Análise de Dados em Informática - Pandemia COVID-19 - Trabalho Prático 2

André Novo
3DE
ISEP
1181628@isep.ipp.pt

Diogo Ribeiro
3DE
ISEP
1180782@isep.ipp.pt

Tiago Ribeiro
3DE
ISEP
1181444@isep.ipp.pt

***Resumo*—Este artigo científico documenta processos de análise de vários indicadores relativos a um conjunto de dados fornecido, através de modelos de regressão e classificação, e fazendo uso de algoritmos de aprendizagem automática. O artigo é constituído por uma breve introdução, onde são descritos os objetivos do trabalho, o seu propósito e metodologia adotada, seguida de uma revisão de literatura referente aos algoritmos de aprendizagem automática relevantes para a realização do mesmo trabalho. Depois, são expostos um conjunto de problemas que foram propostos para regressão e classificação, assim como propostas de resolução aos mesmos, com discussão e análise dos resultados e breves conclusões que podem ser retiradas. Por último, são apresentadas conclusões gerais do trabalho desenvolvido.**

***Palavras-chave*—aprendizagem automática, modelos de regressão/classificação, regressão linear, árvores de regressão/decisão, k-vizinhos-mais-próximos, redes neuronais**

I. Introdução

Atualmente, o mundo vive sob a ação de uma doença infeciosa que ainda não foi erradicada, o COVID-19, que mudou por completo as nossas rotinas e que, infelizmente, tem sido a causadora de milhares de mortes. Partindo deste impacto, podem ser recolhidos diversos dados referentes à pandemia, provenientes de vários países e continentes a nível mundial, dados esses que podem ser usados em diversos estudos, por forma a entender melhor o impacto da mesma e, até mesmo, servirem como base para encontrar uma forma de erradicar a doença.

O presente artigo científico tem como finalidade a realização de uma análise de vários indicadores, como número total de casos, número médio de novos casos, número total de mortos, taxa de transmissibilidade R(t) e outros relativos a cada país, sobre um conjunto de dados reais relativos à pandemia COVID-19 a nível mundial. Esse conjunto de dados encontra-se num ficheiro que foi facultado para ser utilizado no desenvolvimento do trabalho. Os dados presentes nesse mesmo ficheiro foram retirados da base de dados internacional "Our World in Data" [1], dinamizada pela universidade John Hopkins University (JHU).

A análise dos dados é feita através de modelos de classificação/regressão, utilizando os seguintes algoritmos de aprendizagem automática: regressão linear, árvores de regressão/decisão, k-vizinhos-mais-próximos e redes neuronais.

O desenvolvimento dos processos acima mencionados é feito através da resolução de um conjunto de exercícios fornecidos pelos docentes desta unidade curricular, sendo, por isso, o propósito deste artigo puramente académico. Os resultados obtidos são, portanto, analisados e discutidos, retirando-se conclusões acerca dos mesmos.

Por forma a obter os resultados pretendidos, é feito um uso à linguagem de alto nível R, que possui um ambiente vocacionado para a análise de dados e para a componente gráfica [2].

II. Revisão da Literatura

A. Cross-Validation e métricas de avaliação dos modelos de aprendizagem automática

O algoritmo do *Cross-validation* consiste na reserva de uma pequena amostra dos dados a serem utilizados, seguida da construção (ou treino) do modelo, utilizando a parte restante dos dados e, por último, testa a eficácia do modelo utilizando os dados reservados. Se o modelo funcionar bem com esse conjunto, então é adequado [3].

Existem diversos métodos de *Cross-validation*, nomeadamente os seguintes:

O método *Holdout* consiste em dividir os dados em 70-30 de forma aleatória. Sendo 70% dos dados para treino e 30% para teste. Uma desvantagem de se usar este procedimento é que podemos selecionar uma porção de dados de treino e teste que são muito parecidas e consequentemente obter uma boa avaliação do modelo nesse caso, contudo quando colocamos o modelo a funcionar com novos dados muito diferentes dos dados já conhecidos pelo mesmo, este gera-nos resultados péssimos [4].

Um outro procedimento de teste é o *k-fold-cross-validation* que tem como vantagem evitar o problema da aleatoriedade dos dados do *Holdout*, uma vez que através deste procedimento conseguimos treinar/testar o modelo com todos os dados disponíveis. A desvantagem porém, é caso o conjunto de dados seja de elevada dimensão, o custo computacional para efetuar o *k-fold-cross-validation* será também bastante elevado [4].

A avaliação do desempenho dos modelos de regressão é baseada em determinar a taxa de acerto do modelo na previsão dos resultados para observações nunca antes vistas, ou seja, que não sejam utilizadas na construção do modelo. As métricas estatísticas para a quantificação da qualidade/desempenho global dos modelos de regressão incluem [3]:

Root Mean Squared Error (RMSE): É a raíz quadrada da média das diferenças entre os valores conhecidos observados e os valores previstos pelo modelo. Quanto menor o RMSE, melhor será o desempenho do modelo [3]. A fórmula utilizada para o cálculo do RMSE encontra-se descrita na figura 1.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Figura 1. Fórmula para calcular o RMSE.

