EVENTO ON-LINE
ConBRepro
XII CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO
ESG nas Engenharias
30 a 02 de dezembro 2022

Influência de uma base de dados desequilibrada na classificação de pacientes que tiveram AVC por meio de técnicas de *Machine Learning*

Davi Salvini Chixaro
Engenharia de produção – Universidade Federal do Paraná
Ricardo Júnior de Oliveira Silva
Engenharia de produção – Universidade Federal do Paraná
Mariana Kleina
Engenharia de produção – Universidade Federal do Paraná
Fabiano Oscar Drozda
Engenharia de produção – Universidade Federal do Paraná

Resumo: Atualmente, a área de diagnósticos de doenças tem avançado consideravelmente, e, como consequência, tem tido de lidar com volumes substancialmente grandes de informações. Por meio das técnicas tradicionais não se torna viável fazer a análise desse grande volume de informação devido ao despendimento de tempo e esforço. Sendo assim, as técnicas de *Machine Learning* se apresentam como soluções atrativas para essas dificuldades. Como exemplo prático, tem-se a interpretação automatizada do eletrocardiograma (ECG) onde o padrão de reconhecimento é feito a partir de um conjunto limitado de diagnósticos, sendo, dessa forma, uma tarefa de classificação. Mas para que a classificação seja feita de forma correta e confiável, um ponto primordial a ser levado em consideração é o equilíbrio da base de dados escolhida. Tendo isso, o presente trabalho teve como objetivo analisar a influência de uma base de dados desequilibrada na classificação de pacientes que já tiveram AVC por meio de duas técnicas de *Machine Learning*, RNA *MultiLayer Perceptron* e *Support Vector Machine*. Como conclusões, observou-se que embora os resultados com dados desiquilibrados pareçam melhores para casos que não tiveram AVC, pois são mais acurados, deve-se ter cautela pois a classificação para casos que tiveram AVC demonstrou ter uma baixíssima precisão. Além disso, pensando no escopo de saúde pública, um diagnóstico errado de uma pessoa que tem altas chances de sofrer um AVC seria considerado uma falha médica grave caso essa pessoa viesse a sofrer um AVC.

Palavras-chave: *Machine Leaning*, classificação, *MLP, SVM*, AVC.

Influence of an unbalanced database in the classification of patients who had stroke through machine learning techniques

Abstract: Currently, the area of disease diagnosis has advanced considerably, and, consequently, has had to deal with substantially large volumes of information. Through traditional techniques it is not feasible to analyze these large volumes of information due to the time and effort. Thus, machine learning techniques present themselves as attractive solutions to these difficulties. As a practical example, there is the automated interpretation of the ECG (electrocardiogram) where the recognition pattern is made from a limited set of diagnoses, thus being a classification task. But for the

classification to be done correctly and reliably, a primary point to be considered is the balance of the chosen database. Having this, the present work aimed to analyze the influence of an unbalanced database on the classification of patients who have already had stroke through two machine learning techniques, *RNA* MultiLayer Perceptron and Support Vector Machine. As conclusions, it was observed that although the results with unbalanced data seem better for cases that did not have stroke, as they are more accurate, caution should be exercised because the classification for cases that had stroke proved to have very low precision. Also, considering the public health scope, a misdiagnosis of a person who has a high chance of having a stroke would be considered a serious medical failure if that person were to suffer a stroke.

Keywords: Machine Leaning, Classification, MLP, SVM, Stroke.

1. Introdução

Machine Learning (ML), também chamado de Aprendizagem de Máquina, tem se tornado a realidade de muitos setores tais como a de assistência médica, bancária, de transporte, mídias sociais, entre outras. Atualmente, o setor médico, em específico na área de diagnósticos de doenças, tem avançado consideravelmente, e, por consequência, tem tido de lidar com volume substancialmente grande de informações. Por meio das técnicas tradicionais não se torna viável fazer a análise desse grande volume de informações devido ao despendimento de tempo e esforço. Sendo assim, as técnicas de ML se apresentam como soluções atrativas para essas dificuldades (BATTINENI, CHINTALAPUDI e AMENTA, 2019).

Ao tratar sobre ML, existem alguns tipos de aprendizagem utilizados por ela, sendo eles: aprendizado supervisionado, não supervisionado, semi-supervisionado, e por reforço. Sobre o aprendizado supervisionado, tipo escolhido para esse trabalho, segundo Deo (2015), tem um foco maior em problemas de classificação, os quais envolvem escolher entre subgrupos o melhor que descreva uma nova instância de dados, e na previsão, os quais envolvem a estimativa de um parâmetro desconhecido.

