Universidade do Minho
Escola de Engenharia

Licenciatura em Engenharia Informática

Aprendizagem e Decisão Inteligentes

Trabalho Prático

Grupo 10

Jéssica Cunha (a100901)

Martim Redondo (a100664)

Rodrigo Castro (a100694)

Tiago Moreira (a100541)

Ano letivo 2023/24

Índice

Índice de figuras

Introdução

Este relatório emerge no âmbito da Unidade Curricular de Aprendizagem e Decisão Inteligentes, onde nos foi incumbida a conceção de modelos de aprendizagem. O trabalho proposto abrange duas tarefas distintas. A primeira consiste na consulta, análise, exploração e preparação de um conjunto de dados selecionado pelos membros do nosso grupo. Enquanto isso, a segunda tarefa envolve a exploração, análise e preparação de um conjunto de dados designado pelos docentes da Unidade Curricular.

1. Dataset predefinido

Na primeira etapa do projeto, recebemos um dataset. Este continha informações variadas, incluindo dados comuns, como gênero, idade e ano de nascimento, bem como informações mais específicas, tais como Total Bilirubin, Alanine Aminotransferase, Albumin, entre outras.

A metodologia que será utilizada no processo da resolução do problema é o CRISP-DM. O modelo CRISP-DM define um guião para o desenvolvimento de projetos de análise de dados dividido em 6 etapas, contudo serão apenas visadas as primeiras 5, que são:

- Estudo do negócio
- Estudo dos dados
- Preparação dos dados
- Modelação
- Avaliação'

A etapa 6, sendo a implementação prática do projeto no mundo real, será ignorada nesta análise.

Depois de uma pequena introdução sobre esta segunda parte do projeto daremos início à planificação, começando, claramente, com o estudo do negócio.

1.1. Estudo do negócio

O objetivo que buscamos abordar ao aplicar um modelo do KNIME a este conjunto de dados, é a capacidade de prever se uma pessoa possui ou não uma doença hepática, com base em informações médicas derivadas de testes realizados.

Para resolver esse problema, é essencial realizar um estudo prévio e estabelecer os passos necessários para alcançar o nosso objetivo. A primeira etapa envolve a análise minuciosa dos dados para compreender sua natureza e identificar eventuais lacunas ou áreas desconhecidas no conjunto de dados. Em seguida, podemos começar a explorar a correlação entre as variáveis e a criar informações relevantes a partir das existentes. Isso nos proporcionará uma compreensão mais profunda dos fatores que influenciam o atributo que queremos como o centro das "operações" o *selector*.

Posteriormente, é crucial verificar se há inconsistências no *dataset* fornecido. Caso sejam identificados problemas, é imperativo resolvê-los, pois esses erros podem afetar negativamente a modelagem subsequente e, consequentemente, a resolução do problema.

Por fim, desenvolvemos modelos de aprendizado de máquina que serão responsáveis por nos fornecer uma solução para o problema. No entanto, é de extrema importância escolher o modelo mais adequado para resolver a nossa questão específica, para isso é preciso testar e verificar o nível de sucesso.

Cada um destes passos é essencial e não pode ser ignorado, pois sem eles, seria impossível obter uma resposta confiável para nosso problema.

1.2. Análise dos dados

O *dataset*, cujo nome é ilp.csv, contem 583 linhas e 16 colunas. Cada coluna é referente a um atributo aos quais passaremos a citar e a apresentar uma breve explicação, já que alguns são termos mais da vertente científica:

- *id_code*: identificador única para cada pessoa;
- *age*: idade em anos;
- *birth_year*: ano em que nasceu;
- *birth_month*: mês em que faz anos;
- *birth_date*: data de nascimento, com dia e mês a zeros, ou seja, segue uma estrutura deste estilo yyyy/00/00;
- *Gender*: gênero;
- *TB*: A *bilirubin* é um pigmento amarelo produzido durante a quebra normal dos glóbulos vermelhos no corpo. *TB* faz referência a *Total bilirubin* é formada por duas partes a *Direct Bilirubin* e a *Indirect Bilirubin*. Serve para avaliar se a pessoa sofre de problemas em certas partes do corpo, como fígado ou no sistema biliar.
- *DB*: *Direct Bilirubin* é uma das partes que compões a *Total bilirubin*;
- *Alkphos*: *Alkaline Phosphotase* é uma substância presente na nossa corrente sanguínea. Testes ao sangue podem inferir a quantidade de *Alkphos* que o indivíduo tem no sangue. Serve para avaliar se a pessoa sofre de problemas no fígado ou na vesícula.
- *SGPT*: *Alamine Aminotransferase* é uma enzima que é liberada pelo fígado ou vesícula biliar quando há problemas de saúde. Testes ao sangue para medirem este atributo é devido a suspeitas de problemas no fígado ou vesícula biliar;
- *SGOT*: *Aspartate Aminotransferase* é uma enzima que é liberada quando há problemas no fígado ou no coração, é muito parecido ao *SGPT*, mas menos precisa quando o objetivo é perceber problemas no fígado.
- *TB(#1):* Consumo total de proteínas. Estudos nesta área pode ser devido a suspeitas de problemas de saúde no fígado (como a muitas outras áreas), contudo teria de ser usado em complemento com outros fatores por não ser preciso.
- *ALB*: *Albumin* é uma proteína, qualquer teste feito envolvendo ao sangue com pretexto de obter o resultado deste atributo é por suspeita de problemas no fígado ou problemas nutricionais. Sendo que, os problemas nutricionais viriam do problema de fígado.
- *CHOL*: Colesterol.
- *AG_ratio*: *Albumin and Globulin Ratio* é um teste que se faz quando se suspeita de doenças hepáticas.
- *BlLmg*: *Bilirubin* (mg/dL);
- *Selector*: Indica se a pessoa sofre ou não de doença hepática.

Como já foi inferido acima, o *Selector* será o atributo que será o centro de estudo para conseguir resolver o problema ao qual nos propusemos.

1.2.1. Selector

O *selector* diz-nos se a pessoa sofre ou não de doença hepática. Como podemos ver pelo gráfico abaixo, no nosso *dataset* 167 pessoas não sofrem de doenças hepáticas, sendo essa variável representada pela sigla *NLD (no liver disease*). Enquanto, as outras 71.36% sofrem de doenças hepáticas que corresponde a 416 pessoas. A sigla que as representa é *LD (liver desease).*

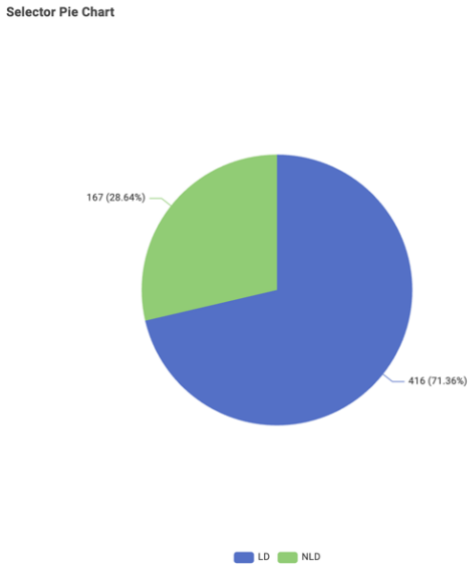

Figura 1 - Divisão entre pessoas com e sem doença hepática

1.2.2. Faixa_etaria

No *dataset*, há cinco faixas etárias diferentes, com a distribuição da seguinte forma: 6 crianças, 30 adolescentes, 142 jovens, 306 adultos e 99 idosos.

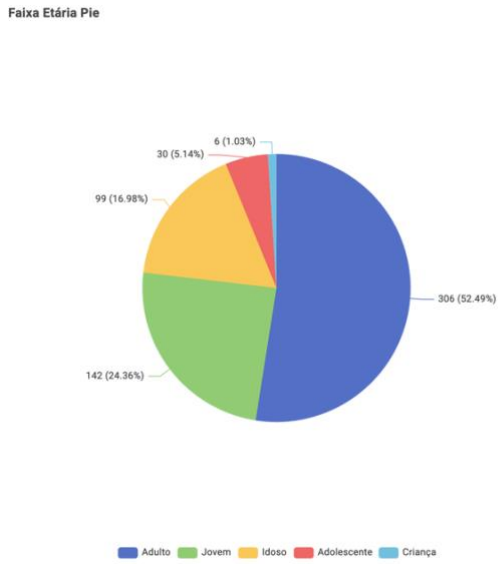

Figura 2 - Divisão das pessoas por faixas etárias

Conforme observado no gráfico abaixo, a idade emerge como um indicador potencial para o desenvolvimento de doenças hepáticas. Embora não possamos afirmar com certeza, é evidente que a proporção de indivíduos doentes em relação aos saudáveis é mais elevada na fase adulta, sugerindo uma possível maior propensão a esse problema nessa faixa etária. Por outro lado, na infância, observa-se o fenômeno oposto. No entanto, devido à escassez de dados, a confiabilidade dessa observação é limitada.

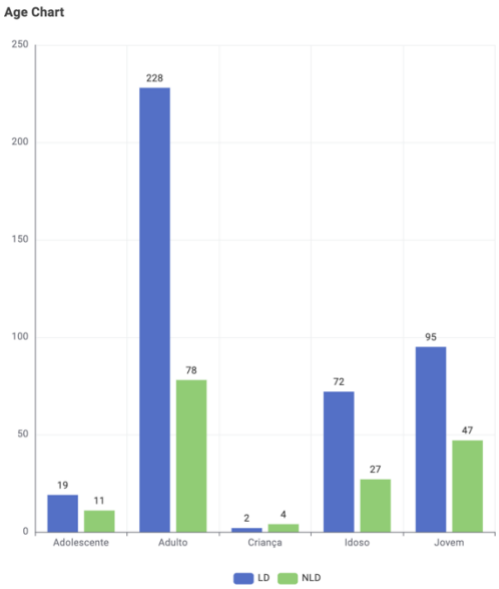

Figura 3 - Distribuição do *selector* tendo em conta

1.2.3. Birth_month

Como podemos ver pelo gráfico circular abaixo as pessoas, no *dataset*, estão bem distribuídas quanto ao mês de nascimento. Sendo que a maior fatia pertence a Maio e a menor pertence a Janeiro.

