



Aprendizagem e Decisões Inteligentes

LICENCIATURA EM ENGENHARIA INFORMÁTICA, 3ºANO 2ºSEMESTRE

04/05/2024



a93320

Augusto C.O. Campos



a93199

Carlos A.F.D. Silva

G27



a93182

Tiago A.F. Silva

Índice

1. Introdução	2
2. Tarefa Dataset Grupo (Car Milage)	2
2.1. Estudo de Negócio	2
2.2. Estudo de Dados	3
2.3. Preparação dos Dados	7
2.4. Modelação	9
2.4.1. Modelos de Controlo (com todos os parâmetros)	9
2.4.2. Modelos com feature selection	10
2.4.3. Modelo de Redes Neurais	12
2.5. Avaliação	14
3. Tarefa DataSet Atribuído	15
3.1. Estudo do negócio	15
3.2. Estudo dos dados	15
3.3. Preparação dos dados	22
3.4. Modelação	23
3.4.1. Objectivo Classificação	23
3.4.2. Modelação Objectivo Contínuo (Regressão)	27
3.4.3. Clustering	28
3.6. Avaliação	29
4. Conclusão	29

1. Introdução

Este relatório foi desenvolvido como parte do trabalho prático da cadeira de Aprendizagem e Decisões Inteligentes, na qual fomos proposto criar modelos de aprendizagem. Este trabalho está dividido em 2 tarefas, sendo a primeira a pesquisa, análise, exploração e preparação de um dataset escolhido pelo grupo e a segunda a análise, exploração e preparação de um dataset escolhido pelos docentes da cadeira.

2. Tarefa Dataset Grupo (Car Milage)

Nesta tarefa 1 consiste na escolha de um dataset por parte do grupo, analisá-lo e recolher o conhecimento mais importante no contexto do problema.

Após explorarmos alguns datasets de modo a encontrar aqueles que melhor se enquadravam com o que procuramos deparamo-nos com 2 dataset que exploramos de maneira mais detalhada. Após a análise de ambos, concluímos que o dataset mais adequado seria o cars_data.csv, devido ao seu diverso leque de parâmetros importantes e também à quantidade de entradas que este possui.

O dataset escolhido possui informação de veículos de diversos tipos, desde supercarros a carros citadinos. O objetivo deste problema será tentar prever o consumo de um carro e ao mesmo tempo descobrir que atributos mais afetam este valor.

A metodologia utilizada na resolução do problema é o **CRISP-DM**. Este modelo define um guião dividido em 6 etapas de maneira a desenvolver um projeto de análise de dados, no entanto só iremos fazer as primeiras 5, sendo estes:

1. Estudo do negócio;
2. Estudo dos Dados;
3. Preparação dos Dados;
4. Modelação;
5. Avaliação;
6. Desenvolvimento;

2.1. Estudo de Negócio

O objetivo deste problema é tentar prever qual será o consumo de um carro dependendo das suas características. Para tal, possuímos um dataset com diversas informações sobre diversos carros e, também o apoio da ferramenta de análise de dados e machine learning KNIME.

Os objetivos a serem cumpridos são:

1. Analisar o dataset por completo;
2. Tratar de inconsistências no dataset (missing values)
3. Criar gráficos, matrizes de correlação de maneira a visualizar os dados
4. Prever a partir de modelos de aprendizagem, o consumo de cada veículo

2.2. Estudo de Dados

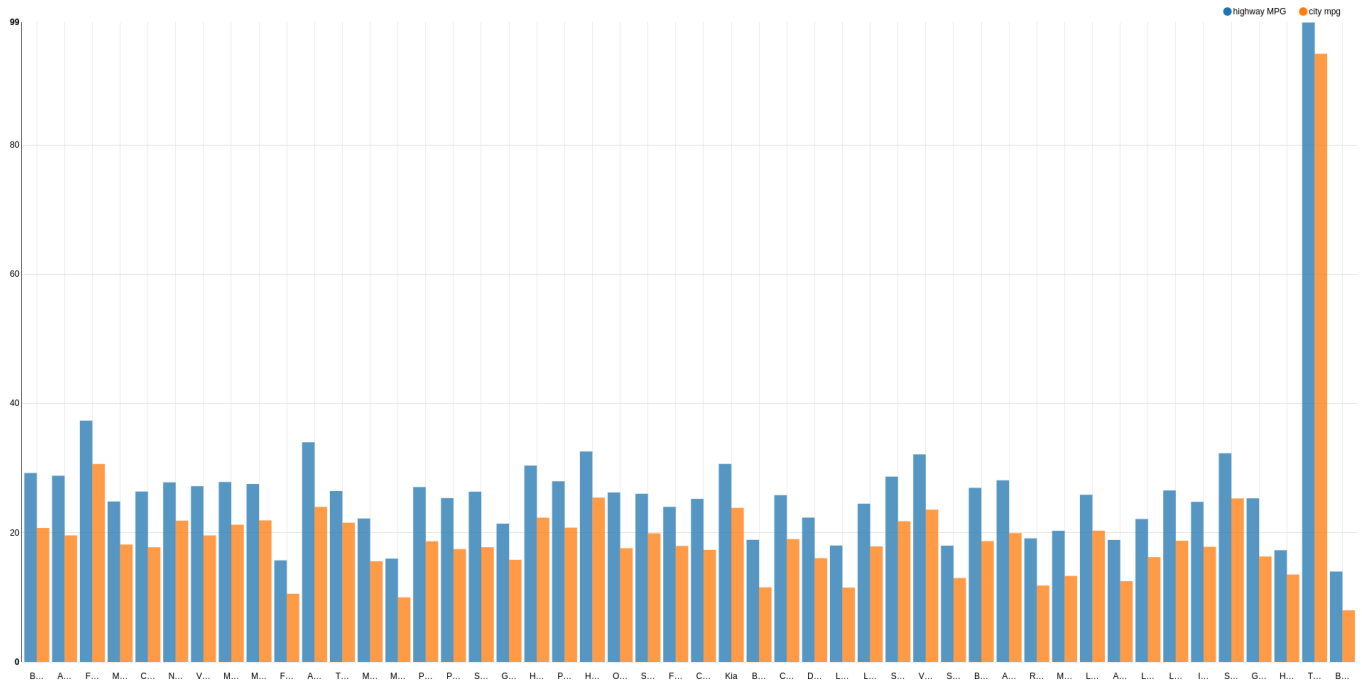
Os dados para este problema foram retirados de um repositório do github sobre [cars data analysis](#), este dataset contém 16 atributos e 11915 linhas. Sendo os atributos os seguintes:

1. **Make:** marca do carro
2. **Model:** modelo do carro
3. **Year:** ano do carro
4. **Engine Fuel Type:** tipo de combustível(incluindo elétricos)
5. **Engine HP:** potência do motor
6. **Engine Cylinders:** quantidade de cilindros do motor
7. **Transmission Type:** tipo de transmissão
8. **Driven_Wheels:** rodas com tração
9. **Number of Doors:** número de portas
10. **Market Category:** categoria do carro
11. **Vehicle Size:** tamanho do veículo
12. **Vehicle style:** estilo de veículo
13. **Highway MPG:** consumo do carro na autoestrada
14. **City MPG:** consumo do carro na cidade
15. **Popularity:** popularidade do carro
16. **MSRP:** preço atual sugerido pela marca

O atributo objetivo será uma média entre o **Highway MPG** e **City MPG** de maneira a conseguirmos um resultado de consumo global.

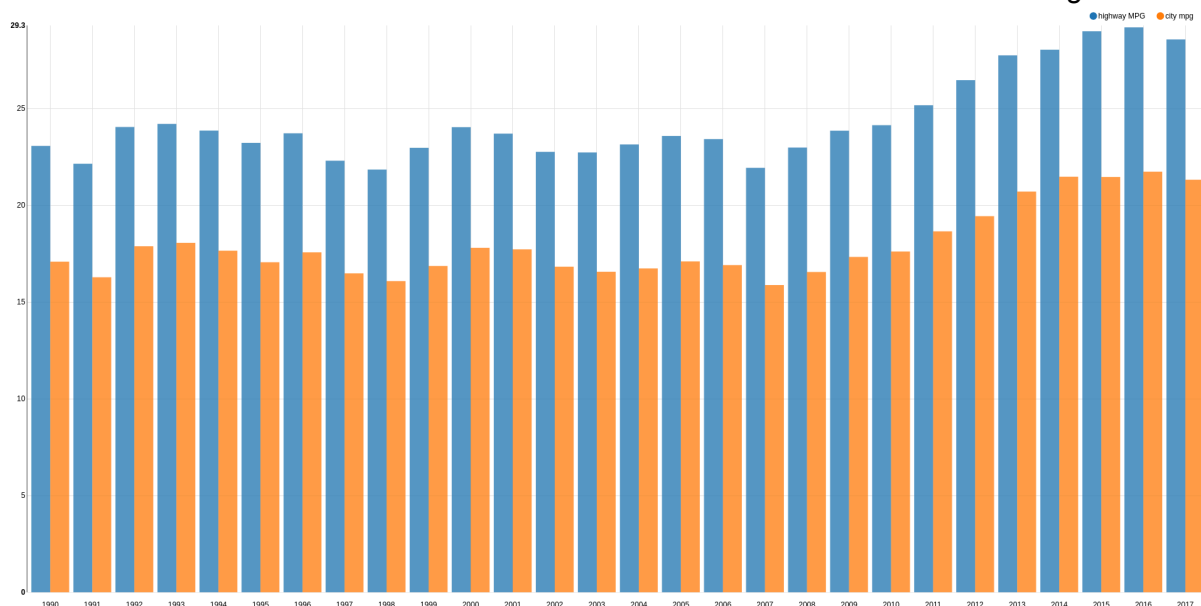
Para ter uma ideia inicial de como os atributos presentes no dataset podem afetar o consumo do carro para além do uso da métrica de correlação (Tabela de Correlações), fizemos uma análise visual do mesmo através de diversos gráficos.

Make: existem 48 marcas diferentes neste dataset e como é visível através dos gráficos este atributo afeta significativamente o consumo médio. Também é possível ver facilmente um outliers, sendo este os carros da tesla, isto deve-se por todos os veículos desta marca serem eléctricos.

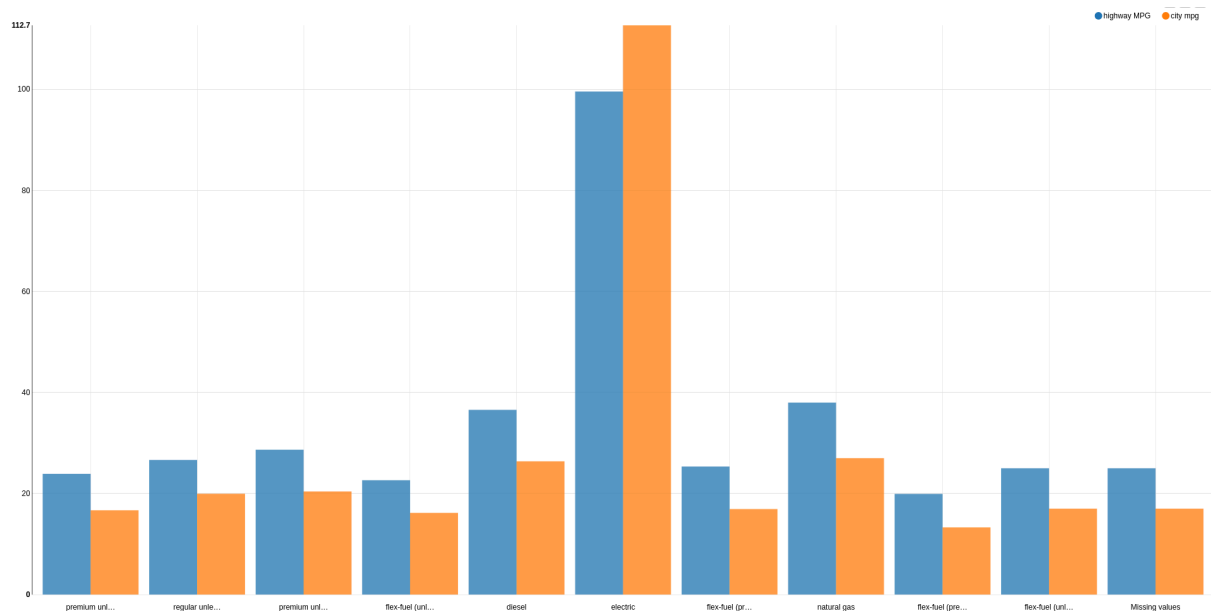


Model: existem 915 modelos diferentes, devido a esta quantidade, mesmo com a ajuda de gráficos é difícil uma boa análise da relação entre estes valores, porém devido a conhecimento comum, sabemos que dentro da mesma marca, dependendo do modelo do carro, os consumos dos carros se podem alterar significativamente.

