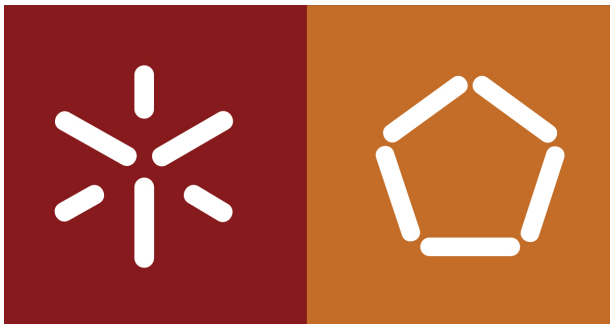


**Universidade do Minho**
**Licenciatura em Engenharia Biomédica**

# Aprendizagem Automática

**Inteligência Artificial em Engenharia Biomédica**
**Ano Letivo 2025/2026**

Ana Filipa Figueiredo – a107239

Dinis Rosa – a107159

Duarte Franco – a107233

Braga, 5 de janeiro de 2026

# 1 Resumo

Este projeto teve como objetivo implementar um sistema de aprendizagem automática (*ML - Machine Learning*) para prever o tipo de diabetes em indivíduos, com base em parâmetros clínicos como glicose, colesterol, pressão arterial e características antropométricas. Recorrendo à plataforma *KNIME* para criar modelos de classificação e prever a condição diabética, desenvolvemos um *pipeline* de ML.

Após uma análise estatística inicial do conjunto de dados e a aplicação de pré-processamento e tratamento dos dados, foram criados diversos modelos de classificação. Com base nos resultados obtidos do desempenho dos modelos, selecionamos o com melhor desempenho.

Como resultado deste trabalho, destacam-se não apenas as aprendizagens relacionadas ao processo de análise de dados e criação de modelos de classificação, mas também a experiência na utilização de plataformas de ML e a compreensão dos fatores de risco associados à diabetes.

# Conteúdo

**1 Resumo** **1**

**2 Introdução** **3**

**3 Preliminares** **4**
3.1 Inteligência Artificial e *Machine Learning* . . . . . 4
3.1.1 Aprendizagem Supervisionada . . . . . 4
3.1.2 Aprendizagem Não Supervisionada . . . . . 5
3.1.3 Aprendizagem por Reforço . . . . . 6
3.2 Plataforma *KNIME* . . . . . 6
3.3 Métricas de Qualidade e Avaliação do Modelo . . . . . 6
3.3.1 Métricas para Classificação . . . . . 6
3.3.2 Métricas para Regressão . . . . . 8

**4 Desenvolvimento do Sistema de Aprendizagem** **9**
4.1 Base de conhecimento . . . . . 9
4.2 Carregamento dos dados . . . . . 10
4.3 Exploração e análise de dados . . . . . 11
4.4 Pré-processamento dos dados . . . . . 12
4.5 Processamento de Dados . . . . . 21
4.6 Visualização dos Resultados . . . . . 23

**5 Modelos de Aprendizagem** **25**
5.1 Redes Neuronais (RProp) . . . . . 25
5.2 Árvore de Decisão . . . . . 27
5.3 Segmentação *K-Means* . . . . . 28
5.4 Regressão Logística . . . . . 30
5.5 Random Forest . . . . . 31

**6 Desenvolvimento dos algoritmos de aprendizagem** **34**
6.1 Aprendizagem supervisionada . . . . . 34
6.2 Aprendizagem não supervisionada . . . . . 38

**7 Comparação entre Modelo Multiclasse e Modelo Binário** **39**
7.1 Comparação do desempenho das Redes Neuronais (RProp): classificação binária vs multiclasse, com e sem *threshold* . . . . . 40
7.2 Comparação do desempenho das Árvores de Decisão: classificação binária vs multiclasse, com e sem *threshold* . . . . . 41
7.3 Comparação do desempenho da Regressão Logística: classificação binária vs multiclasse, com e sem *threshold* . . . . . 41
7.4 Comparação do desempenho do modelo Random Forest: multiclasse vs binário, com e sem *threshold* . . . . . 42

**8 Seleção do algoritmo mais preciso** **43**

**9 Conclusão** **44**

# 2 Introdução

A diabetes é uma doença endócrina crónica caracterizada pelo aumento dos níveis de glicose no sangue, afetando milhões de pessoas a nível mundial. A prevalência crescente das diabetes, especialmente da diabetes tipo 2, representa um desafio significativo para os sistemas de saúde, sendo associada a complicações graves como doenças cardiovasculares, nefropatia e retinopatia. A deteção precoce e a classificação precisa do estado diabético são fundamentais para implementar intervenções terapêuticas de modo a prevenir complicações.

Tradicionalmente, o diagnóstico da diabetes baseia-se em testes laboratoriais específicos como a medição de hemoglobina glicada (HbA1c) e glicose em jejum. No entanto, a integração de múltiplos parâmetros clínicos, antropométricos e metabólicos pode potencializar a capacidade de prever e permitir uma melhor estratificação de risco populacional.

As técnicas de *machine learning* emergiram como ferramentas poderosas na análise de dados clínicos complexos, permitindo identificar padrões e relações não-lineares entre variáveis que podem não ser evidentes através de métodos estatísticos tradicionais. A utilização de plataformas como o *KNIME Analytics Platform* oferece uma abordagem visual e intuitiva para construção de *workflows* de *machine learning*, tornando estas técnicas acessíveis na investigação biomédica.

O objetivo deste projeto consiste em desenvolver e validar modelos de classificação para previsão de diabetes utilizando *machine learning*, avaliando qual o algoritmo com melhor desempenho na distinção entre indivíduos *standard*, sem diabetes, e indivíduos em condição diabética (pré-diabetes e diabetes). Este relatório apresenta a metodologia empregada, os resultados obtidos e as principais conclusões deste estudo.

# 3 Preliminares

## 3.1 Inteligência Artificial e *Machine Learning*

A Inteligência Artificial (IA) é uma área da ciência da computação dedicada ao desenvolvimento de sistemas computacionais capazes de simular comportamentos inteligentes observados em seres racionais, humanos ou animais. Estes sistemas têm como objetivo apoiar ou automatizar processos de decisão, executando tarefas como classificação, previsão, segmentação de dados (*clustering*), associação, otimização, raciocínio baseado em casos e descoberta de padrões em grandes volumes de dados.

Os algoritmos de IA podem ser organizados segundo diferentes paradigmas, nomeadamente o paradigma estatístico, o simbólico (como Árvores de Decisão e métodos de representação do conhecimento e raciocínio), o conexionista (Redes Neuronais Artificiais), o evolutivo (Algoritmos Genéticos) e o paradigma baseado em casos. A aplicação destes paradigmas permite desenvolver modelos capazes de aprender a partir de dados históricos e generalizar esse conhecimento a novas situações.

A Aprendizagem Automática, ou *Machine Learning*, é um subcampo da IA que se foca na criação de algoritmos capazes de aprender automaticamente a partir de dados, sem necessidade de programação explícita. Estes algoritmos são tipicamente *data-driven*, ou seja, constroem modelos com base em exemplos, permitindo identificar padrões, efetuar previsões e apoiar a tomada de decisão. O ML tem demonstrado grande eficácia em áreas como a Engenharia Biomédica, particularmente na análise de dados clínicos e na previsão de doenças.

### 3.1.1 Aprendizagem Supervisionada

A aprendizagem supervisionada baseia-se na utilização de conjuntos de dados rotulados, nos quais existe informação conhecida sobre o resultado pretendido, também designado por atributo classe ou *target*. O processo de aprendizagem envolve a divisão do *dataset* em subconjuntos de treino e teste, sendo o primeiro utilizado para ensinar o modelo e o segundo para avaliar o seu desempenho.

Este tipo de aprendizagem subdivide-se em dois grandes grupos: classificação e regressão. Os problemas de classificação estão associados a respostas qualitativas e têm como objetivo identificar a classe a que pertence uma determinada instância. Por sua vez, os problemas de regressão lidam com respostas quantitativas, tendo como principal finalidade a previsão de valores numéricos contínuos.

Neste trabalho, serão utilizados algoritmos de aprendizagem supervisionada amplamente reconhecidos, nomeadamente Árvores de Decisão, *Random Forest*, Redes Neuronais Artificiais e Regressão Logística, aplicados a problemas de classificação no contexto biomédico.

#### Árvores de Decisão

As Árvores de Decisão são algoritmos de aprendizagem supervisionada utilizados tanto em classificação como em regressão. Estes modelos assumem uma estrutura hierárquica em forma de grafo, onde cada nodo interno representa um teste sobre um atributo, cada ramo corresponde a uma decisão possível e cada folha representa a classe ou valor final previsto.

O processo de construção de uma árvore inicia-se com a seleção do atributo raiz, baseada no conceito de entropia, que mede o grau de impureza de um conjunto de dados.

O atributo que apresenta maior ganho de informação, ou seja, maior redução da entropia, é escolhido como raiz da árvore. A partir deste nodo, o conjunto de dados é sucessivamente dividido até que se obtenha uma estrutura capaz de realizar previsões.

As Árvores de Decisão destacam-se pela sua interpretabilidade e facilidade de utilização, sendo particularmente úteis em problemas de classificação discreta. No entanto, podem apresentar limitações em cenários com grande complexidade ou elevada interação entre atributos.

***Random Forest***

O algoritmo *Random Forest* baseia-se na combinação de múltiplas Árvores de Decisão que operam de forma independente. Cada árvore realiza uma previsão individual, sendo o resultado final determinado pela votação maioritária entre todas as árvores do conjunto.

Uma das principais vantagens deste método é a sua robustez face ao *overfitting*, uma vez que os erros cometidos por árvores individuais tendem a ser compensados pelo conjunto. A *Random Forest* apresenta bom desempenho mesmo em *datasets* de grande dimensão ou com valores em falta, tornando-se uma técnica amplamente utilizada em problemas reais de classificação.

**Redes Neuronais Artificiais**

As Redes Neuronais Artificiais (RNA) são modelos computacionais inspirados no funcionamento do sistema nervoso central. Estas redes são constituídas por neurónios artificiais interligados por sinapses, organizados em camadas, sendo capazes de aprender através do ajuste dos pesos das ligações.

Cada neurónio processa as entradas recebidas, produzindo uma saída com base numa função de ativação. A aprendizagem ocorre através da adaptação dos pesos sinápticos ao longo do tempo, permitindo à rede generalizar o conhecimento adquirido e responder eficazmente a novos dados. As RNA são particularmente adequadas para problemas complexos e não lineares, sendo amplamente utilizadas em aplicações de *Deep Learning*.

**Regressão Logística**

A Regressão Logística é um algoritmo de aprendizagem supervisionada utilizado em problemas de classificação. Este modelo estima a probabilidade de uma instância pertencer a uma determinada classe através da função logística, garantindo que o valor previsto encontra-se no intervalo [0,1].

Apesar do nome, a regressão logística é uma técnica de classificação e é frequentemente aplicada em contextos biomédicos, como a previsão da presença ou ausência de uma doença.

### 3.1.2 Aprendizagem Não Supervisionada

Na aprendizagem não supervisionada, os dados utilizados não se encontram identificados, não existindo informação prévia sobre os resultados pretendidos. O objetivo do algoritmo é identificar padrões, estruturas ou relações ocultas nos dados, modelando a sua distribuição.

Este tipo de aprendizagem é particularmente útil em cenários onde o conhecimento prévio sobre os dados é reduzido. As principais categorias da aprendizagem não supervisionada são a segmentação (*clustering*) e a aprendizagem por associação.

**Segmentação (*Clustering*)**

A segmentação consiste na divisão de um conjunto de dados em grupos ou *clusters*, de forma a maximizar a semelhança entre instâncias do mesmo grupo e minimizar a semelhança entre grupos diferentes. Esta técnica é amplamente utilizada na análise exploratória de dados.

### 3.1.3 Aprendizagem por Reforço

A aprendizagem por reforço é um paradigma no qual o sistema aprende através de um processo de tentativa e erro, recebendo recompensas ou penalizações em função das ações realizadas. O objetivo é maximizar a recompensa total ao longo do tempo.

Embora não exista informação explícita sobre o resultado correto, este tipo de aprendizagem permite avaliar a qualidade das decisões tomadas, sendo utilizado sobretudo em problemas de otimização e controlo.

## 3.2 Plataforma *KNIME*

O *KNIME* é uma plataforma de análise de dados de código aberto que permite o desenvolvimento de modelos de ciência de dados através de uma interface gráfica intuitiva. A criação de soluções é realizada através de *workflows*, compostos por *pipelines* de dados constituídos por nodos interligados.

Cada nodo executa uma tarefa específica, como leitura e escrita de dados, transformação, exploração, treino de modelos ou visualização de resultados. A abordagem visual do *KNIME* facilita o desenvolvimento e a compreensão dos modelos, sendo particularmente adequada para aplicações em Inteligência Artificial e Engenharia Biomédica.

## 3.3 Métricas de Qualidade e Avaliação do Modelo

A avaliação do desempenho de um modelo de aprendizagem automática é uma etapa fundamental no desenvolvimento de sistemas de Inteligência Artificial, pois permite analisar a sua capacidade de generalização e a fiabilidade das previsões. As métricas de qualidade dependem do tipo de problema abordado, sendo diferentes para tarefas de classificação e regressão.

No contexto do presente trabalho, serão desenvolvidos modelos de classificação, sendo necessário utilizar métricas específicas. Em aplicações biomédicas, a escolha adequada das métricas é crucial, uma vez que erros de previsão podem ter impacto significativo nos diagnósticos e decisões clínicas.

### 3.3.1 Métricas para Classificação

**Matriz de Confusão**

A matriz de confusão é uma ferramenta essencial para avaliar modelos de classificação. Ela compara os valores previstos pelo modelo com os valores reais, permitindo identificar:

- **Verdadeiros Positivos (TP)**: casos corretamente classificados como positivos.
- **Verdadeiros Negativos (TN)**: casos corretamente classificados como negativos.
- **Falsos Positivos (FP)**: casos classificados como positivos, quando na realidade são negativos.

- **Falsos Negativos (FN)**: casos classificados como negativos, quando na realidade são positivos.

A partir destes valores, podem ser calculadas diversas métricas de desempenho.

***Accuracy***

A *Accuracy* mede a proporção de previsões corretas em relação ao total de observações:

$$\textbf{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Embora seja intuitiva, a *Accuracy* pode ser enganadora em conjuntos de dados não equilibrados, sendo recomendada a sua análise em conjunto com outras métricas.

**Precisão (*Precision*)**

Indica a proporção de previsões positivas que são realmente corretas:

$$\textbf{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

**Sensibilidade (*Recall*)**

Avalia a capacidade do modelo em identificar corretamente os casos positivos:

$$\textbf{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

**Especificidade**

Mede a capacidade do modelo em identificar corretamente os casos negativos:

$$\textbf{Especificidade} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

***F-measure (F1-score)***

Combina precisão e sensibilidade através da média harmónica, sendo útil quando é necessário equilibrar falsos positivos e falsos negativos:

$$\textbf{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

**Curva *ROC* e *AUC***

A curva ROC (*Receiver Operating Characteristic*) relaciona a taxa de verdadeiros positivos com a taxa de falsos positivos para diferentes thresholds. A área sob a curva (AUC) fornece uma medida global da capacidade discriminativa do modelo,variando entre 0,5 (classificação aleatória) e 1 (classificador perfeito).

**Coeficiente de Cohen (*Cohen's Kappa*)**

Avalia a concordância entre as previsões do modelo e os valores reais, considerando a concordância que ocorreria por acaso. Valores próximos de 1 indicam elevada concordância, enquanto valores próximos de 0 indicam concordância fraca.

