquivo contributing.md facilita a participação de novos colaboradores, reduzindo as chances de rejeição de suas contribuições. Além disso, a análise revela variações temporais significativas dessas chances, oferecendo uma compreensão mais detalhada dos fatores que afetam a aceitação e o tempo de vida dos pull requests ao longo do tempo.*

1. Introdução

O desenvolvimento de software colaborativo com a utilização de *pull requests* tem um papel central em projetos de software de código aberto, permitindo que desenvolvedores externos ao time principal contribuam com o projeto. Plataformas como GitHub[1], Gitlab[2] e Bitbucket[3] facilitam esse processo, fornecendo ferramentas para submissão, revisão e aprovação de código. Neste cenário, o time principal é responsável por analisar os *pull requests*, decidindo por sua aceitação ou rejeição.

Aspectos técnicos [Soares et al. 2021][Gousios et al. 2014][Soares et al. 2015], sociais [Tsay et al. 2014][Rastogi et al. 2018][Ford et al. 2019] e características dos contribuidores externos [Soares et al. 2018][Steinmacher et al. 2018] podem influenciar na

[1]https://github.com/

[2]https://about.gitlab.com/

[3]https://bitbucket.org/

análise dos *pull requests*. De acordo com alguns estudos, contribuidores inexperientes têm menos chance de aceitação [Soares et al. 2015], enquanto suas contribuições tendem a ter um tempo de análise longo [Soares et al. 2021].

Nesta direção, é comum que projetos criem diretrizes para orientar a submissão de *pull requests*. No GitHub, por exemplo, repositórios de código aberto usam o arquivo *contributing.md* para este fim, o que acontece em cerca de 35% dos projetos, segundo Fronchetti et al. [Fronchetti et al. 2023].

O arquivo *contributing.md* é um documento colocado na raiz do repositório com o objetivo de guiar os colaboradores externos sobre como contribuir com o projeto. Normalmente, inclui orientações sobre envio de *pull requests*, relatos de problemas, padrões de codificação e testes, buscando alinhar as contribuições às expectativas dos mantenedores.

Estudos anteriores investigaram padrões no processo de avaliação de *pull requests* por meio de técnicas estatísticas [Gousios et al. 2014][Rastogi et al. 2018] e regras de associação [Soares et al. 2021][Soares et al. 2015][Soares et al. 2018]. No entanto, a maioria desses estudos frequentemente negligencia o aspecto temporal dos dados, assumindo que o conhecimento obtido não varia. A análise temporal dos padrões pode fornecer uma nova perspectiva sobre esse processo, revelando mudanças na dinâmica dos *pull requests* ao longo do tempo.

A extração de regras de associação é uma técnica de mineração de dados que identifica padrões e correlações em bases de dados [Kotsiantis and Kanellopoulos 2006], revelando relacionamentos importantes entre atributos. Por exemplo, no projeto Puppet[4], foi extraída a regra FirstPull = "*true*" $\rightarrow$ Status = "aceito" com Confiança de 0,58, o que indica uma probabilidade de aceitação de *pull requests* de novos colaboradores (inexperientes) de 58% neste projeto.

A análise temporal das medidas de interesse das regras de associação revela variações significativas na força das regras. Na Figura 1, é possível observar a variação da Confiança da regra exemplificada. A linha vermelha representa a Confiança da regra considerando toda a base ao longo do tempo. Cada barra representa a Confiança da regra em determinada partição temporal da base. Nota-se que a influência da aceitação dos *pull requests* de novatos oscila com o passar do tempo, evidenciando a necessidade de identificar os fatores responsáveis por essa variação.

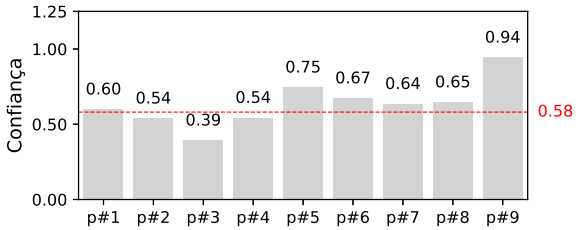

Figura 1. Análise particionada da Confiança da regra: FirstPull = "*true*" $\rightarrow$ Status = "aceito" no projeto Puppet. Particionamento de 12 meses.

