UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RAFAEL OLIVEIRA SANTOS

ANÁLISE DE SENTIMENTOS EM REPOSITÓRIOS DO GITHUB

CAMPINA GRANDE - PB

2021

RAFAEL OLIVEIRA SANTOS

ANÁLISE DE SENTIMENTOS EM REPOSITÓRIOS DO GITHUB

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. Eanes Torres Pereira.

CAMPINA GRANDE - PB

2021

S237a Santos, Rafael Oliveira.
Análise de sentimentos em repositórios do GitHub./ Rafael Oliveira Santos. – 2021.

10 f.

Orientador: Prof. Dr. Eanes Torres Pereira.
Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Repositórios do GitHub. 2. Ambiente GitHub. 3. Pull-requests. 3. Análise de sentimentos - repositórios. 4. Processamento de linguagem natural. 5. Comentários pull-requests. 6. Valence, Arousal and Dominance. 7. Dimensão de Valência. 8. BigQuery – Google. I. Pereira, Eanes Torres. II. Título.

CDU:004(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

RAFAEL OLIVEIRA SANTOS

ANÁLISE DE SENTIMENTOS EM REPOSITÓRIOS DO GITHUB

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Eanes Torres Pereira
Orientador – UASC/CEEI/UFCG

Professor Dr. Herman Martins Gomes
Examinador – UASC/CEEI/UFCG

Professor Dr. Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 25 de março de 2021.

CAMPINA GRANDE - PB

RESUMO (ABSTRACT)[1]

Pull-requests are suggestions for changes or improvements, for a repository of a project on the *GitHub* environment. These suggestions can be commented on by developers, and they can express different sentiments in their comments. In this study, comments present in *pull-requests* were analyzed in order to understand whether positive comments may or may not influence the acceptance of *pull-request*. For this, data extraction techniques, use of state-of-the-art approaches to deal with *Big Data* and pre-trained tools to produce this analysis were applied. The final result verified in this study showed that, yes, there is a relationship between positive comments and the successful acceptance of *pull-requests*. From a covariance calculation, it was understood that there is a positive correlation between the "*score variable*" and the "success variable". Rejecting, through a hypothesis test *T-Student*, the null hypothesis that the average of comments expressing positive sentiments and expressing negative sentiments for *pull-requests* have equal averages. It was understood that if the means between the two variables are different, this is strongly associated with different behaviors, if the comments have sentiments with different intensities.

[1] Caso seu artigo esteja em inglês, coloque aqui o resumo em português; caso esteja o artigo em português, coloque aqui o resumo em inglês.

Análise de sentimentos em repositórios do GitHub

Rafael Oliveira Santos*
rafael.santos@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba

Eanes Torres Pereira*
eanes@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba

RESUMO

Pull-requests são sugestões de mudanças ou melhorias, para um determinado repositório, de um projeto no ambiente do *GitHub*. Essas sugestões podem ser comentadas por outros desenvolvedores que, por sua vez, podem expressar diferentes sentimentos nos seus comentários. Neste estudo, foram analisados comentários presentes em *pull-requests* com o intuito de compreender se comentários positivos podem, ou não, influenciar na aceitação do *pull-request*. Para isso, foram aplicadas técnicas de extração de dados, uso de abordagens do estado da arte para lidar com *Big Data* e ferramentas pré-treinadas para produzir essa análise. O resultado final verificado neste estudo mostrou que, sim, existe uma relação entre comentários positivos e o sucesso na aceitação dos *pull-requests*. A partir de um cálculo de covariância, entendeu-se que existe uma correlação positiva entre as "variáveis de *score*" com a "variável de sucesso". Rejeitando, através de um teste de hipótese *T-Student*, a hipótese nula de que as médias de comentários expressando sentimentos positivos e expressando sentimentos negativos para pull-requests possuem médias iguais. Entendeu-se que se as médias entre as duas variáveis são diferentes, isso está fortemente agregado a comportamentos diferentes, caso os comentários possuam sentimentos com intensidades diferentes.

