Aplicação de Sistema Híbrido para classificação de doenças cardíacas com o uso de biomarcadores

1st Rodrigo Simões Costa
Engenharia da Computação
Universidade de Pernambuco
Recife, PE, Brasil
rsc@ecomp.poli.br

2nd Giordano Araújo Régis Toscano
Engenharia da Computação
Universidade de Pernambuco
Recife, PE, Brasil
gart@ecomp.poli.br

3rd Tiago Medeiros Guedes
Engenharia da Computação
Universidade de Pernambuco
Recife, PE, Brasil
tmg@ecomp.poli.br

***Abstract*—Este estudo apresenta uma abordagem de classificação binária para identificar a presença de doença cardíaca em pacientes com base em dados clínicos. Utilizamos um sistema híbrido de classificadores em Python, integrando Regressão Logística (LR), Máquina de Vetores de Suporte (SVM), K-Nearest Neighbors (KNN) e XGBoost. Cada classificador foi treinado individualmente e suas saídas foram combinadas usando um sistema de votação majoritária ponderada para determinar a classificação final. O conjunto de dados clínicos foi pré-processado para lidar com valores ausentes, normalizar atributos numéricos e codificar variáveis categóricas. O desempenho do modelo híbrido foi avaliado utilizando as métricas de acurácia, precisão, recall e F1-score. Os resultados demonstram que o sistema híbrido atinge uma classificação robusta e confiável, oferecendo uma ferramenta promissora para o diagnóstico de doenças cardíacas.**

***Index Terms*—Sistema Híbrido, Classificação de Doença Cardíaca, Machine Learning, inteligência Artificial**

I. Introdução

O diagnóstico precoce e preciso de doenças cardíacas é de fundamental importância para o tratamento eficaz e a prevenção de complicações graves. Nos últimos anos, avanços significativos têm sido alcançados na utilização de biomarcadores e algoritmos de machine learning(ML) para a identificação precoce e o prognóstico de doenças cardíacas[4,13,15]. O emprego dessas tecnologias não apenas tem potencial para aprimorar a precisão do diagnóstico, mas também pode fornecer insights valiosos sobre a fisiopatologia subjacente e os padrões de progressão da doença. Os biomarcadores são características mensuráveis e indicativas de processos biológicos normais ou patológicos. Na cardiologia, biomarcadores como colesterol, frequência cardíaca, sinais do eletrocardiograma, nível de açúcar no sangue, entre outros têm sido tradicionalmente utilizados para auxiliar no diagnóstico de eventos cardíacos agudos[5,16]. No entanto, o advento de novas tecnologias e abordagens analíticas tem expandido consideravelmente o leque de biomarcadores disponíveis, permitindo uma avaliação mais abrangente da função cardíaca e do risco cardiovascular. Paralelamente, os algoritmos de ML têm emergido como ferramentas poderosas para analisar grandes conjuntos de dados e identificar padrões complexos e inter-relações entre variáveis. A capacidade desses algoritmos de aprender com os dados e adaptar seus modelos pode melhorar significativamente a precisão do diagnóstico e a capacidade de prognóstico em comparação com métodos tradicionais[6,17]. Métodos tradicionais de machine learning (Logistic Regression, SVMs, XGBoost, KNN) têm mostrado sucesso em várias aplicações, mas enfrentam desafios em termos de interpretabilidade e integração de múltiplos tipos de dados. Neste contexto, os sistemas híbridos de classificação têm ganhado destaque por sua capacidade de combinar diferentes técnicas de IA para melhorar a precisão e a robustez dos diagnósticos[2]. Estes sistemas geralmente integram métodos de aprendizado supervisionado e não supervisionado, redes neurais artificiais, algoritmos de otimização e técnicas de mineração de dados[3]. Esta abordagem permite a análise de grandes volumes de dados complexos, como eletrocardiogramas (ECGs), exames de imagem, dados clínicos e históricos médicos, proporcionando uma visão mais abrangente do estado do paciente. Um dos principais benefícios dos sistemas híbridos é sua capacidade de lidar com a heterogeneidade dos dados médicos. A combinação de diferentes algoritmos permite explorar características complementares dos dados, o que resulta em modelos de classificação mais precisos e generalizáveis. Além disso, esses sistemas podem ser adaptados e personalizados para diferentes populações de pacientes, aumentando sua aplicabilidade clínica. Embora haja estudos promissores, ainda há uma necessidade significativa de pesquisas que explorem a eficácia de sistemas híbridos específicos para a classificação de doenças cardíacas usando biomarcadores. A motivação para este estudo está em melhorar a precisão e a rapidez do diagnóstico para salvar vidas e reduzir custos de tratamento. Os Sistemas híbridos podem fornecer uma solução mais holística e precisa ao integrar diferentes tipos de dados biomédicos. Este estudo visa mostra uma aplicação prática na aplicação de sistemas híbridos para doenças cardíacas, oferecendo *insights* valiosos para a comunidade científica e clínica. Os objetivos deste estudo são: desenvolver e validar um Sistema Híbrido que combina

diferentes técnicas de *machine learning* para a classificação de doenças cardíacas com base em biomarcadores; comparar a eficácia do sistema híbrido com métodos tradicionais e analisar sua precisão, sensibilidade e especificidade; bem como investigar como os resultados do sistema podem ser interpretados clinicamente para apoiar a tomada de decisão médica. Para atingir os objetivos, este artigo inicia com a introdução, onde são descritos a motivação do estudo, a contextualização, o *problem statement*, em seguida são apresentados os trabalhos relacionados, a metodologia proposta, o experimento realizado com os seus resultados, a discussão do que foi realizado e por fim a conclusão do estudo.

