

Universidade do Minho

Licenciatura em Engenharia Informática

Aprendizagem e Decisão Inteligentes

Conceção de modelos de aprendizagem

Grupo 12

Ana Murta (A93284)
Ana Henriques (A93268)
Rui Coelho (A58898)
Tiago Carneiro (A93207)

maio, 2022

Conteúdo

1	Introdução	7
2	<i>Music genre</i>	8
2.1	<i>Dataset</i>	8
2.2	<i>KNIME workflow</i>	9
2.3	Análise preliminar dos dados	10
2.4	Pré-Processamento	12
2.5	Análise dos dados após o pré-processamento	14
2.6	Validação dos modelos de <i>Machine Learning</i>	15
2.7	Modelos de <i>Machine Learning</i>	16
2.8	Análise dos modelos de <i>Machine Learning</i>	22
3	<i>Salary Classification</i>	23
3.1	<i>Dataset</i>	23
3.2	<i>KNIME workflow</i>	24
3.3	Análise preliminar dos dados	25
3.4	Pré-Processamento	27
3.5	Análise dos dados após o pré-processamento	28
3.6	Modelos de <i>Machine Learning</i>	31
3.7	Análise dos modelos de <i>Machine Learning</i>	35
4	Conclusão	36
A	<i>Music_genre</i>	37

Lista de Figuras

2.1	Excerto do <i>dataset music_genre</i>	9
2.2	<i>KNIME</i> : <i>workflow</i>	10
2.3	<i>KNIME</i> : exploração inicial dos dados.	11
2.4	<i>KNIME</i> : <i>Data Explorer</i>	11
2.5	<i>KNIME</i> : <i>Box Plot</i>	12
2.6	<i>KNIME</i> : <i>Line Plot</i>	12
2.7	<i>KNIME</i> : <i>Linear Correlation</i>	12
2.8	<i>KNIME</i> : <i>Rank Correlation</i>	12
2.9	<i>KNIME</i> : nodos de pré-processamento de dados.	13
2.10	<i>KNIME</i> : <i>Box Plot</i>	14
2.11	<i>KNIME</i> : <i>Line Plot</i>	14
2.12	<i>KNIME</i> : <i>Linear Correlation</i>	15
2.13	<i>KNIME</i> : <i>Rank Correlation</i>	15
2.14	<i>KNIME</i> : <i>Partitioning</i>	16
2.15	<i>KNIME</i> : <i>X-Partitioner</i>	16
2.16	<i>KNIME</i> : Modelos de ML.	16
2.17	ML: Modelo 1.	17
2.18	ML: Modelo 2.	18
2.19	ML: Modelo 3.	18
2.20	ML: Modelo 4.	19
2.21	ML: Modelo 5.	20
2.22	ML: Modelo 6.	21

2.23	ML: Modelo 7	21
2.24	<i>KNIME: Backward Feature Elimination</i>	22
2.25	<i>KNIME: Forward Feature Selection</i>	22
3.1	<i>KNIME: Estrutura do ficheiro salary-classification.csv</i>	24
3.2	<i>KNIME: workflow</i>	24
3.3	<i>KNIME: Reconhecimento de strings ? como missing values</i>	25
3.4	<i>KNIME: análise inicial dos dados</i>	25
3.5	<i>KNIME: Output de Data Explorer para valores numéricos</i>	26
3.6	<i>KNIME: Output de Data Explorer para valores nominais</i>	26
3.7	<i>KNIME: Output de Box Plot</i>	26
3.8	<i>KNIME: Output de Rank Correlation</i>	27
3.9	<i>KNIME: Output de Linear Correlation</i>	27
3.10	<i>KNIME: pré-processamento dos dados do dataset</i>	27
3.11	<i>KNIME: Análise posterior dos dados</i>	29
3.12	<i>KNIME: Output de Data Explorer para valores numéricos</i>	29
3.13	<i>KNIME: Output de Data Explorer para valores nominais</i>	29
3.14	<i>KNIME: Output de Box Plot</i>	30
3.15	<i>KNIME: Output de Rank Correlation</i>	30
3.16	<i>KNIME: Output de Linear Correlation</i>	30
3.17	<i>KNIME: Modelos de Machine Learning</i>	31
3.18	<i>KNIME: Modelo 1 – Árvore de Decisão</i>	32
3.19	<i>KNIME: Modelo 2 – Random Forest com Feature Selection</i>	32
3.20	<i>KNIME: Backward Feature Elimination</i>	33
3.21	<i>KNIME: Forward Feature Selection</i>	33
3.22	<i>KNIME: Modelo 3 – Regressão Logística</i>	34
3.23	<i>KNIME: Modelo 4 – Clustering</i>	34
3.24	<i>KNIME: Script do Rule Engine</i>	35
A.1	<i>Modelo 1: Scorer para hold-out validation</i>	37

A.2	Modelo 1: <i>Scorer para cross-validation.</i>	38
A.3	Modelo 2: <i>Scorer hold-out validation.</i>	38
A.4	Modelo 2: <i>Scorer para cross-validation.</i>	39
A.5	Modelo 3: <i>Scorer hold-out validation.</i>	39
A.6	Modelo 3: <i>Scorer para cross-validation.</i>	40
A.7	Modelo 4: <i>Scorer hold-out validation.</i>	40
A.8	Modelo 4: <i>Scorer para cross-validation.</i>	41
A.9	Modelo 5: <i>Scorer hold-out validation.</i>	41
A.10	Modelo 5: <i>Scorer para cross-validation.</i>	42
A.11	Modelo 6: <i>Scorer hold-out validation.</i>	42
A.12	Modelo 6: <i>Scorer para cross-validation.</i>	43
A.13	Modelo 7: <i>Scorer hold-out validation (Backward Feature Elimination).</i>	43
A.14	Modelo 7: <i>Scorer hold-out validation (Forward Feature Selection).</i>	44
A.15	Modelo 7: <i>Scorer cross-validation (Backward Feature Elimination).</i>	44
A.16	Modelo 7: <i>Scorer cross-validation (Forward Feature Selection).</i>	45
B.1	Modelo 1: <i>Scorer para hold-out validation.</i>	46
B.2	Modelo 1: <i>Scorer para cross validation.</i>	46
B.3	Modelo 2: <i>Scorer para hold-out validation – Forward Feature Selection.</i>	47
B.4	Modelo 2: <i>Scorer para hold-out validation – Backward Feature Elimination.</i>	47
B.5	Modelo 2: <i>Scorer para cross validation – Forward Feature Selection.</i>	47
B.6	Modelo 2: <i>Scorer para cross validation – Backward Feature Elimination.</i>	48
B.7	Modelo 3: <i>Scorer para hold-out validation.</i>	48
B.8	Modelo 3: <i>Scorer para cross validation.</i>	48
B.9	Modelo 4: <i>Scorer para hold-out validation.</i>	49
B.10	Modelo 4: <i>Scorer para cross validation.</i>	49

Capítulo 1

Introdução

O presente relatório visa acompanhar o desenvolvimento do trabalho prático elaborado ao longo do semestre. Como tal, contempla dois grandes grupos: um grupo exclusivamente dedicado ao *dataset* fornecido pela equipa docente – *Salary classification* – e outro grupo que confere foco ao *dataset* escolhido pelo grupo de trabalho – *music genre*. Para a realização do trabalho foi utilizada a plataforma *KNIME*.

Ambas as seções do relatório pretendem demonstrar e explicar o trabalho conduzido, apresentando os *datasets*, o trabalho elaborado sobre os mesmos, e os modelos de *Machine Learning* (ML) desenvolvidos – fornecendo *insight* para as decisões adotadas pelo grupo.

O conjunto de dados referente ao género musical¹ foi obtido através da plataforma *Kaggle*. Após a pesquisa de vários conjuntos de dados para a elaboração do trabalho, o grupo selecionou o *dataset* acima referido devido ao interesse demonstrado pelo tópico, i.e., por se centrar em música, e pela informação disponível acerca das *features* que integram o *dataset*.

¹c.f. <https://www.kaggle.com/code/braincl/music-genre-data-preprocessing-predictions-model/data>

Capítulo 2

Music genre

2.1 *Dataset*

O conjunto de dados selecionado pelo grupo de trabalho consiste num *dataset* com 18 *features*, i.e., colunas, e 50006 entradas. A primeira entrada do ficheiro *CSV* corresponde à nomenclatura das diversas variáveis e as restantes representam os dados – onde cada linha corresponde a dados sobre uma única música. O conjunto de dados selecionado tem como *target* o género musical, ou seja, os dados podem ser usados para a previsão do estilo musical das músicas.

A seguinte listagem apresenta as *features* presentes no conjunto de dados, descrevendo de um modo sucinto o seu significado, assim como o tipo de dados usados para a sua representação:

- *instance_id*: identificador único de cada música (atributo qualitativo nominal, representado sob a forma de um *double*);
- *artist_name*: nome do artista (atributo qualitativo nominal, representado sob a forma de uma *string*);
- *track_name*: nome da música (atributo qualitativo nominal, representado sob a forma de uma *string*);
- *popularity*: nível de popularidade, medido entre 0 e 99 (atributo quantitativo intervalar discreto, representado sob a forma de uma *string*);
- *acousticness*: medida de confiança, entre 0.0 e 1.0, de que uma dada música é acústica (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *danceability*: medida de confiança, entre 0.0 1.0, de que uma música é adequada para dançar (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *duration_ms*: duração em milisegundos (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *energy*: medida perceptual, entre 0.0 1.0, de intensidade e atividade (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *instrumentalness*: medida, entre 0.0 1.0, de presença de voz numa música (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);

- *key*: estimativa da nota musical mais frequente na música (atributo qualitativo categórico *string*);
- *liveness*: medida de confiança, entre 0.0 e 1.0, de deteção da presença de público aquando da gravação da música (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *loudness*: medida geral da sonoridade em Decibel (dB) da música (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *mode*: modalidade da música (*major* ou *minor*) (atributo qualitativo categórico, representado sob a forma de uma *string*);
- *speechiness*: medida de confiança, entre 0.0 e 1.0, da deteção da presença de palavras faladas numa música (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *tempo*: estimativa do tempo de uma música, medido em batimentos por minuto (BPM) (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *obtained_date*: data de recolha da música (atributo qualitativo categórico, representado sob a forma de uma *string*);
- *valence*: medida, entre 0.0 e 1.0, da positividade associada a cada música (atributo quantitativo intervalar contínuo, representado sob a forma de um *double*);
- *music_genre*: género musical (atributo qualitativo categórico, representado sob a forma de uma *string*).

O carregamento de dados para a plataforma *KNIME* foi efetuado com recurso ao nodo *CSV Reader*. A Figura 2.1 apresenta um excerto do *dataset*, com dados para as diversas *features* anteriormente descritas.

instance_id	artist_name	track_name	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	obtained_date	valence	music_genre	
32894.0	Röyksopp	Röyksopp's Night Out	27.0	0.00468	0.652	-1.0	0.941	0.792	A#	0.115	-5.2010000000000005	Minor	0.0748	100.889	4-Apr-0759	Electronic		
46652.0	Theivery Corporation	The Shining Path	31.0	0.0127	0.622	218293.0	0.89	0.95	D	0.124	-7.042999999999999	Minor	0.03	115.00280000000001	4-Apr-0531	Electronic		
30097.0	Dillon Francis	Hurricane	28.0	0.00306	0.62	215613.0	0.755	0.018	G#	0.534	-4.617	Major	0.0345	127.994	4-Apr-0329999999999999	Electronic		
62177.0	Upload	Nitro	34.0	0.0254	0.774	166875.0	0.70	0.0253	C#	0.157	-4.498	Major	0.239	128.014	4-Apr-027	Electronic		
24967.0	What So Not	Divide & Conquer	32.0	0.00465	0.638	222369.0	0.5870000000000001	0.909	F#	0.157	-6.266	Major	0.0413	145.036	4-Apr-0322999999999999	Electronic		
89064.0	Axel Boman	Hello	47.0	0.005229999999999999	0.755	519468.0	0.731	0.8540000000000001	B	0.216	-10.517	Minor	0.0412	7.4-Apr-014	Electronic			
43760.0	Jordan Comilli	Clash	46.0	0.0289	0.5720000000000001	214408.0	0.80	0.8299999999999999	7	0.746-06	B	0.106	-4.294	Major	0.351	149.995	4-Apr-023	Electronic
30738.0	Hraach	Delirio	43.0	0.0297	0.809	416132.0	0.705	0.903	G	0.0635	-9.339	Minor	0.0484	128.008	4-Apr-0760999999999999	Electronic		
84950.0	Kayzo	NEVER ALONE	19.0	0.00299	0.509	292800.0	0.921	0.000276	F	0.179	-3.175	Minor	0.268	149.94799999999998	4-Apr-0.273	Electronic		
56590.0	Shlump	Lazer Beam	22.0	0.00934	0.578	204800.0	0.731	0.0112	A	0.111	-7.091	Minor	0.179	139.533	4-Apr-0.203	Electronic		
49030.0	Chase & Status	Lost & Not Found - Acoustic	30.0	0.00855	0.607	178463.0	0.158	0.0	F#	0.106	-13.787	Minor	0.0345	57.528	4-Apr-0.307	Electronic		
22654.0	G Jones	Mind	27.0	0.00377	0.513	165132.0	0.628	0.569	B	0.109	-5.439	Minor	0.0609	178.543	3-Apr-0.0591	Electronic		
69056.0	Champagne Drip	Satellite (Feat. Len X)	31.0	0.016	0.66	236089.0	0.892	0.000685	C	0.163	-3.464	Major	0.0645	128.043	4-Apr-0.111	Electronic		
62039.0	DJ Shadow	Broken Levee Blues	31.0	0.086	0.737	-1.0	0.405	0.0361	A	0.173	-10.536	Minor	0.0424	154.745	4-Apr-0.647	Electronic		
49205.0	Getter	Bonesaw	27.0	0.000194	0.681	243857.0	0.969	0.7759999999999999	D	0.409	-1.5530000000000002	Major	0.185	139.911	4-Apr-0.198	Electronic		

Figura 2.1: Exerto do *dataset music_genre*.

2.2 KNIME workflow

O trabalho desenvolvido, na plataforma *KNIME*, para o *dataset music_genre* encontra-se apresentado na Figura 2.2, que se encontra dividido em diversas seções:

- Leitura de dados;
- Estudos de correlação;
- Estatística descritiva;
- Pré-processamento de dados;
- Validação e geração de modelos de ML.

Esta organização procurou refletir os principais passos a executar aquando da geração de modelos de ML.



Figura 2.2: *KNIME: workflow*.

Assim sendo, o primeiro passo consiste na leitura do *dataset*, com recurso ao nodo *CSV Reader* do *KNIME*.

2.3 Análise preliminar dos dados

O trabalho com este *dataset* segue-se com a elaboração de um estudo geral do estado inicial dos dados, de modo a determinar o tratamento necessário a conduzir para a utilização dos dados na criação de modelos de aprendizagem automática. Para conduzir esta análise inicial, foram utilizadas as seções referentes aos estudos de correlação e à análise estatística descritiva. A Figura 2.3 apresenta a expansão dos metanodos presentes nas referidas seções.

