Modelos preditivos de doenças cardíacas usando técnicas de classificação de aprendizado de máquina

Ítalo L. F. Portinho[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Av. Gal. Milton Tavares de Souza, s/nº – 24.210-346 – Niterói – RJ – Brazil

italons@gmail.com, italoleite@id.uff.br

Abstract. *Cardiovascular diseases are the leading cause of death globally, taking an estimated 17.9 million lives each year, according to the World Health Organization. In this paper we will apply machine learning classification techniques to a dataset with diagnostic information of heart attacks with the aim of obtaining a predictive model to assist healthcare professionals in the early diagnosis of such diseases. The decision tree, random forest, support vector machine, naive bayes, artificial neural network and KNN algorithms were applied, with the latter obtaining the best result with an accuracy of 91.8%.*

Resumo. *As doenças cardiovasculares são a principal causa de morte em todo o mundo, tirando 17.9 milhões de vidas a cada ano, segundo a Organização Mundial da Saúde. Neste artigo vamos aplicar técnicas de classificação de aprendizado de máquina em um conjunto de dados com informações de diagnóstico de ataques cardíacos com o intuito de obter um modelo preditivo para auxiliar os profissionais de saúde no diagnóstico precoce de tais doenças. Foram aplicados os algoritmos árvore de decisão, random forest, máquina de vetor de suporte, naive bayes, rede neural artificial e KNN, com este último obtendo o melhor resultado com uma acurácia de 91.8%.*

1. Introdução

As doenças cardiovasculares são um grupo de desordens do coração e dos vasos sanguíneos que incluem doença arterial coronariana, acidente vascular cerebral(AVC), aneurisma, cardiopatia reumática e outras condições. Os mais importantes fatores de risco comportamentais para doenças cardiovasculares são dieta ruim, sedentarismo, tabagismo e alcoolismo. Os efeitos desses fatores de risco comportamentais podem se manifestar no indivíduo como hipertensão arterial, glicose elevada, colesterol alto, sobrepeso e obesidade. Esses fatores de risco podem ser medidos em unidades de atenção básica à saúde e indicar um risco elevado de ataque cardíaco, AVC, falência cardíaca e outras complicações.

Identificar esses fatores de risco e garantir o tratamento adequado pode prevenir mortes prematuras. No Brasil, cerca de 14 milhões de pessoas têm alguma doença cardiovascular e, pelo menos, 400 mil mortes ocorrem por ano, em decorrência dessas enfermidades, o que corresponde a 30% de todos os óbitos no país[1]. O diagnóstico de doenças cardíacas é uma das mais críticas e desafiadoras tarefas na área da saúde, precisa ser rápido, eficiente e correto para salvar vidas. Exige que o paciente faça muitos exames, e os profissionais de saúde devem examinar cuidadosamente os resultados.

É por isso que os pesquisadores têm se interessado em prever doenças cardíacas, e diferentes sistemas preditivos de doenças cardíacas foram desenvolvidos usando vários algoritmos de aprendizado de máquina[2]. Alguns deles alcançaram melhores resultados do que outros. Muitos usaram conjuntos de dados disponíveis em universidades ou plataformas como o *kaggle* para treinar e testar seu classificador, que é o caso deste artigo, enquanto outros usaram dados obtidos de outros hospitais acessíveis a eles. Neste trabalho vamos utilizar o conjunto de dados "*Cleveland"* disponibilizado pela Universidade da Califórnia em Irvine (UCI), EUA[3], para aplicar os algoritmos árvore de decisão, *random forest*, SVM, *Naive Bayes*, rede neural artificial e KNN para obter um modelo preditivo e comparar seus resultados. As seções seguintes deste artigo vão analisar os resultados obtidos em trabalhos anteriores, descrever o conjunto de dados, aplicar os algoritmos e analisar os resultados e finalmente discutir possíveis trabalhos futuros na área.

