Desarrollo de un sistema de aprendizaje automático para la detección y clasificación de situaciones de obesidad

Iago Grandal Del Río, Hugo Bouzón Fernández, Óscar Vilela Rodríguez

April 2024

1 Introducción

La obesidad es una enfermedad en la que el paciente adquiere un exceso de grasa, que puede estar relacionado con distintos trastornos de la alimentación. Este problema afecta a millones de personas en el planeta, especialmente en la actualidad, pues se ha visto un aumento de casos que sobrepasa límites históricos. Esta enfermedad supone un grave riesgo para la salud, ya que es muy probable que derive en situaciones amenazantes para la vida, lo que provoca que su detección y tratamiento sean esenciales de detectar en etapas tempranas.

La inteligencia artificial, por su parte, ha visto un fuerte auge en los últimos años, destacando su aplicación a problemas de clasificación con técnicas como redes de neuronas o árboles de decisión.
Es una ocurrencia natural combinar estos dos factores, aplicando la inteligencia artificial y todo su potencial al ámbito médico, para tratar de propiciar la detección temprana de problemas de salud, en este caso, la obesidad.

El objetivo de este proyecto consta no solo de desarrollar dicho sistema que aplique aprendizaje automático para detectar y clasificar situaciones de obesidad, sino que también se realizará una comparativa de distintos tipos de modelos, probando también distintas combinaciones de hiperparámetros, para tratar de hallar una buena configuración. Los tipos de modelos son redes de neuronas artificiales (RRNNAA), máquinas de soporte vectorial (SVM), árboles de decisión (DT) y "k nearest neighbors" (kNN).

2 Descripción del problema

Para entrenar los distintos modelos de aprendizaje automático se han extraído las instancias de un dataset público, obtenida del repositorio "UCI Machine Learning Repository"[6].

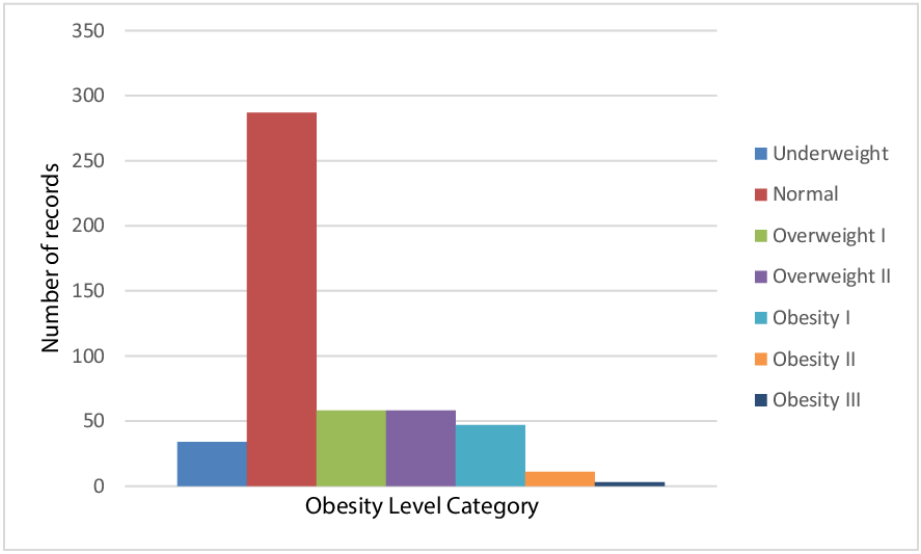
El dataset en cuestión contiene numerosas instancias útiles para este entrenamiento. Fue publicado por Fabio Mendoza Palechor y Alexis de la Hoz Manotas, en el año 2019 [8], y contiene numerosos datos recopilados a través de una página web en algunos países de América del Sur, en concreto México, Perú y Colombia.

2.1 Concepción de instancias

Estos datos, por estar fuertemente desbalanceados en favor de la categoría "Normal", fueron sometidos a un proceso de creación de instancias sintéticas, aplicándoles un filtro SMOTE. Estos datos se crearon mediante la herramienta

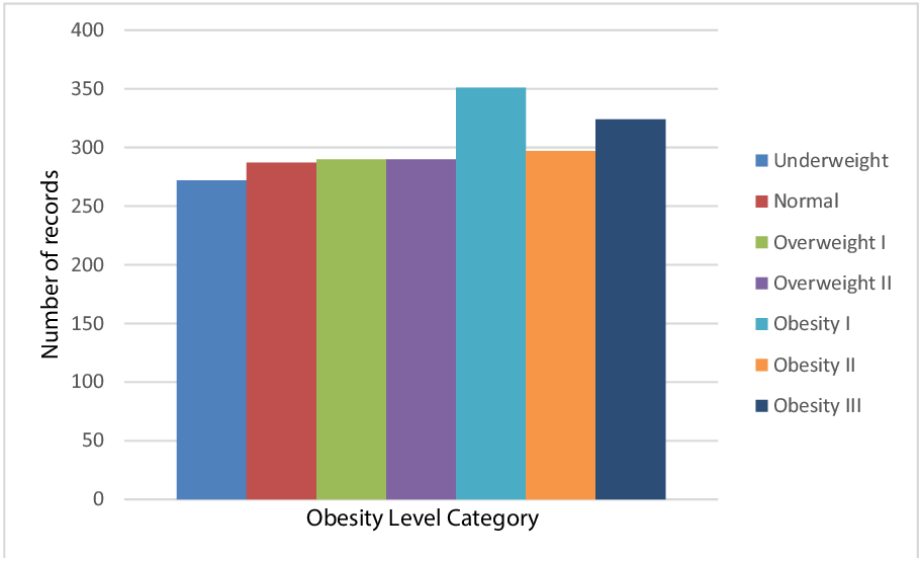
"Weka" [10], y componen un 77% de las instancias totales en el dataset final.

Figure 1: Distribución de instancias antes de balancear sintéticamente



1

Figure 2: Distribución de instancias tras balancear sintéticamente



2

A la vista de estos esfuerzos por equilibrar las instancias, la medida de pref-

erencia para comparar la eficacia de los modelos es la precisión, ya que funciona bien en los casos en los que conocemos que el dataset está balanceado.

