
Aprendizagem e Decisão Inteligentes



Universidade do Minho
Escola de Engenharia

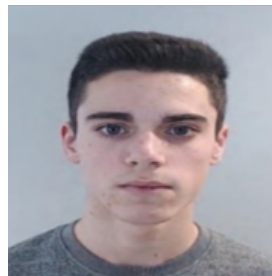
TRABALHO REALIZADO POR:

GONALO MARTINS DOS SANTOS
TELMO JOS  PEREIRA MACIEL
JO O PEDRO SILVA NEVES
GUILHERME SANTIAGO LOPES PEREIRA

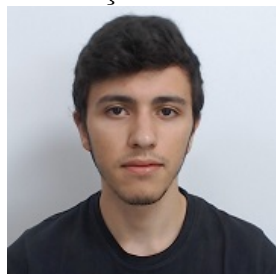
Grupo 33



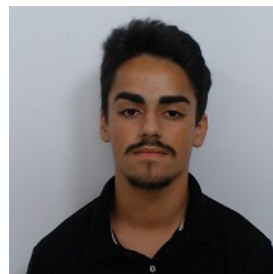
A95354
Gonalo Santos



A96569
Telmo Maciel



a89604
Jo o Neves



a89479
Guilherme Pereira

Índice

1	Introdução	2
2	Tarefa A	2
2.1	Descrição do dataset	2
2.2	Definição do objetivo	3
2.3	Exploração e Análise dos Dados	4
2.3.1	Exploração por Tempo	4
2.3.2	Exploração por Outro	7
2.3.3	Exploração por Veículo	8
2.3.4	Exploração por Localização	11
2.3.5	Exploração por Feridos e Fatais	13
2.4	Tratamento dos Dados	14
2.4.1	Remoção de colunas irrelevantes	14
2.4.2	Criação de colunas	14
2.4.3	Alteração de colunas	15
2.4.4	Missing Values	15
2.4.5	Reordenação das colunas	16
2.5	Modelos concebidos	16
2.5.1	Regressão	17
2.5.2	Classificação	17
2.6	Análise dos resultados obtidos	18
2.6.1	Regressão	18
2.6.2	Classificação	20
3	Tarefa B	22
3.1	Descrição do dataset	22
3.2	Exploração e Análise dos Dados	23
3.3	Tratamento dos Dados	24
3.3.1	Valores impossíveis da coluna actual_productivity	24
3.3.2	Missing Values	24
3.4	Exploração e Análise dos Dados - pós tratamento	25
3.5	Modelos concebidos	26
3.5.1	Regressão	26
3.5.2	Classificação	27
3.6	Análise dos resultados obtidos	28
3.6.1	Regressão	28
3.6.2	Classificação	30
4	Conclusão	31

1 Introdução

Neste trabalho prático foi nos pedido para realizar a análise, exploração e tratamento dos dados de dois data sets diferentes, cada um correspondente a uma tarefa diferente.

Para a Tarefa A, o nosso grupo escolheu o dataset "Acidentes de Trânsito Porto Alegre (2018 - 2022)" que contém os registros de acidentes na cidade de Porto Alegre, no Brasil, entre os anos de 2018 e 2022. Como objetivo decidimos realizar modelos de regressão para prever o número ferimentos e fatalidades causados por um dado acidente e ainda realizamos alguns modelos de classificação.

Para a Tarefa B, o nosso grupo teve o data set sobre a produção de vestuário no qual o **target** era a coluna "**actual_productivity**". Para realizar o estudo deste dataset, utilizamos várias técnicas de exploração, análise e tratamento de dados para, posteriormente, concebermos vários modelos tanto de regressão como de classificação.

2 Tarefa A

Para a tarefa A o nosso grupo decidiu escolher o dataset "Acidentes de Trânsito Porto Alegre (2018-2022)". Vamos começar por analisar o caso de estudo que temos em mãos.

2.1 Descrição do dataset

Este dataset retrata vários registros de acidentes de trânsito na cidade de Porto Alegre (Brasil) entre os anos de 2018 a 2022. Neste dataset temos várias variáveis, ou colunas, as quais passamos a explicar o seu significado:

- data_extracao - Data e hora de realização da extração de dados do sistema
- idacidente - Número de identificação do acidente.
- longitude - Coordenada geográfica (eixo X) de localização do ponto onde ocorreu o acidente.
- latitude- Coordenada geográfica (eixo Y) de localização do ponto onde ocorreu o acidente.
- log1- Nome do Logradouro onde ocorreu o acidente.
- log2- Nome do Logradouro que cruza o Logradouro no ponto onde ocorreu o acidente.
- predial1 - Número do Logradouro onde ocorreu o acidente
- tipo_acid - Informação descritiva do tipo de acidente
- queda_arr - Informação se no acidente houve queda de algum veículo em arroio (variável binária).
- data - Data em que ocorreu o acidente.
- dia_sem - Dia da semana em que ocorreu o acidente.
- hora - Hora em que ocorreu o acidente.
- feridos - Número de feridos no acidente.
- feridos_gr - Número de feridos graves no acidente.

-
- mortes - Contagem de vítimas fatais no momento do acidente.
 - morte_post - Contagem de vítimas fatais posteriores ao momento do acidente e relacionadas ao mesmo. É considerado morte posterior a vítima que veio a óbito 30 dias após a data do acidente de trânsito.
 - fatais - Somatório das vítimas fatais no momento do acidente e das vítimas posteriores relacionadas ao mesmo.
 - auto, taxi, lotacao, caminhao, motocicletas, bicicletas, carrocas, outro - Número de veículos do tipo automóvel envolvidos no acidente.
 - onibus_urb - Número de ônibus urbanos envolvidos no acidente
 - onibus_met - Número de ônibus metropolitanos envolvidos no acidente
 - onibus_int - Número de ônibus interurbanos envolvidos no acidente
 - acidentenoite_dia - Turno em que ocorreu o acidente.
 - regioao - Zona da cidade onde ocorreu o acidente de acordo com as divisões dos Postos de Controles Avançados (PCA) da fiscalização da EPTC.
 - cont_vit- Informação -se o acidente possui ou não vítimas.
 - ups - Unidade padrão de severidade: Peso -atribuído aos tipos de acidentes de acordo com a gravidade dos danos causados.
 - consorcio - Consórcio responsável pelo (s) ônibus urbano (s) envolvido (s) no acidente.

Conseguimos facilmente identificar que neste dataset temos variáveis que são facilmente inferidas a partir de outras como é o exemplo das variáveis mortes, morte_post e fatais em que a variável fatais é simplesmente a soma das outras duas.

Outros exemplos de que existem variáveis que são inferidas a partir de outras são:

- A variável dia_sem pode ser inferida a partir da variável data;
- A variável acidentenoite_dia pode ser inferida a partir da variável hora;
- A variável cont_vit pode ser inferida a partir das variáveis fatais, feridos e feridos_gr;

Posto isto, o grupo tinha agora que definir um objetivo de estudo para este dataset.

2.2 Definição do objetivo

Numa primeira tentativa, o grupo tentou definir como objetivo a previsão das fatalidades causadas por um acidente, no entanto, esta abordagem mostrou-se incorreta já que as correlações de todas as colunas com a coluna **fatais** eram todas muito baixas.

A razão para isto é a que em mais de 60 mil entradas do dataset eram muito poucas aquelas que tinham alguma fatalidade, ou seja, quase todas as entradas tinham a coluna fatais com o valor 0.

Então, o grupo decidiu, agora definitivamente, definir como objetivo prever o número de fatalidades e não fatalidades causadas por um acidente, ou seja, prever o somatório das colunas **fatais**, **feridos** e **feridos_gr**.

Agora o novo **target** tinha muito melhores correlações com todas as outras colunas porque o número de feridos e de feridos graves causados por acidente já eram mais variáveis do que o número de fatalidades.

2.3 Exploração e Análise dos Dados

Sabendo que queríamos estudar as fatalidades e os ferimentos, fizemos um pequeno tratamento dos dados para pudermos ter resultados mais produtivos na fase de exploração e análise:

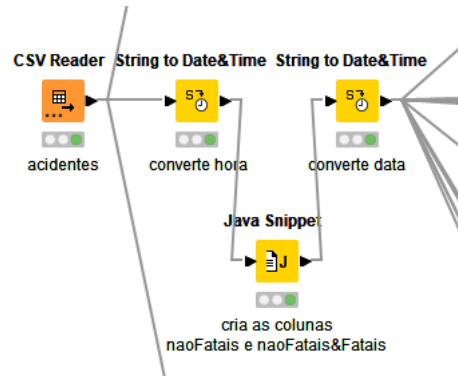


Figure 1: Pequeno tratamento dos dados antes da exploração

Como é possível perceber pela imagem, neste tratamento o que fazemos é converter as colunas hora e data de String para o tipo Date&Time e ainda criamos duas colunas. A coluna **naoFatais** é simplesmente a soma das colunas feridos e feridos_gr. A coluna **naoFatais&Fatais** é a soma entre as colunas feridos e feridos_gr (coluna naoFatais) e a coluna fatais, que por sua vez é a soma das colunas mortes e morte_post.

Tendo este pequeno tratamento feito podemos realmente passar para a exploração e análise dos dados.

2.3.1 Exploração por Tempo

Começamos por explorar de que maneira o factor do tempo influencia a ocorrência de feridos e mortos nos acidentes.



Figure 2: Exploração por Tempo

2.3.1.1 Dia da semana

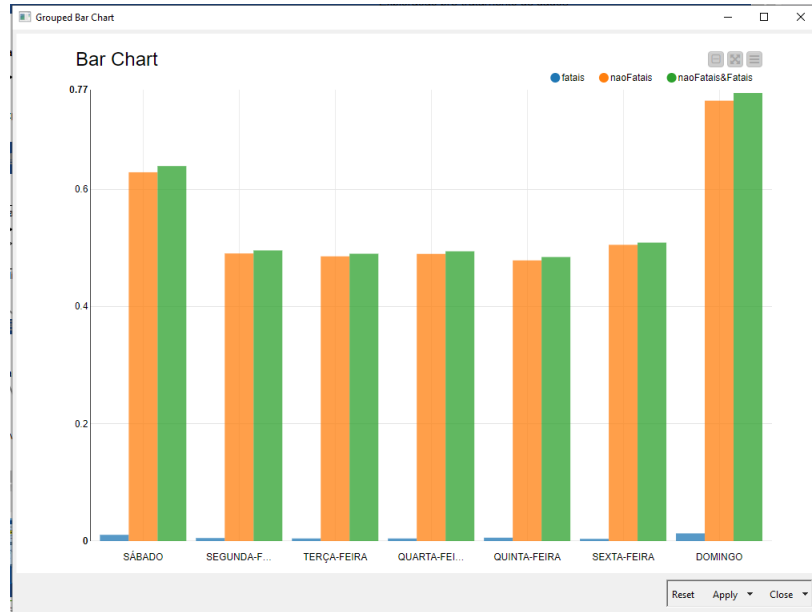


Figure 3: Dia da Semana

Deste gráfico podemos concluir que o número de fatalidades (coluna azul) é bastante reduzido em todos os dias da semana e permanece relativamente igual, já o mesmo não se pode dizer dos não fatais (coluna laranja) que são bem mais altos e as mudanças já são bem mais acentuadas. Concluimos que o dia da semana que provoca mais ferimentos e fatalidades é o Domingo seguido do Sábado, ou seja, que os dias de fim-de-semana são os mais perigosos. Os dias da semana são todos basicamente iguais (curioso de reparar que Sexta-Feira é o pior dos dias de semana).

