

# Avaliação de Estratégias Automáticas para Visualização de Resultados de Algoritmos de Agrupamento

Igor Luiz Costa Ribeiro<sup>1</sup>, Lesandro Ponciano<sup>1</sup>

<sup>1</sup>PUC Minas em Contagem  
Bacharelado em Sistemas de Informação

igorlcr@gmail.com.br, lesandrop@pucminas.br

**Resumo.** *A utilização de algoritmos para auxiliar o usuário na tomada de decisões tem sido aplicada, porém devido à complexidade de tais algoritmos, seus resultados podem não ser compreendidos. A aplicação do requisito de explicabilidade tem como um de seus objetivos promover explicações para tais resultados. Uma das dimensões da explicabilidade é a visualização da explicação. Este trabalho apresenta uma solução que possa gerar visualização para resultados de algoritmos de agrupamento por meio de árvores de decisão a fim de facilitar a compreensão dos resultados por parte dos usuários. A solução é implementada em um estudo de caso de agrupamento e classificação de programadores na plataforma GitHub. Como forma de avaliação do sistema, emprega-se o método de questionário SUS. Os resultados a aplicação do questionário SUS mostram que a solução apresentada possui um grau de usabilidade de 85, considerado excelente pelos parâmetros do SUS. Com este trabalho, motivamos a utilização de visualização para aderência ao requisito de explicabilidade em sistemas a fim de contribuir para a compreensão do usuário e evitar que ele seja controlado pelos algoritmos.*

## 1. Introdução

Atualmente, o emprego da inteligência artificial está presente nos mais diversos sistemas de informação, tais como: *e-commerces*, *sistemas de recomendação*, *sistemas tutores inteligentes* [Batista 2017]. A utilização de algoritmos para tornar um sistema de informação “inteligente” tem se tornado cada vez mais comum. São exemplos disso algoritmos como a árvore de decisão [Gilpin 2018] e o de agrupamento [Faceli 2011]. A compreensão de algoritmos de inteligência artificial é muitas vezes não trivial [Gilpin 2018]. Sem entender como os algoritmos funcionam, as pessoas podem ser induzidas ao erro ou podem ser controladas pelos algoritmos. A partir disso surge a demanda por um novo requisito no desenvolvimento de sistemas que é chamado de “requisito de explicabilidade”. Explicabilidade pode ser definido como o requisito de um sistema que pode auto-explicar as suas funcionalidades [Gilpin 2018].

Os algoritmos de aprendizado em geral podem ser definidos de duas formas, algoritmos de aprendizado supervisionado e algoritmos de aprendizagem não supervisionado [Faceli 2011]. Os algoritmos de aprendizado supervisionado tem como pré-requisito a entrada de dados e que existe uma classe definida. Já nos não

supervisionados, os dados não são previamente classificados. Exemplo de algoritmos não supervisionados são os algoritmos de agrupamento. Tais algoritmos procuram agrupar pontos de dados de acordo com a similaridade entre eles [Jain 1991]. Exemplo de algoritmos supervisionados são os algoritmos de classificação. Tais algoritmos procuram, através de dados já previamente rotulados, construir um modelo para classificar novos casos [Meira 2014].

O problema tratado nesta pesquisa é a **falta de explicabilidade de algoritmos de agrupamento**, ou seja a baixa compreensão das pessoas da dinâmica e dos resultados de algoritmos de agrupamento. Um primeiro passo para a explicabilidade é a visualização de explicações. A ênfase deste trabalho está em tais visualizações. O requisito de explicabilidade pode servir a sete objetivos distintos, eficácia, satisfação, transparência, clareza, confiança, persuasão e eficiência [Tintarev e Masthoff 2012]. Em seu trabalho sobre o requisito de explicabilidade, Tintarev e Masthoff (2012) constata que tais objetivos podem ser colaborativos mas também podem ser opostos uns aos outros, o que implica em uma necessidade de definir qual dos objetivos será investigado, e quais métricas serão utilizadas. Tal análise de objetivos e métricas também será avaliada no contexto deste trabalho.

O objetivo da pesquisa realizada neste documento é **avaliar uma estratégia de visualização dos resultados de algoritmos de agrupamento no contexto de sistemas de informação**. O que se espera é que com essa pesquisa possam ser obtidos avanços no estudo sobre o requisito explicabilidade em termos de como esse requisito deve ser definido e de como ele pode ser verificado. Observa-se, portanto, que os resultados dos algoritmos de agrupamento possam ser compreendidos de forma mais fácil pelos usuários que forem expostos a eles.

Este trabalho utilizou a plataforma *Outsystems* para desenvolver uma solução interativa que aplica o conceito de visualização de explicação por meio de algoritmos de classificação. Foi desenvolvida uma tela na qual o usuário pudesse informar alguns dados e com esses dados o sistema pudesse gerar um perfil ao qual o usuário pertencesse e mostrar de forma visual o porque dele pertencer àquele perfil. Após o desenvolvimento da solução foi aplicado o método de questionário Escala de Usabilidade de Sistema (SUS, do inglês *System Usability Scale*) para que pudesse ser avaliado o grau de usabilidade do sistema [Brooke, 2013]. Este tipo de questionário foi escolhido pela sua praticidade e rapidez, e também, pelo equilíbrio entre suas métricas. O público ao qual o questionário foi aplicado é composto basicamente por programadores estudantes da área de sistemas de informação que possuem contas na plataforma GitHub. A plataforma Google Forms foi utilizada para criar o questionário e por meio da mesma os resultados foram analisados.

