

Universidade do Minho

Licenciatura em Engenharia Informática

Aprendizagem e Decisão Inteligentes

3ºAno, 2ºSemestre

Ano letivo 2023/2024

Conceção de modelos de aprendizagem e decisão

João Magalhães, A100740

Jorge Rodrigues, A101758

Rodrigo Gomes, A100555

Simão Costa, A95176

Índice

1	INTRODUÇÃO	3
2	METODOLOGIA	3
3	TAREFA DATASET ATRÍBUÍDO	3
3.1	Estudo de negócios	4
3.2	Estudo de Dados	4
3.3	Preparação dos Dados	11
3.4	Modelação	12
3.5	Avaliação dos Modelos	15
4	TAREFA DATASET ESCOLHIDO	18
4.1	Estudo de negócios	18
4.2	Estudo de Dados	18
4.3	Preparação dos Dados	26
4.4	Modelação	28
4.5	Avaliação dos Modelos	29

1 Introdução

No âmbito da unidade curricular de Aprendizagem e Decisão Inteligentes, o grupo iniciou o desenvolvimento de dois projetos que envolviam a conceção dos diferentes modelos de aprendizagem estudados durante o semestre. Para esse efeito, foi preciso aplicar técnicas de exploração e preparação de dados, assim como aplicar os modelos corretamente dado o conhecimento extraído dos conjuntos de dados. Desta forma, a primeira tarefa que realizamos foi a do Dataset Atribuído, com um domínio sobre casos de Hepatite c. Já segunda foi escolhida por nós e envolve um Dataset popular que aborda a venda de carros usados.

2 Metodologia

Antes de avançar com o problema, é uma excelente prática definir uma metodologia, através da qual se possa garantir que o produto final desenvolvido seja, efetivamente, construído tendo em conta os modelos que são colocados por empresas e investigadores que laboram na área de *Machine Learning*. Nesta unidade curricular, foram instruídas as seguintes metodologias:

- CRISP-DM
- SEMMA

Ambas as *frameworks* foram apresentadas como boas opções para projetos de dados, contudo, achamos que a vencedora é a metodologia **CRISP-DM**. Mas, dado o trabalho ser realizado no âmbito da UC, optamos por adaptá-la de acordo com as necessidades e particularidades do nosso projeto. Como tal, paramos o nosso trabalho na fase de avaliação, não realizando o *deployment* do projeto. De modo geral, acreditamos que a configuração escolhida é bastante versátil, sendo apropriada para casos onde os analistas não são muito conhecedores do domínio, isto porque oferece um ciclo que começa precisamente com um estudo de negócio e de dados, não dependendo de tecnologias, ao contrário da *framework SEMMA* que é tipicamente utilizada em ambientes da empresa SAS.

3 Tarefa Dataset Atribuído

Nesta etapa do projeto, foi escolhido pela equipa docente um *Dataset* para o grupo manusear. Para esse efeito, a equipa de trabalho teve que implementar diversas técnicas lecionadas na disciplina, desde um estudo inicial sobre o tema até aos modelos resultantes do nosso trabalho.

O *Dataset* atribuído tem como domínio “Saúde e Medicina” e possui 14 colunas, onde o atributo de decisão é discreto e todas as entradas são de pacientes de um hospital (não referido).

3.1 Estudo de negócios

O estudo de negócio é precisamente a compreensão dos objetivos do projeto que vamos realizar. Pelo repositório de onde os dados provêm, é referido que a finalidade da exploração que temos pela frente é classificar um paciente como doador de sangue ou doente contaminado com Hepatite C. No caso de estar infectado, são adicionadas novas categorias, dependendo do progresso da doença. Neste caso, “só” Hepatite C, cirrose e fibrose. Desta forma, o objetivo do nosso modelo será, com precisão, ser capaz de categorizar um paciente entre as opções referidas, tendo em consideração os diversos atributos presentes nos dados.

3.2 Estudo de Dados

3.2.1 Descrição dos dados

O *Dataset* possui um total de 615 entradas e 18 colunas por entrada. Também sabemos que existem valores que estão em falta, mas esses detalhes ficam para a especificação detalhada dos elementos das colunas.

1. **id**: Código identificador do paciente
2. **Age**: Idade em anos
3. **year_of_birth**: Ano de nascimento
4. **month_of_birth**: Mês de nascimento
5. **day_of_birth**: Dia de nascimento
6. **Sex**: Sexo
7. **birth_location**: Local de nascimento
8. **ALB**: Albumina
9. **ALP**: Fosfatase alcalina
10. **ALT**: Alanina transaminase
11. **AST**: Aspartato transaminase
12. **BIL**: Bilirrubina
13. **CHE**: Colinesterase
14. **CHOL**: Colesterol
15. **CREA**: Creatina
16. **GGT**: Gama Glutamil Transferase
17. **PROT**: Proteína
18. **Category**: Diagnóstico

O atributo alvo é a **Category**. Será esta coluna que o nosso modelo deve, após o treino, conseguir realizar uma previsão.

Vamos agora especificar cada atributo, um a um, e perceber que conhecimento surge da respetiva análise. Esta fase antecede a preparação de dados, logo não iremos ainda referir como vamos tratar cada

atributo, mas sim fundamentar o seu tratamento. É relevante mencionar que o desenvolvimento não é feito em cascata, isto é, poderemos sempre voltar a precisar de analisar cada coluna.

3.2.2 Id

Este atributo tem como único propósito ser um código único por paciente do conjunto de dados. É uma prática comum em Bases de Dados, que não deve ser considerada em *Machine Learning*. Isto deve se ao facto de muitos modelos encontrarem relações entre a o atributo alvo e o identificador, que são sempre coincidência. No caso do nosso *Dataset*, os pacientes estão ordenados pela sua categoria, o que faz com que o “id” esteja diretamente relacionado com o atributo alvo. Este aspeto leva a que algoritmos como as árvores de decisão usem este valor incorretamente. Com isto, o “id” deve ser sempre desconsiderado para modelagem.

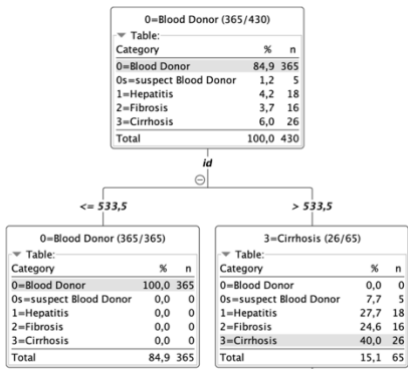


Figura 1 Exemplo do problema descrito numa Árvore de Decisão

3.2.3 Age (idade), Year_of_birth, Month_of_birth, Day_of_birth (data de nascimento)

Começamos por ver o histograma gerado a partir do atributo “age”, pois é relevante perceber como é que as idades estão distribuídas na nossa amostra. De forma análoga, observamos também o histograma dos anos de nascimento.

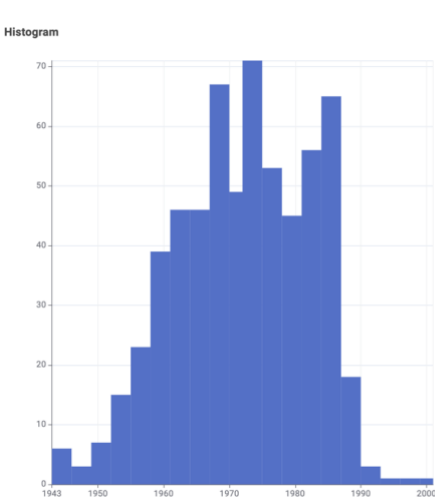
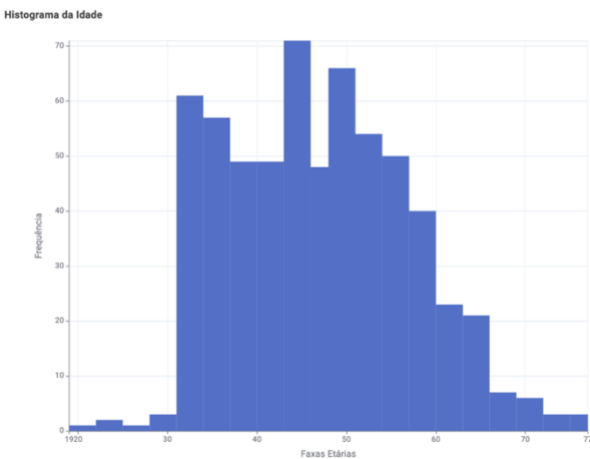


Figura 2 Histogramas sobre a idade e ano de nascimento

Aqui vemos que os diagramas são semelhantes, mas não equivalentes, o que pode indicar que os valores não são concordantes. Deste modo, decidimos ver se a idade está concordante com a data de nascimento (utilizando os três atributos).

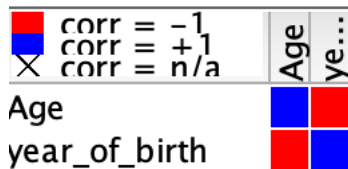


Figura 4 Correlação entre idade e ano de nascimento

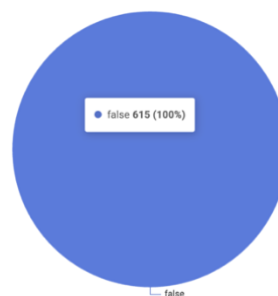


Figura 3 Gráfico de setores idade/data de nascimento

Na figura 3, vemos que não há concordância entre a data de nascimento. Contudo, através da análise da correlação entre os atributos idade e ano de nascimento, reparamos que existe uma correlação inversa perfeita. Este facto leva-nos a acreditar que a idade não está atualizada para a data de realização deste projeto, sendo este o facto que leva aos resultados não concordantes. De um ponto de vista clínico, não é relevante os dados estarem atualizados com 2024, pois o que realmente importa é ter amostras que cubram um maior número de idades. Tentamos também procurar, apesar de não encontrar indícios fortes, alguma relação entre o atributo alvo e a idade. Mais sobre essa questão no anexo 1.

3.2.4 Sex (sexo)

Para esta *feature*, começamos por observar que, para os 615 pacientes, 369 são do sexo masculino, 238 são do sexo feminino e 8 pacientes foram categorizados com “mm”, categorização que não está explícita no *Dataset* e de difícil compreensão. Esta atribuição tanto pode ser uma falha do sistema, como pode ser o resultado de pacientes que não permitiram que esse dado em específico fosse divulgado. Estes dados encontram-se representados no seguinte gráfico de setores:

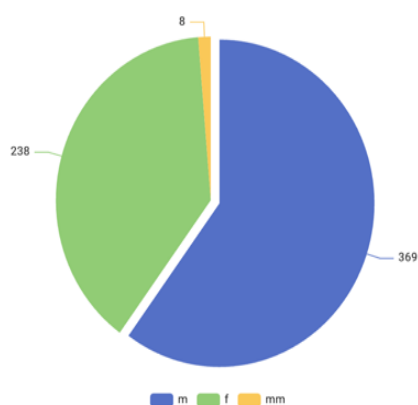


Figura 5 Distribuição dos pacientes pelo atributo sexo

Após limpeza, sem grande sucesso, procuramos encontrar algum tipo de relação entre o sexo e o atributo alvo. Os nossos resultados encontram-se no anexo 1 entregue com este relatório

3.2.5 Birth Location (local de nascimento)

Observando as proporções, chegamos aos seguintes valores:



Figura 6 Pie Chart da distribuição dos pacientes na classe Birth_Location

Deste diagrama, conseguimos observar que a grande maioria dos registos são de pacientes nascidos em “New Delhi”, enquanto os restantes possuem um campo vazio. Este campo vazio pode representar várias situações, dependendo da interpretação: Tanto pode ser verdadeiramente desconhecido, como pode ser um paciente de fora de “New Delhi”.

Assim como para as *features* anteriores, decidimos procurar uma relação entre o local de nascimento e o diagnóstico.

3.2.6 ALB (Albumina)

A Albumina é uma proteína produzida no fígado que é encontrada no plasma sanguíneo. Primeiramente, fomos procurar saber se existiam entradas que não possuíam este valor. De facto, existe um paciente que tem o valor “NA”, o que interpretamos como valor desconhecido. De resto, conseguimos valores para todos os outros pacientes. Desta forma, observamos o seguinte:

Column ↑↓	Exclude Column	Minimum ↑↓	Maximum ↑↓	Mean ↑↓	Standard Deviation ↑↓	Variance ↑↓	Skewness ↑↓	Kurtosis ↑↓	Overall Sum ↑↓
ALB	<input type="checkbox"/>	14.900	82.200	41.620	5.781	33.416	-0.177	5.983	25554.800

Figura 7 Estatísticas sobre a Albumina

Aqui percebemos que a albumina varia entre 14.9 e 82.2, com uma média de 41.620 e desvio padrão de 5.781. O intervalo de referência para um adulto saudável é de 35 a 54 g/L, o que se enquadra bem nos valores do nosso dataset.

