



Universidade do Minho
Escola de Ciências

UNIVERSIDADE DO MINHO
MESTRADO EM MATEMÁTICA E COMPUTAÇÃO

Sistemas Baseados em Similaridade

Ficha Prática Individual 2

Hugo Filipe de Sá Rocha (PG52250)

4 de outubro de 2023

Conteúdo

1	Contextualização	3
2	Concepção das soluções	4
2.1	Tarefa 1	4
2.2	Tarefa 2	6
2.3	Tarefa 3	8
2.4	Tarefa 4	11
2.5	Tarefa 5	13
2.6	Tarefa 6	13
2.7	Workflow completo (sem usar a maioria dos meta-nodos)	15

Capítulo 1

Contextualização

No setor das telecomunicações, *churn* é uma medida do número de clientes que estão a sair de uma operadora. Os clientes poderão estar de saída porque encontraram preços mais baixos na concorrência ou porque estão desagrados com o serviço prestado, entre outros motivos. Assim, para uma operadora de telecomunicações, torna-se imperativo que existam modelos capazes de prever a possibilidade de *churn* de um cliente, isto é, a possibilidade de um cliente estar de saída. Isto permitirá que a operadora tente segurar o cliente antes que este opte pela saída, oferecendo melhores serviços ou preços mais atrativos e é neste sentido que o trabalho é realizado. Com base em dois *datasets* com informação acerca dos clientes, o objetivo passa por criar um modelo de qualidade e com eficácia no que toca à previsão sobre o *churn* de um cliente.

Capítulo 2

Concepção das soluções

2.1 Tarefa 1

- Carregar, no *Knime*, ambos os *datasets*. Utilizar um nodo *Joiner* para agregar, por “*area code*” e “*phone*”, os dados provenientes das duas *readers*. Transformar o atributo *Churn* em nominal.

Visto que um dos *datasets* é um ficheiro em formato *Excel* e o outro em formato *CSV*, utilizei dois nodos distintos, um para cada ficheiro. No caso do ficheiro em formato *Excel*, utilizei o nodo *Excel Reader* e, para o ficheiro em formato *CSV*, utilizei o nodo *CSV Reader*, configurando ambos com o caminho para o respetivo ficheiro. De forma a juntar a informações dos dois ficheiros, utilizei o nodo *Joiner*, configurado de forma a que a informação proveniente dos dois *readers* fosse juntada por “*area code*” e por “*phone*”. Após tudo isto, utilizei o nodo *Number to String* para alterar o tipo do atributo *Churn* para *String*. Para melhor clareza do problema em questão, apliquei ainda o nodo *String Manipulation* para transformar os valores do atributo *Churn* de “0” e “1” para “*Remained*” e “*Abandoned*”, respetivamente. Isto foi feito através da função **replace** que nos permite substituir, neste caso, uma *String* por outra, para cada valor de *Churn*.

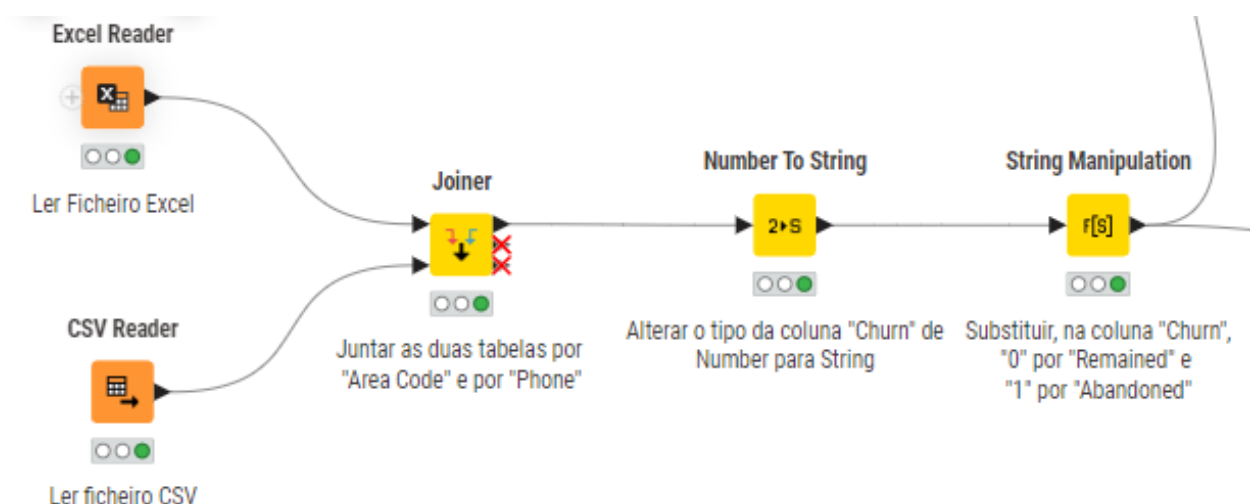


Figura 2.1: *Workflow* da Tarefa 1

Join columns

Match ☒ all of the following ☐ any of the following

Top Input ('left' table)	Bottom Input ('right' table)		
<input type="text" value="I Area Code"/>	<input type="text" value="I Area Code"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="S Phone"/>	<input type="text" value="S Phone"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
		<input type="button" value="+"/>	

Compare values in join columns by ☒ value and type ☐ string representation ☐ making integer types compatible

Include in output

☒ Matching rows

☐ Left unmatched rows

☐ Right unmatched rows

Inner join

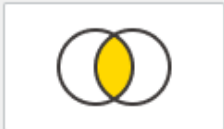


Figura 2.2: Configuração do nodo *Joiner*

Expression

1 `replace(replace($Churn$, "0", "Remained"), "1", "Abandoned")`

☐ Append Column:

☒ Replace Column:

☐ In

☒ S

Figura 2.3: Configuração do nodo *String Manipulation*

2.2 Tarefa 2

- Aplicar nodos para exploração de dados, i.e., analisar os dados em relação às suas características e padrões, procurando extrair informação relevante dos dados.

