Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Análisis y Visualización de Datos Masivos/ Visual Analytics and Big Data

PrediDia: Un Enfoque Predictivo para la Evaluación de la Diabetes

Trabajo fin de estudio presentado por:	Andrea Campillo Piqueras
Tipo de trabajo:	Comparativa de soluciones
Director/a:	Henry Eduardo Baquero Vega
Fecha:	10/07/2024

Resumen

En el presente trabajo fin de máster se compara distintas técnicas de la ciencia de datos, machine learning y deep learning con el objetivo de predecir si un individuo padece diabetes mellitus tipo 2, a partir de características antropomórficas y socioculturales, sin tener en cuenta datos clínicos, normalmente con un coste alto por individuo, reduciéndose con ello los costes sanitarios y pudiendo aplicarse en ámbitos sociales con difícil acceso al sistema de salud, como pudiera ser zonas rurales o regiones del tercer mundo.

Basado en metodologías de machine learning, se ha determinado el mejor modelo posible capaz de predecir la enfermedad objeto de este trabajo. Para ello se han obtenido diez conjuntos de datos, dos de ellos utilizados de forma intensiva, a partir de encuestas masivas realizadas en Estados Unidos por los "Centros de Control de Prevención de Enfermedades". Después se realiza un estudio estadístico, determinando los atributos más relevantes, seleccionando las técnicas de balanceo e hiperparámetros óptimos para los distintos algoritmos de aprendizaje supervisado, consiguiendo los mejores modelos; procediendo a la selección, basado en métricas objetivas, del que mejor cumple los objetivos. Además, se han incluido en la comparativa un modelo basado en redes neuronales artificiales densas y otro basado en el ensamblaje de los mejores algoritmos.

Se demostrará en este trabajo, en el que se obtendrán modelos basados en machine learning y deep learning con capacidades de predicción superior al 80%, hacen viable un triage temprano de la enfermedad, sin usar parámetros clínicos.

Por todo ello, este trabajo podría ser disruptivo en el ámbito sanitario, ahorrando tiempo y costes. Pudiéndose aplicar de forma práctica cómo primer triage en la detección de la enfermedad, además de servir de guía metodológica en proyectos de la misma índole.

Palabras claves: Datos, deep learning, diabetes, machine learning, predicción.

Abstract

In this final master's work, different techniques from data science, machine learning and deep learning are compared with the aim of predicting whether an individual suffers from type 2 diabetes mellitus, based on anthropomorphic and sociocultural characteristics, without considering clinical data, thereby reducing health costs and being able to be applied in social areas with difficult access to the health system.

Following a methodology based on machine learning, the best possible model capable of predicting the disease object of this work has been determined. To this end, ten datasets of data have been obtained, two of them used intensively, from massive surveys carried out in the United States by the "Centers for Disease Control and Prevention". After conducting a statistical study, determining the most relevant attributes, selecting the optimal balancing techniques and hyperparameters for the different supervised learning algorithms, obtaining the best models by proceeding to the selection, based on objective metrics, of the one that best adapts to reaching the objectives. In addition to a model based on dense artificial neural networks, a model based on an ensemble of the best algorithms has also been included in the comparison.

As will be demonstrated in this work, in which models based on machine learning and deep learning will be obtained with prediction capabilities greater than 80%, they make feasible an early triage of the disease, without using clinical parameters.

For all these reasons, this work could be disruptive in the healthcare field, saving time and costs. Focus on potential for practical use as a first triage in the detection of the disease, in addition to serving as a methodological guide in projects of the same nature.

Keywords: Data, deep learning, diabetes, machine learning, prediction.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

1.1. Motivación

La diabetes se considera una epidemia a nivel global, la cual encuentra en un ascenso vertiginoso. Este trabajo se centrará en el estudio de la diabetes mellitus tipo 2, mucho más frecuente que el resto y asociada a hábitos en el estilo de vida. Tal y como afirman numerosos estudios, el estilo de vida actual está directamente relacionado con el mayor número de personas que desarrollan la enfermedad cada día (Petersmann et al., 2019). La preocupación es alarmante no sólo por el padecer diabetes sino también por las consecuencias y las complicaciones asociadas (Garmendia-Lorena et al., 2021). Generándose un gran coste desde el punto de vista del sistema sanitario, actualmente congestionado y sobrepasado.

Numerosos estudios afirman que la solución para reducir este problema sería una detección temprana de la diabetes, ahorrando tiempo, dinero y esfuerzos al sistema sanitario (Narayan et al., 2011). En este momento, el lector podría pensar en que la solución ideal radica en realizar cribados masivos a la población. Sin embargo, el tiempo y coste invertido en diagnosticar la diabetes mediante las pruebas clínicas actuales a gran escala resulta inviable.

Es por ello, que entra en juego la idea de aplicar las técnicas de ciencia de datos e inteligencia artificial para paliar el problema, surgiendo la idea de la predicción de la diabetes mediante estas técnicas con un menor tiempo e inversión monetaria (Sánchez Rosado & Díez Parra, 2022), cómo ya ha ocurrido en otros ámbitos o incluso con otras enfermedades como la tuberculosis, la malaria o el dengue consiguiendo resultados similares a los métodos tradicionales (Ruibal-Tavares et al., 2023).

Existen estudios basados en machine learning, incluyendo deep learning, con resultados prometedores. Sin embargo, los datasets utilizados en la mayoría tienen pocos registros, están sesgados y no están actualizados. Además, hasta ahora gran parte de los datasets contienen datos recogidos a través de pruebas de laboratorio, algo ilógico pues se estaría invirtiendo el tiempo y dinero que se quiere evitar, o incluso a veces estos datasets incluyen datos de la prueba realizada para diagnóstico clínico de la diabetes (Wee et al., 2024).

Por todo ello, un nuevo estudio, como es el caso de este trabajo, que obtenga datasets fiables y actualizados, basados en atributos socioculturales y antropomórficos, excluyendo datos

clínicos, así como la implementación de modelos predictivos basados en técnicas de ciencia de datos, machine learning y deep learning, están más que justificados.

1.2. Planteamiento del trabajo

Por todo lo comentado en puntos anteriores, se propone realizar la predicción de la diabetes mellitus tipo 2 mediante conjuntos de datos que no impliquen pruebas clínicas. Es decir, es primordial obtener un dataset a partir de datos socioculturales, demográficos y antropomórficos, así como otros, obtenidos con encuestas simples, con preguntas que cualquier individuo pueda contestar fácilmente, sin la realización de pruebas clínicas. Además, para paliar las deficiencias actuales en los conjuntos de datos utilizados para la diabetes el dataset deberá estar actualizado y contener un suficiente número de registros.

Posteriormente, utilizando este nuevo dataset se comparan los resultados de algoritmos de machine learning y deep learning, pero a diferencia de otros autores, se propone la comparativa de los algoritmos de machine learning supervisados más utilizados y acordes al tipo de atributo objetivo (discreto), incluyendo en la comparativa una red neuronal artificial.

Cabe destacar que, a diferencia de muchos de los trabajos relacionados, en el presente trabajo no sólo se comparan los algoritmos, sino que también se buscará de forma efectiva el mejor modelo de cada algoritmo, incluyendo la búsqueda de mejores hiperparámetros, así como distintos tipos de entrenamientos.

El objetivo final es conseguir el mejor modelo, entrenado mediante los datasets obtenido sin incluir pruebas clínicas, consiguiendo métricas similares o superiores a las obtenidas por otros autores a partir de conjuntos de datos que incluyen pruebas clínicas.

Este trabajo aporta un avance significativo en el diagnóstico temprano de la diabetes mellitus tipo 2. Consiguiendo un primer diagnóstico de la enfermedad bastante fiable, sin necesidad de invertir la cantidad de dinero y tiempo, pudiendo invertir estos recursos en mejorar otras fases de la enfermedad u otros aspectos del sistema sanitario. Propiciando una disminución de la congestión del sistema sanitario y de aplicación en ambientes sociales donde el recurso sanitario es de difícil acceso, donde actualmente es impensable realizar diagnósticos tempranos de las enfermedades.

1.3. Estructura del trabajo

En primer lugar, se introduce el contexto del problema y el estado del arte, donde se aborda estudios relacionados con este trabajo, proporcionando las bases actuales sobre el tema a tratar.

Después, se plantean los objetivos tanto los generales como específicos. Para pasar a explicar la metodología seguida para la realización de este trabajo. Luego se describe el marco normativo.

Seguidamente se plantea el desarrollo del trabajo, este apartado contiene nueve subapartados: inicialmente se justifican la selección de los atributos a utilizar, y la obtención de los datasets finales, incluyendo en este punto la limpieza y preprocesamiento de estos. A continuación, se realiza un análisis exploratorio de los datos, con el objetivo de comprender los datos con los que se va a trabajar, en este punto se incluye la selección final de los atributos. Posteriormente se prepara el conjunto de datos para aplicarlos a algoritmos de machine y deep learning, con su correspondiente entrenamiento y a continuación, se explica cómo se han seleccionado los distintos hiperparámetros de los algoritmos. Exponiéndose cada algoritmo en específico y los resultados obtenidos, para finalmente compararlos entre ellos y obtener el mejor modelo posible.

Por último, se indica dónde encontrar el código fuente y datos utilizados, conclusiones, limitaciones y trabajos futuros.

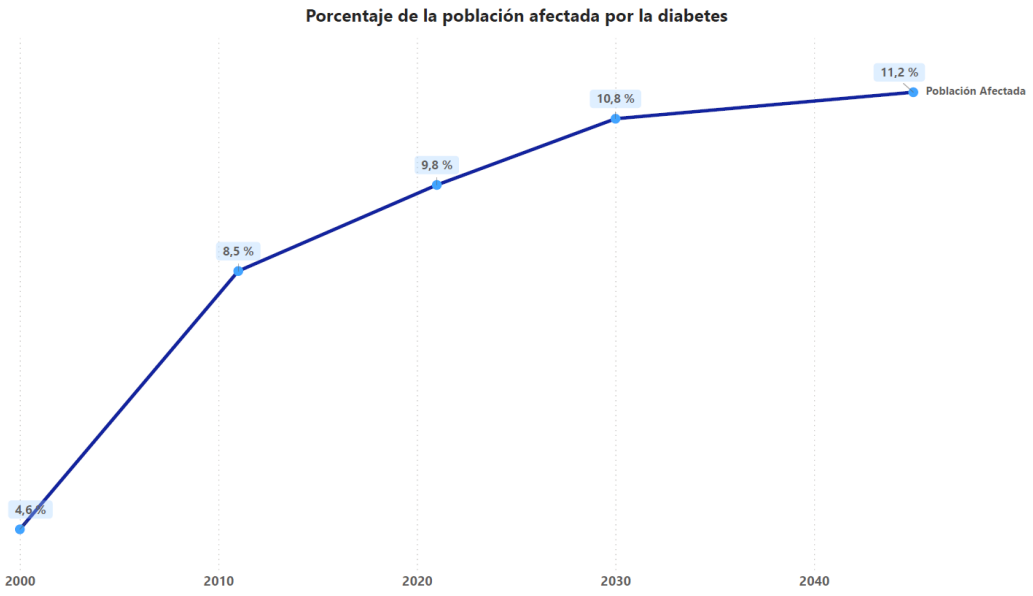
2. Contexto y estado del arte

2.1.Contexto del problema

La diabetes es una enfermedad crónica caracterizada por presentar niveles elevados de glucosa en sangre (Harreiter & Roden, 2023). El aumento de la glucosa en sangre puede estar producido principalmente por dos motivos: debido a un proceso autoinmunitario que destruye las células responsables de producir la insulina, llamadas células beta pancreáticas o por una resistencia endógena a la insulina (Lovic et al., 2020).

Respecto a la prevalencia de la enfermedad, la diabetes se considera una epidemia global. Según la federación Internacional de la Diabetes, en el año 2000 un 4,6% de la población padecía la enfermedad, en 2011 aumentó hasta un 8,5%, en 2021 se incrementó hasta un 9,8%. Según esta tendencia, se prevé que en 2030 un 10,8% sufran la enfermedad y en 2045 aumente hasta un 11,2%, considerándose sin duda, una situación sanitaria alarmante (Figura 1) (International Diabetes Federation, 2021) .

Figura 1 . *Evolución de la población afectada por la diabetes en el tiempo.*



Fuente: Elaboración propia, inspirada en (International Diabetes Federation, 2021) .

Existen tres tipos principales de diabetes: tipo 1, tipo 2 y gestacional. Este trabajo se centra en la diabetes mellitus tipo 2. La elección de este requisito se centra en que la diabetes de tipo 1 está más influenciada por factores genéticos y autoinmunes, sin embargo, la de tipo 2 se caracteriza en gran parte por elementos antropomórficos, hábitos y socio culturales, además

de tener una frecuencia mucho mayor, un 90-95% frente al 5-10% del tipo 1. La diabetes gestacional sólo ocurre durante el embarazo (Petersmann et al., 2019).

La diabetes mellitus tipo 2 puede propiciar algunas consecuencias como: afecciones oculares en algunos casos incluso ceguera, distintas neuropatías, enfermedades renales, complicaciones cardiovasculares, también puede presentar comas diabéticos, aumenta la predisposición de sufrir ciertas infecciones. Además, suele presentar afecciones médicas relacionadas, se citan las más comunes: hipertensión, trastornos lipídicos y sobrepeso u obesidad (Garmendia-Lorena et al., 2021).

Existen tres métodos para el diagnóstico de la enfermedad, la prueba de la tolerancia oral a la glucosa (oGTT), detección de la glucosa en ayunas, medición de la hemoglobina glucosilada (HbA1c) y pruebas de glucosa ocasional sin ayuno (Harreiter & Roden, 2023). Estas pruebas se complementan y en la mayoría de los casos se repiten, ya sea para la confirmación del diagnóstico u observar la evolución de la enfermedad. Si bien no es el tema principal de este trabajo conviene tener en cuenta que estas pruebas clínicas conllevan un proceso largo, costoso y con requerimientos nutricionales específicos en un periodo de tiempo, así como un control de calidad minucioso (Petersmann et al., 2019). Además, de necesitar un personal especializado y laboratorios adecuados para el análisis de las pruebas realizadas, algo inaccesible en algunas regiones (Narayan et al., 2011)

Ahora bien, en este punto, surge la siguiente cuestión ¿Podría el análisis masivo de datos y el big data ser útiles en el diagnóstico precoz de la enfermedad objeto de estudio en este trabajo? Existe una extensa cantidad de datos biomédicos que utilizados junto con las herramientas adecuadas propician una mayor compresión de las enfermedades, una mejora significativa en el diagnóstico y la posibilidad de ofrecer tratamientos personalizados a cada paciente (Manzini et al., 2022). Para procesar y extraer conocimiento de estos datos se requieren tecnologías que superan a lo etiquetado como estadística tradicional, por ejemplo, algoritmos de clustering, aprendizaje automático, procesamiento de datos en tiempo real, entre otros (Manzini et al., 2022).

2.2. Estado del arte

Tras definir la problemática actual de la diabetes se procederá a realizar una revisión bibliográfica de la situación que rodea a la enfermedad y cómo se han aplicado las disciplinas

relacionadas con la ciencia de datos en el ámbito. Para ello, en primer lugar, se identificarán medios actuales para el diagnóstico precoz de la diabetes. A continuación, se ilustrará el uso de la ciencia de datos e inteligencia artificial en el ámbito sanitario general, profundizando en el diagnóstico. Posteriormente, se explicará como estas tecnologías se han aplicado a la diabetes, profundizando en el diagnóstico temprano de la enfermedad. En este punto se describen estudios donde se compararán algoritmos basados en machine learning y deep learning. Después, se expondrán investigaciones sobre la diabetes, basados en conjunto de datos obtenidos desde la misma organización que el presente trabajo, pero que en nuestro caso son de elaboración propia y actualizados a las últimas versiones que ha sido posible. Luego, se detallarán los conjuntos de datos más utilizados para el estudio y diagnóstico de la diabetes. Finalmente, se obtendrán conclusiones y se justificará la importancia de este trabajo.

2.2.1. Medios actuales para el diagnóstico de la diabetes

Como se ha explicado anteriormente, se ha demostrado como la detección temprana de la diabetes mellitus tipo 2, entre otros beneficios, evita ciertas complicaciones relacionadas con la enfermedad y, por tanto, se reducen sus costes (Narayan et al., 2011). Así como el retraso de la enfermedad y propicia un mayor control de la condición (Deepa & Sivasamy, 2023).

Es cierto que para el diagnóstico temprano pueden existir puntuaciones de riesgo. Estas puntuaciones son medidas basada en la estadística tradicional. A través de preguntas se devuelve una puntuación que se asocia con la probabilidad de desarrollar una enfermedad. Aunque su uso como primer paso en el diagnóstico en algunas enfermedades está muy extendido, en el caso de la diabetes no ocurre esto. Hay numerosas encuestas con puntaciones de riesgo asociadas para esta enfermedad, su uso está muy limitado debido a la falta de políticas de concienciación que fomenten su aplicación (Noble et al., 2011). Algunos cuestionarios relevantes para la enfermedad de estudio son por ejemplo QDiabetesR Risk Calculator, Cambridge Diabetes Risk Score, Leicester Risk Assessment (Vizzuett Montoya & López-García, 2021). Entre ellos destaca la prueba de Finish Diabetes Risk Score (FINDRISC), este cuestionario tiene ocho preguntas relacionadas con la edad, el índice de masa corporal (IMC), perímetro abdominal, ejercicio físico, dieta, la toma de antidepresivos, historial hiperglucémico, historial familiar (Vizzuett Montoya & López-García, 2021). Sin embargo, aunque este tipo de cuestionarios podrían ser una estrategia para el diagnóstico temprano de

la enfermedad, tienen un gran inconveniente: requieren de revisiones y actualizaciones manuales, ya que se basan en la estadística tradicional. Proporcionando resultados poco precisos si los comparamos con el uso de herramientas de ciencia de datos, big data e inteligencia artificial.

2.2.2. Ciencia de datos, Big Data e Inteligencia Artificial en la medicina

A continuación, se explicarán el uso de estas novedosas herramientas en la medicina, con un mayor énfasis en el diagnóstico y en concreto en la diabetes.

El potencial del big data y disciplinas relacionadas es inmenso, pero esta posibilidad solo es plausible si se cuenta con procesos eficientes para analizar y convertir grandes volúmenes de datos a información útil (Pastorino et al., 2019). Se ha demostrado como la introducción del big data en el ámbito sanitario trae consigo numerosos beneficios, entre ellos destaca la reducción de tiempo empleado en la obtención de resultados, hasta un 25% (Chipia Lobo, 2020). También, se percibe como las distintas técnicas de la ciencia de datos están transformando las bases de datos medicas tradicionales hacia otras, integradoras de un mayor conocimiento y evidencias (Subrahmanya et al., 2021).

Se debe mencionar que en el sector sanitario la ciencia de datos abarca desde datos biológicos y clínicos hasta ambientales y de estilo de vida (Pastorino et al., 2019). El análisis de datos en medicina se puede aplicar a diversos aspectos como son la vigilancia de enfermedades, procesamiento de imágenes, recopilación de datos en tiempo real, mejora de gestión de datos y recursos, en la salud mental, en epidemiologia, farmacovigilancia, análisis de parámetros fisiológicos y en la prevención de fraude (Subrahmanya et al., 2021). Otros enfoques son el diagnóstico temprano, prevención de patologías, identificación de factores de riesgo, predicciones dando como resultado decisiones clínicas más informadas y óptimas (Pastorino et al., 2019).

Un aspecto relevante relacionado es la aplicación de la inteligencia artificial en la medicina. Aunque es cierto que se encuentra en etapas iniciales, se estima que la incorporación de la inteligencia artificial en el campo de la salud presente beneficios considerables, presentando mejoras con un valor aproximado de un 30%-40% y una reducción de costes de un 50% (Sánchez Rosado & Díez Parra, 2022). Actualmente, el sistema sanitario esta congestionado, la falta de profesionales de la salud, junto con la gran carga de trabajo provocan una situación insostenible. Todo ello da lugar a un mayor número de errores, frecuentemente en etapas de

diagnóstico y tratamiento. Se calcula que en estados unidos el cometer errores es la tercera causa de muerte (Lanzagorta-Ortega et al., 2023). La aplicación de la inteligencia artificial disminuiría drásticamente el número de errores (Ruibal-Tavares et al., 2023). Además, esta disciplina científica podría automatizar tareas repetitivas con una mayor consistencia y velocidad que un ser humano, permitiendo que los profesionales de la salud se centren en tareas de mayor importancia (Ruibal-Tavares et al., 2023) (Lanzagorta-Ortega et al., 2023).

2.2.2.1. Ciencia de datos, Big Data e Inteligencia Artificial en el diagnóstico

Es conveniente señalar como la inteligencia artificial ya se ha aplicado en regiones en situación de pobreza. Principalmente en el diagnóstico y cribado enfermedades contagiosas como por ejemplo la tuberculosis, malaria o dengue. Obteniendo resultados equiparables con otros tipos de diagnósticos tradicionales (Ruibal-Tavares et al., 2023).

Uno de los niveles fundamentales, esencial para esta investigación, en los que se aplica la disciplina de ciencia de datos e inteligencia artificial es el diagnóstico de enfermedades. Existen numerosas herramientas y softwares que han mejorado el diagnóstico de algunas enfermedades, por ejemplo, MYCIN/MYCIN II, CASNET, PIP, AL/RHEUM. Además, Face2Gene es un programa de reconocimiento facial capaz de proporcionar indicios de padecer una enfermedad rara. Por supuesto, cabe mencionar la aplicación de la inteligencia artificial en la interpretación de imágenes (Ávila-Tomás et al., 2020).

A continuación, y según la información extraída de varios autores, se citan distintas técnicas aplicadas en el diagnóstico de algunas patologías más preocupantes en el panorama actual. En primer lugar, para el diagnóstico temprano del alzheimer se han estudiado Support Vector Machine (SVM) y Redes Neuronales Convolucionales (CNN), presentando esta última un mayor rendimiento, al igual que ocurre con el cáncer y la tuberculosis. Respecto a las enfermedades cardiacas son los algoritmos de Support Vector Machine (SVM) y modelos de ensemble los que obtuvieron precisiones mayores (Kumar et al., 2023). Se puede concluir que cuando hay imágenes el algoritmo más apropiado para su diagnóstico son las Redes Neuronales Convolucionales (CNN).

Según Adler-Milstein.J y colaboradores, defienden que no se está extrayendo el máximo potencial en el diagnóstico mediante inteligencia artificial, puesto que simplemente se etiqueta la enfermedad, como diagnóstico final y no es un proceso orientativo como proponen (Adler-Milstein et al., 2021).

2.2.3. Ciencia de datos, Big Data e Inteligencia Artificial en la diabetes

Aunque este trabajo se centre en obtener el mejor rendimiento posible para el diagnóstico de la diabetes, cabe mencionar algunos aspectos donde la inteligencia artificial y disciplinas relacionadas tienen gran importancia para la enfermedad objeto de estudio. Algunos de ellos son: el monitoreo continuo a través de aplicaciones móviles, detección de la retinopatía diabética, toma de decisión en cuanto al tratamiento, recomendaciones personalizadas sobre el tratamiento y aún más importante para este proyecto, el diagnóstico de padecer la enfermedad, incluso su predicción precoz (Ellahham, 2020).

2.2.3.1. Estudios relacionados con el diagnóstico de la diabetes

A continuación, se exponen algunas investigaciones sobre la aplicación de distintas técnicas de la ciencia de datos en el diagnóstico de la diabetes, comparando los distintos algoritmos utilizados en cada uno de ellos y las características de cada estudio.

Hui Yang y colaboradores, detectaron la importancia del diagnóstico temprano de la diabetes a través de la inteligencia artificial, para ello realizaron un estudio donde se seleccionaron los atributos más relacionados con la enfermedad mediante las técnicas de Información Mutua, Análisis de Varianza (ANOVA), Impureza de Gini. Obtuvieron los datos a través del Registro Médico Electrónico (EMR) de la comisión municipal de salud de Luzhou en China. Las características seleccionadas se componen de datos demográficos, signos vitales y clínicos. Los autores obtuvieron 3 subconjuntos de características. Mediante los algoritmos XGBoost, Random Forest y Regresión Logística, compararon los resultados obtenidos con cada subconjunto y entre los algoritmos. Alcanzaron mejores resultados con el subgrupo con los atributos edad, IMC, relación cintura altura, glucosa en ayunas, presión sistólica media, glucosa en la orina. Respecto al algoritmo fue XGboost, el que presentó resultados superiores. Cada modelo se evaluó por medio de medidas como exactitud, precisión, recuperación, tasa de falsos positivos y F1, curva ROC y AUC y además se aplicó la validación cruzada (Yang et al., 2021).

Hang Lai y colaboradores, realizaron una investigación similar, compararon otros algoritmos y utilizaron un conjunto de datos de tan solo doscientos pacientes de un centro médico de Bangladesh, el cual contenía dieciséis atributos en total tanto clínicos como demográficos. Analizaron el rendimiento presentado por los algoritmos Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbor (KNN) y Árbol de Decisión C4.5. Entre ellos, fue el algoritmo

C4.5 el que logró mejores resultados. Se compararon en términos de validación cruzada N-fold (N=10), precisión, exactitud, recuperación, medida F1 y exactitud. C4.5 obtuvo valores del 72% en la precisión, 74% en la recuperación, y un 72% en la medida F1 (Faruque et al., 2019). En otro estudio se abordó el diagnóstico de la diabetes mediante un dataset canadiense con 13.309 registros obtenidos del Canadian Primary Care Sentinel Surveillance Network (CPCSSN). El conjunto de datos contenía datos tanto clínicos como demográficos. Los autores aplicaron el método de ganancia de la información, para explorar el atributo que aportaba más información. Como resultado se obtuvo la glucemia en ayunas, seguido por las lipoproteínas de baja densidad y el IMC. Además, como existía un desequilibrio entre las clases, realizaron el ajuste de umbrales y la ponderación de clases. Todos los resultados los compararon con el dataset Pima Indian Diabetes Dataset (PIDD), será explicado más adelante, y emplearon la validación cruzada. Finalmente, llegaron a la conclusión que la Gradient Boosting Machines (GBM) obtuvo los mejores resultados (AUC-ROC 84,7%, clasificación errónea 18,9%, sensibilidad 71,6%, especificidad 83,7%). Seguido de la Regresión Logística (AUC-ROC 84,0%, clasificación errónea 19,6%, sensibilidad 73,4%, especificidad 82,3%), en cambio, las técnicas de Random Forest y el Árbol de Decisión tuvieron peores resultados (Lai et al., 2019) .

En este caso, Iparraguirre-Villanueva y colaboradores, optaron por utilizar el dataset PIDD. A partir del conjunto de datos original obtuvieron dos conjuntos de datos más. Uno de ellos para solventar el desequilibrio de clases aplicando la técnica SMOTE y otro para disminuir la dimensionalidad mediante PCA. El estudio compara cada técnica con estos tres conjuntos de datos. Realizaron, como en la mayoría de las investigaciones, la división en datos de entrenamiento y prueba (80%-20%) y la técnica de validación cruzada (K-10 fold). Finalmente consiguieron los siguientes valores: K-NN logró los mejores resultados, en concreto, con el dataset SMOTE 79,6%, con el de PCA 44,4% y con el original 55,6%. El segundo fue Bernoulli Naive Bayes, con la técnica SMOTE obtuvo 77,2%, con PCA 59,70% y con el dataset original 66,2%. En cambio, el Árbol de Decisión, Support Vector Machine (SVM), Regresión Logística con el conjunto de datos SMOTE tan solo consiguieron valores de 63%, 71,7% y 72,7% respectivamente (Iparraguirre-Villanueva et al., 2023).

De nuevo, Isfafuzzaman Tasin y colaboradores, en este estudio los autores usaron el dataset PIDD junto a uno propio. Ambos conjuntos de datos se componen de los mismos atributos. En este caso, para solventar el desequilibrio entre clases utilizaron las técnicas SMOTE y ADASYN.

Para la selección de atributos usaron la técnica de Información Mutua. Además, aplicaron técnicas como SHAP y LIME para comprender el funcionamiento de los algoritmos. Los investigadores compararon Random Forest, Support Vector Machine (SVM), Regresión Logística, AdaBoost, XGBoost, Voting Classifier y Bagging Classifier. Se destaca el algoritmo XGBoost, ya que logró los mejores resultados, respecto a la precisión un 81%, puntuación F1 0,81, y respecto a AUC 0,84. Mientras que el Árbol de Decisión tuvo la menor efectividad (Tasin et al., 2022).

Vasudha Rani y colaboradores, comparan un modelo de perceptrón multicapa con algoritmos de machine learning como Random Forest y Rregresión Logística. El conjunto de datos utilizado contiene dieciocho atributos relacionados con la diabetes todos ellos no invasivos pero complicados de afirmar por los pacientes sin un seguimiento médico, por ejemplo, la poliuria, la paresia parcial, candidiasis entre otros. Tampoco se especifica el número de registros con el que cuentan. Sin embargo, este estudio, es interesante porque compara algoritmos de machine learning con un algoritmo de deep learning. Además, se seleccionan los atributos mediante la combinación de tres métodos distintos, selección univariada, importancia de características y matriz de correlación con mapas de calor. Los autores, finalmente, llegan a la conclusión que el deep learning es adecuado para el diagnóstico temprano de la diabetes y necesita menos atributos que los algoritmos de machine learning. Aunque obtuvieron resultados más precisos con los modelos de machine learning. Se presentan a continuación los valores obtenidos, el Random Forest presento una exactitud de 0,98076, la Regresión Logística de 0,97115 y el perceptrón multicapa con 100 interacción logró 0,779221 y con 300 logró 0,813853 (Vasudha et al., 2021). Se ha de comentar que los resultados son algo confusos. En el artículo se menciona como la técnica de deep learning es superior al resto, pero en las tablas que contienen las métricas se observa cómo es Random Forest el algoritmo con mejores resultados.

Existen diversos estudios del deep learning en el diagnóstico de la diabetes, en general estos estudios presentan resultados óptimos y tienen un gran potencial. A continuación, se presentan estudios relacionados relevantes.

Islam Ayón y colaboradores, utilizaron el dataset PIDD, para estudiar la precisión del diagnóstico temprano de la diabetes mediante una Red Neuronal Profunda (DNN). En esta red neuronal la capa oculta tenía cuatro capas, cada capa contaba con doce, dieciséis, dieciséis y

catorce neuronas. La capa de entrada estaba formada por ocho neuronas y la capa de salida por una. Los autores realizaron dos entrenamientos de validación cruzada, uno de cinco pliegues y el otro de diez. Según los autores, el resultado obtenido (con validación cruzada quíntuple) era superior a otros estudios de machine learning. En concreto los resultados fueron: exactitud 98,04%, sensibilidad 98,80%, especificidad 96,64% y puntuación F1 99%, AUC-ROC 98% (Islam Ayon & Milon Islam, 2019).

Los autores María Teresa García-Ordás y colaboradores, mediante una perspectiva totalmente profunda y utilizando el dataset PIDD, estudiaron la predicción de la enfermedad. En este caso, para el desequilibrio de clases utilizaron la técnica de codificadores automáticos variacionales (VAE) y para el aumento de características codificadores dispersos (SAE). Se dividen los datos en 90% entrenamiento y 10% prueba. Para la predicción emplearon las Redes Neuronales Convolucionales (CNN) y Multilayer Perceptron (MLP). Es interesante como además de abordar el tema desde solo el deep learning también demostraron que utilizando y entrenando dos técnicas de manera conjunta de deep learning como son SAE y CNN o SAE y MLP aumenta la precisión. Lograron una precisión del 92,31% y 85,71% respectivamente (García-Ordás et al., 2021).

En este estudio, Zeyu Zhang y colaboradores, investigan la propuesta de una red neuronal profunda de retropropagación (BPNN) para el diagnóstico no invasivo de la diabetes. Los autores compararon tres datasets, PIDD, CDC BRFSS2015 y BIT Mesra. Los datos fueron equilibrados mediante una técnica de submuestreo y se escalaron mediante la estandarización. El conjunto de datos fue dividido en 20% prueba y 80% entrenamiento y aplicaron la validación cruzada quíntuple. Utilizando PIDD se logró un 89,81% de precisión, una sensibilidad de 89,29% y una especificidad de 90,38%. En cambio, con el conjunto de datos CDC BRFSS2015 obtuvieron resultados deficientes 0,7549; 0,7997; 0,7112. Finalmente, con BIT Mesra lograron los mejores 0,9528; 1,0; 0,9219 (Zhang et al., 2024).

Tabla 1. *Tabla comparativa del estado del arte.*

Artículo	Algoritmo	Dataset	Precisión	Especifidad	Recall	Exactitud	F1
Hui Yang et al.	XGBoost	Propio. EMR de Luzhou en China	0.51	0.85	0.74	0.73	0.64
Faisal Faruque et al.	C4.5	Propio. 200 pacientes Bangladesh	0.72	-	0.74	0.73	0.72
Hang Lai et al.	GBM	CPCSSN	-	0.84	0.85	-	-
Orlando Iparraguirre Villanueva et al.	K-NN	PIDD	0.57	-	0.80	0.72	0.67
Isfafuzzaman Tasin el al.	XGBoost	PIDD + Propio	0.81	-	0.81	0.81	0.81
Vasudha Rani et al.	RF	Propio	1.00	-	0.97	0.99	0.98
Safial Islam Ayón & Md. Milon Islam	DNN	PIDD	-	0.97	-	0.98	0.99
Maria Teresa García-Ordás et al.	CNN+SAE	-	0.92	-	-	-	-
Zeyu Zhang et al.	BPNN	PIDD	0.9	0.9	-	-	-
Zeyu Zhang et al.	BPNN	CDC BRFSS2015	0.76	0.71	-	-	-
Zeyu Zhang et al.	BPNN	BIT Mesra	0.95	0.92	-	-	-

Fuente: Elaboración propia, basada en la bibliografía del estado del arte.

2.2.3.2. Estudios relacionados con el origen de datos utilizado

Respecto a los datos en lo que basa el dataset utilizado en este trabajo provienen del Sistema de Vigilancia de Factores de Riesgo del Comportamiento (BRFSS), un sistema de encuestas telefónicas sobre la salud y temas relacionados de Estados Unidos. En el ámbito mundial se le considera puntero por realizar encuestas médicas anuales, siguiendo patrones establecidos, considerándose un sistema de vigilancia e incluyendo más de cuatrocientas mil personas. Siendo financiadas por muchos de Centros para el Control y Prevención de Enfermedades (CDC) (Centers for Disease Control and Prevention, 2014).

Se han realizado algunos estudios de diversas enfermedades y temas de salud utilizando los datos proporcionados por estas encuestas, los cuales se encuentran disponibles sin restricciones, incluyendo gran cantidad de atributos del encuestado, del tipo antropomórficos, socioculturales, hábitos, no invasivos, así como de diversas patologías, pero sin optimizaciones orientadas a la ciencia de datos, podríamos decir que los datos se encuentran en su forma original y sin procesar. En cuanto a la diabetes existen algunas investigaciones fundamentadas en estas encuestas. Por ejemplo, el uso de dataset de 2014-2015 de acceso libre y sin restricciones obtenido de estas encuestas, para realizar predicciones de la diabetes mediante algoritmos de inteligencia artificial (Hussein Mohamed et al., 2024) (Ullah et al., 2022). Otro estudio es el de María L. Alval que mediante las encuestas realizadas por BRFSS en 2016-2017, estudia la relación entre la diabetes y la depresión. Es destacable que oficialmente, sólo hay un 3% de la población que esté diagnosticada de las dos afecciones. Sin embargo, la autora obtuvo una correlación del 17% entre ambas, lo que sugiere un infradiagnóstico. Además, el estudio recomienda, que en el diagnóstico de la diabetes debiera haber preguntas relacionadas con la salud mental, nivel educativo, vida familiar etcétera (Alva,

2020). En otro estudio, Taiwo P.Adesoba y Clare C. Brown, investigaron el aumento de la diabetes en personas con un IMC saludable entre los años 2015-2020 por medio de las encuestas de BRFSS de esos años. Demostraron, como las razas distintas de la blanca, con una edad de más de 45 y en mayor medida las mujeres han aumentado la prevalencia de sufrir esta enfermedad. En cambio, para las personas con sobrepeso u obesas la prevalencia se mantenía como en otros años (Adesoba & Brown, 2023).

2.2.4. Conjuntos de datos existentes para el diagnóstico de la diabetes

Cabe mencionar otros datasets utilizados en el estudio de la diabetes. Pero antes, se debe mencionar, los métodos no invasivos se refieren a métodos que no entran físicamente dentro de las cavidades corporales, pero son pruebas médicas.

En primer lugar, destaca el dataset PIDD, es el más utilizado para estudiar la diabetes, algo impactante ya que está sumamente sesgado. Se puede descargar gratuitamente desde Kaggle (Pima Indians Diabetes Database. 2016) PIDD se compone de setecientos registros de mujeres de la india Pima. Tiene nueve atributos número de embarazos, glucosa obtenida por la OGT, Tensión arterial, grosor de la piel, insulina, IMC, función pedigrí de la diabetes, edad y salida (presencia o ausencia de diabetes). Por tanto, está compuesto por características clínicas y demográficas.

Un conjunto de datos también muy utilizado es Diabetes 130-US hospitals for years 1999-2008 Dataset, está compuesto por datos de diez años (1999-2008) de 130 hospitales de Estados Unidos. Los datos incluyen 101.766 registros y 47 atributos que contienen datos demográficos y clínicos como pruebas de laboratorios, días de ingreso, etcétera (Clore et al., 2014).

Otra fuente de datos utilizada son los de la National Health and Nutrition Examination Survey (NHANES) realizada en Estados Unidos a unas cinco mil personas. La encuesta incluye datos relacionados con la salud, se compone de dos partes, una entrevista con preguntas socioeconómicas, demográficas, relacionadas con la salud y un examen médico, incluyendo exámenes tanto pruebas fisiológicas y como de laboratorio (Centers for Disease Control and Prevention, 2023c).

Existe otra perspectiva, obtener los datos del EMR y Registro de Salud Electrónico (EHR). Consiste en conseguir datos extrayéndolos de la historia clínica de un paciente (Ambinder, 2005). La diferencia entre EMR y EHR es que el uso de EMR es restringido y es sólo utilizado por profesionales de la salud para el diagnóstico y tratamiento. En cambio, los EHR pueden

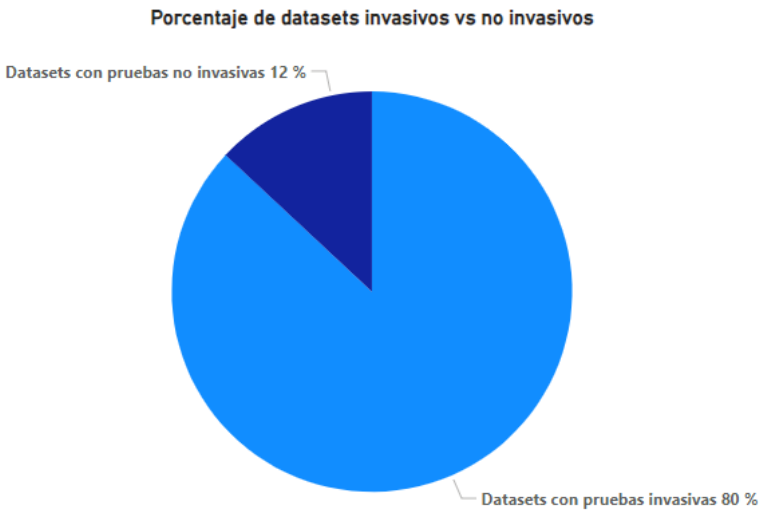
ser compartido y usados ampliamente por diversos profesionales. La Canadian Primary Care Sentinel Surveillance Network (CPCSSN) se ocupa de obtener diversos datos de los EMR de la región. Cuenta con hasta 1,8 millones de registros, con datos de todo tipo relacionados con el ámbito de la salud, demográficos, médicos, pruebas de laboratorio entre otros (Canadian Primary Care Sentinel Surveillance Network, n.d).

2.3.Conclusiones

La ciencia de datos, el big data y la inteligencia artificial pueden ayudar a implementar un sistema de diagnóstico temprano de la diabetes mellitus tipo 2, considerada como una epidemia mundial, liberando al sistema sanitario, hoy día saturado, de cargas rutinarias, reduciendo tiempo y costes.

Hasta la fecha la mayoría de datasets empleados para predecir la diabetes presentan diversas deficiencias están sesgados, contienen pocos registros y aún más importante para este trabajo, incluyen datos de pruebas clínicas o laboratorio. Parece obvio que la inclusión de estas valoraciones clínicas se podría considerar redundante, el resultado de la prueba clínica en sí misma podría ser suficiente para considerar un primer diagnóstico sin necesidad de utilizar otras tecnologías. Por todo ello, parece incoherente incluir una vez invertido tiempo, dinero y esfuerzo, este tipo de pruebas clínicas o laboratorio en los datasets (Wee et al., 2024).

Figura 2. *Porcentaje de datasets invasivos y no invasivos para la diabetes.*



Fuente: Elaboración propia, inspirada en el artículo (Wee et al., 2024).

Tan solo un 12% de los conjuntos de datos para la diabetes utilizan métodos no invasivos (Figura 2) (Wee et al., 2024). Por ello es requisito disponer de un dataset actualizado, masivo

y con características antropomórficas, demográficas y socioculturales de los individuos como alternativa a la mayoría de los utilizados en trabajos anteriores, en muchos casos sesgados, desactualizados e incluyendo datos clínicos y de laboratorio. Sería ideal que el dataset además tuviera los atributos necesarios y en formato que se pudieran utilizar en tecnologías como la ciencia de datos, big data e inteligencia artificial, y constatar si realmente son útiles para un diagnóstico precoz de a diabetes mellitus tipo 2, ahorrando esfuerzo, costes y tiempo al sistema sanitario.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

Desarrollar el mejor modelo posible de predicción basado en la ciencia de datos, machine learning y deep learning, para el diagnóstico temprano de la diabetes mellitus tipo 2 basado en un dataset de elaboración propia, incluyendo atributos demográficos, socioculturales y antropomórficos, sin tener en cuenta pruebas clínicas, invasivas o de laboratorio.

De forma colateral se pondrá en práctica una metodología que permita obtener un dataset actualizado y de calidad, realizando un estudio preliminar de los datos obtenidos, obtener los mejores atributos, descartándose los que no aporten una mayor exactitud en la predicción, determinar los mejores hiperparámetros de los algoritmos de aprendizaje supervisados más relevantes, para finalmente comparar utilizando métricas objetivas y finalmente seleccionar el mejor modelo posible.

Sin olvidar que el objetivo final es poder ayudar a los profesionales de la salud. Logrando una reducción de tiempo, costes y esfuerzos al sector sanitario.

3.2. Objetivos específicos

Los objetivos específicos de este trabajo son:

- Crear un dataset con datos basado en características antropomórficas, demográficas y socioculturales, sin datos de pruebas clínicas a partir de encuestas públicas y masivas.
- Preprocesar y optimizar el dataset, mediante un script de Linux y con la creación de campos calculados.
- Estudiar de forma general los datos obtenidos.
- Comparar e identificar los atributos asociados a la diabetes mellitus tipo 2 más relevantes mediante correlaciones y la técnica “Feature_importances” del algoritmo Random Forest.
- Obtener los mejores hiperparámetros de cada uno de los algoritmos de aprendizaje supervisado utilizados mediante el framework GridSearchCV.
- Comparar los modelos obtenidos mediante métricas objetivas e identificar el mejor modelo.
- Implementar un modelo que combinen los mejores mediante técnicas de ensemble.

- Obtener conclusiones sobre la posibilidad de un diagnóstico temprano de la diabetes mellitus tipo 2 sin pruebas de laboratorio con el fin de apoyar al sistema sanitario y proporcionarle un ahorro de recursos.

3.3. Metodología del trabajo

En este apartado se detallará el entorno tecnológico utilizado, así como la metodología empleada para llevar a cabo este trabajo.

3.3.1. Entorno

Se detalla los entornos utilizados, haciendo diferencia entre el entorno de programación y otros genéricos para el desarrollo como proyecto informático.

Entorno genérico:

Se detalla los entornos utilizados, haciendo diferencia entre el entorno de programación y otros genéricos para el desarrollo como proyecto informático.

