

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
LICENCIATURA EM ENGENHARIA INFORMÁTICA

Aprendizagem e decisão inteligentes - Trabalho Prático
Ano Letivo 2021/2022
Grupo 50

Gonçalo Braz (a93178) Simão Cunha (a93262)
Tiago Silva (a93277) Gonçalo Pereira (a93168)

15 de junho de 2022

1 Introdução

O presente relatório surge no âmbito da UC de Aprendizagem e Decisão Inteligentes, onde nos foi proposto a conceção e desenvolvimento de um projeto utilizando os modelos de aprendizagem abordados ao longo do semestre. O *software* usado no trabalho prático foi o **Knime** [1], que é uma plataforma de análise de dados usada para automatizar o processo de *data science*.

Iremos efetuar neste relatório uma contextualização do problema, relatar o trabalho incidido em cada *dataset* - como o tratamento de *features*, o tratamento dos dados, os modelos de aprendizagem adotados e uma análise dos resultados obtidos - e uma conclusão, onde falaremos sobre as dificuldades encontradas ao longo do projeto e sobre o trabalho futuro que poderá ser efetuado sobre estes conjuntos de dados.

2 Contextualização do problema

Na **1ª fase** do trabalho prático, foi-nos pedido para:

- Escolher um *dataset* disponíveis em *websites* como o *Google Dataset Search*, o *Kaggle* o a *UCI Machine Learning Repository*;
- Identificar qual o *dataset* que a equipa docente atribui ao nosso grupo;
- Explorar, analisar e preparar ambos os *datasets*, procurando extrair conhecimento relevante no contexto dos problemas em questão.

Já na **2ª** e última **fase** do projeto, foi-nos requisitada as seguintes tarefas:

- Conceção de modelos de aprendizagem para ambos os problemas inerentes aos *datasets*;
- Análise crítica dos resultados obtidos;
- Interpretação dos resultados adquiridos e definição da sua utilidade no contexto dos problemas subjacentes aos datasets trabalhados. Determinar e explicitar os resultados mais relevantes
- Quais os domínios a tratar, quais os objetivos a alcançar e como os atingir;
- Qual a metodologia seguida e como foi aplicada;
- Descrição e exploração detalhada de ambos os datasets e do tratamento de dados efetuado;
- Descrição dos modelos desenvolvidos e quais as suas características, parâmetros de treino, entre outros detalhes que enriqueçam a explicação;
- Sumário dos resultados obtidos e respetiva análise crítica;
- Apresentação de sugestões e recomendações após análise dos resultados obtidos e dos modelos desenvolvidos.

Tendo em conta que o nosso grupo é o número 50, coube-nos como *dataset* dos fornecidos pelos docentes, o **USA house regression** (secção 3).

Para o segundo *dataset*, após devidas deliberações entre o grupo, escolhemos um que guarda uma grande quantidade de informação sobre diversos filmes, **IMBD movie dataset** (secção 4).

3 *Dataset* - USA house regression

O *dataset* atribuído pela equipa docente foi o que contém dados relativos a habitações de uma certa região dos Estados Unidos da América. O *target* deste conjunto de dados é inferir acerca do preço de habitações nesta mesma região norte-americana.

3.1 Descrição do Data Set

Este ficheiro de dados contém 5000 habitações, cada uma com os seguintes atributos:

Atributo	Descrição
<i>avg_area_income</i>	rendimento médio dos habitantes da cidade onde a casa está localizada
<i>avg_area_house_age</i>	idade média das casas da mesma cidade que a casa em questão
<i>avg_area_number_of_rooms</i>	número médio de divisões nas casas da mesma cidade que a casa em questão
<i>avg_area_number_of_bedrooms</i>	número médio de quartos nas casas da mesma cidade que a casa em questão
<i>area_population</i>	área populacional da cidade
<i>price</i>	preço a que a casa foi vendida
<i>address</i>	morada da casa

3.2 Tratamento de *features*

O primeiro passo a efetuar é a seleção de *features* a manter para podermos extrair o máximo de conhecimento possível. Notamos que uma das *features* era do tipo nominal - **address**, que contém o estado norte-americano e a rua da habitação. Através de um tratamento destes valores, com uma separação do estado do resto do endereço, ficamos só com a informação destes e, posteriormente, transformamos estes valores em numéricos de modo a se tornar mais facilmente digerível pelos algoritmos.

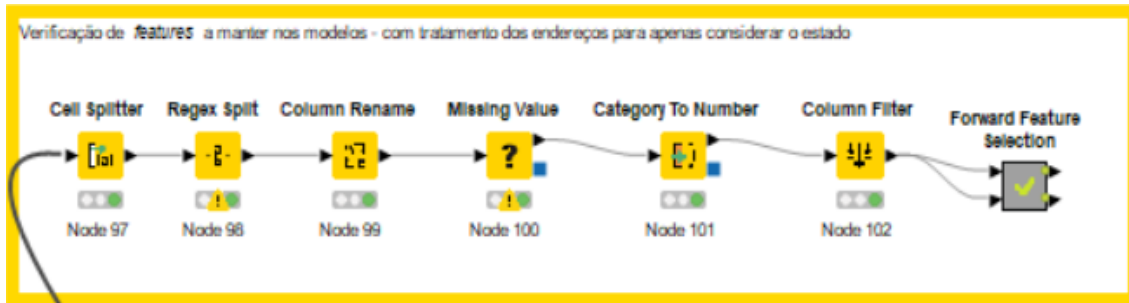


Figura 1: Pipeline que extrai o estado de cada habitação, numerando-os

O nodo *Forward Feature Selection* é utilizado para através de diversas iterações inferir os atributos que melhor contribuem para o modelo, neste caso, de regressão linear.

Os resultados obtidos foram os seguintes:

File

Column Selection Flow Variables Memory Policy

☒ Include static columns
☒ Select features manually
☐ Select best score
☐ Select features automatically by score threshold
Prediction score threshold

Optimization Criterion: The score is being maximized.

R ²	Nr. of features	Features
0,922	5	Avg. Area Income, Avg. Area House Age, Avg. Area Number of Rooms, Avg. Area Number of Bedrooms, Area Population
0,92	4	Avg. Area Income, Avg. Area House Age, Avg. Area Number of Rooms, Price
0,918	6	Avg. Area Income, Avg. Area House Age, Avg. Area Number of Rooms, Avg. Area Number of Bedrooms, Area Population, Price
0,793	3	Avg. Area Income, Avg. Area House Age, Price
0,61	2	Avg. Area Income, Price
0,409	1	State (#1)

Figura 2: Feature Selector no *house dataset*

Como se observa, o atributo que não esteve presente na melhor iteração foi o dos endereços/estados. Todos os outros demonstraram ser benéficos para o estudo do *dataset*.

3.3 Metodologia e casos de estudo

Depois de decidirmos que *features* devemos manter, passamos a decidir os diferentes casos de observação, assim como a metodologia para obter os resultados. Para cada caso de estudo, começaremos por explicar o que se procura concluir/explorar, seguido pelo tratamento de dados necessário o e, por fim, uma apresentação dos resultados obtidos.

Os modelos de aprendizagem que aprofundaremos neste *dataset* serão os que mais foram abordados nas aulas, ou seja, regressão linear, árvores de decisão e redes neurais.

Assim, os casos de estudo a relatar, são:

1. Estudo simples dos resultados - **Decision Tree** (3.3.1);

2. Tratamento dos *outliers* - Decision Tree (3.3.2);
3. Análise da relação entre o preço e o estado - Decision Tree (3.3.3);
4. Estudo com todos as *features binned* - Decision Tree (3.3.4);
5. Estudo simples dos resultados - Linear Regression (3.3.5);
6. Tratamento dos *outliers* - Linear Regression (3.3.6);
7. Análise da relação entre o preço e o estado - Linear Regression (3.3.7);
8. Estudo simples dos resultados - Redes neuronais (3.3.8);

3.3.1 Estudo simples dos resultados - Decision Tree

O primeiro modelo de aprendizagem automática abordado foi o de árvores de decisão. Este método não consegue prever valores numéricos, pelo que a *feature* que se pretende inferir terá de vir sob a forma de *string* (nominal).

Além disso, é de notar que o resultado previsto será portanto um intervalo de preço, e não um valor contínuo, sendo que, portanto, no caso de se pretender prever o valor exato, terá de se utilizar um outro modelo, como exploraremos nas secções posteriores.

A transformação da *feature* alvo, *price*, deu-se através de um *binning*, cujo melhor resultado veio com um valor de 5 bins. Estes bins são depois passados por um *rule engine* que altera os seus valores para nomes que permitem uma análise mais simples, de "low price", até "very high".

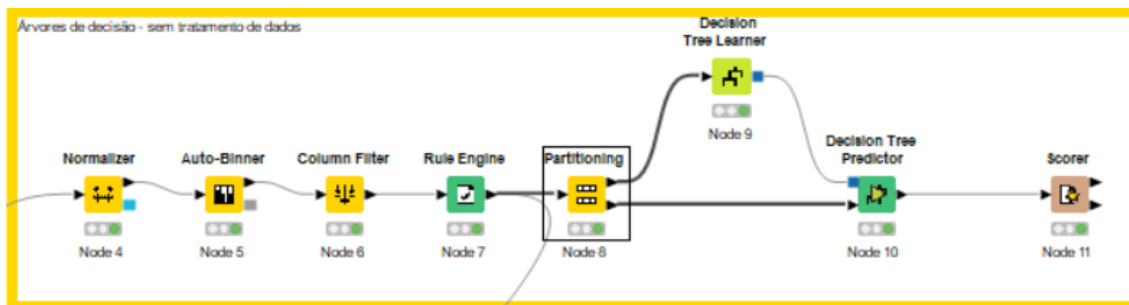


Figura 3: Estudo simples dos resultados - Decision Tree

3.3.1.1 Tratamento de dados

Os valores para aprendizagem são normalizados, onde, de seguida, o atributo-alvo é passado pelo **Auto-binner**, com valor de 5 bins. Depois, foi efetuada uma partição dos dados, onde 80% serão destinados para aprendizagem e os restantes 20% serão para testes.

