LETI-ANADI – Estimativa da obesidade em pessoas com idades entre os 14 e os 61 anos

David Rodrigues, Rui Costa
e-mail: 1211574@isep.ipp.pt , 1210971@isep.ipp.pt

***Abstract*— Neste artigo, irão ser aplicados alguns algoritmos de aprendizagem automática na exploração de dados, juntamente com a respetiva comparação através de testes estatísticos. O conjunto de dados utilizado consistia na estimativa dos níveis de obesidade em pessoas com idades entre os 14 e os 61 anos e diversos hábitos alimentares e condições físicas. Foram recolhidos dados de 2111 pessoas, nos quais foram obtidos 17 atributos relacionados com hábitos alimentares e condição física dos participantes.**

Palavras-chave — Dataset, Machine Learning, Python, Regressão, Classificação

I. INTRODUÇÃO

O *Machine Learning* (ML) é um ramo da inteligência artificial (AI) que se concentra no uso de dados e algoritmos para permitir com que a AI copie a forma de aprendizagem dos humanos, melhorando ao longo do tempo a sua precisão [1]. O ML surgiu para o desenvolvimento de algoritmos para que os computadores pudessem aprender automaticamente modelos a partir de dados [2]. Neste artigo, apresentamos uma revisão do estado da arte em aprendizagem automática e análise de desempenho, juntamente com uma análise e exploração de dados focada em técnicas de regressão e classificação.

II. ESTADO DA ARTE

A. Principais Algoritmos de Aprendizagem Automática

O ML pode ser dividido em 3 categorias de aprendizagem: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço. Cada uma destas categorias aborda diferentes tipos de problemas e utiliza diferentes abordagens para encontrar soluções.

O método mais simples de diferenciar estes três tipos de aprendizagens é a partir da maneira como os modelos são treinados e o tipo de dados utilizados. A maior diferença entre elas, é o tipo de dados utilizados. A aprendizagem supervisionada utiliza "*Label Data*", enquanto a aprendizagem não supervisionada não utiliza. "*Label Data*" possui um atributo especialmente designado, e o objetivo é usar os dados fornecidos para prever o valor desse atributo em instâncias ainda não vistas [2]. Já a aprendizagem por reforço é um método de AI onde os algoritmos aprendem com resultados e feedbacks para tomar decisões autónomas [3] [4].

Visto que grande parte dos resultados apresentados neste artigo foram obtidos através de modelos de aprendizagem supervisionada, iremos dar maior foco a esse tipo de modelos.

A aprendizagem supervisionada pode ser dividida em 2 classes de problemas de ML: Classificação (onde o objetivo é agrupar exemplos numa ou mais classes) e Regressão (onde o objetivo é definir um valor para uma determinada entrada) [2].

Alguns dos algoritmos de aprendizagem automática incluem:

- Regressão Linear: modelam a relação entre duas variáveis, ajustando uma linhas reta contínua aos dados. É um método conhecido de análise de regressão;
- *Support Vector Machines*: desenham um hiperplano entre os pontos de dados mais próximos para diferenciar classes de forma clara;
- Árvores de decisão: dividem os dados em conjuntos homogéneos através da utilização de regras "*if-then*" com base no diferenciador mais significativo. Como resultado, aprendem regressões lineares locais que aproximam a curva senoidal [5];
- K-vizinhos-mais-próximos: classificam novos pontos de dados com base nos pontos de dados mais próximos, medida por uma função de distância [3].
- Rede Neuronal: utiliza os neurónios interconectados em camadas para processar dados, permitindo com que computadores aprendam com erros e melhorem continuamente, sendo eficazes na resolução de problemas complexos [6].

B. Principais Algoritmos de Análise de Desempenho

Relativamente a algoritmos de análise de desempenho, que são utilizados para avaliar e medir a eficácia, eficiência e qualidade de algoritmos, podem ser utilizados os seguintes:

- Métricas de Avaliação de modelos de regressão: as principais métricas incluem o *Mean Squared Error* (*MSE*), o *Mean Absolute Error* (*MAE*), o *Root Mean Squared Error* (*RMSE*) e o R-Squared (R^2). O *MAE* representa a diferença média entre os valores originais e previstos, o *MSE* representa a diferença média ao quadrado, o *RMSE* é a raiz quadrada do *MSE*, e o R^2 representa o coeficiente de determinação que indica o ajuste do modelo aos valores originais, variando de 0 a 1, onde valores mais altos indicam um melhor ajuste.
- *Cross Validation*: envolve a reserva de uma pequena amostra do conjunto de dados, a construção do modelo com os restantes dados e, em seguida, a avaliação da eficácia do modelo na amostra reservada. Se o modelo funcionar bem nos dados de teste, é considerado eficaz.