2.2 Descripción de los datos

En su versión definitiva, el dataset contiene 2111 instancias distribuidas como se muestra en el gráfico anterior, con las categorías "Underweight", "Normal", "Overweight I", "Overweight II", "Obesity I", "Obesity II" y "Obesity III". Existen 17 atributos en el dataset, como se describen a continuación:

Figure 3: Características presentes en el dataset y sus posibles valores

Questions	Possible Answers
¿What is your gender?	• Female • Male
¿what is your age?	Numeric value
¿what is your height?	Numeric value in meters
¿what is your weight?	Numeric value in kilograms
¿Has a family member suffered or suffers from overweight?	• Yes • No
¿Do you eat high caloric food frequently?	• Yes • No
¿Do you usually eat vegetables in your meals?	• Never • Sometimes • Always
¿How many main meals do you have daily?	• Between 1 y 2 • Three • More than three
¿Do you eat any food between meals?	• No • Sometimes • Frequently • Always
¿Do you smoke?	• Yes • No
¿How much water do you drink daily?	• Less than a liter • Between 1 and 2 L • More than 2 L
¿Do you monitor the calories you eat daily?	• Yes • No
¿How often do you have physical activity?	• I do not have • 1 or 2 days • 2 or 4 days • 4 or 5 days
¿How much time do you use technological devices such as cell phone, videogames, television, computer and others?	• 0–2 hours • 3–5 hours • More than 5 hours
¿how often do you drink alcohol?	• I do not drink • Sometimes • Frequently • Always
¿Which transportation do you usually use?	• Automobile • Motorbike • Bike • Public Transportation • Walking

3

Nótese que la imagen incluye 16 características, ya que la restante es el tipo de obesidad descrito anteriormente. La correspondencia de estas características con cómo que aparecen en el dataset es la siguiente:

Table 1: Correspondencia en nombres de características

¿what is your gender?	Gender
¿what is your age?	Age
¿what is your height?	Height
¿what is your weight?	Weight
¿Has a family member suffered or suffers from overweight?	family_history_with_overweight
¿Do you eat high caloric food frequently?	FAVC
¿Do you usually eat vegetables in your meals?	FCVC
¿How many main meals do you have daily?	NCP
¿Do you eat any food between meals?	CAEC
¿Do you smoke?	SMOKE
¿How much water do you drink daily?	CH2O
¿Do you monitor the calories you eat daily?	SCC
¿How often do you have physical activity?	FAF
¿How much time do you use technological devices such as cell phone, videogames, television, computer and others?	TUE
¿How often do you drink alcohol?	CALC
¿Which transportation do you usually use?	MTRANS
Obesity level	NObeyesdad

Este dataset no presenta restricciones o inconvenientes relevantes para el problema a tratar.

3 Análisis bibliográfico

Este problema ha sido abordado con anterioridad desde distintos puntos de vista, incluso dentro del ámbito del aprendizaje máquina.

Un ejemplo de esto puede ser el trabajo de Harika Gozde Gozukara Bag, Fatma Hilal Yağın, Yasin Gormez, Pablo Prieto González, Cemil Çolak and Mehmet Gülü, Georgian Badicu y Luca Paolo Ardigò [1]. Se trata de realizar un sistema similar con técnicas distintas, como "Extreme Gradient Boosting" (XGBoost), "Random Forest" (RF) y regresión logística (LR). Su aproximación, que involucró "Recursive Feature Elimination" y optimización bayesiana, demostró la eficacia de la LR gracias a estas técnicas.

Algunos de los mismos autores decidieron llevar a cabo otra investigación [11], esta vez llevando la optimización bayesiana a las RRNNAA. Este trabajo, a pesar de no obtener comparativas directas entre modelos de aprendizaje automático, fue capaz de determinar los factores que más repercuten en la prevención de la obesidad. No es un resultado negativo, ya que este fue el foco principal de la investigación en lugar de centrarse en comparar distintos modelos. Además, esta aproximación incluye uno de los modelos en los que se centra nuestro estudio, por lo que puede ser útil tomarlo en cuenta.

Un trabajo interesante es el que presenta Akash Choudhuri [3], que utiliza un modelo híbrido de aprendizaje automático. Este modelo aprovecha varias técnicas, como "Extremely Randomized Trees", perceptrón multicapa y XGBoost. Como factor adicional, el proyecto fue desarrollado en Python.

En la investigación de Diayasa, I Gede Susrama Mas y Idhom, Mohammad and Fauzi [4] se puede ver cómo se aplican modelos de SVM, kNN, "Gradient Boosting" y RF, además de "Stacking Ensemble", a este problema. Sus resultados indican que la mejor precisión la obtuvo el modelo de "Gradient Boosting" por encima de los demás.

En el trabajo de Duwi Cahya Putri Buani and Nia Nuraeni [2] se realiza un estudio muy completo y a fondo, comparando 7 tipos distintos de modelos: "Naive Bayes" (NB), RF, kNN, DT, SVM y XGB Classifier, siendo este último modelo el que obtuvo las mejores métricas.

Mahmut Dirik [5] investiga una cantidad incluso mayor de modelos, 10 en total. Estos incluyen perceptrón multicapa, SVM, algunos modelos difusos, árboles, etc. Este trabajo profundiza en estos modelos, comparando varias de sus métricas, como el "kappa statistic" para determinar que RF consigue los mejores resultados. El estudio asegura la validez de estos modelos como herramienta para los profesionales de la salud.

Aplicando Deeplearning, Mehmet Kıvrak [7] publica su trabajo, determinando que su modelo muestra una precisión de 0.82 con la confiiguración óptima de hiperparámetros, un valor bajo al compararlo con resultados de otros modelos.