### 3.3.2 Métricas para Regressão

Para modelos de regressão, que prevêem valores contínuos, são utilizadas métricas diferentes:

- **Erro Médio Absoluto (MAE):** média das diferenças absolutas entre os valores previstos e os reais.
- **Erro Quadrático Médio (MSE):** média dos quadrados das diferenças entre valores previstos e reais, penalizando erros maiores.
- **Raiz do Erro Quadrático Médio (RMSE):** raiz quadrada do MSE, mantendo a mesma unidade dos dados originais.
- **$R^2$ (Coeficiente de Determinação):** indica a proporção da variância dos dados explicada pelo modelo, variando entre 0 e 1, sendo valores próximos de 1 indicativos de bom ajuste.

# 4 Desenvolvimento do Sistema de Aprendizagem

## 4.1 Base de conhecimento

O sistema foi implementado recorrendo à plataforma KNIME e baseia-se na construção, treino e validação de modelos de classificação supervisionada e não supervisionada. Os dados utilizados dizem respeito a um estudo sobre a diabetes e incluem diversas medições clínicas e demográficas dos indivíduos, sendo o atributo target utilizado para classificar os utentes quando estes tem valores equivalentes a standard, pré-diabetes e diabetes.

- **Number**: Id do paciente;
- **Cholesterol**: Colesterol;
- **Stab**.glucose: Glicose Estabilizada;
- **Hdl**: Colesterol de Lipoproteínas de Alta Densidade;
- **Ratio_target**: Estado de suade do utente (Standard, Prediabetes e Diabetes);
- **Glyhb**: Hemoglobina Glicada – HbA1c;
- **Location**: Localização (Louisa, Buckingham);
- **Age**: Idade em anos do paciente;
- **Year**: Ano de nascimento;
- **Month**: Mês de nascimento;
- **Day**: Dia de nascimento;
- **Gender**: Género (Male ou Female);
- **Height**: Altura em centímetros;
- **Weight**: Peso em quilogramas
- **Frame**: Tipo de estrutura corporal (Small, Medion e Large);
- **Bp**.1s: 1ª medição da pressão arterial sistólica;
- **Bp**.1d: 1ª medição da pressão arterial diastólica;
- **Bp**.2s: 2ª medição da pressão arterial sistólica;
- **Bp**.2d: 2ª medição da pressão arterial diastólica;
- **Waist**: Perímetro da cintura em centímetros;
- **Hip**: Peptídeos híbridos de insulina;
- **Time**.ppn: Nutrição parenteral parcial;

Os conjuntos de dados iniciais incluem um atributo "*target*", correspondente à variável dependente do estudo, que representa o estado metabólico dos pacientes. Este atributo permite classificar os indivíduos em três categorias distintas: *standard*, pré-diabetes e diabetes, de acordo com critérios clínicos padronizados. Dado que a variável de saída é conhecida, o desenvolvimento do sistema enquadra-se na abordagem de aprendizagem supervisionada.

Para este estudo, foram utilizados diversos algoritmos de Inteligência Artificial aplicados à classificação, nomeadamente Árvores de Decisão e Redes Neuronais Artificiais, Regressão Logística, Random Forest e Segmentação.

O *software KNIME Analytics Platform* foi utilizado para a implementação destes algoritmos, uma vez que disponibiliza ferramentas robustas e amplamente utilizadas para a construção, treino e validação de modelos de aprendizagem automática. No total, foram desenvolvidos cinco modelos de classificação, correspondentes às técnicas anteriormente mencionadas, com o objetivo de prever o estado glicémico dos pacientes.

## 4.2 Carregamento dos dados

Com o intuito de carregar os conjuntos de dados na plataforma *KNIME* para posterior tratamento e utilização nos modelos de aprendizagem automática, recorreu-se à utilização de um nó de leitura de dados. Embora os *datasets* tenham sido disponibilizados tanto em formato Excel como em formato CSV, optou-se pela utilização do ficheiro CSV, atendendo às suas vantagens em termos de simplicidade, compatibilidade e eficiência no processamento de dados.

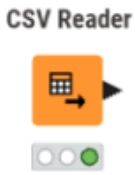



Figura 1: Nodo *CSV Reader*

Desta forma, foi selecionado o nodo ***CSV Reader*** que permite a importação direta de dados estruturados, garantindo uma leitura consistente das variáveis, bem como um melhor desempenho na manipulação de grandes volumes de dados. Esta escolha contribui para uma maior reprodutibilidade do processo, uma vez que o formato CSV é amplamente suportado por diferentes plataformas e ferramentas de análise de dados.

Através da *figura 2*, é possível visualizar uma pequena amostra de um *sub-dataset* que irá ser analisado.

Verifica-se que o *sub-dataset* considerado é constituído por 22 atributos, os quais serão utilizados para efeitos de análise e desenvolvimento dos modelos de classificação, uma vez que cada um fornece informação relevante para a caracterização clínica dos indivíduos.

O atributo alvo ou *target*, identificado pela designação "*ratio_ target*", corresponde à variável dependente do estudo e representa o estado metabólico dos pacientes. Esta variável é utilizada como referência para o treino e validação dos modelos de aprendizagem supervisionada.

Os restantes atributos constituem as variáveis independentes, englobando parâmetros clínicos, antropométricos e laboratoriais, os quais podem influenciar ou não, ou estar

associados à progressão do estado glicémico e ao desenvolvimento de diabetes.

| number | cholesterol | stab.glucose | hdl | ratio_target | glyhb | location | age | year | month | day | gender | height (cm) | weight (kg) | frame | bp.1s | bp.1d | bp.2s | bp.2d | waist (cm) | hip | time.ppn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 203 | 82 | 56 | standard | 4,31 | Buckingham | 46 | 1973 | 5 | 7 | female | 157 | 55 | medium | 118 | 59 | | | 74 | 38 | 720 |
| 2 | 165 | 97 | 24 | diabets | 4,44 | Buckingham | 29 | 1990 | 6 | 16 | female | 163 | 99 | large | 112 | 68 | | | 117 | 48 | 360 |
| 3 | 269 | 82 | 54 | standard | 5,47 | Buckingham | 45 | 1974 | 3 | 24 | male | 175 | 64 | medium | 113 | 75 | | | 89 | 41 | 85 |
| 4 | 228 | 92 | 37 | prediabetes | 4,64 | Buckingham | 58 | 1961 | 7 | 24 | female | 155 | 116 | large | 190 | 92 | 185 | 92 | 124 | 57 | 180 |
| 5 | 174 | 261 | 27 | diabetes | 9,53 | Buckingham | 52 | 1967 | 1 | 20 | male | 182 | 148 | large | 137 | 87 | | | 139 | 50 | 65 |
| 6 | 237 | 95 | 71 | standard | 5,08 | Louisa | 68 | 1951 | 8 | 2 | female | 156 | 70 | medium | 123 | 67 | | | 91 | 41 | 665 |
| 7 | 78 | 93 | 12 | prediabetes | 4,63 | Buckingham | 67 | 1952 | 9 | 13 | male | 170 | 54 | large | 110 | 50 | | | 84 | 38 | 480 |
| 8 | 249 | 90 | 28 | diabetes | 7,72 | Buckingham | 64 | 1955 | 11 | 22 | male | 173 | 83 | medium | 138 | 80 | | | 112 | 41 | 300 |
| 9 | 248 | 94 | 69 | standard | 4,81 | Buckingham | 34 | 1985 | 10 | 29 | male | 180 | 86 | large | 132 | 86 | | | 91 | 42 | 195 |
| 10 | 195 | 92 | 41 | standard | 4,84 | Buckingham | 30 | 1989 | 4 | 14 | male | 175 | 87 | medium | 161 | 112 | 161 | 112 | 117 | 49 | 720 |
| 11 | 227 | 75 | 44 | standard | 3,94 | Buckingham | 37 | 1982 | 11 | 16 | male | 150 | 77 | medium | | | | | 86 | 39 | 1020 |
| 12 | 177 | 87 | 49 | standard | 4,84 | Buckingham | 45 | 1974 | 11 | 12 | male | 175 | 75 | large | 160 | 80 | 128 | 86 | 86 | 40 | 300 |
| 13 | 263 | 89 | 40 | diabeetes | 5,78 | Buckingham | 55 | 1964 | 2 | 11 | female | 160 | 92 | small | 108 | 72 | | | 114 | 50 | 240 |
| 14 | 302 | 85 | 42 | diabetes | 5,19 | Buckingham | 28 | 1991 | 12 | 19 | male | 184 | 95 | large | 95 | 61 | | | 107 | 47 | 560 |
| 15 | 197 | 83 | 50 | standard | 4,55 | Louisa | 19 | 2000 | 8 | 29 | female | 151 | 54 | smaal | 122 | 69 | | | 80 | 36 | 335 |
| 16 | 242 | 82 | 54 | standard | 4,77 | Louisa | 60 | 1959 | 8 | 11 | female | 165 | 71 | medium | 130 | 90 | 130 | 90 | 99 | 45 | 300 |
| 17 | 254 | 340 | 35 | diabetes | 12,74 | Buckingham | 76 | 1943 | 5 | 24 | male | 179 | 92 | large | 152 | 87 | | | 108 | 44 | 25 |
| 18 | 215 | 128 | 34 | prediabetes | 4,97 | Louisa | 38 | 1981 | 6 | 30 | female | 147 | 88 | medium | 102 | 68 | | | 107 | 50 | 90 |
| 19 | 238 | 75 | 36 | diabetes | 4,47 | Louisa | 27 | 1992 | 2 | 24 | female | 152 | 77 | medium | 130 | 80 | | | 89 | 41 | 720 |

Figura 2: *Sub-dataset*

## 4.3 Exploração e análise de dados

***Nodo Data Explorer***

O nodo ***Data Explorer*** é uma ferramenta destinada à exploração inicial dos dados, permitindo uma análise detalhada da estrutura, distribuição e qualidade do conjunto de dados. Este nodo fornece estatísticas descritivas para cada atributo, como valores mínimos e máximos, média, mediana, desvio padrão e frequência de valores categóricos.

Numeric Nominal Data Preview

Search:

| Column | Exclude Column | Minimum | Maximum | Mean | Standard Deviation | Variance |
|---|---|---|---|---|---|---|
| number | ☐ | 1 | 816 | 408.5 | 235.7032032026718 | 55556 |
| cholesterol | ☐ | 70 | 445 | 207.80835380835384 | 44.544822475129926 | 1984.2412093408402 |
| stab.glucose | ☐ | 46 | 390 | 106.57720588235279 | 52.86871267965756 | 2795.1007804041847 |
| hdl | ☐ | 12 | 120 | 50.37469287469287 | 17.157067570471806 | 294.3649676177353 |
| age | ☐ | 19 | 93 | 46.93014705882356 | 16.343063607430118 | 267.0957280765067 |
| year | ☐ | 1926 | 2000 | 1972.0698529411775 | 16.34306360743016 | 267.09572807650807 |
| month | ☐ | 1 | 12 | 6.629901960784309 | 3.208162594015509 | 10.292307229640318 |
| day | ☐ | 1 | 31 | 15.98897058823529 | 8.615786675823959 | 74.23178004330566 |
| height (cm) | ☐ | 131 | 198 | 167.76923076923052 | 10.886488116326944 | 118.5156235069278 |
| weight (kg) | ☐ | 43 | 148 | 80.65970515970521 | 18.41488309015713 | 339.10791922415495 |
| bp.1s | ☐ | 85 | 250 | 136.69727047146392 | 22.6760095406905 | 514.2014086894866 |
| bp.1d | ☐ | 47 | 124 | 83.33746898263027 | 13.583795993241113 | 184.5195135859933 |

Figura 3: Tabela resultado do Nodo *Data Explorer*

Adicionalmente, o ***Data Explorer*** apresenta visualizações automáticas, incluindo histogramas, gráficos de barras e tabelas de frequência, facilitando a identificação de padrões, assimetrias, valores extremos e possíveis inconsistências nos dados. O nodo

também permite detetar valores em falta e avaliar a distribuição das classes do atributo *target*, constituindo uma etapa fundamental para a compreensão global do *dataset* antes da aplicação de técnicas de pré-processamento e modelação.

***Nodo Box Plot***

O nodo ***Box Plot*** é utilizado para a análise visual da distribuição das variáveis numéricas, através da representação gráfica baseada em quartis. Este tipo de gráfico permite identificar de forma clara a mediana, os quartis, a amplitude interquartil e os valores extremos (*outliers*) presentes nos dados.

A utilização do ***Box Plot*** é particularmente relevante na deteção de assimetria na distribuição, comparação entre diferentes grupos ou classes e avaliação da dispersão dos dados. No contexto da análise exploratória, este nodo auxilia na identificação de variáveis que possam necessitar de tratamento adicional, como normalização ou remoção de *outliers*, contribuindo para a melhoria da qualidade dos dados e do desempenho dos modelos de aprendizagem automática.

## 4.4 Pré-processamento dos dados

O pré-processamento dos dados representa uma etapa crítica no desenvolvimento de sistemas de aprendizagem automática, uma vez que a qualidade, consistência e adequação dos dados têm impacto direto no desempenho, estabilidade e capacidade de generalização dos modelos. Assim, antes da fase de treino e validação, foi realizado um conjunto sistemático de procedimentos de análise, limpeza, transformação e normalização dos dados, recorrendo aos nodos disponibilizados pela plataforma *KNIME Analytics Platform*.

***Análise exploratória inicial dos dados***

O processo teve início com a análise exploratória do ficheiro CSV através do nodo ***Data Explorer***, o qual permite uma avaliação abrangente da estrutura do conjunto de dados.

Através desta análise, foi possível identificar:

- Valores extremos fora do intervalo clinicamente expectável;
- Atributos com desvios padrões elevados, indicando elevada dispersão;
- Inconsistências nos tipos de dados, nomeadamente variáveis numéricas armazenadas como texto (*strings*);
- Erros ortográficos e redundâncias em atributos nominais.

Esta etapa foi essencial para orientar as decisões subsequentes de limpeza e transformação dos dados.

***Conversão de tipos de dados***

Durante a análise exploratória, verificou-se que o atributo "*GlyHb*", Hemoglobina Glicada, se encontrava incorretamente representado como variável do tipo *string*. Uma vez que este atributo constitui um parâmetro clínico quantitativo relevante para a classificação do estado glicémico, procedeu-se à sua conversão para formato numérico utilizando o nodo ***String to Number***.

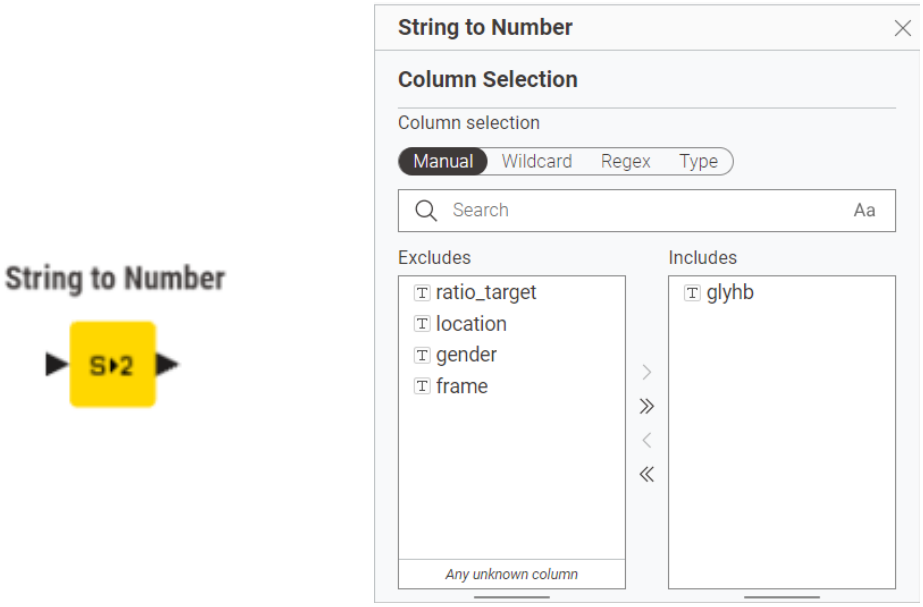



Figura 4: Nodo *String to Number* e sua configuração

Este nodo permite a transformação segura de valores textuais em valores numéricos, assegurando a compatibilidade do atributo com algoritmos de aprendizagem automática que requerem entradas quantitativas.

