**Uso de Aprendizado de Máquina para Predição de Mortes por Doença Cardíaca**

**Alunos**

* Fábio Andrade
* Gismar Barbosa
* João Amazonas
* Laís Gregório
* Thainnara Lima

**Resumo**

<Escrever no final>

**Introdução**

Doenças cardiovasculares afetam o sistema circulatório e são a principal causa de mortalidade global, com 17,9 milhões de mortes em 2016 (OPAS/OMS, 2024). Essas doenças, causadas principalmente por placas nas artérias, podem ser prevenidas por meio do controle de fatores de risco comportamentais, como tabagismo, dieta inadequada e sedentarismo (Bourbon et al., 2016).

Levando em consideração que o aprendizado de máquina contribui para otimizar recursos na saúde, priorizando atendimentos críticos e reduzindo custos, com base no dataset de Chicco e Jurman (2020), contendo os dados de 299 pacientes com 12 características clínicas, este artigo, aplicando técnicas de KDD (Souza, 2023) e, ferramentas de aprendizado de máquina (ML), tem como objetivo principal, analisar a eficiência, dos métodos supervisionados de ML (classificação de dados) para predizer morte por insuficiência cardíaca a partir dos dados clínicos deste dataset. Essa abordagem, em uma aplicação do mudo real, poderia aprimorar a detecção precoce e, gestão das doenças cardiovasculares, sendo uma ferramenta estratégica para diminuição da mortalidade.

**Metodologia**

*Dados*

Os dados, originais (dataset), podem ser visualizados a partir a partir do repositório “Heart Failure Clinical Records” (2020), sendo possível baixá-los (formato de arquivo CSV) na url “https://archive.ics.uci.edu/static/public/519/heart+failure+clinical+records.zip”. Os dados são rotulados pelos seguintes atributos (ver em tabela 1):

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Atributo | Tipo | Descrição | Unidade de Medida |
| age | Inteiro | idade do paciente | Anos |
| anaemia | Binário | diminuição de glóbulos vermelhos ou hemoglobina |  |
| creatinine\_phosphokinase | Inteiro | nível da enzina CPK no sangue | mcg/L |
| diabetes | Binário | se o paciente tem diabetes |  |
| ejection\_fraction | Inteiro | porcentagem de sangue saindo do coração a cada contração | % |
| high\_blood\_pressure | Binário | se o paciente tem hipertensão |  |
| platelets | Continuo | plaquetas no sangue | kiloplateIets/mL |
| serum creatinine | Continuo | nível de sódio no sangue | rnEqL |
| sex | Binário | mulher ou homem |  |
| smoking | Binário | se o paciente fuma ou não |  |
| time | Inteiro | período de acompanhamento do paciente | dias |
| death event | Binário | se o paciente faleceu durante o período de acompanhamento |  |

Esses dados, para facilitar acesso, assim como a reprodução e/ou aplicação deste processo metodológico, tais dados se encontram disponível, para consulta, no repositório (link do repositório) (referência do repositório).

*Weka*

<Explanar, 3 linhas sobre a ferramenta, com refeências>

Waikato Environment for Knowledge Analysis ( Weka ) é uma coleção de software livre, de aprendizado de máquina e análise de dados licenciado sob a [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) . Foi desenvolvido na [Universidade de Waikato](https://en.wikipedia.org/wiki/University_of_Waikato) , [Nova Zelândia](https://en.wikipedia.org/wiki/New_Zealand) e é o software complementar do livro "Data Mining: Practical Machine Learning Tools and Techniques". Referência: https://en.wikipedia.org/wiki/Weka\_(software)

Weka (Waikato Environment for Knowledge Analysis) é uma ferramenta de aprendizado de máquina desenvolvida pela Universidade de Waikato, na Nova Zelândia. Ele é amplamente utilizado para análise de dados e mineração de conhecimento, oferecendo uma interface gráfica intuitiva e suporte a diversos algoritmos para tarefas como classificação, regressão, clustering e pré-processamento de dados. O Weka é especialmente popular por sua flexibilidade, permitindo o uso de técnicas avançadas de análise sem necessidade de programação, além de suportar formatos de dados como .arff e .csv. Ele é frequentemente empregado em contextos acadêmicos e profissionais devido à sua capacidade de facilitar experimentos rápidos e detalhados em aprendizado de máquina.

