UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARIA DE LOURDES MAIA SILVA

ATIVANDO λ-JUSTIÇA: NÃO-DISCRIMINAÇÃO ALGORÍTMICA EM ÁRVORES DE DECISÃO

FORTALEZA

2022

MARIA DE LOURDES MAIA SILVA

ATIVANDO λ-JUSTIÇA: NÃO-DISCRIMINAÇÃO ALGORÍTMICA EM ÁRVORES DE DECISÃO

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de concentração: Banco de Dados.

Orientador: Prof. Dr. Javam Machado.

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Biblioteca Central do Campus do Pici Prof. Francisco José de Abreu Matos

S581a Silva, Maria de Lourdes Maia.

Ativando λ-justiça: não-discriminação algorítimica em árvores de decisão / Maria de Lourdes Maia Silva. – 2022.

73 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2022.

Orientação: Prof. Dr. Javam de Castro Machado.

1. Justiça. 2. Classificação. 3. Inteligência artificial. 4. Pós-processamento. 5. Árvore de decisão. I. Título.

CDD 005

MARIA DE LOURDES MAIA SILVA

ATIVANDO λ-JUSTIÇA: NÃO-DISCRIMINAÇÃO ALGORÍTMICA EM ÁRVORES DE DECISÃO

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de concentração: Banco de Dados.

Aprovada em: 22/04/2022.

BANCA EXAMINADORA

Prof. Dr. Javam Machado (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Cesar Lincoln Mattos
Universidade Federal do Ceará (UFC)

Prof. Dr. Victor Evangelista de Farias
Universidade Federal do Ceará (UFC)

Prof. Dr. Sérgio Lifschitz
Pontifícia Universidade Católica do Rio de Janeiro
(PUC-Rio)

À minha família e amigos, por acreditarem e investirem em mim.

AGRADECIMENTOS

Agradeço a toda a minha família por acreditar em mim e pelo incentivo constante na realização deste trabalho.

Agradeço ao meu orientador e a todos que contribuíram de alguma forma para a realização deste trabalho.

Agradeço a instituição de fomento, FUNCAP, que me financiou durante este período de pesquisa.

“Eventually we learn that there is no shortcut to success."

(MASON, 1990, p. 66)

RESUMO

Com o avanço tecnológico, várias entidades optam por usar modelos computacionais para classificar indivíduos, com o objetivo de negar ou conceder um benefício. Por exemplo, em concessões de empréstimo bancário, em que o banco classifica se uma pessoa é apta a receber um empréstimo ou não. Apesar de aplicações de Inteligência Artificial (Inteligência Artificial (IA)) serem úteis para tomadas de decisão, elas não são livres de discriminação. Quando um algoritmo é treinado com dados historicamente discriminatórios, ou a base de dados é desbalanceada no que diz respeito às características minoritárias, o modelo tende a propagar a discriminação presente nos dados de treinamento. Para classificar indivíduos semelhantes de forma semelhante, ou seja, rotular analogamente as pessoas com habilidades e características similares para a realização de uma tarefa, são necessárias restrições de justiça, que, por sua vez, podem alterar as classificações, prejudicando a acurácia do algoritmo. Neste trabalho, são definidas uma métrica para calcular o quão justo é um modelo e duas propriedades para mitigar o problema gerado pela propagação de discriminação de indivíduos ao mesmo tempo que lidam com o *trade-off* entre utilidade e justiça. As propriedades são ativadas na etapa de pós-processamento. Além disso, propomos a ativação das propriedades definidas para o modelo de Árvore de Decisão. Os resultados obtidos a partir da aplicação das propriedades de justiça propostas usando Árvore de Decisão atingiram altos níveis de utilidade e justiça.

Palavras-chave: justiça; classificação; inteligência artificial; pós-processamento; árvore de decisão.

ABSTRACT

With technological advancements, several entities use computational models to classify individuals to deny or grant a benefit. For example, the bank ranks whether a person can receive a loan in bank loan grants. Although Artificial Intelligence (AI) applications are helpful for decision-making, they are not free from discrimination. When an algorithm is trained with historically discriminatory data or the database is unbalanced concerning minority characteristics, the model tends to propagate the bias present in the training data. To classify similar individuals similarly, that is, to correspondingly label people with similar abilities and characteristics to perform a task, restrictions of fairness are necessary, which, in turn, can change the classifications, impairing accuracy. In this work, we define a metric to measure how fair is a model, and two properties to mitigate the problem generated by the propagation of discrimination of individuals while dealing with the trade-off between utility and fairness. A model achieves the properties in post-processing step. Furthermore, we propose activating the properties defined for the Decision Tree model. The results obtained from the application of the proposed justice properties using the Decision Tree reached high levels of utility and fairness.

Keywords: fairness; classification; artificial intelligence; post-processing; decision tree.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AUC	*Area Under the ROC Curve*
CART	*Classification and Regression Tree*
COMPAS	*Correctional Offender Management Profiling for Alternative Sanction*
FATT	*Fairness aware Training of Decision Trees*
IA	Inteligência Artificial
ML	*Machine Learning*

LISTA DE SÍMBOLOS

A	Atributo
c	Conjunto de classes nas amostras
d	Métrica de distância
D	Métrica de dissimilaridade
F	Variável que representa um conjunto de folhas
f_i	Folha i $\in F$
I	Variável que representa o conjunto de indivíduos
m	Quantidade de atributos necessários para classificação
M	Variável que representa o mapeamento de um indivíduo ao conjunto de possíveis saídas
o	Saída pertencente a O
O	Variável que representa o conjunto de saídas
P	Grupo protegido
p_i	Frequência relativa da classe i
R	Classificador
S	Conjunto de dados
T	Variável que representa uma árvore de decisão
U	Função de utilidade
V	Variável que representa um conjunto de indivíduos
x, y	Indivíduos pertencentes a I
κ, ρ	Marginais
λ	Limiar de justiça
δ	Variável de folga da restrição de *Lipschitz*
τ	Limitante do impacto díspar
σ	Distribuição de probabilidade
Γ	Conjunto de todas as distribuições conjuntas

SUMÁRIO

1 INTRODUÇÃO

O uso de Inteligência Artificial (IA) é cada vez mais comum por permitir a construção de sistemas que imitam a inteligência humana e têm a capacidade de executar tarefas, baseados na informação extraída dos dados coletados. *Machine Learning* (*Machine Learning* (ML)), ou "aprendizado de máquina", é uma subárea contida em IA, que visa a criação de sistemas que aprendem sobre os dados para melhorar o desempenho em tarefas que são atribuídas a eles. Com a ascensão do uso de sistemas para tomadas de decisão automatizadas, médias e grandes empresas utilizam modelos de aprendizado de máquina para rotular indivíduos que submetem seus dados para processos classificatórios. Ademais, devido ao uso de modelos computacionais, as classificações são feitas de forma mais rápida e com menor custo em comparação ao trabalho executado por humanos.

Existe um problema atrelado à tomada de decisões feita por aplicações: se as amostras de dados usadas para treinar o algoritmo são discriminatórias, o modelo automatizado propaga essa discriminação em suas classificações, ou seja, rotula um indivíduo injustamente, negando a ele o recebimento de um benefício. Outra configuração que pode ocasionar uma tendência nas classificações é o desbalanceamento dos dados. Se houver poucas amostras de determinadas características, a classificação tende a ser tendenciosa.

Em 2021, a Comissão da União Europeia propôs novas regulamentações sobre Inteligência Artificial, tratando os riscos inaceitáveis que representam ameaças à segurança, meios de subsistência e direitos, banindo sistemas e aplicativos que as desrespeitem ou manipulem o comportamento humano (European Commission, 2021). Antes da implantação dos sistemas, estes precisam cumprir algumas obrigações, por exemplo, certificar a alta qualidade dos conjuntos de dados que alimentam os sistemas e minimizar o risco de resultados discriminatórios, além de garantir altos níveis de precisão, segurança e robustez. A discriminação de indivíduos em classificações algorítmicas é considerada um risco inaceitável, visto que pode influenciar uma tomada de decisão acerca de uma pessoa devido a valores de atributos protegidos, isto é, atributos nos quais uma tomada de decisão não deve se basear, por exemplo, em gênero, raça, cor, orientação sexual, e religião dentre outros.

O problema de discriminação de indivíduos pode persistir mesmo quando atributos protegidos não estão explícitos no conjunto de dados. Particularidades acerca de pessoas podem ser inferidas a partir de atributos ou de combinações deles. O endereço de um indivíduo é um aspecto que é comumente utilizado na tomada de decisão, no entanto, pode revelar outras

características acerca desse sujeito. De acordo com o Departamento de Censo dos Estados Unidos, em abril de 2020, o condado de Greene, no estado do Alabama, tinha uma população de aproximadamente 7.730 cidadãos e 79,9% destes eram negros ou afro-americanos, 18,5% brancos e 1,6% possuíam raças miscigenadas ou outras raças; enquanto 52,8% da população era do gênero feminino e 47,2% masculino (CENSUS, 2021). Se os dados dos residentes deste condado forem utilizados em processos classificatórios, sem identificar qualquer valor de atributos protegidos, algumas características podem ser pressupostas devido às estatísticas demográficas do condado. Em Greene, há um grande desbalanceamento quanto à raça, que é um atributo protegido, portanto, uma entidade pode supor a raça de um indivíduo deste condado. Por este motivo, suprimir atributos protegidos não é suficiente para garantir que a tomada de decisão não seja tendenciosa.

Para mitigar o problema de discriminação algorítmica, o trabalho Dwork *et al.* (2012) definiu uma propriedade que garante não haver divergência na classificação de indivíduos semelhantes. Com o cumprimento dessa propriedade, pessoas que possuem as mesmas habilidades para uma tarefa são classificadas da mesma forma. Devido à recente criação de leis e regulamentos, é indubitável a necessidade da garantia de não-discriminação de indivíduos em processos classificatórios a partir de conjunto de dados.

Como passo inicial, é necessário modelar as recomendações e designações das normas para reverter a propagação da discriminação gerada por algoritmos de classificação. Após a modelagem matemática, o modelo deve garantir que as propriedades formuladas sejam satisfeitas, chamamos isso de ativação de justiça. Por outro lado, a garantia dessas propriedades prejudica a acurácia do resultado retornado pelo modelo, afetando a qualidade da classificação. Portanto, é necessário o desenvolvimento de técnicas que assegurem o balanceamento entre a acurácia e a justiça de um modelo, provendo utilidade na execução de uma tarefa e respeitando a ética e os regulamentos de não-discriminação de indivíduos.

A fim de conter a propagação da discriminação causada por algoritmos de classificação, pesquisadores vêm estudando formas de garantir propriedades de justiça. Além da abordagem proposta em Dwork *et al.* (2012), existe outra definição que prega a não-discriminação de grupos minoritários (ou grupos protegidos), cujo objetivo é balancear as classificações para os diferentes conjuntos de pessoas, de forma que não haja um grupo privilegiado ou prejudicado (DWORK *et al.*, 2012). Em Barocas *et al.* (2017), são definidas propriedades que ativam justiça para grupos em modelos de ML.

Existem três etapas pelas quais a justiça pode ser ativada em um algoritmo (PITOURA *et al.*, 2021). São elas:

- Pré-processamento: o algoritmo modifica a entrada de dados removendo a tendência de classificações injustas. A modificação pode ser, inclusive, a adição de dados sintéticos para aumentar a quantidade de pessoas com características que não aparecem com frequência, evitando que o modelo as classifique de forma tendenciosa.
- Durante o processamento: neste método, um modelo já existente é modificado para introduzir não-discriminação algorítmica, ou é criado um novo algoritmo. Uma abordagem pode aplicar justiça durante a fase de processamento através da construção de uma etapa no sistema que gera uma representação justa dos dados de entrada, enquanto aprende como classificar os indivíduos do conjunto. Outra forma de ativar justiça é formular e resolver um problema de otimização com restrições que mitigam o problema em questão.
- Pós-processamento: as classificações são modificadas sem alterar o funcionamento ou treinamento do modelo. Uma forma de ativar justiça nesta etapa é equilibrar a quantidade de classes entre os diferentes grupos de pessoas. No nosso caso, a justiça é ativada nesta etapa e equilibramos as classes entre pessoas com habilidades semelhantes para uma tarefa.

Um classificador justo visa rotular indivíduos aptos ou inaptos para a realização de uma tarefa de forma equivalente ou semelhante, podendo ativar propriedades de justiça através de técnicas na etapa de pré-processamento, durante o processamento ou no pós-processamento. Por consequência, atributos protegidos não devem ser relevantes na tomada de decisão, contanto que as pessoas possuam características semelhantes no que diz respeito ao objetivo da organização responsável pela classificação. Como explicitado anteriormente, geralmente atributos protegidos não são incluídos como um recurso para o modelo, no entanto, a combinação de outras características pode revelar o atributo em questão. A descoberta dos valores de atributos protegidos de cada indivíduo no banco de dados e a influência destes na classificação podem levar à inadequação do processo classificatório perante as leis de anti-discriminação contidas na Constituição Federal de 1988 e nas regulamentações de cada país ou região. Assim, como espécies de flores podem ser descobertas a partir de suas medidas de sépala e pétala (FISHER, 1936), seres humanos também podem ter seus atributos protegidos descobertos, a partir da combinação de informações como endereço, altura, idade e outros atributos não protegidos.

A Figura 1 ilustra como se comporta um classificador injusto. Amostras de dados são utilizadas como entrada e o classificador analisa as características das amostras, atribuindo

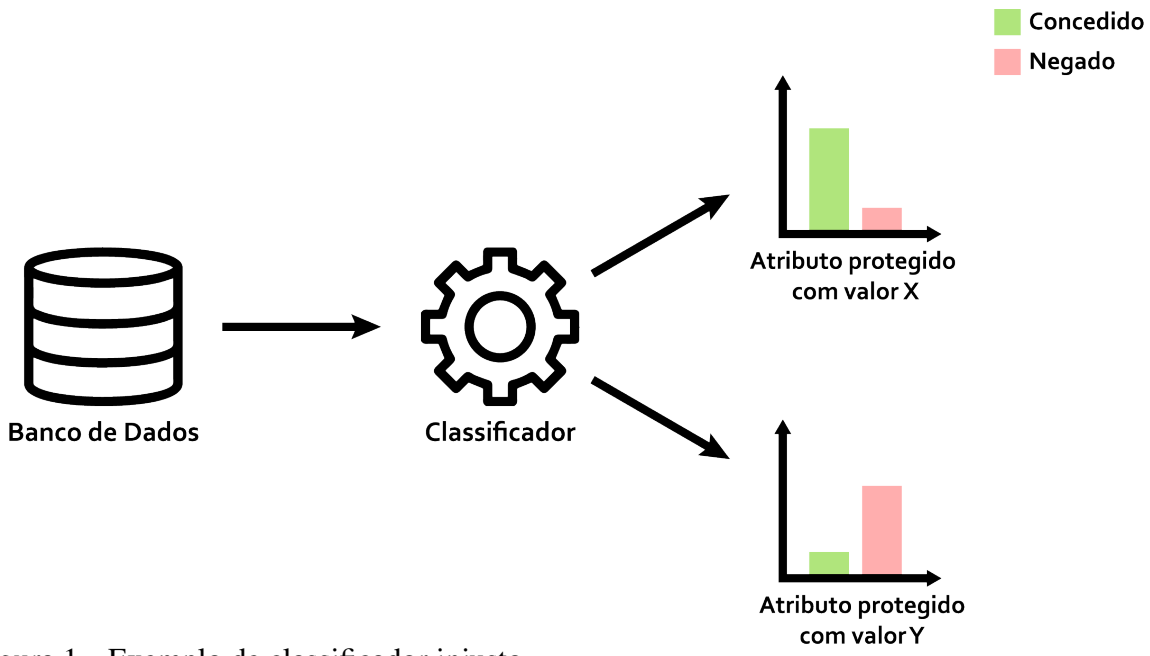


Figura 1 – Exemplo de classificador injusto.

as classes associadas a elas. Supondo um cenário de empréstimo bancário, em que a entidade concede ou nega um empréstimo a um indivíduo, é possível observar na ilustração da Figura 1 que pessoas com valor **X** no atributo protegido são mais prováveis de receber um empréstimo bancário do que pessoas com valor de atributo protegido igual a **Y**. Se houver poucas amostras de dados de pessoas com característica **Y**, o modelo pode classificar indivíduos com esse perfil tendenciosamente, visto que poucas amostras acerca de um grupo podem não ser suficientes para representar a informação real desse conjunto.

Em algumas regiões, existem aplicações de IA para predizer risco de reincidência criminal de pessoas, com base no histórico dos réus e em casos fichados e armazenados anteriormente no sistema criminal. Esses softwares auxiliam os juízes das cortes na tomada de decisão para fianças, medidas preventivas e tempo de prisão. O COMPAS (*Correctional Offender Management Profiling for Alternative Sanctions*) é um sistema usado na corte judiciária do condado de Broward, no estado da Flórida. O algoritmo do COMPAS classifica réus como pessoas com baixo, médio e alto risco de reincidência de crime, baseando-se em uma pontuação de valor inteiro entre 0 e 10, que é associada a um indivíduo. O artigo jornalístico ProPublica (2016b) explicita a tendência nas predições de IAs que são tendenciosas nas classificações de indivíduos. Os jornalistas do veículo *ProPublica*, Julia Angwin e Jeff Larson, foram responsáveis por uma pesquisa que mostra que, com o COMPAS, réus pretos eram mais prováveis de serem incorretamente classificados como de "Alto Risco" do que réus brancos. Analogamente, criminosos brancos são geralmente preditos como menos perigosos do que realmente são (PROPUBLICA,

Figura 2 – Riscos de reincidência criminal de dois indivíduos rotulados pelo COMPAS.

2016a). Há mudança de tratamento, inclusive quando os réus cometeram infrações semelhantes.

A Figura 2 ilustra dois homens que cometeram crimes e que foram rotulados pelo COMPAS, que classificou o risco de reincidência criminal para cada um deles. Sabendo que James Rivelli, a esquerda da imagem, é um homem branco, e Robert Cannon, a direita, um homem preto, ao comparar ambos é notória a discriminação algorítmica em relação às pessoas de cor preta. Apesar do histórico de Cannon ser consideravelmente mais ameno que o de Rivelli, ele foi classificado como médio risco com pontuação 6, já Rivelli foi rotulado como baixo risco com pontuação 3. Após o cumprimento da pena e soltura dos homens, foi averiguado que James reincidiu ao crime, executando um grande roubo, enquanto Robert não cometeu nenhuma infração subsequente. Analisando a predição, é fácil inferir que, se os delitos armazenadas no sistema fossem idênticos para ambos os indivíduos, os tratamentos e predições continuariam desiguais. Este é um dos vários exemplos de discriminação algorítmica, em que classificações tendenciosas geradas por modelos de aprendizado são aplicadas em cenários reais, prejudicando o julgamento de pessoas.

