

# Privacidade Diferencial em Gradient Boosting Decision Trees com Técnicas de Particionamento para Dados Categóricos

Antonio Gabriel M. Alves<sup>1</sup>, Francisco Lucas F. Pereira<sup>1</sup>,  
Iago C. Chaves<sup>1</sup>, Javam C. Machado<sup>1</sup>

<sup>1</sup>Laboratório de Sistemas e Bancos de Dados (LSBD)  
Departamento de Computação – Universidade Federal do Ceará (UFC)  
60.455-760 – Fortaleza – CE – Brasil

{gabriel.alves, lucas.falcao, iago.chaves, javam.machado}@lsbd.ufc.br

**Abstract.** *Gradient Boosting Decision Trees has achieved state-of-the-art performance in various machine learning tasks. This paper investigates enhancements in handling categorical attributes and random selection of split points while providing differential privacy guarantees. The results include a new gain function for these attributes and the sensitivity bounds for this gain function. Additionally, an empirical analysis on six real-world datasets shows that the proposed approach achieves error rates equal to or lower than the baseline models.*

**Resumo.** *Este artigo propõe uma nova abordagem de particionamento de dados categóricos para aplicar a privacidade diferencial em Gradient Boosting Decision Trees. Nele estudamos aprimoramentos no tratamento de atributos categóricos e seleção aleatória de pontos de particionamento enquanto oferecemos garantias de privacidade diferencial. Nossa abordagem define uma nova função de ganho para esses atributos e determina os limites de sensibilidade dessa função. Além disso, realizamos uma análise empírica em 6 conjuntos de dados reais, mostrando que a abordagem proposta alcança taxas de erro menores ou iguais aos modelos de referência.*

## 1. Introdução

Coletar e analisar informações sobre indivíduos é essencial na sociedade orientada por dados do século XXI. As árvores de decisão são modelos importantes no aprendizado de máquina, que utilizam regras de decisão baseadas nos atributos dos dados de treinamento [Breiman 2017]. Essas regras formam um caminho que leva a um nó folha, que representam a uma resposta do problema. Técnicas modernas de aprendizado de máquina, como florestas aleatórias [Breiman 2001] e Gradient Boosting Decision Trees [Chen and Guestrin 2016], usam árvores de decisão como componentes fundamentais.

*Gradient Boosting Decision Trees* alcançou o estado da arte em diversas tarefas de aprendizado de máquina, tais como previsão de índice de precipitação [Danandeh Mehr 2021] e classificação de páginas *web* [Pennacchiotti and Popescu 2011]. Essa técnica consiste em construir árvores de decisão sequencialmente de forma que cada nova árvore ajusta-se ao erro da anterior. Contudo, apesar do grande potencial de aplicação do modelo, a sua popularização e reconhecimento como um modelo eficiente só ocorreu devido à construção de algoritmos otimizados [Chen and Guestrin 2016].

O processo de aprendizagem automática tem como base o grande volume de dados. Estes dados têm o potencial de revelar informações sensíveis sobre indivíduos. Por exemplo, é possível construir um ataque para modelos de Árvores Aleatórias, visando reconstruir o conjunto de dados usado no treinamento [Ferry et al. 2024]. Assim, é essencial que a privacidade dos indivíduos seja considerada no processo de aprendizagem [Shokri et al. 2017, Truex et al. 2018].

Diante dessa necessidade, a Privacidade Diferencial [Dwork 2006] tem se destacado em múltiplas tarefas de aprendizado automático como em redes neurais [Abadi et al. 2016]. A privacidade diferencial visa garantir que qualquer resultado de uma consulta não revele nenhuma informação sobre um indivíduo em específico [Wood et al. 2018, M. Silva et al. 2020]. Essa técnica se destaca por permitir que grandes conjuntos de dados possam ser utilizados, sem expor totalmente os dados dos contribuidores. Além de oferecer garantias matemáticas para a proteção dos indivíduos.

Dado o uso generalizado e a popularidade dos modelos de *Gradient Boosting Decision Trees*, é essencial garantir a privacidade em suas implementações. Algumas soluções já foram propostas para tratar essa necessidade [Li et al. 2020, Zhao et al. 2018, Liu et al. 2018]. Contudo, essas soluções não lidam diretamente com atributos categóricos e, sendo assim, necessitam de uma etapa de pré-processamento para realizar a codificação. Outra limitação é a necessidade de alocar parte do orçamento para selecionar o ponto de particionamento, levando a um resultado impreciso nos nós folhas quando são utilizados orçamentos baixos.

Este artigo descreve uma abordagem para empregar atributos categóricos e aleatoriedade diretamente na construção das árvores. Suas principais contribuições são:

- Uma nova abordagem para lidar com dados categóricos, com a definição de uma nova função de ganho, cálculo da sensibilidade dessa função e uma nova estrutura para a árvore.
- Uma nova abordagem para lidar com altas restrições de privacidade, preservando a acurácia do modelo.
- Uma análise exaustiva do desempenho do modelo em diferentes conjuntos de dados com diferentes características, mostrando a generalidade da nossa abordagem.

A seguir, serão apresentados os conceitos básicos sobre o modelo de aprendizado, a técnica de privacidade e a técnica de codificação utilizada. Em seguida, discutiremos os trabalhos relacionados. Após isso, será apresentado nosso método juntamente a garantia de privacidade. Por fim, será discutido o desempenho do nosso método em relação ao *DPBoost*.