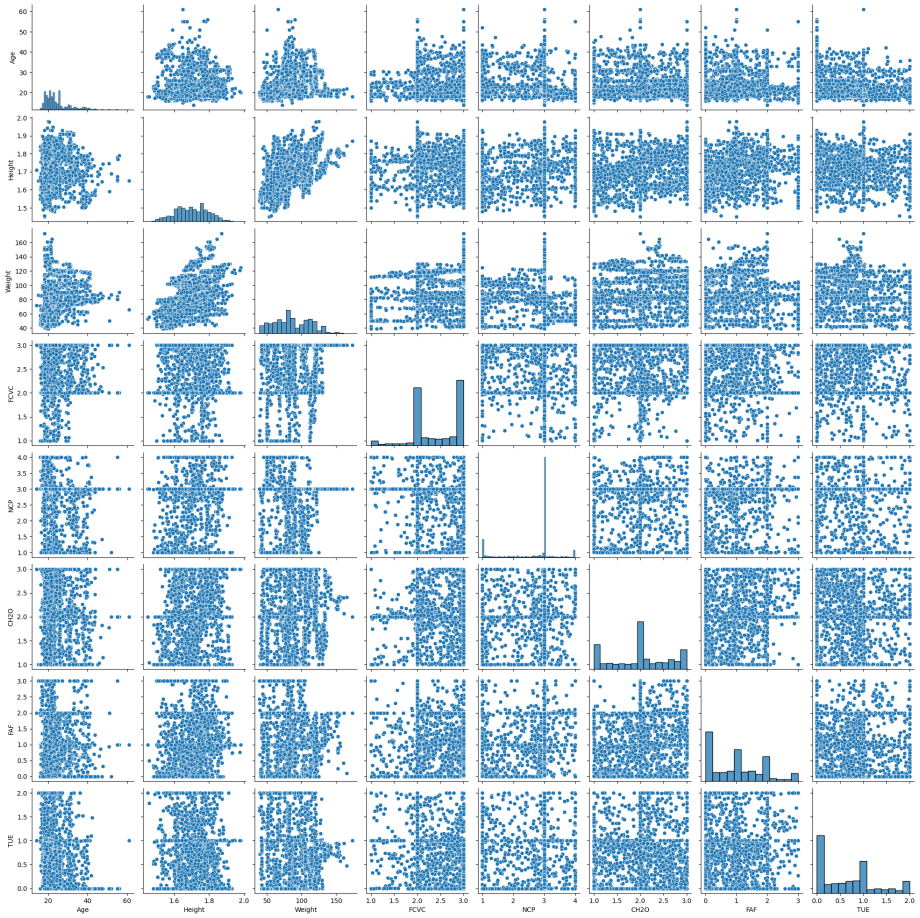
El estudio de Gualan Shao [9] también aplica distintos modelos de aprendizaje automático a este problema, y concluye que el XGBoost obtiene las mejores métricas, apoyando a otros de los estudios citados.

4 Desarrollo

Este dataset contiene varias clases de datos (categóricos, enteros, continuos y binarios). Además, los rangos en los que se encuentran estos datos son diferentes en varios casos.

Para un mejor entendimiento, disponemos de una representación gráfica, en la que se muestran mediante diagramas de dispersión la relación entre cada par de características. En la diagonal, se encuentra un histograma con la distribución de instancias de cada atributo.

Figure 4: Relaciones entre características



4.1 Tratamiento de datos

Los datos han tenido que sufrir un ligero preprocesado, consistente en la normalización de los mismos, para que puedan generar modelos válidos. El tipo de normalización elegido es el máximo mínimo. Nos hemos decantado por esta aproximación, ya que las variables numéricas toman rangos que pueden ser muy distantes en algunos casos, y este proceso asegura que los datos se encontrarán encapsulados en un rango entre 0 y 1.

Para las variables categóricas y las binarias, hemos aplicado una codificación One Hot. Esta permite codificar dichas variables para que sean aceptadas por todos los modelos.

Hablando de estas, cabe decir que las redes neuronales y otros modelos de machine learning son intrínsecamente aleatorios debido a factores como la inicialización de los pesos. Para minimizar el impacto de esta aleatoriedad en nuestros resultados, utilizamos la técnica de validación cruzada (Cross Validation). Esta técnica implica dividir el conjunto de datos completo en un número predeterminado de subconjuntos, o "folds". En nuestro caso, seleccionamos un valor de 5 folds. La validación cruzada se realiza mediante la ejecución de múltiples ciclos de entrenamiento y validación, donde cada fold se utiliza una vez como conjunto de validación, mientras que los restantes se utilizan para el entrenamiento. Por ejemplo, en la primera iteración, el primer fold es el conjunto de validación, y los folds 2 a 5 son el conjunto de entrenamiento. Este proceso se repite hasta que cada fold ha sido utilizado como conjunto de validación una vez. Esta metodología asegura que cada instancia del dataset es utilizada tanto en el entrenamiento como en la validación, lo que permite evaluar la robustez y generalización del modelo de manera más efectiva, reduciendo la variabilidad causada por la aleatoriedad en la selección de datos.

Esto implica que se ha separado el 20% de los datos para la validación, mientras que el 80% restante se emplea en el entrenamiento de los modelos. Es de vital importancia separar un porcentaje de datos de este modo para poder asegurar que se calculan métricas significativas.

4.2 Hiperparámetros

Para cada modelo, hemos realizado varias pruebas, ajustando los hiperparámetros en busca de una buena configuración. A continuación se indican los valores empleados:

- Para DT:
  - max_depth (profundidad máxima del árbol) = 6, 7, 5, 8, 9, 10, 12, 15

- Para kNN:
  - n_neighbors (número de vecinos utilizados para la clasificación) = 1, 2, 3, 4, 5, 6, 7, 15, 50
- Para SVM:
  - kernel (kernel del espacio) = "linear", "poly", "rbf", "sigmoid", "linear", "poly", "rbf", "sigmoid"
  - C (regularizar "trade-off" entre margen y minimizar errores) = 0.5, 1.0, 1.5, 1.25, 0.75, 1.75, 2, 2
  - degree (grado del polinomio, solo se aplica si el kernel es "poly") = 2, 3
  - gamma (modificar noción de proximidad entre instancias, solo se aplica si el kernel es "poly", "rbf" o "sigmoid") = "auto", "auto", "scale", "scale", "auto", "scale"
  - coef0 (afecta a la transformación de los datos entre kernel, solo se aplica si el kernel es "poly" o "sigmoid") = 0.0, 3, 1.0, 2
- Para RRNNAA:
  - topology (número de neuronas en las capas ocultas) = (10, 15), (4, 6), (15), (5, 13), (8, 9), (11), (5), (7, 14)
  - learningRate (grado de ajuste del gradiente) = 0.01
  - maxEpochs (máximo número de iteraciones sin mejora) = 1000
  - numExecutions (número de repeticiones del entrenamiento) = 7