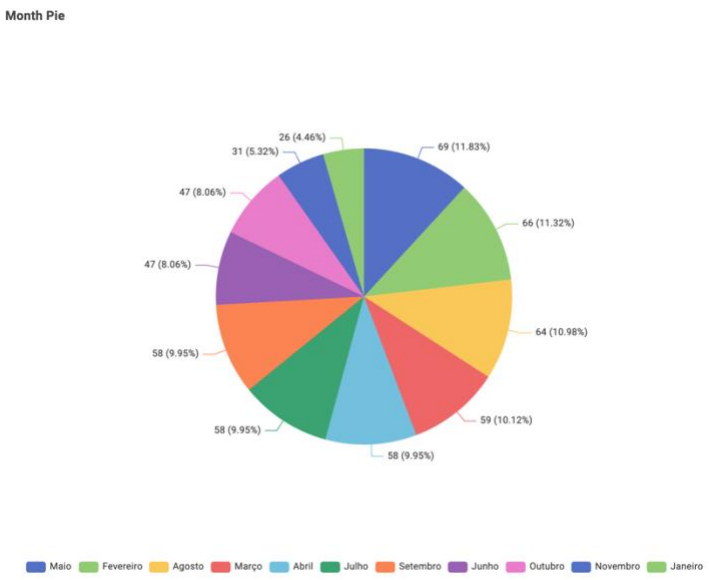

Figura 4 - Divisão das pessoas por mês de nascimento

Como podemos observar no gráfico de barras mostra há diferenças na proporção doentes/não doentes dependendo do mês em que nasceram. Poderá ser um indício de que o mês de nascimento será um fator que afeta ou não o aparecimento da doença.

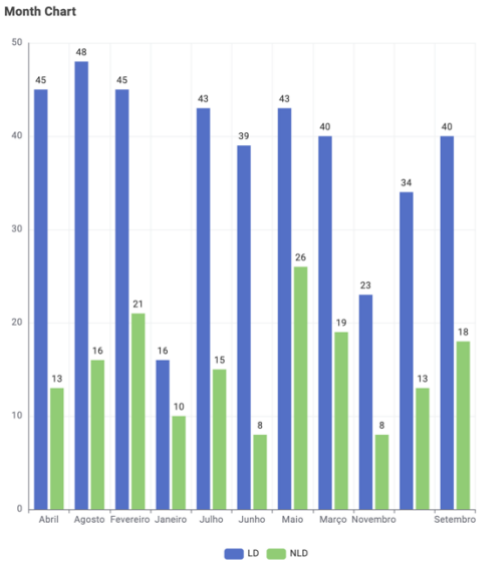

Figura 5 - Distribuição do *selector* tendo em conta o mês de nascimento

1.2.4. Gender

No *dataset*, o número de homens e mulheres é bastante desequilibrado, pois contamos com 441 homens e, apenas, 142 mulheres, como podemos verificar no gráfico abaixo:

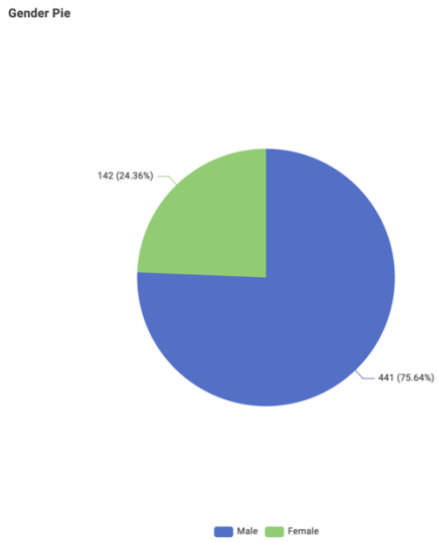

Figura 6 - Distribuição do *gender*

Embora o número de mulheres seja menor, como podemos ver no gráfico abaixo, nada garante que ser homem ou mulher afete no surgimento de doenças hepáticas, pois a proporção doente/não doente entre homens e mulheres é bastante semelhante.

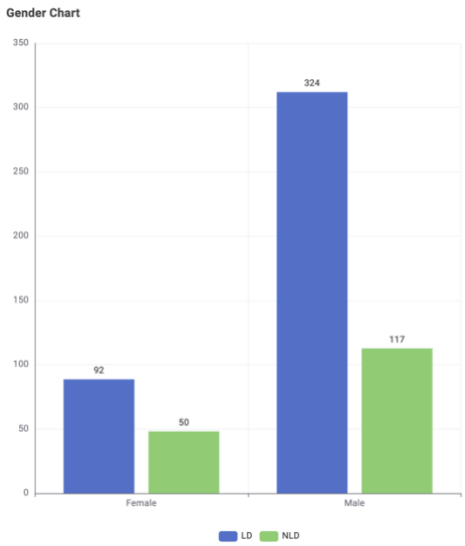

Figura 7 - Distribuição do *selector* tendo em conta o *gender*

1.2.5. TB

No *dataset*, a média de *Total Bilirubin (TB)* em uma pessoa doente é de 4.164, enquanto para uma pessoa que não sofre de doença hepática é de 1.143. Além disso, as informações sobre o mínimo e o máximo de *TB* são cruciais. Quando a pessoa está doente, o valor mínimo é de 0.4 e o máximo é de 75. Por outro lado, para aqueles que não estão doentes, o mínimo é de 0.5 e o máximo é de 7.3.

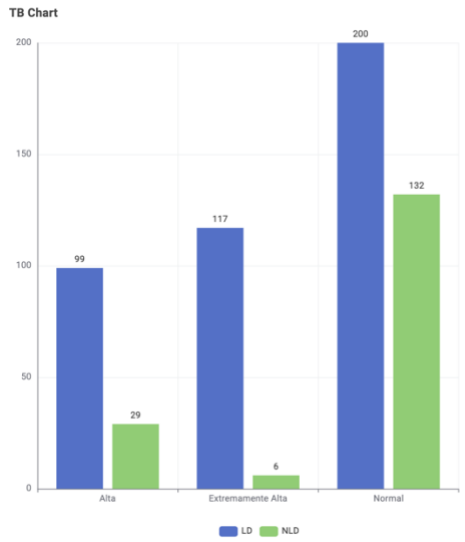

Figura 8 - Distribuição do *selector* tendo em conta o *TB*

1.2.6. DB

O *Direct Bilirubin (DB)*, no nosso dataset, tem as seguintes métricas:

DB *String*	Média *Number (double)*	Máximo *Number (double)*	Mínimo *Number (double)*
LD	1.924	19.7	0.1
NLD	0.396	3.6	0.1

Figura 9 - Métricas do *DB* (média, máximo e mínimo) para doente e não doente

Pode-se observar no gráfico abaixo que conforme os níveis de DB aumentam, também aumenta a proporção de indivíduos doentes em relação aos saudáveis. Isso sugere que o aumento do DB pode desempenhar um papel decisivo no desenvolvimento de doenças hepáticas. Contudo, não há como ter certezas nesta fase da análise.

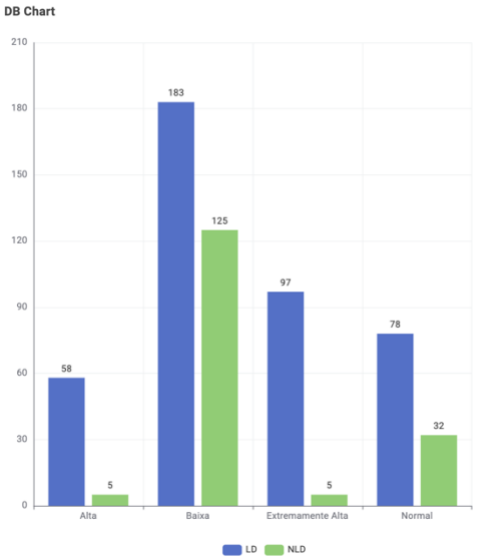

Figura 10 - Distribuição do *selector* tendo em conta o *DB*

1.2.7. Alphkos

No *dataset*, a média de *Alkaline Phosphatase (Alphkos)* em pessoas doentes é de 319.094, enquanto para aqueles que não sofrem de doença hepática é de 219.754. Além disso, informações sobre os valores mínimo e máximo de *Alphkos* são essenciais. Entre os indivíduos doentes, o valor mínimo é de 63 e o máximo é de 2110. Em contrapartida, para os não doentes, os valores variam de 90 a 1580.

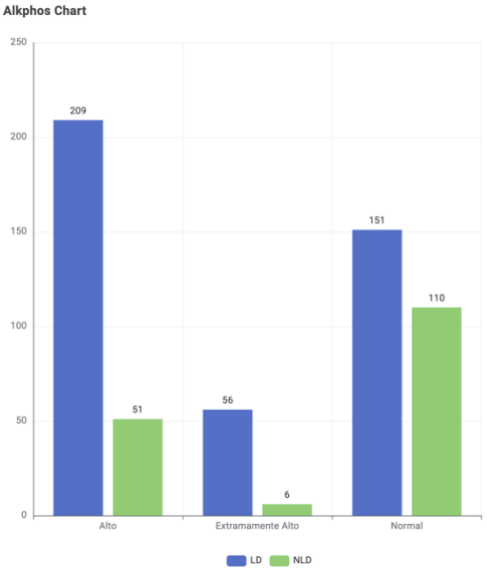

Figura 11 - Distribuição do *selector* tendo em conta o *Alphkos*

No caso do *Alphkos*, não conseguimos tirar algo muito conclusivo por talvez falta de dados nos casos extremamente altos, mas é uma possibilidade que quanto maior o Alphkos, maior a chance de sofrer de doenças hepáticas.

1.2.8. SGPT

No *dataset*, a média de *Alamine Aminotransferase (SGPT)* em pessoas doentes é de 99.606, comparada a 33.653 em indivíduos sem doença hepática. É crucial também considerar os valores mínimo e máximo de *SGPT*. Para os doentes, esses valores variam de 12 a 2000, enquanto para os não doentes, a variação é de 10 a 181.

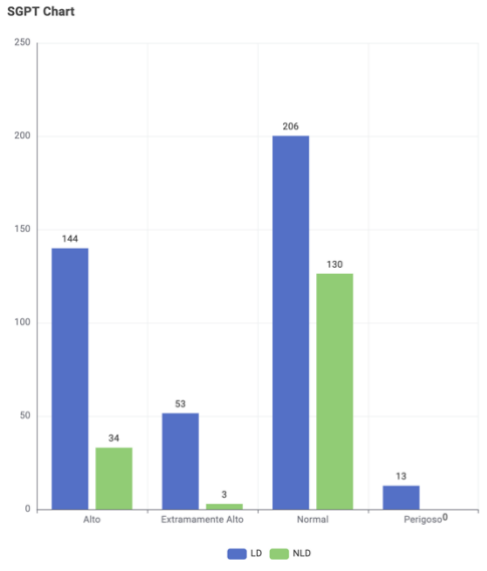

Figura 12 - Distribuição do *selector* tendo em conta o *SGPT*

1.2.9. SGOT

No *dataset*, a média de *Aspartate Aminotransferase (SGOT)* em pessoas doentes é de 137.7 comparada a 40.689 em indivíduos sem doença hepática. É crucial também considerar os valores mínimo e máximo de *SGPT*. Para os doentes, esses valores variam de 11 a 4929, enquanto para os não doentes, a variação é de 10 a 285.

