ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA BIOMÉDICA

TRABAJO FIN DE GRADO

MODELADO DE LA PROGRESIÓN A DIABETES UTILIZANDO MÉTODOS DE MACHINE LEARNING

Autor: Carlos Barquero Gómez de la Venta

Tutor: Óscar Barquero Pérez

Cotutor: Rafael García Carretero

Curso académico 2020/2021

“La naturaleza nos ha dado las semillas del conocimiento,

no el conocimiento mismo.”

- Séneca

Agradecimientos

Me gustaría comenzar agradeciendo a mi familia y amigos todo el apoyo durante estos años, han sido fuente de inspiración y ayuda durante todo el grado. Por ello quiero dedicárselo a todos ellos.

Agradecer a la Universidad Rey Juan Carlos la oportunidad de realizar allí mis estudios, así como a todos mis profesores y compañeros.

Y por supuesto a mi tutor y profesor Óscar Barquero, por su buena disposición y paciencia conmigo a la hora de llevar a cabo el proyecto. Agradecer también a Rafael García por la ayuda que nos ha brindado durante el desarrollo del trabajo.

Muchas gracias a todos.

Resumen

Cada día más personas padecen diabetes, se ha producido un aumento significativo de personas afectadas con el paso de los años y se pronostica que en el futuro este número siga creciendo. Es por ello que es de gran importancia desde el punto de vista clínico ser capaces de definir los factores de riesgo asociados para así poder abordarlos. Debemos tener en cuenta que el conocimiento de los factores biológicos y genéticos no es suficiente para explicar este aumento de pacientes que padecen la enfermedad ya que en numerosos estudios se ha plasmado la existencia de asociaciones entre la enfermedad y factores sociales como nivel socioeconómico, ingresos, educación e índice de desarrollo humano (IDH).

El objetivo de este trabajo será desarrollar un modelo predictivo el cual a partir de una base de datos que cuenta con 1647 pacientes progresores y no progresores a enfermedad, sea capaz de predecir qué pacientes desarrollarán la enfermedad en un futuro con el objetivo de ofrecer un tratamiento temprano para así poder predecirlo y actuar a tiempo.

La base de datos utilizada en este trabajo procede de la unidad de hipertensión del Hospital Universitario de Móstoles (HUM). Los pacientes utilizados para este estudio han sido anonimizados. Utilizaremos variables clínicas, analíticas, antropométricas y demográficas para llevar a cabo un análisis descriptivo y predictivo del conjunto de datos. Contamos con datos longitudinales, los datos fueron recogidos durante años para cada paciente.

El primer paso ha sido preprocesar la base de datos. Es un proceso fundamental para obtener datos de mayor calidad a la hora de realizar predicciones y visualizar los datos para llevar a cabo un análisis exploratorio. Contamos con la ayuda de un profesional clínico que nos arrojó la información necesaria para caracterizar valores anómalos en las variables.

Una vez realizado el preprocesado, llevamos a cabo el análisis predictivo. Para ello aplicaremos técnicas de balaceo de clases (undersampling) y utilizaremos un modelo de Random Forest que nos permita predecir si un paciente desarrollará la enfermedad o no.

El modelo de predicción se llevó a cabo en las tres primeras revisiones, para dos modelos los cuales contaban con variables diferentes.

Los resultados arrojados reflejan que a medida que aumentaba el número de revisiones, no mejoraban las prestaciones. Esto se ha debido principalmente al desbalanceo entre clases el cual es un impedimento a la hora de calibrar las probabilidades.

Índice

Índice de tablas

Índice de figuras

Acrónimos y abreviaturas

HbA1c Prueba de hemoglobina glicosilada
HUM Hospital Universitario de Móstoles
TFG Trabajo Fin de Grado
EDA *Exploratory Data Analysis*
NaN *Not a number*
IA Inteligencia atificial
IDF *International Diabetes Federation*
IMC Índice de masa corporal
HDL *High-density lipoproteins*
LDL *Low-density lipoproteins*
GOT Transaminasa glutamico oxalacética
GPT Enzima transaminasa glutámico pirúvica
GGT Prueba de gamma-glutamil transferasa
HOMA *Homeostatic model assessment*
TAS Tensión arterial sistólica
TAD Tensión arterial diastólica
SMOTE *Synthetic Minority Oversampling Technique*
ADASYN *Adaptive synthetic sampling approach for imbalanced learning*
EE.UU Estado Unidos
AUC Área bajo la curva ROC
ROC Curva de característica operativa del recepto
IDH Índice de desarrollo humano
RF Random Forest

1 Introducción y objetivos

En este Trabajo de Fin de Grado (TFG) desarrollaremos un modelo de *Machine learning* que nos permita predecir la progresión a diabetes de diferentes pacientes. En este punto definiremos el contexto y la motivación del trabajo.

1.1 Contexto y motivación

La diabetes es una enfermedad crónica que aparece cuando el páncreas no produce insulina suficiente o cuando el organismo no utiliza eficazmente la insulina que produce. La insulina es una hormona que regula el azúcar en la sangre. El efecto de la diabetes no controlada es la hiperglucemia (aumento del azúcar en la sangre), que con el tiempo daña gravemente muchos órganos y sistemas, especialmente los nervios y los vasos sanguíneos (1).

En 2014, un 8,5% de los adultos (mayores de 18 años) tenían diabetes. En 2016 la diabetes fue la causa directa de 1,6 millones de muertes y en 2012 la hiperglucemia provocó otros 2,2 millones de muertes. Entre 2000 y 2016, se ha registrado un incremento del 5% en la mortalidad prematura por diabetes. En los países de ingresos altos la tasa de mortalidad prematura debida a la diabetes descendió entre 2000 y 2010, para volver a incrementarse entre 2010 y 2016. En los países de ingresos medianos bajos, la tasa de mortalidad debida a la diabetes se incrementó en los dos periodos (1).

La diabetes de tipo 2 se debe a una utilización ineficaz de la insulina por el organismo. La mayoría de las personas con diabetes tienen la de tipo 2, que se debe en gran medida a un peso corporal excesivo y a la inactividad física. Los síntomas pueden ser similares a los de la diabetes de tipo 1, la diabetes tipo 1 es una enfermedad crónica debida a la cual hay un alto nivel de glucosa en sangre, pero son a menudo menos intensos. En consecuencia, la enfermedad puede que se diagnostique varios años después de manifestarse los primeros síntomas, cuando ya han aparecido complicaciones. Hasta hace poco, este tipo de diabetes solo se observaba en adultos, pero en la actualidad ocurre cada vez más en niños (1).

Datos procedentes del IDF Diabetes Atlas informan que la prevalencia mundial de DM ha sufrido un gran aumento, se ha visto duplicada para los hombres (de 4.3 a 9.0%) y ha sufrido un aumento en un 60% para las mujeres (de 5.0 a 7.9%) desde 1980 hasta 2014 (2).

Un total de 425 millones de personas (8.8%) entre los adultos de 20 a 79 años padecen DM. Al expandir el rango de edad a 18-99 años, este número se eleva a 451 millones (un 8.4%) de casos de DM. Para 2045, se espera que estas cifras aumenten hasta 629 millones de personas entre 20-79 años, que equivalen al 9.9% de la población. Este número se eleva a 693 millones de personas al ampliar el rango de edad a entre 18-99 años (3).

Además de la variación existente en la incidencia entre zonas con diferentes latitudes y diferentes países, se ha comprobado la existencia de otros muchos factores independientes a la situación geográfica que influyen considerablemente. A pesar de que las etiologías específicas de la diabetes aún son inciertas, se cree que la condición se desarrolla a partir de una interacción entre el estilo de vida y los factores genéticos. Se ha demostrado que la activación de los genes que predisponen a un individuo a la diabetes requiere la presencia de factores conductuales y ambientales. Cabe destacar que los aumentos más significativos en la DM tipo 2 por ejemplo, se han producido precisamente entre las poblaciones que han experimentado cambios rápidos e importantes en el estilo de vida. Según la IDF, los factores de riesgo para DM tipo 2 pueden clasificarse como no modificables y modificables como se establece a continuación (4):

- Factores de riesgo no modificables: factores genéticos, edad, historial de diabetes mellitus gestacional, síndrome de ovario poliquístico.
- Factores de riesgo modificables: sobrepeso y obesidad, inactividad física, factores nutricionales, intolerancia a la glucosa previamente identificada o glucemia basal alterada, síndrome metabólico, entorno intrauterino, inflamación

La diabetes no solo influye en la salud de quienes la sufren, además de la cantidad de muertes que provoca y el impacto negativo que genera sobre la salud de las personas, la diabetes también influye también en el ámbito económico. Los autores de (5) estimaron que el coste económico global de la diabetes en 2015 fue de 13.1 billones de dólares americanos. Este dato refleja los costes directos (por ejemplo, hospitalización y medicamentos) e indirectos (por ejemplo, pérdida de productividad debido a la morbilidad y la mortalidad prematura) asociados a la patología (6).

Como hemos visto la diabetes está en aumento, limitando cada día más la salud de quienes la padecen. No solo afecta al ámbito de la salud, sino también al económico.

Dado el contexto de este problema de salud pública junto al acceso que existe a día de hoy a datos de historial médico, así como a diferentes herramientas de aprendizaje automático probados con capacidad de predicción en entornos similares al presentado en este TFG surge la motivación de ser capaces de contribuir a los profesionales médicos en la prevención de la enfermedad mediante el desarrollo de un modelo predictivo que sea capaz de clasificar una serie de pacientes en sujetos que desarrollarán la enfermedad o no.

1.2 Objetivos

El objetivo principal de este trabajo ha sido crear un modelo que nos permita realizar predicciones a partir de unos datos clínicos obtenidos en diferentes revisiones para cada paciente de entrada. Implementaremos un modelo de Random Forest para cada revisión, utilizando las predicciones de un modelo como una nueva variable de entrada en el modelo siguiente.

Se realizó un estudio retrospectivo longitudinal con 1794 pacientes cuyos datos proceden de la unidad de hipertensión del Hospital Universitario De Móstoles (HUM) entre 2005 Y 2017. Las variables medidas fueron las siguientes: Edad, Peso, Talla, IMC, Creatinina, Cistatina, HDL, LDL, Triglicéridos, GOT, GPT, GGT, Albuminuria, Ferritina, HOMA, Insulina, Glucemia,

Hb-glicosilada, PCR, Vitamina-d, TAS, TAD, Fecha**.** Para llevar a cabo el objetivo principal, hemos ido llevando a cabo una serie de objetivos más específicos:

- Recolección de los datos.
- Definición de los datos.
- Pre-procesado de la base de datos, donde se llevó a cabo la limpieza y transformación de los mismos.
- Análisis exploratorio de la base de datos a estudiar.
- Diseño del modelo de clasificación *Random Forest.*
- Evaluación de las prestaciones del modelo.

1.3 Metodología y estructura de la memoria

En este apartado presentaremos la metodología llevada a cabo en el trabajo:

- Revisión de la literatura existente acerca del tema a tratar, así como de los conceptos de machine learning utilizados en la realización del TFG.
- Adquisición de los datos procedentes de la unidad de hipertensión del Hospital universitario de Mostolés.
- Pre-procesamiento de la base de datos.
- Análisis exploratorio de los datos.
- Balanceo de clases.
- Diseño y aplicación del modelo de machine learning.
- Presentación y discusión de las prestaciones obtenidas mediante tablas.

El TFG presenta la siguiente estructura dividida en capítulos:

- Capítulo 1: Introducción y objetivos: explicación de las ideas, objetivos y metodología del trabajo.
- Capítulo 2: Conceptos clínicos previos y antecedentes: se describe la enfermedad implicada en el desarrollo del proyecto además de estudios relacionados con el que llevaremos a cabo.
- Capítulo 3: Introducción al machine learning y algoritmos utilizados: breve introducción al machine learning, así como a los modelos utilizados en el trabajo.
- Capítulo 4: Base de datos y esquema de análisis con machine learning: descripción de la base de datos con la que trabajaremos, así como del pre-procesamiento. Explicación del modelo implementado en este TFG.
- Capítulo 5: Resultados: se presentan las prestaciones obtenidas y la discusión de los resultados.
- Capítulo 6: Líneas futuras y conclusiones: en este apartado se presentarán las limitaciones encontradas durante el desarrollo del trabajo, así como soluciones futuras para estos. Se describirán brevemente las conclusiones del proyecto.

2 Conceptos clínicos previos y antecedentes

A continuación, se presentan los conceptos clínicos que serán la base de este proyecto. Definiremos la enfermedad en cuestión, diabetes mellitus tipo 2. Se presentará también el estado del arte, expondremos una serie de estudios previos en los cuales se aplicaron técnicas de machine learning persiguiendo objetivos similares a los de este TFG.

2.1 Diabetes tipo 2

La diabetes tipo 2, también llamada "diabetes no dependiente de insulina" o "diabetes de inicio en la edad adulta", representa del 90 al 95% de todos los casos de diabetes. Esta forma de diabetes abarca tanto a los individuos que tienen una deficiencia de insulina relativa (en lugar de absoluta, la deficiencia absoluta es debida a la destrucción autoinmune de las células ß) y que tienen resistencia a la insulina periférica. Al inicio, y a menudo durante el resto de su vida, existe la posibilidad de que estos pacientes no requieran de tratamiento con insulina para sobrevivir (7).

Existen dos deficiencias metabólicas que caracterizan la diabetes tipo 2: un descenso de la capacidad de los tejidos periféricos para responder a la insulina (resistencia a la insulina) y disfunción de las células ß. Las células ß son un tipo de células del páncreas que se encargan de producir, almacenar y liberar insulina (8). Una disfunción en este tipo de células provoca que no se secrete insulina o que la proinsulina no se convierta en insulina. Generalmente, la resistencia a la insulina es el primer acontecimiento, seguida por grados crecientes de disfunción de las células ß (9).

Una gran cantidad de órganos pueden verse afectados por la diabetes tipo 2, incluidos el corazón, los vasos sanguíneos, los nervios, los ojos y los riñones. Cabe destacar que los factores que aumentan el riesgo de diabetes son factores de riesgo de otras enfermedades crónicas graves. Llevar un control sobre la diabetes y sobre el azúcar en sangre se puede traducir en una reducción en el riesgo de complicaciones o afecciones coexistentes (comorbilidades) (10).

Asociadas a la diabetes existen una serie de complicaciones y comorbilidades: enfermedad del corazón y de los vasos sanguíneos, daño a los nervios (neuropatía) en las extremidades, nefropatía, daño ocular, condiciones de la piel, curación lenta, discapacidad auditiva, apnea del sueño, demencia, enfermedad de Alzheimer (10).

Podemos utilizar la prueba de hemoglobina glicosilada (HbA1c) como diagnóstico de diabetes tipo 2 y prediabetes, consiste en un análisis de sangre en el que se miden los valores medios de glucosa en sangre durante los tres últimos meses (11). La Asociación Estadounidense de Diabetes (ADA) propuso utilizar HbA1c $\geq$ 6,5% para el diagnóstico de diabetes y 5,7 - 6,4% para el riesgo más alto de progresar a diabetes. El umbral de diagnóstico propuesto de 6,5% se basó en el riesgo de retinopatía en diferentes niveles de HbA1c (12).

La prediabetes se refiere a un estado intermedio de hiperglucemia. En este estado, los indicadores glucémicos están por encima de lo establecido como normal, pero inferiores al umbral de diabetes. A pesas de que no existen unos criterios claramente establecidos respecto al diagnóstico de prediabetes, sigue siendo un estado en el que existe un alto riesgo de desarrollar diabetes con una tasa de conversión anual de entre el 5% y el 10% (13). Se sugiere una asociación entre la prediabetes y las complicaciones de la diabetes, tales como nefropatía temprana, neuropatía de fibras pequeñas, retinopatía temprana y riesgo de enfermedad macrovascular (13). Una de las claves para la prevención de la diabetes en personas que se encuentran en un estado de prediabetes es la modificación de su estilo de vida hacia hábitos más saludables, estas modificaciones pueden conducir a una reducción del riesgo relativo de entre el 40% y el 70% (14).

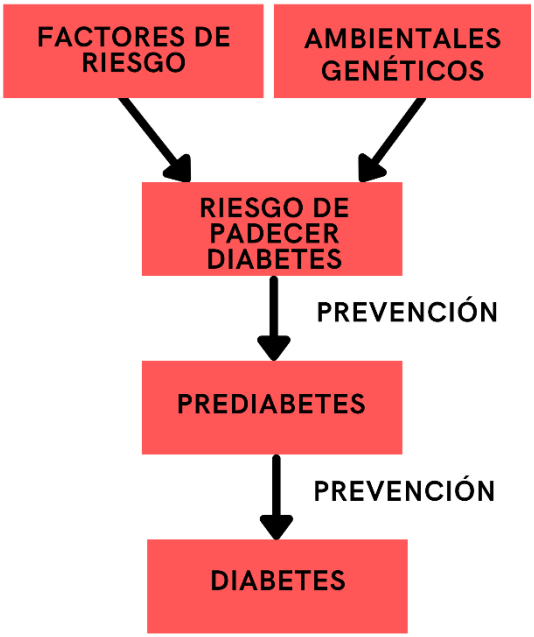

Figura 2.1 De los factores de riesgo a la diabetes.

2.2 Estado del arte

En este apartado revisaremos la literatura existente acerca del problema que vamos a abarcar. El machine learning son un conjunto de técnicas de aprendizaje basadas en datos, cuyo objetivo es obtener modelos que se ajustan utilizando conjuntos de datos etiquetados, es decir, cuya predicción se conoce. A día de hoy es una técnica muy utilizada en aplicaciones médicas, y en concreto para problemas como el estudiado en este TFG. En el capítulo posterior se describen con más detalle las herramientas de machine learning utilizadas en este TFG.

En la literatura podemos encontrar numerosas aplicaciones de machine learning para la clasificación de T2DM:

- En el siguiente trabajo, se propuso como objetivo testar la viabilidad de la utilización de datos obtenidos mediante registros médicos para el desarrollo de modelos de predicción para el riesgo de diabetes. Se utilizó para ello una serie de parámetros demográficos, clínicos y de laboratorio obtenidos de más de dos mil pacientes. Se aplicaron una serie de algortimos de machine learning con los que se evaluó el riesgo de desarrollar diabetes tipo 2 de seis meses a un año después. Los algoritmos utilizados fueron: clasificadores lineales (Gaussian Naïve Bayes y regresión logística), un clasificador basado en muestras (vecino K-más cercano), dos clasificadores basados en árboles de decisión (CART y Bosques aleatorios), y un clasificador basado en kernel (Support Vector Machine). Se logró un AUC superior a 0,8 para predecir la diabetes tipo 2 365 días y 180 días antes del diagnóstico de diabetes, aunque el resultado de AUC es razonable, el valor predictivo positivo fue de 0,24, menor de lo deseado. Por contra, el valor predictivo negativo arrojo unos resultados cercanos al 0,97. Esta diferencia entre resultados es debida al desbalanceo existente entre casos (10%) y controles (90%). En el caso de la diabetes es aceptable un valor predictivo positivo tan bajo, ya que el objetivo es la conducción de los pacientes hacia un estilo de vida saludable, lo que beneficia tanto a pacientes que desarrollarán la enfermedad como a los que no (15).
- En este trabajo (16) utilizaron una serie de modelos de machine learning para ser capaces de predecir diabetes. Los datos fueron obtenidos del Medical Center Chittagong (MCC), Bangladesh. El conjunto de datos está formado por factores de riesgo de diabetes de 200 pacientes. Se emplearon cuatro algoritmos de machine learning, Support Vector Machine (SVM), Naive Bayes (NB), K-Nearest Neighbor (KNN) y C4.5 decision tree (DT), para predecir diabetes mellitus en población adulta. Se obtuvieron las prestaciones para todos los modelos, el modelo que arrojó mejores prestaciones fue C4.5 decision tree con un 72% de tasa de acierto, 74% de sensibilidad y un 72% en f-measure.
- Los autores en el trabajo (17) aplicaron decision tree J48 (DT), K-Nearest Neighbor (KNN), Random Forest (RF) y Support Vector Machine (SVM) como métodos de machine learning a comparar en la predicción de diabetes. Se compararon los resultados en dos situaciones diferentes, antes de preprocesar los datos y después. En el primer caso con decisión tree J48 se obtuvo un accuracy de 73,82%, mayor que con el resto de los clasificadores. En otro caso, después de preprocesar el conjunto de datos, obtenemos tanto con KNN (k = 1) como con Random Forest un rendimiento mucho mejor que los otros tres clasificadores proporcionanando un 100% de precisión

3 Introducción al machine learning y algoritmos utilizados

En este capítulo abordaremos el concepto de machine learning. Profundizaremos en los algoritmos de clasificación utilizados, así como en las métricas que miden las prestaciones de nuestro modelo.

3.1 Introducción al machine learning y conceptos

El machine learning es la ciencia encargada de programar ordenadores para que puedan aprender de datos, es decir, que sean capaces de llevar a cabo una determinada tarea en función de la experiencia (18).

Los métodos de machine learning se pueden clasificar en: sistemas de machine learning en relación a la cantidad y el tipo de supervisión que reciben durante el entrenamiento. Vamos a centrarnos en el aprendizaje supervisado y el aprendizaje no supervisado (18):

- **Aprendizaje supervisado:** En el aprendizaje supervisado, los datos con los que entrenamos al algoritmo incluyen las salidas deseadas, llamadas etiquetas. Existen dos tipos de problemas de aprendizaje supervisado, son clasificación y regresión. Los clasificamos en función de la naturaleza de la variable de salida, si la variable de salida es discreta será un problema de clasificación. Diremos que es un problema de regresión cuando la variable a predecir es discreta.

