Análise de Dados - Técnicas de Aprendizagem Automática

Daniel Braga
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto
Porto, Portugal
1200801@isep.ipp.pt

Gonçalo Nogueira
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto
Porto, Portugal
1201525@isep.ipp.pt

***Abstract* - Este artigo tem como objetivo a descrição e análise de dados de técnicas de aprendizagem automática para aplicar no enunciado do segundo Trabalho Prático da cadeira de Análise de Dados em Informática. De seguida, como Referências teóricas foram descritas as seguintes técnicas: regressão linear, árvores de decisão, k-vizinhos-mais-próximos e redes neuronais e para cada exercício foram explicados os métodos usados para o resolver e uma breve conclusão. Para este estudo foram utilizadas as capacidades de análise de dados da linguagem python, usando o Spyder.**

Palavras-Chave – técnicas de aprendizegem automática; regressão linear; árvores de decisão; k-vizinhos-mais-próximos; redes neuronais.

I. Introdução

O presente artigo científico foi desenvolvido no âmbito da unidade curricular Análise de Dados em Informática (ANADI), do 2º semestre do 3ºano da Licenciatura em Engenharia Informática (LEI) do Instituto Superior de Engenharia do Porto (ISEP). Neste âmbito foi proposta a realização de uma análise de dados de técnicas de regressão linear, árvores de decisão, k-vizinhos-mais-próximos e redes neuronais. Com este artigo deverá ser possível fazer uma análise sobre os dados utilizando técnicas de aprendizagem automática, através dos conhecimentos adquiridos nesta unidade curricular, usando a linguagem python e o Spyder. O projeto é constituído por 3 partes: revisão da literatura, regressão e classificação.

II. Revisão da Literatura

A. *Regressão Linear*

1) *Regressão Linear Simples*

2) *Regressão Linear Múltipla*

B. *Machine Learning*

A Aprendizagem Automática (AA) ou Machine Learning (ML), uma área e subgrupo da Inteligência Artificial (IA), é o processo de utilizar modelos matemáticos de dados para ajudar um computador a aprender sem instruções diretas.

A Aprendizagem Automática explora o estudo e construção de algoritmos que podem aprender com os seus erros e fazer previsões sobre dados, através de amostras (ou inputs). Apesar de a Inteligência Artificial recorrer a dois tipos de raciocínios, indutivo e dedutivo, este subgrupo apenas foca o indutivo (raciocínio que extrai regras e padrões de grandes conjuntos de dados).

As três principais técnicas de AA são:

- Aprendizagem supervisionada - aplicar etiquetas ou estruturas aos conjuntos de dados, aumentando a capacidade do computador fazer predições ou de tomar decisões;
- Aprendizagem não supervisionada - ao contrário da técnica descrita, a aprendizagem não supervisionada não aplica etiquetas ou estruturas, mas procura encontrar padrões e relações através do agrupamento de dados (clusters);
- Aprendizagem por reforço - nesta técnica, um programa de computador ajuda a determinar o resultando, baseando-se num ciclo de feedbacks.

Para além destas três técnicas, também são usados, normalmente, alguns passos para resolver problemas com o Machine Learning. Em primeiro lugar, recolher e formar dados é essencial neste processo, ou seja, após identificar a origem dos dados, estes são compilados. Através dos diferentes tipos de dados, diferentes algoritmos de aprendizagem automática vão ser usados. Neste processo, a estrutura é desenvolvida e são identificadas anomalias que posteriormente estes problemas serão resolvidos. Estes dados vão ser divididos pelos conjuntos de preparação (usado para melhorar os modelos do ML) e de teste (usado para avaliar o desempenho e precisão do modelo de dados final selecionado). Por último, com o resultado final, é possível obter informações, conclusões e resultados.

C. *Árvores de Decisão*

Uma árvore de decisão, de acordo com Lorraine Li, é um modelo de machine learning supervisionado que aprende com as regras de decisão provenientes de funcionalidades e utiliza-as para prever um objectivo. Como resultado, esta abordagem analisa os dados utilizando um processo de tomada de decisão baseado em várias perguntas.

Há dois tipos diferentes de árvores: árvore de decisão de variável categórica (tem como alvo uma variável categórica) e árvore de decisão de variável contínua (tem como alvo uma variável contínua). Estas árvores são muito usadas na área de machine learning e utilizam algoritmos que permitem prever com elevada precisão, estabilidade e fácil entendimento: Árvores de Regressão e Árvores de Classificação.

Por um lado, pode ser estabelecida uma ligação entre as árvores de decisão de varável categórica e o algoritmo das Árvores de Classificação, uma vez que este tipo de árvores permite que o algoritmo identifique a classe a que, possivelmente, a variável pertence. Após o algoritmo ser aplicado, o elemento que possuir uma maior pureza irá estar no topo da árvore.