A coluna verde é a coluna dos não fatais e dos fatais (a soma das outras duas).

2.3.1.2 Hora

Row ID	hora	Mean(fatalis)	Mean(naoFatalis)	Mean(naoFatalis&Fatalis)
Row23	22:00	0.012	0.951	0.963
Row24	23:00	0.014	0.905	0.919
Row22	21:00	0.016	0.871	0.887
Row21	20:00	0.011	0.817	0.829
Row20	19:00	0.01	0.811	0.821
Row6	05:00	0.03	0.748	0.778
Row4	03:00	0.02	0.741	0.761
Row3	02:00	0.03	0.716	0.746
Row1	00:00	0.016	0.724	0.74
Row5	04:00	0.019	0.721	0.739
Row2	01:00	0.012	0.674	0.686
Row19	18:00	0.004	0.651	0.655
Row7	06:00	0.008	0.625	0.633
Row18	17:00	0.006	0.552	0.557
Row0	?	0	0.53	0.53
Row8	07:00	0.006	0.51	0.516
Row17	16:00	0.002	0.461	0.463
Row14	13:00	0.004	0.435	0.439
Row13	12:00	0.004	0.422	0.426
Row9	08:00	0.003	0.419	0.421
Row15	14:00	0.004	0.405	0.409
Row16	15:00	0.004	0.397	0.401
Row12	11:00	0.005	0.378	0.384
Row10	09:00	0.002	0.366	0.369
Row11	10:00	0.003	0.33	0.333

Figure 4: Hora

É importante notar que nesta tabela a hora 00:00 representa todo o período entre a meia-noite e a uma da manhã (o mesmo aplica-se para as outras horas).

Desta tabela podemos concluir que a hora que provoca mais ferimentos e fatalidades é das 22:00 às 23:00 da noite, seguida das 23:00 à 00:00. A hora mais segura é entre as 10:00 e as 11:00 da manhã.

Estes resultados seguem o que seria esperado já que as horas mais perigosas são as horas de noite em que as pessoas podem conduzir embriagadas e as horas mais seguras são as horas da manhã, principalmente entre as 10:00 e as 11:00 que não é uma hora de ponta onde haja muito trânsito.

2.3.1.3 Dia e Noite

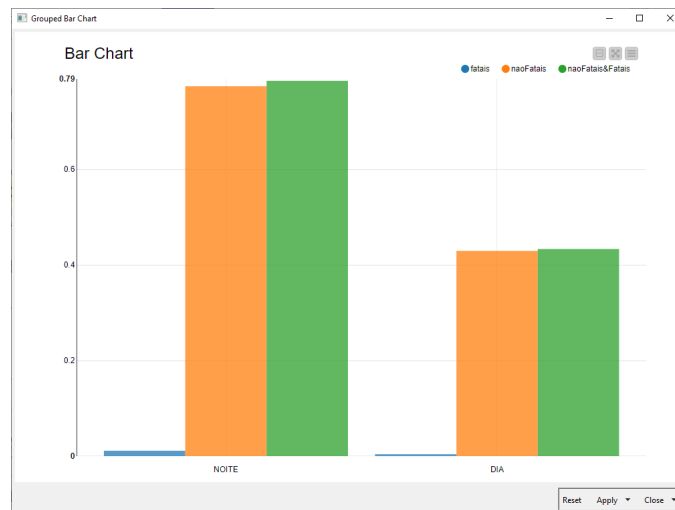


Figure 5: Dia e Noite

Deste gráfico podemos ver que as horas da noite são bem mais perigosas que as horas do dia, ou seja, que causam mais ferimentos e fatalidades.

2.3.1.4 Mês

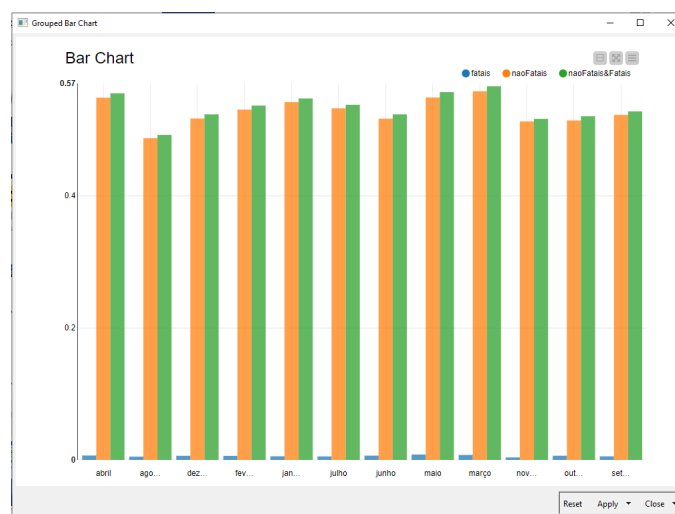


Figure 6: Mês

Observando este gráfico reparamos que o mês em que ocorre o acidente não tem muito impacto já que todos os meses têm relativamente o mesmo número de fatalidades

e ferimentos. O que podemos concluir é que a variável do mês não vai ter impacto no nosso **target**, ou seja, que vai ter uma correlação irrelevante.

2.3.2 Exploração por Outro

Continuamos a explorar de que maneira outras variáveis influenciam a ocorrência de feridos e mortos nos acidentes.

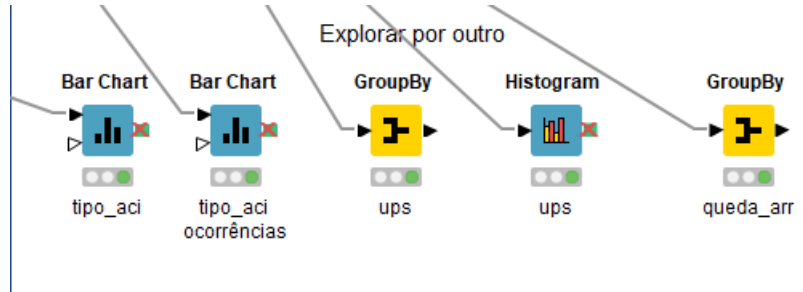


Figure 7: Exploração por outro

2.3.2.1 Tipo de acidente

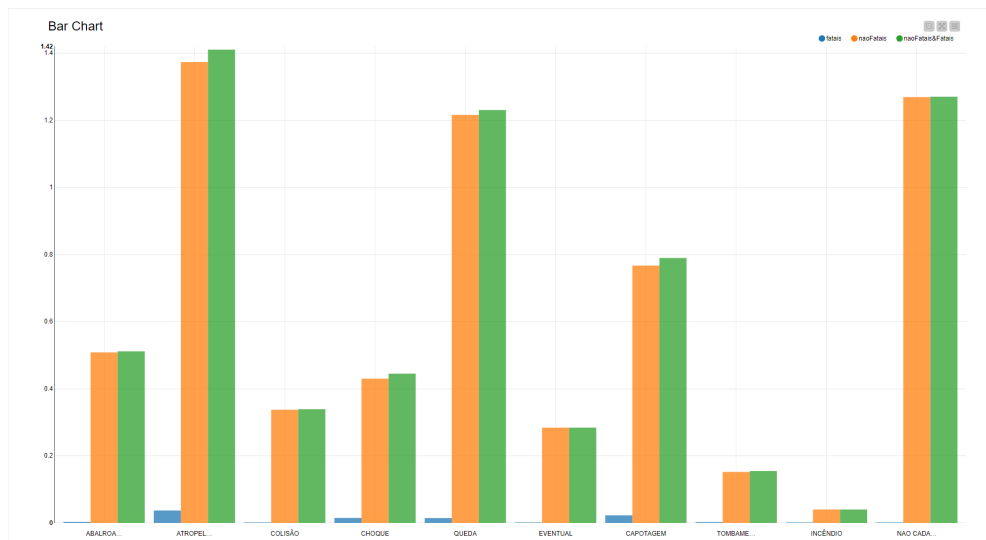


Figure 8: Tipo de Acidente

Deste gráfico podemos observar que o tipo de acidente mais perigoso é o atropelamento (causa mais mortes e ferimentos) e que o menos perigoso é o incêndio (causa menos mortes e ferimentos).

O atropelamento ser o mais perigoso era previsível. O incêndio ser o menos perigoso, é, provavelmente, consequência de ser o tipo de acidente com menos registos no dataset.

2.3.2.2 Unidade Padrão de Severidade (UPS)

Row ID	I ups	D Mean(fatais)	D Mean(iaoFatais)	D Mean(iaoFatais&Fatais)
Row0	1	0	0	0
Row1	5	0	1.408	1.408
Row2	13	1.028	0.417	1.445

Figure 9: Unidade padrão de severidade

Desta tabela podemos observar que esta variável apenas tem 3 valores diferentes (1, 5 e 13). Podemos também reparar que o nível mais seguro é o 1 no qual não existem nem ferimentos nem fatalidades, que o nível com mais ferimentos é o nível 5 e que o nível com mais fatalidades e maior soma de fatalidades com ferimentos (ou seja, o mais perigoso) é o nível 13.

Todos estes resultados são coerentes já que o **ups** mede a gravidade dos danos causados no acidente, ou seja, quanto mais graves forem os danos causados por um acidente mais fatalidades e ferimentos vão haver.

2.3.2.3 Queda em arroio

Row ID	D queda_...	D Mean(fatais)	D Mean(iaoFatais)	D Mean(iaoFatais&Fatais)
Row0	0	0.006	0.526	0.532
Row1	1	0	1.082	1.082

Figure 10: Queda em Arroio

Relembrando que esta variável representa a queda ou não de um veículo num arroio no acidente.