Árvores de Decisão são modelos de aprendizado de máquina que são amplamente utilizados por entidades a fim de classificar itens ou pessoas. Devido à facilidade de interpretação e por ser uma técnica ilustrável, esse modelo é bastante usado dentro de empresas, inclusive por bancos, que as usam para a avaliação de riscos em empréstimos. As árvores de decisão permitem a visualização de cada decisão e de múltiplos cenários, bem como a consequência deles. Para tarefas de classificação de indivíduos, caso a altura da árvore seja baixa o suficiente para humanos conseguirem facilmente identificar combinações de atributos, é simples identificar conjuntos que geram discriminação apenas olhando para a árvore.

Devido à semelhança com um fluxograma, funcionários de uma empresa podem

entender e apresentar informações extraídas dos dados sem precisar de conhecimento estatístico. Além disso, o preparo do conjunto de dados para ser utilizado como entrada do algoritmo exige menos esforço quando comparado a outras técnicas de tomada de decisão. Valores faltantes e *outliers* não geram tanto impacto no resultado gerado pela árvore de decisão. Com tudo isso, tem-se a justificativa da escolha do modelo para ativação de propriedades de justiça.

Além disso, técnicas de pós-processamento não precisam modificar a construção da árvore apenas as saídas, uma solução simples e funcional para o objetivo de alcançar justiça. Considerando também que as distribuições de frequência são armazenadas nas folhas da árvore, a satisfação de restrições de justiça que envolvem distribuições de probabilidade ou de frequência podem ser facilmente identificadas após a construção do modelo original. Então a garantia de justiça pode ser ativada preservando e aproveitando a etapa de construção de um modelo existente de ML.

Neste trabalho, são propostas definições que limitam a quantidade de justiça mínima a ser respeitada pelo modelo, além da construção de uma árvore de decisão que respeita as propriedades apresentadas. São listadas três contribuições no cenário de justiça para indivíduos, os quais passam por um processo de classificação:

- **(i)** métrica de justiça: calcula a taxa de justiça de um modelo dada uma entrada com **n** indivíduos representados por tuplas. A métrica criada leva em consideração a condição de Lipschitz usada para alcançar a propriedade bem definida de justiça para indivíduos em Dwork *et al.* (2012) e se adaptando melhor ao contexto geral da ativação da propriedade em algoritmos de ML;
- **(ii)** definição de λ-justiça: garantia que a taxa de justiça quantificada por **(i)** é de pelo menos λ; além da definição, é proposta a ativação dessa propriedade usando o modelo de árvore de decisão;
- **(iii)** definição de (λ, δ)-justiça: é uma relaxação de **(ii)**, especialmente para atender conjuntos de dados em que a propriedade de λ-justiça não é garantida, também é apresentada a ativação da propriedade usando árvore de decisão.

Atualmente, o tema de não-discriminação algorítmica tem extrema relevância devido ao crescente uso de tecnologias para tomadas de decisões, anteriormente executadas por humanos, e às criações de novas regulamentações. Existem inúmeros trabalhos na literatura visando resolver o problema. Há dois grandes conceitos bem definidos no que diz respeito à justiça: justiça para indivíduos ou *individual fairness* (DWORK *et al.*, 2012) e justiça para grupos ou *group*

fairness (DWORK *et al.*, 2012). A abordagem estabelecida neste trabalho é a primeira, cujo problema é voltado para o tratamento desigual de indivíduos semelhantes. O livro Barocas *et al.* (2017) explicita as preocupações éticas atuais da sociedade e aborda justiça em ML, bem como o presente trabalho, no entanto, no livro utiliza-se o conceito de *group fairness*.

Este trabalho investiga soluções para o problema da construção de modelos discriminatórios realizada por meio da técnica de árvore de decisão. Buscamos soluções na etapa de pós-processamento que minimizam a geração e propagação de discriminação em modelos classificatórios. Acreditamos que o ajuste do modelo nas etapas finais do processo de construção considerando parâmetros de não-discriminação algorítmica produzam resultados satisfatórios alcançando ambos os objetivos, ativar justiça enquanto mantém a utilidade da classificação.

Diferente dos trabalhos existentes na literatura, a abordagem proposta é parametrizada. As variáveis podem ser modificadas de acordo com as necessidades e prioridades da entidade responsável pela classificação, ainda garantindo a justiça entre os indivíduos a serem rotulados pelo modelo de ML. Além disso, executamos as técnicas propostas para árvores de decisão, explicitando o passo a passo da ativação de justiça.

O trabalho está organizado da seguinte forma: o Capítulo **??** faz uma revisão da literatura relacionada ao tema, principalmente no que diz respeito a justiça para indivíduos, em seguida são apresentados conceitos já existentes sobre temas necessários para o entendimento das propriedades e aplicações do trabalho no Capítulo **??**. As métricas, incluindo a métrica de justiça proposta, tecnologias e bases de dados utilizadas na aplicação e experimentos da abordagem proposta são descritas no Capítulo **??**. Adiante, os Capítulos 5 e 6 definem as propriedades λ-justiça e (λ, δ)-justiça, respectivamente, também aplicando-as sobre o modelo de Árvore de Decisão, cujos resultados obtidos são mostrados no Capítulo **??**. Por fim, o Capítulo **??** contém as considerações finais sobre o trabalho.

1.1 Objetivos

1.1.1 Objetivos gerais

- Finalidade: definir e aplicar propriedades que mitiguem o problema de discriminação algorítmica;
- Avaliação: comparar os resultados alcançados com trabalhos existentes na literatura.

1.1.2 Objetivos específicos

- Estabelecer uma métrica que calcule o quão discriminatório é um modelo de classificação dada uma entrada S considerando conceitos bem definidos da literatura e se adaptando a algoritmos de ML;
- Definir uma propriedade que garanta a não-discriminação algorítmica, enquanto assegura altos níveis de utilidade da aplicação;
- Selecionar conjuntos de dados para validar e avaliar a aplicação;
- Comparar o aumento dos níveis de utilidade e justiça após a ativação da propriedade de não-discriminação algorítmica definida.

2 TRABALHOS RELACIONADOS

Devido a abordagem de justiça ser um tema relativamente atual, não existem muitas obras que se assemelham a nossa, que ativa justiça para indivíduos no modelo de árvore de decisão durante a etapa de pós-processamento. Os trabalhos relacionados foram selecionados por proximidade com o assunto ressaltado nesta pesquisa, porém a única relação entre todos os trabalhos é a ativação de justiça para indivíduos. Os outros critérios para incluir um artigo neste capítulo foram: construções de métricas de similaridade, ativação de justiça no pós-processamento e algoritmos para tomada de decisão. O trabalho que mais se assemelha à nossa abordagem é descrito na Seção 2.1.

2.1 Treinamento justo de classificadores de árvore de decisão

No trabalho Ranzato *et al.* (2021), os autores ativam o conceito de justiça para indivíduos em conjuntos de árvores de decisão. É proposto um método chamado FATT (*Fairness Aware Training of Decision Trees*), que utiliza interpretação abstrata dos dados (COUSOT; COUSOT, 1977) para obter restrições para a construção da árvore de decisão, além de usar um *framework* chamado Meta-Silvae para treinar o conjunto de árvores (RANZATO; ZANELLA, 2021). Meta-Silvae é uma árvore de regressão e classificação, e um conjunto de árvores Meta-Silvae é treinado através do modelo *Random Forest*. O critério de divisão da árvore é a seleção de um atributo candidato baseado na pontuação de estabilidade, que se refere ao quão justo é o conjunto originado da divisão.

A verificação do cumprimento da restrição de justiça para indivíduos em um conjunto de árvores é feita por meio de um algoritmo que checa se as propriedades de estabilidade do classificador são satisfeitas. O Meta-Silvae visa maximizar uma função objetivo, que é a soma ponderada das métricas de utilidade e justiça (ou estabilidade). A construção de um conjunto de árvores é gerada por um algoritmo genético que produz uma população de árvores, ranqueadas com base na utilidade e estabilidade.

Assim, FATT é um modelo útil que é construído basicamente por:

- Populações de árvores que evoluem a cada geração, produzidas por meta-heurística;
- Treinamento por *Random Forest*.

Na geração final de árvores, o algoritmo retorna a(s) melhor(es). Para lidar com atributos categóricos, os autores optaram por usar *one-hot-encoding* e instanciam o classificador

em um domínio abstrato. Os conjuntos de dados utilizados nos experimentos são muito utilizados na literatura de justiça: *Adult Income*, COMPAS, *Crime and Communities*, *German Credit Risk* e *Health*. Os resultados alcançados pelo FATT foram comparados com o algoritmo de Árvore de Decisão CART (*Classification and Regression Tree*). FATT obteve valores próximos ou superiores em questão de acurácia e justiça.

A maior diferença entre o FATT e o modelo construído neste trabalho é a parametrização que é dada como entrada pela entidade responsável pelas classificações. Os parâmetros servem para limitar a quantidade mínima de justiça que o modelo deve cumprir, com base nos interesses da entidade respeitando as regulamentações de não-discriminação.

2.2 Operacionalizando Justiça para Indivíduos com Representações Justas de Pares

No trabalho Lahoti *et al.* (2019a), os autores propõem uma operacionalização de justiça para indivíduos que não precisa de uma especificação humana acerca da métrica de distância. Essa abordagem coleta informações sobre indivíduos que sejam igualmente merecedores de um benefício, modelando esse conceito através da construção de um grafo de justiça, e aprende o que chamam de *Pairwise Fair Representation* (PFR), ou representação justa de pares. O trabalho constrói uma operacionalização de justiça, abrangendo inclusive cenários em que é extremamente difícil calcular a distância entre dois indivíduos de grupos diferentes, por exemplo, considerando circunstâncias onde existem políticas de ações afirmativas. É complicado, até mesmo para um humano, mensurar a diferença entre indivíduos de grupos diferentes ao mesmo tempo que cumpre ações afirmativas sem prejudicar nenhum deles.

Esta abordagem contém e executa dois pontos principais:

- Métrica de distância através do grafo de justiça;
- Aprender como representar um par de forma justa.

Os autores apresentaram duas formas de construir um grafo de justiça: (i) grafo para pessoas comparáveis, e (ii) para pessoas incomparáveis. A primeira abordagem compara indivíduos que são do mesmo grupo ou que não recebem ações afirmativas, por exemplo. Neste grafo, uma aresta conecta dois indivíduos se eles são similarmente qualificados para uma certa tarefa, ou quando são da mesma classe de equivalência do grupo ao qual pertencem. A forma (ii) constrói um grafo de justiça para indivíduos que são incomparáveis através da classificação de indivíduos em cada um dos grupos, em outras palavras, cada grupo irá possuir uma classificação de indivíduos baseados numa tarefa de decisão. Então, é construído um grafo que relaciona

indivíduos de grupos diferentes cujas pontuações da classificação pertencem ao mesmo quantil. Se houver dois grupos, o grafo gerado é um grafo bipartido, em que cada indivíduo só pode se relacionar com indivíduos de outro grupo.

É formulado um modelo de otimização para aprender como representar pares de indivíduos de forma justa. O objetivo é minimizar a diferença entre os indivíduos de diferentes grupos que foram classificados similarmente. O artigo representa essa diferença através da norma L_2. Os experimentos foram executados em dois dos conjuntos de dados base na literatura de justiça, sendo eles *Crimes and communities* e COMPAS, e a utilidade foi mensurada através da métrica *Area Under the ROC Curve* (AUC) (*Area Under the ROC Curve*). Os resultados mostraram que o modelo PFR produz benefícios sem necessitar de grande quantidade de julgamentos humanos para a construção do grafo justo e do aprendizado.

2.3 Mitigação de Tendência no Pós-processamento para Justiça de Grupos e Indivíduos

Em Lohia *et al.* (2019), os autores visam ativar tanto justiça para indivíduos quanto para grupos em um algoritmo na etapa de pós-processamento e para isso criam um algoritmo chamado IGD (*Individual+Group Debiasing*). A abordagem primeiro detecta as amostras cujas classificações são tendenciosas e as seleciona para uma mudança na classificação predita pelo modelo. A ideia da ativação de ambos os conceitos se baseia, principalmente, na métrica de impacto díspar (NARAYANAN, 2018), que é originada da literatura de justiça para grupos. A vantagem de usar essa métrica é que podemos trocar os rótulos de amostras de indivíduos pertencentes a coletividades que violam a propriedade de justiça para grupos.

Sendo assim, os indivíduos selecionados para terem o rótulo modificado são aqueles mais prováveis de violar a propriedade de justiça para indivíduos. Então, é proposto um detector de tendência na classificação de indivíduos. Dado um conjunto de saídas $O = \{0, 1\}$, a definição utilizada é que uma amostra é tendenciosa se o resultado predito for diferente quando um indivíduo *i* pertence a um grupo não-minoritário de quando ele pertence a um grupo minoritário. A quantidade de tendência é calculada pela diferença entre as pontuações para as respostas do classificador ao alterar o atributo protegido de um indivíduo.

O método para obter um detector de tendências é utilizar um conjunto de validação, que é uma partição do conjunto total, sem rótulos, e perturbar o valor do atributo protegido das amostras para que seja um valor pertencente ao conjunto não-privilegiado. Após essa etapa, é possível obter a quantidade de tendência presente nas amostras. Então, é construído um conjunto

de dados usado para treinar o detector de tendência. As amostras que possuem o maior valor de tendência têm um parâmetro de viés rotulado como 1 e as outras, como 0. Esse rótulo depende de um limitante τ que é baseado no impacto díspar e, assim, estabelece uma relação entre a justiça para indivíduos e para grupos.

Após a detecção de amostras com classificações tendenciosas, é calculada a tendência para cada amostra do grupo não-privilegiado; se for 1, a amostra retorna a saída que teria caso pertencesse a um grupo privilegiado. Os conjuntos de dados experimentais foram *Adult Income*, *German Credit Risk* e COMPAS, e os resultados mostram que a acurácia do detector de tendência é muito alta para atributos protegidos, como sexo e raça.

2.4 *iFair*: Aprendendo Representações de Dados Individualmente Justas para Tomadas de Decisão Algorítmicas

O trabalho Lahoti *et al.* (2019b) constrói uma aproximação chamada *iFair*, que, similar ao Lahoti *et al.* (2019a), também aprende como representar dados de forma justa, no entanto, sem o uso de grafos, e sim por meio de um modelo de otimização. O modelo proposto não considera nenhuma noção de justiça para grupos na função objetivo, nem necessita que seja dada uma entrada de atributos protegidos específicos. O modelo consiste em uma função objetivo que pondera utilidade e justiça para indivíduos, considerando restrições de justiça para grupos baseadas em regulamentações.

O modelo retorna os dados de entrada com representações que respeitem as propriedades de justiça. Um mapeamento satisfaz justiça para indivíduos se a diferença entre dois valores seja menor ou igual a um limitante. Sendo esses valores: a distância de dois mapeamentos de indivíduos para representações justas deles mesmos e a distância entre os indivíduos caso pertencessem a um conjunto não-protegido. Essa definição de justiça para indivíduos é similar ao trabalho apresentado na Seção 2.3. A perda de utilidade é quantificada pela soma dos erros médios quadráticos dos indivíduos na representação original e na representação justa, enquanto a medida de perda de justiça é quantificada pelo quadrado da diferença entre as distâncias de dois indivíduos nas representações originais e na justa.

O mapeamento justo é selecionado por meio do modelo de otimização, sendo aquele que minimiza a função objetivo. Esta função é calculada por um coeficiente multiplicado pela perda de utilidade, quantificada pelas duas matrizes que representam o conjunto de indivíduos, a matriz original e a representação justa, somado com outro coeficiente que multiplica a perda

de justiça. Ambos os coeficientes são hiper-parâmetros. Os resultados experimentais obtidos, também recebendo como entrada conjuntos de dados utilizados na literatura, mostram que justiça e acurácia podem ser conciliadas e garantem resultados consistentes.

2.5 Comparação entre os trabalhos

A Tabela 1 compara as técnicas criadas em cada um dos trabalhos relacionados e neste trabalho, que denotamos na tabela como FCART. Embora todos os artigos criam técnicas que ativam justiça a nível de indivíduo, apenas dois ativam justiça na etapa de pós-processamento.

Técnica	Justiça para indivíduos	Pós-processamento	Árvore de Decisão
FATT	✓	×	✓
PFR	✓	×	×
IGD	✓	✓	×
iFair	✓	×	×
FCART	✓	✓	✓

Tabela 1 – Tabela comparativa entre os trabalhos relacionados e este trabalho.

3 FUNDAMENTAÇÃO TEÓRICA

Antes de iniciar a fundamentação teórica e as propriedades propostas neste trabalho, existem alguns termos que serão muito utilizados ao longo da leitura. Para facilitar a compreensão, abaixo estão listados os termos e seus respectivos significados.

- **Mapeamento:** função que associa um valor de saída a uma entrada.
- **Rótulo:** a característica que o modelo tenta predizer com base nas observações dos valores de atributos dados como entrada, também pode ser chamado de classe.
- **Par de itens:** dupla de itens que pertencem a um conjunto de dados. Dado um conjunto $C = \{t_1, t_2, t_3\}$, os pares contidos em C são $(t_1, t_2), (t_1, t_3), (t_2, t_3)$. Um par de itens se caracteriza como (t_i, t_j), onde (t_i, t_j) e (t_j, t_i) são o mesmo par, dado que $i \neq j$.

3.1 Aprendizado de máquina e modelos de classificação

ML é o estudo do aprendizado de algoritmos que aprendem sobre os itens conforme a experiência obtida através do uso de dados mantêm a mesma semântica e objetivo (MITCHELL, 1997). Nessa área existem três grandes categorias: aprendizagem supervisionada, aprendizagem não supervisionada, e aprendizagem por reforço. No aprendizado supervisionado, o algoritmo recebe um conjunto de dados rotulados e aprende como classificar novas amostras por meio da observação do comportamento dos dados de treinamento, replicando o aprendizado humano. No aprendizado não-supervisionado os rótulos não são fornecidos e o algoritmo deve predizê-los encontrando padrões inicialmente desconhecidos, por exemplo, agrupando as amostras semelhantes e detectando anomalias. Já o aprendizado por reforço utiliza o conceito de recompensa, por meio da observação do ambiente, o algoritmo pode executar ações que determinam o estado de um agente. O objetivo é determinar o conjunto de ações que maximizam a recompensa.

Os modelos de classificação têm como propósito rotular um conjunto de amostras, associando um conjunto de observações feitas pelo algoritmo. Esses modelos são exemplos de aprendizado supervisionado e predizem os rótulos baseando-se na combinação de atributos de um objeto ou indivíduo. A Figura 3 mostra o esquema do funcionamento dos modelos de aprendizado supervisionado.

