Classificação de Recorrência do Câncer de Tireoide utilizando Curvas Principais

Bruno da Silva Macêdo
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras-MG, Brasil
bruno.macedo2@estudante.ufla.br

Lívia Marques Rodrigues
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras-MG, Brasil
livia.rodrigues3@estudante.ufla.br

Giovanna Gouvea Spuri de Miranda
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras-MG, Brasil
giovanna.s.miranda7@gmail.com

Fernado Elias de Melo Borges
Departamento de Engenharia Agrícola
Universidade Federal de Lavras (UFLA)
Lavras-MG, Brasil
fernandoelias.mb@gmail.com

Camila Martins Saporetti
Departamento de Modelagem Computacional
Universidade do Estado do Rio de Janeiro (UERJ)
Nova Friburgo-RJ, Brasil
camila.saporetti@iprj.uerj.br

Bruno Henrique Groenner
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras-MG, Brasil
brunohb@ufla.br

Danton Diego Ferreira
Departamento de Automática
Universidade Federal de Lavras (UFLA)
Lavras-MG, Brasil
danton@ufla.br

***Resumo*—Este trabalho investiga a aplicação do algoritmo de Curvas Principais (CP), uma técnica não linear que representa dados multidimensionais por meio de curvas suaves unidimensionais, na classificação da recorrência do câncer de tireoide após terapia com iodo radioativo. Considerando o desafio da otimização dos hiperparâmetros de CP, que impactam diretamente sua capacidade de modelagem e desempenho, foi utilizado o método *Grid-Search* para ajuste desses parâmetros. Utilizando o conjunto de dados público *"Thyroid Cancer Recurrence"*, o estudo compara o CP com classificadores tradicionais, como *Naïve Bayes* (NB) e *Support Vector Machines* (SVM), considerando dados desbalanceados e balanceados com a técnica SMOTE. O método CP apresentou desempenho competitivo, alcançando acurácia de 0,937 e coeficiente *kappa* de 0,829 em dados desbalanceados, resultados comparáveis ao SVM. E para os dados balanceados com a abordagem SMOTE ele obteve uma acurácia de 0,917 e um *kappa* de 0,832. A otimização dos hiperparâmetros foi fundamental para aprimorar a capacidade preditiva das CP, destacando seu potencial como uma ferramenta para classificação de dados biomédicos. A análise estatística não evidenciou diferenças significativas entre CP e SVM, confirmando o CP como uma alternativa promissora para suporte à decisão clínica em relação à classificação da recorrência do câncer de tireoide.**

***Palavras-chave*—Inteligência Artificial, Aprendizado de Máquina, Curvas Principais, k-segmentos, *Grid-Search*.**

Agradecimentos a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

I. Introdução

O câncer de tireoide representa uma das neoplasias endócrinas mais comuns, com incidência global em constante crescimento nas últimas décadas. De acordo com dados epidemiológicos, a taxa de incidência aumentou de 4,9 para 14,3 casos por 100.000 habitantes entre 1975 e 2018, o que representa um aumento de quase 200% [1]. Este aumento pode ser parcialmente atribuído ao aprimoramento das técnicas diagnósticas e ao maior acesso a exames de imagem, mas também reflete um aumento real na prevalência da doença [2].

O tratamento inicial geralmente envolve tireoidectomia total ou parcial, frequentemente seguida de terapia com iodo radioativo (RAI) para ablação de tecido tireoidiano residual e tratamento de possíveis micrometástases [3]. Apesar do prognóstico geralmente favorável para a maioria dos pacientes com câncer de tireoide, entre 15-30%, apresentam recorrência da doença após o tratamento inicial [2]. Estas recorrências podem ocorrer localmente, em linfonodos regionais ou como metástases à distância, e são frequentemente detectadas meses ou anos após o tratamento primário [3]. Fatores de risco tradicionalmente associados à recorrência incluem idade avançada, sexo masculino, tamanho tumoral, extensão extratireoidiana, presença de metástases linfonodais ou à distância no momento do diagnóstico, e variantes histológicas mais agressivas [4]. Os sistemas convencionais de classificação da progressão tumoral, como o Tumor-Nódulo-Metástase (TNM) e o sistema Metástase, Idade, Completude da ressecção, Invasão, Tamanho (MACIS), têm sido empregados para categorizar pacientes de

acordo com o risco de mortalidade relacionada à doença. [2]. No entanto, esses sistemas apresentam limitações significativas na predição de recorrência, o que evidencia a necessidade de novas abordagens mais precisas [4].

Neste contexto, o Aprendizado de Máquina (ML) emerge como uma ferramenta promissora para identificação de padrões complexos em dados médicos multidimensionais. A capacidade dos algoritmos de ML de integrar e analisar simultaneamente múltiplas variáveis clínicas, patológicas, moleculares e radiológicas tem revolucionado a medicina personalizada [5]. Estudos recentes têm explorado a aplicação de ML para predição de malignidade em nódulos tireoidianos, classificação de subtipos histológicos e estimativa de prognóstico [4]. Wang et al. [6] desenvolveram cinco algoritmos de ML para predizer a recorrência estrutural em pacientes com câncer papilar de tireoide, onde o modelo SVM obteve o melhor desempenho com acurácia de 87,5%, seguido pelo *XGBoost* com 83,5% e pelo *Random Forest* com 77,5%.