*KDD*

Este projeto está sendo construído como continuidade do projeto da disciplina de Mineração de Dados (CET-0611), onde os processos que envolvem o KDD foram explorados e aplicados ao conjunto de dados (dataset), visando a mineração de dados. Portanto, neste trabalho, como o foco são os processos de classificação de (predição) de dados, não iremos replicar os passos do projeto anterior. O processo de KDD, aqui, será realizado a partir de pré-processamentos necessários para os classificadores de dados (caso necessário) para normalização de dados.

Nesse ponto é interessante citar que, quando houver pré-processamento, ele será citado no processo intrínseco ao classificador utilizado (durante a tarefa de classificação).

Mas, existe um processo necessário, porém não obrigatório, relacionado ao processo de pré-processamento que a conversão do arquivo CSV para o padrão ARFF:

Processo de conversão CSV para ARFF <passo a passo>

Nosso pré-processamento: Utilizando o W|eka abrimos o arquivo CSV, vimos que o Weka não carregou a classe corretamente, então optamos em converter o arquivo CSV em ARFF. Opção “save” em “Files of Types” escolhemos o tipo de arquivo “Arff data files” (\*.ARFF) e em seguida opção “SAVE”. Em seguida abrimos o arquivo novamente agora o no formato (.ARFF), após abrir vimos que o Weka ainda não identificou a classe, para corrigir esse problema utilizando um editor de texto de sua preferência abrimos o arquivo (.ARFF) e na linha 15 substituímos o valor (NUMERIC) por {0,1} e então salvamos o arquivo. Após esse pré-processamento ao abrir o arquivo novamente no Weka, ele irá identificar a classe corretamente.

*Classificadores*

Penso que, ainda que já saibamos o que cada classificador faça, é bom defini-lo, com uma referência para cada um.

Naïve Bayes

O Naïve Bayes é um algoritmo de aprendizado supervisionado baseado no Teorema de Bayes, usado principalmente para classificação. Ele assume que os atributos do conjunto de dados são independentes entre si (hipótese "ingênua") e calcula a probabilidade de uma instância pertencer a cada classe, atribuindo-a à classe com maior probabilidade. É eficiente, fácil de implementar e amplamente aplicado em tarefas como filtragem de spam, análise de sentimentos e categorização de textos.

Passo a Passo:

Classificação NaiveBayes: Com o arquivo aberto no Weka Na aba "Classify", onde estão disponíveis os algoritmos de aprendizado de máquina. No campo "Classifier", clique em "Choose", expanda o menu "Bayes" e selecione "NaiveBayes". No painel "Test Options", escolha a opção "Cross-validation" e configure o número de folds como 10 no campo correspondente. Com tudo configurado, clique em "Start" para iniciar a classificação.

Referencia: <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>

k-NN

O k-NN (k-Nearest Neighbors) é um algoritmo de aprendizado supervisionado usado para classificação e regressão. Ele funciona identificando os "k" vizinhos mais próximos de uma amostra com base em uma métrica de distância, como a Euclidiana, e classifica a amostra com a categoria predominante entre os vizinhos ou prevê o valor médio (no caso de regressão). Simples e eficaz, o k-NN é amplamente utilizado para problemas de análise de dados, especialmente quando as relações entre as classes não são lineares.

Passo a Passo:

Classificação k-NN: Com o arquivo aberto no Weka . Na aba "Classify", no campo "Classifier", foi selecionada a opção "Choose", acessando o menu "Lazy" e escolhendo "IBk", que corresponde ao algoritmo k-NN. No painel "Test Options", foi selecionada a opção "Cross-validation" com o número de folds configurado para 10. Vários testes foram realizados com diferentes valores de k, incluindo 1, 2, 4 e 5, para identificar o parâmetro mais adequado. Após as execuções, verificou-se que o melhor desempenho foi alcançado com k = 3.