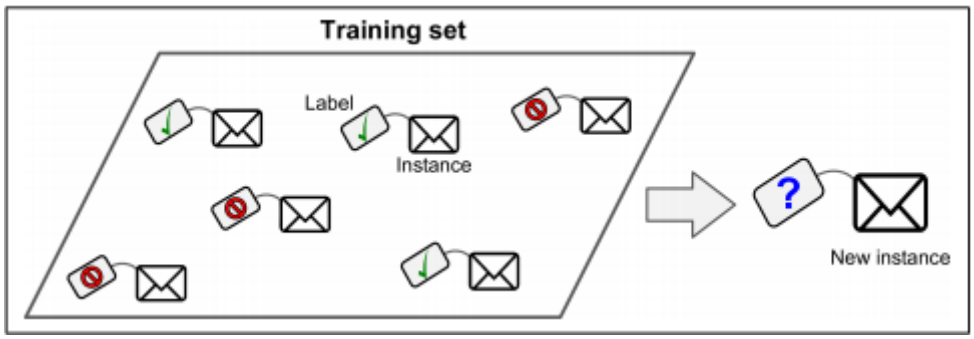

Figura 3.1 Esquema de aprendizaje supervisado. Extraído de (18).

- **Aprendizaje no supervisado:** en este tipo de aprendizaje no contamos con las etiquetas asociadas a los datos. El objetivo será describir asociaciones y patrones entre los datos de entrada (19).

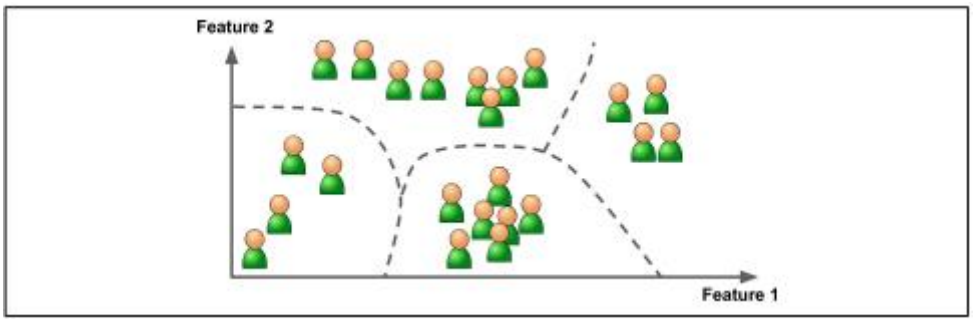

Figura 3.2 Esquema de aprendizaje no supervisado. Extraído de (18).

El problema que tratamos en este TFG es un problema supervisado de clasificación, ya que para cada sujeto tenemos asociada la etiqueta de salida discreta. Es un problema de clasificación binario, cada sujeto puede pertenecer a una clase de las dos existentes y que hacen referencia a ausencia o presencia de la enfermedad. Veamos cómo afrontar un problema de aprendizaje supervisado.

Proceso de entrenamiento y prueba (*training* y *test* en inglés). La fase de entrenamiento y test supone un factor crucial en el éxito del algoritmo de machine learning. Si conseguimos una fase de entrenamiento eficaz, obtendremos un algoritmo de calidad. Los datos son divididos en dos partes, una parte de los datos para entrenar y otra para realizar la fase de test. La fase entrenamiento consiste en una etapa en la que el algoritmo trata de aprender comportamientos y patrones de los datos de entrada (los cuales ya están etiquetados) para más tarde ser capaz de extrapolar lo aprendido a la fase de test, donde trataremos de predecir las etiquetas a partir de unos datos de entrada. Los datos son divididos en base a unas reglas, la división de los datos es uno de los factores críticos en la tasa de éxito del modelo. La partición de los datos varía de acuerdo a la estructura de los datos, aunque siempre debemos utilizar más del 50% de los datos en el conjunto de entrenamiento puesto que un porcentaje menor afectará negativamente a los resultados. La cantidad de entrenamiento y test es el factor más crítico en la tasa de éxito. No se prefiere menos del 50% de los datos de entrenamiento porque los resultados de la prueba se verán afectados negativamente. Una vez que el modelo de machine learning se entrena de acuerdo con los datos de entrenamiento, también se prueba con los datos de entrenamiento. El propósito es construir un algoritmo que a partir de datos ya etiquetado sea capaz de predecir las salidas a partir de otros datos de entrada (20). El siguiente paso es la fase de test, comprobaremos el algoritmo creado con nuestro conjunto de datos de test. Este conjunto contiene entradas que el modelo desconoce, el algoritmo se encargará de predecir la salida asociada a cada dato de entrada. El conjunto de test nos permite evaluar el desempeño de nuestro modelo entrenado en muestras que nunca ha visto y de las que desconoce la salida.

Podemos encontrarnos con problemas a la hora de llevar a cabo este proceso de entrenamiento y test. El ***overfitting*** es uno de los principales problemas en las tareas de machine learning supervisadas. Este ocurre cuando un algoritmo de aprendizaje se ajusta tan bien al conjunto de datos de entrenamiento que se memorizan el ruido y las peculiaridades de los datos de entrenamiento. De esta forma, el modelo de machine learning no es capaz de aprender el patrón subyacente en los datos, por lo que mostrará una baja capacidad de generalización, es decir, una baja capacidad de predicción en datos nunca vistos. Cuando aparece el problema de overfitting, los resultados obtenidos cuando probamos nuestro modelo en la fase de test suelen

ser bastante pobres. La cantidad de datos utilizados para el proceso de aprendizaje es fundamental en este contexto. Los conjuntos de datos pequeños son más propensos a un ajuste excesivo que los conjuntos de datos grandes y, a pesar de la complejidad de algunos problemas de aprendizaje, los conjuntos de datos grandes incluso pueden verse afectados por el ajuste excesivo El sobreajuste de los datos de entrenamiento conduce al deterioro de las propiedades de generalización del modelo y da como resultado un desempeño poco confiable cuando se aplica a datos no vistos. Por otra parte, existe el problema de ***underfitting***. Es lo opuesto a sobreajuste. Ocurre cuando el modelo es incapaz de capturar la variabilidad de los datos, puede ser debido a no tener una cantidad suficiente de muestras con las que entrenar. Si queremos clasificar diversas clases es necesario que tengamos datos de todas ellas y que estás estén equilibradas en cantidad, de este modo reducimos el problema del underfitting. El underfitting ocurre cuando tenemos un modelo que no es capaz de capturar las particularidades y complejidad de los datos, de modo que a la hora de clasificar unos nuevos datos de entrada no será capaz de discernir (21).

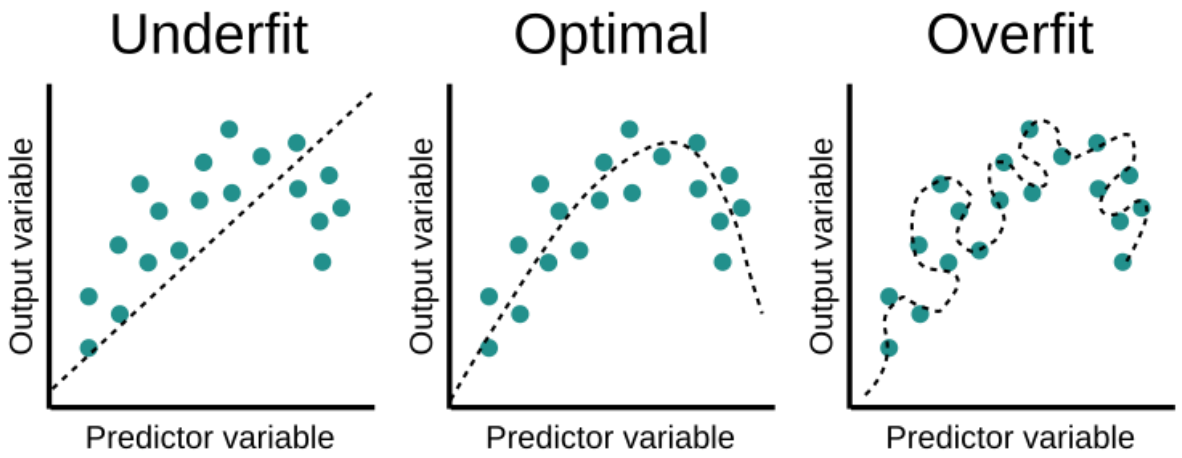

Figura 3.3 Graficós sobre Under-fitting, Appropiate-fitting and Over-fitting. Extraido de (22).

3.2 Modelo de Random Forest para predicción de progresión a diabetes

Para llevar a cabo nuestro proyecto hemos elegido Random Forest como modelo predictivo. Un modelo predictivo puede explicarse como una función matemática con la que obtener resultados mediante la fusión de informaciones (antropométrica, historia clínica, demográfica y biomarcadores del paciente de modo que generamos resultados sobre el riesgo existente en un entorno clínico. A día de hoy existen una gran cantidad de datos y nos encontramos por ello con la oportunidad de poder explotarlos y extraer información valiosa de ellos. El análisis de big data nos permite mejorar la atención, salvar vidas y reducir los costes mediante el descubrimiento de asociaciones, patrones y tendencias de los datos. En el ámbito del cuidado de la salud el análisis de datos nos permite extraer información útil con la que tomar decisiones (23).

El método Random Forest se basa en utilizar un conjunto de árboles de decisión que trabajan

como un conjunto, *ensemble* en inglés. Cada uno de los árboles que compone el conjunto, proporciona una predicción de clase y posteriormente la clase más votada se convertirá en la predcción del modelo Random Forest (ver figura a continuación) (24).

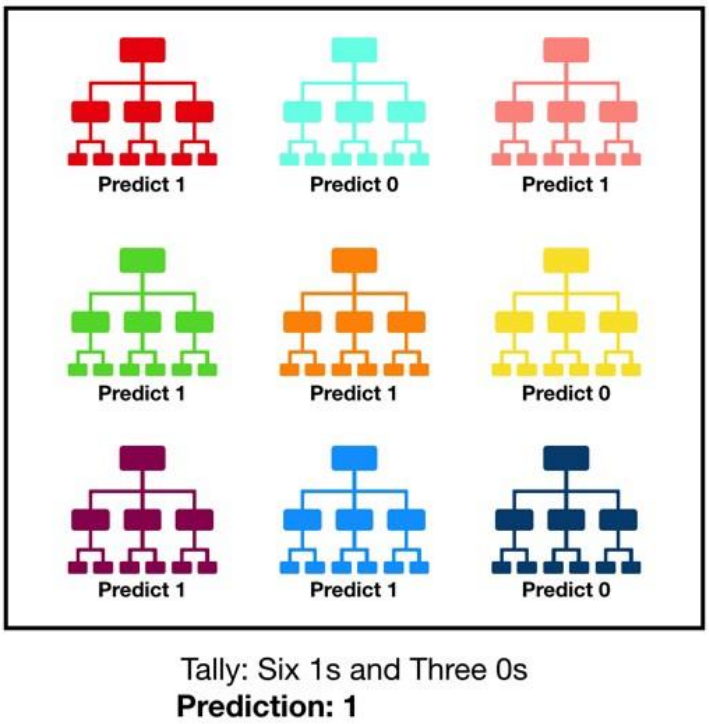

Figura 3.4 Funcionamiento del algortimo Random Forest. Extraído de (24).

3.2.1 Ensemble methods

Antes de continuar explicando el algoritmo Random Forest, vamos a ver qué son los métodos de ensamble, *ensemble methods* en inglés. Estos métodos son un tipo de algoritmo de aprendizaje constituidos por una serie de clasificadores, clasifican nuevos datos mediante el voto (ponderado) de sus predicciones, en el caso de un problema de clasificación. Los algoritmos más recientes de ensamble son *bagging* y *boosting* (25).

El objetivo de estos métodos es obtener una única predicción como salida del modelo mediante la combinación de las diferentes predicciones. En el caso de la clasificación esto se consigue mediante votación (voto ponderado), en el caso de problemas de regresión se trata de calcular el promedio (promedio ponderado). Tanto *bagging* como *boosting* adoptan este enfoque, pero los modelos individuales se tratan de diferente manera en cada uno de ellos. En *bagging*, todos los modelos reciben el mismo peso, mientras que en *boosting*, se da mayor influencia a los más exitosos mediante ponderación (18). Existen más diferencias entre ambos métodos, bagging aprovecha la independencia que hay entre algoritmos simples, ya que al promediar las salidad de los modelos simples podemos reducir la varianza. En boosting buscamos reducir el sesgo, en este caso los modelos se utilizan secuencialmente, de modo que cada modelo dependerá del anterior en función de unos pesos asociados (26). Lo explicamos más en detalle a continuación:

- Bagging: para obtener un conjunto diverso de clasificadores, una de las formas que existen es trabajar con algoritmos de entrenamiento muy diferentes. Otro enfoque sería trabajar con el mismo algoritmo de entrenamiento para todos los predictores, pero cada predictor sería entrenado con diferentes subconjuntos aleatorizados del conjunto de entrenamiento. Si el muestreo se realiza con reemplazo (es decir, un mismo dato puede aparecer repetido varias veces en el conjunto de datos obtenido), este método se denomina *bagging*. Cuando el muestreo se realiza sin reemplazo, se denomina *pasting* (18). Este proceso de muestreo y entrenamiento se representa en la Figura 3.5.

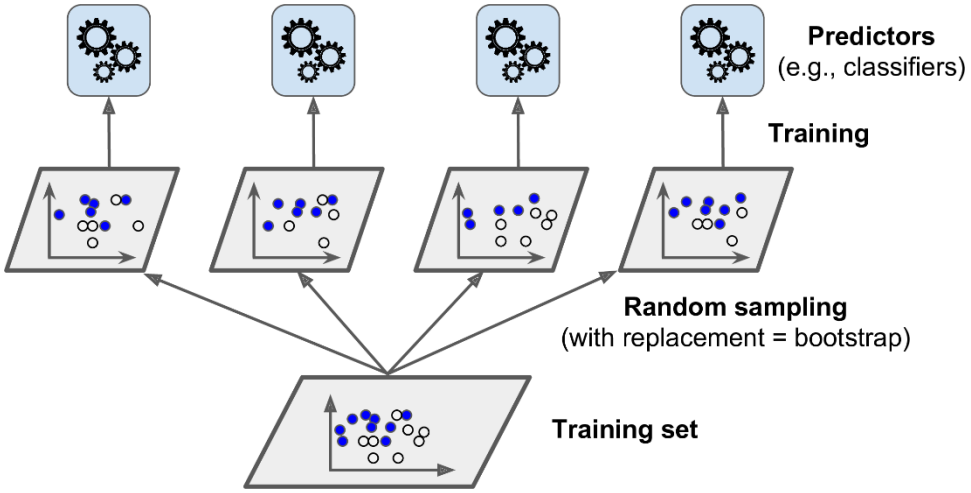

Figura 3.5 Pasting/Bagging training. Extraído de (18).

- Boosting: boosting hace referencia a los métodos de ensamble capaces de combinar una serie de métodos débiles en un método fuerte. Esto consiste en una serie de clasificadores con los cuales iremos realizando las predicciones, a estos clasificadores débiles se les dará un peso diferente en función de la certeza de sus predicciones, es decir, dependiendo de que tan bien clasifiquen. Los clasificadores intentarán mejorar las predicciones de su predecesor con ayuda de los pesos asociados, de modo que la idea principal es entrenar los predictores secuencialmente, tratando de corregir cada uno de ellos a su predecesor obteniendo como clasificador final aquel que es resultado de la combinación ponderada de los clasificadores débiles (18).

Una vez definidos los términos de bagging y boosting es necesario definir los conceptos de sesgo y varianza y ver cómo influyen en los modelos de predicción. Existen tres fuentes de error a la hora de llevar a cabo los modelos de predicción: el sesgo, el error de varianza y el ruido. Los dos primeros se pueden controlar, el ruido sin embargo es inherente a los datos y es irreducible. El objetivo de cualquier algoritmo supervisado de machine learning es lograr un bias bajo y una baja varianza, a su vez, el algoritmo debe lograr un buen rendimiento de predicción (27).

Todos los modelos de aprendizaje estadístico y machine learning sufren el problema de equilibrio entre bias y varianza (28).

El término bias (sesgo) hace referencia a cuánto se alejan en promedio las predicciones de un modelo respecto a los valores reales. Refleja cómo de capaz es el modelo de aprender la relación real que existe entre los predictores y la variable respuesta. Por ejemplo, si la relación sigue un patrón no lineal, por muchos datos de los que se disponga, un modelo de regresión lineal no podrá modelar correctamente la relación, por lo que tendrá un bias alto.

El término varianza hace referencia a cuánto cambia el modelo dependiendo de los datos utilizados en su entrenamiento. Idealmente, un modelo no debería modificarse demasiado por pequeñas variaciones en los datos de entrenamiento, si esto ocurre, es porque el modelo está memorizando los datos en lugar de aprender la verdadera relación entre los predictores y la variable respuesta. Por ejemplo, un modelo de árbol con muchos nodos, suele variar su estructura con que apenas cambien unos pocos datos de entrenamiento, tiene mucha varianza (28).

Cuando aumenta la complejidad del modelo, este dispone de mayor flexibilidad para adaptarse a las observaciones, de este modo se reduce el bias y mejora la capacidad predictiva. A medida que aumenta la complejidad de un modelo, este dispone de mayor flexibilidad para adaptarse a las observaciones, reduciendo así el bias y mejorando su capacidad predictiva. Sin embargo, puede aparecer el problema del overfitting em función del grado de flexibilidad que se alcance, el modelo se ajusta tanto a los datos de entrenamiento que no es capaz de generalizar, es decir, no es capaz de predecir correctamente nuevas observaciones. El modelo óptimo es aquel que consigue un equilibro óptimo entre bias y varianza (29). Los ensemble models buscan reducir la varianza y el sesgo. En *bagging*, una forma de reducir la varianza de las estimaciones es promediando estimaciones de distintos modelos o algoritmos. Si aplicamos *boosting* conseguiremos reducir el sesgo al aprovechar la dependencia entre modelos (28).

Lo ideal es aquel punto en el que el aumento del bias es equivalente a la reducción de la varianza. El objetivo es buscar el compromiso entre bias y varianza que nos permita reducir el error total. Esto es lo que se conoce como compromiso bias/varianza (representado en la figura a continuación).

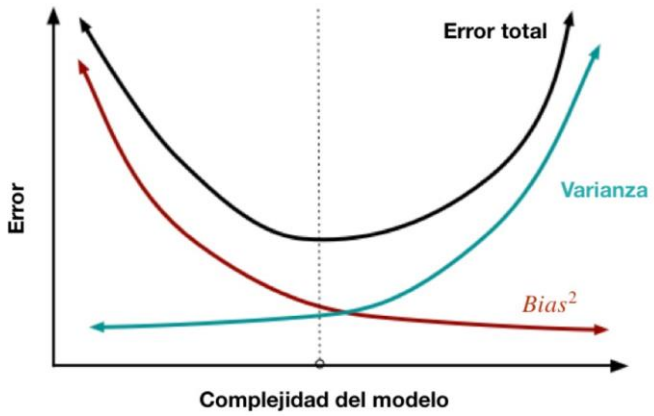

Figura 3.6 Esquema del compromiso bias/varianza. Extraído de (28).

Para llevarlo a nuestro caso, debemos tener en cuenta que los árboles de decisión también sufren de los problemas de sesgo y varianza. Si construimos un árbol pequeño obtendremos un modelo con baja varianza y alto sesgo. Si incrementamos la complejidad del modelo, existirá una reducción en el error de predicción debido a un sesgo más bajo en el modelo. En un punto el modelo será muy complejo y se producirá un sobre-ajuste del modelo el cual empezará a sufrir de varianza alta. El modelo óptimo sería aquel capaz de mantener un balance entre estos dos tipos de errores. A esto se le conoce como "trade-off" (equilibrio) entre errores de sesgo y varianza. El uso de métodos de ensamble es una forma de aplicar este "trade-off" (30).

3.2.2 Árboles de decisión

Los árboles de decisión son una serie de modelos no paramétricos que tienen la posibilidad de utilizarse tanto para regresión como para clasificación, es decir, son capaces de generar una predicción categórica (como si un paciente desarrollará un tipo de enfermedad o no) o una predicción numérica (como el coste de una casa). Los árboles de decisión son modelos flexibles que no aumentan su número de parámetros a medida que agregamos más características (si las construimos correctamente). Normalmente, el objetivo es encontrar el árbol de decisión óptimo mediante la minimización del error de generalización. Podemos definir también otras funciones objetivo, por ejemplo: minimizar el número de nodos o minimizar la profundidad media (31).

Algunas de las ventajas que poseen los árboles de decisión son las siguientes:

- Son relativamente **rápidos** de construir y producen modelos **interpretables** (si los árboles son pequeños).
- En comparación con otros algoritmos de machine learning, los árboles de decisión requieren **menos datos** para entrenar.
- Como dijimos antes, son capaces de realizar **predicciones categóricas** y **numéricas**.
- Son **invariantes** bajo transformaciones (estrictamente monótonas) de los predictores individuales. Como resultado, el escalado y / o las transformaciones más generales no son un problema y son **inmunes** a los efectos de los *missing values* de los predictores.
- Realizan la **selección de características internas** como parte integral del procedimiento. Por lo tanto, son resistentes, si no completamente inmunes, a la inclusión de muchas variables predictoras irrelevantes.

