UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO (DCC)

PEDRO DE OLIVEIRA GUEDES

REVISÃO DE LITERATURA EM EXPLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL: UMA VISÃO VOLTADA PARA MÉTODOS CONTRAFACTUAIS

BELO HORIZONTE
2025

PEDRO DE OLIVEIRA GUEDES

REVISÃO DE LITERATURA EM EXPLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL: UMA VISÃO VOLTADA PARA MÉTODOS CONTRAFACTUAIS

Trabalho de Conclusão de Curso, apresentado ao curso de Sistemas de Informação da Universidade Federal de Minas Gerais (UFMG), como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

ORIENTADOR: RENATO VIMIEIRO

BELO HORIZONTE

2025

AGRADECIMENTOS

A minha irmã Clara, minha mãe Keyla e meu pai Cláudio pelo amor e carinho incondicional ao longo de toda a minha vida, pelo apoio às minhas decisões e por terem acreditado no meu potencial desde criança. Espero algum dia poder retribuir tudo o que vocês fizeram e fazem por mim.

A minha namorada, Marcela Cunha de Jesus, que alegra meus dias desde 2019. Ter você ao meu lado, me apoiando, acreditando em mim e deixando mais leve todo o processo exaustivo de se concluir uma graduação fez toda a diferença.

Aos meus amigos do cursinho e ensino médio que, mesmo escolhendo cursos diferentes do meu, se mantiveram presentes na minha vida. Agradeço também às pessoas especiais que eles me apresentaram ao longo desses anos, é muito gratificante ver nosso grupo crescer ao invés de diminuir.

Aos meus amigos da graduação, com quem pude trocar conhecimento, experiências e risadas nessa trajetória acadêmica, dentro e fora de sala de aula. Em especial ao João Vitor Santana Depollo, sua parceria e ajuda em momentos difíceis foram fundamentais para essa conquista.

Ao meu orientador, Renato Vimieiro, pela dedicação e direcionamentos que foram parte crucial do planejamento, execução e escrita deste trabalho.

Pedro de Oliveira Guedes

RESUMO

Existe uma demanda crescente por explicações de modelos de aprendizado de máquina, movimentada principalmente pelos avanços da *General Data Protection Regulation* (*GDPR*) na Europa e a Lei Geral de Proteção de Dados Pessoais (LGPD) no Brasil. Em paralelo a essa necessidade de explicação, existe também uma complexidade crescente nos modelos empregados na indústria para realizar decisões automáticas, que muitas vezes são caixas pretas não interpretáveis facilmente. Diante deste desafio, o presente trabalho propõe uma revisão de literatura de métodos estado da arte para explicação de decisões de modelos de inteligência artificial, voltada principalmente para métodos que forneçam sugestões de alteração que mudem a decisão tomada, ou em outras palavras: métodos contrafactuais. Para isso, foi feita a coleta de artigos em plataformas digitais, enriquecimento dos dados coletados, filtragem inicial para considerar apenas publicações de interesse, análise bibliométrica para encontrar tendências da área, filtragem final para considerar apenas proposições de novas ferramentas e análise qualitativa das ferramentas propostas.

Palavras-chave: Revisão de literatura; Análise bibliométrica; Análise qualitativa; Explicação de IA; Explicações contrafactuais

ABSTRACT

There is a growing demand for explanations of machine learning models, driven mainly by advances in the General Data Protection Regulation (GDPR) in Europe and the *Lei Geral de Proteção de Dados Pessoais* (LGPD) in Brazil. In parallel with this need for explanation, there is also a growing complexity in the models used in the industry to make automatic decisions, which are often black boxes that are not easily interpretable. Faced with this challenge, this work proposes a literature review of state-of-the-art methods for explaining decisions of artificial intelligence models, focusing mainly on methods that provide modifications that change the decision made, or in other words: counterfactual methods. To this end, articles were collected from digital platforms and programmatically enriched, initial filtering was performed to consider only publications of interest, bibliometric analysis was performed to find trends in the area, final filtering was performed to consider only propositions for new tools and, finally, qualitative analysis of the proposed tools.

Keywords: Literature review; Bibliometric analysis; Qualitative analysis; AI Explanation; Counterfactual Explanations

Sumário

Introdução

De acordo com Peter J. Denning no relatório "*Computing as a Discipline*" de 1989, a ciência da computação tem por objetivo responder e atuar em cima da seguinte questão: "O que pode ser (eficientemente) automatizado?". Embora seja uma área relativamente recente, principalmente em comparação com as adjacências como a matemática, os avanços computacionais ocorridos desde a década de 1950 mudaram estruturalmente a sociedade. Atualmente é raro encontrar organizações públicas ou privadas que não utilizem artifícios da computação para automatizar tarefas, que vão desde o preenchimento e envio de formulários até predições de risco e criação de perfis para os usuários.

Uma das preocupações criadas pela ampla utilização de algoritmos por parte de organizações, em especial as privadas, está na privacidade dos dados dos usuários que, ao serem coletados e manipulados em larga escala, podem sofrer vazamentos e serem explorados de forma indevida. Para lidar com problemas assim, surgiu a *General Data Protection Regulation* (GDPR) em 2016 na Europa, que exige transparência e responsabilidade na coleta e manipulação de dados dos usuários. Com o mesmo propósito, surgiu também a Lei Geral de Proteção de Dados Pessoais (LGPD) em 2020 no Brasil.

Ainda que o principal propósito dessas regulamentações seja garantir a privacidade dos usuários, por exemplo impedindo o compartilhamento de dados com terceiros, a menos que explicitamente consentido pelo usuário, os objetivos delas vão além. O artigo 21 da GDPR dá, aos usuários cujos dados foram coletados, o direito de contestar decisões automáticas, o que, quando combinado com o artigo 22 do mesmo documento, garante o direito de intervenção humana para explicação e argumentação no caso de decisões completamente automatizadas. No mesmo caminho, o artigo 20 da LGPD dá ao usuário o direito de solicitar revisões de decisões unicamente automatizadas, tomadas com base nos dados coletados.

Para que as empresas controladoras dos dados estejam preparadas para lidar com as solicitações dos usuários, é necessário que a lógica de tomada de decisão automática seja explícita, ou ao menos passível de interpretação. Em casos programáticos, em que os dados são tratados por uma série de estruturas condicionais especificamente programadas por desenvolvedores seguindo

regras de negócio bem definidas, fornecer informações sobre o processo é relativamente simples. Porém, em aplicações mais complexas, são raros os casos em que o fluxo de tomada de decisão se mantém separado da utilização de algoritmos de aprendizado de máquina, que por vezes são contra intuitivos e de difícil entendimento por humanos. Nesse cenário, surge a necessidade de técnicas para explicação dos modelos gerados por esses algoritmos.

Pensando na atualidade e relevância do tema, o objetivo do trabalho é realizar uma revisão de literatura das publicações recentes sobre algoritmos e ferramentas de explicação contrafactual de decisões tomadas por inteligências artificiais (IAs), com foco em modelos de regressão e classificação. As demais seções deste relatório de progresso estão organizadas da seguinte forma: Referencial Teórico; Metodologia; Resultados e, por fim, Conclusão.

Referencial Teórico

De acordo com Zhi-Hua Zhou no livro "*Machine Learning*" de 2021, aprendizado de máquina é uma técnica que melhora a performance de sistemas através do aprendizado por experiência através de métodos computacionais. Para alcançar esse objetivo, existem vários métodos computacionais, como a regressão linear que é um dos mais simples na área, consistindo em uma abordagem matemática para realizar análises preditivas (MAULUD; ABDULAZEEZ, 2020).

Para ferramentas como a regressão linear, a obtenção de explicação tende a ser menos complicada já que é possível determinar a importância de uma das características (*features*) da amostra pelo peso atribuído a ela durante o treino do modelo. Nesse caso, se o peso é de grande magnitude, variações na *feature* entre amostras geram grande interferência no resultado final. No caso de modelos baseados em árvores de decisão, que são classificadores expressos em uma partição recursiva do espaço de instâncias (ROKACH; MAIMON, 2005), podem ser usadas técnicas como a de Plots de Dependência Parcial (PDP), que mostra o efeito de uma das *features* da amostra na predição final do modelo (MOLNAR, 2019).

Embora a explicação fornecida pela técnica e PDP seja bastante intuitiva, ela acaba sendo falha no caso de a *feature* analisada ser codependente com uma ou mais das outras, já que pode gerar amostras de dados espúrias e que não refletem a real distribuição dos dados. Para corrigir esse problema, surgiram outros métodos como Efeitos Locais Acumulados (ALE), que utiliza

densidade condicional ao invés de densidade marginal para estimar os valores das *features* adjacentes, e Expectativa Condicional Individual (ICE) que junta as predições de todas as instâncias com os dados originais em um único gráfico para evitar amostras espúrias.

Os métodos citados anteriormente foram projetados para explicar as predições de forma global, ou seja, as explicações obtidas teoricamente se aplicam para todas as instâncias de dados disponíveis. Isso pode ser difícil em casos reais, principalmente se tratando de modelos não interpretáveis, como aqueles gerados por técnicas de ensemble, onde a importância das *features* podem assumir comportamentos inesperados para amostras diferentes.

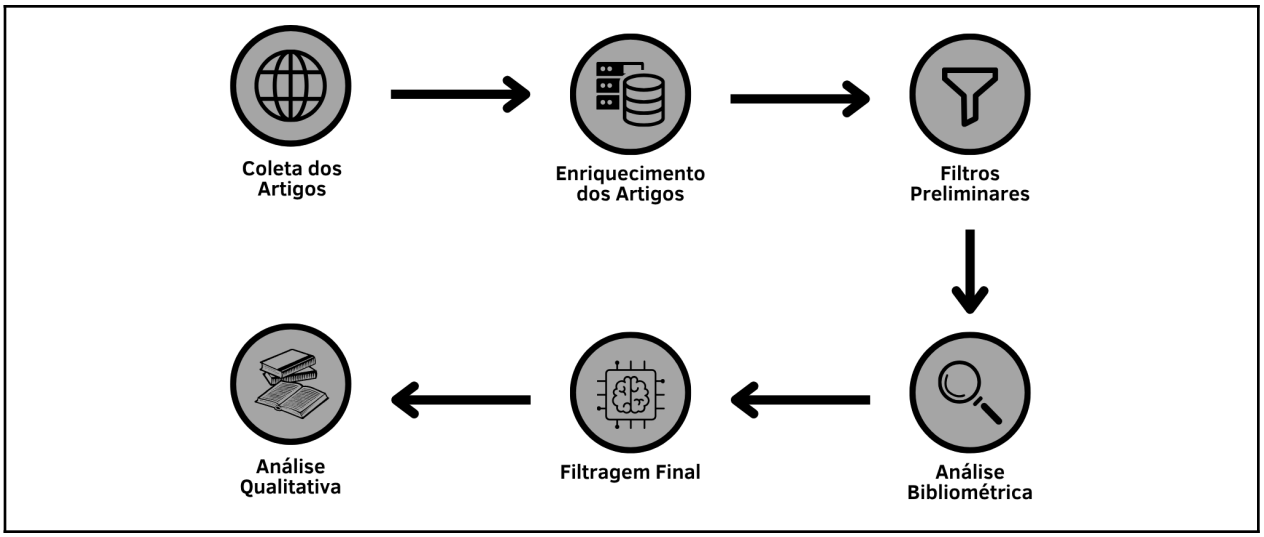
Por vezes, pode ser necessário explicar o comportamento do modelo apenas para uma única instância, como é o caso das revisões de decisão automática previstas como direito pela LGPD e GDPR. Para atuar nesse cenário surge a técnica de modelos substitutos, mais especificamente a técnica de Explicações Locais Interpretáveis Agnósticas ao Modelo (LIME).

LIME é um algoritmo capaz de explicar fidedignamente as predições de qualquer modelo, ao realizar aproximações dele localmente com modelos substitutos interpretáveis (RIBEIRO; SINGH; GUESTRIN, 2016). Para isso, a partir da instância local a ser interpretada, são realizadas pequenas perturbações nos dados, de forma a criar artificialmente outras amostras similares à original e obter predições para elas. O dataset artificial é então utilizado para treinar um modelo interpretável, que será usado para fornecer explicações sobre o efeito das *features* na tomada de decisão.

Outra técnica local é a de Explicações Aditivas por valores Shapley (SHAP) que, assim como o LIME, também realiza perturbações nos dados para verificação da importância das *features* para a amostra. A análise de importância é feita de forma diferente, o LIME utiliza métodos como distribuição probabilística para verificar a presença das *features* nos resultados do modelo treinado localmente, já o SHAP utiliza valores Shapley que é um método bastante utilizado em teoria de jogos para distribuir recompensas entre jogadores. Além disso, enquanto o LIME prioriza amostras similares à original para dar explicações, o SHAP privilegia amostras que tenham diferenças focadas em poucas *features*. Por exemplo, se uma amostra possui uma única *feature* do conjunto ativada, a instância é útil para verificar o impacto individual da *feature*, o mesmo vale para a ausência de uma única *feature*.

Metodologia

Em termos mais específicos, o trabalho consiste em uma revisão de literatura sobre métodos de explicação contrafactual para decisões de modelos de aprendizado de máquina treinados para tarefas de regressão e classificação. Para isso, será feita a apreciação de materiais científicos de leitura sobre o tema, publicados entre janeiro de 2015 e junho de 2024, com foco em publicações que abordam novos algoritmos ou ferramentas para a explicação contrafactual de modelos de IA. As etapas necessárias para condução do trabalho são ilustradas, de forma condensada, na imagem abaixo.



As próximas subseções possuem títulos idênticos aos das etapas da imagem, sendo utilizadas para descrevê-las e apresentar resultados, quando cabível.

Coleta dos Artigos

Para coleta das publicações, foram escolhidas as plataformas "*ACM Digital Library*", "*IEEE Xplore*" e "Portal de Periódicos da CAPES". Buscando incluir artigos de autoria e conteúdo brasileiro, as palavras chave no critério de busca incluíam também traduções de português para inglês nos termos, realizando disjunções nos termos. Como as plataformas são diferentes entre si, as linguagens para escrita dos critérios de busca também são. Sendo assim, foi produzida a tabela abaixo, em que cada linha é uma condicional que deve ser respeitada em conjunção com as demais, como demonstração do conteúdo das consultas, que é comum entre as plataformas.

Elemento do artigo	Operação	Termos
Qualquer texto do artigo	Deve conter um dos termos	[“algorithm”, “algoritmo”]
Qualquer texto do artigo	Deve conter um dos termos	[“explain”, “explica”]
Qualquer texto do artigo	Deve conter um dos termos	[“ai”, “ia”]
Qualquer texto do artigo	**NÃO** pode conter um dos termos	[“recommendation system”, “sistema de recomendação”, “recommender”, “privacy”, “privacidade”]
Título do artigo	**NÃO** pode conter um dos termos	[“neural”, “network”, “graph”, “reinforcement”, “reforço”, “benchmarking”, “survey”, “comparison”, “comparação”]
Data de publicação	Deve estar entre	01/01/2015 e 30/06/2024

As negações incluídas nas consultas foram realizadas iterativamente, ao ler o resumo dos artigos retornados, identificando palavras chave comuns entre aqueles que, para os objetivos do trabalho, deveriam ser considerados falsos positivos. Para garantir que os resultados contemplam apenas artigos, foi incluído o filtro “Research Article” no tipo de publicação da plataforma *ACM*, reduzindo a quantidade total de publicações de 216 para 185. Como as demais plataformas retornaram menos resultados, não foi feito nenhum processo de filtragem adicional. A coleta foi feita no mês de setembro de 2024, por isso, o filtro de data da publicação foi ajustado para iniciar em janeiro de 2015, como já era planejado, mas precisou finalizar na metade do ano de 2024. Essa decisão de filtro de data pode impactar as análises bibliométricas de tendência de publicação, uma vez que o ano de 2024 não está completo e pode dar a falsa impressão de queda de popularidade na área.

Com os filtros realizados, foram recuperados 35 publicações na “CAPES”, 14 na “*IEEE Xplore*” e 185 na “*ACM Digital Library*”, totalizando 234 artigos para a análise. As informações dos artigos foram exportadas no formato BibTex em cada plataforma, sendo posteriormente reunidas em um único arquivo “.*bib*”. Após análise manual deste arquivo conjunto, foram identificadas diversas inconsistências no formato de exportação entre as plataformas, que precisaram ser corrigidas antes das demais etapas. Por exemplo, artigos exportados a partir da plataforma

"CAPES" possuíam a anotação de tipo do conteúdo em português, "*@artigo*" ao invés de "*@article*", que foram substituídas para padronizar os conteúdos.

Por fim, o processo de exportação das plataformas levou a problemas de formatação no nome dos autores e em alguns títulos de publicações, fazendo com que caracteres como "*&*" se tornassem "*&*" e acentuação como "*ã*" se tornasse "*\~{a}*". Para manter a integridade das análises, esses problemas também foram manualmente identificados e corrigidos. Após o processo de deduplicação por igualdade de "DOI", os 234 artigos se tornaram 230.

