



Trabalho de Conclusão de Curso

Seleção de características para otimização de modelos de aprendizagem de máquina:

Uma abordagem baseada em algoritmos genéticos

Hugo Gabriel de Melo Santos

orientado por

Prof. Dr. Erick Barboza

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
20 de Maio de 2025

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

**SELEÇÃO DE CARACTERÍSTICAS PARA OTIMIZAÇÃO DE
MODELOS DE APRENDIZAGEM DE MÁQUINA:
UMA ABORDAGEM BASEADA EM ALGORITMOS
GENÉTICOS**

Trabalho de Conclusão de Curso apresentado
ao Instituto de Computação da Universidade
Federal de Alagoas como requisito parcial
para a obtenção do grau de Bacharel em En-
genharia de Computação.

Hugo Gabriel de Melo Santos

Orientador: Prof. Dr. Erick Barboza

Banca Examinadora:

Erick Barboza Prof. Dr., IC-UFAL

Maceió, Alagoas
20 de Maio de 2025

Catlogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Jorge Raimundo da Silva – CRB-4 – 1528

S237s	Santos, Hugo Gabriel de Melo. Seleção de características para otimização de modelos de aprendizagem de máquina: Uma abordagem baseada em algoritmos genéticos / Hugo Gabriel de Melo Santos. – 2025. 54 f. Orientador: Erick Barboza. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Computação.) Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2025. Bibliografia: f. 52-54 1. Algoritmos genéticos – Seleção de características. 2. Otimização – Aprendizado de máquina. 3. Algoritmo – BRKGA. I. Título.
-------	--

CDU: 004.021



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Instituto de Computação - IC
Campus A. C. Simões - Av. Lourival de Melo Mota, BL 12
Tabuleiro do Martins, Maceió/AL - CEP: 57.072-970
Telefone: (082) 3214-1401



Trabalho de Conclusão de Curso - TCC

Formulário de Avaliação

Nome do Aluno:

HUGO GABRIEL DE MELO SANTOS

Nº de Matrícula:

18112109

Título do TCC (Tema):

Seleção de características para otimização de modelos de aprendizagem de máquina: Uma abordagem baseada em algoritmos genéticos

Banca Examinadora

Erick de Andrade Barboza
Nome do Orientador

BRUNO COSTA E SILVA NOGUEIRA
Nome do Professor

MARCELO COSTA OLIVEIRA
Nome do Professor

Documento assinado digitalmente



ERICK DE ANDRADE BARBOZA
Data: 20/05/2025 10:18:29-0300
Verifique em <https://validar.iti.gov.br>

Documento assinado digitalmente



BRUNO COSTA E SILVA NOGUEIRA
Data: 23/05/2025 15:03:49-0300
Verifique em <https://validar.iti.gov.br>

Assinatura

Documento assinado digitalmente



MARCELO COSTA OLIVEIRA
Data: 23/05/2025 15:45:09-0300
Verifique em <https://validar.iti.gov.br>

Assinatura

Data da Defesa

20 / 05 / 2025

Nota Obtida

9,0 (nove)

Observações

Coordenador do Curso
De Acordo

Documento assinado digitalmente



JOBSON DE ARAUJO NASCIMENTO
Data: 28/05/2025 12:34:32-0300
Verifique em <https://validar.iti.gov.br>

Assinatura

Agradecimentos

Quero começar agradecendo à minha família, que sempre me deu a oportunidade de estudar e me desenvolver desde a base. Um agradecimento especial ao meu pai Gerônimo, minha mãe Ana e meus irmãos Gerônimo Junior e Iago.

A caminhada foi cheia de desafios, mas tive a sorte de contar com professores incríveis que fizeram a diferença nessa jornada: Thiago Damacesno Cordeiro, Marcelo Costa Oliveira e Erick de Andrade Barboza — sou muito grato pelo apoio e ensinamentos.

Acima de tudo, pode-se falar que foi uma jornada divertida com os amigos que participaram da minha turma, o Clube da Luluzinha, com Luana Cordinha, Lucas Massa e seu companheiro Alfred, Big Mage (Mateus do Quadrado), Jhonnye, Cláudio, João Britinho, Bruna, Hiago, Cabral, Ruan e Zigo. Vocês foram essenciais para superar cada etapa dessa graduação. E também amigos que conheci a partir da faculdade, como Larissa, Ester DeL, Victor, Marcus, Lineu, Laura e Luiza.

Por último, expresso minha gratidão a todos que, mesmo não mencionados, contribuíram de alguma forma ao longo dessa trajetória, e também a mim mesmo, por ter enfrentado os desafios com resiliência e seguido em frente, apesar de todas as dificuldades.

Em resumo, foi legal.

Resumo

Modelos preditivos de aprendizado de máquina dependem de um conjunto adequado de atributos para realizar classificações e previsões de forma eficiente. No entanto, a elevada dimensionalidade de dados apresenta desafios significativos, onde a qualidade e a relevância das características influenciam diretamente o desempenho dos modelos, podendo acarretar em problemas como aumento da complexidade computacional e dificuldade de encontrar padrões relevantes, fenômeno conhecido como "maldição da dimensionalidade". Este trabalho investiga a aplicação de algoritmos genéticos para a seleção de características em modelos de aprendizado de máquina, com o objetivo de otimizar seu desempenho preditivo, selecionando apenas as dimensões relevante para o problema. Os algoritmos genéticos (GAs) são meta-heurísticas inspiradas no processo de seleção natural, utilizando operações como seleção, cruzamento e mutação para evoluir soluções ao longo de sucessivas gerações. Essas técnicas são amplamente utilizadas para resolver problemas de otimização complexos, especialmente aqueles que envolvem espaços de busca de alta dimensionalidade. A pesquisa aborda a aplicação tanto do Algoritmo Genético Convencional (GA) quanto de sua variante baseada em chaves aleatórias, BRKGA (Biased Random-Key Genetic Algorithm). Os experimentos demonstraram que ambas as abordagens são eficazes na redução da dimensionalidade dos dados, melhorando as métricas de desempenho, a partir da otimização da métrica F1-Score. A metodologia proposta não apenas contribui para o desenvolvimento de modelos mais robustos e escaláveis, mas também oferece maior interpretabilidade e eficiência computacional.

Palavras-chave: Algoritmos Genéticos, Seleção de Características, Otimização, Aprendizado de Máquina, BRKGA.

Abstract

Predictive machine learning models rely on an appropriate set of attributes to perform classifications and predictions efficiently. However, the high dimensionality of data presents significant challenges, where the quality and relevance of features directly influence model performance, potentially leading to problems such as increased computational complexity and difficulty in finding relevant patterns, a phenomenon known as the "curse of dimensionality." This work investigates the application of genetic algorithms for feature selection in machine learning models, aiming to optimize their predictive performance by selecting only the relevant dimensions for the problem. Genetic algorithms (GAs) are metaheuristics inspired by the process of natural selection, employing operations such as selection, crossover, and mutation to evolve solutions across successive generations. These techniques are widely used to solve complex optimization problems, especially those involving high-dimensional search spaces. The study addresses the application of both the Conventional Genetic Algorithm (GA) and its random-key-based variant, BRKGA (Biased Random-Key Genetic Algorithm). The experiments demonstrated that both approaches are effective in reducing the dimensionality of data, improving performance metrics through the optimization of the F1-Score. The proposed methodology not only contributes to the development of more robust and scalable models but also provides greater interpretability and computational efficiency.

Palavras-chave: Genetic Algorithms, Feature Selection, Optimization, Machine Learning, BRKGA.

Lista de Figuras

2.1	Etapas de um algoritmo genético. Fonte: Autor, 2025.	18
2.2	Como funciona um KNN classificador. Fonte: Autor, 2025.	24
2.3	Exemplo de modelo simples de árvore de decisão. Fonte: Autor, 2025.	25
2.4	Funcionamento de uma regressão logística para problemas de classificação.	
	Fonte: Autor, 2025.	27
2.5	Estrutura de uma Matriz de Confusão. Fonte: Autor, 2025.	28
3.1	Exemplo de um cromossomo único. Fonte: Autor, 2025.	30
3.2	Exemplo de uma população inicial. Fonte: Autor, 2025.	31
3.3	Exemplo de uma população onde cada cromossomo tem seu valor de aptidão	
	calculado utilizando o <i>recall</i> para um classificador específico. Fonte: Autor,	
	2025.	32
3.4	Cromossomos da elite que serão selecionados para próxima geração. Fonte:	
	Autor, 2025.	33
3.5	Cruzamento entre um pai pertencente a elite e um pai selecionado do res-	
	tante da população. Fonte: Autor, 2025.	34
3.6	Mutação realizada nos genes de um cromossomo separado para a próxima	
	geração. Fonte: Autor, 2025.	34
3.7	Exemplo de cromossomo aleatório gerado pelo BRKGA. Fonte: Autor, 2025.	35
3.8	Processo de transição entre populações no BRKGA. Fonte: Autor, 2025.	36
3.9	Processo de cruzamento entre cromossomos no BRKGA. Fonte: Autor, 2025.	37
3.10	Exemplo de mutação do cromossomo aleatório gerado pelo BRKGA. Fonte:	
	Autor, 2025.	38
4.1	Comparação de Métricas (KNN). Fonte: Autor, 2025.	41
4.2	Matriz de confusão em caso médio (KNN). Fonte: Autor, 2025.	42
4.3	Comparação de Métricas (Árvore de Decisão). Fonte: Autor, 2025.	43
4.4	Matriz de confusão em caso médio (Árvore de Decisão). Fonte: Autor, 2025.	45
4.5	Comparação de Métricas (Regressão Logística) Fonte: Autor, 2025.	46
4.6	Matriz de confusão em caso médio (Regressão Logística) . Fonte: Autor,	
	2025.	47

Lista de Tabelas

2.1	Relação entre a composição da Genética e Algoritmos Genéticos. Fonte: Autor, 2025.	18
3.1	Desempenho dos modelos a base de dados antes da otimização.	38
4.1	Tabela de parâmetros utilizados em cada algoritmo genético. Fonte: Autor, 2025.	41
4.2	Desempenho médio do modelo (KNN) não otimizado e otimizado para cada métrica. Fonte: Autor, 2025.	42
4.3	Comparação entre os tempos médios de execução de um modelo não otimizado para um otimizado (KNN). Fonte: Autor, 2025.	43
4.4	Desempenho médio do modelo (Árvore de Decisão) não otimizado e otimizado para cada métrica. Fonte: Autor, 2025.	44
4.5	Comparação entre os tempos médios de execução de um modelo não otimizado para um otimizado (Árvore de Decisão). Fonte: Autor, 2025.	45
4.6	Desempenho médio do modelo (Regressão Logística) não otimizado e otimizado para cada métrica. Fonte: Autor, 2025.	47
4.7	Comparação entre os tempos médios de execução de um modelo não otimizado para um otimizado (Regressão Logística). Fonte: Autor, 2025.	47
4.8	Desempenho dos modelos a base de dados após a otimização.	49

Lista de Símbolos

- d Subconjunto de características selecionadas por uma solução.
- R Subconjunto de características não selecionadas por uma solução.
- D Conjunto de características totais na base utilizada.
- Y Variável de saída da base de dados.
- X_i Variáveis independentes utilizadas nos modelos.
- P Probabilidade da instância pertencer à classe positiva.
- z Combinação linear das variáveis preditoras e seus respectivos coeficientes.
- β_i Coeficientes associados às variáveis independentes da regressão logística que medem o impacto de cada variável na previsão do desfecho.
- G Quantidade de genes nos cromossomos daquela base ($|D|$).

Lista de Abreviações

GA *Algoritmo Genético (em inglês, Genetic Algorithm)*

BRKGA *Algoritmo Genético de Chaves Aleatórias Enviesadas (em inglês, Biased Random Key Genetic Algorithm)*

KNN *K-Vizinhos Mais Próximos (em inglês, K-Nearest Neighbors)*

IA Inteligência Artificial

ML Aprendizado de Máquina (em inglês, Machine Learning)

VP Verdadeiros Positivos

VN Verdadeiros Negativos

FP Falsos Positivos

FN Falsos Negativos

Sumário

1	Introdução	12
1.1	Objetivo Geral	13
1.1.1	Objetivos Específicos	13
1.2	Organização do Trabalho	13
2	Fundamentação Teórica	15
2.1	Revisão da Literatura	15
2.2	Problema de Seleção de Características	16
2.3	Algoritmo Genético	17
2.3.1	População Inicial	19
2.3.2	Métodos de Seleção	19
2.3.3	Cruzamento	20
2.3.4	Mutação	21
2.3.5	Algoritmo Genético de Chaves Aleatórias Enviesadas (BRKGA)	21
2.4	Aprendizado de Máquina	22
2.4.1	Conceitos Fundamentais	22
2.4.2	K-Vizinhos Mais Próximos (KNN)	23
2.4.3	Árvore de Decisão	24
2.4.4	Regressão Logística	26
2.4.5	Métricas	27
3	Metodologia	30
3.1	Algoritmo Genético	30
3.1.1	Codificação do Cromossomo	30
3.1.2	População Inicial	31
3.1.3	Avaliação de Aptidão	31
3.1.4	Método de Seleção	33
3.1.5	Cruzamento e Mutação	33
3.1.6	Algoritmo Genético de Chaves Aleatórias Enviesadas (BRKGA)	35
3.2	Aprendizado de Máquina	38

4 Resultados e Discussões	40
4.1 K-Vizinhos mais próximos (KNN)	41
4.2 Árvore de Decisão	43
4.3 Regressão Logística	46
4.4 Comparação entre modelos	48
4.4.1 Eficiência Computacional	49
5 Conclusão	51
Bibliografia	52

Capítulo 1

Introdução

A crescente disponibilidade de dados em diversas áreas do conhecimento tem impulsionado o desenvolvimento de modelos preditivos cada vez mais sofisticados, com aplicações que vão desde a medicina até a engenharia e os negócios. Contudo, a elevada dimensionalidade desses dados representa um desafio significativo, pois nem todas as características disponíveis contribuem de forma relevante para a construção de modelos precisos e eficientes. Muitas variáveis podem ser redundantes ou irrelevantes, comprometendo o desempenho dos algoritmos de aprendizado de máquina (Jianqing Fan and Zhu (2018)).

