BRUNO DA SILVA MACÊDO

EXPLORANDO CURVAS PRINCIPAIS COM A META-HEURÍSTICA LOBO CINZENTO PARA A CLASSIFICAÇÃO DE DADOS SINTÉTICOS E DE DESEMPENHO ACADÊMICO DOS ESTUDANTES NO ENEM

LAVRAS – MG

2026

BRUNO DA SILVA MACÊDO

EXPLORANDO CURVAS PRINCIPAIS COM A META-HEURÍSTICA LOBO CINZENTO PARA A CLASSIFICAÇÃO DE DADOS SINTÉTICOS E DE DESEMPENHO ACADÊMICO DOS ESTUDANTES NO ENEM

Dissertação apresentada à Universidade Federal de Lavras (UFLA), como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. Dr. Bruno Henrique Groenner Barbosa
Orientador

Prof. Dr. Danton Diego Ferreira
Coorientador

LAVRAS – MG
2026

Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Macêdo, Bruno da Silva

Explorando Curvas Principais com a Meta-Heurística Lobo Cinzento para a Classificação de Dados Sintéticos e de Desempenho Acadêmico dos Estudantes no ENEM / Bruno da Silva Macêdo. – Lavras : UFLA, 2026.

136p. : il.

Dissertação (Mestrado)–Universidade Federal de Lavras, 2026.

Orientador: Prof. Dr. Bruno Henrique Groenner Barbosa.

Coorientador: Prof. Dr. Danton Diego Ferreira.

Bibliografia.

1. Dados Educacionais. 2. Aprendizado de Máquina. 3. Curvas Principais. I. Barbosa, Bruno Henrique Groenner. II. Universidade Federal de Lavras. III. Título.

BRUNO DA SILVA MACÊDO

EXPLORANDO CURVAS PRINCIPAIS COM A META-HEURÍSTICA LOBO CINZENTO PARA A CLASSIFICAÇÃO DE DADOS SINTÉTICOS E DE DESEMPENHO ACADÊMICO DOS ESTUDANTES NO ENEM

EXPLORING PRINCIPAL CURVES WITH THE GREY WOLF META-HEURISTIC FOR THE CLASSIFICATION SYNTHETIC DATA AND ACADEMIC PERFORMANCE OF STUDENTS IN THE ENEM

Dissertação apresentada à Universidade Federal de Lavras (UFLA), como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 13 de março de 2026.

Prof. Dr. Bruno Henrique Groenner Barbosa	UFLA
Prof. Dr. Danton Diego Ferreira	UFLA
Prof. Dr. Wilian Soares Lacerda	UFLA
Prof. Dr. Giovani Bernardes Vitor	UNIFEI

gov.br
Documento assinado digitalmente
BRUNO HENRIQUE GROENNER BARBOSA
Data: 26/03/2026 09:58:08-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Bruno Henrique Groenner Barbosa
Orientador

Prof. Dr. Danton Diego Ferreira
Coorientador

LAVRAS – MG
2026

Dedico este trabalho aos meus pais e aos meus queridos irmãos.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me permitir realizar o sonho de cursar o mestrado e por me fortalecer diariamente para seguir firme em meus objetivos. Aos meus pais, registro minha gratidão pelo apoio constante, por acreditarem em mim e por não me deixarem desistir, mesmo diante das dificuldades que surgem ao longo da jornada acadêmica. Agradeço também aos meus amigos, amigas e professores, pelo incentivo e pela ajuda em diferentes momentos. Ao professor Bruno, meu orientador, e Danton, meu coorientador, agradeço pela disponibilidade em vários momentos, pelas orientações e por contribuir significativamente para o meu aprendizado. Aos meus colegas de mestrado, que de alguma forma colaboraram para o meu aprendizado. Os docentes e ao Programa de Pós-Graduação em Engenharia de Sistemas e Automação (PPGESISA), que de alguma forma contribuíram para a minha formação e crescimento. Agradeço o grupo de pesquisa *Artificial Intelligence and Automation* (AIA) pelo espaço fornecido e suporte para realização desta pesquisa. Agradeço a Universidade Federal de Lavras (UFLA) pelo suporte para realização desta pesquisa. A presente pesquisa foi realizada com apoio da Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

“A persistência é o menor caminho do êxito.”

(Charles Chaplin)

RESUMO

A educação é fundamental para o desenvolvimento de um país e, no Brasil, destaca-se a necessidade de melhorias que podem ser impulsionadas pelo uso da Tecnologia da Informação. Por meio do Exame Nacional do Ensino Médio (ENEM), maior exame educacional do país e uma das principais portas de entrada para o ensino superior, é possível avaliar aspectos da qualidade da educação por meio da construção de indicadores educacionais baseados no desempenho dos estudantes no exame. Para prever o desempenho dos estudantes no exame com base nas variáveis presentes na base de dados do ENEM, tais como informações socioeconômicas, características escolares e dados de participação, técnicas de Aprendizagem de Máquina (ML) têm sido cada vez mais utilizadas nesse contexto, permitindo identificar padrões de desempenho, possíveis irregularidades e personalizar estratégias pedagógicas. O problema de prever o desempenho dos estudantes no ENEM tem sido investigado por diversos autores, porém, muitos não têm explorado outras técnicas de ML, como as Curvas Principais (CP). Nos últimos anos, o método de CP tem sido aplicado em diversas áreas, demonstrando potencial em problemas de classificação. Nesse contexto, esta pesquisa tem como objetivo aplicar o método de extração de CP K-segmentos para a classificação do desempenho acadêmico dos estudantes que realizaram o ENEM 2023, considerando a classe 0 para alunos que não apresentaram um desempenho esperado e a classe 1 para aqueles com bom desempenho no exame, e avaliá-lo em bases sintéticas. Além disso, aplicar o método de otimização de hiperparâmetros Lobo Cinzento (*Grey Wolf Optimizer* (GWO)) para determinar automaticamente os valores dos hiperparâmetros do método de CP K-segmentos, no qual determinar de forma manual é uma tarefa complexa. A metodologia compreende as etapas de preparação, redução de dimensionalidade, balanceamento das classes e transformação das variáveis de entrada da base do ENEM 2023, incluindo variáveis socioeconômicas, características escolares e informações dos participantes, além da aplicação da técnica GWO para a otimização dos hiperparâmetros dos modelos. Os métodos de classificação foram avaliados por meio de métricas como acurácia, *F1-Score*, precisão, *recall*, coeficiente de *Kappa*. Nos experimentos realizados em bases de dados sintéticas, o método apresentou bom desempenho nas bases compacta, alongada, esférica e espiral, com métricas superiores a 0,9700. Já nos experimentos realizados com a base do ENEM 2023, a abordagem de CP apresentou resultados competitivos em relação aos métodos da literatura comparados (*Extreme Learning Machine*, *Naive Bayes* e *Random Forest*). Entre as abordagens avaliadas, o pior resultado foi observado com t-SNE, com acurácia e *recall* de 0,7310, precisão de 0,7314,

F1-Score de 0,7309 e coeficiente de *Kappa* de 0,4621 no conjunto de teste. Enquanto com *Select K-Best* o método obteve os melhores resultados, com acurácia e *recall* de 0,7603, precisão de 0,7612, *F1-Score* de 0,7601 e coeficiente de *Kappa* de 0,5206 no conjunto de teste, superando o método *Naive Bayes* nessa configuração. Esses resultados indicam que a abordagem proposta é promissora para a classificação do desempenho acadêmico, especialmente quando combinada a estratégias adequadas de técnicas de redução de dimensionalidade e otimização.

Palavras-chave: Dados Educacionais; Desempenho Acadêmico; ENEM; Aprendizado de Máquina; Reconhecimento de Padrões; Curvas Principais; K-segmentos.

ABSTRACT

Education is fundamental to a country's development, and in Brazil, the need for improvements that can be driven by the use of Information Technology stands out. Through the National High School Exam (ENEM), the country's largest educational exam and one of the main gateways to higher education, it is possible to assess aspects of the quality of education by constructing educational indicators based on student performance on the exam. To predict student performance on the exam based on variables present in the ENEM database, such as socioeconomic information, school characteristics, and participation data, Machine Learning (ML) techniques have been increasingly used in this context, allowing the identification of performance patterns, possible irregularities, and the customization of pedagogical strategies. The problem of predicting student performance on the ENEM has been investigated by several authors, but many have not explored other ML techniques, such as Principal Curves (PC). In recent years, the PC method has been applied in various areas, demonstrating potential in classification problems. In this context, this research aims to apply the K-segment PC extraction method to classify the academic performance of students who took the 2023 ENEM exam, considering class 0 for students who did not present the expected performance and class 1 for those with good performance on the exam, and to evaluate it on synthetic bases. Furthermore, the Grey Wolf Optimizer (GWO) hyperparameter optimization method was applied to automatically determine the hyperparameter values for the K-segment PC method, a task that can be determined manually but is complex. The methodology comprises the steps of preparation, dimensionality reduction, class balancing, and transformation of the input variables from the 2023 ENEM database, including socioeconomic variables, school characteristics, and participant information, in addition to applying the GWO technique to optimize the model hyperparameters. The classification methods were evaluated using metrics such as accuracy, F1-Score, precision, recall, and Kappa coefficient. In experiments conducted on synthetic datasets, the method showed good performance in compact, elongated, spherical, and spiral datasets, with metrics greater than 0.9700. In experiments conducted using the ENEM 2023 database, the PC approach showed competitive results compared to the literature-referenced methods (Extreme Learning Machine, Naive Bayes, and Random Forest). Among the evaluated approaches, the worst result was observed with t-SNE, with an accuracy and recall of 0.7310, precision of 0.7314, F1-Score of 0.7309, and a Kappa coefficient of 0.4621 in the test set. While the Select K-Best method obtained the best results, with an accuracy and recall of 0.7603, precision of 0.7612, F1-Score of 0.7601, and a Kappa co-

efficient of 0.5206 in the test set, it outperformed the Naive Bayes method in this configuration. These results indicate that the proposed approach is promising for classifying academic performance, especially when combined with appropriate dimensionality reduction and optimization techniques.

Keywords: Educational Data; Academic Performance; ENEM; Machine Learning; Pattern Recognition; Principal Curves; K-segments.

INDICADORES DE IMPACTO

Esta pesquisa oferece uma contribuição significativa para o progresso metodológico na utilização da Inteligência Artificial no campo educacional, ao propor e avaliar uma abordagem baseada no método de Curvas Principais, K-segmentos. Essa abordagem realiza a otimização de hiperparâmetros por meio da meta-heurística do Lobo Cinzento (*Grey Wolf Optimizer* (GWO)) para classificar o desempenho acadêmico dos estudantes no Exame Nacional do Ensino Médio Brasileiro (ENEM). A pesquisa apresenta impacto tecnológico ao investigar, de maneira sistemática, o desempenho do método em bases de dados com características variadas e, em seguida, aplicá-lo em um cenário real de grande escala, utilizando a base de dados do ENEM 2023. Além disso, a pesquisa inclui técnicas de pré-processamento, balanceamento e redução de dimensionalidade, além da análise do impacto das variáveis na classificação, proporcionando suporte para interpretações mais claras e reproduzíveis dos resultados alcançados. Os resultados têm o potencial de serem empregados para auxiliar análises educacionais e diagnósticos de desempenho em grande escala. Isso pode contribuir para entender os fatores que afetam o rendimento dos estudantes e para criar ferramentas que apoiem a tomada de decisões em contextos educacionais. Os impactos concentram-se nas áreas social e tecnológica, pois a pesquisa enfatiza a importância de fatores ligados ao ambiente escolar, familiar e socioeconômico, o que pode ajudar a direcionar políticas e iniciativas para melhorar o aprendizado e diminuir as disparidades na educação. A pesquisa se relaciona com as áreas temáticas 4 - Educação e 7 - Tecnologia e produção, e está conectada aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU, especificamente o ODS 4 (Educação de qualidade), ODS 9 (Indústria, Inovação e Infraestrutura) e ODS 10 (Redução das desigualdades), ao sugerir uma solução computacional que seja aplicável e interpretável para a análise de dados educacionais em grande escala. Apesar do estudo não ter um caráter extensionista direto, instituições de ensino, gestores públicos e pesquisadores podem aproveitar seu potencial de aplicação prática. Isso também contribui para a formação acadêmica e científica no campo da Inteligência Artificial aplicada à educação.

IMPACT INDICATORS

This research offers a significant contribution to methodological progress in the use of Artificial Intelligence in the educational field, by proposing and evaluating an approach based on the Principal Curves, K-segment method. This approach optimizes hyperparameters using the Grey Wolf Optimizer (GWO) metaheuristic to classify students' academic performance in the Brazilian National High School Exam (ENEM). The research demonstrates technological impact by systematically investigating the method's performance on databases with varied characteristics and then applying it to a large-scale real-world scenario using the 2023 ENEM database. Furthermore, the research includes preprocessing, balancing, and feature selection techniques, as well as analysis of the impact of variables on classification, providing support for clearer and more reproducible interpretations of the results achieved. The results have the potential to be used to assist educational analyses and large-scale performance diagnostics. This can contribute to understanding the factors that affect student achievement and to creating tools that support decision-making in educational contexts. The impacts are concentrated in the social and technological areas, as the research emphasizes the importance of factors linked to the school, family, and socioeconomic environment, which can help guide policies and initiatives to improve learning and reduce disparities in education. The research relates to thematic areas 4 - Education and 7 - Technology and Production, and is connected to the UN Sustainable Development Goals (SDGs), specifically SDG 4 (Quality Education), SDG 9 (Industry, Innovation and Infrastructure), and SDG 10 (Reduced Inequalities), by suggesting a computational solution that is applicable and interpretable for the analysis of large-scale educational data. Although the study does not have a direct extension character, educational institutions, public managers, and researchers can take advantage of its potential for practical application. This also contributes to academic and scientific training in the field of Artificial Intelligence applied to education.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ACC	Acurácia
AD	Árvores de Decisão
AM	Aprendizado de Máquina
ANOVA	Análise de Variância
AP	Aprendizado Profundo
AUC_ROC	Area under curve ROC
CART	Classification and Regression Tree
CH	Ciências Humanas e suas Tecnologias
CN	Ciências Naturais e suas Tecnologias
CP	Curvas Principais
CPC	Curvas Principais Classifier
CSV	Comma Separated Values
ELM	Extreme Learning Machine
ENEM	Exame Nacional do Ensino Médio
F1	F1-Score
Fies	Fundo de Financiamento ao Estudante do Ensino Superior
FN	Falsos Negativos
FP	Falsos Positivos
GWO	Grey Wolf Optimizer
IA	Inteligência Artificial
IDH	Índice de Desenvolvimento Humano
IES	instituições de Ensino Superior
IFES	instituições Federais de Ensino Superior
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbors
KS2_MAX	K-squared Maximum
LC	Linguagens, Códigos e suas Tecnologias
LK	Lazy Kstar
MEC	Ministério da Educação
MLP	Multilayer Perceptron Networks

MT	Matemática e suas Tecnologias
NB	Naive Bayes
NM	Near Miss
PCA	Principal Component Analysis
PR	Precisão
Prouni	Programa Universidade para Todos
RE	Recall
RED	Redação
RF	Random Forest
RL	Regressão Loística
RLM	Regressão Logística Multinomial
RLS	Regressão Linear *Stepwise*
SFS	Sequential Feature Selector
SHAP	SHapley Additive exPlanations
Sisu	Sistema de Seleção Unificada
STLS	Spatial Two-Stage Least Squares
t-SNE	t-Distributed Stochastic Neighbor Embedding
TVP	Taxa de Verdadeiros Positivos
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos
XGBoost	eXtreme Gradient Boosting
XLS	Excel Spreadsheet
XLSX	Excel Open XML Spreadsheet

LISTA DE SÍMBOLOS

α	alfa
β	beta
δ	delta
ω	ômega
$\vec{X}_p$	Vetor da Posição da Presa
$\vec{X}$	Vetor da Posição de um Lobo Cinzento
$\vec{r}_1$	Vetor aleatório em [0,1]
$\vec{r}_2$	Vetor aleatório em [0,1]
$\vec{D}$	Distância Vetorial
$\vec{X}_\alpha$	Localização de α
$\vec{X}_\beta$	Localização de β
$\vec{X}_\delta$	Localização de δ
$\vec{C}_1$	Vetor Arbitrário
$\vec{C}_2$	Vetor Arbitrário
$\vec{C}_3$	Vetor Arbitrário
$\vec{D}_\alpha$	Distância Vetorial do lobo alfa em relação à presa
$\vec{D}_\beta$	Distância Vetorial do lobo beta em relação à presa
$\vec{D}_\delta$	Distância Vetorial do lobo delta em relação à presa
$\vec{A}_1$	Vetor Gerado Aleatoriamente
$\vec{A}_2$	Vetor Gerado Aleatoriamente
$\vec{A}_3$	Vetor Gerado Aleatoriamente

SUMÁRIO

1 INTRODUÇÃO

A educação desempenha um papel crucial no crescimento e desenvolvimento de uma nação. A análise recente no Brasil destaca a necessidade de melhorias neste setor, que podem ser alcançadas com o suporte da Tecnologia da Informação (Silva; Morino; Sato, 2014). Neste contexto, a avaliação educacional assume um papel central, pois um de seus principais propósitos é assegurar a excelência na qualidade do ensino (Luckesi, 2014).

O Exame Nacional do Ensino Médio (ENEM) é o exame que busca avaliar as habilidades básicas e competências dos estudantes e, através do resultado obtido nele, possibilitar que os estudantes ingressem em instituições de ensino superior. No Brasil, o ENEM é o maior exame e o segundo maior vestibular do mundo, ficando atrás apenas do exame chinês *Gaokao*, que lidera em número de participantes (Silveira; Mauá, 2018).

O desempenho dos estudantes no ENEM possibilita a realização de estudos e a criação de indicadores a respeito da educação no Brasil. O ENEM possui grande importância para a sociedade, pois muitos estudantes vêem nele uma das principais oportunidades de ingressar no ensino superior. Devido às dificuldades socioeconômicas enfrentadas por alguns estudantes, o ENEM se destaca como o exame que oferece possibilidades de acesso à graduação em instituições públicas ou privadas, sem a necessidade de arcar com o custo total do curso ou com apenas uma parte dele em alguns casos (Noguera; Aguiar, 2023; Júnior, 2023).

O uso de ferramentas de análise e processamento de dados tem se destacado em áreas essenciais para a sociedade, como a educação e a avaliação acadêmica. No caso do ENEM, essas tecnologias podem ajudar a identificar padrões de desempenho dos candidatos, apontar possíveis irregularidades nas respostas e até permitir a personalização de estratégias pedagógicas com base nos resultados das avaliações. Além disso, os métodos de reconhecimento de padrões e Aprendizado de Máquina (AM) estão sendo utilizados para prever o desempenho dos estudantes, tornando o processo de avaliação mais eficiente e preciso.

No âmbito do ENEM, a aplicação de ferramentas tecnológicas vai desde a análise automatizada das respostas até a melhoria dos processos de correção (Pinho *et al.*, 2024; Bertucci, 2021), avaliação (Nunes *et al.*, 2023) e predição do desempenho dos estudantes (Neto, 2023). Com o uso de métodos de análise e processamento de dados, torna-se viável monitorar padrões de desempenho dos candidatos, identificar tendências nos resultados e apontar possíveis inconsistências, assegurando maior precisão e transparência no processo avaliativo. Além disso, essas

ferramentas permitem extrair informações relevantes que contribuem para aprimorar a formulação das provas e ajustar os critérios de correção. O crescente uso dessas técnicas em avaliações educacionais tem colocado essa área de pesquisa em evidência. Entre os métodos utilizados 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 forma compacta e unidimensional (Kégl *et al.*, 2000).

Hastie & Stuetzle (1989) desenvolveram o conceito de CP como uma abordagem que é uma generalização não linear do método de Análise de Componentes Principais. Essas curvas suaves, de dimensão unidimensional, atravessam o centro de distribuições multidimensionais, permitindo uma representação mais compacta e adaptável da estrutura dos dados. As CP têm se mostrado uma ferramenta poderosa para o reconhecimento de padrões em conjuntos de dados complexos (Macêdo *et al.*, 2025; Sousa *et al.*, 2020; Ferreira *et al.*, 2015; Ferreira *et al.*, 2013). Elas possuem a capacidade de representação, uma vez que conseguem modelar classes com distribuições complexas, sejam elas alongadas, esféricas ou circulares (Kégl *et al.*, 2000). Além disso, uma curva pode representar melhor os dados do que um único centroide ou um único neurônio (como em redes neurais tradicionais), pois acompanha a estrutura contínua da distribuição (Kégl *et al.*, 2000; Moraes *et al.*, 2020).

Um dos algoritmos de extração de CP mais utilizados é o k-segmentos não suave (Verbeek; Vlassis; Kröse, 2001; Verbeek; Vlassis; Kröse, 2002), que realiza o processo de extração das 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 (Moraes; Ferreira, 2016a).

No entanto, ajustar os hiperparâmetros do método de CP K-segmentos é uma tarefa complexa que impacta diretamente a capacidade do modelo de capturar adequadamente a estrutura dos dados. Parâmetros como o número de segmentos, coeficiente de suavização e largura de banda (complexidade da curva) exigem calibração cuidadosa, pois valores inadequados podem comprometer significativamente o desempenho do modelo. Diversos trabalhos na literatura, como os de Syarif, Prugel-Bennett & Wills (2016), Bergstra & Bengio (2012a), Kégl *et al.*

(2000) e Meng & Eloyan (2021), destacam a importância da escolha apropriada de hiperparâmetros em modelos não lineares, como as CP, e ressaltam que a complexidade e a flexibilidade das CP estão fortemente ligadas aos valores dos hiperparâmetros, e que uma escolha inadequada pode levar ao sobreajuste ou à sub-representação dos dados.

Diante desse contexto, esta pesquisa apresenta duas contribuições principais. A primeira consiste em investigar o uso da meta-heurística Lobo Cinzento (*Grey Wolf Optimizer* (GWO)) para ajustar automaticamente os hiperparâmetros do método de CP K-segmentos, avaliando sua capacidade, primeiramente, em bases de dados simuladas e da literatura, com características distintas, como diferentes dimensões, números de classes, diferentes geometrias e graus de separabilidade. A segunda contribuição consiste em aplicar a abordagem proposta em um problema real, realizando a classificação do desempenho acadêmico dos estudantes no ENEM.

1.1 Objetivos

Nesta seção, são apresentados o objetivo geral e os objetivos específicos desta pesquisa.

1.1.1 Objetivo Geral

Esta pesquisa tem como objetivo aplicar o método de extração de CP K-segmentos para a classificação de dados sintéticos e de desempenho acadêmico dos estudantes que realizaram o ENEM 2023.

1.1.2 Objetivos Específicos

- Aplicar o método de otimização de hiperparâmetros GWO para determinar automaticamente os valores dos hiperparâmetros do método de CP K-segmentos.
- Avaliar a abordagem proposta em bases de dados simuladas e da literatura, com diferentes características, verificando sua robustez.
- Comparar o método de CP com métodos tradicionais (*Naïve Bayes* (NB), *Random Forest* (RF) e *Extreme Learning Machine* (ELM)) em problemas simulados e da literatura.

- Analisar os resultados obtidos pelo método proposto na classificação do desempenho dos estudantes que realizaram o ENEM 2023.
- Investigar a influência dos atributos presentes na base de dados do ENEM 2023 sobre os métodos de classificação.

1.2 Justificativa

Essa pesquisa se justifica, uma vez que o problema de prever o desempenho dos estudantes no ENEM tem sido investigado por diversos autores, porém muitos não têm explorado outras técnicas de AM. Nos últimos anos, técnicas de CP têm sido aplicadas em diversas áreas, demonstrando potencial em problemas de modelagem, previsão e classificação (Moraes *et al.*, 2020; Ferreira *et al.*, 2015).

1.3 Organização do trabalho

O restante da pesquisa está organizado da seguinte maneira: O Capítulo 2 apresenta a revisão da literatura relacionada ao tema abordado nesta pesquisa. No Capítulo 3 é apresentada a metodologia utilizada para o desenvolvimento desta pesquisa, incluindo as ferramentas computacionais, a preparação dos dados, o diagrama de trabalho e uma descrição dos dados do ENEM. Já no Capítulo 4 são apresentados os resultados desta pesquisa. Por fim, no Capítulo 5 são apresentadas as conclusões, os trabalhos futuros e as publicações realizadas. Destaca-se, também, que no Apêndice A são apresentados os demais atributos presentes na base de dados do ENEM 2023, que não foram considerados na pesquisa.

2 REFERENCIAL TEÓRICO

Neste capítulo, são descritos a área de estudo desta pesquisa, os principais tipos de Aprendizado de Máquina, os conceitos de Reconhecimento de Padrões, os métodos de redução de dimensionalidade e balanceamento de dados, bem como as técnicas de validação para aprimorar a generalização dos modelos. Além disso, são abordados os classificadores selecionados, as métricas de avaliação e, por fim, estudos relacionados à área de estudo, com foco no ENEM.

2.1 ENEM

Em 1998, no Brasil, foi criado pelo Ministério da Educação (MEC) o Exame Nacional do Ensino Médio (ENEM), com o intuito de avaliar as habilidades e competências desenvolvidas pelos estudantes que já concluíram ou que ainda estão concluindo o ensino médio. Visando auxiliar as escolas na construção do aprendizado dos estudantes, o ENEM é conduzido pelo MEC. Desde a sua criação, o ENEM tem buscado ir além de uma simples avaliação diagnóstica da educação brasileira, servindo também como uma ferramenta para ajudar os estudantes a fazerem escolhas alinhadas com suas habilidades. Além disso, o ENEM é amplamente utilizado como uma alternativa ou complemento a outros exames para ingresso no ensino superior e no mercado de trabalho (Santos, 2011).

O governo brasileiro tem implantado várias iniciativas em relação ao ENEM para a entrada no ensino superior, nas quais diversos programas utilizam a nota do ENEM como critério para a entrada em instituições de ensino superior. Existem três programas, denominados Sistema de Seleção Unificada (Sisu), Programa Universidade para Todos (Prouni) e Fundo de Financiamento ao Estudante do Ensino Superior (Fies). Por meio deles, os estudantes podem utilizar suas notas obtidas no ENEM para ingressar em universidades públicas. Também é possível conseguir uma bolsa de estudos ou financiamento ao optar por ingressar em universidades particulares (Westphalen-RS; Vargas, 2021).

Por meio do Sisu, os estudantes podem ingressar em universidades públicas, usufruindo do acesso gratuito às vagas disponibilizadas. Já o Prouni oferece bolsas de estudo para estudantes de baixa renda em instituições privadas, que podem ser parciais ou totais, dependendo da disponibilidade da instituição e dos critérios de seleção adotados. Por fim, o Fies é um programa

de financiamento do governo que permite aos estudantes pagarem os custos da graduação após a conclusão do curso (Westphalen-RS; Vargas, 2021).

A nota no exame serve como processo seletivo para aproximadamente 539 Instituições de Ensino Superior (IES). Com a modificação realizada pelo MEC, em 2010, o ENEM é usado como processo seletivo unificado para Instituições Federais de Ensino Superior (IFES). Até o ano de 2008, a prova tinha 63 questões em um dia e não era estruturada diretamente com base nos conteúdos curriculares do Ensino Médio, além disso, não existia a comparação das notas dos estudantes de um ano para o outro. Com o novo formato do exame a partir de 2010, existe a possibilidade de comparar o desempenho dos estudantes ao longo dos anos, podendo assim organizar e analisar o crescimento educacional no decorrer dos anos (Andriola, 2011).

O novo modelo do exame é dividido em cinco áreas de estudo (Tabela 2.1), Ciências Humanas e suas Tecnologias (CH), Ciências Naturais e suas Tecnologias (CN), Matemática e suas Tecnologias (MT), Linguagens, Códigos e suas Tecnologias (LC) e Redação (RED) feita na língua portuguesa e permite avaliar as habilidades em diversas áreas do conhecimento. As áreas de CH, CN, MT e LC contêm 45 questões de múltipla escolha cada, totalizando 180 questões, sendo aplicado em dois dias, juntamente com a RED, na qual é obrigatória a realização. Além disso, a pontuação máxima que pode ser atingida é de até 1000 pontos em cada área (Viggiano; Mattos, 2013).

Tabela 2.1 – Áreas do ENEM.

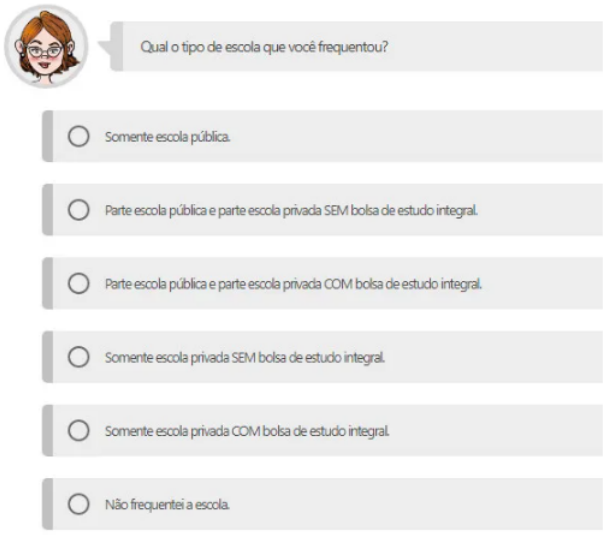
Área do conhecimento	**Componentes curriculares**
Linguagens, Códigos e suas tecnologias	Língua Portuguesa, Literatura, Língua Estrangeira (Inglês ou Espanhol), Artes, Educação Física e Tecnologias da Informação e Comunicação
Ciências Humanas e suas tecnologias	História, Geografia, Filosofia e Sociologia
Ciências da Natureza e suas tecnologias	Química, Física e Biologia
Matemática e suas tecnologias	Matemática

Fonte: Do Autor (2026).

Além das provas, para complementar a análise do desempenho dos estudantes, durante a inscrição no exame, os estudantes preenchem um questionário socioeconômico, relacionado com os dados socioeconômicos e demográficos deles. Nesse questionário, são realizadas perguntas como idade, cor/raça, renda familiar, tipo de escola que estudou (pública ou particular), escolaridade dos pais, condições de moradia, acesso a bens e serviços, região em que vivem, como é a situação social e econômica deles (Anjos, 2017). Essas informações obtidas atra-

vés do questionário socioeconômico servem para a elaboração de indicadores que revelam as desigualdades socioeconômicas e seu impacto no desempenho dos estudantes (Kleinke, 2017; Dutra; Júnior; Fernandes, 2023). Como exemplo, a Figura 2.1 ilustra uma das questões presentes no questionário no momento da inscrição, sendo o tipo de escola (pública ou privada) que o estudante frequenta ou frequentava e a existência de bolsa ou não, isso refletindo as influências do contexto socioeconômico sobre o acesso ao ensino.

Figura 2.1 – Tipo de Escola que o Estudante Estudou.



Fonte: Inep (2023).

2.2 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é um subcampo da Inteligência Artificial (IA) que constrói algoritmos computacionais, utilizando técnicas matemáticas e estatísticas, aprendendo com base em um banco de dados pré-definido, gerando ao final um modelo de classificação, detecção ou previsão (Mitchell, 2006). Dentro desse campo, destaca-se o Aprendizado Profundo (AP), que é um subcampo do AM. O AP consiste em redes neurais artificiais, construídas a partir de múltiplas camadas para modelar e aprender representações hierárquicas de dados. Essa abordagem permite simular o processo de aprendizagem humana, identificando padrões de forma iterativa e extraindo características de diferentes tipos de dados, como imagens, textos e áudios (Bezerra, 2016). Por meio de técnicas como retropropagação e otimização baseada em gradientes, os pesos das redes são ajustados para a minimização do erro e melhores desempenhos nas previsões. Isso permite que o AP estenda sua funcionalidade a novos dados,

tornando-o uma excelente ferramenta em muitas áreas, incluindo visão computacional, processamento de linguagem natural e reconhecimento de objetos, entre outros (Goodfellow, 2016; Dong; Wang; Abbas, 2021).

O desenvolvimento de um algoritmo de AM consiste em três etapas: pré-processamento, treinamento e avaliação do modelo. Na primeira etapa, o banco de dados é organizado, o problema que se deseja resolver é definido e os dados de treinamento e teste são divididos. O treinamento do modelo de AM pode ocorrer de maneira supervisionada, não supervisionada, semi-supervisionada, entre outros. Na etapa de avaliação, a saída prevista é adquirida com os resultados gerados pelo modelo. Portanto, os modelos de AM aprendem por meio de observações repetidas e estabelecem um padrão para criar um modelo capaz de generalizar as informações, para que novos dados sejam rotulados com precisão (Rajkomar; Dean; Kohane, 2019). É importante destacar que, na fase de desenvolvimento de um algoritmo de AM, um banco de dados consolidado deve ser utilizado para evitar a geração de resultados enganosos (Chen; Mao; Liu, 2014). Nesta seção, será realizada uma introdução sobre os principais tipos de aprendizado: supervisionado, não supervisionado, por reforço, semi-supervisionado, auto-supervisionado, por transferência e federado.

2.2.1 Aprendizado de Máquina Supervisionado

Consiste na utilização de um banco de dados no qual os dados são rotulados através do conhecimento de um profissional da área (Haykin, 2001). Neste processo de aprendizagem, infere-se uma regra para atribuir uma classe a um objeto nunca visto a partir de dados de treinamento previamente classificados. A aprendizagem supervisionada prevê uma saída para cada objeto de entrada, com base no padrão aprendido em cada uma das classes durante o treinamento realizado com dados rotulados (Mitchell, 1997). O aprendizado supervisionado trabalha com dois tipos principais de problemas: classificação e regressão. Na classificação, deseja-se prever um valor que pertence a uma determinada classe ou a um conjunto finito de possibilidades. Por exemplo, prever um tipo de animal (cachorro, gato, rato) usando dados de imagens. Já a regressão tenta prever um valor numérico nos dados, como o preço de um carro, utilizando variáveis como rodas, tipo, ano e quilometragem (Freitas, 2024).

2.2.2 Aprendizado de Máquina Não Supervisionado

É um tipo de aprendizado que utiliza um banco de dados no qual os dados não são rotulados. Desta forma, busca-se encontrar um padrão e determinar como os dados estão organizados (Mitchell, 1997). É frequentemente utilizado em casos onde há uma grande quantidade de dados, e rotular esses dados manualmente não é uma tarefa trivial. As técnicas de aprendizagem não supervisionada incluem quatro categorias principais de algoritmos, que são clusterização, redução de dimensionalidade, detecção de anomalias e regras de associação. Na clusterização, busca-se encontrar grupos que possuem características semelhantes. A redução de dimensionalidade simplifica ou comprime os dados sem que ocorra perda de informações significativas. Já a detecção de anomalias identifica padrões que não são frequentes ou anômalos em uma base de dados, como defeitos de fabricação de um equipamento, fraudes em cartões de crédito, entre outros. Por fim, nas regras de associação, são identificadas relações entre atributos de dados, como se os pais compram fraldas no final de semana também costumam comprar cervejas, ou se uma pessoa compra pipoca ela também costuma comprar batata frita (Freitas, 2024).

2.2.3 Aprendizado de Máquina Por Reforço

É uma área de aprendizado de máquina em que um agente aprende a tomar decisões em um ambiente interativo para maximizar uma recompensa cumulativa. Nesse aprendizado não existe um conjunto de dados pré-existente. A máquina interage com o ambiente para coleta desses dados, por intermédio das recompensas recebidas, aprende um padrão de ações em que as melhores recompensas são retornadas, ou seja, a máquina aprende buscando maximizar as recompensas. Diferente de métodos supervisionados, em que um modelo é treinado com dados rotulados, no aprendizado por reforço o agente explora o ambiente e recebe um feedback em forma de recompensas ou penalidades, para ajustar suas ações para melhorar o seu desempenho ao longo do tempo. Esse tipo de aprendizado tem aplicações em várias áreas, como robótica, jogos e controle de sistemas complexos, em que decisões são sequenciais (Ris-Ala, 2023; Freitas, 2024). Algumas definições importantes nesse tipo de aprendizado são:

- Agente: consiste na entidade que irá interagir com o ambiente, quem vai tomar as decisões. Como exemplo, um robô, gato, cachorro, entre outros.

- Ambiente: o meio no qual o agente irá interagir. O ambiente pode ser o mundo exterior, por exemplo, uma casa, um labirinto, não necessariamente precisa ser um espaço físico.
- Ação: comportamento do agente. Um movimento que ele pode realizar, como norte, sul, oeste e leste, ou pegar um objeto e largar.
- Estado: refere-se à configuração do agente em relação ao ambiente em que está inserido. Por exemplo, a localização de um robô em um ponto específico de um labirinto.
- Recompensa: corresponde à avaliação (positiva ou negativa) atribuída ao agente ao transitar para um novo estado. Por exemplo, receber 20 pontos.
- Política: estratégia de ação que orienta a transição de estados visando alcançar os melhores resultados. Por exemplo, a trajetória que o agente deverá seguir de forma mais eficiente.
- Episódio: conjunto completo de ações realizadas até alcançar um objetivo específico. Considerando que cada movimento equivale a um único passo, o episódio é formado pela sequência desses passos até atingir o estado final.

2.2.4 Aprendizado de Máquina Semi-Supervisionado

É uma abordagem que consiste no aprendizado a partir de dados rotulados e não rotulados. Podendo ser utilizado em problemas de classificação, em que exemplos rotulados ajudam na rotulagem de outros exemplos, como também em tarefas de clusterização, em que os exemplos rotulados auxiliam na formação dos grupos (Bruce, 2001). Esse tipo de aprendizado é útil em situações em que obter grandes quantidades de dados rotulados é custoso ou demorado, assim pelo fato de exemplos não rotulados existirem em grande abundância e os rotulados serem, na maioria das vezes, escassos.

O objetivo desse aprendizado é utilizar dados rotulados para extrair informações sobre o problema para serem utilizados no processo de aprendizado de dados não rotulados (Cai *et al.*, 2023). Esse processo permite que o modelo desenvolva representações mais robustas e generalizáveis, sendo aplicável em áreas como processamento de linguagem natural, visão computacional e reconhecimento de fala. Ao combinar informações de ambas as fontes de dados, o aprendizado semi-supervisionado busca melhorar a precisão e a eficiência, especialmente em

contextos nos quais os recursos de rotulagem são limitados (Sanches, 2003). Alguns algoritmos utilizados nesse tipo de aprendizado são o *COP-K-Means* (Wagstaff *et al.*, 2001), *SEEDED-K-Means* (Basu; Banerjee; Mooney, 2002), *CONSTRAINED-K-Means* (Basu; Banerjee; Mooney, 2002), entre outros.

2.2.5 Aprendizado de Máquina Auto-Supervisionado

O Aprendizado de Máquina Auto-Supervisionado é uma técnica de aprendizado de máquina composta por métodos que são capazes de aprender por meio de dados não rotulados, nos quais o próprio algoritmo gera tarefas auxiliares e utiliza partes das amostras originais como rótulos artificiais ao longo do processo de aprendizado (Liu *et al.*, 2021). Assim, esses métodos recebem esse nome devido à sua capacidade de gerar um "par" de entrada e saída, similar ao utilizado no treinamento de métodos supervisionados, a partir de um único dado de entrada não rotulado. Essas informações podem assumir diferentes formas e são utilizadas na etapa de pré-treino, a qual pode ser complementada ou não por uma etapa de *fine-tuning*, em que se aproveita a experiência do pré-treino.