3.3.1.2 Resultados

Através da análise dos resultados do nodo **Scorer**, mais propinquo na matriz de confusão, podemos observar que a *accuracy* deste modelo é de 54.7% e o erro é de 45.3%. Para uma aproximação inicial do modelo, este resultado com pouco tratamento é positivo e demonstra que existe uma boa correlação entre atributos, permitindo extrair o conhecimento pretendido a partir deles.

Price [Binned] \ Prediction (Price [Binned])	medium low	high	medium	low	very high
medium low	89	8	57	48	1
high	14	79	43	0	46
medium	42	49	95	4	7
low	54	1	4	134	0
very high	2	62	11	0	150

Correct classified: 547
 Accuracy: 54,7%
 Cohen's kappa (κ): 0,434%

Wrong classified: 453
 Error: 45,3%

Figura 4: Estudo simples - Scorer

3.3.2 Tratamento dos *outliers* - Decision Tree

O próximo caso de estudo é muito semelhante ao anterior, mas aqui tratamos dos *outliers*, que são os dados que dispersam muito do contexto da generalidade do *dataset*. Tentamos, assim, verificar se, através de um tratamento das entradas com dados que fogem do padrão natural, terá um bom resultado nas previsões.

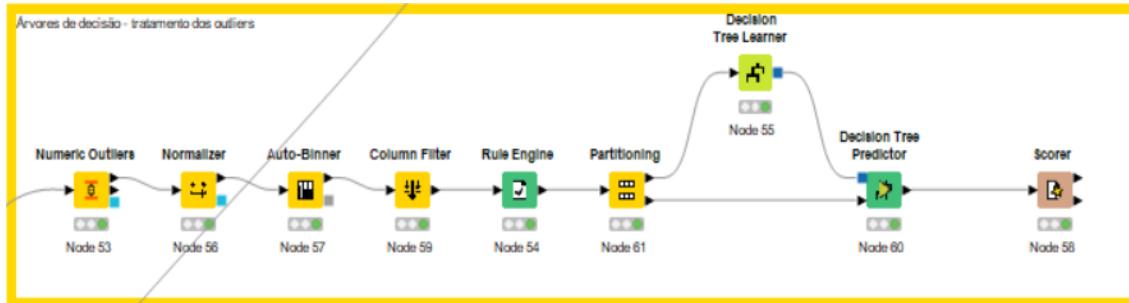


Figura 5: Tratamento dos outliers - Decision Tree

3.3.2.1 Tratamento de dados

O processo de tratamento de dados foi igual à da secção 3.3.1. No entanto, efetuamos o tratamento dos dados *outliers* com recurso ao nodo *Numeric Outliers*. Os valores dos *outliers* são substituídos pelo primeiro valor aceitável.

3.3.2.2 Resultados

Através da análise dos resultados do nodo *Scorer*, mais proximante na matriz de confusão, podemos observar que a *accuracy* deste modelo é de 55,3% e o erro é de 44,7%. O acerto do modelo melhorou, mas apenas ligeiramente.

Price [Binned] \ Prediction (Price [Binned])	medium low	high	medium	low	very high
medium low	96	13	54	39	2
high	13	87	53	1	37
medium	56	55	68	9	8
low	39	1	10	147	0
very high	0	54	3	0	155

Correct classified: 553
 Accuracy: 55,3%
 Cohen's kappa (κ): 0,441%

Wrong classified: 447
 Error: 44,7%

Figura 6: Tratamento de outliers - Scorer

3.3.3 Análise da relação entre o preço e o estado - Decision Tree

Embora durante a análise de *features*, obtivemos resultados que indicavam que o atributo *address* não era incluído no modelo com melhor resultado de previsão. Neste caso, decidimos manualmente, seguindo os resultados dos anteriores estudos, verificar se realmente isto se aplica.

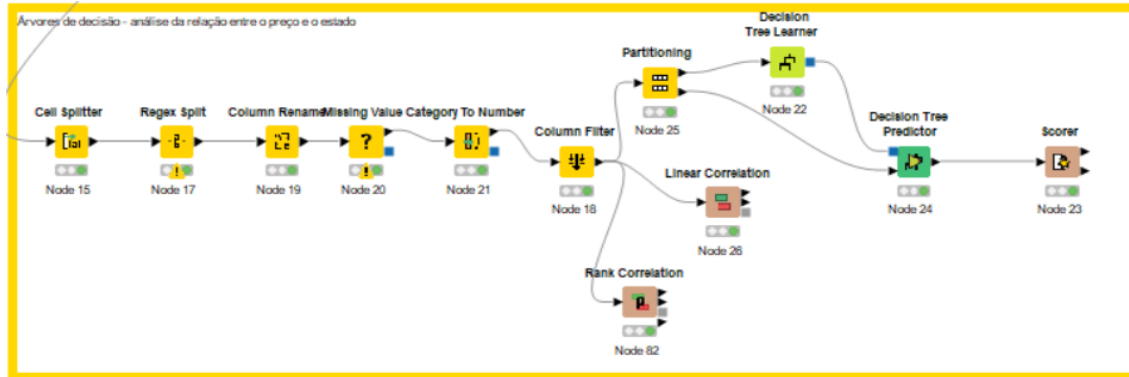


Figura 7: Análise da relação entre o preço e o estado - Decision Tree

3.3.3.1 Tratamento de dados

O tratamento de dados segue o do estudo 3.3.1, seguido de uma transformação do atributo *address* em inteiros, tal como no seleção de features (3.2).

3.3.3.2 Resultados

Através da análise dos resultados do nodo **Scorer**, mais propriamente na matriz de confusão, podemos observar que a *accuracy* deste modelo é de 54.4% e o erro é de 45.6%. Os resultados condizem com os obtidos no nodo *Forward Feature Selector*, não tendo o modelo melhorado, mas também não piorou significativamente.

Price [Binned] \ Prediction (Price [Binned])	medium low	high	medium	low	very high
medium low	88	10	55	42	1
high	11	103	58	0	29
medium	59	42	74	6	13
low	59	2	9	138	0
very high	1	47	12	0	141
Correct classified: 544			Wrong classified: 456		
Accuracy: 54,4%			Error: 45,6%		
Cohen's kappa (κ): 0,43%					

Figura 8: Relação entre o preço e o estado - Scorer

3.3.4 Estudo com todos as *features binned* - Decision Tree

O último estudo de árvores de decisão consiste na análise do efeito de *bin* de todos as *features* passadas para os nodos de aprendizagem.

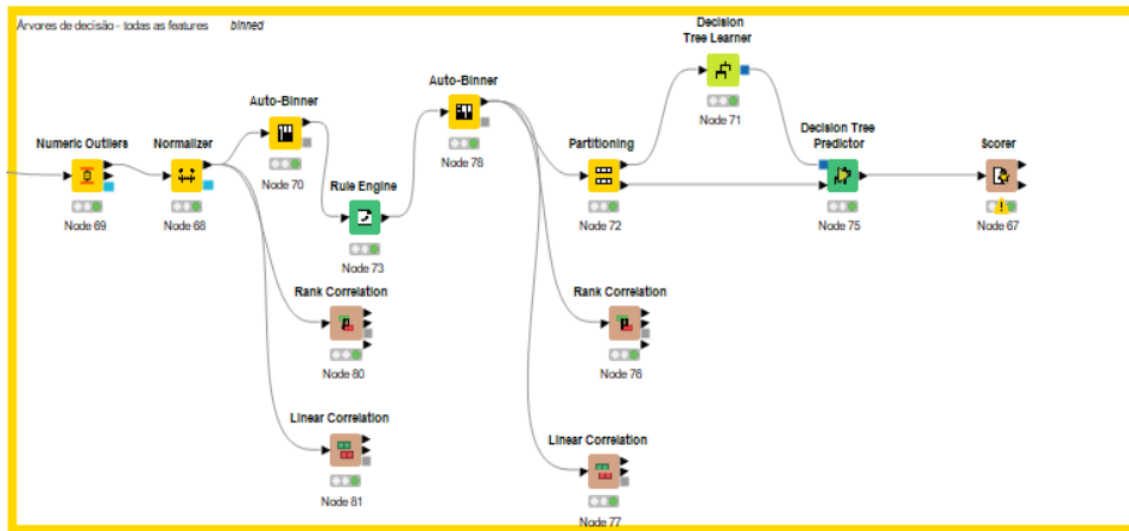


Figura 9: Estudo com todos as features binned - Decision Tree

3.3.4.1 Tratamento de dados

O tratamento de dados segue o do caso de tratamento de outliers, sendo a única diferença que no nodo de *Auto-Binner*, todos os atributos levam *bin*.

