22302823_tfm_VascoOliveiraCarapinheiro.pdf

by Vasco OLIVEIRA CARAPINHEIRO

Submission date: 05-Oct-2024 06:57PM (UTC+0200)
Submission ID: 2475889412
File name: 22302823_tfm_VascoOliveiraCarapinheiro.pdf (2.84M)
Word count: 12505
Character count: 68155

Máster en Bioinformática

Predicción de Diabetes tipo 2 mediante algoritmos de aprendizaje supervisado

Autor: Vasco Oliveira Carapinheiro

Tutor: Ana Monsalve Torra

Curso 2023-24

Fecha: 05 / 10 / 2024

AGRADECIMIENTOS

Me gustaría agradecer a la Universidad Europea y más concretamente a mi tutora, Ana Monsalve, por su ayuda durante la realización de este proyecto. También a la directora de la titulación, Rocío González, por su ayuda y disponibilidad durante el curso. Agradecer a Irene, Juan, Jaime y Román que han sabido levantarme cuando lo he necesitado y que me han demostrado que los buenos amigos están ahí en los buenos y en los malos momentos. Agradecer a mi madre, que ha sido mi gran apoyo durante este año y sin quién no podría haber llegado hasta aquí.

Me gustaría agradecer especialmente a mi padre, que me contagió su interés por la informática y la programación. Su amor y apoyo incondicional me han hecho crecer como persona y me han preparado para el resto de mi vida y por ello estaré eternamente agradecido. Pudo ver como empezaba este proyecto y seguro que estaría orgulloso del resultado.

RESUMEN

La diabetes tipo 2 en adultos supone un gran problema sanitario a nivel mundial y las complicaciones asociadas a la enfermedad (amputaciones, ceguera, accidentes cerebrovasculares, etc.) podrían ser mitigadas mediante el diagnóstico precoz. La Inteligencia Artificial (IA) ha supuesto un gran avance en el sector sanitario y se presenta como una herramienta contrastada para el diagnóstico precoz de la diabetes tipo 2. En este proyecto de investigación se implementan una serie de modelos de IA (*Random Forest, Support Vector Machine, Naive Bayes* y *Multilayer Perceptron*) para el diagnóstico de la enfermedad y se comparan sus resultados. Tras el preprocesamiento de los datos se entrenan los diferentes modelos en busca de la combinación óptima de hiperparámetros y posteriormente se evalúa la eficacia de los modelos con los hiperparámetros óptimos. Los modelos *Random Forest* y *Multilayer Perceptron* se destacan como los modelos con mejor desempeño ambos con un F1 Score superior a 0.73 mientras que el *Naive Bayes* es el modelo que obtiene los peores resultados. Los resultados obtenidos indican que la IA podría servir como solución al problema diagnóstico de la diabetes tipo 2 pero que todavía es necesario trabajo para mejorar.

Palabras clave: Inteligencia Artificial, *Random Forest*, *Support Vector Machine*, *Naive Bayes*, *Multilayer Perceptron*, Diabetes tipo 2, Predicción.

ABSTRACT

Type 2 diabetes in adults is a major health problem worldwide and the complications associated with the disease (amputations, blindness, strokes, etc.) could be mitigated by early diagnosis of the disease. Artificial Intelligence (AI) has been a breakthrough in the healthcare sector and is presented as a proven tool for the early diagnosis of type 2 diabetes. In this research project, a series of AI models (Random Forest, Support Vector Machine, Naive Bayes and Multilayer Perceptron) are implemented for the diagnosis of the disease and their results are compared. After data preprocessing, the different models are trained in search of the optimal combination of hyperparameters and then the efficiency of the models with the optimal hyperparameters is evaluated. The Random Forest and Multilayer Perceptron models stand out as the best performing models both with an F1 Score above 0.73 while Naive Bayes is the worst performing model. The results obtained indicate that AI could serve as a solution to the diagnostic problem of type 2 diabetes, but work is still needed to improve.

Keywords: Artificial Intelligence, Random Forest, Support Vector Machine, Naive Bayes, Multilayer Perceptron, Type 2 Diabetes, Prediction.

INDICE

1. INTRODUCCIÓN

La diabetes supone un gran desafío para la sanidad pública mundial por su gran incidencia y su elevada mortalidad, siendo la novena causa de muerte a nivel mundial (American Diabetes Association, 2013). En el año 2021, un total de 38.1 millones de adultos en los Estados Unidos sufrían diabetes lo que supone un 14.7% de los adultos en ese país y 8.7 millones de ellos estaban sin diagnosticar (CDC, 2024a). Esta enfermedad puede causar graves complicaciones como ceguera, infartos de miocardio, accidentes cerebrovasculares y amputaciones (World Health Organization, 2023). Esto supone un gran problema sanitario a nivel mundial y hace imperativo el desarrollo de nuevas formas de diagnóstico precoz de la enfermedad.

La Inteligencia Artificial (IA) surge como un gran candidato para llevar a cabo esta tarea debido a su crecimiento exponencial en los últimos años y su excelente desempeño en otras áreas dentro del sector sanitario. Por este motivo, este proyecto de investigación profundizará en la IA y en su uso para la predicción precoz de la diabetes tipo 2 en adultos con el objetivo de prevenir las complicaciones asociadas a la enfermedad.

1.1. Diabetes

La diabetes comprende un grupo de enfermedades metabólicas caracterizadas por altos niveles de glucosa en sangre provocadas por una deficiencia en la actuación de la insulina o en su secreción. Su incidencia ha aumentado considerablemente pasando de 108 millones de personas con diabetes en 1980 a 422 millones en el 2014 (World Health Organization, 2023). Esta enfermedad es la novena causa de mortalidad y provoca más de un millón de muertes anuales a nivel mundial (American Diabetes Association, 2013; Nurdin et al., 2023). En España, esta enfermedad fue la causante de 23 muertes por cada 100000 habitantes en el año 2022 (Instituto Nacional de Estadística, s. f.). La diabetes puede dividirse en dos categorías, la diabetes tipo 1, en la que existe una deficiencia total en la secreción de insulina y la diabetes tipo 2, en la que existe una resistencia a los efectos de la insulina y por lo tanto una secreción inadecuada de insulina como respuesta. La diabetes tipo 2 supone más del 90% de los casos totales de la enfermedad (American Diabetes Association, 2013) y

suele tardar muchos años en ser diagnosticada ya que la hiperglucemia se va desarrollando de manera gradual y no es hasta pasado un tiempo que empieza el paciente a darse cuenta de sus síntomas (American Diabetes Association, 2013). Los pacientes con este tipo de diabetes suelen ser obesos y aquellos que no lo son suelen contar con un mayor porcentaje de grasa abdominal (American Diabetes Association, 2013). La hiperglucemia asociada a la enfermedad puede provocar daños severos en diferentes órganos entre los que se incluyen los riñones, corazón y nervios. Otras complicaciones conocidas de la diabetes son las úlceras en los pies que pueden llevar a la amputación, pérdida de visión y aumento de la frecuencia de accidentes cerebrovasculares (American Diabetes Association, 2013).

La prevención de las complicaciones asociadas a la enfermedad se destaca como un tema muy importante a nivel mundial y necesita de nuevas formas de diagnóstico, y más accesibles, que permitan la detección precoz. El uso de la IA en el diagnóstico de la diabetes surge como una gran oportunidad y se presenta como una posibilidad real y contrastada para solucionar este problema debido a su ya amplio uso en el sector sanitario.

1.2. Inteligencia Artificial en el sector sanitario

La Inteligencia Artificial es la rama más reconocida actualmente en el campo de las ciencias computacionales y se define como una serie de metodologías cuyo objetivo es emular la inteligencia humana (Secinaro et al., 2021). Estas metodologías ya están muy presentes en el sector sanitario y se están produciendo importantes avances en los últimos años, por ejemplo, ya existen algoritmos capaces de obtener mejores resultados que radiólogos en la detección de tumores malignos (Davenport & Kalakota, 2019).

Una de las grandes ventajas de la Inteligencia Artificial es que puede trabajar con la gran cantidad de datos que se generan en el sector de la salud y ayudar a encontrar información que no sería accesible de otra forma (Nurdin et al., 2023; Secinaro et al., 2021). La IA ya ha sido aplicada en numerosos casos en el sector sanitario y biomédico, el uso de redes neuronales convolucionales permite la interpretación de electrocardiogramas e incluso la identificación de patrones irreconocibles por las personas (Siontis et al., 2021), también ha

permitido el diagnóstico de neumonía causada por COVID-19 mediante tomografía computarizada de manera comparable a radiólogos experimentados (K. Zhang et al., 2020). La IA ya ha sido utilizada también para asistir en el diagnóstico de la diabetes, diferentes algoritmos como las redes neuronales artificiales, *Support Vector Machines* y los árboles de decisión ya han sido utilizados para el diagnóstico de la enfermedad (Nurdin et al., 2023). También han sido usados algoritmos como el perceptrón multicapa para el reconocimiento de patrones asociados a diabetes tipo 2 en lesiones de la sustancia blanca a través de imágenes de resonancia magnética (Luna et al., 2021).

La IA también es útil en el campo del desarrollo de fármacos ya que el gran volumen de datos y el gran coste que supone su análisis requiere de nuevas herramientas que permitan agilizar el proceso, el aprendizaje profundo y más concretamente las redes neuronales surgen como una buena opción en este ámbito (Zhu, 2020).

1.3. Pregunta de investigación e hipótesis

Conocidas las complicaciones asociadas a la diabetes tipo 2 y las múltiples posibilidades que ofrece la Inteligencia Artificial surge la siguiente pregunta de investigación:

¿Puede la Inteligencia Artificial ser utilizada para diagnosticar precozmente la diabetes tipo 2 en adultos y prevenir las graves complicaciones asociadas a la enfermedad?

Habiendo profundizado en casos de éxito del uso de Inteligencia Artificial para el diagnóstico de enfermedades, incluida la diabetes, la hipótesis de partida de este trabajo es que la Inteligencia Artificial será eficaz en el diagnóstico de la diabetes tipo 2 en adultos y que puede suponer un gran avance en la prevención de las complicaciones asociadas a la enfermedad.

1.4. Objetivo general

Para demostrar la hipótesis, el objetivo principal de este trabajo de investigación es implementar una herramienta que permita predecir la diabetes tipo 2 en población adulta mediante algoritmos de Inteligencia Artificial para reducir las complicaciones asociadas a esta enfermedad.

1.5. Objetivos específicos

Para alcanzar dicho objetivo principal, se plantean una serie de objetivos secundarios:

- Realizar un análisis exploratorio de los datos para identificar las variables relevantes para el entrenamiento del modelo y eliminar aquellas que puedan causar confusión.
- Determinar cuál de los algoritmos implementados presenta mejor desempeño en la predicción de la diabetes tipo 2 en adultos.
- Determinar cuál de los métodos de evaluación de los algoritmos es el más adecuado para determinar el desempeño de estos.

1.6. Metodología y *pipeline*

El *pipeline* a seguir en este proyecto de investigación se presenta en la Figura 1. A partir de la base de datos se realizará una búsqueda de datos duplicados, posteriormente se realizará un análisis exploratorio de los datos y se estudiará la correlación entre variables. Se balanceará la base de datos de acuerdo con la variable principal y se realizará la partición de los datos para el entrenamiento y evaluación seguido de la normalización de estos. Posteriormente, se entrenarán los distintos modelos utilizando una validación cruzada con el objetivo de encontrar los hiperparámetros óptimos. Los modelos optimizados se evaluarán con la partición de test y se comparará el desempeño de los modelos y se procederá a la elección del modelo con mejores resultados. Los lenguajes de programación utilizados en este proyecto son Python en su versión 3.12.5 y R en su versión 4.4.0. No se ha hecho uso de herramientas de IA para la realización de este proyecto y se adjunta la declaración responsable del uso de IA como Anexo 1.