A seleção de características surge como uma etapa fundamental para lidar com esse problema, buscando identificar o subconjunto mais informativo de atributos que maximize o desempenho dos modelos. Essa prática oferece uma série de vantagens, como a redução da complexidade computacional, a melhoria da capacidade de generalização dos modelos e a maior interpretabilidade dos resultados. Entretanto, a busca por esse subconjunto ótimo é um problema computacionalmente complexo, especialmente em cenários com grandes volumes de dados (Isabelle Guyon (2003)).

Nesse contexto, técnicas baseadas em meta-heurísticas, como os algoritmos genéticos, têm se mostrado promissoras para a solução do problema de seleção de características. Inspirados nos mecanismos evolutivos da natureza, esses algoritmos realizam uma busca inteligente no espaço de soluções, combinando operações de seleção, cruzamento e mutação para encontrar configurações eficientes. Sua flexibilidade e robustez os tornam adequados para lidar com problemas não lineares e de alta dimensionalidade, características comuns em cenários reais de aplicação.

Este trabalho investiga a aplicação de algoritmos genéticos, incluindo a variante Biased Random-Key Genetic Algorithm (BRKGA), na seleção de características para otimizar modelos de aprendizado de máquina. A pesquisa busca avaliar a eficiência dessa abordagem na melhoria do desempenho preditivo dos modelos, além de explorar seus benefícios na redução da dimensionalidade e na mitigação de problemas como overfitting.

O presente estudo foi conduzido com base em um conjunto de dados sobre pacientes com doenças cardíacas, coletados desde 1988 (Janosi and Detrano (1989)). O conjunto

totaliza 303 pacientes, contendo em cada um 14 atributos, incluindo a variável-alvo. A saída do modelo assume valores binários, onde 1 representa pacientes diagnosticados com doença cardíaca e 0 indica pacientes saudáveis.

A proposta apresentada neste trabalho contribui significativamente para o avanço das técnicas de seleção de características, oferecendo uma abordagem robusta e escalável, passível de aplicação em contextos diversos que demandem análises preditivas. Além disso, o estudo estabelece uma base para futuras investigações relacionadas à integração de algoritmos genéticos com técnicas avançadas de aprendizado de máquina, visando aprimorar a eficiência e a precisão dos modelos desenvolvidos.

1.1 Objetivo Geral

O objetivo geral deste trabalho é melhorar o desempenho de modelos de aprendizado de máquina para classificação de doenças cardíacas, por meio da otimização da seleção de características do modelo.

1.1.1 Objetivos Específicos

1. Desenvolver e implementar modelos baseados em algoritmos genéticos para a seleção de características em conjuntos de dados utilizados por algoritmos de aprendizado de máquina;
2. Avaliar o desempenho dos modelos preditivos resultantes, considerando métricas como acurácia, precisão, *recall* e *F1-score*, após a aplicação dos algoritmos genéticos;
3. Comparar os resultados dos modelos otimizados com aqueles obtidos sem qualquer processo de seleção de características, destacando ganhos em eficiência computacional e capacidade preditiva;

1.2 Organização do Trabalho

Este trabalho está organizado em capítulos que descrevem detalhadamente as etapas seguidas durante a pesquisa e a modelagem de algoritmos genéticos aplicados à otimização de modelos de aprendizado de máquina. Além disso, são apresentados conceitos computacionais e definições matemáticas que fundamentam a abordagem proposta.

No Capítulo 2, são abordados os fundamentos teóricos relacionados ao problema da seleção de características, a teoria subjacente aos algoritmos genéticos, bem como uma descrição dos diferentes modelos de aprendizado de máquina utilizados, incluindo K-Nearest Neighbors (KNN), Árvores de Decisão e Regressão Logística. Também são discutidas as principais métricas empregadas na avaliação dos modelos.

O Capítulo 3 apresenta a modelagem dos algoritmos genéticos, detalhando os parâmetros utilizados e as adaptações realizadas para atender às especificidades do problema em questão.

O Capítulo 4 contempla a análise e a discussão dos principais resultados obtidos, com uma comparação sistemática entre as métricas dos modelos sem otimização e os resultados alcançados pelos modelos otimizados com algoritmos genéticos.

Por fim, o Capítulo 5 reúne as considerações finais acerca do desenvolvimento e dos resultados alcançados, além de discutir possíveis direções para trabalhos futuros envolvendo a aplicação de algoritmos genéticos na otimização de modelos de aprendizado de máquina.

Capítulo 2

Fundamentação Teórica

2.1 Revisão da Literatura

A seleção de características é uma etapa crucial no desenvolvimento de modelos de aprendizagem de máquina, especialmente quando se busca reduzir a dimensionalidade dos dados, melhorar o desempenho preditivo e diminuir o custo computacional (Guyon and Elisseeff, 2003). Métodos tradicionais de seleção, como algoritmos baseados em filtro, envolvem métricas estatísticas simples, enquanto métodos baseados em envoltória e incorporação tendem a considerar o desempenho do modelo como critério de seleção.

Entretanto, com o aumento da complexidade dos conjuntos de dados e da não linearidade das relações entre variáveis, abordagens baseadas em algoritmos evolutivos têm ganhado destaque. Dentre essas, os Algoritmos Genéticos (GAs) se destacam por sua capacidade de explorar grandes espaços de busca de maneira eficiente e adaptativa (Holland, 1975).

Os GAs inspiram-se nos processos de seleção natural para encontrar soluções aproximadas para problemas complexos de otimização. Cada solução é representada como um indivíduo em uma população, sendo evoluída ao longo das gerações por meio de operadores como seleção, cruzamento e mutação.

Em particular, os Algoritmos Genéticos com Chaves Aleatórias Viés (*Biased Random-Key Genetic Algorithms* — BRKGA) representam uma evolução dessa técnica. A ideia de utilizar chaves aleatórias foi inicialmente proposta por Gonçalves and Almeida (2005) e Ericsson et al. (2002), de forma independente, como forma de representar as soluções por vetores de números reais, permitindo uma aplicação mais flexível dos operadores genéticos.

Posteriormente, Tavares et al. (2013) formalizaram o BRKGA como uma abordagem estruturada, com viés no cruzamento, em que indivíduos mais aptos têm maior probabilidade de influenciar a próxima geração. Essa abordagem tem se mostrado eficaz em diversos problemas combinatórios e, mais recentemente, na seleção de características para modelos de aprendizagem de máquina.

A aplicação de BRKGA na seleção de características permite balancear a exploração e a intensificação na busca por subconjuntos ótimos de atributos, favorecendo a generalização dos modelos e evitando o sobreajuste. Diversos estudos têm demonstrado que métodos baseados em BRKGA superam abordagens tradicionais tanto em desempenho quanto em robustez (Silva et al., 2020).

Essa linha de pesquisa reforça a importância dos algoritmos genéticos como ferramentas promissoras para a otimização de modelos de aprendizagem de máquina, especialmente em contextos de alta dimensionalidade e interdependência entre variáveis.

2.2 Problema de Seleção de Características

De acordo com Oh (2004), o problema de seleção de características envolve a seleção de um subconjunto de d características de um total de D características, com base em um critério de otimização dado. A principal finalidade, é projetar um classificador compacto, no qual o desempenho é otimizado da melhor forma possível. Dessa forma, as características removidas devem ser redundantes ou de utilidade mínima.

A seleção de características é um processo essencial em aprendizado de máquina, pois influencia diretamente tanto a eficiência computacional quanto a precisão dos modelos preditivos. Um conjunto de características mal selecionado pode levar à inclusão de variáveis irrelevantes, que aumentam o risco de sobreajuste (overfitting) e impactam negativamente a generalização do modelo para novos dados. Por outro lado, a remoção de características relevantes pode resultar em subajuste (underfitting), reduzindo a capacidade do modelo de capturar padrões essenciais nos dados (Muhammad Salman Pathan (2017)).

Nesse contexto, um algoritmo de seleção de características é uma solução computacional motivada por uma determinada definição de relevância, contudo, a relevância de uma característica, sob a perspectiva do aprendizado indutivo, pode ter várias definições, dependendo do objetivo almejado (Molina (2002)).

Impacto no Desempenho

A redução do número de características tem implicações diretas no tempo de treinamento e no uso de memória, pois menos variáveis significam menos cálculos e menor complexidade computacional. Além disso, um subconjunto adequado de características pode melhorar a robustez do modelo, tornando-o menos suscetível a ruídos e variações nos dados (Muhammad Salman Pathan (2017)).

Impacto na Interpretabilidade

Além de impactar o desempenho dos modelos, a seleção de características desempenha um papel essencial na interpretabilidade das predições. Modelos que utilizam um grande

número de variáveis podem apresentar alta complexidade, tornando-se caixas-pretas e dificultando a compreensão dos critérios subjacentes às decisões tomadas. Nesse contexto, a escolha de um subconjunto otimizado de características permite reduzir a complexidade do modelo, ao mesmo tempo em que possibilita a identificação das variáveis mais relevantes para a tarefa preditiva. Dessa forma, a seleção adequada de características contribui para o aumento da transparência do modelo e facilita a extração de ideias sobre os fatores que influenciam os resultados obtidos, variando dependendo da problemática em questão (Molina (2002)).

2.3 Algoritmo Genético

Algoritmos Genéticos constituem uma técnica de busca e otimização, altamente paralela, inspirada no princípio Darwiniano de seleção natural e reprodução genética (Goldenberg (1989)). De acordo com Darwin (2014), o princípio de seleção privilegia os indivíduos mais aptos com maior longevidade e, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações, tais códigos genéticos constituem a identidade de cada indivíduo e estão representados nos cromossomos.

Estes princípios são reproduzidos na construção de algoritmos computacionais que buscam uma melhor solução para um determinado problema, através da evolução de populações de soluções codificadas através de cromossomos artificiais.

No contexto de GAs, um cromossomo é uma estrutura de dados que representa uma das possíveis soluções do espaço de busca do problema. Cromossomos são então submetidos a um processo evolucionário que envolve avaliação, seleção, recombinação sexual (cruzamento) e mutação. Sendo assim, após vários ciclos de evolução a população deverá conter indivíduos cada vez mais aptos (Goldenberg (1989)).

A analogia entre algoritmos genéticos e o sistema natural é representada através da tabela abaixo (Babatunde (2014)):

As etapas que constituem a estrutura de um algoritmo genético estão detalhadamente representadas na Figura 2.1. Essa representação ilustra o fluxo completo do processo evolutivo, desde a inicialização da população até a convergência para uma solução otimizada, destacando as operações fundamentais, como seleção, cruzamento e mutação, bem como os critérios para avaliação de aptidão e a geração sucessiva de populações.

Natureza	Algoritmos Genéticos
Cromossomo	Vetor binário
Gene	Característica do Problema
Alelo	Valor da característica
Lócus	Posição no Vetor binário
Genótipo	Estrutura
Fenótipo	Estrutura submetida ao problema
Indivíduo	Solução
Geração	Ciclo

Tabela 2.1: Relação entre a composição da Genética e Algoritmos Genéticos. Fonte: Autor, 2025.

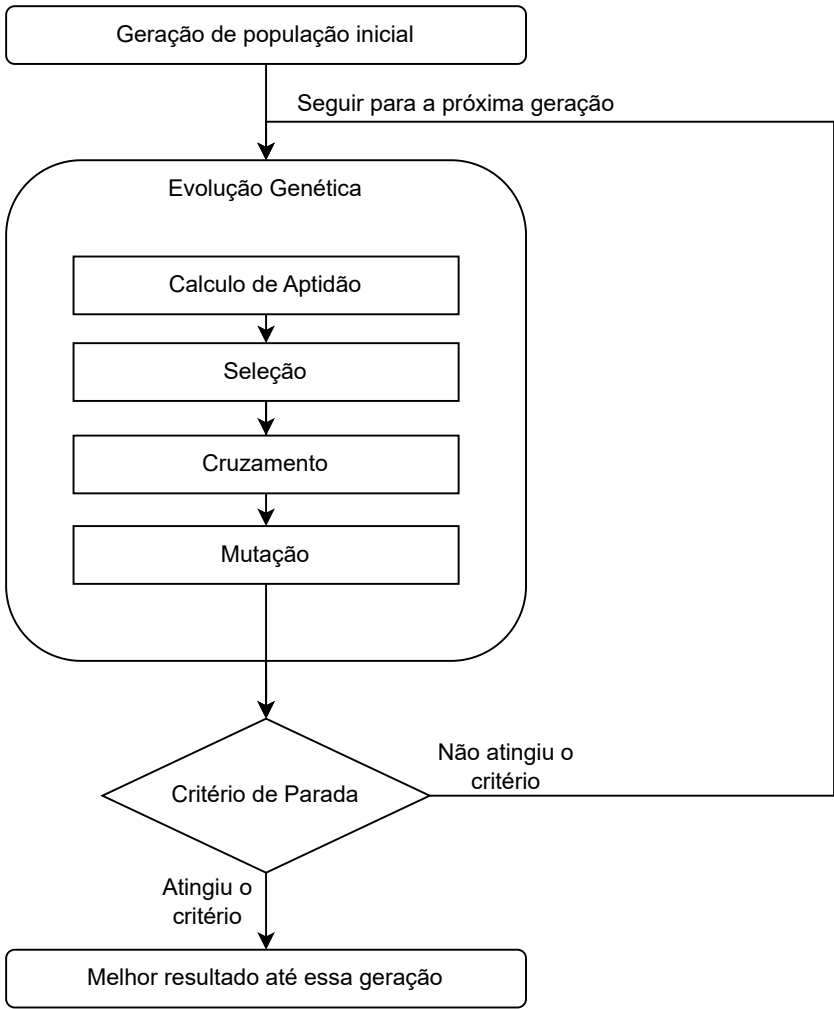


Figura 2.1: Etapas de um algoritmo gen tico. Fonte: Autor, 2025.

