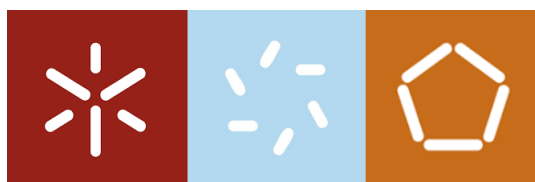


Universidade do Minho

Mestrado em Matemática e Computação

Sistemas Baseados em Similaridade - TP5



Simão Pedro Batista Caridade Quintela - PG52257

Outubro
2023

Universidade do Minho
Mestrado em Matemática e Computação

Relatório

Relatório realizado no âmbito do TP5 da UC Sistemas Baseados em Similaridade do Mestrado em Matemática e Computação.

Novembro
2023

Conteúdo

1	Contextualização	1
2	Tarefas	2
2.1	Tarefa 1 e 2	2
2.2	Tarefa 3	7
2.3	Tarefa 4	10
2.4	Tarefa 5	17
2.5	Tarefa 6	19
3	Conclusão	20

1 Contextualização

Para a realização do TP4 foi-nos proposto a aplicação de métodos de **clustering** sobre um dataset de **vinhos**. Para isso foi-nos fornecido um dataset de treino e um de teste.

Para além de aplicar técnicas de clustering, este trabalho tem também como objetivo a aplicação de técnicas de exploração e tratamento de dados, bem como a parametrização do workflow desenvolvido.

2 Tarefas

2.1 Tarefa 1 e 2

Enunciado: Carregar, no Knime, os dois primeiros datasets, juntá-los e explorar os dados utilizando vistas gráficas que permitam perceber a análise efetuada.

a) Fazer label encoding à feature isHoliday (1 deve corresponder ao valor True)

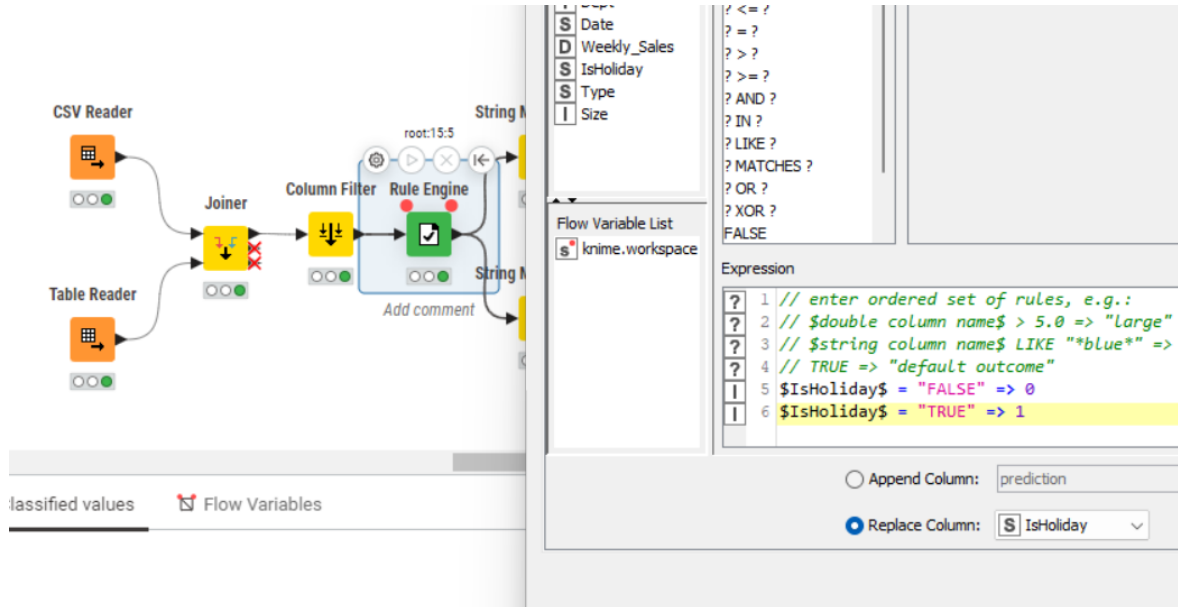


Figura 1: Label Encoding

Para a realização de label encoding à feature **isHoliday**, utilizei o nodo **Rule Engine** com a configuração mostrada na figura. b) Adicionar, a cada registo, as features ano e mês

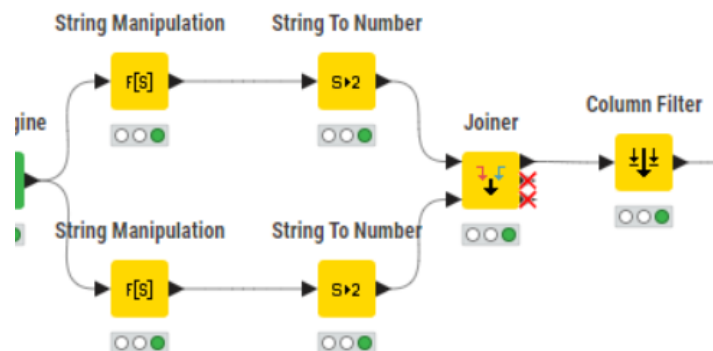


Figura 2: Workflow de manipulação da data

Para adicionar as features ano e mês utilizei os nodos **String Manipulation** e **String to Number**. Após obter a data através de manipulação das strings, utilizei o nodo **Joiner** e **Column Filter** para juntar os dados e remover as colunas excedentárias.

c) Agrupar os registos por loja, tipo, tamanho, ano e mês, agregando de forma a obter o somatório das vendas semanais de cada loja e a indicação da existência de feriados nesse mês

