Universidade do Minho
Departamento de Informática

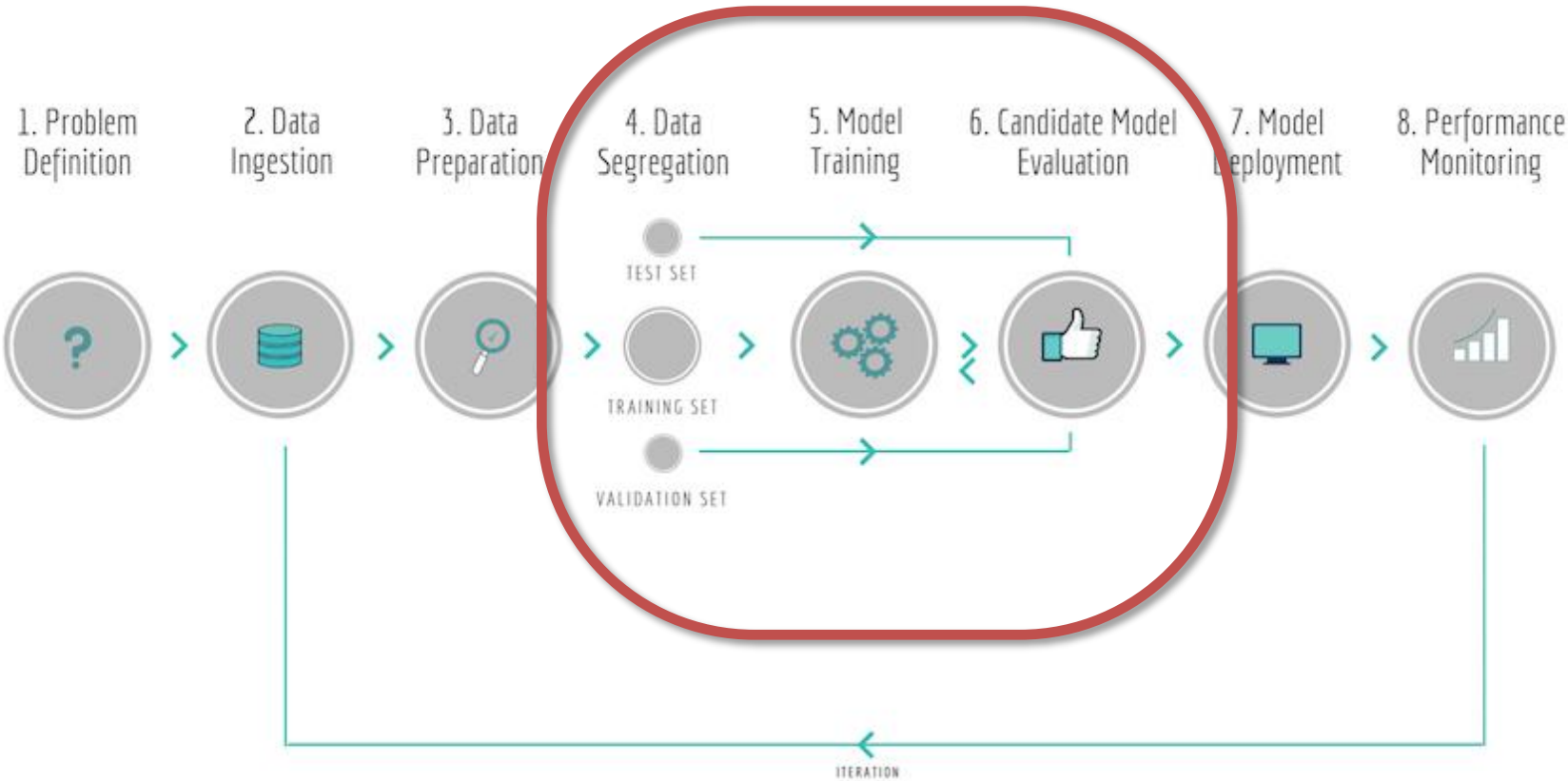
Redes Neuronais Artificiais @ KNIME

LEI/MiEI @ 2024/2025, 2º sem

Agenda

- O fluxo Learner-Predictor para Redes Neuronais em KNIME
- MLP: Multi-Layer Perceptron em KNIME
- Experimentação
  (*hands on*)
- Outros nodos KNIME

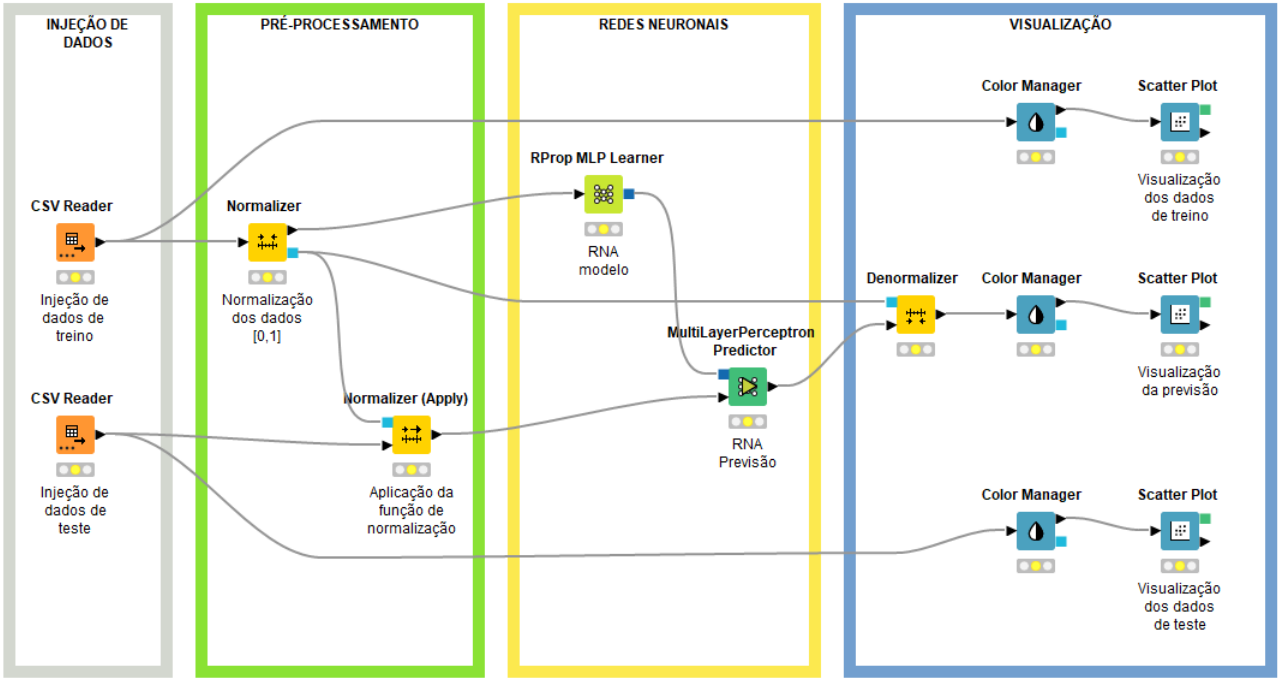
A Machine Learning Pipeline



(https://towardsdatascience.com/architecting-a-machine-learning-pipeline-a847f094d1c7)

