REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

TÉCNICAS DE INTELIGÊNCIA COMPUTACIONAL PARA ANÁLISE E PROCESSAMENTO DE DADOS: APLICAÇÃO DE MODELOS DE *MACHINE LEARNING* NO DIAGNÓSTICO DE DOENÇAS GERIÁTRICAS

COMPUTATIONAL INTELLIGENCE TECHNIQUES FOR DATA ANALYSIS AND PROCESSING: APPLICATION OF MACHINE LEARNING TECHNIQUES IN THE DIAGNOSIS OF GERIATRIC DISEASES

TÉCNICAS DE INTELIGENCIA COMPUTACIONAL PARA EL ANÁLISIS Y PROCESAMIENTO DE DATOS: APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING EN EL DIAGNÓSTICO DE ENFERMEDADES GERIÁTRICAS

Ruy de Morais e Silva[1], Gabriela Gomes Cavalcanti Alves Monteiro[1], Samara Martins Nascimento Gonçalves[2], Veronica Maria Lima Silva[1]

e6116913
https://doi.org/10.47820/recima21.v6i11.6913

PUBLICADO: 11/2025

RESUMO

O Acidente Vascular Cerebral é uma das principais causas de morte e incapacidade no mundo. A doença exige um diagnóstico rápido para minimizar sequelas e melhorar o prognóstico dos pacientes. Dessa forma, algumas técnicas de Aprendizado de Máquina vêm se mostrando ferramentas promissoras para auxiliar na triagem pré-hospitalar. Este trabalho apresenta a implementação e avaliação de quatro modelos de classificação: *K-Nearest Neighbors*, *Random Forest*, *eXtreme Gradient Boosting* e *Support Vector Machine*, aplicados ao conjunto de dados "*Stroke Prediction*" do Kaggle, que foi submetido a etapas de pré-processamento, balanceamento de classes e otimização de hiperparâmetros. Além disso, aplicou-se a técnica *SelectKBest* para identificar as variáveis mais relevantes, visando futuras aplicações em sistemas embarcados. Os resultados indicaram um bom desempenho em todos os modelos, com destaque para o *Random Forest*, que alcançou acurácia de 98,9% com 12 variáveis e manteve 96,9% ao ser reduzido para apenas quatro variáveis de maior relevância (idade, hipertensão, doença cardíaca e glicemia média). Os experimentos demonstram que modelos podem apoiar de forma eficaz a detecção precoce da doença, possibilitando sua integração em aplicações móveis ou dispositivos de baixo custo voltados para triagem rápida e confiável.

PALAVRAS-CHAVE: Acidente Vascular Cerebral. Triagem. Aprendizagem de Máquina.

ABSTRACT

Stroke is one of the leading causes of death and disability worldwide, requiring rapid diagnosis to minimize sequelae and improve patient prognosis. Therefore, some Machine Learning techniques have shown promise as tools to support pre-hospital triage. This work presents the implementation and evaluation of four classification models: K-Nearest Neighbors, Random Forest, eXtreme Gradient Boosting and Support Vector Machine. These models were applied to the Kaggle "Stroke Prediction" dataset, which was subjected to preprocessing, class balancing, and hyperparameter. Furthermore, the SelectKBest technique was applied to identify the most relevant variables, targeting future applications in embedded systems. The results indicated good performance across all models, with Random Forest standing out, achieving 98.9% accuracy with 12 variables and maintaining 96.9% accuracy when reduced to just four key variables (age, hypertension, heart disease, and average blood glucose). The experiments demonstrate that models can effectively support early

[1] Universidade Federal da Paraíba.
[2] Universidade Federal Rural do Semi-Árido: Mossoró.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

stroke detection, enabling their integration into mobile applications or low-cost devices for rapid and reliable screening.

KEYWORDS: *Stroke. Triage. Machine Learning*

RESUMEN
El Accidente Cerebrovascular es una de las principales causas de muerte y discapacidad en el mundo. La enfermedad requiere un diagnóstico rápido para minimizar secuelas y mejorar el pronóstico de los pacientes. En este sentido, algunas técnicas de Aprendizaje Automático han demostrado ser herramientas prometedoras para apoyar el triaje prehospitalario. Este trabajo presenta la implementación y evaluación de cuatro modelos de clasificación: K-Nearest Neighbors, Random Forest, eXtreme Gradient Boosting y Support Vector Machine, aplicados al conjunto de datos "Stroke Prediction" de Kaggle, el cual fue sometido a etapas de preprocesamiento, balanceo de clases y optimización de hiperparámetros. Además, se aplicó la técnica SelectKBest para identificar las variables más relevantes, con vistas a futuras aplicaciones en sistemas embebidos. Los resultados indicaron un buen desempeño en todos los modelos, destacándose Random Forest, que alcanzó una precisión del 98,9 % con 12 variables, y mantuvo el 96,9 % al reducirse a solo cuatro variables clave (edad, hipertensión, enfermedad cardíaca y glucemia media). Los experimentos demuestran que los modelos de aprendizaje automático pueden apoyar eficazmente la detección temprana de enfermedad, permitiendo su integración en aplicaciones móviles o dispositivos de bajo costo orientados al triaje rápido y confiable.

PALABRAS CLAVE*: Accidente Cerebrovascular. Triaje. Aprendizaje Automático.*

1. INTRODUÇÃO

O Acidente Vascular Cerebral (AVC) pode ser definido como o aparecimento súbito de um *déficit* neurológico provocado por uma alteração nos vasos do cérebro (MIRANDA, 2025). Trata-se de uma emergência neurológica em que cada minuto conta. Assim, quanto mais rapidamente o quadro for reconhecido, triado e encaminhado ao serviço apropriado, maiores são as chances de reduzir sequelas e melhorar o prognóstico. Nesse contexto, a triagem pré-hospitalar é fundamental, pois permite identificar com rapidez os casos suspeitos de AVC, priorizar o transporte para centros capazes de oferecer terapias reperfusivas, ou seja, a desobstrução do vaso cerebral ocluído (por exemplo, trombólise e trombectomia) e preparar a equipe para intervenções imediatas, tornando o atendimento mais ágil e eficaz.

A Inteligência Artificial tem transformado a prática médica ao possibilitar o processamento e a análise de grandes volumes de dados clínicos (*big data*) para detectar padrões, apoiar hipóteses diagnósticas e automatizar tarefas de triagem e interpretação de imagens (Lobo, 2017). Juntas, elas podem apoiar decisões clínicas, reduzir erros diagnósticos e gerar recomendações mais rápidas e consistentes. No diagnóstico por imagens, as redes profundas têm apresentado desempenho próximo ao de especialistas, com sensibilidade e especificidade elevadas em tarefas como detecção de câncer de pele, por exemplo. Hoje, há aplicações clínicas regulamentadas e *pipelines* de Processamento de Linguagem Natural (PLN) que auxiliam triagem e classificação (Jiang *et al.,* 2017).