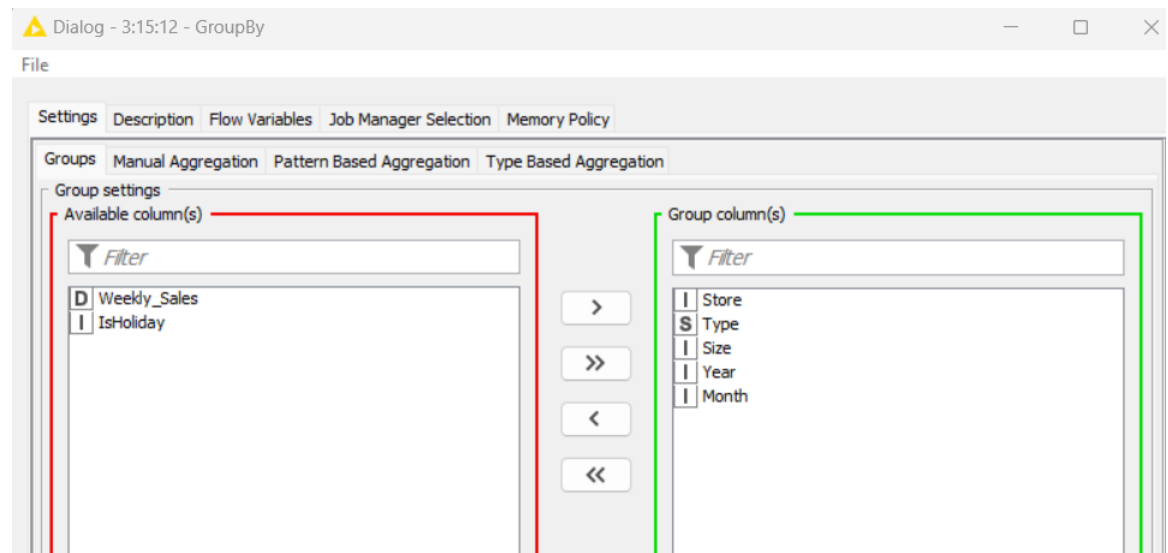


Figura 3: Configuração do nodo GroupBy

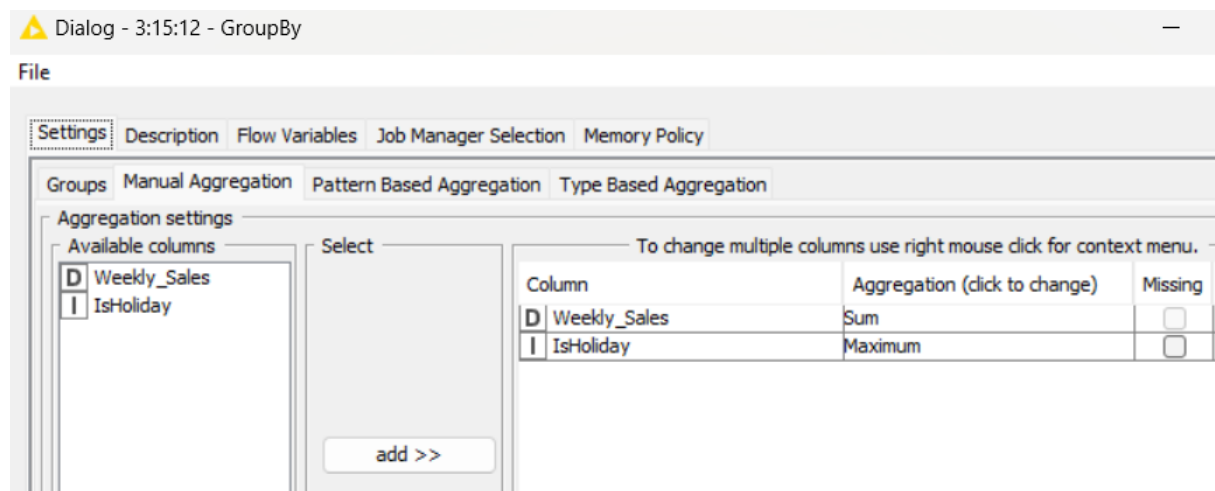



Figura 4: Configuração do nodo GroupBy

► 1: Group table  Flow Variables

Rows: 476 | Columns: 7 Table Statistics

#	Row...	Store Number (integer)	Type String	Size Number (integer)	Year Number (integer)	Month Number (integer)	Sum(Weekly_S... Number (double)	Max*(IsHoliday) Number (integer)
1	Row0	1	A	151315	2010	2	6,307,344.1	1
2	Row1	1	A	151315	2010	3	5,871,293.98	0
3	Row2	1	A	151315	2010	4	7,422,801.92	0
4	Row3	1	A	151315	2010	5	5,929,938.64	0
5	Row4	1	A	151315	2010	6	6,084,081.46	0
6	Row5	1	A	151315	2010	7	7,244,483.04	0
7	Row6	1	A	151315	2010	8	6,075,952.95	0
8	Row7	1	A	151315	2010	9	5,829,793.92	1

Figura 5: Output de agregação

Para agrupar os registos da forma pedida, utilizei o nodo **GroupBy** e os dados resultantes estão visíveis na figura 7.

d) Normalizar o somatório das vendas semanais utilizando a transformação linear Min-Max entre 0 e 1

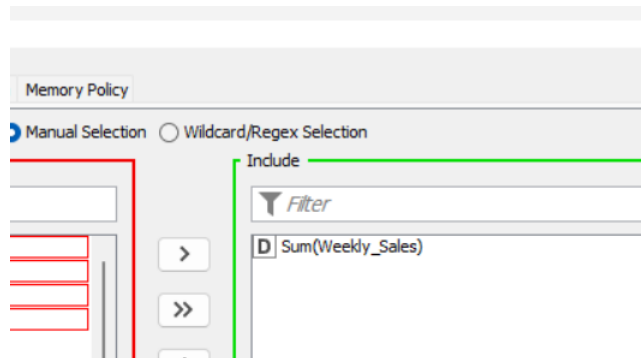


Figura 6: Configuração do nodo Normalizer

Para normalizar os dados foi utilizado o nodo **Normalizer**, e apenas incluí o atributo **Weekly Sales** na normalização, como pedido.

e) Criar 4 bins de igual frequência sobre o valor normalizado no passo anterior (ligando a opção `replace target column(s)`)