Referencia https://link.springer.com/chapter/10.1007/978-0-387-88615-2\_4

MLP

O perceptron multicamadas (MLP) é uma [rede neural](https://pt.wikipedia.org/wiki/Rede_neural) semelhante ao [perceptron](https://pt.wikipedia.org/wiki/Perceptron) simples, porém possui mais de uma camada de [neurônios](https://pt.wikipedia.org/wiki/Neur%C3%B4nio_artificial). Em casos em que não há a possibilidade de uma única reta separar os elementos, há o uso da MLP que, gera mais de uma reta classificadora.

Referencia: https://pt.wikipedia.org/wiki/Perceptron\_multicamadas#:~:text=O%20perceptron%20multicamadas%20(MLP)%20%C3%A9,mais%20de%20uma%20reta%20classificadora.

Passo a Passo:

Classificação MLP: Com o arquivo aberto no Weka Na aba “Classify” opção “Classifier” escolhemos a opção “Functions” e selecionamos “MultilayerPerceptron”. Nos parâmetros do MLP utilizamos no camp “hiddenLayers” colocamos 3 camadas com 10 neurônios cada (10,10,10), “nominalToBinaryFilter” = True, “normalizeAttributes” = True e “normalizeNumericClass” = True e clicamos em “OK”. Utilizamos “Cross-validation Folds” = 10 e Start.

Realizamos algumas rodadas de testes até chegar nesse ajuste fino de 3 Camadas com 10 Neurônios como o que nos trouxe o melhor resultado a ser discutido. Toda a normalização dos campos ficou por conta do próprio classificador através dos seus parâmetros mencionado acima.

Ensembles Bagging

Métodos ensemble são técnicas poderosas em machine learning que combinam múltiplos modelos para melhorar a precisão geral da predição e a estabilidade do modelo. Bootstrap Aggregating, mais conhecido como Bagging, se destaca como um método ensemble popular e amplamente implementado.

<Referencia:https://www.datacamp.com/tutorial/what-bagging-in-machine-learning-a-guide-with-examples >

Passo a Passo:

Bagging NaiveBayes: Com o arquivo aberto no Weka na aba "Classify", onde estão disponíveis os algoritmos de aprendizado de máquina. No campo "Classifier", clique em "Choose", expanda o menu "Meta" e selecione "Bagging". Nos parâmetros do Bagging em “classifier” clicamos em “choose”, expanda o menu "Bayes" e selecione "NaiveBayes" e “OK”. No painel "Test Options", escolha a opção "Cross-validation" e configure o número de folds como 10 no campo correspondente. Com tudo configurado, clique em "Start" para iniciar a classificação.

Bagging IBK(KNN): Com o arquivo aberto no Weka na aba "Classify", onde estão disponíveis os algoritmos de aprendizado de máquina. No campo "Classifier", clique em "Choose", expanda o menu "Meta" e selecione "Bagging". Nos parâmetros do Bagging em “classifier” clicamos em “choose”, expanda o menu "lazy" e selecione "IBK" e “OK”. Nos parametros do IBK iremos colocar o “KNN” = 3 e “OK” no painel "Test Options", escolha a opção "Cross-validation" e configure o número de folds como 10 no campo correspondente. Com tudo configurado, clique em "Start" para iniciar a classificação.

Bagging MLP: Com o arquivo aberto no Weka na aba "Classify", onde estão disponíveis os algoritmos de aprendizado de máquina. No campo "Classifier", clique em "Choose", expanda o menu "Meta" e selecione "Bagging". Nos parâmetros do Bagging em “classifier” clicamos em “choose”, expanda o menu "functions" e selecione "MultilayerPerceptron" e “OK”. Nos parâmetros do MLP utilizamos no campo “hiddenLayers” colocamos 3 camadas com 10 neurônios cada (10,10,10), “nominalToBinaryFilter” = True, “normalizeAttributes” = True e “normalizeNumericClass” = True e clicamos em “OK”. No painel "Test Options", escolha a opção "Cross-validation" e configure o número de folds como 10 no campo correspondente. Com tudo configurado, clique em "Start" para iniciar a classificação.

