

Universidade do Minho
Escola de Engenharia

Aprendizagem e Decisão Inteligentes

Licenciatura em Engenharia Informática

Ano Letivo de 2023/2024

Relatório de Desenvolvimento

Grupo 33

Diogo Gabriel Lopes Miranda (a100839)

João Ricardo Ribeiro Rodrigues (a100598)

Sandra Fabiana Pires Cerqueira (a100681)

Tiago Miguel Marques Pereira (a96429)

Março, 2024



Índice

1 Introdução	3
2 Tarefa 1: Classificação de quadros clínicos de Hepatite	3
2.1 Estudo do Negócio	3
2.2 Estudo dos dados.....	4
2.2.1 Category	5
2.2.2 Age	5
2.2.3 Sex	6
2.2.4 ALB.....	6
2.2.5 ALP.....	6
2.2.6 ALT	7
2.2.7 AST.....	7
2.2.8 BIL.....	7
2.2.9 CHE.....	8
2.2.10 CHOL	8
2.2.11 CREA	9
2.2.12 GGT	9
2.2.13 PROT	9
Análise da correlação	10
2.3 Preparação de dados.....	10
Tratamento de missing values e valores errados.....	11
Análise de outliers	12
2.4 Modelação.....	12
2.4.1 Modelação com dados normalizados	12
Resultados obtidos com tratamento de dados normalizados.....	13
2.5 Avaliação e Comparação dos diferentes modelos.....	15
3 Tarefa 2.....	16
3.1 Estudo do negócio.....	16
3.2 Estudo dos dados	16
3.2.1 Preço.....	17
3.2.2 Características do imóvel	17
3.2.3 Localização	21
3.2.4 Condições de venda	24
3.3 Preparação dos dados.....	25
3.4 Modelação.....	26
3.5 Avaliação.....	31

4. Conclusão	32
5. ANEXOS.....	32

1. Introdução

Este trabalho foi desenvolvido no âmbito da unidade curricular de Aprendizagem e Decisão Inteligentes, na qual nos foi proposta a conceção de modelos de aprendizagem e decisão. O trabalho é constituído por duas partes.

A primeira consiste na exploração, análise e preparação de um *dataset* fornecido pelos professores, procurando extraír conhecimento relevante no contexto do problema em questão com o objetivo de conceber modelos de *Machine Learning* de classificação e realizar uma análise crítica dos resultados obtidos. A segunda por sua vez também consiste, na consulta, análise, exploração e preparação de um *dataset*, mas desta vez escolhido pelos elementos do grupo, com o objetivo de conceber e otimizar diversos modelos de *Machine Learning* de regressão e realizar uma análise critica dos resultados obtidos.

2. Tarefa 1: Classificação de quadros clínicos de Hepatite

Para a tarefa 1, foi-nos fornecido um *dataset* com informação acerca dos resultados de análises clínicas a amostras de sangue.

A metodologia que será utilizada na resolução do problema será o CRISP -DM. Este modelo de processos define um guião constituído por 6 etapas, sendo elas:

- Estudo do negócio;
- Estudo dos dados;
- Preparação dos dados;
- Modelação;
- Avaliação;
- Desenvolvimento;

De seguida, iremos então abordar cada uma destas etapas e o processo feito com as mesmas para o nosso projeto.

2.1. Estudo do Negócio

O nosso **problema é de classificação**, pois, o nosso objetivo é desenvolver um modelo que preveja qual a categoria à qual um indivíduo pertence com base em determinados dados ou características. O objetivo é diagnosticar a progressão da Hepatite C em diferentes

estágios. Para isto, foi-nos fornecido um *dataset* com informação acerca dos resultados de análises clínicas a amostras de sangue de pessoas. Com estes dados, e com o auxílio da ferramenta de análise de dados e *machine learning KNIME* iremos criar modelos de aprendizagem com o objetivo de solucionar o problema.

Pretendemos que os nossos modelos sejam capazes de analisar os dados e chegar à conclusão certa acerca do quadro clínico da pessoa em questão, sendo que os resultados podem ser:

- **Blood Donors:** Indivíduos que doaram sangue e não possuem Hepatite C.
- **Hepatitis C:** Indivíduos com Hepatite C, mas sem progressão para Fibrose ou Cirrose.
- **Fibrosis :**Indivíduos com Hepatite C que progrediram para o estágio de Fibrose. ○ **Cirrhosis:** Indivíduos com Hepatite C que progrediram para o estágio de Cirrose.
- **Suspect Blood Donor:** Indivíduos que não tem quadro clínico de nenhum dos estágios da Hepatite mas que não possuem valores considerados normais.

Os objetivos que pretendemos alcançar são então:

- Analisar o *dataset* na sua totalidade;
- Tratar de inconsistências no *dataset* e extrair conhecimento extra dos dados,
- Criar diversas formas de visualização de dados (ex: gráficos etc);
- Criar modelos de aprendizagem usando diferentes técnicas.

2.2. Estudo dos dados

O *dataset* é constituído por 615 linhas e 18 atributos, sendo eles:

1. **Id:** id do registo
2. **Age:** idade
3. **Birth year:** ano de nascimento;
4. **Birth month:** mês de nascimento;
5. **Birth day:** dia de nascimento;
6. **Sex:** sexo;
7. **Birth location:** local de nascimento;
8. **Albumin (ALB):**Albumina;
9. **Alkaline phosphatase(ALP):** Fosfatase Alcalina;
10. **Alanine transferase(ALT):** Alanina Aminotranferase;
11. **Aspartate transferase(AST):** Aspartato Aminotransferase;
12. **Bilirubin(BIL):** bilirrubina;
13. **Cholinesterase(CHE):** Colinesterase;
14. **Cholesterol(CHOL):** Colesterol;

15. **Creatinine(CREA)**: Creatinina;
16. **Gama Glutamil Transferase(GGT)**: Gama-Glutamil Transpeptidase;
17. **Protein(PROT)**: Proteina;
18. **Category (target variable)**: Categoria.

A Categoria é o que vamos tentar prever com os modelos que iremos desenvolver ao longo deste projeto. De seguida, vamos descrever a análise feita a cada um dos atributos e tirar algumas conclusões.

2.2.1. Category

Esta é a variável objetivo e pode ser: **Blood Donor**, **Suspect Blood Donor**, **Hepatitis**, **Fibrosis**, e **Cirrhosis**. Como podemos ver na imagem, a categoria **Blood Donor** é a mais representada, constituindo cerca 88,83% dos casos, evidenciando um **desequilíbrio muito significativo nos dados**.

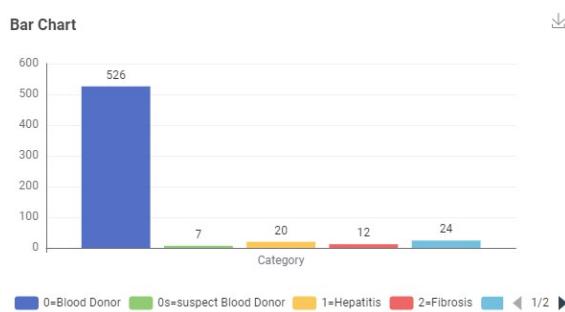
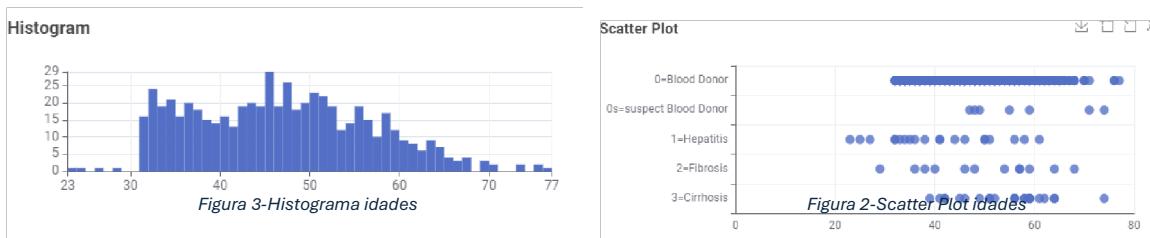


Figura 1-BarChart Categories

2.2.2. Age

As idades do dataset variam entre 19 e 77 anos, com uma média de 47.408 anos. A distribuição de idades, como se vê no histograma, mostra que a maioria das pessoas está entre 32 e 60 anos. Os dados do **scatter plot** indicam que indivíduos com fibrose geralmente são mais velhos do que aqueles apenas com hepatite. Por outro lado, o grupo com cirrose, embora tenha uma idade máxima semelhante à do grupo com fibrose,



começa com uma idade inicial mais elevada que os outros dois grupos. Este padrão ocorre porque a cirrose, a fibrose e a hepatite representam estágios progressivos da doença, levando a que os estágios mais avançados se manifestem mais tarde na vida.

2.2.3. Sex

No dataset existem 363 homens e 226 mulheres, ou seja, cerca de 1.5 homens para cada mulher. Na seguinte imagem conseguimos ver a distribuição de homens e mulheres em cada categoria e podemos reparar que a proporção entre homens e mulheres nos grupos da hepatite, fibrose e cirrose são diferentes da proporção da população (aproximadamente 4, 2 e 2 homens para cada mulher, respetivamente) o que pode indicar uma tendência do

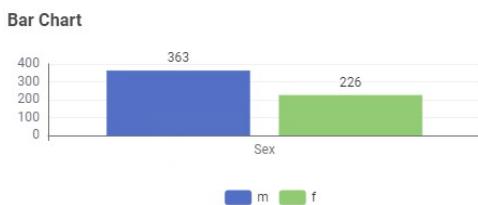


Figura 5-Bar Chart sex count

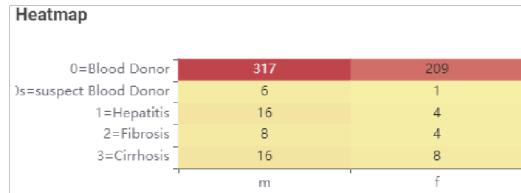


Figura 4- Heatmap sex

sexo masculino no desenvolvimento destes problemas se a quantidade de dados for significativa.

2.2.4. ALB

Os gráficos que se seguem, referem-se à Albumina. Após uma análise do gráfico de barras podemos concluir que os valores médios que o valor de ALB no sangue de pacientes com Hepatite e Fibrose se assemelham bastante à concentração de ALB de um paciente normal, não sendo um bom indicativo para poder detetar uma condição desse tipo. Por outro lado, valores mais baixos podem ser um indicativo de Cirrose ou que a amostra de sangue é suspeita.

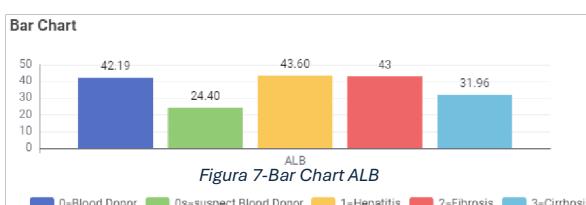


Figura 7-Bar Chart ALB



Figura 6-Box plot alb por categoria

2.2.5. ALP

Os seguintes gráficos, referem-se à concentração média de ALP no sangue dos pacientes. Após uma análise dos mesmos, podemos concluir que valores baixos de ALP no sangue, são um possível indicativo de que o paciente pode possuir Hepatite ou Fibrose, por outro lado, valores mais altos, indicam que o paciente possa ter Cirrose. Uma concentração muito elevada de ALP no sanguineo é um indicativo de que o paciente possui uma outra condição clínica, classificando-o como suspeito.

