Classificação da síndrome do doente Eutireoideo com algoritmos de *machine learning*: Uma aplicação de suporte ao diagnóstico

Vinicius A. de Almeida [1], Caio M. V. Cavalcante [2], Náthalee C. de A. Lima[3], Rosana C. B. Rego[4]

[1] Graduando em Tecnologia da Informação, Universidade Federal Rural do Semi-Árido, Pau dos Ferros, Brasil
(vinicius.almeida37366@alunos.ufersa.edu.br)
[2] Graduando em Tecnologia da Informação, Universidade Federal Rural do Semi-Árido, Pau dos Ferros, Brasil
(caio.cavalcante@alunos.ufersa.edu.br)
[3] Departamento de Engenharia e Tecnologia, Universidade Federal Rural do Semi-Árido, Pau dos Ferros, Brasil
(nathalee.lima@ufersa.edu.br)
[4] Departamento de Engenharia e Tecnologia, Universidade Federal Rural do Semi-Árido, Pau dos Ferros, Brasil
(rosana.rego@ufersa.edu.br)

Resumo: A síndrome do doente eutireoideo é uma condição médica que afeta a glândula tireoide. O diagnóstico precoce e preciso é crucial para o tratamento. No entanto, a interpretação dos resultados dos exames pode ser subjetiva. Este trabalho visa utilizar algoritmos de aprendizado de máquina para classificar o problema com base em atributos obtidos via exame de sangue, tais como T4, TSH, e T3. Os resultados do trabalho indicam que a utilização dos algoritmos pode ser eficaz na classificação da síndrome.

Palavras-chave: *Machine learning*; Ciência de dados; Classificação; Saúde

INTRODUÇÃO

A síndrome do doente Eutireoideo (ou *Euthyroid Sick Syndrome* - ESS) é uma condição em que ocorre um distúrbio na regulação hormonal das glândulas da tireóide em pacientes que apresentam alguma outra doença ou infecção (Goemann e Wajner, 2009). Embora a tireóide esteja funcionando normalmente, a sua produção de hormônios tireoidianos é afetada pela inflamação, infecção ou outra condição (Coronado et al., 2017). Isso pode resultar em níveis anormais de hormônios tireoidianos no sangue e sintomas semelhantes aos da tireoidite ou hipotireoidismo. Neste contexto, o tratamento da síndrome do paciente com eutireoideo geralmente inclui o tratamento da doença subjacente, e a avaliação da função tireoidiana pode ser necessária para monitorar a evolução da condição (Novik, 1996; Lee, 2011).

O diagnóstico precoce e preciso é crucial para o tratamento eficaz da doença (Tyagi, 2018). No entanto, a interpretação dos resultados dos exames de sangue pode ser difícil e subjetiva, levando a erros no diagnóstico (Sonuç, 2021). Neste contexto, os algoritmos de *machine learning* (ML) podem ser capazes de auxiliar os médicos na interpretação dos resultados (Ha, 2021).

Os modelos de ML são capazes de aprender e se ajustar às características dos dados, o que pode aumentar a precisão das previsões (Duggal, 2020). Ao utilizar técnicas de ML para classificar a síndrome do doente Eutireoideo, é possível identificar precocemente pacientes que possam estar desenvolvendo a doença, o que pode levar a um tratamento mais efetivo (Islam, 2022). Além disso, a utilização de algoritmos para classificar a ESS pode economizar tempo e recursos em comparação com métodos convencionais de análise de dados (Islam, 2022).

A aplicação de algoritmos de ML é uma abordagem promissora para a detecção precoce da ESS, pois permite uma análise rápida e precisa dos dados, bem como a identificação de padrões e tendências nos dados clínicos que possam ser utilizados para melhorar o diagnóstico e o tratamento da doença (Islam, 2022).

Neste contexto, este trabalho visa utilizar algoritmos de ML para classificar a síndrome do doente Eutireoideo com base em atributos obtidos via exame

de sangue, tais como T4 (hormônio Tiroxina), TSH (hormônio Tireoestimulante), T4 livre e T3 (hormônio Triiodotironina). O objetivo é fornecer um suporte ao diagnóstico que aumente a precisão e eficiência do processo de diagnóstico.

