Aprendizagem automática

1st Pedro Allen
LEI-DEI
ISEP
Vila Nova de Gaia, Portugal
1211266@isep.ipp.pt

2nd Paulo Reis
LEI-DEI
ISEP
Senhora da Hora, Portugal
1081376@isep.ipp.pt

3rd Rita Azevedo
MEI-DEI
ISEP
Rio Tinto, Portugal
1231439@isep.ipp.pt

***Abstract*—Este artigo aborda o problema da obesidade, incluindo fatores que contibuem para o mesmo. O objetivo principal é desenvolver modelos de regressão e classificação para compreender melhor a relação entre este fatores e a obesidade. O artigo foca principalmente na análise dos dados estudados.**

***Index Terms*—classificação, regressão, obesidade, machine, learning**

I. Introduction

A obesidade é uma doença crónica caracterizada pelo excesso de gordura acumulada no organismo, tem crescido globalmente e que acarreta consequências para a saúde do indivíduo. Existem diversos fatores que podem contrubuir para o desenvolvimento da obesidade, incluindo, hábitos alimentares inadequados, sedentarismo, tabagismo e um histórico familiar de obesidade.

O conjunto de dados fornecido inclui informações detalhadas sobre hábitos alimentares, idade, peso, altura e outras características de indivíduos de diferentes faixas etárias. O objetivo é desenvolver e comparar modelos de regressão e classificação para entender melhor a relação entre esses fatores e a obesidade.

O conjunto de dados foi inicialmente estudado e preparado, passando por processos de limpeza, normalização e análise exploratória. Em seguida, foram aplicados diversos modelos de aprendizagem automática para analisar os dados. Os resultados obtidos foram examinados com o objetivo de tirar conclusões e fazer previsões sobre como os hábitos alimentares e as características individuais influenciam o risco de obesidade. O estudo pretende fornecer insights que possam ajudar na prevenção e tratamento da obesidade.

II. Equipa de desenvolvimento

No desenvolvimento e documentação deste trabalho a equipa é composta pelos autores deste artigo. As tarefas foram dividas entre os mesmos, constado na tabela 1 o valor em %

TABLE I
Atribuição de Tarefas

Nome	%
Paulo Reis	33.3
Pedro Allen	33.3
Rita Azevedo	33.3

III. Estado da Arte

A. Regressão

A regressão é uma técnica estatística usada para modelar a relação entre uma variável dependente e uma ou mais variáveis independentes. Existem dois tipos principais: regressão simples e regressão múltipla.

B. Regressão Simples

A regressão simples é usada para modelar a relação entre uma variável dependente e uma única variável independente. A fórmula básica da regressão linear simples é:

$$y = \beta_0 + \beta_1 x + \epsilon \tag{1}$$

onde y é a variável dependente, x é a variável independente, β_0 é o intercepto, β_1 é o coeficiente de inclinação, e ϵ é o termo de erro. Este método é eficaz para prever o valor da variável dependente com base na variável independente.[1].

C. Regressão Múltipla

A regressão múltipla, por outro lado, é utilizada para modelar a relação entre uma variável dependente e duas ou mais variáveis independentes. A fórmula básica da regressão linear múltipla é:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon \tag{2}$$

onde y é a variável dependente, $x_1, x_2, \ldots, x_k$ são as variáveis independentes, β_0 é o intercepto, $\beta_1, \beta_2, \ldots, \beta_k$ são os coeficientes de inclinação, e ϵ é o termo de erro. Este método permite capturar a influência combinada de múltiplos fatores sobre a variável dependente. [2].

A regressão múltipla pode ser usada para identificar a importância relativa de cada variável independente na predição da variável dependente e é frequentemente utilizada em análises de previsão e explicativas[3].