Estas propiedades de los árboles de decisión son en gran parte la razón por la que se han convertido en el método de aprendizaje más popular para la machine learning (19) (32).

Cabe matizar algunos puntos:

Las últimas referencias indican que el uso de algoritmos de árbol de decisión solo es factible en problemas pequeños. En consecuencia, se requieren métodos heurísticos para resolver el problema. En términos generales, estos métodos se pueden dividir en dos grupos: *Top-Down* y *Bottom-Up* con una clara preferencia en la literatura al primer grupo. Hay varios inductores de árboles de decisión *Top-Down* como ID3, C4.5, CART, Algunos de los cuales constan de dos

fases conceptuales: crecimiento y Poda (C4.5 y CART). Otros inductores realizan solo la fase de crecimiento (33).

Mediante una serie de parámetros, podemos controlar la complejidad del modelo. Es preferible un árbol de decisión menos complejo, ya que se considera más comprensible. Además, la complejidad del árbol tiene un efecto crucial en su rendimiento de precisión. La complejidad del árbol se controla explícitamente mediante los criterios de detención utilizados y el método de poda empleado. Por lo general, la complejidad del árbol está determinada por los siguientes hiperparámetros (31):

- El número total de nodos
- Número total de hojas
- Profundidad del árbol
- Número de atributos utilizados

No obstante, lo dicho, los árboles de decisión también poseen una serie de desventajas:

- *Overfitting*: son bastante propensos a ajustarse demasiado a los datos de entrenamiento y pueden ser sensibles a los valores atípicos.

- La capacidad predictiva es, en general, baja comparada con otros modelos de machine learning, De esta forma, a menudo se combinan varios árboles para formar "bosques" para dar lugar a modelos de conjuntos más sólidos.

Cuando se están construyendo, los árboles de decisión se construyen evaluando de forma recursiva diferentes características y utilizando en cada nodo la característica que mejor divide los datos (32).

La siguiente figura muestra la estructura general de uno de estos árboles.

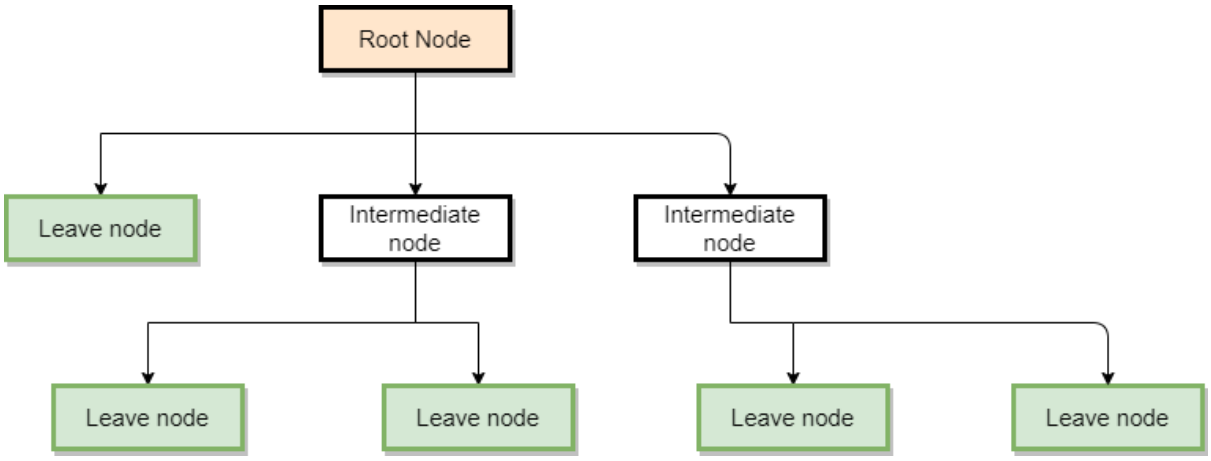

Figura 3.7 Estructura de un árbol de decisión. Extraído de (32).

En esta figura podemos observar tres tipos de nodos (32).

- **El nodo raíz**: es el nodo que inicia el gráfico. En un árbol de decisión normal, evalúa la variable que mejor divide los datos.
- **Nodos intermedios**: son nodos donde se evalúan las variables pero que no son los nodos finales donde se hacen las predicciones.
- **Nodos hoja**: Son los nodos finales del árbol, donde se realizan las predicciones de una categoría o un valor numérico.

El objetivo es generar nodos puros, es decir, nodos que contienen muestras de una única clase. Existen diferentes métricas utilizadas como criterio de partición de los nodos del árbol, nosotros utilizaremos el índice de gini como medida de pureza. La pureza hace referencia a la homogeneidad de clases representadas en un nodo. Un valor pequeño del índice de gini indica que en un nodo existen principalmente observaciones de una única clase por lo que será un nodo más homogéneo y generará menos incertidumbre a la hora de realizar predicciones (34).

3.2.3 Random Forest

Un modelo Random Forest se conforma por un grupo o conjunto (ensemble) de árboles de decisión individuales, generalmente entrenado mediante el método bagging. Cada uno de los árboles es entrenado con unos datos ligeramente diferentes al extraerse una muestra aleatoria de los datos de entrenamiento originales por medio de bootstrapping. El hecho de tener árboles diferentes para realizar la predicción es que buscamos que estén lo más incorrelados posible entre ellos, para que cada predicción aporte la mayor cantidad de información independiente a la predicción global. En cada árbol individual, las observaciones se van distribuyendo por bifurcaciones (nodos) generando la estructura del árbol hasta llegar a un nodo terminal. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que componen el modelo. Cabe destacar que el algoritmo de random forest introduce otra peculiaridad, a la hora de realizar el split en cada nodo de un árbol, busca la característica que mejor divide entre un subconjunto aleatorio de estas. Esto da como resultado una mayor diversidad de árboles que reduce la varianza dando como resultado un mejor modelo (29), (18).

Desde un punto de vista computacional, los bosques aleatorios son atractivos porque (35):

- Se pueden implementar tanto para regresión como para clasificación (multiclase).
- Son relativamente rápidos de entrenar y predecir.
- Dependen de pocos hiperparámetos.
- Se puede calcular una estimación del error de generalización, en la fase de entrenamiento, gracias al bootstrap.
- Se pueden utilizar directamente para problemas de gran dimensión.
- Se puede implementar fácilmente en paralelo.

Presenta también una serie de desventajas (36):

- La visualización gráfica de los resultados puede ser difícil de interpretar, definitivamente más complicada que para el caso de un árbol de decisión.

- Puede sobreajustar en presencia de ruido.
- Las predicciones no son de naturaleza continua y no puede predecir más allá del rango de valores del conjunto de datos usado para entrenar el modelo. En el caso de predictores categóricos con diferente número de niveles, los resultados pueden sesgarse hacia los predictores con más niveles.
- Se tiene poco control sobre lo que hace el modelo (en cierto sentido es como una caja negra).

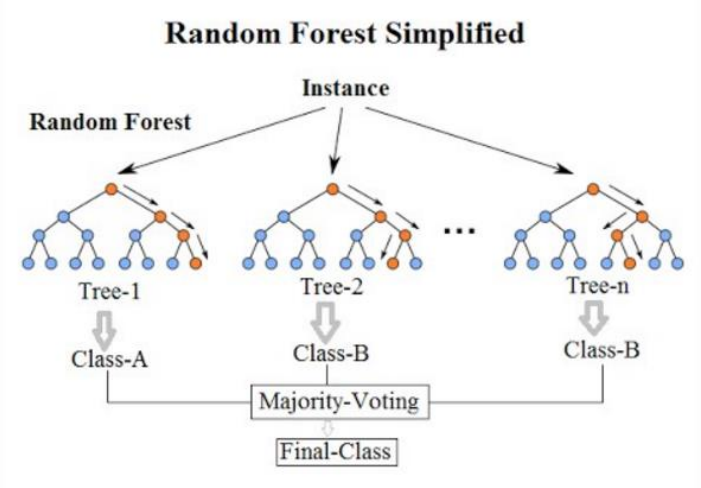

Figura 3.8 Esquema gráfico del funcionamiento del método Random Forest en un problema de clasificación binario. Extraído de (37).

3.3 Evaluación de las prestaciones

Evaluaremos el desempeño de nuestro clasificador mediante la matriz de confusión. Para calcular la matriz de confusión, necesitamos tener un conjunto de predicciones, de modo que puedan compararse con las etiquetas reales.

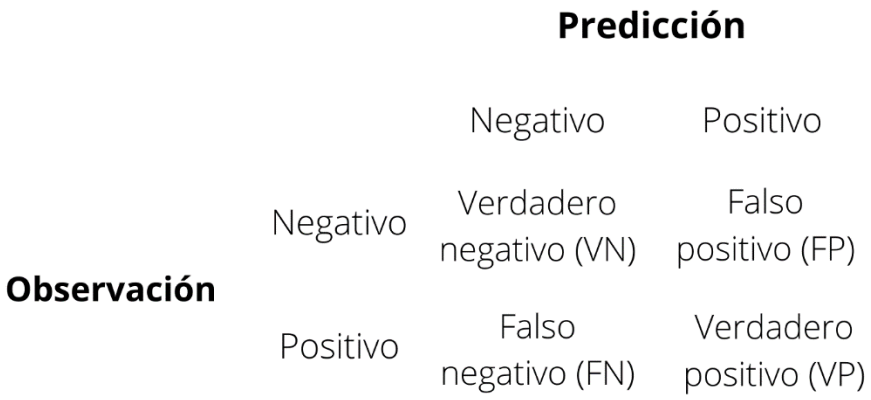

Figura 3.9 Matriz de confusión.

Cada fila en una matriz de confusión representa una clase real, mientras que cada columna representa una clase predicha.

Definamos cada elemento de la matriz de confusión, para un caso de clasificación binaria:

- Verdadero positivo es un resultado en el que el modelo predice correctamente la clase positiva.
- Verdadero negativo es un resultado en el que el modelo predice correctamente la clase negativa.
- Falso positivo es un resultado en el que el modelo predice incorrectamente la clase positiva.
- Falso negativo es un resultado en el que el modelo predice incorrectamente la clase negativa.

A partir de la matriz de confusión podemos conocer la cantidad de veces en las que las clases son correctamente e incorrectamente clasificadas.

La matriz de confusión nos brinda mucha información, pero a veces queremos recurrir a otras métricas, que resumen en forma de un único número el desempeño de nuestro modelo. Por ello calcularemos también la tasa de acierto, la sensibilidad y especificidad del modelo

Definamos las métricas utilizadas en este TFG:

- Área bajo la curva (AUC) ROC: es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Esta curva representa dos parámetros: Esta curva representa la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos (1- especificidad). AUC nos ofrece una medición del área bidimensional por debajo de la curva ROC completa de (0,0) a (1,1) (38).

- Tasa de acierto (*accuracy*). Este valor se calcula como el cociente del número total de muestras correctamente clasificadas y el total de muestras evaluadas.

$$Tasa\ de\ acierto = \frac{VP + VN}{VP + FP + VN + FN} \tag{3.1}$$

- Sensibilidad (*recall*). Se define como el cociente de muestras correctamente clasificadas de la clase positiva y el total de muestras positivas.

$$Sensibilidad = \frac{VP}{VP + FN} \tag{3.2}$$

- Especificidad (*specificity*). Este valor se calcula como el cociente entre el número de muestras correctamente clasificadas de la clase negativa y el total de muestras negativas.

$$Especificidad = \frac{VN}{VN + FP} \tag{3.3}$$

4 Base de datos y esquema de análisis con machine learning

En este capítulo definiremos la base de datos con la que hemos trabajado, así como su análisis y preprocesado previo necesario para la aplicación del modelo de machine learning. Dedicaremos un apartado para explicar la implementación del modelo de machine learning.

4.1 Descripción de la base de datos

Los datos con los que trabajaremos proceden de pacientes de la unidad de hipertensión del Hospital Universitario de Móstoles (Madrid, España). Los pacientes fueron remitidos por su médico de cabecera u otro servicio hospitalario con un diagnóstico de hipertensión esencial. Los datos fueron extraídos de registros electrónicos posteriores a las visitas ambulatorias. Una vez realizada la primera cita, el paciente regresaba a los 6 meses. Se consideró hipertensión:

- Una presión arterial sistólica (PAD) superior a 138 mmHg y / o diastólica superior a 89 mmHg se consideraba hipertensión, según se registró con un dispositivo Omron HEM-907 (Omron Healthcare Inc., Bannockburn, IL, EE. UU.),
- Si el paciente estaba tomando medicamentos antihipertensivos.

Se evaluaron los registros demográficos, clínicos y de laboratorio de los pacientes en la unidad de hipertensión. En la primera visita se registraron el sexo, la edad, el índice de masa corporal (IMC), la PAS y la PAD. También se registraron colesterol LDL y HDL de laboratorio, cistatina C, creatinina, cociente albúmina / creatinina en orina, nivel de glucosa en sangre y hemoglobina glucosilada (HbA1c). Sin embargo, los registros médicos de nuestra base de datos eran heterogéneos e incluían datos faltantes (39).

Tenemos 1794 archivos cada uno de los cuales hace referencia a un sujeto, cada archivo está identificado mediante un número de historia único para cada paciente. Estos archivos no contienen cabeceras, cada variable es una columna siendo las variables las siguientes: Edad, Peso, Talla, IMC, Creatinina, Cistatina, HDL, LDL, Trigliceridos, GOT, GPT, GGT, Albuminuria, Ferritina, HOMA, Insulina, Glucemia, Hb-glicosilada, PCR, Vitamina-d, TAS, TAD, Fecha.

Aparte, tenemos un archivo llamado “pacientes_progresores”, que vincula un número de historia con una clase: 1 (progresores) y 0 (no progresores). Este consta de 1648 números de historia clínica, mientras que por otro lado tenemos los 1794 archivos anteriomente nombrados. Compararemos que números de historia coinciden entre ambos y eliminaremos aquellos pacientes que no aparezcan en el fichero “pacientes_progresores”.

El total de pacientes incluidos en el estudio fue de 1648 (se eliminaron aquellos pacientes los cuales no tenían un número de historia clínica asociado a una clase). Trabajaremos por tanto

con 1648 archivos. Cada archivo contiene 23 variables (anteriormente nombradas), todas ellas numéricas.

Las variables fueron medidas durante años y todos los datos fueron anonimizados antes de ser utilizados en este proyecto. Podemos encontrarnos pacientes con numerosas revisiones y otros con escasas. A medida que aumenta el número de revisiones, la cantidad de datos va disminuyendo.

En la siguiente figura podemos ver el valor medio de cada variable para cada revisión.

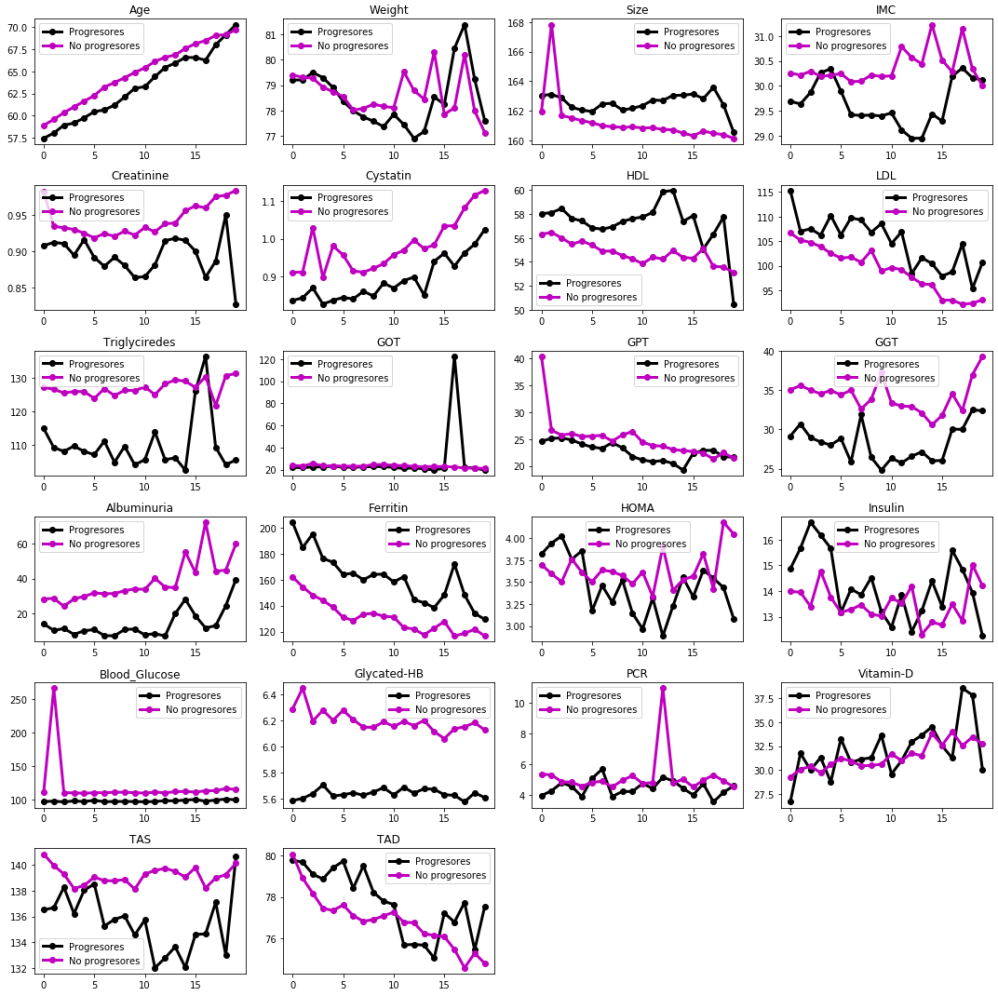

Figura 4.1 Visualización de la media de cada variable en función del número de revisiones.

4.2 Pre-procesado de los datos

A continuación, se explicará detalladamente el pre-procesamiento, donde trataremos tanto los valores perdidos (NaN, del inglés Not a number) como los valores atípicos (*outliers* en inglés).

Ya vistas las variables que componen la base de datos, explicaremos el preprocesamiento llevado a cabo. Este preprocesamiento es necesario, ya que no nos vale solo con tener una gran cantidad de datos, sino que estos a su vez deben ser datos de calidad que permitan ser

explotados para extraer información útil. Perseguimos visualizar, analizar y transformar los datos de modo que queden lo más depurados posibles, sin anormalidades (valores nulos, outliers o missing values), puesto que su presencia puede brindarnos una menor calidad en los resultados.

4.2.1 Creación de las matrices para cada clase

Antes de proceder con la limpieza de la base de datos es necesario que dividamos los pacientes por clases para así poder realizar posteriormente un mejor análisis exploratorio de los datos y ser capaces de realizar comparaciones entre pacientes de diferente clase.

Creamos una función que nos da como resultado dos matrices: matriz de progresores y matriz de no progresores. Ambas son matrices con dimensiones NxRxD, donde N = número de pacientes, R = número de revisión, D = número de características.

4.2.2 Tratamiento de valores atípicos y perdidos

Durante esta etapa trataremos de identificar aquellos valores atípicos u *outliers* que se encuentran fuera del rango dado por los límites que establecen un rango normal y aquellos valores que no han sido registrados o se han perdido (NaN) durante el proceso de toma de datos. En ciertos casos, se tienden a eliminar aquellos pacientes que no tienen datos en todas las variables o que tienen valores atípicos, es decir, obtenemos como resultado una base de datos sin NaN y sin *outliers.* Esto no siempre es conveniente ya que en nuestro caso estaríamos eliminando una gran cantidad de pacientes perdiendo con ello una gran cantidad de información. Para afrontar este problema, hemos llevado a cabo los siguientes pasos:

El primer paso fue la visualización e inspección de las variables mediante diagramas de cajas y bigotes e histogramas, esto nos permite identificar de manera rápida las diferentes anormalidades que puedan existir. Cabe destacar que para cada variable debemos analizar las anormalidades revisión por revisión diferenciando pacientes progresores de no progresores (ver figura a continuación). Cabe destacar que a medida que aumenta el número de revisiones disminuye considerablemente el número de datos disponibles.