A ideia principal é obter informação usando um conjunto de treinamento, em que o modelo aprende quais combinações de características levam a um determinado resultado. A base de dados é dividida entre conjuntos de treino e teste. O primeiro conjunto é responsável pelo

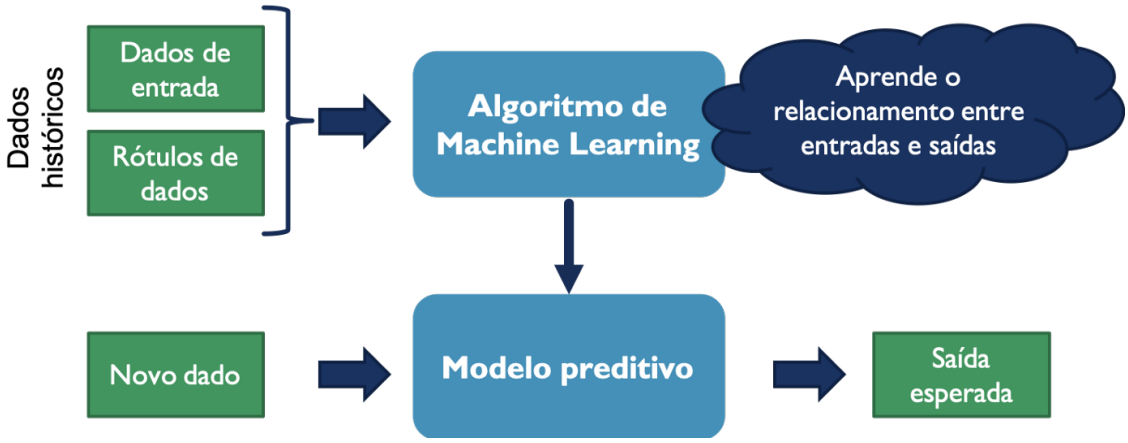


Figura 3 – Esquema do aprendizado supervisionado. Fonte: (ESCOVEDO, 2020).

treinamento do algoritmo e o segundo é usado para testar a capacidade de predição do modelo. A acurácia ou utilidade do modelo é medida por meio da comparação entre as classes preditas no conjunto de teste e os rótulos originais desse conjunto.

3.1.1 Árvore de Decisão

A árvore de decisão é um modelo de classificação que se assemelha à tomada de decisão humana, em que um conjunto de restrições é associado a um resultado. O termo "árvore" é usado, pois a estrutura do modelo é similar a de uma árvore invertida, como ilustra a Figura 4. Essa estrutura é um grafo contendo vértices e arestas, onde os vértices são representados pelos nós e as arestas são as ligações entre os nós. O nó raiz armazena todo o conjunto de dados de entrada, pois nele não existe nenhuma restrição. Os nós folhas, ou nós terminais, armazenam a decisão do modelo, contabilizando a quantidade de amostras que respeitam o conjunto de regras estabelecidas nos caminhos entre o nó raiz e os nós terminais. Nós que não são terminais ou raiz são chamados de nós internos. Cada um contém uma restrição que divide o conjunto anterior em dois ou mais subconjuntos. Os subconjuntos gerados são disjuntos. O rótulo com maior quantidade de amostras é escolhido para ser o majoritário. Sendo assim, qualquer nova amostra que respeite o conjunto de restrições de um caminho é classificada como a classe majoritária do nó folha.

A Figura 5 indica a relação entre os nós. Um nó é chamado de pai quando são gerados subconjuntos que obedecem alguma restrição. Esses subconjuntos são armazenados em nós conectados ao nó pai, chamados de filhos. A altura da árvore é determinada pela quantidade de níveis contidos nela e os níveis representam a geração do conjunto de nós. O nó raiz não tem nenhum pai e está no nível 0, os filhos do nó raiz estão no nível 1 e os filhos dos filhos estão no

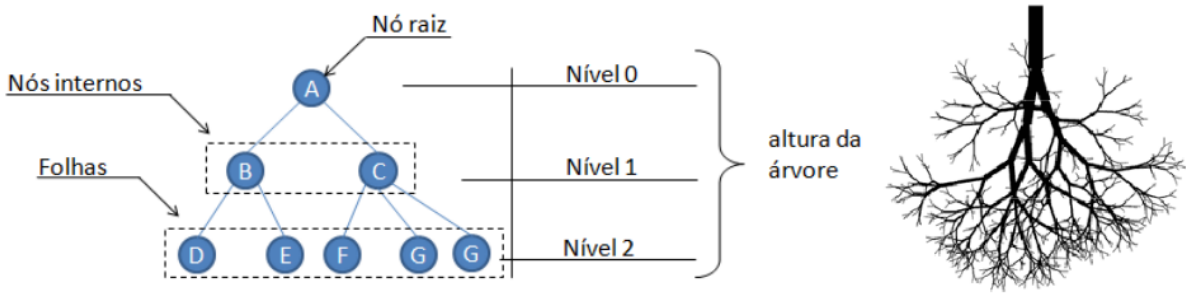


Figura 4 – Estrutura de uma árvore de decisão.

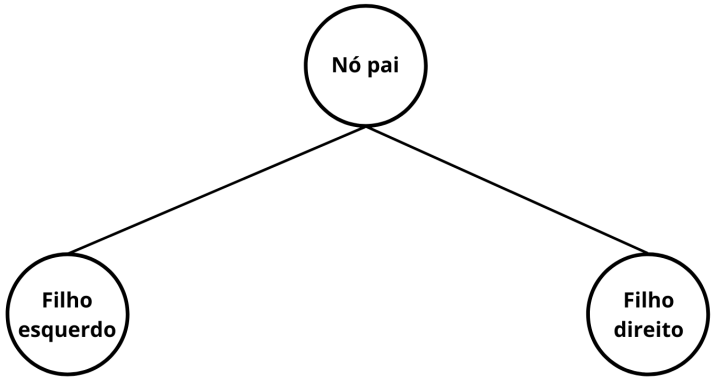


Figura 5 – Relação entre os nós.

nível 2, e assim sucessivamente. A geração final é alcançada quando todos os nós contidos nela não possuem nenhum filho.

Para a construção da árvore, é considerado um critério de divisão. A cada iteração do algoritmo é selecionado um atributo para ser dividido, baseando-se em quão puros são os nós, seguindo os critérios descritos a seguir. Os critérios mais conhecidos para executar essa função são o índice de Gini (CERIANI; VERME, 2012) e o ganho de informação (KULLBACK; LEIBLER, 1951; KULLBACK, 1997), que faz uso da entropia. Ambos os critérios se baseiam na proporção das classes dos subconjuntos. Seja S um conjunto de entrada para a qual o índice está sendo calculado, c o conjunto das classes presentes nas amostras de dados, e p_i a frequência relativa da classe i nas amostras de S, o índice de Gini é calculado por:

$$Gini(S) = 1 - \sum_{i \in c} p_i^2. \tag{3.1}$$

O índice pode apresentar um valor dentro do intervalo [0, 1]. Quanto mais próximo de 0, maior a pureza dos dados e, caso contrário, menor a pureza. O ganho de informação consiste em um conceito semelhante, no entanto, utiliza a função de entropia, cuja fórmula é apresentada abaixo.

$$Entropia(S) = \sum_{i \in c} -p_i log_2 p_i. \tag{3.2}$$

O ganho de informação é calculado sobre algum atributo A, e o resultado é a diferença entre a entropia de S e a entropia das partições geradas por A.

Existem muitos algoritmos de árvores de decisão, alguns performam melhor com atributos categóricos ou finitos e outros com atributos contínuos. Um algoritmo muito conhecido e utilizado é o ID3 (QUINLAN, 1986) que constrói uma árvore não-binária, ou seja, cada nó não-folha da árvore pode ter mais de dois filhos. ID3 é um algoritmo iterativo, cujo critério de divisão é o ganho de informação ou entropia. A cada iteração, o atributo A com menor valor de $Entropia(A)$ é selecionado para particionar o conjunto contido no nó, produzindo nós filhos que armazenam os subconjuntos gerados pelas partições. A recursão termina quando:

- a árvore atinge uma altura definida como parâmetro;
- não há mais atributos a serem selecionados, pois todos já foram particionados;
- todos os itens no subconjunto pertencem a mesma classe.

O ID3 é útil para dados categóricos, que têm uma quantidade limitada de possíveis valores. No entanto, não é recomendado para dados contínuos com muitos valores candidatos. Outra desvantagem do ID3 é que pode acontecer *overfitting*, ou seja, o modelo se adapta ao conjunto de treinamento que tem bons resultados apenas para aquele conjunto. O *overfitting* no algoritmo citado acontece principalmente quando a árvore é alta e particiona todos, ou quase todos, os atributos.

Outro algoritmo popularmente conhecido que implementa árvores de decisão é o CART (*Classification and Regression Tree*) (BREIMAN *et al.*, 1984), que constrói árvores binárias de classificação e regressão. Diferente do ID3, esse algoritmo se comporta bem quando as amostras de entrada possuem atributos contínuos. A função que é comumente utilizada como critério de divisão para esse algoritmo é o índice de Gini. Para cada nó, o algoritmo encontra um limiar caso o atributo selecionado para divisão seja contínuo, ou seja, o melhor ponto de partição, gerando dois subconjuntos. O filho esquerdo armazena as amostras de dados do nó pai cujo valor do atributo particionado é menor que o limiar, e o filho direito armazena o restante do conjunto. Quando o critério de parada é alcançado, a árvore é construída e os dados de entrada serão preditos de acordo com o comportamento observado nos dados de treinamento. A escolha do limiar é feita por meio da análise acerca do valor que gera partições mais puras, ou seja, com

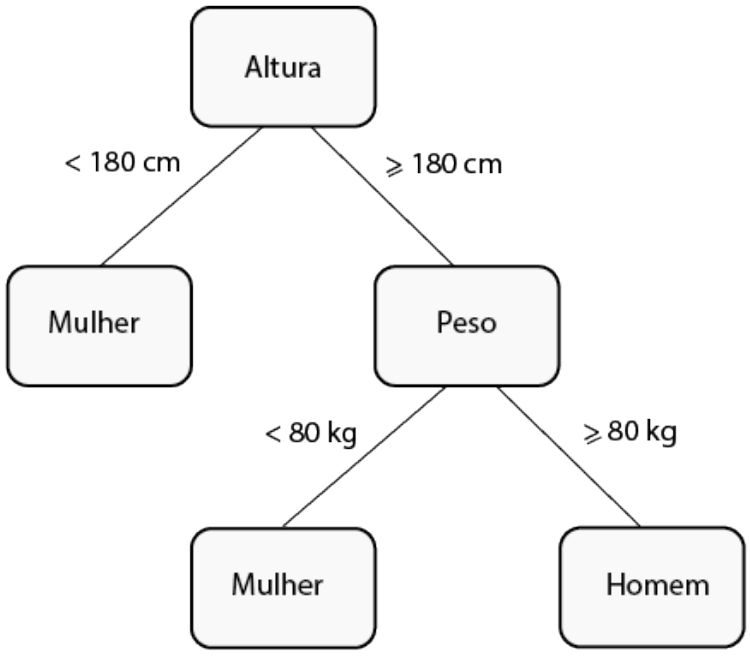


Figura 6 – Árvore CART, que rotula indivíduos como “Homem” ou “Mulher”.

menor valor do índice de Gini. O rótulo de cada tupla dada como entrada é o majoritário do nó folha ao final do caminho de restrições obedecidas pela amostra.

A Figura 6 ilustra um exemplo de árvore CART, em que o algoritmo classifica o gênero de um indivíduo baseando-se nos valores de altura e peso. Para cada um dos atributos particionados, é calculado o melhor limiar. No caso da altura, o limiar é “180 cm” e os indivíduos com valor inferior a esse são classificados como “Mulher”. Quando uma pessoa possui altura maior que o limiar, é necessário observar outra variável, o peso. Novamente, o melhor valor de atributo é selecionado para particionar o conjunto em dois, considerando agora o peso. Na subárvore direita, indivíduos com peso menor que 80 quilos são classificados como “Mulher”, enquanto aqueles com peso maior ou igual são rotulados como “Homem”.

3.2 Justiça para Indivíduos

Existem inúmeros termos em português usados para descrever a palavra inglesa *fairness*. Atualmente, essa é uma área extensamente estudada por pesquisadores, visto a necessidade do cumprimento ético de regulamentações que visam mitigar a mudança de tratamento para pessoas pertencentes a grupos minoritários. Algumas palavras adotadas como tradução são: equidade, justiça e não-discriminação. Neste trabalho, são adotados dois termos: não-discriminação algorítmica e justiça, pois o objetivo é garantir que o algoritmo não discrimine pessoas devido a valores de atributos protegidos, providenciando justiça na classificação dos indivíduos.

As soluções algorítmicas devem se assemelhar às decisões tomadas por humanos, no entanto, sabe-se que ainda hoje existem muitas pessoas que tendem a discriminar devido ao pensamento retrógrado de que indivíduos com característica X são superiores a indivíduos com característica diferente dessa. Por isso, deve ser pressuposto que classificações realizadas por humanos ou bases de dados usadas para o treinamento de algum algoritmo podem ser injustas.

O trabalho Dwork *et al.* (2012) descreve o conceito de justiça para indivíduos, demonstrando como alcançar tal propriedade por meio de uma restrição que limita o quão desigual pode ser a classificação de um par de indivíduos. A grande dificuldade de ativar justiça em modelos de classificação é manter a utilidade dos dados após predizê-los obedecendo restrições de não-discriminação. Para isso, é criado um modelo de otimização linear que visa maximizar a utilidade das classificações preditas pelo algoritmo enquanto atende às restrições.

A justiça baseada em indivíduos se resume a tratar indivíduos semelhantes, no que diz respeito a uma tarefa específica, de forma semelhante. Ou seja, pessoas com habilidades e características parecidas avaliadas para a concessão de um benefício devem ser classificadas de forma similar, evitando assim que algum atributo protegido influencie na tomada de decisão.

Chamamos a entidade responsável pela classificação de "fornecedora", a qual precisa atender os requisitos impostos pelas leis ou regulamentações da região cujos serviços da entidade serão locados. Levando isso em consideração, a métrica que provê o quão semelhantes são dois indivíduos pode ser imposta externamente por autoridades locais ou organizações de direitos humanos. Neste trabalho, consideramos o grau de semelhança de indivíduos para a realização de uma tarefa, sem considerar ações afirmativas, apenas a aptidão relacionada a um objetivo imposto pelo fornecedor.

Os três passos principais para alcançar o objetivo de justiça para indivíduos apresentados em Dwork *et al.* (2012) são:

- Escolha da métrica que computa a "distância" entre um par de pessoas;
- Definição do mapeamento que associa um indivíduo a uma distribuição de probabilidade σ;
- Formulação do problema de otimização que minimiza a perda de utilidade de um classificador.

Considerando V e O como o conjunto de indivíduos e o conjunto de saídas em um banco de dados S, respectivamente, é necessário definir uma métrica d que compute o quão semelhantes ou diferentes são um par de indivíduos $(x, y) \in V$, e outra métrica D que mede o

quão dissimilares são duas distribuições. Supondo o caso binário em que há duas possíveis saídas, $O = \{0,1\}$, um classificador aleatório gera mapeamentos M que associam um indivíduo $x \in V$ a uma distribuição de probabilidade σ sobre o conjunto de saídas.

Para classificar um indivíduo x, é necessário escolher uma saída $o \in O$ de acordo com a distribuição $M(x)$. Para garantir que as distribuições associadas a indivíduos semelhantes também sejam semelhantes, a restrição do mapeamento de Lipschitz deve ser respeitada.

A definição do mapeamento de Lipschitz diz que um mapeamento M satisfaz a propriedade (D,d)-Lipschitz quando para todo $x,y \in V$ a restrição abaixo é assegurada.

$$D(M(x),M(y)) \leq d(x,y) \tag{3.3}$$

Essa inequação é o conceito principal para ativação de justiça e pode ser resumida como a diferença entre os mapeamentos de dois indivíduos deve ser limitado pela distância entre os mesmos. Dessa forma, força-se que indivíduos com características idênticas para a realização de uma tarefa tenham classificações iguais. Para facilidade de entendimento, a inequação 3.3 será chamada de restrição de Lipschitz ao longo do tempo.

Para todo classificador, sempre vai existir um mapeamento que obedece a restrição de Lipschitz. Se todas as distribuições sobre O forem as mesmas para todos os indivíduos em V, a restrição 3.3 é satisfeita. No entanto, o ônus de classificar todos os indivíduos da mesma forma é que a utilidade dos dados é altamente prejudicada e influenciaria diretamente a decisão do fornecedor.

Supondo que uma empresa de *marketing* está recebendo currículos para empregar uma pessoa apta para exercer a função de publicitário, vamos considerar a Tabela 2 como o banco de dados com o perfil dos candidatos. O fornecedor deve rotular um indivíduo como “Classificado” ou “Classificável”. As formas mais simples para ativar a restrição de Lipschitz são (i) classificar todos os indivíduos como “Classificado” ou (ii) classificar todos como “Classificável”. No entanto, caso a opção (i) seja seguida, o fornecedor se prejudica, pois não tem recursos suficientes para manter os cinco candidatos na vaga. Se a opção (ii) fosse acolhida, a função requerida continuaria vazia, também prejudicando a empresa de *marketing*.

O mais lógico é avaliar as qualidades e habilidades de cada um dos candidatos para a vaga e ranqueá-los. Dependendo da quantidade de vagas oferecidas, os primeiros candidatos são selecionados e contemplados com o emprego. Pelo exemplo, os candidatos 2 e 5 são os mais indicados a vaga, levando em consideração a formação e o tempo de atuação na área.

id	Idade	Formação	Experiência
Candidato 1	23	Química Bacharelado	0 anos
Candidato 2	25	Publicidade e Propaganda	5 anos
Candidato 3	21	Publicidade e Propaganda	0 anos
Candidato 4	32	Letras	2 anos
Candidato 5	25	Publicidade e Propaganda	3 anos

Tabela 2 – Conjunto de dados dos candidatos para vaga de publicitário.

Como o fornecedor só abriu uma vaga, o candidato 2 seria o mais indicado. Se houvesse um candidato 6 com as mesmas características do candidato 2, ele também deveria ser contemplado com o emprego, portanto o fornecedor deveria abrir outra vaga para a função de publicitário, obedecendo a restrição de Lipschitz para esses dois indivíduos.

A diferença entre as métricas d e D é que a primeira é utilizada para medir a diferença entre dois indivíduos ou dois vetores, por exemplo, a distância L_1 (BLACK, 2019) ou a métrica L_∞ (CANTRELL, 2000). Já a métrica D computa a diferença entre duas distribuições sobre as possíveis saídas, $M(x)$ e $M(y)$, mapeadas para os indivíduo $x, y \in V$. Alguns exemplos de métricas que calculam a dissimilaridade entre duas distribuições são: distância estatística (ROYALL, 2004), a métrica relativa l_∞ (KÖTHE, 1983), e *Earthmover's distance* (MALLOWS, 1972). Ambas as métricas, d e D, devem estar na mesma escala, para que seja feita a comparação entre as distâncias no mapeamento M da restrição de Lipschitz.