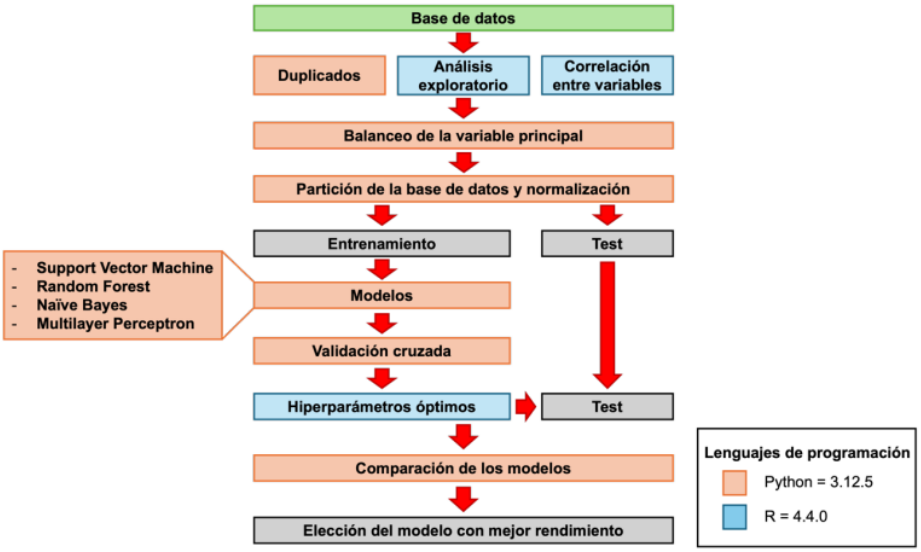

Figura 1. *Pipeline* seguido en la realización del proyecto. Los pasos en naranja se corresponden con los realizados en Python mientras que los pasos en azul fueron realizados con R. Fuente: elaboración propia.

2. ESTADO DEL ARTE

Actualmente, la Inteligencia Artificial ya ha sido utilizada para el diagnóstico de la diabetes tipo 2 siguiendo diferentes enfoques. En el trabajo realizado por (Nurdin et al., 2023) se utilizó una base de datos con énfasis en el contenido sanguíneo (niveles de colesterol, triglicéridos, lipoproteínas, etc.) para entrenar tres algoritmos (*Support Vector Machine (SVM)*, *Random Forest (RF)* y *Multilayer perceptron (MLP)*) y comparar sus resultados. Los resultados alcanzados en este estudio fueron buenos llegando a obtener valores de exactitud superiores a 0.98 pero, tal y como indican los autores, estos valores podrían estar sesgados debido a la base de datos, que está formada mayoritariamente por casos de diabetes y no se llevó a cabo ningún tipo de balanceo.

En el estudio llevado a cabo por (Luna et al., 2021) tenían como objetivo utilizar imágenes de resonancia magnética para detectar la diabetes tipo 2 mediante lesiones en la sustancia blanca del cerebro. Los investigadores implementaron un *pipeline* para detectar y extraer las características de interés en las imágenes y usarlas para el entrenamiento de modelos (SVM, RF, MLP y regresión logística). La regresión logística resultó ser el mejor modelo en este estudio alcanzando un 88% de exactitud y un 100% de sensibilidad. Una vez más, el estudio se realizó sobre una base de datos mayoritariamente diabética y los resultados podrían estar sesgados siendo necesario la evaluación con un mayor número de casos no diabéticos.

El proyecto de investigación llevado a cabo por (khan et al., 2023) tenía como objetivo desarrollar un sistema de diagnóstico no invasivo para la diabetes tipo 2 mediante las señales obtenidas en un fotoplestimograma, una técnica óptica que permite la visualización de cambios en el volumen sanguíneo y en el sistema vascular periférico (khan et al., 2023). El preprocesamiento de los datos incluyó la transformación de las ondas mediante técnicas computacionales con el objetivo de eliminar el ruido y poder analizar segmentos más pequeños de estas. Posteriormente, realizaron la extracción de características y seleccionaron las más relevantes mediante el cálculo de la correlación con otras variables de su base de datos como los valores de la presión arterial sistólica y diastólica. Utilizaron diferentes modelos, como SVM, MLP, *K-Nearest Neighbors* (KNN) y otro modelo conocido como *Adaptive-Network-based Fuzzy Inference Systems*

*(*ANFIS). El estudio consiguió buenos resultados, obteniendo el SVM un F1 Score del 99.1% pero, según comentan los autores, obteniendo mejores resultados clasificando casos extremos de hipertensión que en los casos con hipertensión normal y pre-hipertensión. La principal limitación del estudio es la cantidad de datos limitados y la falta de la cuantificación de la glucosa en sangre (khan et al., 2023).

En el estudio realizado por (Mehedi Hassan et al., 2022) se utilizó una base de datos llamada "Early Stage Diabetes Risk Prediction dataset" que contiene 17 variables obtenidas para 520 adultos. La base de datos contiene variables relacionadas con la diabetes como poliuria, polidipsia, obesidad y fatiga muscular. Este estudio tenía como objetivo la combinación de aprendizaje supervisado y no supervisado. Para ellos, primero utilizaron un algoritmo de *clustering* denominado *K-Means*. Este algoritmo es un método de aprendizaje no supervisado que divide la base de datos en un número determinado de grupos (o *clusters*) (Mehedi Hassan et al., 2022). Mediante este algoritmo obtuvieron tres *clusters* a los que se les asignó una nueva etiqueta dependiendo de su grupo. De esta forma se generó una nueva base de datos que se utilizó para entrenar una serie de modelos (RF, SVM, MLP, KNN y árboles de decisión). El RF obtuvo los mejores resultados con una exactitud el 99.03% y un F1 Score de 99.20% seguidos del MLP y del SVM ambos con una exactitud y F1 Score de 98.07% y 98.40% respectivamente (Mehedi Hassan et al., 2022).

3. MATERIALES Y MÉTODOS

3.1. Base de datos

Los datos utilizados en este trabajo provienen de *Centers for Disease Control and Prevention* (CDC), concretamente de su estudio *Behavioral Risk Factor Surveillance System* (BRFSS) (CDC, 2024b). Este estudio se realiza anualmente en los Estados Unidos mediante entrevistas telefónicas y su objetivo es recoger información sobre hábitos saludables, prevención y conocimiento de enfermedades y factores de riesgo, entre otros. El estudio fue instaurado en 1984 y desde el 2001 participan en él todos los Estados y son entrevistados más de 350000 adultos de todo el país (United States Department of Health and Human Services, s. f.).

En este trabajo se utiliza un subconjunto de los datos obtenidos en este estudio en el año 2015. Este subconjunto de datos está publicado en la página web *Kaggle*[1], una de las más conocidas comunidades de Inteligencia Artificial y *Machine Learning*. La URL para acceder a los datos se presenta en el Anexo 2. La selección del subconjunto de variables está inspirada en el trabajo realizado por (Xie et al., 2019) y consta de 22 variables recogidas para 253.680 personas (Teboul, 2021).

Las variables y su descripción se pueden ver en la Tabla 1, cuentan con variables demográficas como el género, la edad y la educación además de variables relacionadas con la salud como el nivel de colesterol y la presión arterial. También hay variables relacionadas con hábitos saludables como el consumo de fruta y verdura, tabaquismo y ejercicio físico. La base de datos consta de 253600 registros para las 22 variables y no tienen ningún valor faltante o nulo. Los datos para cada variable ya están codificados de forma numérica, por ejemplo, los datos para la variable “Diabetes_binary” están codificados de la siguiente forma:

- 0 = el participante no tiene diabetes.
- 1 = el participante tiene diabetes o prediabetes

La codificación de todas las variables se presenta como Anexo 3.

[1] https://www.kaggle.com

Variable	Descripción
Diabetes_binary	Presenta diabetes/prediabetes o ninguna
HighBP	Tiene la presión arterial elevada o normal
HighChol	Tiene el colesterol elevado o normal
CholCheck	Se ha revisado los niveles de colesterol en los últimos 5 años
BMI	Índice de masa corporal
Smoker	Se considera fumador si ha fumado por lo menos 100 cigarros en su vida
Stroke	Ha sufrido un accidente cerebrovascular
HeartDiseaseorAttack	Ha sufrido o sufre enfermedad coronaria o infarto de miocardio
PhysActivity	Ha realizado ejercicio físico en los últimos 30 días
Fruits	Consume fruta una o más veces por día
Veggies	Consume vegetales una o más veces por día
HvyAlcoholConsump	Consume más de 14 bebidas alcohólicas a la semana si es hombre, más 7 si es mujer
AnyHealthcare	Tiene algún tipo de seguro de salud
NoDocbcCost	En algún momento del pasado año no ha ido al médico debido a su alto coste
GenHlth	Como considera su salud general del 1 al 5
MentHlth	Cuantos de los últimos 30 días considera que su salud mental no ha sido buena
PhysHlth	Cuantos de los últimos 30 días considera que su salud física no ha sido buena
DiffWalk	Tiene dificultad para andar o subir escaleras
Sex	Género
Age	Edad
Education	Nivel educativo
Income	Ingresos

Tabla 1. Descripción de las variables incluidas en la base de datos. Fuente: elaboración propia.

3.2. Análisis exploratorio de la base de datos

El primer paso será realizar un análisis exploratorio de la base de datos para comprender bien su estructura. Se va a analizar la existencia de datos duplicados mediante un *script* de Python utilizando la función duplicated() de la librería pandas. Esta función se aplica sobre un *dataframe* y devuelve como resultado una lista de operadores booleanos (*True* o *False*) dependiendo de si ese registro está duplicado o no. La función duplicated() en conjunto con la función sum() (incluida por defecto en Python) nos dará el número total de duplicados. Una vez identificado el número de valores duplicados se procederá a eliminarlos mediante la función drop_duplicates() de pandas, que se utiliza sobre un *dataframe* y devuelve otro sin los registros duplicados.

Una vez obtenida la base de datos sin duplicados se procederá al análisis de las diferentes variables para entender mejor la estructura de la muestra sobre la que se va a trabajar. Este análisis será realizado mayoritariamente mediante *scripts* en lenguaje R y librerías como ggplot2 para la realización de gráficos y dplyr para trabajar con *dataframes*. Se pretende también con este análisis exploratorio identificar las variables relevantes para el posterior entrenamiento del modelo de forma que no se utilicen aquellas variables que no tengan correlación con la variable objetivo ("Diabetes_binary"). Para ello se utilizará la función cor(), incluida en la librería stats de R, que calcula el coeficiente de correlación entre las variables y la función corrplot(), de la librería corrplot, para graficar los resultados. En caso de encontrar variables con poca correlación con la variable objetivo se eliminarán de la base de datos mediante la función drop() de pandas que excluye de un *dataframe* las columnas (variables) especificadas como argumento.

3.3. Balanceo y creación de las particiones de datos

Tras el análisis exploratorio se procederá a realizar el balanceo de la variable objetivo ("Diabetes_binary"). Para ello se conserva la clase con menos casos en su totalidad, de la clase mayoritaria se eliminarán casos hasta igualar su número a la minoritaria. El balanceo se realizará mediante *scripts* en Python utilizando la función NearMiss() de la librería imblearn. Esta función elimina los registros de la clase mayoritaria más cercanos a la clase minoritaria. Se puede

ver una representación del funcionamiento en la Figura 2. La función calcula las distancias entre las diferentes clases y elimina aquellos puntos más cercanos hasta balancear las clases. La función toma como argumento "n_neighbors" para determinar el número de puntos de la clase minoritaria sobre el que calcular las distancias. En el caso representado en la Figura 2 el argumento sería "n_neighbors" = 3.

