Universidade do Minho

Mestrado em Matemática e Computação

Sistemas Baseados em Similaridade

Ficha Prática Individual 4

Hugo Filipe de Sá Rocha (PG52250)

1 de novembro de 2023

Conteúdo

Capítulo 1

Introdução

Neste trabalho, o objetivo passa por aplicar métodos de *clustering* sobre um *dataset* de vinhos, o qual contém um ficheiro para aprendizagem e outro para teste. Além disso, aplicaram-se técnicas para exploração e tratamento de dados, assim como para parametrização do *workflow* densenvolvido.

Capítulo 2

Concepção das soluções

2.1 Tarefa 1

- **Carregar, no *Knime*, o *dataset* descarregado e explorar os dados.**

Como é tradicional, para leitura do *dataset* de treino e de teste utilizei o nodo ***CSV Reader***, visto ambos os ficheiros estarem no formato *csv*. Além disso, foram aplicados alguns nodos para exploração de dados, sendo eles: ***Statistics***, ***Data Explorer***, ***Box Plot***, ***CrossTab*** e ***Rank Correlation***.

- No nodo ***Statistics*** podemos, por exemplo, analisar o *dataset* no que toca a *missing values* ou a estatísticas sobre os atributos. A título de exemplo, nas imagens abaixo podemos ver que o *dataset* **não possui *missing values*** e que, na análise ao ***pH*** e à quantidade de **álcool** presente nos vinhos, o *pH* varia entre **2.74** e **4.01**, obtendo-se uma média de **3.306** de *pH*. Quanto ao álcool, os valores variam entre **8.4** e **14.9**, obtendo-se uma média de **10.41** de álcool.

Statistics

Rows: 2 | Columns: 5

Name	# Missing values	Minimum	Maximum	Mean
pH	0	2.74	4.01	3.306
alcohol	0	8.4	14.9	10.412

Figura 2.1: `Mínimo, máximo e média dos valores de pH e de álcool.`

Statistics

Rows: 12 | Columns: 2

Name ↓	# Missing values
fixed acidity	0
volatile acidity	0
citric acid	0
residual sugar	0
chlorides	0
free sulfur dioxide	0
total sulfur dioxide	0
density	0
pH	0
sulphates	0
alcohol	0
quality	0

Figura 2.2: `Ausência de` *`missing values`*.

- No nodo **Data Explorer**, assim como no nodo *Statistics*, podemos obter estatísticas sobre os atributos bem como o histograma para cada um deles. A título de exemplo, segue abaixo o histograma do atributo ***quality***, onde podemos observar que a maioria dos vinhos apresenta qualidade **5** ou **6**. De seguida, por ordem de frequência, aparecem os vinhos de qualidade **7**, **4**, **8** e, por fim, **3**.

Column	Exclude Column	No. missings	Unique values	All nominal values	Frequency Bar Chart
quality	☐	0	6	=5, =6, =7, =4, =8, =3	

Figura 2.3: `Histograma do atributo` *`quality`*.

- No nodo ***Box Plot***, o objetivo passou por procurar ***outliers*** nos dados. Na imagem abaixo, podemos observar dois *outliers* relativos ao atributo ***total sulfure dioxide***, onde optei por não os remover do *dataset*.

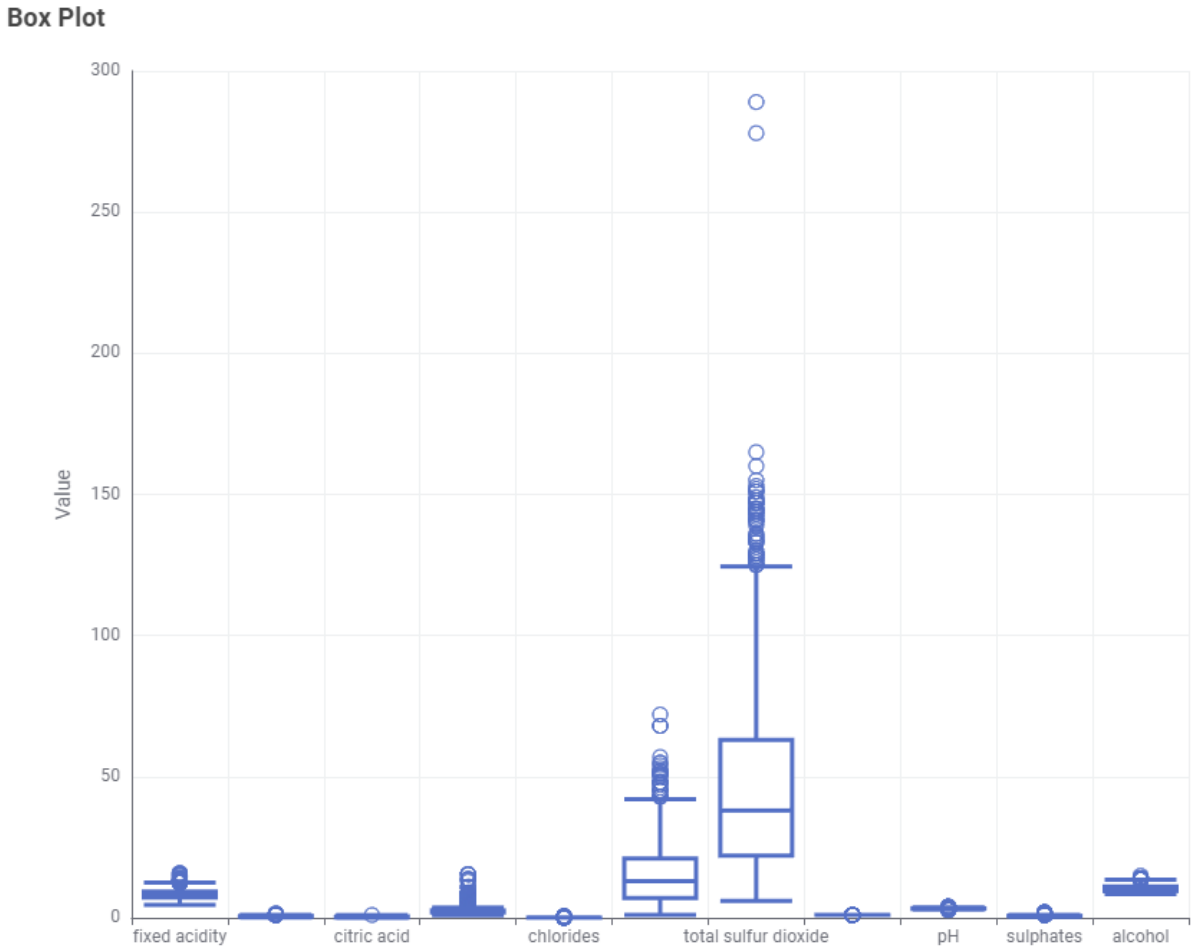

Figura 2.4: `Deteção de outliers.`

- No nodo ***CrossTab***, criei uma tabela ***fixed acidity*** `x` ***pH*** para tentar perceber a relação entre estes dois atributos e percebi que estão relacionados na medida em que, quanto maior for o valor de *fixed acidity*, a tendência é o valor de *pH* ser menor e, por outro lado, quanto menor for o valor de *fixed acidity*, a tendência é o valor de *pH* ser maior. Na imagem abaixo, podemos ver isso mesmo, com os maiores valores de *pH* (3.62 até 4.01), a corresponderem a valores de *fixed acidity* mais baixos.

3.62	3.63	3.66	3.67	3.68	3.69	3.7	3.71	3.72	3.74	3.75	3.78	3.85	3.9	4.01	Total
													1		1
												1			1
							1								1
						1		1	1	1				1	6
				1									1		4
				2			1				1				6
1															4
											1			1	3
															1
	2	1		1			2	2							13

Figura 2.5: *CrossTab fixed acidity x pH*.

- No nodo ***Rank Correlation***, procurou-se explorar a correlação entre todos os atributos (dois a dois) e confirmou-se a relação explorada na *CrossTab* mencionada anteriormente, com o atributo *pH* e *fixed acidity* a terem uma correlação negativa relativamente forte de **-0.6966**. Além disso, existem também outras correlações positivas relativamente fortes, por exemplo: ***fixed acidity*** com ***citric acid*** e ***fixed acidity*** com ***density***.

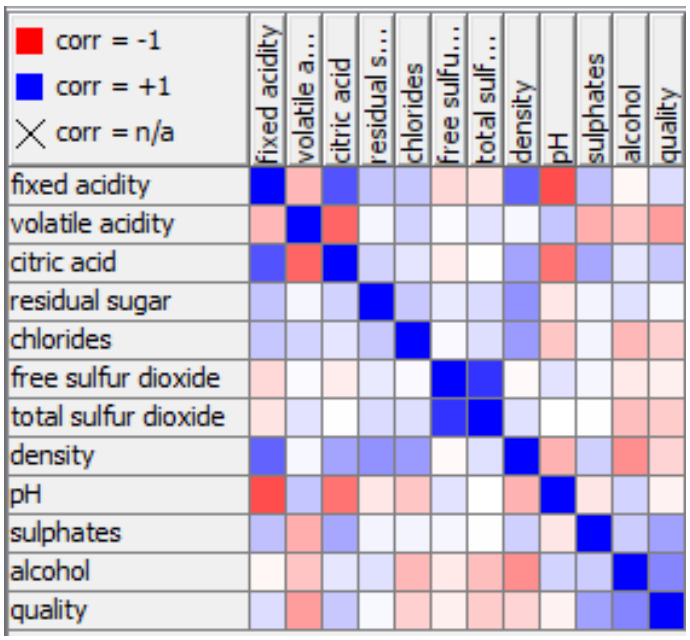

Figura 2.6: Correlação entre atributos

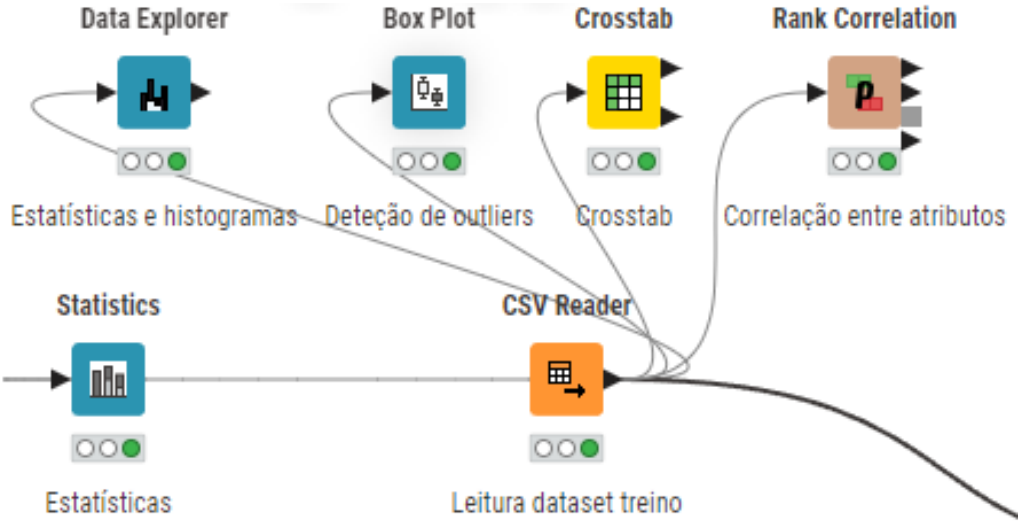

Figura 2.7: *Workflow* relativo à Tarefa 1

2.2 Tarefa 2 (tratamento de dados)

2.2.1 Alínea a)

- **Fazer cast do atributo *"quality"* para inteiro.**

Para fazer *cast* do atributo *quality* para inteiro, utilizaram-se dois nodos: ***String Manipulation*** e ***String to Number***. Como visto anteriormente, as *strings* do atributo *quality* eram do tipo '=i' onde i era um inteiro tal que: $i \in \{3, 4, 5, 6, 7, 8\}$. Dessa forma, para obter apenas a *string* 'i', bastou utilizar a função de *substring* no nodo *String Manipulation*. Após isso, bastou aplicar o nodo *String to Number* de forma a converter as *strings* 'i' apenas no inteiro i. As imagens abaixo mostram a configuração de cada um dos nodos bem como o *workflow* relativo a este exercício.