3.3.4.2 Resultados

Através da análise dos resultados do nodo **Scorer**, mais propriamente na matriz de confusão, podemos observar que a *accuracy* deste modelo é de 59.577% e o erro é de 40.423%. Este foi o melhor resultado obtido até ao momento, provavelmente por ser mais simples para uma árvore de decisão trabalhar com valores nominais estilo categóricos.

⚠ There were missing values in the reference or in the prediction class columns.					
Price \ Prediction (Price)	medium	high	low	medium low	very high
medium	319	49	1	61	1
high	91	109	0	2	16
low	4	0	14	30	0
medium low	97	1	21	129	0
very high	2	25	0	0	20
Correct classified: 591			Wrong classified: 401		
Accuracy: 59,577%			Error: 40,423%		
Cohen's kappa (κ): 0,401%					

Figura 10: Features binned - Scorer

3.3.5 Estudo simples dos resultados - Linear Regression

Os modelos de aprendizagem de regressão linear trabalham com alvos contínuos, como é caso neste trabalho. Consequentemente, espera-se que os resultados obtidos sejam melhores do que para as árvores de decisão.

Para começar, procuramos observar os resultados de mais simples obtidos por este modelo.

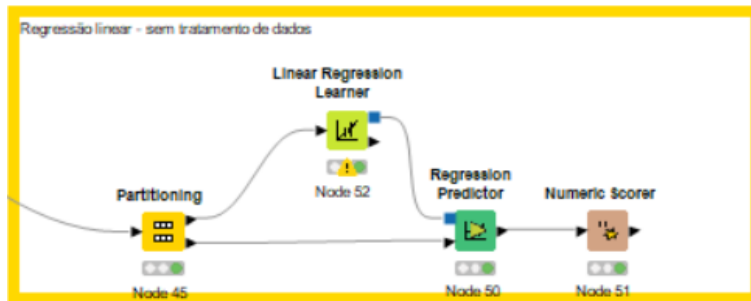


Figura 11: Caso simples - Linear Regression

3.3.5.1 Tratamento de dados

Como a coluna alvo já é um valor contínuo, o único tratamento de dados necessário foi a partição dos mesmos, numa relação de 80/20, que foi a que obteve melhores resultados.

3.3.5.2 Resultados

Statistics - 0:51 - Numeric Scorer	
File	
R^2 :	0,919
Mean absolute error:	79 221,407
Mean squared error:	9 868 786 005,121
Root mean squared error:	99 341,764
Mean signed difference:	264,993
Mean absolute percentage error:	0,073
Adjusted R^2 :	0,919

Figura 12: Estudo simples - Scorer

Sabendo que o objetivo é aproximar o valor de R^2 para 1 e o valor do erro para 0, obtivemos, de acordo com a informação obtida através do nó **Scorer**, que o valor de R^2 é 0.919 e o erro médio é de cerca de 80 mil unidades monetárias. Embora este valor parece extremamente elevado, relativizando com a escala dos valores do preço, que variam entre as várias centenas de milhares a milhões, este erro não é tão significativo como parece numa primeira análise.

3.3.6 Tratamento dos *outliers* - Linear Regression

Seguindo o mesmo pensamento que para o estudo das árvores de decisão, procuramos verificar a consequência do tratamento dos *outliers* na regressão linear.

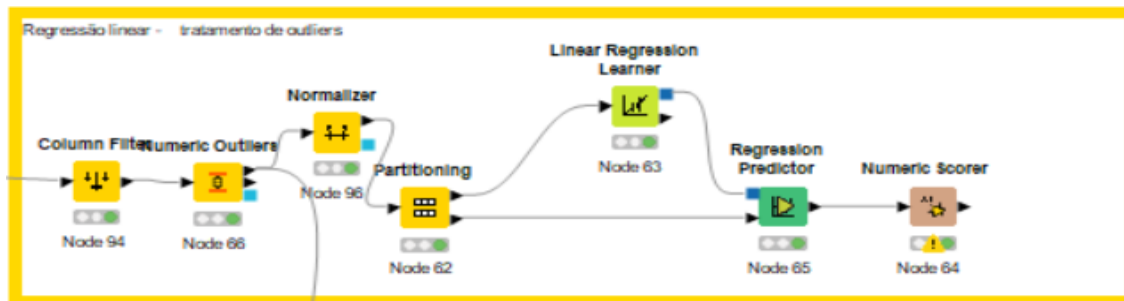


Figura 13: Tratamento dos outliers - Linear Regression

3.3.6.1 Tratamento de dados

Para o tratamento de dados apenas acrescentamos o tratamento de *outliers* que, como anteriormente, substituí os *outliers* pelo valor aceitável mais próximo e, normalizámos os dados por desta forma obtermos melhores resultados.

3.3.6.2 Resultados

Sabendo que o objetivo é aproximar o valor de R^2 para 1 e o valor do erro para 0, obtivemos, de acordo com a informação obtida através do nodo **Scorer**, que o valor de R^2 é 0.922 e o erro médio é 0.044. Podemos que o tratamento de outliers consistentemente melhorou os resultados do modelo, tanto neste caso como em 3.3.2.

Statistics - 0:64 - Numeric Scorer	
File	
Can't calculate Mean Absolute Percentage error: target value is 0! Row1799	
R ² :	0,922
Mean absolute error:	0,044
Mean squared error:	0,003
Root mean squared error:	0,054
Mean signed difference:	0,001
Mean absolute percentage error:	NaN
Adjusted R ² :	0,922

Figura 14: Tratamento de outliers - Scorer

3.3.7 Análise da relação entre o preço e o estado - Linear Regression

O último caso de estudo sobre regressão linear, trata da relação entre o preço e o estado, com a expectativa que talvez este modelo consiga lidar melhor com este atributo do que as árvores de decisão.

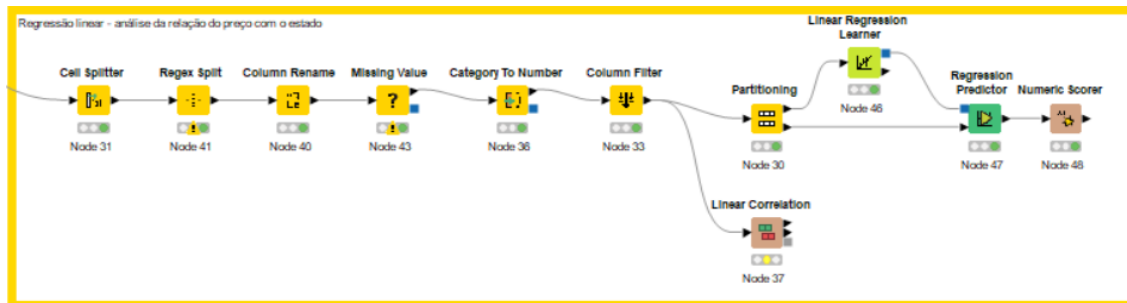


Figura 15: Análise da relação entre o preço e o estado - Linear Regression

3.3.7.1 Tratamento de dados

O tratamento dos endereços foi equivalente ao feito anteriormente. O resto dos dados não sofreram alterações.

3.3.7.2 Resultados

Sabendo que o objetivo é aproximar o valor de R^2 para 1 e o valor do erro para 0, obtivemos, de acordo com a informação obtida através do nodo **Scorer**, que o valor de R^2 é 0.919 e o erro médio é de pouco mais de 80 mil. Embora o valor de R^2 tenha sido semelhante ao do estudo simples, o valor do erro piorou ligeiramente, demonstrando novamente que este atributo negativamente influencia o modelo, embora não com grande significância.

Statistics - 0:48 - Numeric Scorer	
File	
R^2 :	0,919
Mean absolute error:	80 387,043
Mean squared error:	9 920 394 401,264
Root mean squared error:	99 601,177
Mean signed difference:	-1 512,804
Mean absolute percentage error:	0,075
Adjusted R^2 :	0,919

Figura 16: Relação entre o preço e o estado - Scorer

3.3.8 Estudo simples dos resultados - Redes neurais

Como último estudo deste *dataset*, de forma a testar o desempenho entre os dois modelos de previsão contínua, utilizámos os nodos de *RProp MLP* para comparar com os de regressão linear no caso de mínimo tratamento de dados.

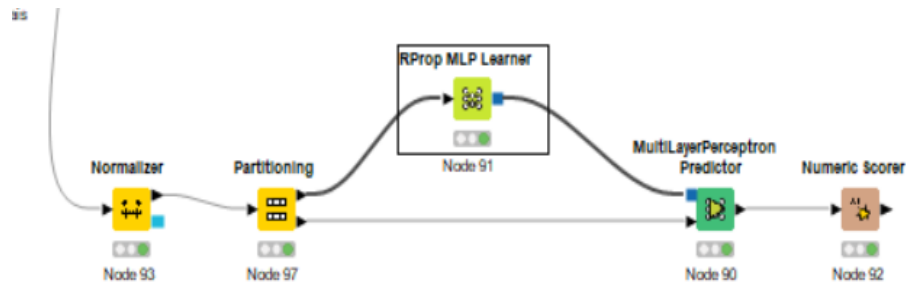


Figura 17: Redes Neurais

3.3.8.1 Tratamento de dados

Para os dados serem processados pelas redes neurais, estes têm de ser todos numéricos e normalizados, por isto, retiramos a coluna de endereço e normalizamos todas as outras.

