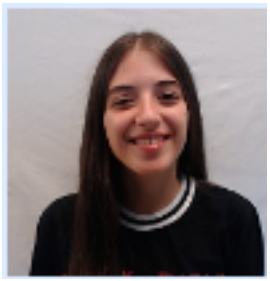
Universidade do Minho
Escola de Engenharia

Aprendizagem e Decisão Inteligentes
Grupo 8
2022/2023

Afonso Xavier Cardoso Marques a94940
Ana Filipa da Cunha Rebelo a90234
Cláudia Peixoto Silva a93177
Simão Paulo da Gama Castel-Branco e Brito a89482

Maio 2023






Índice

List of Figures

1 Introdução

O presente relatório enquadra-se na unidade curricular Aprendizagem e Decisão Inteligentes, na qual nos foi proposta a exploração, modelação e análise de dois datasets, através da plataforma KNIME.

O primeiro, escolhido pelo grupo recorrendo à plataforma Kaggle, focado na previsão do preço do diamante, tratando-se de um Problema de Regressão. O segundo, fornecido pelos professores, com o objetivo de antever o tipo de obesidade o que constitui um Problema de Classificação.

De modo a obtermos uma melhor compreensão, implementação e desenvolvimento dos datasets além de ajudar no planeamento dos mesmos, o grupo decidiu optar pela metodologia CRISP-DM. Esta metodologia é composta por seis etapas: estudar o negócio, estudar os dados, preparar os dados, modelar, avaliar e desenvolver.

2 Apresentação e Exploração dos Datasets

A primeira etapa para a concretização deste projeto consistiu no entendimento e exploração dos datasets, dado que estes podem conter dados incompletos, incoerentes ou errados. Esta etapa é importante para entender a preparação necessária para atingir o objetivo de cada um dos datasets além de compreender qual o método de avaliação do modelo mais apropriado. Para o efeito, inicialmente analisamos cada atributo dos datasets, recorrendo aos nodos de estatística que se encontram disponíveis no KNIME, e retirar algumas conclusões sobre a preparação de dados necessária.

Em seguida apresentamos os datasets, explicando cada atributo bem como os nodos utilizados para a exploração dos mesmos e o objetivo final pretendido.

2.1 Dataset Diamantes

Este dataset foi escolhido pelo grupo através da plataforma Kaggle. Efetivamente, este dataset foi usado com dois targets distintos. Inicialmente com o objetivo de prever os quilates nos diamantes sendo este um problema de regressão. De seguida utilizamos como target a qualidade de corte de cada um sendo este um problema de classificação.

O dataset apresenta 18 atributos e 53940 entries. Dos 10 atributos 3 são categóricos e 7 são numéricos. Apresentamos em seguida os atributos iniciais:

- price: preço em dólar americano
- carat: peso do diamante
- cut: qualidade do corte (Regular, Bom, Muito Bom, Premium, Ideal),determina o brilho
- color: cor do diamante, de J(pior) a D(melhor)
- clarity: medida de quão claro é o diamante(I1 (pior), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (melhor))
- depth: valor da profundidade do diamante
- table: Faceta plana em sua superfície a grande faceta de superfície plana que se pode ver quando se olha para o diamante de cima.
- x: comprimento em mm(0-10.74)
- y: largura em mm(0-58,9)
- z: profundidade em mm (0-31.8)

Tal como é possível verificar pela análise da figura 1, os nodos utilizados para a exploração do dataset foram o Data Explorer, Statistics, Box Plot e Bar Chart.

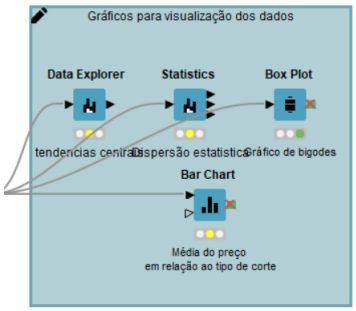


Figure 1: Exploração do dataset

A figura 2 apresenta o output do nodo Data Explorer para dados numéricos,este permitiu obter informação sobre os valores externos das diversas features,bem como a sua média,desvio padrão e a presença ou ausência de valores omissos(valores que devem ser tratados aquando da fase de pré-processamento).

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness
carat	☐	0.200	5.010	0.798	0.474	0.225	1.117
depth	☐	43	79	61.749	1.433	2.052	-0.082
table	☐	43	95	57.457	2.234	4.993	0.797
price	☐	326	18823	3932.800	3989.440	15915629.424	1.618
x	☐	0	10.740	5.731	1.122	1.258	0.379
y	☐	0	58.900	5.735	1.142	1.304	2.434
z	☐	0	31.800	3.539	0.706	0.498	1.522

Figure 2: Data Explorer

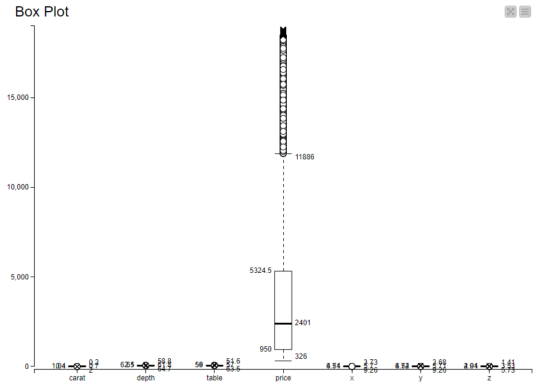


Figure 3: Box Plot

Através da análise de correlação apresentada na figura abaixo é possível medir a força e direção da associação entre duas variáveis,o que pode fornecer informação útil acerca das features que devem incorporar os modelos de aprendizagem automática. Tal como é possível verificar que existem features com um elevado grau de correlação (próximos de 1), tal como o par price-carat, o par price-x, price-y e price-z. É possível verificar também que o par table-depth tem uma baixa correlação,sendo esta correlação negativa. Importante salientar que este par é o único par que apresenta correlação negativa.