Como exemplo prático de aplicação utilizando a aprendizagem supervisionada, tem-se a interpretação automatizada do eletrocardiograma (ECG) onde o padrão de reconhecimento é feito a partir de um conjunto limitado de diagnósticos, sendo, dessa forma, uma tarefa de classificação. Nesse caso, o computador realiza a atividade que um médico treinado é capaz de fazer, claro que ainda de forma aproximada (DEO, 2015). Mas visto que é uma técnica de aprendizagem, ela pode ser aperfeiçoada com mais treinamentos.

A grande diferença entre a aprendizagem humana e a aprendizagem de máquina é que os humanos conseguem aprender para fazer desde associações gerais até mais complexas a partir de uma pequena quantidade de informações. Já as máquinas, em geral, requerem muito mais exemplos do que os humanos para aprender a mesma tarefa, além de não serem dotadas de bom senso. Coincidentemente, o que parece ser um ponto negativo para ML, é, na realidade, seu ponto de vantagem. As máquinas podem aprender com quantidades de dados excepcionalmente grandes, sendo perfeitamente viável para um modelo de ML ser treinado com dezenas de milhões de prontuários de pacientes armazenados, sem quaisquer erros por falta de atenção. O que seria praticamente impossível para um ser humano, levando em consideração que, segundo estimativa, um ser humano médico não vê mais que algumas dezenas de milhares de pacientes em uma carreira inteira (RAJKOMAR, DEAN e KOHANE, 2019).

Outro ponto primordial a ser levado em consideração ao trabalhar com modelos de classificação usando ML é referente ao equilíbrio da base de dados escolhida. Segundo Scalco (2021, p.44) “é importante analisar o conjunto de dados procurando um desequilíbrio

entre as classes, onde uma classe está super-representada, enquanto a outra está sub-representada. Dados desequilibrados afetam diretamente o desempenho final do modelo.”

Ante a essa situação, o presente trabalho teve como objetivo analisar a influência de uma base de dados desequilibrada na classificação de pacientes que já tiveram acidente vascular cerebral (AVC) por meio de duas técnicas de *ML,* RNA *MultiLayer Perceptron* e *Support Vector Machine.*

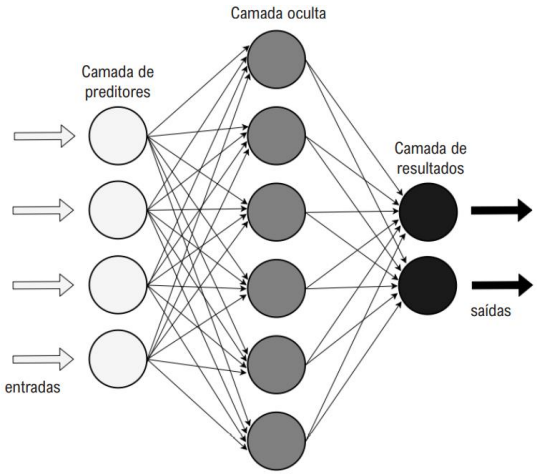
2. Revisão Bibliográfica

Nesta seção, apresenta-se brevemente os conceitos de RNA *MultiLayer Perceptron* (*MLP)* e *Support Vector Machine* (*SVM*) fundamentados na literatura.

2.1 RNA *MultiLayer Perceptron – MLP*

Uma RNA (Rede Neural Artificial) é um modelo computacional e matemático projetado para funcionar como o cérebro humano (PAIXÃO et al., 2022), além de ser também um braço da Inteligência Artificial (IA) (GARDNER e DORLING, 1998). Uma RNA possui vários elementos interconectados (camada de entrada, camada oculta e camada de saída). A relação entre essas camadas é inspirada nas conexões sinápticas entre os neurônios (PAIXÃO et al., 2022), assim como exemplificado pela Figura 1.

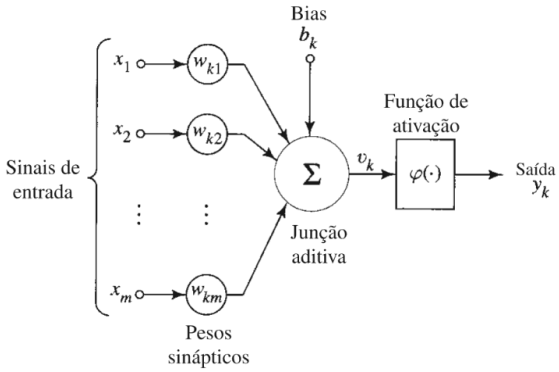
Figura 1 - Estrutura de funcionamento de uma RNA



Fonte: Paixão *et al.* (2022)

Os elementos são conectados por pesos e sinais de saída que são uma função da soma das entradas ao elemento modificado por uma transferência não linear, ou ativação (GARDNER e DORLING, 1998). Por pesos, entende-se conexões modeladas entre os neurônios, tais como matrizes de números (NORIEGA, 2005). Um neurônio, segundo Haykin (2001) é uma unidade de processamento de informação imprescindível para o funcionamento de uma RNA. Analisando de forma mais detalhada o modelo neural, tem-se sua representação na Figura 2.