2.3.1 População Inicial

A população inicial de um algoritmo genético representa o conjunto de soluções candidatas geradas aleatoriamente ou de forma heurística no início do processo evolutivo (Kumar (2010)). Cada indivíduo nessa população é codificado como um cromossomo, que contém a informação genética necessária para definir uma solução particular. A qualidade dessa população inicial exerce um impacto significativo no desempenho do algoritmo, influenciando a velocidade de convergência e a capacidade de explorar o espaço de busca de forma eficiente.

De acordo com Alkafaween (2024), existem diversas estratégias para a geração da população inicial, sendo as mais comuns:

- **Geração aleatória:** Cada gene do cromossomo é inicializado aleatoriamente dentro de seu domínio possível. Essa abordagem garante uma grande diversidade inicial, mas pode levar a um tempo de convergência maior.
- **Geração heurística:** Utiliza-se conhecimento prévio sobre o problema para gerar soluções iniciais mais promissoras. Por exemplo, em problemas de otimização combinatória, a população inicial pode ser gerada a partir de heurísticas conhecidas.
- **Geração híbrida:** Combina a geração aleatória com a geração heurística, buscando um equilíbrio entre diversidade e qualidade das soluções iniciais.

2.3.2 Métodos de Seleção

Segundo Jebari (2013), a seleção é uma etapa fundamental nos algoritmos genéticos, responsável por simular o processo de seleção natural na evolução biológica. Nessa fase, indivíduos da população são escolhidos para gerar a próxima geração, com base em suas aptidões. A escolha do método de seleção adequado impacta diretamente a eficiência e a convergência do algoritmo, influenciando a exploração do espaço de busca e a pressão seletiva sobre a população.

Diversos métodos de seleção têm sido propostos na literatura, cada um com suas características e aplicações. A escolha do método mais adequado depende de diversos fatores, como o problema a ser resolvido, o tamanho da população, a desejada pressão seletiva e a necessidade de manter a diversidade genética.

Os métodos de seleção podem ser classificados em diversas categorias, como probabilísticos e determinísticos. Os métodos probabilísticos atribuem uma probabilidade de seleção a cada indivíduo, enquanto os métodos determinísticos selecionam um número fixo de indivíduos com base em um critério predefinido (Shukla (2013)).

Métodos Probabilísticos

- Seleção por roleta: é um dos métodos mais clássicos e intuitivos utilizados em algoritmos genéticos para determinar quais indivíduos passarão para a próxima geração. A ideia central é simular uma roleta de cassino, onde cada indivíduo ocupa um espaço proporcional à sua aptidão. Quanto maior a aptidão de um indivíduo, maior será o seu setor na roleta e, conseqüentemente, maiores as chances de ser selecionado.
- Seleção por torneio: é um método probabilístico amplamente utilizado em algoritmos genéticos para determinar quais indivíduos passarão para a próxima geração. Ao contrário da seleção por roleta, que atribui uma probabilidade de seleção proporcional à aptidão de cada indivíduo, a seleção por torneio realiza uma série de torneios entre indivíduos escolhidos aleatoriamente.
- Seleção por ranking: uma estratégia utilizada em algoritmos genéticos para determinar quais indivíduos de uma população passarão para a próxima geração. A seleção por ranking baseia-se na posição relativa dos indivíduos em um ranking de aptidão.

Métodos Determinísticos

- Seleção truncada: Uma população inicial diversificada garante que o algoritmo explore diferentes regiões do espaço de busca, aumentando as chances de encontrar a solução ótima.
- Seleção elitista: Uma população inicial bem escolhida pode acelerar a convergência do algoritmo, pois os indivíduos iniciais já podem conter informações relevantes sobre a solução.

2.3.3 Cruzamento

Segundo [Koohestani \(2020\)](#), o cruzamento é um operador genético fundamental nos algoritmos genéticos, responsável por promover a recombinação de informações genéticas entre indivíduos de uma população. Essa operação simula o processo de reprodução sexuada em organismos biológicos, onde a prole herda características de ambos os pais. O cruzamento consiste em selecionar dois indivíduos (pais) da população e combinar partes de seus cromossomos para gerar dois novos indivíduos (filhos). A região onde ocorre a troca de material genético é denominada ponto de cruzamento.

Tipos de Crossover

Existem diversas variantes de cruzamento, cada uma com suas próprias características e aplicações ([Umbarkar \(2015\)](#)), como por exemplo:

- Um ponto: Um único ponto de crossover é escolhido aleatoriamente e as partes dos cromossomos dos pais são trocadas a partir desse ponto.
- Dois pontos: Dois pontos de crossover são escolhidos aleatoriamente e o material genético entre esses pontos é trocado.
- Multiponto: Uma generalização do crossover de dois pontos, onde múltiplos pontos são utilizados.
- Uniforme: Cada gene dos filhos é escolhido aleatoriamente de um dos pais.
- Aritimético: Os valores dos genes dos filhos são calculados como uma combinação linear dos valores dos genes dos pais.

Dessa forma, o cruzamento desempenha um papel crucial na busca por soluções ótimas. Ao realizar a recombinação de informações genéticas entre indivíduos gerando novos com combinações de características herdadas de seus pais. Essa troca de material genético é essencial para a exploração eficiente do espaço de busca e a descoberta de soluções inovadoras.

2.3.4 Mutação

Segundo [Rani \(2013\)](#), a mutação em algoritmos genéticos, é um operador genético que introduz aleatoriamente pequenas alterações nos indivíduos de uma população. Essa operação simula as mutações genéticas que ocorrem na natureza e desempenha um papel crucial na manutenção da diversidade genética e na exploração do espaço de busca.

A mutação é aplicada a um indivíduo selecionado aleatoriamente da população. Essa seleção pode ser feita de forma uniforme ou com base em alguma probabilidade, como a probabilidade de mutação. Uma vez selecionado o indivíduo, um ou mais genes são alterados aleatoriamente. A forma como essa alteração é realizada depende da representação dos indivíduos e do tipo de mutação utilizado.

De acordo com a pesquisa de [Rares \(2015\)](#), a mutação desempenha um papel crucial na busca por soluções ótimas. A mutação é essencial para manter a diversidade genética, explorar regiões do espaço de busca que poderiam ser inacessíveis por outros operadores, evitar a convergência prematura para soluções subótimas e introduzir novas características que podem levar a soluções mais aptas.

2.3.5 Algoritmo Genético de Chaves Aleatórias Enviesadas (BRKGA)

A utilização de *chaves aleatórias* (*random keys*) como codificação em algoritmos genéticos foi proposta de forma independente por [Ericsson and Pardalos \(2002\)](#) e por [Gonçalves](#)

and Almeida (2002), visando representar soluções de forma contínua e permitir operadores genéticos mais eficientes. Posteriormente, essa abordagem foi estendida e formalizada por Tavares et al. (2013), que introduziram o *Biased Random-Key Genetic Algorithm* (BRKGA). O BRKGA diferencia-se dos algoritmos genéticos tradicionais por utilizar um processo de cruzamento enviesado, no qual a seleção de genes favorece sistematicamente indivíduos mais aptos, mantendo a simplicidade e eficiência da codificação por chaves aleatórias. Essa sistematização permite a aplicação eficaz do BRKGA em uma variedade de problemas de otimização combinatória.

Características do BRKGA

- **Representação com Chaves Aleatórias:** Em um BRKGA, cada cromossomo (ou indivíduo) é representado por um vetor de números reais no intervalo $[0, 1]$. Esses valores, chamados de "chaves aleatórias", são posteriormente mapeados para soluções do problema utilizando um procedimento de decodificação.
- **Decodificação das Soluções:** Um mecanismo específico é utilizado para interpretar as chaves aleatórias e convertê-las em soluções válidas para o problema em questão. Por exemplo, os valores podem determinar a ordem de priorização de tarefas ou a seleção de elementos em um conjunto.
- **Cruzamento com Viés:** Durante o processo de cruzamento, o BRKGA introduz um viés controlado, onde um dos pais (geralmente o mais apto) tem maior influência na geração do descendente. Esse viés é implementado garantindo que, em cada posição do cromossomo, o valor herdado tem maior probabilidade de vir do pai mais apto.

2.4 Aprendizado de Máquina

A inteligência artificial (IA) é um campo interdisciplinar que busca desenvolver sistemas capazes de simular a inteligência humana, permitindo que máquinas aprendam, raciocinem, tomem decisões e resolvam problemas (Russell and Norvig (2010)). O aprendizado de máquina (ML), por sua vez, é um subcampo da IA que se concentra no desenvolvimento de algoritmos e modelos estatísticos que permitem que os computadores aprendam com os dados, identificando padrões e fazendo previsões sem serem explicitamente programados para cada tarefa (Al-Sahaf (2019)).

2.4.1 Conceitos Fundamentais

Aprendizado Supervisionado

O aprendizado supervisionado, um dos paradigmas mais difundidos no campo do aprendizado de máquina, consiste em um processo de treinamento de modelos a par-

tir de conjuntos de dados rotulados. Esses dados contêm exemplos de entrada e suas respectivas saídas corretas, permitindo que o algoritmo aprenda a mapear novas entradas para as saídas correspondentes (Janiesch (2021)).

Os pares de dados de entrada e saída no conjunto de treinamento são então utilizados para calibrar os parâmetros ajustáveis do modelo de aprendizado de máquina. Após o treinamento bem-sucedido do modelo, ele pode ser usado para prever a variável alvo y com base em novos dados ou dados não vistos das características de entrada x .

Aprendizado Não Supervisionado

O aprendizado não supervisionado ocorre quando o sistema de aprendizado deve identificar padrões sem a presença de rótulos ou especificações pré-existentes. Dessa forma, os dados de treinamento consistem apenas nas variáveis x , com o objetivo de encontrar informações estruturais de interesse, como grupos de elementos que compartilham propriedades comuns (conhecidos como clustering) ou representações de dados projetadas de um espaço de alta dimensionalidade para um espaço de menor dimensão (conhecidas como redução de dimensionalidade) (Bishop (2006)).

Aprendizado por Reforço

Em um sistema de aprendizado por reforço, em vez de fornecer pares de entrada e saída, descreve-se o estado atual do sistema, especifica-se um objetivo, disponibiliza-se uma lista de ações permitidas e suas restrições ambientais, e permite-se que o modelo de aprendizado de máquina experimente o processo de alcançar o objetivo por conta própria, utilizando o princípio de tentativa e erro para maximizar uma recompensa (Janiesch (2021)).

2.4.2 K-Vizinhos Mais Próximos (KNN)

O algoritmo K-Vizinhos Mais Próximos (KNN), é uma das técnicas mais intuitivas e amplamente utilizadas no campo do aprendizado de máquina. Para classificar um novo ponto de dados, o KNN busca os exemplos mais próximos a ele no conjunto de treinamento e atribui a ele a classe mais frequente entre esses vizinhos (Zhang (2022)).

A aplicação do KNN envolve as seguintes etapas (Zhang (2016)):

- **Cálculo da Distância:** Para cada ponto de dados no conjunto de treinamento, calcula-se a distância até o novo ponto. A métrica de distância mais comum é a distância euclidiana.
- **Identificação dos K Vizinhos:** Os K pontos mais próximos são identificados com base nas distâncias calculadas.

- Votação: A classe do novo ponto é determinada por uma votação majoritária entre os K vizinhos mais próximos. No caso de regressão, a saída é a média dos valores dos K vizinhos mais próximos.

Uma das principais vantagens do KNN é sua simplicidade e facilidade de implementação. Além disso, não exige treinamento explícito, como a maioria dos outros algoritmos de aprendizado de máquina, o que o torna bastante eficiente em termos de tempo de processamento (Zhang (2022)). No entanto, o KNN pode se tornar computacionalmente caro para grandes conjuntos de dados, pois o cálculo da distância para todos os pontos de treinamento a cada nova predição pode ser demorado.

O KNN pode ser aplicado tanto a problemas de classificação quanto de regressão. Em problemas de classificação, o objetivo é prever a classe a qual um novo ponto de dados pertence, assim como pode ser visto na Figura 2.2. Em problemas de regressão, o objetivo é prever um valor numérico contínuo (Zhang (2022)).

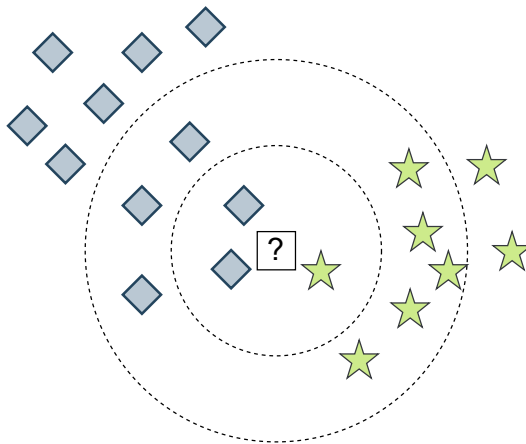


Figura 2.2: Como funciona um KNN classificador. Fonte: Autor, 2025.