D. Árvore de decisão

Uma árvore de decisão é uma estrutura de dados hierárquica que representa dados através de uma estratégia de divisão e conquista. [1]

Árvores de decisão são classificadores para instâncias representadas como vetores de características. Os nós são os testes para os valores das características, as folhas especificam o

rótulo e em cada nó deve haver um ramo para cada valor da característica. [1]

Podem representar qualquer função linear e ser pensadas como uma disjunção de conjunções, ou reescritas como regras em Forma Normal Disjuntiva (FND). [1]

E. Suporte Vector Machine(SVM)

As SVMs são modelos não paramétricos, o que não significa que estes não contenham parâmentros. Foram desenvolvidas em ordem inversa ao desenvolvimento das redes neurais, ou seja evoluíram a partir de uma teoria sólida para a implementação e experimentação, enquanto as redes neurais seguiram um caminho mais heurístico, partindo de aplicações e experimentação extensiva para a teoria. [2]

As SVMs possuem a conhecida capacidade de serem aproximadores universais de qualquer função multivariada com qualquer grau de precisão desejado. [2]

A configuração do problema de aprendizado para SVMs é definida por uma relação desconhecida e não linear (mapeamento ou função) entre um vetor de entrada de alta dimensionalidade x e uma saída escalar y (ou uma saída vetorial y no caso de SVMs multiclasse). Não são fornecidas informações sobre as funções de probabilidade subjacentes. Portanto, é necessário realizar uma aprendizagem livre de distribuição. A única fonte de informação disponível é um conjunto de dados de treinamento $D = \{(x_i, y_i) \in X \times Y\}$, onde $i = 1, \ldots, l$, representando o número de pares de dados de treinamento e, portanto, o tamanho do conjunto de dados de treinamento D. Comumente, y_i é denotado como d_i, onde d representa um valor desejado ou alvo. Portanto, as SVMs são classificadas como técnicas de aprendizado supervisionado. [2]

F. Rede Neuronal

Uma rede neural é uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa ou função de interesse específico, geralmente a rede é implementada usando componentes eletrônicos ou é simulada em software em um computador digital. Para alcançar um bom desempenho, as redes neurais empregam uma interconexão maciça de células de computação simples, denominadas ”neurônios” ou ”unidades de processamento”. [3]

Uma rede neural obtém sua capacidade de computação, primeiro, através de sua estrutura distribuída maciçamente paralela e, segundo, por sua capacidade de aprender e, portanto, generalizar. [3]

G. K-Nearest Neighbour (k-NN)

Na Nearest Neighbour Classification ou em português Classificação pelo vizinho mais próximo, os exemplos são classificados com base na classe dos seus vizinhos mais próximos. É tido em consideração mais de um vizinho, sendo a técnica dada pelo nome k-Nearest Neighbour (k-NN) onde os k vizinhos mais próximos são usados para determinar a classe. [4]

É possível criar um classificador k-NN que utilize uma medida de afinidade que não seja uma métrica formal. No entanto, certas otimizações de desempenho para o algoritmo k-NN básico requerem o uso de uma métrica adequada [5,6]. Em resumo, essas técnicas conseguem identificar o vizinho mais próximo de um objeto sem precisar compará-lo com todos os outros objetos, mas para isso, a medida de afinidade deve ser uma métrica e, em particular, deve obedecer à desigualdade triangular [7].

IV. Exploração do dataset

TABLE II
Descrição dos Atributos

Nome da Coluna	Designação
Genero	Gênero do indivíduo
Idade	Idade do indivíduo
Altura	Altura do indivíduo, medida em metros.
Peso	Peso do indivíduo, medido em quilogramas.
Historico_obesidade_familiar	Indica se há histórico de obesidade na família.
FCCAC	Frequência de Consumo de Comida Altamente Calórica.
FCV	Frequência de Consumo de Vegetais.
NRP	Número de Refeições Principais.
CCER	Consumo de Comida Entre Refeições.
Fumador	Indica se o indivíduo é fumador.
CA	Consumo de Água.
MCC	Monitorização do Consumo de Calorias.
FAF	Frequência de Atividade Física.
TUDE	Tempo de Utilização de Dispositivos Eletrônicos.
CBA	Consumo de Bebidas Alcoólicas.
TRANS	Tipo de Transporte Utilizado.

O conjunto de dados fornecido para este estudo consiste em várias informações coletadas de indivíduos de diferentes faixas etárias e características físicas, permitindo a análise detalhada de fatores que contribuem para a obesidade. As informações contidas no conjunto de dados estão resumidas na tabela com as respectivas designações.