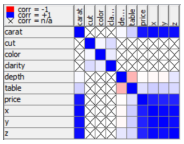


Figure 4: Linear Correlation

Através do nodo Statistics foram obtidas as estatísticas como a média, minimo, máximo e desvio padrão apresentadas na figura 5.

Através da figura 6 é possível verificar que o preço é mais elevado para o corte premium e mais baixo para o ideal.

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
carat	0,2	0,7979	?	5,01	0,474	1,1166	1,2566	0	0	0	
depth	43	61,7494	?	79	1,4326	-0,0823	5,7394	0	0	0	
table	43	57,4572	?	95	2,2345	0,7969	2,8019	0	0	0	
price	326	3 932,7997	?	18 823	3 989,4397	1,6184	2,1777	0	0	0	

Figure 5: Estatísticas relativas aos principais atributos

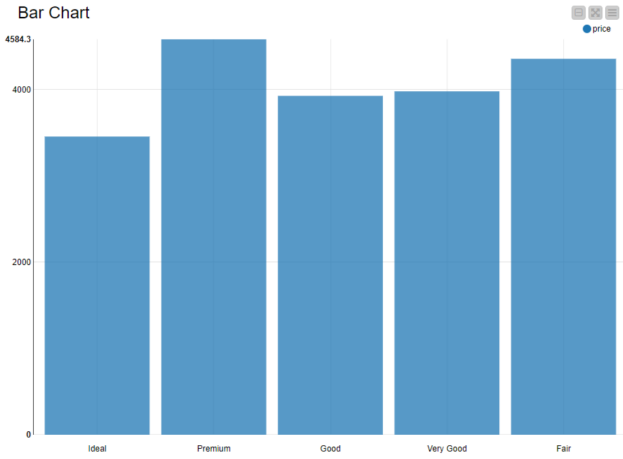


Figure 6: Média do preço em relação ao tipo de corte

2.2 Dataset Obesidade

Dataset fornecido pelos professores, que tem como objetivo prever o tipo de Obesidade. O dataset apresenta 18 atributos e 2111 entries. Dos 18 atributos 11 são categóricos e 7 são numéricos.

O dataset é constituído pelos seguintes atributos:

- rowID: id de registro
- Gender: sexo da pessoa
- Age: idade da pessoa
- Date_of_birth: data de nascimento (DD/MM/AAAA)
- Height: altura em metros
- Weight: peso em Kg
- family_history_with_overweight: histórico familiar de obesidade
- FAVC: Consumo frequente de alimentos altamente calóricos
- FCVC: Frequência de consumo de vegetais
- NCP: Número de refeições principais(1-4)
- CAEC: Consumo de alimentos entre as refeições
- SMOKE: a pessoa fuma
- CH20: Ingestão diária de água(1 = menos de um litro, 2 = 1–2 litros e 3 = mais de 2 litros)
- SCC: Monitoramento do consumo de calorias(Sim/Não)
- FAF: Frequência de atividade física (0 = nenhuma, 1 = 1 a 2 dias, 2 = 2 a 4 dias e 3 = 4 a 5 dias)
- TUE: Tempo usando dispositivos tecnológicos (0 = 0–2 horas, 1 = 3–5 horas e 2 = mais de 5 horas)
- CALC: Consumo de álcool
- MTRANS: Transporte utilizado ,Transporte Público, Mota, Bicicleta, Automóvel e Caminhada
- NObeyesdad: Nível de Obesidade

Tal como é possível verificar pela análise da figura abaixo para a Exploração de dados foram utilizados os nodos Sting To Number,Data Explorer,Statistics,Bar Chart e Box Plot, com objetivo de analisar e compreender o dataset.

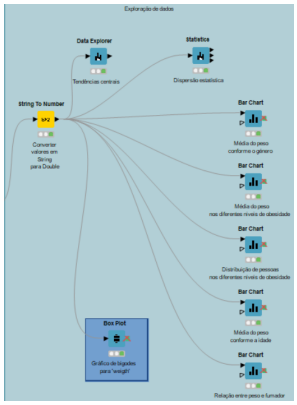


Figure 7: Exploração do dataset

De seguida apresentamos algumas imagens que ilustram alguns resultados da exploração de dados que foi feita com os nodos Bar Chart.

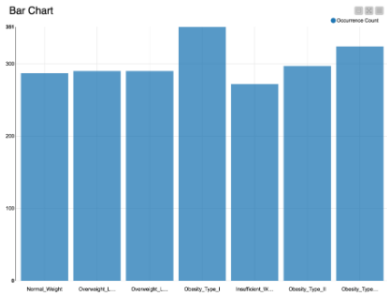


Figure 8: distribuição das pessoas nos diferentes níveis de obesidade

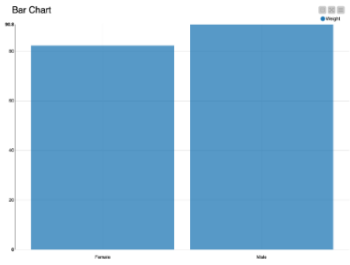


Figure 9: Média do peso conforme o género

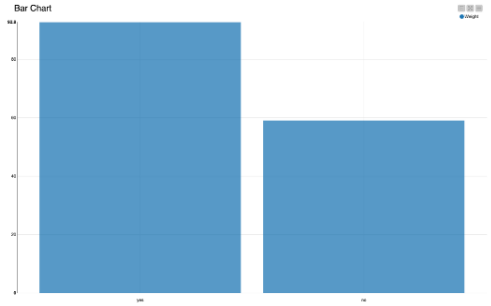


Figure 10: Média do peso conforme o histórico familiar de obesidade

Analisando as figuras 8,9 e 10 apresentadas é possível verificar que o tipo de obesidade da maioria das pessoas é obesity_type_I,o género masculino apresenta um número maior de obesidade e também que a média do peso é superior para pessoas com histórico familiar de obesidade.