2.4.3 Árvore de Decisão

A árvore de decisão, um algoritmo de aprendizado supervisionado, oferece uma representação gráfica e interpretável de um conjunto de regras de decisão. Essa estrutura hierárquica, similar a um fluxograma, divide os dados em subconjuntos cada vez menores, com o objetivo de criar um modelo que possa prever um valor alvo (yan Song (2015)).

A Figura 2.3 ilustra um modelo simples de árvore de decisão que inclui uma única variável-alvo binária Y (0 ou 1) e duas variáveis contínuas, x_1 e x_2 , que variam de 0 a 1.

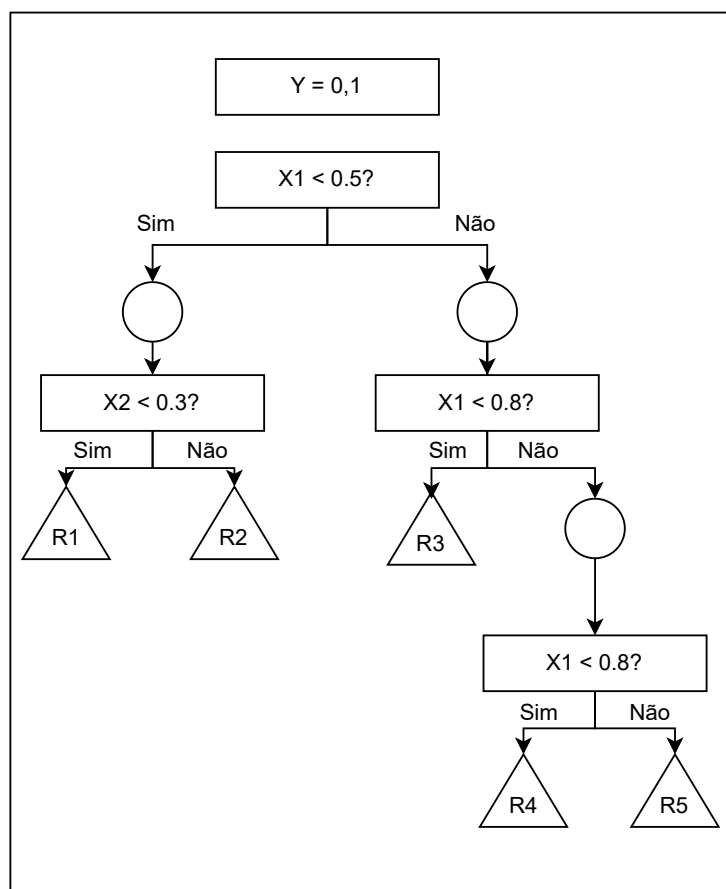


Figura 2.3: Exemplo de modelo simples de árvore de decisão. Fonte: Autor, 2025.

De acordo com o [U.S. National Library of Medicine \(2025\)](#), atualmente, a árvore de decisão esta presente entre os modelos mais utilizados na biomedicina, conforme evidenciado pelo número de publicações:

- 5,723 estudos relacionados à árvores de decisão
- 1,165 estudos relacionados à regressão logística
- 1,786 estudos relacionados à redes neurais
- 646 estudos relacionados à máquinas de vetores de suporte.

Conceitos Básicos

Uma árvore de decisão consiste em nós, que representam atributos ou características dos dados, e arestas, que conectam os nós e representam as decisões tomadas com base nos valores dos atributos ([yan Song \(2015\)](#)).

Nós. Existem três tipos de nós:

- Nó raiz, que representa uma escolha inicial que resulta na subdivisão de todos os registros em dois ou mais subconjuntos mutuamente exclusivos.
- Nós internos, que representam uma das possíveis escolhas disponíveis naquele ponto da estrutura da árvore; a aresta superior do nó está conectada ao nó pai, e a aresta inferior está conectada aos nós filhos ou aos nós folha.
- Nós folha, também chamados de nós finais, que representam o resultado final de uma combinação de decisões ou eventos.

Arestas. As arestas representam resultados ou ocorrências probabilísticas que emanam dos nós raiz e dos nós internos. Um modelo de árvore de decisão é formado por meio de uma hierarquia de arestas. Cada caminho, desde o nó raiz, passando pelos nós internos até um nó folha, representa uma regra de decisão de classificação. Esses caminhos da árvore de decisão também podem ser representados como regras do tipo "se-então".

2.4.4 Regressão Logística

A regressão logística, apesar de seu nome, é um algoritmo de classificação e não de regressão. Ela é utilizada para modelar a probabilidade de um evento ocorrer, dado um conjunto de variáveis preditoras. A saída da regressão logística é um valor entre 0 e 1, que representa a probabilidade da instância pertencer a uma determinada classe (Dreiseitl (2002)).

De acordo com Kurt (2008), a base da regressão logística está na função logística, também conhecida como função sigmoide. Essa função transforma qualquer valor real em um valor entre 0 e 1, o que a torna ideal para modelar probabilidades. A equação da função logística é dada por:

$$P = 1/(1 + e^{-(z)}) \quad (2.1)$$

Onde:

- P : Probabilidade da instância pertencer à classe positiva.
- z : Combinação linear das variáveis preditoras e seus respectivos coeficientes.

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p \quad (2.2)$$

- β_0 : Intercepto do modelo, representando o valor de z quando todas as variáveis independentes são iguais a 0 (Sperandei (2014)).
- $\beta_1, \beta_2, \dots, \beta_p$: Coeficientes associados às variáveis independentes que medem o impacto de cada variável na previsão do desfecho.

- X : Variáveis independentes (ou preditoras) usadas no modelo.

Por fim, o valor da probabilidade calculada é comparado a um limiar pré-definido. Caso a probabilidade exceda esse limiar, o modelo realiza a classificação binária atribuindo a classe 1; caso contrário, a observação é classificada como pertencente à classe 0 (figura 2.4).

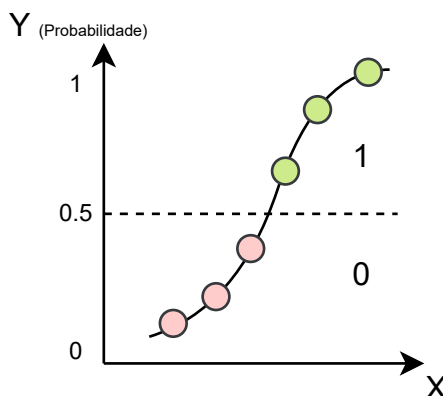


Figura 2.4: Funcionamento de uma regressão logística para problemas de classificação. Fonte: Autor, 2025.

O objetivo da regressão logística é encontrar os coeficientes que melhor ajustam a função logística aos dados de treinamento. Esses coeficientes são estimados utilizando técnicas de otimização, como o descenso do gradiente (Kurt (2008)).

2.4.5 Métricas

As métricas de aprendizado de máquina desempenham um papel essencial na avaliação do desempenho de modelos preditivos, ao possibilitar a quantificação da precisão, acurácia e confiabilidade desses modelos. Tais métricas fornecem informações críticas que subsidiam a tomada de decisões, permitindo a escolha do modelo mais adequado e a implementação de estratégias de otimização eficazes.

Essas métricas variam de acordo com o tipo de problema (classificação, regressão, etc.) e os objetivos específicos de cada aplicação. Em geral, elas são calculadas com base em uma matriz de confusão, que compara as previsões do modelo com os valores reais dos dados. A escolha da métrica adequada depende do contexto do problema e do custo relativo dos diferentes tipos de erros.

Matriz de confusão

A matriz de confusão constitui uma ferramenta indispensável na avaliação do desempenho de modelos de classificação, ao proporcionar uma visualização clara e estruturada do

comportamento do modelo na tarefa de predição. Essa tabela de contingência, também denominada tabela de erro, compara as classes previstas com as classes reais dos dados, permitindo identificar, de forma detalhada, os tipos de erros cometidos pelo modelo. Dessa forma, a matriz de confusão possibilita uma análise mais aprofundada e criteriosa do desempenho do modelo (Russell and Norvig (2010)).

A matriz de confusão é estruturada com base nos valores de verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN). Um verdadeiro positivo (VP) refere-se a um elemento positivo corretamente classificado como tal, enquanto um verdadeiro negativo (VN) corresponde a um elemento negativo corretamente identificado como negativo. Por outro lado, um falso positivo (FP) ocorre quando um elemento negativo é incorretamente (FN) ocorre quando um elemento positivo é erroneamente classificado como negativo (Google (2025)) (Figura 2.5).

		Classe Predita	
		Positivo	Negativo
Classe Real	Positivo	VP Verdadeiro Positivo	FN Falso Negativo
	Negativo	FP Falso Positivo	VN Verdadeiro Negativo

Figura 2.5: Estrutura de uma Matriz de Confusão. Fonte: Autor, 2025.

Partindo da ideia estabelecida acerca da matriz de confusão, algumas métricas podem ser geradas, algumas das principais métricas estabelecidas são:

- Acurácia: tem como objetivo principal estimar a proporção de classes corretamente classificadas. Para alcançar essa finalidade, realiza-se a divisão do número de acertos pelo total de classes, permitindo uma avaliação mais precisa do desempenho do modelo em relação às previsões feitas. (Google (2025)).

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN}$$

(2.3)

- Precisão: é uma métrica que representa a relação entre as previsões positivas realizadas corretamente e o total de previsões positivas, incluindo tanto os acertos

quanto os erros falsos. (Google (2025)).

$$Precisão = \frac{VP}{VP + FP} \quad (2.4)$$

- *Recall*: estabelece um paralelo entre o número de amostras classificadas corretamente como positivas e o total de amostras realmente positivas no conjunto de dados. Essa métrica mede a proporção de verdadeiros positivos (VP) identificados corretamente em relação ao número total de amostras positivas presentes. (Google (2025)).

$$Recall = \frac{VP}{VP + FN} \quad (2.5)$$

- *F1-Score*: é uma métrica que estabelece uma relação entre a precisão e o *recall* por meio da média harmônica de seus valores. Ele combina esses dois indicadores para proporcionar uma medida única que reflete o equilíbrio entre a capacidade de um modelo em prever corretamente os casos positivos (VP) e o quão bem ele recupera esses casos, considerando tanto os acertos quanto os erros. (Google (2025)).

$$F1-Score = \frac{Precisão \times Recall}{Precisão + Recall} \quad (2.6)$$

Capítulo 3

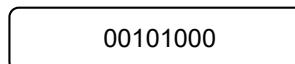
Metodologia

A metodologia proposta foi fundamentada em estudos previamente desenvolvidos e amplamente referenciados na literatura acadêmica, como os de [Oh \(2004\)](#) e [Babatunde \(2014\)](#). No entanto, foram implementadas modificações específicas, abrangendo ajustes em parâmetros, métricas utilizadas e no processo de desenvolvimento do algoritmo genético. Tais alterações foram realizadas com o objetivo de assegurar uma melhor adaptabilidade às características específicas da base de dados utilizada neste trabalho.

3.1 Algoritmo Genético

3.1.1 Codificação do Cromossomo

Para o problema de seleção de características, é utilizada uma sequência com G dígitos binários. Um dígito binário representa uma característica, sendo os valores 1 e 0 indicativos de seleção e remoção, respectivamente. Como exemplo, o cromossomo 00101000, assim como a Figura [3.1](#), significa que as terceira e quinta características estão selecionadas. Ou seja, o cromossomo representa $d = 3, 5$ e $R = 1, 2, 4, 6, 7, 8$. Sendo d o conjunto de índices das características selecionadas. Em outras palavras, d indica as posições dos dígitos binários que possuem o valor 1 no cromossomo. Enquanto que R representa o conjunto de índices das características não selecionadas. Ou seja, R indica as posições dos dígitos binários que possuem o valor 0 no cromossomo ([Oh \(2004\)](#)).



00101000

Figura 3.1: Exemplo de um cromossomo único. Fonte: Autor, 2025.

3.1.2 População Inicial

Uma população inicial de P indivíduos foi gerada aleatoriamente, codificando em cada gene, um subconjunto distinto de atributos. Para a geração de cada cromossomo dessa população, foi desenvolvido um algoritmo que selecionasse de maneira aleatória os bits de um número binário de G dígitos, assim como pode ser mostrado abaixo:

Algoritmo 1: Geração da População Inicial

Resultado: Gerar uma população com cromossomos aleatórios

```

1 for  $i \leftarrow 1$  até  $P$  faça do
2   for cada gene  $g$  no  $i$ -ésimo cromossomo do
3     if  $random.uniform() < 0.3$  then
4        $g \leftarrow 1$ 
5     else
6        $g \leftarrow 0$ 
7     end
8   end
9 end
```

No pseudo-código apresentado, P representa o número de cromossomos que compõem a população inicial gerada. Para cada gene presente nos cromossomos, é atribuída uma probabilidade de 30% de estar ativo. A função $random.uniform()$ é utilizada para gerar um número racional aleatório no intervalo $[0, 1]$. Com base nesse valor, verifica-se se o gene se encontra dentro do intervalo desejado, determinando sua inclusão no cromossomo. Ao fim do processo, teremos uma população inicial gerada de maneira aleatória, assim como pode ser visto na Figura 3.2

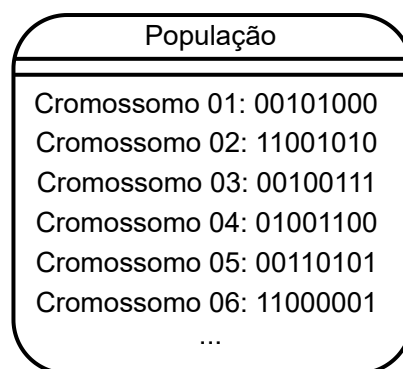


Figura 3.2: Exemplo de uma população inicial. Fonte: Autor, 2025.