Foram analisados os dados relacionados com a obesidade ao longo dos anos, verificando-se diversas tendências e padrões. O conjunto de dados inclui informações detalhadas sobre hábitos alimentares, características físicas e demográficas de indivíduos de diferentes faixas etárias. Esta análise à priori visa entender melhor os fatores que influenciam a obesidade e fornecer informações para o desenvolvimento de modelos preditivos eficazes.

Primeiramente, observamos que a maioria dos indivíduos está concentrada na faixa etária entre 20 e 40 anos, o que pode indicar um grupo demográfico significativo para estudos sobre obesidade e intervenções preventivas. A distribuição do Índice de Massa Corporal (IMC) mostrou uma distribuição normal, com alguns outliers, indicando que, embora a maioria dos indivíduos tenha um IMC dentro da faixa esperada, há casos extremos que necessitam de atenção especial.

A matriz de correlação não destaca qualquer relação forte entre os vários preditores, e o próprio alvo, pelo que sempre

que for tecnicamente possível, iremos utilizar todos os atributos

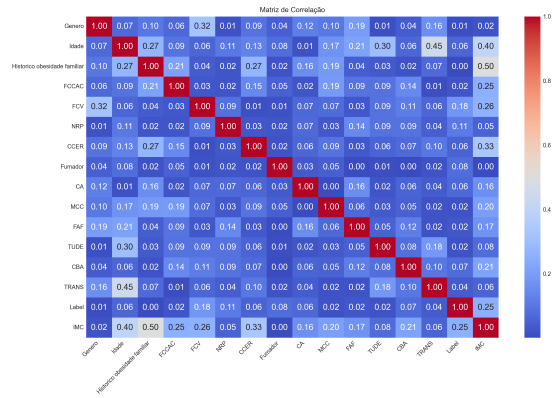

Fig. 1. Matriz de Correlação

Além disso, após análise de todos os valores pertencentes à matriz de correlação, destacam-se, no máximo, correlações moderadas, sendo estas a idade e uso de transportes e histórico de obesidade familiar com o próprio alvo. Este último atributo, numa análise simplista, mostra-se como o melhor preditor de obesidade, corroborando estudos que apontam o histórico familiar de obesidade como um importante preditor da condição [1].

V. Preparação dos dados

Para a preparação inicial dos dados, uma vez que estes iriam ser utilizados de forma diferente para cada algoritmo, não foi aplicada uma codificação uniforme das classes. Foi verificado a não existência de NaN e nulls, pelo que não necessário qualquer correção. Avaliou-se a quantidade de classes em cada atributo não numérico. Na maioria dos algoritmos utilizou-se um *LabelEncoder* que atribui valores numéricos a cada classe, sendo pois necessário realizar um processo de normalização dos valores para eliminar os efeitos da escala desse atributo. Em alguns dos algoritmos, este processo não é adequado pelo que se aplicou o *Pandas.get_dummies*, que cria as várias colunas para cada classe do atributo, tomando o valor 1 quando essa classe se verifica, isto cria-nos n colunas quando temos n classes mas existe colineridade entre elas, exigindo assim a eliminação de uma das colunas dummy.

VI. Regressão Linear Simples

Nas metedologias da Regressão é preciso ter um alvo (variavél dependente ”y”) e um conjunto de variavéis independente (X). Na Regressão Linear Simples o conjunto X só tem um atributo. Para o treino do modelo, foram utilizados 80% dos dados, ficando os restantes 20% para teste, este processo denomina-se por hold-out. De forma a realizar esta divisão de forma aleatória, utiliza-se a função *train_teste_split*. Depois de instanciar o modelo (LinearRegression) é realizado o treino usando a função *fit* deste e a previsão com função *predict*. Com estes resultados, calculam-se as métricas de avaliação do modelo.

TABLE III
Tabela para os valores de MAE e RMSE

Atributo	MAE	RMSE
Idade	6.38	7.77
Genero	6.58	7.98
HOF	5.73	7.075
FCCAC	6.41	7.73
FCV	6.40	7.74
NRP	6.58	8.00
CCER	6.01	7.39
Fumador	6.60	8.01
CA	6.55	7.97
MCC	6.39	7.80
FAF	6.49	7.87
TUDE	6.56	7.96
CBA	6.53	7.77
TRANS	6.60	7.98

A tabela anterior apresenta as métricas da Regressão Linear simples entre o IMC e cada um dos atributos.