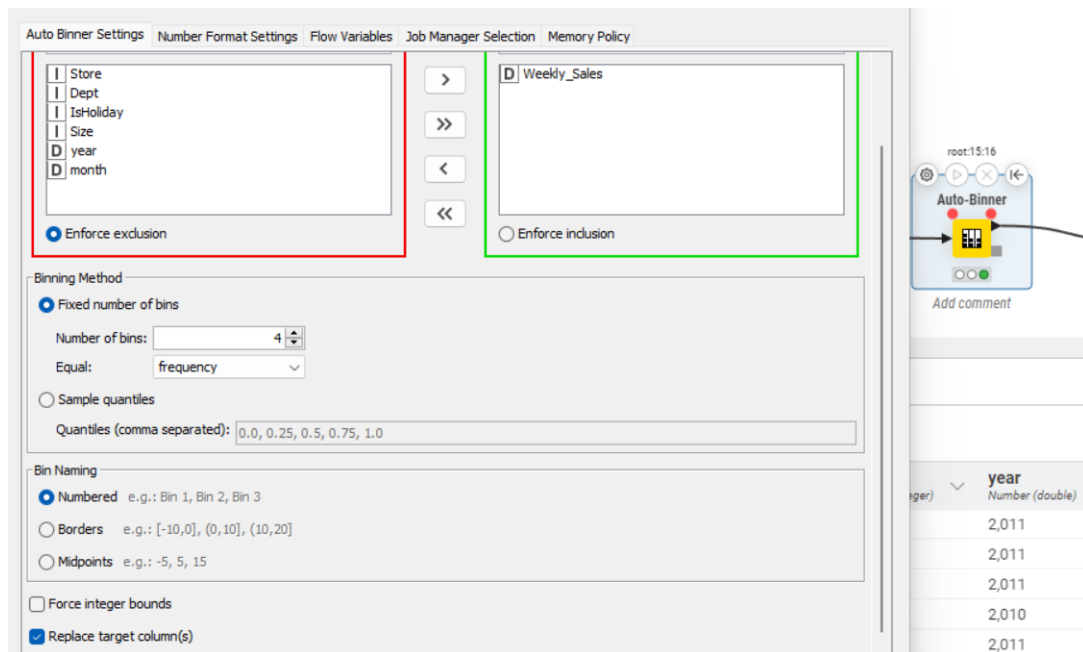


Figura 7: Configuração do nodo Auto-Binner

Para criar os 4 Bins utilizei o nodo **Auto-Binner**, criando 4 bins de igual frequência e substituindo na coluna destino.

f) Renomear cada bin de forma a que o primeiro corresponda a Low, o segundo a Medium, o terceiro a High e o quarto a Very High.

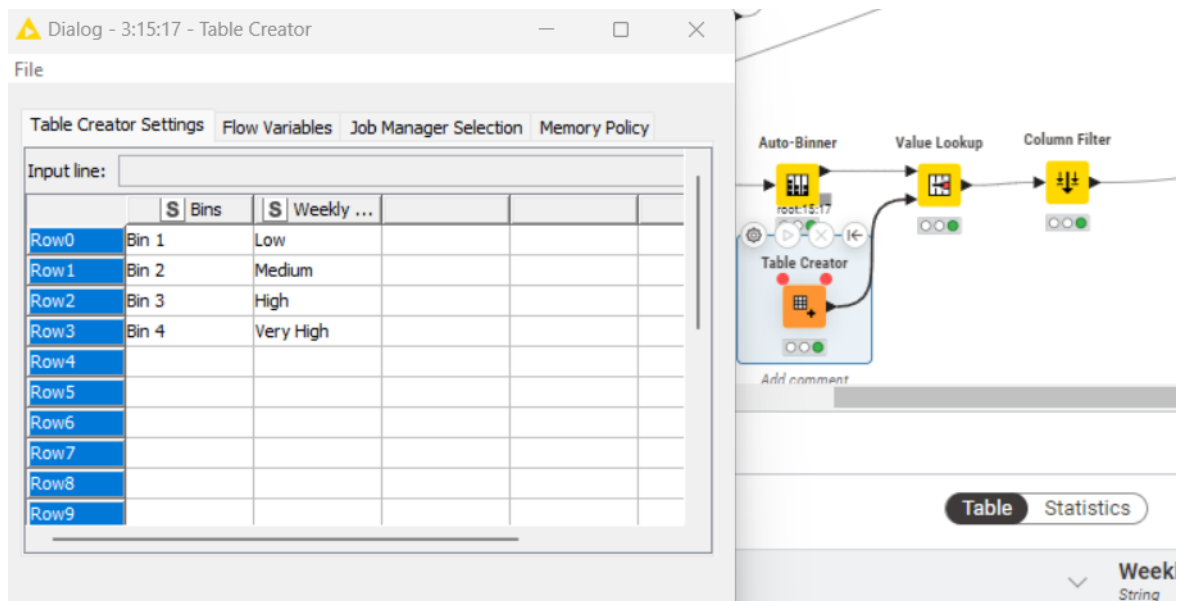


Figura 8: Categorização dos dados

Para realizar o proposto nesta alínea utilizei os nodos Table Creator e Value Lookup. Com o table creator criei uma tabela com 2 colunas e 4 linhas, nas quais as linhas da coluna Bins são os nomes dos Bins criados na alínea anterior, e os valores da coluna Weekly Sales são os valores Low, Medium, High e Very High. Posto isto, utilizei o nodo Value Lookup para dar match às linhas com o nome do bin correspondente e quando o match acontece, o valor é transportado para o valor correspondente ao Bin na coluna Weekly Sales.

2.2 Tarefa 3

a) Treinar uma árvore de decisão.