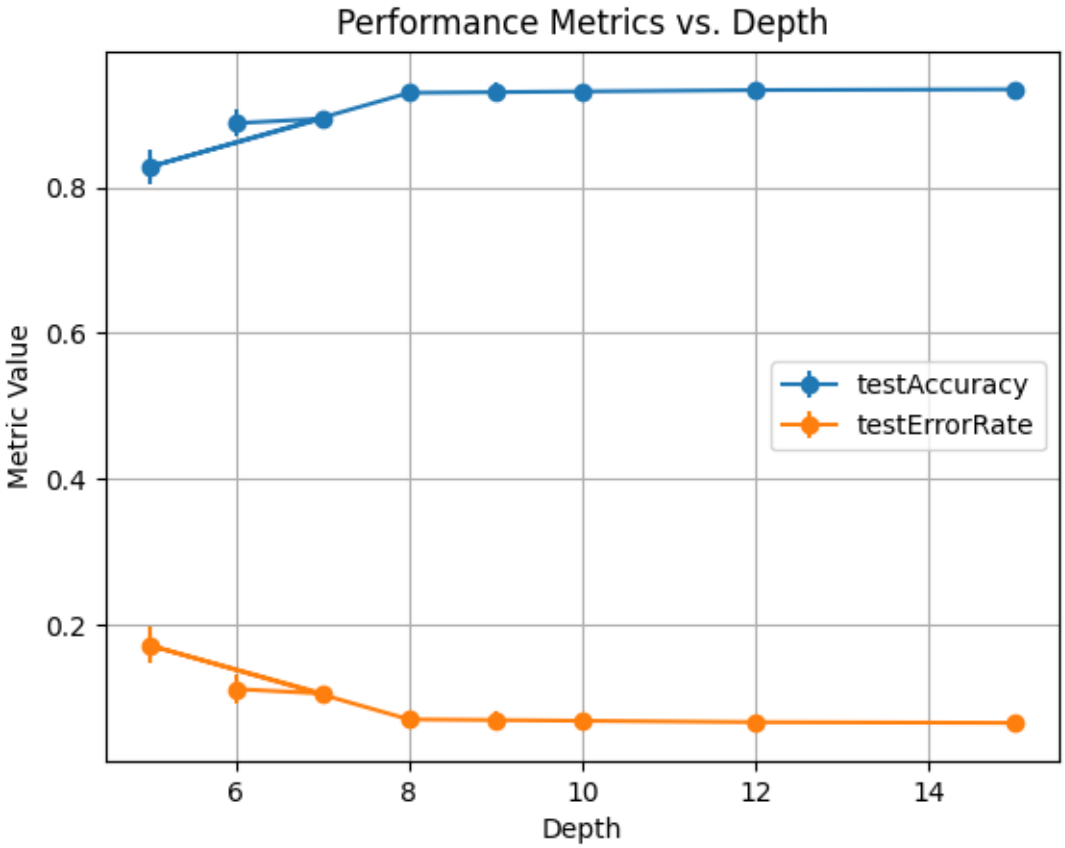
4.3 Resultados

Empleando estos valores para los hiperparámetros, y tras aplicar el preprocesado de datos mencionado, se entrenaron los modelos.

4.3.1 DT

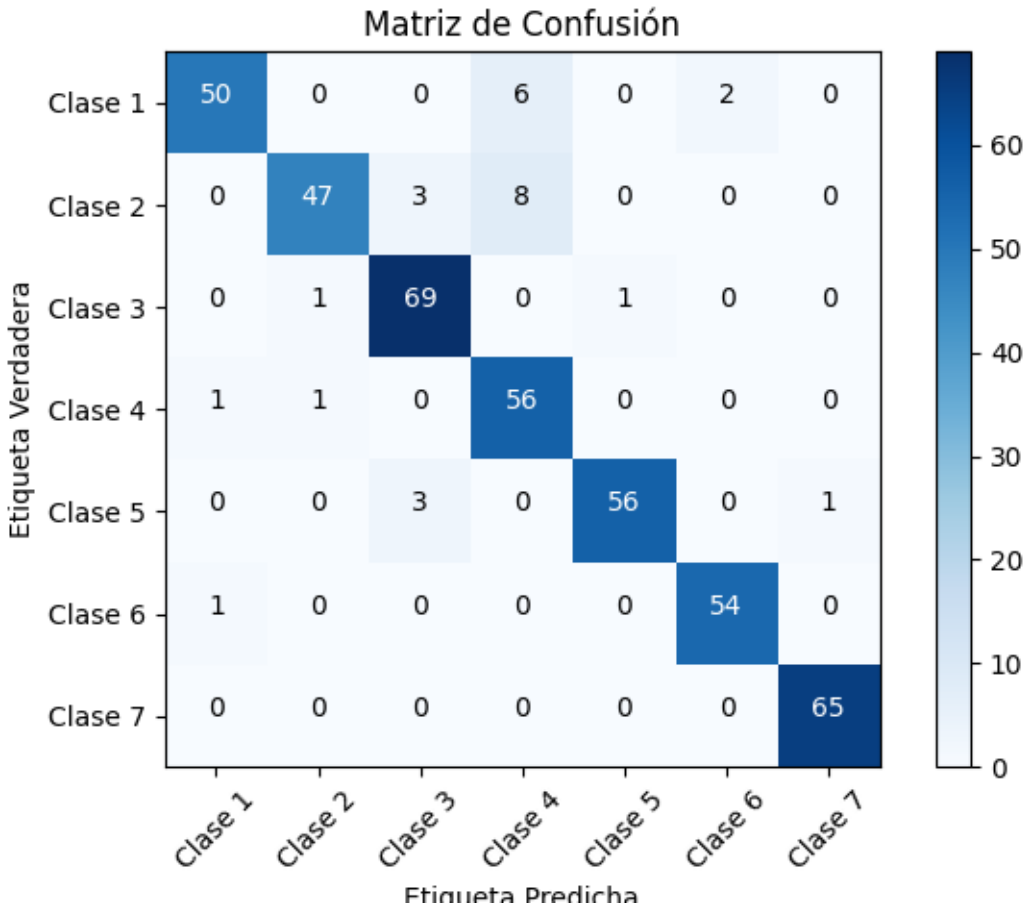
Para los DT, se probaron 8 configuraciones distintas del hiperparámtero max_depth, como ya indicado. Tras esto, se representaron los datos de manera gráfica para poder visualizarlos:

Figure 5: Evolución de precisión y tasa de error en DT



[4]