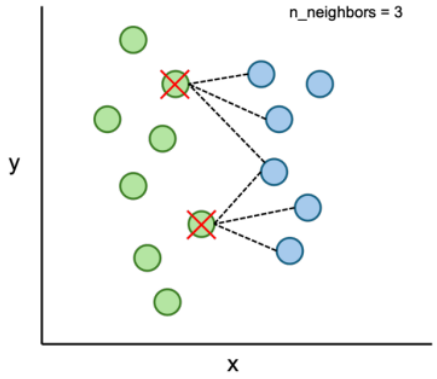

Figura 2. Representación del funcionamiento de la función NearMiss(). Se eliminan datos de la clase mayoritaria (verde) hasta igualar con la minoritaria (azul) teniendo en cuenta las distancias con los tres puntos más cercanos de la clase minoritaria ("n_neighbors" = 3). Fuente: elaboración propia.

Una vez balanceada la base de datos se procede a realizar las particiones para el entrenamiento, validación y test. Primero, se divide la base de datos balanceada en dos particiones, entrenamiento y test. La partición de entrenamiento supondrá un 80% de los datos y la de test el 20% restante. Esta división se realizará mediante la función train_test_split() de la librería sklearn, especificando el argumento "test_split" = 0.2. La partición de entrenamiento se utilizará para entrenar y validar los modelos, y se realizará utilizando una validación cruzada denominada *k-fold (Berrar, 2019)*. Para realizar la validación cruzada se determina el valor de k, que se refiere al número de grupos. La partición de entrenamiento se dividirá en k grupos de los cuáles se utilizarán k-1 para el entrenamiento y el sobrante como validación de dicho entrenamiento. Se van realizando iteraciones hasta que todos los grupos se han utilizado como grupo de validación. Finalmente, se utilizará la partición de test (el 20% de los

datos separados inicialmente) para comprobar los resultados de los modelos en datos no vistos anteriormente y compararlos. Se presenta en la Figura 3 una representación de la creación de las particiones.

La normalización y escalado de los datos se realizará mediante la función StandardScaler() de la librería sklearn. Esta función retira la media de las variables y las escala entorno al 0, es decir, la media pasa a ser 0. La normalización y el escalado de los datos se utiliza para hacer que variables de diferentes magnitudes sean comparables entre sí y que están todas dentro de la misma escala, evitando así que una variable parezca más importante que otra únicamente por su rango de valores (Walach et al., 2018).

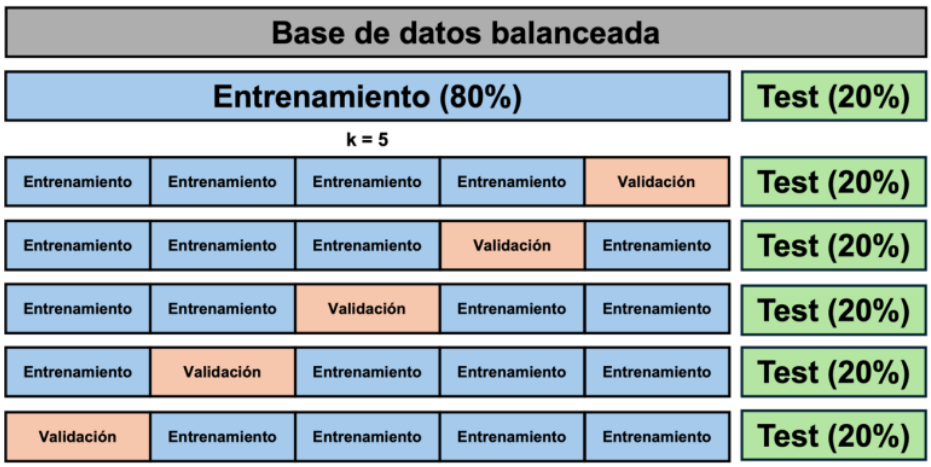

Figura 3. Esquema de la partición de los datos balanceados. Se divide en dos particiones, 80% de los datos se destinan al entrenamiento y el 20% a la evaluación de los modelos. La partición de entrenamiento se utiliza para entrenar los modelos mediante validación cruzada. Una k = 5 significa la división en cinco de los datos que se usarán para entrenar. Fuente: elaboración propia.

3.4. Modelos

Una vez obtenidas las particiones de datos finales se procede a implementar los modelos de IA encargados de predecir la diabetes tipo 2. La implementación de todos los modelos se realizará de manera similar, mediante un script de Python se importará el modelo mediante la librería sklearn y se definirán los hiperparámetros óptimos, los cuales se encontrarán mediante la función GridSearchCV(). Esta función toma como argumento el modelo

importado, el valor de k (para realizar la validación cruzada explicada anteriormente) y un diccionario de Python donde se incluye el hiperparámetro y los diferentes valores que queremos que tome. La función entrena el modelo con todas las combinaciones posibles de los hiperparámetros y devuelve los resultados obtenidos. En este proyecto se implementarán y evaluarán cuatro modelos, los cuáles se detallan a continuación:

- *Support Vector Machine (SVM)*

Esta familia de modelos fue presentada por (Boser et al., 1992) como alternativa a los modelos existentes hasta el momento y con el objetivo de mejorar la generalización a datos no vistos anteriormente. El objetivo principal de los SVM es encontrar un hiperplano que separe de la mejor manera posible los datos en un espacio multidimensional. Este hiperplano puede ser definido por una función lineal (Figura 4A), pero generalmente, los problemas complejos requieren de funciones no lineales (Figura 4B) (Mammone et al., 2009).

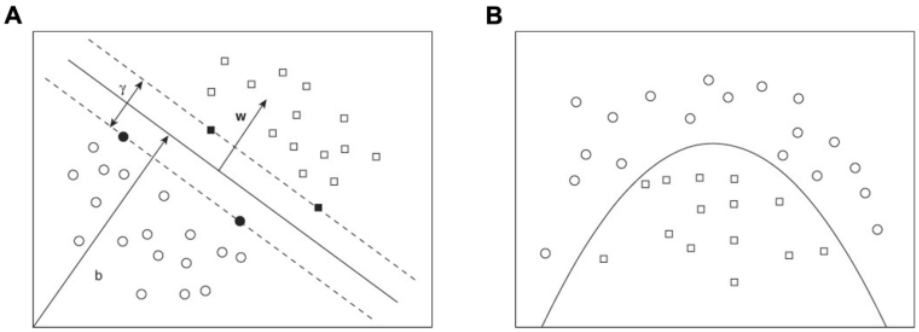

Figura 4. Representación de los hiperplanos en un SVM. A) Hiperplano definido por una función lineal. B) Hiperplano definido por una función no lineal. Fuente: (Mammone et al., 2009).

Este algoritmo ha sido utilizado para múltiples aplicaciones como análisis de expresión génica, predicción de homología en proteínas y clasificación de texto (Mammone et al., 2009).

Se implementará mediante la función svm.SVC()[2] y se ajustarán los siguientes hiperparámetros: “C”, “gamma” y “kernel”. El hiperparámetro “C” es

[2] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

un número decimal, obligatoriamente positivo, y es el factor de regularización. La fuerza de la regularización es inversamente proporcional al valor de "C". El argumento "kernel" tiene valores predefinidos por la función entre los que se puede elegir y define la fórmula matemática utilizada para encontrar el hiperplano que separa las clases. El hiperparámetro "gamma" toma valores decimales positivos e influye en la importancia que se le da a cada uno de los datos de entrenamiento, a menor "gamma" mayor importancia.

- *Random Forests (RF)*

Este modelo destaca por su combinación entre buenos resultados y fácil interpretabilidad, fue presentado por (Breiman, 2001) y está basado en árboles de decisión en los que cada árbol depende de una serie de variables aleatoriamente elegidas de la base de datos. Estos algoritmos utilizan un principio denominado *bagging* en el que se realizan subconjuntos de los datos de entrenamiento (conocido como *bootstraping*) para entrenar modelos individuales que son utilizados para predecir la clase de la muestra de interés mediante un sistema de "votación", es decir, cada modelo individual realiza su predicción y aquella que más se repite es el resultado del RF (Schauberger et al., 2024). Los RF son muy utilizados porque cuentan con numerosas ventajas como que pueden ser usados con datos continuos o categóricos, pueden ser escalados a una gran cantidad de datos y no son muy afectados por valores atípicos (Cutler et al., 2012).

Se implementará mediante la función RandomForestClassifier() [3] y se ajustarán los siguientes hiperparámetros: "n_estimators", "max_depth", "max_features" y "min_samples_leaf". Todos los hiperparámetros toman como valor números enteros positivos. El argumento "n_estimators" determina el número de árboles utilizados en el modelo y "max_depth" la profundidad máxima de cada árbol. El máximo número de variables a tener en cuenta al dividir un nodo viene determinado por el hiperparámetro "max_features" y el número mínimo de muestras en cada nodo viene determinado por "min_samples_leaf".

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

- *Naive Bayes (NB)*

Este modelo está basado en el Teorema de Bayes, que sigue la siguiente ecuación (1):

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \qquad (1)$$

Donde P(A) es la probabilidad de A sin tener en cuenta B y donde P(B) es la probabilidad de B sin tener en cuenta A. P(A|B) se refiere a la probabilidad de A sabiendo B mientras que P(B|A) es la probabilidad de B sabiendo A (Z. Zhang, 2016). En un problema de clasificación, A es la etiqueta que queremos predecir (por ejemplo, diabetes o no diabetes) y B son las variables a tener en cuenta (presión arterial, colesterol, etc.). En este caso la probabilidad de A se convertiría en una fórmula más compleja, P(B1, B2, B3…|A), por este motivo surge el modelo Naive Bayes que presupone que las variables son independientes entre sí lo que hace que el cálculo sea más sencillo, P(B1|A) x P(B2|A) x P(B3|A) (Z. Zhang, 2016).

Este modelo se implementará mediante la función BernoulliNB()[4] y se ajustará el hiperparámetro "alpha" que toma valores decimales positivos y que se añade para evitar probabilidades igual cero.

- *Multilayer perceptron (MLP)*

Este algoritmo es el tipo más utilizado de red neuronal, está compuesto por una capa de entrada con una serie de neuronas que se corresponden con las variables a estudiar, una capa de salida con las clases que se quieren predecir y entre ambas se encuentran una serie de capas denominadas capas ocultas (Wang, 2003). La estructura general de un MLP se presenta en la Figura 5. Las neuronas están conectadas entre sí y tienen asociado un valor, denominado peso, que representa la fuerza de la conexión y que se determina durante el entrenamiento (Wang, 2003). Uno de los elementos más importantes del MLP son las funciones de activación que se utilizan para aplicar transformaciones no lineales a los datos (Dubey et al., 2022). Un concepto clave en los MLP es el *backpropagation* presentado por (Rumelhart et al., 1986) que utiliza el error

[4] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html

obtenido en la capa de salida para ajustar los pesos de las capas anteriores de forma repetida para mejorar el desempeño de la red neuronal.