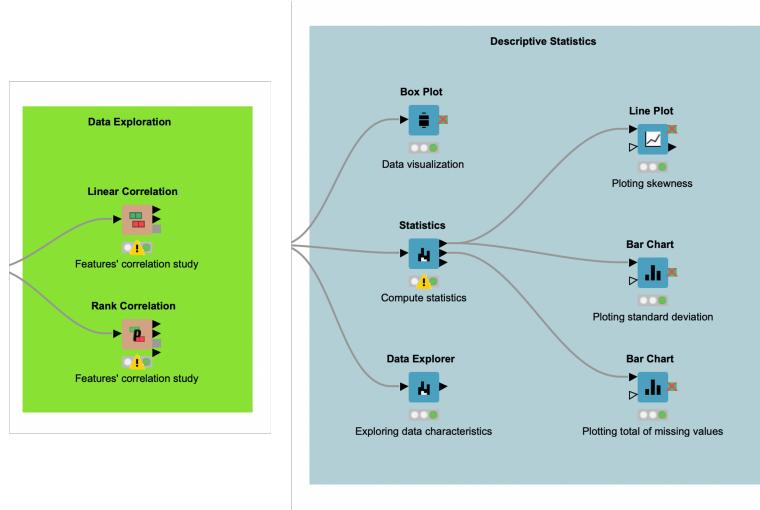


Figura 2.3: *KNIME*: exploração inicial dos dados.

A exploração estatística efetuada permitiu conhecer, de um modo geral, o conteúdo das *features* que integram o conjunto de dados anteriormente apresentado. A Figura 2.4 apresenta o *output* do nodo *Data Explorer*, para os dados numéricos. Este nodo permitiu obter informação acerca dos valores extremos das diversas *features*, da sua média e desvio padrão e, ainda, a presença, ou não, de valores omissos – valores que devem ser tratados aquando da fase de pré-processamento. A análise, do mesmo nodo, refrente às variáveis nominais revelou também a presença de *missing values* para algumas colunas. A Figura 2.5 oferece uma representação gráfica da distribuição dos dados das *features* numéricas.

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness
instance_id	☐	20002	91759	55888.396	20725.256	429536246.747	-0.002
popularity	☐	0	99	44.220	15.542	241.554	-0.305
acousticness	☐	0	0.996	0.306	0.341	0.117	0.882
danceability	☐	0.060	0.986	0.558	0.179	0.032	-0.300
duration_ms	☐	-1	4830606	221252.603	128671.957	16556472558.622	4.265
energy	☐	0.001	0.999	0.600	0.265	0.070	-0.570
instrumentalness	☐	0	0.996	0.182	0.325	0.106	1.487
liveness	☐	0.010	1	0.194	0.162	0.026	2.249
loudness	☐	-47.046	3.744	-9.134	6.163	37.982	-1.871
speechiness	☐	0.022	0.942	0.094	0.101	0.010	2.475
valence	☐	0	0.992	0.456	0.247	0.061	0.132

Figura 2.4: *KNIME*: *Data Explorer*.

Adicionalmente, pode salientar-se a medida de *skewness*, que descreve a assimetria observada nos dados face ao seu valor médio. As colunas com um valor de assimetria distante de 0 apresentam uma tendência, positiva ou negativa, na distribuição dos seus valores face à média – podendo estas *features* ser alvo de tratamento prévio para a criação de modelos de ML. A Figura 2.6 corresponde à representação gráfica da medida de *skewness*.

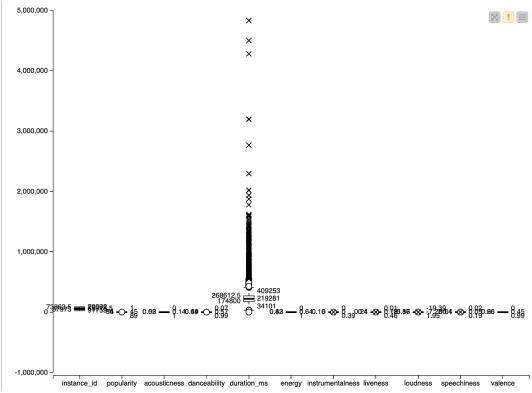


Figura 2.5: KNIME: Box Plot.

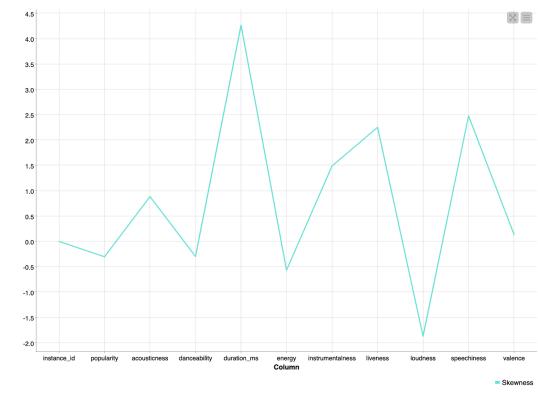


Figura 2.6: KNIME: Line Plot.

A análise de correlação apresentada nas Figuras 2.7 2.8 permite medir a força e direção da associação entre duas variáveis – o que pode fornecer informação útil acerca das *features* que devem incorporar os modelos de aprendizagem automática. De um modo geral, pode observar-se que existem *features* com uma um grau elevado de correlação, como o caso do par *energy-acousticness* e do par *energy-loudness*. Isto pode ser indicativo que a informação fornecida por estas *features* pode ser

Focando na variável *target*, *music_genre*, pode observar-se que existe uma correlação positiva com as colunas *popularity* e *danceability* e uma correlação negativa com a coluna *instrumentalness*.

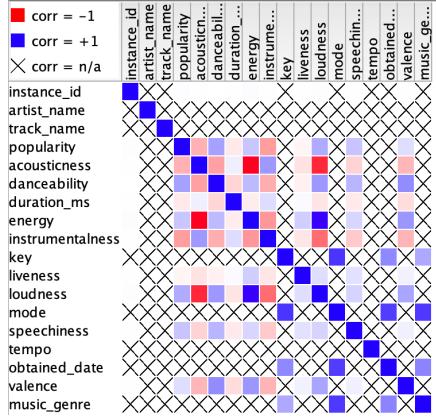


Figura 2.7: KNIME: Linear Correlation.

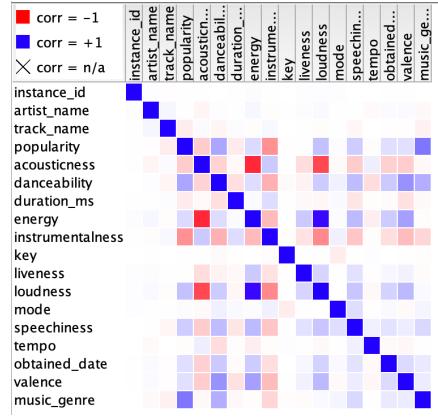


Figura 2.8: KNIME: Rank Correlation.

2.4 Pré-Processamento

O super-nodo *Pre-Processing*, inserido na seção com o mesmo nome, contempla o trabalho desenvolvido em termos de tratamento e limpeza inicial do *dataset*. A preparação de um *dataset* para a criação de modelos de ML não é um processo com solução única, uma vez que alguns modelos de aprendizagem automática podem impor a implementação de determinados passos de tratamento de dados que podem ser supérfluos para outros modelos. Assim sendo, o foco deste tratamento inicial passa por efetuar um conjunto de operações transversais a vários modelos de ML – sendo, numa fase posterior, efetuada uma preparação adicional dos dados, caso tal

seja necessário. A Figura 2.9 demonstra a expansão do metanodo acima referido, com as diversas operações efetuadas em termos de tratamento e pré-processamento de dados do *music_genre*.

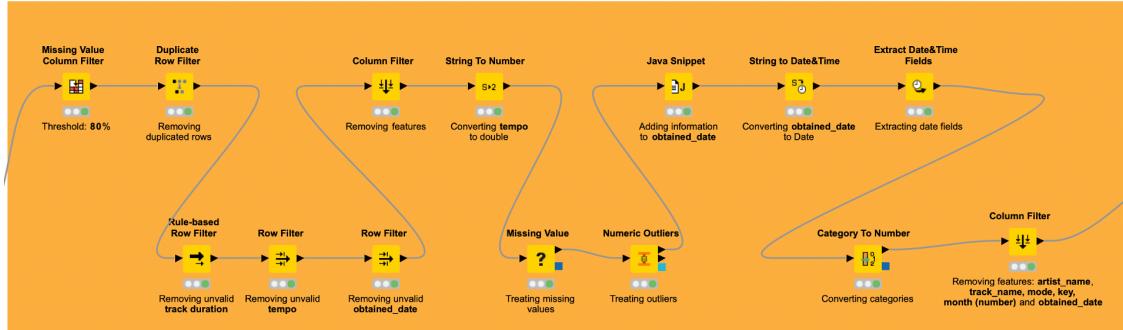


Figura 2.9: *KNIME*: nodos de pré-processamento de dados.

A limpeza inicial dos dados centrou-se em remover:

- features em função do número de *missing values* que apresentam: colunas com um 20% ou mais de valores omissos são automaticamente filtradas¹;
- entradas (linhas) duplicadas;
- linhas com dados inválidos²: o que se traduz na remoção dos dados referentes a músicas com valor não positivo de duração, com valor inválido de BPM ou com uma data de obtenção inválida.

Adicionalmente, foi filtrada a coluna *instance_id*, uma vez que corresponde a um identificador único de cada entrada de dados, não fornecendo, portanto, informação relevante para a deteção de padrões nos dados. O pré-processamento seguiu-se com a conversão dos dados referentes à coluna *tempo*: anteriormente descritos sob a forma de *string*, passaram a ser representados por valores numéricos – *double*.

O passo seguinte constitui o tratamento dos valores omissos. A análise estatística exploratória revelou cerca de 5 *missing values* por cada coluna, pelo que o seu tratamento centrou-se na substituição pelo valor mais frequente (no caso de a variável ser descrita em formato textual) ou pelo seu valor médio (no caso de a variável ser descrita em formato numérico). Esta decisão apoiou-se no facto de que a exclusão de linhas em função dos valores omissos pode conduzir a uma redução significativa no tamanho do *dataset*, pelo que foi optada pela manutenção dos dados, seguindo a estratégia de substituição previamente descrita.

O tratamento de *outliers* foi efetuado com recurso ao nodo *Numeric Outliers*, tendo sido aplicado a todas as variáveis numéricas, de acordo com a seguinte estratégia: todos os *outliers* foram substituídos pelo mais próximo permitido – com vista a não reduzir o tamanho do *dataset* de trabalho.

Uma vez que a coluna *obtained_date* refere uma data, foi trabalhada de modo a ser possível extraír os seus campos. Pela análise estatística conduzida anteriormente, foi possível visualizar que: (i) a informação do ano encontra-se omissa; (ii) todas as datas referem o mesmo mês. Assim

¹Do ponto de vista teórico, o *threshold* poderia ser aumentado até, no máximo, 30% de *missing values*

²A presença deste tipo de valores foi detetada com recurso ao estudo estatístico previamente conduzido e através da observação direta do conjunto de dados

senso, o tratamento desta variável passou pela criação de um artifício correspondente ao ano da data, para que este possibilitasse a conversão do formato textual para o formato *Date*, com o nodo *String to Date&Time*. Posteriormente, foi extraído, com o nodo *Extract Date&Time Fields*, os campos referentes ao mês (número) e ao dia do mês (número)³.

Por fim, as variáveis categóricas *artist_name*, *track_name*, *mode* e *key* foram convertidos em valores numéricos, através do nodo *Category To Number* – tendo sido removidas posteriormente as colunas originais com o nodo *Column Filter*⁴.

2.5 Análise dos dados após o pré-processamento

Uma vez findo o pré-processamento dos dados presentes no *dataset* foi conduzido, novamente, um estudo estatístico e correlacional dos dados. Esta análise permitiu observar o comportamento dos dados após o seu tratamento inicial: não são detetados valores omissos nas diversas colunas do conjunto de dados e não existem *outliers* nas diversas *features* – cf. Figura 2.10. Adicionalmente, foi, novamente, gerado o gráfico que permite observar a medida de *skewness* dos dados – Figura 2.11 – onde se continua a observar alguma assimetria na distribuição dos dados de algumas colunas.

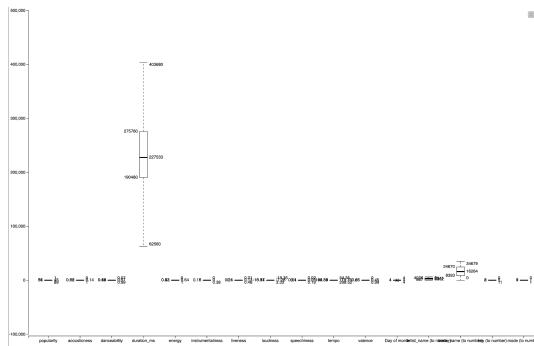


Figura 2.10: KNIME: Box Plot.

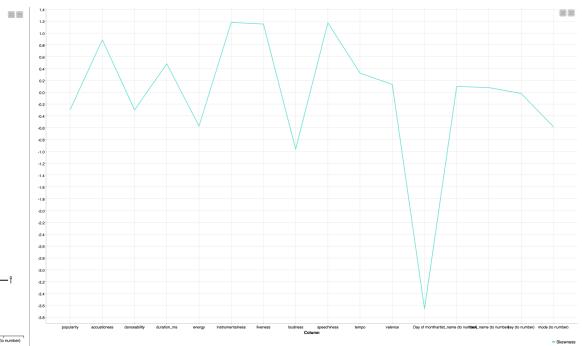


Figura 2.11: KNIME: Line Plot.

Adicionalmente, foi averiguada, novamente, o estado de correlação entre as variáveis, que pode ser observado nas Figuras 2.12 e 2.13, com vista a identificar possíveis diferenças após o tratamento e limpeza incial do conjunto de dados.

³A informação sobre o ano foi, naturalmente, ignorada uma vez que adicionada pelo grupo com o intuito de facilitar a conversão entre os dois formatos.

⁴Foi ainda removida a *feature* extraída *Month (number)*, uma vez que o seu conteúdo era idêntico para todas as entradas

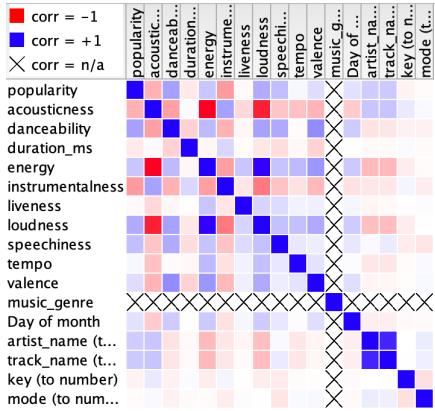


Figura 2.12: *KNIME: Linear Correlation*.

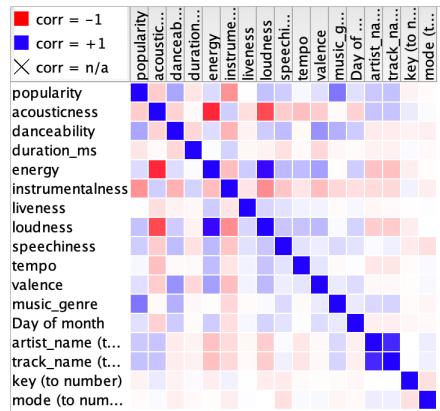


Figura 2.13: *KNIME: Rank Correlation*.