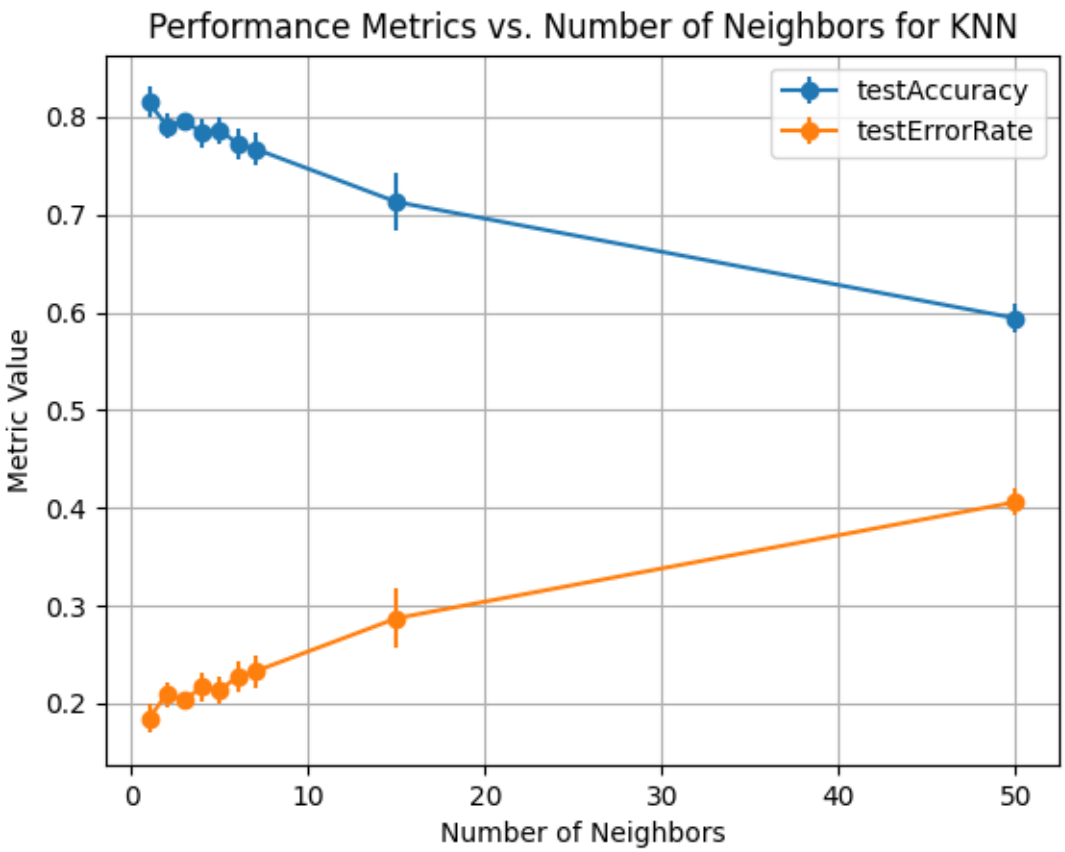
Es más que apreciable la tendencia de estas funciones. Como es claro en la gráfica, se alcanza el valor máximo de precisión al llegar a la profundidad 8, y a partir de ese punto, no mejora. Este valor es 92.98%.
Podemos ver las situaciones en las que el modelo tiene problemas si observamos su matriz de confusión:


Matriz de Confusión
Clase 1
Clase 2
Clase 3
Clase 4
Clase 5
Clase 6
Clase 7
Etiqueta Verdadera
50 0 0 6 0 2 0
0 47 3 8 0 0 0
0 1 69 0 1 0 0
1 1 0 56 0 0 0
0 0 3 0 56 0 1
1 0 0 0 0 54 0
0 0 0 0 0 0 65
Clase 1
Clase 2
Clase 3
Clase 4
Clase 5
Clase 6
Clase 7
Etiqueta Predicha
60
50
40
30
20
10
0

4.3.2 kNN

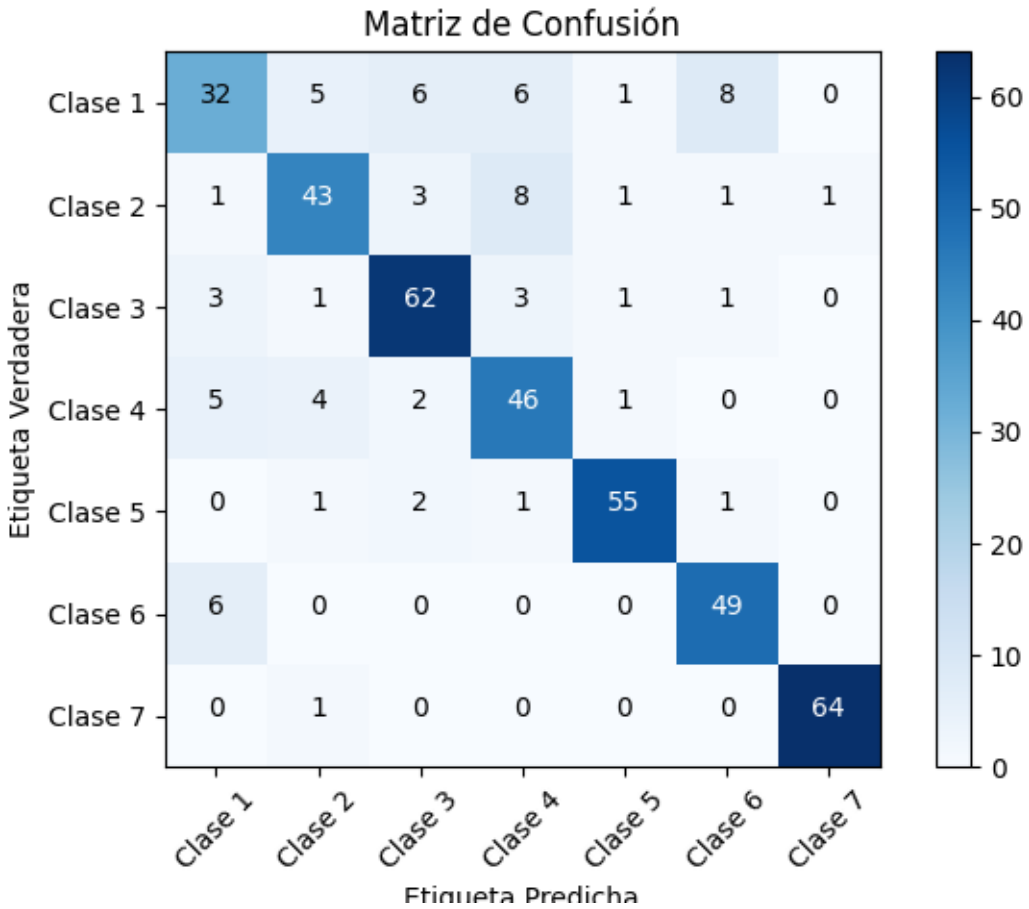
Se entrenaron modelos kNN con 9 configuraciones diferentes para el parámetro n_neighbors, los ya indicados. Tras esto, se representaron los datos de manera gráfica para poder visualizarlos:

Figure 6: Evolución de precisión y tasa de error en kNN



[5]