Os modelos utilizados neste trabalho foram: *decision tree* (DT), *random forest* (RF), regressão logística (RL) e *naive bayes* (NB). Estes são alguns dos algoritmos mais populares e amplamente utilizados em diversas aplicações de *machine learning*. A árvore de decisão é um algoritmo baseado em regras que permite a classificação e previsão de dados utilizando uma estrutura de árvore (Hu, 2019). Já o *random forest* é uma extensão da árvore de decisão, que constrói várias árvores de decisão e combina suas previsões para melhorar a precisão do modelo (Fawagreh, 2014). A regressão logística é um algoritmo usado para resolver problemas de classificação binária (Rodrigues, 2013), enquanto o *naive bayes* é um algoritmo baseado em probabilidades que é usado principalmente para classificação de texto (Poniszewska-Marańda, 2023).

Os algoritmos de ML, utilizados neste trabalho, foram treinados com uma grande quantidade de dados de pacientes previamente diagnosticados. A partir desses dados, os algoritmos aprenderam a identificar padrões e relações entre os atributos dos pacientes e suas condições médicas.

Dessa forma, o trabalho está dividido como segue: uma seção para o Conjunto de dados, onde as características dos dados utilizados serão descritas. Seção de modelos de *machine learning*, em que os fundamentos e características dos modelos utilizados serão apresentados. Seção de resultados e discussões, em que os resultados obtidos serão apresentados. Por fim, a seção das conclusões, onde as considerações finais serão apresentadas.

CONJUNTO DE DADOS

Este estudo utiliza o conjunto de dados referente à síndrome do doente Eutireoideo, fornecido pela *University of California Irvine,* no repositório público *machine learning.* O conjunto de dados contém 25 atributos para cada paciente, incluindo informações sobre idade, sexo, medicação, gravidez, cirurgia, resultados de testes de função da tireoide.

Neste trabalho, foram utilizados os dados: idade, sexo, doente, TSH, T3, TT4, T4U e FTI. Esses parâmetros foram selecionados porque fornecem informações importantes sobre a função tireoidiana e o estado de saúde do paciente. A idade e o sexo podem afetar o risco de desenvolver distúrbios da tireoide, enquanto os hormônio estimulante da tireoide (TSH), triiodotironina (T3), tiroxina (TT4), tiroxina total (T4U) e índice de tiroxina livre (FTI) são indicadores importantes da função da tireoide e podem ajudar a diagnosticar distúrbios da tireoide. Além do mais, esses dados podem ser obtidos através de exame de sangue (Bouknight, 2003).

Os dados disponíveis não estavam balanceados, dessa forma, foi necessário realizar o processo de balanceamento e limpeza do conjunto de dados.

No processo de balanceamento foi utilizado a técnica *Synthetic Minority Over-sampling Technique* (SMOTE). Essa técnica gera dados extras da classe minoritária, com a finalidade de superar o desbalanceamento de dados. A geração dos novos dados tem como base a técnica *K Nearest Neighbours* - KNN, ela leva em consideração os pontos mais próximos em um plano cartesiano para gerar os dados faltantes (Chawla, 2002).

Para realizar a limpeza dos dados, foi adotado o tratamento dos dados faltantes que consistiu em preencher os valores ausentes com a média dos dados presentes, quando se tratava de uma variável numérica, e com a moda (valor mais frequente), quando se tratava de uma variável categórica, como é o caso do gênero. Esse tipo de abordagem é comum e pode ser útil para evitar perda de dados em análises subsequentes, mas é importante lembrar que a escolha da estratégia de imputação pode ter impacto nos resultados finais da análise.

MODELOS DE *MACHINE LEARNING*

Nesta seção, serão apresentados diferentes modelos de aprendizado de máquina, como árvore de decisão, *random forest*, regressão logística e *naive bayes*, a fim de realizar a classificação.