No entanto, um dos principais desafios na aplicação de ML em contextos clínicos é o frequente desbalanceamento de classes nos conjuntos de dados. No caso específico do câncer de tireoide, a baixa proporção de recorrências em relação aos casos sem recidiva resulta em conjuntos de dados altamente assimétricos [7]. Quando algumas classes têm significativamente mais exemplos que outras, os modelos tendem a favorecer as classes majoritárias, comprometendo a detecção das classes minoritárias – que frequentemente são as de maior interesse clínico, como as recorrências tumorais [7], [8].

Para mitigar este problema, algumas estratégias têm sido propostas, como a técnica de *Oversampling* (aumento da classe minoritária) [7]. A técnica *Synthetic Minority Oversampling Technique* (SMOTE) que utiliza a estratégia de *Oversampling*, gerando exemplos sintéticos da classe minoritária, tem demonstrado eficácia superior em diversos cenários clínicos [7], [8]. Clark et al. [9] aplicaram SMOTE nos modelos para prever recorrência de câncer de tireoide, obtendo melhorias significativas no recall (+8,77%), F1 score (+6,97%) e AUC (+3,74%), com ganhos mais expressivos nos algoritmos KNN e SVM, que são particularmente sensíveis ao desbalanceamento de classes.

Além do balanceamento de classes, a otimização dos hiperparâmetros dos algoritmos de ML é essencial para maximizar seu desempenho. Técnicas como *Grid-Search*, que exploram sistematicamente diferentes combinações de parâmetros, permitem identificar a configuração ideal para cada algoritmo e conjunto de dados específicos [5], [10].

Dentre os diversos métodos utilizados em ML, para classificação e análise de dados, destaca-se o emprego de técnicas como as Curvas Principais (CP), que possibilitam representar dados de alta dimensão de maneira mais sintética, de uma forma compacta unidimensional. Haste e Stuetzle [11] desenvolveram o conceito de CP como uma abordagem que é uma generalização não linear do método de Análise de Componentes Principais (PCA). São curvas suaves de dimensão unidimensional que atravessam o centro de distribuições multidimensionais, permitindo uma representação mais compacta e adaptável dos dados, ou seja, de forma unidimensional. As CP tem se mostrado uma ferramenta poderosa para o reconhecimento de padrões em conjuntos de dados complexos.

No algoritmo de extração de CP, o método k-segmentos realiza o processo de extração da CP de maneira incremental, aumentando gradativamente o número de segmentos a cada iteração. Devido à sua robustez, baixa sensibilidade a mínimos locais e garantia de convergência, é amplamente aplicado na extração de curvas. Para utilizá-lo, o usuário precisa definir alguns parâmetros, como o número máximo de segmentos e o coeficiente de suavidade da curva. Por outro lado, determinados valores numéricos que influenciam a representatividade dos dados, como o comprimento dos segmentos que compõem a curva, são fixados [12].

No entanto, ajustar os hiperparâmetros das CP é uma tarefa complexa, que impacta diretamente a capacidade do modelo de capturar adequadamente a estrutura dos dados. Parâmetros como número de segmentos, coeficiente de suavização e largura de banda exigem calibração cuidadosa, pois valores inadequados podem comprometer significativamente o desempenho do modelo. Diversos trabalhos na literatura, como os de Syarif et al. [8], Bergstra e Bengio [5], Kégl et al. [13] e Meng e Eloyan [14] destacam a importância da escolha apropriada de hiperparâmetros em modelos não lineares, como CP, e que a complexidade e a flexibilidade da CP estão fortemente ligadas aos valores dos hiperparâmetros, e que uma escolha inadequada pode levar a *overfitting* ou sub-representação dos dados. Nesse contexto, este trabalho tem como objetivo otimizar os hiperparâmetros das CP por meio da técnica *Grid-Search*, aprimorando sua aplicação na classificação da recorrência do câncer de tireoide. Além disso, propõe-se uma comparação com outros algoritmos de classificação, como *Naïve Bayes* (NB) e *Support Vector Machine* (SVM), utilizando dados desbalanceados e balanceados com a técnica SMOTE.

Este artigo está estruturado da seguinte forma: a Seção II descreve a base de dados utilizada, o pré-processamento realizado e as técnicas de balanceamento aplicadas. Na Seção III são apresentados os métodos de classificação utilizados. Já a Seção IV apresenta as métricas para avaliar o desempenho dos métodos. A Seção V apresenta os experimentos computacionais realizados e discute seus resultados. Por fim, na Seção VI, as conclusões são apresentadas.

II. Materiais e Métodos

Nesta seção são descritas a base de dados utilizada, o pré-processamento realizado e a técnica de balanceamento aplicada.

A. Base de Dados

A base de dados utilizada foi obtida no repositório *Kaggle* [15], intitulada *"Thyroid Cancer Recurrence Dataset"*. Essa base contém dados de recorrência do câncer de tireoide após terapia com iodo radioativo (RAI). Ao todo, a base possui 383 registros de pacientes com 13 atributos:

- Idade (***Age***): idade do paciente (em anos);
- Sexo (***Gender***): sexo do paciente (*Male* ou *Female*);

- Radioterapia Hx (***Hx Radiotherapy***): histórico de radioterapia anterior (*no* ou *yes*);
- Adenopatia (***Adenopathy***): presença de comprometimento dos linfonodos (*no* ou *yes*);
- Patologia (***Pathology***): tipo de câncer de tireoide (por exemplo, *Micropapillary*);
- Focalidade (***Focality***): focalidade do tumor (*Uni-Focal* ou *Multi-Focal*);
- Risco (***Risk***): classificação de risco de câncer (*Low*, *Intermediate*, *High*);
- ***T***: classificação do tumor (T1, T2, etc.);
- ***N***: classificação dos linfonodos (N0, N1, etc.);
- ***M***: classificação de metástases (M0, M1, etc.).
- Estágio (***Stage***): estadiamento do câncer (estágio I, II, III, IV);
- Resposta (***Response***): resposta ao tratamento (*Excellent*, *Indeterminate*, etc.);
- Recorrente (***Recurred***): variável alvo, indicando se o câncer recorreu (*yes* ou *no*).