A aplicação de modelos de aprendizagem de máquina em atendimentos pré-hospitalares vem se mostrando uma maneira promissora de identificar AVCs precocemente, uma vez que estes são capazes de identificar padrões complexos e fornecer previsões, a partir de dados clínicos e demográficos (Alobaida *et al.,* 2024). Portanto, neste trabalho, buscou-se determinar se modelos de aprendizagem de máquina podem suportar a identificação precoce de AVCs com base em dados de triagem pré-hospitalar. Para tanto, foram implementados algoritmos de aprendizagem de máquina sobre conjuntos de dados públicos e calculadas medidas de desempenho para analisar os resultados obtidos.

Diante do exposto, este trabalho está organizado da seguinte forma: além desta introdução, a seção 2 apresenta o resultado do levantamento bibliográfico realizado, cujos trabalhos trouxeram inspiração para produção deste projeto, apresentando formas semelhantes de resolução do mesmo problema. A seção 3 elenca a metodologia usada nesta proposta e a seção 4 relata os detalhes de implementação e o passo a passo utilizado para construção dos modelos de aprendizagem. Por fim, na seção 5 serão apresentados as considerações finais e trabalhos futuros.

2. REFERENCIAL TEÓRICO

Embora a literatura reúna contribuições relevantes, observa-se predominância de descrições de técnicas e resultados, por isso, esta seção promove uma problematização dos limites de generalização de estudos frequentemente baseados em amostras retrospectivas e contextos geográficos específicos. Em termos metodológicos, o estudo aplica e compara algoritmos de aprendizagem de máquina em dados públicos (Kaggle), após etapas de pré-processamento, balanceamento de classes e seleção de atributos via SelectKBest, avaliando KNN, *Random Forest*, SVM e XGBoost. A discussão subsequente dialoga diretamente com os trabalhos que serão apresentados adiante; em especial, Hayashi *et al.,* (2021), Tarkanyi *et al.,* (2022), Uchida *et al.,* (2022) e Chen *et al.,* (2018).

Hayashi *et al.,* (2021) propuseram algoritmos, que realizam diagnósticos baseados em aprendizagem de máquina, para melhorar a precisão na identificação pré-hospitalar de acidentes vasculares cerebrais. Este estudo utilizou quatro diferentes modelos de aprendizagem de máquina: Regressão Logística, *Random Forest*, *Support Vector Machine* (SVM) e *eXtreme Gradient Boosting* (XGBoost). Dentre essas abordagens, o modelo XGBoost obteve o melhor desempenho, com uma acurácia de 0,987 no treino e 0,952 no teste. Os principais fatores preditivos identificados incluíram sintomas como cefaleia súbita, paralisia dos membros superiores, convulsões, alteração súbita da consciência, pressão arterial elevada e presença de arritmia cardíaca. No entanto, uma limitação relevante destacada pelos autores é a aplicabilidade do resultado restrita a uma única região urbana no Japão, levantando questões sobre a generalização dos resultados para outros contextos geográficos ou populações com características distintas (Hayashi *et al.,* 2021). Além disso, o modelo

desenvolvido é relativamente complexo, envolvendo um número elevado de variáveis preditoras, o que pode dificultar sua implementação prática em contextos emergenciais reais, onde a simplicidade operacional é essencial.

Tarkanyi *et al.,* (2022) investigaram o potencial de métodos de aprendizado de máquina para otimizar a detecção precoce de casos de AVC isquêmico agudo. O estudo utilizou dados coletados de 526 pacientes provenientes de um registro multicêntrico. Os autores avaliaram 41 variáveis clínicas, divididas em quatro grupos principais: parâmetros vitais e demográficos, histórico médico, valores de exames laboratoriais e sintomas clínicos. O método LASSO (*Least Absolute Shrinkage and Selection Operator)* foi empregado para selecionar nove variáveis com maior capacidade preditiva para a detecção de AVC, incluindo seis sintomas neurológicos (como fraqueza em membros superiores, desvio do olhar e afasia), fibrilação atrial, insuficiência cardíaca crônica e contagem de leucócitos. Diferentes modelos de aprendizagem de máquina, como Regressão Logística, *Random Forest* e Redes Neurais Simples foram aplicados para validar o desempenho do modelo, resultando em valores elevados de área sob a curva ROC (AUC), variando entre 0,736 e 0,775 após validação cruzada (Tarkanyi *et al.,* 2022).

Apesar do alto desempenho obtido, o estudo de Tarkanyi *et al.,* (2022) reconheceu limitações, como a natureza retrospectiva da análise, o uso restrito de dados provenientes de uma única região geográfica (Hungria), e a complexidade operacional decorrente do número significativo de variáveis. Assim, embora o estudo demonstre claramente que sintomas neurológicos são fundamentais para o diagnóstico precoce de AVC, ele também evidencia que fatores como histórico médico e resultados laboratoriais podem contribuir significativamente para melhorar a precisão preditiva desses modelos (Tarkanyi *et al.,* 2022).

Uchida *et al.,* (2022) desenvolveram o *Japan Urgent Stroke Triage Score* utilizando modelos de aprendizado de máquina (JUST-ML), com o objetivo de prever simultaneamente a probabilidade e o tipo específico de acidente vascular cerebral (AVC) na fase pré-hospitalar. Foram aplicados três algoritmos diferentes: Regressão Logística, *Random Forest* e XGBoost. Estes modelos apresentaram o seguinte as seguintes acurácias no conjunto de treinamento Regressão Logística: 0,615, *Random Forest*: 0,615 e XGBoost 0,623. Já no conjunto de teste os três modelos apresentaram o mesmo resultado, ou seja 0,65 de acurácia. O modelo XGBoost por apresentar melhor generalização no conjunto de teste é considerado o vencedor dentre os demais. Entre as variáveis preditoras incluídas no estudo estão idade, pressão arterial elevada, presença de arritmia, alteração de consciência, afasia, disartria e paralisia dos membros superiores. Uma vantagem significativa do JUST-ML é sua capacidade de simultaneamente discriminar entre tipos específicos de AVC, como hemorragia intracraneana (ICH), hemorragia subaracnóidea (SAH) e outros (Uchida *et al.,* 2022). Apesar do alto desempenho preditivo, Uchida *et al.,* (2022) reconhecem limitações relacionadas ao número relativamente grande de variáveis (19 ao total), que pode ser

operacionalmente desafiador no contexto emergencial (Uchida *et al.,* 2022). Além disso, os resultados foram validados em populações urbanas específicas no Japão, sugerindo a necessidade de avaliar a generalização dos modelos em outras regiões.