Pela análise da Figura 2.12 pode observar-se que as *features* *loudness* e *energy* correlacionam-se positivamente, enquanto os pares de variáveis *loudness-acousticness* e *energy-acousticness* se correlacionam negativamente. Adicionalmente, observando a Figura 2.13, verifica-se que a variável *music_genre* continua a correlacionar-se positivamente com as *popularity* e *danceability* e negativamente com a coluna *instrumentalness*.

2.6 Validação dos modelos de *Machine Learning*

A validação dos modelos é de elevada importância aquando do desenvolvimento de modelos de aprendizagem automática. Tal acontecimento deve-se, essencialmente, pelo facto de que o uso dos dados do *dataset* na íntegra configura um caso de potencial enviesamento aquando da geração de modelos de ML, i.e., ao usar a totalidade dos dados, sem técnicas de validação, os modelos podem obter um bom desempenho por estarem demasiado ajustados aos dados usados e obter más *performances* quando deparados com um novo conjunto de dados. Este problema de *overfitting* pode ser reduzido através de técnicas de validação, procurando assim que o desempenho dos modelos criados seja reflexo de padrões encontrados nos dados e não de eventuais ruídos que possam ocorrer no *dataset*.

O trabalho conduzido recorreu a duas técnicas: *hold-out validation* e *cross-validation*. A técnica de validação *hold-out* consiste no segregação do conjunto inicial de dados em dois *datasets*: um usado para treino dos modelos e outro para teste, com vista a avaliar o comportamento do modelo com dados nunca vistos. A técnica de validação consistiu numa divisão de 80% e 20% para os *datasets* de treino e teste, respetivamente. A estipulação destes valores centrou-se no facto de: (i) ser uma divisão usual para o método em uso; (ii) a divisão do *datasets* deve ser efetuada de modo a que sejam alocados dados suficientes tanto para treino como para teste, evitando assim elevada variação na performance dos modelos de ML.

Por outro lado, a técnica de *cross-validation* permite a divisão do conjunto de dados num número, k , de grupos – motivo pelo qual esta técnica também é conhecida como *k-fold cross-validation*. Um dos grupos é usado como conjunto de teste para o modelo, e os restantes usados como conjunto de treino. O processo repete-se iterativamente, até que os k grupos tenham sido usados como *dataset* de teste. A técnica de validação cruzada fixou o valor de $k=10$, uma vez que: (i) é uma definição do número de *folds* usual em ML; (ii) é um valor que resulta, geralmente, em modelos com baixo enviesamento.

Independentemente do método de validação, a separação dos dados para teste e treino foi realizada de modo aleatório, com uma *seed* igual a 2022. Esta decisão visa permitir que o desempenho dos modelos, dentro de cada método de validação, sejam comparáveis, uma vez que todos os modelos treinam e testam o mesmo conjunto de dados – apesar de estes conjuntos serem gerados aleatoriamente. As configurações usadas no *workflow* para os nodos de *Partitioning* e *X-Partitioner* pode ser consultadas nas Figuras 2.14 e 2.15, respectivamente.

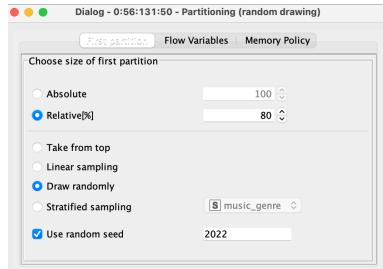


Figura 2.14: *KNIME: Partitioning*.

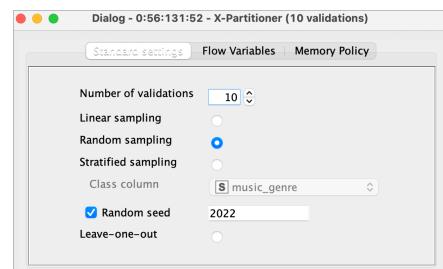


Figura 2.15: *KNIME: X-Partitioner*.

2.7 Modelos de *Machine Learning*

O problema em análise consiste num problema de classificação, motivo pelo qual as técnicas de aprendizagem automática aplicadas centram-se em árvores de decisão, regressão (logística) e redes neurais artificiais. A Figura 2.16 apresenta a expansão do super-nodo *Models*, apresentado na Figura 2.2, e contém uma visão generalista dos modelos de ML criados.

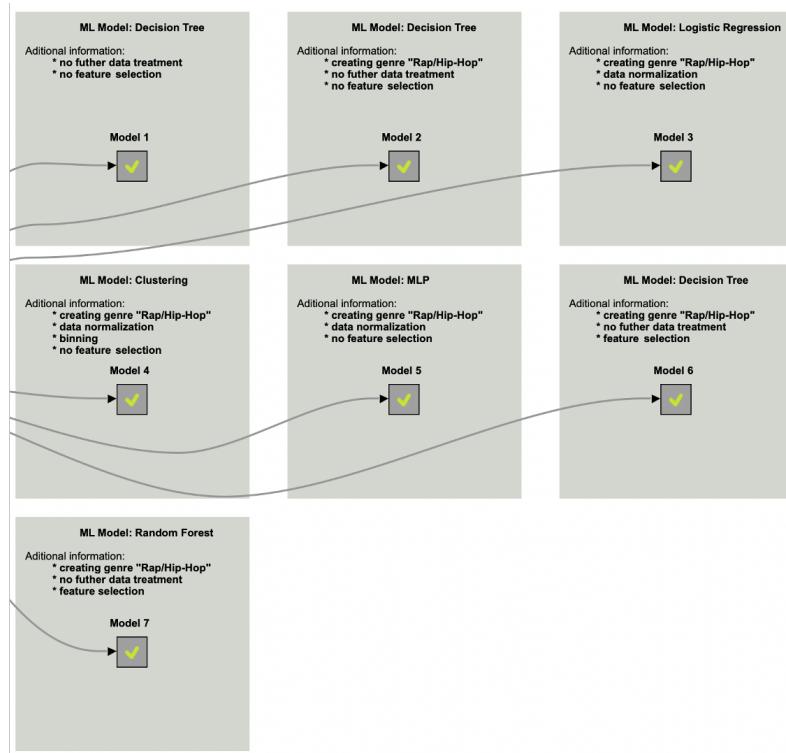


Figura 2.16: *KNIME: Modelos de ML*.

A Figura 2.17 configura o primeiro modelo de ML desenvolvido. Foi usada uma estratégia com recurso a árvores de decisão, sem ter ocorrido qualquer tipo de tratamento adicional de dados nem *feature selection*. A criação deste modelo procurou: (i) avaliar o pré-tratamento de dados realizado anteriormente; e (ii) obter uma *baseline* para a criação de novos modelos.

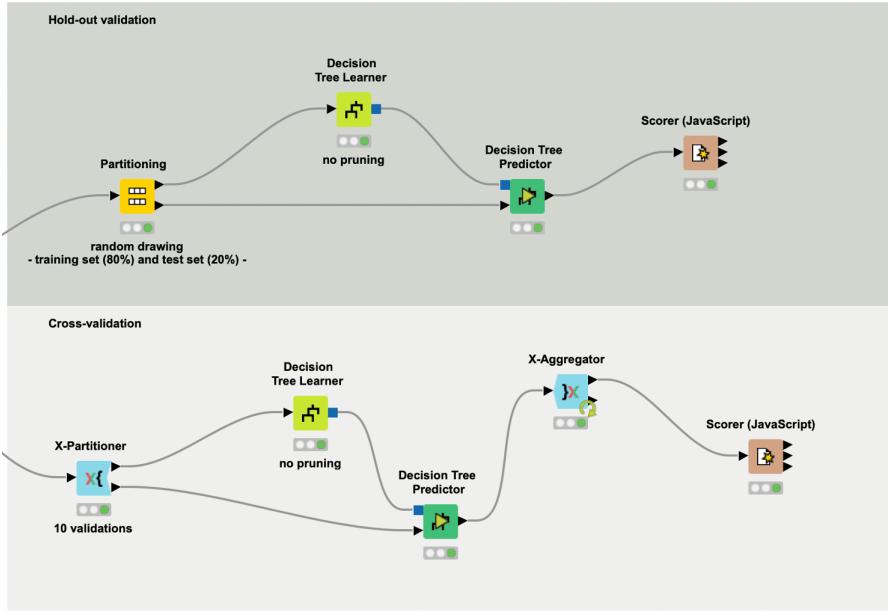


Figura 2.17: ML: Modelo 1.

Após a aplicação do nodo *Scorer* verificou-se que os valores de *accuracy* para os modelos com validação *hold-out* e cruzada são de 85.78% e 85.20%, respetivamente.

Analizando atentamente os Anexos A.1 e A.2, pode verificar-se que, de um modo geral, a percentagem de dados corretamente categorizados pelo modelo é aproximadamente igual, excetuando para as categorias *Rap*, *Hip-Hop*, *Alternative* e *Rock* que apresentam uma taxa de acerto inferior às restantes categorias. Avaliando, mais ao pormenor o conteúdo das matrizes de confusão, pode verificar-se que uma porção das entradas erradamente categorizadas para as categorias *Rap* e *Hip-Hop* pertencem às categorias *Hip-Hop* e *Rap*, respetivamente – o que pode ser indicativo de um padrão homólogo nos dados para estes géneros musicais.

De facto, no mundo real, é bastante usual a categorização conjunta destes dois estilos musicais, o que sustenta a hipótese de estas categorias partilharem similaridades em termos de características. Esta similaridade motivou a criação de um novo modelo, Figura 2.18, onde foi adicionado um novo nodo, *Java Snippet*, para transformar o *target* do *dataset*.

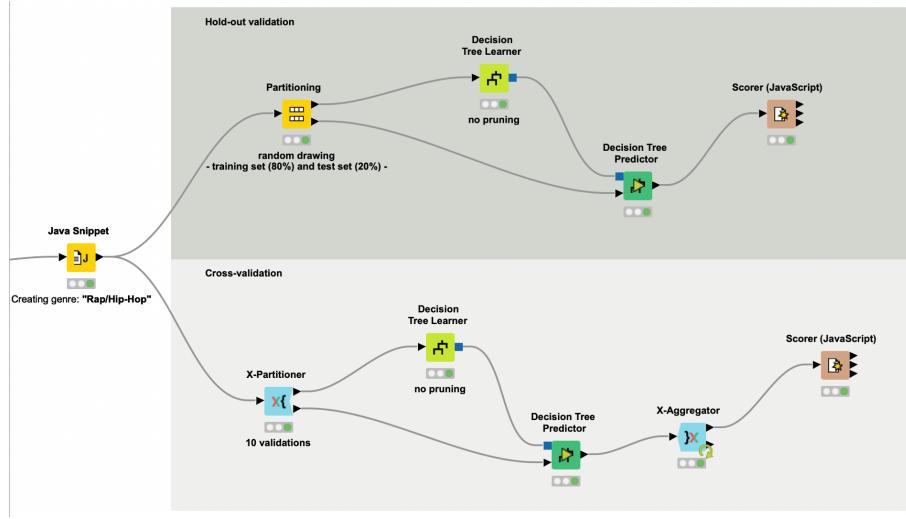


Figura 2.18: ML: Modelo 2.

Este nodo permite transformar o conteúdo da variável *music_genre*, criando uma nova categoria: *Hip-Hop/Rap*⁵. Analisando as Figuras A.3 e A.4 pode verificar-se que o valor de *accuracy* para os modelos com validação *hold-out* e cruzada cresceu para 92.47% e 91.68%, respectivamente. Uma vez que a junção destas duas categorias se traduziu num aumento da performance dos modelos, os modelos seguintes incorporaram esta operação de agregação dos estilos musicais *Hip-Hop* e *Rap*.

O modelo apresentado na Figura 2.19 baseou-se numa técnica de regressão logística para resolver o problema de classificação já apresentado. A primeira versão do modelo aqui apresentado consistia simplesmente num modelo de aprendizagem automática sem tratamento adicional de dados. Contudo, a precisão do modelo, para ambos os métodos de validação, revelou-se baixa⁶, levantando a questão acerca da necessidade de tratamento posterior dos dados.

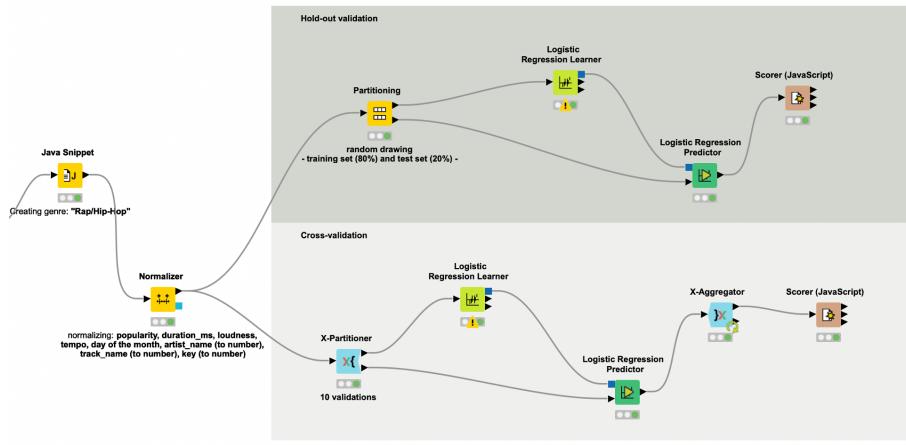


Figura 2.19: ML: Modelo 3.

A baixa precisão demonstrada anteriormente pode estar associada com a disparidade observada

⁵Todas as entradas com o valor de *music_genre* igual a *Hip-Hop* ou *Rap* são convertidas em *Hip-Hop/Rap*.

⁶Aproximadamente 30%.

nas escalas⁷; como tal, foi efetuada uma normalização dos dados, com o nodo *Normalizer*, a fim de uniformizar as escalas entre as diversas *features*. Uma vez que algumas variáveis já se encontravam descritas numa escala entre 0 e 1, e.g., *liveness*, foi usada a normalização Min-Max, com valores de 0 e 1, para as restantes *features*: *popularity*, *duration_ms*, *loudness*, *tempo*, *day of the month*, *artist_name (to number)*, *track_name (to number)*, *key (to number)*.

A visualização dos resultados gerados pelo nodo *Scorer* revelou que os valores de *accuracy* para os modelos com validação *hold-out* e cruzada são de 81.65% e 81.17%, respetivamente – cf. Anexos A.5 e A.6.

O modelo seguinte, apresentado na Figura 2.20, tomou como princípio base uma estratégia de segmentação, vulgo, *clustering*. Como tal, surgiu a necessidade de efetuar algum tratamento adicional aos dados: (i) as variáveis contínuas foram normalizadas, para reduzir o efeito das unidades de medida; (ii) foram criados *bins* para as variáveis nominais – através do nodo *Auto-Binner*. A estratégia de segmentação adotada, *k-means*, procura distribuir os dados observados por um determinado número de segmentos. Uma vez que, no presente momento, o *dataset* encontra-se a trabalhar com 9 géneros musicais distintos, o valor atribuído para *k*, número de *clusters*, foi também 9.