2. Revisão da literatura

Nesta seção vamos apresentar resultados obtidos anteriormente por outros pesquisadores que usaram aprendizado de máquina para obter um modelo preditivo de doenças cardíacas, para avaliar o que já foi feito na área e servir também como base de comparação para a execução do nosso próprio experimento.

Vembandasamy et al. [4] usou o algoritmo Naive Bayes para diagnosticar a presença ou ausência de doença cardíaca. O dataset usado na pesquisa foi obtido de um dos principais centros de pesquisa em diabetes de Chennai, India, e contém registros de 500 pacientes e 11 atributos incluindo o diagnóstico. A acurácia obtida foi de 86.4198%. Medhekar et al.[5] propôs um sistema para categorizar os dados em 5 categorias usando Naive Bayes. As categorias são, ausente, baixo, médio, alto e muito alto. O sistema prediz o grau de possibilidade de uma doença cardíaca, usando o dataset da UCI, o mesmo deste trabalho, e obteve uma acurácia de 88.96%.

Hossen [6] aplicou os algoritmos *Gradient Boosting*, *random forest*, KNN

support vector machine(SVM) e regressão logística em um *dataset* de doenças cardíacas obtendo as acurácias de 80%, 79%, 87%, 90% e 95% respectivamente. Eldouh et al.[7] usou a técnica AHP(*neutrosophic analytical hierarchy process*) para atribuir pesos aos atributos de um *dataset* de doenças cardíacas e posteriormente selecionar os mais relevantes para serem usados em modelos de aprendizado de máquina. Foram usados 9 modelos preditivos com o melhor resultado obtido com árvore de decisão e random forest com 100% de acurácia, seguidos por *Bagging*, KNN, e *gradient boosting* com 99%, 98%, e 97% de acurácia respectivamente. *AdaBoosting* teve 89%, e regressão logística e *Naive Bayes* tiveram 84%, e por último o de menor acurácia foi *support vector machine* (SVM) com 68%.

Sabay et al. [8] propôs superar o problema de poucos dados no dataset *Cleveland* da UCI[3], gerando dados sinteticamente a partir dos dados originais e comparar os resultados do conjunto de dados original com o de dados sintéticos. Usando o pacote *SynthPop* da linguagem R foi possível gerar dados sintéticos para aumentar o tamanho do dataset para 60000 registros (originalmente eram 303) possibilitando treinar uma rede neural artificial e obter um aumento de 16% na acurácia de um modelo preditivo de doenças cardíacas, para 96.7%.

3. O Dataset

Neste trabalho será usado o *dataset Cleveland* disponibilizado pela UCI, que consiste em 303 registros de pacientes com informações de gênero e idade e os resultados de diversos exames, mais o diagnóstico para chance aumentada para doença cardíaca ou não (Tabela 1). O dataset original consiste dos mesmos 303 registros, porém com 76 atributos com diversos valores ausentes e o atributo alvo possui 5 classes [3]. A versão que tivemos acesso foi obtida pela plataforma *kaggle* é a versão somente com os atributos relevantes, 14 incluindo o diagnóstico, e este possui somente 2 classes.

Tabela 1. Descrição dos atributos do dataset.

Atributo	Descrição
age	Idade em anos
sex	gênero (1 = masculino; 0 = feminino)
cp	Dor torácica (0 = angina típica; 1 = angina atípica; 2 = não anginosa; 3 = assintomático)
trtbps	Pressão arterial em repouso (mm Hg)
chol	Colesterol Total
fbs	Glicemia em jejum > 120 mg/dl (1 = verdadeiro; 0 = falso)
restecg	Eletrocardiograma(0 = normal; 1 = ST-T anormal; 2 = hipertrofia do ventrículo esquerdo)
thalachh	Frequência cardíaca máxima
exng	Angina induzida por atividade física (1 = sim; 0 = não)
oldpeak	Depressão do ST induzida por atividade física em relação ao repouso
slp	Inclinação do segmento ST durante exercício (0 = aclive; 1 = linear; 2 = declive)
caa	Número de vasos coloridos na fluoroscopia (0 – 3)
thall	0 = normal; 1 = dano irreversível; 3 = dano reversível
output	Diagnóstico (0 = normal; 1 = chance aumentada para doença cardíaca)