Chen *et al.,* (2018) desenvolveram um modelo inovador para predição pré-hospitalar de AVC utilizando Redes Neurais Artificiais (ANN). O estudo utilizou dados acessíveis no ambiente pré-hospitalar, incluindo fatores demográficos, itens da escala *National Institutes of Health Stroke Scale* (NIHSS), histórico médico e fatores de risco vascular, para treinar e validar a ANN. A metodologia empregada incluiu validação cruzada de 10 vezes para garantir a robustez e generalização do modelo. O desempenho da Rede Neural mostrou-se superior em relação às escalas pré-hospitalares tradicionais (como FAST-ED, RACE e CPSSS), apresentando uma acurácia de 0.82 +- 0.053. A ANN conseguiu capturar relações complexas entre as variáveis preditoras e a target (ALVO), superando métodos tradicionais que assumem relações lineares entre as variáveis preditoras (Chen *et al.,* 2018).

Na Tabela 1, é possível observar o desempenho dos melhores modelos de cada trabalho mencionado anteriormente, destacando as maiores acurácias obtidas em cada trabalho e seus respectivos modelos. Neste contexto, o presente artigo busca replicar na prática utilizando dados do Kaggle, os experimentos realizados nos estudos anteriores, utilizando alguns modelos de aprendizagem de máquina (como modelos de *K-Nearest Neighbour*, *Random Forest*, *Support Vector Machine* e *eXtreme Gradient Boosting*).

Tabela 1. Melhor desempenho (acurácia) dos modelos de aprendizagem de máquina

Trabalho	Modelo	Acurácia (%)
Hayashi *et al. (*2021)	XGBoost	95,2% (teste), 98,7% (treino)
Tarkanyi *et al.* (2022)	Regressão Logística *Random Forest* Rede Neural Simples	(AUC: 73,6% – 77,5%) (AUC: 73,6% – 77,5%) (AUC: 73,6% – 77,5%)
Uchida *et al.* (2022)	XGBoost	62,3% (treino), 65,0% (teste)
Chen *et al*. (2018)	Rede Neural Artificial	82,0%

Fonte: Autoria própria

A aplicação prática de modelos de triagem no Brasil deve observar a Lei Geral de Proteção de Dados Pessoais (LGPD), pois dado de saúde é considerado um tipo de dado sensível. Além disso, o marco regulatório da ANVISA para *Software as a Medical Device* (SaMD), notadamente a RDC n.º 657/2022, que disciplina classificação de risco, avaliação clínica, cibersegurança e requisitos de desempenho. Neste estudo, não foram empregadas técnicas de visão computacional

nem utilizadas imagens de tomografia computadorizada (CTA), restringindo-se a variáveis clínicas estruturadas. Ainda assim, a adoção de IA em medicina requer salvaguardas éticas adicionais: finalidade clínica claramente delimitada, validação externa, monitoramento pós-implementação, supervisão humana e gestão de risco para evitar vieses e danos não intencionais. Em conjunto, esses parâmetros regulatórios e éticos moldam tanto projetos acadêmicos voltados à transferência tecnológica quanto estudos de implementação em serviços de saúde brasileiros. (Brasil, Lei n.º 13.709/2018; ANVISA, RDC n.º 657/2022).

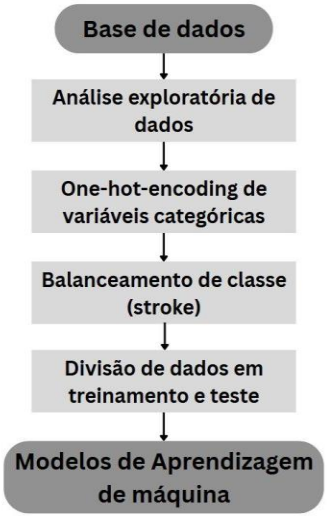
3. MÉTODOS

Esta pesquisa caracteriza-se como quantitativa, aplicada e de abordagem experimental em ambiente computacional. Silva (2014) *apud* Miranda e Ribeiro (2024) definem a pesquisa quantitativa como aquela que utiliza medidas para quantificar resultados e emprega instrumentos estatísticos para inferir conclusões sobre os dados. Do mesmo modo, Gil (2017) descreve a pesquisa experimental como aquela em que o pesquisador manipula variáveis que influenciam o objeto de estudo, obtendo resultados por meio do controle e da observação dessas variáveis.

Após a etapa de levantamento bibliográfico, foi escolhido um conjunto de dados que mais se alinhava aos objetivos do projeto, o "*Stroke Prediction Dataset*", disponibilizado publicamente no Kaggle. Essa base reúne 4981 entradas, cada uma composta por 11 variáveis. Das variáveis mencionadas, são consideradas numéricas (contínuas): *Age*, *avg_glucose_level* e bmi. Seguida pelas variáveis binárias: *hypertension*, *heart_disease* e *Stroke*, e pelas variáveis categóricas: *gender*, *ever_married*, *work_type*, *residence_type* e *smoking_status*. Os testes experimentais foram conduzidos nos ambientes Google Colab e *Jupyter Notebook*, além de serem usadas bibliotecas da linguagem Python para pré-processamento, balanceamento de classes e implementação dos modelos de aprendizado de máquina.

Na Figura 1, podem ser visualizadas as etapas que foram consideradas para início e conclusão desta proposta. Nesse sentido, após a análise exploratória dos dados, foi considerada a etapa de pré-processamento, que utilizou a técnica *One Hot Encoding*, para converter as variáveis categóricas em novas variáveis binárias. Os resultados alcançados indicaram um desbalanceamento de algumas variáveis, que foi corrigido com sub-amostragem de classes majoritárias e sobre amostragem de classes minoritárias. Dessa forma, foram realizadas mudanças nas variáveis para balanceamento dos dados e, ao final do pré-processamento, o conjunto de dados ficou com 19 variáveis, e esse conjunto foi usado na implementação dos modelos.

Figura 1. Etapas seguidas até a implementação dos modelos



Fonte: Autoria própria

Baseado na revisão sistemática conduzida por Alobaida *et al.* (2024), foram selecionados quatro algoritmos de aprendizagem de máquina: *K-Nearest Neighbors* (KNN), *Random Forest*, XGBoost e *Support Vector Machine* (SVM) por terem sido os mais populares nesta aplicação. Para todos os modelos, os dados foram divididos em 80% para treinamento e 20% para teste. Inicialmente, foi selecionado o conjunto de hiperparâmetros, que resulta na melhor acurácia em cada modelo, utilizando todas as características, e então foi utilizado o algoritmo *SelectKBest* para comparar as acurácias obtidas ao treinar os modelos apenas com as K (K = 1,...,19) variáveis mais relevantes.