Para explorar e analisar os dados após toda a organização feita na Tarefa 1, utilizei os nodos *Data Explorer*, *Crosstab (legacy)* e *Statistics* que nos dão todo um conjunto de estatísticas sobre os dados da tabela em análise. Uma das principais características visíveis no *Data Explorer* é que a grandessíssima maioria dos clientes mantêm-se na empresa (2850) em comparação com aqueles que abandonam a empresa (483). Esta estatística poderá ser interessante na análise do nosso modelo, posteriormente, na medida em que, por exemplo, se o modelo prever todos os casos como *Remained*, o modelo deverá ter uma eficácia probabilística relativamente alta mas, no entanto, o mesmo não apresenta qualquer utilidade do ponto de vista prático. Outra estatística boa do *dataset*, visível no nodo *Statistics*, é que não temos campos em falta no conjunto de dados.

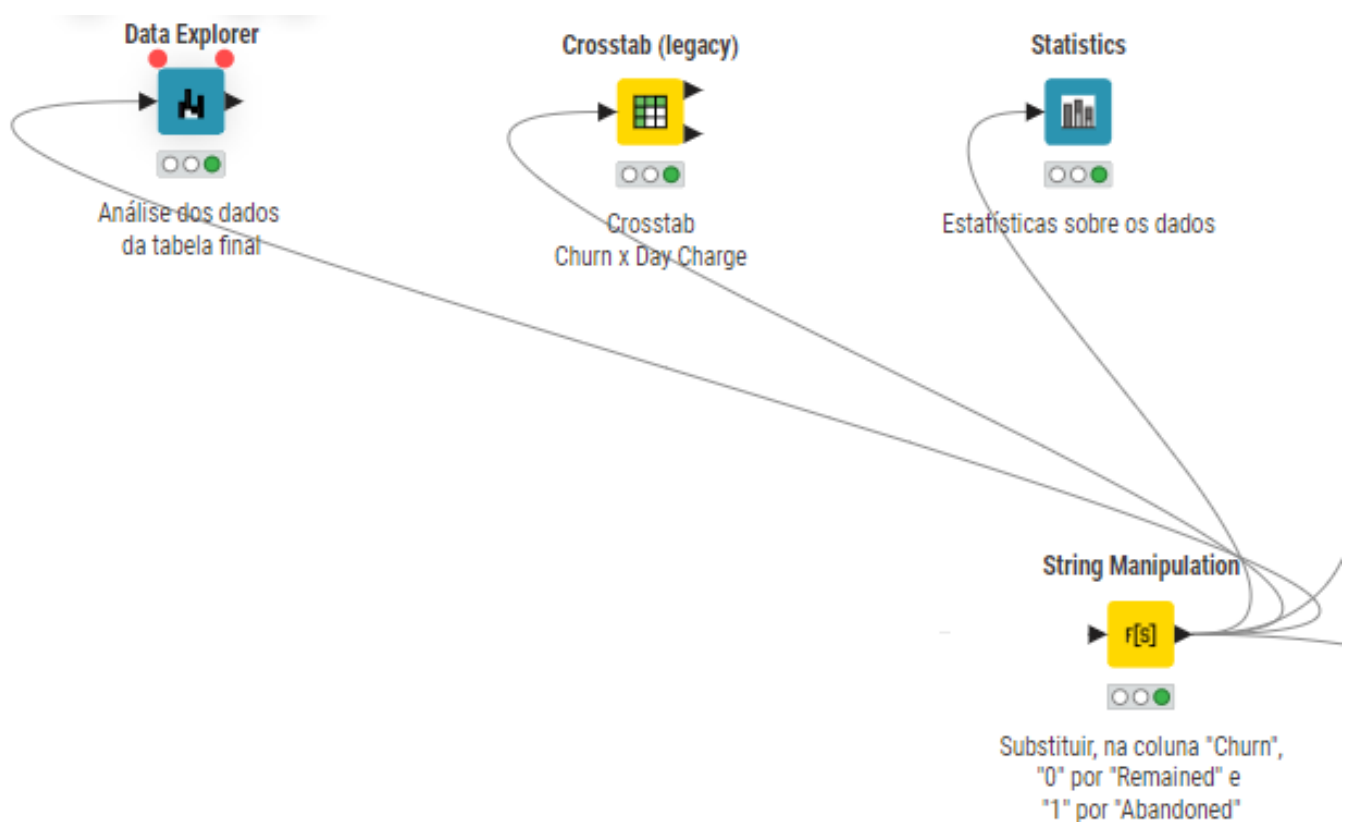


Figura 2.4: *Workflow* completo da Tarefa 2



Figura 2.5: Análise da distribuição do atributo *Churn* (2850 clientes ficam, 483 pessoas abandonam)

