

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

1st Fabio Andrade
ex18245
Aluno
Faculdade de Tecnologia
UNICAMP
São Paulo – SP, Brasil
fabio_andrades@outlook.com

2nd Gismar Barbosa
ex182527
Aluno
Faculdade de Tecnologia
UNICAMP
Limeira – SP, Brasil
gismar_barbosa@yahoo.com.br

3rd João Amazonas
ex182429
Aluno
Faculdade de Tecnologia
UNICAMP
Maracanaú – CE, Brasil
amazonaz@hotmail.com

4th Laís Gregório
ex182382
Aluno
Faculdade de Tecnologia
UNICAMP
Araçatuba – SP, Brasil
lahgregorii@gmail.com

5th Thainnara Lima
ex182385
Aluno
Faculdade de Tecnologia
UNICAMP
Campinas – SP, Brasil
thainnara8@gmail.com

Resumo — Esse relatório descreve o estudo e aplicação de técnicas de aprendizado de máquina supervisionado, mais especificamente, classificação de dados. Foram aplicadas algumas técnicas de KDD, juntamente com o uso dos classificadores “Naïve Bayes”, “k-NN”, “MLP” e, “Ensemble Bagging” e “Vote”. O objetivo central é desenhar um modelo de predição de dados para prever mortes por doenças cardíacas. Ao final, por meio dos resultados obtidos, é possível verificar que o “Ensemble Bagging” de “MLP” é o modelo com melhor performance para prever mortes por doenças cardíacas.

Keywords — Nive Bayes, k-NN, MLP, Ensemble, Bagging, Vote, dataset, aprendizado de máquina, KDD, doenças cardíacas.

I. INTRODUÇÃO

Doenças cardiovasculares afetam o sistema circulatório e são a principal causa de mortalidade global, com 17,9 milhões de mortes em 2016 [1]. 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 [2].

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 [3], contendo os dados de 299 pacientes com 12 características clínicas, este artigo, aplicando técnicas de KDD [4] e, ferramentas de aprendizado de máquina (ML), tem como objetivo principal, analisar a eficiência, dos métodos supervisionados de ML (classificação de dados) para prever morte por insuficiência cardíaca a partir dos dados clínicos deste dataset. Essa abordagem, em uma aplicação do mundo real, poderia aprimorar a detecção precoce e, gestão das doenças cardiovasculares, sendo uma ferramenta estratégica para diminuição da mortalidade.

II. METODOLOGIA

A. Dados

Os dados, originais (dataset), podem ser visualizados a partir a partir do repositório “Heart Failure Clinical Records” [3], sendo possível baixá-los no formato de arquivo CSV. Esse estudo acompanhou pacientes, com problemas cardíacos (e outros indicadores de saúde), durante determinado tempo em que, durante o período do estudo, alguns pacientes vieram a

óbito por conta do problema cardíaco. Um resumo da descrição dos dados pode ser encontrado na TABELA I. É importante ressaltar que esses dados já foram coletados e, receberam tratamento prévio, antes mesmos de serem disponibilizados neste dataset. Esse processo de tratamento foi realizado pelo instituto e autores do estudo. Essa questão nos leva a inferir, por relação causa e efeito, que os dados são fatores importantes para o domínio estudados (saúde – fatores de risco para doenças cardiovasculares).

TABELA I. METADADOS DO DATASET

Nome do Atributo	Tipo	Descrição	Unidade de Medida
age	Inteiro	idade do paciente	Anos
anaemia	Binário	diminuição de glóbulos vermelhos ou hemoglobina	
creatininephosphokinase	Inteiro	nível da enzima 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	Contínuo	plaquetas no sangue	kiloplatelets/mL
sérum_creatinine	Contínuo	nível de sódio no sangue	mmEq/L
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, se

encontram disponíveis, no repositório do Github “heart_attack_machine_learning” [5], na seção “DATASET”.

B. Weka

O Waikato Environment for Knowledge Analysis (Weka) [6], 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.

C. 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 dos 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:

- No Weka, na tela “Weka GUI Chooser” clique sobre o menu “Explorer” e, após isso será aberta a tela “Weka Explorer”. Nessa nova tela, clique sobre o botão “Open file...” e, localize e carregue o arquivo “heart_failure_clinical_records_dataset.csv”. Se atente para alterar o campo “Files of Type” para “CSV data files (*.csv)”, caso não esteja localizando o arquivo CSV. Após o arquivo carregado, clique no botão “Save”, e salve o arquivo com o campo “File of Type” em “Arff data files (*.arff)”.
- Utilizando um editor de texto puro (Notepad, Nano, VIM etc.) abra o arquivo que foi salvo. Localize a linha contendo a seguinte descrição “@attribute DETH_EVENT numeric” e altere o “numeric” para “{0,1}”, ficando a descrição final “@attribute DETH_EVENT {0,1}”. Esse processo, final, irá garantir que o Weka entenda esse campo, como a classe a ser predita.
- Agora, o arquivo poderá ser aberto novamente no Weka, onde a ferramenta irá reconhecer a classe (de estudo), evitando erros de análises dos dados.