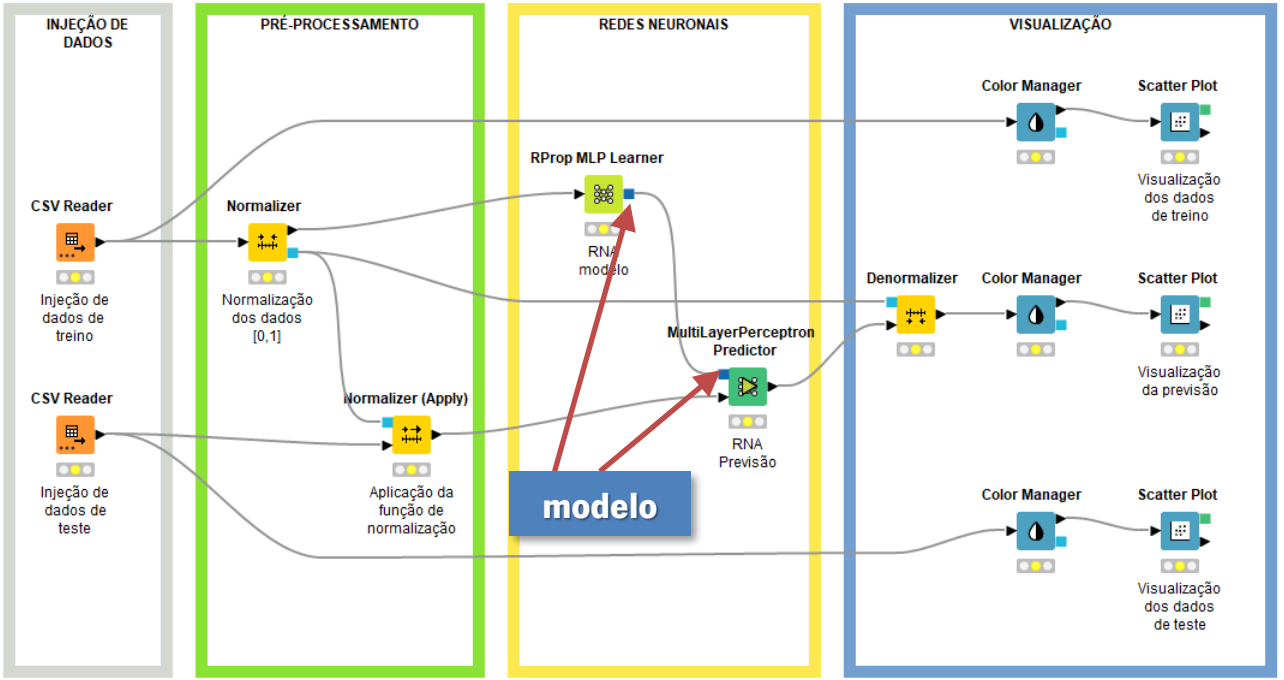
Fluxo de Redes Neuronais Artificiais @ KNIME (Rprop Multi-layer perceptron)

- Em KNIME, o recurso a técnicas de aprendizagem suportadas por Redes Neuronais Artificiais é implementada por uma sequência de nodos *learner* e de *predictor*.



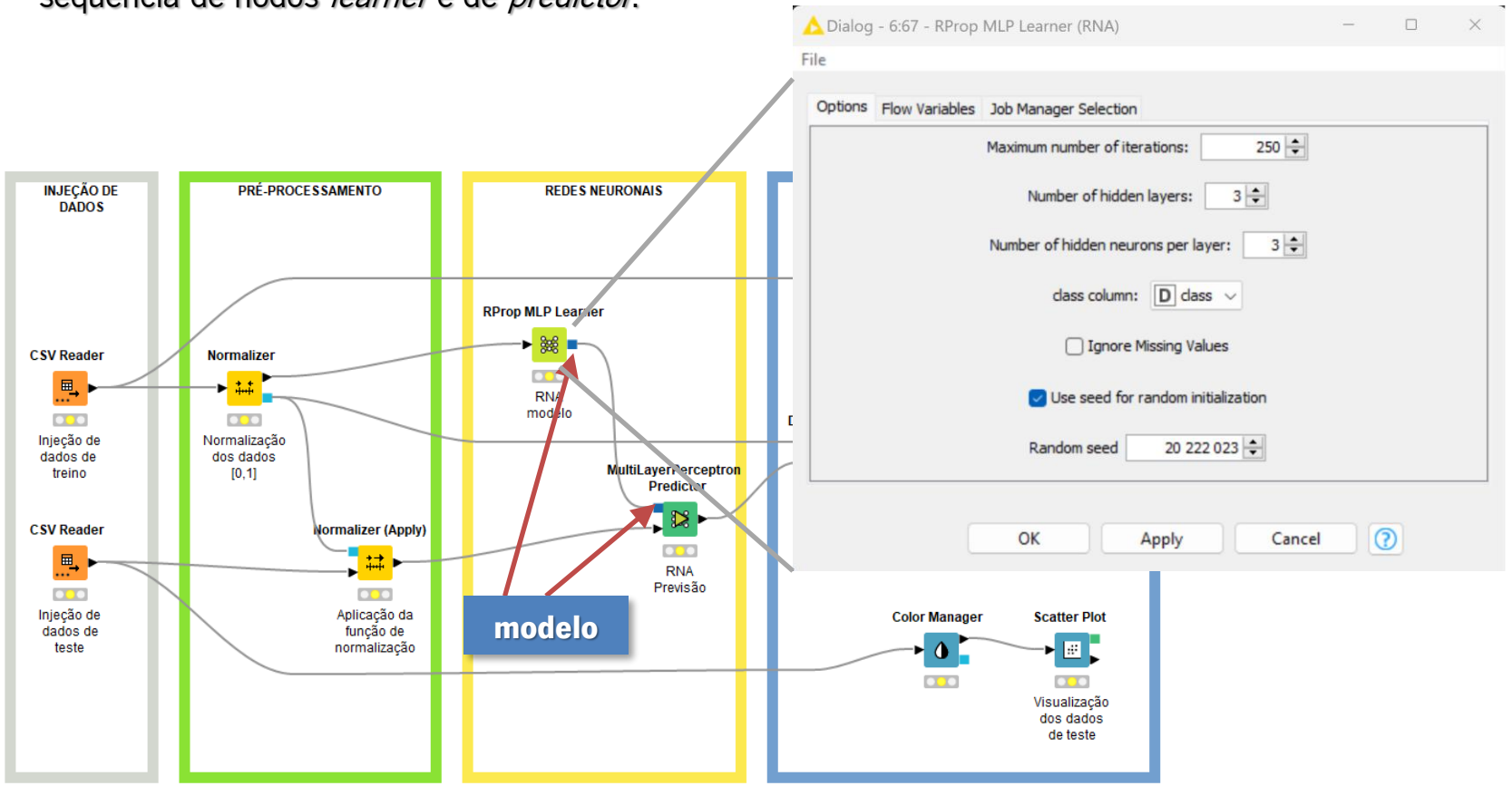
Fluxo de Redes Neuronais Artificiais @ KNIME (Rprop Multi-layer perceptron)

- Em KNIME, o recurso a técnicas de aprendizagem suportadas por Redes Neuronais Artificiais é implementada por uma sequência de nodos *learner* e de *predictor*.