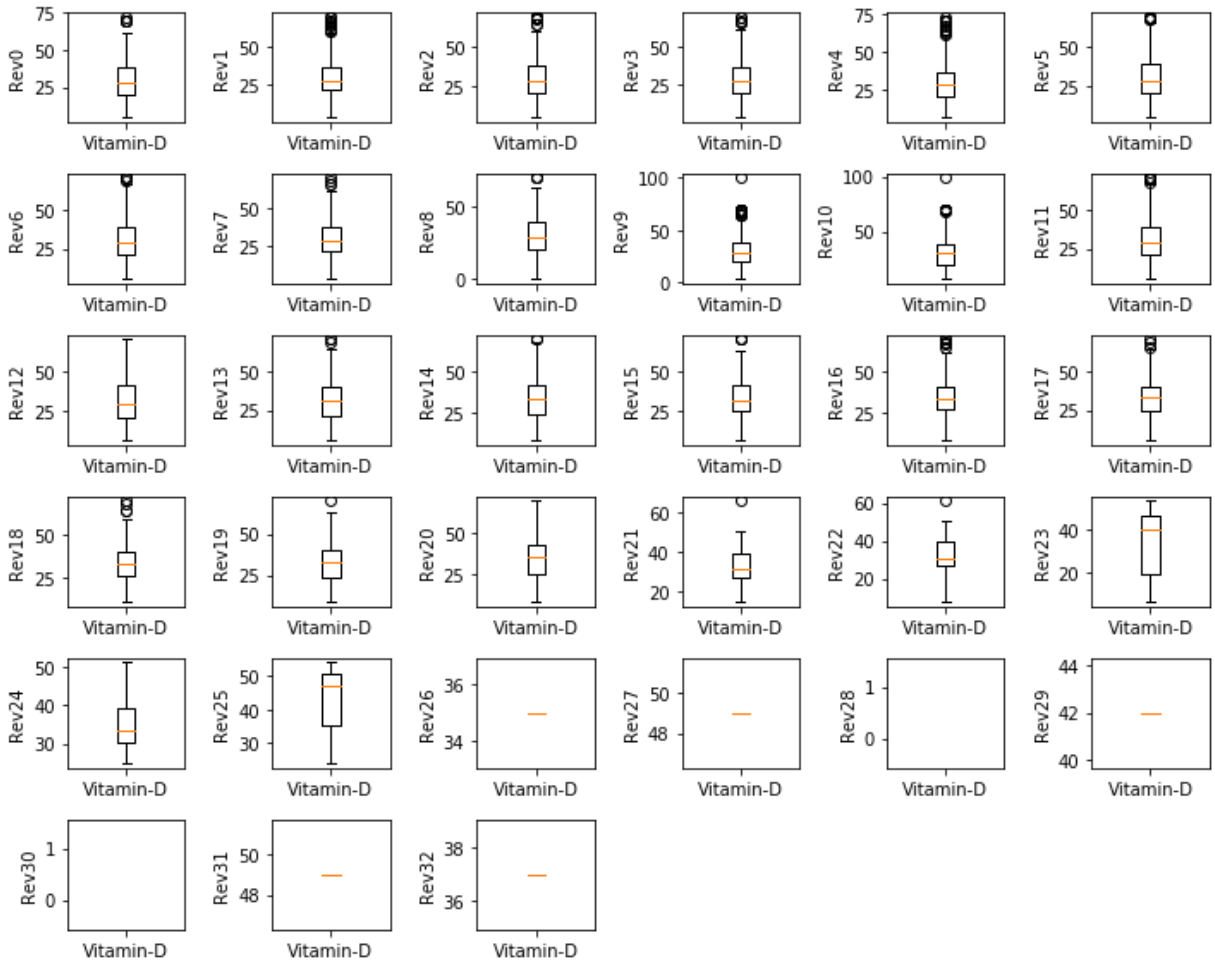

Figura 4.2 Diagrama de cajas y bigotes para todas las revisiones (Vitamina D),(Pacientes no progresores).

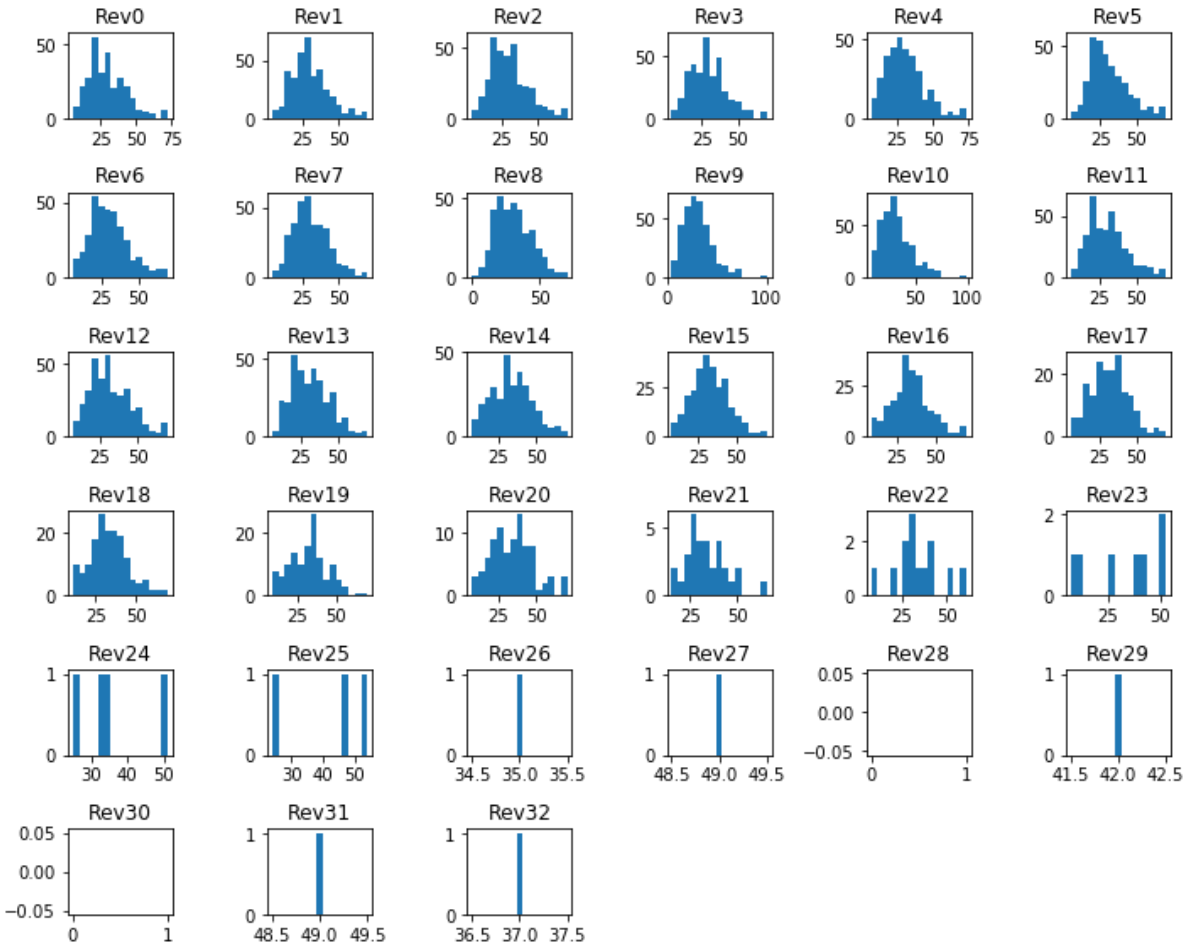

Figura 4.3 Histogramas para todas las revisiones (Vitamina D),(Pacientes no progresores).

Para la detección de los *outliers*, hemos recurrido al siguiente método: método basado en la definición de rangos fisiológicos de las variables definidos por un experto. De esta forma será considerado *outlier* cualquier valor fuera de los límites establecidos por el especialista. Una vez conocidos esos límites, realizamos un gráfico para cada variable donde podemos observar cómo se distribuyen los datos para las variables en función del número de revisiones. De esta manera podemos realizar un esbozo de los *outliers* que existen.

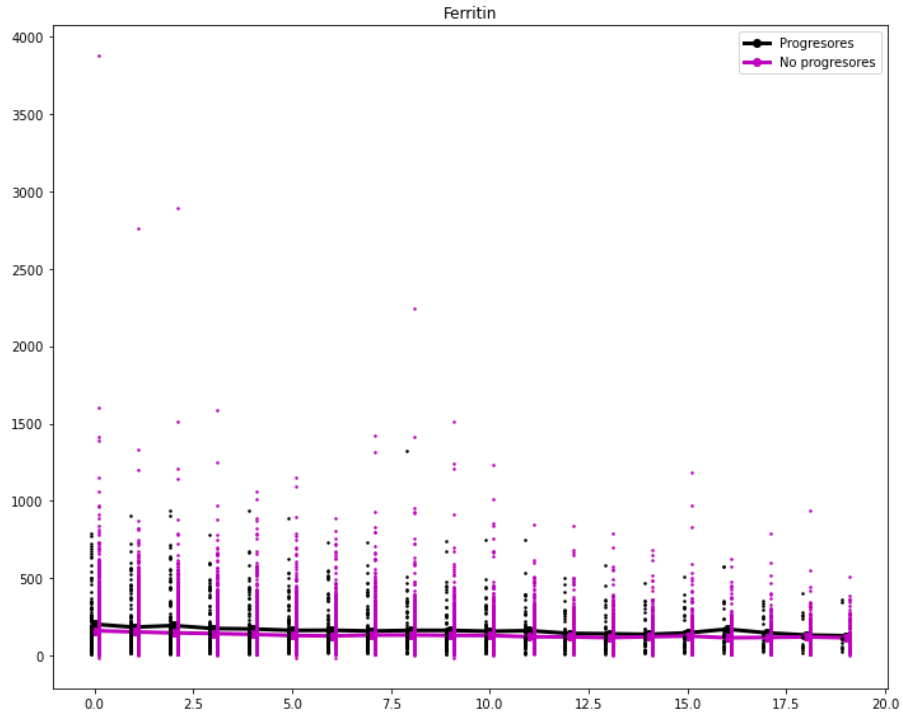

Figura 4.4 Distribución de la variable Ferritina en función del número de revisiones.

Una vez tratados los valores atípicos, debemos hacer frente a los valores perdidos (*missing values*). Son aquellos valores perdidos o no registrados y que hacen que la base de datos este incompleta.

Para llevar a cabo el tratamiento hemos creado una serie de matrices booleanas donde identificaremos en cada una de ellas los NaN y los *outliers.* Estas matrices poseen las mismas dimensiones que las matrices anteriormente creadas (matriz progresores y matriz no progresores). Este proceso lo hacemos tanto para los progresores como para los no progresores.

- La primera matriz es una matriz con True en las posiciones con Nan.
- La segunda matriz es una matriz con True en las posiciones con *outliers*.

Por último, creamos una matriz (una para progresores y otra para no progresores) en la cual encontraremos True en las posiciones con valores válidos y False en aquellos que son NaN o bien *outliers* y que se tienen que imputar en su momento. Esto lo conseguimos creando una matriz en la cual hagamos la unión de la matriz de valores atípicos y la matriz de valores perdidos, realizando también la negación (convertimos los True en False).

4.2.3 Balanceo de clases

Existen casos en los que la base de datos estará desequilibrada, esto es debido a que alguna de las clases posee un mayor número de ejemplos que el resto de las clases. La clase predominante recibe el nombre de clase mayoritaria, mientras que la clase más desprovista de datos se llama

clase minoritaria (40). En nuestro campo, la clase minoritaria hace referencia a la presencia de la enfermedad por lo que será más difícil de detectar debido a su baja frecuencia de aparición. En el ámbito de la salud, clasificar incorrectamente este evento puede tener un alto coste.

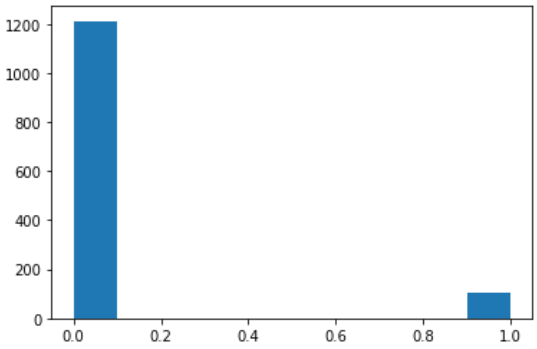

Figura 4.5 Número de pacientes perteneciente a cada clase.

Los métodos más simples existentes para corregir el problema del desbalanceo son el submuestreo, *undersampling* en inglés, y el sobremuestreo, *oversampling* en inglés. Centrándonos primero en el sobremuestreo, cabe destacar que posee un gran inconveniente, y es que puede descartar datos que pueden ser importantes en la fase de aprendizaje. Otra desventaja de este método es que al sobremuestrear generamos copias exactas de datos existentes por lo que el sobreajuste ocurre con mayor frecuencia. Para prevenirlo se han propuesto una serie de soluciones:

- El uso del sobremuestreo de minorías sintéticas (SMOTE). En este método de sobremuestreo los casos sintéticos (copias exactas de las instancias) de la clase minoritaria se generan interpolando instancias de la clase minoritaria adyacentes entre si seleccionadas al azar (k vecinos más cercanos) para sobremuestrear el conjunto de entrenamiento. Los vecinos más cercanos se buscan según distancia euclídea (41).

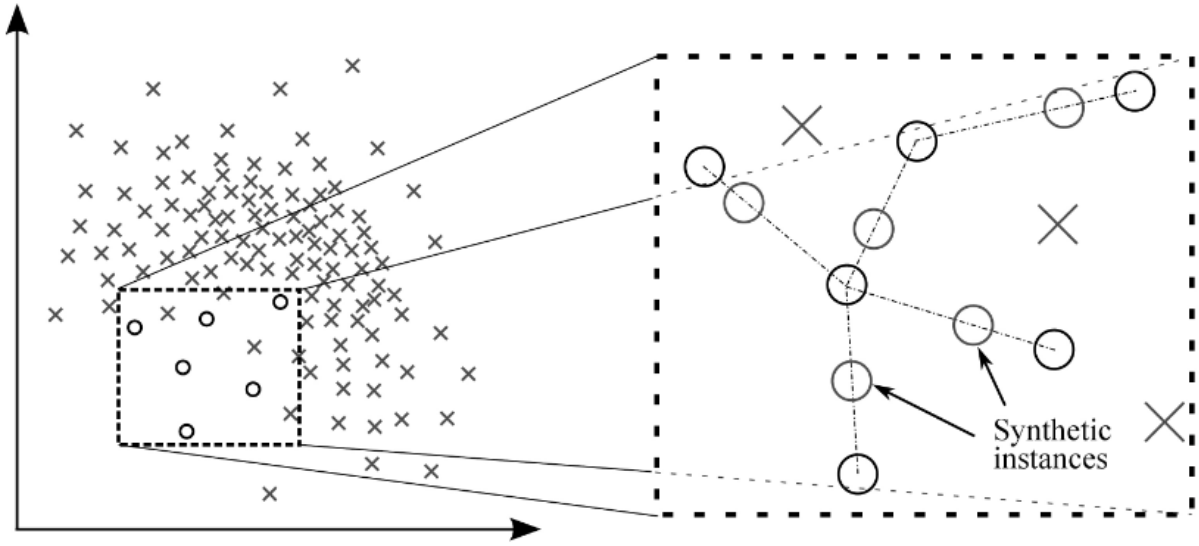

Figura 4.6 Funcionamiento del algoritmo SMOTE. Extraído de (42).

- El otro método existente y que aplicamos junto a SMOTE fue ADASYN. ADAptive SYNthetic (ADASYN) crea datos sintéticos de acuerdo con la densidad de datos. La generacion de nuevos datos sintéticos depende de la densidad, es decir, la generación de los datos es inversamente proporcional a la densidad de la clase minoritaria, de modo que se generan más datos en regiones del espacio de características donde la densidad de datos de la clase minoritaria es baja y menor cantidad o ningún dato en las regiones donde la densidad es alta. Existe un problema con la presencia de valores atípicos, ya que ADASYN al centrarse en regiones donde la densidad de datos es baja, es probable que en esas regiones haya presencia de valores atípicos lo que puede resultar en un mal rendimiento del modelo. Por ello es conveniente eliminar los valores atípicos antes de aplicar este método (43).

En cuanto al submuestro, consiste en reducir los datos eliminando ejemplos pertenecientes a la clase mayoritaria con el objetivo de igualar el número de ejemplos de cada clase (44). Aplicaremos este método mediante la implementación de un Random Forest llamado BalancedRandomForestClassifier, en el cual, el submuestreo se ejecutará sobre cada conjunto Bootstrap generado

4.2.4 Pre-procesado previo a la aplicación del modelo

Antes de implementar el modelo de machine learning realizamos una serie de pasos previos:

- Revisamos si hay alguna característica que tenga más del 50% a NaN, entonces habría que considerar eliminarla. Vamos a considerar que trabajamos sólo con tres revisiones.

Variables	Número total de *missing values* (1º revisión)
Edad	0 %
Peso	1.13 %
Talla	0.45 %
IMC	1.44 %
Creatinina	1.59 %
Cistatina	19.96 %
HDL	8.733 %
LDL	10.25 %
Triglicéridos	1.82 %
GOT	15.03 %
GPT	1.97 %
GGT	4.10 %
Albuminuria	15.86 %
Ferritina	10.09 %
HOMA	72.74 %
Insulina	71.29 %
Glucosa en sangre	2.05 %
Hemoglobina glicosilada	56.18 %
PCR	17.53 %
Vitamina D	81.92 %
TAS	0 %
TAD	0 %

Tabla 4.1 Porcentaje de missing values para cada variable.

- Revisamos si hay algún paciente con más de 8 características NaN, con que esto ocurra en cualquiera de las tres revisiones, deberíamos considerarlo que se elimina, en todas las revisiones.
- Debemos eliminar las variables Blood_Glucose y Glycated-HB de la base de datos, porque son las variables que se utilizan para determinar si un paciente es diabético o no.
- Hay tres variables que presentan un gran número de NaN. Estas variables son HOMA, Insulin y Vitamin-D. Estas son las que más tienen en las tres primeras revisiones. Realizaremos la imputación de los valores por la mediana.

4.3 Diseño experimental del modelo Random Forest

En este apartado, se definirán los pasos seguidos para llevar a cabo la implementación del modelo de predicción.

Para llevar a cabo el modelo se ha utilizado la base de datos definida en la sección 4.1. Disponemos de una base de datos con 1647 pacientes, siendo 1515 (92%) progresores a diabetes y 132 (8%) no progresores. Las variables que conforman la base de datos son las siguientes: Edad, Peso, Talla, IMC, Creatinina, Cistatina, HDL, LDL, Triglicéridos, GOT, GPT, GGT, Albuminuria, Ferritina, HOMA, Insulina, Glucemia, Hb-glicosilada, PCR, Vitamina-d, TAS, TAD, Fecha.

Hemos realizado una partición de los datos para obtener los conjuntos de entrenamiento y test: un 80% de los datos conforman el conjunto de entrenamiento y el 20% restante el conjunto de test.

El modelo inicial se ha entrenado utilizando 50 árboles (n_estimators = 50) y manteniendo el resto de hiperparámetros con su valor por defecto. Al ser hiperparámetros, no se puede saber de antemano cuál es el valor más adecuado, la forma de identificarlos es mediante el uso de estrategias de validación, por ejemplo, validación cruzada.

La validación cruzada (CV, del inglés cross-validation) es una estrategia popular para la selección de algoritmos. La idea principal detrás de CV es dividir los datos, una o varias veces, en dos subconjuntos: train y valdación para estimar el riesgo de cada algoritmo: parte de los datos (la muestra de entrenamiento) se usa para entrenar cada algoritmo, y la parte restante (la muestra de validación) se usa para estimar el riesgo del algoritmo. Luego, CV selecciona el algoritmo con el menor riesgo estimado. En comparación con otras técnicas, CV evita el sobreajuste porque la muestra de entrenamiento es independiente de la muestra de validación (45).

Nombraremos aquellos hiperparámetros los cuales consideramos relevantes a la hora de implementar el modelo y por ello debemos conocer cuál es su valor más adecuado. Para hacer esta búsqueda hemos recurrido a *Grid Search* basado en validación cruzada.

GridSearchCV es una clase disponible en scikit-learn que permite evaluar y seleccionar de forma sistemática los parámetros de un modelo. Indicándole un modelo y los parámetros a probar, puede evaluar el rendimiento del primero en función de los segundos mediante validación cruzada. Hemos utilizado 7-CV. CV indica el número de pliegues que se deben usar para la validación cruzada.

Más pliegues de CV reducen las posibilidades de sobreajuste, pero aumentarlo aumentará el tiempo de ejecución.

Debemos probar una amplia gama de valores y ver qué funciona. Intentaremos ajustar el siguiente conjunto de hiperparámetros:

- Máximo número de variables a considerar para la ramificación = número máximo de características consideradas para dividir un nodo.

- Mínimo número de muestras para dividir un nodo = número mínimo de muestras necesarias antes de dividir este nodo.
- Mínimo de muestras por hoja = número mínimo de muestras que debe haber en un nodo final (hoja).

Establecemos un rango de valores sobre los que se realizará la búsqueda del valor óptimo de los hiperparámetros:

- Máximo número de variables a considerar para la ramificación = ['auto', 'sqrt','log2']
- Mínimo número de muestras para dividir un nodo = [2, 5, 10]
- Mínimo de muestras por hoja: 1, 3, 5, 10 o 25 muestras por hoja = [1, 2, 4,8,10,12,14,16,18,20]

El hecho de restringir los valores de búsqueda y utilizar únicamente los tres hiperparámetros anteriores es debido a que nos permite manejar el sobreajuste y mejorar con ello las prestaciones. Escoger un rango de búsqueda más amplio (lo que conlleva un mayor coste computacional) no supone, en general, un incremento en la mejora de rendimiento.

Una vez que hemos conseguido los valores ideales de los hiperparámetros, implementamos un nuevo modelo *Random Forest* con los valores devueltos por GridSearch. Este nuevo modelo cuenta con 1000 árboles. Este será el modelo utilizado para entrenar nuestros datos.

Todos los pasos definidos a continuación se aplicaron inicialmente sobre los datos obtenidos en la primera revisión.