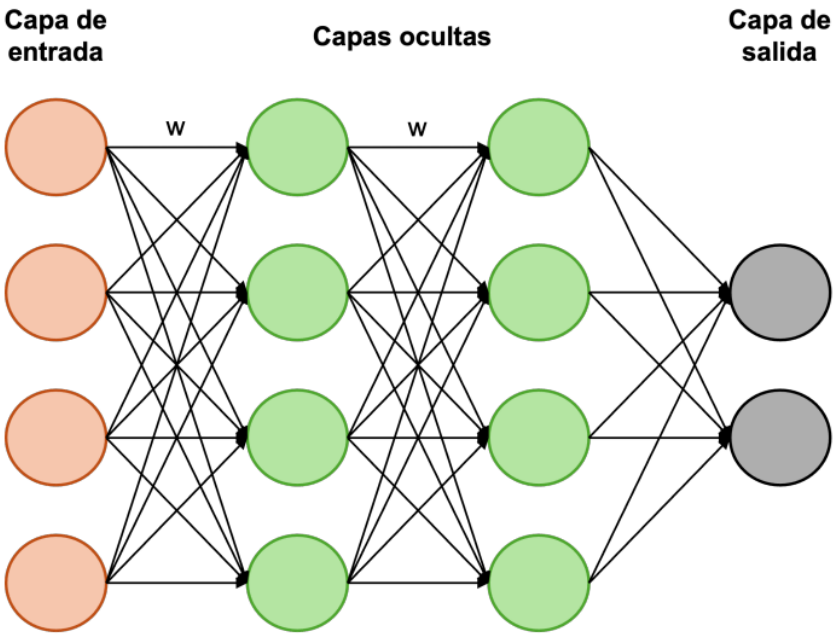

Figura 5. Esquema de la estructura de un MLP. Se pueden observar las capas de entrada (naranja) y salida (gris) además de las capas ocultas (verde). Los pesos (o "weights") asociados a la conexión entre neuronas viene representado por "w". Fuente: elaboración propia.

Este modelo se implementará mediante la función MLPClassifier() [5] incluido dentro del paquete "neural_network" de la librería sklearn y se ajustarán los siguientes hiperparámetros: "hidden_layer_sizes", "activation", "alpha" y "learning_rate". El número y el tamaño de las capas viene determinado por "hidden_layer_sizes" y se especifica con una serie de números entre paréntesis y separados por comas. Por ejemplo, si queremos un modelo MLP con dos capas ocultas de 32 y 64 neuronas el valor pasado al argumento sería (32, 64). La función de activación se elige de entre uno de los valores proporcionados por la función con el argumento "activation". La fuerza de la regularización se determina

[5] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

con “alpha” que toma como valores números decimales positivos y la velocidad de aprendizaje puede ser constante o adaptativa y se determina con el argumento “learning_rate”.

3.5. Evaluación y comparación de los modelos

Tras el entrenamiento de los modelos se procederá a evaluar su rendimiento para su posterior comparación. Para ello se utilizarán una serie de métricas que se detallan a continuación:

- Matriz de confusión

La matriz de confusión se utilizará para obtener posteriormente las diferentes métricas que se quieren comparar. La matriz de confusión se obtiene al cruzar las clases reales con las clases predichas por el modelo (Ting, 2011). De este cruce obtenemos cuatro parámetros: verdaderos positivos (TP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN).

- Exactitud

Esta métrica informa del rendimiento general del modelo, pero es sensible a clases no balanceadas. Su fórmula es la siguiente (2):

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \qquad (2)$$

- Precisión

Informa del rendimiento en las predicciones positivas. Su fórmula es la siguiente (3):

$$Precisión = \frac{TP}{TP + FP} \qquad (3)$$

- Sensibilidad

Informa de la capacidad del modelo de identificar un caso positivo como positivo. Su fórmula es la siguiente (4):

$$Sensibilidad = \frac{TP}{TP + FN} \qquad (4)$$

- F1 *score*

Es una métrica útil en clases no balanceadas y en la comparación de modelos. Su fórmula es la siguiente (5):

$$F1\ score = \frac{2TP}{2TP + FP + FN} \qquad (5)$$

- Curva ROC y área bajo la curva

Para graficar la curva ROC y calcular el área bajo la curva es necesario calcular el ratio de verdaderos positivos (TPR) (6) y verdaderos negativos (FPR) (7). Las fórmulas son las siguientes:

$$TPR = \frac{TP}{TP + FN} \qquad (6)$$

$$FPR = \frac{FP}{TN + FP} \qquad (7)$$

4. RESULTADOS

4.1. Análisis exploratorio de la base de datos

El primer paso es ver si la base de datos presenta duplicados lo cual se hace con la función duplicated() de la librería pandas. Se encuentran 24206 registros duplicados y se procede a eliminarlos con la función drop_duplicates() de la misma librería.

Tras el graficado de la variable principal del estudio (Figura 6A), la cual indica la presencia de diabetes/prediabetes o ninguna podemos observar que la mayor parte de la base de datos está compuesta de personas no diabéticas (194377 personas) seguida por las personas diabéticas (35097 personas). Una minoría de la base de datos presenta prediabetes y se encuentran incluidos en la clase de diabetes. Posteriormente, se realizó el graficado de las variables demográficas de la base de datos. Se puede observar (Figura 6B) que la base de datos está compuesta mayoritariamente por mujeres y que todos los participantes son mayores de edad. El rango de edad más representado (Figura 6C) es el de 60-64 años seguido de los de 65-69 y 55-59 años.

En cuanto al nivel educativo (Figura 6D) la mayor parte de los participantes se han graduado del instituto y han ido a la Universidad, contando una gran parte de ellos con el graduado universitario. La Figura 6E presenta los ingresos reportados por cada participante, se puede observar que el rango con mayor número de casos es el de más de 75.000$ al año. Posteriormente, se analizan los hábitos saludables de los participantes. Los resultados se presentan en la Tabla 2. Un 53.4% de los participantes son considerados no fumadores mientras que el 46.6% restante reportan haber fumado más de 100 cigarros en su vida. La variable sobre el alcoholismo está menos dividida ya que únicamente 13950 personas declaran consumir grandes cantidades de alcohol, lo que supone un 6.1% de la base de datos. El 73.3% de los entrevistados reportan haber realizado ejercicio físico en los últimos 30 días. En cuanto a los hábitos alimenticios, el 61.3% de los entrevistados come fruta una o más veces al día y está cifra se eleva a 79.5% cuando se pregunta por el consumo diario de vegetales.

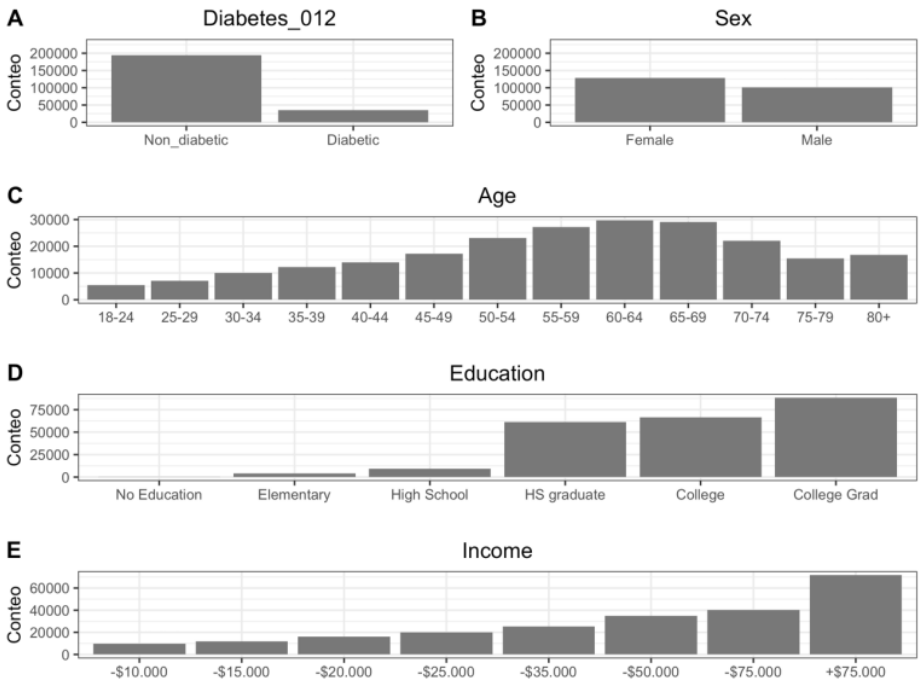

Figura 6. Representación de las variables demográficas de la base de datos. A) Valores de la variable Diabetes_binary. B) Distribución del género en la base de datos. C) Frecuencia de los rangos de edad en la base de datos. D) Nivel educativo en la base de datos. E) Ingresos anuales en la base de datos. Fuente: elaboración propia.

Variable	Si (conteo)	No (conteo)	Si (%)	No (%)
Smoker	106889	122585	46.6	53.4
HvyAlcoholConsump	13950	215524	6.1	93.9
PhysActivity	168214	61260	73.3	26.7
Fruits	140593	88881	61.3	38.7
Veggies	182337	47137	79.5	20.5

Tabla 2. Tabla de frecuencias de las variables relacionadas con hábitos saludables. Fuente: elaboración propia.

Tras el análisis de las variables relacionadas con la salud (Figura 7) comprobamos que la mayoría de los participantes no tiene ni la presión arterial ni el colesterol elevado, pero existe un gran número de personas que sí cuentan

con estos valores elevados. Sin embargo, si profundizamos en estas dos variables y las dividimos según la variable "Diabetes_binary" podemos observar que se comportan de manera diferente según el estado del participante. En la Tabla 3 se presentan los datos de la presión arterial divididos por el valor de "Diabetes_binary", podemos observar que en los casos de participantes no diabéticos la clase predominante es la de presión arterial normal mientras que en los prediabéticos/diabéticos la clase predominante es la de presión arterial elevada. Esto mismo ocurre cuando se estudian los niveles de colesterol (Tabla 4). Estos resultados son de esperar ya que ambas variables elevadas son características conocidas de la diabetes tipo 2.

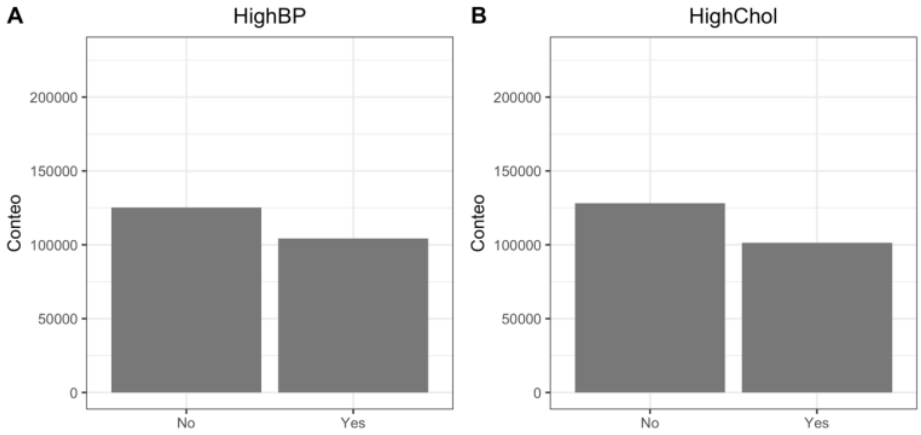

Figura 7. Distribución de variables relacionadas con la salud. A) Distribución de la variable HighBP. B) Distribución de la variable HighChol. Fuente: elaboración propia.

Diabetes_binary	HighBP	Conteo	% Total
No diabetes	Normal	116522	50.8%
	Elevado	77855	33.9%
Diabetes	Normal	8692	3.8%
	Elevado	26405	11.5%

Tabla 3. Tabla de frecuencias de la variable HighBP dividido por Diabetes_binary. Fuente: elaboración propia.

Diabetes_binary	HighChol	Conteo	% Total
No diabetes	Normal	116528	50.8%
	Elevado	77849	33.9%
Diabetes	Normal	11601	5.1 %
	Elevado	23496	10.2%

Tabla 4. Tabla de frecuencias de la variable HighChol dividido por Diabetes_binary. Fuente: elaboración propia.

Posteriormente, analizamos la distribución del índice de masa corporal (BMI) en la base de datos, los resultados se presentan en la Figura 8A. Podemos ver que en todos los casos la mayoría de los participantes se sitúa en un BMI superior a 25 lo cual indica sobrepeso. La gran mayoría de los participantes reportan no haber tenido nunca cardiopatía o infarto (Figura 8B) ni ataque cerebrovascular (Figura 8C). Cerca de 5000 personas tienen dificultad al caminar y al subir escaleras (Figura 8D).

