Relatório da Diciplina *Machine Learning*: Detecção de Estágios da Fibrose Hepática - *Hepatitis C Virus (HCV) Dataset*

John W. S. de Lima[1], **Tasso L. O. de Moraes**[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)

{jwsl, tlom}@cin.ufpe.br

1. Introdução

A Hepatite C é a inflamação do fígado causada pelo vírus HCV (*hepatitis C virus*)[de Morais Carvalho et al. 2020]. Tal inflamação pode levar a lesões no fígado e, devido ao processo de cicatrização, o tecido fibroso do fígado se acumula em excesso, o que constitui a fibrose hepática [Andrade 2005]. De acordo com a Organização Mundial da Saúde (OMS), estima-se que cerca de 120-130 milhões da população mundial estão infectados com HCV e, anualmente, existem cerca de 3-4 milhões de novos casos, representando um dos principais problemas de saúde pública do mundo [Morozov e Lagaye 2018]. A infecção por HCV normalmente é assintomática ou com poucos sintomas visíveis durante o estágio inicial de infecção [Serejo et al. 2007, Elgharably et al. 2017]. Vale ressaltar que não existe vacina para prevenção da Hepatite C [de Oliveira et al. 2020]. Assim, essas infecções podem progredir para infecções agudas, depois para infecções crônicas, seguidas por doenças hepáticas, como cirrose e carcinoma hepatocelular [Elgharably et al. 2017].

Monitorar a evolução da infecção por HCV é fundamental para o acompanhamento do tratamento da hepatite C [Blatt et al. 2009, Nandipati et al. 2020]. Um dos métodos que tem se mostrado útil para a detecção e monitoramento dos estágios da fibrose hepática é a biópsia hepática. No entanto, este método é doloroso, invasivo e de alto custo [Bravo et al. 2001, Nandipati et al. 2020]. Esses pontos negativos levaram à pesquisa de diagnósticos não invasivos e com menor custo, como biomarcadores bioquímicos séricos e diagnósticos por imagem (ultrassom, ressonância magnética, etc.) [Baranova et al. 2011, Patel e Sebastiani 2020]. Esses métodos juntamente com os processos do próprio sistema de saúde podem gerar grandes quantidades de dados. Embora esses dados possam conter informações importantes, muitas vezes eles não são usados de maneira adequada para dar suporte à decisão clínica e ao monitoramento de doenças [Palanisamy e Thirunavukarasu 2019, Nandipati et al. 2020]. Desse modo, a integração de sistemas de suporte à decisão clínica e registros de pacientes com diferentes técnicas de classificação de aprendizado de máquina e métodos de seleção de *features* são úteis para a previsão e monitoramento de diferentes doenças. [Reddy et al. 2019]. Além disso, nos últimos anos pesquisadores e médicos tem aplicando técnicas de classificação de aprendizado de máquina e algoritmos de seleção de *features* para diagnosticar e monitorar doenças usando dados clínicos e bioquímicos obtidos a partir de métodos não invasivos [Nandipati et al. 2020]. Desse modo, o objetivo desse relatório é analisar e aplicar alguns dos principais algoritmos de aprendizado de máquina (Seção 2) para detecção de estágios da fibrose hepática.

2. Algoritmos de Aprendizagem de Máquina

2.1. *K Nearest Neighbor (KNN)*

KNN é um algoritmo supervisionado que pode ser usado tanto para classificação quanto para regressão [Ferrero 2009]. Basicamente ele funciona da seguinte maneira: dado um conjunto de teste, o algoritmo encontra as K instancias mais próximas da instancia de teste Xt. Em seguida, a classe de Xt é definida pela a maior ocorrência de classes entre as K instâncias mais próximas de Xt [Aha et al. 1991]. O calculo da distância pode ser realizado de diversas formas, sendo o mais comum a distância euclidiana. Também pode-se atribuir um peso para cada instância de acordo com a distância da mesma. Embora ele seja um algoritmo eficiente, ele apresenta algumas desvantagens: como todas as instâncias são armazenadas para posterior consulta esse algoritmo pode demandar muita memória; Na fase de testes ele demanda esforço computacional [Darmiton 2020].