Ensembles Vote

Um Voting Classifier é uma técnica de aprendizado de máquina de conjunto que combina as previsões de vários classificadores individuais (também conhecidos como classificadores base ou estimadores) para fazer uma previsão final. É um tipo de abordagem de média de modelo em que cada classificador base contribui com sua previsão, e a previsão final é determinada por um voto majoritário (para classificação) ou uma média (para regressão). O Voting Classifier pode ser usado para tarefas de classificação binária e multiclasse.

<Referencia: https://medium.com/@ranjankumar\_29097/votingclassifier-3f85ba8e4580 >

Passo a Passo

Ensembles Vote: Com o arquivo aberto no Weka Na aba “Classify” opção “Classifier” escolhemos a opção “Meta” e selecionamos “Vote”. Nos parâmetros do Ensembles Vote, no campo “classifiers” irá abrir uma janela para adicionarmos os classificadores. Já vem com o classificador “ZeroR” na opção “Delete” apagamos o “ZeroR” então na opção “choose” escolhemos “classifiers”, “bayes” e “NaiveBayes”, opção “Add”, novamente clicamos “choose” escolhemos “classifiers”, “lazy” e “IBK” clicamos nos parâmetros do IBK e colocamos KNN = 3 e clicamos em “OK” depois em “Add”, novamente clicamos “choose” escolhemos “classifiers”, “functions” e “MultilayerPerceptron” clicamos nos parâmetros do MLP colocamos 3 camadas com 10 neurônios cada (10,10,10), “nominalToBinaryFilter” = True, “normalizeAttributes” = True e “normalizeNumericClass” = True e clicamos em “OK” depois em “Add”. Após adicionar os três classificadores NaiveBayes, IBK (KNN) e MultilayerPerceptron fechamos a janela e na opção “combinationRule” selecionamos “Majority Voting”. Utilizamos “Cross-validation Folds” = 10 e Start.

Lembrar sempre de descrever se o modelo está normalizando ou não, se houve ajustes e quais foram feitos, citando os campos dos ajustes (para cada execução do classificador).

**Resultados de Discussão**

Aqui a proposta é apresentar as tabelas resultantes das matrizes de confusão, criar uma tabela, comparativa com todos os modelos executados, mostrando a eficiência de cada modelo, para apontar qual é os modelos que mais performou. Pode-se criar um gráfico, para deixar visual a diferença entre esses – para essa tarefa podemos utilizar o F-Score.

Nesse ponto, é interessante fazer as análises dos dados apresentados e, baseados nos dados, qual o melhor resultado e por que esse resultado.

**Conclusão**

**Referências**

https://www.mdsaude.com/exames-complementares/valor-de-referencia/

Bourbon, Mafalda, et al. Doenças Cardiovasculares. fevereiro de 2016, p. 1–24.

Chicco, Davide, e Giuseppe Jurman. Heart Failure Clinical Records. UCI Machine Learning Repository, 2020. DOI.org (Datacite), https://doi.org/10.24432/C5Z89R.

Chicco, D., & Jurman, G. (2020). *Heart Failure Clinical Records*.

Chicco, Davide, e Giuseppe Jurman. “Machine Learning Can Predict Survival of Patients with Heart Failure from Serum Creatinine and Ejection Fraction Alone”. BMC Medical Informatics and Decision Making, vol. 20, n°1, dezembro de 2020, p. 16. DOI.org (Crossref), https://doi.org/10.1186/s12911-020-1023-5.

OPAS/OMS. “Doenças cardiovasculares - OPAS/OMS | Organização Pan-Americana da Saúde”. OPAS - Organização Pan-Americana de Saúde, 31 de janeiro de 2024, https://www.paho.org/pt/topicos/doencas-cardiovasculares.

Souza, Alex. “Knowledge Discovery in Databases (KDD)”. Blog Do Zouza, 26 de julho de 2023, https://medium.com/blog-do-zouza/knowledge-discovery-in-databases-kdd462ea2775715.

**Apêndices**