No aprendizado auto-supervisionado, os procedimentos experimentais envolvem a definição de uma *pretext task*, que é uma tarefa aprendida durante o pré-treino, e de uma *downstream task*, destinada à aplicação do modelo em produção, após a etapa de *fine-tuning*, quando aplicável. A *pretext task* é elaborada com o objetivo de capacitar o modelo a aprender a extrair informações úteis nos dados de entrada, mapeadas em um espaço latente. Essas informações serão vantajosas para a tarefa final, a *downstream task*, que pode, por exemplo, envolver classificação. Esse conceito de aprendizado, no qual há um processo específico (ou uma rede dedicada) para a extração de informações ou *features*, seguido de outra etapa ou rede para realizar a tarefa final, é denominado na literatura como *representation learning* (Liu *et al.*, 2021).

2.2.6 Aprendizado de Máquina Por Transferência

O Aprendizado Por Transferência, ou também conhecido como *transfer learning*, tem como ideia utilizar um modelo que é pré-treinado em uma base de dados para resolver uma tarefa, para construir outro modelo, direcionado a resolver uma tarefa diferente (Taylor; Stone, 2009). Esse processo utiliza o conhecimento adquirido pelo primeiro modelo, permitindo trei-

nar o segundo com poucas amostras, reduzindo assim o esforço necessário para resolver uma nova tarefa (Pan; Yang, 2009; Torrey; Shavlik, 2010). Os três principais tipos de Aprendizado Por Transferência são transferência indutiva, aprendizado não supervisionado e transferência transdutiva.

A transferência indutiva é quando as tarefas de origem e de destino diferem, independentemente das semelhanças ou diferenças entre os conjuntos de dados de origem e de destino. O aprendizado multitarefa, que envolve o treinamento simultâneo de duas tarefas distintas (como a classificação de imagens e a detecção de objetos) com o mesmo conjunto de dados, pode ser classificado como transferência indutiva.

No caso do aprendizado não supervisionado, a abordagem pode ser relacionada à transferência indutiva, uma vez que, assim como nela, as tarefas de origem e de destino são distintas, ainda que partilhem a mesma fonte de dados ou representações extraídas. Entretanto, na transferência indutiva, os dados possuem rótulos. Já no aprendizado não supervisionado ocorre sem supervisão, ou seja, sem a presença de dados rotulados manualmente. Um exemplo é a detecção de fraudes, na qual o modelo aprende a identificar possíveis comportamentos de fraude.

Por fim, a transferência transdutiva ocorre quando as tarefas de origem e de destino são idênticas, mas as bases de dados são diferentes. Os dados de origem são rotulados, enquanto os dados de destino não possuem rótulos. Um exemplo de transferência transdutiva é a utilização de um modelo que classifica textos de avaliações de filmes para classificar as avaliações de carros.

2.2.7 Aprendizado de Máquina Federado

A medida que cada vez mais dispositivos estão conectados, incluindo *tablets*, *smartphones* e veículos, grandes quantidades de dados são geradas, e o uso de AM é necessário para explorar essas bases de dados de forma eficiente, autônoma e para responder a perguntas complexas. No entanto, muitas soluções de AM exigem bases de dados centralizadas, para uma melhor capacidade de treinamento. Além disso, essas soluções de AM, exigem custos elevados em termos de armazenamento, processamento centralizados e da preocupação associada à proteção da privacidade dos dados, uma vez que eles podem ser comprometidos durante o envio a servidores remotos (Rosano *et al.*, 2022).

Nesse contexto, surge o conceito de Aprendizado Federado (McMahan; Ramage, 2017), que propõe abordar essas limitações por meio da abordagem distribuída do Aprendizado de Máquina. A metodologia do Aprendizado Federado baseia-se no fato de que múltiplos dispositivos situados na borda da rede colaboram para treinar um modelo de predição, de forma compartilhada, na qual é treinado a partir de dados armazenados em cada equipamento. Dessa forma, os dados não são transferidos para servidores na nuvem, o que preserva a privacidade dos dados, que podem ser extremamente confidenciais em vários contextos.

Nessa abordagem, os dispositivos na borda da rede recebem o modelo global e realizam uma etapa de treinamento local baseada nos dados armazenados localmente. Após essa etapa, os modelos atualizados por cada cliente são retornados ao servidor central, onde são combinados os resultados e o novo modelo global é atualizado. Após isso, o novo modelo é redistribuído para os dispositivos na borda e o processo é repetido em cada nova rodada de treinamento (Bonawitz, 2019; Zhang *et al.*, 2021).

2.3 Reconhecimento de Padrões

O reconhecimento de padrões pode ser estudado em duas categorias principais (Tou; Gonzalez, 1974). A primeira categoria consiste na análise de seres humanos e organismos vivos, buscando compreender como eles desenvolvem e aperfeiçoam suas habilidades de reconhecimento de padrões. A segunda categoria foca na criação de teorias e técnicas destinadas ao desenvolvimento de máquinas ou dispositivos que reproduzam habilidades humanas nesse tipo de tarefa (Beyerer; Hagmanns; Stadler, 2024).

Um sistema de reconhecimento de padrões é dividido em três etapas. A primeira é a representação e mensuração dos dados de entrada, sendo a mensuração dos dados feita a partir do objeto a ser classificado, com o propósito de descrever os padrões do objeto e seu respectivo lugar em uma classe de padrões (Koutroumbas; Theodoridis, 2008; Sa, 2012). Em outras palavras, o vetor que descreve completamente um objeto deve ter dimensionalidade finita, isto é, é representado por um vetor C (Equação 2.1).

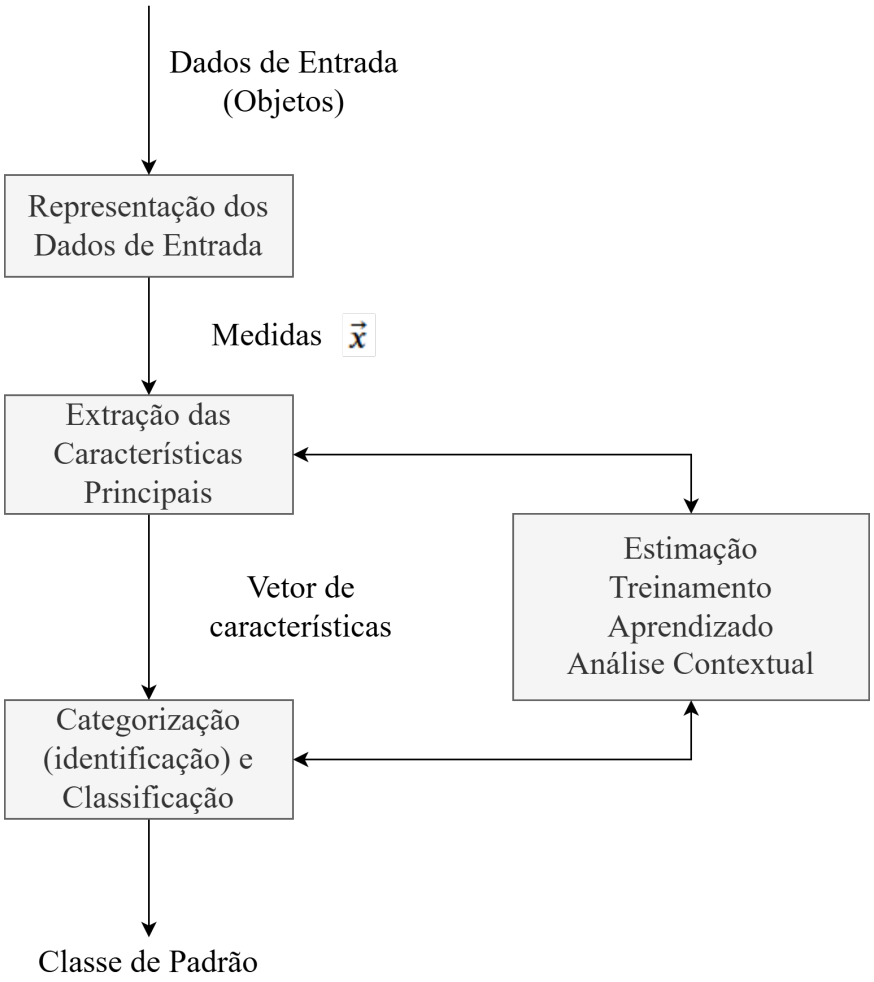
$$C = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_N \end{pmatrix} \tag{2.1}$$

em que $c_1, c_2, c_3, \ldots, c_N$ representam as características.

A segunda etapa é a extração das características intrínsecas e atributos do objeto, a fim de proporcionar a diminuição da dimensionalidade do vetor padrão. Essa fase é essencial para a identificação das características que afetam diretamente o desempenho do classificador. A seleção deve ser realizada com base nos aspectos específicos que se deseja classificar, implicando um conhecimento aprofundado do problema em questão. Esses são os principais objetivos desta etapa: reduzir a dimensão do vetor característico em respeito a informações relevantes para a classificação, reduzir o esforço computacional e selecionar atributos mais significativos para a tarefa de classificação (Koutroumbas; Theodoridis, 2008; Giraldi, 2021).

A terceira etapa do reconhecimento de padrões consiste em estabelecer procedimentos que possam ser utilizados para identificar o objeto e para classificar esse objeto dentro de uma determinada classe. O desenvolvimento do classificador nesta fase não é restrito e pode ser tratado de maneira mais abstrata e independente do tipo de problema. Isso ocorre porque os métodos, quando aplicados em áreas como reconhecimento de voz, análise de imagens, processamento de sinais de radar, inspeção de materiais e visão computacional, em muitos casos, são semelhantes. Essa característica permite que os métodos sejam utilizados em diferentes contextos sem comprometer seu desempenho (Koutroumbas; Theodoridis, 2008; Giraldi, 2021). A Figura 2.2 ilustra as etapas do reconhecimento de padrões.

Figura 2.2 – Diagrama com as etapas do Reconhecimento de Padrões.



Fonte: Do Autor (2026).

2.4 Métodos de Redução de Dimensionalidade

Com o aumento da quantidade e diversidade de dados disponíveis, a manipulação de conjuntos de dados de alta dimensão tornou-se um problema pertinente. Nesse sentido, os métodos de redução de dimensionalidade buscam a transformação de um conjunto de variáveis de entrada inicial em um conjunto reduzido, preservando ao máximo as informações relevantes existentes nos dados. Nesta seção, são apresentados os métodos de redução de dimensionalidade empregados nesta pesquisa.

2.4.1 Principal Component Analysis (PCA)

A Análise de Componentes Principais (*Principal Component Analysis* (PCA)) é uma técnica estatística que converte as variáveis originais em um novo conjunto de variáveis, denominadas componentes principais (Shlens, 2014). Essas componentes se constituem de forma ortogonal e estão organizadas de forma decrescente, conforme a quantidade de variância explicada dos dados. O principal objetivo do PCA é reduzir a dimensionalidade do conjunto de dados, conservando apenas as componentes principais mais significativas, aquelas que explicam a maior parte da variação nos dados (Jolliffe; Cadima, 2016). O funcionamento do PCA consiste em:

1. Centralização dos Dados: Na qual, inicialmente, é subtraída a média de cada atributo dos dados originais para centralizá-los em torno de zero. Dessa forma, garante-se que a análise seja baseada na variância e não nos valores absolutos (Equação 2.2).

$$X_{\text{centralizado}} = X - \mu \quad , \tag{2.2}$$

   em que X representa a matriz de dados, e μ é o vetor com as médias de cada variável.

2. Cálculo da Matriz de Covariância: Realiza-se o cálculo da matriz de covariância para entender como são as relações lineares entre as variáveis. Sejam $x_1, x_2, ..., x_n$ as variáveis, a matriz de covariância Σ é definida pela Equação 2.3.

$$\Sigma = \frac{1}{n-1} X^T_{\text{centralizado}} X_{\text{centralizado}} \quad . \tag{2.3}$$

3. Decomposição em Autovalores e Autovetores: É realizada a decomposição da matriz de covariância em seus autovalores e autovetores. Os autovalores consistem na variância explicada por cada componente principal. Já os autovetores são aqueles que definem a direção das componentes principais (Equação 2.4).

$$\Sigma v = \lambda v \quad , \tag{2.4}$$

   em que v é o autovetor e λ o autovalor correspondente.

4. Ordenação e Seleção de Componentes: Ocorre a ordenação de forma decrescente dos autovalores. Os maiores autovalores correspondentes às componentes principais são selecionados pelo fato de explicarem a maior parte da variância dos dados.

5. Projeção dos Dados: A projeção dos dados originais ocorre no espaço das componentes principais selecionadas (Equação 2.5).

$$X_{\text{reduzido}} = X_{\text{centralizado}} V_k \quad , \tag{2.5}$$

em que V_k é a matriz composta pelos k autovetores principais.

2.4.2 *Select K-Best*

O *Select K-Best* é uma técnica de seleção de atributos baseada em avaliação univariada, na qual cada variável do conjunto de dados é analisada individualmente quanto à sua associação com a variável-alvo. Essa avaliação é realizada por meio de testes estatísticos, como *ANOVA*, *F-value*, *Chi-square* e *Mutual Information*.

O teste de Análise de Variância (ANOVA) verifica a relação entre uma variável-alvo e cada variável do conjunto de dados, utilizando a análise de variância *ANOVA*. O *F-value* realiza a razão entre a variância explicada (entre grupos) e a variância residual (dentro dos grupos). Quanto mais alto for o valor do *F-value*, isso indica que aquela característica contribui significativamente para a separação entre as classes (Equação 2.6) (Edwards, 2005).

$$F = \frac{\text{Variância entre os grupos}}{\text{Variância dentro dos grupos}} \quad , \tag{2.6}$$

sendo a Variância entre os grupos $= \frac{\sum_{i=1}^{k} n_i(\bar{y}_i - \bar{y})^2}{k-1}$,
e a Variância dentro dos grupos $= \frac{\sum_{i=1}^{k} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2}{N-k}$,
onde, k representa o número de grupos, n_i indica o número de observações no grupo i, $\bar{y}_i$ é a média do grupo i, $\bar{y}$ é a média geral de todos os grupos, y_{ij} é o valor da j-ésima observação no grupo i e N indica o número total de observações.

O teste *Chi-square* requer a definição de um parâmetro K, que representa o número de características a serem selecionadas. A técnica atribui uma pontuação a cada atributo com base

em seu cálculo e remove iterativamente os atributos de menor pontuação, preservando apenas os K mais relevantes (Liu; Motoda, 2007). O (chi^2) é definido pela Equação 2.7.

$$chi^2 = \frac{(O-E)^2}{E} \quad . \tag{2.7}$$

em que O representa a frequência observada e E a frequência esperada de classe, assumindo que não exista relação entre o atributo analisado e o atributo alvo. Os atributos são considerados independentes quando a frequência observada é próxima da frequência esperada. As variáveis utilizadas no treinamento do algoritmo são selecionadas com base nas maiores pontuações obtidas em relação à variável alvo, indicando uma forte associação (Liu; Motoda, 2007; Theng; Bhoyar, 2024).

Já o teste *Mutual Information* avalia a quantidade de informação mútua entre uma característica e a variável destino, isto é, a quantidade de informação que a variável preditora consegue fornecer sobre a variável-alvo (Cover, 1999). Pode ser utilizado para variáveis contínuas ou categóricas. Quanto maior o valor, maior a dependência entre a característica e a variável-alvo (Equação 2.8).

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad , \tag{2.8}$$

em que $p(x,y)$ representa a distribuição conjunta, $p(x)$ e $p(y)$ são as distribuições marginais.

2.4.3 *t-Distributed Stochastic Neighbor Embedding (t-SNE)*

O método *t-Distributed Stochastic Neighbor Embedding* (t-SNE) é uma abordagem de extração de características não linear, não supervisionada e baseada em variedade, em que dados de alta dimensão são mapeados para baixa dimensão (normalmente 2 ou 3 dimensões), preservando ao mesmo tempo a estrutura significativa dos dados originais (Maaten; Hinton, 2008). O t-SNE é utilizado para visualização e exploração de dados. Ou seja, ele permite ter uma ideia de como os dados são organizados no espaço de alta dimensão. O método t-SNE torna-se útil para visualizar dados de alta dimensão, mantendo a estrutura e a significância dos dados.

O método aplica primeiro o *Stochastic Neighbor Embedding* (SNE) ao conjunto de dados, que converte as distâncias euclidianas de alta dimensão em probabilidades condicionais que representam semelhanças para cada par de dados (Devassy; George, 2020). A similaridade

dos dados x_a com os dados x_b é representada pela probabilidade condicional $p_{a|b}$, representada pela Equação 2.9.

$$p_{a|b} = \frac{\exp\left(-\frac{\|\mathbf{x}_b - \mathbf{x}_a\|^2}{2\sigma^2}\right)}{\sum_{k \neq a} \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{x}_a\|^2}{2\sigma^2}\right)} \quad . \tag{2.9}$$

A Equação 2.9 mede o quão próximo um ponto de dados x_a está de outro ponto de dados x_b sendo considerado uma distribuição gaussiana em torno x_b com uma determinada variância σ^2. A variação difere para cada dado e é escolhida de forma que os dados em áreas densas tenham menor variação do que os dados em áreas esparsas.

Posteriormente, em vez de utilizar a distribuição gaussiana, uma (Distribuição t de *Student*) com um grau de liberdade, semelhante à Distribuição de *Cauchy*, é usada para obter o segundo conjunto de probabilidades $(Q_{a|b})$ no espaço de baixa dimensão. Se na alta dimensão os dados x_a e x_b estão mapeados corretamente para dados de baixa dimensão y_a e y_b, então a semelhança entre $P_{a|b}$ e $Q_{a|b}$ torna-se igual. Portanto, o t-SNE minimiza a diferença entre essas duas probabilidades de espaços de baixa dimensão para espaços de alta dimensão (Belkina *et al.*, 2019; Devassy; George, 2020). Essa diferença é medida otimizando a função de custo (ϕ) da soma da divergência de *Kullback-Leibler* de acordo com a Equação 2.10.

$$\phi = \sum_a \sum_b P_{a|b} \log \frac{P_{a|b}}{Q_{a|b}} \quad , \tag{2.10}$$

em que ϕ é a função objetivo a ser minimizada no t-SNE, $P_{a|b}$ representa a probabilidade condicional no espaço de alta dimensionalidade, que mede a similaridade entre os pontos a e b, $Q_{a|b}$ representa a probabilidade condicional no espaço de baixa dimensionalidade, que mede a similaridade entre os pontos projetados a e b, $\log \frac{P_{a|b}}{Q_{a|b}}$ é o cálculo da diferença entre as distribuições P e Q para cada par de pontos, usada como medida de divergência.

2.5 Computação Bioinspirada

A computação bioinspirada é um campo de estudo que utiliza a natureza e seus padrões como referência para criar métodos destinados à solução de problemas. A variedade de tecnologias desenvolvidas a partir dessa inspiração é extensa, incluindo desde objetos simples, como o velcro, derivado da análise de estruturas encontradas em certas plantas, até sistemas sofistica-

dos, como aviões e submarinos, projetados com base no estudo do voo das aves e da locomoção dos peixes, respectivamente. As Redes Neurais Artificiais, baseadas no funcionamento dos neurônios humanos, e a Lógica Nebulosa, fundamentada na habilidade linguística do raciocínio humano, são proeminentes no campo da computação (Xavier, 2023).

Além disso, a natureza serve como fonte de inspiração para a criação de métodos heurísticos de otimização. Zou, Chen & Xu (2019) afirmam que a ideia dessas técnicas surge da demanda por encontrar soluções de ótimo global em problemas não diferenciáveis, em que os métodos tradicionais baseados em gradiente se revelam ineficientes. A classificação desses algoritmos pode ser feita de acordo com diferentes critérios. Do ponto de vista do fenômeno natural modelado, é comum dividi-los em duas categorias principais: Algoritmos Evolucionários, como os Algoritmos Genéticos e Evolução Diferencial, e algoritmos baseados em inteligência de populações, como o Algoritmo de Colônia de Formigas (ACO), Otimização do Lobo Cinzento (GWO) e Otimização por Enxame de Partículas (PSO).

Além dos modelos estabelecidos, destacam-se os algoritmos baseados nos princípios dos sistemas imunológicos, assim como em uma variedade de outros mecanismos naturais. A adoção dessas estratégias pode levar a desempenhos superiores quando aplicadas de maneira combinada, permitindo a preservação e a integração das características mais importantes de cada abordagem individual. Embora haja uma infinidade de variantes, o funcionamento dos algoritmos bioinspirados pode, de maneira geral, ser organizado nas seguintes fases:

- **Passo 0 - Inicialização e definição da população**: o algoritmo começa com a criação de uma população inicial, geralmente formada de forma estocástica para assegurar a diversidade no espaço de busca. Nesse cenário, o conjunto de candidatos que pode resolver o problema de otimização em questão é chamado de população.

- **Passo 1 - Seleção dos indivíduos mais aptos**: após avaliar a qualidade das soluções que compõem a população, selecionam-se as que melhor se adaptam ao problema em questão. No entanto, os indivíduos de menor desempenho não são totalmente eliminados, uma vez que sua remoção completa poderia levar o algoritmo a convergir prematuramente para ótimos locais, impedindo a descoberta de soluções globais mais eficazes. A preservação parcial desses indivíduos é fundamental para manter a diversidade populacional, que é um fator determinante para garantir o equilíbrio entre exploração e intensificação no processo evolutivo.

- **Passo 2 - Adaptação dos indivíduos**: nesta etapa, há mudanças nos indivíduos da população, tanto por meio da troca de informações entre eles quanto por transformações estruturais em cada um. Esse procedimento é, normalmente, realizado por meio da interação entre os membros mais bem avaliados da população, escolhidos na fase anterior, e os demais indivíduos, a fim de favorecer a disseminação das características mais promissoras.

- **Passo 3 - Redefinição da população**: os indivíduos criados na etapa anterior são adicionados ao grupo original, o que resulta na atualização da população. Esse procedimento marca a transição para uma nova geração no processo evolutivo, garantindo a continuidade da exploração no espaço de soluções. Ao incorporar tanto indivíduos já selecionados quanto os recém-criados, mantém-se o equilíbrio entre a exploração de novas regiões e o aprofundamento em áreas com potencial, o que é essencial para a eficácia dos algoritmos bioinspirados.

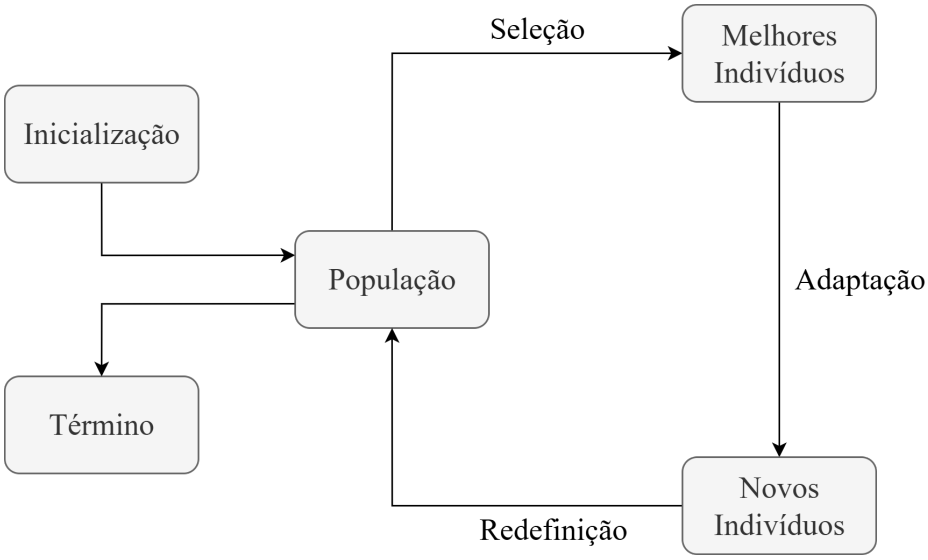
- **Passo 4 - Verificação dos critérios de parada**: nesta etapa, avalia-se se as condições para encerrar o processo evolutivo foram atendidas, seja obtendo soluções que atendam aos critérios de qualidade definidos pelo usuário, seja atingindo o número máximo de iterações estipulado. Se essas condições não forem atendidas, o algoritmo retorna à etapa inicial (Passo 1), reiniciando o ciclo de busca e permitindo que o processo de otimização continue.

O algoritmo apresentado nas etapas anteriores e ilustrado na Figura 2.3 serve como um arcabouço conceitual de natureza geral para algoritmos bioinspirados. No entanto, a implementação real pode apresentar várias variações, tanto devido às particularidades inerentes a diferentes tipos de problemas quanto à natureza específica do mecanismo biológico que inspirou sua criação.

2.6 *Grey Wolf Optimization* (GWO)

A meta-heurística conhecida como Otimização do Lobo Cinzento (*Grey Wolf Optimizer* (GWO)), apresentada por Mirjalili, Mirjalili & Lewis (2014), baseia-se no comportamento de caça e na estrutura social dos lobos cinzentos. Essa espécie (*canis lupus*), integrante da família dos canídeos, é classificada como predadora e ocupa uma posição destacada no topo da cadeia

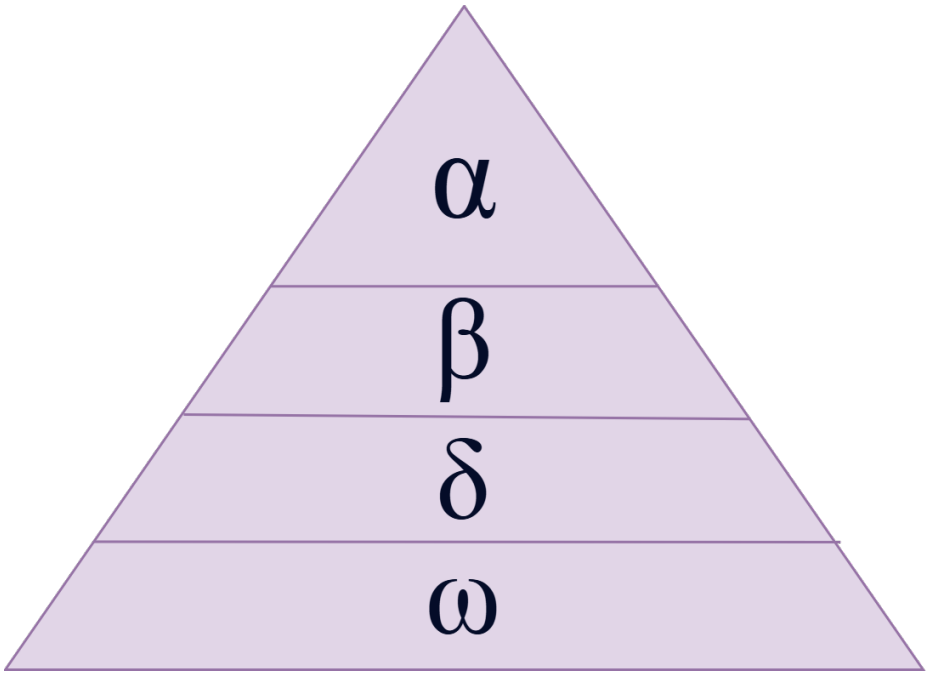
Figura 2.3 – Representação em fluxograma de um algoritmo bioinspirado.



Fonte: Adaptado de Eiben & Smith (2015).

alimentar. Normalmente, os lobos cinzentos organizam-se em bandos com uma média de 5 a 12 membros. Um aspecto relevante é que esses grupos apresentam uma hierarquia social bastante rígida e bem definida, conforme ilustrado na Figura 2.4.

Figura 2.4 – Hierarquia social dos lobos cinzentos (a posição de dominância decresce do topo para a base).



Fonte: Adaptado de Oliveira (2018).

Os lobos alfa (α), tanto machos quanto fêmeas, ocupam o posto mais elevado na hierarquia da matilha, desempenhando um papel de liderança e controle sobre o grupo. Eles são os

únicos com permissão para se reproduzir e também são encarregados de tomar as decisões mais importantes para a sobrevivência da comunidade. Curiosamente, o lobo alfa não precisa ser o mais forte fisicamente, mas sim aquele com maior habilidade em liderar e organizar o grupo. Isso evidencia que a coesão e a disciplina da matilha têm um peso maior do que a simples demonstração de força. Logo abaixo dos alfas estão os lobos beta (β), que desempenham um papel de apoio no processo de tomada de decisão. O beta pode ser tanto macho quanto fêmea e, geralmente, é o membro mais apto a assumir a posição de um lobo alfa, caso haja a ausência de um. No terceiro nível da hierarquia encontram-se os lobos delta (δ), que, apesar de serem subordinados aos alfas e betas, possuem uma posição intermediária, superior aos lobos ômega (ω). Por sua vez, os ômegas ocupam o patamar mais baixo da hierarquia, devendo-se submeter a todos os outros membros dominantes do grupo (Mirjalili; Mirjalili; Lewis, 2014).

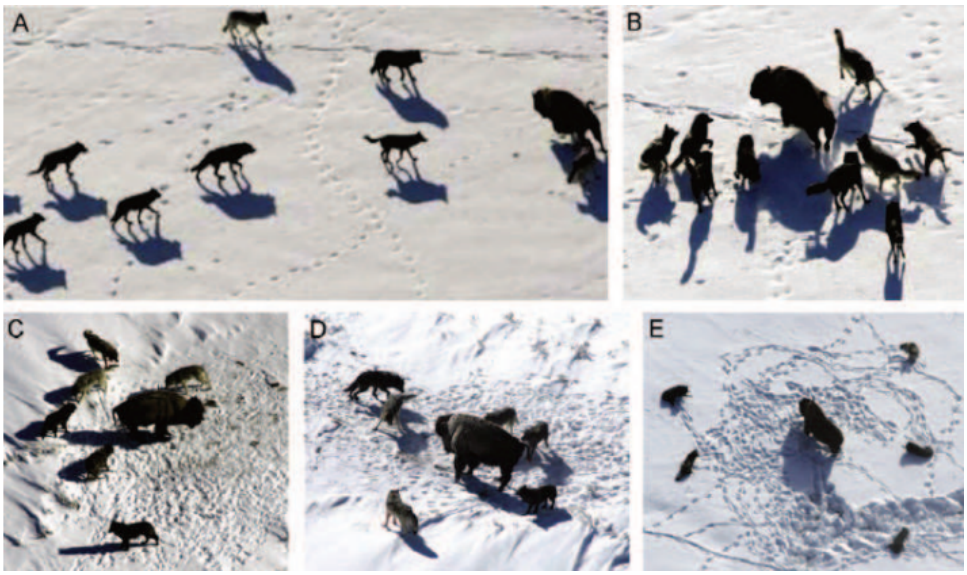
Além da estrutura hierárquica, a caça coletiva constitui outro aspecto social relevante no comportamento dos lobos cinzentos. Segundo Muro *et al.* (2011), as principais fases desse processo estão descritas na Figura 2.5 e podem ser organizadas da seguinte forma:

(A) identificar, seguir e aproximar-se da presa;

(B–D) persegui-la, cercá-la e fatigá-la até que não consiga mais fugir;

(E) realizar o ataque final.

Figura 2.5 – Táticas de caça dos lobos cinzentos.



Fonte: Retirado de Oliveira (2018).

2.6.1 Modelo Matemático e Algoritmo

No processo de otimização, os lobos atualizam suas posições em relação aos indivíduos α, β e δ da seguinte forma (Equação 2.11 a 2.14):

$$\overrightarrow{D} = \mid \overrightarrow{C}\overrightarrow{X_p}(t) - \overrightarrow{X}(t) \mid \quad , \tag{2.11}$$

$$\overrightarrow{X}_{(t+1)} = \overrightarrow{X_p}(t) - \overrightarrow{A}\overrightarrow{D} \quad , \tag{2.12}$$

$$\overrightarrow{A} = 2a\overrightarrow{r_1} - a \quad , \tag{2.13}$$

$$\overrightarrow{C} = 2\overrightarrow{r_2} \quad , \tag{2.14}$$

em que t é a época mais recente, $\vec{X}_p$ é o vetor da posição da presa, $\vec{X}$ é o vetor da posição de um lobo cinzento, $\overrightarrow{C}$ e $\overrightarrow{A}$ são vetores de coeficientes que controlam como o lobo atualiza sua posição em relação à presa, $\vec{r}_1$ e $\vec{r}_2$ são vetores aleatórios em [0,1], a é um paramêtro que diminui linearmente de 2 a zero e $\vec{D}$ é a distância vetorial de um lobo em relação à presa. O algoritmo GWO considera que α, β e δ representam, de forma aproximada, a posição da presa (solução ótima). Durante o processo de otimização, as três melhores soluções encontradas até o momento são atribuídas a α, β e δ, respectivamente. Os demais lobos, classificados como ω, ajustam suas posições tomando como referência esses três líderes. O modelo matemático para o reajuste da posição dos lobos ω é apresentado nas Equações 2.15 a 2.17.

$$\overrightarrow{D_\alpha} = \mid \overrightarrow{C_1}\overrightarrow{X_\alpha} - \overrightarrow{X} \mid \quad , \tag{2.15}$$

$$\overrightarrow{D_\beta} = \mid \overrightarrow{C_2}\overrightarrow{X_\beta} - \overrightarrow{X} \mid \quad , \tag{2.16}$$

$$\overrightarrow{D_\delta} = \mid \overrightarrow{C_3}\overrightarrow{X_\delta} - \overrightarrow{X} \mid \quad , \tag{2.17}$$

em que $\vec{X}_\alpha$, $\vec{X}_\beta$ e $\vec{X}_\delta$ representam os vetores posição dos lobos α, β e δ, respectivamente, isto é, das três melhores soluções candidatas da população. Já $\vec{C}_1$, $\vec{C}_2$, $\vec{C}_3$ são vetores de coeficientes aleatórios, o vetor $\vec{X}$ denota a posição da solução candidata em avaliação e $\vec{D_\alpha}$, $\vec{D_\beta}$ e $\vec{D_\delta}$ são as distâncias vetoriais entre essa solução e as posições de α, β e δ, respectivamente, sendo utilizadas para orientar a atualização das posições dos lobos no espaço de busca.

As Equações 2.15, 2.16 e 2.17 determinam uma estimativa da distância entre a solução atual e α, β e δ. Com base nessas definições, a nova posição da solução é obtida pelas Equações 2.18 a 2.21.

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}\overrightarrow{D_\alpha} \quad , \tag{2.18}$$

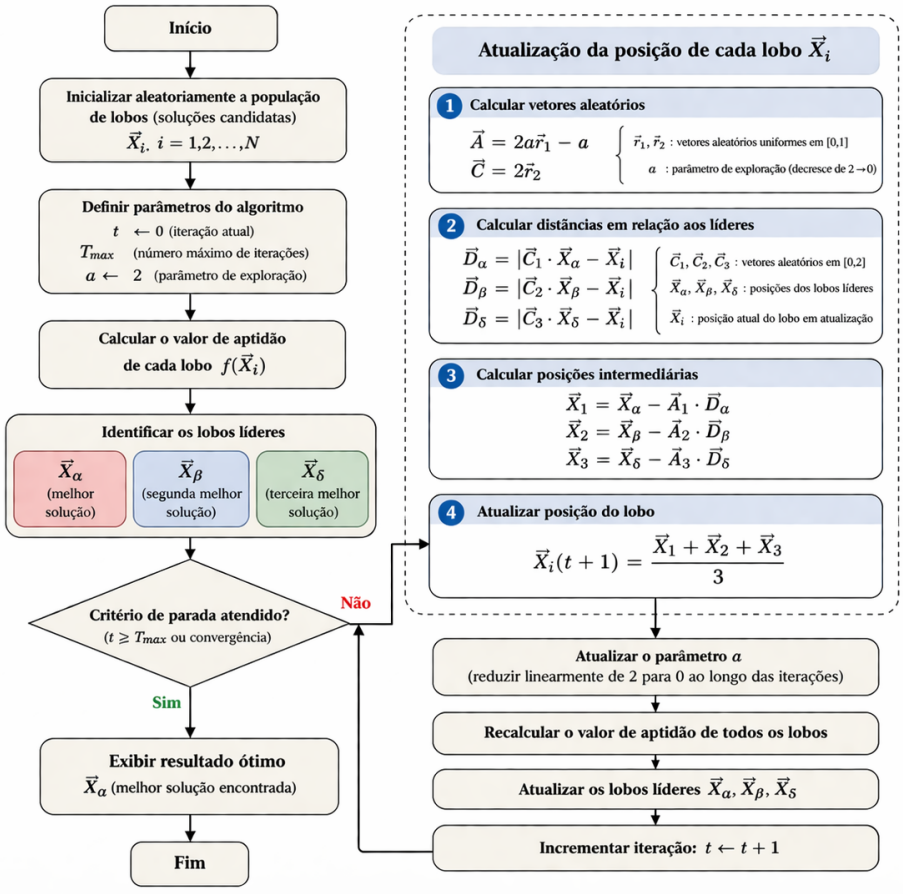
$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}\overrightarrow{D_\beta} \quad , \tag{2.19}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}\overrightarrow{D_\delta} \quad , \tag{2.20}$$

$$\overrightarrow{X}_{(t+1)} = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \quad , \tag{2.21}$$

em que $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, $\overrightarrow{X_\delta}$ mostram a localização atual dos lobos α, β e δ. Já $\vec{A}_1$, $\vec{A}_2$, $\vec{A}_3$ são vetores gerados aleatoriamente e t indica a quantidade de épocas. A Figura 2.6 apresenta o fluxograma do algoritmo GWO.

Figura 2.6 – Fluxograma do algoritmo GWO.



Fonte: Adaptado de Oliveira (2018).

A estrutura geral do GWO encontra-se representada no Algoritmo 1. Esse pseudocódigo sintetiza as operações fundamentais do método, desde a geração inicial da população de lobos até a seleção final da melhor solução, destacando o papel dos parâmetros de exploração (α, A e C) e da atualização iterativa das posições dos agentes.

Algorithm 1 Pseudo-código da Meta-Heurística GWO. Adaptado de Oliveira (2018).

Inicializar aleatoriamente a população de lobos cinzentos $\vec{X}_i$ ($i = 1, 2, \ldots, N$)
Inicializar $t \leftarrow 0$
Inicializar $T_{\max} \leftarrow$ número máximo de iterações
Calcular o valor de aptidão (*fitness*) de cada lobo
$a \leftarrow$ parâmetro de exploração (decresce de 2 a 0)
$\vec{A}$ e $\vec{C} \leftarrow$ componentes aleatórios que auxiliam na geração de soluções
$\vec{X}_\alpha \leftarrow$ melhor lobo (maior aptidão)
$\vec{X}_\beta \leftarrow$ segundo melhor lobo
$\vec{X}_\delta \leftarrow$ terceiro melhor lobo
while $t < T_{\max}$ **do**
    for cada lobo de procura **do**
        Calcular os vetores $\vec{A}$ e $\vec{C}$
        Calcular $\vec{D}_\alpha$, $\vec{D}_\beta$ e $\vec{D}_\delta$
        Calcular $\vec{X}_1$, $\vec{X}_2$ e $\vec{X}_3$
        Atualizar a posição do lobo:
        $$\vec{X}_{i(t+1)} \leftarrow \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$$
    end for
    Atualizar os parâmetros a, $\vec{A}$ e $\vec{C}$
    Recalcular a aptidão de todos os lobos
    Atualizar $\vec{X}_\alpha$, $\vec{X}_\beta$ e $\vec{X}_\delta$
    $t \leftarrow t + 1$
end while
Retorna $\vec{X}_\alpha$

O GWO é uma meta-heurística que apresenta um bom equilíbrio entre exploração, tem poucos hiperparâmetros a serem ajustados em comparação com outras meta-heurísticas e demonstra um desempenho competitivo em várias funções padrão e problemas de engenharia (Nasir *et al.*, 2024, 2024; Bergstra; Bengio, 2012b). Por esse motivo, tende a identificar bons conjuntos de hiperparâmetros com menos avaliações do que nas buscas exaustivas (Bergstra; Bengio, 2012b). Problemas que adotam algoritmos de busca exaustiva são afetados pela exploração combinatória (o número de tentativas aumenta exponencialmente com os hiperparâmetros), consomem avaliações em dimensões de pouca relevância e desperdiçam avaliações em dimensões pouco relevantes (He; Wu, 2023). O GWO se destaca pela sua baixa exigência computacional e facilidade de implementação (Nasir *et al.*, 2024).

