PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

ENG 4502

Ciência de Dados

PLANO DE EXPERIMENTAÇÃO: CLASSIFICAÇÃO

2210095 Juliana Sauerbronn

2210246 Luana Hamond

2210875 Luísa Silveira

Professora:

Fernanda Amorim

Rio de Janeiro

Maio 2024

LISTA DE TABELAS

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

Neste trabalho, usaremos duas técnicas de machine learning para classificar doenças cardíacas, a árvore de decisão e o KNN, dois algoritmos dentro da problemática da Classificação. A Classificação visa prever a classe ou a categoria a que uma nova observação pertence com base em um conjunto de dados de treinamento.

Árvore de decisão é um modelo de aprendizado supervisionado que modela decisões e suas possíveis consequências, ao formar uma estrutura de árvore. Para começar, precisamos definir qual será o melhor atributo para dividir os dados, baseado no ganho de informação. Sobre nosso plano de experimentação, criamos 223 cenários para serem executados e dividimos em três critérios para definir os melhores atributos, ‘gini’, ‘entropy’ e ‘log_loss’.

Já o KNN, que também é um algoritmo de aprendizado supervisionado, classifica um ponto de dados baseado na classe majoritária de seus k vizinhos mais próximos. Para executá-lo, precisamos definir o número de vizinhos (k) a serem considerados e a métrica de distância que será usada. A respeito do plano de experimentação, criamos 107 cenários para serem executados, sendo variado de três a seis vizinhos dependendo de cada cenário. Além disso, para cada cenário, usaremos as três métricas de distância e definiremos qual será a melhor, podendo ser ‘minkowski’, ‘euclidean’ ou ‘manhattan’.

2. PLANEJAMENTO DA EXPERIMENTAÇÃO

O planejamento de experimentação consiste na organização e definição de uma variedade de experimentos a serem realizados visando verificar hipóteses. Assim, são selecionadas diversas variáveis a serem variadas, visando a definição de condições diferentes a cada execução realizada (PRADO, 2020). Por fim, são definidas as métricas que serão utilizadas para avaliar os resultados do experimento.

No contexto de ciência de dados, esse planejamento é fundamental para avaliar quais as técnicas de pré-processamento e quais valores dos parâmetros resultaram em um maior desempenho do modelo. Por exemplo, pode-se descobrir se há alguma variável de entrada que está reduzindo o desempenho. Desse modo, há uma melhoria na compreensão do fenômeno estudado, ao verificar quais fatores impactam mais o modelo e ao perceber qual foi o melhor cenário de execução. Ademais, o plano de experimentação contribui com a diminuição de viés, visto que são variadas diversas variáveis, reduzindo a chance de se haver resultados parciais (BUTTON, 2012). Além disso, a planilha do plano de experimentação realizado se encontra nos anexos.

2.1. Pré-processamento

Para realizar um experimento de classificação de doenças cardíacas, considerando a resposta zero como não possui doença e um como possui, é necessário selecionar quais variáveis do problema irão variar, além de definir os tipos de pré-processamento que serão realizados. Para esse problema, serão escolhidas de onze a treze colunas aleatoriamente por execução, visando verificar quais possuem maior impacto no modelo. Além disso, poderão ser realizadas dois tipos de normalizações: utilizando MinMaxScaler e StandardScaler.

2.1.1. MinMaxScaler

O MinMaxScaler consiste em um modo de pré-processamento que transforma os valores da coluna selecionada em valores entre zero e um. Assim, o maior número da coluna selecionada terá um valor um e o menor será zero (KUMAR, 2023). A vantagem dessa forma de realizar a padronização é que, em geral, não há a centralização dos dados ao redor de uma média e que coloca todas as informações de uma *feature* na mesma escala (DUARTE, 2020). Além disso, o MinMaxScaler possui uma sensibilidade maior aos ruídos e a *outliers*, visto que a escala é influenciada diretamente pelos maiores e menores valores dos dados. Geralmente, há o uso desse modo de padronizar em algoritmos que possuem a distância como base, por exemplo, ao utilizar o KNN, ou em redes neurais (KUMAR, 2023).

2.1.2. StandardScaler

Já o StandardScaler consiste em uma técnica de padronização que transforma os dados para que tenham média zero e variância unitária. Isso significa que ele ajusta os dados de forma que a média de cada característica seja zero e o desvio padrão seja um. Além disso, ele é menos sensível a *outliers*, uma vez que há um foco no desvio padrão. Usualmente, há o uso do StandardScaler em modelos de regressão linear, por exemplo (KUMAR, 2023).

2.2. Mineração de dados

2.2.1. K-Nearest Neighbors

Um algoritmo que será utilizado na experimentação de classificação será o K-Nearest Neighbors (KNN), que consiste em um método de aprendizado supervisionado que prevê o valor de um ponto, usando como referência o dos k valores mais próximos dele (JOSÉ, 2018). Para implementar esse algoritmo, será utilizado o KNeighborsClassifier da biblioteca sklearn. Em relação aos parâmetros selecionados, serão variados: a métrica de distância, o peso e o valor de n_neighbors.

2.2.1.1. Métricas de distância

No (KNN), há o uso das métricas de distância para avaliar a proximidade entre pontos distintos. Essa escolha de métrica a ser utilizada é importante para escolher quais valores serão determinados como semelhantes devido a sua proximidade (JOSÉ, 2018). Logo, determina quais pontos podem ser utilizados para realizar uma previsão. Visando aumentar o desempenho do modelo, é necessário testar diversas métricas de distância para verificar qual consegue ajudar a encontrar a resposta mais adequada para o problema. Nesse sentido, a distância utilizada no KNN irá variar nesse experimento entre euclidiana, minkowski e manhattan.

A distância euclidiana consiste em calcular a raiz quadrada da soma das diferenças quadráticas entre os eixos de dois pontos. Assim, o cálculo resulta em linha reta entre os dois pontos no espaço n-dimensional (IBM, 2023). Geralmente, essa distância é usada para conjunto de dados contínuos, quando as características dos pontos estão em escalas comparáveis.

Já a distância de manhattan é calculada a partir da soma do módulo da diferença dos valores (INÁCIO, 2021). Essa métrica de distância é comumente empregada em situações em que os dados seguem padrões lineares ou quando a variação no peso das características é menos importante do que a direção e a proximidade.

Por fim, a distância de minkowski a partir da distância entre dois pontos, levando em consideração a diferença absoluta entre as coordenadas nos diferentes eixos, elevada a uma potência. Quando o valor dessa potência é equivalente a um, será igual à distância de manhattan. Por outro lado, se for igual a dois, será igual a distância euclidiana (INÁCIO, 2021).

2.2.1.2. Peso

Ao utilizar o (KNN), também é possível escolher o peso. Essa escolha de peso a ser utilizado é importante, pois afeta a contribuição de cada ponto próximo para a previsão. Diversos pesos devem ser testados para melhorar o desempenho do modelo e encontrar a solução mais apropriada para o problema em questão. Assim, no experimento, os pesos usados no KNN irão variar entre uniforme e baseado na distância.

Ao se usar o uniforme, todos os valores da proximidade terão a mesma influência ao realizar a previsão da doença cardíaca. Por outro lado, ao usar o peso baseado na distância, os vizinhos que possuem uma proximidade maior contribuem mais para o valor final da previsão do que os mais distantes (SCIKIT-LEARN, 2024).

2.2.1.3. Valor de n_neighbors

O valor de n_neighbors ao utilizar o algoritmo KNN consiste no número de pontos próximos que deve ser considerado ao realizar uma previsão (JOSÉ, 2018). Para a experimentação a ser realizada, o valor irá variar entre três e seis, o que permitirá verificar a diferença entre diversas execuções e perceber o impacto desse parâmetro na previsão e no desempenho do modelo.