1 INTRODUÇÃO

Uma das principais ferramentas utilizadas no desenvolvimento de softwares, são as chamadas "ferramentas de versionamento de código". Com elas consegue-se produzir, gerenciar e versionar projetos de software e, com isso, torna-se possível entender todo um ecossistema de interações entre desenvolvedores. Segundo dados do *GitHub* são mais de 65 milhões de desenvolvedores, mais de 3 milhões de organizações e 200 milhões de repositórios. Todo esse ecossistema se comunicando e produzindo *metadata* sobre as próprias relações e interações, dentro do que é a uma das maiores e mais populares plataformas de hospedagem de código do mundo.

Para o estudo apresentado neste artigo, foram analisados cerca de 50 mil *pull-requests* que somaram mais de 200 mil comentários, com o objetivo de elucidar se comentários positivos, presentes nos *pull-request*, podem – ou não – influenciar no sucesso dos mesmos. Quanto ao termo "sucesso", pode-se compreender como sendo o fluxo de gerar um *pull-request* e, em algum momento, este ser integrado a *branch* principal do projeto. Quanto à análise dos comentários positivos, na grande maioria dos estudos recentes pode-se observar que estes são rotulados manualmente. Ou seja, os projetos que realizam estudos onde a interação entre os desenvolvedores num *pull-request* são avaliadas, o fazem do ponto de vista humano. Em outras palavras, lendo cada comentário, coletado e rotulando manualmente entre os três tipos de comentários existentes: negativo, positivo e neutro. No presente estudo, optou-se por uma análise automática, deixando essa responsabilidade de rotular para as ferramentas de processamento de linguagem natural. Para validar a análise, foi executado um teste de hipótese *T-Student* e um cálculo de covariância, com o objetivo de responder à seguinte questão:

QP: Os comentários para pull-requests que obtiveram sucesso possuem sentimentos positivos e negativos similares? A resposta que mostrou-se com a mais forte indicação foi a de que, sim, comentários positivos afetam o sucesso dos *pull-requests*.

Análise de Sentimentos, Maio 2021, Brazil

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Análise de sentimentos

O processo de análise de sentimentos é muito mais do que apenas ler um texto e defini-lo como “positivo” ou “negativo” baseado na literalidade da oração. Na análise de sentimentos a interpretação e a subjetividade do texto precisam ser elementos considerados, para que se extraia toda a real intenção do objeto que está sendo analisado. Saber receber, classificar, separar e organizar os dados analisados torna-se de vital importância para uma boa análise de sentimentos. Para que isso ocorra com a maior eficiência possível, também é necessário escolher a melhor metodologia e as melhores ferramentas.

Existem diversas metodologias para a análise de sentimentos, logo, o trabalho inicial deve ser a identificação sobre qual tipo de dado será tratado durante o estudo e de qual tipo de resultado se espera após a apuração. Um dos dados mais comumente analisados, porém não menos complexo, são as palavras. Para esse tipo de dado são usadas, principalmente, dois tipos de abordagens metodológicas: as baseadas nos Processamentos de Linguagem Natural (NLP), e as baseadas em *Machine Learning* (ML).

Ainda no escopo da separação de dados, será observado a partir da Seção 4 o quão computacionalmente complexa é a separação dessas informações. Pois, para consolidar a presente análise, tornou-se necessário isolar a língua que se estava sendo analisada e, como é de se esperar, optou-se pelo uso da principal. Nesse caso, a língua com mais pesquisas e mais utilizada na área é a língua inglesa. Assim sendo, muita atenção foi necessária na separação dos comentários, pois um "no" na língua portuguesa, por exemplo, que poderia assumir um valor neutro; Na língua inglesa, representaria um valor negativo "no". Então, garantir que irá se analisar apenas esse tipo de comentário é essencial.