## 2. Preliminares

Esta seção introduz os conceitos essenciais para a compreensão das soluções propostas. Nela descrevemos noções básicas a respeito do modelo de aprendizado *Gradient Boosting Decision Trees*, sobre a privacidade diferencial e a codificação de atributos categóricos.

### 2.1. O Modelo de Aprendizado

*Gradient Boosting Decision Trees* [Chen and Guestrin 2016] é um *ensemble model* [Opitz and Maclin 1999] que treina uma série de árvores de decisão usando pseudo-residuais, que consistem na diferença entre o valor real e o valor predito em uma etapa

intermediária do treinamento. Sendo assim, os pseudo-residuais originados da árvore anterior são utilizados como entrada para a próxima árvore. Essa estratégia visa minimizar o erro descrito pela seguinte função:  $\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} f_t^2(x_i)] + \Omega(f_m)$  na  $t$ -ésima iteração. Nessa função,  $m$  representa a  $m$ -ésima árvore,  $n$  é o número de instâncias de dados,  $d$  o número de atributos,  $\mathbf{x}_i \in \mathbb{R}^d$  é a  $i$ -ésima instância dos dados,  $\mathbf{y}_i \in \mathbb{R}$  o rótulo associado a  $i$ -ésima instância,  $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$  é o gradiente de primeira ordem sobre a função de custo  $l$  aplicada ao rótulo  $y_i$  e  $\hat{y}$  a predição do modelo [Si et al. 2017].

O algoritmo define como os dados serão particionados conforme a pontuação obtida por uma função de ganho. A ideia é escolher a divisão que melhor separe os gradientes de menor dos de maior magnitude, maximizando assim a função de ganho 1. Além disso, há um fator de regularização ( $\lambda$ ) para evitar o *overfitting* do modelo. Nessa função, é considerado que os dados serão divididos nos conjuntos  $I_L$  e  $I_R$  representando os nós da esquerda e direita, respectivamente.

$$G(I_L, I_R) = \frac{(\sum_{i \in I_L} g_i)^2}{|I_L| + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{|I_R| + \lambda} \quad (1)$$

Ademais, caso o nó não cumpra os requisitos para ser particionado, ele será convertido em um nó folha. O valor desse nó será a média regularizada dos gradientes, conforme a função 2. Para converter um nó em um nó folha, é necessário atingir a profundidade máxima ou o ganho ser menor, ou igual a zero.

$$V(I) = -\frac{\sum_{i \in I} g_i}{|I| + \lambda} \quad (2)$$

Por fim, será aplicado uma taxa de encolhimento  $\eta$  [Friedman 2002] ao valor dos nós folha. O objetivo é evitar que o modelo seja excessivamente influenciado pelas previsões de uma única árvore.

## 2.2. Privacidade Diferencial

Privacidade Diferencial [Dwork 2006] é um padrão de proteção à privacidade de dados que oferece garantias probabilísticas robustas de que um certo resultado não será dependente de um registro específico estar presente nos dados de entrada. Nela, o limite para a perda de privacidade no resultado de uma consulta é calibrado por um parâmetro ( $\epsilon$ ), ao qual será chamado de orçamento de privacidade. De forma que, quanto menor o valor desse parâmetro, menor será a perda de privacidade, ao passo que mais imprecisa será a resposta para a consulta.

**Definição 2.1** ( $\epsilon$ -Privacidade Diferencial). Um mecanismo  $M$  é  $\epsilon$ -diferencialmente privado se para todos os *datasets* vizinhos  $D$  e  $D'$ , onde *datasets* vizinhos são conjuntos de dados que se diferem em apenas um registro; e para todo conjunto  $S$  contido na variação de resultados de  $M$ , isto é, para todo  $S \subset \text{Range}(M)$ , a seguinte condição é satisfeita:  $Pr[M(D) \in S] \leq \exp(\epsilon) \cdot Pr[M(D') \in S]$ .

**Definição 2.2** (Mecanismo de Laplace). Dada uma consulta  $f$ , o mecanismo de Laplace [Dwork 2006] é definido como:  $M_L(x, f, \epsilon) = f(x) + (Y_1, \dots, Y_k)$  tal que  $Y_i$  seja uma variável aleatória que segue a distribuição de Laplace dada por  $Lap(\frac{\Delta f}{\epsilon})$ , onde  $\Delta f =$

$\max \|f(D) - f(D')\|_1$  representa a sensibilidade da consulta  $f$ , mais especificamente a máxima variação de todas as respostas possíveis.

**Definição 2.3** (Mecanismo Exponencial). Seja  $F$  o mecanismo exponencial [McSherry 2009], a probabilidade da resposta  $r \in R$  é proporcional a  $\exp\left(\frac{\epsilon \times u(D,r)}{2 \times \Delta u}\right)$ , onde  $u$  representa uma função de utilidade, ou seja, uma função que responde para cada registro do conjunto de dados, uma pontuação para uma possível resposta  $r \in R$ .  $R$  é um *range* arbitrário que representa os possíveis valores de saída. A sensibilidade  $\Delta u$  é definida como:  $\Delta u = \max_{r \in R} \max_{D, D': \|D - D'\|_1 \leq 1} |u(D, r) - u(D', r)|$

Os mecanismos mencionados antes proveem garantias de privacidade diferencial. Entretanto, diversos algoritmos apresentam vários passos de consulta aos dados, e por isso as seguintes definições mostram como compor diversos mecanismos diferencialmente privados.