ALP (Fosfatase alcalina)

Column ↑↓	Exclude Column	Minimum ↑↓	Maximum ↑↓	Mean ↑↓	Standard Deviation ↑↓	Variance ↑↓	Skewness ↑↓	Kurtosis ↑↓	Overall Sum ↑↓
ALP	<input type="checkbox"/>	11.300	416.600	68.284	26.028	677.473	4.655	54.973	40765.500

Figura 8 Estatísticas sobre a Fosfatase alcalina

Para esta *feature*, seguimos uma abordagem semelhante à anterior, isto porque se assemelha muito às condições referidas anteriormente. Para este atributo, existem 18 pacientes sem valor numérico atribuído. Desta forma chegamos aos seguintes valores:

Aqui percebemos que a fosfatase alcalina varia entre 11.3 e 416.6, com uma média de 68.284 e desvio padrão de 26.028. O intervalo de referência para um adulto saudável é de 36 a 150 g/L, o que indica a presença de indivíduos com padrões muito fora do normal.

3.2.7 ALT (Alanina transaminase)

Ainda dentro dos valores numéricos, temos a Alanina transaminase. Neste valor, para além das estatísticas, decidimos ver a relação em gráfico de barras com o nosso atributo alvo. Neste caso, apenas um paciente possui um valor desconhecido.

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
ALT	<input type="checkbox"/>	0.900	325.300	28.451	25.470	648.705	5.506	47.129	17468.800

Figura 9 Estatísticas sobre a Fosfatase alcalina

Da figura retiramos que os valores variam entre 0.9 e 325.3, com média de 28.451 e desvio padrão de 25.470. Os valores para um indivíduo saudável são geralmente inferiores a 35 g/L, logo há casos demasiados extremos fora do comum.

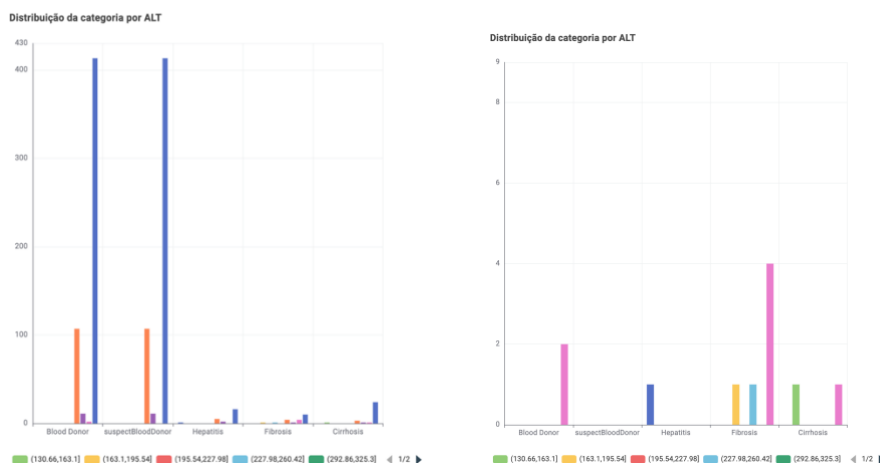


Figura 10 Distribuição de diagnósticos por ALT

Tendo em conta os gráficos da Figura 12, encontramos uma relação promissora entre o que seriam valores elevados ($92.8 >$) de ALT com o diagnóstico, mais concretamente, que estes valores estão relacionados com fases avançadas de hepatite C.

3.2.8 AST (Aspartato transaminase)

Os valores para a Aspartato transaminase também foram incluídos no *Dataset*. Decidimos observar algumas estatísticas sobre este parâmetro, chegando assim à seguinte tabela:

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
AST	<input type="checkbox"/>	10.600	324	34.786	33.091	1094.994	4.940	30.837	21393.600

Figura 11 Estatísticas sobre Aspartato transaminase

Nesta tabela observamos que a média é de 34.786, onde o valor mais baixo encontrado foi de 10.6 e o mais alto de 324, com um desvio padrão de 33.091. Normalmente, estes valores encontram-se abaixo dos 35 g/L, o que mostra a presença de alguns casos fora do comum.

3.2.9 BIL (Bilirrubina)

Dentro das enzimas produzidas no fígado, a Bilirrubina pode também ser bastante útil para a nossa análise.

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
BIL	<input type="checkbox"/>	0.800	254	11.397	19.673	387.033	8.385	83.187	7009

Figura 12 Estatísticas sobre Bilirrubina

Destas estatísticas, podemos ver que o valor varia entre 0.8 e 254, com uma média de 11.397 e um desvio padrão de 19.673. Para este valor, consideramos a Bilirrubina total como valor de referência que, para um adulto, não deverá ser superior a 120 g/L.

3.2.10 CHE (Colinesterase)

Assim, com os valores contínuos já apresentados, para a Colinesterase seguimos o mesmo modelo estatístico que serviu para derivar a seguinte tabela:

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
CHE	<input type="checkbox"/>	1.420	16.410	8.197	2.206	4.865	-0.110	1.315	5040.930

Figura 13 Estatísticas sobre Colinesterase

Daqui retira-se que o intervalo de valores varia entre 1,42 e 16.41, com uma média de 8.197 e um desvio padrão de 2.206. Este atributo é um dos que possui intervalos diferentes para homens e mulheres, mas, de modo geral, os valores devem estar próximos do intervalo 3.9 a 11.5 g/L.

3.2.11 CHOL (Colesterol)

Muito semelhante à Colinesterase, o Colesterol surge com os seguintes valores:

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
CHOL	<input type="checkbox"/>	1.430	9.670	5.368	1.133	1.283	0.376	0.694	3247.700

Figura 14 Estatísticas sobre Colesterol

O intervalo de valores medidos nos pacientes começa em 1.43, e tem como máximo 9.67, com uma média de 5.368 e desvio padrão de 1.133. Para este atributo, fica difícil, sem contexto, perceber que tipo de colesterol o *dataset* se refere. Contudo, acreditamos que se refira a Colesterol total que não deve exceder o valor 2. No caso dos nossos dados, encontramos uma média de 5.34, o que deixa imensas dúvidas.

3.2.12 CREA (Creatina)

A creatina é um composto que, geralmente, aparece muitas vezes no fígado dos mamíferos. Aplicando a mesma análise dos valores anteriores, obtemos a seguinte tabela:

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
CREA	<input type="checkbox"/>	8	1079.100	81.288	49.756	2475.676	15.169	280.100	49992

Figura 15 Estatísticas sobre creatina

Desta tabela vemos que o valor mínimo encontrado é de 8, sendo máximo 1079, com média de 81 e desvio padrão de 49.756. Este caso foi difícil de interpretar, isto porque não sabemos ao certo qual é a

medida escolhida. Contudo, baseados em valores referencia, vimos que este valor deve variar entre 45 a 110. Pelos dados, vemos que há pacientes com valores demasiado alterados.

3.2.13 GGT (Gama Glutamil Transferase)

Seguindo o mesmo processo, observamos a seguinte tabela sobre os valores dos pacientes para o atributo GGT:

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
GGT	<input type="checkbox"/>	4.500	650.900	39.533	54.661	2987.833	5.633	43.713	24312.900

Figura 16 Estatísticas sobre Gama Glutamil Transferase

Da tabela retiramos que o valor mínimo encontrado foi de 4.5, e o máximo de 650.9, a média de 39.533 e o desvio padrão igual a 54.661. Para este atributo, encontramos também uma correlação positiva interessante. Por isso, de forma semelhante à Alanina transaminase, decidimos observar como o atributo se relaciona com o nosso alvo.

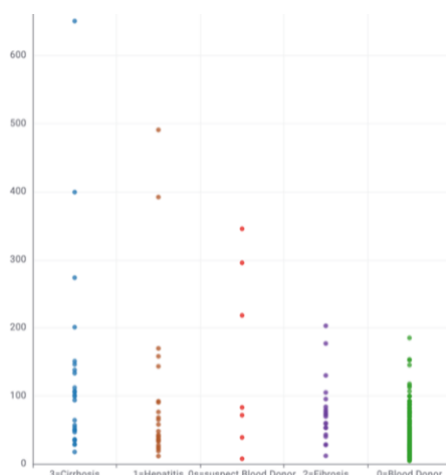


Figura 17 Relação do diagnóstico com os valores de Gama Glutamil Transferase

Através do gráfico, percebemos que os valores de GGT para doadores de sangue não ultrapassa 200, enquanto valores superiores a 200 são geralmente associados a pacientes com diagnósticos de Cirrose e de Hepatite C. Para um individuo saudável, sabemos que deve variar entre 9 a 64 g/L. Tendo em conta os nossos dados, existem casos que fogem muito dos valores de referência

3.2.14 PROT(Proteína)

Por fim, resta ver as mesmas estatísticas que vimos para os outros valores contínuos. Estas estão presentes na tabela que se segue:

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum
PROT	<input type="checkbox"/>	44.800	90	72.044	5.403	29.188	-0.964	3.545	44235.100

Figura 18 Estatísticas sobre proteína

Nesta tabela, percebemos que a média dos valores é de 72.044, com um intervalo de valores que varia entre 44.8 e 90, com um desvio padrão médio é 5.403. Este valor, num humano, deve estar entre 60 a 78.

3.2.15 Valores Numéricos

Sobre os valores expostos, salvo a idade dos pacientes, ficou de parte uma componente bastante importante, mais concretamente, os *outliers*. Para ver estes valores, decidimos observar sob a forma de um *boxplot*.

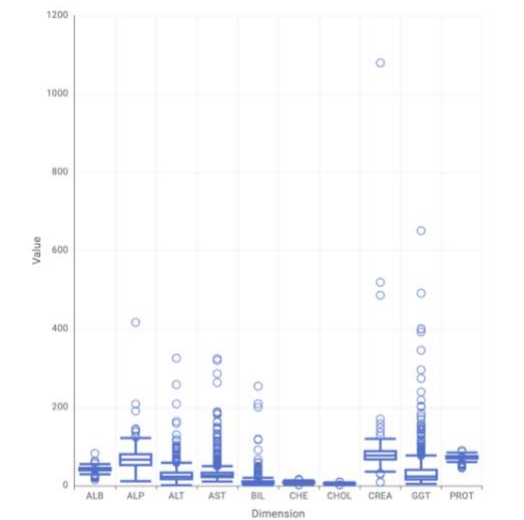


Figura 19 Outliers do conjunto dos dados

Deste gráfico, sobressaem os diversos *outliers* que existem no nosso conjunto de dados. Na medicina, especialmente em casos de humanos não saudáveis, é normal encontrar valores que fogem ao padrão do resto da população. Muitos dos diagnósticos são feitos precisamente graças aos valores que fogem destes padrões. Contudo, podemos sempre encontrar erros nos dados que exponham valores que não podem ser considerados reais.

3.3 Preparação dos Dados

Nesta etapa, iremos referir as transformações que realizamos sobre os dados do *Dataset*, devidamente fundamentadas. Além disso, é pertinente mencionar que alguma preparação foi feita para a secção anterior, sendo que a diferença reside no facto de que a preparação que se sucede é orientada à modelação e não à visualização de dados. Um exemplo pode ser a observação da dispersão dos diagnósticos sobre um atributo, como por exemplo no caso da idade. De seguida, vamos expor que tratamento foi feito aos seguintes atributos:

- **Id:** Não acrescenta conhecimento, portanto, foi removido.
- **Year_of_birth, Month_of_birth, Day_of_birth (data de nascimento):** Estes atributos definem a data de nascimento do paciente que, por sua vez, possui uma correlação negativa perfeita com o atributo **age (idade)**. Desta forma, não faz sentido, do ponto de vista da modelação, manter ambos os atributos. Deste modo, decidimos manter a idade e remover as colunas **Year_of_birth, Month_of_birth, Day_of_birth**. É importante mencionar que é mais realista manter o atributo **age** da época onde o *Dataset* foi contruído, pois a análise feita para chegar aos atributos foi feita nessa altura, e não agora.
- **Sex (sexo):** Esta coluna que tipicamente apresenta um tipo binário, nos nossos dados surge com três tipos únicos. Facilmente percebemos que dois destes tipos representam o sexo masculino e o feminino. O problema está precisamente no terceiro. O terceiro elemento, a nosso ver, poder ser interpretado como um erro de escrita. Por isso, decidimos converter as entradas com o valor 'mm' para 'm'.