2.7 Balanceamento de Dados

Um dos problemas mais comuns em bases de dados é o desbalanceamento, que ocorre quando uma classe corresponde a um número muito maior de amostras do que as demais. Para lidar com esta situação, uma abordagem bastante utilizada é o *Undersampling*. O *Undersampling* consiste em diminuir o número de amostras da classe majoritária, ajustando-a para que a classe majoritária seja igual em termos de amostras da classe minoritária. Nesta seção, é explicado o método *Near Miss* que utiliza a técnica *Undersampling*, que trata o problema de desbalanceamento em bases de dados. Ressalta-se que, nesta pesquisa, optou-se por utilizar apenas *Undersampling*, uma vez que a base de dados possui um número elevado de amostras. Nesse contexto, a redução controlada da classe majoritária permite equilibrar as classes sem comprometer a representatividade global do conjunto de dados, além de reduzir o custo computacional do treinamento.

2.7.1 *Near Miss* (NM)

Near Miss (NM) é baseado na distância dos registros da classe majoritária aos registros da classe minoritária. É uma abordagem de k vizinhos mais próximos, na qual a distância euclidiana (Equação 2.22) pode ser utilizada.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad , \tag{2.22}$$

em que $d(x,y)$ representa a distância euclidiana entre os vetores x e y.

A menor distância média entre a classe majoritária e os três registros mais próximos da classe minoritária é utilizada para identificar as amostras da classe majoritária que serão excluídas (Batista; Prati; Monard, 2004; Brownlee, 2020). Existem algumas variações do *NearMiss* (*NearMiss-1*, *NearMiss-2* e *NearMiss-3*), cada uma adotando um critério específico para decidir quais amostras da classe majoritária deverão ser mantidas.

O *NearMiss-1* faz a seleção das amostras que são da classe majoritária que possuem as menores distâncias médias para as amostras da classe minoritária que são mais próximas. Já o *NearMiss-2* faz a seleção das amostras que são da classe majoritária que possuem as menores distâncias médias para as amostras que são mais distantes da classe minoritária. Por fim, o

NearMiss-3 para cada amostra da classe minoritária, é selecionado um número fixo de amostras da classe majoritária que são mais próximas dela. A Equação 2.23 ilustra o critério do *NearMiss-1*.

$$\bar{d_{ij}} = \frac{1}{k}\sum_{l=1}^{k} d(x_i, y_l) \quad , \tag{2.23}$$

em que $\bar{d_{ij}}$ representa a média das distâncias entre a amostra x_i (classe majoritária) e as k amostras mais próximas y_l da classe minoritária, k é o número de amostras mais próximas consideradas e $d(x_i, y_l)$ é a distância euclidiana entre x_i e y_l.

2.8 Técnicas de Validação dos Métodos

Validar um modelo é uma etapa essencial para avaliar sua capacidade preditiva. Nesta seção, são apresentadas algumas técnicas destinadas à validação de modelos.

2.8.1 *Hold-out*

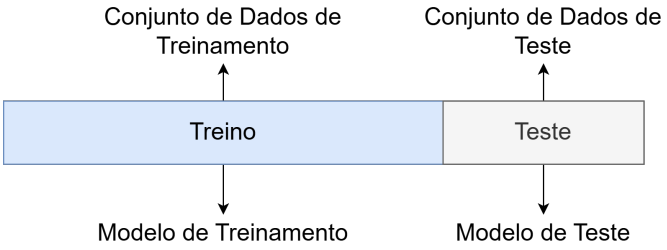
Introduzido por (Devroye; Wagner, 1979), o método *hold-out* também é denominado de validação simples. Ele realiza a divisão de um conjunto de dados d em duas partes, uma para treinamento e outra para validação. Nessa divisão, é utilizada uma proporção p, na qual se determina a quantidade de dados que será destinada à validação.

Considerando um conjunto de dados com n amostras, seleciona-se uma proporção p dos dados para que seja formado um conjunto de validação d_v, no qual o tamanho será $v = p * n$. O restante dos dados $(1-p)$ é utilizado para compor o conjunto de treinamento (d_t), cujo tamanho é $t = (1-p) * n$. O estimador *hold-out* é dado pela Equação 2.24.

$$hop^{-1} = \frac{1}{v}\sum_{i=1}^{v} L(y_i, \hat{f}_t(x_i)) \quad , \tag{2.24}$$

em que $\hat{f}_t(X)$ representa o preditor que é criado com a amostra de treino d_t e L a função de perda que é avaliada em todos os pontos $(y_i, \mathbf{x}_i)$ da amostra de validação d_v. A Figura 2.7 ilustra um exemplo do *Hold-out*, em que 70% dos dados são utilizados para treinamento e 30% para teste.

Figura 2.7 – Ilustração do *Hold-out*.



Fonte: Do Autor (2026).

2.8.2 *K-fold*

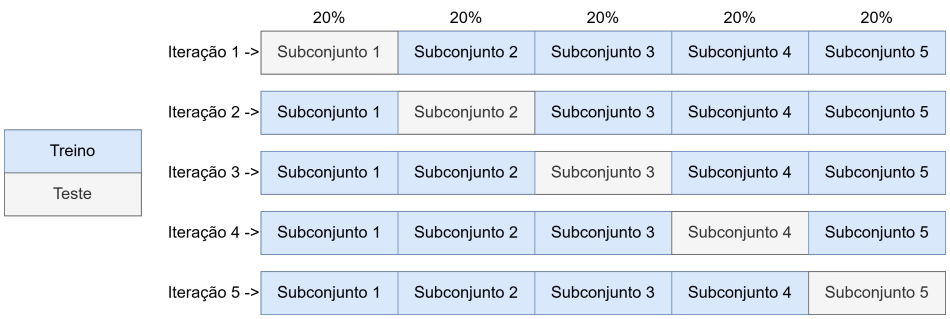
No método *K-fold* (Burman, 1989), o conjunto de dados d é dividido em K partes $(d_1, d_2, \ldots, d_K)$, cada uma com tamanho aproximado m_k, onde $\sum_{k=1}^{K} m_k = n$, na qual o número total de amostras do conjunto de dados é representado por n. O procedimento consiste em K iterações, em que, a cada iteração, uma das partes d_k, para $k = 1, 2, \ldots, K$, é utilizada como amostra de validação, enquanto as $K-1$ partes restantes compõem a amostra de treinamento, representada por $d_{(-k)} = \{d_1, d_2, \ldots, d_{k-1}, d_{k+1}, \ldots, d_K\}$.

Dessa forma, ao final das K iterações, todas as partes do conjunto de dados terão sido usadas tanto para treinamento quanto para validação. O método *K-fold* é representado por:

$$kf = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{m_k} \sum_{i=1}^{m_k} L(y_{ik}, \hat{f}_{(-k)}(x_{ik})) \quad , \tag{2.25}$$

em que kf é o estimador *K-fold*, o preditor $\hat{f}_{(-k)}(X)$ é construído utilizando a amostra de treinamento $d_{(-k)}$, e depois avaliado nas observações pertencentes à amostra de teste d_k para $k = 1, 2, \ldots, K$. A Figura 2.8 ilustra um exemplo do *K-fold* com k igual a 5.

Figura 2.8 – Ilustração do *K-fold* com k igual a 5.



Fonte: Do Autor (2026).

2.9 Métodos de Classificação

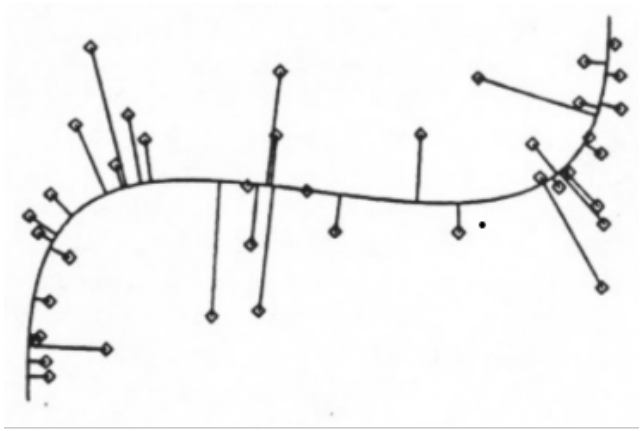
Nesta seção, são apresentados os métodos de classificação de AM supervisionado que foram empregados nesta pesquisa.

2.9.1 Curvas Principais (CP)

2.9.1.1 Conceito de Curvas Principais

Sob a perspectiva computacional, a obtenção das CP de um conjunto de dados inicia-se com a determinação da primeira componente principal como referência para a construção da curva (Jolliffe; Cadima, 2016; Beinert *et al.*, 2024). Esse processo é realizado de forma iterativa, ajustando a curva até que o algoritmo responsável pelo cálculo das CP alcance a convergência. A Figura 2.9 ilustra a representação das CP.

Figura 2.9 – Ilustração de uma CP para um conjunto bidimensional de dados.



Fonte: Hastie & Stuetzle (1989).

A formulação matemática das CP baseia-se no conceito de auto-consistência, estabelecido pelo índice de projeção de um ponto $\mathbf{x}_i$ em uma curva $\mathbf{f}$ (Hastie; Stuetzle, 1989). Em um espaço $\mathbb{R}^d$ de d dimensões, uma curva unidimensional é representada por um vetor $\mathbf{f}(t)$, composto por d funções contínuas dependentes de uma única variável t, conforme expresso na Equação 2.26.

$$\mathbf{f}(t) = \{f_1(t), f_2(t), \ldots, f_d(t)\} \quad . \tag{2.26}$$

em que essas funções são chamadas de funções de coordenada, enquanto o parâmetro t define a sequência ao longo da curva.

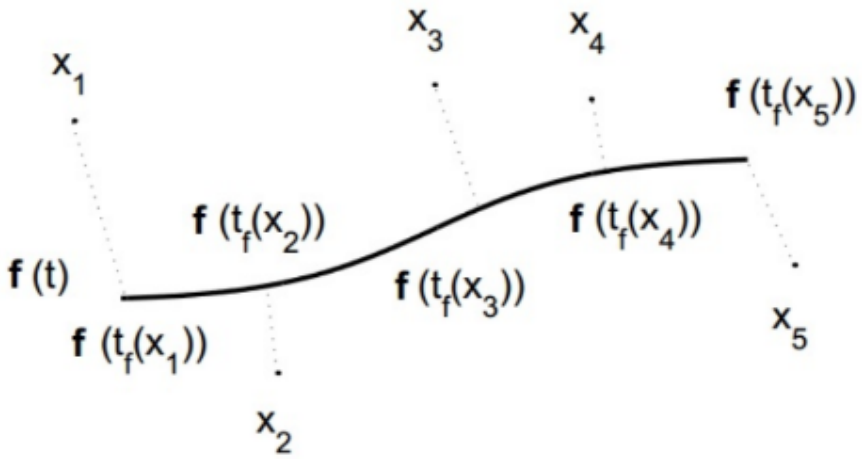
Considere x um vetor aleatório em $\mathbb{R}^d$, com densidade de probabilidade h e momento de segunda ordem finito. Seja f uma curva suave e sem autointerseções, ou seja, se $t_1 \neq t_2$ então $\mathbf{f}(t_1) \neq \mathbf{f}(t_2)$. A curva é parametrizada em um intervalo fechado $I \subseteq \mathbb{R}$. O índice de projeção $t_f : \mathbb{R}^d \to \mathbb{R}$ é representado pela Equação 2.27.

$$t_f(\mathbf{x}) = \sup\{t : \|\mathbf{x} - \mathbf{f}(t)\| = \inf_{\mu} \|\mathbf{x} - \mathbf{f}(\mu)\|\} \quad , \tag{2.27}$$

onde μ é uma variável auxiliar pertencente a $\mathbb{R}$.

O índice de projeção $t_f(\mathbf{x})$ corresponde ao valor de t para o qual a CP $\mathbf{f}(t)$ se encontra mais próxima do ponto $\mathbf{x}$. Caso existam múltiplos valores possíveis, é adotado o maior ou menor deles. A Figura 2.10 ilustra esse conceito, apresentando cinco pontos de referência $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_5$, cujas projeções na curva principal ocorrem nos pontos $\mathbf{f}(t_f(\mathbf{x}_1))$, $\mathbf{f}(t_f(\mathbf{x}_2))$, ..., $\mathbf{f}(t_f(\mathbf{x}_5))$, respectivamente. Observa-se que o parâmetro t, associado ao ponto da curva mais próximo do evento $\mathbf{x}_i$, é determinado por $t = t_f(\mathbf{x}_i)$, representando, assim, o índice de projeção desse ponto na curva.

Figura 2.10 – Mapeamento dos dados sobre a curva principal.



Fonte: Fernandez (2005).

Com base na definição do índice de projeção de um ponto em uma curva, é possível estabelecer as CP.

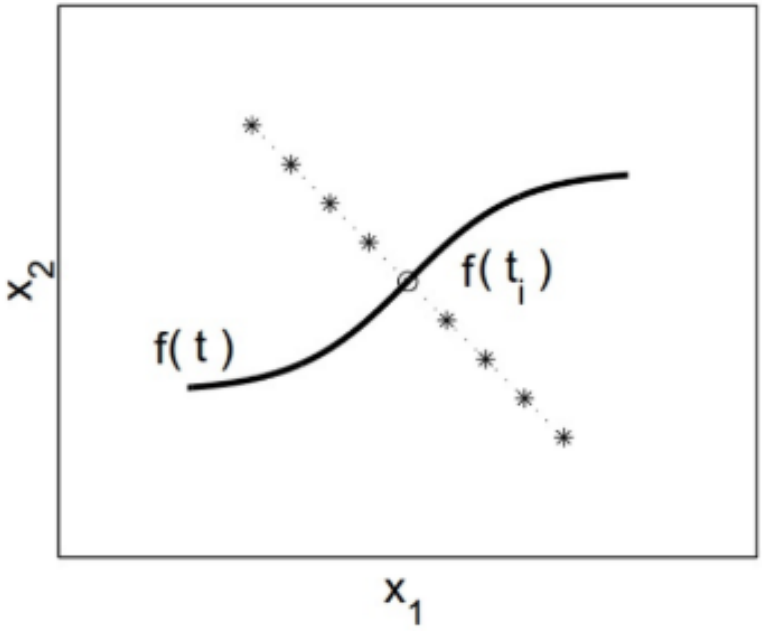
Definição 2.1.1. Uma curva é considerada auto-consistente, ou uma curva principal, se, para qualquer valor do parâmetro t, seus pontos corresponderem à média dos dados que

são projetados ortogonalmente sobre ela (Equação 2.28) (Hastie; Stuetzle, 1989; Beinert *et al.*, 2024).

$$\mathbf{f}(t) = \mathbb{E}\left[\mathbf{x} \mid t_f(\mathbf{x}) = t\right], \quad \forall t \quad . \tag{2.28}$$

A propriedade de auto-consistência resulta do fato de que os pontos que formam uma CP correspondem à média dos dados projetados sobre ela. Na Figura 2.11, que ilustra uma componente principal em um espaço bidimensional, observa-se oito eventos cuja projeção determina o ponto $\mathbf{f}(t_i)$ na curva principal. Assim, esse ponto $\mathbf{f}(t_i)$ representa a média dos oito eventos projetados sobre ele.

Figura 2.11 – Propriedade de auto-consistência de uma curva principal.



Fonte: Fernandez (2005).

De maneira geral, uma CP $\mathbf{f}(t)$, definida em $\mathbb{R}^d$, apresenta as seguintes propriedades:

- Trata-se de uma curva suave, isto é, diferenciável em todos os pontos e de ordem infinita.
- A curva não se sobrepõe em nenhum ponto, ou seja, para quaisquer $t_1 \neq t_2$ tem-se que $\mathbf{f}(t_1) \neq \mathbf{f}(t_2)$;
- A curva tem extensão finita dentro de uma região delimitada em $\mathbb{R}^d$;
- É auto-consistente.

2.9.1.2 Método de Curvas Principais K-segmentos

A confiabilidade desse algoritmo, aliada à sua garantia de convergência e à baixa sensibilidade a mínimos locais, faz com que ele se sobressaia em comparação com outros métodos (Verbeek; Vlassis; Kröse, 2002; Verbeek; Vlassis; Kröse, 2001). A obtenção das CP por meio deste algoritmo ocorre de forma incremental, começando com um único segmento e aumentando gradualmente a quantidade, otimizando a curva a cada nova adição.

O método K-segmentos apresenta uma complexidade computacional de $O(n^2)$, na qual n representa a quantidade de eventos no conjunto de dados empregado para a construção da curva (Kobel; Sagraloff, 2015). Seu processo pode ser dividido em três etapas, detalhadas a seguir:

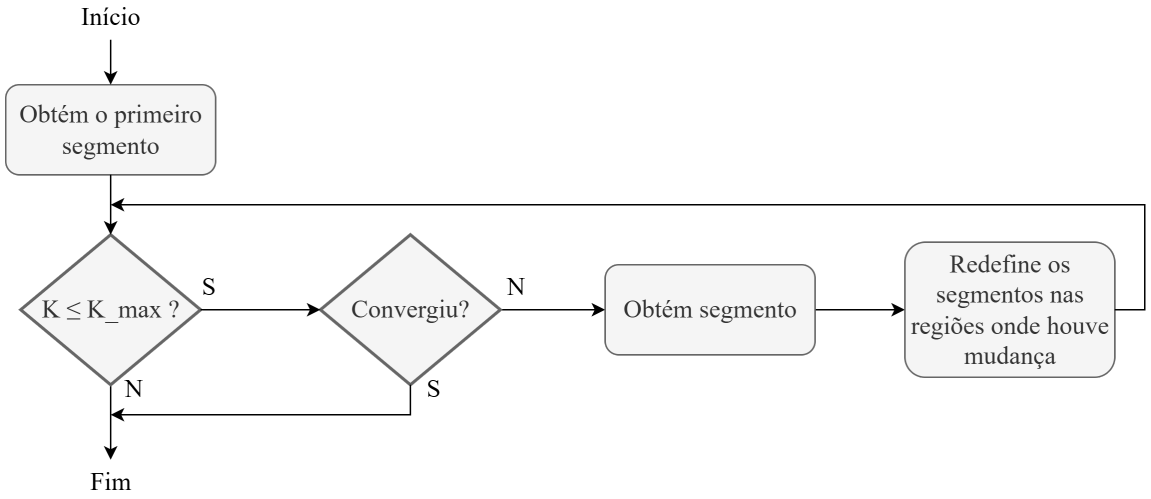
- Etapa 0 - Adição do segmento inicial: O primeiro segmento é inserido levando em conta todos os eventos do conjunto de dados. Inicialmente, é determinado um centro correspondente à média dos valores do conjunto. A partir desse ponto central, o primeiro segmento é traçado na direção do primeiro componente principal, com um comprimento equivalente a 3/2 do desvio padrão associado a esse componente.

- Etapa 1 - Adição de um novo segmento: Para a inclusão de cada segmento subsequente, o primeiro passo consiste em aplicar o algoritmo *K-Means* ao conjunto de dados, de modo a obter um novo ponto central (centroide). Esse ponto central passa a representar uma nova região candidata para a construção do segmento adicional. Na sequência, considerando a partição do espaço em regiões de Voronoi, são atribuídos a esse novo centroide os eventos que se encontram mais próximos dele do que de qualquer um dos segmentos já estabelecidos, constituindo o novo agrupamento. Como essa nova atribuição modifica a distribuição dos eventos entre os grupos, os segmentos previamente existentes são recalculados. Ao final, os segmentos resultantes são conectados por linhas retas, 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.

A Figura 2.12 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.

Figura 2.12 – Diagrama simplificado do funcionamento do método k-segmentos.



Fonte: Adaptado de Fernandez (2005).

Ao final da execução do algoritmo, é sugerido um número ideal de segmentos para a curva principal com base na variação do comprimento total da curva após a adição de um segmento. Como as regiões de Voronoi são selecionadas em ordem decrescente, os segmentos inseridos tornam-se progressivamente menores a cada iteração, até que a inclusão de um novo segmento não provoque uma alteração significativa na curva como um todo. Esse processo determina a quantidade final de segmentos recomendada pelo método. O algoritmo *K-Means*, empregado na construção das CP, permite a segmentação dos dados em regiões de Voronoi, embora não gere curvas que sigam estritamente o conceito de auto-consistência. No entanto, dois fatores garantem que a curva gerada pelo algoritmo de K-segmentos possa ser considerada uma CP:

- Os segmentos orientados na direção da primeira componente principal possuem a propriedade de auto-consistência.
- se a região corresponde a uma região de Voronoi, então os k pontos contidos nela são auto-consistentes.

2.9.1.3 Aplicações de Curvas Principais na Literatura

Na literatura, existem diversas aplicações do algoritmo de CP voltadas para o reconhecimento de padrões, como Borges *et al.* (2023), que buscou explorar a capacidade de representação de dados das CP para construir uma representação compacta de conjuntos de dados sintéticos e reais. Já Sousa *et al.* (2020) exploraram CP para avaliar dados experimentais obtidos de uma língua eletrônica impedimétrica utilizada para avaliar diferentes intensificadores de sabor de composições semelhantes. Além disso, Moraes *et al.* (2020) aplicaram um método de CP baseado em agrupamento em nove bases de dados com diferentes dimensionalidades e números de classes. Os resultados mostraram que o método é adequado para *clusters* com formas alongadas e esféricas e obteve resultados significativamente melhores em alguns conjuntos de dados do que outros algoritmos de agrupamento, como o *K-Means* e os algoritmos de mapas auto-organizáveis.

Já Borges *et al.* (2019) utilizaram um método de classificação baseado em CP para gerar uma fronteira de decisão na qual os dados dentro da fronteira são referentes ao motor de indução sem falha e os dados fora referem-se ao motor de indução com falha. Braga, Ferreira & Barbosa (2019) desenvolveram um classificador baseado em CP, K-segmentos, no qual realizaram testes experimentais em bases de dados sintéticas para demonstrar a eficiência do método proposto em problemas de representação. Moraes & Ferreira (2016b) propuseram um método baseado em CP para o agrupamento de dados, realizando a separação dos dados em grupos com características semelhantes. Ferreira *et al.* (2015) investigaram um índice de desvio da qualidade da energia baseado em CP. O índice proposto é utilizado para realizar uma abordagem direta na detecção de perturbações em sinais de energia.

Cortivo & Marques (2014) propuseram um classificador baseado em CP para a classificação supervisionada dos dados das bases de dados *Iris*, *Wine* e Tiroide, obtidos na UCI – *Machines Respository* (Asuncion; Newman *et al.*, 2007). Ferreira *et al.* (2013) exploraram como as CP podem ser aplicadas ao problema de monitoramento da qualidade da energia elétrica em termos de análise, detecção e classificação de eventos. Faier (2006) utiliza a técnica de CP na identificação de descargas parciais em sistemas de potência e Banfield & Raftery (1992) para a detecção de blocos de gelo em imagens de satélite. As CP também têm sido utilizadas para a extração de características para a identificação de escrita manual (Bai; Zhu, 2012) e no diagnóstico de doenças oculares (You *et al.*, 2011).

2.9.2 *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 (Jamain; Hand, 2005). 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 posterior $P(C_k|X)$ de que um exemplo X se encaixe na classe C_k. A equação geral é expressa pela Equação 2.29.

$$P(C_k \mid X) = \frac{P(C_k) \cdot P(X \mid C_k)}{P(X)} \quad , \tag{2.29}$$

em que $P(C_k)$ representa a probabilidade à priori da classe C_k. Já $P(X|C_k)$ 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).

2.9.3 *Random Forest* (RF)

Random Forest (RF), proposto por (Breiman, 2001), é um método de aprendizagem por *ensemble* que opera construindo k árvores de decisão a partir do conjunto de treinamento em k iterações. Em cada iteração, primeiro é selecionado aleatoriamente um conjunto de amostras do conjunto de treinamento. Para reproduzir uma árvore de decisão desse subconjunto, o RF escolhe aleatoriamente um subconjunto de atributos candidatos para cada nó. Desta forma, cada árvore de decisão é construída através do *ensemble*, empregando subconjuntos aleatórios independentes de características e amostras. A predição de uma nova classe de amostra é realizada da seguinte forma: cada classificador individual vota e a classe mais votada é eleita.

Um RF consiste em uma combinação de classificadores, onde cada classificador contribui com um único voto para a atribuição da classe mais frequente ao vetor de entrada $\hat{C}_{\text{rf}}^{B} =$ voto majoritário $\left\{\hat{C}_b(\mathbf{x})\right\}_1^B$, onde $\hat{C}_b(\mathbf{x})$ é a previsão de classe da b-ésima árvore. O fato de ser

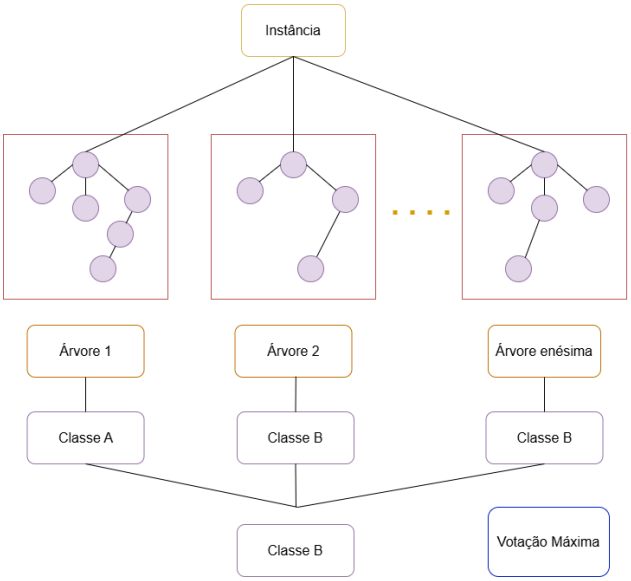
uma combinação de muitos classificadores confere ao RF algumas características que o tornam substancialmente diferentes de uma árvore de classificação tradicional e, portanto, deve ser entendido como um novo conceito de classificadores. Um RF aumenta a diversidade das árvores, fazendo-as crescer a partir de diferentes subconjuntos de dados de treinamento criados por meio de empacotamento ou agregação *bootstrap*. O *bagging* é uma técnica usada para treinar, criada por meio da reamostragem aleatória do conjunto de dados original com substituição (ou seja, sem exclusão dos dados selecionados da amostra de entrada para gerar o próximo subconjunto) (Rodriguez-Galiano *et al.*, 2012).

Um RF geralmente usa o Índice de Gini (Breiman, 2017) como medida para a melhor seleção dividida, que mede a impureza de um determinado elemento em relação ao resto das classes. Para um determinado conjunto de dados de treinamento T, o Índice de Gini pode ser expresso pela Equação 2.30.

$$\sum_{i} \sum_{\substack{j \\ j \neq i}} (f(C_i, T)/|T|)(f(C_j, T)/|T|) \quad , \tag{2.30}$$

em que $(f(C_j, T)/|T|)$ é a probabilidade de um caso selecionado pertencer à classe C_i. Assim, utilizando uma determinada combinação de características, faz-se com que uma árvore de decisão cresça até sua profundidade máxima (sem poda) (Breiman, 2001). A Figura 2.13 ilustra um exemplo do algoritmo RF, na qual a predição final é da classe que mais foi votada dentre das árvores.

Figura 2.13 – Exemplo de classificação do RF.



Fonte: Adaptado de Bhatnagar, Gill & Ghosh (2020).

2.9.4 *Extreme Learning Machine* (ELM)

O método *Extreme Learning Machine* (ELM) é uma rede neural artificial composta por uma única camada oculta (Huang; Zhu; Siew, 2004). O ELM visa equilibrar alta velocidade de treinamento com uma boa capacidade de generalização, evidenciando seu potencial. Quando comparado a redes neurais artificiais, SVM e outros métodos tradicionais de AM, o ELM se sobressai pela sua aprendizagem eficaz, habilidade de generalização e pela simplicidade e facilidade no processo de modelagem (Huang, 2015; Guo; Cheng; Wang, 2017).

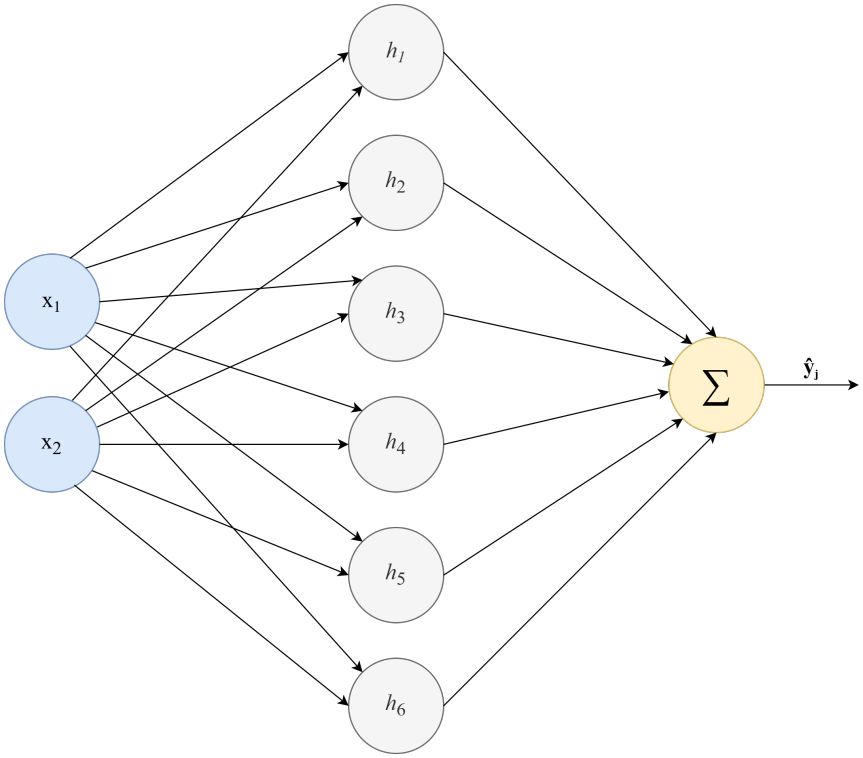
Ao contrário das redes neurais artificiais multicamadas, que é treinada de maneira iterativa utilizando o algoritmo de retropropagação do erro, o ELM possui uma estrutura mais simples e um treinamento bem mais rápido. No ELM, os pesos de entrada e os bias da camada oculta são gerados de maneira aleatória e permanecem inalterados, com apenas os pesos da camada de saída sendo ajustados, normalmente utilizando uma solução analítica. Essa propriedade torna o método mais atrativo para problemas que requerem uma alta eficiência computacional, pois diminui consideravelmente o custo computacional e o tempo de treinamento (Wang *et al.*, 2022).

No ELM, há três níveis principais de aleatoriedade: (1) os parâmetros relacionados à camada oculta são atribuídos de maneira aleatória; (2) a conexão entre entradas e neurônios ocultos não precisa ser completa, ou seja, certos nós de entrada podem não se conectar a um neurônio oculto específico; e (3) um neurônio oculto pode ser organizado como uma sub-rede composta por vários nós, o que facilita a aprendizagem de características locais (Huang *et al.*, 2015). A função de saída do ELM (Equação 2.31) é dada por:

$$\sum_{i=1}^{\tilde{N}} \beta_i f_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i f\left(a_i \cdot x_j + b_i\right) = y_j, \quad j = 1, \ldots, N \tag{2.31}$$

em que $\tilde{N}$ é a quantidade de neurônios da camada oculta, $f(x)$ é a função de ativação, $a_i = [a_{i1}, a_{i2}, \ldots, a_{in}]^T$ é o vetor de pesos que conecta o i-ésimo neurônio oculto aos nós de entrada, e b_i é o limiar do i-ésimo neurônio oculto. $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ é o vetor de pesos que conecta o i-ésimo neurônio oculto aos nós de saída; $a_i \cdot x_j$ representa o produto interno entre a_i e x_j; e a função de ativação geralmente adotada é do tipo "Tangente", "Relu", "Sigmóide", "Seno" ou "RBF" (Ding *et al.*, 2015). A Figura 2.14 ilustra um exemplo do método ELM com duas entradas, uma camada oculta com 6 neurônios e uma saída.

Figura 2.14 – Exemplo de uma arquitetura do método ELM.



Fonte: Do Autor (2026).

2.10 Inteligência Artificial Explicável

A Inteligência Artificial Explicável refere-se ao conjunto de métodos e ferramentas empregadas para tornar os resultados gerados por sistemas de IA mais compreensíveis e interpretáveis (Miller, 2019). A Inteligência Artificial Explicável se opõe aos modelos de "caixa-preta", pois tem como ideia esclarecer e proporcionar transparência às decisões produzidas pelo modelo. De modo geral, isso é feito ao identificar a contribuição dos fatores que têm maior impacto em uma determinada predição. Em muitos casos, esses elementos correspondem às variáveis de entrada do sistema, ou seja, às características (ou atributos) empregadas pelo modelo (Lou; Caruana; Gehrke, 2012).

Modelos classificados como "caixa-preta" são aqueles em que o funcionamento interno é considerado complexo. De modo geral, essas são abordagens nas quais até o engenheiro ou cientista de dados responsável pelo desenvolvimento tem dificuldade em entender e explicar com precisão todas as etapas que o modelo realiza até chegar a um resultado (Miller, 2019). Isso se deve ao fato de serem modelos desenvolvidos diretamente a partir dos dados. Exemplos

de modelos de caixa-preta são *Random Forest*, *Support Vector Machines*, *Naive Bayes*, Redes Neurais Artificiais, entre outros.

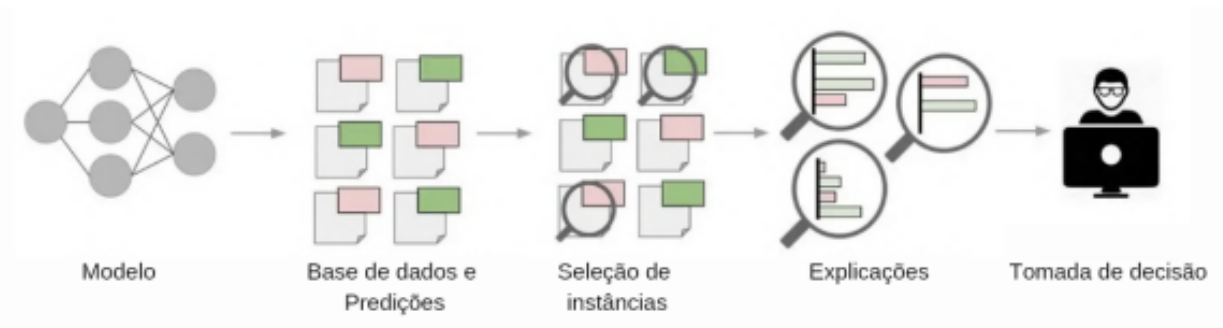
A Subseção 2.10.1 descreve o método *Shapley Additive Explanations* (SHAP), uma técnica de Inteligência Artificial Explicável adotada nesta pesquisa para interpretar e quantificar a contribuição de cada variável de entrada nas predições de modelos de AM.

2.10.1 *SHapley Additive exPlanations* (SHAP)

Lundberg & Lee (2017) apresentaram a sigla *Shapley Additive Explanations* (SHAP) para denominar um método amplamente utilizado na interpretação de modelos de AM. Seu objetivo é criar uma representação mais simples que sintetize o funcionamento de um modelo complexo. Para isso, utiliza-se a teoria dos jogos, um campo da matemática que analisa como atribuir, de maneira justa, a contribuição de cada participante, neste caso, cada variável, ao resultado final das previsões.

No SHAP, a previsão do modelo é entendida como um jogo colaborativo no qual cada variável desempenha o papel de um jogador. O valor de *Shapley* de cada variável é determinado com base em sua contribuição para o resultado, levando em conta todas as combinações possíveis com as outras variáveis. Essa formulação possibilita a captura de relações não lineares e interações características de modelos complexos. Dessa forma, consegue-se uma visão geral do papel de cada variável por meio da média de suas contribuições nas previsões e, simultaneamente, uma análise específica, ou seja, a influência particular de cada variável na previsão de uma observação específica (Rodríguez-Pérez; Bajorath, 2019). A Figura 2.15 ilustra o processo de análise dos resultados de um método utilizando explicações.

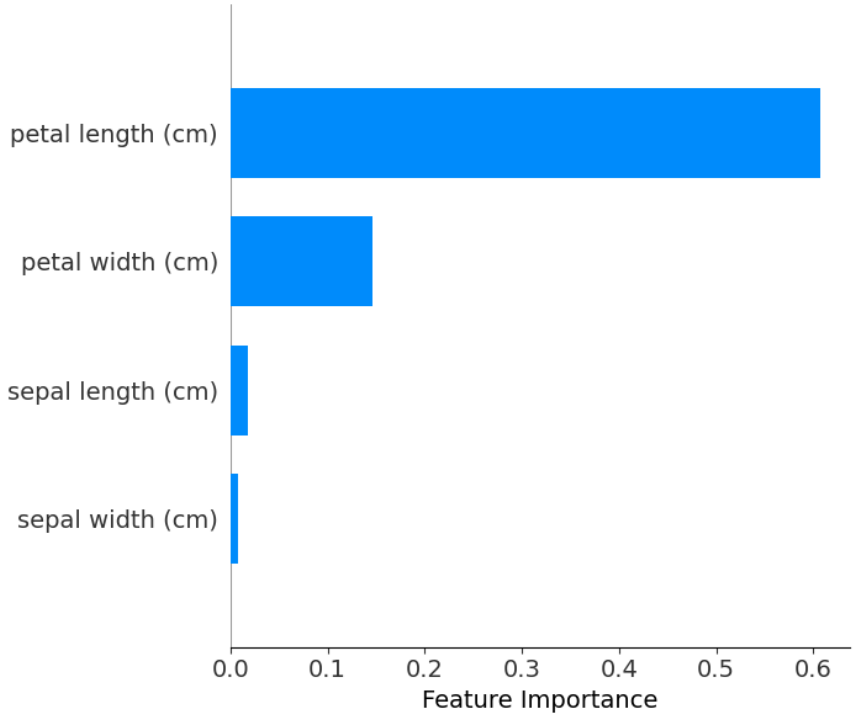
Figura 2.15 – Processo que começa no método e chega às decisões humanas, apoiadas nas explicações geradas.



Fonte: Amoroso (2023).

Para facilitar um melhor entendimento do comportamento de um método, emprega-se a visualização por meio do gráfico de importância dos valores SHAP, que evidencia a contribuição de cada variável nas previsões. O gráfico de importância de atributos (*feature importance*) fornece uma visão geral de quanto cada atributo contribui para o desempenho do método. Ele agrega os valores SHAP calculados em várias observações e gera uma medida de importância para cada variável. Essa métrica representa a média do valor absoluto dos valores SHAP, que indica o impacto médio absoluto da variável nas previsões, independentemente de sua influência ter aumentado ou diminuído o valor previsto. A Figura 2.16 ilustra um exemplo de gráfico de importância obtido por um método na predição do tipo de espécies de flores do gênero *Iris*, com base em medidas morfológicas coletadas de suas pétalas e sépalas. Pode-se observar que os atributos com os maiores valores no gráfico são os que mais impactam no método.

Figura 2.16 – Exemplo de gráfico de importância.