O objetivo deste trabalho é avaliar se a inserção e modificações no arquivo *contributing.md* em projetos de código aberto influenciam a participação de novos contri-

[4]https://github.com/puppetlabs/puppet

buidores. Para isso, propõe-se uma metodologia baseada na análise temporal de regras de associação, avaliando o impacto na aceitação e no tempo de vida dos *pull requests* submetidos por contribuidores externos e que estão contribuindo pela primeira vez no projeto.

Os resultados mostram que a utilização do arquivo *contributing.md* e sua evolução podem facilitar a participação de novos colaboradores em projetos de código aberto, sendo possível identificar que a influência do arquivo pode variar, impactando a dinâmica de aceitação e o tempo de vida dos *pull requests*.

Este trabalho está organizado nas seguintes seções. A Seção II discute os conceitos e trabalhos relacionados. A Seção III apresenta a metodologia utilizada. Na Seção IV, são discutidos os resultados. Por fim, a Seção V apresenta as implicações e limitações do estudo, e recomendações para trabalhos futuros.

2. Conceitos e Trabalhos Relacionados

Alguns estudos têm explorado diferentes aspectos sobre a análise de *pull requests* e sobre regras de associação temporais, os quais são destacados a seguir.

Fronchetti et al. [Fronchetti et al. 2023] analisaram a utilização do arquivo *contributing.md* em projetos de código aberto, identificando barreiras de integração enfrentadas por novatos. Eles concluem que muitos desses projetos precisam ampliar a abrangência de seus arquivos *contributing.md* para melhor atender aos recém-chegados.

Soares et al. [Soares et al. 2018] concluíram, por meio de regras de associação, que em projetos de código aberto que usam *pull requests*, as chances de aceitação de contribuições de desenvolvedores externos ou inexperientes são reduzidas.

Ale e Rossi [Ale and Rossi 2000] abordaram regras de associação que têm um alto valor de confiança, mas que não são geradas por terem um suporte baixo e, como solução, propuseram que seja usado um suporte temporal limitado ao total de transações pertencentes à vida útil dos itens.

Segura-Delgado et al. [Segura-Delgado et al. 2020] investigaram trabalhos sobre regras de associações temporais e propuseram uma taxonomia para representá-las e classificá-las segundo a utilização do atributo tempo dentro do processo de geração das regras.

Já Liu et al.[Liu et al. 2021] mostraram que, em alguns cenários, os dados não estão presentes em todo o período de tempo representado, e oferecem uma abordagem adaptada da técnica de mineração de dados convencional para considerar o tempo de vida útil dos atributos no momento da geração das regras.

Gonçalves et al. [Gonçalves et al. 2021] apresentaram uma metodologia de análise temporal de regras de associação sobre dados de *pull requests* evidenciando a variação da medida de *Lift* das regras geradas em partições temporais.

Este trabalho objetiva demonstrar que a criação e/ou alterações no arquivo *contributing.md* podem estar correlacionadas com variações nas medidas de interesse das regras de associação extraídas de dados de *pull requests*. Ao explorar isso, pretende-se fornecer evidências de que o arquivo *contributing.md* influencia a participação de novos colaboradores em projetos de código aberto.

3. Materiais e Métodos

As subseções a seguir descrevem regras de associação e a metodologia de avaliação temporal. Apresentam-se também informações sobre a base de dados e os atributos utilizados.

3.1. Regras de Associação Multidimensionais

Regras de associação multidimensionais são extraídas de bases de dados relacionais. Elas têm a forma $X_1 \wedge X_2 \wedge \ldots \wedge X_n \rightarrow Y_1 \wedge Y_2 \wedge \ldots \wedge Y_m$, onde o antecedente, representado por X, e o consequente, representado por Y, são conjunções de condições sobre atributos distintos da base de dados. Neste artigo, um exemplo é: FirstPull = "*true*" $\rightarrow$ Status = "aceito", indicando que *pull requests* de novos contribuidores tendem a ser aceitos [Agrawal 1994]. As regras de associação são avaliadas pelas medidas de interesse Suporte, Confiança e *Lift*.