Figura 2 – Modelo não-linear de um neurônio



Fonte: Haykin (2001)

Onde $x_1, x_2, \ldots, x_m$ referem-se aos sinais de entrada; $w_{k1}, w_{k2}, \ldots, w_{km}$ são os pesos sinápticos do neurônio k; u_k é a saída do combinador linear devido aos sinais de entrada; b_k parâmetro adicional na Rede Neural (*bias*); $\varphi(\cdot)$ é a função de ativação; e y_k sinal de saída do neurônio (HAYKIN, 2001). Representando matematicamente, tem-se (1) e (2).

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

$$y_k = \varphi\,(u_k + b_k) \tag{2}$$

As principais funções de ativação usadas em redes neurais são: linear, tangente hiperbólica, sigmoidal, degrau, entre outras (HAYKIN, 2001).

2.2 *Support Vector Machine - SVM*

Support Vector Machine – SVM é uma técnica de *machine learning* a qual tem por objetivo a minimização de erros relacionados ao conjunto de treinamento, bem como de erros relacionados ao conjunto de teste, por assim, o conjunto de dados não utilizado no treinamento do algoritmo (ANDREOLA, 2009). O SVM também ganha destaque devido a pelo menos duas principais características: possui uma base teórica muito bem fundamentada na teoria de aprendizagem estatística, e pode atingir um alto nível de performance em suas aplicações utilizando otimização matemática (DOS SANTOS, 2002). Nesse sentido, esta descrição teórica ocupa-se em tratar sobre o *SVM* na perspectiva de classificação de dados, podendo ser divididos em dados linearmente separáveis e não linearmente separáveis, utilizando um conjunto de vetores de treinamento. Desse modo, é esperado que o *SVM* obtenha um hiperplano para a correta e satisfatória classificação dos dados analisados.

2.2.1 *SVM* – margens rígidas

Para a etapa de treinamento é necessário fornecer os dados de entrada e de saída $T = \{(x_1 y_1), (x_2 y_2), \ldots, (x_l y_l)\} \subseteq (X\,.\,Y\,)^l$, onde T é um conjunto de treinamento, $x_i \in \mathrm{X}$ (espaço de entrada), $y_i \in \mathrm{Y}$ (saída binária {−1, +1}), e $i = 1, \ldots, l$ (LORENA e CARVALHO, 2007; BELTRAMI, 2009). A equação de um hiperplano utilizada para fazer a classificação é dada

por (3), onde $w.\ x$ se refere ao produto escalar entre os vetores w e x, $w \in$ X diz respeito ao vetor normal ao hiperplano, e $\frac{b}{||w||}$ $(b \in \mathbb{R})$ é a distância do hiperplano em relação à origem (LORENA e CARVALHO, 2007).

$$f(x) = \langle w.x \rangle + b = 0 \tag{3}$$

A equação do hiperplano divide o espaço de entrada X em outras duas regiões*:* $w\ .x + b = -1\ (margem\ negativa)$ e $w\ .x + \ b = 1\ (margem\ positiva)$, dessa forma, considera-se que um vetor de treinamento está classificado corretamente quando corresponde ao conjunto de inequações (4) e (5).

$$\begin{cases} w.x_i + b \geq 1\ se\ y_i = 1 \\ w.x_i + b \leq -1\ se\ y_i = -1 \end{cases} \tag{4}$$

$$y_i(w.x_i + b) - 1 \geq 1, \qquad \forall (y_i, y_i) \in T \tag{5}$$

Sabendo que a distância entre as duas margens é dada por $\frac{2}{||w||}$, logo, para achar o hiperplano ótimo, basta maximizar a distância entre as margens. A partir dessas considerações, essa otimização poder ser obtida pela minimização de $\frac{1}{2}\left|\left|w\right|\right|^2$, resultando em (6) (BELTRAMI, 2009).