Fonte: Do Autor (2026).

2.11 Métricas de Desempenho

Nesta seção, são apresentadas as métricas utilizadas para avaliar e comparar o desempenho dos modelos que foram utilizados nesta pesquisa.

2.11.1 Definições

- Verdadeiros Positivos (VP): amostras corretamente classificadas como pertencentes a uma determinada classe.
- Falsos Positivos (FP): número de casos incorretamente classificados como pertencentes à classe.
- Verdadeiros Negativos (VN): número de casos corretamente classificados como não pertencentes a uma classe.
- Falsos Negativos (FN): número de casos incorretamente classificados como não pertencentes à classe avaliada.

2.11.2 Acurácia

A acurácia (ACC) calcula a proporção das previsões corretas em relação ao total de instâncias (Equação 2.32) (Sammut; Webb, 2011).

$$\mathrm{ACC} = \frac{\mathrm{VP} + \mathrm{TN}}{\mathrm{VP} + \mathrm{VN} + \mathrm{FP} + \mathrm{FN}} \quad . \tag{2.32}$$

2.11.3 Precisão

Precisão (PR) é uma métrica que mede o número de VP entre as amostras classificadas como pertencentes a determinada classe (Equação 2.33) (Sammut; Webb, 2011).

$$\mathrm{PR} = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FP}} \quad . \tag{2.33}$$

2.11.4 *Recall*

Também denominado Sensibilidade ou Taxa de Verdadeiros Positivos (TVP), *Recall* (RE) é a medida de VP, isto é, o número de amostras classificadas como pertencentes a deter-

minada classe, entre todas as amostras originalmente pertencentes a essa classe (Equação 2.34) (Sammut; Webb, 2011).

$$\mathrm{RE} = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}} \quad . \tag{2.34}$$

2.11.5 F1-Score

O *F1-score* (Equação 2.35) é uma média harmônica ponderada da PR e RE. Essa métrica é utilizada quando se busca um equilíbrio entre PR e RE, e também quando o conjunto de dados é desbalanceado, podendo apresentar um alto número de amostras negativas (VN) ao avaliar cada classe separadamente. Um bom valor de *F1-Score* indica um baixo número de (FP) e (FN) (Sammut; Webb, 2011).

$$F1 = 2 \cdot \frac{\mathrm{PR} \times \mathrm{RE}}{\mathrm{PR} + \mathrm{RE}} \quad . \tag{2.35}$$

2.11.6 Coeficiente de Kappa

O coeficiente *Kappa* mede o nível de concordância entre dois especialistas, corrigindo a concordância esperada pelo acaso. Um ($k = 1$) indica uma concordância perfeita, para o ($k = 0$) representa concordância similar ao acaso e ($k < 1$) indica uma discordância (Equação 2.36) (Sammut; Webb, 2011).

$$\kappa = \frac{P_o - P_e}{1 - P_e} \quad , \tag{2.36}$$

onde:

- P_o é a proporção de concordâncias observadas.
- P_e é a proporção de concordâncias esperadas ao acaso.

2.11.7 Área Sob a Curva ROC

A Área Sob a Curva (*Area Under the Curve* (AUC)) Característica de Operação do Receptor (*Receiver Operating Characteristic* (ROC)) é uma métrica que avalia a capacidade do

modelo de diferenciar entre as classes positiva e negativa em diversos limiares de decisão. A curva ROC é elaborada com base na relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos. O valor da AUC varia de 0 a 1, sendo que, quanto mais próximo de 1, melhor é o desempenho do modelo, enquanto valores em torno de 0,5 indicam um desempenho equivalente ao acaso. A Equação 2.37 ilustra como é calculado o valor da AUC.

$$\text{AUC} = \sum_{i=1}^{n-1} (\text{TFP}_{i+1} - \text{TFP}_i) \frac{\text{RE}_{i+1} + \text{RE}_i}{2} \quad , \tag{2.37}$$

em que RE é a métrica *Recall* e TFP é a taxa de falsos positivos, que é expressa pela Equação 2.38.

$$\text{TFP} = \frac{\text{FP}}{\text{FP} + \text{VN}} \quad . \tag{2.38}$$

2.12 Trabalhos Relacionados com Estudos a respeito do ENEM

Vários pesquisadores têm estudado como é o desempenho dos estudantes no ENEM e quais são os fatores que contribuem para esse desempenho. Nesses trabalhos, são feitas análises e propostas de ferramentas no contexto de aprendizado de máquina para entender o problema e buscar soluções para obter os resultados.

Filho (2017) apresentou uma solução baseada em mineração de dados com o intuito de prever e estimar o desempenho dos estudantes do Ensino Médio dos institutos federais. Foram integradas as bases de dados do ENEM e as do Censo Escolar de 2014 para a formação de um *data-mart* para auxiliar nas análises. Foram utilizados os algoritmos Árvores de Decisão (AD), Regras de Classificação e Regressão Logística (RL). O algoritmo RL foi o que obteve os melhores resultados, com 0,84 e 0,51 para as métricas *Area under curve ROC* (AUC_ROC) e *K-squared Maximum* (KS2_MAX), respectivamente.

Souza *et al.* (2019) aplicou a descoberta de conhecimento em bases de dados para que pudesse classificar as escolas por nota, selecionar as seis melhores, e obter as características do desempenho dos estudantes do Ensino Médio do Distrito Federal, na qual a fonte de pesquisa foram os microdados do ENEM 2017 com associação a fatores socioeconômicos e da infraestrutura escolar. Foram utilizados os algoritmos AD, NB e *Lazy Kstar* (LK). O algoritmo de AD teve os melhores resultados, com 61,09% de instâncias classificadas corretamente. As análises

destacam que as escolas públicas com as melhores notas são de regiões administrativas com Índice de Desenvolvimento Humano (IDH) mais alto, destacando um desequilíbrio de condições de ensino-aprendizagem no Distrito Federal.

Silva *et al.* (2020) realizou um estudo para identificar o desempenho e as desigualdades sociais na análise dos estudantes de Minas Gerais que fizeram o ENEM em 2019. O algoritmo de agrupamento *K-Means* e o algoritmo Apriori, que têm como foco as Regras de Associação, foram utilizados para identificar o nível de afinidade entre os elementos da base de dados. Foram geradas oito regras visando as variáveis socioeconômicas para caracterizar os grupos. Os resultados indicam que o número ideal de grupos para o algoritmo *K-Means* é igual a 2 (Grupo A e Grupo B), com um valor de *Silhueta* de aproximadamente 0,45. Já em relação as Regras de Associação, uma delas indicam que, com 99,9% de confiança, os alunos com médias entre 585,6 e 718,21 no ENEM estão no Grupo B. Além disso, uma outra regra destaca que, considerando informações originadas do questionário socioeconômico, entre os alunos do Grupo A, 94,5% vem de escolas estaduais, e dentre os estudantes de escolas estaduais que se autodeclararam negros, pardos, amarelos ou indígenas, 70,1% estão no Grupo A.

No estudo de Franco *et al.* (2020), foram utilizados algoritmos para selecionar os melhores atributos da base de dados do ENEM de 1998 a 2019 e algoritmos para classificar se o desempenho do estudante seria alto ou baixo, de acordo com os 20 atributos selecionados, como Língua Estrangeira, se possui um microcomputador na residência, tipo de escola que cursou ou está cursando o ensino médio, renda mensal familiar, entre outros. Algumas das etapas consistiram em calcular as médias aritméticas de todas as notas para cada prova. O limite de 600 pontos determina se o aluno tem desempenho baixo ou alto, sendo considerado alto quando o aluno alcança a média cujo valor é igual ou superior a 600 pontos e baixo se o valor for inferior ao mencionado limite. O algoritmo com melhor desempenho foi o *eXtreme Gradient Boosting* (XGBoost), com 89% de acurácia, e o melhor seletor de características foi o *Sequential Feature Selector* (SFS) e o PCA.

No trabalho de Banni, Oliveira & Bernardini (2021), foram utilizados os dados do ENEM 2018. Além disso, foi utilizada a Mineração de Dados Educacionais com métodos de visualização de dados. Para identificar as características mais relacionadas ao desempenho dos alunos, foram utilizados modelos de AM para previsão. Nos resultados, os algoritmos que obtiveram os melhores resultados foram a RL e AD, com precisão de aproximadamente 80%.

Franco (2021) buscou identificar as características mais relevantes no desempenho dos estudantes no ENEM de 1998 a 2019. Foram utilizados os algoritmos *Select K-Best*, RFE, *GenericUnivariateFeatures* e PCA, da biblioteca *Scikit-learn*[1]. O algoritmo *Sequential Feature Selector* combinado com o classificador de Regressão Linear do *Scikit-learn* e os algoritmos *InfoGainAttributeEval*, *SymmetricalUncertAttributeSetEval* e *GainRatioAttributeEval* do *Weka* também foram utilizados. Os métodos para classificação utilizados foram o *XGBoost*, *LinearSVC* e *ExtraTreesClassifier*. Os melhores resultados para classificação foram com as técnicas de seleção de atributos SFS e PCA, e o método XGBoost, com taxas de acerto superiores a 76%.

Garcia, Rios-Neto & Miranda-Ribeiro (2021) realizou a predição do desempenho dos estudantes que realizaram o ENEM 2016, 2017 e 2018. Os algoritmos de Regressão Logística Multinomial (RLM) e Regressão Linear *Stepwise* (RLS) foram utilizados. A RLM teve um R^2 ajustado de 0,339. Já a RLS teve um R^2 ajustado de 0,336. As análises realizadas destacam que os estudantes de escolas da rede estadual estão em desvantagem e que os de escolas federais e privadas possuem o mesmo desempenho, quando considerado apenas o tipo de escola. Todavia, a qualificação docente se mostrou o fator mais impactante no desempenho escolar. Além disso, os estudantes do ensino regular das escolas federais apresentam o melhor desempenho, cerca de 1,3 vezes maior do que os estudantes das escolas estaduais.

Gomes *et al.* (2021) realizou uma análise preditiva do desempenho dos estudantes que realizaram o ENEM 2011, em relação à área de matemática. Foi utilizado o algoritmo de árvore *Classification and Regression Tree* (CART). O CART conseguiu explicar 29,97% da variância do desempenho em matemática. Por meio das análises, foi observado que as características que influenciam no pior desempenho em matemática são renda familiar de até dois salários mínimos, o sexo feminino, não ter cursado escolas particulares no ensino fundamental e no ensino médio, e residir nas regiões Norte, Nordeste e Centro-Oeste.

O trabalho de Melo *et al.* (2021) teve como objetivo identificar as variáveis com maior influência no desempenho na prova objetiva e na redação, em relação aos municípios no ENEM 2018. Os algoritmos RLM e Regressão Espacial foram utilizados para prever o desempenho. A RLM teve um R^2 ajustado de 0,68. Foi percebido que as variáveis econômicas selecionadas têm um impacto no desempenho dos municípios. Agora, com a Regressão Espacial por meio do método *Spatial Two-Stage Least Squares* (STLS), obteve-se um R^2 ajustado de 0,70. Foi

[1] https://scikit-learn.org/stable/

percebido que variáveis como o percentual de estudantes com bolsa, renda, raça, escolaridade e nível instrucional da mãe são fatores relevantes para o desempenho e a dispersão das notas dos estudantes de cada município.

Alves (2022) buscou desenvolver um método linear para predizer o desempenho dos estudantes na prova da área de Ciências Humanas do Enem 2019. Também foi aplicada estatística descritiva, analisando a diferença de desempenho entre os grupos de características dos examinados via ANOVA, assim como foi investigada a curva característica de itens. Para predizer os resultados, foram consideradas as características socioeconômicas dos estudantes. Os resultados obtidos destacam que características como renda familiar e tipo de escola do Ensino Médio (pública ou privada) influenciam o desempenho no exame.

Noguera & Aguiar (2023) propôs criar uma arquitetura baseada em *Data Warehousing*, mineração de dados, estatística inferencial, processamento paralelo e distribuído para a análise de dados do ENEM de 2016 a 2020. A arquitetura possui cinco camadas: conexão de dados, gerenciamento de dados, análise de dados, apresentação de dados e gerenciador do fluxo de trabalho, automatizando as tarefas complexas. Foram utilizados os algoritmos AD, SVM e *Multilayer Perceptron Networks* (MLP) para predição do desempenho do estudante em cada área do conhecimento do ENEM. O SVM e MLP tiveram os melhores resultados, com acurácia acima de 79% para os dois algoritmos. Além disso, os resultados destacaram que, para as áreas de MT, os estudantes do sexo masculino tiveram um desempenho superior ao sexo feminino. Já para a área de LC, as mulheres tiveram um desempenho superior.

Máximo & Ribeiro (2023) tiveram como objetivo, por meio da base de dados do ENEM 2019, analisar quais características relacionadas ao estudante e à escola são explicativas para a média da nota no exame. Foi utilizado um algoritmo de modelagem linear generalizada multinível, no qual os dados referentes aos inscritos são de acordo com os dados referentes às escolas. O algoritmo foi construído com 36 variáveis, chegando a um valor de máximo de verossimilhança restrita de 65.144,52. Além disso, foi observado que os fatores socioeconômicos dos estudantes têm uma forte relação com dados como a raça/cor, renda, sexo e escolaridade dos pais dos alunos, além de dispositivos de acesso à informação nas escolas.

Na pesquisa de Júnior (2023), foi conduzida uma análise de dados estruturados do ENEM com o objetivo de contribuir para o aprimoramento de políticas educacionais públicas no Brasil. O estudo utilizou ferramentas de análise de dados, como estatística descritiva, mineração de dados e aprendizado de máquina, para explorar os microdados disponibilizados

pelo INEP. Entre as técnicas empregadas, destacam-se modelos de regressão, utilizados para avaliar a influência de variáveis socioeconômicas e escolares no desempenho dos estudantes, e métodos de clusterização, como o *K-Means*, para identificar perfis de participantes com características e necessidades específicas. Os resultados mostraram que fatores como acesso à educação básica de qualidade, regularidade no estudo e maior suporte escolar estão diretamente associados a melhores desempenhos no ENEM. A pesquisa destaca a importância de políticas públicas para reduzir desigualdades educacionais e promover o acesso a oportunidades de aprendizado, especialmente em regiões com maior vulnerabilidade social.

No trabalho de Macedo & Saporetti (2023) foi utilizada a técnica de aprendizado de máquina para analisar os dados do ENEM 2019 e 2020, verificando possíveis desigualdades sociais entre os estudantes destes anos e foi realizada a predição do desempenho dos estudantes no exame. Foi utilizado os algoritmos MLP, *K-Nearest Neighbors* (KNN) e RF para predição do desempenho dos estudantes, sendo considerado classe 1 como aluno com nota igual ou acima da média e 0 aluno abaixo da média. Já o *K-Means* foi utilizado como técnica de agrupamento para verificar as desigualdades sociais. Na predição do desempenho dos estudantes, o algoritmo MLP combinado com *Select K-Best* para seleção das melhores características foi o que obteve os melhores resultados, com uma precisão de 85,18% para o ano de 2019 e 83,63% para 2020. Com a aplicação do *K-Means*, foram obtidos 2 grupos no agrupamento, um por estudantes com menores condições financeiras e outro por estudantes de melhores condições financeiras.

No estudo de NETO *et al.* (2023), foi realizada uma análise dos impactos da pandemia da Covid-19 no ENEM. Foi utilizado agrupamento hierárquico para agrupar estudantes com base em características semelhantes, como região em que vivem, tipo de escola (pública ou privada) e acesso a recursos educacionais. O trabalho analisou os indicadores, como a participação dos estudantes no exame, desempenho médio nas provas e desigualdades socioeconômicas entre os estudantes. Os resultados mostraram uma queda na taxa de inscrição e o aumento de estudantes ausentes no exame, especialmente entre estudantes de escolas públicas e de baixa renda, evidenciando os desafios enfrentados por esses grupos no acesso ao ensino remoto e preparação para a prova. Ademais, houve uma queda no desempenho médio, em disciplinas como CN e MT. Assim, foi evidenciado no trabalho que a pandemia agravou desigualdades educacionais já existentes, reforçando a necessidade de políticas públicas para promover maior igualdade no acesso à educação e na preparação para o exame.

Neiva (2023) utilizou a metodologia SHAP para avaliar a interpretação do algoritmo de classificação *lightgbm classifier* na base de dados do ENEM 2021. O algoritmo obteve uma taxa de acerto de aproximadamente 71%. A variável que mais teve impacto no algoritmo foi computador na casa, devido à pandemia, na qual o ensino foi remoto. Outra variável que teve impacto foi a renda familiar mensal. No estudo, é constatado que as variáveis relacionadas ao perfil dos estudantes que influenciam eles terem maior probabilidade de obter nota mínima no exame são alta renda familiar, pais com ocupação mais valorizada, brancos e homens. Já as relacionadas ao perfil dos estudantes que influenciam na probabilidade de ter um baixo desempenho são ausência de computador na casa, renda familiar baixa e mulheres.

Por fim, no trabalho de Dutra (2024) foram analisados os perfis de estudantes participantes do ENEM durante o período da pandemia da Covid-19, considerando os hábitos de estudo e sua relação com o desempenho no exame. Para o trabalho, foram utilizados o algoritmo de agrupamento *K-Means* para agrupar os estudantes de acordo com semelhanças em seus comportamentos e características socioeconômicas, e o algoritmo de regras de associação *Apriori*. Os resultados destacam que os estudantes que adotavam uma rotina de estudos bem organizada, com uma maior dedicação semanal e diversificação nos métodos de aprendizado, apresentaram melhor desempenho no exame. Além disso, o tipo de escola, sendo pública ou privada, e fatores socioeconômicos, como a renda familiar, impactaram no resultado.

Diante dos estudos relacionados ao ENEM, nota-se que a maioria das abordagens foca na utilização de classificadores tradicionais ou em análises estatísticas do perfil socioeconômico, geralmente empregando estratégias de pré-processamento e seleção de atributos específicas para cada pesquisa. Diferentemente dessas abordagens, este estudo contribui ao explorar e empregar o método de CP K-segmentos como uma alternativa para classificar o desempenho acadêmico dos estudantes. Além disso, incorpora a otimização automática de hiperparâmetros utilizando a meta-heurística GWO.

3 METODOLOGIA

Neste capítulo, é apresentada uma descrição dos dados do ENEM 2023, considerando os atributos disponíveis na base de dados. Além disso, são detalhadas as ferramentas computacionais empregadas, a etapa de preparação dos dados, que envolve a seleção dos atributos mais relevantes, o pré-processamento e a transformação dos dados para adequação aos métodos utilizados. Também é apresentado o diagrama de trabalho adotado para o desenvolvimento desta pesquisa.

A metodologia desta pesquisa foi estruturada para avaliar a abordagem proposta em diferentes cenários. Inicialmente, são conduzidos experimentos em bases de dados simuladas e amplamente utilizadas na literatura, escolhidas por possuírem características variadas, como diferentes números de atributos, números de classes, diferentes geometrias e graus de separabilidade. Essa etapa tem como finalidade analisar e validar a abordagem proposta. Em seguida, a mesma abordagem é utilizada em um problema real, empregando a base de dados do ENEM 2023.

3.1 Ferramentas Computacionais

Esta seção apresenta as ferramentas computacionais abordadas nesta pesquisa, que são a linguagem de programação *Python*, as bibliotecas *Pandas*, *Matplotlib*, *Scikit-learn*, *ocpc_py* e *shap*, além do ambiente *Jupyter Notebook* para o desenvolvimento do código.

Python é uma linguagem de programação de alto nível, interpretada e de código aberto, criada por Guido van Rossum em 1989 (McKinney, 2018). Desde então, tornou-se amplamente utilizada em distintas áreas, como desenvolvimento web, ciência e análise de dados, inteligência artificial e extração de informações. Graças à sua tipagem dinâmica, oferece grande versatilidade ao permitir a integração com componentes escritos em outras linguagens. Além disso, sua ampla variedade de bibliotecas e *frameworks* simplifica a realização de tarefas específicas, auxiliando os desenvolvedores a otimizar processos e aumentar sua eficiência (Borges, 2014).

O *Pandas* é uma das bibliotecas que será empregada nesta pesquisa, sendo um pacote de código aberto desenvolvido para facilitar a análise de dados em *Python* (Chen, 2018). Criada por *Wes McKinney* em 2008, essa ferramenta disponibiliza estruturas de dados robustas e versáteis, projetadas para simplificar e otimizar a manipulação e a exploração de informações. Entre suas

funcionalidades estão operações rápidas de filtragem, agrupamento e agregação, bem como o tratamento eficiente de dados tabulares. A biblioteca também se destaca por gerenciar valores ausentes e duplicados com praticidade, além de suportar a leitura e gravação de arquivos em vários formatos, incluindo CSV, *Excel Open XML Spreadsheet* (XLSX), *Excel Spreadsheet* (XLS) e *JavaScript Object Notation* (JSON) (Team, 2020).

Matplotlib (Tosi, 2009) é um pacote para visualização de dados e gráficos. Ele conta com uma ampla gama de recursos, interações e personalizações, facilitando a criação de gráficos, como gráficos de barras, linhas e de áreas, assim como diagramas de dispersão, histogramas, entre outros. Ademais, a biblioteca é compatível com outras bibliotecas do *Python*, como *Pandas* e *Numpy*. Por conseguinte, o *Matplotlib* possibilita trabalhar com análise de dados, permitindo entender o comportamento dos dados, descobrir padrões e também identificar inconsistências nos dados (Chen, 2018).

O *Scikit-learn* é uma biblioteca de código aberto que permite trabalhar com AM, ela conta com um conjunto de recursos, como algoritmos para realizar análises de dados, métricas para predição, entre outros. Através do *Scikit-learn*, é possível criar diversos modelos de aprendizagem de máquina, como modelos para agrupamento, regressão, classificação, otimização de modelos, seleção de características, visualização dos resultados e métricas para avaliação dos modelos de aprendizagem, entre outros. Outrossim, também é possível integrar o *Scikit-learn* com outras bibliotecas, como *Pandas*, *Matplotlib* e *Numpy*, entre outras, para análise de dados (Pedregosa *et al.*, 2011). Já o *ocpc_py* (Borges, 2023) é uma biblioteca que permite trabalhar, em *Python*, com o algoritmo de CP K-segmentos, disponibilizando, no pacote, o classificador denominado Curvas Principais *Classifier* (CPC). Além disso, o SHAP é uma biblioteca implementada em *Python* que incorpora métodos matemáticos para interpretar localmente as saídas dos modelos de AM.

Já o *Jupyter Notebook* é uma ferramenta *web* de código aberto que permite criar, compartilhar e executar códigos, bem como fazer comentários sobre os códigos e visualizar o código em tempo real. Ele permite executar diversas linguagens de programação, como *Python*, *R*, *Fortran* e *C++*, entre outras. Na utilização do *Jupyter Notebook*, os usuários podem criar células, executar seus códigos em tempo real e inserir imagens explicativas. Também é possível adicionar novas personalizações no *Jupyter*, como *plugins*, para ampliar a gama de recursos da ferramenta. Além disso, é possível compartilhar os documentos que contêm os códigos para que outras pessoas possam contribuir com eles. O *Jupyter Notebook* é útil em áreas como ci-

ência de dados, aprendizado de máquina e desenvolvimento de *software* (PERKEL, 2018). Os experimentos foram executados em um computador com as seguintes especificações: *Intel(R) Core (TM)* i7-14700F, 32 GB de RAM e sistema operacional *Windows* 11.

3.2 Descrição das Bases de Dados da Literatura

As bases de dados da literatura utilizadas foram *Breast Cancer*, *Iris*, *Thyroid* e *Wine*, obtidas através de (Vinay, 2025) e do pacote *scikit-learn* (Pedregosa *et al.*, 2011) do *Python*. A base de dados *Breast Cancer* refere-se a um problema de classificação binária que tem como objetivo principal auxiliar na classificação de tumores de mama como benignos ou malignos, a partir de características obtidas por exames de imagem microscópica de tecidos. A base possui 569 amostras, 30 atributos (incluindo aspectos como textura, suavidade, simetria, concavidade, raio e perímetro das massas celulares) e 2 classes binárias (0 para tumor maligno e 1 para tumor benigno). Já a base de dados *Iris* tem como propósito distinguir espécies de flores do gênero Iris com base em medidas morfológicas coletadas de suas pétalas e sépalas. Ela contém 150 amostras, 4 atributos e 3 classes, divididas igualmente entre três espécies de flores (*Iris* setosa, *Iris* versicolor e *Iris* virginica). Cada amostra é descrita por quatro atributos numéricos contínuos, os quais são o comprimento e a largura das sépalas e das pétalas.

Além disso, a base de dados *Thyroid* contém dados sobre a recorrência do câncer de tireoide após a terapia com iodo radioativo (RAI). Ao todo, a base possui 383 registros de pacientes com 12 atributos, incluindo idade, sexo, estadiamento do câncer, tipo de patologia, classificação de risco e resposta ao tratamento. A variável alvo é o estado de recorrência do câncer, podendo ser "*yes*" ou "*no*", indicando se o câncer recorreu ou não. Por fim, a base de dados *Wine* tem o objetivo de classificar amostras de vinho de acordo com sua origem, a partir de medições físico-químicas realizadas em laboratório. A base possui 178 amostras de vinhos provenientes de três cultivares de uvas diferentes cultivadas na região italiana de Piemonte, cada uma representando uma classe distinta (classe 0, 1 e 2). As amostras são descritas por 13 variáveis numéricas contínuas, que correspondem às características químicas e espectrais dos vinhos, tais como teor alcoólico, concentração de ácido málico, cinzas, magnésio, fenóis totais, flavonoides, intensidade de cor e prolina, entre outros.

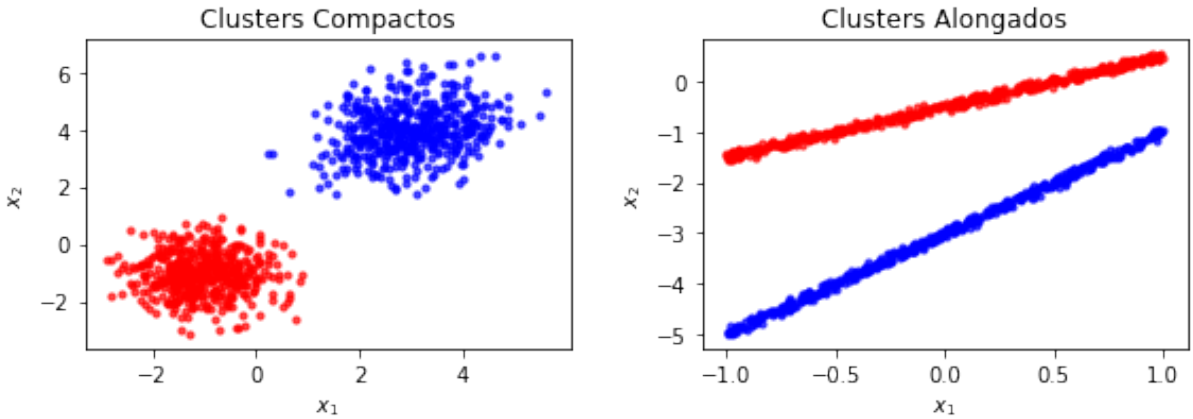
3.3 Descrição das Bases de Dados Sintéticas

A Tabela 3.1 apresenta uma descrição detalhada das bases de dados sintéticas consideradas nesta pesquisa, contemplando suas principais características estruturais, tais como o número de amostras, a quantidade de classes e o total de variáveis existentes em cada conjunto de dados. As Figuras 3.1 e 3.2 ilustram graficamente os conjuntos de dados sintéticos utilizados: Compactos, Alongados, Esféricos e Espirais.

Tabela 3.1 – Descrição das Bases de Dados Sintéticas.

Base de Dados	**Amostras**	**Classes**	**Variáveis**
Compacta	1000	2	2
Alongada	1000	2	2
Esférica	1000	2	2
Espiral	1000	2	2

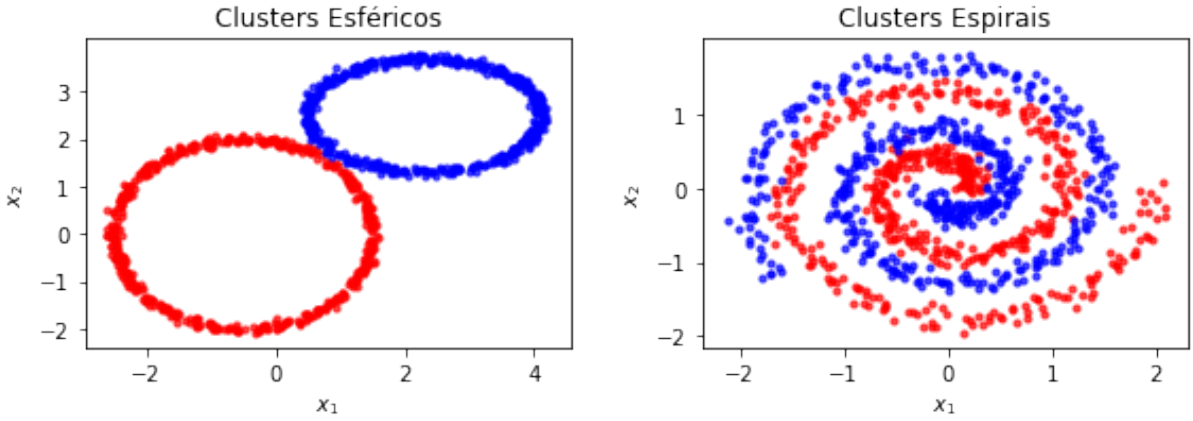
Figura 3.1 – Representação dos *clusters* compactos e alongados.



(a) Gráfico *cluster* compacto. (b) Gráfico *cluster* alongado.

Fonte: Do Autor (2026).

Figura 3.2 – Representação dos *clusters* esféricos e espirais.



(a) Gráfico *cluster* esférico. (b) Gráfico *cluster* espiral.

Fonte: Do Autor (2026).

3.4 Descrição dos Dados do ENEM

Durante o processo de inscrição no exame, o estudante preenche um questionário contendo 25 perguntas relacionadas ao seu perfil socioeconômico, contexto familiar, trajetória educacional e situação profissional. Também são coletadas informações pessoais, como idade, sexo, ano de conclusão do ensino médio e tipo de escola frequentada (pública ou privada), entre outras variáveis que auxiliam na produção de conhecimento.

Os microdados do ENEM fazem parte do conjunto de dados abertos disponibilizados pelo INEP e podem ser acessados gratuitamente em seu site[1]. Essa base de dados é organizada de maneira que cada linha representa um participante inscrito no exame, enquanto as colunas descrevem suas características individuais. Ao todo, a base de dados do ENEM reúne 76 atributos distintos. Para esta pesquisa, foi utilizada a base de dados do ENEM 2023, por se tratar da versão mais recente disponível no período de desenvolvimento do estudo, com todos os ajustes no dicionário de dados e na base de itens atualizados pelo INEP. A base do ENEM 2023 utilizada possui 3.933.955 amostras.

As Tabelas 3.2 e 3.3 apresentam as principais variáveis da base de dados do ENEM utilizadas neste estudo, baseadas nos estudos de Dutra (2024), Neiva (2023), NETO *et al.* (2023) e Máximo & Ribeiro (2023). As demais variáveis estão detalhadas no Apêndice A.

[1] https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/enem

Tabela 3.2 – Descrição das variáveis presentes na base de dados do ENEM.

Variável	Descrição	Tipo da Variável
NU_ANO	Ano do ENEM	Numérica
TP_FAIXA_ETARIA	Faixa etária de idades dos estudantes	Numérica
TP_SEXO	Sexo	Categórica
TP_ESTADO_CIVIL	Estado Civil	Numérica
TP_COR_RACA	Cor/raça	Numérica
TP_NACIONALIDADE	Nacionalidade	Numérica
TP_ST_CONCLUSAO	Situação de conclusão do Ensino Médio	Numérica
TP_DEPENDENCIA_ADM_ESC	Dependência administrativa (Escola)	Numérica
TP_LOCALIZACAO_ESC	Localização (Escola)	Numérica
TP_ANO_CONCLUIU	Ano de Conclusão do Ensino Médio	Numérica
TP_ESCOLA	Tipo de escola do Ensino Médio	Numérica
TP_ENSINO	Tipo de instituição que concluiu ou concluirá o Ensino Médio	Númerica
TP_LINGUA	Língua Estrangeira	Numérica
IN_TREINEIRO	Indica se o inscrito fez a prova com intuito de apenas treinar seus conhecimentos	Numérica
SG_UF_PROVA	Sigla da Unidade da Federação da aplicação da prova	Alfanumérica
TP_STATUS_REDACAO	Situação da redação do participante	Numérica
TP_PRESENCA_CN	Presença na prova objetiva de Ciências da Natureza	Numérica
TP_PRESENCA_CH	Presença na prova objetiva de Ciências Humanas	Numérica
TP_PRESENCA_LC	Presença na prova objetiva de Linguagens e Códigos	Numérica
TP_PRESENCA_MT	Presença na prova objetiva de Matemática	Numérica
NU_NOTA_CN	Nota da prova de Ciências da Natureza	Numérica
NU_NOTA_CH	Nota da prova de Ciências Humanas	Numérica
NU_NOTA_LC	Nota da prova de Linguagens e Códigos	Numérica
NU_NOTA_MT	Nota da prova de Matemática	Numérica
NU_NOTA_COMP1	Nota da competência 1 da redação	Numérica
NU_NOTA_COMP2	Nota da competência 2 da redação	Numérica
NU_NOTA_COMP3	Nota da competência 3 da redação	Numérica
NU_NOTA_COMP4	Nota da competência 4 da redação	Numérica
NU_NOTA_COMP5	Nota da competência 5 da redação	Numérica
NU_NOTA_REDACAO	Nota da prova de redação	Numérica

Fonte: Inep (2026).

Tabela 3.3 – Continuação da Descrição das variáveis presentes na base de dados do ENEM.

Variável	Descrição	Tipo da Variável
Q001	Qual a escolaridade do Pai?	Categórica
Q002	Qual a escolaridade da Mãe?	Categórica
Q003	Qual ocupação do seu Pai?	Categórica
Q004	Qual ocupação da sua Mãe?	Categórica
Q005	Qual a quantidade de moradores na sua residência	Numérica
Q006	Qual a sua renda familiar?	Categórica
Q007	Em sua residência trabalha empregado(a) doméstico(a)?	Categórica
Q008	Na sua residência existe banheiro?	Categórica
Q009	Quantidade de quartos na sua residência?	Categórica
Q010	Na sua residência tem carro?	Categórica
Q011	Na sua residência tem motocicleta?	Categórica
Q012	Na sua residência tem geladeira?	Categórica
Q013	Na sua residência tem freezer?	Categórica
Q014	Na sua residência tem máquina de lavar roupa?	Categórica
Q015	Na sua residência tem máquina de secar roupa?	Categórica
Q016	Na sua residência tem forno micro-ondas?	Categórica
Q017	Na sua residência tem máquina de lavar louça?	Categórica
Q018	Na sua residência tem aspirador de pó?	Categórica
Q019	Na sua residência tem televisão em cores?	Categórica
Q020	Na sua residência tem aparelho de DVD?	Categórica
Q021	Na sua residência tem TV por assinatura?	Categórica
Q022	Na sua residência tem telefone celular?	Categórica
Q023	Na sua residência tem telefone fixo?	Categórica
Q024	Na sua residência tem computador?	Categórica
Q025	Na sua residência tem acesso à Internet?	Categórica

Fonte: Inep (2026).

3.5 Preparação dos Dados

Nesta etapa, foram realizadas tarefas como seleção, formatação e integração dos dados para compor um novo conjunto de informações que servirá de base para a pesquisa. Durante o processo de preparação, uma nova coluna foi adicionada ao conjunto de dados, contendo a pontuação média dos participantes. Essa pontuação é calculada somando as notas das quatro áreas do conhecimento (atributos NU_NOTA_CN, NU_NOTA_CH, NU_NOTA_LC, NU_NOTA_MT) e dividindo o total por quatro (Macedo; Saporetti, 2023). Como o conjunto de dados original disponibiliza apenas as médias individuais de cada área, essa nova coluna foi criada para representar a média geral do estudante.

Além disso, foi gerada outra coluna para classificar o desempenho dos estudantes com base na média do ENEM, que é de 600 pontos, e no estudo de Franco *et al.* (2020). Para isso, atribuiu-se o valor 1 às notas maiores ou iguais a 600, indicando um bom desempenho no exame, e o valor 0 às notas abaixo desse limite, representando um desempenho não esperado no exame. Após essa etapa, todas as linhas com valores ausentes, correspondentes a candidatos que não compareceram à prova, foram removidas, e também os estudantes que não compareceram nos dois dias de prova.

3.5.1 Seleção

Na base de dados do ENEM 2023, foram considerados os atributos mais relevantes dos estudantes para realizar as análises (53 atributos no total), que são os atributos presentes nas Tabelas 3.2 e 3.3, baseados nos estudos de Dutra (2024), Neiva (2023), NETO *et al.* (2023) e Máximo & Ribeiro (2023). Os critérios que esses estudos utilizaram para selecionar os atributos foram características do participante, como sexo, faixa etária e trajetória escolar, informações do contexto escolar (tipo/dependência e localização da escola) e variáveis do questionário socioeconômico e familiar, uma vez que este reúne indicadores de perfil socioeconômico, contexto familiar, trajetória educacional e situação profissional. Para realizar a leitura da base de dados, na qual está no formato *Comma Separated Values* (CSV), foi utilizada a biblioteca *Pandas*[2].

[2] https://pandas.pydata.org/docs/

3.5.2 Pré-Processamento

Foi realizada uma análise das variáveis para identificar a presença de valores nulos, ausentes e *outliers*, que correspondem a dados inseridos incorretamente ou que apresentam valores significativamente maiores ou menores em comparação com o restante do conjunto de dados. Para evitar que essas inconsistências comprometessem as análises, foi aplicado um processo de tratamento adequado. Essa etapa contou com o suporte de funções da biblioteca *Pandas*, como a função *dropna*, utilizada para remover linhas com valores ausentes, e a função *drop*, empregada para excluir atributos não relevantes para a análise, que são atributos de identificação ou controle, como códigos e campos administrativos, os quais não contribuem diretamente para a tarefa de classificação. Após a aplicação dessas funções, a base do ENEM 2023 passou a conter 708.614 amostras.

3.5.3 Transformação

Com o auxílio da biblioteca *Pandas*, foi realizada uma verificação dos tipos de dados dos atributos presentes no conjunto de dados, a fim de identificar possíveis necessidades de conversão para outros formatos, garantindo a integridade das análises. Para essa verificação, utilizou-se a função *dtypes*, que permite visualizar os tipos de dados de cada atributo na base.