3.3.8.2 Resultados

Statist...	
File	
R ² :	0,91
Mean absolute error:	0,034
Mean squared error:	0,002
Root mean squared error:	0,042
Mean signed difference:	-0,001
Mean absolute percentage error:	0,081
Adjusted R ² :	0,91

Figura 18: Redes neuronais - Scorer

Colocando os valores das redes neuronais do seguinte modo, 200 iterações, 10 camadas, 15 neurónios por camada, obtemos os resultados acima, como por exemplo o valor de R^2 de 0.91. Este resultado, embora inferior ao da regressão linear, continua bastante satisfatório.

Um aumento substancial dos recursos disponíveis para as redes, 1000 iterações, 20 camadas e 20 neurónios por camada, ao contrário do esperado, diminui o valor de R^2 para 0.883, aumentando o erro.

3.4 Análise dos resultados

Com os estados de estudo que realizamos, conseguimos determinar algumas conclusões.

No caso do alvo de preço, por ser um valor contínuo, os melhores modelos foram, como se poderia esperar, a regressão linear e as redes neuronais, sendo que mesmo utilizando intervalos de preço, o melhor resultado obtido pelas árvores de decisão foi de cerca de 60%. Nos outros dois modelos a taxa de acerto foi bastante superior, com erros médios inferiores a 80 mil dólares.

Podemos afirmar então que dos modelos estudados, o que trouxe maior confiança foi o de regressão linear e que, das informações oferecidas pelo *dataset*, o endereço foi algo da qual os modelos de aprendizagem não conseguiram extrair conhecimento que conseguisse relacionar com o valor do preço das casas correspondentes.

4 Dataset - IMBD movies dataset

O *dataset* escolhido pelo nosso grupo foi um de filmes do IMDB ([2]) - que é a fonte mais popular do mundo para filmes, televisão e de celebridades desta área - que pode ser descarregado em [3]. A nossa decisão proveio do facto de, além de este ser um tópico que achamos interessante, a quantidade de entradas providas por este *dataset* ser considerável, com cerca de 45 mil entradas no ficheiro *csv* de interesse, e tendo ainda outros 5 *csv* com informação extra, não tendo sido considerados para este estudo. Os outros ficheiros *csv* foram omitidos por terem informação irrelevante para o ficheiro principal. Após estudo dos dados, como é o caso *ratings_small.csv*, detetámos discrepâncias nos resultados dos mesmos atributos do caso de estudo.

4.1 Descrição do Data Set

O ficheiro de dados estudado foi o *movie_metadata.csv* que apresenta informação sobre mais de 45000 filmes, com a seguinte estrutura:

Atributo	Descrição
<i>adult</i>	<i>boolean</i> indicador se o filme é de conteúdo adulto
<i>belongs_to_collection</i>	informação sobre a coleção de filmes a que o filme pertence, se pertencer
<i>budget</i>	orçamento utilizado para a criação do filme
<i>genres</i>	categorias do filme
<i>homepage</i>	<i>link</i> para uma página <i>web</i> do filme
<i>id</i>	<i>id</i> do filme no <i>dataset</i>
<i>imdb_id</i>	<i>id</i> do filme no IMDB
<i>original_language</i>	língua original de produção do filme
<i>original_title</i>	título original do filme
<i>overview</i>	descrição/sinopse do filme
<i>popularity</i>	valor que representa a popularidade
<i>poster_path</i>	<i>link</i> para o poster do filme
<i>production_companies</i>	lista com o conjunto de empresas que participaram na realização do filme
<i>production_countries</i>	conjunto dos países em que o filme foi produzido/filmado
<i>release_date</i>	data de estreia do filme
<i>revenue</i>	quantidade de dinheiro ganha na bilheteria
<i>runtime</i>	tempo de duração do filme
<i>spoken_languages</i>	lista de todas as línguas em que o filme está disponível
<i>status</i>	estado de lançamento do filme
<i>tagline</i>	<i>slogan</i> do filme
<i>title</i>	nome do filme
<i>video</i>	
<i>vote_average</i>	valor médio de votação do filme
<i>vote_count</i>	número de votações feitas do filme

O objetivo do estudo é, através do estudo e tratamento do *dataset*, verificar os resultados de diferentes nodos de aprendizagem automática na previsão dos resultados do atributo *vote_average*. Este é um valor numérico contínuo.

Utilizando o nodo *CSV Reader* para a leitura do *dataset*, observamos que algumas linhas deste são inválidas. Assim sendo, e após variar o número de linhas, acabamos com um total de 10000 entradas válidas. Este número não constitui a totalidade de entradas válidas no *csv*, porém, apresenta um caso de tamanho razoável.

4.2 Tratamento de features

Antes de passarmos ao tratamento de dados, começamos por verificar o interesse dos diversos atributos disponibilizados pelo *dataset*.

Para isto utilizamos o nodo **Forward Feature Selection** de modo a que, através de seguidas combinações dos diferentes atributos utilizando um **Linear Regression Learner**, consigamos ter uma ideia de quais atributos a retirar e quais terão melhor influência.

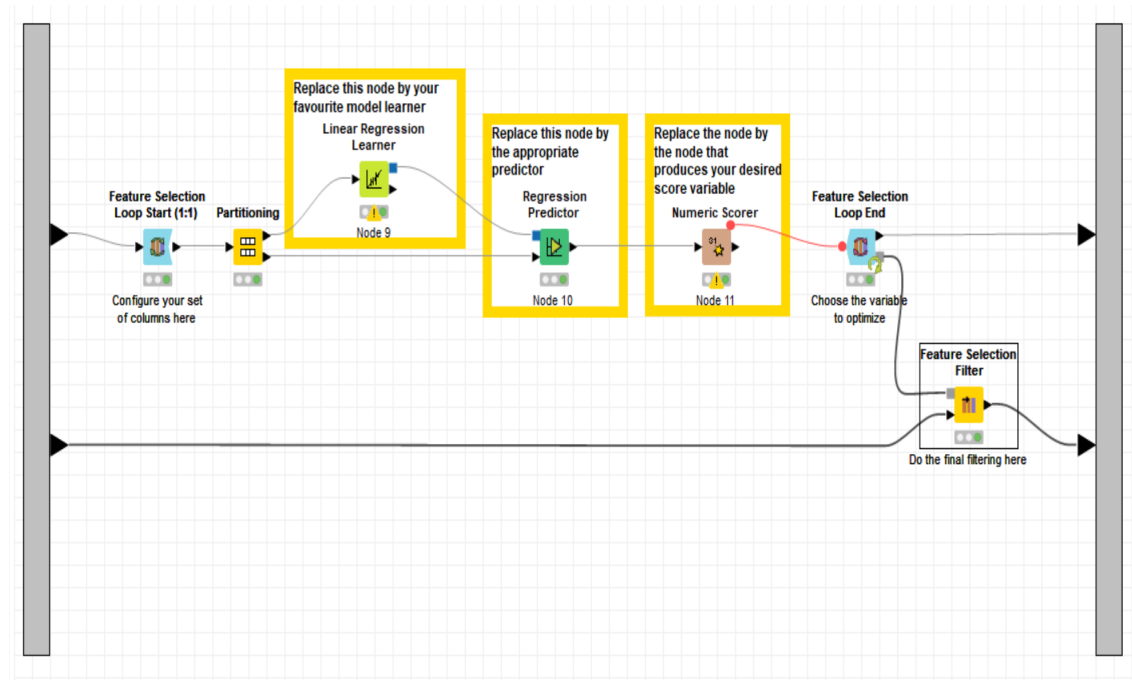


Figura 19: Feature Selector no *movie dataset*

Tendo em conta que alguns dos atributos são complexos, como é exemplo as listas de categorias, utilizamos dois **Forward Feature Selection**: um com estes atributos e outro sem eles para podermos comparar os resultados. No final, concluímos que os atributos complexos deveriam ser manualmente estudados.

Assim sendo, os resultados obtidos sobre os outros atributos no **Forward Feature Selection** foram:

R^2	Nr. of features		
		S	adult
		D	budget
	0,296	S	homepage
	0,286	D	id
	0,279	S	imdb_id
	0,276	S	original_language
	0,274	S	original_title
	0,27	D	popularity
	0,267	D	revenue
	0,257	D	runtime
	0,232	S	status
	0,214	D	vote_average
	0,213	D	vote_count

Figura 20: Resultados do Feature Selector no *movie dataset*

Analisando os resultados, decidimos excluir os seguintes atributos: *adult*, *homepage*, *imdb_id*, *original_title*, *status*.

Estes atributos foram naturalmente removidos por diversos motivos, tais como o facto de serem únicos para todas as linhas como *imdb_id*, ou, então, o facto de não variarem nenhuma vez, como o *adult*.

Embora já tivéssemos intenções de retirar estes atributos - e estes resultados confirmaram esta decisão - decidimos não seguir totalmente os resultados obtidos nas seguintes *features*: *id*, visto ser conceptualmente igual ao *imdb_id*; *popularity* e *revenue* por serem bons indicadores do sucesso de um filme.

