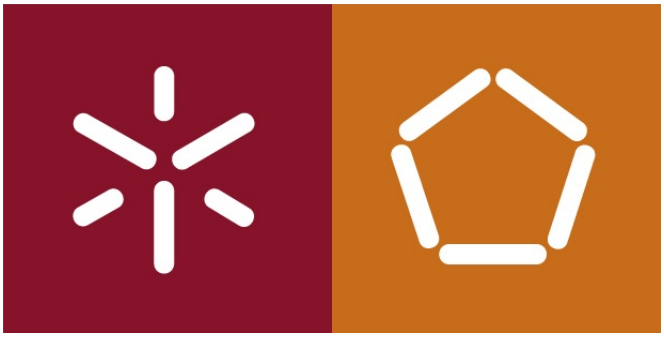


Universidade do Minho
Departamento de Informática

Sistemas Baseados em Similaridade

Trabalho Prático Individual 5

Gonçalo Almeida (A84610)

Novembro 2020

1 Tarefa 1

Após o carregamento e união dos *datasets*, explorei os dados através de nodos estatísticos.

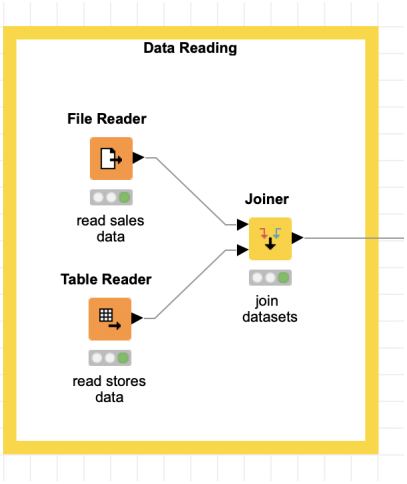


Figure 1: Carregamento e união dos datasets

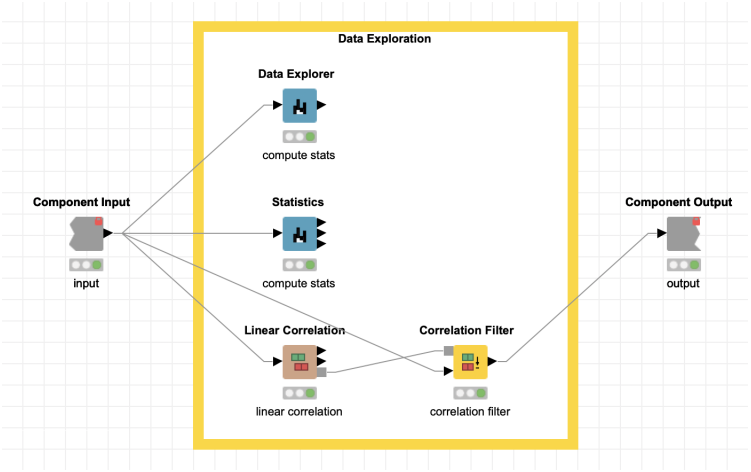


Figure 2: Exploração dos dados

O *workflow* foi desenvolvido dentro de um *component* de modo a poder obter todas as *views* numa só página.

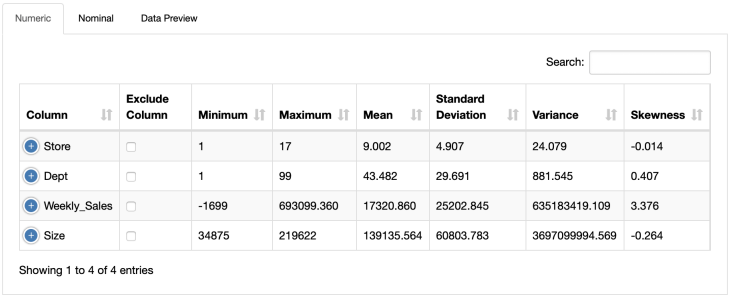


Figure 3: Vista gráfica interativa

2 Tarefa 2

Comecei por substituir os valores booleanos pelos respetivos valores binários e transformar o seu tipo *string* no tipo numérico inteiro.