D. Classificador 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 [7].

- Passo à passo: 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.

E. Classificador k-NN

O k-NN (k-Nearest Neighbors) [8], 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 à passo: 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.

F. Classificador MLP

O perceptron multicamadas (MLP) [9] é uma rede neural semelhante ao perceptron simples, porém possui mais de uma camada de neurônios. 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.

- Passo à passo: Com o arquivo aberto no Weka Na aba “Classify” opção “Classifier” escolhemos a opção “Functions” e selecionamos “MultilayerPerceptron”. 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”. 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.

G. Classificador Ensemble do tipo 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 [10], se destaca como um método ensemble popular e amplamente implementado.

- Passo à passo Bagging Naïve Bayes: 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.
- Passo à passo 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 parâmetros 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.
- Passo à passo 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.

H. Classificador Ensemble do tipo Vote

Um Voting Classifier [11] 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 multi classe.

- Passo à passo Ensemble Vote: 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.

III. RESULTADOS E DISCUSSÃO

Os resultados obtidos dos experimentos se encontram abaixo. Ressaltamos que os resultados refletem os métodos experimentais realizados e, o tratamento de dados após os experimentos foram realizados apenas para demonstrar os resultados, de forma coerente e conclusiva.

Nossos experimentos submeteram o referido dataset (com 299 amostras) ao processo de classificação de 3 classificadores e seus referidos Ensembles (Bagging e Vote). Abaixo, temos as matrizes de confusão de cada modelo (ver em TABELA II. , TABELA III. , TABELA IV. , TABELA V. , TABELA VI. , TABELA VII. , TABELA VIII.). Nas referidas matrizes, é possível os acertos e erros (positivos e negativos), assim como a métrica de acurácia [12], dada pela fórmula:

$$\frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Os dados referentes ao estudo, de onde são retirados os resultados, assim como as matrizes de confusão abaixo, podem ser encontrados no repositório do Github "heart attack machine_learning" [5], na seção "CLASSIFIERS".

TABELA II. MATRIZ DE CONFUSÃO NAÏVE BAYES

a	b	← classified as
25	71	a = 1
14	189	b = 0
Acurácia		72%

TABELA III. MATRIZ DE CONFUSÃO K-NN

a	b	← classified as
29	67	a = 1
19	184	b = 0
Acurácia		71%

TABELA IV. MATRIZ DE CONFUSÃO MLP

a	b	← classified as
61	35	a = 1
31	172	b = 0
Acurácia		78%

TABELA V. MATRIZ DE CONFUSÃO ENSEMBLE BAGGING NAÏVE BAYES

a	b	← classified as
45	51	a = 1
18	185	b = 0
Acurácia		77%

TABELA VI. MATRIZ DE CONFUSÃO ENSEMBLE BAGGING K-NN

a	b	← classified as
25	71	a = 1
14	189	b = 0
Acurácia		72%

TABELA VII. MATRIZ DE CONFUSÃO ENSEMBLE BAGGING MLP

a	b	← classified as
58	38	a = 1
25	178	b = 0
Acurácia		79%

TABELA VIII. MATRIZ DE CONFUSÃO ENSEMBLE VOTE

a	b	← classified as
51	45	a = 1
20	183	b = 0
Acurácia		78%

Outro número importante, é a relação erro / acerto de cada modelo quando aplicado aos dados. Abaixo podemos analisar esses indicadores, de maneira comparativa de forma tabular (ver em TABELA IX.) e, gráfica (ver em Fig. 1):

TABELA IX. RELAÇÃO ERRO E ACERTOS POR CLASSIFICADOR

Classificador	Acerto (%)	Erros (%)
Naïve Bayes	77,59	22,41
k-NN	71,24	28,76
MLP	77,93	22,07
E.B. Naïve Bayes	76,92	23,08
E.B. k-NN	71,57	28,43
E.B. MLP	78,93	21,07
E.V. 3 classificadores	78,26	21,74