- Matriz de Confusão: é um resumo dos dados dos resultados de previsão de um problema de classificação, que mostra o número de previsões corretas e incorretas para cada classe. Revela como o modelo de classificação está confuso ao fazer previsões e fornece *insights* sobre o tipo de erros cometidos.
- Métricas de Avaliação de Modelos de Classificação (*Accuracy*, Precisão, Sensibilidade, *Recall*, Especificidade e *F1-score*. A *accuracy* refere-se à capacidade de um classificador binário de conseguir identificar corretamente tanto os positivos quanto os negativos. A precisão é medida pela proporção de positivos previstos que são realmente positivos. A sensibilidade indica a habilidade do classificador em detetar verdadeiros positivos, sendo o *recall* a proporção de positivos reais classificados corretamente. Por outro lado, a especificidade mede a capacidade do classificador em detetar verdadeiros negativos. O *F1-score*, que varia entre 0 e 1, é a média harmónica da precisão e da sensibilidade, que oferece uma medida combinada do desempenho do classificador [2].

III. Análise e Exploração de Dados - Regressão

Para a análise e exploração dos dados, irão ser utilizados vários gráficos e tabelas, gerados em *Python*. Para cada um deles, iremos tirar uma breve conclusão.

Nesta análise, foi utilizado o conjunto de dados (*dataset*) "Dados_Trabalho_TP2.csv", cujos dados consistem na estimativa dos níveis de obesidade em pessoas com idades compreendidas entre os 14 e os 61 anos e diversos hábitos alimentares e condições físicas. Foram recolhidos dados de 2111 pessoas, nos quais foram obtidos 17 atributos relacionados com os hábitos alimentares e condição física dos participantes.

Os atributos relacionados com os hábitos alimentares são: Frequência de Consumo de Comida Altamente Calórica (FCCAC), Frequência de Consumo de Vegetais (FCV), Número de Refeições Principais (NRP), Consumo de Comida Entre Refeições (CCER), Consumo de água (CA), Consumo de Bebidas Alcoólicas (CBA), Monitorização do Consumo Calorias (MCC), Histórico de Obesidade Familiar, Género, Idade, Peso, Altura, Frequência de Atividade Física (FAF), Tempo de Utilização de Dispositivos Eletrónicos (TUDE), Fumador e Transporte utilizado (TRANS). O conjunto de dados ainda inclui um atributo chamado "*Label*" que corresponde à categoria de risco de obesidade de cada indivíduo.

A primeira análise feita foi a verificação da dimensão do *dataset*. Foi então concluído que o *dataset* tem 2111 linhas e 17 colunas (que correspondem aos atributos descritos anteriormente).

Foi ainda verificada a existência de valores nulos, *Null* e *NaN* no dataset, chegando à conclusão que o mesmo não apresenta nenhum valor nulo nem nenhum valor inválido.

Foram ainda verificados os tipos de dados que temos no *dataset*. O *dataset* apresenta colunas do tipo "*object*" e do tipo "*float64*", indicando que o *dataset* apresenta tanto colunas categóricas, como colunas numéricas.

De seguida, foi derivado um novo atributo, Índice de Massa Corporal (IMC), através da informação dos atributos de Peso e Altura. Esse mesmo atributo foi posteriormente adicionado ao *dataset*.

Para o cálculo do IMC, foi utilizada a seguinte fórmula:

$$\text{IMC} = \frac{\text{Peso (Kg)}}{\text{Altura (m)}^2}$$

Após a adição do novo atributo, partimos para a análise dos atributos do *dataset* mais significativos através de análises gráficas como distribuições numéricas, análise de outliers, entre outras.

Como podemos ver na Figura 1, as variáveis com mais *outliers* são a idade e o NRP. Relativamente à idade, esses *outliers* devem-se ao facto da idade do *dataset* ter um valor médio de 24.31 anos, logo, os *outliers* podem ser considerados como valores normais porque correspondem a possíveis adultos com idades muito superiores à idade média do *dataset*. Relativamente ao NRP, podemos considerar os valores também como normais visto que o número de refeições principais varia de pessoa para pessoa, e os valores obtidos no *dataset* não parecem ser valores 'anormais'.

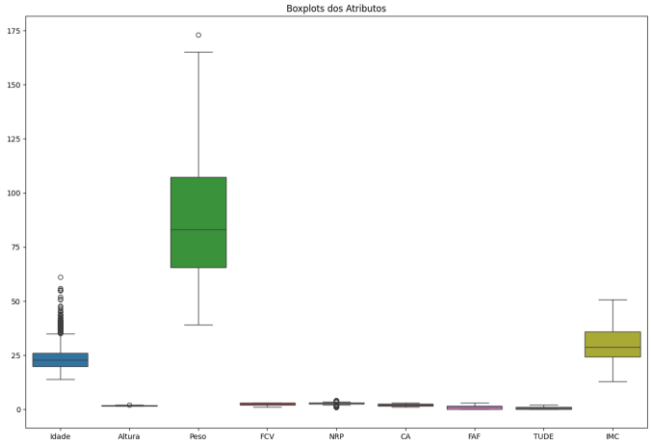

Figura 1 - Boxplots dos atributos

Após estas análises, foi realizado o pré-processamento de dados. Primeiro, através da análise da existência de valores nulos, *Null* e *NaN* no *dataset*, foi possível concluir que não existem valores nulos, *Null* e *NaN* no *dataset*, logo, o mesmo não precisava de ser filtrado para remover esses valores.