Year: o ano do carro afeta o consumo do mesmo, através do gráfico não vemos uma diferença muito grande entre os valores mínimos e máximo, porém a correlação entre o ano do carro e os consumos é significativa.

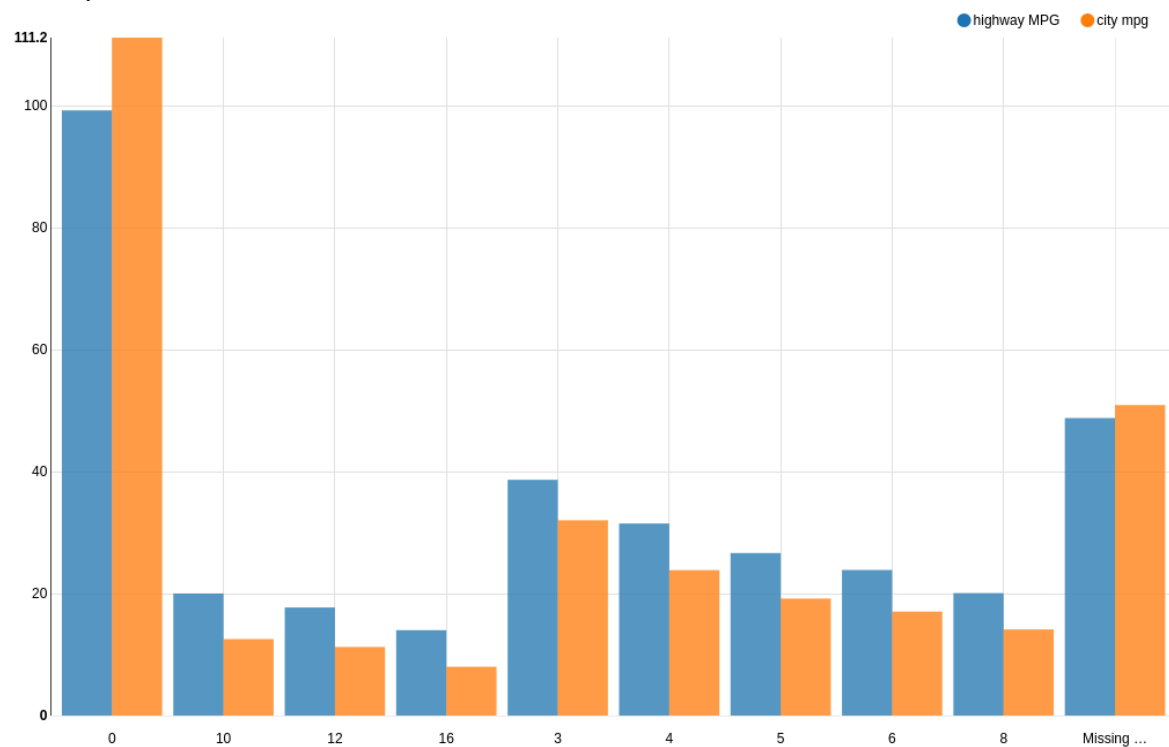


Engine fuel Type: o nosso dataset possui 11 tipos de combustíveis (incluindo elétricos), sendo a maioria combinações de flex-fuel, para além disso, ainda temos a existência de missing value. Novamente vemos que carros elétricos possuem valores muito acima da média de consumos.



Engine HP: este atributo afeta bastante o consumo do carro, como é possível ver pelo seu valor de correlação com os consumos.

Engine Cylinders: tal como o atributo anterior, também este influencia bastante o consumo do carro, seguindo maioritariamente o princípio de mais cilindros, mais consumo. Neste caso, os carros com 0 cilindros são maioritariamente carros elétricos, daí a discrepância entre os valores médios.



Transmission Type: Existem 5 tipos de transmissão e através da análise dos valores do gráficos chegamos a este valores:

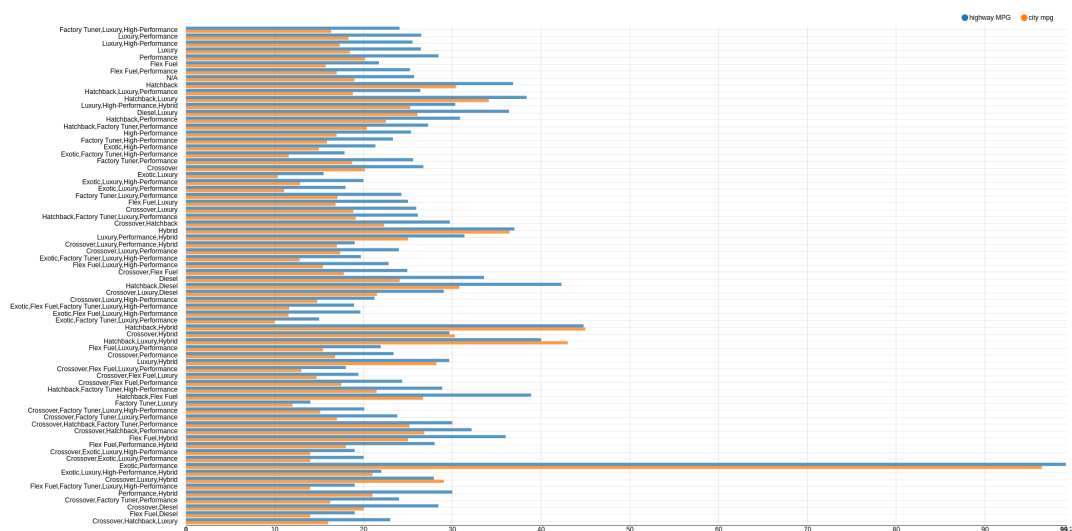
	Manual	Automatic	Automated_ Manual	Direct Drive	Unknown
HighWay MPG	26.88	25.78	29.21	110.80	20.47
City MPG	19.66	18.93	20.80	97.97	14.47
Nº de occ	2935	8266	626	68	19

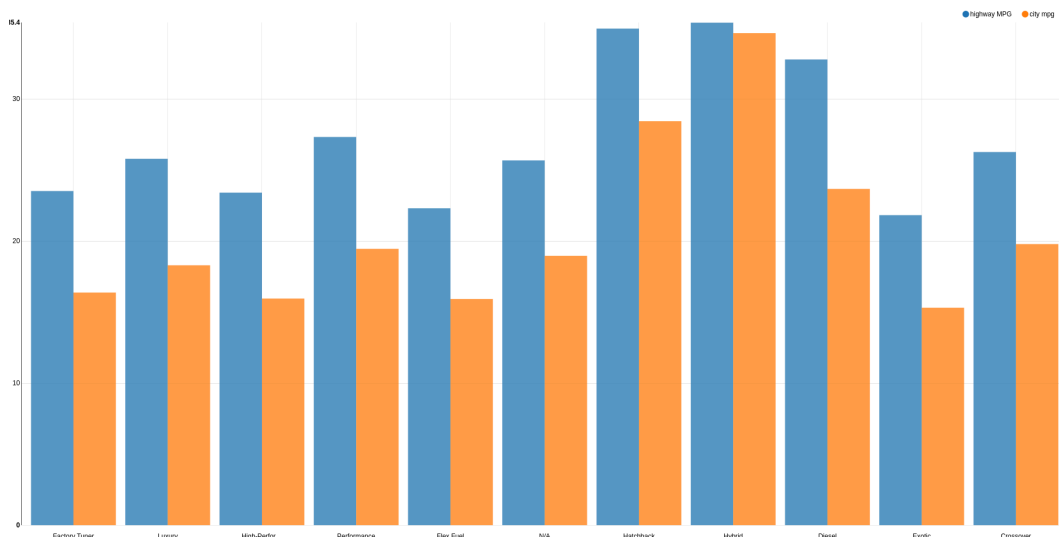
Transmissões Direct Drive são comuns em carros elétricos e, por isso, esta diferença de resultados.

Driven_Wheels: este valor indica qual é o tipo de tração do carro, havendo 4 tipos diferentes: *Rear Wheel Drive* (rwd), *Front Wheel Drive* (fwd), *All Wheel Drive* (awd) e *Four Wheel Drive* (4wd),

	rwd	fwd	awd	4wd
HighWay MPG	22.99	31.45	26.15	19,79
City MPG	16.52	23.67	19.10	15.07
Nº de occ	3371	4787	2353	1403

Market Category: este atributo é apresentado como uma lista de categorias, podemos assim ver a diferença que cada conjunto de categorias pode fazer no consumo, ou então ver a diferença que cada categoria individual pode fazer no consumo, para este segundo caso temos de dividir a lista de categorias usando o cell splitter, seguido de um ungroup, e assim chegamos aos gráficos abaixo. Analisando os gráficos podemos chegar à conclusão que as categorias como uma lista influenciam mais o consumo do carro do que estas de forma isolada.





Vehicle Size: Este atributo representa o tamanho do carro de forma categórica, havendo apenas 3 valores possíveis e como se esperava, quanto menor o carro, melhor o seu consumo.

	Compact	Midsize	Large
Highway MPG	28.94	26.80	22.42
City MPG	22.20	19.36	16.07
Nº occ	4764	4373	2777

MSRP: Este atributo representa o preço atual do carro sugerido pela marca e devido ao valor de correlação deste atributo com os consumos, podemos verificar que existe uma relação entre estes atributos.

2.3. Preparação dos Dados

Com o estudo dos dados feitos, seguimos para a preparação dos dados.

Começamos então por unir duas colunas, no caso as colunas relativas ao consumo, criando uma nova coluna “MPG” com o valor médio de ambas as colunas, tornando-se esta a coluna que queremos prever.

Devido a ser um outlier decidimos remover os carros eléctricos, além de também não fazer sentido um carro eléctrico ter como medida de consumo MPG.

Depois destas alterações iremos tratar dos missing values, que se encontram nos atributos EngineHP, Number of Doors, Engine Cylinders e Engine Fuel Type. Através dum nodo de Data Explorer vemos que existem 25 missing values de EngineHP, 1 missing value de Number of Doors, 20 missing values de Engine Cylinders e 3 missing values de Engine Fuel Type. Decidimos tratar estes missing values de maneiras diferentes, devido ao baixo valor de missing values de Number of Doors e Engine Fuel Type decidimos remover essas linhas, em relação ao Engine HP decidimos também os eliminar visto a ser um reduzido número de casos e não conseguimos descobrir os valores mais adequados, por fim no caso de Engine Cylinders reparamos que os casos em que possuíam missing values eram em carros que possuíam um motor “rotary” que não possuem cilindros, assim tornamos estes missing values em 0.

Por fim, removemos possíveis linhas repetidas e colunas não importantes, tais como Highway MPG e City MPG, que se tornam inúteis depois da criação da nova coluna MPG, outra coluna removida é a popularidade que em nada se relaciona com o consumo.

Após esta preparação geral dos dados verificamos novamente os dados de maneira a ver se alguma coisa mudou significativamente e, também para verificar se mais alterações eram necessárias.