Para solucionar o problema desencadeado pela ativação de justiça, (DWORK *et al.*, 2012) apresentou um modelo que maximiza a utilidade do classificador, enquanto considera a restrição de Lipschitz para todos os indivíduos em V. Esse modelo é resolvido utilizando programação linear. Para o desenvolvimento da solução do problema, é necessário que todas as métricas e variáveis sejam bem definidas no contexto do fornecedor.

Seja U a função de utilidade do classificador R dado um indivíduo x e uma saída o. O modelo de otimização segue a estrutura apresentada abaixo.

$$\max_{\{M(x)\}_{x\in V}} \frac{1}{|V|} \sum_{v\in V} \frac{1}{|M(x)|} \sum_{o\in M(x)} U(x,o) \tag{3.4}$$

$$\forall x, y \in V : D(M(x), D(y)) \leq d(x,y), \tag{3.5}$$

onde a equação 3.4 tem como objetivo a maximização da acurácia do modelo classificador, e a inequação 3.5 representa a restrição de Lipschitz, apresentada anteriormente. Com essa função

objetivo combinada com a condição, é levada em consideração a preocupação quanto à influência na utilidade do modelo quando os requisitos de justiça para indivíduos são atendidos. Com isso, ambos os problemas motivados na Seção **??**, relacionados a um classificador de aprendizado de máquina, são solucionados.

3.2.1 Problemas da paridade estatística para justiça de indivíduos

A paridade estatística (DWORK *et al.*, 2012), que ativa justiça para grupos, é insuficiente para garantir que os indivíduos de um conjunto de dados sejam tratados de forma justa. Existem alguns problemas atrelados ao conceito de paridade estatística, no que diz respeito aos indivíduos dos grupos:

- A utilidade é prejudicada, visto que um fornecedor pode classificar um subgrupo do conjunto de atributos protegidos como o melhor para uma tarefa, sendo que, existem outros subgrupos cujas pessoas pertencentes a eles têm habilidades mais indicadas para o cumprimento da tarefa;
- Pessoas não qualificadas para uma tarefa podem ser selecionadas pelo fornecedor a fim de justificar uma futura discriminação;
- A paridade estatística para um grupo protegido P não implica na paridade estatística dos subgrupos de P.

Neste trabalho serão aplicadas técnicas de justiça para indivíduos que se baseiam nos conceitos apresentados em Dwork *et al.* (2012), explicitados na Seção 3.2, sem considerar paridade estatística que se refere ao conceito de justiça para grupos. Os experimentos serão guiados pelo modelo de árvore de decisão, utilizando especificamente o algoritmo CART.

4 METODOLOGIA

Para a realização do estudo, foram selecionadas e desenvolvidas métricas necessárias para a aplicação e avaliação de justiça para indivíduos, baseando-se na definição proposta por DWORK *et al.*, cujo trabalho é a base do presente estudo.

No decorrer do trabalho, definimos duas propriedades. Para assegurar o cumprimento da justiça para indivíduos, utiliza-se a condição de Lipschitz, explicitada no Capítulo 3, para ambas as propriedades sendo que uma delas utiliza uma relaxação desta restrição. Também são demonstradas aplicações das propriedades no modelo de árvore de decisão, a fim de avaliar a eficácia da proposta. Este capítulo é dividido em duas seções, a primeira contém as métricas utilizadas para a execução das propriedades na abordagem de árvore de decisão, e a segunda explicita as tecnologias e bases de dados utilizadas na avaliação experimental.

4.1 Métricas

Para a ativação das propriedades definidas neste trabalho, a métrica de distância d utilizada na restrição de Lipschitz é a distância de Manhattan ou L_1 (BLACK, 2019), que calcula a distância entre dois itens como a soma das diferenças absolutas dos elementos de cada item. A escolha dessa distância reflete matematicamente na diferença real entre cada um dos itens de dois indivíduos, para os valores categóricos consideramos 1 se forem iguais e 0 caso contrário, assim quando normalizar os dados, todos estarão na mesma escala. Sejam $x, y \in I$, com I representando um conjunto de indivíduos, e m a quantidade de atributos necessários para classificação, o cálculo de d para atributos numéricos na abordagem proposta é:

$$d = \sum_{i=1}^{m} |x_i - y_i|. \tag{4.1}$$

Neste trabalho, a métrica *Earthmover's distance* (EMD) (MALLOWS, 1972), também chamada de distância de *Wasserstein*, foi selecionada para ativar a justiça para indivíduos no modelo proposto de árvore de decisão justa para mensurar a dissimilaridade entre duas distribuições de probabilidade, neste caso, duas distribuições de frequência de classes, que optamos por utilizar devido a quantidade de classificações contabilizadas para cada classe, comumente computada para a escolha do majoritário. Esta métrica computa o fluxo mínimo necessário para igualar duas distribuições de frequência, ou seja, quantas unidades precisam ser movidas de um histograma a outro para que as frequências das classes sejam iguais. Seja $\Gamma(\kappa, \rho)$ o conjunto

de todas as distribuições conjuntas cujas marginais são κ e ρ, o EMD pode ser calculado como indica a Equação 4.2.

$$EMD(\kappa,\rho) = \inf_{\gamma\in\Gamma(\kappa,\rho)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||] \tag{4.2}$$

A Equação 4.2 indica o valor que minimiza a esperança da norma da diferença entre x, y que seguem essa distribuição. Em outras palavras, é equivalente ao fluxo mínimo para igualar duas distribuições.

O cálculo da utilidade é feito por meio da análise entre as predições corretas do modelo e a quantidade de amostras dadas como entrada. Para isso usaremos a métrica de acurácia. A acurácia (ou precisão) é uma taxa referente à pontuação das classificações do modelo. Essa métrica computa o quão boa é uma previsão, cujo resultado é 0 se o modelo classificou incorretamente todas as amostras e 1 se a taxa de sucesso é máxima. Essa métrica compara os valores originais das classes e os valores previstos p de um subconjunto de dados s e gera a fração de amostras classificadas corretamente. Sejam $|s|$ o número de amostras do conjunto e $|p_{verd}|$ a quantidade de amostras cujo modelo classificou corretamente, o cálculo da acurácia é apresentado na Equação 4.3.

$$Accuracy(s,p) = \frac{|p_{verd}|}{|s|}. \tag{4.3}$$

Além de avaliar a utilidade, também deve ser definida uma métrica que mensure a justiça entre os indivíduos presentes nas amostras do conjunto de dados. Para isso, definimos uma função baseada na quantidade de pares de indivíduos $(x,y) \in I \times I$ que respeitam a Equação 3.3.

Definição 4.1 *Seja I o conjunto de indivíduos em uma base de dados, A um modelo de classificação e $n = |I|$. Considerando que $C(k)$ é a notação para o número de pares gerados pela combinação de k itens, e*

$$a_{xy} = \begin{cases} 1, & \text{if } D(M(x),M(y)) \leq d(x,y); \\ 0, & \text{caso contrário.} \end{cases}$$

Para calcular o quão justo é um modelo que classifica um conjunto de indivíduos, a métrica de justiça é computada por

$$Fairness(I,A) = \frac{\sum_{x\in I}\sum_{y\in I\setminus x} a_{xy}}{2\times C(|I|)}. \tag{4.4}$$

A taxa de justiça de um modelo A é quantificada como a taxa de pares justos. O numerador da Equação 4.4 indica a quantidade de pares que satisfazem a restrição de Lipschitz dentre o conjunto de todos os pares de indivíduos. Sabendo que os pares (x,y) e (y,x) são iguais, para evitar que a equação considere-os como pares diferentes, o denominador é multiplicado por 2.

A normalização de um vetor com dados numéricos v é feita pela divisão entre cada um dos valores de v pelo valor máximo do vetor inicial, isto é, $i = normaliza(i,v)$, $\forall i \in v$.

$$normaliza(i,v) = \frac{i}{max(v)} \tag{4.5}$$

4.2 Tecnologias

A implementação do algoritmo de árvore de decisão justa foi realizada na linguagem de programação *Python*, na versão 3.7.6, usando o ambiente *Jupyter Notebook*. As bibliotecas de apoio são *numpy*, *pandas*, *math*, *random*, *itertools*, *functools*, *graphviz*, *multiprocessing*, *scipy* e *scikit-learn*. A fim de reduzir o tempo de processamento, foram utilizadas *threads* para executar paralelamente diferentes fluxos de programas.

4.3 Configurações da árvore de decisão

O modelo de árvore de decisão foi implementado utilizando uma modificação do algoritmo CART, com profundidade máxima selecionada através de um *Grid Search*[1], que escolhe a profundidade da árvore que maximiza a acurácia, e com o índice de Gini como critério de divisão.

4.4 Bases de dados

Os experimentos foram executados usando bases de dados tendenciosas, comumente utilizadas em trabalhos da literatura de justiça. Para o processo de pré-processamento dos dados, foram adotados os seguintes passos:

- Remoção de registros/campos contendo valores faltantes;
- Normalização dos valores numéricos;

[1] Busca iterativa pelos melhores valores de hiper-parâmetros aplicados ao modelo, dado um conjunto de entrada. Um valor de hiper-parâmetro é considerado o melhor possível quando este maximiza uma função objetivo.

Dataset	#atributos	Conjunto de treino tamanho	positivos	Conjunto de Teste tamanho	positivos
German	21	799	30.28%	200	29.00%
COMPAS	54	3794	21.00%	949	20.86%
Adult	103	39073	23.92%	9769	23.87%
Crime	128	1594	59.73%	399	63.66%

Tabela 3 – Tabela com informações gerais sobre os bancos de dados usados para os experimentos.

– Divisão do conjunto de dados entre conjunto de treinamento e teste.

Os conjuntos de dados utilizados são descritos abaixo:

German. A base de dados *German Credit Risk* (DUA; GRAFF, 2017) contém registros de pessoas descritas por um conjunto de atributos categóricos e numéricos, cujo risco de crédito desses indivíduos pode ser classificados como "bom" ou "ruim". Cada registro contém 21 atributos.

COMPAS. O conjunto de dados *COMPAS* (LARSON *et al.*, 2016) contém o histórico criminal, se houve prisão e tempo de prisão, demografia e pontuações de risco computados pelo COMPAS, para réus do condado de Broward, localizado na Flórida. Os dados coletados correspondem aos anos de 2013 e 2014. Para os experimentos, consideramos que a tarefa de classificação é identificar se uma pessoa é reincidente no crime ou não. Cada registro contém 54 atributos.

Adult. O conjunto *Adult Income* (DUA; GRAFF, 2017) foi extraído do banco de dados do Censo dos EUA de 1994 por Barry Becker. Cada amostra contém informações pessoais, educacionais e ocupacionais de um indivíduo, atribuídas a uma classificação binária da renda anual (abaixo ou acima de 50 mil dólares). Nessa base de dados, cada registro possui 103 atributos.

Crime. O conjunto *Communities and Crime* (DUA; GRAFF, 2017) é uma combinação de dados extraídos do Censo dos EUA de 1990, do Relatório Uniforme de Crimes do FBI dos EUA de 1995 e do Levantamento de Estatísticas Administrativas e Gestão de Aplicação da Lei dos EUA de 1990. A base de dados contém amostras com dados socioeconômicos, aplicação da lei e dados criminais para comunidades que vivem nos EUA. A classificação de uma comunidade está acima ou abaixo de $0,15$, que é o valor médio (normalizado) de crimes violentos por população. Este conjunto de dados possui o maior tamanho de registro, com 128 atributos.

As bases de dados foram divididas de forma que 80% dos dados foram destinados para o treinamento do modelo e 20% para teste. A Tabela 3 mostra a quantidade de atributos e

de registros dos conjuntos de treino e teste, além da porcentagem de rótulos positivos para cada subconjunto.

As métricas e bases de dados deste capítulo serão utilizadas para experimentar as propriedades de justiça propostas neste trabalho.

5 λ-JUSTIÇA

Dados o problema e as motivações do Capítulo 1, nosso objetivo principal é construir um modelo de classificação justo ao mesmo tempo que garante alta utilidade. A abordagem de justiça adotada deve classificar pessoas semelhantes de forma similar, evitando a discriminação de características de indivíduos que não são relacionadas a tarefa de interesse da classificação.

5.1 Propriedade de λ-justiça

Nesta seção, é construída a propriedade de λ-justiça e a definição de um modelo λ-justo. Visando mitigar o problema gerado pela propagação de discriminação dos modelos de classificação, existe um ponto principal que deve ser levado em consideração durante o processo de construção da proposta: a ativação de justiça prejudica a acurácia do modelo. Como adaptar um algoritmo de classificação para ser útil a um fornecedor seguindo os protocolos de não-discriminação algorítmica impostos por leis e regulamentações?

A base principal para ativar justiça para indivíduos em um conjunto de dados é a restrição de Lipschitz. Considerando isso, sempre lidaremos com pares de indivíduos para calcular a distância d e a dissimilaridade D entre os mapeamentos. A métrica de justiça também é baseada em pares de indivíduos e na restrição de Lipschitz.

Ao limitar a quantidade mínima de justiça do modelo, mensurada pela métrica apresentada na Equação 4.4, evitamos que o cumprimento restrito da Equação 3.3 altere as classificações de tal forma que o algoritmo rotule todos os indivíduos com a mesma classe. Conforme explicado na Seção 3.2, a maneira mais fácil de ativar justiça para indivíduos é modificar os mapeamentos $M(i)$, $\forall i \in I$, de tal forma que a dissimilaridade D entre todos os pares de mapeamentos seja 0. Entretanto, a utilidade do modelo é prejudicada, descumprindo o objetivo principal de um algoritmo de classificação. Logo, a solução adotada para equilibrar o *trade-off*[1], indicado pela Figura 7, entre utilidade e justiça é limitar a quantidade mínima de algum, ou ambos conforme os interesses do fornecedor.

O foco do estudo é o tratamento justo entre pessoas semelhantes, então, a variável de interesse que será limitada é a justiça. O limitante é um hiper-parâmetro, cujo valor é selecionado pelo fornecedor com base em seus interesses, sempre priorizando altas taxas de justiça.

[1] Influência negativa que uma variável tem sobre outra, por exemplo, maximizar um objetivo o_1 tende a prejudicar outro objetivo o_2.

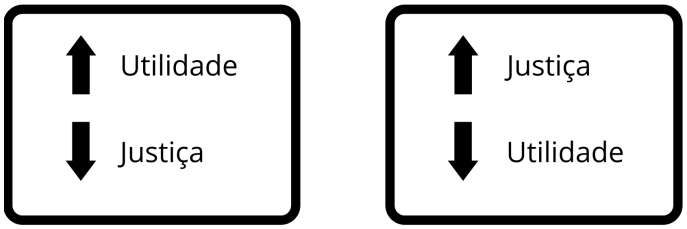


Figura 7 – *Trade-off* entre utilidade e justiça para indivíduos (ou não-discriminação algorítimica).

Definição 5.1 *Um modelo é λ-justo quando pelo menos uma taxa λ dos pares de indivíduos satisfazem a restrição de Lipschitz.*

Isto é, um modelo de classificação A, que tem como entrada um conjunto de dados contendo indivíduos I, obedece a propriedade de λ-justiça se

$$Fairness(I,A) \geq \lambda. \tag{5.1}$$

Ao ativar λ-justiça, a taxa de justiça do modelo, mensurada por 4.4, deve ser necessariamente maior ou igual a λ. Quanto maior o valor desse hiper-parâmetro, mais prejudicada é a taxa de utilidade.

5.2 Árvore de Decisão λ-justa

5.2.1 CART não-binário

Para demonstrar como ativar a propriedade recém definida, este trabalho propõe fazer uma aplicação em um modelo bem conhecido e muito utilizado, a Árvore de Decisão. Conforme explicado no Capítulo 3, o algoritmo CART é ideal para conjuntos de dados que possuem atributos com valores contínuos. Com isso, utilizamos este algoritmo para implementar o λ-justiça, no entanto, fizemos uma modificação para que o algoritmo dividisse a árvore e suas subárvores em mais de dois ramos, permitindo que a árvore construída seja não-binária.

A justificativa para essa modificação se dá pelo fato de que atributos categóricos também serão considerados durante a construção da árvore no momento da divisão do atributo mais puro, conforme calculado pelo índice de Gini. Dessa forma, quando um atributo categórico a_c é selecionado para ser ramificado em um nó, serão gerados $|a_c|$ nós filhos, correspondentes a cada valor que a_c pode assumir. Para atributos contínuos, o algoritmo é idêntico ao do CART convencional.

Considere um exemplo de um cenário de processo seletivo para admissão de novos alunos em uma universidade, em que a tomada de decisão é automatizada, e para classificação dos candidatos, utiliza-se uma árvore CART não-binária. O sistema rotula um candidato como "Classificado", quando os requisitos de um aluno são suficientes para entrar na universidade e "Desclassificado", caso contrário.

O processo de construção da árvore é feito em etapas e o passo inicial é a coleta e limpeza dos dados. Na etapa seguinte, o fornecedor – que neste cenário é a universidade – acessa as classificações efetuadas em processos seletivos anteriores para o treinamento do modelo. O treinamento se baseia na divisão de atributos com maior pureza e ramificações da árvore que podem ser binárias ou não-binárias. Finalmente, os dados dos candidatos são usados como entrada e o algoritmo os classifica como "Classificado" ou "Desclassificado", e os candidatos rotulados positivamente podem adentrar a universidade.

A Figura 8 exemplifica a estrutura de uma árvore CART não-binária. Três atributos foram considerados para classificar os candidatos, conforme mostra o Quadro 1. Os nós folhas, responsáveis por armazenar as classificações dos indivíduos cujas características são correspondentes ao caminho da raiz até a folha, estão coloridos com azul turquesa. Todos os bairros existentes no conjunto de dados devem ser ramificados quando o atributo "Bairro" for selecionado para divisão, pois se trata de um predicado categórico. Enquanto isso, para valores contínuos são divididos dois grupos que dependem do limiar.

Quadro 1 – Atributos do exemplo da Figura 8 e descrição.

Atributo	**Descrição**
Idade	Atributo contínuo, cujo valor pode assumir algum número no intervalo [16-30].
Nota	Atributo contínuo referente a nota do exame de admissão da universidade, que pode variar entre 0 e 10.
Bairro	Atributo categórico que indica o bairro b onde reside o candidato, onde $b \in$ {Benfica, Paupina, Pirambu}.

O rótulo de um nó folha f é equivalente a classe majoritária do subconjunto de indivíduos que obedecem às restrições de um caminho da raiz até f. Para a tomada de decisão, todos os nós folhas devem armazenar a quantidade de indivíduos para cada uma das classes ou rótulos. Para fim de simplicidade, chamamos o CART não-binário λ-justo de λ-FCART. Com tudo isso, a próxima seção mostra como ativar a propriedade de λ-justiça para um modelo de árvore de decisão.