En primer lugar, definiremos dos modelos con los que trabajar:

- Modelo en el cual han sido eliminadas las variables Blood Glucose y Glycated HB, ya que son variables predictoras de diabetes.
- Modelo en el cual han sido eliminadas las variables adicionalmente las variables Vitamin D, Insulin y HOMA puesto que estas variables poseen más de la mitad de los valores como NaN.

Una vez definidos los modelos, probaremos estrategias de balanceo de clases sobre ambos. Las estrategias de balanceo de clases serán las siguientes:

- Oversampling: SMOTE y ADASYN.
- Undersampling.

Por último, una vez elegida la técnica de balanceo de clases que mejores prestaciones arroja, aplicamos el algoritmo Random Forest sobre los dos modelos previamente creados. Esto lo repetimos para las tres primeras revisiones. El esquema de trabajo es el siguiente (ver figura a continuación):

- Aplicamos el modelo random forest a la primera revisión.
- Guardamos las predicciones de clase arrojadas por el modelo para utilizarlas como nueva variable de entrada en la revisión posterior.
- Entrenamos la segunda revisión mediante el modelo random forest habiendo añadido las predicciones de clase de la revisión anterior como nueva variable de entrada al modelo.

- Vemos si mejoran las prestaciones a medida que aumentamos el número de revisiones y repetimos sucesivamente los pasos para las revisiones posteriores.

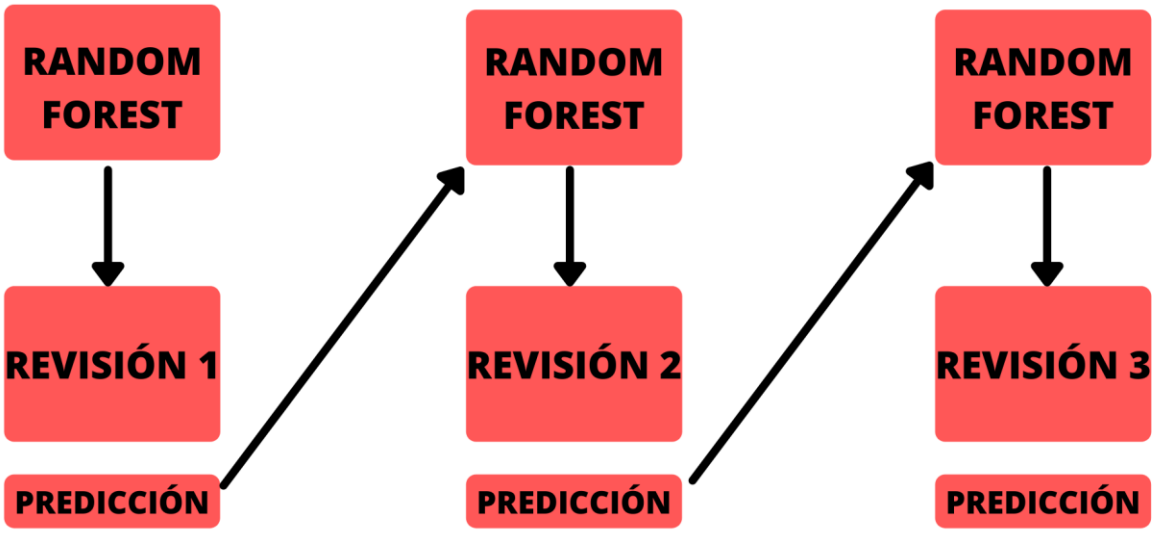

Figura 4.7 Esquema de trabajo llevado a cabo para la aplicación de random forest en las tres primeras revisiones.

5 Resultados

5.1 Resultados obtenidos tras el preprocesado

Una vez aplicado el preprocesado explicado en la sección 4.2.2 obtenemos la gráfica de cómo quedan las variables distribuidas a lo largo de las revisiones sin NaN y sin outliers.

Vemos que con respecto a la gráfica obtenida antes de aplicar el preprocesado obtenemos gráficas sin tantos picos debidos a los NaN y outliers.

Hay una serie de variables que se mantienen reativamente estables durante el tiempo como por ejemplo Cystatin, HDL, Blood Glucose y Glycated HB.

Existen variables que van muy juntas para progresores y no progresores e incluso se cruzan como por ejemplo HOMA, Insulin, TAS y TAD. Son variables que van cambiando mucho en las diferentes revisiones.

Hay variables que divergen al final de las revisiones como HDL, GGT, Albuminuria y Triglycerides.

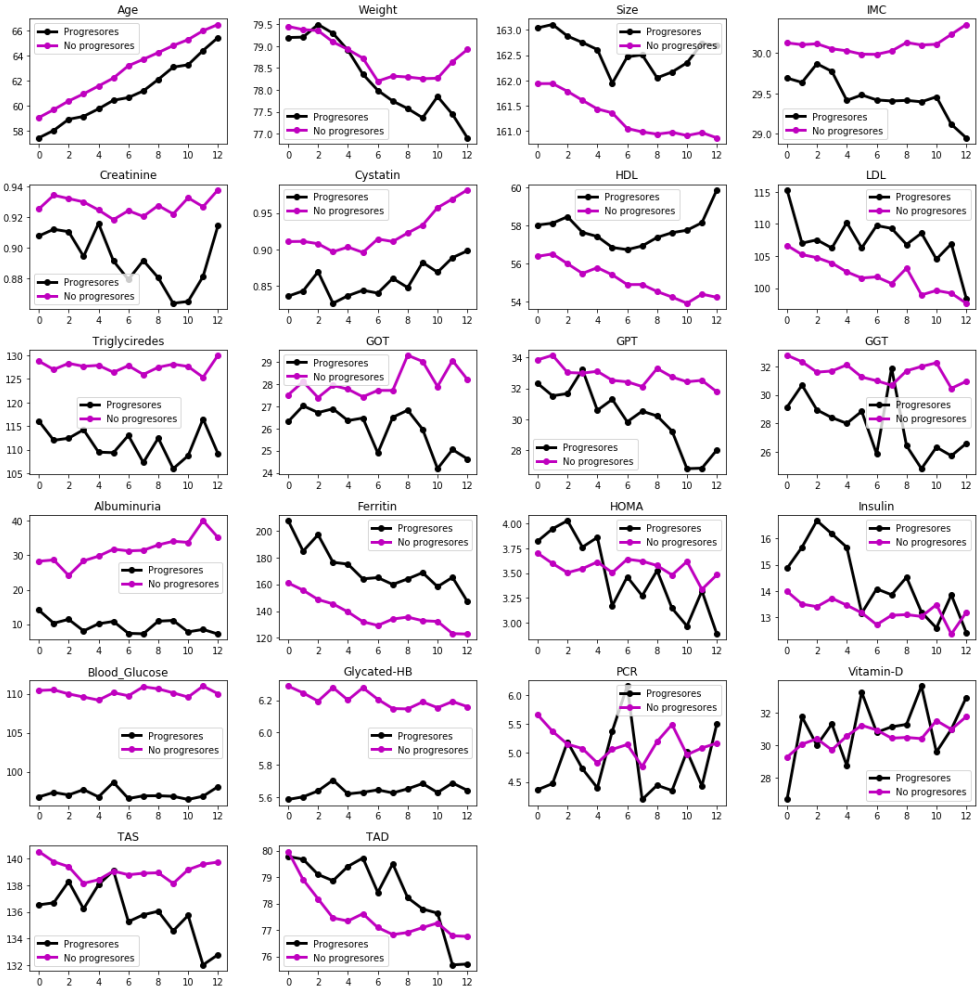

Figura 5.1 Visualización de la media de cada variable en función del número de revisiones sin NaN ni outliers.

5.2 Matriz de confusión y prestaciones obtenidas tras la aplicación del modelo

En este capítulo vamos a detallar y comparar los resultados obtenidos tras la aplicación de los diferentes modelos de RF. En primer lugar, compararemos las prestaciones obtenidas al aplicar técnicas de oversampling vs undersampling, descritas en la sección 4.2.3.

Los resultados que compararemos a continuación se han obtenido únicamente teniendo en cuenta la primera revisión. Esto lo hacemos para elegir la técnica de balanceo de clases que mejores resultados nos arroje y así trabajar con ella en revisiones posteriores.

Para facilitar la nomenclatura, llamaremos modelo 1 al modelo en el que eliminamos Blood Glucose y Glycated HB y modelo 2 al modelo en el que eliminamos también Vitamin D, Insulin y HOMA.

En primer lugar, vamos a comparar las prestaciones (en el conjunto de entrenamiento) habiendo eliminado las variables Blood Glucose y Glycated-HB.

Prestaciones en Train	**SMOTE**	**ADASYN**	**BALANCED RF**
Tasa de acierto	1	1	0.70
Sensibilidad	1	1	1
Especificidad	1	1	0.67

Tabla 5.1 Prestaciones obtenidas en el conjunto de entrenamiento para el modelo 1.

Hacemos los mismo para el conjunto de test.

Prestaciones en Test	**SMOTE**	**ADASYN**	**BALANCED RF**
Tasa de acierto	0.89	0.88	0.58
Sensibilidad	0.07	0.07	0.70
Especificidad	0.96	0.96	0.57

Tabla 5.2 Prestaciones obtenidas en el conjunto de test para el modelo 1.

Ahora, vamos a comparar las prestaciones (en el conjunto de entrenamiento) habiendo eliminado también las variables Insulin. HOMA y Vitamin-D.

Prestaciones en Train	**SMOTE**	**ADASYN**	**BALANCED RF**
Tasa de acierto	1	1	0.69
Sensibilidad	1	1	1
Especificidad	1	1	0.66

Tabla 5.3 Prestaciones obtenidas en el conjunto de entrenamiento para el modelo 2.

Hacemos los mismo para el conjunto de test.

Prestaciones en Test	**SMOTE**	**ADASYN**	**BALANCED RF**
Tasa de acierto	0.89	0.89	0.59
Sensibilidad	0.14	0.18	0.70
Especificidad	0.95	0.95	0.58

Tabla 5.4 Prestaciones obtenidas en el conjunto de test para el modelo 2.

A vista de los resultados obtenidos, podemos observar que los modelos de oversampling se están sobreajustando a los datos, ya que los tan buenos resultados en el conjunto de entrenamiento no son extrapolables al conjunto de test. En este punto decidimos por tanto utilizar undersampling como la técnica de balanceo de clases. La manera en la que aplicaremos undersampling será mediante la creación de un modelo de Random Forest llamado BalancedRandomForestClassifier, este tipo de Random Forest, es un método en el cual a cada árbol del modelo se le proporciona una muestra boostrap balanceada.

Una vez elegido el método de balanceo, compararemos las prestaciones obtenidas en el conjunto de test en la primera revisión al eliminar las variables Blood Glucose y Glycated HB con las obtenidas al eliminar estas mismas más Vitamin D, Insulin y HOMA.

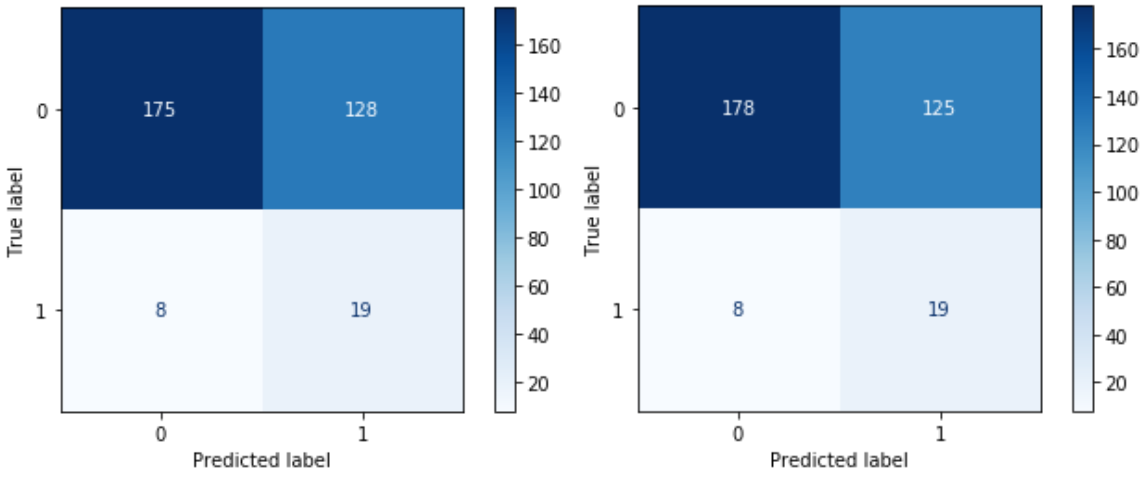

Figura 5.2 Matrices de confusión para modelo 1 y modelo 2.

Prestaciones	**Modelo 1**	**Modelo 2**
Tasa de acierto	0.58	**0.59**
Sensibilidad	0.70	0.70
Especificidad	0.57	**0.58**

Tabla 5.5 Comparación entre las prestaciones obtenidas para ambos modelos en la primera revisión.

A vista de los resultados obtenidos en la tabla 6.5, podemos apreciar que con ambos modelos obtenemos resultados casi idénticos.

Una vez analizadas las prestaciones obtenidas en la primera revisión, procedemos a obtener las mismas para las revisiones siguientes. A medida que aumentamos el número de revisiones, aumentamos el número de variables de entrada al modelo puesto que las predicciones realizadas por el modelo en la revisión anterior a la que estemos estudiando serán una nueva característica de entrada al modelo actual. El hecho de introducir esta nueva variable debería arrojarnos una mejora en las prestaciones a medida que aumentamos el número de revisiones. Compararemos a continuación los resultados obtenidos entre las tres primeras revisiones tanto para el modelo 1 como para el modelo 2.

- Modelo 1:

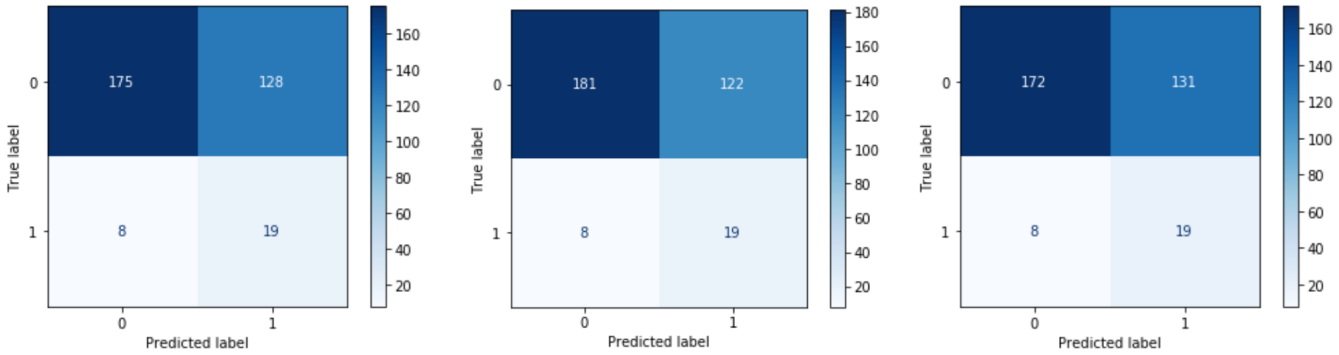

Figura 5.3 Matrices de confusión para modelo 1 en las tres primeras revisiones.

Prestaciones	Modelo 1 rev 1	Modelo 1 rev2	Modelo 1 rev3
Tasa de acierto	0.58	0.60	0.57
Sensibilidad	0.70	0.70	0.70
Especificidad	0.57	0.59	0.56
AUC Score	0.68	0.66	0.61

Tabla 5.6 Comparación entre las prestaciones obtenidas para el primer modelo en las tres primeras revisiones.

Obtenemos una ligera mejoría en la segunda revisión que se mantiene durante la tercera. El modelo no mejora considerablemente a medida que aumentamos el número de revisiones.

- Modelo 2:

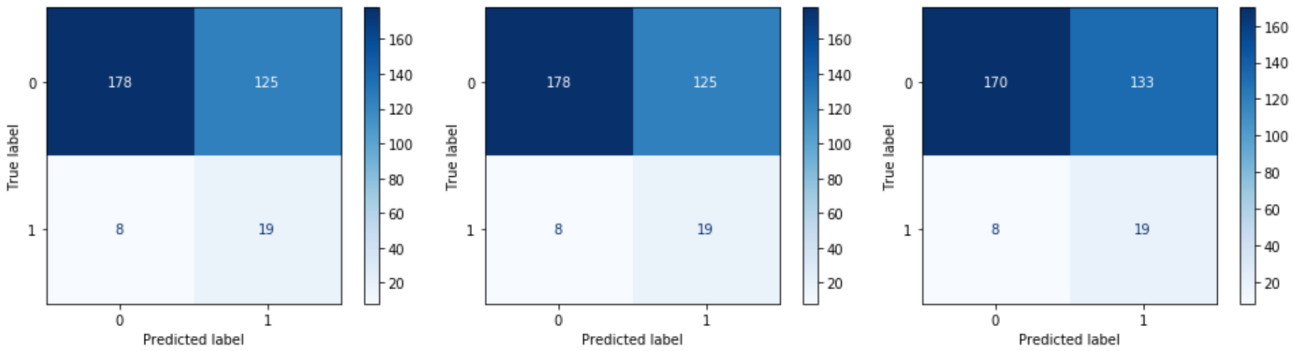

Figura 5.4 Matrices de confusión para modelo 2 en las tres primeras revisiones.

Para el segundo caso, las prestaciones no mejoran de una revisión a otra, de hecho, en la tercera revisión disminuyen con respecto a la anterior.

Prestaciones	Modelo 2 rev 1	Modelo 2 rev2	Modelo 2 rev3
Tasa de acierto	0.59	0.59	0.57
Sensibilidad	0.70	0.70	0.70
Especificidad	0.58	0.58	0.56
AUC Score	0.66	0.63	0.61

Tabla 5.7 Comparación entre las prestaciones obtenidas para el segundo modelo en las tres primeras revisiones.

6 Líneas futuras y conclusiones

En este capítulo expondremos aquellas limitaciones que hemos encontrado a medida que se desarrollaba el trabajo, soluciones propuestas a ellas en otros estudios y trabajo futuro que se puede aplicar para mejorar los resultados obtenidos.

6.1 Conclusiones

Para lograr el objetivo, una vez realizado el preprocesado, obtuvimos las tres primeras revisiones de todos los pacientes. El proceso llevado a cabo fue el siguiente: para cada revisión aplicamos RF a dos modelos diferentes, el primero de ellos habiendo eliminado las variables Blood_Glucose y Glycated-HB puesto que son variables predictoras de diabetes y otro modelo habiendo eliminado junto a las anteriores Vitamin-D, Insulin y HOMA ya que contenían más del 50% de sus valores como NaN. La idea es comparar para que modelo obtenemos mejores prestaciones y si a medida que aumentamos el número de revisiones mejoran las prestaciones. Para ello una vez obtenidas las predicciones de clase arrojadas por el algoritmo de RF en la primera revisión, las almacenamos y las pasamos como nueva variable a los datos de la revisión 2 y así sucesivamente.

Podemos concluir que a medida que aumentamos el número de revisiones las prestaciones no sufren mejoras significativas, lo mismo pasa con los dos modelos. Los resultados obtenidos en todos los casos son similares.

6.2 Limitaciones y líneas futuras

El desbalanceo existente entre los pacientes progresores a enfermedad y los no progresores es significativo e implica la obtención de prestaciones con resultados más pobres. El desbalanceo entre clases supone una gran limitación a la hora de realizar un problema de clasificación. Existen problemas de clasificación binaria en los cuales el coste de fallar a la hora de clasificar es muy alto (por ejemplo, en la detección del cáncer). Por ello en muchas aplicaciones nos interesa conocer la probabilidad de pertenecer a cada clase. Existen modelos de machine learning capaces de estimar el valor de probabilidad que va asociado a cada predicción.

Un modelo calibrado es aquel en el que el valor estimado de probabilidad puede interpretarse directamente como la confianza que se tiene de que la clasificación predicha es correcta. Por ejemplo, si para un modelo de clasificación binaria (perfectamente calibrado) se seleccionan las predicciones cuya probabilidad estimada es de 0.8, en torno al 80% estarán bien clasificadas (46).

Las estimaciones de probabilidad de clase obtenidas a través del aprendizaje supervisado en escenarios desequilibrados subestiman sistemáticamente las probabilidades para instancias de clases minoritarias.

Existen artículos en la literatura que han tratado de lidiar con este problema. En (47) se realiza una revisión de la naturaleza del problema para que sirva como aporte para investigaciones futuras. Más específico como (48), que se centra en cómo los datos desbalanceados afectan en concreto a los árboles de decisión. En (49) se muestra que:

- Los métodos de aprendizaje supervisado no arrojan buenas prestaciones en presencia de datos desequilibrados.
- El submuestreo antes de la calibración mantiene una calibración general razonablemente buena al tiempo que mejora drásticamente la calibración con respecto a las instancias minoritarias.
- Los predictores bagging sobre muestras de bootstrap balanceadas mejoran aún más la calibración y reducen la varianza.

Propone como solución a este problema, inducir estimadores de probabilidad calibrados sobre muestras de bootstrap balanceadas de los datos de entrenamiento.

Respecto a la cantidad de datos existe cierta limitación. A medida que aumenta el número de revisiones disminuye el número de datos recolectados para los pacientes por lo que contamos con menos información a explotar.