Dentro da análise de sentimentos podemos checar não só sentimentos positivos e negativos como também:

- Polidez da escrita - o nível de cortesia que é conseguido enxergar a partir da análise.
- *Valence, Arousal and Dominance.(VAD)* "...A dimensão de valência representa o quão agradável ou desagradável é o estímulo que originou a emoção. Excitação representa a intensidade da resposta emocional ao estímulo. Dominância representa a posição percebida em relação ao estímulo no que diz respeito a controle ou submissão. Uma emoção em particular é representada pela atribuição de valores para cada uma das dimensões, em uma escala que vai de 0 a 10."[2]

2.2 Processamento de Linguagem Natural(NLP)

Uma das principais e mais utilizadas metodologias para a análise de sentimentos em palavras é a NLP (Natural Language Processing). O Processamento de Linguagem Natural (NLP) pode ser compreendido, de maneira geral, como uma ramificação da área de Inteligência Artificial que pode ser facilmente relacionada com *Machine Learning* e Deep Learning. Existem hoje diversos *toolkits* – pacotes de ferramentas desenvolvidos para facilitar e/ou conectar tarefas – com diferentes linguagens de programação base, que são utilizados ao longo dos projetos que se valem dessa metodologia. No estudo apresentado neste artigo foi utilizado especificamente o NLTK[1], pacote inteiramente em Python.

Através dessa metodologia é possível analisar, não apenas a forma literal das palavras, como também as subjetividades dentro do texto, o que podem indicar diferentes tipos de sentimentos expressados através das Figuras de linguagem utilizadas como: Frustração, ironia, sarcasmo, descontentamento, aceitação, empolgação, entre outros.

A metodologia NLP pode ser inserida e utilizada em várias outras aplicações, como por exemplo:

- Extração de entidades como nome de locais e palavras-chave
- Pesquisa semântica - muito usado em motores de busca, ajuda no refinamento da busca por resultados esperados.
- ChatBots - Entender perguntas simples e respondê-las, para isso é preciso NLP para identificar a pergunta que está sendo feita com a resposta adequada, quanto mais acurada melhor é a resposta.

3 TRABALHOS RELACIONADOS

Como visto em *Destefanis et al*[1], é possível verificar a diferença entre um desenvolvedor – que está envolvido numa *issue* – e um usuário comum – que não contribui com aquele repositório – através de seus comentários. Partindo das métricas: *emotions*, *politeness*, *sentiment*, *valence*, *arousal* e *dominance* (VAD). Consegue-se identificar que: usuários que única e exclusivamente comentam no repositório, tendem a ser menos educados, menos positivos e, de modo geral, expressam um nível mais baixo de emoções em seus comentários do que os desenvolvedores que interagem diretamente com *commits* e *pull-request* na *branch* principal.

Em *Jin Ding et al*[3], é apresentado o desenvolvimento da ferramenta SentiSW onde – a partir de uma entrada de dados rotulada manualmente – consegue-se entregar valores em nível de entidade. O que torna possível responder questões como: se o comentário realizado é positivo, negativo ou neutro; assim como se a entidade indicada é uma entidade "Pessoa" ou "Projeto". Todo o pré-processamento dos dados é parte crucial para os resultados, e essa etapa é englobada diretamente dentro da própria ferramenta, o que proporciona a remoção de caracteres que não pertencem à língua inglesa, remoção de *snippets* de códigos como "Markdown", remoção de *stopwords* e remoção de *urls*. Os levantamentos relacionados ao tratamento dos dados foram de vital importância para este projeto como, por exemplo, a remoção de *stopwords*. Como poderá ser visto na Seção 4, esses levantamentos serviram de inspiração para boa parte das técnicas usadas nesse artigo.

Em *Marco Ortu et al*[6] segue-se um viés de análise muito semelhante à *Destefanis et al*[1], porém, uma divergência bastante notável encontra-se na possibilidade de perceber e avaliar a existência – não apenas das "glosas" relacionadas aos comentários – como também a identificação dos *emojis* utilizados dentro destes, como um símbolo representativo de algo positivo ou negativo. Após descobertos os pontos acima citados, os mesmos tornaram-se de grande importância para as tomadas de decisão durante o desenvolvimento das análises do estudo. Optou-se por manter qualquer tipo de *emoji* presente nos comentários. Focou-se em mantê-los, já que os mesmos são coletados em formato de texto. Exemplo: :smile:, esse *emoji* é coletado nesse formato dentro dos comentários e ele por si só já tem um valor semântico enorme. Isso difere um pouco do que é relatado em *Jing Ding et al*[3], pois o mesmo remove qualquer tipo de *markdown*, *emoji*, que encontre. Nesse estudo foi mantido.