B. Pré-Processamento

No pré-processamento, foi realizada a transformação da variável alvo, utilizando o método *Label Encoder* [16] para a transformação de categórica para numérica. Ou seja, as categorias *no* e *yes* foram transformadas em classes 0 e 1, respectivamente. O número de amostras para as classes 0 e 1 é 275 e 108, respectivamente. Foi aplicada a técnica *Holdout* [17] para dividir a base em treinamento e teste, sendo 80% e 20%, respectivamente. Além disso, as variáveis utilizadas para prever a variável alvo também foram transformadas de categóricas para numéricas. Por fim, foi realizada a normalização *MinMax* [16], que consiste em ajustar os valores das variáveis para um intervalo compreendido entre 0 e 1. Esse processo é útil para garantir que todas as variáveis tenham a mesma escala, prevenindo que variáveis com magnitudes significativamente maiores impactem de forma desproporcional a otimização e o treinamento dos modelos de ML [18].

C. Balanceamento

Um dos obstáculos mais comuns em conjuntos de dados é o desbalanceamento, que acontece quando uma classe tem um número de amostras consideravelmente maior do que as demais. Uma estratégia comum para abordar essa questão é o *Oversampling*. O *Oversmpling* eleva o número de amostras da classe menos representada, equiparando-o ao da classe predominante. A técnica *Synthetic Minority Oversampling Technique* (SMOTE) utiliza o *Oversampling*.

O SMOTE é um método de sobreamostragem que gera novas amostras sintéticas para a classe minoritária. Este procedimento consiste em escolher uma amostra da classe minoritária e criar exemplos sintéticos ao longo das linhas que conectam a amostra selecionada aos k vizinhos mais próximos dentro da mesma classe [7].

III. Métodos de Classificação

Nesta seção, são descritos o método de CP, os métodos de classificação utilizados para fins de comparação, bem como a técnica de otimização de hiperparâmetros e a validação cruzada utilizada.

A. Curvas Principais (CP)

O método de extração de CP conhecido como k-segmentos é utilizado para aproximar e simplificar curvas por meio de uma divisão em segmentos, com o objetivo de representar a sequência de dados de maneira eficiente, preservando suas características essenciais. Esse método é particularmente útil em problemas de compressão de dados, permitindo uma redução significativa da complexidade sem perder informações relevantes [19]. Seu processo pode ser dividido em três etapas, detalhadas a seguir:

- Etapa 0 - Adição do segmento inicial: O primeiro segmento é inserido levando em conta todos os eventos do conjunto de dados. Inicialmente, é determinado um centro correspondente à média dos valores do conjunto. A partir desse ponto central, o primeiro segmento é traçado na direção do primeiro componente principal, com um comprimento equivalente a 3/2 do desvio padrão associado a esse componente.
- Etapa 1- Adição de um novo segmento: Para os segmentos subsequentes, determina-se um ponto central utilizando o agrupamento *k-means*. Com base nas regiões de Voronoi, que contêm os eventos mais próximos desse novo centro do que de qualquer um dos segmentos já estabelecidos, selecionam-se os eventos que formarão o novo agrupamento. Como a distribuição dos eventos muda, os segmentos previamente existentes são recalculados. Por fim, os segmentos são conectados por uma linha reta sem suavização.
- Etapa 2- Avaliação das condições de término do método: Verifica-se se o algoritmo atingiu a convergência ou se o número máximo de segmentos, K_max, definido pelo usuário, foi alcançado. A convergência é considerada quando o maior agrupamento possível contém menos de três segmentos. Se nenhum dos critérios de parada for satisfeito, o processo retorna ao Passo 1 para continuar a iteração.

O algoritmo k-segmentos possui três parâmetros, que são *k_max*, f e o *lamda*. O *k_max* define o número máximo de segmentos lineares que podem ser ajustados à curva principal. Já o f é o parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva. E o *lamda* controla a penalização da complexidade da curva. Valores maiores de *lamda* favorecem curvas mais suaves e menos flexíveis, enquanto valores menores permitem ajustes mais próximos aos dados.

A Fig. 1 exibe um fluxograma que representa o funcionamento do algoritmo, em que K indica a quantidade atual de segmentos e K_max corresponde ao limite máximo de segmentos estabelecido pelo usuário.

Já na Fig. 2 é apresentado um exemplo de classificação das CP, na qual "f", "i" e "h" representam as CP em um espaço de características. Ao analisar uma amostra "K", é possível calcular a distância Kf que indica a separação entre K e a

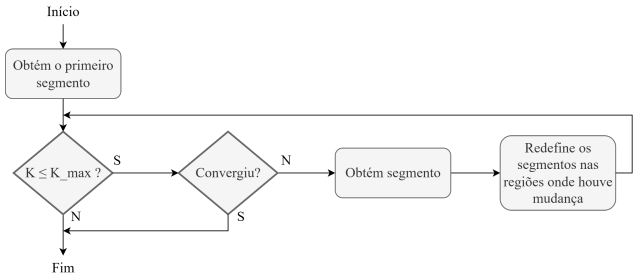

Fig. 1. *Etapas do método k-segmentos.*

curva "f". Já a distância K_i mede a distância entre K e a curva "i". E a distância Kh representa a distância entre K e a curva "h". Com o cálculo dessas distâncias, é possível avaliar e classificar "K" como pertencente à classe "h", por ter a menor distância.