De seguida, foi feita a identificação de dados inconsistentes e *outliers*. Para isso, foram definidos alguns intervalos para as variáveis numéricas, tal como está representado na Tabela 1.

Tabela 1 – Intervalos de Valores

Variável	Intervalo Esperado
Idade	0 a 120 anos
Altura	0.5 a 2.5 metros
Peso	2 a 200 kilos
FCV	0 a 4
NRP	0 a 5
CA	0 a 4
FAF	-1 a 4
TUDE	-1 a 3
IMC	0 a 70

Os valores escolhidos foram baseados nos *boxplots* representados na Figura 1 e no sumário estatístico das variáveis numéricas, que nos permitiram identificar os valores mínimos e máximos de cada variável numérica. Foi chegada à conclusão que, após a utilização deste método, o *dataset* não apresenta praticamente *outliers* nenhuns nem valores inconsistentes. Apresenta *outliers* nas colunas da "Idade" e do "NRP", mas a explicação para a permanência desses dados no *dataset* já foi dada previamente no artigo.

Para finalizar o pré-processamento dos dados, foi feita a normalização dos mesmos através da seguinte fórmula:

$$y' = \frac{y - min_y}{max_y - min_y}$$

Esta função realiza a normalização min-max que mapeia os valores das variáveis no intervalo [0-1].

Para ser possível normalizar os dados, foi necessário converter os dados categóricos em dados numéricos, ou seja, cada valor único na coluna foi substituído por um número inteiro. Foi realizada a normalização dos dados porque quase todas as variáveis têm escalas diferentes e algumas delas têm uma disparidade significativa.

Depois do pré-processamento dos dados, foi criado um diagrama de correlação entre todos os atributos, para verificar a relação de cada atributo com os restantes. O diagrama de correlação apresenta-se na Figura 2.

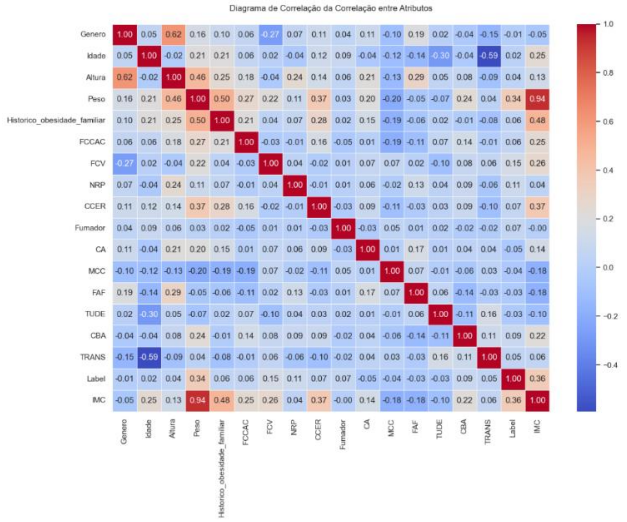

Figura 2 - Diagrama de correlação

Como podemos ver, a correlação entre os atributos é bastante baixa, o que indica que os atributos são independentes uns dos outros. Existem, no entanto, algumas relações fortes entre alguns atributos. Como por exemplo, o Peso e o IMC estão fortemente positivamente correlacionados, o que é esperado, visto que o IMC é calculado com base no peso. Relativamente à altura, a baixa correlação com o IMC pode ser explicada pelo facto que o IMC é mais influenciado pelo peso do que pela altura, devido à fórmula do IMC, ou seja, a baixa correlação entre a altura e o IMC não se deve apenas à pequena variação absoluta nas alturas entre pessoas, mas sim à maneira como o IMC é calculado e como o peso varia mais significativamente entre indivíduos. A forte correlação positiva entre a Altura e o Género também é esperada, visto que os homens tendem a ser mais altos que as mulheres.

A moderada correlação negativa entre os Transportes e a Idade pode ser explicada pelo facto de as pessoas mais jovens tenderem a usar mais os transportes públicos do que as pessoas mais velhas. Realçar que os transportes públicos é o tipo de transporte mais abundante na coluna "TRANS".