Statistics

Rows: 23 | Columns: 23

Name	Type	# Missing v...	# Unique val...	Minimum	Maximum	1% Quantile	5% Quantile	10% Quantile	25% Quantile	50% Quantil...	75% Quantile
VMail Messa...	Number (inte...	0	46	0	51	0	0	0	0	0	20
Day Mins	Number (dou...	0	1667	0	350.8	51.602	89.77	110.3	143.65	179.4	216.45
Eve Mins	Number (dou...	0	1611	0	363.7	79.036	118.77	136.7	166.6	201.4	235.3
Night Mins	Number (dou...	0	1591	23.2	395	78.508	118	136.3	167	201.2	235.3
Intl Mins	Number (dou...	0	162	0	20	3.3	5.7	6.7	8.5	10.3	12.1
CustServ Calls	Number (inte...	0	10	0	9	0	0	0	1	1	2
Day Calls	Number (inte...	0	119	0	165	54	67	74	87	101	114
Day Charge	Number (dou...	0	1667	0	59.64	8.777	15.264	18.75	24.42	30.5	36.8
Eve Calls	Number (inte...	0	123	0	170	53	67	75	87	100	114
Eve Charge	Number (dou...	0	1440	0	30.91	6.72	10.097	11.62	14.16	17.12	20
Night Calls	Number (inte...	0	120	33	175	56.34	68	75	87	100	113
Night Charge	Number (dou...	0	933	1.04	17.77	3.53	5.31	6.13	7.52	9.05	10.59
Intl Calls	Number (inte...	0	21	0	20	1	1	2	3	4	6
Intl Charge	Number (dou...	0	162	0	5.4	0.89	1.54	1.81	2.3	2.78	3.27
Area Code	Number (inte...	0	3	408	510	408	408	408	408	415	510
Phone	String	0	3333	📞	📞	📞	📞	📞	📞	📞	📞
Account Leng...	Number (inte...	0	212	1	243	12	35	50	74	101	127
Churn	String	0	2	📞	📞	📞	📞	📞	📞	📞	📞
Int'l Plan	Number (inte...	0	2	0	1	0	0	0	0	0	0
VMail Plan	Number (inte...	0	2	0	1	0	0	0	0	0	1
State	String	0	51	📞	📞	📞	📞	📞	📞	📞	📞

Figura 2.6: Nodo *statistics* (nota que não há *Missing values*)

Relativamente ao nodo *Crosstab*, criei uma tabela *Churn x Day Charge* onde é possível observar que todas os clientes que têm o parâmetro *Day Charge* igual ou superior a **54.03** acabaram todas por **abandonar** a empresa. Já os clientes com este mesmo parâmetro igual ou inferior a **7.65** acabaram por **ficar** na empresa.

54.03	54.59	54.62	54.67	54.79	54.81	54.83	55.2	55.47	55.51	55.78	56.07	56.59	56.83	57.04	57.36	58.7	58.96	59.64	Total
1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	483
																			2 850
1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3 333

Figura 2.7: *Day Charge* ≥ 54.03 (todas os clientes abandonaram a empresa)

Cross Tabulation of Churn by Day Charge

Frequency	0.0	0.44	1.33	1.34	2.13	2.99	3.21	3.32	4.4	4.59	5.08	5.25	5.78	5.97	6.41	6.43	6.72	6.87	6.95	7.12	7.63	7.65
Abandoned	1																					
Remained	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Total	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figura 2.8: *Day Charge* ≤ 7.65 (clientes ficaram na empresa)

2.3 Tarefa 3

- Particionar os dados de forma estratificada (pela *feature* "*Churn*"), utilizando 70% para aprendizagem e 30% para teste. Aplicar um *Decision Tree Learner* e um *Decision Tree Predictor*. Avaliar a precisão (*accuracy*) do modelo e a respetiva matriz de confusão.

Para particionar os dados, utilizei o nodo *Partitioning* configurado com vista a que a partição seja feita de forma estratificada pelo atributo "*Churn*" e que, 70% dos dados sejam usados para treino do modelo e, os restantes 30%, para teste. Além disto, utilizei o nodo *Scorer* para obter a matriz de confusão do modelo, bem como a precisão (*accuracy*) do mesmo.