2.2.2. Árvore de decisão

Outro algoritmo que será usado na experimentação de classificação será a árvore de decisão, que consiste em um método de aprendizado supervisionado que prevê o valor de um ponto, usando divisões iterativas baseado em uma medida de pureza. Esse algoritmo reparte o conjunto de dados em subconjuntos menores cada vez mais semelhantes entre si, criando uma estrutura parecida com uma árvore (SACRAMENTO, 2024). Objetivando implementar esse algoritmo, será utilizado o DecisionTreeClassifier da biblioteca sklearn. Os parâmetros escolhidos para realizar a variação serão:max_features, min_samples_leaf e criterion.

2.2.2.1. Valor de max_features

O parâmetro max_features consiste no número máximo de features que serão levadas em consideração ao realizar uma repartição de nó. Enquanto os valores inferiores tem tendência de reduzir a variabilidade do modelo, os maiores podem elevar o risco de overfitting, já que capturam mais detalhes dos dados do problema (SCIKIT-LEARN, 2024). Para o experimento, esse parâmetro irá variar entre “sqrt”, *None* e “log2”.

2.2.2.2. Valor de min_samples_leaf

O parâmetro min_samples_leaf consiste no número mínimo de amostras necessárias ao realizar uma divisão de um nó (SCIKIT-LEARN, 2024). Se o valor selecionado para esse parâmetro, por exemplo, for de um, é possível montar uma árvore detalhada. Durante a realização do experimento, será considerado que esse parâmetro irá variar entre um a três.

2.2.2.3. Criterion

O parâmetro *criterion* consiste no critério a ser usado como medida de qualidade ao se repartir um nó. Para esse experimento, serão usados dois tipos de critérios diferentes: entropy e gini. Enquanto o gini irá medir o quão puro um nó é, dividindo os nós até possuir apenas uma classe em um subconjunto, a entropia mede a quantidade de incerteza dos dados. Quanto menor entropia, mais puras são as amostras (CÂNDIDO, 2023).

2.3. Pós-processamento

Em seguida, é importante verificar quais serão as métricas de avaliação selecionadas. Para esse problema de classificação, as métricas escolhidas foram a acurácia, a precisão, o recall e a medida F1. Também será realizada um k-fold cross validation, variando k (k igual a 3 ou k igual a 7), e serão verificados os valores de verdadeiros positivos, verdadeiros negativos, falsos negativos e falsos positivos da matriz de confusão.

3. EXECUÇÃO DO EXPERIMENTO

Visando executar os algoritmos de KNN e de árvore de decisão, foi feito um código no Google Collab que verifica diversos cenários considerando os parâmetros dados. Utilizando bibliotecas como pandas, numpy, scikit-learn e matplotlib, o código carregou e pré-processou (usando MinMaxScaler e StandardScaler) os dados sobre doenças cardíacas. Após o pré-processamento, os dados foram divididos em dados de treino e de teste, sendo 75% para o treinamento.

Em seguida, usando um loop ‘for’, que possuía outros loops dentro dele, foi feito um código que criava cada cenário, utilizando parâmetros com valores diferentes em cada um. Assim, rodando o código, os resultados obtidos foram organizados em uma planilha, documentando cada cenário executado. Estes dados foram então transferidos para um arquivo de planejamento de experimento, facilitando a análise e comparação das diferentes abordagens. Por fim, foram feitos gráficos para identificar as melhores configurações para a aplicação dos algoritmos na classificação de doenças cardíacas.

4. ANÁLISE DA EXECUÇÃO DO EXPERIMENTO

4.1. KNN

Foram executados 107 cenários utilizando o KNN, modificando o valor dos parâmetros e o modo de realizar a padronização. Objetivando verificar quais parâmetros foram os que mais afetaram o desempenho do modelo, foram feitas diversas análises utilizando gráficos como base.

Na Figura 1 abaixo, é possível verificar a diferença entre a precisão média ao se padronizar com o MinMaxScaler e ao se usar o StandardScaler. É observável que o MinMaxScaler, numa média, possui uma maior precisão para os cenários testados, o que poderia melhorar o desempenho do modelo. Logo, para o KNN, escolher o MinMaxScaler na padronização poderia melhorar o modelo do aprendizado de máquina de classificação de doenças cardíacas.

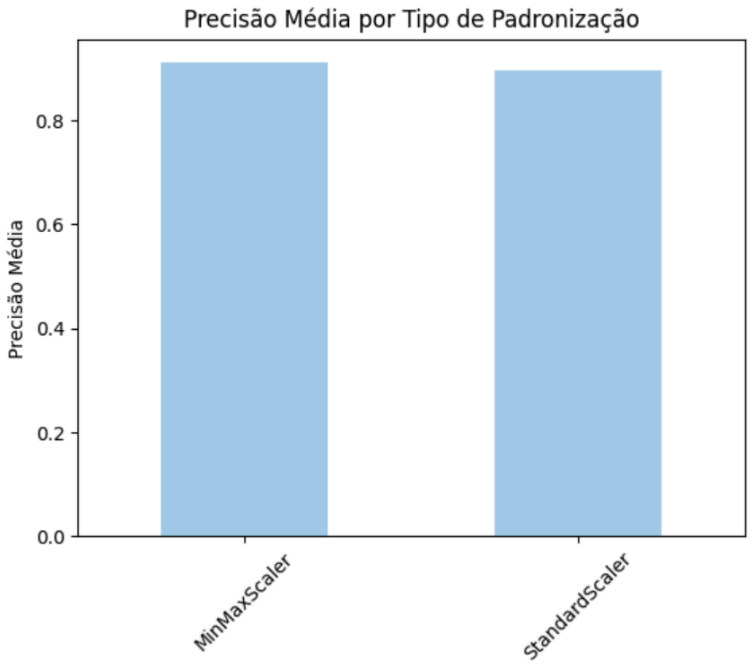

Figura 1 - Comparação da precisão das padronizações pro KNN

Na Figura 2 abaixo, podemos perceber que a acurácia do modelo também varia ao se padronizar com o MinMaxScaler ou ao se usar o StandardScaler. É notável que o MinMaxScaler, numa geral, têm uma maior acurácia média para os

cenários testados. Portanto, ao optar pelo MinMaxScaler na normalização, o desempenho do modelo de classificação de doenças cardíacas usando KNN pode ser aprimorado.

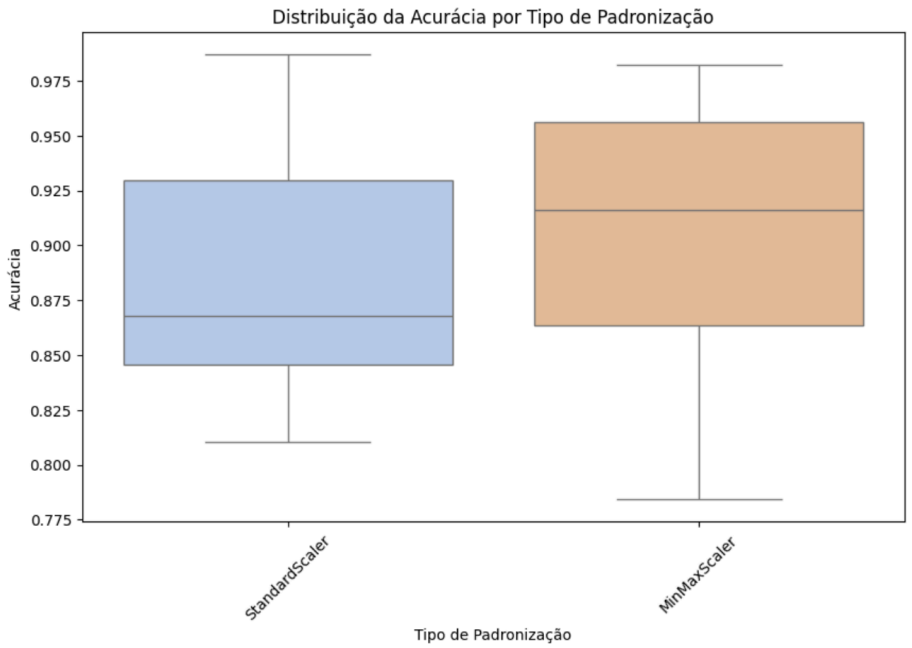

Figura 2 - Comparação da acurácia das padronizações pro KNN

É possível notar que nos cenários onde o tipo de métrica utilizada foi o minkowski, pois a média das suas precisões e acurácias foram as maiores na Figura 3 e Figura 4. Ao analisar, pode-se observar que, mesmo o quantil 2 da métrica manhattan ser maior, tanto o limite superior quanto o inferior da métrica minkowski são maiores, mostrando um resultado melhor na média dos valores de acurácia resultantes da métrica minkowski. Na comparação com a euclidean, no gráfico vemos que o limite superior e inferior, a mediana e o quantil 1 apresentam valores de acurácia inferior, resultando num melhor desempenho da métrica minkowski.