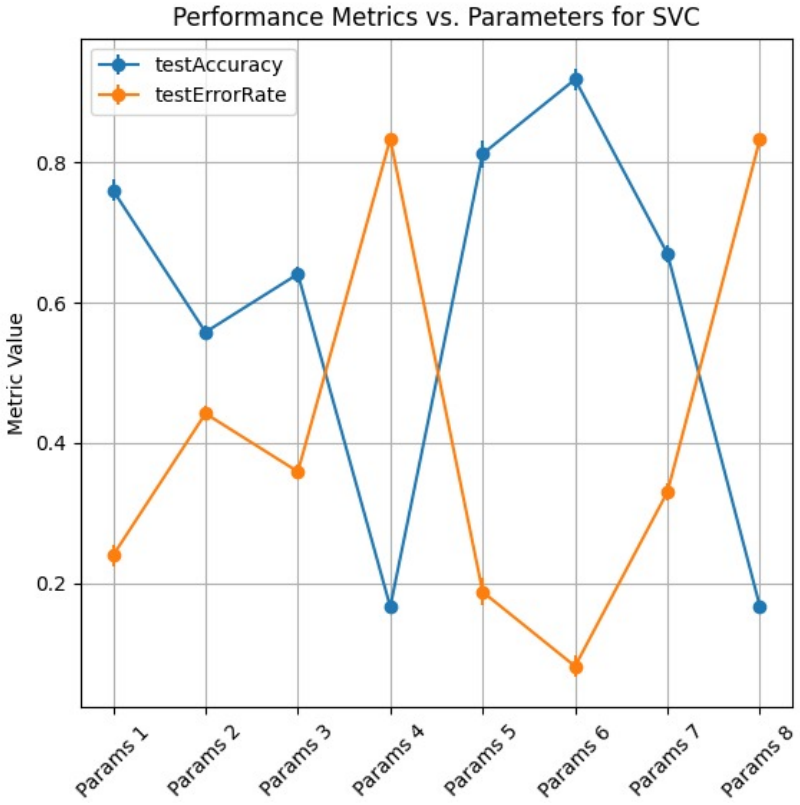
En este caso, los mejores resultados se obtienen con un número menor del hiperparámetro. A partir de este punto, tienden a empeorar constantemente. El mejor valor de precisión que se obtuvo (con n_neighbors = 1) fue 81.52%. Podemos ver las situaciones en las que el modelo tiene problemas si observamos su matriz de confusión:


Matriz de Confusión
Etiqueta Verdadera
Clase 1
Clase 2
Clase 3
Clase 4
Clase 5
Clase 6
Clase 7
32 5 6 6 1 8 0
1 43 3 8 1 1 1
3 1 62 3 1 1 0
5 4 2 46 1 0 0
0 1 2 1 55 1 0
6 0 0 0 0 49 0
0 1 0 0 0 0 64
Clase 1
Clase 2
Clase 3
Clase 4
Clase 5
Clase 6
Clase 7
Etiqueta Predicha
60
50
40
30
20
10
0

4.3.3 SVM

Con respecto a las SVM, se han probado 8 configuraciones, variando los valores de "kernel", "C", "degree", "gamma" y "coef0". Tras esto, se representaron los datos de manera gráfica para poder visualizarlos:

Figure 7: Evolución de precisión y tasa de error en SVM



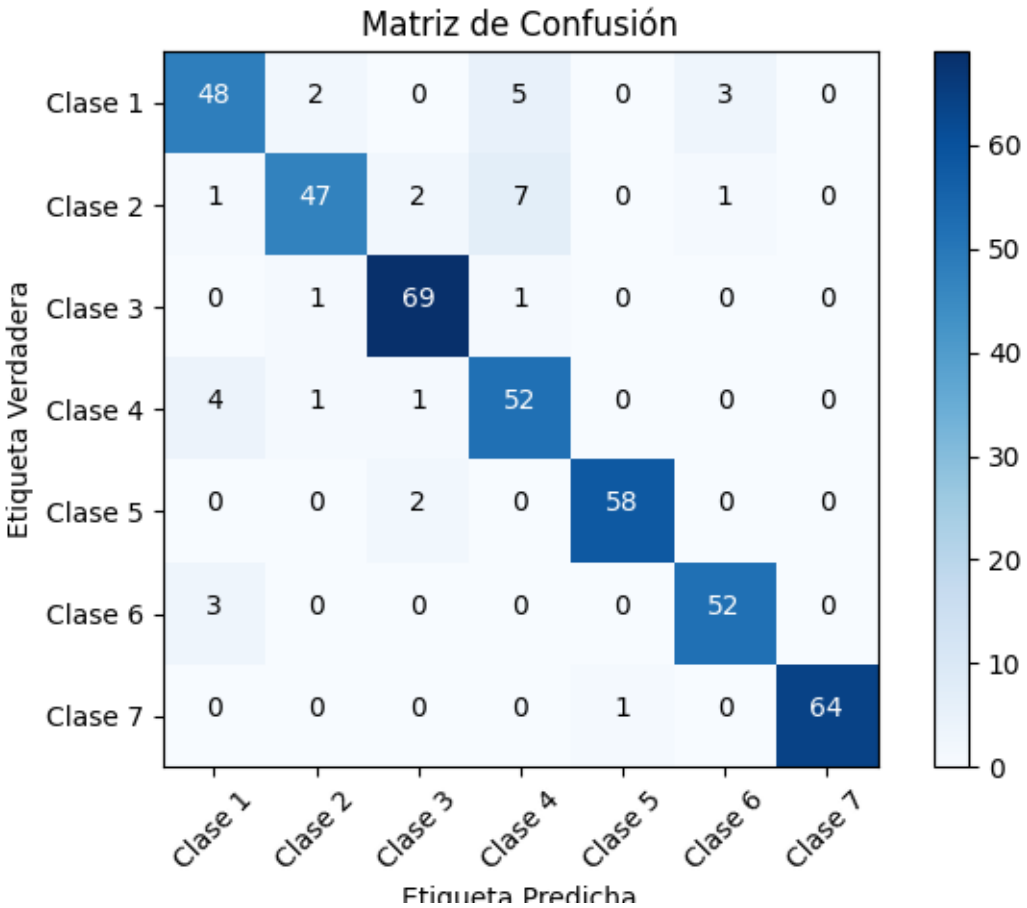
6

En este caso, los parámetros presentan mucha variación, al verse vistos a combinaciones de varias variables distintas. La configuración con mejor precisión usa los siguientes hiperparámetros:

- kernel = "poly"
- C = 1.75
- degree = 3
- gamma = "scale"