Mean Absolute Error (MAE): Uma alternava ao RMSE que é menos sensível a *outliers*. É a soma das diferenças absolutas entre os valores previstos e os valores reais. Quanto menor o MAE, melhor será o desempenho do modelo [3]. A fórmula utilizada para o cálculo do MAE encontra-se descrita na figura 2.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Figura 2. Fórmula para calcular o MAE.

A avaliação do desempenho dos modelos de classificação pode ser feita através de uma matriz de confusão. É uma tabela que, de certa forma, sumariza os resultados previstos num problema de classificação. A mesma apresenta os seguintes campos [5]:

True positive (TP) → Acontece quando no conjunto real, a classe que pretendemos prever foi prevista corretamente.

False positive (FP) → Aconteceu quando no conjunto real, a classe que pretendemos prever foi prevista incorretamente.

True negative (TN) → Acontece quando no conjunto real, a classe que não pretendemos prever foi prevista corretamente.

False negative (FN) → Acontece quando no conjunto real, a classe que não pretendemos prever foi prevista incorretamente.

As métricas de avaliação que podem ser utilizadas são as seguintes [3]:

Accuracy → É a habilidade de um classificador binário detetar com precisão tanto os positivos, como os negativos.

Precision → Informa a proporção dos positivos que realmente são positivos.

Sensivity/Recall → Informa a proporção dos positivos que realmente está bem classificada.

Specificity → É a habilidade de um classificador binário detetar *true negatives.*

F1 → É um número entre 0 e 1 e representa a média harmónica da *precision* e do *recall*

Na figura 3, encontram-se as fórmulas para o cálculo das métricas supracitadas.

Metric	
Accuracy	$\text{Accuracy} = \frac{FP + FN}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
Precision (P)	$\text{Precision} = \frac{TP}{\text{TP} + \text{FP}}$
Sensitivity/Recall(R)	$\text{Sensitivity/recall} = \frac{TP}{\text{TP} + \text{FN}}$
Specificity	$\text{Specificity} = \frac{TN}{\text{TN} + \text{FP}}$
F1	$\text{F1} = \frac{2 * P * R}{\text{P} + \text{R}}$

Figura 3. Fórmulas para as métricas de avaliação de modelos de classificação.

B. Regressão Linear

Podemos afirmar que a análise de uma regressão linear é usada para explicar a relação entre uma variável aleatória e um (possível) conjunto de variáveis não aleatórias. Dizemos que a variável aleatória é dependente e aquelas que pertencem ao conjunto das variáveis não aleatórias, independentes. Ora, quando temos apenas uma variável independente, a regressão linear em causa assume a nomenclatura de simples, em contrapartida caso haja mais que uma variável preditora (não aleatória), estamos na presença de uma regressão múltipla. Ora, acrescentando a isto, a variável aleatória deve ser sempre contínua enquanto que as variaveis nao aleatórias podem ser contínuas, discretas ou categóricas.

Podemos representar um modelo de regressão simples pela seguinte fórmula:

$$Y = \beta 0 + \beta 1 X + \epsilon \quad (1)$$

Onde $\beta 0$ representa a ordenada na origem, $\beta 1$ o declive da reta (coeficientes) e a variável ϵ representa as flutuações aleatorias causadas por erros nas medições dos dados ou por outros factores externos [6].

C. Árvore de Regressão/Decisão

Podemos definir uma árvore de decisão como sendo um tipo de algoritmo de aprendizagem supervisionado que pode ser usado tanto em problemas no âmbito da regressão como da classificação [7].

Uma árvore de decisão consiste, basicamente, num conjunto de nós de decisão, conectados por ramos. Começando no nó raiz, que por convenção é colocado no topo do diagrama da árvore de decisão, as variáveis são testadas nos nós de decisão, com cada resultado possível resultando daí uma ramificação. Cada ramo, então, leva a outro nó de decisão ou a um nó terminal [8].

D. k-vizinhos-mais-próximos

Podemos definir o algoritmo kNN como sendo um algoritmo de aprendizagem supervisionada que de certa forma, basicamente, limita-se a armazenar os dados resultantes durante a fase de treinamento. Por esta mesma razão, o kNN é também designado de "algoritmo de aprendizagem preguiçoso", uma vez que o processamento dos dados resultantes dos exemplos de teste é adiado até que se façam previsões novamente.

O algoritmo encontra os k-vizinhos-mais-próximos de um determinado ponto (ponto de consulta) e a partir daí, calcula a *label class* (no caso de ser classificação) ou o *continuous target* (no caso da regressão), tendo por base os k pontos mais próximos/similares. A ideia geral consiste em invés de aproximar a função alvo globalmente, em cada previsão, o kNN aproxima a função alvo localmente. Na prática, é mais fácil aprender a aproximar uma função localmente do que globalmente [9].

E. Redes Neuronais

As redes neurais representam são uma tentativa muito básica de imitar o processo de aprendizagem não linear que acontece nas redes de neurónios da própria natureza, como as dos humanos, por exemplo [10].

III. Regressão

Todos os exercícios resolvidos para este tópico utilizam uma *seed* de 123.