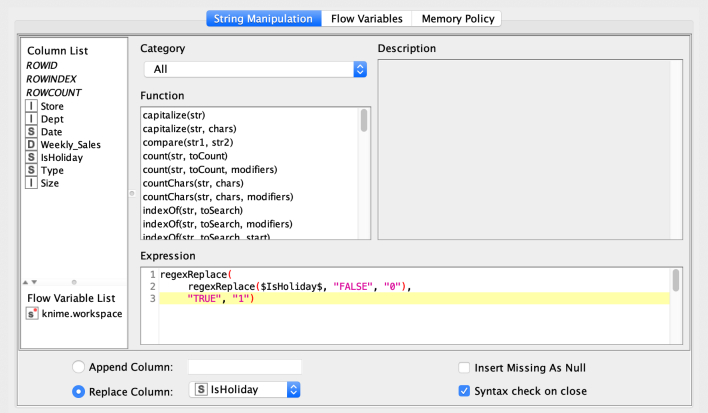


Figure 4: Nodo String Manipulation

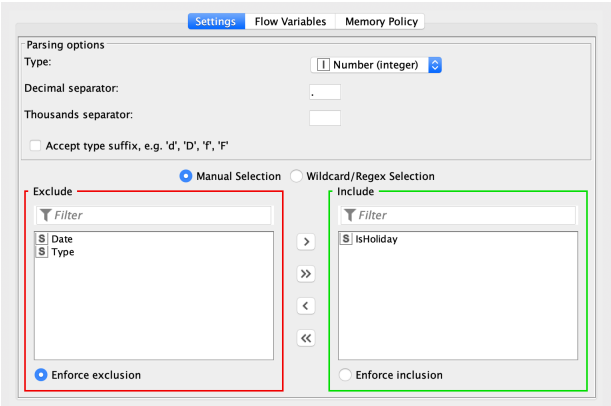


Figure 5: Nodo String To Number

Para extrair os campos ano e mês do atributo data mudei o seu tipo para *Date&Time*.

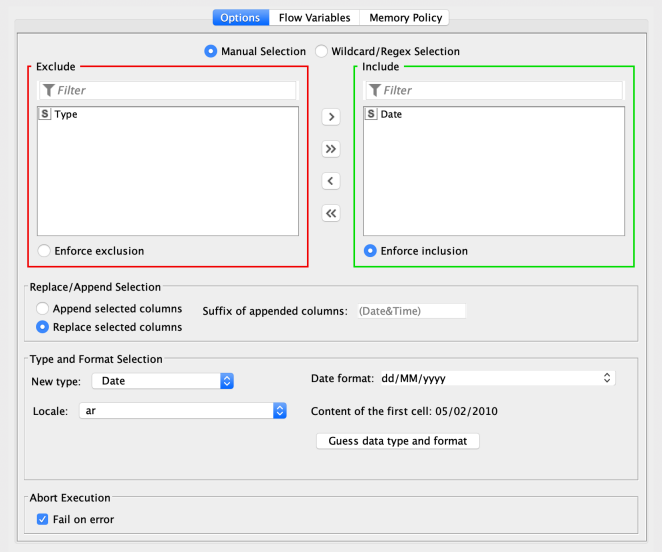


Figure 6: Nodo String To Date&Time

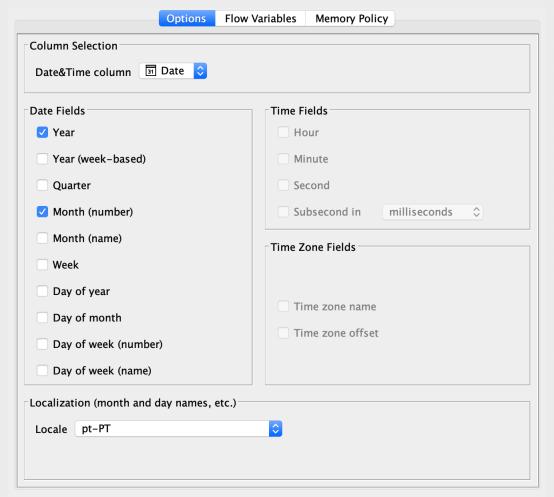


Figure 7: Nodo Extract Date&Time Fields

Agrupando os dados pelos atributos loja, tipo, tamanho, ano e mês determinei o somatório das vendas semanais por loja, e obtendo o valor máximo do atributo *IsHoliday* determinei a existência de feriados nesse mês.