**Teorema 2.4** (Composição Sequencial). Assuma que  $M = \{M_1, \dots, M_m\}$  seja uma série de funções executadas sequencialmente sobre os dados. Se  $M_i$  provém  $\epsilon_i$ -Privacidade Diferencial, então  $M$  provém  $\sum_{i=1}^m \epsilon_i$ -Privacidade Diferencial.

**Teorema 2.5** (Composição Paralela). Assuma que  $M = \{M_1, \dots, M_m\}$  seja uma série de funções executadas separadamente sobre subconjuntos disjuntos de um conjunto de dados. Se  $M_i$  provém  $\epsilon_i$ -Privacidade Diferencial, então  $M$  provém  $\max(\epsilon_1, \dots, \epsilon_m)$ -Privacidade Diferencial.

### 2.3. Codificação de Atributos Categóricos

*One-hot encoding* é uma estratégia bastante popular para tornar dados não-numéricos compatíveis com modelos de aprendizado de máquina [Seeger 2018]. O funcionamento desse método se dá mediante uma lista de 1's e 0's que identificam qual dos valores possíveis o atributo assume. Dessa forma, apenas um elemento dessa lista assumirá o valor 1 enquanto todos os outros serão 0. Podemos representar o *one-hot encoding* como um vetor de base canônica  $e$ , por exemplo, se um atributo categórico  $A$  apresenta 3 possíveis valores  $\{a_1, a_2, a_3\}$ , então representamos  $e_{a_1} = [1, 0, 0]$ ,  $e_{a_2} = [0, 1, 0]$  e  $e_{a_3} = [0, 0, 1]$ .

## 3. Trabalhos Relacionados

O principal desafio em aplicar privacidade diferencial no modelo de *Gradient Boosting Decision Trees* é como alocar o orçamento de privacidade entre as árvores, já que a previsão de uma árvore depende da previsão das anteriores. Essencialmente, as soluções propostas operam em duas frentes: (1) buscam otimizar o uso do orçamento de privacidade ( $\epsilon$ ) ao utilizar composição paralela, evitando dividi-lo entre todas as árvores; e (2) permitem que as árvores usem todos os exemplos disponíveis aplicando composição sequencial, ou seja, dividindo o orçamento entre todas as árvores de decisão.

No trabalho de [Liu et al. 2018], o orçamento é dividido igualmente entre todas as árvores. Já na parte interna da árvore, o orçamento é dividido em  $S_i + S_l$  partes. Tal que,  $S_i = \sum_{j=1}^c \frac{1}{c}$  é a fração do orçamento destinado à seleção do ponto de particionamento nos nós internos, e  $S_l = 1$  é destinado aos nós folhas. Ademais, também são definidos os limites de sensibilidade para a escolha do ponto de particionamento e calculo do valor de uma folha em  $c^2$  e 1, respectivamente. No contexto do trabalho,  $c$  é o número de atributos que poderão ser utilizados no treinamento, de forma que  $c$  é menor ou igual a número de atributos totais.

Já no trabalho de [Zhao et al. 2018], conjuntos disjuntos são adotados como alternativa à divisão do orçamento de privacidade. Nesse caso, o orçamento  $\epsilon$  é dividido em  $\epsilon_1$  e  $\epsilon_2$  destinados aos nós internos e folhas, respectivamente. Em relação aos limites de sensibilidade, foram definidos para 4 e 1 para os nós internos e folhas, respectivamente.

O trabalho em [Li et al. 2020], *DPBoost*, propõe limites mais restritos para a sensibilidade das funções 1 e 2 através do conceito de Filtragem de Dados Baseada em Gradiente (FDB) e Recorte Geométrico de Folha (RGF). O FDB consiste em filtrar os elementos cujo módulo do gradiente seja maior que um certo limiar. Já o RGF consiste em calcular o valor aproximado de uma folha caso o resultado seja maior que um certo limiar. Para esse trabalho, os limites de sensibilidade dos nós internos e folhas se baseiam no módulo do maior gradiente ( $g^*$ ), sendo  $3g^{*2}$  e  $\frac{g^*}{1+\lambda}$ .

As abordagens descritas não lidam diretamente com dados categóricos, sendo necessária uma etapa de pré-processamento para realizar a codificação. Além disso, essas soluções sempre consideram usar parte do orçamento para selecionar um bom particionamento, o que consequentemente pode gerar um resultado mais impreciso no valor das folhas em situações onde o orçamento total é pequeno. Propomos então o *PrivCatBoost* e o *PrivRandBoost* para tratar as limitações dos trabalhos anteriores em relação a atributos categóricos e otimização do uso do orçamento, respectivamente.

A Tabela 1 relaciona os seguintes critérios de comparação entre as abordagens: (1) sensibilidade da seleção do ponto de particionamento (Sensibilidade Nós Internos); (2) sensibilidade do cálculo do valor do nó folha (Sensibilidade Nós Folha); (3) orçamento destinado aos nós internos ( $\epsilon$  Nós Internos); (4) orçamento destinado aos nós folhas ( $\epsilon$  Folhas); (5) Se tratam diretamente os atributos categóricos (Categórico?). Considere  $g^* = 1$ ,  $\lambda \in \mathbb{R}^+$ ,  $h$  é a profundidade máxima da árvore,  $d$  é o número de atributos.

