ANADI – Trabalho Prático 2

Gabriel Gonçalves
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto
Porto, Portugal
1191296@isep.ipp.pt

Tiago Leite
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto
Porto, Portugal
1191369@isep.ipp.pt

Francisco Bogalho
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto
Porto, Portugal
1211304@isep.ipp.pt

***Abstract* — A Unidade Curricular de Análise de Dados em Informática (*ANADI*) propôs a resolução de um conjunto de exercícios, com o objetivo de permitir aos alunos aplicar e aprofundar os seus conhecimentos. Este artigo apresenta uma solução possível para os referidos exercícios, feita com a linguagem de programação *Python* no ambiente interativo *Jupyter* N*otebook*, incluindo as explicações dos processos adotados e a fundamentação teórica que suporta as decisões tomadas.**

Keywords* — *Keras, tensorflow, árvores de decisão, SVM, redes neuronais, K-vizinhos-mais-próximos, machine-learning, modelos, analise de dados, tratamento de dados, camadas, K-fold, teste, treino, erro médio absoluto, erro quadrático médio, acurácia, sensibilidade, especificidade, F1-Score, MLPRegressor, parâmetros, prever, regressão, classificação.

I. INTRODUÇÃO

Com o grande aumento do volume de dados a serem processados, a área de Análise de Dados tem vindo a ganhar cada vez mais destaque. Neste contexto, os exercícios propostos para a Unidade Curricular de Análise de Dados em Informática (*ANADI*) surgem como uma oportunidade para explorar e aplicar conceitos fundamentais.

Este artigo começa com uma explicação teórica dos métodos utilizados nos exercícios, seguida pela descrição detalhada dos procedimentos adotados para resolver cada problema. Por fim, apresentaremos as conclusões gerais decorrentes deste trabalho analítico.

II. ESTADO DA ARTE

Nesta secção serão apresentadas informações teóricas sobre modelos usados, métricas utilizadas para avaliar os modelos e conceitos importantes para o enquadramento da situação.

A. Tipos de algoritmo

Os seguintes algoritmos foram usados no nosso trabalho e a seguir serão apresentados formalmente.

1) Regressão

Regressão é uma técnica de machine learning e estatística utilizada para modelar a relação entre uma variável dependente contínua e uma ou mais variáveis independentes.

a) Regressão linear simples

A regressão linear simples consiste num modelo estatístico que analisa a relação entre a variável dependente (Y) e a variável independente (X).

É aplicada quando se tem uma relação causal entre duas variáveis e é definida pela expressão: $Y = \beta_0 + (\beta_1 \times X) + \epsilon$ [1].

b) Regressão linear múltipla

A regressão linear múltipla consiste num modelo estatístico que analisa a relação entre a variável dependente (Y) e duas ou mais variáveis independentes (X).

Para se aplicar é necessário que: exista uma relação linear entre a variável alvo e as variáveis preditoras, os resíduos da regressão devem ser normalmente distribuídos e não deve existir multicolineariadade. O modelo é definido pela expressão: $Y = \beta_0 + (\beta_1 \times X_1) + \ldots + (\beta_n \times X_n) + \epsilon$ [1].

c) Árvore de regressão

Árvores de regressão utilizam uma estrutura de árvore de decisão para modelar a relação entre variáveis. Em cada nó da árvore, uma condição é aplicada para dividir os dados em subgrupos mais homogêneos, com base em uma variável explicativa. A previsão final é feita na folha da árvore, que contém um valor constante [1].

d) Rede neuronal

A rede neuronal de regressão, oferece uma grande flexibilidade e pode ser ajustado para atender a diferentes necessidades através da configuração de suas camadas, funções de ativação e métodos de otimização. O método usado na prática foi o *MLPRegressor* [2].

2) Classificação

Classificação é uma técnica de *Machine Learning* que categoriza dados em classes ou categorias discretas. O objetivo principal da classificação é prever a categoria ou rótulo de uma variável alvo com base em características de entrada, categorizando assim novos exemplos em uma ou mais classes pré-definidas.

a) Árvores de Decisão

As árvores de decisão são diagramas em forma de árvore que utilizam regras baseadas em características dos dados para dividir um conjunto de dados em subconjuntos homogeneizados, terminando em folhas que representam classes ou categorias. Cada nó interno representa uma característica ou atributo, cada ramo representa uma regra de decisão e cada folha representa uma saída. O método usado na prática foi o DecisionTreeClassifier [3].

b) SVM

O SVM é um método de aprendizagem supervisionado que procura um hiperplano ótimo para separar os dados em diferentes classes. Ele tenta maximizar a margem entre as classes, ou seja, a distância mínima entre o hiperplano e os pontos mais próximos de qualquer classe, chamados de vetores de suporte. O método usado na prática foi o SVC [4].

c) Redes neuronais

As redes neuronais de classificação consistem numa camada de entrada, com o número de nós correspondente aos valores de entrada, uma ou mais camadas escondidas com um número de nós e ativadores mais adequados ao problema e uma camada de output com o número de nós correspondentes aos

valores de saída e o ativador mais adequado. Na prática foi usada a biblioteca do keras [2].

d) K-vizinhos-mais-próximos

O K-vizinhos-mais-próximos é um algoritmo de apredizagem supervisionada que classifica novos casos com base em um conjunto de exemplos conhecidos. Ele calcula a distância entre os pontos e atribui a nova instância à classe maioritária entre os *K* vizinhos mais próximos. O método usado na prática foi o *KNeighborsClassifier* [5].

B. Métricas de avaliação de modelos

Depois de criados os modelos, precisam de ser avaliados para se poder comparar e saber quais podem estar a presentar problemas como overfitting ou underfitting.

1) Erro Médio Absoluto (MAE)

O Erro Médio Absoluto (MAE) é uma métrica de avaliação utilizada em modelos de regressão que calcula a média das diferenças absolutas entre as previsões feitas pelo modelo e os valores reais observados [6].

2) Erro Quadrático Médio (MSE)

O Erro Quadrático Médio (MSE) é uma métrica de avaliação de modelos de regressão que calcula a média dos quadrados das diferenças entre os valores preditos e os valores reais [6].