- **Birth_Location (local de nascimento):** Para este atributo discreto, é importante lembrar a *Figura 8*, onde fica claro que um paciente ou possui o valor “New Delhi”, ou não possui valor. De certa forma, ficamos numa situação onde o atributo pode ser visto com um valor binário, que pode vir a ser útil para o modelo, especialmente em cenários onde aconteceram surtos. Contudo, a ausência de um valor dá espaço para especulação, pois o campo não estar preenchido não expressa necessariamente que o paciente não é natural de “New Delhi”. Para a pré-preparação, optamos por considerar como “desconhecido”.
- **AST, BIL, CHE, CREA e GGT ->** Estes valores, para o nosso contexto, faz sentido que estejam representados como um número real. Cada atributo deste grupo possui o seu próprio intervalo de valores, que chegam a ser bastante diferentes entre eles, como por exemplo, **AST** possui um valor máximo de 324, enquanto o **CHE** possui um máximo de 16,41. Uma outra questão relevante é a existência de *outliers*. Nenhum dos membros do grupo é especialista no domínio, contudo certos pacientes apresentam valores muito superiores à média, que podem ser prejudiciais para os resultados do modelo.
- **ALB, ALP, ALT, CHOL e PROT->** Estes valores possuem as mesmas características que o grupo anterior, com a agravante de possuírem valores em falta. Em valores contínuos existem diversas formas de lidar com estes problemas. Na nossa opinião, remover as entradas com valores em falta ou preencher os valores são as melhores opções para o problema. Todavia, só testando estas técnicas é que teremos a certeza.
- **Valores numéricos:** Para estes valores, normalizar é uma boa escolha, pois nem todos aparentam estar expressos nas mesmas unidades. Relativamente a *outliers*, é bastante difícil remover entradas no nosso caso, pois existem doentes entre os dados, que podem introduzir valores elevados. Depois de diversas pesquisas, chegamos à conclusão que é muito difícil saber que dados deixar e remover só pelos valores do parâmetro. Assim, vamos experimentar com e sem *outliers*. Estes *outliers*, serão apenas um conjunto reduzido do conjunto real dos *outliers*, precisamente por causa do que tem vindo a ser explicado.

Desta forma, o grupo obteve a seguinte pré-preparação:

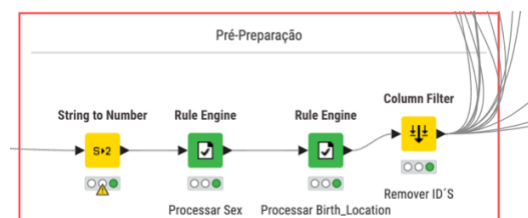


Figura 20 Pré-preparação de dados para o Dataset ímpar

A restante preparação referida nas descrições, como por exemplo, a normalização de dados, surge diferente para cada modelo.

3.4 Modelação

Na fase de modelação, o grupo optou por considerar todas as técnicas estudadas, mesmo aquelas que não se enquadram tanto nos dados do problema. Desta forma, agrupamos os modelos em **Classificação**, **Regressão**, **Regressão Logística com dados Binários** e **Clustering**. A preparação para estes modelos varia, mas, de modo geral, optamos por normalizar valores e tratar valores em falta. Casos como a Regressão logística exigem algum tratamento extra.

3.4.1 Modelos de Classificação

Para os modelos de classificação, consideramos os seguintes modelos sujeitos à mesma preparação, para fins de comparação

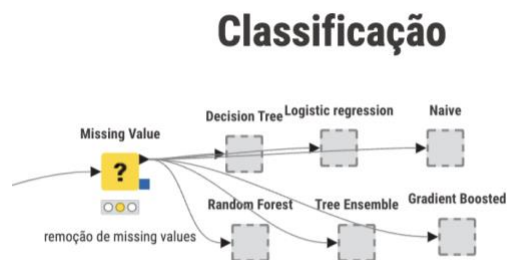


Figura 21 Modelos de classificação

Na figura encontram-se 4 algoritmos que não foram trabalhados nas aulas práticas, mais concretamente, os algoritmos **Naïve Bayes**, **Random Forest**, **Tree Ensemble** e **Gradient Boosted Trees**. Dado que não foram estudados e, por curiosidade e interesse do grupo, foram incluídos no projeto, vamos primeiro sintetizar o seu funcionamento.

- **Naïve Bayes** é um algoritmo de classificação que se baseia em probabilidades sobre as regras de Bayes, onde para cada conjunto de *features* é calculada a probabilidade para todas as classes possíveis. A com maior probabilidade é a selecionada. Este algoritmo assume que as *features* não se correlacionam entre si.
- **Random Forest**, é um algoritmo que constrói diversas árvores decisões durante o treino de forma aleatória.
- **Tree Ensemble**, é uma espécie de floresta que é uma variante do algoritmo anterior, mas com um maior número de configurações.
- **Gradient Boosted Trees**, é também um conjunto de árvores, com a diferença na construção das árvores, que é sequencial, onde as novas árvores corrigem erros feitos pelas árvores anteriores.

3.4.2 Modelos de Regressão

Já nas técnicas de regressão, primeiro é preciso tornar a nossa categoria num atributo contínuo. De facto, este processo não é bastante comum, mas existem métodos que podemos seguir. O método escolhido foi *Label Encoding*, que no fundo consiste em atribuir um valor numérico inteiro por valor único de uma coluna. Desta forma, por exemplo, o diagnóstico de doador de sangue seria o inteiro “1”. Achamos que esta forma é interessante, pois existe uma ordem implícita entre as categorias. Neste caso, a nossa base seria o caso saudável, e o número aumenta com o avanço da doença. Para os modelos, utilizamos os seguintes conjuntos de algoritmos.

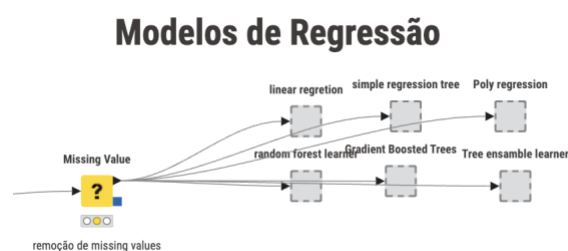


Figura 22 Modelos de regressão

Novamente, utilizamos algoritmos que não exploramos nas aulas práticas da Unidade Curricular. Estes algoritmos funcionam de maneira muito semelhante à sua contraparte de classificação, a diferença reside no facto de a unidade base ser uma árvore de regressão.

3.4.3 Regressão Logística com dados Binários

Por necessidade, achamos que só retirávamos o verdadeiro potencial da Regressão Logística se fizéssemos uma alteração à nossa categoria. Deste modo, como este algoritmo deve ser usado para problemas com alvo binário, decidimos converter o nosso diagnóstico para dois casos possíveis: “Saudável” e “Doente”.

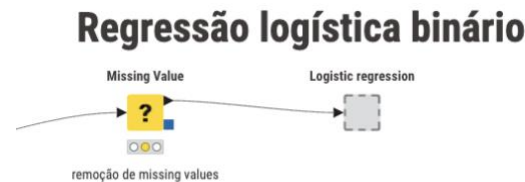


Figura 23 Modelos de regressão logística

3.4.4 Clustering

Para este conjunto de dados, testamos também técnicas de aprendizagem não supervisionada. Para tornar estas medidas eficientes, dada a natureza desequilibrada dos dados, foi preciso um conjunto de técnicas para tornar este processo possível. Para isso, utilizamos técnicas de introdução de dados, e tratamos também o problema de forma binária como forma de teste.

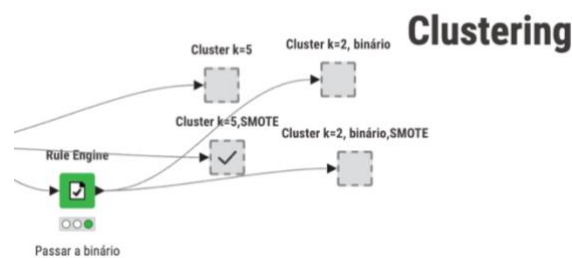


Figura 24 Modelação com Clustering

3.4.5 Redes Neurais Artificiais

Para terminar, resta experimentar com redes neurais. Para este efeito, optamos por utilizar tanto o nodo RProp, que não oferece muita escolha de hiperparâmetros, como redes utilizando os nós DL4j, que são bastante mais manobráveis.

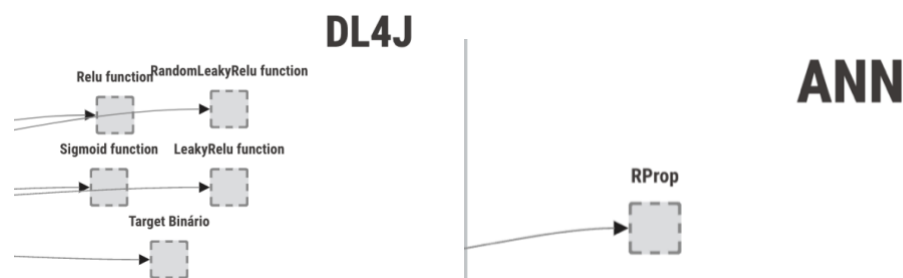


Figura 25 Modelos de redes neuronais

3.5 Avaliação dos Modelos

De modo a avaliar os modelos, decidimos recolher os diferentes resultados obtidos utilizando parâmetros diferentes de modo a conceber as melhores configurações possíveis para cada nó. Além disso, tentamos igualar o consumo computacional de cada algoritmo de modo a manter alguma justiça na avaliação.

3.5.1 Classificação

Valores em falta	Hiperparametro	Algoritmos/Nodos	Hold-out validation		Cross validation	
			Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem valores em falta	Gain Ratio com Poda	Decision Tree	0.941	0.613	0.924	0.604
	Stochastic average gradient	Logistic Regression	0.957	0.767	0.935	0.596
	Default probability = $\sigma = 0,0001$	Naive Bayes Learner	0.924	0.639	0.930	0.656
	100 trees, depth = 40, Info Gain	Random Forest	0.966	0.821	0.941	0.658
	100 trees, depth = 40, Gain Ratio	Tree Ensemble	0.924	0.582	0.951	0.724
	Surrogate	Gradient Boosted Tree	0.941	0.709	0.941	0.659
Mediana	Gain Ratio sem Poda	Decision Tree	0.927	0.700	0.904	0.583
	Stochastic average gradient	Logistic Regression	0.910	0.514	0.923	0.616
	Default probability = $\sigma = 0,0001$	Naive Bayes Learner	0.910	0.602	0.905	0.618
	100 trees, depth = 40, Gain Ratio	Random Forest	0.943	0.730	0.920	0.620
	100 trees, depth = 40, Gain Ratio	Tree Ensemble	0.926	0.664	0.931	0.685
	100 trees, depth = 40, Gini index	Tree Ensemble	0.943	0.739	0.923	0.643
Média	Surrogate	Gradient Boosted Tree	0.935	0.749	0.920	0.645
	Gini index sem Poda	Decision Tree	0.935	0.681	0.907	0.564
	Stochastic average gradient	Logistic Regression	0.910	0.531	0.925	0.622
	Default probability = $\sigma = 0,0001$	Naive Bayes Learner	0.910	0.602	0.901	0.627
	100 trees, depth = 40, Gain Ratio	Random Forest	0.951	0.776	0.928	0.654
	100 trees, depth = 40, Gain Ratio	Tree Ensemble	0.951	0.776	0.930	0.675
Interpolação linear	Surrogate	Gradient Boosted Tree	0.894	0.514	0.913	0.607
	Gain Ratio com Poda	Decision Tree	0.935	0.659	0.909	0.616
	Iteratively	Logistic Regression	0.935	0.734	0.921	0.671
	Default probability = $\sigma = 0,0001$	Naive Bayes Learner	0.918	0.659	0.910	0.633
	100 trees, depth = 40, Info Gain	Random Forest	0.943	0.738	0.927	0.642
	100 trees, depth = 40, Gain Ratio	Random Forest	0.951	0.769	0.927	0.653
Interpolação média	100 trees, depth = 40, Gain Ratio	Tree Ensemble	0.902	0.489	0.939	0.724
	100 trees, depth = 40, Gini index	Tree Ensemble	0.927	0.616	0.928	0.671
	Surrogate	Gradient Boosted Tree	0.910	0.547	0.925	0.655
	Gain Ratio com Poda	Decision Tree	0.935	0.682	0.905	0.602
	Stochastic average gradient	Logistic Regression	0.951	0.761	0.920	0.597
	Default probability = $\sigma = 0,0001$	Naive Bayes Learner	0.935	0.682	0.915	0.655
Interpolação média	100 trees, depth = 40, Gini index	Random Forest	0.926	0.616	0.925	0.643
	100 trees, depth = 40, Info Gain	Tree Ensemble	0.934	0.692	0.926	0.649
	100 trees, depth = 40, Gain Ratio	Tree Ensemble	0.951	0.789	0.936	0.709
	Surrogate	Gradient Boosted Tree	0.926	0.684	0.922	0.640

Tabela 1 Resumo dos resultados dos modelos de classificação

Da tabela de resultados, estão destacados os melhores modelos por variante de tratamento de valores em falta. De modo geral, os valores de *accuracy* foram bastante bons para todos os modelos que desenvolvemos. Como esperado, os modelos construídos com algoritmos que combinam árvores foram os vencedores, destacando as **Random Forest** e **Tree Ensemble**. Contudo, vale a pena compreender resultados de outros modelos. No caso do algoritmo **Naive Bayes**, o resultado foi bom, mas da exploração de dados sabíamos que havia alguma correlação entre atributos, o que prejudica o desempenho. No caso da **Regressão logística**, foi uma surpresa, pois não estávamos à espera que o algoritmo tivesse resultados tão bons dada a natureza binária do algoritmo. Achamos que estes valores estão relacionados com a falta de equilíbrio entre categorias, isto é, existem muitos mais doadores de sangue do que qualquer outra categoria. Para terminar, esperávamos que os modelos de **Gradient Boosted Trees** fossem superar todos os outros, mas sendo que, por motivos de justiça, limitamos os parâmetros igual para todos. Noutro cenário onde recursos computacionais não seriam um constrangimento, o cenário certamente seria diferente. Além disso, reparamos que o tratamento dos valores em falta é bastante relevante, assim como a avaliação de modelos com métodos diferentes. Em muitos casos, a performance do algoritmo ficou aquém na validação cruzada, sobressaindo na validação *hold out*.