Com as respostas do questionário SUS podemos observar que o grau de usabilidade da solução desenvolvida é de 85, um número considerado excelente [Boucinha e Tarouco, 2013] o que nos faz constatar que a aplicação do requisito de explicabilidade através de árvores de decisão para visualização de resultados de agrupamento se mostra eficiente.

Neste trabalho puderam ser realizadas duas principais contribuições: Primeiro desenvolver uma solução que aplicasse o conceito de visualização por meio de algoritmos de classificação, podendo assim explicar ao usuário os resultados de

algoritmos de agrupamento fazendo com que o usuário consiga compreender melhor os resultados apresentados pelo sistema. Segundo foi levantar a opinião dos usuários sobre a aplicação do requisito de explicabilidade por meio de classificação.

Nas próximas falaremos sobre o referencial teórico onde serão apresentados os principais conceitos trabalhados no estudo, logo após falaremos sobre os trabalhos relacionados que motivaram e auxiliaram na construção deste, em seguida vem a metodologia onde falaremos sobre os materiais e métodos utilizados na construção do trabalho, em seguida tem a proposta de solução desenvolvida, depois os resultados encontrados com a avaliação da solução apresentada e por fim as conclusões que tiramos com este desenvolvimento.

## **2. Referencial Teórico**

Nesta seção são apresentados os principais conceitos e algoritmos tratados no estudo. Primeiro, discute-se o algoritmo de agrupamento k-means e o algoritmo de classificação de Naive Bayes. Após isso, apresenta-se uma análise do conceito de explicabilidade e de estratégias de implementação de explicabilidade.

### **2.1 Classificação e Agrupamento**

A comunidade de inteligência artificial historicamente tem uma longa relação de necessidade com raciocínio e explicações automáticas [Hendricks, Akata, Rohrbach et al 2016]. Para suprir a necessidade têm sido utilizados algoritmos de aprendizagem. Tais algoritmos podem ser classificados de duas maneiras, algoritmos de aprendizagem supervisionada e algoritmos de aprendizagem não supervisionada.

Quando temos um conjunto de treinamento  $X$ , no qual o valor da função de  $X$  é conhecida, temos uma aprendizagem supervisionada [Quevedo 1999]. Ou seja quando conhecemos o resultado da entrada dos dados, temos um algoritmo de aprendizado supervisionado. A tarefa realizada por algoritmos supervisionados geralmente é particionar o grupo de dados recebidos, atribuindo cada dado a classes específicas de acordo com as relações existentes entre eles [Quevedo 1999]. Assim, tais algoritmos utilizam das características previamente conhecidas dos dados para classificá-los de acordo com sua semelhança. Um exemplo de classificação é identificar através das características em uma foto e definir se há um cachorro ou um gato naquela imagem a partir de um conjunto de dados de treinamento já em que aparecem dados assim classificados.

O algoritmo de Naive Bayes é um exemplo de um algoritmo de aprendizagem supervisionada muito utilizado em diversas aplicações como uma solução para problemas de classificação e previsão [Vural, Gok 2017]. Este algoritmo é simples e oferece um ótimo desempenho na categorização de textos em comparação a outros algoritmos [Forman 2003]. O algoritmo de Naive Bayes é um modelo de classificação probabilístico baseado no teorema de Bayes juntamente com a hipótese da independência de recursos [Vural, Gok 2017].

Outro algoritmo de aprendizagem supervisionada é o ID3[Quinlan 1979] este é um algoritmo baseado em árvores de decisão. O ID3 consiste em um algoritmo guloso que constrói a árvore de decisão de cima para baixo [Castro e FERRARI 2016]. A árvore se inicia com apenas um nó que representa o atributo com maior ganho de

informação. A partir deste nó podem ocorrer dois casos, o primeiro é quando todos os objetos deste nó pertencem à mesma classe, logo este nó se torna um nó folha, e no segundo caso o algoritmo calcula o próximo atributo que tem o maior ganho de informação. Assim o algoritmo segue recursivamente até encontrar todos os nós folha.

Quando não se conhecem características sobre o conjunto de dados recebidos, é preciso encontrar um novo meio de agrupá-los, para estas ocasiões são utilizados algoritmos de aprendizagem não supervisionada [Quevedo 1999]. Tais algoritmos buscam agrupar ou organizar os dados de alguma forma fazendo com que eles pareçam mais compreensíveis de acordo com algum atributo calculável, por exemplo a distância que os dados possuem de *clusters* que serão definidos pelo algoritmo.

Quando falamos em algoritmos de aprendizado não supervisionado, o algoritmo mais utilizado é o k-means [Pena, Lozano e Larranaga 1999]. Esse algoritmo de agrupamento consiste em agrupar dados de acordo com um atributo chamado de distância. Para inicializar o algoritmo k-means é necessária a definição dos *clusters* (k). Após definidos os *clusters* é calculada a distância de cada dado do conjunto analisado para cada um dos *clusters*. O dado é atribuído ao *cluster* mais próximo, ou seja o que o atributo calculado distância, foi menor. Após os dados serem agrupados de acordo com seus centróides uma média (*means*) é feita a partir dos pontos de dados relacionados a cada *cluster* e o resultado será o novo centróide, esta mudança de centróide é chamada de movimentação. A segunda e a terceira etapa são iterativas, ou seja, serão repetidas até que não haja nenhuma alteração em nenhum dos centróides. Quando isto acontecer pode dizer que o algoritmo k-means chegou ao seu resultado.