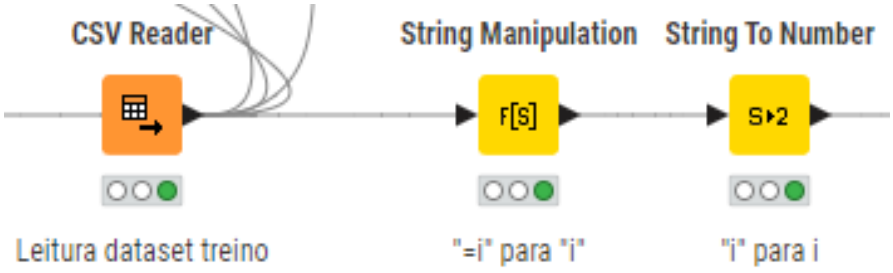

Figura 2.8: *Workflow* relativo à alínea a).

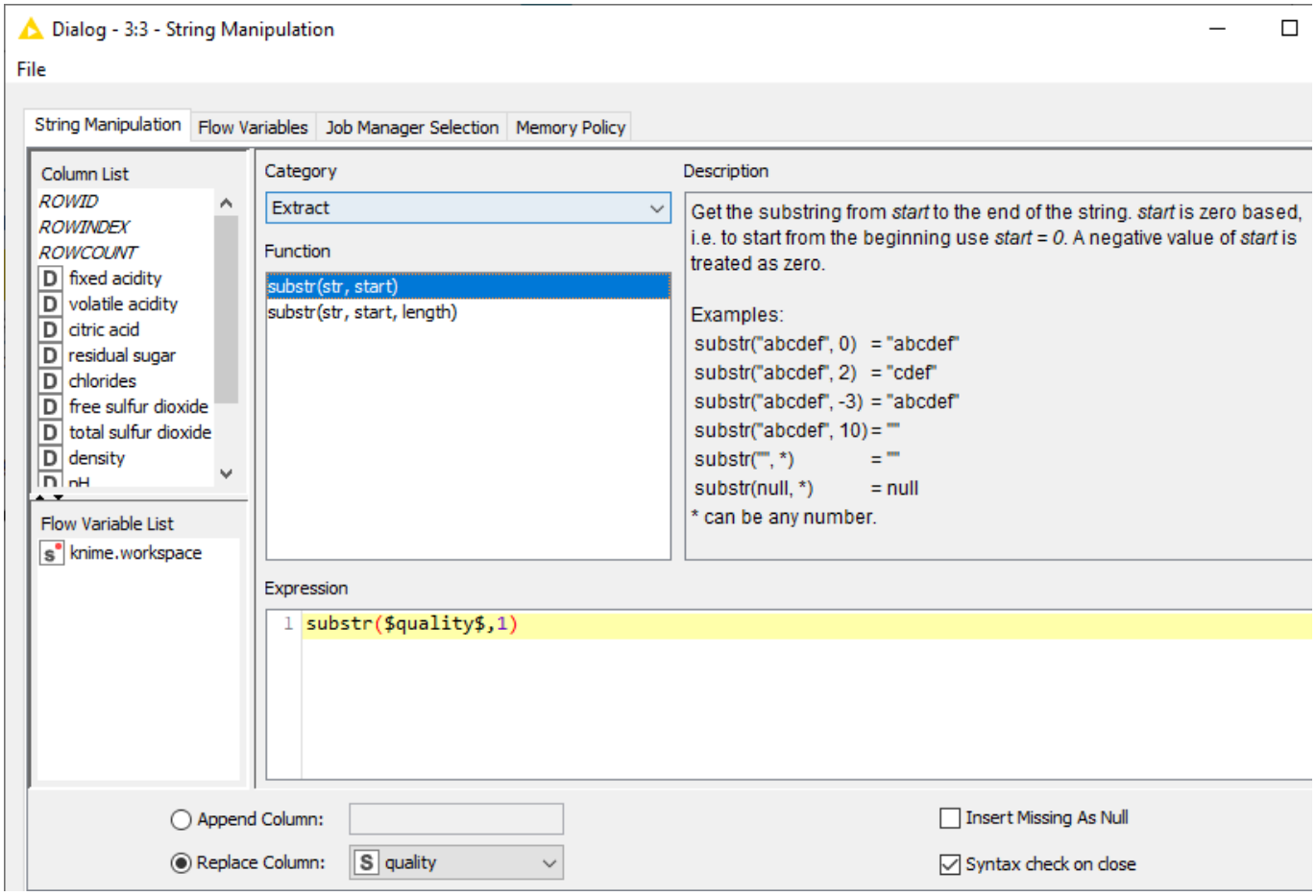

Figura 2.9: Configuração do nodo *String Manipulation*.

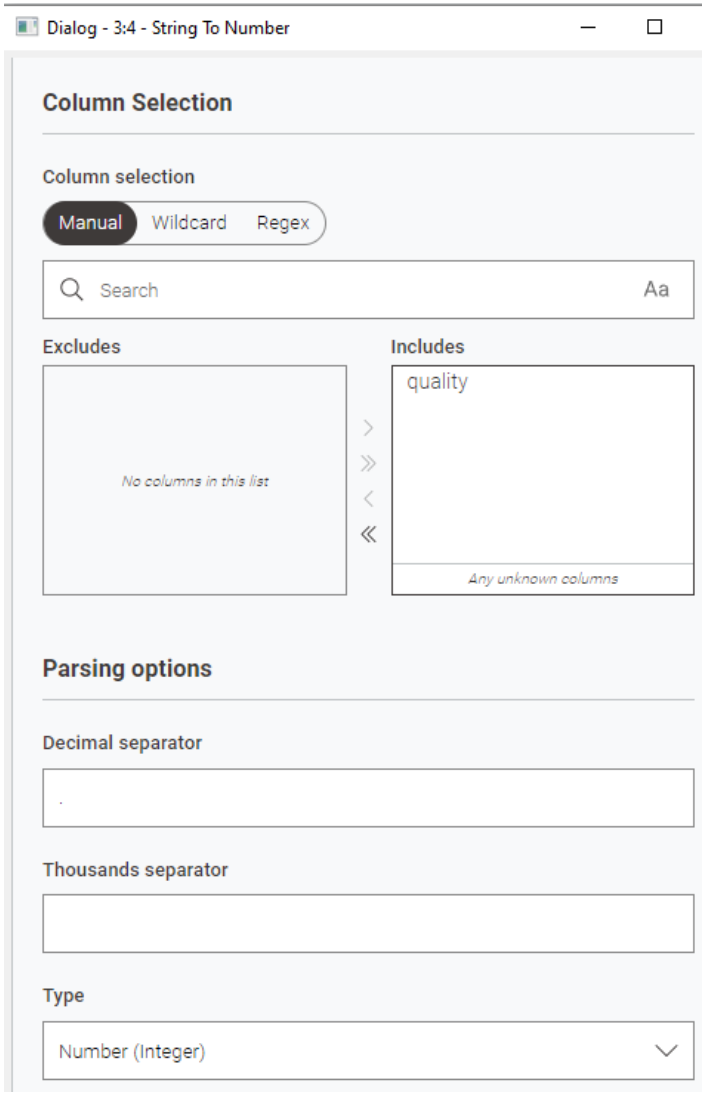

Figura 2.10: `Configuração do nodo` *`String to Number`*.

2.2.2 Alínea b)

- **Normalizar todos os atributos numéricos utilizando a transformação linear Min-max de forma a produzir um input normalizado entre 0 e 1.**

Para este exercício utilizei o nodo ***Normalizer*** configurado como pedido na pergunta, isto é, aplicado a todos os atributos numéricos (que neste momento corresponde a todos os atributos do *dataset*) e com transformação linear Min-max de forma a produzir um input normalizado entre 0 e 1. As imagens abaixo mostram o nodo utilizado, bem como a sua configuração.

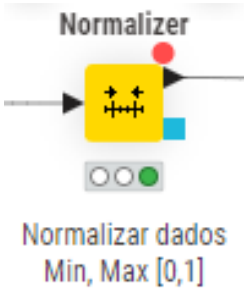

Figura 2.11: Nodo *Normalizer*.