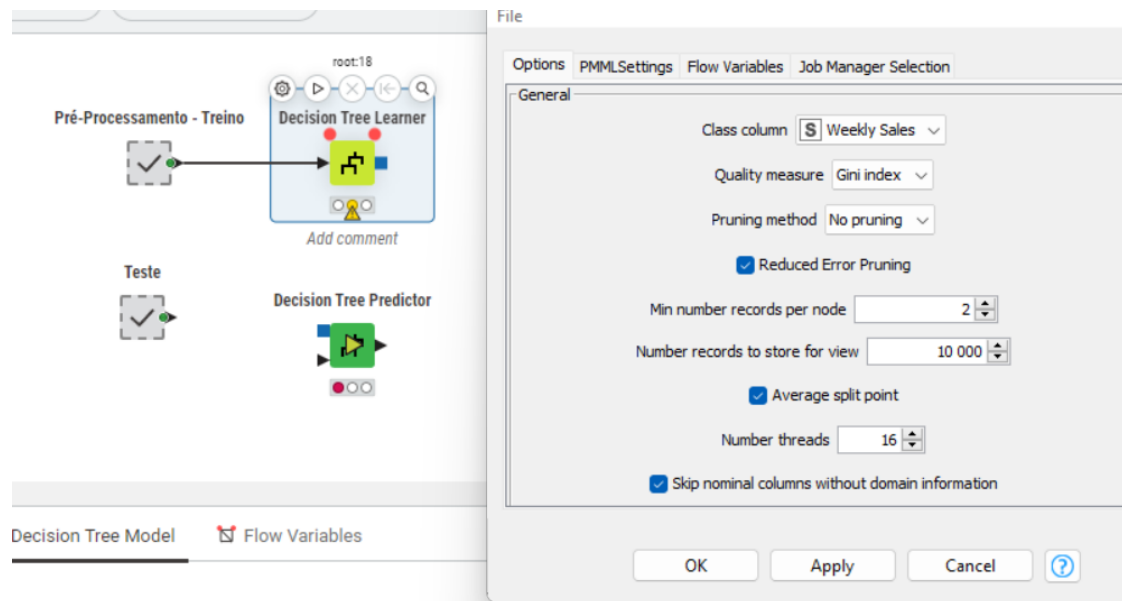


Figura 9: Treino de um nodo Decision Tree

Para realizar o treino numa Decision Tree, selecionei a coluna Weekly Sales e usei Gini Index como Gain Ratio.

b) Carregar o dataset de teste e prever o valor de vendas de cada mês para cada uma das 17 lojas.

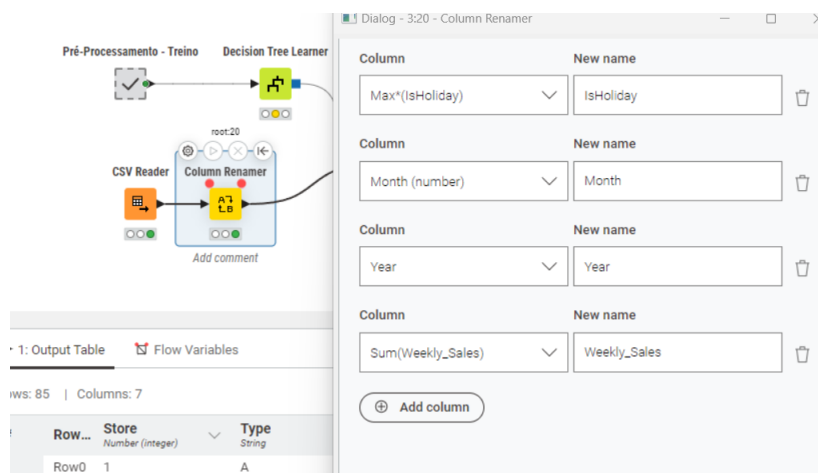


Figura 10: Alteração do nome de colunas

Inicialmente alterei o nome das colunas do dataset de teste para corresponder ao nome das colunas do dataset de treino.

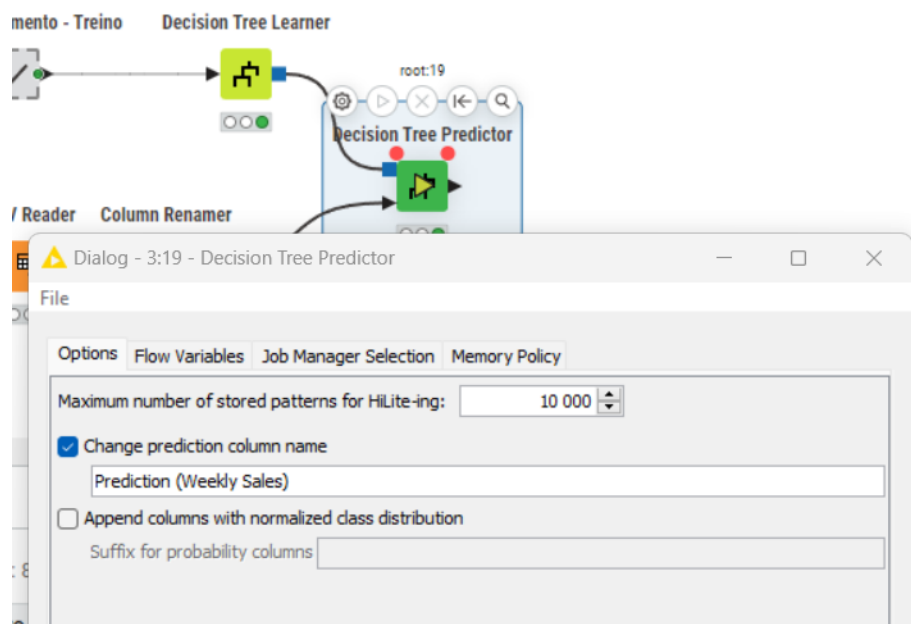
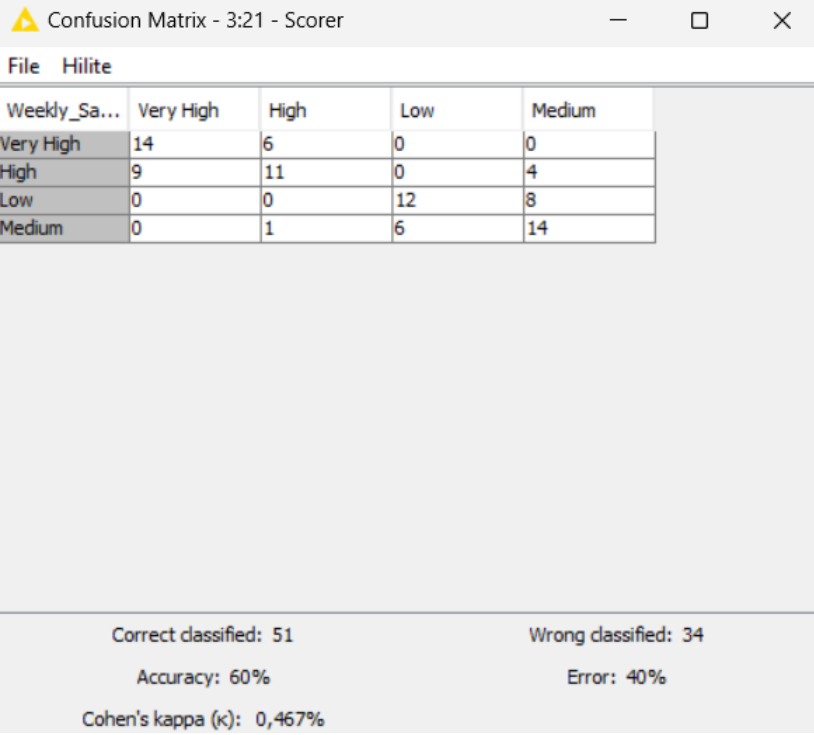