Ao analisar a Matriz de Correlação apresentada na figura 11 foi possível inferir que caraterísticas como weight e height têm uma correlação positiva de 0.4631. Do mesmo modo, o family_history_with_overweight com NObeyesdad e Gender com NObeyesdad apresentam uma correlação de 0.5428 e 0.5582, respetivamente. Assim, é possível concluir que estes pares de atributos estão em grande parte a dar o mesmo conhecimento ao modelo sendo expectável que,por exemplo, o atributo NObeyesdad esteja intrinsecamente relacionado ao seu Gender.

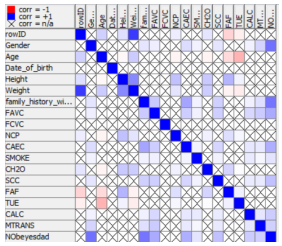


Figure 11: Linear Correlation

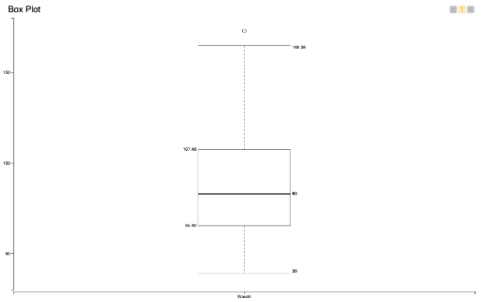


Figure 12: Deteção de outliers

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis
rowID	☐	1	2111	1056	609.538	371536	0	-1.200
Age	☐	14	61	24.313	6.346	40.271	1.529	2.826
Height	☐	1.450	1.980	1.702	0.093	0.009	-0.013	-0.563
Weight	☐	39	173	86.586	26.191	685.977	0.255	-0.700
NCP	☐	1	4	2.686	0.778	0.605	-1.107	0.386
CH2O	☐	1	3	2.008	0.613	0.376	-0.105	-0.879
FAF	☐	0	3	1.010	0.851	0.724	0.498	-0.621
TUE	☐	0	2	0.658	0.609	0.371	0.619	-0.549

Figure 13: Data Explorer

3 Preparação dos Datasets

Após uma análise completa efetuada para cada um dos datasets, segue-se a preparação dos dados com o objetivo de eliminar qualquer informação não pretendida.

Começamos por efetuar uma limpeza de dados, tratando dos missing values e outliers. De seguida é realizado um aumento da informação de acordo com o pretendido do problema. Deste modo, os datasets ficam prontos para a avaliação final do modelo desenhado. Apresentamos em seguida todo o tratamento de dados efetuado em cada um dos datasets:

3.1 Dataset Diamantes

Para uma melhor obtenção de resultados foram utilizados diversos modelos sendo efetuada uma preparação para cada um destes modelos. Em seguida, explicamos em mais detalhe a preparação feita para cada um destes modelos.

3.1.1 Logistic Regression

Para este modelo foram utilizados os seguintes nodos: Column Filter, Auto-binner, Normalizer, Numeric Outliers e Missing Values.

Efetivamente usamos o nodo column filter após verificarmos que havia uma grande correlação entre os atributos x, y, z e price. Após testar diferentes combinações (dando drop de x e z, drop de x, z e y, entre outras) foi possível verificar que, a que no fim apresentou melhor accuracy, foi quando se dava retirava a coluna do atributo z.

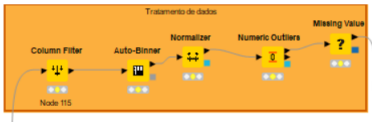


Figure 14: Preparação dos dados feita no modelo Logistic Regression

Foi testado também o nodo auto-binner que demonstrou influenciar o modelo de forma positiva fazendo bins de todos os atributos. A utilização do nodo Auto-Binner permite lidar com dados numéricos de forma mais eficiente e simplificada.

O nodo Normalizer foi utilizado para normalizar todos os dados numéricos no intervalo entre 0 e 1. Para o tratamento de outliers foi utilizado o nodo Numeric Outliers, tendo sido aplicado a todas as variáveis numéricas, de acordo com a seguinte estratégia: todos os outliers foram dados como missing values que posteriomente foram tratados. Quando apresentava um missing value inteiro ele punha o valor da média caso fosse uma string ele corrigia o valor.

3.1.2 Decision Tree

A preparação feita para este modelo foi a mesma do modelo Logistic Regression.

3.1.3 Clustering com Target

O preprocessing utilizado neste modelo foi igual ao do Logistic Regression pois foi o que apresentou os melhores resultados.No entanto, sendo que a decision tree já lida com os missing values não seria necessário implementar esse nodo.

Figure 15: Preparação dos dados feita no modelo Clustering com target

3.1.4 Redes Neuronais RProp

Para este modelo foram utilizados os nodos Missing Values, Parttioning,Normalizer, Numeric Outliers e Column filter. Uma vez que estes foram os nodos que apresentaram melhor desempenho relativamente a todos os nodos testados.

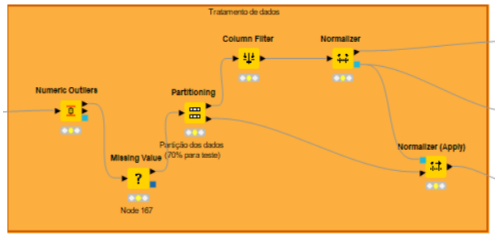


Figure 16: Preparação dos dados feita no modelo Redes Neuronis RProp

3.1.5 Linear Regression Learner

Para este modelo foram utilizados os nodos Column Filter, Auto-Binner, Normalizer, Numeric Outliers, Missing Values.