II. Trabalhos Relacionados

Durante a construção deste trabalho foram encontrados excelentes artigos que forneceram uma avaliação completa e atualizada do que é aplicado nesta área. Saranya[12] utilizou o conjunto de dados de doenças cardíacas de Cleveland e aplicou diversas técnicas para prever doenças cardíacas nos pacientes através de dados clínicos. O modelo proposto, baseado em Random Forest(RF), se apresentou com o melhor resultado, obtendo uma acurácia de 86%, SVM e Gradiente Boosted Tree alcançaram 78%, enquanto LR apresentou 77% de acurácia. A melhor sensibilidade apresentada foi de 87,3% e foi obtida pelo mesmo modelo baseado em RF.

Raphael[16] teve como objetivo desenvolver modelos com algoritmos de ML utilizando biomarcadores clinicamente relevantes para prever a presença de DAC obstrutiva estável. Oito modelos de aprendizado de máquina para prever DAC obstrutiva foram treinados em uma coorte de 1.312 pacientes. Foram utilizadas doze características clínicas e de biomarcadores sanguíneos avaliadas na admissão. A acurácia do modelo ML foi de 74,6%.

Em Xuewen[14], os autores construiram um modelo ideal para prever a ocorrência de doenças cardíacas em pacientes, levando em consideração a avaliação de dados clínicos e comparando sete algoritmos de machine learning(ML). Foram incluidos pacientes de 2018 a 2021, divididos em grupos com doenças cardíacas e saudáveis. Para avaliação dos modelos foram levados em consideração métricas como acurácia, precision, sensibilidade e curva ROC. Os resultados mostraram que o XGBoost obteve o melhor desempenho entre os sete algoritmos de ML. A AUC (área sob a curva ROC) do modelo de previsão HF-Lab9 construído pelo algoritmo XgBoost foi de 96,6% e apresentou bons benefícios clínicos.

III. Metodologia

A Fig. 1 mostra o fluxograma do desenvolvimento deste trabalho. Foram utilizados os dados de uma base de dados consolidada e muito usada para estudos de classificação de doenças cardíacas. Antes da aplicação dos classificadores, os dados passaram por um rigoroso pré-processamento, que incluiu agrupamento dos valores target; tratamento de valores ausentes, onde foram aplicadas as modas por grupo de pacientes (com ou sem doença cardíaca); normalização dos dados e codificação de variáveis categóricas, que foram transformadas em numéricas.

Para processamento dos algoritmos, foi usado a linguagem python e o Colab. Foi implementado um sistema híbrido composto por quatro algoritmos de classificação: Regressão Logística (LR), escolhido devido à sua simplicidade e interpretabilidade; Máquina de Vetores de Suporte (SVM), escolhido por sua eficácia em classificações binárias e capacidade de lidar com espaços de alta dimensionalidade; K-Nearest Neighbors (KNN), selecionado por sua simplicidade e eficácia em pequenas quantidades de dados; e XGBoost, utilizado por sua alta performance e capacidade de lidar com dados desbalanceados; e a classificação por voto majoritário ponderado, que classifica os pacientes avaliando a votação dos modelos anteriores multiplicado pelos pesos atribuídos a cada algoritmo.

Cada classificador foi treinado individualmente utilizando os dados de treinamento e foram realizados ajustes de hiperparâmetros específicos para cada modelo. As saídas individuais de cada classificador foram combinadas utilizando um sistema de votação majoritária. Neste sistema, a classificação final foi determinada pela classe mais frequentemente predita pelos classificadores individuais. A performance do sistema híbrido foi avaliada utilizando métricas como acurácia, precisão, recall, e F1-score. As métricas foram calculadas utilizando um conjunto de dados de teste, separado do conjunto de dados de treinamento, para garantir a generalização do modelo. Foi aplicada ainda o teste de hipótese para saber se os modelos são estatisticamente equivalentes ou não, podendo selecionar assim os melhores modelos.

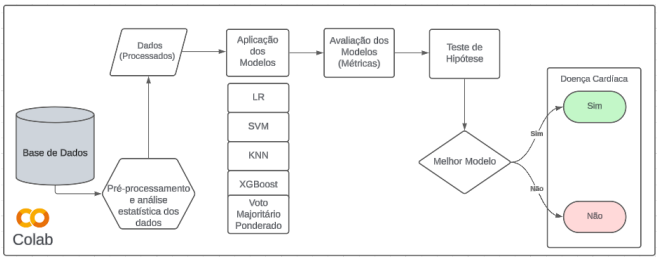

Fig. 1. Fluxograma do trabalho desenvolvido.

A. Base de Dados

A base de dados utilizada para este estudo contém 920 instâncias, com 76 atributos, mas todos os experimentos publicados referem-se ao uso de um subconjunto de apenas 14 atributos [1]. As variáveis independentes são formadas por informações de idade, sexo, dor no peito, pressão arterial, colesterol, nível de açúcar no sangue, resultado do eletrocardiograma (ECG), frequência cardíaca, número de vasos principais coloridos por fluoroscopia, depressão do segmento ST induzida por exercício e angina induzida por exercício, inclinação do segmento ST máxima e talassemia, sendo a variável dependente e target a gravidade da doença cardíaca. A gravidade da doença cardíaca apresenta um valor inteiro

de 0 (sem presença) a 4. Os experimentos com este banco de dados se concentraram em simplesmente tentar distinguir presença (valores 1,2,3,4) de ausência (valor 0).