Figura 11: Previsão das Weekly Sales

Após a alteração do nome das colunas utilizei o Decision Tree Predictor para prever as Weekly Sales.

c) Mostrar, graficamente, uma tabela com a matriz de confusão do modelo



Weekly_Sa...	Very High	High	Low	Medium
Very High	14	6	0	0
High	9	11	0	4
Low	0	0	12	8
Medium	0	1	6	14

Correct classified: 51 Wrong classified: 34
Accuracy: 60% Error: 40%
Cohen's kappa (κ): 0,467%

Figura 12: Matriz de Confusão

Na figura podemos ver a matriz de confusão do modelo, cuja accuracy foi de 60%.

2.3 Tarefa 4

Enunciado: Fazer o tuning do modelo criado no passo anterior usando:

a) Todos os valores, entre 2 e 10, para o número mínimo de registos por nodo;

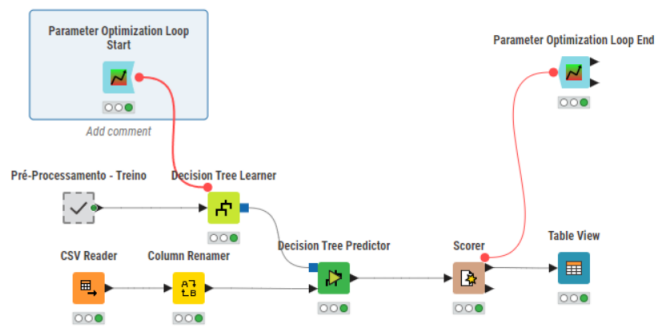


Figura 13: Workflow após realização de Tuning

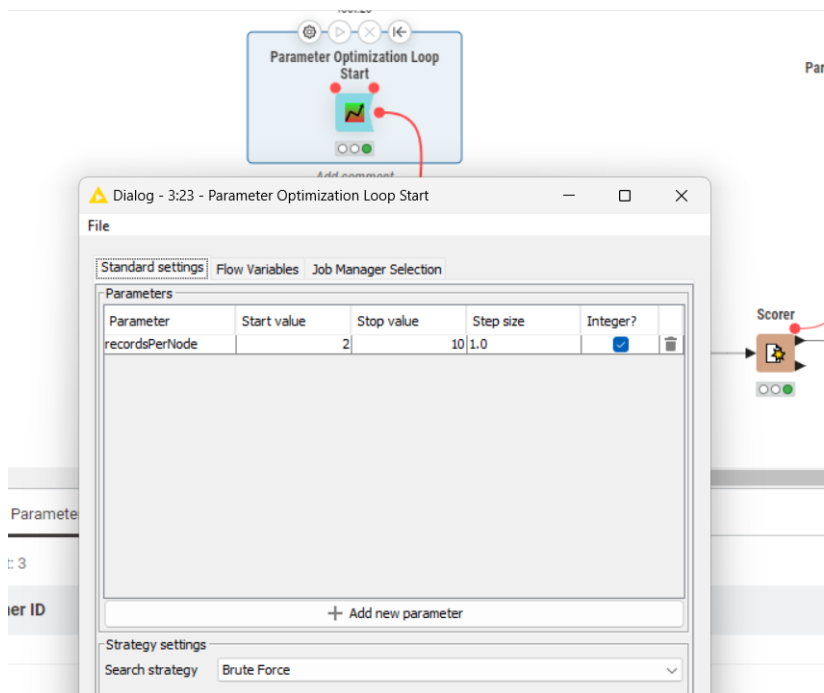


Figura 14: Parameter Optimization Loop Start

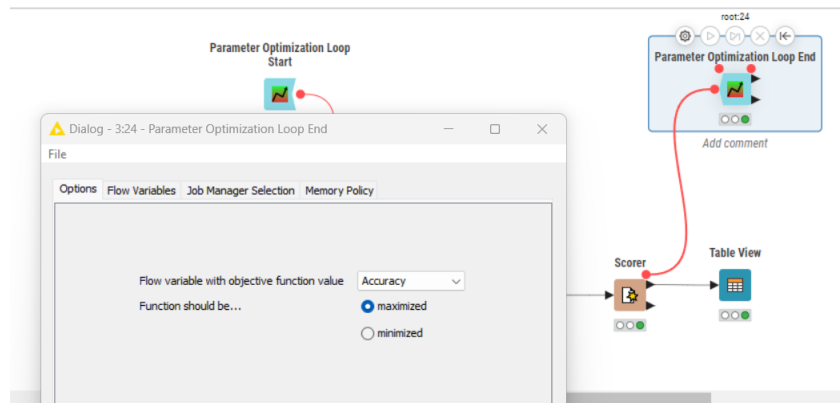


Figura 15: Parameter Optimization Loop End

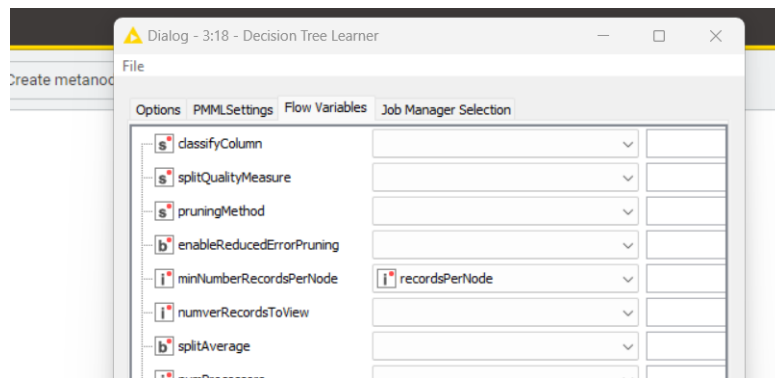



Figura 16: Aplicação da variável de fluxo

Para a realização de **Tuning**, inicialmente, adicionei ao workflow os nodos **Parameter Optimization Loop** que criam um loop no workflow, em que a variável criada **recordsPerNode** vai, a cada iteração, sendo incrementada uma unidade. Isto é aplicado ao parâmetro **minRecordsPerNode** na Decision Tree.