3) Raiz do Erro Quadrático Médio (RMSE)

A Raiz do Erro Quadrático Médio (RMSE) é a raiz quadrada do MSE, representando a magnitude média dos erros de predição em unidades comparáveis às dos dados originais [6].

4) Média da precisão

A média da precisão é a média das proporções de previsões corretas para cada classe, avaliando a capacidade de um modelo em identificar corretamente cada classe.

5) Variância da precisão

A variância da precisão mede a variabilidade da precisão de um modelo em relação a diferentes subconjuntos dos dados.

6) Precisão

A precisão é a proporção de verdadeiros positivos entre todas as instâncias que foram classificadas como positivas. Ela avalia a exatidão das previsões positivas, ou seja, quantas previsões que o modelo acerta [7].

7) Sensibilidade

A sensibilidade, também conhecida como recall ou taxa de verdadeiros positivos, é a proporção de verdadeiros positivos entre todas as instâncias que são realmente positivas [7].

8) Especificidade

A especificidade é a proporção de verdadeiros negativos entre todas as instâncias que são realmente negativas. Ela avalia a capacidade do modelo de identificar corretamente as instâncias negativas [7].

9) F1-Score

O F1-Score é a média harmônica da precisão e da sensibilidade. Ele fornece um equilíbrio entre essas duas métricas, especialmente quando há um desequilíbrio entre as classes [8].

C. Conceitos importantes

1) Overfitting

O *Overfitting* ocorre quando o algoritmo não consegue aprender um modelo que represente adequadamente os conceitos dos exemplos de treino, resultando assim numa incapacidade de generalizar para novos dados. Podemos então dizer que, o *overfitting* refere-se a um modelo que se ajusta muito bem aos dados de treino, e que não captura apenas os padrões relevantes, mas também os detalhes e ruídos específicos dos dados de treino [9].

Esta especialização excessiva compromete o desempenho do modelo com a utilização de novos dados, levando a um bom desempenho nos dados de treino conhecidos, mas uma má generalização para outros conjuntos de dados [9].

2) Underfitting

O *underfitting* refere-se a um modelo que não consegue representar corretamente os dados de treino e que não tem a capacidade de generalizar para novos dados [9].

Isso acontece quando um modelo não consegue aprender de forma adequada as relações presentes nos dados de treino, resultando em previsões problemáticas ou erróneas para novos dados ou dados para os quais o modelo não foi treinado. Frequentemente, apresenta um mau desempenho até mesmo nos dados de treino [9].

De uma forma resumida, podemos dizer que este modelo não consegue modelar os dados de treino nem generalizar para novos dados [9].

3) Generalization

O *generalization* refere-se à capacidade de um algoritmo ser eficaz numa variedade de entrada e aplicações de dados, pelo que, por outras palavras, se pode dizer que ele se refere à capacidade do modelo reagir a novos dados. Um algoritmo de *Machine Learning* deve generalizar a partir dos dados de treino para ajudar a realizar previsões precisas ao utilizar o modelo [9].

Com isto, devemos ter em mente que o *overfitting* e o *underfitting* causam uma má capacidade de generalização no conjunto de teste [9].

4) Inteligência artificial

A Inteligência Artificial (*IA*) é uma tecnologia que permite que os computadores e máquinas simulem a inteligência humana e as capacidades de resolução de problemas [10]. Alguns exemplos da utilização da *IA* são no processamento de linguagem natural e no reconhecimento de imagens [9].

A *IA* é uma área multidisciplinar onde existem várias subáreas e técnicas como *Machine Learning*, *Deep Learning*, Redes Neuronais, entre outros [9].

5) Machine learning

Machine Learning (ML) surgiu como um subcampo da IA, em que tem como preocupação o desenvolvimento de algoritmos de forma a permitir que os computadores consigam aprender de forma automática modelos (preditivos) a partir de dados [9].

Existem três tipos de problemas de aprendizagem [9]:

- **Aprendizagem Supervisionada –** Neste tipo de aprendizagem, a cada saída, ou seja, parâmetro a avaliar, é associado um rótulo (valor numérico ou categoria). O algoritmo é treinado com base em valores conhecidos de entradas e saídas, de modo que, posteriormente, consiga prever o rótulo de saída com base na entrada recebida.

  Quando o rótulo de saída é um valor real, estamos perante um algoritmo de regressão. Se a saída puder assumir apenas um conjunto de rótulos predefinidos, trata-se de um algoritmo de classificação;
- **Aprendizagem Não Supervisionada** – Na aprendizagem não supervisionada não é atribuído um rótulo aos dados de saída. O algoritmo identifica

padrões a partir de um grande conjunto de dados, com o objetivo de encontrar grupos de itens semelhantes;

- **Aprendizagem por Reforço** – Na aprendizagem por reforço existem dois componentes principais: o agente e o ambiente. O agente aprende por tentativa e erro, recebendo feedback sob a forma de recompensas ou penalizações após realizar determinadas ações, com vista a maximizar esse sinal de recompensa.

6) Train/test split

O *train/test split* serve para dividir o set de dados completo em um set de treino e um de teste, segundo uma percentagem que é passada em parâmetro, essa percentagem controla o tamanho dos sets de treino e teste, a divisão dos dados é aleatória.

7) K-fold cross validation

O *K-fold-cross-validation* avalia o desempenho do modelo em diferentes subconjuntos dos dados de treinamento e, em seguida, calcula a média da taxa de erro de predição. O algoritmo é o seguinte:

- Divida o conjunto de dados aleatoriamente em k subconjuntos (ou *k-fold*) (por exemplo, 5 subconjuntos).
- Reserve um subconjunto e treine o modelo em todos os outros subconjuntos.
- Teste o modelo no subconjunto reservado e registe o erro de predição.
- Repita este processo até que cada um dos k subconjuntos tenha servido como o conjunto de teste.
- Calcule a média dos k erros registados. Isso é chamado de erro de validação cruzada, servindo como a métrica de desempenho para o modelo.

A validação cruzada *k-fold* (CV) é um método robusto para estimar a precisão de um modelo [6].