O dataset está razoavelmente balanceado em relação as classes do atributo alvo (Figura 1), com 54.5% na classe 1 e 45.5% na classe 0, e não possui valores ausentes em nenhum outro atributo, portanto não serão necessárias etapas adicionais de pré-processamento de dados além da normalização dos valores.

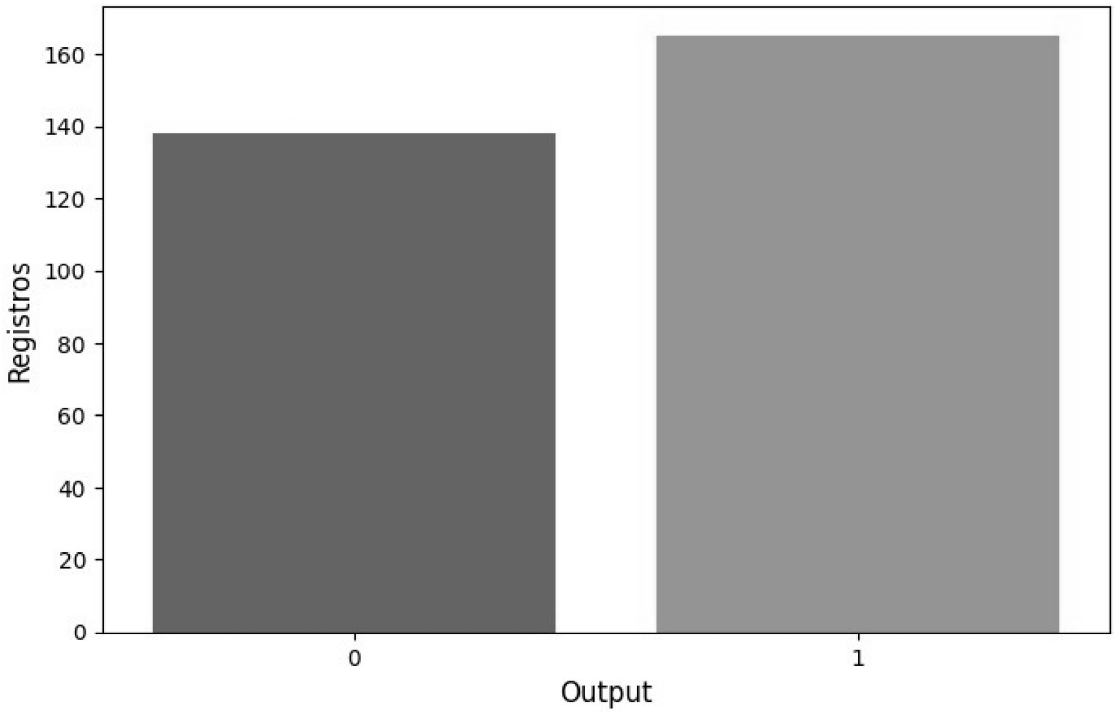

Figura 1. Distribuição dos valores nas classes do atributo alvo

4. O Experimento

Foi utilizado o Google Colab para criar jupyter notebooks usando a linguagem Python 3 e aplicar os algoritmos SVM, árvore de decisão, random forest, Naive Bayes, rede neural artifical e KNN ao dataset. O dataset foi divido entre conjunto de treino e teste usando o método Holdout, com o conjunto de teste variando entre 20% e 25% dependendo do algoritmo, para obter melhores resultados. Todos os valores númericos não binários foram normalizados e foi utilizado o framework scikit-learn [9] na implementação dos modelos, exceto pela rede neural artificial no qual também foi utilizado o framework TensorFlow [10]. Para comparar o desempenho dos modelos preditivos gerados foram utilizados a acurácia, matriz de confusão (Tabela 3), índice f1 e a curva ROC. Os índices de acurácia, indicam melhor desempenho como o do KNN com 91.8% e o pior com árvore de decisão com 83.6% (Tabela 2).