▶ 1: Best parameters
▶ 2: All parameters
 Flow Variables

Rows: 9 | Columns: 2

Table
Statistics

#	Row...	minRecords <i>Number (integer)</i>	✓ Objective value <i>Number (double)</i>
1	Row0	2	0.6
2	Row1	3	0.6
3	Row2	4	0.6
4	Row3	5	0.6
5	Row4	6	0.6
6	Row5	7	0.6
7	Row6	8	0.6
8	Row7	9	0.6
9	Row8	10	0.6

Figura 17: Resultados da realização de Tuning

Os resultados do Tuning são os mostrados na figura, não se tendo notado nenhum efeito na accuracy do modelo.

b) Todas as possibilidades para a medida de qualidade

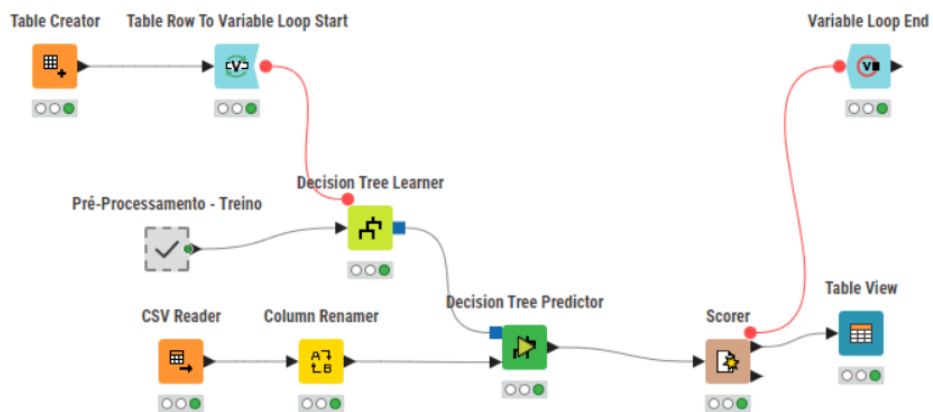


Figura 18: Workflow para a medida de qualidade

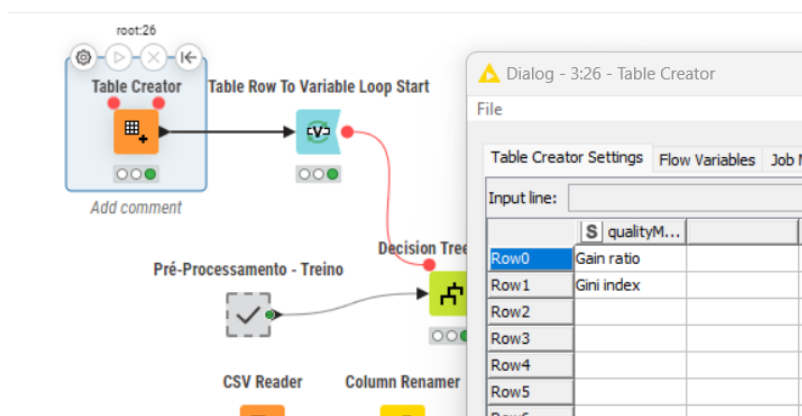


Figura 19: Tabela com os registos de qualidade

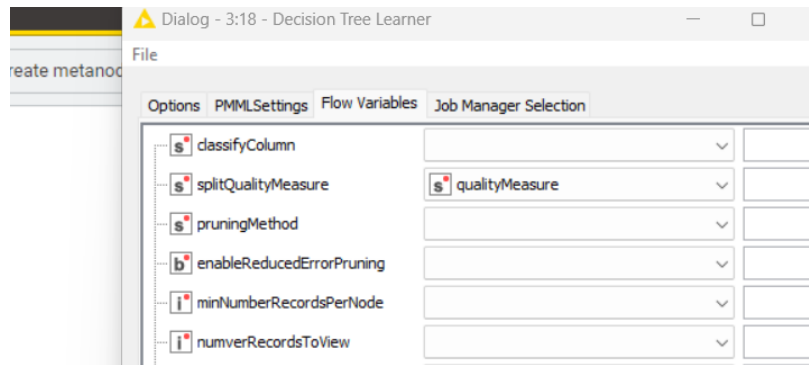


Figura 20: Aplicação da variável criada

Rows: 2 Columns: 3				Table	Statistics
#	Row...	Accuracy <i>Number (double)</i>	quality <i>String</i>		
1	Row0	0.647	Gain Ratio		
2	Row1	0.647	Gini Index		

Figura 21: Tabela com os registos de qualidade

Para calcular qual o melhor registo de qualidade criei uma tabela com os respetivos valores de qualidade para uma Decision Tree. Após isso utilizei o nodo **Table Row to Variable Loop** e, de forma idêntica à última alínea, associei a variável criada ao parâmetro associado à qualidade na Decision Tree.