***Normalização de valores nominais inconsistentes***

A análise da aba "*Nominal*"do ***Data Explorer*** revelou a existência de categorias semanticamente equivalentes, mas com grafias distintas, como por exemplo "diabetes" e "diabets", "female" e "woman", ou "small" e "smaal". Estas inconsistências introduzem ruído nos dados e podem levar a interpretações erradas por parte dos modelos.

| Column | Exclude Column | No. missings | Unique values | All nominal values | Frequency Bar Chart |
|---|---|---|---|---|---|
| ratio_target | ☐ | 2 | 6 | standard, prediabetes, diabetes, diabets, standart, diabeetes | |
| location | ☐ | 0 | 2 | Louisa, Buckingham | |
| gender | ☐ | 0 | 4 | female, male, woman, man | |
| frame | ☐ | 24 | 5 | medium, large, small, smaal, laarge | |

Figura 5: Resultados nominais do nodo *Data Explorer*

Para resolver este problema, foi utilizado o nodo ***Rule Engine***, que permite definir regras condicionais para mapear diferentes representações textuais para uma única categoria normalizada. Este processo assegura a coerência semântica dos atributos categóricos e reduz a dimensionalidade implícita do *dataset*.

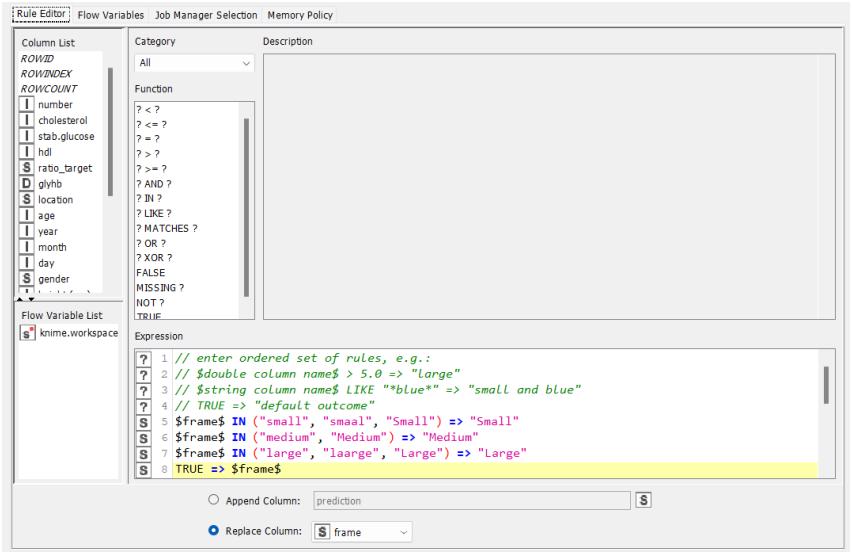



Figura 6: Nodo *Rule Engine* para a coluna "*frame*"

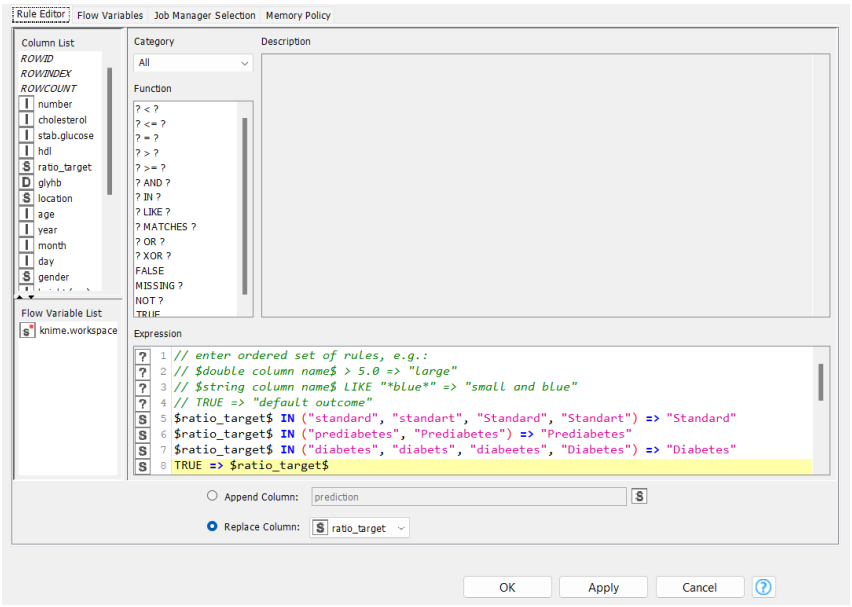



Figura 7: Nodo *Rule Engine* para a coluna "*ratio_target*"

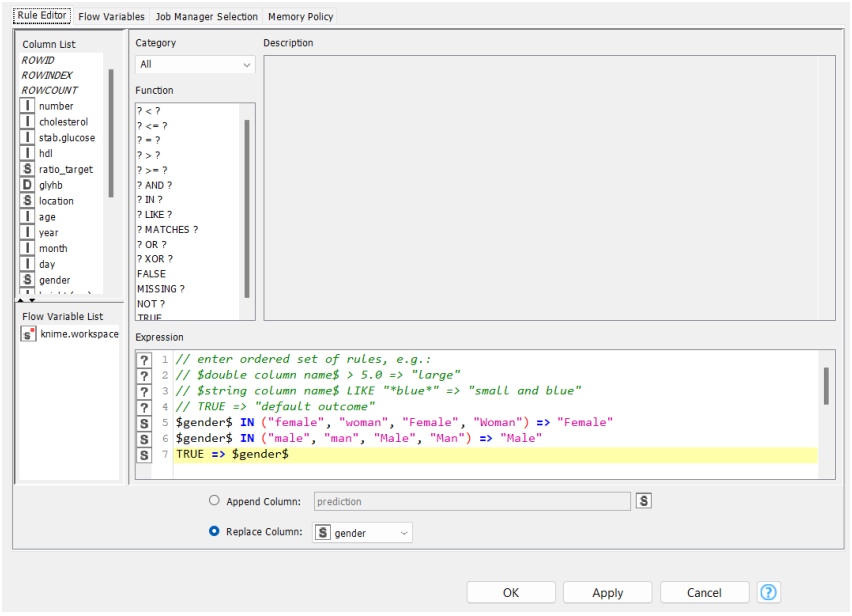



Figura 8: Nodo *Rule Engine* para a coluna *gender*

***Deteção e remoção de valores atípicos (outliers)***

Posteriormente, recorreu-se ao nodo ***Numeric Outliers***, cuja função é identificar e remover observações que se encontrem fora do intervalo estatisticamente esperado. A presença de *outliers* pode distorcer métricas estatísticas, influenciar negativamente a aprendizagem dos modelos e aumentar o erro de previsão.

Neste estudo, o nodo foi aplicado a todas as variáveis numéricas, com exceção do atributo "*time.ppn*", uma vez que este apresenta uma natureza específica e foi tratado separadamente numa fase posterior.

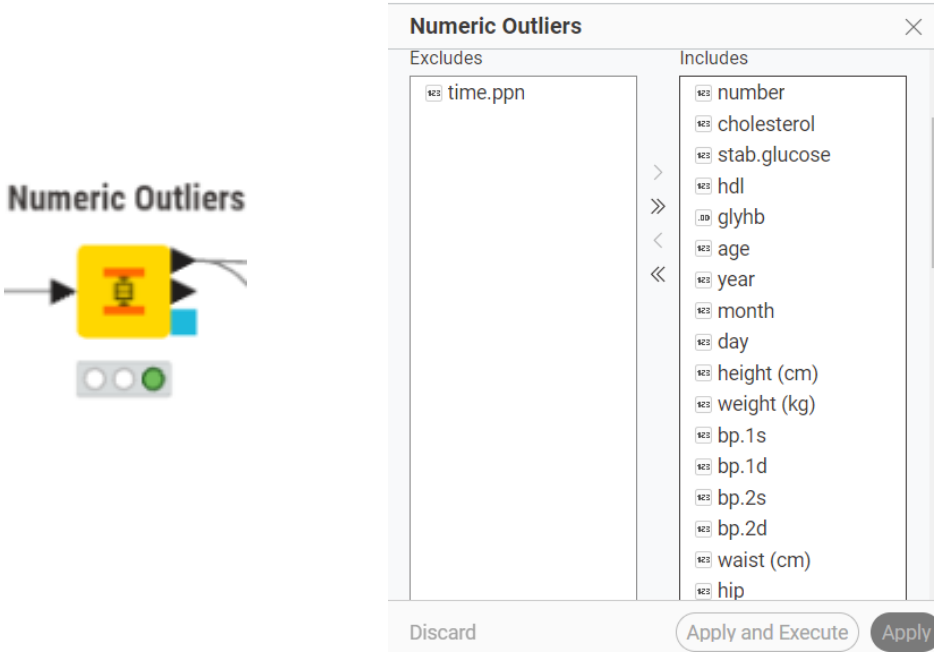



Figura 9: Nodo *Numeric Outliers* e sua configuração

***Seleção de medições clínicas mais fiáveis***

Durante a análise dos dados, verificou-se que nem todos os pacientes apresentavam o mesmo número de medições da pressão arterial sistólica e diastólica. Em alguns casos, encontrava-se disponível apenas uma medição, enquanto noutros estavam registadas duas medições distintas.

Para os pacientes com uma única medição, essa medição foi integralmente mantida no conjunto de dados. Nos casos em que existiam duas medições, optou-se por considerar exclusivamente a segunda medição ("*bp.2s*"e "*bp.2d*"). Esta decisão fundamenta-se no pressuposto clínico de que a repetição da medição ocorre quando os profissionais de saúde identificam a necessidade de uma avaliação adicional, seja para confirmar valores iniciais, reduzir a variabilidade ou corrigir possíveis erros associados à primeira medição. Assim, a segunda medição foi considerada mais representativa e fiável do estado clínico do paciente.

Como resultado deste processo, foram criados dois novos atributos finais, "*bp.1sfinal*"e "*bp.1dfinal*"que agregam, para cada paciente, a medição mais fiável da pressão arterial sistólica e diastólica, respetivamente. As colunas originais de medições foram posteriormente removidas, assegurando um conjunto de dados mais consistente e adequado à fase de modelação.

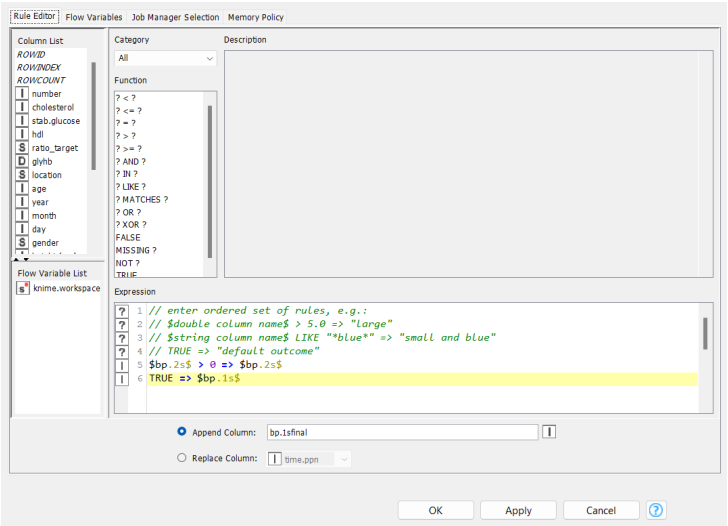



Figura 10: Nodo *Rule Engine* para a coluna *bp.1sfinal*

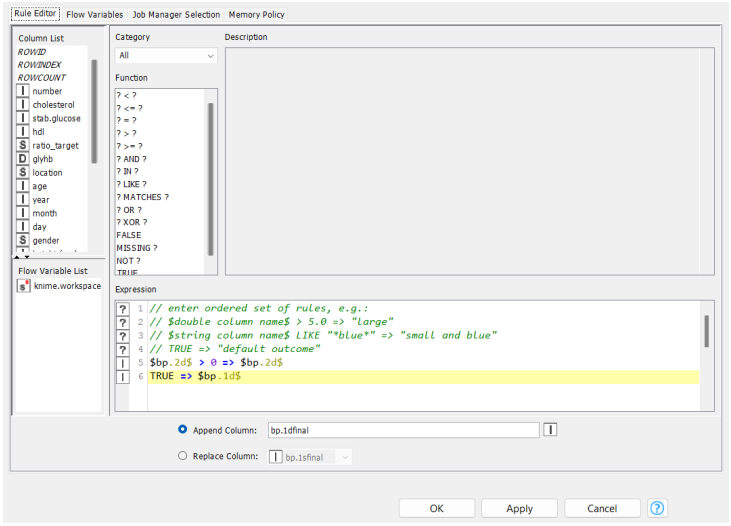



Figura 11: Nodo *Rule Engine* para a coluna *bp.1dfinal*

***Discretização do atributo time.ppn***

O atributo *time.ppn* corresponde a uma variável numérica contínua que representa o intervalo temporal associado à medição, apresentando uma elevada variabilidade e um amplo intervalo de valores. Esta heterogeneidade pode dificultar o processo de aprendizagem dos modelos de classificação, uma vez que valores muito dispersos tendem a introduzir ruído, aumentar a complexidade do espaço de decisão e reduzir a capacidade de generalização dos algoritmos.

Com o objetivo de mitigar estes efeitos e tornar a variável mais clinicamente significativa e computacionalmente eficiente, procedeu-se à discretização do atributo *time.ppn*, agrupando os valores contínuos em três categorias bem definidas, de acordo com intervalos temporalmente interpretáveis:

- **Pós-prandial ($\leq$ 120):** corresponde a medições realizadas num período próximo após a ingestão alimentar, no qual os níveis de glicose tendem a sofrer variações fisiológicas significativas;
- **Intermédio ($>$ 120 textbfe $\leq$ 480):** representa um intervalo de transição entre o estado pós-prandial e o jejum, no qual os valores metabólicos tendem a estabilizar progressivamente;
- **Jejum ($>$ 480):** corresponde a medições realizadas após um período prolongado sem ingestão alimentar, refletindo o estado basal do metabolismo glicémico.