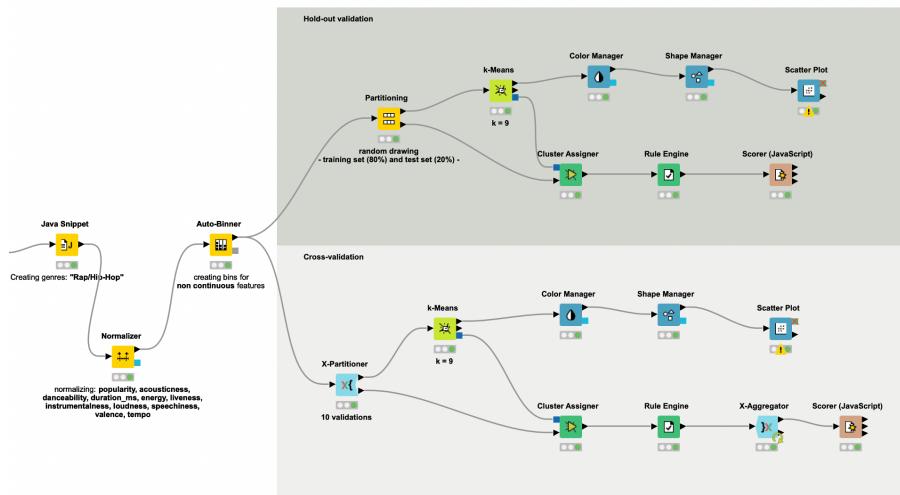


Figura 2.20: ML: Modelo 4.

Os resultados obtidos com o *Scorer* revelou que os valores de *accuracy* para os modelos com validação *hold-out* e cruzada são de 7.31% e 13.92%, respetivamente – cf. Anexos A.7 e A.8. Uma vez que os resultados de precisão obtidos ficaram aquém daquilo que se tem averiguado nos restantes modelos, foram efetuadas algumas variações ao modelo apresentado na Figura 2.20:

- Variação no número de *bins*: considerando valores diferentes daquele estipulado (e.g., 3, 4, 9, 10, 15) os modelos produzidos apresentam uma baixa, ou nenhuma, variação em termos de *accuracy*;
- Não normalização de dados: a não normalização dos dados contínuos traduz num pequeno aumento da precisão do modelo com validação *hold-out*, e numa pequena descida da precisão para o modelo com validação cruzada, 9.10% e 11.58%, respetivamente;
- *Rule Engine*: alteração dos segmentos atribuídos a cada género musical;

⁷Revelada pela análise estatística efetuada inicialmente

- *Feature selection*: foram removidas algumas colunas, e.g., *tack_name* e *artist_name*, mas sem produzir alterações significativas nos resultados observados.

As estratégias até então adotadas para melhorar a performance do modelo revelaram-se pouco, ou nada, eficazes, pelo que surgiu a questão acerca do que poderia explicar a disparidade de *scores* entre o Modelo 4 e os restantes modelos, previamente apresentados. Um possível problema pode centrar-se na configuração do nodo *Rule Engine*. Este nodo encontra-se responsável por etiquetar os 9 segmentos gerados pelo modelo, mapeando-os com um género musical: caso este mapeamento se encontre incorreto seria expectável que o modelo de aprendizagem automática obtivesse uma baixa performance⁸. Adicionalmente, uma estratégia de *clustering* pode não ser a abordagem mais sensata para o problema em questão, em parte pelo número elevado de categorias distintas existentes em termos dos géneros musicais e, ainda, pelo vasto número valores distintos nas variáveis nominais *tack_name* e *artist_name*.

O modelo seguinte, apresentado na Figura 2.21, foi construído segundo uma estratégia de redes neuronais artificiais – usando um *Multi Layer Perceptron*. Devido à elevada sensibilidade deste tipo de modelos em termos da distribuição dos dados, foi necessário proceder à normalização das features, com vista a não produzir resultados enviesados.

A normalização dos dados ocorre após a divisão do *dataset* em treino e teste⁹: numa fase inicial, são tratados os *outliers* do *dataset* de teste, sendo posteriormente normalizado¹⁰. A normalização é depois aplicada, com o nodo *Normalizer (Apply)*, aos dados usados para treino, garantindo assim que os dois conjuntos encontram-se sempre normalizados.

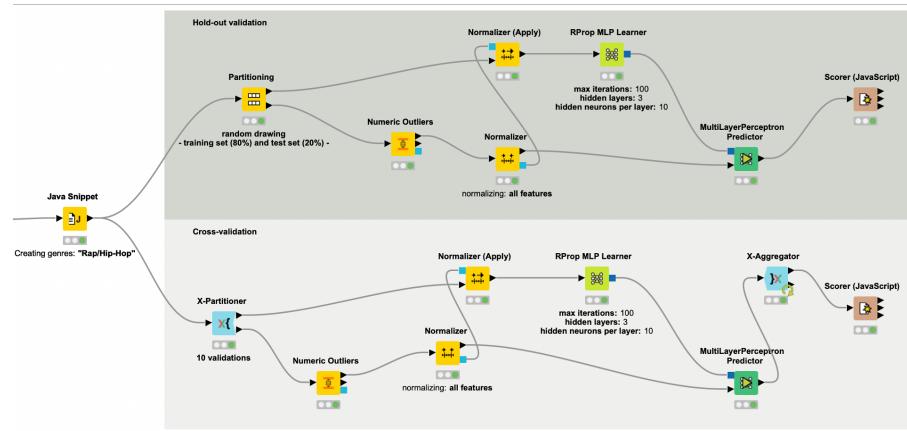


Figura 2.21: ML: Modelo 5.

A aplicação do *Scorer* demonstrou que os valores de precisão para os modelos com validação *hold-out* e cruzada são de 80.13% e 79.58%, respetivamente – cf. Anexos A.9 e A.10.

O Modelo 6, apresentado na Figura 2.22, centra-se na seleção de *features* para a geração de um modelo de aprendizagem com recurso a uma estratégia baseada em árvores de decisão. A filtragem de colunas do conjunto de dados tem como principal objetivo reduzir a complexidade do problema. Como tal, este modelo pretende averiguar se a redução de dimensionalidade em termos de *features* pode traduzir-se numa alteração na performance dos modelos.

⁸Foram efetuadas sucessivas tentativas de reconstruir o mapeamento de forma diferente, mas sem grandes alterações nos resultados obtidos

⁹Note-se que a realização da normalização antes da divisão dos dados entre treino e teste não garante que os conjuntos gerados posteriormente estejam normalizados.

¹⁰Para diferir do método usado anteriormente nos modelos apresentados, foi usada *decimal scaling normalization*.

A seleção das colunas que serão usadas no Modelo 6 segue dois princípios de filtragem: (i) em função da variância dos dados¹¹; e (ii) em função da assimetria (*skewness*) dos dados¹².

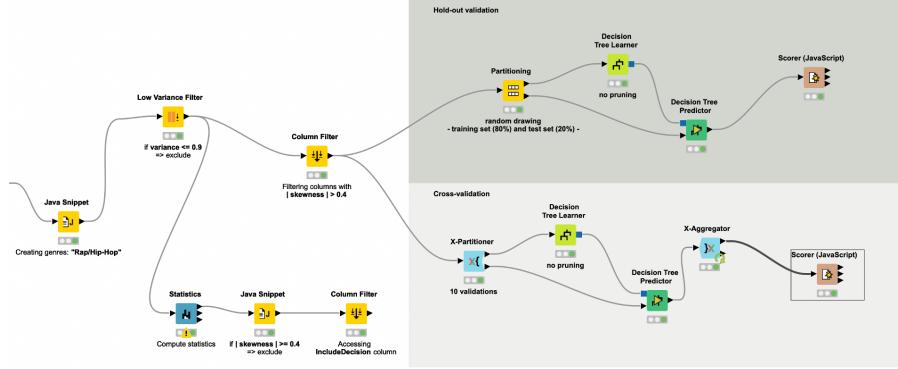


Figura 2.22: ML: Modelo 6.

Assim sendo, com recurso ao nodo *Low Variance Filter*, foram inicialmente removidas as colunas com um valor baixo de variância (*threshold*: 0.9). Seguidamente, foram filtradas as colunas com um valor de assimetria superior a 0.4: inicialmente, foi calculado o valor de *skewness* das colunas, com o nodo *Statistics*, e através do nodo *Java Snippet* foi produzida uma nova coluna, *IncludeDecision*, onde é atribuído o valor "include" caso o módulo da assimetria se encontre acima do valor estipulado, ou "exclude" em caso contrário. Uma vez visualizados os resultados desta nova coluna, são filtradas as variáveis cujo valor de *skewness* ultrapassa o limiar. Com a aplicação do *Scorer* pode ser verificado que os valores de precisão para os modelos com validação *hold-out* e cruzada são de 92.78% e 92.21%, respetivamente – cf. Anexos A.11 e A.12.

Por fim, foi criado o Modelo 7, Figura 2.23, que configura outra abordagem para a *feature selection*, recorrendo a um *Random Forest*.

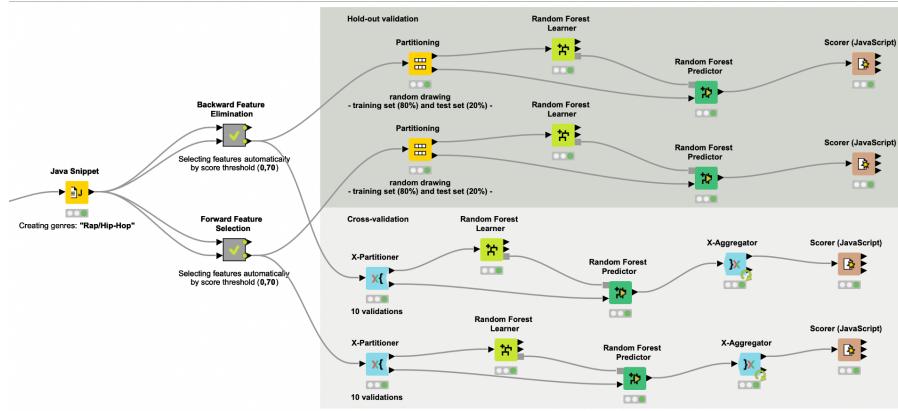


Figura 2.23: ML: Modelo 7.

A exploração da seleção das colunas, foi efetuada com recurso aos nodos *Backward Feature Elimination* e *Forward Feature Selection*. Para a seleção automática das *features*, considerando a performance dos modelos já apresentados, estipulou-se que o limiar inferior de desempenho

¹¹Uma vez que colunas com pouca variação podem ser pouco vitais para a determinação de padrões nos dados.

¹²Dados assimétricos podem enviesar os modelos, criando tendências com base na sua falta de simetria.

desejado seria de 0,7. O resultado da filtragem para *Backward Feature Elimination* e *Forward Feature Selection* encontra-se presente nas Figuras 2.24 e 2.25, respetivamente. Como pode ser observado, em ambos os caso foram selecionadas apenas duas *features* para prever o género musical dos dados referentes às músicas: *artist_name (to number)* e *popularity*.

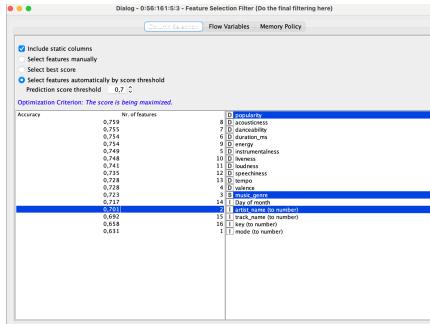


Figura 2.24: *KNIME: Backward Feature Elimination*. Figura 2.25: *KNIME: Forward Feature Selection*.

Os resultados observados através do *Scorer* revelam valores de precisão para os modelos gerados através de *Backward Feature Elimination* de 87,90% (para ambos os métodos de validação) e de 87,10% para os modelos gerados através de *Forward Feature Selection* (para ambos os métodos de validação).

2.8 Análise dos modelos de *Machine Learning*

A Tabela 2.1 apresenta um sumário dos valores registados para a *accuracy* dos modelos de ML previamente explicados¹³

Tabela 2.1: Precisão (%) dos modelos de ML desenvolvidos.

Modelo	Hold-out validation	Cross-validation
1	85,78	85,20
2	92,47	91,68
3	81,65	81,17
4	7,31	13,92
5	80,13	79,58
6	92,23	92,21
7	87,90 (BFE) — 87,10 (FFS)	87,90 (BFE) — 87,10 (FFS)

De um modo geral, os modelos produzidos geraram resultados satisfatórios, sendo capaz de prever os géneros musicais das entradas contidas no *dataset*. Dentro dos modelos desenvolvidos, destaca-se, pela sua baixa performance, o Modelo 4 (Segmentação) que se demonstrou bastante desajustado para resolver o problema de classificação. Os Modelos 2 e 6 destacam-se, pela positiva, uma vez que a sua performance alcançou os 90%. No que diz respeito às estratégias de validação usadas, e tal como se observa na Tabela 2.1, não são observadas grandes disparidades entre as performances dos modelos quando usada uma estratégia de validação *hold-out* ou cruzada.

¹³BFS: *Backward Feature Elimination*; FFS: *Forward Feature Selection*

Capítulo 3

Salary Classification

3.1 *Dataset*

De entre os *dataset* fornecidos, o conjunto de dados sorteado para o nosso grupo (visto ser o número 12) foi o *dataset salary_classification.csv*. Destinando a sua primeira entrada ao cabeçalho, que identifica o significado das diversas colunas, ficheiro *CSV* é constituído por 48842 registos, onde cada linha corresponde a dados sobre um determinado trabalhador. O objetivo deste trabalho prático é, então, prever o salário de um trabalhador, atendendo a um conjunto de características.