Já no gráfico de precisão, o limite inferior, o quantil 3 e o quantil 1 da métrica minkowski apresentam precisão maior, sendo essa métrica a de melhor desempenho na maioria dos cenários.

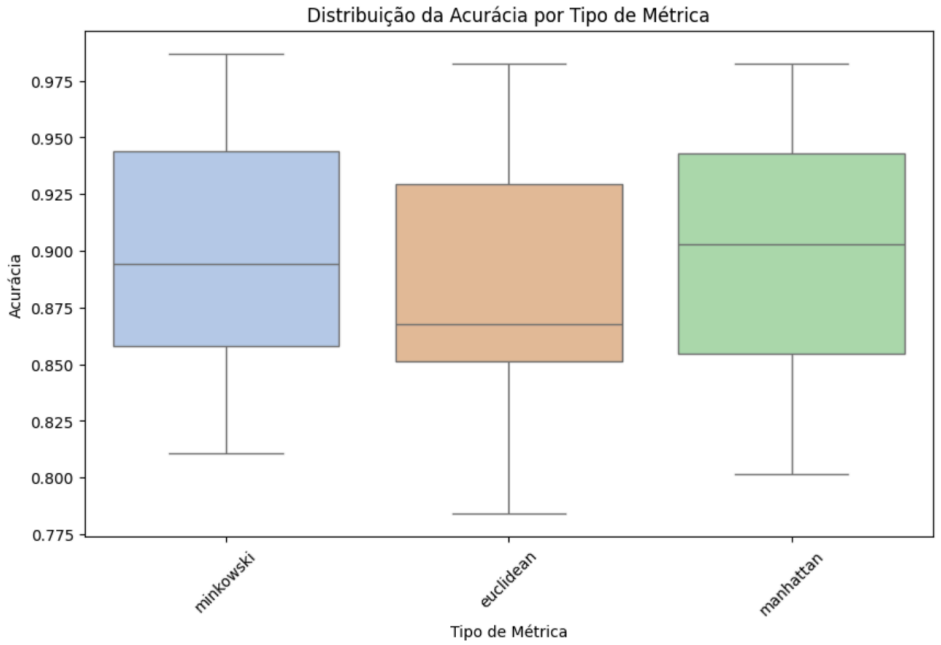

Figura 3 - Comparação da acurácia das distâncias pro KNN

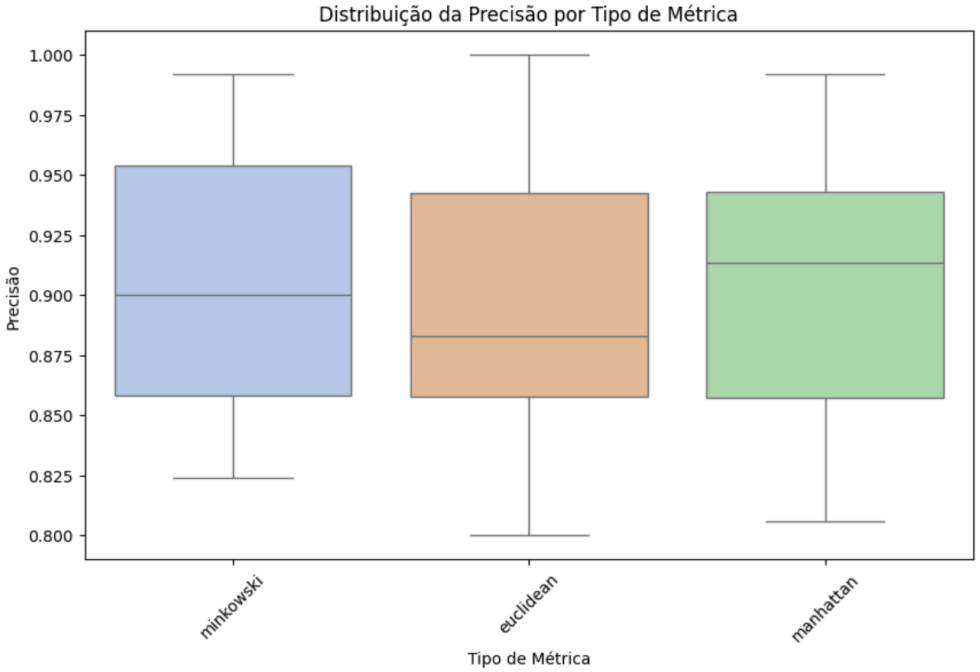

Figura 4 - Comparação da precisão das distâncias pro KNN

Pela Figura 5 abaixo, é observável que quando k é igual a sete no K-fold Cross Validation, a acurácia tende a ser maior do que nos casos que k é igual a três. Isso ocorre, pois um número superior de folds pode ocasionar uma redução de viés,

captando padrões mais sutis nos dados e, consequentemente, melhorando a acurácia. Assim, o modelo tende a ter um desempenho melhor com k do cross validation igual a sete.

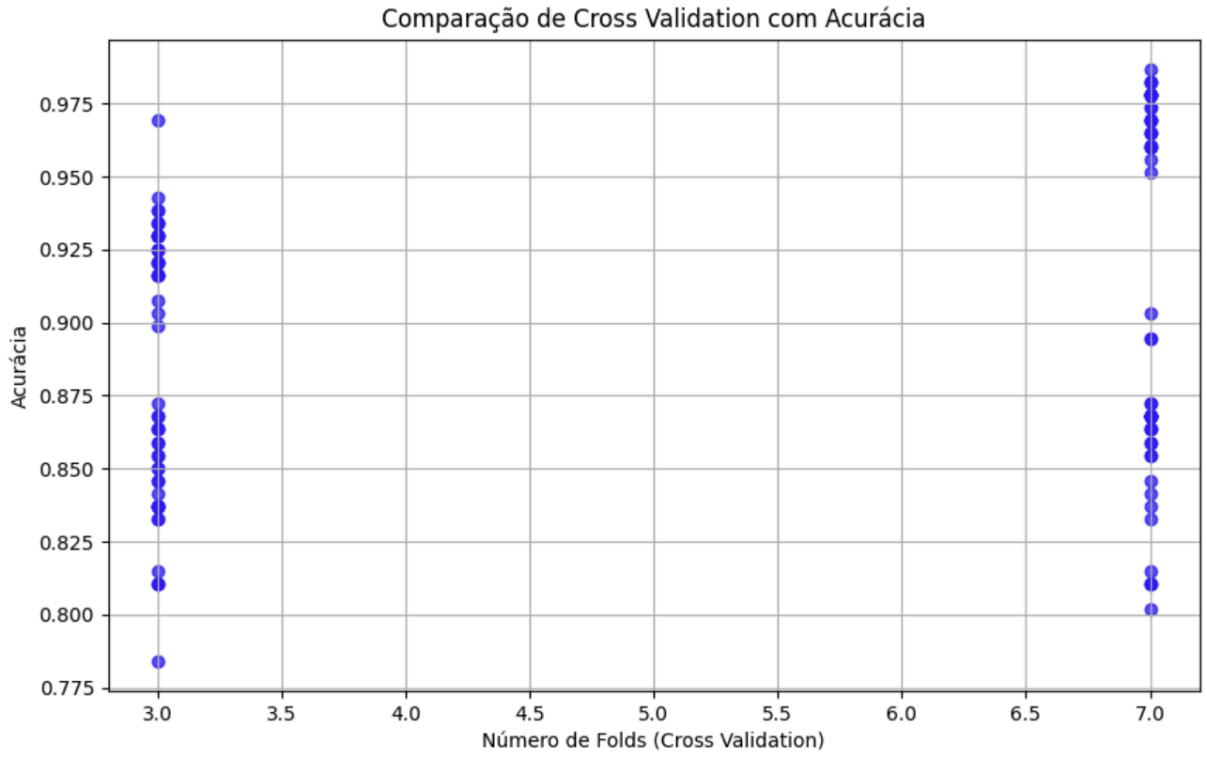

Figura 5 - Comparação da acurácia do k do cross validation pro KNN

Além disso, é possível notar, ao analisar o gráfico da Figura 6 a seguir, que quanto maior a quantidade de vizinhos, menor os valores médios da acurácia. Assim, é perceptível que, em k igual a três vizinhos, num geral, a acurácia do modelo é aumentada, o que aumenta o seu desempenho.