**Tabela 1. Comparação entre as Abordagens Anteriores e a Proposta.**

	[Zhao et al. 2018]	[Liu et al. 2018]	[Li et al. 2020]	PrivCatBoost	PrivRandBoost
(1) Sensibilidade Nós Internos	4	$c^2$	$3g^{*2}$	$3g^{*2}$	–
(2) Sensibilidade Nós Folha	1	1	$g^*/1+\lambda$	$g^*/1+\lambda$	$g^*/1+\lambda$
(3) $\epsilon$ Nós Internos	$\epsilon/h$	$\epsilon * S_i / (S_i)$	$\epsilon/2 * h$	$\epsilon/2 * h$	–
(4) $\epsilon$ Folhas	$\epsilon_2$	$\epsilon/S_i+1$	$\epsilon/2$	$\epsilon/2$	$\epsilon$
(5) Categórico?	<b>X</b>	<b>X</b>	<b>X</b>	✓	✓

#### 4. PrivCatBoost e os Atributos Categóricos

Propomos o PrivCatBoost<sup>1</sup>, um algoritmo de *Gradient Boosting Decision Trees* diferencialmente privado, para lidar com a limitação dos modelos anteriores, em particular com o tratamento de atributos categóricos. Para isso, PrivCatBoost utiliza uma nova função de ganho que permite avaliar a qualidade de um particionamento feito sobre esses atributos e, além disso, uma nova possibilidade de estrutura para as árvores.

Nossa solução baseia-se na abordagem do *DPBoost* [Li et al. 2020], a qual é provada ser  $\epsilon$ -diferencialmente privado. Os algoritmos 1 e 2 ilustram o funcionamento do DPBoost. Essencialmente, a privacidade será aplicada em dois pontos: (1) seleção de um ponto de particionamento (linha 7); (2) cálculo do valor nó (linha 13) no algoritmo 2. O DPBoost utiliza o mecanismo exponencial para inserir aleatoriedade na escolha de um

<sup>1</sup><https://github.com/Magalhaes-Alves/PrivCatBoost>

ponto de particionamento de forma que não seja totalmente previsível qual será o ponto escolhido. Além disso, o algoritmo aplica o mecanismo de laplace no valor das folhas. Portanto, via composição sequencial e paralela dos mecanismos utilizados, o algoritmo satisfaz a definição da privacidade diferencial.

---

**Algoritmo 1:** Treino DPBoost
 

---

**Entrada:**  $\mathcal{D}$ : Dataset,  $Profundidade_{max}$ : profundidade máxima,  $\epsilon$ : orçamento de privacidade,  $\lambda$ : parâmetro de regularização,  $T$ : número total de árvores,  $T_e$ : número de árvores em um *ensemble*

**Saída:** Gradient Boosting Decision Trees Diferencialmente Privado

```

1 início
2    $N_e \leftarrow \lceil \frac{T}{T_e} \rceil, \epsilon_e \leftarrow \frac{\epsilon}{N_e};$ 
3   para  $t \leftarrow 1$  até  $T$  faça
4      $t_e \leftarrow t \bmod T_e;$ 
5     se  $t_e = 1$  então
6        $I \leftarrow \mathcal{D};$ 
7     fim
8     SELECIONE ALEATORIAMENTE  $\left( \frac{|\mathcal{D}| \eta (1-\eta)^{t_e}}{1-(1-\eta)^{T_e}} \right)$  INSTÂNCIAS DO
9     CONJUNTO  $I$  E CONSTRUA O CONJUNTO  $I_t;$ 
10     $I \leftarrow I - I_t;$ 
11    Treinar_árvore
12     $\left( I_t, Profundidade_{max}, \epsilon_e, 3g_t^{*2}, \min \left( \frac{g_t^*}{1+\lambda}, 2g_t^* (1-\eta)^{t-1} \right) \right);$ 
13  fim
14 fim

```

---

#### 4.1. Tratamento para Atributos Categóricos

Como visto na Seção 2, o cálculo do ganho no *DPBoost* (equação 1) considera apenas o particionamento do conjunto de dados em exatamente dois subconjuntos disjuntos. Isso ocorre, pois as regras inferidas pela árvore se baseiam em comparações numéricas entre os valores dos atributos com um certo ponto de particionamento, processo utilizado na divisão entre os ramos da árvore. Portanto, são observadas duas limitações: uma única forma de particionar os dados e a necessidade que os dados sejam numéricos [Dahouda and Joe 2021].

**Limitações DPBoost:** Diante da necessidade de lidar com dados categóricos, foram propostos métodos para codificar esse tipo de atributo e tornar os dados compatíveis com o DPBoost. Entre esses métodos, o mais utilizado é o *one-hot encoding* (OHE) [Seger 2018]. No entanto, esse método pode gerar matrizes grandes e esparsas, afetando o tempo de treinamento do modelo, além do grande uso de memória para armazenar essa matriz de representação [Chollet 2021]. Portanto, para evitar a necessidade de utilizar representações com matrizes esparsas e de uma etapa de pré-processamento dos dados categóricos, propomos: (1) uma função para avaliar o quão bom é realizar o particionamento usando um atributo categórico; (2) uma nova estrutura para a árvore.