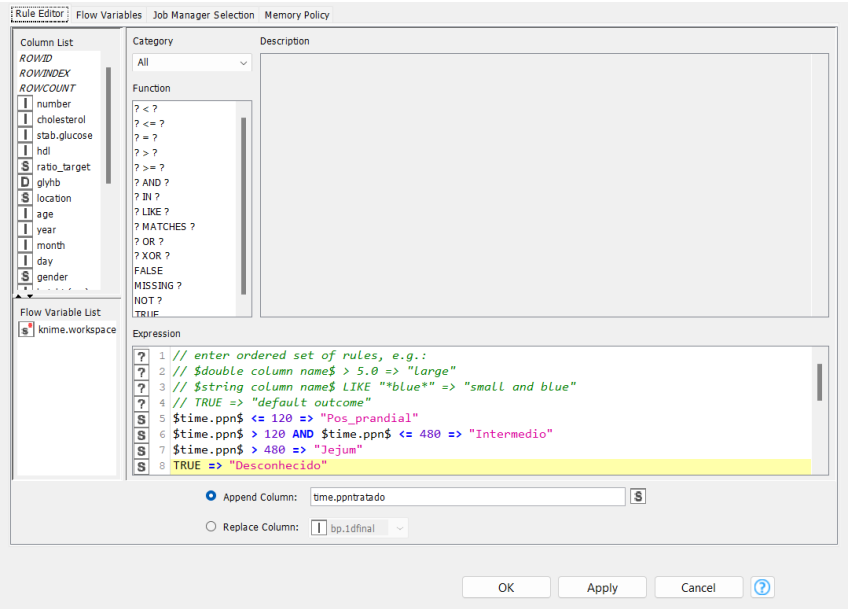



Figura 12: Nodo *Rule Engine* para a coluna *time.ppntratado*

Este processo originou um novo atributo categórico denominado "*time.ppntratado*", que substitui a variável contínua original. A discretização permite reduzir a complexidade do espaço de entrada, facilitar a interpretação clínica dos dados e melhorar o desempenho dos modelos de aprendizagem automática, contribuindo para um aumento da precisão e para a redução do erro de classificação.

***Codificação de variáveis categóricas***

As variáveis categóricas "*gender*"e "*time.ppntratado*"foram transformadas utilizando o nodo ***One to Many***, o qual implementa a técnica de *one-hot encoding*. Este método consiste na conversão de uma variável categórica numa representação binária, criando uma coluna binária distinta para cada classe existente dentro do atributo original.

Por exemplo, no caso da variável "*gender*", as categorias "*male*"e "*female*"são transformadas em colunas independentes, nas quais o valor 1 indica a presença da respetiva categoria e o valor 0 indica a sua ausência. De forma análoga, a variável "*time.ppntratado*"é desdobrada em múltiplas colunas binárias correspondentes às categorias pós-prandial, intermédio, jejum e desconhecido, permitindo representar explicitamente situações em que a informação não se encontra disponível ou não é passível de classificação.

A inclusão da categoria desconhecido assegura que os valores em falta ou ambíguos não são descartados nem introduzem enviesamentos artificiais no processo de aprendizagem, preservando a integridade dos dados e permitindo que os modelos aprendam a lidar com incerteza de forma adequada.

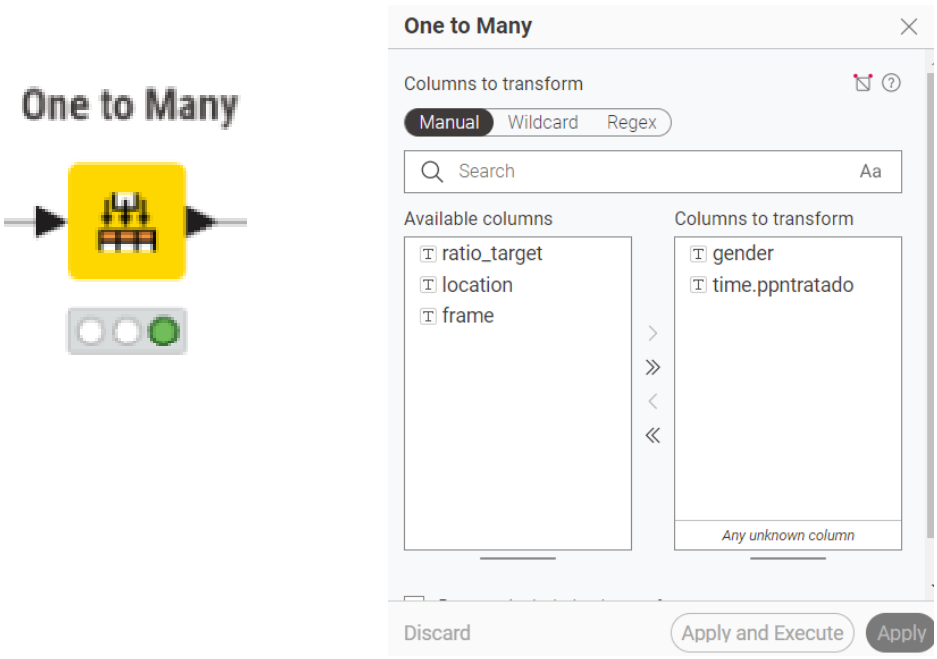



Figura 13: Nodo *One to Many* e sua configuração

A utilização do *one-hot encoding* é fundamental para evitar a introdução de relações ordinais artificiais entre categorias que não possuem uma ordem natural. Caso estas categorias fossem codificadas diretamente como valores inteiros (por exemplo, 0, 1, 2), os modelos poderiam interpretar incorretamente uma hierarquia inexistente entre as classes.

Além disso, esta técnica garante a compatibilidade com algoritmos de aprendizagem automática que requerem exclusivamente dados numéricos, permitindo que cada categoria contribua de forma independente para o processo de aprendizagem. Desta forma, o nodo ***One to Many*** facilita a correta interpretação das variáveis categóricas pelos modelos, contribuindo para uma aprendizagem mais robusta, interpretável e precisa.

***Seleção e eliminação de atributos redundantes***

Com o objetivo de reduzir a complexidade do modelo, minimizar a dimensionalidade do conjunto de dados e evitar redundâncias que possam afetar negativamente o desempenho dos algoritmos de aprendizagem automática, foi utilizado o nodo ***Column Filter*** para remover atributos considerados irrelevantes ou redundantes.

No caso particular da variável "*gender*", após a aplicação da técnica de *one-hot encoding* através do nodo ***One to Many***, foram geradas colunas binárias correspondentes às categorias "*male*"e "*female*". No entanto, estas colunas não são independentes entre si, uma vez que a presença de uma categoria implica necessariamente a ausência das restantes. Assim, a manutenção de todas as colunas introduziria redundância linear no conjunto de dados.

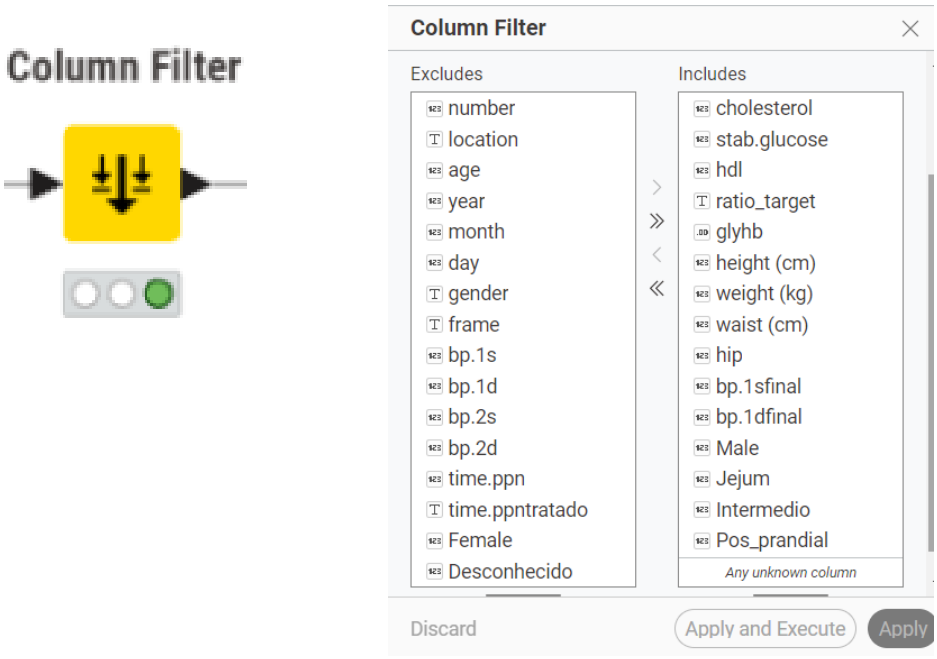



Figura 14: Nodo *Column Filter* e sua configuração

Por este motivo, optou-se por remover a coluna "*female*", mantendo apenas a coluna "*male*"(e, quando aplicável, a coluna "*desconhecido*"). Nesta configuração, um valor 1 na coluna "*male*"indica indivíduos do sexo masculino, enquanto um valor 0 indica automaticamente indivíduos do sexo feminino.

***Tratamento de valores em falta***

Para o tratamento de valores ausentes, foi utilizado o nodo ***Missing Value*** (*figura 15*). A substituição de valores numéricos pela média foi escolhida por preservar a tendência central da distribuição. Para variáveis categóricas, foi aplicada a substituição pelo valor mais frequente, assegurando coerência sem introduzir categorias artificiais.

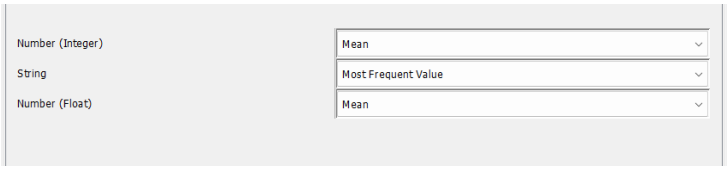



Figura 15: Configuração do nodo *Missing Value*

***Normalização dos dados*** Por fim, foi aplicado o nodo ***Normalizer***, escalando todas as variáveis numéricas para o intervalo [0,1]. A normalização garante que todas as variáveis contribuem de forma equilibrada para o treino dos modelos, sendo particularmente relevante para algoritmos sensíveis à escala dos dados, como as redes neuronais artificiais.

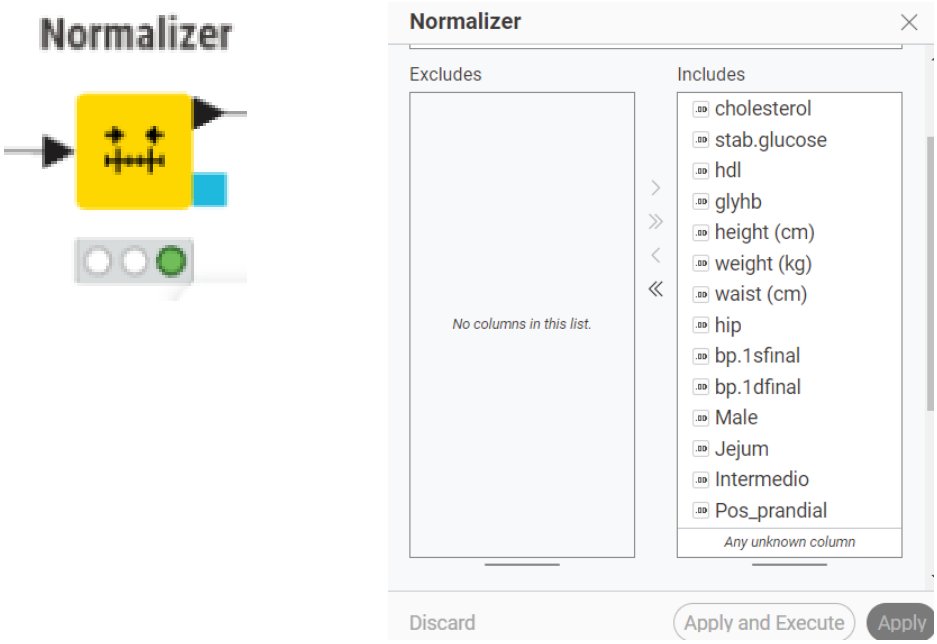



Figura 16: Nodo *Normalizer* e sua configuração

## 4.5 Processamento de Dados

Com vista à modelação dos dados previamente preparados na plataforma *KNIME*, foram aplicados vários nodos de aprendizagem automática, integrando métodos de aprendizagem supervisionada e não supervisionada. Entre os nodos utilizados destacam-se o ***K-Means***, em articulação com o ***Cluster Assigner***, bem como os pares ***Decision Tree Learner / Decision Tree Predictor***, ***Logistic Regression Learner / Logistic Regression Predictor***, ***RProp MLP Learner / MultiLayerPerceptron Predictor*** e, adicionalmente, o ***Random Forest Learner*** juntamente com o ***Random Forest Predictor***.

No que diz respeito às Redes Neuronais Artificiais, o treino do modelo é realizado através do ***RProp MLP Learner***, que implementa uma arquitetura de *perceptron* multicamada, enquanto o ***MultiLayerPerceptron Predictor*** é utilizado na fase de inferência (figura 17).

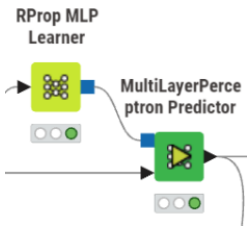



Figura 17: *RProp MLP Learner* com o *MultiLayerPerceptron Predictor*

O ***Decision Tree Learner*** permite a construção de um modelo baseado em árvores de decisão, o qual é posteriormente explorado pelo ***Decision Tree Predictor*** para efetuar a classificação do atributo *target* (figura 18).

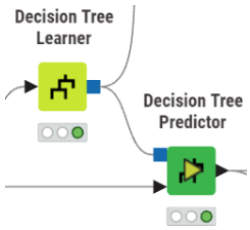



Figura 18: *Decision Tree Learner* com o *Decision Tree Predictor*

De forma análoga, o ***Logistic Regression Learner*** é responsável pelo treino do modelo de regressão logística, sendo o respetivo ***Predictor*** utilizado para gerar as previsões de classe (figura 19).

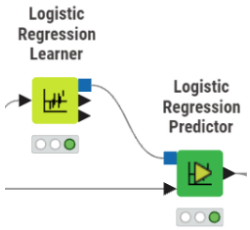



Figura 19: *Logistic Regression Learner* com o *Logistic Regression Predictor*

Por sua vez, o modelo ***Random Forest*** é obtido através do ***Random Forest Learner***, que combina múltiplas árvores de decisão, sendo o processo de previsão assegurado pelo ***Random Forest Predictor*** (figura 20).

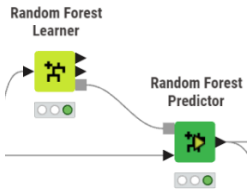



Figura 20: *Random Forest Learner* com o *Random Forest Predictor*

Relativamente à aprendizagem não supervisionada, o nodo ***K-Means*** é aplicado para agrupar os dados em clusters com base na sua similaridade. O ***Cluster Assigner*** é então utilizado para associar cada observação ao respetivo *cluster*. De modo a facilitar a interpretação e comparação dos resultados obtidos, recorreu-se ao ***Rule Engine*** para atribuir rótulos interpretáveis aos diferentes segmentos identificados (figura 21).

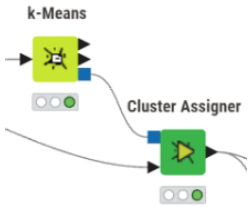



Figura 21: *K-Means* com o *Cluster Assigner*

## 4.6 Visualização dos Resultados

Com o objetivo de analisar e interpretar os resultados obtidos, bem como avaliar o desempenho dos modelos de aprendizagem automática desenvolvidos, foram utilizados diversos nodos de visualização. Entre os nodos selecionados encontram-se o ***Scorer***, o ***Color Manager*** em conjunto com o ***Scatter Plot***, o ***Box Plot*** e a ***ROC Curve***.

O nodo ***Scorer*** é utilizado para comparar os valores reais do atributo *target* com os valores previstos pelos modelos de classificação. Este nodo gera uma matriz de confusão, na qual é possível observar o número de instâncias corretamente e incorretamente classificadas para cada classe. Adicionalmente, o ***Scorer*** disponibiliza métricas de avaliação relevantes, tais como a *accuracy*, o erro de classificação, entre outros indicadores de desempenho, permitindo uma avaliação quantitativa da qualidade das previsões.