8) One hot encoding

O one-hot encoding é uma técnica utilizada na preparação de dados para modelos de machine learning, especialmente para tratar variáveis categóricas. Essas são variáveis que representam categorias ou rótulos e não têm uma ordem intrínseca, ou seja, cria uma coluna para cada valor diferente que houver e para cada uma dessas colunas codifica-as binariamente para representar o valor inicial [11].

9) Label encoding

O label encoding é uma técnica de pré-processamento de dados utilizada em machine learning para converter variáveis categóricas em um formato numérico que pode ser facilmente interpretado por algoritmos de aprendizado. Essa técnica é essencial quando se trabalha com dados categóricos, pois muitos algoritmos de machine learning requerem entradas numéricas para processar os dados de maneira eficiente [12].

III. MÉTODOS E RESULTADOS OBTIDOS

Para a realização deste trabalho foi utilizado em todas as instâncias um nível de significância de 5% para todos os testes de hipótese efetuados. Além disso, é importante ressaltar que o parâmetro *random_state* foi definido como 42 para garantir a reprodutibilidade dos resultados.

A. Regressão

1) Exercício 1

Através da utilização da linguagem *Python*, importou-se um ficheiro para o projeto, possuindo informação relativa aos hábitos alimentares e condição física dos participantes do estudo realizado. Uma síntese destes dados revelou a avaliação de dois mil cento e onze indivíduos em dezassete categorias distintas.

Com base no resumo dos dados, verificou-se que a maioria das informações são de caráter categórico. Isto implica a necessidade de realizar uma preparação específica desses dados para possibilitar a aplicação de modelos de aprendizagem automática em estudos futuros.

2) Exercício 2

Após a importação dos dados, foi instruída a criação de um novo atributo chamado "IMC" usando a informação dos atributos do Peso e Altura. Com isto, calcularam-se os valores deste novo atributo através da fórmula:

$$IMC = \frac{Peso}{Altura^2}$$

3) Exercício 3

Para analisar os atributos mais significativos do conjunto de dados em relação ao IMC, foram utilizados gráficos de dispersão e um *boxplot.* A análise incluiu as seguintes variáveis: Idade, Altura, Peso, Frequência Cardíaca em Repouso, Número de Refeições por Dia, Consumo de Álcool, Frequência de Atividade Física e Tempo Usado em Dispositivos Eletrónicos. Foram criados gráficos de dispersão para cada par de atributos, permitindo visualizar possíveis correlações entre cada variável e o IMC.

Além disso, foi criado um *boxplot* do IMC, apresentado na Figura 1, que permite concluir que a mediana do IMC está em torno de 29 e a maioria dos valores estão entre 25 e 35. Os resultados indicam que, enquanto a mediana e a amplitude interquartil do IMC são relativamente estáveis, há variabilidade suficiente para justificar uma análise mais aprofundada sobre a influência de cada atributo no IMC.

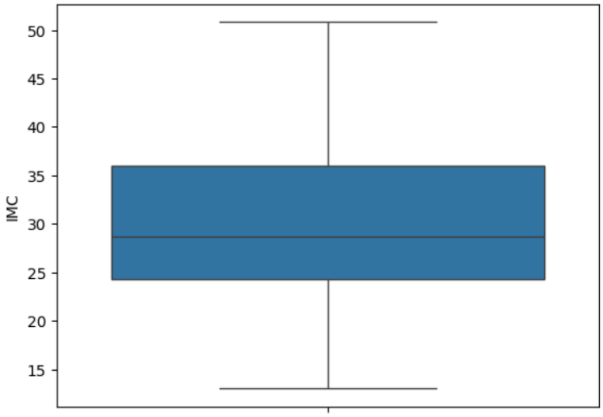

Figura 1 - Boxplot do IMC

4) Exercício 4

a) Alínea a)

Como a limpeza do *dataset* foi efetuada na importação dos dados ou na criação de novas variáveis de dados, este requisito não foi aplicado especificamente aqui, mas foi aplicado sempre que necessário.

b) Alínea b)

Para eliminar os *outliers* foi usado o método do intervalo interquartil (IQR), onde se calcula o primeiro (Q1) e terceiro (Q3) quartil, depois é calculado o intervalo interquartil. Para se

calcular o limite inferior e superior recorre-se a Q1 - 1.5 * IQR e Q3 + 1.5 * IQR, respetivamente. Os dados que são *outliers* são aqueles que se encontram fora dos limites, ou abaixo do inferior ou a acima do superior.

c) Alínea c)

A seleção dos atributos não foi implementada nesta fase, porque se decidiu implantá-la quando fosse necessária para os atributos necessários.

d) Alínea d)

Para normalizar os dados recorreu-se a definir uma função que calcule a normalização para uma coluna da seguinte forma, começa-se por calcular a diferença entre cada valor `y` e o valor mínimo de `y`, o que essencialmente leva todos os valores de `y` a começarem a partir de 0. Depois calcula-se a amplitude dos dados, ou seja, a diferença entre o valor máximo e o valor mínimo de `y`. Ao dividir o resultado do primeiro passo pelo segundo os valores estarão todos dentro de um intervalo de 0 a 1.

5) Exercício 5

Para este exercício criou-se um *dataset* com todos os valores (numéricos e categóricos), removeu-se os *outliers*, normalizou-se os numéricos e depois converteu-se as variáveis categóricas para variáveis *dummies*.

Só as correlações maiores que 0,30 e menores que -0,30 é que foram consideradas como relevantes.

A idade influência o peso, IMC, se o tipo de transporte predileto é automóvel ou público e o grau de obesidade. O peso, IMC e o grau de obesidade é porque com o envelhecimento, os indivíduos tendem a ganhar mais gordura por causa do abrandamento do metabolismo e do sedentarismo, o meio de transporte está também associa á idade por que com o aumento da mesma, os transportes pessoais são mais práticos e a capacidade de adquirir os mesmos é maior.

A altura influência o peso, porque quanto maior a altura maior o volume e consequentemente, maior o peso. Já o género é que influência a altura, sendo que os homens tendem a ser mais altos que as mulheres.