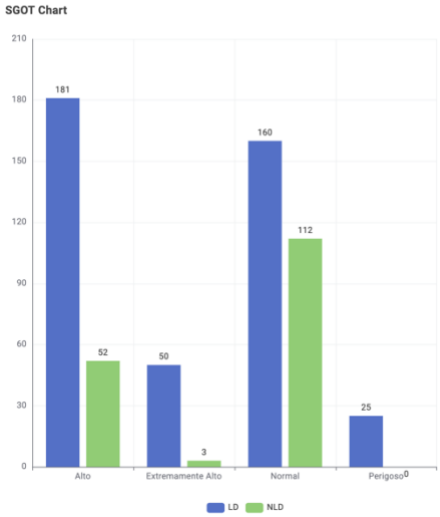

Figura 13 - Distribuição do *selector* tendo em conta o *SGOT*

1.2.10. Total Protein

No conjunto de dados, a média de *Total Proteins* em pessoas doentes é de 6.459, em comparação com 6.543 em indivíduos sem doença hepática. É fundamental também considerar os valores mínimo e máximo de *Total Proteins*. Para os doentes, esses valores variam de 2.7 a 9.6, enquanto para os não doentes, a variação é de 3.7 a 9.2. Notavelmente, em comparação com outros atributos, este apresenta uma média muito semelhante tanto para indivíduos doentes quanto para não doentes.

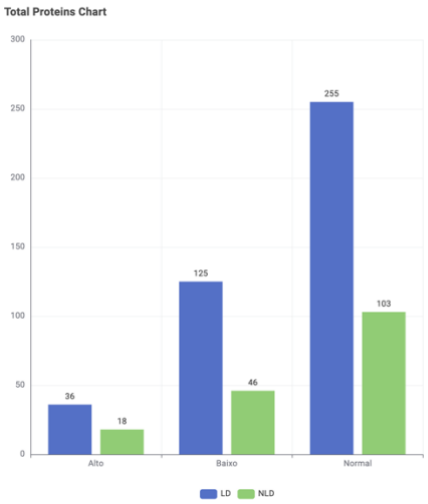

Figura 14 - Distribuição do *selector* tendo em conta o *Total Protein*

1.2.11. ALB

No *dataset*, a média de *Albumin* em pessoas doentes é de 3.061, comparada a 3.344 em indivíduos sem doença hepática. É crucial também considerar os valores mínimo e máximo de *Albumin*. Para os doentes, esses valores variam de 0.9 a 5.5, enquanto para os não doentes, a variação é de 1.4 a 5.

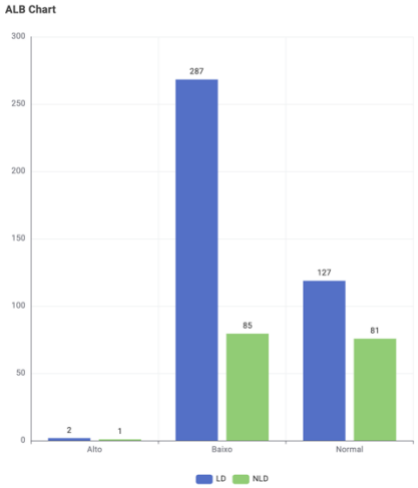

Figura 15 - Distribuição do *selector* tendo em conta o *ALB*

1.2.12. AG_ratio

No *dataset*, a média de Total *AG_Ratio* em pessoas doentes é de 0.916, em comparação com 1.033 em indivíduos sem doença hepática. É importante considerar também os valores mínimo e máximo de *AG_Ratio*. Para os doentes, esses valores variam de 0.3 a 2.8, enquanto para os não doentes, a variação é de 0.4 a 1.9. Esses dados são cruciais para entender a variação dos níveis de *AG_Ratio* entre os grupos doentes e não doentes, fornecendo insights valiosos para a análise e diagnóstico de doenças hepáticas.

É de ter em conta que aqueles valores de *missvalue* irão desaparecer no tratamento de dados.

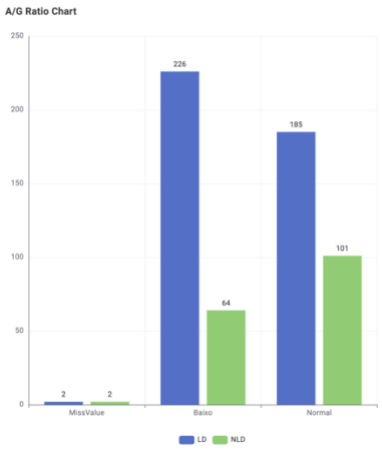

Figura 16 - Distribuição do *selector* tendo em conta o AG_ratio

1.2.13. BlLmg

As métricas do BlLmg são as seguintes:

BIMG *String*	Média *Number (double)*	Máximo *Number (double)*	Mínimo *Number (double)*
LD	0.241	4.386	0.023
NLD	0.066	0.427	0.029

Figura 17 - Métricas do *BlLmg*

Este foi o único gráfico que ficou um pouco disforme, pois não conseguimos uma forma concreta de criar *Bins* por serem valores muito pequenos e estarem sempre a oscilar. Entretanto, é importante destacar que os valores para os pacientes doentes são ligeiramente mais elevados em comparação com os valores dos não doentes. Isso sugere que, assim como outros atributos, o *BlLmg* pode ser um fator crucial na compreensão dos motivos subjacentes ao desenvolvimento de doenças hepáticas.

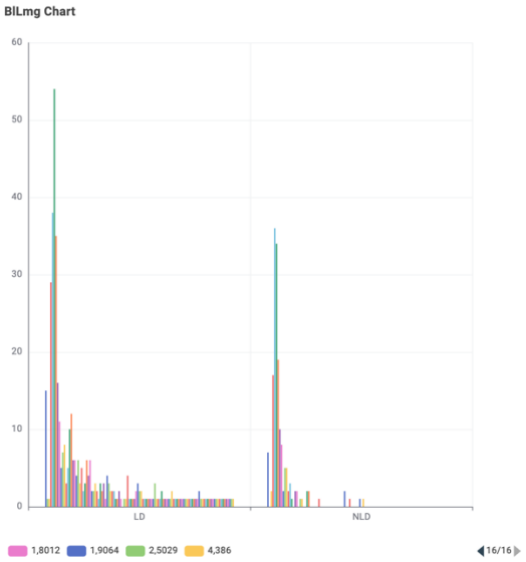

Figura 18 - Distribuição do *selector* tendo em conta o *BlLmg*

Conclusão da análise dos dados:

Embora nem todos os gráficos apresentem legendas explicativas, essa omissão decorre de sua redundância. Todos os atributos mostram uma tendência semelhante: à medida que aumentam em quantidade, a proporção de indivíduos doentes em relação aos saudáveis também aumenta. Isso sugere que, independentemente do atributo analisado - seja TB, DB, Proteínas Totais, ALB, entre outros - um aumento em sua quantidade está associado a um maior risco de doenças hepáticas.

As incertezas e ressalvas deixadas nesta seção do relatório serão esclarecidas ao longo do processo de resolução do problema.

1.3. Preparação de dados

Após uma análise inicial do conjunto de dados, identificamos a necessidade de avaliar a relevância de todas as colunas ou atributos para a resolução do problema em questão. Após extensa pesquisa sobre o tema e uma análise detalhada da base de dados, concluímos que alguns atributos seriam dispensáveis para o nosso objetivo.

O primeiro atributo eliminado do conjunto de dados foi o *id_code*, uma vez que o KNIME já fornece uma coluna com o mesmo propósito, o *RowID*. Dessa forma, manter duas colunas com informações idênticas não é ideal.

Além disso, observamos que três colunas continham informações redundantes: *Age*, *birth_year* e *birth_date*. Uma vez que a idade, o ano de nascimento e a data de nascimento são equivalentes em termos de informação, decidimos que era mais prático e eficiente trabalhar com a faixa etária da pessoa. No entanto, antes de removermos essas colunas, verificamos se todas continham as mesmas informações, evitando qualquer perda de dados potencialmente úteis. Após essa verificação, confirmamos que as informações eram exatamente as mesmas em todas as três colunas. Como tal, removemos as colunas *birth_year* e *birth_date* da nosso *dataset*.

Durante a análise das colunas, duas delas chamaram a nossa atenção. A primeira foi a coluna do *CHOL*, onde todos os valores estavam zerados e depois de verificar usado o *rank correlation*, removemo-la. No entanto, houve outra coluna que nos deixou em dúvida quanto à sua relevância para o *dataset*. Embora, esta coluna contenha valores, estes não têm uma interpretação viável e após uma análise mais aprofundada, concluímos que a manteríamos, por enquanto, pois apesar de a sua correlação não dar próximo de 1 ou –1, dá um valor suficientemente alto para que não arrisquemos a haver perda de informação que poderá ser fulcral para a resolução do problema.

Depois de filtrados os atributos passamos à parte onde verificamos os dados que temos, para isso fomos coluna a coluna.

- *age*: não havia qualquer problema quanto aos dados desse atributo. Contudo, para uma melhor análise dos dados decidimos criar 5 *bins*. Os intervalos de idade escolhidos e os respetivos nomes foram:
    -]0,9] - Criança;
    -]9,18] - Adolescente;
    -]18,35] - Jovem;
    -]35, 60] - Adulto;
    -]60,+[- Idoso.

Desta forma tornar-se-á mais fácil quando quisermos associar este atributo com qualquer outro atributo. Para não haver perda de informação criamos outra coluna à qual designamos *Faixa_Etária*.

- *gender*: neste atributo identificamos algumas incongruências, uma vez que havia cinco tipos diferentes de gênero: *"male", "female",* "*Masculine", "Male"* e *"Female".* Para normalizar esses dados, optamos por definir apenas dois tipos de gênero. Utilizando o nodo *Rule Engine*, consolidamos os dados, reduzindo os cinco tipos para apenas dois: *"Male"* e *"Female".*

- *TB*: não havia qualquer problema quanto aos dados deste atributo. A única coisa que teve de ser alterada foi o seu tipo de *String* para *Number (double);*
- *DB*: identificamos algumas incongruências, especificamente três casos em que o valor de *DB* era maior que o valor de *TB*. Isso não é possível, já que *DB (Direct Bilirubin)* somado a *IB (Indirect Bilirubin)* é igual a *TB (Total Bilirubin),* no máximo, *DB* seria igual a *TB*. Para resolver esse problema, optamos por remover as linhas que continham essas inconsistências, mesmo que isso resultasse na perda de dados de outras colunas. Além disso, assim como em *TB*, alteramos o tipo de dados de *DB* de *String* para *Number (double)*.
- *Alkphos* e *ALB*: não havia qualquer problema quanto aos dados desse atributo. Tal como outros atributos foi de *String* para *Number (double)*.
- *SGPT* e *SGOT*: não havia qualquer problema quanto aos dados desse atributo. Tal como outros atributos foi de *String* para *Number (integer)*.
- *TB(#1):* que representa o número total de proteínas teve a sua coluna com o nome reescrito não só para ser mais intuitivo, mas também para não haver sobreposição de nomes, o nome da coluna passou a ser *Total_Proteins.* Tal como outros atributos foi de *String* para *Number (double)*.
- *AG_ratio*: continha *MissValues*, as linhas com *MissValues* foram retiradas, pois eram apenas quatro. Para concluir passamos de de *String* para *Number (double)* o tipo da coluna.
- *BlLmg*: continha, também, *MissValues*, contudo neste caso já era em 22 linhas e como tal era muita perda de informação e depois de analisar percebemos que a melhor saída para o tratamento deste problema seria usar a média, pois como todos os valores são muito pequenos e estão todos muito "comprimidos" num intervalo muito pequeno não haveria mudanças significativas no resultado alcançado.