Em *Vinayak Sinha et al*[8], foram analisadas algumas informações. Como, por exemplo, em quais dias da semana os usuários tendiam a se expressar com comentários mais negativos. Para o tratamento desses dados, um dos principais pontos levantados foi o uso da ferramenta *SentiStrength* (SE) – uma ferramenta de análise de sentimentos projetada principalmente para engenharia de software –. Esta mesma ferramenta que também é citada em *Destefanis et al*[1] para a medição de sentimentos; Já em *Jin Ding et al*[3], ela é citada com grande consideração como uma importante ferramenta. Contudo, como a intenção apresentada foi a da criação de uma nova ferramenta, que pudesse avaliar também o nível de entidade, esta ferramenta, apesar de citada, não chega a ser utilizada para o desenvolvimento daquele projeto.

4 METODOLOGIA

A metodologia utilizada ao longo deste trabalho consiste em estudar, entender e analisar dados referentes a repositórios do GitHub, utilizando-se de representações gráficas e refletindo acerca dos resultados encontrados através de análises quantitativas.

Primeiramente foi obtido um conjunto inicial base, fornecido pelo projeto GHTorrent[5], com variadas informações – que vão desde dados do projeto em si, até mesmo os observadores e usuários presentes no *GitHub*, separados em classes –. Assim sendo, inicialmente foi preciso partir para o cruzamento de dados e, com isso, produzir informações quantitativas sobre o mesmo. Explicando de forma mais clara, dentro dos dados não existe a quantidade de *pull-requests* de um projeto, por exemplo. Porém, existe a "classe projeto" que possui um *id* e a "classe *pull-request*" que possui o *id* do projeto,

[1]https://www.nltk.org/, Acessado em 08/04/2021, 18:53

onde aquele *pull-request* foi criado. Sendo assim, consegue-se obter os dados de uma classe com outra através desses *ids* e, com isso, contabilizar a quantidade de *pull-requests* daquele projeto.

Logo após, projeta-se a base que será utilizada para o estudo empírico, onde serão realizados os pré-processamentos com foco nas colunas referentes aos comentários de *pull-requests*. Ao final, serão produzidas duas novas bases de apoio para a produção das análises quantitativas dos dados, que serão utilizadas para gerar o entendimento do quanto comentários positivos podem – ou não – afetar na aceitação dos *merge-requests*. Com isso, existem três fases na metodologia adotada: coleta inicial, pré-processamentos e *insights*.

4.1 Coleta inicial

Para a coleta inicial foi utilizada a ferramenta do Google chamada "BigQuery" [7] – um data *warehouse* para *petabytes* de informações na nuvem em conjunto com uma plataforma de consulta –. Uma vez que mostrou-se necessária a utilização de uma base gigantesca de informações relacionadas a projetos e usuários do *GitHub*, tornou-se essencial o emprego de ferramentas de *BigData* para a coleta inicial dos dados. E, assim, caso houvesse a aplicação de ferramentas inferiores, o tempo gasto para processamento dos dados se tornaria inviável.

Nessa fase, o objetivo é a busca pelos comentários presentes em *pull-requests*, principalmente naqueles que conseguiram ser anexados com a *branch* principal do projeto. Visto que, o objetivo aqui é entender se existe uma relação entre comentários positivos e a aceitação dessas requisições.

Na Figura 1, é possível observar a relação presente entre as duas classes a partir dos seus "`pull_request_id`", com enfoque na classe "`pull_request_history`" e no atributo "action". Onde busca-se os "action" com valores iguais a "merged", e no "`pull_request_comment`" busca-se o atributo "body". Nessa fase, são eliminados comentários presentes no atributo "body" que possuam caracteres não-ingleses, presentes na linguagem russa e mandarim, por exemplo.