VII. Regressão Linear Multipla

A metedologia de execução do algoritmo assemelhou-se à anterior descrita, a diferença está em que existem várias variáveis independentes, pelo que apresentamos apenas os resultados das métricas de erro.

TABLE IV
Métricas nos valores de Teste

R squared	47.835
MAE	4.5282
MSE	31.935
RMSE	5.6511

VIII. Árvore de decisão

Semelhante aos anteriores em execução, a diferença está na função invocada: *DecisionTreeRegressor*.

TABLE V
Métricas nos valores de Teste

MAE	2.911
RMSE	3.496

IX. MLPRegressor

Este algoritmo tem uma maior quantidade de parâmetros para afinação apesar da sua execução unitária ser igual ao anterior. Foi realizada uma série de cálculos (1992) usando a combinação dos vários algoritmos de ativação, solvers e estrutura de camadas internas (escondidas). Verificou-se que, independentemente de usar os mesmos dados de treino e teste e as mesmas parameterizações, os resultados podem ser diferentes a cada execução, com isto, a nossa escolha incidiu sobre o resultado da última execução realizada, cujos resultados se encontram no ficheiro *resultados_MLPRegressor.csv*.

X. Comparação dos métodos

Pela análise de métricas, considerou-se que o melhor modelo seria aquele que apresentasse menor valor da métrica em estudo, tendo assim escolhida a RMSE. Conclui-se, pelos resultados apresentados, que os dois melhores modelos seriam o MLRRegressor e LinearRegression. Sobre estes dois, recolhemos 50 amostras de execução sobre conjuntos treino-teste aleatórios, e aplicou-se um t-test para amostras emperalhadas. Após análise do valor de p-value sobre um nível de significância de 0.05 inferimos que não há diferenças significativas no resultados dos métodos.

XI. Classificação

Para os modelos de classificação, realizámos inicialmente um conjunto de procedimentos, como a transformação das colunas categóricas em colunas binárias. Esta etapa é necessária para garantir que os algoritmos de classificação possam processar corretamente os dados categóricos, convertendo-os em um formato numérico adequado.

De seguida procedemos ao treino dos nossos dados. Primeiro, definimos a nossa variável alvo (y), que representa os níveis de obesidade. Em seguida, separamos 20% dos dados para teste e utilizámos os restantes 80% para treino.

A. Arvore de decisão

Inicialmente construimos a nossa árvore de decisão usando apenas o random_state de valor 42, que é o valor padrão utilizado. Para ajustar os valores obtendo melhores resultados, fizemos o sobreajuste, tentando encontrar um valor indicado para a profundidade máxima da árvore e número de amostras mínimas para os nós da árvore, o que permite consilidar os dados de teste e de treino. Após essa implementação a árvore ficou com profundidade 5 e 16 amostras mínimas.

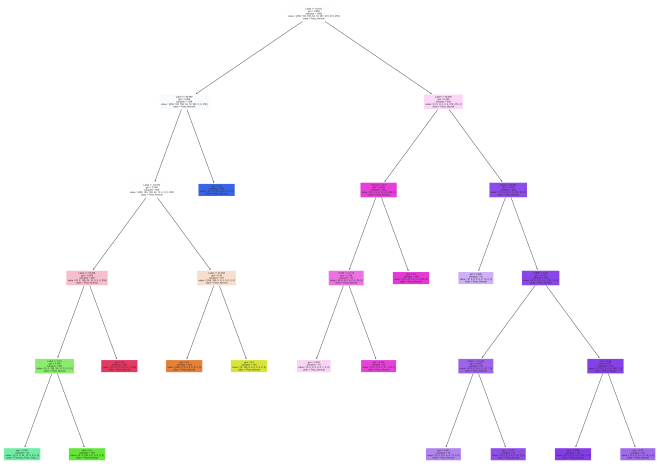

Fig. 2. Árvore de decisão

Para avaliar o modelo, utilizamos uma série de técnicas.

Inicialmente, calculamos a precisão dos conjuntos de dados de teste e de treinamento, obtendo um valor de 0,98 o que indica uma alta taxa de previsões corretas feitas pelo modelo. Além disso, avaliamos a acurácia do modelo nos conjuntos de dados, obtendo valores de 0,98 para os dados de treinamento e 0,98 para os dados de teste. Esses resultados sugerem que o modelo está apresentando um excelente desempenho, sendo capaz de generalizar bem para novos dados além dos dados de treinamento, com uma taxa de previsões corretas muito alta em ambos os conjuntos de dados.