A. Exercício 1

Para este exercício, foi pedido o seguinte: *"Comece por carregar o ficheiro ("countryagregatedata.xlsx") para o ambiente do R, verifique a sua dimensão e obtenha um sumário dos dados."*

Procedeu-se, numa primeira fase, à exportação dos dados presentes no ficheiro para uma estrutura de dados. Por motivos de entrega do trabalho, foi-nos pedido o mesmo ficheiro em formato CSV, e, portanto, optou-se por carregar o ficheiro nesse formato, e não no formato xlsx.

Com o uso da função *head*, visualizou-se as primeiras 4 linhas da estrutura, e denotou-se que a coluna "row"não apresenta valores relevantes para o trabalho em si. Por esse motivo, a mesma foi descartada da estrutura.

Em seguida, recorrendo à função *dim*, verificou-se a dimensão dessa mesma estrutura, composta por 209 linhas e 24 colunas.

Após esse passo, procedeu-se à obtenção do sumário dos dados, através da função *summary*. A partir do sumário obtido, pode-se verificar, para os dados numéricos, diversos dados relativos aos mesmos, nomeadamente o mínimo, 1º quartil, mediana (ou 2º quartil), média, 3º quartil e máximo. Já para os não numéricos, pode-se observar o tamanho dos mesmos, a classe e o modo.

B. Exercício 2

Este exercício pede o seguinte: *"Crie um diagrama de correlação entre todos os atributos e comente o que se observa."*

Para que 2 ou mais atributos possam ser correlacionados, é necessário que todos sejam numéricos. Os dados exportados contêm 2 colunas não numéricas, no caso, as colunas *location* e *continent*, que, consequentemente, não poderão ser utilizadas para a construção do diagrama. Por isso, foi criada uma nova estrutura com a ausência dessas mesmas colunas.

Com essa nova estrutura, num primeiro momento, fez-se a correlação entre todos os atributos da mesma, com recurso à função cor. O parâmetro *method* foi omitido, sendo utilizado o método por defeito, que, no caso, corresponde ao *pearson*, o que faz sentido, uma vez que considera-se que os dados de todos os atributos são contínuos, sendo que, no caso dos outros 2 métodos (*spearman* e *kendall*), pelo menos um dos dados deve ser ordinal [11].

Como os valores possuem bastantes casas decimais, fez-se um arredondamento de 3 casas decimais para cada um, usando a função *round*.

Por último, procedeu-se à criação do diagrama de correlação, a partir da função *corrplot*, da biblioteca *corrplot*. O diagrama obtido encontra-se descrito na figura 4.

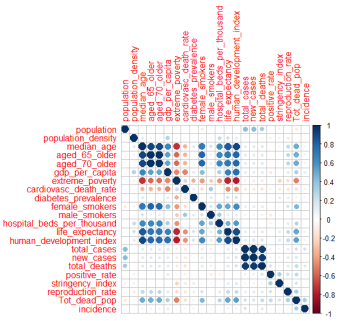

Figura 4. Diagrama de Correlação.

A partir do diagrama, podemos constatar que que os círculos a azul correspondem a correlações positivas, enquanto que os vermelhos correspondem a correlações negativas. Quando mais intensa for a cor, mais fortemente correlacionados os atributos estão, ou seja, no caso das positivas, mais próximos estão de 1 e, no caso das negativas, de -1. Já os brancos correspondem a correlações cujo coeficiente é igual ou muito próximo de 0, indicando uma fraca correlação entre atributos com essa cor. Todos os círculos na diagnonal correspondem a atributos fortemente correlacionados, cujo coeficiente de correlação é igual a 1, uma vez que é verificada a correlação entre atributos iguais.

C. Exercício 3

Neste exercício é pedido o seguinte: *"Obtenha um modelo de regressão linear simples para a variável objetivo para determinar o "total_deaths"usando o número de novos casos ("new_cases")."*

De acordo com o enunciado supracitado, a variável objetivo corresponde ao "total_deaths".

Por forma a construirmos e avaliarmos o modelo que será obtido, utilizou-se o método Holdout. Ou seja, partindo os dados originais, dividiu-se, de forma aleatória, os mesmos em 2 conjuntos, sendo que 70% desses dados correspondem aos dados de treino e 30% aos de teste.

Em seguida, construiu-se o modelo de regressão linear simples, usando como variável dependente ou objetivo o "total_deaths", e como variável independente o "new_cases", usando os dados de treino.

O exercício possui 3 alíneas, sendo que na primeira alínea é pedido o seguinte: *"Apresente a função linear resultante."*

Partindo do modelo obtido, obteve-se um sumário do mesmo. A partir do sumário, podemos constatar os coeficientes da reta de regressão. No caso, $\beta 0$ corresponde a 639,97 e $\beta 1$ corresponde a 3,18. Com estes valores, a função linear resultante é:

total_deaths = 639,97 + 3,18 * new_cases

A segunda alínea pede o seguinte: *"Visualize a reta correspondente ao modelo de regressão linear simples e o respetivo diagrama de dispersão."*

Para a construção do diagrama de dispersão, considerou-se como variável no eixo do x o "total_deaths"e no eixo do y o "new_cases".

Por fim, adicionou-se a reta correspondente ao modelo, recorrendo à função abline. O diagrama de dispersão, com a reta, encontra-se representado na figura 5.