De facto voltamos a filtrar apenas a coluna z uma vez que foi ao filtrar apenas esta coluna que obtivemos o melhor resultado. A preparação feita para este modelo foi de facto muito parecida à do modelo Logistic Regression.

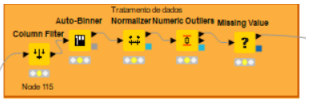


Figure 17: Preparação dos dados feita no modelo Linear Regression Learner

3.1.6 Simple Regression Tree Learner

Para este modelo foi utilizado apenas normalizer, numeric outlier e missing values.

Figure 18: Preparação dos dados feita no modelo Linear Regression Learner

Foram também testadas outras técnicas de pre-processing como filter-columns (onde se retiraram as colunas com um grande coeficiente de correlaçao) e o auto-binner mas não foram obtidos melhores resultados. Sendo que a decision tree já lida com os missing values não seria necessário implementar esse nodo.

3.2 Dataset Obesidade

A preparação deste dataset foi feita de diversas formas dependendo do modelo em questão. Para este dataset foram utilizados os seguintes modelos: Decision Tree (com e sem prunning), Logistic Regression Learner, Linear Regression Learner, Clustering e Redes Neuronais RProp. De seguida explicamos em mais detalhe a preparação feita para cada um dos modelos.

3.2.1 Decision Tree

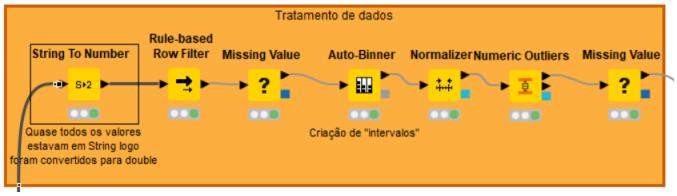


Figure 19: Preparação dos dados feita no modelo Decision Tree

A figura acima representada demonstra todos os nodos utilizados para o processamento e tratamento de dados com o modelo Decision Tree para o dataset da obesidade. Inicialmente convertemos todos os valores String para Double através do nodo String To Number. A limpeza inicial de dados centrou-se em remover:

- linhas com dados inválidos: o que se traduz na remoção de dados cujo o número de refeições seja inferior a 1 ou superior a 4,e também,dados cuja a ingestão de água seja inferior a 1 ou superior a 3
- colunas com missing values

A criação de intervalos para os atributos height e weight foi feita recorrendo ao nodo Auto-Binner. O tratamento de outliers foi efetuado com recurso ao nodo Numeric Outliers, tendo sido aplicado à coluna weight tendo sido utilizada a estratégia de substituir os valor do outlier por outro valor.

3.2.2 Logistic Regression Learner

A preparação feita para este modelo foi similar à anterior porém neste começamos por filtrar as colunas id,Date_of_birth e FCVC.

Figure 20: Preparação dos dados feita no modelo Logistic Regression Learner

3.2.3 Linear Regression Learning

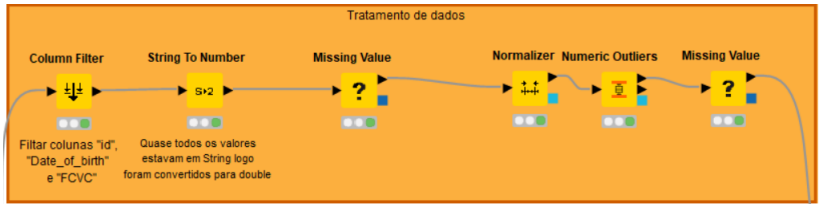


Figure 21: Preparação dos dados feita no modelo Logistic Regression Learning

A preparação efetuada para este modelo foi semelhante à anterior porém neste modelo não utilizamos o nodo Auto-Binner.

3.2.4 Clustering

Figure 22: Preparação dos dados feita no modelo Clustering

A preparação realizada para este modelo foi semelhante à anterior porém neste modelo no nodo Column Filter filtramos a coluna id.

3.2.5 Redes Neuronais RProp

Para este modelo começamos por dividir os dados em dois subconjuntos menores de modo a serem utilizados para diferentes fins recorrendo ao nodo Partitioning. Para o primeiro subconjunto utilizamos os nodos String To Number,Column Filter e Normalizer. Enquanto que no segundo subconjunto utilizamos apenas os nodos String To Number e Normalizer.

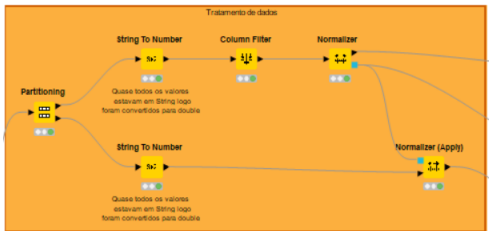


Figure 23: Preparação dos dados feita no modelo Redes Neuronais RProp

4 Modelos Desenvolvidos

Finalizada a preparação de dados dos datasets, passamos para o desenvolvimento dos modelos de aprendizagem. Deste modo, conseguimos testar todo o tratamento que foi realizado e explicado na secção anterior, dando uma resposta ao problema proposto para cada um dos datasets.

De seguida iremos apresentar os algoritmos testados,bem como as suas características e os parâmetros de treino utilizados.

4.1 Dataset Diamantes

4.1.1 Modelos de classificação

Todos os testes foram realizados utilizando holdout com o partitioning.Este selecionava 70% dos dados aleatoriamente para teste.