Outro exemplo de aprendizado não supervisionado é o agrupamento hierárquico. Tal agrupamento pode ser dividido em duas abordagens a aglomerativa e a divisiva. A fase inicial da abordagem aglomerativa consiste em *n clusters* com um único objeto cada, tais *clusters* são agrupados sucessivamente. A abordagem divisiva trabalha ao contrário, inicialmente se tem um único *cluster* com todos os objetos que é dividido sucessivamente para obter os grupos de objetos [Faceli et al 2011].

Podemos citar também o algoritmo de agrupamento DBSCAN (do inglês, *Density-Based Spatial Clustering of Applications with Noise*). Tal algoritmo procura agrupar dados analisando um termo definido como vizinhança [Ester et al 1996]. A partir desta análise cada objeto é denominado ponto central ou ponto de borda, caso o objeto seja um ponto central, ele é definido como um *cluster*, com tal *cluster* definido e a partir da análise da vizinhança, o algoritmo agrupa ao *cluster* os objetos pertencentes à sua vizinhança.

## **2.2 Explicabilidade**

O requisito de explicabilidade tem sido cada vez mais importante mediante os avanços na inteligência artificial que faz com que sejam cada vez mais frequentes algoritmos que tomam decisões para os seres humanos [Gilpin et al. 2018]. O requisito de explicabilidade é um requisito não funcional de sistemas, ou seja, ele não descreve diretamente uma funcionalidade do sistema mas sim um atributo de qualidade do mesmo. Explicações podem servir para diversos objetivos tais como, eficácia, satisfação, transparência, confiança, persuasão e eficiência [Tintarev, Masthoff 2012]. A eficácia, se resume na habilidade que o sistema possui para fazer com que os usuários

tomem boas decisões. A característica de satisfação, diz respeito a facilidade de acesso e ao conforto que o usuário obtém ao utilizar o sistema. A transparência refere-se a capacidade do sistema se auto-explicar para que o usuário possa compreender o seu funcionamento. O atributo confiança é basicamente fazer com que o usuário confie nas decisões tomadas pelo sistema. A persuasão é a disposição do sistema para convencer o usuário a experimentar determinada ação. A eficiência é a aptidão do sistema para fazer com que usuários tomem suas decisões de forma rápida. Mas o que define uma explicação eficaz? Uma explicação eficaz é aquela que consegue auxiliar um usuário a tomar uma decisão de acordo com suas próprias preferências [Tintarev, Masthoff 2012]. Como exemplo temos um sistema que recomenda filmes para os seus usuários, com base nas ações do usuário o algoritmo do sistema provê recomendações de filmes, a explicabilidade nesse contexto entraria para fornecer explicações dos resultados do algoritmo de recomendação para que o usuário possa compreender o por que daquele resultado.

### **3. Trabalhos Relacionados**

Nesta seção são apresentados artigos relacionados que tratam de estratégias de explicabilidade, de avaliação de explicabilidade e da associação de explicabilidade com algoritmos de classificação e agrupamento.

Gilpin et al. (2018) constroem uma taxonomia para apresentar conceitos e estudos relacionados a interpretabilidade e explicabilidade aplicadas na inteligência artificial. Boas explicações dependem da pergunta, “*why questions*”, que são basicamente as questões de porquê, foram as utilizadas nesse artigo. Explicações podem ser avaliadas em duas maneiras, de acordo com sua interpretação e a sua completude. Processos são muitas vezes complexos, a arquitetura ResNet cria um novo modelo do processo tornando ele mais fácil de ser explicado, ou destacar uma parte crucial para que possa ser explicada separadamente. As diversas literaturas examinadas caem em 3 categorias (justificativas, representativas e produzir explicações). Foi concluído que existe necessidade de transparência nas explicações. Este artigo trata da aplicação da explicabilidade e a necessidade deste requisito o que inspirou este trabalho, atendo-se ao caso de algoritmos de agrupamento.

Nunes e Jannach (2017) realizam uma revisão sistemática sobre o assunto com base em dois termos principais, “*explanations*” e “*decision support systems*”. Algoritmos de recomendação são na maioria das vezes complexos o que se torna uma barreira ao tentar explicar o funcionamento dos mesmos para prover explicabilidade. Confiança do usuário nas decisões recomendações e decisões tomadas por sistemas de informação são chaves para o sucesso do mesmo. A confiança pode ser induzida através de diversos fatores, um dos mais utilizados é a transparência. Foram analisados diversos códigos que fornecem explicabilidade. Foi constatado que a inferência é uma das formas mais utilizadas de se prover explicabilidade. Falando sobre como a explicabilidade é apresentada ao usuário, na maioria das vezes ela é apresentada na linguagem natural. Saber como a explicabilidade pode ser utilizada de melhor forma, esse artigo contribuiu com o presente trabalho facilitando a aplicação do requisito na tentativa de prover explicações transparentes e que transmitam confiança ao usuário. Os critérios desenvolvidos são considerados no presente estudo.