B. Técnicas de Aprendizado de Máquinas

O problema proposto neste trabalho se refere a necessidade de classificação binária com dados supervisionados, ou seja, dados classificados por especialistas e usados nesta classificação para treinamento dos nossos modelos. O algoritmo utiliza as características e valores encontrados nos atributos para, através destes, classificar se o paciente possui ou não doença cardíaca. Para atender a estes requisitos usamos algoritmos de classificação, sendo eles: Regressão Logística (LR)[7,8], K-Nearest Neighbors (KNN)[10], SVM[9] e XGBoost[11]; em seguida aplicamos o voto majoritário ponderado para melhorar a performance individual dos modelos. Ao final faremos a comparação dos resultados.

1) Regressão Logística: As técnicas de classificação são uma parte essencial nos aplicativos de aprendizado de máquina e mineração de dados. Grande parte dos problemas em *Data Science* são de classificação e a Regressão Logística é um método de regressão útil para resolver o problema de classificação binária. A Regressão Logística ou *Logistic Regression* (LR) é um algoritmo de classificação usado para estimar o valor discreto (0 ou 1, sim / não, verdadeiro / falso), medindo a relação entre a variável dependente categórica e uma ou mais variáveis independentes, estimando a probabilidade de ocorrência de um evento através da sua função logística. O resultado ou variável de destino é de natureza dicotômica que significa a existência de apenas duas classes possíveis, calculando a probabilidade de ocorrência de um evento.

A LR é um caso especial de regressão linear onde a variável alvo é categórica por natureza. LR é um classificador linear onde usamos uma função linear

$$f(x) = b_0 + b_1x_1 + ... + b_rx_r \tag{1}$$

cujas variáveis $b_0, b_1, ..., b_r$ são os estimadores dos coeficientes de regressão, também chamados de pesos previstos ou apenas coeficientes. A probabilidade da função da regressão logística $p(x)$ é a função sigmóide de f(x):

$$p(x) = (1 + exp(-f(x)) \tag{2}$$

Como tal, muitas vezes é próximo de 0 ou 1. $p(x)$ é frequentemente interpretada como a probabilidade prevista de que a saída para um dado x seja igual a 1. Portanto, $1 - p(x)$ é a probabilidade de que a saída seja 0. A regressão linear determina os melhores pesos previstos $b_0, b_1, ..., b_r$ de modo que a função logística $p(x)$ seja o mais próximo possível de todas as respostas reais $y_i, i = 1, ..., n$, onde n é o número de observações[8].

Para maior esclarecimento, podemos fazer uma comparação entre a regressão linear e a regressão logística, onde a regressão linear fornece uma saída contínua, enquanto a regressão logística fornece uma saída discreta(0 ou 1). A saída contínua pode ser exemplificada como o preço da casa e o preço das ações que podem receber vários valores, por outro lado, a saída discreta pode ser a previsão de um paciente ter diabetes ou não, caindo em uma resposta binária (sim ou não). A regressão linear é estimada usando Mínimos Quadrados Ordinários (OLS), enquanto a regressão logística é estimada usando a abordagem Estimativa de Máxima Verossimilhança (MLE).

A estimativa OLS é um método de aproximação de minimização de distância e são calculadas ajustando uma linha de regressão em determinados pontos de dados que tem a soma mínima dos desvios quadrados (erros de mínimos quadrados), enquanto a MLE usa a maximização da função de verossimilhança e determina os parâmetros com maior probabilidade de produzir os dados observados. Estatisticamente, o MLE define a média e a variância como parâmetros na determinação dos valores paramétricos específicos para um determinado modelo. Este conjunto de parâmetros pode ser usado para prever os dados necessários em uma distribuição normal[18].

A função sigmóide ou função logística, fornece uma curva em forma de S, onde recebe valores de entrada e após cálculos matemáticos, retorna como saída valores entre 0 e 1. A Fig. 2 mostra o comportamento da função sigmóide.

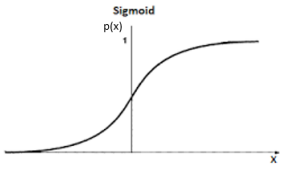

Fig. 2. Gráfico da Função Sigmóide: Mostrando que os limites dos valores resultantes ficam entre 0 e 1.

2) K-Nearest Neighbors (KNN): O algoritmo K-Nearest Neighbors (KNN) é um dos métodos mais intuitivos e amplamente utilizados no campo do aprendizado de máquina para tarefas de classificação e regressão. O princípio fundamental do KNN é baseado na ideia de que pontos de dados semelhantes tendem a estar próximos uns dos outros no espaço de características[10]. O KNN pode ser descrito pelas principais etapas:

- Escolha do valor de K, onde o parâmetro K representa o número de vizinhos mais próximos que serão considerados para a decisão de classificação ou previsão. A escolha do valor de K é crítica para o desempenho do modelo e geralmente é feita com base em técnicas de validação cruzada.
- Cálculo das Distâncias: Para classificar ou prever o valor de um novo ponto de dados, o KNN calcula a distância entre esse ponto e todos os pontos do conjunto de treinamento. A distância Euclidiana é a métrica mais comum utilizada e sua formulação matemática é dada pela equação:

$$Distância = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

A Fig. 3 mostra a representação gráfica da Distância Euclidiana. Entretanto, outras métricas como a distância de Manhattan ou a distância de Minkowski também podem ser empregadas dependendo da natureza dos dados e do problema específico.

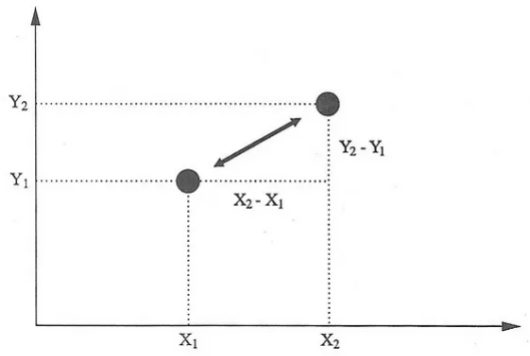

Fig. 3. Gráfico da Distância Euclidiana: Mostrando a distancia entre os pontos X1 e X2, bem como Y1 e Y2.