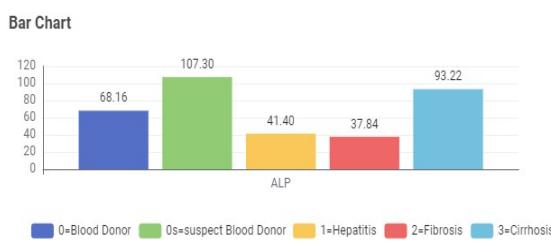


Figura 8- Box Plot ALP

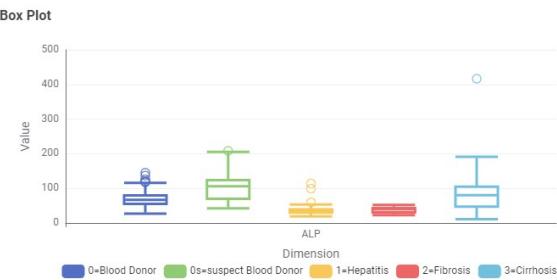


Figura 9- Box Plot ALP

2.2.6. ALT

Os gráficos que se seguem, são relativos à concentração média de *ALT* nas amostras. Através de uma análise aos mesmos, podemos concluir que: a média de *ALT* é menor em pacientes com Hepatite ou Fibrose do que em saudáveis; a queda de *ALT* é mais acentuada em casos de Cirrose e que altos níveis de *ALT* estão associados a amostras de sangue suspeitas, indicando possivelmente outra condição.

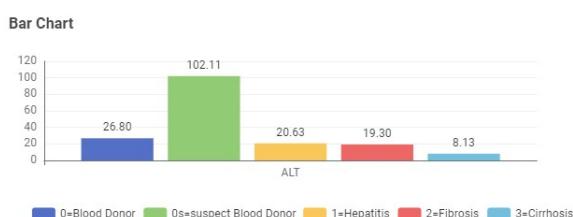


Figura 10- Bar Chart ALT

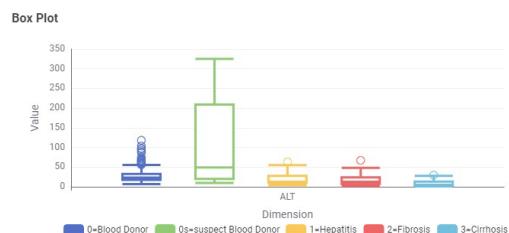


Figura 11- Box Plot ALT

2.2.7. AST

Os gráficos que se seguem, dizem respeito aos níveis médios de concentração de *AST* no sangue, das amostras de dados. Observando os resultados, podemos notar uma tendência clara no aumento nos valores de *AST* à medida que avançamos para diagnósticos mais graves (*Fibrosis* e *Cirrosis*).

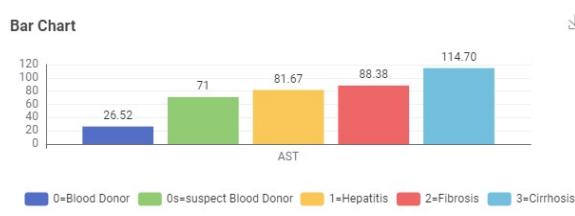


Figura 12-Bar Chart AST

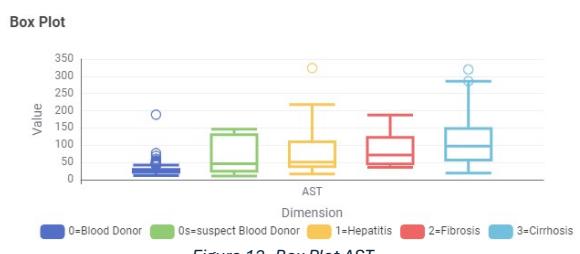


Figura 13- Box Plot AST

2.2.8. BIL

Os gráficos que se seguem, dizem respeito aos níveis médios de concentração de *BIL* no sangue das amostras dos dados. Analisando-os verificamos uma variação significativa nos níveis médios de *BIL* entre os diagnósticos. Doadores de sangue saudáveis e suspeitos têm

as médias mais baixas, enquanto pacientes com Cirrose têm uma média muito mais alta. Isto sugere uma forte associação entre os níveis de BIL e a gravidade das condições hepáticas, especialmente a Cirrose.

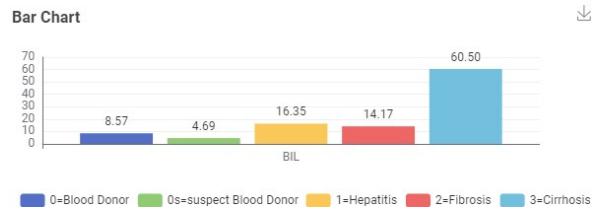


Figura 15-Bar Chart BIL

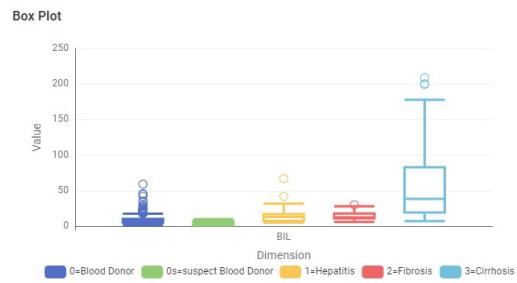


Figura 14-Box Plot BIL

2.2.9. CHE

Os gráficos que se seguem, dizem respeito aos níveis médios de concentração de CHE no sangue das amostras de dados. Analisando-os conseguimos ver que os níveis médios de CHE variam ligeiramente entre os diagnósticos. Pacientes com Hepatite têm a média mais

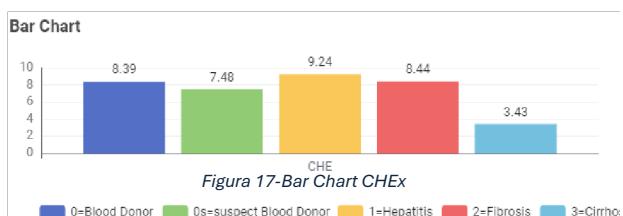


Figura 17-Bar Chart CHE



Figura 16-Box Plot CHE

alta, seguidos por doadores de sangue saudáveis e aqueles com Fibrose. Pacientes com Cirrose têm uma média muito mais baixa, sugerindo uma possível associação entre os níveis de Colinesterase e o diagnóstico de Cirrose.

2.2.10. CHOL

Os gráficos que se seguem, dizem respeito aos níveis médios de concentração de CHOL no sangue das amostras de dados. Analisando os resultados, podemos verificar que o valor médio de CHOL varia ligeiramente entre os diferentes diagnósticos, destacando-se, no entanto, o diagnóstico de Cirrose, no qual o valor é o mais baixo e com maior diferença para os restantes.

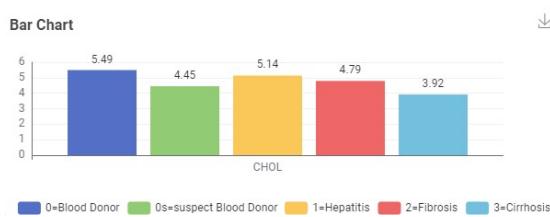


Figura 19-Bar Chart CHOL



Figura 18-Box Plot CHOL

2.2.11. CREA

Os gráficos que se seguem, dizem respeito aos níveis médios de concentração de *Creatinine* nas amostras de dados. Analisando os resultados, podemos verificar os valores médios de CREA para Hepatite e Fibrose são mais baixos do que para os Blood Donors, sugerindo uma relação com o diagnóstico. No entanto, para a Cirrose, o valor é significativamente mais elevado, sugerindo uma relação entre altos níveis de CREA e a Cirrose.

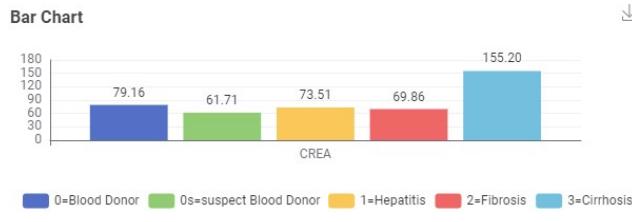


Figura 21-Bar Chart CREA

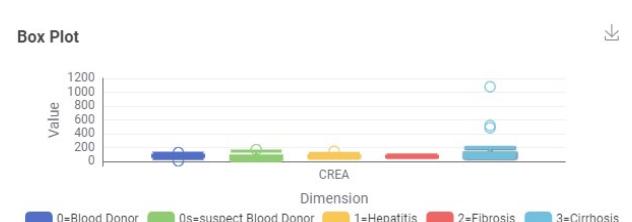


Figura 20-Box Plot CREA

2.2.12. GGT

Os gráficos mostram que o valor médio de GGT varia entre os diagnósticos. Um valor baixo pode indicar um Blood Donor, um valor alto um Suspect Blood Donor e valores entre os 68 e os 136 os restantes casos. Indicando que valores acima do normal, definem casos de doença.

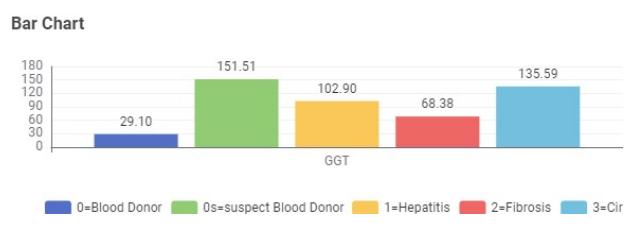


Figura 22-Bar Chart GGT

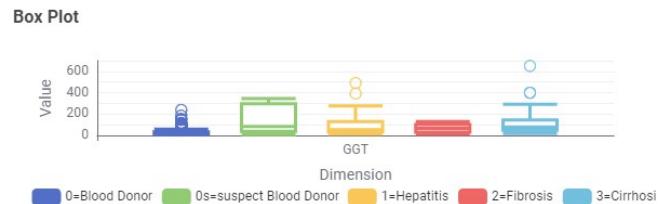


Figura 23-Box Plot GGT

2.2.13. PROT

Os gráficos indicam que o valor médio de PROT é minimamente mais alto em casos de Hepatite e Fibrose, enquanto é mais baixo em Cirrose e Blood Donor. Isso sugere uma relação entre os níveis de PROT e estes diagnósticos, especialmente, Suspect Blood Donor.

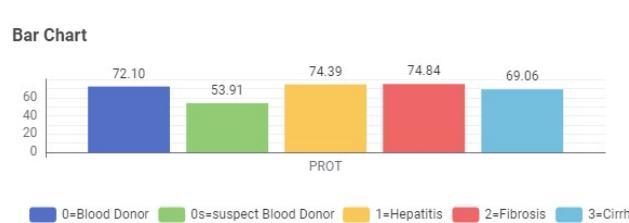


Figura 24- Bar Chart PROT

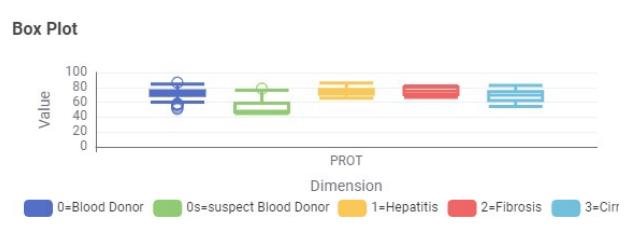


Figura 25-Box Plot PROT

Análise da correlação

De forma a conseguirmos avaliar a correlação entre as várias variáveis utilizamos um nodo de Rank Correlation. Através da matriz que este cria, podemos ver que poucas variáveis apresentam uma correlação significativa entre si.