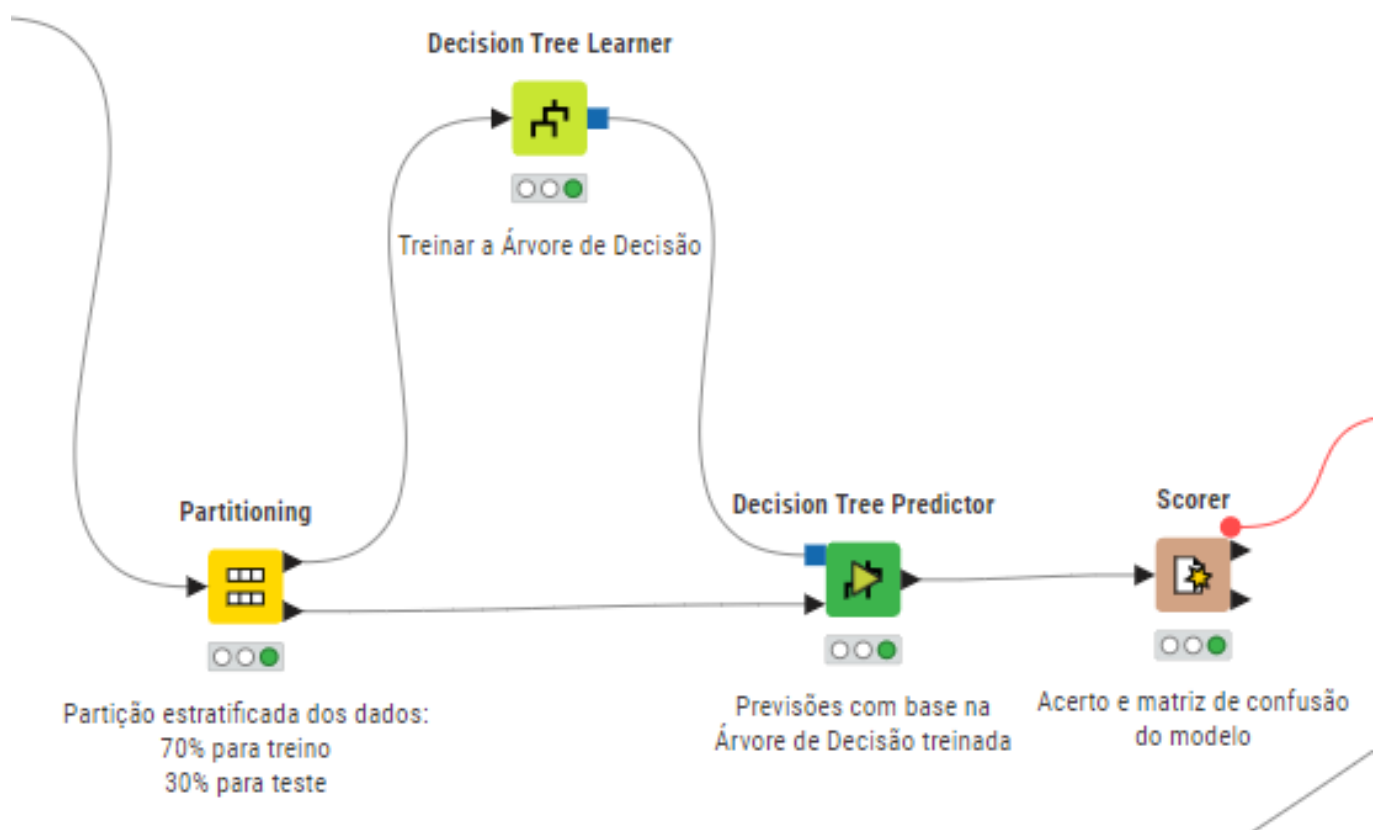


Figura 2.9: *Workflow* da Tarefa 3

First partition	Flow Variables	Job Manager Selection	Memory Policy
Choose size of first partition			
<input type="radio"/> Absolute		<input type="text" value="100"/>	
<input checked="" type="radio"/> Relative[%]		<input type="text" value="70"/>	
<input type="radio"/> Take from top			
<input type="radio"/> Linear sampling			
<input type="radio"/> Draw randomly			
<input checked="" type="radio"/> Stratified sampling		<input type="text" value="S"/> Churn	
<input type="checkbox"/> Use random seed		<input type="text" value="1 696 349 744 4"/>	

Figura 2.10: Partição estratificada dos dados (70% para treino e 30% para teste)

Options PMMLSettings Flow Variables Job Manager Selection

General

Class column **S** Churn ▾

Quality measure Gini index ▾

Pruning method No pruning ▾

☒ Reduced Error Pruning

Min number records per node 2 ▴ ▾

Number records to store for view 10 000 ▴ ▾

☒ Average split point

Number threads 8 ▴ ▾

☐ Skip nominal columns without domain information

Root split

☐ Force root split column

Root split column **D** Day Mins ▾

Binary nominal splits

☐ Binary nominal splits

Max #nominal 10 ▴ ▾

☐ Filter invalid attribute values in child nodes

Figura 2.11: Configuração do nodo *Decision Tree Learner*

Churn \ Pr...	Remained	Abandoned
Remained	804	25
Abandoned	63	71

Correct classified: 875	Wrong classified: 88
Accuracy: 90,862%	Error: 9,138%
Cohen's kappa (κ): 0,567%	

Figura 2.12: Avaliação do modelo através do nodo *Scorer*

2.4 Tarefa 4

- Remover, iterativamente, *features* do *dataset* e reavaliar a performance dos modelos candidatos. Descrever os resultados obtidos.

Para remover, de forma iterativa, *features* do *dataset* para posterior análise dos modelos candidatos, fiz uso de um *loop* denominado **Feature Selection Loop** composto por dois nodos distintos: **Feature Selection Loop Start** e **Feature Selection Loop End**. Na configuração do primeiro, garanti que o atributo *Churn* era um atributo estático, isto é, que nunca poderá ser removido visto ser a classe do nosso modelo (aquilo que queremos prever). Relativamente à estratégia de seleção de atributos, escolhi a opção **Backward Feature Elimination**. Por fim, utilizei o nodo **Feature Selection Filter** para avaliar todos os modelos candidatos, selecionando o modelo com melhor *accuracy*.