Figura 22: *Scorer*

O nodo ***Scatter Plot***, em articulação com o ***Color Manager***, permite a representação gráfica dos dados sob a forma de um gráfico de dispersão. O ***Color Manager*** é responsável pela atribuição de cores distintas às observações, de acordo com a classe ou categoria definida, facilitando a identificação visual de padrões, separações entre classes e possíveis sobreposições nos resultados obtidos.

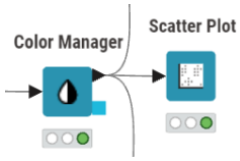



Figura 23: *Scatter Plot* com o *Color Manager*

O ***Box Plot*** é utilizado para a análise da distribuição dos dados e dos resultados das previsões, permitindo visualizar medidas estatísticas como a mediana, os quartis e a presença de valores extremos (*outliers*). Esta visualização é particularmente útil para avaliar a dispersão dos dados e verificar a consistência das variáveis após o pré-processamento.


Figura 24: *Box Plot*

Por fim, a ***ROC Curve*** é utilizada para avaliar o desempenho dos modelos de classificação em termos da sua capacidade discriminativa. Esta curva representa a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos para diferentes limiares de decisão. A análise da área sob a curva (AUC) permite comparar o desempenho dos diferentes modelos, sendo que valores mais elevados indicam uma melhor capacidade de distinção entre as classes.


Figura 25: *ROC Curve*

# 5 Modelos de Aprendizagem

Para a previsão do tipo de diabetes neste projeto, foram selecionados e implementados cinco algoritmos distintos de ML, cada um com características e aplicações específicas. A utilização de múltiplos algoritmos permite não apenas comparar o seu desempenho relativo, mas também compreender as vantagens e limitações de diferentes abordagens na classificação de dados biomédicos.

A seleção destes algoritmos fundamenta-se na sua aplicabilidade comprovada em contextos de classificação médica, na diversidade de abordagens (supervisionadas e não-supervisionadas), e na facilidade de implementação em plataformas como o *KNIME*. Nas secções seguintes, será apresentada a fundamentação teórica de cada algoritmo, a sua implementação no *workflow KNIME*, e a análise comparativa do seu desempenho na previsão de diabetes.

## 5.1 Redes Neuronais (RProp)

O workflow inicia-se com o nodo ***CSV Reader***, seguido de um ***Table Partitioner***, configurado para dividir aleatoriamente a base de dados em 80% para treino e 20% para teste.

O treino do modelo é realizado através do nodo ***RProp MLP Learner***, configurado para classificação multiclasse. Após o treino, o modelo é aplicado ao conjunto de teste utilizando o nodo ***MultiLayerPerceptron Predictor***, que gera as probabilidades associadas a cada uma das classes. De forma a permitir a aplicação de *thresholds* específicos por classe, o ***MultiLayerPerceptron Predictor*** foi configurado para disponibilizar separadamente as probabilidades previstas para cada categoria.

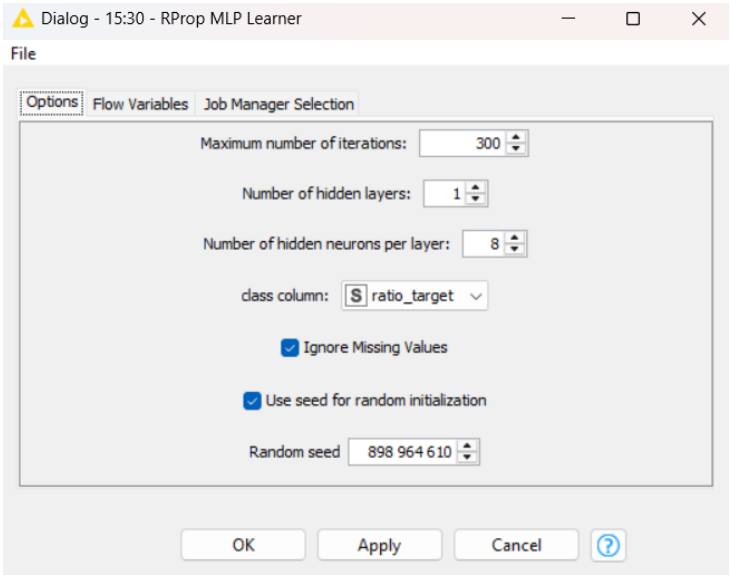



Figura 26: Nodo *RProp MLP Learner*, utilizado para o treino de um modelo de redes neuronais do tipo *Multilayer Perceptron*

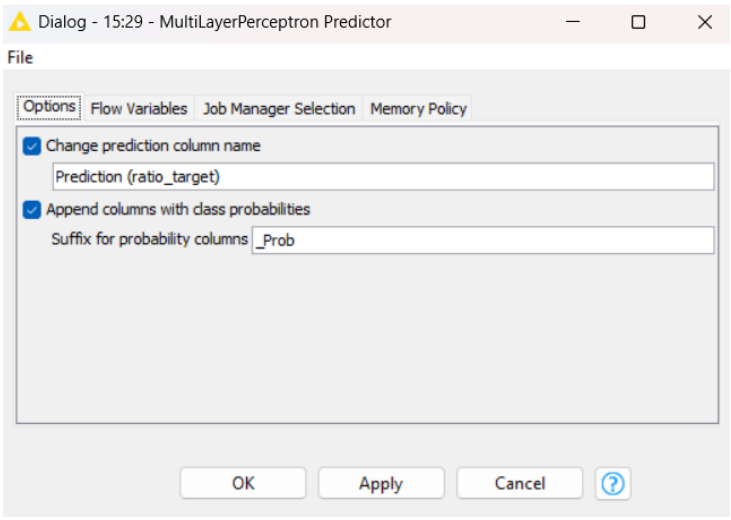



Figura 27: Nodo *Multilayer Perceptron Predictor*, utilizado para aplicar o modelo de redes neuronais treinado aos dados de teste

Seguidamente, é aplicado um ***Rule Engine***, onde foram definidos *thresholds* distintos para cada uma das classes (diabetes, pré-diabetes e standard). Sempre que um indivíduo não satisfaz nenhum dos *thresholds* definidos, é atribuída uma classe adicional designada por "incerto", permitindo identificar explicitamente situações de maior ambiguidade na decisão do modelo.

A avaliação do desempenho é realizada através de nodos ***Scorer***, antes e depois da aplicação dos *thresholds*, permitindo analisar o impacto direto das regras no erro total e na distribuição dos tipos de erro. Para complementar esta análise, foram utilizadas representações gráficas, nomeadamente o ***ROC Curve***, ***Scatter Plot (JavaScript – Legacy)*** e ***Box Plots***, que permitem uma análise mais detalhada do comportamento do modelo.

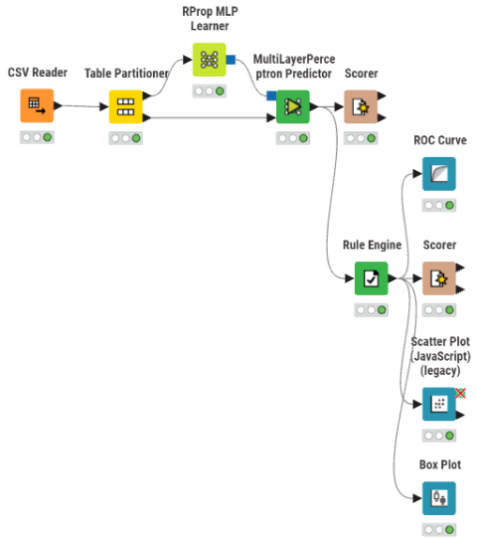



Figura 28: Redes Neuronais (RProp)

## 5.2 Árvore de Decisão

O *workflow* inicia-se com o nodo ***CSV Reader***, seguido de um ***Table Partitioner***, configurado para dividir aleatoriamente a base de dados em 80% para treino e 20% para teste.

O treino do modelo é realizado através do nodo ***Decision Tree Learner***, configurado para classificação. A estrutura da árvore aprendida foi analisada através da ***Decision Tree View (JavaScript – Legacy)***, permitindo compreender as regras de decisão criadas pelo modelo e a forma como as variáveis contribuem para a classificação das diferentes classes.

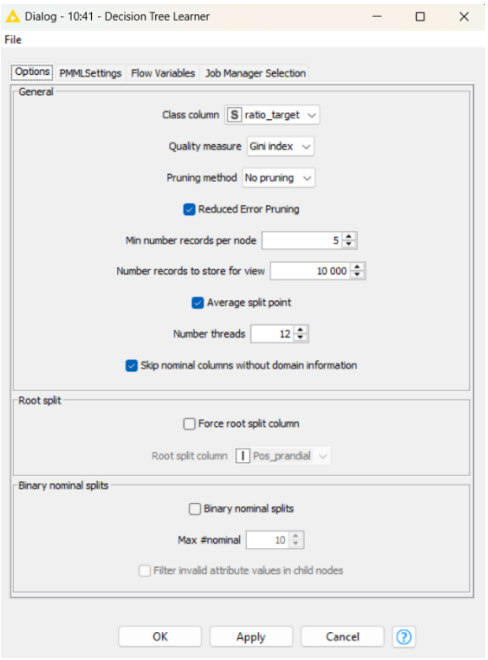



Figura 29: Nodo *Decision Tree Learner*, que permite a seleção, divisão e classificação de atributos através de árvores de decisão

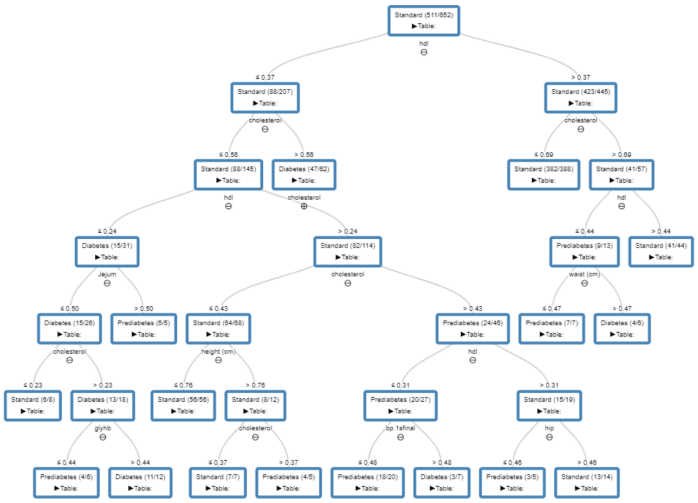


Figura 30: Nodo *Decision Tree View (JavaScript – Legacy)*, utilizado para visualizar a estrutura e as regras da árvore de decisão

O modelo treinado é aplicado ao conjunto de teste utilizando o ***Decision Tree Predictor***, que gera as previsões para cada instância. Estas previsões são inicialmente avaliadas com um ***Scorer***, permitindo obter uma referência do desempenho do modelo sem qualquer ajuste adicional.

Posteriormente, foi introduzido um ***Rule Engine***, onde se tentou aplicar *thresholds* específicos para cada uma das classes, com o objetivo de reduzir erros clinicamente mais relevantes. Após a aplicação destas regras, os resultados foram novamente avaliados através de um segundo ***Scorer***. Para complementar a análise, foram utilizados nodos de visualização, nomeadamente o ***ROC Curve***, ***Scatter Plot (JavaScript – Legacy)*** e ***Box Plots***, permitindo analisar a distribuição das classificações e a capacidade discriminativa do modelo.

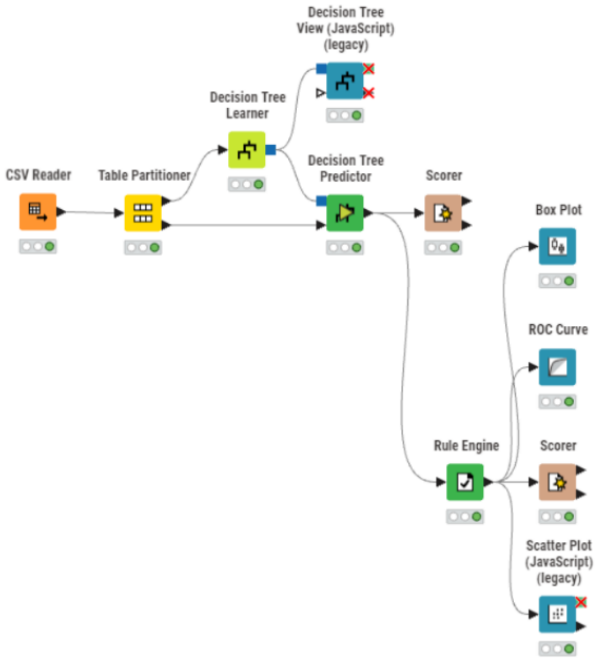



Figura 31: Árvore de Decisão

## 5.3 Segmentação *K-Means*

O modelo *K-Means* inicia-se com o nodo ***CSV Reader***, que importa o ficheiro com os dados já pré-processados, seguido do ***Column Filter***, onde são escolhidas apenas as variáveis numéricas relevantes para o *clustering* excluindo a coluna "*ratio_target*" para manter a natureza não supervisionada do método.

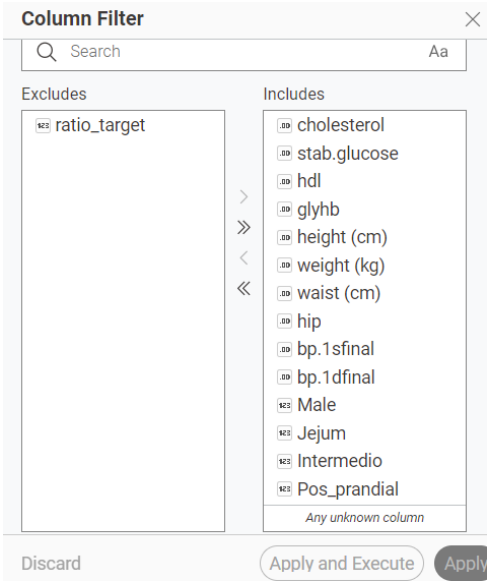



Figura 32: Nodo *Column Filter*, utilizado para selecionar os atributos relevantes da base de dados

Os dados filtrados alimentam o nodo ***K-Means***, no qual foi definido K = 3 para obter três grupos coerentes com as classes clínicas standard, pré-diabetes e diabetes. Em seguida, o nodo ***Cluster Assigner*** atribui a cada indivíduo o cluster correspondente, adicionando uma nova coluna "*cluster*"à tabela.

A qualidade da segmentação é avaliada com o nodo ***Silhouette Coefficient***, que calcula o coeficiente médio de silhueta como medida de coesão *intra-cluster* e separação entre *clusters*. Para relacionar os *clusters* com as categorias reais de diabetes, utiliza-se o nodo ***Joiner*** para combinar a informação de "*clusters*" com a tabela original (através do "*RowID*") e o ***Rule Engine*** para mapear cada *cluster* para uma classe prevista (Standard, Pré-diabetes, Diabetes), criando uma coluna de previsão.