Logistic Regression Learner:

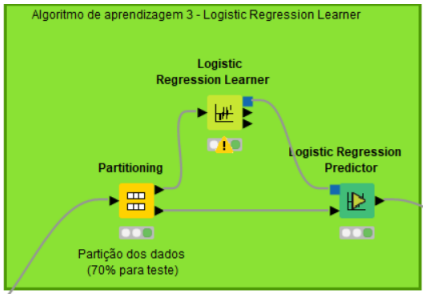


Figure 24: ML:M1

Uma vez que o target era cut foi utilizado a logisitic regression que é um algoritmo de classificação.

Decision Tree:

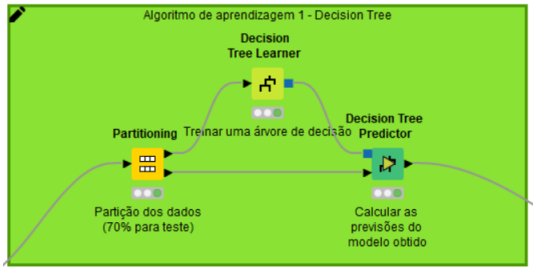


Figure 25: ML:M2

Uma vez que o target era cut foi utilizado a decision tree que é um algoritmo de classificação. Este foi utilzado com pruning de maneira a reduzir o over-fitting com o número minimo de records por nodo equivalente a dois e o número de divisões equivalente a 4.

Clustering com KMean:

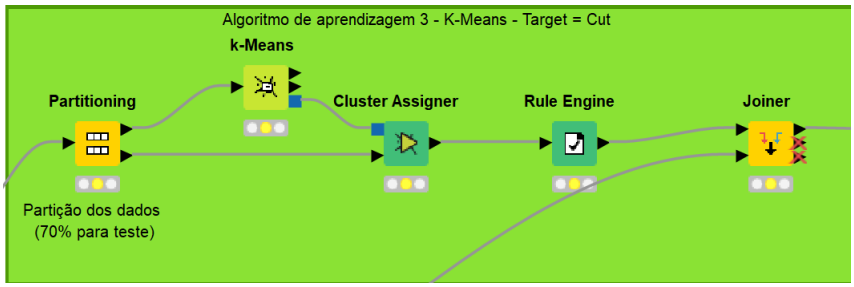


Figure 26: ML:M3

No k-means foram usados 5 clusters correspondentes ao tipo de corte que existiam. Posteriormente, foram também testados diferentes números máximos de iterações, no entanto, não se obteve grande diferença nos resultados obtidos.Através do rule engine atribuímos um nome a cada cluster correspondente ao corte:

```
$Cluster$ = "cluster_0" => "Ideal"
$Cluster$ = "cluster_1" => "Premium"
$Cluster$ = "cluster_2" =>"Good"
$Cluster$ = "cluster_3" =>"Very Good"
$Cluster$ = "cluster_4" =>"Fair"
```

Por fim, foi realizado um joiner que pega no dataset original e junta ao resultado devolvido pelo rule-engine.

4.1.2 Modelos de Regressão

Relativamente aos modelos de regressão utilizados para prever os quilates foi também usada uma estratégia holdout com o nodo partitioning.

Redes Neuronais RProp:

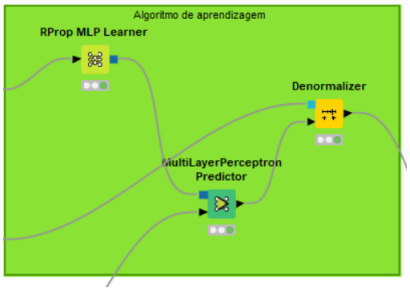


Figure 27: ML:M4

No RProp foi obtido o melhor resultado com o número máximo de iterações equivalente a 250 e com 3 hidden layers uma vez que a alteração destes valores não demonstrou mudar o erro final.

Linear Regression Learner:

Uma vez que o target são os quilates nos diamantes foram testados diversos algoritmos de regression.

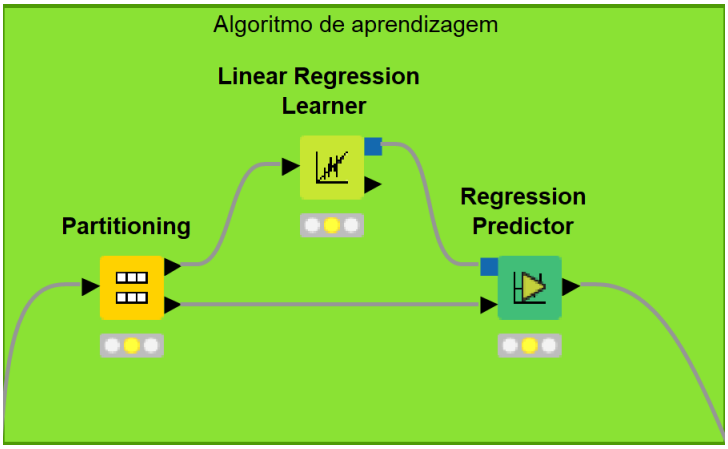


Figure 28: ML:M5

Simple Regression Tree Learner:

Para este algoritmo foi utilizado o XGBoost como maneira de lidar com os missing values .