KNN obteve seu melhor desempenho com k = 7 e conjunto de testes igual a 20% do *dataset*. *Random Forest* foi aplicado com entropia como critério de decisão e 26 estimadores (árvores) obtendo acurácia de 88.5%. A rede neural foi implementada como uma MLP(*Multi Layer Perceptron*) com 2 camadas intermediárias de 8 neurônios usando função de ativação linear, e na camada de saída função sigmóide como ativação, obtendo 88.06% de acurácia. Os algoritmos *Naive Bayes*, SVM e árvore de decisão obtiveram acurácias de 89.55%, 88.15% e 83.6% respectivamente. A comparação das curvas ROC de KNN e árvore de decisão pode ser vista na figura 2 e os indicadores estão sumarizados na tabela 2.

Para efeito de comparação foram feitos testes usando validação cruzada como amostragem usando o método *cross_val_score* do scikit-learn dividindo o dataset em 8 partições e usando a acurácia como métrica de desempenho. O método retorna um *array*

das acurácias de cada iteração do k-fold, foi calculada a média e o algoritmo *random forest* obteve uma acurácia média de 88.94% com k= 8, uma melhora de apenas 0.47% em relação à acurácia obtida usando *holdout* e considerada desprezível. Todos os outros algoritmos apresentaram resultados inferiores usando validação cruzada como amostragem, usando diferentes valores de k, e seus resultados não serão considerados neste estudo.

Tabela 2. Desempenho dos modelos gerados pelos algoritmos(Holdout)

	Acurácia	F1	TVP	TFP
KNN	91.8	92.06	93.10	9.38
Bayes	89.55	89.85	90.63	12.12
DTREE	83.6	82.75	93.10	25.00
Random Forest	88.52	88.88	89.66	12.50
SVM	88.15	89.15	85.7	9.76
MLP	88.06	88.57	87.50	12.12

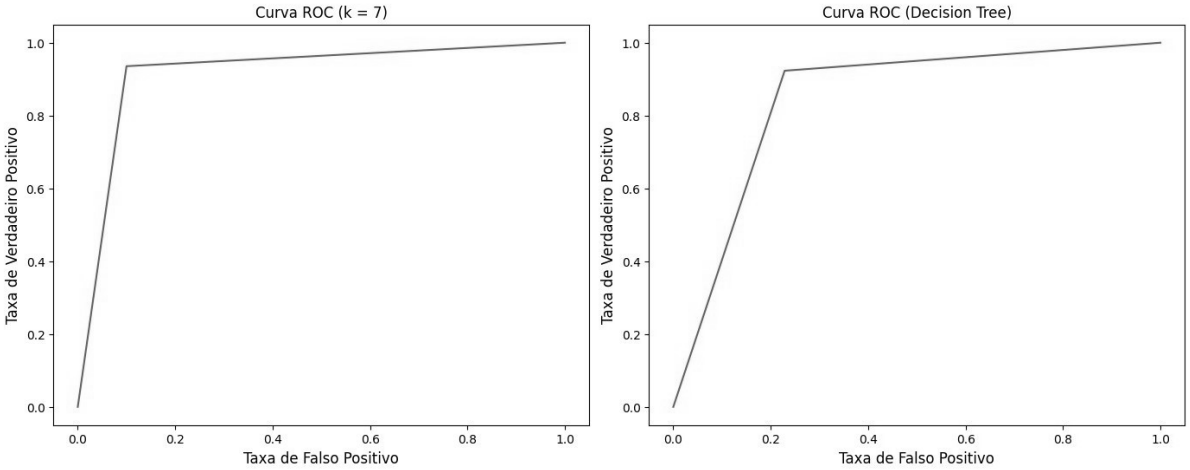

Figura 2. Curvas ROC do KNN (esquerda) e árvore de decisão(direita).