- coef0 = 1.0

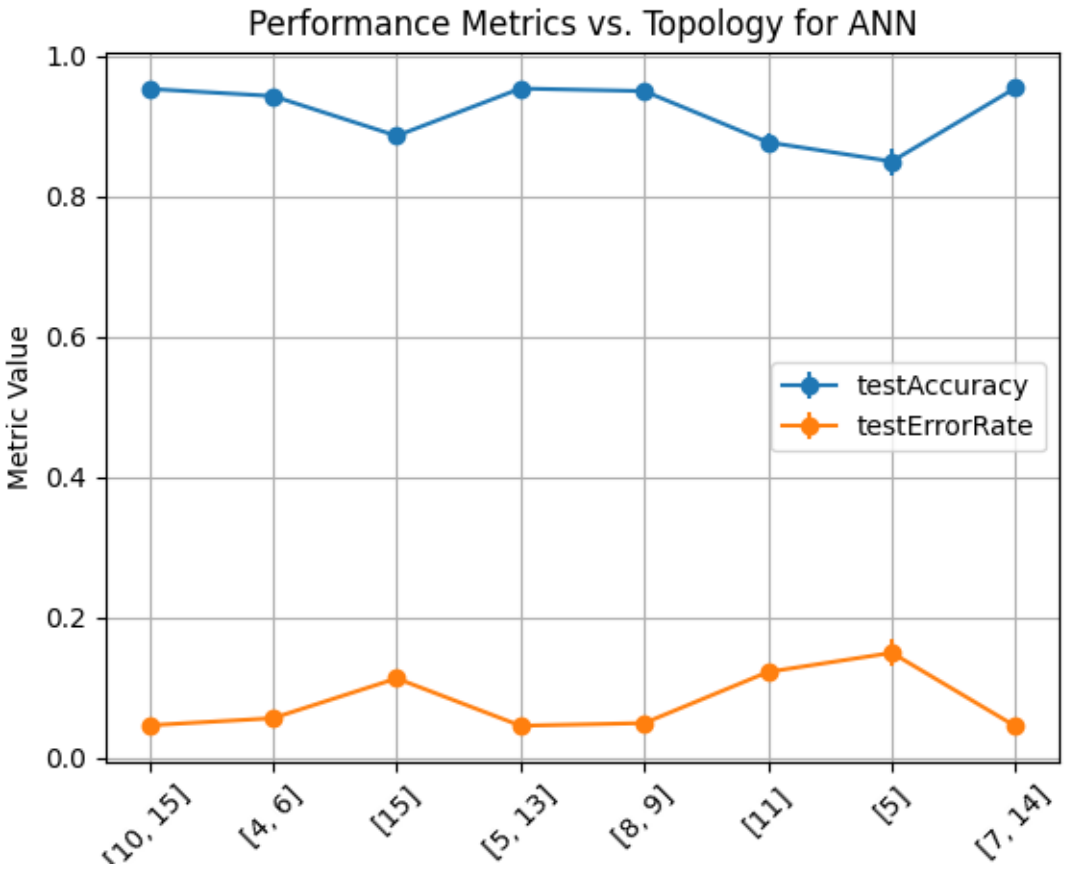
Con esta configuración, se alcanzó una precisión del 91.85%.
Podemos ver las situaciones en las que el modelo tiene problemas si observamos su matriz de confusión:



4.3.4 RRNNAA

En caso de las RRNNAA, se ha entrenado con 8 configuraciones distintas, correspondientes a los distintos tipos de topología. Tras esto, se representaron los datos de manera gráfica para poder visualizarlos:

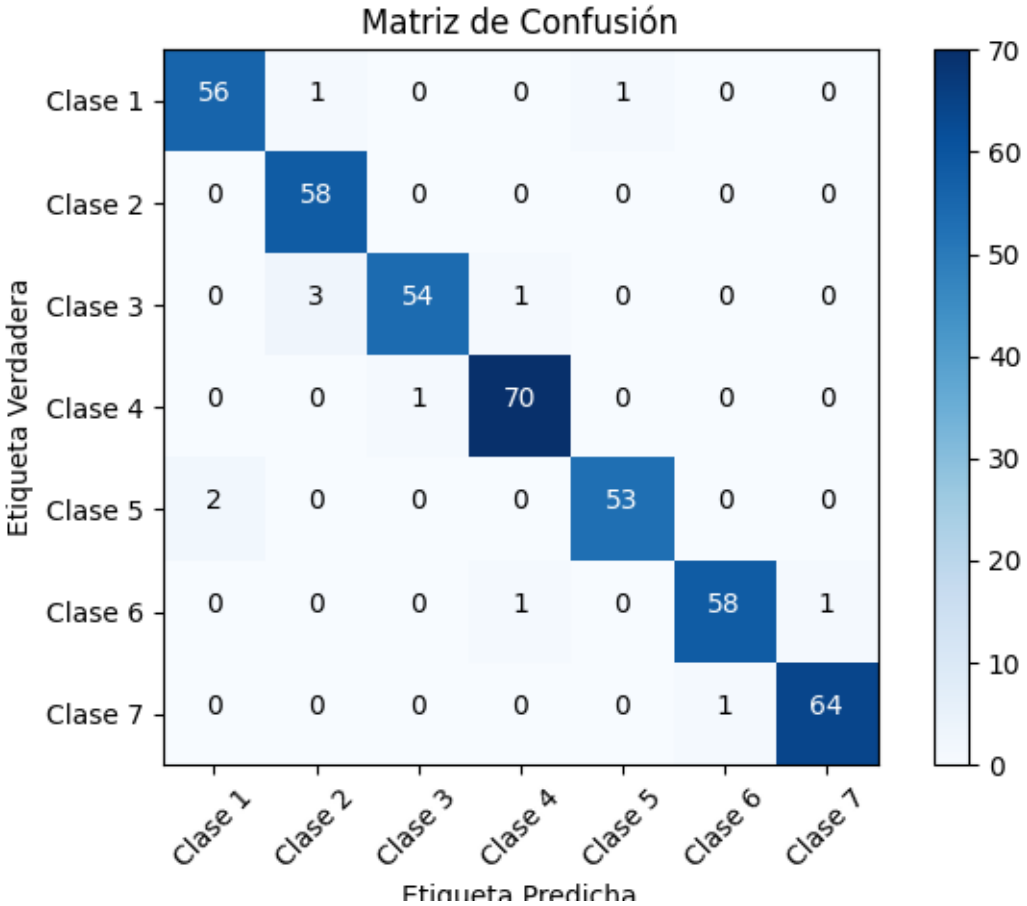
Figure 8: Evolución de precisión y tasa de error en RRNNAA



[7]

Se hace envidente la dominancia de los modelos con una topología basada en dos capas ocultas. Todos ellos obtienen una tasa de acierto que consideramos suficiente. En particular, la mayor tasa de acierto fue obtenida por la topología con siete y catorce (7, 14) neuronas, que logró una precisión del 95.47%. La mejor topología con una capa oculta fue con quince (15) neuronas, con una precisión del 88.63%.
Podemos ver las situaciones en las que el modelo tiene problemas si observamos

su matriz de confusión (el de mejor precisión, dos capas ocultas):



4.4 Discusión

En estos experimentos, se aprecian claramente los límites de algunos de los modelos. Los DT alcanzan un límite al alcanzar la profundidad 8, debido a que se dejan de hallar separaciones significativas en las características. Para kNN, es evidente que aumentar el número de vecinos tan solo introduce ruido en la clasificación, para estos datos en concreto.