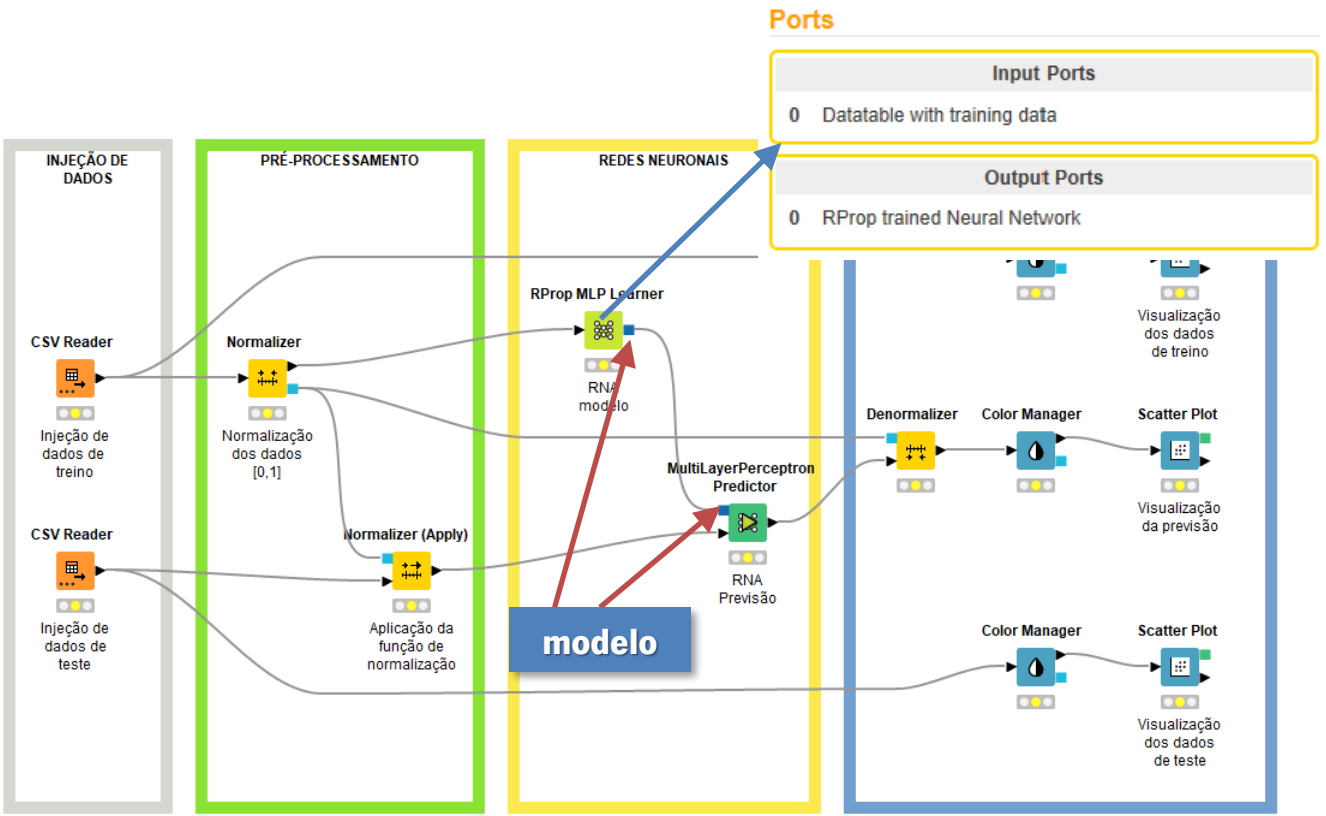
Fluxo de Redes Neuronais Artificiais @ KNIME (Rprop Multi-layer perceptron)

- Em KNIME, o recurso a técnicas de aprendizagem suportadas por Redes Neuronais Artificiais é implementada por uma sequência de nodos *learner* e de *predictor*.



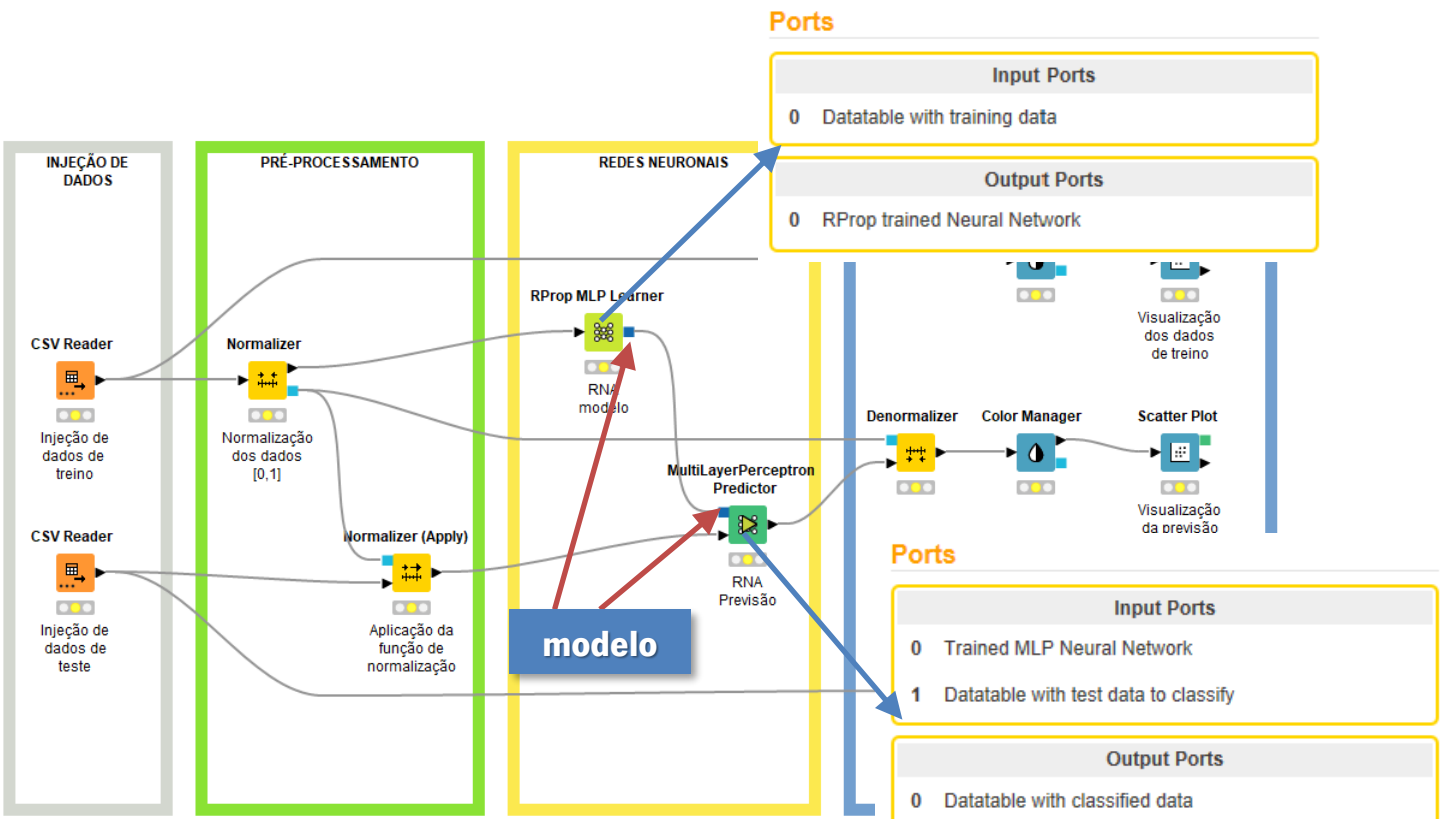
Fluxo de Redes Neuronais Artificiais @ KNIME (Rprop Multi-layer perceptron)

- Em KNIME, o recurso a técnicas de aprendizagem suportadas por Redes Neuronais Artificiais é implementada por uma sequência de nodos *learner* e de *predictor*.