Enriquecimento dos Artigos

Para realizar a leitura do arquivo *BibTex* foi utilizado o pacote *python pyBibX*, que é uma biblioteca para análise bibliométrica e cientométrica com recursos de inteligência artificial (PEREIRA; BASILIO; CARLOS, 2023). Fazendo uma análise inicial dos dados lidos, foi descoberto que apenas 88,70% deles tiveram o DOI capturado. Após conferência manual no documento, foi constatado que na verdade todos os artigos apontados possuem essa informação preenchida, compartilhando o prefixo comum "10.5555/".

Devido à incapacidade da biblioteca em lidar com esses artigos, já que o DOI é uma das informações mais importantes para o enriquecimento de informações, aliada à falta de documentação acessível, o pacote foi utilizado apenas para transformar os dados em formato *JSON*. Com a anotação manual do DOI nos artigos que não tiveram a informação capturada, somente as seguintes informações presentes inicialmente foram incluídas no novo arquivo:

- DOI (agora presente em 100% das publicações);
- Título (100% de presença);
- Resumo (80% de presença);
- Ano (100% de presença);
- Autores (100% de presença);
- Palavras-chave (92,61% de presença).

As demais informações a serem usadas para análise foram obtidas com a biblioteca Crossref, que utiliza referências cruzadas entre diversas publicações científicas para retornar informações mais completas sobre artigos. A obtenção de informações se baseou principalmente no DOI dos artigos, de forma a utilizar informações confiáveis na análise. Em último caso, quando a API do Crossref não retornava informações com base no DOI, foram feitas buscas baseadas no título e nos autores também através de código para maximizar a completude dos dados.

Com os dados retornados pelo Crossref, foram obtidas as seguintes informações:

- **Resumo do artigo**: Alguns dos resumos faltantes foram completados com os dados retornados da busca.
- **ORCID dos autores**: ORCID é um identificador persistente para pesquisadores, buscando diferenciá-los por algo além do nome. Por não haver consenso de utilização desse recurso na comunidade científica, uma parcela dos autores retornados nas consultas ao Crossref vinham com esse dado vazio.
    - Para maximizar a identificação de autores, foram geradas chaves únicas no formato "*UNKNOWN_XYZ*" baseadas na coincidência de nomes dos autores, onde "*XYZ*" é um número inteiro incrementado a cada ausência de ORCID.
- **Países dos autores**: Essa informação é obtida a partir dos dados de afiliação retornados pelo Crossref, que contêm o nome da instituição à qual o autor faz parte, o país dessa instituição e, em alguns casos, o departamento separados por vírgula. Assumindo que o país é a última informação do texto de afiliação, ele foi separado dos demais e analisado manualmente para todos os autores antes de ser incluído nos dados do artigo.
    - A análise manual se fez necessária porque, além de a presença do nome do país na afiliação não ser garantida, nas vezes em que ela ocorre existem diferenças, como "*United States*", "*USA*", "*US*", entre outras variantes. Dessa forma, foram feitas correções manuais para padronizar os nomes dos países e buscar o país de origem das instituições, quando apenas elas estavam incluídas na afiliação.

- **Instituições dos autores**: De forma similar à obtenção do nome do país, o nome das instituições também foi obtido a partir dos dados de afiliação retornados, havendo agrupamentos para as organizações "*IBM*", "*Microsoft*", "*Google*" e "*MIT*" em nomes padronizados. Isso se fez necessário porque, após verificação manual, existiam muitas ocorrências dessas instituições mas com nomes diferentes, como "*IBM Research*" e "*IBM Research AI*".
    - Eventualmente, algumas afiliações apresentavam o nome do departamento em maior destaque em relação ao nome da universidade, o que podia ser prejudicial às análises posteriores. Dessa forma, foi priorizada a captura de nomes de universidades através da detecção do texto "universidade" nos idiomas português, inglês, espanhol, alemão, francês e italiano.
    - Na ausência deste texto na afiliação, foram usadas as palavras "faculdade", "instituto", "escola", "centro" e "pesquisa" nas mesmas linguagens mencionadas anteriormente.
    - No caso da ausência de todos os termos citados anteriormente, o texto antes da primeira vírgula na afiliação foi considerado como a instituição do autor.
- **Referências utilizadas**: Obtidas iterando pela lista de referências utilizadas pelo artigo, de acordo com a resposta concedida pela Crossref API e contando a quantidade de DOIs únicos.
- **Quantidade de citações**: Informação diretamente obtida pela resposta da Crossref API, sendo preenchida como "-1" quando a chave não era retornada.

Ao fim do processo de enriquecimento, a quantidade de resumos desconhecidos caiu de 46 para 34, foram identificados 323 autores por ORCID e criados 477 ORCIDs no formato "*UNKNOWN_XYZ*". Além disso, foram também identificadas 250 instituições distintas para os artigos, que pertencem a 36 países distintos, mas 148 autores não possuíam dados de afiliação retornados pela Crossref. De todos os 230 artigos, 24 deles não possuem quantidade de citações.

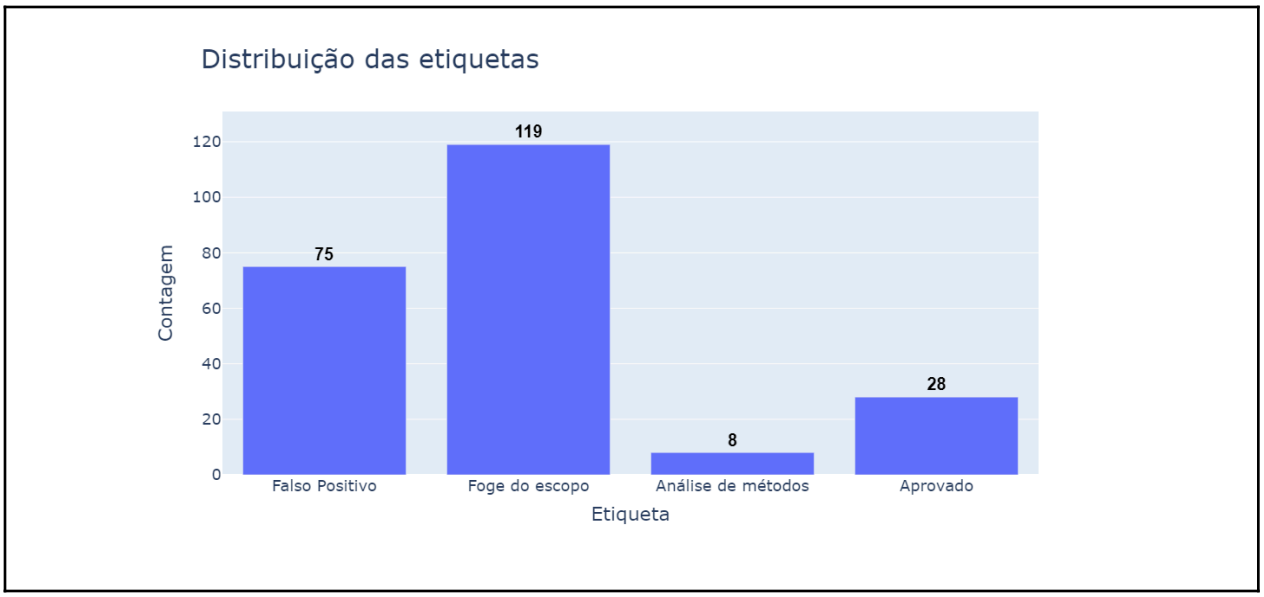
Filtros Preliminares

Antes de iniciar as análises, é importante reconhecer as limitações do sistema de filtragem das plataformas de publicação de artigos online, que possuem funcionalidades simples, como critérios de inclusão ou exclusão de palavras em determinados locais do texto. Enquanto esses critérios sejam o suficiente para eliminar uma grande quantidade de artigos irrelevantes, eles não levam em consideração o contexto dos textos produzidos, o que pode levar a um grande volume de falsos positivos, onde uma palavra desejada foi usada em um contexto indesejado.

Para eliminar a possibilidade de falsos positivos nas análises posteriores, foi feita a leitura dos títulos, resumos e palavras-chave de todos os 230 artigos restantes do filtro anterior, observando as características das publicações e o contexto das informações. Durante a leitura, os artigos foram classificados entre:

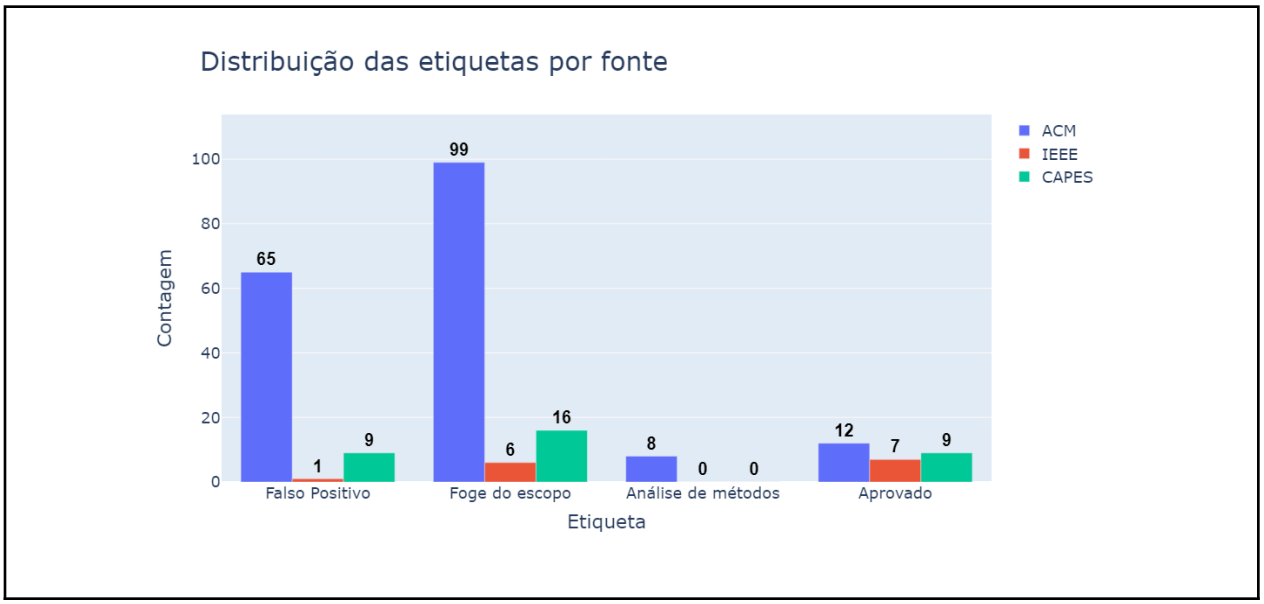
- **Falso Positivo**: não aborda explicação de IA;
- **Foge do Escopo**: aborda explicação de IA, mas não apresenta novos métodos;
- **Análise de Métodos**: aborda métodos já existentes;
- **Aprovado**: aparenta trazer novos métodos de explicação de IA.

Além disso, como ainda existiam artigos que não possuíam resumo, foi feita uma busca por eles em outras plataformas digitais com o objetivo de realizar o preenchimento manual dos resumos faltantes. A imagem a seguir mostra a distribuição dos artigos entre as classes propostas:



Durante a análise dos textos, foi priorizado o conceito de cobertura, onde se há a suspeita de que a publicação é de interesse para o trabalho, ela é automaticamente incluída na classe “Aprovado”. A eliminação dos artigos incluídos como material de leitura, mas que não se enquadram em todas as características necessárias, será detalhada na subseção “Filtragem Final”.

A imagem a seguir mostra a distribuição das classes atribuídas às publicações de cada uma das plataformas digitais (fontes) utilizadas. Apesar da plataforma IEEE ter sido a que retornou a menor quantidade de artigos, ela foi a que proporcionalmente entregou mais documentos que receberam a classe “Aprovado” (50%).



Ao fim da classificação manual, 75 artigos, representando 32,6% do total, foram descartados por terem recebido a classe “Falso Positivo”.

Análise Bibliométrica

Todos os 155 artigos que restaram após a classificação manual abordam o tema de explicação de IA em alguma extensão, como na descrição do uso de métodos contrafactuais para detecção de viés, ou na apresentação de novos métodos de explicação para tarefas diferentes de regressão e classificação. Sendo assim, todos eles serão considerados para a análise bibliométrica, pois auxiliam no entendimento e diagnóstico da área de explicações contrafactuais como um todo.

A análise bibliométrica dos artigos selecionados consiste em compilar dados quantitativos a partir das informações extraídas na etapa de enriquecimento, construindo tabelas, gráficos e grafos que ilustrem as relações entre as publicações. São criadas visões sobre os artigos em si, como a distribuição deles ao longo dos anos, tendências de palavras mais frequentes, análises geográficas sobre os países que realizaram mais contribuições. Além disso, são feitas análises sobre os autores mais frequentes, buscando relações de colaboração entre eles através de grafos de comunidade. As instituições de publicação também são exploradas de forma similar, identificando as mais frequentes e relações de comunidade.

As informações extraídas serão apresentadas na subseção “Análise Bibliométrica” do tópico de “Resultados” do presente documento.

Filtragem Final

Após as análises nos artigos relevantes ao tema, foram selecionados e baixados somente os 28 que foram classificados como “Aprovados” para a etapa de filtro final, que consiste na verificação do conteúdo dos documentos para determinar se eles são de fato de interesse para o trabalho. Para auxiliar neste processo, foi utilizada a ferramenta Notebook LM do Google, que é um modelo de linguagem para o qual se pode enviar arquivos e realizar perguntas sobre os elementos contidos neles. Para determinar a relevância das publicações para a revisão de literatura, o seguinte prompt foi construído e requisitado para cada um dos artigos individualmente:

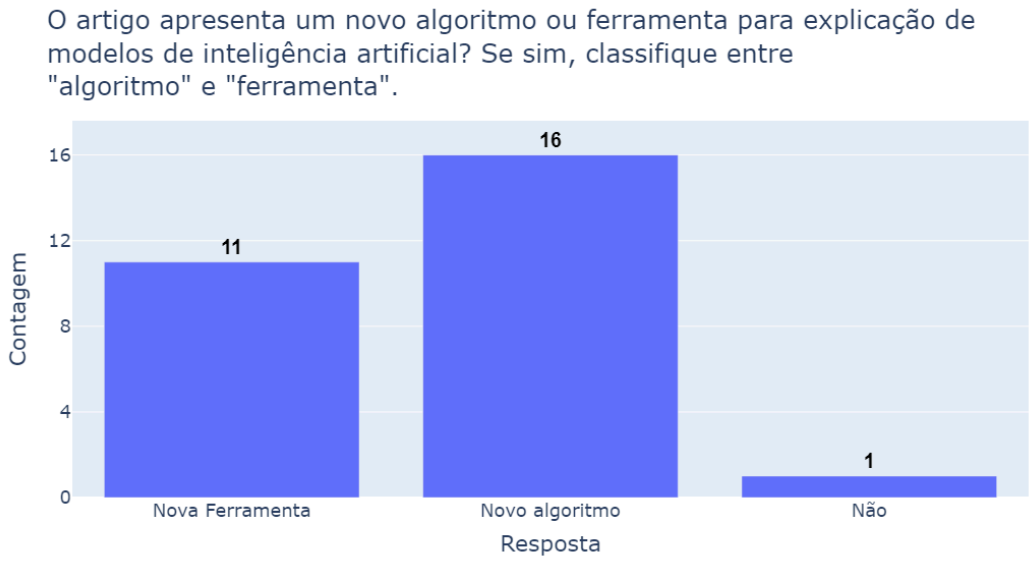
Para as próximas perguntas, priorize respostas curtas e objetivas:

1. O artigo apresenta um novo algoritmo ou ferramenta para explicação de modelos de inteligência artificial? Se sim, classifique entre "algoritmo" e "ferramenta".

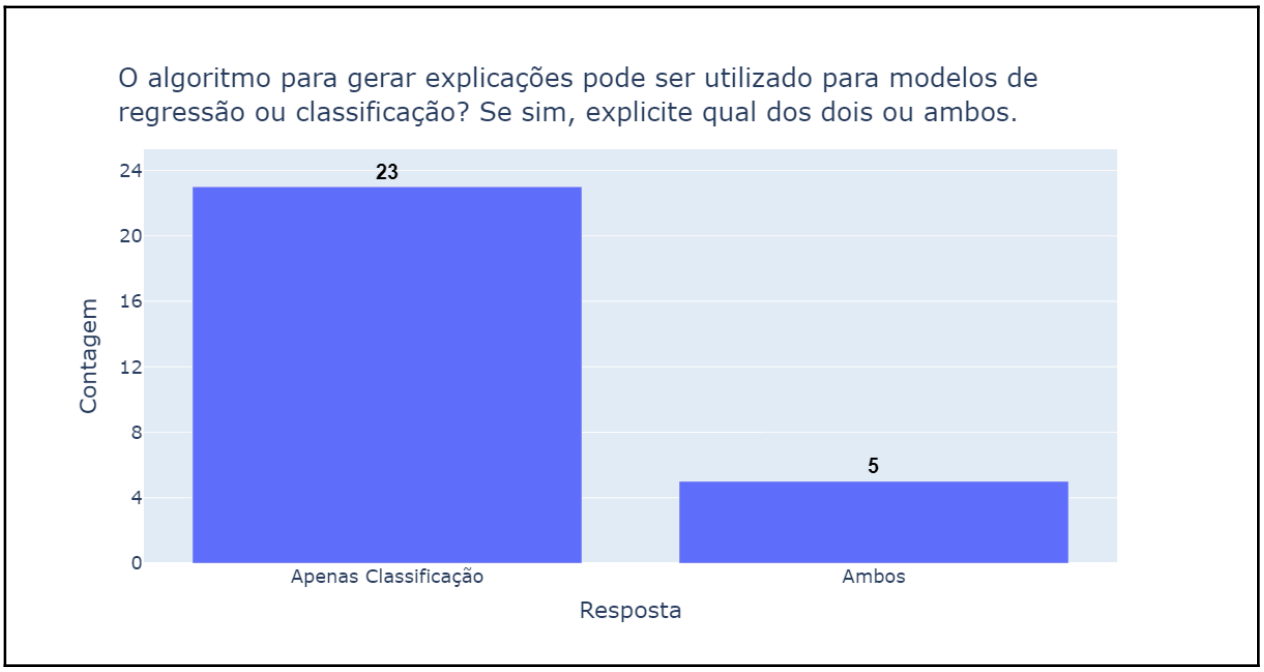
2. O algoritmo para gerar explicações pode ser utilizado para modelos de regressão ou classificação? Se sim, explicite qual dos dois ou ambos.