Por outro lado, as árvores de decisão de variável contínua podem estar relacionadas ao algoritmo das Árvores de Regressão, visto que este algoritmo usa este tipo de árvores para antecipar o valor da variável contínua.

Esta abordagem emprega então cada uma das variáveis independentes num modelo de regressão que é atribuído à variável alvo. Depois disso, os dados são separados em muitos pontos, para cada uma das variáveis independentes, e o quadrado da diferença entre os valores previstos e obtidos é calculado para cada um desses pontos, obtendo-se uma "Soma de Erros Quadrados" ("Sum of Squared Errors" - SSE). Para cada variável, o SSE é comparado, e aquele com o valor SSE mais baixo é utilizado para determinar o ponto de divisão, que é depois repetido recursivamente.

*D. **K-Vizinhos-Mais-Próximos***

K-Vizinhos-Mais-Próximos é uma abordagem de reconhecimento de padrões muito utilizada. A base do seu funcionamento é encontrar o vizinho mais próximo de uma determinada instância e, consequentemente, a noção básica do K-nearest-neighbours (KNN) é classificar uma instância de dados com base nos seus vizinhos e pode ser usado para resolver problemas de classificação e regressão. O KNN funciona calculando as distâncias entre uma consulta e todos os exemplos disponíveis, escolhendo o número especificado de exemplos que estão mais próximos da consulta, e votando depois pelo valor mais frequente (no caso de classificação) ou calculando a média dos valores (no caso de regressão). Ao classificar em casos de teste, esta técnica envolve o treino completo, o que aumenta o tempo de reação. Isto é uma desvantagem das abordagens de aprendizagem preguiçosas, que carecem de uma fase de construção de modelos e adiam todo o trabalho de cálculo para depois de as amostras de teste terem sido classificadas.

*E. **Redes Neuronais***

Redes neuronais são modelos computacionais que se baseiam no funcionamento dos neurónios biológicos do corpo humano. Estes sistemas são capazes de aprendar a realizar tarefas através de dados, sem, na maioria dos casos, serem programados com regras específicas da tarefa. Uma rede neural é construida a partir de ligações de nodes ("artificial neurons"), obtidos a partir de dados, que irão permitir a passagem de informação. No entanto, para uma rede neural poder ser usada é necessário "treiná-la", ou seja, através de um processo de feedback (chamado backpropagation) é comparado o resultado obtido com o resultado esperado. Após a rede estar devidamente "treinada", uma rede neural pode ser não supervisionada (em casos de cluster) ou supervisionada (com finalidade de classificação).

*F. **Análise de Desempenho***

Para a análise e teste dos modelos anteriormente apresentados considerou-se o uso do algoritmo crossvalidation. Neste algoritmo, uma pequena parte dos dados é usada para validação da eficácia do modelo, enquanto que os restantes dados são utilizador para o treino do algoritmo.

Para o desenvolvimento deste trabalho recorremos ao uso de vários algoritmos de cross-validation: Hold Out (divide aleatoriamente os dados disponíveis em duas partes, uma será a amostra para treino do algoritmo e a outra para a fase de testes que irá efetuar as previsões) e K-fold cross-validation (divide a base de dados de forma aleatória em K subconjuntos com aproximadamente a mesma quantidade de amostras, em que, a cada treino e teste, um conjunto formado por K-1 subconjuntos são utilizados para treinamento e o subconjunto restante será utilizado para teste gerando os erros de predição, que irão ser usados para o cálculo da média dos erros registados).

Para as situações de regressão recorremos às métricas de avaliação do erro médio absoluto (somas das diferenças absolutas entre as previsões e os valores reais), da raiz quadrada do erro médio (calculo da raiz quadrada da média do quadrado das diferenças entre a previsão e os valores reais) e do coeficiente de determinação (critério mais usado na regressão linear para testar o ajuste do modelo).

Para os casos de modelos de classificação binária recorremos à matriz de confusão, que permite obter o número de positivos verdadeiros, positivos falsos, negativos verdadeiros e negativos falsos. A tabela em baixo representa uma matriz de confusão:

	Negativo (Previsto)	Positivo (Previsto)
Negativo (Atual)	Negativo verdadeiro	Positivo falso
Positivo (Atual)	Negativo falso	Positivo verdadeiro

Através desta matriz conseguimos calcular algumas métricas para a avaliação do modelo de classificação:

• **Accuracy**: É a razão entre as previsões corretas e o total número de previsões feitas.

$$Accuracy = \frac{Positivos\ verdadeiros + Negativos\ verdadeiros}{Total\ de\ exemplos}$$