Os três pares de variáveis com maior correlação são o par ALB/PROT, com um valor de correlação de 0.526, o par Sex/CREA, com um valor de 0.525, e o par ALT/AST, com um valor de 0.5, o que são valores considerados moderados.

Para além disso, é importante ver as variáveis com maior correlação com a variável objetivo, pois estas podem ter um maior impacto na fase de previsão. As duas variáveis com maior correlação com a Category são a AST, 0.425, e a GGT, 0.3510.

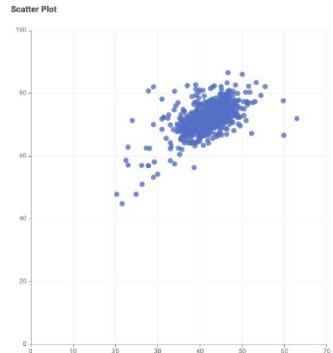
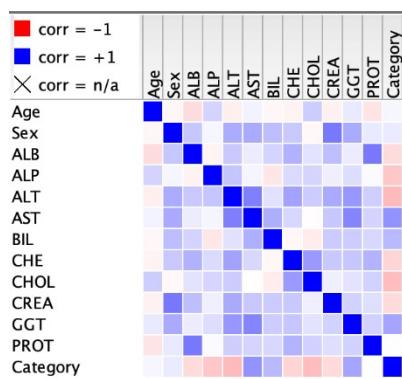


Figura 27 – Scatter Plot ALB e PROT Figura 26-Matriz de correlação

2.3. Preparação de dados

Nesta fase crucial do projeto começamos por analisar os atributos do *dataset* e fazer as alterações que fossem necessárias para que o valor dos atributos estivesse de acordo com o que era esperado.

- A primeira ação que realizámos foi a alteração no csv Reader, onde inserimos os dados. No dataset, o separador decimal é a vírgula “,” e não o ponto “.”. Assim, **modificámos o campo do separador decimal** neste nodo, como se pode ver no [Anexo](#)

, para ter isto em conta. Apercebemo-nos ainda que alguns dos atributos possuíam “NA”, pois devido a isto o tipo da coluna não mudou para number, como seria esperado, problema que iremos explicar como resolvemos mais a frente.

- De seguida, recorremos ao auxílio dos nodos *Rank Correlaltion* e *Scatter Plot Matrix*, como se pode ver pelas seguintes imagens:

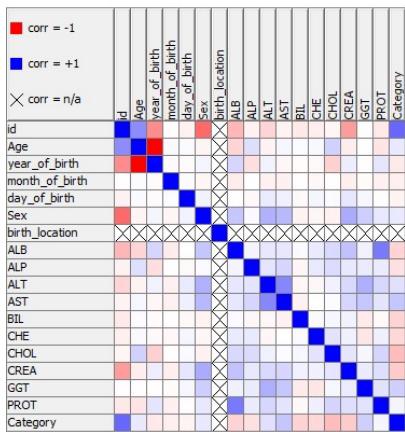


Figura 29-Matriz de Correlação

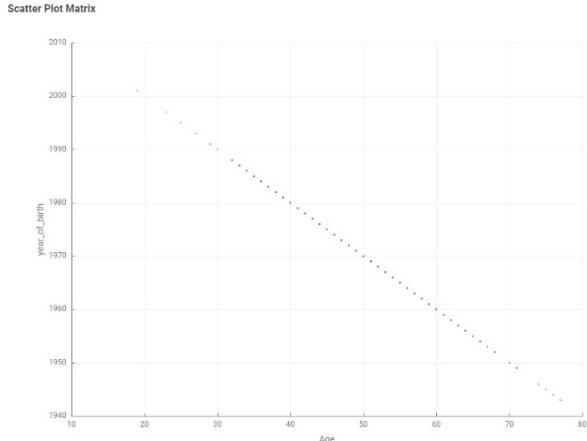


Figura 28-Scatter Plot Matrix

Pelos gráficos, vemos que manter atributos como “*birth_location*” e “*year_of_birth*” no nosso conjunto de dados é inútil. O primeiro só tem correlação com ele mesmo, como se vê na Figura 2, e o segundo é basicamente o mesmo que o atributo “*age*”. Como a Figura 3 mostra, os atributos “*age*” e “*year_of_birth*” formam uma linha contínua, mostrando que a idade e o ano de nascimento são a mesma coisa, e estão em concordância uma vez que as idades estavam relativas a 2020. Por isso, decidimos ficar com a idade, que é mais útil para análise de dados e estatísticas.

- Os atributos '*day_of_birth*' e '*month_of_birth*' também foram **removidos** do conjunto de dados, pois, baseando-se em conhecimento especializado do domínio médico e biométrico, entende-se que a data de nascimento de um indivíduo não tem impacto direto sobre as concentrações de componentes sanguíneos, as quais são determinadas por fatores biológicos e fisiológicos que não estão correlacionados com a idade específica dentro de um ano. Com isso em mente, usamos o nodo **Colum Filter** para tirar todas as colunas que correspondem aos atributos que mencionamos antes.

Tratamento de missing values e valores errados

- Utilizando o nodo **Data Explorer**, apercebemo-nos de que os **valores nominais** do atributo “*sex*” **não estavam todos corretos**. Havia entradas de dados que em vez de terem “m” tinham “mm”, como se pode ver pela segunda [imagem do i.Anexo](#)

- . Para tratar deste problema, utilizamos o nodo *Java Snipped*, para substituir os “*mm*” por “*m*”.

- Posteriormente, com recurso ao *Data Explorer*, detetámos quais os dos atributos, **ALP, CHOL, ALT, PROT e ALB**, possuíam campos preenchidos com “**NA**”, como se pode ver pela 3ª [imagem do i.Anexo](#)

- _ Posto isto, atualizamos o nosso *Java Snippet* para passar os “**NA**” para null, o código modificado do *Java Snippet* encontra-se na 4ª [imagem do i.Anexo](#)

:

- Posto isto, utilizamos o nodo Missing Value, para **lidar** com os ***missing values***. Decidimos que iríamos explorar **três alternativas** diferentes para este tratamento: **remover as linhas; substituir pela média e interpolação polinomial**.

-Depois disto resolvido, fizemos então a **passagem** dos atributos **ALP, CHOL, ALT, PROT e ALB** para **number**, com o nodo String to number.

Análise de outliers

Para assegurar a qualidade das análises de dados, o grupo decidiu analisar os outliers nos atributos. A presença de outliers pode distorcer medidas estatísticas, como a média e o desvio padrão, e afetar a eficácia de modelos estatísticos e algoritmos de machine learning.

Recorremos ao uso do nodo **Box Plot**, do KNIME, que permitiu que visualizássemos a distribuição dos dados e identificássemos rapidamente a presença dos *outliers* como se pode ver na 5^a imagem nos [i.Anexo](#)

Como estes valores podem afetar significativamente os resultados, o grupo optou por explorar **três alternativas** para **tratar outliers: remoção e substituição** por valores próximos, com auxílio do nodo Numeric Outliers, e **não os tratar**. Testaremos diferentes modelos com estes métodos para encontrar o melhor resultado.

2.4. Modelação

2.4.1. Modelação com dados normalizados

Apesar de estarmos a explorar alternativas de tratamento de dados diferentes (*tratar outliers (remover, substituir, ignorar), lidar com missing values (remover linhas, substituir pela média, usar interpolação polinomial)*), para cada uma das estratégias desenvolvemos os seguintes modelos presentes na primeira imagem do [iiAnexo](#).

Como o nosso *dataset* se tratava de um problema de classificação, utilizamos nodos de árvores de decisão com os algoritmos de aprendizagem **Decision Tree, Random Forest e Gradient Boosting**.

Adotámos duas abordagens principais para validar a eficácia destes modelos: **Cross Validation**, para correr os algoritmos de aprendizagem várias vezes, o que permite obter resultados mais fiéis e **Hold-out Validation**.

No nodo **X-Partitioner** foi utilizada a opção **Stratified sampling** na coluna objetivo “Category” com a **random seed 2024**. Por sua vez, no nodo **Partitioning** também foi usada a opção **Stratified Sampling**, com a **random seed 2024** e **30%** dos dados foram deixados para **teste** e **70%** para **treino** dos modelos.

Resultados obtidos com tratamento de dados

De modo a conseguirmos identificar qual o melhor modelo de classificação desenvolvido, fizemos uma tabela, que se encontra no [iiiAnexo](#) que reúne todos os resultados dos modelos desenvolvidos, tendo em conta os diferentes tratamentos de dados efetuados.

Uma vez que a tabela possui muitas entradas e fica confusa de analisar, reunimos os **melhores 10 modelos** desenvolvidos, tendo por base o valor da **accuracy** e do **cohen's Kappa (K)** para os conseguirmos analisar melhor.

Tratamento Missing Values	Tratamento Outliers	Nodo de decisão utilizado	Validação	Feature Selection	Nº de casos testados	Classificados corretamente	Classificados incorretamente	Accuracy	Cohen's Kappa (K)
Remoção	Nenhum Tratamento	Gradient Basted Tree	Hold-out Validation	Sim	118	115	3	97.46%	0.871
Remoção	Nenhum Tratamento	Gradient Basted Tree	Cross Validation	Não	589	563	26	95.59%	0.773
Remoção	Nenhum Tratamento	Random Forest	Cross Validation	Não	589	563	26	95.59%	0.766
Remoção	Próximo valor permitido	Gradient Basted Tree	Hold-out Validation	Sim	118	112	6	94.92%	0.751
Interpolação polinomial	Nenhum Tratamento	Random Forest	Hold-out Validation	Não	185	172	13	92.97%	0.710
Interpolação polinomial	Nenhum Tratamento	Gradient Basted Tree	Hold-out Validation	Sim	123	114	9	92.68%	0.709
Interpolação polinomial	Nenhum Tratamento	Random Forest	Cross Validation	Não	615	570	45	92.68%	0.701
Remoção	Próximo valor permitido	Gradient Basted Tree	Cross Validation	Não	589	553	36	93.89%	0.704
Substituir pela média	Nenhum Tratamento	Random Forest	Cross Validation	Não	615	570	45	92.68%	0.693
Substituir pela média	Nenhum Tratamento	Gradient Basted Tree	Hold-out Validation	Não	185	170	15	91.89%	0.693

Figura 30 - 10 melhores modelos desenvolvidos

Tal como podemos verificar na tabela, o melhor resultado foi obtido para o modelo com Feature Selection, sendo que a validação foi feita com hold out validation, com o nodo gradient basted tree, onde nenhum tratamento dos outliers foi efetuado e as linhas com missing values foram removidas. De seguida iremos analisar detalhadamente esse modelo.

Resultados obtidos com Feature Selection

O nosso melhor modelo, atingiu então, um Cohn's Kappa de 0.871 e uma accuracy de 97.46%. Um Cohn's Kappa de 0.871 indica que há uma excelente concordância entre as classificações previstas pelo modelo e as observações reais. Isto é, o modelo é altamente eficaz em prever corretamente os diferentes quadros clínicos de hepatite. O que também sugere que o modelo é robusto e confiável, minimizando tanto os falsos positivos quanto os falsos negativos.