O peso influência o IMC e os graus de magreza e obesidade, porque ambos são calculados com base no peso. Já o histórico de obesidade familiar, se consome frequentemente comidas altamente calóricas, se consome comida entre refeições e se consome bebidas alcoólicas frequentemente ou não influenciam o peso, porque todos estes preditores descrevem comportamentos que envolvem consumir bebidas ou comidas altamente calóricas, ou consumir comidas ou bebidas em excesso.

O FCV é encontrado a influenciar a presença de obesidade mórbida e o IMC, possivelmente por causa de uma população cujos hábitos e dietas saem fora do que o IMC está habilitado para analisar fidedignamente. Já o género influência as mulheres a fazerem escolhas mais saudáveis que os homens, no que a consumo de vegetais.

O IMC é influenciado pelo histórico familiar, o consumo de comida entre as refeições e o consumo de bebidas alcoólicas. Ele influência os graus de obesidade, sendo os graus classificações para diferentes valores de IMC.

O género feminino tem alguma propensão a ser obesamente mórbido. Já o masculino tem o mesmo nível de propensão a não o ser.

O histórico de obesidade familiar inexistente indica uma chance razoável de se manifestar num individuo na forma de peso normal. Já um histórico de obesidade existente indica que há uma boa chance do individuo com esse histórico não ter um peso normal.

O consumo de comida entre as refeições pode ser ocasional, frequente ou sempre, quem faz um tem uma chance baixíssima de trocar, quem faz esta prática frequentemente tem uma chance razoável de ser magro, mas que o faz ocasionalmente tem uma chance alta de não ser magro ou ter um peso normal.

Quem opta por um dos tipos de transporte: caminhada, automóvel ou transporte público tem alta chance de não trocar de modo predileto.

6) Exercício 6

a) Alínea a)

Começo por criar um objeto do tipo *LinearRegression*, da biblioteca do *Scikit-Learn*, e treino esse objeto com o *dataset* de treino, depois calculo o score do modelo com o *dataset* de teste. Resultando em 0.1247 no *model score* e com a seguinte equação da reta: y= 0.3329 + 0.3320x.

b) Alínea b)

Na Figura 2, é possível visualizar o diagrama de dispersão que representa a relação entre o Índice de Massa Corporal (IMC) e a Idade. A linha vermelha representa a reta de regressão linear simples ajustada aos dados.

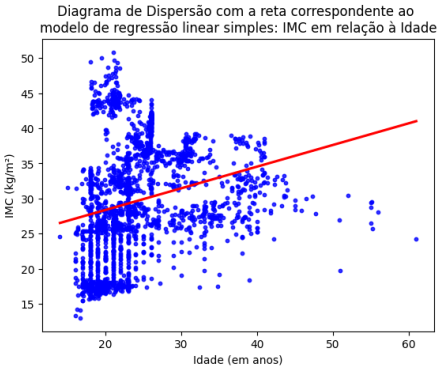

Figura 2 - Diagrama de dispersão com a reta correspondente ao modelo de regressão linear simples: IMC em relação à Idade

No eixo do *X* temos a Idade, em que esta varia aproximadamente entre os 18 e os 60 anos. Já no eixo do *Y* temos o IMC, em que este varia aproximadamente dos 15 kg/m² até aos 50 kg/m².

A linha de regressão vermelha mostra uma tendência de aumento do IMC com o aumento da idade. Isso indica que, em média, à medida que as pessoas envelhecem, o seu IMC tende a aumentar.

Os pontos (em azul) que representam os dados estão bastante dispersos ao longo da faixa etária, indicando uma variabilidade significativa no IMC entre indivíduos da mesma idade. Esta dispersão sugere que a idade sozinha não é um forte preditor do IMC, embora exista uma tendência geral de aumento.

c) Alínea c)

Para calcular o que foi pedido, começo por re-treinar o modelo com o *dataset* de treino e depois uso o modelo treinado para fazer previsões do y do *set* de treino e teste. Com estas previsões posso calcular o MAE para ambos os *sets* e o RMSE, cálculos estes que resultaram no seguinte:

- MAE do set de treino: 0.1785
- MAE do set de teste: 0.1670
- RMSE: 0.2006

d) Alínea d)

Para verificar se é possível obter um modelo de regressão linear simples com melhor desempenho do que aquele utilizando a idade como preditor, testou-se várias outras variáveis do conjunto de dados: Altura, Peso, FCV, NRP, CA, FAF e TUDE.

Para cada variável, foi treinado um modelo de regressão linear simples e avaliadas as métricas de Score (R^2), MAE e RMSE. Os resultados foram comparados para identificar a variável com melhor desempenho.

A variável Peso apresentou o melhor desempenho, com um Score de 86,96%, MAE de 0.0638 e RMSE de 0.0774, isto deve-se ao facto desta variável estar diretamente relacionado ao cálculo do IMC. A segunda melhor variável foi a Frequência Cardíaca de Repouso (FCV), que apresentou um desempenho próximo ao da Idade. As restantes variáveis apresentaram desempenhos inferiores à Idade.

7) Exercício 7

Através do conjunto de dados apresentado, previu-se o atributo IMC aplicando os seguintes modelos: Regressão Linear Múltipla, Árvore de Regressão e Rede Neuronal. Para realizar este exercício, definiu-se como variáveis independentes todos os atributos do *dataset* à exceção do Peso, da Altura e do campo IMC anteriormente calculado. Excluiu-se estes campos, devido ao IMC já ser o próprio IMC, pelo que não faz sentido prever algo que já se encontra definido. Também se removeram os campos Altura e Peso, pois o IMC é calculado utilizando estes dois atributos, pelo que não faz sentido eles estarem presentes pois determinam logo o IMC. Já como variável dependente, utilizou-se o campo IMC. Posteriormente, utilizou-se a técnica *holdout* para separar os dados, onde 80% deles foram utilizados para treino, e os restantes 20% para teste. Também se utilizou o parâmetro "*random_state = 42*" para garantir reprodutibilidade.

a) Alínea a)

Nesta alínea, utilizou-se a regressão linear múltipla para realizar a previsão do atributo "IMC". Primeiro, começou-se por criar uma instância do modelo de regressão linear e treinou-se esse mesmo modelo com os dados de treino. De seguida, fez-se a previsões sobre os dados de teste e calculou-se o MAE e o RMSE para avaliar a performance do modelo, obtendo os valores 0.0280 e 0.0377, respetivamente. Também se mediu a capacidade de generalização do modelo através do cálculo do R^2 no conjunto de teste, obtendo-se um valor de 96.90%, o que indica uma boa generalização do modelo para novos dados. De seguida, criou-se um gráfico de dispersão para comparar os valores reais dos previstos, que é possível ser visualizado na Figura 3.