De seguida, foi desenvolvido um modelo de regressão linear simples para a variável "IMC", utilizando a idade como atributo. Dividimos os dados em conjuntos de treino e teste, com uma distribuição de 80% e 20%, respetivamente. Obtivemos a seguinte função linear e o respetivo diagrama de dispersão apresentado na Figura 3:

$$\text{IMC} = 0.28 \times Idade + 22.75$$

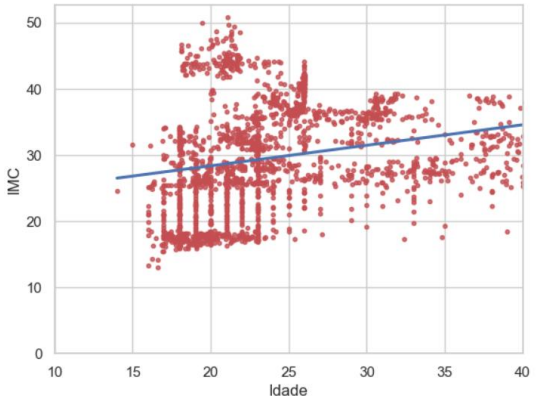

Figura 3 - Diagrama de dispersão do modelo de regressão linear entre o IMC e a Idade

Foram ainda calculados o *MAE* e a o *RMSE* do modelo sobre os 20% casos de teste, obtendo os seguintes valores:

- *MAE* = 6.519
- *RMSE* = 7.8541

Mediante os resultados obtidos, podemos concluir que os valores obtidos não parecem ser excessivamente elevados, embora isso indique que o modelo de regressão linear pode não ser totalmente preciso.

Após obtermos este modelo de regressão linear, foi feito outro, utilizando outra variável dos preditores disponíveis no *dataset*, para verificar se era possível obter um modelo de regressão linear simples com melhor resultado. A variável escolhida foi o "Peso", visto ser a variável com melhor correlação relativamente ao IMC. Foi aplicado exatamente o mesmo procedimento, obtendo o seguinte diagrama de dispersão (Figura 4) e os seguintes resultados:

$$\text{IMC} = 0.28 \times Peso + 5.02$$

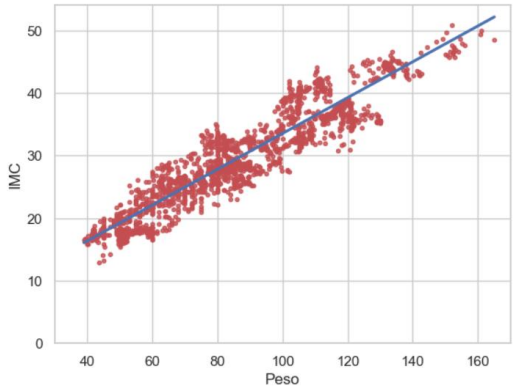

Figura 4 – Diagrama de dispersão do modelo de regressão linear simples entre o IMC e o Peso

- *MAE* = 2.3652
- *RMSE* = 2.9047

Como podemos ver, o *MAE* e o *RMSE* obtidos são bastante baixos, o que pode indicar que o modelo de regressão linear é preciso quando se utiliza o Peso como variável independente em relação ao IMC. Podemos então concluir que este modelo apresenta um melhor resultado relativamente ao anterior.

De seguida, e tendo em conta o conjunto de dados, pretendeu-se prever o atributo "IMC" através de 3 modelos: Regressão Linear Múltipla, Árvore de Regressão e Redes Neuronais.

Na Regressão Linear Múltipla, utilizamos as 4 variáveis com maior correlação relativamente ao IMC, sendo elas o Peso, Histórico de obesidade familiar, o CCER e a "*Label"*. Fizemos a divisão dos dados em dados de treino e de teste (desta vez, 30% dos dados para teste e 70% para treino), obtendo a seguinte equação de regressão:

$$\text{IMC} = 4.406 + (0.274 \times Peso) + (0.359 \times Histórico\ obesidade\ familiar) + (0.411 \times CCER) + (0.124 \times Label)$$

Utilizando esta equação de regressão, foram ainda previstos os valores do IMC para o conjunto de teste, obtendo os seguintes resultados (Tabela 2):

Tabela 2 – Valores atuais vs Valores previstos

	Actual value	***Predicted value***
1989	41.889	36.666
1700	36.035	35.586
1801	42.912	44.068
1482	32.648	28.106
1303	34.660	35.935

Como podemos ver, os valores estão próximos uns dos outros, tirando o primeiro que apresenta uma diferença considerável.

Após fazer a previsão, calculamos o R^2, o MAE, o *MSE* e o *RMSE* do modelo de regressão linear múltipla, obtendo os seguintes resultados:

- R^2 = 87.7441
- *MAE* = 2.2887
- *MSE* = 7.9187
- *RMSE* = 2.814

Como podemos ver, o modelo de regressão linear múltipla pode não ser muito preciso para fazer previsões, visto que tem um R^2 um bocado baixo e um *MSE* bastante elevado.