Conclui-se que a accuracy aumenta ao utilizar uma quality measure, no entanto, o aumento de accuracy é independente do método que se usa, neste caso.

c) Todas as possibilidades para o método de pruning

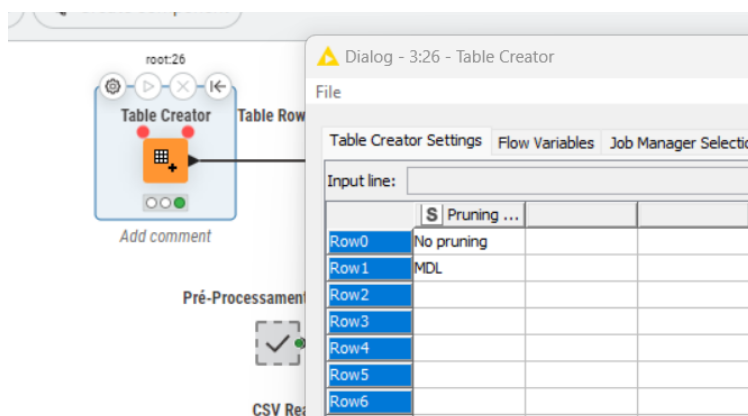



Figura 22: Criação da flow variable

► 1: Data table of flow variables  Flow Variables

Rows: 2 | Columns: 3 Table Statistics

#	Row...	Accuracy <i>Number (double)</i>	Pruning <i>String</i>
1	Row0	0.6	No pruning
2	Row1	0.694	MDL

Figura 23: Resultados obtidos

Utilizando os mesmos nodos da última alínea, alterei apenas o nome da variável de fluxo utilizada e apliquei-a no nodo **Decision Tree**.

Os resultados obtidos dizem que utilizando o método de pruning MDL a accuracy é bastante superior, neste caso 69%.

d) Fazer o tuning dos parâmetros anteriores num único workflow. Guardar e analisar todos os resultados obtidos para cada combinação de hiper-parâmetros. Qual a combinação que oferece melhor performance? Existem grandes discrepâncias?

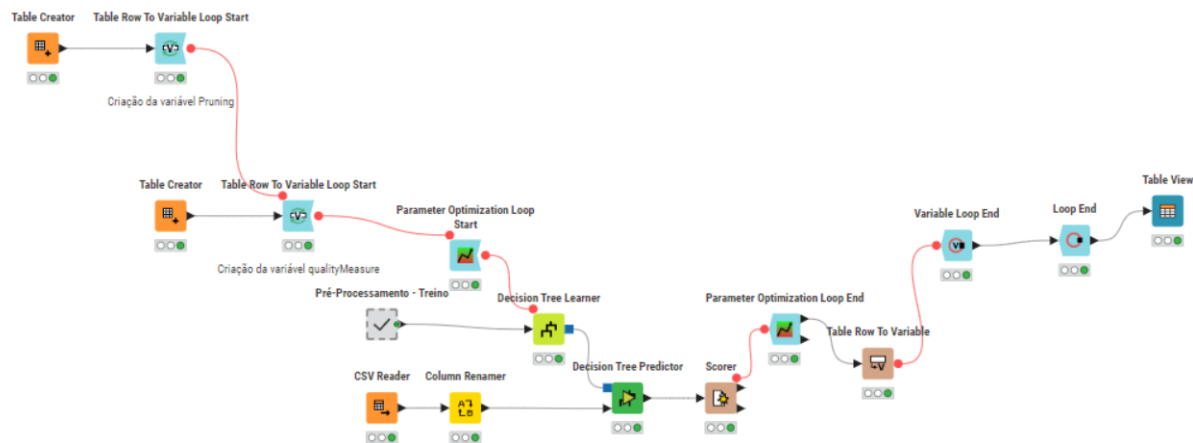


Figura 24: Workflow com todas as combinações

► 1: Collected results 📄 Flow Variables

Rows: 4 | Columns: 3

Table Statistics

#	Row...	Objective value Number (double)	Pruning String
1	Row...	0.682	No pruning
2	Row...	0.706	MDL
3	Row...	0.682	No pruning
4	Row...	0.706	MDL

Figura 25: Resultados obtidos

Para realizar todas as combinações possíveis utilizei loops aninhados para garantir que cobria todos os parâmetros. No fim, verificou-se algum aumento mas não foi substancial relativamente a resultados anteriores.

2.4 Tarefa 5

Enunciado: Treinar e fazer o tuning de uma Random Forest. Guardar e analisar todos os resultados obtidos para cada combinação de hiper-parâmetros;