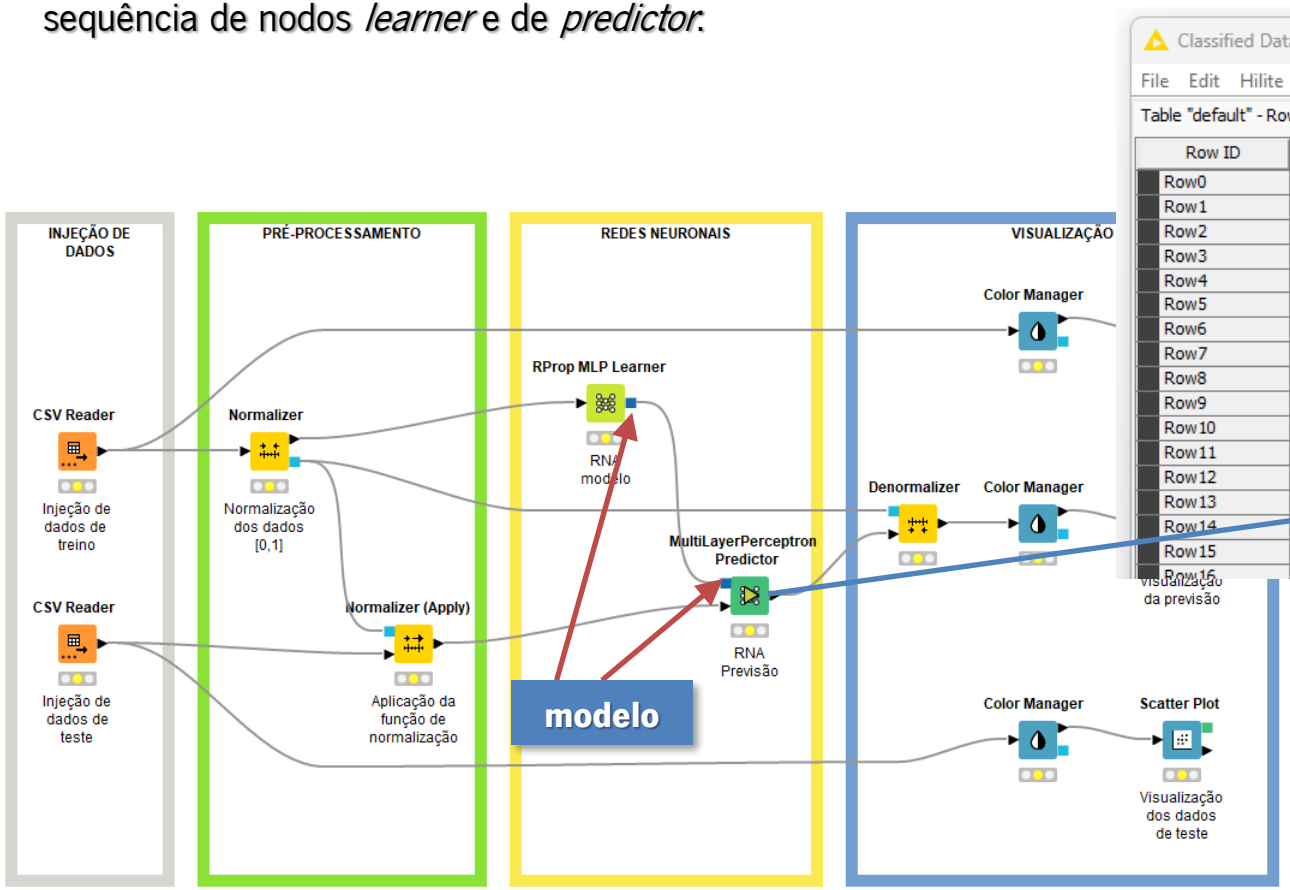
Fluxo de Redes Neuronais Artificiais @ KNIME (Rprop Multi-layer perceptron)

- Em KNIME, o recurso a técnicas de aprendizagem suportadas por Redes Neuronais Artificiais é implementada por uma sequência de nodos *learner* e de *predictor*.



Fluxo de Redes Neuronais Artificiais @ KNIME (Rprop Multi-layer perceptron)

- Em KNIME, o recurso a técnicas de aprendizagem suportadas por Redes Neuronais Artificiais é implementada por uma sequência de nodos *learner* e de *predictor*.



Classified Data - 6:64 - MultiLayerPerceptron Predictor (RNA)