4.3 Metodologia e casos de estudo

De forma análoga ao procedimento no *dataset* da secção 3, estabelecemos quais os casos de estudo a abordar:

1. Estudo simples dos resultados - **Linear Regression** (4.4.1)
2. Arredondamento dos average scores - **Linear Regression** (4.4.2)
3. Binned average scores - **Linear Regression** (4.4.3)
4. Empresas de produção - **Linear Regression** (4.4.4)
5. Categorias de filmes - **Linear Regression** (4.4.5)
6. Estudo simples dos resultados - **Decision Tree** (4.4.6)
7. Empresas de produção - **Decision Tree** (4.4.7)
8. Categorias de filmes - **Decision Tree** (4.4.8)
9. Estudo simples dos resultados - **RProp MLP** (4.4.9)
10. Arredondamento dos average scores - **RProp MLP** (4.4.10)
11. Binned average scores - **RProp MLP** (4.4.11)

12. Categorias de filmes - RProp MLP (4.4.12)

13. Outros learners (4.4.13)

Os *learners* em que nos iremos concentrar serão aqueles que foram mais abordados nas aulas, tendo em conta que estes são aqueles com que temos mais perícia no manuseio e, portanto, acreditamos conseguir influenciar com maior benefício os resultados. No entanto, no último caso, conseguimos obter resultados através da utilização de outros nodos não abordados anteriormente.

4.4 Casos de estudo

4.4.1 Estudo simples dos resultados - Linear Regression

No caso do estudo simples, pretendemos apenas ver quais dos *learners* terão melhor resultados através do mínimo tratamento de dados, utilizando as colunas inferidas anteriormente como úteis e excluindo as complexas, como o caso das listas, que têm de ser abordadas separadamente.

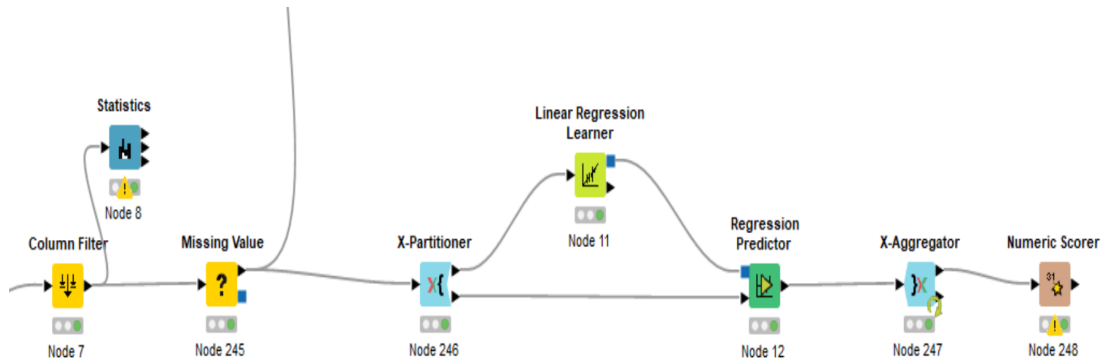


Figura 21: Caso simples - Linear Regression

4.4.1.1 Tratamento de dados

No caso de Linear Regression, como este prevê valores contínuos, e a nossa coluna alvo é também ela um valor deste tipo, o único tratamento que realizamos foi uma limpeza dos *missing values*, substituindo valores *string* por um simples '-' e valores numéricos por 0.

De seguida, realizamos um partição dos dados para teste utilizando os nodos de `cross validation`, com 10 iterações, de modo a podermos observar os resultados de várias iterações.

4.4.1.2 Resultados

Os resultados obtidos de `cross validation` e `numeric scorer` foram:

Row ID	D Total s...	D Mean s...	I Size of ...
fold 0	1,640.11	1.64	1000
fold 1	1,521.816	1.522	1000
fold 2	1,559.639	1.56	1000
fold 3	1,561.911	1.562	1000
fold 4	1,354.696	1.355	1000
fold 5	1,472.8	1.473	1000
fold 6	1,916.435	1.916	1000
fold 7	1,168.837	1.169	1000
fold 8	1,473.367	1.473	1000
fold 9	1,610.111	1.61	1000

(a) Caso simples - Cross validation

Statist...	
File	
⚠ Can't calculate Mean Absolute ...	
R ² :	0,158
Mean absolute error:	0,824
Mean squared error:	1,528
Root mean squared error:	1,236
Mean signed difference:	-0,001
Mean absolute percentage error:	NaN
Adjusted R ² :	0,158

(b) Caso simples - Numeric Scorer

Como esperado, os resultados não são muito otimistas, tendo em conta que um dos objetivos será aproximar o valor de R^2 para 1 e o valor do erro para 0. Tendo em conta que os valores do *average score* se encontram no intervalo de $[0,10]$, podemos concluir que um erro médio no valor de 0.824 não é um bom resultado.

4.4.2 Arredondamento dos average scores - Linear Regression

No caso do arredondamento dos *average scores*, o objetivo é analisar se a transformação dos valores objetivos de contínuos para inteiros traria melhores resultados.

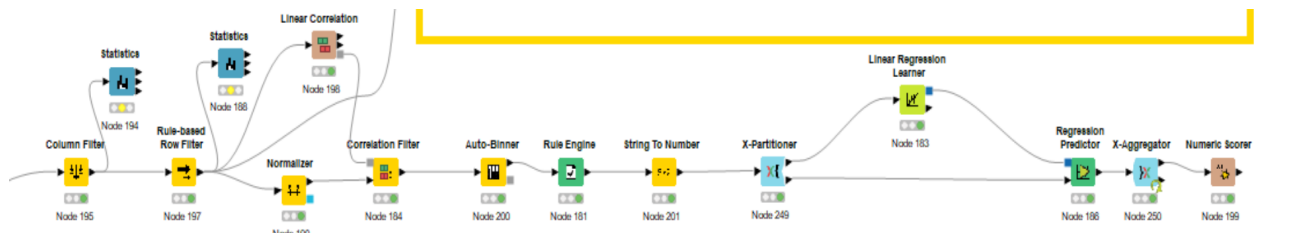


Figura 22: Arredondamento dos average scores - Linear Regression

4.4.2.1 Tratamento de dados

Para o tratamento de dados, mantemos as mesmas *features* que no caso anterior e procedemos a um filtração das entradas para o caso do *budget* do filme ser inferior ou igual a 0, vistos estas linhas não serem boas entradas, considerando que ou são *outliers*, ou não oferecem informação completa. Com isto, o número de entradas é reduzido em 7000 (mais de metade). Seguido disto, fazemos uma normalização dos dados e utilizamos um *correlation filter* de modo a eliminar *features* que possam ter-se tornado irrelevantes ou não benéficas.

Por fim, tratamos da coluna alvo. Para isto, realizamos um *binning* com 10 *bins*, 1 para cada *range* entre inteiros de 0 a 10, seguido de um *rule engine* para transformar as *strings* em números, por exemplo "Bin 1" → 1. Finalmente, utilizamos o nodo *String to Number* para transformar estes valores em numéricos.

4.4.2.2 Resultados

Row ID	D Total s...	D Mean s...	I Size of ...
fold 0	264.487	0.76	348
fold 1	214.739	0.617	348
fold 2	263.868	0.758	348
fold 3	289.787	0.833	348
fold 4	223.853	0.645	347
fold 5	251.281	0.722	348
fold 6	243.602	0.7	348
fold 7	305.837	0.879	348
fold 8	276.664	0.795	348
fold 9	301.871	0.87	347

(a) Arredondamento do Average Score - Cross validation

File	
R ² :	0,261
Mean absolute error:	0,665
Mean squared error:	0,758
Root mean squared error:	0,871
Mean signed difference:	0,002
Mean absolute percentage error:	0,113
Adjusted R ² :	0,261

(b) Arredondamento do Average Score - Numeric Scorer

Observando os resultados, verificamos que os erros obtidos nas várias partições diminuíram consideravelmente, e o valor do R^2 na partição com 0.758 de erro quadrado aumentou. Assim sendo, este modelo obteve resultados razoavelmente melhores do que o anterior. No entanto, não podemos descartar o facto de que isto pode ter ocorrido devido ao tamanho de estudo ter diminuído.

4.4.3 Binned average scores - Linear Regression

Este caso consiste na criação de uma nova coluna com os valores de *average score binned*, no mesmo estilo do caso anterior. Porém, esta coluna servirá como auxílio ao modelo, um indicador do intervalo em que o valor do *average score* se encontra, sendo portanto um caso que na vida real apenas se aplica se houver uma *feature* que possa servir como atrator para o resultado verdadeiro. Assim, esta coluna adicionada terá bastante *bias*, mas consideramos que seria um estudo interessante e que, embora muitos dos *datasets* não possuem uma *feature* deste estilo, podemos observar o valor de um atributo deste estilo.

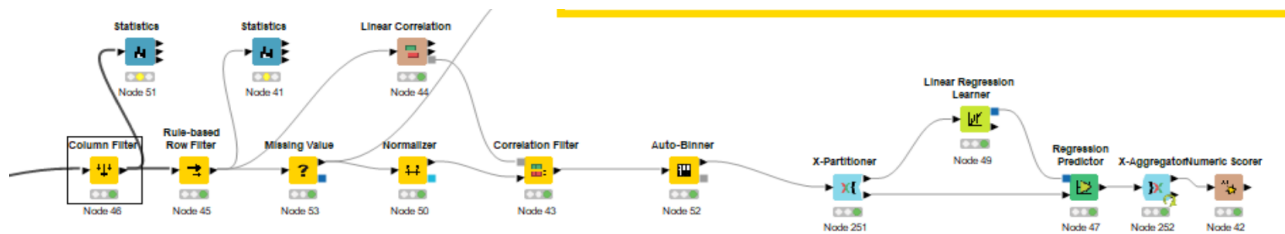


Figura 23: Arredondamento dos average scores - Linear Regression

4.4.3.1 Tratamento de dados

O tratamento de dados é homogêneo ao anterior, exceto no tratamento da coluna *binned*, que, em vez de trocar a coluna alvo, será uma nova coluna e, será mantida como *string* e com os seus valores automaticamente impostos.