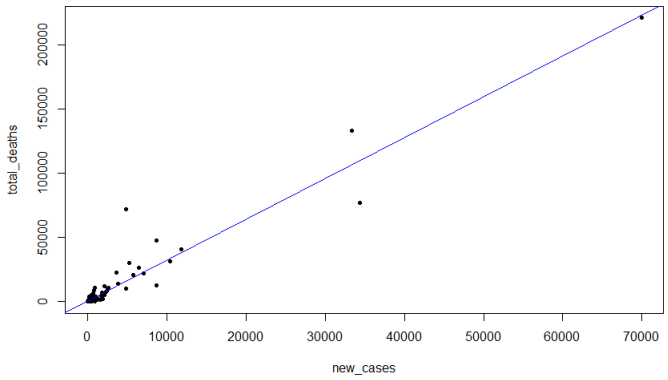

Figura 5. Diagrama de Dispersão, com reta correspondente ao modelo de regressão linear.

Com o diagrama apresentado, conseguimos visualizar a relação entre os 2 atributos.

Por último, última alínea pede o seguinte: *"Calcule o erro médio absoluto (MAE) e raiz quadrada do erro médio (RMSE) do modelo sobre os 30% casos de teste."*

Num primeiro passo, o modelo é avaliado usando os dados de teste, recorrendo à função *predict* para prever, usando esses mesmos dados e o próprio modelo, para que o mesmo tente prever o *total_deaths*.

Após ter sido criada a estrutura com os dados que foram previstos, é feita uma comparação destes com os originais dos dados de teste, através do cálculo do erro médio absoluto (MAE) e da raiz quadrada do erro médio (RMSE) do modelo.

Para esses cálculos, criou-se 2 funções com as fórmulas mencionadas na secção II deste artigo. Após os cálculos, obteve-se os valores presentes na tabela 1.

Tabela I
TABELA COM OS VALORES DO MAE E RMSE

MAE	RMSE
2317.264	5394.300

D. Exercício 4

O enunciado deste exercício pede o seguinte: *"Tendo em conta o conjunto de dados apresentado, pretende-se prever a esperança de vida ('life_expectancy'), aplicando: a) Regressão linear múltipla. b) Árvore de regressão, usando a função rpart. Apresente a árvore de regressão obtida. c) Rede neuronal usando a função neuralnet, fazendo variar os parâmetros. Apresente a rede obtida. Compare os resultados obtidos pelos modelos referidos na questão 4, usando o erro médio absoluto (MAE) e raiz quadrada do erro médio (RMSE). Justifique se os resultados obtidos para os dois melhores modelos são estatisticamente significativos (para um nível de significância de 5%). Identifique o modelo que apresenta o melhor desempenho. "*

O exercício foi dividido em 2 partes: numa primeira parte, recorrendo ao método Holdout, aplicou-se os 3 algoritmos enunciados aos conjuntos de dados obtidos a partir desse método e concluiu-se qual o algoritmo que apresentou pior desempenho, a partir dos valores do MAE e RMSE. Numa segunda parte, utilizando os 2 melhores modelos, recorreu-se ao uso do método k-fold-cross-validation, por forma a obter valores médios do MAE e do RMSE, para, posteriormente, serem comparados.

Relativamente à primeira parte, no início, realizou-se uma normalização *minimax* dos dados, com exclusão das colunas "continent"e "location". Os dados foram normalizados devido ao facto dos atributos terem escalas diferentes entre si, sendo que essa diferença afeta o desempenho desses modelos [12].

Com esses dados, aplicou-se o método Holdout, gerando 2 conjuntos, um de treino e outro de teste. Com estes conjuntos, aplicou-se dos 3 algoritmos pedidos no enunciado. Para cada um deles, criou-se o modelo, seguido das previsões do mesmo. Por forma a obtermos os valores do MAE e RMSE corretos, os dados das previsões foram desnormalizados.

De acordo com os resultados obtidos no modelo de regressão linear múltipla, podemos aferir, com o valor de p-value para cada um dos atributos previsores, a probabilidade do mesmo ser ou não um previsor significativo para o modelo.

Relativamente à contrução da rede neuronal, foram testadas 3 configurações de rede, configurações essas retiradas de uma ficha de exercícios realizada nas aulas (TP7): rede com 1 nó no nível interno, com 3 nós no nível interno e com 2 níveis internos, com 6 e 2 nós, respetivamente. A configuração que apresentou melhor desempenho foi a primeira.

A árvore de regressão obtida e a rede obtida encontram-se descritas nas figuras 6 e 7, respetivamente.

Por fim, foram obtidos os valores de MAE e RMSE, que se encontram descritos na tabela 2.

Tabela II
TABELA COM VALORES DO MAE E RMSE PARA 3 MODELOS.

Modelo	MAE	RMSE
Regressão linear múltipla	5.416697	24.494985
Árvore de regressão	2.907016	3.828154
Rede neuronal	2.816419	4.663553

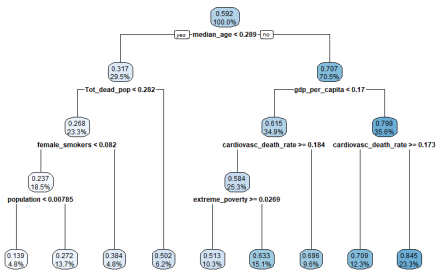

Figura 6. Árvore de regressão obtida.