Desta tabela podemos ver que quando algum veículo que estava envolvido no acidente caí num arroio, não houveram fatalidades registadas mas o número de ferimentos foi maior do que quando nenhum caia num arroio. Tendo tudo em conta, é mais perigoso um acidente quando um veículo cai num arroio do que no caso contrário.

2.3.3 Exploração por Veículo

Depois de explorar o fator temporal e as diferentes variáveis acima referidas, decidimos explorar de que maneira os veículos envolvidos no acidente afetam os feridos e as fatalidades causadas pelo mesmo.

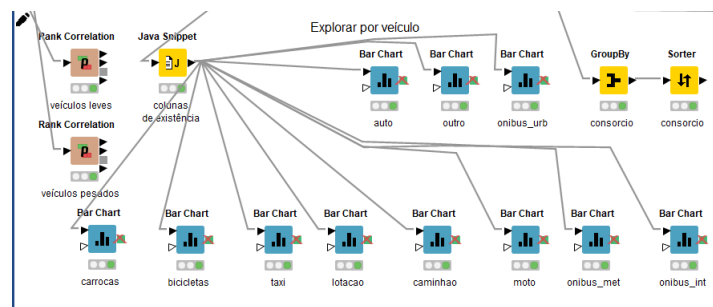


Figure 11: Exploração por Veículo

Começamos por realizar um mais um pequeno tratamento de dados para obter-mos melhores resultados da exploração. O nosso dataset tinha várias colunas (auto, outro,

carrocas, taxi, etc.) que representam o número de veículos envolvidos no acidente que são desse tipo, ou seja, se num registo o valor de carrocas for 3 significa que estiveram 3 carroças envolvidas no acidente. Posto isto, decidimos criar uma coluna para cada tipo de veículo para saber se esse tipo estava ou não envolvido no acidente ao invés de ser o número de veículos envolvidos.

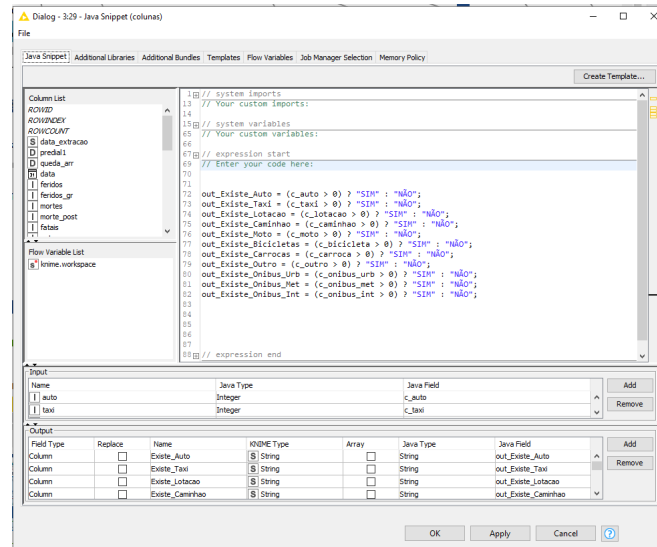


Figure 12: Java Snippet que realiza o tratamento dos dados referido

A nossa exploração por veículo vai se centrar muito nestas colunas que criamos agora. De maneira a simplificar a exploração vamos mostrar apenas os resultados mais relevantes.

2.3.3.1 Auto

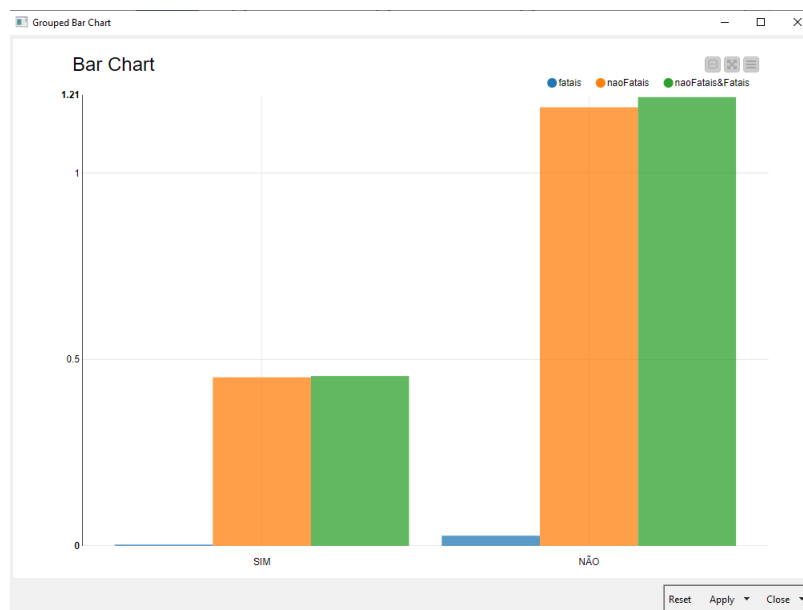


Figure 13: Existência de veículos do tipo auto no acidente

Deste gráfico podemos concluir que quando não existem automóveis do tipo auto envolvidos no acidente o número de fatalidades e ferimentos é muito maior, ou seja, os

acidentes acabam por ser mais perigosos.

2.3.3.2 Caminhão

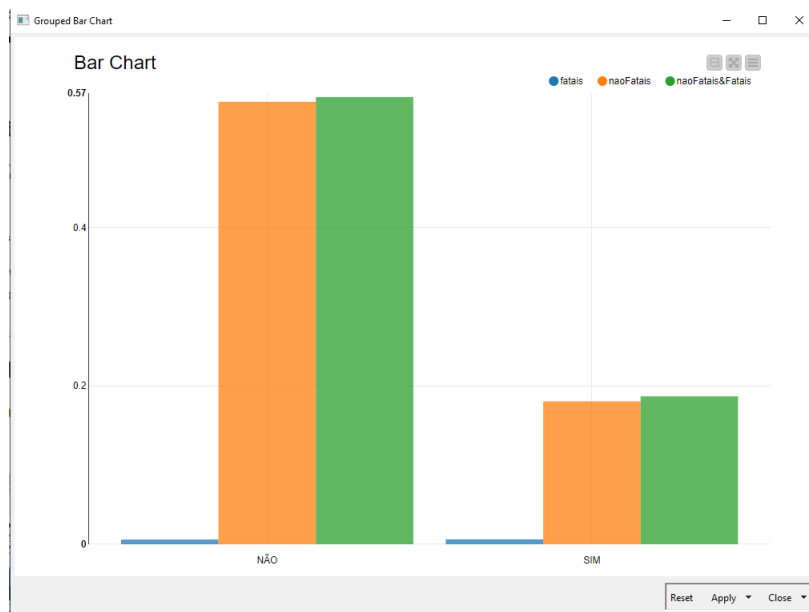


Figure 14: Existência de veículos do tipo caminhão no acidente

Deste gráfico podemos concluir que quando não existem caminhões envolvidos no acidente o número de ferimentos é muito maior do que quando existem, já as fatalidades são basicamente as mesmas em ambos os casos.

2.3.3.3 Moto

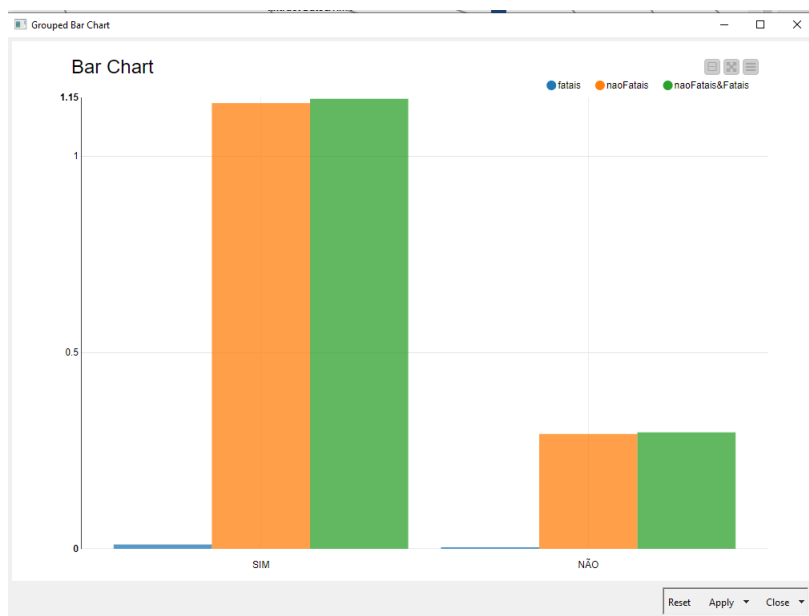


Figure 15: Existência de veículos do tipo moto no acidente

Deste gráfico podemos concluir que quando existem motos envolvidas num acidente

o número de fatalidades é ligeiramente superior e o número de ferimentos é muitíssimo superior do que quando as motos não estão envolvidas.

2.3.3.4 Bicicletas

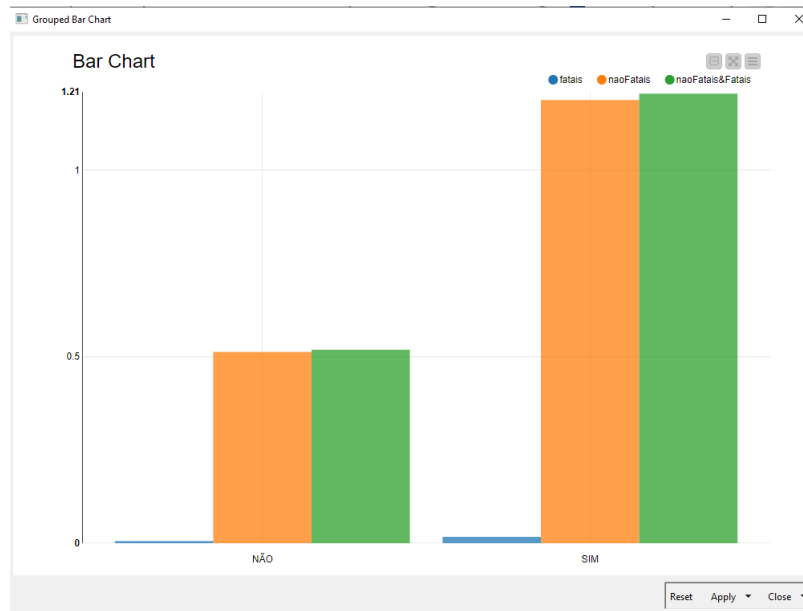


Figure 16: Existência de veículos do tipo bicicleta no acidente

Deste gráfico podemos concluir que quando existem bicicletas envolvidas num acidente, o número de fatalidades é ligeiramente superior e o número de ferimentos é muitíssimo superior do que quando as bicicletas não estão envolvidas.