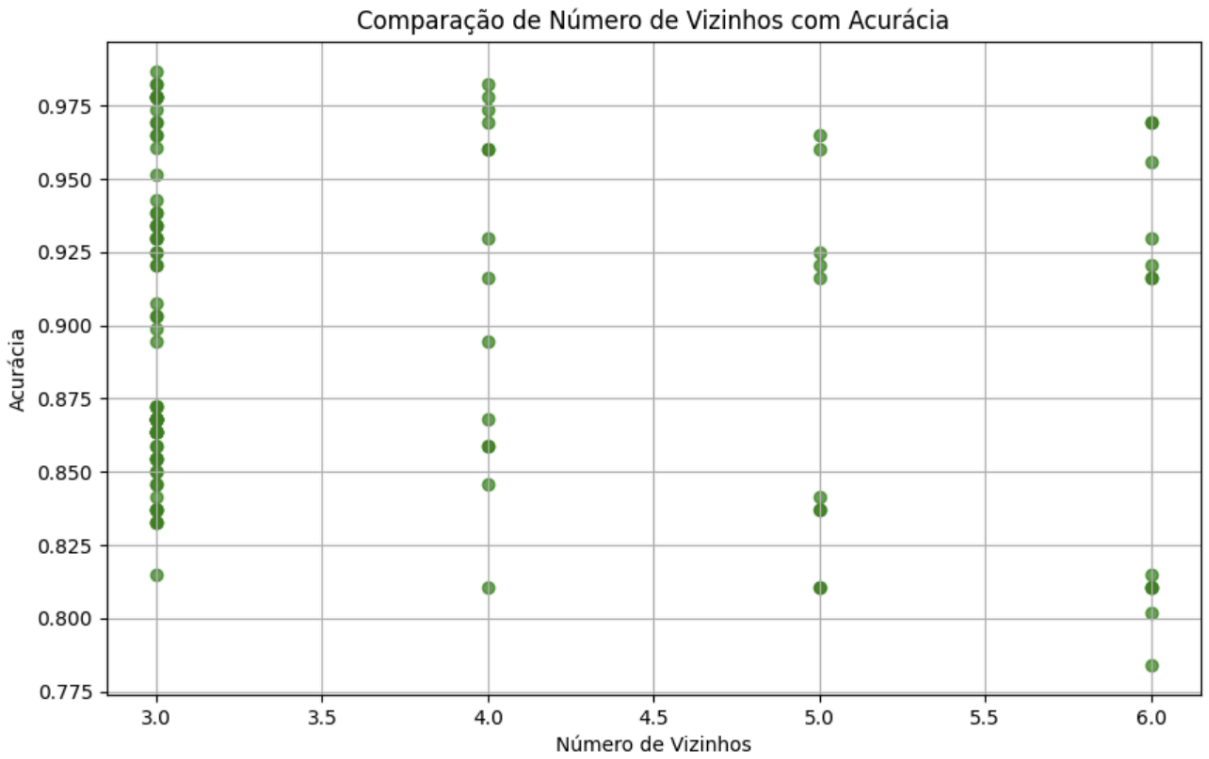

Figura 6 - Comparação da acurácia do k do número de vizinhos pro KNN

A Figura 7 abaixo compara a acurácia com dois tipos de peso, por distância ou uniforme. Em ‘distance’, os vizinhos são ponderados de acordo com a distância. Já em ‘uniform’, todos os vizinhos são ponderados igualmente. É notório que, ao ponderar considerando a distância, há maior acurácia. Logo, visando aumentar o desempenho do modelo, seria recomendável utilizar ‘distance’ em vez de ‘uniform’.

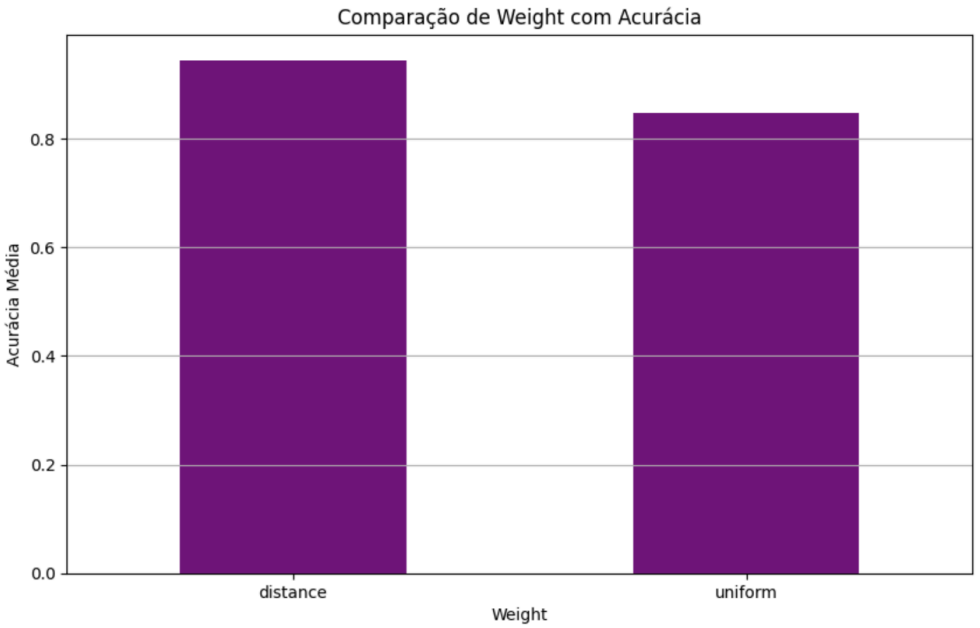

Figura 7 - Comparação da acurácia dos diferentes pesos pro KNN

Por fim, em relação às colunas que devem ser usadas como dados de entrada para o KNN, é notável que as colunas ca e thal aparecem, respectivamente, em um e em dois dos cinco piores cenários. Logo, como aparecem menos frequentemente, devem permanecer no modelo. Entretanto colunas como fbs, thalach e exang foram selecionadas como colunas de entrada em todos os 5 piores cenários. Sendo assim, é recomendada a sua retirada do modelo, objetivando um aumento de desempenho.

4.2 Árvore de decisão

Foram executados 223 diferentes cenários na árvore de decisão, variando o valor dos parâmetros e da forma de normalização. Visando verificar quais parâmetros foram os que mais afetaram o desempenho do modelo, foram feitas diversas análises utilizando gráficos como base.

A acurácia média é uma medida do desempenho do modelo, calculada como a proporção de previsões corretas sobre o total de previsões feitas, normalmente avaliada através de validação cruzada. No eixo X da Figura 8, há diferentes valores

para o parâmetro ‘min_samples_leaf’ e cada valor representa o número mínimo de amostras que cada nó folha deve conter após a divisão. Para valores pequenos, pode ocorrer overfitting, mas para valores grandes, o modelo pode ser muito simplista, podendo ocorrer underfitting. Vemos, no gráfico, que não há uma padronização de grandeza de valores com alta ou baixa acurácia. Entretanto, a maior acurácia exibida no gráfico possui um ‘min_samples_leaf’ igual a três.