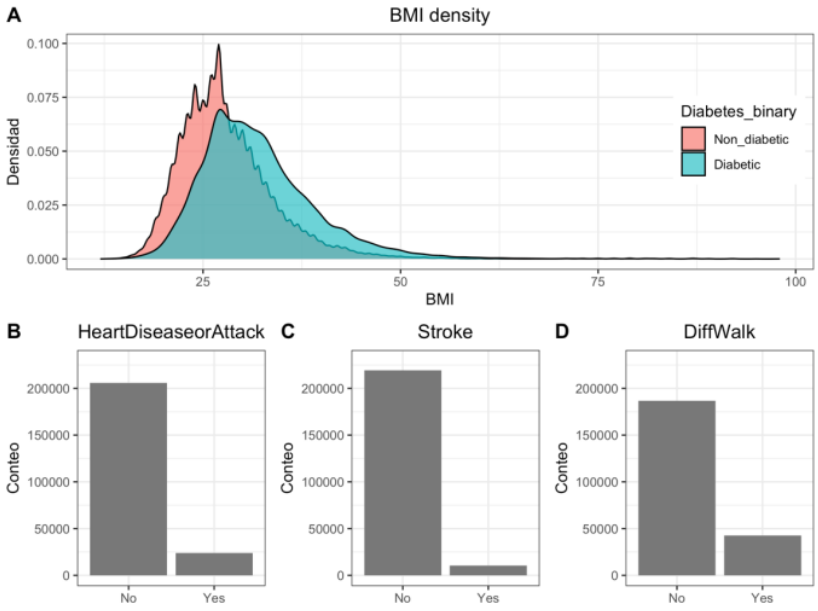

Figura 8. Representación de variables relacionadas con la salud. A) Densidad de la variable BMI según Diabetes_012. B) Distribución de la variable HeartDiseaseorAttack. C) Distribución de la variable Stroke. D) Distribución de la variable DiffWalk. Fuente: elaboración propia.

Por último, analizamos la correlación de las variables para identificar aquellas que no guardan relación con la variable de interés. Para este análisis se utilizan las funciones cor() y corrplot() en R para calcular la correlación y realizar el graficado (Figura 9). Se pueden observar variables que, como es de esperar, guardan mayor relación con la diabetes (valores de correlación alejados de 0) como la presión arterial ("HighBP"), colesterol ("HighChol") y el índice de masa corporal ("BMI"). Otras variables menos evidentes que guardan relación con la diabetes son los ingresos ("Income") y la educación ("Education"). Hay variables que guardan poca relación con la variable principal, aquellas que tienen correlación cercana a 0 en la Figura 9, como "Fruits", "Veggies", "Sex", "CholCheck" y "AnyHealthcare". Estas variables serán eliminadas para el entrenamiento de los modelos.

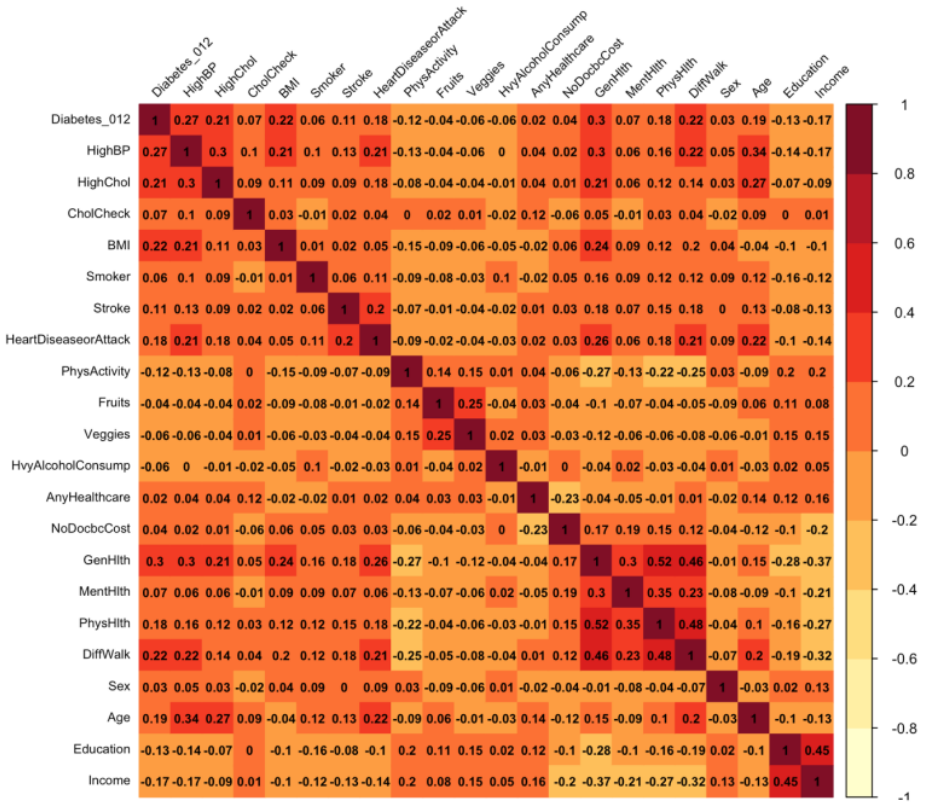

Figura 9. Representación de la correlación entre las variables de la base de datos. Las variables con valores más cercanos a 1 (colores más oscuros) tienen correlación positiva mientras que las más cercanas a -1 (colores más claros) tienen correlación negativa Fuente: elaboración propia.

4.2. Balanceo, división y normalización de los datos

Una vez realizado el análisis exploratorio de los datos y eliminadas las variables que no guardan relación con la diabetes se procede a realizar el balanceo de los datos para su posterior división en entrenamiento y test. La clase mayoritaria en la base de datos es la de no diabéticos (194377 casos) por lo que habrá que reducirla hasta que haya aproximadamente el mismo número de casos que la clase de diabéticos (35097 casos). Esto se hace para evitar que los diferentes modelos clasifiquen únicamente la variable mayoritaria. El balanceo se realiza mediante la función NearMiss() de la librería imblearn, especificando los argumentos "versión" = 1 y "n_neighbors" = 10. Tras utilizar esta función se obtiene una base de datos con 70194 registros los cuáles son 50% diabéticos y 50% no diabéticos.

Tras el balanceo se procede a la división de la base de datos utilizando la función train_test_split() de la librería sklearn. Se especifica el tamaño de la partición de test con el argumento "test_size" (en este caso "test_size" = 0.2). De esta forma obtenemos una partición de entrenamiento que contiene el 80% de los datos y una de test que contiene el 20%. Por último, se realiza la normalización de los datos mediante la función StandardScaler() de la librería sklearn.

4.3. Entrenamiento de *Random Forest*

Una vez obtenidas las particiones de datos se entrena el modelo *Random Forest* como se ha explicado anteriormente, se utiliza la función GridSearchCV() en la cual se especifica el modelo a utilizar con la función RandomForestClassifier() y la matriz de hiperparámetros. Los hiperparámetros utilizados y sus respectivos valores están representados en la Tabla 5.

Hiperparámetro	Valores
"max_depth"	10, 12, 15, 17
''n_estimators"	10, 50, 100, 150, 200
"max_features"	2, 5, 7, 10
"min_samples_leaf"	2, 3, 4, 5

Tabla 5. Valores de los hiperparámetros utilizados para Random Forest. Fuente: elaboración propia.

Teniendo en cuenta los hiperparámetros indicados anteriormente y que se realiza una validación cruzada con k = 5 se va a entrenar el RF un total de 1600 veces con 320 combinaciones de valores diferentes. En cada entrenamiento obtenemos lo valores de precisión, sensibilidad y exactitud tanto para la partición de entrenamiento como para la partición de validación. De estos valores se obtiene la media de los cinco entrenamientos con los mismos hiperparámetros. En la Figura 10A están representados los valores de precisión y sensibilidad obtenidos con ambas particiones. Podemos observar que los resultados obtenidos en el entrenamiento son más variados y siempre mejores que los obtenidos en la validación, los valores de precisión están por encima de 0.75 y llegan hasta 0.90 mientras que los de la sensibilidad se encuentran sobre 0.70 y su valor máximo está entorno a 0.85. Los resultados en la validación se encuentran más concentrados y se pueden ver de manera ampliada en la Figura 10B. La mayor parte de los valores se encuentra entorno al 0.75 de precisión y entre 0.710 y 0.715 de sensibilidad.

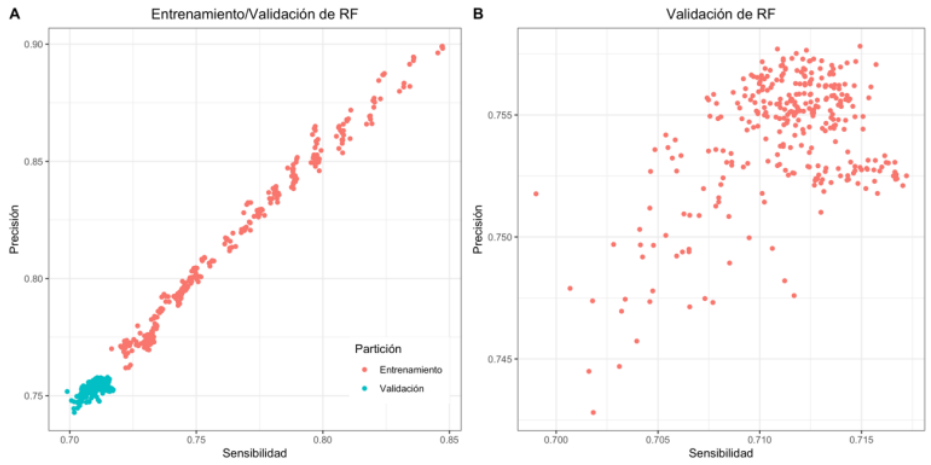

Figura 10. Resultados del entrenamiento del RF. A) Resultados de entrenamiento y validación. B) Resultados de la validación ampliados. Fuente: elaboración propia.

Para escoger los valores de los hiperparámetros más adecuados nos centramos en los valores de exactitud. El conjunto de hiperparámetros que obtiene mejores resultados, y el cuál usaremos para la comparación de modelos, está representado en la Tabla 6. El modelo con estos hiperparámetros obtiene una exactitud de 0.743 en la evaluación.

Hiperparámetro	Valor
"max_depth"	15
''n_estimators"	200
"max_features"	5
"min_samples_leaf"	4

Tabla 6. Valores de los hiperparámetros con mejor resultado en Random Forest. Fuente: elaboración propia.

4.4. Entrenamiento de *Naive Bayes*

Para el entrenamiento del *Naive Bayes* solo es necesario ajustar el hiperparámetro "alpha", los valores utilizados se presentan en la Tabla 7. Teniendo en cuenta la validación cruzada se entrenaron un total de 60 modelos con 15 valores distintos de "alpha".

Hiperparámetro	Valores
"alpha"	0.0, 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 10.0, 20.0, 40.0, 50.0

Tabla 7. Valores de los hiperparámetros utilizados para Naive Bayes. Fuente: elaboración propia.

Los resultados obtenidos en el entrenamiento y en la validación se presentan en la Figura 11. Se puede observar que los resultados presentan una tendencia contraria a los obtenidos en el RF ya que a medida que aumenta la sensibilidad disminuye la precisión. Los valores de precisión obtenidos con ambas particiones se encuentran sobre 0.72 y cercanos a 0.73 mientras que los valores de sensibilidad no superan en ningún caso el 0.68. Estos valores de sensibilidad suponen los valores más bajos obtenidos en el entrenamiento de los modelos. En la Figura 11B se presentan únicamente los valores obtenidos con la partición de evaluación, se observan valores con una precisión más alta que el resto pero que, en cambio, tienen una peor sensibilidad que las demás combinaciones e hiperparámetros.