- Identificação dos K Vizinhos mais Próximos: Após calcular as distâncias, o algoritmo seleciona os K pontos de treinamento mais próximos ao ponto de interesse. Essa seleção é feita com base na menor distância calculada.
- Classificação ou Predição: No contexto de problemas de classificação, o ponto de dados é atribuído à classe mais frequente entre seus K vizinhos mais próximos. Em problemas de regressão, o valor de resposta é geralmente a média ou a mediana dos valores de resposta dos K vizinhos mais próximos.

A Fig. 4 mostra a aplicação do algoritmo KNN na base de dados IRIS, separando os dados em 3 classes de classificação.

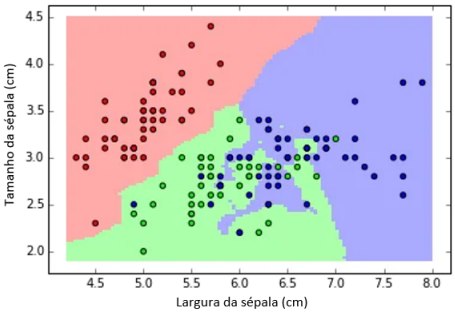

Fig. 4. Aplicação do KNN na base de dados IRIS.

3) SVM ou Support Vector Machine: É uma técnica de aprendizado de máquina amplamente usada para classificação, a qual recebe os vetores de entrada e calcula um hiperplano que os separa, ou seja, é um classificador linear binário não probabilístico. Esse hiperplano busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes, sendo chamada de margem. A SVM coloca em primeiro lugar a classificação das classes, definindo assim cada ponto pertencente a cada uma das classes e em seguida maximiza a margem[9].

Na regressão logística, pegamos a saída da função linear e restringimos o valor dentro do intervalo entre 0 e 1 através da função sigmóide. No SVM, se a saída da função linear for maior que 1, a identificamos com uma classe e se for menor que -1, a identificamos com outra classe. Como os valores de limiar são alterados para 1 e -1 no SVM, obtemos essa faixa de valores de reforço ([-1,1]) que atua como margem. Na Fig. 5 é apresentado o gráfico do modelo SVM, onde temos os vetores de suporte, o hiperplano ótimo e as maximizações das margens de separação.

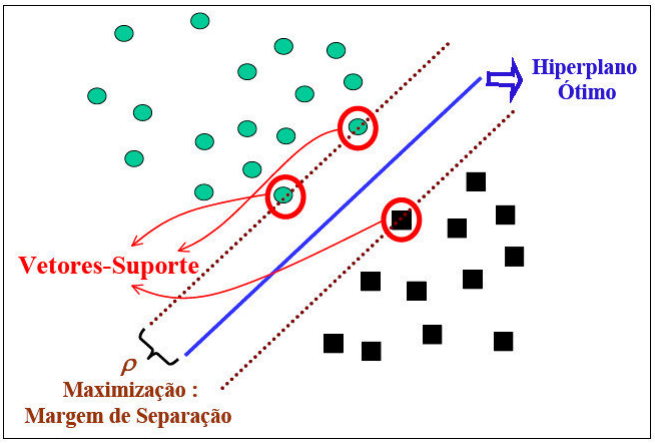

Fig. 5. Representação do Modelo SVM: Mostrando o Hiperplano Ótimo, os Vetores de Suporte e a Maximização das Margens de Separação do Dados.

Sua formulação matemática é descrita da seguinte forma: Dados vetores de treinamento $x_i \epsilon R^p$, $i = 1, \ldots, n$, em duas classes, e um vetor $y \epsilon \{ 1, -1\}$n, nosso objetivo é encontrar w ϵR^p e $b \epsilon$ R tal que a previsão dada por $sign(w^T \phi(x) + b)$ está correto para a maioria das amostras. O SVC resolve o seguinte problema primário:

$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$

Subjetivo para $y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i$, onde $\zeta_i \geq 0, i = 1, ..., n$

Intuitivamente, estamos tentando maximizar a margem (minimizando $||w||^2 = w^T w$), enquanto incorre em penalidade quando uma amostra é classificada incorretamente ou dentro do limite da margem. Idealmente, o valor $y_i(w^T \phi(x_i) + b) seria \geq 1$ para todas as amostras, o que indica uma previsão perfeita. Mas os problemas geralmente nem sempre são perfeitamente separáveis com um hiperplano, então permitimos que algumas amostras estejam à distância ζ_i de seu limite de margem correto. O termo de penalidade C controla a força dessa penalidade e, como resultado, atua como um parâmetro de regularização inverso.

O problema secundário para o primário é

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

Subjetivo para $y^T \alpha = 0, 0 \leq \alpha_i \leq C, i = 1, ..., n$

Onde *e* é o vetor de todos os uns, e Q é um n por n matriz semi definida positiva, $Q_{ij}y_iy_jK(x_i,x_j), OndeK(x_i,x_j) = \phi(x_i)T\phi(x_j)$ é o núcleo. Os termos $_i$ são chamados de coeficientes secundários e são limitados superiormente por C. Essa representação dupla destaca o fato de que os vetores de treinamento são mapeados implicitamente em um espaço dimensional superior (talvez infinito) pela função ϕ. Uma vez resolvido o problema de otimização, a saída da função-decisão para uma determinada amostra x torna-se:

$$\sum_{i\epsilon SV} y_i\alpha_i K(x_i,x) + b$$

e a classe prevista corresponde ao seu sinal. Nós só precisamos somar os vetores de suporte (ou seja, as amostras que estão dentro da margem) porque os coeficientes secundários α_i são zero para as outras amostras. Esses parâmetros podem ser acessados através dos atributos dual-coef- que contém o produto $y_i\alpha_i$, support-vectors- que contém os vetores de suporte e intercept- que contém o termo independente[9].