3.1.3 Avaliação de Aptidão

A avaliação da aptidão dos cromossomos em algoritmos genéticos aplicados à classificação desempenha um papel fundamental na identificação de modelos de alto desempenho

(Melanie (1999)). Nesse trabalho, cada cromossomo, representando um conjunto específico de atributos, era utilizado para treinar os modelo de classificação, empregando exclusivamente as características selecionadas nos dados de treinamento. A qualidade desses modelos era mensurada, prioritariamente, por meio da métrica de *F1-Score*, cujo valor era diretamente empregado como critério para a avaliação da aptidão de cada cromossomo para aquele modelo (Figura 3.3).

População	Aptidão
Cromossomo 01: 00101000	0.45152
Cromossomo 02: 11001010	0.87524
Cromossomo 03: 00100111	0.65231
Cromossomo 04: 01001100	0.25638
Cromossomo 05: 00110101	0.54587
Cromossomo 06: 11000001	0.78452
...	...

Figura 3.3: Exemplo de uma população onde cada cromossomo tem seu valor de aptidão calculado utilizando o *recall* para um classificador específico. Fonte: Autor, 2025.

O *F1-Score* é uma métrica que mede o equilíbrio entre precisão e o *recall* em modelos de classificação sendo particularmente útil quando na base de dados há um desequilíbrio entre as classes.

A precisão mede a proporção de diagnósticos corretos de uma doença em relação ao total de diagnósticos positivos realizados pelo modelo. Ela avalia a capacidade do modelo de evitar falsos positivos, ou seja, situações em que um paciente saudável é incorretamente diagnosticado como doente.

O *recall*, mede a proporção de instâncias positivas (pacientes com a doença) corretamente identificadas pelo modelo. Em outras palavras, o *recall* quantifica a capacidade do modelo de encontrar todos os casos verdadeiros da classe positiva. Ao maximizar o *recall*, o objetivo era garantir que o maior número possível de pacientes com a doença fosse corretamente diagnosticado.

Após o treinamento de cada modelo, este era submetido a um conjunto de dados de teste, inédito durante o treinamento. As predições geradas pelo modelo eram comparadas com as classes verdadeiras dos exemplos de teste, permitindo a computação do *recall*. O valor do *F1-Score* obtido era então atribuído como a aptidão do cromossomo correspondente. Cromossomos que geravam modelos com maior *F1-Score* eram considerados soluções mais promissoras, indicando que o subconjunto de atributos selecionado era eficaz na identificação de padrões relevantes para a classificação.

Ao adotar o *F1-Score* como métrica de aptidão, o algoritmo genético foi direcionado a explorar o espaço de busca em busca de soluções que equilibrassem tanto a precisão quanto

a revocação. Essa estratégia mostrou-se particularmente adequada para o problema da detecção de doenças cardíacas, onde é fundamental não apenas identificar corretamente os pacientes positivos, mas também minimizar diagnósticos incorretos, garantindo uma avaliação robusta e confiável para a tomada de decisões clínicas e a melhoria dos resultados de tratamento.

3.1.4 Método de Seleção

Neste trabalho, foi adotado o elitismo como método de seleção. O elitismo é uma técnica amplamente utilizada em algoritmos genéticos (GA) que tem como objetivo preservar as melhores soluções de uma geração para a próxima, assegurando a retenção dos cromossomos com maior aptidão ao longo do processo evolutivo (Figura 3.4). Essa abordagem visa minimizar o risco de perda de informações relevantes que possam ocorrer devido a operações genéticas, como o cruzamento e a mutação, as quais podem introduzir variações indesejadas (Melanie (1999)). Dessa forma, adotou-se uma taxa de 20% dos cromossomos pertencentes a elite que irão ser preservados para a próxima geração.

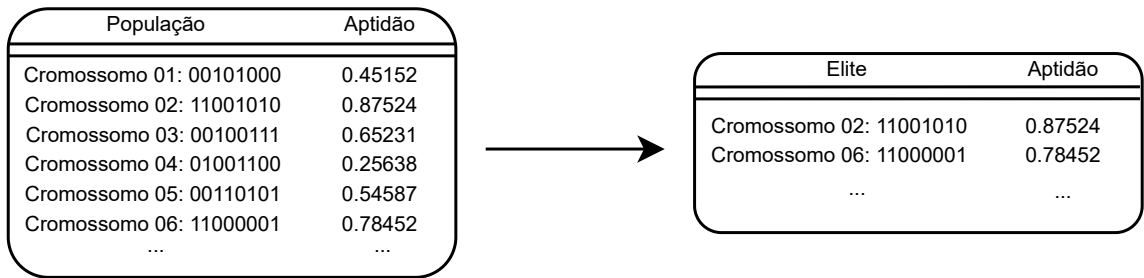


Figura 3.4: Cromossomos da elite que serão selecionados para próxima geração. Fonte: Autor, 2025.

3.1.5 Cruzamento e Mutação

Neste trabalho, foi utilizado o operador de cruzamento de um ponto para combinar características promissoras de diferentes soluções. Após a preservação da elite, os indivíduos restantes na nova população são gerados por meio de operações genéticas, como cruzamento e mutação, aplicadas à população atual. Dessa forma, dois cromossomos parentais são combinados de forma aleatória em um ponto de cruzamento selecionado ao longo dos bits do cromossomo, e as seções em cada lado desse ponto são trocadas (Rani (2013)). Considere, por exemplo, dois cromossomos, sendo um pertencente ao conjunto de elite e o outro à população geral, ambos compostos por 8 bits binários, conforme ilustrado na Figura 3.5. Após o processo de cruzamento, novos cromossomos, também denominados descendentes ou filhos, são gerados.

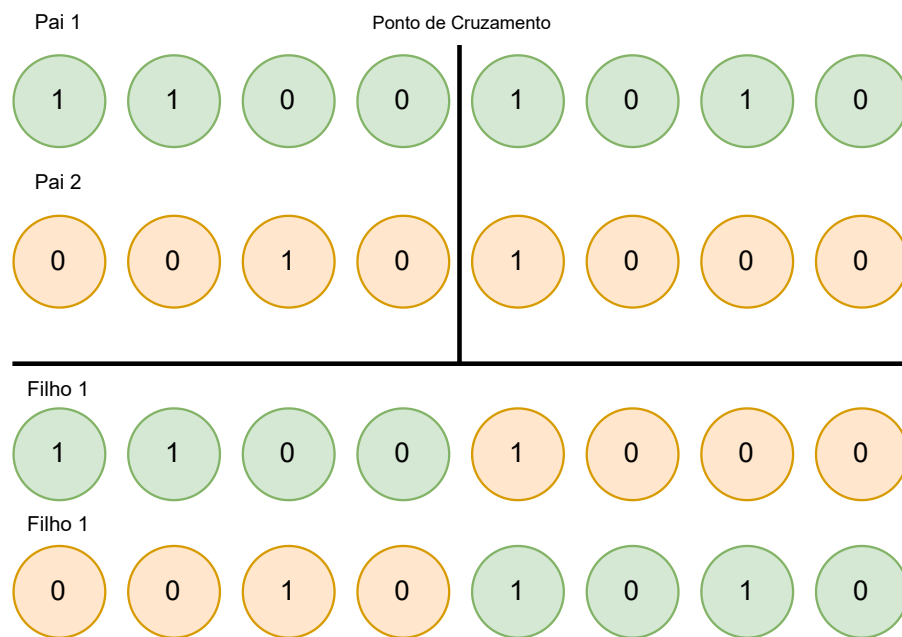


Figura 3.5: Cruzamento entre um pai pertencente a elite e um pai selecionado do restante da população. Fonte: Autor, 2025.

Após a inclusão dos cromossomos pertencentes à elite na nova população, o processo de cruzamento e mutação é realizado para preencher os 80% restantes da próxima geração. Por fim, aplica-se a mutação sobre os cromossomos gerados, promovendo a introdução de variabilidade genética. A mutação do algoritmo genético utilizada neste trabalho foi projetada para sistemas binários, na qual a operação é realizada em genes selecionados de forma aleatória com base em uma taxa de mutação pré-definida (Melanie (1999)). Durante o processo, os genes selecionados têm seus valores invertidos, alternando de 0 para 1 ou de 1 para 0, conforme apropriado (Figura 3.6). A taxa de mutação escolhida foi de 1% para cada gene, garantindo uma exploração suficiente do espaço de busca sem comprometer a convergência.

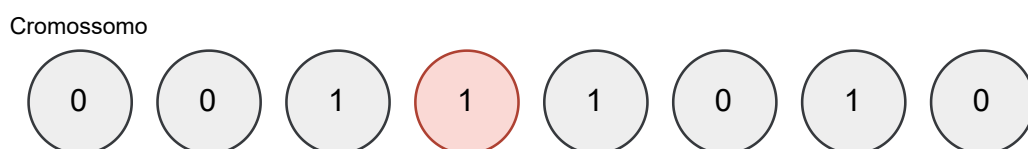


Figura 3.6: Mutação realizada nos genes de um cromossomo separado para a próxima geração. Fonte: Autor, 2025.

3.1.6 Algoritmo Genético de Chaves Aleatórias Enviesadas (BRKGA)

Além do modelo convencional de algoritmo genético, foi desenvolvido, com o objetivo de comparação de métricas, um modelo baseado no BRKGA, configurado como uma segunda variante do algoritmo genético.

Criação da População

Cada cromossomo na população é representado da maneira tradicional de um BRKGA (C. Martinez (2011) e Alixandre (2017)), como um vetor de números reais, normalmente pertencentes ao intervalo $[0,1]$. Neste estudo, a geração de cada gene do cromossomo foi realizada por meio de uma função responsável por produzir valores aleatórios dentro do referido intervalo. Um exemplo de cromossomo utilizando essa abordagem pode ser visto na Figura 3.7.



Figura 3.7: Exemplo de cromossomo aleatório gerado pelo BRKGA. Fonte: Autor, 2025.

Decodificação e Avaliação de Aptidão

O processo de decodificação adotado neste trabalho foi desenvolvido com base na necessidade de determinar se um gene seria considerado selecionado ou não na solução final. Para isso, utilizou-se um limiar de 0.7, de modo que todos os valores iguais ou superiores a 0.7 fossem classificados como presentes no cromossomo. Essa abordagem permitiu a identificação das características selecionadas para avaliação de aptidão, a qual era recalculada a cada iteração com a nova população. Nesse processo, o valor do *F1-Score* foi utilizado como métrica principal para avaliar a configuração, e a aptidão do cromossomo foi atualizada com base nesse valor.

Método de Seleção

O método de seleção empregado no BRKGA, assim como no algoritmo genético convencional, foi baseado no elitismo, garantindo a preservação dos cromossomos mais aptos para a próxima geração. Além disso, aplicou-se um cruzamento enviesado entre um pai pertencente a elite e um cromossomo não elitista para a geração de indivíduos para as próximas gerações, assim como pode ser visto na Figura 3.8.

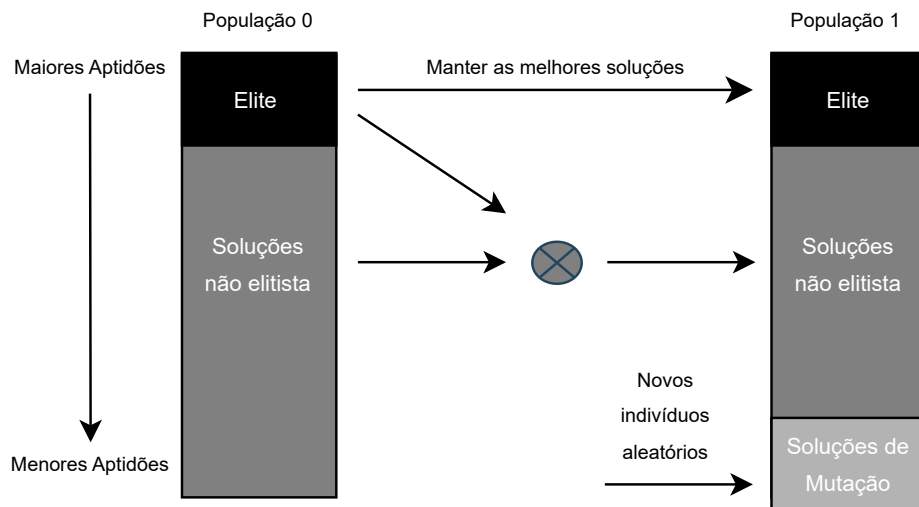


Figura 3.8: Processo de transição entre populações no BRKGA. Fonte: Autor, 2025.

Cruzamento Enviesado

No processo de cruzamento enviesado do modelo de BRKGA desenvolvido nesse trabalho, um dos pais foi selecionado do conjunto de indivíduos mais aptos, pertencente à elite, enquanto o outro pai é escolhido aleatoriamente da população completa. Os descendentes gerados por essa combinação recebem uma porção das características do pai elitista, o que promove a preservação das soluções de alta qualidade ao longo das gerações. Essa abordagem foi desenvolvida baseada no trabalho de [Alixandre \(2017\)](#) e busca equilibrar a exploração de boas soluções já existentes com a exploração de novas alternativas, aumentando as chances de convergência para resultados otimizados. Um exemplo pode ser visto na Figura 3.9.