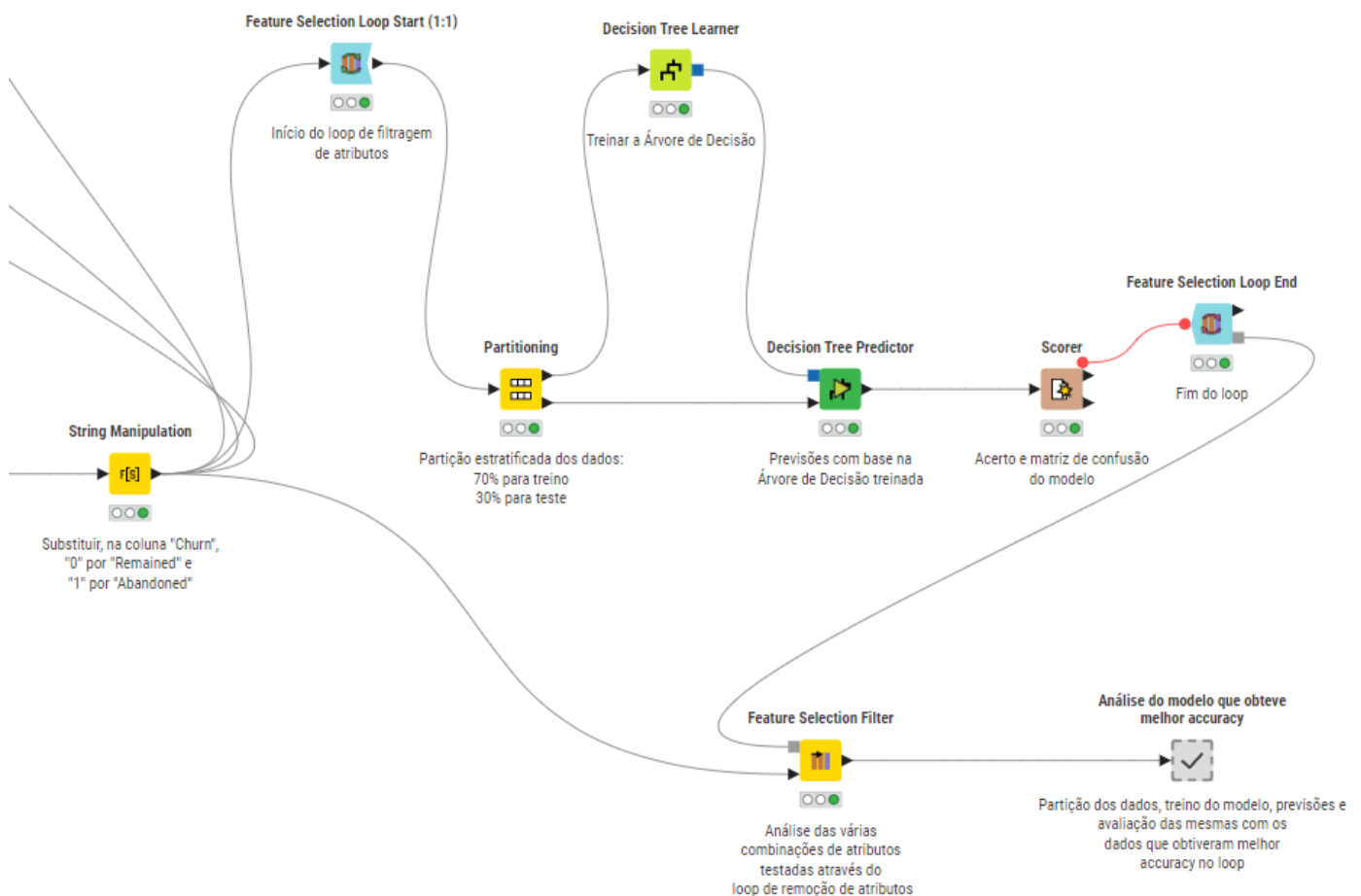


Figura 2.13: *Workflow* da Tarefa 4

The list on the left contains 'static' columns such as the target column.
The columns to choose from need to be in the list on the right.

☒ Manual Selection ☐ Wildcard/Regex Selection

Static Columns

Filter

S Churn

☒ Enforce exclusion

Variable Columns ('Features')

Filter

I VMail Message
D Day Mins
D Eve Mins
D Night Mins
D Intl Mins
I CustServ Calls
I Day Calls
D Day Charge
I Eve Calls
D Eve Charge
I Night Calls

☐ Enforce inclusion

> >> < <<

Feature selection strategy: Backward Feature Elimination

Sequential Algorithm Settings

☐ Use threshold for number of features

Select threshold for number of features: 20

Figura 2.14: Configuração do loop *Feature Selection Loop*

☒ Include static columns

☐ Select features manually

☒ Select best score

☐ Select features automatically by score threshold

Prediction score threshold: 0

Optimization Criterion: The score is being maximized.

Accuracy	Nr. of features	
0,951	14	I VMail Message
0,945	19	D Day Mins
0,944	12	D Eve Mins
0,943	10	D Night Mins
0,942	20	D Intl Mins
0,941	18	I CustServ Calls
0,941	15	I Day Calls
0,941	11	D Day Charge
0,94	17	I Eve Calls
0,94	13	D Eve Charge
0,938	9	I Night Calls
0,937	21	D Night Charge
0,936	16	I Intl Calls
0,934	7	D Intl Charge
0,931	8	I Area Code
0,911	6	S Phone
0,906	22	I Account Length
0,901	5	S Churn
0,886	3	I Int'l Plan
0,884	4	I VMail Plan
0,864	1	S State
0,861	2	I Area Code (right)
		S Phone (right)

Figura 2.15: Análise dos modelos candidatos e seleção do modelo com melhor *accuracy*

- Através desta análise dos modelos candidatos, percebemos que o modelo com melhor *accuracy* obteve um percentagem de acerto de **95,1%** a usar **14** atributos (selecionados a azul).

2.5 Tarefa 5

- Seguir as práticas de bons-hábitos na construção de *workflows*.