Valores de Correlação antes de depois das alterações:

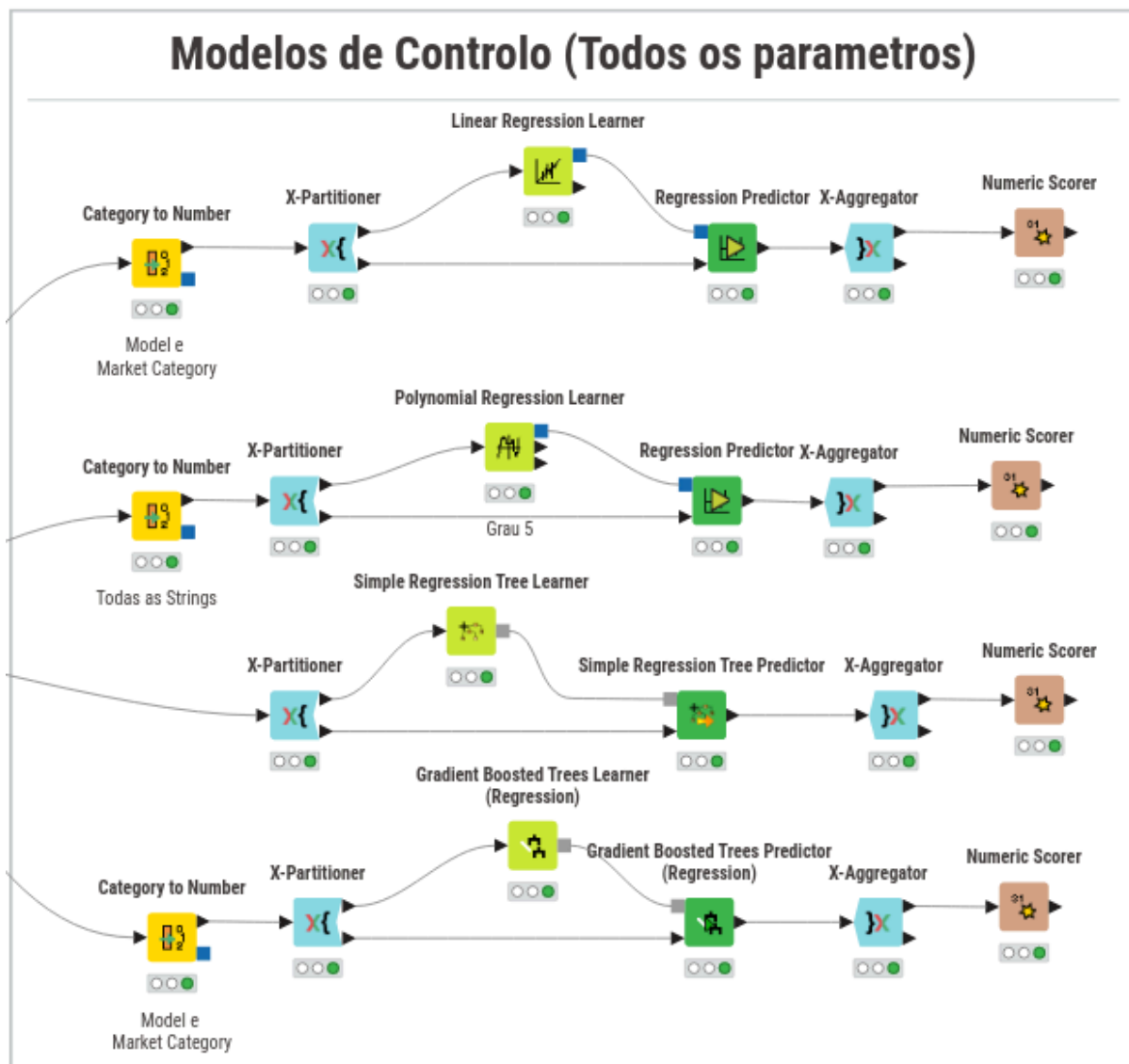
	Antes		Depois
	HighWayMPG	City MPG	MPG
Make	0.1583	0.2245	0.1637
Model	-0.1770	-0.1208	-0.1480
Year	0.3292	0.29787	0.3128
Fuel	0.0437	0.1020	0.0936
HP	-0.4925	-0.5840	-0.5540
Cylinders	-0.7622	-0.8358	-0.7951
Transmission	0.0310	0.0482	0.0554
Driven Wheels	-0.0747	-0.1103	-0.0950
Nº of Doors	0.1350	0.1563	0.1465
Market Category	0.0438	0.0291	0.0302
Size	-0.1193	-0.1796	-0.1425
Style	0.0651	0.0012	0.0394
Popularity	0.0105	0.0185	0.0180

MSRP	-0.2091	-0.2821	-0.2701
------	---------	---------	---------

Para além da preparação inicial dos dados, depois para certos modelos que criamos precisamos de fazer ainda mais preparações de forma a usar os modelos criados, que serão referidos aquando a explicação dos mesmos.

2.4. Modelação

2.4.1. Modelos de Controlo (com todos os parâmetros)



Nesta modelação inicial, testamos os modelos com todos os parâmetros de maneira a ter um grupo de resultados de controlo.

Foi usado **Cross Validation** de maneira a avaliar os modelos com diferentes grupos de treino e teste e evitar resultados não válidos devido ao grupos de treino e teste escolhidos.

Os algoritmos de aprendizagem utilizados foram **Linear Regression Learner**, **Polynomial Regression Learner**, **Simple Regression Learner** e **Gradient Trees Learner (Regression)**.

Como referido anteriormente, devido a como os alguns dos nodos de aprendizagem funcionam foi necessário fazer uma preparação de dados extras de maneira a usá-los.

No caso a passagem de certas Categorias para um número, no caso de **Linear Regression Learner** e **Gradient Trees Learner (Regression)** foi usado em Model e Market Category, uma vez que devido ao grande número de valores únicos(na forma de string), estes parâmetros eram ignorados, uma vez passados para number estes já não são ignorados. Já no **Polynomial Regression Learner**, transformamos todos os parâmetros categóricos para valores numéricos, visto que este nodo não tem em conta parâmetros categóricos.

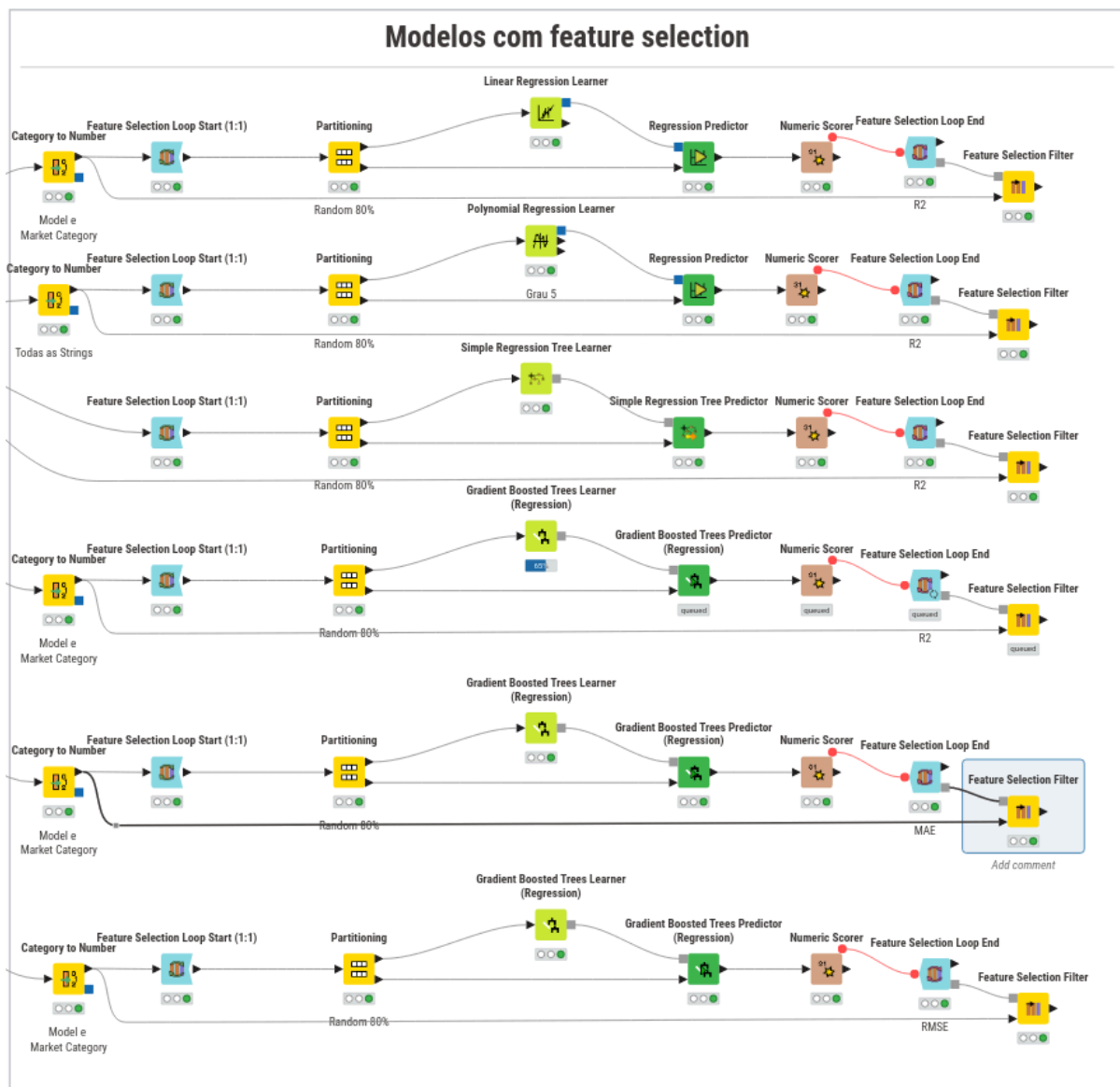
Assim obtivemos os seguintes resultados:

Linear Regression Learner		Polynomial Regression Learner	
R²:	0.765	R²:	0.768
Mean absolute error:	1.801	Mean absolute error:	1.811
Mean squared error:	8.492	Mean squared error:	8.396
Root mean squared error:	2.914	Root mean squared error:	2.898
Mean signed difference:	0.001	Mean signed difference:	-0.198
Mean absolute percentage error:	0.08	Mean absolute percentage error:	0.08
Adjusted R²:	0.765	Adjusted R²:	0.768
Simple Regression Learner		Gradient Trees Learner (Regression)	
R²:	0.854	R²:	0.865
Mean absolute error:	0.253	Mean absolute error:	0.244
Mean squared error:	5.281	Mean squared error:	4.892
Root mean squared error:	2.298	Root mean squared error:	2.212
Mean signed difference:	0.042	Mean signed difference:	0.026
Mean absolute percentage error:	0.011	Mean absolute percentage error:	0.01
Adjusted R²:	0.854	Adjusted R²:	0.865

Como é possível observar a partir dos resultados obtidos, o algoritmo **Gradient Trees Learner (Regression)** obteve os melhores resultados, sendo assim, o melhor algoritmo para este problema, não estando o algoritmo **Simple Regression Learner** muito atrás a nível de resultado. Vamos agora tentar alterar os parâmetros que estes algoritmos vão receber de maneira a tentar melhorar os resultados.

2.4.2. Modelos com feature selection

Nesta segunda modelação usamos como base os algoritmos que já tínhamos usado, mas desta vez usamos *Feature Selection*, que é um processo de seleção dos atributos mais importantes para a construção de um modelo de previsão. O *Knime* possui uma série de nodos que nos ajudam a determinar quais são estes atributos utilizando um modelo de previsão e testando de maneira a descobrir quais são os atributos que mais peso têm para determinar o resultado a ser previsto, dependendo do algoritmo que usarmos, sendo estes nodos *Feature Selection Loop Start*, *Feature Selection Loop End* e *Feature Selection Filter*.



Como podemos ver na imagem acima, foram realizado um total de 6 testes, usando os mesmos algoritmos utilizados anteriormente, no caso, nos algoritmos **Linear Regression Learner**, **Polynomial Regression Learner** e **Simple Regression Learner** apenas foi avaliado a métrica R^2 , visto que estes algoritmos não são ótimos para a resolução do problema onde foram vistas melhorias substanciais, no caso:

- **Linear Regression Learner** : $R^2 = 0,819$ (anteriormente: 0.765)
 - **Parâmetros:** Make; Year; Engine Fuel Type; Engine HP; Engine Cylinders; Driven_Wheels; Number of Doors; Market Category; Vehicle Size; Vehicle Style; MSRP;
- **Polynomial Regression Learner** : $R^2 = 0.829$ (anteriormente: 0.768)
 - **Parâmetros:** Make; Model; Year; Engine Fuel Type; Engine HP; Engine Cylinders; Transmission Type; Driven_Wheels; Market Category; Vehicle Category; Vehicle Style; MSRP;
- **Simple Regression Learner:** $R^2 = 0.975$ (anteriormente: 0.854)

- **Parâmetros:** Make; Model; Year; Engine HP; Engine Cylinders; Transmission Type; Driven_Wheels; Number of Doors; Market Category;

No caso do algoritmo **Gradient Trees Learner (Regression)**, devido a ter os melhores resultados inicialmente, para além de calcular a métrica R^2 , a esta se juntam o MAE (*mean absolute error*) e RMSE (*root mean squared error*), obtendo estes resultados:

- **R^2** : 0.987 (anteriormente 0.865)
 - **Parâmetros:** Model; Year; Engine HP; Engine Cylinders; Transmission Type; Driven_Wheels; Market Category;
- **MAE** : 0.218 (anteriormente 0.244)
 - **Parâmetros:** Make; Model; Year; Engine Fuel Type; Engine HP; Engine Cylinders; Transmission Type; Driven_Wheels; Vehicle Size; Vehicle Style;
- **RMSE** : 0.664 (anteriormente 2.212)
 - **Parâmetros:** Model; Year; Engine HP; Engine Cylinders; Transmission Type; Driven_Wheels; Market Category

Como podemos ver, dependendo da métrica que pretendemos melhorar os parâmetros escolhidos mudam, para além disso é importante notar que mesmo estes sendo os atributos que obtiveram os melhores resultados, outros conjuntos de parâmetros obtiveram valores bastantes próximos, senão até idênticos, o que revela que dependendo dos cenários outros conjuntos possam obter melhores resultados.