Após analisar a matriz de confusão e calcular a taxa de erro para os conjuntos de dados de treinamento e teste, não identificamos nenhum erro significativo. Todos os valores na matriz foram verdadeiros positivos e verdadeiros negativos, sugerindo que o modelo está a fazer previsões precisas e acertadas em ambos os conjuntos de dados.

Ao analisar o relatório de confusão, observamos uma acurácia de 0,98, o que indica que o modelo está acertando 98% das previsões. No entanto, para a classe 4, notamos que tanto o recall, precisão quanto o F1-score são iguais a 0, indicando que o modelo não conseguiu identificar corretamente nenhuma amostra para essa classe. Para as restantes classes, os valores de recall, precisão e F1-score são altos, com a maioria acima de 0,95 ou igual a 1, o que significa que o modelo está fazendo previsões precisas e confiáveis para essas classes.

Após realizar a avaliação k-fold, observamos que os valores para cada fold variam entre 0,90 e 1. A acurácia média foi de 0,95, com um desvio padrão de 0,049. Esses resultados indicam um desempenho geral bastante consistente do modelo durante a validação cruzada, com uma alta acurácia média e um desvio padrão relativamente baixo. Isso sugere que o modelo é robusto e está generalizando bem para diferentes conjuntos de dados.

B. Rede Neuronal

Para a construção da rede neural, começamos por definir uma semente aleatória para garantir a consistência dos resultados. Em seguida, efetuamos o pré-processamento dos dados, garantindo que tanto os dados de treino quanto os de teste estivessem padronizados.

Passamos, então, ao desenvolvimento do modelo utilizando a biblioteca Keras. A rede neural é composta pela camada de entrada com 15 características, correspondentes ao número de colunas do dataset, excluindo a variável dos níveis de obesidade.

A primeira camada oculta é uma camada densa com 64 neurónios, utilizando a função de ativação ReLU para capturar as não-linearidades presentes no conjunto de dados. Nesta camada, utilizamos a regularização kernel para penalizar pesos elevados e evitar overfitting. A primeira camada dropout desativa aleatoriamente 50% dos neurónios durante cada iteração do treino, ajudando a evitar que o modelo se ajuste excessivamente aos dados de treino.

A segunda camada densa possui 32 neurónios, também utilizando a ativação ReLU, e inclui regularização L2 para controlar o overfitting. A segunda camada dropout aplica

novamente uma taxa de 50% na segunda camada densa, proporcionando robustez ao modelo para prevenir overfitting.

A camada final é composta por 9 neurónios, correspondendo a cada uma das classes dos graus de obesidade. Esta camada utiliza a ativação softmax, uma vez que se trata de um problema de classificação multiclasse.

Inicialmente, o modelo contava apenas com duas camadas ocultas cada uma com ativação ReLU, cada uma com apenas 3 neurónios. A utilização de mais neurónios e camadas permitiu obter melhores resultados, pois o modelo conseguiu identificar padrões mais complexos nos dados.