Ponciano e Brasileiro (2014) apresentaram uma pesquisa a fim de traçar perfis de engajamento de voluntários em projetos de ciência cidadã usando algoritmos de agrupamento. Para tal pesquisa foram utilizadas as bases de dados dos projetos “*Galaxy Zoo*” e “*Milk Way Project*”. Foram utilizadas duas métricas para calcular o engajamento dos voluntários, relação de atividade e tempo diário dedicado. Mais duas métricas foram usadas para definir o tempo de engajamento duração de atividade relativa e variação na periodicidade. Após definir as métricas foi aplicado o algoritmo de agrupamento hierárquico e a partir do resultado foi aplicado o algoritmo k-means. Foram definidos 5 perfis a partir da análise dos resultados do agrupamento (*hardworking, spasmodic, persistent, lasting, and moderate*). A maioria dos voluntários caem no perfil “moderate engagement”. Os voluntários definidos como “*persistent*” segundo a pesquisa são os mais esforçados, e procuram fazer a maior parte das tarefas. A idéia de explicar o agrupamento em perfis realizada por algoritmos motiva o requisito de explicabilidade trazendo explicações transparentes para resultados de algoritmos complexos.

Tintarev e Masthoff (2012) apresentam uma pesquisa que consiste em submeter voluntários a testes a fim de reconhecer a satisfação dos usuários e a eficácia da explicabilidade em três formas, sendo elas, “*Baseline.*”, “*Non-personalized, feature-based*” e “*Personalized, feature-based*”. Foi usada a métrica de Bilgic and Mooney’s (2005) para avaliar os resultados. Os testes consistem em que os usuários julguem itens antes e depois de ser aplicado o conceito de explicabilidade. Os itens a serem julgados consistem em câmeras e filmes. A partir do resultado constatou-se que explicações personalizadas são mais satisfatórias que as não personalizadas, porém não houve grande diferença entre explicações personalizadas e *baseline*. Explicabilidade personalizada aumenta a eficácia. Esse artigo nos traz maneiras diferentes de se aplicar o requisito de explicabilidade como por exemplo explicações personalizadas e não personalizadas, o que nos ajuda a propor e avaliar estratégias para satisfazer tal requisito.

## **4. Metodologia**

O estudo realizado neste trabalho possui caráter experimental e qualitativa. A partir de grupos gerados por algoritmos de agrupamento, foi construída uma solução que fornece visualizações baseadas em árvores de decisão que facilitem a compreensão do resultado dos algoritmos. A estratégia de visualização desenvolvida é baseada em algoritmos de classificação. Nesta seção são apresentadas as etapas de obtenção de dados, agrupamento e classificação no R-Studio, a implementação da solução e a avaliação.

### **4.1. Obtenção de dados e materiais**

Os dados utilizados para o desenvolvimento deste trabalho foram previamente coletados em outro trabalho [Costa e Ponciano 2018]. Uma API foi utilizada para coletar os dados da plataforma GitHub, um total de 325 repositórios com diferentes frequências de utilização da plataforma foram coletados e analisados. Após a coleta os dados foram armazenados em arquivos do excel. Vista a necessidade de normalização foi utilizado o algoritmo min-max para normalizar os dados e colocá-los em uma escala entre 0 e 1 [Jain 1991]. Na obtenção de dados foram considerados 6 parâmetros, (quantidade de

seguidores) que diz respeito a quantidade de seguidores que o repositório possui, (quantidade seguindo) que quer dizer a quantidade de usuários que seguem o repositório em questão, (quantidade de repositórios públicos) que significa a quantidade de repositórios públicos que a conta em questão possui, (quantidade de *pull requests*) que é a quantidade de pull requests realizados pelo programador dono da conta analisada, (quantidade de *issues* incluídos) que é a quantidade de issues reportados pelo programador, (quantidade de *commits* incluídos) que diz respeito a quantidade de commits que foram incluídos pelo programador analisado.

## **4.2. Agrupamento e classificação**

A partir dos dados coletados da plataforma GitHub e normalizados através do algoritmo Min-Max, foi possível realizar o agrupamento dos repositórios para que assim pudessem ser separados em perfis distintos de acordo com suas características analisando a semelhança entre os parâmetros. O algoritmo K-means foi utilizado para realizar o agrupamento visando encontrar os mesmos grupos estudados no trabalho [Costa e Ponciano 2018]. Para realizar o agrupamento dos dados foi utilizada a plataforma R-Studio que permite a programação através da linguagem R. A linguagem R possibilita a implementação do algoritmo de agrupamento K-means e juntamente a plataforma R-Studio permite a manipulação de grande volumes de dados.

Com os dados já agrupados, foi aplicado um algoritmo de classificação baseado em árvores binárias de decisão para que tais grupos fossem classificados com base nos parâmetros que fossem relevantes. A classificação também foi feita utilizando a linguagem R na plataforma R-Studio. Feita a classificação, foi gerada a árvore de decisão para o problema apresentado para que dessa forma os parâmetros analisados pudessem ser estudados.