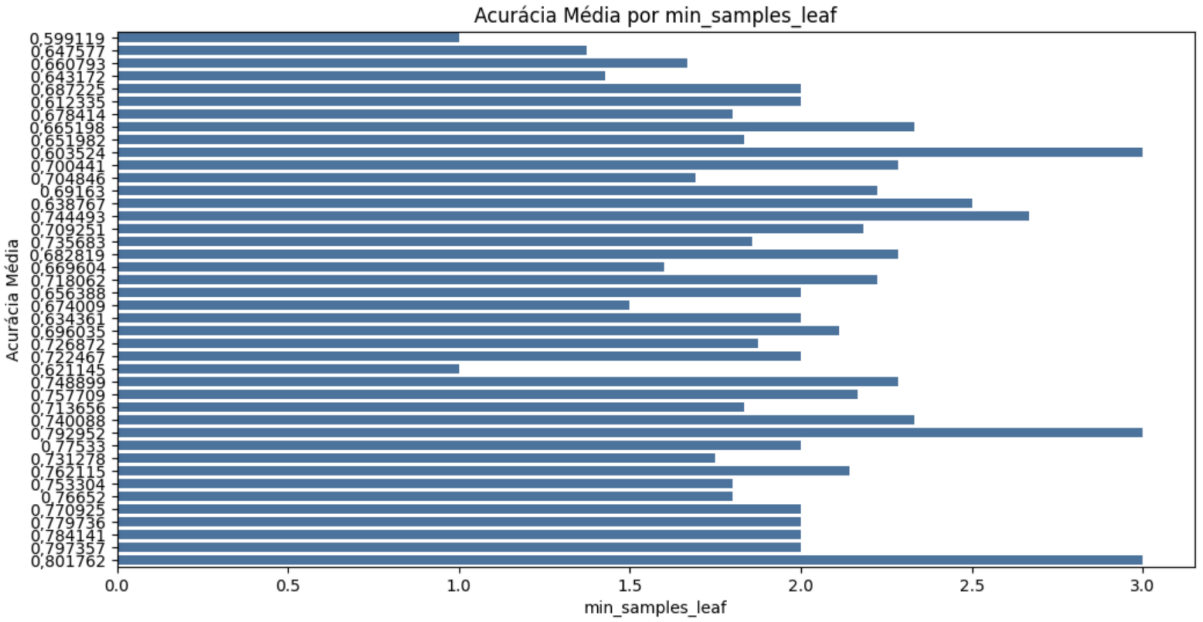

Figura 8 - Comparação da acurácia dos diferentes min_samples_leaf para árvore de decisão

O parâmetro ‘max_features’ controla o número máximo de características a serem consideradas para encontrar a melhor divisão para cada nó da árvore. Os pontos são divididos em três tipos: Gini, que representa o critério de impureza Gini para a divisão de nós; Entropia, que leva em conta o ganho de informação, e ‘Log Loss’ que utiliza o critério de perda logística para a divisão dos nós. Pelo gráfico da Figura 9, é observada uma grande variedade na distribuição dos critérios, tanto em ‘None’, como em raiz e log 2. Porém, é interessante observar que em ‘None’, há critérios com acurácia com as porcentagens mais baixas, enquanto log2 tem as porcentagem maiores.

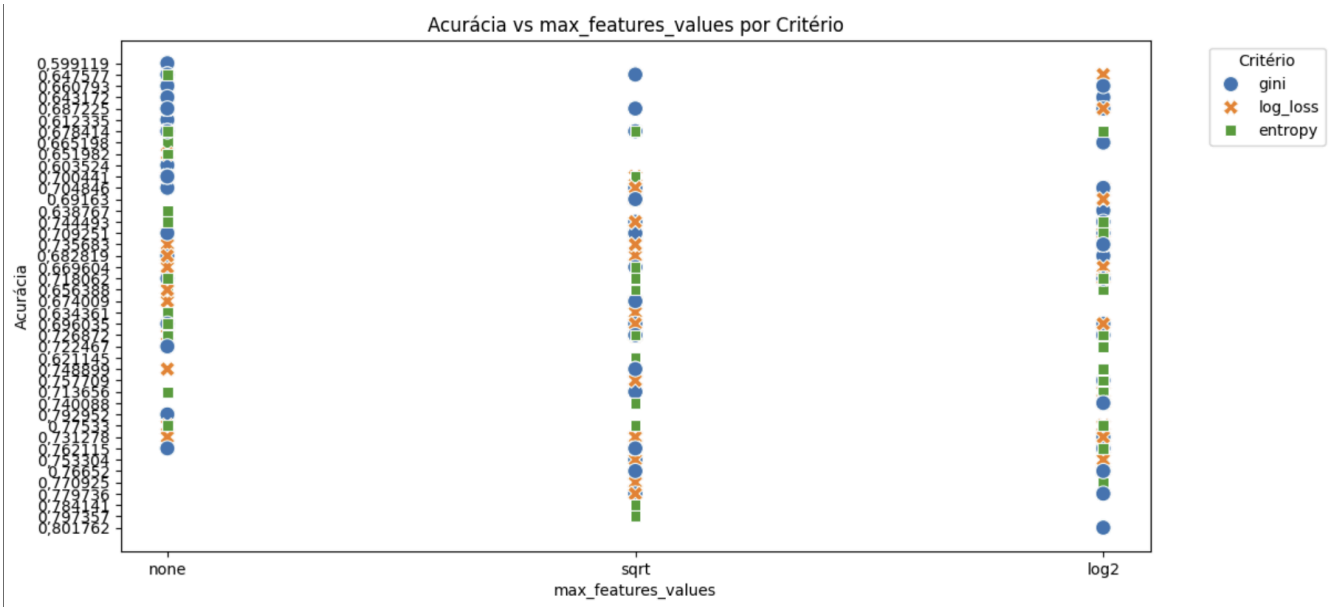

Figura 9 - Acurácia dos diferentes min_samples_leaf para árvore de decisão por critério

Na Figura 10, podemos notar que quanto maior o valor de k, maior a precisão tem nosso modelo, independente de qual seja o critério escolhido. Além disso, a crescente precisão é linear, variando apenas o coeficiente angular de cada critério. Observando as três retas, por exemplo, a azul, que representa Gini, tem maior variação, ou seja, maior aumento de precisão, conforme aumenta o valor de k.