Entorno genérico:

- Sistemas operativos:
  - Windows 11
  - Linux: Ubuntu 22.04.3 como aplicación integrada en Windows 11
- Editores:
  - Notepad++ v8.4.2
  - Windoword para MS365
- Gráficos:
  - Power BI
  - PowerPoint para MS365
- Hojas de cálculo:
  - Excel para MS365

Entorno programación:

- Sistemas de versionado:
  - Git: 2.44.0.windows.1

    - Git Hub: Repositorio en nube (Campillo Piqueras, 2024a). Se puede acceder directamente al repositorio (https://github.com/AndreaCampillo/TFM_PrediDia)

- Lenguajes de programación:
    - Python 3.11.9
    - Comandos bash de Linux: Incluida en Ubuntu 22.04.3
- Entorno Big Data:
    - Anaconda 2.6.0
    - Jupyter Notebook 6.5.4
    - Google Colaboratory (Colab)
- Librerías: Para consultar las versiones de las librerías utilizadas en el trabajo se ejecuta el código Python que se muestra en la figura 3:
    - Pandas: 2.1.4
    - scikit-learn: 1.2.2
    - imblearn: 0.11.0
    - seaborn: 0.12.2
    - matplotlib.pyplot: 3.8.0
    - tensorflow: 2.16.1

Figura 3. *Código Python sobre las versiones de las librerías utilizadas.*

```
import pandas as pd
import sklearn as sc
import imblearn as im
import seaborn as sns
import matplotlib as plt
import tensorflow as tf
```

```
print('Ver. pandas: ',pd.__version__)
print('Ver. sklearn: ',sc.__version__)
print('Ver. imblearn: ',im.__version__)
print('Ver. seaborn: ',sns.__version__)
print('Ver. matplotlib: ',plt.__version__)
print('Ver. tensorflow: ',tf.__version__)
```

```
Ver. pandas:  2.1.4
Ver. sklearn:  1.2.2
Ver. imblearn:  0.11.0
Ver. seaborn:  0.12.2
Ver. matplotlib:  3.8.0
Ver. tensorflow:  2.16.1
```

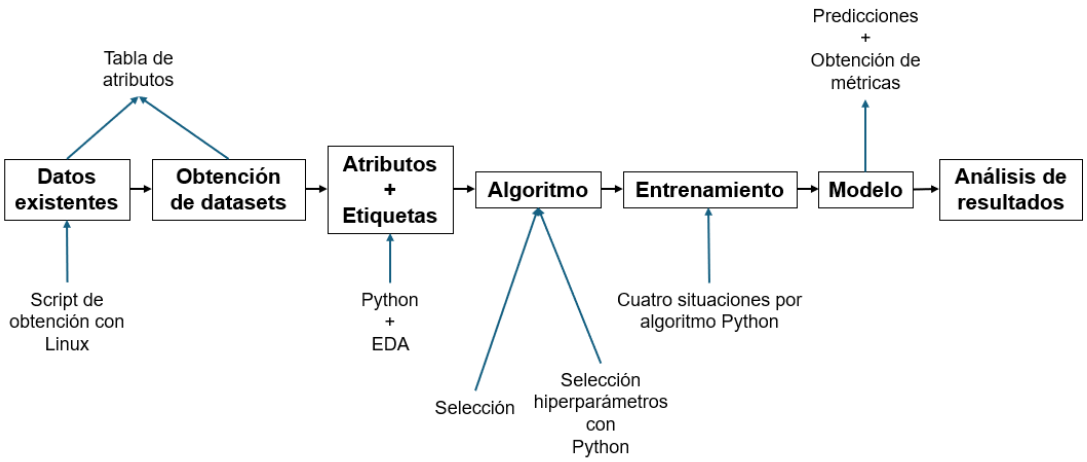
Fuente: Elaboración propia.

3.3.2. Esquema general de la metodología empleada

A continuación, se describen a grandes rasgos los pasos seguidos durante este trabajo. Posteriormente se desglosará en mayor profundidad el procedimiento empleado, además se acompaña de una imagen (Figura 4) resumen para facilitar el entendimiento del lector:

- Generación de los datasets.
    - Justificación de los atributos elegidos.
    - Búsqueda, recolección de los datos.
    - Obtención del dataset.
- Selección del algoritmo e hiperparámetros.
- Entrenamiento del algoritmo.
- Análisis del modelo.
- Si los resultados son óptimos despliegue en producción.
- Si los resultados no son óptimos o pueden ser mejorados se analiza como perfeccionar los siguientes aspectos:
    - Datos.
    - Cambio de hiperparámetros del algoritmo.
    - Cambio de algoritmo.
- Regreso al punto de partida.

Figura 4. *Esquema general de la metodología empleada.*



Fuente: Elaboración propia.

3.3.3. Explicación en profundidad de la metodología empleada

Se destaca que en todo momento se han empleado técnicas para evitar el underfitting y overfitting. A lo largo de la descripción de esta metodología se harán referencia, entre ellas se distinguen: el ajuste de hiperparámetros, la ampliación del número de registros, técnicas de balanceo, reducción del número de características y el uso de un conjunto de pruebas independiente del proceso de entrenamiento y validación, desde el que se han obtenido las métricas finales de cada modelo.

3.3.3.1. Obtención de los dataset

La obtención de los datasets utilizados en este proyecto consta de cuatro fases. Se parte dos encuestas (2021 y 2022) las cuales contienen datos socioculturales, antropomórficos y de salud, incluida la diabetes, almacenado en ficheros ASCII. Las fases para conseguir los datasets son:

- Estudio de las preguntas de las encuestas:
    - Selección de las preguntas relacionadas con rasgos socioculturales y antropomórficos asociados a la diabetes.
    - Justificación de las características seleccionadas.
    - Estudio de cómo se almacenan estos atributos en el fichero ASCII.
- Generación en formato csv de los conjuntos de datos utilizados posteriormente por los algoritmos en formato csv:
    - Obtención de los atributos desde el fichero ASCII.
    - Generación de campos calculados más adecuados para los algoritmos.
    - Transformación y limpieza de los atributos, incluyendo la eliminación de registros sin información en sus atributos: blancos, caracteres no esperados y no aportados por el encuestado.
- Generación de tabla resumen de los atributos incluidos en los conjuntos de datos.
- Descripción de los datasets obtenidos.

3.3.3.2. Análisis Exploratorio de Datos (EDA) y exclusión de atributos

- Se realiza el Análisis Exploratorio de Datos (EDA) para comprender los datos y poder seleccionar los algoritmos óptimos de acuerdo con las características de los datos.

- La exclusión de atributos cuyo uso no influye en el resultado final de la predicción, pero pueden afectar negativamente en el final del rendimiento del modelo.

  Esta fase se ha dividido en se realiza en tres partes:

  - Descarte de atributos con correlaciones elevadas entre ellos.
  - Descarte de atributos con escasa correlación con la variable objetivo.
  - Estudio de la importancia de los atributos con la técnica "Feature_importances" del algoritmo Random Forest.

3.3.3.3. Elección de los algoritmos

Tras el estudio de los conjuntos de datos, se llega a la conclusión que los algoritmos más adecuados para realizar predicciones sobre ellos son de tipo supervisado. Los algoritmos supervisados son adecuados cuando los datos de entrenamiento tienen etiquetas asociadas, es decir son conocidas. En cambio, en los algoritmos de tipo no supervisados los datos no tienen etiquetas e intenta buscar patrones sin contar con el resultado final.

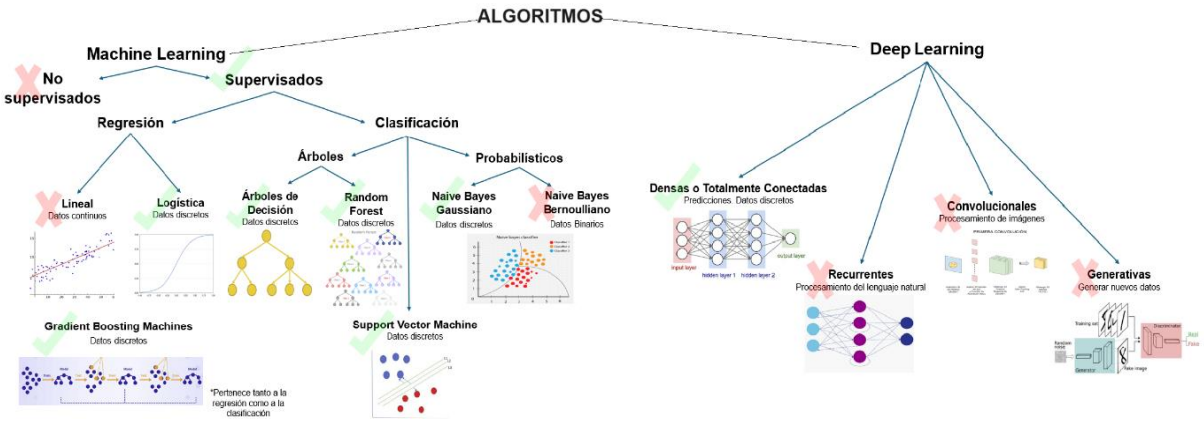
Como nuestros datos cuentan con etiquetas asociadas (si el individuo es diabético o no), las cuales se intentarán predecir, es un claro ejemplo de uso de algoritmos supervisados. Sin embargo, dentro de los algoritmos supervisados existen algoritmos que funcionan de forma óptima con datos continuos, discretos o ambos. En nuestro caso los valores de salida son discretos (tener o no tener diabetes), por lo que el estudio se centrará en algoritmos que sean capaces de trabajar con ese tipo de datos. A continuación, se presentan los algoritmos adecuados de acuerdo con la naturaleza del conjunto de datos utilizado, además se presenta un esquema (Figura 5):

- Algoritmos de regresión:
  - **Logística**.
- Algoritmos de clasificación:
  - Basados en árboles:
    - **Árboles de decisión**.
    - **Conjunto de árboles (Radom Forest)**
  - **Support Vector Machine**.
  - Probabilísticos:
    - **Naive Bayes Gausiano**.

- **Gradient Boosting Machines** (Pertenece tanto a algoritmos de clasificación como de regresión)

- Redes Neuronales Artificiales.
    - **Densas o totalmente conectadas.**

Figura 5 . *Elección de algoritmos.*



Fuente: Elaboración propia.

3.3.3.4. Selección de hiperparámetros de cada algoritmo

Para obtener los hiperparámetros más adecuados para cada uno de los algoritmos se ha hecho uso del framework GridSearchCV. Sin embargo, por el alto coste computacional, en algunos algoritmos no ha sido posible utilizar GridSearchCV y se han empleado los hiperparámetros que se han considerado más adecuados. Un ejemplo de ello es el algoritmo Support Vector Machine y sus distintos kernels, donde finalmente se ha decido usar Radial Basis Function (RBF).

3.3.3.5. Entrenamiento de los algoritmos

Para entrenar cada algoritmo seleccionado, incluyendo los mejores hiperparámetros, se han seguido los siguientes pasos:

- Importación de librerías.
- Obtención F1 Score 2021 con 25 características, en este primer paso no se excluyen las determinadas menos importantes mediante la técnica Feature_importances de Random Forest, con el objetivo de comparar que la eliminación de las características no afecta al rendimiento del modelo final.
    - Partición de los datos 2021 incluyendo 25 características en:

        - Entrenamiento (60%)
        - Validación (20%)
        - Prueba (20%)
    - Entrenamiento
    - Obtención de F1 Score con los datos de validación.
- Entrenamiento datos 2021 con 21 características, se excluyen las determinadas menos importantes mediante la técnica Feature_importances de Random Forest.
    - Partición de los datos 2021 incluyendo las 21 características en:
        - Entrenamiento (60%)
        - Validación (20%)
        - Prueba (20%)
    - Obtención de F1 Score con datos de validación y comparación con el F1 Score obtenido con las 25 características.
    - Obtención de métricas con los datos de prueba.
- Entrenamiento datos 2021 con 21 características, se excluyen las determinadas como menos importantes mediante la técnica Feature_importances de Random Forest, y aplicando la técnica SMOTE, para balancear el conjunto de datos.
    - Partición de los datos 2021 con las 21 características en:
        - Entrenamiento (60%)
        - Validación (20%)
        - Prueba (20%)
    - Aplicación de SMOTE en los datos de entrenamiento.
    - Obtención de F1 Score con datos de validación.
    - Obtención de métricas con los datos de prueba.
- Entrenamiento datos 2021_22 con 18 características se excluyen las determinadas menos importantes mediante la técnica Feature_importances de Random Forest y las no existentes en los datos de 2022 (Presión arterial, colesterol y consumo de frutas y vegetales)
    - Partición de los datos 2021_22 con las 21 características en:
        - Entrenamiento (60%)
        - Validación (20%)
        - Prueba (20%)

- Obtención de F1 Score con datos de validación.
- Obtención de métricas con los datos de prueba.

3.3.3.6. Estudio de cada modelo y métricas

Para analizar cada modelo y compararlos entre ellos se calculas las siguientes métricas a partir de los datos de prueba, independientes de los datos de entrenamiento y de validación:

- F1 score:
- Precisión (Precision)
- Exactitud (Accuracy)
- Especificidad (Specificity)
- AUC-ROC

3.3.3.7. Comparación de modelos

A partir de las métricas se genera una tabla donde se pueden consultar de forma unificada facilitándose la comparación entre los modelos.

3.3.3.8. Datos nuevos y predicciones

Se proporciona al modelo óptimo un conjunto de datos no utilizados durante el resto del proyecto, resultante de aquellos individuos que no indicaron si eran diabéticos y estudio de las predicciones.

3.3.4. Técnicas para evitar el desbalanceo

Como se verá a lo largo del proyecto los datos se encuentran desbalanceados (15% diabéticos y 85% no diabéticos) por ello se han empleado tres técnicas de balanceo. Todas ellas han sido comparadas en cada algoritmo, son:

- Uso del hiperparámetro class_weight=balanced, siempre que el algoritmo lo permita.
- Creación de datos sintéticos mediante SMOTE
- Incremento del número de registros mediante el uso del dataset 20021_22. Aunque se prescinda de alguna característica importante para la diabetes mellitus tipo 2 como el colesterol alto, la presión arterial o el consumo de frutas y vegetales, por no haberse incluido en las encuestas del 2022.

4. Marco normativo

4.1. DATOS

El presente trabajo ha mitigado los riesgos que pudieran estar afectados por el uso de datos personales y debido a la naturaleza del objetivo "Predecir si un paciente es diabético" podrían ser considerados datos de salud. Sin embargo, los **datos** están **anonimizados** durante todo proceso del tratamiento de la información, siendo imposible determinar un individuo en particular. Por todo ello, se estima **no necesario tomar medidas respecto al marco normativo** que afecta al tratamiento de datos personales, tales como la "Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD)" y el "Reglamento general de protección de datos (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (RGPD)"

En este contexto es oportuno describir las medidas tomadas por los Centros para el Control y la Prevención de Enfermedades (CDC) de Estados Unidos, organismos encargados del tratamiento de la información resultante de las encuestas dentro del ámbito del Sistema de Vigilancia de Factores de Riesgo del Comportamiento (BRFSS) (Centers for Disease Control and Prevention, 2023d):

- **Número de teléfonos aleatorios**: Las encuestas se realizan utilizando técnicas de Marcación Aleatoria de Dígitos (RDD), tanto en teléfonos fijos como en teléfonos móviles (Centers for Disease Control and Prevention, 2022).
- Del estudio de los datos y tras análisis de la información proporcionada en el sitio web de BRFSS (Centers for Disease Control and Prevention, 2024a) se deducen las siguientes acciones:
    - **Eliminación de identificadores directos**: Eliminación de cualquier información que pueda identificar directamente a un individuo, como nombre, dirección, número de teléfono y fecha de nacimiento.
    - **Codificación**: Las respuestas se almacenan con códigos que no identifican a las personas.
    - **Agregación**: Inclusión de datos de múltiples encuestados para que sea difícil identificar a los individuos.

- **Grupos amplios**: Publicación de grandes grupos, evitándose los específicos por áreas geográficas.

4.2. LICENCIAMIENTO

Se ha determinado que todas las partes del trabajo esté bajo el paraguas de una licencia tipo MIT (La Licencia MIT. 2024) con derechos de autor (c) 2024 para Andrea Campillo Piqueras. Siendo el resultado final del Trabajo fin Máster Universitario en Análisis y Visualización de Datos Masivos/ Visual Analytics and Big Data cursado en la Universidad Internacional de la Rioja - España (UNIR), titulado “PrediDia: Un Enfoque Predicitivo para la Evaluación de la Diabetes”.

Se concede permiso, sin cargo, a cualquier persona que obtenga una copia del presente software y de los archivos de documentación asociados (el "Software"), para tratar el Software sin restricción alguna, incluyendo, sin limitación, los derechos a usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y/o vender copias del Software, y a permitir que las personas a las que se les proporciona el Software lo hagan, sujeto a las siguientes condiciones: El aviso de derechos de autor anterior y este aviso de permiso deberán incluirse en todas las copias o partes sustanciales del software o documentación.

El software y documentación se proporciona "tal cual", sin garantía de ningún tipo, expresa o implícita, incluidas, pero sin limitarse a las garantías de comercialización, idoneidad para un fin particular y no infracción. En ningún caso los autores o titulares de los derechos de autor serán responsables de ninguna reclamación, daño o responsabilidad, ya sea por contrato, agravio o de otro modo, que surja de, o esté relacionado con, el software o su uso u otros acuerdos relacionados con el software.

Este trabajo se basa en datos del Sistema de Vigilancia de Factores de Riesgo Conductuales (BRFSS) de los Centros para el Control y la Prevención de Enfermedades (CDC) de Estados Unidos. La información se ha utilizado sin manipular su contenido original y no se pretende que este proyecto represente o esté avalado por el CDC. Los datos y materiales producidos por las agencias federales son generalmente de dominio público y pueden reproducirse sin permiso. Sin embargo, el CDC solicita que cualquier material publicado derivado de los datos reconozca al BRFSS del CDC como fuente original.

5. Desarrollo específico de la contribución

5.1.JUSTIFICACIÓN DE LA ELECCIÓN DE LOS ATRIBUTOS

A continuación, se justificará la importancia de los atributos escogidos para la creación del dataset, explicándose cada uno de ellos y su relación con la diabetes mellitus tipo 2. En los anexos "A. Estructura de los atributos de la encuesta" y "B. Valores de los atributos de la encuesta" se puede consultar de forma resumida la información más relevante de los atributos seleccionados.

Índice de Masa Corporal (IMC). Se calcula dividiendo el peso al cuadro entre la altura. En un estudio llevado a cabo por Belkis Martínez-Vasallo y colaboradores, se demostró que un 42% de personas que padecían diabetes mellitus tipo 2 eran obesas (IMC > 30), un 38% tenían sobrepeso (25 <= IMC <= 29,9), y tan sólo un 20% tenían normo peso (18,5<= IMC <= 24,9). Esta correlación se debe a que un aumento de peso corporal produce una disminución de sensibilidad a la insulina (Martínez-Vasallo et al., 2021).

Actividad física. Se conoce extensamente la asociación entre el ejercicio y la obesidad. El sedentarismo conduce a una mayor ganancia de peso. Pero más allá de este conocimiento general se ha demostrado como el ejercicio mejora la sensibilidad a la insulina y al control glucémico. Además, el ejercicio mejora la absorción de la glucosa por los músculos y reduce la HbA1c (Blanco Naranjo et al., 2021).

Dieta. Una mayor ingesta calórica provoca un aumento de peso. Pero también se ha demostrado como un menor consumo calórico provoca el uso de glucógeno hepático, disminuyendo la producción de glucosa hepática. Mediante la terapia nutricional la HbA1c disminuye hasta un 2% y mejora el control glicémico, la resistencia a la insulina y la secreción de la insulina (Blanco Naranjo et al., 2021).

Colesterol. Se podría pensar que está indirectamente relacionado con la diabetes. Esta afirmación es cierta, una mala dieta puede dar lugar a esta condición, y el padecer colesterol en muchos casos está relacionado con el sobrepeso. Pero, además, se relaciona directamente ya que la insulina es primordial para regular el control del uso y almacenamiento de grasas. Cuando se padece resistencia a la insulina este control se resiente. Como consecuencia de esta desregularización aumentan las partículas VLDL encargadas de transportar el colesterol en la

sangre. Además, la insulina es crucial para la descomposición de las VLDL, si no actúa correctamente estas partículas se acumulan (Columbié et al., 2016).

Hipertensión arterial. Un descontrol de los niveles de insulina causa estrechamiento de las arterias, provoca el crecimiento del músculo liso en los vasos sanguíneos y desregula el transporte de fluidos e iones, todo ello conlleva a un aumento de la presión arterial, estando ambas condiciones sumamente relacionadas (Martínez-Vasallo et al., 2021).

E**nfermedades coronarias (ECV)**. Comunes en pacientes con diabetes mellitus tipo 2. A nivel global, un 34,8% de los pacientes diabéticos sufren ECV (Dieuzeide et al., 2022). La relación entre ambas enfermedades reside en lo explicado anteriormente, la resistencia a la insulina provoca mayores niveles de tensión arterial y dislipidemia, ambos principales causantes de las ECV. Además, la alteración de los azúcares produce aterosclerosis.

I**ctus**, el 70% de los pacientes que sufren un ictus tenían diabetes mellitus tipo 2 (Fuentes et al., 2021). Un ictus es una enfermedad vascular y la diabetes afecta a los vasos sanguíneos.

Enfermedades renales. Un 20% de personas que padecen diabetes mellitus tipo 2 también sufren de nefropatía. La nefropatía desarrollada por la diabetes se denomina nefropatía diabética. Estas dos enfermedades están muy relacionadas ya que la hiperglucemia causa daños en los riñones, impidiendo el correcto filtrado de estos. Además, la hipertensión enfermedad que frecuentemente coexiste con la diabetes, también causa daños renales (Orellana-Suarez et al., 2024).

Enfermedades pulmonares. Aunque no se comprende exactamente, se ha demostrado la relación entre ambas. Algunas enfermedades pulmonares estudiadas junto con la diabetes son el asma y la enfermedad pulmonar obstructiva (EPOC). En el asma, se ha observado como la diabetes causa una hiperactividad en las vías respiratorias y además ambas enfermedades producen inflamación. Se ha encontrado una mayor prevalencia de asma en pacientes con diabetes mellitus tipo 2. En cuanto al EPOC, se ha observado un aumento en los pacientes con diabetes, aunque no se ha discernido exactamente la causa clínica se cree que es debido a la elevada y continua inflamación que provoca EPOC puede dar lugar a desarrollar diabetes (Khateeb et al., 2019).

Enfermedades reumatoides, como la gota, la artritis y el lupus. En un estudio de 827 pacientes se observó que el 31% de pacientes con alguna enfermedad reumática padecía diabetes

mellitus tipo 2. Aunque, hasta ahora no se sabe cuál es exactamente la relación de ambas enfermedades se han deducido dos posibilidades: La inflamación producida por las enfermedades reumatoides afecte a las células betapancráticas, las cuales producen la insulina. La segunda es que ambas enfermedades están relacionadas con el sistema inmunológico (Fiallos et al., 2019).

Deterioro cognitivo. Se ha observado que la diabetes mellitus tipo 2 debido a la hiperglucemia puede causar daños en el cerebro. Además, mediante resonancia magnética se ha comprobado anormalidades en la materia blanca del cerebro. Asimismo, se debe comentar como el estrés oxidativo afecta también a las neuronas. Por ello, muchos de los individuos que padecen diabetes presentan una reducción de su concentración y de su nivel de memoria (Matar-Khalil & Rubio-Sandoval, 2021).

Enfermedades oculares. Debido a los altos niveles de glucosa los pacientes con diabetes son más propensos a desarrollar problemas de visión. Entre ellas la más común es la retinopatía diabética, pero también se incluyen el glaucoma, la degeneración macular relacionada con la edad, las cataratas, cambios en la refracción y en la sensibilidad al contraste (Khan et al., 2017).

Salud mental y la diabetes. En concreto, el 20% de personas que sufren diabetes tienen depresión. La depresión aumenta el riesgo de sufrir ansiedad y viceversa (Woon et al., 2020). Aunque no se conoce exactamente el vínculo entre las enfermedades psicológicas y la diabetes, muchas teorías defienden que es debido a que la hiperglucemia y los daños en los vasos sanguíneos causen daños cerebrales dando lugar a la depresión y viceversa. Además, del estrés que puede suponer una enfermedad crónica como es la diabetes (Healthline, 2022).

Calidad de vida y el estado de salud general. Según el estudio la diabetes tiene un impacto directo en la salud general del paciente, tanto en el ámbito físico como psicológico. Por ello, es importante tener en cuenta este punto en estudio de la diabetes (Gálvez Galán et al., 2021).

Respecto a los atributos de hábitos diarios, sin incluir el ejercicio y la alimentación puesto que se han explicado anteriormente, toman relevancia el tabaquismo y el alcohol.

Tabaquismo. Aumenta el riesgo de sufrir diabetes mellitus tipo 2, incluso en fumadores pasivos. El consumo de tabaco da lugar a el desarrollo de complicaciones relacionadas con la diabetes, como las explicadas anteriormente y en concreto destaca, el aumento de desarrollar

nefropatías (Soto, 2017). Actualmente, está de moda el uso de cigarrillos electrónicos. Sin embargo, estos cigarrillos contienen pequeñas dosis de nicotina lo que propicia la resistencia a la insulina y a largo plazo diabetes mellitus tipo 2 (Song & Zou, 2017).

Consumo de alcohol. Se ha observado en numerosos estudios como aumenta el riesgo de desarrollar diabetes mellitus tipo 2. Además, se ha de tener en cuenta que el consumo de dos cervezas o dos copas de vino diarias aumenta de forma exponencial el desarrollo de la enfermedad. El alcohol también produce una desregulación del control de la glucosa (Veleiro, 2023) ya que estas sustancias causan un daño al hígado, órgano clave en este control.

A continuación, se explica la relación de la diabetes con características demográficas y finalmente socioculturales.

Edad, **género** y **etnia.** Actualmente no existe una explicación, pero se ha demostrado que la diabetes mellitus tipo 2 es hasta un 10% más común en mujeres que en hombres (Martínez-Vasallo et al., 2021). La CDC ha indicado que las personas mayores de 45 años presentan un mayor riesgo de desarrollar diabetes mellitus tipo 2 (Centers for Disease Control and Prevention, 2024b). Posiblemente esta situación sea debida a que con la edad algunos órganos como el páncreas empeoran su funcionalidad, disminuye la sensibilidad a la insulina y por supuesto se tiende a ser menos activas físicamente. Por otro lado, parece que la **etnia** es otro factor influyente en cuanto al predisposición a desarrollar diabetes. Las personas hispanas, asiáticas y nativos americanos son más propensas que la raza blanca (Martínez Candela, 2015). Posiblemente debido a la genética, culturas o niveles económicos distintos.

Factores socioeconómicos. Se ha demostrado como en países con mejor calidad de vida, las personas con un poder adquisitivo mayor son menos propensas a desarrollar diabetes mellitus tipo 2, posiblemente por la capacidad de optar a mejores condiciones médicas y alimentos más saludables. Sin embargo, en países más pobres, se observa lo contrario, las personas con ingresos mayores tienden a llevar un peor estilo de vida y abusar de ciertos alimentos ricos en grasas (Seiglie et al., 202) .

Nivel educativo. En los países ricos un mayor nivel educativo tiende a entender mejor los riesgos para la salud y tener un poder económico mayor. Al contario que, en los países más pobres, ya que una mayor educación propicia en general mayor nivel económico, dando lugar a las consecuencias explicadas anteriormente (Seiglie et al., 202).

Estado civil. Las mujeres viudas presentan mayores complicaciones, seguidas de hombres separados y hombres casados. Todo ello es debido a cambios en el estilo de vida, estrés y apoyo (Kposowa et al., 2021).

Horas de sueño. Se ha demostrado como la falta de sueño desregula algunas hormonas entre ellas las asociadas con el índice glucémico sanguíneo y otras que controlan factores que propician al desarrollo de la diabetes como la presión arterial y el peso. Además, esta falta de descanso altera el estilo de vida y produce cambios en la actividad física (Woods et al., 2023).

Consumo de marihuana. Se ha demostrado como esta sustancia empeora el control glucémico. En personas diabéticas el consumo de esta sustancia se asocia a un mayor riesgo de desarrollar complicaciones relacionadas con la enfermedad, así como una posible interferencia con fármacos para tratar la diabetes (Bermúdez Silva & Romero Zerbo, 2023).

5.2.OBTENCIÓN DE LOS DATASETS

5.2.1. Obtención de los atributos

A continuación, en este apartado se explicará cómo se ha obtenido cada atributo de una forma general, el nombre que tiene dicho atributo en el dataset final y a qué pregunta responde dicho atributo. Para mayor claridad del lector los atributos se han dividido en **tres grupos** dependiendo de la forma que se hayan obtenido, en los anexos “Anexo A. Estructura de los atributos de la encuesta” y “Anexo B. Valores de los atributos” el lector podrá consultar el código de cada atributo, el origen del dato, el concepto de este, los valores según BRFSS, los valores calculados en este trabajo, así como las columnas donde se ubican los campos en el fichero ASCII de origen y su longitud en caracteres:

- **Atributos directo de la encuesta**. Obtenidos directamente de la pregunta de la encuesta.
- **Atributos calculados por BRFSS**. Derivados mediante cálculos de otros respondidos en la encuesta.
- **Atributos calculados en este trabajo**. Con el fin de explorar la optimización de los modelos de machine learning se han generados atributos propios.

5.2.1.1. Atributos directos de la encuesta

A continuación, se detallan los atributos obtenidos directamente de la encuesta, se indicará por cada uno el nombre dado en este trabajo y entre paréntesis su correspondencia con la fuente de datos original. Para más información véase los anexos “Anexo A. Estructura de los atributos de la encuesta” y “Anexo B. Valores de los atributos”.

Stroke (CVDSTRK3), responde a la pregunta si se ha tenido un ictus alguna vez.

NoMedCost (MEDCOST1), responde a la pregunta si en los últimos doce meses ha necesitado ir a un médico y no se lo ha podido permitir.

CogDiff (DECIDE), responde a la pregunta si se encuentran dificultades en la concentración, toma de decisiones o memoria.

Depression (ADDEPEV3), responde a la pregunta si se ha tenido depresión.

WalkDiff (DIFFWALK), responde a la pregunta si se tiene dificultades para andar o subir escaleras.

AnnIncome (INCOME3), responde a la pregunta sobre el salario.

UrologyDz (CHCKDNY2), responde a la pregunta si se han tenido enfermedades urológicas.

ViSíonDiff (BLIND), responde a la pregunta si se han tenido dificultades visuales.

LungDiseases (CHCCOPD3), responde a la pregunta si se han tenido dificultades pulmonares.

MaritalSt (MARITAL), responde a la pregunta sobre el estado civil actual.

LastMedChk (CHECKUP1), responde a la pregunta sobre el tiempo transcurrido desde el último chequeo médico.

Awareness (DIABEDU), responde a la pregunta si se ha atendido algún curso sobre el manejo de la diabetes.

FootIrrita (FEETCHK3), responde a la pregunta a cuantas veces por día, semana, mes y año se ha tenido irritación en los pies.

HighBP (BPHIGH6), responde a la pregunta si alguna vez se ha tenido la tensión arterial alta. Se destaca que este atributo sólo existe en la encuesta de 2021.

HighChol (TOLDHI3), responde a la pregunta si alguna vez se ha tenido el colesterol alto. Se destaca que este atributo sólo existe en la encuesta de 2021.

MarijuanaCon (MARIJAN1), responde a la pregunta sobre cuantos días se ha fumado marihuana o cannabis en los últimos 30 días. Se destaca que este atributo sólo existe en la encuesta de 2022.

SleepHours (SLEPTIM1), responde a la pregunta del número de horas que se duermen. Se destaca que este atributo sólo existe en la encuesta de 2022.

BrDiabetes (DIABETE4), responde a la pregunta si alguna vez se ha tenido diabetes.

5.2.1.2. Atributos calculados por BRFSS

CatBMI (_BMI5CAT), es calculado a partir del atributo de índice de masa corporal (_BMI5). Responde a la pregunta si se tiene un peso normal, sobre peso u se es obeso.

HeartDis (_MICHD), es calculado a partir de los atributos de si se ha tenido un infarto de miocardio (CVDINFR4) y si se ha tenido una enfermedad coronaria (CVDCRHD4). Responde a la pregunta si se ha tenido una enfermedad coronaria o un infarto de miocardio.

PhysExer (_TOTINDA), es calculado a partir del atributo si se ha hecho ejercicio físico durante los últimos 30 días (EXERANY2). Responde a la pregunta si se ha realizado ejercicio físico durante los últimos 30 días.

HealthIns (_HLTHPLN), es calculado a partir del atributo de qué tipo de seguro médico se tiene (PRIMINSR). Responde a la pregunta si se tiene algún seguro médico.

GenHealth (_RFHLTH), es calculado a partir del atributo del nivel de salud consideran que tienen (GENHLTH). Responde a la pregunta sobre cómo es su salud en general.

MentalHlth (_MENT14D), es calculado a partir del atributo de cuántos días se considera que se ha tenido mala salud mental en un plazo de 30 días (MENTHLTH). Responde a la pregunta sobre cuántos días se ha tenido mala salud mental en los últimos 30 días.

PhysHlth (_PHYS14D), es calculado a partir del atributo de cuántos días se considera que se ha tenido mala salud física en un plazo de 30 días (PHYSHLTH). Responde a la pregunta cuántos días se ha tenido mala salud física en los últimos 30 días.

Gender (_SEX), es calculado a partir de los atributos obtenidos directamente de la encuesta como por ejemplo BIRTHSEX, responde a la pregunta cuál fue su sexo al nacer, devolviendo valores si el individuo es sexo femenino o masculino.

AgeRange (_AGE_G), es calculado a partir del atributo AGE, que responde a la pregunta cuál es su edad y utiliza un campo intermedio _IMPAGE, devolviendo a que rangos de edad pertenece el individuo.

EdLevel (_EDUCAG), es calculado a partir del atributo sobre distintos niveles escolares (EDUCA). Responde a la pregunta sobre el nivel de educación.

Asthma (_LTASTH1), es calculado a partir del atributo sobre si se ha tenido asma (ASTHMA3). Responde a la pregunta si se ha tenido asma.

Arthritis (_DRDXAR3), es calculado a partir del atributo sobre si se ha padecido algún tipo de artritis (HAVARTH5). Responde a la pregunta si se ha tenido algún tipo de artritis como reuma, gota o lupus.

SmokerTrad (_SMOKER3), es calculado a partir del atributo resultante de sendas preguntas de la encuesta: Si ha fumado al menos 100 cigarrillos en su vida (SMOKE100) y si fuma cada día (SMOKDAY2).

ECigSmok (_CURECI1), es calculado a partir del atributo con cuanta frecuencia se consume cigarrillos electrónicos u otros similares (ECIGNOW1). Responde a la pregunta si se suele fumar cigarrillos electrónicos.

AlcDrinker (_RFBING5), es calculado a partir de los atributos sobre cuantas veces en los últimos 30 días se ha consumido alcohol (DRNK3GE5) y cuantos días por semana y mes se ha consumido alguna bebida alcohólica durante los últimos 30 días (ALCDAY5). Responde a la pregunta si se suele consumir alcohol.

Race (_RACEGR3), es calculado a partir del atributo _RACE derivado de otros que tiene en cuenta aspectos multirraciales (_MRACE1) y distinción entre distintas etnias latinas (_HISPANC). Responde a la pregunta a la raza que pertenece un individuo.

FruitCons (_FRTLT1A), es calculado a partir del atributo _FRUTSU1, derivado de otros que almacenan la cantidad de frutas consumidas, responde a la pregunta sobre si el individuo come frutas. Se destaca que este atributo sólo existe en la encuesta de 2021.

VegCons (_VEGLT1A), es calculado a partir del atributo _VEGESU1, derivado de otros que almacenan los distintos tipos de vegetales consumidos, responde a la pregunta sobre si el individuo come verduras. Se destaca que este atributo sólo existe en la encuesta de 2021.

5.2.1.3. Atributos calculados durante el desarrollo de este trabajo

A continuación, se describen los atributos calculados durante el desarrollo de este trabajo y no incluidos en los ficheros ASCII de las encuestas. Las razones de la creación de estos campos pueden ser tres:

- Agrupar varios atributos en uno sólo y comprobar de forma analítica si pueden sustituir a los atributos tomados como referencia, incluso mejorar el comportamiento de los algoritmos de machine learning.
- Reducir la complejidad de los atributos de referencia y proporcionar otros que puedan utilizarse de forma más global, sin estar vinculados al país donde se realizaron las encuestas (Estados Unidos).
- Dar distintas opciones al analista en lo relativo al atributo objetivo, ser o no ser diabético, por existir valores intermedios como ser prediabético, debiéndose analizar y determinando la mejor agrupación, con el objetivo de mejorar el comportamiento de los algoritmos de machine learning utilizados en este trabajo.

Año de obtención (Year)

Podrá tener dos valores posibles (2021, 2022) dependiendo el año en el que se realizó la encuesta.

MentalState (Estado Mental)

Agrupa en un solo atributo otros tres relacionados con el estado mental del individuo: Dificultades cognitivas (CogDiff), Salud Mental (MentalHlth) y Depression (Depresión)

Los posibles valores serán los siguientes:

- Si el individuo ha tenido buena salud mental, no ha tenido dificultades cognitivas y no ha tenido depresión tomará el valor 1 (Buena salud mental).
- Si el individuo ha tenido mala salud mental o ha tenido dificultades cognitivas o ha tenido depresión tomará el valor 2 (Mala salud mental).
- En el resto de los casos tomará valor 9 (No sabe, no contesta, dato omitido).

Clase Social (SocClass)

Clasifica el individuo en clase social alta, media o baja tomando como referencia su salario (AnnIncome). La razón de la creación de este campo es doble:

- Simplificar el campo AnnIcome, que según se puede consultar la tabla 4 del "Anexo B. Valores de los atributos" puede tomar hasta doce valores distintos.
- Internacionalizar el atributo por estar muy vinculado a los salarios de Estados Unidos. De esta forma en una futura modificación de la encuesta solo se preguntaría por la clase social del individuo o su salario anual, pudiéndose en este caso personalizar y transformar según los rangos del país de origen del encuestado.

Para determinar la clase social en función de los ingresos anuales en Estados Unidos se ha basado en el artículo (Leyva, 2023), tomando los siguientes valores en función de aquellos devueltos por el atributo AnnIcomme:

- 1 - Clase baja si el atributo AnnIncomme tiene alguno de los valores: 01, 02, 03, 04, 05, 06.
- 2 - Clase media si el atributo AnnIncomme tiene alguno de los valores: 07, 08, 09.
- 3 - Clase alta si el atributo AnnIncomme tiene alguno de los valores: 10, 11.
- 9 en el resto de los casos (No sabe, no contesta, dato omitido).

Tiene irritaciones o llagas en los pies (FecFootIrrita)

Indica si el individuo tiene irritaciones o llagas en los pies. La razón de la creación de este campo es doble:

- Unificar este atributo respecto a 2021 y 2022, por no existir en la encuesta de 2021.
- Simplificar la complejidad del atributo "Irritaciones o llagas en los pies" (FootIrrita). Como se puede comprobar en la tabla 4 del "Anexo B. Valores de los atributos" este campo tiene valores difíciles de gestionar por un algoritmo de marchine learning, por ello se dará la oportunidad al analista determinar si la agrupación resultante en este atributo es de utilidad para mejorar el comportamiento de los algoritmos utilizados en este trabajo.

Tomará los siguientes valores:

- 1 - Tiene irritaciones o llagas en los pies, si en el 2021 el individuo indica en el atributo "Irritaciones o llagas en los pies" (FootIrrita) que revisa las irritaciones en los pies de forma diaria, semanal o anual. En el caso de 2022 se toma directamente el valor del atributo FEETSORE del fichero ASCII de la encuesta.
- 2 - No tiene irritaciones ni llagas en los pies, si en el 2021 el individuo indica en el atributo "Irritaciones o llagas en los pies" (FootIrrita) que nunca revisa sus pies. En el caso de 2022 se toma directamente el valor del atributo FEETSORE del fichero ASCII de la encuesta.
- 9 - En el resto de los casos tanto para el 2021, como el 2022 cuando el individuo no sabe, no contesta, dato omitido o no tiene pies.

Consume Vegetales o Fruta (FruitOrVegCon)

Indica si el individuo come fruta o vegetales. La razón de crear este campo es estudiar si los algoritmos de machine learning se comportan mejor que tener dos campos independientes: Consume fruta (FruitCons) y Consume vegetales (VegCons).

Tomará los siguientes valores:

- 1 - Sí consume vegetales o fruta.
- 2 - No consume ni vegetales ni fruta.
- 9 - No sabe, no contesta, dato omitido.

Se destaca que estos atributos sólo existen en la encuesta de 2021.

Consume Vegetales y Fruta (FruitAndVegCon)

Similar al anterior, pero en este caso los valores son:

- 1 - Sí come vegetales y fruta.
- 2 - No como vegetales o fruta o ninguna de los dos.
- 9 - No sabe, no contesta, caracteres dato omitido.

Se destaca que estos atributos sólo existen en la encuesta de 2021.

Distintos atributos objetivos:

El atributo objetivo de este trabajo tiene una especial relevancia por lo que se han generado tres posibles alternativas para evaluar posteriormente cuál es la que mejor se comporta

durante el uso de los algoritmos de machine learning. Todos ellos toman como referencia el atributo Diabetes (BrDiabetes) y considerándose en todos los casos haber sufrido diabetes en el embarazo se estima que no tiene diabetes:

- **Grupo Diabetes (GrDiabetes)**

Este atributo tomará los siguientes valores, destacando que tener prediabetes se considera un caso aparte:

  - 1 - Sí tiene diabetes.
  - 2 - Sí tiene prediabetes.
  - 3 - No tiene diabetes (Diabetes durante el embarazo se considera no tiene diabetes).
  - 9 - No sabe, no contesta, dato omitido.

- **Super Grupo Diabetes (SupGrPreDiabetes)**

Este atributo tomará los siguientes valores, destacando que tener prediabetes se considera se diabético:

  - 1 - Sí tiene Diabetes (Incluye prediabetes).
  - 2 - No tiene Diabetes (Diabetes durante el embarazo se considera no tiene diabetes)
  - 9 - No sabe, no contesta, dato omitido.

- **Super Grupo Diabetes (SupGrNoPreDiabetes)**

Este atributo tomará los siguientes valores, destacando que tener prediabetes se considera se no ser diabético:

  - 1 - Sí tiene diabetes.
  - 2 - No tiene diabetes (Incluye prediabetes, diabetes durante el embarazo se considera no tiene diabetes).
  - 9 - No sabe, no contesta, dato omitido.

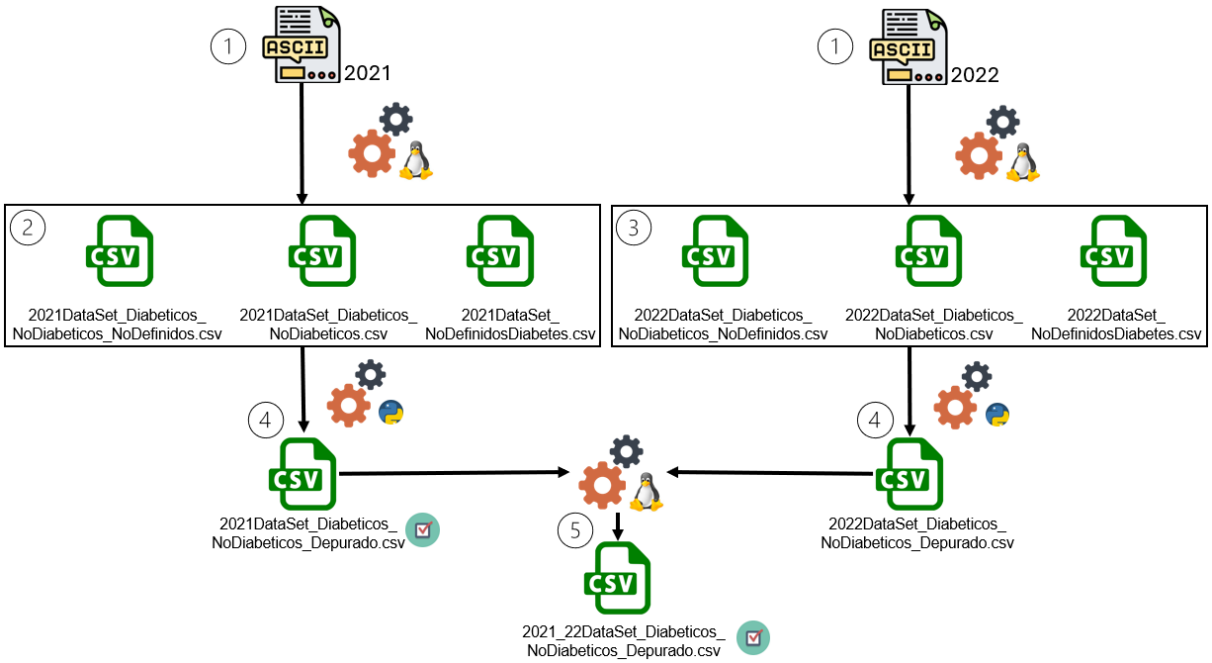
5.2.2. Preprocesamiento, limpieza y obtención de los datasets

A continuación, se describe el procedimiento de obtención de los datasets utilizados en este trabajo que además se dejarán disponibles en el repositorio (Campillo Piqueras, 2024c) el cual se puede revisar en el siguiente enlace (https://github.com/AndreaCampillo/TFM_PrediDia/tree/Datasets) para que la comunidad interesada pueda utilizarlos libremente en futuras investigaciones.

Antes de la descripción detallada de cada una de las etapas es interesante su enumeración para mejor seguimiento (Figura 6):

- Paso 1: Descarga de los ficheros resultantes de las encuestas.
- Paso 2: Generación de los conjuntos de datos correspondientes al 2021.
- Paso 3: Generación de los conjuntos de datos correspondientes al 2022.
- Paso 4: Eliminación de los registros en los que las características relacionadas con tener la enfermedad no han sido actualizadas en la encuesta.
- Paso 5: Fusión de los conjuntos de datos del 2021 y 2022.
- Paso 6: Resumen de los conjuntos de datos obtenidos.

Figura 6. *Etapas de la obtención de los datasets.*



Fuente: Elaboración propia.