En este TFG hemos llevado a cabo un modelo predictivo basado en Random Forest que nos permitiera predecir aquellos pacientes que progresarán a diabetes. Trabajamos con Random Forest ya que es un algoritmo con un buen balance entre complejidad y resultados, puede ser utilizado para reducción de la dimensionalidad ya que puede estimar los predictores más importantes y es una técnica muy certera en bases de datos muy grandes. Para que el algoritmo arrojará mejores resultados, recurrimos a técnicas de balanceo de clases (undersampling) ya que de otro modo podríamos obtener prestaciones poco significativas.

7 Referencias

1. Organización Mundial de la Salud. Diabetes. [Online].; 2021 [cited 2021 Mayo. Available from: https://www.who.int/es/news-room/fact-sheets/detail/diabetes.

2. Fowler M,J. Microvascular and Macrovascular Complications of Diabetes. Clin Diabetes. 2014; 29(3)(116-22).

3. NH C, Shaw J, Karuranga S, Huang Y, da Rocha Fernandes J, Ohlrogge A, et al. IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045. Diabetes Res Clin Pract. 2018 Apr; 138: p. 271-281.

4. Valle JMV. Diseño e implementación de nuevas herramientas para el análisis y prevalencia de Diabetes Mellitus. 2018..

5. Bommer C, Sagalova V, Heesemann E, Manne-Goehler J, Atun R, Bärnighausen T, et al. Global Economic Burden of Diabetes in Adults: Projections From 2015 to 2030. Diabetes Care. 2018 May; 41(5): p. 963-970.

6. C B, Heesemann E, Sagalova V, Manne-Goehler J, Atun R, Bärnighausen T, et al. The global economic burden of diabetes in adults aged 20-79 years: a cost-of-illness study. Lancet Diabetes Endocrinol. 2017 Jun; 5(6): p. 423-430.

7. American Diabetes Association. Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes. Diabetes Care. 2018 Jan; 41(Supplement 1): p. S13-S27.

8. Diabetes.co.uk. Beta Cells. [Online].; 2019 [cited 2021 Julio. Available from: https://www.diabetes.co.uk/body/beta-cells.html.

9. Sanz-Sánchez I. BMA. Diabetes mellitus: Su implicación en la patología oral y periodontal. Av Odontoestomatol. 2009 Oct ; 25(5): p. 249-263.

10. Mayo Clinic. Type 2 diabetes. [Online].; 2021 [cited 2021 Mayo. Available from: https://www.mayoclinic.org/diseases-conditions/type-2-diabetes/symptoms-causes/syc-20351193.

11. MedlinePlus. Prueba de hemoglobina glicosilada (HbA1c). [Online].; 2020 [cited 2021 Julio. Available from: https://medlineplus.gov/spanish/a1c.html.

12. Committee TIE. International Expert Committee Report on the Role of the A1C Assay in the Diagnosis of Diabetes. Diabetes Care. 2009 Jul; 32(7): p. 1327–1334.

13. N B. Prediabetes diagnosis and treatment: A review. World J Diabetes. 2015 Mar; 6(2): p. 296-303.

14. Tabák A, Herder C, Rathmann W, Brunner E, Kivimäki M. Prediabetes: a high-risk state for diabetes development. Lancet. 2012 Jun; 379(9833): p. 2279-90.

15. Mani S, Chen Y, Elasy T, Clayton W, Denny J. Type 2 diabetes risk forecasting from EMR data using machine learning. AMIA Annu Symp Proc. 2012; 2012: p. 606-15.

16. Faruque MF, Asaduzzaman , Sarker IH. Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus. 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). 2019;: p. 1-4.

17. Kandhasamy, J. Pradeep ; Balamurali, S. Performance Analysis of Classifier Models to Predict Diabetes Mellitus. Procedia Computer Science. 2015; 47: p. 45-51.

18. Géron A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. 2nd ed.: O'Reilly Media; 2019.

19. Trevor H, Tibshirani R, Friedman J. The Elements of Statistical Learning. 2nd ed.: Springer; 2009.

20. Kürşad Uçar M, Majid N, Hatem S, Kemal P. The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets. Mathematical Problems in Engineering. 2020 May;(1-17).

21. Khalaf Jabbar H, Zaman Khan R. Methos to avoid over-fitting and under-fitting in supervised machine learning (comparative study). 2014 Dec.

22. OlexSys. Data/Parameter Ratio. [Online]. [cited 2021 Junio. Available from: https://www.olexsys.org/olex2/docs/reference/diagnostics/data-parameter-ratio/.

23. W R, Raghupathi V. Big data analytics in healthcare: promise and potential. Health Inf Sci Syst. 2014 Feb; 2:3.

24. Towards data science. Understanding Random Forest. [Online].; 2019 [cited 2021 Junio. Available from: https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

25. T.G. D. Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science. ; 1857.

26. Gonzalo Á. Machine learning para todos. [Online].; 2020 [cited 2021 Julio. Available from: https://machinelearningparatodos.com/cual-es-la-diferencia-entre-los-metodos-de-bagging-y-los-de-boosting/.

27. aprendeIA. Introducción a Bias y Varianza. [Online].; 2018 [cited 2021 Junio. Available from: https://aprendeia.com/bias-y-varianza-en-machine-learning/.

28. Machine Learning para todos. Algoritmos de boosting. [Online].; 2020 [cited 2021 Junio. Available from: https://machinelearningparatodos.com/cual-es-la-diferencia-entre-los-metodos-de-bagging-y-los-de-boosting/#:~:text=En%20los%20algoritmos%20de%20boosting,detr%C3%A1s%20de

%20otro%20modelo%20simple.&text=La%20diferencia%20con%20el%20bagging,los%20errores%20d.

29. Rodrigo JA. Random Forest con Python. [Online].; 2020 [cited 2021 Junio. Available from: https://www.cienciadedatos.net/documentos/py08_random_forest_python.html.

30. Alvear JO. Árboles de decisión y Random Forest. [Online].; 2018 [cited 2021 Junio. Available from: https://bookdown.org/content/2031/.

31. Maimon O, Rokach L. Data Mining and Knowledge Discovery Handbook. 2nd ed.: Springer; 2010.

32. Towards data science. Decision Trees Explained. [Online].; 2020 [cited 2021 Junio. Available from: https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6.

33. Maimon O, Rokach. Top-down induction of decision trees classifiers - a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2005 Nov; 35(4): p. 476-487.

34. James G, Witten D, Hastie T, Tibshirani T. An Introduction to Statistical Learning: Springer; 2013.

35. Zhang C, Ma Y. Ensemble Machine Learning (Methods and Applications). 1st ed.: Springer; 2012.

36. Espinosa Zuñiga JJ. Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito. Ing. invest. y tecnol. 2020; 21(3).

37. Koehrsen W. Random Forest Simple Explanation. [Online].; 2021 [cited 2020 Julio. Available from: https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d.

38. (Google) CIdaa. Clasificación: ROC y AUC. [Online].; 2020 [cited 2021 Junio. Available from: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=es-419.

39. García Carretero R, Virgil Medina L, Mora-Jiménez I, Soguero Ruiz C, Barquero Pérez O, Ramos López J. Use of a K-nearest neighbors model to predict the developmentof type 2 diabetes within 2 years in an obese, hypertensivepopulation. Medical & Biological Engineering & Computing. 2020 May; 58(5): p. 991-1002.

40. Haixiang G, Yiging L, Shang J, Mingyun G, Yuanye H, Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. 2017 May; 73: p. 220-239.

41. Bach M, Werner A, Żywiec J, Pluskiewicz W. The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis. Information Sciences. 2016 Sep; 384.

42. Walimbe R. Data Science Central: Handling imbalanced dataset in supervised learning using family of SMOTE algorithm. [Online].; 2017 [cited 2021 Junio. Available from: https://www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family.

43. ICHI.PRO. 5 técnicas SMOTE para sobremuestrear sus datos de desequilibrio. [Online].; 2020 [cited 2021 Junio. Available from: https://ichi.pro/es/5-tecnicas-smote-para-sobremuestrear-sus-datos-de-desequilibrio-202401874961077.

44. Herrera A, Fernández , García M, Ronaldo C P, Krawczyk B, Herrera F. Learning from Imbalanced Data Sets. Primera ed.: Springer; 2018.

45. Arlot S, Celisse. A survey of cross-validation procedures for model selection. Statistics Surveys. 2009; 4: p. 40-79.

46. Amat Rodrigo J. Calibrar modelos de machine learning. [Online].; 2020 [cited 2021 Junio. Available from: https://www.cienciadedatos.net/documentos/py11-calibrar-modelos-machine-learning.html.

47. He H, García EA. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering. 2009 Sep; 21(9): p. 1263-1284.

48. Japkowicz N, Stephen S. The Class Imbalance Problem: A Systematic Study. Intelligent Data Analysis. 2001 Jan; 6(5): p. 429 – 449.

49. Wallace BC, Dahabreh IJ. Improving class probability estimates for imbalanced data. Knowl Inf Syst. 2014; 41: p. 33-52.

50. Education IC. Machine Learning. [Online].; 2020 [cited 2021. Available from: https://www.ibm.com/cloud/learn/machine-learning.

51. Zelada C. RPubs. [Online].; 2017 [cited 2021 Junio. Available from: https://rpubs.com/chzelada/275494.

52. Luo W, Phung D, Tran T, Gupta S, Rana S, Karmakar C, et al. Guidelines for Developing and Reporting Machine Learning Predictive Models in Biomedical Research: A Multidisciplinary View. J Med Internet Res. 2016 Dec; 18(12): p. 323.

53. ICHI.PRO. 5 técnicas SMOTE para sobremuestrear sus datos de desequilibrio. [Online].; 2020 [cited 2021 Junio. Available from: https://ichi.pro/es/5-tecnicas-smote-para-sobremuestrear-sus-datos-de-desequilibrio-202401874961077.

Apéndice A

Descripción de la base de datos

Variable	Unidad	Descripción de la variable
Edad	años	Edad
Peso	kg	Peso
Talla	cm	Altura
IMC	kg/m^2	Índice de masa corporal
Creatinina	mg/dl	Niveles sanguíneos de creatinina
Cistatina	mg/dl	Niveles sanguíneos de cistatina
HDL	mg/dl	Niveles sanguíneos de colesterol HDL
LDL	mg/dl	Niveles sanguíneos de colesterol LDL
Triglicéridos	mg/dl	Niveles sanguíneos de transaminasa glutámico-pirúvica
GOT	mg/dl	Niveles sanguíneos de transaminasa glutámico oxalacética
GPT	mg/dl	Niveles sanguíneos de Transaminasa glutámico-pirúvica
GGT	mg/dl	Niveles sanguíneos de enzima gamma glutamil transferasa
Albuminuria	mg/g	Relación albúmina/creatinina
Ferritina	mg/dl	Niveles sanguíneos de ferritina
HOMA		Índice de grado de resistencia a la insulina
Insulina	microUI/mL	Niveles sanguíneos de insulina
Glucemia	mg/dl	Niveles de glucosa en sangre
Hb-glicosilada	mg/dl	Nivel promedio de glucosa o azúcar en la sangre durante los últimos tres meses
PCR	mg/dl	Niveles sanguíneos de proteína c reactiva

Vitamina-d	ng/l	Niveles sanguíneos de vitamina-d
TAS	mmHg	Tensión sistólica
TAD	mmHg	Tensión diástolica
Fecha	año-mes-día	Fecha de la revisión

Tabla A. 1 Descripción de las variables contenidas en la base de datos.

Apéndice B

Visualización de outliers

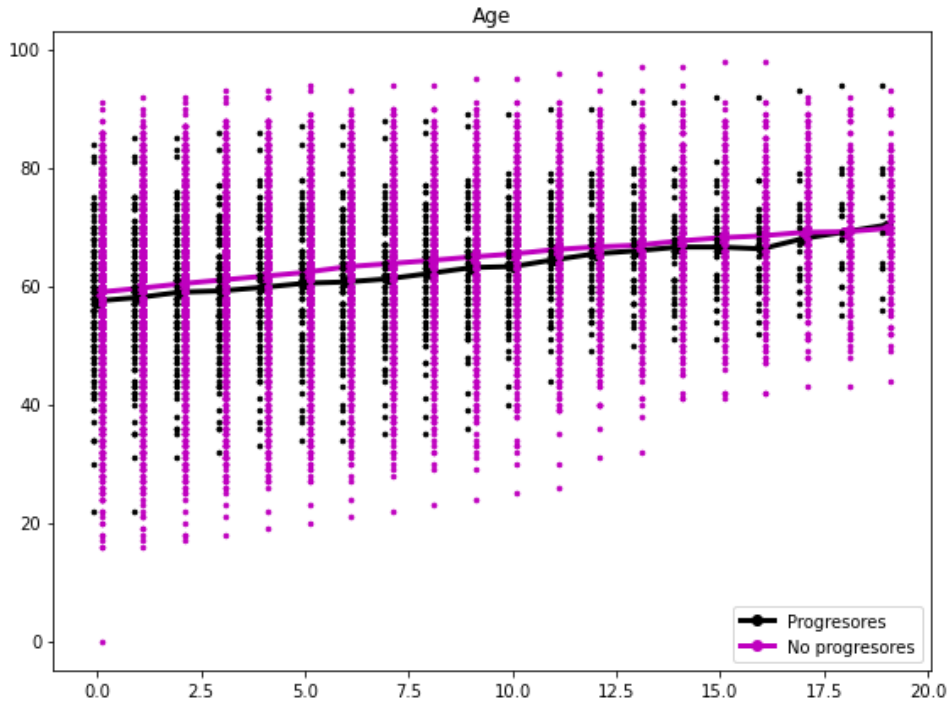

Figura B. 1 Visualización de outliers para la variable age

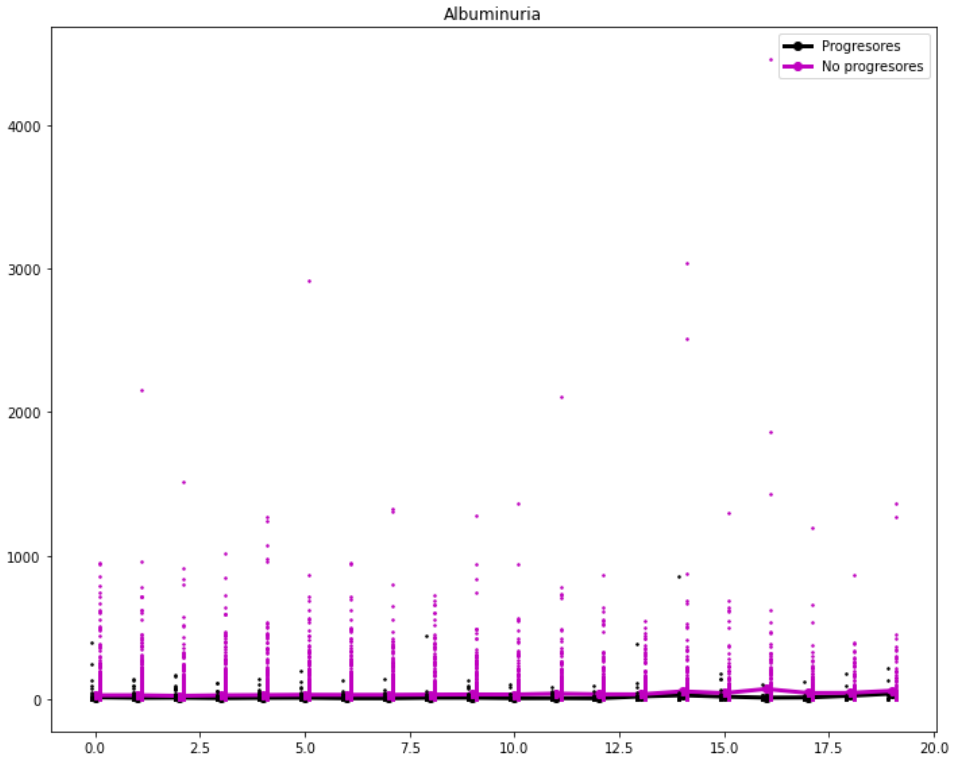

Figura B. 2 Visualización de outliers para la variable albuminuria.

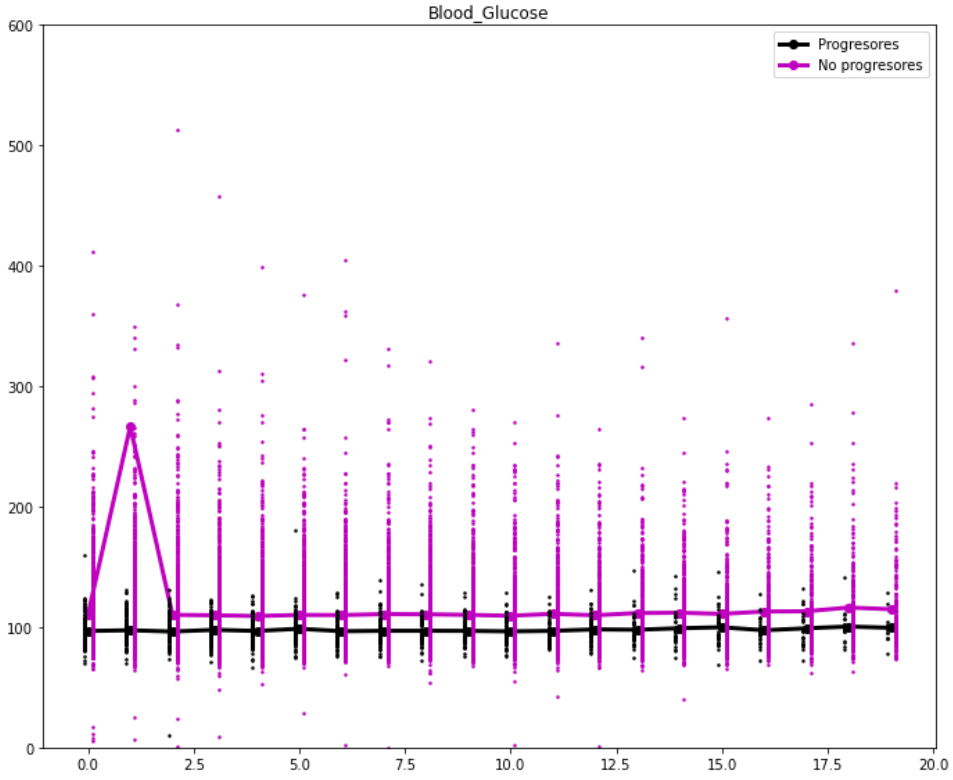

Figura B. 3 Visualización de outliers para la variable blood Glucose.

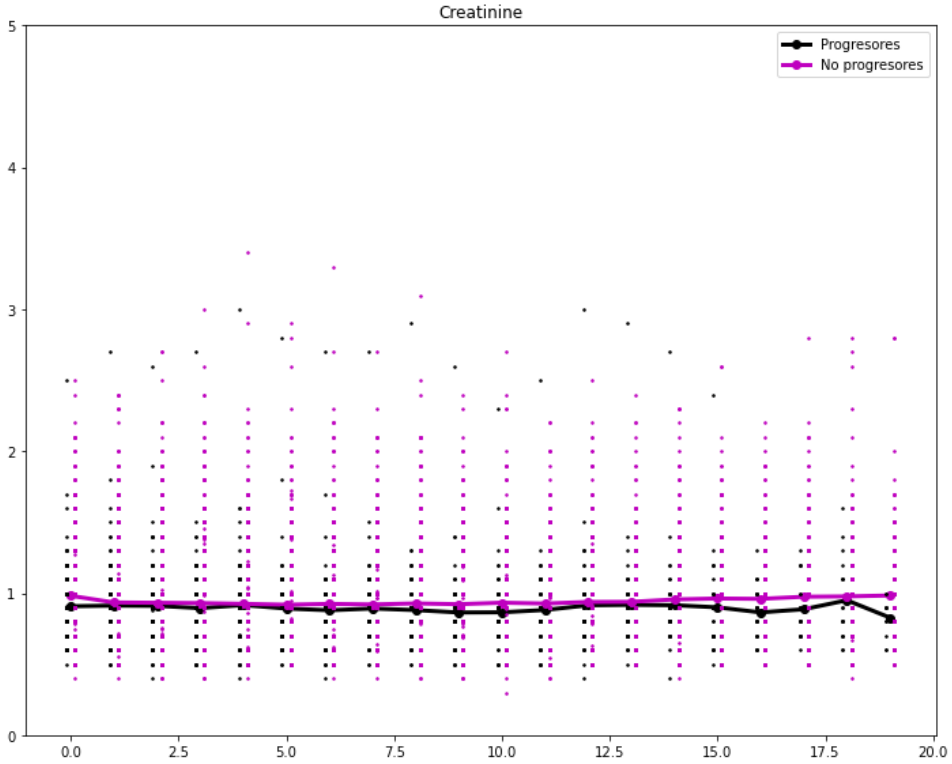

Figura B. 4 Visualización de outliers para la variable creatinine.