No modelo de árvore de regressão, foram escolhidos todos os atributos do *dataset* para a previsão do IMC. Isso deve-se ao facto de que, após alguns testes, foi possível verificar que a árvore de regressão tem melhor desempenho ao utilizar todos os atributos do *dataset* do que ao utilizar apenas alguns atributos (como foi feito na regressão linear múltipla). Separamos os dados em dados de treino e de teste (30% dos dados para teste e 70% para treino) e obtemos a árvore de regressão com os seguintes resultados:

- *MAE* nos dados de treino: 0.464
- *MAE* nos dados de teste: 0.575
- *RMSE*: 0.7656

Como podemos ver, o modelo de árvore de regressão tem um *MAE* (tanto no teste como no treino) e um *RMSE* bastante baixos, o que indica que o modelo é bastante preciso para fazer previsões.

Relativamente ao modelo de Redes Neuronais, utilizamos novamente todos os atributos do *dataset*, mas tomamos uma abordagem diferente. Foram criadas duas redes neuronais, uma onde era variado o número de neurónios de um *Hidden* Layer (de 4 a 30) e outra onde era variado o número de neurónios de 2 *Hidden Layers* (de 4 a 10 em ambos os *layers*). Em cada uma das redes, era também variada a função de ativação (*relu*, *tanh* e *identity*).
No final de cada teste, foram selecionadas as melhores redes neuronais para cada uma das funções de ativação.

Na rede 1 obtivemos os seguintes resultados (Tabela 3):

Tabela 3 - Melhores redes neuronais - Rede 1

Activation	***Hidden Layers***	R^2	*MAE*	*RMSE*
relu	22	0.971	1.065	1.388
tanh	24	0.989	0.621	0.839
identity	11	0.934	1.614	2.075

Como podemos ver, o melhor modelo é claramente o modelo com a função de ativação "*tanh*", visto que tem o maior R^2, o *MAE* mais baixo e o *RMSE* mais baixo.

Na rede 2, obtivemos os seguintes resultados: (Tabela 4):

Tabela 4 - Melhores redes neuronais - Rede 2

Activation	***Hidden Layers***	R^2	*MAE*	*RMSE*
relu	(6, 5)	0.9573	1.291	1.674
tanh	(8, 4)	0.993	0.408	0.683
identity	(5, 4)	0.937	1.603	2.041

Como podemos ver, o melhor modelo é claramente o modelo com a função de ativação "*tanh*", visto que tem o maior R^2, o *MAE* mais baixo e o *RMSE* mais baixo.

Para descobrir qual o melhor modelo para prever o "IMC", foram comparados os resultados obtidos pelos 3 modelos utilizados, através da comparação do *MAE* e da *RMSE*.

Através da análise do *MAE* e da *RMSE* dos modelos de regressão linear, árvore de regressão e redes neuronais, podemos concluir que o melhor modelo é o modelo com a função de ativação "*tanh*" com 8 e 4 neurónios em cada *Hidden Layer*, respetivamente, da rede 2 do modelo de redes neuronais, visto que tem o *MAE* e *RMSE* mais baixos. Foi ainda apresentado o R^2, para esse mesmo modelo, e podemos ver que é igual a 0.993, indicando que o modelo é bastante preciso.

Para finalizar a nossa análise relativamente à regressão, foi ainda verificado se os resultados obtidos para os dois melhores modelos são estatisticamente significativos.

Com base nos valores de *MAE* e *RMSE* fornecidos acima, podemos concluir que o modelo com a função de ativação *tanh* com 8 e 4 neurónios em cada *Hidden Layer* da rede 2 do modelo de redes neuronais e o modelo de Árvore de Regressão são os melhores modelos para prever o IMC de cada pessoa. Com isto, vamos agora verificar se os resultados obtidos entre ambos são estatisticamente significativos (para um nível de significância de 5%). Para isso, realizamos um teste t para duas amostras emparelhadas. Podemos assumir que são duas amostras emparelhadas porque estamos a utilizar o mesmo conjunto de dados para ambos os modelos.

Para realizar o teste, foram utilizados os resíduos de ambos os modelos. Ou seja, após as previsões de cada modelo, calculamos a diferença entre o valor real e o valor previsto. Foram então formuladas duas hipóteses:

- Hipótese Nula (H_0): Não há diferença significativa entre os desempenhos dos dois modelos.
- Hipótese Alternativa (H_1): Há uma diferença significativa entre os desempenhos dos dois modelos.

A conclusão do teste indica que o *p_value* é igual a 0.2852. Como o *p_value* é maior que o nível de significância (0.05), não rejeitamos H_0, logo, os resultados entre a rede 2 do modelo de rede neuronal e o modelo da árvore de regressão não são estatisticamente significativos.