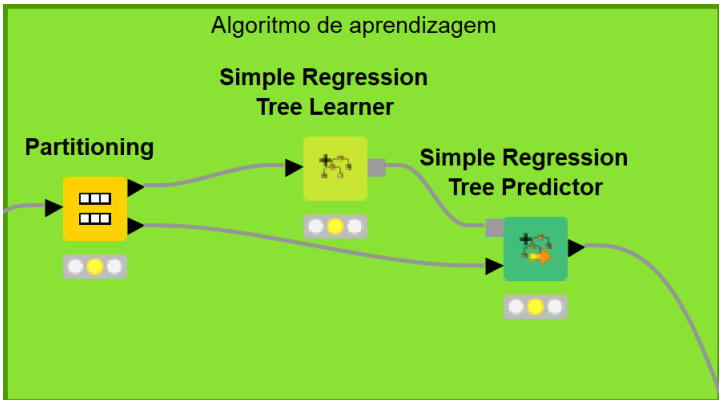


Figure 29: ML:M6

4.2 Dataset Obesidade

Decision Tree: algoritmo utilizado para categorizar ou prever o valor de uma variável, aprendendo regras de decisão simples inferidas através de dados anteriores (dados de treinamento). Uma Decision Tree é um grafo hierarquizado (árvore) em que cada ramo representa a seleção entre um conjunto de alternativas e as folhas representam uma decisão.

Este modelo foi implementado com e sem prunning.

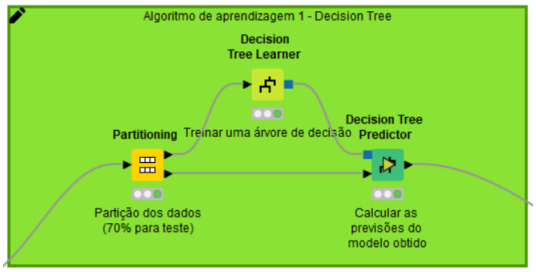


Figure 30: ML:M1

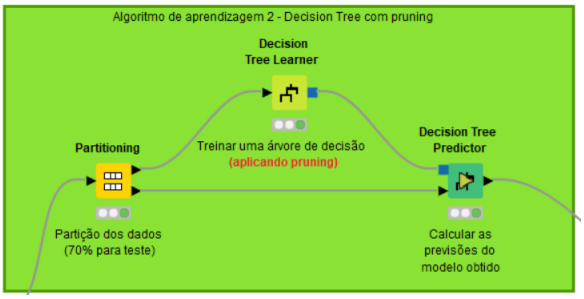


Figure 31: ML:M1 com Prunning

Para este modelo obtivemos uma precisão de 94.48% sem prunning e 93.38% com prunning o que nos permite concluir que este algoritmo é melhor sem prunning. Este modelo sem prunning foi o que apresentou o valor maior de accuracy e como tal vai ser o modelo escolhido para validação do modelo final.

Logistic Regression Learner: algoritmo de aprendizagem supervisionado que utiliza uma função logística para produzir um modelo com previsões de valores a partir de várias outras variáveis. É um modelo linear que utiliza a regressão logística para fazer a classificação dos dados em duas ou mais categorias.

Com este algoritmo obtivemos uma accuracy de 88.96%, este valor é inferior ao valor obtido no modelo anterior.

Linear Regression Learning: é um algoritmo de aprendizagem supervisionado que tem como objetivo estabelecer uma relação linear entre uma variável de saída (variável dependente) e uma ou mais variáveis de entrada (variáveis independentes).

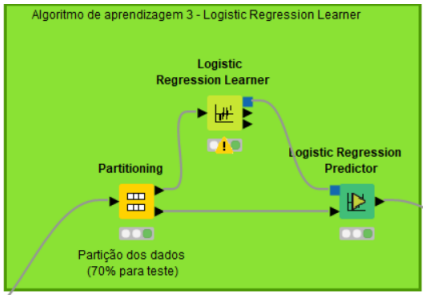


Figure 32: ML:M2

O algoritmo de regressão linear pode ser utilizado tanto para fins de previsão (ou seja, para estimar o valor da variável de saída para uma dada entrada) quanto para análise de correlação (ou seja, para avaliar a força da relação entre as variáveis).

Existem duas formas principais de regressão linear: a regressão linear simples e a regressão linear múltipla. Na regressão linear simples, existe apenas uma variável preditora, enquanto na regressão linear múltipla, existem duas ou mais variáveis preditoras. No nosso modelo utilizamos a regressão linear simples.

Este tipo de modelo é mais adequado para problemas de regressão e por isso o grupo optou por apenas incluir um modelo desta natureza num dataset que é claramente de classificação. Foi necessário mudar o target de NObeyesdad para Weigth e obtivemos os seguintes resultados:

- R-Square = 0.959
- Mean Absolute Error (MAE) = 0,026
- Mean Squared Error (MSE) = 0,002
- Root Mean Squared Error (RMSE) = 0,039

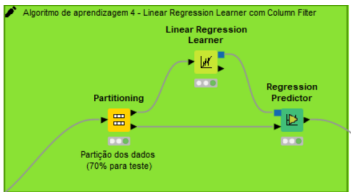


Figure 33: ML:M3

Clustering: o clustering de dados é um processo através do qual se particiona um conjunto de dados em segmentos/clusters de menor dimensão, que agrupam conjuntos de dados similares.

Um segmento/cluster é uma coleção de valores/objetos que:

- São similares entre si, dentro de um mesmo segmento;
- São diferentes dos valores/ objetos de outros segmentos.

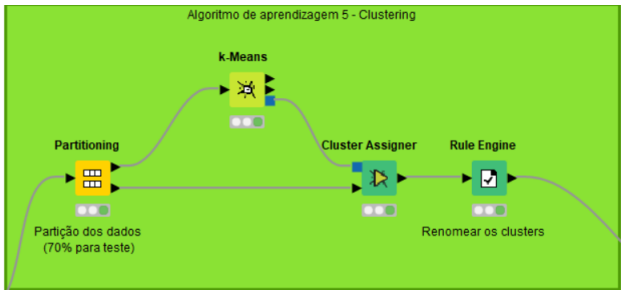


Figure 34: ML:M4

Este algoritmo obteve uma accuracy de 13.25%. Comparando com os outros valores obtidos, este é o valor mais baixo por uma diferença bastante substancial.