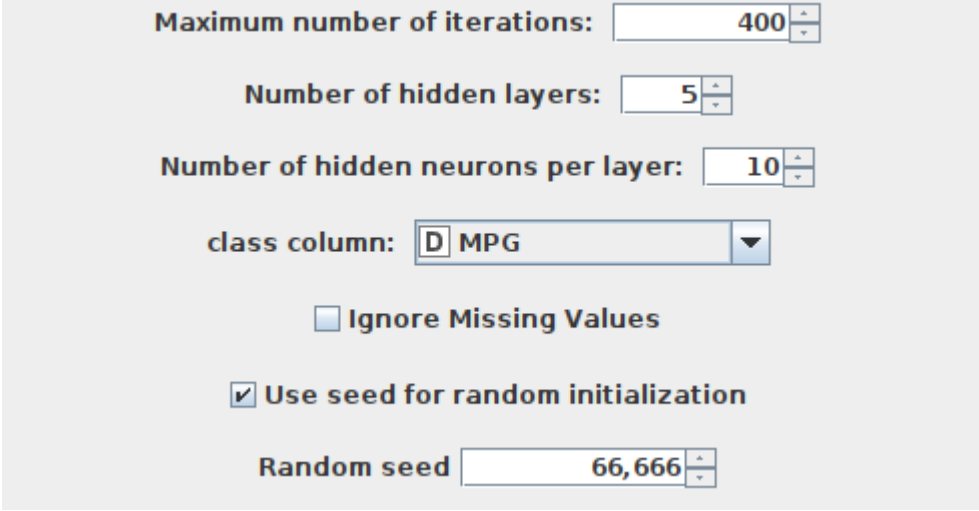
2.4.3. Modelo de Redes Neurais

Redes neurais são um tipo de modelo inspirado no cérebro humano. Elas consistem em várias camadas de neurônios conectados por sinapses. Esses neurônios recebem informações, processam-as e criam uma saída que é passada para a próxima camada. Assim, as redes neurais podem aprender através de exemplos, identificando padrões e relações entre os atributos que estes recebem. Em suma, é um modelo que imita a forma como o cérebro humano assimila informações.

Como algoritmo de rede neuronal usamos o nodo *RProp MLP Learner*. Este algoritmo implementado pelo nodo é o *Multilayer Feedforward Networks* que como o nome indica é *feedforward* e permite a existência de várias camadas de neurônios.

Para além da preparação inicial dos dados, nesta fase também foi necessário transformar todos os atributos em *Double's* visto que o nodo de aprendizagem apenas aceita esse tipo de atributo. Desta maneira, utilizamos o nodo *Category to number* para transformar todos os parâmetros categorias em inteiros e usamos, de seguida, o nodo *Math Formula* para transformar todos os inteiros em *doubles*.

Depois de transformar todos os atributos em *doubles*, é necessário normalizar todos os valores utilizando o nodo *Normalizer* de maneira a conseguir valores entre 0 e 1. Isto é necessário para o correto funcionamento do nodo de aprendizagem, mas, aquando a visualização dos resultados no *scorer* queremos comparar os resultados obtidos com os valores reais, o que implica algumas mudanças. Iremos duplicar a nossa coluna MPG e dar-lhe a coluna duplica o nome de Prediction (MPG), que irá corresponder à coluna com os valores previstos pelo modelo, e depois iremos usar o nodo *Normalizer*, depois iremos usar o modelo criado por esse normalizer nos nossos datasets de treino e teste, através do nodo *Normalizer(Apply)* e no fim, quando tivermos os nossos resultados, antes do *Scorer* iremos usar o nodo *Denormalizer*, com o modelo conseguido inicialmente.



Maximum number of iterations: 400

Number of hidden layers: 5

Number of hidden neurons per layer: 10

class column: D MPG

☐ Ignore Missing Values

☒ Use seed for random initialization

Random seed 66,666

Configuração do nodo RProp MLP

Nesta configuração conseguimos os melhores valores entre os nossos testes, tendo um número máximo de iterações de 400, 5 camadas escondidas e 10 neurónios por camada.

Gradient Trees Learner (Regression)		Rede Neuronal	
R ² :	0.865	R ² :	0.89
Mean absolute error:	0.244	Mean absolute error:	1.48
Mean squared error:	4.892	Mean squared error:	3.656
Root mean squared error:	2.212	Root mean squared error:	1.912
Mean signed difference:	0.026	Mean signed difference:	0.034
Mean absolute percentage error:	0.01	Mean absolute percentage error:	0.068
Adjusted R ² :	0.865	Adjusted R ² :	0.89

Podemos ver que comparando os resultados obtidos pela rede neuronal com os do algoritmo **Gradient Trees Learner (Regression)**, o primeiro consegue um valor de **R²** mais alto, indicando uma maior precisão dos valores obtidos. Porém os valores de **MAE** e **RSME** indicam que a rede neuronal possui menos erros de grande valor, devido ao seu **RSME** mais baixo, porém a sua média de erros é maior, devido ao seu **MAE** ser mais elevado.

2.5 Avaliação

De acordo com a nossa análise deste dataset, podemos concluir que, embora os atributos presentes nele não exijam muito tratamento, uma vez que se encontravam relativamente preparados, notamos que nem todos os atributos tinham a mesma relevância para prevermos o consumo médio dos carros. Através da nossa modelação, conseguimos chegar à conclusão que o melhor algoritmo para resolver este problema é o **Gradient Boosted Trees (Regression)** destacando-se por apresentar a menor média de erro. No entanto, em alguns casos, este algoritmo fez previsões mais discrepantes em relação ao esperado. Por esse motivo, é importante também considerar a modelação realizada com redes neurais, pois esta obteve valores mais próximos dos reais, com menos erros de grande magnitude.

3. Tarefa DataSet Atribuído

A metodologia usada nesta tarefa foi **CRISP-DM**.

G27-DataSet Ímpar <https://archive.ics.uci.edu/dataset/571/hcv+data>

O dataset atribuído apresenta 615 linhas e 18 colunas.

3.1 Estudo do negócio

Explorar, analisar e preparar os datasets, procurando extrair conhecimento deste mesmo, sempre com a coluna *Category (Category)* em mente, pois esta será a coluna que iremos tentar prever, a nossa coluna alvo.

O nosso objectivo passará então por, através de diversos modelos, prever esta coluna anteriormente falada. Para tal, usaremos a ferramenta KNIME.

3.2 Estudo dos dados

Age *Age*
Birth year *year_of_birth*
Birth month *month_of_birth*
Birth day *day_of_birth*
Sex *Sex*
Birth location *birth_location*
Albumin *ALB*
Alkaline phosphatase *ALP*
Alanine transferase *ALT*
Aspartate transferase *AST*
Bilirubin *BIL*
Cholinesterase *CHE*
Cholesterol *CHOL*
Creatinine *CREA*
Gama Glutamyl Transferase *GGT*
Protein *PROT*
Category *Category* ,

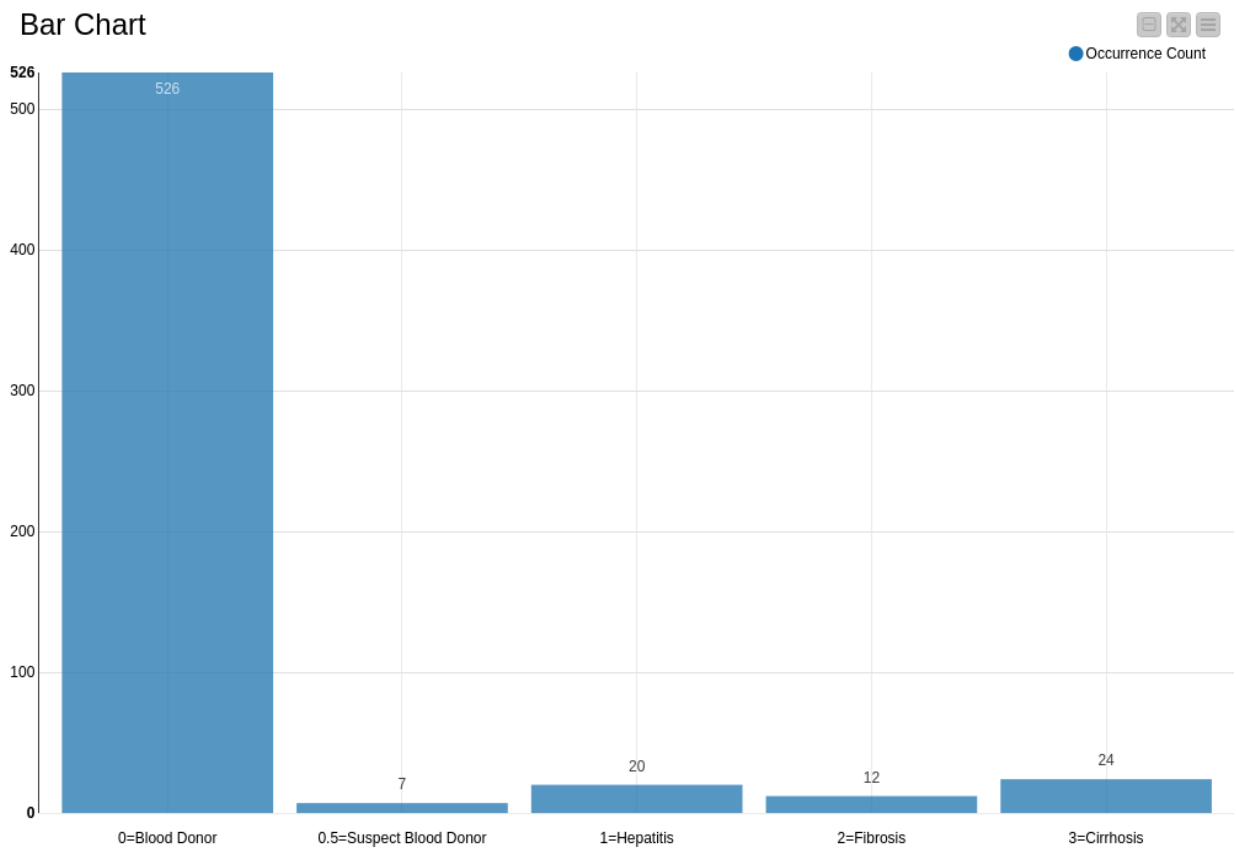
Estas são as colunas do nosso dataset, sendo então as linhas representadas pelas entradas na tabela.

Iremos nos referenciar a todas estas 10 colunas:

“(ALB,ALP,ALT,AST,BIL,CHE,CHOL,CREA,GGT,PROT)”

Como “Substâncias”, pois estas são isto mesmo, substâncias encontradas (em análise) ao sangue.