A. *DECISION TREES*

O modelo *decision tree*, ou em portugês Árvore de decisão, é um algoritmo de *machine learning* que utiliza a estrutura de árvores para fazer predições. Ela é composta por ramos, nós, sub-árvores e folhas, sendo que o nível raiz representa o conjunto inteiro de dados. A cada ramo é feita uma decisão, a quantidade de perguntas realizadas é conhecida como profundidade da árvore. Ao chegar à folha, temos a classificação final (Quinlan, 1990).

Durante o treinamento do modelo, que é supervisionado, é utilizado um conjunto de dados de entrada e saída, onde as saídas representam as classes. A árvore é construída a partir de uma série de perguntas e decisões, cada uma delas é representada por um nó ou ramo, e o número total de perguntas é conhecido como a profundidade da árvore (Silva, 2005).

Estes modelos utilizam a estratégia de dividir para conquistar: um problema complexo é decomposto em sub-problemas mais simples e recursivamente esta

técnica é aplicada a cada sub-problema (Gama, 2004).

As árvores de decisão estão entre os mais populares algoritmos de inferência e tem sido aplicado em várias áreas como, por exemplo, diagnóstico médico e risco de crédito (Mitchell, 1997), e deles pode-se extrair regras do tipo "se-então" que são facilmente compreendidas. A capacidade de discriminação de uma árvore vem da divisão do espaço definido pelos atributos em sub-espaços e a cada sub-espaço é associada uma classe.

Na implementação das árvores de decisão, foi utilizada a função entropia para avaliar as divisões dos dados em cada árvore. Além disso, foi selecionada a profundidade máxima de uma árvore como 6. Ao final, foi atribuído pesos inversamente proporcionais às frequências das classes nos dados de entrada.

B. *RANDOM FOREST*

O modelo *random forest*, ou no português Floresta aleatória, é um algoritmo de *machine learning* que utiliza a aprendizagem supervisionada para classificação ou regressão. O algoritmo utiliza da simplicidade proporcionada pelas árvores de decisão com a aleatoriedade. Ele pode construir várias árvores de decisão, criando assim uma floresta, e em seguida, combinar as saídas das árvores para melhorar a precisão do modelo. Entretanto, se o crescimento da floresta não for controlado, o modelo pode apresentar *overfitting* ou *underfitting* (Cutler, 2012).

De forma resumida, o algoritmo funciona utilizando *N* árvores combinadas para predizer o melhor resultado. No processo de classificação, o algoritmo conta as classificações feitas por cada árvore e seleciona a classificação mais frequente. Este algoritmo tem uma boa capacidade de generalização (Fawagreh, 2014).

Neste trabalho, foi implementado o algoritmo do *random forest* para classificação. Para implementação do modelo proposto no trabalho, foram utilizadas 10 árvores na floresta, com uma profundidade máxima de 10. Além disso, foi utilizado a função logarítmica para avaliar a qualidade das divisões dos dados entre as árvores e avaliação do modelo. Ainda, foi estabelecido o número mínimo de amostras necessárias para está no nó folha como 5, e o número mínimo de amostras para dividir um nó interno foi estabelecido como 2. Por fim, assim como no modelo de árvores de decisão, foi atribuído pesos inversamente proporcionais às frequências das classes nos dados de entrada.

C. NAIVE BAYES

Naive Bayes é um algoritmo de classificação na área de *machine learning* baseado em probabilidade que utiliza o teorema de Bayes para prever a categoria de uma nova entrada (Webb, 2010). Ele supõe que as *features* são independentes entre si e que cada *feature* tem uma distribuição de probabilidade conhecida. Ele se baseia no Teorema de Bayes, originalmente desenvolvido pelo matemático inglês Thomas Bayes (Webb, 2010; Berrar, 2018).