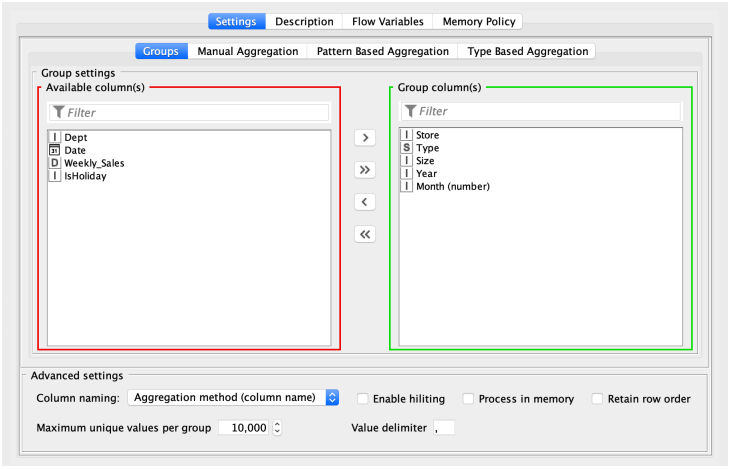


Figure 8: Nodo GroupBy

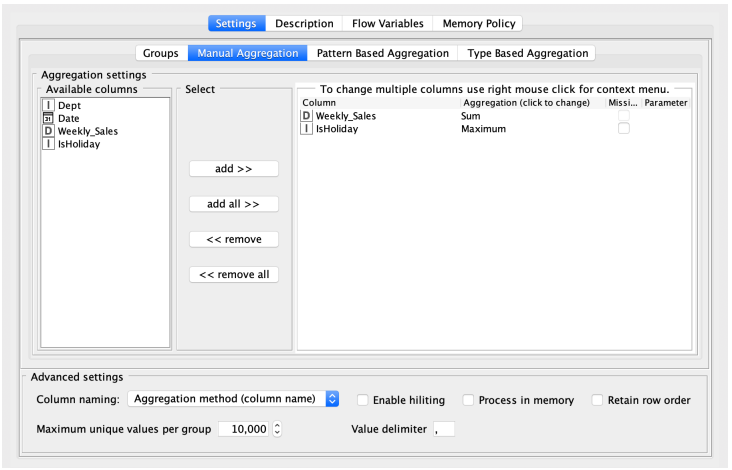


Figure 9: Nodo GroupBy

Através de uma transformação linear Min-Max entre 0 e 1 normalizei o somatório das vendas semanais.

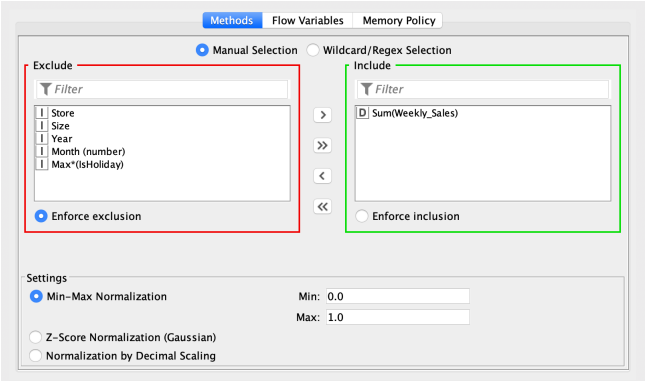


Figure 10: Nodo Normalizer

Criei 4 bins de igual frequência sobre o valor normalizado, substituindo a respetiva coluna.

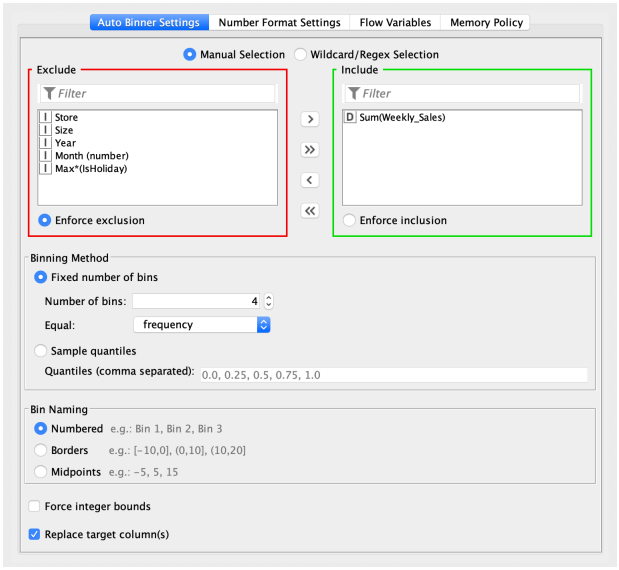


Figure 11: Nodo Auto-Binner

Por fim, renomeei os bins para o respetivo valor nominal.