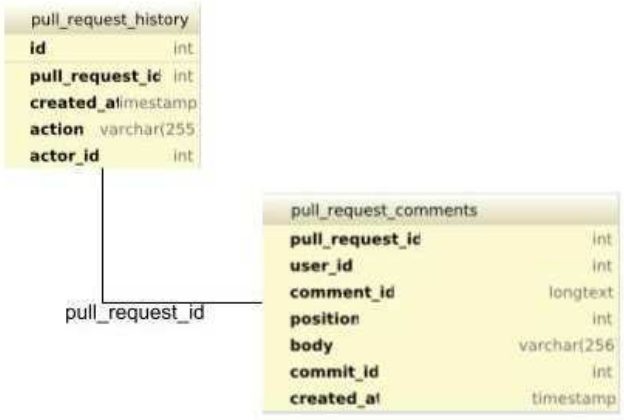

Figura 1: Classes usadas e relacionamentos entre elas.

Tabela 1: Base inicial tratada

Informações usadas	Quantidade
pull-requests	397118
comentários	2637650

4.2 Pré-Processamento

Nessa fase, inicialmente foram aplicados métodos para a filtragem dos *pull-requests* com palavras no idioma inglês. Logo depois, foram excluídas as *stopwords*, caracteres especiais e linhas nulas. Com os comentários já tratados, usou-se uma ferramenta para análise de sentimentos chamada "Sentiment Intensity Analyzer" (SIA), presente no pacote NLTK, que retorna um "*score* de sentimento" para cada frase presente neste *dataset*.

Esse *score* gerado tem um valor de [-1, 1], onde valores negativos indicam o quão negativo é aquele comentário, valores positivos indicam quão positivo é aquele comentário e o valor 0 indica neutralidade naquele comentário. Cada comentário de cada *pull-request* é analisado separadamente. Ou seja, nos dados existe a possibilidade de um "`pull_request_id`" possuir vários *scores* diferentes por possuir vários comentários.

Ainda nessa fase, foram identificados também os dados dos "`pull_request_id`" – aqueles que obtiveram êxito no seu pedido em integrar a *branch* principal do projeto –. Com isso, foi gerada a base para a fase subsequente. Onde, a coluna "success" indica por 1 o êxito de se integrar na *branch* principal e por 0 o fracasso do mesmo.

4.3 Insights

Nessa fase, foram formatados os dados da fase anterior para visualização de alguns resultados. Inicialmente foram adicionadas as quantidades de caracteres por comentário – a ideia era usar na análise apenas os comentários com mais de 10 caracteres –, foram rotulados os *pull-requests* que conseguiram ser integrados a *branch* principal em "sucesso" e "fracasso", assim como, também foram rotulados os comentários entre: negativo, positivo e neutro.

Como pode-se observar na Figura 2, que representa a distribuição do número de caracteres nos comentários. A maior concentração dos dados está situada nos comentários com valores entre 40 caracteres, até um valor um pouco maior do que 100. Assim sendo, mostrou-se como ideal optar apenas por essa faixa durante o estudo. Inicialmente, como a intenção era manter a maior quantidade de dados possível, comentários menores também foram avaliados. Contudo, após algumas revisões manuais, percebeu-se que a maioria dos comentários com valores muito pequenos eram compostos de palavras como "ok", "this", entre outros termos desconexos, e que esses dados acabavam por gerar muitos valores neutros, que atrapalhavam durante a avaliação. Essa situação se torna evidente na Figura 3, onde é possível perceber e uma quantidade consideravelmente grande de comentários neutros, que até mesmo ultrapassam os outros tipos de comentários. Claro que quanto mais polidez e/ou mais objetivo é o texto, mais neutro ele fica e isso é também explorado em [1]. A ideia aqui não foi a de acabar com todos os neutros possíveis desse *dataset*. Contudo, como a avaliação do que é neutro, positivo e negativo está totalmente a cargo de modelos, tornou-se

possível optar por esse tipo de abordagem. Por isso, em sequência, será explicado como foram resolvidos os problemas com os neutros.