Para tratar atributos categóricos representados em formato de texto, foi empregada a função *LabelEncoder*, disponível no módulo *preprocessing* da biblioteca *Scikit-learn*[3]. Essa função transforma valores do tipo *String* em representações numéricas inteiras, o que possibilita que os algoritmos de AM tratem essas variáveis corretamente, sem prejudicar a extração de conhecimento dos dados. Por exemplo, o atributo de sexo, originalmente representado por "F" para feminino e "M" para masculino, foi transformado em valores numéricos, onde "F" corresponde a 1 e "M" a 0. Da mesma forma, o atributo de cor/raça passou por uma codificação, em que "Não declarado" foi atribuído ao valor 0, "Branca" ao valor 1, "Preta" ao valor 2, "Parda" ao valor 3, "Amarela" ao valor 4 e "Indígena" ao valor 5. Além disso, foi empregada a função *MinMaxScaler* para normalizar os dados, disponível no módulo *preprocessing* da biblioteca *Scikit-learn*. Essa função ajusta cada atributo de forma independente, convertendo seus valores para um intervalo predefinido entre 0 e 1, com base nos valores mínimo e máximo de

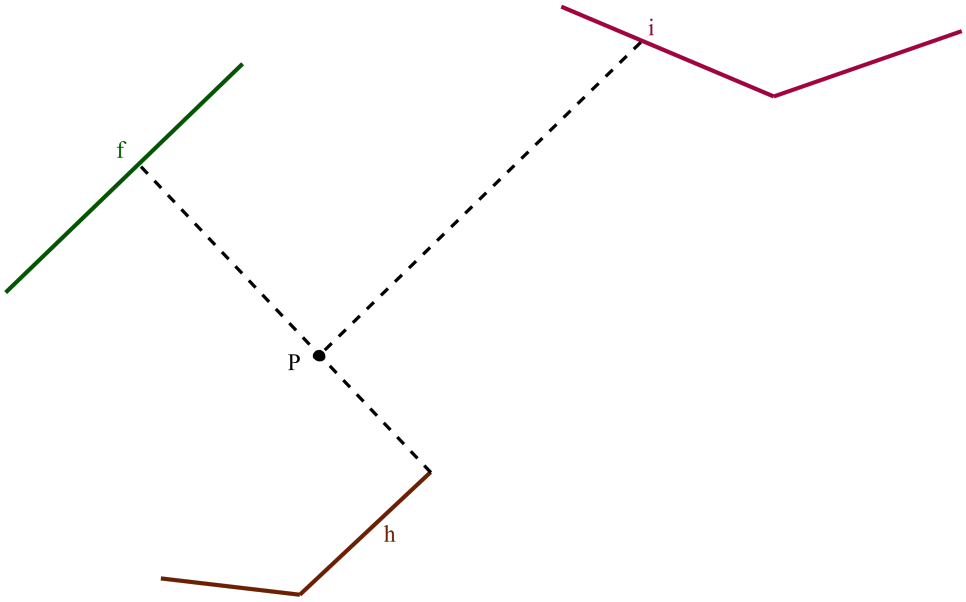
[3] https://scikit-learn.org/stable/

cada atributo. Assim, a forma da distribuição original é mantida, ao mesmo tempo em que se reduz a influência de diferenças de escala entre os atributos.

3.5.4 Classificação com o Algoritmo de CP K-segmentos

A Figura 3.3 apresenta um exemplo de classificação do algoritmo de CP K-segmentos, no qual “f”, “i” e “h” representam as CP em um espaço de características. Ao analisar uma amostra “P”, é possível calcular a distância Pf que indica a separação entre P e a curva “f”. Já a distância Pi mede a distância entre P e a curva “i”. E a distância Ph representa a distância entre P e a curva “h”. Com o cálculo dessas distâncias, é possível avaliar e classificar “P” como pertencente à classe “h”, por ter a menor distância.

Figura 3.3 – Exemplo de classificação com o método de extração de CP K-segmentos.



Fonte: Do Autor (2026).

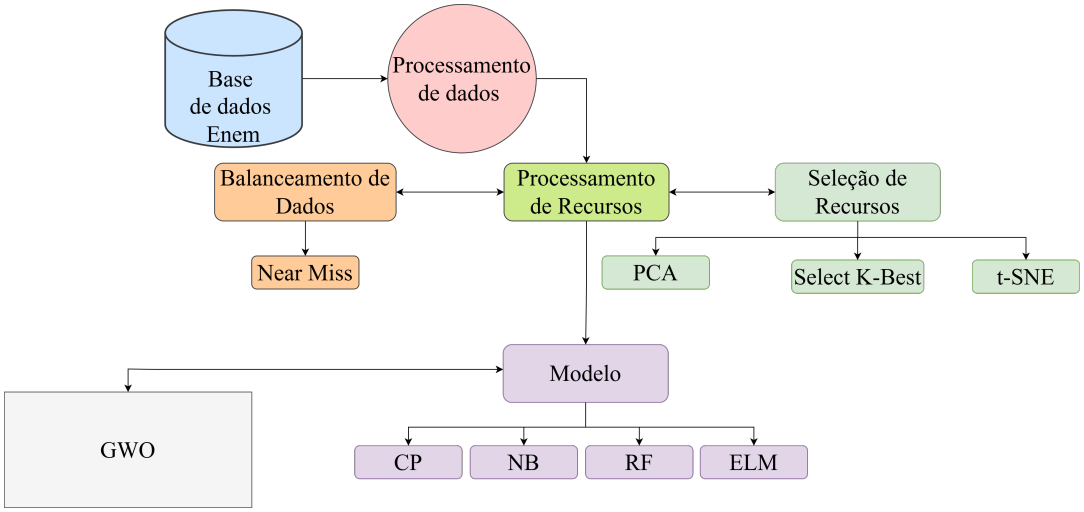
3.5.5 Diagrama de Trabalho

A Figura 3.4 apresenta um diagrama que ilustra a metodologia adotada nesta pesquisa para a construção e avaliação de modelos preditivos, utilizando a base de dados do ENEM 2023 para a classificação do desempenho acadêmico dos estudantes. O processo começa com a etapa de pré-processamento, que engloba a verificação de valores ausentes, nulos ou incorretamente registrados nos dados. Além disso, o diagrama de trabalho incorpora uma técnica de balancea-

mento para tratar o desbalanceamento de classes, empregando o método *Near Miss* (que reduz a classe majoritária, igualando-a à classe minoritária). Antes do balanceamento das classes, a distribuição era de 572.400 amostras para a Classe 0 e 136.214 amostras para a Classe 1. Após a aplicação do *Near Miss*, obteve-se uma distribuição balanceada, com 136.214 amostras para as Classes 0 e 1. Na sequência, realiza-se a seleção de atributos, uma etapa crucial para reduzir a dimensionalidade do conjunto de dados e, consequentemente, melhorar o desempenho dos modelos preditivos. Diferentes técnicas são utilizadas nesse processo, como PCA, *Select K-Best* e t-SNE.

Após a etapa de preparação, os dados são utilizados no treinamento de diferentes algoritmos de AM, incluindo CP, NB, RF e ELM. Para aprimorar o desempenho desses modelos, aplica-se a técnica de otimização GWO, que é uma meta-heurística que explora de forma contínua o espaço de busca, definindo um intervalo de valores para cada hiperparâmetro e testando diversas combinações. O conjunto de dados foi particionado em três subconjuntos: treinamento, validação e teste. No processo de ajuste e avaliação dos modelos, a validação cruzada *5-fold* é aplicada sobre o conjunto de treinamento (e seus respectivos subconjuntos de validação em cada *fold*), dividindo-o em cinco partes e alternando, a cada iteração, quais partições são utilizadas para treino e para validação. Após esse processo, o subconjunto de teste, mantido separado, é utilizado para validar os resultados obtidos e verificar a capacidade de generalização do modelo fora das partições utilizadas no *k-fold*. Na etapa final, o desempenho dos modelos é analisado com base em métricas como acurácia, precisão, *recall*, *F1-score* e *kappa*. Por fim, os resultados obtidos são avaliados para identificar o modelo mais eficiente e sua configuração ideal.

Figura 3.4 – Diagrama do Trabalho.



Fonte: Do Autor (2026).

4 RESULTADOS

Neste capítulo, são apresentados e analisados os resultados obtidos a partir da aplicação dos métodos de classificação considerados nesta pesquisa. Os resultados são discutidos com base nas métricas de desempenho adotadas, permitindo a comparação entre os diferentes métodos e a avaliação do comportamento dos métodos frente as bases de dados analisadas. Na primeira etapa, a metodologia proposta é validada em bases de dados de referência da literatura e simuladas, a fim de avaliar o desempenho sob diferentes características. Em seguida, procede-se à avaliação na base de dados do ENEM.

4.1 Experimentos em Bases de Dados da Literatura

A Tabela 4.1 apresenta a descrição dos métodos utilizados no GWO, 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, além de indicar os valores usados durante as 30 execuções independentes para avaliar a metodologia.

Tabela 4.1 – Descrição dos Métodos.

Método	**Parâmetros**	**Variação dos Parâmetros**
CPC	*k_max*	[2, 5]
	f	[0.1, 1.0]
	lambda	[0.05, 1.0]
RF	*n_estimators*	[1, 100]
	max_depth	[1, 30]
	min_samples_split	[2, 30]
	min_samples_leaf	[1, 10]
	criterion	[“*gini*”, “*entropy*”]
NB	*var_smoothing*	$[1e^{-9}, 1e^{-1}]$
ELM	*n_neurons*	[1, 600]
	ufunc	[“*relu*”, “*tanh*”]

Os primeiros experimentos com o algoritmo CPC otimizado pelo método GWO foram realizados em bases de dados da literatura (Tabelas 4.2 a 4.8), utilizando uma configuração de parâmetros adotada em estudos da área. O número de gerações (iterações) foi estabelecido em 30, um valor considerado apropriado para assegurar a convergência do método sem gerar um alto custo computacional (Dagal *et al.*, 2025; Shehab; Taherdangkoo; Butscher, 2024; Li *et al.*, 2022). Esse limite de iterações determina quantas vezes a população de soluções é atualizada durante o processo de otimização. Em relação ao tamanho da população, escolheu-se 50 lobos, o que significa que 50 soluções candidatas são preservadas e atualizadas a cada geração (Dagal *et al.*, 2025; Shehab; Taherdangkoo; Butscher, 2024; Li *et al.*, 2022). Esse valor foi selecionado por representar um equilíbrio entre a diversidade da busca, minimizando o risco de estagnação prematura, e a viabilidade computacional, dado que populações excessivamente grandes elevam consideravelmente o tempo de execução sem melhorias significativas no desempenho (Xia; Wang, 2025; Li *et al.*, 2025; Li *et al.*, 2022).

A Tabela 4.2 apresenta a análise do desempenho do algoritmo CPC, empregando o método GWO na base de dados *Breast_Cancer*. Essa avaliação compara o CPC com os métodos RF, NB e ELM, levando em conta as métricas de ACC, F1, PR, RE, Kappa e tempo de execução (em segundos). Observa-se que o CPC apresenta um desempenho competitivo em todas as métricas. O método alcançou uma ACC de 0,9747, F1 de 0,9746, PR igual a 0,9752 e RE de 0,9747. Esses valores são um pouco superiores aos do RF e ELM. Ademais, a métrica do *Kappa*, que avalia a concordância entre previsões e rótulos reais, demonstra a robustez do CPC, com um valor de 0,9456, indicando uma excelente consistência.

Em termos de tempo de execução, o método CPC leva aproximadamente 1408 segundos para ser executado, um valor superior ao observado para os métodos ELM e NB, mas comparável ao RF. Ressalta-se que esse tempo refere-se à fase de projeto (desenvolvimento) do método, pois envolve sua construção. Além disso, é nessa etapa que ocorre a extração das CPs, o que contribui para o aumento do custo computacional. Isso reflete o custo computacional relacionado à otimização de segmentos. Esse equilíbrio entre precisão e tempo de processamento é comum em técnicas que utilizam o cálculo da distância, nas quais a demanda computacional aumenta.

A Tabela 4.3 apresenta o melhor método do CPC, no conjunto de teste. O método é composto por um *k_max* igual a 4, indicando que a curva será representada por até quatro segmentos. Um f de parâmetro de suavização ou fração de vizinhança utilizada durante a esti-

Tabela 4.2 – Avaliação do algoritmo baseado em CP com GWO na base de dados da literatura *Breast_Cancer*.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	0,9747 ± 0,0029	0,9746 ± 0,0030	0,9752 ± 0,0029	0,9747 ± 0,0029	0,9456 ± 0,0065	1408,4651 ± 283,2643
RF	0,9686 ± 0,0036	0.9685 ± 0,0036	0,9693 ± 0,0035	0,9686 ± 0,0036	0,9323 ± 0,0077	883,2674 ± 275,3582
NB	0,9406 ± 0,0034	0,9399 ± 0,0033	0,9426 ± 0,0039	0,9406 ± 0,0034	0,8704 ± 0,0069	47,0223 ± 1,6271
ELM	0,9612 ± 0,0017	0,9609 ± 0,0068	0,9622 ± 0,0067	0,9612 ± 0,0067	0,9157 ± 0,0145	65,8110 ± 3,5837

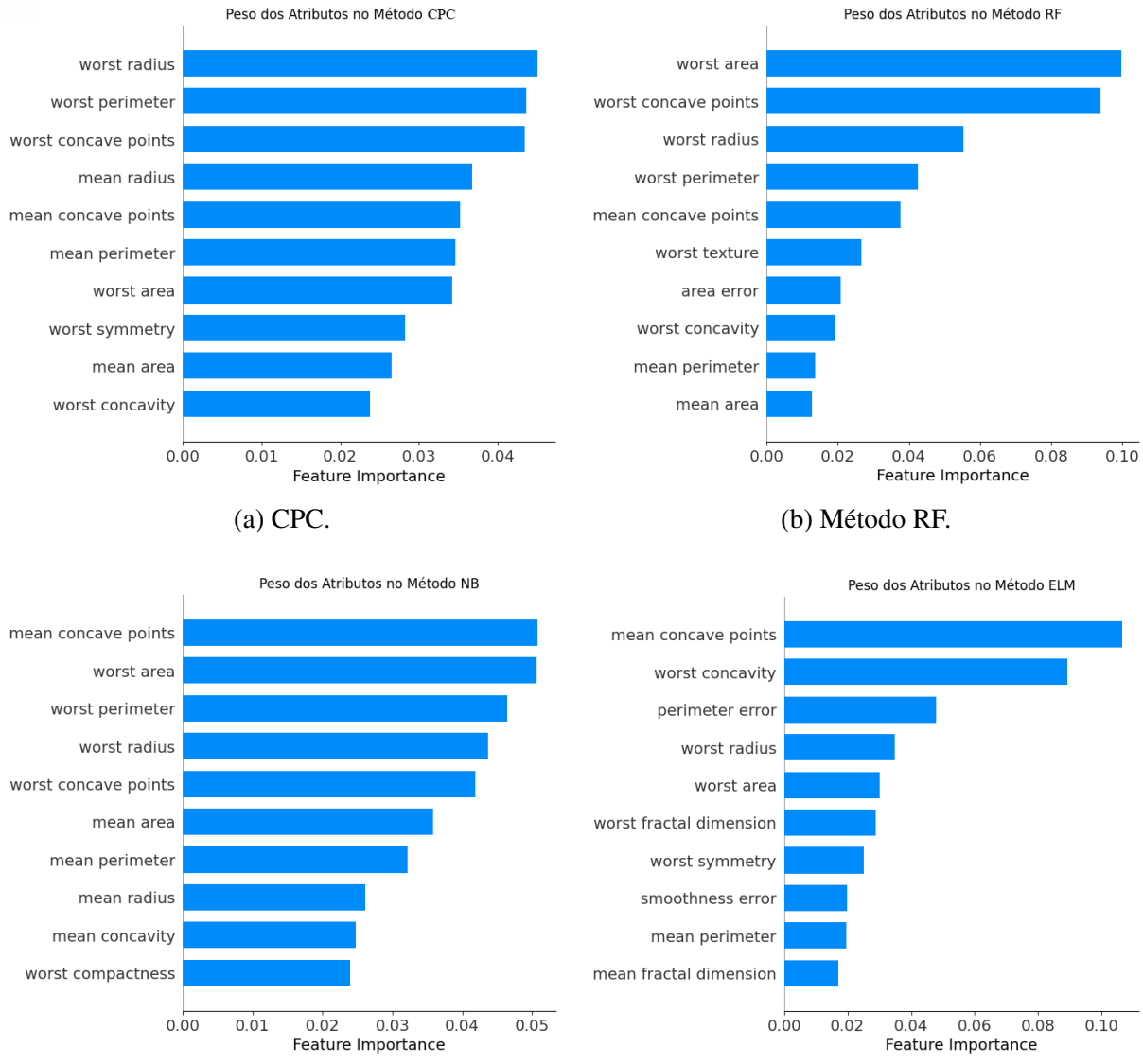
mativa local da curva igual a 0,8688 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,9379. Tal configuração resultou, no conjunto de teste, em acurácia de 0,9737, precisão de 0,9739, *recall* de 0,9737, *F1-Score* de 0,9737 e um *kappa* no valor de 0,9442. Além disso, após o método ser ajustado, o tempo de execução na fase operacional, quando o método é utilizado apenas para realizar predições, foi de aproximadamente 0,4362 segundos, refletindo um processo mais simples e eficiente em comparação à fase de projeto, na qual ocorre a extração das curvas e o ajuste dos parâmetros.

Tabela 4.3 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados da literatura *Breast_Cancer*.

Parâmetros	ACC	F1	PR	RE	Kappa	Tempo (seg)
k_max: 4, f: 0,8688, *lambda*: 0,9379	0,9737	0,9737	0,9739	0,9737	0,9442	0,4362

A Figura 4.1 ilustra a importância dos atributos dos métodos na classificação da base *Breast_Cancer*, utilizando o método SHAP. Nota-se que cada método prioriza conjuntos distintos de características, o que reflete a natureza particular de cada algoritmo na identificação de padrões. Embora os métodos variem, todos enfatizam características relacionadas à forma e à concavidade das células, ressaltando sua relevância na classificação do câncer de mama. Para o algoritmo CPC, os atributos mais relevantes foram *worst radius*, *worst perimeter* e *worst concave points*. Isso indica que características ligadas a medidas extremas da forma e concavidade das células foram cruciais para a classificação. Entretanto, o RF possui *worst area* e *worst concave points* como os atributos mais relevantes. Já os métodos NB e ELM possuem *mean concave points* como o atributo mais significativo.

Figura 4.1 – Melhores Características dos Métodos na Base de Dados da Literatura *Breast_Cancer*.



(a) CPC.

(b) Método RF.

(c) Método NB.

(d) Método ELM.

Fonte: Do Autor (2026).

Já a Tabela 4.4 mostra a análise do desempenho do algoritmo CPC otimizado com GWO usando a base de dados *Iris*. Essa análise compara o CPC com os métodos RF, NB e ELM, empregando as métricas de ACC, F1, PR, RE, Kappa e tempo de execução (em segundos). Os resultados mostram que o CPC também teve um bom desempenho na base de dados *Iris*. O algoritmo obteve ACC de 0,9750, F1 de 0,9749, PR de 0,9772 e RE igual a 0,9750. Esses valores superaram os métodos NB, RF e ELM. Além disso, o valor de Kappa foi de 0,9625, demonstrando uma elevada consistência do modelo, indicando confiabilidade na previsão das classes da base de dados *Iris*. Em relação ao tempo de execução, o CPC apresentou aproximadamente 433 segundos, um valor maior que o ELM e o NB, mas consideravelmente menor que o RF. Esse tempo refere-se à fase de projeto do método, que envolve a construção e extração das CPs.

Tabela 4.4 – Avaliação do algoritmo baseado em CP com GWO na base de dados da literatura *Iris*.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,9750 ± 0,0043	0,9749 ± 0,0043	0,9772 ± 0,0044	0,9750 ± 0,0043	0,9625 ± 0,0064	433,6238 ± 203,8622
RF	0,9583 ± 0,0071	0,9581 ± 0,0073	0,9623 ± 0,0068	0,9583 ± 0,0071	0,9375 ± 0,0107	659,4184 ± 310,1502
NB	0,9478 ± 0,0057	0,9474 ± 0,0057	0,9525 ± 0,0058	0,9478 ± 0,0057	0,9216 ± 0,0085	44,2302 ± 1,3908
ELM	0,9625 ± 0,0033	0,9623 ± 0,0129	0,9672 ± 0,0116	0,9625 ± 0,0033	0,9437 ± 0,0193	52,4609 ± 41,8152

A Tabela 4.5 apresenta o melhor método do CPC no conjunto de teste. O método é composto por um *k_max* igual a 2, indicando 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 0,2230 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,2307. Tal configuração resultou, no conjunto de teste, em 1,0000 para todas as métricas e um tempo de 0,0529 segundos, que é o tempo na fase operacional.

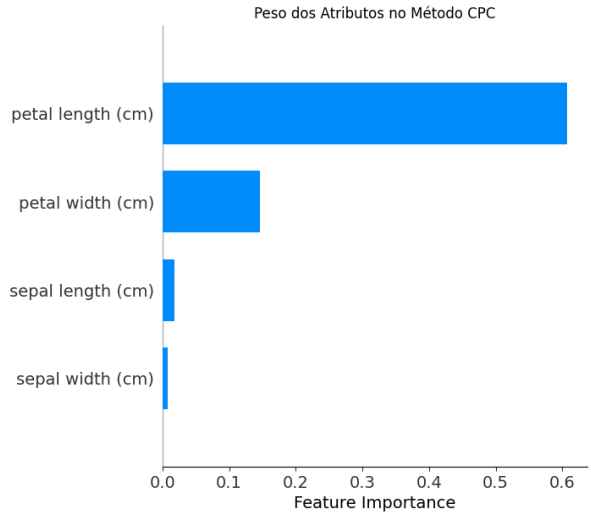
Tabela 4.5 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados da literatura *Iris*.

Parâmetros	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
k_max: 2, f: 0,2230, *lambda*: 0,2307	1,0000	1,0000	1,0000	1,0000	1,0000	0,0529

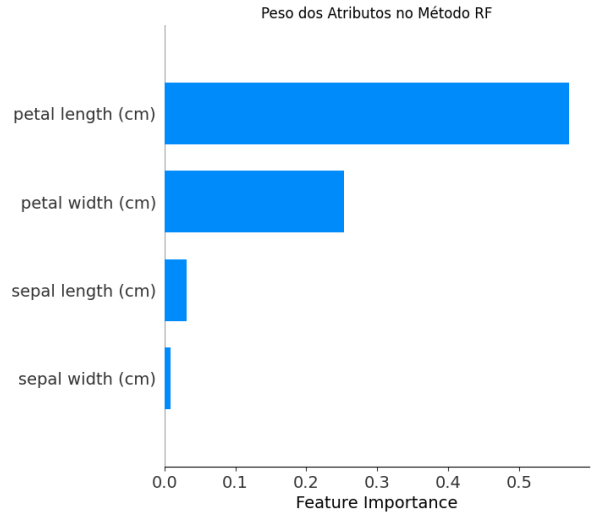
A Figura 4.2 apresenta a importância dos atributos empregados pelos métodos CPC, RF, NB e ELM na classificação da base *Iris*. Percebe-se que todos os métodos priorizam os atributos ligados às pétalas, particularmente *petal length (cm)* e *petal width (cm)*, ao passo que os atributos associados às sépalas (*sepal length (cm)* e *sepal width (cm)*) têm menor relevância.

No CPC, destaca-se o comprimento da pétala, seguido pela largura da pétala, indicando que as medidas das pétalas são fundamentais para a classificação das espécies de *Íris*. Para o RF, também se destaca como maior importância o comprimento da pétala, porém mantém a largura da pétala como o segundo atributo mais relevante, evidenciando que ambas as medidas são fundamentais para definir as fronteiras de decisão. Por outro lado, o NB atribui importância de maneira mais equilibrada entre o comprimento e a largura da pétala, com uma contribuição menor das sépalas. Ademais, no ELM, segue o padrão semelhante dos outros métodos, priorizando *petal length* e *petal width*.

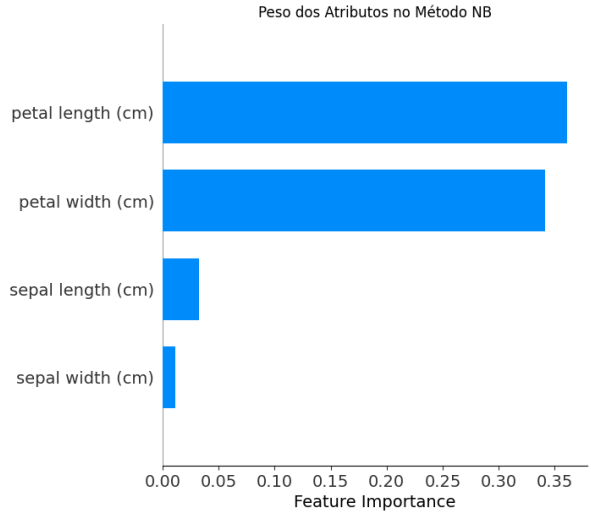
Figura 4.2 – Melhores Características dos Métodos na Base de Dados da Literatura *Iris*.



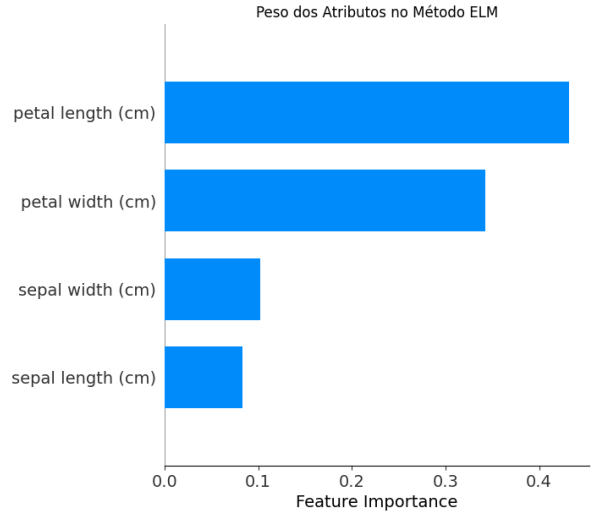
(a) Método CPC.



(b) Método RF.



(c) Método NB.



(d) Método ELM.

Fonte: Do Autor (2026).

Na Tabela 4.6, apresenta-se a análise do desempenho do algoritmo CPC otimizado com o GWO na base de dados *Wine*. Também foram utilizados os algoritmos RF, NB e ELM para comparação. O CPC apresentou ACC igual a 0,9731, F1 de 0,9730, PR de 0,9763 e RE igual a 0,9731, valores que superam os algoritmos NB e ELM e que são comparáveis ao RF. A estatística de *Kappa* igual a 0,9594 indica uma excelente concordância entre as previsões do modelo e os rótulos reais, evidenciando um bom desempenho na classificação da base *Wine*. Quanto ao tempo de execução (na fase de projeto), o CPC apresentou aproximadamente 556 segundos, maior do que os métodos ELM e NB, mas competitivo em relação ao RF.

Tabela 4.6 – Avaliação do algoritmo baseado em CP com GWO na base de dados da literatura *Wine*.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	0,9731 ± 0,0091	0,9730 ± 0,0091	0,9763 ± 0,0081	0,9731 ± 0,0091	0,9594 ± 0,0137	556,0105 ± 219,1962
RF	0,9843 ± 0,0050	0,9843 ± 0,0049	0,9860 ± 0,0044	0,9843 ± 0,0050	0,9764 ± 0,0076	759,1893 ± 307,7103
NB	0,9626 ± 0,0057	0,9625 ± 0,0059	0,9666 ± 0,0052	0,9626 ± 0,0057	0,9435 ± 0,0087	44,8086 ± 2,4155
ELM	0,9613 ± 0,0044	0,9611 ± 0,0144	0,9655 ± 0,0129	0,9613 ± 0,0142	0,9416 ± 0,0214	181,2734 ± 70,5106

Já a Tabela 4.7 apresenta o melhor método do CPC no conjunto de teste. O método é composto por um *k_max* igual a 5, indicando que a curva será representada por até cinco segmentos. Um f de parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva igual a 0,2592 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,0733. Tal configuração resultou, no conjunto de teste, em acurácia de 1,0000, precisão de 1,0000, *recall* de 1,0000 e um *F1-Score* de 1,0000, *kappa* igual a 1,0000 e um tempo de 0,1029 segundos na fase operacional, quando o método é utilizado apenas para realizar predições, refletindo um processo mais simples à fase de projeto.

Tabela 4.7 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados da literatura *Wine*.

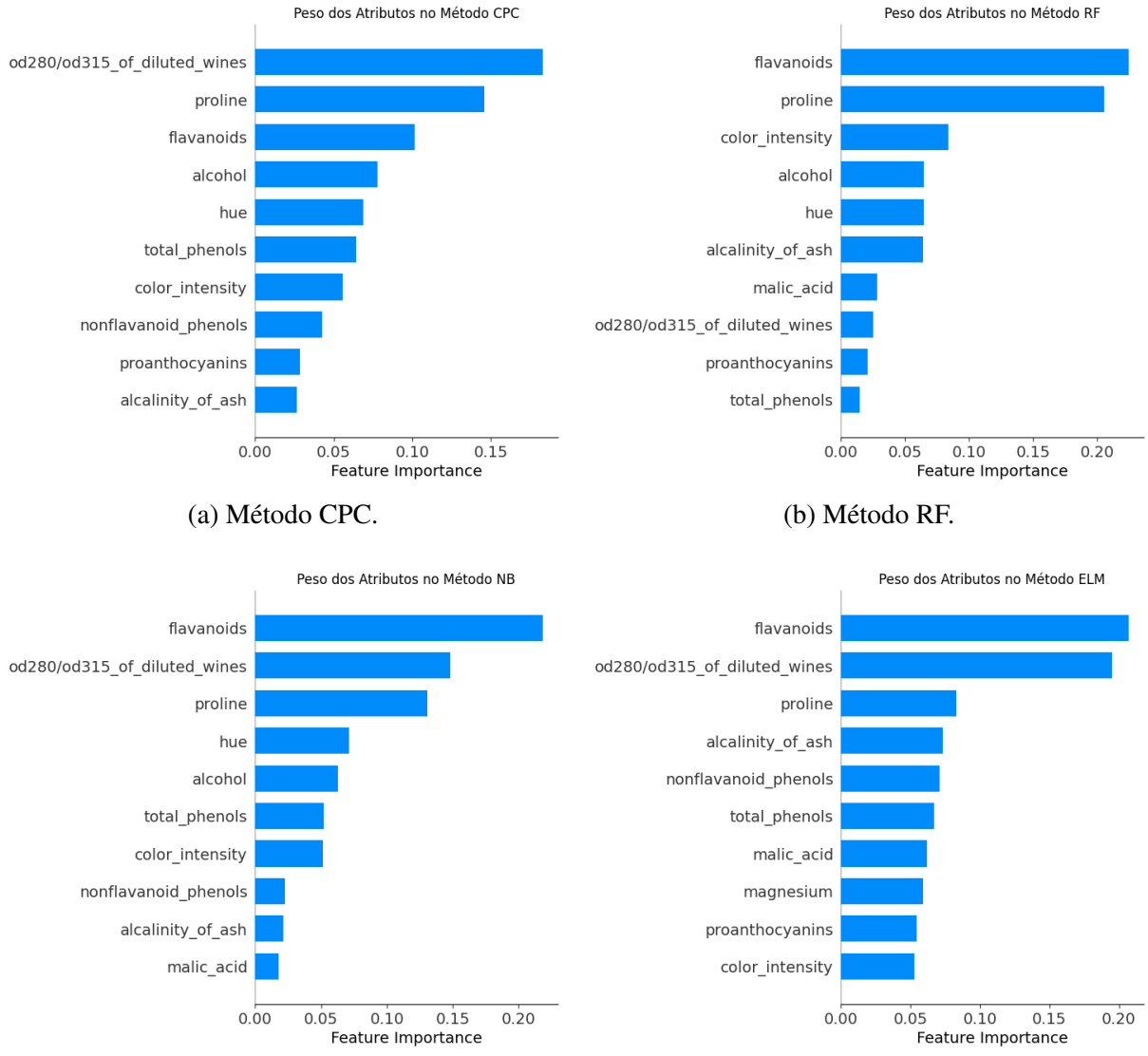
Parâmetros	ACC	F1	PR	RE	Kappa	Tempo (seg)
k_max: 5, f: 0,2592, *lambda*: 0,0733	1,0000	1,0000	1,0000	1,0000	1,0000	0,1029

A Figura 4.3 apresenta a importância dos atributos para a classificação na base *Wine*, ressaltando as variações na importância de cada *feature* entre os diferentes métodos. No CPC, os atributos mais relevantes são *od280/od315_of_diluted_wines* e *proline*, sugerindo que as análises químicas ligadas à concentração de compostos fenólicos e aminoácidos são essenciais

para a classificação das amostras. *Flavonoides*, *alcohol* e *hue* têm relevância intermediária, ao passo que atributos como *total_phenols* e *alcalinity_of_ash* são considerados menos importantes. Além disso, o ELM prioriza *flavanoids* e *od280/od315_of_diluted_wines*, destacando a importância dessas medidas químicas na distinção das classes.

Já o RF destaca *flavanoids* e *proline* como principais atributos, seguido por *color_intensity*. A presença de *alcohol* e *hue* reforça a relevância de características relacionadas à cor e teor alcoólico das amostras. Por fim, no NB, os atributos *flavanoides* e *od280/od315_of_diluted_wines* são os mais relevantes, seguidos por *proline* e *hue*. Isso sugere que o método probabilístico integra dados de concentração química e propriedades físicas para realizar a classificação.

Figura 4.3 – Melhores Características dos Métodos na Base de Dados da Literatura *Wine*.



(a) Método CPC.

(b) Método RF.

(c) Método NB.

(d) Método ELM.

Fonte: Do Autor (2026).

Por fim, a Tabela 4.8 mostra a avaliação do desempenho do algoritmo CPC otimizado pelo GWO utilizando a base de dados *Thyroid*. Os resultados foram comparados com os métodos RF, NB e ELM. Observa-se que, diferentemente das outras bases, o CPC apresentou desempenho inferior aos métodos comparados para esta base. O CPC apresentou uma ACC de 0,9000, F1 de 0,8999, PR de 0,9023 e RE igual a 0,9000. O valor de *Kappa* igual a 0,8001 também indica menor concordância em relação aos demais métodos, evidenciando que o CPC teve menor consistência na classificação dos dados da base *Thyroid*. Já em relação ao tempo de execução, o CPC levou aproximadamente 628 segundos (fase de projeto) para ser executado, sendo mais lento que o RF, NB e ELM. O tempo de execução do processamento do CPC deixa evidente que as técnicas que utilizam o cálculo da distância têm uma demanda maior no custo computacional durante o treinamento.

Tabela 4.8 – Avaliação do algoritmo baseado em CP com GWO na base de dados da literatura *Thyroid*.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,9000 ± 0,0051	0,8999 ± 0,0052	0,9023 ± 0,0054	0,9000 ± 0,0051	0,8001 ± 0,0103	628,3005 ± 199,1843
RF	0,9548 ± 0,0043	0,9547 ± 0,0043	0,9561 ± 0,0041	0,9548 ± 0,0043	0,9095 ± 0,0087	460,9006 ± 234,3851
NB	0,9053 ± 0,0051	0,9044 ± 0,0055	0,9077 ± 0,0053	0,9053 ± 0,0051	0.7675 ± 0.0138	45,7428 ± 2,0053
ELM	0,9038 ± 0,0067	0,9019 ± 0,0135	0,9053 ± 0,0138	0,9038 ± 0,0132	0,7591 ± 0,0333	141,5565 ± 96,1147

Já a Tabela 4.9 apresenta o melhor método do CPC, no conjunto de teste. O método é composto por um *k_max* igual a 3, indicando que a curva será representada por até três segmentos. Um f de parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva igual a 0,4305 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,7988. Tal configuração resultou, no conjunto de teste, em uma acurácia de 0,9091, precisão de 0,9078, *recall* de 0,9091, um *F1-Score* de 0,9083, *kappa* igual a 0,7510 e um tempo de 0,0522 segundos (fase operacional).

Tabela 4.9 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados da literatura *Thyroid*.

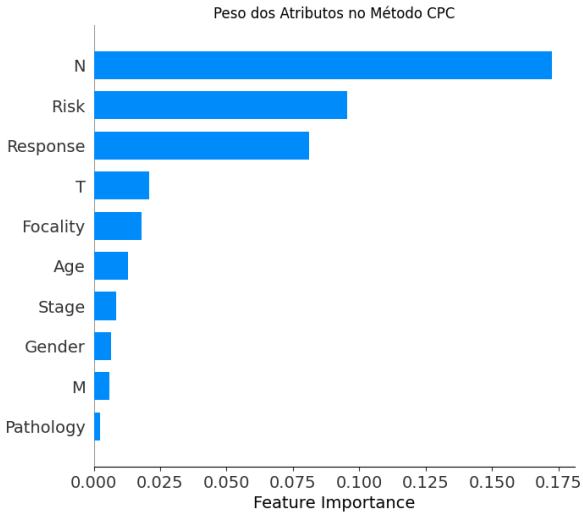
Parâmetros	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
k_max: 3, f: 0,4305, *lambda*: 0,7988	0,9091	0,9083	0,9078	0,9091	0,7510	0,0522

A Figura 4.4 ilustra a importância dos atributos na classificação da base *Thyroid* utilizando os métodos CPC, RF, NB e ELM. O CPC prioriza os atributos *N* e *Risk*, com *Response*

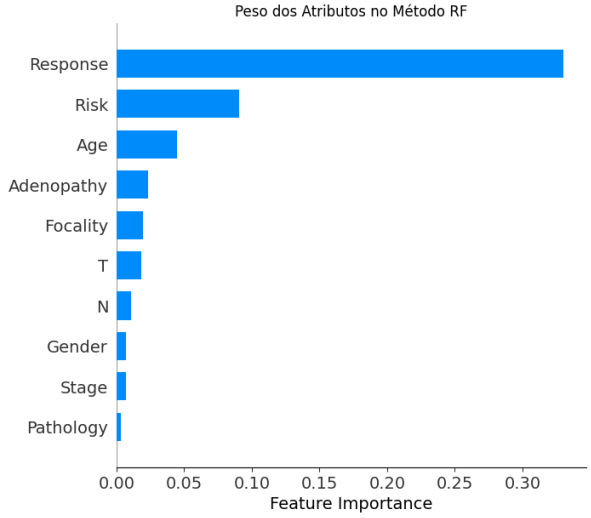
e *T* em seguida, sugerindo que elementos clínicos associados ao risco e à contagem de nódulos são os mais relevantes para a classificação das amostras. Já o ELM considera *Response* como o atributo mais importante, seguido por *N* e *Stage*, indicando que o modelo dá mais importância a variáveis relacionadas à resposta clínica.

Ademais, o RF prioriza *Response* como o atributo mais relevante, com *Risk* e *Age* em seguida. No NB, *Risk* é o atributo mais relevante, seguido por *Response* e N, o que indica que o modelo probabilístico foca nos fatores clínicos mais significativos para a classificação. De forma geral, todos os métodos priorizaram *Risk* e *Response* como os atributos mais relevantes na base de dados *Thyroid*, destacando a relevância dos fatores clínicos para a classificação.

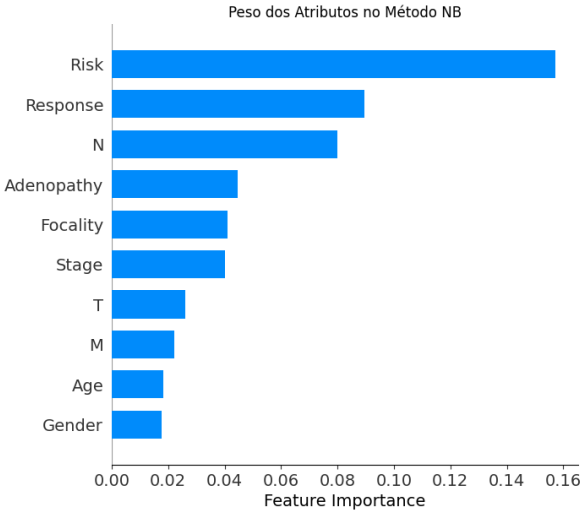
Figura 4.4 – Melhores Características dos Métodos na Base de Dados da Literatura *Thyroid*.