3. As explicações geradas pelo algoritmo proposto são contrafactuais?

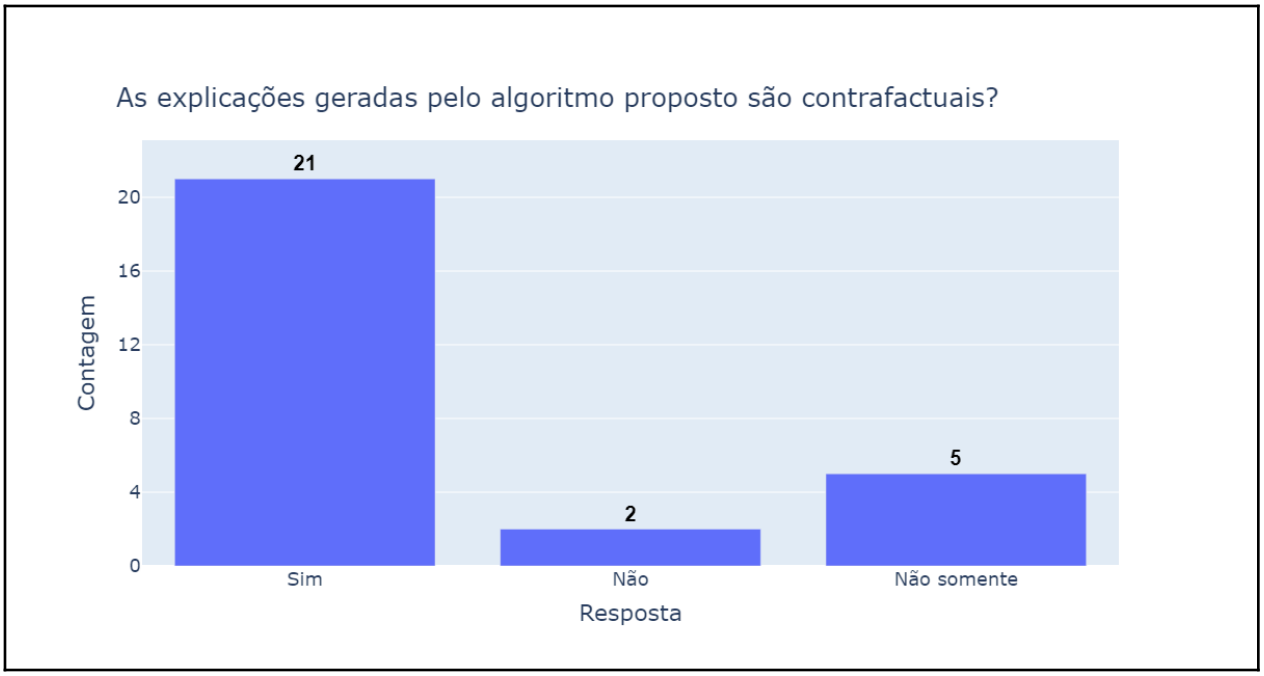
Na primeira pergunta um dos artigos foi eliminado por se tratar de melhorias feitas em uma ferramenta já existente. O gráfico a seguir mostra a distribuição das quantidades de artigos entre as possibilidades de resposta.



Nenhum dos artigos selecionados na classificação manual trata de tarefas de IA diferentes de classificação ou regressão, portanto, nenhum deles foi eliminado na segunda pergunta, como mostra o gráfico a seguir.



Já na terceira pergunta, foram eliminadas duas publicações por não fornecerem resultados baseados em explicações contrafactuais. O gráfico a seguir mostra a distribuição de artigos classificados entre respostas possíveis para a pergunta.



Ao fim do processo de filtragem com o Notebook LM, restaram 25 artigos para leitura e análise qualitativa posterior. Entretanto, um novo filtro foi realizado, buscando priorizar os textos

publicados em veículos de alto impacto acadêmico e elevar a assegurar a qualidade geral das publicações. Para isso, foram identificados os veículos de publicação de todos os 25 artigos e coletadas as informações de QUALIS CAPES, *Journal Impact Factor* (*JIF*) e *h-Index*.

O nível QUALIS foi obtido a partir do relatório da área de computação da comissão de eventos quadrienal 2017 - 2020[1], já o fator de impacto foi coletado com consultas à página online do Web of Science Journal[2] e o h-Index foi recuperado da página online SciMago[3]. É importante destacar que muitos artigos da computação são publicados em eventos e conferências, que não costumam receber níveis de qualidade da mesma forma que os periódicos e revistas, tornando a tabela incompleta. No entanto, como a fonte do QUALIS CAPES é voltada especificamente para computação, esta limitação pode ser superada já que as conferências recebem a mesma importância que os periódicos.

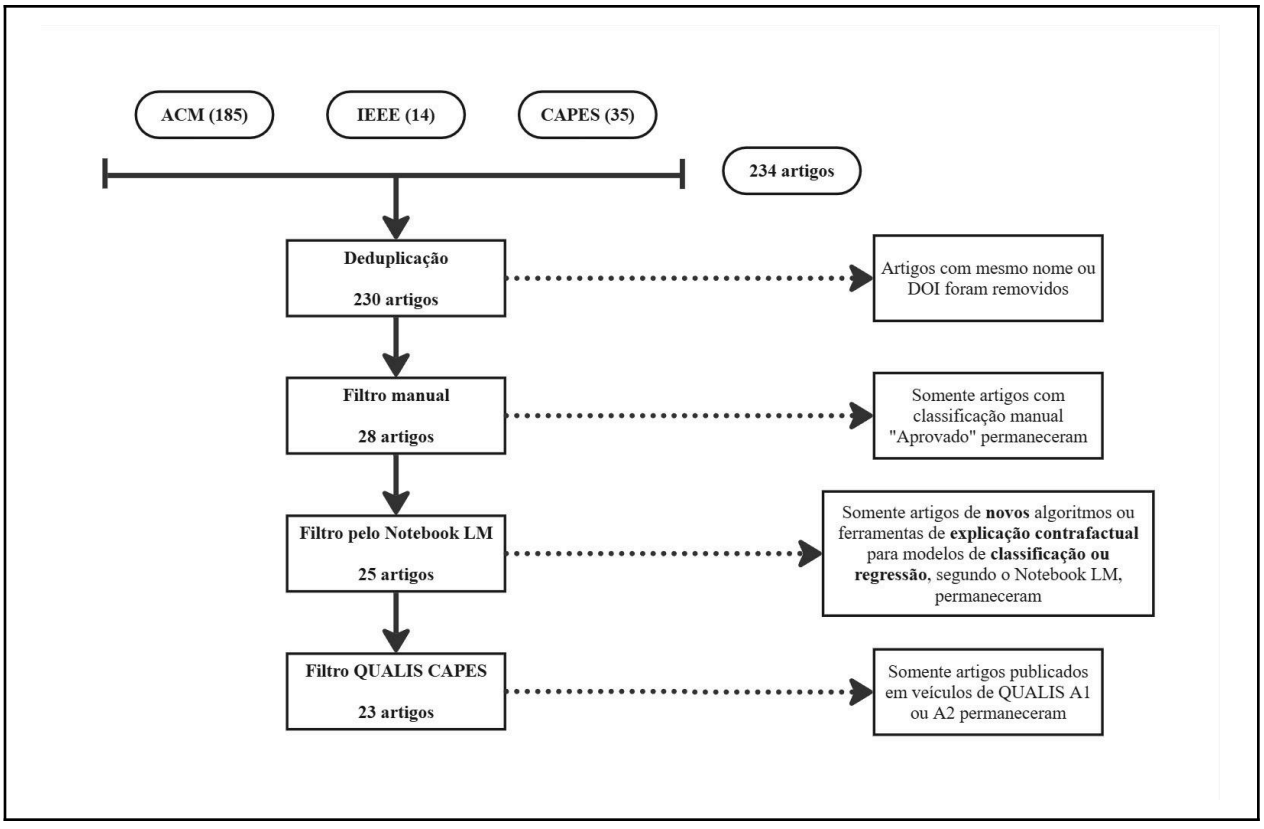
A tabela a seguir mostra as informações coletadas para cada um dos veículos de publicação, ordenados pela quantidade de artigos publicados em cada um. Como o QUALIS é a única métrica presente em todas as linhas, ele foi utilizado para a filtragem, permanecendo somente os artigos publicados em veículos que tenham nível A1 ou A2, o que representa a eliminação de duas outras publicações antes das análises qualitativas.

Veículo	Quantidade de publicações	Qualis CAPES	Journal Impact Factor	h-Index
ACM COMPUTING SURVEYS	4	A1	21.1	232
IEEE Access	3	A1	3.7	290
DATA MINING AND KNOWLEDGE DISCOVERY	2	A1	5.3	123
ACM International Conference on Information & Knowledge Management (CIKM)	2	A1	-	-
IEEE INTELLIGENT SYSTEMS	1	A1	5.6	145
INFORMATION FUSION	1	A1	16.1	179
PROCEEDINGS OF THE VLDB ENDOWNMENT	1	A1	3.4	165
International journal of data science and analytics	1	A1	2.6	36

[1] https://shorturl.at/q39Vz
[2] https://wos-journal.info/
[3] https://www.scimagojr.com/

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH	1	A1	5.9	319
IEEE Congress on Evolutionary Computation (CEC)	1	A1	-	-
Proceedings of the ACM on management of data (SIGMOD)	1	A1	1.2	149
ACM Genetic and Evolutionary Computation Conference (GECCO)	1	A1	-	-
MACHINE LEARNING	1	A1	5.8	175
IEEE International Joint Conference on Neural Network (IJCNN)	1	A1	-	-
International Conference on Multimedia Retrieval (ICMR)	1	A2	-	-
Computer Graphics Forum	1	A2	2.9	148
PRICAI	1	Indefinido	-	-
IEEE Asia-Pacific Conference on Computer Science and Data Engineering	1	Indefinido	-	-

Para fins de ilustração, a figura a seguir mostra o progresso dos filtros realizados para a redução nas quantidades de publicações desde o início da execução do trabalho até o momento atual:



Análise Qualitativa

Ao fim do processo de filtro, todos os 23 artigos restantes foram lidos e analisados de forma a identificar características marcantes deles, que serão discutidas e explicadas ao longo da subseção "Análise Qualitativa" da seção de resultados. Uma das análises feitas é relativa à validade do filtro realizado pelo Notebook LM, em que falsos positivos aprovados pela ferramenta foram removidos da análise. Esses artigos ainda serão citados na seção de resultados, explicando o motivo pelo qual não se enquadram e qual a possível explicação para o erro de interpretação da inteligência artificial.

Entre as características extraídas dos métodos propostos, está a **Tarefa dos Modelos Alvo**, que verifica se as explicações são feitas para modelos apenas de classificação ou também abrangem modelos de regressão. A identificação dessa característica foi baseada na disponibilidade imediata do recurso, sem a necessidade de ajustes maiores do que informar ao método que o modelo a ser explicado é de classificação ou regressão. No entanto, para métodos agnósticos ao modelo e que não utilizam a estrutura interna de funcionamento dele, é possível segmentar os resultados da regressão entre intervalos desejáveis e indesejáveis como forma de simular uma classificação binária. Dessa forma, quando as ferramentas possuem suporte somente à tarefa de classificação, é possível realizar esta extensão de forma relativamente simples para que elas também lidem com tarefas de regressão.

De forma similar à característica anterior e considerando que todos os métodos podem ser utilizados para tarefas de classificação, também foi observado se os métodos **Suportam Modelos Multiclasse** de forma imediata. Nos casos negativos, é possível empregar uma estratégia parecida à proposta no parágrafo anterior para estender os métodos para funcionar em tarefas de regressão, ao agrupar as diversas classes disponíveis entre desejáveis e indesejáveis fazendo com que a classificação passe a ser essencialmente binária,

No geral, a maioria dos métodos permite algum nível de interação com o usuário, seja para definição de hiperparâmetros, mudanças em funções objetivo ou seleção do modo de cálculo de distância. No entanto, alguns se sobressaem aos demais ao permitir que os usuários definam restrições e preferências na geração dos contrafactuais, como um conjunto de *features* que devem permanecer inalteradas, intervalos de valores permitidos, sentidos de alteração (incremento ou

decremento), entre outras. Durante a análise das publicações, foi observada a capacidade de **Interação com o Usuário** na produção e seleção de contrafactuais, como uma forma de identificar métodos mais flexíveis.

Além disso, os métodos também podem ser discernidos em relação ao **Tipo de Dados** esperados na entrada, como tabulares, textuais, imagens, entre outros. Note que quando um método espera apenas dados tabulares não significa uma impossibilidade dele de fornecer explicações para textos e imagens, apenas que esses formatos não têm suporte imediato, sendo necessária uma etapa de pré-processamento do usuário antes da utilização da ferramenta. No caso de textos, por exemplo, é possível remover os acentos e separar as palavras em um vetor binário indicando a presença ou ausência, que é um formato tabular válido. Para imagens, também é possível fazer a segmentação das regiões em superpixels e codificá-los em um vetor binário de forma semelhante. Nesses casos de adaptação, é necessário o cuidado de converter os vetores de volta em palavras ou imagens antes de enviá-los ao modelo das predições, evitando erros no processo.

Os métodos também foram observados em relação à **Especificidade de Modelos Alvo**, de forma a identificar aqueles que são agnósticos ao algoritmo utilizado e os que são específicos. Outra característica extraída é relativa ao **Tipo de Explicação** fornecida, que separa métodos que entregam explicações locais ou globais, além de especificar se as explicações são apenas contrafactuais, ou se também são factuais, por exemplo.

Também foram feitas observações sobre o **Método de Geração dos Contrafactuais** como uma forma de organizar as explicações dos algoritmos e ferramentas propostos, onde métodos similares são abordados próximos uns dos outros. Foram coletadas informações sobre a necessidade de **Acesso ao Modelo Alvo**, que verifica se existe a necessidade de consultas irrestritas ao modelo para gerar as explicações, o que pode ser um impeditivo em alguns casos. Por fim, foi observado se existe necessidade de **Acesso a um Dataset** que seja relevante ao domínio da tarefa para obter as explicações, que também pode ser um impeditivo para alguns usuários.

Alguns dos algoritmos e ferramentas são baseados em modelos substitutos para a produção dos resultados, de forma a evitar a utilização do modelo alvo ou gerar explicações a partir de modelos mais simples. No entanto, essa técnica pode ser acompanhada por falhas de aproximação do

modelo original devido à simplicidade do substituto, que levam a contrafactuais pouco confiáveis. Com isso, também foi observado se os métodos **Utilizam Modelos Substitutos** como técnica de explicação, como forma de alertar os usuários durante a escolha para essas limitações. Para facilitar o acesso às implementações criadas pelos autores, durante a leitura também foi verificado se os métodos são de **Código Aberto**, coletando links para o repositório em caso positivo.

Resultados

Os resultados da análise bibliométrica e qualitativa foram obtidos em dois momentos diferentes do processo de filtragem, o que leva a um corpo composto por 155 artigos para a bibliometria, contra 23 para a análise qualitativa preliminar. É importante destacar que dos 23 documentos restantes, durante o processo de leitura foram identificados que 4 deles se tratavam de falsos positivos que passaram pelo crivo do Notebook LM, reduzindo o universo final de análises qualitativas para 19 publicações.