Respecto a SVM, es probable que, si se continúan probando combinaciones, se logren obtener mejores métricas, ya que 8 combinaciones no cubren un gran porcentaje del total.

Por último, las RRNNAA podrían obtener mejores métricas si se entrenasen

con distintas inicializaciones aleatorias, ya que estos modelos dependen fuertemente de ese factor aleatorio.

Hemos determinado que los valores de precisión son aceptables en la mayoría de modelos (ya que kNN se queda corto con respecto a los demás). Una precisión ligeramente superior al 90% indica que el modelo no sufrió de sobreentrenamiento, y además obtuvo una cantidad de aciertos elevada. No es necesario que alcance mayor precisión, ya que en el contexto de uso de estos modelos, el dato obtenido se utilizaría como aproximación y debería ser contrastado por pruebas médicas adecuadas posteriormente.

Aunque los modelos de kNN estén claramente por detrás de los demás en estos casos, no es motivo para descartar esta aproximación. Otros conjuntos de datos pueden ajustarse mejor al dominio preferido por los modelos kNN, y pueden obtener resultados mejores que otros.

Con respecto a las matrices de confusión, todas siguen un mismo patrón: la mayor parte de los errores se concentran en el triángulo superior izquierdo de la matriz. La explicación para esto probablemente esté relacionada con similitud entre las instancias de estas clases; el desbalanceo de instancias es muy poco probable tras todo el proceso de equilibrio aplicado dataset. Este fenómeno es más apreciable en el caso de kNN, al ser el modelo que más errores comete. De todos modos, esto nos muestra que los sistemas tienen una precisión mucho mayor en la zona de interés, es decir, en la que se concentran los casos de obesidad. Esto hace que las tasas de error previas aumenten ligeramente si nos limitamos a los casos objetivo de los modelos.

5 Conclusiones

A la luz de estos hechos, el modelo más útil sería, probablemente, el DT. Esto se debe a que, ante mejoras insignificantes en la precisión, es preferible el esquema más sencillo, y en este caso (entre DT, SVM y RNA) ese es el DT.

Como ya mencionado, es importante que estos sistemas sean empleados por profesionales y solo como aproximación inicial, y que por tanto sean respaldados por pruebas médicas exhaustivas posteriormente para determinar la situación particular de cada paciente y sus necesidades específicas.

En calidad de estudiantes, cabe mencionar que este proyecto supuso una de las primeras tomas de contacto con lo que es una aplicación real del trabajo práctico realizado. Gracias a este trabajo, hemos obtenido cierta experiencia para realizar experimentación en el campo del apredizaje automático, así como para redactar informes completos y documentar un proyecto de investigación.

6 Trabajo futuro

Los sistemas obtenidos en esta investigación cumplen el propósito original del proyecto. Sin embargo, no están restringidos a este ámbito. Resaltando que los modelos no se entrenan con un campo concreto, sino codificando datos, es posible aplicar estos modelos de aprendizaje automático a cualquier ámbito en el que se logre codificar el problema.

Con respecto a estos datos, como se deduce de muchos de los trabajos citados, los modelos de XGBoost obtienen mejores resultados que los aquí investigados, por lo que probablemente sea conveniente considerar utilizar esa aproximación, dependiendo de las necesidades particulares del entorno.

Evidentemente, siempre existe espacio para mejora y experimentación, por lo que los resultados de esta investigación siempre pueden estar sujetos a variabilidad para cada situación particular.

7 Bibliografía

References

[1] Harika Gozde Gozukara Bag et al. "Estimation of Obesity Levels through the Proposed Predictive Approach Based on Physical Activity and Nutritional Habits". In: *Diagnostics* 13 (2023). URL: https://api.semanticscholar.org/CorpusID:261989719.

[2] Duwi Cahya Putri Buani and Nia Nuraeni. "Application of XGB Classifier for Obesity Rate Prediction". In: *Jurnal Riset Informatika* (2023). URL: https://api.semanticscholar.org/CorpusID:266484135.

[3] Akash Choudhuri. "A Hybrid Machine Learning Model for Estimation of Obesity Levels". In: *medRxiv* (2022). DOI: 10.1101/2022.08.17.22278905. eprint: https://www.medrxiv.org/content/early/2022/08/18/2022.08.17.22278905.full.pdf. URL: https://www.medrxiv.org/content/early/2022/08/18/2022.08.17.22278905.

[4] I Gede Susrama Mas Diayasa et al. "Stacking Ensemble Methods to Predict Obesity Levels in Adults". In: *2022 IEEE 8th Information Technology International Seminar (ITIS)*. 2022, pp. 339–344. DOI: 10.1109/ITIS57155.2022.10010260.

[5] Mahmut Dirik. "Application of machine learning techniques for obesity prediction: a comparative study". In: *Journal of Complexity in Health Sciences* (2023). URL: https://api.semanticscholar.org/CorpusID:265757593.

[6] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2019. URL: http://archive.ics.uci.edu/ml.

[7] Mehmet Kıvrak. "DEEP LEARNING-BASED PREDICTION OF OBESITY LEVELS ACCORDING TO EATING HABITS AND PHYSICAL CONDITION". In: *The Journal of Cognitive Systems* 6.1 (2021), pp. 24–27. DOI: 10.52876/jcs.939875.

[8] Fabio Mendoza Palechor and Alexis De la Hoz Manotas. *Estimation of Obesity Levels Based On Eating Habits and Physical Condition*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5H31Z. 2019.

[9] Guanlan Shao. "Comparison of prediction of obesity status based on different machine learning approaches with different factor quantities". In: *International Conference on Biomedical and Intelligent Systems (IC-BIS 2022)*. Ed. by Ahmed El-Hashash. Vol. 12458. International Society for Optics and Photonics. SPIE, 2022, 124583U. DOI: 10.1117/12.2660726. URL: https://doi.org/10.1117/12.2660726.

[10] Ian H. Witten et al. *Weka – A Machine Learning Workbench for Data Mining*. https://www.cs.waikato.ac.nz/ml/weka/. En el caso mencionado, se utilizó una versión anterior de la herramienta. 2021.

[11] Fatma Hilal Yagin et al. "Estimation of Obesity Levels with a Trained Neural Network Approach optimized by the Bayesian Technique". In: *Applied Sciences* 13.6 (2023). ISSN: 2076-3417. DOI: 10.3390/app13063875. URL: https://www.mdpi.com/2076-3417/13/6/3875.