4) XGBoost ou Extreme Gradient Boosting: O XGBoost (Extreme Gradient Boosting) é uma biblioteca de machine learning poderosa e eficiente, amplamente utilizada para problemas de classificação e regressão. Desenvolvida por Tianqi Chen e colaboradores, essa técnica é uma implementação aprimorada do algoritmo de gradient boosting, que combina de maneira sequencial vários modelos fracos (geralmente árvores de decisão) para criar um modelo forte. A característica principal do XGBoost é sua capacidade de otimização em termos de velocidade e desempenho, devido ao uso de algoritmos paralelos para a construção de árvores, regularização para evitar overfitting e tratamento eficiente de dados esparsos. Essas melhorias permitem que o XGBoost lide com grandes volumes de dados e produza resultados altamente precisos.

O mecanismo do XGBoost baseia-se no princípio de minimizar uma função de perda utilizando um método de otimização por gradiente. Em cada iteração, o algoritmo ajusta uma nova árvore para corrigir os erros cometidos pelas árvores anteriores. Essa abordagem iterativa e aditiva permite que o modelo aprenda de maneira mais eficaz a partir dos dados de treinamento. Além disso, o XGBoost incorpora técnicas de regularização L1 (lasso) e L2 (ridge) para penalizar a complexidade do modelo, reduzindo assim a possibilidade de overfitting. A combinação desses elementos torna o XGBoost altamente flexível e robusto, capaz de se adaptar a diferentes tipos de dados e problemas[11]. A formulação matemática pode ser descrita como segue:

Dado um conjunto de dados de treinamento $\{(x_i,y_i)\}_{i=1}^n$, onde x_i representa as características dos dados e y_i os rótulos, o objetivo do XGBoost é minimizar a seguinte função de perda regularizada:

$$L(\phi) = \sum_{i=1}^{n} l(y_i,\hat{y}_i) + \sum_{k=1}^{K}\Omega(f_k), \tag{4}$$

onde $\hat{y}_i$ é a previsão do modelo para a instância i, l é a função de perda (por exemplo, erro quadrático para regressão ou log-loss para classificação), e Ω é o termo de regularização definido como:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2. \tag{5}$$

Aqui, T é o número de folhas na árvore, γ e λ são parâmetros de regularização que controlam a complexidade do modelo, e w_j são os pesos associados às folhas.

A previsão do modelo $\hat{y}_i$ na t-ésima iteração é dada por:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i), \tag{6}$$

onde $f_k \in \mathcal{F}$ são as funções de regressão (árvores) na iteração t, e $\mathcal{F}$ é o espaço das funções de regressão (árvores de decisão).

Durante o treinamento, o objetivo é adicionar uma nova árvore que minimiza a função de perda regularizada. A função objetivo na t-ésima iteração pode ser aproximada usando uma expansão de Taylor de segunda ordem:

$$L^{(t)} \approx \sum_{i=1}^{n}\left[l(y_i,\hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t), \tag{7}$$

onde $g_i = \frac{\partial l(y_i,\hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ e $h_i = \frac{\partial^2 l(y_i,\hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)^2}}$ são os primeiros e segundos gradientes da função de perda, respectivamente.

A nova árvore f_t é ajustada para minimizar essa função objetivo aproximada, resultando em uma melhoria incremental na previsão do modelo.

Uma das vantagens notáveis do XGBoost é sua capacidade de paralelização, que acelera significativamente o processo de treinamento. Isso é alcançado dividindo os dados em blocos e processando-os simultaneamente, aproveitando ao máximo os recursos de hardware disponíveis. Além disso, o XGBoost possui funcionalidades avançadas, como suporte para missing values, técnica de pruning (podamento) para evitar o crescimento excessivo das árvores, e capacidade de lidar com dados em diferentes formatos, incluindo matrizes esparsas. Essas características, juntamente com sua alta acurácia preditiva, fazem do XGBoost uma escolha preferida em competições de machine learning e em aplicações industriais onde a precisão e a eficiência são cruciais.

5) Sistema Híbrido: Um sistema híbrido para classificação é uma abordagem que combina múltiplas técnicas de aprendizagem de máquina e métodos de inteligência artificial para melhorar o desempenho de modelos de classificação. Esses sistemas aproveitam as vantagens de diferentes algoritmos para superar as limitações individuais, resultando em uma classificação mais precisa e robusta.

Os sistemas híbridos são particularmente eficazes em situações onde os dados são complexos ou possuem características heterogêneas, exigindo uma abordagem mais sofisticada para capturar padrões subjacentes. Eles são amplamente utilizados em diversas áreas, como diagnóstico médico,

detecção de fraudes, reconhecimento de padrões e análise preditiva. Ao combinar a força de diferentes algoritmos, os sistemas híbridos conseguem alcançar uma precisão superior e uma maior capacidade de generalização, tornando-os uma escolha poderosa para tarefas de classificação desafiadoras.