Redes Neuronais RProp: é um algoritmo de otimização usado em redes neuronais artificiais para treinamento supervisionado. O objetivo do RProp é ajustar os pesos sinápticos da rede neuronal de forma a minimizar uma função de custo, que representa a diferença entre a saída prevista e a saída real da rede para um determinado conjunto de dados.

A técnica RProp é uma variação do algoritmo de retropropagação (backpropagation), que é um dos métodos mais populares para treinar redes neurais. O algoritmo RProp utiliza um conjunto de regras de atualização de peso baseado no sinal do gradiente da função de custo. A principal vantagem do RProp em relação ao algoritmo de retropropagação clássico é que ele ajusta os pesos de forma mais rápida e eficiente, mesmo quando a superfície da função de custo é muito complexa.

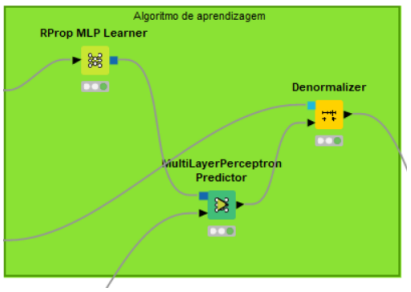


Figure 35: ML:M5

Para este algoritmo obtivemos uma accuracy de 71.61% que é um valor bastante superior ao valor obtido no modelo anterior e inferior aos restantes modelos.

5 Resultados Finais e Análise Crítica

Tendo todos os modelos de aprendizagem testados,apresentamos agora o algoritmo final escolhido com os resultados obtidos para ambos os datasets. Apresentamos ainda uma análise crítica a esses mesmos resultados.

5.1 Dataset Diamantes

Na seguinte tabela temos um sumário dos valores registados para a accuracy dos modelos de Machine Learning previamente explicados e que foram aplicados ao dataset A - Diamantes.

Modelos	**Accuracy**
Logistic Regression Learner	65,98%
Decision Tree	76,25%
Clustering com KMeans	17,32%

Table 1: Tabela de resultados de Classificação do dataset A

Modelos	**MAE**	**MSE**	**RMSE**
Redes Neuronais com RPRop	0,712	0,676	0,82
Linear Regression Learner	0,035	0,003	0,05
Simple Regression Tree Learner	0,009	0	0,019

Table 2: Tabela de resultados de Regressão do dataset A

Após analisarmos os resultados obtidos é possível concluir que o algoritmo de classificação que obteve um melhor desempenho foi o Decision tree. No entanto, era espectável que o clustering k-means obtivesse um valor bastante superior a 17,32%. Tal deve acontecer devido a atribuição dos diferentes valores a clusters que não seria suposto , no entanto, em todos os testes apresentados não foram obtidos melhores resultados. Relativamente aos algoritmos de regression onde o target são os quilates de um diamante os mesmos apresentaram melhores resultados com um menor erro no algoritmo Simple Regression Tree Learner.Efetivamente era espectável que as redes neuronais obtivessem um melhor desempenho. O gráficos demostram o desempenho do algoritmo k-means relativamente aos clusters e ao valores cut.

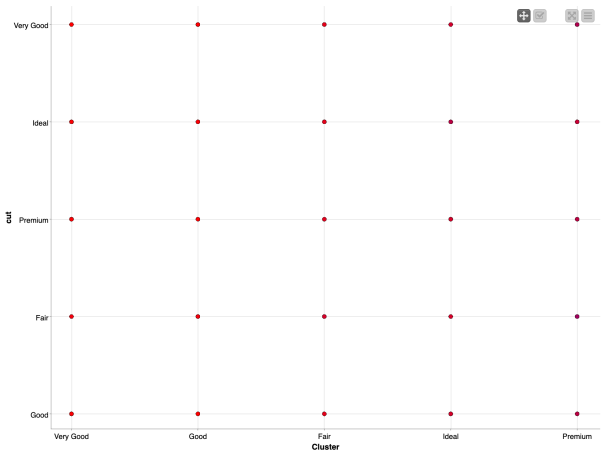


Figure 36: Scatter k-means

Após a análise do gráfico é possível verificar que os pontos se encontram bastante dispersos relativamente aos clusters uma vez que o número dos clusters equivale ao nome do cut. No entanto, é possível verificar que há cuts que não se encontram no cluster apropriado pois o espectável seria cada cluster ter apenas elementos com um certo cut como ,por exemplo, o cut Good teria apenas o cut good. No entanto, não conseguimos perceber porque isto acontece.

O gráfico seguinte demonstra o comportamento do algoritmo R-prop relativamente ao valor previsto do cut e ao cut real.

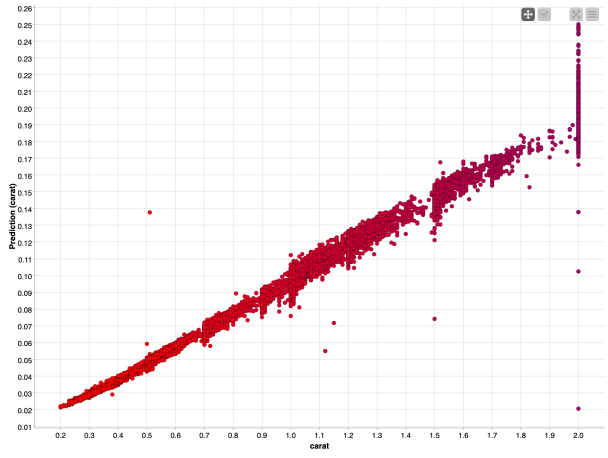


Figure 37: Rprop diamantes

A direção do relacionamento entre as variáveis pode ser avaliada pela inclinação da linha de tendência no gráfico. Se a linha de tendência estiver inclinada para cima da esquerda para a direita, isso sugere que as duas variáveis têm uma relação positiva - ou seja, quando uma variável aumenta, a outra também tende a aumentar.