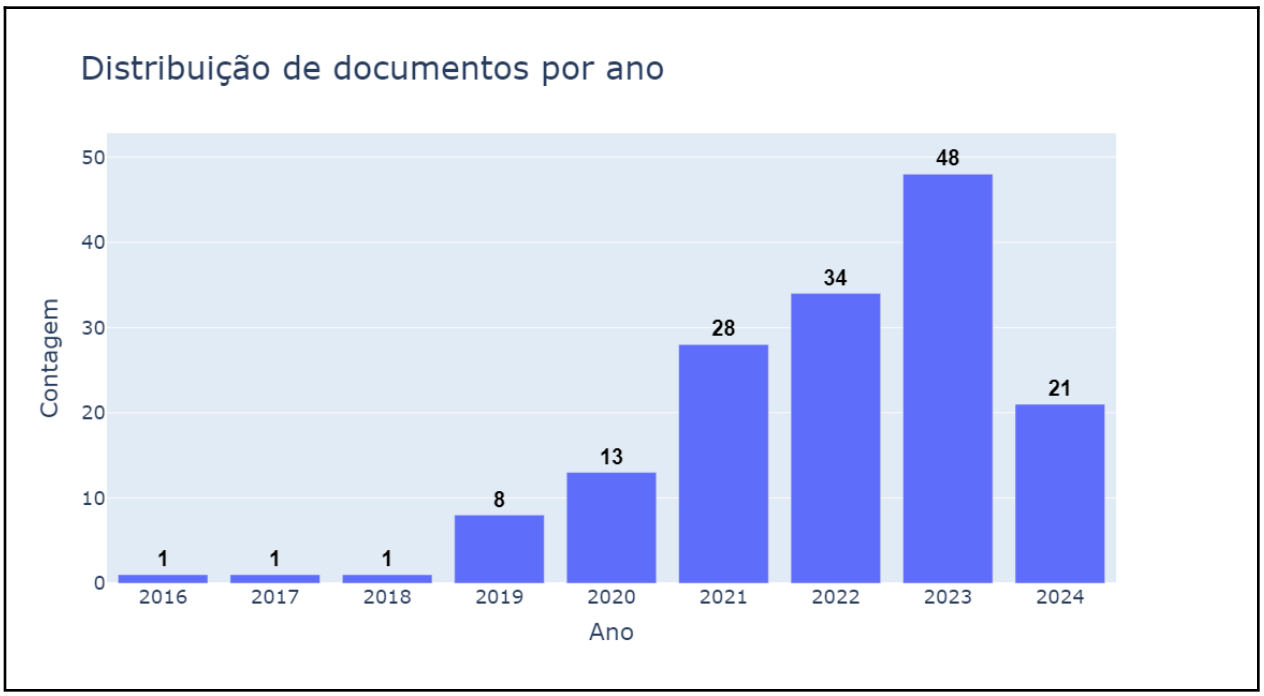
Os quatro artigos descartados podem ser divididos em 2 grupos de dois documentos cada, de forma a agrupar as justificativas de exclusão. O primeiro grupo engloba as publicações "*COGAM: Measuring and Moderating Cognitive Load in Machine Learning Model Explanations*" e "*Decision Boundary Visualization for Counterfactual Reasoning*", que não são métodos de explicação contrafactual. As ferramentas propostas nos dois documentos são na verdade utilizadas para criar análises globais do desempenho das *features* dos modelos, apresentando características como correlação e valor de importância. Essas informações eram então fornecidas aos usuários, que poderiam criar explicações contrafactuais para as decisões do modelo a partir delas.

O segundo grupo conta com os artigos "*Counterfactual Explanations of Black-box Machine Learning Models using Causal Discovery with Applications to Credit Rating*" e "*Counterfactual Explanations With Multiple Properties in Credit Scoring*". O primeiro deles não gera explicações, mas sim grafos causais entre as *features* disponíveis no modelo, de forma a facilitar a geração de contrafactuais por outros métodos que possam se beneficiar deste recurso, como o *LEWIS*, que é uma das ferramentas analisadas neste trabalho. O segundo é uma extensão direta do método

baseado em algoritmo genético produzido pelos mesmos autores chamado *MAGECE*, que também é analisado neste trabalho, mas desta vez permitindo que múltiplas propriedades sejam consideradas na função objetivo usada para a geração dos contrafactuais.

Análise Bibliométrica

Análise bibliométrica é um método quantitativo para analisar a produção científica de determinada área, identificando padrões, tendências e relações através de dados estatísticos coletados. O primeiro ponto de análise será relacionado aos artigos, produzindo recursos visuais que ilustram as descobertas feitas. O gráfico a seguir mostra a quantidade de artigos publicados por ano desde 2016, já que todos os artigos de 2015 foram excluídos por serem falsos positivos durante os filtros preliminares.



Note que a partir do ano de 2019 começa a haver um crescimento no número de publicações da área, chegando a um pico em 2023. Porém, como somente foram contemplados artigos publicados até a primeira metade de 2024, não é possível concluir se a área está caminhando ou não para uma estagnação. Além da análise de tendência anual na quantidade de publicações, foi feita também uma análise de tendência nos termos mais frequentes dos resumos e títulos ao longo dos anos. Para produzir o gráfico de distribuição das palavras mais frequentes ao longo dos anos,

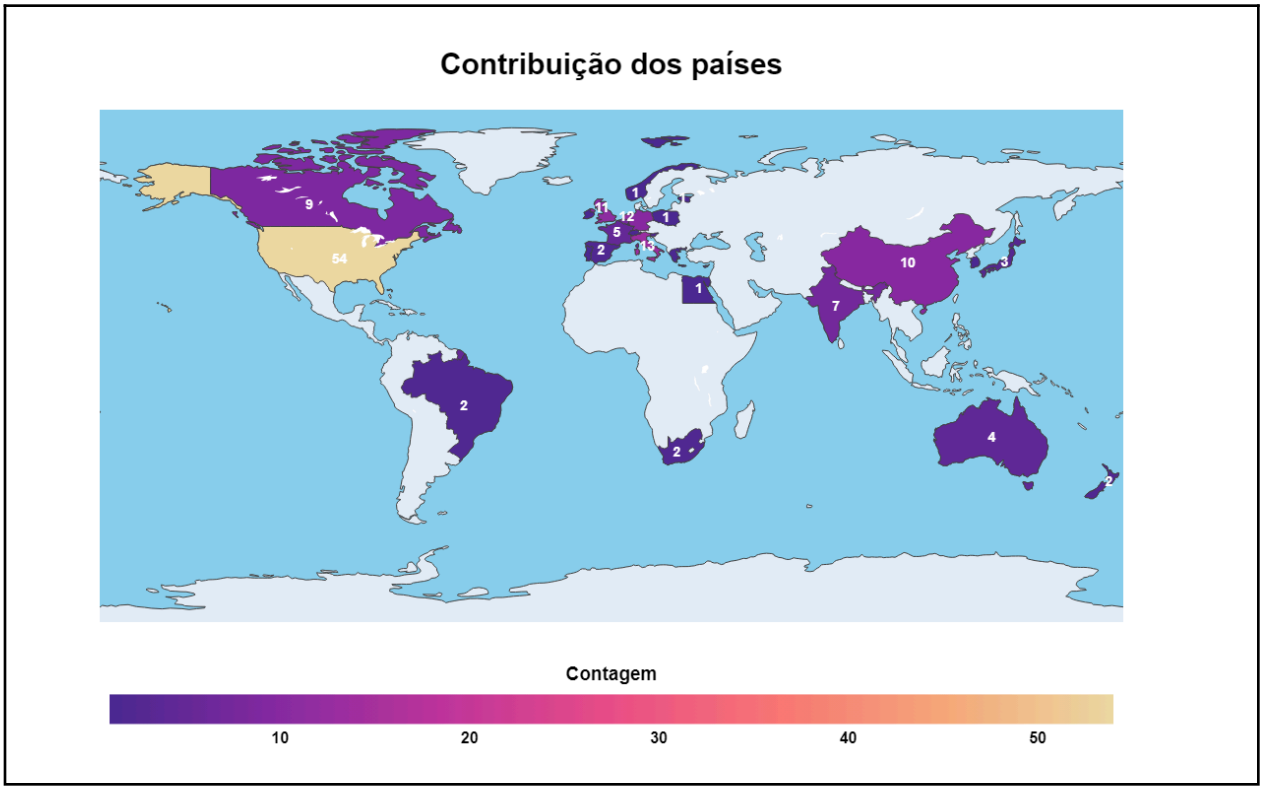
inicialmente os títulos e resumos das publicações foram concatenados e passaram por um processo de remoção de *stopwords*, que são palavras que não trazem significado ao texto, como artigos e conectivos. As palavras restantes passaram então por uma acumulação de frequência, sendo selecionadas somente as dez palavras mais frequentes de cada ano para formar o gráfico abaixo:

Pela construção dos filtros na seção “Coleta dos Artigos”, todas as publicações devem possuir o termo “*counterfactual*” em ao menos um local do documento, seja no título, resumo ou conteúdo. Entretanto, esse termo só começou a ser mais popular nos artigos a partir do ano de 2020, sendo a palavra mais utilizada por quatro anos seguidos e estando em quarto lugar nas publicações feitas até junho de 2024. Esses gráficos indicam não somente uma popularidade crescente da área, mas também uma grande importância de métodos contrafactuais para realizar a explicação de IAs.

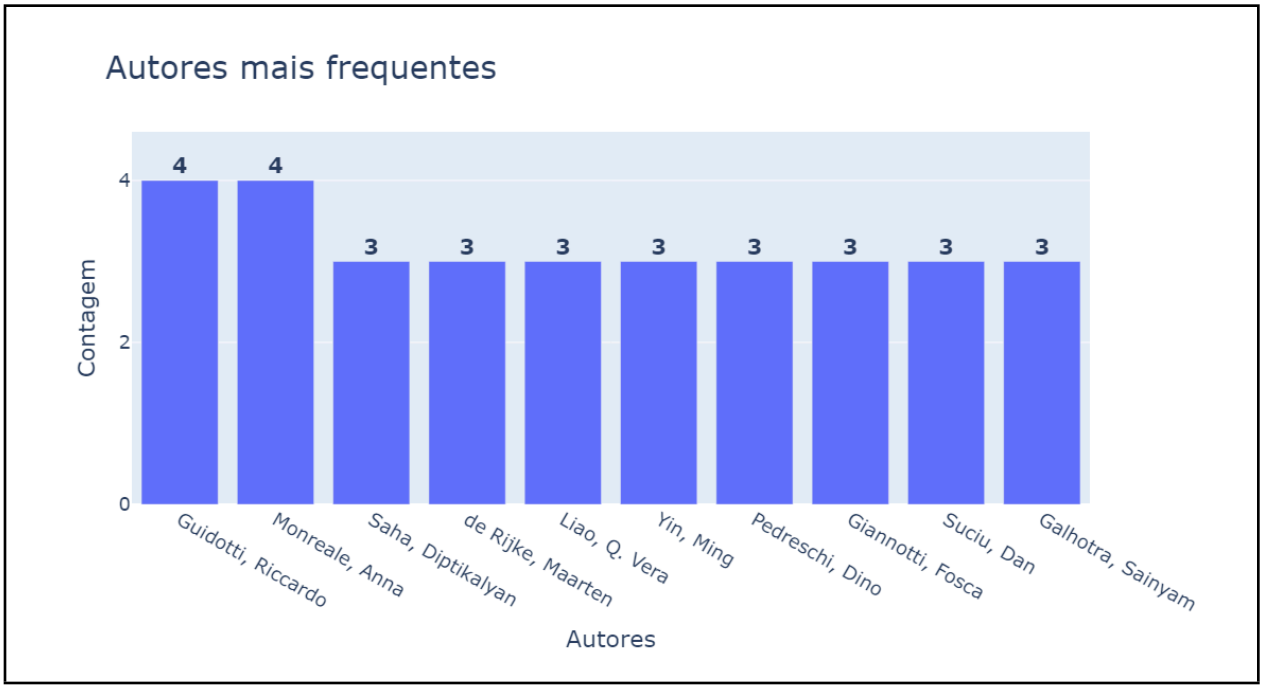
A partir dos dados de localização extraídos manualmente durante o processo de enriquecimento, foi construída uma análise geográfica sobre os padrões de publicações, mostrando a distribuição dos artigos nos países em que as instituições associadas a cada um dos autores estão localizadas. A tabela a seguir mostra os dez países com maior quantidade de publicações únicas em ordem decrescente.

País	Publicações únicas
United States	54
Italy	13
Netherlands	12
United Kingdom	11
China	10
Germany	10
Canada	9
India	7
Belgium	7
France	5

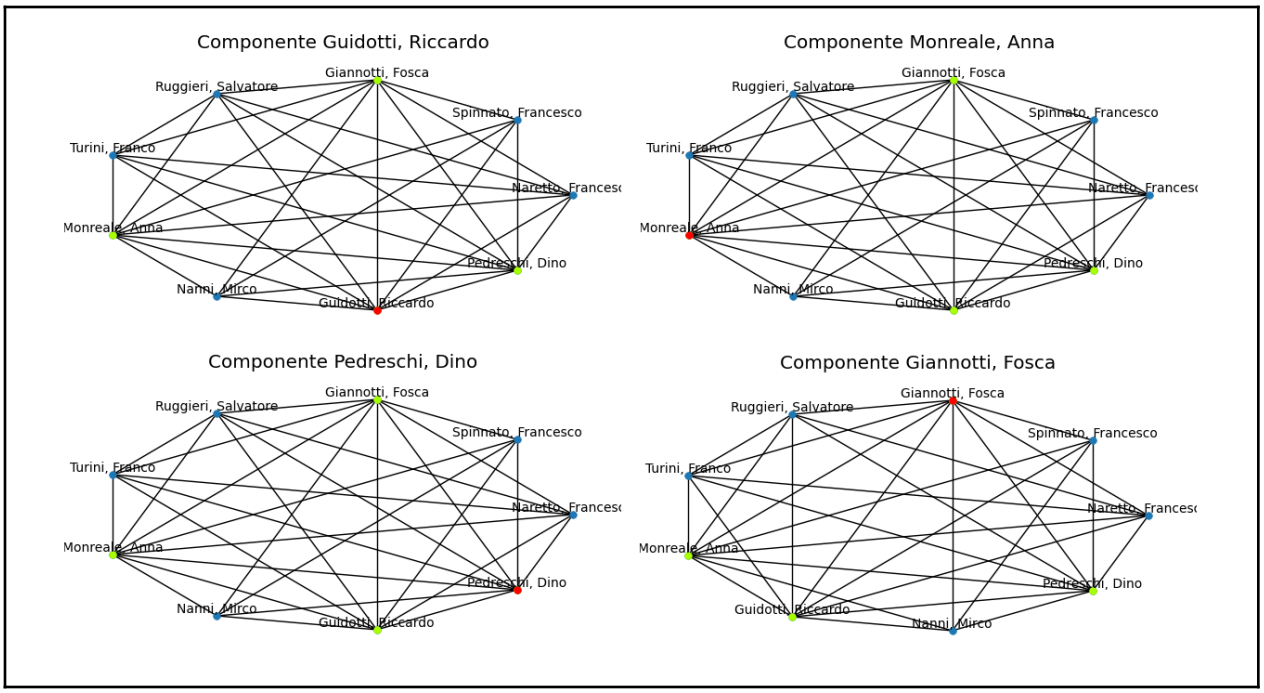
Note que há uma diferença expressiva na quantidade de artigos publicados nos Estados Unidos e nos demais países, em uma liderança marcada principalmente pela grande concentração de empresas de tecnologia que financiam pesquisas na área. Em uma perspectiva mais visual, contemplando todos os países do globo, observe o mapa de calor a seguir, mostrando as quantidade de publicações únicas por nação.



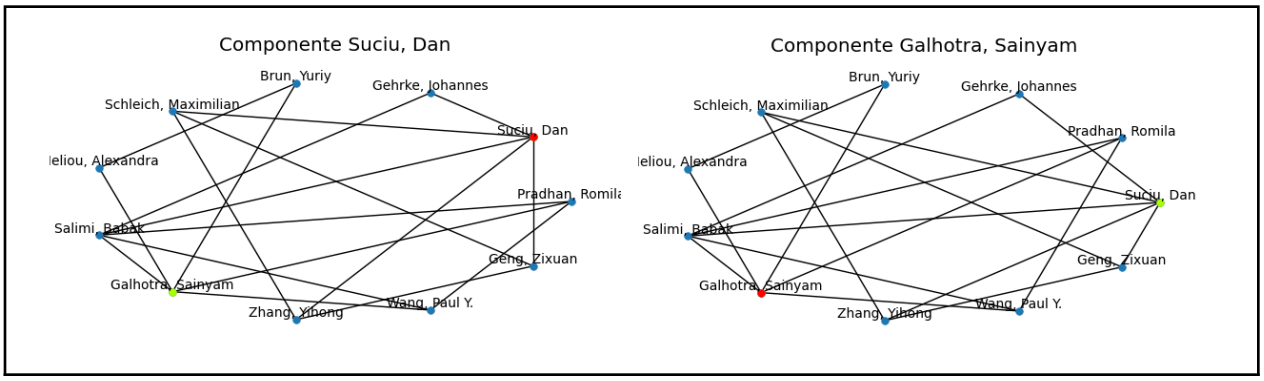
Apesar de não existirem artigos escritos em português na base, dois deles foram produzidos por brasileiros, colocando o Brasil à frente de países europeus, como Portugal, Suíça, Irlanda e Noruega, ainda que por uma diferença muito pequena. Além disso, enquanto a Itália está em segundo lugar no ranking de países que mais produzem publicações relacionadas ao tema, os dois primeiros lugares de autores mais frequentes pertencem aos italianos Riccardo Guidotti e Anna Monreale, como mostra o gráfico a seguir. Ambos são professores na Universidade de Pisa, sendo nomes famosos na área de explicação de IA.