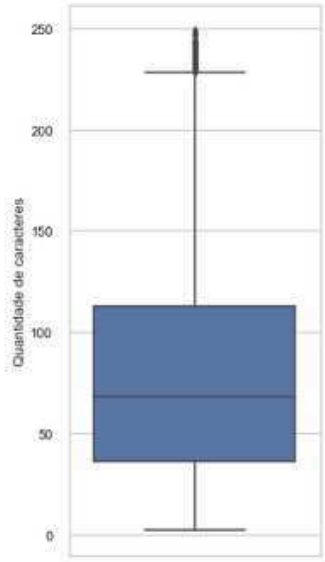

Figura 2: Distribuição de caracteres no *dataset*.

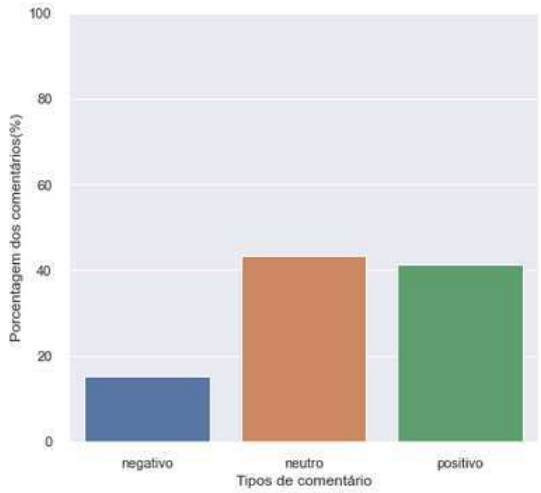

Figura 3: Porcentagem de comentários por tipo antes da aplicação das técnicas.

Mesmo com as técnicas utilizadas voltadas para o *score*, citadas anteriormente, ainda havia uma quantidade de neutros bastante considerável. Então, aplicou-se um segundo modelo verificador. Neste, foi separado um novo *dataset* com os valores de comentários classificados como "neutro" e executou-se o modelo SentiStrength-SE [9], única e exclusivamente nos valores neutros. Tudo isso com o objetivo de aproveitar o máximo de comentários durante a avaliação.

Para uma melhor compreensão da metodologia utilizada, é necessário compreender um pouco sobre a ferramenta *SentiStrength-SE* e, para tanto, é necessário explicar que a mesma foi projetada para uso em palavras voltadas para a área de Engenharia de Software, isso explica a utilização do sufixo SE no nome do mesmo.

Gerando um par de valores para uma frase inteira, essa tupla de resultados representam a menor e maior nota de sentimento positivo e negativo envolvidos na frase, como visto em [8]. Com esse resultado da tupla para chegar ao cálculo final, simplesmente somaram-se esses dois valores resultantes e chegou-se ao valor final. Para esse caso em específico, se a soma for igual a 0 entende-se o resultado como neutro, se for positivo ou negativo ele irá prosseguir para fase final da avaliação e será contado para chegarmos a resposta do QP.A Tabela 2 é um exemplo do *dataset* de suporte que foi usado para diminuição da quantidade de neutros usando o *SentiStrength*.

Com os *scores* assim gerados para cada comentário, todos estes foram somados e então divididos pelo número de comentários, desta maneira, esses dados passaram por um agrupamento dos "`pull_request_id`", com o objetivo de obter um único *score* relacionado a cada *pull-request* e, assim, analisar se os *pull-requests* com mais comentários positivos obtiveram mais sucesso do que os com comentários negativos.

Após isso a coluna de *score* também sofreu transformações em detrimento de sua magnitude, foi utilizada uma técnica baseada em *MinMaxScaler*, para assim colocar os valores de *score* em magnitude iguais, ou seja, *scores* negativos e positivos depois das transformações, passam a possuir um range de valores entre 0 e 1. Isso será importante para a aplicação do teste de hipótese *T-Student*.