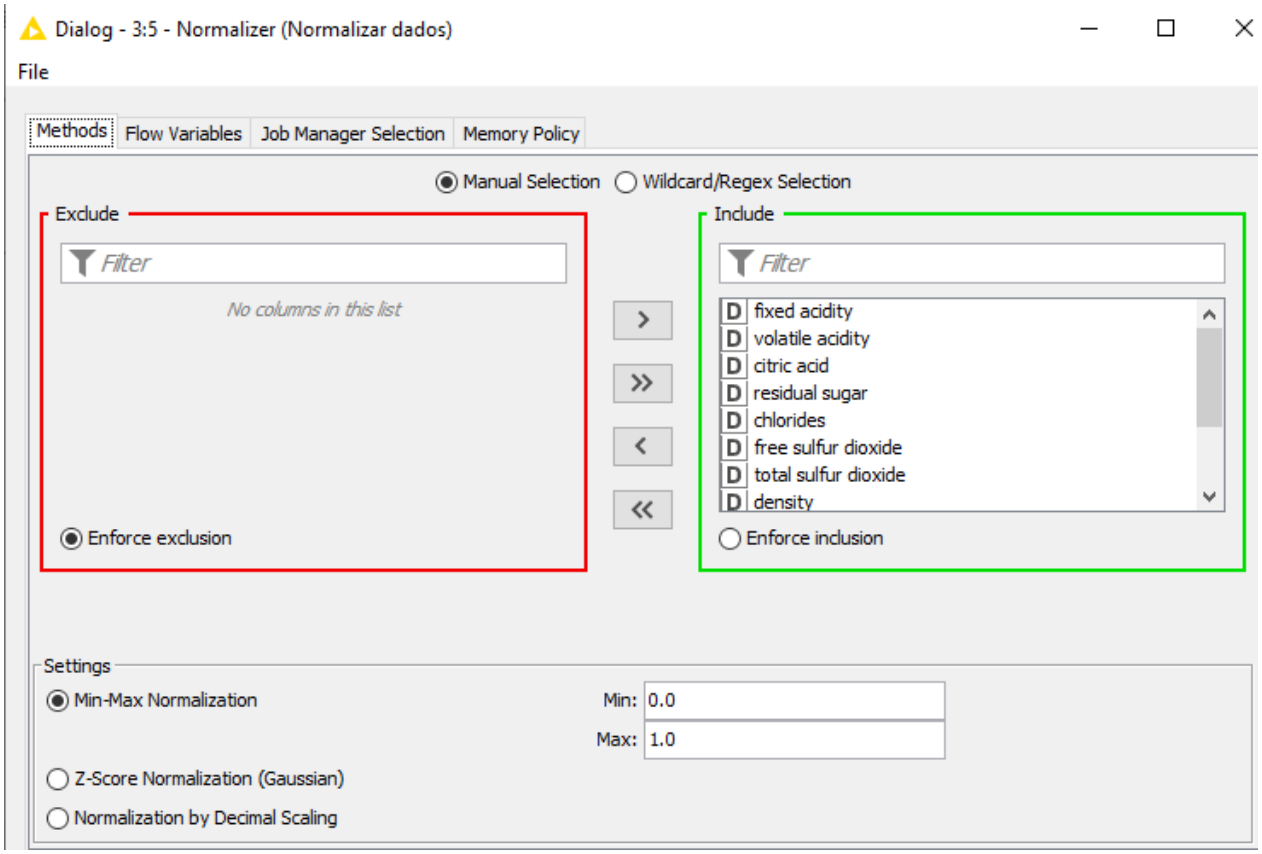

Figura 2.12: Configuração do nodo *Normalizer*.

2.2.3 Alínea c)

- **Criar 4 *bins* de igual frequência para a *feature "citric acid"*, substituindo a *feature* original.**

Para este exercício, utilizei o nodo ***Auto-Binner*** configurado como pede o enunciado, isto é, de forma a criar 4 *bins* de igual frequência para a *feature citric acid*, substituindo a coluna em questão. Além disso, utilizei o nodo ***Data Explorer*** para observar e analisar os bins criados. As imagens abaixo mostram o *workflow* correspondente, a configuração do nodo *Auto-Binner* e a análise aos *bins* criados.

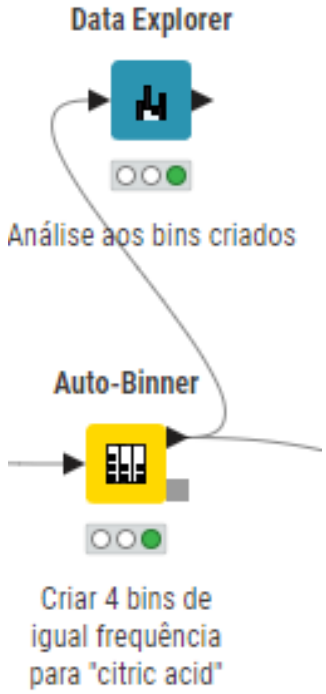

Figura 2.13: *Workflow* correspondente à alínea c).

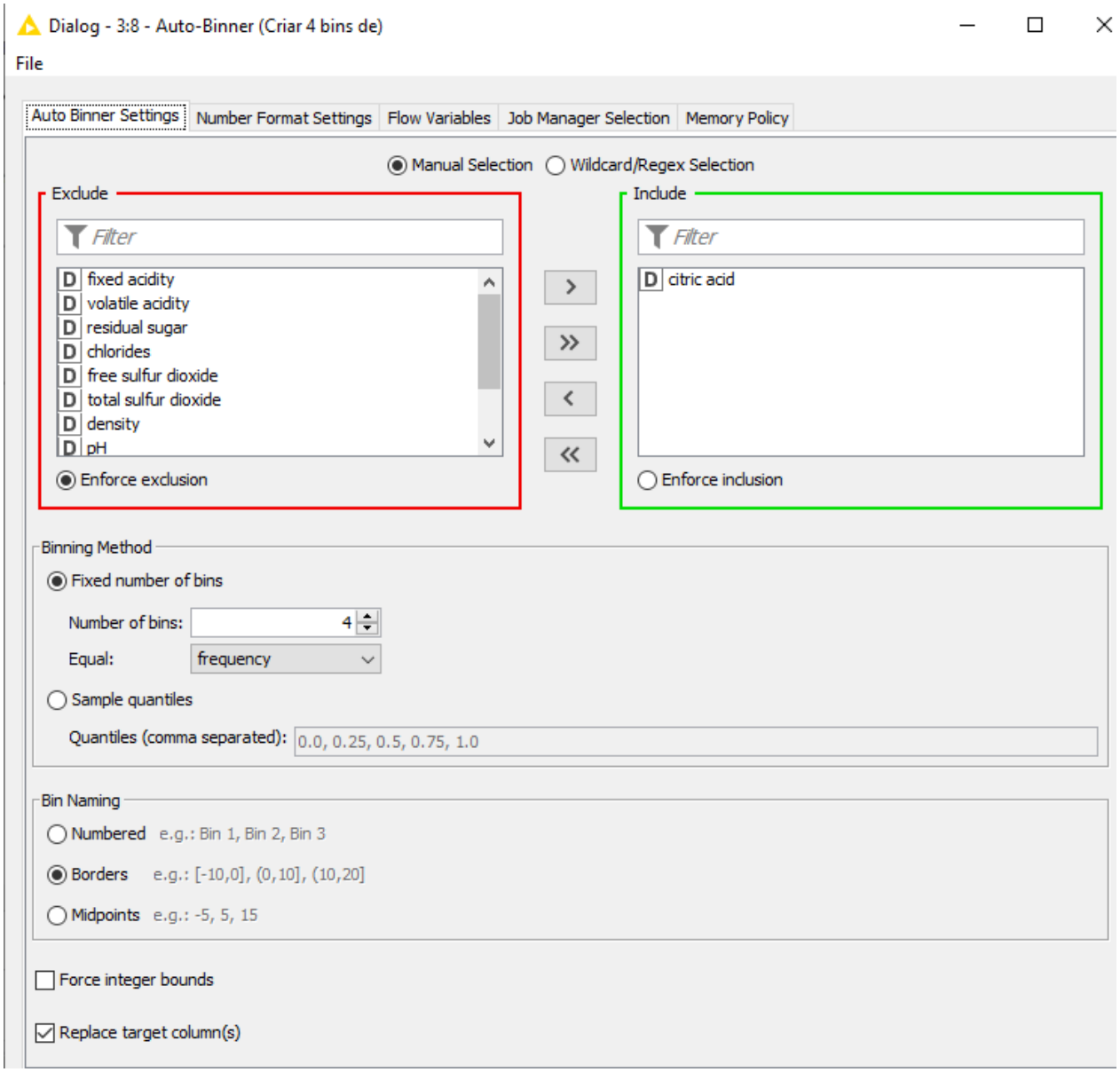

Figura 2.14: Configuração do nodo *Auto-Binner*.

Column	Exclude Column	No. missings	Unique values	All nominal values	Frequency Bar Chart
citric acid	☐	0	4	(0.09,0.26], [0,0.09], (0.43,1], (0.26,0.43]	

Figura 2.15: Análise aos *bins* criados.

2.2.4 Alínea d)

- **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*.**

Para renomear cada um dos *bins* em ***Low, Medium, High*** e ***Very High*** pela respetiva ordem, utilizei o nodo ***String Manipulation*** utilizando a função ***replace*** de forma recursiva para o efeito, com a ajuda do *Data Explorer*, mencionado anteriormente, para saber quais os *bins* que foram criados. Segue abaixo a configuração usada no nodo *String Manipulation*, bem como o *workflow* correspondente a este exercício.

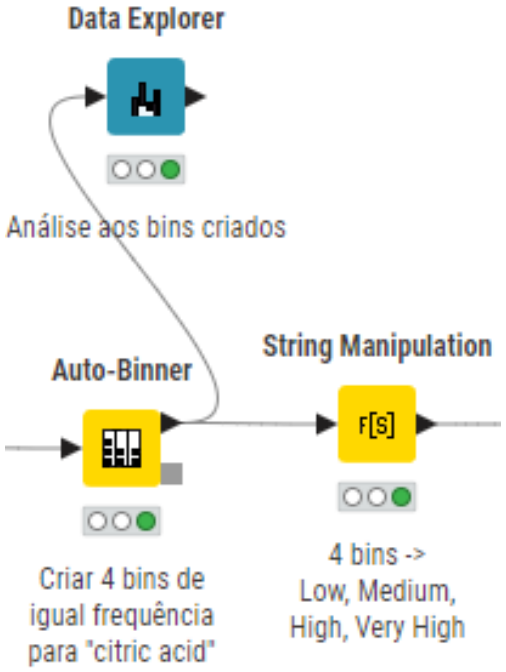

Figura 2.16: *Workflow* correspondente à alínea d).