No que toca a práticas de bons-hábitos na construção de *workflows*, utilizei denominações diferentes para cada nodo, bem como a criação de meta-nodos com uma breve descrição do que está a ser feito em cada meta-nodo.

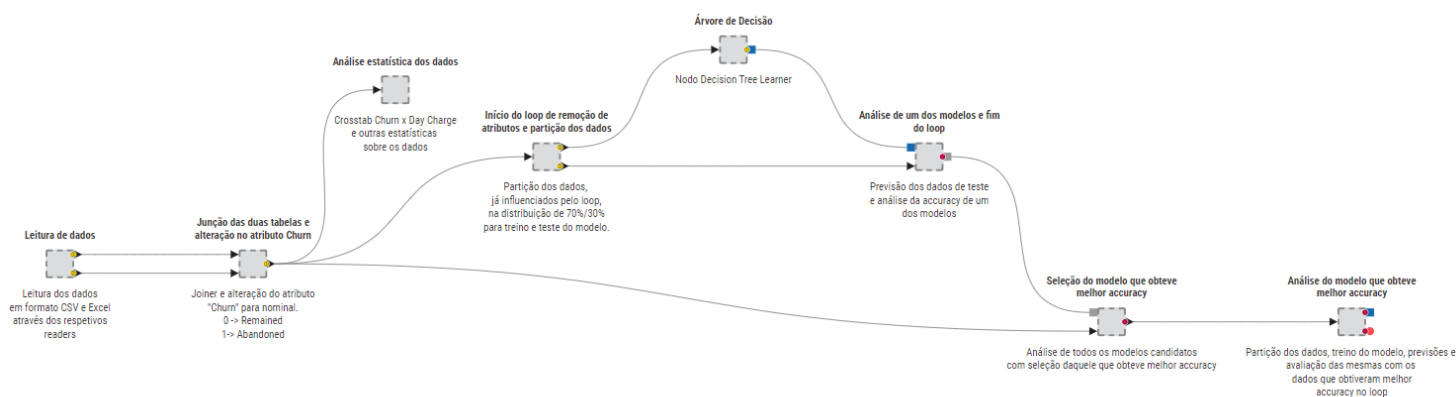


Figura 2.16: *Workflow* completo com uso de meta-nodos e de anotações

2.6 Tarefa 6

- Utilizar o output de um nodo *Decision Tree Learner* para criar uma imagem de uma Árvore de Decisão e guardar essa imagem no ambiente de trabalho.

Para criar uma imagem de uma Árvore de Decisão, utilizei o nodo *Decision Tree Image* conectado ao output do nodo *Decision Tree Learner* com vista a obter uma imagem da árvore de decisão. Obtida essa imagem, guardei a mesma no ambiente de trabalho através do nodo *Image Writer (Port)*.