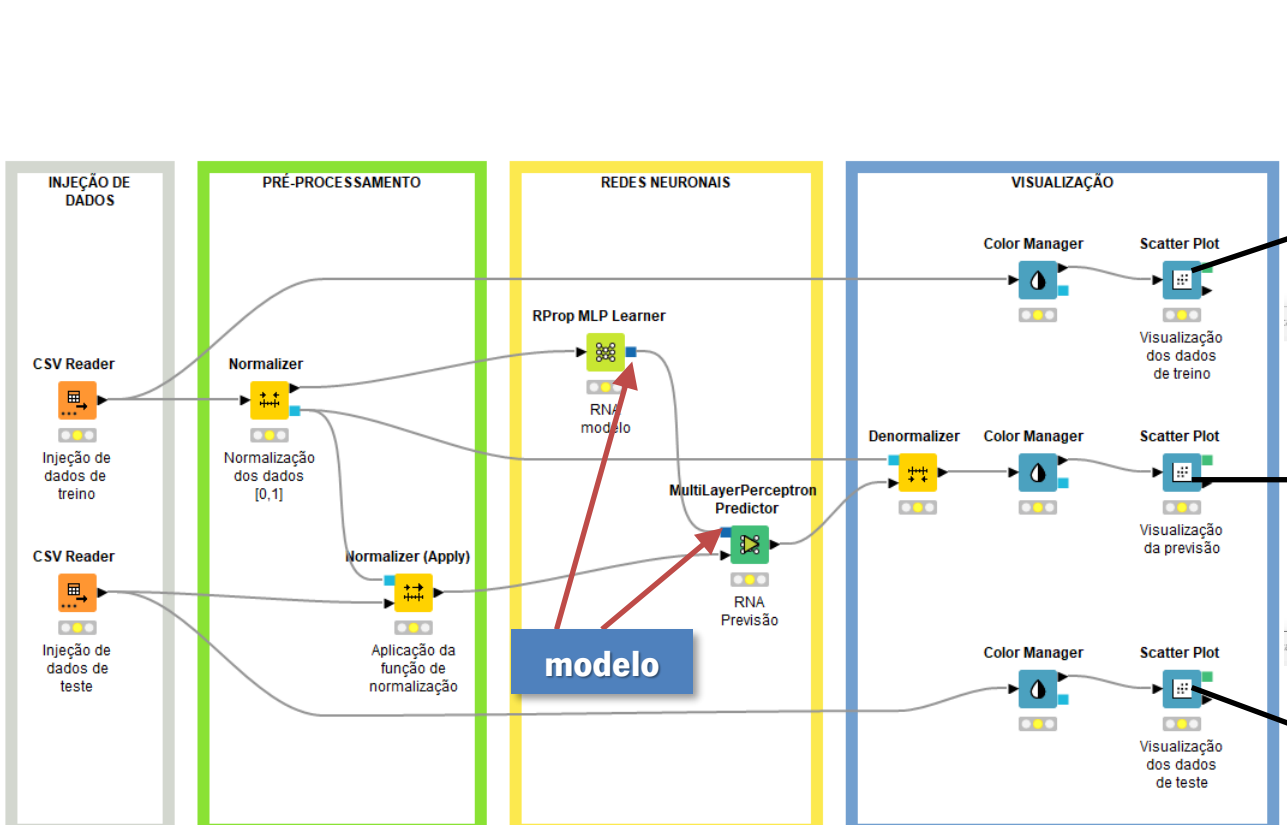
File Edit Hilite Navigation View

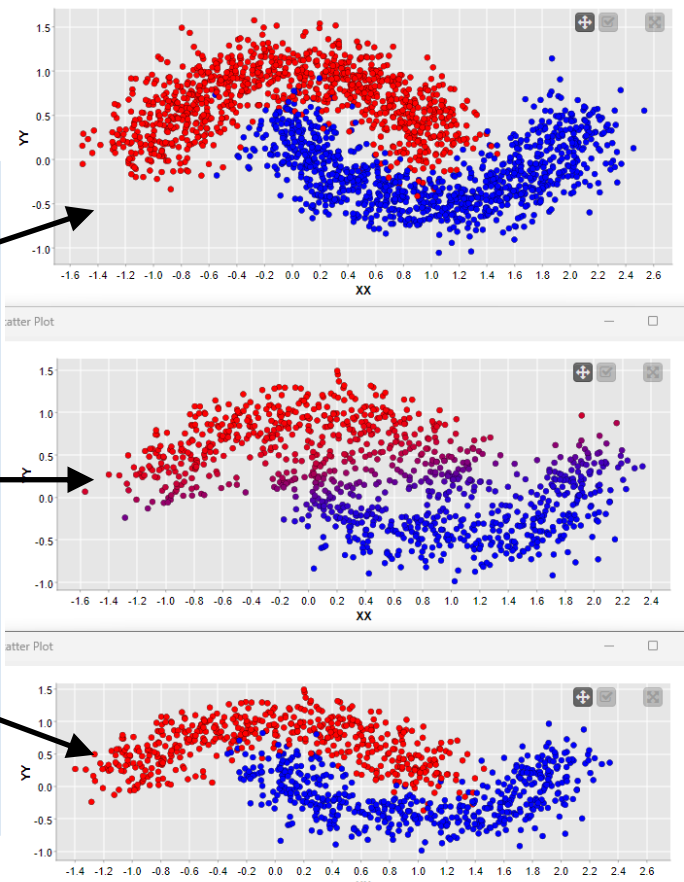
Table "default" - Rows: 1000 | Spec - Columns: 4 | Properties | Flow Variables

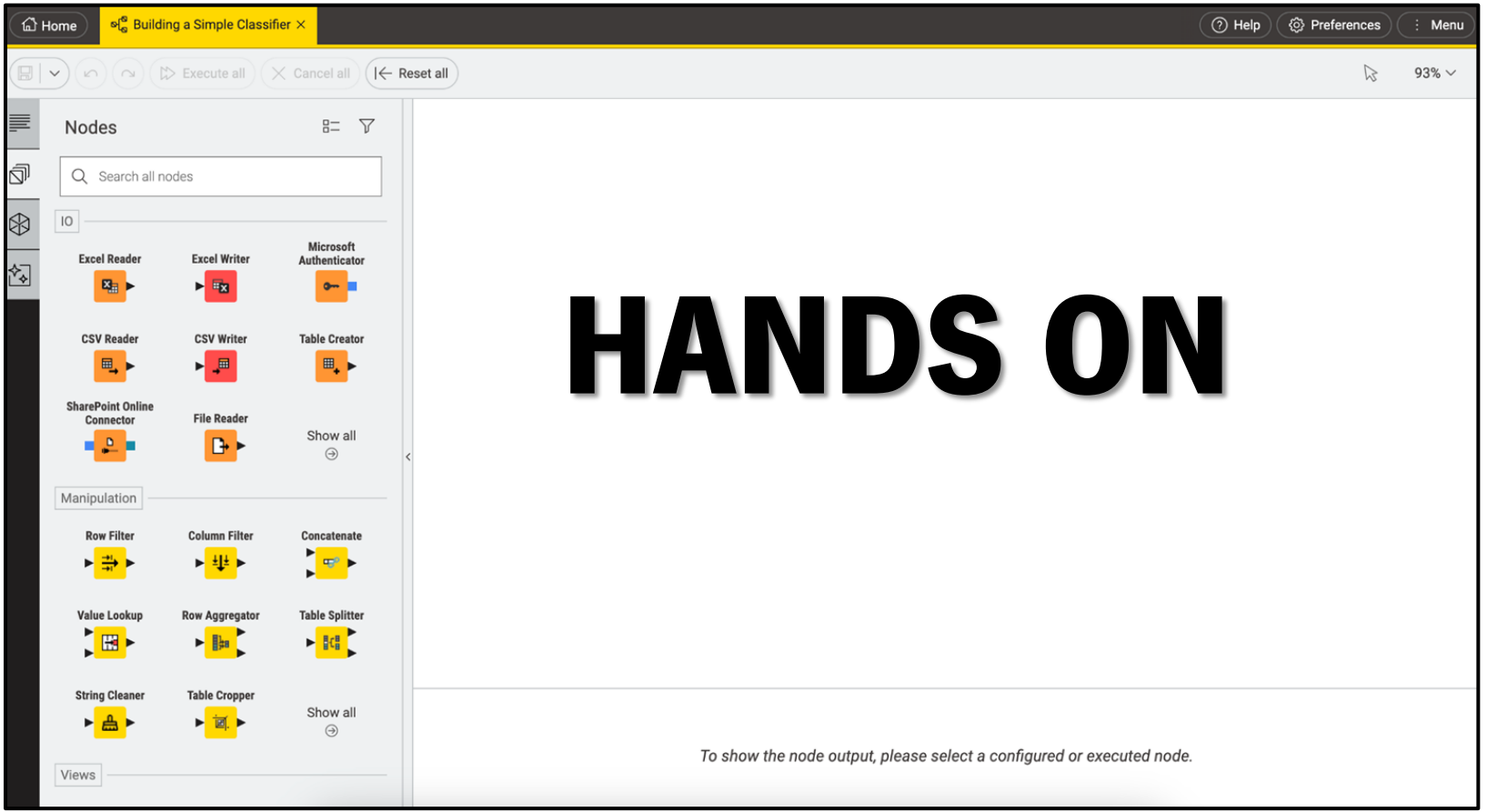
Row ID	XX	YY	class	Prediction (class)
Row0	0.25	0.663	0	0.046
Row1	0.421	0.271	1	0.923
Row2	0.62	0.653	0	0.23
Row3	0.119	0.531	0	0.102
Row4	0.383	0.083	1	0.987
Row5	0.346	0.683	0	0.056
Row6	0.514	0.258	1	0.957
Row7	0.381	0.426	1	0.572
Row8	0.861	0.505	1	0.82
Row9	0.549	0.493	0	0.568
Row10	0.465	0.352	1	0.833
Row11	0.12	0.432	0	0.271
Row12	0.594	0.239	1	0.975
Row13	0.172	0.582	0	0.127
Row14	0.4	0.353	1	0.782
Row15	0.539	0.753	0	0.063
Row16	0.533	0.738	0	0.071

Fluxo de Redes Neuronais Artificiais @ KNIME (Rprop Multi-layer perceptron)

- Em KNIME, o recurso a técnicas de aprendizagem suportadas por Redes Neuronais Artificiais é implementada por uma sequência de nodos *learner* e de *predictor*.