4. RESULTADOS E DISCUSSÃO

Esta seção detalha os resultados obtidos neste trabalho. São destacadas as etapas de pré-processamento sobre os dados e limpeza realizada. Após isso, são mostrados os experimentos utilizando cada modelo e, por fim, é apresentada uma discussão sobre os resultados alcançados.

4.1. Pré-processamento e Limpeza dos Dados

Para realização dos experimentos, a primeira etapa foi a definição da base de dados cuja escolha foi obtida a partir da plataforma Kaggle e está relacionada a um estudo sobre AVCs. O conjunto de dados contém informações sobre 4.981 registros de pacientes, com diversas características, que ajudam a prever se um paciente sofreu um acidente vascular cerebral ou não (isso ocorreu por meio de uma classificação binária).

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

O conjunto de dados original possui 11 variáveis, sendo elas as características preditoras (que serão utilizadas pelo modelo para predição da variável alvo) e a variável alvo, chamada *stroke*, indicando se o paciente teve um AVC ou não. As características são apresentadas a seguir: *gender*: Gênero do paciente; *age*: Idade do paciente; *hypertension*: Indicação de se o paciente tem hipertensão; *heart_disease*: Indicação de se o paciente tem doença cardíaca; *ever_married*: Indicação de se o paciente já foi casado; *work_type*: Tipo de ocupação do paciente; *residence_type*: Tipo de residência; *avg_glucose_level*: Nível médio de glicose no sangue; bmi: Índice de massa corporal; *smoking_status*: Status de tabagismo do paciente; *stroke*: Variável alvo, indica se o paciente sofreu um AVC ou não.

Após o carregamento dos dados no ambiente de trabalho, foi preciso fazer uma inspeção para saber quantos exemplos há no conjunto de dados e quantos destes apresentam dados faltantes em uma ou mais variáveis analisadas. Diante dos resultados obtidos, ficou evidente que o conjunto de dados possui um total de 4.981 registros e, dentre estes, não houve nenhum dado faltante. Além disso, algumas colunas no conjunto de dados são do tipo categórico (por exemplo, *gender*, *ever_married*, *work_type*, *residence_type*, *smoking_status*). Para que esses dados sejam usados em modelos de *Machine Learning* (ML), as variáveis categóricas precisam ser convertidas para um formato numérico. Uma das abordagens mais comuns para isso é o uso do *One Hot Encoding*, que cria colunas binárias para cada categoria presente em uma variável. Por exemplo, a coluna *gender* contém as categorias "Male" e "Female", e o *One Hot Encoding* gerará duas novas colunas: "*gender_Male*" e "*gender_Female*", com valores 0 ou 1, indicando a presença ou ausência de cada categoria. Seguindo a mesma lógica, para todas as variáveis categóricas do conjunto de dados, foi aplicado o *One Hot Encoding*.

Durante a análise da variável preditora *stroke* foi possível perceber que há um desbalanceamento nesta classe, onde a classe 0 (pacientes que não apresentaram AVC) possui 4.733 amostras, enquanto a classe 1 (pacientes que apresentaram AVC) tem apenas 248 amostras. Essa diferença na quantidade de exemplos de cada classe pode levar a predições enviesadas. Para resolver este problema, foram utilizadas, simultaneamente, técnicas de balanceamento de classes, considerando o *oversampling*, que aumenta a quantidade de amostras da classe minoritária (classe "1"); e *undersampling*, que reduz a quantidade de amostras da classe majoritária (classe "0").

Para isso, foram utilizados os métodos *RandomUnderSampler* e *RandomOverSampler*, que fazem a escolha aleatória de alguns exemplos, onde o *RandomOverSampler* duplica aleatoriamente exemplos da classe minoritária, enquanto o *RandomUnderSampler* visa remover aleatoriamente amostras da classe majoritária, para que haja um equilíbrio das classes. Após o balanceamento, as classes se encontraram equilibradas com a mesma quantidade de exemplos em cada classe.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

Por fim, foi feita a remoção da variável id, já que ela não será utilizada pelo modelo de aprendizagem, e então o novo conjunto de dados resultante foi salvo em CSV para que fosse utilizado em todos os modelos que serão apresentados.

4.2. Modelo KNN

O *K-Nearest Neighbors* (KNN) é amplamente aplicado em reconhecimento de padrões e mineração de dados para tarefas de classificação, sendo reconhecido por sua simplicidade e baixo erro empírico. O algoritmo atribui a classe de um novo exemplo com base nas classes dos seus k vizinhos mais próximos no espaço de características, a escolha de k permite ajustar o equilíbrio entre viés e variância, controlando a propensão ao *overfitting* (Kuang; Zhao, 2009).

Inicialmente, o modelo KNN foi configurado com os parâmetros padrões da biblioteca *Scikit-Learn*. Após isso, foi necessário otimizar estes parâmetros utilizando o algoritmo *Grid Search*, permitindo realizar uma busca exaustiva sobre um conjunto de hiperparâmetros definidos pelo usuário (neste caso, número de vizinhos, pesos, algoritmos e distância). Sendo assim, ele testa todas as combinações possíveis de hiperparâmetros e seleciona a combinação que resulta no melhor desempenho do modelo, ou seja, na melhor acurácia.

Os principais hiperparâmetros utilizados no KNN são mostrados a seguir:

- *n_neighbors*: O número de vizinhos a serem considerados na decisão de classificação. Foram testados os valores 5, 7, 9 e 11.
- *weights*: Define como a distância entre os pontos influencia a decisão. O parâmetro pode ser "*uniform*" (todos os vizinhos têm o mesmo peso) ou "*distance*" (os vizinhos mais próximos têm maior influência). O parâmetro "*distance*" geralmente é mais utilizado.
- *algorithm*: O algoritmo utilizado para calcular os vizinhos mais próximos. Os algoritmos disponíveis são: *ball_tree*, *kd_tree* e *brute*.
- p: Representa o parâmetro de distância. Quando p=1, a distância de Manhattan (soma das diferenças absolutas) é usada; quando p=2, a distância Euclidiana (raiz quadrada da soma dos quadrados das diferenças) é utilizada.

A otimização foi realizada com o uso do *GridSearchCV*, que executa uma busca exaustiva sobre as combinações de todos os parâmetros mencionados acima, utilizando validação cruzada para avaliar o desempenho do modelo em diferentes subconjuntos dos dados de treino, ajudando a evitar o *overfitting*.