O algoritmo primeiro calcula a probabilidade de cada categoria dado às *features* da nova entrada, e em seguida escolhe a categoria com a maior probabilidade (Yang, 2018). Isso é feito usando a equação de Bayes para calcular a probabilidade posterior de cada categoria, dado as *features* da nova entrada. Dessa forma, o algoritmo se utiliza dos dados para classificar qual a probabilidade de um possível evento 'A' ocorrer, dado que um evento 'B' ocorreu (Jiang, 2007). Ou seja, o teorema refere-se a probabilidade condicional. Também utilizado no processo de linguagens naturais (Poniszewska-Marańda, 2023) e diagnósticos médicos (Jahangiri, 2023).

Para implementação do classificador, foi utilizado a biblioteca *scikit-learn*. Além disso, para realizar o ajuste dos parâmetros no algoritmo, o método do *grid search* foi utilizado.

Naive Bayes é um algoritmo rápido e fácil de implementar, e tem boa performance em muitos casos, especialmente quando a quantidade de dados é pequena (Metsis, 2006). No entanto, a suposição de independência entre as *features* nem sempre é verdadeira, o que pode levar a resultados menos precisos.

D. REGRESSÃO LOGÍSTICA

Regressão logística é um algoritmo de classificação que emprega a técnica de análise multivariada aplicada para verificar a probabilidade de ocorrer um dado evento a partir da identificação de características específicas dentro de cada categoria determinada pela divisão da área pertencente. Essa técnica possui testes estáticos diretos, incorporamento de efeitos não lineares e outros diagnósticos (Gouvêa et al., 2012).

O algoritmo utiliza em sua modelagem o método padrão de resposta binária, possuindo apenas duas possíveis respostas: sucesso (ou evento), codificado como 1, e fracasso (ou não-evento), codificado como 0. A razão disso é a variável aleatória com distribuição de Bernoulli que sugere tais respostas (Rodrigues, 2013).

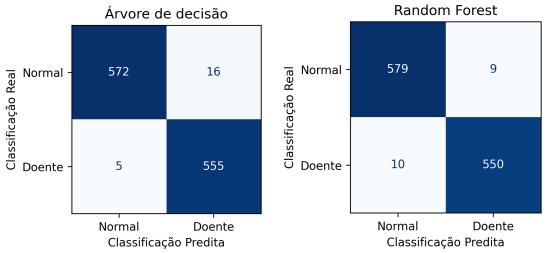
Para implementação do algoritmo, utilizou-se a biblioteca *scikit-learn,* sendo recorrido o esquema de treinamento *one-vs-rest* (OvR). Foi atribuído pesos inversamente proporcionais às frequências das classes nos dados de entrada. Além disso, também foi utilizado o modo de estado aleatório, utilizado para controlar o gerador de números aleatórios usado.

RESULTADOS E DISCUSSÃO

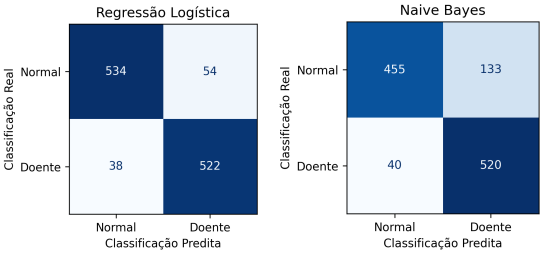
Os resultados de classificação dos modelos de árvore de decisão, *random forest*, regressão logística e *naive bayes* foram avaliados com base no número de pacientes corretamente classificados. Para analisar a performance dos modelos foi utilizado métricas como: matriz de confusão, curva de erro de classificação, curva ROC, curva de aprendizado, acurácia, recall, precisão e F1-score.

As Figuras 1 (a), (b), (c) e (d), mostram as matrizes de confusão para cada modelo. Observando a Figura 1 (a), percebe-se que a árvore de decisão corretamente classificou 572 pacientes como normais e 555 pacientes como doentes, com erros de classificação de 16 pacientes normais como doentes e 5 pacientes doentes como normais. Já o *random forest*, na Figura 1 (b), classificou 579 pacientes como normais e 550 pacientes como doentes, com erros de classificação de 9 pacientes normais como doentes e 10 pacientes doentes como normais.