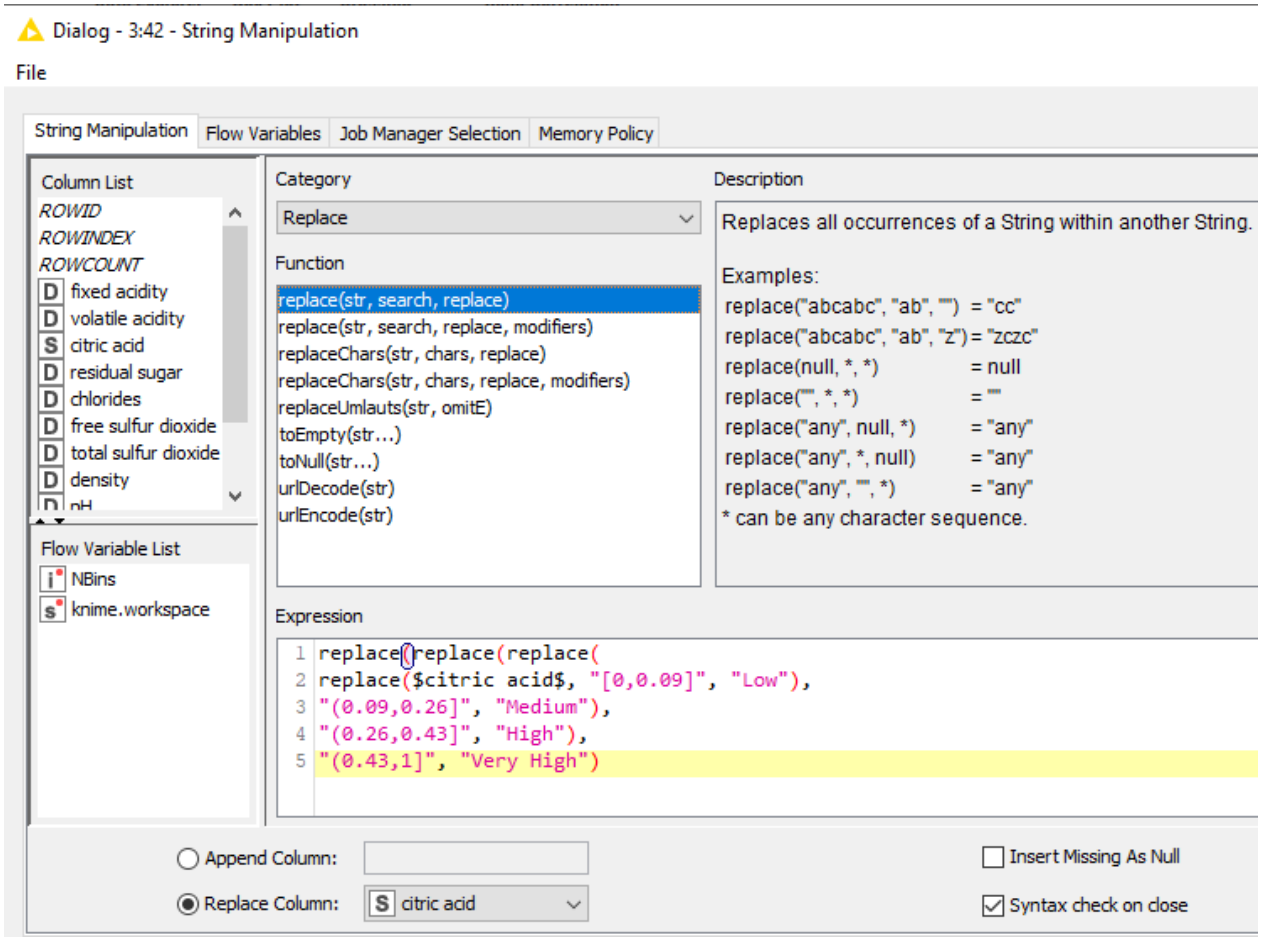

Figura 2.17: Configuração do nodo *String Manipulation*.

2.3 Tarefa 3

2.3.1 Alínea a)

- **Aplicar uma Análise de Componentes Principais (PCA) de forma a projetar os dados em apenas duas dimensões.**

Para este exercício, usei o nodo ***PCA*** configurado como pedido no enunciado, isto é, de forma aos dados serem projetados em apenas duas dimensões. As imagens abaixo ilustram o nodo e a configuração do mesmo.

Figura 2.18: Nodo *PCA*.

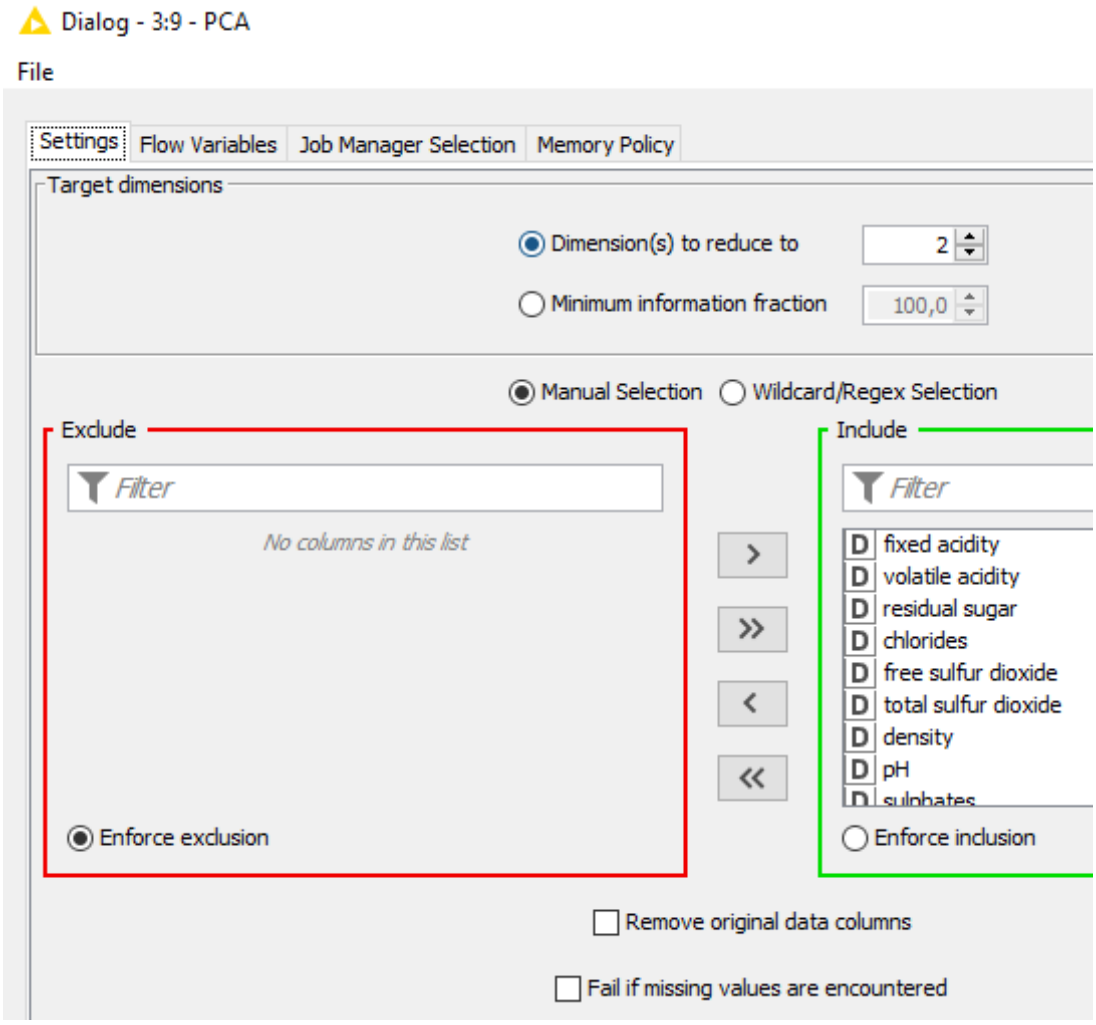

Figura 2.19: Configuração do nodo *PCA*.

2.3.2 Alínea b)

- **Utilizar um *scatter plot* para visualização dos resultados obtidos pelo PCA.**

Neste exercício utilizei o nodo ***Scatter Plot*** para obter um gráfico dos dados projetados nas duas dimensões obtidas no nodo *PCA*. As imagens abaixo ilustram o gráfico e o *workflow* correspondente.

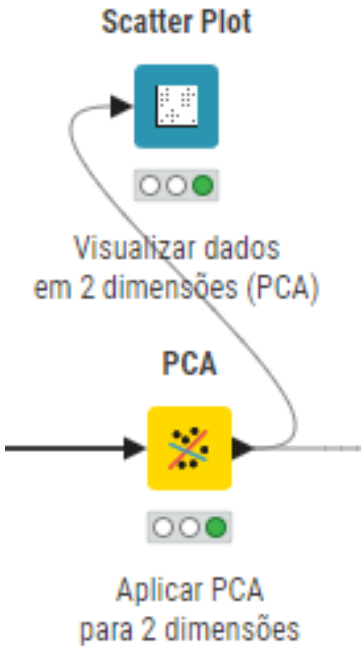

Figura 2.20: *Workflow* correspondente à alínea b).

Figura 2.21: `Visualização dos dados projetados nas duas dimensões obtidas no` *`PCA`*.

2.4 Tarefa 4 (Segmentação do *dataset*)

2.4.1 Alínea a)

- **Aplicar o método *k-means*.**

Neste exercício utilizei o nodo ***K-Means*** que nos permite criar um modelo de ***clustering*** com base nos dados de treino. Neste caso, o modelo está configurado para separar os dados em 5 *clusters*. As imagens abaixo representam o nodo *K-Means*, bem como a configuração do mesmo.

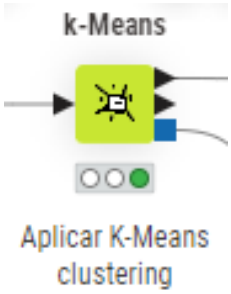

Figura 2.22: Nodo *K-Means*.

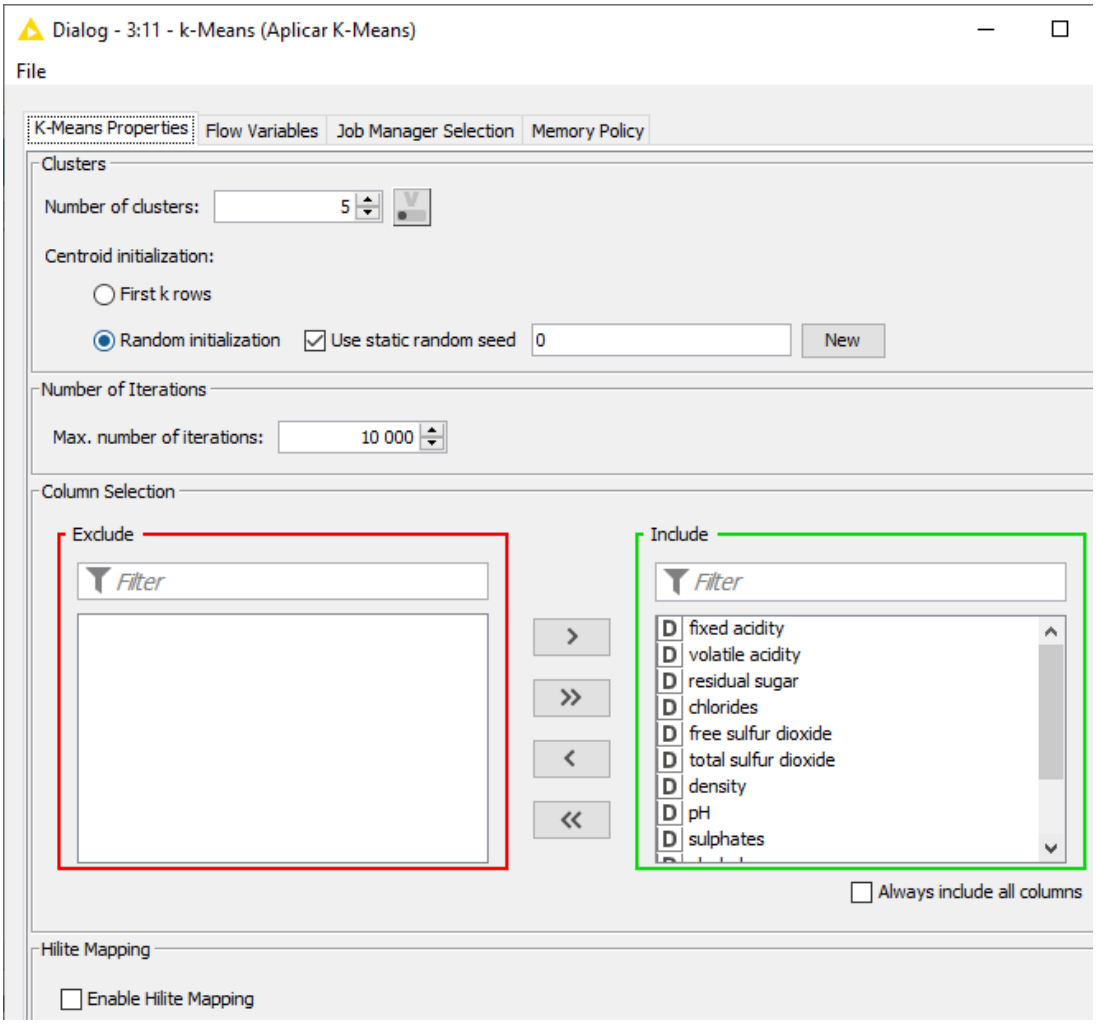

Figura 2.23: Configuração do nodo *K-Means*.