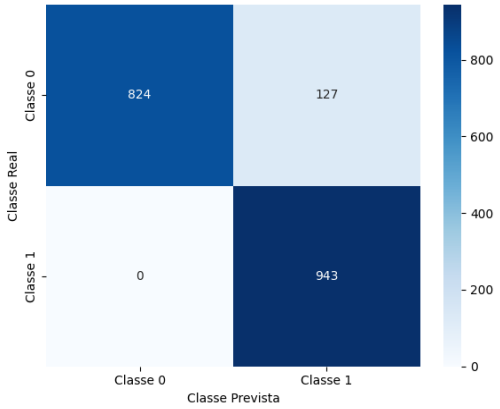
A validação cruzada utilizada foi do tipo "5-fold", ou seja, os dados foram divididos em 5 partes, e o modelo foi treinado 5 vezes, com os dados separados em 5 partes (4 para treino e 1 para teste). O melhor desempenho foi selecionado com base na métrica de acurácia.

Os resultados do Grid Search são elencados a seguir: *n_neighbors*: 5, *weights*: *distance*, *algorithm*: *kd_tree*, p: 2 (distância Euclidiana).

ISSN: 2675-6218 - RECIMA21

A Figura 2 apresenta a matriz de confusão alcançada após os experimentos do KNN. Conforme mostrado, foram previstos 824 verdadeiros negativos, 943 verdadeiros positivos, 127 falsos positivos, e nenhum falso negativo. Desse modo, o modelo encontrou uma acurácia de 93,29%.

Figura 2. Matriz de confusão do KNN



Fonte: Autoria própria

Além do experimento que utiliza todo o conjunto de dados, foi necessário realizar um segundo experimento, com o objetivo de avaliar a performance do modelo KNN, utilizando um número reduzido de características selecionadas por meio do algoritmo *SelectKBest*. Essa técnica escolhe as k melhores características do conjunto de dados, com base em um critério estatístico. No caso desse experimento, o critério utilizado foi o *f_classif*, que é um teste para avaliar a relação entre cada característica e a variável alvo. O resultado retornado do *SelectKBest* é uma lista com as K melhores características. Exemplo: para k = 5, as 5 melhores características são utilizadas, para k = 11, as 11 melhores.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

Tabela 2. Acurácias para todos os valores de K do *SelectKBest* do KNN

Valores de K	Acurácia
1	75,71%
2 e 3	78,25%
4	94,19%
5	94,14%
6	93,93%
7	93,88%
8	93,82%
9	93,61%
10	93,66%
11, 14 e 16	93,51%
12 e 17	93,35%
13 e 18	93,40%
15	93,45%
19	93,29%

Fonte: Autoria própria

A partir disso, o *SelectKBest* foi utilizado em um *loop* para testar diferentes valores de k de 1 a 19, já que existem 19 características no conjunto de dados, considerando as adicionadas no *One Hot Encoding*, e a lista das acurácias pode ser vista na Tabela 2. Para cada valor de k no loop, um novo modelo de KNN é treinado e o conjunto de dados vai possuir k características, então se for a décima iteração e k for igual a 10, o modelo vai ser treinado com um conjunto de dados que contém apenas as 10 melhores características. O objetivo é verificar como se comporta a acurácia quanto mais características são removidas para futuramente implementar um sistema que consiga fazer uma boa predição com o mínimo de características possível.

Todos os modelos a seguir utilizaram a mesma estratégia de fazer um primeiro experimento para descobrir com o *Grid Search* os melhores parâmetros e depois submetidos a um *loop* com as K melhores características do *SelectKBest*.

ISSN: 2675-6218 - RECIMA21

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

4.3. Modelo *Random Forest*

O *Random Forest* é um meta-estimador que ajusta múltiplos classificadores do tipo *Decision Tree* em subamostras do conjunto de dados e combina suas previsões — por média (regressão) ou votação majoritária (classificação) — para melhorar a precisão preditiva e controlar o *overfitting* (SILVA; NETO, 2022). A ideia principal do *Random Forest* é criar várias árvores de decisão a partir de diferentes subconjuntos dos dados de treino, e então combinar suas previsões para obter uma decisão mais confiável.

No experimento com o *Random Forest*, os dados foram carregados da mesma forma que no experimento anterior. Em seguida, os dados foram divididos em conjuntos de treino e teste com 80% dos dados para treinamento e 20% para teste. Isso garante que o modelo seja avaliado com dados não vistos. Além disso, o *GridSearchCV* foi utilizado novamente para otimizar os hiperparâmetros do modelo *Random Forest* e da seleção de características. Os principais parâmetros testados são elencados a seguir.

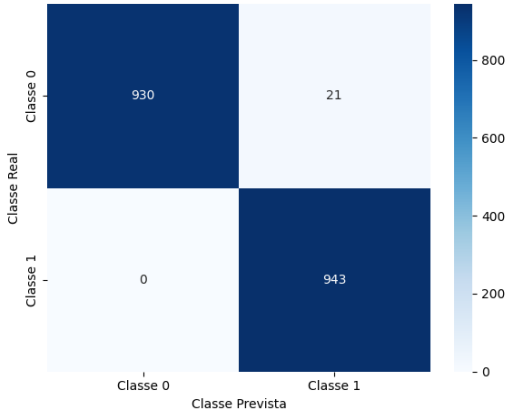
- *rf_n_estimators*: O número de árvores na floresta. No experimento, foram utilizadas apenas 50 árvores, que é um número comum em modelos de *Random Forest.*
- *rf_max_depth*: A profundidade máxima das árvores. Os valores testados foram *None* (sem limite de profundidade), 5, 10, e 20. A profundidade de uma árvore afeta diretamente o *overfitting*; árvores muito profundas podem aprender demais sobre os dados de treino, o que pode prejudicar a generalização.
- *rf_criterion*: O critério de divisão dos nós da árvore. Os valores possíveis são *"gini"* (impureza de Gini) e *"entropy"* (entropia), que são duas maneiras de calcular a "pureza" de uma divisão. O *"gini"* é mais comum em *Random Forests*.

  Os resultados do Grid Search são mostrados a seguir.
- *rf_n_estimators*: O número de árvores na floresta foi mantido em 50.
- *rf_max_depth*: Não houve limite para a profundidade das árvores, ou seja, o *Random Forest* pode ter um valor de profundidade que o modelo melhor se ajustar, sem limites.
- *rf_criterion*: O critério de Gini foi o mais eficaz na divisão de nós.

A Figura 3 apresenta a matriz de confusão do *Random Forest*, onde foram previstos 930 verdadeiros negativos, 943 verdadeiros positivos, 21 falsos positivos, e nenhum falso negativo. Desse modo, o modelo encontrou uma acurácia de 98,84%.