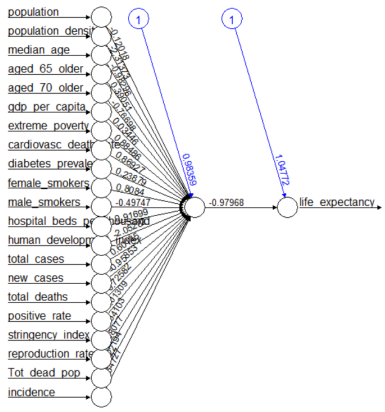

Figura 7. Rede neuronal obtida.

Como podemos constatar, o modelo com pior desempenho é o da regressão linear múltipla.

Na segunda parte, recorrendo ao método k-fold, com k igual a 10, realizou-se 10 iterações com os 2 melhores modelos, tendo-se obtido, no final, uma lista com os valores de MAE e RMSE calculados.

Por forma a verificármos se existem diferenças significativas ou não, recorreu-se ao uso da função t.test.Para a realização do mesmo (1 teste para o MAE e outro para o RMSE), foram assumidas as seguintes hipóteses, tratando-se de um teste bilateral, no caso:

H0: Não existem diferenças significativas no desempenho dos 2 modelos

H1: Existem diferenças significativas no desempenho dos 2 modelos

De acordo com os resultados do teste, obtivemos um p-value de 0.2256 (MAE) e 0.5118 (RMSE). Como ambos os valores são superiores a 0.05, não se rejeita H0. Logo, podem existir ou não diferenças significativas no desempenho dos modelos. No entanto, de acordo com os valores obtidos com o método k-fold (descritos na tabela 3) a rede neuronal apresenta os melhores, sendo, por isso, considerado o modelo com melhor desempenho.

IV. Classificação

Todos os exercícios resolvidos para este tópico utilizam uma seed de 123.

Tabela III
TABELA COM VALORES MÉDIOS DO MAE E RMSE PARA 2 MODELOS

Modelo	MAE	RMSE
Árvore de regressão	2.929790	3.877188
Rede neuronal	2.527784	3.465363

A. Exercício 5

O enunciado deste exercício pede o seguinte: *"Derive um novo atributo NiveldeRisco, discretizando o atributo "stringency_index" em 2 classes: "low" e "high" usando como valor de corte a média do atributo."*

Primeiramente, foi criado um novo vetor, vetor esse que irá conter os dados referentes ao novo atributo NiveldeRisco, que contém 2 valores possíveis: *high* e *low*. Estes valores são escolhidos a partir dos valores do atributo *stringency_index*. Como é dito que o valor de corte para escolher os valores do NiveldeRisco é a média desse mesmo atributo, e recorrendo à função *ifelse*, aplicou-se a seguinte restrição:

Para cada linha, se o valor do atributo *stringency_index* for superior à média dos valores desse mesmo atributo, o valor do novo atributo é *high*, senão é *low*.

Com o vetor preenchido, criou-se uma nova coluna na estrutura dos dados originais e na estrutura dos dados normalizados, denominada *NiveldeRisco*, coluna essa que contém os dados do vetor.

Recorrendo à função *table*, visualizou-se o nº de ocorrências dos valores *high* e *low* da coluna *NiveldeRisco*, números esses que podem ser visualizados na tabela 4.

Tabela IV
TABELA COM O Nº DE OCORRÊNCIAS DOS VALORES *high* E *low*

high	low
118	91

Por último, procedeu-se á remoção da coluna *stringency_index* da estrutura de dados originais e da estrutura dos dados normalizados.

B. Exercício 6

No enunciado do exercício, é pedido o seguinte: *"Estude a capacidade preditiva relativamente a este novo atributo NiveldeRisco usando os seguintes métodos: a) árvore de decisão; b) rede neuronal; c) K-vizinhos-mais-próximos. Usando o método k-fold cross validation obtenha a média e o desvio padrão da taxa de acerto da previsão do atributo NiveldeRisco com os dois melhores modelos. Verifique se existe diferença significativa no desempenho dos dois melhores modelos obtidos anteriormente (use um nível de significância de 5%). Identifique o modelo que apresenta o melhor desempenho."*

Recorrendo aos dados normalizados obtidos no exercício 4, foi utilizado o método *Holdout* para dividir esses mesmos dados em 2 conjuntos: o conjunto de treino, correspondente a 70% dos dados normalizados, e o conjunto de teste, correspondente a 30% desses mesmos dados. Para cada conjunto,

foi removida a coluna *NiveldeRisco*, por não possuir valores numéricos. No entanto, os valores dessas colunas foram guardados em 2 outras estruturas. Este passo foi realizado para que o seguinte pudesse ser cumprido.

De seguida, aplicou-se o algoritmo K-vizinhos-mais-próximos, por forma a prever o atributo *NiveldeRisco*, fazendo uso dos valores ímpares de K no intervalo (1..50), sendo que, para cada k, recolheu-se a taxa de acerto da previsão. O propósito deste passo foi encontrar o valor de k que maximiza a taxa de acerto, valor esse que será utilizado no método *k-fold*, uma vez que não é fornecido nenhum valor para o k, então optou-se por utilizar este método, método este baseado numa ficha de trabalho realizada nas aulas (TP8). O valor de k obtido foi 35.