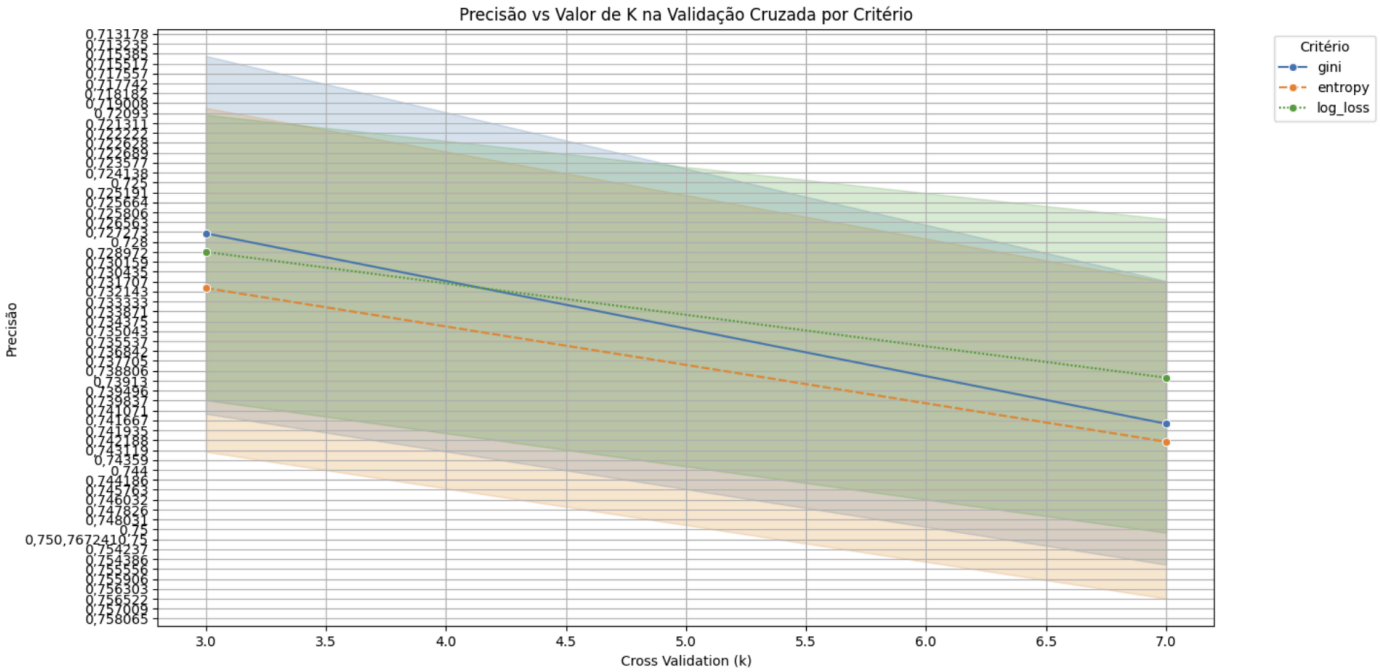

Figura 10 - Precisão vs Valor de K no cross validation por critério

Além disso, na Figura 11 abaixo, podemos observar que os dados manipulados têm alta variação no valor da acurácia. Entretanto, a padronização realizada pelo StandardScaler possui valores inferiores de acurácia do modelo.

Figura 11 - Acurácia dos diferentes modos de padronização para árvore de decisão

Nesse gráfico da Figura 12, a seguir, podemos observar um overfitting inicial, pois a F1-score começa alta e depois decai, pois é maior no treinamento do que na validação. A acentuação da curva em 2 indica que esse valor pode ser considerado bom para evitar overfitting e underfitting. Porém, a queda rápida indica que o modelo está se tornando muito simplista.

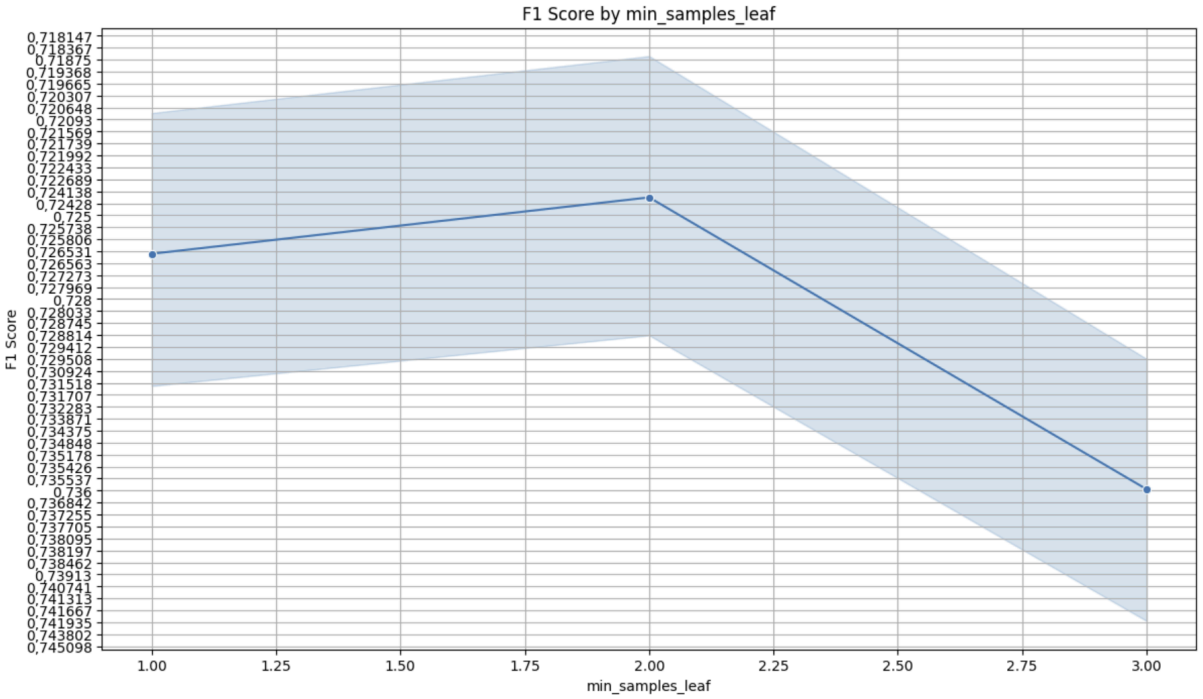

Figura 12 - F1 score by min_sample_leaf

A Figura 13 abaixo relaciona a F1-score e a validação cruzada, mostrando como o desempenho do modelo, medido pela F1-score, varia através dos diferentes folds da validação cruzada. Nesse caso, pode-se observar que a relação é linear, aumentando o valor de F1-score conforme aumenta a quantidade de folds.

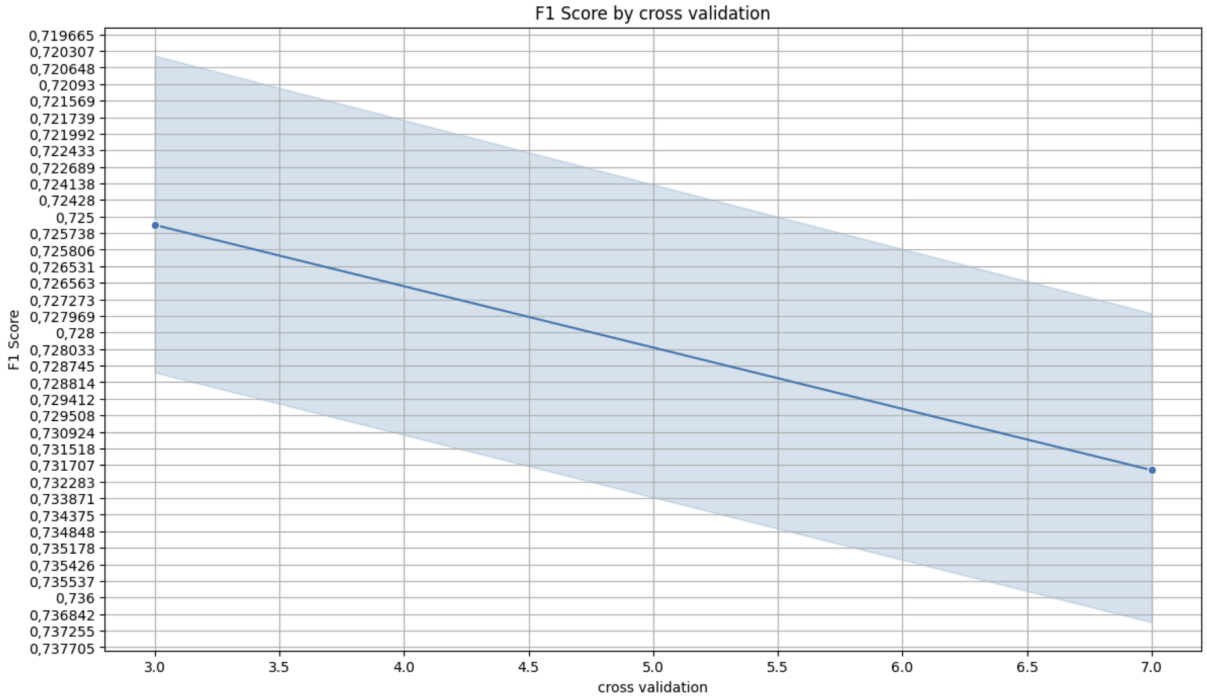

Figura 13 - F1 score por k do cross validation

Por fim, em relação às features que devem ser utilizadas como colunas de entrada para a árvore de decisão, é importante ressaltar que naquelas que, num geral, possuíam a menor acurácia, não se utilizou duas das colunas do modelo, sendo elas: ca e thal. Assim, essas features que devem ser selecionadas visando obter um melhor desempenho. Além disso, dez dos dezenove melhores cenários da árvore de decisão não utilizam a coluna restecg, logo, essa coluna não deve ser usada como dado de entrada para melhorar o desempenho do modelo. Como a coluna restecg não tem uma importância muito alta pro modelo, como comprovado pela Figura 14 abaixo, ela pode ser retirada. Ademais, segundo a imagem a seguir, é possível perceber que a coluna fbs também pode ser removida, visto que possui pouca importância para o modelo.