ISSN: 2675-6218 - RECIMA21

Figura 3. Matriz de confusão do *Random Forest*



Fonte: Autoria própria

Os melhores hiperparâmetros encontrados foram K=18 (isto é, todas as características) e número máximo de árvores 50. Com esses valores, realizou-se um segundo experimento análogo ao aplicado ao modelo KNN, empregando o método *SelectKBest* para seleção de características. Para avaliar o impacto do número de variáveis, implementou-se um loop que, para cada valor de K é treinado um novo modelo com as K melhores características; essa abordagem visa identificar o menor subconjunto de características possível para uma futura implementação em sistemas embarcados, onde os recursos de hardware são limitados, uma vez que estes possuem processadores mais simples e com menor poder de processamento, pouca capacidade de armazenamento, e limite de componentes e de consumo de energia, o que requer um conjunto de dados reduzido. As acurácias obtidas com o *SelectKBest* estão apresentadas na Tabela 3.

Tabela 3. Acurácias para todos os valores de K do *SelectKBest* do *Random Forest*

Valores de K	Acurácia
1	77,72%
2 e 3	78,78%
4	96,94%
5 e 7	97,10%
6	96,78%
8	98,52%
9, 16 e 19	98,84%
10	98,57%

ISSN: 2675-6218 - RECIMA21

11	98,63%
12, 15, 17 e 18	98,89%
13	98,68%
14	98,79%

Fonte: Autoria própria

4.4. Modelo XGBoost

O XGBoost (*Extreme Gradient Boosting*) é um dos algoritmos mais eficazes para problemas de aprendizado supervisionado, especialmente em tarefas de classificação e regressão. Ele é uma implementação eficiente do algoritmo de *Gradient Boosting*, que combina várias árvores de decisão em uma sequência, onde cada árvore tenta corrigir os erros cometidos pelas árvores anteriores. O XGBoost se destaca por sua velocidade e desempenho, sendo amplamente utilizado em competições de aprendizado de máquina (Machado *et al.,* 2021).

A principal ideia por trás do XGBoost é otimizar a forma como as árvores são construídas, ao usar técnicas como regularização (para evitar *overfitting*), paralelização para acelerar o treinamento e ajustes finos nas taxas de aprendizado e profundidade das árvores. Analogamente aos modelos anteriores, foi criada uma *pipeline* que utiliza o *Grid Search* para encontrar os melhores parâmetros para o modelo. Dentre os hiperparâmetros a serem avaliados, destacam-se:

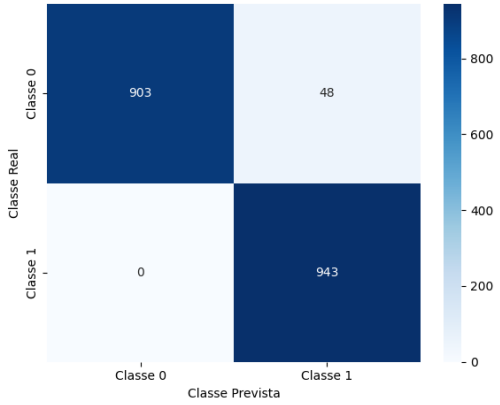
- *xgb_n_estimators*: O número de estimadores (árvores) a serem construídos pelo XGBoost. Foi testado o valor de 50 árvores.
- *xgb_max_depth*: A profundidade máxima das árvores. Foram testados os valores 3, 5, e 10.
- *xgb_learning_rate*: A taxa de aprendizado, que controla o quanto cada árvore contribui para o modelo final. Foram testados os valores 0.01, 0.1, e 0.3.

O processo de *Grid Search* analogamente aos anteriores continua utilizando validação cruzada com 5 *folds*, ou seja, 4 partes dos dados são para treino e 1 para teste, cujos resultados são:

- *xgb_n_estimators*: 50;
- *xgb_max_depth*: 10;
- *xgb_learning_rate*: 0.3.

A Figura 4 apresenta a matriz de confusão do XGBoost, onde foram previstos 903 verdadeiros negativos, 943 verdadeiros positivos, 48 falsos positivos, e nenhum falso negativo. Desse modo, o modelo encontrou uma acurácia de 95,78%.

Figura 4. Matriz de confusão do XGBoost



Fonte: Autoria própria

Após a obtenção dos melhores parâmetros do modelo XGBoost, foi feito novamente o experimento com o *SelectKBest*, e os resultados obtidos estão na Tabela 4.

Tabela 4. Acurácias para todos os valores de K do *SelectKBest* do XGBoost

Valores de K	Acurácia
1	77,72%
2 e 3	78,78%
4 e 5	93,72%
6 e 7	93,66%
8 e 11	95,25%
9 e 15	94,98%
10	94,77%
12, 13, 16 e 17	95,46%
14	95,72%
18	95,35%
19	95,78%

Fonte: Autoria própria

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

4.5. Modelo SVM (*Support Vector Machine*)

O SVM (*Support Vector Machine*) também é um modelo de aprendizagem supervisionada, e tem como objetivo encontrar o hiperplano de separação ideal que maximiza a margem entre as classes. Para encontrar o hiperplano ótimo, o algoritmo separa as classes maximizando a margem entre os pontos de diferentes classes mais próximas (vetores de suporte), de modo que os novos pontos são classificados com base em qual lado do hiperplano caem (Addan, 2019).

Para a implementação do modelo foi utilizado a classe SVC (*Support Vector Classification*) da biblioteca *scikit-learn*, e também foi utilizado o *Grid Search* para melhorar a acurácia do modelo, cujos hiperparâmetros utilizados são mostrados a seguir.

- *kernel*: Define como o SVM transforma os dados de entrada para encontrar uma fronteira de decisão, e pode ser: “linear”, que usa os próprios dados no espaço original (fronteira linear), “poly”, que usa um polinômio para o mapeamento e “rbf” (*Radial Basis Function*), que mapeia os dados para um espaço de alta dimensão (não linear), capaz de capturar relações complexas.
- C: Controla o *trade-off* entre a margem de separação e o erro de classificação, e pode ter valores mais baixos (que permite mais erros de classificação nos dados de treino e resulta em um modelo com mais generalização) ou mais altos (que penaliza fortemente os erros e tenta classificar tudo corretamente).
- *Gamma*: Define quanto influência um único ponto de dado tem, e também pode ter valores mais altos (onde a influência é pequena e local e o modelo mais complexo, podendo causar *overfitting*) ou mais baixos (com a influência de cada ponto ampla, gerando um modelo mais simples, e menos propenso a *overfitting*).

