

UNIVERSIDADE DO MINHO

Machine Learning

Licenciatura em Engenharia Biomédica

Inteligência Artificial em Engenharia Biomédica
1º semestre/2022-2023

A96353 Inês Margarida Mendes
A95637 Luís Manuel Gonçalves
A95524 Miguel Borges Vale

Braga, 21 de dezembro de 2022

Resumo

Machine Learning é um tipo de inteligência artificial que permite que os sistemas aprendam e melhorem automaticamente a partir da experiência, sem precisar ser explicitamente programados. Ele envolve o uso de algoritmos que podem analisar e reconhecer padrões em dados e tomar decisões ou fazer previsões com base nesses dados.

Estes tipos de técnica têm vários benefícios em diferentes áreas, tais como, maior precisão, podem analisar e processar grandes quantidades de dados aumentam a eficiência e possibilita uma tomada de decisão mais rápida.

Para trabalhar com machine learning a programa escolhido foi o Knime. O KNIME é uma plataforma de análise de dados, que integra vários componentes de Machine Learning e tratamento de dados através de um conceito de construção de fluxos de dados modular. Possui uma interface gráfica que permite ao usuário a combinação de fontes de dados diferentes para modelagem, análise e visualização de dados sem a necessidade de programação.

Índice

1.Introdução	4
2.Análise da Base de Dados	4
3.Métodos Escolhidos	5
3.1K-means	5
3.2Decision Tree	5
3.3Multi Layer Perceptron	6
4.Tratamento dos dados	7
5.Nodos de Aprendizagem	16
6.Análise de Resultados	17
7.Conclusão	19
8.Bibliografia	19

1.Introdução

No âmbito da unidade curricular de Inteligência Artificial em Engenharia Biomédica, foi proposto o desenvolvimento de um sistema aprendizagem automática para estimar o atributo target de um conjunto de estudos relativos a doenças cardíacas realizados em 4 sítios diferentes. Este trabalho consistiu na utilização dos diferentes nodos disponíveis na plataforma Knime para automatizar o diagnóstico de pacientes com doenças cardíacas e analisar e quantificar a gravidade da doença, através de dados da base de dados fornecida. Ao longo deste relatório vai ser apresentado todo o processo para o alcance do objetivo final, bem como a estratégia e o raciocínio adotados.

2.Análise da Base de Dados

Para a realização deste trabalho foram fornecidas algumas bases de dados referentes a estudos realizados em vários locais sobre doenças cardíacas, sendo estes Cleveland, Long Beach, Suíça e Hungria. Nestes estudos, foram avaliados diversos parâmetros de cada paciente de modo a realizar o diagnóstico sobre doença cardíaca. Em cada dataset existem 14 colunas, sendo que as 13 primeiras dizem respeito a parâmetros e a última é referente ao diagnóstico.

Como a maioria dos parâmetros estavam escritos como siglas ou em inglês, para uma melhor compreensão dos dados fornecidos, foi consultado o link fornecido no enunciado do trabalho prático e, dessa forma, conseguimos retirar as seguintes informações:

- cp – dor de peito, sendo o valor 1 atribuído a angina típica, o valor 2 a angina atípica, o valor 3 a dor não angínica e o valor 4 a assintomáticos.
- trestbps – pressão arterial, em mmHg
- chol – colesterol, em mg/dL
- fbs – nível de glicemia em jejum, sendo o valor 1 atribuído se corresponder a um nível abaixo de 120 mg/dL e o valor 0 caso contrário
- restecg – eletrocardiograma, sendo o valor 0 atribuído quando exame é normal, o valor 1 quando apresenta anormalidades na onda ST-T e o valor 2 quando mostra hipertrofia ventricular esquerda provável ou definitiva pelos critérios de Estes
- thalach – maior pulsação
- exang – angina induzida por exercício, sendo o valor 1 quando há indução e o valor 0 caso contrário
- oldpeak – depressão do segmento ST
- slope – declive do segmento ST em esforço, sendo atribuído o valor 1 se for ascendente, o valor 2 se for plano e o valor 3 se for descendente
- ca – vasos marcados pela fluoroscopia, que vai de 0 a 3
- thal – talassemia, sendo atribuído o valor 3 a casos normais, o valor 6 quando existe um defeito fixo e o valor 7 quando existe um defeito reversível
- num – target, sendo atribuído o valor 0 no caso de inexistência de doença cardíaca e, no caso de existência, valores de 1 a 4 de acordo com o grau de severidade da doença

Outra ferramenta útil usada para melhor compreensão das bases de dados foi o nodo statistics da plataforma Knime. Tal como o nome diz, através dele é possível analisar diversas estatísticas relativas às bases de dados de forma simples. Conseguimos perceber, por exemplo, que havia vários parâmetros com um número de valores em falta elevado, como por exemplo, os “vasos marcados pela fluoroscopia”, que possuía 611 missing values. Um número tão elevado de valores em falta não favorece o desenvolvimento de um sistema de aprendizagem bom. Conseguimos também perceber que o número de pacientes do sexo masculino é muito superior ao número de pacientes do sexo feminino e que a média das idades foi de 53,5 anos.

O objetivo principal neste trabalho era desenvolver um sistema de aprendizagem automática capaz de realizar o diagnóstico mais assertivo possível e que fosse aplicável de forma geral aos mais diversos locais.

Na análise dos datasets fornecidos, verificou-se que para os estudos realizados em Cleveland, em Long Beach e na Suíça, o target variava entre os valores inteiros de 0 e 4. Porém, ao analisar o dataset do estudo realizado na Hungria, verificou-se que o target apenas apresentava dois valores, ou 0, ou 1. Sendo assim, assumiu-se que o diagnóstico neste caso não foi feito baseado no grau de severidade da doença cardíaca, mas baseado na existência ou não existência desta. Ora, usar diretamente esta base de dados para treino ou teste não seria o mais adequado visto que o método de diagnóstico foi diferente relativamente aos restantes datasets. Como solução para este problema, foi decidido inicialmente usar uma parte dos datasets de Cleveland, Long Beach e Suíça para treino e a parte de diagnósticos positivos do dataset da Hungria para teste. Deste modo, conseguimos atribuir uma classificação de 1 a 4 para o target do estudo realizado na Hungria. De seguida, com a restante parte das bases de dados de Cleveland, Long Beach e Suíça e com a base de dados da Hungria “modificada” e os diagnósticos negativos desta, procedeu-se ao desenvolvimento do sistema de aprendizagem final. Todo este processo, vai ser explicado mais à frente.