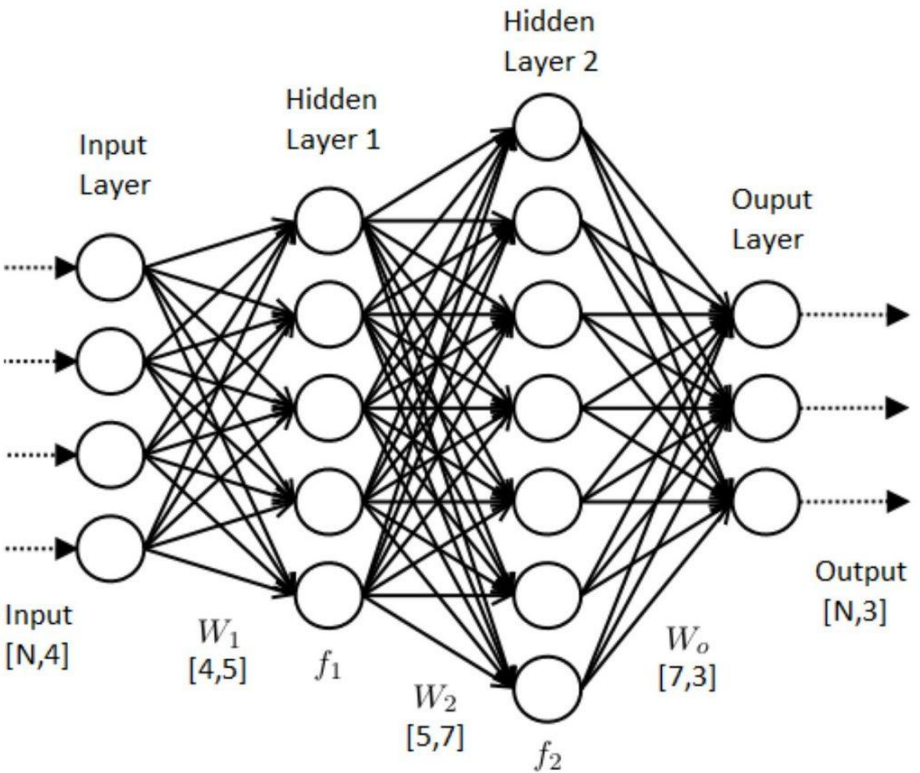



Home
Building a Simple Classifier
Help
Preferences
Menu
Execute all
Cancel all
Reset all
93%
Nodes
Search all nodes
IO
Excel Reader
Excel Writer
Microsoft Authenticator
CSV Reader
CSV Writer
Table Creator
SharePoint Online Connector
File Reader
Show all
Manipulation
Row Filter
Column Filter
Concatenate
Value Lookup
Row Aggregator
Table Splitter
String Cleaner
Table Cropper
Show all
Views
HANDS ON
To show the node output, please select a configured or executed node.

O que são Redes Neuronais Artificiais?

- As Redes Neurais Artificiais são um modelo computacional que consiste em vários elementos de processamento que recebem entradas e entregam saídas com base nas funções de ativação e transferência definidas.

- As Redes Neurais Artificiais podem ser aplicadas tanto em problemas de Regressão como de Classificação.



RNA: Problema de Classificação

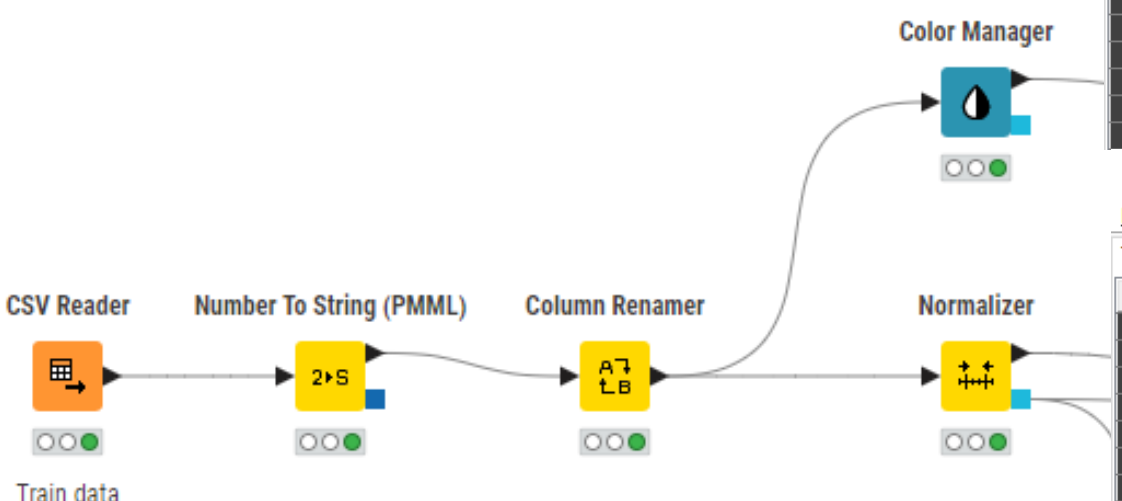
- Considere-se o desenvolvimento e teste de uma solução baseada em aprendizagem automática para um problema de classificação binária:
  - classificar os casos como 'moon' ou 'not moon' a partir das suas características;

- O *workflow* mostra como criar um Multilayer Perceptron com uma camada 'softmax' para classificação;

- Neste exemplo, o MLP é usado para classificar um conjunto de dados simples com dois atributos;

Carregar e Visualizar os Dados de Treino

- Preparar os dados de treino:
  - Class: classe numérica binária (converter para *string*)
  - XX, YY: normalizar atributos *double*;
- Visualizar a distribuição de dados por classe:
  - Classe 0: azul
  - Classe 1: vermelho



File Table - 3:1 - File Reader (deprecated) (Training Data)

File Edit Hilite Navigation View

Table "moon_data_train.csv" - Rows: 2000 | Spec - Columns: 3 | Properties | Flow Variables

Row ID	I Col0	D Col1	D Col2
Row0	1	0.611	-0.46
Row1	1	-0.556	0.731
Row2	0	-0.585	0.663
Row3	1	1.198	-0.695
Row4	0	-0.555	0.797
Row5	0	-0.524	0.883
Row6	1	0.624	-0.782
Row7	0	0.06	0.911
Row8	0	0.443	0.648
Row9	0	0.629	0.893
Row10	0	-0.794	0.244