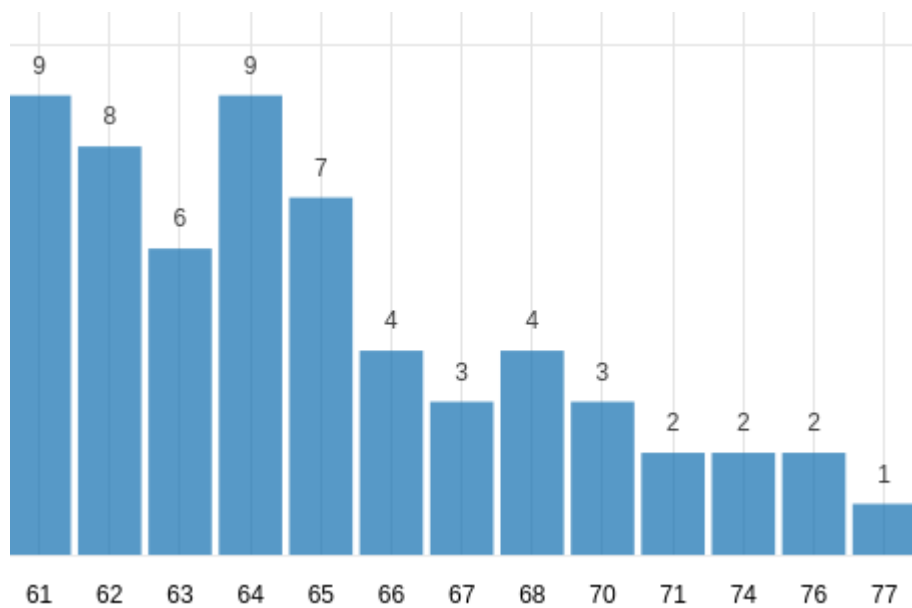
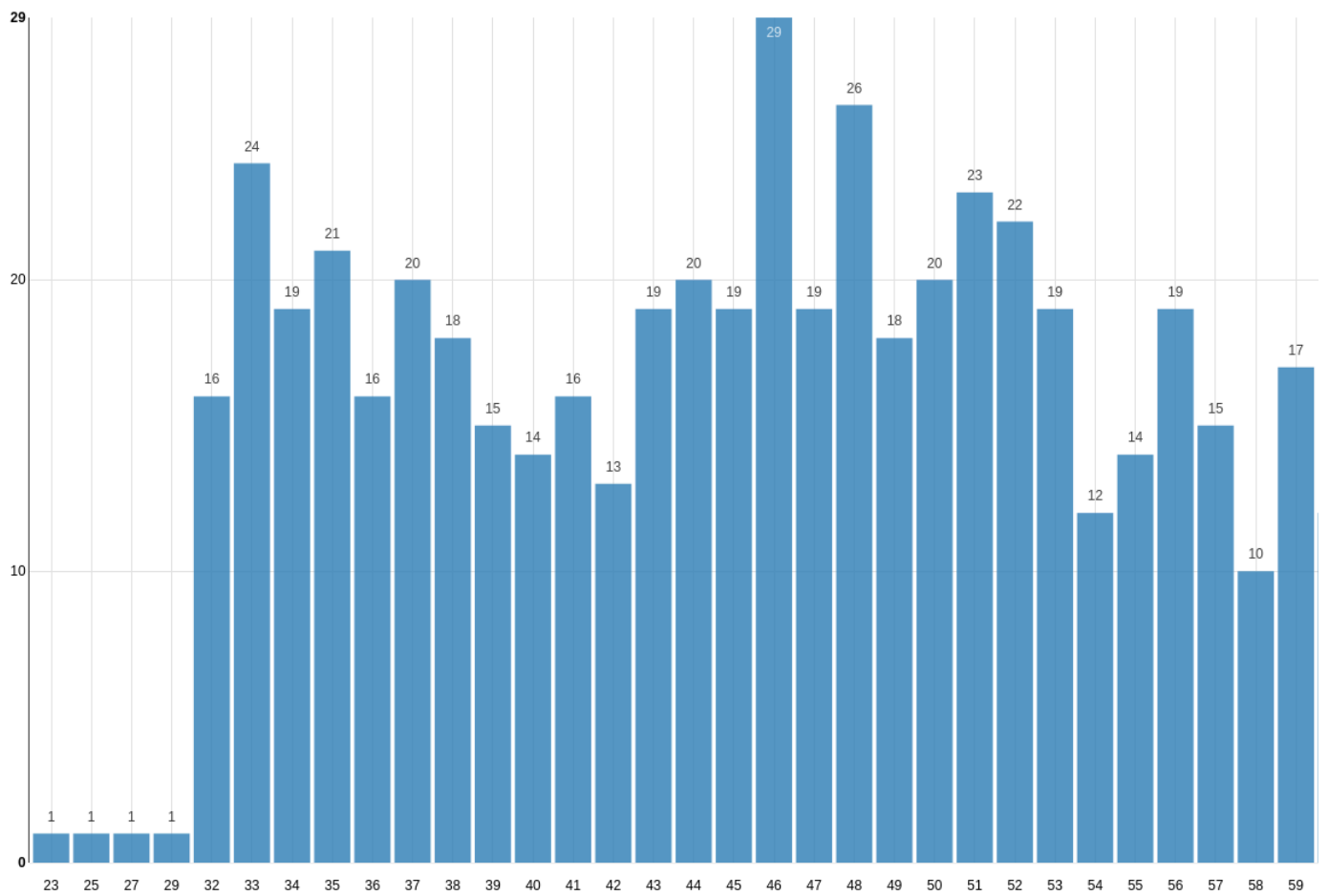
Category



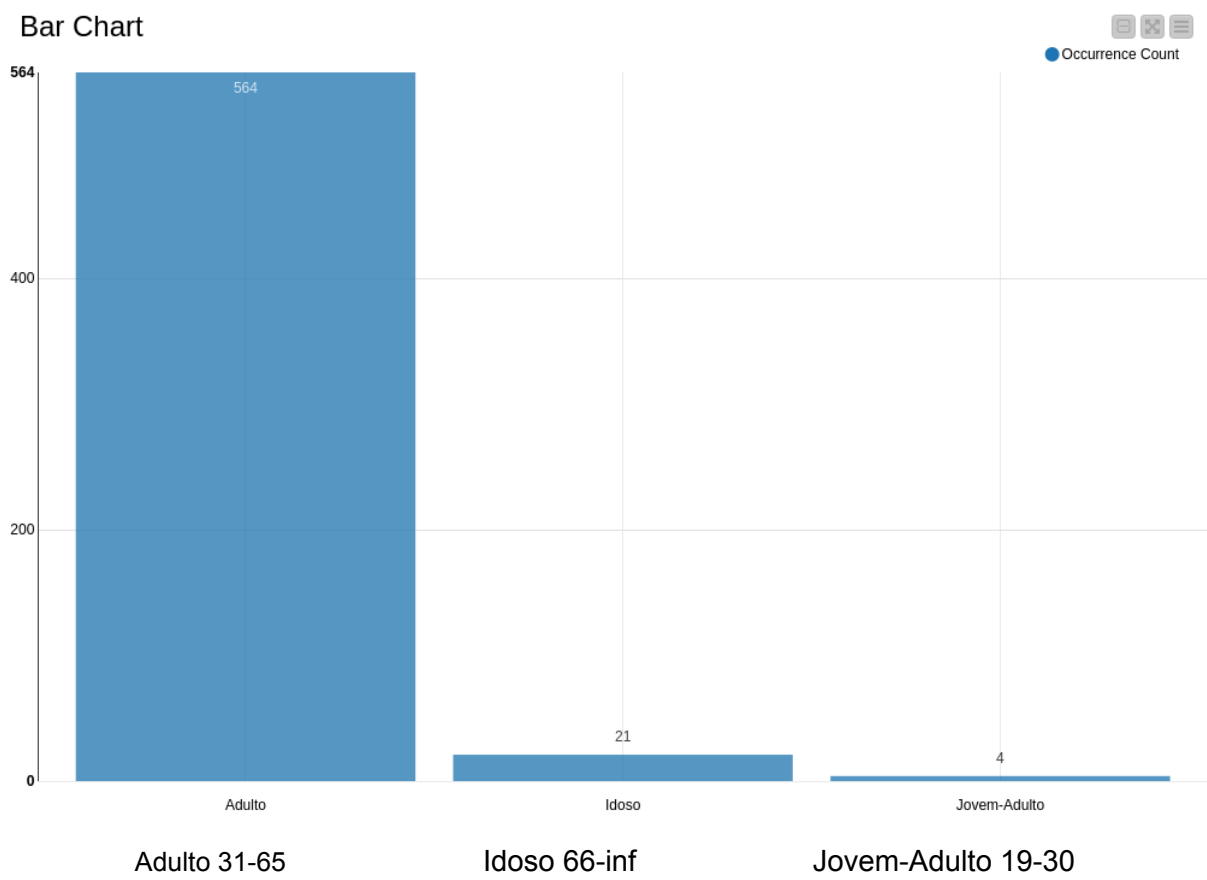
Existe uma esmagadora maioria da categoria **0=Blood Donor**, referenciado desde já, que na pré-preparação dos dados, foram retiradas entradas em que valores de substância não existiam, e achamos por bem retirar estes valores.

Idade

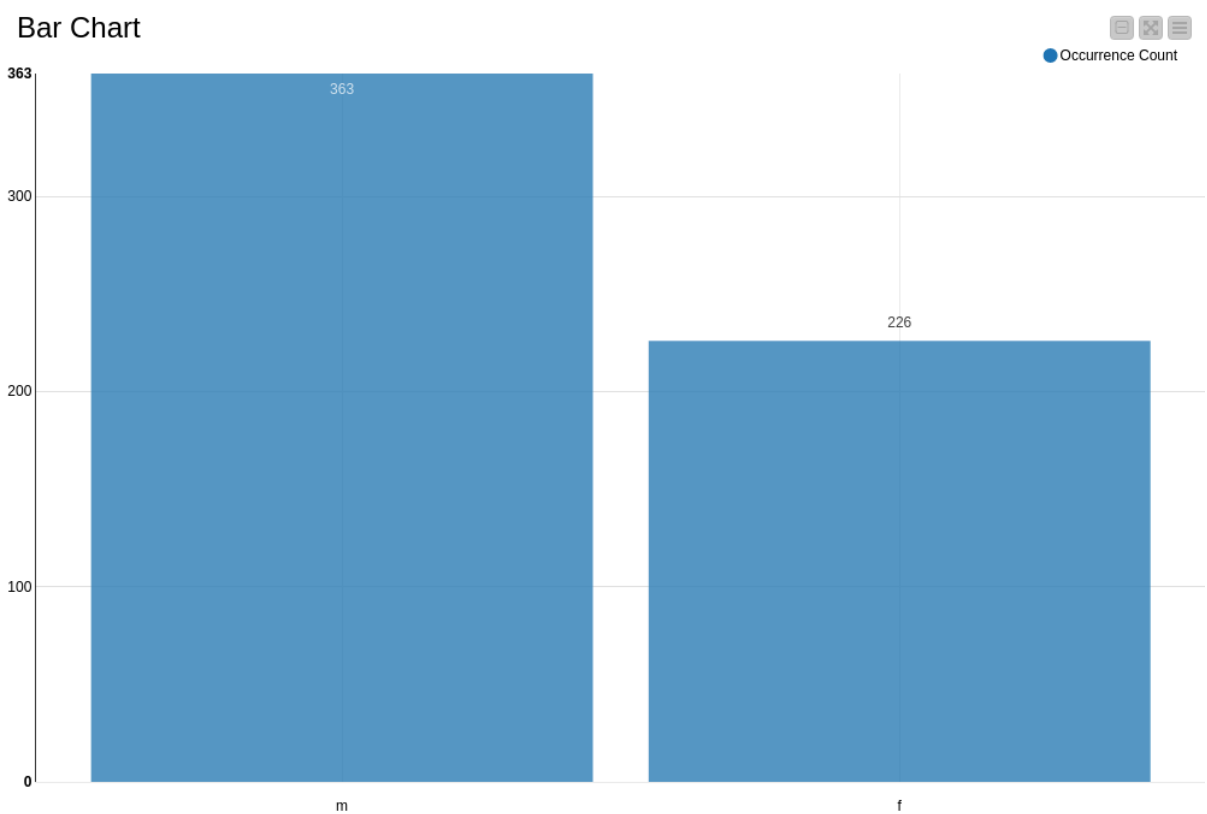
Bar Chart



Organizamos também as idades por grupos, para (tentar) verificar alguma tendência