3. Métodos Escolhidos

O Knime oferece uma variedade de métodos para trabalhar com Machine Learning dentro deles iremos trabalhar com apenas 3 destes métodos. Para eles escolhemos o método da Segmentação, Árvore de Decisão e Redes Neurais, também conhecidos como K-means, Decision Tree e MultiLayer Perceptron, respetivamente.

3.1. K-means

O clustering k-means é um método de agrupamento, que visa particionar n observações em k clusters em que cada observação pertence ao cluster com a distância mais pequena de n centroides, posicionados de modo aleatório no espaço de valores, servindo como um protótipo de o cluster. O conjunto de dados de cada cluster formam um ponto médio calculável pela posição de cada ponto no cluster, sendo esse ponto médio o novo centroide, o processo repete-se sucessivamente até que os pontos mais próximos de cada centroide sejam os pontos pertencentes aos próprios clusters, isto resulta num particionamento do espaço de dados em células. O agrupamento k-means minimiza as variâncias dentro do cluster.

3.2. Decision Tree

Uma árvore de decisão é uma técnica de classificação de machine learning, atendendo ao apoio à decisão representada por um grafo hierarquizado. Este contém um conjunto de nodos, que testam os atributos de um dado dataset, ramos que identificam valores de separação do nodo pai e folhas que estão associadas às decisões a tomar nesse caso.

A procura pela folha de encaixe com uma nova decisão ou com um objeto segue sempre o mesmo método de começar pela “raiz” da árvore e descer de modo a alcançar sucessivos nodos cujos ramos sejam verdadeiros para o objeto escolhido até alcançar um nodo folha, sendo que esta indica a decisão a tomar no problema.

As árvores de decisão representam árvores binária que podem dividir recursivamente o conjunto de dados até que fiquemos com nodos folha sem bipartição, ou seja, os dados com apenas um tipo de classe, ou construir o modelo a partir da identificação das relações entre os atributos do

dataset, sendo a forma da árvore induzida por generalização até conseguir definir todos os atributos num só nodo. Os algoritmos utilizados para este modelo baseiam-se no maior ganho de informação de todas as possibilidades de divisão dos dados, este algoritmo aprende qual conjunto de valores escolher com base nas informações obtidas nessa escolha. Dependendo das escolhas, o algoritmo irá aplicar equações relacionadas diretamente com a entropia, sendo esta uma boa identificadora do “grau de desorganização” dos dados, isto é, o modelo calcula a melhor escolha a ser feita para diminuir a incerteza de um estado. O cálculo poderá ser feito do nodo raiz para baixo, de forma a decidir como fazer a bipartição dos dados que deem mais informação, ou a partir das melhores possíveis escolhas de agrupamento dos atributos, de modo a maximizar ganho de informação.

3.3. Multi Layer Perceptron

O Multi Layer Perceptron, corresponde a uma rede neuronal artificial (RNA), este modelo é baseado no sistema nervoso central do ser humano, sendo os componentes mais básicos designados de neurónios baseados nos mesmos da mente humana, que em junção com múltiplos destes, formam uma rede neuronal com a capacidade de resolver problemas de maior magnitude e complexidade que os outros modelos.

Este modelo contém pelo menos dois neurónios de entrada e um de saída, sendo o número de camadas entre as duas e o número de neurónios variável, cada neurónio é definido pelo seu valor de estado, definido pelos valores recebidos dos neurónios anteriores e o seu peso que representa a sua importância. Este sinal é modificado pela função de ativação podendo ser binária, linear ou sigmoide e este valor de transferência correspondente será transmitido para o neurónio seguinte, repetindo este processo até alcançar o neurónio de saída. Sendo este resultado representativo do modelo treinado, para este treino é importante a variação dos pesos das ligações entre neurónios, tornando cada um destes como excitativo, inibidor ou nulo.

O número correto de camadas, neurónios e valores de pesos das ligações destes, irão definir o sucesso do modelo para um problema específico, tal como o tipo de regras de aprendizagem utilizada.

4. Tratamento dos dados

A maioria das vezes os dados recolhidos do mundo real são incompletos, contêm informações que não são importantes ou apresentam inconsistências, o que faz com que o dataset não seja adequado para uma determinada ferramenta de análise de dados. Por isso, é necessário fazer o tratamento dos dados para que a informação neles contida seja adequada para a elaboração de um melhor sistema de aprendizagem.

Neste trabalho, foram desenvolvidos vários sistemas de aprendizagem em diferentes workflows de modo a determinar qual seria o melhor. Porém a maioria dos nodos utilizados no tratamento de dados é comum a todos os workflows. Todos os sistemas de aprendizagem contêm duas instâncias de learning, como já tinha sido referido anteriormente neste relatório, que vão ser abordadas separadamente. Os nodos usados na primeira parte foram:

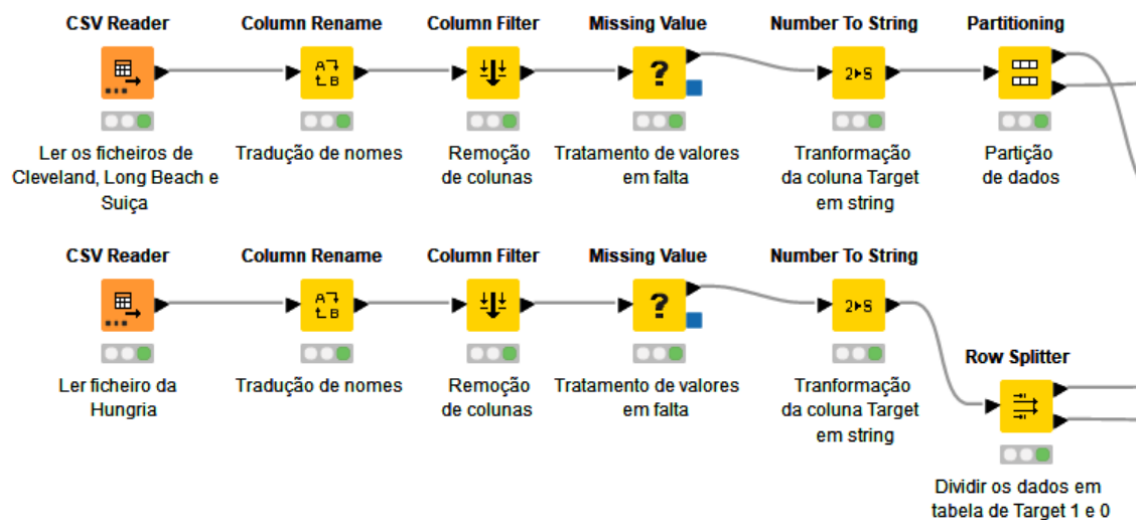


Figura 1 – Conjunto de nodos usados para tratamento de dados

- *2 CSV Reader*: o primeiro *CSV Reader* foi usado para a leitura dos datasets de Cleveland, Long Beach e Suíça conjuntamente, e o segundo foi usado para a leitura do dataset da Hungria.
- *Column Rename*: este nodo foi usado para renomear todas colunas de todos os datasets de modo a ficar mais fácil a análise e exploração destes.

thalach	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Maior Pulsação"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
exang	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Angina induzida por Exercício"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
oldpeak	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Depressão do segmento ST"/> <input type="button" value="D"/> DoubleValue <input type="button" value="v"/>
slope	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Decive do segmento ST em es"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
ca	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Vasos marcados pela fluorosco"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
thal	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Talassemia"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
age	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Idade"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
sex	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Gênero"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
cp	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Dor de Peito"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
trestbps	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Pressão Arterial"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
num	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Target"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
chol	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Colesterol"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
fbs	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Nível de Glicemia em Jejum"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>
restecg	Remove
<input checked="" type="checkbox"/> Change:	<input type="text" value="Eletrocardiograma"/> <input type="button" value="I"/> IntValue <input type="button" value="v"/>

Figuras 2 (a) e 2 (b) - Alterações feitas no nodo *Column Rename*

- *Column Filter*: com o auxílio deste nodo, foi filtrada a coluna relativa aos “Vasos marcados pela fluoroscopia”. A filtragem desta coluna deve-se ao facto de este parâmetro possuir um número extremamente elevado de valores em falta (611 valores). Mesmo que se fizesse algum tipo de tratamento a esta coluna, isto iria interferir muito com o sistema de aprendizagem pois a maior parte dos valores iriam ser valores “assumidos” e não valores reais.

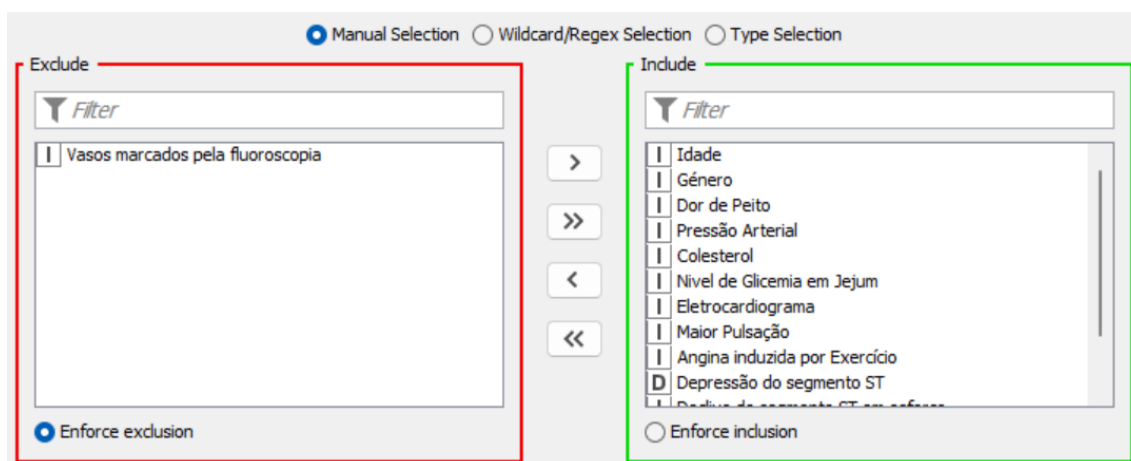


Figura 3 - Alterações feitas no nodo *Column Filter*

- *Missing Value*: este nodo foi usado para o tratamento dos valores em falta nos datasets. Para todos os valores em falta, estes foram substituídos por o respetivo valor normal do parâmetro. Para a talassemia, fizemos a substituição dos missing values por 3, para o nível de glicemia em jejum foi usado o valor 85, para a pressão arterial o valor 120 e para o colesterol o valor 200. Para a depressão do segmento ST, angina induzida por exercício e eletrocardiograma foi usado o valor 0 e para o declive do segmento ST em esforço o valor 1. No caso do parâmetro “Maior pulsação”, o valor normal é dado pela Fórmula de Tanaka - $208 - 0,7 \times \text{idade}$ - por isso, para este cálculo usamos a idade média que é 53,5 anos, obtendo assim o valor 170.