**Definindo uma Função de Ganho para Atributos Categóricos:** Nessa nova abordagem, propomos agrupar os dados conforme o valor do atributo selecionado para realizar

---

**Algoritmo 2:** Treinar Árvore: Treina uma árvore de decisão diferencialmente privada

---

**Entrada:**  $I$ : Dados para Treinamento,  $Profundidade_{max}$ : profundidade máxima,  $\epsilon_t$ : orçamento de privacidade,  $\Delta G$ : Sensibilidade Função de Ganho,  $\Delta V$ : sensibilidade do calculo da folha

**Saída:** Árvore  $\epsilon_t$ -Diferencialmente Privada

```

1 REALIZA A FILTRAGEM DE DADOS BASEADO NO GRADIENTE;
2  $\epsilon_{folha} \leftarrow \frac{\epsilon_t}{2}$ ;
3  $\epsilon_{nao_folha} \leftarrow \frac{\epsilon_t}{2 * profundidade_{max}}$ ;
4 para CADA PONTO DE PARTICIONAMENTO, EM TODOS OS NÓS DE UM
  NÍVEL E EM TODOS OS NÍVEIS faça
5   COMPUTE O GANHO CONFORME A FUNÇÃO 1;
6    $P_i \leftarrow \exp\left(\frac{\epsilon_{nao_folha} G_i}{2\Delta G}\right)$ ;
7   SELECIONAR UM PONTO  $s$  COM PROBABILIDADE  $\frac{P_s}{\sum_i P_i}$ ;
8   PARTICIONE O NÓ ATUAL DE ACORDO O VALOR  $s$  DO ATRIBUTO;
9 fim
10 para CADA NÓ FOLHA CRIADO faça
11   CALCULE O VALOR DA FOLHA  $V_i$  CONFORME A FUNÇÃO 2;
12   REALIZA O RECORTE GEOMÉTRICO DA FOLHA  $V_i$ ;
13    $V_i \leftarrow V_i + Lap\left(\frac{\Delta V}{\epsilon_{folha}}\right)$ ;
14 fim

```

---

o particionamento. Por exemplo, suponha que um certo atributo possua  $k$  valores distintos e esse atributo foi selecionado para realizar o particionamento dos dados. Logo, serão criados  $k$  nós e cada nó receberá um subconjunto dos dados com um dos  $k$  valores possíveis.

Contudo, para que esse particionamento seja selecionado pelo modelo é necessário um ganho associado. Para isso, definimos uma função de ganho baseado nos agrupamentos estabelecidos anteriormente. Suponha um conjunto de dados  $I$ , um atributo  $A$  e um valor  $k$  que representa o número de valores possíveis para  $A$ . Após agrupar o conjunto  $I$  baseado no atributo  $A$ , obteremos  $k$  conjuntos disjuntos,  $C_1, C_2, \dots, C_k$ , tal que  $I = \bigcup_{j=1}^k C_j$ . A função 3 recebe esse  $k$  conjuntos e calcula o somatório das  $k$  médias regularizadas dos gradientes em cada partição.

$$G_{cat}(C_1, \dots, C_k) = \sum_{j=1}^k \frac{(\sum_{i \in C_j} g_i)^2}{|C_j| + \lambda} \quad (3)$$

**Estendendo os Limites de Sensibilidade para Particionamento Categórico:** Propomos a função 3 para avaliar a qualidade de um particionamento categórico. Contudo, para utilizar essa nova função no mecanismo exponencial, é preciso calcular o quão o resultado é influenciado pela presença ou ausência de um indivíduo, ou seja, a sensibilidade da função. Abaixo está proposto o limite de sensibilidade para a função 3.

**Lema 4.1.** Assuma  $g^* = \max_{i \in \mathcal{D}} |g_i|$ , então  $\Delta G_{cat} \leq 3g^{*2}$

*Demonstração.* Considere dois conjuntos adjacentes que diferem em uma única instância, ou seja,  $I_1 = \{\mathbf{x}_i\}_{i=1}^n$  e  $I_2 = I_1 \cup x_s$ . Assuma que  $I_1 = \bigcup_{j=1}^k C_j$ , onde cada  $C_j$  representa os dados particionados do conjunto  $I_1$  baseado em um atributo categórico  $A$ . A prova será dividida em dois casos:

1.  $x_s \in C_j$  para algum  $j$  tal que  $1 \leq j \leq k$ .
2.  $x_s \notin C_j$  para algum  $j$  tal que  $1 \leq j \leq k$ .

1.  $x_s \in C_j$  para algum  $j$  tal que  $1 \leq j \leq k$ : Será usado  $n_j$  para representar  $|C_j|$ . Como no conjunto  $I_2$ , apenas um dos elementos é da forma,  $C_j \cup x_s$  então todos os outros subconjuntos são iguais em  $I_1$  e  $I_2$ , logo a sensibilidade para esses conjuntos será 0. Dessa forma, será calculado a sensibilidade para o único conjunto que difere entre  $I_1$  e  $I_2$

$$\begin{aligned} \Delta G_{cat} &= \left| \frac{(\sum_{i \in C_j} g_i + g_s)^2}{n_j + \lambda + 1} - \frac{(\sum_{i \in C_j} g_i)^2}{n_j + \lambda} \right| \\ &= \left| \frac{(n_j + \lambda)g_s^2 + 2(n_j + \lambda)g_s \sum_{i \in C_j} g_i - (\sum_{i \in C_j} g_i)^2}{(n_j + \lambda + 1)(n_j + \lambda)} \right| \end{aligned} \quad (4)$$