A seguir, é apresentada uma lista das *features* presentes no conjunto de dados – Figura 3.1, caracterizando o seu significado e o tipo de dados utilizados para a sua representação:

- ***age***: a idade do trabalhador, entre 17 e 90 (representado sob a forma de um *int*);
- ***workclass***: medida que descreve o trabalhador numa determinada classe social (representado sob a forma de uma *string*);
- ***fnlwgt***: número de pessoas que se acredita verificar a entrada apresentada (representado sob a forma de um *int*);
- ***education***: nível de educação do trabalhador (representado sob a forma de uma *string*);
- ***education-num***: nível de educação do trabalhador, entre 1 e 16 (representado sob a forma de um *int*);
- ***material-status***: estado civil do trabalhador (representado sob a forma de uma *string*);
- ***occupation***: profissão/função do trabalhador (representado sob a forma de uma *string*);
- ***relationship***: relação do trabalhador com os outros (representado sob a forma de uma *string*);
- ***race***: descrição sobre a raça do trabalhador (representado sob a forma de uma *string*);
- ***sex***: descrição sobre o sexo do trabalhador (representado sob a forma de uma *string*);
- ***capital-gain***: ganhos de capital do trabalhador (representado sob a forma de um *int*);
- ***capital-loss***: perdas de capital do trabalhador (representado sob a forma de um *int*);

- **hours-per-week**: número de horas de trabalho por semana do trabalhador, entre 1 e 99 (representado sob a forma de um *int*);
- **native-country**: país nativo do trabalhador (representado sob a forma de uma *string*);
- **salary-classification**: descrição se o trabalhador ganha ou não mais de 50K\$ por ano (representado sob a forma de uma *string*);

```
age; workclass; fnlwgt; education; education-num; marital-status; occupation; relationship; race; sex; capital-gain; capital-loss; hours-per-week; native-country; salary-classification
39; State-gov; 7516; Bachelors; 13; Never-married; Adm-clerical; Not-in-family; White; Male; 2174; 0; 0; United-States; <=50K
50; Self-emp-not-inc; 83311; Bachelors; 13; Married-civ-spouse; Exec-managerial; Husband; White; Male; 0; 0; 13; United-States; <=50K
38; Private; 215646; HS-grad; 9; Divorced; Handlers-cleaners; Not-in-family; White; Male; 0; 0; 40; United-States; <=50K
53; Private; 234721; 11th; 7; Married-civ-spouse; Handlers-cleaners; Husband; Black; Male; 0; 0; 40; United-States; <=50K
28; Private; 338409; Bachelors; 13; Married-civ-spouse; Prof-specialty; Wife; Black; Female; 0; 0; 40; Cuba; <=50K
37; Private; 284582; Masters; 14; Married-civ-spouse; Exec-managerial; Husband; White; Female; 0; 0; 40; United-States; <=50K
49; Private; 168187; 9th; 5; Married-spouse-absent; Other-service; Not-in-family; Black; Female; 0; 0; 16; Jamaica; <=50K
52; Self-emp-not-inc; 209642; HS-grad; 9; Married-civ-spouse; Exec-managerial; Husband; White; Male; 0; 0; 45; United-States; >50K
31; Private; 45781; Masters; 14; Never-married; Prof-specialty; Not-in-family; White; Female; 14084; 0; 50; United-States; >50K
42; Private; 159449; Bachelors; 13; Married-civ-spouse; Exec-managerial; Husband; White; Male; 5178; 0; 40; United-States; >50K
37; Private; 280464; Some-college; 10; Married-civ-spouse; Exec-managerial; Husband; Black; Male; 0; 0; 80; United-States; >50K
30; State-gov; 141297; Bachelors; 13; Married-civ-spouse; Prof-specialty; Husband; Asian-Pac-Islander; Male; 0; 0; 40; India; >50K
23; Private; 122272; Bachelors; 13; Never-married; Adm-clerical; Own-child; White; Female; 0; 0; 30; United-States; <=50K
```

Figura 3.1: *KNIME*: Estrutura do ficheiro *salary_classification.csv*

3.2 KNIME workflow

A estrutura do trabalho realizado para o *dataset salary_classification* encontra-se representado na Figura 3.2, na qual se observam os seguintes grupos:



Figura 3.2: *KNIME*: workflow

3.3 Análise preliminar dos dados

Ao conhecer o ficheiro *salary_classification.csv*, previamente introduzido na secção anterior, verificou-se a ausência de dados em algumas entradas. No entanto, como estes campos vazios estavam preenchidos por pontos de interrogação em formato de *string*, não eram reconhecidos como *missing values*. Para solucionar o problema, resolveu-se aplicar um ciclo, ilustrado na Figura 3.3, que itera cada uma das colunas e que substitui qualquer *string* ? por um *null*. No final do ciclo, já é possível identificar todos os *missing values* presentes no ficheiro CSV, ainda por processar.

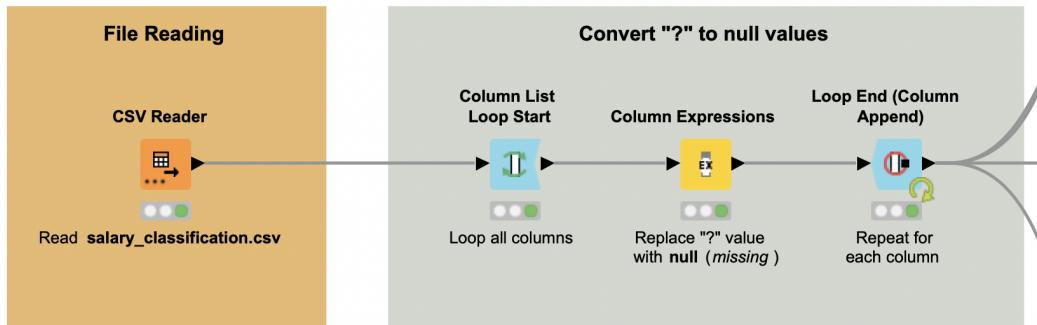


Figura 3.3: *KNIME*: Reconhecimento de *strings* ? como *missing values*

Antes de começar o pré-processamento dos dados, foi necessário realizar uma análise geral do *dataset* em mãos – Figura 3.4, de modo a ser possível definir um esquema eficiente para o tratamento dos dados e, posteriormente, aplicar algumas técnicas de *Machine Learning*, pondo em prática o conhecimento adquirido nas aulas teóricas e trabalhado nas aulas práticas.

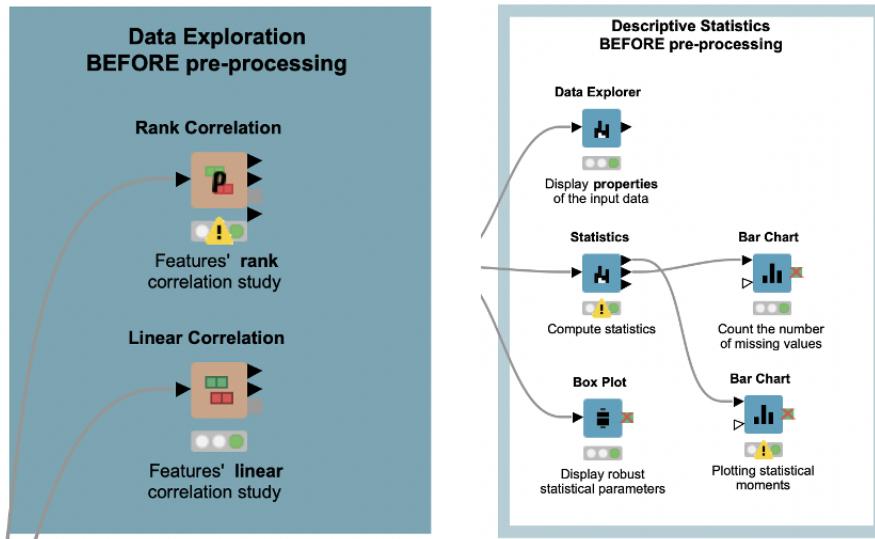


Figura 3.4: *KNIME*: análise inicial dos dados

Através do *output* gerado pelo nodo *Data Explorer*, adquirido com a instalação da extensão *KNIME JavaScript Views (Labs)*, consegue-se observar, na Figura 3.5, os momentos estatísticos de todas as *features* com valores numéricos, nomeadamente os extremos (mínimo e máximo), a

média e o desvio padrão. Na Figura 3.6, observa-se os momentos estatísticos, como o número de *missing values*, produzidos pelo mesmo nodo, mas para as *features* de valores nominais.

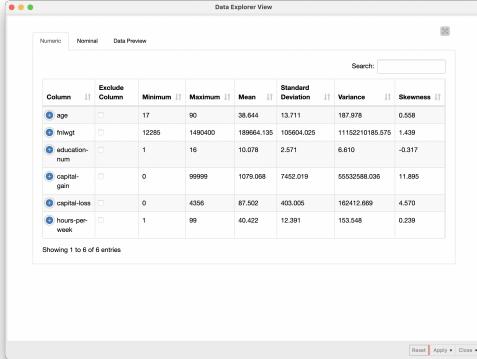


Figura 3.5: *KNIME: Output de Data Explorer*
para valores numéricos

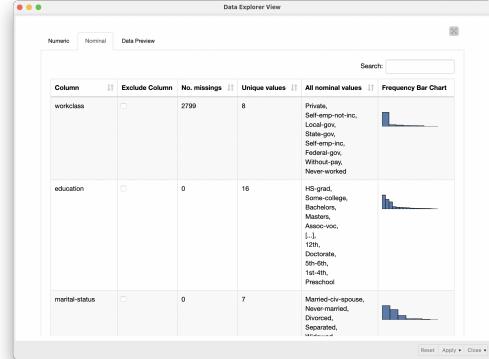


Figura 3.6: *KNIME: Output de Data Explorer*
para valores nominais

Adicionalmente, na Figura 3.7, temos o *output* gerado pelo nodo *Blox Plot*, no qual se destaca a *feature* *fnlwgt* por possuir uma gama de valores muito diferente da das restantes. Esta *feature* também se sobressai em relação às outras por possuir inúmeros *outliers*, i.e. dados que se diferenciam drasticamente de todos os outros, o que pode causar anomalias nos resultados obtidos pelos modelos de aprendizagem automática adotados.

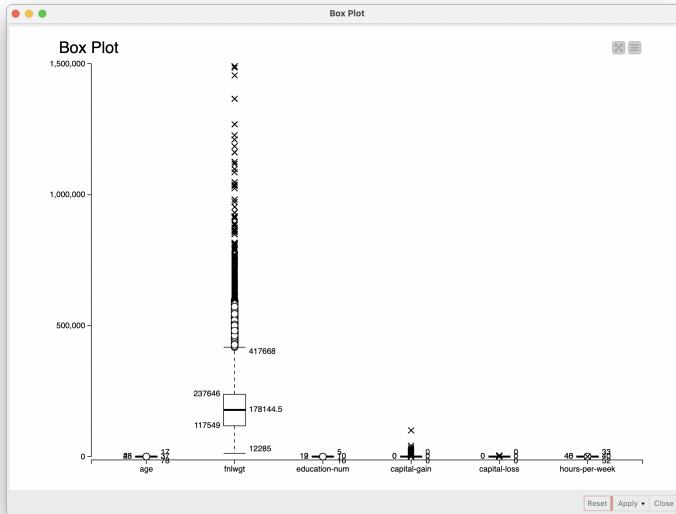


Figura 3.7: *KNIME: Output de Box Plot*

Para além da matriz de *linear correlation*, analisou-se a matriz de *rank correlation*, representada na Figura 3.8, para conhecer a medida de relação estatística entre duas variáveis do nosso *dataset*. Isto pode revelar ser uma informação muito útil na decisão de que *features* devem incorporar os nossos modelos de *Machine Learning*. Neste caso, existe uma correlação positiva entre as variáveis *education* e *education-num* e uma correlação negativa entre *sex* e *relationship*.

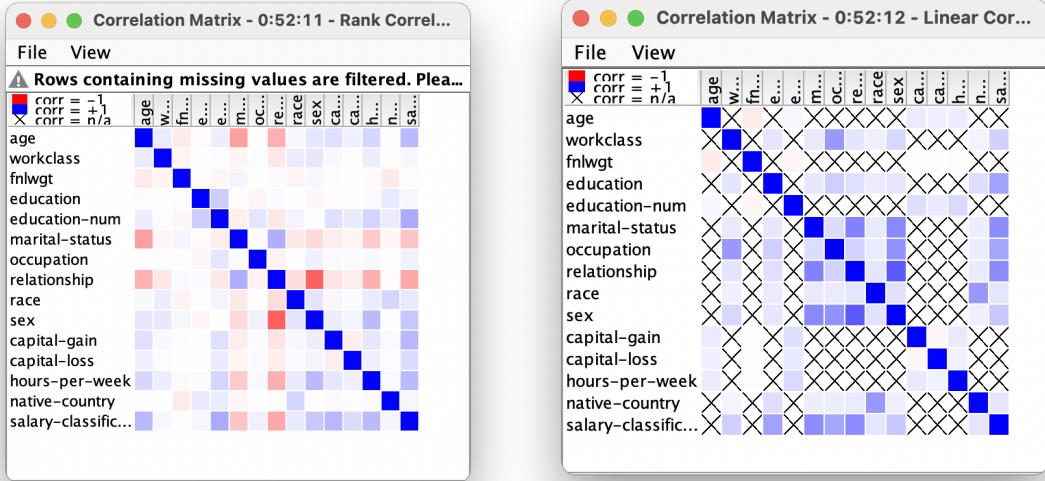


Figura 3.8: *KNIME: Output de Rank Correlation*

Figura 3.9: *KNIME: Output de Linear Correlation*

3.4 Pré-Processamento

Após ter sido feito um estudo cuidado ao conjunto de dados sorteado, desde existência de *missing values*, à relação entre duas variáveis, à identificação das *features* relevantes para os modelos de aprendizagem automática, foi possível idealizar uma forma de tratar e preparar o *dataset* para a criação de modelos de ML, estando esta representada na Figura 3.10.

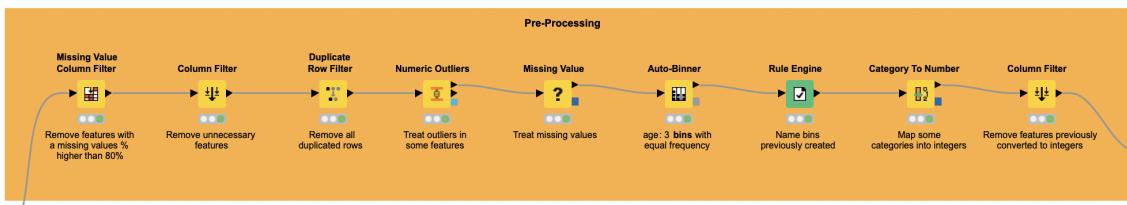


Figura 3.10: *KNIME: pré-processamento dos dados do dataset*

Primeiramente, fez-se uma limpeza geral aos dados, começando por remover:

- todas as colunas com uma percentagem de *missing values* superior a 80%;
- todos os registos (linhas) repetidos;
- todas as *features* que não influenciam o salário de um trabalhador, como é o caso de *marital-status* e *relationship*.

Para além destas duas *features*, removeu-se, também, a *education* porque, tal como analisámos anteriormente na Figura 3.8, esta variável possuía elevada correlação com *education-num*. Indu-

zindo com isto que a informação fornecida por ambas reduz-se à mesma, acabámos por excluir aquela em formato de *string*, contribuindo, deste modo, para uma melhor preparação dos dados.

Recorrendo à Figura 3.7, percebe-se que é necessário tratar dos *outliers* que se sobressaem em algumas features, em particular na *fnlwgt*. A decisão tomada perante isto foi substituir todos os *missing values* em variáveis do tipo *int* pela mediana dessa coluna e todos os *missing values* em variáveis do tipo *string* pela *string* mais frequente. Por outro lado, os *outliers* presentes nas features *education-num* e *hours-per-week* não sofreram qualquer tratamento visto serem valores que afetam em grande dimensão o salário de um trabalhador, i.e. alguém que tenha apenas concluído o secundário não ganhará tanto como se tivesse um mestrado, assim como alguém que trabalhe só 35 horas por semana recebe menos do que se trabalhasse 70 horas.

Seguidamente, como a *feature age* possuía um grande intervalo de valores ($\min = 17$, $\max = 90$), decidimos agrupar as idades em 3 *bins* de igual largura, categorizando-os da seguinte forma:

- 43- inclui idades inferiores a 43 anos e iguais ou superiores a 17 anos;
- [43-66] inclui idades entre os 43 (inclusive) e os 66 anos (inclusive);
- 66+ inclui idades superiores a 66 anos e iguais ou inferiores a 90 anos.