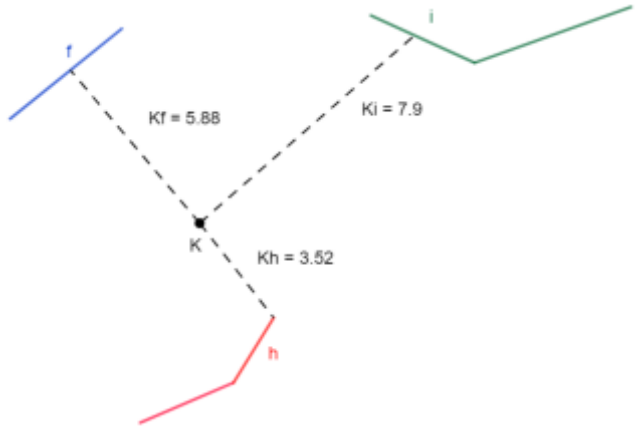

Fig. 2. Classificação com Curvas Principais. Fonte: Retirado de [20].

B. Naïve Bayes (NB)

O método *Naïve Bayes* (NB) se baseia na previsão de probabilidades. O algoritmo, para cada classe, determina a chance de um objeto específico pertencer a essa classe [21]. As probabilidades são obtidas pelo NB através da frequência de associação entre os valores da classe e os valores dos diversos atributos de entrada observados nos dados de treinamento. Contudo, é fundamental destacar uma premissa essencial para o funcionamento desse classificador: a suposição de independência condicional entre os atributos, dado que a classe é conhecida. Essa hipótese implica que, para cada classe, as variáveis de entrada são consideradas estatisticamente independentes umas das outras. O teorema de NB é utilizado para calcular a probabilidade futura $P(C_k|X)$ de que um exemplo X se enquadre na classe C_k. A equação geral é expressa por:

$$P(C_k \mid X) = \frac{P(C_k) \cdot P(X \mid C_k)}{P(X)} \tag{1}$$

Onde $P(C_k|X)$ representa a probabilidade posterior da classe C_k com base em um exemplo X, $P(C_k)$ representa a probabilidade inicial da classe C_k. $P(X|C_k)$ representa a chance de observar X, presumindo que ele pertença à classe C_k e $P(X)$ representa a chance de observar o exemplo X (igual para todas as classes).

C. Support Vector Machine (SVM)

O *Support Vector Machine* (SVM) é um método de aprendizagem supervisionada, com o objetivo de criar um limite de decisão entre duas ou mais classes, permitindo a predição de rótulos de um ou mais recursos de vetores [22]. O limite de decisão é chamado de hiperplano, é orientado de forma que fique o mais longe possível dos pontos de dados com maior proximidade de cada uma das classes. São chamados de vetores de suporte estes pontos que estão mais próximos [23].

Seja um conjunto de dados de treinamento rotulado expresso pela Eq. (2):

$$(x_1, y_1), \ldots, (x_n, y_n), x_i \in \mathbb{R}^d \text{e} y_i \in (-1, +1) \tag{2}$$

onde x_i é a representação do vetor de recursos, y_i a classe, podendo ser negativa ou positiva de um item de treinamento i. A otimização do hiperplano é definida pela Eq. (3):

$$w^T x + b = 0 \tag{3}$$

onde w é o vetor de pesos, x é o vetor de recursos de entrada, e b é o viés. w e b devem satisfazer as seguintes desigualdades para todo conjunto de treinamento, conforme a Eq. (4).

$$\begin{aligned} w^T x + b \geq +1 \quad \text{se} \quad y_i = 1 \\ w^T x + b \leq -1 \quad \text{se} \quad y_i = -1 \end{aligned} \tag{4}$$

O modelo SVM é treinado para encontrar um w e b que permita realizar a separação dos dados no hiperplano e maximize a margem. Muitas vezes, os dados reais não são linearmente separáveis no espaço original das características. Para contornar essa limitação, o SVM utiliza a técnica de funções *kernel*, que consiste em mapear os dados para um espaço de dimensão superior, onde a separação linear torna-se possível. O modelo SVM, por se tratar de um problema de classificação, é denominado como *Support Vector Classification* (SVC).

D. Grid-Search

O desempenho ideal de algoritmos de aprendizado de máquina está diretamente ligado ao ajuste preciso de seus parâmetros. Para desenvolver um modelo de classificação com alta taxa de acerto, é essencial escolher o algoritmo mais apropriado e ajustar seus parâmetros de forma eficiente. Entretanto, realizar esse processo manualmente pode ser bastante trabalhoso, especialmente em casos em que o algoritmo possui uma quantidade significativa de parâmetros [8].

O método *Grid-Search* consiste em uma busca exaustiva dentro de um subconjunto previamente definido do espaço de hiperparâmetros. Para isso, os valores desses hiperparâmetros são determinados com base em um limite inferior, um limite superior e a quantidade de intervalos entre esses valores [5]. O

desempenho de cada combinação de parâmetros é avaliado por meio de métricas como acurácia, precisão, *recall*, entre outras. Quando integrado à validação cruzada, o *Grid-Search* avalia detalhadamente cada conjunto de hiperparâmetros ajustados ao classificador, com o objetivo de maximizar o desempenho do modelo [8]. A Fig. 3 apresenta um fluxograma com a visão geral do processo de seleção de parâmetros e avaliação de um modelo com *GridSearch*.