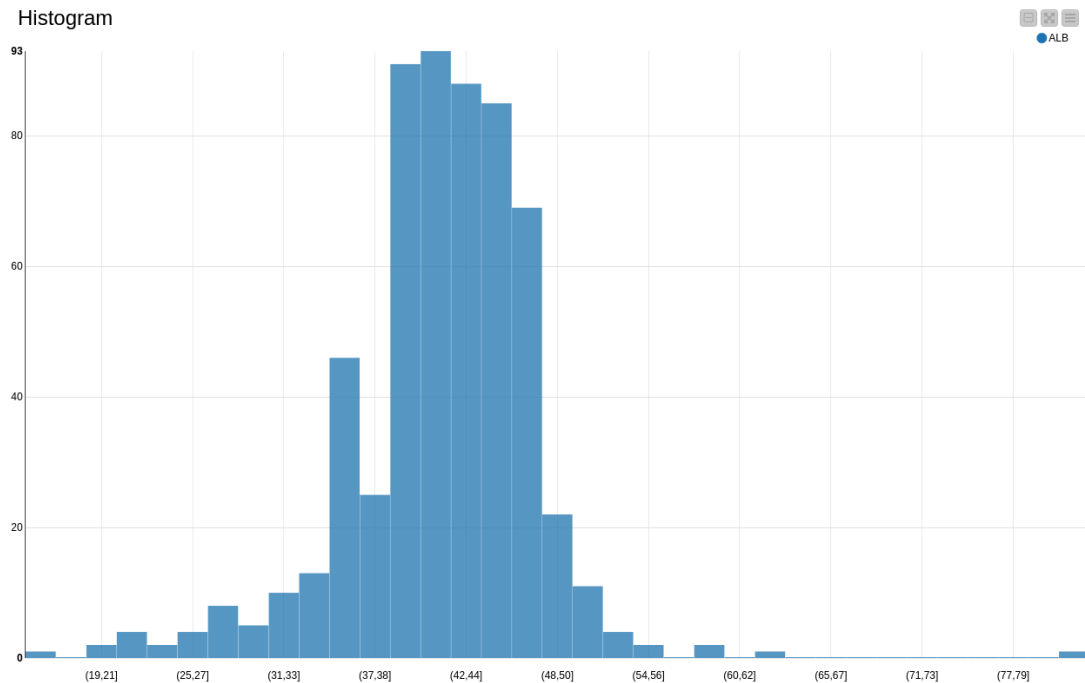


Sexo

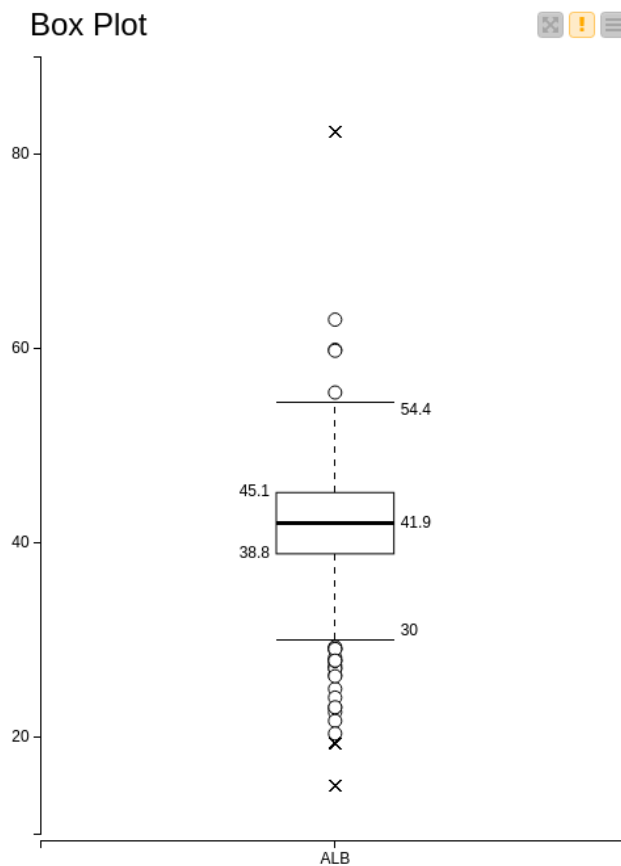


Visualização da Distribuição e Frequência da cada Substância e Detecção de *Outliers* nestes mesmos

Visualizamos e detectamos para todas e cada substância, usando BoxPlots e Histogramas
(Exemplo com Substância ***Albumin*** *ALB*)