Finalmente, o último passo do pré-processamento dos dados foi converter as variáveis categóricas *workplace*, *race*, *sex*, *native-country*, *age/Binned* para valores numéricos. Para finalizar esta etapa, procedeu-se à remoção destas features originais que continham as ditas variáveis categóricas.

3.5 Análise dos dados após o pré-processamento

Uma vez terminado o tratamento incial do *dataset*, foi novamente conduzida uma análise dos dados, que procurou, em parte, confirmar que o trabalho conduzido sobre os dados foi eficaz em termos de, por exemplo, tratamento de *outliers* e de valores omissos. Adicionalmente, foi conduzido uma avaliação correlacional, a fim de observar eventuais diferenças que possam surgir entre as variáveis após o seu tratamento. A Figura 3.11 apresenta o trabalho executado nesta secção.

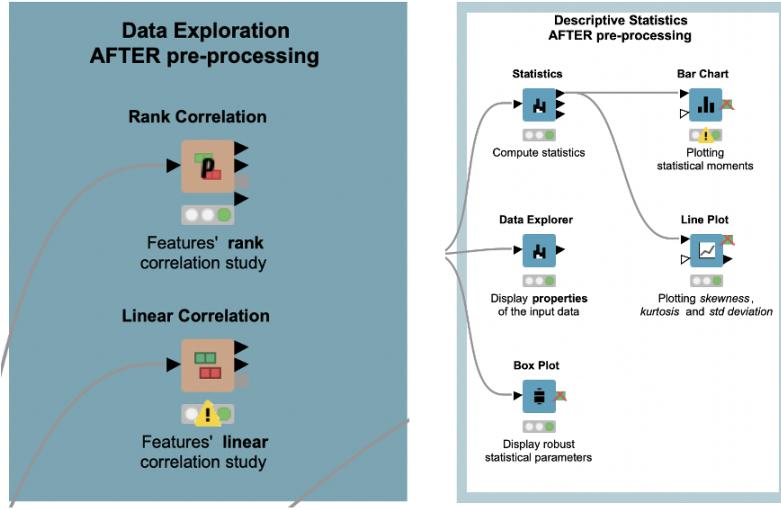


Figura 3.11: *KNIME*: Análise posterior dos dados

As Figuras 3.12 e 3.13 apresentam o *output* gerado para o nodo *Data Explorer* do *KNIME*, onde podem ser observadas as características das variáveis numéricas e nominais, respetivamente. Dentro das várias observações que podem ser realizadas, pode salientar-se o facto de que as colunas já não possuem *missing values*. Adicionalmente, importa referir que as escalas das diversas variáveis é altamente variável – basta olhar para a *feature fnlwgt* que possui um máximo de 417668 e a variável *hours-per-week* que tem um valor máximo de 52. Tal disparidade em termos da descrição dos valores pode acarretar problemas posteriormente aquando do desenvolvimento de alguns modelos de aprendizagem automática, pelo que, dependendo do modelo, pode ser necessário converter as categorias para uma escala comum.

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum	No. zero	No. missing	No. null	No. not null	Histogram
fnlwgt		12500	417668	167616.345	9320.055	808872791.511	0.355	-0.123	6107305051	0	0	0	0	
education		0	16	10.644	2.429	5.967	0.368	-0.271	614110	0	0	0	0	
capital-gain		0	0	0	0	0	0	0	48757	0	0	0	0	
capital-loss		0	0	0	0	0	0	0	48757	0	0	0	0	
hours-per-week		0	52	44.451	5.709	30.480	0.293	-1.596	2187298	0	0	0	0	
workclass (0=private)		0	7	3.105	3.916	0.842	1.150	3.207	163011	1681	0	0	0	
name-by-number		0	4	0.229	0.828	0.392	3.543	14.090	11710	11802	0	0	0	
sex-by-number		0	1	0.332	0.471	0.222	0.716	-1.408	16103	92594	0	0	0	
native-country (0=usa)		0	43	1.127	4.788	25.989	0.147	27.823	54358	44818	0	0	0	
age (0=under)		0	2	0.360	0.501	0.304	0.968	-0.254	18966	30566	0	0	0	

Figura 3.12: *KNIME*: Output de *Data Explorer* para valores numéricos

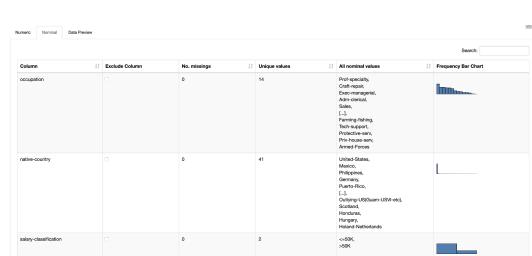


Figura 3.13: *KNIME*: Output de *Data Explorer* para valores nominais

Adicionalmente, esta disparidade na distribuição dos valores também pode ser observado no *box plot* apresentado na Figura 3.14, onde, tal como indicado previamente, se verifica que coluna *fnlwgt* se encontra descrita numa escala bastante díspar das restantes colunas. Ainda na Figura 3.14, pode ser observado que já não são detetados *outliers* para os dados que integram o *dataset*.

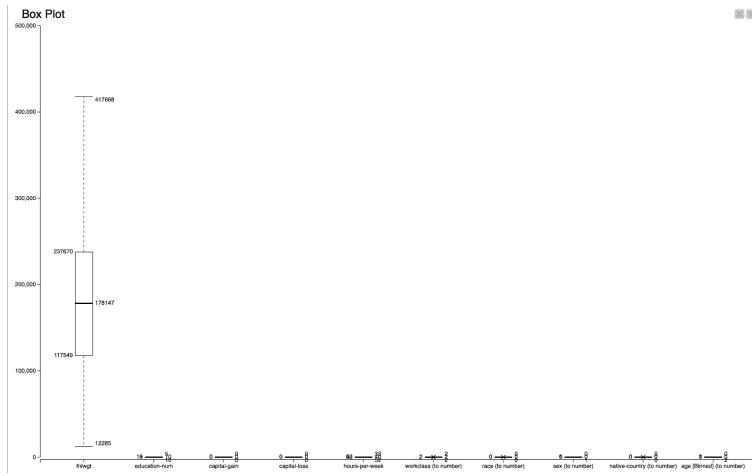


Figura 3.14: KNIME: Output de Box Plot

Por fim, tal como foi acima mencionado, foi avaliada a correlação entre as diversas variáveis do conjunto de dados. Para tal, e como foi anteriormente realizado, recorreram-se aos nodos *Rank Correlation* e *Linear Correlation*, cujo *output* se encontra apresentado nas Figuras 3.15 e 3.16, respectivamente.

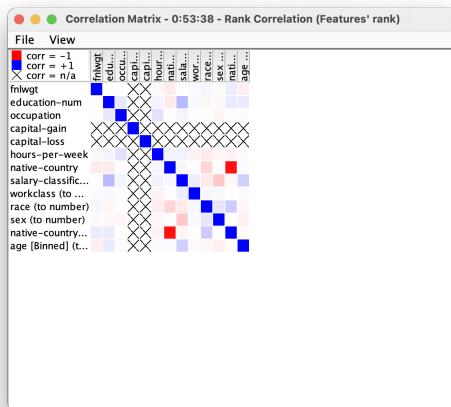


Figura 3.15: KNIME: Output de Rank Correlation

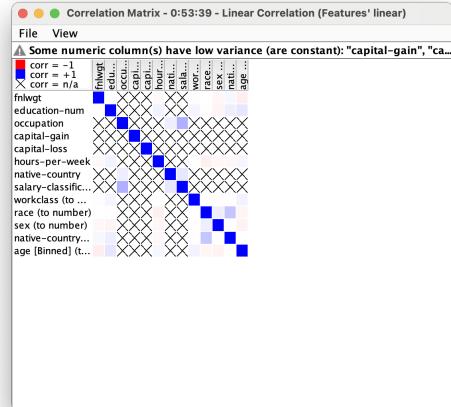


Figura 3.16: KNIME: Output de Linear Correlation

A análise das matrizes de correlação evidencia a manutenção da correlação positiva entre as variáveis *salary_classification* e *occupation*, *salary_classification* e *age*, e uma correlação negativa entre *sex* e *salary_classification*.

3.6 Modelos de *Machine Learning*

Após ter sido feita a validação dos modelos ML¹, e sendo também este um problema de classificação como o *dataset music_genre* previamente apresentado, procedeu-se à aplicação de técnicas de aprendizagem automática, i.e. árvores de decisão, florestas aleatórias, regressão (logística) e estratégias de segmentação. Na Figura 3.17, temos a estruturação dos modelos de *Machine Learning* adotados, que constituem o super-nodo *ML Models* apresentado na Figura 3.2

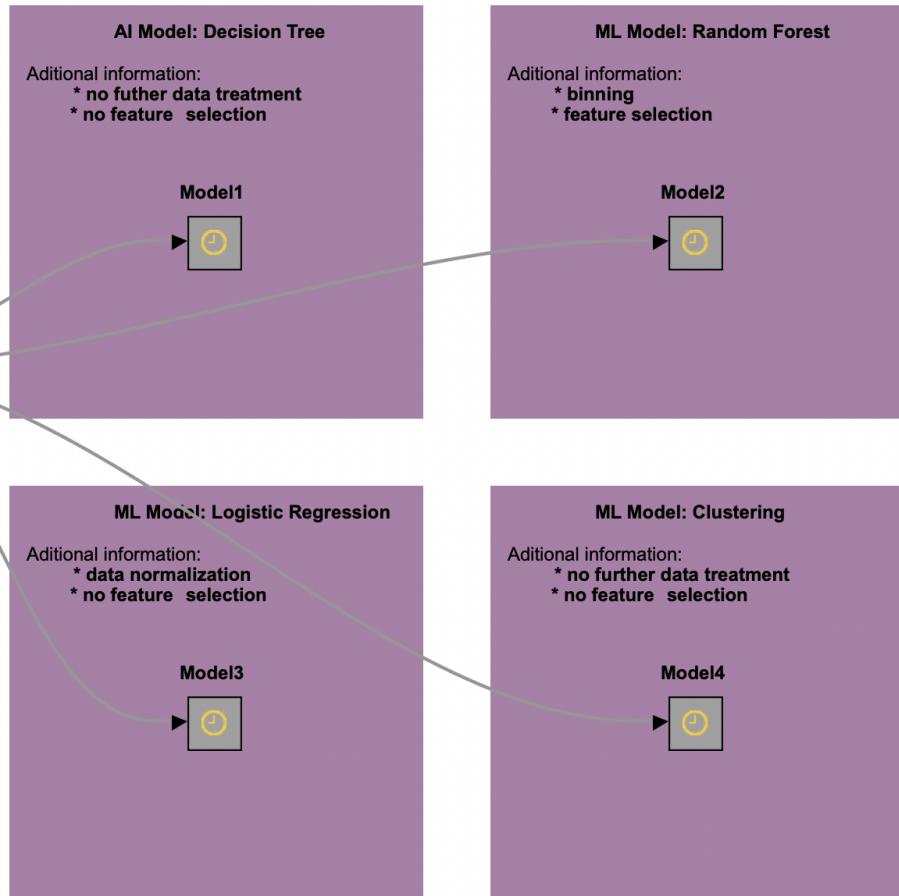


Figura 3.17: *KNIME*: Modelos de *Machine Learning*

Na Figura 3.18, está ilustrado o primeiro modelo de ML desenvolvido, idêntico ao adotado para o *dataset music_genre*². Através do nodo *Scorer*, é obtém-se os resultados de *accuracy* para os modelos com *hold-out* e *cross validation*, que são, respectivamente, 75.25% e 75.23%.

Analizando atentamente os Anexos B.1 e B.2, verifica-se que a percentagem de dados corretamente categorizados pelo modelo difere consideravelmente entre as categorias $\leq 50K$ e $> 50K$, apresentando a primeira um valor superior comparativamente com a segunda.

¹Para mais informação acerca dos métodos de validação dos modelos, consultar Seção 2.6.

²Consultar mais informação acerca do primeiro modelo na secção 2.7

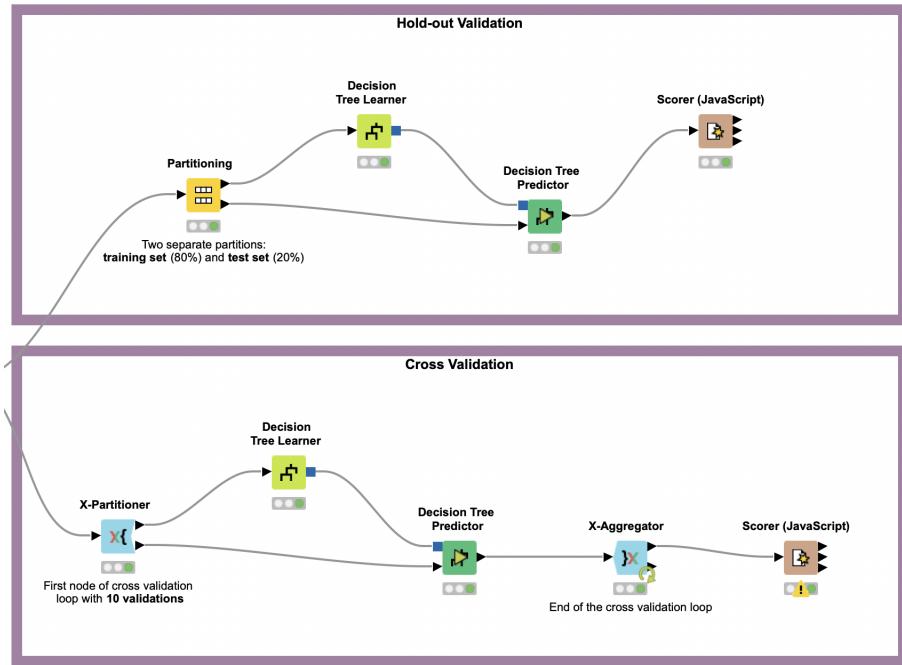


Figura 3.18: *KNIME*: Modelo 1 – Árvore de Decisão

Na Figura 3.19, temos o segundo modelo de ML, idêntico ao aplicado ao *dataset music_genre*³, que recorre aos super-nodos *Backward Feature Elimination* e *Forward Feature Selection* a fim de configurar uma abordagem para *feature selection*, recorrendo a um *Random Forest*.