Assuma  $h(g_s, \sum_{i \in C_j} g_i) = (n_j + \lambda)g_s^2 + 2(n_j + \lambda)g_s \sum_{i \in C_j} g_i - (\sum_{i \in C_j} g_i)^2$ . Suponha  $\partial_{g_s} h = 0$  e  $\partial_{\sum_{i \in C_j} g_i} h = 0$ , temos  $g_s = 0$  e  $\sum_{i \in C_j} g_i = 0$ . Comparando os pontos estacionários e os limites (isto é,  $g_s = \pm g^*$  e  $\sum_{i \in C_j} g_i = n_j g^*$ ), é possível encontrar quando  $g_s = -g^*$ ,  $\sum_{i \in C_j} g_i = n_j g^*$  ( $g_s = g^*$ ,  $\sum_{i \in C_j} g_i = -n_j g^*$ ),  $n_j \rightarrow \infty$ , a equação alcança seu máximo. Tem-se então:

$$\begin{aligned} \Delta G_{cat} &= \left| \frac{(n_j - 1)^2 g^{*2}}{n_j \lambda + 1} - \frac{n_j^2 g^{*2}}{n_j + \lambda} \right| \\ &= \left| \frac{-3n_j^2 + (1 - 2\lambda)n_j + \lambda}{n_j^2 + (2\lambda + 1)n_j + \lambda(\lambda + 1)} \right| g^{*2} \\ &\leq 3g^{*2} \end{aligned} \quad (5)$$

2.  $x_s \notin C_j$  para algum  $j$  tal que  $1 \leq j \leq k$ . Suponha que o novo registro  $x_s$  não pode ser agrupado junto a nenhum dos registros existentes, logo o registro será alocado no conjunto  $I_{k+1}$ . Sendo assim, como  $C_{k+1} \notin I_1$ . Para a demonstração abaixo  $\lambda \in \mathbb{R}^+$  e  $g^{*2}$  é sempre positivo.

$$\begin{aligned} \Delta G_{cat} &= \left| \frac{(\sum_{i \in I_{k+1}} g_i)^2}{n_{k+1} + \lambda} \right| = \left| \frac{g_s^2}{n_{k+1} + \lambda} \right| \\ &\leq \left| \frac{1}{n_{k+1} + \lambda} \right| g^{*2} = \left| \frac{1}{1 + \lambda} \right| g^{*2} \leq g^{*2} \leq 3g^{*2} \end{aligned} \quad (6)$$

□

Portando, como  $\Delta G_{cat} = \Delta G \leq 3g^{*2}$ , então o ruído necessário para escolher o ponto de particionamento é equivalente ao  $\Delta G$ . Assim, temos o necessário para

utilizar essa nova função de ganho no mecanismo exponencial. Como o DPBoost é  $\epsilon$ -diferencialmente privado e o limite da sensibilidade da função 3 é equivalente à função 1, então, ao alterar como o ganho é calculado, PrivCatBoost também é  $\epsilon$ -diferencialmente privado. Para implementar essa funcionalidade, será utilizado a função 1 para atributos numéricos e para atributos categóricos será usado a nova abordagem.

**Solução Baseada em Florestas Aleatórias: PrivRandBoost** *Gradient Boosting Decision Trees* que incorporam privacidade diferencial utilizam o orçamento de privacidade em dois momentos principais: (1) na seleção do ponto de particionamento e (2) no cálculo dos valores das folhas. Ademais, o orçamento destinado à seleção do ponto de particionamento é dividido pela profundidade máxima da árvore. Isso resulta em frações pequenas de orçamento disponíveis para cada consulta. Levando a resultados que se afastam dos valores reais. No contexto da seleção de ponto de particionamento, esse afastamento se traduz na escolha de divisões que não separam adequadamente os gradientes.

Para lidar com essa limitação, propomos selecionar o ponto de particionamento de forma aleatória [Bojarski et al. 2014], uma estratégia adotada em Florestas Aleatórias [Breiman 2001]. Como a seleção do ponto de particionamento será aleatória, não é necessário calcular o ganho nos nós internos e, portanto, não é necessário alocar parte do orçamento de privacidade para eles. Consequentemente, esse orçamento não gasto será destinado aos nós folha para calcular seu valor (Função 2). Este método também trata atributos categóricos, utilizando apenas a estrutura de particionamento definida na subseção 4.1, sem calcular o ganho. Para tal, não será utilizado o mecanismo exponencial para selecionar o ponto de particionamento, não beneficiando os pontos que possuem ganhos maiores.