A integração de múltiplos algoritmos em um sistema híbrido pode ocorrer de várias maneiras. Neste trabalho, utilizamos algoritmos como RF, SVM, KNN e XGBoost para fazer a classificação dos nossos dados, ao final, utilizamos a técnica de voto majoritário ponderado para definir a classificação final dos dados. Desta maneira, os dados são classificados de acordo com a classificação individual de cada algoritmo, multiplicado pelo peso atribuído a cada um. Os pesos atribuídos foram 5 para LR e KNN, 1 para SVM e 10 para XGBoost, conforme fórmula abaixo:

$$pMH = (5*pLR)+(5*pKNN)+(1*pSVM)+(10*pXGB) \quad (8)$$

Nesta fórmula, pMH, pLR, pKNN, pSVM, pXGB são as previsões dos modelos híbrido, Logistic Regression, KNN, SVM e XGBoost respectivamente. Na classificação final do modelo de voto majoritário ponderado, se o pMH for maior ou igual a 10, é classificado como 1, se não, é classificado como 0.

IV. Métricas de avaliação

Antes de falarmos das métricas de avaliação utilizadas, precisamos explicar um conceito bastante útil que é sobre a matriz de confusão que se forma como resultado das avaliações. Uma matriz de confusão é uma tabela que indica os erros e acertos do seu modelo, comparando os resultados reais com os resultados esperados. A Fig. 6 mostra um exemplo de uma matriz de confusão.

		Avaliada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fig. 6. Representação de uma Matriz de Confusão.

Como mostrado na Matriz de Confusão, teremos 4 avaliações para as nossas classificações, que são elas:

- Verdadeiros Positivos: classificação correta da classe Positivo, onde todos os exemplos foram avaliados como positivos e realmente são positivos;
- Falsos Negativos: erro em que o modelo previu a classe Negativo quando o valor real era classe Positivo;
- Falsos Positivos: erro em que o modelo previu a classe Positivo quando o valor real era classe Negativo;
- Verdadeiros Negativos: classificação correta da classe Negativo, onde todos os exemplos foram avaliados como negativos e realmente são negativos;

As métricas usadas para a avaliação dos resultados foram Acurácia, Precisão, *Recall* e *F1-score* e suas fórmulas são mostradas na Fig. 7.

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

$$\text{Precision} = \frac{VP}{VP + FP}$$

$$\text{Recall} = \frac{VP}{VP + FN}$$

$$\text{F1-score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}}$$

Fig. 7. Métricas da Avaliação usadas para avaliação dos resultados deste trabalho.

Acurácia - É a quantidade de acertos do nosso modelo (verdadeiro positivo + verdadeiro negativo) dividido pelo total da amostra (verdadeiro positivo + verdadeiro negativo + falso positivo + falso negativo).

Precisão - É o resultado de todos os dados classificados como positivos, quantos são realmente positivos (verdadeiro positivo / (verdadeiro positivo + falso positivo)).

Recall - É o resultado da quantidade de dados positivos (verdadeiro positivo) dividido pela quantidade real de resultados positivos (verdadeiro positivo + falso negativo) que existem em nossa amostra.

F1-score - Une *precision* e *recall* afim de trazer um número único que determina a qualidade geral do nosso modelo e é calculada da seguinte forma: (2*Precision*Recall)/(Precision+Recall) [19].

V. Experimento Realizado

O trabalho foi realizado usando a base de dados Heart Disease[1] com o objetivo de classificar os pacientes com relação a existência ou não de doença cardíaca. As etapas realizadas são mostradas na Fig. 8.

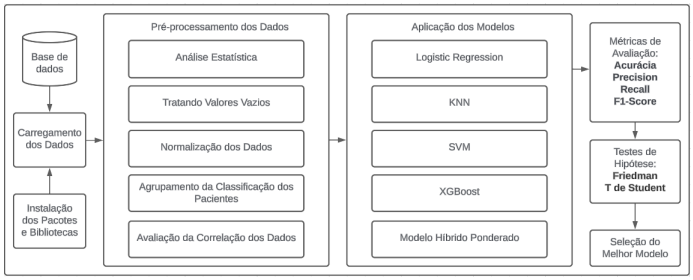

Fig. 8. Etapas realizadas no desenvolvimento do estudo.

Inicialmente, houve a preparação do ambiente de processamento do código com as instalações das bibliotecas necessárias e os dados foram baixados da base de dados. Na etapa seguinte, foi realizado o pré-processamento dos dados, onde passou pela análise estatísticas dos dados, tratamento

dos valores vazios, normalização dos dados, agrupamento da classificação dos pacientes entre existência ou não de doença cardíaca e a avaliação da correlação das variáveis. Com a saída dos dados tratados, foram aplicados os modelos de classificação Logistic Regression, KNN, SVM, XGBoost e um Modelo Híbrido Ponderado. Na etapa seguinte é feita a mensuração das métricas de avaliação dos modelos e para esta avaliação, por se tratar de um problema de classificação binária com uma quantidade de 920 instâncias, foram usadas métricas de Acurácia, Precision, Recall e F1-Score. A penúltima etapa foi a aplicação dos testes de hipóteses de Friedman, de Nemenyi e T de Student onde foi avaliada se existe diferença estatística enter os modelos. Após a avaliação estatística, o melhor modelo foi selecionado.

Foi aplicado o Grid Search para encontrar os melhores parâmetros para serem usados nos modelos. O resultando é mostrado na Tabela 1.

TABLE I
PARÂMTEROS USADOS NOS MODELOS DE CLASSIFICAÇÃO.

Modelo	Parâmetros
Logistic Regression	C=1.0,max$_i ter = 100, penalty = ' l1'$, solver='saga'
K-Neighbors	n-neighbors=10
Support Vector Machine	C=10,gamma=0.1,kernel='rbf'
XGBoost	learning-rate=0.2,max-depth=3, n-estimators=50,subsample=1.0

VI. DISCUSSÃO

Na sequência do estudo apresentamos os resultados obtidos após a aplicação dos modelos de classificação. A Fig. 9 mostra as matrizes de confusão dos modelos aplicados. O modelo SVM se apresentou complementar ao modelo XGBoost, levando em considerações os tipos de erros encontrados, um errou mais a previsão dos casos positivos para a doença enquanto o outro errou mais os casos negativos.