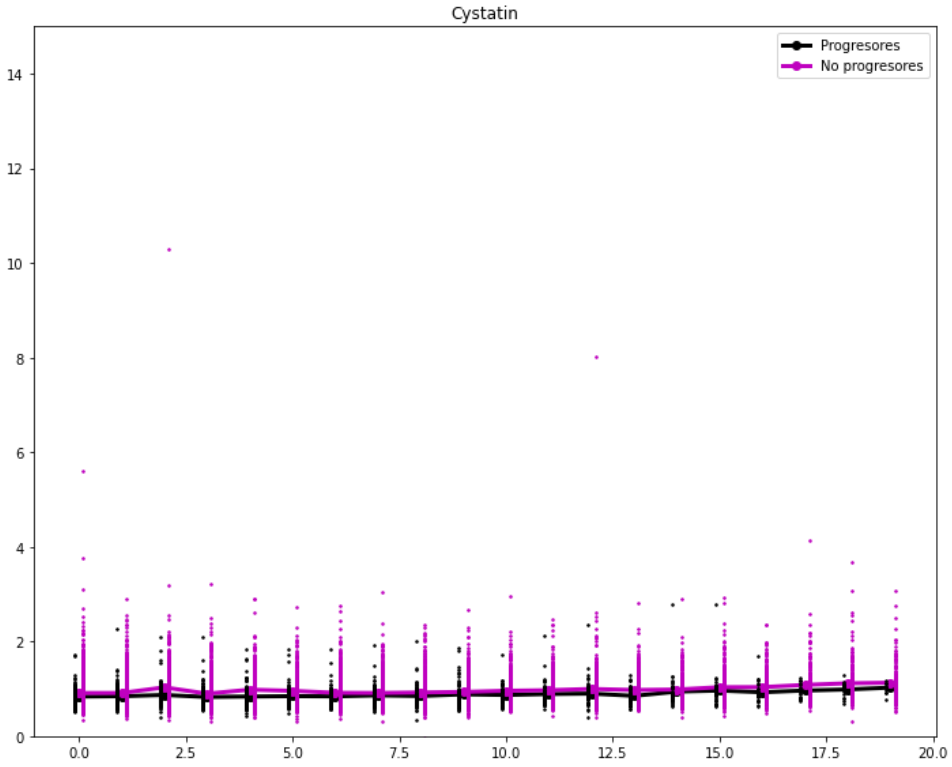

Figura B. 5 Visualización de outliers para la variable cystatin.

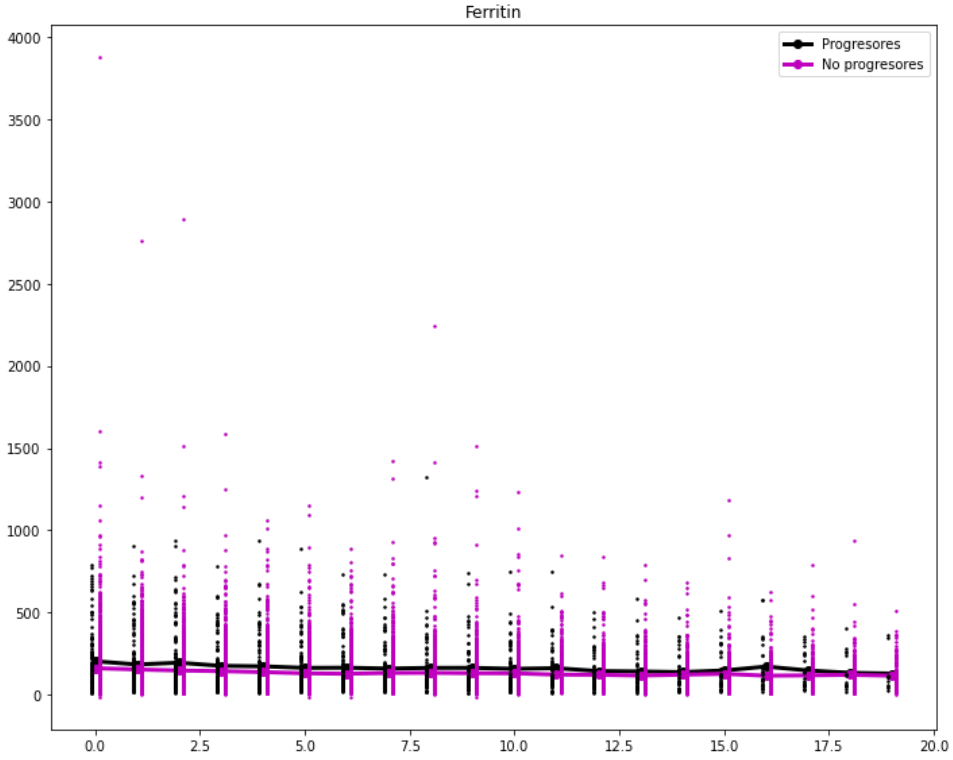

Figura B. 6 Visualización de outliers para la variable ferritin.

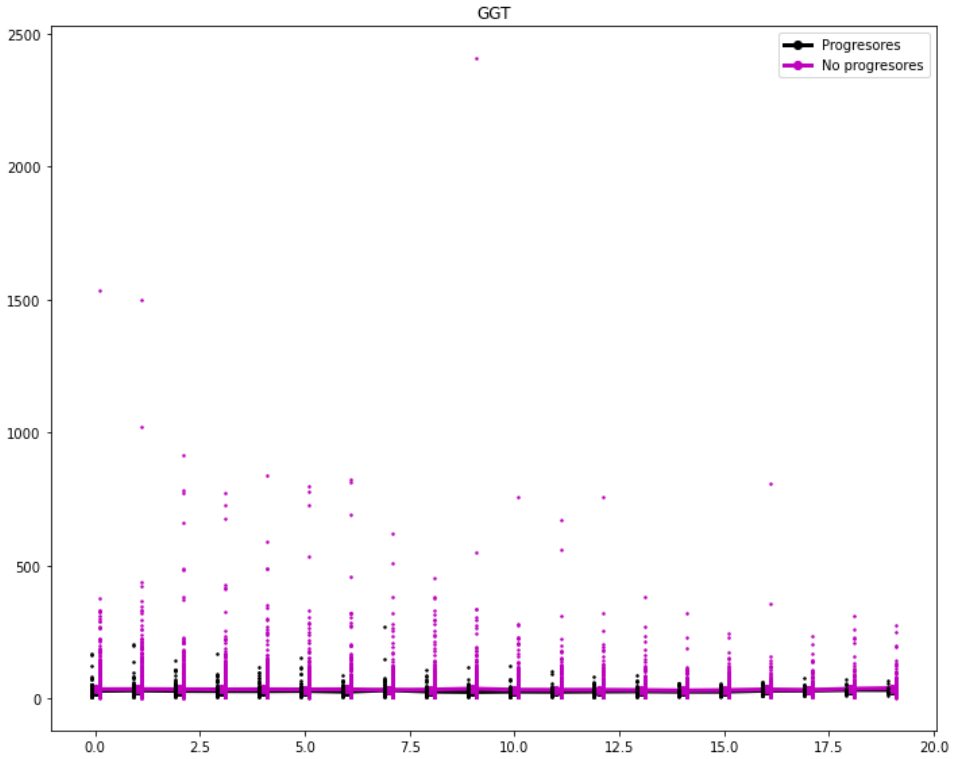

Figura B. 7 Visualización de outliers para la variable GGT.

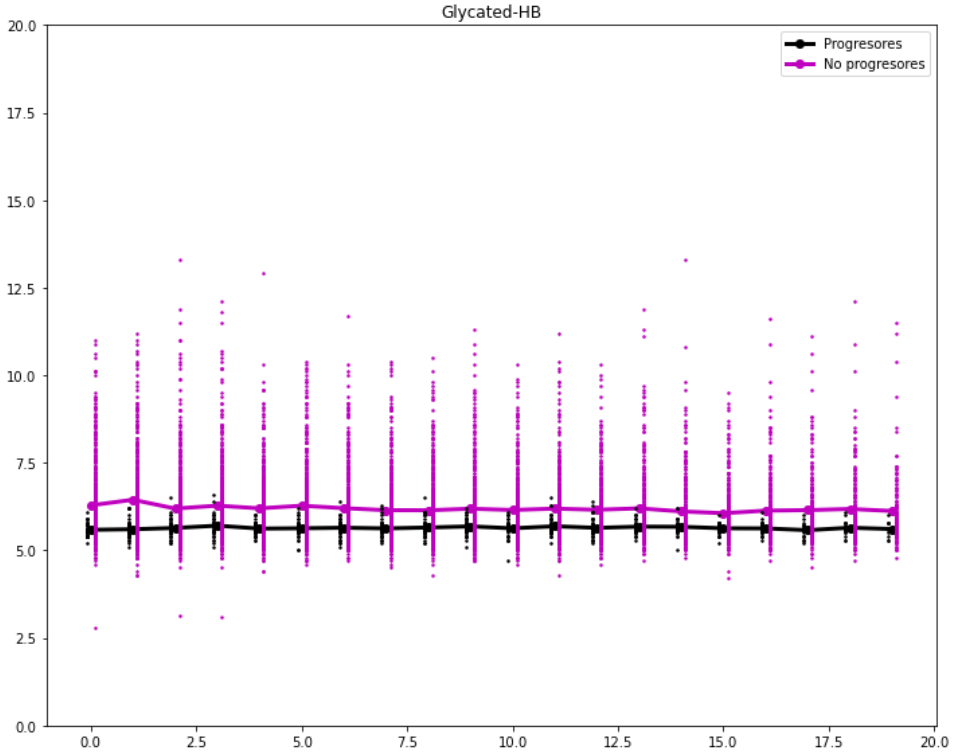

Figura B. 8 Visualización de outliers para la variable glycated HB.

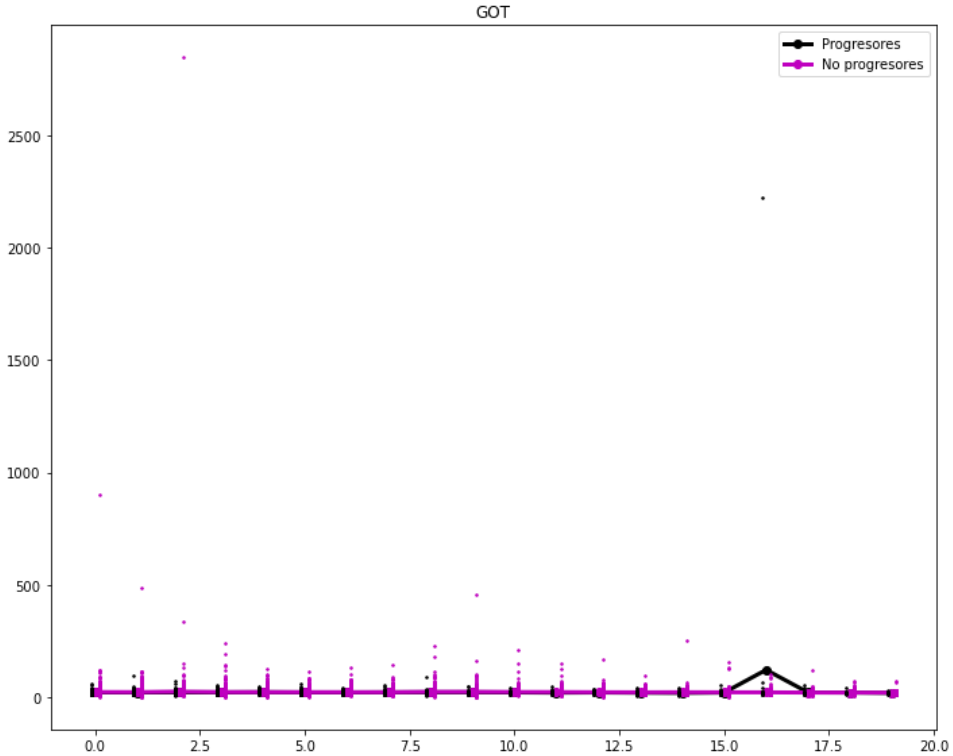

Figura B. 9 Visualización de outliers para la variable GOT.

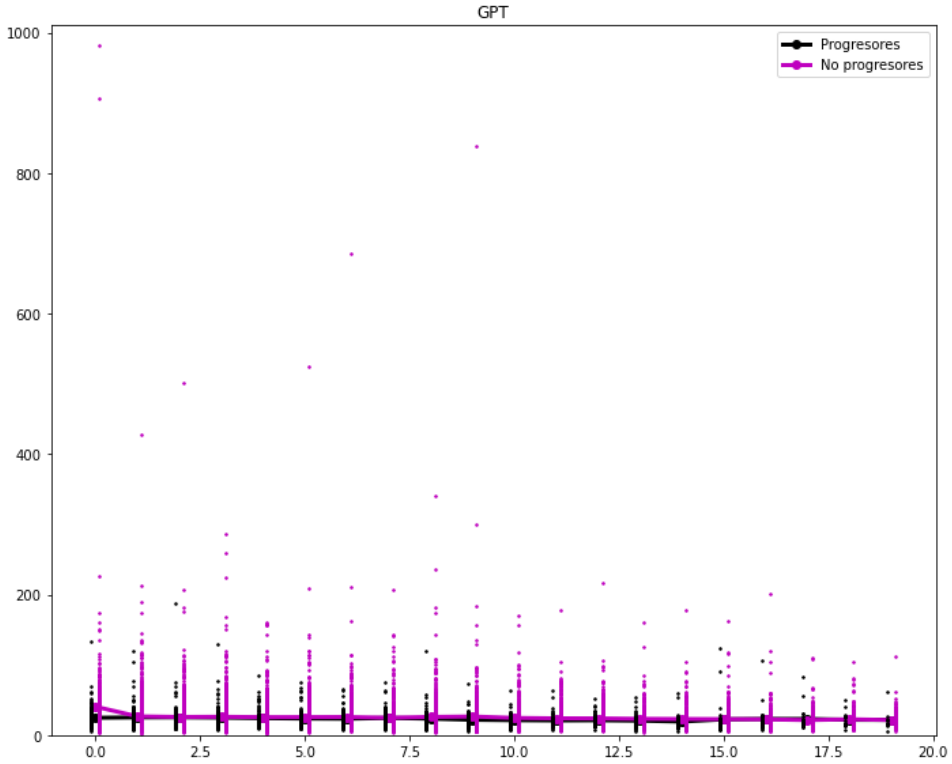

Figura B. 10 Visualización de outliers para la variable GPT.

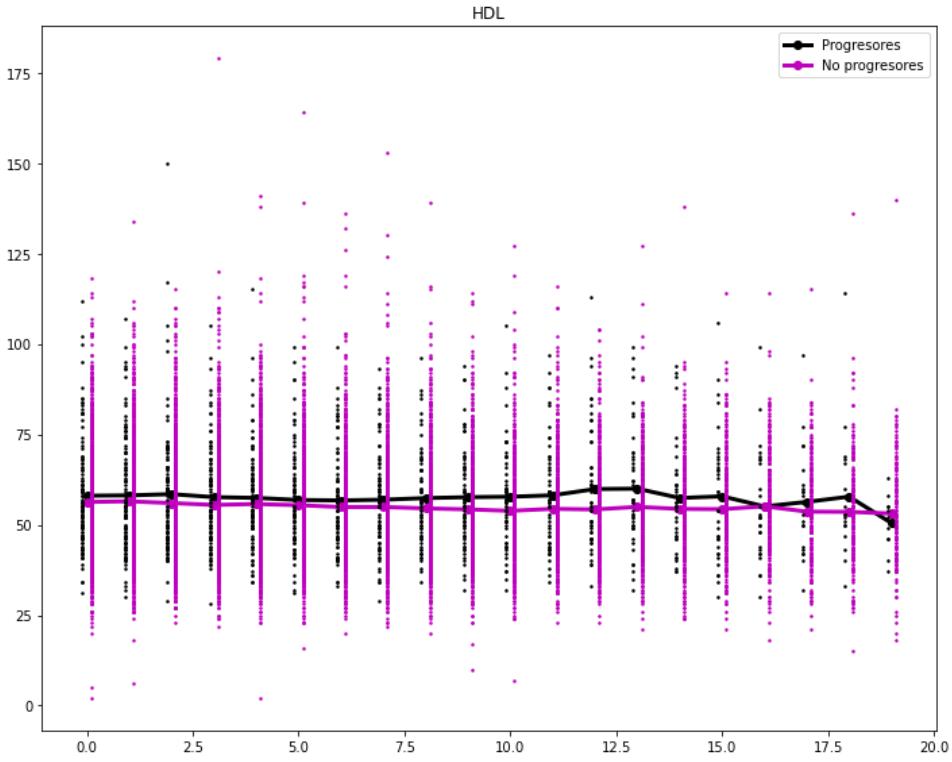

Figura B. 11 Visualización de outliers para la variable HDL.

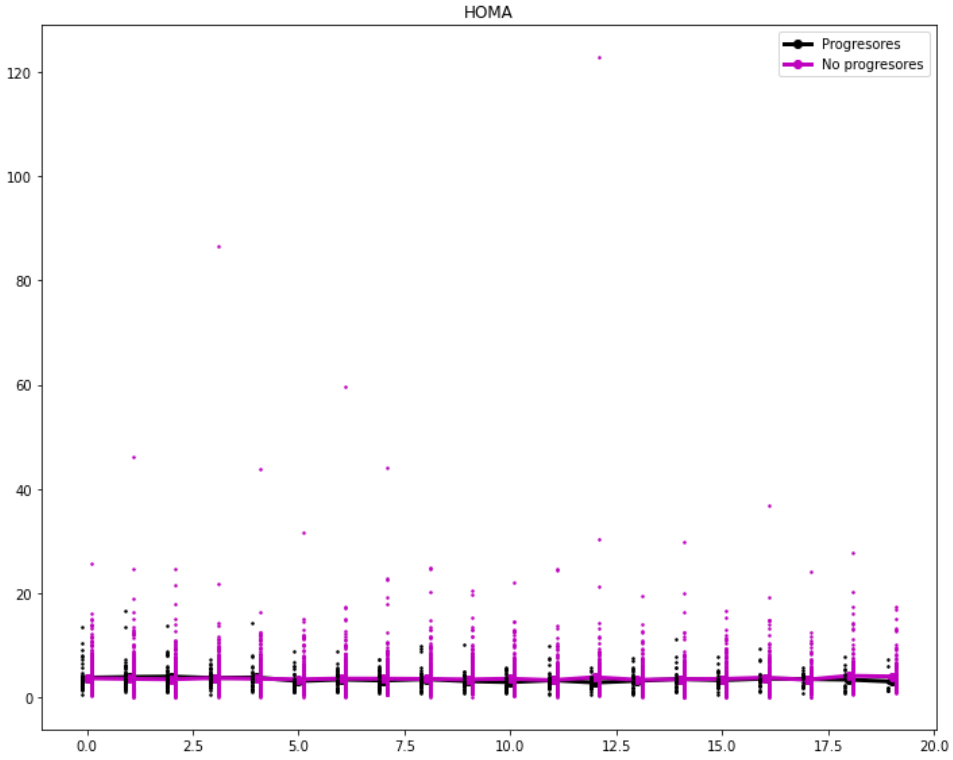

Figura B. 12 Visualización de outliers para la variable HOMA.

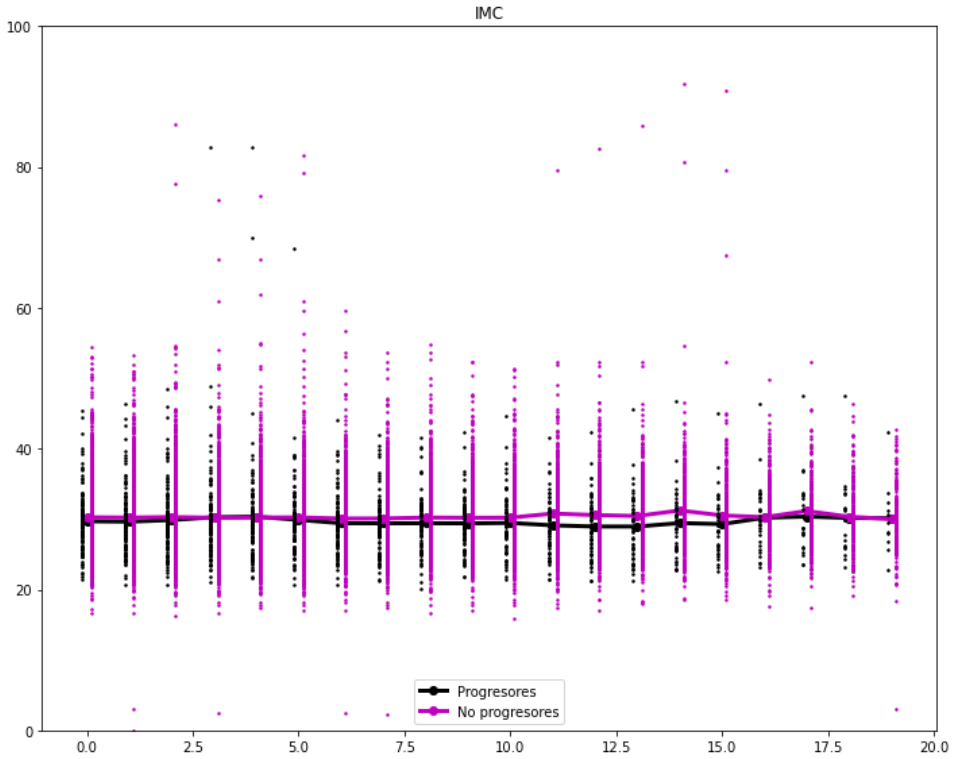

Figura B. 13 Visualización de outliers para la variable IMC.