• **Precision**: É a razão entre as previsões verdadeiras corretas e o total de previsões verdadeiras.

$$Precision = \frac{Positivos\ verdadeiros}{Positivos\ verdadeiros + Positivos\ falsos}$$

• **Recall**: É a razão entre as previsões verdadeiras corretas e todas as observações positivas.

$$Recall = \frac{Positivos\ verdadeiros}{Positivos\ verdadeiros + Negativos\ falsos}$$

• **F1**: É a média pesada da precision e recall.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

III. REALIZAÇÃO DO TRABALHO

Regressão

1) Para iniciar o exercício, o dataset foi carregado usando a função *read_csv* da biblioteca *pandas*. Uma verificação preliminar foi feita para remover a coluna "Unnamed: 0", se presente.

Em seguida, foram obtidas as dimensões do dataset e um sumário estatístico dos dados, incluindo medidas de tendência central e dispersão. Além disso, as primeiras linhas do dataset foram exibidas para entender melhor sua estrutura.

2) O IMC é uma medida comum para avaliar se uma pessoa está acima do peso, abaixo do peso ou dentro do peso considerado saudável.

Este é calculado dividindo o peso pela altura ao quadrado. Após calcular o IMC, as primeiras linhas do conjunto de dados, agora com o IMC incluído, são exibidas para verificação.

3) Aqui são realizadas análises visuais e estatísticas dos atributos do conjunto de dados.

Inicialmente, as colunas categóricas relevantes são convertidas em numéricas, se necessário.

Em seguida, são criados histogramas para as variáveis numéricas, gráficos de dispersão entre pares de variáveis e boxplots para visualizar a distribuição do IMC em relação a diferentes variáveis categóricas, como género, histórico de obesidade familiar e hábitos de fumar.