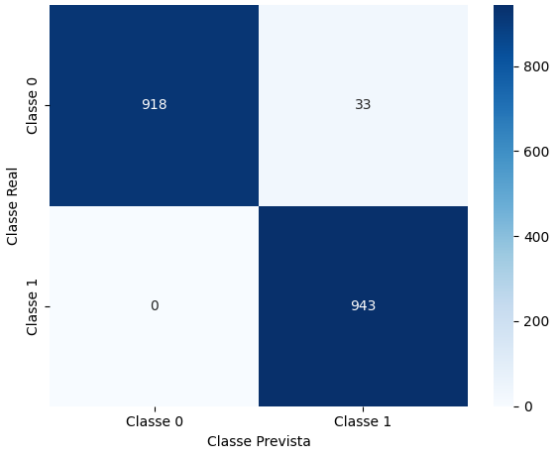
Dessa forma, os hiperparâmetros encontrados com o *Grid Search* são:

- *Kernel*: “rbf”;
- C: 10;
- *Gamma*: ’auto’.

A Figura 5 apresenta a matriz de confusão do SVM, onde foram previstos 918 verdadeiros negativos, 943 verdadeiros positivos, 33 falsos positivos, e nenhum falso negativo. Desse modo, o modelo encontrou uma acurácia de 98,26%. Com os melhores hiperparâmetros escolhidos, foi implementado o *SelectKBest* para o SVM, e as acurácias podem ser vistas na Tabela 5.

ISSN: 2675-6218 - RECIMA21

Figura 5. Matriz de confusão do SVM



Fonte: Autoria própria

Tabela 5. Acurácias para todos os valores de K do *SelectKBest* do SVM

Valores de K	Acurácia
1	77,72%
2 e 3	78,78%
4	96,30%
5 e 6	96,04%
7	96,46%
8, 12, 13, 15 e 16	98,47%
9, 14 e 18	98,42%
10 e 11	98,36%
17 e 19	98,26%

Fonte: Autoria própria

4.6. Discussão dos Resultados

Com base nos experimentos realizados, foi possível observar que os modelos apresentaram acurácias consideradas altas, em relação aos trabalhos apresentados na revisão sistemática (*Random Forest* 99%, SVM 98%, XGBoost 97%, KNN 95%) à maioria dos estudos citados, por exemplo, Hayashi *et al.,* (XGBoost 95,2% teste / 98,7% treino), Uchida *et al.,* (XGBoost ≈62–65%) e Chen *et al.* (RNA 82%). Essas diferenças provavelmente decorrem de variações nos conjuntos de dados, pré-processamento e implementação de cada modelo, além do uso de *Grid*

Search e *SelectKBest* neste trabalho, que contribuíram para o melhor desempenho mesmo com K reduzido.

A Tabela 6 apresenta os resultados alcançados em cada modelo, cujas informações foram obtidas a partir da combinação dos melhores hiperparâmetros, após a aplicação do algoritmo *Grid Search.* O resultado mostrado utiliza todas as características presentes no conjunto de dados.

Tabela 6. Comparação de desempenho entre modelos de classificação.

Modelo	**Acurácia**	**Precisão**		**Recall**		**F1-Score**	
		Classe 0	Classe 1	Classe 0	Classe 1	Classe 0	Classe 1
Random Forest	0.99	1.00	0.98	0.98	1.00	0.99	0.99
SVM	0.98	1.00	0.97	0.97	1.00	0.98	0.98
KNN	0.95	1.00	0.90	0.89	1.00	0.94	0.95
XGBoost	0.97	1.00	0.95	0.95	1.00	0.97	0.98

Fonte: Autoria própria

A Tabela 7 foi construída com base nos hiperparâmetros definidos pelo *Grid Search* da Tabela 6, mas com a adição da seleção de características utilizando o algoritmo *SelectKBest*. Nesse caso, é possível entender, de forma comparativa, as acurácias de cada modelo sem o *SelectKBest*, ou seja, com K = 19 (utilizando todas as características), e também com o *SelectKBest*, onde é possível observar “melhor K”, indicando que a partir de cada valor de “K” não é possível obter melhora significativa na acurácia. Além disso, também é possível observar um “menor valor aceitável de K”, onde não se buscava uma acurácia ótima, mas um valor que equilibra o modelo com uma acurácia aceitável e uma quantidade pequena de características (neste caso as 4 melhores), para o modelo em sistema embarcado.

Tabela 7. Desempenho dos modelos com e sem *SelectKBest*, detalhando o valor de K e a acurácia

Modelo	**Sem *SelectKBest***	**Com *SelectKBest***			
		Melhor K		**Menor valor aceitável de K**	
		Valor de K	Acurácia	Valor de K	Acurácia
Random Forest	98,84%	12	98,89%	4	96,94%
SVM	98,26%	8	98,47%	4	96,30%
KNN	93,29%	4	94,19%	4	94,19%
XGBoost	95,78%	19	95,78%	4	93,72%

Fonte: Autoria própria

Diante do exposto e dos resultados apresentados, percebe-se que os quatro modelos têm como menor valor aceitável de K = 4, ou seja, os parâmetros que são mais relevantes para a classificação são: idade, hipertensão, histórico de doenças cardiovasculares e nível de glicose no sangue.

Na comparação entre os modelos testados, é possível perceber que o *Random Forest* apresentou o melhor desempenho dentre eles, principalmente quando comparado com o KNN, o que pode ser explicado por sua menor sensibilidade a *outliers*, que não foram eliminados no pré-processamento, mas que todos os modelos apresentaram uma acurácia acima de 94% ao aplicar técnicas de ajuste de parâmetros.

Por fim, ao realizar um comparativo com os resultados obtidos nesta pesquisa, com os que foram pesquisados na fase de revisão bibliográfica, fica evidente que o modelo *Random Forest* apresentou uma acurácia de 98,84%, superando os resultados reportados em outros trabalhos relacionados a este. Por exemplo, Hayashi *et al.,* (2021) reportaram XGBoost com acurácia de teste de 95,2%, e Chen *et al.* (2018) obtiveram uma rede neural com acurácia de 82,0%. Contudo, vale ressaltar que diferenças nos conjuntos de dados, tamanho da amostra, balanceamento de classes, e a origem dos dados podem explicar parte dos resultados encontrados.

5. CONSIDERAÇÕES

O AVC é uma doença neurológica causada pela interrupção do fluxo sanguíneo ao cérebro, podendo provocar déficit motor, alterações da fala, comprometimento cognitivo e até óbito, gerando dependência e grande impacto social. Como as intervenções eficazes dependem de uma detecção muito precoce, soluções tecnológicas que automatizem a triagem e forneçam suporte à decisão clínica no ambiente pré-hospitalar são essenciais. Sendo assim, este projeto propõe o uso e comparação de modelos de aprendizagem de máquina, para diagnóstico precoce da AVC. O intuito é identificar o modelo que apresenta o melhor desempenho, dada a manipulação deste conjunto de dados. Sendo assim, foram considerados dados públicos para realização de experimentos.