2.2. *Decision tree (DT)*

DT é um algoritmo supervisionado onde os dados são representados em uma estrutura de árvore. Além disso, cada nó interno da árvore denota um teste com um atributo de entrada, cada ramificação representa um resultado do teste e cada nó folha contém o rótulo de classe em que um atributo de entrada pode pertencer. A DT possui uma boa interpretabilidade e é uma estrutura rápida para encontrar a classificação de atributos. No entanto, o seu treinamento tem um alto custo computacional e são pouco eficientes para a tarefa de predição de valores contínuos [Quinlan 1986, Singh e Kumar 2020].

2.3. *Multilayer Perceptron (MLP)*

MLP é um algoritmo supervisionado e bioinspirado nos neurônios do cérebro humano. Foi desenvolvido a partir das limitações encontradas pelo algoritmo *perceptron*. Enquanto o *perceptron* apenas consegue resolver problemas lineares, o MLP pode resolver problemas de classificação lineares e não-lineares. No MLP existem camadas de elementos (chamados de neurônios) que são ligados a outros neurônios de uma outra camada posterior. As ligações entre neurônios ocorrem através de pesos e cada neurônio tem uma função de ativação associada. Desse modo, os dados das instâncias entram na rede e, a depender dos cálculos realizados, a função de ativação pode habilitar ou não o disparo de um neurônio. O MLP é um algoritmo muito bom para aquisição de conhecimentos empíricos a partir de uma base de conhecimento. No entanto, o seu processo de treinamento é lento [Gardner e Dorling 1998].

2.4. *Naive Bayes (NB)*

É um classificador inspirado no teorema de Bayes. De maneira breve, ele busca calcular a probabilidade de uma instância pertencer a uma determinada classe dado um conjunto de características de suas variáveis. Classe em que a instância apresenta a maior probabilidade de pertencer será a classe determinada pelo algoritmo. Embora esse algoritmo possui um treinamento rápido e, também, é rápido para classificar uma nova instância, ele assume que as características da instância são independentes [Frank et al. 2000].

2.5. *Support Vector Machine (SVM)*

SVM é um algoritmo supervisionado que pode ser usado tanto para classificação quanto para regressão. Dado um conjunto de atributos, o SVM buscará delimitar esses atributos entre fronteiras (retas ou hiperplanos, dependendo do espaço N-dimensional) de modo a agrupa-los em classes. O SVM é eficiente no trabalho com grandes conjuntos de exemplos e o processo de classificação é rápido. Entretanto, o tempo de treinamento pode ser bem longo [Noble 2006].

3. Experimentos

3.1. Banco de Dados

O conjunto de dados (*dataset*) utilizado foi disponibilizado em 2019 no repositório de Aprendizagem de Máquina [Dua e Graff 2017] da Universidade da California campus Irvine, contendo 1385 registros de pacientes Egipicios que realizaram tratamento para HCV. Além disso, esse conjunto de dados contém 29 variáveis. Dessas 29 variáveis, 27 são variáveis de entrada e 2 são variáveis que podem ser consideradas *target*. As variáveis de entrada contém informações sobre o paciente (idade, indice de massa corpórea e etc), informações sobre alguns sintomas clínicos (febre, dor de cabeça e etc) e informações sobre resultados de exames por métodos não invasivos. As variáveis *target* contém informações sobre o grau do processo necroinflamatório nas células do fígado (chamado de *grading* [Batts e Ludwig 1995]) e o grau de fibrose hepática (chamado de *staging* [Shiha e Zalata 2011]). Esse trabalho foca no estudo do estágio da doença a partir da fibrose hepática. Assim, estaremos apenas considerando a variável target que foca no *staging*. Tal variável que no *dataset* é chamada de *"Baseline histological staging"* contém 4 classificações (F1,F2,F3 e F4) baseadas no sistema de pontuação de METAVIR [Shiha e Zalata 2011]. Segue a metodologia utilizada nesse estudo:

1. Primeiramente, foi feito a análise exploratória de Dados objetivando conhecer o comportamento dos dados. Para tanto, foi realizado o pré-processamento dos dados, verificando dados ausentes e *outliers*, por exemplo. E ainda, realizando uma análise estatística a fim de verificar a distribuição dos dados. Desse modo, constatamos que os dados estão aproximadamente balanceados com os *labels* dos classificadores tendo as seguintes proporções 24.25%(F1), 23.97%(F2), 25.63%(F3) e 26.13%(F4).
2. Em seguida, foram feitas as aplicações dos algoritmos de aprendizado de máquina incluindo os métodos *10-fold cross-validation* e *grid-search*. Vale ressaltar que para o método *10-fold cross-validation* a divisão dos *datastes* fica de 90% para treino e 10% para teste. A fim de que os dados fossem comparados utilizando o mesmo conjunto de dados, cada modelo citado na seção anterior foi aplicado em cada *fold*. Antes de cada modelo ser executado os seus parâmetros eram escolhidos utilizando o método *grid-search* com os 90% dedicado ao conjunto de treino e só depois o modelo é aplicado nos outros 10% dedicados ao conjunto de teste para se obter os resultados. Esse processo é ilustrado no fluxograma da Figura 1.
   A função utilizada para realizar o *grid-search* realiza internamente um *cross-validation* dividindo os dados em teste e validação. Neste estudo foi utilizado um 3-fold como parâmetro de tal função para a tarefa de escolha das variáveis. Os valores utilizados na variação dos parâmetros dos modelos estão descriminados na

Tabela 1. O significado de cada parâmetro pode ser encontrado na documentação das funções do *sickit-learn* para cada modelo de classificação.

3. Uma vez obtidos os restultados dos modelos para todos os 10 *folds*, tirou-se a média e o desvio padrão para cada uma das métricas, explicadas na próxima seção e apresentadas na Tabela 2, com exceção da matriz de confusão que apesar de também ser representada pela média dos resultados não possui desvio padrão. Os resultados de *f1-score* e *Area Under the ROC Curve* levam em conta a distribuição das classes no conjunto de dados utilizando um peso para cada classe de acordo com sua quantidade no conjunto de dados.

As ferramentas utilizadas para realização do trabalho foram: a linguagem *Python*; a biblioteca *scikit-learn, pandas, mlxtend, numpy, matplotlib, seaborn* e o ambiente *Jupyter Notebook*. Toda a implementação pode ser encontrada em: `https://github.com/tassomoraes/ml-hepatitis`

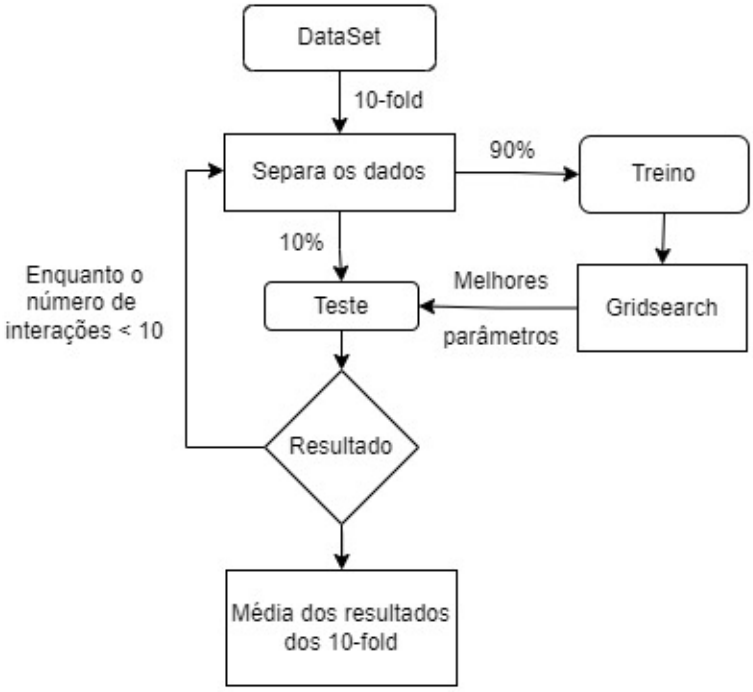

Figura 1. Fluxograma da obtenção dos resultados.

3.2. Métricas

Para a avaliação dos experimentos usamos as seguintes métricas: matriz de confusão, *f1-score, accuracy* e *Area Under the ROC Curve*.