A regressão logística, classificou 534 pacientes como normais e 522 pacientes como doentes, com erros de classificação de 54 pacientes normais como doentes e 38 pacientes doentes como normais. Por outro lado, o modelo de *bayes* classificou 455 pacientes como normais e 520 pacientes como doentes, com erros de classificação de 133 pacientes normais como doentes e 40 pacientes doentes como normais. É possível notar que cada modelo teve resultados diferentes.



(a) (b)



(c) (d)

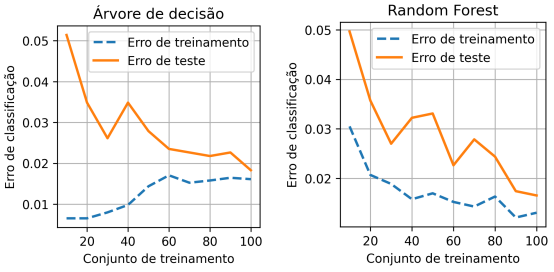
Figura 1. Matriz de confusão: (a) modelo árvore de decisão, (b) modelo *random forest*, (c) modelo regressão logística, (d) modelo *naive bayes*.

As Figuras 2 (a), (b), (c) e (d), mostram as curvas de erro de classificação dos modelos. A análise das curvas ajuda a identificar se o modelo está sofrendo de *overfitting* ou *underfitting*.

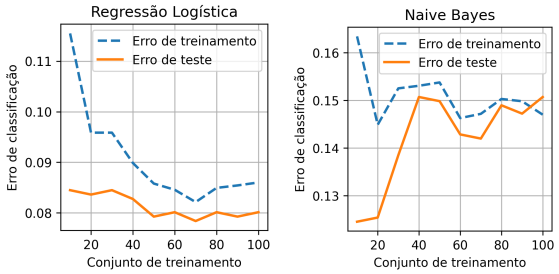
O modelo árvore de decisão, na Figura 2 (a), em sua curva de erro, mostrou-se eficiente no começo, porém com o aumento do conjunto de dados, foi errando cada vez mais em sua predição, isto é, o modelo apresentou uma tendência de aumento do erro com o aumento do número de dados, portanto, o modelo está sofrendo de *overfitting*.

De acordo com a Figura 2 (b), o modelo *random forest* teve um bom ajuste, inicialmente com um erro alto durante o treinamento e teste, que diminuiu gradualmente ao adicionar mais dados e foi nivelando gradualmente. Igualmente, na Figura 2 (c), o modelo de regressão logística se beneficia da adição de mais dados, tanto durante o treinamento quanto durante o teste.

Na Figura 2 (d), é possível observar que a curva de teste, no modelo do Naive, apresenta uma tendência de aumento de erro com o aumento do número de dados, isso significa que o modelo está sofrendo de *underfitting*.



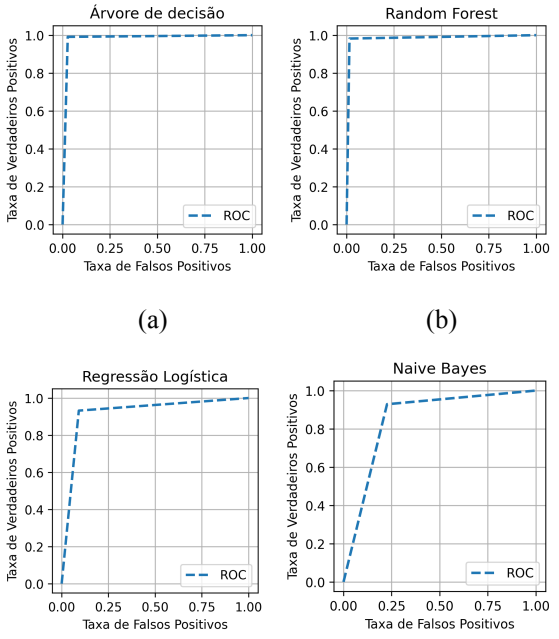
(a) (b)



(c) (d)

Figura 2. Curvas de erro de classificação: (a) modelo árvore de decisão, (b) modelo *random forest*, (c) modelo regressão logística, (d) modelo *naive bayes*.