2.3.4 Exploração por Localização

Nesta fase da exploração, o grupo achou apropriado passar a explorar de que maneira a localização do acidente afeta os feridos e as fatalidades causadas pelo mesmo.



Figure 17: Exploração por Localização

2.3.4.1 Região

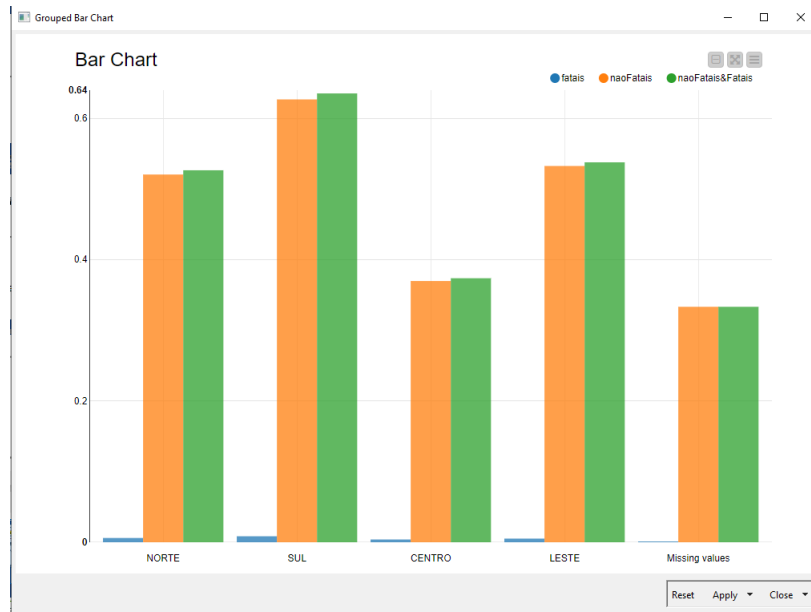


Figure 18: Região

A primeira coisa que podemos reparar é que esta variável contém Missing Values (no caso são 3 em mais de 60000 entradas). Ainda podemos observar que a região que causa mais ferimentos e fatalidades é a Sul e que a que causa menos de ambos é a Centro.

2.3.4.2 Predial

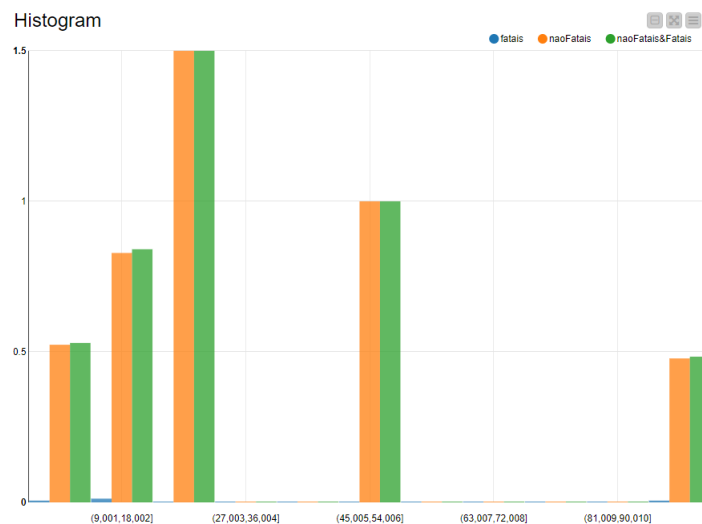


Figure 19: Região

Deste gráfico podemos observar que entre os valores de 18 002 e 27 003 o número de feridos é bastante maior do que em todos os outros. E podemos ver que entre os 9 001 e os 18 002 o número de fatalidades é o maior.

2.3.5 Exploração por Feridos e Fatais

Agora, decidimos comparar como as fatalidades e os ferimentos se relacionam entre si.

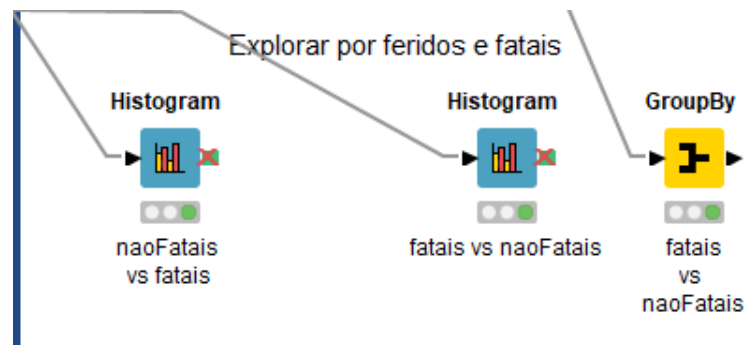


Figure 20: Exploração por feridos e fatais

2.3.5.1 Comparação: não fatais com fatais

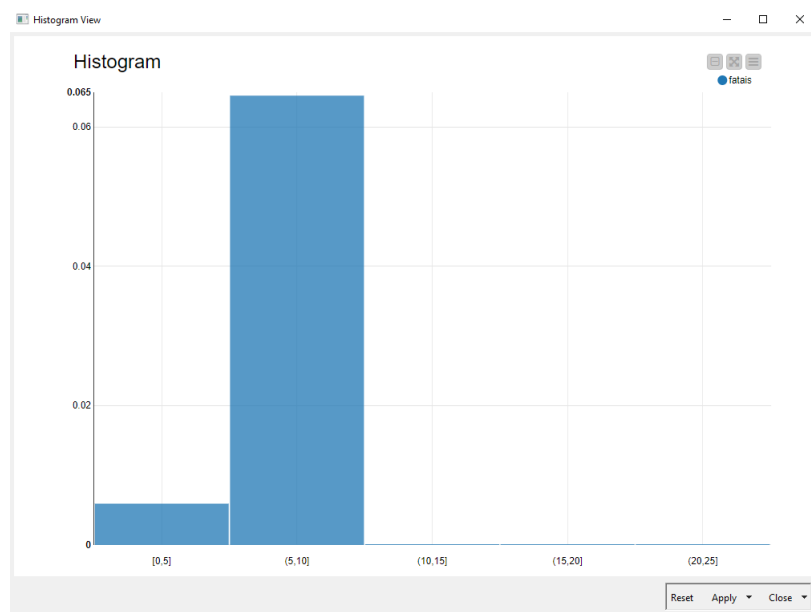


Figure 21: Comparação dos não fatais com as fatalidades

Deste gráfico podemos reparar que temos registos com 0 não fatais até registos com 25 não fatais (não fatais = feridos + feridos graves). Ainda podemos reparar que acidentes com não fatais entre os 5 e 10 são os que têm também mais fatalidades, a partir de 10 não fatais para cima não há registos com fatalidades.

2.3.5.2 Comparação: fatais com não fatais

Row ID	I fatais	D Mean(naoFatais)
Row0	0	0.527
Row1	1	0.394
Row2	2	0.2
Row3	3	4
Row4	4	6

Figure 22: Comparação das fatalidades com os não fatais

Desta tabela podemos reparar que o número de fatalidades nos acidentes varia desde as 0 até às 4. Para além disso podemos reparar que quando o número de fatalidades é o maior (4 fatalidades) o número de não fatais também é o maior e que quando o número de fatalidades são 2 o número de não fatais é o menor.

2.4 Tratamento dos Dados

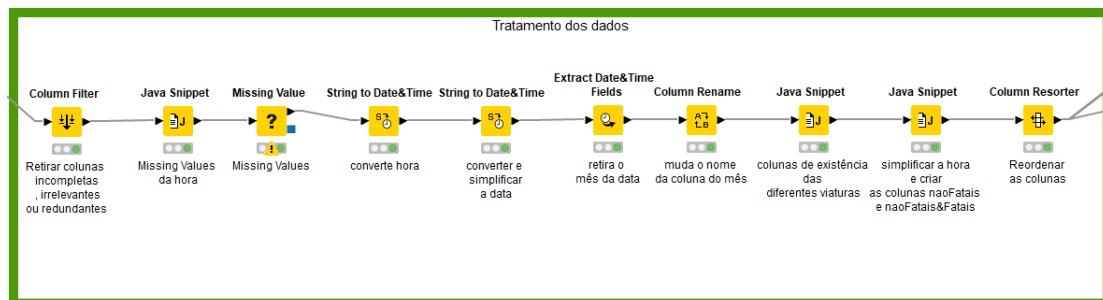


Figure 23: Tratamento dos Dados

2.4.1 Remoção de colunas irrelevantes

Logo no início do tratamento dos dados, procedemos a remover colunas que não são relevantes para o nosso **target**. As colunas removidas foram:

- **idacidente**: esta coluna é simplesmente um id do acidente, logo não contém informação relevante para o estudo que estamos a fazer;
- **data_extracao**: esta coluna apenas contém a data de quando os dados foram guardados na plataforma do **kaggle**, logo é irrelevante para o estudo;

2.4.2 Criação de colunas

Como já foi no caso na exploração dos dados, vamos ter que criar colunas para tentar encontrar novas relações com o nosso **target**.

As colunas que foram adicionadas foram as seguintes:

- **mês**: representa o mês no qual o acidente ocorreu;
- as várias colunas de existência de cada tipo de viatura (**Existe_Auto**, **Existe_Taxi**, etc.): representa o envolvimento de cada tipo de viatura no acidente;
- a coluna **naoFatais**: esta coluna é a soma entre as colunas **feridos** e **feridos_gr**;

- a coluna `naoFatais&Fatais` (**target**): esta coluna é a soma entre as colunas `feridos`, `feridos_gr` e `fatais`, a coluna `fatais` por sua vez é a soma das colunas `mortes` e `morte_post`;

2.4.3 Alteração de colunas

Tal como foi feito antes da Exploração de Dados, convertemos as colunas `data` e `hora` de `String` para o tipo `Date&Time` e, para além disso, simplificamos ambas as colunas retirando à coluna `data` a parte das horas e retirando os minutos à coluna `hora`. Por exemplo, uma data com o valor `2020-10-17 00:00:00` passaria a ficar `2020-10-17` e uma hora com o valor `00:30` passa a ficar `00:00`.

2.4.4 Missing Values

Existiam várias colunas com Missing Values que detetamos na fase de Exploração dos Dados, essas colunas eram: `hora`, `log1`, `log2`, `regiao`, `consorcio`, `predial1`, `longitude` e `latitude`.