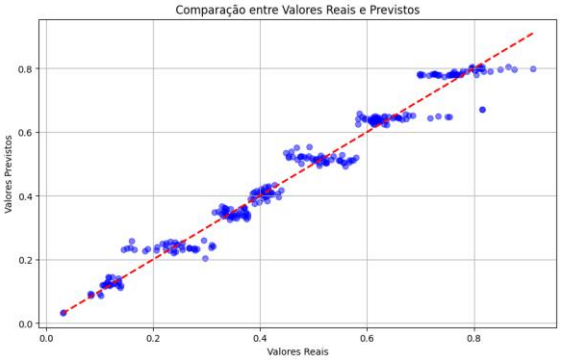

Figura 3 - Regressão linear múltipla, comparação entre valores reais e preditos

A linha vermelha tracejada serve como referência ideal onde os valores previstos seriam iguais aos valores reais. A distância dos pontos a esta linha reflete a precisão das previsões.

Os pontos azuis representam os pares de valores reais e preditos. A proximidade desses pontos à linha vermelha indica a precisão do modelo. No gráfico, a maioria dos pontos está bastante próxima da linha, o que sugere que o modelo tem um bom desempenho.

A distribuição dos pontos ao longo da linha vermelha mostra uma boa correlação linear, indicando que o modelo é capaz de capturar bem a relação entre as variáveis preditoras e o IMC.

b) Alínea b)

Nesta alínea, criou-se um modelo de Árvore de Regressão para realizar a previsão proposta, obtendo-se o modelo ilustrado na Figura 4.

Os parâmetros do modelo foram ajustados para uma profundidade máxima de 6 e um mínimo de 3 amostras para dividir um nó interno.

O modelo foi treinado usando os dados de treino, e foram feitas previsões para os conjuntos de treino e teste. Por fim, calculou-se e obteve-se para o *MAE* o valor de 0.02291 e, para o *RMSE*, o valor de 0.0351.

Além disso, o modelo apresentou um *score* de 0.9731, indicando um excelente ajuste aos dados e uma elevada precisão nas previsões. Assim, o modelo de Árvore de Regressão é altamente eficaz para a tarefa proposta.

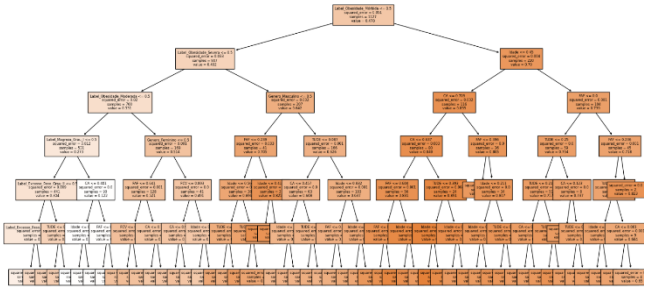

Figura 4 - Modelo da Árvore de Regressão obtido

c) Alínea c)

Para prever o atributo IMC com um *MLPRegressor*, foi definida uma lista de arquiteturas plausíveis para serem testadas no treino do modelo. Depois percorreu-se toda a lista de arquiteturas e treinou-se o modelo para cada arquitetura, de seguida foi calculada a performance do modelo, guardando esse valor numa variável, se é que foi maior do que o valor anterior, e este processo repete-se por todos os valores da lista. Desta forma descobre-se a melhor arquitetura.

De seguida treina-se o modelo com todas as combinações de ativadores, optimizadores e a melhor arquitetura descoberta anteriormente, e avalia-se a performance para cada combinação, descobrindo assim o melhor modelo para efetuar a previsão pedida. As métricas deste modelo são as seguintes:

- Coeficiente de Determinação (R^2): 0.9605
- Erro Médio Quadrático (MSE): 0.0018
- Raiz quadrada Do Erro Médio Quadrático (RMSE): 0.0426
- Erro Médio Absoluto (MAE): 0.0296

Pode-se ver pelos valores obtidos que este modelo exibe uma performance excecional.

8) Exercício 8

Para avaliar a performance de um modelo, é essencial calcular o *MAE* e o *RMSE*. Portanto, foram realizados esses cálculos para os modelos desenvolvidos na alínea anterior, obtendo-se os resultados apresentado na Tabela 1.

Tabela 1 - Resultados do MAE e RMSE para cada modelo criado

Modelo	MAE	RMSE
Regressão Linear Múltipla	0.028045	0.037744
Árvore de Regressão	0.022917	0.035140
Rede Neuronal	0.029643	0.042616

Entre os três modelos testados, o modelo de Árvore de Regressão apresentou a maior precisão e o menor erro na previsão do IMC, indicando uma forte relação não-linear com as variáveis independentes. O modelo de Regressão Linear Múltipla e a Rede Neuronal apresentaram desempenhos similares, mas inferiores ao da Árvore de Regressão, possivelmente devido à natureza dos dados ou à necessidade de uma melhor sintonização dos parâmetros.

9) Exercício 9

Para avaliar se os resultados dos dois melhores modelos, a Árvore de Regressão e a Regressão Linear Múltipla, são estatisticamente significativos, realizamos um teste *t* associado com um nível de significância de 5%.

Com base nos resultados obtidos, conclui-se que o valor *p* (0.3245) é superior ao nível de significância. Assim, não existem evidências estatísticas suficientes para rejeitar a hipótese nula, isto significa que a diferença entre os erros dos dois modelos não é estatisticamente significativa.