Após a obtenção do k a ser utilizado para o K-vizinhos-mais-próximos, procedeu-se à aplicação do método *k-fold*. O valor de k escolhido para esse método foi 10, originando 10 folds. Para o algoritmo rede neuronal, foi escolhida a configuração de rede de 1 nível interno.

Para cada iteração do ciclo, gerou-se os dados de treino e de teste com base nos folds criados anteriormente e nos dados normalizados. Para cada um dos modelos pedidos no enunciado, fez-se a criação do mesmo, seguida do cálculo dos valores previstos. Com os valores atuais e os previstos, criou-se a matriz de confusão, sendo que os atuais encontravam-se nas linhas e os previsores nas colunas. Com a matriz criada, procedeu-se, enfim, ao cálculo da taxa de acerto de previsão do modelo. Por fim, o valor é guardado num vetor.

Com os valores da taxa de acerto dos 3 modelos para as 10 iterações, foi obtido um conjunto de valores da taxa de acerto, para cada modelo. A partir dos valores obtidos, calculou-se a média e desvio padrão das taxas de acerto dos 3 algoritmos, com uso da função *apply*, obtendo-se os valores descritos nas tabela 5 e 6, respetivamente.

Tabela V
TABELA COM MÉDIAS DA TAXA DE ACERTO DOS 3 ALGORITMOS.

Árvore de decisão	Rede neuronal	K-vizinhos-mais-próximos
0,592	0,674	0,652

Tabela VI
TABELA COM DESVIOS PADRÕES DA TAXA DE ACERTO DOS 3 ALGORITMOS.

Árvore de decisão	Rede neuronal	K-vizinhos-mais-próximos
0,132	0,111	0,170

De acordo com os resultados, podemos concluir que o modelo com o pior desempenho é a Árvore de decisão, que apresenta, em média, a taxa de acerto mais baixa.

Por forma a verificármos se existem diferenças significativas no desempenho dos 2 melhores modelos, procedeu-se à realização de um teste para médias, utilizando-se a função t.test. Para a realização do mesmo, foram assumidas as seguintes hipóteses, tratando-se de um teste bilateral, no caso:

H0: Não existem diferenças significativas no desempenho dos 2 modelos

H1: Existem diferenças significativas no desempenho dos 2 modelos

De acordo com os resultados do teste, p-value assumiu o valor de 0.3908. Como o valor é superior ao nível de significância de 5%, não se rejeita H0. Logo, não podemos concluir se existem diferenças significativas no desempenho dos 2 modelos. No entanto, de acordo com os valores de média e desvio padrão de ambos, a rede neuronal apresenta os melhores, sendo, por isso, considerado o modelo com melhor desempenho.

C. Exercício 7

O presente exercício pede o seguinte: *"Derive um novo atributo ClassedeRisco, discretizando o atributo "reproduction_rate" Taxa de Transmissibilidade R(t) e o atributo "incidence" Incidência em 3 classes (conforme Figura 1): "Vermelho", "Amarelo" e "Verde". Por simplificação considera-se que a Incidência corresponde à razão entre o atributo "total_cases" e "population" multiplicado por 100.000 habitantes."*

Inicialmente, foi criado um novo vetor, vetor esse que irá conter os dados referentes ao novo atributo *ClassedeRisco*, que contém 3 valores possíveis: "Vermelho", "Amarelo"e "Verde". Estes valores são escolhidos a partir dos valores dos atributos *reproduction_rate* e *incidence*. As restrições para a atribuição de cada uma das cores é feita a partir da matriz de risco da figura 8, retirada do enunciado do trabalho.

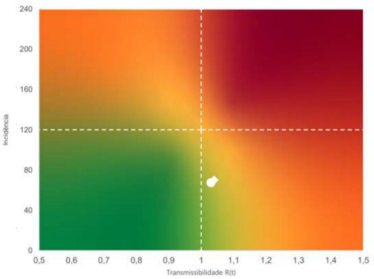

Figura 8. Matriz de risco [13].

De acordo com a matriz apresentada, para a *ClassedeRisco* ser "Verde", a *reproduction_rate* deve ser inferior a 1 e a *incidence* deve ser inferior a 120. Para ser "Vermelho", a "reproduction_rate"deve ser superior a 1 e a "incidence"deve ser superior a 120. Por último, caso os valores não respeitem as 2 condições anteriores, a cor será "Amarela".

As restrições supracitadas foram aplicadas sobre os dados, recorrendo à função *ifelse*.

Após o vetor ter sido criado, foi criada uma nova coluna na estrutura dos dados originais e na estrutura dos dados normalizados, denominada *ClassedeRisco*, coluna essa que contém os dados do vetor.

Recorrendo à função *table*, visualizou-se o nº de ocorrências dos valores "VerdeVermelho"e "Amarelo"da coluna *Classede-*

Risco, números esses que podem ser visualizados na tabela 7.

Tabela VII
TABELA COM O Nº DE OCORRÊNCIAS DOS VALORES "VERDE", "VERMELHO"E "AMARELO

Verde	Vermelho	Amarelo
34	120	55

Por último, procedeu-se á remoção das colunas *reproduction_rate* e *incidence* da estrutura de dados originais e da estrutura dos dados normalizados.