Normalized table - 0:41 - Normalizer

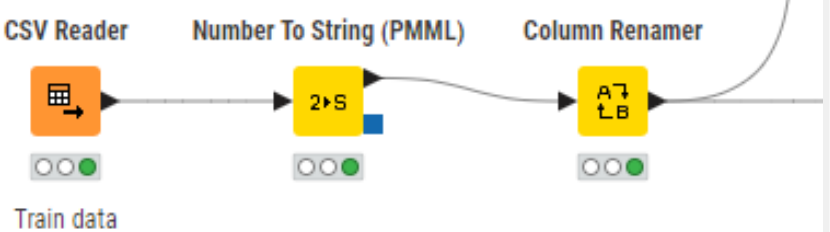
File Edit Hilite Navigation View

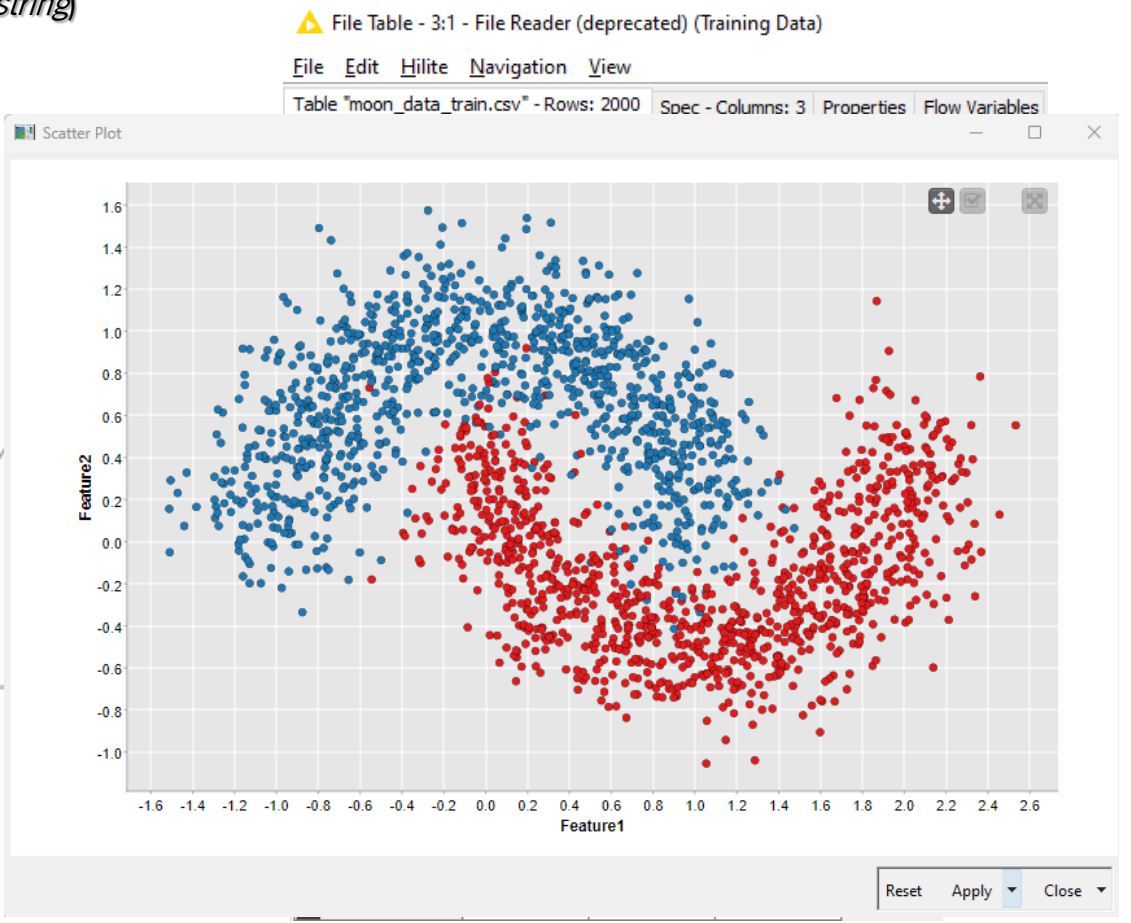
Table "default" - Rows: 2000 | Spec - Columns: 3 | Properties | Flow Variables

Row ID	S Label	D Feature1	D Feature2
Row0	1	0.525	0.226
Row1	1	0.236	0.679
Row2	0	0.229	0.654
Row3	1	0.67	0.137
Row4	0	0.237	0.704
Row5	0	0.244	0.737
Row6	1	0.528	0.104
Row7	0	0.389	0.748
Row8	0	0.483	0.648
Row9	0	0.529	0.741
Row10	0	0.177	0.494

Carregar e Visualizar os Dados de Treino

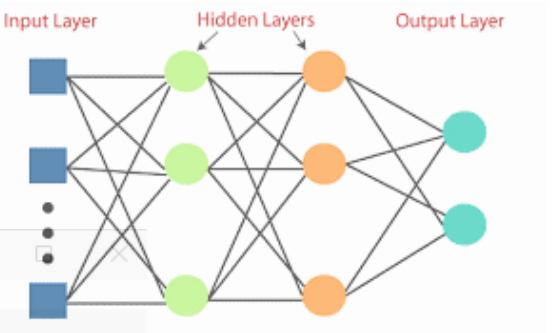
- Preparar os dados de treino:
  - Class: classe numérica binária (converter para *string*)
  - XX, YY: normalizar atributos *double*;
- Visualizar a distribuição de dados por classe:
  - Classe 0: azul
  - Classe 1: vermelho

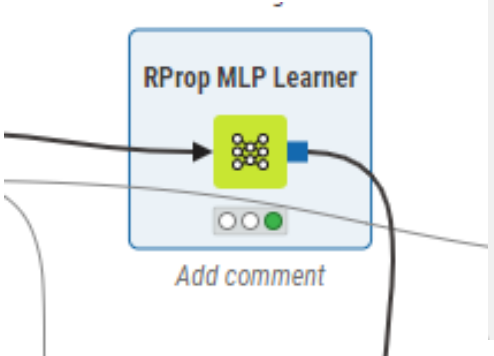


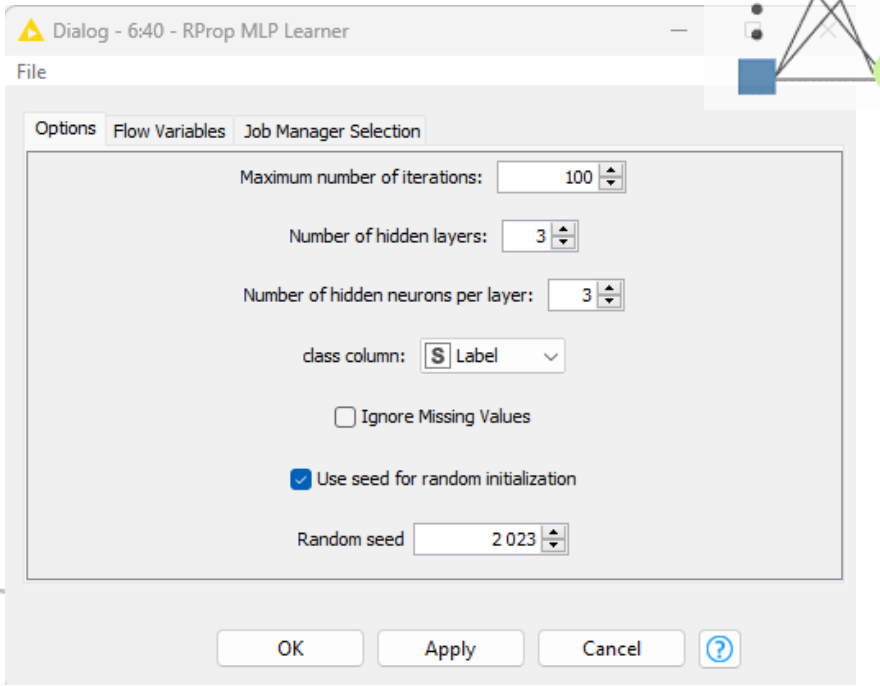


Definir e Treinar uma RNA (MLP)

- Criar um Multi-Layer Perceptron, composto por três camadas totalmente ligadas:
  - Número de iterações: 100
  - Número de camadas ocultas: 3
  - Número de neurónios ocultos por camada: 3
  - Coluna de classe: «label»
  - *Seed* aleatória: 2023