I Talassemia	<div>Remove</div> <div>Fix Value</div> <div>Value 3</div>
I Nivel de Glicemia em Jejum	<div>Remove</div> <div>Fix Value</div> <div>Value 85</div>
I Maior Pulsação	<div>Remove</div> <div>Fix Value</div> <div>Value 170</div>
I Pressão Arterial	<div>Remove</div> <div>Fix Value</div> <div>Value 120</div>
D Depressão do segmento ST	<div>Remove</div> <div>Fix Value</div> <div>Value 0,0</div>
I Angina induzida por Exercício	<div>Remove</div> <div>Fix Value</div> <div>Value 0</div>

<div> <div>Colesterol</div> </div>	<div>Remove</div> <div>Fix Value</div> <div>Value 200</div>
<div> <div>Eletrocardiograma</div> </div>	<div>Remove</div> <div>Fix Value</div> <div>Value 0</div>
<div> <div>Declive do segmento ST em esforço</div> </div>	<div>Remove</div> <div>Fix Value</div> <div>Value 1</div>

Figuras 4(a), 4(b) e 4(c) - Alterações feitas no nodo *Missing Value*

- *Number to String*: através deste nodo foi feita a conversão dos valores da coluna Target de número para string. Este passo deve-se ao facto de para fazer o treino, este atributo ter de ser nominal.

☒ Manual Selection
 ☐ Wildcard/Regex Selection

Exclude

Filter

Idade

Género

Dor de Peito

Pressão Arterial

Colesterol

Nível de Glicemia em Jejum

Eletrocardiograma

Maior Pulsação

Angina induzida por Exercício

Depressão do segmento ST

Declive do segmento ST em esforço

☒ Enforce exclusion

Include

Filter

Target

☐ Enforce inclusion

Figura 5 - Alteração feita no nodo *Number to String*

- *Partitioning*: este nodo foi apenas usado no pipeline superior, ou seja, apenas foi aplicado ao conjunto das bases de dados de Cleveland, Long Beach e Suíça. Este conjunto de datasets foi dividido de modo a obter uma parte para usar para treino para

um sistema de aprendizagem que foi aplicado posteriormente ao dataset da Hungria, neste caso 60%, e outra a ser utilizada depois, os restantes 40%. Esta partição foi feita de forma aleatória sendo usada uma *random seed* igual a 2022, de modo a obter a mesma partição aquando de uma nova execução.

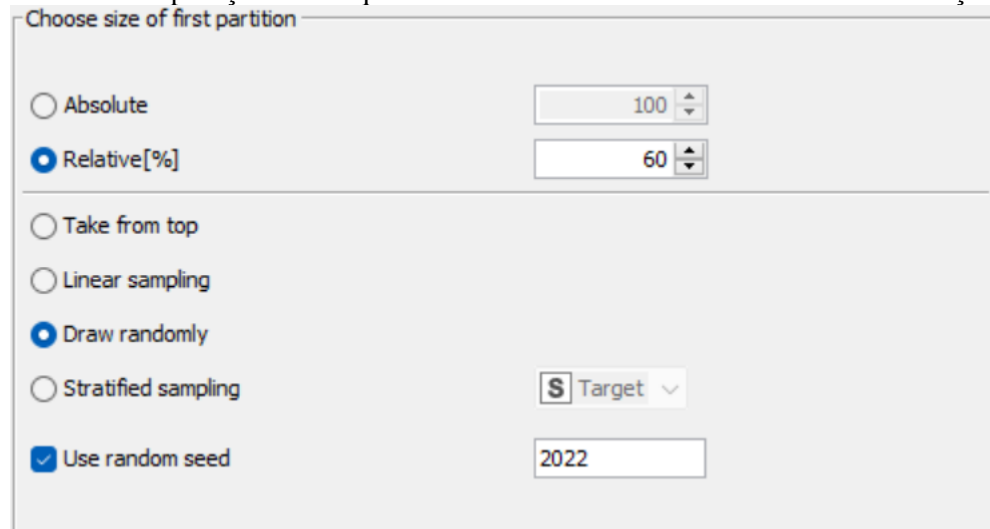


Figura 6 - Alteração feita no nodo *Partitioning*

- *Row Splitter*: este nodo foi apenas usado no pipeline inferior, ou seja, apenas foi aplicado ao dataset da Hungria, de modo a separar as linhas que continham um target igual a 1 das que tinham um target igual a 0. Assim, o primeiro conjunto foi utilizado como teste para conseguirmos obter valores de 1 a 4 de target. O segundo conjunto foi utilizado posteriormente.