4.4.3.2 Resultados

Row ID	D	Total s...	D	Mean s...	I	Size of ...
fold 0		17.476		0.068		258
fold 1		18.137		0.07		258
fold 2		15.396		0.06		257
fold 3		18.17		0.07		258
fold 4		17.005		0.066		257
fold 5		18.702		0.072		258
fold 6		18.828		0.073		258
fold 7		18.119		0.071		257
fold 8		19.082		0.074		258
fold 9		19.505		0.076		257

(a) Arredondamento do Average Score - Cross validation

File	
R ² :	0,917
Mean absolute error:	0,223
Mean squared error:	0,07
Root mean squared error:	0,265
Mean signed difference:	0
Mean absolute percentage error:	0,036
Adjusted R ² :	0,917

(b) Arredondamento do Average Score - Numeric Scorer

Como esperado, o valor de R^2 aumentou exponencialmente, assim como a diminuição do valor do erro, notando-se que este é notavelmente o melhor modelo, embora claro, não sendo o mais realista.

4.4.4 Empresas de produção - Linear Regression

Tendo em conta as diferentes empresas de produção que existem no mercado, várias delas acabam por ter melhor reputação do que as outras, e isto deve-se, em muitos casos, o facto de consistentemente serem capazes de produzir bons filmes. Deste modo, procuramos saber se este conhecimento poderia ser utilizado pela aprendizagem automática para obter melhores resultados.

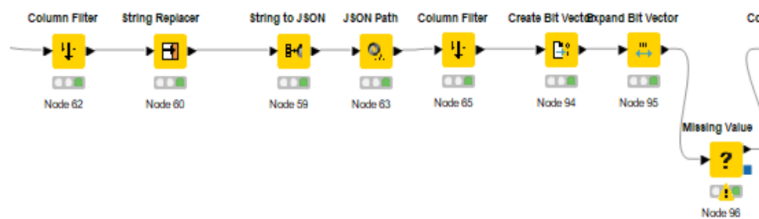


Figura 24: Tratamento das empresas de produção

Neste cenário, o *learner* utilizado foi semelhante ao do caso anterior. Como tanto este caso como o do *decision tree* utilizam o mesmo tratamento de dados das companhias, utilizamos a mesma ilustração para os dois.

4.4.4.1 Tratamento de dados

Aquilo que difere é, como mencionado acima, o tratamento da lista das empresas de produção. Para isto, primeiro é preciso reconhecer que estas estão num formato extremamente semelhante a JSON, sendo que a única diferença é o facto de em vez de terem sido utilizadas aspas como se deve utilizar neste formato, foram utilizadas plicas (').

Para tratar disto, utilizamos o nodo **String Replacer** para colocar o JSON no formato correto com as aspas.

A partir daqui já podemos converter a coluna diretamente num JSON com o nodo *String to JSON*. Com os nodos de manipulação de JSON *JSON Path*, utilizamos a query "\$.name" para retirar da lista apenas o nome das empresas, ficando assim a coluna como sendo uma lista de nomes.

O último passo é então tornar esta lista num *bit vector* com o nodo **Create Bit Vector**, que pegará em todos os nomes únicos das listas, e utilizar essa informação para, em todas as entradas, trocar a lista de nomes por um array de bits, que estarão preenchidos de acordo com o valor das empresas que aquele filme tem como produtoras. Com estes vetores, podemos utilizar o nodo **Expand Bit Vector** para os separar em colunas únicas, tantas quanto as empresas únicas e, automaticamente preencher os valores de se um filme foi produzido por uma certa empresa através dos valores dos vetores.

Para a partição dos valores, por a quantidade de colunas ter aumentado imenso, o peso de processamento resultou na decisão de utilizar um simples nodo **partitioner**.

4.4.4.2 Resultados

Row ID	D Predicti...
R^2	-52.253
mean absolut...	6.39
mean square...	41.613
root mean sq...	6.451
mean signed ...	-6.39
mean absolut...	1
adjusted R^2	-52.253

Figura 25: Empresas de produção - Numeric Scorer

Embora à partida o bias que temos perante as diferentes empresas, parece que pode-se influenciar positivamente os resultados. Na realidade, a quantidade de informação que esta coluna trouxe foi demasiado grande para o **linear regression learner** conseguir tratar, sendo que acabou por defeito atribuir o valor de 0 a todas as previsões. Concluímos portanto que esta coluna é demasiado abrangente para um **regression learner** utilizar.

4.4.5 Categorias de filmes - Linear Regression

As categorias são, da mesma forma que as empresas de produção, uma coluna que tem como valores, uma lista das diferentes categorias de cada filme.

Utilizando estes dados, procuramos verificar se as categorias de um filme têm alguma relação com a sua qualidade/classificação, por exemplo, se um filme de ação tem mais tendência a apelar a um maior público e consequentemente tem uma melhor média, ou o contrário para um filme de horror.

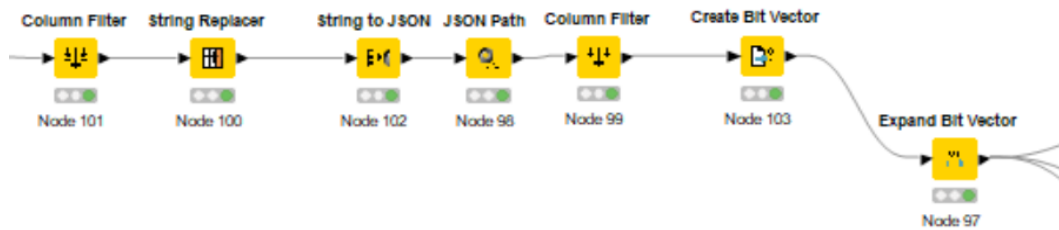


Figura 26: Tratamento de categorias

4.4.5.1 Tratamento de dados

De forma a retirar a informação da lista de categorias, seguimos o mesmo plano que foi utilizado para as empresas de produção.

Para o resto do tratamento de dados utilizámos os mesmos métodos que no cenário 4.4.2.

4.4.5.2 Resultados

File	
R ² :	0,35
Mean absolute error:	0,582
Mean squared error:	0,57
Root mean squared error:	0,755
Mean signed difference:	0,043
Mean absolute percentage error:	0,092
Adjusted R ² :	0,35

Figura 27: Categorias de filmes - Numeric Scorer

Ao contrário do caso anterior, o número de colunas geradas foi muito menor e portanto ainda no espectro que o **Linear Regression Learner** consegue aceitar.

Ainda distinto do caso das empresas de produção foi o resultado, que teve uma melhoria em relação ao cenário do arredondamento da coluna alvo, tanto no valor de R^2 , como no valor do erro.

Concluimos então, que as categorias dos filmes têm uma influência relevante na média de um filme.

4.4.6 Estudo simples dos resultados - Decision Tree

O segundo nodo de aprendizagem automática que exploramos foi o de árvores de decisão.

Como estas não conseguem prever valores contínuos, a coluna alvo e os resultados terão de ser valores nominais correspondentes a um arredondamento dos valores das médias para o inteiro mais próximo.

O primeiro estudo com este *learner* omitirá, como no caso do **linear regression**, as colunas complexas.

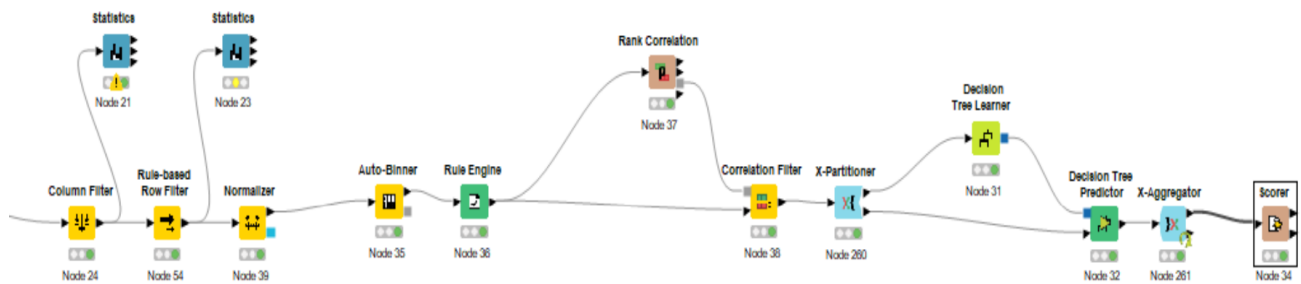


Figura 28: Estudo simples - Decision Tree

4.4.6.1 Tratamento de dados

Os dados passados para aprendizagem são filtrados de forma a ficar com linhas que tenham informação suficiente/produtiva. Os valores resultantes são depois normalizados e por fim o atributo alvo é passado pelo **auto-binner**, com valor de 10 bins.

O *binner* da coluna alvo é feito pela motivo referido no início do caso.

o *partitioning* é novamente realizado pelos nodos de **cross validation**.