## **4.3. Implementação da solução**

Com o agrupamento e a classificação aplicados, uma aplicação foi construída de modo que pudesse implementar a árvore de decisão e fornecer visualizações para os resultados gerados pelo algoritmo de agrupamento. A aplicação foi construída através da plataforma Outsystems que é uma plataforma de desenvolvimento ágil que permitiu maior foco na lógica visto que a plataforma cuida da parte visual, que são basicamente campos para a inserção dos dados e um campo para demonstrar o resultado, veja a tela desenvolvida na Figura 1. Para implementar a árvore binária de decisão foram utilizadas estruturas condicionais que reproduziam exatamente os caminhos da árvore gerada pelo algoritmo de classificação até que chegasse em suas folhas. Cada folha fornecia um resultado, que consistia no perfil ao qual o usuário pertence, as variáveis dentro dos parâmetros que foram relevantes para que o algoritmo apresentasse este resultado e não outro, e uma mensagem explicativa que busca tornar explícitos os passos executados pelo algoritmo até chegar a tal resultado.

**Figura 1. Tela da Aplicação Desenvolvida**

#### 4.4. Avaliação

A metodologia de pesquisa SUS desenvolvida por Jhon Brooke em 1986 foi criada para solucionar problemas como o longo tempo que o criador usava para aplicar questionários e reduzir consideravelmente a complexidade dos mesmos. Esta metodologia consiste em aplicar um questionário composto por 10 questões padrões apresentadas na Tabela 1 após o usuário realizar um teste na solução a ser avaliada. O resultado do questionário é o grau de usabilidade que pode variar entre 0 e 100. A versão em português das questões do SUS foi extraída de [Boucinha e Tarouco, 2013].

**Tabela 1. Afirmações do Questionário SUS**

Nº	Afirmações
1	Eu acho que gostaria de usar esse sistema com frequência.
2	Eu acho o sistema desnecessariamente complexo.
3	Eu achei o sistema fácil de usar.
4	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

5	Eu acho que as várias funções do sistema estão muito bem integradas.
6	Eu acho que o sistema apresenta muita inconsistência.
7	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8	Eu achei o sistema atrapalhado de usar.
9	Eu me senti confiante ao usar o sistema.
10	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

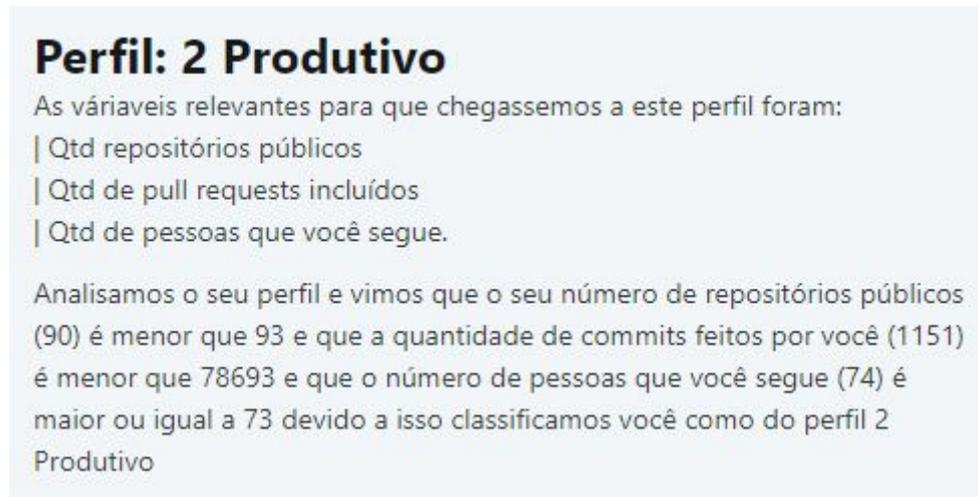
O questionário SUS foi aplicado a voluntários pertencentes ao público alvo da aplicação para que por meio das respostas pudessem ser calculados resultados sobre a usabilidade do sistema. Foi aplicado o questionário no dia 22 de novembro de 2019 na PUC Contagem a um total de 7 participantes. O resultado do questionário SUS é obtido através de um cálculo que consiste em: Para as questões de número ímpares (1,3,5,7,9) deve-se subtrair 1 da pontuação que o usuário respondeu. Para as questões pares (2,4,6,8) deve-se subtrair a resposta do usuário de 5. Ao final deve-se somar os resultados encontrados e multiplicar este valor por 2,5. O resultado será um valor entre 0 e 100 sendo que de 0 a 25 o resultado é considerado pior imaginável, de 26 a 38, pobre, de 39 a 73, considerado bom, 74 a 85, considerado excelente, 86 a 100, considerado melhor imaginável. Uma nota menor que 50 no questionário SUS pode levar ao insucesso do sistema.

## **5. Proposta de Solução: Explicando agrupamento por meio de classificação**

O principal objetivo deste trabalho foi aplicar o conceito de explicabilidade por meio de visualização buscando explicar os resultados de algoritmos de agrupamento que muitas vezes são complexos através de classificação. Para isto o método de classificação por árvore binária de decisão foi utilizado, com ele foi possível gerar uma estrutura de árvore de decisões que pelos parâmetros passados traça um caminho até o resultado do algoritmo de agrupamento. O entendimento de árvores de decisão por si só pode não ser trivial a todos os usuários de aplicações que utilizam de algoritmos de agrupamento para fornecer algum tipo de resultado.