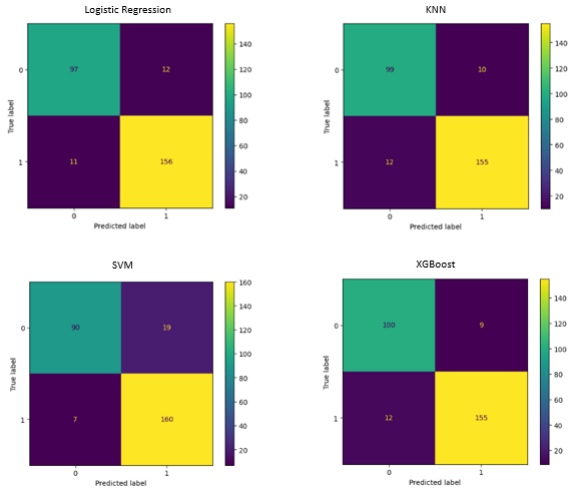

Fig. 9. Matriz de confusão de cada modelo aplicado.

O teste de hipóteses de Friedman e de Nemenyi foram realizados para avaliar se existe diferença estatisticamente significativa entre os modelos. Os modelos foram representados como modelo 1 o LR, modelo 2 o KNN, modelo 3 o SVM e o modelo 4 o XGBoost. Conforme mostrado na Fig. 10, os modelos apresentam diferença estatística significativa. O modelo XGBoost se apresenta com o melhor resultado, seguido do modelo KNN. Na sequência, foi aplicado o teste T de Student para avaliar se existe diferença estatisticamente significativa entre dois modelos. O resultado mostrou que o modelo SVM apresentou diferença estatisticamente significativa para os outros modelos. Os modelos Logistic Regression, KNN e XGBoost não apresentaram diferença estatisticamente significativa entre si.

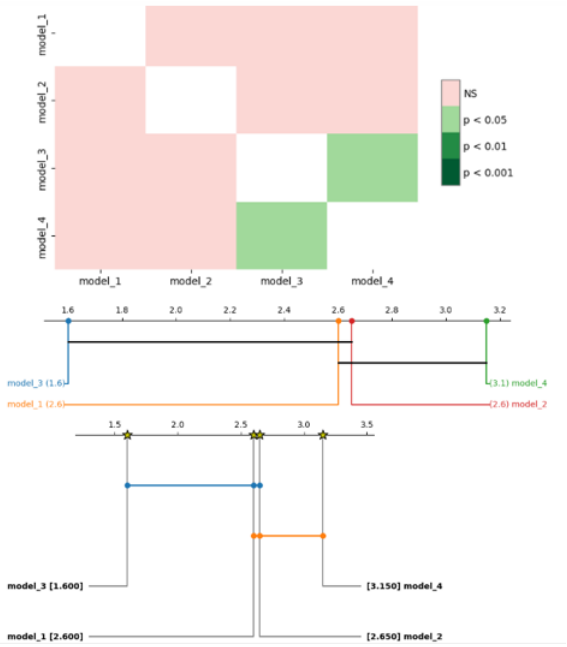

Fig. 10. Teste de Hipótese de Friedman - Diagrama de diferença crítica de classificações de pontuação média.

Na sequência, tendo em vista que os modelos erram e acertam classificações diferentes, aplicamos a classificação pelo modelo de voto majoritário ponderado. A matriz de confusão deste modelo é apresentada na Fig. 11.

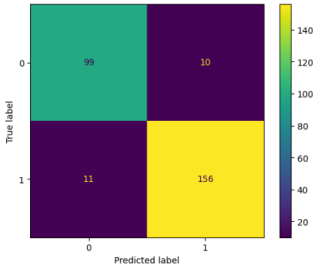

Fig. 11. Matriz de Confusão do modelo de voto majoritário ponderado.

O resultado do modelo de voto majotirtário ponderado

melhorou o resultado individual dos modelos aplicados anteriormente e pode ser visto na Fig. 12.

Métrica	Modelo				
	LR	KNN	SVM	XGBoost	Voto Majoritário
Acurácia	0.9167	0.9203	0.9058	**0.9239**	**0.9239**
Precision_1	0.9286	0.9394	0.8939	**0.9451**	0.9398
Precision_0	0.8981	0.8919	**0.9278**	0.8929	0.900
Recall_1	0.9341	0.9281	**0.9581**	0.9281	0.9341
Recall_0	0.8899	0.9083	0.8257	**0.9174**	0.9083
F1-SCORE_1	0.9313	0.9337	0.9249	0.9366	**0.9369**
F1-SCORE_0	0.8940	0.9000	0.8738	**0.9050**	0.9041
Validação Cruzada	0.9223	0.9224	0.8882	0.9302	
Desvio Padrão	0.0252	0.0368	0.0382	0.0312	

Fig. 12. Resultado obtidos dos modelos aplicados.

VII. Conclusão

Neste trabalho ficou evidenciado que o uso de machine learning para a classificação de doenças cardíacas pode trazer bons resultados e podem ser aplicado no dia a dia das análises clínicas, ajudando a tomada de decisão dos médicos e melhorando a qualidade de vida da população. Na comparação dos modelos a serem usados, os modelos híbridos podem melhorar os resultados individuais dos modelos tradicionais de machine learning aplicados na classificação de doenças cardíacas. O principal motivo é a capacidade de aproveitar as melhores características de cada modelo, resultando em um modelo robusto e de alta performance.