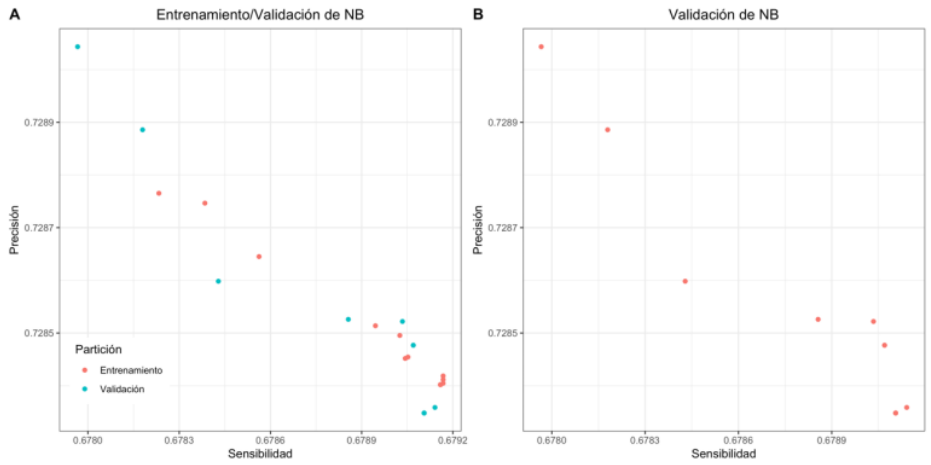

Figura 11. Resultados del entrenamiento del NB. A) Resultados de entrenamiento y validación. B) Resultados de la validación ampliados. Fuente: elaboración propia.

El valor de "alpha" que presenta mejores resultados es 5.0 y obtiene una exactitud de 0.713 en la evaluación. El entrenamiento de este modelo se ha destacado por su rapidez, ya que cada una de las combinaciones tuvo un tiempo medio de entrenamiento inferior a un segundo, convirtiéndolo en el modelo significativamente más rápido de todos los implementados.

4.5. Entrenamiento de SVM

Para el entrenamiento del SVM se utiliza la misma metodología y los valores utilizados para los hiperparámetros se presentan en la Tabla 8. Se van a probar dos valores para "kernel" y para cada uno de ellos los valores de "C" indicados en la Tabla 8. El parámetro "gamma" solo se utiliza cuando el "kernel" utilizado es "rbf". Combinando todos los valores y teniendo en cuenta la validación cruzada se entrenan un total de 140 modelos en 28 combinaciones diferentes.

Hiperparámetro	Valores
"C"	1, 10, 100, 300, 500, 750, 1000
''kernel"	"linear", "rbf"
"gamma"	0.01, 0.001, 0.0001

Tabla 8. Valores de los hiperparámetros utilizados para SVM. Fuente: elaboración propia.

Los resultados obtenidos en el entrenamiento con ambas particiones se presentan en la Figura 12A. La mayor parte de los valores obtenidos de sensibilidad tanto de entrenamiento como de validación se encuentran por debajo de 0.70 mientras que los valores de precisión van desde 0.75 hasta 0.80. En este modelo se puede observar que los valores obtenidos en el entrenamiento son, en general, parecidos a los obtenidos en la validación. Si nos fijamos únicamente en la evaluación del entrenamiento (Figura 12B) podemos observar que los valores de precisión y sensibilidad parecen tener una correlación inversa, a medida que aumenta la sensibilidad disminuye la precisión.

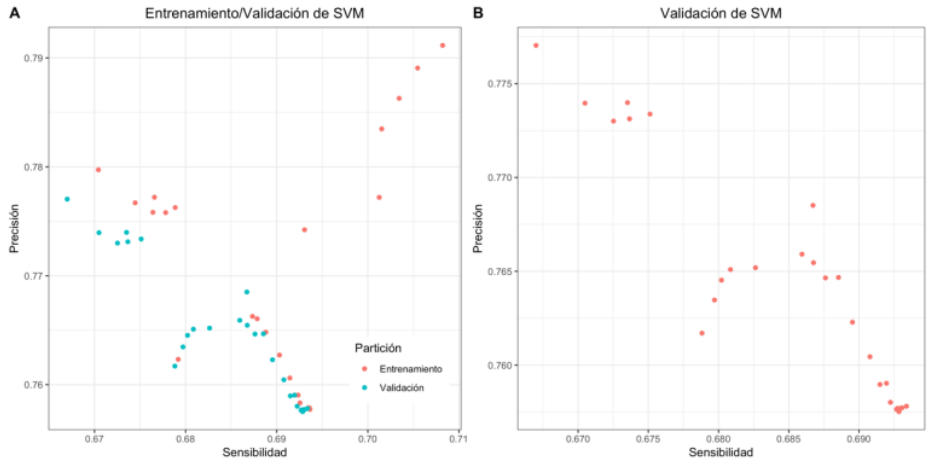

Figura 12. Resultados del entrenamiento del SVM. A) Resultados de entrenamiento y validación. B) Resultados de la validación ampliados. Fuente: elaboración propia.

El modelo con los hiperparámetros presentados en la Tabla 9 obtiene los mejores resultados con una exactitud de 0.739.

También comentar sobre el SVM que los tiempos de entrenamiento han sido ampliamente superiores a los demás modelos utilizados con un tiempo de entrenamiento máximo de 168 minutos. El parámetro “kernel” es el que más influye en el tiempo ya que cuando este es “linear” (66,6 minutos de media) los tiempos de entrenamiento son mayores que cuando es “rbf” (1,6 minutos de media).

Hiperparámetro	Valores
"C"	10
''kernel"	"rbf"
"gamma"	0.01

Tabla 9. Valores de los hiperparámetros con mejor resultado en SVM. Fuente: elaboración propia.

4.6. Entrenamiento del MLP

Para el entrenamiento del MLP hay que asignar el número y tamaño de las capas ocultas, para ello se prueba una serie de combinaciones presentadas en la Tabla 10. También se prueban diferentes funciones de activación, "logistic", "tanh" y "relu" y diferentes velocidades de aprendizaje ("learning_rate"). También se ajusta el factor de regularización "alpha". Todos los valores utilizados se presentan en la Tabla 10. Se realizan un total de 900 entrenamientos con 180 combinaciones de hiperparámetros.

Hiperparámetro	Valores
"hidden_layer_sizes"	(32), (32,64), (32,64,128), (64), (64,32), (128,64,32), (16), (128), (8), (8,8)
''activation"	"logistic", "tanh", "relu"
"alpha"	0.0001, 0.001, 0.01
"learning_rate"	"constant", "adaptive"

Tabla 10. Valores de los hiperparámetros utilizados para MLP. Fuente: elaboración propia.

Los resultados obtenidos se presentan en la Figura 13. Como se puede observar en la Figura 13A hay una distinción clara entre los resultados obtenidos con la partición de entrenamiento y con la de validación. Los resultados de entrenamiento tienen valores por encima de 0.7 para la sensibilidad y por encima de 0.75 para la precisión. Hay valores de entrenamiento muy cercanos a 1 en ambas métricas, pero en la validación se quedan sobre 0.75 para la precisión y 0.71 en sensibilidad lo que indica que se está produciendo un sobreajuste en el entrenamiento que luego no es capaz de generalizar a los datos no vistos anteriormente. En la Figura 13B podemos observar únicamente los resultados de la validación y se puede ver como a medida que sube una de las métricas también sube la otra a diferencia de lo que pasaba en el SVM.

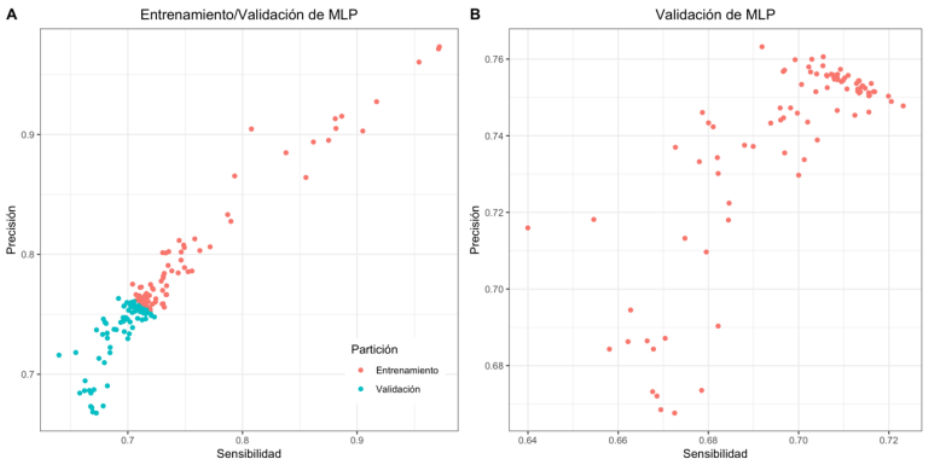

Figura 13. Resultados del entrenamiento del MLP. A) Resultados de entrenamiento y validación. B) Resultados de la validación ampliados. Fuente: elaboración propia.

La combinación de hiperparámetros que obtiene mejores resultados se presenta en la Tabla 11. Este modelo obtiene una exactitud de 0.742 en la validación.

Hiperparámetro	Valor
"hidden_layer_sizes"	(64)
''activation"	"logistic"
"alpha"	0.01
"learning_rate"	"constant"

Tabla 11. Valores de los hiperparámetros con mejor resultado en MLP. Fuente: elaboración propia.

4.7. Comparación de los modelos

Una vez obtenidos los mejores hiperparámetros para cada modelo se procedió a evaluar sus resultados con la partición de test (el 20% de los datos divididos inicialmente). Se retiró la etiqueta de los datos (la variable "Diabetes_binary") y se utilizó cada modelo para predecir la clase a la que pertenece. La matriz de confusión obtenida se presenta en la Figura 14. El eje "y" se corresponde con la etiqueta verdadera mientras que el eje "x" con la

etiqueta predicha por los modelos. Siguiendo la codificación de las variables explicada anteriormente y presentada en el Anexo 3 la etiqueta 0 se corresponde con la clase de no diabéticos mientras que la etiqueta 1 con la clase de diabéticos/prediabéticos. Se puede observar que todos los modelos han sido capaces de clasificar correctamente la mayor parte de la base de datos de test. En la Figura 14A se presentan los resultados del SVM en los que se ve que es el modelo que mejores resultados ha tenido en la clase no diabética ya que cuenta con el mayor número de verdaderos negativos (5594) y el menor número de falsos positivos (1432). Sin embargo, este modelo también es el que ha obtenido los peores resultados en la clase positiva, llegando a los 2165 falsos negativos. En las Figuras 14B y 14C se presentan los resultados del RF y MLP respectivamente, ambos modelos tienen un desempeño muy similar siendo el RF superior en la clase negativa (5444 verdaderos positivos frente a 5415 del MLP). En cambio, el MLP ha obtenido resultados ligeramente mejores en la clase positiva (5030 verdaderos positivos frente a 5025 del RF).