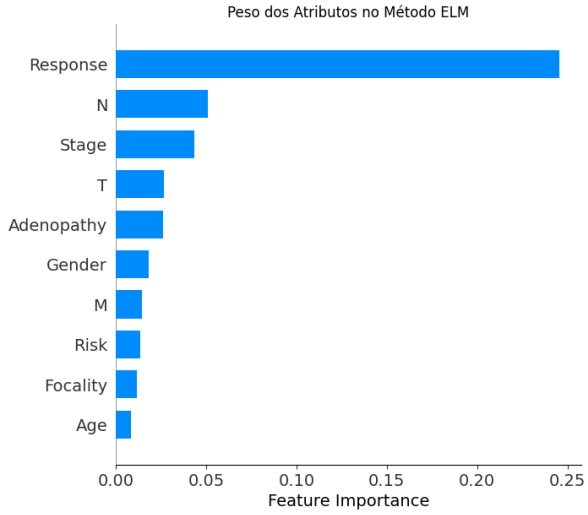
(a) Método CPC.



(b) Método RF.



(c) Método NB.



(d) Método ELM.

Fonte: Do Autor (2026).

Os experimentos realizados mostraram que os algoritmos CPC e RF alcançaram os melhores desempenhos nas bases de dados *Breast_Cancer*, *Iris* e *Wine*. Entretanto, para a base de dados *Thyroid*, o melhor algoritmo foi o RF, mas o CPC teve resultados competitivos com o NB e ELM, mostrando-se também uma boa alternativa.

4.2 Experimentos em Bases de Dados Sintéticas

Para expandir a avaliação empírica da metodologia, foram realizados testes experimentais adicionais do método GWO com o algoritmo CPC em bases de dados sintéticas, descritas na Seção 3.3. O objetivo foi avaliar se o algoritmo preserva um bom desempenho frente a diferentes geometrias e graus de separabilidade, antes de aplicá-lo aos dados do ENEM.

4.2.1 Experimentos na Base de Dados Compacta

A Tabela 4.10 mostra a avaliação do desempenho do algoritmo CPC otimizado pelo GWO em relação a base de dados de *clusters* compactos. Percebe-se um desempenho ótimo das métricas de classificação (ACC, F1, PR, RE e Kappa) com valores iguais a 1,0000 para os métodos CPC, NB e ELM. Já o RF obteve um resultado inferior ao dos outros métodos. Em relação ao custo computacional, o CPC levou aproximadamente 1320 segundos no treinamento, sendo mais lento do que o RF, NB e ELM. Esse tempo justifica-se pelo fato de estar na fase de projeto, que envolve a construção do método CPC. Nesta etapa, ocorre a extração das CPs, o que contribui para o aumento do custo computacional.

Tabela 4.10 – Avaliação do algoritmo baseado em CP com GWO na base de dados de *clusters* compactos.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	1,0000 ± 0,0000	1,0000 ± 0,0000	1,0000 ± 0,0000	1,0000 ± 0,0000	1,0000 ± 0,0000	1319,9461 ± 588,1153
RF	0,9999 ± 0,0003	0,9999 ± 0,0003	0,9999 ± 0,0003	0,9999 ± 0,0003	0,9998 ± 0,0006	384,1456 ± 208,6238
NB	1,0000 ± 0,0000	1,0000 ± 0,0000	1,0000 ± 0,0000	1,0000 ± 0,0000	1,0000 ± 0,0000	48,8592 ± 5,6871
ELM	1,0000 ± 0,0000	0,9998 ± 0,0005	0,9998 ± 0,0005	0,9998 ± 0,0005	0,9996 ± 0,0009	200,6441 ± 87,7179

Na Tabela 4.11 é apresentado o melhor método do CPC no conjunto de teste. O método é composto por um *k_max* igual a 2, indicando que a curva será representada por até dois segmentos. Um f de parâmetro de suavização ou fração de vizinhança utilizada durante a esti-

mativa local da curva igual a 0,3954 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,5193. Tal configuração resultou, no conjunto de teste, em uma acurácia de 1,0000, precisão de 1,0000, *recall* de 1,0000, um *F1-Score* de 1,0000, *Kappa* igual a 1,0000 e um tempo de 0,9358 segundos (fase operacional).

Tabela 4.11 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados compacta.

Método	**Parâmetros**	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	*k_max*: 2, f: 0,3954, *lambda*: 0,5193	1,0000	1,0000	1,0000	1,0000	1,0000	0,9358
RF	*n_estimators*: 83, *max_depth*: 2, *min_samples_split*: 28, *min_samples_leaf*: 7, *criterion*: *entropy*	1,0000	1,0000	1,0000	1,0000	1,0000	0,1775
NB	*var_smoothing*: 0,0151	1,0000	1,0000	1,0000	1,0000	1,0000	0,1104
ELM	*n_neurons*: 338, *ufunc*: *tanh*	1,0000	1,0000	1,0000	1,0000	1,0000	0,5989

4.2.2 Experimentos na Base de Dados Alongada

Na base de dados alongada, observa-se na Tabela 4.12 um desempenho máximo para os métodos CPC e ELM. Enquanto o RF tem um desempenho quase perfeito, o NB apresenta um desempenho inferior em relação aos outros métodos. Em relação ao custo computacional, o CPC teve um custo de treinamento superior ao dos outros métodos. Isso evidencia que técnicas que abordam o cálculo da distância têm uma demanda maior em termos de custo computacional durante o treinamento. Além disso, pelo fato de a fase de projeto do método envolver a construção e a extração das CPs, isso contribui para o aumento do custo computacional.

Tabela 4.12 – Avaliação do algoritmo baseado em CP com GWO na base de dados de *clusters* alongados.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	1319,9461 $\pm$ 588,1153
RF	0,9996 $\pm$ 0,0008	0,9996 $\pm$ 0,0008	0,9996 $\pm$ 0,0008	0,9996 $\pm$ 0,0008	0,9992 $\pm$ 0,0016	654,8393 $\pm$ 294,1733
NB	0,9422 $\pm$ 0,0015	0,9419 $\pm$ 0,0015	0,9483 $\pm$ 0,0014	0,9422 $\pm$ 0,0015	0,8843 $\pm$ 0,0029	33,1976 $\pm$ 3,7707
ELM	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	1,0000 $\pm$ 0,0000	185,1527 $\pm$ 57,7485

Já na Tabela 4.13 é apresentado o melhor método do CPC no conjunto de teste. O método é composto por um *k_max* igual a 3, indicando que a curva será representada por até três segmentos, um f igual a 0,9774 e um *lambda* igual a 0,5662. Tal configuração resultou, no

conjunto de teste, em métricas com valores iguais a 1,0000 e um tempo 0,1460 segundos (fase operacional).

Tabela 4.13 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados alongada.

Método	Parâmetros	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	*k_max*: 3, *f*: 0,9774, *lambda*: 0,5662	1,0000	1,0000	1,0000	1,0000	1,0000	0,1460
RF	*n_estimators*: 47, *max_depth*: 15, *min_samples_split*: 5, *min_samples_leaf*: 1, *criterion*: *gini*	1,0000	1,0000	1,0000	1,0000	1,0000	0,1083
NB	*var_smoothing*: 0,0151	0,9000	0,8994	0,9172	0,9000	0,8013	0,0062
ELM	*n_neurons*: 338, *ufunc*: *tanh*	1,0000	1,0000	1,0000	1,0000	1,0000	0,0282

4.2.3 Experimentos na Base de Dados Esférica

Na base de dados esférica, os resultados na Tabela 4.14 ilustram que o CPC obteve bons resultados comparados ao RF e ELM, com uma acurácia de 0,9984, precisão de 0,9985, *recall* de 0,9984, um *F1-Score* de 0,9984 e *kappa* igual a 0,9968. Todavia, o método NB obteve resultados inferiores aos outros métodos. Quanto ao custo computacional, o CPC apresenta um tempo de treinamento superior aos demais métodos. Os resultados obtidos na base de dados esférica corroboram o potencial do CPC para dados com geometria esférica.

Tabela 4.14 – Avaliação do algoritmo baseado em CP com GWO na base de dados de *clusters* esféricos.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	0,9984 $\pm$ 0,0018	0,9984 $\pm$ 0,0018	0,9985 $\pm$ 0,0017	0,9984 $\pm$ 0,0018	0,9968 $\pm$ 0,0036	1694,8139 $\pm$ 775,9651
RF	0,9972 $\pm$ 0,0018	0,9972 $\pm$ 0,0018	0,9973 $\pm$ 0,0018	0,9972 $\pm$ 0,0018	0,9945 $\pm$ 0,0037	829,0259 $\pm$ 238,1403
NB	0,9785 $\pm$ 0,0021	0,9785 $\pm$ 0,0021	0,9797 $\pm$ 0,0019	0,9785 $\pm$ 0,0021	0,9571 $\pm$ 0,0043	34,1163 $\pm$ 4,4849
ELM	0,9968 $\pm$ 0,0012	0,9918$\pm$ 0,0059	0,9920 $\pm$ 0,0057	0,9918 $\pm$ 0,0059	0,9837 $\pm$ 0,0117	136,5621 $\pm$ 89,7504

A Tabela 4.15 apresenta o melhor método do CPC no conjunto de teste. O método é composto por um *k_max* igual a 4, indicando que a curva será representada por até quatro segmentos, um *f* igual a 0,3926 e um *lambda* igual a 0,9336. Tal configuração resultou, no conjunto de teste, em métricas com valores iguais a 1,0000 e um 0,1734 segundos (fase operacional).

Tabela 4.15 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados esférica.

Método	Parâmetros	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	*k_max*: 4, *f*: 0,3926, *lambda*: 0,9336	1,0000	1,0000	1,0000	1,0000	1,0000	0,1734
RF	*n_estimators*: 49, *max_depth*: 26, *min_samples_split*: 2, *min_samples_leaf*: 1, *criterion*: *gini*	1,0000	1,0000	1,0000	1,0000	1,0000	0,6409
NB	*var_smoothing*: 0,0019	0,9750	0,9750	0,9761	0,9750	0,9498	0,0054
ELM	*n_neurons*: 269, *ufunc*: *tanh*	1,0000	1,0000	1,0000	1,0000	1,0000	0,3401

4.2.4 Experimentos na Base de Dados Espiral

A tabela 4.16 apresenta os resultados para a base de dados espiral, que impõe fronteiras não lineares entre as classes. O método CPC apresenta o melhor desempenho entre os métodos, com uma ACC de 0,9868, PR de 0,9869, RE de 0,9868, um F1 de 0,99867 e *kappa* igual a 0,9735. Os resultados foram superiores aos métodos RF, NB e ELM. Além do nível de ACC, a baixa variabilidade das métricas do CPC indica a estabilidade frente às partições, evidenciando sua capacidade de capturar curvaturas complexas. Entretanto, em relação ao custo computacional, o tempo de treinamento do CPC foi mais elevado em comparação com os demais métodos, evidenciando novamente que os métodos que utilizam o cálculo da distância exigem mais custo computacional no treinamento. Além disso, pelo fato de a fase de projeto do método envolver a construção e a extração das CPs, isso contribui para o aumento do custo computacional.

Tabela 4.16 – Avaliação do algoritmo baseado em CP com GWO na base de dados de *clusters* espirais.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	0,9868 ± 0,0011	0,9867 ± 0,0011	0,9869 ± 0,0010	0,9868 ± 0,0011	0,9735 ± 0,0021	5469,0892 ± 1199,3086
RF	0,9590 ± 0,0041	0,9590 ± 0,0041	0,9596 ± 0,0039	0,9590 ± 0,0041	0,9181 ± 0,0082	1028,9638 ± 243,2612
NB	0,5145 ± 0,0053	0,5136 ± 0,0054	0,5147 ± 0,0054	0,5145 ± 0,0053	0,0289 ± 0,0107	34,7955 ± 2,7617
ELM	0,6790 ± 0,0079	0,6457± 0,0147	0,6487 ± 0,0143	0,6467 ± 0,0145	0,2937 ± 0,0289	78,1396 ± 88,0131

A Tabela 4.17 apresenta o melhor método do CPC no conjunto de teste. O método é composto por um *k_max* igual a 16, indicando que a curva será representada por até dezesseis segmentos, um *f* igual a 0,8272 e um *lambda* igual a 0,1702. Tal configuração resultou, no conjunto de teste, em uma acurácia de 0,9900, *f1-score* de 0,9899, precisão de 0,9902, *recall* de 0,9900, *kappa* igual a 0,9799 e um tempo de 2,2188 segundos (fase operacional). Para essa

base de dados em espiral, foi testada uma variação do parâmetro *k_max* de 2 a 20 segmentos, devido a complexidade das curvaturas.

Tabela 4.17 – Desempenho do melhor modelo no conjunto de teste do algoritmo baseado em CP com GWO na base de dados espiral.

Método	**Parâmetros**	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	*k_max*: 16, *f*: 0,8272, *lambda*: 0,1702	0,9900	0,9899	0,9902	0,9900	0,9799	2,2188
RF	*n_estimators*: 97, *max_depth*: 30, *min_samples_split*: 2, *min_samples_leaf*: 1, *criterion*: *entropy*	0,9550	0,9550	0,9554	0,9550	0,9097	0,5702
NB	*var_smoothing*: 0,0002	0,4550	0,4536	0,4532	0,4550	0,0948	0,0047
ELM	*n_neurons*: 59, *ufunc*: *tanh*	0,6850	0,6851	0,6851	0,6850	0,3692	0,2480

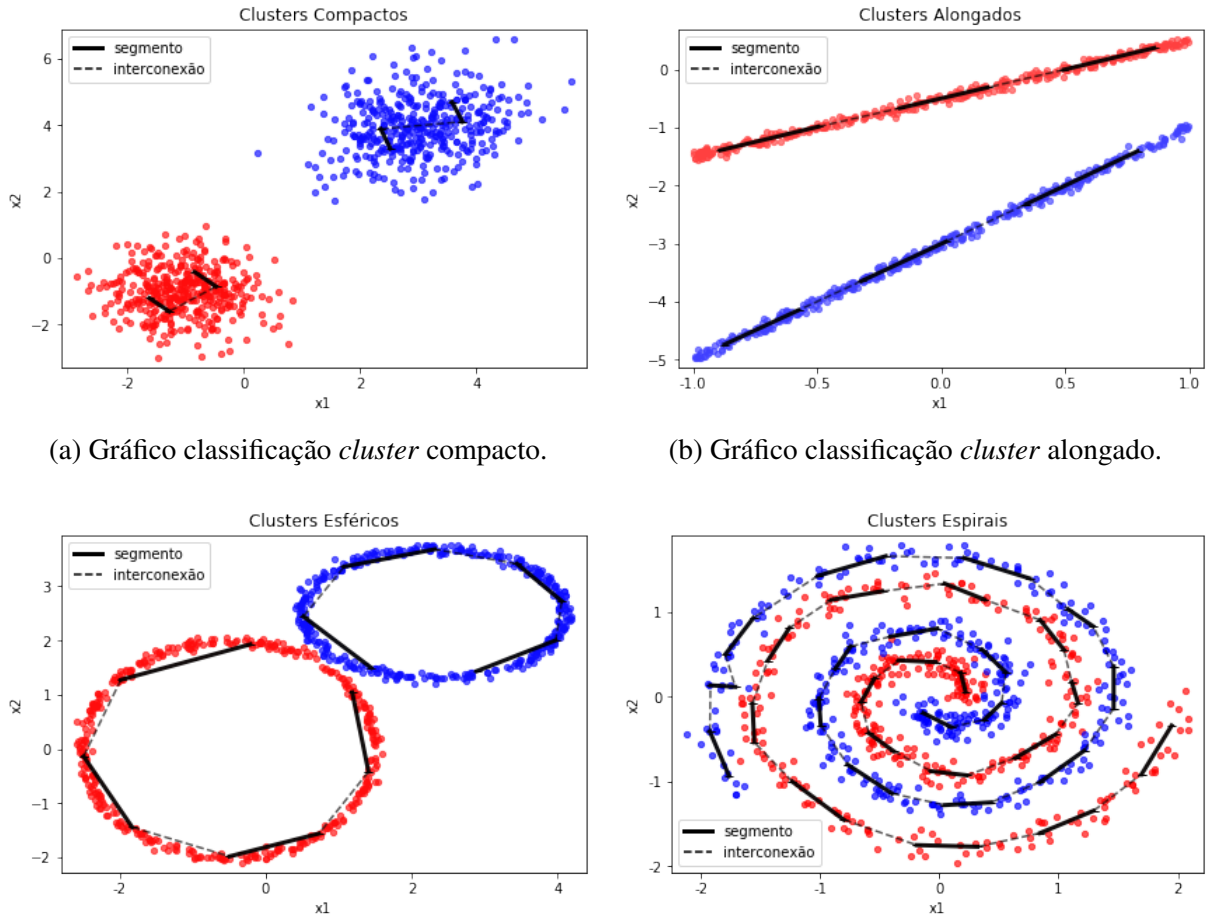
As Figuras 4.5a, 4.5b, 4.5c e 4.5d ilustram graficamente os conjuntos de dados sintéticos (*clusters* compactos, *clusters* alongados, *clusters* esféricos e *clusters* em espiral) considerados nos experimentos quanto a separação entre classes produzida pelo método CPC. Observa-se que, em cada cenário, os padrões geométricos dos dados são capturados de maneira consistente, possibilitando avaliar a capacidade do método em modelar distribuições com diferentes graus de separabilidade e complexidade, desde configurações aproximadamente lineares até formas não lineares mais complexas.

Os resultados mostram que o algoritmo CPC apresenta desempenho competitivo em relação aos algoritmos consolidados na literatura e que, para algumas bases de dados da literatura, como *Breast_Cancer* e *Iris*, os resultados foram superiores aos dos algoritmos RF, NB e ELM. Além disso, para bases de dados com geometrias espirais, esféricas e alongadas, os resultados também foram superiores aos dos algoritmos RF, NB e ELM. Assim, o algoritmo CPC mostra-se uma alternativa promissora.

Por fim, ressalta-se que os tempos computacionais apresentados ao longo dos experimentos devem ser interpretados considerando duas etapas distintas do ciclo de vida do método. A primeira corresponde à fase de projeto, na qual são realizadas a construção do modelo e a calibração dos hiperparâmetros. No caso do CPC, é uma etapa mais custosa devido ao fato de envolver a extração e o ajuste das CPs, além da otimização dos segmentos. A segunda corresponde à fase operacional, em que o modelo já está ajustado e é utilizado apenas para predição, caracterizando um processo mais simples e, portanto, mais rápido. Dessa forma, embora o tempo na fase de projeto seja maior devido ao conjunto de operações envolvidas, o tempo

operacional tende a ser menor, evidenciando a viabilidade do método quando empregado em cenários de uso contínuo após o treinamento.

Figura 4.5 – Resultados da classificação dos conjuntos sintéticos com o método CPC.



(a) Gráfico classificação *cluster* compacto.

(b) Gráfico classificação *cluster* alongado.

(c) Gráfico classificação *cluster* esférico.

(d) Gráfico classificação*cluster* espiral.

Fonte: Do Autor (2026).

4.3 Experimentos na Base de Dados do ENEM 2023

A Tabela 4.18 apresenta a descrição dos métodos utilizados no GWO 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, assim como os valores usados durante a execução para avaliar a metodologia. Em relação ao GWO, o número de gerações (iterações) foi estabelecido em 30. Quanto ao tamanho da população, foram utilizados 50 lobos, o que significa que 50 soluções candidatas são preservadas e atualizadas a cada geração.

Tabela 4.18 – Descrição dos Métodos.

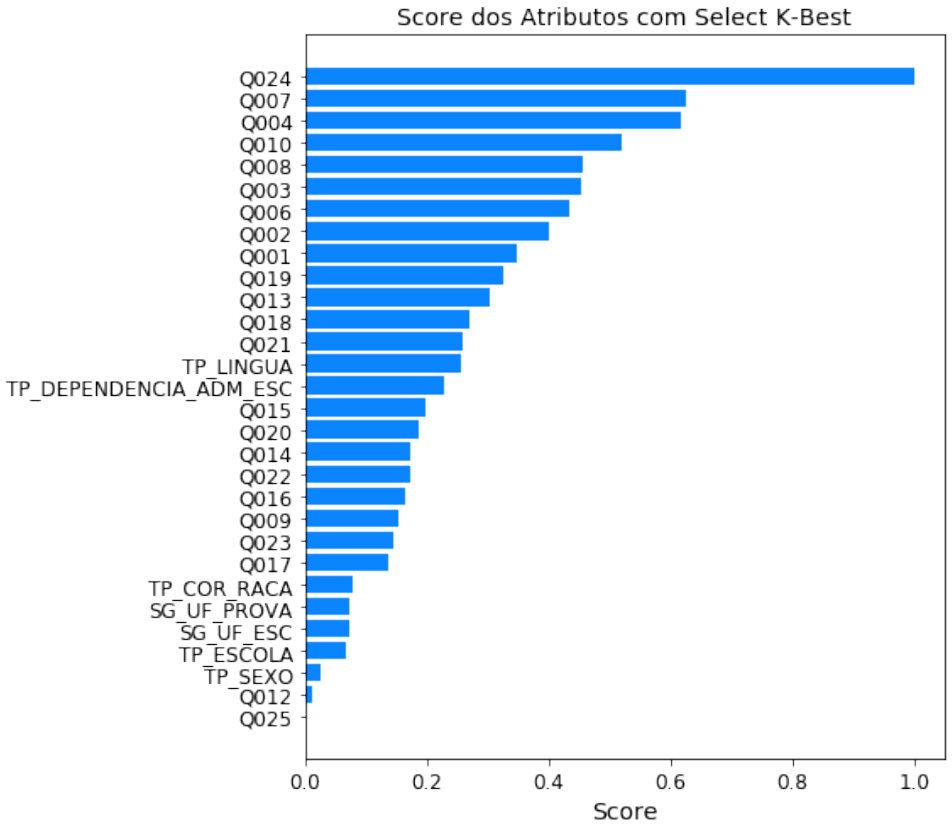
Método	**Parâmetros**	**Variação dos Parâmetros**
CPC	*k_max*	[2, 10]
	f	[0.1, 1.0]
	lambda	[0.05, 1.0]
RF	*n_estimators*	[1, 300]
	max_depth	[1, 50]
	min_samples_split	[2, 50]
	min_samples_leaf	[1, 50]
	criterion	[“*gini*”, “*entropy*”]
NB	*var_smoothing*	$[1e^{-9}, 1e^{-1}]$
ELM	*n_neurons*	[1, 600]
	ufunc	[“*relu*”, “*tanh*”]

Os experimentos com a base do ENEM 2023 são formulados como um problema de classificação supervisionada, no qual o objetivo é atribuir a cada estudante uma classe entre as duas classes de desempenho acadêmico. As classes são definidas a partir do critério de que a Classe 0 e a Classe 1 correspondem a dois grupos distintos de estudantes. A Classe 0 representa estudantes com um desempenho não esperado no exame. Já a Classe 1 indica estudantes com bom desempenho no exame. Assim, dadas as características do estudante no exame, como o contexto escolar e as variáveis socioeconômicas, busca-se classificar seu nível de desempenho.

4.3.1 Experimentos com *Select K-Best*

A Figura 4.6 apresenta o peso dos atributos selecionados pela técnica *Select K-Best*, evidenciando quais variáveis possuem a maior associação estatística com a variável-alvo. Observa-se que a variável Q024 (presença de computador na residência) se destaca de forma expressiva, apresentando o maior peso entre todos os atributos, o que indica sua elevada capacidade discriminativa para a classificação proposta. Em seguida, aparecem variáveis do questionário socioeconômico, como Q007 (presença de empregado(a) doméstico na residência), Q004 (ocupação da mãe), Q006 (renda familiar mensal), Q008 (presença de banheiro na residência) e Q010 (presença de carro na residência), além de itens como Q001/Q002 (escolaridade dos pais), que tem contribuições menores, porém relevantes. De maneira geral, o *ranking* indica que a classificação é influenciada por um indicador direto de desempenho (redação) e, em menor medida, por fatores socioeconômicos e familiares. Por outro lado, características ligadas ao contexto escolar e geográfico (como TP_ESCOLA, SG_UF_ESC e SG_UF_PROVA) têm um peso consideravelmente menor no critério univariado utilizado pelo *Select K-Best*.

Figura 4.6 – Importância das principais variáveis com a técnica *Select K-Best*.



Fonte: Do Autor (2026).

Já a Tabela 4.19 apresenta a análise do desempenho do algoritmo CPC, empregando o método GWO e o método de redução de dimensionalidade *Select K-Best*. Essa avaliação compara o CPC com os métodos RF, NB e ELM, levando em conta as métricas de ACC, F1, PR, RE, Kappa e tempo de execução (em segundos). Os resultados mostram que o CPC obteve resultados competitivos em relação ao NB. O CPC alcançou uma ACC de 0,7706, F1 de 0,7705, PR igual a 0,7707, RE de 0,7706 e um *Kappa* de 0,5412. Em termos de tempo de execução, o método CPC leva aproximadamente 269312 segundos para ser executado, o que é mais do que os métodos RF, ELM e NB. Isso reflete o custo computacional relacionado à otimização de segmentos. Esse tempo de processamento é comum em técnicas que utilizam o cálculo da distância, nas quais a demanda computacional aumenta. Ressalta-se que esse tempo refere-se à fase de projeto (desenvolvimento) do método, pois envolve sua construção. Além disso, é nessa etapa que ocorre a extração das CP, o que contribui para o aumento do custo computacional. Isso reflete o custo computacional relacionado à otimização de segmentos.

Tabela 4.19 – Avaliação do algoritmo baseado em CP no conjunto de treinamento com GWO na base de dados do ENEM com método de redução de dimensionalidade *Select K-Best*.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,7706	0,7705	0,7707	0,7706	0,5412	269312,3307
RF	**0,8195**	**0,8194**	**0,8207**	**0,8195**	**0,6391**	35817,8946
NB	0,7501	0,7498	0,7512	0,7501	0,5002	**192,8923**
ELM	0,8093	0,8092	0,8099	0,8093	0,6186	2707,4491

A Tabela 4.20 apresenta os melhores hiperparâmetros do método CPC, encontrados ao aplicar o GWO. O método é composto por um *k_max* igual a 10, indicando que a curva será representada por até dez segmentos. Um f de parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva igual a 0,1840 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,9010. Tal configuração resultou, no conjunto de teste (Tabela 4.21), em ACC de 0,7603, PR de 0,7612, RE de 0,7603, F1 de 0,7601, *kappa* no valor de 0,5206 e um tempo de execução de aproximadamente 74 segundos. Esse tempo é o tempo de execução na fase operacional, quando o método é utilizado apenas para realizar predições, sendo um processo mais simples e eficiente em comparação à fase de projeto, na qual ocorre a extração das curvas e o ajuste dos parâmetros.

Tabela 4.20 – Melhores parâmetros do algoritmo baseado em CP com GWO com método de redução de dimensionalidade *Select K-Best* no conjunto de treinamento.

Método	**Parâmetros**
CPC	*k_max*: 10, *f*: 0,1840, *lambda*: 0,9010
RF	*n_estimators*: 286, *max_depth*: 35, *min_samples_split*: 16, *min_samples_leaf*: 1, *criterion*: *entropy*
NB	*var_smoothing*: 0,0208
ELM	*n_neurons*: 500, *ufunc*: *relu*

Tabela 4.21 – Avaliação do algoritmo baseado em CP no conjunto de teste com GWO na base de dados do ENEM com método de redução de dimensionalidade *Select K-Best*.

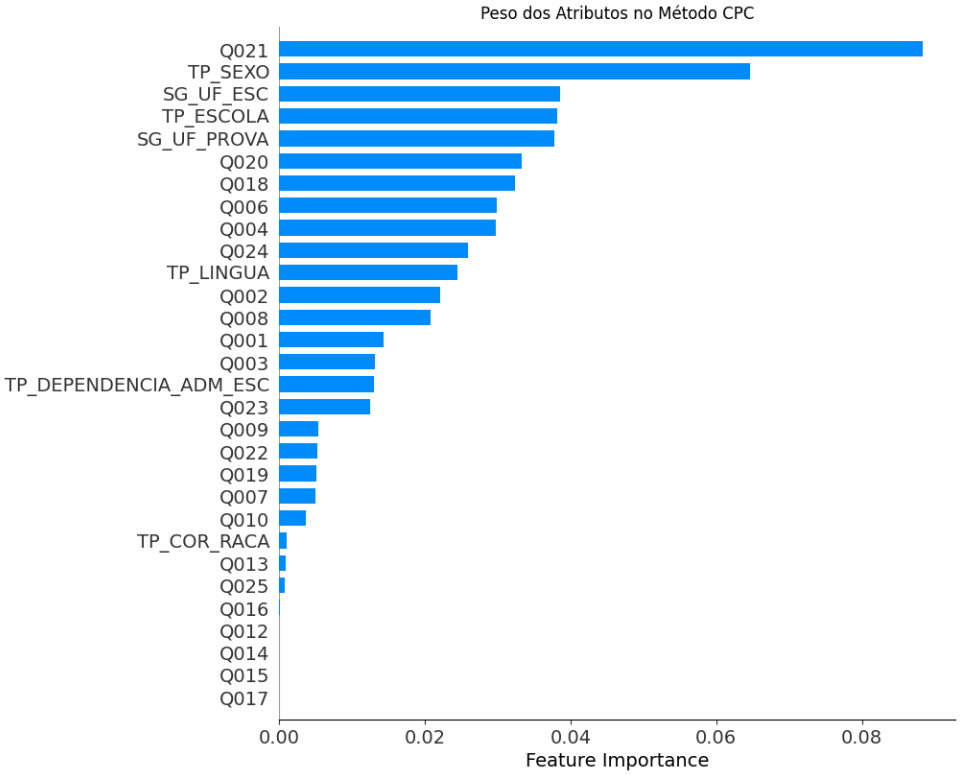
Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,7603	0,7601	0,7612	0,7603	0,5206	73,8209
RF	**0,8188**	**0,8187**	**0,8197**	**0,8188**	**0,6376**	13,0888
NB	0,7468	0,7465	0,7480	0,7468	0,4936	**0,1230**
ELM	0,8076	0,8076	0,8079	0,8076	0,6152	2,1893

As Figuras 4.7a, 4.7b, 4.8a e 4.8b ilustram as melhores características e os respectivos pesos nos métodos de classificação CPC, RF, NB e ELM na base de dados do ENEM 2023. Nota-se que cada método prioriza conjuntos distintos de características, o que reflete a natureza particular de cada algoritmo na identificação de padrões. Para o algoritmo CPC, as características mais relevantes refletem, em sua maioria, o contexto socioeconômico e educacional do estudante, o que é fundamental para a classificação do desempenho no ENEM, uma vez que essas variáveis tendem a distinguir grupos com perfis bastante diferentes de preparação e acesso a oportunidades. Variáveis como TP_SEXO, Q021 (presença de TV por assinatura na residência), TP_ESCOLA (tipo de escola, sendo pública ou privada), SG_UF_PROVA (unidade de aplicação da prova), Q020 (presença de aparelho de DVD na residência), Q006 (renda familiar mensal), Q024 (presença de computador na residência), Q004 (ocupação da Mãe), Q002 (escolaridade da mãe), Q001 (escolaridade do pai), TP_LINGUA (língua estrangeira), Q008 (presença de banheiro na residência) e Q018 (presença de aspirador de pó na residência) representam capital educacional e condições familiares que influenciam a trajetória escolar. Assim, ao utilizar essas características, o CPC consegue aumentar sua capacidade de diferenciar estudantes com diferentes níveis de desempenho.

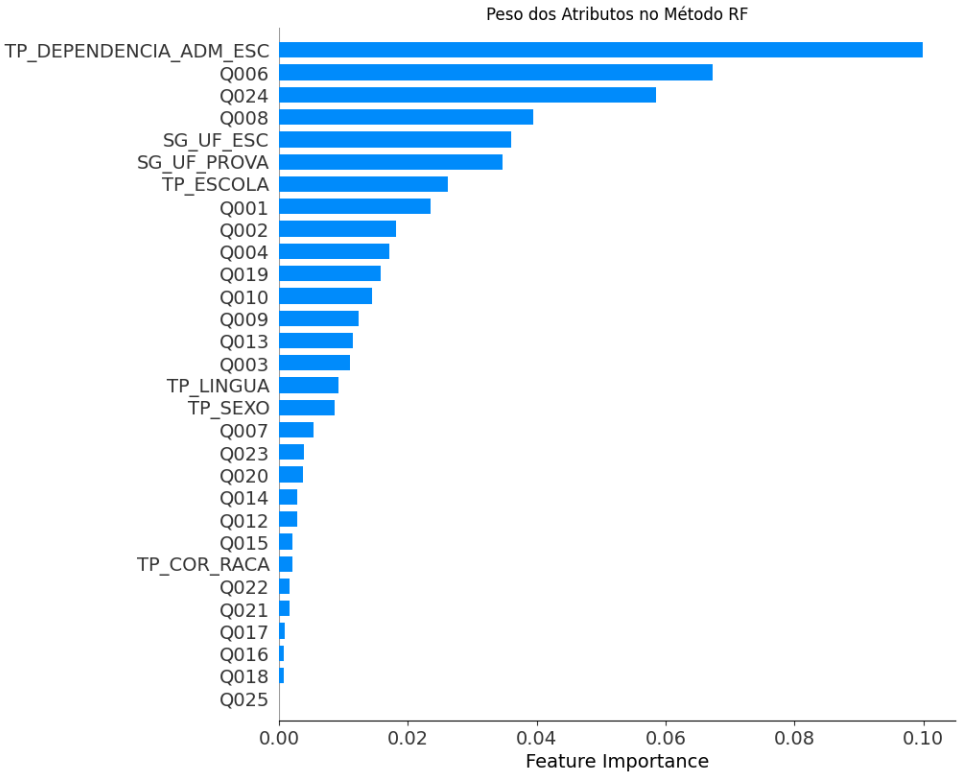
Para o RF, o gráfico de importância dos atributos indica que a variável mais importante no processo de classificação é a TP_DEPENDENCIA_ADM_ESC, evidenciando que a dependência administrativa (escola) do estudante impacta a separação das classes. Em seguida, destacam-se Q006 (renda familiar mensal), além de Q008 (presença de banheiro na residência) e Q024 (presença de computador na residência), sugerindo que fatores socioeconômicos também contribuem significativamente para a decisão do modelo. Os resultados mostram que o RF combina informações de indicadores do contexto escolar e socioeconômico para classificação. Já para o NB, o gráfico de importância evidencia que as variáveis de contexto socioeconômico e infraestrutura domiciliar influenciam a capacidade de separação entre as classes, com destaque para Q024 (presença de computador na residência) e Q006 (renda familiar mensal). Em seguida, aparecem TP_ESCOLA e itens relacionados às condições do domicílio, como Q008 (presença de banheiro na residência), além de variáveis familiares, como Q004 (ocupação da mãe) e Q019/Q025 (televisão em cores e acesso à internet na residência), reforçando a influência do ambiente social e educacional no desempenho.

Por fim, no método ELM, o gráfico de importância indica que TP_ESCOLA é o atributo com maior influência na classificação, sugerindo que o tipo de escola é um dos principais fatores para a separação das classes definidas. Em seguida, destacam-se variáveis relacionadas ao contexto escolar e demográfico, como Q024 (presença de computador na residência) e Q006 (renda familiar mensal), evidenciando a relevância de indicadores socioeconômicos e de acesso a recursos. Também aparecem com peso intermediário TP_SEXO, TP_DEPENDENCIA_ADM_ESC e variáveis regionais (SG_UF_ESC e SG_UF_PROVA), bem como atributos familiares, como Q004 (ocupação da mãe). Os demais itens do questionário possuem contribuições menores, mas ainda complementam o método. Os resultados mostram que o ELM combina informações do ambiente escolar e socioeconômico para a classificação.

Figura 4.7 – Características mais importantes da Base de Dados do ENEM 2023 nos métodos CPC e RF.



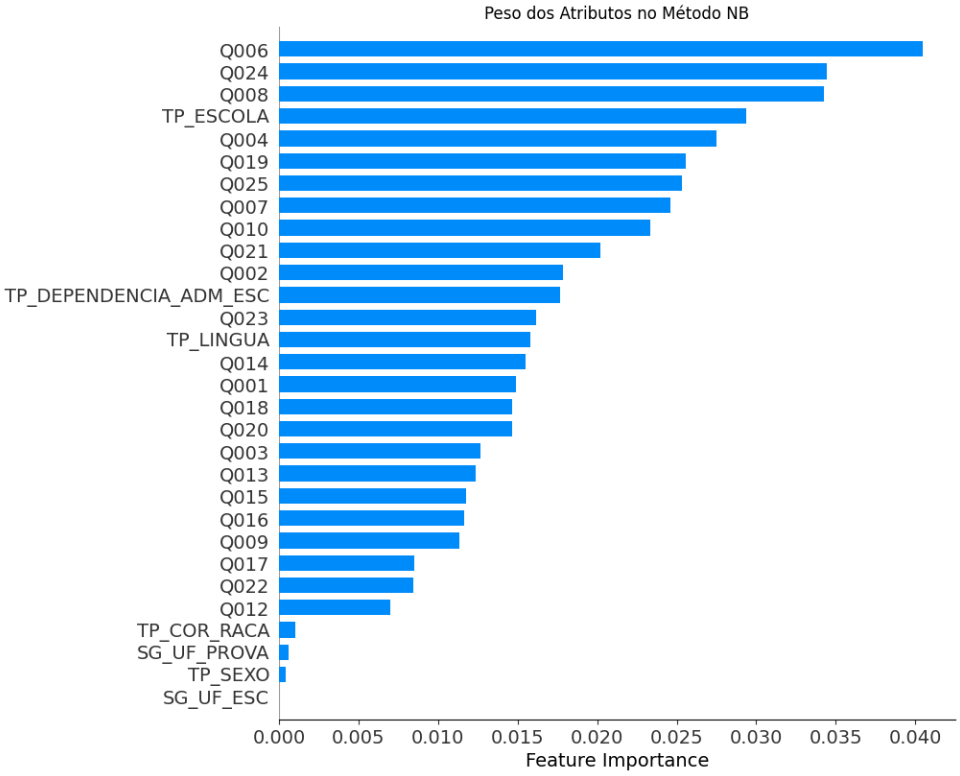
(a) Método CPC.



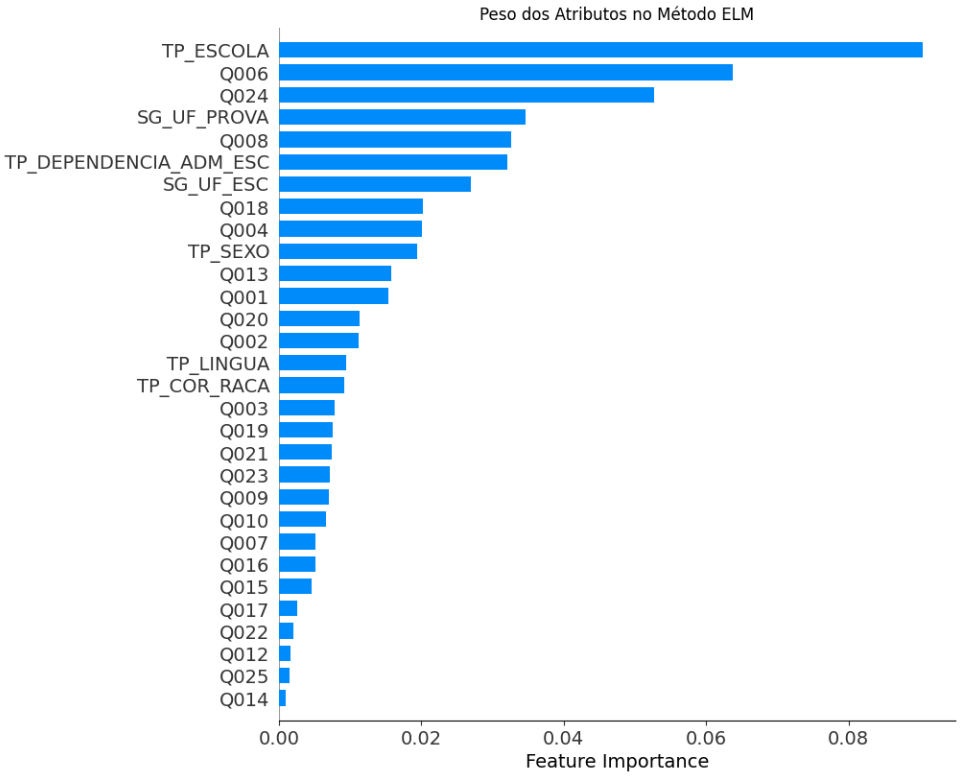
(b) Método RF.