3.2.1. Matriz de Confusão e Métricas

Uma matriz de confusão é uma matriz N x N usada para avaliar o desempenho de um modelo de classificação, onde N é o número de classes alvo. A matriz compara os valores reais das variáveis alvo com os previstos pelo modelo de aprendizado de máquina. Ou seja, ela mostra o número de classificações corretas versus as classificações preditas para

Tabela 1. Valores dos parâmetros

KNN	$n_neighbors$	1,3,5,...,31
	weights	uniform, distance
	dist	euclidian, manhattan, chebyshev
Decision Tree	max_depth	1,2,3,4,...,32
	criterion	gini, entropy
MLP	hideen_leyer_sizes	(100,), (50, 15, 5), (100, 25, 10)
	learning_rate_init	0.05, 0.0001
	solver	sgd, adam
	activation	tanh, relu
	learning_rate	constant, adaptive
SVM	C	0.1, 1, 10, 100
	kernel	rbf, sigmoid
	gamma	1, 0.1, 0.01, 0.001
Gaussian Naive Bayes	não paramétrico	-

cada classe, sobre um conjunto de exemplos [Castro e Braga 2011]. A Figura 2 mostra uma matriz de confusão para um conjunto com duas classes. Note, por exemplo, que FP significa a quantidade de falsos positivos previsto pelo modelo de classificação. A partir da matriz da Figura 2, vamos mostrar as principais métricas usadas nas avaliações das máquinas de aprendizagem [Castro e Braga 2011]:

- *Accuracy (Acc)*: é a taxa de previsões corretas para os dados de teste. Ela é calculada dividindo o número de previsões corretas pelo número de previsões totais.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- *Precision (P)*: é a taxa de exemplos relevantes (verdadeiros positivos) entre todos os exemplos que foram previstos para pertencer a uma determinada classe.

$$P = \frac{TP}{TP + FP} \tag{2}$$

- *Recall (R)*: é a taxa de exemplos que foram previstos como pertencentes a uma classe em relação a todos os exemplos que realmente pertencem a esta classe. Ele também é chamado de Taxa de Verdadeiros Positivos.

$$R = \frac{TP}{TP + FN} \tag{3}$$

- *F1-Score (F1)*: é uma medida que combina P e R. A informação mostrada pelo F1 se torna mais útil do que a *Accuracy* quando os dados estão desbalanceados.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{4}$$

- Taxa de Falsos Positivos: é a taxa de exemplos que não foram previstos como pertencentes a uma classe em relação a todos os exemplos que foram previstos para pertencer a uma determinada classe. Essa métrica será importante para a construção da curva ROC.

$$TaxadeFalsosPositivos = \frac{FP}{TP + FP} \tag{5}$$

Classe	predita C_+	predita C_-
verdadeira C_+	Verdadeiros positivos T_P	Falsos negativos F_N
verdadeira C_-	Falsos positivos F_P	Verdadeiros negativos T_N

Figura 2. Matriz de confusão para um conjunto com duas classes [Monard e Baranauskas 2003].

3.2.2. Curva ROC e *Area Under the ROC Curve (AUC)*

A ROC(*Receiver Operating Characteristics*) é uma curva (Figura 3) desenhada a partir da Taxa de Verdadeiros Positivos (eixo Y) e da Taxa de Falsos Positivos (eixo X). A Curva ROC é bastante útil no trato com conjuntos cujas classes estejam desbalanceadas. Além disso, a partir da curva ROC, podemos obter uma importante medida chamada de *Area Under the ROC Curve (AUC)*. A área AUC varia de 0 a 1 e fornece uma medida geral da capacidade de discriminação de um classificador, possibilitando a avaliação de diferentes classificadores. Vale ressaltar que, quanto maior a área AUC mais eficiente será o algoritmo [Prati et al. 2008, Castro e Braga 2011].

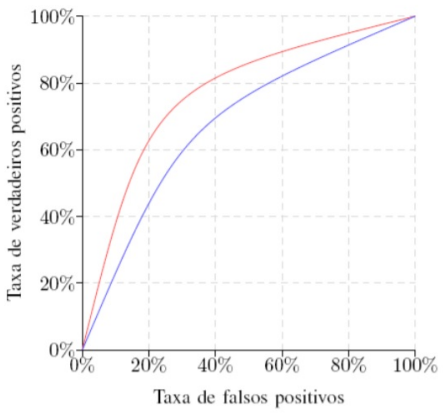

Figura 3. Exemplos com duas curvas ROC [Prati et al. 2008].