2.4.2 Alínea b)

- **Atribuir diferentes cores por qualidade do vinho e diferentes formas aos clusters.**

Para este exercício utilizei dois nodos: ***Color Manager*** e ***Shape Manager***. O primeiro foi utilizado com o intuito de atribuir diferentes cores por qualidade de vinho e o segundo para definir diferentes formas para cada *cluster*. As imagens abaixo ilustram o *workflow* relativo a este exercício, bem como a configuração dos nodos *Color Manager* e *Shape Manager* que contêm as cores e as formas escolhidas para o efeito, respetivamente.

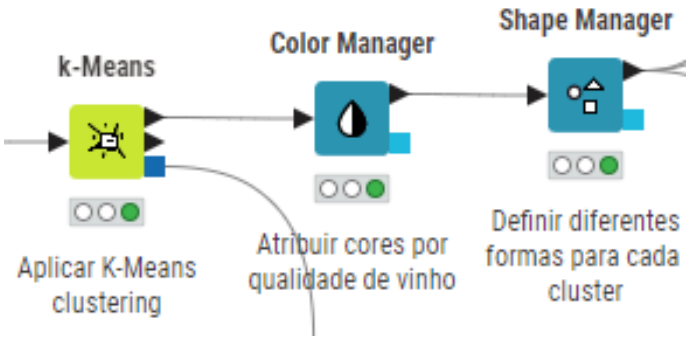

Figura 2.24: *Workflow* relativo à alínea b).

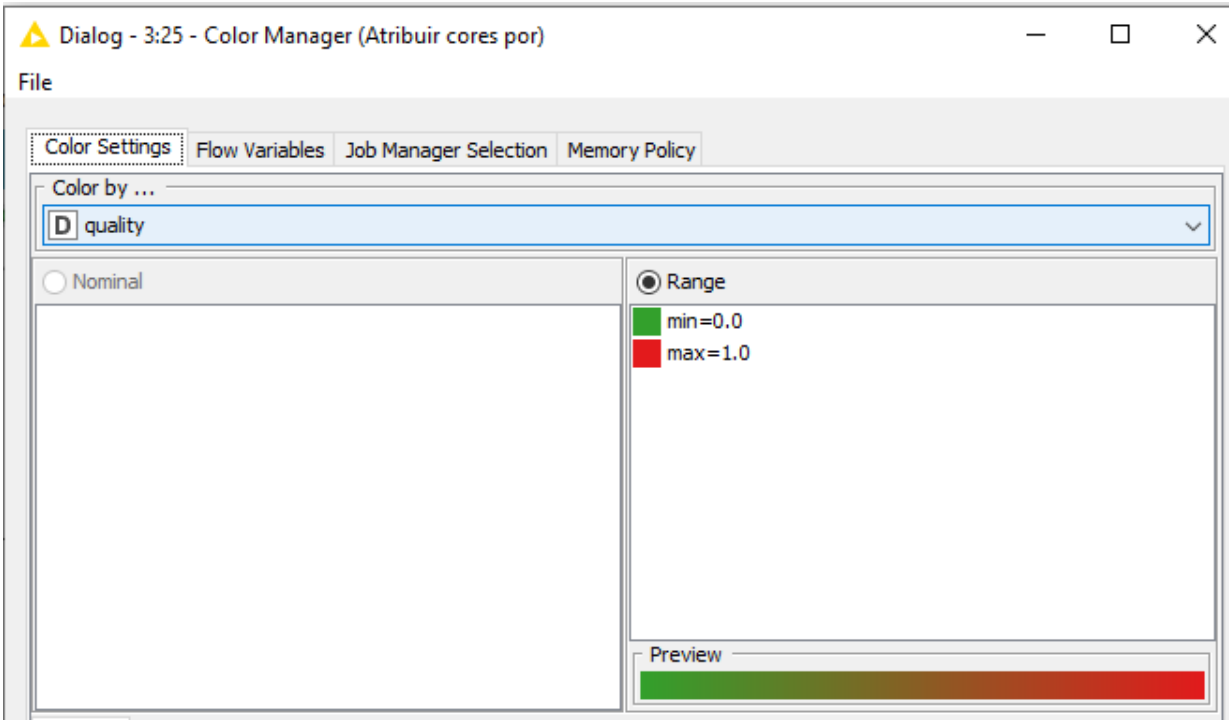

Figura 2.25: Configuração do nodo *Color Manager*.

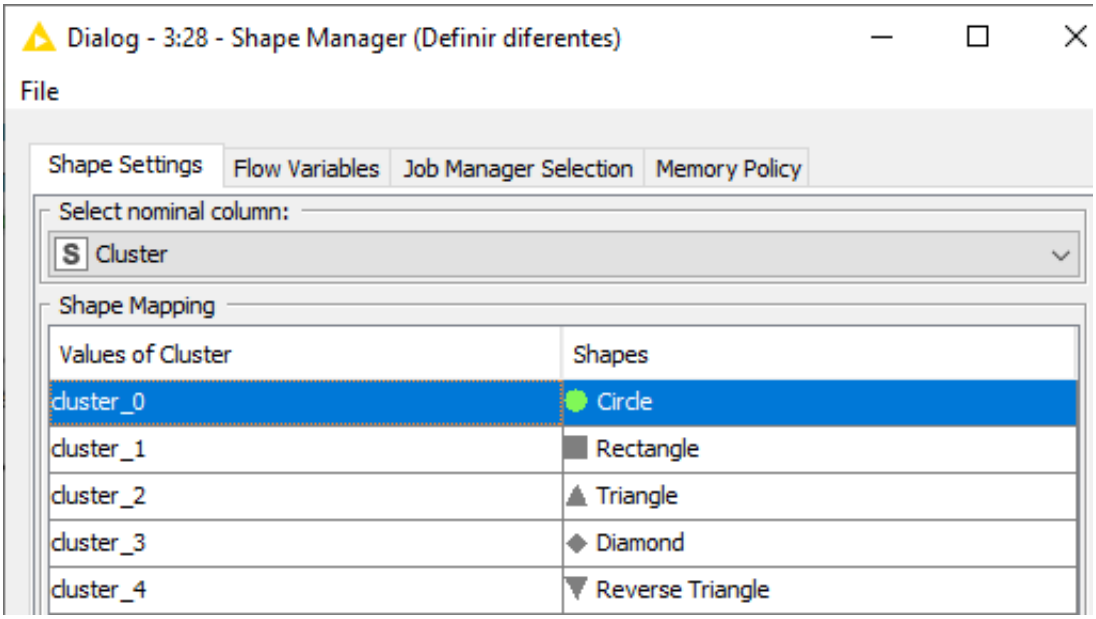

Figura 2.26: `Configuração do nodo` *`Shape Manager`*.

2.4.3 Alínea c)

- **Criar *scatter plots* e *scatter matrixes* que permitam ter uma noção gráfica, em duas dimensões, dos atributos e dos clusters criados.**

Para visualizar os atributos e os *clusters* em duas dimensões, utilizei os nodos ***Scatter Plot***, ***Scatter Plot (legacy)*** e ***Scatter Matrix (legacy)***. O primeiro permite-nos visualizar as duas dimensões do *PCA* com as cores associadas à qualidade do vinho. Já no segundo, acrescenta-se as formas definidas para cada *cluster* conseguindo perceber, por isso mesmo, o *cluster* associado a cada dado. Por fim, no nodo *Scatter Matrix (legacy)*, fiz *plots*, em duas dimensões, das duas dimensões do *PCA* e do atributo *Cluster* (criado no nodo *K-Means*) com cores e formas, obtendo por isso 6 *plots*. As imagens abaixo mostram o *workflow* relativo a este exercício, bem como todos os gráficos gerados.

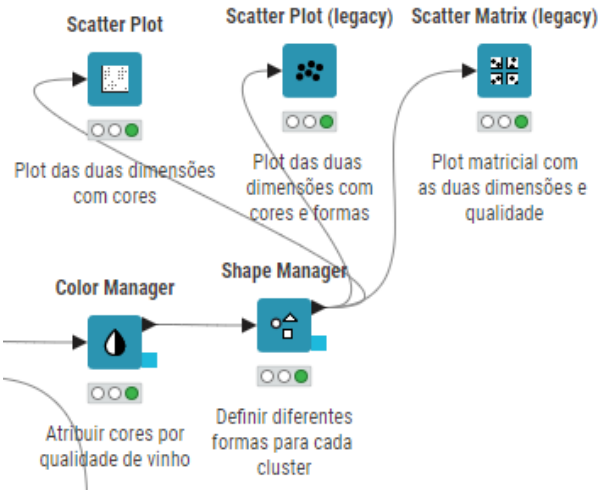

Figura 2.27: *`Workflow`* `relativo à alínea c).`

Figura 2.28: *Scatter Plot*.