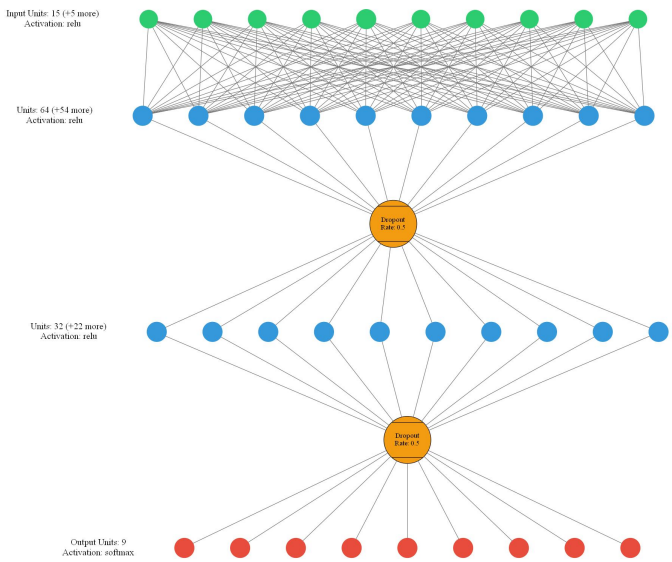

Fig. 3. Rede Neuronal

Para realizar previsões utilizando o modelo, utilizamos as bibliotecas scikit-learn e MLPRegressor, aplicando as funções de ativação tangente hiperbólica (tanh) e o algoritmo de otimização Limited-memory Broyden-Fletcher-Goldfarb-Shanno (lbfgs). Os dados são treinados e, em seguida, aplicamos o modelo para fazer previsões nos dados de teste. Calculamos o coeficiente de determinação e o RMSE (Erro Quadrático Médio) para avaliar o desempenho do modelo. Obtivemos um coeficiente de determinação de 0.22, indicando que o modelo explica 22% da variabilidade nos dados de teste, sugerindo que não está a capturar adequadamente os padrões presentes nos dados. Portanto, são necessárias melhorias significativas. Quanto ao RMSE, foi observado um valor de 2.43. Ao observar os dados, constatamos que o valor a ser previsto é ”Magreza Grau I”.

Para avaliar o modelo, utilizamos um conjunto de técnicas.

Primeiramente, analisamos as métricas, começando pela observação das curvas de perda do treino e validação da rede neural ao longo do tempo, onde constatamos que a perda está diminuindo gradualmente. Em seguida, examinamos com mais detalhe a acurácia do treino e validação do modelo, que aumenta à medida que o treinamento avança. Em resumo, a interpretação conjunta dos dois gráficos sugere que o modelo está aprendendo bem os dados, sem apresentar uma grande perda de desempenho.

Analisamos o relatório de classificação e encontramos uma acurácia de 0,77, indicando que o modelo está fazendo previsões corretas em cerca de 77% das amostras. No entanto, observamos que as classes 2 e 3 apresentaram um desempenho significativamente inferior, com valores de recall, precisão e F1-score iguais a 0. Isso sugere que o modelo teve dificuldades em identificar corretamente as amostras dessas classes. Por outro lado, para as restantes classes classes, os valores de recall, precisão e F1-score variaram entre 0,63 e 0,91, indicando um desempenho melhor em equilibrar precisão e recall. Em resumo, enquanto o modelo se saiu bem para a maioria das classes, ele enfrentou desafios específicos ao lidar com as classes 2 e 3.

Por último, realizamos uma avaliação k-fold. Os valores para cada fold variam entre 0.0094 e 0.2156. A acurácia média foi de 0.1398, com um desvio padrão de 0.0797. Esses resultados não são considerados satisfatórios, indicando a necessidade de melhorias.

O facto de nas métricas estar a mostrar valores nulos para as classes 2 e 3, respetivas a ”Magreza Grau I” ”Magreza Grau II” mas as previsões preverem a classe 2 é um caso a ser analisado e corrigido.

C. K-Nearest Neighbors (k-NN)

Criámos o classificador KNN utilizando a distância de Manhattan e definimos 50 vizinhos para análise, com um incremento de 1. Para cada valor de k, calculámos o MSE, a matriz de confusão e a acurácia das previsões. Ao analisarmos as acurácias dos 50 vizinhos, obtivemos valores que variaram entre 0.79 e 0.65, sendo o vizinho 1 o que apresentou a melhor acurácia.

Após identificar o melhor vizinho, juntamente com os valores do erro (MSE), a matriz de confusão e a acurácia, comparámos os resultados para cada vizinho. O vizinho 1 destacou-se com uma acurácia de 0.794.

Observámos detalhadamente a matriz de confusão do vizinho 1 e verificámos a acurácia para cada classe. A classe ”Excesso de Peso I” obteve a melhor acurácia. Por outro lado, uma das classes teve apenas uma instância corretamente classificada, enquanto outra classe não teve nenhuma classificação correta.

D. Support Vector Machine(SVM)

Além de tambem se usar label encoder para o SVM utilizamos o fit.transform onde cada valor único da coluna é mapeado para um inteiro diferente.