Como se pode ver pela matriz de confusão, este modelo conseguiu categorizar com sucesso várias amostras sanguíneas do dataset, apesar de não ter categorizado nenhum “Os=suspect Blood Donor”, conseguiu determinar a categoria corretamente das outras amostras. Tendo errado, apenas 3, dos 118 testes que realizou.

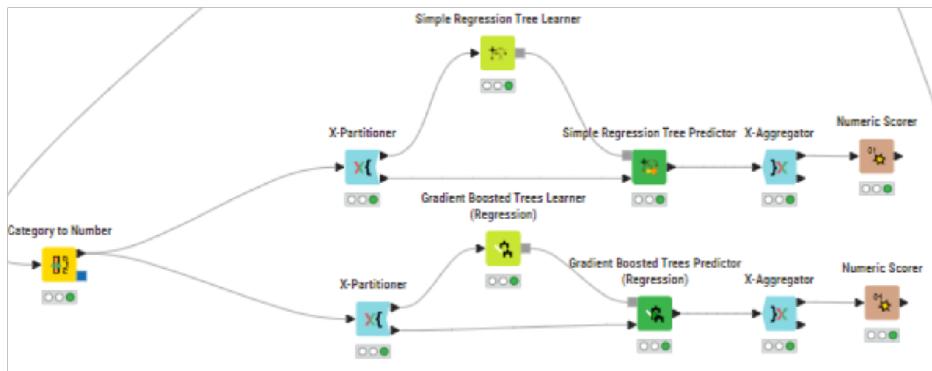


Figura 33- Modelos de regressão

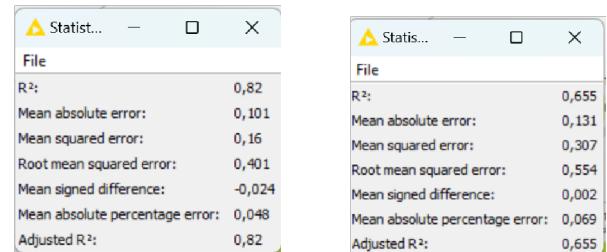


Figura 35-Simple Regression Tree
Learner

Figura 34-Gradient Boosted Trees Learner
(Regression)

Estes

foram os resultados obtidos, nos nossos nodos do modelo de regressão, sendo eles o Simple Regression e o Gradient Boosted Tree (Regression). Tal como podemos verificar pelas figuras acima, o melhor resultado obtido proveio do nodo Gradient Boosted, no qual obtemos um R² igual a 0.82. Isto é, 82% da variabilidade da condição da amostra, pode ser determinada pelos valores das proteínas das nossas análises, presentes no dataset.

2.5. Avaliação e Comparação dos diferentes modelos

Com base na tabela de resultados obtidos e nas análises elaboradas até ao momento para os vários modelos desenvolvidos, com diferentes abordagens de tratamento de dados, podemos concluir que os nodos de decisão *Gradient Booster Tree* e *Random Forest* foram os que alcançaram os valores mais elevados de *Cohen's Kappa* e *accuracy*. O *Gradient Booster* destacou-se, sendo o nodo utilizado no nosso modelo de melhor desempenho.

Outro ponto a salientar foi o tratamento dos *Missing Values*. A forma como lidámos com estes valores variou de modelo para modelo, mas em geral, remover estes valores ou substituí-los pela média dos valores, provaram ser as formas mais eficientes, do que alterar estes valores através da interpolação polinomial.

No que diz respeito ao tratamento dos **outliers**, a **abordagem** que se mostrou **mais eficaz** foi **não fazer** qualquer **tratamento** destes valores. Isto deve-se ao facto de grande parte do nosso conjunto de dados serem amostras sanguíneas de pessoas saudáveis, e as amostras com valores anormais(*outliers*) podem ser relativas a pacientes com alguma patologia

clínica. Desta forma, qualquer tratamento que façamos a estes valores vai diminuir ainda mais a nossa amostra, já reduzida de amostras de pacientes com alguma condição clínica não saudável. Assim sendo, não efetuar qualquer tratamento com os outliers, resulta num melhor resultado do que substituir ou remover as linhas relativas a esses valores.

Assim, podemos concluir que, para construirmos um modelo fiável e atingir o valor máximo de *Cohen's Kappa* e *accuracy*, devemos manter os nossos *outliers* para não eliminar casos de estudo úteis para o nosso modelo, e que melhor forma de tratar dos *Missing Values*, seria removê-los completamente ou substituí-los pela média dos valores. Por fim, concluímos que os nodos que mostraram maior desempenho nos nossos modelos foram o *Gradient Booster Tree* e o *Random Forest*, com altos valores de *Cohen's Kappa* e *accuracy*.

3. Tarefa 2

Como o *dataset* fornecido pela equipa docente era um problema de classificação, o grupo optou por escolher um que *dataset* fosse um problema de regressão.

3.1. Estudo do negócio

O objetivo deste problema é descobrir os atributos que mais contribuem para o preço do imobiliário em Melbourne, Austrália. Desta forma, pretende-se utilizar a ferramenta de ciência de dados KNIME para análise de dados e, posteriormente, para a construção de modelos de *machine learning*.

3.2. Estudo dos dados

Os dados para este problema foram retirados do site Kaggle em [Melbourne Housing](#). O *dataset* contém 22 colunas e 18396 linhas, sendo que cada linha corresponde a uma propriedade e cada coluna a um atributo. Os atributos são os seguintes:

1. **Id:** Identificador único do registo.
2. **Suburb:** Subúrbio do imóvel.
3. **Address:** Endereço de morada.
4. **Rooms:** Número de espaços comuns.
5. **Type:** Tipo de imóvel (h - casa, casa de campo, vila, casa geminada; u - apartamento, duplex; t - moradia)
6. **Price:** Preço do imóvel em dólares.
7. **Method:** Método de venda do imóvel
 - a. S (Sold) - Imóvel vendido,
 - b. SP (Sold prior) - Imóvel vendido antecipadamente,
 - c. PI (Property passed in) - Imóvel não vendido em leilão,

- d. VB (Vendor bid) - Licitação do vendedor,
 - e. SA (Sold after auction) - Imóvel vendido depois do leilão.
8. **SellerG**: Agente do imóvel.
 9. **Date**: Data de venda do imóvel.
 10. **Distance**: Distância ao centro económico da cidade.
 11. **Postcode**: Código postal.
 12. **Bedroom2**: Número de quartos (obtidos de fontes secundárias)
 13. **Bathroom**: Número de casas de banho.
 14. **Car**: Número de lugares de estacionamento.
 15. **Landsize**: Área total do terreno em metros quadrados.
 16. **BuildingArea**: Área de construção em metros quadrados.
 17. **YearBuilt**: Ano de construção do imóvel.
 18. **CouncilArea**: Município.
 19. **Latitude**: Latitude do imóvel.
 20. **Longitude**: Longitude do imóvel.
 21. **RegionName**: Região geral (Norte, Sul, Sudeste, etc.)
 22. **PropertyCount**: Número de propriedades no subúrbio.

3.2.1. Preço

A análise do dataset revela que os imóveis custam entre \$85,000 e \$9,000,000, com uma média de \$1,056,697. A maioria custa menos de \$2,000,000 e o número de propriedades diminui à medida que o preço aumenta. O objetivo é identificar o que faz um imóvel ser mais caro.

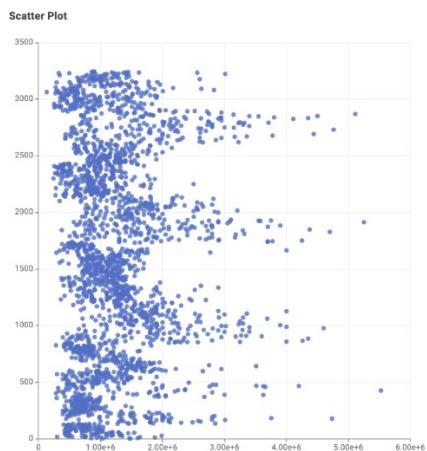


Figura 36 - Scatter Plot

3.2.2. Características do imóvel

Neste dataset, o imóvel é caracterizado pelos seguintes atributos: **Rooms**; **Type**; **Bedroom2**; **Bathroom**; **Car**; **Landsize**; **BuildingArea** e **YearBuilt**.

Rooms: Os espaços comuns variam de 1 a 12, sendo 2 o mais frequente (43%) e mais de 90% dos imóveis têm até 4. Um maior número de espaços comuns geralmente indica um preço mais alto, mas é incerto se essa tendência se mantém para imóveis com muitos espaços comuns, uma vez que são poucos os que os tem.

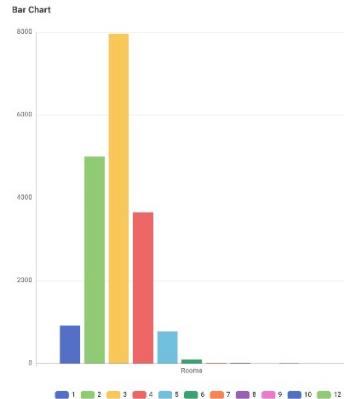


Figura 37-Bar chart Rooms

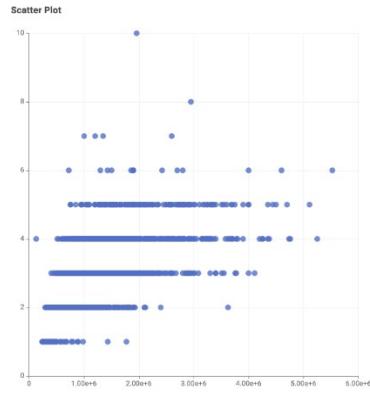


Figura 38-Scatter plot Rooms

Type: O tipo mais comum de imóvel são casas (inclui casas geminadas, casas de campo e vilas), depois moradias e, por fim, apartamentos. O preço médio de uma casa é de

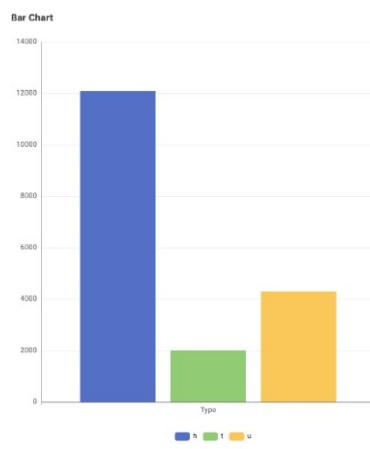
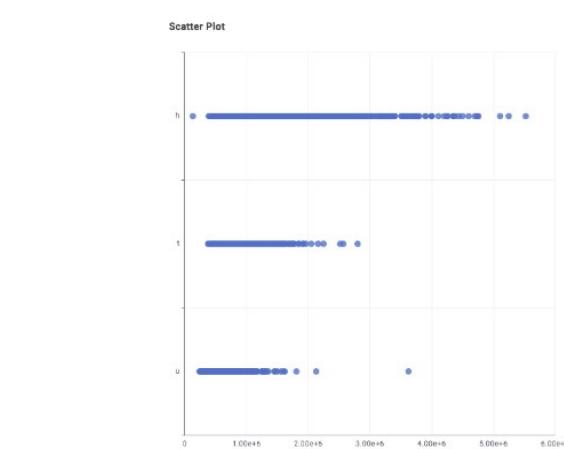


Figura 40-

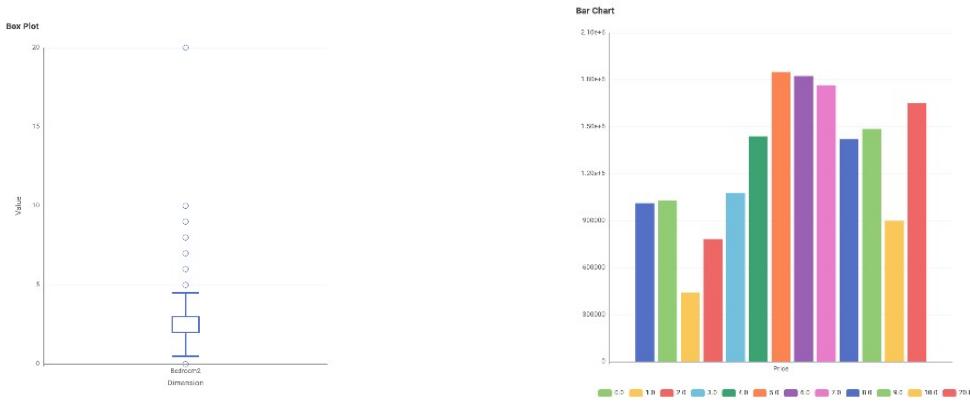


Bar Chart Type

Figura 39-Scatter plot type

\$1,234,250, de um apartamento é de \$924,060 e de uma moradia \$618,715.