Para concluir a preparação dos dados para nos facilitar a análise e comparação de dados decidimos criar *bins* para os atributos científicos com o intuito do *bar chart* desconforme:

- *TB_bin*:
  -]0, 0.3] - Baixa;
  -]0.3, 1.2] - Normal;
  -]1.2, 3] - Alta;
  -]3,+[- Extremamente Alta.
- *DB_bin*:
  -]0, 0.3] - Baixa;
  -]0.3, 1] - Normal;
  -]1, 2] - Alta;
  -]2,+[- Extremamente Alta.
- *Alkphos_bin*:
  -]0,20] - Baixa;
  -]20,200] - Normal;
  -]200,500] - Alta;
  -]500,+[- Extremamente Alto.
- *Sgpt_bin***:**
  -]0,40] - Normal;
  -]40,120] - Alta;
  -]120,500] - Extremamente Alto;
  -]500+[- Perigoso.
- *Sgot_bin*:
  -]0,40] - Normal;

    -]40,150] - Alta;
    -]150,500] - Extremamente Alto;
    -]500,+[- Perigoso.

- *Total_Proteins_bin*:
    -]0, 3.5] - Baixa;
    -]3.5, 5] - Normal;
    -]5,+[- Alta.
- ALB:
    -]0, 3.4] – Baixa;
    -]3.4, 5] – Normal;
    -]5+[- Alta.
- *AG_Ratio_bin*:
    -]0, 1] - Baixa;
    -]1, 2] - Normal;
    -]2,+[- Alta.

1.4. Modelação

Para este dataset dividimos a modelação em 6 maneira diferentes:

1. Modelação com árvores de decisão sem *Bins;*
2. Modelação com árvores de decisão com *Bins;*
3. Modelação com uso de Regressão;
4. Modelação com uso de *Clustering;*
5. Modelação com uso de *Feature Selection;*
6. Modelação com uso de Redes Neurais.

1.4.1. Modelação com árvores de decisão sem Bins

O primeiro dos modelos que decidimos implementar foi algo simples onde se usaram as informações que saíram da fase *Tratamento de dados.*

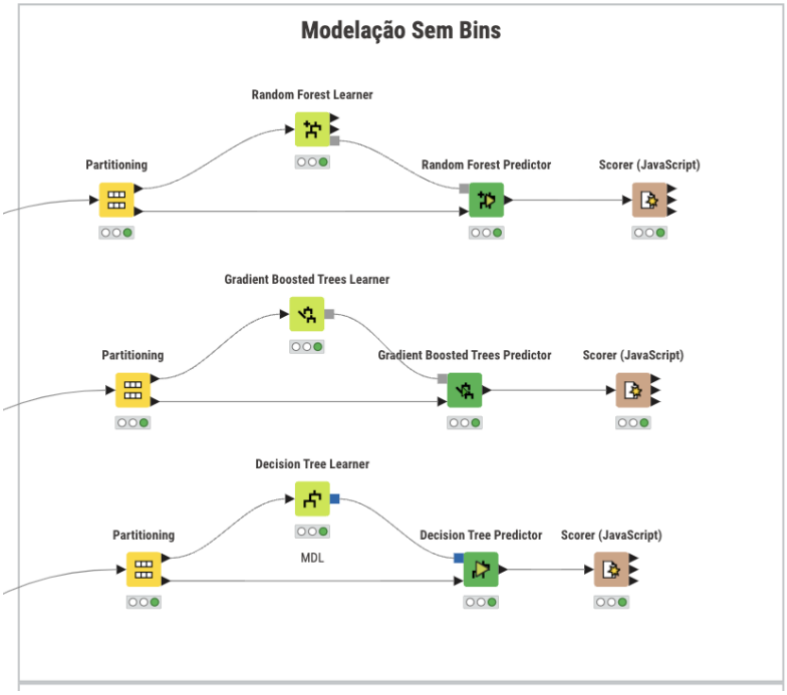

Figura 19 - Modelação com árvores de decisão sem Bin

O *dataset* PAR é um problema de classificação, onde o atributo alvo assume uma de duas classes: "LD" ou "NLD".

Para abordar este tipo de problema, optamos por utilizar os nós ***Random Forest Learner***, ***Gradient Boosted Trees Learner*** e ***Decision Tree Learner***. Combinamos esses nós com a técnica de p*artitioning* usando *stratified sampling*, sem o uso de uma *seed* aleatória. A escolha pelo *partioning* em vez do x-*partioner* foi motivada pelo tamanho reduzido do conjunto de dados, pois o *partioning* oferece maior robustez e precisão em nossas previsões. Esta decisão foi confirmada empiricamente, uma vez que observamos resultados inferiores ao usar o *x-partioner.* Além disso, optamos por não usar uma *seed* aleatória em nossa configuração para evitar possíveis variações negativas nos resultados.

Statistics

Confusion Matrix

	LD (Predicted)	NLD (Predicted)	
LD (Actual)	83	0	100.00%
NLD (Actual)	31	2	6.06%
	72.81%	100.00%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
73.28%	26.72%	0.085	85	31

Figura 20 - Melhor resultado deste modelo (Decision Tree Learner)

O nó que deu melhor *accuracy* foi o ***Decision Tree Learner,*** apesar de ser apesar de 73.28%, contudo os outros deram valores inferiores a 70%. Isto deve-se ao facto de haver um desequilíbrio muito grande na quantidade de doentes e de não doentes no nosso *dataset*.

1.4.2. Modelação com árvores de decisão com Bins

Após o mau desempenho das árvores de decisão decidimos tentar aplicar o mesmo método, só que usando *Bins* para ver se conseguiríamos melhorar de certa forma a *accuracy.*

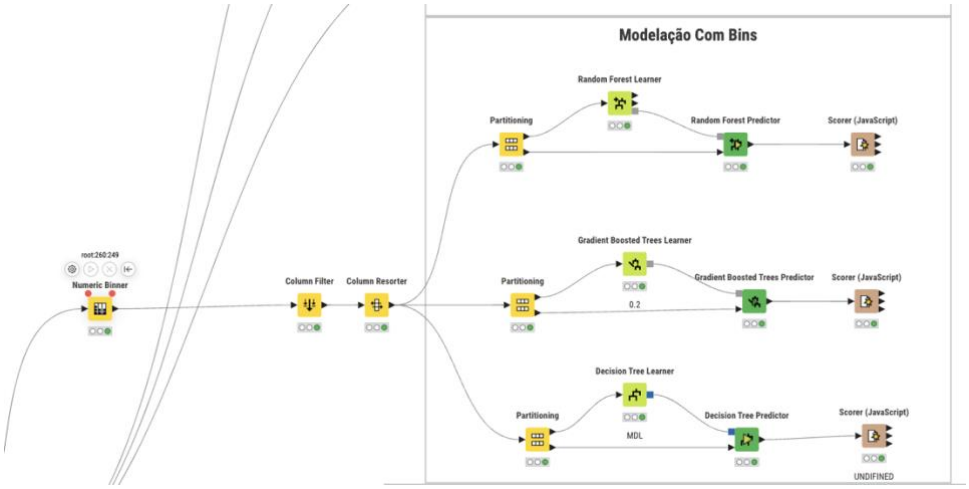

Figura 21 - Modelação com árvores de decisão com Bin

Os *bins* criados são exatamente os mesmos usados para quando da análise de dados e todas as características principais do modelo também se mantiveram.

Scorer View

Confusion Matrix

	LD (Predicted)	NLD (Predicted)	
LD (Actual)	70	13	84.34%
NLD (Actual)	19	14	42.42%
	78.65%	51.85%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
72.41%	27.59%	0.283	84	32

Figura 22 - Melhor resultado deste modelo (Random Forest Learner)

Houve melhorias substâncias na análise que usa o ***Random Forest Learner*****,** contudo usando o modelo anterior o valor diminui. No geral podemos dizer que houve melhorias, pois os três modelos ficaram com taxas de *accuracy* acima dos 70%, contudo o objetivo é conseguir a maior percentagem possível, logo esta abordagem não é a ideal.

Um dos motivos que levou o valor da *accuracy* a diminuir com o uso de intervalos é a perda de informação, devido à ambiguidade dos dados. Esta ambiguidade gerada pela criação de bins não afetou em nada o ***Random Forest Learner*** devido à sua natureza de aleatoriedade aquando do treino da árvore.

1.4.3. Modelação com uso de Regressão

Como estávamos a ter resultados não satisfatórios, decidimos tentar aplicar regressão, mesmo não sendo a melhor das opções, pois fomos obrigados a converter "LD" e "NLD" para números, 1 e 2, respetivamente.

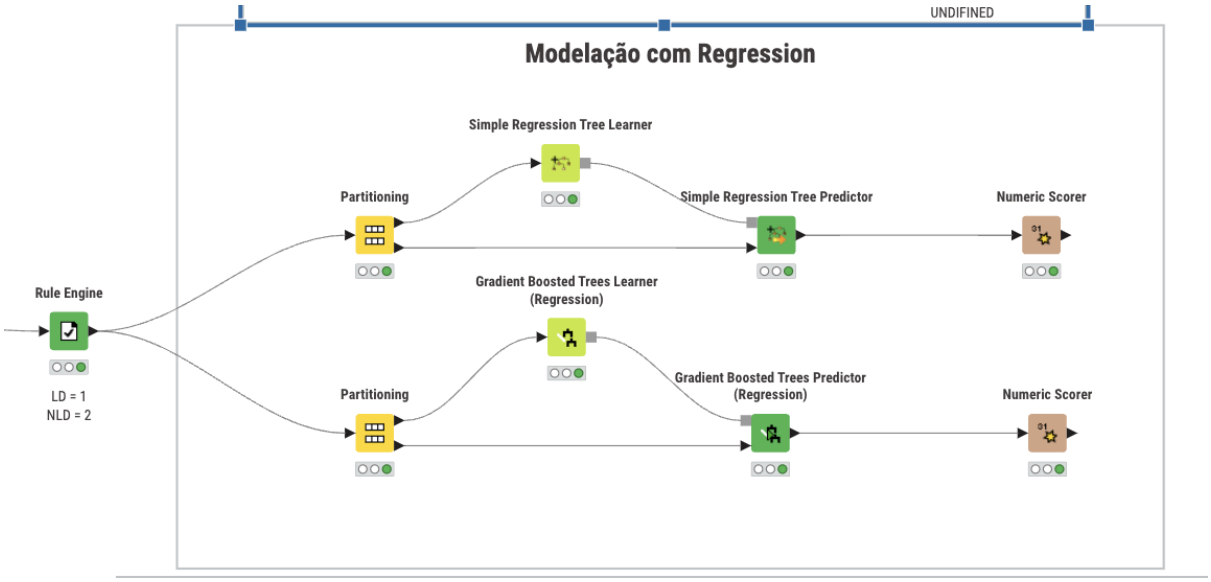

Figura 23 - Modelação com uso de Regressão

Tal como nos outros modelos usamos o *Partitioning*, pois, para o nosso *dataset*, tem um melhor desempenho na maior parte dos casos.