O Suporte é definido como a proporção de registros na base que satisfazem simultaneamente as condições do antecedente X e do consequente Y, sendo calculado pela Fórmula 1. Nessa equação, $T(X \cup Y)$ representa o número de registros que atendem a essas condições, enquanto T corresponde ao total de registros na base de dados [Agrawal 1994].

$$Sup(X \rightarrow Y) = \frac{T(X \cup Y)}{T} \tag{1}$$

A Confiança indica a probabilidade condicional de Y ocorrer dado que X ocorreu, sendo calculada pela Fórmula 2. Na equação, $T(X \cup Y)$ indica o número de registros que satisfazem simultaneamente as condições de X e Y, enquanto $T(X)$ representa o número de registros que atendem às condições do antecedente X [Agrawal 1994].

$$Conf(X \rightarrow Y) = \frac{T(X \cup Y)}{T(X)} \tag{2}$$

O *Lift* [Agrawal 1994] avalia a correlação entre X e Y, e é calculado pela razão entre a confiança da regra e a probabilidade de ocorrência de Y ($Sup(Y)$), ilustrado pela Fórmula 3. O suporte de Y ($Sup(Y)$) representa a porcentagem de registros que satisfazem a condição de Y.

$$Lift(X \rightarrow Y) = \frac{Conf(X \rightarrow Y)}{Sup(Y)} \tag{3}$$

O *Lift* pode ser interpretado da seguinte forma: (i) Lift$>$1 indica que X e Y apresentam dependência positiva. A ocorrência de X aumenta a chance da ocorrência de Y; (ii) Lift$<$1 indica que X e Y apresentam dependência negativa, ou seja, a ocorrência de X diminui a chance de Y ocorrer; e (iii) Lift = 1 indica que X e Y são independentes um do outro [Kotsiantis and Kanellopoulos 2006].

Neste estudo, a base de dados analisada é proveniente de um banco de dados relacional que consolida informações multidimensionais sobre *pull requests* enviados como contribuições para projetos de código aberto no GitHub. Os registros dessa base contêm

atributos sobre a contribuição em si, as características do projeto no momento da submissão do *pull request* e o perfil do desenvolvedor responsável pela entrega.

A partir dessa base de dados, esta pesquisa apresenta análises das medidas de interesse Confiança e *Lift*, demonstrando que a probabilidade e a relevância de uma regra podem mudar ao longo do tempo, influenciadas pelas estratégias e práticas adotadas pela equipe principal do software.

3.2. Análise Temporal de Regras de Associação

Para demonstrar que a utilização ativa do arquivo *contributing.md* [Fronchetti et al. 2023] em projetos de código aberto pode favorecer a entrada de novos contribuidores [Soares et al. 2018], está sendo proposta uma metodologia de análise temporal de regras de associação que envolve cinco etapas sequenciais, a saber:

1. Extração das regras de associação da base de dados completa do projeto de software em questão. As regras de associação serão extraídas com limite mínimo de 1% para suporte e confiança, garantindo que os padrões encontrados não sejam aleatórios.
2. Definição dos intervalos temporais: nesta etapa, é realizada uma análise qualitativa do histórico do projeto para identificar marcos temporais relevantes que servirão de referência para o particionamento dos dados. Esses marcos representam eventos significativos (criação ou alteração do arquivo *contributing.md*) e são definidos caso a caso, de acordo com o contexto do projeto e os objetivos da análise. Assim, tanto a quantidade quanto o posicionamento desses marcos pode variar nos projetos, conforme a interpretação do analista responsável.
3. Particionamento dos dados: O conjunto de dados é dividido com base nos marcos temporais definidos, resultando em partições que podem variar tanto na quantidade de registros presentes quanto na duração do período que abrangem. Dessa forma, uma partição pode conter registros de apenas alguns meses, enquanto outra pode abranger anos de dados, dependendo da distribuição temporal das informações.
4. Extração das regras de associação em cada uma das bases particionadas resultantes da etapa anterior.
5. Análise dos padrões: os padrões identificados nas etapas 1 e 4 são organizados e comparados para verificar se houve variação das medidas considerando o conjunto de dados completo e os conjuntos particionados. O objetivo é determinar se a causa estudada resultou em alguma mudança nos padrões observados através da variação das medidas de interesse. No presente estudo, a ação analisada foi a inserção e a modificação do arquivo *contributing.md*, um documento que orienta colaboradores sobre como contribuir com o projeto.