Esta conclusão sugere que, embora a Árvore de Regressão tenha apresentado um melhor desempenho, ambos os modelos podem ser considerados igualmente eficazes na previsão do IMC neste conjunto de dados.

B. Classificação

Para a resolução desta secção, foi proposto o uso de Modelos de Aprendizagem Automática de Classificação, utilizando o conjunto de dados importados na secção anterior.

Na abordagem à Árvore de Decisão, ajustou-se os parâmetros do modelo usando a técnica de *GridSearchCV*, que testou várias combinações de parâmetros, especificamente o limite máximo da profundidade da árvore, o número mínimo de amostras necessárias para dividir um nó interno e o número mínimo de amostras necessárias para ser um nó folha.

Os melhores parâmetros encontrados foram: profundidade máxima da árvore de 9, com um número mínimo de amostras para dividir um nó interno de 8 e um número mínimo de amostras para ser um nó folha de 1. Isto resultou em uma precisão média de cerca de 60.46%, com um desvio padrão de 7.71%.

Na abordagem à *SVM*, utilizou-se novamente o *GridSearchCV* para ajustar os parâmetros do modelo. Foram testadas diversas combinações de parâmetros, incluindo diferentes valores de *C*, *gamma* e *kernels* (*linear, rbf e sigmoid*).

Os melhores parâmetros encontrados foram: *C* igual a 10, *gamma* igual a 1 e o *kernel rbf*. Isto resultou numa precisão média de aproximadamente 69.05%, com um desvio padrão de 13.18%.

Na abordagem á rede neuronal optou-se pelo método de otimização aleatório do *keras tuner*, ou seja, foi realizada uma busca de hiperparâmetros aleatória. Para tal definiu-se uma classe *HyperModel* onde estará presente o código que define todas as possibilidades que o modelo pode assumir, ou seja, ter de 1 a 3 *hiddenlayers*, cada uma com, desde 16 nós a 512 e cada camada, incluindo a final, pode também ter um dos seguintes ativadores: *relu*, *tanh*, *sigmoid*, *softmax*. Este *HyperModel* foi usado pelo *RandomSearch* para explorar o espaço de hiperparâmetros de forma aleatória, guardando tais parâmetros no *tunner*, para depois os melhores serem procurados com *tuner.search*. Para o *RandomSearch* foi definido um máximo de 5 tentativas e 2 execuções por tentativa. O *tuner.search* está a ser executado dentro de um for de *K-fold cross validation*, por isso o *search* está a treinar com os sets de treino do *fold* em que estiver.

Os melhores parâmetros foram uma camada com 192 nós e um ativador *relu* e um ativador *softmax* para a última camada, note-se que por causa da natureza do *RandomSearch*, os melhores parâmetros podem ser indicados como outros, mas a performance destes parâmetros é fidedigna.

Na abordagem ao algoritmo dos K-vizinhos-mais-próximos (*KNN*), realizou-se também uma pesquisa em *GridSearchCV* para encontrar os melhores parâmetros do modelo. Para isso ajustou-se parâmetros como o número de vizinhos considerados, o método de atribuição de pesos e a métrica de distância usada para medir a proximidade entre os pontos.

Os melhores parâmetros encontrados foram: número de vizinhos igual a 7, método de pesos *'distance'* e métrica de distância *'manhattan'*. Estes parâmetros resultaram numa precisão média de aproximadamente 67.13%, com um desvio padrão de 10.46%.

1) Exercício 1

a) Alínea a)

Neste exercício, avaliou-se a performance dos modelos de classificação criados, utilizando a validação cruzada (*k-fold cross validation*). A média e o desvio padrão da métrica de precisão (*Accuracy*) de cada modelo encontram-se apresentados na Tabela 2.

Tabela 2 - Resultados obtidos no estudo da capacidade para prever o risco de obesidade

Modelo	Média	Desvio Padrão
Árvores de Decisão	~0.604606	~0.077144
SVM	~0.690460	~0.131798
Rede Neuronal	~0.666429	~0.019429
K-vizinhos-mais-próximos	~0.671306	~0.104556

Após a análise dos resultados obtidos, identificaram-se os dois melhores modelos: *SVM* e K-vizinhos-mais-próximos. De seguida, realizou-se um teste *t* de *Student* para comparar as pontuações destes modelos, obtendo um valor *p* de 0.3670, que indica que não existe uma diferença significativa no desempenho dos dois modelos, dado que o valor *p* é superior ao nível de significância de 5%.

Apesar de não haver diferença significativa entre os dois melhores modelos, o modelo *SVM* apresentou a maior média de precisão (0.6905) e, portanto, foi identificado como o modelo com melhor desempenho.

b) Alínea b)

As árvores de decisão demonstram uma precisão e sensibilidade equilibradas, ambas em torno de ~60%, indicando sua capacidade de classificar corretamente instâncias positivas. A alta especificidade de ~95% sugere que o modelo é eficaz na identificação de instâncias negativas. No entanto, o F1-Score de ~65% indica que o modelo pode não ser tão robusto na presença de desequilíbrios nas classes ou em cenários de *overfitting*.

O SVM exibe uma precisão e sensibilidade consideráveis, ambos em torno de ~70%. Isso indica uma capacidade melhorada de classificar corretamente instâncias positivas. Além

disso, a especificidade de ~85% sugere uma boa capacidade de identificar instâncias negativas. O F1-Score de ~70% indica um equilíbrio entre precisão e sensibilidade, sugerindo um desempenho geral sólido do modelo.

A rede neuronal exibe uma precisão de ~60%, indicando que cerca de ~60% das instâncias classificadas como positivas estão corretas. No entanto, a sensibilidade de moderada sugere que o modelo é exímio a identificar corretamente todas as instâncias positivas. A alta especificidade sugere que o modelo é eficaz na identificação de instâncias negativas. O F1-Score relativamente inferior indica que o modelo tem um equilíbrio entre a precisão e sensibilidade.

O método dos K-vizinhos-mais-próximos demonstra uma precisão e sensibilidade consistentes, ambas em torno de ~70%, indicando sua capacidade de classificar corretamente instâncias positivas. No entanto, a especificidade de ~80% sugere que o modelo pode ter uma tendência ligeiramente maior de classificar instâncias negativas como positivas. O F1-Score de ~70% indica um equilíbrio sólido entre precisão e sensibilidade, sugerindo um desempenho geral confiável do modelo.