Bedroom2: Os imóveis têm entre 0 e 20 quartos, com uma média de 2,9 quartos. Existem 3469 propriedades sem informação sobre quartos. A maioria tem entre 0 e 5 quartos, como se vê no **box plot**, com 75% tendo até 3. O **gráfico de barras** mostra que o preço médio dos



imóveis tende a aumentar para aqueles com 2 a 4 quartos. No entanto, essa tendência não é observada em propriedades com mais quartos, possivelmente devido à escassez de dados.

Bathroom: Os imóveis têm entre 0 e 8 quartos de banho, com uma média de 1.5. Existem 3471 propriedades sem informação sobre quartos de banho e 75% têm até 2. O preço tende a subir com o número de quartos de banho, como se vê pelo gráfico, com exceções possivelmente devido à falta de dados. A correlação de 0.477 entre quartos de banho e preço indica influência no valor do imóvel.

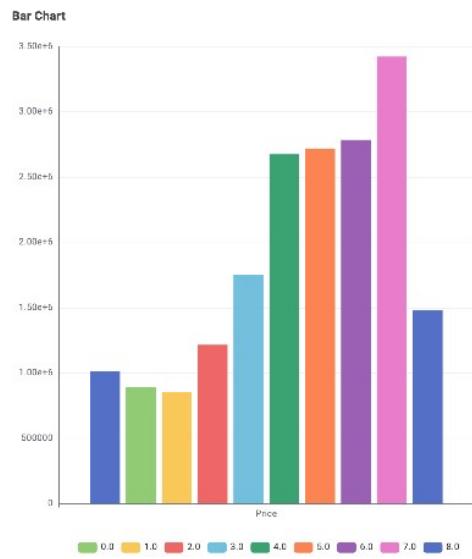


Figura 43-Bar chart bathroom

Car: Os imóveis têm de 0 a 10 lugares de estacionamento, com média de 1.6. Existem 3576 propriedades sem informação sobre estacionamento e 75% têm até 2 lugares. A correlação é de 0.3038 entre estacionamento e preço indica uma relação, embora menor do que outros atributos.

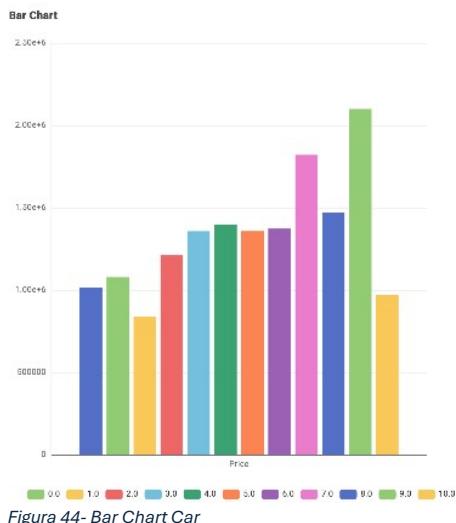


Figura 44- Bar Chart Car

Landsize: O histograma do atributo mostra uma distribuição bimodal. Decidimos então criar categorias para os picos (terrenos pequenos e médios) e para os restantes (grandes terrenos). Terrenos maiores têm preços mais altos, em média. Uma quarta categoria foi criada para valores em falta, de forma a facilitar a escolha de uma estratégia de substituição de missing values numa fase posterior. A correlação entre landsize e price é de 0.3655.

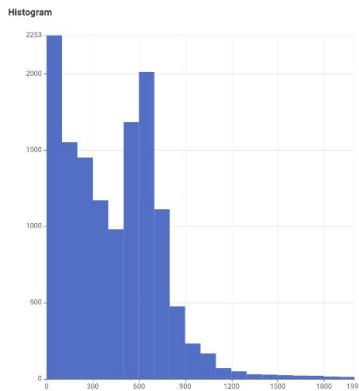


Figura 46-Histograma landsize

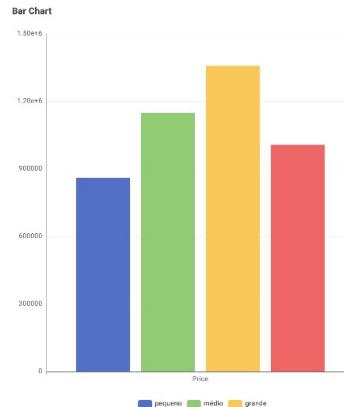


Figura 45- Bar chart landsize

BuildingArea: O espaço de construção varia de 0 a 44,515 m², e os valores da média e do percentil 75 são 151,22 e 174 m² respectivamente. O histograma mostra uma distribuição normal, útil para modelação, embora os dados estejam mais concentrados à esquerda. A correlação com o preço é uma das mais altas no dataset. Existem 10634 valores em falta que requerem estratégias de preenchimento.

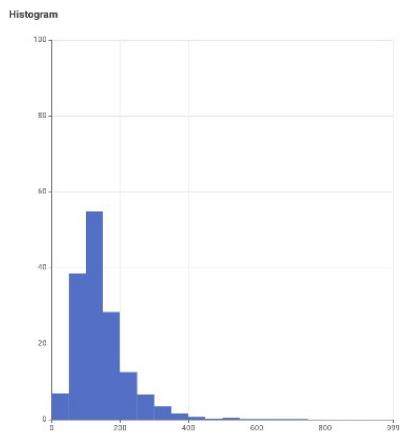


Figura 47-buildingArea histograma

YearBuilt: O valor mínimo do ano de construção é 1196, no entanto, este valor é um *outlier* e como tal deve ser removido. Assim, este atributo passa a estar entre 1830 e 2018. O valor da correlação entre *YearBuilt* e *Price* é -0.3658, o que significa que o preço tende a diminuir à medida que o ano de construção aumenta, como podemos observar no gráfico seguinte, onde dividimos *YearBuilt* em 6 categorias:

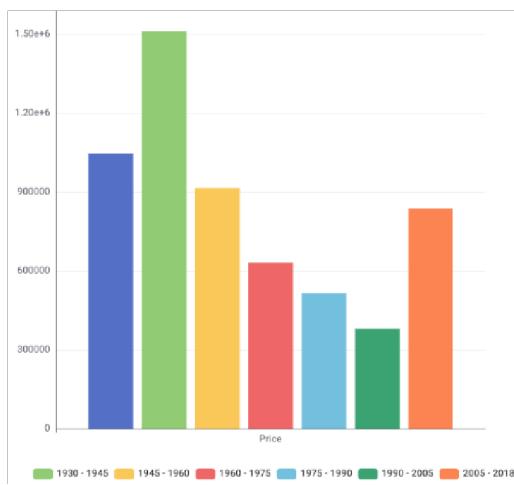


Figura 48- Bar chart year buitl

A única exceção foi entre 2005 e 2018, em que os preços subiram, voltando a estar próximos dos valores de 1945 a 1960.

3.2.3. Localização

Os atributos relacionados com a localização do imóvel são os seguintes: **Suburb; Address; Distance; Postcode; CouncilArea; Latitude; Longitude; RegionName ;PropertyCount**.

Suburb: Existem 308 subúrbios diferentes neste *dataset* e não há nenhum que se destaque em termos de quantidade de imóveis. No entanto, com recurso a um gráfico de barras,

podemos observar que dois subúrbios têm preços, em média, mais elevados: Canterbury e Middle Park.

Address: 98.5% das linhas deste atributo categórico contém valores únicos pelo que o mesmo não oferece valor preditivo e deve ser retirado na fase de tratamento de dados.

Distance: Este atributo segue uma distribuição normal assimétrica à esquerda e tem os seus valores compreendidos entre 0 e 48.1, sendo o valor da média 9.7. A correlação com a variável objetivo é bastante baixa, -0.1173.

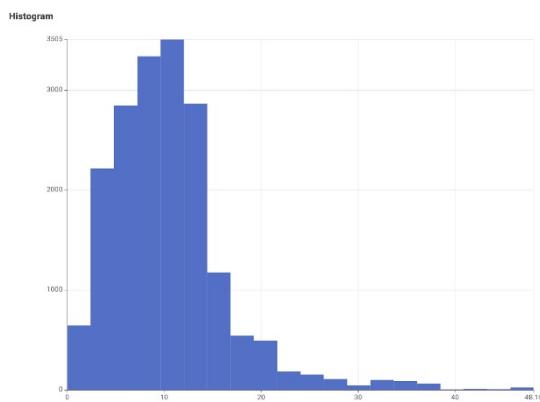


Figura 49-Histograma distance

Postcode: De acordo com o nodo **Rank Correlation**, a correlação *Postcode* e o preço do imóvel, pelo que, neste *dataset*, o código postal tem pouca influência na variável objetivo.

CouncilArea: Neste dataset existem propriedades, no mínimo, em 33 municípios diferentes e, pelo gráfico de barras, podemos observar que alguns destes, têm propriedades com valor médio relativamente alto, algo que poderá ser explorado mais tarde. No entanto, 6163 valores estão em falta, pelo que será necessário arranjar uma estratégia de forma a não perder todas essas linhas.