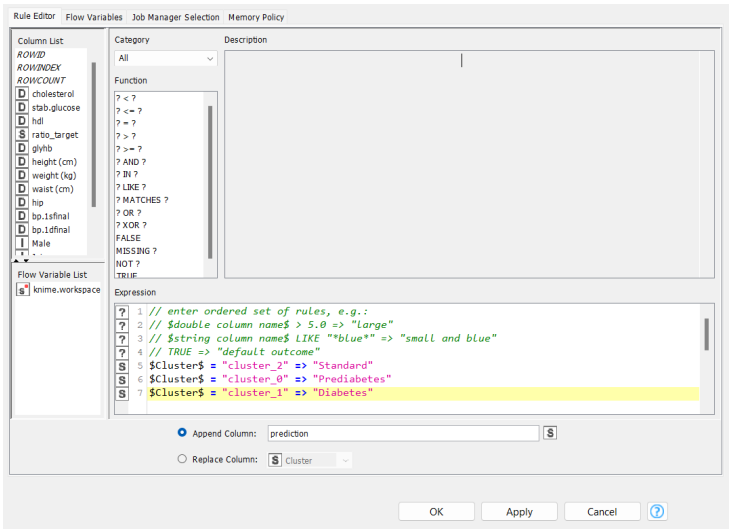



Figura 33: Nodo *Rule Engine*, utilizado para aplicar regras de decisão e thresholds às previsões do modelo

Por fim, o nodo ***Scorer (JavaScript)*** compara a coluna "*ratio_target*" com a coluna

de previsão derivada dos *clusters*, gerando a matriz de confusão e métricas de desempenho, enquanto os nodos de visualização como ***Color Manager*** e ***Scatter Plot (JavaScript)*** permitem representar graficamente os grupos encontrados e apoiar a interpretação dos resultados.


Figura 34: Segmentação *K-Means*

## 5.4 Regressão Logística

O *workflow* utilizado inicia-se com o nodo ***CSV Reader***, seguido de um ***Table Partitioner*** configurado para realizar uma divisão aleatória dos dados, utilizando 80% para treino e 20% para teste. O treino do modelo é efetuado através do nodo ***Logistic Regression Learner***, configurado para classificação. Uma vez que existem três classes possíveis, foi necessário definir uma *reference category*. Esta escolha foi realizada com base no desempenho inicial do modelo sem *thresholds*, tendo sido selecionada a classe pré-diabetes, por apresentar o melhor *score* global nesta fase inicial.

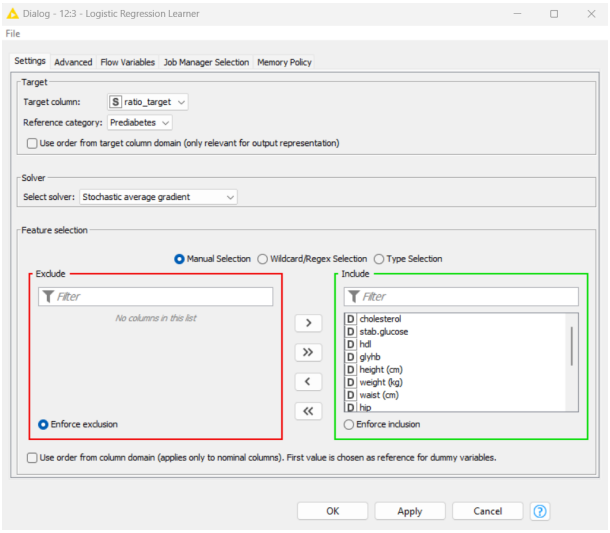



Figura 35: Nodo *Logistic Regression Learner*, utilizado para o treino de um modelo de regressão logística

Após o treino, o modelo é aplicado ao conjunto de teste através do nodo ***Logistic Regression Predictor***. Para permitir a aplicação posterior de *thresholds* às probabilidades associadas a cada classe, foi necessário configurar o ***Logistic Regression Predictor*** com

a opção de *suffix*, garantindo que as probabilidades de todas as classes ficassem disponíveis como colunas distintas.

Seguidamente, é utilizado um ***Rule Engine***, onde são definidos *thresholds* específicos para cada uma das classes (diabetes, pré-diabetes e standard). Estes *thresholds* foram ajustados experimentalmente com o objetivo de minimizar o erro global do modelo, dando particular atenção à redução de classificações clinicamente mais críticas. Foi ainda introduzida uma classe adicional designada por "incerto", atribuída aos casos que não satisfazem nenhum dos *thresholds* definidos.

Os resultados são avaliados antes e depois da aplicação dos *thresholds* através de nodos ***Scorer***, permitindo comparar diretamente o impacto das regras no desempenho do modelo. Complementarmente, foram utilizados nodos ***Box Plot (Legacy)*** para analisar graficamente a distribuição das probabilidades previstas.

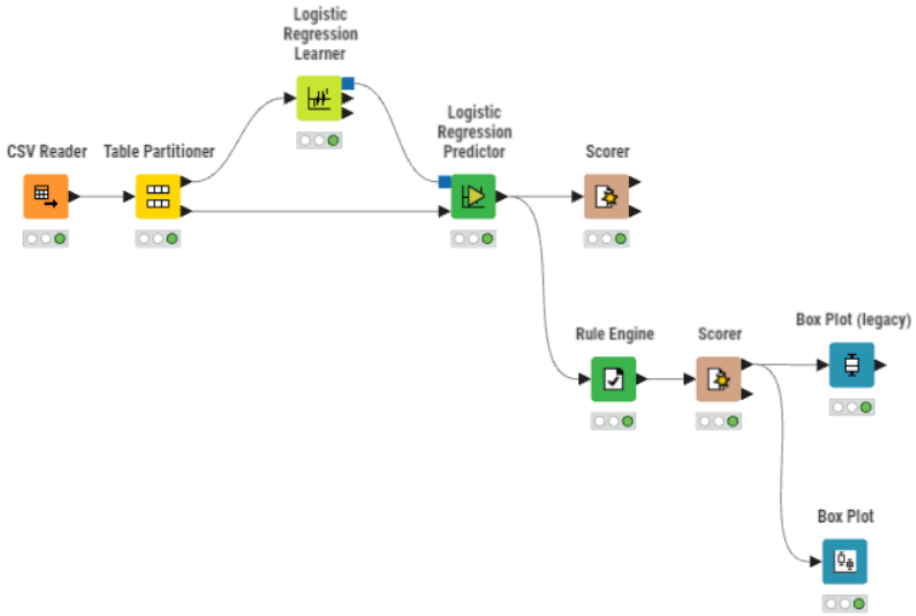



Figura 36: Regressão Logística

## 5.5 Random Forest

Para a construção do modelo de *Random Forest*, foram utilizados dois nodos principais: o ***Random Forest Learner*** e o ***Random Forest Predictor***. O processo iniciou-se com a divisão aleatória do *dataset* através do nodo ***Table Partitioner***, configurado para separar os dados em 80% para treino e 20% para teste. A primeira saída deste nodo corresponde ao conjunto de treino, enquanto a segunda saída contém o conjunto de teste.

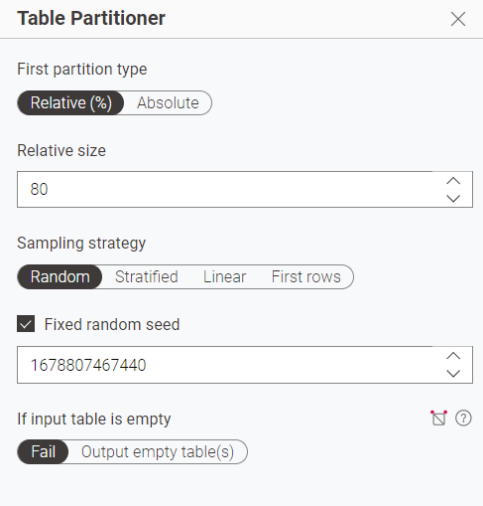



Figura 37: Nodo *Table Partitioning* e respetiva configuração

O conjunto de treino foi ligado ao nodo ***Random Forest Learner***, responsável pelo treino do modelo de classificação, sendo definido o atributo *target* como a variável a prever. Este algoritmo baseia-se na criação de um conjunto de árvores de decisão, geradas a partir de diferentes subconjuntos dos dados e dos atributos, permitindo reduzir o risco de sobreajuste e aumentar a robustez do modelo.

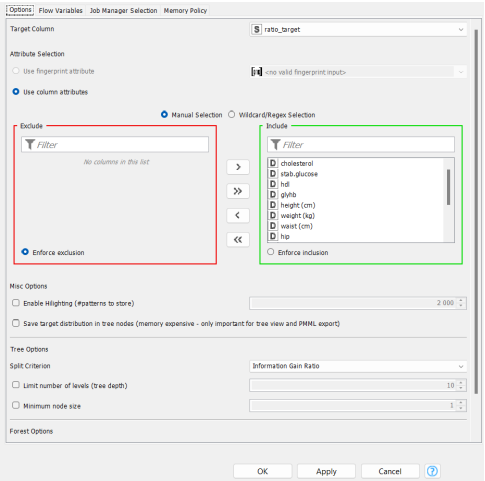



Figura 38: Nodo *Random Forest Learner* e respetiva configuração

Após o treino, o modelo gerado foi utilizado pelo nodo ***Random Forest Predictor***, cuja função consiste em aplicar o classificador previamente treinado a novos dados. A primeira entrada deste nodo foi conectada à saída do ***Random Forest Learner***, que contém o modelo treinado, enquanto a segunda entrada foi ligada à saída do ***Table Partitioner*** correspondente aos dados de teste. Desta forma, o modelo *Random Forest* foi aplicado ao conjunto de teste, permitindo obter as previsões do atributo *target* para cada instância.

As previsões obtidas possibilitaram posteriormente a avaliação do desempenho do modelo através dos nodos de validação e visualização, concluindo assim o processo de construção, aplicação e análise do modelo de *Random Forest*.

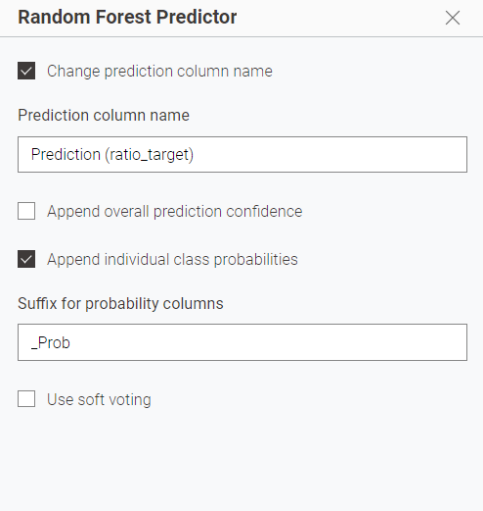



Figura 39: Nodo *Random Forest Predictor*

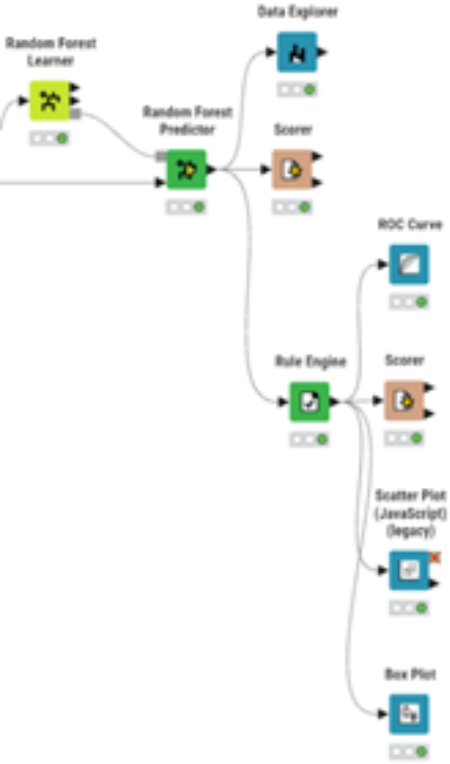


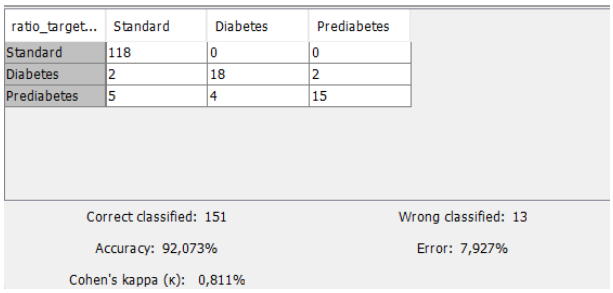

Figura 40: *Random Forest*

# 6 Desenvolvimento dos algoritmos de aprendizagem

## 6.1 Aprendizagem supervisionada

***Redes Neuronais (RProp)***

Na avaliação inicial do modelo de Redes Neuronais (RProp), sem aplicação de *thresholds*, observou-se um total de 13 classificações incorretas, correspondendo a uma taxa de erro de aproximadamente 7,9%. Analisando a matriz de confusão, verificou-se a presença de erros clinicamente indesejáveis, nomeadamente casos em que indivíduos pré-diabéticos eram classificados como standard ou como diabéticos, o que representa uma atribuição inadequada do nível de risco.



| ratio_target... | Standard | Diabetes | Prediabetes |
|---|---|---|---|
| Standard | 118 | 0 | 0 |
| Diabetes | 2 | 18 | 2 |
| Prediabetes | 5 | 4 | 15 |

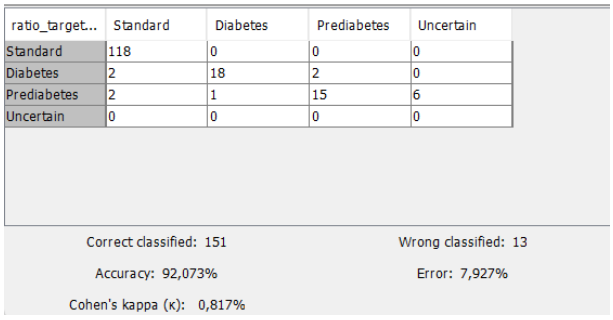
Correct classified: 151 Wrong classified: 13

Accuracy: 92,073% Error: 7,927%

Cohen's kappa (κ): 0,811%

Figura 41: Matriz de confusão do modelo Redes Neuronais sem aplicação de *thresholds*

Com a aplicação dos *thresholds*, o erro total manteve-se inalterado, continuando a verificar-se 13 classificações incorretas. No entanto, apesar de não se observar uma melhoria ao nível do erro global ou da *accuracy*, registou-se uma alteração relevante na distribuição dos erros. Em particular, alguns casos anteriormente classificados incorretamente como pertencentes a uma classe clínica específica passaram a ser classificados como "incertos", refletindo uma maior cautela do modelo na tomada de decisão.



| ratio_target... | Standard | Diabetes | Prediabetes | Uncertain |
|---|---|---|---|---|
| Standard | 118 | 0 | 0 | 0 |
| Diabetes | 2 | 18 | 2 | 0 |
| Prediabetes | 2 | 1 | 15 | 6 |
| Uncertain | 0 | 0 | 0 | 0 |

Correct classified: 151 Wrong classified: 13

Accuracy: 92,073% Error: 7,927%

Cohen's kappa (κ): 0,817%

Figura 42: Matriz de confusão do modelo Redes Neuronais com aplicação de *thresholds*

Esta situação verifica-se quando as probabilidades previstas para um determinado indivíduo não satisfazem nenhum dos *thresholds* definidos para as classes diabetes, pré-diabetes ou standard. A introdução da classe "incerto" permitiu, assim, evitar classificações forçadas em situações ambíguas, tornando o modelo mais conservador e transparente quanto às suas limitações. Embora fosse possível eliminar esta classe intermédia, tal conduziria a uma redistribuição dos erros pelas restantes classes, sem benefícios ao nível do desempenho global.

A análise dos ***ROC Curve*** e ***Box Plots*** confirmou que a aplicação dos *thresholds* não teve impacto significativo na capacidade discriminativa global do modelo, mas contribuiu

para uma gestão mais cuidadosa dos casos ambíguos. Desta forma, conclui-se que, no modelo de Redes Neuronais (RProp), a aplicação de *thresholds* não melhorou o erro total, mas permitiu uma reorganização qualitativa dos erros, alinhada com uma abordagem mais conservadora e clinicamente prudente.