O algoritmo Apriori [Agrawal 1994] foi utilizado para minerar as regras de associação.

3.3. Bases de Dados

Esta pesquisa analisou dados de *pull requests* de três projetos de código aberto, selecionados a partir da base de dados de [Zhang et al. 2020], que havia enriquecido as informações disponíveis na ferramenta GHTorrent [Gousios et al. 2014]. Além desses dados, foram

coletadas informações adicionais, como a identificação e data de criação dos *pull requests*, o *login* do responsável pela contribuição e o *login* do membro da equipe principal que o fechou. Esses dados complementares foram extraídos diretamente da API do GitHub[5]. A seleção dos projetos considerou critérios como a quantidade de *pull requests* e a presença significativa de contribuintes externos, incluindo desenvolvedores iniciantes.

No pré-processamento dos dados, foram excluídos os registros que apresentavam inconsistências. Por exemplo, contribuições que foram "aceitas", mas sem assinatura do revisor. Além disso, foram removidos *pull requests* autoanalisados, uma vez que, nesses casos, as contribuições foram enviadas por membros da equipe principal do projeto.

A Tabela 1 apresenta algumas das características dos projetos analisados: número de *pull requests* (#PR), percentual de rejeição (%Rej.), percentual de *pull requests* enviados por membros da comunidade externa (%PR Externo) e percentual de *pull requests* que representam a primeira contribuição de um desenvolvedor (%FirstPull).

Tabela 1. Características dos projetos selecionados.

Projeto	#PR	%Rej.	%PR Externo	%FirstPull
Serverless[1]	1.632	11,45	54,10	40,37
Matplotlib[2]	6.562	9,35	33,21	13,21
Puppet	6.397	15,81	23,84	8,80

[1] `https://github.com/serverless/serverless`
[2] `https://github.com/matplotlib/matplotlib`

3.4. Atributos Utilizados na Pesquisa

A Tabela 2 apresenta os atributos das bases de dados que são utilizados no processo de extração de regras de associação. Esses atributos fazem parte das condições que formam o antecedente e o consequente das regras.

Tabela 2. Atributos utilizados para extração de regras de associação.

Atributo	Descrição
Status	Registra se a contribuição foi aceita ou rejeitada. Os valores são "*aceito*" ou "*rejeitado*"
Lifetime	Representa o tempo decorrido entre a submissão e o fechamento de um *pull request*. Os intervalos são categorizados da seguinte forma: "*very short*": até 1 hora; "*short*": entre 1h e 1 dia; "*medium*": entre 1 dia e 1 semana; e "*lengthy*": mais de 1 semana.
FirstPull	Indica se o *pull request* enviado é a primeira contribuição de um desenvolvedor no projeto. Os valores são "*true*" ou "*false*".
Tipo de Des.	Indica o tipo de desenvolvedor que enviou o *pull request*, se membro do time principal ou externo. Os valores são "*time principal*" ou "*comunidade externa*".

Cada registro da base possui um atributo que representa a data e a hora de criação do *pull request*. Este atributo foi utilizado para ordenar os registros e identificar os pontos de particionamento da base e não é considerado na extração das regras.

[5]https://docs.github.com/pt/rest?apiVersion=2022-11-28

Para demonstrar a eficácia da metodologia proposta na captura das variações temporais das medidas de interesse, apresentamos uma análise temporal com particionamento baseado nos marcos temporais específicos de cada projeto, os quais foram identificados através de uma análise qualitativa.

Neste artigo, foram utilizados recursos de inteligência artificial generativa, como ChatGPT[6] e Copilot[7], para correção e ajuste do texto.

4. Resultados e Discussão

Os resultados apresentados nas próximas seções estão agrupados por projeto. Para facilitar a identificação de variações das medidas de interesse ao longo do tempo, os gráficos apresentados possuem barras que representam os valores das medidas de interesse avaliadas, resultantes do particionamento com os marcos temporais da base de dados. A linha pontilhada vermelha indica o valor da medida extraída do conjunto de dados completo.

4.1. Resultados do Projeto Puppet

A regra analisada neste projeto é FirstPull = "*true*" $\rightarrow$ Status = "aceito", que representa a relação entre a primeira contribuição de um desenvolvedor no projeto e a aceitação do referido *pull request*.