A partir desta premissa foi desenvolvida uma aplicação que busca abstrair a representação da árvore de decisão e fornece ao usuário por meio de texto uma explicação do que foi executado no algoritmo. A aplicação considera dois fatores, o primeiro diz respeito às variáveis que foram relevantes na execução do algoritmo, essas variáveis são as que influenciaram de em algum dos caminhos que o algoritmo tomou até chegar ao resultado, a partir disto o sistema apresenta ao usuário estas variáveis. O segundo fator considerado pela solução é, o valor que seria a linha tênue que separa o usuário de um caminho ou outro na árvore de decisão, este dado é apresentado ao usuário por meio de um texto que busca responder perguntas do tipo “quanto desta variável eu deveria ter para pertencer a outro perfil?”. As variáveis relevantes juntamente ao texto com os valores que faz com que o algoritmo tome um caminho e

não outro são concatenados e apresentados ao usuário juntamente ao resultado do algoritmo de agrupamento, veja na Figura 2.



**Perfil: 2 Produtivo**

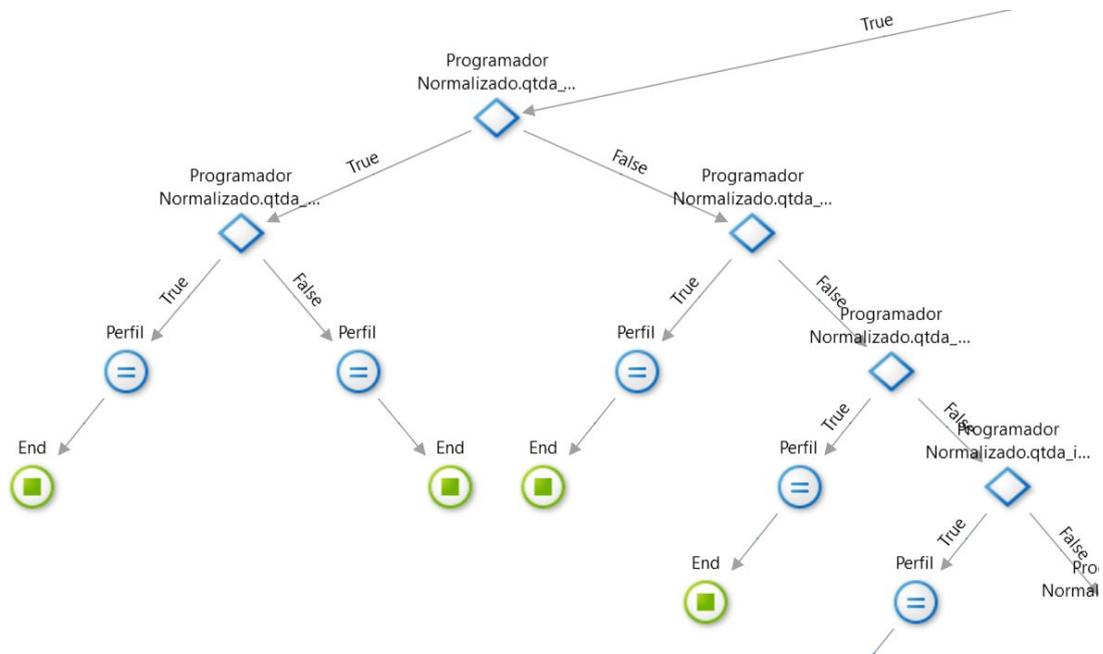
As variáveis relevantes para que chegássemos a este perfil foram:

- | Qtd repositórios públicos
- | Qtd de pull requests incluídos
- | Qtd de pessoas que você segue.

Analisamos o seu perfil e vimos que o seu número de repositórios públicos (90) é menor que 93 e que a quantidade de commits feitos por você (1151) é menor que 78693 e que o número de pessoas que você segue (74) é maior ou igual a 73 devido a isso classificamos você como do perfil 2 Produtivo

**Figura 2. Exemplo de Resultado Apresentado Pela Aplicação Desenvolvida**

Para chegar a tal resultado, analisamos a árvore gerada pelo algoritmo de classificação executado na plataforma R-Studio afim reproduzir todos os caminhos e possibilidades de resultados. Para isso analisamos todos os valores que faziam com que o algoritmo tomasse um caminho e não o outro e passa-los para a lógica da aplicação. Com os valores descobertos foi construída uma lógica na plataforma Outsystems (Figura 3) que reproduzia a estrutura da árvore de decisão que foi gerada pelo algoritmo de classificação, a lógica consiste em estruturas condicionais que que comparam o valor digitado pelo usuário com o valor que o divide entre seguir por um ou por outro caminho (Figura 4).



**Figura 3. Visão Geral de Parte da Árvore Implementada na Plataforma Outsystems**

ProgramadorNormalizado.qtda\_rep\_publicos >= 0.105598? Condition

ProgramadorNormalizado.qtda\_rep\_publicos >= 0.105598

↑ Valor digitado

↑ Valor encontrado encontrado pelo algoritmo de classificação

+ - \* / and or not True False = <> < > <= >= () [] null

- Id
- id\_programador
- qtda\_seguidores
- qtda\_seguinto
- qtda\_rep\_publicos

Description

✓ The expression is ok (Type: Boolean)