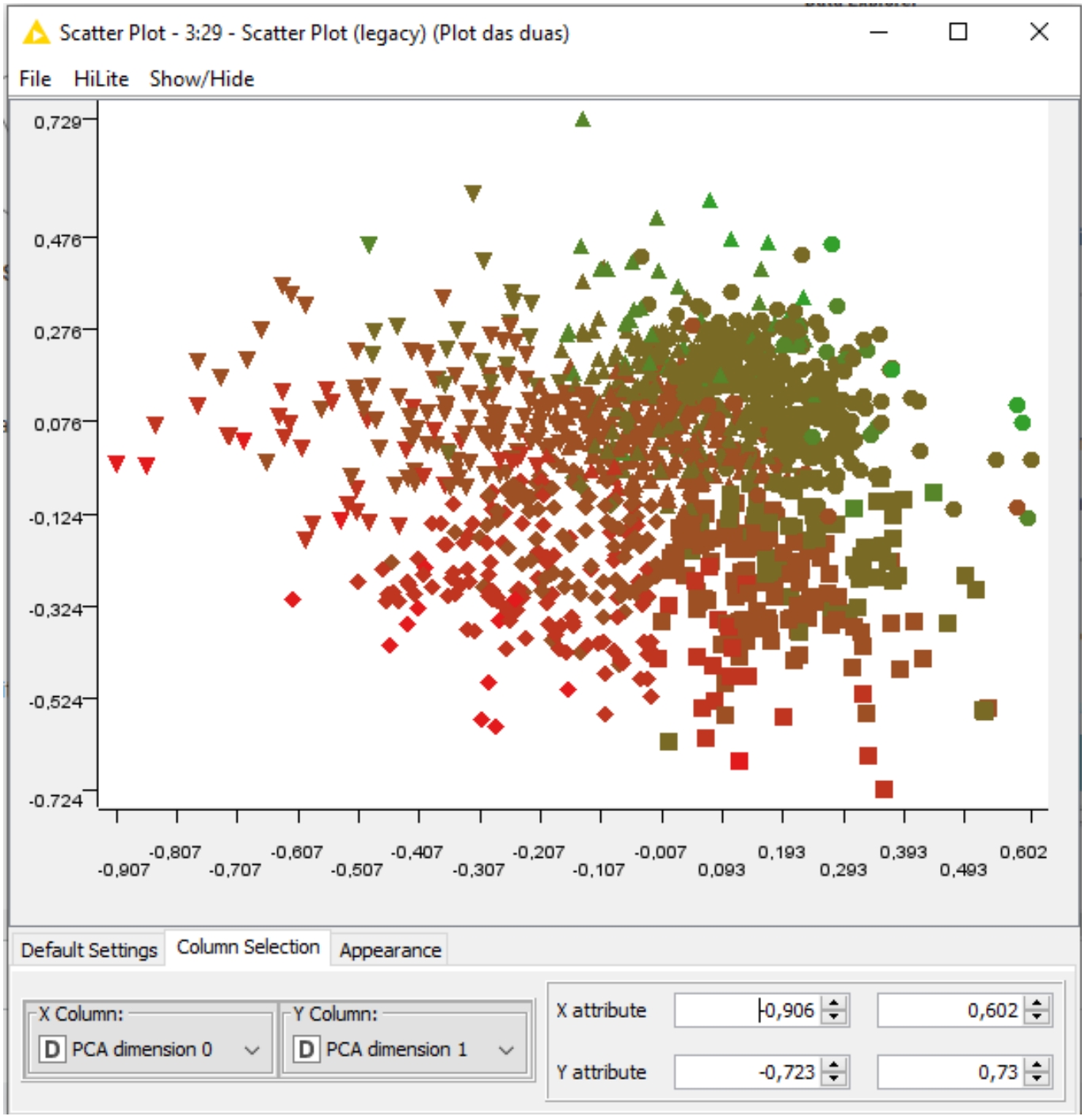

Figura 2.29: *Scatter Plot (legacy).*

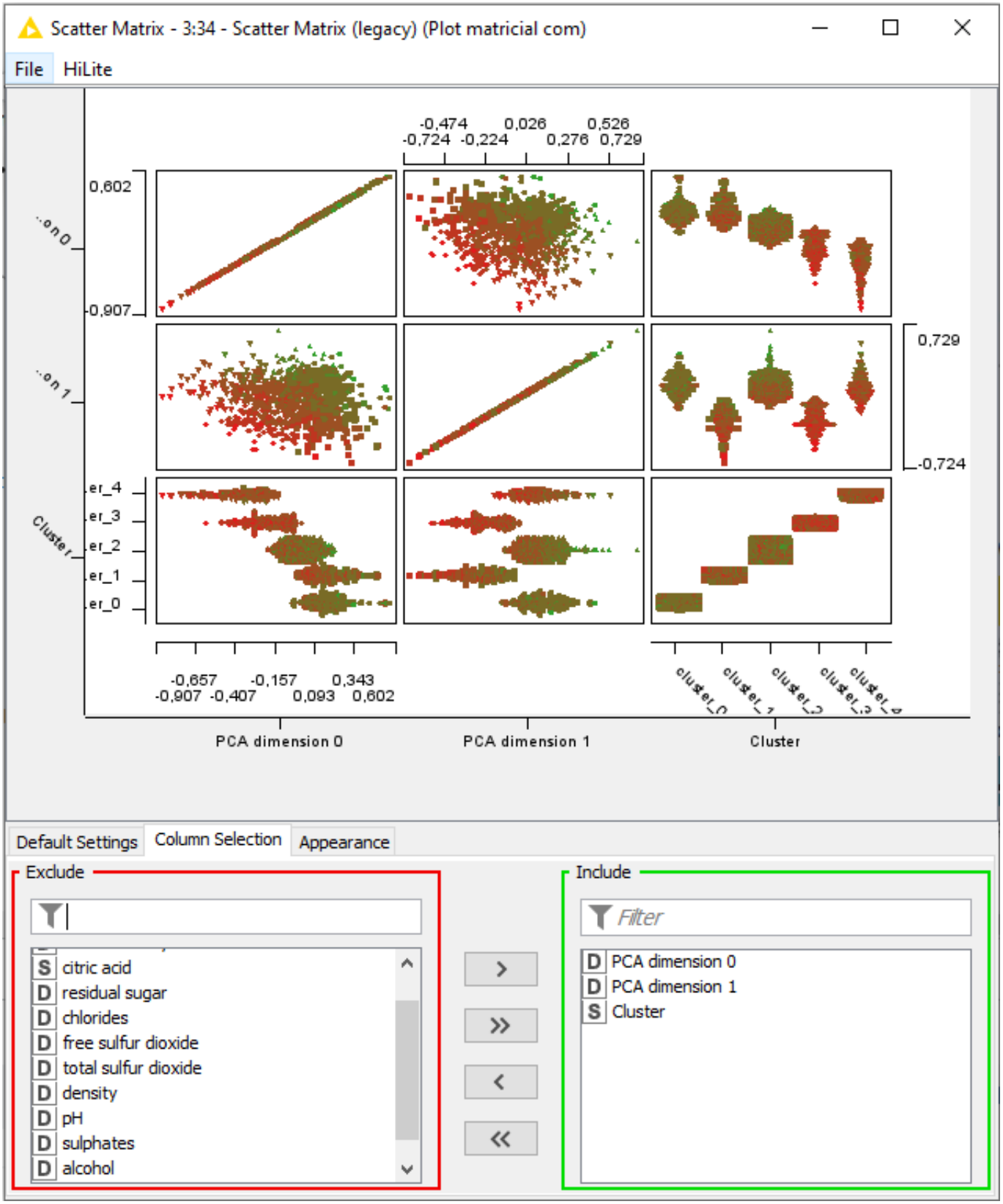

Figura 2.30: *Scatter Matrix (legacy).*

2.4.4 Alínea d)

- **Ler e tratar os dados de teste de forma a que, com base no modelo desenvolvido nos passos anteriores, seja atribuído um cluster a cada registo deste ficheiro.**

Neste exercício, comecei por aplicar todo o tratamento que foi feito nos dados de treino aos dados de teste, até à aplicação do *PCA* (inclusivamente). Após isso, bastou utilizar o nodo ***Cluster Assigner*** que, com base no modelo de *clustering K-Means* feito anteriormente, atribui a cada registo dos dados de teste, um dos 5 clusters. As imagens abaixo ilustram todo o tratamento feito nos dados de teste (igual ao tratamento dos dados de treino), o nodo *Cluster Assigner* (conectado ao modelo no nodo *K-Means*) e ainda as previsões dos 9 primeiros registos dos dados de teste (para fins de exemplificação).

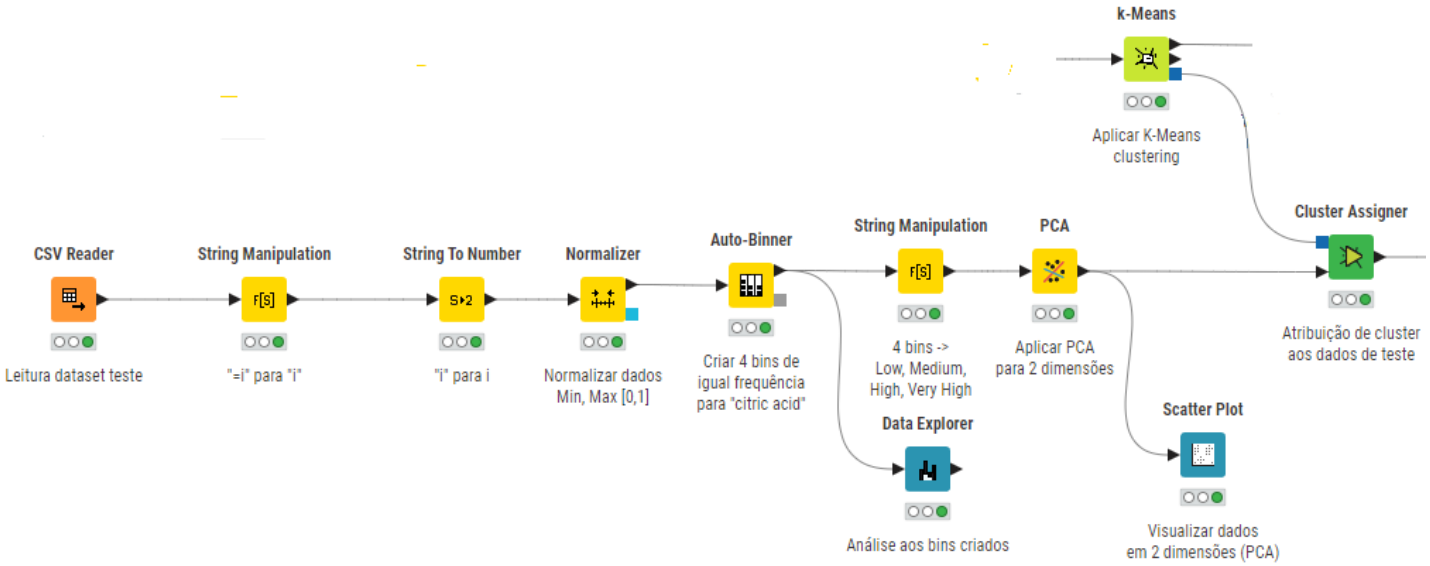

Figura 2.31: *Workflow* relativo à alínea d).

PCA dim... Number (dou...	PCA dim... Number (dou...	Cluster String
-0.312	-0.31	cluster_4
0.152	-0.29	cluster_2
0.612	-0.108	cluster_0
0.358	-0.04	cluster_0
-0.338	-0.082	cluster_4
-0.205	0.038	cluster_3
0.07	-0.569	cluster_2
-0.187	-0.433	cluster_3
-0.205	0.038	cluster_3

Figura 2.32: `Previsão dos primeiros 9 registos dos dados de teste.`

2.4.5 Alínea e)

- **Guardar o resultado da atribuição num ficheiro csv.**

Para guardar o resultado da atribuição num ficheiro *csv*, bastou utilizar o nodo ***CSV Writer*** conectado ao nodo *Cluster Assigner*, de modo a guardar a tabela com as previsões, proveniente do *Cluster Assignment*, num ficheiro *csv*, na diretoria especificada na configuração do nodo *CSV Writer*. As imagens abaixo ilustram o nodo utilizado e a ainda a sua configuração.