A precisão e a sensibilidade iguais em alguns casos indicam que a proporção de instâncias classificadas corretamente como positivas é igual à proporção de instâncias positivas corretamente identificadas.

Em suma, a escolha do melhor modelo depende do objetivo e do *dataset*, neste caso o modelo que se melhor adapta às nossas necessidades é o SVM, porque apresenta os melhores resultados.

c) Alínea c)

Inicialmente, para se resolver este exercício, a seleção de atributos baseou-se muito na análise do diagrama de correlação anteriormente criado, mais especificamente no Exercício 5 do tópico Regressão. Essa seleção, baseou-se na seleção de atributos que tivessem uma correlação forte com o IMC. Numa primeira fase, definiu-se que uma forte correlação teria de ser acima do valor 0.30 ou abaixo dos -0.30. Como os resultados obtidos não foram o esperado, definiu-se que os atributos a selecionar, teriam de ter valores de correlação acima dos 0.20 e valores abaixo de -0.20. Contudo, continuou-se a obter valores de precisão bastantes inferiores ao esperado. Dessa forma, testaram-se diferentes combinações de atributos.

Para todos os modelos, os atributos que os faziam aumentar a sua média de precisão, eram os seguintes: idade, frequência de consumo de vegetais, número de refeições principais, consumo de água, frequência de atividade física, tempo de utilização de dispositivos eletrónicos, género, histórico de obesidade familiar, frequência de consumo de comida altamente calórica, consumo de comida entre refeições, fumador, monitorização do consumo de calorias, consumo de bebidas alcoólicas.

Para além disso, os modelos árvore de decisão, SVM e rede neuronal, também utilizam o atributo tipo de transporte utilizado.

Isto demonstra que cada atributo diferente, acrescenta uma informação diferente e útil na previsão do IMC de um dado individuo.

2) Exercício 2

Para realizar este exercício, utilizaram-se os dois melhores modelos obtidos anteriormente, sendo estes aqueles que possuem uma melhor precisão nos resultados, mais concretamente o SVM e o K-vizinhos-mais-próximos.

A partir da análise dos preditores já existentes, pensou-se em criar 5 preditores de forma a verificar se estes conseguiam melhorar o desempenho dos dois melhores modelos. Os preditores criados foram os seguintes:

- **Densidade Calórica (DC)** – Procura entender a qualidade da dieta que um dado individuo possa estar a fazer;
- **Tempo Ajustado de Atividade Física (TAAF)** – Permite avaliar o equilíbrio entre atividade física e o sedentarismo;
- **Hábito Alimentar Saudável (HAS)** – Verifica se o hábito alimentar do indivíduo é saudável, o que permite também verificar a sua preocupação com a saúde;
- **Índice do Hábito de Consumo Calórico (IHCC)** - Permite verificar o comportamento alimentar de uma pessoa, indicando a probabilidade de ela ter uma alimentação que pode contribuir para o aumento de peso ou obesidade;
- **Hábitos Diários *Fit* (HDF)** - Permite verificar a relação que existe entre a frequência de consumo de vegetais e a frequência de realização de atividade física, mostrando assim se o individuo tem hábitos diários saudáveis.

Após a criação destes novos atributos, avaliou-se para um nível de significância de 5%, se eles contribuíam para a existência de uma diferença significativa no desempenho dos dois melhores modelos já referidos.

Após a introdução e avaliação dos novos atributos em ambos os modelos, concluiu-se que com um nível de 5% de significância, os novos preditores não melhoraram significativamente a precisão de nenhum dos modelos. Na realidade, os novos preditores até reduziram um pouco a precisão (~1%, em cada).

Isto sugere que os novos preditores criados (DC, TAAF, HAS, IHCC e HDF) não estão a fornecer informações adicionais que melhorem a capacidade dos modelos de distinguir entre as classes.

3) Exercício 3

Neste exercício, realizou-se o estudo da capacidade preditiva relativamente ao atributo "Género" utilizando os métodos Rede Neuronal e *SVM*.

a) Alínea a)

Para a criação do modelo *SVM*, procedeu-se com a pesquisa em grelha para ajustar os parâmetros do modelo, considerando diferentes valores de regularização *C*, valores de *gamma* e diferentes *kernels*. Cada combinação de parâmetros foi avaliada utilizando a validação cruzada *k-fold* e foram calculadas métricas de desempenho, incluindo precisão, sensibilidade, especificidade e F1-Score. Estes valores fornecem uma visão abrangente do desempenho dos modelos *SVM* em prever o atributo "Género" com base nos dados fornecidos.

Após avaliação de todas as combinações, os cinco melhores modelos *SVM* foram identificados com base na média da precisão. Estes modelos foram selecionados para posterior análise e comparação.

Como no caso anterior, foi usado o *RandomSearch* para se descobrir a melhor combinação de parâmetros, mas desta vez não foi usado o *K-fold cross validation*, desta forma, duas camadas, uma com 208 nós e o *tanh* como ativador, 16 nós na segunda e o *relu* como ativador e *softmax*, apresenta-se como uma ótima opção de modelo.

Os resultados obtidos foram:

Tabela 3 - Resultados obtidos no estudo da capacidade preditiva relativamente ao atributo "Género"

Modelo	Média	Desvio Padrão
Rede Neuronal	~0.921	~0.020

SVM (C=1000.0, Gamma=0.01, Kernel=rbf	~0.860	~0.115
SVM (C=100.0, Gamma=0.1, Kernel=rbf)	~0.854	~0.123
SVM (C=10.0, Gamma=0.1, Kernel=rbf)	~0.848	~0.126
SVM (C=100.0, Gamma=0.01, Kernel=rbf)	~0.847	~0.107
SVM (C=1000.0, Gamma=0.1, Kernel=linear)	~0.846	~0.091

Através da análise dos resultados obtidos que se encontram na Tabela 3, podemos verificar que a Rede Neuronal obteve a melhor performance com uma média de taxa de acerto de ~92% e um desvio padrão de ~2%, indicando alta consistência nas previsões.