Atendendo aos objetivos propostos neste estudo, foi desenvolvido e validado um Sistema Híbrido que combina diferentes técnicas de *machine learning* para a classificação de doenças cardíacas com base em biomarcadores. O modelo de voto majoritário ponderado, apresentou uma acurácia de 92,39%, com uma precisão para a classe 0 de 90,00% e para a classe 1 de 93,98%. A métrica Recall obtida para a classe 0 foi de 90,83% e para a classe 1 de 93,41%. A métrica F1-Score para a classe 0 foi 90,41% e para a classe 0 foi de 93,69% . Na sequência, foi comparado a eficácia do sistema híbrido com métodos tradicionais e por fim, foram analisados a sua precisão, sensibilidade e especificidade. O modelo baseado no voto majoritário híbrido melhorou alguns parâmetros se comparado ao melhor modelo individual.

O modelo proposto se apresenta com uma possibilidade de ser aplicado na prática das análise clínicas, servindo de apoio nas tomadas de decisões e procedimentos a serem adotados. Como trabalho futuro, fica o desafio da ampliação da base de dados trabalhada neste estudo, para consolidação das técnicas utilizadas e seus resultados. Outro desafio a ser alcançado na continuidade destas aplicações é a possibilidade da utilização destes modelos nos diagnósticos de outras doenças que trazem padrões associados aos seus marcadores e que sejam possíveis suas avaliações pelas técnicas de Aprendizado de Máquinas. Por fim, podemos integrar este tipo de sistema inteligente à máquinas para geração automática de alertas de saúde, tornando o processo proativo no diagnóstico e tratamento de doentes. Acreditamos que a ciência aplicada na prática, pode trazer grandes contribuições para a melhoria da qualidade de vida das pessoas e a Inteligência Artificial

Referências

References

[1] Janosi,Andras, Steinbrunn,William, Pfisterer,Matthias, and Detrano,Robert. (1988). Heart Disease. UCI Machine Learning Repository. https://doi.org/10.24432/C52P4X.

[2] DIETTERICH, Thomas G. Ensemble methods in machine learning.In: International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2000. p. 1-15.

[3] ROKACH, Lior. Pattern classification using ensemble methods. World Scientific, 2010.

[4] de Bakker, Marie, et al. "Machine learning-based biomarker profile derived from 4210 serially measured proteins predicts clinical outcome of patients with heart failure." European Heart Journal-Digital Health 4.6 (2023): 444-454. doi.org/10.1093/ehjdh/ztad056.

[5] Kim, Juntae, et al. "Machine learning models of clinically relevant biomarkers for the prediction of stable obstructive coronary artery disease." Frontiers in Cardiovascular Medicine 9 (2022): 933803. doi.org/10.3389/fcvm.2022.933803

[6] Berikol, Göksu Bozdereli, Oktay Yildiz, and I. Türkay Özcan. "Diagnosis of acute coronary syndrome with a support vector machine." Journal of medical systems 40.4 (2016): 84. doi.org/10.1007/s10916-016-0432-6.

[7] Mueller, Christian, et al. "Multicenter evaluation of a 0-hour/1-hour algorithm in the diagnosis of myocardial infarction with high-sensitivity cardiac troponin T." Annals of emergency medicine 68.1 (2016): 76-87. doi.org/10.1016/j.annemergmed.2015.11.013.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.linearmodel.LogisticRegression.html , acessado em 23/07/2024.

[9] https://scikit-learn.org/stable/modules/generated /sklearn.svm.SVC.html, acessado em 23/07/2024.

[10] https://scikit-learn.org/stable/modules/generated /sklearn.neighbors.KNeighborsClassifier.html, acessado em 23/07/2024.

[11] https://xgboost.readthedocs.io/en /stable/python/sklearnestimator.html, acessado em 23/07/2024.

[12] Saranya, G., and A.Pravin."A novel feature selection approach with integrated feature sensitivity and feature correlation for improved prediction of heart disease." Journal of Ambient Intelligence and Humanized Computing9(2023): 12005-12019.doi.org/10.1007/s12652-022-03750-y.

[13] Twerenbold, Raphael, et al. "Clinical use of high-sensitivity cardiac troponin in patients with suspected myocardial infarction." Journal of the American College of Cardiology 70.8 (2017): 996-1012. doi.org/10.1016/j.jacc.2017.07.718.

[14] Li, Xuewen, et al. "Development and comparison of machine learning-based models for predicting heart failure after acute myocardial infarction." BMC Medical Informatics and Decision Making 23.1 (2023): 165. doi.org/10.1186/s12911-023-02240-1.

[15] Kaier, Thomas E., et al. "A 0/1h-algorithm using cardiac myosin-binding protein C for early diagnosis of myocardial infarction." European Heart Journal Acute Cardiovascular Care 11.4 (2022): 325-335. doi.org/10.1093/ehjacc/zuac007.

[16] Mokhtari, Arash, et al. "A 1-h combination algorithm allows fast rule-out and rule-in of major adverse cardiac events." Journal of the American College of Cardiology 67.13 (2016): 1531-1540. doi.org/10.1016/j.jacc.2016.01.059.

[17] Than, Martin P., et al. "Machine learning to predict the likelihood of acute myocardial infarction." Circulation 140.11 (2019): 899-909. doi.org/10.1161/CIRCULATIONAHA.119.041980.

[18] Bishop, M. Christopher: Pattern Recognition and Machine Learning, (2006)

[19] https://scikit-learn.org/stable/modules/modelevaluation.html, acessado em 23/07/2024.