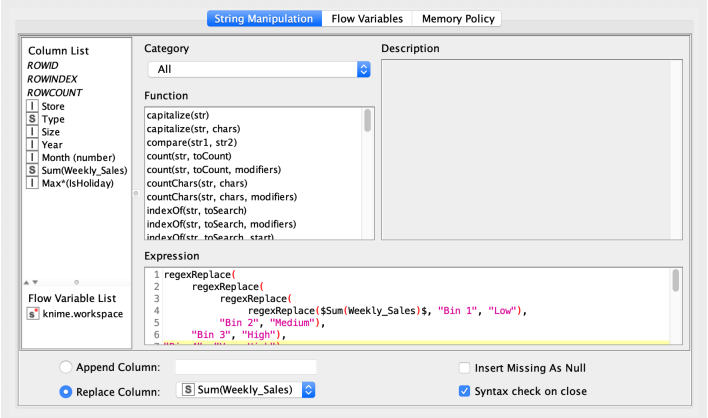


Figure 12: Nodo String Manipulation

O seguinte workflow representa todos os passos realizados para esta tarefa.

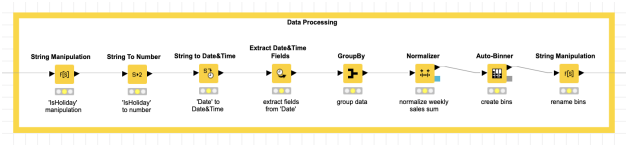


Figure 13: Processamento ds dados

3 Tarefa 3

Comecei por treinar uma árvore de decisão simples, carregando outro *dataset* para servir de conjunto de dados de teste.

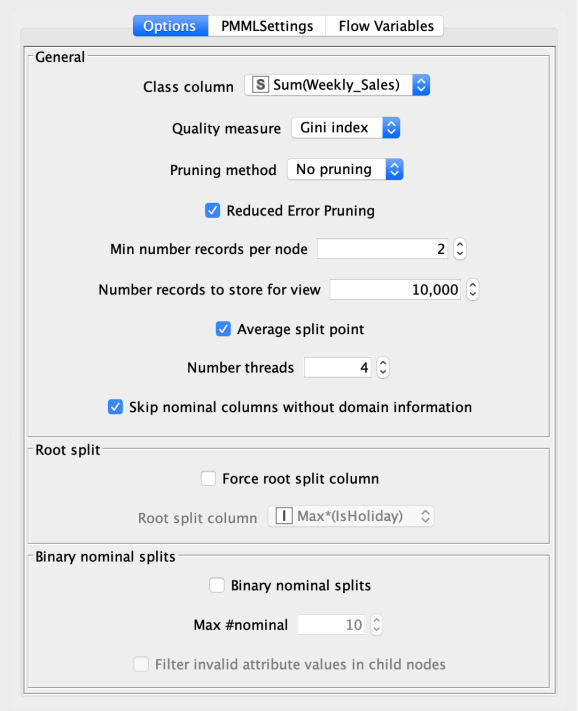


Figure 14: Nodo Decision Tree Learner

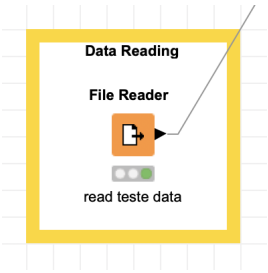


Figure 15: Leitura do dataset de teste

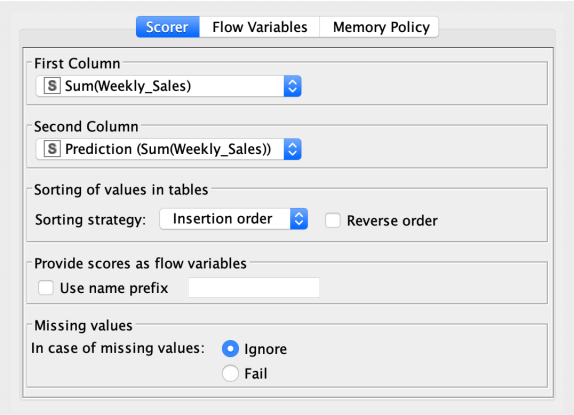


Figure 16: Nodo Scorer

Obtive a seguinte matriz de confusão resultante do nodo *Scorer*.