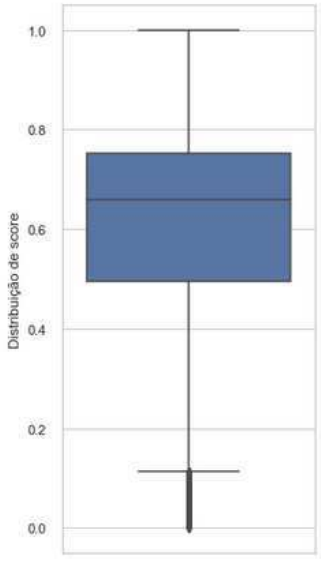

Figura 4: Distribuição do *score* após o uso do MinMaxScaler (Anteriormente os valores variavam entre o intervalo [-1, 1])

Tabela 2: Exemplo das tuplas de valores do SentiStrength-SE

comentário	SentiStrengthScore
The sultan did not know what to do.	[3, -1]
I burst out under blue skies.	[1, -1]

Tabela 3: Exemplo dos dados usados antes da aplicação do MinMaxScaler

pull_request_id	success	*score*	type_comment
0	1	0.456	positive
1	1	-0.233	negative
2	0	0.0	neutral
3	0	-0.15	negative

Na tabela 3, são apresentados os dados que são utilizados para análise, para cada *id* temos um valor único de *score*. sucesso e tipo de comentário. Ou seja, no id = 0, é percebido que a *pull-request* que obteve-se sucesso em ser integrada a *branch* principal, possui um *score* positivo e o tipo de comentário no geral para esse *pull-request* é positivo.

5 RESULTADOS

Depois de todas as transformações realizadas nos dados, com o intuito de manter o máximo possível de informações que fossem úteis para avaliação do QP, nessa Seção será avaliado o seguinte questionamento:

QP: Os comentários para pull-requests que obtiveram sucesso possuem sentimentos positivos e negativos similares?

Para responder essa pergunta, serviu-se de um teste de hipótese conhecido como teste "T de *Student*". Esse teste consiste em verificar se as médias de duas amostras independentes são significativamente diferentes. Ele consiste em primeiramente aplicar uma hipótese nula, que representará se as médias têm distribuições iguais. Caso o *p-value*[2] – para estatística, *p-value* é uma medição de confiança quanto ao resultado de um teste – retornado pela função aplicada nas amostras, que pretende-se testar. Após isso, comparamos seu valor *p-value* e verificamos se o mesmo é inferior a 0.05. Rejeita-se a hipótese nula e assume-se que a hipótese alternativa é a válida.

Abaixo pode-se observar uma breve separação de cada hipótese para melhor entendimento:

Hipótese nula. *As médias dos scores são iguais para pull-requests com comentários positivos e negativos.*

Hipotese. *As médias dos scores são diferentes para pull-requests com comentários positivos e negativos.*

Com a intenção de avaliar se comentários positivos afetam na aceitação do *pull-request*, foram separadas duas amostras: A primeira são os *pull-requests* com média de *score* positiva de comentários, que foram aceitos na integração com a *branch* principal. A segunda acontece com média de *score* negativa nos comentários dos *pull-requests* que foram aceitos. Aplicando o teste *T-Student* nessas duas amostras, obteve-se um valor de *p-value* bem pequeno, muito próximo do valor zero. Para facilmente visualizar o resultado, vamos adotar que o nosso *p-value* = 0. Logo, percebendo que o valor é inferior a 0.05, como descrito acima, pode-se – de acordo com o teste – rejeitar a hipótese nula criada, pois quanto menor o *p-value*, maior é a confiança na rejeição da hipótese nula, adotando então a em que as duas amostras possuem médias diferentes. Isso mostra que as amostras possuem comportamentos diferentes. Esperava-se, para um caso hipotético, que os comentários não influenciassem na aceitação das médias iguais.

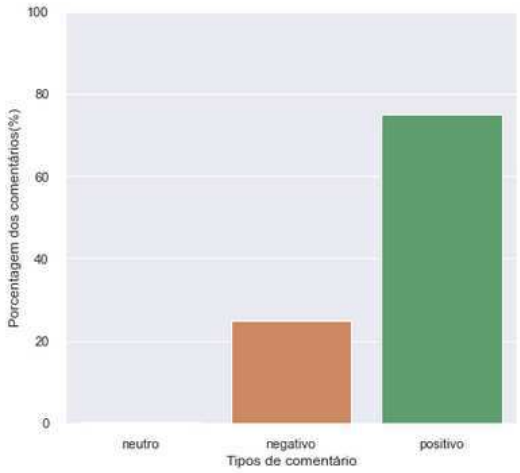

Figura 5: Porcentagem de comentários por tipo que obtiveram sucesso.