A curva ROC é gerada a partir dos valores do ponto de corte da curva que é usado para classificar os dados em positivos ou negativos. A curva é plotada comparando a taxa de verdadeiros positivos (TPR) versus a taxa de falsos positivos (FPR). Quanto maior a área abaixo da curva ROC, maior a capacidade do modelo de separar positivos de negativos. Com base nas Figuras 3, (a), (b), (c) e (d), o modelo *random forest,* parece ser o modelo mais bem avaliado, seguido pelo modelo *decision tree*. É importante lembrar que a curva ROC é apenas uma das métricas que podem ser usadas para avaliar o desempenho dos modelos. É importante levar em conta o equilíbrio entre a precisão e o *recall* que serão mostrados na Tabela 1.



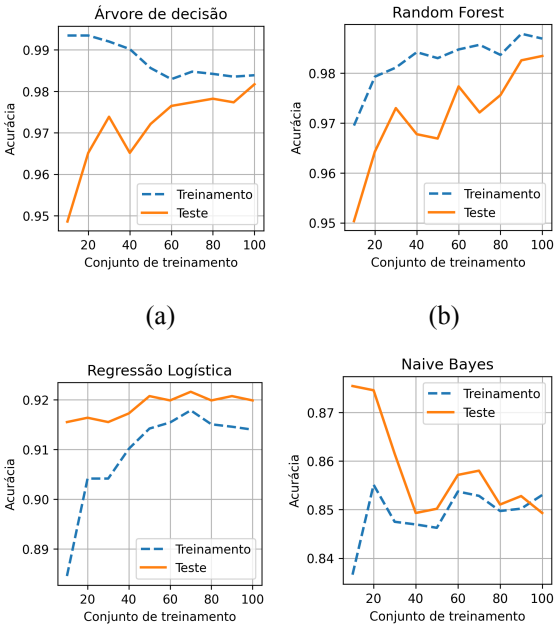
(a) (b)

(c) (d)

Figura 3. Curva ROC: (a) modelo árvore de decisão, (b) modelo *random forest*, (c) modelo regressão logística, (d) modelo *naive bayes*.

As Figuras 4 (a), (b), (c) e (d), mostram as curvas de aprendizado com relação à evolução da acurácia de cada modelo. A curva de aprendizado ajuda a encontrar o ponto ótimo de complexidade do modelo, que é o ponto onde a acurácia no conjunto de validação é a mais elevada. Analisando a Figura 4 (a), é possível observar que durante o processo de treinamento a acurácia do modelo árvore de decisão começa alta e com a adição de dados, diminui. Dessa forma, o modelo está inadequado para o problema. Situação parecida acontece com o modelo Naive bayes, Figura 4 (d), mas, neste caso, tanto no treinamento quanto no teste a acurácia diminui.

Na Figura 4 (b), observa-se que a acurácia aumenta com a adição de novos dados, com uma diferença pequena entre o treinamento e teste, o que indica que o modelo não está sofrendo de *overfitting*. De forma semelhante, na Figura 4 (c), o modelo de regressão logística, apresenta um aumento da acurácia com a adição de mais dados, no entanto, ao final a acurácia do modelo diminui um pouco. Portanto, dentre os 4 modelos, o *random forest* foi o que apresentou melhor comportamento de acordo com a curva de aprendizado.



(a) (b)

(c) (d)

Figura 4. Curvas de aprendizado: (a) modelo árvore de decisão, (b) modelo *random forest*, (c) modelo regressão logística, (d) modelo *naive bayes*.

De acordo com a Tabela 1, o modelo *random forest* apresenta a melhor performance, com acurácia de 0.9834, recall de 0.9821 e precisão de 0.9838, além de um F1-score de 0.9830. Isso indica que ele classificou corretamente a maior parte das amostras e com baixo número de falsos positivos e falsos

negativos. A acurácia é a porcentagem de amostras corretamente classificadas. O *recall* é a porcentagem de amostras relevantes recuperadas corretamente. A precisão é a porcentagem de amostras positivas corretamente identificadas. O F1-*score* é a média harmônica entre precisão e *recall*.