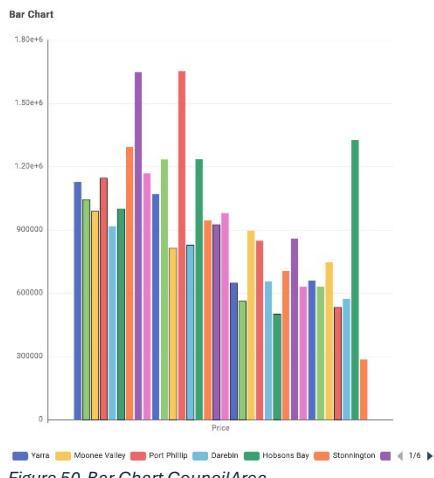


Figura 50-Bar Chart CouncilArea

Latitude e Longitude: Estes dois atributos são semelhantes tanto em termos estatísticos como na forma como se relacionam com o preço. Ambos seguem uma distribuição aproximadamente normal e têm um desvio padrão relativamente baixo, 0.081 e 0.1, indicando o quanto pouco estes valores variam em relação à média. Também os valores de correlação com a variável objetivo são parecidos: 0.2556 para a latitude e 0.267 para a longitude. Seria de esperar que estes dois atributos influenciassem mais o preço, já que a localização é um dos aspectos mais importantes na área do imobiliário. No entanto, tal não acontece pelo que será importante explorar mais em detalhe outros dados relacionados com a localização de forma a perceber que fatores, combinados com a latitude e longitude, contribuem para o preço do imobiliário.

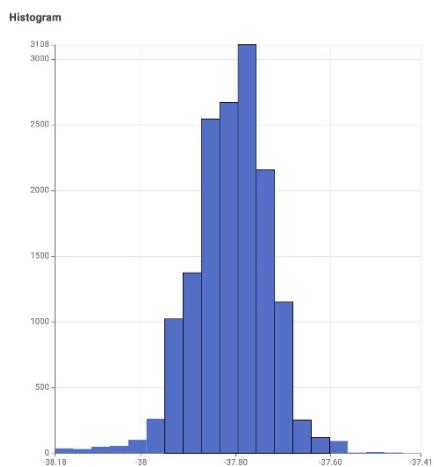


Figura 52- Histograma Latitude

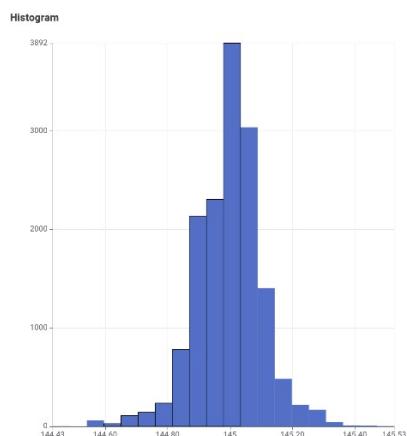


Figura 51-Histograma longitude

RegionName: Existem 8 regiões diferentes no dataset, sendo que a que tem um preço médio mais elevado é a região Southern Metropolitan e a que tem um preço mais baixo é Western Victoria. No entanto, podemos observar pelo gráfico de barras que 4 das 8 regiões

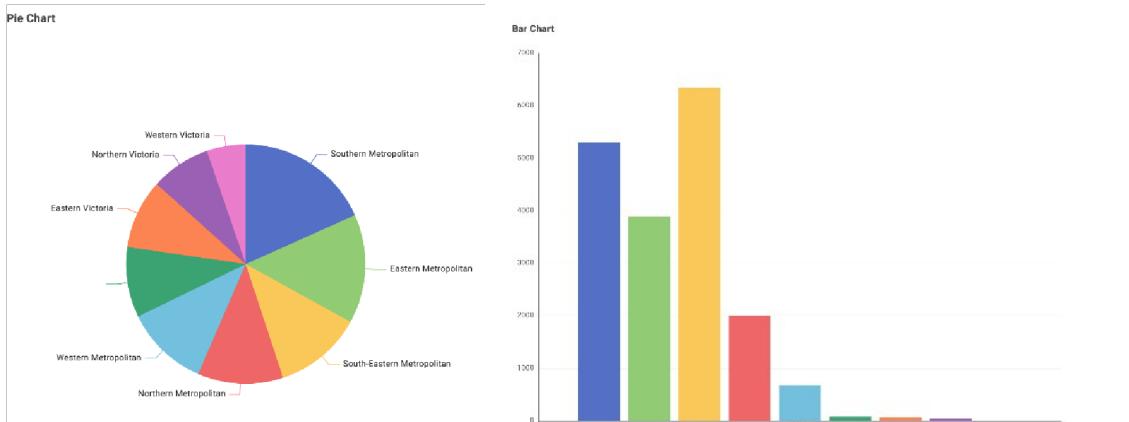


Figura 54- PieChart region Name

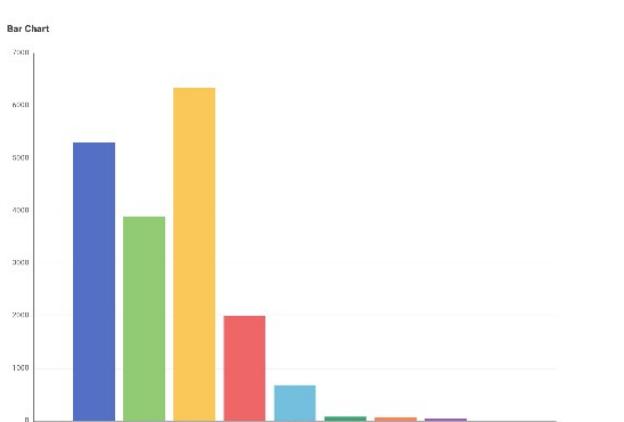


Figura 53- Barchart regionName

estão muito sub-representadas em relação às outras, o que pode levar a conclusões potencialmente enviesadas.

PropertyCount: O atributo *PropertyCount* indica o número de propriedades no subúrbio, daí o número de valores únicos, 324, ser tão próximo com o número total de subúrbio, 330. Para além disso, o número de propriedades varia entre 249 e 21,650 e apresenta um valor de correlação com o preço de apenas 0.032, ou seja, não existe qualquer relação linear entre os dois atributos.

3.2.4. Condições de venda

Por fim, os restantes atributos estão relacionados com a venda do imóvel: **Method;SellerG Date;**

Method: O método de venda do imóvel, aparentemente, não afeta o preço. Isto é evidente pois: O preço tem pouca variação entre categorias, conforme o gráfico de barras, e imóveis vendidos pelo método S (mais comum) têm preços semelhantes aos do método SA (menos comum), reforçando que o método de venda não influencia significativamente o preço.

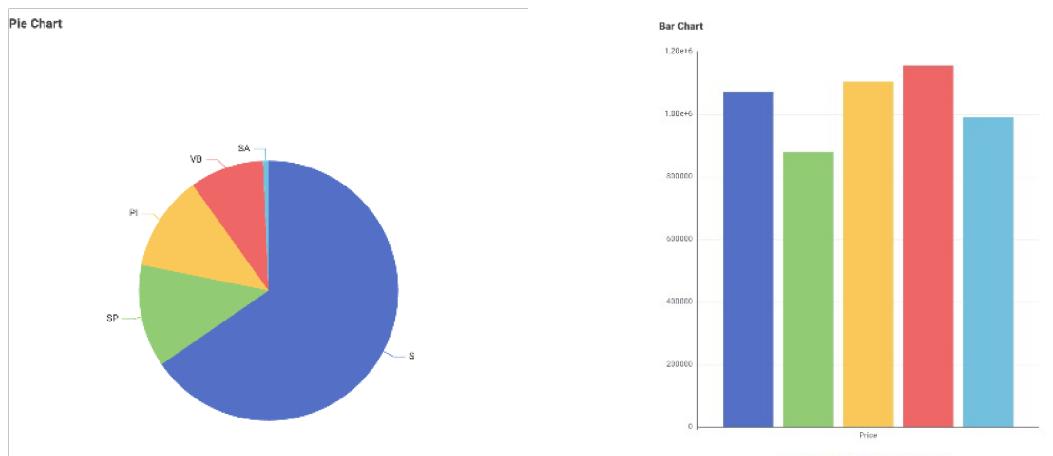


Figura 56-Pie Chart method

Figura 55-Bar chart method

SellerG: Este atributo parece não influenciar o preço da casa por ter um valor de correlação próximo de 0. Para além disso, por conhecimento da área, podemos deduzir que o agente imobiliário provavelmente não é um atributo determinante.

Date: Para conseguirmos analisar este atributo foi necessário utilizar o nodo **String to Date&Time** para converter a o atributo, que originalmente era uma *string*, para o formato *date*. Assim, pudemos observar que a data mais antiga corresponde a 28-01-2016 e a mais recente a 23-09-2017. No total, existem 58 datas distintas no *dataset*.

3.3. Preparação dos dados

Depois da realização do estudo de dados, passámos para a preparação dos mesmos.

- O primeiro passo foi converter o tipo de dados da coluna **Date** de *string* para *date*. Idealmente, este processamento seria feito antes da ingestão dos dados na plataforma Knime, mas o nodo **CSV Reader** não conseguia fazer a conversão, de forma que a fizemos posteriormente com o nodo **String to Date&Time**. Com a coluna neste formato, utilizámos o nodo **Extract Date&Time Fields** para extrair o ano, mês e dia, e depois removemos a coluna *date* original.
- De seguida, retirámos a coluna **Id** porque apenas contém valores únicos, o que não acrescenta valor preditivo a modelos de *machine learning*, pela falta de informação.
- Para além do **Id**, removemos ainda a coluna **Address** também pela elevada cardinalidade (98.5% de valores únicos). Num projeto de maior complexidade, estes valores poderiam ser agrupados por regiões de forma tentar determinar que regiões mais impactam o preço. No entanto, por já termos informação do subúrbio e da região geral, e pela complexidade, decidimos apenas retirar a coluna.
- De seguida, utilizámos o nodo **Category to Number** para converter as colunas categóricas em números através do método **Label encoding**, que atribui um valor único (que começa em 0 e é incrementado por 1) a cada categoria. A utilização do nodo **Category to Number** criou uma coluna nova para cada variável categórica de forma que tivemos de utilizar o nodo **Column Filter** de seguida para remover as colunas originais.
- Para tratar dos *outliers*, decidimos testar duas estratégias, uma em que removemos por completo as linhas que contêm *outliers*, e outra em que os substituímos por *missing values*, os quais são tratados no passo seguinte.
- Através do nodo **Missing Value**, substituímos todos os valores em falta que fossem *strings* pelo valor mais comum, e para os valores numéricos testámos duas abordagens: substituir pelo valor médio e pela mediana, o que, depois da

modelagem, não demonstrou ter um impacto significativo no poder preditivo dos modelos.

- Finalmente, decidimos dividir a *pipeline* de preparação de dados de forma a testarmos se a remoção de variáveis com elevada correlação entre si (valor superior a 0.8) teria um impacto positivo.