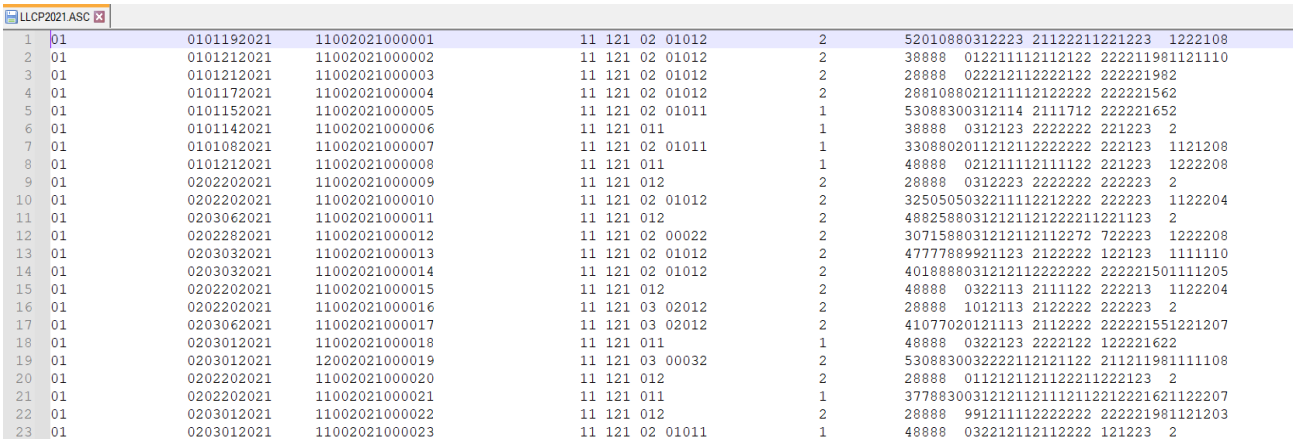
5.2.2.1. Descarga de los ficheros resultantes de las encuestas

Del sitio web en los que se encuentran detallados los distintos ficheros puestos a disposición del público en general por los Centers for Disease Control and Prevention (CDC) relativos a las encuestas Behavioral Risk Factor Surveillance (BRFSS) del 2021 (Centers for Disease Control and Prevention, 2023a) y 2022 (Centers for Disease Control and Prevention, 2023b)se descargan los ficheros comprimidos ASCII de los siguientes enlaces para el 2021 (

https://www.cdc.gov/brfss/annual_data/2021/files/LLCP2021ASC.zip) y 2022 (https://www.cdc.gov/brfss/annual_data/2022/files/LLCP2022ASC.zip).

Se descomprimen ambos ficheros, teniendo cada uno de ellos un aspecto similar al presentado en la figura, quedando disponibles para la siguiente etapa.

Figura 7. *Muestra de fichero ASCII 2021.*



LLCP2021.ASC

1	01	0101192021	11002021000001	11 121 02 01012	2	52010880312223 21122211221223 1222108
2	01	0101212021	11002021000002	11 121 02 01012	2	38888 012211112112122 222211981121110
3	01	0101212021	11002021000003	11 121 02 01012	2	28888 022212112222122 222221982
4	01	0101172021	11002021000004	11 121 02 01012	2	28810880212111112122222 222221562
5	01	0101152021	11002021000005	11 121 02 01011	1	53088300312114 2111712 222221652
6	01	0101142021	11002021000006	11 121 011	1	38888 0312123 2222222 221223 2
7	01	0101082021	11002021000007	11 121 02 01011	1	330880201121211222222 222123 1121208
8	01	0101212021	11002021000008	11 121 011	1	48888 021211112111122 221223 1222208
9	01	0202202021	11002021000009	11 121 012	2	28888 0312223 2222222 222223 2
10	01	0202202021	11002021000010	11 121 02 01012	2	32505050322111122122222 222223 1122204
11	01	0203062021	11002021000011	11 121 012	2	48825880312121121222211221123 2
12	01	0202282021	11002021000012	11 121 02 00022	2	30715880312121121122272 722223 1222208
13	01	0203032021	11002021000013	11 121 02 01012	2	47777889921123 2122222 122123 1111110
14	01	0203032021	11002021000014	11 121 02 01012	2	401888803121211222222 222221501111205
15	01	0202202021	11002021000015	11 121 012	2	48888 0322113 2111122 222213 1122204
16	01	0202202021	11002021000016	11 121 03 02012	2	28888 1012113 2122222 222223 2
17	01	0203062021	11002021000017	11 121 03 02012	2	41077020121113 2112222 222221551221207
18	01	0203012021	11002021000018	11 121 011	1	48888 0322123 2222122 122221622
19	01	0203012021	12002021000019	11 121 03 00032	2	530883003222211211122 211211981111108
20	01	0202202021	11002021000020	11 121 012	2	28888 011212112112221122123 2
21	01	0202202021	11002021000021	11 121 011	1	377883003121211211121122122211621122207
22	01	0203012021	11002021000022	11 121 012	2	28888 991211112222222 222221981121203
23	01	0203012021	11002021000023	11 121 02 01011	1	48888 032212112112222 121223 2

Fuente: Fichero ASCII de BRFSS.

5.2.2.2. Generación de los conjuntos de datos correspondientes al 2021

Se implementa un script (se puede consultar el código fuente en https://github.com/AndreaCampillo/TFM_PrediDia/raw/Scripts/ScriptExtracionDatos2021.sh) que utiliza comandos de la Shell de Linux para la extracción de la información a partir del fichero ASCII del 2021, con el objetivo de obtener los conjuntos de datos en formato csv, con carácter separador ";", siguiendo los siguientes pasos:

- Utilizando el comando "echo" se crea un primer fichero denominado DataSetPrimeraPasada.csv y su cabecera (Figura 8).

Figura 8. *Código Linux de la creación de DataSetPrimeraPasada.csv y su cabecera.*

```
echo "Year;CatBMI;Stroke;HeartDis;PhysExer;HealthIns;NoMedCost;GenHealth;CogDiff;Depression;MentalHlth;MentalState;PhysHlth;WalkDiff;
Gender;AgeRange;EdLevel;AnnIncome;SocClass;UrologyDz;VisionDiff;Asthma;LungDiseases;Arthritis;SmokerTrad;ECigSmok;AlcDrinker;Race;MaritalSt;
LastMedChk;Awareness;FootIrrita;FecFootIrrita;HighBP;HighChol;FruitCons;VegCons;FruitOrVegCon;FruitAndVegCon;MarijuanaCon;SleepHours;
BrDiabetes;GrDiabetes;SupGrPreDiabetes;SupGrNoPreDiabetes" > DataSetPrimeraPasada.csv #Cabecera
```

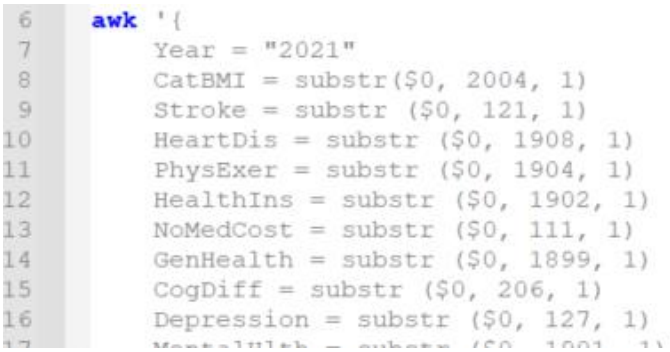
Fuente: Elaboración propia.

- Mediante el comando "awk" y sentencias de tipo "if", "else if", "else" se extraen columnas determinadas del fichero ASCII y se transforman con el objetivo de dejar valores para cada uno de los campos optimizados para el posterior tratamiento

mediante algoritmos de machine learning y al mismo tiempo depurando aquellos caracteres no deseados, como blancos u otros no esperados.

En primer lugar, se definen los atributos haciendo referencia a la posición y longitud según se puede consultar en el "Anexo B. Valores de los atributos". Un ejemplo se puede ver en la figura 9:

Figura 9. *Muestra de código Linux para la extracción de columnas del fichero ASCII.*



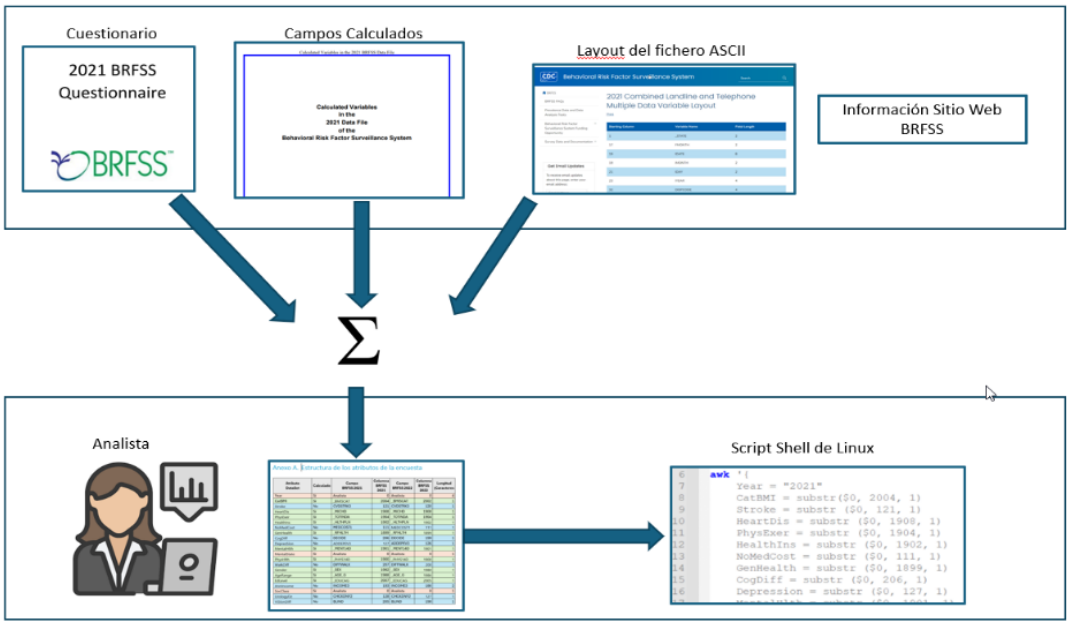
```
awk '{
    Year = "2021"
    CatBMI = substr($0, 2004, 1)
    Stroke = substr ($0, 121, 1)
    HeartDis = substr ($0, 1908, 1)
    PhysExer = substr ($0, 1904, 1)
    HealthIns = substr ($0, 1902, 1)
    NoMedCost = substr ($0, 111, 1)
    GenHealth = substr ($0, 1899, 1)
    CogDiff = substr ($0, 206, 1)
    Depression = substr ($0, 127, 1)
```

Fuente: Elaboración propia.

Para definir los atributos se hace referencia a la posición y longitud en el fichero ASCII de origen. En la figura 10 se detalla el razonamiento para obtener dichos parámetros desde el fichero ASCII de origen. Destacar que en el repositorio del trabajo se ha creado una rama denominada "brfss" (Campillo Piqueras, 2024b), pudiéndose consultar en el siguiente enlace (https://github.com/AndreaCampillo/TFM_PrediDia/tree/brfss) con los principales documentos y ficheros necesarios para obtener los datasets, además se puede consultar un fichero en el que se han aglutinado todas las urls de brfss utilizadas en este trabajo (https://github.com/AndreaCampillo/TFM_PrediDia/blob/brfss/TFM_PredDia_URLsBrfss.txt).

De esta forma, tras estudiar el cuestionario de la encuesta, el documento de campos calculados y el layout del fichero ASCII, confeccionó la tabla 3 del "Anexo A. Estructura de los atributos de la encuesta" y la tabla 4 del "Anexo B. Valores de los atributos". Es de especial utilidad el "Anexo A. Estructura de los atributos de la encuesta", por ser donde queda resumida información clave utilizada en la extracción de información para la elaboración de los datasets utilizados en este trabajo, como por ejemplo la posición y longitud de la variable contenida en el fichero ASCII de origen que se necesita extraer.

Figura 10. *Esquema de obtención de longitud y posición de los atributos.*



Fuente: Elaboración propia.

Como se ha comentado mediante sentencias de tipo if else se depuran los atributos, incluso se calculan otros nuevos como se ha descrito en el punto "5.2.1 Obtención de los atributos."

La mayoría de los valores de salida tratan de simplificar los valores de salida respecto a los que se tienen como referencia en el fichero ASCII. En general, se mantienen los valores útiles, pero otros como suelen ser el valor ".", 7, 9 y caracteres no esperados se fusionan en el valor 9. Se puede ver un ejemplo en la figura 11, donde se trata el atributo "Categoría índice de masa corporal" (CatBMI).

Figura 11. *Código Linux para la obtención y limpieza de atributos.*

```
45      if (CatBMI == "1" || CatBMI == "2" || CatBMI == "3" || CatBMI == "4") {
46        CatBMI = substr($0, 2004, 1)
47      } else {
48        CatBMI = "9" # No sabe, no contesta, caracteres no esperados
49      }
```

Fuente: Elaboración propia.

De la misma forma se tratan la mayoría de los atributos excepto aquellos que son creados durante el desarrollo de este trabajo, por poner un ejemplo la clase social (SocClass) se basa en los ingresos anuales (AnnIncome). En el punto "5.2.1 Obtención de los atributos" se puede consultar otros atributos generados de forma exprofeso para este trabajo, así como el motivo.

Figura 12. *Ejemplo de código Linux de la creación de atributos nuevos.*

```
clase = substr($0, 193, 2)  # Extraer el código salarial codificado
if (clase == "01" || clase == "02" || clase == "03" || clase == "04" || clase == "05" || clase == "06") {
    clase_salario = 1  # Clase Baja
} else if (clase == "07" || clase == "08" || clase == "09") {
    clase_salario = 2  # Clase Media
} else if (clase == "10" || clase == "11") {
    clase_salario = 3  # Clase Alta
} else {
    clase_salario = 9  # No sabe, valor no esperado, caracteres no esperados
}
```

Fuente: Elaboración propia.

- Utilizando el comando "print" se generan los atributos que se han ido gestionando en el tratamiento anterior, añadiéndolos al fichero DataSetPrimeraPasada.csv creado al comienzo del script.

Figura 13. *Ejemplo de uso del comando print.*

```
    print Year";"CatBMI";"Stroke";"HeartDis";"PhysExer";"HealthIns";"NoMedCost";"GenHealth";"CogDiff";
    "Depression";"MentalHlth";"MentalState";"PhysHlth";"WalkDiff";"Gender";"AgeRange";"EdLevel";
    "AnnIncome";"clase_salario";"UrologyDz";"VisionDiff";"Asthma";"LungDiseases";"Arthritis";"SmokerTrad";
    "ECigSmok";"AlcDrinker";"Race";"MaritalSt";"LastMedChk";"Awareness";"FootIrrita";"FecFootIrrita";"HighBP ";
    "HighChol";"FruitCons";"VegCons";"FruitOrVegCon";"FruitAndVegCon";"MarijuanaCon";"SleepHours";"BrDiabetes";
    "GrDiabetes";"SupGrPreDiabetes";"SupGrNoPreDiabetes
}' LLCP2021.ASC >> DataSetPrimeraPasada.csv
```

Fuente: Elaboración propia.

- Con los comandos "awk "y "sed" se limpian caracteres no deseados, generándose un fichero denominado temporal "DataSetSegundaPasada.csv" y otro "2021DataSet_Diabeticos_NoDiabeticos_NoDefinidos.csv" en los que se obtienen los datos ya depurados, incluyendo aquellos registros en los que no se definió si padecen o no la enfermedad.

Figura 14. *Ejemplo de uso de los comandos awk y sed.*

```
awk 'BEGIN {FS=OFS=";"} {gsub("7", "9", $9); gsub("7", "9", $10)} 1'
    DataSetPrimeraPasada.csv > DataSetSegundaPasada.csv

sed 's/ /9/g' DataSetSegundaPasada.csv >
    2021DataSet_Diabeticos_NoDiabeticos_NoDefinidos.csv
```

Fuente: Elaboración propia.

- Se obtiene un fichero "2021DataSet_Diabeticos_NoDiabeticos_NoDefinidos.csv" con los registros en los que se ha definido si tienen la enfermedad o no.

Figura 15. *Código Linux de la obtención del fichero 2021DataSet_Diabeticos_NoDiabeticos.csv.*

```
awk -F';' '$43 != 9' 2021DataSet_Diabeticos_NoDiabeticos_NoDefinidos.csv >
    2021DataSet_Diabeticos_NoDiabeticos.csv
```

Fuente: Elaboración propia.

- Y otro con aquellos que no se han definido si tienen la enfermedad, denominado "2021DataSet_NoDefinidosDiabetes.csv".

Figura 16. *Código Linux de la obtención del fichero 2021DataSet_NoDefinidosDiabetes.csv.*

```
66  awk -F';' '$43 == 9' 2021DataSet_Diabeticos_NoDiabeticos_NoDefinidos.csv  >
67  2021DataSet_NoDefinidosDiabetes.csv
```

Fuente: Elaboración propia.

5.2.2.3. Generación de los conjuntos de datos correspondientes al 2022

La forma de obtener los conjuntos de datos del 2022 es idéntica al 2021 (puede consultar el código fuente en https://github.com/AndreaCampillo/TFM_PrediDia/raw/Scripts/ScriptExtracionDatos2022.sh), excepto que las columnas varían y algunos atributos que no existen, en ese caso se actualizan con valor "9". Para más detalle de los atributos que no existen en cada uno de los años consultar el punto "5.2.1 Obtención de atributos".

5.2.2.4. Eliminación de los registros en los que las características relacionadas con tener la enfermedad no han sido actualizadas en la encuesta.

Con el objetivo de obtener un fichero con los conjuntos de datos sean coherentes e incluyan todas las características con valores definidos del 2021 se utilizó un script de Python (puede consultar el código fuente en https://github.com/AndreaCampillo/TFM_PrediDia/raw/Jupyter_Notebooks/TFM_PrediDia_GestionFicheros.ipynb) como el que se muestra a continuación, tomado como referencia el fichero generado en los pasos anteriores "2021DataSet_Diabeticos_NoDiabeticos.csv":

Figura 17. *Código Python de la limpieza del dataset 2021DataSet_Diabeticos_NoDiabeticos.csv.*

```
df_21_Dia_NoDia_Depurado = df_21_Dia_NoDia.loc[(df_21_Dia_NoDia['CatBMI'] != 9) &
                                               (df_21_Dia_NoDia['Stroke'] != 9) &
                                               (df_21_Dia_NoDia['HeartDis'] != 9) &
                                               (df_21_Dia_NoDia['PhysExer'] != 9) &
                                               (df_21_Dia_NoDia['HealthIns'] != 9) &
                                               (df_21_Dia_NoDia['NoMedCost'] != 9) &
                                               (df_21_Dia_NoDia['GenHealth'] != 9) &
                                               (df_21_Dia_NoDia['CogDiff'] != 9) &
                                               (df_21_Dia_NoDia['Depression'] != 9) &
                                               (df_21_Dia_NoDia['MentalHlth'] != 9) &
                                               (df_21_Dia_NoDia['MentalState'] != 9) &
                                               (df_21_Dia_NoDia['PhysHlth'] != 9) &
                                               (df_21_Dia_NoDia['WalkDiff'] != 9) &
                                               (df_21_Dia_NoDia['Gender'] != 9) &
                                               (df_21_Dia_NoDia['AgeRange'] != 9) &
                                               (df_21_Dia_NoDia['EdLevel'] != 9) &
                                               (df_21_Dia_NoDia['SocClass'] != 9) &
                                               (df_21_Dia_NoDia['UrologyDz'] != 9) &
                                               (df_21_Dia_NoDia['VisionDiff'] != 9) &
                                               (df_21_Dia_NoDia['Asthma'] != 9) &
                                               (df_21_Dia_NoDia['LungDiseases'] != 9) &
                                               (df_21_Dia_NoDia['Arthritis'] != 9) &
                                               (df_21_Dia_NoDia['SmokerTrad'] != 9) &
                                               (df_21_Dia_NoDia['ECigSmok'] != 9) &
                                               (df_21_Dia_NoDia['AlcDrinker'] != 9) &
                                               (df_21_Dia_NoDia['Race'] != 9) &
                                               (df_21_Dia_NoDia['MaritalSt'] != 9) &
                                               (df_21_Dia_NoDia['LastMedChk'] != 9) &
#                                               (df_21_Dia_NoDia['Awareness'] != 9) &
#                                               (df_21_Dia_NoDia['FootIrrita'] != 9) &
#                                               (df_21_Dia_NoDia['FecFootIrrita'] != 9) &
                                               (df_21_Dia_NoDia['HighBP'] != 9) &
                                               (df_21_Dia_NoDia['HighChol'] != 9) &
                                               (df_21_Dia_NoDia['FruitCons'] != 9) &
                                               (df_21_Dia_NoDia['VegCons'] != 9) &
                                               (df_21_Dia_NoDia['FruitOrVegCon'] != 9)]
```

Fuente: Elaboración propia.

Figura 18. *Código Python de la creación del dataset 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv.*

```
# Algunos datos del DataSet
print('Numero of filas: ', len(df_21_Dia_NoDia_Depurado) , ' y columnas: ', len(df_21_Dia_NoDia_Depurado.columns) , '\n')
```

```
Numero of filas:  229655  y columnas:  45
```

```
# Generación del dataset depurado preparado para el EDA
df_21_Dia_NoDia_Depurado.to_csv('2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv', sep=';', index=False)
```

Fuente: Elaboración propia.

Resultando el fichero "2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv" que será la fuente de datos de referencia para la mayoría de los algoritmos que se utilizaran en el resto de las fases del trabajo.

Con el mismo procedimiento se obtuvo el fichero para el 2022: "2022DataSet_Diabeticos_NoDiabeticos.csv",

Figura 19. *Código Python de la limpieza del dataset 2022DataSet_Diabeticos_NoDiabeticos.csv.*

```
df_22_Dia_NoDia_Depurado = df_22_Dia_NoDia.loc[(df_22_Dia_NoDia['CatBMI'] != 9) &
                                               (df_22_Dia_NoDia['Stroke'] != 9) &
                                               (df_22_Dia_NoDia['HeartDis'] != 9) &
                                               (df_22_Dia_NoDia['PhysExer'] != 9) &
                                               (df_22_Dia_NoDia['HealthIns'] != 9) &
                                               (df_22_Dia_NoDia['NoMedCost'] != 9) &
                                               (df_22_Dia_NoDia['GenHealth'] != 9) &
                                               (df_22_Dia_NoDia['CogDiff'] != 9) &
                                               (df_22_Dia_NoDia['Depression'] != 9) &
                                               (df_22_Dia_NoDia['MentalHlth'] != 9) &
                                               (df_22_Dia_NoDia['MentalState'] != 9) &
                                               (df_22_Dia_NoDia['PhysHlth'] != 9) &
                                               (df_22_Dia_NoDia['WalkDiff'] != 9) &
                                               (df_22_Dia_NoDia['Gender'] != 9) &
                                               (df_22_Dia_NoDia['AgeRange'] != 9) &
                                               (df_22_Dia_NoDia['EdLevel'] != 9) &
                                               (df_22_Dia_NoDia['SocClass'] != 9) &
                                               (df_22_Dia_NoDia['UrologyDz'] != 9) &
                                               (df_22_Dia_NoDia['VisionDiff'] != 9) &
                                               (df_22_Dia_NoDia['Asthma'] != 9) &
                                               (df_22_Dia_NoDia['LungDiseases'] != 9) &
                                               (df_22_Dia_NoDia['Arthritis'] != 9) &
                                               (df_22_Dia_NoDia['SmokerTrad'] != 9) &
                                               (df_22_Dia_NoDia['ECigSmok'] != 9) &
                                               (df_22_Dia_NoDia['AlcDrinker'] != 9) &
                                               (df_22_Dia_NoDia['Race'] != 9) &
                                               (df_22_Dia_NoDia['MaritalSt'] != 9) &
                                               (df_22_Dia_NoDia['LastMedChk'] != 9)]
#                                               (df_22_Dia_NoDia['Awareness'] != 9) &
#                                               (df_22_Dia_NoDia['FootIrrita'] != 9) &
#                                               (df_22_Dia_NoDia['FecFootIrrita'] != 9) &
#                                               (df_22_Dia_NoDia['HighBP'] != 9) &
#                                               (df_22_Dia_NoDia['HighChol'] != 9) &
#                                               (df_22_Dia_NoDia['FruitCons'] != 9) &
#                                               (df_22_Dia_NoDia['VegCons'] != 9) &
#                                               (df_22_Dia_NoDia['FruitOrVegCon'] != 9)
```

Fuente: Elaboración propia.

Figura 20. *Código Python de la creación del dataset 2022DataSet_Diabeticos_NoDiabeticos_Depurado.csv.*

```
# Algunos datos del DataSet
print('Numero of filas: ', len(df_22_Dia_NoDia_Depurado) , ' y columnas: ', len(df_22_Dia_NoDia_Depurado.columns) , '\n')
```

```
Numero of filas:  273937  y columnas:  45
```

```
# Generación del dataset depurado preparado para el EDA
df_22_Dia_NoDia_Depurado.to_csv('2022DataSet_Diabeticos_NoDiabeticos_Depurado.csv', sep=';', index=False)
```

Fuente: Elaboración propia.

5.2.2.5. Fusión de los de los conjuntos de datos del 2021 y 2022 en uno único.

Una vez obtenidos los conjuntos de datos depurados del 2021 y 2022 se utiliza un script (puede consultar el código fuente en https://github.com/AndreaCampillo/TFM_PrediDia/raw/Scripts/ScriptFusionDatos2021_22.sh), también basado en comandos de Linux, con el que se fusionan los ficheros en uno sólo para su posterior tratamiento con los algoritmos de machine learning.

Con el comando "cp" se genera el fichero inicial "2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv" con datos de aquellos registros que se ha definido si tienen la enfermedad en 2021.

Se utiliza a continuación el comando "tail" para eliminar la cabecera del fichero del 2022.

Finalmente, se fusionan ambos ficheros con del comando "cat" quedando como resultado el fichero "2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv" con todos los registros del 2021 y 2022 en los que se definieron si los individuos tenían o no la enfermedad.

Figura 21. *Código Python de la creación del dataset 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv.*

```
cp "2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv"
   "2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv"

tail -n +2 "2022DataSet_Diabeticos_NoDiabeticos_Depurado.csv" >
           "2022DataSet_Diabeticos_NoDiabeticos_Depurado_SinCabecera.csv"

cat "2022DataSet_Diabeticos_NoDiabeticos_Depurado_SinCabecera.csv" >>
    "2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv"
```

Fuente: Elaboración propia.

5.2.2.6. Generación del conjunto de datos con individuos que no definieron su estado relativo a la diabetes.

Con el objetivo de obtener un fichero con los conjuntos de datos capaz de poder simular predicciones y que no se haya utilizado durante el resto de las fases del trabajo, se ha generado el fichero 2021DataSet_NoDefinidosDiabetes_Depurado.csv con 126 registros cuyos individuos no especificaron su estado sobre la enfermedad, pero el resto de atributos tienen valores coherentes (puede consultar código el fuente en https://github.com/AndreaCampillo/TFM_PrediDia/raw/Jupyter_Notebooks/TFM_PrediDia_GestionFicheros.ipynb). Este fichero tiene como origen el fichero 2021DataSet_NoDefinidosDiabetes.csv, cuya generación se describe e en el punto "5.2.2.2. Generación de los conjuntos de datos correspondientes al 2021" y luego se le aplican las condiciones necesarias para que el resto de los atributos sean coherentes (Figura 17).

5.2.2.7. Resumen de los conjuntos de datos obtenidos.

Del proceso descrito en los pasos anteriores se han obtenido diez ficheros con los conjuntos de datos que se utilizarán el resto del trabajo. También se publicarán en la rama "Datasets"

del repositorio otros ficheros que finalmente no se emplean, pero pueden ser útiles para la comunidad:

- Ficheros utilizados en el trabajo:
  - 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv: Fichero correspondiente al 2021 con 229.655 registros en los que todos tienen definido el atributo relativo a si los individuos tienen diabetes y el resto de los atributos tienen valores distintos a no sabe, no contestas o dato omitido.
  - 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv: Fichero correspondiente al 2021 y 2022 con 503.592 registros en los que todos tienen definido el campo relativo a si los individuos tienen diabetes y el resto de los atributos tienen valores distintos a no sabe, no contestas o dato omitido.
  - 2021DataSet_NoDefinidosDiabetes_Depurado.csv: Fichero correspondiente al 2021 con 126 registros con individuos que informaron sobre sus datos socioculturales y antropomórficos, pero no definieron su estado respecto a la Diabetes.
- Otros ficheros de interés:
  - 2021DataSet_Diabeticos_NoDiabeticos_NoDefinidos.csv: Fichero correspondiente al 2021 con 438.693 registros en los que se incluyen aquellos con el valor respeto a la diabetes definido y también otros que no se han registrado, por no saber, no contestar o dato omitido.
  - 2021DataSet_Diabeticos_NoDiabeticos.csv: Fichero correspondiente al 2021 con 437.708 registros en los que se incluyen aquellos con el valor respeto a la diabetes definido.
  - 2021DataSet_NoDefinidosDiabetes.csv: Fichero correspondiente al 2021 con 985 registros en los que se incluyen aquellos no definidos respecto al valor de la diabetes.
  - 2022DataSet_Diabeticos_NoDiabeticos_NoDefinidos.csv: Fichero correspondiente al 2022 con 445.132 registros en los que se incluyen aquellos con el valor respeto a la diabetes definido y también otros que no se han registrado, por no saber, no contestar o dato omitido.

- 2022DataSet_Diabeticos_NoDiabeticos.csv: Fichero correspondiente al 2022 con 444.045 registros en los que se incluyen aquellos con el valor respeto a la diabetes definido.
- 2022DataSet_NoDefinidosDiabetes.csv: Fichero correspondiente al 2022 con 1.087 registros en los que se incluyen aquellos no definidos respecto al valor de la diabetes.
- 2022DataSet_Diabeticos_NoDiabeticos_Depurado.csv: Fichero correspondiente al 2022 con 273.937 registros en los que todos tienen definido el atributo relativo a si los individuos tienen diabetes y el resto de los atributos tienen valores distintos a no sabe, no contestas o dato omitido.

5.3. ANÁLISIS ESTADÍSTICO DE DATOS

En este Análisis Exploratorio de Datos (EDA), se analizarán las características de los datos, para conseguir una mejor comprensión de estos. Los datasets utilizados se han obtenido como se ha indicado en el punto “5.2 Obtención de los datasets”. En este apartado se estudiarán tres datasets, uno de ellos del 2021, otro del 2022 y otro procedente de la unión de ambos, 2021_22. En una primera instancia, se ha optado por la opción de estudiar los tres datasets, porque hay atributos interesantes para la diabetes en 2021 como la dieta, el colesterol o la presión arterial que no aparecen en 2022, un detalle de interés es que en 2022 se preguntó por las horas de sueño y consumo de marihuana, ausentes en 2021. También se determinarán que atributos se pueden descartar en las siguientes fases del trabajo, justificándose su eliminación.

Para realizar este EDA se ha utilizado el lenguaje Python utilizando librerías como pandas, seaborn, matplotlib y numpy, además de la herramienta Power Bi para algunas visualizaciones.

Se muestra la carga de los tres data set mediante la url del repositorio git hub, donde se alojan en la figura 22:

Figura 22. *Código Python de la carga de los datasets.*

```
In [2]: url21 = 'https://media.githubusercontent.com/media/AndreaCampillo/Data-Science-Diabetes/main/2021DataSet_Diabeticos_NoDiabeticos_
        diabetes_df_2021 = pd.read_csv(url21, sep=';')

In [3]: url22 = 'https://media.githubusercontent.com/media/AndreaCampillo/Data-Science-Diabetes/main/2022DataSet_Diabeticos_NoDiabeticos_
        diabetes_df_2022 = pd.read_csv(url22, sep=';')

In [4]: url21_22 = 'https://media.githubusercontent.com/media/AndreaCampillo/Data-Science-Diabetes/main/2021_22DataSet_Diabeticos_NoDiabe
        diabetes_df_2021_22 = pd.read_csv(url21_22, sep=';')
```

Fuente: Elaboración propia.

Para todos los datasets las columnas de las que se componen son:

'Year', 'CatBMI', 'Stroke', 'HeartDis', 'PhysExer', 'HealthIns', 'NoMedCost', 'GenHealth', 'CogDiff', 'Depression', 'MentalHlth', 'MentalState', 'PhysHlth', 'WalkDiff', 'Gender', 'AgeRange', 'EdLevel', 'AnnIncome', 'SocClass', 'UrologyDz', 'VisionDiff', 'Asthma', 'LungDiseases', 'Arthritis', 'SmokerTrad', 'ECigSmok', 'AlcDrinker', 'Race', 'MaritalSt', 'LastMedChk', 'Awareness', 'FootIrrita', 'FecFootIrrita', 'HighBP', 'HighChol', 'FruitCons', 'VegCons', 'FruitOrVegCon', 'FruitAndVegCon', 'MarijuanaCon', 'SleepHours', 'BrDiabetes', 'GrDiabetes', 'SupGrPreDiabetes', 'SupGrNoPreDiabetes'.

Se pueden encontrar tres posibles atributos objetivo, las cuales provienen del mismo atributo (BrDiabetes), pero agrupando los datos de distinta forma. Para mayor claridad, las variables objetivo se escribirán haciendo alusión a las categorías que compone cada una, donde N no padecen la enfermedad, P prediabéticos y D diabéticos:

- GrDiabetes (N, P, D) se compone de tres categorías diabéticos, prediabéticos y no diabéticos (Normal).
- SupGrPreDiabetes (N, P+D) se compone de dos categorías diabéticos + prediabéticos y no diabéticos (Normal).
- SupGrNoPreDiabetes (N+P, D) diabéticos y prediabéticos + no diabéticos (Normal).

5.3.1. Características generales del dataset 2021

El conjunto de datos que corresponde a 2021 contiene 229.655 registros y 45 columnas. No tiene ningún valor en blanco y los datos son de tipo categórico int 64, es decir número enteros.

Las frecuencias de cada categoría se pueden consultar en el “Anexo C. Frecuencias de cada categoría”.

5.3.2. Características generales del dataset 2022

El conjunto de datos que corresponde a 2022 contiene 273.937 registros y 45 columnas. No tiene ningún valor en blanco y los datos son de tipo categórico int 64, es decir número enteros.

Las frecuencias de cada categoría se pueden consultar en el "Anexo C. Frecuencias de cada categoría".

5.3.3. Características generales del dataset 2021_22

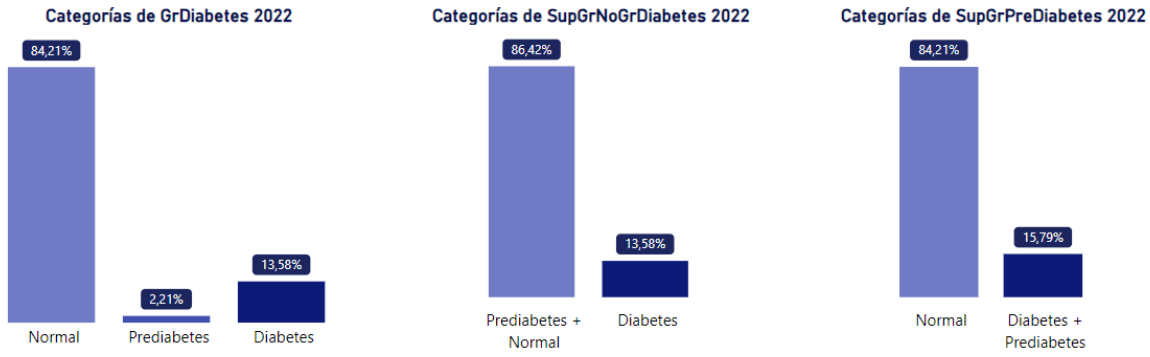
El conjunto de datos que corresponde a 2021 contiene 503.592 registros y 45 columnas. No tiene ningún valor en blanco y los datos son de tipo categórico int 64, es decir número enteros.

Las frecuencias de cada categoría se pueden consultar en el "Anexo C. Frecuencias de cada categoría".

5.3.4. Comparativa y análisis gráfico de los tres datasets

A continuación, se compara de forma general las categorías en términos de tanto por ciento de las tres variables objetivo. Para comprobar que en los tres conjuntos de datos se mantienen las proporciones de cada categoría y proporcionar una idea de cómo se distribuyen los datos.

Figura 23. *Proporciones de cada categoría y distribución de los datos de los tres datasets.*



Fuente: Elaboración propia.

Como muestra la figura 23 los tres conjuntos de datos conservan las mismas proporciones. Siendo la categoría mayoritaria la clase normal presentando el 84% de los datos, la segunda mayoritaria es diabetes rondando entorno al 14% de los datos y por último la clase prediabetes que presenta tan solo al 2% de los datos. Las otras dos variables objetivo también mantienen las proporciones, para la variable SupNoGrDiaebetes (N+P, D) el grupo prediabetes + normal es la suma de las categorías normal y prediabetes de la variable GrDiabetes (N, P, D), rondando el 86 %, mientras que el grupo diabetes se mantiene exactamente igual entorno al 14%. En cambio, respecto la variable SupGrPreDiabetes (N, P+D), la categoría diabetes + prediabetes, es la suma de la categoría diabetes y prediabetes del grupo GrDiabetes (N, P, D) correspondiendo entorno a un 16 % de los datos y el grupo normal se mantiene igual, en un 84 %.

5.3.5. Estudio y análisis de cada atributo

Tras observar como las proporciones y frecuencias por categorías de cada variable se mantienen en los tres datasets, se realiza un estudio de cada atributo en relación con la diabetes, pero sólo con el dataset 2021_22 y con el GrDiabetes (N, P, D). Puesto que todas las proporciones se mantienen, el dataset 2021_22 tiene un mayor número de datos y para este apartado el GrDiabetes (N, P, D) aportará una mayor información, por estar representadas las tres categorías. Si se quisiera estudiar cualquiera de las otras dos variables objetivo bastaría con sumar los porcentajes de las categorías correspondientes. De esta forma se proporciona una idea aproximada de cómo se distribuyen los datos en función de las distintas características. Se realizará con esta metodología los mismos estudios para la mayoría de las variables, excepto para las variables 'HighBP', 'HighChol', 'FruitAndVegCon', 'MarijuanaCon', 'SleepHours' que se estudiarán mediante el dataset del año en el que se han recogido estos datos, de otra forma al no tener la mitad de los datos se produciría un sesgo en los porcentajes.

Al realizar las gráficas mediante porcentajes relativos no existe ningún inconveniente de realizar estas excepciones.

En primer lugar, se presentarán las gráficas con los resultados esperados según la bibliografía consultada, en relación con enfermedades, atributos demográficos, actividad física y salud mental. Posteriormente se presentarán las rechazadas a pesar de la información recogida mediante la literatura. Finalmente se presentan aquellas visualizaciones que aportan información adicional al estudio. **Todas las gráficas se muestran en el "Anexo D. Gráficas del estudio y análisis de cada atributo".**

5.3.5.1. Gráficas acordes a la bibliografía consultada.

Atributos según patologías

Se observa como respecto al tanto por ciento relativo, un mayor número de personas que padecen enfermedades cardiacas, urológicas, pulmonares, han tenido algún ictus, artritis, asma o dificultades visuales tienen también diabetes. Es decir, con mayor frecuencia las personas diabéticas padecen alguna de estas patologías.

Las enfermedades urológicas presentan la mayor diferencia entre el porcentaje de pacientes que la padecen y son diabéticos (40,82%) a diferencia de los pacientes que no padecen este tipo de enfermedades, pero sí son diabéticos (12,60%). Otra patología destacable son las enfermedades cardiacas, con una diferencia de aproximadamente del 22%. Tras analizar las visualizaciones, se podría concluir que hay una relación entre padecer alguna de las enfermedades mencionadas anteriormente y tener diabetes, tal y como se indicó en el apartado "5.1 Justificación de la elección de los atributos".

Atributos demográficos

Respecto a los atributos demográficos, con mayor frecuencia las personas con mayor IMC sufren diabetes tal y como muestra la gráfica correspondiente. Lo mismo ocurre con la edad, cuanto mayor rango de edad se estudie, mayor es la proporción de personas afectadas por la diabetes.

Respecto al estado civil, también sigue la conclusión extraída de la literatura, las personas viudas son más propensas a sufrir diabetes seguidas por las divorciadas o separadas. Y por último respecto el nivel de educación y la clase social, teniendo en cuenta que estas encuestas están realizadas en Estados Unidos, considerado un país desarrollado, se cumple que las clase

baja o personas con menor nivel de estudios presentan un mayor porcentaje de diabéticos que las clases altas y personas con un mayor nivel de estudios.

Actividad física

En cuanto al ejercicio físico, corroboran las conclusiones extraídas de la literatura. Cuando se realiza ejercicio físico frecuentemente o no se presentan dificultades para andar la frecuencia de la diabetes es menor.

Salud Mental

Respecto a los atributos englobados dentro de la salud mental, aunque no son determinantes, se observa diferencias. En términos generales, cuando se presenta una mala salud mental ya sea por depresión, dificultades en la toma de decisiones, periodos de tiempo o en la salud mental en general se presenta un ligero ascenso en la frecuencia de la diabetes. Sin embargo, esta diferencia no es tan significativa como ocurre en otros atributos.

5.3.5.2. Gráficas de información general

Finalmente, estas características aportan información como que un mayor porcentaje de personas con diabetes tienen seguro médico, algo predecible, pues estas personas necesitan revisiones constantes y mayores cuidados. Además, se muestra que un mayor porcentaje de personas diabéticas tienen chequeos médicos anuales, posiblemente por la misma razón que la mencionada anteriormente. Respecto al coste médico, no se hayan diferencias entre permitirse ir al médico.

5.3.5.3. Gráficas realizadas mediante el dataset de 2021

En este apartado se contemplan atributos de los que únicamente se tiene datos en 2021. En primer lugar, la presión arterial muestra un aumento de frecuencia de personas diabéticas cuando se padece una presión elevada. Es decir, cuando se tiene presión arterial elevada, hay un mayor porcentaje de personas afectadas por la diabetes. En cambio, aunque la importancia de la dieta en el desarrollo de la diabetes es fundamental, no se observa que las personas que consumen frutas y/o vegetales tengan una menor frecuencia de la diabetes. Un atributo incluido en este grupo es el del colesterol, como se estudiará en el apartado “5.3.6. Atributos contradictorios respecto a la literatura consultada”, tras su análisis se concluirá que tener colesterol alto no tiene porqué influir en padecer diabetes.

5.3.5.4. Gráficas realizadas mediante el dataset de 2022

Existen dos atributos que sólo presentan datos en 2022 son las horas de sueño y la marihuana.

Respecto a las horas de sueño, no se muestra ninguna visualización porque debido a como han sido recogidos los datos por la encuesta no aporta información relevante.

En cambio, respecto la marihuana, aunque la bibliografía confirma que un mayor consumo de esta sustancia resulta en un peor control glucémico, en los datos las proporciones son las contrarias. Es decir, las personas que consumen marihuana tienen una menor proporción de diabetes. Sin embargo, está diferencia entre la información de la bibliografía y los datos pudiera estar ocasionada por la edad, ya que entre los jóvenes es más común su consumo que en personas adultas, por lo que antes de realizar cualquier afirmación se deberían hacer estudios sobre si la edad influye en el consumo de estas sustancias.

5.3.5.5. Atributos gráficamente incoherentes

En este grupo se engloban los atributos relacionados con las llagas en los pies y la realización del curso sobre la diabetes. No se presentan gráficamente puesto que al ser preguntado sólo a las personas que contestaban que sí tenían diabetes y no a personas que no padecían la enfermedad la información aportada al trabajo no es relevante.

5.3.6. Atributos contradictorios respecto a la literatura consultada

Merecen especial relevancia los hallazgos obtenidos de forma colateral, donde **tras analizar los datos utilizados en este trabajo contradicen a la literatura consultada**, cuyos resultados se han representado gráficamente en el "Anexo D. Gráficas del estudio y análisis de cada atributo", apartado "Graficas no acordes a la bibliografía consultada" En primer lugar, destaca el alcohol ya que según la bibliografía podría ser importante en el desarrollo de la diabetes, pero según los datos recogidos, aunque se consuma alcohol no aumenta la frecuencia de las personas diabéticas. Otro atributo con un comportamiento similar es el colesterol, obtenido únicamente del dataset 2021, por falta de datos en 2022, en algunos artículos se afirma sobre su relación con la enfermedad de estudio. Sin embargo, se puede observar cómo existe descenso de las personas diabéticas cuando se sufre colesterol alto, quizá este hecho sea atribuible a alguna medicación concreta para esta situación y no al colesterol como tal.

Por otro lado, en diversos se estudios se afirma la prevalencia de la diabetes en las mujeres. Sin embargo, se puede observar un ligero aumento de diabetes en hombres, pero esta diferencia es de tan solo el 2%.

En cuanto a la etnia se presenta una situación similar, aunque es cierto que parece que las personas de raza negra no hispánicas son más propensas a padecer diabetes, en general no se presentan grandes diferencias.

Por último, un caso curioso es el tabaco, al igual que los anteriores, la literatura afirma que el consumo de tabaco propicia el padecer diabetes. No obstante, mediante los datos se observa que el porcentaje de diabéticos, fumadores y no fumadores es muy similar. Aunque si se puede apreciar que los no fumadores o los ocasionales sufren menos diabetes, la diferencia porcentual es de tan sólo el 5%, por lo que se necesitaría un mayor estudio sobre ello.

5.4. PREPARACIÓN Y PARTICIÓN DEL CONJUNTO DE DATOS

Tras realizar el EDA y comprender los datos con los que vamos a trabajar se deben de particionar los datos, para poder entrenar y probar los algoritmos. Para ello se dividirán en tres grupos:

- **Entrenamiento** un 60%. Serán los datos utilizados para entrenar el modelo.
- **Validación** un 20%. Serán los datos utilizados para comprobar que el modelo realiza las predicciones de forma adecuada.
- **Prueba** un 20%. Una segunda comprobación, donde estos datos se utilizarán únicamente para asegurar la ausencia de overfitting y no intervendrán en ninguna otra fase del entrenamiento.

Siempre y cuando sea necesario particionar un conjunto de datos antes de la aplicación de algunos de algoritmos utilizados en este trabajo, se realizan según se indica a continuación.

En primer lugar, con la función train_test_split y el parámetro test_size=0.4 se dividen los datos en un 60 % entrenamiento y un 40% el resto. Posteriormente, este 40% se dividirá a su vez en un 20% datos de validación y 20% datos de prueba con la misma función, pero con el parámetro test_size=0.5 para dividirlo a la mitad. Para mejorar estas particiones se ajustan algunos otros parámetros, por ejemplo, random_state=14 para garantizar la reproducibilidad, shuffle=True para excluir datos entre grupos, es decir los datos de entrenamiento, validación y prueba no se solapen entre ellos, y stratify, para mantener la proporción en número de

registros en las distintas particiones, teniendo en cuenta la variable objetivo. En la figura 24 se muestra cómo se implementan las particiones de datos suponiendo que la variable objetivo fuera SupGRPreDiabetes.