3.3. Resultados

Nos resultados foi possível perceber que todos os modelo apresentaram uma baixa capacidade de classificação para esse conjunto de dados. Apesar de o modelo MLP apresentar os piores resultados em relação a valores: acurácia de $22.95\,\% \pm 2.90\,\%$, *f1-score* de $21.08\,\% \pm 3.18\,\%$ e *AUC* de $47.54\,\% \pm 2.60\,\%$ é o SVM que se comporta como pior modelo de classificação. No modelo SVM, é possível observar que o valor da acurácia ($26.06\,\% \pm 0.05\,\%$) tem uma considerável diferença do valor do *f1-score* ($10.78\,\% \pm 0.04\,\%$) e o valor da *AUC* é igual a 50%, tais valores indicam que o modelo não está conseguindo diferenciar as classes de forma correta. Tal afirmação pode ser confirmada pela matriz de confusão do modelo SVM mostrada na Figura 4 que indica que o modelo só classificou todas as instâncias como classe 4.

Apesar dos resultados apresentados na Tabela 2 possuírem valores muito próximos, pode-se dizer que o KNN é o modelo com melhor performance e que obteve valores da acurácia ($25.70\,\% \pm 4.34\,\%$) e do *f1-score* ($25.51\,\% \pm 4.38\,\%$) próximos a pesar de ter um valor da *AUC* próximo de 50% ($50.43\,\% \pm 3.09\,\%$). Também é possível verificar a classificação deste modelo de forma mais detalhada observando sua matriz de confusão na Figura 2. Métodos de seleção de atributos e uma busca mais minuciosa pelos parâmetros dos modelos podem melhorar os resultados.

Tabela 2. Resultados dos modelos

Modelo	**Acurácia**	**F1-Score**	**AUROC**
KNN	$25{,}706\% \pm 4{,}343\%$	$25{,}512\% \pm 4{,}383\%$	$50{,}438\% \pm 3{,}090\%$
Decision Tree	$25{,}052\% \pm 2{,}987\%$	$20{,}955\% \pm 4{,}499\%$	$49{,}060\% \pm 1{,}494\%$
MLP	$22{,}951\% \pm 2{,}905\%$	$21{,}083\% \pm 3{,}184\%$	$47{,}541\% \pm 2{,}607\%$
Gaussian Naive Bayes	$23{,}900\% \pm 3{,}840\%$	$22{,}566\% \pm 3{,}919\%$	$48{,}118\% \pm 3{,}039\%$
SVM	$26{,}068\% \pm 0{,}056\%$	$10{,}780\% \pm 0{,}041\%$	$50{,}000\% \pm 0{,}000\%$

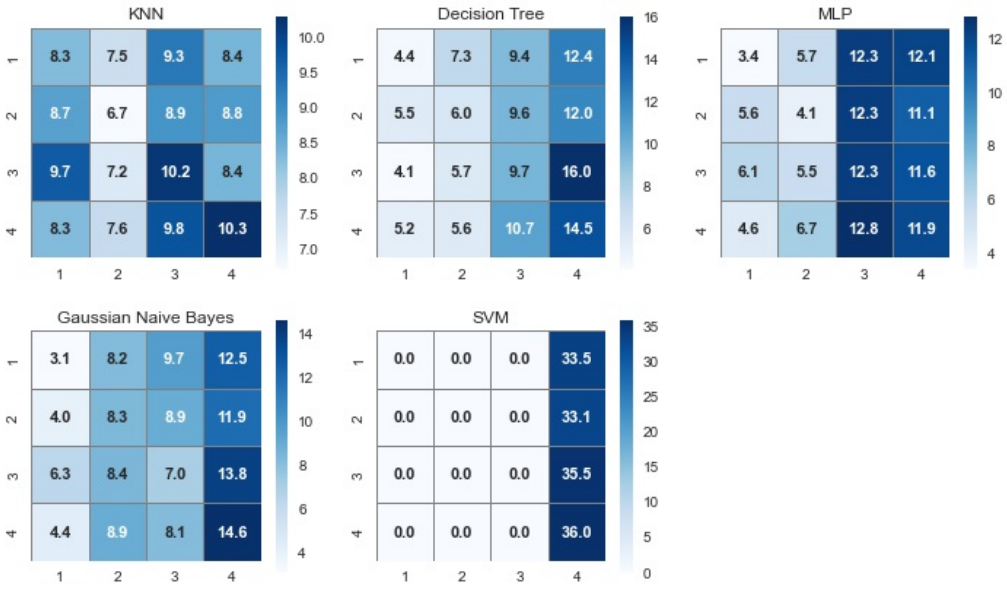

Figura 4. Matrizes de confusão [Prati et al. 2008].