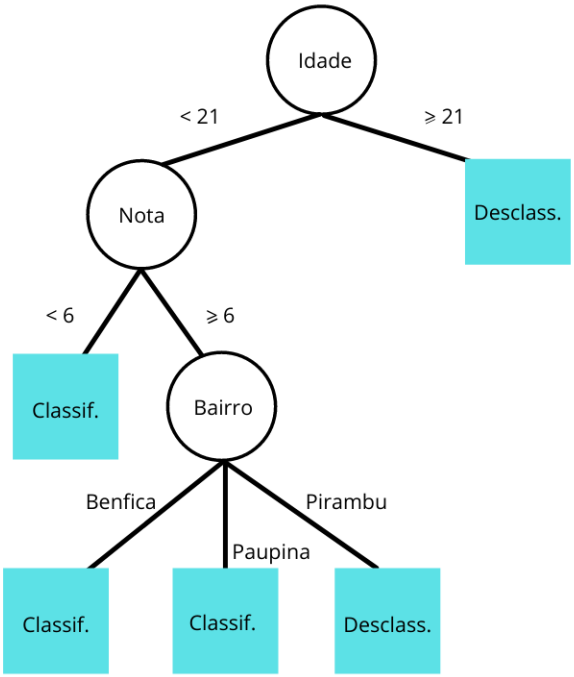


Figura 8 – Árvore CART não-binária gerada para classificar candidatos à vaga na Universidade do cenário fictício.

5.2.2 Propriedade de justiça para Árvores de Decisão

Com o algoritmo de árvore de decisão bem definido para a tarefa de classificação de indivíduos, o próximo passo é ativar a não-discriminação algorítmica no CART não-binário. Como sabemos, para ativar justiça para indivíduos é necessário garantir que o modelo assegure a restrição de Lipschitz, para todo indivíduo $x \in I$. A propriedade de λ-justiça serve para relaxar o cumprimento estrito da restrição de Lipschitz para todos os indivíduos, forçando que haja um limite inferior da taxa de pares justos de indivíduos, ou seja, é necessário que o modelo force que pelo menos uma taxa λ de pares de pessoas obedeça a restrição. O objetivo dessa propriedade é equilibrar a utilidade e a não-discriminação algorítmica de indivíduos, visto que o cumprimento estrito da Equação 3.3 pode acabar prejudicando demasiadamente a utilidade das classificações preditas pelo modelo.

As métricas utilizadas para distância d e dissimilaridade D entre duas distribuições são a distância de Manhattan e a distância EMD, respectivamente. Uma forma de mapear indivíduos para distribuições de frequência em uma árvore de decisão é consultar os nós folhas resultantes do subconjunto de indivíduos que obedecem uma sequência de restrições. Os nós folhas armazenam a quantidade de aparições de cada uma das classes em um determinado subconjunto. Portanto, para facilidade de compreensão, considera-se que cada folha armazena um

conjunto de frequências que é retratado por um histograma. A diferença entre dois histogramas pode ser facilmente visualizada, então, usaremos esta representação nos exemplos deste capítulo.

Com essa escolha de métricas, tem-se que d se trata da distância entre indivíduos do conjunto de dados de entrada, e D é calculado utilizando apenas as informações coletadas pelo modelo e armazenadas nos nós folhas. Isto é, temos dois conjuntos com dimensões diferentes, um conjunto de indivíduos, cujo tamanho é a quantidade de registros do conjunto dado como entrada, e outro, cujo tamanho é a quantidade de folhas da árvore de decisão. Como a dissimilaridade D se resume à diferença entre as distribuições de frequências das folhas, trataremos d como uma variável que também pode ser quantificada para cada folha.

Existe uma motivação que condiz com o interesse de armazenar um limitante de distância em cada folha, que é manter a propriedade de Lipschitz para qualquer nova tupla dada como entrada para ser classificada pelo modelo. Um método para garantir isso é definir um limitante de distância para cada folha da árvore. Dessa forma, previne-se que a condição de Lipschitz seja insatisfeita por algum par de indivíduos.

O limitante de uma folha f_i é a distância mínima entre duas tuplas. Uma é representada pelo caminho da raiz até f_i e a outra é representada pelo caminho da raiz até uma folha f_k, com $k \neq i$. Sabendo que cada aresta de um caminho representa uma restrição, o subconjunto de indivíduos que respeita o conjunto de restrições do caminho da raiz até f_i é usado para contabilizar as classes. Além disso, para esta abordagem, usaremos os subconjuntos de indivíduos para calcular as distâncias d de todos os registros que pertencem a subconjuntos de diferentes folhas. A Figura 9 representa o conjunto de restrições de um dado caminho, o subconjunto de indivíduos de cada folha pode ser encontrado através da seleção de indivíduos que respeitam o conjunto de restrições Φ_P de um caminho P.

Dado um conjunto de indivíduos I, e seja uma *Restrição c* representada pelo formato (at_c, op_c, val_c) – que indica atributo, operação e valor da restrição, respectivamente –, a seleção do subconjunto representado pelo caminho **P**, da Figura 9, pode ser encontrada através da consulta de álgebra relacional abaixo. A consulta resulta no conjunto de indivíduos que obedece uma sequência de restrições.

$$(\sigma_{at_1 op_1 val_1}(I)) \wedge (\sigma_{at_2 op_2 val_2}(I)) \wedge (\sigma_{at_3 op_3 val_3}(I))$$

A distância d selecionada para ser armazenada e representada em cada folha é

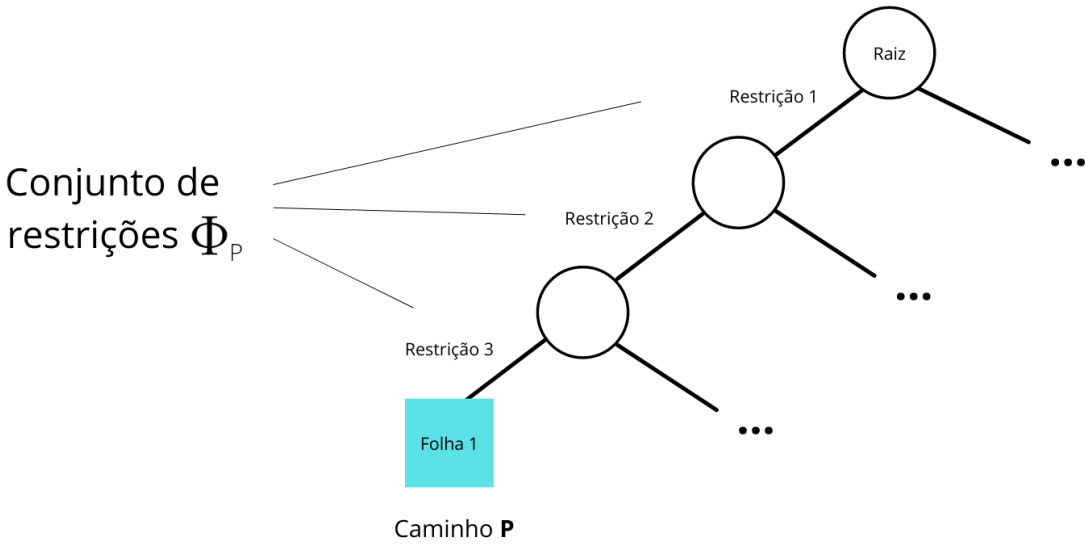


Figura 9 – Conjunto de restrições de um caminho.

puramente a menor distância entre dois indivíduos de subconjuntos diferentes. Um desses subconjuntos é gerado pelo caminho da folha em questão. Dessa forma, as distâncias precisam ser calculadas uma única vez e são armazenadas para verificar se a propriedade de Lipschitz é satisfeita. Computacionalmente falando, calcular as distâncias para todos os pares de indivíduos pertencentes a subconjuntos diferentes ainda é custoso, então optamos e recomendamos explorar o paralelismo por meio de *threads*[2] na implementação desta etapa.

Para a compreensão do processo de seleção de d, cujo resultado que será armazenado em cada folha, considere uma árvore binária com três folhas, conforme o exemplo da Figura 10. Cada caminho ou conjunto de restrições gera um subconjunto de indivíduos, em outras palavras, existe um conjunto de pessoas cujos valores de determinados atributos asseguram o cumprimento de uma sequência de restrições.

Considerando que os subconjuntos gerados para cada caminho são representados como na Figura 11 (a), a distância d armazenada em uma folha f é a menor distância entre um indivíduo do subconjunto do caminho da raiz até f e de outro indivíduo pertencente a um caminho diferente. Chamaremos esse valor de d_f, em que cada folha f mantém uma distância armazenada. Para **Folha 1** do exemplo, seria calculada a distância entre os pares de indivíduos indicados pelas setas na Figura 11 (b). A menor distância calculada dentre estes é a distância selecionada para ser armazenada na **Folha 1**, ou seja, d_{Folha1}. Assim, a restrição de Lipschitz é verificada utilizando apenas informações armazenadas nas folhas.

[2] Execução paralela do código, dividindo o esforço computacional entre diferentes núcleos do processador.

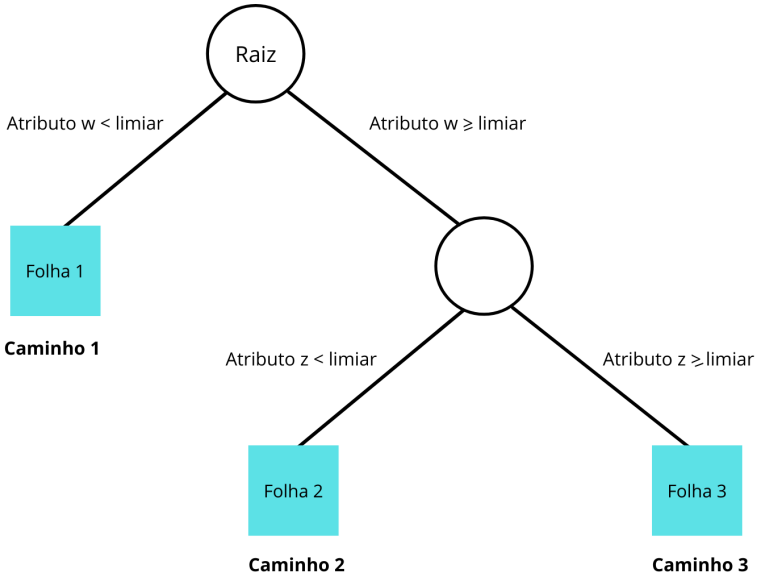
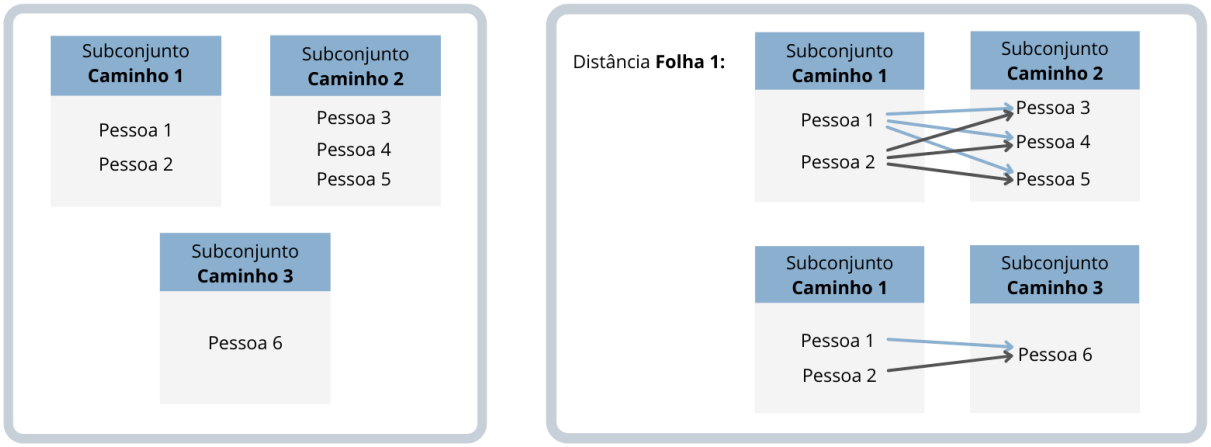


Figura 10 – Árvore binária com três conjuntos de restrições, um conjunto para cada caminho da árvore.



(a) Subconjuntos de indivíduos para cada caminho.

(b) Pares de indivíduos considerados para calcular um limiar d para cada folha.

Figura 11 – Exemplos de subconjuntos e pares de indivíduos considerados para o cálculo de d.

Esse método garante que a distância mínima nunca seja zero, visto que os indivíduos com características iguais pertencem ao mesmo caminho da árvore e, consequentemente, obedecem a restrição de Lipschitz, já que são mapeados para a mesma distribuição de frequência. Com tudo o que foi apresentado nesta seção, podemos assumir que os objetos a serem considerados na restrição de Lipschitz aplicada à árvore de decisão são as folhas, visto que, armazenam as informações necessárias para o cálculo da distância e dissimilaridade.

Agora que sabemos que cada folha f deposita apenas um valor de d_f, a parte direita da Inequação 3.3 será o menor valor de d_f dentre um par de folhas. Assim, o cumprimento da restrição de Lipschitz é garantida para qualquer indivíduo pertencente ao conjunto de entrada em que o modelo deve classificar. Sendo assim, no contexto de árvore de decisão, podemos

restringir ainda mais a Inequação 3.3.

Lema 5.1 *Sejam f_i, f_j folhas pertencentes ao conjunto de nós terminais F de uma árvore de decisão T já treinada e $M(f)$ a distribuição de frequência em uma folha f, temos que a restrição de Lipschitz é uma generalização de*

$$D(M(f_i), M(f_j)) \leq min(d_{f_i}, d_{f_j}). \tag{5.2}$$

Prova 5.1 (Prova 5.1) *Dado que f_i é uma folha cujo conjunto de restrições do caminho representa o subconjunto no qual o indivíduo x pertence, e seguindo a mesma intuição, o indivíduo y pertence ao caminho da raiz até a folha f_j. Queremos provar que 5.3 é uma generalização de 5.4.*

$$D(M(x), M(y)) \leq d(x, y). \tag{5.3}$$

$$D(M(f_i), M(f_j)) \leq min(d_{f_i}, d_{f_j}). \tag{5.4}$$

Afirmamos que os lados esquerdos das inequações são iguais, pois um indivíduo é mapeado para a distribuição de frequência da folha ao final do caminho o qual representa. Em outras palavras, as distribuições de frequência sempre estarão armazenadas nas folhas. Logo, um subconjunto de indivíduos que obedecem um conjunto de restrições de um caminho serão mapeados para a mesma distribuição. Com isso, temos que $D(M(x), M(y)) = D(M(f_i), M(f_j))$, dado que x obedece o conjunto de restrições do caminho da raiz até a folha f_i, e y obedece o conjunto de restrições do caminho da raiz até f_j. Precisamos provar então que

$$d(x, y) \geq min(d_{f_i}, d_{f_j}), \tag{5.5}$$

quando x, y pertencem a subconjuntos distintos. Quando pertencem ao mesmo subconjunto de indivíduos que obedecem a restrições de um caminho, a dissimilaridade entre as distribuições mapeadas a eles é 0. Portanto, a condição de Lipschitz é satisfeita, pois quando ambos pertencem ao mesmo subconjunto, são mapeados a mesma distribuição de frequência.

Se d_{f_i} armazena a menor distância computada entre dois indivíduos de subconjuntos diferentes, dado que um dos subconjuntos contém os indivíduos que respeitam o conjunto de restrições do caminho da raiz até f_i, e por outro lado d_{f_j} também armazena o menor valor de distância. Então, no pior caso temos que a distância entre os indivíduos x e y é a menor distância calculada para ambas as folhas e, consequentemente,

$$d(x,y) = min(d_{f_i}, d_{f_j}). \tag{5.6}$$

No melhor caso, x e y pertencem aos subconjuntos das folhas f_i e f_j, respectivamente. No entanto, não possuem a menor distância dentre os subconjuntos, e

$$d(x,y) > min(d_{f_i}, d_{f_j}). \tag{5.7}$$

Logo, fazendo um paralelo com a Definição 5.1, dizemos que uma árvore de decisão é λ-justa quando pelo menos uma taxa λ dos pares de **folhas** satisfazem a restrição de Lipschitz. Considerando as variáveis da Definição 5.1 e definindo

$$a(f_i, f_j) = \begin{cases} 1, & \text{if } D(M(f_i), M(f_j)) \leq min(d_{f_i}, d_{f_j}); \\ 0, & \text{caso contrário.} \end{cases}$$

O cálculo da métrica de justiça (Equação 4.4) utilizando as informações das folhas de uma árvore de decisão é

$$Fairness(F,T) = \frac{\sum_{f_i \in F} \sum_{f_j \in F \setminus f_i} a(f_i, f_j)}{2 \times C(|F|)}. \tag{5.8}$$

O cálculo acima segue a mesma lógica da Equação 4.4. A diferença é que agora a restrição de Lipschitz é aplicada sobre as folhas e não diretamente sobre os indivíduos do conjunto de entrada.

5.2.3 Ativação de justiça e Balanceamento das Folhas

Com tudo o que foi apresentado nas seções anteriores, temos base suficiente para ativar a propriedade de justiça para um par de pessoas no modelo de árvore de decisão. A ideia é modificar as classificações, equilibrando-as quando dois indivíduos são semelhantes para a realização de uma tarefa. Para a árvore de decisão, representamos um conjunto de indivíduos

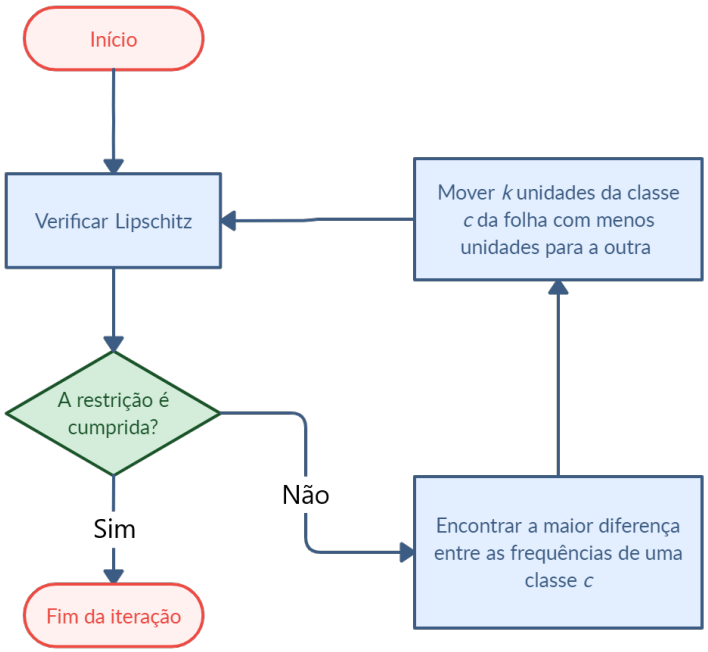


Figura 12 – Fluxograma do funcionamento da etapa de balanceamento das folhas.

como sendo folhas, reduzindo o espaço de busca. A verificação da restrição de Lipschitz é essencial para capturar os pares de folhas que não cumprem o conceito de justiça para indivíduos.