Para a coluna `região`, com 3 Missing Values, começamos por tentar derivar estes 3 valores através dos valores das colunas `latitude` e `longitude`. Rapidamente percebemos que todos os registos que continham Missing Values na região também os tinham na latitude e na longitude, ou seja, já não era possível fazer isto. Estando de fora esta hipótese, o grupo achou que o mais indicado seria tratar os Missing Values removendo as 3 linhas já que, primeiro apenas são 3 linhas em 60000 e segundo esta opção é melhor que a de substituir pelo valor mais frequente já que tanto o Norte como o Leste têm quase o mesmo número de ocorrências.

Para as colunas `longitude` e `latitude` a história já é diferente. Primeiro, ambas têm 10403 Missing Values em cerca de 60000 entradas. Segundo, como já temos as colunas de `log1` e de `região` as colunas de `latitude` e `longitude` acrescentam informação que é redundante e que não trás muito mais valor aos dados. Por estes motivos decidimos remover estas colunas.

A coluna `predial1`, com 1192 Missing Values, representa o número do Logradouro onde ocorreu o acidente, ou seja, com a coluna `log1`, esta coluna passa a ter informação redundante.

Para além disto, o grupo decidiu testar duas maneiras de tratar os Missing Values desta coluna: removendo a coluna em si e removendo as linhas com Missing Values. No final de contas os modelos tiveram melhor desempenho quando a coluna era simplesmente removida.

Por estes dois motivos, decidimos remover a coluna `predial1`.

Para a coluna `hora`, com 198 Missing Values, a questão era mais complicada. Esta é uma coluna nominal que tem mais de 1000 unique values e, para além disso, esta coluna apenas pode ser derivada a partir da coluna `noite_dia` o problema é que esta coluna contém ela própria várias possíveis horas.

A melhor opção que o grupo descobriu foi usar a coluna `noite_dia` como base para descobrir o valor da coluna `hora`.




Row ID	 <code>noite_dia</code>	 <code>Min(hora)</code>	 <code>Max(hora)</code>
Row0	DIA	06:00	17:59
Row1	NOITE	00:00	23:59

Figure 24: Resultado do nodo GroupBy

Usando o nodo GroupBy reparamos que o dia é considerado das 6 da manhã às 6 da tarde e que a noite são todas as outras horas. O que decidimos fazer foi dependendo do valor da coluna `noite_dia` substituir os MissingValues da coluna `hora` pela hora com mais ocorrências dentro do intervalo da coluna `noite_dia`.

Row ID	S <code>noite_dia</code>	🕒 Mode(<code>hora</code>)
Row0	DIA	16:00
Row1	NOITE	18:00

Figure 25: Resultado do nodo GroupBy

Com uma pequena exploração reparamos que a hora mais frequente da parte do DIA era as 16:00 da tarde e que a hora mais frequente da parte da NOITE era as 18:00 da noite.

Então, para tratarmos os Missing Values, o que fizemos foi, se o valor da coluna `noite_dia` fosse DIA então o Missing Value era substituído por 16:00 acaso contrário era substituído por 18:00.

Para a coluna `log1`, com 49 missing values em 60000 entradas o que fizemos foi algo parecido com o que fizemos com as horas. Neste caso, usamos a coluna `região` para inferir o valor da coluna `log1`.

Row ID	S <code>regiao</code>	S Mode(<code>log1</code>)
Row0	?	?
Row1	CENTRO	AV IPIRANGA
Row2	LESTE	AV PROTASIO ALVES
Row3	NORTE	AV ASSIS BRASIL
Row4	SUL	AV PROF OSCAR PEREIRA

Figure 26: Resultado do nodo GroupBy

Desta tabela podemos ver qual é a Avenida com mais ocorrências para cada região. O que fizemos foi substituir o Missing Value pelo valor do `log1` correspondente à região. De notar, de que no caso em que a região também era um Missing Value o valor do `log1` foi atribuído o mais frequente (AV PROTASIO ALVES).

A coluna `log2` foi simplesmente removida por três motivos: primeiro, tem 43371 Missing Values em cerca de 60000 entradas, segundo, tem mais de 1000 unique values, terceiro, porque tem pouca relevância com o nosso **target**.

A coluna `consorcio` foi simplesmente removida por dois motivos: primeiro tem 58765 Missing Values em cerca de 60000 entradas e segundo porque tem pouca relevância com o nosso **target**.

2.4.5 Reordenação das colunas

No final, fizemos uma reordenação das colunas apenas para seguir a convenção de que a coluna **target** aparece em último.

2.5 Modelos concebidos

Nesta secção vamos abordar os modelos que o nosso grupo concebeu ao longo do trabalho e, posteriormente, proceder à avaliação dos mesmos.

2.5.1 Regressão

Os modelos de Regressão que o grupo concebeu foram os seguintes:

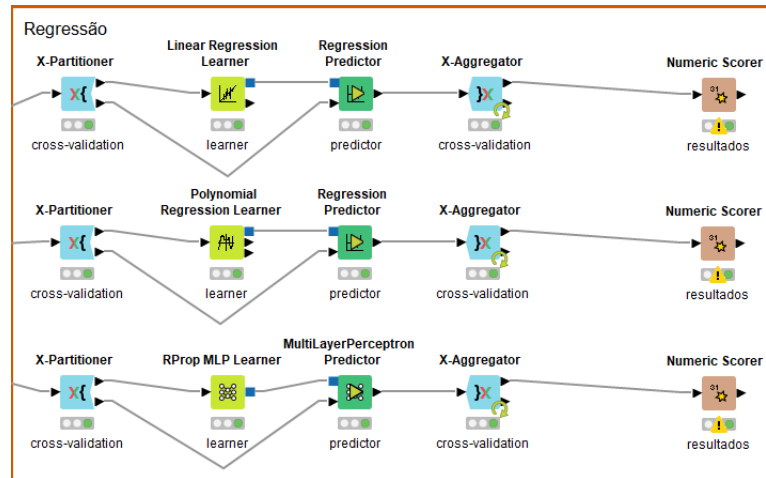


Figure 27: Modelos de Regressão concebidos pelo grupo

O **modelo de regressão linear** tem como objetivo determinar o efeito de várias variáveis independentes numa variável dependente (**naoFatais&Fatais**). As diferentes variáveis são combinadas de forma linear e cada uma tem seu próprio coeficiente de regressão.

O **modelo de regressão polinomial** tem como objetivo determinar o efeito de diversas variáveis independentes numa variável dependente (**naoFatais&Fatais**). As diferentes variáveis são combinadas de forma polinomial e cada uma tem seu coeficiente de regressão.

O **modelo das redes neuronais artificiais (RNA)** usa uma rede com várias camadas, cada uma com vários neurónios, que guardam valores e as ligações entre estes neurónios têm pesos. Este modelo pode fazer várias iterações para realizar o seu processo de aprendizagem.

2.5.2 Classificação

Antes de mais, para gerarmos um problema de classificação iríamos que ter que tornar o nosso **target** numa variável discreta, então, para isso, usamos o nodo **Numeric Binner** para criar intervalos de valores na coluna **naoFatais&Fatais**.

Ao analisar o histograma abaixo, reparamos que a grande parte das ocorrências encontra-se entre as 0 e os 3 ferimentos ou fatalidades.

- Sem fatalidades ou ferimentos:

$$]-\infty; 0]$$

- Fatalidades e ferimentos médios:

$$]0; 3[$$

- Fatalidades e ferimentos altos:

$$[3; +\infty[$$

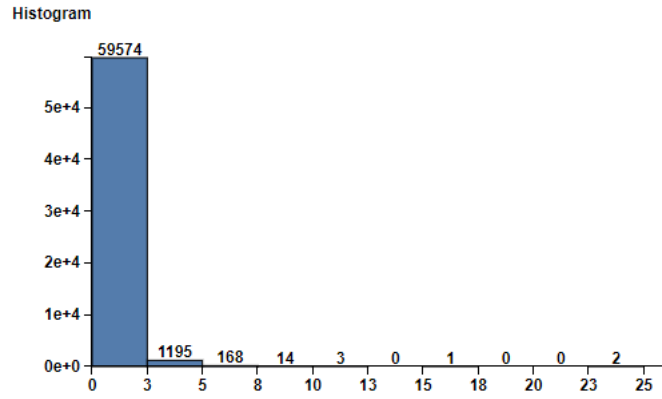


Figure 28: Histograma da coluna naoFatais&Fatais

Com estes 3 intervalos definidos, estamos prontos para inicializar a conceção dos modelos de classificação.

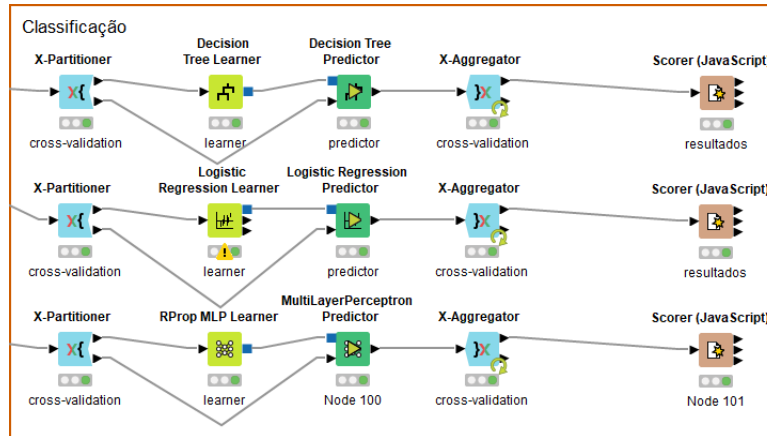


Figure 29: Modelos de Classificação concebidos pelo grupo

O **modelo das árvores de decisão** usa, como é óbvio, árvores de decisão que são grafos hierarquizados em que: cada ramo representa a seleção entre um conjunto de alternativas e cada folha representa uma decisão.

O **modelo de regressão logística** é um pouco parecido ao outros modelos de regressão falados anteriormente (linear e polinomial) mas, na regressão logística, a linha de regressão é substituída por uma curva de regressão logística.

O **modelo das redes neuronais artificiais (RNA)** funciona da mesma maneira que foi explicada anteriormente.

2.6 Análise dos resultados obtidos

Nesta secção vamos passar a analisar os resultados que o grupo obteve dos modelos e proceder à avaliação dos mesmos.

2.6.1 Regressão

Para a avaliação dos modelos de regressão o grupo sabe que temos 3 principais métricas às quais temos que ter atenção, essas métricas são: **MAE** (Mean Absolute Error), **MSE** (Mean Squared Error) e **RMSE** (Root Mean Squared Error).