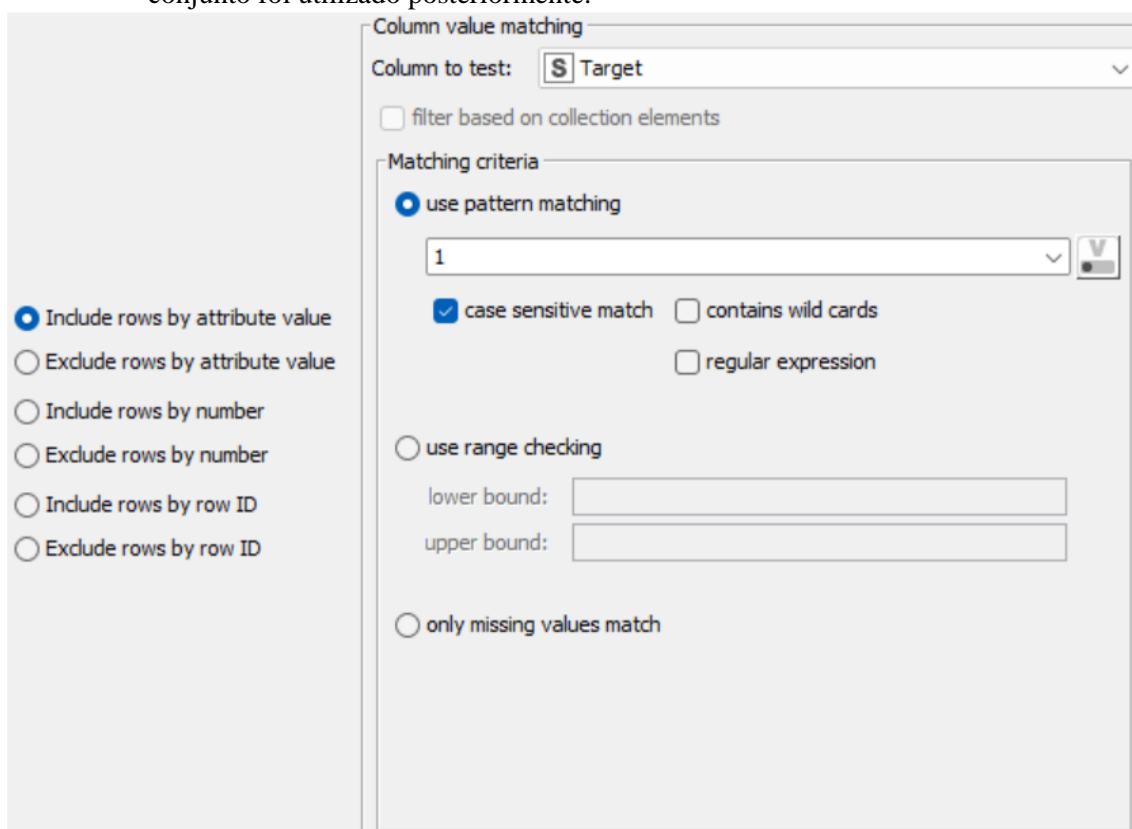


Figura 7 - Alteração feita no nodo *Row Splitter*

Nos workflows onde foi usado o método do MultiLayer Perceptron foi usado ainda outro nodo chamado Normalizer porque este método assim o exige. O objetivo deste nodo é fazer uma transformação que mantém a característica da distribuição dos dados, mas coloca os valores dentro de um intervalo específico, neste caso, entre 0 e 1. Neste nodo, foi excluído o parâmetro target exatamente porque o objetivo do trabalho é prever o valor deste parâmetro entre 0 e 4, logo não podemos colocar os valores entre 0 e 1.

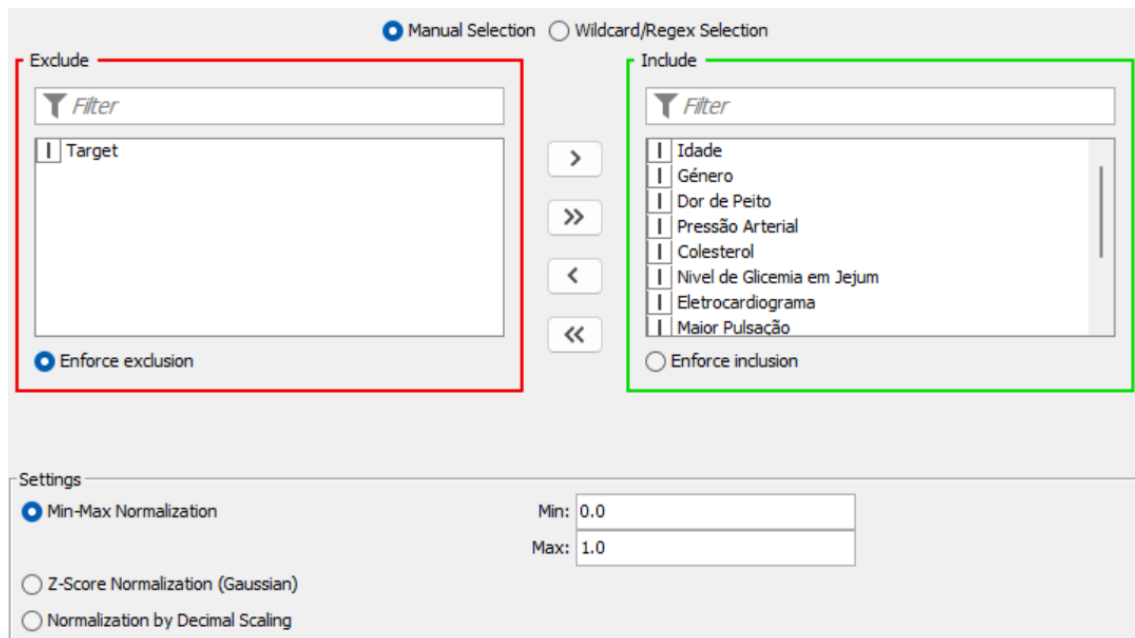


Figura 8 - Alteração feita no nodo *Normalizer*

Quanto à segunda parte do workflow, ou seja, após a aplicação do sistema de aprendizagem aos valores de diagnóstico positivos do dataset da Hungria, temos o seguinte conjunto de nodos:

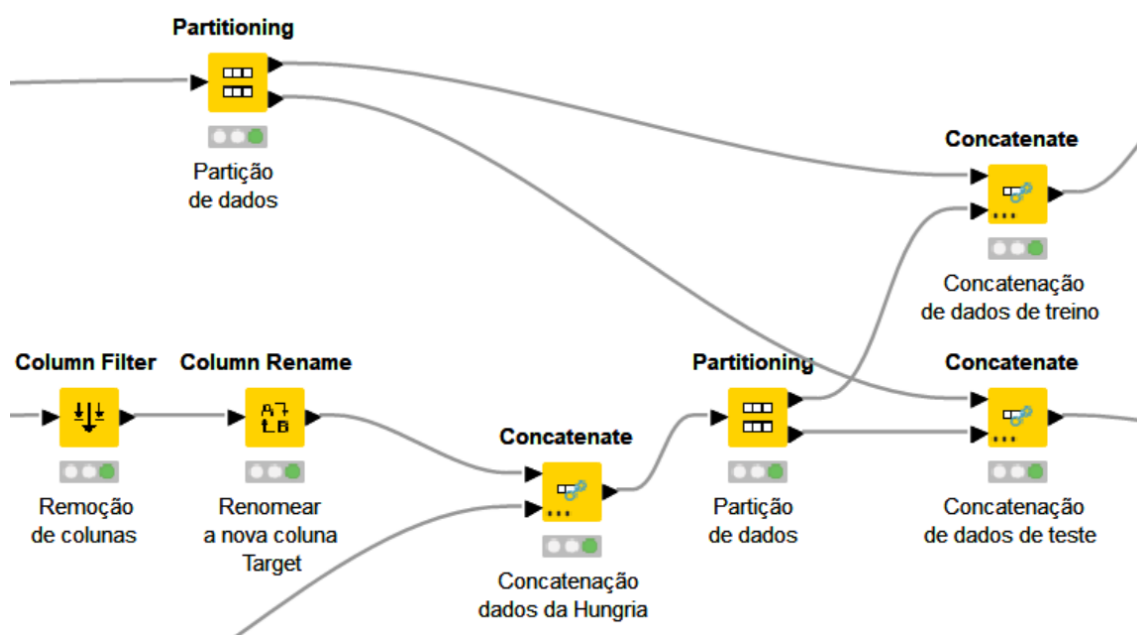


Figura 9 – Conjunto de nodos da segunda parte do workflow

Pipeline Superior:

- *Partitioning*: este nodo recebe o 40% do conjunto dos datasets de Cleveland, Long Beach e Suíça que tinham sido particionados anteriormente, e divide em dois conjuntos diferentes, um que vai utilizado para treino e outro que vai ser utilizado para teste. Foram separados 80% para treino e 20% para teste. Foi usada uma elevada percentagem usada para treino pois em comparação com o outro dataset disponível, os valores de target são mais “confiáveis” pois são os reais, enquanto os da base de dados da Hungria são fruto do primeiro sistema de aprendizagem realizado, que já vêm associados a um erro.

Choose size of first partition

☐ Absolute

☒ Relative[%]

☐ Take from top

☐ Linear sampling

☒ Draw randomly

☐ Stratified sampling

☒ Use random seed

Figura 10 – Alteração ao nodo *Partitioning 1*

- *Concatenate*: este nodo foi usado para juntar os conjuntos de dados previamente particionados para treino.

Pipeline Inferior:

- *Column Filter*: o conjunto de dados resultante do primeiro sistema de aprendizagem contém uma coluna chamada “Prediction (Target)” que é a previsão do valor do target de 1 a 4 relativo aos dados apresentados. Esses serão os valores que serão usados como target posteriormente, por isso, é necessário filtrar a coluna “Target” visto que esses valores já não são necessários.

Manual Selection Wildcard/Regex Selection Type Selection

Exclude

Filter

S Target

Enforce exclusion

Include

Filter

Idade
Género
Dor de Peito
Pressão Arterial
Colesterol
Nível de Glicemia em Jejum
Eletrocardiograma
Maior Pulsação
Angina induzida por Exercício
Depressão do segmento ST
Radio do segmento ST no repouso

Enforce inclusion

Figura 11- Alteração ao nodo *Column Filter*

- *Column Rename*: para que os dados sejam adequados para a ferramenta de análises de dados é necessário que todas as colunas usadas tenham o mesmo nome, por isso, é necessário substituir o nome da coluna “Prediction (Target)” por “Target”.

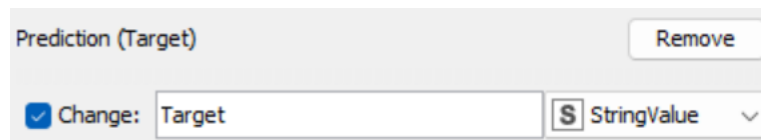


Figura 12 - Alteração ao nodo *Column Rename*

- *Concatenate*: este nodo junta o conjunto de dados resultante do primeiro sistema de aprendizagem com a parte que tinha sido particionada anteriormente referente aos diagnósticos negativos do estudo realizado na Hungria.
- *Partitioning*: este nodo divide a base de dados da Hungria numa parte que vai ser usada para treino (concatenada depois com a parte de treino referente aos datasets de Cleveland, Long Beach e Suíça) e uma parte para teste. Neste caso, foram particionados 60% para treino e 40% para teste. Estes valores são menores do que os referentes aos outros 3 datasets por razões já anteriormente mencionadas, nomeadamente, o facto de os valores deste dataset serem menos fidedignos.

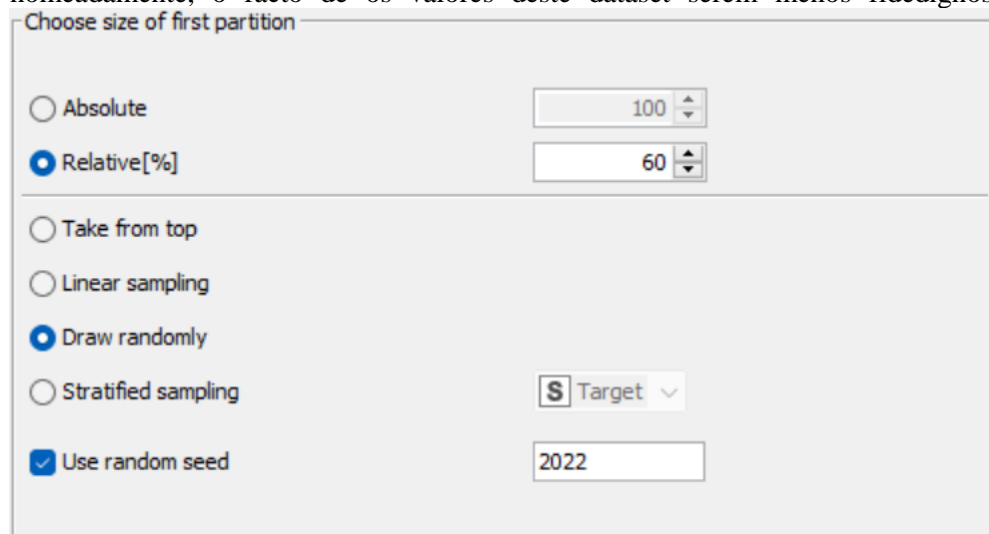


Figura 13 – Alteração ao nodo *Partitioning 2*

- *Concatenate*: este nodo foi usado para juntar os conjuntos de dados previamente particionados para teste, tanto do conjunto Cleveland, Long Beach e Suíça, como da Hungria.