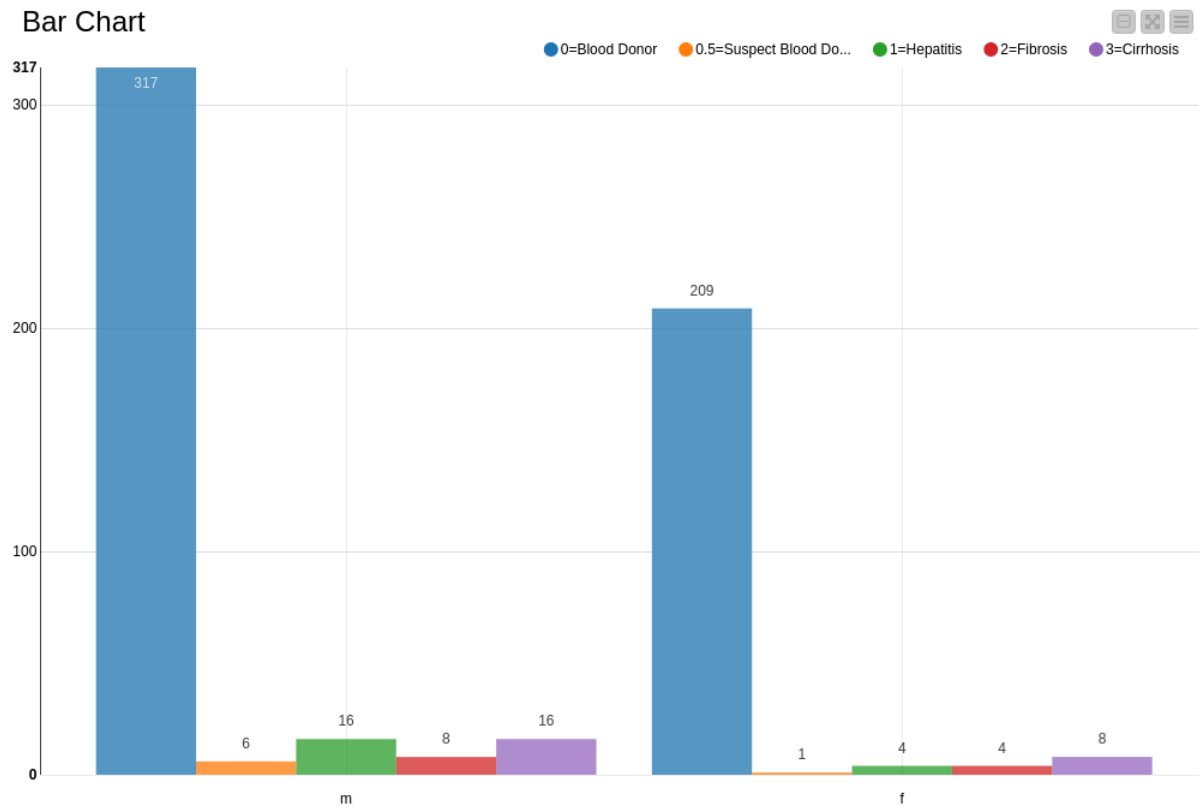
Box Plot



Cruzamento de (algumas) Colunas

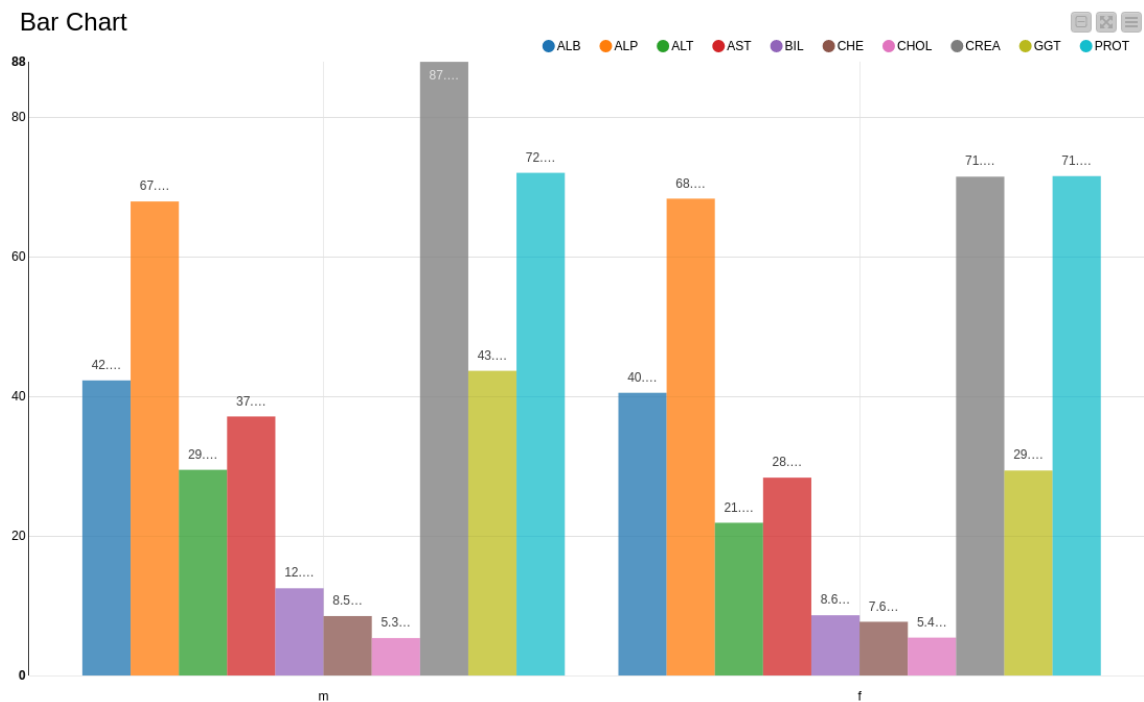
Sexo/Category

Bar Chart

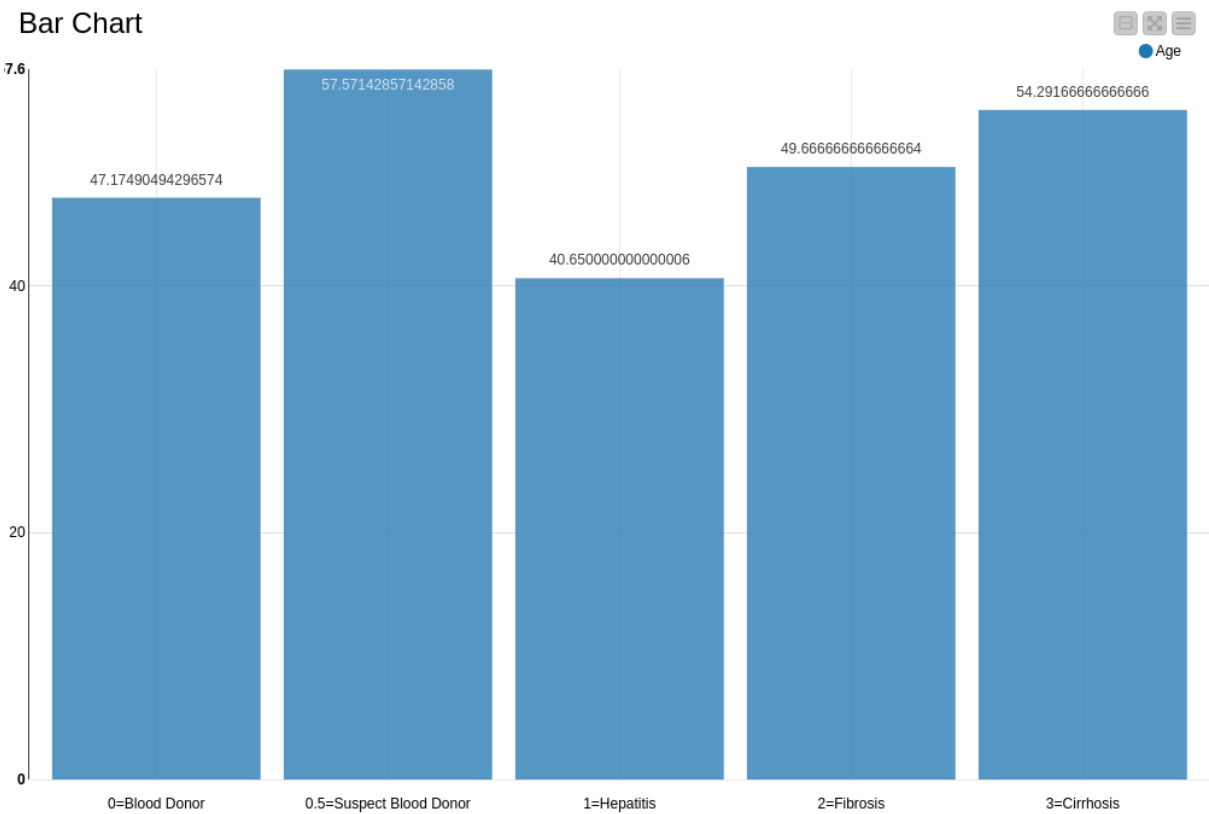


Sexo/Substancias

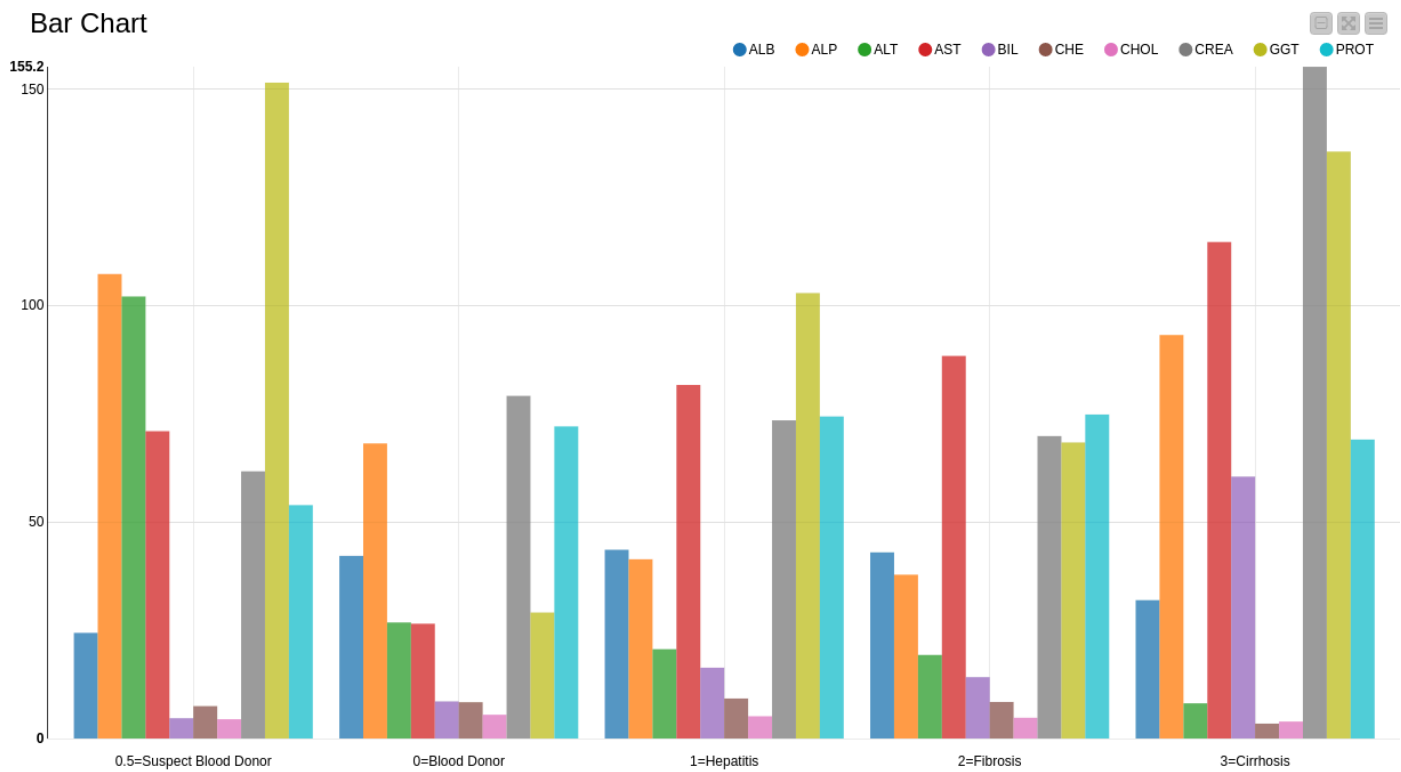
Bar Chart



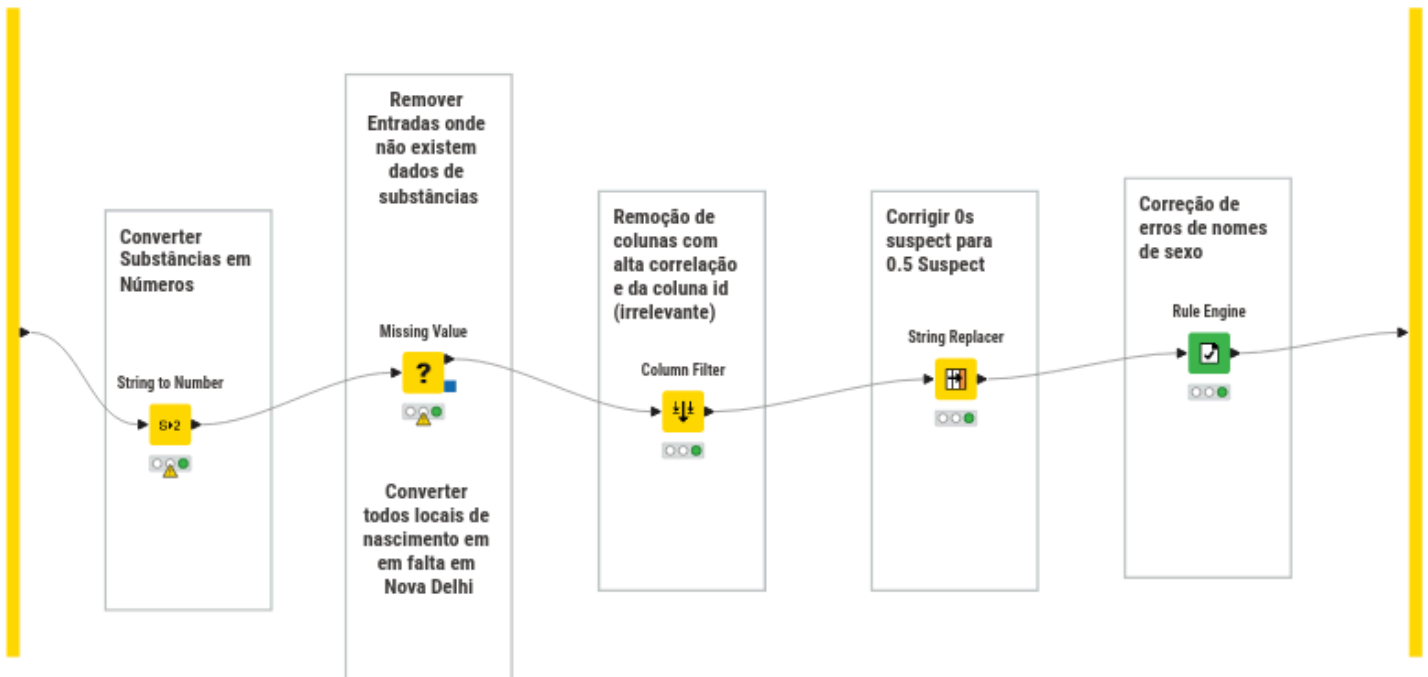
Categoria/Média Idades



Categoria / Média Substâncias



3.3 Preparação dos dados



1- Valores de Substâncias eram *strings* e foram convertidas em doubles.

2.1-Entradas onde não existem dados sobre substâncias foram retirados

2.2-Visto que todos os locais de nascimento são Nova Delhi, os locais de nascimento em falta foram também colocadas a Nova Delhi, mas esta informação é irrelevante e, foi, mais a frente, retirada.

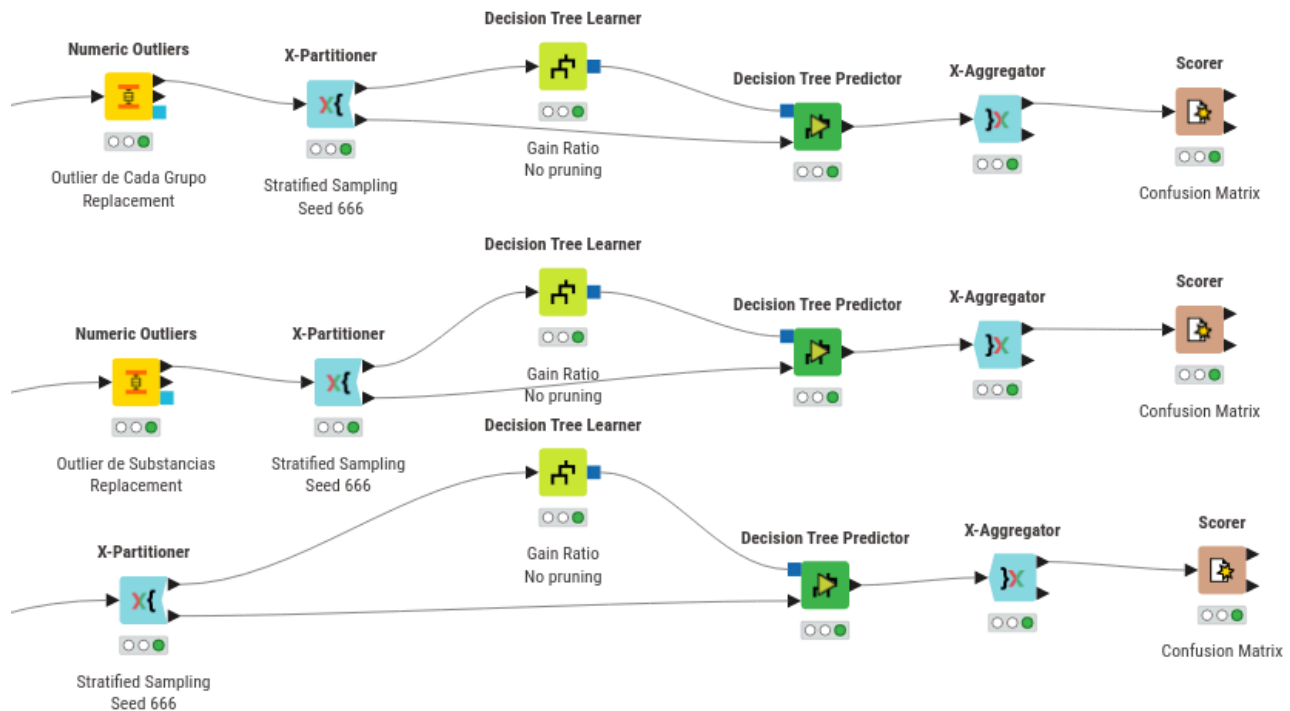
3-Removidas colunas de data de nascimento, dia de nascimento e mês de nascimento pois estes tornam-se irrelevantes devido a existência da coluna idade.

4-Mudado o nome da categoria 0s=suspect Blood Donor para 0.5=Suspect Blood Donor pelo simples motivo de ser mais legível.

5- Existiam entradas onde o sexo "m" (masculino) era "mm".

3.4 Modelação

3.4.1. Objectivo Classificação



Começamos a criação dos modelos com 3 modelos simples, utilizando, na terceira linha, os dados tratados nas fases anteriores, que são vistos como um modelo de “controle”. Todos os modelos aqui apresentados tem como nodo de partição o *X-Partitioner*, onde foi usado Stratified Sampling, sobre a coluna em alvo, a *Category*, e uma Seed de 666, assim como 10 validações.

Depois, decidimos compará-lo com outros modelos, onde foram aplicados outliers. A primeira aplicação do nodo *Numeric Outliers* foi a cada grupo, onde foram retiradas entradas de outliers de cada Categoria e outra aplicação foi feita para retirar outliers de substâncias, o que, devido ao facto de, como antes visto, o nosso dataset ser muito carregado de 0=Blood Donors, todas as pessoas que não se encontravam nesta categoria e que por sua vez tinham valores de substâncias (em algumas casos, muito) diferentes, foram retiradas.

Por esta razão a *accuracy* de cada modelo foi, respectivamente e em ordem decrescente, da remoção dos outliers por Categoria, para o dataset sem alterações, para os outliers do dataset por substâncias.

Em todos estes foram usados um *Decision Tree Learner* sem *pruning* e com *Gain Ratio* como *Quality Measure*.

1. Outliers Grupo

Category ...	0=Blood ...	0.5=Susp...	1=Hepati...	2=Fibrosis	3=Cirrhosis
0=Blood ...	525	0	0	0	1
0.5=Sus...	1	2	1	1	2
1=Hepati...	7	0	5	6	2
2=Fibrosis	3	0	4	4	1
3=Cirrhosis	3	0	3	0	18

Correct classified: 554	Wrong classified: 35
Accuracy: 94.058%	Error: 5.942%
Cohen's kappa (κ): 0.67%	

2.Outliers Substância

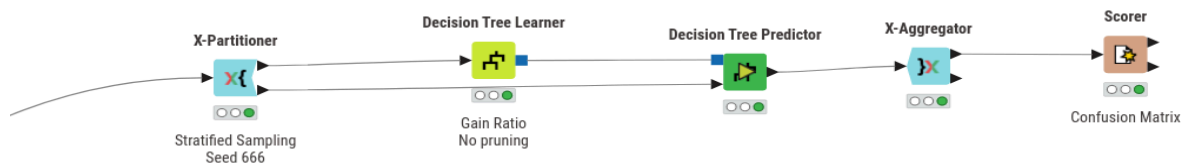
Category ...	0=Blood ...	0.5=Susp...	1=Hepati...	2=Fibrosis	3=Cirrhosis
0=Blood ...	514	5	4	3	0
0.5=Sus...	5	0	0	0	2
1=Hepati...	6	1	6	6	1
2=Fibrosis	2	0	3	4	3
3=Cirrhosis	1	2	1	3	17

Correct classified: 541	Wrong classified: 48
Accuracy: 91.851%	Error: 8.149%
Cohen's kappa (κ): 0.585%	