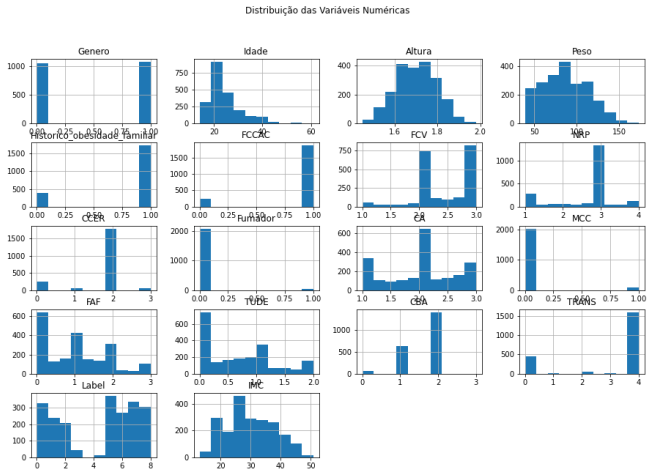

Figura 1 – Distribuição das variáveis numéricas

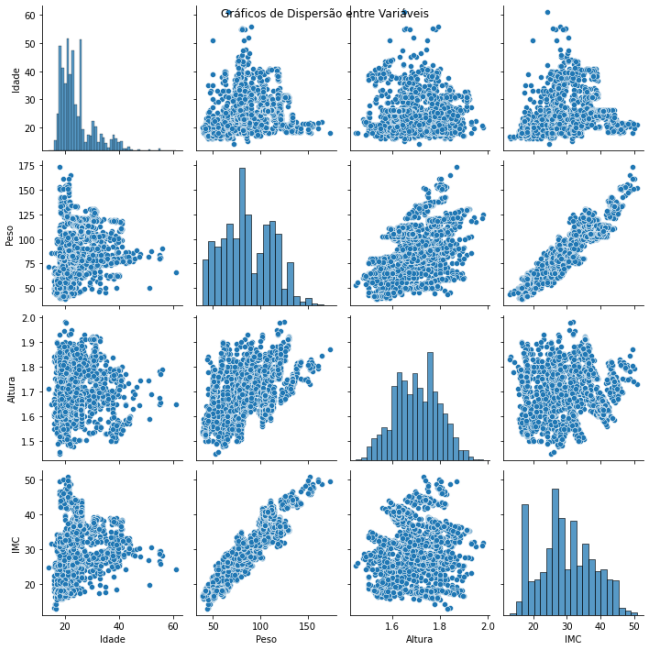

Figura 2 – Dispersão entre variáveis

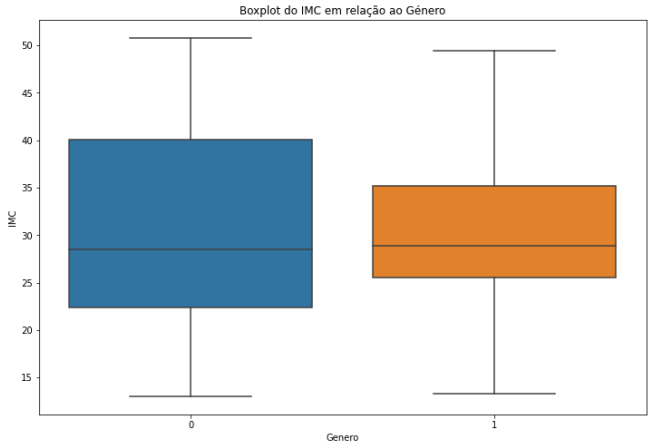

Figura 3 – Boxplot to IMC em relação ao género

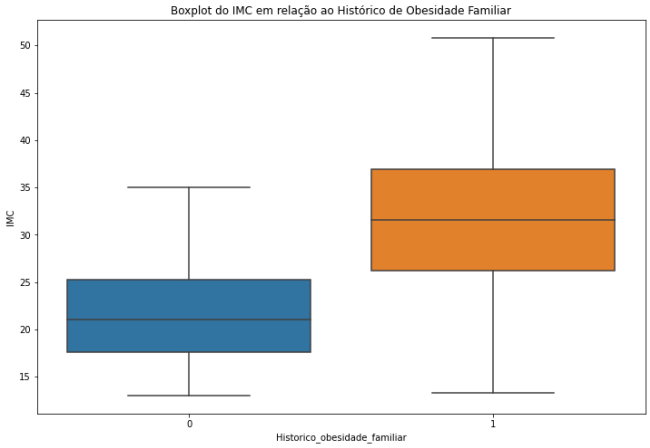

Figura 4 – Boxplot do IMC em relação ao histórico de obesidade familiar

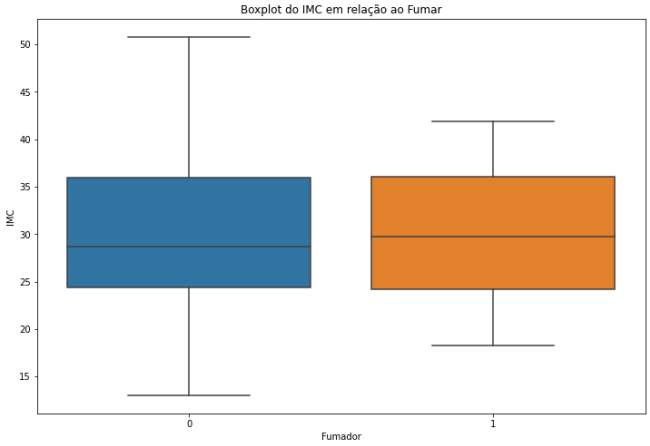

Figura 5 – Boxplot do IMC em relação ao fumar

4) Neste exercício são realizadas operações de limpeza e pré-processamento nos dados. Primeiramente, são removidas as entradas com valores ausentes para garantir a qualidade dos dados.

Em seguida, os atributos numéricos são normalizados usando a técnica de padronização para garantir que todas as variáveis estejam na mesma escala, o que é importante para muitos algoritmos de machine learning.

5) Foi criada uma matriz de correlação para analisar as relações entre todos os atributos do conjunto de dados. A matriz de correlação é visualizada usando um mapa de calor (heatmap), onde os valores de correlação entre os pares de variáveis são codificados por cores.

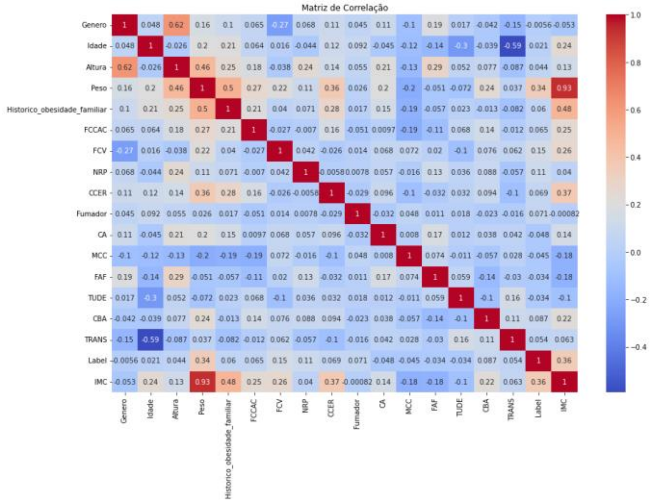

Figura 6 – Matriz de correlação

A matriz de correlação revela várias relações entre os atributos do conjunto de dados relacionados à estimativa de níveis de obesidade. Observa-se uma forte correlação positiva entre Peso e IMC (0.83), o que é esperado, dado que o IMC é calculado diretamente a partir do peso e altura. A relação entre Idade e IMC é positiva, mas fraca a moderada (0.33), indicando um aumento ligeiro do IMC com a idade, possivelmente devido a mudanças metabólicas e de estilo de vida. A frequência de atividade física (FAF) tem uma correlação negativa moderada com o risco de obesidade (-0.42), reforçando a ideia de que a atividade física regular está associada a um menor risco de obesidade. Estes resultados são consistentes com a literatura existente e fornecem uma base sólida para a seleção de atributos e desenvolvimento de modelos preditivos mais precisos para a previsão de IMC e risco de obesidade.