Fonte: Do Autor (2026).

Figura 4.8 – Características mais importantes da Base de Dados do ENEM 2023 nos métodos NB e ELM.



(a) Método NB.

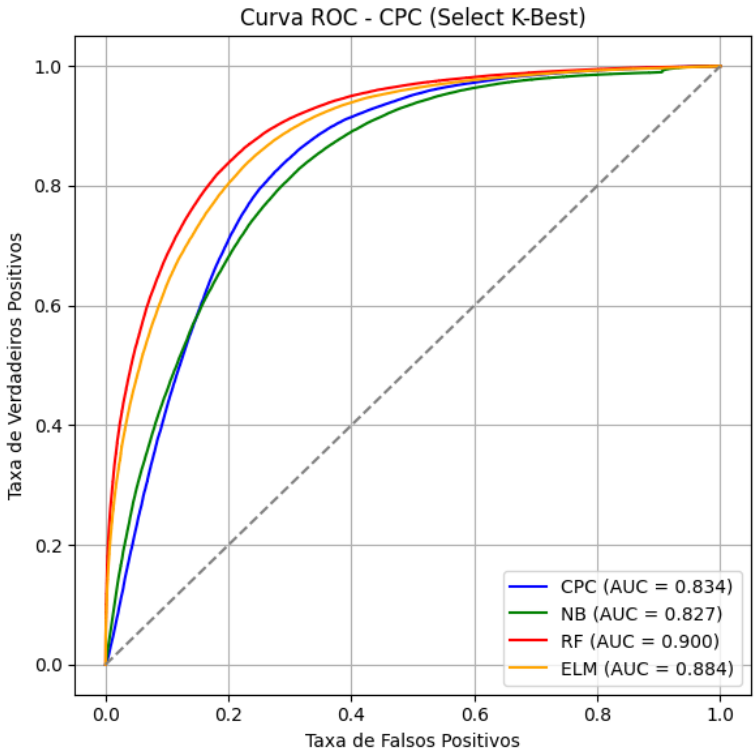


(b) Método ELM.

Fonte: Do Autor (2026).

Já a Figura 4.9 apresenta a curva ROC obtida pelo algoritmo CPC, aplicando a redução de dimensionalidade com o método *Select K-Best* (os melhores atributos). Observa-se que a curva se mantém consideravelmente acima da linha diagonal (classificador aleatório), indicando uma boa capacidade de distinção entre as classes ao longo de diferentes limiares de decisão. O valor de AUC igual a 0,834 evidencia um desempenho consistente, indicando que o modelo pode alcançar altas taxas de verdadeiros positivos com aumentos controlados na taxa de falsos positivos, o que reforça sua eficácia para a tarefa de classificação proposta.

Figura 4.9 – Curva ROC do método CPC com *Select K-Best*.



Fonte: Do Autor (2026).

A Tabela 4.22 apresenta o número de segmentos e o comprimento em unidades arbitrárias das curvas principais estimadas para cada classe no ENEM. A quantidade de segmentos está ligada à complexidade geométrica necessária para representar a distribuição dos dados de cada classe no espaço de atributos, enquanto o comprimento da curva reflete o grau de dispersão e diversidade dos perfis dos estudantes dentro daquela classe. Nota-se que as Classes 0 e 1 foram representadas pelo mesmo número de segmentos, o que indica uma complexidade semelhante sob a perspectiva do método de CP. No entanto, a Classe 1 apresentou um comprimento um pouco maior em relação à Classe 0, indicando que os alunos da Classe 1 estão, em média, mais dispersos no espaço de atributos, ou seja, apresentam uma maior diversidade de características.

Por outro lado, a Classe 0 apresenta um comprimento de curva menor, o que sugere um agrupamento mais homogêneo, no qual os perfis representados pelo conjunto de variáveis utilizadas apresentam menor variabilidade.

Tabela 4.22 – Número de segmentos e comprimento (u.a.) das curvas principais (abordagem com *Select K-Best*).

Classes	**Número de Segmentos**	**Comprimento (u.a.)**
0	10	11,3967
1	10	11,8474

4.3.2 Experimentos com PCA

A Tabela 4.23 apresenta os resultados do desempenho do algoritmo CPC, utilizando o método de redução de dimensionalidade PCA (com as três primeiras componentes principais, as quais representam 90% da variabilidade dos dados). Os resultados comparam o CPC com os métodos RF, NB e ELM. Os resultados mostram que o CPC obteve resultados competitivos em relação aos métodos comparados. O CPC alcançou uma ACC de 0,7618, F1 de 0,7612, PR igual a 0,7645, RE de 0,7618 e um *Kappa* de 0,5235. Em termos de tempo de execução, o método CPC levou aproximadamente 13873 segundos para ser executado (fase de projeto), o que foi menor do que o RF. A redução de dimensionalidade via PCA diminui o custo das operações do CPC, pois o método opera em um espaço de menor dimensionalidade e classifica em um espaço com menos variáveis. Por outro lado, o RF envolve o processamento de um conjunto de múltiplas árvores de decisão, exigindo a avaliação de diversos nós e regras em cada árvore para cada amostra, o que aumenta o custo computacional, principalmente quando há um grande volume de dados. Portanto, o CPC se mostrou uma boa alternativa, demonstrando-se computacionalmente mais eficiente do que o RF neste cenário.

Tabela 4.23 – Avaliação do algoritmo baseado em CP no conjunto de treinamento com GWO na base de dados do ENEM com método de redução de dimensionalidade PCA.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	0,7618	0,7612	0,7645	0,7618	0,5235	13872,6246
RF	**0,7844**	**0,7839**	**0,7868**	**0,7844**	**0,5688**	14539,0796
NB	0,7566	0,7564	0,7576	0,7566	0,5132	**76,0462**
ELM	0,7836	0,7834	0,7849	0,7836	0,5673	2670,2724

A Tabela 4.24 apresenta os melhores hiperparâmetros do método CPC com a técnica de redução de dimensionalidade PCA, encontrados ao aplicar o GWO. O método é composto por um *k_max* igual a 2, indicando 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 0,6610 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,1008. Tal configuração resultou, no conjunto de teste (Tabela 4.25), em ACC de 0,7485, PR de 0,7539, RE de 0,7485, F1 de 0,7472, *kappa* no valor de 0,4971 e um tempo de execução de aproximadamente 3 segundos (fase operacional). O tempo tende a ser menor no conjunto de teste, pois o modelo já está definido e não há repetição de ajustes. Além disso, percebe-se que o CPC é computacionalmente mais eficiente do que o RF neste cenário.

Tabela 4.24 – Melhores parâmetros do algoritmo baseado em CP com GWO com método de redução de dimensionalidade PCA no conjunto de treinamento.

Método	Parâmetros
CPC	*k_max*: 2, f: 0,6610, *lambda*: 0,1008
RF	*n_estimators*: 25, *max_depth*: 10, *min_samples_split*: 24, *min_samples_leaf*: 50, *criterion*: *gini*
NB	*var_smoothing*: 0,0021
ELM	*n_neurons*: 599, *ufunc*: *tanh*

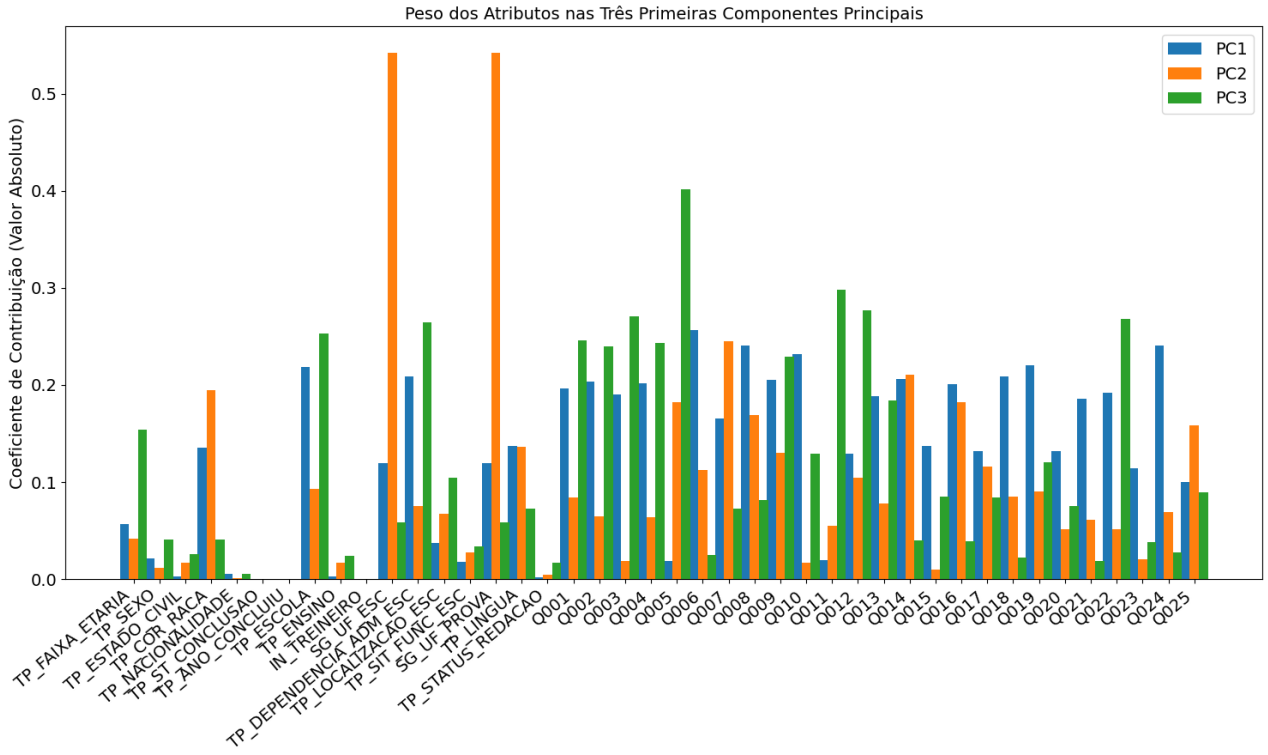
Tabela 4.25 – Avaliação do algoritmo baseado em CP no conjunto de teste com GWO na base de dados do ENEM com método de redução de dimensionalidade PCA.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	0,7485	0,7472	0,7539	0,7485	0,4971	2,8597
RF	**0,7827**	**0,7823**	**0,7843**	**0,7827**	**0,5653**	5,6528
NB	0,7536	0,7533	0,7549	0,7536	0,5073	**0,0240**
ELM	0,7819	0,7817	0,7827	0,7819	0,5638	2,0842

Já a Figura 4.10 apresenta um gráfico com os pesos (em valor absoluto) das principais características da base de dados do ENEM 2023 nas três primeiras componentes principais do PCA (PC1, PC2 e PC3), que explicam a maior parte da variabilidade dos dados. Observa-se que a PC2 é fortemente influenciada por variáveis relacionadas ao contexto administrativo/-geográfico da escola (com picos de contribuição bem superiores aos demais), indicando que essa componente tende a representar diferenças estruturais associadas ao ambiente escolar. Já a PC1 distribui sua contribuição de forma mais equilibrada entre diversos itens do questionário e características do participante, como Q006 (renda familiar mensal), Q008 (presença de banheiro na residência), Q010 (presença de carro na residência), Q024 (presença de computador na residência), Q018 (presença de aspirador de pó na residência), Q019 (presença de televisão em cores na residência) e TP_ESCOLA (Tipo de escola no ensino médio (pública/particular)). Desta forma, sugere uma dimensão mais "global" do perfil do estudante.

Por sua vez, a PC3 destaca principalmente variáveis relacionadas aos pais e às condições socioeconômicas do estudante. Entre as variáveis, destacam-se Q002 (escolaridade da mãe), Q003 (ocupação do pai), Q005 (quantidade de moradores na residência), Q009 (quantidade de quartos na residência), Q011 (presença de motocicleta na residência), Q012 (presença de geladeira na residência), Q022 (presença de telefone celular na residência) e TP_ESCOLA (tipo de escola no ensino médio (pública/particular)). O gráfico evidência que uma parte importante da variância dos dados está associada a indicadores de acesso a recursos e ao perfil familiar.

Figura 4.10 – Importância dos Atributos nas Componentes Principais.

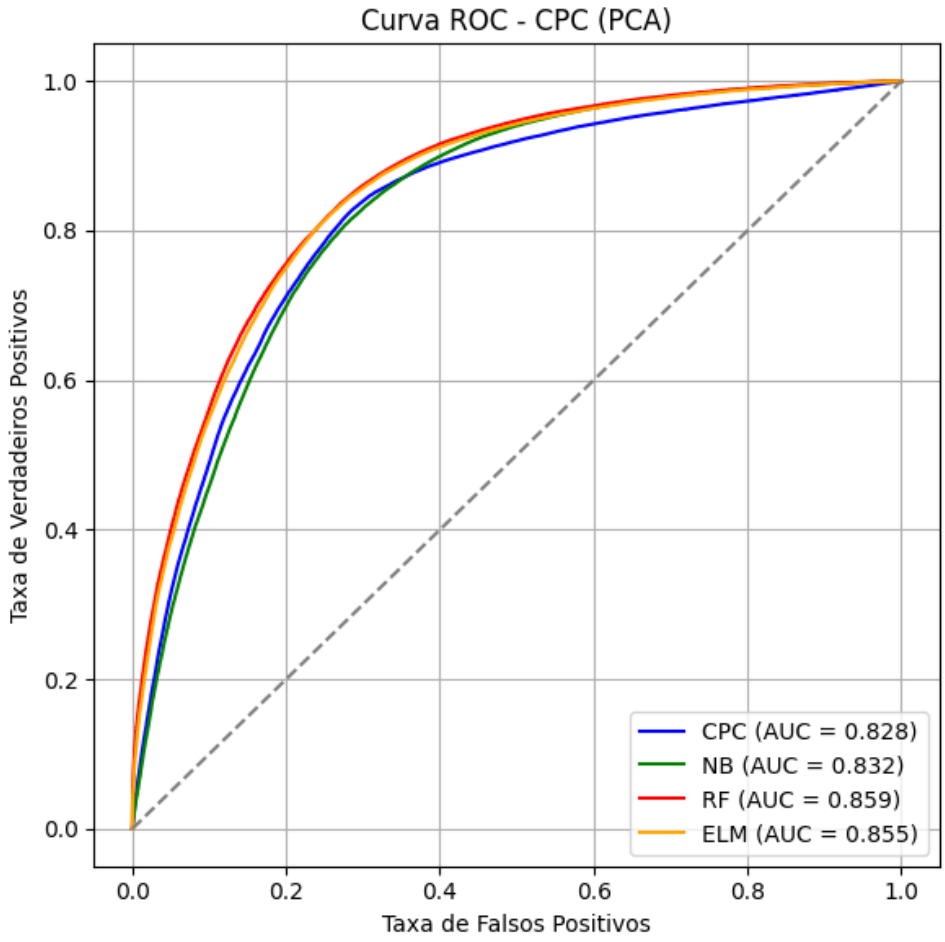


Fonte: Do Autor (2026).

Além disso, a Figura 4.11 apresenta a curva ROC obtida pelo algoritmo CPC, utilizando a técnica de redução de dimensionalidade PCA. Observa-se que a curva se mantém consideravelmente acima da linha diagonal (classificador aleatório), indicando a capacidade de distinção entre as classes ao longo de diferentes limiares de decisão. O valor de AUC igual a 0,828 evidencia um desempenho consistente, refletindo uma boa generalização.

A Tabela 4.26 apresenta o número de segmentos e o comprimento em unidades arbitrárias das curvas principais estimadas para cada classe no ENEM. Nota-se que as Classes 0 e 1 foram representadas pelo mesmo número de segmentos, o que indica uma complexidade semelhante sob a perspectiva do método de CP. Todavia, a Classe 1 apresentou um comprimento um pouco maior em relação à Classe 0, indicando que os alunos da Classe 1 estão, em média, mais dispersos no espaço de atributos, ou seja, apresentam uma maior diversidade de características. Entretanto, a Classe 0 apresenta um comprimento de curva menor, o que sugere um agrupamento mais homogêneo, no qual os perfis representados pelo conjunto de variáveis utilizado apresentam menor variabilidade.

Figura 4.11 – Curva ROC do método CPC com PCA.



Fonte: Do Autor (2026).

Tabela 4.26 – Número de segmentos e comprimento (u.a.) das curvas principais (abordagem com PCA).

Classes	Número de Segmentos	Comprimento (u.a.)
0	2	0,5773
1	2	0,5888

4.3.3 Experimentos com t-SNE

Já a Tabela 4.27 apresenta os resultados de desempenho do algoritmo CPC em comparação com os métodos RF, NB e ELM, considerando, em todos os casos, a aplicação da técnica de redução de dimensionalidade t-SNE (espaço bidimensional, utilizando as componentes t-SNE1 e t-SNE2 como representação final). Os resultados mostram que o CPC obteve desempenho competitivo em comparação com os métodos analisados. O CPC alcançou uma ACC de

0,7357, F1 de 0,7356, PR igual a 0,7360, RE de 0,7357 e um *Kappa* de 0,4713. Em termos de tempo de execução, o método CPC levou aproximadamente 12086 segundos (fase de projeto) para ser executado, o que foi bem menor do que o RF, que obteve um tempo de execução de aproximadamente 82362 segundos. A redução de dimensionalidade via t-SNE também diminui o custo das operações do CPC, pois o método opera em um espaço de menor dimensionalidade e classifica em um espaço com menos variáveis, ou seja, realiza cálculos de distância sobre uma representação de menor dimensão. Por outro lado, o RF envolve o processamento de um conjunto de múltiplas árvores de decisão, exigindo a avaliação de diversos nós e regras em cada árvore para cada amostra, o que aumenta o custo computacional, principalmente quando há um grande volume de dados. Assim, o CPC mostra-se uma alternativa mais eficiente do ponto de vista computacional neste cenário do que o RF.

Tabela 4.27 – Avaliação do algoritmo baseado em CP no conjunto de treinamento com GWO na base de dados do ENEM com método de redução de dimensionalidade t-SNE.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,7357	0,7356	0,7360	0,7357	0,4713	12085,9725
RF	**0,7748**	**0,7746**	**0.7760**	**0,7748**	**0,5496**	82361,6653
NB	0,7318	0,7317	0,7319	0,7318	0,4635	**71,4212**
ELM	0,7384	0,7384	0,7386	0,7384	0,4768	412,5253

A Tabela 4.28 apresenta os melhores hiperparâmetros do método CPC, com a técnica de redução de dimensionalidade t-SNE, encontrados ao aplicar o GWO. O método é composto por um *k_max* igual a 2, indicando 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 0,3023 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,3364. Tal configuração resultou, no conjunto de teste (Tabela 4.29), em ACC de 0,7310, PR de 0,7314, RE de 0,7310, F1 de 0,7309, *kappa* no valor de 0,4621 e um tempo de execução de aproximadamente 2,6 segundos (fase operacional). O tempo do CPC tende a ser menor no conjunto de teste, pois o modelo já está definido e não há repetição de ajustes. Além disso, percebe-se que o CPC é computacionalmente mais eficiente do que o RF no cenário de aplicação de redução de dimensionalidade com o t-SNE.

Tabela 4.28 – Melhores parâmetros do algoritmo baseado em CP com GWO com método de redução de dimensionalidade t-SNE no conjunto de treinamento.

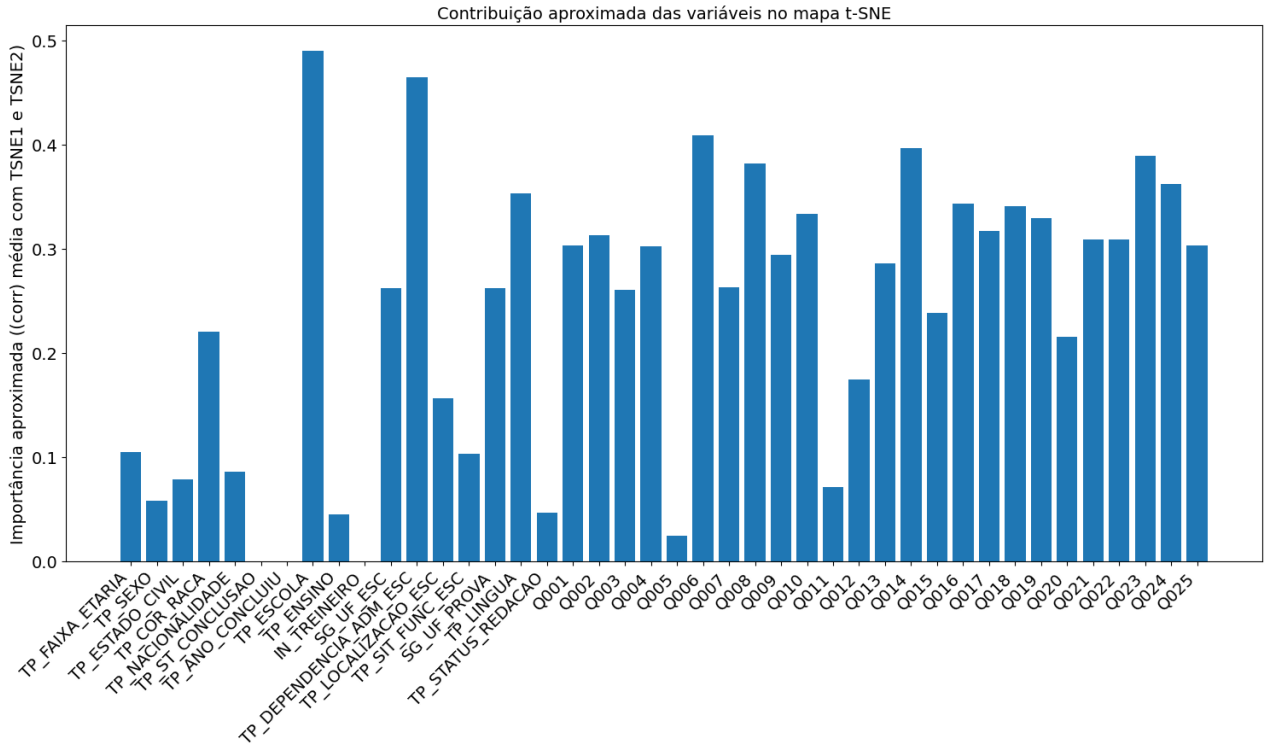
Método	**Parâmetros**
CPC	*k_max*: 2, *f*: 0,3023, *lambda*: 0,3364
RF	*n_estimators*: 281, *max_depth*: 38, *min_samples_split*: 23, *min_samples_leaf*: 12, *criterion*: *entropy*
NB	*var_smoothing*: 0,0151
ELM	*n_neurons*: 25, *ufunc*: *relu*

Tabela 4.29 – Avaliação do algoritmo baseado em CP no conjunto de teste com GWO na base de dados do ENEM com método de redução de dimensionalidade t-SNE.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,7310	0,7309	0,7314	0,7310	0,4621	2,6018
RF	**0,7714**	**0,7712**	**0,7725**	**0,7714**	**0,5428**	21,1510
NB	0,7291	0,7291	0,7292	0,7291	0,4582	**0,0190**
ELM	0,7316	0,7316	0,7317	0,7316	0,4632	0,0435

A Figura 4.12 apresenta um gráfico com uma estimativa da contribuição das variáveis originais para a projeção gerada pela técnica t-SNE, calculada a partir da correlação média (em módulo) de cada variável com as dimensões t-SNE1 e t-SNE2. Nota-se que as maiores contribuições concentram-se em variáveis associadas ao contexto escolar, administrativo e familiar, como TP_ESCOLA (tipo de escola no ensino médio (pública/particular)), TP_LINGUA (língua estrangeira), TP_DEPENDENCIA_ADM_ESC (dependência administrativa da escola), Q023 (presença de telefone fixo na residência), Q014 (presença de máquina de lavar roupa na residência), Q008 (presença de banheiro na residência), Q016 (presença de forno micro-ondas na residência), Q006 (renda familiar mensal), Q010 (presença de carro na residência) e Q024 (presença de computador na residência). Isso indica que esses fatores são os que mais influenciam a organização dos pontos no mapa t-SNE. Assim, o gráfico ajuda a interpretar quais atributos têm maior relação com a estrutura preservada pelo t-SNE, ainda que se trate de uma medida aproximada, uma vez que o t-SNE é um método não linear e não fornece pesos diretos como no PCA.

Figura 4.12 – Contribuição das Variáveis no mapa t-SNE1 e t-SNE2.

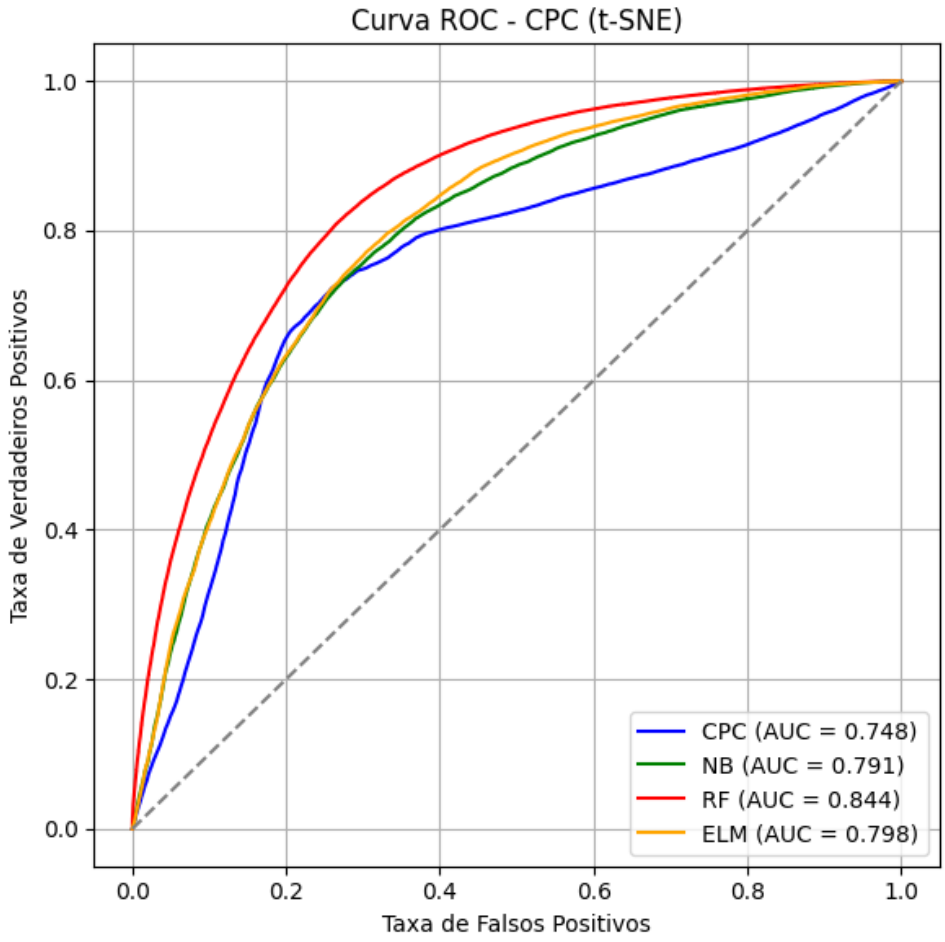


Fonte: Do Autor (2026).

Já a Figura 4.13 apresenta a curva ROC obtida pelo algoritmo CPC com a abordagem de redução de dimensionalidade t-SNE. Nota-se que a curva se mantém consideravelmente acima da linha diagonal (classificador aleatório), indicando a capacidade de distinção entre as classes ao longo de diferentes limiares de decisão. O valor de AUC igual a 0,748 evidencia um desempenho consistente, refletindo uma boa generalização.

Na Tabela 4.30 são apresentados o número de segmentos e o comprimento em unidades arbitrárias das curvas principais estimadas para cada uma das classes de estudantes no ENEM. Nota-se que as Classes 0 e 1 também foram representadas pelo mesmo número de segmentos, o que indica uma complexidade semelhante sob a perspectiva do método de CP. Todavia, a Classe 0 apresentou um comprimento um pouco maior em comparação com a Classe 1, indicando que os alunos da Classe 0 estão, em média, mais dispersos no espaço de atributos e apresentam uma maior diversidade de características. Entretanto, a Classe 1 apresenta um comprimento de curva menor, o que sugere um agrupamento mais homogêneo.

Figura 4.13 – Curva ROC do método CPC com t-SNE.



Fonte: Do Autor (2026).

Tabela 4.30 – Número de segmentos e comprimento (u.a.) das curvas principais (abordagem com t-SNE).

Classes	Número de Segmentos	Comprimento (u.a.)
0	2	0,6522
1	2	0,5775

4.3.4 Experimentos com a Distância de *Gower*

Com o objetivo de investigar alternativas para melhorar o desempenho do algoritmo CPC, foram realizados experimentos utilizando a distância de *Gower*, por se tratar de uma métrica adequada a bases de dados mistas que combinam atributos numéricos e categóricos (Gower, 1971). Além disso, nos experimentos, foi considerada a técnica *Select K-Best* como redução de dimensionalidade, utilizando os mesmos atributos considerados na subseção 4.3.1.

A distância de *Gower* tende a representar de forma mais apropriada a dissimilaridade entre instâncias nesse tipo de base de dados, o que pode contribuir para uma melhor separação entre as classes. Nesse contexto, buscou-se avaliar se o uso dessa métrica poderia gerar uma representação mais adequada das classes e, consequentemente, melhorar os resultados do método CPC.

A distância de *Gower* é uma medida de distância utilizada para calcular a distância entre duas entidades cujos atributos possuem uma combinação de valores categóricos e numéricos (Gower, 1971). Para isso, a distância de *Gower* atribui uma pontuação entre dois pontos de dados, realizando cálculos de distância distintos para atributos numéricos e categóricos e, posteriormente, calculando uma média ponderada das similaridades entre esses atributos. A pontuação é 1 quando os valores são iguais, caso sejam diferentes, a pontuação é zero. Todavia, essa métrica consome muita memória e requer manter na memória uma matriz de tamanho $(n \times n)$ (Gower, 1971).

Neste experimento, para avaliar a técnica da distância de *Gower* no algoritmo CPC, utilizou-se um subconjunto de 8000 amostras. Essa restrição é resultado das limitações de recursos computacionais, uma vez que o cálculo da distância de *Gower* exige a criação e o armazenamento em memória de uma matriz de dissimilaridade entre pares de amostras, cuja dimensão aumenta quadraticamente com o número de instâncias $(n \times n)$. Como resultado, conforme n aumenta, o uso de memória cresce, tornando inviável a execução com todo o conjunto de dados disponível no ambiente computacional utilizado para a realização dos experimentos. Dessa forma, o valor de 8000 amostras representou o limite suportado pela máquina, possibilitando a execução dos testes sem afetar a estabilidade do processamento.

Na Tabela 4.31, apresenta-se o resultado do desempenho do algoritmo CPC, utilizando a distância de *Gower*. Os resultados melhoram com a utilização da distância de *Gower* como método para o cálculo da distância. O CPC alcançou uma ACC de 0,8342, F1 de 0,8342, PR igual a 0,8344, RE de 0,8342 e um *Kappa* de 0,6684. Em termos de tempo de execução, o método CPC levou aproximadamente 2457 segundos para ser executado. Esse tempo refere-se à fase de projeto do método, pois envolve sua construção. Além disso, é nessa etapa que ocorre a extração das CPs, o que contribui para o aumento do custo computacional. Isso reflete o custo computacional relacionado à otimização de segmentos.

Tabela 4.31 – Avaliação do algoritmo baseado em CP no conjunto de treinamento com GWO na base de dados do ENEM com a distância de *gower*.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,8342	0,8342	0,8344	0,8342	0,6684	2457,3068

A Tabela 4.32 apresenta os melhores hiperparâmetros do método CPC, encontrados ao aplicar o GWO. O método é composto por um *k_max* igual a 3, indicando que a curva será representada por até três segmentos. Um f de parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva igual a 1,0288 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,9865. Tal configuração resultou, no conjunto de teste (Tabela 4.33), em ACC de 0,7813, PR de 0,7924, RE de 0,7813, F1 de 0,7792, *kappa* no valor de 0,5625 e um tempo de execução de aproximadamente 0,3878 segundos (fase operacional). O tempo do CPC tende a ser menor no conjunto de teste, pois o modelo já está definido e não há repetição de ajustes.

Tabela 4.32 – Melhores parâmetros do algoritmo baseado em CP com GWO com a distância de *gower* no conjunto de treinamento.

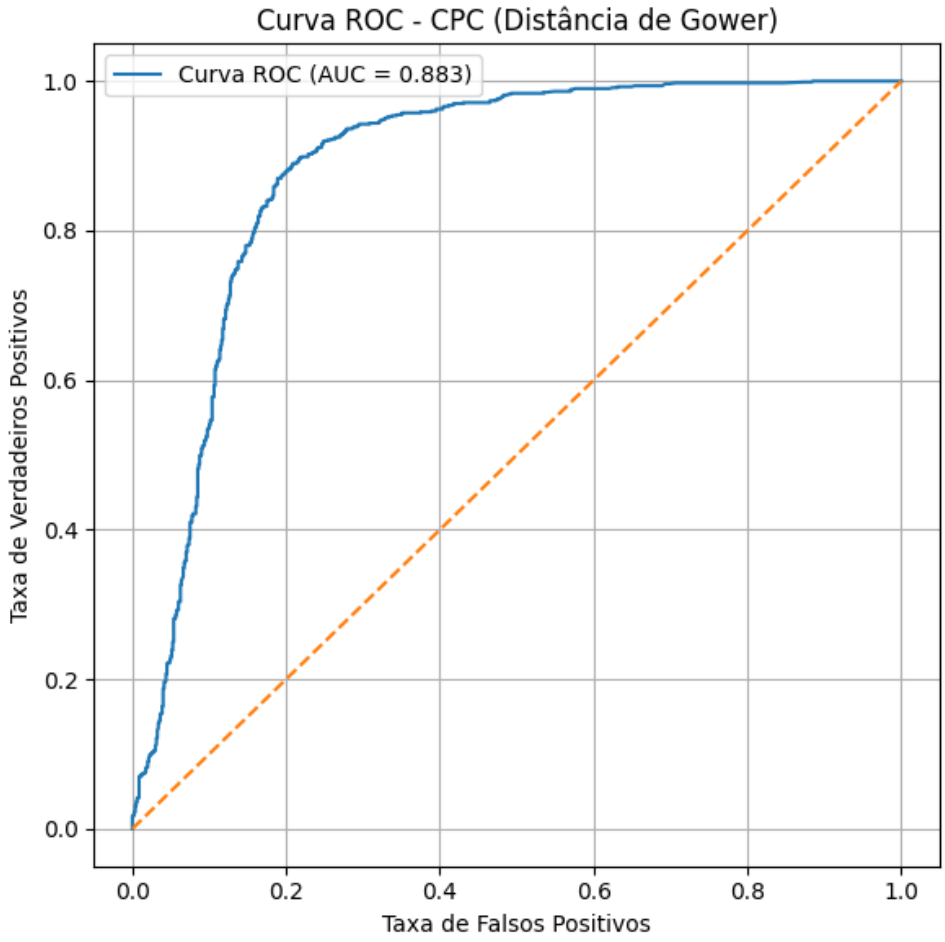
Parâmetros
k_max: 3, f: 1,0288, *lambda*: 0,9865

Tabela 4.33 – Avaliação do algoritmo baseado em CP no conjunto de teste com GWO na base de dados do ENEM com a distância de *gower*.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,7813	0,7792	0,7924	0,7813	0,5625	0,3870

Na Figura 4.14 é apresentada a curva ROC obtida pelo algoritmo CPC utilizando a abordagem baseada na distância de *Gower*. Observa-se que a curva permanece significativamente acima da linha diagonal (classificador aleatório), indicando uma boa capacidade de discriminação entre as classes para diferentes limiares de decisão. O valor de AUC igual a 0,883 evidencia um desempenho satisfatório no processo de classificação.

Figura 4.14 – Curva ROC do método CPC com a abordagem de distância *gower*.



Fonte: Do Autor (2026).

A Tabela 4.34 apresenta o número de segmentos e o comprimento em unidades arbitrárias das curvas principais estimadas para cada uma das classes de estudantes no ENEM. As Classes 0 e 1 também foram representadas pelo mesmo número de segmentos, o que indica uma complexidade semelhante sob a perspectiva do método de CP. Entretanto, a Classe 1 apresentou um comprimento um pouco maior em comparação com a Classe 0, indicando que os alunos da Classe 1 estão, em média, mais dispersos no espaço de atributos e apresentam uma maior diversidade de características. Todavia, a Classe 0 apresenta um comprimento de curva menor, o que sugere um agrupamento mais homogêneo.

Tabela 4.34 – Número de segmentos e comprimento (u.a.) das curvas principais (com a distância de *gower* como abordagem).

Classes	Número de Segmentos	Comprimento (u.a.)
0	3	5,1143
1	3	5,6120

Além disso, foram realizados experimentos utilizando a métrica de distância euclidiana para testar o método CPC nessa mesma base de dados, composta por 8000 amostras, a fim de verificar se a métrica apresenta resultados comparavéis.

Na Tabela 4.35, apresenta-se o resultado do desempenho do algoritmo CPC, utilizando a distância euclidiana. O CPC alcançou uma ACC de 0,8327, F1 de 0,8326, PR igual a 0,8333, RE de 0,8327 e um *Kappa* de 0,6653. Em termos de tempo de execução, o método CPC levou aproximadamente 4729 segundos para ser executado. Esse tempo refere-se à fase de projeto do método, pois envolve sua construção. Além disso, é nessa etapa que ocorre a extração das CPs, o que contribui para o aumento do custo computacional.

Tabela 4.35 – Avaliação do algoritmo baseado em CP no conjunto de treinamento com GWO na base de dados do ENEM com a distância euclidiana.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC	0,8327	0,8326	0,8333	0,8327	0,6653	4729,6948

Na Tabela 4.36 é apresentado os melhores hiperparâmetros do método CPC, encontrados ao aplicar o GWO. O método é composto por um *k_max* igual a 5, indicando que a curva será representada por até cinco segmentos. Um f de parâmetro de suavização ou fração de vizinhança utilizada durante a estimativa local da curva igual a 0,3220 e um *lambda* responsável por controlar a penalização da complexidade da curva igual a 0,9248. Tal configuração resultou, no conjunto de teste (Tabela 4.37), em ACC de 0,8363, PR de 0,8363, RE de 0,8363, F1 de 0,8363, *kappa* no valor de 0,6725 e um tempo de execução de aproximadamente 0,7660 segundos (fase operacional). O tempo do CPC tende a ser menor no conjunto de teste, pois o modelo já está definido e não há repetição de ajustes.