3.4. Modelação

Modelação com todos os atributos

Para começarmos com a fase da modelação, decidimos experimentar três modelos - **árvore de regressão simples, regressão linear e regressão polinomial**. Para os *missing*

File	
R ² :	0,54
Mean absolute error:	246 712,245
Mean squared error:	189 578 007 875,831
Root mean squared error:	435 405,567
Mean signed difference:	29 546,035
Adjusted R ² :	0,54

Figura 59-regressão simples

File	
R ² :	0,485
Mean absolute error:	280 379,749
Mean squared error:	212 370 543 422,405
Root mean squared error:	460 836,786
Mean signed difference:	1 370,196
Adjusted R ² :	0,485

Figura 58-regressão linear

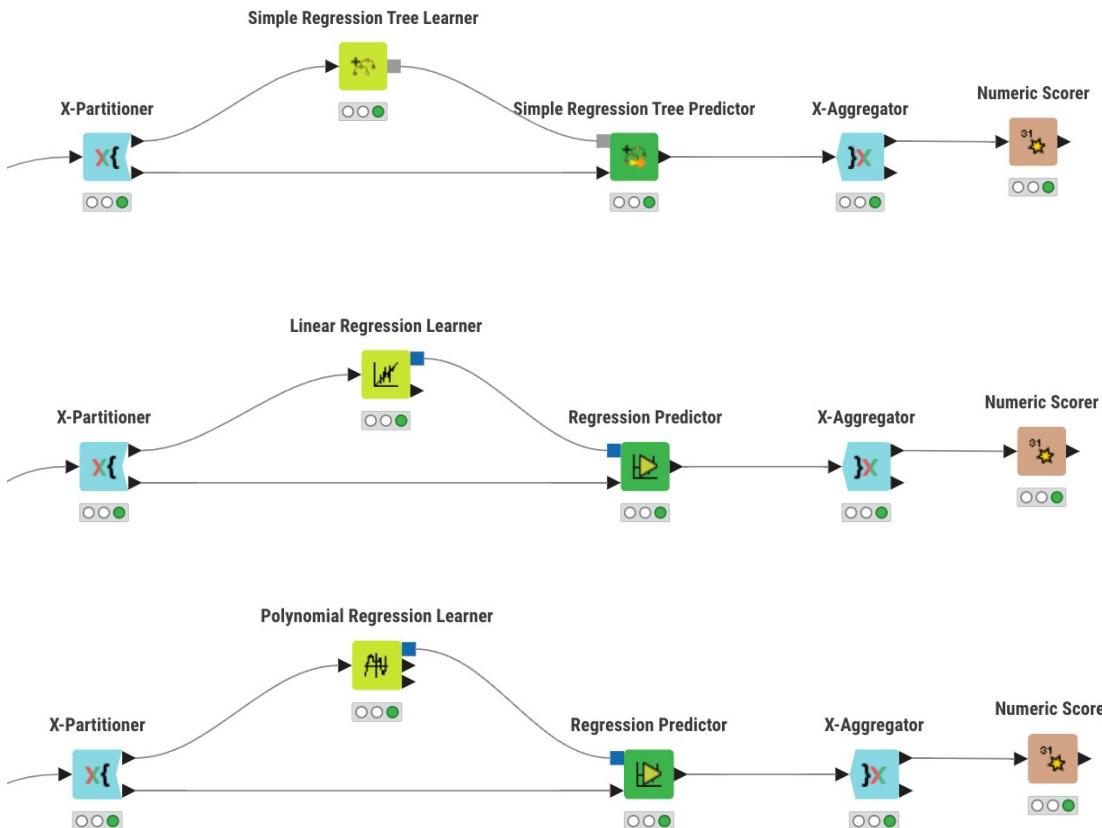
File	
R ² :	-18,435
Mean absolute error:	292 412,873
Mean squared error:	8 008 210 326 955,944
Root mean squared error:	2 829 878,147
Mean signed difference:	-25 406,889
Mean absolute percentage error:	0,304
Adjusted R ² :	-18,435

Figura 57-regressão polinomial

values experimentámos substituir pela média e pela mediana, no entanto não chegamos a nenhuma conclusão dado de que num modelo os resultados foram ligeiramente melhores, noutro ligeiramente piores e no último obtemos aproximadamente os mesmos valores.

Pelas tabelas acima, podemos observar os modelos iniciais tiveram uma performance muito baixa e a principal razão que identificamos foi a elevada quantidade de valores extremos no *dataset*, de forma que prosseguimos com duas estratégias para tratar estes valores.

Fizemos também ***cross-validation*** desde o início para **reduzirmos o risco de overfitting**, ou seja, para garantir que o nosso modelo não apenas se ajusta bem aos dados de treino, mas que também mantém uma boa performance quando exposto a novos dados.



Remoção de linhas com outliers

Como vimos na fase de exploração, este *dataset* contém muitos valores extremos e portanto, como previsto, as métricas dos modelos não são aceitáveis.

Utilizámos então o nodo **Numeric Outliers** para remover todas as linhas com *outliers* e os resultados melhoraram consideravelmente:

File
R ² :
Mean absolute error:
Mean squared error:
Root mean squared error:
Mean signed difference:
Mean absolute percentage error:
Adjusted R ² :

Figura 62-regressão simples

File
R ² :
Mean absolute error:
Mean squared error:
Root mean squared error:
Mean signed difference:
Mean absolute percentage error:
Adjusted R ² :

Figura 60-regressão linear

File
R ² :
Mean absolute error:
Mean squared error:
Root mean squared error:
Mean signed difference:
Mean absolute percentage error:
Adjusted R ² :

Figura 61-regressão polinomial

Substituição de outliers por missing values

Para esta fase da modelação decidimos experimentar substituir todos os *outliers* por valores em falta e depois testar como anteriormente – substituindo pela média e mediana.

No entanto, com esta alteração os resultados pioraram consideravelmente.

R ² :	0,379
Mean absolute error:	224 740,048
Mean squared error:	116 610 687 180,651
Root mean squared error:	341 483,07
Mean signed difference:	26 665,129
Mean absolute percentage error:	0,24
Adjusted R ² :	0,379

Figura 65-regressão simples

R ² :	0,503
Mean absolute error:	229 121,461
Mean squared error:	93 308 442 266,215
Root mean squared error:	305 464,306
Mean signed difference:	-26,098
Mean absolute percentage error:	0,265
Adjusted R ² :	0,503

Figura 63-regressão linear

R ² :	0,535
Mean absolute error:	222 335,878
Mean squared error:	87 296 973 663,619
Root mean squared error:	295 460,613
Mean signed difference:	-171,182
Mean absolute percentage error:	0,257
Adjusted R ² :	0,535

Figura 64-regressão polinomial

Modelação com normalização

De forma a garantir que os dados numéricos estejam na mesma escala, mas que mantenham a distribuição original, utilizámos o nodo **Normalizer** para fazer normalização MinMax entre 0 e 1 e aplicámos a todos os modelos anteriores.

Os resultados foram semelhantes aos obtidos anteriormente, mas decidimos fazer normalização para todos os modelos seguintes, dado que este método apresenta outras vantagens como o aumento da velocidade de convergência para algoritmos que usam *gradient descent* como técnica de otimização.

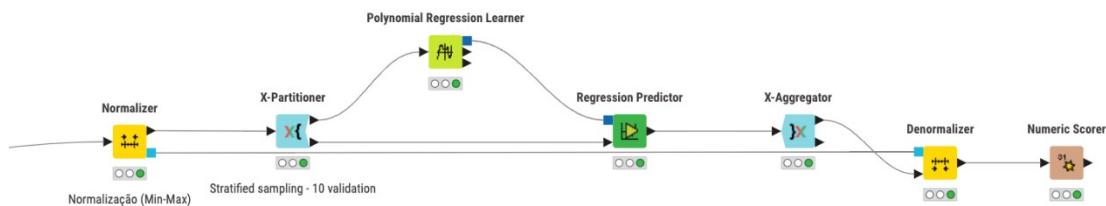


Figura 66- Modelo com normalizer

Modelação com feature selection

Na fase de exploração de dados, foram identificadas algumas *features* que provavelmente teriam pouco valor preditivo na fase de modelação. No entanto, decidimos não retirar por completo estas colunas, devido à probabilidade da análise de dados realizada anteriormente não capturar a existência de relações complexas que as variáveis tenham entre si.

Decidimos então prosseguir com a seleção de *features* usando os nodos de **Feature Selection**. Novamente, testamos todos os modelos anteriores, mas com normalização.

A conclusão a que chegamos é que todas as *features* são importantes para estes modelos iniciais (árvore de regressão simples, regressão linear e regressão polinomial) dado que, consistentemente, os melhores modelos foram os que usaram todas as *features*. Por exemplo, para o melhor modelo - regressão polinomial onde as linhas com *outliers* são removidas e os valores em falta são substituídos pela média - fazendo *feature selection*

com 200 combinações diferentes e maximizando o R-quadrado, o melhor resultado ocorre quando todas as 21 colunas são utilizadas:

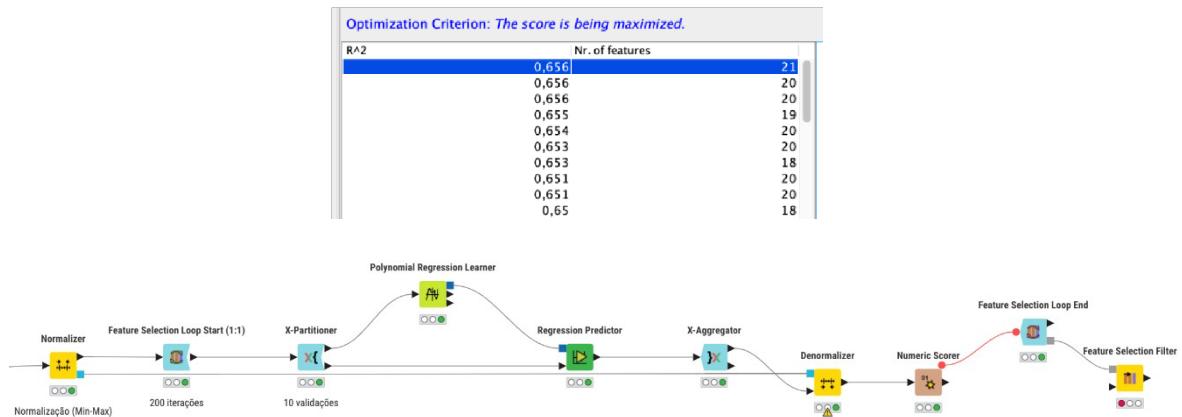


Figura 67-Modelação com feature selection

Gradient Boosted Tree

O próximo passo na tarefa de modelação foi encontrar um modelo mais robusto para melhorarmos a qualidade dos resultados. Para isso, decidimos utilizar uma *Gradient Boosted Tree*, que tem várias vantagens em comparação com os modelos anteriores, nomeadamente a capacidade de captação de relações não lineares e a maior facilidade em absorver o impacto de *outliers*.

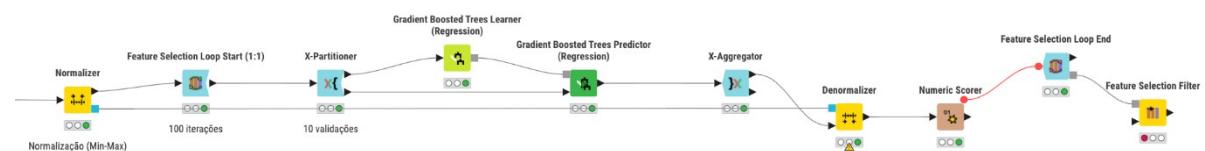


Figura 68- Gradient boosted tree model

A utilização deste novo modelo teve um impacto elevado na qualidade das previsões, com o melhor modelo, representado na figura acima, a ter um valor de R-quadrado próximo de 0.8 comparado com o valor observado anteriormente de 0.656.

Novamente, o nodo de **Feature Selection** indicou que a utilização de todas as *features* leva ao melhor resultado possível quando o critério de otimização é o R-quadrado.