DONE

**Figura 4. Lógica da Estrutura Condicional**

## 6. Resultados da Percepção de Usuários acerca da Solução

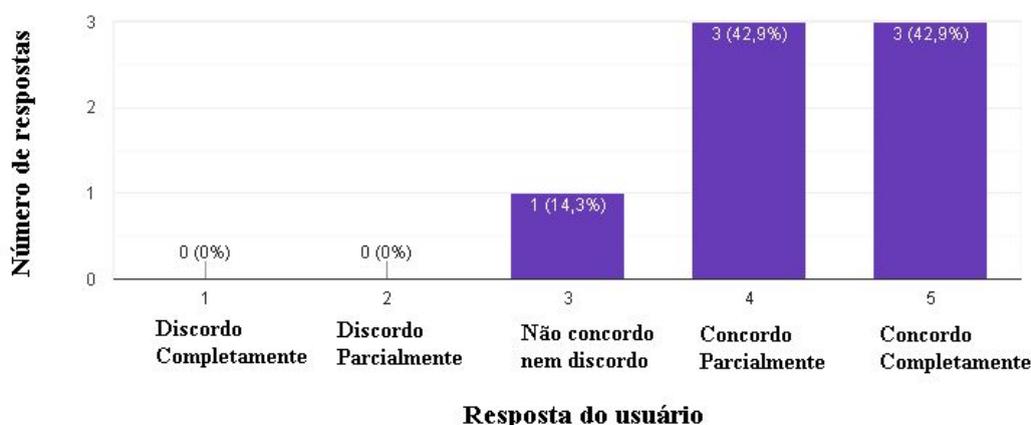
Nesta seção apresentaremos os resultados obtidos a partir da aplicação do conceito de explicabilidade por meio de árvore binária de decisão visando explicar os resultados de algoritmos de agrupamento. Serão apresentados os dados obtidos através da aplicação do questionário aos voluntários pertencentes ao público alvo e por meio destes dados mostrar resultados relacionados a relevância da solução apresentada pela aplicação.

### 6.1. Avaliação do questionário SUS

Foi feita uma avaliação da forma com que a metodologia SUS propõe para todos os voluntários de forma individual, e a partir disto foi calculada uma média simples para definir o grau de usabilidade da solução proposta. O grau de usabilidade varia entre 1 e 100 e o encontrado pelo questionário SUS em relação à solução proposta, foi de 85, o que é considerado um resultado excelente.

### 6.2. Análise de confiança

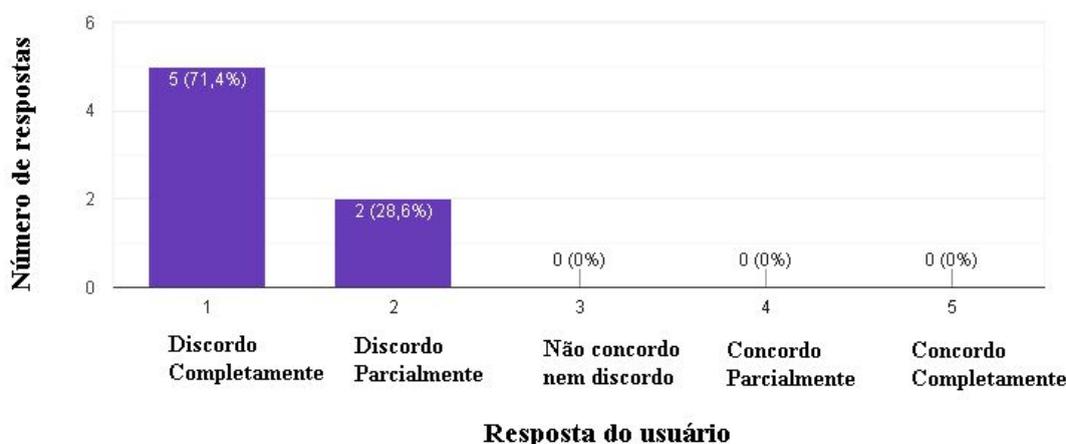
A premissa que fez com que a solução fosse desenvolvida foi a aplicação do requisito de explicabilidade de sistemas e um dos itens que é crucial na explicabilidade é a confiança que o usuário tem nos resultados apresentados. A afirmação de número 9 do questionário SUS diz o seguinte, “Eu me senti confiante ao usar o sistema.”, fazendo uma análise separada dessa afirmação, podemos ver que a grande maioria dos voluntários concorda completa ou parcialmente com a afirmação (um total de 85,8%), apenas um dos voluntários disse que não concorda nem discorda com a afirmação (14,3%) veja na Figura 5. Essa análise nos ajudou a avaliar se a solução estaria cumprindo seu papel.



**Figura 5. Respostas à Afirmação “Eu me senti confiante ao usar o sistema.” do Questionário SUS**

Relacionada com as respostas à afirmação 9 temos as respostas à afirmação 6 que diz: “Eu acho que o sistema apresenta muita inconsistência.”. A maioria (71,4%) disse que discorda completamente com a afirmação e o restante (28,6%) disse que discorda parcialmente (Figura 6), o resultado desta questão em particular, de certa

forma, também indica uma certa confiança dos usuários no sistema, pois a maioria considera que o sistema não é inconsistente, ou seja, o sistema consegue por meio dos resultados apresentados se, se sustentar.