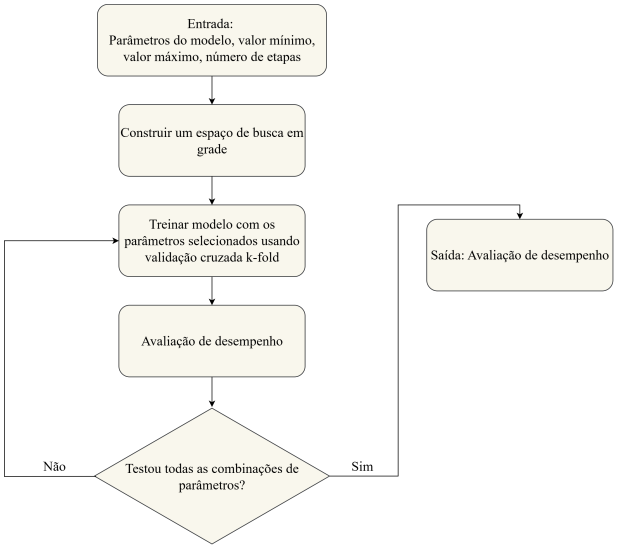

Fig. 3. *Grid-Search*.

Para a avaliação dos classificadores, foi utilizada a estratégia de validação cruzada *K-Fold* (KF) [24]. Nesta abordagem, o conjunto de dados disponível, contendo N amostras, é particionado em K subconjuntos, onde $K > 1$. Após a divisão, $K - 1$ subconjuntos são utilizados para o treinamento do modelo, enquanto o subconjunto restante é reservado para teste. Ao final do processo, calcula-se o erro de validação com base nos resultados obtidos. Esse procedimento é repetido K vezes, alternando o subconjunto de teste a cada iteração, garantindo que todas as amostras sejam utilizadas tanto para treinamento quanto para teste.

IV. Métricas de Avaliação

Nesta seção são apresentadas as métricas empregadas para avaliar o desempenho dos métodos. Também é apresentado o teste estatístico utilizado, que foi o teste de *Tukey*, com o intuito de verificar se houve diferença significativa entre os métodos.

Foram empregadas as seguintes métricas para avaliar o desempenho dos métodos: Acurácia (Ac), Precisão (Pr), *Recall* (Re) e Coeficiente de *Kappa*. Nas equações dessas métricas, *True Positives* (TP) se refere ao número de amostras corretamente identificadas como pertencentes a uma classe específica, enquanto *False Positives* (FP) representa a quantidade de amostras erroneamente classificadas como não pertencentes à classe em questão. *True Negative* (TN) representa a quantidade de amostras classificadas de maneira errônea como não pertencentes à classe em questão.

A. Acurácia

A acurácia (Eq. (5)) estabelece a porcentagem de previsões acertadas em relação ao total de instâncias analisadas.

$$\text{Acurácia} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

B. Precisão

A precisão (Eq. (6)) é um indicador que mede o número de *True Positives* nas amostras que foram categorizadas como pertencentes a uma categoria específica.

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

C. Recall

O *Recall* (Eq. (7)), também chamado de Sensibilidade ou *Rate of True Positive Rate* (TPR), diz respeito ao número de *True Positives*, isto é, a quantidade de amostras corretamente reconhecidas como pertencentes a uma classe, entre todas as amostras que de fato pertencem a essa classe.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

D. Coeficiente de Kappa

O Coeficiente *Kappa* (Eq. (8)) avalia o nível de concordância entre dois avaliadores, modificando a concordância observada para considerar a probabilidade de coincidência ao acaso. Um valor de $k = 1$ indica uma concordância perfeita, enquanto $k = 0$ equivale a uma concordância equivalente ao acaso, e valores de $k < 1$ sinalizam discordância.

$$kappa = \frac{P_o - P_e}{1 - P_e} \tag{8}$$

onde P_o simboliza a porcentagem de concordâncias observadas, enquanto P_e representa a porcentagem de concordâncias que seriam esperadas ao acaso. A análise da Estatística Kappa é utilizada para avaliar o nível de acordo entre os avaliadores.

E. Estatística de Tukey

O teste de comparações múltiplas de *Tukey*, também conhecido como teste de diferença honestamente significativa (HSD) de *Tukey* [25], é particularmente útil quando se deseja realizar comparações pareadas entre múltiplos grupos. Este teste controla a taxa de erro tipo I para o conjunto completo de comparações, sendo mais rigoroso que o teste *t* de *Student* aplicado repetidamente. A estatística do teste baseia-se na distribuição da amplitude studentizada q, calculada conforme a Eq. 9.

$$q = \frac{M_i - M_j}{\sqrt{\frac{MSE}{n}}} \tag{9}$$

onde M_i e M_j são as médias dos grupos i e j, respectivamente, MSE é o quadrado médio do erro obtido na ANOVA, e n é o número de observações em cada grupo [26].

Na análise comparativa dos classificadores, o teste de *Tukey* permite identificar quais diferenças entre os algoritmos são estatisticamente significativas, considerando um nível de significância de 5% ($\alpha = 0,05$). Uma entrada marcada como "*True*" indica que há evidência estatística suficiente para concluir que o desempenho dos dois métodos comparados difere significativamente, enquanto "*False*" sugere que as diferenças observadas podem ser atribuídas à variabilidade amostral.

Para avaliação das diferenças entre os classificadores, é calculado o valor crítico de *Tukey* (q_{crit}) baseado no número de grupos comparados, nos graus de liberdade do erro e no nível de significância. Com esse valor, determina-se a diferença mínima significativa (DMS), permitindo afirmar com maior confiança quais configurações de algoritmos superam realmente as demais em termos de desempenho preditivo.