Figura 24. *Partición de los datos en entrenamiento, validación y prueba.*

```
train_set, test_set = train_test_split(dfDiabetes, test_size=0.4, random_state=14, shuffle=True, stratify=dfDiabetes['SupGrPreDia
val_set, test_set = train_test_split(test_set, test_size=0.5, random_state=14, shuffle=True, stratify=test_set['SupGrPreDiabetes'
```

Fuente: Elaboración propia.

Una vez los datos se encuentran divididos en los tres grupos con las condiciones explicadas se procede separar las variables de entrada de la variable objetivo. Para ello, en cada partición generada se generan los siguientes grupos:

- X_train, donde mediante la función drop y axis=1 se elimina la variable objetivo, conteniendo únicamente los datos de entrada.
- y_train, donde se copia la última columna, es decir, la variable objetivo.

Esto se realiza de la misma forma con el grupo de validación y prueba.

Figura 25. *División de los datos en datos de entrada y etiquetas.*

```
X_train = train_set.drop('SupGrPreDiabetes', axis=1)
y_train = train_set['SupGrPreDiabetes'].copy()

X_val = val_set.drop('SupGrPreDiabetes', axis=1)
y_val = val_set['SupGrPreDiabetes'].copy()

X_test = test_set.drop('SupGrPreDiabetes', axis=1)
y_test = test_set['SupGrPreDiabetes'].copy()
```

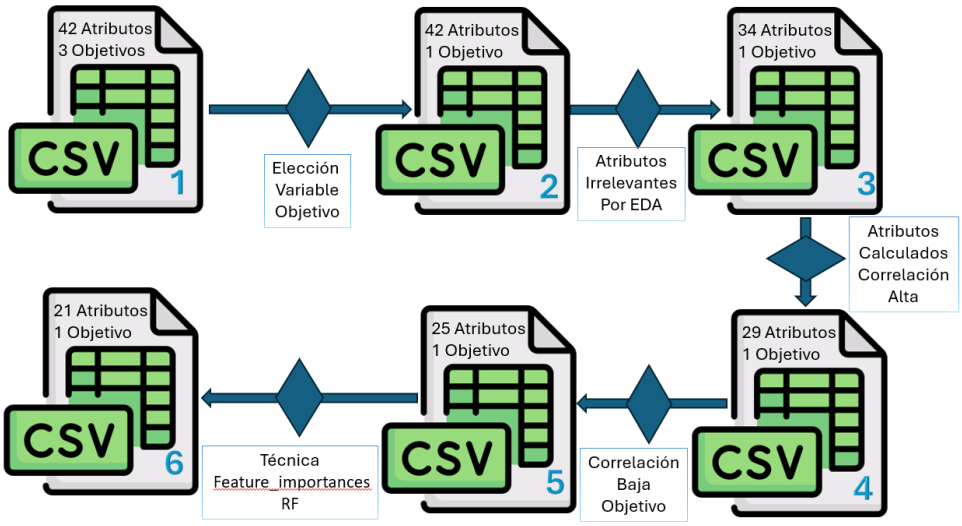
Fuente: Elaboración propia.

Este proceso se reproducirá a lo largo del trabajo siempre y cuando sea necesario generar un modelo a partir de un algoritmo determinado, teniendo en cuenta que como paso previo se deberán eliminar los atributos que el analista haya deducido como irrelevantes.

5.5. CORRELACIONES Y OPTIMIZACIÓN DE ATRIBUTOS IRRELEVANTES

Se procede al estudio de las correlaciones de los distintos atributos con el objetivo de eliminar aquellos que son irrelevantes para el buen funcionamiento de los algoritmos utilizados en las siguientes fases del trabajo. En la figura 26 se muestra las distintas fases realizadas de la eliminación de atributos.

Figura 26. *Fases realizadas para la eliminación de atributos.*



Fuente: Elaboración propia.

Además, puede consultarse el código fuente de los pasos que se describen a continuación en el repositorio del trabajo:

- https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_Correlaciones.ipynb
- https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_EDA.ipynb

5.5.1. Primeras correlaciones y elección de datasets

A continuación, se realiza un estudio de las correlaciones tomando como referencia los atributos de los siguientes conjuntos de datos (para más información véase punto “5.2.2.7. Resumen de los conjuntos de datos obtenidos”):

- 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv.
- 2022DataSet_Diabeticos_NoDiabeticos_Depurado.csv.
- 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv.

Para ayuda a la toma de decisiones respecto a que datasets se van a utilizar, atributo objetivo se va a elegir, así como que atributos se pueden descartar en un primer impacto y tomando como referencia los posibles atributos objetivo ('GrDiabetes', 'SupGrPreDiabetes', ‘SupGrNoPreDiabetes’) se han generado gráficas de las correlaciones se pueden consultar en el “Anexo E. Gráficas de las correlaciones”, donde se pueden visualizar los resultados de mayor

a menor. Además, en el "Anexo F. Valor numérico de las correlaciones" pueden consultar de forma numérica.

Cabe recordar, la importancia del valor absoluto en las correlaciones, midiendo éste la relación entre las dos variables, independientemente si es una relación positiva o negativa.

Se destaca como los atributos de 'HighBP', 'HighChol', en el dataset de 2021 obtienen correlaciones mucho mayores que en el dataset 2021_22. Esto es debido a la falta de datos en 2021_22 de estos atributos ya que en la encuesta de 2022 no se hizo mención en las preguntas a estas características. Lo mismo ocurre, aunque en menor medida con los atributos 'FruitCons', 'VegCons', 'FruitOrVegCon', 'FruitAndVegCon'. Además, por la misma razón, ocurre el caso contrario, los atributos 'MarijuanaCon', 'SleepHours'.

Respecto a las tres variables objetivo se ha observado que en los tres datasets presentan el mismo patrón, el 'SupGrNoPreDiabetes' correlaciona algo mejor con los tres primeros atributos, pero con el resto de los atributos correlaciona mejor el 'GrDiabetes' o el 'SupGrPreDiabetes'. Por lo que por ahora no hay motivos suficientes para descartar alguna de estas variables.

Finalmente, resaltar los diez atributos con correlaciones más altas, en todos los casos: 'Awareness', 'FootIrrita', 'FecFootIrrita', 'WalkDiff', 'UrologyDz', 'HeartDis', 'GenHealth', 'AgeRange', 'CatBMI', 'PhysExer'. Además de estos atributos se incluyen la presión arterial y el colesterol, ('HighBP' y 'HighChol') si sólo se tuviera en cuenta 2021.

Tras realizar un estudio de estas correlaciones se descarta utilizar el dataset de 2022 por no contener información sobre dos de los atributos más correlacionados con la diabetes: el colesterol y la presión arterial. Concluyendo que se en las siguientes fases del trabajo se utilizarán los siguientes datasets:

- 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv
- 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv

Se utilizará este último (2021_22), descartándose los atributos de presión alta y colesterol al no incluirse en las encuestas del 2022, por el hecho que duplica en número de registros al 2021 y pudiera ser que neutralizará la inexistencia de estos dos atributos, mereciendo la pena estudiar cómo afecta a las métricas finales de los modelos de predicción.

5.5.2. Elección de la variable objetivo

Como se comentó en el punto "5.2.1.3 Atributos calculados durante el desarrollo de este trabajo", el analista tiene a su disposición tres atributos sobre la diabetes podrían ejercer el rol de la variable objetivo. Es por ello por lo que en este apartado se estudiará la elección de la mejor variable objetivo entre las tres opciones. Cabe recordar que estas variables presentan la misma información, pero agrupada de forma distinta.

En primer lugar, para elegir la variable objetivo, como se mencionó en el apartado "5.5.1 Primeras correlaciones y elección de datasets" las correlaciones no aportan gran información ya que son muy similares.

En segundo lugar, se descarta el atributo 'SupGrNoPreDiabetes' (N+P, D) porque agrupa los prediabéticos con los no diabéticos, proporcionando aún más datos a la clase mayoritaria y aumentando el desbalanceo de los datos en un 2%. Además, con mucha probabilidad las personas prediabéticas acabaran desarrollando la enfermedad. Ahora bien, se contemplan dos opciones, usar el 'GrDiabetes' (N, P, D) o el 'SupGrPreDiabetes' (N, P+D).

El 'GrDiabetes' (N, P, D), se divide en tres categorías mientras que el 'SupGrPreDiabetes' (N,P+D), en general los algoritmos de inteligencia artificial se comportan mejor con menos categorías. Además, la variable 'SupGrPreDiabetes' (N, P+D) por las razones anteriormente explicadas sobre el desarrollo de la enfermedad, agrupa los prediabéticos con los diabéticos, amentando el número de datos de la clase minoritaria. Por lo tanto, se intuye que sería más lógico escoger la variable 'SupGrPreDiabetes' (N, P+D) como variable objetivo.

Otra razón de peso de elegir el 'SupGrPreDiabetes' (N, P+D) frente a 'GrDiabetes' (N, P, D) es que 'SupGrPreDiabetes' (N, P+D) tiene dos clases posibles, es decir es una variable binaria. Sin embargo, 'GrDiabetes' (N, P, D) es multiclase, tiene tres clases posibles. A priori, tener un variable objetivo binaria puede redundar en una mejor precisión de los algoritmos que se utilizarán en las siguientes fases del trabajo.

5.5.3. Eliminación de atributos irrelevantes

A continuación, se explicará el proceso para eliminar algunos atributos:

5.5.3.1. Primeros atributos eliminados

- **Year**, su eliminación es obvia, puesto que simplemente nos indica el año (2021 ó 2022) al que pertenece cada dato.
- **MarijuanaCon** y **SleepHours**, ambos son atributos pertenecientes al dataset de 2022 y presentaron correlaciones muy bajas con la variable objetivo.
- **AnnIncome**, aunque correlaciona mejor que 'SoscClass', tiene demasiadas categorías, por ello se opta por usar 'SoscClass' ya que agrupa estos valores en únicamente tres categorías. Para más información véase punto "5.2.1.3. Atributos calculados durante el desarrollo de este trabajo".
- **Awareness**, **FootIrrita y FecFootIrrita,** se decide no tener en cuenta estos atributos por el proceso de recogida de datos. Utilizarlos significaría falsear los resultados debido a que sólo se les preguntaba por este síntoma a individuos que habían declarado ser diabéticos.
- **BrDiabetes**, por redundancia de información. Ya que es el atributo original del que se crearon las variables objetivo, por lo que aportaría el mismo tipo de información.
- **GrDiabetes** y **SupGrNoPreDiabetes**, como el caso anterior, se eliminan por redundancia de información, se pueden considerar distintos puntos de vista de la variable objetivo.

5.5.3.2. Segundos atributos eliminados

En esta segunda fase, se estudian los atributos creados a partir de otros atributos, y por tanto tan relacionados entre ellos que proporcionan información muy similar. Para corroborarlo, se realizan las correlaciones entre estos atributos ('CogDiff','Depression',' MentalHlth', 'MentalState', 'FruitCons', 'VegCons',' FruitOrVegCon', 'FruitAndVegCon'), el resultado se muestra en la figura 27.

Figura 27. *Resultado de las correlaciones de los atributos creados en este trabajo.*

	CogDiff	Depression	MentalHlth	MentalState	FruitCons	\
CogDiff	1.000000	0.326701	-0.335294	-0.338276	-0.048196	
Depression	0.326701	1.000000	-0.426838	-0.872581	-0.050572	
MentalHlth	-0.335294	-0.426838	1.000000	0.633652	0.058220	
MentalState	-0.338276	-0.872581	0.633652	1.000000	0.054963	
FruitCons	-0.048196	-0.050572	0.058220	0.054963	1.000000	
VegCons	-0.055755	-0.035636	0.033674	0.041336	0.215359	
FruitOrVegCon	-0.050927	-0.042194	0.044804	0.047365	0.437279	
FruitAndVegCon	-0.057998	-0.050431	0.054802	0.055858	0.870212	
SupGrPreDiabetes	0.075709	0.052008	-0.015420	-0.051008	-0.031038	

	VegCons	FruitOrVegCon	FruitAndVegCon	SupGrPreDiabetes
CogDiff	-0.055755	-0.050927	-0.057998	0.075709
Depression	-0.035636	-0.042194	-0.050431	0.052008
MentalHlth	0.033674	0.044804	0.054802	-0.015420
MentalState	0.041336	0.047365	0.055858	-0.051008
FruitCons	0.215359	0.437279	0.870212	-0.031038
VegCons	1.000000	0.749596	0.507640	-0.046940
FruitOrVegCon	0.749596	1.000000	0.380525	-0.031413
FruitAndVegCon	0.507640	0.380525	1.000000	-0.046565
SupGrPreDiabetes	-0.046940	-0.031413	-0.046565	1.000000

Fuente: Elaboración propia.

Para la obtención de los datos de esta correlación en primer lugar se obtiene un subconjunto del dataset 2021 con los atributos creados durante el desarrollo de este trabajo (Figura 28):

Figura 28. *Código obtención subconjunto de atributos.*

```
# Correlacion entre todos los campos creados por el analista al extraer los datos con SupGrPreDiabetes
subset_dfDiabetes = dfDiabetes[['CogDiff','Depression','MentalHlth','MentalState','FruitCons','VegCons',
```

Fuente: Elaboración propia.

Para después realizar la correlación entre ellos (Figura 29):

Figura 29. *Código implementación de correlaciones.*

```
# Correlación de todos los campos
correlation_matrix = subset_dfDiabetes.corr()
print(correlation_matrix)
```

Fuente: Elaboración propia.

- **Atributos sobre la salud mental**, en este grupo se engloban 'CogDiff','Depression',' MentalHlth', 'MentalState'. En la figura 27 se observa que estos atributos entre ellos tienen correlaciones muy altas, por lo que se elige ‘Depression’ y ‘CogDiff’ como los atributos a utilizar en el modelado, por presentar las correlaciones más elevadas de este grupo de atributos con la variable objetivo.
- **Atributos sobre la dieta**, en este grupo se engloban 'FruitCons', 'VegCons',' FruitOrVegCon', 'FruitAndVegCon', como el caso anterior presentan correlaciones

elevadas entre ellos. Se selecciona 'FruitAndVegCon' por su mayor correlación con la variable objetivo.

5.5.3.3. Terceros atributos eliminados

Para descartar el uso de los atributos menos relacionados con la variable objetivo, se realizan de nuevo el estudio de correlaciones (Figura 30). Se decide eliminar todos los atributos con correlaciones por debajo de 0,03. Los atributos eliminados son: 'HealthIns','ECigSmok','NoMedCost','MaritalSt'.

Figura 30. *Correlaciones ordenadas con el grupo SupGrPreDiabetes (N, P+D).*

```
Out[21]: SupGrPreDiabetes    1.000000
         HighBP              0.263572
         WalkDiff            0.212205
         HighChol            0.207535
         HeartDis            0.171213
         UrologyDz           0.153237
         SocClass            0.142661
         Arthritis           0.132432
         LastMedChk          0.127680
         EdLevel             0.108239
         LungDiseases        0.101177
         Stroke              0.098698
         VisionDiff          0.094550
         AlcDrinker          0.086821
         CogDiff             0.075709
         Depression          0.052008
         SmokerTrad          0.038260
         Gender              0.032266
         HealthIns           0.023001
         ECigSmok            0.021186
         NoMedCost           0.018210
         MaritalSt           0.007762
         Race               -0.034126
         FruitAndVegCon     -0.046565
         Asthma             -0.049569
         PhysHlth           -0.146570
         PhysExer           -0.147603
         CatBMI             -0.190952
         AgeRange           -0.198577
         GenHealth          -0.235988
         dtype: float64
```

Fuente: Elaboración propia.

Para obtener las correlaciones indicadas se utiliza el siguiente código (Figura 31):

Figura 31. *Código obtención correlaciones ordenadas de mayor a menor***.**

```
# Correlacion de los campos restantes con SupGrPreDiabetes
dfDiabetes.corrwith(dfDiabetes['SupGrPreDiabetes']).sort_values(ascending=False)
```

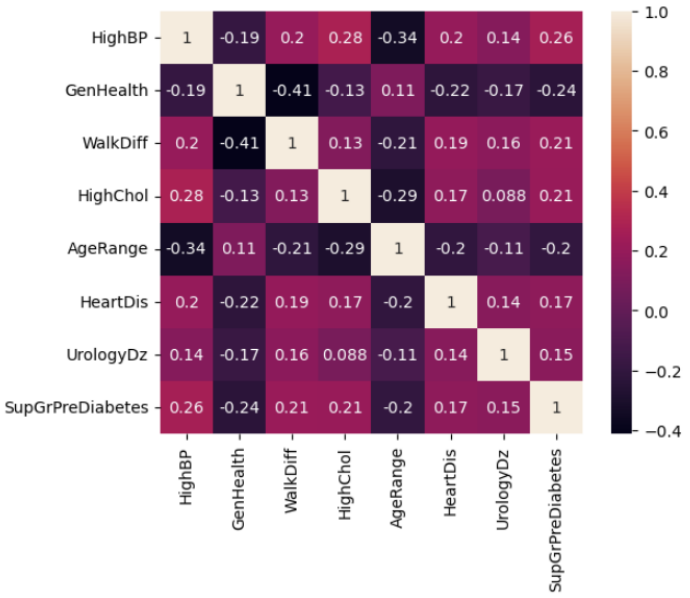
Fuente: Elaboración propia.

5.5.4. Mapas de calor

Antes de pasar a la siguiente de fase de eliminación de atributos con otras técnicas conviene constatar visualmente que los atributos no descartados tienen correlación con la variable objetivo.

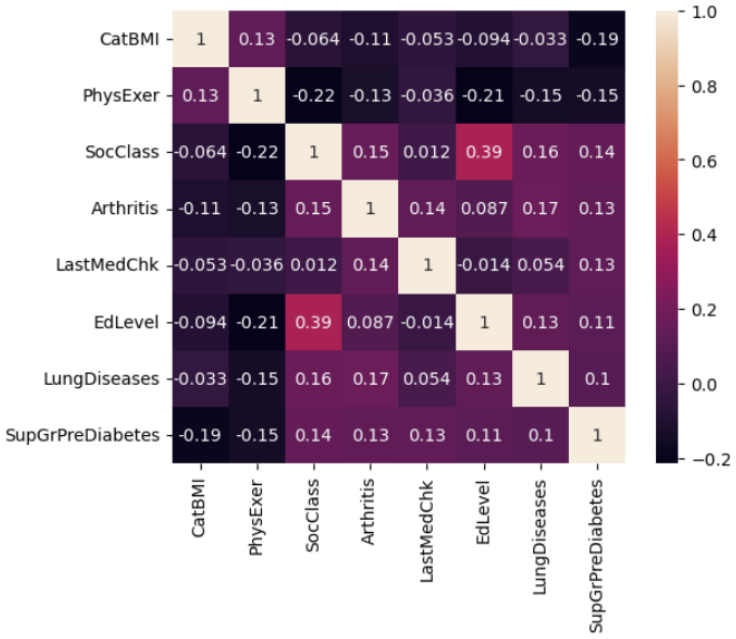
Para realizar los mapas de calor y que sean lo más visuales posibles, se han divido los atributos en cuatro grupos de siete elementos. El grupo uno contiene los atributos más correlacionados y el cuatro los que menos (Figuras 32, 33, 34, 35).

Figura 32. *Mapa de calor del grupo 1*.



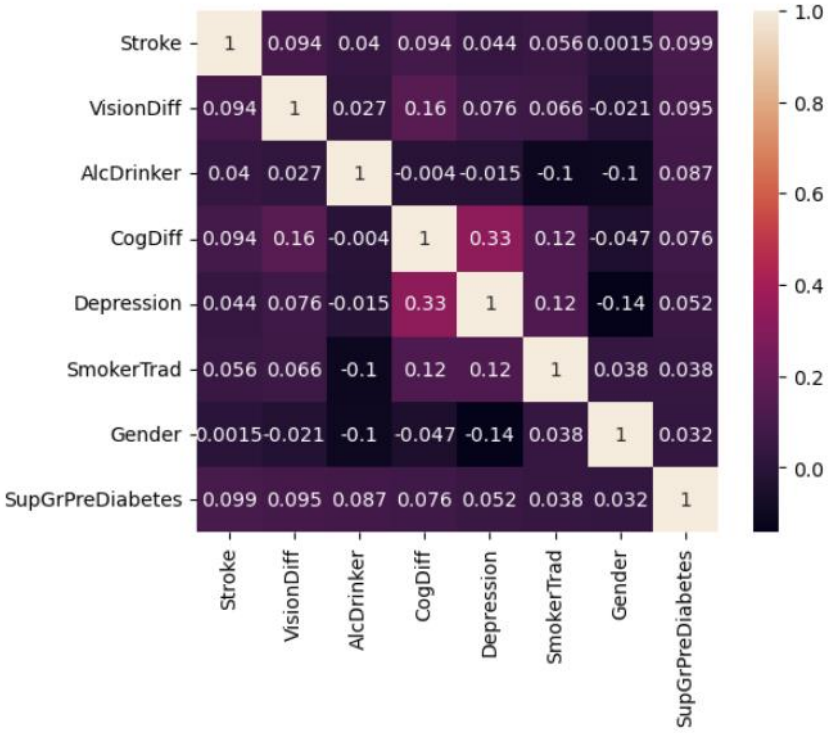
Fuente: Elaboración propia.

Figura 33. *Mapa de calor del grupo 2.*



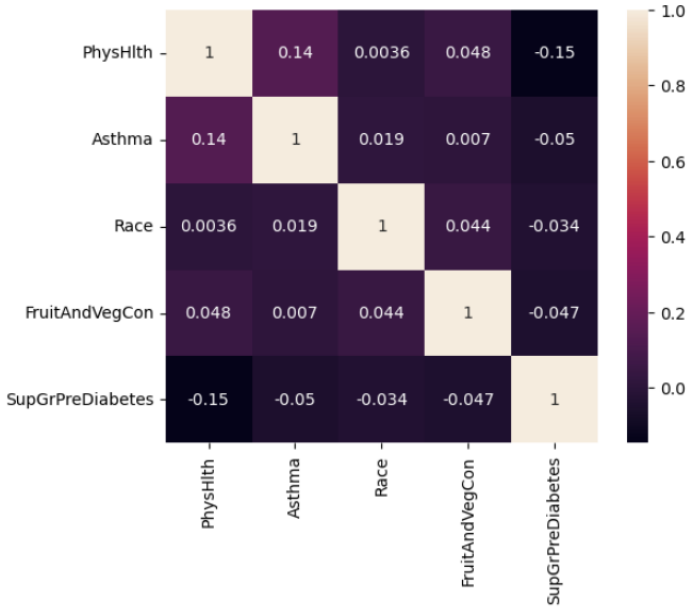
Fuente: Elaboración propia.

Figura 34. *Mapa de calor del grupo 3*.



Fuente: Elaboración propia.

Figura 35. *Mapa de calor del grupo 4.*



Fuente: Elaboración propia.

La obtención de estos mapas de calor ha seguido el mismo patrón de generación, por ejemplo, el código fuente del Grupo 1 se muestra en la figura 36:

Figura 36. *Código fuente generación mapas de calor.*

```
#GRUPO1
grupo1_2021_22 = ['Awareness', 'FecFootIrrita','WalkDiff','GenHealth', 'AgeRange', 'SupGrPreDiabetes' ]
diabetes_df_grupo1_2021_22 = diabetes_df_2021_22[grupo1_2021_22]
correlacion_grupo1_2021_22 = diabetes_df_grupo1_2021_22.corr()
sns.heatmap(correlacion_grupo1_2021_22, annot = True, square = True)
```

Fuente: Elaboración propia

5.5.5. Eliminación de atributos mediante la técnica "Feature_importances_" de Random Forest

De los pasos anteriores quedan 25 características y una variable objetivo: 'CatBMI', 'Stroke', 'HeartDis', 'PhysExer', 'GenHealth', 'CogDiff', 'Depression', 'PhysHlth', WalkDiff, 'Gender', 'AgeRange', 'EdLevel', 'SocClass', 'UrologyDz', 'VisionDiff', 'Asthma', 'LungDiseases', 'Arthritis', 'SmokerTrad', 'AlcDrinker', 'Race', 'LastMedChk', 'HighBP', 'HighChol', 'FruitAndVegCon', 'SupGrPreDiabetes'.

Con el objetivo de reducir aún más el número de atributos irrelevantes se utiliza la técnica basada en la propiedad "Feature_importances_" del algoritmo Random Forest. Para aplicar dicha técnica se dan los siguientes pasos:

- Partición del conjunto de datos del 2021, tal como se detalla en el punto "5.4 Preparación y partición del conjunto de datos", pero incluyendo únicamente las 25 características no descartadas en fases anteriores y la variable objetivo 'SupGrPreDiabetes'.
- Entrenamiento y validación de los datos de entrenamientos con un algoritmo Random Forest y parámetros coherentes, pero sin análisis previo. Así como la obtención de un F1 Score de referencia (Figura 37).

Figura 37. *Obtención F1 Score con Random Forest.*

```
model = RandomForestClassifier(n_estimators=50, class_weight='balanced', random_state=14, n_jobs=-1)
model.fit(X_train, y_train)
```

```
RandomForestClassifier(class_weight='balanced', n_estimators=50, n_jobs=-1,
                       random_state=14)
```

```
y_pred = model.predict(X_val)
```

```
print("F1 score: {:.3f}".format(f1_score(y_val, y_pred, average='weighted')))
```

```
F1 score: 0.793
```

Fuente: Elaboración propia.

- Obtención de lista ordenada de los atributos por importancia para el algoritmo Random Forest una vez entrenado y basado en la propiedad "Feature_importances_" (Figura 38).

Figura 38. *Lista ordenada por importancia de los atributos para el algoritmo Random Forest.*

```
AgeRange          0.096639
HighBP            0.079420
EdLevel           0.077039
SmokerTrad        0.067614
CatBMI            0.065419
PhysHlth          0.054198
Race              0.049281
SocClass          0.048555
HighChol          0.046052
FruitAndVegCon    0.042497
GenHealth         0.039869
Arthritis         0.037147
Gender            0.034291
LastMedChk        0.032422
WalkDiff          0.032163
Depression        0.030542
PhysExer          0.028224
Asthma            0.026513
HeartDis          0.019404
AlcDrinker        0.018468
CogDiff           0.017675
LungDiseases      0.017523
VisionDiff        0.013284
UrologyDz         0.012930
Stroke            0.012832
```

Fuente: Elaboración propia.

Para obtención de esta información se ha implementado el código fuente que se muestra en la figura 39:

Figura 39. *Código fuente obtención de las características siguiendo técnica Random Forest.*

```
# Clasificación de las características por orden de importancia según el modelo Random Forest.
# Para ello se dan los siguientes pasos:
# 1. Crear un diccionario de importancias de características
diccImportances = dict(zip(dfDiabetes.columns, model.feature_importances_))
# 2. Convertir el diccionario a una serie de pandas y ordenar los valores en orden descendente
serieImportancesSorted = pd.Series(diccImportances).sort_values(ascending=False)
# 3. Mostrar las 25 características por orden de importancia
serieImportancesSorted.head(25)
```

Fuente: Elaboración propia.

- Descarte progresivo de los atributos menos importantes y comparación con el F1 Score de referencia.

Finalmente se eliminaron cuatro atributos más ('LungDiseases', 'VisionDiff', 'UrologyDZ', 'Stroke'). Para llegar a la conclusión de que estos atributos no eran relevantes se siguió la siguiente lógica. Una vez ordenadas, se observa que 'LungDiseases', 'VisionDiff', 'UrologyDz' y 'Stroke' ocupan los últimos puestos con valores de 0,01752; 0,013284; 0,012930 y 0,012832 respectivamente. Ahora bien, por qué se eligió eliminar cuatro y no cualquier otro número de

características. Se consideró este número de características ya que se realizaron diferentes pruebas eliminando una, dos, tres y cuatro características y ejecutando el Random Forest con las características restantes en cada caso. Si se compara con la predicción de Random Forest con las 25 características el F1 score fue de 0,793, se observa que cuando se eliminan los cuatro atributos menos importantes según feature_importances_ la F1 score es de 0,790, reduciéndose el F1 Scores en sólo 3 centésimas.

5.6. SELECCIÓN DE MEJORES HIPERPARÁMETROS

Una fase clave en la búsqueda del modelo óptimo es la búsqueda de los mejores hiperparámetros para los algoritmos. Para cada algoritmo la búsqueda del mejor modelo se lleva a cabo de la siguiente forma (puede consultar el código fuente en https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_BusquedaMejoresHiperparametros.ipynb):

- En primer lugar, se prepara y particiona el dataset de 2021, tal y como se describe en el punto “5.4. Preparación y partición del conjunto de datos”, tomando como referencia las 21 mejores características.
- Se asigna a la variable model el tipo de algoritmo utilizado en cada caso (Figura 40).
- Se asigna a la variable param_grid los distintos hiperparámetros que se quieren probar (Figura 40).
- Se asigna a la variable grid_search el resultado del proceso de búsqueda para encontrar la mejor combinación de los parámetros contenidos en param_grid respecto a la métrica f1_weighted, realizada con la función GridSearchCV. La variable model hace referencia al algoritmo que se esté utilizando en cada momento y el parámetro cv=5 son los pliegues que se utilizan para la validación cruzada (Figura 40).
- Se entrena el modelo con el dataset de 2021 y las 21 características seleccionadas (Figura 40).

Figura 40. *Código fuente framework GridSearchCV.*

```
model = # Algoritmo del que se quieren buscar los mejores hiperparámetros
param_grid = { # Rejilla con los hiperparámetros que se desean evaluar }
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid, cv=5, scoring="f1_weighted")
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

- La propiedad grid_search.best_params_ indica cuales fueron los mejores hiperparámetros con sus respectivos valores.
- La propiedad grid_search.best_estimator_ muestra cual fue el mejor modelo que se encontró, con cada uno de los valores de los hiperparámetros. En ocasiones, no se muestran algunos hiperparámetros debido a que el modelo los contiene implícitos.

Este proceso se realiza con cada uno de los algoritmos seleccionados, estudiando los hiperparámetros específicos para cada uno de ellos. En algunos casos, ciertos hiperparámetros se añaden como fijos al modelo, por ejemplo, siempre que el algoritmo lo permite se opta por la selección del balanceo de los datos.

Destacar que la búsqueda de hiperparámetros puede requerir una potencia computacional muy elevada, en nuestro caso algunos de ellos como por ejemplo Random Forest ha llevado más de cinco horas y otros como Support Vector Machine (SVM) varios días de proceso sin interrupciones en ordenador dedicado exclusivamente a esta tarea.

A continuación, se explica los hiperparámetros de cada algoritmo en específico, así como el modelo final. La información de los hiperparámetros se ha obtenido de la página oficial de la librería scikit-learn (Pedregosa et al., 2024)

5.6.1. Navie Bayes Gaussiano

En primer lugar se define el algoritmo a utilizar, en este caso es GaussianNB(). Posteriormente se definen los hiperparámetros a estudiar, además la figura 41 muestra una imagen del código utilizado:

'priors': especifica las probabilidades que tiene cada clase. En este caso se han comparado valores como None para calcular automáticamente las probabilidades a partir de los datos de entrenamiento y otros dos conjuntos de valores posibles [0.999,0.001], [0.85, 0.15].

'var_smoothing': Añade una mayor varianza a los datos, para que los cálculos sean más estables. Se han comparado distintos valores, se debe tener en cuenta que cuanto menor sea este valor menos varianza será añadida.

Figura 41. *Código Python para la búsqueda de mejores hiperparámetros del algoritmo Navie Bayes Gaussiano.*

```
# Definir el modelo GaussianNB
model = GaussianNB()
# Definir los parámetros para GridSearchCV
param_grid = {
    'priors': [None, [0.999,0.001],[0.85,0.15]],
    'var_smoothing': [1e-9, 1e-8, 1e-7]
}
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid, cv=5, scoring="f1_weighted")
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

Tras la ejecución del código se seleccionan los mejores hiperparámetros y valores de estos con grid_search.best_params_ (Figura 42) y con grid_search.best_estimator_ (Figura 42) se genera el mejor modelo.

Figura 42. *GridSearchCV. Mejores hiperparámetros del algoritmo Navie Bayes Gaussiano.*

```
grid_search.best_params_
{'priors': [0.999, 0.001], 'var_smoothing': 1e-09}
```

Fuente: Elaboración propia.

Figura 43. *GridSearchCV. Mejor estimador del algoritmo Navie Bayes Gaussiano.*

```
grid_search.best_estimator_
GaussianNB(priors=[0.999, 0.001])
```

Fuente: Elaboración propia.

5.6.2. Gradient Boosting Classifier

En primer lugar se define el algoritmo a utilizar, en este caso es GradientBoostingClassifier(), se ha decidido dejar fijo el hiperparámetro random_state=14 para paliar la aleatoriedad. Posteriormente se definen los hiperparámetros a estudiar, además la figura 44 muestra una imagen del código utilizado:

"n_estimators": indica la cantidad de árboles que tendrá el modelo. En este caso se comparan los valores de 50, 100 y 200.

"learning_rate": contribución de cada uno de los árboles. Cuanto menor sea el valor menos propenso al sobreajuste, pero será necesario un mayor número de árboles. Se estudian los valores de 0.1, 0.05, 0.01.

"max_depth": Profundidad de cada árbol, cuanto mayor sea el valor más riesgo de sobre ajuste se estudian los valores 3, 5, 7.

"min_samples_split": Número de muestra necesarias para que el nodo pueda dividirse, cuanto mayor sea este valor el árbol será más general y simple, cuanto menor sea el valor, más específico y mayor riesgo de sobreajuste. Se estudian los valores de 2, 5, 10.

"min_samples_leaf": Número mínimo de muestras para cada nodo hoja. Cuanto mayor sea este valor más general y simple será el árbol. Se estudian los valores de 1, 2, 4.

Figura 44. *Código Python para la búsqueda de mejores hiperparámetros del algoritmo Gradient Boosting Classifier.*

```python
# Definir el modelo Gradient Boosting Classifier
model = GradientBoostingClassifier(random_state=14)
# Definir los parámetros para GridSearchCV
param_grid = {
     "n_estimators": [50, 100, 200],
     "learning_rate": [0.1, 0.05, 0.01],
     "max_depth": [3, 5, 7],
     "min_samples_split": [2, 5, 10],
     "min_samples_leaf": [1, 2, 4],
}
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid, cv=5, scoring="f1_weighted")
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

Tras la ejecución del código se seleccionan los mejores hiperparámetros y valores de estos con grid_search.best_params_ (Figura 45) y con grid_search.best_estimator_ (Figura 46) se genera el mejor modelo.

Figura 45. *GridSearchCV. Mejores hiperparámetros del algoritmo Gradient Boosting Classifier.*

```
grid_search.best_params_
```

```
{'learning_rate': 0.1,
 'max_depth': 5,
 'min_samples_leaf': 4,
 'min_samples_split': 10,
 'n_estimators': 200}
```

Fuente: Elaboración propia.

Figura 46. *GridSearchCV. Mejor estimador del algoritmo Gradient Boosting Classifier.*

```
grid_search.best_estimator_
```

```
GradientBoostingClassifier(max_depth=5, min_samples_leaf=4,
                           min_samples_split=10, n_estimators=200,
                           random_state=14)
```

Fuente: Elaboración propia.

5.6.3. Árbol de Decisión

En primer lugar se define el algoritmo a utilizar, en este caso es DecisionTreeClassifier(), se ha decidido dejar fijo el hiperparámetro random_state=14 para paliar la aleatoriedad y class_weight='balanced' para equilibrar el desbalance de los datos. Posteriormente se definen los hiperparámetros a estudiar, además la figura 47 muestra una imagen del código utilizado:

'max_depth': Profundidad máxima del árbol, si es demasiado profundo, aunque capture de forma más exhaustiva los datos puede propiciar al sobreajuste si el valor es demasiado pequeño se puede generar un árbol demasiado general. Se estudian los valores None (sin límite de profundidad) 10 y 20.

'min_samples_split': Número de muestra necesarias para que el nodo pueda dividirse, cuanto mayor sea este valor el árbol será más general y simple, cuanto menor sea el valor, más específico y mayor riesgo de sobreajuste. Se estudian los valores de 2, 5, 10.

'min_samples_leaf': Número mínimo de muestras para cada nodo hoja. Cuanto mayor sea este valor más general y simple será el árbol. Se estudian los valores de 1, 2, 4.

'max_features': Número de características para la división. 'sqrt' y 'log2' tienen en cuenta menos características generando un árbol más general, en cambio None, tiene en cuenta todas las características propiciando el sobreajuste.

'criterion': La función para medir la calidad de las divisiones. Se estudian las opciones 'gini','entropy' y 'log_loss'.

Figura 47. *Código Python para la búsqueda de mejores hiperparámetros del algoritmo Árbol de Decisión.*

```
# Definir el modelo Árboles de Decisión
model = DecisionTreeClassifier(random_state=14, class_weight='balanced')
# Definir los parámetros para GridSearchCV
param_grid = {
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2', None],
    'criterion':['gini', 'entropy', 'log_loss']
}
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid,cv=5,scoring='f1_weighted')
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

Tras la ejecución del código se seleccionan los mejores hiperparámetros y valores de estos con grid_search.best_params_ (Figura 48) y con grid_search.best_estimator_ (Figura 49) se genera el mejor modelo.

Figura 48. *GridSearchCV. Mejores hiperparámetros del algoritmo Árbol de Decisión.*

```
grid_search.best_params_
```

```
{'criterion': 'entropy',
 'max_depth': None,
 'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 2}
```

Fuente: Elaboración propia.

Figura 49. *GridSearchCV. Mejor estimador del algoritmo Árbol de Decisión.*

```
grid_search.best_estimator_
```

```
DecisionTreeClassifier(class_weight='balanced', criterion='entropy',
                       max_features='sqrt', random_state=14)
```

Fuente: Elaboración propia.

5.6.4. Regresión Logística

En primer lugar se define el algoritmo a utilizar, en este caso es LogisticRegression(), se ha decidido dejar fijo el hiperparámetro random_state=14 para paliar la aleatoriedad y class_weight='balanced' para equilibrar el desbalance de los datos. Posteriormente se definen los hiperparámetros a estudiar, además la figura 50 muestra una imagen del código utilizado:

'C': Este parámetro es la regularización inversa. Cuanto menores son los valores mayor regularización. Se estudian los valores de 0.01, 0.1, 1, 10 y 100.

'solver': Algoritmo a utilizar en la optimización. Se estudian los siguientes valores: 'liblinear' (óptimo para conjuntos de datos pequeños y binarios), 'sag' (óptimo para conjuntos de datos grandes y tanto para problemas binarios como multiclase), 'lbfgs' (óptimos para tamaños medianos o grandes y multiclases).

'tol': Tolerancia al criterio de parada, cuanto mayor es el valor el algoritmo se detendrá antes pero será menos preciso. Se comparan los valores: 1e-4, 1e-3 y 1e-2.

'max_iter': Número máximo de interacciones por el resolver. Cuanto mayor sea el valor más tiempo de entrenamiento, pero en algunos casos es necesario para conseguir la convergencia. Se estudian los valores: 100, 200 y 500.

Figura 50. *Código Python para la búsqueda de mejores hiperparámetros del algoritmo Regresión Logística.*

```
# Definir el modelo regresión logística
model = LogisticRegression(class_weight='balanced', random_state=14)
# Definir los parámetros para GridSearchCV
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'solver': ['liblinear', 'sag', 'lbfgs'],
    'tol': [1e-4, 1e-3, 1e-2],
    'max_iter': [100, 200, 500]
}
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid, cv=5, scoring="f1_weighted")
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

Tras la ejecución del código se seleccionan los mejores hiperparámetros y valores de estos con grid_search.best_params_ (Figura 51) y con grid_search.best_estimator_ (Figura 52) se genera el mejor modelo.

Figura 51. *GridSearchCV. Mejores hiperparámetros del algoritmo Regresión Logística.*

```
grid_search.best_params_
```

```
{'C': 10, 'max_iter': 100, 'solver': 'sag', 'tol': 0.01}
```

Fuente: Elaboración propia.

Figura 52. *GridSearchCV. Mejor estimador del algoritmo Regresión Logística.*

```
grid_search.best_estimator_
```

```
LogisticRegression(C=10, class_weight='balanced', random_state=14, solver='sag',
                   tol=0.01)
```

Fuente: Elaboración propia.

5.6.5. Random Forest

En primer lugar, se define el algoritmo a utilizar, en este caso es RandomForestClassifier (), se ha decidido dejar fijo el hiperparámetro random_state=14 para paliar la aleatoriedad y class_weight='balanced' para equilibrar el desbalance de los datos. Posteriormente se definen los hiperparámetros a estudiar, además la figura 53 muestra una imagen del código utilizado:

'n_estimators': indica la cantidad de árboles que tendrá el modelo. En este caso se comparan los valores de 50, 100 y 200.

'max_features': Número de características para la división. 'sqrt' y 'log2' tienen en cuenta menos características generando un árbol más general, en cambio ‘auto’, tiene en cuenta todas las características propiciando el sobreajuste.

'max_depth': Profundidad máxima del árbol, si es demasiado profundo, aunque capture de forma más exhaustiva los datos puede propiciar al sobreajuste si el valor es demasiado pequeño se puede generar un árbol demasiado general. Se estudian los valores None (sin límite de profundidad) 5 y 10.

'min_samples_leaf': Número mínimo de muestras para cada nodo hoja. Cuanto mayor sea este valor más general y simple será el árbol. Se estudian los valores de 1, 2, 4.

'min_samples_split': Número de muestra necesarias para que el nodo pueda dividirse, cuanto mayor sea este valor el árbol será más general y simple, cuanto menor sea el valor, más específico y mayor riesgo de sobreajuste. Se estudian los valores de 2, 5, 10.

'criterion': La función para medir la calidad de las divisiones. Se estudian las opciones 'gini' y 'entropy'.

'**bootstrap**': si es óptimo o no usar muestras con reemplazo.Se estudia los valores True y False.

Figura 53. *Código Python para la búsqueda de mejores hiperparámetros del algoritmo Random Forest.*

```
# Definir el modelo Random Forest
model = RandomForestClassifier(n_jobs=-1,random_state=14, class_weight='balanced')
# Definir los parámetros para GridSearchCV
param_grid = {'n_estimators':[50, 100, 200],
              'max_features':['sqrt', 'log2', 'auto'],
              'max_depth':[None, 5, 10],
              'min_samples_leaf':[1, 2, 4],
              'min_samples_split': [2, 5, 10],
              'criterion':['gini', 'entropy'],
              'bootstrap': [True, False]
              }
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid,cv=5,scoring='f1_weighted', return_train_score=True)
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

Tras la ejecución del código se seleccionan los mejores hiperparámetros y valores de estos con grid_search.best_params_ (Figura 54) y con grid_search.best_estimator_ (Figura 55) se genera el mejor modelo.

Figura 54. *GridSearchCV. Mejores hiperparámetros del algoritmo Random Forest.*

```
grid_search.best_params_
```

```
{'bootstrap': True,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'sqrt',
 'min_samples_leaf': 1,
 'min_samples_split': 5,
 'n_estimators': 100}
```

Fuente: Elaboración propia.

Figura 55. *GridSearchCV. Mejor estimador del algoritmo Random Forest.*

```
grid_search.best_estimator_
```

```
RandomForestClassifier(class_weight='balanced', min_samples_split=5, n_jobs=-1,
                       random_state=14)
```

Fuente: Elaboración propia.

5.6.6. Support Vector Machine (SVM)

En primer lugar se define el algoritmo a utilizar, en este caso es SVC(), se ha decidido dejar fijo el hiperparámetro random_state=14 para paliar la aleatoriedad y class_weight='balanced' para equilibrar el desbalance de los datos. Posteriormente se definen los hiperparámetros a estudiar, además la figura 56 muestra una imagen del código utilizado:

'C': Este parámetro es la regularización inversa. Cuanto menores son los valores mayor regularización. Se estudian los valores de 0.01, 0.1, 1, 10 y 100.

'kernel': Se selecciona el tipo de kernel a usar. Se estudian los valores de 'linear', 'poly', 'rbf' y 'sigmoid'.

'gamma': Cuánta influencia tiene una muestra. Los valores más bajos tienen una influencia grande mientras que los altos tendrán influencias más pequeñas. Se estudian valores: 0.01, 0.1, 1, 10.

'degree': En caso de usar el kernel 'poly' se estudia los distintos valores del grado de este, cuanto menor sea el valor menor complejidad. Se estudian los valores de: 2, 3 y 4.