D. Exercício 8

O exercício pede o seguinte: *"Avalie a capacidade preditiva relativamente a este novo atributo ClassedeRisco usando os métodos: a) árvore de decisão b) rede neuronal c) K-vizinhos-mais-próximos Compare os resultados dos modelos anteriores. Discuta em detalhe qual o modelo que apresentou o melhor e o pior desempenho de acordo com os critérios: Accuracy; Sensitivity; Specificity e F1. "*

Num primeiro momento, converteu-se os dados da coluna *NiveldeRisco* nos dados normalizados, que possui valores não numéricos, em valores numéricos, por forma a que a mesma possa ser utilizada posteriormente nos algoritmos.

Assim como no exercício 6, foi utilizado o método *Holdout* para dividir os dados normalizados, sendo que, no caso, a coluna *ClassedeRisco* foi removida dos conjuntos gerados pela divisão, por não possuir valores numéricos. No entanto, os valores foram guardados em 2 outras estruturas.

O passo anterior foi realizado para que se pudesse, enfim, aplicar o algoritmo K-vizinhos-mais-próximos, por forma a prever o atributo *ClassedeRisco*, fazendo uso dos valores ímpares de K no intervalo (1..50), sendo que, para cada k, recolheu-se a taxa de acerto de previsão. O propósito é igual ao do exercício 6, no caso. O valor do k obtido foi 1.

Após a obtenção do k a ser utilizado para o K-vizinhos-mais-próximos, procedeu-se à aplicação do método *k-fold*. O valor de k escolhido para esse método foi 10, originando 10 folds. Para o algoritmo rede neuronal, foi escolhida a configuração de rede de 2 níveis internos, com 6 e 2 nós, respetivamente.

Aquando da execução do ciclo, verificou-se a existência de um erro originado pelo facto de, no caso da rede neuronal, as previsões do modelo não preveram uma das 3 classes do atributo *ClassedeRisco*, pelo que, na criação da matriz de confusão, obtivemos um erro e não conseguimos mitigá-lo.

Tendo em atenção esse detalhe, uma vez que o erro ocorria durante a 2º iteração, e a configuração da rede mencionada anteriormente foi escolhida por conveniência.

Para essa iteração, gerou-se os dados de treino e de teste com base nos folds criados anteriormente e nos dados normalizados. Para cada um dos modelos pedidos no enunciado, fez-se a criação do mesmo, seguida do cálculo dos valores previstos. Com os valores atuais e os previstos, recorreu-se à função *ml_test* por forma a que fosse efetuado o cálculos das métricas exigidas pelo enunciado.

A métrica *Accuracy* foi calculada automaticamente. No entanto, a fórmula consiste na razão entre a soma dos campos TP (*True Positive*) de todas as submatrizes (isto é, matrizes para cada uma das classes do atributo) e a soma de todos os valores da matriz. A primeira soma é referente a essas submatrizes, uma vez que o atributo em estudo possui 3 classes, pelo que a matriz não contém, de forma direta, os valores globais dos vários campos (TP, FN, FP e TN).

Já para o cálculo das restantes métricas, a função calcula as métricas para cada uma das submatrizes de forma automática. No entanto, para se obter medidas globais, optou-se pela média dos 3 valores para cada uma das métricas.

No final da iteração, guardou-se os valores das métricas de cada modelo em 4 matrizes, uma para armazenar cada métrica.

Os valores obtidos encontram-se presentes na tabela 8.

Tabela VIII
TABELA COM AS MÉTRICAS EXIGIDAS PARA CADA ALGORITMO

Métrica	Árvore de decisão	Rede neuronal	K-vizinhos-mais-próximos
Accuracy	0.7333333	0.6666667	0.4666667
Sensitivity	0.7388889	0.6416667	0.4444444
Specificity	0.8674242	0.8134921	0.6349206
F1	0.7387205	0.6158730	0.4437229

A partir dos resultados observados, pode-se concluir o seguinte:

Em relação a todas as métricas apresentadas, pode-se constatar que o modelo com pior desempenho é o K-vizinhos-mais-próximos, sendo que o que apresenta melhor desempenho é a árvore de decisão, uma vez que o primeiro modelo apresenta os valores mais baixos, enquanto que o segundo apresenta os mais elevados.

V. CONCLUSÕES

No âmbito deste trabalho, foi-nos pedido para realizarmos uma análise de vários indicadores sobre um conjunto de dados reais relativos à pandemia COVID-19 a nível mundial, através da resolução de um conjunto de exercícios propostos. Os dados encontravam-se descritos num ficheiro de dados fornecido, sendo que as resoluções foram realizadas com recurso à linguagem R.