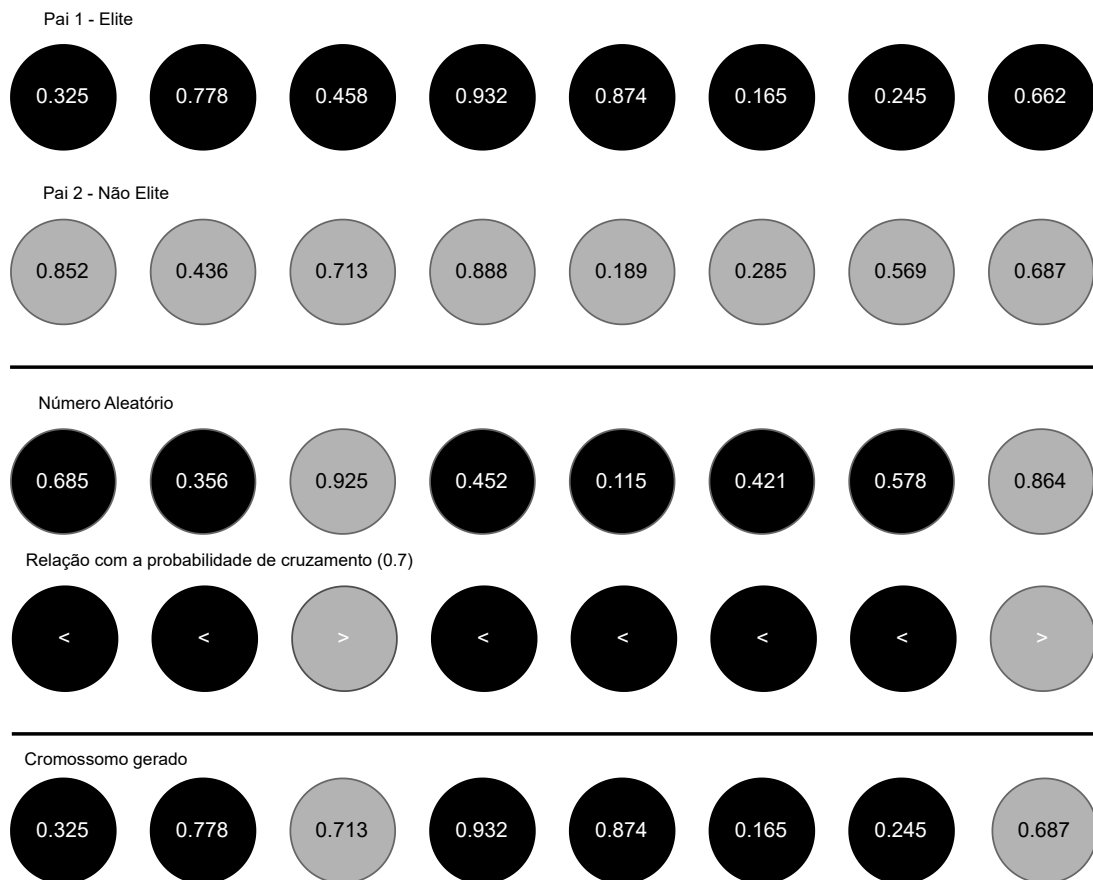


Figura 3.9: Processo de cruzamento entre cromossomos no BRKGA. Fonte: Autor, 2025.

Mutação

Além da inserção de indivíduos totalmente aleatórios por geração, para cada gene da população atual não elitista, foi aplicada uma taxa de mutação de 1% para a geração de alterações aleatórias. Caso um gene seja selecionado para realizar a mutação, um novo número real aleatório dentro do intervalo $[0, 1]$ é gerado, substituindo o valor original pelo novo número. Esse processo visa adicionar diversidade à população, evitando a convergência prematura e promovendo a exploração de diferentes regiões do espaço de soluções, além de compor o restante da próxima geração (Figura [3.10](#)).



Figura 3.10: Exemplo de mutação do cromossomo aleatório gerado pelo BRKGA. Fonte: Autor, 2025.

3.2 Aprendizado de Máquina

Neste estudo, a base de dados foi avaliada utilizando oito modelos distintos de aprendizado de máquina. No entanto, a otimização baseada no algoritmo genético foi implementada apenas em três desses modelos. A escolha desses modelos foi fundamentada na análise de desempenho inicial, priorizando a inclusão do modelo com o melhor resultado, o modelo com o pior desempenho e um modelo com desempenho intermediário. Para a obtenção desses valores de desempenho, dividiu-se os dados em 80% para a utilização de treinamento dos modelos e 20% para a realização dos testes, que consistiam na predição da saída do modelo e comparação com o valor real. Dessa forma, foram obtidos os valores da métrica *F1-Score*, utilizados para a composição da Tabela 3.1.

A métrica *F1-score* foi selecionada por proporcionar um equilíbrio entre *precision* e *recall*, garantindo uma avaliação mais robusta do desempenho do modelo. Considerando que a presente problemática está inserida no contexto da área da saúde, a escolha dessa métrica se justifica pela necessidade de minimizar tanto os falsos negativos, que poderiam resultar em diagnósticos incorretos e atrasos no tratamento, quanto os falsos positivos, que podem gerar exames e procedimentos desnecessários, otimizando, assim, a confiabilidade das predições.

Modelo	<i>F1-Score</i>
Regressão Logística	0.885246
Máquina de Vetores de Suporte Linear	0.868852
Floresta Aleatória	0.852459
Impulso Adaptativo	0.803279
Árvore de Decisão	0.786885
Impulso por Gradiente	0.770492
Máquina de Vetores de Suporte com Função de Base Radial	0.704918
K-Vizinhos Mais Próximos	0.688525

Tabela 3.1: Desempenho dos modelos a base de dados antes da otimização.

Desas forma, os modelos escolhidos para aplicar a otimização foram:

- **Regressão Logística:** Como o modelo que apresentou os melhores resultados antes da otimização.

- **Árvore de Decisão:** Como o modelo que apresentou resultados medianos antes da otimização.
- **K-Vizinhos Mais Próximo:** Como o modelo que apresentou o pior resultado dentre os modelos antes da otimização.

Capítulo 4

Resultados e Discussões

Diante dos modelos de algoritmo genéticos propostos, nesta seção, são apresentados e discutidos os resultados obtidos com a aplicação das técnicas de otimização de seleção de características utilizando os algoritmos genéticos convencionais (GA) e o algoritmo genético de chaves aleatórias enviesadas (BRKGA). Os experimentos foram realizados com o objetivo de avaliar a eficácia dessas técnicas na melhoria do desempenho de modelos de aprendizado de máquina, considerando métricas como *F1-Score*, que leva em consideração o *recall* e a precisão.

Inicialmente, foram analisados os três modelos de aprendizado de máquina sem qualquer técnica de seleção de características, visando estabelecer uma linha de base para comparação. Em seguida, os resultados obtidos com os algoritmos de otimização foram contrastados com a linha de base, destacando os impactos na redução da dimensionalidade, no desempenho preditivo dos modelos e no tempo de processamento.

As configurações adotadas para os algoritmos genéticos estão apresentadas na Tabela 4.1. Parte desses parâmetros, incluindo o tamanho da população, a taxa de mutação, a taxa de elitismo e o método de seleção, foram definidos com base no estudo de Oh (2004). A taxa de viés, por sua vez, foi estabelecida conforme as diretrizes propostas por Alexandre (2017). Os demais valores, como o número de gerações e o número de execuções, foram determinados empiricamente, com base na obtenção de resultados considerados satisfatórios pelo autor.

Parâmetro	GA	BRKGA
Número de Gerações	10	10
Número de Execuções	50	50
Tamanho da População	100	100
Taxa de Elitismo	20%	20%
Taxa de Mutação	10%	10%
Método de Seleção	Elitismo	Elitismo
Avaliação de Aptidão	F1-Score	F1-Score
Taxa de Viés	—	0.7

Tabela 4.1: Tabela de parâmetros utilizados em cada algoritmo genético. Fonte: Autor, 2025.

4.1 K-Vizinhos mais próximos (KNN)

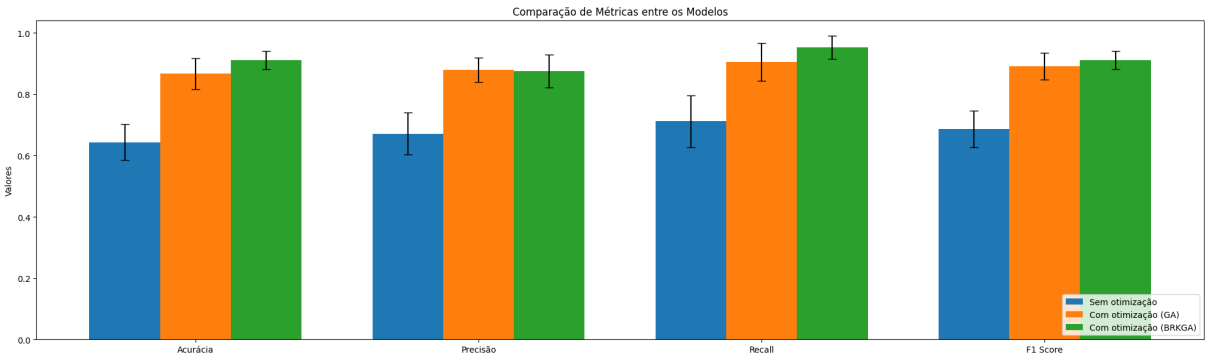


Figura 4.1: Comparação de Métricas (KNN). Fonte: Autor, 2025.

A Figura 4.1 apresenta o gráfico de barras que compara o desempenho do modelo em quatro métricas distintas (Acurácia, Precisão, Recall e F1-Score) sob três condições: sem otimização, com otimização utilizando um algoritmo genético (GA) e com otimização utilizando um algoritmo genético com chaves aleatórias (BRKGA). Os resultados obtidos na comparação podem ser vistos detalhadamente na Tabela 4.2.

Observa-se que a otimização resultou em melhorias expressivas em todas as métricas analisadas. O *F1-Score*, métrica utilizada como critério de aptidão, apresentou um aumento significativo, passando de 0,66 no modelo não otimizado para 0,87 com GA e 0,89 com BRKGA, representando incrementos de 31,82% e 34,85%, respectivamente.

A acurácia do modelo também foi aprimorada, elevando-se de 0,63 para 0,86 com GA e 0,90 com BRKGA, o que representa um incremento de 36,51% e 42,86%, respectivamente. Adicionalmente, as métricas de precisão e *recall* demonstraram ganhos consistentes. A precisão aumentou de 0,65 para 0,87 com GA e 0,86 com BRKGA, enquanto o *recall* passou de 0,70 para 0,90 com GA e 0,93 com BRKGA, indicando que a otimização proporcionou

um modelo mais equilibrado entre a minimização de falsos negativos e falsos positivos.

Ao comparar os algoritmos empregados, verifica-se que para o KNN, o BRKGA obteve um desempenho ligeiramente superior ao GA na otimização da métrica *F1-Score*, apresentando também melhores resultados na acurácia e *recall*. Mesmo assim, ambos os algoritmos obtiveram resultado de otimização extremamente satisfatório, corroborando para a eficácia dos algoritmos genéticos na otimização de modelos de aprendizado de máquina por meio da seleção de características.

Métrica	Não otimizado	GA	BRKGA	Porcentagem de Melhora	
				GA	BRKGA
Acurácia	0.63	0.86	0.90	(+36.51%)	(+42.86%)
Precisão	0.65	0.87	0.86	(+33.85%)	(+32.31%)
Recall	0.70	0.90	0.93	(+28.57%)	(+32.86%)
F1-Score	0.66	0.87	0.89	(+31.82%)	(+34.85%)

Tabela 4.2: Desempenho médio do modelo (KNN) não otimizado e otimizado para cada métrica. Fonte: Autor, 2025.

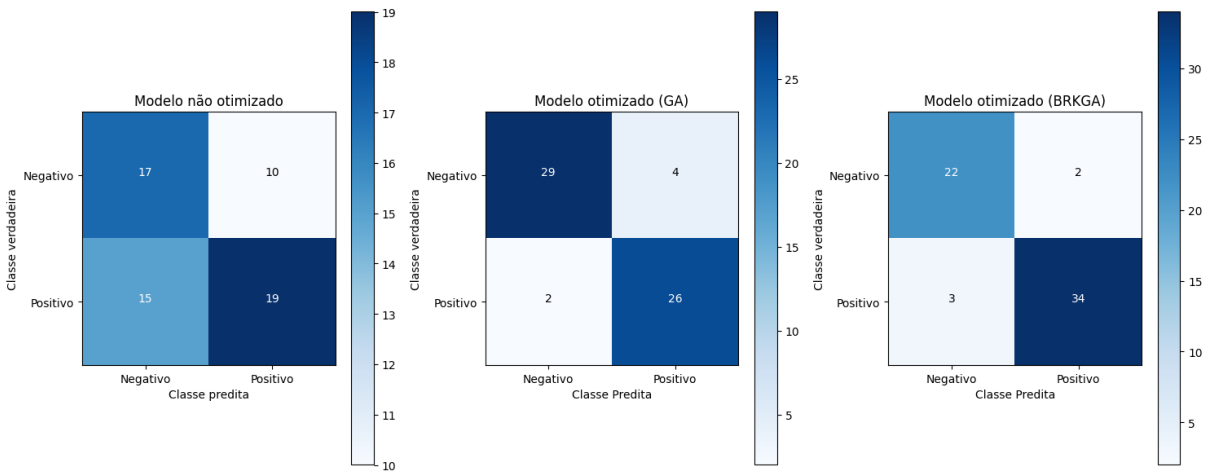


Figura 4.2: Matriz de confusão em caso médio (KNN). Fonte: Autor, 2025.

A Figura 4.2 apresenta as matrizes de confusão correspondentes ao desempenho do modelo K-Vizinhos Mais Próximos (KNN) nas versões não otimizada, otimizada com Algoritmo Genético (GA) e otimizada com Biased Random-Key Genetic Algorithm (BRKGA) a partir de um caso médio. No cenário sem otimização, observa-se uma taxa elevada de erros, com 10 falsos positivos e 15 falsos negativos, totalizando 25 classificações incorretas. Após a aplicação da otimização via GA, verifica-se uma expressiva redução nos erros, com apenas 4 falsos positivos e 2 falsos negativos, o que representa uma melhora significativa na capacidade do modelo em distinguir corretamente ambas as classes. A abordagem

otimizada com BRKGA apresentou o melhor desempenho entre as três configurações, reduzindo os falsos positivos para 2 e os falsos negativos para apenas 3, resultando em uma performance mais equilibrada e precisa.