4.4.6.2 Resultados

Row ID	Error in %	Size of ...	Error C...
fold 0	52.713	258	136
fold 1	55.426	258	143
fold 2	52.529	257	135
fold 3	53.101	258	137
fold 4	51.751	257	133
fold 5	55.426	258	143
fold 6	58.527	258	151
fold 7	58.366	257	150
fold 8	55.426	258	143
fold 9	56.031	257	144

(a) Estudo simples - Cross validation

File	Hilite				
vote_aver...	8	7	6	9	5
8	305	205	37	21	7
7	206	561	249	2	49
6	59	275	254	2	72
9	27	4	1	14	1
5	6	58	93	0	27
4	2	3	14	0	5
3	1	1	4	0	1
2	0	0	1	0	0
<div><</div>					
Correct classified: 1 161			Wrong classified: 1 415		
Accuracy: 45,07%			Error: 54,93%		
Cohen's kappa (κ): 0.214%					

(b) Estudo simples - Scorer

Pelos valores das matrizes de confusão, verificamos que os resultados variam entre os 40-50% de acertos, mostrando que existe ainda espaço para melhorar o modelo. Tendo ainda em conta que estas são apenas previsões dos valores arredondados, temos ainda a certeza que comparativamente aos valores originais, a quantidade de acerto é muito reduzida.

4.4.7 Empresas de produção - Decision Tree

Como no caso 4.4.4, experimentamos utilizar a informação das empresas de produção para validar o modelo.

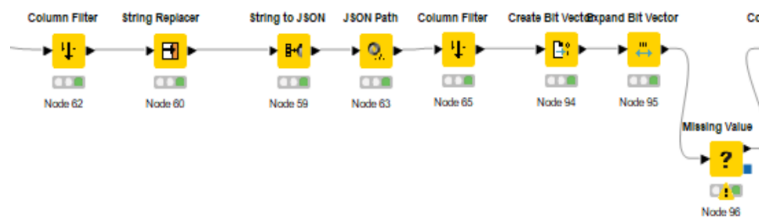


Figura 29: Tratamento das empresas de produção

4.4.7.1 Tratamento de dados

Os dados tiveram o mesmo tratamento que na secção com o mesmo objetivo, até ao ponto em que os dados têm de ser passados para os **decision tree**, onde o tratamento é igual à secção anterior excluindo o *partitioning*, onde utilizamos o simples com uma partição de 80/20 por ter tido o melhor resultado.

4.4.7.2 Resultados

vote_aver...	8	7	6	9	5
8	38	75	9	0	0
7	32	146	38	0	1
6	9	78	34	0	8
9	7	2	0	1	0
5	3	15	10	0	1
4	1	0	4	0	0
3	0	3	0	0	1
2	0	0	0	0	0

<		>
Correct classified: 220	Wrong classified: 296	
Accuracy: 42,636%	Error: 57,364%	
Cohen's kappa (κ): 0,12%		

Figura 30: Empresas de produção - Scorer

Como observado no caso de **linear regression**, os dados fornecidos por esta *feature* foram detrimntosos ao modelo. Este modelo foi capaz de utilizar os dados para obter um resultado que, embora não perfeito, foi bastante superior ao do **linear regression**.

4.4.8 Categorias de filmes - Decision Tree

Estudo da influência das categorias dos filmes nas árvores de decisão.

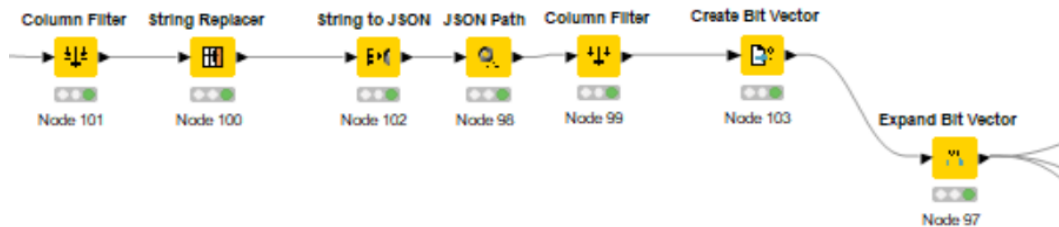


Figura 31: Tratamento de categorias

4.4.8.1 Tratamento de dados

O tratamento da lista de categorias segue os passos do caso 4.4.5, e antes de serem passados para as árvores de decisão, os dados passam por uma sequência de nodos igual à secção inicial de `decision trees` (4.4.6).

4.4.8.2 Resultados

File	HiLite				
vote_aver...	8	7	6	9	5
8	64	48	6	1	1
7	32	117	46	1	5
6	12	57	56	0	16
9	8	1	0	3	0
5	2	10	10	0	10
4	0	0	4	0	2
3	0	1	0	0	0
2	0	1	0	0	0
< <div></div>					
Correct classified: 250			Wrong classified: 266		
Accuracy: 48,45%			Error: 51,55%		
Cohen's kappa (κ): 0,263%					

Figura 32: Categorias de filmes - Scorer

Tal como ocorreu para a regressão linear, as categorias beneficiaram os resultados do modelo.

4.4.9 Estudo simples dos resultados - RProp MLP

O último modelo de aprendizagem que explorámos com profundidade foram as redes neuronais, mais concretamente, os nodos de `RProp MLP`.

Tal como nos nodos de regressão linear, os valores que as redes neuronais calculam são contínuos, embora, todas as *features* que recebem no modelo têm de ser numéricas e normalizadas.

Os resultados que as redes neuronais calculam irão variar de forma volátil dependendo não só do tratamento de dados, mas das opções no próprio *learner*, mas especificamente do número de camadas de neurónios, número de neurónios por camada e número de iterações.

Nesta primeira abordagem, tal como para os outros modelos, começamos por excluir as *features* complexas.

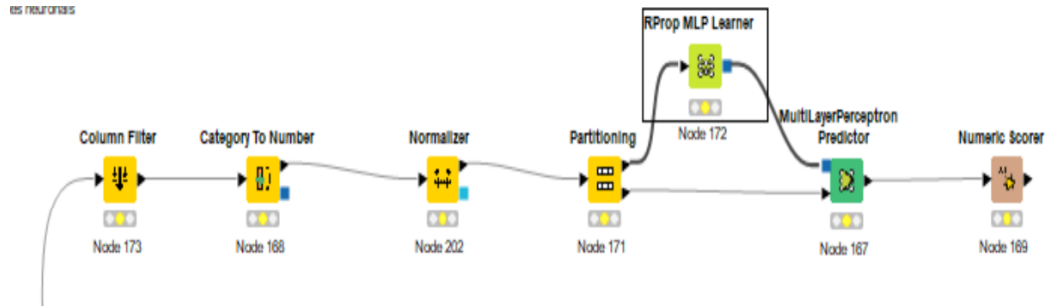


Figura 33: Estudo Simples - RProp MLP

4.4.9.1 Tratamento de dados

Para os dados poderem ser processados pelas redes neuronais, as colunas nominais são transformadas em numéricas através do nodo **category to number**.

Todos os valores são normalizados e, por último, faz-se o *partitioning simples* (80/20), visto que as redes neuronais são computacionalmente bastante mais pesadas do que os outros nodos.

4.4.9.2 Resultados

Can't calculate Mean Absolute ...	
R ² :	0,315
Mean absolute error:	0,082
Mean squared error:	0,013
Root mean squared error:	0,114
Mean signed difference:	0,001
Mean absolute percentage error:	NaN
Adjusted R ² :	0,315

Figura 34: Estudo Simples - Scorer

Colocando os valores das redes neuronais do seguinte modo: 500 iterações, 15 camadas, 10 neurónios por camada, obtemos os resultados acima.

Analisamos que, comparativamente ao outro modelo que prediz valores contínuos (regressão linear), este teve um resultado no primeiro estudo de R^2 bastante melhor.

O aumento dos recursos das redes não mostrou melhorias significativas.

4.4.10 Arredondamento dos average scores - RProp MLP

Para os dados arredondados, como antes visto na secção 4.4.2, estudamos se uma mudança na coluna alvo traz diferenças nos resultados.

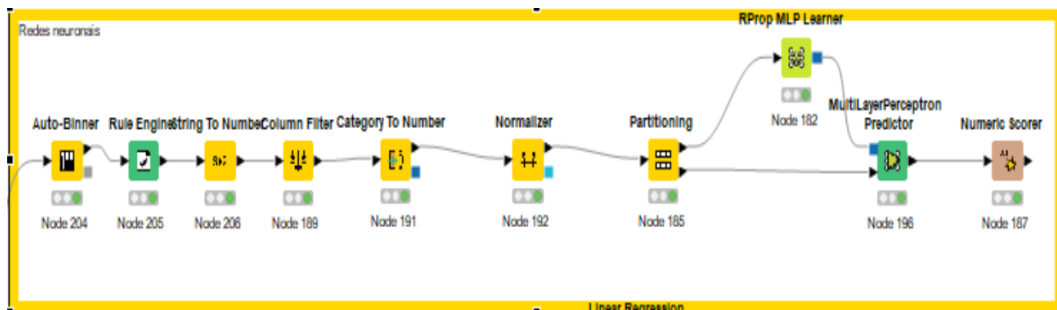


Figura 35: Arredondamento - RProp MLP

4.4.10.1 Tratamento de dados

O tratamento de dados assemelha-se ao utilizado na secção 4.4.2, e transformação para serem válidos no modelo é igual à da caso anterior.

4.4.10.2 Resultados

File	
R ² :	0,305
Mean absolute error:	0,091
Mean squared error:	0,013
Root mean squared error:	0,114
Mean signed difference:	-0,009
Mean absolute percentage error:	0,15
Adjusted R ² :	0,305

Figura 36: Arredondamento - Scorer

As características do nodo de redes neuronais foram mantidas iguais.

Ao contrário do que se demonstrou no caso de regressão linear, este arredondamento decrementou a taxa de sucesso do modelo.

O aumento dos recursos das redes não mostrou melhorias significativas.