3.5.2 Regressão

Avançando para os modelos de regressão, realizamos um levantamento de dados semelhante aos modelos de classificação.

Valores em falta	Hiperparametro	Algoritmos/Nodos	Hold-out validation				Cross validation			
			R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE
Sem valores em falta	//	Linear Regression	0.664	0.325	0.342	0.582	0.627	0.3	0.333	0.577
	Surrogate,40 de profundidade	Simple Regression Tree	0.809	0.068	0.153	0.391	0.729	0.119	0.241	0.491
	100 nodos, 40 profundidade	Random Forest	0.838	0.211	0.232	0.482	0.833	0.155	0.148	0.385
	Fator maximo 5	Polynomial Regression	0.747	0.212	0.091	0.302	0.776	0.252	0.2	0.447
	100 nodos, 40 profundidade	Tree Ensemble	0.830	0.176	0.187	0.433	0.809	0.158	0.17	0.412
	100 nodos, 40 profundidade	Gradient Boosted Tree	0.665	0.174	0.312	0.558	0.681	0.162	0.284	0.533
Mediana	//	Linear Regression	0.629	0.396	0.404	0.635	0.562	0.407	0.485	0.696
	Surrogate,40 de profundidade	Simple Regression Tree	0.773	0.138	0.285	0.533	0.61	0.187	0.431	0.656
	100 nodos, 40 profundidade	Random Forest	0.723	0.221	0.319	0.564	0.793	0.216	0.229	0.479
	Fator maximo 5	Polynomial Regression	0.541	0.338	0.392	0.626	0.667	0.338	0.368	0.606
	100 nodos, 40 profundidade	Tree Ensemble	0.799	0.200	0.199	0.446	0.784	0.219	0.239	0.489
	100 nodos, 40 profundidade	Gradient Boosted Tree	0.815	0.135	0.219	0.468	0.713	0.173	0.317	0.563
Média	//	Linear Regression	0.544	0.403	0.400	0.633	0.561	0.408	0.485	0.697
	Surrogate,40 de profundidade	Simple Regression Tree	0.567	0.179	0.374	0.612	0.572	0.190	0.473	0.688
	100 nodos, 40 profundidade	Random Forest	0.849	0.167	0.154	0.392	0.782	0.223	0.242	0.491
	Fator maximo 5	Polynomial Regression	0.580	0.342	0.398	0.631	0.622	0.341	0.374	0.612
	100 nodos, 40 profundidade	Tree Ensemble	0.751	0.159	0.154	0.392	0.781	0.219	0.242	0.492
	100 nodos, 40 profundidade	Gradient Boosted Tree	0.631	0.253	0.406	0.637	0.681	0.183	0.353	0.594
Interpolação linear	//	Linear Regression	0.524	0.413	0.45	0.671	0.563	0.409	0.484	0.695
	Surrogate,40 de profundidade	Simple Regression Tree	0.407	0.26	0.715	0.864	0.644	0.166	0.393	0.627
	100 nodos, 40 profundidade	Random Forest	0.827	0.196	0.2	0.447	0.795	0.209	0.226	0.479
	Fator maximo 5	Polynomial Regression	0.631	0.329	0.407	0.638	0.735	0.321	0.293	0.542
	100 nodos, 40 profundidade	Tree Ensemble	0.799	0.213	0.229	0.479	0.788	0.208	0.235	0.484
	100 nodos, 40 profundidade	Gradient Boosted Tree	0.545	0.097	0.237	0.487	0.689	0.176	0.344	0.586
Interpolação média	//	Linear Regression	0.502	0.409	0.386	0.621	0.562	0.407	0.484	0.696
	Surrogate,40 de profundidade	Simple Regression Tree	0.685	0.122	0.317	0.563	0.582	0.185	0.462	0.68
	100 nodos, 40 profundidade	Random Forest	0.736	0.217	0.21	0.459	0.786	0.213	0.237	0.487
	Fator maximo 5	Polynomial Regression	0.754	0.357	0.342	0.585	0.664	0.34	0.372	0.61
	100 nodos, 40 profundidade	Tree Ensemble	0.723	0.182	0.173	0.416	0.788	0.213	0.235	0.485
	100 nodos, 40 profundidade	Gradient Boosted Tree	0.483	0.189	0.439	0.663	0.633	0.212	0.406	0.637

Tabela 2 Resumo dos resultados dos modelos de regressão

Para estes modelos, a seleção do melhor desempenho depende de um conjunto de métricas. Neste caso, é preciso estar atento aos 3 valores que definem os erros dos modelos. De modo geral, os algoritmos que melhor sobressaíram foram os algoritmos **Random Forest** e **Tree Ensemble**, semelhante às suas contrapartes de classificação. Observámos também que estes modelos são muito mais sensíveis ao tratamento de *missing values*, comparativamente aos modelos de classificação. Olhando agora para as métricas de avaliação, achamos que, para o nosso problema, o MSE e o RMSE são os fatores mais importantes. O RMSE é um parâmetro que atribui um peso maior a erros grandes, o que no nosso caso, pode implicar diagnósticos muito longe do esperado. O pior destes cenários seria um diagnóstico de saudável a um individuo que está doente. A nível de avaliação e treino, a validação cruzada superou a validação *hold out* precisamente porque os modelos treinados apresentavam valores inferiores de RMSE.

3.5.3 Regressão Logística com alvo Binário

Para a regressão logística com alvo binário, conseguimos os seguintes valores:

Valores em falta	Hiperparametro	Algoritmos/Nodos	Hold-out validation		Cross validation	
			Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem valores em falta	Stochastic average gradient	Logistic Regression	0.975	0.829	0.979	0.876
Mediana	Stochastic average gradient	Logistic Regression	0.951	0.758	0.960	0.802
Média	Stochastic average gradient	Logistic Regression	0.910	0.531	0.925	0.622
Interpolação linear	Stochastic average gradient	Logistic Regression	0.943	0.709	0.951	0.752
Interpolação média	Stochastic average gradient	Logistic Regression	0.935	0.606	0.960	0.804

Tabela 3 Resumo dos resultados da regressão logística

Os resultados foram bastante interessantes. Não só conseguimos perceber que este algoritmo é sensível ao tratamento que damos aos valores em falta, como também é um algoritmo que produz uma *accuracy* e um *kappa* bom para o nosso problema. Desta forma, caso não haja necessidade de um diagnóstico que inclua o avanço da doença, é uma excelente opção.

3.5.4 Clustering

Sendo a categoria um atributo conhecido para todas as entradas, não é natural o uso de técnicas de aprendizagem não supervisionada como vamos expor agora.

Valores em falta	Valores	Hiperparâmetro	Algoritmos/Nodos	Métrica face ao grupo total		Avaliação	
				Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem valores em falta	Com adição	5 clusters, 10 000 iterações, SMOTE	K-Means	0.500	0.375	0.336	0.048
	Com adição	5 clusters, 10 000 iterações, Euclidean, SMOTE	K-Medoids	0.598	0.498	---	---
	Com adição	2 clusters, 10 000 iterações	K-Means	0.579	0.081	0.677	0.053
	Com adição	2 clusters, 10 000 iterações, Euclidean	K-Medoids	0.951	0.758	----	---
	Com adição	2 clusters, 10 000 iterações, SMOTE	K-Means	0.721	0.443	0.902	0.519
	Com adição	2 clusters, 10 000 iterações, Euclidean, SMOTE	K-Medoids	0.598	0.197	----	---

Tabela 4 Resumo dos resultados de Clustering

A partir desta tabela, conseguimos perceber que não é um método que fornece resultados muito bons. No caso do domínio binário, os resultados foram interessantes, mas não superam a aprendizagem supervisionada. Achamos importante referir que este algoritmo só foi viável com a utilização de técnicas de *SMOTE* para equilíbrio do atributo alvo, e que pode acrescentar um pouco de processamento extra. De modo geral, achamos que a experiência foi academicamente proveitosa, mas não se ajusta ao nosso problema.

3.5.5 Redes Neurais

Apresentamos também uma análise detalhada dos resultados da aplicação de redes neurais artificiais, recorrendo a algoritmos como o Rprop e o DL4J:

Valores em falta	Hiperparâmetro	Algoritmos/Nodos	Hold-out validation		Cross validation	
			Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem valores em falta	1000 iterações, 4 camadas, 8 nodos em cada	RProp	0.958	0.785	0.954	0.768
Interpolação linear	1000 iterações, 4 camadas, 8 nodos em cada	RProp	0.919	0.648	0.924	0.681
Interpolação média	1000 iterações, 4 camadas, 8 nodos em cada	RProp	0.943	0.747	0.912	0.642
Mediana	1000 iterações, 4 camadas, 8 nodos em cada	RProp	0.935	0.719	0.930	0.707
Média	1000 iterações, 4 camadas, 8 nodos em cada	RProp	0.935	0.718	0.928	0.694
Sem valores em falta, cargas balanceadas	100 iterações, 3 camadas, 10 epochs, Relu	DL4J Feedforward	0.906	0	-----	-----
Sem valores em falta, Valores transformados em Binário, cargas balanceadas	10 iterações, 3 camadas, 1 epochs, Relu	DL4J Feedforward	0.906	0	-----	-----
Sem valores em falta, Valores transformados em Binário	10 iterações, 3 camadas, 1 epochs, Relu	DL4J Feedforward	0.898	0	0.904	0

Tabela 5 Resultados das Redes neurais

Antes de analisar os resultados, já se esperava que os algoritmos de RNA apresentassem um bom desempenho, não só devido ao potencial dos modelos utilizados, bem como ao facto de que estes algoritmos se encaixam bem no problema em questão, permitindo lidar com sua complexidade de forma eficaz. E, olhando para a tabela, vemos que os resultados não fugiram muito daquilo que foi a nossa previsão. Já para o algoritmo DL4J, a modelação não foi um sucesso. Acreditamos que o problema está

no volume reduzido de dados do *dataset*. Mesmo com técnicas de *SMOTE*, independentemente dos parâmetros selecionados, os resultados eram sempre os mesmos. Aqui o que importa não é *accuracy* mas sim a matriz confusão produzida por este algoritmo, onde todas as previsões acabavam sempre no mesmo valor alvo. Este cenário repetiu-se mesmo depois de converter a coluna *target* para binário.

3.5.6 Sem Outliers

Finalmente, decidimos experimentar com o melhor algoritmo, sem grandes restrições no que toca a performance, o quão bom poderia ser o modelo desenvolvido, e também tentar trabalhar a questão dos *outliers*. Como referido anteriormente, é natural encontrar *outliers* em questões que envolvam saúde, por isso, o tratamento não foi muito ríspido, mas serviu para limpar apenas as entradas com valores muito mais elevados que os demais.

Valores	Hiperparametro	Algoritmos/Nodos	Hold-out validation		Cross validation	
			Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem adição/remoção	1000 trees, sem limite, Info Gain	Tree Ensemble	0.957	0.776	0.951	0.741
Remoção Outliers	1000 trees, sem limite, Info Gain	Tree Ensemble	0.989	0.829	0.980	0.519

Tabela 6 Resultados de modelos extras

Desta experiência, primeiramente, reparamos que um maior número de árvores melhorou a *accuracy* em troco de performance, tal como esperado. O mesmo se pode apontar aos *outliers*, que após removidos, melhorou o desempenho do algoritmo. Este caso concreto não pode ser visto como algo totalmente verdadeiro, pois é difícil para o grupo fundamentar quais ou que valores levam a que um paciente seja removido. Além disso, remover estes pacientes criou desequilíbrio nos dados, onde pacientes doentes ficaram ainda com menos presença.

4 Tarefa Dataset Escolhido

4.1 Estudo de negócios

Para esta tarefa, escolhemos um [Dataset](#) com dados retirados do *website* Craigslist, que se dedica à venda de carros usados nos EUA. Com estes dados, esperamos construir um modelo que, baseado em características dos veículos como o ano, modelo, fabricante, entre outros, consiga chegar ao preço de venda.

4.2 Estudo de Dados

4.2.1 Descrição de dados

O *Dataset* possui um total de 426880 entradas e 22 colunas por entrada. Também sabemos que existem valores que estão em falta, mas esses detalhes ficam para a especificação detalhada dos

elementos das colunas. É importante mencionar que removemos algumas colunas referentes a *urls* e *descrições* dos anúncios.

1. **id**: Código identificador do anúncio
2. **region**: Representa a região onde o veículo se encontra
3. **year**: Ano de montagem
4. **manufactor**: Fabricante
5. **model**: Modelo do carro
6. **condition**: Condição do veículo
7. **cylinders**: Número de cilindros do motor
8. **fuel**: Combustível
9. **odometer**: Número de milhas do carro
10. **title_status**: Estado do veículo
11. **transmisson**: Transmissão
12. **VIN**: Número de série
13. **drive**: Tração do carro
14. **size**: Tamanho
15. **type**: Tipo de carro
16. **paint_color**: cor da pintura do carro
17. **county**: Condado
18. **state**: Estado
19. **lat**: Latitude
20. **long**: Longitude
21. **posting_date**: Data do anúncio
22. **price**: Preço de compra

O atributo alvo é o **price**. Será esta coluna que o nosso modelo deve, após o treino, conseguir realizar uma previsão sobre.