Treinamento (Não otimizado)	Treinamento (Otimizado)	Porcentagem de Melhora
0.7261 segundos	0.5131 segundos	(-29,33%)

Tabela 4.3: Comparação entre os tempos médios de execução de um modelo não otimizado para um otimizado (KNN). Fonte: Autor, 2025.

A Tabela 4.3 apresenta a comparação entre os tempos médios de treinamento do modelo K-Vizinhos Mais Próximos (KNN), antes e após a aplicação da otimização por seleção de características. Observa-se uma redução de 0,7261 segundos para 0,5131 segundos, o que equivale a uma melhora percentual de 29,33% na eficiência computacional. Embora a diferença absoluta em tempo possa parecer modesta, o impacto torna-se expressivo em contextos envolvendo grandes volumes de dados, nos quais o custo computacional é significativamente ampliado. Assim, os resultados reforçam que a aplicação de métodos de seleção de características não apenas aprimora a performance preditiva, mas também torna o treinamento mais eficiente em termos de tempo, especialmente em modelos sensíveis à dimensionalidade, como o KNN.

4.2 Árvore de Decisão

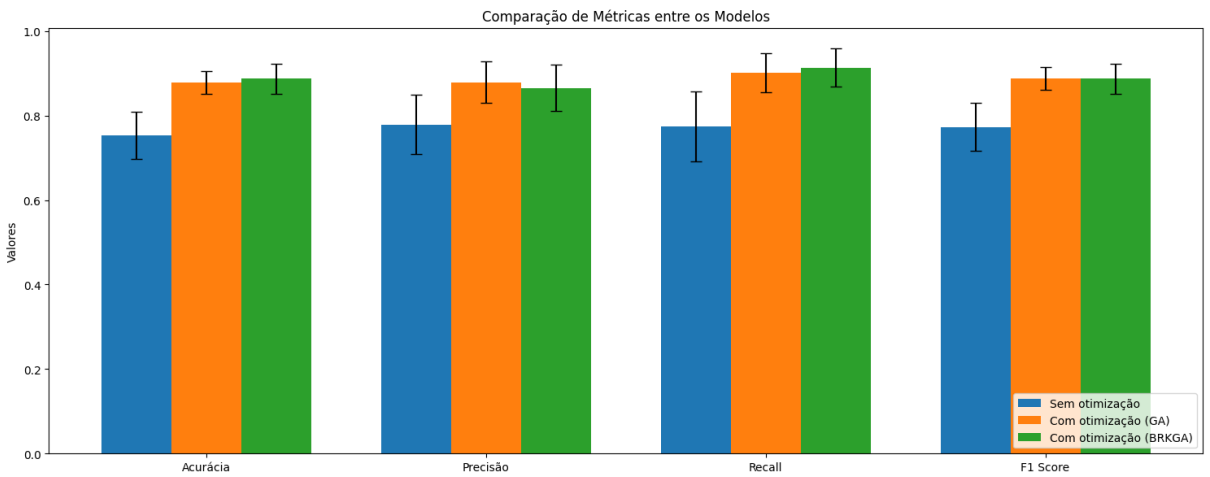


Figura 4.3: Comparação de Métricas (Árvore de Decisão). Fonte: Autor, 2025.

A Figura 4.3 apresenta os resultados do modelo de Árvore de Decisão antes e após a aplicação dos algoritmos genéticos. Esses resultados podem ser analisados de forma detalhada na Tabela 4.4. Segundo os resultados obtidos, na árvore de decisão, os ganhos

obtidos foram significativos, mas com uma diferença relevante em relação ao KNN: ambos os algoritmos genéticos apresentaram desempenhos semelhantes.

O *F1-Score* apresentou um aumento menor do que o modelo KNN, porém ainda expressivo, passando de 0,75 no modelo sem otimização para 0,85 após a aplicação dos algoritmos genéticos GA e BRKGA, representando um acréscimo de 7,59% em ambos os casos no critério otimizado. A acurácia também foi significativamente aprimorada, elevando-se de 0,76 para 0,86 com GA e 0,87 com BRKGA, o que corresponde a um aumento de 13,16% e 14,47%, respectivamente.

As métricas de precisão e *recall* também apresentaram ganhos, embora mais modestos quando comparados ao modelo KNN. A precisão evoluiu de 0,78 para 0,87 com GA e 0,86 com BRKGA, enquanto o *recall* subiu de 0,78 para 0,90 com GA e 0,91 com BRKGA. Também contribuindo para tornar o modelo mais equilibrado no que se refere à redução de falsos positivos e falsos negativos.

A melhoria conjunta entre precisão e *recall* evidencia que a seleção de características auxiliou na construção de um modelo mais estável e confiável, embora os impactos não tenham sido tão pronunciados quanto os observados no KNN. Além disso, o desempenho semelhante de GA e BRKGA sugere que o modelo de Árvore de Decisão já dispunha de uma estrutura sólida para a tarefa de classificação, exigindo ajustes mais sutis para alcançar seu desempenho ideal.

Métrica	Não otimizado	GA	BRKGA	Porcentagem de Melhora	
				GA	BRKGA
Acurácia	0.76	0.86	0.87	(+13.16%)	(+14.47%)
Precisão	0.78	0.87	0.86	(+11.54%)	(+10.26%)
Recall	0.78	0.90	0.91	(+15.38%)	(+16.67%)
F1-Score	0.77	0.85	0.85	(+7.59%)	(+7.59%)

Tabela 4.4: Desempenho médio do modelo (Árvore de Decisão) não otimizado e otimizado para cada métrica. Fonte: Autor, 2025.

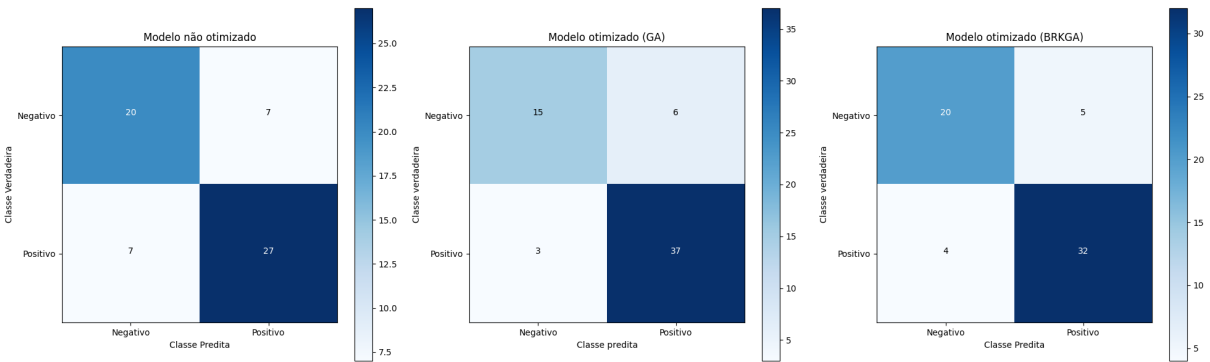


Figura 4.4: Matriz de confusão em caso médio (Árvore de Decisão). Fonte: Autor, 2025.

A Figura 4.4 ilustra as matrizes de confusão resultantes da aplicação do modelo de Árvore de Decisão nas três configurações avaliadas. Na versão não otimizada, o modelo cometeu um total de 14 erros, divididos igualmente entre falsos positivos e falsos negativos (7 cada). Com a aplicação da otimização por meio de algoritmos genéticos (GA), observou-se uma melhoria considerável, com a redução dos falsos positivos para 3 e uma leve diminuição nos falsos negativos, que passaram a 6, totalizando 9 erros. A configuração otimizada com BRKGA apresentou desempenho semelhante, com 9 erros no total, sendo 5 falsos positivos e 4 falsos negativos, indicando uma ligeira redistribuição dos erros, mas mantendo o mesmo nível de acurácia geral em relação ao GA.

Treinamento (Não otimizado)	Treinamento (Otimizado)	Porcentagem de Melhora
0.2252 segundos	0.1804 segundos	(-19,89%).

Tabela 4.5: Comparação entre os tempos médios de execução de um modelo não otimizado para um otimizado (Árvore de Decisão). Fonte: Autor, 2025.

A Tabela 4.5 ilustra a comparação entre os tempos médios de execução do modelo de Árvore de Decisão antes e após o processo de otimização por seleção de características. Nota-se uma diminuição no tempo de treinamento, que caiu de 0,2252 segundos para 0,1804 segundos, o que corresponde a uma melhora de 19,89% na performance temporal. Esse ganho, assim como o observado no modelo KNN, embora modesto, reforça que a aplicação da otimização também promove ganhos computacionais em modelos de menor complexidade, como a Árvore de Decisão, contribuindo para uma execução mais eficiente.

4.3 Regressão Logística

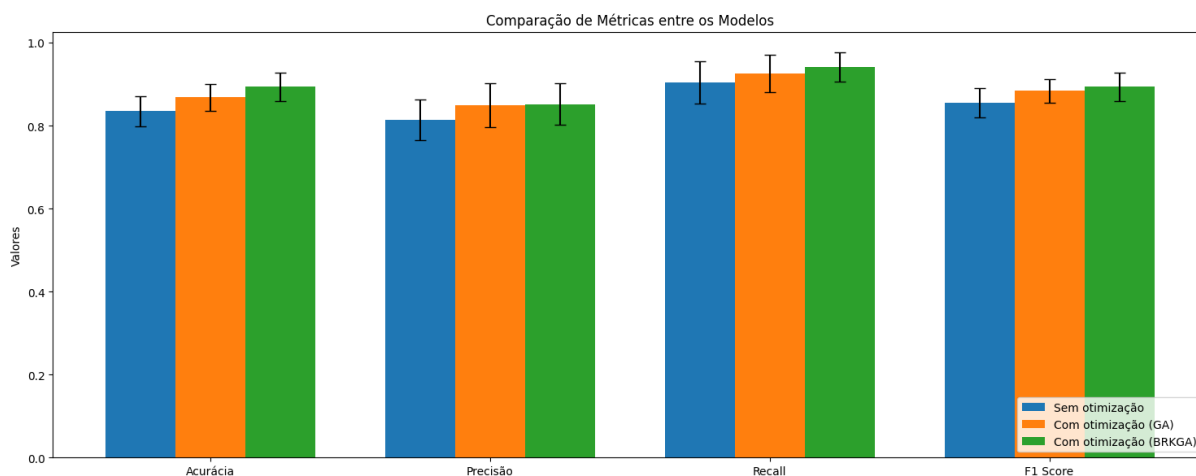


Figura 4.5: Comparação de Métricas (Regressão Logística) Fonte: Autor, 2025.

A Figura 4.5 que os dados podem ser visualizados na Tabela 4.6 mostra que a Regressão Logística apresentou os menores ganhos com a aplicação dos algoritmos genéticos. O *F1-Score*, por exemplo, teve um aumento relativamente modesto, passando de 0,85 no modelo não otimizado para 0,87 com GA e 0,88 com BRKGA, o que representa incrementos de apenas 2,35% e 3,53%, respectivamente. Esses valores estão significativamente abaixo dos observados nos demais modelos, evidenciando um impacto limitado da otimização por seleção de características nesse caso.

De maneira semelhante, a acurácia do modelo aumentou de 0,82 para 0,85 com GA e 0,87 com BRKGA, correspondendo a incrementos de 3,66% e 6,10%. As métricas de precisão e *recall* também apresentaram ganhos discretos. A precisão subiu de 0,80 para 0,83 em ambos os algoritmos, representando uma melhoria de 3,75%. Já o *recall* aumentou de 0,90 para 0,92 com GA e 0,93 com BRKGA, refletindo incrementos de 2,22% e 3,33%. Esses resultados indicam que a Regressão Logística já estava bem ajustada à tarefa de classificação, e que a aplicação de algoritmos genéticos, embora útil, teve um efeito mais limitado nesse modelo em comparação com os demais.

Métrica	Não otimizado	GA	BRKGA	Porcentagem de Melhora	
				GA	BRKGA
Acurácia	0.82	0.85	0.87	(+3.66%)	(+6.10%)
Precisão	0.80	0.83	0.83	(+3.75%)	(+3.75%)
Recall	0.90	0.92	0.93	(+2.22%)	(+3.33%)
F1-Score	0.85	0.87	0.88	(+2.35%)	(+3.53%)

Tabela 4.6: Desempenho médio do modelo (Regressão Logística) não otimizado e otimizado para cada métrica. Fonte: Autor, 2025.

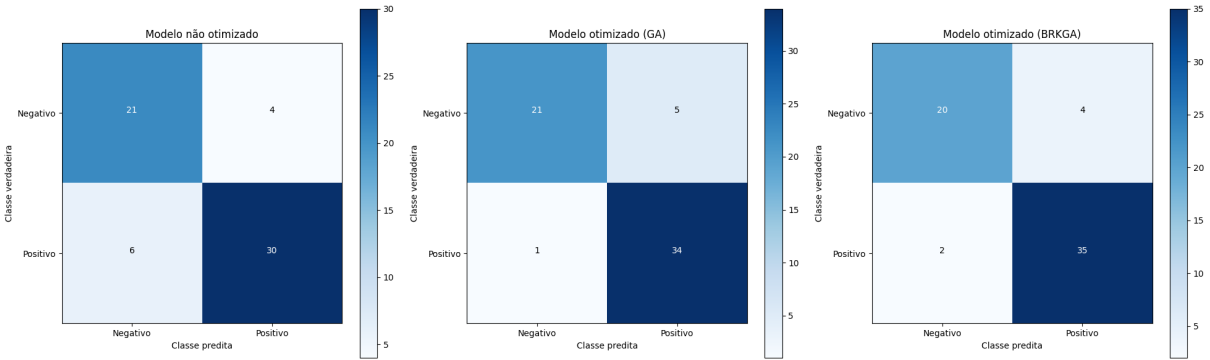


Figura 4.6: Matriz de confusão em caso médio (Regressão Logística) . Fonte: Autor, 2025.