$$Minimizar\ \tfrac{1}{2}\left|\left|w\right|\right|^2 \tag{6}$$

$$Restrições:\ y_i(w.x_i + b) - 1 \geq 1,\ \forall_i = 1, \dots, l$$

O problema primal (6) pode ser consideravelmente complexo, dependendo da situação, devido às restrições de desigualdade. Como solução, faz-se a representação do problema no espaço dual e aplica-se a função Lagrangiana, que, de acordo com Lorena e Carvalho, (2007, p. 55) “engloba as restrições à função objetivo, associadas a parâmetros denominados multiplicadores de Lagrange” $(\alpha_i \geq 0)$ Equação (7).

$$L\ (w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i (y_i(w.x_i + b) - 1) \tag{7}$$

A fim de obter o ponto de sela da função, ela deve ser minimizada, o que, consequentemente, implica em maximizar as variáveis α_i e minimizar w e b, vide (8).

$$\frac{\partial L}{\partial b} = 0 \ \text{ e } \ \frac{\partial L}{\partial w} = 0 \tag{8}$$

Resolvendo as derivadas e substituindo-as na função Lagrangiana, obtém-se (9).

$$Maximizar \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i . x_j) \tag{9}$$

$$Restrições: \sum_{i=1}^{l} y_i \alpha_i = 0$$

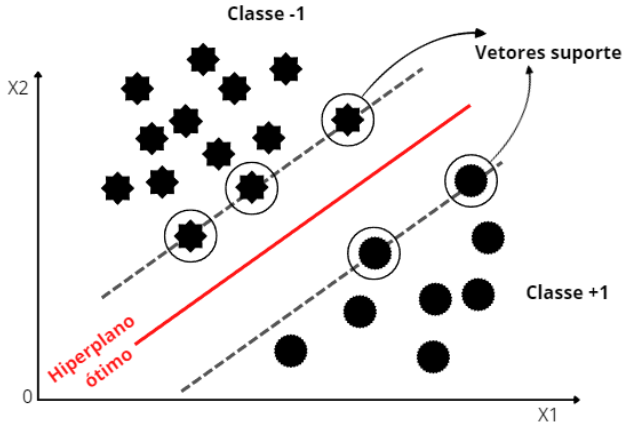
$$\alpha_i \geq 0, \qquad \forall_i = 1, \dots, l$$

Após calculado α^*(solução do problema dual), calcula-se o vetor de pesos ótimo w^* por meio de $w^* = \sum_{i=1}^{l} y_i \alpha_i^* x_i$. De mesmo modo, também é possível obter b^*, sendo ele definido por α^* e pelas condições de Kühn-Tucker. Dessa forma, tem-se (10).

$$\alpha_i^* (y_i(w^*.x_i + b^*) - 1) = 0, \qquad \forall_i = 1, \dots, l \tag{10}$$

Portanto, unicamente os dados de entrada com margem igual a 1 e que atendem a $y_i(w.x_i + b) = 1$ possui seu respectivo multiplicador de Lagrange $\alpha_i \neq 0$. Os demais multiplicadores são iguais a zero. Os multiplicadores diferentes de zero ($\alpha_i \neq 0$) empregados no cálculo de w^*são chamados de vetores suporte, como exemplificado pela Figura 3.

Figura 3 – Exemplificação de um hiperplano ótimo e vetores suporte



Fonte: Os autores (2022)

Por fim, pode-se expressar o hiperplano ótimo na sua representação dual por meio da Equação (11) (BELTRAMI, 2009).

$$f(x, \alpha^*, b^*) = \sum_{i=1}^{l} y\, \alpha_i^*(x_i, x) + b^* \tag{11}$$

2.2.2 *SVM* – margens flexíveis

Existem algumas situações em que não é possível separar os dados linearmente. Nesses casos, uma possível solução apresentada é permitir que o *SVM* "cometa alguns erros" em algumas amostras (ZHOU, 2021), em outras palavras, que ele seja mais flexível. Para isso, são usadas variáveis de folga $\xi_i \geq 0$, elas são mais flexíveis em relação as restrições impostas ao problema de otimização (ANDREOLA, 2009; LORENA e CARVALHO, 2007). Já se $\xi_i = 0$, o vetor está corretamente separado. Ao inserir as variáveis de folga, considera-se que o vetor de treinamento está satisfatoriamente classificado se (12) é atendida.

$$y_i(w.x_i + b) \geq 1 - \xi_i, \ \forall_i = 1, \dots, l \tag{12}$$

Sendo assim, o problema primal a ser resolvido é apresentado por (13).