Já o modelo *decision tree*, apresenta uma performance boa, mas um pouco abaixo do *Random Forest,* com acurácia de 0.9817, recall de 0.9910 e precisão de 0.9719. O modelo regressão logística tem acurácia de 0.9198, *recall* de 0.9321 e precisão de 0.90625, indicando uma performance abaixo dos outros modelos. O F1-*score* de 0.9190 indica que ele não está conseguindo equilibrar corretamente a precisão e o *recall*.

Por fim, o modelo *naive bayes* apresenta a pior performance, com acurácia de 0.8493, *recall* de 0.9285 e precisão de 0.7963. O F1-*score* de 0.8573 indica que ele não está conseguindo equilibrar corretamente a precisão e o *recall*. Em geral, a escolha do melhor modelo dependerá das necessidades do projeto e dos critérios de avaliação, mas neste caso, o modelo *random forest* parece ser a escolha mais indicada.

Tabela 1. Métricas para avaliação dos modelos.

Modelo	Acurácia	*Recall*	Precisão	F1 *Score*
RF	0.9834	0.9821	0.9838	0.9830
DT	0.9817	0.9910	0.9719	0.9814
RL	0.9198	0.9321	0.90625	0.9190
NB	0.8493	0.9285	0.7963	0.8573

CONCLUSÃO

Os resultados demonstraram que a utilização dos algoritmos de aprendizado de máquina para classificação da síndrome do doente Eutireoideo é uma ferramenta valiosa para o auxílio no diagnóstico eficiente da doença. O trabalho demonstra que a combinação de dados médicos e a tecnologia do aprendizado de máquina é uma abordagem promissora para melhorar os processos de diagnóstico e possivelmente aumentar a qualidade de vida dos pacientes. Para trabalhos futuros, será desenvolvido o sistema de apoio ao diagnóstico.

AGRADECIMENTOS

À PICI/UFERSA pelo apoio financeiro na concessão de bolsa de Iniciação Científica. Ao grupo de pesquisa CiLab/UFERSA (Computational Intelligence Laboratory): https://github.com/cilab-ufersa ©.

REFERÊNCIAS

BERRAR, Daniel. Bayes' theorem and naive Bayes classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, v. 403, p. 412, 2018.

BOUKNIGHT, Anna Lee. Thyroid physiology and thyroid function testing. Otolaryngologic Clinics of North America, v. 36, n. 1, p. 9-15, 2003.

CORONADO ROBLES, Celia Margarita et al. Insuficiência de múltiplos órgãos e resultados clínicos em pacientes sépticos com síndrome do doente eutireoidiano. Medicina crítica (Colegio Mexicano de Medicina Crítica), v. 31, n. 3, p. 116-121, 2017.

CUTLER, Adele; CUTLER, D. Richard; STEVENS, John R. Random forests. Ensemble machine learning: Methods and applications, p. 157-175, 2012.

CHAWLA, Nitesh V. et al. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, v. 16, p. 321-357, 2002.

DUGGAL, Priyanka; SHUKLA, Shipra. Prediction of thyroid disorders using advanced machine learning techniques. In: 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2020. p. 670-675.

FAWAGREH, Khaled; GABER, Mohamed Medhat; ELYAN, Eyad. Random forests: from early developments to recent advancements. Systems Science & Control Engineering: An Open Access Journal, v. 2, n. 1, p. 602-609, 2014.

GAMA, João. Árvores de decisão. Palestra ministrada no Núcleo da Ciência de Computação da Universidade do Porto, Porto, 2002.