Figura 56. *Código Python para la búsqueda de mejores hiperparámetros del algoritmo Support Vector Machine.*

```
# Definir el modelo SVM
model = SVC(class_weight="balanced", random_state=14)
# Definir los parámetros para GridSearchCV
param_grid = {'C': [0.01, 0.1, 1, 10, 100],
              'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
              'gamma': [0.01, 0.1, 1, 10],
              'degree': [2, 3, 4]
              }
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid,cv=5,scoring='f1_weighted', return_train_score=True)
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

Sin embargo, conviene destacar distintas dificultades afrontadas. Como se ha comentado se buscaron los mejores hiperparámetros para el Support Vector Machine (SVM), pero debido al alto coste computacional, incluso llegando a utilizar en él entorno Colab de Google CPUs de tipo Unidades de Procesamiento de Tensor de Cloud (TPU v2) sin resultados finales por utilizar

las opciones gratuitas que limitan el tiempo de uso, hubo que reducir los hiperparámetros candidatos como puede verse en la figura 57, eliminándose el kernel polinómico y el hiperplano separador (‘gamma’).

Figura 57. *Código Python para la búsqueda reducida de mejores hiperparámetros del algoritmo SVM.*

```
# Definir el modelo SVM
model = SVC(class_weight="balanced", random_state=14)
# Definir los parámetros para GridSearchCV
param_grid = {'C': [0.01, 0.1, 1, 10],
              #'C': [0.01, 0.1, 1, 10, 100],
              #'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
              'kernel': ['linear', 'rbf', 'sigmoid']
              #'gamma': [0.01, 0.1, 1, 10]
              #'degree': [2, 3, 4]
              }
# Instanciar GridSearchCV
grid_search = GridSearchCV(model, param_grid,cv=5,scoring='f1_weighted', return_train_score=True)
# Ajustar GridSearchCV a los datos de entrenamiento
grid_search.fit(X_train, y_train)
```

Fuente: Elaboración propia.

Finalmente se obtuvieron como mejores parámetros {'C': 0.01, 'kernel': 'sigmoid'} y mejor estimador SVC (C=0.01, class_weight='balanced', kernel='sigmoid', random_state=14), el cual se intentó llevar a la práctica, pero de nuevo, debido al alto coste computacional requerido resultó impracticable.

No obstante, se buscaron de forma manual otros hiperparámetros asumibles desde el punto de vista de capacidad de proceso, encontrado el siguiente estimador SVC(kernel=’rbf’, class_weight=’balanced’), basado en el kernel Radial Basis Function (RBF), apropiado para tareas de clasificación y regresiones (Eskandar, 2023). El resultado final se puede consultar en notebook TFM_PrediDia_SVM.ipynb (Se puede acceder mediante el enlace: TFM_PrediDia/TFM_PrediDia_SVM.ipynb at Jupyter_Notebooks · AndreaCampillo/TFM_PrediDia (github.com)), donde se puede comprobar que alto coste computacional en la creación y entrenamiento del modelo, con tiempos que oscilan entre 26 minutos y 2 horas, así como la obtención de métricas muy bajas si las comparamos con el resto de algoritmos, pareció no merecer la pena seguir invirtiendo esfuerzos por esta vía.

5.6.7. Red Neuronal Artificial Densa

El caso de las Redes Neuronales Artificiales se han explorado distintas configuraciones de forma manual. Después de distintas pruebas y configuraciones se llegó a la conclusión que

una Redes Neuronales Artificiales Densa con buenos resultados relativo a la predicción de la variable objetivo es la que se expone en la figura 58:

Figura 58. *Código Python del algoritmo de la Red Neuronal Artificial Densa.*

```
model = models.Sequential()
model.add(layers.Dense(128, activation='relu', input_shape=(X_train.shape[1],)))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['acc'])
```

Fuente: Elaboración propia.

En primer lugar, se define el modelo a utilizar, en este caso es **Sequential ()**, a continuación se añade la primera capa de entrada densa con 128 neuronas, todas conectadas entre sí, se ha optado por la función de activación relu, **activation='relu'**, una activación típica en las capas profundas y como entrada esperada se ha definido como i**nput_shape=(X_train.shape[1],)** puesto que nos devuelve el número de características que se espera como entrada. Se debe mencionar que se probó con una capa inicial de 64 neuronas, pero se descartó, al obtener un menor F1 Score.

Tras ello, se añaden tres capas densas de 64, 32, 16 neuronas respectivamente, para propiciar un mayor aprendizaje sobre los patrones de los datos, sin llegar al sobreajuste. Después, se añade una capa densa de salida, **layers.Dense(1, activation='sigmoid')**, se opta por poner un valor de uno y la activación sigmoid ya que se está frente a una clasificación binaria.

A continuación se configura el modelo para su posterior entrenamiento, mediante **model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])**. El optimizador adam es eficiente cuando se trabaja con grandes conjuntos de datos y características, además, se opta por utilizar loss='binary_crossentropy' ya que es adecuada para clasificaciones binarias siendo monitorizada la precisión durante el entrenamiento, para observar cómo evoluciona esta métrica durante el entrenamiento.

El siguiente paso es entrenar el modelo mediante el método **model.fit()** (Figura 59). En este caso, **X_train** e **Y_train** son los atributos y etiquetas del entrenamiento respectivamente, **epochs=20** es el número de pasadas completas a los datos de entrenamiento, **batch_size=512** siendo este el número de muestra de datos procesados simultáneamente durante la

actualización de los pesos internos de la red, finalmente los datos **X_val** e **Y_val** son los datos de validación, para monitorizar el sobreajuste. Se debe mencionar que se hicieron diversas pruebas con distinto número de epochs, en concreto con 10, 15, 25, 30. Sin embargo se obtuvo un mayor valor de F1 score con 20 epochs.

Figura 59. *Código del entrenamiento de la Red Neuronal Artificial Densa.*

```
# Entrenamos el algoritmo
history = model.fit(X_train,
                    y_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(X_val, y_val))
```

Fuente: Elaboración propia.

5.7. ENTRENAMIENTO DE LOS ALGORITMOS

En esta fase del trabajo y siguiendo la metodología expuesta el punto "3.3.2. Esquema general de la metodología empleada" se han determinado:

- Datasets.
- Carácterísticas más relevantes.
- Algoritmos supervisados y redes neuronales artificiales densas.
- Mejores hiperpárametros para cada uno de los algoritmos.

Por tanto, ha llegado el momento de la generación y entrenamiento de los algoritmos con el objetivo de obtener las métricas (puede consultar para más información el punto "5.9.1. Métricas obtenidas") con las que poder comparar y seleccionar los mejores modelos.

La generación y entrenamiento de todos los algoritmos siguen un patrón similar, en este punto se describirá de forma genérica los pasos seguidos, pudiéndose consultar de forma detallada los códigos fuentes de cada uno de los algoritmos en el repositorio en nube utilizado:

- Conjunto de árboles (Random Forest): https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_RandomForest.ipynb
- Regresión logística (Logistic Regression): https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_RegresionLogistica.ipynb
- Árboles de decisión (Decision Tree Classifier):

https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_DecisionTreeClassifier.ipynb

- Gradient Boosting Machines (GMB): https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_GradientBoostingMachines.ipynb
- Navie Bayes Gaussiano: https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_NaiveBayesGausiano.ipynb
- Support Vector Machine (SVM): https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_SVM.ipynb
- Redes Neuronales Artificiales Densas: https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_RedesNeuronales.ipynb

5.7.1. Caso 1: Dataset 2021 y 25 características

En este caso se tienen en cuenta las 25 características (sin excluir las que resultaron de la técnica "Feature_importances_" de Random Forest) (Figura 60) con el objetivo de obtener el F1 Score y poderlo comparar con otros modelos basados en 21 características (Caso 2) y comprobar cómo afecta la ausencia de las características 'LungDiseases', 'VisionDiff', 'UrologyDZ' y 'Stroke', descartadas mediante la técnica "Feature_importances_" de Random Forest.

Figura 60. *Muestra del código Python de las 25 características escogidas.*

```
columns = ['CatBMI', 'Stroke', 'HeartDis', 'PhysExer', 'GenHealth',
           'CogDiff', 'Depression', 'PhysHlth', 'WalkDiff','Gender',
           'AgeRange', 'EdLevel', 'SocClass', 'UrologyDz','VisionDiff',
           'Asthma', 'LungDiseases','Arthritis', 'SmokerTrad', 'AlcDrinker',
           'Race', 'LastMedChk', 'HighBP','HighChol', 'FruitAndVegCon',
           'SupGrPreDiabetes']
dfDiabetes = dfDiabetes_load[columns].copy()
```

Fuente: Elaboración propia.

Tras realizar la partición en el conjunto de datos en entrenamiento, validación y prueba (Figura 61):

Figura 61. *Código Python de la partición de los datos.*

```
train_set, test_set = train_test_split(dfDiabetes, test_size=0.4, random_state=14, shuffle=True, stratify=dfDiabetes['SupGrPreI
val_set, test_set = train_test_split(test_set, test_size=0.5, random_state=14, shuffle=True, stratify=test_set['SupGrPreDiabet

X_train = train_set.drop('SupGrPreDiabetes', axis=1)
y_train = train_set['SupGrPreDiabetes'].copy()

X_val = val_set.drop('SupGrPreDiabetes', axis=1)
y_val = val_set['SupGrPreDiabetes'].copy()

X_test = test_set.drop('SupGrPreDiabetes', axis=1)
y_test = test_set['SupGrPreDiabetes'].copy()
```

Fuente: Elaboración propia.

Se calcula mediante los datos de validación el F1 score mediante el algoritmo utilizado en cada caso (Figura 62).

Figura 62. *Código Python para la obtención del F1 score de datos de validación.*

```
start_time = time.time()
model = #Aqui iría el modelo tratado con sus hiperparámetros
model.fit(X_train, y_train)
print("Tiempo en generación del modelo:", round(time.time()-start_time,3), " sg.")
```

```
Tiempo en generación del modelo: 34.111  sg.
```

```
#Predecimos con el el conjunto de validación
y_pred = model.predict(X_val)
```

```
print("F1 score: {:.3f}".format(f1_score(y_val, y_pred, average='weighted')))
```

```
F1 score: 0.816
```

Fuente: Elaboración propia.

Esto servirá para comparar la métrica con la obtenida mediante el dataset 2021 y 21 características, comprobando que la ausencia de las características ('LungDiseases', 'VisionDiff', 'UrologyDZ', 'Stroke') características no influyen en la predicción de ningún algoritmo.

5.7.2. Caso 2: Dataset 2021 y 21 características

En este caso, tras entrenar el modelo con los datos de entrenamiento que contienen las 21 características (excluyendo las que resultaron de la técnica "Feature_importances_" de Random Forest) (Figura 63).

Figura 63. *Muestra del código Python de las 21 características escogidas.*

```
columns = ['CatBMI','HeartDis','PhysExer','GenHealth','CogDiff',
           'Depression', 'PhysHlth', 'WalkDiff','Gender','AgeRange',
           'EdLevel', 'SocClass','Asthma', 'Arthritis', 'SmokerTrad',
           'AlcDrinker','Race', 'LastMedChk', 'HighBP','HighChol', 'FruitAndVegCon',
           'SupGrPreDiabetes']
dfDiabetes = dfDiabetes_load[columns].copy()
```

Fuente: Elaboración propia.

Se calcula mediante los datos de validación el F1 score. Y una vez que se ha comprobado que esta métrica es muy similar a la obtenida con los datos que contenían las 25 características, se calculan las métricas a comparar (F1 score, precisión, sensibilidad, exactitud, especificidad y AUC-ROC) con el dataset de prueba (Figura 64), el cual no ha intervenido en ningún momento durante el proceso de entrenamiento y validación, descartando situaciones de sobre entrenamiento (overfitting).

Figura 64. *Código Python para el cálculo de las métricas con las 21 características.*

```
print("F1 score: {:.3f}".format(f1_score(y_test, y_pred, average='weighted')))
print("Precisión (Precision): {:.3f}".format(precision_score(y_test, y_pred, average='weighted')))
print("Exactitud (Accuracy): {:.3f}".format(accuracy_score(y_test, y_pred)))
print("Especificidad (Specificity): {:.3f}".format(specificity_score(y_test, y_pred)))
print("AUC-ROC: {:.3f}".format(roc_auc_score(y_test, y_pred)))
```

Fuente: Elaboración propia.

5.7.3. Caso 3: Dataset 2021, 21 características y SMOTE

En este caso, antes del entrenamiento del modelo con el objetivo de estudiar técnicas de mitigación del desbalanceo de la clase objetivo (15% diabéticos, 85% no diabéticos), se aplica la técnica de SMOTE (Figura 65) pero sólo a los datos de entrenamiento, manteniendo los datos de validación y prueba como se obtuvieron originalmente.

Figura 65. *Código Python de la aplicación de la técnica SMOTE.*

```
smote = SMOTE(sampling_strategy='auto', random_state=14)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

Fuente: Elaboración propia.

A continuación, se entrena el modelo con estos nuevos datos generados. Y como en el caso anterior se calcula el F1 score con los datos de validación y con los datos de prueba se obtienen las métricas.

5.7.4. Caso 4: Dataset 2021_22 y 18 características

En este caso, se realiza de la misma forma que con el caso 2: Dataset 2021 y 21 características, pero cambiando el dataset por el de 2021_22 que contiene aproximadamente el doble de registros y elimina tres características (‘HighBP’, ‘HighChol’, ‘FruitAndVegCon’) por no estar presentes en los datos de 2022, calculándose mediante los datos de validación el F1 score. Y posteriormente se calculan las métricas a comparar con los datos de prueba.

5.7.5. Técnicas para mitigar el desbalanceo

Como se puede comprobar en el punto "5.3.4. Comparativa y análisis gráfico de los tres datasets", existen unos porcentajes que rondan 85% para no diabéticos y 15% para diabéticos, por ello se han utilizado distintas técnicas para mitigar esta problemática siguiendo las siguientes premisas:

- Para todos los algoritmos se han obtenido métricas utilizando un modelo basado en SMOTE.
- En la obtención de las métricas F1 Score y Precisión se ha utilizado el hiperparámetro average='weighted'.
- Siempre y cuando el algoritmo lo soporte, se ha utilizado el hiperparámetro class_weight='balanced'. Además, sólo por estudiar el comportamiento de los algoritmos también se han generados modelos sin este hiperparámetro.

5.8. ENSAMBLAJE DE MODELOS

Una vez ejecutados todos los modelos de forma individual y obtenidas las métricas, se ha considerado otro punto de vista, realizar un ensamblaje de modelos, de forma que la decisión final sobre la predicción no recaiga en uno sólo.

Con esta idea se ha utilizado la clase VotingClassifier de la librería scikit-learn. Así, partiendo de los modelos obtenidos en fases anteriores se implementa un nuevo modelo basado en los principios de votación por mayoría, es decir la decisión final de la predicción dependerá de la que más ocurrencias tenga (la más votada), proporcionando predicciones, al menos teóricamente, más precisas y robustas que las obtenidas de forma independientes.

Para ello se han realizado distintas pruebas combinando los modelos individuales, en concreto se han obtenido cinco ensamblajes, incluyendo cada uno los siguientes modelos (para un mayor detalle puede consultar el código fuente en https://github.com/AndreaCampillo/TFM_PrediDia/blob/Jupyter_Notebooks/TFM_PrediDia_VotingClassifier.ipynb):

- **Voting Classifier (GBC+GBN+RL+RF) 2021 con class_weight='balanced'**. Gradient Boosting Classifier, Gaussian NB, Logistic Regression y Random Forest.

- **Voting Classifier (GBC+GBN+RL) 2021 con class_weight='balanced'**. Gradient Boosting Classifier, Gaussian NB y Logistic Regression.
- **Voting Classifier (GBC+GBN+RF) 2021 con class_weight='balanced'**. Gradient Boosting Classifier, Gaussian NB y Random Forest.
- **Voting Classifier (GBC+GBN+RL) 2021 con SMOTE**. Gradient Boosting Classifier, Gaussian NB y Logistic Regression.
- **Voting Classifier (GBC+GBN+RL) 2021_22 con class_weight='balanced'**. Gradient Boosting Classifier, Gaussian NB y Logistic Regression.

Para obtener estos modelos se ha trabajado con los datasets de 2021 con 21 características y 2021_22 con 18 características. Los datos se han dividido y preparado en cada caso de la misma forma que en el punto "5.7 Entrenamiento de los algoritmos".

Para la obtención de cada modelo Voting Classifier, descritos anteriormente, se sigue la misma metodología, únicamente varían los algoritmos que incluye cada uno de ellos. Para mostrarlo, se ilustrará un ejemplo de cómo se ha obtenido el modelo Voting Classifier (GBC+GBN+RL+RF), por utilizar mayor número de algoritmos, y con el dataset 2021. Siguiendo este patrón se construirán el resto de los modelos basados en la clase VotingClassifier, utilizando distintas técnicas para mitigar el desbalanceo de datos.

En primer lugar, como se muestra en la figura 66 se generan los modelos individuales de cada algoritmo, con los mejores hiperparámetros en el punto "5.6 Selección de mejores hiperparámetros", se entrena cada uno de ellos y a continuación, se crea el modelo Votting Classifier.

Como se muestra en el código (Figura 66) el algoritmo utilizado es VotingClassifier() y los estimadores son cada uno de los modelos que componen el clasificador, incluyendo la opción de voting='hard'.

Figura 66. *Código Python de generación del algoritmo Voting Classifier.*

```
# Obtención del modelo
start_time = time.time()
modelGbc = GradientBoostingClassifier(max_depth=5, min_samples_leaf=4, min_samples_split=10, n_estimators=200, random_state=14)
modelGnb = GaussianNB(priors=[0.999, 0.001])
modelLr = LogisticRegression(C=10, class_weight='balanced', random_state=14, solver='sag', tol=0.01)
modelRf = RandomForestClassifier(class_weight='balanced', min_samples_split=5, n_jobs=-1, random_state=14)

modelGbc.fit(X_train, y_train)
modelGnb.fit(X_train, y_train)
modelLr.fit(X_train, y_train)
modelRf.fit(X_train, y_train)

modelVoting = VotingClassifier(estimators=[('gbc',modelGbc),('gnb', modelGnb),('lr',modelLr),('rf',modelRf)], voting='hard')

modelVoting.fit(X_train, y_train)
print("Tiempo en generación del modelo:", round(time.time()-start_time,3), " sg.")
```

Fuente: Elaboración propia.

Luego se realizan las predicciones mediante los datos de validación y prueba, para comprobar la ausencia de overfitting. Finalmente se obtiene cada una de las métricas mediante los datos de prueba (Figura 67), el resultado de estos se muestra en los anexos "Anexo G. Modelos obtenidos" y "Anexo H. Métricas y clasificación de los modelos obtenidos".

Figura 67. *Código Python de la obtención de métricas del modelo VotingClassifier.*

```
# Obtención de métricas
y_pred = modelVoting.predict(X_val)
print("F1 score: {:.3f}".format(f1_score(y_val, y_pred, average='weighted')))
```

...

```
# Se comprueba con el dataset de prueba
y_pred = modelVoting.predict(X_test)
```

```
print("F1 score: {:.3f}".format(f1_score(y_test, y_pred, average='weighted')))
print("Precisión (Precision): {:.3f}".format(precision_score(y_test, y_pred, average='weighted')))
print("Exactitud (Accuracy): {:.3f}".format(accuracy_score(y_test, y_pred)))
print("Especificidad (Specificity): {:.3f}".format(specificity_score(y_test, y_pred)))
print("AUC-ROC: {:.3f}".format(roc_auc_score(y_test, y_pred)))
```

...

Fuente: Elaboración propia.

Por último, se ha de comentar que los modelos obtenidos mediante el ensamblaje serán comparados con resto en el punto "5.9 Resultados y comparativa".

5.9. RESULTADOS Y COMPARATIVA

A continuación, se revisan las métricas que se obtuvieron con el objetivo de determinar los mejores modelos respecto a los analizados durante las distintas fases de este trabajo. Para ello se ha ido cumplimentando dos tablas:

- "Anexo G. Modelos obtenidos": Agrupada por tipos de modelos, conteniendo todos los datos recolectados durante las distintas fases del trabajo.

- "Anexo H. Métricas y clasificación de los modelos obtenidos": Contiene un subconjunto de atributos de la mencionada anteriormente, imprescindibles para facilitar la clasificación de mejor a peor modelo.

Además, por cada modelo analizado se generó una métrica diseñada exprofeso para este trabajo, permitiendo de forma aséptica determinar una clasificación de la idoneidad de los modelos. Como anticipo, se podrá comprobar como los modelos basados en Redes Neuronales Artificiales Densas, el modelo resultante del ensamblaje de modelos (Voting Classifier) y Gradient Boosting Machines son buenos candidatos a ser los óptimos. Finalmente, se aplicarán los mejores modelos a un dataset, no utilizado en el resto del trabajo, elaborado a partir de los datos de los encuestados en 2021 que no definieron su estado respecto a la diabetes, pero que el resto de las características tienen un valor definido y coherente. Esto permitirá simular una aplicación real de estos modelos, de lo cual se obtendrán conclusiones muy interesantes para su puesta en producción.

5.9.1. Métricas obtenidas

Es esencial tener métricas objetivas, como las que se describirán más adelante, de los modelos estudiados en este trabajo para poder realizar una clasificación de mejor a peor en contexto de los objetivos de este trabajo (predecir si un individuo es candidato a ser diabético). Por esta razón y como producto final del estudio de los modelos se fueron anotando cada una de las métricas, pudiéndose consultar "Anexo G. Modelos obtenidos". A modo de recordatorio es interesante las métricas son el producto final como resultado de aplicar metodología descrita en el punto "3.3. Metodología del trabajo" consistente en las siguientes fases desarrolladas a en este trabajo:

- Los atributos imprescindibles para obtener los mejores resultados en la predicción de la enfermedad, según el estudio realizado en el punto "5.5. Correlaciones y optimización de atributos irrelevantes".
- La mejor variable objetivo.
- Los datasets más apropiados.
- Los algoritmos de aprendizaje supervisado más adecuados a la problemática del trabajo.
- Las técnicas de balanceo de datos para los algoritmos supervisados seleccionados.

Siguiendo una rigurosa exploración de las capacidades de cada modelo y aplicando distintas técnicas de balanceo de la información, los datasets de 2021 y 2022, así como los atributos más importantes se obtuvieron **treinta y cuatro modelos** distribuidos de la siguiente forma:

- Conjunto de árboles (Random Forest): 5
- Regresión logística (Logistic Regression): 5
- Árboles de decisión (Decision Tree Classifier): 5
- Gradient Boosting Machines (GMB): 3
- Navie Bayes Gaussiano: 3
- Support Vector Machine (SVM): 3
- Redes Neuronales Artificiales Densas: 5
- Voting Classifier (Ensamblaje de modelos): 5

La información de los anexos "Anexo G. Modelos obtenidos" y "Anexo H. Métricas y clasificación de los modelos obtenidos" es la siguiente:

- **Algoritmo:** Cada uno de los utilizados durante el trabajo, teniendo todos en común estar en el ámbito de ser supervisados.
- **Hiperparámetros**: Obtenidos en el apartado "5.6 Selección de hiperparámetros", utilizando la técnica grid_SearchCV y otras pruebas más específicas para las redes neuronales artificiales densas y Support Vector Machine (SVM).
- **Dataset**: Obtenidos y explicados en el apartado "5.2. Obtención de los datasets", con especial referencia a 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv y 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv
- **Registros de cada dataset**: Se indica el número de registros en total del dataset utilizado.
- **Número de diabéticos de cada dataset**: Se indica el número de individuos diabéticos respecto al dataset utilizado.
- **Número de no Diabéticos de cada dataset**: Se indica el número de individuos no diabéticos respecto al dataset utilizado.
- **Técnica de balanceo empleada**: debido al desbalanceo de datos entre los individuos, como se puede comprobar en el punto "5.3.4. Comparativa y análisis gráfico de los tres datasets", donde tenemos unos porcentajes que rondan 85% para no diabéticos y 15%

para diabéticos, se han explorado los distintos modelos utilizando técnicas para amortiguar dicho desequilibrio, indicándose cual se ha empleado (hiperparámetro class_weight='balanced', técnica SMOTE y en las propias métricas).

- **F1 Score (F1)**: Evalúa el rendimiento general del modelo. Un valor cercano a 1 indica que el modelo tiene un buen rendimiento tanto en la clasificación de positivos como en la de negativos.
- **Precisión (P):** Proporción de predicciones positivas que son realmente positivas. Una mayor precisión implica que el modelo es capaz de identificar correctamente los casos positivos.
- **Exactitud (Ex):** Proporción de todas las predicciones correctas, tanto positivas como negativas.
- **Especificidad (Es):** Proporción de negativos que se predicen correctamente como negativos. Una mayor especificidad implica que el modelo es capaz de identificar correctamente los casos negativos.
- **AUC-ROC (A):** Un valor de AUC-ROC cercano a 1 indica que el modelo es capaz de diferenciar muy bien entre las clases, mientras que un valor cercano a 0,5 indica que el modelo no es mejor que una predicción aleatoria.
- **Tiempo invertido en la generación del modelo (T):** Tiempo invertido en segundos relativo a la implementación y entrenamiento.
- **Atributos utilizados**: se indica si se han utilizado 21 atributos más importantes, en el caso del dataset de 2021 ó 18 atributos para el dataset 2021_2022 por las razones que se explicó en el punto “5.2.2. Preprocesamiento, limpieza y obtención de los datasets”.
- **Calidad global del algoritmo (C)**: Como se ha mencionado esta métrica ha sido creada exprofeso para este trabajo para aglutinar en una sola métrica el resto descritas anteriormente, pudiéndose establecer una clasificación de idoneidad y pueda ayudar a discernir los mejores modelos de forma aséptica. Se trata de métrica ponderada basada en el resto, descritas anteriormente, teniendo cada una de ellas un peso asignado teniendo en cuenta el caso de estudio al que se enfrenta, diagnóstico clínico temprano con desbalanceo de la variable objetivo. La fórmula es la siguiente (las siglas de las variables se pueden encontrar en la definición de cada una de las métricas realizada anteriormente):

$$C = (0{,}21 \times F1) + (0{,}205 \times P) + (0{,}1 \times Ex) + (0{,}3 \times Es) + (0{,}18 \times A) - ((0{,}005 \times T) / 179)$$

En otras consideraciones:

- Se otorga un mayor peso a la especificidad (0,3), asignando una mayor importancia a que los clasificados como negativos sean verdaderos negativos y evitar continuar con el proceso de diagnóstico, por tratarse de una problemática en el contexto de un diagnóstico temprano.
- A la precisión se le asigna un peso alto (0,205), puesto que al estar en un entorno clínico es importante cerciorarse que los diagnosticados como positivos sean verdaderos positivos.
- El segundo mayor peso asignado es en el F1Score (0,21), porque es una métrica muy representativa del rendimiento global del algoritmo. Además, se ha utilizado el hipérparámetro average='weighted' para mitigar el desbalanceo de la variable objetivo.
- A la métrica AUC-ROC se le asigna una importancia menor (0,18), aunque es una métrica del rendimiento del modelo, no es decisivo en el entorno de diagnóstico clínico, ya que los datos suelen estar muy desbalanceados, lo que podría dar lugar a resultados engañosos cuando hay una clase predominante.
- Respecto la exactitud (0,1), no es tan interesante en este contexto, puesto que existe un desbalanceo de clases, porque gran parte de personas están sanas, por lo que esta métrica podría ser alta prediciendo simplemente que la mayoría no padecen la enfermedad, pudiendo resultar poco representativa.
- En cuanto al tiempo de creación y entrenamiento (0,005), se le otorgado una menor importancia, ya que, si el modelo fuera extremadamente bueno a pesar del tiempo podría ser útil, si se crea y entrena como paso previo al análisis masivo de datos. Se ha dividido por 179, tiempo mayor de los modelos a excepción de SVM, cuyos tiempos se han descartado por ser muy altos (desde 1.600 hasta 7.738 segundos), pudiendo desvirtuar el resultado final de la fórmula.

Se concluye de la métrica creada que es se cerciora de los verdaderos negativos y verdaderos positivos por separado, teniendo en cuenta el desbalanceo de clases entre personas sanas y enfermas.

5.9.2. Comparativa de los modelos obtenidos

Una vez obtenidas todas las métricas, de todos los modelos, mostradas en el "Anexo H. Métricas y clasificación de los modelos obtenidos", se proceden a comparar entre ellos, con el objetivo de elegir los óptimos. A efectos de identificar los modelos más adecuados, la información resultante del "Anexo H. Métricas y clasificación de los modelos obtenidos", se ha ordenado por el índice de calidad y creado cuatro grupos con código de colores:

- Índice de calidad igual a 0,82 (verde)
- Índice de calidad igual a 0,81 (azul)
- Índices de calidad menores a 0,81 pero aceptables (naranja)
- Modelos descartados (rojo)

Modelos con índice de calidad 0,82 (verde)

Se observa que hay seis modelos con el mismo índice de calidad 0,82. cuyas métricas más relevantes se presentan en la tabla 2 "Tabla métricas de los mejores modelos", para un mayor detalle se puede consultar la tabla 6 "Modelos, métricas y clasificación" en el "Anexo H. Métricas y clasificación de los modelos obtenidos"

Nota: Para mantener la integridad y legibilidad de la información, la tabla 2 "Tabla métricas de los mejores modelos" ha sido trasladada a la página siguiente debido a su tamaño. El resto de la página se ha dejado intencionalmente en blanco por este motivo.

Tabla 2. *Tabla de métricas de los mejores modelos.*

Ranking	Algoritmo	DataSet	Atributos	F1 Score	Precisión	Exactitud	Especificidad	AUC-ROC	Tiempo Normalizado
1	Redes Neuronales Artificiales Densas	2021_22	18	0,81	0,81	0,85	0,97	0,57	0,15
2	Voting Classifier (GBC+GBN+RF)	2021	21	0,82	0,81	0,84	0,96	0,61	0,40
3	Gradient Boosting Machines (GBM)	2021	21	0,81	0,81	0,84	0,97	0,59	0,46
4	Redes Neuronales Artificiales Densas	2021	21	0,81	0,82	0,85	0,97	0,59	0,07
5	Redes Neuronales Artificiales Densas	2021	25	0,81	0,82	0,85	0,98	0,58	0,07
6	Voting Classifier (GBC+GBN+RL+RF)	2021	21	0,82	0,81	0,84	0,96	0,61	0,43

Fuente: Elaboración propia.

Estos seis mejores modelos se pueden dividir por tipo de algoritmo utilizado en tres grupos: Redes Neuronales Artificiales Densas (RNAD), Gradient Boosting Machines (GBM) y Voting Classifier (VC). A continuación, se compara cada uno de estos grupos por tipo de algoritmo:

- Respecto a las **RNAD** la mayor diferencia entre ellas es el número de características utilizadas. En una primera instancia se podría pensar que el modelo que trabaja con las 21 o 25 características son mejores que el que utiliza 18 características ya que la precisión es una centésima menor y la métrica AUC-ROC también es una o dos

centésimas menor respectivamente. Sin embargo, el conseguir valores tan similares con menos características es un hallazgo de gran relevancia, pues con menos información, incluso omitiendo información respecto a la tensión arterial y el colesterol se puede llegar a las mismas conclusiones que el resto de los modelos, siempre y cuando se utilicen más registros durante la fase de entrenamiento, debe tenerse en cuenta que este modelo se entrenó con la fusión de los datasets de los años 2021 y 2022. Es por ello, se decide como mejor modelo de las RNAD el modelo entrenado con 18 características y el dataset fusionado 2021 y 2022.

- En cuanto al **VC**, hay dos modelos posibles entre los seis mejores, uno de ellos incluye los modelos Gradient Boosting Machines (GBC) + Navie Bayes Gaussiano (GBN) + Regresión Logística (RL) + Random Forest (RF) y el otro los modelos Gradient Boosting Machines (GBC) + Navie Bayes Gaussiano (GBN) + Random Forest (RF). Ambos obtienen exactamente los mismos valores en las métricas, excepto en el tiempo de generación y entrenamiento, donde el segundo de ellos es de cinco segundos menor. Por esta razón, se selecciona como mejor modelo del grupo VC el VC (GBC+GBN+RF).
- Respecto al **Gradient Boosting Machines (GBM)** destaca por ser el mejor modelo basado en un algoritmo de machine learning supervisado tradicional. Obtiene métricas muy similares a las conseguidas por los modelos VC (GBC+GBN+RF) y RNAD, pero invirtiendo un tiempo superior en la creación y entrenamiento del modelo, aunque no excesivo si se compara con VC (GBC+GBN+RF).

De esta forma, se puede llegar a las siguientes conclusiones respecto a los mejores modelos:

1. RNAD con 18 atributos y dataset 2021_22 puede ser considerado como el mejor modelo debido a una mayor especificidad respecto el VC (GBC+GBN+RF) y una mayor exactitud respecto el GBM, además de necesitar un menor tiempo de generación y entrenamiento, así como menor uso de atributos.
2. En segundo lugar, VC (GBC+GBN+RF), aunque el GBM tiene un valor de una centésima mayor que el VC (GBC+GBN+RF) este último tiene un mayor F1 score y AUC-ROC, además de un menor tiempo en la generación y el entrenamiento. Por todo es elegido segunda opción.
3. En tercera posición, el modelo GBM.

Modelos con índice de calidad 0,81 (azul)

Los diez siguientes modelos obtienen un índice de calidad de 0,81, siendo todas las métricas muy similares entre ellas, sin nada relevante que destacar excepto en el tiempo de generación y entrenamiento, aunque no se considera una limitación en el contexto de este trabajo, sí cabe mencionar como los modelos de Navie Bayes Gaussiano y la Regresión Logística tardaron menos de un segundo en generarse.

Modelos con índices de calidad menores a 0,81 pero aceptables (naranja)

El tercer gran grupo representado en naranja, considerados como menos óptimos, pero con cierto grado de tolerancia. Se destacan los modelos basados en Árboles de Decisión por su rapidez la generación y entrenamiento. Otras métricas relevantes son F1 Score del Naive Bayes gaussiano con SMOTE (0,81). También de cierta importancia los modelos de la Regresión Logística con SMOTE, Redes Neuronales Artificiales Densas con SMOTE con 21 y 25 características, Gradient Boosting Machine con SMOTE y Regresión Logística con class_weight='balanced, los cuales consiguen algunas métricas muy similares a los mejores modelos. Tiene una mención especial que algunos de los modelos de este grupo han llegado a obtener mayores valores en cuanto al AUC-ROC, pero como se mencionó en el apartado anterior esta métrica tiene menor importancia teniendo en cuenta nuestra problemática.

Modelos descartados (rojo)

Como se describió en el punto “5.6.6. Support Vector Machine (SVM)”, los modelos basados en estos algoritmos han sido descartados debido a su altísimo coste computacional, haciéndolos impracticables para una puesta en producción.

5.9.3. Aplicación de los mejores modelos a nuevos datos de individuos

Una vez recorridas todas las fases del trabajo en el cual se han seleccionado los mejores:

- Conjuntos de datos.
- Atributos.
- Hiperparámetros de los distintos algoritmos de aprendizaje supervisado.
- Modelos basados en los algoritmos de aprendizaje supervisado.

Y como se puede constatar en el notebook TFM_PrediDia_AplicacionMejoresModelos.ipynb (https://github.com/AndreaCampillo/TFM_PrediDia/raw/Jupyter_Notebooks/TFM_PrediDia_

AplicacionMejoresModelos.ipynb), se procede a poner en práctica con un conjunto de datos no utilizado hasta el momento, con el que se pretende simular una implementación con casos reales. De esta forma, se realiza un experimento para estudiar el comportamiento de los mejores modelos en un primer triage de la enfermedad.

Para ello se llevarán a cabo las siguientes tareas:

- Selección de un conjunto de datos, novedoso en el trabajo, en los que se incluyen individuos que no definieron su estado respecto a la Diabetes.
- Obtención de los dos mejores modelos desarrollados a lo largo del trabajo.
- Predicción de la enfermedad en función del conjunto de datos novedoso referenciado anteriormente.
- Estudio de los resultados obtenidos.
- Recomendaciones en el uso conjunto de los mejores modelos.

Como **conjunto de datos con la capacidad de simular una implementación en un sistema de producción real** se ha seleccionado el fichero 2021DataSet_NoDefinidosDiabetes_Depurado.csv (https://github.com/AndreaCampillo/TFM_PrediDia/raw/Datasets/2021DataSet_NoDefinidosDiabetes_Depurado.csv), para más información sobre su obtención puede consultar el punto "5.2.2.6 Generación del dataset de individuos que no tienen definida información relativa a la diabetes.", compuesto de 126 registros, en la que los encuestados no proporcionaron información relativa a la diabetes y el resto de atributos tienen información coherente.

A continuación, se implementan los **dos mejores modelos** utilizando las **21 características más importantes:**

- Voting Classifier(GBC+GBN+RL) compuesto de Gradient Boosting Machines, Navie Bayes Gaussiano y Regresión Logística (Figura 68):

Figura 68. *Código Python de la creación del modelo Voting Classifier(GBC+GBN+RL).*

```
# Modelo Voting Classifier(GBC+GBN+RL)
start_time = time.time()
modelGbc = GradientBoostingClassifier(max_depth=5, min_samples_leaf=4, min_samples_split=10, n_estimators=200, random_state
modelGnb = GaussianNB(priors=[0.999, 0.001])
modelLr = LogisticRegression(C=10, class_weight='balanced', random_state=14, solver='sag', tol=0.01)

modelGbc.fit(X_train, y_train)
modelGnb.fit(X_train, y_train)
modelLr.fit(X_train, y_train)

modelVoting = VotingClassifier(estimators=[('gbc',modelGbc),('gnb', modelGnb),('lr',modelLr)], voting='hard')

modelVoting.fit(X_train, y_train)
print("Tiempo en generación del modelo:", round(time.time()-start_time,3), " sg.")
```

Fuente: Elaboración propia.

- Red Neuronal Artificial Densa (RNAD) (Figura 69):

Figura 69. *Código Python de la creación del modelo de la Red Neuronal Artificial Densa.*

```
start_time = time.time()
modelRNA = models.Sequential()
modelRNA.add(layers.Dense(128, activation='relu', input_shape=(X_train.shape[1],)))
modelRNA.add(layers.Dense(64, activation='relu'))
modelRNA.add(layers.Dense(32, activation='relu'))
modelRNA.add(layers.Dense(16, activation='relu'))
modelRNA.add(layers.Dense(1, activation='sigmoid'))

modelRNA.compile(optimizer='adam',
             loss='binary_crossentropy',
             metrics=['acc'])

history = modelRNA.fit(X_train,
                    y_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(X_val, y_val))
print("Tiempo en generación del modelo:", round(time.time()-start_time,3), " sg.")
```

Fuente: Elaboración propia.

Procediéndose a realizar las predicciones con ambos modelos, obteniéndose los siguientes resultados, se recuerda que hasta este punto no se conoce ningún estado relativo a la enfermedad para los 126 individuos tomados como muestra:

- Voting Classifier(GBC+GBN+RL) compuesto de Gradient Boosting Machines, Navie Bayes Gaussiano y Regresión Logística (Figura 70):

Figura 70. *Código Python para la predicción mediante el modelo Voting Classifier(GBC+GBN+RL).*

```
# Se realiza la predicción con individuos con la característica Diabetes no definida
y_pred_modelVoting = modelVoting.predict(dfDiabetesNoDefinidos)
```

Fuente: Elaboración propia.

  - 107 No diábeticos: 85%
  - 19 Diabéticos: 15%

- Red Neuronal Artificial Densa (RNAD) (Figura 71):

Figura 71. *Código Python para la predicción mediante el modelo de la Red Neuronal Artificial Densa.*

```
# Se realiza la predicción con individuos con la característica Diabetes no definida
y_pred_proba = modelRNA.predict(dfDiabetesNoDefinidos)
y_pred_modelRNA = (y_pred_proba > 0.5).astype(int)
```

Fuente: Elaboración propia.

  - 114 No diábeticos: 90%
  - 12 Diabéticos: 10%

Una vez obtenidos los resultados anteriores se procede a su **estudio y comparación**:

1. Se observa que respecto a los porcentajes de individuos que presuntamente pudieran presentar Diabetes o ausencia de ésta, **ambos son aceptables**, aunque el modelo Voting Classifier(GBC+GBN+RL) se acerca con gran precisión a los datos representados en las gráficas del punto "5.3.4. Comparativa y análisis gráfico de los tres datasets" en los que rondan 85% para no diabéticos y 15% para diabéticos, alejándose ligeramente de estas métricas la RNAD.
2. Se destaca que el modelo **RNAD**, aunque es consiste en la obtención de resultados, no siempre son idénticos. Este fenómeno es normal y es debido a la **incertidumbre** derivada de la naturaleza aleatoria de los datos y del proceso de entrenamiento, se puede leer una interesante discusión sobre este tema en (Quora, n.d). Aunque este hecho pudiera parecer poco relevante, basado en estas evidencias, la autora dará una serie de recomendaciones al final de este punto, para obtener un sistema de predicción lo más fiable posible.
3. Utilizando métricas se comparan ambos resultados (Figura 72):

Figura 72. *Código Python y resultado de la comparación de las predicciones.*

```
print("F1 score: {:.3f}".format(f1_score(y_pred_modelVoting, y_pred_modelRNA, average='weighted')))
print("Precisión (Precision): {:.3f}".format(precision_score(y_pred_modelVoting, y_pred_modelRNA, average='weighted')))
print("Exactitud (Accuracy): {:.3f}".format(accuracy_score(y_pred_modelVoting, y_pred_modelRNA)))
print("Especificidad (Specificity): {:.3f}".format(specificity_score(y_pred_modelVoting, y_pred_modelRNA)))
print("AUC-ROC: {:.3f}".format(roc_auc_score(y_pred_modelVoting, y_pred_modelRNA)))
```

```
F1 score: 0.922
Precisión (Precision): 0.928
Exactitud (Accuracy): 0.929
Especificidad (Specificity): 0.991
AUC-ROC: 0.785
```

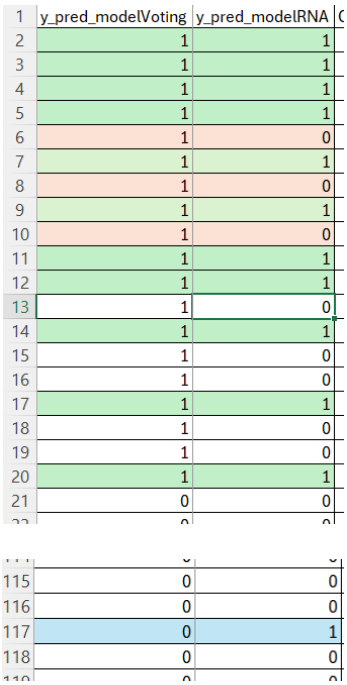
Fuente: Elaboración propia.

Deduciéndose que ambos modelos tienen un grado alto de similitud en sus predicciones.

4. Durante el proceso de comparación se ha podido deducir que ambos modelos son capaces de predecir de forma similar, pero no exacta, por ello y debido a que son pocos registros se procede a realizar una comparación manual entre ambas predicciones. Por esta razón, se genera el fichero 2021DiabetesNoDefinidos_Voting_RNA.csv (https://github.com/AndreaCampillo/TFM_PrediDia/raw/Datasets/2021DiabetesNoDefinidos_Voting_RNA.csv) en los que se incluyen dos columnas al conjunto de datos original, una con las predicciones del modelo Voting Classifier(GBC+GBN+RL) - y_pred_modelVoting y otra del modelo de la RNAD - y_pred_modelRNA. Ordenándose por la que más ocurrencias tiene respecto a la enfermedad (y_pred_modelVoting) con herramientas tipo Excel, como puede verse en la figura, se concluye:

- En todos los casos excepto uno (fila 117) (Figura 73), siempre que el modelo RNDA predice que el individuo tiene la enfermedad, el modelo Voting Classifier(GBC+GBN+RL) lo corrobora.
- El modelo Voting Classifier(GBC+GBN+RL) tiene mayor número de predicciones positivas (tiene Diabetes) que el modelo Red Neuronal Artificial Densa.

Figura 73. *Comparación de los valores de predicción.*



	y_pred_modelVoting	y_pred_modelRNA
2	1	1
3	1	1
4	1	1
5	1	1
6	1	0
7	1	1
8	1	0
9	1	1
10	1	0
11	1	1
12	1	1
13	1	0
14	1	1
15	1	0
16	1	0
17	1	1
18	1	0
19	1	0
20	1	1
21	0	0
115	0	0
116	0	0
117	0	1
118	0	0

Fuente: Elaboración propia.

Tras el análisis comparativo y el estudio del comportamiento de los dos mejores modelos frente a datos como reales el **sistema final debería implementarse con las siguientes premisas y en el orden que se describe** a continuación:

1. Realizar una predicción con el modelo modelo Voting Classifier(GBC+GBN+RL), priorizándolo frente al modelo RNAD.
2. Predecir con el modelo RNAD, para corroborar los datos obtenidos.
3. Aquellos que casos que ambos difieran, avisar al profesional sanitario que existe una alta probabilidad que el individuo tuviera indicios de padecer la enfermedad. En estos casos, al no ser masivas las diferencias en las predicciones, es preferible tratar los falsos positivos, la ausencia de acciones con ese colectivo podría provocar diagnosticar individuos como sanos, cuando en realidad existe probabilidad de tener la enfermedad.

6. Código fuente y datos analizados

6.1. Código fuente

El código fuente, así como los datasets generados durante el trabajo se ha puesto a disposición del público en general en un repositorio de tipo Git Hub (Campillo Piqueras, 2024a) se puede acceder directamente mediante el siguiente enlace: https://github.com/AndreaCampillo/TFM_PrediDia).

Para mayor claridad el presente repositorio se ha organizado en cinco ramas:

- Main (Campillo Piqueras, 2024a)(https://github.com/AndreaCampillo/TFM_PrediDia/tree/main) : Rama por defecto con información sobre el tipo de licenciamiento, introducción y estructura del respositorio.
- Datasets (Campillo Piqueras, 2024c)(https://github.com/AndreaCampillo/TFM_PrediDia/tree/Datasets): Datasets generados durante el trabajo.
- Jupyter_Notebooks (Campillo Piqueras, 2024d) (https://github.com/AndreaCampillo/TFM_PrediDia/tree/Jupyter_Notebooks): Código fuente de los notebooks escritos en leguaje Python utilizando el entorno de programación Jupyter.
- Scripts (Campillo Piqueras, 2024e) (https://github.com/AndreaCampillo/TFM_PrediDia/tree/Scripts): Código fuente basados en lenguaje de scripting de la Shell de Linux Bash utilizado en la generación de los datasets.
- brfss (Campillo Piqueras, 2024b) (https://github.com/AndreaCampillo/TFM_PrediDia/tree/brfss): Ficheros pdfs, URLs y datasets originales del proveedor (brfss) utilizado como referencia en el proyecto. La información se puede encontrar en su ubicación original, pero ha parecido conveniente dejar copia en este repositorio, con una doble finalidad: El lector de este trabajo tenga centralizada la información y evitar que en el futuro la ubicación original pudiera ser cambiada.