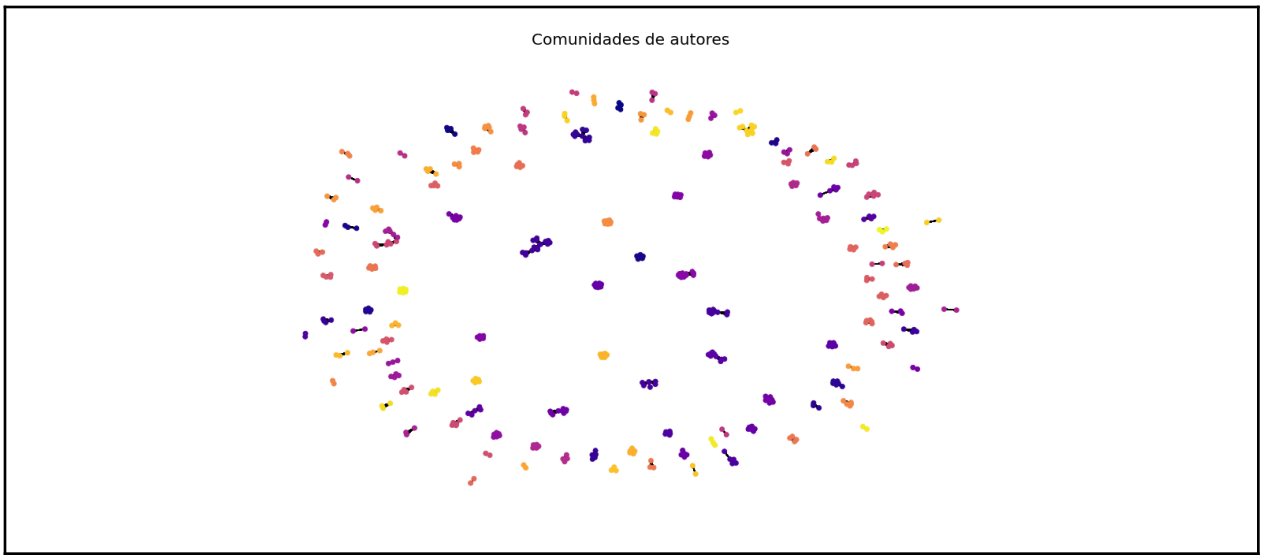
Buscando explorar as relações dos principais pesquisadores, foram construídos grafos de colaboração, onde as arestas indicam que existe ao menos um artigo publicado que foi co-autorado pelos pares de vértices. Ao analisar os grafos gerados, foi constatado que o círculo de colaborações de Riccardo Guidotti, Anna Monreale, Dino Pedreschi e Fosca Giannotti, todos professores na Universidade de Pisa, é exatamente igual.



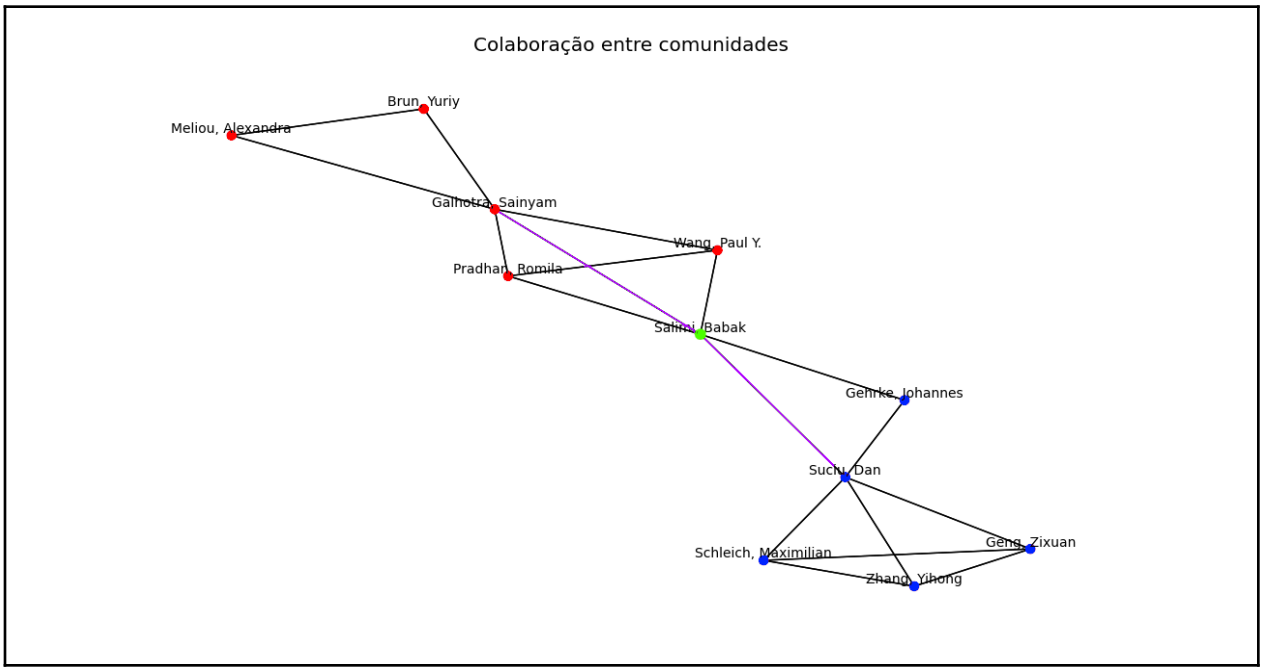
Um comportamento similar ocorre para os autores Dan Suciu e Sainyam Galhotra, que apesar de não compartilharem a instituição de ensino, também possuem o mesmo círculo de colaboração para publicações, como mostra a imagem a seguir. Note, porém, que não existe aresta direta entre esses dois autores, com a única ligação entre ambos sendo o autor Babak Salimi.



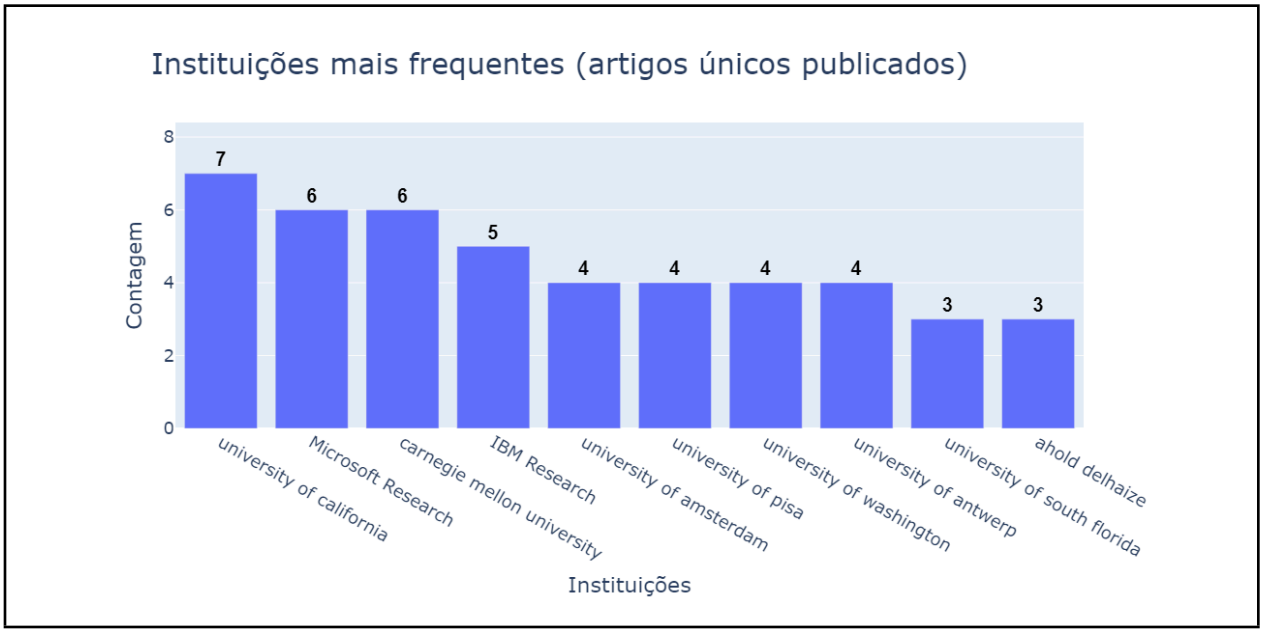
Com a detecção dos padrões de colaboração mostrados nos grafos acima, foi feita uma análise de comunidades com o algoritmo Girvan Newman, buscando identificar relações entre diferentes comunidades. A imagem a seguir mostra uma visão global de todas as comunidades identificadas, baseadas na colaboração entre diferentes autores em um mesmo artigo.



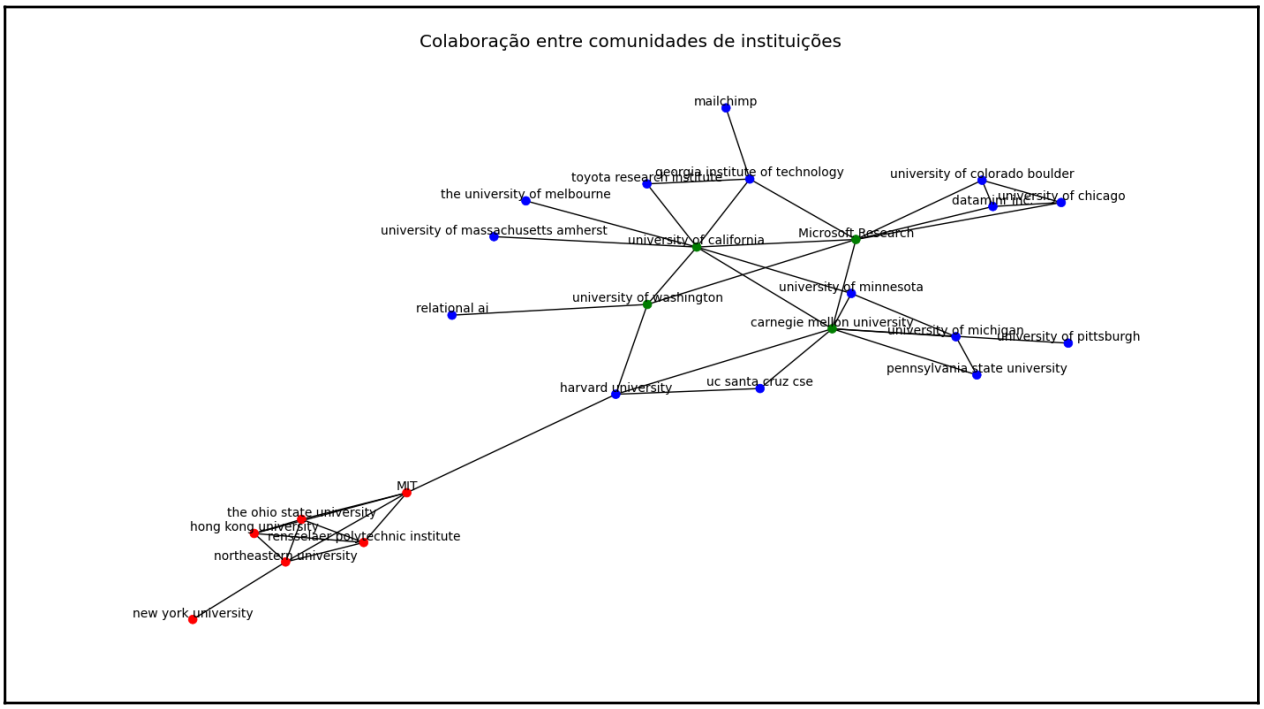
Com base nas comunidades construídas, foi detectada uma única colaboração entre comunidades, sendo ela já discutida anteriormente neste tópico. A imagem abaixo mostra como as comunidades dominadas por Dan Suciu e Sainyam Galhotra, que possuem arestas para todos os vértices de suas próprias comunidades, estão sendo conectadas pelo autor Babak Salimi. As cores dos vértices foram modificadas manualmente para vermelho e azul, buscando uma melhor distinção. O vértice de Babak Salimi também foi modificado para a cor verde, dando maior destaque. As arestas na cor roxa representam a união entre os autores dominantes de cada comunidade.



Além das análises feitas quanto aos artigos e autores, também serão feitas análises relativas às instituições, acadêmicas ou não, associadas às publicações. Porém, como um único artigo pode ser escrito por múltiplos autores, que tendem a ser de uma mesma instituição, foram consideradas quantidade únicas de publicações para evitar desigualdades entre organizações que dispõem de grandes equipes de pesquisa e as demais. Com isso, a instituição com maior quantidade de artigos únicos publicados é a Universidade da Califórnia com 7 publicações, seguida pela Microsoft Research e Universidade Carnegie Mellon, com 6 cada.



De forma similar ao que foi feito com os autores, a colaboração entre comunidades também foi analisada para as instituições, chegando à conclusão de que as três principais organizações pertencem a um mesmo agrupamento. Novamente, as comunidades foram coloridas manualmente com azul e vermelho para melhorar a distinção. A principal diferença é que as instituições que aparecem na listagem de mais frequentes da imagem anterior foram coloridas de verde, destacando o fato de uma única comunidade contar com 4 das 10 principais instituições.



A ligação entre os grupos é feita pela Universidade de Harvard e o Instituto de Tecnologia de Massachusetts (MIT), que possuem uma organização diferente entre as comunidades. Além de a comunidade azul apresentar maior quantidade de vértices, ela também tem como característica a colaboração com com instituições privadas, como Mailchimp, Toyota Research Institute, Microsoft Research, entre outros. Já a comunidade vermelha possui menos vértices e tem colaborações apenas com outras instituições de ensino.

Análise Qualitativa

As informações já explicadas durante a seção “Análise Qualitativa” da metodologia deste trabalho foram coletadas durante o processo de leitura e condensadas na tabela da página seguinte, que tem o objetivo de facilitar a escolha de métodos estado da arte para aplicação em problemas reais. Os métodos estudados foram agrupados pelo método de geração de contrafactuais implementado, de forma a criar uma ordem lógica de explicação baseada na similaridade entre as técnicas empregadas. A continuidade desta seção após a apresentação da tabela consiste na explicação da ideia geral das técnicas de geração, destacando qualquer inovação ou característica especial que cada um dos métodos apresente

Nome	Técnica	Dados	Agnóstico	Interativo	Múltiplas Explicações[4]	Multiclasse	Regressão	Dataset	Modelo Alvo	Modelo Substituto	Código Aberto
SkylineCF	Gulosa	Tabulares	✔	✔			✔	✔	✔		
RF-OCSE	Gulosa	Tabulares				✔			✔		✔
FCE	Força bruta	Tabulares	✔	✔					✔		✔
COVISATEE*	Força bruta	Imagens	✔		✔	✔		✔			
FACET	Força bruta	Tabulares		✔		✔		✔	✔		✔
CF-SHAP	Força bruta	Tabulares						✔			✔
LEWIS	Programação Inteira	Tabulares	✔	✔	✔			✔	✔		
ARLIC*	Programação Inteira	Tabulares		✔				✔	✔		✔
LORESa	Algoritmo Genético	Tabulares	✔	✔	✔	✔			✔	✔	✔
GeCo	Algoritmo Genético	Tabulares	✔	✔				✔	✔		✔
CARE	Algoritmo Genético	Tabulares	✔	✔		✔	✔	✔		✔	✔
LORE	Algoritmo Genético	Tabulares	✔		✔			✔	✔	✔	✔
MAGECE*	Algoritmo Genético	Tabulares	✔					✔	✔		
RASCEN-PSO*	Enxame de Partículas	Tabulares	✔			✔			✔		
CF-PSO* & CF-DE*	Enxame de Partículas & Evolução Diferencial	Tabulares	✔			✔			✔		✔
CFNOW	Busca Tabu	Tabulares, Imagens e Textos	✔			✔			✔		✔
ASTERYX	Heurística	Tabulares	✔		✔			✔		✔	
MAPOCAM	Branch and Bound	Tabulares	✔	✔				✔	✔		✔
SINAR-RL*	Aprendizado por Reforço	Tabulares	✔	✔				✔	✔		✔

A tabela da página seguinte associa os nomes dos métodos com os respectivos artigos em que foram publicados, como forma de facilitar a localização.

[4] A coluna de "Múltiplas Explicações" está relacionada ao fato do método propor ou não explicações extras que sejam de tipos diferentes das contrafactuais, como factuais ou globais.
* Os nomes de métodos marcados com asterisco foram nomeados durante a execução deste projeto, já que os autores não nomearam, para facilitar referências.