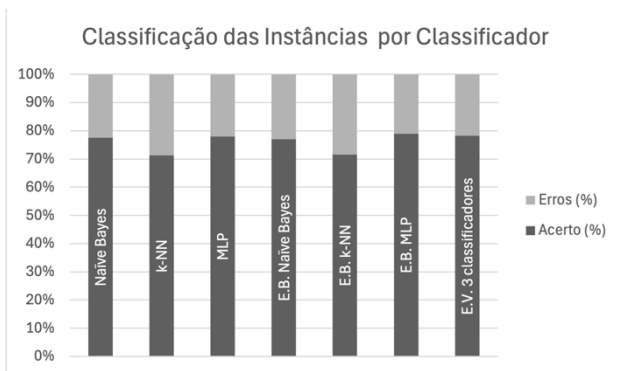


Fig. 1. Gráfico de Classificação das instâncias por classificador

E, por fim, aplicando as métricas de performance “F-score” [12]:

$$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2)$$

Que, diretamente, combina (equilibra) as métricas de “Precision” [12], que indica a exatidão das previsões positivas, designado pela equação:

$$\frac{TP}{TP + FP} \quad (3)$$

E “Recall” [12], que indica o quão bem o modelo consegue identificar os casos positivos reais, designado pela equação:

$$\frac{TP}{TP + FN} \quad (4)$$

Podemos analisar a mesma comparação anterior (de forma tabular e gráfica), para essas métricas abaixo (ver em TABELA X. , e Fig. 2):

TABELA X. MÉTRICAS POR CLASSIFICADOR

Classificador	F-score	Precision	Recall
Naïve Bayes	0,761	0,770	0,776
k-NN	0,680	0,692	0,712
MLP	0,778	0,777	0,779
E.B. Naïve Bayes	0,754	0,762	0,769
E.B. k-NN	0,673	0,699	0,716
E.B. MLP	0,785	0,784	0,789
E.V. 3 classificadores	0,773	0,776	0,783

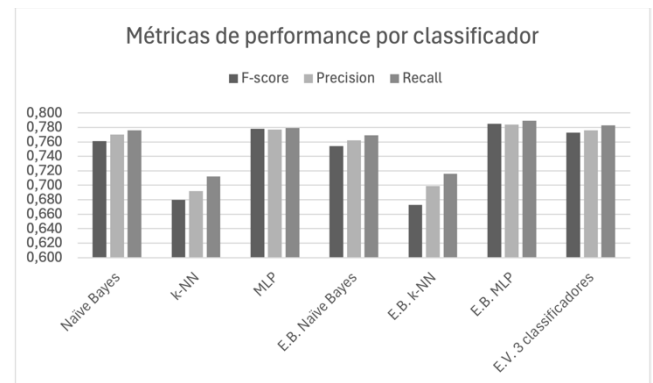


Fig. 2. Gráfico de Metas de Performance por Classificador

Tais dados e, suas respectivas metodologias amostrais, finalizam essa seção demonstrativa.

IV. CONCLUSÃO

Ao final deste relatório, voltamos ao início do objetivo deste projeto que, resumidamente era encontrar um modelo de predição (classificador) que fosse capaz de prever, de forma eficiente, eventos de morte ou não por doença cardíaca. a partir do treinamento dos classificadores com os dados presentes no dataset [3] (fruto de estudo deste projeto).

Desta forma, após realizarmos o processamento dos dados, fazendo alguns ajustes e, quando necessário, o pré-processamento dos dados, submetemos esses dados aos classificadores “Naïve Bayes” [7]; “k-NN” [8]; MLP [9]; Ensembles Bagging [10] e Vote [11], para assim, como esperado, encontrar o melhor modelos aplicável para resolver o problema deste projeto: prever mortes ou não.

Após esses processos, foi possível, dentre os classificadores, percebermos que, na relação direta, de classificação das instâncias de dados, o classificador “Ensemble Bagging” de “MLP”, dentro dos parâmetros utilizados neste projeto, foi o que teve o melhor resultados (78,9%) de acertos. E, da mesma forma, seguindo a mesma metodologia, o classificador “k-NN”, teve a pior performance (71,2%). Mas, para análise assertiva dos dados dos classificadores, assim como podemos analisar as matrizes de confusão (presentes neste relatório), não podemos simplesmente atribuir melhor ou pior, apenas nessa relação já que, no domínio da área de estudo (saúde) um “falso positivo” e um “falso negativo” podem ter valores, reais, totalmente diferentes: errar a previsão de que uma pessoa não morrerá, poderia ser pior que errar que ela morrerá dado que, se estamos atuando com predições, uma pessoa com sentença de morte poderia, ainda, tentar reverta o quadro (o esperado). Mas, uma pessoa com uma, falsa, previsão de não morte, nada iria fazer para rever quadro e, possivelmente iria morrer. Com esse exemplo, podemos entender a necessidade e importância das matrizes de erro nesse projeto.

E, adicionando ao que já foi dito, para diminuir ao máximo o erro nas previsões, utilizamos as principais métricas de performance para os modelos de predição (classificadores): “F-score”, “Precision” e “Recall”. Aplicando essas métricas, visando, de fato, inferir o melhor modelo para predição de dados, neste estudo, temos o “Ensemble Bagging” de “MLP” como sendo o mais performático, onde ele fica mais próximo de 1 (ficando com a pontuação de 0,785).