Primeiramente, a análise desses indicadores foi feita através de modelos de regressão, utilizando-se diversos algoritmos de aprendizagem automática. Começou-se por importar o ficheiro com os dados a serem utilizados, seguido da criação de um diagrama de correlação, onde se pôde constatar as diversas correlações entre os vários atributos presentes nesse mesmo ficheiro. Em seguida, obtivemos um modelo de regressão linear, recorrendo ao método *Holdout*, para dividir o conjunto de dados, em que a variável dependente é o *total_deaths* e a independente o *new_cases*. A partir do modelo, obteve-se a função linear resultante, que nos deu a reta correspondente ao modelo, reta essa que foi representada num diagrama

de dispersão, que permitiu verificar a relação entre as 2 variáveis. Ainda foram pedidos os cálculos dos valores de MAE e RMSE. Por último, por forma a prever o atributo *life_expectancy*, foram utilizados os seguintes modelos: regressão linear múltipla, árvore de regressão e rede neuronal. Recorremos ao método k-fold para obtermos um conjunto de dados de MAE e RMSE para cada modelo, a cada iteração. A partir do conjunto obtido, concluimos que o modelo com pior desempenho foi o da regressão linear múltipla. Com os 2 restantes modelos, procedeu-se à realização de um teste de médias para averiguar se existiam ou não diferenças significativas entre o desempenho de ambos. Chegou-se, no entanto, à conclusão de que não era possível verificar tais hipóteses. Contudo, de acordo com os dados que tinham sido obtidos, o modelo com melhor desempenho foi a rede neuronal.

Para a análise dos indicadores através de modelos de classificação, numa primeira fase, derivámos um novo atributo, denominado por *NiveldeRisco*, discretizando o atributo *stringency_index* em 2 classes: *low* e *high*, usando a média desse atributo como valor de corte, sendo que, caso o valor do atributo fosse superior à média, seria atribuído, na nova coluna, o valor *high*, senão *low*. De seguida, foi nos pedido o estudo da capacidade preditiva relativamente ao atributo derivado anteriormente, utilizando os seguintes métodos: árvore de decisão, rede neuronal e k-vizinhos-mais-próximos. Recorrendo, assim como o enunciado pedia, ao método k-fold, obtivemos a média e o desvio padrão da taxa de acerto de previsão para os 3 modelos. Partindo desses resultados, concluiu-se que o que teve pior desempenho foi a árvore de decisão. Utilizando os restantes modelos, verificou-se se existiam diferenças significativas no desempenho desses modelos, através de um teste de médias. A partir dos resultados do teste, não foi possível concluir que existiam ou não diferenças significativas. No entanto, de acordo com os resultados da média e desvio padrão, o modelo que apresentou melhor desempenho foi a rede neuronal. No exercício seguinte, derivou-se um novo atributo *ClassedeRisco*, discretizando o atributo *reproduction_rate* e *incidence*, com base nos valores presentes numa matriz de risco disponibilizada, em 3 classes: *Vermelho*, *Amarelo* e *Verde*. Por último, foi nos pedido para avaliarmos a capacidade preditiva deste novo atributo, usando os métodos: árvore de decisão, rede neuronal e k-vizinhos-mais-próximos e, posteriormente, compararmos os resultados de acordo com os seguintes critérios: *Accuracy; Sensitivity; Specificity* e F1. Para atingir o objetivo pedido, recorreu-se ao método k-fold, por forma a obtermos um conjunto de cada métrica mencionada. No entanto, devido a um erro que obtivemos aquando da resolução do exercício, só conseguimos obter um resultado para cada modelo, para cada métrica. Partindo da análise desses resultados, concluimos que em relação a todas as métricas, o modelo que apresentou pior desempenho foi o k-vizinhos-mais-próximos, enquanto que o que apresentou melhor desempenho foi a árvore de decisão.

REFERÊNCIAS

[1] Ritchie, H. (n.d.). Coronavirus source data. Retrieved April 19, 2021, from https://ourworldindata.org/coronavirus-source-data

[2] Informática, E., Isep, D. E. I. (2021). 1 . 1 Instalar o R e o RStudio 1 . 2 Iniciar o R no RStudio. 1–29

[3] Cross-Validation. (n.d.).

[4] Rodrigo Santana. (2020, February 6). Validação Cruzada: Aprenda de forma simples como usar essa técnica. Minerando Dados. https://minerandodados.com.br/validacaocruzada-aprenda-de-forma-simples-como-usar-essa-tecnica/

[5] Souza, E. G. de. (2019, April 19). Entendendo o que é Matriz de Confusão com Python. Medium. https://medium.com/data-hackers/entendendo-o-que-é-matriz-de-confusãocom-python-114e683ec509.

[6] Inform, E. (2020). Regressão Linear Análise de Dados em Informática. 1–33.

[7] Madureira, A. (2018). McGraw-Hill, 1997. • Catarina Silva e Bernardete Ribeiro, Aprendizagem Computacional em Engenharia.

[8] Madureira, A. (1997). Decision Trees. McGraw-Hill.

[9] A. M. Madureira, "K-Nn Algorithm," Dei-Isep, no. 1, pp. 34–37, 2020.

[10] A. M. Madureira, "Neural Network - Historic Perspective," pp. 1–12, 2021.

[11] Madureira, A. (2020a). Aulas T - Testes de Correlação. 1–9.

[12] Singh, D. (2019, November 6). Normalizing Data with R — Pluralsight. Retrieved June 16, 2021, from https://www.pluralsight.com/guides/normalizing-data-r

[13] Trabalho Prático Iteração 2 Análise de Dados em Informática. (2021).