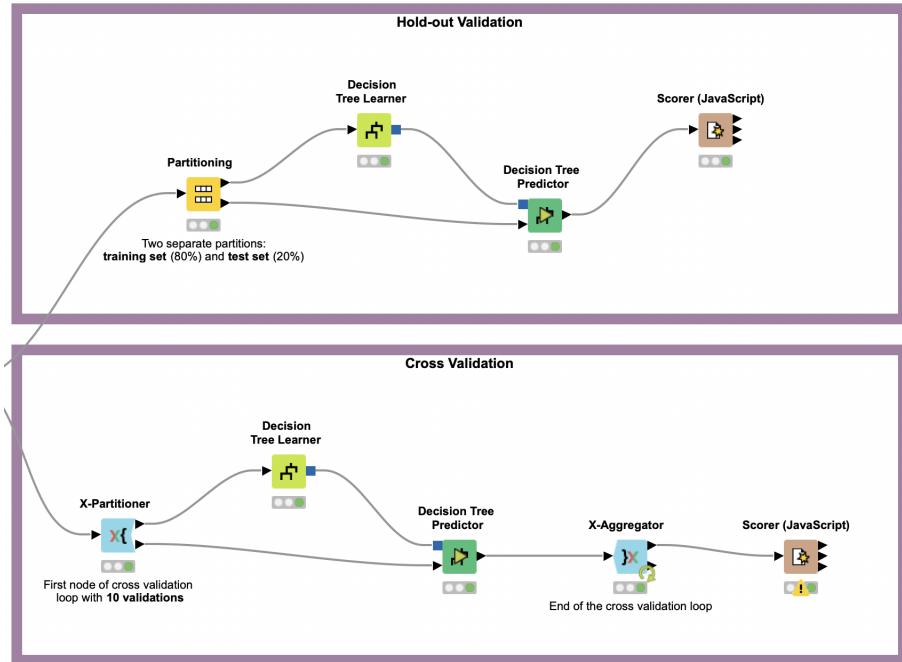


Figura 3.19: *KNIME*: Modelo 2 – *Random Forest* com *Feature Selection*

³Consultar mais informação acerca do segundo modelo na secção 2.7

O resultado da filtragem para *Backward Feature Elimination* e *Forward Feature Selection*, após ter sido estipulado que o limiar inferior de desempenho desejado seria de 0.7, está ilustrado nas Figuras 3.20 e 3.21.

Os resultados observados através do *Scorer* destacam-se pelos elevados valores de *accuracy* obtidos para ambos os modelos: através de *Backward Feature Elimination*, 80,28% para o modelo com validação *hold-out* e 80,13% para o modelo com validação cruzada; e através de *Forward Feature Selection*, 77,18% para o modelo com validação *hold-out* e 77,07% para o modelo com validação cruzada.

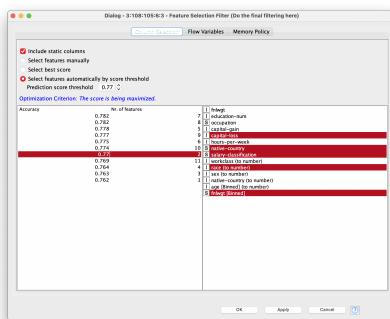
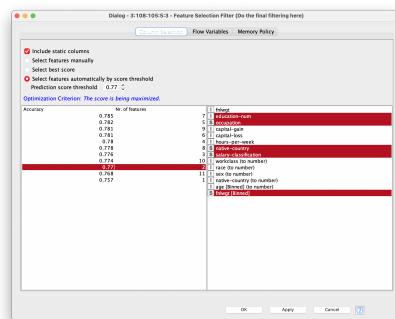


Figura 3.20: *KNIME: Backward Feature Elimination*. Figura 3.21: *KNIME: Forward Feature Selection*.



- Figura 3.21: *KNIME: Forward Feature Selection.*

Na Figura 3.22, está representado o terceiro modelo de ML, baseado numa técnica de regressão logística, recorrendo ao nodo *Logistic Regression Learner*. Numa primeira tentativa, o modelo não apresentava qualquer tratamento adicional de dados e, assim como no modelo aplicado ao *dataset music_genre*, os resultados obtidos pelo *Scorer* revelaram-se baixos, sugerindo uma reanálise do conjunto de dados em mãos e, consequentemente, um tratamento posterior dos dados.

Na secção 3.5, analisou-se que havia uma grande disparidade na distribuição dos valores das variáveis, em que a feature *fnlwgt* apresenta uma escala bastante díspar das restantes. Concluindo que esta disparidade poderia ter contribuído para os resultados baixos obtidos com este modelo, decidiu-se efetuar uma normalização *Min-Max* dos dados de modo a converter a escala de *fnlwght* para uma escala comum entre 0 e 1.

Com estas mudanças, atingiu-se melhores valores em termos de *accuracy*: para o modelo com validação *hold-out*, atingiu-se uma *accuracy* de 75.49%; para o modelo com validação cruzada, obteve-se uma *accuracy* de 70.97% – cf. Anexos B.7 e B.8.

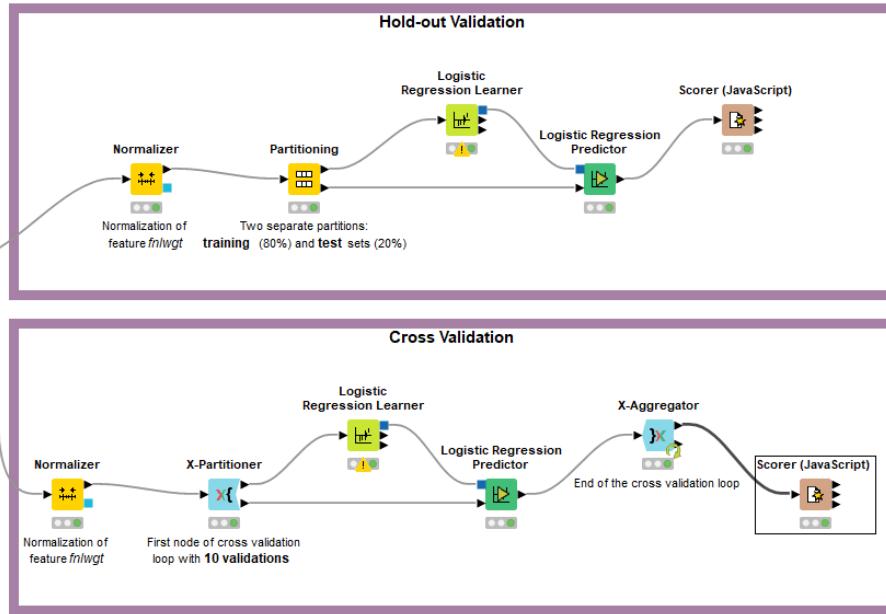


Figura 3.22: KNIME: Modelo 3 – Regressão Logística

O último modelo a apresentar, ilustrado na Figura ??, é idêntico ao aplicado ao *dataset music_genre*⁴ e baseou-se numa estratégia de *clustering*, tendo sido primeiramente efetuada uma normalização *Min-Max* dos dados em que se converteu a escala de *fnlwght* para uma escala comum entre 0 e 1. Como temos em mãos um conjunto de dados que trabalha 2 intervalos do salário que um funcionará pode ganhar, atribui-se o número de *clusters*, *k*, igual a 2.

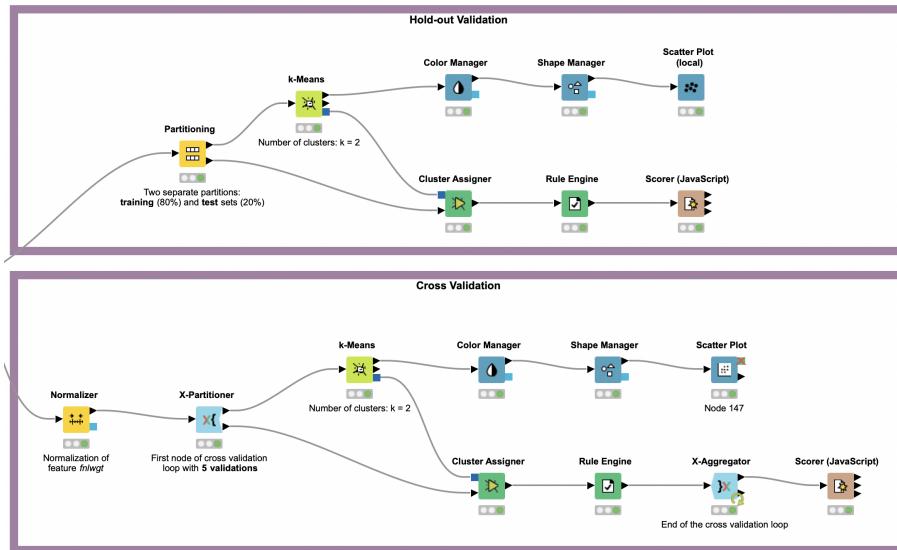


Figura 3.23: KNIME: Modelo 4 – Clustering

A atribuição dos segmentos aos dois valores da *feature salary_classification* encontra-se no *script* do nodo *Rule Engine*, apresentado na Figura 3.24.

⁴Consultar mais informação acerca do quarto modelo na secção 2.7

No que diz respeito aos valores de *accuracy*, estes modelos revelaram resultados bastante baixos: para o modelo com validação *hold-out*, atingiu-se uma *accuracy* de 57,90%; para o modelo com validação cruzada, obteve-se uma *accuracy* de 48,57% – cf. Anexos B.9 eB.10.

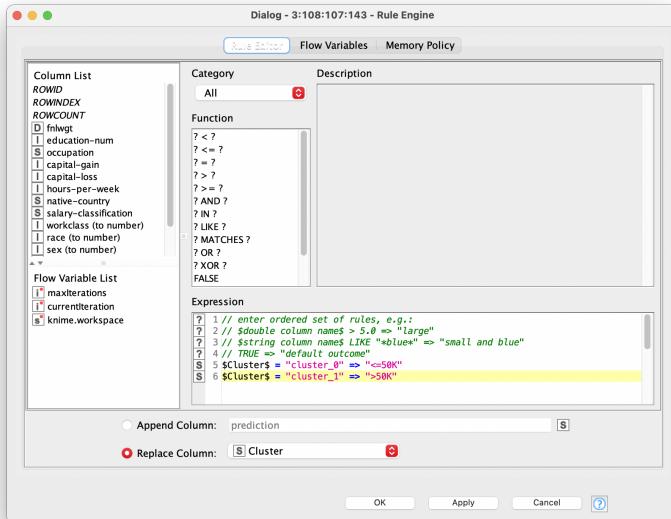


Figura 3.24: *KNIME: Script do Rule Engine*

3.7 Análise dos modelos de *Machine Learning*

Na Tabela 3.1, temos um sumário dos valores registados para a *accuracy* dos modelos de Machine Learning previamente explicados⁵.

Tabela 3.1: Precisão (%) dos modelos de ML desenvolvidos.

Modelo	<i>Hold-out validation</i>	<i>Cross-validation</i>
1	75,25	75,23
2	80,28 (BFE) — 77,18 (FFS)	80,13 (BFE) — 77,07 (FFS)
3	75,49	70,97
4	57,90	48,57

Ao contrário do quarto modelo (*clustering*) que apresenta valores de *accuracy* muito baixos, tanto para o modelo com validação *hold-out* como para com validação cruzada, os outros modelos evidenciam-se pela positiva visto apresentarem resultados satisfatórios, com valores de *accuracy* acima dos 70%. De facto, o segundo modelo (*random forest* com *feature selection*) é o que se destaca mais por apresentar os melhores resultados entre os 4 modelos.

⁵BFS: *Backward Feature Elimination*; FFS: *Forward Feature Selection*

Capítulo 4

Conclusão

O presente projeto objetivou explorar, analisar e preparar dois *datasets* para a criação de modelos de aprendizagem automática. A plataforma de apoio utilizada para o efeito foi o *KNIME*, que providencia uma interface de alto-nível para o desenvolvimento de trabalho no âmbito de ML. O uso desta plataforma permitiu consolidar os conteúdos teóricos abordados, através da sua aplicação num contexto prático.

Tal como mencionado anteriormente, os *datasets* trabalhados foram o *Salary_classification* e *music_genre*. A estrutura do trabalho desenvolvido para ambos os *datasets* centrou-se em: (i) realizar um estudo preliminar do conteúdo dos conjuntos de dados; (ii) tratamento dos dados; (iii) estudo estatístico e correlacional após o tratamento; (iv) criação de modelos de aprendizagem automática.

As tarefas realizadas exigiram domínio dos conteúdos teóricos abordados, assim como algum conhecimento acerca da lógica de negócio inerente ao problema em estudo. Este conhecimento foi essencial, uma vez que todas as decisões adotadas necessitam de ser devidamente fundamentadas, quer em termos de análise/tratamento de dados, quer no que diz respeito à criação do modelos de ML. Esta foi, certamente, uma das tarefas que maior dificuldade acarretou no decurso do projeto, uma vez que uma certa experiência nesta área de trabalho parece ser fundamental para a criação de bons modelos de aprendizagem automática.

O processo de desenvolvimento dos modelos apresentados no decurso do trabalho revelou-se um processo desafiante que exigiu bastante reflexão por parte do grupo e, por vezes, a reconstrução dos modelos ou do tratamento dos dados. Não obstante, o grupo considera que os objetivos estipulados foram alcançados com sucesso, uma vez que foram utilizadas as diversas estratégias abordadas para a criação dos modelos, tendo tal facto culminado, na sua maioria, na criação de modelos de aprendizagem com uma boa performance.

Apêndice A

Music_genre

Hold-out validation											
Confusion Matrix											
	Altern...	Anime	Blues	Classi...	Country	Electr...	Hip-Hop	Jazz	Rap	Rock	
Altern...	658	0	1	0	3	4	20	5	16	78	83.82%
Anime	0	793	0	0	0	0	0	1	0	0	99.87%
Blues	14	1	767	0	5	0	0	17	2	32	91.53%
Classi...	0	2	1	816	0	0	1	1	0	2	99.15%
Country	10	0	6	0	736	0	0	1	1	58	90.64%
Electr...	3	0	0	0	1	800	0	25	0	0	96.50%
Hip-Hop	28	0	2	0	3	2	529	4	257	10	63.35%
Jazz	8	1	14	2	0	26	2	784	0	1	93.56%
Rap	18	0	1	0	9	2	259	4	466	26	59.36%
Rock	58	3	13	0	60	2	5	0	23	610	78.81%
	82.56%	99.13%	95.28%	99.76%	90.09%	95.69%	64.83%	93.11%	60.92%	74.66%	
Overall Statistics											
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified							
85.78%	14.22%	0.842	6959	1154							

Figura A.1: Modelo 1: *Scorer* para *hold-out validation*.