Sobre estas 3 métricas sabemos ainda que:

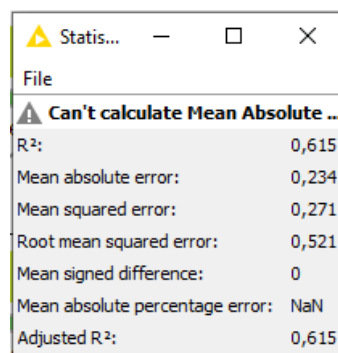
- A **MAE** mede a magnitude média dos erros num conjunto de previsões;
- A **MSE** consiste no cálculo da média das diferenças, ao quadrado, entre os erros num conjunto de previsões;
- A **RMSE** consiste no cálculo da média das diferenças, ao quadrado, entre os erros num conjunto de previsões (todo sobre a raiz quadrada);
- Todas expressam o erro médio de previsão do modelo (valores mais baixos são melhores);
- Todas variam de 0 a ∞ e são indiferentes à direção dos erros.

É ainda importante referir outra métrica, a R^2 , que mede a proporção de variação na variável dependente que é previsível através das variáveis independentes.

Com isto em mente, passemos então aos resultados de cada modelo.

2.6.1.1 Modelo de regressão linear

Para o modelo de regressão linear os resultados foram:



File	
⚠ Can't calculate Mean Absolute ...	
R²:	0,615
Mean absolute error:	0,234
Mean squared error:	0,271
Root mean squared error:	0,521
Mean signed difference:	0
Mean absolute percentage error:	NaN
Adjusted R²:	0,615

Figure 30: Resultados no modelo de regressão linear

Analisando esta imagem podemos reparar que o valor de MAE foi de 0.234, o valor de MSE foi de 0.271 e o valor de RMSE foi de 0.521. De notar que o R^2 teve um valor de 0.615.

A MAE ter um valor de 0.234 significa que o erro médio absoluto entre as previsões e os valores reais foi de 0.234. No contexto em que nos encontra-mos, significa que o modelo teve um erro médio de 0.234 ferimentos/fatalidades previstas comparativamente ao dados reais.

A RMSE tem quase o mesmo significado que a métrica MAE mas como a RMSE dá mais peso a erros maiores, a métrica RMSE acabou por ter um valor maior de 0.521 do que os 0.234 da MAE.

Tendo tudo em conta, estes são resultados bastante satisfatórios já que o modelo não apresentou grandes erros.

2.6.1.2 Modelo de regressão polinomial

Analisando esta imagem podemos reparar que o valor de MAE foi de 0.221, o valor de MSE foi de 0.235 e o valor de RMSE foi de 0.485. De notar que o R^2 teve um valor de 0.666.

Comparativamente com o modelo de regressão linear, este apresentou melhores resultados já que os erros (**MAE** e **RMSE**) foram menores e o R^2 foi maior.

Can't calculate Mean Absolute ...	
R²:	0,666
Mean absolute error:	0,221
Mean squared error:	0,235
Root mean squared error:	0,485
Mean signed difference:	0
Mean absolute percentage error:	NaN
Adjusted R²:	0,666

Figure 31: Resultados no modelo de regressão polinomial

2.6.1.3 Modelo das redes neuronais artificiais

Para o modelo das redes neuronais artificiais os resultados foram:

Can't calculate Mean Absolute ...	
R²:	0,669
Mean absolute error:	0,008
Mean squared error:	0
Root mean squared error:	0,019
Mean signed difference:	0
Mean absolute percentage error:	NaN
Adjusted R²:	0,669

Figure 32: Resultados do modelo das redes neuronais artificiais

Analisando esta imagem podemos reparar que o valor de MAE foi de 0.008, o valor de MSE foi de 0 e o valor de RMSE foi de 0.019. De notar que o R^2 teve um valor de 0.669.

Comparativamente com os outros dois modelos,, este apresentou melhores resultados já que os erros (**MAE** e **RMSE**) foram menores e o R^2 foi maior.

2.6.2 Classificação

O nosso grupo teve a ideia de tentar conceber **curvas ROC**, usando o nodo **ROC Curve**, para melhor avaliar o desempenho dos modelos, mas rapidamente percebemos que não o poderíamos fazer porque, no nosso caso, temos 3 possíveis "outcomes" para o problema (os três possíveis valores da coluna **naoFatais&Fatais**) o que impossibilitava o desenvolvimento da curva ROC.

Com esta hipótese fora da mesa, o que nos restava analisar era os resultados do nodo **Scorer (JavaScript)**.

Este nodo apresenta a **matriz de confusão** e a **accuracy**. Para além destes, ainda podemos inferir, com base na informação deste nodo, a **sensitivity** e a **specificity**.

As fórmulas para a sensitivity e a specificity são:

$$Sensitivity = \frac{TP}{TP + FP}$$

$$Specificity = \frac{TP}{TP + FN}$$

2.6.2.1 Modelo das árvores de decisão

Para o modelo das árvores de decisão os resultados foram:

Scorer View

Confusion Matrix

	Fatalidades e ferimen...	Fatalidades e ferimen...	Sem fatalidades ou fe...	
Fatalidades e ferimen...	0	1383	0	0.00%
Fatalidades e ferimen...	0	21649	1	> 99.99%
Sem fatalidades ou fe...	0	0	37924	100.00%
	undefined	94.00%	> 99.99%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
97.73%	2.27%	0.953	59573	1384

Figure 33: Resultados do modelo das decision trees

Podemos verificar que o modelo teve uma **accuracy** de 97.73% e ainda conseguimos identificar a **matriz de confusão**. Através desta, podemos inferir as outras duas métricas.

Os valores da Sensitivity e Specificity são ambos 0 já que o valor de TP (True Positives) é 0.

O que podemos reparar é que o modelo concebido tem dificuldades a prever fatalidades/ferimentos altos já que todas as 1383 entradas que realmente tinham 3 ou mais ferimentos/fatalidades foram incorretamente previstas. De resto, o modelo prevê quase sem qualquer erro.

2.6.2.2 Modelo da regressão logística

Para o modelo da regressão logística os resultados foram:

Scorer View

Confusion Matrix

	Fatalidades e ferimen...	Fatalidades e ferimen...	Sem fatalidades ou fe...	
Fatalidades e ferimen...	1	1382	0	0.07%
Fatalidades e ferimen...	7	21642	1	99.96%
Sem fatalidades ou fe...	0	0	37924	100.00%
	12.50%	94.00%	> 99.99%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
97.72%	2.28%	0.952	59567	1390

Figure 34: Resultados do modelo da regressão logística

Podemos verificar que o modelo teve uma **accuracy** de 97.72% e ainda conseguimos identificar a **matriz de confusão**. Através desta, podemos inferir as outras duas métricas.

A sensitivity teve um valor de 0.125.

A specificity teve um valor de 0.00072.

O que podemos reparar é que o modelo concebido tem dificuldades a prever fatalidades/ferimentos altos já que apenas uma das as 1383 entradas que realmente tinham 3 ou mais ferimentos/fatalidades foi corretamente prevista. De resto, o modelo prevê quase sem qualquer erro.

2.6.2.3 Modelo das redes neuronais

Para o modelo das redes neuronais os resultados foram:

Scorer View

Confusion Matrix

	Fatalidades e ferimen...	Fatalidades e ferimen...	Sem fatalidades ou fe...	
Fatalidades e ferimen...	1	1382	0	0.07%
Fatalidades e ferimen...	2	21646	2	99.98%
Sem fatalidades ou fe...	0	0	37924	100.00%
	33.33%	94.00%	99.99%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
97.73%	2.27%	0.953	59571	1386

Figure 35: Resultados do modelo das redes neuronais

Podemos verificar que o modelo teve uma **accuracy** de 97.73% e ainda conseguimos identificar a **matriz de confusão**. Através desta, podemos inferir as outras duas métricas.

A sensitivity teve um valor de 0.33333.

A specificity teve um valor de 0.00072.

O que podemos reparar é que o modelo concebido tem dificuldades a prever fatalidades/ferimentos altos já que apenas uma das as 1383 entradas que realmente tinham 3 ou mais ferimentos/fatalidades foi corretamente prevista. De resto, o modelo prevê quase sem qualquer erro.

Podemos concluir que o melhor modelo concebido foi o das redes neuronais (RNA) e que uma abordagem para melhorar os modelos seria melhorar a maneira como o modelo prevê ferimentos/fatalidades altas.

3 Tarefa B

Como já foi referido anteriormente, o dataset que o nosso grupo estudou para a Tarefa B foi o da produção de vestuário, no qual o objetivo era prever a produção do vestuário, ou seja, prever a coluna "**actual_productivity**".

3.1 Descrição do dataset

Nesta secção vamos falar sobre a informação que consta no ficheiro **producao_vestuário.txt**.

Este ficheiro descreve as 14 variáveis (colunas) presentes no dataset em questão das quais 13 são independentes e 1 é dependente. A variável dependente é a coluna **actual_productivity**, que é a coluna target.

Passemos então à descrição de cada variável independente:

- **rowID**: representa o ID da linha;
- **date**: representa a data;
- **department**: o departamento associado com a linha;
- **team**: o número da equipa associada à linha;

-
- **targeted_productivity**: a produtividade objetivo proposta pela autoridade para cada equipa em cada dia;
 - **smv**: o tempo alocado para uma tarefa;
 - **wip**: o número de itens por acabar para produtos;
 - **over_time**: representa a quantidade de tempo extra para cada equipa em minutos;
 - **incentive**: representa a quantidade de incentivo financeiro num curso de ação específico;
 - **idle_time**: representa a quantidade de tempo em que não houve produção devido a várias razões;
 - **idle_men**: representa o número de trabalhadores que estiveram parados devido à interrupção na produção;
 - **no_of_workers**: número de trabalhadores em cada equipa;
 - **no_of_style_change**: número de alterações no estilo de um produto em específico.

Para além destas, temos a única variável dependente:

- **actual_productivity**: A produção efetiva, em percentagem, produzida pelos trabalhadores.

Após a análise desta informação sobre as variáveis ficamos a perceber que o objetivo passa por verificar de que maneira cada uma das variáveis independentes afetam a variável dependente para que posteriormente consigamos desenvolver o melhor modelo possível.

3.2 Exploração e Análise dos Dados

Começamos por utilizar o nodo "**Data Explorer**", já que nos dá uma grande noção sobre o estado dos dados:

- Diz-nos se existem Missing Values ou não.
- Mostra-nos informações estatísticas importantes como: a média, kurtosis e skewness de uma determinada coluna.
- Para colunas nominais, mostra-nos os diferentes valores que a coluna toma o que pode ser útil para, por exemplo, verificar algum possível erro de ortografia.