A Figura 4.6 apresenta as matrizes de confusão obtidas pelo modelo de Regressão Logística nas três configurações avaliadas. Na versão não otimizada, o modelo já demonstrou desempenho consistente, com 4 falsos positivos e 6 falsos negativos. A aplicação do algoritmo genético (GA) resultou em uma melhora expressiva na redução de falsos negativos, que caíram para apenas 1, embora tenha havido um pequeno aumento nos falsos positivos, que passaram para 5 — mantendo, ainda assim, o total de erros em 6. De forma semelhante, a otimização com BRKGA obteve desempenho comparável, com 4 falsos positivos e 2 falsos negativos, também totalizando 6 erros. Esses resultados indicam que a Regressão Logística, mesmo sem otimização, já é eficaz na base analisada, e que as abordagens de otimização contribuem para refinar ainda mais sua capacidade de discriminação.

Treinamento (Não otimizado)	Treinamento (Otimizado)	Porcentagem de Melhora
7.5595 segundos	1.6462 segundos	(-78,22%)

Tabela 4.7: Comparação entre os tempos médios de execução de um modelo não otimizado para um otimizado (Regressão Logística). Fonte: Autor, 2025.

Por fim, a Tabela 4.7 expõe o tempo médio de treinamento do modelo de Regressão

Logística antes e após a aplicação da seleção de características via algoritmos genéticos. O tempo necessário para o ajuste do modelo foi reduzido de 7,56 segundos para 1,65 segundos, correspondendo a uma diminuição de 78,22% no custo computacional. Esse resultado evidencia que a Regressão Logística foi particularmente beneficiada pela redução no número de atributos, uma vez que seu tempo de treinamento está diretamente associado ao número de variáveis e à necessidade de ajustes iterativos dos coeficientes. Com menos características a serem processadas, o algoritmo converge mais rapidamente, o que justifica a expressiva diminuição no tempo de execução observada após a aplicação da técnica de seleção.

4.4 Comparação entre modelos

Com a finalização do processo de otimização dos modelos de aprendizado de máquina por meio de algoritmos genéticos, foi possível atualizar a tabela apresentada na Seção 3.2 incorporando os novos resultados obtidos. A Tabela 4.8 evidencia que todos os modelos analisados apresentaram melhora no *F1-Score* após a aplicação das técnicas de otimização, o que confirma a eficácia da seleção de características na elevação do desempenho preditivo. Dentre os classificadores, o modelo K-Vizinhos Mais Próximos (KNN) foi o que mais se beneficiou do processo, alcançando um *F1-Score* de 0,893 com BRKGA, um incremento significativo frente à sua versão original.

A comparação entre os métodos de otimização revela que o BRKGA apresentou um desempenho superior ao GA em dois dos três modelos otimizados com ambos os algoritmos, embora a diferença entre os resultados seja, em sua maioria, discreta. Esses achados sugerem que a efetividade do algoritmo de otimização depende da natureza do classificador e de sua sensibilidade à composição do conjunto de atributos selecionados.

No caso do KNN, o impacto da seleção de características foi particularmente acentuado devido à forma como esse modelo opera. Por ser um algoritmo baseado em distância, o KNN é altamente sensível à presença de atributos irrelevantes ou redundantes, que podem distorcer a medida de similaridade entre as instâncias. Ao reduzir o conjunto de características às mais relevantes, os algoritmos genéticos contribuíram para melhorar a representatividade do espaço amostral, resultando em classificações mais precisas. Esse ajuste fino no espaço de atributos favoreceu diretamente o desempenho do KNN, explicando o expressivo ganho observado após a otimização. Por outro lado, modelos como a Regressão Logística já demonstravam um bom desempenho antes da otimização, e os ganhos observados, embora positivos, foram mais modestos. De forma geral, a aplicação de algoritmos genéticos demonstrou ser uma abordagem válida e eficaz para aprimorar o desempenho dos classificadores, sendo particularmente vantajosa para modelos baseados em vizinhança e regressão linear.

Modelo	<i>F1-Score</i>
K-Vizinhos Mais Próximos com otimização (BRKGA)	0.893232
Regressão Logística com otimização (BRKGA)	0.884352
Regressão Logística com otimização (GA)	0.878657
K-Vizinhos Mais Próximos com otimização (GA)	0.874693
Árvore de Decisão com otimização (GA)	0.856435
Regressão Logística	0.855246
Árvore de Decisão com otimização (BRKGA)	0.853212
Máquina de Vetores de Suporte Linear	0.848852
Floresta Aleatória	0.832459
Impulso Adaptativo	0.783279
Árvore de Decisão	0.766885
Impulso por Gradiente	0.750492
Máquina de Vetores de Suporte com Função de Base Radial	0.684918
K-Vizinhos Mais Próximos	0.668525

Tabela 4.8: Desempenho dos modelos a base de dados após a otimização.

4.4.1 Eficiência Computacional

Observou-se um aumento na eficiência computacional tanto no treinamento quanto na predição dos modelos após a aplicação da seleção otimizada de características. Esse ganho ocorre devido à eliminação de atributos irrelevantes, permitindo que o modelo processe um volume reduzido de dados, o que acelera significativamente as etapas de leitura e aprendizado. As Tabelas 4.3, 4.5 e 4.7 apresenta os tempos obtidos e a porcentagem de melhoria alcançada, sendo construídas a partir do cálculo do tempo médio de treinamento do modelo em 50 execuções. Inicialmente, as medições foram realizadas sem a aplicação de seleção de características. Em seguida, o tempo médio foi obtido considerando a utilização de um cromossomo representando o subconjunto otimizado de características selecionadas pelos algoritmos genéticos.

A análise comparativa entre os modelos otimizados indica que a economia de tempo de execução gerada pela seleção de características está ligada à maneira como cada algoritmo lida com a dimensionalidade dos dados e à sua complexidade interna. A regressão logística, por depender fortemente de operações matriciais que crescem com o número de atributos, foi a que mais se beneficiou da otimização, com uma redução significativa no tempo de treinamento (-78,22%). Já os modelos KNN e Árvore de Decisão, embora também tenham apresentado melhorias, tiveram ganhos mais discretos (-29,33% e -19,89% respectivamente), porém de impacto considerável para grandes volumes de dados. Isso faz sentido, uma vez que o KNN concentra seu custo computacional na fase de teste e as árvores de decisão já conseguem lidar relativamente bem com atributos irrelevantes. Dessa forma, fica evidente que a seleção de características tem efeitos distintos conforme o modelo, sendo especialmente útil em algoritmos mais sensíveis à dimensionalidade dos

dados.

Capítulo 5

Conclusão

Neste trabalho, investigamos a aplicação de algoritmos genéticos para a seleção de características no diagnóstico de doenças cardíacas, utilizando técnicas de aprendizado de máquina para aprimorar a precisão dos modelos preditivos. Os resultados demonstraram que a aplicação de algoritmos genéticos, tanto na forma convencional (GA) quanto em sua variante baseada em chaves aleatórias (BRKGA), constitui uma estratégia eficaz para otimização da seleção de atributos, contribuindo significativamente para o desempenho dos classificadores.

A análise dos experimentos evidenciou que a seleção inteligente de atributos proporcionou ganhos relevantes em métricas como *F1-Score*, especialmente no modelo K-Vizinhos Mais Próximos (KNN), cujo desempenho foi significativamente elevado após a otimização. Essa melhoria está relacionada à alta sensibilidade do KNN à presença de atributos irrelevantes, o que torna a seleção de características particularmente benéfica nesse caso. Além disso, a redução da dimensionalidade contribuiu para melhorar a eficiência computacional e a interpretabilidade dos modelos, promovendo maior robustez preditiva.

A abordagem proposta mostra-se promissora para aplicações na área médica, especialmente em sistemas de apoio à decisão clínica voltados ao diagnóstico de doenças cardíacas. Sua versatilidade também permite aplicação em diversos domínios que envolvem análise preditiva, com potencial para manter desempenho consistente mesmo em cenários com conjuntos de dados distintos. Quando aliada a estratégias avançadas de aprendizado de máquina, a seleção de características por algoritmos genéticos pode ampliar seu alcance, tornando-se uma ferramenta poderosa em tarefas complexas de análise de dados.

Como perspectivas futuras, propõe-se a investigação de outras técnicas de meta-heurística, como a otimização por enxame de partículas e a busca harmônica, a fim de comparar sua eficácia com os algoritmos genéticos. Além disso, a integração com métodos de aprendizado profundo pode aprimorar ainda mais a identificação de padrões fisiológicos relevantes. Finalmente, recomenda-se a aplicação dessa abordagem em bases de dados maiores e mais heterogêneas, o que poderá consolidar sua aplicabilidade em cenários reais e em larga escala.

Referências Bibliográficas

- Al-Sahaf, H. (2019). A survey on evolutionary machine learning. *Journal of the Royal Society of New Zealand*.
- Alixandre, B. F. de F., . D. M. (2017). D-brkga: A distributed biased random-key genetic algorithm. *2017 IEEE Congress on Evolutionary Computation (CEC)*.
- Alkafaween, E. (2024). An efficiency boost for genetic algorithms: Initializing the ga with the iterative approximate method for optimizing the traveling salesman problem—experimental insigh. *Applied Sciences*.
- Babatunde, O. H., A. L. L. J. . D. D. (2014). A genetic algorithm-based feature selection. *International Journal of Electronics Communication and Computer Engineering*.
- Bishop, C. M. (2006). Pattern recognition and machine learning. *Springer-Verlag N*.
- C. Martinez, I. Loiseau, M. R. c. S. R. (2011). Brkga algorithm for the capacitated arc routing problem. *Electronic Notes in Theoretical Computer Science*.
- Darwin, C. (2014). *A Origem das Espécies 1ª Edição*. Martin Claret.
- Dreiseitl, S., . O.-M. L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatic*.
- Ericsson, K., Arroyo, J., and Pardalos, P. M. (2002). A genetic algorithm for the multiobjective multidimensional knapsack problem. *European Journal of Operational Research*, 141(2):347–358.
- Ericsson, M., M. R. and Pardalos, P. (2002). A genetic algorithm for the weight setting problem in ospf routing. *Journal of Combinatorial Optimization* 6.
- Goldenberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company.
- Gonçalves, J. F. and Almeida, J. R. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167(1):77–95.

- Gonçalves, J. F. and Almeida, J. (2002). A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics* 8.
- Google (2025). Curso intensivo de machine learning.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3:1157–1182.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- Isabelle Guyon, A. E. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* 3.
- Janiesch, C. (2021). Machine learning and deep learning. *Springer Nature*.
- Janosi, Andras, S. W.-P. M. and Detrano, R. (1989). Heart Disease. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C52P4X>.
- Jebari, K. (2013). Selection methods for genetic algorithms. *International Journal of Emerging Sciences*.
- Jianqing Fan, Kaizheng Wang, Y. Z. and Zhu, Z. (2018). Robust high-dimensional factor models with applications to statistical machine learning. *Statistical science : a review journal of the Institute of Mathematical Statistics*.
- Koohestani, B. (2020). A crossover operator for improving the efficiency of permutation-based genetic algorithms. *Expert Systems with Applications*.
- Kumar, M. (2010). Genetic algorithm: Review and application. *International Journal of Information Technology and Knowledge Management*.
- Kurt, I., T. M. . K. A. T. (2008). Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease. *Expert Systems with Applications*.
- Melanie, M. (1999). *An Introduction to Genetic Algorithms*. MIT Press.
- Molina, L. C. (2002). Feature selection algorithms: A survey and experimental evaluation. *2002 IEEE International Conference on Data Mining*.
- Muhammad Salman Pathan, Avishek Nag, M. M. P. S. D. (2017). Analyzing the impact of feature selection on the accuracy of heart disease prediction. *Elsevier*.
- Oh, I.-S. (2004). Hybrid genetic algorithms for feature selection. *IEEE*, 11:488 – 501.

- Rani, D. (2013). Genetic algorithms and their applications to water resources systems. *Metaheuristics in Water, Geotechnical and Transport Engineering*.
- Rares, M. (2015). Adaptive mutation in genetic algorithms for shortest path routing problem. *7th International Conference on Electronics, Computers and Artificial Intelligence*.
- Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition.
- Shukla, A. (2013). Comparative review of selection techniques in genetic algorithm. *International Conference on Futuristic Trends on Computational Analysis and Knowledge Management*.
- Silva, R., Souza, L., and Pereira, M. (2020). Application of biased random-key genetic algorithm in feature selection for classification problems. *Expert Systems with Applications*, 150:113290.
- Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica*.
- Tavares, J. N., Resende, M. G., and Ribeiro, C. C. (2013). A biased random-key genetic algorithm for the minimization of open stacks problem. *Optimization Letters*, 7(7):1523–1537.
- Umbarkar, A. (2015). Crossover operators in genetic algorithms: A review. *ICTACT JOURNAL ON SOFT COMPUTING*.
- U.S. National Library of Medicine (2025). MEDLINE – Premier Bibliographic Database for Biomedical Literature. https://www.nlm.nih.gov/medline/medline_home.html. Accessed: 2025-05-19.
- yan Song, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*.
- Zhang, S. (2022). Challenges in knn classification. *IEEE*.
- Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*.