Statistics - 3:260:258 - Nu...

File	
R^2:	-0,461
Mean absolute error:	0,304
Mean squared error:	0,297
Root mean squared error:	0,545
Mean signed difference:	0,06
Mean absolute percentage error:	0,243
Adjusted R^2:	-0,461

Statistics - 3:260:254 - Nu...

File	
R^2:	-0,525
Mean absolute error:	0,31
Mean squared error:	0,31
Root mean squared error:	0,557
Mean signed difference:	0,034
Mean absolute percentage error:	0,241
Adjusted R^2:	-0,525

Figura 24 - Resultados da Regressão

Como podemos ver o coeficiente de determinação (R^2) é negativo em ambas as análises o que confirma a suspeita inicial de que não é uma boa ideia aplicar regressão nesta situação.

1.4.4. Modelação com uso de Clustering

Visto que não estávamos a conseguir achar uma solução satisfatória continuamos a aplicar os nossos conhecimentos de maneira a tentar o máximo possível conseguir chegar a um resultado, pelo menos, acima de 80%. O próximo modelo é de *clustering.*

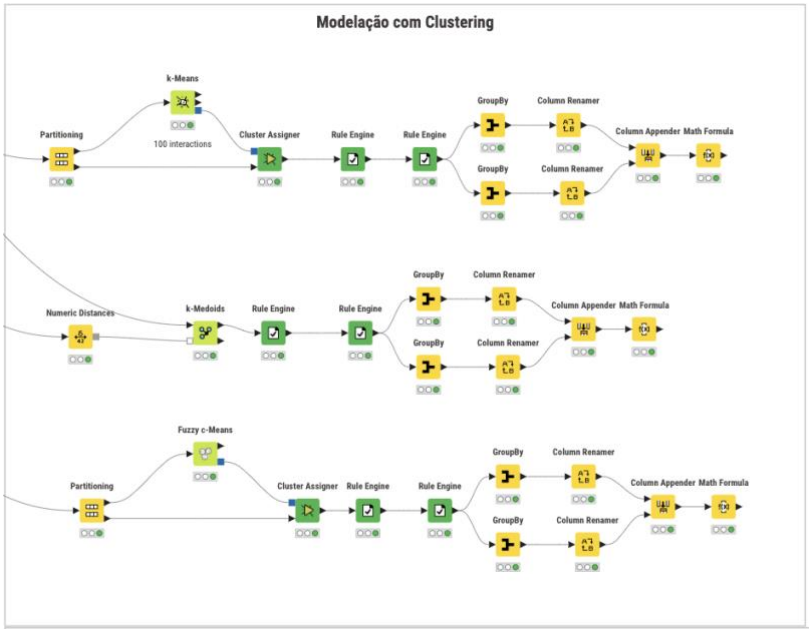

Figura 25 - Modelação com uso de Clustering

Aplicamos três diferentes tipos de *clustering*, depois de ser criado uma coluna com os clusters (como no caso do nodo ***K-Means*** e ***Fuzzy c-Means***) ou com Rows (como no caso do nodo ***K-medoids***). Depois vamos substituir os nomes pela respetiva classe “LD” ou “NLD”, após isso fazemos a média para ver qual a percentagem de acertos.

#	RowID	Checkados *Number (integer)*	Total *Number (integer)*	Result *Number (double)*
1	Row0	82	116	0.707

Figura 26 - Melhor resultado de Clustering (Nodo – K-Means)

Como podemos ver através da imagem o melhor desempenho foi do nodo *K-means* com um acerto de 70.7% não sendo suficientemente alto para ser uma opção viável, os outros dois nodos ficaram muito aquém, não chegando nem a 60%.

1.4.5. Modelação com Feature Selection

Depois de muito pensar, pensamos que poderia haver atributos que pudessem estar a atrasar o desempenho do nosso *dataset*. Com tal ideia em mente decidimos implementar um modelo parecido igual ao primeiro ponto, contudo fazendo o filtro dos atributos previamente com *Feature Selection.*

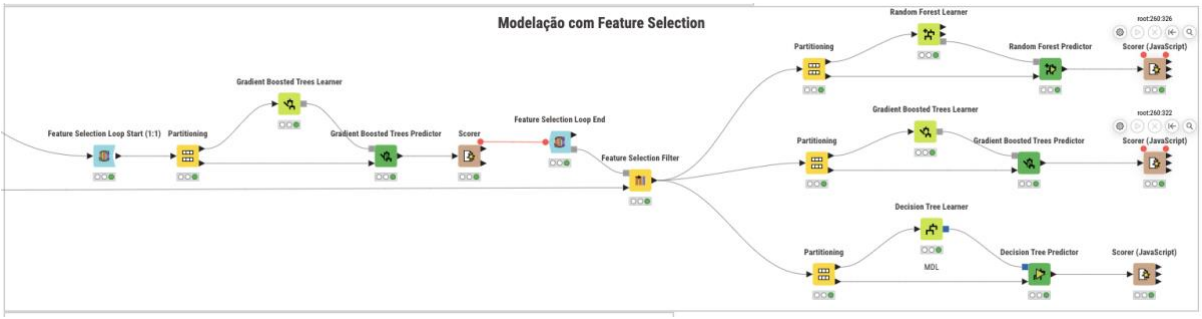

Figura 27 - Modelação com Feature Selection

Primeiro, fazemos um *loop* e quando esse *loop* acabar o nodo **Feature Selection Filter** terá selecionado quais os melhores atributos para conseguir uma melhor *accuracy*. Depois, passamos esses atributos às árvores de decisão.

Scorer View

Confusion Matrix

	LD (Predicted)	NLD (Predicted)	
LD (Actual)	72	11	86.75%
NLD (Actual)	18	15	45.45%
	80.00%	57.69%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
75.00%	25.00%	0.344	87	29

Figura 28 - Melhor resultado do uso do Feature Selection(Gradient Boosted Trees Learner)

No primeiro modelo, sem *bins*, o nó com maior *accuracy* foi o ***Decision Tree Learner***, no segundo modelo, com *bins*, o nó com maior *accuracy* foi o ***Random Forest Learner***, porém, agora, a maior percetagem já conseguida no accuracy foi de 75% e foi com o nó ***Gradient Boosted Trees Learner.***

Algo que suscitou ainda mais curiosidade foi o seguinte facto:

A mudança nos atributos selecionados pelo *Feature Selection Filter*, ao variar a *seed* estática do nó ***Feature Selection Loop Start***, indica uma sensibilidade significativa do conjunto de dados a pequenas variações. Essa instabilidade levanta questões sobre a viabilidade do conjunto de dados para análise. Se até mesmo pequenas alterações na *seed* podem causar mudanças drásticas nos atributos selecionados, isso sugere que o *dataset* pode não ser robusto o suficiente para produzir resultados consistentes e confiáveis. Essa observação destaca a importância de avaliar cuidadosamente a qualidade dos dados e considerar possíveis fontes de instabilidade ao realizar análises e construir modelos a partir desse conjunto de dados.

1.4.6. Modelação com Redes Neuronais Artificiais

Por último, decidimos testar redes neurais para ver qual seria o resultado quando comparado com o valor mais alto até agora, 75%.

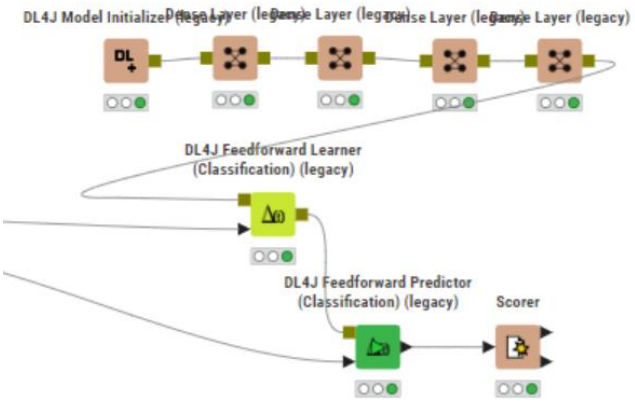

Figura 29 - Modelação com Redes Neurais

Apesar de obtermos este resultado, este pode não ser o melhor, dado que a configurações dos nodos de Redes Neuronais Artificiais contribui muito para o resultado final.

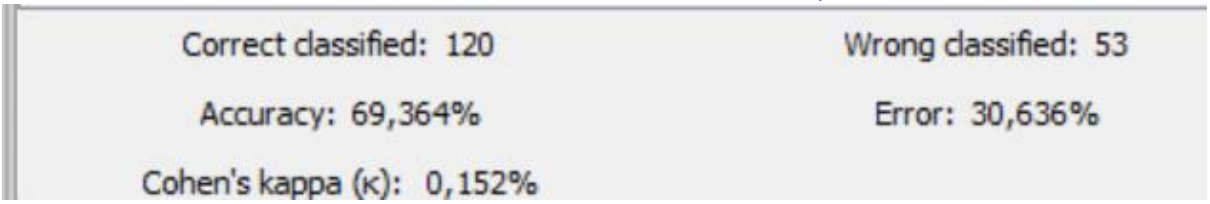

Figura 30 - Resultado do uso das Redes Neurais

Como podemos verificar um bom resultado, mas não superior ao anterior. Logo, esta não é uma abordagem que queremos.

1.5. Avaliação do Dataset dado

Como já referimos acima, não conseguimos obter um resultado satisfatório mesmo aplicando diferente método de análise. Outro ponto importante a referir são os *outliers*. Se removermos ou alterarmos os *outliers* deste *dataset* os resultados que temos agora, vão ficar piores, pois corrompe muitas das linhas da nossa tabela, por tal motivo decidimos não mexer nos *outliers*, mesmo que não seja o mais correto a ser feito.

2. Dataset escolhido

Nesta etapa do projeto, o grupo foi encarregue de selecionar um *dataset*, analisá-los e, com base nas informações coletadas, resolver o problema proposto. Optamos pelo conjunto de dados "*The Boston Housing Dataset".* O *dataset* contém informação dos Censos americanos sobre a habitação na área de *Boston*.