Com o uso deste nodo reparamos em várias questões que tivemos que ter em consideração na fase do tratamento dos dados:

- **A coluna wip**, tinha 506 NaN, ou seja, tinha 506 Missing Values (neste context, podemos considerar um NaN um Missing Value já que têm o mesmo significado) que mais tarde teríamos que tratar.
- **A coluna department**, tinha apenas 2 departamentos diferentes mas 5 Unique Values porque existiam bastantes erros ortográficos.
- **A coluna date**, estava em formato de string mas poderia ser convertida para Date&Time para posteriormente retirarmos o dia da semana (as razões pelas quais fizemos isto vão ser explicadas na parte do **Tratamento dos Dados**).

-
- Reparamos que a **coluna rowID** é completamente inútil para o estudo que estamos a fazer já que apenas representa o número da linha.
 - Reparamos que a coluna **actual_productivity** (que é a coluna target) tinha valores a cima de 1 (mais de 100% de produtividade), o que é impossível e teria de ser corrigido.

Ainda na exploração e análise dos dados, usamos vários gráficos (especialmente bar charts, histogramas e scatter plots) para percebermos melhor como cada uma das diferentes colunas se relacionavam com a coluna "**actual_productivity**" que era a coluna **target**. Desta exploração, já conseguimos perceber quais as colunas que tinham uma maior correlação com a coluna target, mas vamos abordar este tema mais a fundo na secção **Exploração e análise dos dados - pós tratamento**.

3.3 Tratamento dos Dados

Nesta fase da tarefa, prosseguimos com o tratamento dos dados que tínhamos identificado na fase anterior, ou seja:

- Removemos a coluna rowID.
- Mudamos o formato da coluna date para Date&Time e, posteriormente, extraímos o seu dia da semana.

Decidimos extrair o dia da semana da coluna date porque achavamos que poderia existir alguma correlação entre o dia da semana de uma determinada entrada e a "actual_productivity" alcançada, por exemplo, a princípio o grupo achou que em dias de fim-de-semana a "actual_productivity" seria, previsivelmente, menor. Na secção **Exploração e análise dos dados - pós tratamento** iremos explicar os resultados obtidos desta extração do dia da semana.

- Convertemos todos os NaN da coluna wip para Missing Value e corrigimos os erros ortográficos da coluna department.
- No final, ainda usamos o nodo **Column Resorter** para ordenar as colunas de forma mais consistente.

3.3.1 Valores impossíveis da coluna actual_productivity

Como foi referido na secção anterior, existiam algumas entradas da coluna **actual_productivity** que tinham valores a cima de 1, ou seja mais de 100%.

Mais concretamente, existiam 16 registos com mais de 100% de produtividade. O grupo decidiu remover as linhas já que eram apenas 16 em 1197 linhas.

3.3.2 Missing Values

Ainda tínhamos outro aspeto importante no qual teríamos de meditar: como iríamos resolver os Missing Values da coluna wip?

Um aspeto muito importante ao qual tínhamos que ter atenção era que esta coluna tinha 506 Missing Values num data set com 1197 entradas, ou seja, quase metade do data set não tinha o valor da coluna wip. A partir daí conseguimos perceber que remover as linhas que tinham Missing Values seria uma terrível opção.

Também percebemos que não é possível inferir o valor da coluna wip a partir de mais nenhuma coluna do dataset.

As melhores opções que nos restavam eram: **remover a coluna** ou tratar os Missing Values usando a **média**.

Começamos primeiro então por tratar dos Missing Values substituindo pela média, obtivemos os seguintes resultados:

 wip	<input type="checkbox"/>	7	23122	1159.974	1340.076	1795803.940	13.614
---	--------------------------	---	-------	----------	----------	-------------	--------

Figure 36: Informação do nodo Data Explorer sobre a coluna wip

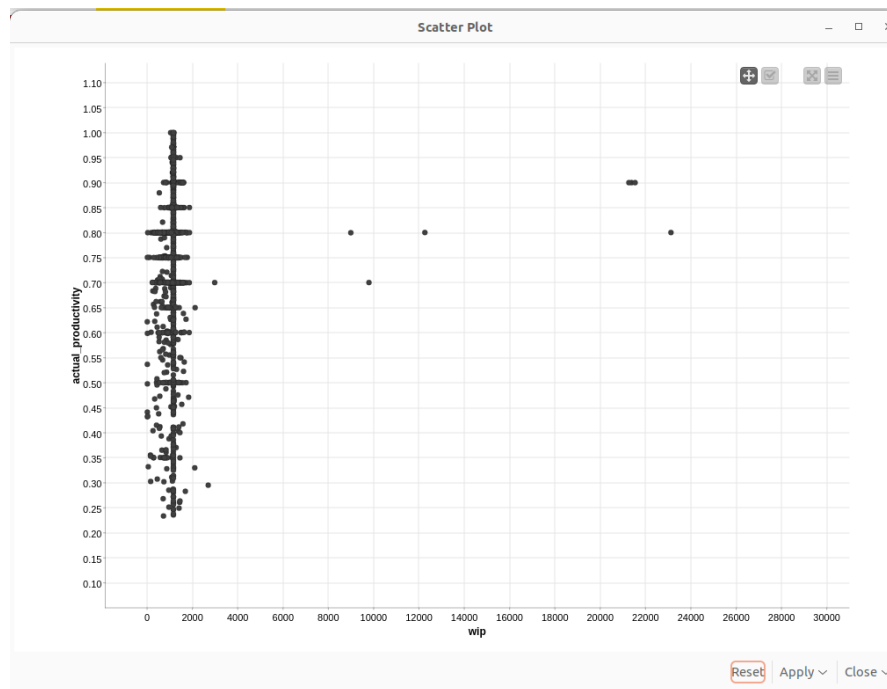


Figure 37: Scatter plot: eixo x-> wip; eixo y-> actual_productivity

Apartir destas imagens, conseguimos perceber que os dados da coluna wip estão todos bastante centrados na média, exceto alguns outliers que podem ser ignorados, ou seja, que a coluna wip tem um desvio padrão não muito alto. Sendo que o desvio padrão da variável wip não é alto, sabemos que substituir os Missing Values da coluna pela média não vai trazer muitos impactos negativos.

Por todas estas razões decidimos tratar os Missing Values da coluna wip usando a média.

3.4 Exploração e Análise dos Dados - pós tratamento

Agora que tínhamos já extraído o dia da semana de cada entrada, podíamos realmente testar se a nossa teoria (que em dias de fim-de-semana a "actual_productivity" seria menor) se verificava.

Os resultados são apresentados em baixo.

A primeira coisa que reparamos foi que não existiam entradas com uma data correspondente a uma sexta-feira. Depois conseguimos perceber que a nossa teoria não se verificava e que, em qualquer dia-da-semana (exceto sexta-feira), a "actual_productivity" não era influenciada e que basicamente era sempre a mesma. Esta conclusão foi reforçada com a consulta do nodo **Rank Correlation** que nos mostrou uma correlação entre a coluna target e a Day of week completamente irrelevante.

Row ID	S Day of week (name)	D Mean(actual_productivity)
Row0	domingo	0.718
Row1	quarta-feira	0.719
Row2	quinta-feira	0.713
Row3	segunda-feira	0.729
Row4	sábado	0.745
Row5	terça-feira	0.733

Figure 38: Informação do nodo Group-By

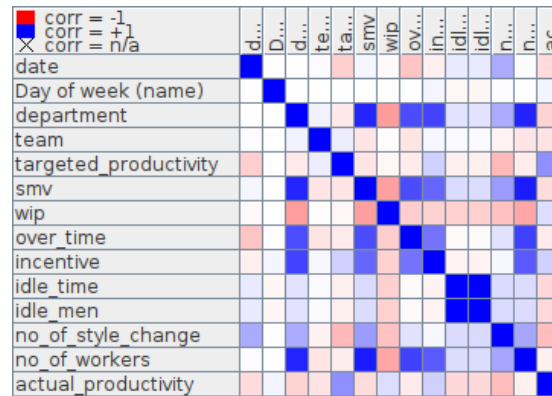


Figure 39: Informação do nodo Rank Correlation

Um dos aspetos mais importantes da exploração e análise de dados - pós tratamento é a correlação entre as diferentes features, mais concretamente, a correlação entre as features independentes e a feature dependente (target).

Analisando a imagem anterior concluímos que existem várias features com nenhum ou irrelevante impacto na target feature, features estas que podem e devem ser removidas dos modelos a desenvolver de seguida. Sendo assim, as features a serem removidas são: date, Day of week (name), department, team, smv, wip, over_time e no_of_workers.

Ficando assim as seguintes features para a concepção dos diferentes modelos: **targeted_productivity**, **incentive**, **idle_time**, **idle_men**, **no_of_style_change**.

De notar que obviamente que, para além das 5 features referidas acima, a feature target (actual_productivity) também não foi removida.

3.5 Modelos concebidos

3.5.1 Regressão

O problema proposto da previsão da coluna **actual_productivity** já era, de sua natureza, um problema de regressão, por isso, não tivemos que proceder a mais nenhum tratamento de dados para tornar o problema num problema de regressão (coisa que não aconteceu nos modelos de classificação).

Os modelos de regressão desenvolvidos para este problema foram:

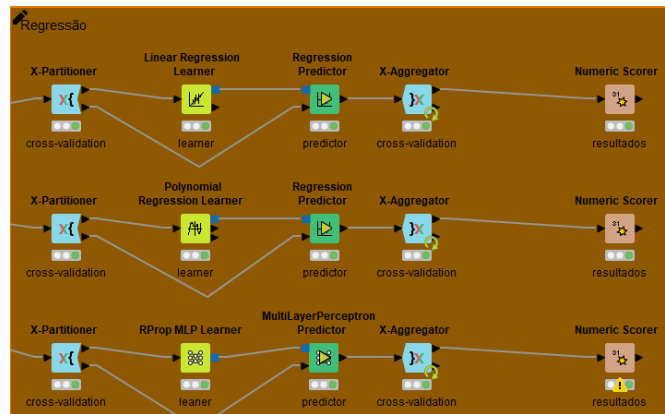


Figure 40: Modelos de Regressão concebidos

3.5.2 Classificação

Como já foi explicado anteriormente, o problema que foi proposto era já um problema de regressão, por isso para conceber modelos de classificação tivemos que proceder a mais um pouco de tratamento de dados.