O nosso modelo SVM realiza um grid search para otimizar os seus parâmetros. Utilizamos o parâmetro de regularização C e as funções kernel para projetar os dados em um espaço de características superior, com os possíveis valores linear, polinomial, sigmoid e Radial Basis Function (RDF). O parâmetro gamma define o quanto uma única amostra influencia o modelo. Os melhores parâmetros encontrados pelo modelo

foram: regularização 1000, gamma 1 e kernel linear, os quais foram utilizados para fazer previsões.

O modelo obteve uma acurácia e um valor F1-score de 96%, o que indica um alto desempenho e robustez do modelo.

E. Observações

Ao remover as colunas 'FCCAC', 'TRANS', 'CA', 'MCC', 'Idade', 'CCER', 'NRP' e 'FCV' nos diferentes modelos, observou-se uma pequena melhoria nos valores das acurácias. No entanto, a melhoria foi mais notável em alguns modelos do que em outros. Por exemplo, a árvore de decisão mostrou apenas uma melhoria quase imperceptível.

Por outro lado, ao remover a coluna do IMC, houve uma queda acentuada nas acurácias para todos os modelos. Esta observação sugere que o IMC é uma característica importante para a precisão dos modelos.

Analisámos cada um dos modelos utilizando a métrica da precisão para determinar qual é o melhor. Com valores de precisão de 98% para a árvore de decisão, 77% para a rede neural, 79% para o método k-neighbors e 96% para o SVM, concluímos que o modelo da árvore de decisão demonstrou o melhor desempenho, registando a maior precisão entre todos os modelos testados. Por outro lado, a rede neuronal revelou o pior desempenho, seguida pelo k-neighbors, que obteve o segundo valor de precisão mais baixo.

XII. Conclusão

Após cuidada análise de toda a informação recolhida durante a elaboração deste trabalho, conclui-se que relativamente à obesidade, o melhor fator de previsão é Histórico de Obesidade Familiar pelo o estudo da Regressão confirmou a nossa tese inicial.

Nos modelos de classificação, alcançamos bons resultados de acurácia com os modelos da árvore de decisão e SVM, enquanto os modelos da rede neural e k-neighbors não obtiveram desempenhos tão satisfatórios. Observamos que o IMC foi o atributo mais influente nos modelos, indicando sua importância na construção das previsões.

Os modelos com acurácia mais baixa precisam de melhorias. Num futuro, poderemos efetuar melhorias nos modelos, realizar mais previsões e aplicar os mesmos métodos de avaliação a todos os modelos.

References

1 Romero-Ibarguengoitia ME, Vadillo-Ortega F, Caballero AE, et al. "Family history and obesity in youth, their effect on acylcarnitine/aminoacids metabolomics and non-alcoholic fatty liver disease (NAFLD)." PLoS ONE, vol. 13, no. 2, 2018. https://doi.org/10.1371/journal.pone.0193138

1 D. Montgomery, E. Peck, and G. Vining, "Introduction to Linear Regression Analysis," 5th ed., Wiley, 2012. Disponível em: Wiley

2 A. Gelman and J. Hill, "Data Analysis Using Regression and Multilevel/Hierarchical Models," Cambridge University Press, 2006. Disponível em: Cambridge University Press

3 A. J. Dobson, "An Introduction to Generalized Linear Models," 3rd ed., CRC Press, 2008. Disponível em: CRC Press

3 D. Roth. Decision Trees. 2016. Available: https://www.cis.upenn.edu/ danroth/Teaching/CS446-17/LectureNotesNew/dtree/main.pdf

4 V. Kecman. Basics of Machine Learning by Support Vector Machines. Virginia Commonwealth University. Virginia. May 2005

5 S. Haykin. Neural Networks and Learning Machines Third Edition. Canada. 2009

6 P. Cunningcham, S. J. Delany. K-Nearest Neighbour Classifiers. Dublin. March 2007.

7 A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In Proceedings of 23rd International Conference on Machine Learning (ICML 2006), 2006.

8 J.W. Schaaf. Fish and Shrink. A Next Step Towards Efficient Case Retrieval in Large-Scale Case Bases. In I. Smith and B. Faltings, editors, European Conference on Case-Based Reasoning (EWCBR'96, pages 362–376. Springer, 1996

9 P. Cunningham, S. J. Delany. K-Nearest neighbour classifiers. Dublin. 2007