4.4.11 Binned average scores - RProp MLP

Caso de atributo de auxílio, com o mesmo propósito do caso 4.4.3.

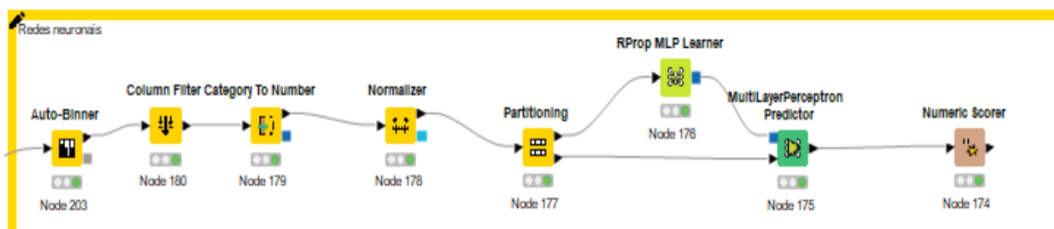
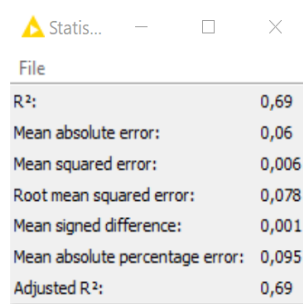


Figura 37: Binned average scores - RProp MLP

4.4.11.1 Tratamento de dados

Como `linear regression` e RProp MLP conseguem trabalhar com os mesmo tipos de dados, o tratamento destes é idêntico, à exclusão da normalização que no caso das redes neuronais é aplicada a todos os atributos.

4.4.11.2 Resultados



File	
R ² :	0,69
Mean absolute error:	0,06
Mean squared error:	0,006
Root mean squared error:	0,078
Mean signed difference:	0,001
Mean absolute percentage error:	0,095
Adjusted R ² :	0,69

Figura 38: Binned average score - Scorer

Como já se havia verificado, este atributo atrator fortemente influencia o modelo, trazendo este para valores de acerto muito superiores.

O aumento dos recursos das redes mostrou melhorias significativas, dobrando o número de iterações, o valor do R^2 atingiu valores superiores a 0.8.

4.4.12 Categorias de filmes - RProp MLP

Por último, estudamos também a influência das categorias nas redes neuronais. Embora nos outros dois cenários estas tenham tido um impacto positivo, o facto que a quantidade de *inputs/features* vai aumentar consideravelmente poderá significar uma necessidade de aumentar a quantidade de recursos disponibilizados para conseguir resultados que sigam esta tendência de melhoria.

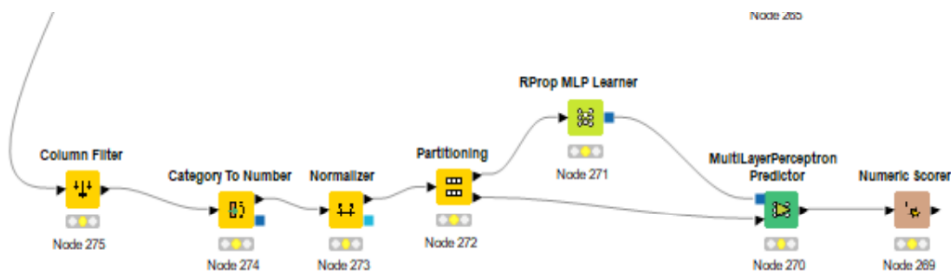
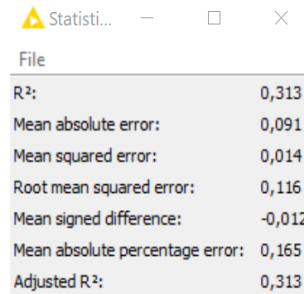


Figura 39: Categorias de filmes - RProp MLP

4.4.12.1 Tratamento de dados

O tratamento seguiu os mesmos moldes da regressão linear na secção 4.4.5.

4.4.12.2 Resultados



A screenshot of a software window titled 'Statisti...' with standard window controls. It displays a table of statistical metrics for a model's performance on movie categories. The metrics include R-squared, Mean absolute error, Mean squared error, Root mean squared error, Mean signed difference, Mean absolute percentage error, and Adjusted R-squared.

File	
R ² :	0,313
Mean absolute error:	0,091
Mean squared error:	0,014
Root mean squared error:	0,116
Mean signed difference:	-0,012
Mean absolute percentage error:	0,165
Adjusted R ² :	0,313

Figura 40: Categorias de filmes - Scorer

Como inferido, os resultados não variaram muito do primeiro caso, tendo até levemente piorado no entanto, tal como no caso anterior, o aumento dos recursos providos às redes melhoram o seu sucesso, embora neste caso este aumento seja mais custoso. Um aumento, para 2000 iter, 20 camadas e 20 neurons levou a um R^2 de 0.435.

4.4.13 Outros learners

De modo a termos um ideia do potencial de outros modelos de aprendizagem que não foram aprofundados nas aulas, corremos um teste equivalente ao estudo simples para os seguintes modelos:

- Naive Bayes
- Random Forest Learner
- Polynomial Regression learner
- PNN Learner

4.4.13.1 Tratamento de dados

Os dois primeiros modelos tratam de atributos alvos semelhantes a árvores de decisão, ou seja, nominais, tendo portanto os dados um tratamento semelhante ao realizado no caso 4.4.6.

Os outros dois modelos trabalham com colunas alvo de valores contínuo, tal como a regressão linear. Assim, seguirão o mesmo tratamento que esta.

4.4.13.2 Resultados

Os resultados obtidos foram:

Modelo	Accuracy/ R_2
<i>Naive Bayes</i>	45,543%
<i>Random Forest</i>	44,574%
<i>Polynomial Regression</i>	0.167
<i>PNN</i>	-0.282

Pelos valores obtidos, verificamos que em pé de igualdade, o resultado do modelo de **Naive Bayes** foi ligeiramente superior ao da **Decision Tree**. Também **Polynomial Regression** foi ligeiramente melhor em termos de R_2 do que o **Linear Regression**.

Embora as partições escolhidas possam ter influenciado estas diferenças, principalmente tendo em conta que não são muito significativas, poderão ter potencial para estudo futuro.

4.5 Análise dos resultados

Com todos os casos de estudo finalizado, podemos finalmente chegar a algumas conclusões gerais sobre o *dataset* e os modelos utilizados para o estudar. Infelizmente a melhor percentagem de *accuracy* utilizando árvores de decisão foi de cerca de 50%, e, tendo em conta que estas foram tratadas de modo a preverem os *average scores* arredondados, este resultado não foi muito surpreendente, nem viu grandes melhorias com os diferentes tratamentos de dados. Isto, no entanto, era espectável, visto que este tipo de modelo não é apropriado para previsões concretas de valores contínuos.

No caso dos modelos de previsão de valores contínuos, de regressão linear e redes neuronais, o melhor resultado foi de R^2 igual a 0.435 que, embora ainda longe de ter sido maximizado, sofreu uma significativa melhoria através do tratamento de dados relativamente à aplicação base destes modelos.

Podemos ainda afirmar que diferentes atributos complexos, como as listas que foram tratadas devem ser manualmente investigadas separadamente, visto que os seus resultados podem variar negativamente pelo simples excesso de informação como foi o caso das empresas de produção.

5 Conclusão

Através da resolução deste trabalho fomos conduzidos a aprofundar o nosso conhecimento sobre exploração de dados, assim como *machine learning* e utilização do *software* de navegação e manipulação de dados KNIME.

No primeiro *dataset* que estudámos, o objetivo foi a previsão do valor das casas dentro da uma extensa região que os Estados Unidos da América abrange, através do conhecimento de apenas alguns dados sobre as áreas em que estas se localizavam. Neste contexto, os nodos capazes de lidar com este tipo de *target* brilharam, nomeadamente **Linear Regression** e **RProp MLP**, tendo obtido resultados bastante satisfatórios, com margens de erro relativamente baixas (tendo em conta a escala dos valores tratados).

Por outro lado, o modelo de árvores de decisão, como seria de esperar por não ser o mais indicado para encontrar este tipo de solução, não foi capaz de atingir taxas de acerto superiores a 60%.

Seguindo este padrão, no segundo *dataset*, embora os resultados não tenham sido tão notáveis como no primeiro, os dois primeiros modelos conseguiram resultados que consideramos ser mais aceitáveis do que o **Decision Tree**. Novamente, isto deveu-se ao facto de este objetivo não ser o apropriado para este tipo de modelo tratar. Ainda assim, o conjunto de dados escolhidos pelo nosso grupo trouxeram alguns desafios que foram interessantes de resolver, como foi o caso da extração de informação das listas, que implicou a exploração de novos nodos, e no caso das categorias dos filmes demonstrou que o aproveitamento destes atributos mais complexos podia beneficiar o modelo.

Concluindo, a realização deste trabalho permitiu aumentar a nossa experiência com a exploração e manipulação de dados, tendo extraído desta novos conhecimentos que serão proveitosos para o futuro, como é exemplo, o facto de que diferentes *datasets* terão modelos para os quais serão melhor direcionados, tendo portanto de, para testar modelos como **Decision Tree**, procurar *targets* que sejam categóricos, como foi exemplo a categorização de uma flor nas aulas práticas.

Referências

- [1] Website do software Knime
- [2] Website da plataforma IMDB
- [3] Repositório com o *dataset* dos filmes