6) Aqui é construído um modelo de regressão linear simples para prever o IMC com base na idade das pessoas. Primeiramente, os dados são preparados, com a variável "Idade" como X e o IMC como y. Foi obtida a seguinte função linear: IMC = 0.2341 * Idade + -0.0040

Os dados são divididos em conjuntos de treino e teste. Em seguida, o modelo de regressão linear é criado e ajustado aos dados de treino.

Após ajustar o modelo, são feitas previsões nos dados de teste. É exibido um gráfico que mostra os dados reais e a linha de regressão linear ajustada para avaliar visualmente o desempenho do modelo.

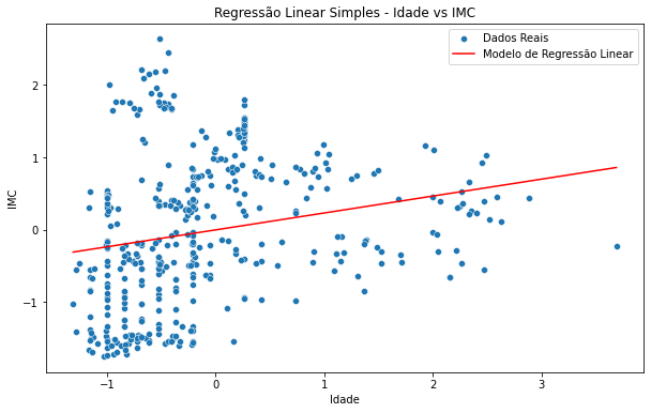

Figura 7 – Regressão Linear Simples

Finalmente, são calculadas as métricas de avaliação do modelo, como o Erro Absoluto Médio (MAE) e o Erro Quadrático Médio (RMSE), para quantificar a precisão do modelo, sendo estes 0,8176 e 0,9801 respetivamente.

Vários modelos de regressão linear simples são depois treinados para prever o IMC usando diferentes variáveis preditoras do conjunto de dados. Um loop é utilizado para iterar sobre cada variável preditora relevante. Para cada variável, o modelo de regressão linear é treinado e avaliado.

Com isto concluimos que a variável do peso terá o melhor resultado, tendo um MAE de 0,2895 e um RMSE de 0,3583.

7) Primeiramente, é ajustado um modelo de regressão linear múltipla usando todas as variáveis disponíveis como preditoras do IMC.

Os dados são divididos em conjuntos de treino e teste, o modelo é criado e ajustado aos dados de treino. Em seguida, são feitas previsões nos dados de teste e calculadas as métricas de avaliação do modelo, como MAE e RMSE, sendo estes 0,072 e 0,0955 respetivamente.

Depois é ajustado um modelo de árvore de regressão com parâmetros padrão. O modelo é treinado usando todas as variáveis disponíveis como preditoras do IMC.

São feitas previsões nos dados de teste e calculadas as métricas de avaliação do modelo, sendo o MAE 0,053 e o RMSE 0,1153.

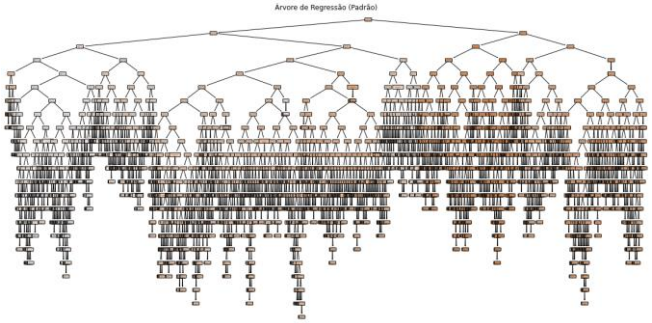

Figura 8 – Árvore de Regressão (Padrão)

Após a árvore com parâmetros padrão, uma árvore de regressão otimizada foi treinada utilizando o GridSearchCV para encontrar os melhores hiperparâmetros. Depois da otimização, o modelo foi treinado novamente com os hiperparâmetros ideais. Os resultados da avaliação do modelo otimizado, incluindo MAE e RMSE, com valores de 0,0941 e 0,1445, respetivamente, foram comparados com o modelo padrão para determinar a melhoria na performance.