$$Minimizar \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \tag{13}$$

$$Restrições: \ y_i(w.x_i + b) \geq 1 - \xi_i, \forall_i = 1, \dots, l$$

$$\xi_i > 0, \qquad \forall_i = 1, \dots, l$$

"A constante C é um termo de regularização que impõe um peso à minimização dos erros no conjunto de treinamento em relação à minimização da complexidade do modelo" (LORENA e CARVALHO, 2007, p. 58). Além disso, $w \in \mathbb{R}^n$ e $b \in \mathbb{R}$. O problema dual é apresentado por (14).

$$Maximizar \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i.x_j) \tag{14}$$

$$Restrições: \sum_{i=1}^{l} y_i \alpha_i = 0$$

$$0 \geq \alpha_i \geq C, \qquad \forall_i = 1, \dots, l$$

As Equações (15) e (16) relacionam as condições de Kühn-Tucker ao problema dual (14).

$$\alpha_i (y_i (w.x_i + b) - 1 + \xi_i) = 0, \qquad \forall_i = 1, \dots, l \tag{15}$$

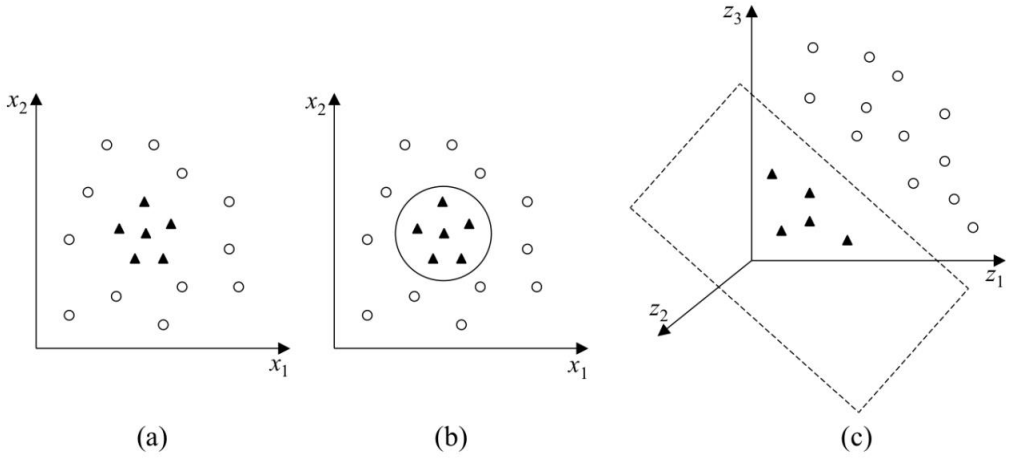
$$\xi_i (\alpha_i - C) = 0, \qquad \forall_i = 1, \dots, l \tag{16}$$

Como um *SVM* com margem rígida, seus pontos $\alpha_i > 0$, chamados de vetores de suporte (*SVs*), são os dados que participam da formação do hiperplano separador (LORENA e CARVALHO, 2007). De mesmo modo, se $0 > \alpha_i > C$ (vetores suportes não limitados), por assim $\xi_i (\alpha_i - C) = 0$, tem-se que $\xi_i = 0$, estes estão localizados sobre a margem de sua classe (BELTRAMI, 2009). Ainda, para $\alpha_i = C$, é possível que $\xi_i > 1$ (erro de classificação) ou $0 < \xi_i \leq 1$ (classificados corretamente, mas entre as margens). Por fim, à exceção dos vetores suporte, podendo ser eles limitados ou não, tem-se que os demais pontos com $\alpha_i = 0$ não interferem na construção do hiperplano ótimo; b^*, w^*e a superfície de decisão são determinadas como nos *SVMs* com margens rígidas.

2.2.3 *SVM* não linear

Os *SVMs* lineares funcionam muito bem na classificação de conjuntos de dados que são linearmente separáveis ou que possuam uma distribuição aproximadamente linear, como no caso das margens flexíveis. Entretanto, existem situações em que não é possível dividir de forma satisfatória os dados de treinamento por um hiperplano, como no caso apresentado pela Figura 4 (LORENA e CARVALHO, 2007).