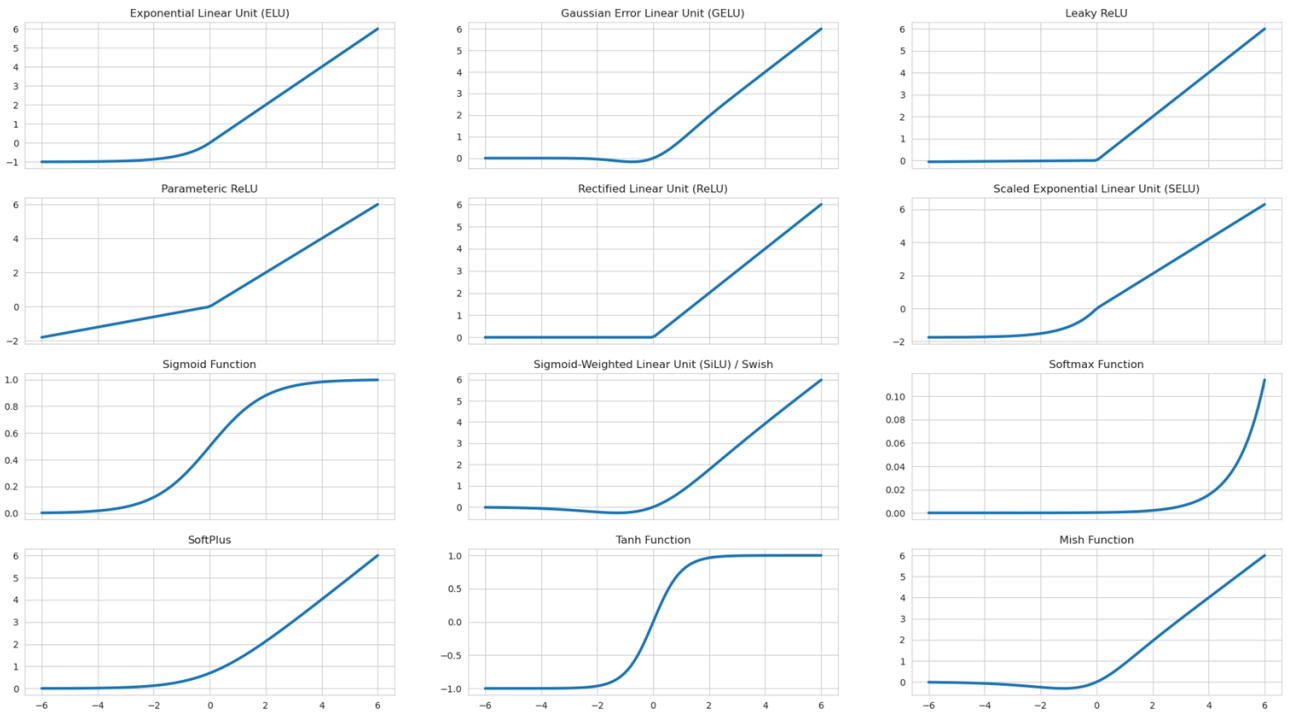


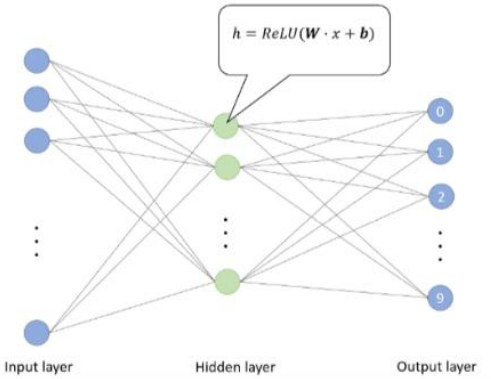


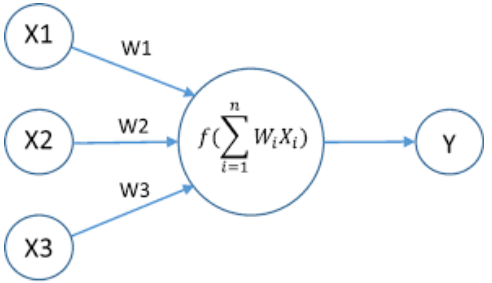


Definir e Treinar uma RNA (MLP)

- Funções de transferência (*activation*)







Carregar os Dados de Teste

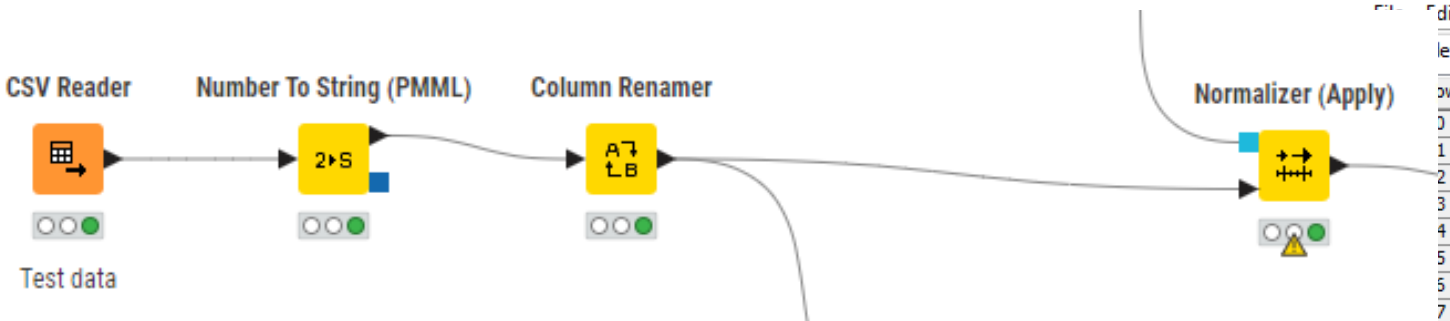
- Carregar e preparar os dados de teste
  (de acordo com as regras usadas para os dados de treino):
  - Class: classe numérica binária (converter para *string*)
  - XX, YY: normalizar atributos *double*;

Renamed/Retyped table - 3:36 - Column Rename

File Edit Hilite Navigation View

Table "default" - Rows: 1000 | Spec - Columns: 3 | Properties | Flow Variables

Row ID	S Label	D Feature1	D Feature2
Row0	0	-0.501	0.687
Row1	1	0.19	-0.341
Row2	0	0.995	0.663
Row3	0	-1.031	0.342
Row4	1	0.038	-0.837
Row5	0	-0.114	0.74
Row6	1	0.568	-0.376
Row7	1	0.029	0.066
Row8	1	1.971	0.274
Row9	0	0.709	0.241
Row10	1	0.37	-0.128



Normalized output - 0:42 - Normalizer (Apply)

dit Hilite Navigation View

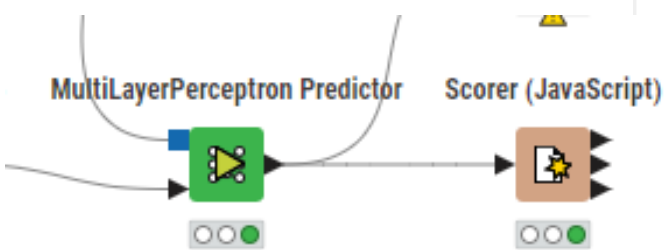
efault" - Rows: 1000 | Spec - Columns: 3 | Properties | Flow Variables

ow ID	S Label	D Feature1	D Feature2
0	0	0.25	0.663
1	1	0.421	0.271
2	0	0.62	0.653
3	0	0.119	0.531
4	1	0.383	0.083
5	0	0.346	0.683
6	1	0.514	0.258
7	1	0.381	0.426
Row8	1	0.861	0.505
Row9	0	0.549	0.493
Row10	1	0.465	0.352

Prever e Avaliar

- Após o término do treino, usar o nodo «Multi-Layer Perceptron Predictor» para calcular as previsões;

- Aplicar o nodo «Scorer» para avaliar o desempenho da classificação do modelo MLP;



Confusion Matrix

Scorer View

Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	428	69	86.12%
1 (Actual)	70	433	86.08%
	85.94%	86.25%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (ð□□¹)	Correctly Classified	Incorrectly Classified
86.10%	13.90%	0.722	861	139

RNA: Boas Práticas

- Em relação às Redes Neurais Artificiais, ter em consideração as seguintes boas práticas:
  - São exigentes:
    - Preferir dados dimensionados!
    - Normalizar sempre que adequado!
  - Ajustar os diversos parâmetros:
    - número de iterações;
    - número de camadas ocultas;
    - número de neurónios ocultos por camada;
  - Não esquecer de usar uma semente aleatória específica;