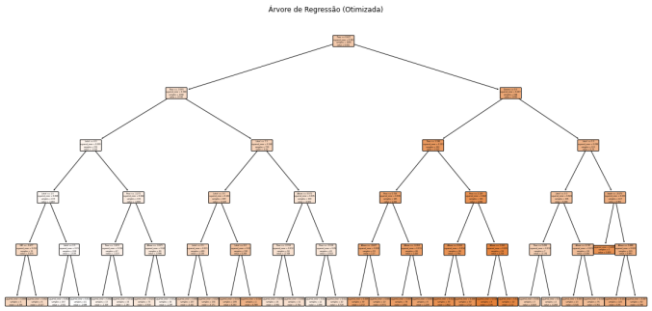

Figura 9 – Árvore de Regressão (Otimizada)

Por fim, são treinadas várias redes neuronais usando diferentes arquiteturas. Para cada arquitetura, o modelo MLPRegressor é criado, ajustado aos dados de treino e avaliado usando as métricas MAE e RMSE nos dados de teste. A melhor configuração de rede neuronal é (100, 50) com MAE de 0,0442 e RMSE de 0,0641.

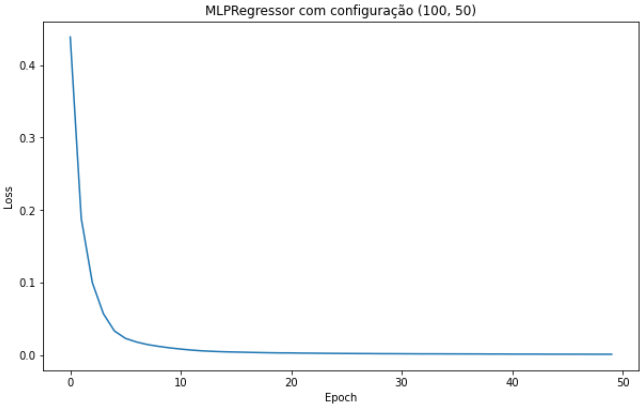

Figura 10 – MLPRegressor com configuração (100, 50)

8) Aqui, os resultados de todos os modelosda questão 7 são comparados. São avaliadas as métricas MAE e RMSE para cada modelo obtendo os seguintes resultados:

	MAE	RMSE
Regressão Linear Múltipla	0.072248	0.095481
Árvore de Regressão (Padrão)	0.053222	0.115326
Árvore de Regressão (Otimizada)	0.094150	0.144513
Rede Neuronal MLPRegressor	0.044161	0.064068

Figura 11 – Comparação de resultados dos modelos da questão 7

9) Neste exercício é realizada uma análise estatística para verificar se há diferenças significativas entre os dois melhores modelos em termos de MAE e RMSE, sendo estes o modelo da regressão linear múltipla e da rede neuronal.

Os p-valores são calculados usando o teste t pareado para determinar se as diferenças observadas são estatisticamente significativas, obtendo os seguintes resultados:

```
P-valor MAE (Regressão Linear Múltipla vs Rede Neuronal MLP): 0.6008
P-valor RMSE (Regressão Linear Múltipla vs Rede Neuronal MLP): 0.0002
```

Figura 12 – P-valor MAE e RMSE entre os modelos de regressão linear múltipla e rede neuronal

Podemos concluir, com α = 0.05, que a diferença entre os modelos não é estatisticamente significativa para MAE mas é estatisticamente significativa para RMSE.

Classificação

1)

-**Árvores de decisão**: Foi realizada uma busca pelos melhores hiperparâmetros ao utilizar *GridSearchCV*. A busca incluiu a profundidade máxima da árvore (max_depth) e o número mínimo de amostras necessárias para dividir um nó (min_samples_split).

A árvore de decisão resultante foi então obtida:

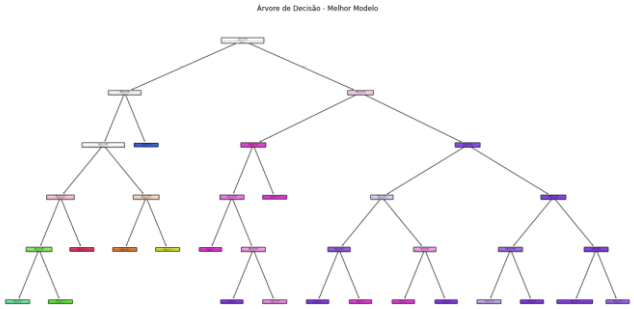

Figura 13 – Árvore de decisão melhor modelo

Esta foi obtida com os seguintes resultados: {'max_depth': 5, 'min_samples_split': 2}; média = 0.9673 e desvio Padrão = 0.0098.