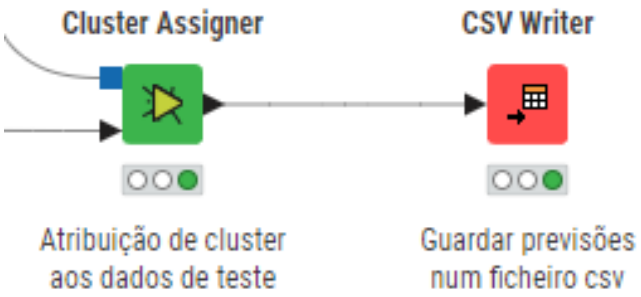

Figura 2.33: *`Workflow`* `relativo à alínea e).`

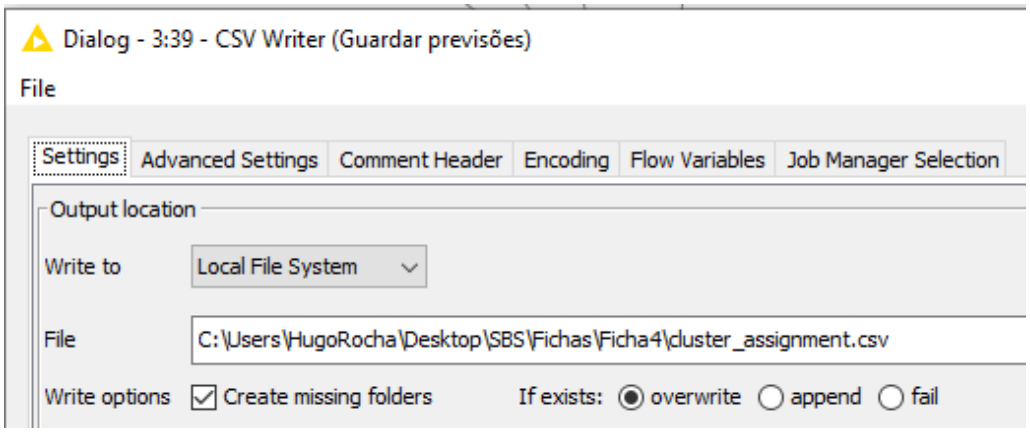

Figura 2.34: `Configuração do nodo` *`CSV Writer`*.

2.5 Tarefa 5

- **Parametrizar o *workflow*, utilizando variáveis de fluxo para definir o número de *bins*, o número de *clusters* e os títulos dos gráficos criados.**

. Para este exercício, criei variáveis de fluxo nas configurações dos respetivos nodos, neste caso, *Auto-Binner*, *K-Means* e todos os nodos de criação de gráficos, ficando essas mesmas variáveis com os valores especificados na configuração do nodo. A imagens abaixo ilustram as *flow variables* criadas e ainda a forma como são criadas.

Flow Variables

Count: 7

Owner ID	Data Type	Variable Name	Value
3:45:0:21	StringType	VisualizaçãoPCAtesteTitle	Visualização PCA dataset teste
3:45:0:41	StringType	OutliersTitle	Deteção de Outliers
3:45:0:37	StringType	VisualizaçãoPCAcoresTitle	Visualização PCA com cores por qualidade de vinho
3:45:0:14	StringType	VisualizacaoPCAtreinoTitle	Visualização PCA dataset treino
3:45:0:42	IntType	NClusters	5
3:45:0:42	IntType	NBins	4

Figura 2.35: *`Flow variables`* `criadas.`

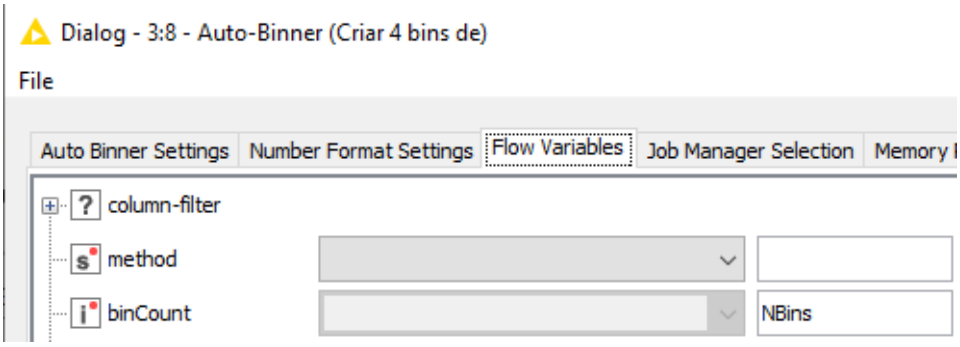

Figura 2.36: Exemplo da criação da *flow variable* que define o número de *bins*.

2.6 Tarefa 6

- **Produzir o *workflow* de maneira a que seja possível visualizar, numa única página, todos os componentes visuais implementados.**

Para visualizar todos os componentes visuais implementados numa só página, criei um componente composto por todos os nodos relativos a gráficos. Dessa forma, ao clicar em *Open View* sobre o componente, é possível observar todos os gráficos numa única página. Além disso, ao abrir o componente, tive o cuidado de passar as variáveis de fluxo já criadas para dentro do componente através do ***Component Input***. Neste componente são definidas as variáveis de fluxo relativas aos títulos dos gráficos sobre as quais tive o cuidado de deixar sair do componente através do ***Component Output***. As imagens abaixo ilustram todo este processo.

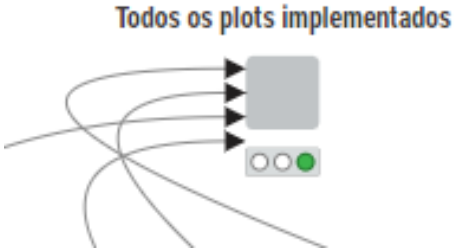

Figura 2.37: Componente criado.