Nome do método	Artigo publicado
SkylineCF	The Skyline of Counterfactual Explanations for Machine Learning Decision Models
RF-OCSE (**R**andom **F**orest **O**ptimal **C**ounterfactual **S**et **E**xtractor)	Random forest explainability using counterfactual sets
FCE (**F**eedback based **C**ounterfactual **E**xplanations)	FCE: Feedback Based Counterfactual Explanations for Explainable AI
COVISATEE* (**CO**unter **VI**sual **AT**tributes and **E**xamples **E**xplanations)	Explaining with Counter Visual Attributes and Examples
FACET (**F**ast **A**ctionable **C**ounterfactuals for **E**nsembles of **T**rees)	FACET: Robust Counterfactual Explanation Analytics
CF-SHAP (**C**ounter**F**actual **SHAP**)	Counterfactual Shapley Additive Explanations
LEWIS (Nomeado em homenagem ao filósofo analítico David Lewis)	Explaining Black-Box Algorithms Using Probabilistic Contrastive Counterfactuals
ARLIC* (**A**ctionable **R**ecourse in **L**inear **C**lassification)	Actionable Recourse in Linear Classification
LOREsa (**LO**cal **R**ule based **E**xplanations but **s**table and **a**ctionable)	Stable and actionable explanations of black-box models through factual and counterfactual rules
GeCo	GeCo: Quality Counterfactual Explanations in Real Time
CARE (**C**oherent **A**ctionable **R**ecourse **E**xplanations)	CARE: coherent actionable recourse based on sound counterfactual explanations
LORE (**LO**cal **R**ule based **E**xplanations)	Factual and Counterfactual Explanations for Black Box Decision Making
MAGECE* (**M**odel-**A**gnostic **GE**netic **C**ounterfactual **E**xplanations)	Model-Agnostic Counterfactual Explanations in Credit Scoring
RASCEN-PSO* (**RA**shomon **S**ets of **C**ounterfactual **E**xplanations with **N**iching **P**article **S**warm **O**ptimization)	Producing Diverse Rashomon Sets of Counterfactual Explanations with Niching Particle Swarm Optimization Algorithms
CF-PSO* (**C**ounter**F**actuals by **PSO**) CF-DE* (**C**ounter**F**actuals by **DE**)	Evolving Counterfactual Explanations with Particle Swarm Optimization and Differential Evolution
CFNOW (**C**ounter**F**actuals **NOW**)	A model-agnostic and data-independent tabu search algorithm to generate counterfactuals for tabular, image, and text data
ASTERYX	ASTERYX: A model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations
MAPOCAM (**M**odel-**A**gnostic **P**areto-**O**ptimal **C**ounterfactual **A**ntecedents **M**ining)	Mining Pareto-optimal counterfactual antecedents with a branch-and-bound model-agnostic algorithm
SINAR-RL* (**S**ynthesized **I**nterventions for **A**lgorithmic **R**ecourse with **R**einforcement **L**earning)	Synthesizing explainable counterfactual policies for algorithmic recourse with program synthesis

Os algoritmos **Gulosos** são parte de um paradigma que prioriza a melhor escolha imediata, de acordo com algum critério de otimização, com o objetivo de se obter uma melhor escolha global ao final do processo, mas sem a garantia de que isso ocorrerá em todos os casos. Este paradigma pode ser aplicado a diversas tarefas, como selecionar o melhor candidato a contrafactual baseado em uma função multi objetivos aplicada a um conjunto ordenado para formar uma fronteira de Pareto entre o resultado desejado e os indesejados, que é a ideia principal do método *SkylineCF*. Embora os autores argumentem que a definição de preferências na geração de contrafactuais pode ser prejudicial quando o usuário não é um especialista do domínio, eles permitem a definição de *features* acionáveis (podem mudar), semi-acionáveis (só podem mudar em uma direção) e imutáveis.

O método *SkylineCF* é um dos poucos estudados que permite a aplicação em modelos de regressão sem a necessidade de adaptações externas, já que a componente de perda da função multi objetivos é genérica para tarefas de classificação e regressão. Outro ponto positivo importante da ferramenta proposta é a disponibilização de uma interface de consultas à fronteira construída, em linguagem similar a *SQL*, para reduzir o tempo necessário para a geração dos contrafactuais, que é uma das principais barreiras para a utilização desses métodos comercialmente. Com essa abordagem de consultas, a fronteira é computada uma única vez em relação à instância do usuário que recebeu uma classe indesejada, delimitando o espaço da classe desejada e processando as requisições em tempo quase imediato para retornar candidatos que satisfaçam as restrições impostas

O algoritmo *RF-OCSE*[5] é específico para modelos *Random Forest* e utiliza o paradigma guloso após converter a floresta em uma única árvore de decisão, extraindo conjuntos de decisão contrafactuais, que são as condições de divisão da árvore acumuladas até um nó folha que leve à classe desejada. Essas regras são ordenadas de acordo com a probabilidade de se alcançar a classe alvo da explicação, o algoritmo então extrai as regras ordenadas e as adiciona em um conjunto contrafactual até que a probabilidade acumulada ultrapasse um limiar. Enquanto métodos concorrentes geralmente entregam instâncias contrafactuais exatas, com um valor fixo, essa abordagem tem o objetivo de facilitar o entendimento e generalização das explicações, que por serem regras nos domínios das *features* permitem ao usuário selecionar um valor que seja mais

[5] https://github.com/rrunix/libfastcrf

acionável dado o contexto. Note que, por realizar operações individuais nas árvores de decisão que compõem a floresta, este método não necessita do modelo alvo para consultas como a maioria dos outros, mas sim da estrutura lógica para realizar operações sobre ela e garantir o funcionamento.

Algoritmos de **Força Bruta** são característicos por ter uma implementação tecnicamente mais simples e performar busca exaustiva no espaço de possibilidades, visitando todos os candidatos até determinar um que seja garantidamente ótimo como a solução. O método *FACET*[6] possui uma abordagem exaustiva similar à empregada no *RF-OCSE*, no sentido de ser específico para *ensembles* de árvores e iterar por todos os caminhos das árvores de decisão que compõem a floresta, mas por não empregar nenhuma estratégia gulosa posterior, os dois ficam em grupos diferentes. O funcionamento do *FACET* consiste em iterar pelos caminhos de todas as árvores de decisão, simplificando o conjunto de regras do caminho em hiper-retângulos delimitando os mínimos e máximos das *features* disponíveis. A partir de um *dataset* relevante ao problema, ele aumenta os dados com uma estratégia de ruído gaussiano e itera por todos eles obtendo a predição do ensemble, mapeando as decisões para os hiper-retângulos de cada árvore e combinando-os em único que será adicionado para um conjunto de regiões para indexação.

Assim como o *SkylineCF*, o *FACET* também disponibiliza uma interface para consultas similares a *SQL* como forma de agilizar a recuperação de contrafactuais, já que a geração precisa ser executada uma única vez. Esse processo de busca é baseado na análise da estrutura da consulta, indexação por vetores binários das regiões de hiper-retângulos construída na geração através do método *COREX*, também proposto pelos autores, e seleção das regiões mais próximas através de outro algoritmo, conhecidamente de força bruta, que é o *KNN*. Diferentemente do SkylineCF, este método implementa uma linguagem de consultas muito mais complexa e abrangente, sendo possível definir pesos diferentes em cada uma das *features* para o cálculo de distância na aplicação do KNN. Por ser específico para ensembles e analisar as árvores individuais, assim como o RF-OCSE, este método também precisa da estrutura lógica do modelo para funcionar. Outra limitação desta implementação é que para a extração dos hiper-retângulos funcionar, todas as features precisam estar em formato numérico.

[6] https://github.com/PeterVanNostrand/FACET

O método *CF-SHAP*[7] consiste em aplicar o algoritmo *KNN* no *dataset* relativo ao domínio do problema, que necessariamente precisa já estar classificado pelo modelo alvo da explicação, para obter uma população contrafactual, em relação à instância original da explicação, de tamanho "k". Essa população é então utilizada para calcular o valor de Shapley das *features*, marcando quais são as mais importantes para a obtenção da classe desejada. A população também é utilizada para a extração da média de valores e construção de tendências derivadas da instância original, marcando as features como acima ou abaixo da média para guiar a construção de contrafactuais. O retorno é composto pelos valores de Shapley e o vetor de tendências, sendo responsabilidade do usuário construir uma instância contrafactual válida a ser passada posteriormente ao modelo. Este método é específico para modelos de árvores devido à escolha de implementação, que utiliza um método de obtenção do SHAP específico para árvores.

O algoritmo FCE[8] torna a interação com o usuário parte obrigatória do processo de explicação, utilizando as restrições e preferências para delimitar o espaço de geração de contrafactuais, que é feita amostrando a instância original aleatoriamente através de perturbações, garantindo que todas as novas instâncias geradas estejam dentro do espaço. O método então itera por toda a população gerada, obtendo a classificação do modelo e aplicando-a em uma função de otimização, que leva em consideração se a instância é contrafactual e o quão distante ela está da instância original. Por fim, o único contrafactual retornado pelo método é aquele que minimiza a função de otimização

Encerrando os métodos de força bruta está o *COVISATEE*, que é a única ferramenta estudada que é totalmente dedicada à explicação de imagens, sendo que as imagens devem estar anotadas quanto às características, por exemplo, para *datasets* de pássaros, informar que a classe Pica-Pau possui a característica "cabeça vermelha". Ele funciona extraindo *features* das imagens através de uma *ResNet* e projetando os vetores de *features* em um espaço n-dimensional através de um modelo *Structured Joint Embedding* (*SJE*). Com essa etapa de pré-processamento finalizada, o método localiza a imagem original no espaço de *embeddings* e aplica um KNN com "k" igual a 1 para encontrar uma outra imagem que seja da mesma classe, para apresentação como explicação factual. Já a explicação contrafactual é obtida realizando perturbações na imagem original através do método *Iterative Fast Gradient Sign Method* (*IFGSM*), garantindo que a perturbação esteja

[7] https://github.com/jpmorganchase/cf-shap
[8] https://github.com/msnizami/FCE (Link quebrado em 20/06/2025)

próxima o suficiente da original, até que a classe predita seja diferente da inicial. A *ResNet* é aplicada na perturbação para projeção no espaço de *embedding* pelo *SJE*, dando início a outra aplicação de *KNN* com "k" igual a 1 para encontrar uma imagem do *dataset* original que seja um exemplo contrafactual.

Os algoritmos de **Programação Inteira (IP)** são baseados em proposições matemáticas de otimização para encontrar soluções ótimas para problemas constituídos, geralmente, apenas por variáveis inteiras. O método *ARLIC*[9] é capaz de lidar com features de todos os tipos, tendo uma estratégia própria para adequá-las à função de otimização, que consiste em garantir que a instância gerada esteja na classe contrafactual e que possui custo mínimo de alteração em relação à distribuição de valores das *features* no *dataset* informado. A otimização desta função é feita uma quantidade de vezes informada por hiperparâmetro com solucionadores padrão de problemas de IP, como o *CPLEX* que segundo os autores leva por volta de 0.1 segundo para finalizar. Ao fim da otimização, a solução é comparada em relação à instância original e a diferença, chamada de "ação", é adicionada a um conjunto que é posteriormente filtrado de forma a permanecer somente as ações que respeitem as restrições informadas pelo usuário, antes de realizar o retorno final. Os autores construíram a função de otimização utilizando o vetor de pesos do modelo de classificação linear, o que torna este método similar ao *FCE* e ao *FACET* analisados anteriormente, no sentido de necessitar da estrutura lógica do modelo para funcionar.

Já o método *LEWIS* propõe pontuações de explicação para auxiliar na produção de resultados, que podem ser contrafactuais para uma instância local, contextuais quando a população geral é segmentada de acordo com as *features* e as pontuações são exibidas somente para elas, ou globais quando apresentam as pontuações de todas as *features*. Uma das pontuações é chamada de *necessity score*, que mede a importância de determinado valor de uma feature para que a classe predita seja a desejada, outra é o *sufficiency score*, que mede se aumentar os valores da feature levam a uma mudança para a classe desejada. Existe também a *necessity-sufficiency score*, que é uma junção das duas anteriores, medindo a importância geral das alterações da feature na mudança de classe predita. Os autores não especificam qual solucionador de IP é utilizado, mas informam que a geração de candidatos leva em consideração apenas o *sufficiency score*, garantindo que todos estejam acima de um limiar informado como hiperparâmetro. A função de

[9] https://github.com/ustunb/actionable-recourse

otimização é modelada como a distância do valor proposto para cada uma das *features* em relação ao valor original, sendo que somente as *features* informadas como acionáveis pelo usuário podem ser modificadas. Os usuários também podem definir a forma de cálculo da distância.

Como principais limitações do *LEWIS*, pode-se citar que, por ser um método baseado em programação inteira, ele espera que todas as variáveis estejam transformadas para atender domínios discretos e finitos. Além disso, para obter pontuações de explicação mais robustas e corretamente interpretadas, o método também assume a disponibilidade de um modelo probabilístico causal (*PCM*) para o classificador ou regressor que se deseja explicar. Embora exista a possibilidade de cálculo das pontuações e geração das explicações sem esse recurso, os autores argumentam que os resultados podem se beneficiar bastante caso ele esteja disponível. Em cenários em que se tem o modelo à disposição mas não o *PCM*, métodos de descoberta causal como o proposto no artigo descartado anteriormente na seção de resultados "*Counterfactual Explanations of Black-box Machine Learning Models using Causal Discovery with Applications to Credit Rating*" podem ser empregados. Embora haja menção dos autores no artigo que o método pode ser usado para tarefas de regressão e classificação multiclasse, eles argumentam uma necessidade de alteração no cálculo das pontuações de explicação, o que não configura uma disponibilidade imediata dessas features, levando à não marcação dessas colunas na tabela.

O paradigma de **Algoritmos Genéticos** é utilizado para problemas de otimização, seguindo o processo de seleção natural onde os candidatos mais fortes, de acordo com alguma função de otimização, trocam informações estruturais uns com os outros, incorporando certa estocasticidade, para formar a população da próxima geração. A estrutura geral de todo algoritmo que segue este paradigma consiste em inicializar a população inicial seguindo alguma estratégia, geralmente aleatória, avaliar os candidatos pela função de otimização, selecionar os melhores, combinar as informações dos selecionados, aplicar mutação aleatória nos resultados e substituir a população anterior pelos candidatos produzidos. Este processo é repetido até que uma quantidade parametrizada de gerações seja atingida, ou algum outro critério de parada seja alcançado, como a estabilização dos resultados da função de otimização, para retornar o melhor candidato encontrado durante as gerações.

O método *LORE*[10] consiste na criação de um dataset artificial baseado em vizinhos próximos da instância original para treinamento de um modelo substituto de árvore de decisão, que será utilizado para gerar explicações factuais e contrafactuais. A construção do dataset artificial é feita aplicando algoritmo genético em duas etapas, a primeira com uma função de otimização que busca instâncias próximas da original e com a mesma classe predita que ela, a segunda para instâncias também próximas, mas dessa vez obtendo a classe desejada pelo modelo preditor. Durante o processo de mutação, o método leva em consideração o *dataset* original fornecido, para produzir valores para as features que estejam de acordo com a distribuição padrão dos dados do domínio do problema. Ao final de cada execução, ao invés de retornar apenas o melhor candidato, os algoritmos fornecem toda a população resultante, que é combinada para formar o *dataset* artificial. Todas as instâncias da população construída são enviadas ao modelo original para classificação, esses dados rotulados são utilizados para o treinamento da árvore de decisão. Esse modelo substituto é explorado de forma similar à feita nos métodos *RF-OCSE* e *FACET*, de forma que o caminho percorrido para a instância original é retornado como regra factual de explicação, enquanto todos os caminhos que levem à classe desejada são avaliados para retornar aqueles cujas regras levem a menor quantidade de *features* alteradas.

O método *LORESa*[11] surge como uma extensão direta do *LORE*, sendo proposto pelos mesmos autores[12] como forma de endereçar as principais limitações da ferramenta anterior, que são a falta de estabilidade, devido ao uso de uma única árvore de decisão como modelo substituto, e a falta de acionabilidade por não existir interação com o usuário. Para tornar as explicações estáveis, o processo de geração dos *datasets* artificiais por algoritmo genético, combinação dos dados, rotulação das instâncias e treinamento da árvore de decisão como modelo substituto é repetido uma determinada quantidade de vezes. Ao fim de cada iteração, a árvore de decisão construída é adicionada a um conjunto para combinação final em uma única árvore, garantindo maior estabilidade das decisões em relação a variações nos dados de entrada. Para garantir a acionabilidade dos contrafactuais, o método permite a interação com o usuário através de uma linguagem lógica para definir quais *features* podem ser modificadas e o intervalo de valores permitido para as modificações.

[10] https://github.com/riccotti/LORE
[11] https://github.com/francescanaretto/LORE_sa
[12] Com exceção de Francesca Naretto, que participou apenas da construção do método *LORESa*

A obtenção das regras factuais e do conjunto de regras contrafactuais de explicação é feita da mesma forma que no método *LORE*, porém, enquanto as regras factuais permanecem inalteradas ao fim, as contrafactuais passam por um processo de filtragem baseado nas restrições impostas pelo usuário através da linguagem lógica disponível. Note que, enquanto *LORE* precisa de acesso ao *dataset* original de treinamento do modelo, *LOREsa* descarta essa necessidade, mas em troca precisa de uma base de conhecimento "*K*" informando o domínio, média, variância, distribuição de probabilidade, entre outras métricas sobre cada uma das *features* disponíveis. Na ausência do *dataset*, esse conjunto de informações pode ser gerado por um especialista do domínio do problema, sendo importante para a geração de candidatos do *dataset* artificial que sejam condizentes com dados reais. Embora os autores afirmem a capacidade do método de trabalhar com explicações de imagens e textos, realizando até mesmo experimentos com esses formatos de dados, é necessário que o usuário realize a conversão para o formato tabular, o que não configura uma disponibilidade imediata desta funcionalidade.