Figura 4 - (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características



Fonte: Lorena e Carvalho (2007)

Para realizar esse tipo de classificação, muda-se a representação dos dados por meio de $x = (x_1, x_2, \dots, x_n) \mapsto \phi(x) = \big(\phi_1(x), \dots, \phi_n(x)\big)$, onde $F = \{\phi(x) \mid x \in \mathrm{X}\}$, bastando substituir na função objetivo, nesse caso (13), as entradas x por $\phi(x)$, sendo $\phi: \mathbb{R}^n \to \mathbb{R}^N$, com $N \gg n$. Entretanto, "calcular o produto interno $\phi(x_i) \,.\, \phi\big(x_j\big)$ diretamente no espaço de características pode se tornar computacionalmente inviável, devido à sua alta

dimensionalidade" (BELTRAMI, 2009, p. 61 *apud* LIMA, 2002). Por esse motivo, utiliza-se como alternativa mais econômica as funções *Kernel*, $K(x,z) = \phi(x).\phi(z)$, dependentes somente das variáveis de entrada. Sendo ϕ o mapeamento de X para o espaço característico. As funções *Kernel* mais usadas são a linear, polinomial, sigmoidal e base radial. Dessa forma, o problema (14) se torna em (17), onde α_i são os multiplicadores de Lagrange.

$$Maximizar \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i . x_j) \quad (17)$$

$$Restrições: \sum_{i=1}^{l} y_i \alpha_i = 0$$

$$0 \geq \alpha_i \geq C, \qquad \forall_i = 1, \dots, l$$

Dependendo dos valores assumidos por C e diferentes funções *Kernel,* implicarão em resultados diferentes.

3. Metodologia

Este trabalho utilizou a base de dados *Stroke Prediction,* obtida pela plataforma on-line Kaggle (2022), e, por meio do software *R Studio*, feita a manipulação dos dados e aplicação dos métodos de classificação - *MLP* e *SVM*.

Sobre o conjunto de dados original utilizado, este possui um total de 12 variáveis e 5110 observações. 5 variáveis foram descartadas pois tinham baixa correlação com a variável principal, resultando em 7, as quais foram utilizadas. A Tabela 1 descreve cada uma delas.

Tabela 1 – Variáveis usadas na classificação de pacientes que já tiveram AVC

Nome da Variável	Significado	Tipo
Id	Identificação única	Valor inteiro entre 1 e 5110
Age	Idade do paciente	Valor inteiro entre 30 dias e 82 anos
Hypertension	O paciente tem hipertensão?	Valor binário: 0 não tem e 1 tem
Heart disease	O paciente tem problemas cardíacos?	Valor binário: 0 não tem e 1 tem
Ever_married	O paciente já se casou?	Valor binário: 0 não se casou e 1 se casou
Avg_glucose_level	Nível médio de glicose	Valor mínimo 59 e máximo 271,74
Stroke	O paciente já teve um AVC?	Valor binário: 0 não tem e 1 tem

Fonte: Os autores (2022) – base de dados Kaggle.

Como primeira etapa o arquivo foi formatado dentro do *Excel* e posteriormente carregado dentro do *R Studio* utilizando o pacote *readxl()* para ler o arquivo, em seguida foi feita a matriz de correlação e eliminado as 5 colunas com variáveis que obtiveram as menores correlações com a variável *Stroke.* Dos 5110 dados apenas 249 eram paciente que já tiveram AVC e o restante, 4861, não tiveram AVC, para equilibrar os dados, utilizou-se todos os pacientes que já tiveram AVC, 249, e 249 que não tiveram, os quais foram embaralhados aleatoriamente antes de serem escolhidos. O treinamento dos algoritmos foi feito com 70% dos dados e o teste com os 30% restante. Essa separação foi feita para ambos os modelos de *ML* e dados (equilibrados e desequilibrados). Além disso, optou-se por normalizar todos os dados, utilizando a normalização padrão 0-1, a fim de diminuir a redundância e aumentar a integridade dos dados juntamente com o desempenho.

Da utilização dos dados desequilibrados, o número de pacientes que tiveram AVC permaneceu o mesmo, o total de 249, já os dados de pessoas que não tiveram AVC se alteraram para 498, 747, 996, 1992 e 4861, sendo o último a amostra total destes dados, para que fosse possível analisar a influência desse parâmetro na classificação final. Esses valores foram escolhidos por abrangerem um intervalo baixo, médio, alto e total da amostra.

Para a rede neural *MultiLayer Perceptron,* utilizou-se a função *mlp*() do pacote *RSNNS* presente no *R Studio*. Já para realizar a previsão dos dados do conjunto de teste após o treinamento do modelo, a função *predict(),* do mesmo pacote, foi usada. O modelo foi rodado com 100 núcleos na camada interna, função aprendizagem 0,1 e máximo de 500 iterações.

Já para o método *SVM* foi utilizada a função *svm()* do pacote *e1071*, do *R Studio*. Para realizar a previsão dos dados do conjunto de teste após o treinamento do modelo, foi utilizada a função *predict()* do mesmo pacote e usado a função sigmoide.