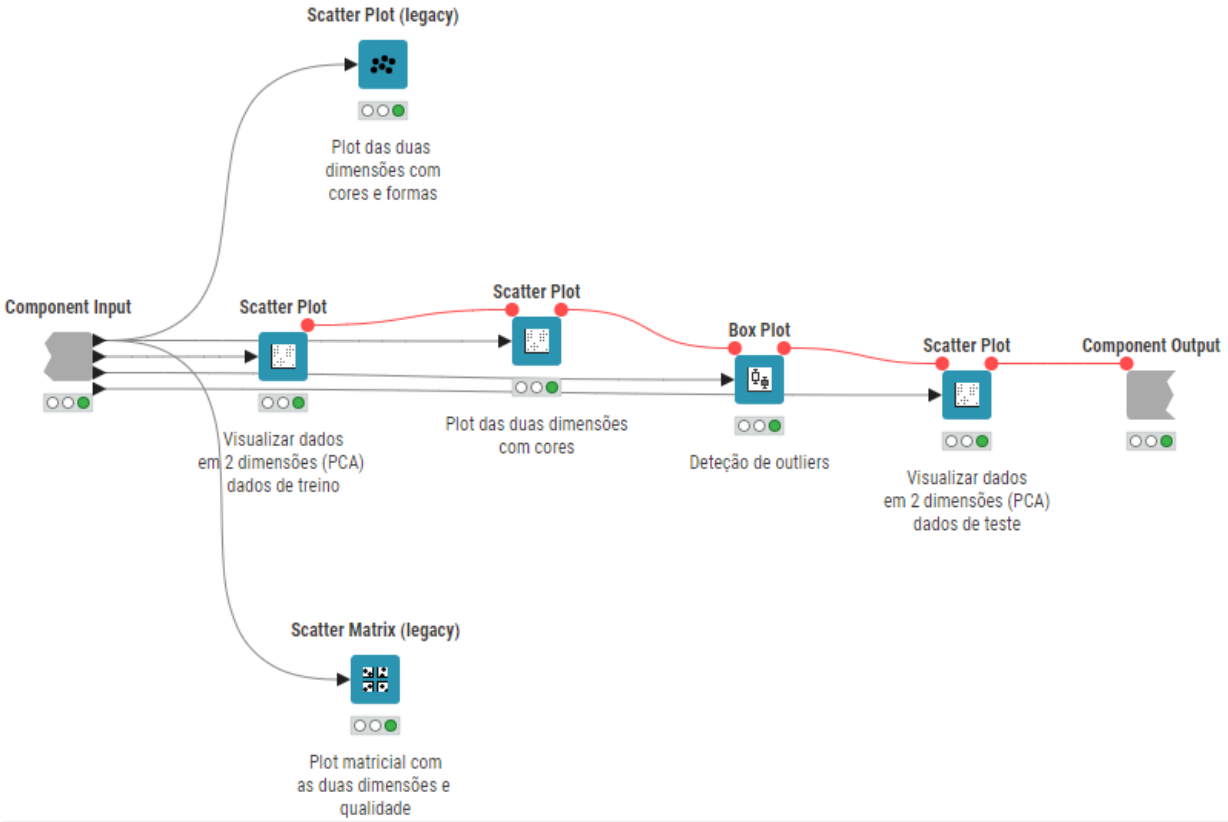

Figura 2.38: `Visualização interior do componente.`

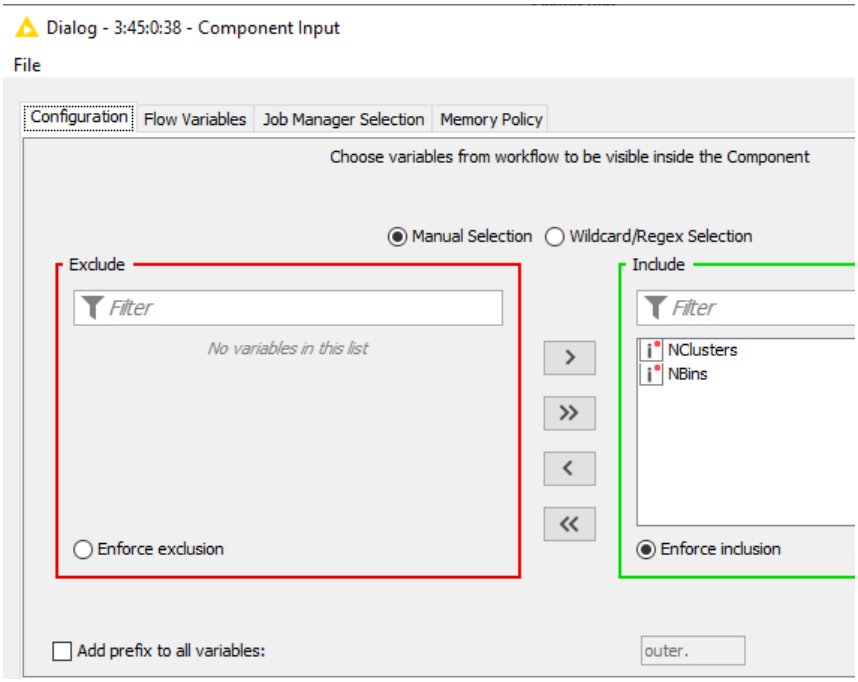

Figura 2.39: `Entrada de` *`flow variables`* `já criadas, no componente.`

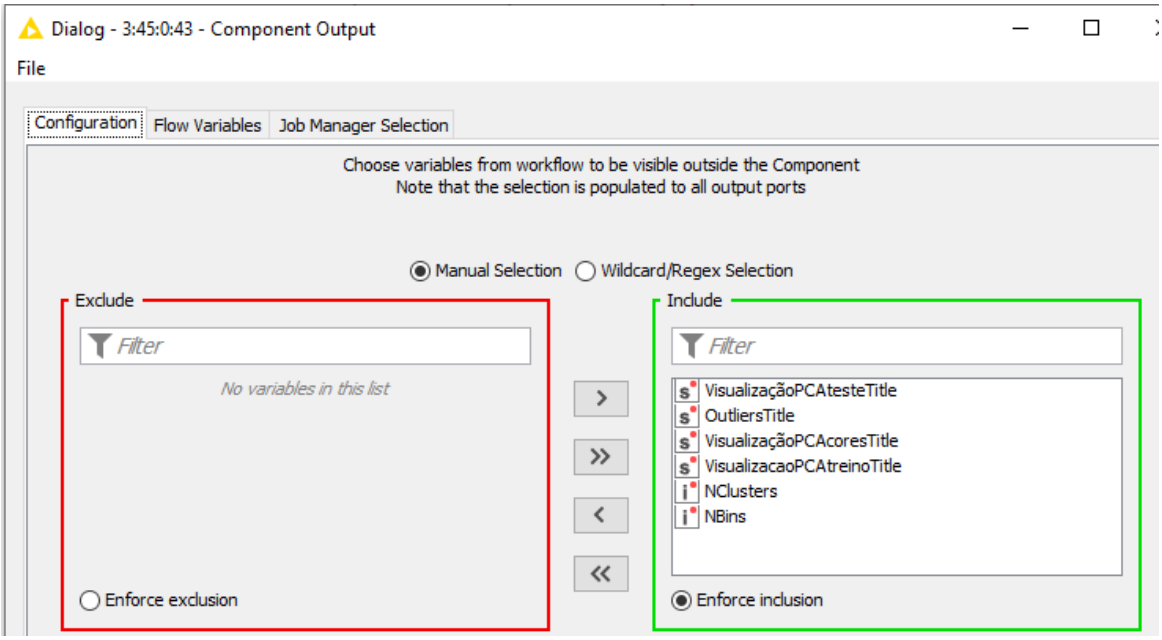

Figura 2.40: `Saída de` *`flow variables`* `criadas do componente.`

2.7 Tarefa 7

- **Experimentar, avaliar e comparar outros métodos de segmentação.**

Nesta tarefa experimentei outros metodos de segmentação tais como: ***K-Medoids*** e ***DBSCAN***. A utilizar *K-Medoids* escolhi de novo separar os dados em 5 *clusters* usando a métrica de *Manhattan* e obtive uma divisão diferente daquela que foi feita com o *K-Means*. Já com o método *DBSCAN* aconteceu algo curioso visto que, com o valor de *epsilon* a 1 e com número de pontos mínimos a 3, a usar a métrica euclidiana, estranhamente (ou talvez não), o modelo colocou todos os dados num só *cluster*. As imagens abaixo ilustram a divisão dos dados em cada método de segmentação.

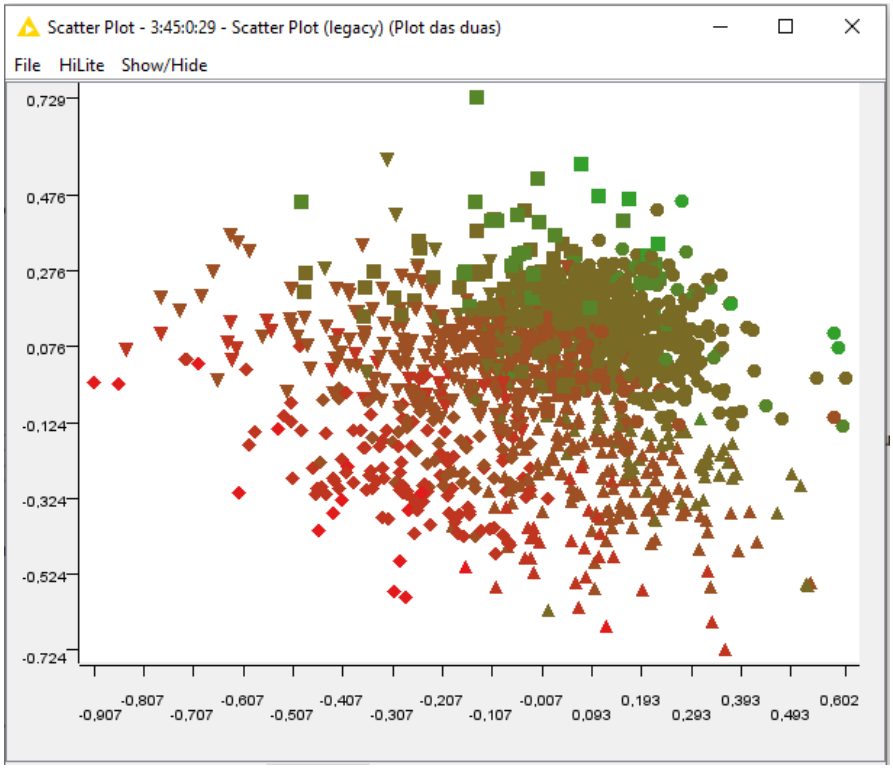

Figura 2.41: *K-Medoids*.

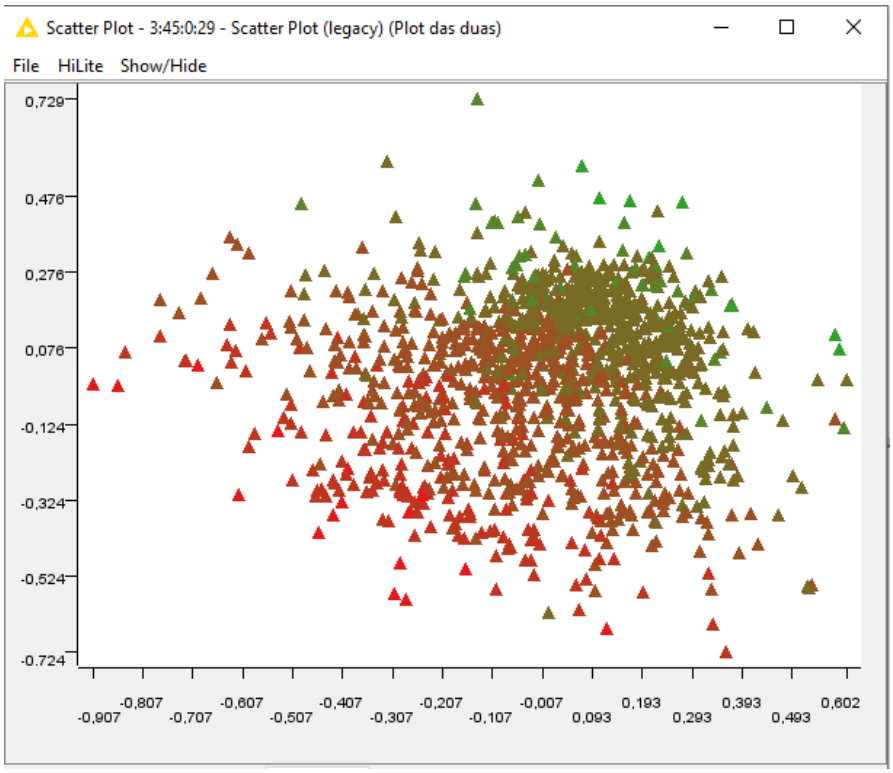

Figura 2.42: *DBSCAN*.

2.8 *Workflow* completo

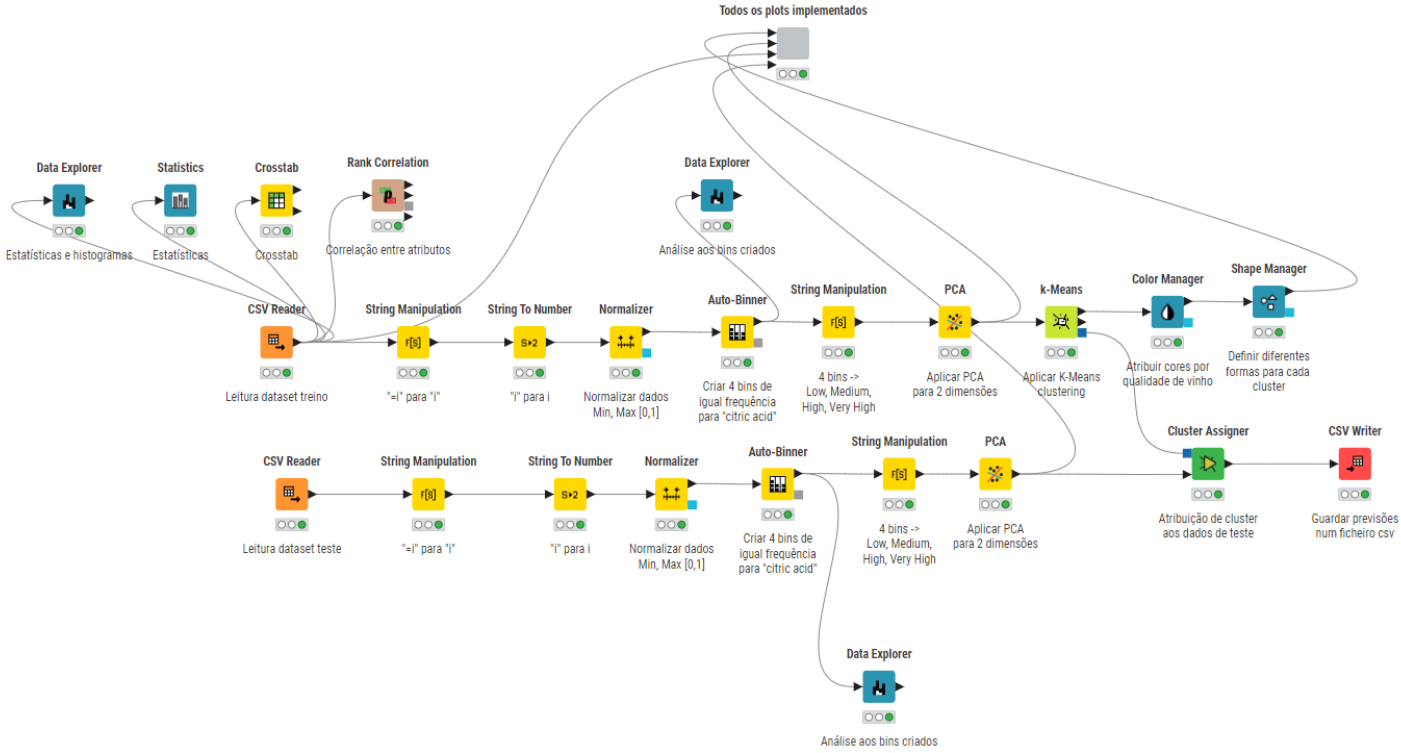

Figura 2.43: *Workflow* completo de toda a ficha.