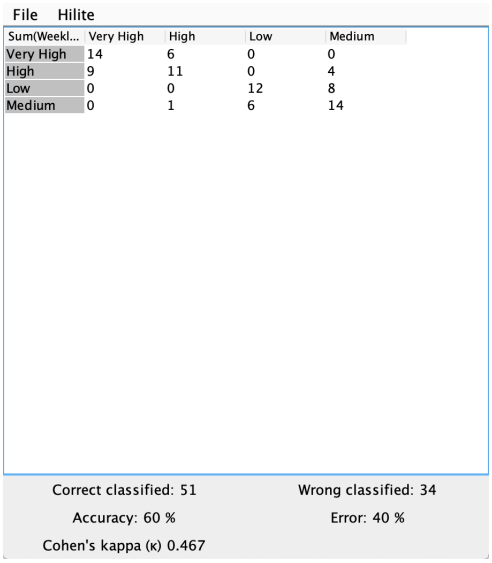


Figure 17: Matriz de confusão

O seguinte *workflow* representa todos os passos realizados para esta tarefa.

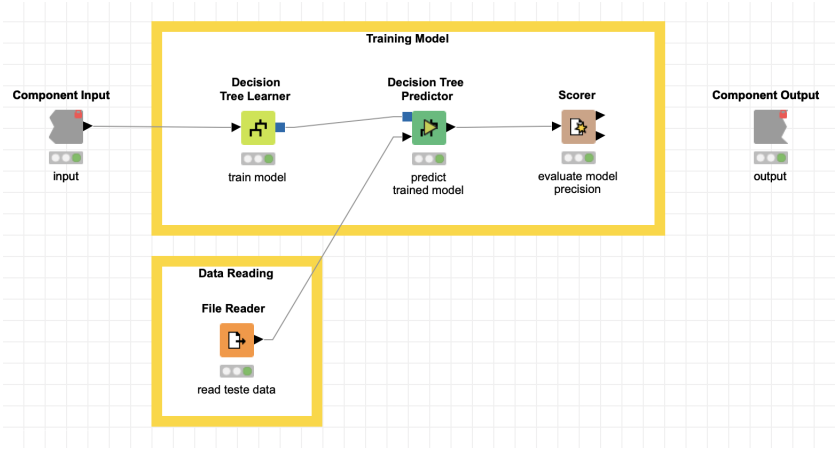


Figure 18: Workflow

4 Tarefa 4

O tuning do modelo da tarefa anterior foi realizado tendo em conta os parâmetros do número mínimo de registos por nodo, a medida de qualidade e o método de pruning.

Começando pelo primeiro parâmetro, criei uma variável que itera entre os valores 2 e 10 (de 1 em 1) e associei-a ao respetivo parâmetro no nodo *Decision Tree Learner*