Referências

Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, 6(1):37–66.

Andrade, Z. A. (2005). Regressão da fibrose hepática. *Revista da Sociedade Brasileira de Medicina Tropical*, 38:514–520.

Baranova, A., Lal, P., Birerdinc, A., and Younossi, Z. M. (2011). Non-invasive markers for hepatic fibrosis. *BMC gastroenterology*, 11(1):1–15.

Batts, K. P. and Ludwig, J. (1995). Chronic hepatitis. an update on terminology and reporting. *The American journal of surgical pathology*, 19(12):1409–1417.

Blatt, C. R., Rosa, J., Sander, G., and Farias, M. R. (2009). Tratamento da hepatite c e qualidade de vida. *Rev. Bras. Farm*, 90(1):19–26.

Bravo, A. A., Sheth, S. G., and Chopra, S. (2001). Liver biopsy. *New England Journal of Medicine*, 344(7):495–500.

Castro, C. L. d. and Braga, A. P. (2011). Supervised learning with imbalanced data sets: an overview. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 22(5):441–466.

Darmiton, G. (2020). Máquinas que aprendem (map) k-vizinhos mais próximos: uma análise. Disponível em: `https://maquinasqueaprendem.com/2020/06/22/k-vizinhos-mais-proximos-uma-analise/`. Acesso em: 09 janeiro 2021.

de Morais Carvalho, C. Í., da Silva Ferreira, V., and de Rodrigues Leitão, J. M. S. (2020). Perfil epidemiológico de pacientes com hepatite c no componente especializado da assistência farmacêutica do piauí. *Research, Society and Development*, 9(3):e06932265–e06932265.

de Oliveira, E. H., Holanda, E. C., de Almeida, A. J. A., Verde, R. M. C. L., Sousa, F. d. C. A., de Andrade, S. M., and Cunha, M. A. (2020). Aspectos clínicos e epidemiológicos dos casos de hepatite c no estado do maranhão, brasil. *Research, Society and Development*, 9(7):e120973720–e120973720.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Elgharably, A., Gomaa, A. I., Crossey, M. M., Norsworthy, P. J., Waked, I., and Taylor-Robinson, S. D. (2017). Hepatitis c in egypt–past, present, and future. *International journal of general medicine*, 10:1.

Ferrero, C. A. (2009). *Algoritmo kNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia*. PhD thesis, Universidade de São Paulo.

Frank, E., Trigg, L., Holmes, G., and Witten, I. H. (2000). Naive bayes for regression. *Machine Learning*, 41(1):5–25.

Gardner, M. W. and Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636.

Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32.

Morozov, V. A. and Lagaye, S. (2018). Hepatitis c virus: Morphogenesis, infection and therapy. *World journal of hepatology*, 10(2):186.

Nandipati, S. C., XinYing, C., and Wah, K. K. (2020). Hepatitis c virus (hcv) prediction by machine learning techniques. *Applications of Modelling and Simulation*, 4:89–100.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

Palanisamy, V. and Thirunavukarasu, R. (2019). Implications of big data analytics in developing healthcare frameworks–a review. *Journal of King Saud University-Computer and Information Sciences*, 31(4):415–425.

Patel, K. and Sebastiani, G. (2020). Limitations of non-invasive tests for assessment of liver fibrosis. *JHEP Reports*, 2(2):100067.

Prati, R., Batista, G., Monard, M., et al. (2008). Curvas roc para avaliação de classificadores. *Revista IEEE América Latina*, 6(2):215–222.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Reddy, N. S. C., Nee, S. S., Min, L. Z., and Ying, C. X. (2019). Classification and feature selection approaches by machine learning techniques: Heart disease prediction. *International Journal of Innovative Computing*, 9(1).

Serejo, F., Marinho, R., Velosa, J., Costa, A., and Moura, M. (2007). Elastografia hepática transitória: um método não invasivo para avaliação da fibrose em doentes com hepatite c crónica. *Jornal Português de Gastrenterologia*, 14(1):8–15.

Shiha, G. and Zalata, K. (2011). Ishak versus metavir: terminology, convertibility and correlation with laboratory changes in chronic hepatitis c. *Liver biopsy*, 10:155–170.

Singh, A. and Kumar, R. (2020). Heart disease prediction using machine learning algorithms. In *2020 international conference on electrical and electronics engineering (ICE3)*, pages 452–457. IEEE.