Ao analisar a base de dados completa, observa-se que a Confiança da regra (linha vermelha na Figura 2), ou seja, a probabilidade de um *pull request* ser aceito quando submetido por um desenvolvedor que contribui pela primeira vez, é de 58%.

Neste projeto, o primeiro *pull request* foi submetido em 28/09/2010 e o último em 28/05/2019. O arquivo *contributing.md* foi inserido em 06/02/2014. O arquivo sofreu algumas alterações ao longo do tempo, mas em 03/08/2018 foi identificada uma alteração que apresenta diretrizes mais claras sobre algumas etapas necessárias para a aceitação de *pull requests*. Com base nesses marcos temporais, a base de dados foi dividida em três partições: a primeira, de 28/10/2010 a 06/02/2014, contém 2.446 registros; a segunda, de 07/02/2014 a 03/08/2018, possui 3.565 registros; e a terceira, de 04/08/2018 a 28/05/2019, é composta por 386 registros.

A análise temporal, apresentada na Figura 2, mostra que a probabilidade de aceitação de uma contribuição submetida por um desenvolvedor inexperiente (*FirstPull*) aumentou de 51% para 64% após a inserção do arquivo *contributing.md*. Após a modificação do arquivo, houve um aumento ainda maior, de 64% para 95%.

Se a análise fosse feita apenas na base de dados completa, o impacto da inserção e atualização do arquivo *contributing.md* no projeto não teria sido identificadoAlexandre .

Ao analisar a base de dados completa (linha vermelha na Figura 3), observa-se que o *Lift* da regra é de 0,69, o que significa que *pull requests* de novos contribuidores têm 31% menos chance de serem aceitos em comparação com a probabilidade de um *pull request* ser aceito no cenário geral, que é de 84,19%, conforme Tabela 1. Esse resultado indica uma tendência negativa, sugerindo que desenvolvedores que contribuem pela primeira vez enfrentam maior dificuldade para ter suas submissões aprovadas.

[6]https://chatgpt.com/

[7]https://copilot.microsoft.com/

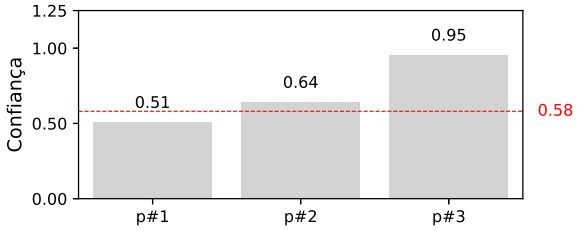

Figura 2. Análise temporal da Confiança da regra: FirstPull = "*true*" → Status = "aceito" no projeto Puppet.

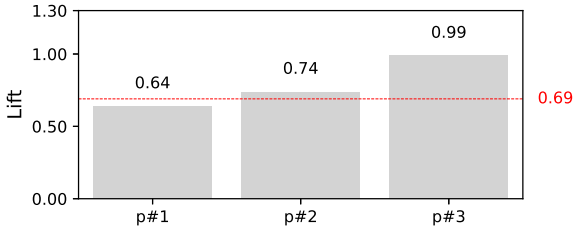

Figura 3. Análise temporal do *Lift* da regra: FirstPull = "*true*" → Status = "aceito" no projeto Puppet.

A análise temporal dessa métrica revela como a relação entre a inexperiência do desenvolvedor e a aceitação do *pull request* evoluiu ao longo do tempo. Após a inclusão do arquivo *contributing.md*, no início da segunda partição, a influência negativa diminuiu, refletida no aumento do *Lift* de 0,64 para 0,74, sugerindo que novos desenvolvedores passaram a ter mais chances de ter suas contribuições aceitas. Com a atualização do documento, essa influência negativa praticamente desapareceu, tornando a aceitação do *pull request* independente do fato de o desenvolvedor ser inexperiente (situação em que o *Lift* é próximo de 1). Essas avaliações indicam que as mudanças implementadas tornaram o processo de contribuição mais claro e acessível para novos desenvolvedores.

As análises da Confiança e do *Lift* para a regra *FirstPull* = "*true*" → Status = "aceito" sugerem que a inclusão e alteração no arquivo *contributing.md*, feitas pela equipe principal, contribuíram para aumentar a taxa de aceitação dos *pull request* enviados por contribuidores novatos no projeto. Essa influência só pôde ser identificada graças à abordagem de análise temporal adotada, que permitiu capturar variações ao longo do tempo.