**Figura 6. Respostas à Afirmação “Eu acho que o sistema apresenta muita inconsistência.” do Questionário SUS**

## 7. Conclusões

Este trabalho aplicou o conceito de visualização por meio de uma árvore de decisão e investigou a relevância deste conceito para os prover explicações a usuários. Para se obter respostas sobre a investigação primeiro foram realizadas pesquisas na área de explicabilidade, agrupamento e classificação para se obter embasamento teórico sobre o assunto tratado. A partir dos dados coletados dos repositórios da plataforma GitHub, aplicou-se um algoritmo de agrupamento para identificar os diferentes perfis dentre os repositórios. Foi executado um algoritmo de classificação para gerar uma árvore de decisão referente ao resultado do algoritmo de agrupamento para que a partir dela, pudesse ser construída a solução que geraria visualizações para o resultado do algoritmo de agrupamento.

Os resultados obtidos mostram que soluções que possam exibir o visualizações de algoritmos facilitam a compreensão dos usuários sobre agrupamento. A implementação da árvore de decisão, mesmo que de forma fixa, traz clareza aos resultados de algoritmos de agrupamento que são muitas vezes complexos. Em contrapartida, a mudança do contexto traz dificuldades a esta forma de implementação, pois uma nova árvore de decisão teria que ser gerada automaticamente para cada contexto diferente em que a solução for aplicada.

Como trabalhos futuros, partindo da implementação de árvores de decisão para visualizar os resultados de algoritmos de agrupamento, sugere-se buscar desenvolver soluções que se adaptem a mudança de parâmetros e de contextos, tornando assim a solução polimórfica e aplicável a diversos contextos sem necessidade de alterações bruscas ou re-desenvolvimento.

## Referências Bibliográficas

- BATISTA, E. D. O. (2017). *Sistemas de informação*. Editora Saraiva.
- BILGIC, M.; MOONEY, R. J. (2005). Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI* (Vol. 5, p. 153).
- BOUCINHA, Rafael Marimon; TAROUÇO, Liane Margarida Rockenbach. Avaliação de ambiente virtual de aprendizagem com o uso do sus-system usability scale. *RENOTE*, v. 11, n. 3, 2013.
- BROOKE, John. SUS: a retrospective. *Journal of usability studies*, v. 8, n. 2, p. 29-40, 2013.
- CASTRO, L. N. D.; FERRARI, D. G. (2016). *Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações*. São Paulo: Saraiva.
- ECKART, Thomas; GRADL, Tobias. Working towards a Metadata Federation of CLARIN and DARIAH-DE. *CLARIN Annual Conference*, 2017
- ESTER, M.; KRIEGEL, H. P.; SANDER, J.; XU, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).
- FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. D. L. (2011). *Inteligência Artificial: Uma abordagem de aprendizado de máquina*.
- FORMAN, George. (2003) An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, v. 3, n. Mar, p. 1289-1305.
- GILPIN, Leilani H.; BAU, David; YUAN, Ben Z.; BAJWA, Ayesha; SPECTER, Michel; KAGAL, Lalana (2018). Explaining Explanations: An Overview of Interpretability of Machine Learning. In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, p. 80-89.
- HENDRICKS, L. A.; AKATA, Z.; ROHRBACH, M.; DONAHUE, J.; SCHIELE, B.; DARRELLI, T (2016). Generating visual explanations. In: *European Conference on Computer Vision*. Springer, Cham. p. 3-19.
- JAIN, A. K., FARROKHINA, F. (1991). Unsupervised texture segmentation using Gabor filters. *Pattern recognition*, 24(12), 1167-1186.
- JAIN, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*.
- KIRANAGI, Bapu B; HARISH, B S; MANJUNATH, S; GURU, D S (2014). *New Symbolic Proximity Approximation Techniques and Clustering Algorithms*. Technical Report, LTU, [S. l.], 28 maio.
- NUNES, Ingrid; JANNACH, Dietmar (2017). A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, v. 27, n. 3-5, p. 393-444.

- PENA, José M.; LOZANO, Jose Antonio; LARRANAGA, Pedro (1999). An empirical comparison of four initialization methods for the k-means algorithm. *Pattern recognition letters*, v. 20, n. 10, p. 1027-1040.
- PONCIANO, Lesandro; BRASILEIRO, Francisco (2014). Finding volunteers' engagement profiles in human computation for citizen science projects. *Human Computation* vol. 1, no. 2, pp. 245-264.
- QUEVEDO, Miguel Ángel Cazorla (1999). *Fundamentos de inteligência artificial*. Universidad de Alicante.
- QUINLAN, J. R. (1979). Discovering rules by induction from large collections of examples. *Expert systems in the micro electronics age*.
- COSTA, Victor. PONCIANO, Lesandro (2018). Profiling Patterns of Interaction of Developers with Repositories on the GitHub Platform. Trabalho de Conclusão de Curso em Sistemas de Informação. PUC Minas. Unidade Contagem.
- TINTAREV, Nava; MASTHOFF, Judith (2012). Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, v. 22, n. 4-5, p. 399-439.
- VURAL, Mehmet Sait; GÖK, Mustafa (2017). Criminal prediction using Naive Bayes theory. *Neural Computing and Applications*, v. 28, n. 9, p. 2581-2592.
- WANG, Dianhui (2013). B-Miscore: A new similarity metric for self-organization of DNA k-mers. La Trobe Univ., Melbourne, VIC, Australia, Tech. Rep. LTU-22-06-2013.
- ZAKI, Mohammed J.; MEIRA, Wagner Meira (2014). *DATA MINING AND ANALYSIS: Fundamental Concepts and Algorithms*. Cambridge: Cambridge University Press. ISBN 978-0-521-76633-3.