6.2. Datos Analizados

Como se comentó en apartados anteriores, en especial punto "5.2.2.1. Descarga de los ficheros resultantes de las encuestas" los datos analizados en este trabajo tienen como origen el sitio web de los Centers for Disease Control and Prevention (CDC) relativos a las encuestas Behavioral Risk Factor Surveillance (BRFSS) del 2021 (Centers for Disease Control and Prevention, 2023a) (https://www.cdc.gov/brfss/annual_data/annual_2021.html) y 2022 (Centers for Disease Control and Prevention, 2023b) (https://www.cdc.gov/brfss/annual_data/annual_2022.html)

Pero además se han utilizado otros recursos de dicho sitio web que a continuación se enumeran y se han resumido en el fichero del repositorio https://github.com/AndreaCampillo/TFM_PrediDia/raw/brfss/TFM_PredDia_URLsBrfss.txt :.

- Información de las encuestas desde 1988 hasta 2022: https://www.cdc.gov/brfss/annual_data/annual_data.htm
- Año 2021:
    - Información encuesta:

https://www.cdc.gov/brfss/annual_data/annual_2021.html

    - Cuestionario:

https://www.cdc.gov/brfss/questionnaires/pdf-ques/2021-BRFSS-Questionnaire-1-19-2022-508.pdf

    - Layout del fichero LLCP2021.ASC:

https://www.cdc.gov/brfss/annual_data/2021/llcp_varlayout_21_onecolumn.html

    - Matriz de campos calculados:

https://www.cdc.gov/brfss/annual_data/2021/summary_matrix_21.html

    - Explicación campos calculados:

https://www.cdc.gov/brfss/annual_data/2021/pdf/2021-calculated-variables-version4-508.pdf

    - Dataset LLCP2021.ASC (comprimido .zip):

https://www.cdc.gov/brfss/annual_data/2021/files/LLCP2021ASC.zip

- Año 2022:
  - Información encuesta:

https://www.cdc.gov/brfss/annual_data/annual_2022.html

  - Cuestionario:

https://www.cdc.gov/brfss/questionnaires/pdf-ques/2022-BRFSS-Questionnaire-508.pdf

  - Layout del fichero LLCP2022.ASC:

https://www.cdc.gov/brfss/annual_data/2022/llcp_varlayout_22_onecolumn.html

  - Matriz de campos calculados:

https://www.cdc.gov/brfss/annual_data/2022/summary_matrix_22.html

  - Explicación campos calculados:

https://www.cdc.gov/brfss/annual_data/2022/pdf/2022-calculated-variables-version4-508.pdf

  - Dataset LLCP2022.ASC(comprimido .zip):

https://www.cdc.gov/brfss/annual_data/2022/files/LLCP2022ASC.zip

7. Conclusiones

Gracias a este trabajo se confirma la viabilidad de obtener un diagnóstico temprano de la diabetes tipo II, sin la inclusión de datos clínicos.

En primer lugar, se ha mostrado cómo las métricas de algunos de los mejores modelos obtenidos se igualan o incluso en algunos casos superan a las obtenidas por otros autores que utilizaron datos clínicos.

Cabe destacar la importancia de datasets completos y actualizados que incluyan características relevantes para la diabetes tipo II. Este trabajo proporciona varios datasets de los años 2021 y 2022. Además de haberse hecho un estudio de características socioculturales y antropomórficas relacionadas con la enfermedad.

Por otro lado, tras el análisis de las características relacionadas con la diabetes se concluye que no todas ellas son relevantes para la predicción de la enfermedad, reduciendo el número a 21 características. Sin embargo, en el caso del dataset 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv se han utilizado 18 características debido a la falta de información de tres características en la encuesta de 2022, pero es de suma importancia destacar que los modelos que utilizan este dataset con las 18 características obtienen métricas muy similares a las obtenidas mediante las 21 características del dataset 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv. Esto es debido a que el dataset 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv es una fusión de los dataset 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv y 2022DataSet_Diabeticos_NoDiabeticos_Depurado.csv teniendo el doble de registros que estos por separado. Es por ello, que se concluye que es tanta la importancia de la calidad de los datos, así como su cantidad, puesto que, con un menor número de características, pero un mayor número de registros también se pueden obtener buenos resultados.

Se concluyó que los mejores modelos eran Redes Neuronales Artificiales Densas con 18 características y el dataset 2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv, Voting Classifier (GBC+GBN+RF) con 21 características y el dataset 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv y en tercer lugar Gradient Boosting Machines (GBM) con 21 características y el dataset 2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv. Sin embargo, se ha conseguido dar un

paso más allá aplicando estos mejores modelos seleccionados a un conjunto de datos de individuos que no contestaron si tenían diabetes o no, simulando pruebas similares a las que podrían ser sometidos los modelos obtenidos. En este caso se ha probado con los dos mejores las Redes Neuronales Artificiales Densas y con el Voting Classifier (GBC+GBN+RF) ambos con 21 características para facilitar la comparación. Finalmente, una vez obtenidas las predicciones con ambos modelos se comprueba y demuestra que los modelos funcionan correctamente ya que presentan proporciones muy similares a las que teníamos en los datasets con individuos que habían definido su estado respecto a la enfermedad. Además, en la mayoría de los casos concuerda el resultado de ambos modelos, aunque es cierto que el modelo Voting Classifier (GBC+GBN+RF) detecta en un porcentaje pequeño más personas que supuestamente tienen diabetes que la Red Neuronal Artificial Densa, es por ello que se propone finalmente, el predecir la diabetes con ambos modelos y contrastar los resultados y en caso de que haya discordancia entre los resultados realizar las pruebas clínicas a estos individuos.

Como se ha explicado, con el sistema propuesto y realizando la predicción con ambos modelos es posible realizar un diagnóstico temprano de la diabetes tipo II. Todo esto ahorraría tiempo y recursos al sistema sanitario y reduciendo el número de pruebas clínicas. Además, sería posible aplicarlo a regiones con pocos recursos ya que no tendrían que invertir grandes cantidades de dinero ni tiempo consiguiendo obtener un diagnóstico para la enfermedad.

Indicar que probablemente este trabajo sea pionero en el diagnóstico precoz de la Diabetes tipo II respecto al uso del dataset del 2022, no se tiene constancia de que se haya utilizado. Durante el estudio de viabilidad, se evaluó descartarse por carecer de información clave en el diagnóstico de la enfermedad (colesterol y presión arterial). Finalmente, aunque el trabajo de generación de los datasets se duplicaba se decidió gestionar con la esperanza de que la no existencia de características claves se pudiera mitigar con la posibilidad de disponer de un dataset masivo (más de 500.000 registros) que fusionaran las encuestas de 2021 y 2022. Definitivamente, el esfuerzo mereció la pena, por dos razones, la primera y más importante, el mejor modelo (Red Neuronal Artificial Densa) tiene como dataset de referencia el resultante de la fusión de 2021 y 2022. La segunda, quizá colateral, dicho superdataset se ha publicado en el repositorio en nube para la comunidad interesada en mejorar lo desarrollado en este trabajo puedan utilizarlo o incluso ser de utilidad en otros proyectos.

Aunque no es objetivo principal y como tarea colateral a este trabajo se han estudiado el comportamiento de atributos que según la literatura podría afectar a padecer diabetes, por ejemplo, el tabaquismo, el sexo del individuo, el colesterol alto o la etnia, sin embargo, el análisis exploratorio de datos (EDA) muestra información contradictoria a lo indicado en la literatura.

Por último, indicar que este trabajo también podría ser considerado un patrón metodológico en proyectos similares. En todo momento se ha seguido la metodología de facto de proyectos basado en machine Learning y deep learning, pudiendo ser de utilidad a otros analistas que requieran una guía de implementación de diagnósticos de enfermedad basadas en estas tecnologías.

8. Limitaciones y prospectiva

8.1. Limitaciones

A continuación, se citan algunas limitaciones que se han encontrado a la hora de desarrollar este trabajo:

- Limitación computacional. Debido al inesperado alto coste computacional requerido por el algoritmo Support Vector Machine (SVM) y a pesar de haber utilizado el entorno Colab de Google con CPUs de tipo Unidades de Procesamiento de Tensor de Cloud (TPU v2), en la modalidad gratuita, no fue posible encontrar los mejores hiperparámetros, por ello y aunque se ha intentado de varias formas (por ejemplo, dejando un ordenador encendido durante días procesando de forma dedicada) los resultados obtenidos por falta de potencia computacional deja los resultados obtenidos de este algoritmo (SVM) inconclusos.
- Los datos de las encuestas son únicamente de Estados Unidos pudiendo conllevar sesgos si se extrapola al resto de regiones mundiales.
- Las encuestas de 2022 no incluyen tres características importantes, colesterol, presión arterial y consumo de frutas o vegetales. Sobre todo, sería relevante tener datos del colesterol y la presión arterial ya que se posicionan como características más relacionadas con la enfermedad. Si se hubieran tenido, probablemente el dataset resultante la fusión de los datos de 2021 y 2022 hubiera sido la base para obtener mejores resultados.

8.2. Trabajo futuro

A continuación, se citan algunas líneas futuras que serían interesante para mejorar o continuar con el trabajo:

- Utilizar Python durante la generación de los datasets, en lugar de la Shell de Linux para conseguir una mayor homogeneidad desde el punto de los leguajes de programación utilizados.
- Almacenar en la generación de los datasets los valores de la clase objetivo como 0 y 1, en lugar de 1 y 2, evitando tener que realizar transformaciones durante las distintas fases del trabajo.

- Implementar funciones con las partes comunes del código, concentrando toda la programación de los modelos en un solo notebook.
- Estudio de la reducción de dimensionalidad y agrupación de características utilizando técnicas de tipo Análisis de Componentes Principales (PCA).
- Obtener financiación para la creación de una cuenta de pago en Google Colab, evitando las limitaciones computacionales y pudiendo ejecutar en un menor tiempo el código de obtención de los mejores hiperparámetros entre otros, sobre todo respecto al modelo Support Vector Machine (SVM).
- Realizar un estudio más exhaustivo sobre los hiperparámetros de las Redes Neuronales Artificiales Densas o implementar búsquedas de parámetros automatizadas similares a las realizadas con el framework GridSearchCV.
- Incluir las características de colesterol, presión arterial y consumo de frutas y vegetales en las siguientes encuestas, manteniendo las 21 características en años venideros y no 18 como ocurre en 2022.
- Si en futuras encuestas se incluyen características relacionadas con el colesterol y tensión arterial, crear otro superdataset y entrenar los mejores modelos con datos masivos, estudiando a continuación como afecta a las métricas obtenidas.
- Aplicar las encuestas otras regiones del mundo distintas a Estados unidos.
- Estudio de las razones por las que el análisis exploratorio de datos de algunos atributos contradice la literatura.
- Crear un formulario de encuesta o triage con las 21 mejores características, que pudieran ser actualizadas en una aplicación que devolviera de forma inmediata al profesional médico si el individuo encuestado pudiera predecir si un paciente padece o no diabetes tipo II, con su correspondiente implementación masiva en el sistema sanitario.

Referencias bibliográficas

Adesoba,T. P., & Brown, C. C. (2023). Trends in the Prevalence of Lean Diabetes Among U.S.Adults, 2015–2020. *American Diabetes Association, 46*(4), 885–889. doi:10.2337/dc22-1847

Adler-Milstein,J., H. Chen, J., & Dhaliwal, G. (2021). Next-Generation Artificial Intelligence for Diagnosis: From Predicting Diagnostic Labels to "Wayfinding". *JAMA, 326*(24), 2467–2468. doi:10.1001/jama.2021.22396

Alva,M. L. (2020). Co-occurrence of diabetes and depression in the U.S. *PloS one, 15*(6), 1-10. doi:10.1371/journal.pone.0234718

Ambinder,E. P. (2005). Electronic health records. *Journal of oncology practice, 1*(2), 57–63. doi:10.1200/JOP.2005.1.2.57

Ávila-Tomás,J. F., Mayer-Pujadas, M. A., & Quesada-Varela, V. J. (2020). La inteligencia artificial y sus aplicaciones en medicina II: importancia actual y aplicaciones prácticas. *Atención Primaria, 53*(1), 81-88. doi:10.1016/j.aprim.2020.04.014

Bermúdez Silva,F. J., & Romero Zerbo, S. Y. (2023). Cannabis-cannabinoides-y-diabetes-¿amigos-o-enemigos. *Diabetes,* 1-6.

Blanco Naranjo,E. G., Chavarría Campos, G. F., & Garita Fallas, Y. M. (2021). Estilo de vida saludable en diabetes mellitus tipo 2. *Revista Médica Sinergia, 6*(2), 1-10. doi:10.31434/rms.v6i2.639

Campillo Piqueras, A. (2024a), TFM_PrediDia. https://github.com/AndreaCampillo/TFM_PrediDia

Campillo Piqueras, A. (2024b), TFM_PrediDia_brfss. https://github.com/AndreaCampillo/TFM_PrediDia/tree/brfss

Campillo Piqueras, A. (2024c), TFM_PrediDia_Datasets. https://github.com/AndreaCampillo/TFM_PrediDia/tree/Datasets

Campillo Piqueras, A. (2024d), TFM_PrediDia_Notebooks. https://github.com/AndreaCampillo/TFM_PrediDia/tree/Jupyter_Notebooks

Campillo Piqueras, A. (2024e), TFM_PrediDia_Scripts. https://github.com/AndreaCampillo/TFM_PrediDia/tree/Scripts

Canadian Primary Care Sentinel Surveillance Network. (n.d). (CPCSSN). https://cpcssn.ca/

Centers for Disease Control and Prevention. (2014, Mayo, 16). About BRFSS. CDC. https://www.cdc.gov/brfss/about/index.htm

Centers for Disease Control and Prevention. (2022, Octubre, 27). Behavioral Risk Factor Surveillance System. BRFSS FAQs. https://www.cdc.gov/brfss/about/brfss_faq.htm#print

Centers for Disease Control and Prevention. (2023a, Julio, 21). 2021 Data. https://www.cdc.gov/brfss/annual_data/annual_2021.html

Centers for Disease Control and Prevention. (2023b, Diciembre, 2). 2022 Data. https://www.cdc.gov/brfss/annual_data/annual_2022.html

Centers for Disease Control and Prevention. (2023c, Mayo, 31). About NHANES. CDC. https://www.cdc.gov/nchs/nhanes/about_nhanes.htm

Centers for Disease Control and Prevention. (2023d, Agosto, 30). Survey Data and Documentation. https://www.cdc.gov/brfss/data_documentation/index.htm#print

Centers for Disease Control and Prevention. (2024a, Mayo, 17). Behavioral Risk Factor Surveillance System. https://www.cdc.gov/brfss/

Centers for Disease Control and Prevention. (2024b, Abril, 19). Factores de riesgo de la diabetes. https://www.cdc.gov/diabetes/es/risk-factors/factores-de-riesgo-de-la-diabetes.html?CDC_AAref_Val=https://www.cdc.gov/diabetes/spanish/basics/risk-factors.html

Chipia Lobo,J. F. (2020). BIG DATA: UTILIDAD EN CIENCIAS DE LA SALUD. *Revista GICOS, 5*(1), 1-2.

Clore, J., Cios, K., DeShazo, J. & Strack, B. (2014), Diabetes 130-US Hospitals for Years 1999-2008.

Columbié,Y. L., Enrique, P., Rivas Vázquez, D., & Borrego, Y. (2016). Factores de riesgo asociados con la aparición de diabetes mellitus tipo 2 en personas adultas. *Revista Cubana de Endocrinología, 27*(2), 123-133.

Deepa,R., & Sivasamy, A. (2023). Advancements in early detection of diabetes and diabetic retinopathy screening using artificial intelligence. *AIP Advances, 13*(11), 1-9. doi:10.1063/5.0172226

Dieuzeide,G., Waitman, J., Pugnaloni Rodríguez, N. S., Rodríguez, M. V., Nardone, L., Oviedo, A., Representación, E. N., Grupo De, D., De, I., Del, A., & Capture, E. (2022). ESTUDIO CAPTURE: RESULTADOS ARGENTINOS SOBRE PREVALENCIA DE ENFERMEDAD CARDIOVASCULAR EN DIABETES MELLITUS TIPO 2 [CAPTURE Study: Argentine results on prevalence of cardiovascular disease in type 2 diabetes mellitus]. *Medicina, 82*(2), 398-407.

Ellahham,S. (2020). Artificial Intelligence: The Future for Diabetes Care. *The American journal of medicine, 133*(8), 895-900. doi:10.1016/j.amjmed.2020.03.033

Eskandar, S. (2023, Marzo, 16). Introduction to RBF SVM: A Powerful Machine Learning Algorithm for Non-Linear Data. https://medium.com/@eskandar.sahel/introduction-to-rbf-svm-a-powerful-machine-learning-algorithm-for-non-linear-data-1d1cfb55a1a

Faruque,M. F., Sarker, I. H., & Asaduzzaman (2019). Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus. I*nternational Conference on Electrical, Computer and Communication Engineering (ECCE),* 1-4. doi:10.1109/ECACE.2019.8679365

Fiallos,P. C., Castillo Vera, L. J., Evelyn, K., Cudco, C., & Parra Cazar, T. H. (2019). Relación entre enfermedades reumáticas y diabetes mellitus. Relationship between rheumatic diseases and diabetes mellitus. *Revista Cubana de Reumatología, 21*(3), 1-10.

Fuentes,B., Amaro, S., Alonso De Leciñana, M., Arenillas, J. F., Ayo-Martín, O., Castellanos, M., Freijo, M., García-Pastor, A., Gomis, M., Gómez Choco, M., López-Cancio, E., Martínez Sánchez, P., Morales, A., Palacio-Portilla, E. J., Rodríguez-Yáñez, M., Roquer, J., Segura, T., Serena, J., & Vivancos-Mora, J. (2021). Prevención de ictus en pacientes con diabetes mellitus tipo 2 o prediabetes. Recomendaciones del Grupo de Estudio de Enfermedades Cerebrovasculares de la Sociedad Española de Neurología. *Neurología, 36*(4), 305-323. doi:10.1016/j.nrl.2020.04.030

Gálvez Galán,I., Cáceres León, M. C., Guerrero-Martín, J., López Jurado, C. F., & Durán-Gómez, N. (2021). Calidad de vida relacionada con la salud en pacientes con diabetes mellitus en una zona básica de salud. *Enfermería Clínica, 31*(5), 313-322. doi:10.1016/j.enfcli.2021.03.001

García-Ordás,M. T., Benavides, C., Benítez-Andrades, J. A., Alaiz-Moretón, H., & García-Rodríguez, I. (2021). Diabetes detection using deep learning techniques with oversampling and feature augmentation. *Computer Methods and Programs in Biomedicine, 202* doi:10.1016/j.cmpb.2021.105968

Garmendia-Lorena,F., Peruana, A. M., & De Revisión, A. (2021). Situación actual de la prevención en la diabetes mellitus tipo 2. *Acta Médica Peruana, 39*(1), 51-58. doi:10.35663/amp.2022.391.2162

Harreiter,J., & Roden, M. (2023). Diabetes mellitus: definition, classification, diagnosis, screening and prevention. *Wiener klinische Wochenschrift, 135*(S1), 7–17. doi:10.1007/s00508-022-02122-y

Healthline. (2022, Enero). Diabetes y depresión ¿Cuál es el vínculo? https://www.healthline.com/health/es/diabetes-y-depresion#pronostico

Hussein Mohamed,M., Helmy Khafagy, M., Mohamed, N., & Kamel, M. (2024). DIABETIC MELLITUS PREDICTION WITH BRFSS DATA SETS. *Journal of Theoretical and Applied Information Technology, 102*(3), 883-897.

International Diabetes Federation. (2021) Global and regional diabetes data. *International Diabetes Federation.* https://diabetesatlas.org/data/en/world/

Iparraguirre-Villanueva,O., Espinola-Linares, K., Flores Castañeda, R. O., & Cabanillas-Carbonell, M. (2023). Application of Machine Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes. *Diagnostics, 13*(14), 1-16. doi:10.3390/diagnostics13142383

Islam Ayon,S., & Milon Islam, M. D. (2019). Diabetes Prediction: A Deep Learning Approach. *I.J. Information Engineering and Electronic Business, 11*(2), 21-27. doi:10.5815/ijieeb.2019.02.03

Khan,A., Petropoulos, I. N., Ponirakis, G., & Malik, R. A. (2017). Visual complications in diabetes mellitus: beyond retinopathy. *Diabetic medicine : a journal of the British Diabetic Association, 34*(4), 478–484. doi:10.1111/dme.13296

Khateeb,J., Fuchs, E., & Khamaisi, M. (2019). Diabetes and Lung Disease: An Underestimated Relationship. *The review of diabetic studies : RDS, 15*(1), 1–15. doi:10.1900/rds.2019.15.1

Kposowa,A. J., Aly Ezzat, D., & Breault, K. (2021). Diabetes Mellitus and Marital Status: Evidence from the National Longitudinal Mortality Study on the Effect of Marital Dissolution and the Death of a Spouse. *International journal of general medicine, 14*(1), 1881–1888. doi:10.2147/ijgm.s307436

Kumar,Y., Koul, A., Singla, R., & Ijaz, M. F. (2023). Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. *Journal of ambient intelligence and humanized computing, 14*(7), 8459–8486. doi:10.1007/s12652-021-03612-z

Lai,H., Huang, H., Keshavjee, K., Guergachi, A., & Gao, X. (2019). Predictive models for diabetes mellitus using machine learning techniques. *BMC Endocr Disord, 19*(1), 1-9. doi:10.1186/s12902-019-0436-6

Lanzagorta-Ortega,D., Carrillo-Pérez, D. L., & Carrillo-Esper, R. (2023). Inteligencia artificial en medicina: presente y futuro. *GACETA MÉDICA DE MÉXICO, 158*(1), 17-21. doi:10.24875/gmm.m22000688

Leyva, I. (2023, Abril, 28). Cuánto necesitas ganar para pertenecer a la clase baja, media y alta en EEUU. https://es-us.finanzas.yahoo.com/noticias/cuanto-necesitas-ganar-para-pertenecer-a-la-clase-baja-media-y-alta-en-eeuu-120040947.html?guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmVzLw&guce_referrer_sig=AQAAAEYVDByz8uTo1Ht65NLBoSDV3cXJMz7DSzPN0stCg70GXko4WA43XnXygZK4FO4wNUaPZDsYRADKSLOqG5eLfkwxtfbzk0IhhIyejjSoqvyN6rX8PJRFYoqW5WkJyKXMGG4rkmOVbuYH5RxGdbcnk21biHIFUlX3vmfGWGccmaNJ&_guc_consent_skip=1717609278

La Licencia MIT. (2024, Abril, 10). https://es.wikipedia.org/wiki/Licencia_MIT

Lovic,D., Piperidou, A., Zografou, I., Grassos, H., Pittaras, A., & Manolis, A. (2020). The Growing Epidemic of Diabetes Mellitus. *Current vascular pharmacology, 18*(2), 104–109. doi:10.2174/1570161117666190405165911

Manzini,E., Altirriba, M. B., & Perera Lluna, A. (2022). ¿Big data para la diabetes mellitus? *Habilidades prácticas, 13*(4), 141-172. doi:10.52102/diabet/pract/2022.4/art3

Martínez Candela,J. (2015). ¿Cuáles son los factores de riesgo para desarrollar diabetes mellitus tipo 2? *Preguntas clínicas redGDPS,* 16-18.

Martínez-Vasallo,B., Méndez-Macón, Y., & Valdez-Gasmuri, I. (2021). Factores de riesgo asociados a diabetes mellitus tipo 2. Policlínico Docente José Jacinto Milanés. Matanzas, 2019. *Revista Médica Electrónica, 43*(6), 1534-1546.

Matar-Khalil,S. R., & Rubio-Sandoval, F. C. (2021). El deterioro cognitivo como una complicación de la Diabetes Mellitus Tipo 2. *Nova, 19*(37), 25-41. doi:10.22490/24629448.5473

Narayan,K. M. V., Chan, J., & Mohan, V. (2011). Early Identification of Type 2 Diabetes: policy should be aligned with health systems strengthening. *34*(1), 244–246. doi:10.2337/dc10-1952

Noble,D., Mathur, R., Dent, T., Meads, C., & Greenhalgh, T. (2011). Risk models and scores for type 2 diabetes: systematic review. *BMJ (Clinical research), 343*(1), 1-31. doi:10.1136/bmj.d7163

Orellana-Suarez,K. D., Álava-Vélez, G. A., & Medina-Solís, K. B. (2024). Caracterización epidemiológica y diagnóstico de laboratorio de las nefropatías en pacientes con diabetes mellitus. *MQRInvestigar, 8*(1), 2554–2573. doi:10.56048/mqr20225.8.1.2024.2554-2573

Pastorino,R., De Vito, C., Migliara, G., Glocker, K., Binenbaum, I., Ricciardi, W., & Boccia, S. (2019). Benefits and challenges of Big Data in healthcare: an overview of the European initiatives. *European journal of public health, 29*(3), 23–27. doi:10.1093/eurpub/ckz168

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É. (2024) scikit-learn 1.4.2. https://scikit-learn.org/1.4/modules/classes.html

Petersmann,A., Müller-Wieland, D., Müller, U. A., Landgraf, R., Nauck, M., Freckmann, G., Heinemann, L., & Schleicher, E. (2019). Definition, Classification and Diagnosis of Diabetes Mellitus. *Experimental and clinical endocrinology & diabetes : official journal, German Society of Endocrinology [and] German Diabetes Association, 127*(S 01), S1–S7. doi:10.1055/a-1018-9078

Pima Indians Diabetes Database. (2016) https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data?select=diabetes.csv

Quora. (n.d). Does the order of training data matter when training neural networks? https://www.quora.com/Does-the-order-of-training-data-matter-when-training-neural-networks

Ruibal-Tavares,E., René, J., Calleja-López, T., Cristián, N., Rivera-Rosas, & Aguilera-Duarte, L. J. (2023). INTELIGENCIA ARTIFICIAL EN MEDICINA: PANORAMA ACTUAL. *REMUS - Revista Estudiantil de Medicina de la Universidad de Sonora,* (10), 21-31. doi:10.59420/remus.10.2023.178

Sánchez Rosado,J. C., & Díez Parra, M. (2022). Impacto de la Inteligencia Artificial en la Transformación de la Sanidad: Beneficios y Retos. (424), 129-144.

Seiglie,J. A., Marcus, M. E., Ebert, C., Prodromidis, N., Geldsetzer, P., Theilmann, M., Agoudavi, K., Andall-Brereton, G., Aryal, K. K., Bicaba, B. W., Bovet, P., Brian, G., Dorobantu, M., Gathecha, G., Gurung, M. S., Guwatudde, D., Msaidié, M., Houehanou, C., Houinato, D., . . . Manne-Goehler, J. (202). Diabetes Prevalence and Its Relationship With Education, Wealth, and BMI in 29 Low-and Middle-Income Countries. *Diabetes care, 43*(4), 767–775. doi:10.2337/dc19-1782

Song,P., & Zou, M. (2017). Electronic Cigarettes, Diabetes, and Cardiovascular Disease.

Soto,I. N. (2017). Tabaquismo y Diabetes. *Revista chilena de enfermedades respiratorias, 33*(3), 222-224. doi:10.4067/s0717-73482017000300222.

Subrahmanya,S. V. G., Shetty, D. K., Patil, V., Hameed, B. M. Z., Paul, R., Smriti, K., Naik, N., & Somani, B. K. (2021). The role of data science in healthcare advancements: applications, benefits, and future prospects. *Irish journal of medical science, 191*(4), 1473–1483. doi:10.1007/s11845-021-02730-z

Tasin,I., Nabil, T. U., Islam, S., & Khan, R. (2022). Diabetes prediction using machine learning and explainable AI techniques. *Healthcare Technology Letters, 10*(1), 1-10. doi:10.1049/htl2.12039

Ullah,Z., Saleem, F., Jamjoom, M., Fakieh, B., Kateb, F., Ali, A. M., & Shah, B. (2022). Detecting High-Risk Factors and Early Diagnosis of Diabetes Using Machine Learning Methods. *Computational intelligence and neuroscience,* 1-10. doi:10.1155/2022/2557795

Vasudha,V., Vasavi, G., & Kiran Kumar, K. R. N. (2021). SIGNIFICANCE OF MULTILAYER PERCEPTRON MODEL FOR EARLY DETECTION OF DIABETES OVER ML METHODS. *Journal of University of Shanghai for Science and Technology, 23*(8), 148-160.

Veleiro,I. N. (2023). Consumo de alcohol y Diabetes Mellitus. *Diabetes.*

Vizzuett Montoya,A. R., & López-García, M. D. C. (2021). Uso del test FINDRISC para el tamizaje de Diabetes Mellitus tipo 2 en salud ocupacional. *Revista Colombiana de Salud Ocupacional, 10*(1), 6419-6419. doi:10.18041/2322-634x/rcso.1.2020.6419

Wee,B. F., Sivakumar, S., Lim, K. H., Wong, W. K., & Juwono, F. H. (2024). Diabetes detection based on machine learning and deep learning approaches. *Multimed Tools Appl, 83*(8), 24153–24185. doi:10.1007/s11042-023-16407-5

Woods,N. P., Tangpukdee, J., Thepa, T., & Methakanchanasak, N. (2023). Consequences of Sleep Deprivation in Adult Diabetes Mellitus Type 2 Patients: An Integrative Review. *Open Access Macedonian Journal of Medical Sciences, 11*(F), 1-10. doi:10.3889/oamjms.2023.10029

Woon,L. S., Sidi, H. B., Ravindran, A., Gosse, P. J., Mainland, R. L., Kaunismaa, E. S., Hatta, N. H., Arnawati, P., Zulkifli, A. Y., Mustafa, N., & Leong Bin Abdullah, M. F. I. (2020). Depression, anxiety, and associated factors in patients with diabetes: evidence from the anxiety, depression, and personality traits in diabetes mellitus (ADAPT-DM) study. *BMC psychiatry, 20*(1), 1-14. doi:10.1186/s12888-020-02615-y

Yang,H., Luo, Y., Ren, X., Wu, M., He, X., Peng, B., Deng, K., Yan, D., Tang, H., & Lin, H. (2021). Risk Prediction of Diabetes: Big data mining with fusion of multifarious physical examination indicators. *Information Fusion, 75*140-149. doi:10.1016/j.inffus.2021.02.015

Zhang,Z., Asif, K., Hasan, Gedeon, T., & Hossain, M. Z. (2024). A Deep Learning Approach to Diabetes Diagnosis. *arXiv,* 1-13. doi:10.48550/arXiv.2403.07483

Anexo A. Estructura de los atributos de la encuesta

Tabla 3. *Variables, posición y longitud en el fichero ASCII.*

Atributo DataSet	Calculado	Campo BRFSS 2021	Columna BRFSS 2021	Campo BRFSS 2022	Columna BRFSS 2022	Longitud (Caracteres
Year	Sí	Analista	0	Analista	0	4
CatBMI	Sí	_BMI5CAT	2004	_BMI5CAT	2002	1
Stroke	No	CVDSTRK3	121	CVDSTRK3	120	1
HeartDis	Sí	_MICHD	1908	_MICHD	1908	1
PhysExer	Sí	_TOTINDA	1904	_TOTINDA	1904	1
HealthIns	Sí	_HLTHPLN	1902	_HLTHPLN	1902	1
NoMedCost	No	MEDCOST1	111	MEDCOST1	111	1
GenHealth	Sí	_RFHLTH	1899	_RFHLTH	1899	1
CogDiff	No	DECIDE	206	DECIDE	199	1
DepresSíon	No	ADDEPEV3	127	ADDEPEV3	126	1
MentalHlth	Sí	_MENT14D	1901	_MENT14D	1901	1
MentalState	Sí	Analista	0	Analista	0	1
PhysHlth	Sí	_PHYS14D	1900	_PHYS14D	1900	1
WalkDiff	No	DIFFWALK	207	DIFFWALK	200	1
Gender	Sí	_SEX	1982	_SEX	1980	1
AgeRange	Sí	_AGE_G	1988	_AGE_G	1986	1
EdLevel	Sí	_EDUCAG	2007	_EDUCAG	2005	1
AnnIncome	No	INCOME3	193	INCOME3	186	2
SocClass	Sí	Analista	0	Analista	0	1
UrologyDz	No	CHCKDNY2	128	CHCKDNY2	127	1
ViSíonDiff	No	BLIND	205	BLIND	198	1

Atributo DataSet	Calculado	Campo BRFSS 2021	Columna BRFSS 2021	Campo BRFSS 2022	Columna BRFSS 2022	Longitud (Caracteres
Asthma	Sí	_LTASTH1	1909	_LTASTH1	1909	1
LungDiseases	No	CHCCOPD3	126	CHCCOPD3	125	1
Arthritis	Sí	_DRDXAR3	1912	_DRDXAR2	1912	1
SmokerTrad	Sí	_SMOKER3	2009	_SMOKER3	2019	1
ECigSmok	Sí	_CURECI1	2011	_CURECI2	2021	1
AlcDrinker	Sí	_RFBING5	2016	_RFBING6	2042	1
Race	Sí	_RACEGR3	1980	_RACEGR4	1978	1
MaritalSt	No	MARITAL	175	MARITAL	168	1
LastMedChk	No	CHECKUP1	112	CHECKUP1	112	1
Awareness	No	DIABEDU	275	DIABEDU1	277	1
FootIrrita	No	FEETCHK3	264	No existe	0	3
FecFootIrrita	Sí	Analista	0	FEETSORE	278	1
HighBP	No	BPHIGH6	114	No existe	0	1
HighChol	No	TOLDHI3	117	No existe	0	1
FruitCons	Sí	_FRTLT1A	2066	No existe	0	1
VegCons	Sí	_VEGLT1A	2067	No existe	0	1
FruitOrVegCon	Sí	Analista	0	No existe	0	1
FruitAndVegCon	Sí	Analista	0	No existe	0	1
MarijuanaCon	No	No existe	0	MARIJAN1	371	2
SleepHours	No	No existe	0	SLEPTIM1	114	2
BrDiabetes	No	DIABETE4	129	DIABETE4	129	1
GrDiabetes	Sí	Analista	0	No existe	0	1
SupGrPreDiabetes	Sí	Analista	0	No existe	0	1
SupGrNoPreDiabetes	Sí	Analista	0	No existe	0	1

Fuente: Elaboración propia

Anexo B. Valores de los atributos

Tabla 4. *Valores atributos.*

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
Year	Sí	Analista	Año de obtención	No existe, calculado por Analista	2021 - Año encuesta 2021 2022 - Año encuesta 2022
CatBMI	Sí	brfss_21_22	Categoría índice de masa corporal	1 - Bajo peso 2 - Peso normal 3 - Sobrepeso 4 - Obeso . - No sabe, no contesta, dato omitido	1 - Bajo peso 2 - Peso normal 3 - Sobrepeso 4 - Obeso 9 - No sabe, no contesta, dato omitido
Stroke	No	brfss_21_22	Derrame Cerebral	1 - Sí tuvo derrame cerebral 2 - No tuvo derrame cerebral 7 - No sabe 9 - No contesta	1 - Sí tuvo derrame cerebral 2 - No tuvo derrame cerebral 9 – No sabe, no contesta
HeartDis	Sí	brfss_21_22	Enfermedades coronarias	1 - Sí tiene enfermedades cardiacas 2 - No tiene enfermedades cardiacas . - No sabe, no contesta, dato omitido	1 - Sí tiene enfermedades cardiacas 2 - No tiene enfermedades cardiacas 9- No sabe, no contesta, dato omitido
PhysExer	Sí	brfss_21_22	Ejercicio físico	1 - Sí hace ejercicio 2 - No hace ejercicio 9 - No sabe, no contesta, dato omitido	1 - Sí hace ejercicio 2 - No hace ejercicio 9 - No sabe, no contesta, dato omitido
HealthIns	Sí	brfss_21_22	Tiene Seguro de Salud	1 - Sí tiene seguro de salud 2 - No tiene seguro de salud 9 - No sabe, no contesta, dato omitido	1 - Sí tiene seguro de salud 2 - No tiene seguro de salud 9 - No sabe, no contesta, dato omitido
NoMedCost	No	brfss_21_22	No se puede permitir ir al medico por costes	1 - Sí se puede permitir gastos en salud 2 - No se puede permitir gastos en salud 7 - No sabe 9 - No contesta, dato omitido	1 - Sí se puede permitir gastos en salud 2 - No se puede permitir gastos en salud 9 - No sabe, no contesta, dato omitido
GenHealth	Sí	brfss_21_22	Estado general de salud	1 - Buen estado de salud 2 - Mal o pobre estado de salud 9 - No sabe, no contesta, dato omitido	1 - Buen estado de salud 2 - Mal o pobre estado de salud 9 - No sabe, no contesta, dato omitido
CogDiff	No	brfss_21_22	Dificultades cognitivas	1 - Sí tiene problemas cognitivos 2 - No tiene problemas cognitivos 7 - No sabe 9 - No contesta, dato omitido	1 - Sí tiene problemas cognitivos 2 - No tiene problemas cognitivos 9 - No sabe, no contesta, caracteres no esperados
Depression	No	brfss_21_22	DepreSión	1 - Sí tiene depresión 2 - No tiene depresión 7 - No sabe 9 - No contesta, dato omitido	1 - Sí Tiene depresión 2 - No tiene depresión 9 - No sabe, no contesta, caracteres no esperados
MentalHlth	Sí	brfss_21_22	Salud Mental	1 - Buen estado mental 2 - Mal estado mental durante trece días 3 - Mal estado mental durante treinta días o más 9 - No sabe, no contesta, dato omitido	1 - Buen estado mental 2 - Mal estado mental durante trece días 3 - Mal estado mental durante treinta días o más 9 - No sabe, no contesta, dato omitido
MentalState	Sí	Analista	Estado Mental	No existe, calculado por Analista	1 - Buena salud mental 2 - Mala salud mental 9 - No sabe, no contesta, caracteres no esperados

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
PhysHlth	Sí	brfss_21_22	Estado físico	1 - Buen estado físico 2 - Mal estado físico durante trece días 3 - Mal estado físico durante mas de 14 días 9 - No sabe, no contesta, dato omitido	1 - Buen estado físico 2 - Mal estado físico durante trece días 3 - Mal estado físico durante mas de 14 días 9 - No sabe, no contesta, dato omitido
WalkDiff	No	brfss_21_22	Dificultad de caminar	1 - Sí tiene dificultad para caminar 2 - No tiene dificultad para caminar 7 - No sabe 9 - No contesta, dato omitido	1 - Sí tiene dificultad para caminar 2 - No tiene dificultad para caminar 9 - No sabe, no contesta, dato omitido
Gender	Sí	brfss_21_22	Sexo	1 - Masculino 2 - Femenino	1 - Masculino 2 - Femenino 9 - No sabe, no contesta, dato omitido
AgeRange	Sí	brfss_21_22	Rangos de Edad	1 - 18 hasta 24 años 2 - 25 hasta 34 años 3 - 35 hasta 44 años 4 - 45 hasta 54 años 5 - 55 hasta 64 años 6 - Más de 65 años	1 - 18 hasta 24 años 2 - 25 hasta 34 años 3 - 35 hasta 44 años 4 - 45 hasta 54 años 5 - 55 hasta 64 años 6 - Más de 65 años 9 - No sabe, no contesta, dato omitido
EdLevel	Sí	brfss_21_22	educación Completada	1 - Sin bachillerato 2 - Con bachillerato 3 - Empezó estudios superiores 4 - Terminó estudios superiores 9 - No sabe, no contesta, dato omitido	1 - Sin bachillerato 2 - Con bachillerato 3 - Empezó estudios superiores 4 - Terminó estudios superiores 9 - No sabe, no contesta, dato omitido

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
AnnIncome	No	brfss_21_22	Salario anual	Rangos de salario ($): 01 - Menos de 10.000 02 - 10.000:15.000 03 - 15.000:20.000 04 - 20.000:25.000 05 - 25.000:35.000 06 - 35.000:50.000 07 - 50.000:75.000 08 - 75.000:100.000 09 - 100.000:150.000 10 - 150.000:200.000 11 - Más de 200.00 77 - No sabe 99 - No contesta, dato omitido	Rangos de salario ($): 01 - Menos de 10.000 02 - 10.000:15.000 03 - 15.000:20.000 04 - 20.000:25.000 05 - 25.000:35.000 06 - 35.000:50.000 07 - 50.000:75.000 08 - 75.000:100.000 09 - 100.000:150.000 10 - 150.000:200.000 11 - Más de 200.00 99 - No sabe, no contesta, dato omitido
SocClass	Sí	Analista	Clase Social	No existe, calculado por Analista	1 - Clase Baja 2 - Clase Media 3 - Clase Alta 9- No sabe, no contesta, caracteres no esperados
UrologyDz	No	brfss_21_22	Enferemedades urologicas	1 - Sí tiene enfermedades urológicas 2 - No tiene enfermedades urológicas 7 - No sabe 9 - No contesta, dato omitido	1 - Sí tiene enfermedades urológicas 2 - No tiene enfermedades urológicas 9 - No sabe, no contesta, dato omitido

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
ViSionDiff	No	brfss_21_22	Dificultad de viSión	1 - Sí tiene dificultades de visión 2 - No tiene dificultades de visión 7 - No sabe 9 - No contesta, dato omitido	1 - Sí tiene dificultades de visión 2 - No tiene dificultades de visión 9 - No sabe, no contesta, dato omitido
Asthma	Sí	brfss_21_22	Asma	1 - No tiene asma 2 - Sí tiene asma 9 - No sabe, no contesta, dato omitido	1 - No tiene asma 2 - Sí tiene asma 9 - No sabe, no contesta, dato omitido
LungDiseases	No	brfss_21_22	Enfermedades pulmonares	1 - Sí tiene enfermedades respiratorias 2 - No tiene enfermedades respiratorias 7 - No sabe 9 - No contesta, dato omitido	1 - Sí tiene enfermedades respiratorias 2 - No tiene enfermedades respiratorias 9 - No sabe, no contesta, dato omitido
Arthritis	Sí	brfss_21_22	Artritis	1 - Sí tiene artritis 2 - No tiene artritis . - No sabe, no contesta, dato omitido	1 - Sí tiene artritis 2 - No tiene artritis 9 - No sabe, no contesta, dato omitido
SmokerTrad	Sí	brfss_21_22	Tabaquismo	1 - Fumador habitual 2 -Fumador ocasional 3 - Exfumador 4 - No fumador 9 - No sabe, no contesta, dato omitido	1 - Fumador habitual 2 -Fumador ocasional 3 - Exfumador 4 - No fumador 9 - No sabe, no contesta, dato omitido
ECigSmok	Sí	brfss_21_22	Fumador e-Cigarrillos	1 - No es fumador de eCigarrillos 2 - Sí es fumador de eCigarrillos 9 - No sabe, no contesta, dato omitido	1 - No es fumador de eCigarrillos 2 - Sí es fumador de eCigarrillos 9 - No sabe, no contesta, dato omitido

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
AlcDrinker	Sí	brfss_21_22	Alcoholismo	1 - No es bebedor 2 - Sí es bebedor 9 - No sabe, no contesta, dato omitido	1 - No es bebedor 2 - Sí es bebedor 9 - No sabe, no contesta, dato omitido
Race	Sí	brfss_21_22	Raza	1 - Blanca, no hispano 2 - Negro, no hispano 3 - Otras razas, no hispanos 4 - Multirracial, no hispano 5 - Hispano 9 - No sabe, no contesta, dato omitido	1 - Blanca, no hispano 2 - Negro, no hispano 3 - Otras razas, no hispanos 4 - Multirracial, no hispano 5 - Hispano 9 - No sabe, no contesta, dato omitido
MaritalSt	No	brfss_21_22	Estado Civil	1 - Casado 2 - Divorciado 3 - Viudo 4 - Separado 5 - Soltero 6 - Pareja de hecho 9 - No sabe, no contesta, dato omitido	1 - Casado 2 - Divorciado 3 - Viudo 4 - Separado 5 - Soltero 6 - Pareja de hecho 9 - No sabe, no contesta, dato omitido
LastMedChk	No	brfss_21_22	Ultimo chequeo medico	1 - Paso reconocimiento médico menos de 1 año 2 - Paso reconocimiento médico entre 1 y 2 años 3 - Paso reconocimiento médico entre 2 y 5 años 4 - Paso reconocimiento médico más 5 años 7 - No sabe 8 - Nunca paso un reconocimiento médico 9 - No contesta, dato omitido	1 - Paso reconocimiento médico menos de 1 año 2 - Paso reconocimiento médico entre 1 y 2 años 3 - Paso reconocimiento médico entre 2 y 5 años 4 - Paso reconocimiento médico más 5 años 5 - Nunca paso un reconocimiento médico 9 - No sabe, no contesta, dato omitido

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
Awareness	No	brfss_21_22	Curso sobre Diabetes	1 - Sí asistió a cursos de concienciación 2 - No asistió a cursos de concienciación 7 - No sabe 9 - No contesta, dato omitido	1 - Sí asistió a cursos de concienciación 2 - No asistió a cursos de concienciación 9 - No sabe, no contesta, dato omitido
FootIrrita	No	brfss_21_22	Irritaciones o llagas en los pies	1xx - Veces revisa sus pies veces por día 2xx - Veces revisa sus pies veces por semana 3xx - - Veces revisa sus pies veces por mes 4xx - Veces revisa sus pies veces por año 555 - No tiene pies 888 - - Nunca revisa sus pies 777 - No sabe 999 - No contesta, dato omitido	1xx - Veces revisa sus pies veces por día 2xx - Veces revisa sus pies veces por semana 3xx - - Veces revisa sus pies veces por mes 4xx - Veces revisa sus pies veces por año 555 - No tiene pies 888 - - Nunca revisa sus pies 999 - No sabe, no contesta, dato omitido
FecFootIrrita	Sí	Analista	Tiene irritaciones o llagas en los pies	No existe, calculado por Analista	1 - Tiene irritaciones o llagas en los pies 2 - No tiene irritaciones ni llagas en los pies 9 -No tiene pies, no sabe, no contesta, dato omitido
HighBP	No	brfss_21	Tensión Alta	1 - Sí tiene tensión alta 2- Sí tuvo tensión alta durante el embarazo 3 - Tiene tensión normal 4 - Está en el límite tensión normal 7- No sabe 9 - No contesta	1 - Sí tiene tensión alta (se incluye límite tensión normal) 2 - Tiene tensión normal (tensión alta durante el embarazo se considera tensión normal) 9 - No sabe, no contesta, caracteres inesperados