***Árvore de Decisão***

Na avaliação inicial do modelo de Árvore de Decisão, sem aplicação de *thresholds*, verificou-se um erro total relativamente baixo. No entanto, com a presença de erros clinicamente indesejáveis, em particular casos em que indivíduos pertencentes às classes de diabetes ou pré-diabetes eram classificados como standard, este tipo de erro foi considerado prioritário de reduzir, uma vez que representa uma subavaliação do risco clínico do indivíduo.

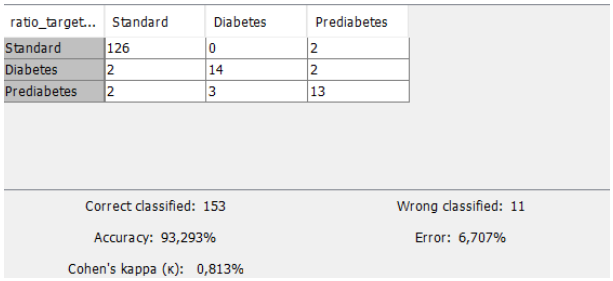



Figura 43: Matriz de confusão do modelo Árvore de Decisão sem aplicação de *thresholds*

Com a aplicação dos *thresholds* foram realizadas várias tentativas de ajuste, procurando reduzir estes falsos negativos sem comprometer o desempenho global do modelo. No entanto, verificou-se que a variação dos *thresholds* não conduziu a melhorias efetivas no erro total, nem permitiu reduzir os erros mais críticos sem introduzir outros problemas, como o aumento do número de classificações incertas ou a redistribuição dos erros para outras classes.

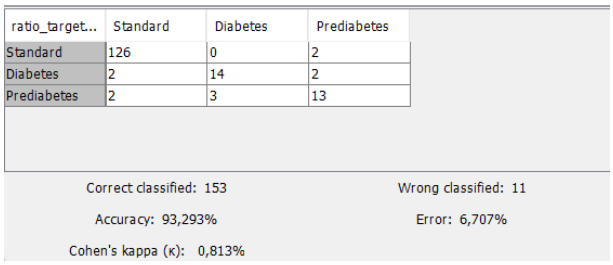


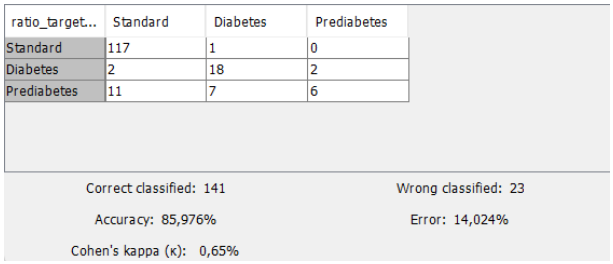

Figura 44: Matriz de confusão do modelo Árvore de Decisão com aplicação de *thresholds*

Desta forma, a solução inicialmente obtida, sem ajustes significativos nos *thresholds*, foi considerada a mais adequada. Esta decisão é justificada pelo facto de as Árvores de Decisão basearem as suas previsões em regras explícitas e hierárquicas, o que limita o impacto de ajustes posteriores baseados apenas em probabilidades.

A análise das ***curvas ROC***, dos ***Scatter Plots*** e dos ***Box Plots*** confirmou a estabilidade do modelo e a ausência de ganhos relevantes com a aplicação de *thresholds*. Assim, o modelo de Árvore de Decisão apresentou um desempenho consistente e interpretável, embora menos flexível a ajustes quando comparado com modelos probabilísticos como a Regressão Logística ou as Redes Neuronais.

***Regressão Logística***

Na avaliação inicial do modelo de Regressão Logística, sem aplicação de *thresholds*, verificou-se um erro total de 23 classificações incorretas, com especial incidência em casos em que indivíduos pré-diabéticos eram classificados como standard, uma situação considerada clinicamente indesejável.



| ratio_target... | Standard | Diabetes | Prediabetes |
|---|---|---|---|
| Standard | 117 | 1 | 0 |
| Diabetes | 2 | 18 | 2 |
| Prediabetes | 11 | 7 | 6 |

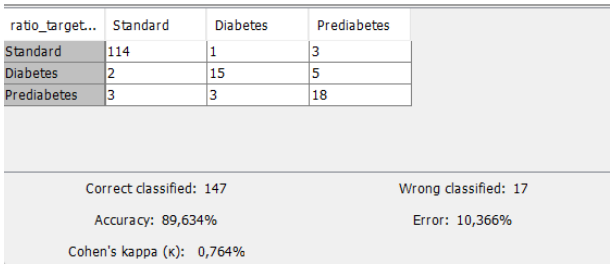
Correct classified: 141 Wrong classified: 23

Accuracy: 85,976% Error: 14,024%

Cohen's kappa (κ): 0,65%

Figura 45: Matriz de confusão do modelo Regressão Logística sem aplicação de *thresholds*

Com a introdução dos *thresholds*, o número de classificações incorretas reduziu para 17, representando uma melhoria significativa do desempenho global do modelo. Esta redução esteve principalmente associada à diminuição dos falsos positivos e falsos negativos mais críticos, em particular na distinção entre indivíduos standard e pré-diabéticos.



| ratio_target... | Standard | Diabetes | Prediabetes |
|---|---|---|---|
| Standard | 114 | 1 | 3 |
| Diabetes | 2 | 15 | 5 |
| Prediabetes | 3 | 3 | 18 |

Correct classified: 147 Wrong classified: 17

Accuracy: 89,634% Error: 10,366%

Cohen's kappa (κ): 0,764%

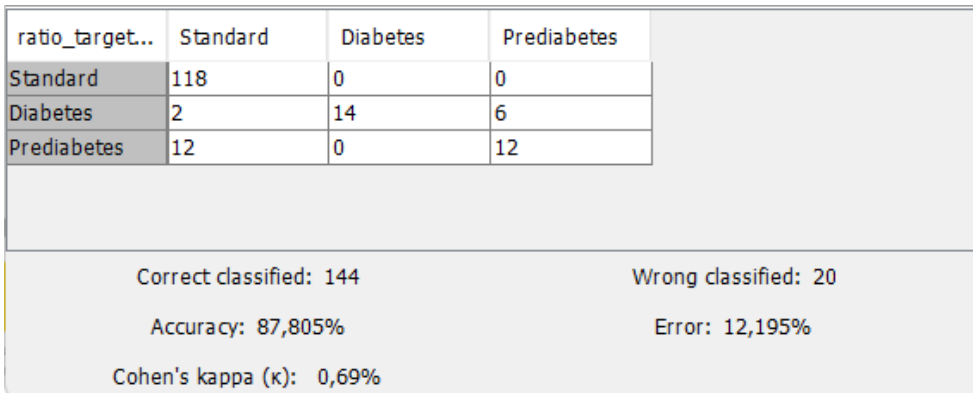
Figura 46: Matriz de confusão do modelo Regressão Logística com aplicação de *thresholds*

Apesar da aplicação dos *thresholds* ter levado a um ligeiro aumento de erro em categorias menos sensíveis, esta troca revelou-se aceitável, uma vez que permitiu reduzir erros com maior impacto clínico. Assim, a estratégia adotada privilegiou a segurança e a coerência clínica das classificações em detrimento de uma otimização puramente numérica do erro total.

A análise gráfica através dos ***Box Plots*** confirmou o efeito dos *thresholds* na redistribuição das probabilidades e na separação entre classes. Conclui-se, assim, que a aplicação de *thresholds* no modelo de Regressão Logística multiclasse contribuiu para uma melhoria do desempenho e para uma tomada de decisão mais alinhada com os objetivos clínicos do problema.

***Random Forest***

Para avaliar a qualidade do modelo *Random Forest*, foram utilizados os nodos ***Scorer*** e ***ROC Curve***. O nodo ***Scorer*** permite comparar os valores reais do atributo *target* com os valores previstos pelo modelo, produzindo a respetiva matriz de confusão, bem como métricas estatísticas relevantes. Numa primeira abordagem, o desempenho do modelo foi avaliado considerando diretamente a classe com maior probabilidade prevista.



| ratio_target... | Standard | Diabetes | Prediabetes |
|---|---|---|---|
| Standard | 118 | 0 | 0 |
| Diabetes | 2 | 14 | 6 |
| Prediabetes | 12 | 0 | 12 |

Correct classified: 144 Wrong classified: 20

Accuracy: 87,805% Error: 12,195%

Cohen's kappa (κ): 0,69%

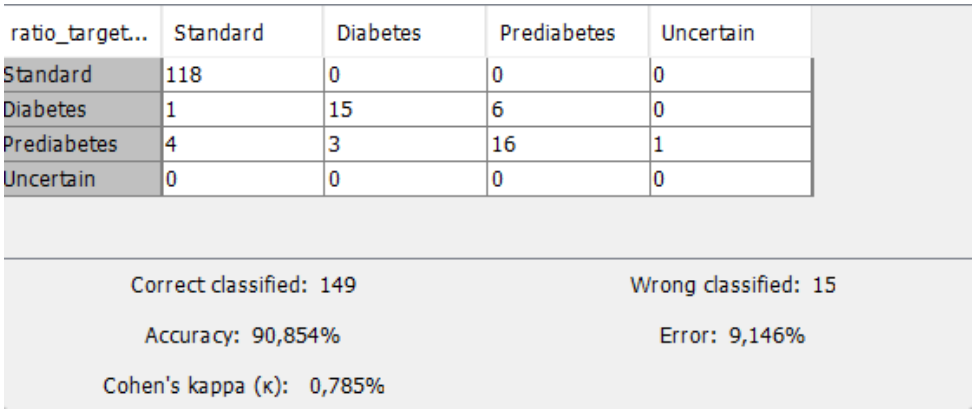
Figura 47: Matriz de confusão do modelo *Random Forest* sem aplicação de *thresholds*

A análise desta matriz evidencia que a classe Standard é corretamente identificada na maioria dos casos. No entanto, verifica-se a existência de falsos negativos, particularmente nas classes Prediabetes e Diabetes, onde alguns pacientes pertencentes a estas categorias são incorretamente classificados como Standard ou como outra classe metabólica. Este tipo de erro é especialmente crítico em contextos clínicos, uma vez que pode conduzir à não identificação de indivíduos em risco ou com patologia instalada.

Globalmente, o modelo classificou corretamente 144 instâncias, apresentando uma *accuracy* de 87,805% e um coeficiente de Cohen (k = 0,69%), indicando uma concordância substancial, embora com margem para melhoria na redução dos erros de classificação mais críticos.

A introdução dos *thresholds* permitiu tornar o processo de decisão mais conservador, reduzindo a probabilidade de classificar incorretamente pacientes com Prediabetes ou Diabetes como Standard.

Os resultados obtidos, apresentados na *figura 48*, demonstram uma melhoria clara do desempenho do modelo. O número de instâncias corretamente classificadas aumentou para 149, enquanto o número de classificações incorretas diminuiu para 15, correspondendo a uma *accuracy* de 90,854% e a um coeficiente de Cohen (k = 0,785%).



| ratio_target... | Standard | Diabetes | Prediabetes | Uncertain |
|---|---|---|---|---|
| Standard | 118 | 0 | 0 | 0 |
| Diabetes | 1 | 15 | 6 | 0 |
| Prediabetes | 4 | 3 | 16 | 1 |
| Uncertain | 0 | 0 | 0 | 0 |

Correct classified: 149 Wrong classified: 15

Accuracy: 90,854% Error: 9,146%

Cohen's kappa (κ): 0,785%

Figura 48: Matriz de confusão do modelo *Random Forest* com aplicação de *thresholds*

De forma particularmente relevante, observa-se uma redução significativa dos falsos negativos, sobretudo nas classes Prediabetes e Diabetes.

Este aspeto é fundamental do ponto de vista clínico, uma vez que a diminuição de falsos negativos contribui para uma melhor identificação de indivíduos em risco metabó-

lico, reduzindo a probabilidade de omissão de casos que requerem acompanhamento ou intervenção médica.

## 6.2 Aprendizagem não supervisionada

***Segmentação K-Means***

Na avaliação do modelo de Segmentação *K-Means* com K=3 *clusters*, observou-se que os resultados demonstram limitações fundamentais da abordagem não-supervisionada quando aplicada a classificação clínica. O modelo dividiu o conjunto de dados em três *clusters* baseado em características biomédicas (colesterol, glucose, HbA1c, medidas antropométricas), mapeando-os posteriormente para as categorias clínicas através do ***Rule Engine***.

A Matriz de Confusão revelou um desempenho global deficiente, com *Accuracy* de apenas 44.12%. O modelo foi incapaz de discriminar entre as três classes, demonstrando um enviesamento sistemático: nunca previu "Diabetes"ou "Pré-diabetes"corretamente, classificando quase todos os casos como "Standard". Este padrão é clinicamente indesejável, representando uma subestimação crítica do risco diabético.

Scorer View

Confusion Matrix

| | Diabetes (Predicted) | Prediabetes (Predicted) | Standard (Predicted) | |
|---|---|---|---|---|
| Diabetes (Actual) | 39 | 18 | 34 | 42.86% |
| Prediabetes (Actual) | 33 | 27 | 26 | 31.40% |
| Standard (Actual) | 179 | 166 | 294 | 46.01% |
| | 15.54% | 12.80% | 83.05% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 44.12% | 55.88% | 0.067 | 360 | 456 |

Figura 49: Nodo *Scorer*, utilizado para avaliar o desempenho do modelo através da matriz de confusão e métricas de classificação

A Segmentação *K-Means* mostrou-se inadequado para previsão de diabetes, pois a sua dependência da geometria euclidiana não se alinha com as fronteiras clínicas definidas biologicamente. A abordagem não-supervisionada não se recomenda como método primário para classificação médica neste contexto.

# 7 Comparação entre Modelo Multiclasse e Modelo Binário

Com o objetivo de reduzir o erro de classificação, em particular o número de falsos negativos, foi considerada uma abordagem alternativa baseada na simplificação do atributo *target*. Nesta abordagem, as classes Pré-diabetes e Diabetes foram agregadas numa única classe designada por Condição Diabética, resultando num problema de classificação binária, em oposição à classificação multiclasse inicialmente adotada.

***Motivação para a abordagem binária***

Na classificação multiclasse, o modelo é obrigado a distinguir entre estados metabólicos clinicamente próximos, como Prediabetes e Diabetes, o que aumenta a probabilidade de confusão entre estas classes. Embora tais erros possam ser aceitáveis do ponto de vista estatístico, do ponto de vista clínico são menos críticos do que a classificação incorreta de um indivíduo com alteração metabólica como Standard. Ao agregar as classes Pré-diabetes e Diabetes numa única categoria, o modelo passa a focar-se na distinção entre:

- indivíduos metabolicamente normais (Standard);
- indivíduos com algum grau de alteração glicémica (Condição Diabética).

Esta reformulação permite reduzir significativamente o risco de falsos negativos, isto é, casos em que pacientes com alteração metabólica são incorretamente classificados como saudáveis.