Além disso, é importante avaliar a dispersão dos pontos no gráfico. Como os pontos não estão bastante dispersos,

isso pode indicar que há uma relação entre as variáveis.

5.2 Dataset Obesidade

Na seguinte tabela temos um sumário dos valores registados para a accuracy dos modelos de Machine Learning previamente explicados e que foram aplicados ao dataset B - Obesidade. Optamos por não incluir o modelo de regressão linear (Linear Regression Learner) pois as métricas de comparação não são as mesmas que os restantes modelos.

Modelos	**Accuracy**
Decision Tree	94,48%
Decision Tree com prunnig	93,38%
Logistic Regression Learner	88,96%
Clustering com KMeans	13,25%
Redes Neuronais com RPRop	71,61%

Table 3: Tabela de resultados do dataset B

Num panorama geral o melhor modelo foi o de Decision Tree sem prunning sendo o pior o de Clustering. Estavamos à espera de obter melhores resultados no Clustering e nas redes neuronais, e acreditamos que a razão para estes valores menos ideais esteja relacionada com o pré-processamento de dados que foi aplicado. Fizemos um esforço para não alterar muito o dataset e de certa forma isso reflete-se no resultado final. Através dos gráficos seguintes é possível constatar alguns aspetos relativamente ao modelo RProp e ao k-means:

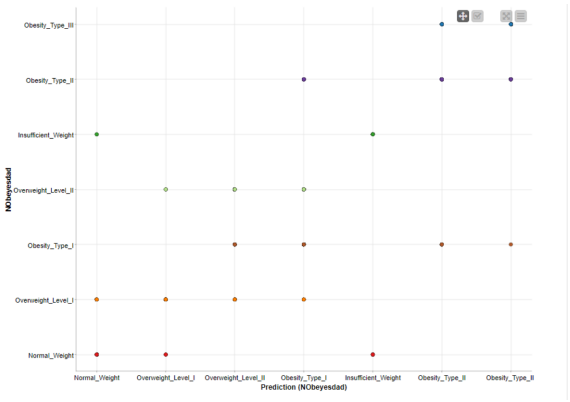


Figure 38: Scatter Plot RProp

Após uma análise ao gráfico vê-se que não ha uma relação clara entre os clusters e o tipo de obesidade.

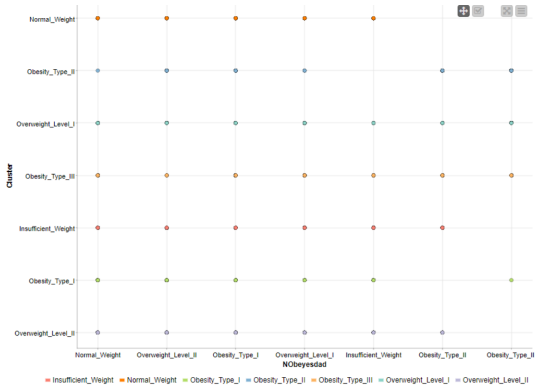


Figure 39: Scatter kmeans

Após uma análise ao gráfico vê-se que não ha uma relação clara entre os valores previstos e o tipo de obesidade.

6 Sugestões e Recomendações

Por fim, consideramos ser necessário realizar uma análise dos resultados obtidos no contexto do problema de cada um dos datasets e aos modelos finais desenvolvidos. Deste modo,conseguimos dar algumas sugestões e recomendações sobre como obter os melhores resultados para os problemas em questão.

A mais óbvia das melhorias que podiam ser feitas são nos algoritmos de aprendizagem com clustering (KMeans) nas duas tarefas. Inicialmente, decidimos experimentar vários valores para iterações no nodo de KMeans mas não teve um impacto significativo nos resultados finais. De seguida, experimentamos fazer pequenas alterações no pré-processamento dos dados, mas os resultados ou tinham valores muito baixos ou não tinham valores nenhuns devido a erros provocados pelas alterações feitas. Decidimos que no final seria mais sensato manter os valores baixos mas "estáveis" que tinhamos obtido. Acreditamos que talvez a razão dos resultados baixos para estes dois modelos de aprendizagem esteja relacionada com a renomeação dos clusters no nodo de Rule Engine.

Outro aspeto do nosso trabalho que admitimos que poderia precisar de melhorias é o nivel de complexidade que se pretendeu atingir. Tentamos não complicar muito e, dependendo do ponto de vista, isso pode ser uma coisa positiva ou negativa. É certo que se tivessemos aumentado mais um bocado a complexidade dos modelos, poderiamos ter tido resultados melhores.

Em último lugar a escolha do dataset para a tarefa A poderia ter sido outra. Talvez um dataset com aspetos mais variados que resultariam em gráficos mais interessantes ou então um dataset dividido em multiplos ficheiros que nós teriamos de agrupar.

7 Conclusão

Dado por concluído o trabalho prático, consideramos importante realizar uma análise crítica, e ainda, realçar os pontos positivos e negativos do trabalho realizado.

A extensa preparação de dados elaborada em ambos os datasets e a boa organização dos modelos desenvolvidos com alguns nodos comentados salienta o bom entendimento dos problemas com o objetivo de melhorar o resultado final. O grupo ter optado por desenvolver diversos modelos para testagem de resultados de modo a obter os melhores resultados constitui um ponto positivo do nosso trabalho.

Por fim, o grupo considera que o trabalho realizado é bastante positivo, pois cumpre todos os requisitos propostos no enunciado.