4.2. Resultados do Projeto Serverless

No projeto Serverless, a regra analisada, Tipo de Des. = "comunidade externa" → Status = "aceito", revela a relação entre o fato de o desenvolvedor que enviou a contribuição ser membro da comunidade externa do projeto e a aceitação do *pull request*.

O primeiro registro da base deste projeto data de 06/10/2015. O arquivo *contributing.md* foi inserido em 09/08/2015, o que significa que os *pull requests* anteriores a essa data foram excluídos durante a etapa de pré-processamento, pois haviam sido autoanalisados. Dessa forma, a inclusão do arquivo não foi considerada como um marco temporal para o particionamento da base.

Entre as alterações realizadas no arquivo *contributing.md*, destaca-se a atualização de 20/05/2016, que incluiu novos pré-requisitos para a aceitação de contribuições, como a necessidade de passar pelo *Travis CI* e *Coveralls*, além de seguir explicitamente os *Code*

Styles do projeto. O não cumprimento dessas diretrizes resultava na rejeição dos *pull requests*. Em 09/03/2017, as recomendações para a submissão de *pull requests* foram novamente atualizadas.

As datas das alterações no arquivo *contributing.md* foram utilizadas para particionar a base de dados de forma a observar os efeitos dessas mudanças. A primeira partição contém 274 registros, abrangendo o período de 06/08/2015 a 19/05/2016. A segunda partição, composta por 503 registros, vai de 20/05/2016 a 08/03/2017. A terceira partição, por sua vez, inclui 855 *pull requests* registrados entre 09/03/2017 e 17/05/2019.

A Figura 4 apresenta a análise da Confiança da regra Tipo de Dev. = "comunidade externa" $\rightarrow$ Status = "aceito". Na base de dados completa, a Confiança é de 0,85, o que significa que os *pull requests* submetidos por membros da comunidade externa têm uma probabilidade de 85% de serem aceitos. No entanto, essa análise geral não permite identificar os efeitos específicos das alterações no arquivo *contributing.md* nessa probabilidade de aceitação. Somente ao observar a evolução temporal dessa métrica é possível avaliar se as modificações implementadas tiveram impacto na aceitação das contribuições externas.

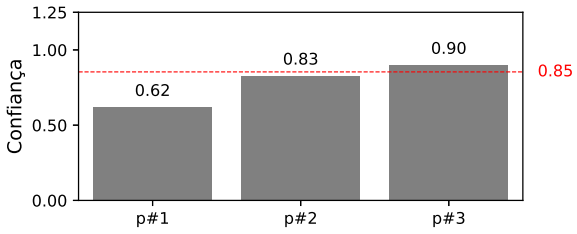

Figura 4. Análise temporal da *Confiança* da regra: Tipo de Dev. = "comunidade externa" $\rightarrow$ Status = "aceito" no projeto Serverless.

A análise temporal da Confiança da regra revela que a primeira modificação no arquivo *contributing.md* elevou a taxa de aceitação desses *pull requests* de 62% para 83%, enquanto a segunda alteração impulsionou ainda mais esse valor, para 90%. Esses resultados demonstram que as mudanças no documento tiveram um impacto positivo, facilitando a aceitação das contribuições feitas por membros externos ao projeto.

A Figura 5 apresenta a análise do *Lift* para a referida regra, avaliando a influência da contribuição ter sido feita por um desenvolvedor que é membro da comunidade externa na aceitação de seu *pull request*. Na base de dados completa, o *Lift* é de 0,96, indicando que essa característica não afeta significativamente a aceitação dos *pull requests* destes desenvolvedores externos. No entanto, ao observar a evolução dessa métrica ao longo do tempo, é possível identificar mudanças na relação entre esses fatores, refletindo o impacto das modificações no arquivo *contributing.md*.