O que reparamos logo de partida foi que iríamos que ter que realizar alguma técnica de binning para tornar a coluna target numa coluna nominal e assim transformar o problema num problema de classificação. Para isso o grupo começou por analisar o comportamento dos dados da coluna target com mais atenção:

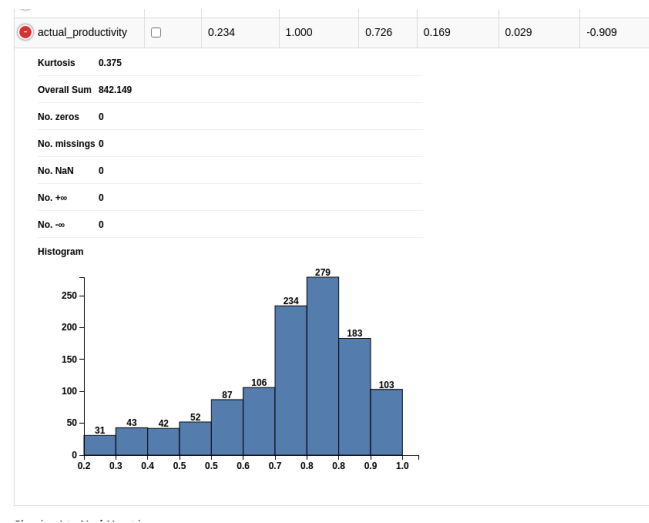


Figure 41: Informação da coluna target no nodo Data Explorer

Rapidamente achamos pertinente definir 3 intervalos para a geração do problema de regressão:

- Baixa produção:

$$]-\infty; 0.6[$$

- Média produção:

$$[0.6; 0.9]$$

- Alta produção:

$$]0.9; +\infty[$$

Todos os intervalos foram criados usando o nodo **Numeric Binner**.

Agora, com estes 3 intervalos definidos podemos realizar modelos de classificação sobre a coluna target.

Posto isto, os modelos de Classificação desenvolvidos foram:

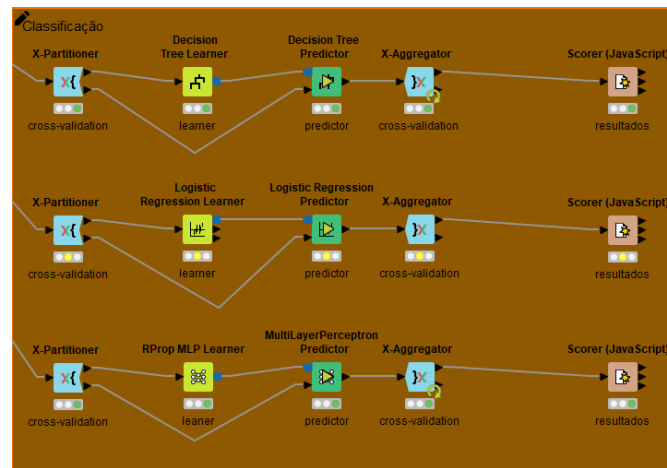


Figure 42: Modelos de Classificação concebidos

3.6 Análise dos resultados obtidos

Nesta secção vamos passar a analisar os resultados que o grupo obteve dos modelos e proceder à avaliação dos mesmos.

3.6.1 Regressão

As métricas usadas para avaliar os modelos de regressão agora na tarefa B são as mesmas que as usadas na tarefa A.

3.6.1.1 Modelo de Regressão Linear

Os resultados foram:

File	
R ² :	0,201
Mean absolute error:	0,107
Mean squared error:	0,023
Root mean squared error:	0,151
Mean signed difference:	0
Mean absolute percentage error:	0,192
Adjusted R ² :	0,201

Figure 43: Informações do nodo Numeric Scorer

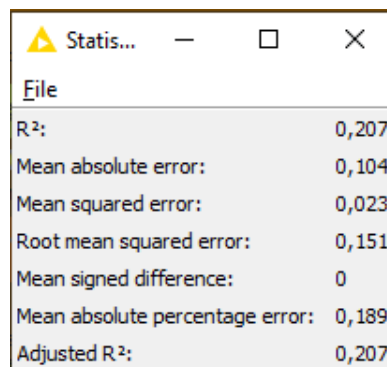
Analisando esta imagem podemos reparar que o valor de **MAE** foi de 0.107, o valor de **MSE** foi de 0.023 e o valor de **RMSE** foi de 0.151. De notar que o R^2 teve um valor de 0.201.

A **MAE** ter um valor de 0.107 significa que o erro médio absoluto entre as previsões e os valores reais foi de 0.107. No contexto em que nos encontra-mos, esta diferença é considerável já que isto significa 10% de erro na previsão da produção real.

A **RMSE** tem quase o mesmo significado que a métrica **MAE** mas como a RMSE dá mais peso a erros maiores a métrica RMSE acabou por ter um valor maior de 0.151 do que os 0.107 da MAE.

3.6.1.2 Modelo de Regressão Polinomial

Os resultados foram:



Statist...	
File	
R ² :	0,207
Mean absolute error:	0,104
Mean squared error:	0,023
Root mean squared error:	0,151
Mean signed difference:	0
Mean absolute percentage error:	0,189
Adjusted R ² :	0,207

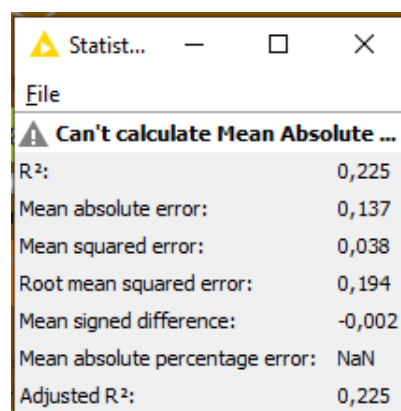
Figure 44: Informações do nodo Numeric Scorer

Analisando esta imagem podemos reparar que o valor de **MAE** foi de 0.104, o valor de **MSE** foi de 0.023 e o valor de **RMSE** foi de 0.151. De notar que o R^2 teve um valor de 0.207.

Comparativamente ao modelo de regressão linear, este modelo apresentou uma performance similiar tanto nos erros (**MAE** e **RMSE**) como no valor do R^2 .

3.6.1.3 Modelo das redes neuronais artificiais (RNA)

Os resultados foram:



Statist...	
File	
⚠ Can't calculate Mean Absolute ...	
R ² :	0,225
Mean absolute error:	0,137
Mean squared error:	0,038
Root mean squared error:	0,194
Mean signed difference:	-0,002
Mean absolute percentage error:	NaN
Adjusted R ² :	0,225

Figure 45: Informações do nodo Numeric Scorer

Analisando esta imagem podemos reparar que o valor de **MAE** foi de 0.137, o valor de **MSE** foi de 0.038 e o valor de **RMSE** foi de 0.194. De notar que o R^2 teve um valor

de 0.225.

Comparativamente ao dois modelos anteriores, este apresentou erros maiores (**MAE** e **RMSE**) tendo apenas apresentado um melhor R^2 .

Concluimos que os dois modelos iniciais (regressão linear e polinomial) foram os que tiveram uma melhor performance.

3.6.2 Classificação

Tal como aconteceu na tarefa A, não é possível usar a curva ROC já que temos 3 classes na coluna da produtividade. As métricas usadas para avaliar os modelos de classificação são as mesmas que as usadas para a tarefa A.

3.6.2.1 Modelo das árvores de decisão

Os resultados foram:

Scorer View

Confusion Matrix

	Alta (Predicted)	Baixa (Predicted)	Média (Predicted)	
Alta (Actual)	41	16	109	24.70%
Baixa (Actual)	2	92	122	42.59%
Média (Actual)	9	46	723	92.93%
	78.85%	59.74%	75.79%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
73.79%	26.21%	0.372	856	304

Figure 46: Informações do nodo Scorer (JavaScript)

Podemos verificar que o modelo teve uma **accuracy** de 73.79% e ainda conseguimos identificar a **matriz de confusão**. Através desta, podemos inferir as outras duas métricas.

A sensitivity teve um valor de 0.7885.

A specificity teve um valor de 0.2470.

O que podemos retirar mais desta imagem é que o modelo concebido tem algumas dificuldades a prever produtividades baixas e muitas dificuldades a prever produtividades altas.

3.6.2.2 Modelo de Regressão Logística

Os resultados foram:

Scorer View

Confusion Matrix

	Alta (Predicted)	Baixa (Predicted)	Média (Predicted)	
Alta (Actual)	0	7	159	0.00%
Baixa (Actual)	0	57	159	26.39%
Média (Actual)	0	20	758	97.43%
	undefined	67.86%	70.45%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
70.26%	29.74%	0.184	815	345

Figure 47: Informações do nodo Scorer (JavaScript)

Podemos verificar que o modelo teve uma **accuracy** de 70.26% e ainda conseguimos identificar a **matriz de confusão**. Através desta, podemos inferir as outras duas métricas.

Como o modelo concebido não preveu nenhuma produção alta, a sensitivity e a specificity têm o valor de 0.

Podemos ainda reparar que o modelo não acertou em nenhuma previsão de produção alta e teve muitas dificuldades a prever produções baixas.

Comparativamente ao modelo anterior, este apresentou uma performance pior.

3.6.2.3 Modelo das redes neuronais artificiais (RNA)

Os resultados foram:

Scorer View

Confusion Matrix

	Alta (Predicted)	Baixa (Predicted)	Média (Predicted)	
Alta (Actual)	9	14	143	5.42%
Baixa (Actual)	2	74	140	34.26%
Média (Actual)	6	41	731	93.96%
	52.94%	57.36%	72.09%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's Kappa (κ)	Correctly Classified	Incorrectly Classified
70.17%	29.83%	0.237	814	346

Figure 48: Informações do nodo Scorer (JavaScript)

Podemos verificar que o modelo teve uma **accuracy** de 70.17% e ainda conseguimos identificar a **matriz de confusão**. Através desta, podemos inferir as outras duas métricas.

A sensitivity teve um valor de 0.5294.

A specificity teve um valor de 0.0542.

Podemos ainda reparar que o modelo teve uma taxa de sucesso muito baixa ao prever produções altas e baixas.

Comparativamente aos dois modelos anteriores, este teve uma performance similar ao modelo da regressão logística e uma pior performance que o modelo das árvores de decisão.

Conclui-se então que o modelo que teve uma melhor performance foi o modelo das árvores de decisão.

4 Conclusão

Com este trabalho prático tivemos a oportunidade de testar os nossos conhecimentos em diversas áreas desde a estatística até à informática. Mais concretamente, aprendemos na prática a como construir um *workflow* na plataforma **KNIME** de maneira correta de início ao fim fazendo exploração, análise e preparação de dados e, no final, concebendo modelos o mais adequadamente possíveis ao problema que temos em mãos.

Concluindo, este trabalho foi uma ótima maneira para nos introduzir ao mundo do **Machine Learning**.