Tabela 3. Matrizes de confusão dos modelos gerados pelos algoritmos

KNN	Bayes	DTree	R.Forest	SVM	MLP
27 2 3 29	29 3 4 31	27 2 8 24	26 3 4 28	30 5 4 37	28 4 4 31

5. Trabalhos futuros

Tendo em vista os resultados obtidos por Eldouh et al.[7] e Sabay et al. [8] surge a necessidade de conseguir reproduzi-los com a mesma eficácia. O processo analítico hierárquico utilizado foi aplicado em um dataset muito pequeno e, poderia ser combinado com a técnica de produção de dados sintéticos para observar o desempenho em um conjunto de dados maior. Segundo a documentação do SynthPop ele deve ser aplicado em *datasets* com não menos que 500 registros [11], que não é o caso do *dataset* Cleveland, usado em [8] e também gostaria de replicar o experimento e ver se consigo obter os mesmos resultados.

Outro problema que surge no dataset utilizado neste trabalho e em outros, é que

devido ao seu pequeno tamanho, ele pode estar sujeito às características demográficas da população da qual ele foi obtido. Em outras palavras, ele pode não representar os resultados que seriam obtidos na população brasileira, e um futuro trabalho consistiria em construir um dataset com dados obtidos de instituições da área da saúde no Brasil, a Fiocruz e o Hospital Universitário da UFF.

Referências

[1] Saúde, Ministério da. “Usar o coração para cada coração”: 29/9 – Dia Mundial do Coração, Biblioteca Virtual em Saúde (2022). “Disponível em: <https://bvsms.saude.gov.br/usar-o-coracao-para-cada-coracao-29-9-dia-mundial-do-coracao/>

[2] N. Khateeb and M. Usman, “Efficient heart disease prediction system using k-nearest neighbor classification technique,” in Proceedings of the International Conference on Big Data and Internet of Thing (BDIOT), New York, NY, USA: ACM, 2017, pp. 21–26

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

[4] K. Vembandasamy, R. Sasipriya, and E. Deepa, “Heart diseases detection using naive bayes algorithm,” International Journal of Innovative Science, Engineering & Technology, vol. 2, no. 9, pp. 441–444, 2015.

[5] D. Medhekar, M. Bote, and S. Deshmukh, “Heart disease prediction system using naive bayes,” International Journal of Enhanced Research In Science Technology & Engineering, vol. 2, no. 3, pp. 1–5

[6] Hossen, Mohammed Khalid. “Heart Disease Prediction Using Machine Learning Techniques”. American Journal of Computer Science and Technology. Vol. 5, No. 3, 2022, pp. 146-154. doi: 10.11648/j.ajcst.20220503.11

[7] Eldouh, Ahmed & Lu, SongFeng & Abdelhafeez, Ahmad & Ali, Ahmed & Aziz, Alber. (2023). Heart Disease Prediction under Machine Learning and Association Rules under Neutrosophic Environment. Neutrosophic Systems with Applications. 10. 35-52. 10.61356/j.nswa.2023.75.

[8] Sabay, Alfeo; Harris, Laurie; Bejugama, Vivek; and Jaceldo-Siegl, Karen (2018) "Overcoming Small Data Limitations in Heart Disease Prediction by Using Surrogate Data," SMU Data Science Review: Vol. 1: No. 3, Article 12. Available at: <https://scholar.smu.edu/datasciencereview/vol1/iss3/12>

[9] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[10] Abadi, Martin; et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[11] Nowok, B., G.M. Raab & C. Dibben (2016), synthpop: Bespoke creation of synthetic data in R. Journal of Statistical Software, 74:1-26; DOI:10.18637/jss.v074.i11. Available at: https://www.jstatsoft.org/article/view/v074i11