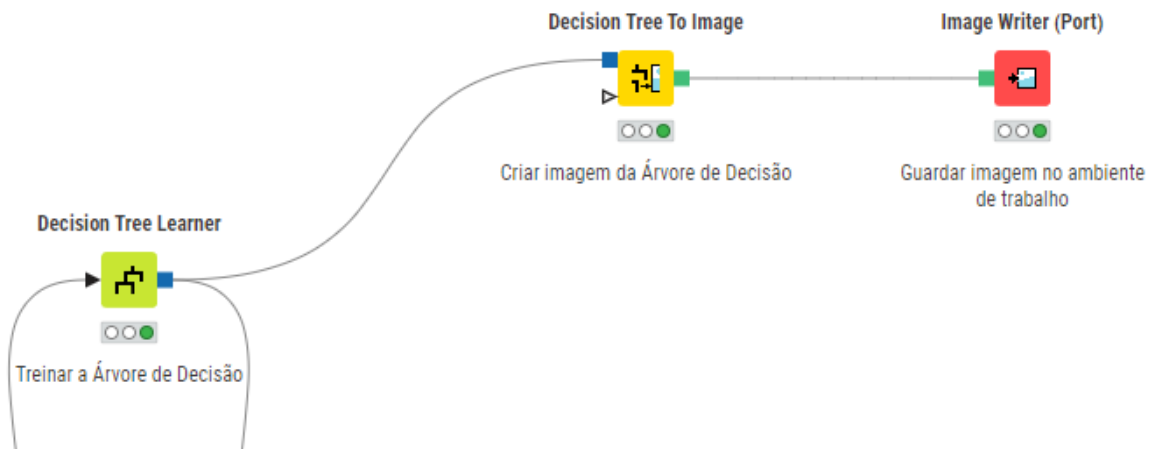


Figura 2.17: *Workflow* da Tarefa 6

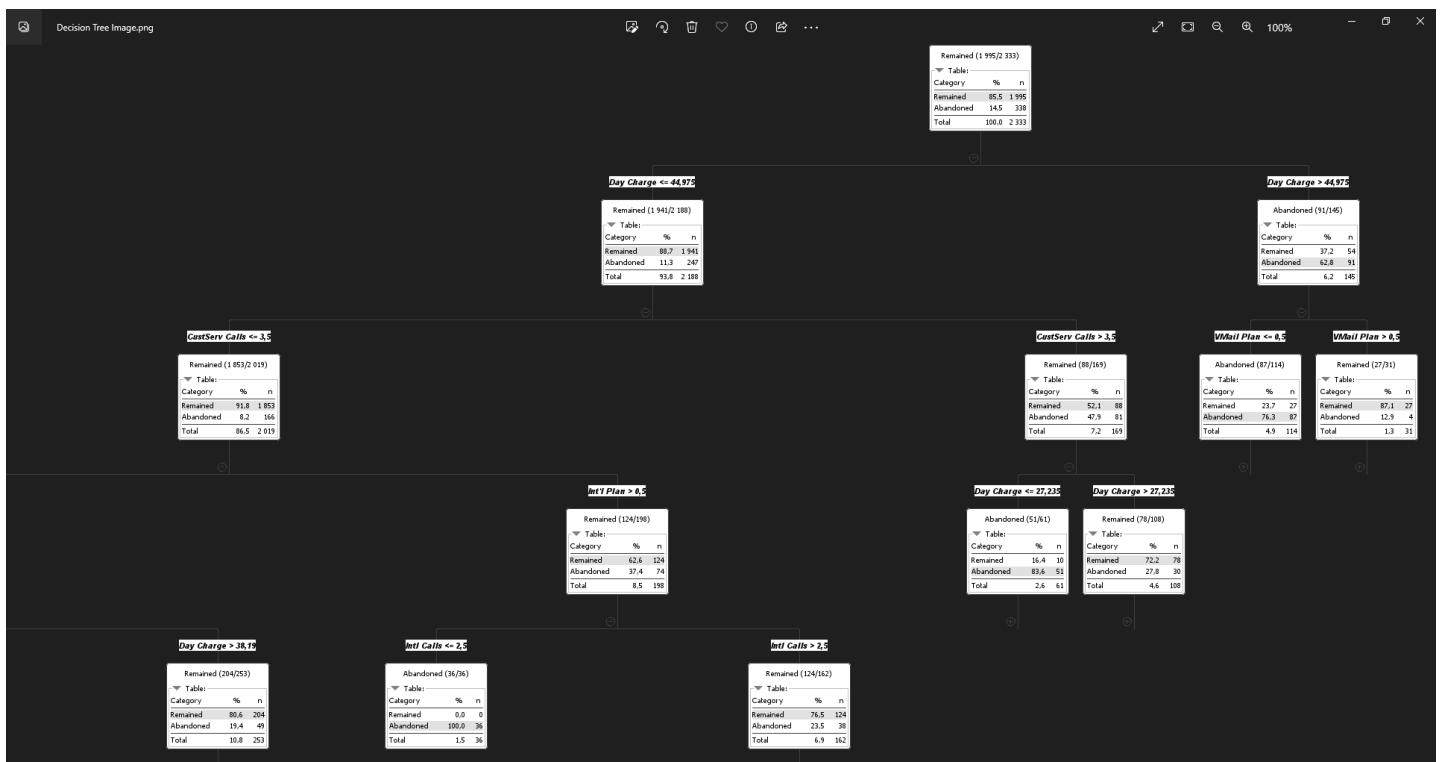


Figura 2.18: Imagem da Árvore de Decisão guardada no *desktop*

- Esta imagem ilustra apenas parte da Árvore de Decisão, sendo possível no ficheiro guardado no *desktop* e enviado no trabalho, vaguear por toda a árvore e fazer zoom na mesma.