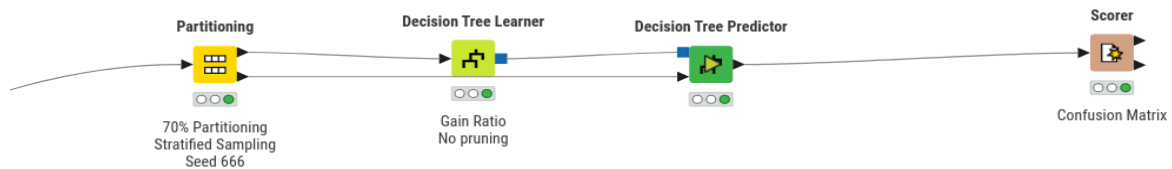
3.Controle

Category ...	0=Blood ...	0.5=Susp...	1=Hepati...	2=Fibrosis	3=Cirrhosis
0=Blood ...	517	2	5	1	1
0.5=Sus...	0	4	1	0	2
1=Hepati...	7	0	5	5	3
2=Fibrosis	3	0	6	2	1
3=Cirrhosis	0	3	2	2	17

Correct classified: 545	Wrong classified: 44
Accuracy: 92.53%	Error: 7.47%
Cohen's kappa (κ): 0.622%	



Teste com X-Partitioner / Partitioning



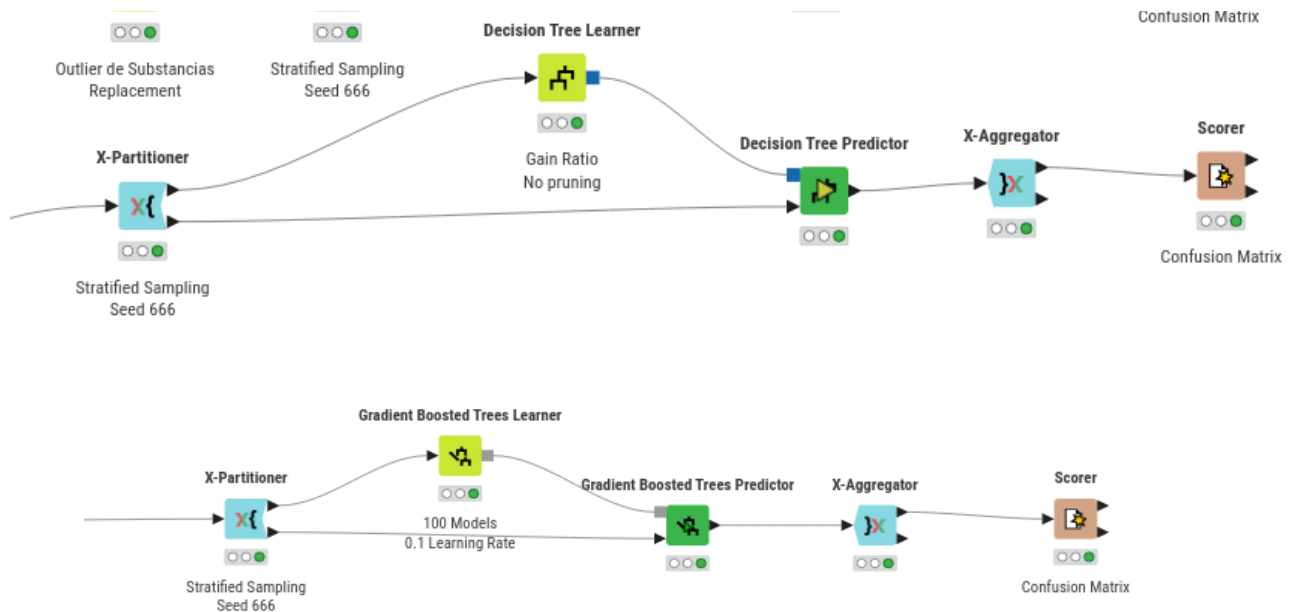
Comparando mais uma vez ao nosso Controlo, e mais uma vez mantendo os parâmetros na *Decision Tree Learner*, foi apenas feita uma comparação, mantendo a mesma estrutura de modelo, mas mudando o *partitioner*.

O nosso *Partitioning* apresenta como parâmetros, 70% do dataset para treino, e os restantes para testes. Mais uma vez somos deparados com Stratified Sampling, da coluna *Category*, e uma seed de 666.

Partitioning

Category ...	0=Blood ...	0.5=Susp...	1=Hepati...	2=Fibrosis	3=Cirrhosis
0=Blood ...	154	0	1	1	2
0.5=Sus...	0	0	1	1	0
1=Hepati...	2	0	2	1	1
2=Fibrosis	2	0	1	1	0
3=Cirrhosis	0	1	1	0	5
Correct classified: 162			Wrong classified: 15		
Accuracy: 91.525%			Error: 8.475%		
Cohen's kappa (κ): 0.576%					

Novamente, comparando o nosso controlo, desta vez, mudando o nosso *Learner* de *Decision Tree* -> *Gradient Boosted Trees Learner*

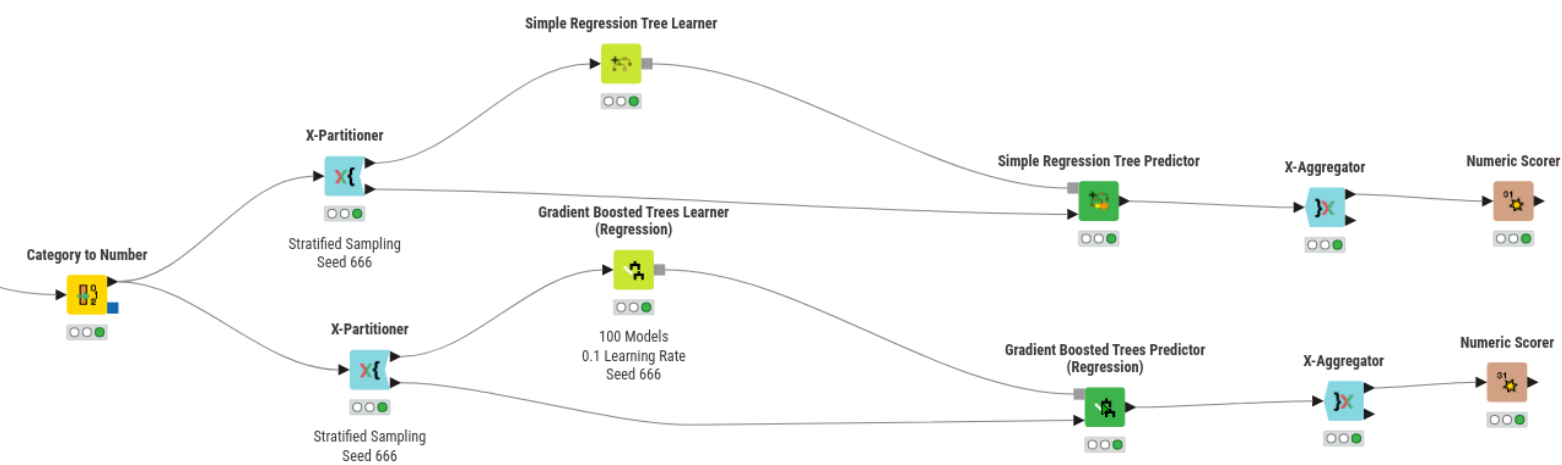


Gradient Boosted Trees Learner

Category ...	0=Blood ...	0.5=Susp...	1=Hepati...	2=Fibrosis	3=Cirrhosis
0=Blood ...	523	1	1	1	0
0.5=Sus...	1	4	1	1	0
1=Hepati...	5	0	6	7	2
2=Fibrosis	2	0	5	3	2
3=Cirrhosis	0	0	2	0	22
<div> <div>Correct classified: 558</div> <div>Wrong classified: 31</div> <div>Accuracy: 94.737%</div> <div>Error: 5.263%</div> <div>Cohen's kappa (κ): 0.725%</div> </div>					

Como previsto, a *accuracy* do *Gradient Boosted Trees Learner* foi o que teve melhor *accuracy*, tendo usado no seu nodo 100 modelos e uma *Learning Rate* de 0.1.

3.4.2. Modelação Objectivo Contínuo (Regressão)



Para tal ser possível, de usarmos modelos de aprendizagem sobre objetivos contínuos, numa coluna categórica, tivemos que aplicar o nodo *Category To Number*, na coluna *Category*.

Mais uma vez, usando *X-Partitioner* com Stratified Sampling e Seed 666.

No primeiro modelo, usamos a simples, *Simple Regression Tree Learner*, e no segundo, o nodo *Gradient Boosted Trees Learner (Regression)*, com 100 como número de modelos e uma *Learning Rate* de 0.1, onde como esperado, foi obtido um melhor R^2 .

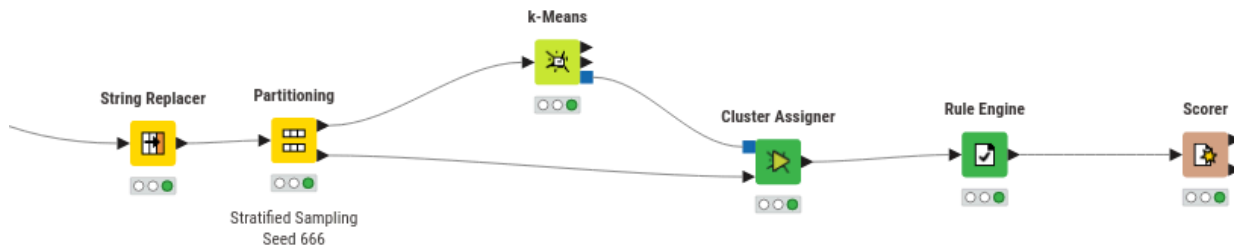
File	
R ² :	0.767
Mean absolute error:	0.105
Mean squared error:	0.207
Root mean squared error:	0.455
Mean signed difference:	0
Mean absolute percentage error:	0.053
Adjusted R ² :	0.767

Simple Regression Tree Learner
←←←←

Gradient Boosted Trees Learner (Regression)
→→→→

File	
R ² :	0.993
Mean absolute error:	0.018
Mean squared error:	0.006
Root mean squared error:	0.08
Mean signed difference:	-0.014
Mean absolute percentage error:	0.004
Adjusted R ² :	0.993

3.4.3. Clustering



Após análise dos dados, e de diversas tentativas, achamos que havia muito *overlap* do que era alguém de categoria 0=Blood Donor e 0.5=Suspect Blood Donor.

Devido a esta mesma razão, os cálculos deste modelo eram mais consistentes (dentro da sua inconsistência) quando agregamos ambas estas categorias numa só.

Rule Engine code

```
5 $Cluster$ = "cluster_0" => "0=Blood Donor"
6 $Cluster$ = "cluster_1" => "1=Hepatitis"
7 $Cluster$ = "cluster_2" => "2=Fibrosis"
8 $Cluster$ = "cluster_3" => "3=Cirrhosis"
```

Tivemos, como antes referido, problemas na consistência destes resultados, pois qualquer mudança mínima em qualquer parâmetro deste modelo, causava imensas variações na accuracy.

3.6 Avaliação

Podemos concluir, após análise detalhada deste dataset, que, apesar de uma extensa análise e estudo, foi possível resolver o problema em questão. Durante a fase de modelagem, experimentamos diversas abordagens para determinar a melhor estratégia de criação de um modelo de previsão. Ao fim da modelagem, concluímos que o modelo, embora todas elas tenham se demonstrado boas e eficazes, e embora a semelhança a nível de resultados, foram obtidos os melhores com Gradient Boosted Trees. Quanto aos modelos de com linhas removidas em caso de outliers, penso que, embora a accuracy tenha melhorado em relação ao dataset apenas tratado na preparação dos dados, afete a credibilidade e a validade deste grupo, pois, por muitas vezes, o que pode ser visto como outlier, é só alguém com níveis de substâncias elevado, o que por muitas vezes é traduzido, por exemplo, em alguém numa fase mais avançada de uma doença. Um problema que num caso de um dataset mais extenso, mais facilmente se chegaria a uma conclusão sobre este tema. O Clustering foi um caso excepcional, em que a capacidade de agrupar não foi tão bem sucedida como esperado, e assumimos pelo facto de que, muitas vezes, estas substâncias (sozinhas) não serão suficientes para calcular se o doente analisado estará ou não, doente, e com que tipo de doença.

4. Conclusão

Em suma, com este trabalho fomos capazes de aplicar diversos conceitos relacionados com o desenvolvimento de modelos de aprendizagem que foram discutidos ao longo do semestre, bem como alguns conceitos que não foram abordados.

Durante o desenvolvimento do projeto, exploramos os dados em profundidade, além de criar modelos de aprendizagem, também realizamos exploração e o pré-processamento de dados para compreender melhor o problema que tínhamos em mãos e identificar as melhores abordagens para obter respostas satisfatórias.

Embora tenham existido dificuldades, estamos contentes com o projeto realizado. Conseguimos desenvolver modelos de aprendizagem satisfatórios para os datasets explorados e documentação adequada para todo o processo de desenvolvimento do projeto.