O *dataset* contém 507 linhas e 14 colunas, sendo os atributos os seguintes:

- CRIM – Taxa de crime per capita da cidade;
- ZN - Proporção de terrenos residenciais zoneados para lotes com mais de 25 000 pés quadrados (7 600 metros quadrados);
- INDUS - Proporção de acres não usado para uso comercial da cidade;
- CHAS – Indica se a habitação segue o rio, 1 se sim 0 se não;
- NOX - Concentração de óxidos nítricos;
- RM - Número médio de quartos por habitação;
- AGE - Proporção de habitações habitáveis construídas antes de 1940;
- DIS – Distâncias ponderadas para cinco centros de emprego de *Boston*;
- RAD – Index de acessibilidade a autoestradas;
- TAX – Taxa de imposto sobre a propriedade de valor total por 10 000 $;
- PTRATIO - Proporção de aluno-professor da cidade;
- B – $1000 * (Bk - 0{,}63)^2$ onde Bk é a proporção de população negra da cidade;
- LSTAT – Percentagem de população de classe baixa;
- MEDV – Valor médio das casas ocupados pelos proprietários em milhares de dólares.

O atributo MEDV é o nosso target.

2.1. Estudo de Negócio

O objetivo deste problema é criar um modelo para prever o valor mediano das casas ocupadas pelos proprietários (em milhares de dólares americanos) com base nas características fornecidas. Para alcançar isto vamos precisar de um dataset e um programa para trabalhar sobre este, neste caso é o KNIME o programa escolhido.

2.2. Análise dos dados

2.2.1. MEDV

O nosso target, MEDV, flutua entre 5 e 50, como podemos ver na figura abaixo, a maioria das habitações possui um MEDV menor que 25. O gráfico usa *bins* para melhor compreensão dos dados.

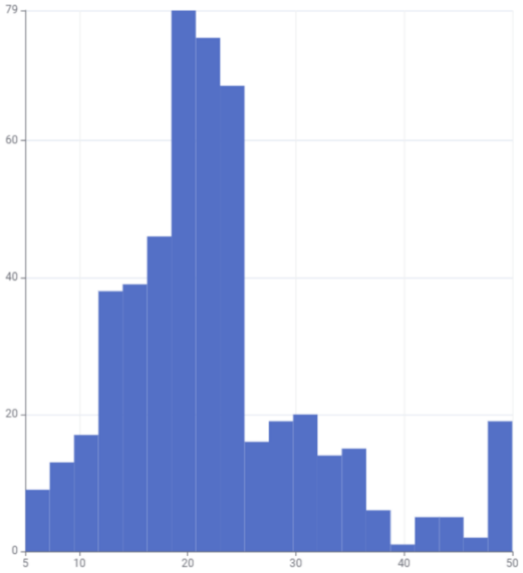

Figura 31 - Histograma referente a MEDV

2.2.2. LSTAT

A percentagem de população de classe baixa também flutua bastante, variando de 1,73 até 37,97. É observado uma relação entre LSTAT e MEDV, quando LSTAT é menor, maior o valor de MEDV.

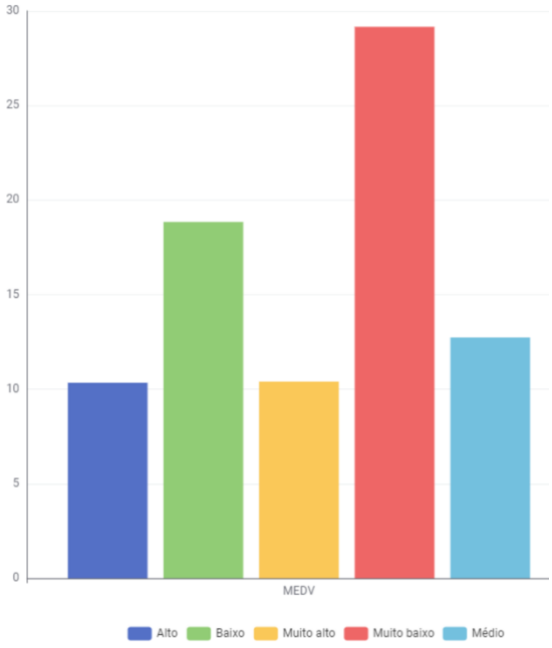

Figura 32 - Gráfico de barras, média MEDV de cada bin de LSTAT

2.2.3. B

O histograma do valor de B revela uma concentração de linhas com o valor de B à volta de 380 unidades, a relação entre B e MEDV é o contrário da relação de LSTAT e MEDV, quando B aumenta, MEDV tende a aumentar.

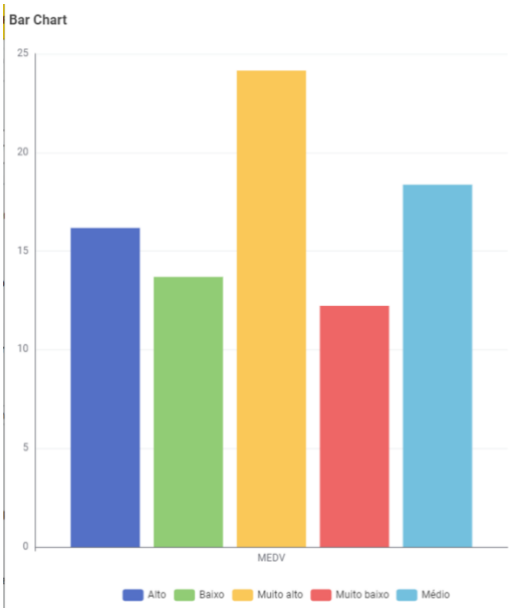

Figura 33 - Gráfico de barras, média MEDV de cada bin de B

2.2.4. PTRATIO

Os valores de PTRATIO são mais bem distribuídos, sendo o seu mínimo 12,6 e o seu máximo 22. Quando PTRATIO aumenta MEDV tende a ser menor.

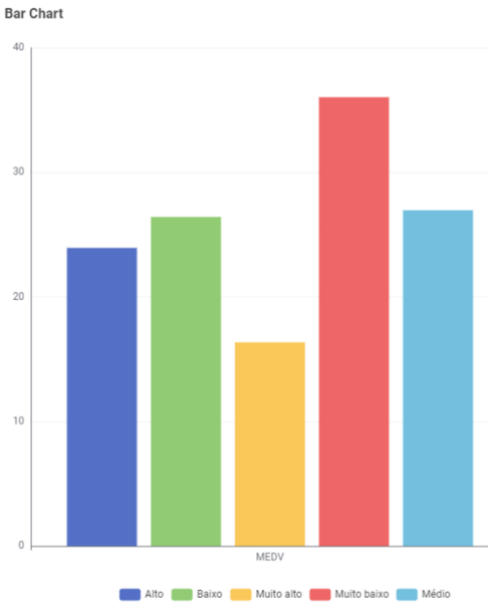

Figura 34 - Gráfico de barras, média MEDV de cada bin de PTARTIO

2.2.5. TAX

A taxa de imposto tem um grande intervalo de dispersão, apesar de ser bem distribuída desde 187-450, há muitos poucos valores de 450-650, e depois uma grande concentração entre 650-700.

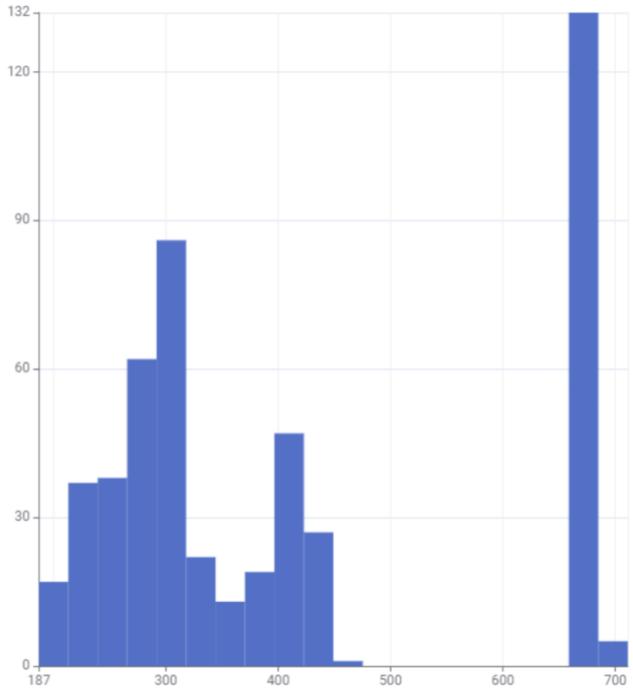

Figura 35 - Histograma referente a TAX

2.2.6. RAD

Este valor varia entre 1 e 24, tendo uma grande concentração no polo superior e inferior e pouca distribuição nos valores intermédios.

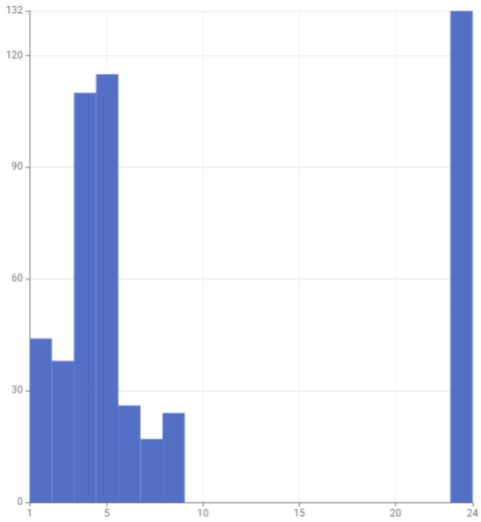

Figura 36 - Histograma referente a RAD

2.2.7. CHAS

A relação entre CHAS e MEDV é ambígua, não dá para deduzir qualquer informação. Há também um maior número de habitações que não estão perto do rio.

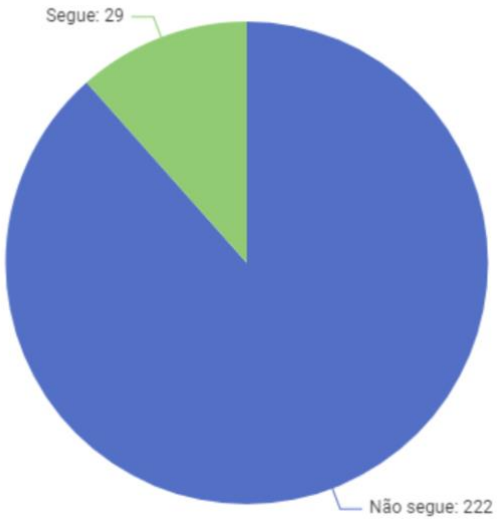

Figura 37 - Gráfico de fatias de CHAS

2.2.8. DIS

Este valor varia entre 1,129 e 12,126, através do histograma, notamos uma melhor distribuição comparada com os outros valores, mesmo assim existe uma maior concentração em valores menores.