GOEMANN, Iuri Martin; WAJNER, Simone Magagnin. Efeito das citoquinas inflamatórias e do estresse oxidativo sobre a desiodase tipo 2 na Síndrome do Doente Eutireoideo. Salão de Iniciação Científica (21.: 2009 out. 19-23: Porto Alegre, RS). Livro de resumos. Porto Alegre: UFRGS, 2009., 2009.

GOUVÊA, Maria Aparecida; GONÇALVES, Eric Bacconi; MANTOVANI, Daielly Melina Nassif. Aplicação de Regressão Logística e Algoritmos Genéticos na Análise de Risco de Crédito. Revista Universo Contábil, Blumenau, v. 8, n. 2, p. 84-102,

30 abr. 2012. Revista Universo Contabil. http://dx.doi.org/10.4270/ruc.2012214.

HA, Eun Ju; BAEK, Jung Hwan. Applications of machine learning and deep learning to thyroid imaging: where do we stand?. Ultrasonography, v. 40, n. 1, p. 23, 2021.

HU, Xiyang; RUDIN, Cynthia; SELTZER, Margo. Optimal sparse decision trees. Advances in Neural Information Processing Systems, v. 32, 2019.

ISLAM, Saima Sharleen et al. Application of machine learning algorithms to predict the thyroid disease risk: an experimental comparative study. PeerJ Computer Science, v. 8, p. e898, 2022.

JAHANGIRI, Sonia; NIAKI, Seyed Taghi Akhavan. An Improved Naïve Bayes Approach to Diagnose Cardiovascular Disease: A Case Study. 2023.

JIANG, Liangxiao et al. Survey of improving naive bayes for classification. In: Advanced Data Mining and Applications: Third International Conference, ADMA 2007 Harbin, China, August 6-8, 2007. Proceedings 3. Springer Berlin Heidelberg, 2007. p. 134-145.

LEE, Sun; FARWELL, Alan P. Euthyroid sick syndrome. Comprehensive Physiology, v. 6, n. 2, p. 1071-1080, 2011.

METSIS, Vangelis; ANDROUTSOPOULOS, Ion; PALIOURAS, Georgios. Spam filtering with naive bayes-which naive bayes?. In: CEAS. 2006. p. 28-69.

MITCHELL, Tom; MCGRAW-HILL, Machine Learning. Edition. 1997.

NOVIK ASSAEL, Victoria. Síndrome de Eutiroidismo enfermo. Bol. Hosp. Viña del Mar, p. 101-9, 1996.

PONISZEWSKA-MARAŃDA, Aneta; VYNOGRADNYK, Elina; MARAŃDA, Witold. Medical Data Transformations in Healthcare Systems with the Use of Natural Language Processing Algorithms. Applied Sciences, v. 13, n. 2, p. 682, 2023.

QUINLAN, J. Ross. Decision trees and decision-making. IEEE Transactions on Systems, Man, and Cybernetics, v. 20, n. 2, p. 339-346, 1990.

RODRIGUES, Agatha Sacramento. Regressão logística com erro de medida: comparação de métodos de estimação. 2013. 115 f. Dissertação (Mestrado) - Curso de Ciências, Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2013.

SILVA, L. M. Uma aplicação de árvores de decisão, redes neurais e KNN para a identificação de modelos ARMA não-sazonais e sazonais. Rio de Janeiro. 145p. Tese de Doutorado-Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, 2005.

SONUÇ, Emrullah et al. Thyroid disease classification using machine learning algorithms. In: Journal of Physics: Conference Series. IOP Publishing, 2021. p. 012140.

TYAGI, Ankita; MEHRA, Ritika; SAXENA, Aditya. Interactive thyroid disease prediction system using machine learning technique. In: 2018 Fifth international conference on parallel, distributed and grid computing (PDGC). IEEE, 2018. p. 689-693.

WEBB, Geoffrey I.; KEOGH, Eamonn; MIIKKULAINEN, Risto. Naïve Bayes. Encyclopedia of machine learning, v. 15, p. 713-714, 2010.

YANG, Feng-Jen. An implementation of naive bayes classifier. In: 2018 International conference on computational science and computational intelligence (CSCI). IEEE, 2018. p. 301-306.