IV. Análise e exploração de dados - Classificação

Nesta parte do artigo, utilizaremos diferentes modelos de classificação para prever o risco de obesidade de cada indivíduo (atributo "*Label*"). Os modelos utilizados incluem a Árvore de Decisão (DT), *Support Vector Machine* (SVM), Rede Neuronal (NN) e o K-vizinhos-mais-próximos (KNN), sendo que serão avaliados através da técnica *k-fold cross validation*. Mas antes da avaliação serão mencionados alguns pontos importantes efetuados em cada modelo para se obter os melhores resultados possíveis.

A. Classificação usando todos os atributos como variável X excepto pelo atributo "Label"

Para começarmos a usar cada um dos modelos mencionados, primeiro devemos definir a nossa variável de recurso (X) e a nossa variável de alvo (Y). Para este caso, a variável X vai possuir todos os atributos do *dataset* (incluindo IMC), exceto pelo "Label", que será a nossa variável Y.

a. Árvore de Decisão - *Overfitting*

Para evitar que ocorra *overfitting*, ou seja, para se evitar que o modelo treinado não se ajuste muito bem aos dados de treino ao invés de generalizar para novos dados, podíamos aplicar o método *GridSearchCV* da biblioteca *sklearn* para encontrarmos os melhores hiperparâmetros da árvore de decisão de forma a limitar o crescimento da mesma para evitar *overfitting*. No entanto, não foi necessário aplicar este método, porque nos resultados obtidos para a técnica *k-fold cross validation*, explicado mais á frente, percebemos claramente que o modelo treinado não apresenta quais queres sinais de *overfitting*.

b. SVM – Escolha do *kernel* e do parâmetro de ajuste

Para se encontrar o melhor *kernel* e o melhor parâmetro, foram realizados alguns testes através de uma busca em grade (*GridSearchCV*) usando a validação cruzada para encontrar os melhores parâmetros para o classificador SVM. Percebemos que o melhor *kernel* é o linear e que o melhor parâmetro, C, é 5.

c. Rede Neuronal – *Epochs* e Arquitetura da NN

Para o treino da NN, é necessário decidir quantos ciclos (*epochs*) são necessários para um bom treino do modelo. Para se determinar o número de *epochs*, foi realizado uma análise com a melhor arquitetura de NN encontrada de forma a tentar perceber qual é o melhor número de *epochs* necessários para treinar o modelo tendo em conta tempo e precisão.
Os resultados obtidos estão apresentados na tabela abaixo (Tabela 5)

Tabela 5 - Resultados Redes Neuronais

Epochs	Precisão	Perda	Val. precisão	Val. perda
600	0.95	0.11	0.90	0.29
700	0.96	0.08	0.89	0.45
800	0.95	0.11	0.88	0.31
900	0.96	0.10	0.90	0.40
1000	0.96	0.07	0.91	0.37
1500	0.97	0.06	0.92	0.42
2000	0.96	0.06	0.91	0.42

Concluímos que 1000 *epochs* tem um bom balanço tanto em termos de tempo como precisão e perda, em comparação aos outros.

Para a escolha da arquitetura do modelo NN foram propostas as seguintes três arquiteturas:

Arquitetura 1 (Figura 5):

```
model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=(17,)),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(9, activation='softmax')
])
```

Figura 5 - Arquitetura 1

Arquitetura 2 (Figura 6):

```
model2 = keras.models.Sequential([
    keras.layers.Flatten(input_shape=(17,)),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(9, activation='softmax')
])
```

Figura 6 - Arquitetura 2

Arquitetura 3 (Figura 7):

```
model3 = keras.models.Sequential([
    keras.layers.Flatten(input_shape=(17,)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(9, activation='softmax')
])
```

Figura 7 - Arquitetura 3

As três arquiteturas têm a mesma camada de entrada e saída, sendo que a primeira camada (*Flatten*) serve para converter a entrada num formato que pode ser usado pelas camadas subsequentes, no caso como temos 17 colunas/ atributos precisamos de colocar 17 no "*input_shape*". Na última camada temos 9 neurónios, sendo que a função *softmax* é usada para classificação multiclasse, uma vez que possuímos 9 classes diferentes para o atributo "*Label*". As restantes camadas possuem diferentes números de neurónios e possuem a função de ativação ReLu ("*relu*"), que é uma função que ajuda a introduzir não-linearidades no modelo.

Após treinarmos o modelo com 1000 *epochs* para cada uma destas arquiteturas, foram obtidos os resultados apresentados na Tabela 6.