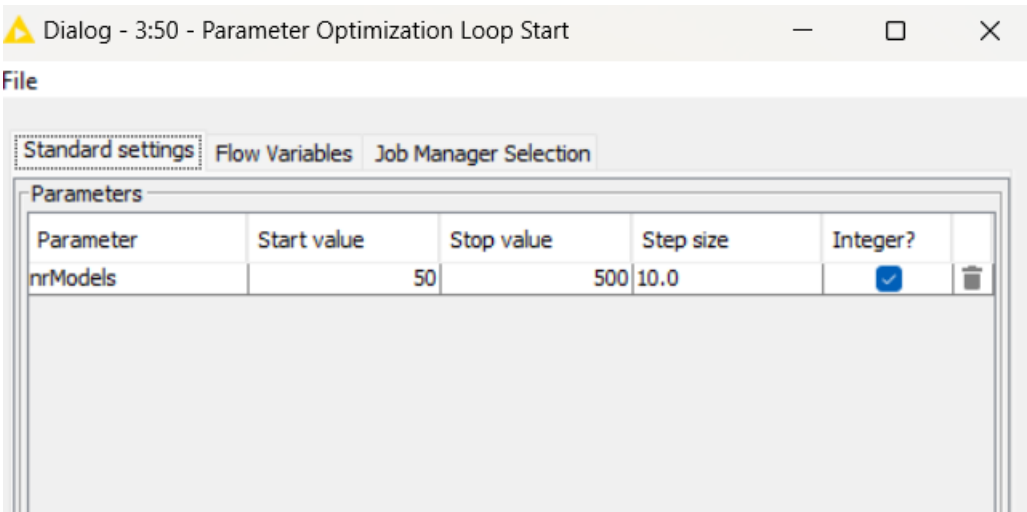


Figura 26: Variáveis de fluxo criadas

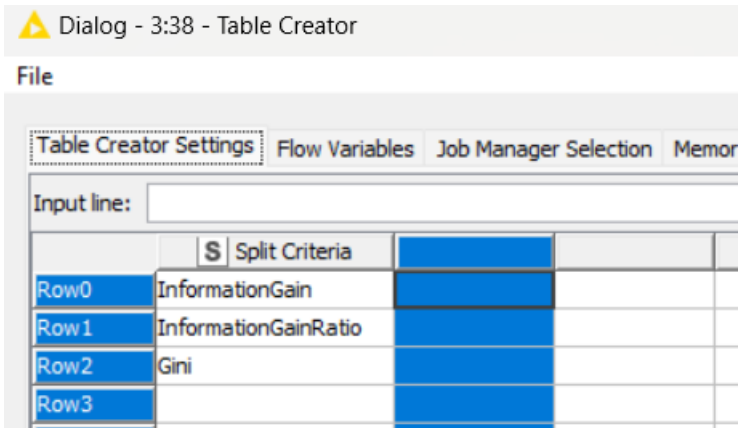


Figura 27: Variáveis de fluxo criadas

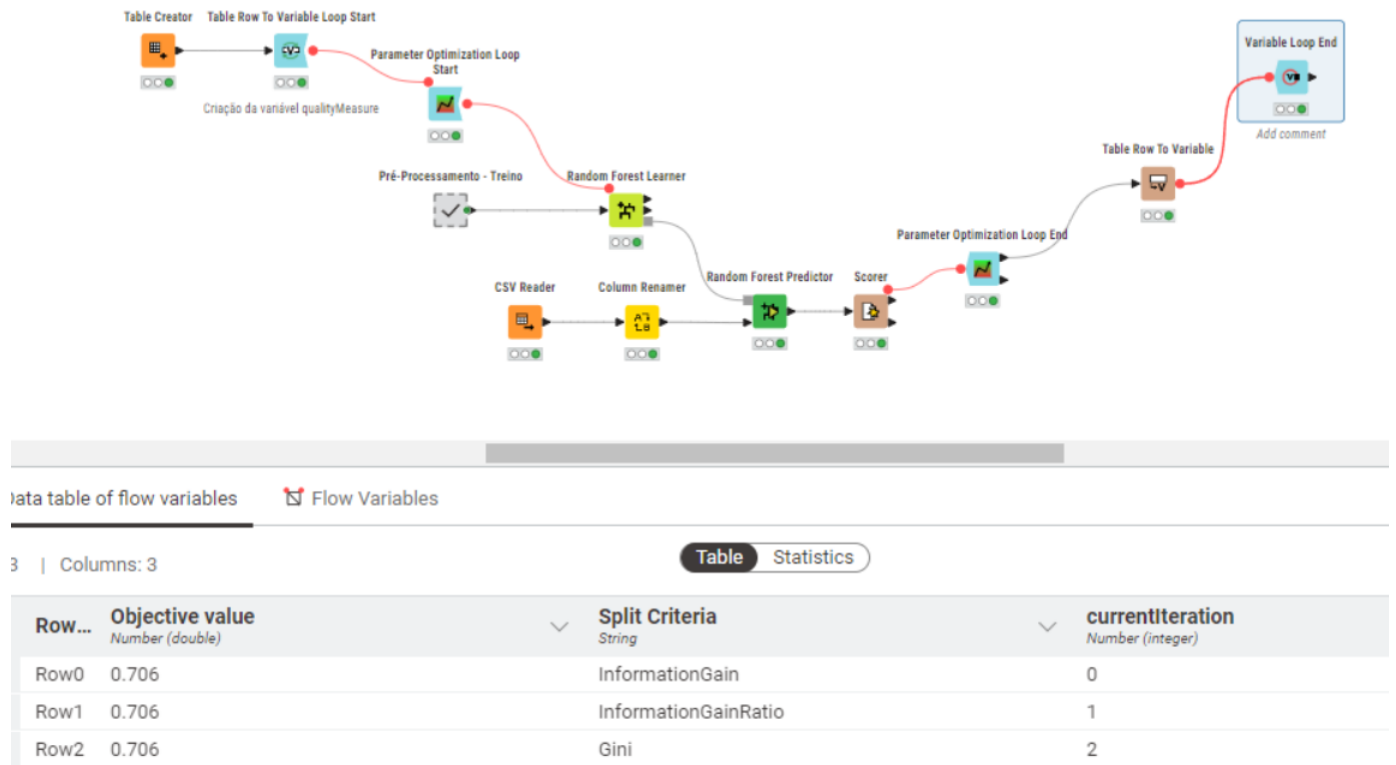


Figura 28: Resultados obtidos após realização de tuning

O workflow utilizado para realizar tuning foi bastante semelhante ao workflow usado anteriormente. As variáveis a que realizei tuning foram **Split Criteria** e **nrModels**.

O resultado final foi bastante semelhante aos resultados anteriores.

2.5 Tarefa 6

Enunciado: Analisar e comparar as performances dos modelos treinados em T4 e T5. Que conclusões se podem tirar?

► 1: Collected results 📄 Flow Variables

Rows: 4 | Columns: 3

Table Statistics

#	Row...	Objective value <i>Number (double)</i>	Pruning <i>String</i>
1	Row...	0.682	No pruning
2	Row...	0.706	MDL
3	Row...	0.682	No pruning
4	Row...	0.706	MDL

Figura 29: Resultados da Decision Tree

► 1: Data table of flow variables 📄 Flow Variables

Rows: 3 | Columns: 3

Table Statistics

#	Row...	Objective value <i>Number (double)</i>	Split Criteria <i>String</i>
1	Row0	0.706	InformationGain
2	Row1	0.706	InformationGainRatio
3	Row2	0.706	Gini

Figura 30: Resultados da Random Forest

Comparando ambos os métodos vemos que ambos apresentam resultados bastante semelhantes em termos de accuracy, o que vai contra o esperado. Esperava-se ver um aumento de accuracy com a utilização do algoritmo Random Forest.

3 Conclusão

Para concluir, este trabalho foi importante porque foram aprofundados algoritmos de **Decision Tree** e **Random Forest**. Para além disso, também foi introduzido o conceito de tuning de modelo. É de enorme importância estarmos conscientes da existência destes procedimentos, visto que nos ajuda a escolher os melhores parâmetros possíveis para treinar o nosso modelo.