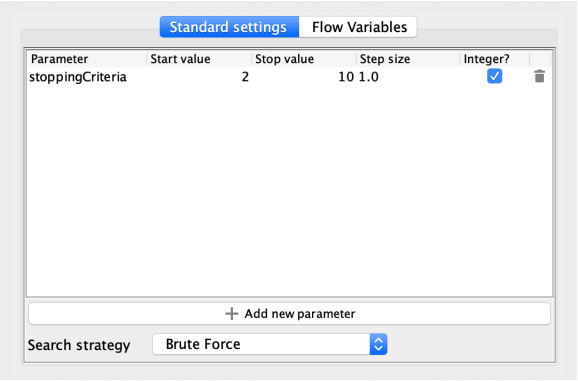


Figure 19: Nodo Parameter Optimization Loop Start

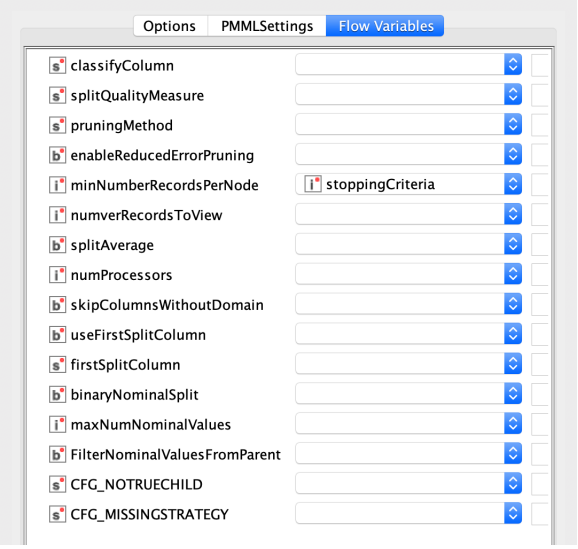


Figure 20: Nodo Decision Tree Learner

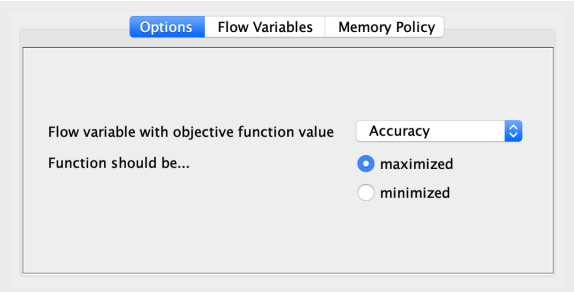


Figure 21: Nodo Parameter Optimization Loop End

O seguinte *workflow* representa todos os passos realizados.

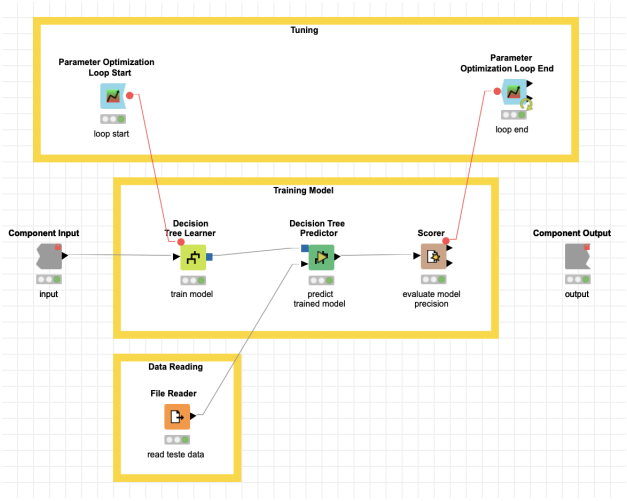


Figure 22: Tuning do modelo

Para o segundo parâmetro, criei duas colunas cujos valores são as opções para a medida de qualidade e as opções para o método de pruning respetivamente e associei-as aos respetivos parâmetros no nodo *Decision Tree Learner*

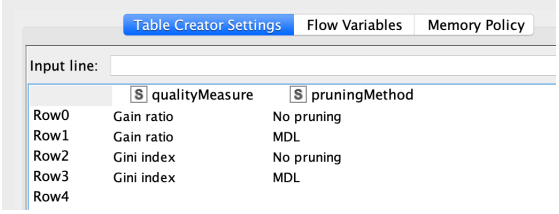


Figure 23: Nodo Table Creator

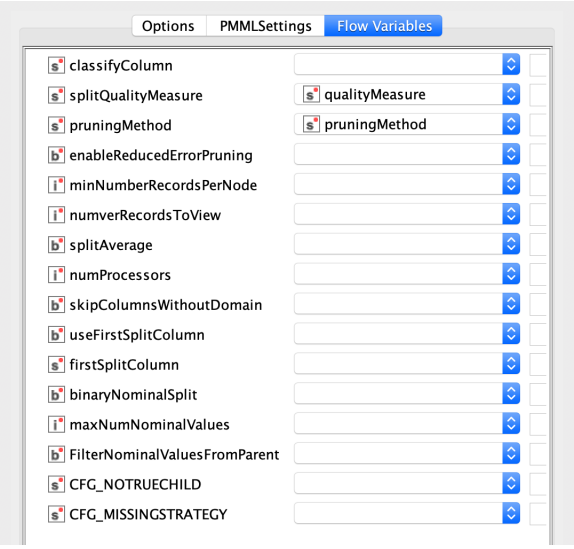


Figure 24: Nodo Decision Tree Learner

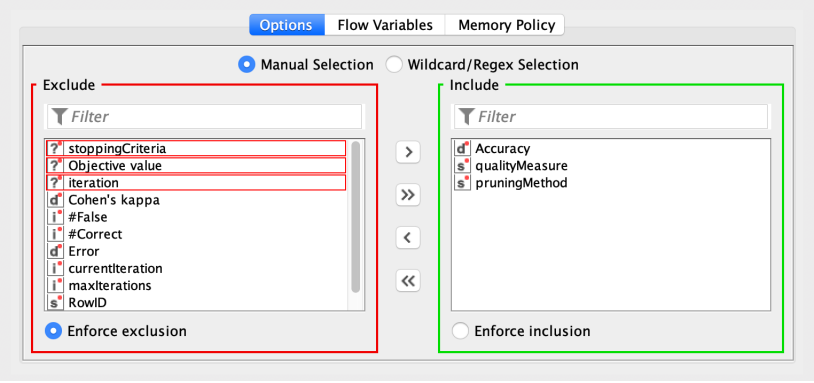


Figure 25: Nodo Variable Loop End

O seguinte *workflow* representa todos os passos realizados.