E, finalizando essa conclusão, apesar encontrarmos um modelo de classificador para predição de eventos de morte que, consegue ter uma pontuação próxima de 80% de acerto (como demonstrado, o Ensemble Bagging de MLP), não podemos dizer que esse resultado satisfaz os objetivos. Assim como já citado anteriormente, o domínio de aplicação da saúde não pode trabalhar com margens de erros grandes como essa (acima de 20%).

Talvez, levando em consideração o referido cenário, seria necessário, em um próximo projeto, voltar aos dados e, coletar mais amostras para esse dataset, assim como, incluir possíveis novas variáveis (atributos) – sem descartar os atributos já presentes.

REFERÊNCIAS

- [1] OPAS, “Doenças cardiovasculares,” OPAS/OMS Organização Pan-Americana de Saúde, 17 Maio 2023. [Online]. Disponível: <https://www.paho.org/pt/topicos/doencas-cardiovasculares>. [Acesso em 10/12/2024].
- [2] D. B. Précoma, G. M. Moraes de Oliveira,, A. F. Simão e O. P. Dutra, “Pocket SBC - Atualização da Diretriz de Prevenção Cardiovascular da Sociedade Brasileira de Cardiologia - 2019,” 2019. [Online]. Disponível: <http://www.cardiometro.com.br/include-cardiometro/teste2/pdf/POCKET%20DIRETRIZ%20DE%20PREV ENCAO.pdf>. [Acesso em 09 de Dezembro de 2024].
- [3] D. Chicco and G. Jurman, "Heart Failure Clinical Records," 4 Fevereiro 2020. [Online]. Disponível:

<https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records>. [Acesso em 10 de Dezembro de 2024].

- [4] A. Souza, “Knowledge Discovery in Databases (KDD),” 25 Junho 2023. [Online]. Disponível: <https://medium.com/blog-douzouza/knowledge-discovery-in-databases-kdd-462ea2775715>. [Acesso em 01/12/2024].
- [5] F. Andrade, G. Barbosa, J. Amazonas, G. Laís e T. Santos, “heart_attack_machine_learning,” 9 Dezembro 2024. [Online]. Disponível: https://github.com/gismarb/heart_attack_machine_learning/tree/main. [Acesso em 14 Dezembro de 2024].
- [6] Waikato University, "The Weka Workbench," Waikato University, 2024. [Online]. Disponível: <https://ml.cms.waikato.ac.nz/weka>. [Acesso em 20 de Novembro de 2024].
- [7] Wikipedia, "Naive Bayes classifier," Wikipedia The Free Encyclopedia, 28 Novembro 2024. [Online]. Disponível: https://en.wikipedia.org/wiki/Naive_Bayes_classifier. [Acesso em 29 11 2024].
- [8] A. Mucherino e P. Papajorgji, “k-Nearest Neighbor Classification,” *Springer Nature Link*, vol. 34, pp. 83-106 , 2009.
- [9] Wikipedia, “Perceptron multicamadas,” Wikipedia A enciclopédia livre, 18 Agosto 2022. [Online]. Disponível: https://pt.wikipedia.org/wiki/Perceptron_multicamadas. [Acesso em 9 Dezembro 2024].
- [10] A. A. Awan, "What is Bagging in Machine Learning? A Guide With Examples," Datacamp, 20 Novembro 2023. [Online]. Disponível: <https://www.datacamp.com/tutorial/what-bagging-in-machine-learning-a-guide-with-examples>. [Acesso em 09 de Dezembro 2024].
- [11] R. Kumar, "VotingClassifier," Medium, 12 Agosto 2023. [Online]. Disponível: https://medium.com/@ranjankumar_29097/votingclassifier-3f85ba8e4580. [Acesso em 14 de Dezembro 2024].
- [12] GeeksforGeeks, "F1 Score in Machine Learning," GeeksforGeeks, 17 Outubro 2024. [Online]. Disponível: <https://www.geeksforgeeks.org/f1-score-in-machine-learning/>. [Acesso em 12 de Dezembro de 2024].

APÊNDICE (AVALIAÇÃO POR PARES)

O formato escolhido para realização do trabalho, foi feito em encontros semanais (2 encontros por semana), onde, nesses encontros, fizemos os testes experimentais com as amostras de dados (dataset). Nesses encontros, realizamos o debate sobre os processos aplicados (KDD e classificadores), assim como os resultados obtidos. No encontro final, cada membro do grupo compartilhou com suas análises, onde, após as discussões, chegamos ao entendimento final para, assim, de fato, escrevermos o relatório do projeto.

Desta forma, a nota para todos do grupo (Fábio Andrade, Gismar Barbosa, João Amazonas, Laís Gregório e Thainnara Santos) é 10.

O resultado foi dado, em comum acordo, após todos refletirem sobre sua "jornada de conhecimento", durante o processo de construção do trabalho.