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
HighChol	No	brfss_21	Colesterol Alto	1 - No tiene colesterol 2 - Sí tiene colesterol 7 - No sabe 9 - No contesta, dato omitido	1 - No tiene colesterol 2 - Sí tiene colesterol 9 - No sabe, no contesta, dato omitido
FruitCons	Sí	brfss_21	Consume fruta	1 - Sí consume fruta 2 - No consume fruta 9 - No sabe, no contesta, dato omitido	1 - Sí consume fruta 2 - No consume fruta 9 - No sabe, no contesta, dato omitido
VegCons	Sí	brfss_21	Consume Vegetales	1 - Sí consume vegetales 2 - No consume vegetales 9 - No sabe, no contesta, dato omitido	1 - Sí consume vegetales 2 - No consume vegetales 9 - No sabe, no contesta, dato omitido
FruitOrVegCon	Sí	Analista	Consume Vegetales o Fruta	No existe, calculado por Analista	1 - Sí come vegetales o fruta 2 - No come ni vegetales ni fruta 9 - No sabe, no contesta, caracteres inesperados
FruitAndVegCon	Sí	Analista	Consume Vegetales y Fruta	No existe, calculado por Analista	1 - Sí come vegetales y fruta 2 - No como vegetales o fruta o ninguna de los dos 9 - No sabe, no contesta, caracteres inesperados
MarijuanaCon	No	brfss_22	Consumo de marihuana	01-30 - Días consume marihuana 88 - No consume marihuana 77 - No sabe, no está seguro 99 - No contesta, dato omitido	1 - Sí consume 2 - No consume 9 - No sabe, no contesta, caracteres inesperados
SleepHours	No	brfss_22	Horas de sueño	01-24 - Horas duerme al día 77 - No sabe, no está seguro 99 - No contesta, dato omitido	01-24 - Horas duerme al día 99 - No sabe, no contesta, dato omitido

Atributo DataSet	Calcul.	Origen Dato	Concepto	Valores BRFSS	Valores Analista
BrDiabetes	No	brfss_21_22	Diabetes	1 - Sí tiene Diabetes 2 - Sí tuvo Diabetes, pero solo en el embarazo 3 - No tiene Diabetes 4 - Sí tiene Prediabetes 7 - No sabe 9 - No contesta, dato omitido	1 - Sí tiene Diabetes 2 - Sí tuvo Diabetes, pero solo en el embarazo 3 - No tiene Diabetes 4 - Sí tiene Prediabetes 7 - No sabe 9 - No contesta, dato omitido
GrDiabetes	Sí	Analista	Grupo Diabetes	No existe, calculado por Analista	1 - Sí tiene Diabetes 2 - Sí tiene Prediabetes 3 - No tiene Diabetes (Diabetes durante el embarazo se considera no tiene Diabetes) 9 - No sabe, no contesta, dato omitido
SupGrPreDiabetes	Sí	Analista	Super Grupo Diabetes	No existe, calculado por Analista	1 - Sí tiene Diabetes (Incluye prediabetes) 2 - No tiene Diabetes (Diabetes durante el embarazo se considera no tiene Diabetes) 9 - No sabe, no contesta, dato omitido
SupGrNoPreDiabetes	Sí	Analista	Super Grupo Diabetes	No existe, calculado por Analista	1 - Sí tiene Diabetes 2 - No tiene Diabetes (Incluye prediabetes, Diabetes durante el embarazo se considera no tiene Diabetes) 9 - No sabe, no contesta, dato omitido

Fuente: Elaboración propia

Anexo C. Frecuencias de cada categoría

Frecuencias de cada categoría del dataset 2022

Figura 74 . *Frecuencias de cada categoría del dataset 2021 (I).*

```
Columna: Year
Year
2021    229655
Name: count, dtype: int64

Columna: CatBMI
CatBMI
3    82709
4    82360
2    61877
1     2709
Name: count, dtype: int64

Columna: Stroke
Stroke
2    220861
1      8794
Name: count, dtype: int64

Columna: HeartDis
HeartDis
2    210093
1     19562
Name: count, dtype: int64

Columna: PhysExer
PhysExer
1    179394
2     50261
Name: count, dtype: int64

Columna: HealthIns
HealthIns
1    221183
2      8472
Name: count, dtype: int64

Columna: NoMedCost
NoMedCost
2    215400
1     14255
Name: count, dtype: int64

Columna: GenHealth
GenHealth
1    194868
2     34787
Name: count, dtype: int64
```

```
Columna: CogDiff
CogDiff
2    207900
1     21755
Name: count, dtype: int64

Columna: Depression
Depression
2    183143
1     46512
Name: count, dtype: int64

Columna: MentalHlth
MentalHlth
1    143638
2     58890
3     27127
Name: count, dtype: int64

Columna: MentalState
MentalState
1    172213
2     57442
Name: count, dtype: int64

Columna: PhysHlth
PhysHlth
1    153140
2     50386
3     26129
Name: count, dtype: int64

Columna: WalkDiff
WalkDiff
2    195013
1     34642
Name: count, dtype: int64

Columna: Gender
Gender
2    120222
1    109433
Name: count, dtype: int64
```

```
Columna: AgeRange
AgeRange
6    82297
5    49473
4    38541
3    32037
2    20743
1     6564
Name: count, dtype: int64

Columna: EdLevel
EdLevel
4    105835
3     63276
2     51095
1      9449
Name: count, dtype: int64

Columna: AnnIncome
AnnIncome
7     40716
9     35372
8     34001
6     30944
5     26302
10    14920
11    14698
4     12158
3      8458
2      6541
1      5545
Name: count, dtype: int64

Columna: SocClass
SocClass
2    110089
1     89948
3     29618
Name: count, dtype: int64

Columna: UrologyDz
UrologyDz
2    220249
1      9406
Name: count, dtype: int64
```

```
Columna: VisionDiff
VisionDiff
2    219307
1     10348
Name: count, dtype: int64

Columna: Asthma
Asthma
1    197632
2     32023
Name: count, dtype: int64

Columna: LungDiseases
LungDiseases
2    211855
1     17800
Name: count, dtype: int64

Columna: Arthritis
Arthritis
2    149999
1     79656
Name: count, dtype: int64

Columna: SmokerTrad
SmokerTrad
4    135622
3     66101
1     20277
2      7655
Name: count, dtype: int64

Columna: ECigSmok
ECigSmok
1    220902
2      8753
Name: count, dtype: int64

Columna: AlcDrinker
AlcDrinker
1    198028
2     31627
Name: count, dtype: int64
```

Fuente: Elaboración propia

Figura 75 . *Frecuencias de cada categoría del dataset 2021 (II).*

```
Columna: Race
Race
1    178467
5     18345
2     16305
3     11818
4      4720
Name: count, dtype: int64

Columna: MaritalSt
MaritalSt
1    130912
5     32086
2     31096
3     22873
6      8568
4      4120
Name: count, dtype: int64

Columna: LastMedChk
LastMedChk
1    185185
2     26314
3     10915
4      6786
5       455
Name: count, dtype: int64

Columna: Awareness
Awareness
9    217107
1      7002
2      5546
Name: count, dtype: int64

Columna: FootIrrita
FootIrrita
999    217305
101      6222
888      1699
201      1103
102       545
       ...
191         1
106         1
424         1
299         1
213         1
Name: count, Length: 67, dtype: int64

Columna: FecFootIrrita
FecFootIrrita
9    217375
1     10581
2      1699
Name: count, dtype: int64

Columna: HighBP
HighBP
2    131281
1     98374
Name: count, dtype: int64

Columna: HighChol
HighChol
2    137531
1     92124
Name: count, dtype: int64

Columna: FruitCons
FruitCons
1    142907
2     86748
Name: count, dtype: int64

Columna: VegCons
VegCons
1    190337
2     39318
Name: count, dtype: int64

Columna: FruitOrVegCon
FruitOrVegCon
1    205771
2     23884
Name: count, dtype: int64

Columna: FruitAndVegCon
FruitAndVegCon
1    127473
2    102182
Name: count, dtype: int64

Columna: MarijuanaCon
MarijuanaCon
9    229655
Name: count, dtype: int64

Columna: SleepHours
SleepHours
99    229655
Name: count, dtype: int64

Columna: BrDiabetes
BrDiabetes
3    189865
1     32375
4      5418
2      1997
Name: count, dtype: int64

Columna: GrDiabetes
GrDiabetes
3    191862
1     32375
2      5418
Name: count, dtype: int64

Columna: SupGrPreDiabetes
SupGrPreDiabetes
2    191862
1     37793
Name: count, dtype: int64

Columna: SupGrNoPreDiabetes
SupGrNoPreDiabetes
2    197280
1     32375
Name: count, dtype: int64
```

Fuente: Elaboración propia

Frecuencias de cada categoría del dataset 2022

Figura 76 . *Frecuencias de cada categoría del dataset 2022(I).*

```
Columna: Year
Year
2022    273937
Name: count, dtype: int64

Columna: CatBMI
CatBMI
3    97854
4    95098
2    77112
1     3873
Name: count, dtype: int64

Columna: Stroke
Stroke
2    262886
1     11051
Name: count, dtype: int64

Columna: HeartDis
HeartDis
2    249738
1     24199
Name: count, dtype: int64

Columna: PhysExer
PhysExer
1    213822
2     60115
Name: count, dtype: int64

Columna: HealthIns
HealthIns
1    261479
2     12458
Name: count, dtype: int64

Columna: NoMedCost
NoMedCost
2    252649
1     21288
Name: count, dtype: int64

Columna: GenHealth
GenHealth
1    229501
2     44436
Name: count, dtype: int64

Columna: CogDiff
CogDiff
2    244180
1     29757
Name: count, dtype: int64

Columna: Depression
Depression
2    215445
1     58492
Name: count, dtype: int64

Columna: MentalHlth
MentalHlth
1    163444
2     73784
3     36709
Name: count, dtype: int64

Columna: MentalState
MentalState
1    200793
2     73144
Name: count, dtype: int64

Columna: PhysHlth
PhysHlth
1    168686
2     70910
3     34341
Name: count, dtype: int64

Columna: WalkDiff
WalkDiff
2    234191
1     39746
Name: count, dtype: int64

Columna: Gender
Gender
2    139258
1    134679
Name: count, dtype: int64

Columna: AgeRange
AgeRange
6    95324
5    53030
4    42395
3    39303
2    30517
1    13368
Name: count, dtype: int64

Columna: EdLevel
EdLevel
4    125374
3     75015
2     61764
1     11784
Name: count, dtype: int64

Columna: AnnIncome
AnnIncome
7     47343
9     41850
8     39623
6     35829
5     31182
11    19825
10    18887
4     14694
3     10140
2      7697
1      6867
Name: count, dtype: int64

Columna: SocClass
SocClass
2    128816
1    106409
3     38712
Name: count, dtype: int64

Columna: UrologyDz
UrologyDz
2    261665
1     12272
Name: count, dtype: int64

Columna: VisionDiff
VisionDiff
2    260567
1     13370
Name: count, dtype: int64

Columna: Asthma
Asthma
1    232770
2     41167
Name: count, dtype: int64

Columna: LungDiseases
LungDiseases
2    253054
1     20883
Name: count, dtype: int64

Columna: Arthritis
Arthritis
2    180654
1     93283
Name: count, dtype: int64

Columna: SmokerTrad
SmokerTrad
4    163348
3     77535
1     24002
2      9052
Name: count, dtype: int64

Columna: ECigSmok
ECigSmok
1    259260
2     14677
Name: count, dtype: int64

Columna: AlcDrinker
AlcDrinker
1    230228
2     43709
Name: count, dtype: int64
```

Fuente: Elaboración propia

Figura 77 . *Frecuencias de cada categoría del dataset 2022 (II).*

```
Columna: Race
Race
1    208652
5     24679
2     20685
3     13673
4      6248
Name: count, dtype: int64

Columna: MaritalSt
MaritalSt
1    147295
5     46316
2     36656
3     26548
6     12019
4      5103
Name: count, dtype: int64

Columna: LastMedChk
LastMedChk
1    218404
2     26441
3     15953
4     11904
5      1235
Name: count, dtype: int64

Columna: Awareness
Awareness
9    266618
1      3820
2      3499
Name: count, dtype: int64

Columna: FootIrrita
FootIrrita
999    273937
Name: count, dtype: int64

Columna: FecFootIrrita
FecFootIrrita
9    266332
2      6888
1       717
Name: count, dtype: int64

Columna: HighBP
HighBP
9    273937
Name: count, dtype: int64

Columna: HighChol
HighChol
9    273937
Name: count, dtype: int64

Columna: FruitCons
FruitCons
9    273937
Name: count, dtype: int64

Columna: VegCons
VegCons
9    273937
Name: count, dtype: int64

Columna: FruitOrVegCon
FruitOrVegCon
9    273937
Name: count, dtype: int64

Columna: FruitAndVegCon
FruitAndVegCon
9    273937
Name: count, dtype: int64

Columna: MarijuanaCon
MarijuanaCon
9    207511
2     57906
1      8520
Name: count, dtype: int64

Columna: SleepHours
SleepHours
7     86502
8     75789
6     60510
5     18229
9     12945
4      7095
10     5853
3      1743
12     1554
99     1268
2       765
1       589
11      358
15      164
14      160
16      159
13       81
18       75
20       59
24       11
17       10
23        7
22        6
19        5
Name: count, dtype: int64

Columna: BrDiabetes
BrDiabetes
3    228342
1     37201
4      6066
2      2328
Name: count, dtype: int64

Columna: GrDiabetes
GrDiabetes
3    230670
1     37201
2      6066
Name: count, dtype: int64

Columna: SupGrPreDiabetes
SupGrPreDiabetes
2    230670
1     43267
Name: count, dtype: int64

Columna: SupGrNoPreDiabetes
SupGrNoPreDiabetes
2    236736
1     37201
Name: count, dtype: int64
```

Fuente: Elaboración propia

Frecuencias de cada categoría del dataset 2021_22

Figura 78 . *Frecuencias de cada categoría del dataset 2021_22 (I).*

```
Columna: Year
Year
2022    273937
2021    229655
Name: count, dtype: int64

Columna: CatBMI
CatBMI
3    180563
4    177458
2    138989
1      6582
Name: count, dtype: int64

Columna: Stroke
Stroke
2    483747
1     19845
Name: count, dtype: int64

Columna: HeartDis
HeartDis
2    459831
1     43761
Name: count, dtype: int64

Columna: PhysExer
PhysExer
1    393216
2    110376
Name: count, dtype: int64

Columna: HealthIns
HealthIns
1    482662
2     20930
Name: count, dtype: int64

Columna: NoMedCost
NoMedCost
2    468049
1     35543
Name: count, dtype: int64

Columna: GenHealth
GenHealth
1    424369
2     79223
Name: count, dtype: int64
```

```
Columna: AgeRange
AgeRange
6    177621
5    102503
4     80936
3     71340
2     51260
1     19932
Name: count, dtype: int64

Columna: EdLevel
EdLevel
4    231209
3    138291
2    112859
1     21233
Name: count, dtype: int64

Columna: AnnIncome
AnnIncome
7     88059
9     77222
8     73624
6     66773
5     57484
11    34523
10    33807
4     26852
3     18598
2     14238
1     12412
Name: count, dtype: int64

Columna: SocClass
SocClass
2    238905
1    196357
3     68330
Name: count, dtype: int64

Columna: UrologyDz
UrologyDz
2    481914
1     21678
Name: count, dtype: int64
```

```
Columna: CogDiff
CogDiff
2    452080
1     51512
Name: count, dtype: int64

Columna: Depression
Depression
2    398588
1    105004
Name: count, dtype: int64

Columna: MentalHlth
MentalHlth
1    307082
2    132674
3     63836
Name: count, dtype: int64

Columna: MentalState
MentalState
1    373006
2    130586
Name: count, dtype: int64

Columna: PhysHlth
PhysHlth
1    321826
2    121296
3     60470
Name: count, dtype: int64

Columna: WalkDiff
WalkDiff
2    429204
1     74388
Name: count, dtype: int64

Columna: Gender
Gender
2    259480
1    244112
Name: count, dtype: int64
```

```
Columna: VisionDiff
VisionDiff
2    479874
1     23718
Name: count, dtype: int64

Columna: Asthma
Asthma
1    430402
2     73190
Name: count, dtype: int64

Columna: LungDiseases
LungDiseases
2    464909
1     38683
Name: count, dtype: int64

Columna: Arthritis
Arthritis
2    330653
1    172939
Name: count, dtype: int64

Columna: SmokerTrad
SmokerTrad
4    298970
3    143636
1     44279
2     16707
Name: count, dtype: int64

Columna: ECigSmok
ECigSmok
1    480162
2     23430
Name: count, dtype: int64

Columna: AlcDrinker
AlcDrinker
1    428256
2     75336
Name: count, dtype: int64
```

Fuente: Elaboración propia

Figura 79 . *Frecuencias de cada categoría del dataset 2021_22 (II).*

```
Columna: Race
Race
1    387119
5     43024
2     36990
3     25491
4     10968
Name: count, dtype: int64

Columna: MaritalSt
MaritalSt
1    278207
5     78402
2     67752
3     49421
6     20587
4      9223
Name: count, dtype: int64

Columna: LastMedChk
LastMedChk
1    403589
2     52755
3     26868
4     18690
5      1690
Name: count, dtype: int64

Columna: Awareness
Awareness
9    483725
1     10822
2      9045
Name: count, dtype: int64

Columna: FootIrrita
FootIrrita
999    491242
101      6222
888      1699
201      1103
102       545
        ...
191         1
106         1
424         1
299         1
213         1
Name: count, Length: 67, dtype: int64

Columna: FecFootIrrita
FecFootIrrita
9    483707
1     11298
2      8587
Name: count, dtype: int64

Columna: HighBP
HighBP
9    273937
2    131281
1     98374
Name: count, dtype: int64

Columna: HighChol
HighChol
9    273937
2    137531
1     92124
Name: count, dtype: int64

Columna: FruitCons
FruitCons
9    273937
1    142907
2     86748
Name: count, dtype: int64

Columna: VegCons
VegCons
9    273937
1    190337
2     39318
Name: count, dtype: int64

Columna: FruitOrVegCon
FruitOrVegCon
9    273937
1    205771
2     23884
Name: count, dtype: int64

Columna: FruitAndVegCon
FruitAndVegCon
9    273937
1    127473
2    102182
Name: count, dtype: int64

Columna: MarijuanaCon
MarijuanaCon
9    437166
2     57906
1      8520
Name: count, dtype: int64

Columna: SleepHours
SleepHours
99    230923
7      86502
8      75789
6      60510
5      18229
9      12945
4       7095
10      5853
3       1743
12      1554
2        765
1        589
11       358
15       164
14       160
16       159
13        81
18        75
20        59
24        11
17        10
23         7
22         6
19         5
Name: count, dtype: int64

Columna: BrDiabetes
BrDiabetes
3    418207
1     69576
4     11484
2      4325
Name: count, dtype: int64
```

```
Columna: BrDiabetes
BrDiabetes
3    418207
1     69576
4     11484
2      4325
Name: count, dtype: int64
```

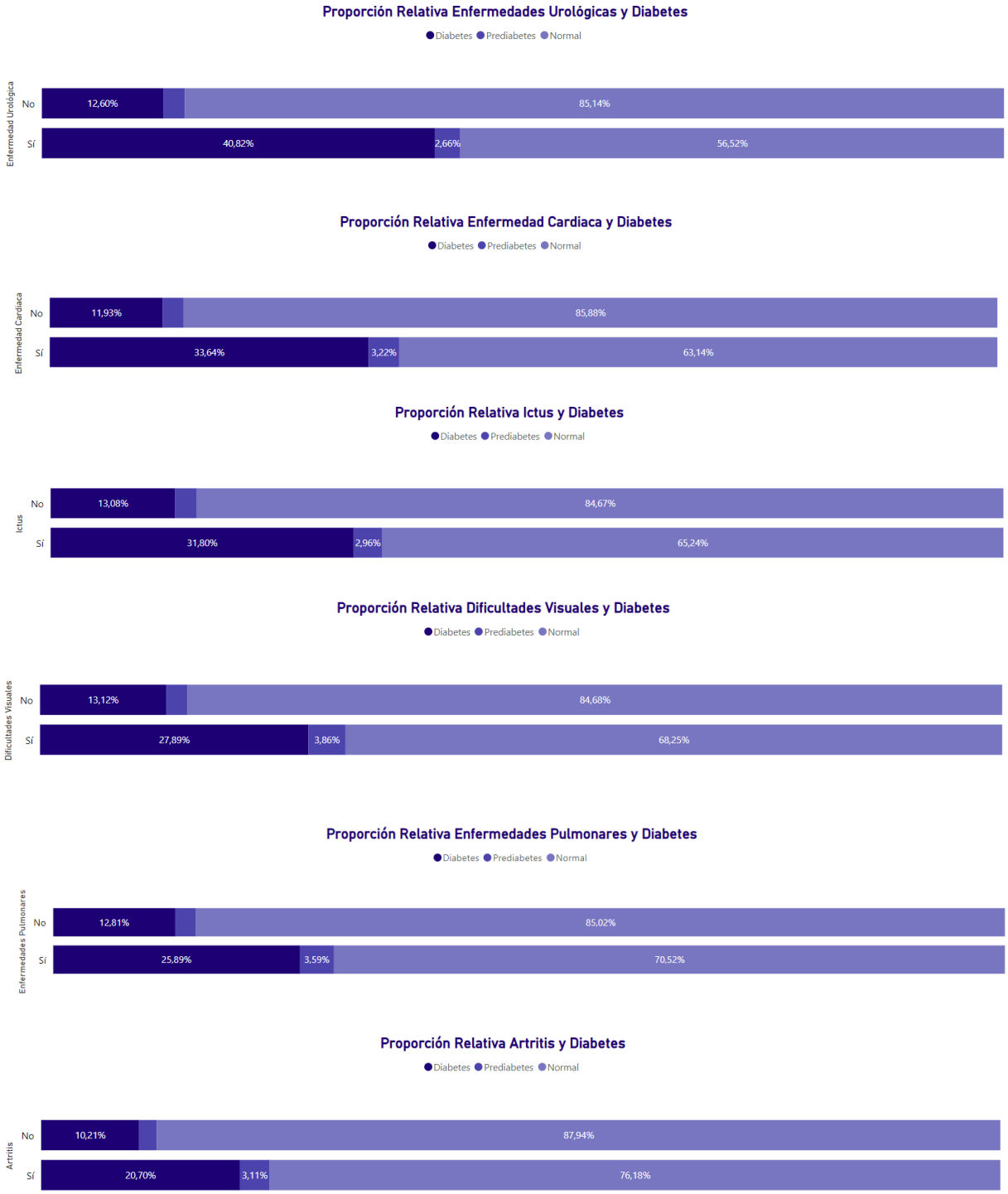
```
Columna: GrDiabetes
GrDiabetes
3    422532
1     69576
2     11484
Name: count, dtype: int64

Columna: SupGrPreDiabetes
SupGrPreDiabetes
2    422532
1     81060
Name: count, dtype: int64

Columna: SupGrNoPreDiabetes
SupGrNoPreDiabetes
2    434016
1     69576
Name: count, dtype: int64
```

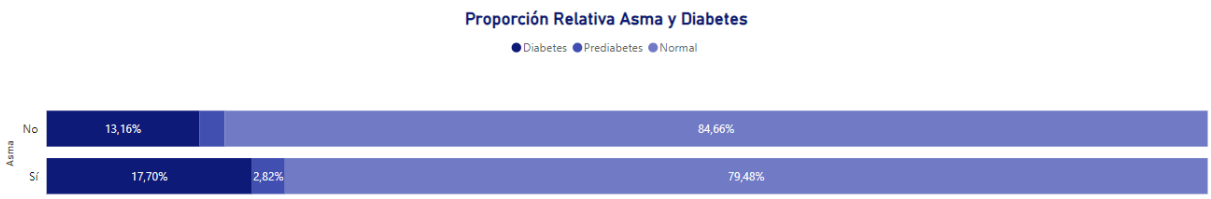
Fuente: Elaboración propia

Anexo D. Gráficas del estudio y análisis de cada atributo

Gráficas acordes a la bibliografía consultada.

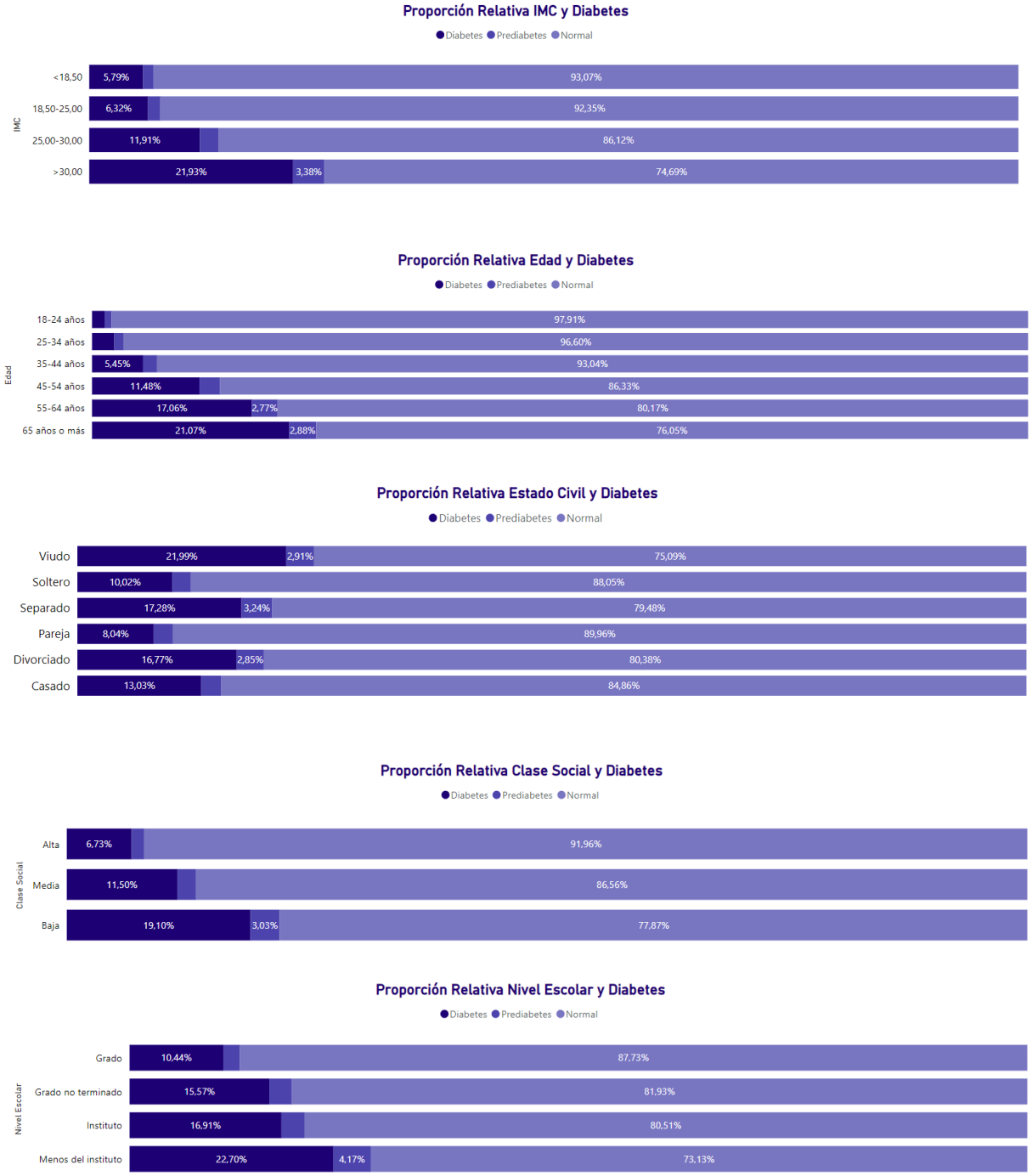
Figura 80 . *Atributos según patologías.*





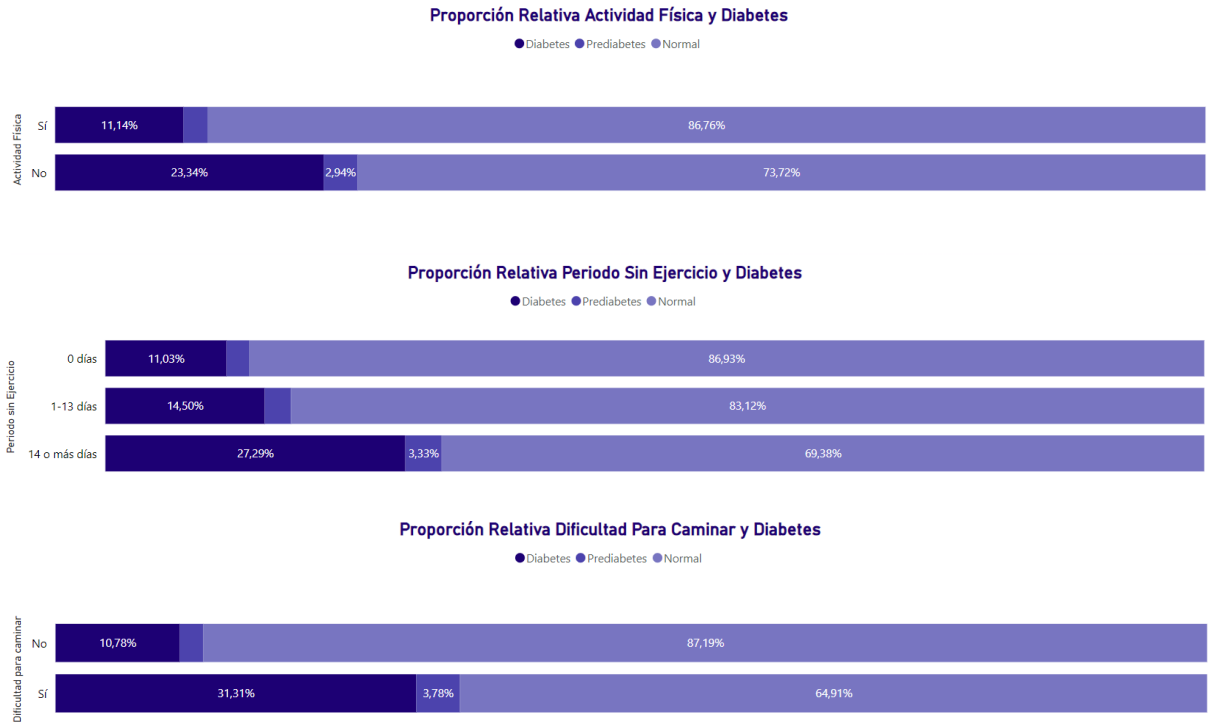
Fuente: Elaboración propia

Figura 81 . *Atributos demográficos.*



Fuente: Elaboración propia

Figura 82 . *Actividad física.*



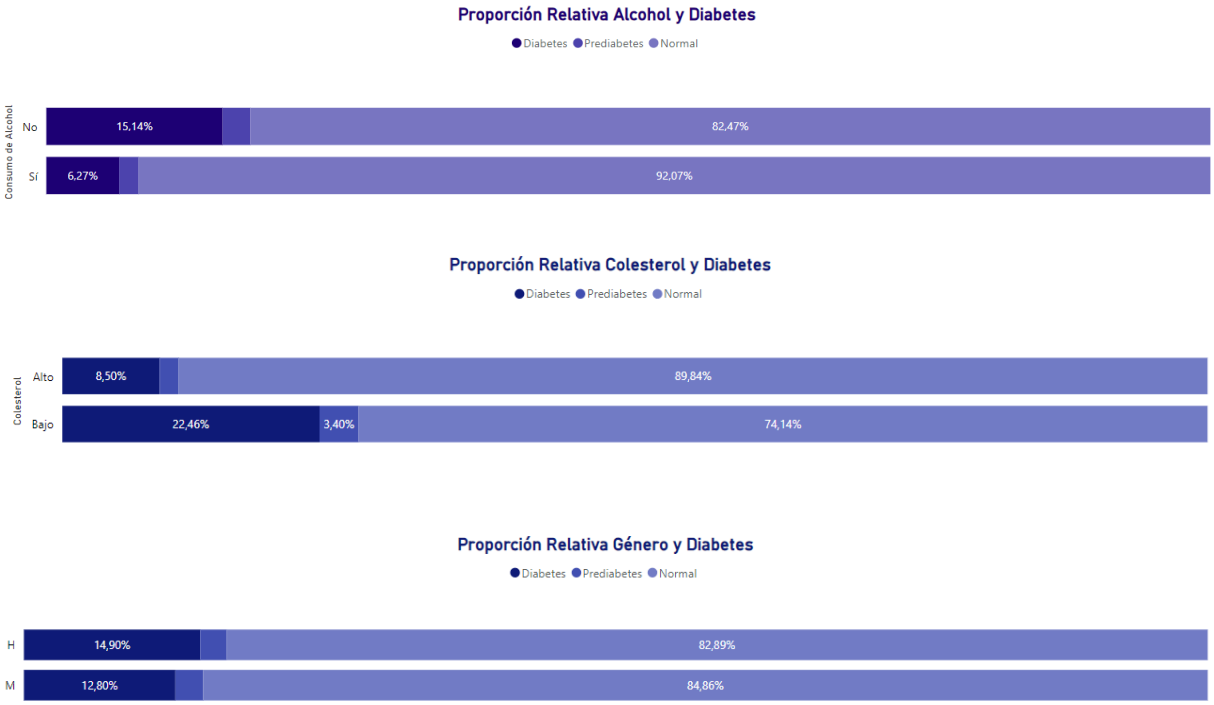
Fuente: Elaboración propia

Figura 83 . *Salud Mental.*

Fuente: Elaboración propia

Gráficas no acordes a la bibliografía consultada

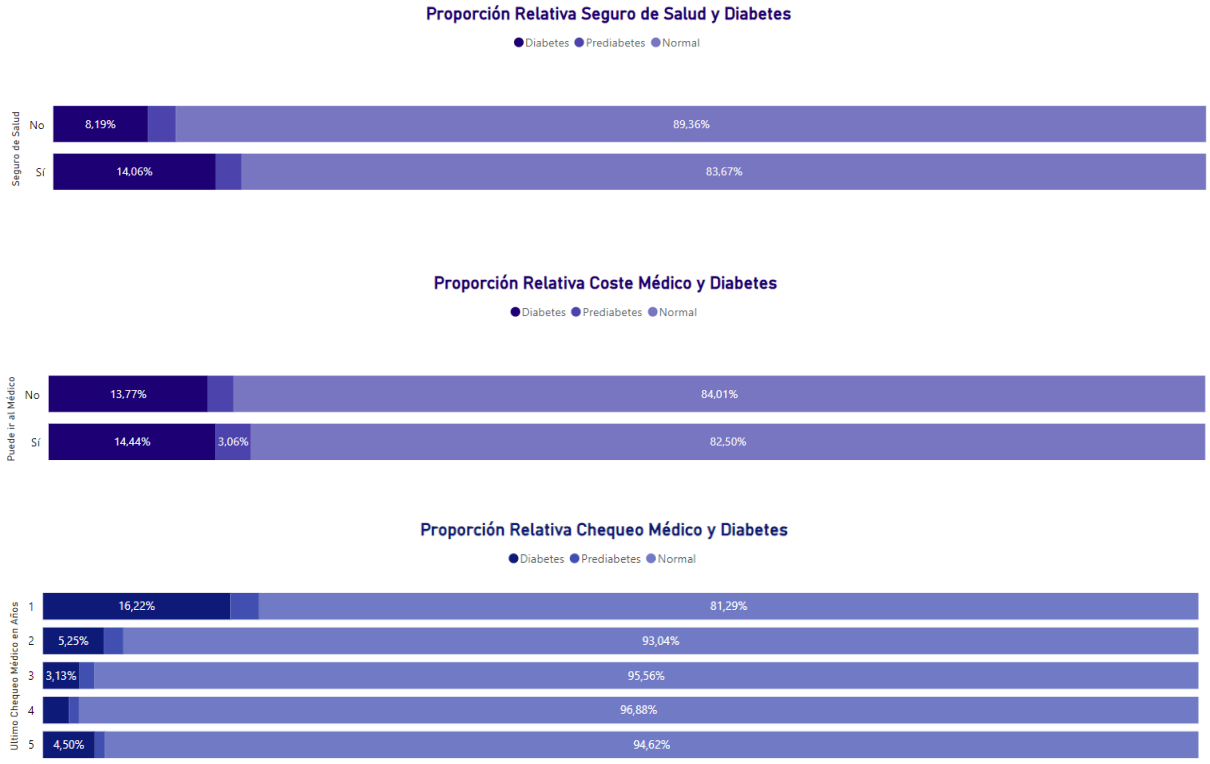
Figura 84 . *Resultados no acordes a la bibliografía consultada.*