O modelo SVM com Kernel RBF (C=1000.0, Gamma=0.01), apresentou a segunda melhor média de taxa de acerto com ~86%, mas com um desvio padrão de ~11.5%, sugerindo maior variabilidade nos resultados.

Embora os modelos SVM apresentem médias de acerto relativamente próximas, o SVM com os hiperparâmetros C=1000.0 e Gamma=0.01 teve um desempenho ligeiramente superior. Os modelos com menores valores de C e diferentes valores de Gamma mostraram uma pequena redução na média de acerto e maior variação no desvio padrão. Notavelmente, o SVM com kernel linear (C=1000.0, Gamma=0.1) apresentou um desvio padrão menor, indicando maior consistência, apesar da média de acerto inferior.

A Rede Neuronal demonstrou ser superior aos modelos SVM testados, tanto em termos de média de acerto quanto de consistência. Embora os modelos SVM ofereçam taxas de acerto competitivas, apresentam maior variabilidade nos resultados. Assim, para a previsão do atributo "Género", a Rede Neuronal é o modelo mais robusto e confiável.

b) Alínea b)

Nesta alínea, realizou-se o teste *t* de *Student* para comparar o desempenho dos dois melhores modelos obtidos na alínea anterior, que foram uma Rede Neuronal e um SVM com *kernel rbf* (*C=1000.0, Gamma=0.01*). Depois de calcular a estatística *t* e o valor *p*, conclui-se que não há uma diferença significativa no desempenho destes modelos, uma vez que o valor *p* (0.2905) é superior ao nível de significância de 5%.

Estes resultados sugerem que tanto a Rede Neuronal como o modelo *SVM* indicado são igualmente competentes na previsão do atributo "Género" com base nos dados fornecidos.

c) Alínea c)

Para as redes neuronais usou-se o *tuner* para descobrir os melhores parâmetros, logo não foi preciso procurar manualmente, como foi o caso das SVM, o que resultou em haver só um modelo com os melhores parâmetros possíveis para o caso em estudo.

A rede neuronal construída é capaz de prever mais de 90% dos casos corretamente, apresenta uma sensibilidade igual á precisão o que significa que tem um desempenho consistente na deteção de casos positivos e não á falsos negativos. A especificidade é a menor métrica, ou seja, o modelo é melhor a identificar casos positivos do que casos negativos. Já o F1-score, sendo alto, significa que o modelo é capaz de minimizar tanto os falsos positivos quanto os falsos negativos.

O modelo que usa o *kernel* linear é o pior modelo de todos, porque o mesmo não é próprio para *sets* de dados que não são linearmente separáveis. Já os modelos que usam o *kernel rbf*, adaptam-se muito melhor ao caso em estudo. Para o *kernel rbf* os melhores parâmetros foram *C* = 100 e *Gamma* = 0.1, que tiveram uma precisão maior que 90% e uma sensibilidade igual á precisão. A especificidade é a menor métrica, ou seja, o modelo é melhor a identificar casos positivos do que casos negativos. Já o F1-score, sendo alto, significa que o modelo é capaz de minimizar tanto os falsos positivos quanto os falsos negativos.

Em suma, ambos o SVM e a rede neuronal estão muitos aptos para o trabalho de previsão, mas o SVM é o melhor modelo, pois consegue melhores resultados.

IV. Conclusão

Durante a resolução dos exercícios, foi possível aprimorar os conhecimentos sobre Análise de Dados, com especial ênfase em temas como a Regressão e Classificação

Além disso, adquiriu-se conhecimentos sobre a importância de um bom tratamento inicial dos dados e a identificação das suas características, permitindo uma análise mais precisa do conjunto de dados. Notou-se também um avanço no uso de gráficos para a análise de dados que permitem extrair conclusões sobre a dispersão dos dados e a identificação de possíveis *outliers*.

Em suma, a experiência adquirida através da resolução dos exercícios foi extremamente valiosa. Não só permitiu a aplicação prática de conceitos teóricos, como também destacou a importância da análise de dados no mundo atual. A capacidade de interpretar e analisar dados é uma habilidade essencial em muitos campos e indústrias, portanto, aprofundar estes conhecimentos e habilidades é de grande relevância para qualquer profissional.

Referências

[1] A. M. Madureira, "McGraw-Hill, 1997. • Catarina Silva e Bernardete Ribeiro, Aprendizagem Computacional em Engenharia," 2018.

[2] "Neural Networks".

[3] A. M. Madureira, "Decision Trees," 2022.

[4] A. M. Madureira, "Support Vector Machine," 2023, Accessed: Jun. 09, 2024. [Online]. Available: https://www.ibm.com/topics/support-vector-machine

[5] "kNN Algorithm." Accessed: Jun. 09, 2024. [Online]. Available: https://moodle.isep.ipp.pt/pluginfile.php/358848/mod_resource/content/9/kNN%20algorithm.pdf

[6] "Tree Regression Regression model evaluation metrics".

[7] Christian Thieme, "Understanding Precision, Sensitivity, and Specificity In Classification Modeling and How To Calculate Them With A Confusion Matrix." Accessed: Jun. 09, 2024. [Online]. Available: https://towardsdatascience.com/understanding-common-classification-metrics-titanic-style-8b8a562d3e32

[8] "F1 Score in Machine Learning." Accessed: Jun. 09, 2024. [Online]. Available: https://www.geeksforgeeks.org/f1-score-in-machine-learning/

[9] A. M. Madureira, "Análise de Dados em Informática Licenciatura em Engenharia Informática ISEP/IPP," 2023.

[10] "What is Artificial Intelligence (AI)? | IBM." Accessed: Jun. 06, 2024. [Online]. Available: https://www.ibm.com/topics/artificial-intelligence

[11] "One Hot Encoding in Machine Learning." Accessed: Jun. 09, 2024. [Online]. Available: https://www.geeksforgeeks.org/ml-one-hot-encoding/

[12] "Label Encoding in Python." Accessed: Jun. 09, 2024. [Online]. Available: https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/