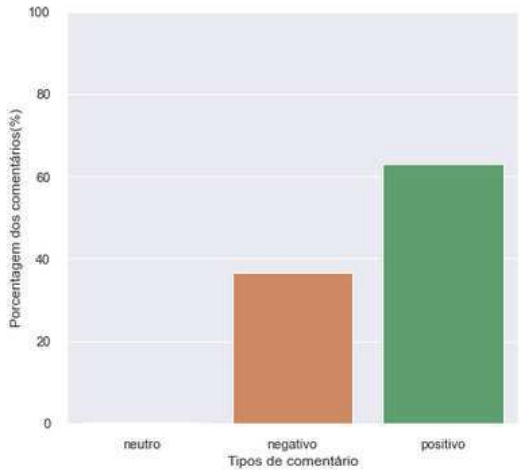

Figura 6: Porcentagem de comentários por tipo que fracassaram no seu pull-request.

Na Figura 5, pode-se observar que a quantidade de *pull-request* que obtiveram sucesso com comentários positivos, também é uma

[2]https://www.simplypsychology.org/p-value.html, Acessado em 12/03/2021, 17:24

demonstração visual do êxito presente na resposta a **QP**. Ainda dentro desse escopo, realizou-se mais uma avaliação, onde foi calculada a covariância entre a métrica de sucesso com o valor de *score*. Com o objetivo de verificar a existência de uma correlação. Antes do resultado final e verificando os resultados anteriores, espera-se que o resultado seja pelo menos positivo, ou seja, que tenham uma correlação positiva.

Na Figura 6, pode-se enxergar um aumento visível da quantidade de comentários negativos, quando torna-se a ótica para os *pull-requests* que fracassaram na sua integração.

6 TRABALHOS FUTUROS

Espera-se que em trabalhos futuros, com rótulos de comentários menos automáticos ou modelos treinados especialmente para produções do *GitHub* [4], consiga-se obter uma forte relação entre os comentários positivos e a aceitação dos mesmos. Pois, ainda existem limitações no entendimento da máquina acerca dos sentimentos, comentários que contenham ironia, sarcasmo e etc, que são inerentes ao ser humano, "aos olhos da máquina" um comentário como este pode passar despercebido e afetar nos resultados mais acurados.

REFERÊNCIAS

[1] Giuseppe Destefanis, Marco Ortu, David Bowes, Michele Marchesi, and Roberto Tonelli. 2018. On Measuring Affects of GitHub Issues' Commenters. In *2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. 14–19.

[2] Guilherme Augusto Dias. 2018. Análise de Sentimento em Artefatos de Software. Monografia (Bacharel em Ciênciada Computação), UFRGS (Universidade Federal do Rio Grande do Sul), Porto Alegre, Brazil.

[3] Jin Ding, Hailong Sun, Xu Wang, and Xudong Liu. 2018. Entity-Level Sentiment Analysis of Issue Comments. In *2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. 7–13.

[4] GitHub. 2008. GitHub, Where the world builds software. In *)*. https://github.com/

[5] Georgios Gousios. 2013. The GHTorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13)*. 233–236. Best data showcase paper award.

[6] Marco Ortu, Giuseppe Destefanis, Daniel Graziotin, Michele Marchesi, and Roberto Tonelli. 2020. How do you Propose Your Code Changes? Empirical Analysis of Affect Metrics of Pull Requests on GitHub. *IEEE Access* 8, 110897–110907.

[7] Google Big Query. 2012. Big Query. https://console.cloud.google.com/bigquery

[8] Vinayak Sinha, Alina Lazar, and Bonita Sharif. 2016. Analyzing Developer Sentiment in Commit Logs. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. 520–523.

[9] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment Strength Detection in Short Informal Text. *J. Am. Soc. Inf. Sci. Technol.* 61, 12 (Dec. 2010), 2544–2558.