A implementação do *GeCo*[13] introduz uma série de alterações no algoritmo genético através da proposição de ferramentas auxiliares, que são utilizadas para tornar as o processo mais ágil e as explicações mais coerentes e acionáveis. A primeira é um banco de dados "*D*" que contém o *dataset* de treinamento e agrupamento de features correlacionadas, que pode ser feito por um especialista de domínio, para garantir a coerência dos contrafactuais gerados com a estrutura original dos dados. A segunda é a linguagem lógica "*PLAF*", que permite ao usuário definir não apenas as *features* que podem ser modificadas e seus intervalos, mas também estruturas condicionais básicas onde uma feature é alterada sempre que outra também for, por exemplo, sentidos de alteração, como somente aumentar a idade, e passos de incremento ou decremento dos valores. A última é a representação Δ, que somente armazena os valores das instâncias geradas que são diferentes da original, como uma forma de economizar memória durante as operações do algoritmo genético. Essa representação também é benéfica para modelos de árvore, onde é possível realizar uma avaliação parcial somente para as *features* alteradas na instância, ao invés de percorrer todo o caminho a partir da raiz.

O método GeCo inicia fazendo a adequação das restrições impostas pela linguagem *PLAF* com valores reais, utilizando-a para definir o espaço factível através dos dados observados no *dataset*

[13] https://github.com/mjschleich/GeCo.jl

"*D*", que somente são coletados para construir a distribuição de valores quando respeitam às restrições. A obtenção da população inicial para o algoritmo genético é feita com sucessivas mutações da instância original, que são representadas no formato Δ e incluem também uma representação em *bitset* para facilitar a identificação de quais foram alteradas, e rotulação com o modelo original. Outra alteração importante no fluxo padrão dos algoritmos genéticos está na seleção de instâncias para o cruzamento, que é feita obtendo pares de *bitsets* que indiquem alterações em *features* diferentes, selecionando os melhores candidatos relativos aos *bitsets* escolhidos de acordo com a função de otimização. Além disso, durante as etapas de cruzamento e mutação, a linguagem *PLAF* é aplicada novamente para garantir que as instâncias geradas respeitem as restrições impostas, alterando os valores em *features* não concordantes. A condição de parada do algoritmo genético busca estabilidade na geração de contrafactuais, estando condicionada a que duas gerações seguidas construam somente instâncias da classe desejada.

Diferentemente dos demais, o funcionamento do método *MAGECE* depende também da forma de construção do modelo original, propondo técnicas para a de seleção de *features* e a normalização dos intervalos como forma de diminuir o espaço de busca do algoritmo genético, tornando a geração de contrafactuais mais rápida e esparsa. Os autores descrevem um primeiro processo de filtro das features utilizando uma *random forest* treinada com os dados originais, que será explorada calculando a entropia, ou Índice de Gini, das condicionais propostas para ranquear a impureza, que é alta quando os intervalos de valores estudados levam a duas classes diferentes. As *features* que tenham redução mais significativa da média de impureza nos nós condicionais da floresta são selecionadas, sendo filtradas novamente através do cálculo do Coeficiente de Spearman para aplicação de um teste de hipótese, que é utilizado para selecionar as *features* que tenham correlação com o alvo estatisticamente relevante. Elas passam então por processo de discretização das variáveis categóricas em intervalos numéricos, que são normalizados entre [0, 1], e são utilizadas na construção, treinamento e validação cruzada do modelo a ser explicado.

A geração dos contrafactuais consiste apenas na aplicação de um algoritmo genético, com função de otimização baseada na probabilidade do modelo atribuir a classe desejada e na distância da instância original. Embora a formulação da função objetivo considere propriedades como validade, garantindo que a instância é de fato contrafactual, e similaridade, garantindo uma distância máxima da instância original, ela deixa a desejar na geração de candidatos factíveis.

Para resolver isso, os mesmos autores[14] do artigo do *MAGECE* propõe o método *MACEMP*[15], que permite interação com o usuário para definir quais *features* podem ou não ser modificadas, como forma de incluir a propriedade de acionabilidade na função de otimização, garantindo que somente *features* acionáveis são modificadas. O *MACEMP* também inclui a propriedade de esparsidade na otimização do problema, garantindo que a diferença absoluta entre os valores de feature das instâncias construídas pelo algoritmo genético e a instância original não seja muito grande. Por fim, o *MACEMP* adiciona a propriedade de plausibilidade à função de otimização, como forma de garantir que os valores propostos para as instâncias geradas estão dentro da distribuição dos dados reais fornecidos ao método.

Finalizando os algoritmos genéticos está a ferramenta *CARE*[16], que utiliza uma implementação pronta do algoritmo *Nondominated Sorting Genetic Algorithm III* (*NSGA-III*) para gerar explicações que atendam a múltiplas propriedades, assim como o *MACEMP*. A principal contribuição dos autores com essa ferramenta é a formalização das propriedades necessárias para que um contrafactual seja de boa qualidade, implementando um fluxo que consiste em duas fases: *fitting* e *explaining*. Na primeira fase são construídas ferramentas auxiliares com base no *dataset* original informado, como modelos substitutos *Local Outlier Factor* (*LOF*) para cada uma das classes, no caso de tarefas de classificação multiclasse, ou intervalos de valores, no caso de tarefas de regressão. Além disso, também é construído um modelo de agrupamento *HDBSCAN* para cada uma das classes ou intervalos, como forma de identificar os vizinhos das instâncias geradas pelo algoritmo. O método também constrói um modelo de correlação de features baseado no *dataset* original, identificando sequências de modificações que devem ser respeitadas. Todas essas ferramentas são utilizadas para calcular funções de objetivo nos módulos *validity*, *soundness*, *coherency* e *actionability* que combinam diversas propriedades necessárias para os contrafactuais. A etapa seguinte, chamada "*explaining*", consiste em aplicar o algoritmo genético NSGA-III para gerar candidatos a contrafactual, seguindo as funções de otimização definidas, com base em uma instância original. O retorno de explicações ao usuário permite a priorização entre os diferentes módulos utilizados, aumentando a personalização do método.

---

[14] Com exceção de Hans Vandierendonck, que é autor somente do artigo do *MAGECE*

[15] Método proposto no artigo "*Counterfactual Explanations With Multiple Properties in Credit Scoring*" citado na seção de resultados como não incluído na análise. Assim como o *MAGECE*, os autores não nomearam o método, a decisão de nomeá-lo foi feita para facilitar as referências no decorrer do texto

[16] https://github.com/peymanrasouli/CARE

Algoritmos baseados em otimização por **Enxame de Partículas** (*PSO*), assim como os genéticos, são metaheurísticas utilizadas para explorar um espaço de busca com o objetivo de encontrar um candidato que possa ser considerado ótimo segundo algum critério. Eles possuem estrutura geral similar, contando com a inicialização das partículas e as respectivas velocidades através de alguma estratégia, geralmente aleatória, atualização da posição da partícula considerando a melhor posição obtida por ela e a melhor posição global e, por último, a finalização de acordo com algum critério, geralmente número de iterações excedido. Para realizar a atualização do melhor ponto da partícula, é considerada uma função de otimização a ser minimizada que é, no geral, o principal ponto de alteração entre as diferentes implementações dos algoritmos, já que ela varia de acordo com o problema a ser solucionado.

O primeiro método de *PSO* analisado é o *RASCEN-PSO*, que emprega a técnica de nicho nas partículas como forma de garantir maior diversidade nas explicações, ao invés de diversas instâncias que convergem essencialmente para a melhor partícula global (*gbest*). Os autores realizam experimentos com as técnicas *Species-based PSO* (*SPSO*), *Equilibrium SPSO* (*E-SPSO*), *Ring Topology* e *Fitness based on Euclidean distance Ratio PSO* (*FER-PSO*), constatando ao fim que a técnica de nicho *FER-PSO* obteve melhor resultado para o método. Para aplicar o nicho, o algoritmo itera pelos resultados imediatos do enxame de partículas calculando o *FER* entre a melhor posição (*pbest*) da partícula atual da iteração e o *pbest* de todas as outras partículas, escolhendo aquela que maximize o resultado como um ponto melhor local (*lbest*). A etapa descrita anteriormente é performada logo antes da atualização da posição atual das partículas, que tem a fórmula alterada para levar em consideração não somente o *pbest* e o *gbest*, como é o padrão de algoritmos *PSO*, mas também o *lbest* recém obtido.

Ao fim de um número determinado de iterações, o método *RASCEN-PSO* itera pelo *pbest* de todas as partículas da população final verificando, para cada uma das *features*, se as alterações feitas em relação à instância original a ser explicada é significativa (maior que 1%) e armazena essa informação em um *bitset*, marcando 1 para cada feature alterada significativamente. Para cada um dos *bitsets* únicos formados, o algoritmo itera pelas partículas relacionadas a ele selecionando aquela que possua o melhor *pbest* de acordo com a função de otimização, que consiste no candidato ser de fato contrafactual e com baixa distância da instância original, e adicionando-a a um conjunto que será retornado ao usuário. A principal limitação deste método

está no requisito de que os dados tabulares fornecidos a ele estejam todos discretizados em intervalos numéricos, que devem ser normalizados entre [0, 1]. Como ele faz consultas frequentes ao modelo para cálculo da função de otimização, caso o modelo não tenha sido treinado com os dados discretizados e normalizados, é feita uma conversão antes da chamada para que tudo funcione corretamente.

Note que na tabela construída os métodos *CF-PSO & CF-DE*[17] estão citados na mesma linha, já que ambos foram desenvolvidos pelos mesmos autores no mesmo artigo como uma forma de comparar as técnicas de *PSO* e evolução diferencial (*DE*) na geração de contrafactuais. Nas análises finais, os autores constataram que o *CF-PSO* obteve melhor resultado, focando a conclusão do artigo principalmente nele. Além disso, ao explorar a implementação de código aberto disponibilizada no artigo, foi constatado que a implementação do *CF-DE* consiste basicamente na invocação da função "*differential_evolution*" do pacote *Scipy*. Considerando esses fatores, foi optado por descartar a análise do CF-DE.

O método CF-PSO é bastante similar ao RASCEN-PSO, com a principal diferença sendo que o CF-PSO não emprega técnica de nicho para garantir a diversidade do conjunto final. Apesar dessa diferença, a função de otimização utilizada para determinar o *pbest* e *gbest* também leva em consideração somente a distância da instância original e o fato de o candidato ser ou não classificado pelo modelo com a classe desejada. Além disso, o CF-PSO também exige que os dados sejam discretizados em intervalos numéricos e normalizados entre [0, 1].

Finalizando as técnicas baseadas em metaheurísticas está a **Busca Tabu**, que assim como os demais é utilizado para encontrar uma solução ótima em um espaço de busca, de acordo com uma função de otimização. A principal característica deste paradigma consiste na utilização de uma memória para armazenar o histórico de alterações feitas na instância durante a exploração do espaço de busca, definindo uma quantidade de iterações máximas em que elas permanecem guardadas. Essa memória é implementada como uma fila de capacidade máxima, sendo chamada de "lista tabu", no sentido de as alterações presentes nela serem fortemente desencorajadas como uma forma de incentivar a diversidade das soluções geradas. O único momento em que uma alteração da lista tabu pode ser utilizada é quando o critério de aspiração, que é definido

[17] https://github.com/HaydenAndersen/ECCounterfactuals

geralmente como um melhor resultado na função de otimização do candidato atual em relação aos demais, é atingido. A estrutura geral de algoritmos baseados neste paradigma é a inicialização da lista tabu e seleção da solução inicial, seguida pela geração de vizinhos seguindo alguma estratégia, avaliação dos candidatos de acordo com uma função de otimização, filtragem da vizinhança de acordo com a lista tabu e atualização da memória e da melhor solução encontrada.

O método *CFNOW*[18], por se propor a lidar com tipos de dados além de tabulares, inclui uma etapa de pré-processamento para imagens, que consiste na segmentação em superpixels pelo método *quickshift* para codificação binária, e textos, que consiste em também criar uma representação binária para presença ou ausência de cada palavra. A primeira etapa do método é a de busca, que consiste gerar ações de modificação candidatas na instância original fornecida considerando a lista tabu, selecionar a modificação que tem a maior probabilidade de levar a predição para a classe alvo, incluí-la na lista tabu, modificar a instância original e verificar se a alteração levou à classe desejada. Esta etapa se repete até que as modificações acumuladas na instância original levem a um contrafactual, ou até que um número máximo de iterações seja alcançado.

A segunda etapa é executada apenas se a etapa anterior fornecer um contrafactual válido, caso contrário é retornada uma falha para o usuário. Ela consiste na melhoria da explicação gerada de acordo com a função de otimização, que consiste na distância do candidato em relação à instância original, mas pode ser alterada para outras, como quantidade de *features* modificadas. Caso o candidato represente uma melhoria em relação à instância gerada com a melhor pontuação na função de otimização, ela é armazenada para ser retornada posteriormente. Após isso, o algoritmo atua de forma similar à primeira etapa, gerando possíveis modificações, removendo-as de acordo com a lista tabu e selecionando a modificação que leve à maior probabilidade de mudança de classe, desta vez para a classe predita na instância original. Caso a regressão da instância modificada à classe original ocorra, a modificação escolhida é incluída na lista tabu e a primeira etapa recomeça, caso contrário, o processo da segunda etapa continua ocorrendo até que um limite seja alcançado.

[18] https://github.com/rmazzine/CFNOW

Algoritmos baseados em **Heurística** são produzidos para encontrar soluções satisfatórias para problemas complexos em uma quantidade de tempo razoável, abrindo mão da solução ótima. Eles são frequentemente empregados em problemas NP-Difíceis, como o de encontrar um conjunto de correção mínima (*MCS*) em problemas de satisfatibilidade (*SAT*), que é o objetivo do método *ASTERYX*. O primeiro passo deste algoritmo é utilizar o *dataset* fornecido como insumo para a criação de um modelo substituto *random forest*, que tem as árvores de decisão constituintes exploradas para criar regras booleanas na forma normal conjuntiva (*CNF*). Vale ressaltar que é necessário um processo de binarização das *features* antes do fornecimento do *dataset* ao algoritmo, ou do fornecimento da instância original para explicação.

Após a codificação na *CNF*, o *ASTERYX* extrai os conjuntos de cláusulas que representem uma razão mínima para a decisão do modelo para a instância original, através de uma formulação do estado como um problema de subconjunto mínimo insatisfazível (*MUS*), configurando uma explicação factual. Já a explicação contrafactual é feita extraindo os conjuntos mínimos de cláusulas que devem ter o resultado invertido para a instância original de forma a modificar a predição do modelo, através da formulação do estado como um problema de *MCS*, que é resolvido aplicando a ferramenta heurística *EnumELSRMRCache*. Os conjuntos gerados como solução do *MCS* e *MUS* são ordenados e selecionados de acordo com métricas que avaliam o tamanho da explicação, generalidade de acordo com os vizinhos da instância original e responsabilidade da explicação no resultado obtido. Além dessas explicações, chamadas de simbólicas pelos autores, o método também fornece explicações baseadas em pontuação das *features* dentro dos conjuntos factual e contrafactual, como a presença nas diversas regras, generalidade na vizinhança e responsabilidade nas predições.

Durante o processo de confecção da tabela, foi considerado o caso geral da ferramenta, que foi o explicado anteriormente, envolvendo um modelo substituto para a construção da *CNF* e das explicações. Nesse caso, o modelo original não é utilizado, mas um *dataset* relativo ao problema é necessário para a construção de um modelo substituto. No entanto, caso o usuário disponha de um modelo, tenha acesso à estrutura lógica dele e esse modelo possa ser convertido para *CNF*, como é o caso de *random forest*, *Binarized Neural Networks* (*BNN*), *Naive and Latent-Tree Bayesian Networks*, entre outros, ele pode ser utilizado no lugar do substituto e a necessidade do *dataset* é descartada.

A técnica de ***Branch and Bound*** (*BnB*) também é empregada para resolver problemas NP-Difíceis mas, diferentemente de algoritmos heurísticos, ela garante atingir a solução ótima ao fim da execução. Ela utiliza busca sistemática no espaço através da construção de ramificações em uma árvore, que passam por um processo de estimativa do melhor resultado obtenível, podando buscas mais profundas na ramificação criada caso a estimativa seja inferior ao melhor resultado já obtido. O método *MAPOCAM*[19], que utiliza a técnica *BnB*, define dois conjuntos para auxiliar no processo de poda da árvore de busca, o primeiro sendo um conjunto de *features* já alteradas "*F*" e o segundo sendo o conjunto de contrafactuais já produzidos durante a busca "*C*". Para guiar o processo de ramificação, ele também disponibiliza uma interface para o usuário definir quais *features* podem ser modificadas, o valor mínimo e máximo delas, sentido de alteração (somente incrementos, ou somente decrementos) e o passo dos incrementos e decrementos.

A construção das ramificações é feita selecionando *features* definidas pelo usuário como alteráveis, mas que ainda não foram modificadas, através do conjunto "*F*", utilizando o fator de importância[20] de cada uma delas de forma decrescente como parte dos critérios de seleção. A ramificação é feita iterando pelos valores possíveis de acordo com o intervalo e passos de incremento e decremento definidos pelo usuário, incluindo a feature selecionada em "*F*" e realizando uma série de verificações de poda sobre o candidato antes de iniciar a recursão. A poda da ramificação ocorre quando o tamanho de "*F*" excede um limiar pré-definido, ou quando o candidato produzido é dominado por algum dos contrafactuais já consolidados em "*C*". Essa verificação de dominação é feita de forma similar à empregada em problemas de Fronteira de Pareto, como o *SkylineCF*, onde um candidato é dominado por outro caso todas as alterações feitas tenham magnitude maior ou igual às do adversário, com exceção de ao menos uma alteração que é necessariamente maior.