-**SVM**: Foi novamente utilizado o *GridSearchCV* para realizar uma busca pelos melhores hiperparâmetros. Foi obtido no melhor resultado *linear* como melhor kernel, com uma média de 0,9578 e desvio padrão de 0,0135.

- **Rede Neuronal:** Foi desenvolvida uma rede neural utilizando *Keras*. A arquitetura inclui uma camada de entrada, duas camadas densas escondidas com funções de ativação ReLU, e uma camada de saída com função de ativação sigmoide. O modelo foi compilado com a função de perda *binary_crossentropy* e o otimizador *adam*.

Com isto foi obtido, com este modelo, uma média de 0,1554 e desvio padrão de 0,0019.

- **K-vizinhos-mais-próximos:** Mais uma vez foi usado o *GridSearchCV* para encontrar os melhores hiperparâmetros. A incluiu o número de vizinhos (*n_neighbors*) e o tipo de ponderação (*weights*).

Como tal o melhor resultado encontrado (melhor k=5) tem uma média de 0,8186 e um desvio padrão de 0,0259.

a) Primeiramente, identificamos os dois modelos que apresentaram as maiores accuracy de médias durante a validação cruzada. Com isto definimos que os melhores modelos são a árvore de decisão e o SVM.

Para comparar o desempenho dos dois melhores modelos, utilizamos o teste T pareado (ttest_rel).

Obtemos então o p-valor de 0,0319 e, como este é menor que 0,05, podemos afirmar que a diferença entre os modelos é estatisticamente significativa.

b) Foi utilizada a função *calculate_metrics* foi para calcular as métricas desejadas. Esta função utiliza *cross_val_predict* para obter as previsões de validação cruzada, garantindo uma avaliação robusta e imparcial do desempenho do modelo.

Depois de aplicarmos a função aos modelos obtemos os seguintes resultados:

```
Árvore de Decisão - Accuracy: 0.9673, F1: 0.9651, Sensitivity: 0.9673, Specificity: 0.8681
SVM - Accuracy: 0.9578, F1: 0.9578, Sensitivity: 0.9578, Specificity: 0.9291
Rede Neuronal - Accuracy: 0.1554, F1: 0.0418, Sensitivity: 0.1554, Specificity: 0.1111
KNN - Accuracy: 0.8186, F1: 0.8159, Sensitivity: 0.8186, Specificity: 0.7376
```

Figura 14 – Resultados dos modelos de acordo com as métricas Accuracy; Sensitivity; Specificity e F1

O modelo que apresentou o melhor desempenho foi a Árvore de Decisão. Este modelo teve a maior Accuracy, F1 Score e Sensitivity.

O modelo com o pior desempenho foi a Rede Neuronal. Este modelo apresentou os valores mais baixos de Accuracy, F1 Score, Sensitivity e Specificity.

c) Vamos incialmente selecionar apenas os 10 melhores atributos com recurso à função *SelectKBest:*

```
Atributos selecionados: Index(['Genero', 'Idade', 'Peso', 'Historico_obesidade_familiar',
'FCCAC',
       'FCV', 'NRP', 'CCER', 'CBA', 'IMC'],
```

Figura 15 – Melhores atributos selecionados

Depois são gerados novamente os modelos considerando os atributos selecionados para conseguimos então comparar os desempenhos:

```
             Modelo  Accuracy Média  Accuracy Desvio Padrão
0  Árvore de Decisão        0.969210                0.011845
1               SVM        0.957364                0.010392
2      Rede Neuronal        0.155376                0.001872
3               KNN        0.823310                0.028178
Árvore de Decisão (Atributos selecionados) - Accuracy: 0.9692, F1: 0.9670, Sensitivity:
0.9692, Specificity: 0.8694
SVM (Atributos selecionados) - Accuracy: 0.9574, F1: 0.9573, Sensitivity: 0.9574,
Specificity: 0.9540
Rede Neuronal (Atributos selecionados) - Accuracy: 0.1554, F1: 0.0418, Sensitivity: 0.1554,
Specificity: 0.1111
KNN (Atributos selecionados) - Accuracy: 0.8233, F1: 0.8204, Sensitivity: 0.8233,
Specificity: 0.7211
```

Figura 16 – Valores dos modelos que utilizam os atributos selecionados

Como os modelos que utilizam os atributos selecionados têm, na maior parte dos casos, valores maiores de accuracy de média e desvio padrão, assim como nas diferentes métricas, podemos concluir que os modelos obtêm um melhor desempenho se utilizarem apenas alguns dos atributos disponíveis.