Vamos agora especificar cada atributo de forma semelhante ao *Dataset* da tarefa anterior.

4.2.2 Id, lat, long, county

Decidimos agrupar estes atributos, pois não introduzem conhecimento útil para os nossos dados. No caso do id, serve apenas para identificar cada anúncio. No caso da latitude e longitude, é uma informação bastante específica para o que se deseja. No mesmo *Dataset*, temos o atributo região e estado que oferecem a mesma informação de uma forma muito mais interessante que estes atributos. Resta mencionar o caso da coluna condado que, à partida, parece relevante, mas todos os anúncios possuem este campo vazio. Por estes motivos, decidimos remover estas *features* já na fase do estudo de dados.

4.2.3 Region (Região) e state (Estado)

No processo de exploração, decidimos que analisar estes dois atributos ao mesmo tempo seria uma melhor opção. Ao fim e ao cabo, são dois atributos que expressam a localização geográfica do veículo. À partida,

achamos que a localização do veículo pode de facto impactar o preço de venda. Por exemplo, um veículo do tipo “convertible”, conhecido em português como um descapotável, pode ser mais apetecível por compradores que habitam em regiões de mais calor, como na região costeira e sul do país.

Na totalidade dos dados, existem 404 valores para o campo região e 51 para os estados. Para o caso dos estados, facilmente vemos que este valor é correto (50 estados mais 1 distrito federal), mas a questão da região é mais complexa, pois não existe uma listagem oficial das regiões do país.

4.2.4 Year (Ano de montagem)

Antes de visualizar os dados, é quase certo que a idade será um fator impactante no preço de venda dos carros.

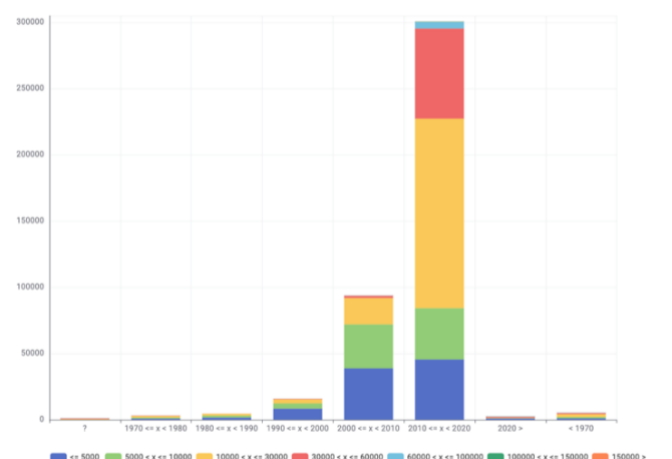


Figura 26 Preços por década de montagem do carro

Do gráfico conseguimos retirar que, a grande maioria dos carros que se encontra à venda é na segunda década do século 21, e os valores encontram-se especialmente no intervalo de 10000 a 30000 dólares americanos. Além disso, para as restantes décadas, salvo veículos mais antigos do que 1980, encontram-se mais carros com valores menores a 5000 dólares do que a outro preço.

4.2.5 Manufactor (fabricante) e model (modelo)

Estes dois atributos representam a marca e o modelo do veículo em questão. Antes de ver os dados, esperávamos que ambos atributos tivessem um impacto notável no preço de venda. A verdade é que muitos fabricantes possuem diferentes gamas que ocupam diferentes preços, salvo certas exceções de marcas que são consideradas de luxo. O campo dos modelos possui um total de 29668 entradas únicas. Isto deve-se ao facto de o *website* não forçar os utilizadores a colocarem os modelos de um set já definido. Por este motivo, achamos que o modelo não vai conseguir. As marcas têm o problema de oferecer várias gamas, por isso decidimos, a partir da marca, reduzir para dois casos: Gama e Local de fabrico.

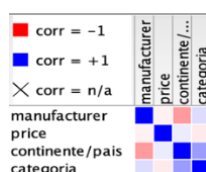


Figura 27 Correlação entre Gama, Local de Fabrico e preço

A lógica por de trás desta decisão reside no facto de os custos de importação de marcas estrangeiras serem significativo, e também porque a gama acaba por ser a associação que fazemos quando pensamos numa marca. Deste modo, chegamos aos seguintes resultados:

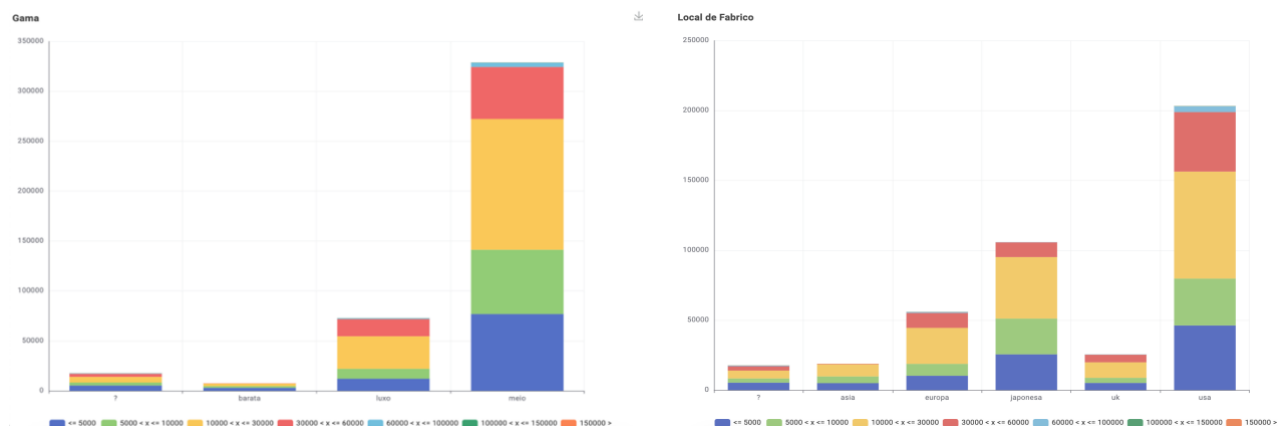


Figura 28 Distribuição de preços por gama e local de fabrico

Destes gráficos podemos ver que os preços variam bastante independentemente da categoria. Contudo, fica uma distribuição mais interessante e fácil de ler do que os atributos antigos. Vimos ainda sem nenhum tipo de redução como é que a dispersão acontecia com o fabricante.

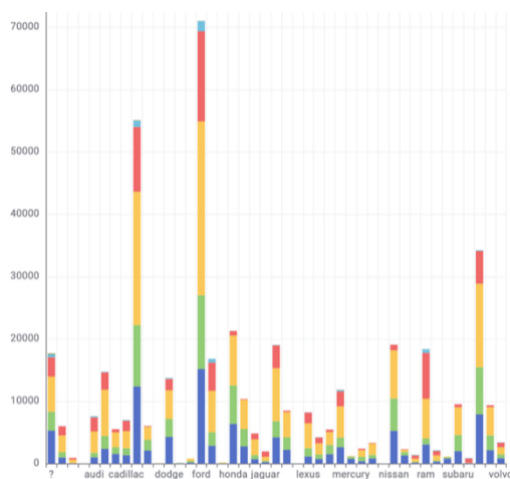


Figura 29 Distribuição de preços por marca

Deste gráfico, retiramos que existem marcas que oferecem veículos mais caros que outras, por isso pode ser um atributo importante para o modelo.

4.2.6 Condition e title_status

A condição e o estado do veículo são dois atributos semelhantes, na medida que descrevem a situação onde se encontra o veículo.

Column	Exclude Column	No. missings	Unique values	All nominal values	Frequency Bar Chart
condition	<input type="checkbox"/>	174104	6	good, excellent, like new, fair, new, salvage	
title_status	<input type="checkbox"/>	8242	6	clean, rebuilt, salvage, lien, missing, parts only	

Figura 30 Condição e estatuto do veículo

Nas seguintes figuras conseguimos averiguar que os valores para o atributo condição descrevem de forma qualitativa com uma ordem, enquanto o título é bastante mais objetivo. Duas particularidades que visualizámos são a condição e título “salvage” e “clean”. Não só aparece como em ambos os campos (que representam coisas diferentes) como existem veículos que descrevem a condição com este atributo, mas colocam algo diferente como estado do veículo.

4.2.7 Cylinders (Número de cilindros do motor)

Uma das características mais relevantes sobre os motores dos carros é o número de cilindros. Geralmente, um número de cilindros mais elevado está diretamente relacionado com a potência e capacidades do motor.

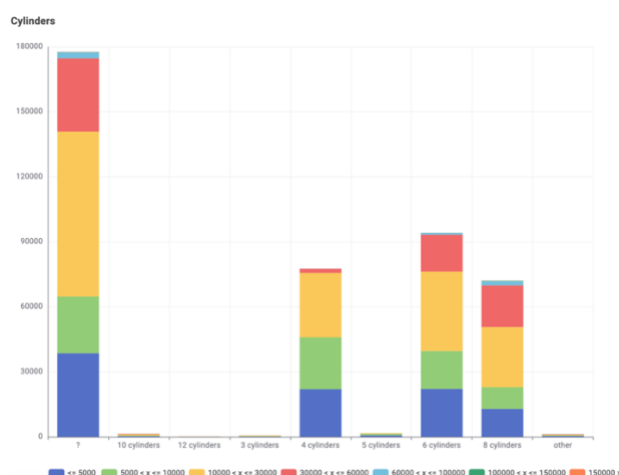


Figura 31 Distribuição de preços por número de cilindros

Do gráfico conseguimos perceber que o número de cilindros varia entre 3 e 12, com dois casos onde não é especificado. Nestes dois casos, idealmente só surgiam casos onde os veículos são elétricos, mas infelizmente não é o caso. Ainda do gráfico, conseguimos observar uma relação entre o número de cilindros e o preço do veículo, isto é, preços mais altos estão ligados a veículos com motores com mais cilindros.

4.2.8 Fuel (Combustível)

O tipo de combustível é, por norma, um atributo importante para a escolha de um veículo. No mercado automóvel, existe uma pequena seleção de combustíveis que são usados pelos carros, com algumas exceções, como motores a hidrogénio ou até a biomassa.

Column	Exclude Column	No. missings	Unique values	All nominal values	Frequency Bar Chart
fuel	<input type="checkbox"/>	3013	5	gas, other, diesel, hybrid, electric	

Figura 32 Tipos de combustível

Da figura retiramos os 3 principais combustíveis de um carro, a sinalização para caso seja híbrido e um caso que deve agrupar carros que fogem ao que é esperado para veículos regulares. Contudo,

observámos uma situação parecida ao caso anterior onde encontrávamos veículos de todos os tipos, algo bastante indesejável.

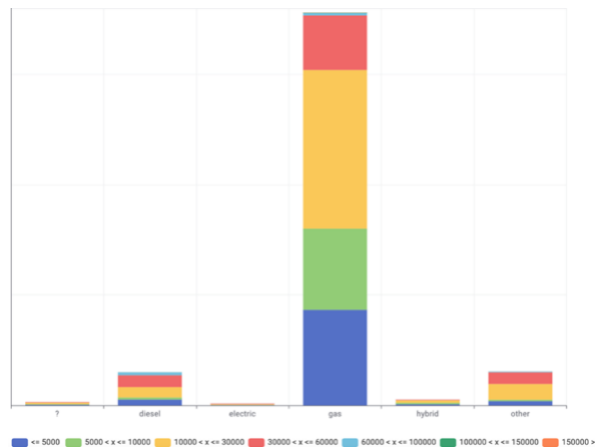


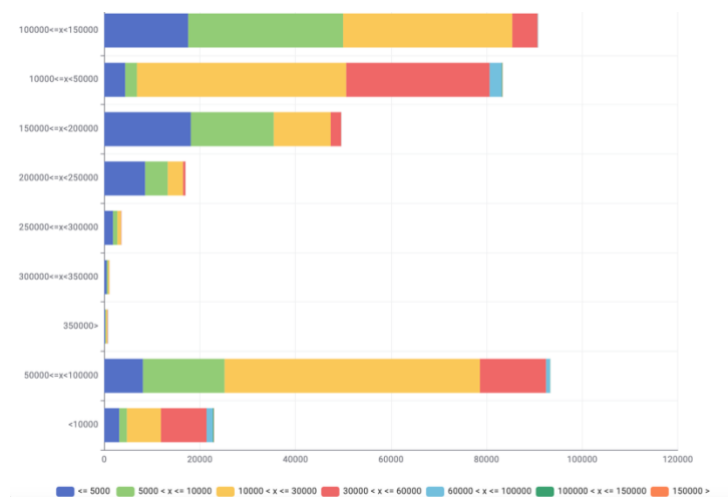
Figura 33 Distribuição de preços por tipo de combustível

Do gráfico retiramos que o combustível mais popular é a gasolina, e que, para carros que não sejam a gasolina, observamos uma tendência para preços mais altos.