Por fim, os resultados dos testes foram analisados via matriz de confusão, a qual indica a quantidade de dados de teste que foram classificados correta e incorretamente. Após, utilizando os resultados provenientes das matrizes, calculou-se a acurácia de cada um dos modelos por meio da equação (18) e a precisão usando (19).

$$Acurácia = \frac{VP\ (Verdadeiro\ positivo) + VN\ (Verdadeiro\ negativo)}{Total}.100\% \quad (18)$$

Onde, $Total$ representa $VP\ (verdadeiro\ positivo) + VN\ (verdadeiro\ negativo) + FP(falso\ positivo) + FN(falso\ negativo)$.

$$Precisão = \frac{Verdadeiro\ positivo\ (VP)}{VP\ (verdadeiro\ positivo) + FP(falso\ positivo)}.100\% \quad (19)$$

4. Resultados e Discussões

Com a realização dos passos apresentados na metodologia para o *MLP*, obteve-se os seguintes resultados para os dados equilibrados e desequilibrados. Vide Tabela 2.

Tabela 2 – Matrizes confusão para os dados equilibrados e desequilibrados (MLP)

249/249	**previsto**		
resposta	Não Teve	Teve	**Precisão**
Não Teve	50	18	**74%**
Teve	21	61	**74%**
Acurácia	**74%**		

249/498	**previsto**		
resposta	Não Teve	Teve	**Precisão**
Não Teve	135	18	**88%**
Teve	29	43	**60%**
Acurácia	**79%**		

249/747	**previsto**		
resposta	Não Teve	Teve	**Precisão**
Não Teve	207	30	**87%**
Teve	33	29	**47%**
Acurácia	**79%**		

249/996	**previsto**		
resposta	Não Teve	Teve	**Precisão**
Não Teve	270	25	**92%**
Teve	55	24	**30%**
Acurácia	**79%**		

249/1992	**previsto**		
resposta	Não Teve	Teve	**Precisão**
Não Teve	587	21	**97%**
Teve	50	15	**23%**
Acurácia	**89%**		

249/4861	**previsto**		
resposta	Não Teve	Teve	**Precisão**
Não Teve	1469	2	**100%**
Teve	61	1	**2%**
Acurácia	**96%**		

Fonte: Os autores (2022)

Percebe-se que ao utilizar dados perfeitamente equilibrados (249/249), obteve-se uma acurácia de 74%. Isto é, o modelo acertou 50 casos que não tiveram AVC, e 61 casos que tiveram. Mas errou 21 casos dizendo que não tiveram, quando na realidade tiveram, e classificou 18 casos como que tiveram, mas na verdade não tiveram. A precisão, nesse caso, foi de 74% para “não teve” e “teve”.

Já com os dados desequilibrados, como em (249/4861), o algoritmo classificou corretamente 1469 casos como que não tiveram e 1 caso como que teve. Em contrapartida, classificou erroneamente 61, dizendo que não tiveram e 2 afirmando que tiveram. A acurácia nesse caso foi de 96%, a precisão para “não teve” foi de 100% (utilizando arredondamento) e para “teve” foi de 2%.

Para o *SVM*, os resultados estão contidos na Tabela 3:

Tabela 3 - Matrizes confusão para os dados equilibrados e desequilibrados (SVM)

249/249	previsto		
resposta	Não Teve	Teve	**Precisão**
Não Teve	56	19	**75%**
Teve	17	57	**77%**
Acurácia	**76%**		

249/498	previsto		
resposta	Não Teve	Teve	**Precisão**
Não Teve	126	24	**84%**
Teve	36	38	**51%**
Acurácia	**73%**		

249/747	previsto		
resposta	Não Teve	Teve	**Precisão**
Não Teve	203	27	**88%**
Teve	44	25	**36%**
Acurácia	**76%**		

249/996	previsto		
resposta	Não Teve	Teve	**Precisão**
Não Teve	262	35	**88%**
Teve	53	23	**30%**
Acurácia	**76%**		

249/1992	previsto		
resposta	Não Teve	Teve	**Precisão**
Não Teve	567	29	**95%**
Teve	69	7	**9%**
Acurácia	**85%**		

249/4861	previsto		
Resposta	Não Teve	Teve	**Precisão**
Não Teve	1428	27	**98%**
Teve	72	6	**7%**
Acurácia	**93%**		

Fonte: Os autores (2022)

O modelo treinado pelo SVM para os dados perfeitamente equilibrados (249/249) acertou aproximadamente 76% dos casos. Ele errou 19 casos classificando-os em como tiveram AVC, sendo que não tiveram, e 17 como se não tiveram, mas que na verdade tiveram. A precisão para “não teve” foi de 75% e de 77% para “teve”.