V. Resultados e Discussões

Os experimentos computacionais foram realizados utilizando a linguagem de programação *Python* e implementações baseadas nas bibliotecas de estrutura *Scikit-Learn* [16], *ocpc_py* [27] e *Pandas* [28]. Todos os experimentos foram executados em um computador com as seguintes especificações: *Intel(R) Core (TM)* i5-1135G7, 8 GB de RAM e sistema operacional *Windows* 10. Foram realizadas 30 iterações independentes para avaliar a metodologia.

A Tabela I apresenta a descrição dos métodos utilizados no *Grid-Search* com a validação cruzada com K igual a 5, indicando o método, os parâmetros para maximizar o desempenho do modelo e qual a configuração utilizada neles, indicando os valores usados durante as execuções.

TABELA I
DESCRIÇÃO DOS MÉTODOS.

Método	Parâmetros	Variação Parâmetros
CP	k_max	[2, 3, 4, 5, 6]
	f	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5]
	lamda	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
NB	*var_smoothing*	$[1 \times 10^{-9}, 1 \times 10^{-8}, 1 \times 10^{-7}, 1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4} 1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$
SVC	C	$[1 \times 10^{-2}, 1 \times 10^{-1}, 1 \times 10^{0}]$ $[1 \times 10^{1}, 1 \times 10^{2}]$
	gamma	$[1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$
	kernel	['*linear*', '*rbf*', '*poly*', '*sigmoid*']

Na Tabela II pode-se visualizar os resultados obtidos, com a média e o desvio padrão das iterações para cada métrica, com os dados desbalanceados. Os melhores resultados foram obtidos com o algoritmo CP e SVC. A CP teve uma acurácia de 0,937, precisão de 0,939, *recall* de 0,937 e um *kappa* de 0,829, mostrando um nível de concordância perfeita do resultado. Isso indica que o CP é capaz de captar padrões complexos dos dados sem perder generalização, mesmo em situações desafiadoras com classes desbalanceadas. Já o SVC teve uma acurácia de 0,945, precisão de 0,946, *recall* de 0,945 e um *kappa* de 0,854, também apresentando um nível de concordância perfeita do resultado.

TABELA II
RESULTADO DAS MÉTRICAS DE DESEMPENHO DOS MODELOS PARA OS DADOS DESBALANCEADOS.

Método	**Ac**	**Pr**	**Re**	**Kappa**
CP	**0,937 ± 0,025**	**0,939 ± 0,025**	**0,937 ± 0,025**	**0.829 ± 0.076**
NB	0,899 ± 0,034	0.904 ± 0,031	0,899 ± 0,034	0,730 ± 0,105
SVC	**0,945 ± 0,033**	**0,946 ± 0,032**	**0,945 ± 0,033**	**0,854 ± 0,092**

A Tabela III apresenta a configuração dos parâmetros do melhor modelo de CP no conjunto de teste. O melhor modelo de CP é composto por um *k_max* igual a 2, indica que a curva será representada por até dois segmentos. Um *f* de parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva igual a 0,6 e um *lamda* responsável por controlar a penalização da complexidade da curva igual a 0,1. Tal configuração resultou, no conjunto de teste, acurácia de 0,844, precisão de 0,842, *recall* de 0,844 e *kappa* de 0,638.

TABELA III
DESEMPENHO DO MELHOR MODELO NO CONJUNTO DE TESTE COM DADOS DESBALANCEADOS PARA O ALGORITMO DE CP.

Parâmetros	Ac	Pr	Re	*Kappa*
k_max: 2, *f*: 0,6, *lamda*: 0,1	0,844	0,842	0,844	0,638

A Tabela IV apresenta a configuração dos parâmetros do pior modelo de CP testado no conjunto de teste. O pior modelo de CP é composto por um *k_max* igual a 3, *f* igual a 1,3 e *lamda* igual a 0,1. Tal configuração resultou, no conjunto de teste, acurácia de 0,805, precisão de 0,803, *recall* de 0,805 e *kappa* de 0,534. Destaca-se a importância de otimizar os hiperparâmetros de CP para melhorar o desempenho do modelo.

TABELA IV
DESEMPENHO DO PIOR MODELO NO CONJUNTO DE TESTE COM DADOS DESBALANCEADOS PARA O ALGORITMO DE CP.

Parâmetros	Ac	Pr	Re	*Kappa*
k_max: 3, *f*: 1,3, *lamda*: 0,1	0,805	0,803	0,805	0,534

A Tabela V apresenta os resultados do teste de *Tukey* (um teste de comparação pareada) para os métodos de classificação utilizados com os dados desbalanceados. As entradas marcadas como *"True"* indicam que houve diferença significativa entre os modelos (valor de $p < 0.05$), enquanto *"False"* indica que não houve diferença significativa (valor de $p > 0.05$ com 95% de confiança). Pode-se observar através da tabela que não houve diferença significativa entre o método de CP e SVC.

A Tabela VI apresenta os resultados obtidos para os dados balanceados com o SMOTE, com a média e o desvio padrão das iterações para cada métrica. Os melhores resultados foram obtidos com o algoritmo CP e SVC. A CP teve uma acurácia de 0,917, precisão de 0,921, *recall* de 0,917 e um *kappa* de 0,832, mostrando um nível de concordância perfeita do resultado. Já o SVC teve uma acurácia de 0,924, precisão de 0,926, *recall* de 0,924 e um *kappa* de 0,847, mostrando também um nível de concordância perfeita do resultado.