4.2.9 Odometer (Número de milhas do carro)

A distância feita por um veículo é determinante para a continuidade da sua função como meio de transporte. Este atributo é dos mais importantes para determinar o preço de venda, o que pode ser comprovado com o valor elevado de correlação com o preço.

A primeira observação que deparamos ao analisar os dados é que existem veículos com valores absurdos, que ultrapassam muitas vezes o valor médio esperado por um veículo. Desta forma, posteriormente será preciso tomar medidas face a este problema.



estamos à espera que a esmagadora maioria dos carros listados sejam automáticos. Observando os dados, existem 3 valores únicos, sendo estes: automático, manual e outro.

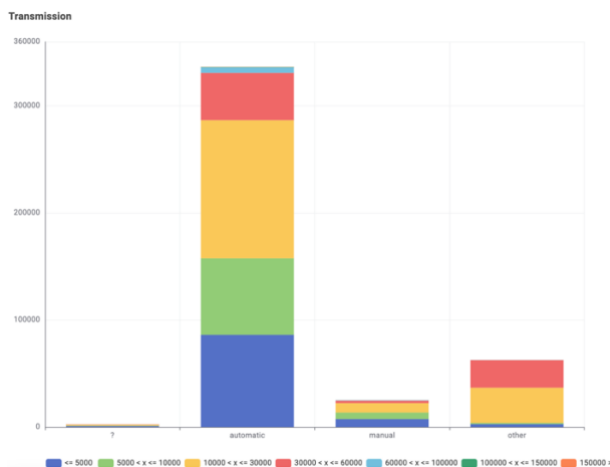


Figura 35 Distribuição de preço por Transmissão

No diagrama de barras podemos ver que não há um grande padrão de distribuição dos preços dos veículos. Vemos que, como esperado, carros automáticos são mais populares do que os outros e que os carros mais caros também costumam ser automáticos. O atributo outro expressa uma relação que nós consideramos perigosa, pois muitos veículos que se encontram neste grupo estão falsamente atribuídos.

4.2.11 VIN: Número de série

O número de série é um aspeto que serve para identificar um veículo. Desta forma, esperamos que o comportamento seja algo semelhante ao atributo *id*. Contudo, facilmente vemos que existem imensos valores em falta e, talvez mais grave ainda, existem anúncios onde este valor está repetido. O facto de estar repetido, pode significar várias coisas para o comprador, mas certamente não deveria ser um atributo relevante para um modelo.

4.2.12 drive: Tração do carro

A tração do veículo, à primeira vista, não pareceu ser o atributo mais relevante. Porém, a correlação deste atributo com o preço final de venda era superior ao que nós imaginávamos, por isso decidimos averiguar e gastar algum tempo com este atributo. Antes de mais, os veículos possuem 3 trações possíveis, sendo estas tração dianteira, traseira e às quatro rodas.

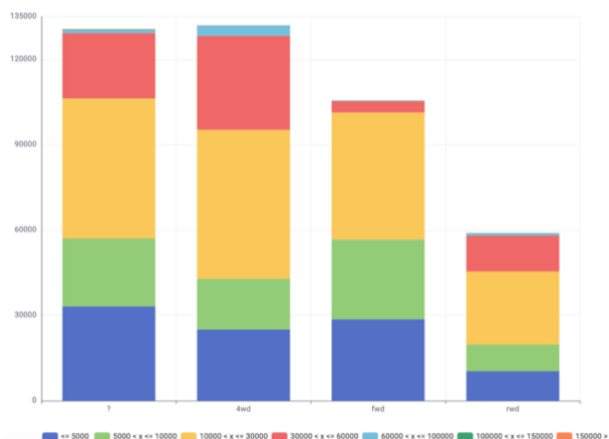


Figura 36 Distribuição de preços por tração

Deste gráfico podemos ver que existem muitas entradas sem a tração especificada, e que, proporcionalmente, veículos com tração traseira são mais baratos que veículos com outras trações.

4.2.13 *Type* e *size* (Tipo de carro e tamanho do carro)

Para este atributo, começamos por ver quais os tipos de veículos que aparecem para venda.

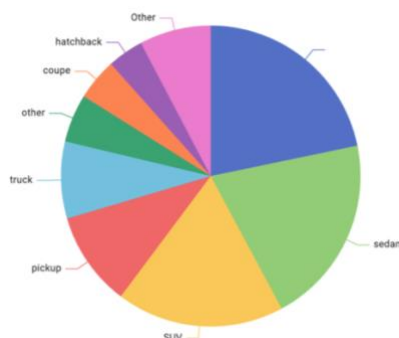


Figura 37 Tipos de veículos

Através da seguinte imagem, conseguimos perceber que o tipo “sedan” e “SUV” são os mais populares, e que existem muitos anúncios com o tipo em falta ou não especificado. A correlação deste atributo com o preço é baixa, isto pode ser explicado pelo facto de existirem diferentes gamas de preços para cada tipo, assim como anúncios.

Relativamente ao tamanho, é ainda menos significativo do que o tipo, pois não só existem menos tipos, como acontece a mesma situação de existirem muitas gamas diferentes para cada tamanho.

4.2.14 *paint_color*: cor da pintura do carro

Antes de explorar os dados, não achávamos que a cor do veículo fosse um aspeto relevante para o preço do veículo. Contudo, o valor da correlação mostrou pouca, mas alguma relação.

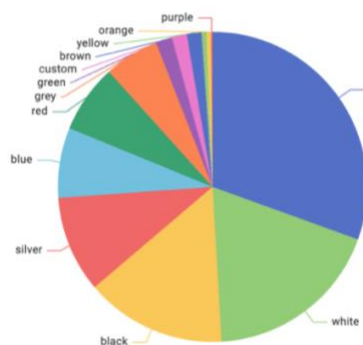


Figura 38 Diferentes cores dos dados

Do gráfico percebemos que existe variedade nas cores, onde o branco e preto são as mais populares. Existe também um número muito grande de anúncios onde a cor não está especificada. Isto muito possivelmente deve-se ao facto de o *website* permitir o *upload* de fotografias do veículo. De resto, não vimos alguma relação particular entre a cor e o preço.

4.2.15 *posting_date* (Data do anúncio)

A data do anúncio pode ser algo com impacto no preço do veículo, pois é possível fazer uma ligação entre a data atual para o estado da economia e do mercado naquela altura. No entanto, nesta amostra de dados,

o intervalo de valores não é significativo para tirar esse tipo de conclusões. Todas as amostras são do ano de 2021 e que se encontram no mês de Abril e Maio. Por este motivo, achamos que não vai ser um atributo relevante para o modelo.

4.2.16 Price (preço)

Finalmente, é muito importante explorar como é que o nosso atributo alvo varia de anúncio para anúncio. Começamos por ver alguns dados estatísticos sobre este atributo.

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis
price	<input type="checkbox"/>	0	3736928711	75199.033	12182282.174	148407998957166.094	254.407	69205.089

Figura 39 Primeira exploração do preço

Da tabela conseguimos perceber que existem valores que não são realistas para carros novos, muito menos para carros usados. Por isso, decidimos fazer uma limpeza para remover estes valores.

A seleção feita para a exploração foi manual, onde conseguimos excluir os anúncios falsos que danificavam a qualidade dos dados. Agora conseguimos perceber que a média de preços é de 17545 dólares, com um desvio padrão médio de 15668. Um aspeto importante de referir é o valor “0”. O que o grupo entendeu, é que de facto à indivíduos que querem ver se livres dos carros, mas também existem casos onde o anunciante, ou errou a inserir os valores, ou quer atrair cliques para o seu anúncio.

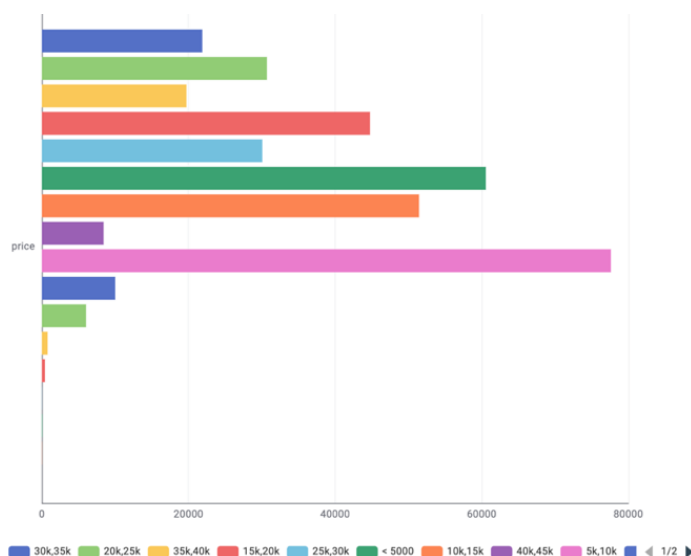


Figura 40 Distribuição do preço

4.3 Preparação dos Dados

À semelhança do ponto anterior, esta secção serve para mostrar as transformações que realizamos sobre os dados do *Dataset*. Desta forma, os seguintes pontos visam traduzir essas questões.

- **Id:** Mero identificador, não caracteriza os veículos há venda.
- **lat,long,county:** Não consideramos relevantes dado ao facto de existir atributos como a região e o estado. De certa forma, estes atributos estão a introduzir um conhecimento demasiado específico.
- **Region e state:** Ao contrário dos atributos anteriores, estas referências espaciais são muito mais interessantes. Contudo por a região continua a ser demasiado específico e irrelevante, dado que os impostos são definidos por estado.

- **Year:** Esta *feature* é das mais relevantes. Contudo, existem entradas que não parecem corretas, especialmente as que se encontram antes do ano de 1903 como exposto no ponto 3.2.4. Deste modo vamos excluir estas entradas do treino do nosso modelo.
- **Manufacturer e model:** O atributo modelo é uma questão muito difícil de lidar. Como foi referido, existem várias formas de representar o mesmo modelo, e também existem muitos modelos por marca. No caso da marca, encontramos um atributo com um valor de entradas únicas mais aceitável. Para a exploração, tentamos aplicar reduções no conhecimento para respostas que ocupassem um número menor de casos. No caso da gama, a escolha não foi fundamentada, baseamos nos apenas no que tipicamente associamos a uma marca. No caso do continente, acabo por ser o país/continente de origem da marca. Estes últimos atributos apresentam melhor correlação com o preço do que a marca em si. Contudo achamos que é perigoso colocar o atributo gama, pois não têm uma base segura.
- **Condition e title_status:** No ponto 3.2.6, conseguimos ver que existem alguns problemas que têm de ser resolvidos. Baseados nos dados levantados, o atributo title_status parece ser o mais relevante, isto porque é um título atribuído por uma entidade reguladora e não por um particular. Deste modo, decidimos remover a condição do veículo.
- **Cylinders:** Este atributo discreto possui uma correlação interessante com o preço. Infelizmente, nem todos os carros possuem esta característica, como por exemplo, os carros elétricos. Por isso, o campo “other” não é muito conclusivo. Relativamente aos valores em falta, o número de entradas é muito elevado e, por essa razão, decidimos não remover.
- **Fuel:** Este atributo é bastante útil e requer pouco tratamento. O único problema neste caso será o caso dos missing values que pode ou não ser removido.
- **Odometer:** Dada a dispersão de valores que conseguimos ver no ponto 3.2.9, é difícil perceber se vale a pena ou não fazer tratamento de *outliers*. O que esperávamos é que a partir de um determinado número de milhas o preço dos veículos fosse tão pouco que não haveria diferenças. Pelos gráficos, vemos que as proporções são bastante semelhantes entre os diferentes intervalos. Por isso, vamos apenas remover valores que consideramos falsos ou demasiado elevados para o que é esperado para a vida de um veículo. O valor escolhido foi de 600000 milhas. Este valor surge por que ainda é plausível do ponto de vista mecânico do carro, e também porque, em casos de veículos de trabalho, por vezes chega-se a valores bastante altos. Este valor deve ser normalizado.
- **Transmission:** Para este atributo, não é preciso fazer nada.
- **Vin:** Este atributo funciona como um identificador para os veículos. Tendo em conta a exploração, perceberemos que não é muito útil e, em muitos dos casos, não está sequer preenchido.
- **Drive:** Muito semelhante ao caso da transmissão, não modificamos o campo.
- **Type e size:** Baseando-nos no ponto 3.2.13, achamos que não acrescenta nada ao modelo, isto porque os valores de preço estão muito bem distribuídos por estes campos.
- **Paint_color:** Como esperado, a cor não é um fator muito relevante para o preço de um carro. Mais sobre esta questão pode ser vista em 3.2.14.
- **Posting_date:** Não é relevante para o problema, pois o intervalo de dados do *Dataset* é bastante curto.
- **Price:** Curiosamente, o próprio preço precisa de um pequeno ajuste. Isto deve-se ao facto de vários utilizadores colocarem valores desproporcionais, como está explícito em 3.2.16. Desta forma, achamos que o valor 566567 é plausível como máximo, pois corresponde a um automóvel de luxo. Também consideramos inaceitável um veículo ser listado com o valor “0”.