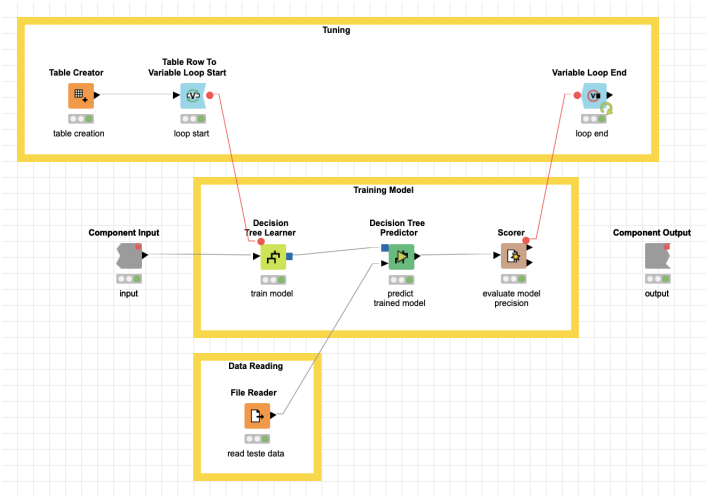


Figure 26: Workflow

Combinado os tunings anteriores desenvolvi o seguinte *workflow*.

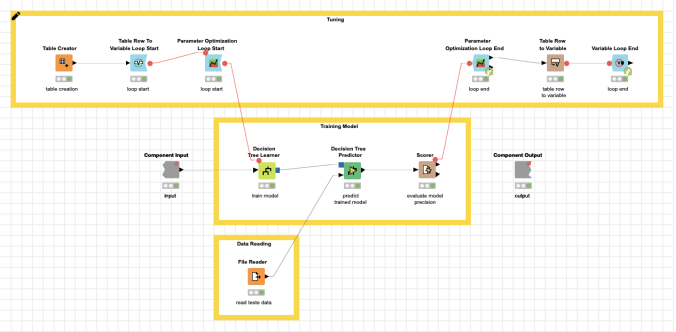


Figure 27: Workflow

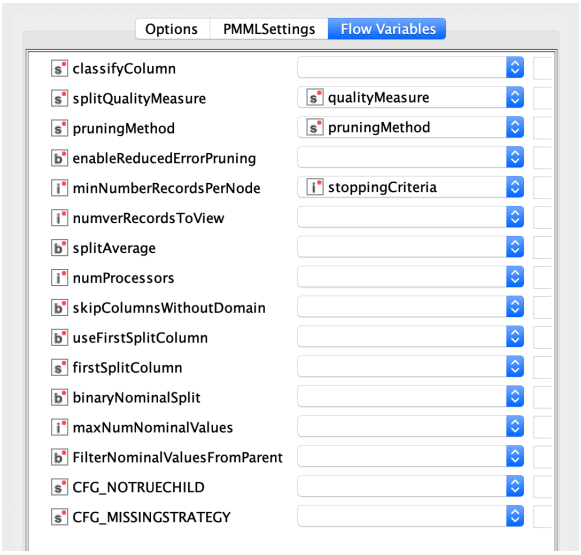
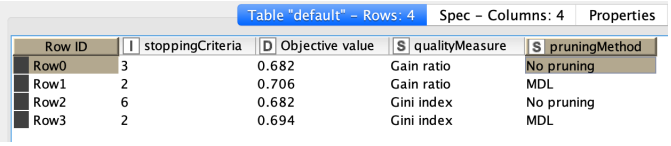


Figure 28: Nodo Decision Tree Learner



Row ID	stoppingCriteria	Objective value	qualityMeasure	pruningMethod
Row0	3	0.682	Gain ratio	No pruning
Row1	2	0.706	Gain ratio	MDL
Row2	6	0.682	Gini index	No pruning
Row3	2	0.694	Gini index	MDL

Figure 29: Tabela resultante

Observando a tabela resultante do pruning do modelo, podemos concluir que a melhor combinação de parâmetros para maximizar a *accuracy* é 2 registos mínimos por nodo, a medida de qualidade *Gain ratio* e método de pruning *MDL*. Apesar de não haver grandes discrespâncias entre os valores, as diferenças são significativas, especialmente quando o modelo é exposto a um dataset com um maior número de instâncias.