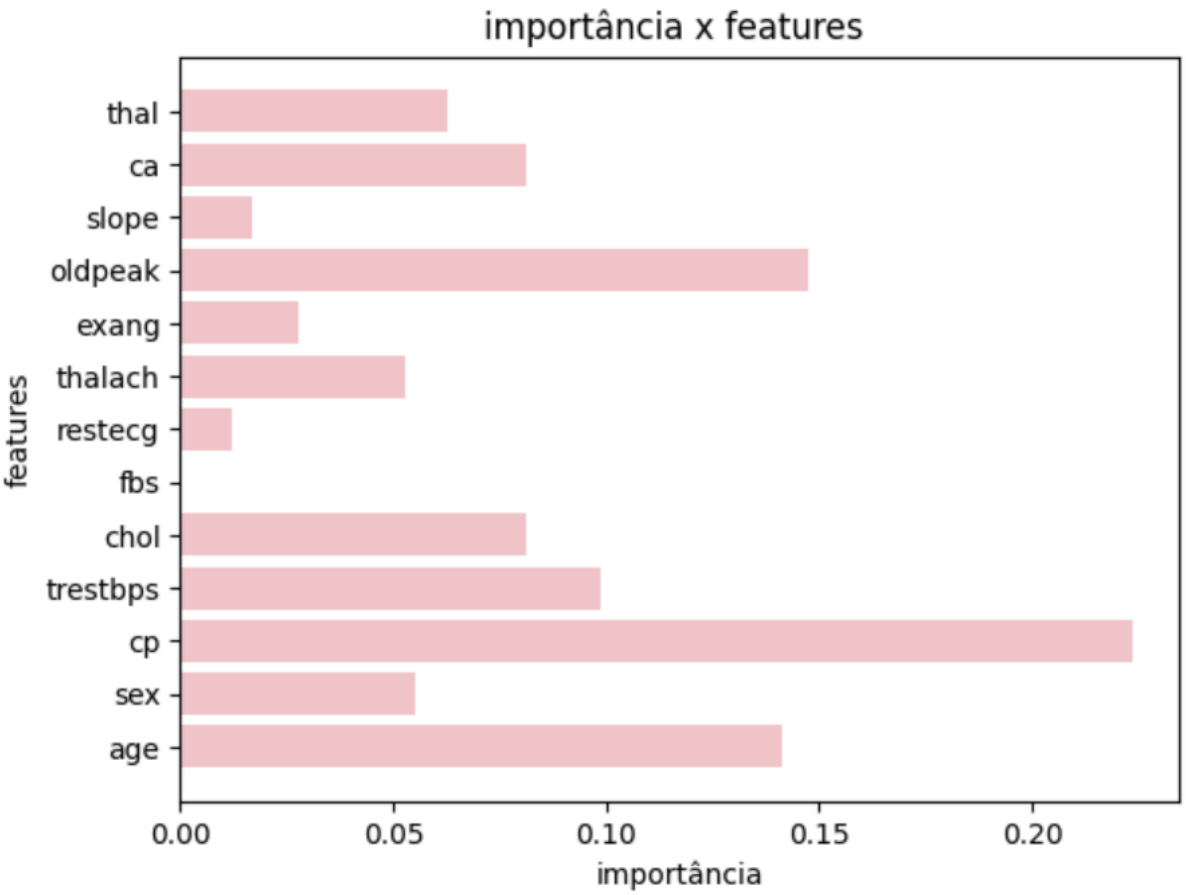

Figura 14 - Importância x features

5. CONCLUSÃO

O melhor cenário de execução do KNN foi o 26 e ele obteve 0,9868 de acurácia, 0,9919 de precisão, 0,9839 de recall e 0,979 de medida F1. O cenário pode ser visualizado na Tabela 1 abaixo com uma coloração verde (as colunas correspondem, respectivamente, a VN, VP, FN,FP, Acurácia, Precisão, Recall e Medida F1). Num geral, essas métricas estavam mais altas nesse cenário mencionado. Além disso, nesse cenário do KNN, o k do cross validation é igual a sete, o que pode melhorar o desempenho do modelo. Ele também utilizou o StandardScaler como forma de padronização e obteve um alto desempenho apesar dos cenários com o uso do MinMaxScaler terem uma precisão média maior. Além disso, o seu n_neighbours é igual a três, que (num geral) garante maior acurácia para as execuções testadas. A métrica é de minkowski e o peso é *distance,* que possuíam, respectivamente, uma maior precisão e acurácia média, aumentando o desempenho do KNN.

77	108	16	26	0,8150	0,8060	0,8710	0,8372
99	119	5	4	0,9604	0,9675	0,9597	0,9636
87	110	14	16	0,8678	0,8730	0,8871	0,8800
95	118	6	8	0,9383	0,9365	0,9516	0,9440
84	110	14	19	0,8546	0,8527	0,8871	0,8696
102	122	2	1	0,9868	0,9919	0,9839	0,9879
100	122	2	3	0,9780	0,9760	0,9839	0,9799
86	104	20	17	0,8370	0,8595	0,8387	0,8490
87	111	13	16	0,872247	0,874016	0,895161	0,884462
96	115	9	7	0,929515	0,942623	0,927419	0,934959
96	115	9	7	0,929515	0,942623	0,927419	0,934959

Tabela 1 - Melhor cenário pro KNN

Já o melhor cenário de execução da árvore de decisão foi o 218 e ele obteve 0,801762 de acurácia, 0,811024 de precisão, 0,830645 de recall e 0,820717 de medida F1. O cenário pode ser visualizado na imagem abaixo (as colunas correspondem, respectivamente, a acurácia, precisão, recall e medida F1). Essas quatro métricas estavam superiores nesse cenário mencionado em relação ao

restante das execuções, por isso, foi selecionado. Além disso, nesse cenário da árvore de decisão, o k do cross validation é igual a sete, melhorando o desempenho do modelo. Ele também utilizou o StandardScaler como forma de padronização. Além disso, o seu critério é igual a log_loss, que (num geral) garante maior acurácia para as execuções testadas. A min_samples_leaf é de três e o max_features é log2, o que tem a tendência a aumentar o desempenho da árvore de decisão baseado nos gráficos analisados anteriormente. Nesse cenário, apenas onze das treze features foram utilizadas para a execução desse cenário. Esse cenário teve um desempenho menor em relação ao melhor cenário do KNN e suas métricas são representadas pela linha de cor verde abaixo (Tabela 2).

0,740088	0,755906	0,774194	0,76494
0,753304	0,783333	0,758065	0,770492
0,69163	0,721311	0,709677	0,715447
0,77533	0,811966	0,766129	0,788382
0,801762	0,811024	0,830645	0,820717
0,748899	0,768	0,774194	0,771084
0,762115	0,786885	0,774194	0,780488
0,718062	0,767857	0,693548	0,728814

Tabela 2 - Melhor cenário para árvore de decisão

REFERÊNCIAS BIBLIOGRÁFICAS

BUTTON, Sérgio Tonini. **METODOLOGIA PARA PLANEJAMENTO EXPERIMENTAL E ANÁLISE DE RESULTADOS**. 2012. 88 f. Dissertação (Doutorado) - Curso de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, 2012. Disponível em: https://www.fem.unicamp.br/~sergio1/pos-graduacao/IM317/apostila2012.pdf. Acesso em: 20 maio 2024.

CANDIDO, Gustavo. **Árvore de Decisão**: um dos algoritmos mais poderosos do aprendizado de máquina.. Um dos algoritmos mais poderosos do aprendizado de máquina.. 2023. Disponível em: https://medium.com/data-hackers/%C3%A1rvore-de-decis%C3%A3o-88c7d0fd7a31. Acesso em: 19 maio 2024.