TABELA V
RESULTADOS DO TESTE DE *Tukey* INDICANDO DIFERENÇAS PAREADAS ESTATISTICAMENTE SIGNIFICATIVAS ENTRE OS MÉTODOS COM DESBALANCEAMENTO (P $< 0{,}05$)

Método 1	**Método 2**	**Ac**	**Pr**	**Re**	**Kappa**
CP	NB	*True*	*True*	*True*	*True*
CP	**SVC**	***False***	***False***	***False***	***False***
NB	SVC	*True*	*True*	*True*	*True*

TABELA VI
RESULTADO DAS MÉTRICAS DE DESEMPENHO DOS MODELOS PARA OS DADOS BALANCEADOS COM O MÉTODO SMOTE.

Método	**Ac**	**Pr**	**Re**	**Kappa**
CP	**0,917 ± 0,026**	**0,921 ± 0,024**	**0,917 ± 0,026**	**0,832 ± 0,052**
NB	0,879 ± 0,031	0,887 ± 0,027	0,879 ± 0,031	0,758 ± 0,061
SVC	**0,924 ± 0,028**	**0,926 ± 0,027**	**0,924 ± 0,028**	**0.847 ± 0.059**

Já a Tabela VII apresenta a configuração dos parâmetros do melhor modelo de CP no conjunto de teste. O melhor modelo de CP é composto por um *k_max* igual a 4, indica que a curva será representada por até quatro segmentos. Um *f* de parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva igual a 1,6 e um *lamda* responsável por controlar a penalização da complexidade da curva igual a 0,1. Tal configuração resultou, no conjunto de teste, acurácia de 0,882, precisão de 0,883, *recall* de 0,882 e *kappa* de 0,747.

TABELA VII
DESEMPENHO DO MELHOR MODELO NO CONJUNTO DE TESTE COM DADOS BALANCEADOS COM O MÉTODO SMOTE.

Parâmetros	Ac	Pr	Re	*Kappa*
k_max: 4, *f*: 1,6, *lamda*: 0,1	0,882	0,883	0,882	0,747

A Tabela VIII apresenta a configuração dos parâmetros do pior modelo de CP testado no conjunto de teste. O pior modelo de CP é composto por um *k_max* igual a 5, *f* igual a 0,3 e *lamda* igual a 0,1. Tal configuração resultou, no conjunto de teste, acurácia de 0,863, precisão de 0,863, *recall* de 0,863 e *kappa* de 0,730. Percebe-se, portanto, a relevância do processo de otimização dos hiperparâmetros do algoritmo de CP, uma vez que escolhas inadequadas desses parâmetros podem comprometer significativamente o desempenho preditivo do modelo.

TABELA VIII
DESEMPENHO DO PIOR MODELO NO CONJUNTO DE TESTE COM DADOS BALANCEADOS COM O MÉTODO SMOTE.

Parâmetros	Ac	Pr	Re	*Kappa*
k_max: 5, *f*: 0,3, *lamda*: 0,1	0,863	0,863	0,863	0,730

Na Tabela IX são apresentados os resultados do teste de *Tukey* (um teste de comparação pareada) para os métodos de classificação utilizados com os dados balanceados com o *SMOTE* como abordagem. As entradas marcadas como *"True"* indicam que houve diferença significativa entre os modelos (valor de $p < 0.05$), enquanto *"False"* indica que não houve diferença significativa (valor de $p > 0.05$ com 95% de confiança). Pode-se observar através da tabela que não houve diferença significativa também entre o método de CP e SVC para a abordagem de balanceamento *SMOTE*.

TABELA IX
RESULTADOS DO TESTE DE *Tukey* INDICANDO DIFERENÇAS PAREADAS ESTATISTICAMENTE SIGNIFICATIVAS ENTRE OS MÉTODOS COM BALANCEAMENTO *SMOTE* COMO ABORDAGEM (P $< 0{,}05$)

Método 1	**Método 2**	**Ac**	**Pr**	**Re**	**Kappa**
CP	NB	*True*	*True*	*True*	*True*
CP	**SVC**	***False***	***False***	***False***	***False***
NB	SVC	*True*	*True*	*True*	*True*

Os experimentos realizados mostraram que os algoritmos de CP e SVM alcançaram os melhores desempenhos tanto em dados desbalanceados quanto balanceados com SMOTE, indicando o uso dessas abordagens para a tarefa de classificação da recorrência do câncer de tireoide. Os resultados indicam que a otimização dos hiperparâmetros por *Grid-Search* foi essencial para melhorar o desempenho de CP, evidenciando que a escolha adequada dos parâmetros *k_max*, *f* e *lamda* impacta diretamente a capacidade do modelo no conjunto de dados.

VI. CONCLUSÕES

Neste trabalho, foi avaliada a aplicação do método de CP com otimização de hiperparâmetros por *Grid-Search* para a classificação da recorrência do câncer de tireoide após terapia com iodo radioativo. Além disso, foi realizada uma comparação dele com os métodos de classificação NB e SVM. Os resultados demonstraram que o algoritmo CP é uma alternativa eficaz, apresentando desempenho comparável ao modelo SVM, tanto com dados desbalanceados quanto balanceados com a abordagem SMOTE. Ademais, o uso do *Grid-Search* para ajuste dos hiperparâmetros do CP aprimorou significativamente sua capacidade preditiva, demonstrando que a escolha adequada dos hiperparâmetros é fundamental para a aplicação desse método em bases reais de dados médicos.