## 5. Experimentos

Nessa seção, será avaliado o desempenho da nossa abordagem, em particular a seleção aleatória e o particionamento em categorias, contra o nosso principal competidor [Li et al. 2020] descrito na seção 3. A forma de comparação será usando as seguintes abordagens:

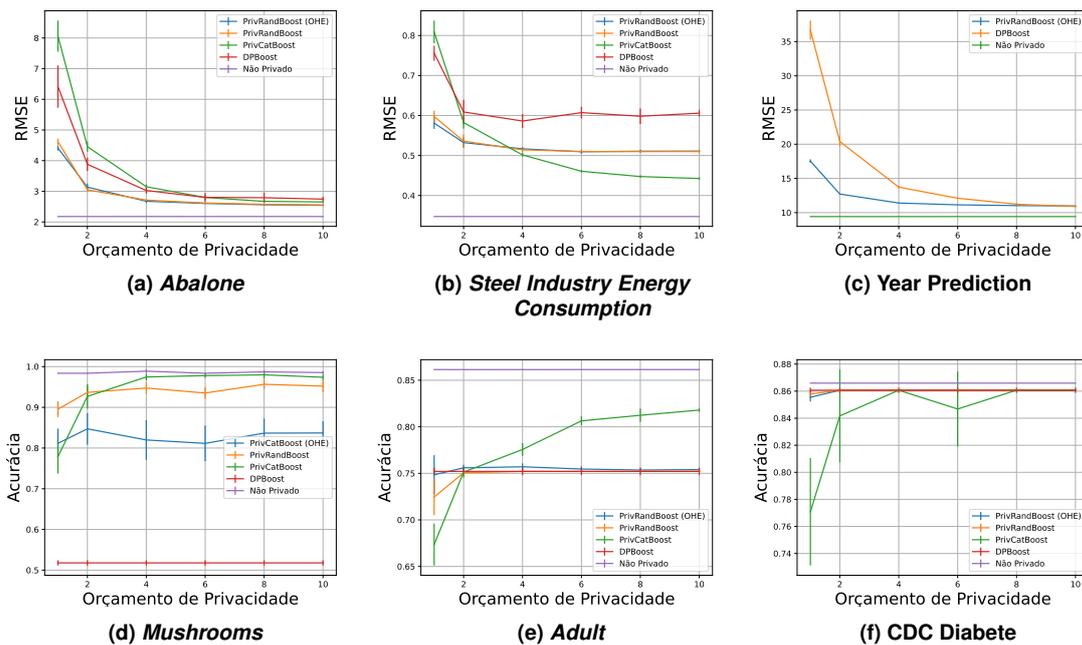
1. PrivCatBoost: *DPBoost* com tratamento de atributos categóricos.
2. PrivRandBoost (OHE): *DPBoost* com seleção aleatória sem particionamento categórico.
3. PrivRandBoost: *DPBoost* com seleção aleatória com particionamento categórico.
4. *DPBoost* (OHE): Implementação original utilizando *one-hot encoding*.
5. Não-Privado: Implementação *scikit-learn* do *Gradient Boosting Decision Trees*.

Os experimentos fizeram uso exaustivo de seis conjuntos de dados com diferentes características relacionados na Tabela 2. Há conjuntos de dados com poucos ou nenhum atributo categórico como *Year Prediction* e *Abalone*. Conjuntos cujos atributos são todos categóricos, *Mushrooms*. E por fim conjuntos com a proporção mais equilibrada entre categóricos e numéricos, como *adult*, *Steel Industry Energy Consumption*, *CDC Diabete*.

Para a avaliação dos modelos, reportamos a Raiz do Erro Quadrático Médio (RMSE) para as tarefas de regressão e a acurácia para as tarefas de classificação, cujos resultados podem ser visualizados na Figura 1. Todos os modelos tiveram seus hiperparâmetros padronizados, com a taxa de aprendizado definida em 0,05, a profundidade

**Tabela 2. Descrição dos Conjuntos de Dados.**

Dados	#Instâncias	#Numéricas	#Categóricas	Tarefa
<i>Abalone</i>	4176	7	1	Regressão
<i>Steel Industry Energy Consumption</i>	35040	7	2	
<i>Year Prediction MSD</i>	515345	90	0	
<i>Mushrooms</i>	5644	0	22	Classificação
<i>Adult</i>	45170	5	8	
<i>CDC Diabetes</i>	253680	19	3	



**Figura 1. Comparação entre as Abordagens em Relação ao Orçamento.**

máxima estabelecida em 6, e o parâmetro de regularização ( $\lambda$ ) fixado em 0,1. Nos modelos baseados em *DPBoost*, o número total de árvores e o número de árvores em cada *ensemble* foram fixados em 50. Tanto para as tarefas de regressão quanto para as de classificação, os rótulos foram escalonados para o intervalo  $[-1,1]$  antes do treinamento. Para a classificação binária, o gradiente foi inicializado como o gradiente de segunda ordem para o valor constante 0. A avaliação dos modelos foi realizada utilizando *5-fold cross-validation*. Ademais, foi utilizado o método mais comum de codificação, *one-hot encoding*, para os modelos que não processam atributos categóricos nativamente.

### 5.1. Avaliando os Resultados

Os testes realizados em conjuntos de dados com poucos ou nenhum atributo categórico, *Abalone* – Figura 1(a) e *YearPrediction* – Figura 1(c), indicaram que a seleção aleatória gera um erro menor em comparação com as abordagens não-aleatórias como o *DPBoost*, na tarefa de regressão. Contudo, os testes realizados sobre *Steel Industry Energy Consumption* – Figura 1(b) mostram que, a partir de um certo  $\epsilon$ , *PrivCatBoost* obtém resultados melhores e com baixo desvio padrão. Como *Steel Industry Energy Consumption* possui atributos categóricos com alta cardinalidade, ele se beneficia de não utiliza *one-*

*hot encoding*. Além disso, selecionar um bom particionamento com certa probabilidade se mostrou superior à seleção nesse caso.