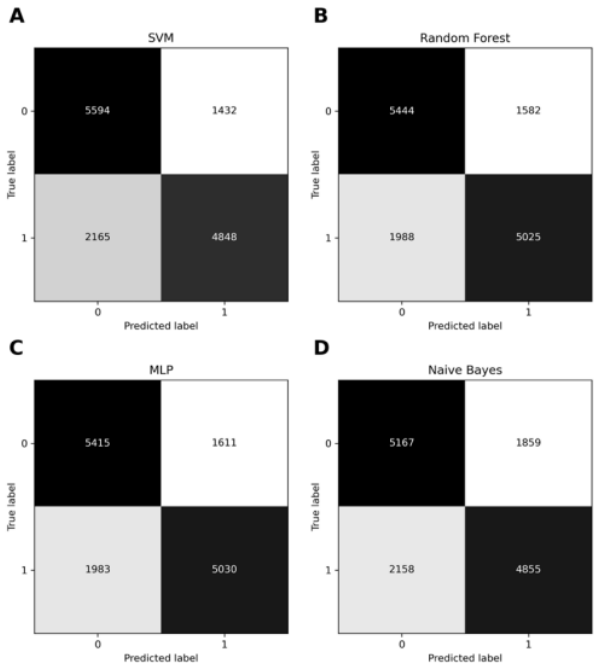

Figura 14. Matrices de confusión con la partición de test. A) Matriz de confusión del *Support Vector Machine*. B) Matriz de confusión del *Random Forest*. C) Matriz de confusión del *Multilayer perceptron*. D) Matriz de confusión del *Naive Bayes*.

Finalmente, en la Figura 14C se presentan la matriz de confusión del NB. Este modelo ha obtenido resultados bastante inferiores en la clase negativa con 248 falsos positivos más que el segundo peor modelo en esta categoría (MLP). Este modelo también ha obtenido resultados considerablemente peores en la clase positiva al compararlo con RF y MLP y solamente ligeramente superior al SVM (una diferencia de únicamente 7 casos mal clasificados).

Con las matrices de confusión se calcularon las métricas que se presentan en la Tabla 12. Se puede observar que las pequeñas diferencias observadas anteriormente se corresponden con los valores obtenidos en las métricas, el modelo NB que tenía los peores resultados en general es el que obtiene menor valor de exactitud (0.7138). Los otros tres modelos obtienen valores muy similares en la exactitud entre ellos. El SVM a pesar de tener el mejor desempeño en la precisión (0.7719) se ve afectado por su pobre rendimiento en la sensibilidad (0.6913) lo cual lo penaliza a la hora de calcular el F1 Score ya que obtiene valores inferiores a MLP y RF pese a tener una exactitud muy similar. Tanto el RF como el MLP son los que mejores resultados obtienen en las métricas más generales (exactitud y F1 Score) y las diferencias entre ellos son mínimas.

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.7457	0.7605	0.7165	0.7379
SVM	0.7437	0.7719	0.6913	0.7294
Naïve Bayes	0.7138	0.7231	0.6923	0.7074
MLP	0.7440	0.7574	0.7172	0.7368

Tabla 12. Resultados de los modelos con la partición de test. Fuente: elaboración propia.

Por último, se graficaron las curvas ROC y *precision-recall* (precisión-sensibilidad) para corroborar los resultados obtenidos hasta el momento. En la Figura 15A se presenta las curvas ROC de los cuatro modelos y el cálculo del área bajo la curva (AUC). Se puede observar que, siguiendo con los resultados expuestos hasta ahora, las curvas de los modelos SVM, RF y MLP son muy similares y cuentan con la misma área bajo la curva (0.83) mientras que el NB, que se destacaba por tener los peores resultados, obtiene el peor AUC (0.79).

Esto se puede seguir observando en la curva de *precisión-recall* (Figura 15B) en las que una vez más el NB obtiene los peores resultados (0.79) y los otros tres modelos obtienen resultados idénticos (0.85).

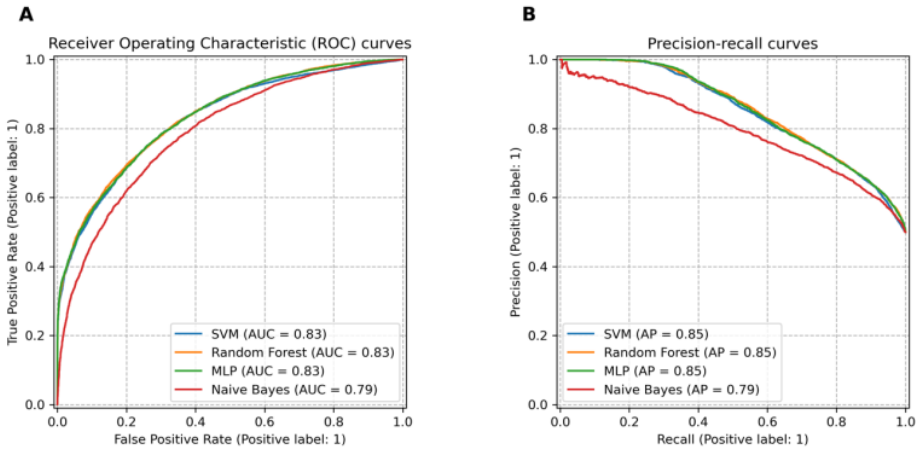

Figura 15. Curvas obtenidas con la partición de test. A) Curva ROC y área bajo la curva calculada para los modelos. B) Curva de *precision-recall* y área calculada para los modelos.

5. DISCUSIÓN

El gran problema sanitario que supone la diabetes tipo 2 en adultos y la necesidad de aumentar las tasas de predicción precoz para evitar las complicaciones asociadas a esta conlleva la exploración de nuevos métodos diagnósticos. Siendo la Inteligencia Artificial una herramienta cada vez más ampliamente usada en el sector sanitario se presenta como una gran alternativa para la predicción de esta enfermedad. Por este motivo, este proyecto de investigación planteaba el uso de modelos de IA para la predicción precoz de la diabetes tipo 2 en adultos.

Para esta tarea se ha utilizado un subconjunto de datos del año 2015 del estudio BRFSS (CDC, 2024b) llevado a cabo por la CDC en los Estados Unidos. Este subconjunto de datos está compuesto por 253680 registros de adultos para 22 variables. Tras un primer análisis en busca de duplicados se realizó un análisis exploratorio de los datos en los que se observó que la mayoría de las personas no tenían el colesterol ni la presión arterial elevados (Figura 7A y B). Sin embargo, al dividir estas dos características según la variable "Diabetes_binary" (Tabla 3 y Tabla 4) encontramos que eso solo se cumple en el caso de las personas no diabéticas y que el número de personas con el colesterol y la presión arterial elevada es mucho mayor en las personas diabéticas que las personas con ambos valores normales. Esta observación se corresponde con lo encontrado en la literatura, ya que la hipertensión es mucho más frecuente en las personas con diabetes que en las personas sin esta enfermedad (Petrie et al., 2018). Otra observación interesante obtenida en el análisis exploratorio es la correlación de los ingresos y la educación con la enfermedad (Figura 9). Ambas parecen tener una correlación negativa con la diabetes tipo 2, es decir, cuanto menor es el nivel educativo o de ingresos mayor es la probabilidad de tener diabetes. *A priori,* la correlación entre la educación y los ingresos parece clara ya que al tener un mayor nivel educativo se puede tener acceso a trabajos con mejores ingresos. Sin embargo, la correlación entre los ingresos y la diabetes no está clara. Esa correlación negativa podría deberse a la diferencia en el estilo de vida entre las personas con ingresos altos y bajos. Una persona con ingresos altos podría permitirse alimentos de mayor cualidad e incluso mejoras en el estilo de vida como, por ejemplo, el pago mensual de un

gimnasio que permita realizar ejercicio regularmente. Esta correlación negativa hace pensar que la economía tiene un papel fundamental en la diabetes tipo 2 y esto se corrobora con la información disponible en la literatura, en la que se reporta que los mayores aumentos en la incidencia de la diabetes se han producido en los países con ingresos medios o bajos mientras que en los países con mayores ingresos la tendencia es mucho menos pronunciada (Rosengren, 2018).

Una vez realizado el análisis exploratorio de los datos se procedió a entrenar y validar los modelos. Se han implementado cuatro modelos ampliamente utilizados en la literatura, *Random Forest, Naive Bayes, Support Vector Machine* y *Multilayer perceptron.* En este trabajo de investigación el RF y el MLP han obtenido mejores resultados que los otros dos modelos (F1 Score de 0.7379 y 0.7368 respectivamente y AUC de 0.83 en ambos casos) y parecen estar mejor cualificados para problemas de predicción de este tipo, lo que se corresponde con otros trabajos de investigación como el realizado por (Nurdin et al., 2023) que utiliza una base de datos similar. El SVM ha conseguido los mejores resultados en la precisión, pero, en cambio, los peores en la sensibilidad lo que provoca que sus resultados generales sean ligeramente inferiores a los otros dos modelos (F1 Score de 0.7294). El NB ha obtenido los peores resultados entre todos los modelos lo que sugiere que no es la elección más adecuada para este tipo de problemas. Teniendo en cuenta los resultados obtenidos el siguiente modelo en ser descartado sería el SVM ya que pese a obtener los mejores resultados en la precisión en este tipo de problemas es preferible elegir aquellos modelos que tengan mejores resultados en la sensibilidad. Es preferible que el modelo clasifique bien a los enfermos, aunque eso suponga algún falso positivo (que después sería correctamente diagnosticado con otras pruebas) a que haya falsos negativos que se queden sin tratamiento. Al tener el RF resultados ligeramente mejores al MLP y al tener una mayor tolerancia al ruido en los datos (Nurdin et al., 2023) se considera el modelo de elección para el problema planteado.

En el caso de una base de datos que se encuentra balanceada la exactitud es una métrica que nos aporta buena información sobre cómo se comportan los modelos pero que estaría sesgada hacia la clase mayoritaria en el caso de una

base de datos sin balancear. El F1 Score y el AUC de las curvas ROC nos aporta información mucho más robusta y sin sesgos por lo que suponen una métrica más eficaz para comparar los modelos. Sin embargo, en este problema en concreto el AUC no ha conseguido demostrar las pequeñas diferencias entre los modelos con mejor desempeño (RF, SVM y MLP todos han obtenido un AUC de 0.83) mientras que el F1 Score sí. Por este motivo se considera que el F1 Score es la métrica de evaluación preferible en este proyecto.

Pese a pequeñas diferencias los resultados obtenidos por los modelos son muy similares entre ellos e incluso no presentaban una mejoría considerable con combinaciones diferentes de hiperparámetros. Esto sugiere que la base de datos tiene un valor predictivo limitado y que para mejorar el rendimiento de los modelos será necesario incorporar nuevas variables o mejorar las ya presentes. Por ejemplo, en vez de tener una variable categórica para la presión arterial se podría tener el valor de la presión sistólica y diastólica o en vez de la variable categórica para el colesterol se podrían tener los valores en sangre de LDL y HDL.

Líneas futuras de trabajo podrían incluir la utilización de un sistema de "votación" entre los modelos similar al concepto de *bagging* en el RF y similar a otros proyectos de investigación encontrados en la literatura como el realizado por (Kalagotla et al., 2021). En este sistema cada uno de los modelos entrenados realizaría una predicción y la que más se repita sería la elegida, de esta manera un modelo podría compensar las carencias de otro. También sería necesario mejorar la base de datos de la forma que se ha explicado anteriormente y también aumentando el número de casos ya que pese a ser una base de datos grande la diferencia entre clases es muy amplia y se pierden muchos datos al realizar el balanceo. Otra línea futura de trabajo es el desarrollo de aplicaciones web para profesionales sanitarios o incluso para el público general utilizando estos modelos para realizar un cribado más rápido y eficaz de la enfermedad.

6. CONCLUSIONES

La primera conclusión obtenida en este proyecto, a través del análisis exploratorio, es que las variables relacionadas con la presión arterial y el colesterol presentan una gran correlación con la diabetes tipo 2 y que pueden ser clave en la predicción de esta enfermedad. También que, pese a estar menos documentado en la literatura, los ingresos pueden suponer un factor importante en el desarrollo de la enfermedad.