As folhas armazenam distribuições de frequência e a ideia da nossa abordagem é obrigar que a diferença entre essas distribuições seja limitada pelo menor valor de d_f de um par de folhas. A Figura 12 indica o fluxograma do algoritmo na etapa de balanceamento entre duas folhas. Para cada par de folhas, é feita uma verificação. Se o par de folhas obedece a restrição de Lipschitz, então, o algoritmo pode ser executado para o próximo par. Caso a restrição não seja cumprida, é necessário modificar as distribuições de frequência das folhas até que a condição seja satisfeita.

Para facilitar a compreensão, representamos uma distribuição de frequências por meio de um histograma. Supondo que o modelo está resolvendo um problema de classificação binária, isto é, existem duas classificações possíveis para um indivíduo em que o conjunto de saídas é representado por $O = \{0, 1\}$ e temos que comparar duas folhas, **Folha 1** e **Folha 2**. Essas folhas armazenam os dados de d_f já normalizados e a maior dissimilaridade computada no modelo para normalizar as dissimilaridades. Possibilitando assim, calcular a restrição de Lipschitz com d e D na mesma escala. O Quadro 2 indica os valores que compõem nosso exemplo.

O algoritmo verifica se o par de folhas (**Folha 1**, **Folha 2**), apresentado na Figura 13, satisfaz a restrição de Lipschitz, isto é, se a dissimilaridade entre as distribuições de frequência

Quadro 2 – Quadro com dados para exemplificar o funcionamento do algoritmo.

Dado Armazenado	Valor
d_{Folha1}	0.2
d_{Folha2}	0.25
Distribuição de frequência da Folha 1	3 indivíduos rotulados como 0, e 11 indivíduos rotulados como 1.
Distribuição de frequência da Folha 1	29 indivíduos rotulados como 0, e 9 indivíduos rotulados como 1.
Maior dissimilaridade do conjunto	15

são menores ou iguais a 0.2, de acordo com o exemplo do Quadro 2. A dissimilaridade entre as distribuições das folhas 1 e 2 calculada pela distância EMD é 12. Ao normalizar esse valor, por meio da divisão deste pela maior dissimilaridade do conjunto, obtemos o valor 0.8. Como $0.8 \geq 0.2$, a restrição de Lipschitz é violada levando à necessidade de balancear as folhas. Para isso, é essencial seguir os dois passos do fluxograma da Figura 12 que sucedem a tomada de decisão quando a restrição não é cumprida.

Primeiro é preciso identificar a classe que contém a maior diferença entre as frequências das duas folhas. O algoritmo compara as classes das diferentes folhas conforme indica a Figura 13. Na classe 0, a diferença entre as folhas 1 e 2 é de $|3-29| = 26$ unidades, e na classe 1 a diferença é de $|11-9| = 2$ unidades. Neste caso, contabilizando as frequências, a classe mais diferente dentre as distribuições é a classe 0. Para equilibrar essas distribuições e reduzir a dissimilaridade entre elas, movemos k unidades da classe de um histograma a outro. O cálculo de k é simplesmente o valor inteiro que iguala (ou mais se aproxima de igualar) as frequências da classe mais diferente entre o par de folhas. Dado que $F(C_i)$ indica a frequência de indivíduos rotulados como a classe i na folha F, a Equação 5.9 contém a fórmula de como encontrar o valor de k em uma classificação binária, com $O = \{0,1\}$.

$$k = \left\lfloor \frac{max(|Folha1(C_0) - Folha2(C_0)|, |Folha1(C_1) - Folha(C_1)|)}{2} \right\rfloor \tag{5.9}$$

Seguindo as etapas do fluxograma, após mover $k = \frac{26}{2} = 13$ unidades da classe 0 da **Folha 2** para a classe 0 da **Folha 1**, obtemos os histogramas da Figura 14, e o algoritmo novamente verifica se a restrição de Lipschitz é satisfeita. Temos que a dissimilaridade normalizada é igual a

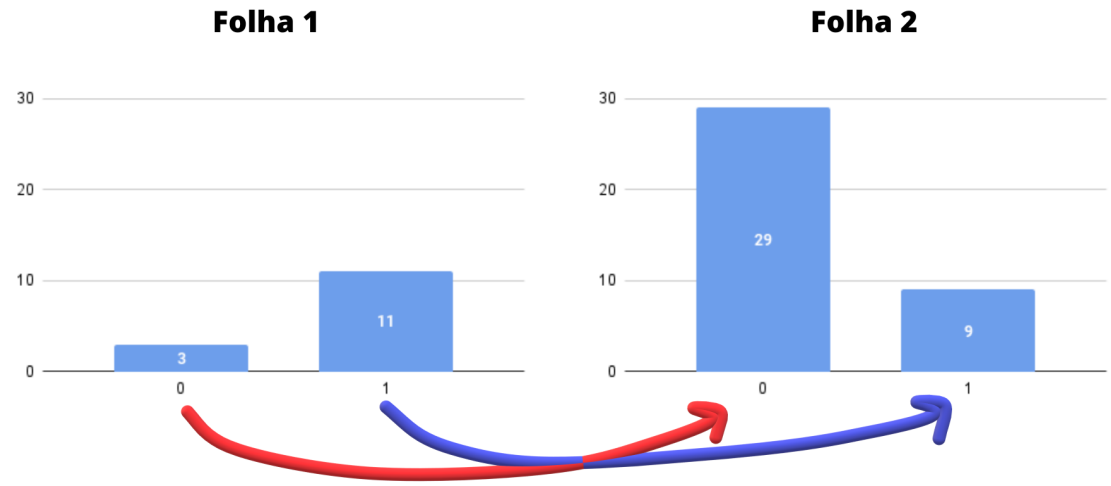


Figura 13 – Comparação de classes entre dois histogramas.

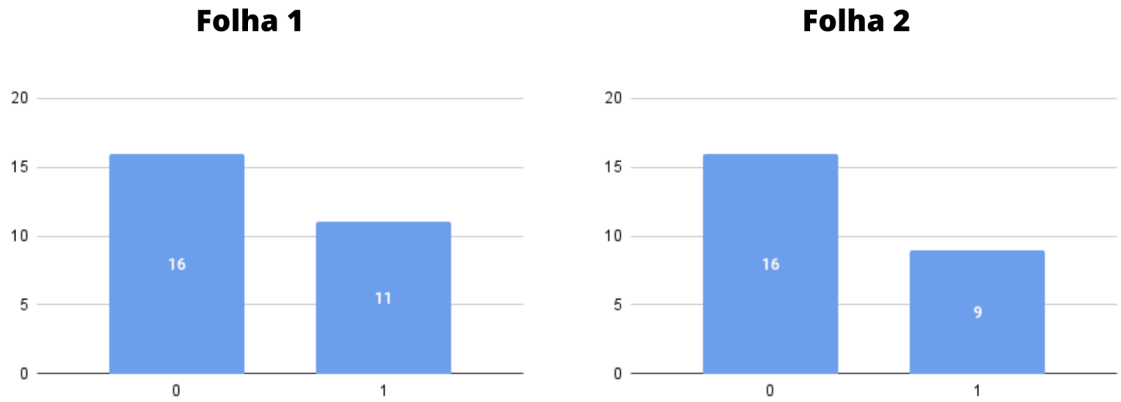


Figura 14 – Histogramas das folhas 1 e 2 após o algoritmo equilibrar as frequências.

$$\frac{EMD(Folha1, Folha2)}{15} = \frac{1}{15} = 0.066$$

que é menor que 0.2, cumprindo a restrição de justiça.

5.2.4 λ-justiça em Árvores de Decisão

Para aplicar a propriedade de λ-justiça na árvore de decisão, deve-se garantir que o modelo cumpra a restrição da Equação 5.1, em que a métrica que mensura justiça é a Equação 5.8 e λ é um hiper-parâmetro definido na entrada do modelo pelo fornecedor. Sendo assim, o modelo deve assegurar que

$$Fairness(F,T) \geq \lambda. \tag{5.10}$$

O algoritmo ativa λ-justiça iterativamente, e a cada iteração todos os pares de folhas que não satisfazem a restrição de Lipschitz são balanceados. Depois da primeira iteração, o algoritmo verifica se a Equação 5.10 é verdadeira. Caso não seja, todos os pares injustos remanescentes são recuperados e balanceados até que a Equação 5.10 seja verdade. O Algoritmo 5.1 mostra o pseudocódigo formulado pela ideia apresentada neste parágrafo.

Algoritmo 5.1: λ-justiça em Árvore de Decisão

1 **Entrada:** Árvore T, taxa mínima de justiça permitida λ.
2 $F \leftarrow$ Folhas(T)
3 pares_injustos $\leftarrow$ Encontrar_pares_injustos(T)
4 Fairness(F, T) $\geq \lambda$ **retornar** T Fairness(F, T) $< \lambda$ **para** *(f_i, f_j) in pares_injustos* **faça**
5 $\quad f_i, f_j \leftarrow$ Balancear(f_i, f_j)
6 pares_injustos $\leftarrow$ Encontrar_pares_injustos(T) **retornar** T

A cada chamada da função Balancear na linha 8 do Algoritmo 5.1, a árvore é modificada de tal forma que as classificações das folhas injustas passadas como parâmetros da função são redistribuídas para que as distribuições de frequência fiquem mais semelhantes, reduzindo o valor de dissimilaridade entre elas. Dessa forma, a taxa de justiça deve ser recalculada a cada iteração para que seja feita a verificação da linha 6.

Algoritmo 5.2: Encontrar_pares_injustos

1 **Entrada:** Árvore T
2 pares_injustos $\leftarrow \emptyset$
3 $a \leftarrow 1$
4 $F \leftarrow$ Folhas(T)
5 $S \leftarrow$ F
6 **para** f_i *in* F **faça**
7 $\quad S \leftarrow S \setminus f_i$
8 $\quad d_{f_i} \leftarrow$ menor distância entre um indivíduo do subconjunto da folha f_i e de um indivíduo de outro subconjunto.
9 $\quad f_j$ in S
10 $\quad d_{f_j} \leftarrow$ menor distância entre um indivíduo do subconjunto da folha f_j e de um indivíduo de outro subconjunto.
11 $\quad$ D($M(f_i), M(f_j)$) $\geq \min(d_{f_i}, d_{f_j})$ pares_injustos$_a \leftarrow (f_i, f_j)$
12 $\quad a \leftarrow a + 1$
13 **retornar** pares_injustos

O Algoritmo 5.2 encontra os pares de folhas na árvore T que não satisfazem a Equação 5.2, que é a restrição de Lipschitz, a qual adaptamos para árvore de decisão. Os

pares de folhas injustos são balanceados conforme o Algoritmo 5.3, que executa o processo de redistribuição de unidades de frequência de um histograma a outro (Figuras 13 e 14).

Nas linhas 2 a 5 do Algoritmo 5.2, são declaradas variáveis que são utilizadas para encontrar os pares de folhas que não obedecem a restrição de Lipschitz e armazená-los no vetor criado na linha 2. Como um par de folhas **(Folha 1, Folha 2)** é equivalente ao par **(Folha 2, Folha 1)**, são criadas duas variáveis que armazenam as folhas F de uma árvore T. Para reduzir o espaço de busca, os pares de folhas são comparados de forma triangular a fim de evitar que um mesmo par seja contado duas vezes. Desta forma, S é uma partição de F que não considera as folhas que já foram chamadas na linha 6. Os valores de distância armazenados em cada folha do par são guardados em duas variáveis para que a verificação da restrição da linha 11 seja possível. Caso o par viole a condição, ele é armazenado no vetor de pares injustos. A próxima iteração considera outro par de folhas ainda não verificado e reexecuta o código.

Algoritmo 5.3: Balancear

1 **Entrada:** f_i, f_j
2 $c \leftarrow$ a classe mais diferente entre f_i e f_j
3 $k \leftarrow \left\lfloor \frac{|freq(f_i,c) - freq(f_j,c)|}{2} \right\rfloor$
4 $freq(f_i,c) > freq(f_j,c)\ freq(f_i,c) \leftarrow freq(f_i,c) - k$
5 $freq(f_j,c) \leftarrow freq(f_j,c) + k$
6 $freq(f_i,c) \leftarrow freq(f_i,c) + k$
7 $freq(f_j,c) \leftarrow freq(f_j,c) - k$
8 **retornar** f_i, f_j

O Algoritmo 5.3 evidencia como equilibrar duas distribuições de frequência. Dada uma entrada com um par de folhas, é identificada a classe c cujas frequências são mais diferentes entre os dois objetos. Em seguida, o valor k é computado, sendo ele o valor inteiro necessário, ou mais próximo, para igualar as duas frequências na classe c. Nas linhas 4 a 9, a redistribuição de frequências é feita de tal forma que a folha com mais unidades em c transfere k unidades para a outra folha. Ao final da execução, o algoritmo retorna as folhas com as distribuições já modificadas.

A propriedade de λ-justiça pode ser ativada em qualquer modelo de classificação. A fins de exemplo, aplicamos a propriedade no modelo de árvore de decisão utilizando o algoritmo CART adaptado para lidar com atributos categóricos. A ideia principal da proposta é balancear as distribuições de probabilidade ou frequência mapeadas para dois indivíduos, de forma a garantir que ambos sejam tratados de forma justa.

6 (λ, δ)-JUSTIÇA

6.1 Quando λ-justiça é impraticável?

A segunda contribuição é a definição e ativação do que chamamos de (λ, δ)-justiça. Essa propriedade é aplicada quando não é possível alcançar λ-justiça. Isso acontece quando três situações acontecem simultaneamente, são elas:

(i) Quando muitos pares de indivíduos têm distâncias (normalizadas) próximas de zero;

(ii) Quando todos os indivíduos são mapeados para distribuições de probabilidade parecidas, e consequentemente o maior valor de dissimilaridade usado para normalizar D não implica em uma grande diferença dos valores normalizados, a não ser que a dissimilaridade entre duas distribuições de probabilidade já seja zero;

(iii) Quando não é possível tornar duas distribuições de probabilidade iguais; no caso da árvore de decisão, os indivíduos são mapeados para distribuições de frequência, e, como não existe frequência fracionada, ou seja, uma parcela de um indivíduo, nem sempre é possível tornar os histogramas iguais.

Quando (i) e (ii) acontecem e não é possível igualar as frequências de dois histogramas, a propriedade λ-justiça pode não ser factível. Um exemplo disso ocorre quando é impossível satisfazer a restrição de Lipschitz a não ser que as distribuições sejam iguais. Vamos considerar um exemplo em que $d = 0.08$, e a maior dissimilaridade computada no conjunto é 12. Almejamos modificar as distribuições de frequência tal que

$$\frac{EMD(Folha1, Folha2)}{12} \leq d.$$

A Figura 15 compõe o exemplo com dois histogramas que devem ser balanceados

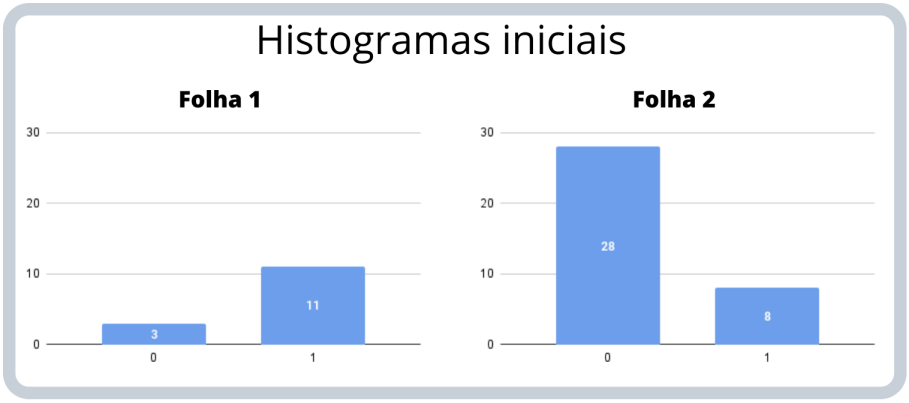


Figura 15 – Exemplo de distribuições de frequência antes de serem balanceadas.

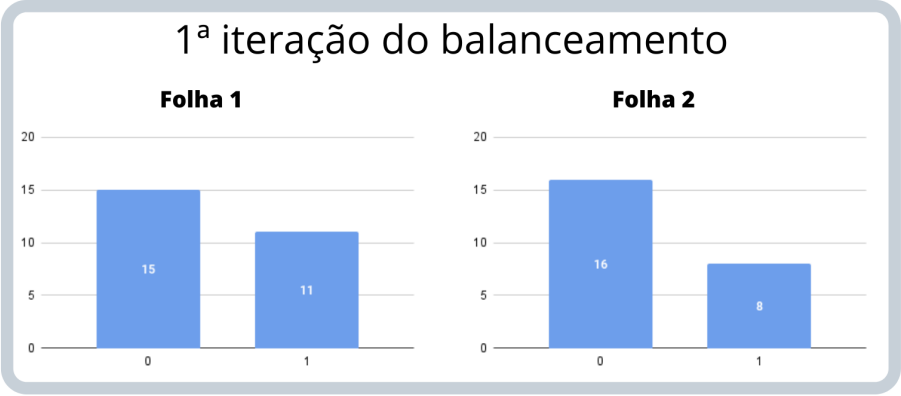


Figura 16 – Histogramas do exemplo da Figura 15 após uma iteração do Algoritmo 5.3.

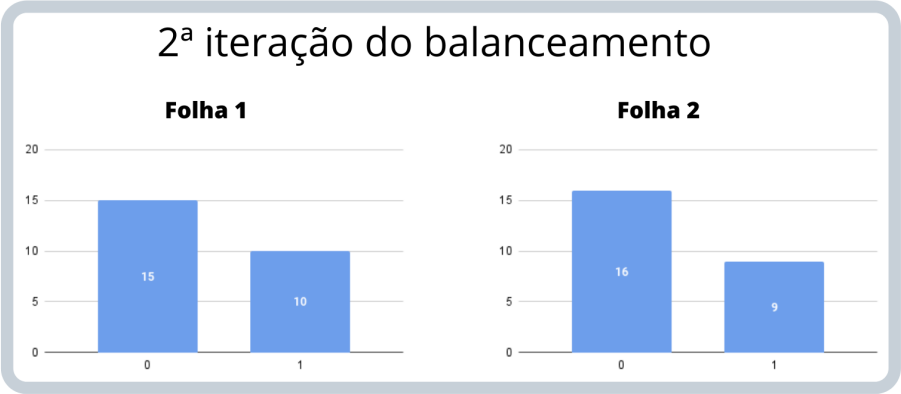


Figura 17 – Histogramas do exemplo da Figura 15 após duas iterações do Algoritmo 5.3

até que a dissimilaridade (normalizada) entre eles seja menor ou igual a d.