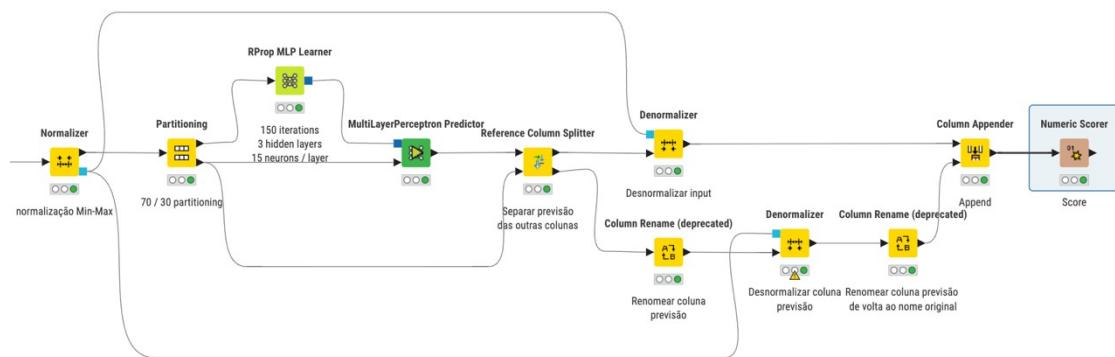
R^2	Nr. of features
0,788	21
0,787	20
0,785	20
0,783	18
0,783	20
0,782	16
0,78	18

Figura 69- Informação do feature selection

Apesar da implementação de *Gradient Boosted Trees* produzir modelos superiores, o tempo e a utilização de recursos como GPU e CPU aumenta consideravelmente. Cada iteração do nodo **Feature Selection** demorou aproximadamente 10 vezes mais quando comparado com a regressão polinomial utilizada anteriormente.

Modelação com redes neurais

Finalmente, utilizamos o nodo **RProp MLP Learner** para treinar uma rede neuronal com 3 *hidden layers* e 15 neurónios por *layer*. Inicialmente, foi necessário normalizar todos os dados incluindo a variável objetivo e, após treinar a rede, de forma a desnormalizar a previsão, utilizámos o nodo **Reference Column Splitter** para isolar este novo atributo e desnormalizá-lo à parte antes de o juntar com o resto dos dados para correr o nodo **Numeric Scorer**.



Desta forma, conseguimos explorar a criação de uma rede neuronal *feedforward* e que pode facilmente ser testada para várias arquiteturas ajustando os valores no no **RProp MLP Learner**.

As métricas de avaliação foram ligeiramente inferiores comparadas com o a *Gradient Boosted Tree* que criamos anteriormente, na qual obtemos um R-quadrado de 0.788.

Statistics - 5:33 - Numeric Scorer (Score)	
File	
R ² :	0,718
Mean absolute error:	167 581,584
Mean squared error:	52 963 598 895,346
Root mean squared error:	230 138,217 Open view
Mean signed difference:	-911,804
Mean absolute percentage error:	0,181
Adjusted R ² :	0,718

Experimentámos também várias combinações de hiperparâmetros, onde alterámos o número de iterações, o número de *layers* e o número de neurónios por *layer*. Os resultados podem ser observados em anexo.

3.5. Avaliação

Os modelos de árvore de decisão e regressão polinomial apresentaram melhorias significativas após a remoção de *outliers* e a normalização dos dados. No entanto, o modelo que se destacou foi o *Gradient Boosted Tree*, que apresentou um R-quadrado de 0.788, demonstrando uma elevada capacidade de prever o preço do imóvel de acordo com as características disponíveis.

Apesar dos bons resultados com *Gradient Boosted Trees*, observou-se um aumento considerável no tempo de processamento e uso de recursos, o que pode ser uma limitação em ambientes com restrições de *hardware*.

Feature Selection

A seleção de *features* indicou que todas as variáveis inicialmente consideradas eram importantes para os modelos testados. Isso reforça a complexidade do mercado imobiliário e a interdependência entre as várias características dos imóveis.

Missing values e outliers

As estratégias para lidar com os dados em falta e *outliers* foram fundamentais para melhorar a *performance* dos modelos devido à elevada quantidade de *missing values* em atributos alguns dos atributos mais importantes. Desta forma, no futuro pretendemos implementar novas estratégias para lidar com *missing values*, nomeadamente a imputação com recurso ao algoritmo KNN, que utiliza dados próximos para estimar o valor em falta.

Para os *outliers*, vamos considerar a implementação de técnicas mais robustas para os detetar e tratar, por exemplo, através da análise de componentes principais - PCA.

4. Aspetos a melhorar

Durante a realização deste trabalho diversas técnicas foram utilizadas, tanto na preparação dos dados como nos modelos criados. No entanto, à medida que explorávamos cada vez mais os modelos, novas ideias de como preparar os dados e os modelos iam surgindo. Gostaríamos de ter explorado mais formas de tratamento de *outliers* e *missing values*, assim como procurar relações entre variáveis que nos permitissem otimizar os nossos modelos. Também poderíamos ter explorado melhor os hiper parâmetros, sempre com objetivo de obter o melhor modelo possível. Relativamente ao relatório também

gostaríamos de ter melhorado a justificação para a retirada das colunas do *day_of_birth* e *month_of_birth* com mais argumentos para além da justificação através do contexto.

5. Conclusão

Ao longo deste trabalho aplicámos vários conceitos de desenvolvimento de modelos de aprendizagem. Explorámos e pré-processámos dados para compreender melhor o problema em estudo. Apesar das dificuldades iniciais na escolha do dataset para a Tarefa 2, estamos satisfeitos com o resultado. Conseguimos desenvolver modelos de aprendizagem satisfatórios para os datasets estudados e documentámos todo o processo de desenvolvimento do projeto de forma breve, mas detalhada. Acreditamos que este projeto demonstra a nossa capacidade de aplicar conceitos teóricos na prática.

6. Anexos

i. Anexo

Imagens Preparação dos dados dataset tarefa1

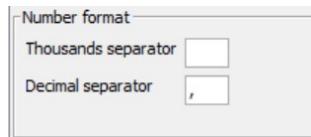


Figura 70-Nodo csv reader

Column	Exclude Column	No. missings	Unique values	All nominal values
Sex	<input type="checkbox"/>	0	3	m, f, mm

Figura 71- Data Explorer View atributo "sex"

ALP	<input type="checkbox"/>	0	415	NA, 52,5, 61,2, 84,1, 59,5, t 1
-----	--------------------------	---	-----	--

Figura 72-Excerto do Data Explorer View com NA

```

if (c_ALP.equals("NA")) {
    out_ALP = null;
} else {
    out_ALP = c_ALP;
}

if (c_CHOL.equals("NA")) {
    out_CHOL = null;
} else {
    out_CHOL = c_CHOL;
}

if (c_ALT.equals("NA")) {
    out_ALT = null;
} else {
    out_ALT = c_ALT;
}

if (c_PROT.equals("NA")) {
    out_PROT = null;
} else {
    out_PROT = c_PROT;
}

if (c_ALB.equals("NA")) {
    out_ALB = null;
} else {
    out_ALB = c_ALB;
}

```

Figura 73-Excerto Java Snippet

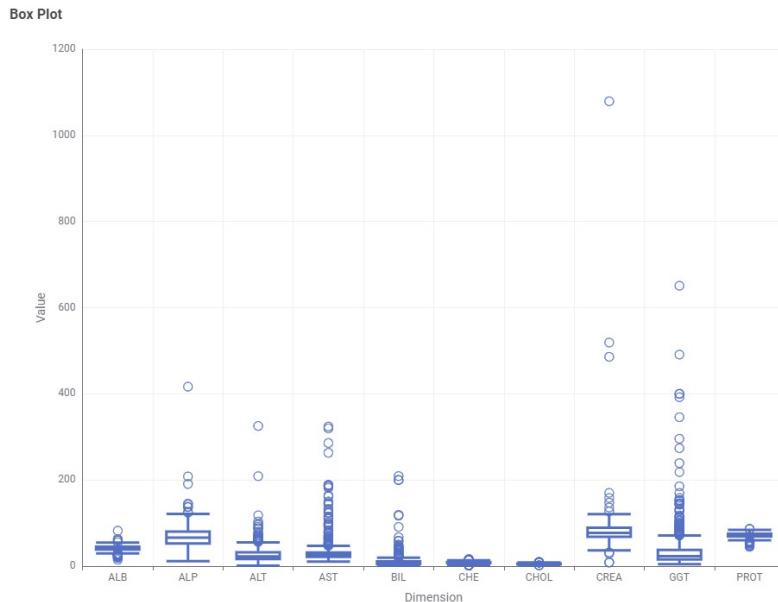


Figura 74- Box Plot outliers

ii. Anexo – Modelos

Tratamento Missing Values	Tratamento Outliers	Iterações	Hidden Layers	Neurónios / Layer	R2	MAE	RMSE
Média	Remover linhas	100	3	10	0.698	174677.96	237981.09
Média	Remover linhas	100	3	15	0.708	171128.07	234078.29
Média	Remover linhas	100	3	20	0.706	171909.52	234888.53
Média	Remover linhas	150	3	10	0.71	169969.70	233322.27
Média	Remover linhas	150	3	15	0.718	167581.58	230138.21
Média	Remover linhas	150	3	20	0.716	168317.43	231128.56
Média	Remover linhas	200	3	10	0.716	168255.57	231123.12
Média	Remover linhas	200	3	15	0.72	166766.29	229312.0
Média	Remover linhas	200	3	20	0.724	164983.52	227871.23
Média	Remover linhas	100	4	10	0.711	170013.32	232814.87
Média	Remover linhas	100	4	15	0.699	173885.0	237825.56
Média	Remover linhas	100	4	20	0.698	173497.64	238228.32
Média	Remover linhas	150	4	10	0.717	167693.53	230398.72
Média	Remover linhas	150	4	15	0.716	168163.50	230999.57
Média	Remover linhas	150	4	20	0.715	168946.06	231322.53
Média	Remover linhas	200	4	10	0.721	166024.43	228967.75
Média	Remover linhas	200	4	15	0.722	165425.32	228339.0
Média	Remover linhas	200	4	20	0.724	164932.56	227592.22

Figura 79 – Resultados das redes neuronais (regressão)

Tratamento Missing Values	Tratamento Outliers	Modelo	Feature Selection - 100 iterações	Feature Selection - nº features	Normalização	R2	MAE	RMSE
Média	Remover linhas	Gradient Boosted Tree	Sim	-	21 Sim	0.788	142371.22	198794.26
Mediana	Remover linhas	Gradient Boosted Tree	Sim	-	21 Sim	0.787	142952.79	199395.57
Média	Substituir por missing values	Gradient Boosted Tree	Sim	-	21 Sim	0.676	173246.10	246579.27
Mediana	Remover linhas	Polynomial Regression	Não	-	-	Não	0.658	192948.96
Mediana	Remover linhas	Polynomial Regression	Não	-	-	Sim	0.657	193737.96
Média	Remover linhas	Polynomial Regression	Não	-	-	Sim	0.656	192599.06
Média	Remover linhas	Polynomial Regression	Não	-	-	Não	0.656	192976.56
Mediana	Remover linhas	Polynomial Regression	Sim	-	20 Sim	0.656	193076.23	253214.77
Média	Remover linhas	Polynomial Regression	Sim	-	21 Sim	0.656	193334.89	253266.94
Mediana	Substituir por missing values	Gradient Boosted Tree	Sim	-	20 Sim	0.648	179405.14	257368.06

Figura 79 – 10 melhores modelos de regressão (não inclui redes neuronais)