5 Tarefa 5

Para uma *Random Forest* decidi fazer o *tuning* dos parâmetros *Split Criterion* e *Number of models*.

Para o primeiro parâmetro criei uma coluna com os valores *Information Gain, Information Gain Ration* e *Gini Index*.

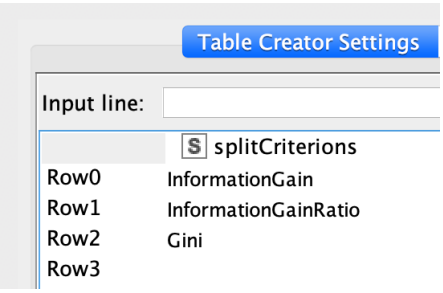


Figure 30: Nodo Table Creator

Relativamente ao segundo parâmetro criei uma variável que itera entre os valores 100 e 200 (de 1 em 1).

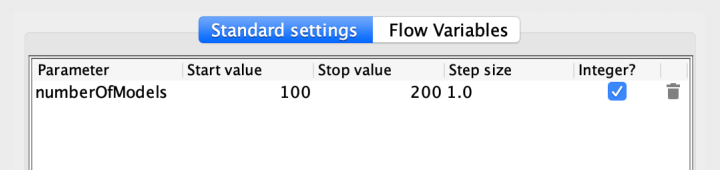


Figure 31: Nodo Parameter Optimization Loop Start

No nodo *Random Forest Learner*, na janela das *Flow Variables*, associei as variáveis criadas anteriormente aos parâmetros do modelo.

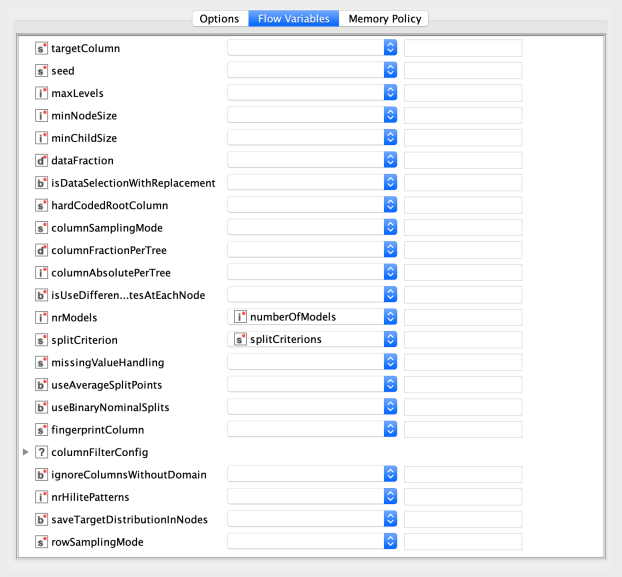


Figure 32: Nodo Random Forest Learner

Por fim, obtive a seguinte combinação de parâmetros.

Table "default" - Rows: 3 | Spec - Columns: 3

Row ID	numberOfModels	Objective value	splitCriterions
Row0	101	0.706	InformationGain
Row1	101	0.706	InformationGainRatio
Row2	101	0.694	Gini

Figure 33: Tabela resultante

O seguinte *workflow* representa todos os passos realizados para esta tarefa.

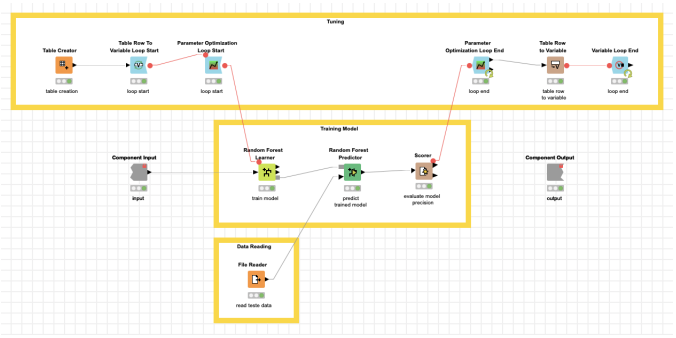


Figure 34: Workflow

6 Tarefa 6

Comparando as performances dos modelos treinados nas duas tarefas anteriores, posso concluir que, apesar da melhor *accuracy* obtida em ambos ser semelhante, em geral, o modelo *Random Forest* apresenta melhores resultados.