Aplicando o Algoritmo 5.3 neste exemplo, obtemos as modificações representadas na Figura 16. É nítido que, apesar de balancear as folhas por meio da redistribuição de frequências na classe 0, as unidades permanecem diferindo, mas agora em apenas uma unidade. Vamos verificar se esse par ainda viola a restrição e se ainda precisa ser modificado:

$$\frac{EMD(Folha1, Folha2)}{12} = 0.1666 \geq 0.08.$$

A restrição de Lipschitz continua sendo violada e consequentemente esse par permanece sendo injusto. Ao balancear novamente as distribuições de frequência, tem-se o resultado ilustrado na Figura 17. O mesmo acontece com a classe 1, as frequências entre os dois histogramas também diferem em apenas uma unidade. Se a condição de Lipschitz não for satisfeita nesta iteração, então não é possível satisfazê-la.

Como lidamos com frequências, isto é, números inteiros que quantificam os indivíduos que são rotulados com uma determinada classe, não é possível dividir um indivíduo ou uma unidade de frequência em partes fracionais. Ao verificar novamente se a Equação 5.2 é satisfeita,

temos que

$$\frac{EMD(Folha1, Folha2)}{12} = 0.0833 \geq 0.08.$$

Sendo assim, é impraticável assegurar a condição de Lipschitz nesta situação. Se todos ou quase todos os pares se comportam dessa forma, a propriedade λ-justiça pode não ser satisfeita, visto que não há modificações factíveis na abordagem que sejam suficientes para garantir que uma taxa λ dos pares de folhas seja justa. Com isso, formulamos outra propriedade que é uma relaxação do λ-justiça que é aplicada em casos como o exemplificado.

6.2 O que é (λ, δ)-justiça?

A propriedade (λ, δ)-justiça possui o mesmo princípio do λ-justiça, com uma pequena alteração na restrição de Lipschitz. A atualização é feita por meio da adição de uma folga no lado direito da Equação 3.3. O objetivo é que um modelo alcance λ-justiça, tal que pelo menos uma taxa λ de pares de indivíduos $(x, y) \in I \times I$ assegure que

$$D(M(x), M(y)) \leq d(x, y) + \delta, \tag{6.1}$$

em que δ é um valor inteiro positivo e $\delta \in (0, 1)$. O recomendado é que $\delta << 1$, pois quanto maior o valor da folga maior a perda de justiça.

A propriedade λ-justiça é equivalente a aplicação de (λ, δ)-justiça para $\delta = 0$. A variável de folga trata-se de outro hiper-parâmetro que é dado como entrada do algoritmo justo.

Definição 6.1 *Um modelo é (λ, δ)-justo quando pelo menos uma taxa λ do total de pares de indivíduos satisfaz a restrição de Lipschitz com a adição de uma folga δ.*

Fazendo um paralelo com a Equação 5.1, temos que um modelo é (λ, δ)-justo quando, dado

$$a_{xy} = \begin{cases} 1, & \text{if } D(M(x), M(y)) \leq d(x, y) + \delta; \\ 0, & \text{caso contrário,} \end{cases}$$

é garantido que

$$\frac{\sum_{x \in I} \sum_{y \in I \setminus x} a_{xy}}{2 \times C(|I|)} \geq \lambda. \tag{6.2}$$

6.3 Árvore de Decisão (λ, δ)-justa

A aplicação da propriedade (λ, δ)-justiça para o modelo de árvore de decisão é similar àquela apresentada na Subseção 5.2.4. A única diferença é que no Algoritmo 5.2 a linha 11 é modificada para atender a relaxação, sendo assim, a atualização da condição é

$$D(M(f_i), M(f_j)) \geq min(d_{f_i}, d_{f_j}) + \delta, \tag{6.3}$$

e quando a Equação 6.3 for verdade é necessário balancear as folhas.

A escolha de δ depende do conjunto de dados utilizado para treinar o modelo. O fornecedor pode selecionar o melhor valor, conforme os seus interesses, analiticamente ou automaticamente por meio de um *Grid Search*, cujo o objetivo é maximizar os valores das métricas de utilidade e justiça, dado que o modelo deve garantir a propriedade de λ-justiça, ou neste caso, (λ, δ)-justiça.

Apesar de um modelo λ-justo ser o suficiente para garantir justiça na maioria dos casos, podem existir conjuntos de dados que não permitem o cumprimento da propriedade. Quando as situações (i), (ii) e (iii) ocorrem, a melhor opção é ativar a propriedade relaxada. É imprescindível que o fornecedor avalie a qualidade do conjunto de dados e do mapeamento analiticamente, por meio da observação dos resultados da ativação das propriedades em um modelo.

A propriedade (λ, δ)-justiça utiliza os mesmos conceitos do λ-justiça, no entanto, é necessária apenas nos três casos apresentados neste capítulo, especificamente quando as distribuições mapeadas aos indivíduos se tratam de distribuições de frequência. Como é o caso adotado neste trabalho com a árvore de decisão.

7 RESULTADOS

Neste capítulo, são apresentados os resultados alcançados durante todo o trabalho, bem como uma discussão e comparação com os resultados encontrados na literatura, destacando a importância desta pesquisa. Avaliaremos a qualidade das abordagens propostas comparando-a com os resultados obtidos pelo *Fairness aware Training of Decision Trees* (FATT). Os experimentos consistem em uma avaliação analítica da árvore de decisão justa, variando os hiper-parâmetros λ e δ. A altura é escolhida a partir de um *Grid Search* que retorna a profundidade da árvore que maximiza a acurácia dado um conjunto de treinamento.

Lembrando que o FATT também constrói um modelo justo de árvore de decisão, especificamente, usando um conjunto de árvores. A técnica proposta em (RANZATO *et al.*, 2021) ativa justiça para indivíduos durante a construção do modelo enquanto a nossa técnica aplica na etapa de pós-processamento.

Configurações experimentais. A implementação da árvore de decisão aplicando as propriedades definidas neste trabalho foram feitas utilizando a linguagem *Python* na versão 3.7.6. Para a execução do código, foi utilizado o ambiente *Jupyter Notebook* e, por fim, os experimentos foram conduzidos em uma máquina com processador *Intel Core i7-7800X* de 12 núcleos, e com 16GB de memória RAM. Utilizando-se *threads*, dividimos tarefas demoradas entre os 12 núcleos do processador, explorando o paralelismo e otimizando o tempo de execução.

Os conjuntos de dados utilizados para a experimentação são conjuntos conhecidos e utilizados na literatura de não-discriminação algorítmica, sendo eles os conjuntos de dados descritos na Subseção 4.4: *German Credit Risk*, COMPAS, *Adult Income*, e *Crime and Communities*. Todos são definidos com dois possíveis rótulos ou saídas, portanto, é atribuído um problema de classificação binária para a árvore de decisão justa.

Inicialmente apresentamos os resultados para a árvore de decisão λ-justa a fim de comparar a eficácia do modelo em relação ao que já foi apresentado anteriormente por outros pesquisadores. Por fim, mostraremos como se comporta a acurácia do modelo quando ativada a propriedade de (λ, δ)-justiça, com a finalidade de analisar quanto de justiça é perdida com a ativação da propriedade. Como todos os conjuntos de dados selecionados para a avaliação experimental aceitam a restrição de λ-justiça, ou seja, não ocorrem as situações descritas na Seção 6, que impedem a garantia da propriedade, avaliamos o quão prejudicial para a justiça é a adição da folga δ.

Executamos o algoritmo cinco vezes para cada combinação de hiper-parâmetros,

Conjunto de Dados	Profundidade
German	3
COMPAS	3
Adult	3
Crime	8

Tabela 4 – Profundidades selecionadas pelo *Grid Search*, baseando-se na melhor acurácia.

	Taxa de Acurácia			Taxa de Justiça		
Dataset	CART	FATT	0.8-FCART	CART	FATT	0.8-FCART
German	0.722	0.7200	0.672	0.58	0.9950	0.8835
COMPAS	0.9883	0.6411	0.8269	0.5	0.8598	0.8667
Adult	0.8459	0.8084	0.8459	0.8269	0.9521	0.8269
Crime	0.8531	0.7945	0.8531	0.9357	0.7519	0.9357

Tabela 5 – Comparação entre modelos acerca das taxas de acurácia e justiça.

λ e δ, e calculamos os valores médios das métricas para cada uma das combinações, para acompanharmos o comportamento do algoritmo justo. As profundidades selecionadas pelo *Grid Search*, para as árvores de decisão com entradas referentes a cada um dos conjuntos, são destacadas na Tabela 4, em que os valores possíveis de profundidade eram 3,6,8,10,12.

A Figura 18 mostra quão acurados e justos são os conjuntos de dados selecionados para a experimentação, quando aplicados no modelo de árvore de decisão, especificamente no CART. Podemos observar que o conjunto COMPAS, apesar de ter a maior acurácia, possui a menor taxa de justiça, com 0.9883 e 0.5 para as respectivas métricas. Vale ressaltar que este conjunto é o mais desbalanceado em relação às classificações. Em contraste, o modelo aplicado ao conjunto *Crime* computou uma taxa de justiça de 0.9357, que já atende a propriedade λ-justiça para valores altos de λ como $\lambda = 0.9$, sem necessitar de nenhuma modificação nas classificações do modelo para satisfazer a propriedade. O modelo aplicado ao conjunto *German*, computou a menor taxa de acurácia, com 0.722 e uma taxa de justiça equivalente a 0.58. Ademais, o conjunto *Adult* possui taxas semelhantes para ambas as métricas sendo elas 0.8459 e 0.8269 para acurácia e justiça, respectivamente. Vale ressaltar que a árvore mais profunda teve como entrada o conjunto *Crime*, que obteve a maior taxa de justiça dentre todos os conjuntos.

A Tabela 5 mostra os resultados do modelo de árvore de decisão justo e compara esses resultados com aqueles das abordagens do FCART e do FATT. Com os dados mostrados, podemos notar que dois conjuntos de dados já satisfazem 0.8-justiça quando classificados pelo modelo de árvore de decisão. Os conjuntos *Adult* e *Crime* alcançam 0.8269 e 0.9357 de justiça, portanto não precisam ser modificados quando o hiper-parâmetro $\lambda = 0.8$. Ademais, os resultados obtidos com o CART e 0.8-FCART (que neste caso é equivalente ao CART) superam o FATT tendo como entrada esses dois conjuntos.

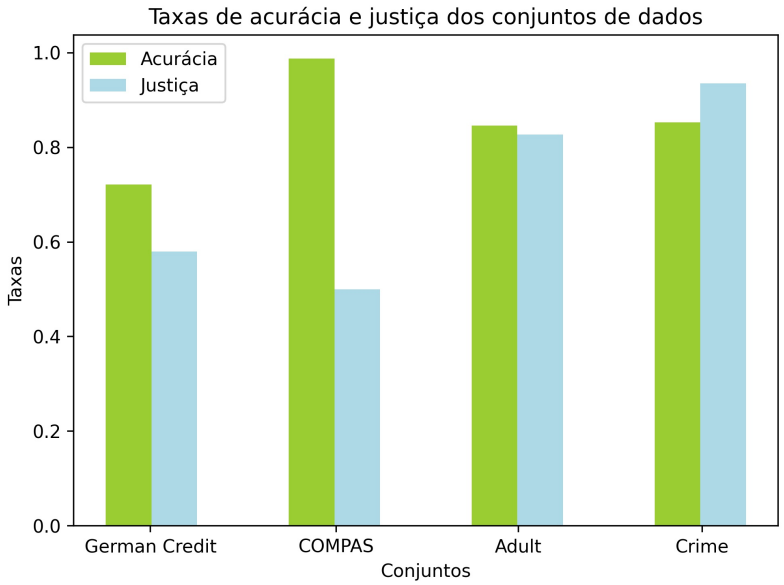


Figura 18 – Taxas de acurácia e justiça do modelo aplicado a cada um dos conjuntos de dados.

Para os demais conjuntos de dados, *German* e COMPAS, o modelo CART precisou ser modificado para o 0.8-FCART com a finalidade de ativar a propriedade de 0.8-justiça. No conjunto COMPAS, nosso modelo justo superou o FATT em ambas as métricas. No entanto, com a entrada sendo o conjunto de dados *German*, nosso modelo foi superado pelo FATT, o que nos faz concluir que nossa aproximação lida melhor com dados em maiores escalas.

As escolhas dos valores do hiper-parâmetro δ foram feitas analiticamente para que fosse possível analisar o comportamento dos gráficos em crescimento ou decrescimento, dados λ e δ. Para todos os gráficos apresentados sobre os diferentes conjuntos de dados, é possível observar um comportamento similar no que diz respeito ao crescimento ou decrescimento da reta na medida que δ aumenta. Ao ampliar a relaxação na restrição de Lipschitz, a acurácia aumenta e a taxa de justiça diminui. Isso acontece devido ao modelo justo se aproximar mais do original (ou não justo) quando o valor de δ aumenta, visto que a restrição fica mais suave. Ao se aproximar dos valores originais, a taxa de justiça diminui, pois a restrição da propriedade de $(\lambda,0)$-justiça é mais estrita que (λ,δ)-justiça, quando $\delta > 0$. Logo, percebemos que $\delta > 0$ afeta a taxa de justiça. Todos os gráficos de linhas apresentados neste capítulo indicam o quanto o modelo ganha de acurácia e perde de justiça para diferentes valores de δ.

Os gráficos de linhas representados nas Ilustrações 19 e 20 representam, respectivamente, a acurácia do modelo aplicado no conjunto *German* variando os valores de λ e δ e a taxa de justiça para os mesmos parâmetros. No conjunto *German Credit Risk*, houve um crescimento no comportamento da reta que indica a acurácia do modelo ao variar os hiper-parâmetros, na

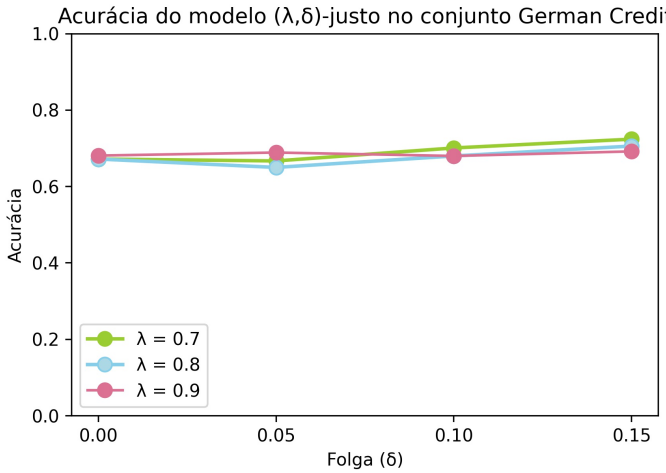


Figura 19 – Acurácia do modelo performando no conjunto de dados *German Credit Risk* variando os hiper-parâmetros λ e δ.

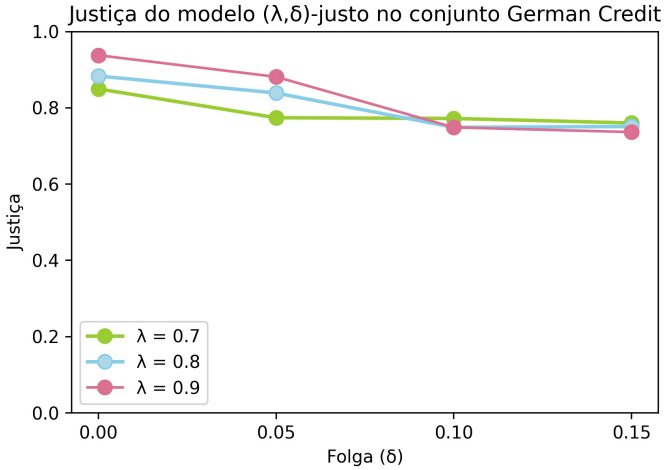


Figura 20 – Justiça do modelo performando no conjunto de dados *German Credit Risk* variando os hiper-parâmetros λ e δ.

Figura 19. As retas verde e azul possuem um comportamento similar, enquanto a acurácia se mantém a mesma ao variar δ, quando $\lambda = 0.9$. Quanto menor o valor de λ, menor a quantidade de modificações necessárias para cumprir a propriedade de λ-justiça, portanto é compreensível o motivo das linhas que representam o menor valor de λ tenham maior acurácia e menor taxa de justiça. Na Figura 20, a taxa de justiça converge a aproximadamente 0.78 para $\lambda = 0.7$ e δ entre 0.05 e 0.15.

O segundo conjunto de dados a ser analisado é o COMPAS, como indicam as figuras 21 e 22. Quando $\lambda = 0.7$ e $\lambda = 0.8$, as retas possuem o mesmo comportamento, enquanto que para $\lambda = 0.9$ as taxas de justiça e acurácia se alteram quando $\delta = 0.10$, ou seja, este valor de folga não foi o suficiente para alterar nitidamente o valor de acurácia comparado a $\delta = 0$. Ainda como reflexo do *trade-off* entre as duas métricas, enquanto a acurácia aumenta

quando δ cresce, a taxa de justiça diminui. A partir de um determinado valor da folga, os resultados se estabilizam, pois se aproximam mais do modelo original, sem necessitar de mais modificações para ajustar o algoritmo para o cumprimento das propriedades.

Conforme a Figura 23, para o conjunto *Adult*, o comportamento das retas quando λ é 0.7 e 0.8 é quase o mesmo, com exceção para o valor de hiper-parâmetro $\delta = 0$. As linhas se mantêm constantes para os valores de $\delta \in [0.015, 0.05]$ com $\lambda = 0.7$ e $\lambda = 0.8$. Como a média das taxas de justiça do modelo com a entrada de dados sendo o conjunto *Adult* tem valor equivalente a 0.8269, não são necessárias modificações para satisfazer 0.8-justiça.

A pequena diferença de valores de acurácia para $\lambda = 0.8$ e $\delta = 0$ quando comparado a $\lambda = 0.7$ e $\delta = 0$ é porque uma das iterações da execução obteve taxa de justiça do modelo original equivalente a 0.78, que não obedece 0.8-justiça, e consequentemente precisou ter as classificações do modelo modificadas e a média da acurácia diminuiu para esses hiper-parâmetros.

Na Figura 24, a reta que representa as taxas de justiça obtidas quando $\lambda = 0.7$ se mantém constante para todos os valores de δ indicados, enquanto que para $\lambda = 0.8$ a taxa de justiça é um pouco maior quando $\delta = 0$, pelo mesmo motivo que explica a acurácia do modelo ser menor. Como houve uma iteração em que o modelo não satisfez 0.8-justiça, as classificações precisaram ser modificadas para aumentar a taxa de justiça até satisfazer a restrição. E, como esperado, a taxa de justiça decresceu para $\lambda = 0.9$ conforme a folga aumentou.