Tabela 6 - Resultados de cada arquitetura

	Precisão	Perda
Arquitetura 1	0.983	0.039
Arquitetura 2	0.990	0.026
Arquitetura 3	0.994	0.017

O que podemos concluir, é que a arquitetura 3 é a melhor arquitetura entre as 3 arquiteturas, porque possui melhor precisão e menor perda.

d. K-vizinhos-mais-próximos

Para obtermos os melhores resultados deste modelo, foi testado diferentes valores de "k" para encontrar o menor *RMSE* e a maior precisão. No nosso caso foi k = 1, com valor de *RMSE* de 2.16 e de precisão de 0.78.

e. K-fold cross validation

Por fim, passamos para as avaliações, sendo primeiramente necessário entender melhor o que é *k-fold cross validation*. *K-fold cross-validation* é uma técnica para avaliar modelos preditivos em que o *dataset* é dividido em *k folds*. O modelo é treinado e avaliado k vezes, usando um *fold* diferente como conjunto de validação (teste) a cada vez. No final, é calculada a média das métricas de desempenho de cada *fold* para estimar o desempenho de generalização do modelo. E é exatamente isto que foi aplicado a todos os modelos de classificação, sendo que foram usados 10 *folds* para todas as avaliações [7].

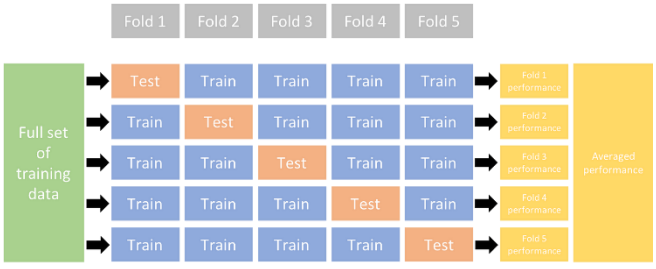

Figura 8 - K-fold cross-validation [7]

As avaliações obtidas para cada uns dos modelos efetuados estão apresentados na tabela abaixo (Tabela 7).

Tabela 7 – K-fold cross validation

Modelos	Performance média	Desvio padrão
DT	0.97	0.05
SVM	0.96	0.01
NN	0.86	0.00
KNN	0.86	0.08

O que podemos ver é que o melhor modelo é DT, visto que apresenta uma maior performance média. Mas para vermos se existe uma diferença significativa no desempenho dos dois melhores modelos (DT e SVM), foi realizado um *ttest* (função ttest_rel() da biblioteca *scipy*) que compara se duas amostras possuem valores médios idênticos. Se o nível de significância considerado ($\alpha = 0.05$) for maior do que *p-value* obtido no teste, então rejeitamos H0, que por sua vez significa que os modelos obtiveram um desempenho significativamente diferente [8].

Ao realizarmos o teste com os resíduos dos dois melhores modelos, obtivemos um *p-value* de 0.3308, o que significa que os desempenhos entre estes modelos não apresentam diferenças estatisticamente significativas.

f. Resultados dos modelos em certas métricas

Nos modelos mencionados anteriormente foram determinadas as métricas: Precisão, Sensibilidade, Especificidade e F1, tendo-se obtido os seguintes resultados (Tabela 8):

Tabela 8 - Resultados de cada modelo

Modelos	Precisão	Sens.	Espec.	F1
DT	0.965	0.965	0.994	0.965
SVM	0.944	0.944	0.991	0.944
NN	0.856	0.856	0.977	0.855
KNN	0.788	0.788	0.966	0.785

Podemos claramente ver que o modelo DT possui melhor performance que os restantes modelos.

B. Classificação usando apenas alguns atributos como variável X

Ao contrário do que foi feito anteriormente, vamos fazer a classificação apenas usando os atributos "IMC" e "Peso", como variável X. E claro, para a variável Y, manteve-se o atributo "Label". Os atributos de X foram selecionados com base no diagrama de correlação entre os atributos, pois estes dois atributos apresentaram a maior correlação com o atributo "Label", em comparação com os restantes atributos.

Aplicando exatamente as mesmas mudanças anteriormente mencionadas, foram obtidos os seguintes resultados para a técnica *k-fold cross validation.*

Tabela 9 - K-fold cross validation

Modelos	Performance média	Desvio padrão
DTs	0.96	0.04
SVM	0.97	0.01
NN	0.96	0.002
KNN	0.92	0.06

Como os resultados subiram positivamente da Tabela 7 para a Tabela 9, podemos concluir que uma seleção de atributos bem pensada não só reduz o tempo de duração do treino de modelos, como também possibilita a obtenção de melhores resultados.

C. Novos preditores

Nesta parte do artigo, foram adicionados ao ponto anterior mais três atributos preditores que foram:

- BMI: uma métrica simples da relação entre peso e altura; [9]
- BMR: quantidade mínima de energia que o corpo precisa para manter as suas funções vitais enquanto está em repouso; [10]
- Necessidades calóricas diárias; [10]

Sendo que foram apenas avaliados nos dois melhores modelos obtidos anteriormente. Os resultados para a técnica *k-fold cross validation* foram (Tabela 10):

Tabela 10 - K-fold cross validation SMV e NN

Modelos	Performance média	Desvio padrão
SVM	0.96	0.009
NN	0.96	0.001

O que podemos concluir é que a atribuição de novos preditores não ajudou a melhorar a performance destes modelos. Com o mesmo teste já mencionado neste artigo, concluímos que ambos os desempenhos destes modelos não apresentam diferenças estatisticamente significativas.