***Vantagens da substituição para classificação binária***

A adoção da classificação binária apresenta várias vantagens relevantes:

- **Redução dos falsos negativos**: ao eliminar a distinção final entre Pré-diabetes e Diabetes, o modelo torna-se mais sensível à deteção de qualquer condição diabética, reduzindo a probabilidade de não identificar indivíduos em risco.
- **Maior sensibilidade clínica**: em contextos médicos, é preferível identificar um paciente como potencialmente diabético e encaminhá-lo para avaliação adicional do que classificá-lo incorretamente como saudável.
- **Simplificação do problema de aprendizagem**: a redução do número de classes diminui a complexidade do modelo, facilitando o processo de aprendizagem e melhorando a estabilidade das previsões.
- **Melhoria da interpretabilidade**: os resultados tornam-se mais fáceis de interpretar, especialmente em sistemas de apoio à decisão clínica, nos quais a distinção entre "risco" e "não risco" é frequentemente mais relevante.
- **Melhoria potencial das métricas globais**: a simplificação pode conduzir a um aumento da *accuracy*, da sensibilidade (*recall*) e do valor da AUC, especialmente quando existe sobreposição entre as classes originais.

***Implementação no pré-processamento dos dados***

Esta alteração foi realizada durante a fase de pré-processamento, recorrendo ao nodo ***Rule Engine*** da plataforma *KNIME*. A regra definida permitiu mapear as classes Pré-diabetes e Diabetes para uma única categoria binária, mantendo a classe Standard inalterada.

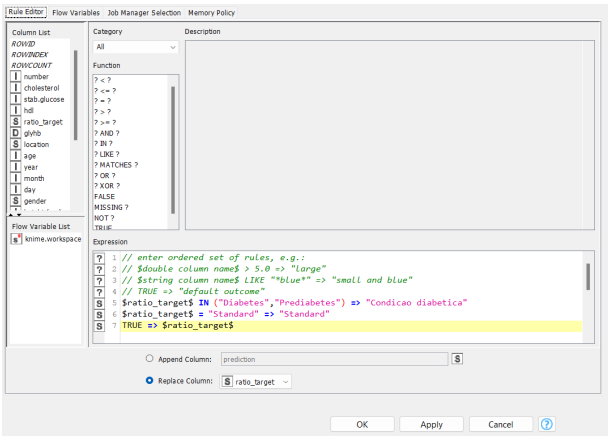



Figura 50: Nodo *Rule Engine* e a sua configuração

Importa salientar que esta foi a única modificação introduzida no pré-processamento dos dados, garantindo que a comparação entre os modelos multiclasse e binário mantém-se justa e consistente, uma vez que todas as restantes etapas de preparação dos dados e configuração dos modelos permaneceram inalteradas.

## 7.1 Comparação do desempenho das Redes Neuronais (RProp): classificação binária vs multiclasse, com e sem *threshold*

A análise comparativa entre as abordagens binária e multiclasse das Redes Neuronais (RProp) evidencia que o melhor desempenho é obtido na classificação binária com aplicação de *thresholds*. Sem *thresholds*, o modelo binário já apresenta um desempenho superior ao multiclasse (93,293% $>$ 92,073%), registando um menor número de classificações incorretas (11 $<$ 13) e maior estabilidade na distinção entre indivíduos standard e indivíduos com alteração metabólica.

Com a introdução dos *thresholds*, o modelo binário registou uma redução adicional do erro total, atingindo o melhor resultado observado para este algoritmo (4,878%). Esta melhoria refletiu-se sobretudo na diminuição dos falsos positivos (10 para 7) clinicamente mais relevantes, reforçando a adequação do modelo para contextos de apoio à decisão clínica.

No modelo multiclasse, a aplicação de *thresholds* não conduziu a uma redução do erro total, mantendo-se o número de classificações incorretas. No entanto, verificou-se uma alteração qualitativa na distribuição dos erros, com a introdução da classe "incerto", permitindo identificar explicitamente situações ambíguas em que o modelo não apresenta confiança suficiente para atribuir uma classe clínica específica.

Assim, conclui-se que as Redes Neuronais (RProp) binárias com *thresholds* apresentam o melhor compromisso entre desempenho e segurança clínica. Em contraste, a abordagem multiclasse, embora ofereça maior granularidade diagnóstica, demonstra maior sensibilidade a ambiguidades entre classes metabolicamente próximas, sendo menos robusta em termos de erro global.

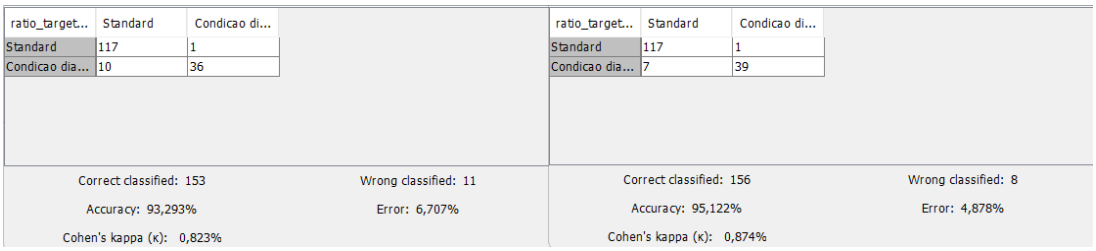



Figura 51: Nodo *Scorer* do modelo Redes Neuronais (RProp) binário sem e com *threshold*

## 7.2 Comparação do desempenho das Árvores de Decisão: classificação binária vs multiclasse, com e sem *threshold*

A análise comparativa do modelo de Árvore de Decisão revela diferenças marcadas entre as abordagens binária e multiclasse. Na classificação binária, o modelo apresentou o melhor desempenho global entre todos os modelos analisados (98,78%), atingindo o menor erro total após a aplicação de *thresholds* (1,22%), com especial destaque para a redução dos falsos negativos (3 para 1) clinicamente mais críticos.

Por outro lado, na abordagem multiclasse, a Árvore de Decisão demonstrou uma menor sensibilidade à aplicação de *thresholds*. Apesar de várias tentativas de ajustamento, não se observaram melhorias significativas no erro total nem na redução dos erros mais críticos, sendo a configuração inicial considerada a mais adequada.

Desta forma, a Árvore de Decisão binária revelou-se particularmente eficaz e robusta, beneficiando da sua natureza baseada em regras explícitas. Em contraste, a abordagem multiclasse, embora interpretável, mostrou-se menos flexível e menos eficaz na distinção entre classes metabolicamente próximas.

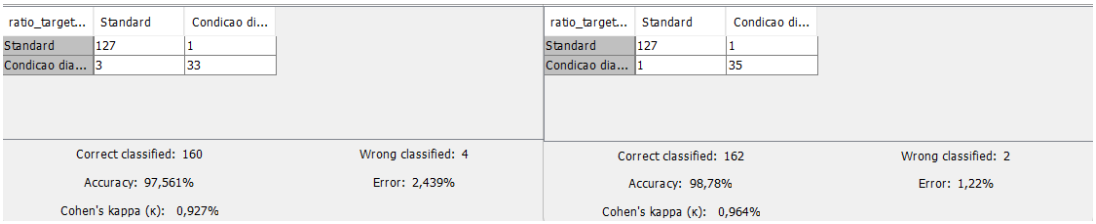



Figura 52: Nodo *Scorer* do modelo Árvore de Decisão binário sem e com *threshold*

## 7.3 Comparação do desempenho da Regressão Logística: classificação binária vs multiclasse, com e sem *threshold*

A análise comparativa evidencia que a classificação binária apresenta desempenho superior à multiclasse, com aplicação de threshold. Sem threshold, o modelo binário alcança uma *accuracy* de 86,585%, superando o modelo multiclasse (85,976%), com uma redução do erro global (de 14,024% para 13,415%), refletindo sobretudo uma diminuição do número de falsos negativos.

A aplicação de threshold probabilístico melhora o desempenho em ambas as abordagens. No modelo multiclasse, a *accuracy* aumenta para 89,634% e o coeficiente de Cohen passa de 0,65 para 0,764, indicando maior concordância. No modelo binário, o impacto é mais expressivo, com a *accuracy* a atingir 92,683%, o erro a reduzir-se para 7,317% e um coeficiente de Cohen de 0,821, correspondente a uma boa concordância.

De forma global, conclui-se que a Regressão Logística binária com *thresholds* apresenta o melhor compromisso entre desempenho, robustez e segurança clínica, enquanto a abordagem multiclasse, apesar de fornecer maior granularidade diagnóstica, revela limitações na separação consistente entre múltiplas classes.

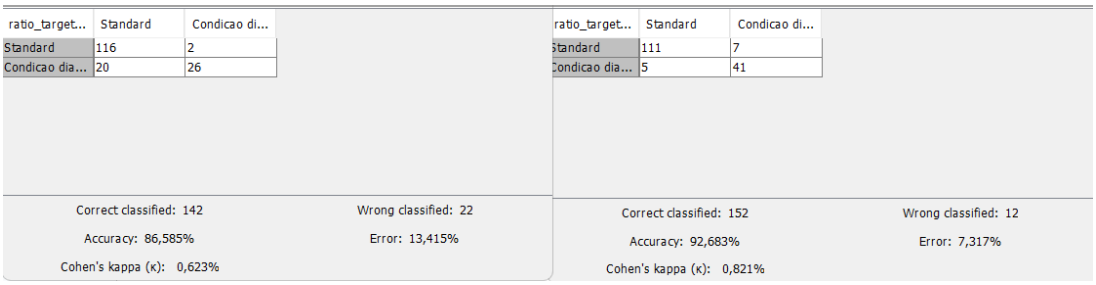



Figura 53: Nodo *Scorer* do modelo Regressão Logística binário sem e com *threshold*

## 7.4 Comparação do desempenho do modelo Random Forest: multiclasse vs binário, com e sem *threshold*

A análise comparativa evidencia que a classificação binária apresenta desempenho superior à multiclasse, tanto com como sem aplicação de *threshold*. Sem *threshold*, o modelo binário alcança uma *accuracy* de 93,293%, superando o modelo multiclasse (87,805%), com uma redução significativa do erro global (de 12,195% para 6,707%), refletindo sobretudo uma diminuição do número de falsos negativos.

A aplicação de *threshold* probabilístico melhora o desempenho em ambas as abordagens. No modelo multiclasse, a *accuracy* aumenta para 90,854% e o coeficiente de Cohen passa de 0,69 para 0,785, indicando maior concordância. No modelo binário, o impacto é mais expressivo, com a *accuracy* a atingir 96,341%, o erro a reduzir-se para 3,659% e um coeficiente de Cohen de 0,909, correspondente a uma concordância quase perfeita.

De forma global, a abordagem binária com *threshold* apresenta o melhor compromisso entre desempenho e segurança clínica, ao maximizar a deteção de indivíduos com alteração metabólica. Em contrapartida, a abordagem multiclasse oferece maior granularidade diagnóstica, embora com menor robustez na distinção entre classes metabolicamente próximas.

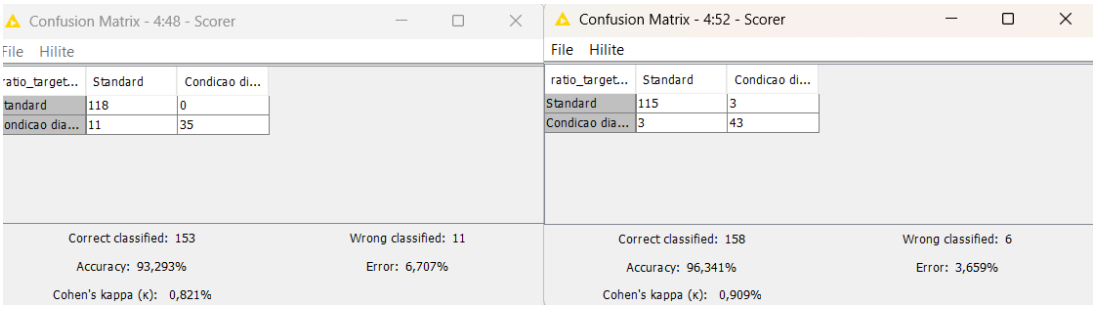



Figura 54: Nodo *Scorer* do modelo *Random Forest* binário sem e com *threshold*

# 8 Seleção do algoritmo mais preciso

Uma etapa fundamental deste trabalho consiste na análise comparativa dos resultados obtidos pelos diferentes algoritmos de aprendizagem automática implementados. Desta forma, considerando exclusivamente os modelos de classificação multiclasse, conclui-se que o algoritmo que apresentou o melhor desempenho global foi o modelo de Árvore de Decisão.

Após a avaliação das métricas de desempenho e da análise das matrizes de confusão dos vários modelos multiclasse testados, verificou-se que a Árvore de Decisão apresentou o menor erro total (6,707%), bem como uma distribuição de erros mais coerente do ponto de vista clínico. Em particular, este modelo demonstrou uma maior capacidade de distinguir corretamente entre indivíduos standard, pré-diabéticos e diabéticos, reduzindo classificações clinicamente mais críticas quando comparado com os restantes algoritmos.

Para além do desempenho quantitativo, a Árvore de Decisão destacou-se também pela sua estabilidade face à aplicação de *thresholds*, uma vez que alterações nos parâmetros não conduziram a degradações significativas do desempenho global. Esta característica revela uma maior robustez do modelo, sobretudo quando comparada com modelos probabilísticos, que apresentaram maior sensibilidade a ambiguidades entre classes metabolicamente próximas.

Adicionalmente, a natureza interpretável da Árvore de Decisão constitui uma vantagem relevante no contexto biomédico, permitindo compreender de forma explícita as regras de decisão utilizadas pelo modelo. Esta transparência facilita a validação clínica dos resultados e reforça a confiança na utilização do modelo como ferramenta de apoio à decisão.

Assim, tendo em conta o conjunto das métricas analisadas, a coerência clínica dos resultados e a interpretabilidade do modelo, conclui-se que a Árvore de Decisão é o algoritmo mais preciso e adequado para a base de dados e o problema em estudo.

# 9 Conclusão

O presente trabalho teve como objetivo o desenvolvimento e avaliação de um sistema de apoio à decisão clínica para a classificação do estado glicémico de indivíduos, recorrendo a técnicas de Inteligência Artificial e Aprendizagem Automática, implementadas na plataforma *KNIME Analytics Platform*. Para o efeito, foi utilizada uma base de dados biomédica composta por variáveis clínicas, antropométricas e laboratoriais relevantes para o estudo da diabetes.

A análise exploratória e o pré-processamento dos dados revelaram-se etapas fundamentais, permitindo identificar e corrigir inconsistências, tratar valores em falta, normalizar variáveis e preparar adequadamente os dados para a fase de modelação. Estas etapas contribuíram de forma significativa para a melhoria da qualidade dos dados e para o desempenho dos modelos desenvolvidos. Foram implementados diversos algoritmos de aprendizagem automática, incluindo Redes Neuronais Artificiais (RProp), Árvores de Decisão, Regressão Logística, *Random Forest* e o método não supervisionado Segmentação. Os resultados evidenciaram que os modelos supervisionados apresentaram um desempenho claramente superior ao da Segmentação, o qual se mostrou inadequado para classificação clínica neste contexto.

A aplicação de *thresholds* probabilísticos revelou-se uma estratégia eficaz na redução de erros clinicamente críticos, permitindo introduzir a classe "incerto" em situações de maior ambiguidade. Adicionalmente, a reformulação do problema para uma abordagem de classificação binária demonstrou melhorias consistentes ao nível da *accuracy*, da sensibilidade e da concordância global, reduzindo o risco de falsos negativos e aumentando a segurança clínica das previsões.

Considerando os modelos multiclasse, a Árvore de Decisão destacou-se como o algoritmo com melhor desempenho global, tendo um baixo erro de classificação, estabilidade e elevada interpretabilidade, uma característica particularmente relevante em aplicações biomédicas. De forma global, os resultados obtidos demonstram o potencial da Aprendizagem Automática como ferramenta de apoio à decisão clínica na identificação de alterações metabólicas associadas à diabetes.