Por outro lado, em conjuntos com mais atributos categóricos ou com alta cardinalidade, como *Adult* – Figura 1(e) e *Mushrooms* – Figura 1(d), observam-se comportamentos variados. Analisando os resultados obtidos com o conjunto de dados *mushrooms*, observa-se que os métodos que usaram *one-hot encoding* para codificar os dados (PrivRandBoost (OHE), *DPBoost*) obtiveram modelos com baixa acurácia quando comparadas as que tratam atributos categóricos diretamente (PrivCatBoost, PrivRandBoost).

Esse comportamento ocorre devido às altas dimensões geradas pelo *one-hot encoding*, prejudicando a generalização da árvore. Logo, como PrivCatBoost e PrivRandBoost tratam esses atributos diretamente, então eles evitam o problema de altas dimensões. Os experimentos com o conjunto de dados *Adult* – Figura 1(e), que possui uma quantidade mais equilibrada de atributos categóricos e numéricos, também demonstraram um aumento na acurácia ao utilizar parte do orçamento para selecionar o ponto de particionamento e permitir particionamento categórico (PrivCatBoost). As abordagens aleatórias (PrivRandBoost (OHE), PrivRandBoost) obtiveram resultados semelhantes ao *DPBoost*.

Por fim, as abordagens aleatórias (PrivRandBoost, PrivRandBoost (OHE)) se mostraram superiores que a abordagem que usa ganho e atributos categóricos (PrivCatBoost), obtendo resultados semelhantes ao *DPBoost* em *CDC Diabetes* – Figura 1(f), devido à predominância de atributos numéricos. Um ponto importante a ser observado é que para orçamentos muito pequenos ( $\leq 2$ ), abordagens aleatórias performam melhor que não-aleatórias na maioria dos testes.

## Agradecimentos

Esta pesquisa foi financiada pela Lenovo Brasil como parte dos seus investimentos em P&D no contexto da Lei de Informática.

## 6. Conclusão

Neste trabalho propomos uma nova função de ganho e estrutura para a árvore, além de propor a alocação mais eficiente para o orçamento quando este possui valores pequenos. Também descrevemos os nossos principais competidores, realizamos uma implementação do trabalho descrito em [Li et al. 2020] e a usamos para comparar experimentalmente nossa abordagem por meio de seis conjuntos de dados diferentes. Nossa abordagem aleatória (PrivRandBoost (OHE) e PrivRandBoost) se mostrou eficiente em situações onde a maioria dos atributos são numéricos, obtendo resultados melhores que o *DPBoost*. Já quando há mais atributos categóricos, PrivCatBoost obteve resultados melhores.

Neste artigo, comparamos a nossa abordagem com o método *one-hot encoding*, todavia entendemos ser um trabalho futuro relevante a avaliação comparativa com outros métodos de codificação. Além disso, planejamos estudar novas estratégias de seleção de pontos de particionamento em algoritmos de *Gradient Boosting Decision Trees*, pois acreditamos que desta forma poderemos alcançar resultados de acurácia ainda melhores.

## Referências

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.
- Bojarski, M., Choromanska, A., Choromanski, K., and LeCun, Y. (2014). Differentially- and non-differentially-private random decision trees. *arXiv preprint arXiv:1410.6973*.
- Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.
- Breiman, L. (2017). *Classification and regression trees*. Routledge.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.
- Dahouda, M. K. and Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9:114381–114391.
- Danandeh Mehr, A. (2021). Drought classification using gradient boosting decision tree. *Acta Geophysica*, 69(3):909–918.
- Dwork, C. (2006). Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.
- Ferry, J., Fukasawa, R., Pascal, T., and Vidal, T. (2024). Trained random forests completely reveal your dataset. *arXiv preprint arXiv:2402.19232*.
- Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378.
- Li, Q., Wu, Z., Wen, Z., and He, B. (2020). Privacy-preserving gradient boosting decision trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 784–791.
- Liu, X., Li, Q., Li, T., and Chen, D. (2018). Differentially private classification with decision tree ensemble. *Applied Soft Computing*, 62:807–816.
- M. Silva, M. d. L., C. Chaves, I., and C. Machado, J. (2020). Private reverse top-k algorithms applied on public data of covid-19 in the state of ceará. *Journal of Information and Data Management*, 12(5).
- McSherry, F. D. (2009). Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *ACM SIGMOD Int. Conf. on Management of data*, pages 19–30.
- Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198.
- Pennacchiotti, M. and Popescu, A.-M. (2011). A machine learning approach to twitter user classification. In *Proceedings of the international AAAI conference on web and social media*, volume 5, pages 281–288.
- Seger, C. (2018). An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.

- Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.
- Si, S., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., and Hsieh, C.-J. (2017). Gradient boosted decision trees for high dimensional sparse output. In *International conference on machine learning*, pages 3182–3190. PMLR.
- Truex, S., Liu, L., Gursoy, M. E., Yu, L., and Wei, W. (2018). Towards demystifying membership inference attacks. *arXiv preprint arXiv:1807.09173*.
- Wood, A., Altman, M., Bembenek, A., Bun, M., Gaboardi, M., Honaker, J., Nissim, K., O’Brien, D. R., Steinke, T., and Vadhan, S. (2018). Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209.
- Zhao, L., Ni, L., Hu, S., Chen, Y., Zhou, P., Xiao, F., and Wu, L. (2018). Inprivate digging: Enabling tree-based distributed data mining with differential privacy. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 2087–2095. IEEE.