Para todos os modelos usados havia duas instâncias de learning, uma para adaptar os dados relativos á Hungria, e outra que é mais capaz de fazer o diagnostico de doenças cardíacas. Há assim uma primeira partição dos dados de Cleveland, Long Beach e Suíça, escolhendo aleatoriamente 60% dos dados e usando-os como teste para adaptar os dados da Hungria, mais especificamente as linhas que têm um valor em Target igual a 1. De seguida no conjunto que faz o diagnostico, são concatenados todos os dados da Hungria e particionados numa taxa de 50% para teste e 50% para treino, havendo ainda os restantes 40% das outras localidades, sendo esta particionada com 80% para treino e 20% para teste.

5. Nodos de Aprendizagem

No método de clustering foram usados 3 diferentes nodos, o nodo *k-means*, o nodo *Cluster assigner*, e o nodo *Rule Engine*. Como foi dito anteriormente, neste método e com estes 3 blocos é possível dividir os dados que temos em vários agrupamentos de modo a juntar os que mais têm relação em si.

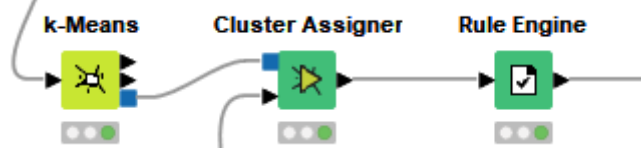


Figura 14 – Blocos associados ao método de Clustering

Assim no nodo do *k-Means* são definidos o número de clusters para este problema sendo no primeiro conjunto apenas 4 clusters, pois só testa de 1 a 4, para o segundo conjunto já irá ter 5 clusters, um para cada valor de target. Este nodo encontra o melhor método de agrupamento depois de receber os dados de treino.

Este método de agrupamento é seguido para o *Cluster Assigner* onde o utiliza para agrupar os dados de teste que recebe. Por fim o *Rule Engine* muda os nomes dos dados da coluna Cluster. Para o primeiro conjunto de segmentação muda de cluster_x para x+1, sendo x o valor associado ao cluster. Para o segundo conjunto, muda de cluster_x para x.

Quanto ao método da árvore de decisão foram usados 2 nodos distintos, *Decision Tree Learner* e o *Decision Tree Predictor*. Este técnico de aprendizagem como foi explicada anteriormente consiste na toma de decisões baseadas em conjuntos de regras. Ela é chamada de árvore de decisão porque ela possui uma estrutura semelhante a uma árvore, com vários nós internos que representam as decisões a serem tomadas e folhas que representam os resultados possíveis dessas decisões.

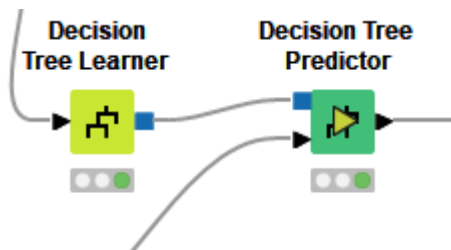


Figura 15 – Blocos associados ao método de Árvore de Decisão

Nestes nodos o *Decision Tree Learner* aprende com os dados de treino que nele entram a criar um conjunto de regras que mais se adequem ao nosso objetivo, isto é, adivinhar corretamente o resultado da coluna target. O *Decision Tree Predictor* usa o conjunto de regras criadas no nodo anterior e aplica aos dados de teste fazendo assim uma previsão.

Neste bloco foi utilizado ainda o pruning, isto é, uma técnica usada para reduzir o tamanho de um modelo treinado removendo parâmetros ou conexões desnecessárias. Esta técnica foi usada para tornar o processo mais rápido, e evitar o acontecimento de overfitting.

No método de Redes Neurais foram usados 2 nodos diferentes, *RProp MLP Learner* e *MultiLayerPerceptron Predictor*. Como já foi explicado o Multi-Layer Perceptron é um tipo de modelo de aprendizagem que é composto por uma rede de neurônios conectados. Cada neurônio em uma camada recebe entradas de neurônios anteriores, aplica uma função de ativação e passa o resultado para a próxima camada.

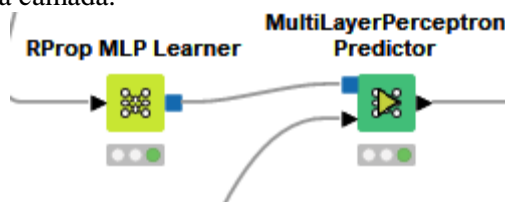


Figura 16 – Blocos associados ao método de Redes Neurais

No primeiro nodo haverá a aprendizagem de como cada neurónio irá se comportar, escolhendo a sua função, o que deverá receber e o que deverá enviar para o próximo. O MultiLayerPerceptron Predictor, recebe esta informação de como deverá tratar os dados e os utiliza para os novos dados que irá receber, isto é, os dados de teste.

Neste primeiro bloco tivemos de escolher e adaptar o número de neurónios por camada e o número de camadas para cada um dos diferentes Perceptrons usados, começando sempre por um número mais baixos, uma vez que estes dados são relativamente simples, mas aumentado os números até encontrar uma combinação que mais combina com caso em questão.

Por fim, depois de cada método usado para a aprendizagem completa dos dados, utilizamos o nodo *Scorer (JavaScript)* para nos apresentar assim a matriz de confusão final contendo toda a aprendizagem feita. Mostrando dados como a precisão geral dos dados, o erro geral, entre outros.