Com o detalhamento proporcionado pela metodologia proposta, percebe-se que a influência do tipo de desenvolvedor na aceitação dos *pull requests* nem sempre foi neutra. Inicialmente, na partição 1, o *Lift* era de 75%, indicando uma influência negativa, ou seja, contribuições de membros da comunidade externa tinham 25% menos chances de serem aceitas em comparação com a probabilidade no caso geral. No entanto, ao longo do tempo, essa influência negativa foi diminuindo, e na partição 3 o *Lift* se aproximou de 1, indicando que a aceitação passou a ocorrer sem viés significativo. Isso sugere que as atualizações no arquivo *contributing.md* ajudaram a tornar o processo mais inclusivo para

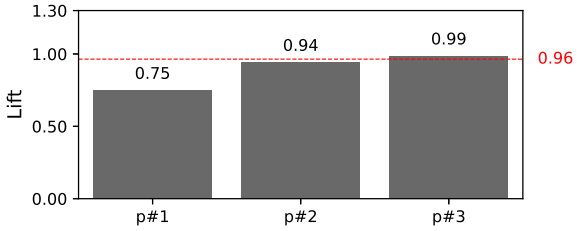

Figura 5. Análise temporal do *Lift* da regra: Tipo de Dev. = "comunidade externa" → Status = "aceito" no projeto Serverless.

novos contribuidores.

4.3. Resultados do Projeto Matplotlib

No projeto Matplotlib, analisou-se a relação entre o tempo de vida muito curto (menos de 24 horas) de um *pull request* e seu autor, considerando dois cenários: contribuições de membros da comunidade externa (Tipo de Des. = "comunidade externa" → Lifetime = "*very short*") e de desenvolvedores que participam pela primeira vez no projeto (FirstPull = "*true*" → Lifetime = "*very short*"). O objetivo é avaliar se a inserção e modificações no arquivo *contributing.md* influenciaram a rapidez na análise dessas contribuições.

A base de dados inclui *pull requests* submetidos entre 19/02/2011 e 31/05/2019. O arquivo *contributing.md* foi adicionado em 29/08/2016 e atualizado em 15/07/2017 com instruções mais detalhadas sobre instalação e reconstrução do código-fonte. As Figuras 6 e 7 apresentam as análises temporais da Confiança, segmentadas conforme esses marcos.

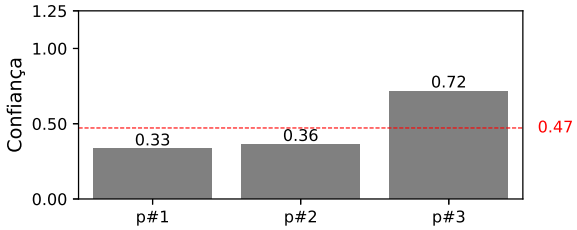

Figura 6. Análise temporal da Confiança da regra: Tipo de Dev. = "comunidade externa" → Lifetime = "*very short*" no projeto Matplotlib.

Na Figura 6, a análise da base de dados completa indica que *pull requests* enviados por desenvolvedores externos têm 47% de chance de serem analisados rapidamente. Temporalmente, essa probabilidade variou: no início do projeto, era de 33%, subindo para 36% após a inserção do *contributing.md* e alcançando 72% após sua atualização. Isso sugere que a adoção de diretrizes mais claras pode ter acelerado o processo de revisão.

Na Figura 7, observa-se que, considerando toda a base de dados, *pull requests* de novos desenvolvedores têm 35% de chance de serem analisados rapidamente. Temporalmente, essa probabilidade era de 29% no início do projeto, subiu para 33% após a introdução do *contributing.md* e atingiu 48% na terceira partição, superando a média geral. Isso sugere que as mudanças no arquivo podem ter facilitado a contribuição, reduzindo erros e retrabalho, tornando o processo de revisão mais ágil.

Assim, a análise temporal permitiu identificar o impacto positivo dessas mudanças, evidenciando que a inserção e atualização do *contributing.md* contribuíram

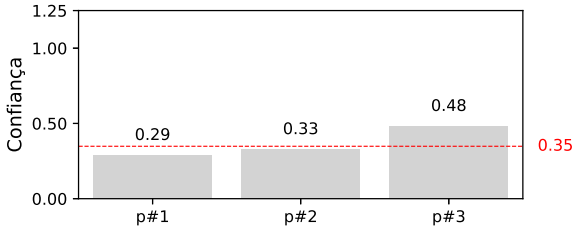

Figura 7. Análise temporal da Confiança da regra: FirstPull = "*true*" → Lifetime = "*very short*" no projeto Matplotlib.

para um aumento na eficiência da revisão de contribuições de novos desenvolvedores. Sem a avaliação temporal das regras de associação, esses efeitos não seriam perceptíveis.