Posto isto, no Knime, conseguimos chegar à seguinte preparação de dados que será comum a todos os modelos. É de realçar que, dependendo do modelo, pode ser requerido alguns cuidados extras.

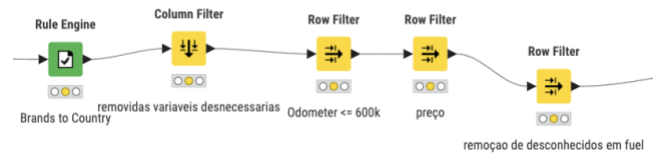


Figura 41 Pré-preparação de dados

4.4 Modelação

Para a modelação deste *dataset*, utilizamos tanto técnicas de regressão como técnicas de classificação. Além disso, achamos também interessante experimentar com aprendizagem não supervisionada, mais concretamente, técnicas de *clustering*. Quando possível, utilizamos dois tipos de validação, sendo estes a validação cruzada e a validação *hold out*.

Um aspeto importante na nossa modelação foi a redução de dados. O grupo sentiu a necessidade de baixar o volume dos dados que estava a ser usado para treino por causa de limites de recursos computacionais. Para a grande maioria dos nodos, conseguimos testar bem com metade dos dados. Para outros nós mais exigentes, a redução teve de ser ainda maior.

4.4.1 Modelos de Regressão

Começando com as técnicas de regressão, utilizamos os seguintes modelos com os nós de regressão do Knime. Estes são exatamente os mesmos modelos que usamos para o *dataset* anterior, apenas com algumas variações para facilitar a visualização de dados e também a nível de parâmetros.

Modelos de Regressão

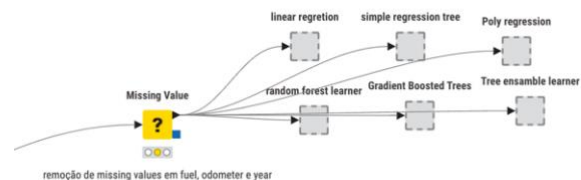


Figura 42 Modelos de regressão

4.4.2 Modelos de Classificação

No que toca a técnicas de classificação, dada a natureza numérica do nosso atributo alvo, precisamos de fazer a conversão para intervalos. A discretização do atributo preço foi uma abordagem difícil, pois não conseguíamos escolher entre técnicas que usam intervalos iguais e técnicas que usam intervalos da mesma frequência. Assim, optamos por utilizar as duas. Deste modo, os modelos que utilizamos foram os seguintes:

Classificação

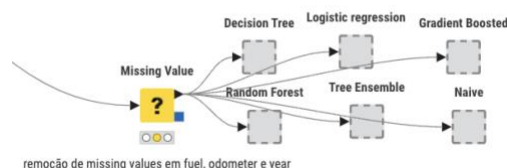


Figura 43 Modelos de classificação

4.4.3 Clustering

Utilizamos também as técnicas de *clustering* para tentar resolver o nosso problema. Para esta técnica, é preciso também discretizar o nosso atributo alvo, de forma semelhante aos modelos de classificação. Os Modelos desenvolvidos foram os seguintes:

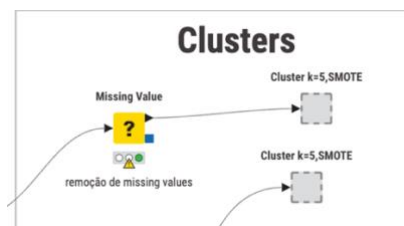


Figura 44 Modelos de Clustering

4.4.4 Redes Neurais Artificiais

Recorremos também às técnicas de RNA para fazer previsões para o nosso modelo, mais propriamente os algoritmos de Rprop e DL4J. A composição foi a seguinte:



Figura 45 Modelos de redes neurais

4.4.5 Modelação sem Outliers

Achamos relevante, dada a natureza do domínio, experimentar modelar sem *outliers*. Esta decisão é relevante, pois, entre os dados, existem situações excepcionais que podem estar a prejudicar os resultados dos modelos.

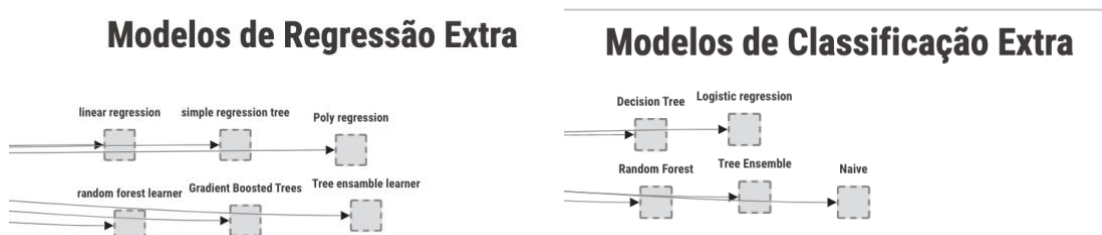


Figura 46 Modelos sem outliers

4.5 Avaliação dos Modelos

Novamente, chegou a fase de avaliar os modelos construídos. De forma semelhante ao *dataset* anterior, vamos observar os resultados obtidos através dos 4 tipos de modelos, começando desta vez com a regressão.

4.5.1 Regressão

Valores em falta	Hiperparametro	Algoritmos/Nodos	Hold-out validation				Cross validation			
			R ²	MAE	MSE	RMSE	R ²	MAE	MSE	RMSE
Sem valores em falta	//	Linear Regression	0.505	6.620	1.2E8	10.976	0.508	6.663	1.16E8	10.781
	Surrogate, 40 de profundidade	Simple Regression Tree	0.354	5.769	1.48E8	12.199	0.408	5.646	1.40E8	11.835
	25 nodos, 40 de profundidade	Random Forest	0.644	5.059	9.0E7	9.497	0.678	4.989	7.6E7	8.729
	Fator maximo 5	Polynomial Regression	0.367	8.037	1.41E8	11.893	0.348	8.077	1.54E8	12.415
	25 nodos, 40 de profundidade	Tree Ensemble	0.695	4.956	7.2E7	8.500	0.676	4.998	7.6E7	8.754
	25 nodos, 40 de profundidade	Gradient Boosted Tree	0.393	5.920	1.41E8	11.915	0.492	5.256	1.20E8	10.963
Mediana	//	Linear Regression	0.505	6.684	1.18E8	10.898	0.517	6.637	1.12E8	10.608
	Surrogate, 40 de profundidade	Simple Regression Tree	0.486	4.841	1.17E8	10.849	0.504	4.792	1.15E8	10.744
	25 nodos, 40 de profundidade	Random Forest	0.695	4.815	7.16E7	8.466	0.707	4.812	6.82E7	8.263
	Fator maximo 5	Polynomial Regression	0.353	8.052	1.50E8	12.262	0.354	8.050	1.50E8	12.269
	25 nodos, 40 de profundidade	Tree Ensemble	0.686	4.869	7.53E8	8.681	0.702	4.825	6.94E8	8.331
	25 nodos, 40 de profundidade	Gradient Boosted Tree	0.527	4.860	1.10E8	10.518	0.586	4.307	9.65E7	9.824
Média	//	Linear Regression	0.512	6.616	1.13E8	10.647	0.518	6.624	1.11E8	10.567
	Surrogate, 40 de profundidade	Simple Regression Tree	0.423	5.218	1.29E8	11.395	0.467	5.186	1.23E8	11.112
	25 nodos, 40 de profundidade	Random Forest	0.700	4.831	6.92E7	8.320	0.694	4.883	7.08E7	8.414
	Fator maximo 5	Polynomial Regression	0.351	8.035	1.53E8	12.377	0.357	8.028	1.49E8	12.209
	25 nodos, 40 de profundidade	Tree Ensemble	0.667	4.908	8.19E7	9.053	0.690	4.920	7.18E7	8.476
	25 nodos, 40 de profundidade	Gradient Boosted Tree	0.483	5.295	1.23E8	11.099	0.553	4.693	1.03E8	10.170
Interpolação linear	//	Linear Regression	0.518	6.627	1.06E8	10.315	0.501	6.788	1.16E8	10.806
	Surrogate, 40 de profundidade	Simple Regression Tree	0.407	5.657	1.43E8	11.966	0.418	5.649	1.36E8	11.680
	25 nodos, 40 de profundidade	Random Forest	0.681	5.129	7.30E7	8.548	0.680	5.073	7.50E7	8.662
	Fator maximo 5	Polynomial Regression	0.359	8.027	1.48E8	12.166	0.354	8.053	1.51E8	12.304
	25 nodos, 40 de profundidade	Tree Ensemble	0.686	5.047	7.64E7	8.745	0.678	5.082	7.54E7	8.688
	25 nodos, 40 de profundidade	Gradient Boosted Tree	0.470	5.673	1.25E8	11.198	0.526	5.090	1.11E8	10.539
Interpolação média	//	Linear Regression	0.531	6.582	1.06E8	10.308	0.517	6.740	1.09E8	10.459
	Surrogate, 40 de profundidade	Simple Regression Tree	0.418	5.687	1.33E8	11.552	0.423	5.605	1.30E8	11.427
	25 nodos, 40 de profundidade	Random Forest	0.692	4.972	6.77E7	8.229	0.690	5.001	7.02E7	8.378
	Fator maximo 5	Polynomial Regression	0.365	7.996	1.44E8	12.040	0.366	7.993	1.43E8	11.979
	25 nodos, 40 de profundidade	Tree Ensemble	0.687	4.953	7.18E7	8.475	0.695	4.991	6.90E7	8.311
	25 nodos, 40 de profundidade	Gradient Boosted Tree	0.538	5.063	1.01E8	10.061	0.530	5.070	1.06E8	10.313

Tabela 7 Resultados dos modelos de regressão

Tendo em conta o nosso objetivo de tentar prever os preços de venda dos anúncios dos veículos, achamos que os resultados são positivos. Isto deve ao facto de alguns dos nossos modelos conseguirem valores perto e até inferiores a 5000 para a componente do erro médio absoluto. Para este caso concreto, este parâmetro é o que achamos mais significativo. Esta apreciação é fruto de os anúncios serem colocados por particulares que podem especular o preço dos seus veículos. Anúncios deste tipo levam a que fatores como o MSE e o RMSE a serem muito altos. Sobre estes modelos, os algoritmos de **Random Forest** e **Tree Ensemble** foram novamente os melhores em termos de métricas, contudo requerem algum esforço computacional que outros algoritmos não precisam. Dentro do espaço dos recursos, o algoritmo **Gradient Boosted Tree** foi sem dúvida o que mais recursos gastou. Por este motivo, fomos forçados a reduzir o volume de dados para 25% do conjunto total. Com outros equipamentos, achamos que este último algoritmo poderia ser uma escolha muito mais significativa do que os valores que conseguimos levantar para o relatório.

4.5.2 Classificação

Avançando para a modelação com algoritmos de classificação, lembrando as diferentes estratégias de *binning*, obtemos os seguintes resultados:

Valores em falta	Hiperparametro	Algoritmos/Nodos	Igual Largura				Igual Altura			
			Hold-out validation		Cross validation		Hold-out validation		Cross validation	
			Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem valores em falta	25 trees, 40 de profundidade, Gini Index, No Pruning	Decision Tree	0.516	0.438	0.517	0.439	0.391	0.354	0.397	0.357
	Stochastic average gradient	Logistic Regression	0.362	0.255	0.362	0.254	0.229	0.177	0.230	0.178
	25 trees, 40 de profundidade, Info Gain	Random Forest	0.663	0.611	0.662	0.610	0.530	0.500	0.530	0.499
	Default probability = $\alpha = 0.0001$	Naive Bayes Learner	0.336	0.228	0.336	0.228	0.202	0.149	0.201	0.148
	25 trees, 40 de profundidade, Info Gain	Tree Ensemble	0.653	0.599	0.661	0.609	0.528	0.497	0.529	0.498
	25 trees, 40 de profundidade, Surrogate	Gradient Boosted Tree	Falha computacional				Falha computacional			
Mediana	25 trees, 40 de profundidade, Gini Index, No Pruning	Decision Tree	0.523	0.451	0.528	0.457	0.394	0.354	0.397	0.357
	Stochastic average gradient	Logistic Regression	0.361	0.253	0.362	0.254	0.234	0.183	0.229	0.178
	25 trees, 40 de profundidade, Info Gain	Random Forest	0.663	0.611	0.661	0.608	0.528	0.497	0.527	0.496
	Default probability = $\alpha = 0.0001$	Naive Bayes Learner	0.330	0.220	0.335	0.225	0.202	0.149	0.201	0.148
	25 trees, 40 de profundidade, Info Gain	Tree Ensemble	0.660	0.607	0.662	0.609	0.526	0.495	0.529	0.498
	25 trees, 40 de profundidade, Surrogate	Gradient Boosted Tree	Falha computacional				Falha computacional			
Média	25 trees, 40 de profundidade, Gini Index, No Pruning	Decision Tree	0.530	0.458	0.528	0.456	0.398	0.355	0.399	0.359
	Stochastic average gradient	Logistic Regression	0.366	0.259	0.362	0.254	0.228	0.177	0.228	0.177
	25 trees, 40 de profundidade, Info Gain	Random Forest	0.663	0.611	0.663	0.610	0.528	0.497	0.529	0.498
	Default probability = $\alpha = 0.0001$	Naive Bayes Learner	0.333	0.222	0.334	0.225	0.204	0.152	0.201	0.148
	25 trees, 40 de profundidade, Info Gain	Tree Ensemble	0.659	0.607	0.663	0.610	0.529	0.498	0.529	0.499
	25 trees, 40 de profundidade, Surrogate	Gradient Boosted Tree	Falha computacional				Falha computacional			
Interpolação linear	25 trees, 40 de profundidade, Gini Index, No Pruning	Decision Tree	0.534	0.463	0.528	0.457	0.398	0.358	0.398	0.359
	Stochastic average gradient	Logistic Regression	0.363	0.255	0.361	0.254	0.229	0.178	0.229	0.178
	25 trees, 40 de profundidade, Info Gain	Random Forest	0.664	0.611	0.662	0.609	0.530	0.499	0.528	0.497
	Default probability = $\alpha = 0.0001$	Naive Bayes Learner	0.335	0.225	0.334	0.225	0.204	0.151	0.202	0.150
	25 trees, 40 de profundidade, Info Gain	Tree Ensemble	0.666	0.614	0.662	0.609	0.532	0.502	0.529	0.498
	25 trees, 40 de profundidade, Surrogate	Gradient Boosted Tree	Falha computacional				Falha computacional			
Interpolação média	25 trees, 40 de profundidade, Gini Index, No Pruning	Decision Tree	0.536	0.465	0.531	0.459	0.398	0.358	0.398	0.358
	Stochastic average gradient	Logistic Regression	0.358	0.249	0.363	0.255	0.227	0.175	0.229	0.177
	25 trees, 40 de profundidade, Info Gain	Random Forest	0.660	0.607	0.663	0.610	0.532	0.501	0.530	0.500
	Default probability = $\alpha = 0.0001$	Naive Bayes Learner	0.335	0.225	0.334	0.225	0.202	0.149	0.201	0.148
	25 trees, 40 de profundidade, Info Gain	Tree Ensemble	0.656	0.602	0.663	0.611	0.533	0.502	0.530	0.499
	25 trees, 40 de profundidade, Surrogate	Gradient Boosted Tree	Falha computacional				Falha computacional			

Tabela 8 Resultados dos modelos de classificação

O primeiro aspeto que sobressai dos dados da tabela é a clara desvantagem no uso de discretização orientada a frequência. Este processo foi feito automaticamente pelo *knime*, de forma muito cómoda, mas os resultados foram piores do que o outro formato de discretização. Além disso, vemos também que o tratamento de valores em falta não foi um fator tão marcante para a *accuracy* do modelo como em casos anteriores. A diferença entre validação cruzada e *hold out* também não foi muito notável. Novamente, os algoritmos de **Random Forest** e **Tree Ensemble** demonstraram ser as melhores opções, uma tendência neste relatório.

Comparativamente aos resultados anteriores, achamos que tratar o problema com modelos de classificação não é uma escolha indicada por diversos fatores. Começando pela forma como a previsão é feita. No caso de modelos de regressão, é atribuído um valor contínuo que pode ou não estar perto do valor real. No caso dos modelos de regressão, cada entrada é colocada num intervalo predefinido, o que é bastante limitante dada a variedade de preços de veículos. Além disso, métricas como o MAE são mais fáceis de interpretar neste contexto do que *accuracy*, pois são bastante mais informativas do quão bom ou mau o modelo treinado é.

4.5.3 Clustering

Nos modelos de *Clustering*, alcançamos os seguintes resultados:

Valores em falta	Valores	Parametros	Algoritmos/Nodos	Métrica face ao grupo total		Avaliação	
				Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem valores em falta	Sem alteração, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Means	0.376	0.220	0.376	0.220
	Redução para 10%, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Medoids	0.392	0.240	---	---
Mediana	Sem alteração, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Means	0.336	0.171	0.374	0.218
	Redução para 10%, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Medoids	0.376	0.220	---	---
Média	Sem alteração, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Means	0.349	0.187	0.377	0.222
	Redução para 10%, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Medoids	0.371	0.214	---	---
Interpolação linear	Sem alteração, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Means	0.349	0.187	0.334	0.169
	Redução para 10%, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Medoids	0.325	0.145	---	---
Interpolação média	Sem alteração, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Means	0.335	0.169	0.322	0.153
	Redução para 10%, frequência igual	5 clusters, 10 000 iterações, igual frequência	K-Medoids	0.325	0.157	---	---

Tabela 9 Resultados dos modelos de Clustering

Primeiramente, não achamos interessante experimentar com *binning* de igual largura, pois os dados ficavam bastante desequilibrados e não faria sentido experimentar com o *dataset* resultante do equilíbrio das cargas, pois era muito reduzido. Mesmo assim, para frequência igual, obtemos resultados que vale a pena explorar. Como esperado, não supera técnicas de aprendizagem supervisionada, mas existem modelos que ficaram perto de uma *accuracy* de 0.4, o que quer dizer que os *clusters* não representam bem os alvos. Em termos de performance, o algoritmo *k-means* treina muito mais rápido que o algoritmo *k-medoids* que, com uma porção ainda mais reduzida, ficava com tempos de treino de 30 a 60 minutos. Além disso, não sentimos que o tratamento de valores em falta foi muito significativo para estes modelos.

4.5.4 Redes Neurais

Já para as Redes Neurais, obtivemos os seguintes resultados:

Valores em falta	Hiperparâmetro	Algoritmos/Nodos	Hold-out validation				Cross validation			
			R^2	MAE	MSE	RMSE	R^2	MAE	MSE	RMSE
Sem valores em falta	500 iterações, 3 camadas com 3 nodos cada	Rprop	0.412	7 144	9.2E7	9 640	0.415	7 117	9.1E7	9 586
Média	500 iterações, 3 camadas com 3 nodos cada	Rprop	0.412	7 144	9.2E7	9 640	0.415	7 117	9.1E7	9 586
Mediana	500 iterações, 3 camadas com 3 nodos cada	Rprop	0.413	7 136	9.2E7	9 629	0.416	7 104	9.1E7	9 573
Interpolação linear	500 iterações, 3 camadas com 3 nodos cada	Rprop	0.412	7 114	9.2E7	9 640	0.415	7 117	9.1E7	9 586
Interpolação média	500 iterações, 3 camadas com 3 nodos cada	Rprop	0.413	7 136	9.2E7	9 629	0.416	7 104	9.1E7	9 573
Sem valores em falta, sem outliers	20 iterações, 100 camadas, 1 epochs, tanh	DL4J Feedforward	0.153	9 845	1.3E8	11 600	0.011	10 778	1.5E8	12 623
Sem valores em falta, sem outliers	20 iterações, 100 camadas, 1 epochs, relu	DL4J Feedforward	-1.92	18 947	4.7E8	21 390	-0.017	10 631	1.6E8	12 662
Sem valores em falta, sem outliers	20 iterações, 100 camadas, 1 epochs, sigmoid	DL4J Feedforward	-1.862	18 737	4.4E8	21 142	-1.153	16 175	3.3E8	18 421
Sem valores em falta, sem outliers	20 iterações, 100 camadas, 1 epochs, leakyrelu	DL4J Feedforward	-1.910	18 903	4.5E8	21 351	-1.190	16 042	3.4E8	18 275

Tabela 10 Resultados das redes neuronais

Os resultados das redes neurais não atingiram as expectativas estabelecidas. A análise detalhada dos resultados revelou que o desempenho dos modelos foi abaixo do esperado. Este desempenho insatisfatório foi observado em vários aspetos-chave do modelo, revelando não serem as melhores técnicas de previsão para este problema. Para as redes DL4J, achamos que os resultados são significativos, na medida em que conseguimos observar a diferença que os parâmetros fazem nos resultados, mais concretamente, a função ativação. Antes de correr, esperávamos que a função *Relu* fosse obter os melhores resultados, pois esta função apenas devolve valores entre 0 e 1, precisamente o nosso domínio.

4.5.5 Modelação sem outliers

Finalmente, decidimos ver como é que a falta de *outliers* afetaria alguns dos modelos que construímos anteriormente.

Valores em falta	Hiperparâmetro	Algoritmos/Nodos	Hold-out validation				Cross validation			
			R^2	MAE	MSE	RMSE	R^2	MAE	MSE	RMSE
Sem valores em falta	//	Linear Regression	0.642	5 330	5.6E7	7 485	0.632	5 418	5.7E7	7 594
	Surrogate, 40 de profundidade	Simple Regression Tree	0.509	4 818	7.6E7	8 727	0.522	4 765	7.4E7	8 655
	25 nodos, 40 profundidade	Random Forest	0.745	4 240	3.9E7	6 314	0.744	4 236	4.0E7	6 329
	Fator maximo 5	Polynomial Regression	0.408	7 099	9.1E7	9 564	0.408	7 151	9.2E7	9 634
	25 nodos, 40 profundidade	Tree Ensemble	0.743	4 242	4.0E7	6 375	0.744	4 244	4.0E7	6 333
Média	//	Linear Regression	0.588	5 595	6.3E7	7 944	0.583	5 700	6.3E7	7 996
	Surrogate, 40 de profundidade	Simple Regression Tree	0.514	4 805	7.4E7	8 607	0.509	4 799	7.5E7	8 682
	25 nodos, 40 profundidade	Random Forest	0.715	4 475	4.3E7	6 626	0.714	4 481	4.3E7	6 625
	Fator maximo 5	Polynomial Regression	0.395	7 212	9.3E7	9 654	0.392	7 216	9.6E7	9 655
	25 nodos, 40 profundidade	Tree Ensemble	0.714	4 481	4.3E7	6 622	0.713	4 486	4.4E7	6 636

Tabela 11 Modelos de regressão sem outliers

Valores em falta	Hiperparâmetro	Algoritmos/Nodos	Igual Largura		Igual Largura	
			Accuracy	Cohen's kappa(k)	Accuracy	Cohen's kappa(k)
Sem valores em falta	25 trees, 40 de profundidade, Gini Index, No Pruning	Decision Tree	0.544	0.474	0.542	0.471
	Stochastic average gradient	Logistic Regression	0.408	0.306	0.406	0.306
	25 trees, 40 de profundidade, Info Gain	Random Forest	0.676	0.624	0.675	0.624
	Default probability = $\alpha = 0.0001$	Naive Bayes Learner	0.372	0.264	0.369	0.264
	25 trees, 40 de profundidade, Info Gain	Tree Ensemble	0.677	0.626	0.674	0.623
Média	25 trees, 40 de profundidade, Gini Index, No Pruning	Decision Tree	0.476	0.384	0.470	0.383
	Stochastic average gradient	Logistic Regression	0.404	0.304	0.406	0.306
	25 trees, 40 de profundidade, Info Gain	Random Forest	0.672	0.620	0.674	0.622
	Default probability = $\alpha = 0.0001$	Naive Bayes Learner	0.367	0.261	0.370	0.264
	25 trees, 40 de profundidade, Info Gain	Tree Ensemble	0.676	0.620	0.674	0.622

Tabela 12 Modelos de classificação sem outliers

À primeira vista, podemos logo afirmar que os resultados melhoraram face aos modelos que foram treinados com os *outliers*. No caso dos modelos de regressão, esta técnica melhorou todas as métricas como o MAE, mas mais importante ainda, o RMSE e o MSE baixaram muito face aos outros modelo. No caso da métrica MSE, passou quase para metade, mesmo continuando a ser um valor elevado. Nas suas contrapartes de classificação, também houve melhoria, apesar de não ser tão notável. Este último caso está relacionado à forma como as métricas para algoritmos de classificação funciona, onde o que realmente importa é se a classe alvo foi a escolhida, não havendo uma margem de erro.

5 Conclusão

Do desenvolvimento do presente trabalho prático, podemos certamente afirmar que o grupo ficou consciencializado para as dificuldades que é conceber modelos de aprendizagem. Muito antes de decidir sequer qual o modelo a utilizar, a exploração e preparação de dados são duas tarefas extensas e importantíssimas para obter bons resultados na modelação. Apercebemos nos que a própria modelação não é uma ciência exata e requer bastante paciência e diversas tentativas para alcançar os melhores resultados possíveis. O desenvolvimento, ao contrário do que achávamos, não foi em cascata, exigindo diversas vezes voltar à preparação e exploração dos dados, sempre refletindo à cerca de que o poderia levar a melhores resultados.

Refletindo à cerca do que foi feito para cada conjunto de dados, sentimos nos realizados com ambos, mas, por constrangimentos temporais e de *hardware*, achamos que certas situações poderiam ter sido mais bem exploradas. Esta questão pesou na modelação do *dataset* do grupo, pois possuía um enorme conjunto de entradas que não estávamos a conseguir modelar. Mesmo assim, achamos que estes obstáculos não diminuíram o entendimento que o grupo retirou da experiência, mesmo que os resultados pudessem ter sido melhores.