2) Inicialmente, vamos derivar dois novos preditores a partir das variáveis existentes: *Altura_Peso* (produto da Altura e Peso) e *IMC_Hist_Obes* (produto do IMC (Índice de Massa Corporal) e o Histórico de Obesidade Familiar).

Depois vamos reavaliar os dois melhores modelos identificados anteriormente.

Realizamos o teste t pareado (*ttest_rel*) comparando a accuracy dos modelos com e sem os novos preditores. Utilizamos um nível de significância de 5% para determinar a significância estatística.

Árvore de Decisão sem os novos preditores- Accuracy: Média = 0.9673, Desvio Padrão = 0.0098
Árvore de Decisão com os novos preditores- Accuracy: Média = 0.9678, Desvio Padrão = 0.0089
SVM sem os novos preditores- Accuracy: Média = 0.9578, Desvio Padrão = 0.0135
SVM com os novos preditores- Accuracy: Média = 0.9536, Desvio Padrão = 0.0143

P-valor da significância do desempenho da árvore de decisão: 0.3434
P-valor da significância do desempenho do SVM: 0.0415

Com um nível de significância de 5%, podemos afirmar que a inclusão dos novos preditores resultou numa melhoria significativa no desempenho do SVM, enquanto não houve uma diferença significativa para a Árvore de Decisão.

3)

a) Primeiramente, preparamos os dados removendo a coluna "Género" do conjunto de atributos X e definindo y como sendo a coluna "Género".

Para prever o género com uma Rede Neuronal, utilizamos a mesma arquitetura otimizada previamente. A avaliação é feita ao utilizar a validação cruzada com 10 folds para garantir a robustez dos resultados.

Obtemos os resultados: Média = 0.9223, Desvio Padrão = 0.0200.

Para prever o género com o SVM, utilizamos o melhor estimador encontrado anteriormente através da Grid Search com validação cruzada. Este modelo também é avaliado utilizando validação cruzada com 10 folds.

Obtemos os resultados: Média = 0.8820, Desvio Padrão = 0.0154.

b) Para verificar a diferença significativa entre os desempenhos dos dois modelos (Rede Neuronal e SVM), utilizamos o teste t pareado (paired t-test).

Realizamos o teste t pareado para comparar as acurácias dos dois melhores modelos. O resultado é apresentado através do p-valor, que neste caso é 0,0001.

Então podemos afirmar que, com um nível de significância de 5%, existe uma diferença estatisticamente significativa entre os desempenhos da Rede Neuronal e do SVM para prever o género.

c) Primeiramente, definimos as métricas para cada modelo utilizando a função *calculate_metrics*.

Após calcular as métricas, comparamos os modelos com base na média das quatro métricas (Accuracy, Sensitivity, Specificity e F1 Score):

	Modelo	Accuracy	Sensitivity	Specificity	F1 Score
0	SVM	0.882046	0.882046	0.882080	0.882050
1	Rede Neuronal	0.921364	0.921364	0.921263	0.921353

Figura 15 – Resultados dos modelos de acordo com as métricas Accuracy; Sensitivity; Specificity e F1

Com base nos resultados apresentados, o modelo de Rede Neuronal apresentou o melhor desempenho geral, com os valores mais altos em Accuracy, Sensitivity, Specificity e F1 Score.

IV. Conclusões

Este estudo demonstra que a seleção e derivação cuidadosa de preditores, junto com a utilização de uma variedade de modelos de machine learning, pode proporcionar uma compreensão abrangente dos dados e melhorar significativamente a precisão das previsões.

A abordagem metodológica utilizada, que incluiu desde a exploração de dados até a validação cruzada e testes de significância estatística, assegurou uma análise robusta e confiável dos modelos preditivos desenvolvidos.

References

[1] N. Chauhan, "Decision Tree Algorithm — Explained," KD Nuggets, 2019. https://www.kdnuggets.com/2020/01/decision-tree algorithmexplained.html (accessed June 06, 2024).

[2] A. Mehta, "A Beginner's Guide to Classification and Regression Trees," Digital Vidya, 2019. https://www.digitalvidya.com/blog/classification-and-regressiontrees/ (accessed June 06, 2024).

[3] A. Pacheco, "K vizinhos mais próximos - KNN | Computação Inteligente," 2017. http://computacaointeligente.com.br/algoritmos/k-vizinhos-maisproximos/ (accessed June 06, 2024).

[4] C. Woodford, "How neural networks work - A simple introduction." https://www.explainthatstuff.com/introduction-to-neuralnetworks.html (accessed June 06, 2024).

[5] R. Joshi, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog," 2016. https://blog.exsilio.com/all/accuracy-precision-recall-f1-scoreinterpretation-of-performance-measures/ (accessed June 06, 2024)

[6] ChatGPT, 2023. https://chat.openai.com/ (accessed June 07, 2024)