Fuente: Elaboración propia

Información general

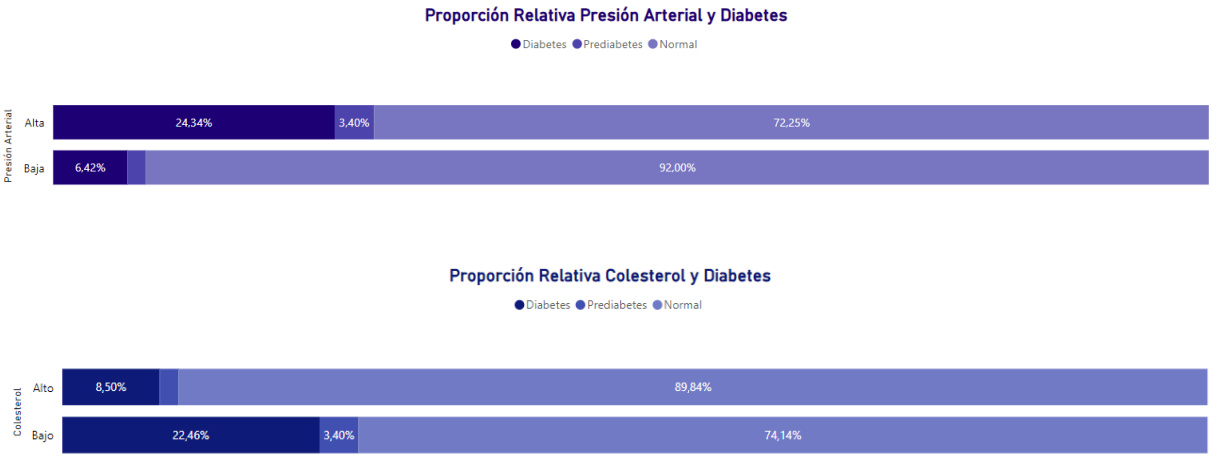
Figura 85 . *Información general.*



Fuente: Elaboración propia

Gráficas realizadas mediante el dataset de 2021

Figura 86 . *Atributos exclusivos 2021.*

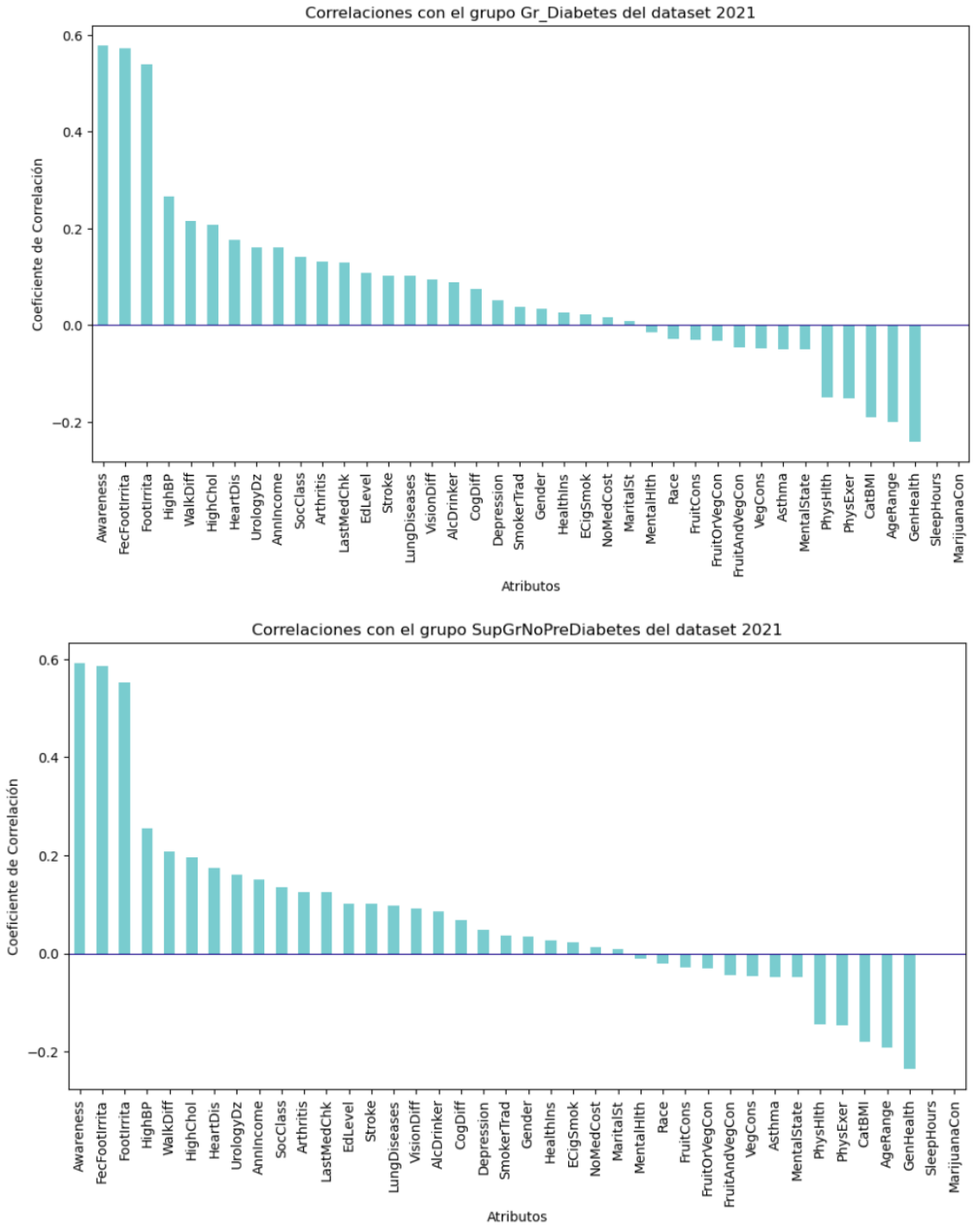


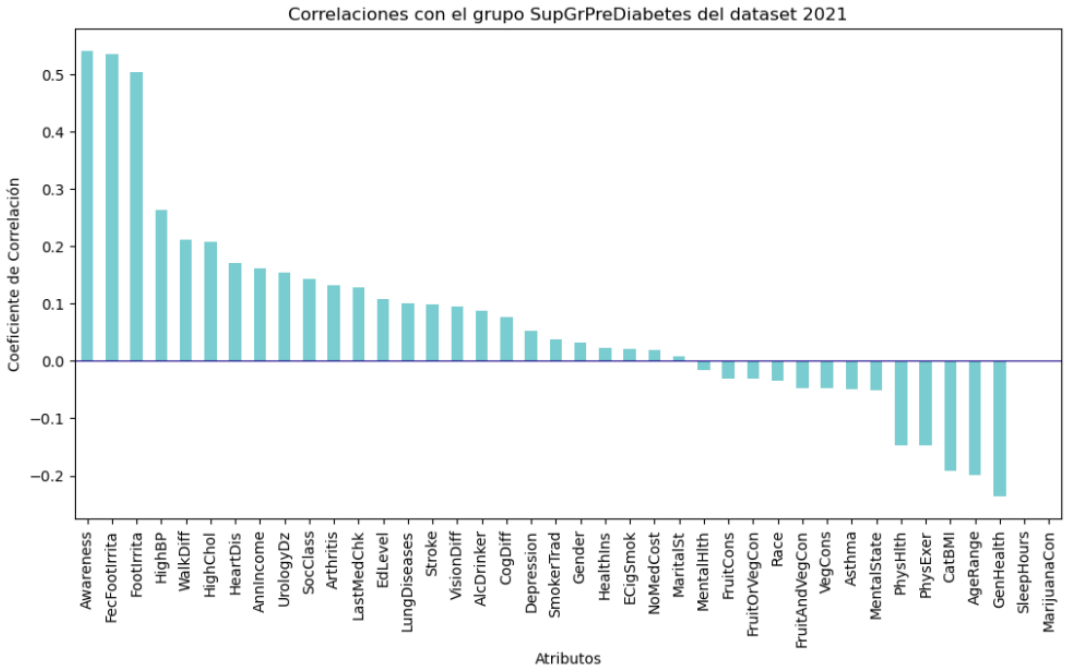
Fuente: Elaboración propia

Anexo E. Gráficas de las correlaciones

Correlaciones del dataset 2021

Figura 87 . *Gráfica correlaciones del dataset 2021.*

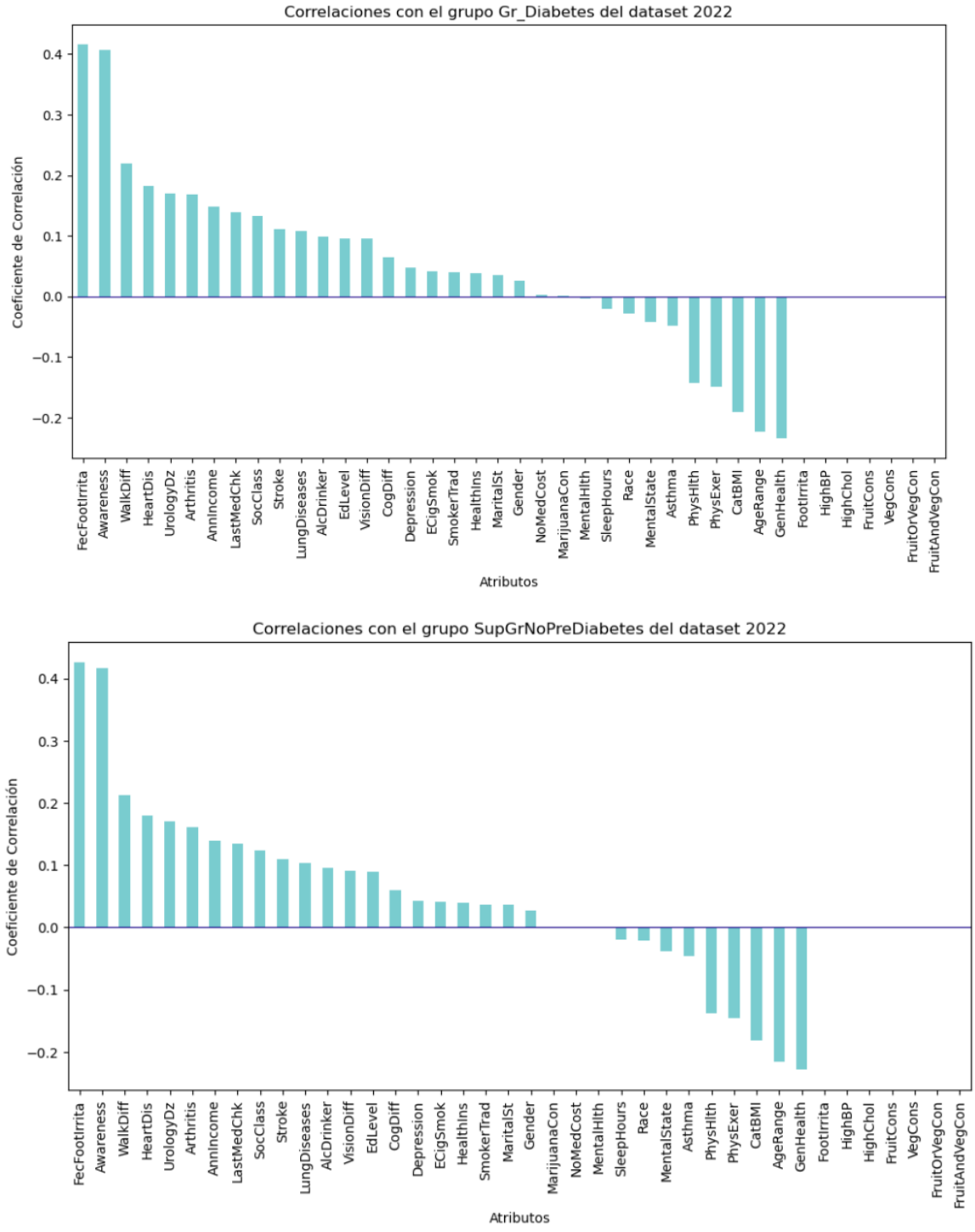


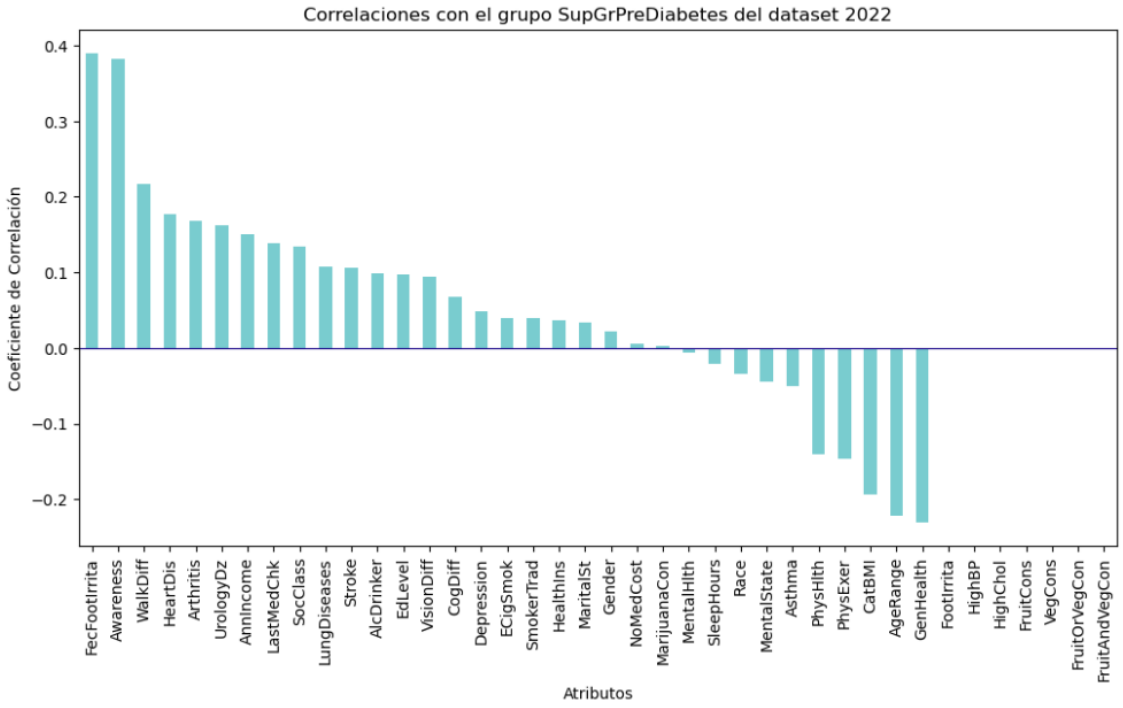


Fuente: Elaboración propia

Correlaciones del dataset 2022

Figura 88 . *Gráfica Correlaciones del dataset 2022.*

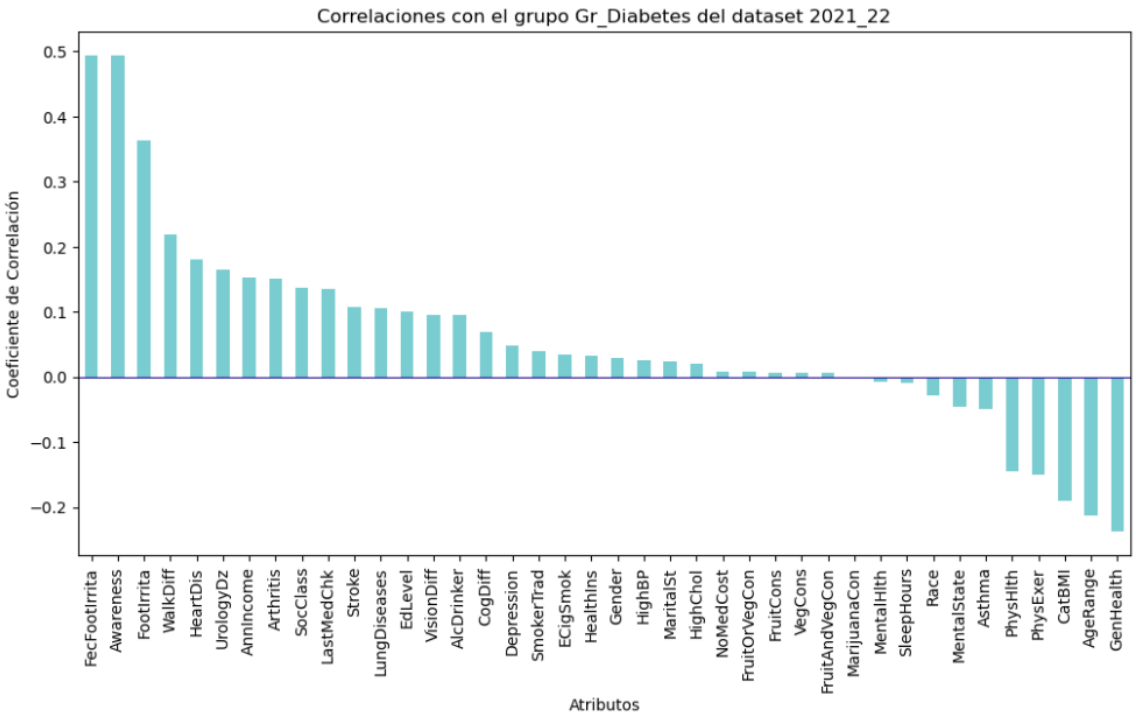


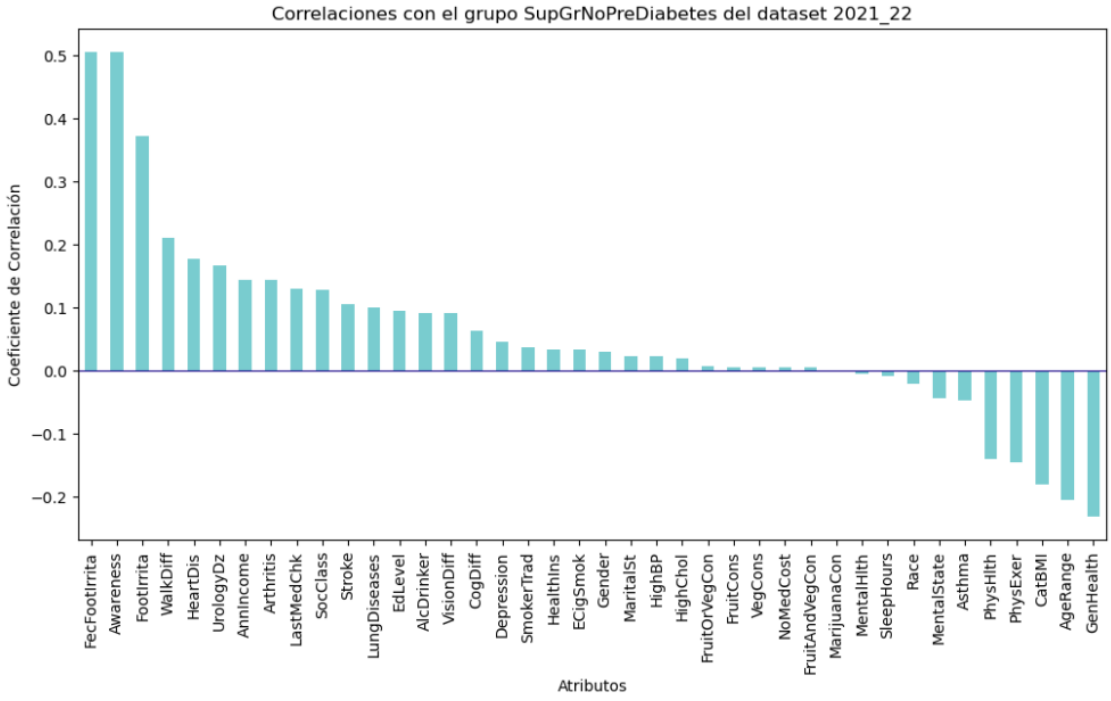


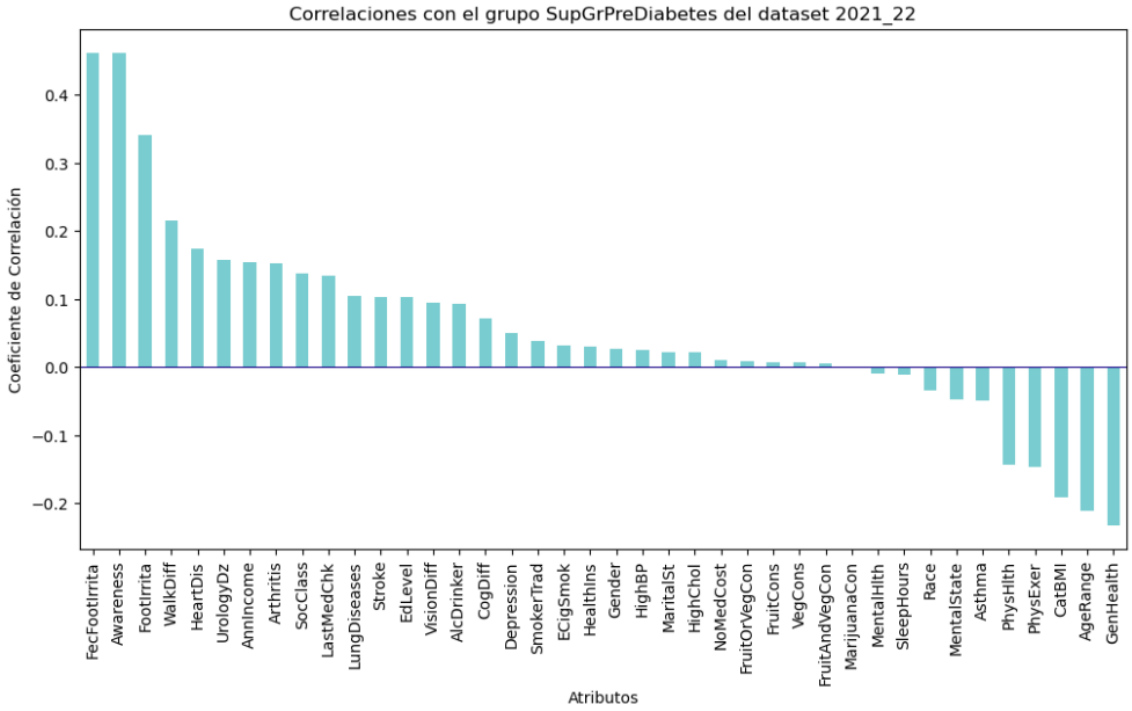
Fuente: Elaboración propia

Correlaciones del dataset 2021_22

Figura 89 . *Gráfica Correlaciones del dataset 2021_22.*







Fuente: Elaboración propia

Anexo F. Valor numérico de las correlaciones

Valor numérico de las correlaciones del dataset 2021

Figura 90 . *Correlaciones con GrDiabetes 2021.*

```
GrDiabetes        1.000000
Awareness         0.578193
FecFootIrrita     0.572354
FootIrrita        0.539312
HighBP            0.265223
WalkDiff          0.215201
HighChol          0.206800
HeartDis          0.176794
UrologyDz         0.160624
AnnIncome         0.159948
SocClass          0.141987
Arthritis         0.132021
LastMedChk        0.129103
EdLevel           0.107296
Stroke            0.101910
LungDiseases      0.101755
VisionDiff        0.095121
AlcDrinker        0.088427
CogDiff           0.073995
Depression        0.051618
SmokerTrad        0.038399
Gender            0.034259
HealthIns         0.025354
ECigSmok          0.022788
NoMedCost         0.016112
MaritalSt         0.008235
MentalHlth       -0.013827
Race             -0.028488
FruitCons        -0.030617
FruitOrVegCon    -0.031363
FruitAndVegCon   -0.046427
VegCons          -0.047260
Asthma           -0.049637
MentalState      -0.050667
PhysHlth         -0.149218
PhysExer         -0.150587
CatBMI           -0.189637
AgeRange         -0.199872
GenHealth        -0.240951
SleepHours             NaN
MarijuanaCon           NaN
Name: GrDiabetes, dtype: float64
```

Fuente: Elaboración propia

Figura 91 . *Correlaciones con SupGrNoPreDiabetes 2021.*

```
SupGrNoPreDiabetes    1.000000
Awareness             0.592103
FecFootIrrita         0.586123
FootIrrita            0.552286
HighBP                0.254913
WalkDiff              0.208599
HighChol              0.196605
HeartDis              0.174695
UrologyDz             0.161184
AnnIncome             0.151529
SocClass              0.134807
Arthritis             0.125575
LastMedChk            0.124743
EdLevel               0.101409
Stroke                0.100694
LungDiseases          0.097741
VisionDiff            0.091401
AlcDrinker            0.086112
CogDiff               0.068800
Depression            0.048854
SmokerTrad            0.036799
Gender                0.034823
HealthIns             0.026707
ECigSmok              0.023456
NoMedCost             0.013143
MaritalSt             0.008364
MentalHlth           -0.011500
Race                 -0.021183
FruitCons            -0.028775
FruitOrVegCon        -0.029883
FruitAndVegCon       -0.044168
VegCons              -0.045450
Asthma               -0.047451
MentalState          -0.047998
PhysHlth             -0.145248
PhysExer             -0.146914
CatBMI               -0.179605
AgeRange             -0.192157
GenHealth            -0.235274
SleepHours                 NaN
MarijuanaCon               NaN
Name: SupGrNoPreDiabetes, dtype: float64
```

Fuente: Elaboración propia

Figura 92 . *Correlaciones con SupGrPreDiabetes 2021.*

```
SupGrPreDiabetes    1.000000
Awareness           0.540443
FecFootIrrita       0.534984
FootIrrita          0.504100
HighBP              0.263572
WalkDiff            0.212205
HighChol            0.207535
HeartDis            0.171213
AnnIncome           0.161019
UrologyDz           0.153237
SocClass            0.142661
Arthritis           0.132432
LastMedChk          0.127680
EdLevel             0.108239
LungDiseases        0.101177
Stroke              0.098698
VisionDiff          0.094550
AlcDrinker          0.086821
CogDiff             0.075709
Depression          0.052008
SmokerTrad          0.038260
Gender              0.032266
HealthIns           0.023001
ECigSmok            0.021186
NoMedCost           0.018210
MaritalSt           0.007762
MentalHlth         -0.015420
FruitCons          -0.031038
FruitOrVegCon      -0.031413
Race               -0.034126
FruitAndVegCon     -0.046565
VegCons            -0.046940
Asthma             -0.049569
MentalState        -0.051008
PhysHlth           -0.146570
PhysExer           -0.147603
CatBMI             -0.190952
AgeRange           -0.198577
GenHealth          -0.235988
SleepHours               NaN
MarijuanaCon             NaN
Name: SupGrPreDiabetes, dtype: float64
```

Fuente: Elaboración propia

Valor numérico de las correlaciones del dataset 2022

Figura 93 . *Correlaciones con GrDiabetes 2022.*

```
GrDiabetes          1.000000
FecFootIrrita       0.416205
Awareness           0.407518
WalkDiff            0.220072
HeartDis            0.182145
UrologyDz           0.169757
Arthritis           0.168207
AnnIncome           0.147516
LastMedChk          0.139459
SocClass            0.132085
Stroke              0.110406
LungDiseases        0.108391
AlcDrinker          0.099224
EdLevel             0.096006
VisionDiff          0.094721
CogDiff             0.064696
Depression          0.046863
ECigSmok            0.041206
SmokerTrad          0.039227
HealthIns           0.038870
MaritalSt           0.035571
Gender              0.025104
NoMedCost           0.002339
MarijuanaCon        0.001521
MentalHlth         -0.003485
SleepHours         -0.021005
Race               -0.028369
MentalState        -0.043124
Asthma             -0.048991
PhysHlth           -0.142674
PhysExer           -0.149091
CatBMI             -0.191584
AgeRange           -0.223400
GenHealth          -0.234562
FootIrrita               NaN
HighBP                   NaN
HighChol                 NaN
FruitCons                NaN
VegCons                  NaN
FruitOrVegCon            NaN
FruitAndVegCon           NaN
Name: GrDiabetes, dtype: float64
```

Fuente: Elaboración propia

Figura 94 . *Correlaciones con SupGrNoPreDiabetes 2022.*

```
SupGrNoPreDiabetes    1.000000
FecFootIrrita         0.425906
Awareness             0.417018
WalkDiff              0.213045
HeartDis              0.179513
UrologyDz             0.170220
Arthritis             0.160395
AnnIncome             0.138593
LastMedChk            0.133985
SocClass              0.124063
Stroke                0.109737
LungDiseases          0.103642
AlcDrinker            0.096019
VisionDiff            0.090224
EdLevel               0.090143
CogDiff               0.059139
Depression            0.043054
ECigSmok              0.040656
HealthIns             0.039372
SmokerTrad            0.036961
MaritalSt             0.035648
Gender                0.026866
MarijuanaCon          0.000842
NoMedCost            -0.000794
MentalHlth           -0.001198
SleepHours           -0.019769
Race                 -0.020599
MentalState          -0.039092
Asthma               -0.045787
PhysHlth             -0.138076
PhysExer             -0.145580
CatBMI               -0.181584
AgeRange             -0.215312
GenHealth            -0.228369
FootIrrita                 NaN
HighBP                     NaN
HighChol                   NaN
FruitCons                  NaN
VegCons                    NaN
FruitOrVegCon              NaN
FruitAndVegCon             NaN
Name: SupGrNoPreDiabetes, dtype: float64
```

Fuente: Elaboración propia

Figura 95 . *Correlaciones con SupGrPreDiabetes 2022.*

```
SupGrPreDiabetes    1.000000
FecFootIrrita       0.389831
Awareness           0.381695
WalkDiff            0.217546
HeartDis            0.177064
Arthritis           0.168569
UrologyDz           0.162282
AnnIncome           0.149780
LastMedChk          0.138818
SocClass            0.134144
LungDiseases        0.108356
Stroke              0.106455
AlcDrinker          0.098121
EdLevel             0.097532
VisionDiff          0.095017
CogDiff             0.067233
Depression          0.048498
ECigSmok            0.040014
SmokerTrad          0.039729
HealthIns           0.036786
MaritalSt           0.034024
Gender              0.022407
NoMedCost           0.005185
MarijuanaCon        0.002096
MentalHlth         -0.005489
SleepHours         -0.021294
Race               -0.034492
MentalState        -0.045123
Asthma             -0.049969
PhysHlth           -0.141077
PhysExer           -0.146206
CatBMI             -0.193032
AgeRange           -0.221733
GenHealth          -0.230652
FootIrrita               NaN
HighBP                   NaN
HighChol                 NaN
FruitCons                NaN
VegCons                  NaN
FruitOrVegCon            NaN
FruitAndVegCon           NaN
Name: SupGrPreDiabetes, dtype: float64
```

Fuente: Elaboración propia

Valor numérico de las correlaciones del dataset 2021_22

Figura 96 . *Correlaciones con GrDiabetes 2021_22.*

```
GrDiabetes        1.000000
FecFootIrrita     0.493599
Awareness         0.493348
FootIrrita        0.363716
WalkDiff          0.217865
HeartDis          0.179627
UrologyDz         0.165491
AnnIncome         0.153219
Arthritis         0.151571
SocClass          0.136656
LastMedChk        0.134886
Stroke            0.106480
LungDiseases      0.105342
EdLevel           0.101124
VisionDiff        0.094786
AlcDrinker        0.094563
CogDiff           0.068570
Depression        0.048900
SmokerTrad        0.038877
ECigSmok          0.033628
HealthIns         0.033106
Gender            0.029179
HighBP            0.024636
MaritalSt         0.023679
HighChol          0.020983
NoMedCost         0.008044
FruitOrVegCon     0.007294
FruitCons         0.006601
VegCons           0.006311
FruitAndVegCon    0.005582
MarijuanaCon     -0.001962
MentalHlth       -0.007892
SleepHours       -0.009963
Race             -0.028236
MentalState      -0.046373
Asthma           -0.049134
PhysHlth         -0.145083
PhysExer         -0.149762
CatBMI           -0.190787
AgeRange         -0.212951
GenHealth        -0.237243
Name: GrDiabetes, dtype: float64
```

Fuente: Elaboración propia

Figura 97 . *Correlaciones con SupGrNoPreDiabetes 2021_22.*

```
SupGrNoPreDiabetes    1.000000
FecFootIrrita         0.505276
Awareness             0.505019
FootIrrita            0.372320
WalkDiff              0.211030
HeartDis              0.177245
UrologyDz             0.166007
AnnIncome             0.144521
Arthritis             0.144383
LastMedChk            0.129870
SocClass              0.129010
Stroke                0.105569
LungDiseases          0.100931
EdLevel               0.095257
AlcDrinker            0.091728
VisionDiff            0.090650
CogDiff               0.063203
Depression            0.045580
SmokerTrad            0.036914
HealthIns             0.033956
ECigSmok              0.033554
Gender                0.030406
MaritalSt             0.023735
HighBP                0.023050
HighChol              0.019408
FruitOrVegCon         0.006388
FruitCons             0.005752
VegCons               0.005433
NoMedCost             0.005019
FruitAndVegCon        0.004762
MarijuanaCon         -0.002084
MentalHlth           -0.005620
SleepHours           -0.008915
Race                 -0.020700
MentalState          -0.042981
Asthma               -0.046406
PhysHlth             -0.140819
PhysExer             -0.146179
CatBMI               -0.180760
AgeRange             -0.204998
GenHealth            -0.231302
Name: SupGrNoPreDiabetes, dtype: float64
```

Fuente: Elaboración propia

Figura 98 . *Correlaciones con SupGrPreDiabetes 2021_22.*

```
SupGrPreDiabetes    1.000000
FecFootIrrita       0.461883
Awareness           0.461648
FootIrrita          0.340345
WalkDiff            0.215124
HeartDis            0.174312
UrologyDz           0.158049
AnnIncome           0.154945
Arthritis           0.151948
SocClass            0.138091
LastMedChk          0.133924
LungDiseases        0.105055
Stroke              0.102859
EdLevel             0.102382
VisionDiff          0.094685
AlcDrinker          0.093249
CogDiff             0.070727
Depression          0.049960
SmokerTrad          0.039086
ECigSmok            0.032283
HealthIns           0.030917
Gender              0.026801
HighBP              0.025089
MaritalSt           0.022630
HighChol            0.021581
NoMedCost           0.010546
FruitOrVegCon       0.007838
FruitCons           0.007121
VegCons             0.006870
FruitAndVegCon      0.006117
MarijuanaCon       -0.001764
MentalHlth         -0.009695
SleepHours         -0.010529
Race               -0.034126
MentalState        -0.047609
Asthma             -0.049629
PhysHlth           -0.142987
PhysExer           -0.146830
CatBMI             -0.192181
AgeRange           -0.211466
GenHealth          -0.232847
Name: SupGrPreDiabetes, dtype: float64
```

Fuente: Elaboración propia

Anexo G. Modelos obtenidos

Tabla 5. *Modelos, estimadores y métricas.*

	Algoritmo	Mejores Hiperparámetros	Mejor Estimador	DataSet	Atributos	Técnica Balanceo	Registros	Diabeticos	No Diabeticos	F1 Score	Precisión (Precision)	Exactitud (Accuracy)	Especificidad (Specificity)	AUC-ROC	Tiempo Generación Modelo	Tiempo Normalizado	Índice de Calidad
2	Conjunto de árboles (Random Forest)	'bootstrap': True 'criterion': 'gini' 'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 5 'n_estimators': 100	class_weight='balanced' min_samples_split=5 n_jobs=-1 random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	class_weight='balanced' Mémicas	229.655	37.793	191.862	0,80	0,80	0,79	0,67	0,85	8,25	0,05	0,79
3	Conjunto de árboles (Random Forest)	'bootstrap': True 'criterion': 'gini' 'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 5 'n_estimators': 100	min_samples_split=5 n_jobs=-1 random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	Mémicas	229.655	37.793	191.862	0,81	0,80	0,84	0,96	0,58	8,05	0,04	0,81
4	Conjunto de árboles (Random Forest)	'bootstrap': True 'criterion': 'gini' 'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 5 'n_estimators': 100	min_samples_split=5 n_jobs=-1 random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Mémicas	229.655	37.793	191.862	0,78	0,80	0,76	0,81	0,66	24,73	0,14	0,76
5	Conjunto de árboles (Random Forest)	'bootstrap': True 'criterion': 'gini' 'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 5 'n_estimators': 100	class_weight='balanced' min_samples_split=5 n_jobs=-1 random_state=14	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	class_weight='balanced' Mémicas	503.532	81.060	422.532	0,78	0,81	0,76	0,81	0,66	19,31	0,11	0,76
6	Conjunto de árboles (Random Forest)	'bootstrap': True 'criterion': 'gini' 'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 5 'n_estimators': 100	min_samples_split=5 n_jobs=-1 random_state=14	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	Mémicas	503.532	81.060	422.532	0,80	0,80	0,84	0,96	0,57	19,66	0,11	0,81
7	Regresión Logística (Logistic Regression)	'C': 10 'max_iter': 100 'solver': 'sag' 'tol': 0.01	C=10, class_weight='balanced' random_state=14 solver='sag' tol=0.01	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	class_weight='balanced' Mémicas	229.655	37.793	191.862	0,75	0,84	0,72	0,71	0,73	2,16	0,01	0,75
8	Regresión Logística (Logistic Regression)	'C': 10 'max_iter': 100 'solver': 'sag' 'tol': 0.01	C=10, random_state=14 solver='sag' tol=0.01	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	Mémicas	229.655	37.793	191.862	0,81	0,81	0,84	0,97	0,58	0,87	0,00	0,81

	Algoritmo	Mejores Hiperparámetros	Mejor Estimador	DataSet	Atributos	Técnica Balanceo	Registros	Diabeticos	No Diabeticos	F1 Score	Precisión (Precision)	Exactitud (Accuracy)	Especificidad (Specificity)	AUC-ROC	Tiempo Generación Modelo	Tiempo Normalizado	Índice de Calidad
9	Regresión Logística (Logistic Regression)	'C': 10 'max_iter': 100 'solver': 'sag' 'tol': 0.01	C=10, random_state=14 solver='sag' tol=0.01	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Mémicas	229.655	37.793	191.862	0,75	0,83	0,72	0,73	0,71	11,17	0,06	0,74
10	Regresión Logística (Logistic Regression)	'C': 10 'max_iter': 100 'solver': 'sag' 'tol': 0.01	C=10, class_weight='balanced' random_state=14 solver='sag' tol=0.01	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	class_weight='balanced' Mémicas	503.532	81.060	422.532	0,74	0,84	0,71	0,70	0,72	3,13	0,02	0,74
11	Regresión Logística (Logistic Regression)	'C': 10 'max_iter': 100 'solver': 'sag' 'tol': 0.01	C=10, random_state=14 solver='sag' tol=0.01	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	Mémicas	503.532	81.060	422.532	0,81	0,81	0,84	0,96	0,57	2,79	0,02	0,81
12	Árboles de Decisión (Decision Tree Classifier)	'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 2	class_weight='balanced' max_features='sqrt' random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	class_weight='balanced' Mémicas	229.655	37.793	191.862	0,76	0,77	0,75	0,82	0,60	0,25	0,00	0,75
13	Árboles de Decisión (Decision Tree Classifier)	'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 2	max_features='sqrt' random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	Mémicas	229.655	37.793	191.862	0,79	0,78	0,79	0,89	0,59	0,20	0,00	0,78
14	Árboles de Decisión (Decision Tree Classifier)	'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 2	max_features='sqrt' random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Mémicas	229.655	37.793	191.862	0,75	0,78	0,73	0,78	0,62	10,34	0,06	0,74
15	Árboles de Decisión (Decision Tree Classifier)	'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 2	class_weight='balanced' max_features='sqrt' random_state=14	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	class_weight='balanced' Mémicas	503.532	81.060	422.532	0,75	0,79	0,72	0,77	0,62	0,57	0,00	0,73
16	Árboles de Decisión (Decision Tree Classifier)	'max_depth': None 'max_features': 'sqrt' 'min_samples_leaf': 1 'min_samples_split': 2	max_features='sqrt' random_state=14	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	Mémicas	503.532	81.060	422.532	0,79	0,78	0,81	0,92	0,57	0,52	0,00	0,79

	Algoritmo	Mejores Hiperparámetros	Mejor Estimador	DataSet	Atributos	Técnica Balanceo	Registros	Diabeticos	No Diabeticos	F1 Score	Precisión (Precision)	Exactitud (Accuracy)	Especificidad (Specificity)	AUC-ROC	Tiempo Generación Modelo	Tiempo Normalizado	Índice de Calidad
17	Gradient Boosting Machines (GBM)	'learning_rate': 0.1 'max_depth': 5 'min_samples_leaf': 4 'min_samples_split': 10 'n_estimators': 200	max_depth=5, min_samples_leaf=4, min_samples_split=10, n_estimators=200, random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	Mémicas	229.655	37.793	191.862	0,81	0,81	0,84	0,97	0,59	82,07	0,46	0,82
18	Gradient Boosting Machines (GBM)	'learning_rate': 0.1 'max_depth': 5 'min_samples_leaf': 4 'min_samples_split': 10 'n_estimators': 200	max_depth=5, min_samples_leaf=4, min_samples_split=10, n_estimators=200, random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Mémicas	229.655	37.793	191.862	0,75	0,83	0,72	0,72	0,71	163,46	0,91	0,74
19	Gradient Boosting Machines (GBM)	'learning_rate': 0.1 'max_depth': 5 'min_samples_leaf': 4 'min_samples_split': 10 'n_estimators': 200	max_depth=5, min_samples_leaf=4, min_samples_split=10, n_estimators=200, random_state=14	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	Mémicas	503.532	81.060	422.532	0,81	0,81	0,85	0,96	0,57	178,96	1,00	0,81
20	Navie Bayes Gaussiano	'priors': [0.999, 0.001] 'var_smoothing': 1e-09	priors=[0.001, 0.999]	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	Mémicas	229.655	37.793	191.862	0,81	0,80	0,83	0,94	0,60	0,07	0,00	0,81
21	Navie Bayes Gaussiano	'priors': [0.999, 0.001] 'var_smoothing': 1e-09	priors=[0.001, 0.999]	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Mémicas	229.655	37.793	191.862	0,81	0,80	0,83	0,94	0,60	10,09	0,06	0,80
22	Navie Bayes Gaussiano	'priors': [0.999, 0.001] 'var_smoothing': 1e-09	priors=[0.001, 0.999]	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	Mémicas	503.532	81.060	422.532	0,81	0,80	0,83	0,95	0,59	0,11	0,00	0,81
23	Support Vector Machine (SVM)	'C': 0.01, 'kernel': 'sigmoid'	class_weight='balanced' random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	class_weight='balanced' Mémicas	229.655	37.793	191.862	0,74	0,84	0,70	0,68	0,73	1.601,00	8,94	0,69
24	Support Vector Machine (SVM)	'C': 0.01, 'kernel': 'sigmoid'	random_state=14	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeanDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','ValidDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Mémicas	229.655	37.793	191.862	0,75	0,83	0,71	0,72	0,71	2.749,23	15,36	0,66

	Algoritmo	Mejores Hiperparámetros	Mejor Estimador	DataSet	Atributos	Técnica Balanceo	Registros	Diabeticos	No Diabeticos	F1 Score	Precisión (Precision)	Exactitud (Accuracy)	Especificidad (Specificity)	AUC-ROC	Tiempo Generación Modelo	Tiempo Normalizado	Indice de Calidad
1																	
25	Support Vector Machine (SVM)	'C': 0.01, 'kernel': 'sigmoid'	class_weight='balanced' random_state=14	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	class_weight='balanced' Métricas	503.592	81.060	422.532	0,73	0,84	0,69	0,67	0,72	7.736,92	43,23	0,51
26	Redes Neuronales Artificiales Densas	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	25 'CatBMI','Stroke','HeartDis','PhysExer','GenHealth', 'CogDif','Depression','PhysHlth','WalkDif','Gender', 'AgeRange','EdLevel','SocClass','UrologyDis', 'VisionDif','Asthma','LungDiseases','Arthritis', 'SmokerTrad','AlcDrinker','Race','LastMedChk', 'HighBP','HighChol','FruitAndVegCon',	Métricas	229.655	37.793	191.862	0,81	0,82	0,85	0,98	0,58	13,15	0,07	0,82
27	Redes Neuronales Artificiales Densas	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	Métricas	229.655	37.793	191.862	0,81	0,82	0,85	0,97	0,59	12,67	0,07	0,82
28	Redes Neuronales Artificiales Densas	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	25 'CatBMI','Stroke','HeartDis','PhysExer','GenHealth', 'CogDif','Depression','PhysHlth','WalkDif','Gender', 'AgeRange','EdLevel','SocClass','UrologyDis', 'VisionDif','Asthma','LungDiseases','Arthritis', 'SmokerTrad','AlcDrinker','Race','LastMedChk', 'HighBP','HighChol','FruitAndVegCon',	SMOTE Métricas	229.655	37.793	191.862	0,75	0,83	0,72	0,73	0,71	31,96	0,18	0,74
29	Redes Neuronales Artificiales Densas	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Métricas	229.655	37.793	191.862	0,75	0,83	0,71	0,72	0,71	30,96	0,17	0,74

	Algoritmo	Mejores Hiperparámetros	Mejor Estimador	DataSet	Atributos	Técnica Balanceo	Registros	Diabeticos	No Diabeticos	F1 Score	Precisión (Precision)	Exactitud (Accuracy)	Especificidad (Specificity)	AUC-ROC	Tiempo Generación Modelo	Tiempo Normalizado	Indice de Calidad
1																	
30	Redes Neuronales Artificiales Densas	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	layers.Dense(128, activation='relu') layers.Dense(64, activation='relu') layers.Dense(32,activation='relu') layers.Dense(16, activation='relu') layers.Dense(1, activation='sigmoid') optimizer='adam' loss='binary_crossentropy' metricas=['acc'] epochs = 20	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	Métricas	503.592	81.060	422.532	0,81	0,81	0,85	0,97	0,57	26,24	0,15	0,82
31	Voting Classifier (GBC+GBN+RL+RF)	Gradient Boosting Classifier Gaussian NB) Logistic Regression Random Forest	Gradient Boosting Classifier Gaussian NB) Logistic Regression Random Forest	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	class_weight='balanced' Métricas	229.655	37.793	191.862	0,82	0,81	0,94	0,96	0,61	76,51	0,43	0,82
32	Voting Classifier (GBC+GBN+RL)	Gradient Boosting Classifier Gaussian NB) Logistic Regression	Gradient Boosting Classifier Gaussian NB) Logistic Regression	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	class_weight='balanced' Métricas	229.655	37.793	191.862	0,82	0,81	0,83	0,93	0,62	63,81	0,36	0,81
33	Voting Classifier (GBC+GBN+RF)	Gradient Boosting Classifier Gaussian NB) Random Forest	Gradient Boosting Classifier Gaussian NB) Random Forest	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	class_weight='balanced' Métricas	229.655	37.793	191.862	0,82	0,81	0,94	0,96	0,61	71,50	0,40	0,82
34	Voting Classifier (GBC+GBN+RL)	Gradient Boosting Classifier Gaussian NB) Logistic Regression	Gradient Boosting Classifier Gaussian NB) Logistic Regression	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk','HighBP','HighChol', 'FruitAndVegCon'	SMOTE Métricas	229.655	37.793	191.862	0,81	0,81	0,94	0,97	0,58	72,74	0,41	0,81
35	Voting Classifier (GBC+GBN+RL)	Gradient Boosting Classifier Gaussian NB) Logistic Regression	Gradient Boosting Classifier Gaussian NB) Logistic Regression	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18 'CatBMI','HeartDis','PhysExer','GenHealth','CogDif', 'Depression','PhysHlth','WalkDif','Gender','AgeRange', 'EdLevel','SocClass','Asthma','Arthritis','SmokerTrad', 'AlcDrinker','Race','LastMedChk'	class_weight='balanced' Métricas	503.592	81.060	422.532	0,81	0,81	0,83	0,94	0,61	145,94	0,82	0,81

Fuente: Elaboración propia

Anexo H. Métricas y clasificación de los modelos obtenidos

Tabla 6. *Modelos, métricas y clasificación.*

	Algoritmo	DataSet	Atributos	Técnica Balanceo	Registros	Diabeticos	No Diateticos	F1 Score	Precisión (Precision)	Exactitud (Accuracy)	Espedificidad (Specificity)	AUC-ROC	Tiempo Generación Modelo (sg)	Tiempo Normalizado	Índice de Calidad
2	Redes Neuronales Artificiales Densas	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	Métricas	503.592	81.060	422.532	0,81	0,81	0,85	0,97	0,57	26,24	0,15	0,82
3	Voting Classifier (GBC+GBN+RF)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	class_weight='balanced' Métricas	229.655	37.793	191.862	0,82	0,81	0,84	0,96	0,61	71,50	0,40	0,82
4	Gradient Boosting Machines (GBM)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	Métricas	229.655	37.793	191.862	0,81	0,81	0,84	0,97	0,59	82,07	0,46	0,82
5	Redes Neuronales Artificiales Densas	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	Métricas	229.655	37.793	191.862	0,81	0,82	0,85	0,97	0,59	12,87	0,07	0,82
6	Redes Neuronales Artificiales Densas	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	25	Métricas	229.655	37.793	191.862	0,81	0,82	0,85	0,98	0,58	13,15	0,07	0,82
7	Voting Classifier (GBC+GBN+RL+RF)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	class_weight='balanced' Métricas	229.655	37.793	191.862	0,82	0,81	0,84	0,96	0,61	76,51	0,43	0,82
8	Voting Classifier (GBC+GBN+RL)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	SMOTE Métricas	229.655	37.793	191.862	0,81	0,81	0,84	0,97	0,58	72,74	0,41	0,81
9	Regresión Logistica (Logistic Regression)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	Métricas	229.655	37.793	191.862	0,81	0,81	0,84	0,97	0,58	0,87	0,00	0,81
10	Regresión Logistica (Logistic Regression)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	Métricas	503.592	81.060	422.532	0,81	0,81	0,84	0,98	0,57	2,79	0,02	0,81
11	Gradient Boosting Machines (GBM)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	Métricas	503.592	81.060	422.532	0,81	0,81	0,85	0,98	0,57	178,96	1,00	0,81
12	Conjunto de árboles (Random Forest)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	Métricas	229.655	37.793	191.862	0,81	0,80	0,84	0,96	0,58	8,05	0,04	0,81
13	Voting Classifier (GBC+GBN+RL)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	class_weight='balanced' Métricas	229.655	37.793	191.862	0,82	0,81	0,83	0,93	0,62	63,81	0,36	0,81
14	Navie Bayes Gaussiano	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	Métricas	229.655	37.793	191.862	0,81	0,80	0,83	0,94	0,60	0,07	0,00	0,81
15	Navie Bayes Gaussiano	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	Métricas	503.592	81.060	422.532	0,81	0,80	0,83	0,95	0,59	0,11	0,00	0,81
16	Conjunto de árboles (Random Forest)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	Métricas	503.592	81.060	422.532	0,80	0,80	0,84	0,96	0,57	19,66	0,11	0,81
17	Voting Classifier (GBC+GBN+RL)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	class_weight='balanced' Métricas	503.592	81.060	422.532	0,81	0,81	0,83	0,94	0,61	145,94	0,82	0,81

	Algoritmo	DataSet	Atributos	Técnica Balanceo	Registros	Diabeticos	No Diateticos	F1 Score	Precisión (Precision)	Exactitud (Accuracy)	Espedificidad (Specificity)	AUC-ROC	Tiempo Generación Modelo (sg)	Tiempo Normalizado	Índice de Calidad
19	Árboles de Decisión (Decision Tree Classifier)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	Métricas	503.592	81.060	422.532	0,79	0,78	0,81	0,92	0,57	0,52	0,00	0,79
20	Conjunto de árboles (Random Forest)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	class_weight='balanced' Métricas	229.655	37.793	191.862	0,80	0,80	0,79	0,87	0,65	8,25	0,05	0,79
21	Árboles de Decisión (Decision Tree Classifier)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	Métricas	229.655	37.793	191.862	0,79	0,78	0,79	0,89	0,59	0,20	0,00	0,78
22	Conjunto de árboles (Random Forest)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	class_weight='balanced' Métricas	503.592	81.060	422.532	0,78	0,81	0,76	0,81	0,66	19,31	0,11	0,76
23	Conjunto de árboles (Random Forest)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	SMOTE Métricas	229.655	37.793	191.862	0,78	0,80	0,76	0,81	0,66	24,73	0,14	0,76
24	Regresión Logistica (Logistic Regression)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	class_weight='balanced' Métricas	229.655	37.793	191.862	0,75	0,84	0,72	0,71	0,73	2,16	0,01	0,75
25	Árboles de Decisión (Decision Tree Classifier)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	class_weight='balanced' Métricas	229.655	37.793	191.862	0,76	0,77	0,75	0,82	0,60	0,25	0,00	0,75
26	Regresión Logistica (Logistic Regression)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	SMOTE Métricas	229.655	37.793	191.862	0,75	0,83	0,72	0,73	0,71	11,17	0,06	0,74
27	Redes Neuronales Artificiales Densas	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	25	SMOTE Métricas	229.655	37.793	191.862	0,75	0,83	0,72	0,73	0,71	31,96	0,18	0,74
28	Redes Neuronales Artificiales Densas	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	SMOTE Métricas	229.655	37.793	191.862	0,75	0,83	0,71	0,72	0,71	30,98	0,17	0,74
29	Gradient Boosting Machines (GBM)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	SMOTE Métricas	229.655	37.793	191.862	0,75	0,83	0,72	0,72	0,71	163,46	0,91	0,74
30	Regresión Logistica (Logistic Regression)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	class_weight='balanced' Métricas	503.592	81.060	422.532	0,74	0,84	0,71	0,70	0,72	3,13	0,02	0,74
31	Árboles de Decisión (Decision Tree Classifier)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	SMOTE Métricas	229.655	37.793	191.862	0,75	0,78	0,73	0,78	0,62	10,34	0,06	0,74
32	Árboles de Decisión (Decision Tree Classifier)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	class_weight='balanced' Métricas	503.592	81.060	422.532	0,75	0,79	0,72	0,77	0,62	0,57	0,00	0,73
33	Support Vector Machine (SVM)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	class_weight='balanced' Métricas	229.655	37.793	191.862	0,74	0,84	0,70	0,68	0,73	1.601,00	8,94	0,69
34	Support Vector Machine (SVM)	2021DataSet_Diabeticos_NoDiabeticos_Depurado.csv	21	SMOTE Métricas	229.655	37.793	191.862	0,75	0,83	0,71	0,72	0,71	2.749,23	15,36	0,66
35	Support Vector Machine (SVM)	2021_22DataSet_Diabeticos_NoDiabeticos_Depurado.csv	18	class_weight='balanced' Métricas	503.592	81.060	422.532	0,73	0,84	0,69	0,67	0,72	7.738,92	43,23	0,51

Fuente: Elaboración propia