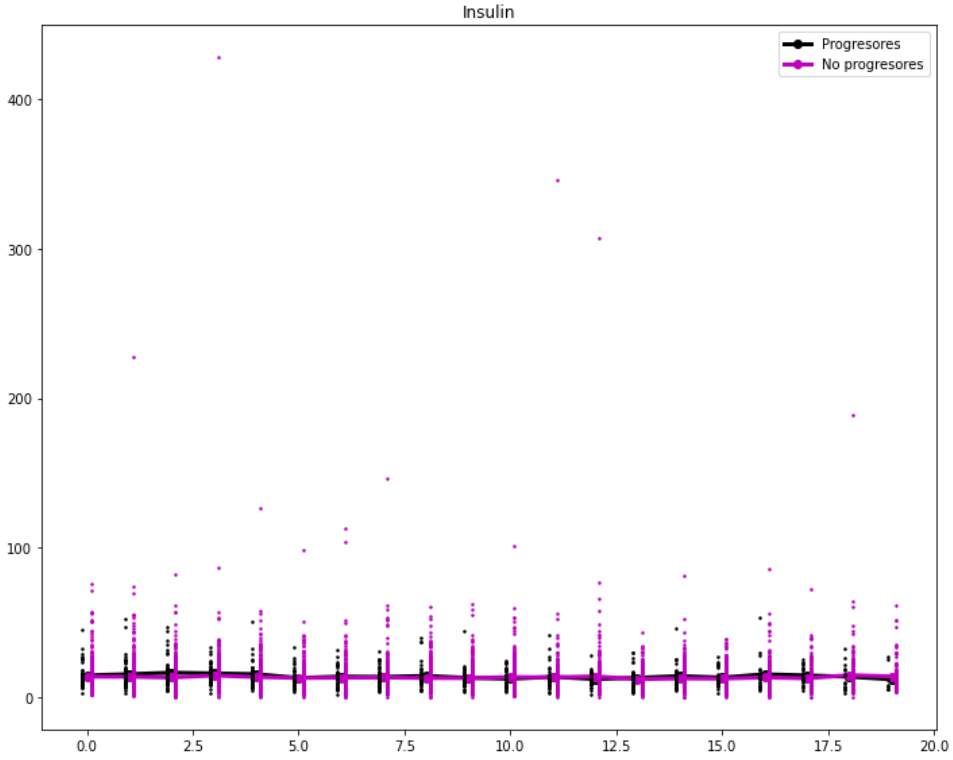

Figura B. 14 Visualización de outliers para la variable insulin.

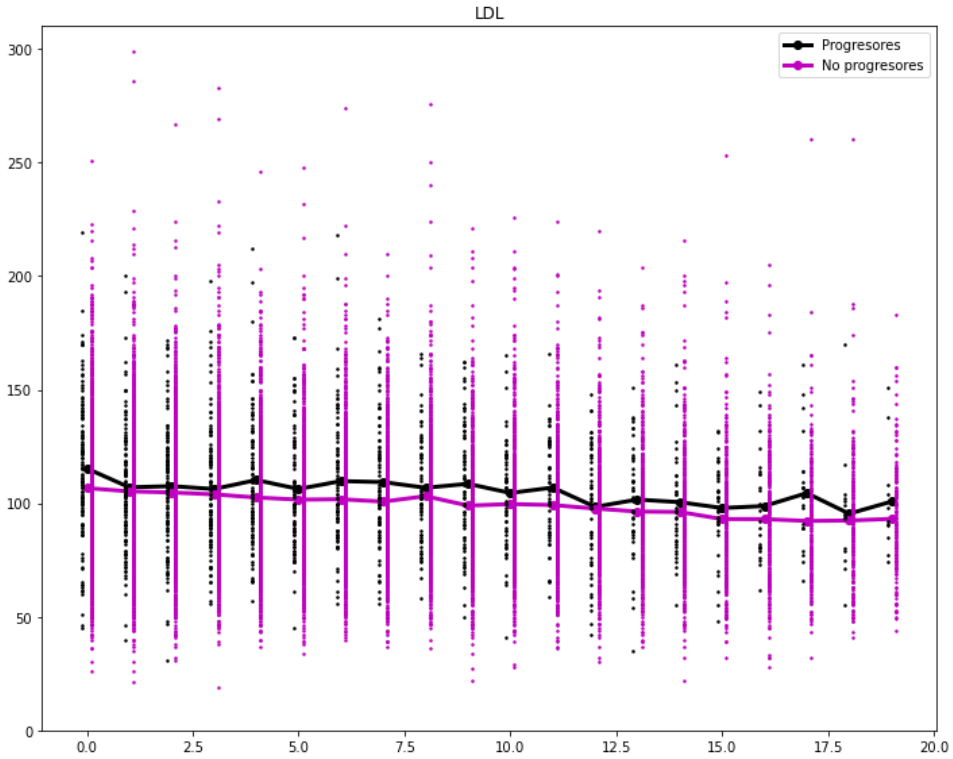

Figura B. 15 Visualización de outliers para la variable LDL.

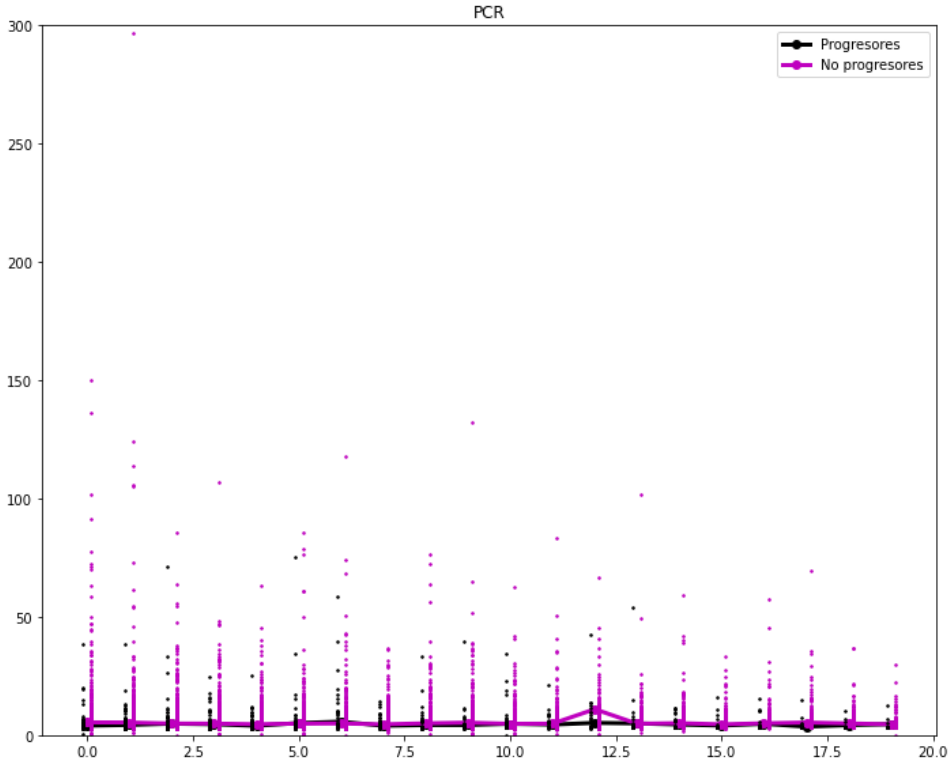

Figura B. 16 Visualización de outliers para la variable PCR.

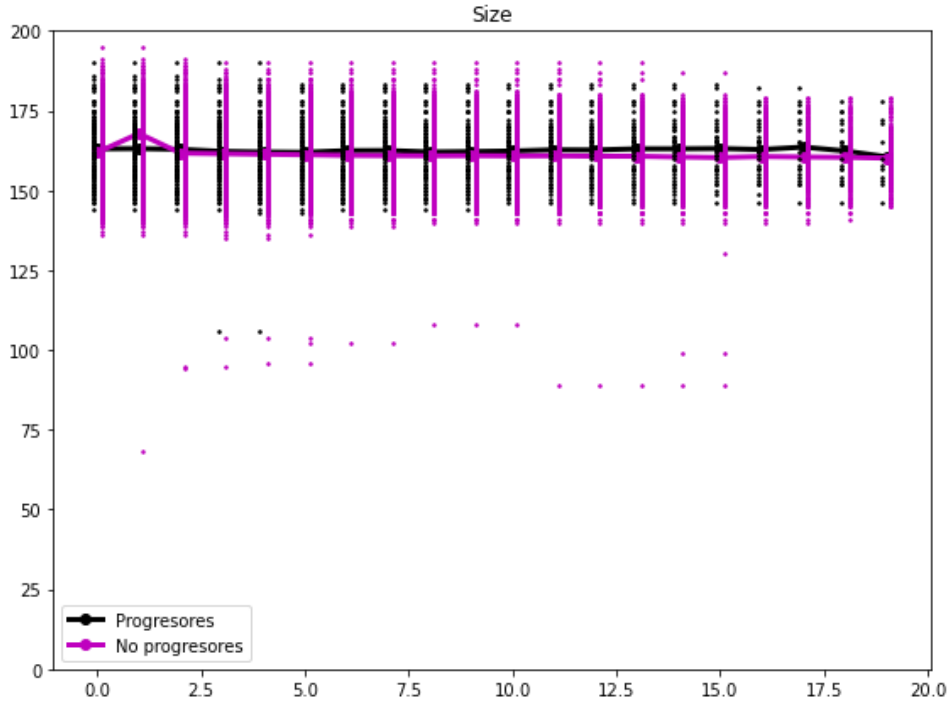

Figura B. 17 Visualización de outliers para la variable size.

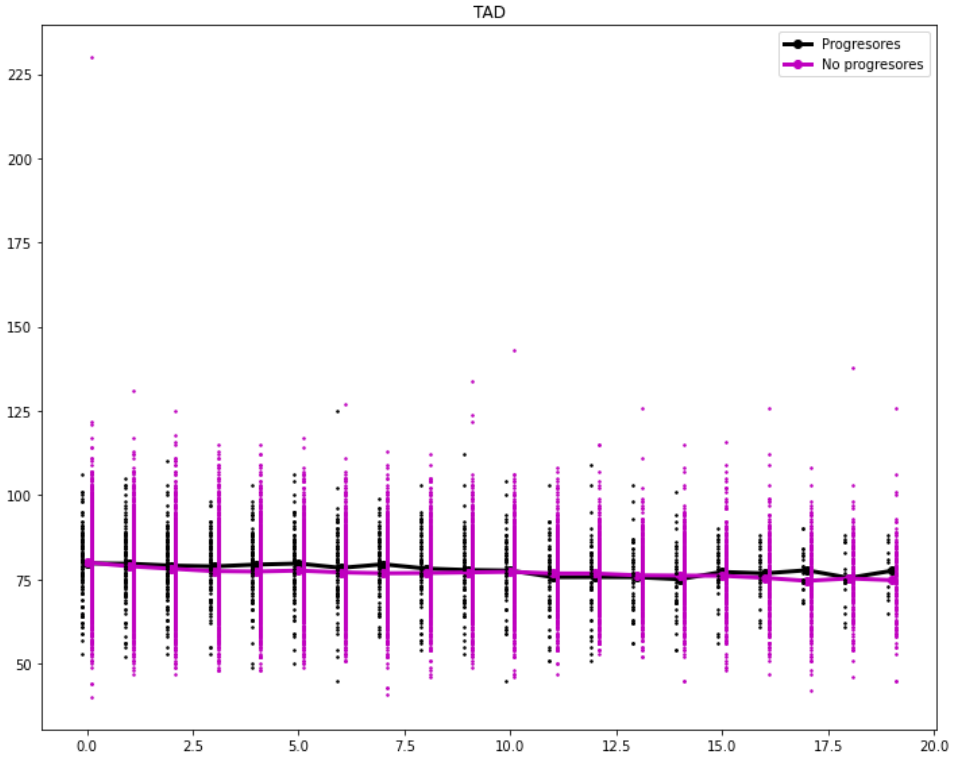

Figura B. 18 Visualización de outliers para la variable TAD.

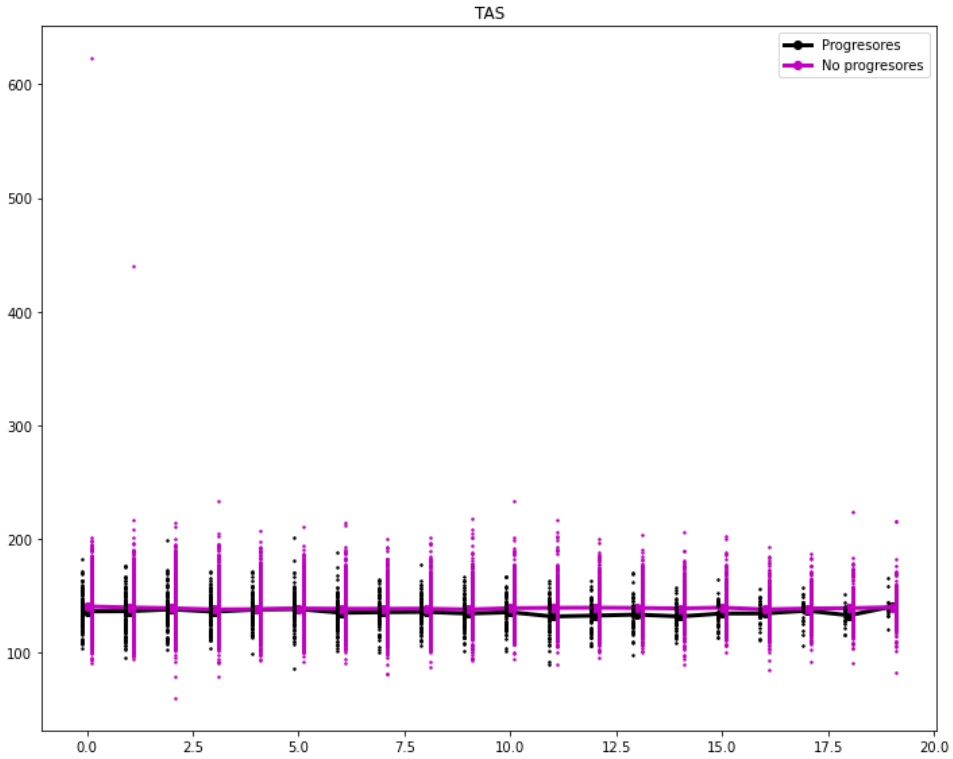

Figura B. 19 Visualización de outliers para la variable TAS.

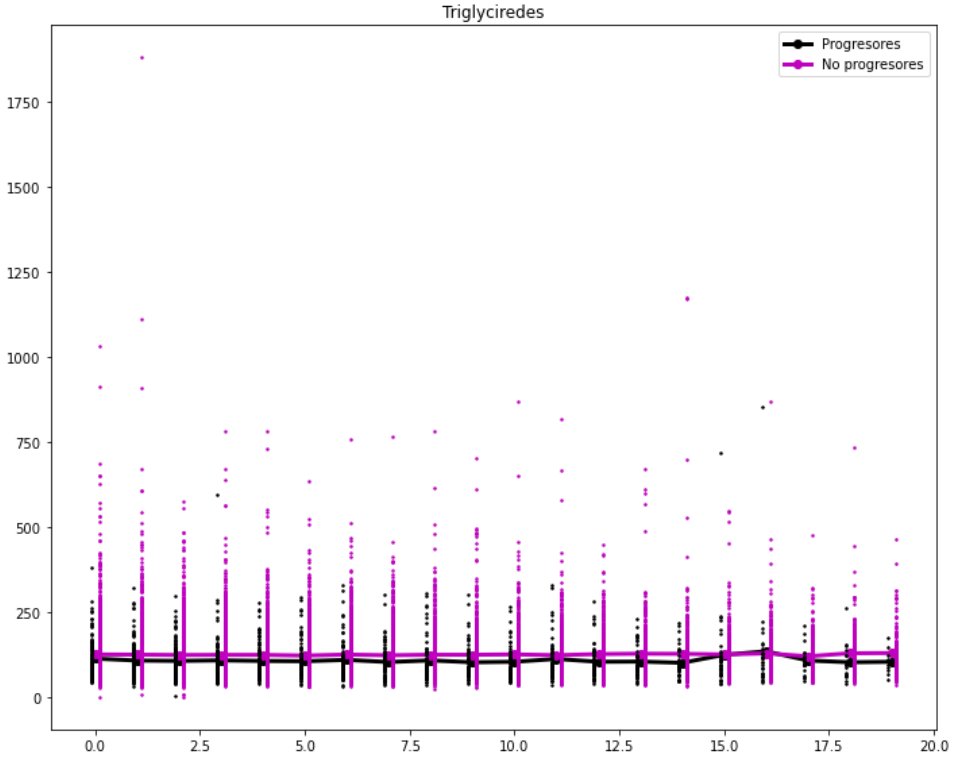

Figura B. 20 Visualización de outliers para la variable triglyciredes.

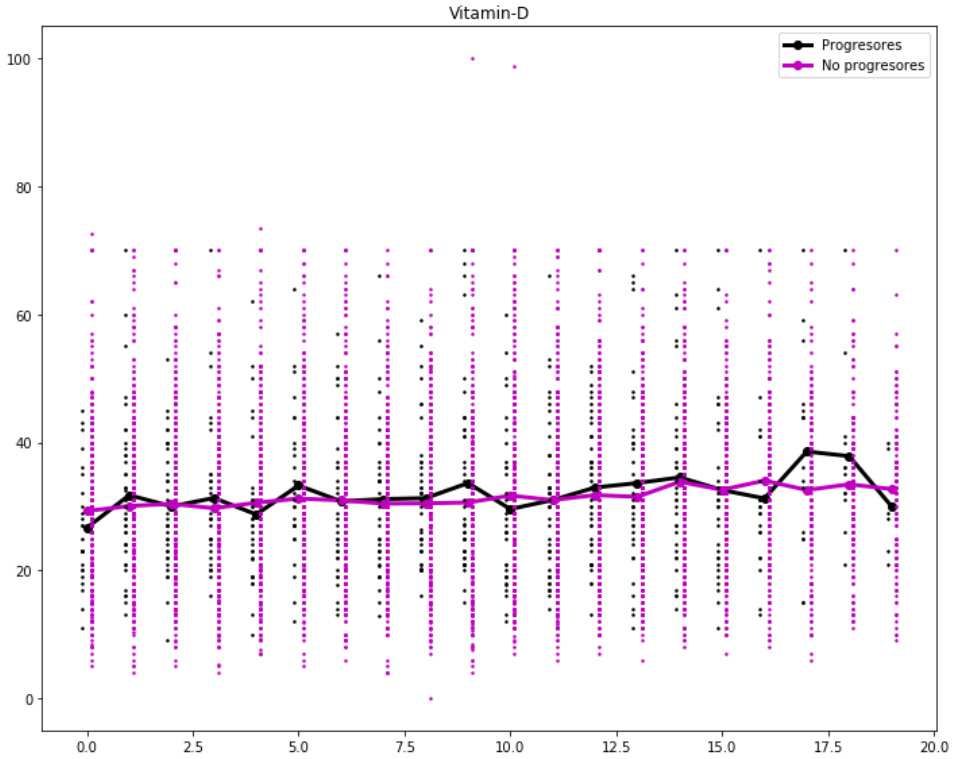

Figura B. 21 Visualización de outliers para la variable vitamin-d.

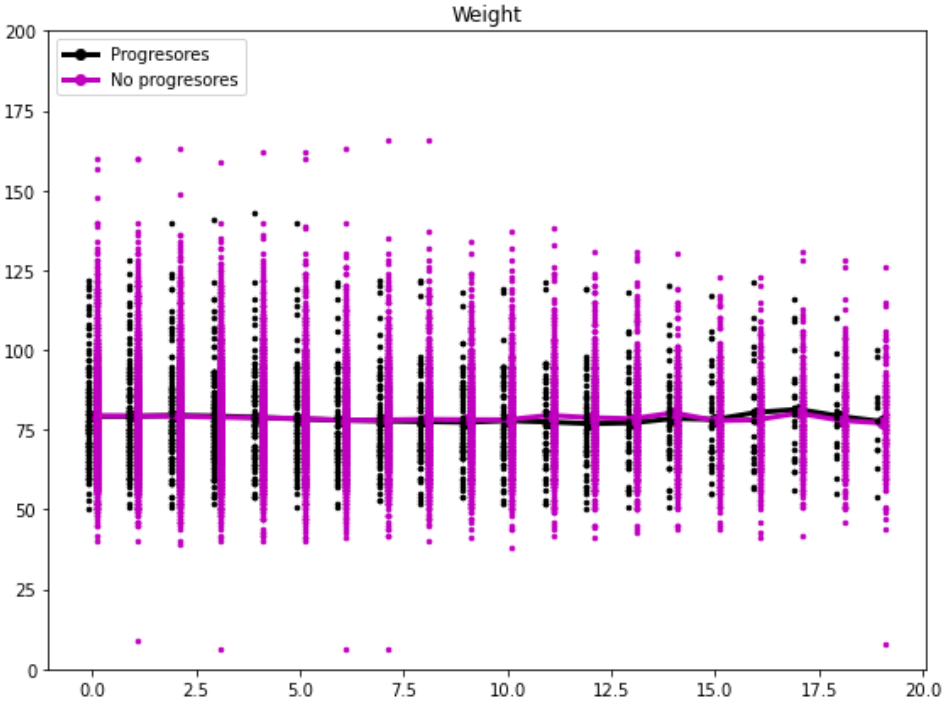

Figura B. 22 Visualización de outliers para la variable weight.