Além de fornecer o modelo de classificação para consultas, o usuário também deve informar o método se o modelo possui estrutura de predições monotônica, onde ao aumentar o valor das *features* de entrada, a saída sempre aumenta ou diminui. Não é necessário que os modelos sejam monotônicos para o método funcionar, mas quando eles são, o processo de poda se torna mais

[19] https://github.com/visual-ds/cfmining

[20] Os autores reforçam a flexibilidade na forma de seleção das features, mas esta técnica foi a observada na implementação.

completo e o tempo de execução do algoritmo pode diminuir significativamente. A propriedade de monotonicidade é explorada fixando o valor das *features* presentes em "*F*" e definindo todas as outras marcadas como modificáveis pelo usuário para o maior valor possível. Essa instância produzida é predita pelo modelo e, caso a classe de resultado não seja a desejada, a ramificação é descartada, já que seguindo a propriedade de monotonicidade do modelo, nenhuma variação com menor magnitude de valores nas *features* seria capaz de alterar a predição.

A cada nova instância construída pelo algoritmo na exploração do espaço, o modelo de classificação é invocado para verificar se a classe predita é a desejada, incluindo o candidato no conjunto "*C*" em caso afirmativo antes de seguir com o funcionamento. Buscando alcançar otimalidade nos contrafactuais propostos, sempre que uma nova instância é inserida em "*C*" ocorre uma verificação de dominância, onde tanto o candidato atual quanto os demais membros de "C" são excluídos caso sejam dominados por alguma outra solução. O método precisa do dataset de treinamento do modelo para identificar a distribuição de valores, mínimos e máximos das *features* disponíveis, que são usados quando o usuário não informa nenhuma restrição.

Por fim, algoritmos de **Aprendizado por Reforço** buscam construir um agente autônomo para navegar em um ambiente, cujo estado é codificado em um vetor *features*, através de tentativa e erro, seguindo uma função de valor definida de acordo com o domínio do problema, que informa as estimativas de recompensa ao tomar determinadas ações nos estados. O aprendizado do agente é armazenado em uma política, que fornece informações de qual ação tomar quando o ambiente se encontra em determinado estado. Quando o espaço de exploração é muito grande, é computacionalmente custoso codificar todos os estados possíveis em representações de *features*, surgindo assim métodos por aproximação de função, que generalizam os estados similares e fornecem funções de valor aplicadas a eles para estimar recompensas. No caso de aprendizado por reforço profundo, as aproximações são feitas por redes neurais profundas, que na abordagem ator-crítico são divididas entre uma rede para aprender generalizações para a política (ator) e outra para a função de valor (crítico).

O método *SINAR-RL*[21] utiliza aprendizado por reforço profundo com a abordagem ator-crítico para aprender pares de *feature* e valor alvo que devem ser modificadas em uma instância, como

[21] https://github.com/unitn-sml/syn-interventions-algorithmic-recourse

forma alterar o resultado de predição do modelo de classificação. Para isso, ele utiliza uma rede neural para aprender a política gulosa da escolha de *features* para modificação a partir do estado atual informado, em conjunto com uma outra rede neural que aprende a política gulosa para determinar o melhor valor que a feature escolhida deve assumir. As duas redes neurais da política (ator) são gulosas e do tipo feedforward, assim como a rede neural da função de valor, cujas recompensas estimadas levam em consideração não apenas se a instância proposta passa para a classe desejada pelo modelo preditor, mas também a quantidade de *features* modificadas para penalizar grandes alterações. O aprendizado das políticas também está sujeito a um componente *Long Short Term Memory* (*LSTM*), que garante maior coerência das ações sugeridas pelo agente ao fim do treinamento, evitando repetições nas alterações de *feature*.

As políticas aprendidas são utilizadas em uma *Monte Carlo Tree Search* (*MCTS*), que tenta encontrar intervenções bem sucedidas nas instâncias originais do dataset de interesse do domínio fornecido, minimizando o custo das alterações. A busca leva em consideração a estimativa de valor do estado-ação aprendida pela rede neural, em conjunto com uma função para balanceamento de exploração e explotação e uma função de perda que modela o custo das alterações nas *features* disponíveis a partir de um grafo causal. O grafo causal é construído pelo usuário no início do processo de geração, definindo alterações em *features* que devem preceder outras, como um aumento em nível educacional antes de um aumento de renda. Quanto mais predecessores uma *feature* tiver, maior o custo (ou perda) da alteração nela. Quando a política aprendida sugere a finalização da intervenção através de uma ação especial "*STOP*", a instância é avaliada pelo modelo de predição e. Caso o resultado seja da classe desejada, a simulação é adicionada a um *buffer* de *replay*, que é utilizado na próxima iteração de treinamento do agente de aprendizado por reforço.

Com as iterações de treinamento finalizadas, as políticas aprendidas são usadas uma última vez na *MCTS* para gerar intervenções bem sucedidas, seguindo os mesmos direcionamentos do treinamento, e salvando as sequências que levaram aos contrafactuais no *buffer*. Todas as ações disponíveis, incluindo a especial "*STOP*" já mencionada e outra especial "*INTERVENE*" para marcar o ponto inicial comum, são transformadas em nós de um grafo. Para criar as arestas desse grafo, o algoritmo itera por cada uma das as intervenções armazenadas no *buffer*, considerando que todas elas têm estados iniciais que partem de "*INTERVENE*", formando arestas para as ações

tomadas de acordo com a sequência da intervenção. A cada novo nó visitado, o estado atual da intervenção é armazenado em conjunto com a ação de modificação de *feature* e valor a ser assumido, que são definidos a partir dele. Esses dados são utilizados para criar árvores de decisão em cada um dos nós, considerando os estados guardados como dados de entrada e a ação tomada em conjunto com o valor a ser assumido como o alvos da classificação.

Ao guardar as árvores de decisão em um grafo os autores realizam a sintetização das políticas aprendidas em uma estrutura navegável e explicável, que é capaz de fornecer explicações generalizando as decisões para estados ainda não vistos. Para construir explicações para qualquer contrafactual, basta iniciar o caminho no grafo construído a partir do nó "*INTERVENE*" e acumular cada uma das ações propostas ao longo das árvores de decisão, até que o nó "*STOP*" seja visitado. Nesse ponto, o *SINAR-RL* é bastante parecido com o *SkylineCF* e o *FACET*, já que ele realiza um grande processamento inicial que permite que consultas sejam feitas em tempo significativamente menor para qualquer instância que se deseja explicar.

Conclusão

A metodologia empregada para a revisão de literatura já é bastante consolidada em outros trabalhos similares, consistindo basicamente na coleta de artigos, filtragem, análises quantitativas, análises qualitativas e conclusão. No entanto, no melhor dos meus conhecimentos, a utilização de inteligência artificial (IA) como ferramenta para auxiliar o processo de filtro das publicações ainda é pouco explorada. Ainda que os resultados do *Notebook LM* tenham incluído quatro falsos positivos, a falta de discernimento de contexto exato da ferramenta parece contribuir para uma priorização da cobertura ao invés da precisão. Como contribuição futura, é possível construir *prompts* com mais informações sobre as regras a serem seguidas pela IA para a classificação, além de realizar uma análise mais extensa em relação às diferentes decisões tomadas para os artigos.

Durante a análise bibliométrica foi possível identificar uma tendência de crescimento na área de explicação de IAs desde 2019, em especial para métodos contrafactuais a partir de 2020, além de concluir sobre os principais pesquisadores do campo e as principais instituições que contribuem com as pesquisas. As análises de colaboração e de comunidade também revelaram

comportamentos ocultos sobre como alguns dos principais autores se relacionam entre si, contribuindo inclusive antecipadamente para a verificação de instituições. Durante as análises voltadas para as instituições, foram identificadas dois grandes grupos colaborando entre si, sendo que um deles é caracterizado por publicar artigos somente entre instituições de ensino, enquanto o outro é mais flexível e também inclui empresas nas publicações.

Ao analisar os artigos selecionados qualitativamente, foi constatado que o processo de se encontrar explicações contrafactuais é modelado principalmente como um problema de otimização, de forma a encontrar a instância com menor número de alterações em relação à original que leve ao resultado desejado. Para isso, são empregados uma grande variedade de métodos, desde implementações relativamente mais simples utilizando o paradigma guloso ou de força bruta, até propostas mais complexas que utilizam aprendizado por reforço profundo. No geral, as metaheurísticas são os principais paradigmas implementados representando mais de 40% dos métodos estudados, entre algoritmos genéticos, enxames de partículas, evolução diferencial e busca tabu.

Com relação ao formato de dados, quase 90% dos métodos estão preparados para receber apenas dados tabulares, mesmo considerando algumas restrições, como suporte apenas a dados numéricos ou binários, que foram discutidas durante as apresentações das respectivas ferramentas. No geral, os algoritmos propostos pelos autores tendem a ser genéricos, funcionando de forma agnóstica ao modelo de decisões em mais de 80% da base, sendo que os demais possuem alguma justificativa de implementação para serem específicos. Já a capacidade de interagir com o usuário, para formar explicações que sejam mais adequadas às realidades de cada um, ainda é pouco explorada pelos métodos, estando presente em apenas metade das obras estudadas.

De forma geral, os objetivos propostos pelo presente trabalho foram cumpridos integralmente, fornecendo uma análise robusta da área de explicações de IA, além de compilar os dados qualitativos incluindo as principais características e limitações dos métodos estado da arte em uma tabela. Como extensões desta revisão, pode-se citar a realização de benchmarks entre os diferentes métodos de código aberto analisados, buscando identificar características como velocidade na geração de contrafactuais, além de propriedades das explicações como esparsidade

e acionabilidade. Além disso, pode-se confeccionar uma outra tabela destacando todas as propriedades garantidas por cada um dos métodos nas explicações fornecidas, como forma de guiar ainda mais a seleção das ferramentas de acordo com as necessidades do usuário.

Referências Bibliográficas

ABDUL, A. et al. **COGAM: Measuring and Moderating Cognitive Load in Machine Learning Model Explanations**. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–14, 23 abr. 2020.

ALBINI, E. et al. **Counterfactual Shapley Additive Explanations**. FAccT '22: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, pp. 1054 - 1070, 20 jun. 2022.

ANDERSEN, H. et al. **Evolving Counterfactual Explanations with Particle Swarm Optimization and Differential Evolution**. 2022 IEEE Congress on Evolutionary Computation (CEC), pp. 01–08, 18 jul. 2022.

ANDERSEN, H. et al. **Producing Diverse Rashomon Sets of Counterfactual Explanations with Niching Particle Swarm Optimization Algorithms**. Proceedings of the Genetic and Evolutionary Computation Conference, pp. 393–401 12 jul. 2023.

APLEY, D. W.; ZHU, J. **Visualizing the effects of predictor variables in black box supervised learning models**. Journal of the Royal Statistical Society: Series B (Statistical Methodology), v. 82, n. 4, pp. 1059–1086, 11 jun. 2020.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. Brasília, 2018.

DASTILE, X.; CELIK, T.; VANDIERENDONCK, H. **Model-agnostic Counterfactual Explanations in Credit Scoring**. IEEE Access, v. 10, pp. 69543–69554, 25 mai. 2022.

DASTILE, X.; CELIK, T. **Counterfactual Explanations With Multiple Properties in Credit Scoring**. IEEE Access, v. 12, p. 110713–110728, 2024.

Denning, P. et al. **Computing as a discipline**. Commun. ACM 32, 1 (Jan. 1989), pp. 9–23.

European Union. **General Data Protection Regulation**. Official Journal of the European Union. 4 mai. 2016.

FERNÁNDEZ, R. R. et al. **Random forest explainability using counterfactual sets**. Information Fusion, v. 63, p. 196–207, nov. 2020.

Goldstein, Alex et al. **Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation**. Journal of Computational and Graphical Statistics 24 (2013), pp. 44 - 65.

GUIDOTTI, R. et al. **Factual and Counterfactual Explanations for Black Box Decision Making**. IEEE Intelligent Systems, v. 34, n. 6, pp. 14–23, 3 dez. 2019.

GUIDOTTI, R. et al. **Stable and actionable explanations of black-box models through factual and counterfactual rules**. Data Mining and Knowledge Discovery, v. 38, pp. 2825–2862 14 nov. 2022.

GRÉGOIRE, E.; IZZA, Y.; LAGNIEZ, J.-M. **Boosting MCSes Enumeration**. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18).

MAULUD, D.; ABDULAZEEZ, A. M. **A Review on Linear Regression Comprehensive in Machine Learning**. Journal of Applied Science and Technology Trends, [S. l.], v. 1, n. 2, pp. 140–147, 2020.

MAZZINE, R.; SÖRENSEN, K.; MARTENS, D. **A model-agnostic and data-independent tabu search algorithm to generate counterfactuals for tabular, image, and text data**. European journal of operational research, 1 ago. 2023.

MOLNAR, C. **Interpretable machine learning: a guide for making Black Box Models interpretable**. Morisville, North Carolina: Lulu, 2019.

MOORE, J.; NILS HAMMERLA; WATKINS, C. **Explaining Deep Learning Models with Constrained Adversarial Examples**. PRICAI 2019: Trends in Artificial Intelligence, p. 43–56, 23 ago. 2019.

PEREIRA, V.; BASILIO, M. P.; CARLOS, S. **pyBibX -- A Python Library for Bibliometric and Scientometric Analysis Powered with Artificial Intelligence Tools**. 27 abr. 2023.

RAIMUNDO, M. M.; NONATO, L. G.; POCO, J. **Mining Pareto-optimal counterfactual antecedents with a branch-and-bound model-agnostic algorithm**. Data Mining and Knowledge Discovery, v. 38, n. 5, pp. 2942–2974, 16 dez. 2022.

RASOULI, P.; CHIEH YU, I. **CARE: coherent actionable recourse based on sound counterfactual explanations**. International Journal of Data Science and Analytics, 30 set. 2022.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. **Why should I trust you? Explaining the Predictions of Any Classifier**. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16, pp. 1135–1144, 2016.

ROKACH, L.; MAIMON, O. **Decision Trees**. In: Data Mining and Knowledge Discovery Handbook. [s.l: s.n.], pp. 165–192, 2005.

RYMA BOUMAZOUZA et al. **ASTERYX: A model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations**. The 30th ACM International Conference on Information and Knowledge Management, Nov 2021, Virtual Event Queensland Australia, Australia, pp.120-129.

SADAF GULSHAD; SMEULDERS, A. **Explaining with Counter Visual Attributes and Examples**. ICMR '20: Proceedings of the 2020 International Conference on Multimedia Retrieval, pp. 35 - 43, 8 jun. 2020.

SAINYAM GALHOTRA; PRADHAN, R.; SALIMI, B. **Explaining Black-Box Algorithms Using Probabilistic Contrastive Counterfactuals**. SIGMOD '21: Proceedings of the 2021 International Conference on Management of Data, pp. 577 - 590, 9 jun. 2021.

SCHLEICH, M. et al. **GeCo: Quality Counterfactual Explanations in Real Time**. Proceedings of the VLDB Endowment, v. 14, n. 9, pp. 1681–1693, 1 mai. 2021.

SHEKHAR, S. et al. **A PSO Based Method to Generate Actionable Counterfactuals for High Dimensional Data**. 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pp. 1–8, dez. 2023.

SOHNS, J.; GARTH, C.; LEITTE, H. **Decision Boundary Visualization for Counterfactual Reasoning**. Computer Graphics Forum, v. 42, n. 1, p. 7–20, 20 set. 2022.

SOVRANO, F.; VITALI, F.; PALMIRANI, M. **Making Things Explainable vs Explaining: Requirements and Challenges Under the GDPR**. Lecture notes in computer science, pp. 169–182, 1 jan. 2021.

SUFFIAN, M. et al. **FCE: Feedback Based Counterfactual Explanations for Explainable AI**. IEEE Access, v. 10, pp. 72363–72372, 7 jul. 2022.

TAKAHASHI, D.; SHIMIZU, S.; TANAKA, T. **Counterfactual Explanations of Black-box Machine Learning Models using Causal Discovery with Applications to Credit Rating**. 2024 International Joint Conference on Neural Networks (IJCNN), 5 fev. 2024.

DE TONI, G.; LEPRI, B.; PASSERINI, A. **Synthesizing explainable counterfactual policies for algorithmic recourse with program synthesis**. Machine Learning, v. 112, pp. 1389–1409, 2 fev. 2023.

USTUN, B.; SPANGHER, A.; LIU, Y. **Actionable Recourse in Linear Classification**. FAT* '19: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 10 - 19, 29 jan. 2019.

WANG, Y. et al. **The Skyline of Counterfactual Explanations for Machine Learning Decision Models**. CIKM '21: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2030–2039, 26 out. 2021.

ZHOU, Z.-H. **Machine Learning**. Singapore: Springer Singapore, 2021.