6.Análise de Resultados

Começamos então por analisar o comportamento dos três métodos por si, isto é, apenas o método da Segmentação, Árvore de Decisão e Redes Neurais. Assim obtivemos as seguintes matrizes de confusão:

Scorer View		K-means					
Confusion Matrix							
		0 (Predicted)	1 (Predicted)	2 (Predicted)	3 (Predicted)	4 (Predicted)	
0 (Actual)		36	39	2	4	16	37.11%
1 (Actual)		17	8	0	3	1	27.59%
2 (Actual)		2	7	3	5	11	10.71%
3 (Actual)		1	11	0	2	0	14.29%
4 (Actual)		1	0	0	0	0	0.00%
		63.16%	12.31%	60.00%	14.29%	0.00%	
Overall Statistics							
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified			
28.99%	71.01%	0.024	49	120			

Figura 17 – Matriz de confusão método de clustering

Scorer View		Decision Tree					
Confusion Matrix							
		0 (Predicted)	1 (Predicted)	2 (Predicted)	3 (Predicted)	4 (Predicted)	
0 (Actual)		109	6	0	0	0	94.78%
1 (Actual)		5	21	0	1	0	77.78%
2 (Actual)		4	13	1	0	0	5.56%
3 (Actual)		0	6	0	2	0	25.00%
4 (Actual)		0	0	1	0	0	0.00%
		92.37%	45.65%	50.00%	66.67%	undefined	
Overall Statistics							
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified			
78.70%	21.30%	0.556	133	36			

Figura 18 – Matriz de confusão método de Arvore de Decisão

Scorer View		Perceptron				
Confusion Matrix		0 (Predicted)	1 (Predicted)	2 (Predicted)	3 (Predicted)	4 (Predicted)
0 (Actual)		112	2	0	2	1
1 (Actual)		5	20	3	2	0
2 (Actual)		5	5	5	1	0
3 (Actual)		0	1	0	3	1
4 (Actual)		1	0	0	0	0
		91.06%	71.43%	62.50%	37.50%	0.00%
Overall Statistics		Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
		82.84%	17.16%	0.628	140	29

Figura 19 – Matriz de confusão método de Redes Neurais

Assim observa-se que de um modo geral, ao fazermos duas vezes o método de Redes Neurais, obtém-se um melhor resultado, uma vez que há uma maior precisão, cerca de 82.84% e este é o que tem maior coeficiente kappa de cohen. Apesar de a Árvore de Decisão não ficar muito longe da qualidade deste, notou-se que na coluna prevista para 4 este não previu nenhum paciente com um estado de doença mais grave.

Notou-se ainda que o K-mean comparativamente a estes dois métodos, teve uma performance muito pior com apenas 28.99% de precisão.

Então colocou-se uma questão: “Será que se unirmos os dois melhores métodos teremos um melhor resultado?”. Assim criamos mais dois workflows, um que começa com a árvore de decisão e acaba com o método de redes neurais, e outro que era exatamente o inverso. Nestes obteve-se as seguintes matrizes de confusão:

Scorer View		Decision Tree + Perceptron				
Confusion Matrix		0 (Predicted)	1 (Predicted)	2 (Predicted)	3 (Predicted)	4 (Predicted)
0 (Actual)		100	6	1	2	0
1 (Actual)		7	12	2	3	1
2 (Actual)		5	13	2	4	1
3 (Actual)		1	5	1	1	1
4 (Actual)		0	0	0	1	0
		88.50%	33.33%	33.33%	9.09%	0.00%
Overall Statistics		Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
		68.05%	31.95%	0.395	115	54

Figura 20 – Matriz de confusão método de árvore de Decisão + Redes Neurais

Scorer View		Perceptron+ Decision Tree				
Confusion Matrix		0 (Predicted)	1 (Predicted)	2 (Predicted)	3 (Predicted)	4 (Predicted)
0 (Actual)		108	8	0	0	0
1 (Actual)		5	25	0	4	0
2 (Actual)		0	7	0	7	0
3 (Actual)		1	2	0	1	0
4 (Actual)		0	1	0	0	0
		94.74%	58.14%	undefined	8.33%	undefined
Overall Statistics		Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
		79.29%	20.71%	0.572	134	35

Figura 21 – Matriz de confusão método de Redes Neurais + Arvore de Decisão

De um modo geral viu-se que estes dois novos modelos não conseguiram ultrapassar o modelo de Redes Neurais por si só. Mas notou-se também que o Redes Neurais + Árvore de Decisão superou o modelo onde apenas havia Árvore de Decisão.

Assim conseguimos chegar à conclusão de que dentro destes 5 modelos criados o que foi mais sucedido em acertar os diagnósticos dos pacientes foi o modelo de apenas Redes Neurais.

7. Conclusão

As inteligências artificiais têm o potencial de revolucionar o diagnóstico de doenças, fornecendo diagnósticos rápidos, precisos e acessíveis. Uma forma como a IA pode ser usada para este propósito é através do desenvolvimento de algoritmos de Machine Learning que podem analisar grandes quantidades de dados, incluindo imagens médicas e registos de pacientes, para identificar padrões e prever a probabilidade de um paciente ter uma condição específica. Isso pode ajudar os médicos a tomar decisões mais informadas sobre como tratar melhor seus pacientes e até mesmo permitir o rastreamento precoce de doenças cardíacas, o que pode ser fundamental para evitar complicações graves.

Com este trabalho podemos ver de uma maneira simplista como a inteligência artificial pode ajudar na evolução dos diagnósticos médicos, mais especificamente na área da cardiologia, é necessário sobressaltar o facto de que mesma com automatização da mesma ainda será necessário o auxílio médico para concluir este mesmo diagnostico uma vez que terá um erro associado.

Foi possível observar ainda que para cada diferente método associado implicará sempre um tratamento dos dados diferentes, bem como um resultado distinto sendo sempre necessário a análise profunda da base de dados, para que seja possível escolher assim o melhor método associado, e por sua vez o tratamento, de dados. Assim para esta base de dados nos fornecida o melhor método encontrado para a automatização dos diagnósticos de doenças cardíacas foi no que foram usados dois métodos de MultiLayeredPerceptron, com um erro de cerca de 17%.

8. Bibliografia

<https://medprev.online/blog/saude/frequencia-cardiaca/>

<https://www.hospitaldaluz.pt/pt/dicionario-de-saude/pressao-arterial-e-hipertensao>

<https://rstudio-pubs->

static.s3.amazonaws.com/535487_2eeab91f7445468b9c39a00705538525.html