Analisando o extremo oposto, dados desequilibrados (249/4861), o modelo errou somente 27 casos quando os classificou como já tiveram, e 72 casos os quais foram classificados como que não tiveram. Entretanto, ele acertou 1428 casos dizendo que não tiveram, e 6 casos que tiveram.

5. Conclusões

Com esse estudo pode-se observar que de fato o equilíbrio dos dados afeta diretamente a qualidade dos resultados, embora os resultados para casos que não tiveram AVC pareçam ser mais acurados, eles não são dentro do cenário de dados desequilibrados. Isso pode ser explicado pelas considerações e o treinamento que os modelos realizam. Recapitulando a ideia apresentada no início desse trabalho de que um algoritmo de *ML* se torna mais preciso quando treinado com mais exemplos, ao dar mais amostras de um tipo de caso do que de outro, o algoritmo se torna muito mais preciso para fazer a classificação daqueles casos que ele mais teve contato. Por consequência, ele acaba ficando mais despercebido para os casos com os quais teve menos contato. Em outras palavras, por mais que haja alguns

casos que não tiveram AVC e o algoritmo tenha acertado alguns deles, isso o deixa desbalanceado para fazer a correta e confiável classificação, pois acabará considerando alguns casos como não tendo tido AVC. Isso se torna um grande problema para uma aplicação na medicina, já que para novos dados, pacientes que tem reais chances de sofrer um AVC seriam classificados como pessoas saudáveis e nenhuma ação seria tomada para cuidado de sua saúde.

Igualmente, mesmo que o objetivo fosse classificar as pessoas que não tiveram AVC, ele ainda assim não seria confiável. É preciso, e recomendável, utilizar dados equilibrados para que ele seja treinado de forma mais acurada e homogênea possível, considerando ambos os casos. Dessa forma, considera-se que o objetivo inicial desse trabalho de analisar a influência de uma base de dados desequilibrada na classificação de pacientes que já tiveram AVC por meio de duas técnicas de ML, RNA *MultiLayer Perceptron* e *Support Vector Machine*, foi alcançado com sucesso. Adicionalmente ao estudo, percebe-se que técnicas de *ML* são ferramentas fortíssimas para o auxílio na tomada de decisão, entretanto, não devem ser consideradas individualmente. Principalmente em áreas da saúde, um diagnóstico médico deve ser levado em consideração para suporte a decisão.

Para estudos futuros, recomenda-se ampliar o escopo da pesquisa utilizando técnicas para equilíbrio de amostras naturalmente desequilibradas, tais como a criação de dados sintéticos. Como também a variação de parâmetros dos algoritmos de *MLP* e *SVM* a fim de obter uma melhor acurácia e precisão.

Referências

ANDREOLA, R. **Support Vector Machines na classificação de imagens hiperespectrais**. 2009. Porto Alegre. Dissertação (Mestrado) – Universidade Federal do Paraná.

BATTINENI, G.; CHINTALAPUDI, N.; AMENTA, F. Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). **Informatics in Medicine Unlocked**, v. 16, p. 100200, 2019.

BELTRAMI, Mônica. **Precificação de opções sobre ações por modelos de Support Vector Regression**. Curitiba: UFPR, 2009. Dissertação (Mestrado) – Pós-graduação em Métodos Numéricos em Engenharia, Universidade Federal do Paraná, 2009.

GARDNER, M. W.; DORLING, S. R. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric environment**, v. 32, n. 14-15, p. 2627-2636, 1998.

DEO, Rahul C. Machine learning in medicine. **Circulation**, v. 132, n. 20, p. 1920-1930, 2015.

HAYKIN, Simon. **Redes Neurais: Princípios e prática**. Porto Alegre: Bookman, 2 ed., 2001.

LORENA, A. C.; DE CARVALHO, A. C. P. L. F. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.

NORIEGA, Leonardo. Multilayer perceptron tutorial. **School of Computing. Staffordshire University**, 2005.

RAJKOMAR, Alvin; DEAN, Jeffrey; KOHANE, Isaac. Machine learning in medicine. **New England Journal of Medicine**, v. 380, n. 14, p. 1347-1358, 2019.

SCALCO, F. F. **Visualização de dados em processos de Machine Learning**. 2021. Caxias do Sul. Universidade de Caxias do Sul.

ZHOU, Zhi-Hua. **Machine learning**. Springer Nature, 2021.