Antes de iniciar a análise e processamento dos algoritmos, foram realizados processos de pré-processamento e balanceamento das classes. Após essa etapa, quatro algoritmos de classificação foram selecionados para estudos e implementados, sendo eles: KNN, *Random Forest*, XGBoost e SVM. Além disso, a técnica *SelectKBest* foi utilizada para selecionar as variáveis mais relevantes, buscando otimizar o desempenho dos modelos, tornando-os adequados para futuras implementações em sistemas embarcados.

Os quatro modelos classificadores implementados atingiram desempenho considerado elevado para as tarefas de triagem pré-hospitalar de AVC. O *Random Forest* foi o modelo com melhor desempenho alcançando uma acurácia global de 98,9% com 12 variáveis, e mantendo 96,9% após a redução para apenas quatro variáveis mais importantes (idade, hipertensão, doença

cardíaca e glicemia média), devido ao *SelectKBest*. Assim, pode-se destacar que estes modelos podem ser facilmente embarcados em hardwares dedicados, restrito a apenas quatro entradas clínico-demográficas de fácil obtenção e conseguindo preservar desempenho próximo ao máximo observado com todas as variáveis, permitindo realizar um porte do modelo *Random Forest* otimizado (com K = 4) para aplicativos mobile ou *Raspberry Pi*, levando em consideração o número menor de variáveis.

Como trabalhos futuros, propõe-se validar prospectivamente o modelo em dados hospitalares externos, para avaliar sua generalização e fazer implementação do modelo com melhor desempenho em uma API, para ser utilizada em dispositivos embarcados.

REFERÊNCIAS

ADDAN, D. **Support Vector Machine**. [S. l.]: Unibrasil, 2019. Presentation slides (34 pages).

AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA (ANVISA). **Resolução de Diretoria Colegiada – RDC n.º 657, de 24 de março de 2022**. Dispõe sobre software como dispositivo médico (SaMD). 2022. Disponível em: https://in.gov.br/en/web/dou/-/resolucao-de-diretoria-colegiada-rdc-n-657-de-24-de-marco-de-2022-389603457.

ALOBAIDA, M.; JODDRELL, M.; ZHENG, Y.; LIP, G. Y. H.; ROWE, F. J.; EL-BOURI, W. K.; HILL, A.; LANE, D. A.; HARRISON, S. L. Systematic review and meta-analysis of prehospital machine learning scores as screening tools for early detection of large vessel occlusion in patients with suspected stroke. **J Am Heart Assoc**., v. 13, n. 12, p. e033298, jun. 2024.

BRASIL. **Lei n.º 13.709, de 14 de agosto de 2018**. Lei Geral de Proteção de Dados Pessoais (LGPD). Brasília: Casa Civil, 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm#art65.

CHEN, Z.; ZHANG, R.; XU, F.; GONG, X.; SHI, F.; ZHANG, M.; LOU, M. Novel prehospital prediction model of large vessel occlusion using artificial neural network. Frontiers in Aging Neuroscience, v. 10, p. 181, jun. 2018. Erratum in: **Front Aging Neurosci**., v. 10, p. 222, 17 jul. 2018. doi:10.3389/fnagi.2018.00222.

HAYASHI, Y.; SHIMADA, T.; HATTORI, N.; SHIMAZUI, T.; YOSHIDA, Y.; MIURA, R. E.; YAMAO, Y.; ABE, R.; KOBAYASHI, E.; IWADATE, Y.; NAKADA, T. A. A prehospital diagnostic algorithm for strokes using machine learning: a prospective observational study. **Scientific Reports**, v. 11, n. 1, p. 20519, out. 2021.

JIANG, F.; JIANG, Y.; ZHI, H.; DONG, Y.; LI, H.; MA, S.; WANG, Y.; DONG, Q.; SHEN, H.; WANG, Y. Artificial intelligence in healthcare: past, present and future. **Stroke and Vascular Neurology**, v. 2, p. e000101, 2017. Publicado online em 22 jun. 2017.

KUANG, Q.; ZHAO, L. A practical gpu based knn algorithm. *In:* HUANGSHAN, P. R. **China**. p. 151–155, Dec. 26–28 2009. AP-PROC-CS-09CN005. Supported by National Natural Science Foundation of China (No. 60873047) and Natural Science Foundation of Jiangsu Province (No. BK2008154).

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

LOBO, L. C. Inteligência artificial e medicina. **Revista Brasileira de Educação Médica**, v. 41, n. 2, p. 185–193, 2017. Texto em português. Disponível em: https://doi.org/10.1590/1981-52712015v41n2esp.

MACHADO, A. L. A.; JUNIOR, L. G. de Q. S.; NUNES, M. A. **XGBoost na Previsão da Geração de Energia Elétrica em Parques Eólicos**. [*S. l.: s. n.*], 2021. p. 1–6. Documento em português. Extraído do PDF fornecido.

MIRANDA, F. J.; RIBEIRO, R. M. (Ed.). **Manual de metodologia da pesquisa científica**: diretrizes e métodos. Sobral, CE: Faculdade Luciano Feijão, 2024. Disponível em: https://fucianofeijao.com.br/ff/wp-content/uploads/2024/02/MANUAL-DE-METODOLOGIA-DA-PESQUISA_ADMINISTRACAO.pdf.

MIRANDA, M. **Acidente Vascular Cerebral**. [*S. l.: s. n.*], 2025. Disponível em: https://avc.org.br/pacientes/acidente-vascular-cerebral/.

SILVA, R.; NETO, D. R. S. Inteligência artificial e previsão de óbito por covid-19 no brasil: uma análise comparativa entre os algoritmos logistic regression, decision tree e random forest. **Saúde em Debate**, v. 46, n. Especial 8, p. 118–129, dec. 2022.

TARKANYI, G.; TENYI, A.; HOLLOS, R.; KALMAR, P. J.; SZAPARY, L. Optimization of large vessel occlusion detection in acute ischemic stroke using machine learning methods. **Life** (Basel), v. 12, n. 2, p. 230, fev. 2022.

UCHIDA, K.; KOUNO, J.; YOSHIMURA, S.; KINJO, N.; SAKAKIBARA, F.; ARAKI, H.; MORIMOTO, T. Development of machine learning models to predict probabilities and types of stroke at prehospital stage: the japan urgent stroke triage score using machine learning (just-ml). **Translational Stroke Research**, v. 13, n. 3, p. 370–381, jun. 2022.