DUARTE, Rafael. **Guia Básico de Pré-Processamento de Dados**. 2020. Disponível em: https://sigmoidal.ai/guia-basico-de-pre-processamento-de-dados/#:~:text=MinMaxScaler,escala%20em%20um%20determinado%20range.. Acesso em: 19 maio 2024.

IBM. **Distâncias Medidas de Dissimilaridade para Dados Interval**. 2023. Disponível em: https://www.ibm.com/docs/pt-br/spss-statistics/saas?topic=measures-distances-dissimilarity-interval-data. Acesso em: 18 maio 2024.

INÁCIO, Diego. **Métricas de distância e dissimilaridade**: uma abordagem simplificada sobre métricas de distância e dissimilaridade, aplicáveis ao data science.. Uma abordagem simplificada sobre métricas de distância e dissimilaridade, aplicáveis ao Data Science.. 2021. Disponível em: https://diegoinacio.medium.com/metricas-de-distancia-e-dissimilaridade-94f9d8d962d4. Acesso em: 19 maio 2024.

JOSÉ, Italo. **KNN (K-Nearest Neighbors) #1**: como funciona?. Como funciona?. 2018. Disponível em: https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e. Acesso em: 19 maio 2024.

KUMAR, Ajitesh. **MinMaxScaler vs StandardScaler – Python Examples**. 2023. Disponível em: https://vitalflux.com/minmaxscaler-standardscaler-python-examples/. Acesso em: 19 maio 2024.

MARIO FILHO,. **O Que É Acurácia Em Machine Learning?** 2023. Disponível em: https://mariofilho.com/o-que-e-acuracia-em-machine-learning/. Acesso em: 20 maio 2024.

PRADO, Tatiana. **O que é e para que serve o planejamento de experimentos?**: descubra o passo a passo de como fazer um planejamento de experimentos para a sua empresa poder realizar mudanças e inovar o seu processo.. Descubra o passo a passo de como fazer um planejamento de experimentos para a sua empresa poder realizar mudanças e inovar o seu processo.. 2020. Disponível em: https://www.voitto.com.br/blog/artigo/planejamento-de-experimentos. Acesso em: 19 maio 2024.

SACRAMENTO, Gabriel. **ÁRVORE DE DECISÃO: ENTENDA ESSE ALGORITMO DE MACHINE LEARNING**: a árvore de decisão é um importante algoritmo de ml para dominar, devido à sua versatilidade. entenda como funcionam as decision trees.. A árvore de decisão é um importante algoritmo de ML para dominar, devido à sua versatilidade. Entenda como funcionam as decision trees.. Disponível em: https://blog.somostera.com/data-science/arvores-de-decisao. Acesso em: 19 maio 2024.

SCIKIT-LEARN. **Sklearn.neighbors.KNeighborsClassifier**. 2024. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. Acesso em: 19 maio 2024.

SCIKIT-LEARN. **Sklearn.tree.DecisionTreeClassifier**. 2024. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. Acesso em: 19 maio 2024.

ANEXOS

ANEXO A - Árvore de decisão pré-processamento

# Cenário	Pré - Processamento													
	Padronização		Dataset											
	MinMaxScaler	StandardScaler	Excel	Colunas de entrada										
				age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
1		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
2		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
3		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
4		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
5		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
6		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
7		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
8		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
9		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
10		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
11		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
12		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	
13		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x	

ANEXO B - Árvore de decisão mineração e pós-processamento

	Mineração					Pós - Processamento							
	ÁRVORE DE DECISÃO (sklearn.tree.DecisionTreeClassifier)					Cross Validation	Medidas						
			criterion										
thal	max_features	min_samples_leaf	gini	log_loss	entropy	k	VN	VP	FN	FP	Acurácia	Precisão	Recall	Medida F1
	none	1	x			3	57	79	45	46	0,599119	0,632	0,637097	0,634538
	none	1	x			7	60	87	37	43	0,647577	0,669231	0,701613	0,685039
	none	1		x		3	59	91	33	44	0,660793	0,674074	0,733871	0,702703
	none	1		x		7	67	79	45	36	0,643172	0,686957	0,637097	0,661088
	none	1			x	3	61	95	29	42	0,687225	0,693431	0,766129	0,727969
	none	1			x	7	66	80	44	37	0,643172	0,683761	0,645161	0,6639
	none	2	x			3	63	76	48	40	0,612335	0,655172	0,612903	0,633333
	none	2	x			7	65	82	42	38	0,647577	0,683333	0,66129	0,672131
	none	2		x		3	62	92	32	41	0,678414	0,691729	0,741935	0,715953
	none	2		x		7	69	77	47	34	0,643172	0,693694	0,620968	0,655319
	none	2			x	3	58	93	31	45	0,665198	0,673913	0,75	0,709924
	none	2			x	7	71	77	47	32	0,651982	0,706422	0,620968	0,660944
	none	3	x			3	58	79	45	45	0,603524	0,637097	0,637097	0,637097

ANEXO C - KNN pré-processamento

Cenário	Pré - Processamento															
	Padronização		Dataset													
	MinMaxScaler	StandardScaler	Excel	Colunas de Entrada												
				age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	minkowski
1		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x			x
2		x	heart-disease	x	x	x	x	x	x	x	x	x	x	x			x
3		x	heart-disease	x	x	x	x	x	x	x	x	x	x		x		x
4		x	heart-disease	x	x	x	x	x	x	x	x	x	x			x	
5		x	heart-disease	x	x	x	x	x	x	x	x	x	x			x	
6		x	heart-disease	x	x	x	x	x	x	x	x	x	x		x		x
7		x	heart-disease	x	x	x	x	x	x	x	x		x	x	x		
8		x	heart-disease	x	x	x	x	x	x	x	x		x	x	x		
9		x	heart-disease	x	x	x	x	x	x	x	x		x		x	x	
10		x	heart-disease	x	x	x	x	x	x	x	x		x		x	x	
11		x	heart-disease	x	x	x	x	x	x	x	x		x	x	x		
12		x	heart-disease	x	x	x	x	x	x	x	x		x		x	x	

ANEXO D - KNN mineração e pós-processamento

KNN (sklearn.neighbors.KNeighborsClassifier)						Cross Validation (CV)	Pós - Processamento							
metric			weights		n_neighbors	(k)	Medidas							
minkowski	euclidean	manhattan	uniform	distance			VN	VP	FN	FP	Acurácia	Precisão	Recall	Medida F1
x			x		3	3	81	108	16	22	0,8326	0,8308	0,8710	0,8504
x				x	3	3	93	117	7	10	0,9251	0,9213	0,9435	0,9323
x			x		3	3	84	108	16	19	0,8458	0,8504	0,8710	0,8606
	x			x	3	3	84	114	10	19	0,8722	0,8571	0,9194	0,8872
		x	x		3	3	78	111	13	25	0,8326	0,8162	0,8952	0,8538
x			x		3	7	84	112	12	19	0,8634	0,8550	0,9032	0,8784
	x			x	3	7	100	119	5	3	0,9648	0,9754	0,9597	0,9675
		x		x	3	7	102	120	4	1	0,9780	0,9917	0,9677	0,9796
	x		x		3	7	81	111	13	22	0,8458	0,8346	0,8952	0,8638
	x			x	3	3	93	113	11	10	0,9075	0,9187	0,9113	0,9150
		x	x		3	3	85	108	16	18	0,8502	0,8571	0,8710	0,8640
		x	x		3	3	82	113	11	21	0,8590	0,8433	0,9113	0,8760
x			x		3	3	85	105	19	18	0,8370	0,8537	0,8468	0,8502
x				x	3	7	100	120	4	3	0,9692	0,9756	0,9677	0,9717

https://docs.google.com/spreadsheets/d/1rad7b1bkZ5-yJeFJZp-1O90KFegK1Ff3/edit?usp=sharing&ouid=106633824793947144469&rtpof=true&sd=true