Tabela 4.36 – Melhores parâmetros do algoritmo baseado em CP com GWO com a distância euclidiana no conjunto de treinamento.

Parâmetros
k_max: 5, f: 0,3220, *lambda*: 0,9248

Tabela 4.37 – Avaliação do algoritmo baseado em CP no conjunto de teste com GWO na base de dados do ENEM com a distância euclidiana.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC	0,8363	0,8363	0,8363	0,8363	0,6725	0,7660

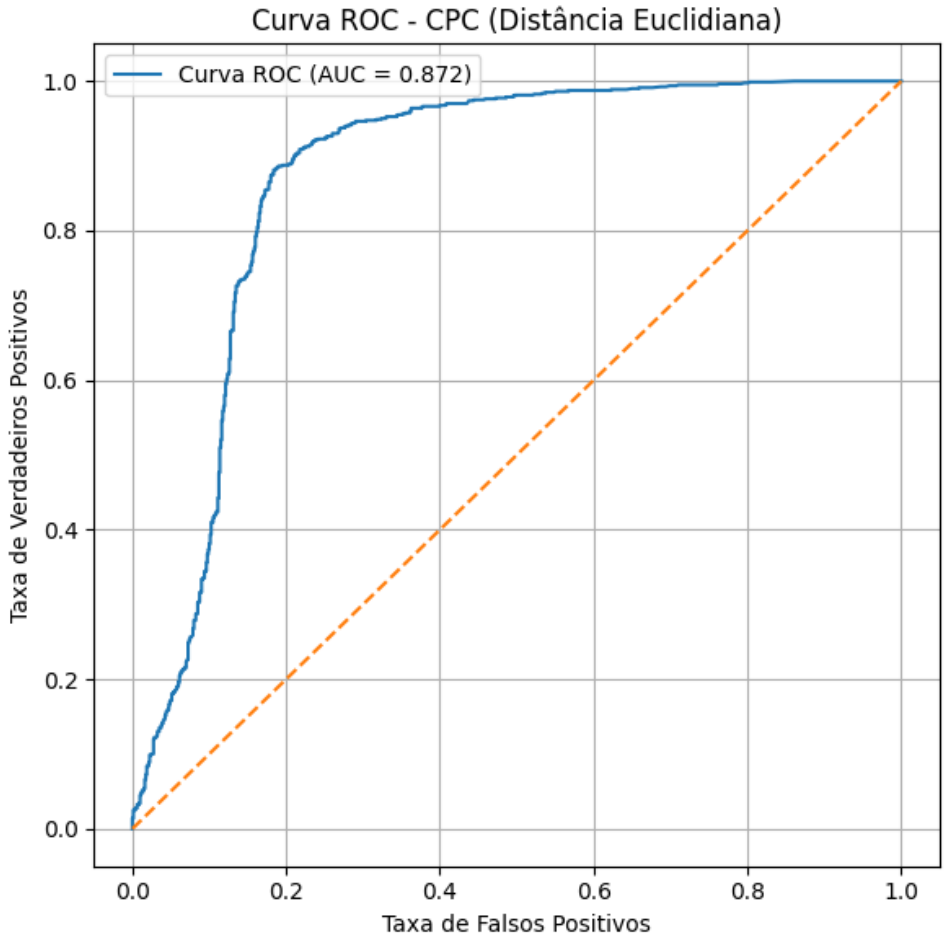
A Tabela 4.38 apresenta o número de segmentos e o comprimento em unidades arbitrárias das curvas principais estimadas para cada uma das classes de estudantes no ENEM. As Classes 0 e 1 também foram representadas pelo mesmo número de segmentos, o que indica uma complexidade semelhante sob a perspectiva do método de CP. Entretanto, a Classe 1 apresentou um comprimento um pouco maior em comparação com a Classe 0, indicando que os alunos da Classe 1 estão, em média, mais dispersos no espaço de atributos e apresentam uma maior diversidade de características. Todavia, a Classe 0 apresenta um comprimento de curva menor, o que sugere um agrupamento mais homogêneo.

Tabela 4.38 – Número de segmentos e comprimento (u.a.) das curvas principais (com a distância euclidiana como abordagem).

Classes	**Número de Segmentos**	**Comprimento (u.a.)**
0	5	5,0195
1	5	5,8085

Na Figura 4.15 é apresentada a curva ROC obtida pelo algoritmo CPC utilizando a abordagem baseada na distância euclidiana. Observa-se também que a curva permanece significativamente acima da linha diagonal (classificador aleatório), indicando uma boa capacidade de discriminação entre as classes para diferentes limiares de decisão. O valor de AUC igual a 0,872 evidencia um desempenho satisfatório no processo de classificação.

Figura 4.15 – Curva ROC do método CPC com a abordagem de distância euclidiana.



Fonte: Do Autor (2026).

Os experimentos com a distância de *Gower* indicam que essa métrica pode ser uma alternativa promissora para aprimorar os resultados do método CPC em bases de dados com atributos mistos. A métrica de *Gower* apresentou resultados comparavéis aos da métrica euclidiana (Tabela 4.39 e 4.40). No entanto, seu uso em larga escala é limitado pelo alto custo computacional e pelo grande consumo de memória exigido para o cálculo e armazenamento da matriz de dissimilaridade.

Tabela 4.39 – Comparação do algoritmo baseado em CP com as métricas de distância no conjunto de treinamento com GWO.

Método	**ACC**	**F1**	**PR**	**RE**	**Kappa**	**Tempo (seg)**
CPC (Distância *Gower*)	**0,8342**	**0,8342**	**0,8344**	**0,8342**	**0,6684**	**2457,3068**
CPC (Distância Euclidiana)	0,8327	0,8326	0,8333	0,8327	0,6653	4729,6948

Tabela 4.40 – Comparação do algoritmo baseado em CP com as métricas de distância no conjunto de teste com GWO.

Método	ACC	F1	PR	RE	Kappa	Tempo (seg)
CPC (Distância *Gower*)	0,7813	0,7792	0,7924	0,7813	0,5625	**0,3870**
CPC (Distância Euclidiana)	**0,8363**	**0,8363**	**0,8363**	**0,8363**	**0,6725**	0,7660

Os resultados apresentados nos experimentos realizados evidenciam que a escolha do método de redução de dimensionalidade impacta diretamente tanto o desempenho preditivo quanto o custo computacional dos métodos avaliados. Embora o RF tenha apresentado os melhores resultados em comparação ao CPC, para os métodos de redução de dimensionalidade t-SNE e PCA, o tempo computacional foi maior. O método CPC opera em um espaço de menor dimensionalidade e classifica em um espaço com menos variáveis. Todavia, o RF envolve o processamento de um conjunto de múltiplas árvores de decisão, exigindo a avaliação de diversos nós e regras em cada árvore para cada amostra, o que aumenta o custo computacional. Assim, nota-se que o CPC é uma alternativa mais eficiente do ponto de vista computacional nestes cenários do que o RF.

Por fim, as análises sobre a importância dos atributos reforçam a influência predominante de variáveis relacionadas ao contexto escolar, familiar e socioeconômico, como o tipo de escola, a renda familiar mensal, a escolaridade dos pais e os itens de infraestrutura domiciliar. Os resultados também destacam que, embora técnicas como PCA e t-SNE possam reduzir o custo em determinadas etapas e tornar alguns modelos mais eficientes, o ganho de tempo nem sempre se traduz em ganho de desempenho.

5 CONCLUSÃO

Esta pesquisa teve o objetivo de aplicar e aprimorar o método CPC para a classificação do desempenho acadêmico dos estudantes no ENEM. Para isso, a pesquisa explorou o uso de CP associadas à técnica GWO para determinar automaticamente os valores dos hiperparâmetros do método CPC. Além disso, foram utilizados métodos tradicionais da literatura (ELM, NB e RF) para comparação, assim como abordagens de redução de dimensionalidade. Com base nos experimentos realizados, conclui-se que o objetivo foi alcançado, uma vez que o método de CP associado a técnica GWO mostrou-se promissor para o problema, apresentando um desempenho competitivo em relação aos métodos da literatura comparados. Assim, reforça-se o potencial das CP como alternativa para tarefas de classificação em dados educacionais.

Além disso, os experimentos evidenciaram que, quando as CP foram combinadas com PCA ou t-SNE, apresentaram menor custo computacional do que o RF, embora o RF tenha obtido os melhores resultados nessas configurações. Essa questão evidencia que, em alguns cenários operacionais, como em análises em larga escala, o método com a melhor métrica não necessariamente será o mais adequado ao se considerar o tempo de processamento e a escalabilidade. O método CPC opera em um espaço de menor dimensionalidade e classifica em um espaço com menos variáveis, desta forma, realiza cálculos de distância sobre uma representação de menor dimensão. Já o RF envolve o processamento de um conjunto de múltiplas árvores de decisão, exigindo a avaliação de diversos nós e regras em cada árvore para cada amostra, o que aumenta o custo computacional, principalmente quando há um grande volume de dados. Ressalta-se que, no tempo computacional do CPC, deve-se considerar duas fases, a fase de projeto e a fase operacional. A fase de projeto é mais cara computacionalmente, pois é a fase de construção do método, em que ocorrem a extração e o ajuste das CP, assim como a otimização dos segmentos. Já na fase operacional, o método está pronto e o processamento se limita à classificação, sendo consideravelmente mais rápido.

Outrossim, esta pesquisa identificou que variáveis do contexto escolar, familiar e socioeconômico aparecem entre as mais influentes para a classificação, como TP_ESCOLA (tipo de escola no ensino médio (pública/particular)), TP_LINGUA (língua estrangeira), Q023 (presença de telefone fixo na residência), TP_DEPENDENCIA_ADM_ESC (dependência administrativa da escola), Q014 (presença de máquina de lavar roupa na residência), Q008 (presença de banheiro na residência), Q016 (presença de forno micro-ondas na residência), Q006 (renda

familiar mensal), Q010 (presença de carro na residência), Q024 (presença de computador na residência), Q002 (escolaridade da mãe) e Q003 (ocupação do pai). Isso reforça que o desempenho no ENEM, tal como capturado pelos métodos, não é explicável apenas por fatores individuais, mas relaciona-se fortemente com condições estruturais.

Por fim, quanto às implicações para a área educacional, esta pesquisa aponta duas direções principais. A primeira é que as condições escolares e socioeconômicas estão associadas ao desempenho, o que reforça a importância de políticas de infraestrutura, apoio socioeducacional, permanência e condições de estudo como parte do debate sobre resultados em avaliações em grande escala. A segunda é que a metodologia proposta nesta pesquisa pode apoiar análises educacionais e o monitoramento de tendências, desde que seja usada como ferramenta de suporte à decisão.

5.1 Trabalhos Futuros

A partir desta pesquisa, é possível realizar novas investigações. Como trabalhos futuros, pode-se destacar:

- Realizar a criação de uma aplicação *Web* na qual o estudante poderá obter uma compreensão sobre o ENEM por meio de seus dados e dos demais estudantes, como a evolução do desempenho dos estudantes ao longo dos anos;
- Criar uma abordagem de *ensembles* com CP, explorando o potencial de múltiplas CP para classificação;
- Desenvolver uma biblioteca com uma abordagem de Inteligência Artificial Explicável com CP.

5.2 Publicações

Nesta subseção, são apresentadas as publicações decorrentes da pesquisa.

- A primeira publicação foi o artigo "*Socioeconomic Analysis of students who took the Enem between 2019 and 2022 using Machine Learning*" . O artigo apresenta uma análise socioeconômica do Enem de 2019 a 2022, visando identificar possíveis desigualdades

sociais e fatores que possam influenciar o desempenho dos estudantes no Enem. O artigo foi publicado e apresentado no *XLVI Ibero-Latin American Congress on Computational Methods in Engineering* (Macêdo *et al.*, 2024), em novembro de 2024.

- Adicionalmente, foi produzido um artigo intitulado "Classificação de Recorrência do Câncer de Tireoide utilizando Curvas Principais". O artigo buscou investigar a aplicação do algoritmo de CP na classificação da recorrência do câncer de tireoide após terapia com iodo radioativo. Foi realizada a otimização dos hiperparâmetros de CP, que impactam diretamente sua capacidade de modelagem e desempenho do método. O artigo foi publicado e apresentado no XVII Congresso Brasileiro de Inteligência Computacional (Macêdo *et al.*, 2025), em outubro de 2025.

- Outro artigo produzido foi "Otimização do Algoritmo de Curvas Principais K-segmentos com o *Grey Wolf Optimizer* (GWO)". O artigo explora o desempenho da otimização dos hiperparâmetros do classificador de CP utilizando a meta-heurística GWO em conjuntos de dados da literatura (*Breast_Cancer*, *Iris*, *Wine* e *Thyroid*). O artigo foi publicado e apresentado no XXXIV Congresso de Pós-Graduação da UFLA, em novembro de 2025.

REFERÊNCIAS

ALVES, F. I. A. B. Entre privilégios e barreiras: modelagem linear para estimar desempenho de participantes no enem. 2022.

AMOROSO, F. S. Inteligência artificial explicável com lime e shap aplicada à rede neural convolucional. 2023.

ANDRIOLA, W. B. Doze motivos favoráveis à adoção do exame nacional do ensino médio (enem) pelas instituições federais de ensino superior (ifes). **Ensaio: avaliação e políticas públicas em educação**, v. 19, n. 70, p. 107–125, 2011.

ANJOS, T. R. d. Projeto de vida e enem: uma análise do questionário socioeconômico e suas implicações para o ensino médio. Universidade Federal de São Carlos, 2017.

ASUNCION, A.; NEWMAN, D. *et al.* **UCI machine learning repository**. [*S.l.*]: Irvine, CA, USA, 2007.

BAI, Q.; ZHU, X. Features of off-line handwritten digit recognition based on principal curves. In: IET. **Symposium on ICT and Energy Efficiency and Workshop on Information Theory and Security (CIICT 2012)**. [*S.l.*], 2012. p. 209–214.

BANFIELD, J. D.; RAFTERY, A. E. Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. **Journal of the American statistical Association**, Taylor & Francis, v. 87, n. 417, p. 7–16, 1992.

BANNI, M. R.; OLIVEIRA, M. V. d. P.; BERNARDINI, F. C. Uma análise experimental usando mineração de dados educacionais sobre os dados do enem para identificação de causas do desempenho dos estudantes. In: SBC. **Workshop sobre as Implicações da Computação na Sociedade (WICS)**. [*S.l.*], 2021. p. 57–66.

BASU, S.; BANERJEE, A.; MOONEY, R. J. Semi-supervised clustering by seeding. In: **Proceedings of the nineteenth international conference on machine learning**. [*S.l.: s.n.*], 2002. p. 27–34.

BATISTA, G. E.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. **ACM SIGKDD explorations newsletter**, ACM New York, NY, USA, v. 6, n. 1, p. 20–29, 2004.

BEINERT, R. *et al.* On the dynamical system of principal curves in. **Communications in Statistics-Simulation and Computation**, Taylor & Francis, v. 53, n. 6, p. 2864–2879, 2024.

BELKINA, A. C. *et al.* Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. **Nature communications**, Nature Publishing Group UK London, v. 10, n. 1, p. 5415, 2019.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **The journal of machine learning research**, v. 13, n. 1, p. 281–305, 2012.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **The journal of machine learning research**, JMLR. org, v. 13, n. 1, p. 281–305, 2012.

BERTUCCI, R. A. Propriedades linguísticas da redação do enem: uma análise computacional. **Revista de Estudos da Linguagem**, v. 29, n. 2, p. 999–1032, 2021.

BEYERER, J.; HAGMANNS, R.; STADLER, D. **Pattern recognition: introduction, features, classifiers and principles**. [*S.l.*]: Walter de Gruyter GmbH & Co KG, 2024.

BEZERRA, E. Introdução à aprendizagem profunda. **Artigo–31º Simpósio Brasileiro de Banco de Dados–SBBD2016–Salvador**, 2016.

BHATNAGAR, S.; GILL, L.; GHOSH, B. Drone image segmentation using machine and deep learning for mapping raised bog vegetation communities. **Remote Sensing**, MDPI, v. 12, n. 16, p. 2602, 2020.

BONAWITZ, K. Towards federated learning at scale: Syste m design. **arXiv preprint arXiv:1902.01046**, 2019.

BORGES, F. E. d. M. *et al.* Classificador nao supervisionado baseado em curvas principais para detecçao de falhas em motor de induçao. In: **Congresso Brasileiro de Automática-CBA**. [*S.l.: s.n.*], 2019. v. 1, n. 1.

BORGES, F. E. de M. **OCPC-PY: One Class Classifier based on Principal Curves in Python**. 2023. Disponível em: https://pypi.org/project/ocpc-py/.

BORGES, F. E. de M. *et al.* One-class classifier based on principal curves. **Neural Computing and Applications**, Springer, v. 35, n. 26, p. 19015–19024, 2023.

BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. [*S.l.*]: Novatec Editora, 2014.

BRAGA, A. D. A.; FERREIRA, D. D.; BARBOSA, B. H. G. Seleção autom´´ atica de parâmetros iniciais do algoritmo k-segmentos com teaching-learning-based optimization. In: **Congresso Brasileiro de Automática-CBA**. [*S.l.: s.n.*], 2019. v. 1, n. 1.

BREIMAN, L. Random forests. **Machine learning**, p. 5–32, 2001. ISSN 0885-6125. Disponível em https://doi.org/10.1023/A:1010933404324.

BREIMAN, L. **Classification and regression trees**. [*S.l.*]: Routledge, 2017.

BROWNLEE, J. Undersampling algorithms for imbalanced classification. **Machine Learning Mastrey**, v. 27, 2020.

BRUCE, R. F. A bayesian approach to semi-supervised learning. In: **NLPRS**. [*S.l.: s.n.*], 2001. p. 57–64.

BURMAN, P. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. **Biometrika**, Oxford University Press, v. 76, n. 3, p. 503–514, 1989.

CAI, J. *et al.* A review on semi-supervised clustering. **Information Sciences**, Elsevier, v. 632, p. 164–200, 2023.

CHEN, D. Y. **Análise de dados com Python e Pandas**. [*S.l.*]: Novatec Editora, 2018.

CHEN, M.; MAO, S.; LIU, Y. Big data: A survey. **Mobile networks and applications**, Springer, v. 19, p. 171–209, 2014.

CORTIVO, Z. D.; MARQUES, J. M. Classificação de dados amostrais baseado no algoritmo k-segmentos. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 2, n. 1, 2014.

COVER, T. M. **Elements of information theory**. [*S.l.*]: John Wiley & Sons, 1999.

DAGAL, I. *et al.* Prioritized multi-step decision-making gray wolf optimization algorithm for engineering applications. **Engineering Reports**, Wiley Online Library, v. 7, n. 5, p. e70154, 2025.

DEVASSY, B. M.; GEORGE, S. Dimensionality reduction and visualisation of hyperspectral ink data using t-sne. **Forensic science international**, Elsevier, v. 311, p. 110194, 2020.

DEVROYE, L.; WAGNER, T. Distribution-free performance bounds for potential function rules. **IEEE Transactions on Information Theory**, IEEE, v. 25, n. 5, p. 601–604, 1979.

DING, S. *et al.* Extreme learning machine: algorithm, theory and applications. **Artificial Intelligence Review**, Springer, v. 44, n. 1, p. 103–115, 2015.

DONG, S.; WANG, P.; ABBAS, K. A survey on deep learning and its applications. **Computer Science Review**, Elsevier, v. 40, p. 100379, 2021.

DUTRA, J. F. **Análise de perfis de estudantes no ENEM considerando hábitos de estudo**. Dissertação (Mestrado), 2024.

DUTRA, J. F.; JÚNIOR, J. B. F.; FERNANDES, D. Y. de S. Fatores que podem interferir no desempenho de estudantes no enem: uma revisão sistemática da literatura. **Revista Brasileira de Informática na Educação**, v. 31, p. 323–351, 2023.

EDWARDS, A. W. Ra fischer, statistical methods for research workers, (1925). In: **Landmark writings in western mathematics 1640-1940**. [*S.l.*]: Elsevier, 2005. p. 856–870.

EIBEN, A. E.; SMITH, J. E. **Introduction to evolutionary computing**. [*S.l.*]: Springer, 2015.

FAIER, J. M. Curvas principais aplicadas na identificação de descargas parciais em equipamentos de potência. **Engineering Department (COPPE). Universidade Federal do Rio de Janeiro, Rio de Janeiro**, 2006.

FERNANDEZ, H. L. Classificaçao de navios baseada em curvas principais. **Tesede Mestrado, Programa de Engenharia Elétrica, COPPE/UFRJ, Rio de Janeiro, RJ**, 2005.

FERREIRA, D. D. *et al.* Exploiting principal curves for power quality monitoring. **Electric power systems research**, Elsevier, v. 100, p. 1–6, 2013.

FERREIRA, D. D. *et al.* A new power quality deviation index based on principal curves. **Electric Power Systems Research**, Elsevier, v. 125, p. 8–14, 2015.

FILHO, R. L. C. S. **Modelo de análise e predição do desempenho dos alunos dos Institutos Federais de Educação usando o ENEM como indicador de qualidade escolar**. Universidade Federal de Pernambuco, 2017.

FRANCO, J. J. Fatores e evidências sobre o exame nacional do ensino médio (enem): uma abordagem exploratória e experimental com mineração de dados. 2021.

FRANCO, J. J. *et al.* Usando mineraçao de dados para identificar fatores mais importantes do enem dos últimos 22 anos. In: SBC. **Anais do XXXI Simpósio Brasileiro de Informática na Educação**. [*S.l.*], 2020. p. 1112–1121.

FREITAS, N. C. A. de. **Machine learning: técnicas e cases**. [*S.l.*]: Editora Senac São Paulo, 2024.

GARCIA, R. A.; RIOS-NETO, E. L. G.; MIRANDA-RIBEIRO, A. d. Efeitos rendimento escolar, infraestrutura e prática docente na qualidade do ensino médio no brasil. **Revista brasileira de Estudos de População**, SciELO Brasil, v. 38, p. e0152, 2021.

GIRALDI, G. A. Machine learning and pattern recognition. **Lecture Notes-Graduate Program in Nanobiosystems, UFRJ-FIOCRUZ-INMETRO-LNCC**, 2021.

GOMES, C. M. A. *et al.* Preditores do desempenho em matemática de estudantes do ensino médio. **Psicologia: Teoria e Pesquisa**, SciELO Brasil, v. 36, p. e3638, 2021.

GOODFELLOW, I. **Deep learning**. [*S.l.*]: MIT press, 2016.

GOWER, J. C. A general coefficient of similarity and some of its properties. **Biometrics**, JSTOR, p. 857–871, 1971.

GUO, P.; CHENG, W.; WANG, Y. Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems. **Expert Systems with Applications**, Elsevier, v. 71, p. 57–68, 2017.

HASTIE, T.; STUETZLE, W. Principal curves. **Journal of the American statistical association**, Taylor & Francis, v. 84, n. 406, p. 502–516, 1989.

HAYKIN, S. **Redes neurais: princípios e prática**. [*S.l.*]: Bookman Editora, 2001.

HE, P.; WU, W. Levy flight-improved grey wolf optimizer algorithm-based support vector regression model for dam deformation prediction. **Frontiers in Earth Science**, Frontiers Media SA, v. 11, p. 1122937, 2023.

HUANG, G. *et al.* Trends in extreme learning machines: A review. **Neural Networks**, Elsevier, v. 61, p. 32–48, 2015.

HUANG, G.-B. What are extreme learning machines? filling the gap between frank rosenblatt's dream and john von neumann's puzzle. **Cognitive Computation**, Springer, v. 7, n. 3, p. 263–278, 2015.

HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE. **2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)**. [*S.l.*], 2004. v. 2, p. 985–990.

Inep. **Saiba como se inscrever no ENEM 2023**. 2023. Acesso em: 28 out. 2024. Disponível em: https://www.gov.br/inep/pt-br/assuntos/noticias/enem/saiba-como-se-inscrever-no-enem-2023.

INEP. **Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira | Inep**: Microdados. 2026. Acesso em: 04 jan. 2026. Disponível em: https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/enem.

JAMAIN, A.; HAND, D. J. The naive bayes mystery: A classification detective story. **Pattern Recognition Letters**, v. 26, n. 11, p. 1752–1760, 2005.

JOLLIFFE, I. T.; CADIMA, J. Principal component analysis: a review and recent developments. **Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences**, the Royal Society publishing, v. 374, n. 2065, p. 20150202, 2016.

JÚNIOR, E. M. Análise de dados estruturados do enem para aprimorar políticas educacionais públicas. 2023.

KÉGL, B. *et al.* Learning and design of principal curves. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 22, n. 3, p. 281–297, 2000.

KLEINKE, M. U. Influência do status socioeconômico no desempenho dos estudantes nos itens de física do enem 2012. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 39, n. 2, p. e2402, 2017.

KOBEL, A.; SAGRALOFF, M. On the complexity of computing with planar algebraic curves. **Journal of Complexity**, Elsevier, v. 31, n. 2, p. 206–236, 2015.

KOUTROUMBAS, K.; THEODORIDIS, S. **Pattern recognition**. [*S.l.*]: Academic Press, 2008.

LI, B. *et al.* An improved grey wolf algorithm and its localization research in complex indoor environments. **Scientific Reports**, Nature Publishing Group UK London, v. 15, n. 1, p. 7329, 2025.

LI, K. *et al.* Grey wolf optimization algorithm based on cauchy-gaussian mutation and improved search strategy. **Scientific Reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 18961, 2022.

LIU, H.; MOTODA, H. **Computational methods of feature selection**. [*S.l.*]: CRC press, 2007.

LIU, X. *et al.* Self-supervised learning: Generative or contrastive. **IEEE transactions on knowledge and data engineering**, IEEE, v. 35, n. 1, p. 857–876, 2021.

LOU, Y.; CARUANA, R.; GEHRKE, J. Intelligible models for classification and regression. In: **Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining**. [*S.l.: s.n.*], 2012. p. 150–158.

LUCKESI, C. C. **Avaliação da aprendizagem escolar: estudos e proposições**. [*S.l.*]: Cortez editora, 2014.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. **Advances in neural information processing systems**, v. 30, 2017.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. **Journal of machine learning research**, v. 9, n. 11, 2008.

MACÊDO, B. da S. *et al.* Socioeconomic analysis of students who took the enem between 2019 and 2022 using machine learning. In: **Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE)**. [*S.l.: s.n.*], 2024.

MACÊDO, B. da S. *et al.* Classificação de recorrência do câncer de tireoide utilizando curvas principais. In: AES, F. G.; FERREIRA, D.; BARRETO, G. (Ed.). **Anais do XVII Congresso Brasileiro de Inteligência Computacional (CBIC 2025)**. Belo Horizonte, MG: SBIC, 2025. p. 1–8.

MACEDO, B. da S.; SAPORETTI, C. M. Analysis of the impact of the pandemic on social inequalities in enem 2019 and 2020 using machine learning. **Semina: Ciências Exatas e Tecnológicas**, v. 44, p. e48234–e48234, 2023.

MÁXIMO, B. C.; RIBEIRO, L. F. C. Análise com modelagem multinível: Contribuição para estudo sobre as variáveis significantes para as notas no enem. **CONNECTION LINE-REVISTA ELETRÔNICA DO UNIVAG**, n. 30, 2023.

MCKINNEY, W. **Python para análise de dados: Tratamento de dados com Pandas, NumPy e IPython**. [*S.l.*]: Novatec Editora, 2018.

MCMAHAN, B.; RAMAGE, D. Federated learning: Collaborative machine learning without centralized training data. **Google Research Blog**, v. 3, 2017.

MELO, R. O. *et al.* Impacto das variáveis socioeconômicas no desempenho do enem: uma análise espacial e sociológica. **Revista de Administração Pública**, SciELO Brasil, v. 55, n. 6, p. 1271–1294, 2021.

MENG, K.; ELOYAN, A. Principal manifold estimation via model complexity selection. **Journal of the Royal Statistical Society Series B: Statistical Methodology**, Oxford University Press, v. 83, n. 2, p. 369–394, 2021.

MILLER, T. Explanation in artificial intelligence: Insights from the social sciences. **Artificial intelligence**, Elsevier, v. 267, p. 1–38, 2019.

MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey wolf optimizer. **Advances in engineering software**, Elsevier, v. 69, p. 46–61, 2014.

MITCHELL, T. M. **Machine learning**. [*S.l.*]: McGraw-hill, 1997.

MITCHELL, T. M. **The discipline of machine learning**. [*S.l.*]: Carnegie Mellon University, School of Computer Science, Machine Learning . . . , 2006. v. 9.

MORAES, E. C. C.; FERREIRA, D. D. A principal curve-based method for data clustering. In: IEEE. **2016 International Joint Conference on Neural Networks (IJCNN)**. [*S.l.*], 2016. p. 3966–3971.

MORAES, E. C. C.; FERREIRA, D. D. A principal curve-based method for data clustering. In: IEEE. **2016 International Joint Conference on Neural Networks (IJCNN)**. [*S.l.*], 2016. p. 3966–3971.

MORAES, E. C. C. *et al.* Data clustering based on principal curves. **Advances in Data Analysis and Classification**, Springer, v. 14, n. 1, p. 77–96, 2020.

MURO, C. *et al.* Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations. **Behavioural processes**, Elsevier, v. 88, n. 3, p. 192–197, 2011.

NASIR, M. *et al.* A comprehensive review on applications of grey wolf optimizer in energy systems. **Archives of Computational Methods in Engineering**, Springer, p. 1–41, 2024.

NEIVA, D. K. **Interpretação de modelos complexos de aprendizado de máquina**. Tese (Doutorado) — Universidade de São Paulo, 2023.

NETO, A. S. Regressão logística em microdados da educação. 2023.

NETO, N. W. *et al.* Minerando dados para entender o impacto da pandemia da covid-19 no exame nacional do ensino médio. Universidade Federal do Maranhão, 2023.

NOGUERA, V. E. R.; AGUIAR, C. D. d. Análise de dados do enem baseada em data warehousing, mineração de dados, estatística inferencial e processamento paralelo e distribuído. 2023.

NUNES, D. *et al.* Evaluating gpt-3.5 and gpt-4 models on brazilian university admission exams. **arXiv preprint arXiv:2303.17003**, 2023.

OLIVEIRA, C. T. Otimização de um trocador de calor casco e tubos utilizando o algoritmo lobo cinzento. Universidade do Vale do Rio dos Sinos, 2018.

PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 22, n. 10, p. 1345–1359, 2009.

PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011.

PERKEL, J. M. By jupyter, it all makes sense. 2018.

PINHO, C. *et al.* Aplicação de técnicas de inteligência artificial para classificação de fuga ao tema em redações. **Educação em Revista**, SciELO Brasil, v. 40, p. e39773, 2024.

RAJKOMAR, A.; DEAN, J.; KOHANE, I. Machine learning in medicine. **New England Journal of Medicine**, Mass Medical Soc, v. 380, n. 14, p. 1347–1358, 2019.

RIS-ALA, R. **Fundamentos de Aprendizagem por Reforço**. [*S.l.*]: Rafael Ris-Ala, 2023.

RODRIGUEZ-GALIANO, V. F. *et al.* An assessment of the effectiveness of a random forest classifier for land-cover classification. **ISPRS journal of photogrammetry and remote sensing**, Elsevier, v. 67, p. 93–104, 2012.

RODRÍGUEZ-PÉREZ, R.; BAJORATH, J. Interpretation of compound activity predictions from complex machine learning models using local approximations and shapley values. **Journal of medicinal chemistry**, ACS Publications, v. 63, n. 16, p. 8761–8777, 2019.

ROSANO, B. *et al.* Aprendizado federado hierárquico. 2022.

SA, J. M. D. **Pattern recognition: concepts, methods and applications**. [*S.l.*]: Springer Science & Business Media, 2012.

SAMMUT, C.; WEBB, G. I. **Encyclopedia of machine learning**. [*S.l.*]: Springer Science & Business Media, 2011.

SANCHES, M. K. **Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados**. Tese (Doutorado) — Universidade de São Paulo, 2003.

SANTOS, J. M. C. T. Exame nacional do ensino médio: entre a regulação da qualidade do ensino médio e o vestibular. **Educar em revista**, SciELO Brasil, p. 195–205, 2011.

SHEHAB, M.; TAHERDANGKOO, R.; BUTSCHER, C. Towards reliable barrier systems: a constrained xgboost model coupled with gray wolf optimization for maximum swelling pressure of bentonite. **Computers and Geotechnics**, Elsevier, v. 168, p. 106132, 2024.

SHLENS, J. A tutorial on principal component analysis. **arXiv preprint arXiv:1404.1100**, 2014.

SILVA, L. A.; MORINO, A. H.; SATO, T. M. C. Prática de mineração de dados no exame nacional do ensino médio. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [*S.l.: s.n.*], 2014. v. 3, n. 1, p. 651.

SILVA, V. A. A. da *et al.* Identificação de desigualdades sociais a partir do desempenho dos alunos do ensino médio no enem 2019 utilizando mineração de dados. In: SBC. **Anais do XXXI Simpósio Brasileiro de Informática na Educação**. [*S.l.*], 2020. p. 72–81.

SILVEIRA, I. C.; MAUÁ, D. D. Advances in automatically solving the enem. In: IEEE. **2018 7th Brazilian Conference on Intelligent Systems (BRACIS)**. [*S.l.*], 2018. p. 43–48.

SOUSA, L. P. O. *et al.* A principal curves-based method for electronic tongue data analysis. **IEEE Sensors Journal**, IEEE, v. 21, n. 4, p. 4957–4965, 2020.

SOUZA, K. R. G. *et al.* Um estudo sobre o desempenho das escolas públicas do df sob o ponto de vista do enem. Universidade Católica de Brasília, 2019.

SYARIF, I.; PRUGEL-BENNETT, A.; WILLS, G. Svm parameter optimization using grid search and genetic algorithm to improve classification performance. **TELKOMNIKA (Telecommunication Computing Electronics and Control)**, v. 14, n. 4, p. 1502–1509, 2016.

TAYLOR, M. E.; STONE, P. Transfer learning for reinforcement learning domains: A survey. **Journal of Machine Learning Research**, v. 10, n. 7, 2009.

TEAM, T. pandas development. **pandas-dev/pandas: Pandas**. Zenodo, 2020. Disponível em: https://doi.org/10.5281/zenodo.3509134.

THENG, D.; BHOYAR, K. K. Feature selection techniques for machine learning: a survey of more than two decades of research. **Knowledge and Information Systems**, Springer, v. 66, n. 3, p. 1575–1637, 2024.

TORREY, L.; SHAVLIK, J. Transfer learning. In: **Handbook of research on machine learning applications and trends: algorithms, methods, and techniques**. [*S.l.*]: IGI global, 2010. p. 242–264.

TOSI, S. **Matplotlib for Python developers**. [*S.l.*]: Packt Publishing Ltd, 2009.

TOU, J. T.; GONZALEZ, R. C. Pattern recognition principles. 1974.

VERBEEK, J. J.; VLASSIS, N.; KRÖSE, B. A soft k-segments algorithm for principal curves. In: SPRINGER. **International Conference on Artificial Neural Networks**. [*S.l.*], 2001. p. 450–456.

VERBEEK, J. J.; VLASSIS, N.; KRÖSE, B. A k-segments algorithm for finding principal curves. **Pattern Recognition Letters**, Elsevier, v. 23, n. 8, p. 1009–1017, 2002.

VIGGIANO, E.; MATTOS, C. O desempenho de estudantes no enem 2010 em diferentes regiões brasileiras. **Revista Brasileira de Estudos Pedagógicos**, INEP, v. 94, n. 237, p. 417–438, 2013.

VINAY, A. **Thyroid Cancer Recurrence Dataset**. 2025. Kaggle. Disponível em https://www.kaggle.com/datasets/aneevinay/thyroid-cancer-recurrence-dataset/data.

WAGSTAFF, K. *et al.* Constrained k-means clustering with background knowledge. In: **Icml**. [*S.l.: s.n.*], 2001. v. 1, p. 577–584.

WANG, J. *et al.* A review on extreme learning machine. **Multimedia Tools and Applications**, Springer, v. 81, n. 29, p. 41611–41660, 2022.

WESTPHALEN-RS, C. d. F.; VARGAS, A. S. de. As políticas públicas para a educação superior no brasil pós ldb/96: O enem, sisu, prouni e fies e suas (des) continuidades. 2021.

XAVIER, R. S. **Uma Estrutura Conceitual para o Estudo da Computação Natural**. [*S.l.*]: Editora Dialética, 2023.

XIA, X.; WANG, X. Fault diagnosis of planetary gearbox based on hierarchical refined composite multiscale fuzzy entropy and optimized lssvm. **Entropy**, MDPI, v. 27, n. 5, p. 512, 2025.

YOU, S. *et al.* Principal curved based retinal vessel segmentation towards diagnosis of retinal diseases. In: IEEE. **2011 IEEE First International Conference on Healthcare Informatics, Imaging and Systems Biology**. [*S.l.*], 2011. p. 331–337.

ZHANG, C. *et al.* A survey on federated learning. **Knowledge-Based Systems**, Elsevier, v. 216, p. 106775, 2021.

ZOU, F.; CHEN, D.; XU, Q. A survey of teaching–learning-based optimization. **Neurocomputing**, Elsevier, v. 335, p. 366–383, 2019.

APÊNDICES

APÊNDICE A – OUTRAS VARIÁVEIS PRESENTES NA BASE DE DADOS DO ENEM

Tabela 1 – Descrição das variáveis presentes na base de dados do ENEM.

Variável	Descrição	Tipo da Variável
NU_INSCRICAO	Número de Inscrição	Numérica
CO_MUNICIPIO_ESC	Código do município da escola	Numérica
NO_MUNICIPIO_ESC	Nome do município da escola	Alfanumérica
SG_UF_ESC	Sigla da unidade da federação da escola	Alfanumérica
CO_UF_ESC	Código da Unidade da Federação da escola	Numérica
TP_SIT_FUNC_ESC	Situação de funcionamento (Escola)	Numérica
CO_MUNICIPIO_PROVA	Código do município da aplicação da prova	Numérica
NO_MUNICIPIO_PROVA	Nome do município da aplicação da prova	Alfanumérica
CO_UF_PROVA	Código da Unidade da Federação da aplicação da prova	Alfanumérica
CO_PROVA_CN	Código do tipo de prova de Ciências da Natureza	Numérica
CO_PROVA_CH	Código do tipo de prova de Ciências Humanas	Numérica
CO_PROVA_LC	Código do tipo de prova de Linguagens e Códigos	Numérica
CO_PROVA_MT	Código do tipo de prova de Matemática	Numérica
TX_RESPOSTAS_CN	Vetor com as respostas da parte objetiva da prova de Ciências da Natureza	Alfanumérica
TX_RESPOSTAS_CH	Vetor com as respostas da parte objetiva da prova de Ciências Humanas	Alfanumérica
TX_RESPOSTAS_LC	Vetor com as respostas da parte objetiva da prova de Linguagens e Códigos	Alfanumérica
TX_RESPOSTAS_MT	Vetor com as respostas da parte objetiva da prova de Matemática	Alfanumérica
TX_GABARITO_CN	Vetor com o gabarito da parte objetiva da prova de Ciências da Natureza	Alfanumérica
TX_GABARITO_CH	Vetor com o gabarito da parte objetiva da prova de Ciências Humanas	Alfanumérica

Fonte: Inep (2026).

Tabela 2 – Continuação da Descrição das variáveis presentes na base de dados do ENEM.

Variável	Descrição	Tipo da Variável
TX_GABARITO_LC	Vetor com o gabarito da parte objetiva da prova de Linguagens e Códigos	Alfanumérica
TX_GABARITO_MT	Vetor com o gabarito da parte objetiva da prova de Matemática	Alfanumérica

Fonte: Inep (2026).