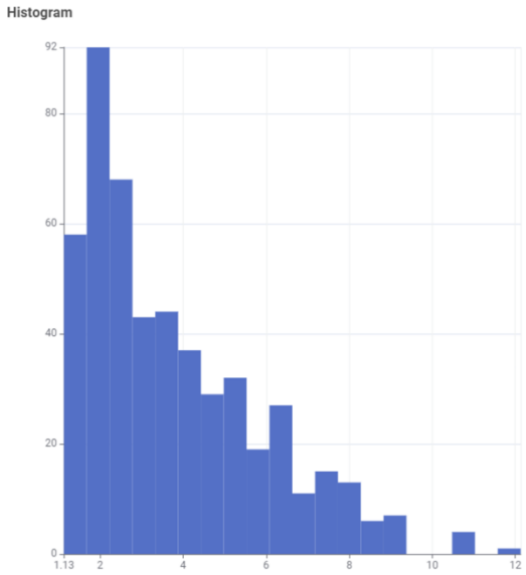

Figura 38 - Histograma referente a DIS

2.2.9. AGE

Esta proporção varia entre 2.90 e 100, e quanto maior esta for, MEDV tende a ser menor, ou seja, áreas com habitações mais recentes são mais valiosas, o que é de esperar, mas confirmado pelos dados.

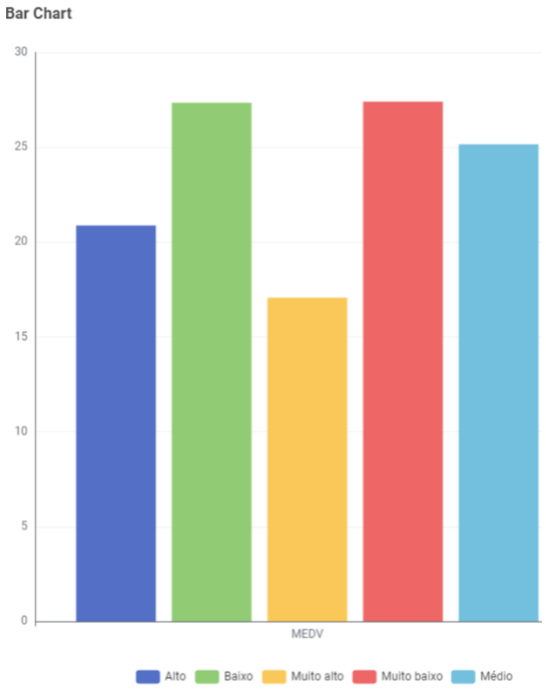

Figura 39 - Gráfico de barras, média MEDV de cada bin de AGE

2.2.10. RM

O valor de RM varia de 3,56 a 8,78 e quando este aumenta, MEDV tende a aumentar.

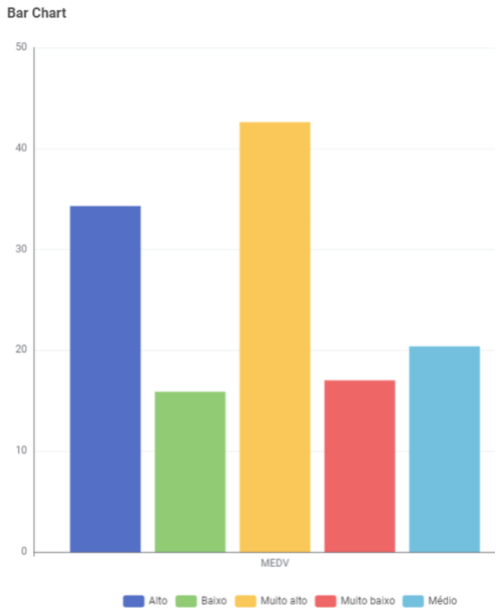

Figura 40 - Gráfico de barras, média MEDV de cada bin de RM

2.2.11. NOX

NOX varia entre 0,39 e 0,87, tendo um intervalo sem ocorrências (0,75-0,84).

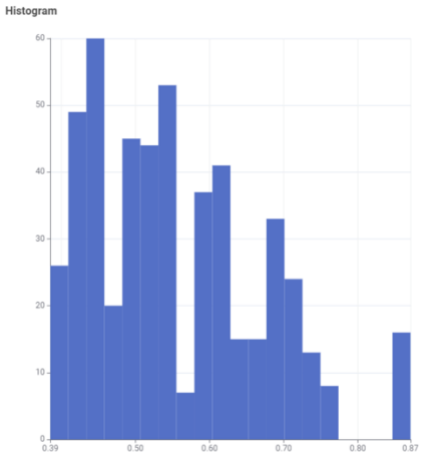

Figura 41 - Histograma referente a NOX

2.2.12. INDUS

INDUS tem como valor mínimo 0,46 e 27,74 como máximo, sendo a maior concentração de valores entre 16,8 e 18,2.

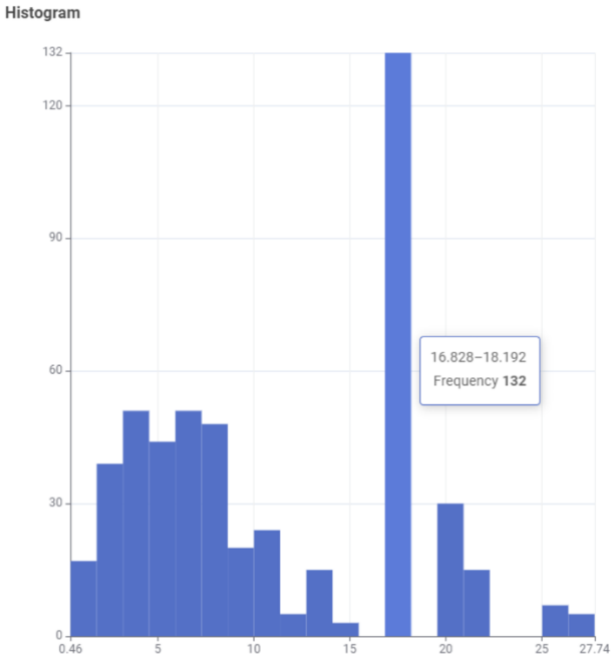

Figura 42 - Histograma referente a INDUS

2.2.13. ZN

ZN tem muitos valores concentrados entre 0-5, o seu valor máximo é 100, mas só o atinge uma vez. Podemos então concluir que a proporção de terrenos residenciais zoneados para lotes com mais de 25 000 pés quadrados é relativamente baixa nas entradas do dataset, havendo exceções.

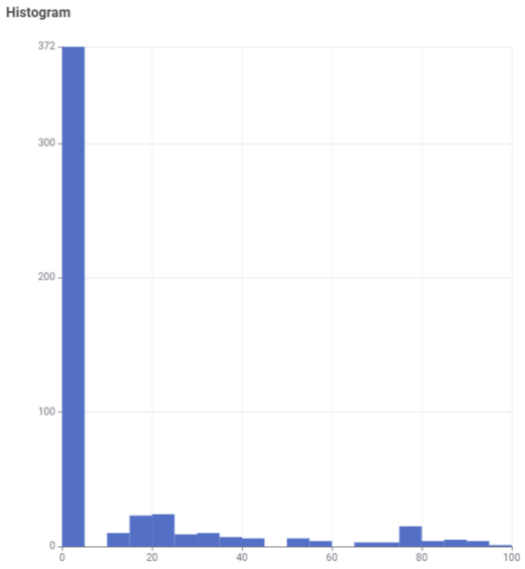

Figura 43 - Histograma referente a ZN

2.2.14. CRIM

Apesar de variar entre 0,01 e 88,98 CRIM tem uma alta concentração em 0,01-4,5

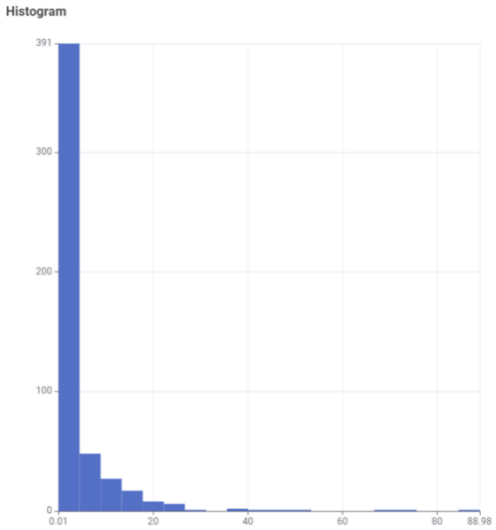

Figura 44 - Histograma referente a CRIM

2.3. Preparação de dados

O *dataset* escolhido continha 2 entradas, com *Missing value* no nosso *target*, *MEDV*, então decidimos remover as linhas completamente, já que era um número pequeno de *missing values* e o atributo em falta era o *target* não valia a pena tentar aproveitar esta entrada.

Com o node "*box plot*" conseguimos também descobrir que a maioria das colunas continha *outliers*, então tivemos de fazer o tratamento destes. Uma vez que era um grande número de *outliers,* decidimos usar o valor mais perto permitido como forma de substituição destes.

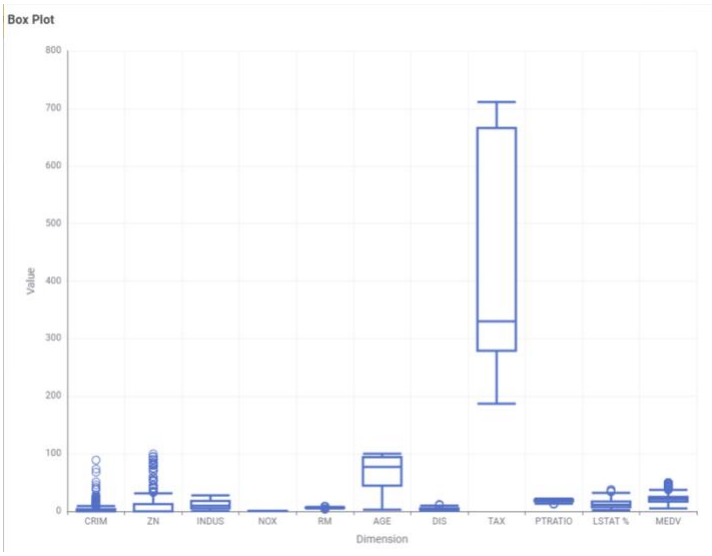

Figura 45 - Box Plot

Observamos com o node "*rank correlation*" a correlação entre as colunas, e decidimos remover 2 delas, sendo estas *B* e *CHAS*, uma vez que o valor de correlação era muito perto de 0.