D. Prever Género

Para se prever o atributo "Género" foi utilizado apenas o atributo "Altura" como preditor, visto que este é o único atributo que apresenta uma boa correlação com o atributo "Género". Antes de avaliarmos com os dois melhores modelos obtidos anteriormente, foi alterado a arquitetura da NN que esta apresentada na Figura 9.

```
model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=(1,)),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(2, activation='sigmoid')
])
```

Figura 9 - Mudança da arquitetura da NN

Sendo que a única mudança importante foi na última camada, pois agora utiliza a função de ativação *sigmoid* que é mais apropriada para problemas de classificação binária ("Masculino" – 1 e "Feminino" – 0). Os resultados para a técnica *k-fold cross validation* foram (Tabela 11):

Tabela 11 - K-fold cross validation SVM e NN - Género

Modelos	Performance média	Desvio padrão
SVM	0.78	0.02
NN	0.77	0.005

Já os resultados para outras métricas foram (Tabela 12):

Tabela 12 - Outras métricas - Género

Modelos	Precisão	Sens.	Espec.	F1
SVM	0.783	0.783	0.783	0.782
NN	0.774	0.774	0.765	0.772

Novamente foi realizado o *ttest* para estes modelos, e percebeu-se que os desempenhos entre estes modelos não apresentam diferenças estatisticamente significativas.

V. Conclusões

Após a análise de dados e a aplicação de vários modelos de regressão e classificação, podemos chegar às seguintes conclusões:

- Entre os modelos de regressão testados para prever o IMC, a rede neuronal com a função de ativação "*tanh*" com 8 e 4 neurónios em cada *Hidden Layer* provou ser o melhor modelo, apresentando os menores valores de MAE e RMSE, além de um elevado valor de R^2, indicando uma alta precisão. O modelo da árvore de regressão também apresentou um bom desempenho. Com o teste t realizado, ficou provado que não existem diferenças significativas entre o desempenho de ambos os modelos.

- Para a classificação do risco de obesidade (*Label*), foram utilizados diversos modelos, incluindo Árvores de Decisão, SVM, Redes Neuronais e K-vizinhos mais próximos. Todos os modelos foram avaliados através da técnica de *K-fold cross validation*, sendo que quando aplicados quase todos os atributos do *dataset*, percebeu-se que não existem diferenças significativas entre o desempenho dos dois melhores modelos de classificação (Árvore de decisão e SVM).

- No caso da classificação, também foi possível concluir que uma escolha bem pensada de atributos preditores, pode levar a melhores resultados do que simplesmente utilizar todos os atributos do *dataset*.

Em resumo, a análise detalhada dos dados e a aplicação de técnicas de regressão e classificação permitiram indicar os melhores modelos para prever o IMC e classificar o risco de obesidade (*Label*). Com este artigo foi possível desenvolver o nosso conhecimento em técnicas avançadas de análise de dados, modelos preditivos e classificatórios.

VI. Referências

[1] IBM, "What is machine learning (ML)?," n.d. [Online]. Available: https://www.ibm.com/topics/machine-learning.

[2] ISEP, "DEI - Análise de Dados em Informática - 2º Semestre 2023/2024," 2024. [Online]. Available: https://moodle.isep.ipp.pt/course/view.php?id=4965.

[3] Microsoft, "Algoritmos de aprendizagem automática," n.d. [Online]. Available: https://azure.microsoft.com/pt-pt/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms.

[4] Google, "Supervised vs. unsupervised learning: What's the difference?," n.d. [Online]. Available: https://cloud.google.com/discover/supervised-vs-unsupervised-learning.

[5] scikit learn, "Decision Tree Regression," n.d. [Online]. Available: https://scikit-learn.org/dev/auto_examples/tree/plot_tree_regression.html.

[6] AWS Amazon, "O que é uma rede neural?," n.d. [Online]. Available: https://aws.amazon.com/pt/what-is/neural-network/.

[7] Ultralytics, "K-Fold Cross Validation with Ultralytics," 12 novembro 2023. [Online]. Available: https://docs.ultralytics.com/guides/kfold-cross-validation/.

[8] Scipy, "scipy.stats.ttest_rel," n.d. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html.

[9] CDC, "About Child & Teen BMI," n.d.. [Online]. Available: https://www.cdc.gov/healthyweight/assessing/bmi/childrens_bmi/about_childrens_bmi.html. [Acedido em 05 06 2024].

[10] P. Concall, "Basal Metabolic Rate (BMR)," n.d.. [Online]. Available: https://www.pediatriconcall.com/calculators/basel-metabolic-rate-bmr-calculator. [Acedido em 04 06 2024].

[11] OpenAI, "ChatGPT," n.d.. [Online]. Available: https://chat.openai.com/. [Acedido em 06 05 2024].

[12] pediatriconcall. [Online].