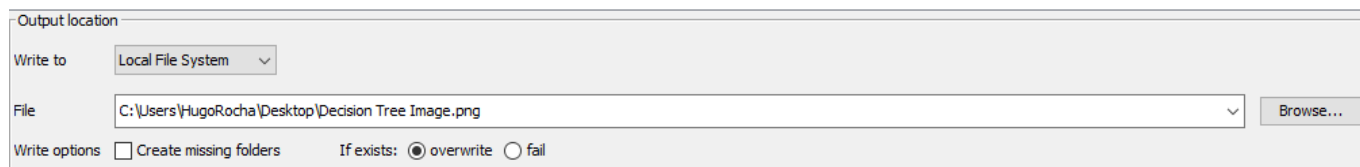


Figura 2.19: Configuração do nodo *Image Writer (Port)*

2.7 Workflow completo (sem usar a maioria dos meta-nodos)

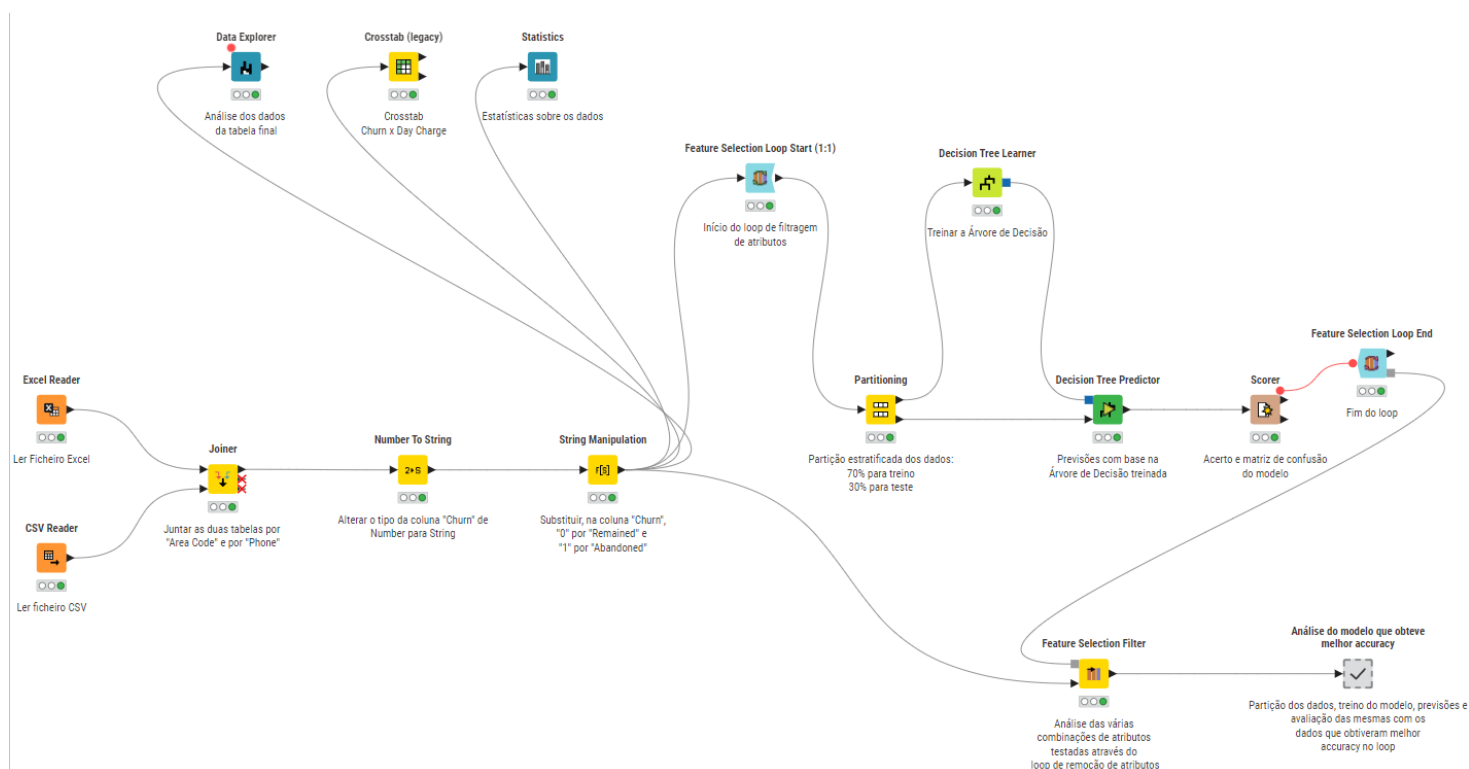


Figura 2.20: *Workflow* completo

- O único meta-nodo apresentado nesta imagem, corresponde a uma análise opcional que fiz sobre os dados que obtiveram melhor *accuracy* no loop, isto é, optei por correr várias vezes um modelo com diferentes partições desse *dataset* e obtive sempre uma *accuracy* igual ou superior a 92%. Destacar também que o modelo nunca prevê apenas **Remained**, o que torna o modelo útil.

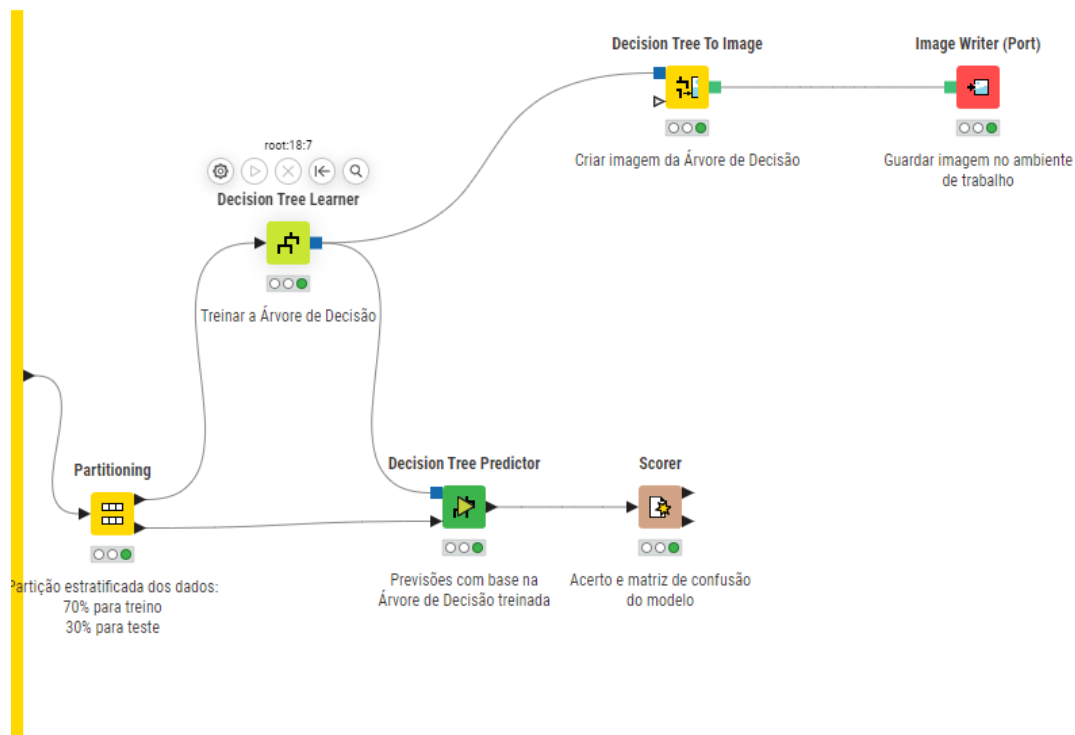


Figura 2.21: *Workflow* do último meta-nodo

Predictio...
String
Remained
Remained
Remained
Abandoned
Abandoned
Remained
Remained
Remained
Remained
Remained
Remained
Abandoned
Remained
Remained
Remained
Remained

Figura 2.22: Algumas previsões do modelo