Cross-validation										
Confusion Matrix										
	Altern...	Anime	Blues	Classi...	Country	Electr...	Hip-Hop	Jazz	Rap	Rock
Altern...	3278	4	25	0	40	33	84	27	100	460
Anime	4	4031	1	16	0	4	0	1	0	7
Blues	55	2	3715	2	46	3	3	88	8	124
Classi...	0	19	14	3975	1	0	7	12	0	8
Country	47	1	33	0	3633	4	8	9	34	280
Electr...	22	0	0	0	3	3879	1	115	1	11
Hip-Hop	108	2	5	5	8	6	2654	17	1229	43
Jazz	19	5	78	2	7	133	6	3797	4	13
Rap	115	0	2	0	38	3	1375	8	2390	111
Rock	375	8	112	0	248	15	33	8	94	3207
		81.48%	98.99%	93.22%	99.38%	90.28%	95.07%	63.63%	93.02%	61.92%
		75.21%								
Overall Statistics										
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified						
85.20%	14.80%	0.836	34559	6002						

Figura A.2: Modelo 1: *Scorer para cross-validation.*

Hold-out validation									
Confusion Matrix									
	Alterna...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)
Alterna...	683	0	0	0	2	4	5	27	64
Anime ...	0	793	0	0	0	0	1	0	0
Blues (...)	14	1	767	0	7	0	17	1	31
Classic...	0	2	1	818	0	0	1	1	0
Countr...	9	0	9	0	728	0	1	4	61
Electro...	3	0	0	0	1	800	25	0	0
Jazz (A...	8	1	14	2	0	26	784	2	1
Rap/Hi...	45	0	1	0	15	2	8	1527	22
Rock (...)	61	3	10	1	59	2	0	36	602
		82.99%	99.13%	95.64%	99.63%	89.66%	95.92%	93.11%	95.56%
		77.08%							
Overall Statistics									
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified					
92.47%	7.53%	0.914	7502	611					

Figura A.3: Modelo 2: *Scorer hold-out validation.*

Cross-validation									
Confusion Matrix									
	Altera...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)
Altera...	3297	3	30	0	33	33	27	197	431
Anime ...	4	4031	1	16	0	4	1	0	7
Blues (...)	56	2	3720	1	45	3	87	3	129
Classic...	0	19	14	3977	1	0	12	4	9
Countr...	54	1	21	0	3646	4	9	38	276
Electro...	24	0	0	0	3	3876	115	4	10
Jazz (A...	18	5	79	2	6	133	3796	13	12
Rap/Hi...	216	2	6	1	42	10	13	7668	161
Rock (...)	379	8	110	2	243	15	9	157	3177
	81.45%	99.02%	93.44%	99.45%	90.72%	95.05%	93.29%	94.85%	75.43%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
91.68%	8.32%	0.906	37188	3373

Figura A.4: Modelo 2: *Scorer para cross-validation.*

Hold-out validation									
Confusion Matrix									
	Altera...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)
Altera...	499	4	6	0	119	6	46	66	39
Anime ...	8	744	0	0	0	9	33	0	0
Blues (...)	22	2	661	29	45	0	21	10	48
Classic...	2	3	27	767	6	0	2	2	14
Countr...	58	0	71	0	543	0	7	39	94
Electro...	0	28	0	0	0	794	6	1	0
Jazz (A...	65	44	0	0	6	30	691	2	0
Rap/Hi...	84	0	6	4	16	5	9	1399	97
Rock (...)	72	1	14	5	43	3	2	108	526
	61.60%	90.07%	84.20%	95.28%	69.79%	93.74%	84.58%	85.99%	64.30%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
81.65%	18.35%	0.791	6624	1489

Figura A.5: Modelo 3: *Scorer hold-out validation.*

Cross-validation

Confusion Matrix

	Altera...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)	
Altera...	2505	19	40	0	672	42	224	377	172	61.84%
Anime ...	47	3765	0	0	0	48	204	0	0	92.64%
Blues (...)	94	13	3152	118	257	5	121	51	235	77.90%
Classic...	9	12	175	3742	13	1	17	9	58	92.72%
Countr...	258	7	327	0	2717	6	59	231	444	67.10%
Electro...	0	111	0	0	0	3893	27	1	0	96.55%
Jazz (A...	267	164	3	0	29	149	3439	12	1	84.62%
Rap/Hi...	403	0	36	21	105	17	30	6980	527	85.97%
Rock (...)	437	2	74	39	251	15	20	531	2731	66.61%
62.31%		91.99%	82.79%	95.46%	67.19%	93.22%	83.05%	85.21%	65.52%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
81.17%	18.83%	0.786	32924	7637

Figura A.6: Modelo 3: Scorer para cross-validation.

Hold-out validation

Confusion Matrix

	Altera...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)	
Altera...	6	77	20	113	213	123	18	140	75	0.76%
Anime ...	132	33	46	117	109	55	11	160	131	4.16%
Blues (...)	24	170	54	152	95	40	75	126	102	6.44%
Classic...	585	10	148	3	13	5	19	21	19	0.36%
Countr...	2	160	33	245	175	36	36	121	4	21.55%
Electro...	29	18	8	62	132	123	10	69	378	14.84%
Jazz (A...	129	132	86	64	27	56	50	20	274	5.97%
Rap/Hi...	2	84	3	174	228	996	18	100	15	6.17%
Rock (...)	5	86	22	228	182	38	30	134	49	6.33%
0.66%		4.29%	12.86%	0.26%	14.91%	8.36%	18.73%	11.22%	4.68%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
7.31%	92.69%	-0.042	593	7520

Figura A.7: Modelo 4: Scorer hold-out validation.

Cross-validation									
Confusion Matrix									
	Altera...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)
Altera...	263	685	745	36	582	89	492	752	407
Anime ...	494	526	525	482	394	380	226	675	362
Blues (...)	373	601	780	109	724	216	230	623	390
Classic...	81	40	205	1851	244	1417	19	67	112
Countr...	19	793	967	29	884	175	172	640	370
Electro...	1434	367	395	86	261	90	412	415	572
Jazz (A...	959	249	430	489	552	522	245	201	417
Rap/Hi...	73	1284	931	7	907	49	3607	655	606
Rock (...)	223	781	918	44	726	116	201	738	353
	6.71%	9.88%	13.23%	59.08%	16.76%	2.95%	4.37%	13.74%	9.84%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
13.92%	86.08%	0.031	5647	34914

Figura A.8: Modelo 4: *Scorer* para *cross-validation*.

Hold-out validation									
Confusion Matrix									
	Altera...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)
Altera...	506	12	0	0	98	13	71	83	2
Anime ...	0	746	0	0	0	27	21	0	0
Blues (...)	26	7	580	34	67	0	47	8	69
Classic...	1	3	58	729	5	0	2	2	23
Countr...	55	2	30	1	639	0	28	36	21
Electro...	0	11	0	0	0	818	0	0	0
Jazz (A...	34	32	0	0	0	36	720	16	0
Rap/Hi...	53	0	1	0	23	9	23	1415	96
Rock (...)	82	1	30	2	94	8	21	188	348
	66.84%	91.65%	82.98%	95.17%	69.01%	89.79%	77.17%	80.95%	62.25%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
80.13%	19.87%	0.774	6501	1612

Figura A.9: Modelo 5: *Scorer hold-out validation*.

Cross-validation

Confusion Matrix

	Altera...	Anime ...	Blues (...)	Classic...	Countr...	Electro...	Jazz (P...	Rap/Hi...	Rock (...)	
Altera...	2921	59	14	0	249	54	400	351	3	72.11%
Anime ...	14	3727	0	0	0	159	163	1	0	91.71%
Blues (...)	180	22	2811	120	321	5	169	61	357	69.48%
Classic...	15	14	284	3573	21	3	23	9	94	88.53%
Countr...	405	16	209	1	2841	7	124	283	163	70.17%
Electro...	1	130	0	0	0	3874	27	0	0	96.08%
Jazz (A...	151	174	6	0	6	171	3523	33	0	86.69%
Rap/Hi...	419	0	23	3	91	25	81	6895	582	84.92%
Rock (...)	545	14	160	35	348	26	76	782	2114	51.56%
	62.80%	89.68%	80.15%	95.74%	73.28%	89.59%	76.82%	81.94%	63.81%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
79.58%	20.42%	0.768	32279	8282

Figura A.10: Modelo 5: Scorer para cross-validation.

Hold-out validation

Confusion Matrix

	Altera...	Anime	Blues	Classical	Country	Electro...	Jazz	Rap/Hi...	Rock	
Altera...	660	0	4	0	1	5	6	32	77	84.08%
Anime	0	793	0	1	0	0	0	0	0	99.87%
Blues	13	1	775	0	7	0	17	0	25	92.48%
Classical	0	3	1	817	0	0	1	1	0	99.27%
Country	11	0	3	0	735	0	1	10	52	90.52%
Electro...	2	1	1	0	1	801	21	1	1	96.62%
Jazz	5	1	6	2	1	25	791	3	4	94.39%
Rap/Hi...	47	0	1	0	13	2	7	1510	40	93.21%
Rock	65	3	11	0	54	3	1	36	601	77.65%
	82.19%	98.88%	96.63%	99.63%	90.52%	95.81%	93.61%	94.79%	75.13%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
92.23%	7.77%	0.912	7483	630

Figura A.11: Modelo 6: Scorer hold-out validation.

Cross-validation									
Confusion Matrix									
	Altera...	Anime	Blues	Classical	Country	Electro...	Jazz	Rap/Hi...	Rock
Altera...	3419	4	17	0	23	33	30	171	354
Anime	1	4038	0	13	0	4	1	1	6
Blues	58	2	3739	2	46	3	82	0	114
Classical	0	19	16	3977	1	0	10	4	9
Country	51	0	26	0	3667	4	9	32	260
Electro...	21	0	0	0	1	3880	115	5	10
Jazz	17	6	35	3	9	134	3837	18	5
Rap/Hi...	230	2	4	1	46	9	24	7645	158
Rock	360	9	109	4	242	11	9	157	3199
	82.25%	98.97%	94.75%	99.42%	90.88%	95.14%	93.20%	95.17%	77.74%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
92.21%	7.79%	0.911	37401	3160

Figura A.12: Modelo 6: *Scorer para cross-validation*.

Hold-out validation									
Confusion Matrix									
	Altera...	Anime	Blues	Classical	Country	Electro...	Jazz	Rap/Hi...	Rock
Altera...	676	0	4	0	2	16	14	24	49
Anime	0	777	6	4	0	3	4	0	0
Blues	33	5	680	3	28	10	51	4	24
Classical	1	13	11	773	0	16	7	1	1
Country	24	0	4	0	733	7	13	13	18
Electro...	4	3	6	0	0	808	3	5	0
Jazz	2	3	8	1	0	67	748	9	0
Rap/Hi...	106	1	0	0	15	3	11	1454	30
Rock	138	2	10	0	86	3	4	49	482
	68.70%	96.64%	93.28%	98.98%	84.84%	86.60%	87.49%	93.26%	79.80%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
87.90%	12.10%	0.863	7131	982

Figura A.13: Modelo 7: *Scorer hold-out validation (Backward Feature Elimination)*.

Hold-out validation									
Confusion Matrix									
	Altera...	Anime	Blues	Classical	Country	Electro...	Jazz	Rap/Hi...	Rock
Altera...	676	0	4	0	2	16	14	24	49
Anime	0	777	6	4	0	3	4	0	0
Blues	33	5	680	3	28	10	51	4	24
Classical	1	13	11	773	0	16	7	1	1
Country	24	0	4	0	733	7	13	13	18
Electro...	4	3	6	0	0	808	3	5	0
Jazz	2	3	8	1	0	67	748	9	0
Rap/Hi...	106	1	0	0	15	3	11	1454	30
Rock	138	2	10	0	86	3	4	49	482
	68.70%	96.64%	93.28%	98.98%	84.84%	86.60%	87.49%	93.26%	79.80%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
87.90%	12.10%	0.863	7131	982

Figura A.14: Modelo 7: Scorer hold-out validation (Forward Feature Selection).

Cross-validation									
Confusion Matrix									
	Altera...	Anime	Blues	Classical	Country	Electro...	Jazz	Rap/Hi...	Rock
Altera...	3434	0	28	0	12	82	81	153	261
Anime	2	3966	38	21	2	6	26	1	2
Blues	158	48	3202	26	152	31	288	18	123
Classical	8	90	51	3758	4	22	88	9	6
Country	153	2	22	4	3571	10	94	64	129
Electro...	20	21	24	17	7	3838	69	33	3
Jazz	19	10	26	9	13	276	3678	31	2
Rap/Hi...	547	0	2	2	78	9	47	7258	176
Rock	671	6	86	5	406	11	11	282	2622
	68.52%	95.73%	92.04%	97.81%	84.12%	89.57%	83.93%	92.47%	78.88%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
87.10%	12.90%	0.853	35327	5234

Figura A.15: Modelo 7: Scorer cross-validation (Backward Feature Elimination).

Cross-validation									
Confusion Matrix									
	Altera...	Anime	Blues	Classical	Country	Electro...	Jazz	Rap/Hi...	Rock
Altera...	3434	0	28	0	12	82	81	153	261
Anime	2	3966	38	21	2	6	26	1	2
Blues	158	48	3202	26	152	31	288	18	123
Classical	8	90	51	3758	4	22	88	9	6
Country	153	2	22	4	3571	10	94	64	129
Electro...	20	21	24	17	7	3838	69	33	3
Jazz	19	10	26	9	13	276	3678	31	2
Rap/Hi...	547	0	2	2	78	9	47	7258	176
Rock	671	6	86	5	406	11	11	282	2622
	68.52%	95.73%	92.04%	97.81%	84.12%	89.57%	83.93%	92.47%	78.88%

Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
87.10%	12.90%	0.853	35327	5234

Figura A.16: Modelo 7: *Scorer cross-validation (Forward Feature Selection)*.

Apêndice B

Salary_classification

Hold-Out Validation

Confusion Matrix

		<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)		6333	1097	85.24%
>50K (Actual)		1317	1005	43.28%
		82.78%	47.81%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
75.25%	24.75%	0.295	7338	2414

Figura B.1: Modelo 1: Scorer para hold-out validation.

Cross Validation

Confusion Matrix

		<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)		31292	5466	85.13%
>50K (Actual)		6505	5060	43.75%
		82.79%	48.07%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
75.23%	24.77%	0.298	36352	11971

Figura B.2: Modelo 1: Scorer para cross validation.

Hold-Out Validation

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	10653	469	95.78%
>50K (Actual)	2869	637	18.17%
	78.78%	57.59%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
77.18%	22.82%	0.182	11290	3338

Forward Feature Selection

Figura B.3: Modelo 2: Scorer para hold-out validation – Forward Feature Selection.

Hold-Out Validation

Backward Feature Elimination

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	10464	658	94.08%
>50K (Actual)	2227	1279	36.48%
	82.45%	66.03%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
80.28%	19.72%	0.361	11743	2885

Figura B.4: Modelo 2: Scorer para hold-out validation – Backward Feature Elimination.

Cross Validation

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	35432	1644	95.57%
>50K (Actual)	9536	2145	18.36%
	78.79%	56.61%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
77.07%	22.93%	0.181	37577	11180

Forward Feature Selection

Figura B.5: Modelo 2: Scorer para cross validation – Forward Feature Selection.

Cross Validation

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	35072	2004	94.59%
>50K (Actual)	7683	3998	34.23%
	82.03%	66.61%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
80.13%	19.87%	0.346	39070	9687

Backward Feature Elimination

Figura B.6: Modelo 2: *Scorer* para *cross validation* – Backward Feature Elimination.

Hold-Out Validation

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	7362	0	100.00%
>50K (Actual)	2390	0	0.00%
	75.49%	undefined	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
75.49%	24.51%	0.000	7362	2390

Figura B.7: Modelo 3: *Scorer* para *hold-out validation*.

Cross Validation

Confusion Matrix

	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	30008	7068	80.94%
>50K (Actual)	7085	4596	39.35%
	80.90%	39.40%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
70.97%	29.03%	0.203	34604	14153

Figura B.8: Modelo 3: *Scorer* para *cross validation*.

Hold-Out Validation

Confusion Matrix

		<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)		4650	2780	62.58%
>50K (Actual)		1326	996	42.89%
		77.81%	26.38%	
Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
57.90%	42.10%	0.045	5646	4106

Figura B.9: Modelo 4: Scorer para hold-out validation.

Cross Validation

Confusion Matrix

		<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)		17685	19391	47.70%
>50K (Actual)		5686	5995	51.32%
		75.67%	23.62%	
Overall Statistics				
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
48.57%	51.43%	-0.007	23680	25077

Figura B.10: Modelo 4: Scorer para cross validation.