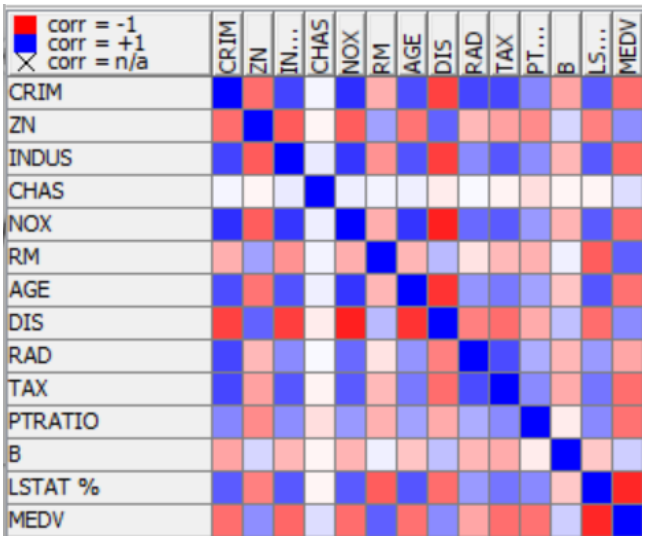

Figura 46 - Rank Correlation das colunas

2.4. Modelação

2.4.1. Modelação com Tree e Forest Learners

Para realizar esta modelação usamos os nodos, Simple Regression Tree Learner, Gradient Boosted Trees Learner (Regression) e Random Forest Learner (Regression) e os respetivos nodos de predição.

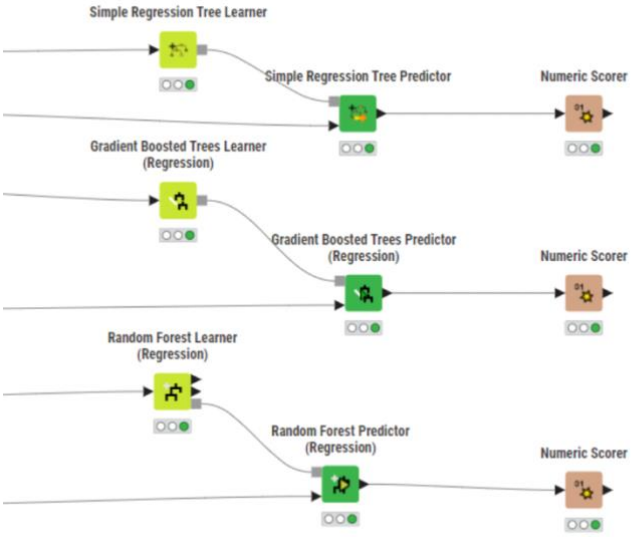

Figura 47 - Modelação com Tree e Forest Learnerers

É de notar que para obter os resultados observados na figura, foi necessário utilizar o nodo Normalizer, isto fez com que o erro do Previsor fosse muito menor. Concluímos então que o nodo com maior precisão dos três utilizados foi o Random Forest Learner (Regression), mostrando valores próximos dos valores obtidos pelo Gradient Boosted Trees Learner (Regression).

File	
R^2:	0,649
Mean absolute error:	0,073
Mean squared error:	0,014
Root mean squared error:	0,117
Mean signed difference:	0,011
Mean absolute percentage error:	0,285
Adjusted R^2:	0,649

File	
R^2:	0,835
Mean absolute error:	0,056
Mean squared error:	0,006
Root mean squared error:	0,08
Mean signed difference:	-0
Mean absolute percentage error:	0,218
Adjusted R^2:	0,835

File	
R^2:	0,871
Mean absolute error:	0,055
Mean squared error:	0,005
Root mean squared error:	0,071
Mean signed difference:	-0,001
Mean absolute percentage error:	0,235
Adjusted R^2:	0,871

Figura 48 - Simple Regression Tree Learner VS Gradient Boosted Trees Learner (Regression) VS Random Forest Learner (Regression)

2.4.2. Modelação com Regressão linear e polinomial

Neste passo usamos os nodos Linear Regression Leaner e Polynomial Regression Learner, no nodo polinomial foi usado o grau 2, uma vez que foi este que demonstrou melhores resultados.

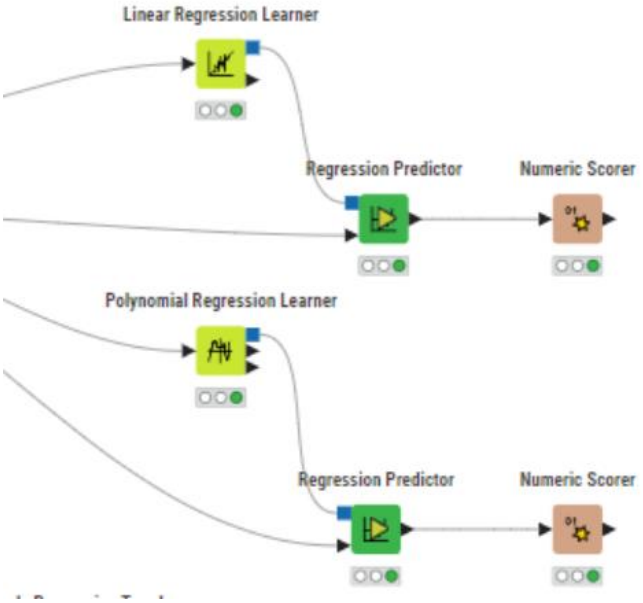

Figura 49 - Modelação com Regressão Linear e Polinomial

Como foi realizado na modelação anterior, os dados foram normalizados, para obtenção de melhores resultados. É facilmente observado que o nodo Polynomial Regression Learner com grau 2, obteve melhores valores, apesar de ser esperado.

File		File	
R^2:	0,752	R^2:	0,835
Mean absolute error:	0,075	Mean absolute error:	0,058
Mean squared error:	0,01	Mean squared error:	0,006
Root mean squared error:	0,099	Root mean squared error:	0,08
Mean signed difference:	0,014	Mean signed difference:	0,015
Mean absolute percentage error:	0,295	Mean absolute percentage error:	0,264
Adjusted R^2:	0,752	Adjusted R^2:	0,835

Figura 50 - Linear Regression Leaner VS Polynomial Regression Learner

2.4.3. Modelação com Redes Neuronais Artificiais

Para a modelação com redes neuronais artificiais usamos os nodos DL4J Feedforward Learner e RProp MLP Learner, da extensão DL4. Foi desafiador encontrar a configuração mais otimizada nesta etapa, uma vez que testar cada configuração demorava muito tempo. É de notar que foi preciso normalizar os valores das colunas para executar estes algoritmos.

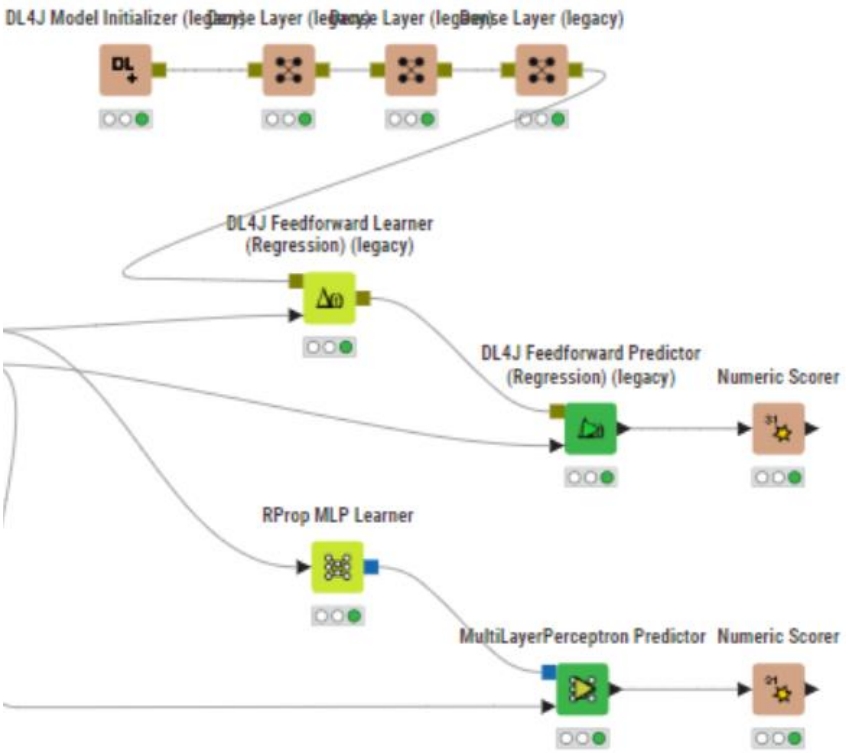

Figura 51 - Modelação com Redes Neuronais Artificiais

O nodo RProp MLP Learner obteve os melhores resultados comparado ao FeedForward Learner que teve uma média de erro bastante elevada. É de notar que estes valores podem mudar muito consoante a configuração dos nodos, esta foi as melhores combinações que encontramos e os seus resultados.

File		File	
R²:	0,585	R²:	0,821
Mean absolute error:	0,103	Mean absolute error:	0,058
Mean squared error:	0,016	Mean squared error:	0,007
Root mean squared error:	0,128	Root mean squared error:	0,084
Mean signed difference:	-0,08	Mean signed difference:	0,005
Mean absolute percentage error:	0,339	Mean absolute percentage error:	0,268
Adjusted R²:	0,585	Adjusted R²:	0,821

Figura 52 - DL4J Feedforward Learner VS RProp MLP Learner

2.5. Avaliação do Dataset escolhido

Após tratar dos outliers, das entradas com missing values, e de testar vários algoritmos, chegamos ao que melhor para o nosso problema. Este é o Random Forest Learner (Regression), pois é aquele que contém a menor média de erro e maior R^2. No entanto, também usamos algoritmos que se destacaram e ficaram pertos do resultado do referido.

Reconhecemos que devíamos de ter dedicado mais tempo aos nodos de redes neuronais artificiais, uma vez que estes demoram muito tempo a serem executados e os seus resultados mudam drasticamente conforme a sua configuração.

Conclusão

Com a finalização deste trabalho prático, pudemos colocar em prática uma ampla gama de conceitos relacionados ao desenvolvimento de modelos de aprendizado, abordados ao longo do semestre, bem como explorar novos conceitos não previamente discutidos.

Durante o desenvolvimento deste projeto, não apenas criamos modelos de aprendizado, mas também nos envolvemos na exploração e pré-processamento dos conjuntos de dados. Essa abordagem nos permitiu obter uma compreensão mais profunda do problema em questão e selecionar as estratégias mais adequadas para alcançar resultados satisfatórios.

Apesar dos desafios iniciais, especialmente na seleção do conjunto de dados para a Tarefa A, estamos contentes com o trabalho realizado. Conseguimos desenvolver modelos de aprendizado que atenderam às nossas expectativas para os conjuntos de dados analisados e documentamos cuidadosamente todo o processo de desenvolvimento do projeto.