Apesar dos bons resultados, observou-se uma leve queda na acurácia dos métodos no conjunto de teste em comparação à validação cruzada, sugerindo que futuras pesquisas devem focar na redução do *overfitting* e na generalização dos modelos, por meio de técnicas de regularização adicionais ou validação em bases externas. Como trabalhos futuros, pretende-se aplicar algumas técnicas de seleção de características como Análise de Componentes Principais (PCA) e *Select K-Best*, e algumas meta-heurísticas, como Colônia de Abelhas e Otimização de Lobos Cinzentos, para encontrar os parâmetros ótimos, realizando uma busca contínua.

AGRADECIMENTOS

Os autores agradecem o apoio financeiro da Universidade Federal de Lavras (UFLA), da Coordenação de

Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

REFERÊNCIAS

[1] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, "Cancer statistics, 2021," *CA: a cancer journal for clinicians*, vol. 71, no. 1, pp. 7–33, 2021. [Online]. Available: https://acsjournals.onlinelibrary.wiley.com/doi/full/10.3322/caac.21654

[2] B. R. Haugen, E. K. Alexander, K. C. Bible, G. M. Doherty, S. J. Mandel, Y. E. Nikiforov, F. Pacini, G. W. Randolph, A. M. Sawka, M. Schlumberger *et al.*, "2015 american thyroid association management guidelines for adult patients with thyroid nodules and differentiated thyroid cancer: the american thyroid association guidelines task force on thyroid nodules and differentiated thyroid cancer," *Thyroid*, vol. 26, no. 1, pp. 1–133, 2016. [Online]. Available: https://www.liebertpub.com/doi/full/10.1089/thy.2015.0020

[3] R. M. Tuttle, R. I. Haddad, D. W. Ball, D. Byrd, P. Dickson, Q.-Y. Duh, H. Ehya, M. Haymart, C. Hoh, J. P. Hunt *et al.*, "Thyroid carcinoma, version 2.2019, nccn clinical practice guidelines in oncology," *Journal of the National Comprehensive Cancer Network*, vol. 17, no. 12, pp. 1428–1456, 2019. [Online]. Available: https://jnccn.org/view/journals/jnccn/17/12/article-p1428.xml

[4] B. Claus, D. Ross, and A. L. A. Okunola, "The role of machine learning in data-driven decision-making," 2024.

[5] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The journal of machine learning research*, vol. 13, no. 1, pp. 281–305, 2012.

[6] H. Wang, C. Zhang, Q. Li, T. Tian, R. Huang, J. Qiu, and R. Tian, "Development and validation of prediction models for papillary thyroid cancer structural recurrence using machine learning approaches," *BMC Cancer*, vol. 24, p. 427, 2024. [Online]. Available: https://bmccancer.biomedcentral.com/articles/10.1186/s12885-024-12146-4

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[8] I. Syarif, A. Prugel-Bennett, and G. Wills, "Svm parameter optimization using grid search and genetic algorithm to improve classification performance," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 14, no. 4, pp. 1502–1509, 2016.

[9] E. Clark, S. Price, T. Lucena, B. Haberlein, A. Wahbeh, and R. Seetan, "Predictive analytics for thyroid cancer recurrence: A machine learning approach," *Knowledge*, vol. 4, pp. 557–570, 2024. [Online]. Available: https://doi.org/10.3390/knowledge4040029

[10] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[11] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American statistical association*, vol. 84, no. 406, pp. 502–516, 1989.

[12] E. C. C. Moraes and D. D. Ferreira, "A principal curve-based method for data clustering," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2016, pp. 3966–3971.

[13] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "Learning and design of principal curves," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 3, pp. 281–297, 2000.

[14] K. Meng and A. Eloyan, "Principal manifold estimation via model complexity selection," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 83, no. 2, pp. 369–394, 2021.

[15] A. Vinay, "Thyroid cancer recurrence dataset," Kaggle. Disponível em https://www.kaggle.com/datasets/aneevinay/thyroid-cancer-recurrence-dataset/data, 2025.

[16] P. Fabian, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, p. 2825, 2011.

[17] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, 1995, pp. 1137–1143.

[18] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, "The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis," in *International conference on soft computing models in industrial and environmental applications*. Springer, 2023, pp. 344–353.

[19] J. J. Verbeek, N. Vlassis, and B. Kröse, "A k-segments algorithm for finding principal curves," *Pattern Recognition Letters*, vol. 23, no. 8, pp. 1009–1017, 2002.

[20] D. H. H. de Castro, C. A. S. Silva, and D. D. Ferreira, "Curvas principais para a triagem de pacientes com tuberculose via análise de imagens de raio x," 2024.

[21] A. Jamain and D. J. Hand, "The naive bayes mystery: A classification detective story," *Pattern Recognition Letters*, vol. 26, no. 11, pp. 1752–1760, 2005.

[22] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*. Academic Press, 2020, pp. 101–121.

[23] S. Suthaharan, "Support vector machine," in *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*. Boston, MA: Springer US, 2016, pp. 207–235.

[24] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2, August 1995, pp. 1137–1145.

[25] D. C. Montgomery, *Design and Analysis of Experiments*, 8th ed. John Wiley & Sons, 2013.

[26] R. E. Kirk, *Experimental Design: Procedures for the Behavioral Sciences*, 4th ed. SAGE Publications, 2013.

[27] F. E. d. M. Borges, "Ocpc-py: One class classifier based on principal curves in python," Disponível em https://pypi.org/project/ocpc-py/, 2023.

[28] J. Reback, W. McKinney, J. Van Den Bossche, T. Augspurger, P. Cloud, A. Klein, S. Seabold *et al.*, "pandas-dev/pandas: Pandas 1.0. 5," 2020.