Como o modelo com entrada equivalente ao conjunto de dados *Crime* já possuía altos valores na taxa de justiça, as modificações geradas pela folga são mínimas ou inexistentes. Os resultados se mantêm constantes conforme os parâmetros mudam, como indicam os gráficos

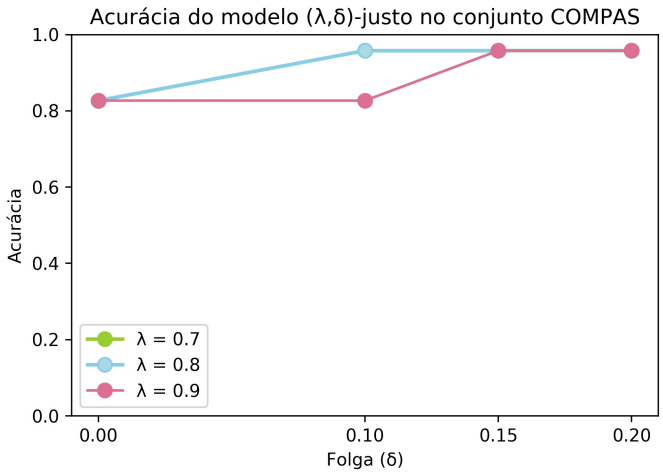


Figura 21 – Acurácia do modelo performando no conjunto de dados COMPAS variando os hiper-parâmetros λ e δ.

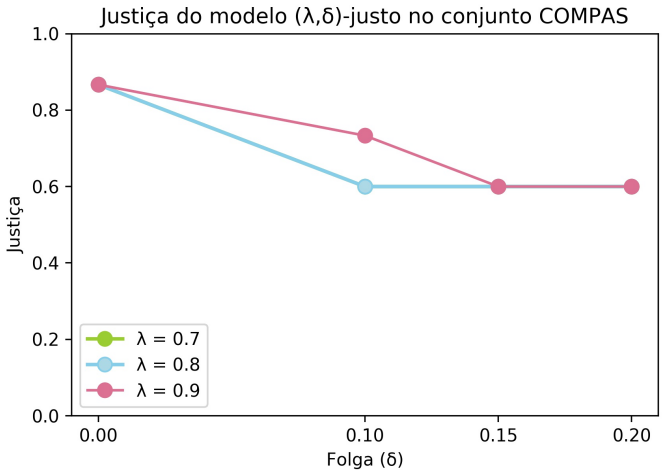


Figura 22 – Justiça do modelo performando no conjunto de dados COMPAS variando os hiperparâmetros λ e δ.

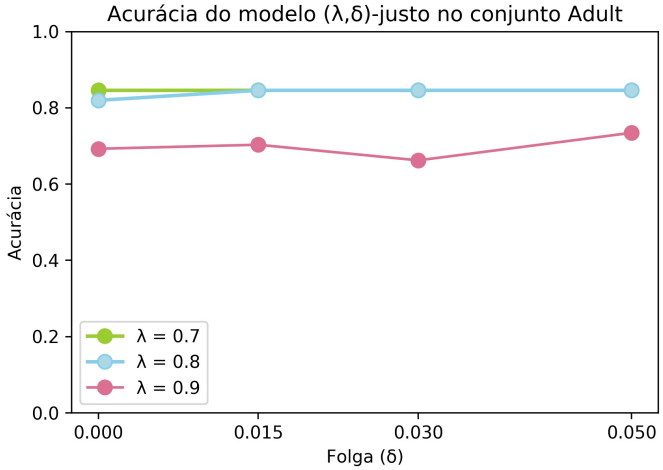


Figura 23 – Acurácia do modelo performando no conjunto de dados *Adult* variando os hiperparâmetros λ e δ.

das figuras 25 e 26, em que as linhas que representam $\lambda = 0.7$, $\lambda = 0.8$ são iguais à linha visível que representa $\lambda = 0.9$.

Com isso, pode-se concluir que o impacto que a variável de folga vai desencadear depende do conjunto de dados. Enquanto no COMPAS houve um grande efeito, no *Adult* não houve grandes decrescimentos na taxa. De modo geral, os resultados cumpriram o que foi proposto, isto é, alavancar as taxas de justiça até um certo limiar definido pelo fornecedor, mantendo altos níveis de acurácia.

De maneira geral, os experimentos mostram que existe um *trade-off* entre ambas as métricas, utilidade e justiça. A ativação da propriedade λ-justiça cumpre o objetivo de providenciar classificações justas baseando-se em um limiar de justiça dado como entrada do algoritmo. A folga da restrição de Lipschitz, utilizada na propriedade relaxada, (λ, δ)-justiça, de

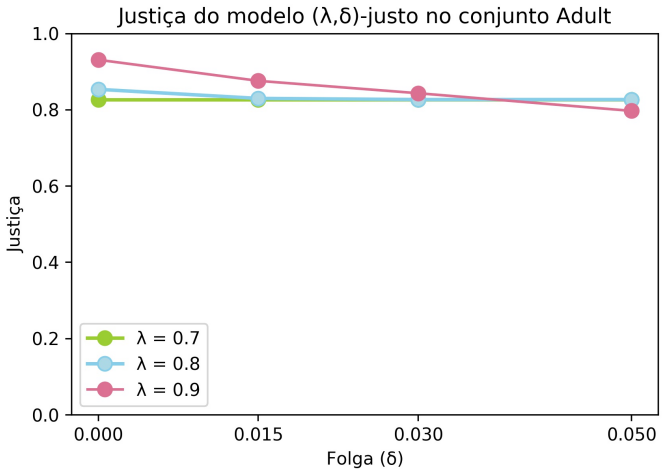


Figura 24 – Justiça do modelo performando no conjunto de dados *Adult* variando os hiper-parâmetros λ e δ.

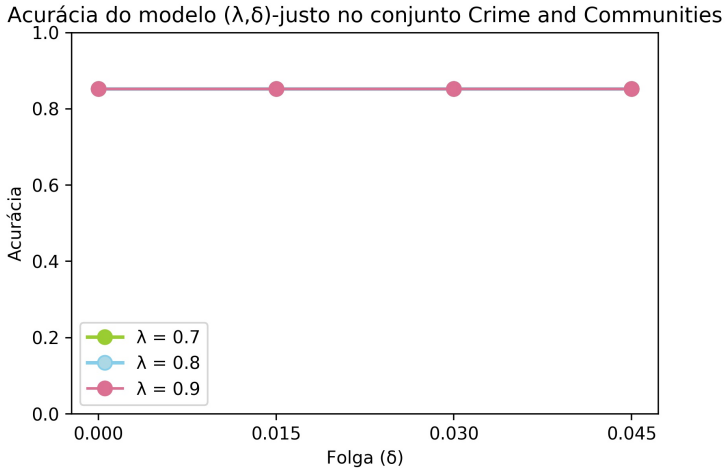


Figura 25 – Acurácia do modelo performando no conjunto de dados *Crime and communities* variando os hiper-parâmetros λ e δ.

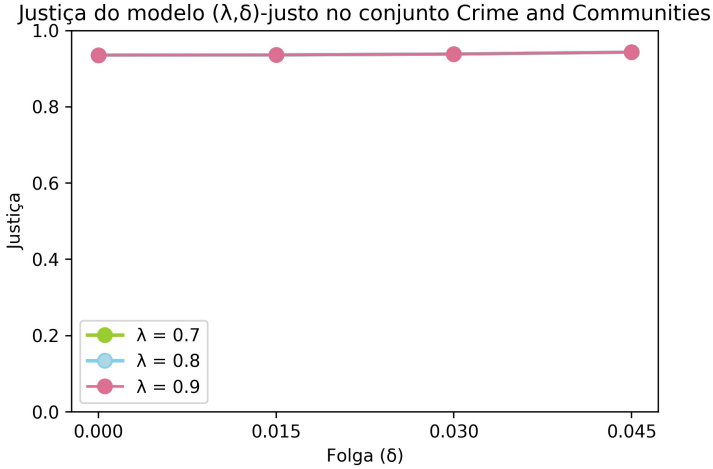


Figura 26 – Justiça do modelo performando no conjunto de dados *Crime and Communities* variando os hiper-parâmetros λ e δ.

fato influencia na qualidade do modelo. Ademais, dependendo do conjunto de entrada o impacto pode ser muito grande, como vemos no exemplo do COMPAS, portanto, o ideal é fazer uma seleção de valores de δ dentro de um pequeno intervalo de forma que não prejudique a utilidade do modelo demasiadamente.

8 CONCLUSÕES E TRABALHOS FUTUROS

Nesta sessão, são apresentados de forma sucinta os resultados obtidos e um fechamento de todo trabalho desenvolvido. A ativação da não-discriminação algorítmica no que diz respeito aos indivíduos de um conjunto é essencial para algoritmos de classificação que tendem a propagar discriminações já existentes nos dados de treinamento. Devido às novas regulamentações destinadas às entidades que mantêm dados de terceiros, é imprescindível que o cuidado sobre os dados e classificações geradas por eles seja redobrado. Com o cenário atual, muitos pesquisadores vêm estudando e construindo técnicas para mitigar o problema de discriminação de indivíduos em classificações automatizadas.

Este trabalho define uma técnica que ativa o que chamamos de justiça para indivíduos (DWORK *et al.*, 2012), equilibrando o *trade-off* existente entre justiça e utilidade. Por meio de um hiper-parâmetro λ, é possível restringir a quantidade mínima de justiça que um modelo deve ter. Definimos a propriedade de λ-justiça e uma relaxação desta nos casos em que for infactível. A propriedade relaxada é chamada de (λ, δ)-justiça, que altera a restrição de Lipschitz, adicionando uma pequena folga δ no limitante superior suficiente para garantir a propriedade sem grandes perdas de justiça. Ademais, é apresentado como as propriedades podem ser aplicadas no modelo de árvore de decisão e os resultados dessa aplicação são apresentados na avaliação experimental.

Os experimentos mostraram que a árvore de decisão λ-justa performa bem em comparação a outro trabalho já existente, chamado FATT (RANZATO *et al.*, 2021), e ao algoritmo convencional de CART. Experimentos adicionais mostraram que (λ, δ)-justiça também cumpre com o objetivo, no entanto, o fornecedor tem um papel crucial para as escolhas dos hiper-parâmetros de acordo com seus interesses, por meio de uma avaliação analítica dos resultados do modelo para um determinado conjunto. Altos valores de δ tendem a prejudicar os níveis de justiça, e o recomendado é utilizar pequenos valores baseados na análise do conjunto de indivíduos.

8.1 Publicações Realizadas

Os artigos científicos seguintes são originados das metodologias propostas:

(i) Silva, Maria LM and Machado e Javam C, 2021. Classificação diferencialmente privada e não discriminatória utilizando árvore de decisão. (SILVA; MACHADO, 2021)

(ii) Silva, Maria LM, Chaves, Iago e Machado, Javam C, 2022. λ-Fair Decision Tree Classification.

O trabalho (i) foi apresentado no WTDBD 2021 (*Workshop* de Teses e Dissertações em Bancos de Dados) e o (ii) está em fase de revisão pelo comitê de programa da conferência internacional EDBT (*International Conference on Extending Database Technology*).

Durante a realização do mestrado, foram ainda publicados os seguintes artigos que, embora não tenham sido diretamente decorrentes dos resultados científicos aqui apresentados, influenciaram sobremaneira a realização desse trabalho:

(i) Silva, Maria LM, Chaves, Iago e Machado, Javam C, 2020. Aplicação de top-k reverso com privacidade sobre os dados públicos de COVID-19 no estado do Ceará. (SILVA *et al.*, 2020)

(ii) Filho, Manuel EB, Silva, Maria LM, Barros, Patricia VS, Mattos, César LC e Machado, Javam C, 2020. Detectando doença de Parkinson - Uma comparação de modelos de aprendizagem de máquina com redução de dimensionalidade diferencialmente privada. (FILHO *et al.*, 2020)

(iii) Silva, Maria LM, Chaves, Iago e Machado, Javam C, 2021. Private reverse top-k algorithms applied on public data of COVID-19 in the state of Ceará. (LOURDES *et al.*, 2021)

Os trabalhos (i) e (ii) foram apresentados no SBBD 2020 (Simpósio Brasileiro de Banco de Dados) e o (iii) é uma extensão de (i) que foi publicado na revista JIDM (*Journal of Information and Data Management*).

8.2 Trabalhos Futuros

Para continuar o estudo, pretendemos ativar as mesmas propriedades para diferentes modelos, tais como *Random Forest* (HO, 1998) e *k-NN* (ALTMAN, 1992), e avaliar os resultados a partir de uma análise comparativa. Além disso, também pretendemos trabalhar em formas de lidar com dados desbalanceados, por meio de criação de dados sintéticos na etapa de pré-processamento. Finalmente, estudaremos a aplicação das nossas definições em algoritmos de aprendizado por reforço.

REFERÊNCIAS

ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. **The American Statistician**, Taylor & Francis, v. 46, n. 3, p. 175–185, 1992.

BAROCAS, S.; HARDT, M.; NARAYANAN, A. Fairness in machine learning. **Nips tutorial**, v. 1, p. 2017, 2017.

BLACK, P. E. **Manhattan distance**. 2019. Disponível em: https://www.nist.gov/dads/HTML/manhattanDistance.html.

BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. A. **Classification and regression trees**. Boca Raton, Flórida, EUA: CRC press, 1984.

CANTRELL, C. D. **Modern mathematical methods for physicists and engineers**. Cambridge, UK: Cambridge University Press, 2000. ISBN 0-521-59827-3.

CENSUS, B. of the. **United States Census Bureau**. 2021. Disponível em: https://www.census.gov/quickfacts/greenecountyalabama.

CERIANI, L.; VERME, P. The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini. **The Journal of Economic Inequality**, Springer, v. 10, n. 3, p. 421–443, 2012.

COUSOT, P.; COUSOT, R. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: **Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**. New York, NY, United States: Association for Computing Machinery, 1977. v. 4, p. 238–252.

DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Disponível em: http://archive.ics.uci.edu/ml.

DWORK, C.; HARDT, M.; PITASSI, T.; REINGOLD, O.; ZEMEL, R. Fairness through awareness. In: **Proceedings of the 3rd innovations in theoretical computer science conference**. New York, NY, United States: Association for Computing Machinery, 2012. p. 214–226.

ESCOVEDO, T. **Machine Learning: Conceitos e Modelos — Parte I: Aprendizado Supervisionado***. 2020. Disponível em: https://tatianaesc.medium.com/machine-learning-conceitos-e-modelos-f0373bf4f445.

European Commission. **Europe fit for the Digital Age: Commission proposes new rules and actions for excellence and trust in Artificial Intelligence**. 2021. Disponível em: https://ec.europa.eu/commission/presscorner/detail/en/IP_21_1682.

FILHO, M. E. B.; SILVA, M. d. L. M.; BARROS, P. V. da S.; MATTOS, C. L. C.; MACHADO, J. de C. Detectando doença de parkinson-uma comparação de modelos de aprendizagem de máquina com redução de dimensionalidade diferencialmente privada. In: SBC. **Anais do XXXV Simpósio Brasileiro de Bancos de Dados**. Porto Alegre, RS, Brasil, 2020. p. 253–258.

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936.

HO, T. K. The random subspace method for constructing decision forests. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 20, n. 8, p. 832–844, 1998.

KÖTHE, G. Topological vector spaces. In: **Topological Vector Spaces I**. New York, NY, EUA: Springer, 1983. p. 123–201.

KULLBACK, S. **Information theory and statistics**. North Chelmsford, Chelmsford, Massachusetts, EUA: Courier Corporation, 1997.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **The annals of mathematical statistics**, JSTOR, v. 22, n. 1, p. 79–86, 1951.

LAHOTI, P.; GUMMADI, K. P.; WEIKUM, G. Operationalizing individual fairness with pairwise fair representations. **Proceedings of the VLDB Endowment**, v. 13, n. 4, 2019a.

LAHOTI, P.; GUMMADI, K. P.; WEIKUM, G. ifair: Learning individually fair data representations for algorithmic decision making. In: IEEE. **35th IEEE international conference on data engineering (ICDE)**. Piscataway, Nova Jersey, EUA, 2019b. p. 1334–1345.

LARSON, J.; ROSWELL, M.; ATLIDAKIS, V. **COMPAS**. [*S. l.*]: ProPublica, 2016. https://github.com/propublica/compas-analysis. May, 2016.

LOHIA, P. K.; RAMAMURTHY, K. N.; BHIDE, M.; SAHA, D.; VARSHNEY, K. R.; PURI, R. Bias mitigation post-processing for individual and group fairness. In: IEEE. **2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)**. Piscataway, Nova Jersey, EUA, 2019. p. 2847–2851.

LOURDES, M. S. Maria de; CHAVES, I. C.; MACHADO, J. C. Private reverse top-k algorithms applied on public data of covid-19 in the state of ceará. **Journal of Information and Data Management**, v. 12, n. 5, 2021.

MALLOWS, C. L. A Note on Asymptotic Joint Normality. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 43, n. 2, p. 508 – 515, 1972. Disponível em: https://doi.org/10.1214/aoms/1177692631.

MITCHELL, T. M. Artificial neural networks. **Machine learning**, McGraw-Hill New York, v. 45, p. 81–127, 1997.

NARAYANAN, A. Translation tutorial: 21 fairness definitions and their politics. In: **Proc. Conf. Fairness Accountability Transp.** New York, NY, USA: ACM, 2018. v. 1170, p. 3.

PITOURA, E.; STEFANIDIS, K.; KOUTRIKA, G. Fairness in rankings and recommenders: Models, methods and research directions. In: IEEE. **2021 IEEE 37th International Conference on Data Engineering (ICDE)**. Piscataway, Nova Jersey, EUA, 2021. p. 2358–2361.

PROPUBLICA. **How We Analyzed the COMPAS Recidivism Algorithm**. 2016. Disponível em: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

PROPUBLICA. **Machine Bias**. 2016. Disponível em: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, n. 1, p. 81–106, 1986.

RANZATO, F.; URBAN, C.; ZANELLA, M. Fairness-aware training of decision trees by abstract interpretation. In: **Proceedings of the 30th ACM International Conference on Information & Knowledge Management**. [*S. l.*: *s. n.*], 2021. p. 1508–1517.

RANZATO, F.; ZANELLA, M. Genetic adversarial training of decision trees. In: **Proceedings of the Genetic and Evolutionary Computation Conference**. Nova York, NY, EUA: Association for Computing Machinery, 2021. p. 358–367.

ROYALL, R. The likelihood paradigm for statistical evidence. **The nature of scientific evidence: Statistical, philosophical, and empirical considerations**, p. 119–152, 2004.

SILVA, M. d. L. M.; CHAVES, I. de C.; MACHADO, J. de C. Aplicação de top-k reverso com privacidade sobre os dados públicos de covid-19 no estado do ceará. In: SBC. **Anais do XXXV Simpósio Brasileiro de Bancos de Dados**. Porto Alegre, RS, Brasil, 2020. p. 193–198.

SILVA, M. L.; MACHADO, J. C. Classificação diferencialmente privada e não discriminatória utilizando árvore de decisão. In: SBC. **Anais Estendidos do XXXVI Simpósio Brasileiro de Bancos de Dados**. Porto Alegre, RS, Brasil, 2021. p. 148–154.

ANEXO A – REGULAMENTAÇÃO DE IAS DA COMISSÃO DA UE

"All AI systems considered a clear threat to the safety, livelihoods and rights of people will be banned, from social scoring by governments to toys using voice assistance that encourages dangerous behaviour." (European Commission, 2021)