5. Conclusão

Este estudo propõe uma metodologia de análise temporal de regras de associação, utilizando o particionamento do conjunto de dados com base em marcos temporais, com o objetivo de avaliar os efeitos de ações representadas por essas datas. Especificamente, foi investigado o impacto da inserção e modificações do arquivo *contributing.md* em três projetos de código aberto disponíveis no GitHub.

Os resultados mostram que, após a inserção e alterações no arquivo *contributing.md*, houve uma melhora nos padrões de aceitação e tempo de vida dos *pull requests* submetidos por desenvolvedores externos ou que contribuíam pela primeira vez no projeto. Embora não seja possível afirmar que os desenvolvedores passaram a seguir as diretrizes explicitadas, os dados sugerem uma correlação entre a inserção e atualização do arquivo e a melhora nesses indicadores. Em todos os casos ilustrados, as alterações no *contributing.md* refletem o amadurecimento das diretrizes dos projetos, o que pode ter contribuído para facilitar a entrada de novos colaboradores.

Os resultados evidenciam o potencial da metodologia para analisar o efeito de uma ação sobre os padrões do projeto. Como trabalho futuro, propõe-se analisar o sentido inverso: a partir da identificação da variação dos padrões, investigar, qualitativamente, a causa do fenômeno.

Os resultados não podem ser generalizados para todos os projetos de código aberto, mas oferecem uma abordagem útil para entender padrões em bases de dados, considerando a evolução temporal.

Referências

Agrawal, R. (1994). Fast algorithms for mining association rules. In *In Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499.

Ale, J. M. and Rossi, G. H. (2000). An approach to discovering temporal association rules. In *Procs. of the 2000 ACM Symp. on Applied computing-Volume 1*, pages 294–300.

Ford, D., Behroozi, M., Serebrenik, A., and Parnin, C. (2019). Beyond the code itself: How programmers really look at pull requests. In *2019 IEEE/ACM 41st Int. Conf. on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 51–60.

Fronchetti, F., Shepherd, D. C., Wiese, I., Treude, C., Gerosa, M. A., and Steinmacher, I. (2023). Do contributing files provide information about oss newcomers' onboarding barriers? In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 16–28.

Gonçalves, S., Soares, D., and Silva, D. (2021). Temporal analysis on pull request patterns: an approach with sliding window. In *Proceedings of the 15th Brazilian Symposium on Software Components, Architectures, and Reuse*, pages 90–99.

Gousios, G., Pinzger, M., and Deursen, A. v. (2014). An exploratory study of the pull-based software development model. In *Proceedings of the 36th international conference on software engineering*, pages 345–355.

Kotsiantis, S. and Kanellopoulos, D. (2006). Association rules mining: A recent overview. *GESTS Int. Transactions on Computer Science and Engineering*, pages 71–82.

Liu, X., Feng, F., Wang, Q., Yager, R. R., Fujita, H., and Alcantud, J. C. R. (2021). Mining temporal association rules with temporal soft sets. *Journal of Mathematics*, page 7303720.

Rastogi, A., Nagappan, N., Gousios, G., and van der Hoek, A. (2018). Relationship between geographical location and evaluation of developer contributions in github. In *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*, pages 1–8.

Segura-Delgado, A., Gacto, M. J., Alcalá, R., and Alcalá-Fdez, J. (2020). Temporal association rule mining: An overview considering the time variable as an integral or implied component. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1367.

Soares, D. M., de Lima Júnior, M. L., Murta, L., and Plastino, A. (2015). Acceptance factors of pull requests in open-source projects. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 1541–1546.

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

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*, pages 32–43.

Steinmacher, I., Pinto, G., Wiese, I. S., and Gerosa, M. A. (2018). Almost there: A study on quasi-contributors in open source software projects. In *Proceedings of the 40th international conference on software engineering*, pages 256–266.

Tsay, J., Dabbish, L., and Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the 36th international conference on Software engineering*, pages 356–366.

Zhang, X., Rastogi, A., and Yu, Y. (2020). On the shoulders of giants: A new dataset for pull-based development research. In *Proceedings of the 17th international conference on mining software repositories*, pages 543–547.