El *Random Forest* se ha destacado como el mejor modelo en este caso debido a su resultado obtenido en el F1 Score y su mayor sensibilidad. Por este motivo se considera que es el modelo de elección para el problema planteado y se recomienda su utilización en los pasos futuros.

El F1 Score ha sido el método de evaluación que mejor ha conseguido demostrar las diferencias entre los modelos por ello ha sido la métrica elegida para la elección del modelo con mejor desempeño. Su robustez frente a bases de datos no balanceadas hace que esta métrica sea ideal para la comparación de modelos en distintos tipos de problemas.

Las limitaciones encontradas durante el proyecto suponen futuras líneas de trabajo. La gran diferencia del número casos entre clases y la existencia de variables poco informativas implican que una mejoría en los datos es necesaria para aumentar el desempeño de los modelos. También, de las carencias que se observan en los modelos surge la hipótesis de que una combinación de estos podría ser eficaz para mejorar sus resultados. La necesidad de un cribado más rápido de la enfermedad sugiere que el desarrollo de aplicaciones web basadas en estos modelos podría ser un gran avance en el diagnóstico precoz de la enfermedad.

Una vez realizado este proyecto de investigación se llega a la conclusión de que la Inteligencia Artificial podría ser utilizada para predecir la diabetes tipo 2 en adultos y que podría ser eficaz en prevenir las complicaciones asociadas a esta enfermedad. Todavía queda mucho camino por recorrer y este tipo de herramientas diagnósticas aún no están a la altura de los métodos diagnósticos tradicionales. Sin embargo, sí que pueden complementar las técnicas utilizadas hasta ahora y auguran un futuro prometedor en el ámbito del diagnóstico clínico y en el sector sanitario.

7. BIBLIOGRAFÍA

American Diabetes Association. (2013). Diagnosis and Classification of Diabetes Mellitus. *Diabetes Care*, *36*(Supplement_1), S67-S74. https://doi.org/10.2337/dc13-S067

CDC. (2024a, mayo 15). *National Diabetes Statistics Report*. https://www.cdc.gov/diabetes/php/data-research/index.html

World Health Organization. (2023, abril 5). *Diabetes*. https://www.who.int/es/news-room/fact-sheets/detail/diabetes

Nurdin, A., Tane, M. M., Tumewu, R. W. T., Suryaningrum, K. M., & Saputri, H. A. (2023). Using Machine Learning for the Prediction of Diabetes with Emphasis on Blood Content. *Procedia Computer Science*, *227*, 990-1001. https://doi.org/10.1016/j.procs.2023.10.608

Instituto Nacional de Estadística. (s. f.). *Tasa de mortalidad atribuida a las enfermedades cardiovasculares, el cáncer, la diabetes o las enfermedades respiratorias crónicas por comunidad autónoma, edad, sexo y periodo*. Recuperado 2 de octubre de 2024, de https://www.ine.es/jaxi/Tabla.htm?tpx=46687

Secinaro, S., Calandra, D., Secinaro, A., Muthurangu, V., & Biancone, P. (2021). The role of artificial intelligence in healthcare: a structured literature review. *BMC Medical Informatics and Decision Making*, *21*(1), 125. https://doi.org/10.1186/s12911-021-01488-9

Davenport, T., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. *Future Healthcare Journal*, *6*(2), 94-98. https://doi.org/10.7861/futurehosp.6-2-94

Siontis, K. C., Noseworthy, P. A., Attia, Z. I., & Friedman, P. A. (2021). Artificial intelligence-enhanced electrocardiography in cardiovascular disease management. *Nature Reviews Cardiology*, *18*(7), 465-478. https://doi.org/10.1038/s41569-020-00503-2

Zhang, K., Liu, X., Shen, J., Li, Z., Sang, Y., Wu, X., Zha, Y., Liang, W., Wang, C., Wang, K., Ye, L., Gao, M., Zhou, Z., Li, L., Wang, J., Yang, Z., Cai, H., Xu, J., Yang, L., … Wang, G. (2020). Clinically Applicable AI System for Accurate Diagnosis, Quantitative Measurements, and Prognosis of COVID-19 Pneumonia Using Computed Tomography. *Cell*, *181*(6), 1423-1433.e11. https://doi.org/10.1016/j.cell.2020.04.045

Luna, J., Peláez, E., Loayza, F. R., Alvarado, R., & Pastor, M. A. (2021). Pattern Recognition of White Matter Lesions Associated With Diabetes Mellitus Type 2. *IFAC-PapersOnLine*, *54*(15), 370-375. https://doi.org/10.1016/j.ifacol.2021.10.284

Zhu, H. (2020). Big Data and Artificial Intelligence Modeling for Drug Discovery. *Annual Review of Pharmacology and Toxicology*, *60*(1), 573-589. https://doi.org/10.1146/annurev-pharmtox-010919-023324

khan, M., Kumar Singh, B., & Nirala, N. (2023). Expert diagnostic system for detection of hypertension and diabetes mellitus using discrete wavelet decomposition of photoplethysmogram signal and machine learning technique. *Medicine in Novel Technology and Devices*, *19*, 100251. https://doi.org/10.1016/j.medntd.2023.100251

Mehedi Hassan, Md., Mollick, S., & Yasmin, F. (2022). An unsupervised cluster-based feature grouping model for early diabetes detection. *Healthcare Analytics*, *2*, 100112. https://doi.org/10.1016/j.health.2022.100112

CDC. (2024b, septiembre 27). *Behavioral Risk Factor Surveillance System*. https://health.gov/healthypeople/objectives-and-data/data-sources-and-methods/data-sources/behavioral-risk-factor-surveillance-system-brfss

United States Department of Health and Human Services. (s. f.). *Behavioral Risk Factor Surveillance System (BRFSS)*. Recuperado 2 de octubre de 2024, de https://health.gov/healthypeople/objectives-and-data/data-sources-and-methods/data-sources/behavioral-risk-factor-surveillance-system-brfss

Xie, Z., Nikolayeva, O., Luo, J., & Li, D. (2019). Building Risk Prediction Models for Type 2 Diabetes Using Machine Learning Techniques. *Preventing Chronic Disease*, *16*, 190109. https://doi.org/10.5888/pcd16.190109

Teboul, A. (2021). *Diabetes Health Indicators Dataset*. https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset

Berrar, D. (2019). Cross-Validation. En *Encyclopedia of Bioinformatics and Computational Biology* (pp. 542-545). Elsevier. https://doi.org/10.1016/B978-0-12-809633-8.20349-X

Walach, J., Filzmoser, P., & Hron, K. (2018). *Data Normalization and Scaling: Consequences for the Analysis in Omics Sciences* (pp. 165-196). https://doi.org/10.1016/bs.coac.2018.06.004

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, 144-152. https://doi.org/10.1145/130385.130401

Mammone, A., Turchi, M., & Cristianini, N. (2009). Support vector machines. *WIREs Computational Statistics*, *1*(3), 283-289. https://doi.org/10.1002/wics.49

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Schauberger, G., Klug, S. J., & Berger, M. (2024). Random forests for the analysis of matched case–control studies. *BMC Bioinformatics*, *25*(1), 253. https://doi.org/10.1186/s12859-024-05877-5

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random Forests. En *Ensemble Machine Learning* (pp. 157-175). Springer New York. https://doi.org/10.1007/978-1-4419-9326-7_5

Zhang, Z. (2016). Naïve Bayes classification in R. *Annals of Translational Medicine*, *4*(12), 241-241. https://doi.org/10.21037/atm.2016.03.38

Wang, S.-C. (2003). Artificial Neural Network. En *Interdisciplinary Computing in Java Programming* (pp. 81-100). Springer US. https://doi.org/10.1007/978-1-4615-0377-4_5

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92-108. https://doi.org/10.1016/j.neucom.2022.06.111

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536. https://doi.org/10.1038/323533a0

Ting, K. M. (2011). Confusion Matrix. En *Encyclopedia of Machine Learning* (pp. 209-209). Springer US. https://doi.org/10.1007/978-0-387-30164-8_157

Petrie, J. R., Guzik, T. J., & Touyz, R. M. (2018). Diabetes, Hypertension, and Cardiovascular Disease: Clinical Insights and Vascular Mechanisms. *Canadian Journal of Cardiology*, *34*(5), 575-584. https://doi.org/10.1016/j.cjca.2017.12.005

Rosengren, A. (2018). Cardiovascular disease in diabetes type 2: current concepts. *Journal of Internal Medicine*, *284*(3), 240-253. https://doi.org/10.1111/joim.12804

Kalagotla, S. K., Gangashetty, S. V., & Giridhar, K. (2021). A novel stacking technique for prediction of diabetes. *Computers in Biology and Medicine*, *135*, 104554. https://doi.org/10.1016/j.compbiomed.2021.104554

8. ANEXOS

Anexo 1. Declaración responsable del uso de IA
https://drive.google.com/file/d/1Nr7qR3CTMmB42WrqG9JWmuGR2EPTMwEE/view?usp=share_link

Anexo 2. URL de acceso a los datos utilizados
https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset

Anexo 3. Codificación de las variables incluidas en la base de datos. Fuente: elaboración propia.

Variable	Descripción
Diabetes_binary	0 = No diabetes, 1 = Diabetes/Prediabetes
HighBP	0 = Presión arterial normal, 1 = Presión arterial alta
HighChol	0 = Colesterol normal, 1 = Colesterol alto
CholCheck	0 = No ha revisado su colesterol en los últimos 5 años, 1 = Si ha revisado su colesterol en los últimos 5 años
Smoker	0 = No fumador, 1 = Fumador
Stroke	0 = No ha tenido un accidente cerebrovascular, 1 = Si ha tenido un accidente cerebrovascular
HeartDiseaseorAttack	0 = No ha tenido un infarto, 1 = Si ha tenido un infarto
PhysActivity	0 = No ha realizado ejercico en los últimos 30 días, 1 = Sí ha realizado ejercico en los últimos 30 días
Fruits	0 = No consume fruta 1 o más veces al día, 1 = Si consume fruta 1 o más veces al día
Veggies	0 = No consume verduras 1 o más veces al día, 1 = Si consume verduras 1 o más veces al día
HvyAlcoholConsump	0 = No tiene un consumo de alcohol elevado, 1 = Si tiene un consumo de alcohol elevado
AnyHealthcare	0 = No tiene seguro de salud, 1 = Si tiene seguro de salud
NoDocbcCost	0 = No ha pospuesto una cita médica por su coste, 1 = Si ha pospuesto una cita médica por su coste
GenHlth	1 = Excelente, 2 = Muy buena, 3 = Buena, 4 = Regular, 5 = Pobre
DiffWalk	0 = No tiene dificultad, 1 = Sí tiene dificultad
Sex	0 = Mujer, 1 = Hombre
Age	1 = 18-24, 2 = 25-29, 3 = 30-34, 4 = 35-39, 5 = 40-44, 6 = 45-49, 7 = 50-54, 8 = 55-59, 9 = 60-64, 10 = 65-69, 11 = 70-74, 12 = 75-79, 13 = +80
Education	1 = Sin educación, 2 = Primaria, 3 = Secundaria, 4 = Graduado secundaria, 5 = Universidad, 6 = Graduado Universidad
Income	1 = -10000$, 2 = -15000$, 3 = -20000$, 4 = -25000$, 5 = -35000$, 6 = -50000$, 7 = -75000$, 8 = +75000$

22302823_tfm_VascoOliveiraCarapinheiro.pdf

ORIGINALITY REPORT

3%	3%	1%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	hdl.handle.net Internet Source	1%
2	repositorio.unapiquitos.edu.pe Internet Source	1%
3	bibliotecadigital.udea.edu.co Internet Source	1%
4	e-spacio.uned.es Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		