Evaluación de Implementación de Modelos de Machine Learning para predecir diabetes

Gabriela Chimali Nava Ramírez | A01710530

15/09/2025

ABSTRACT *La detección temprana de la diabetes es un reto crítico de salud, ya que su diagnóstico tardío incrementa significativamente las complicaciones y los costos asociados al tratamiento. En este trabajo se implementaron y evaluaron modelos de Machine Learning, específicamente Regresión Logística y Random Forest, para predecir la presencia de diabetes a partir de indicadores clínicos y demográficos. El análisis incluyó la preparación de datos, ajuste de hiperparámetros y comparación de métricas como Accuracy, Recall, F1 Score y AUC-ROC. Los resultados muestran que, aunque los modelos alcanzan una alta exactitud global, el mayor desafío radica en equilibrar sensibilidad y precisión frente a un dataset desbalanceado, Se concluye que priorizar el recall resulta clínicamente más apropiado para evitar falsos negativos, a costa de incrementar falsos positivos, y que técnicas adicionales de balanceo de clases y optimización de umbrales pueden potenciar la utilidad del modelo como herramienta de apoyo diagnóstico.*

1. INTRODUCCIÓN

La diabetes es una enfermedad crónica que afecta a millones de personas en el mundo y cuya detección temprana es esencial para reducir complicaciones y mejorar la calidad de vida. El uso de algoritmos de Machine Learning en la medicina ha cobrado relevancia ha permitido grandes volúmenes de información para identificar patrones que no siempre son fáciles de detectar.

En este reporte se plantea la aplicación de modelos de clasificación binaria para predecir la condición diabética a partir de indicadores de salud y demográficos. Cabe recalcar que el análisis está orientado no sólo a la comparación entre algoritmos, sino también a la reflexión sobre su impacto y ajuste práctico para la toma de decisiones médicas.

2. DESCRIPCIÓN DEL DATASET

El análisis se llevó a cabo utilizando el conjunto de datos “diabetes_prediction_dataset.csv” [1], que integra información médica y demográfica de pacientes. Es popularmente usado para la investigación de las relaciones entre estos factores y la probabilidad de desarrollar diabetes.

2a.- Features

El conjunto de datos incluye la variable de clasificación u dependiente y 8 variables predictoras o independientes que abarcan indicadores de salud y características sociodemográficas.

Variable dependiente:

diabetes	0 = No diabético 1 = Diabético

Variables numéricas:

bmi	Índice de Masa Corporal. Medida de la grasa corporal basada en el peso y altura. Valores altos suelen estar presentes en individuos con diabetes.	10 . 95 18.5 (bajo peso) 18.5-24.9 (normal) 25-29.9 (sobrepeso) >=30 (obesidad)
age	Edad, es un factor importante pues las personas mayores es más común que sean diagnosticadas con diabetes.	0.08 – 80 años
HbA1c_level	Nivel de azúcar promedio en la sangre en los 2 a 3 meses pasados. Un nivel alto (aproximadamente mayor a 6.5 implica un alto riesgo de desarrollar diabetes.	3.5 - 9
blood_glucose_level	Cantidad de glucosa en el torrente sanguíneo. Un nivel alto es un indicador clave de diabetes.	80 - 300

Variables categóricas binarias:

hypertension	Hipertensión es una afección médica en la que la presión arterial está persistentemente elevada.	0 = Sin hipertensión 1 = Con hipertensión
heart_disease	Enfermedad cardíaca, es otra condición médica que está asociada con un mayor riesgo de desarrollar diabetes.	0 = Sin una enfermedad cardiaca 1 = Tienen una enfermedad cardiaca

Variables categóricas nominales:

gender	Género, se refiere al sexo biológico del	Female = Femenino

	individuo, que puede tener un impacto en su susceptibilidad a la diabetes.	Male = Masculino Other = Otro género
smoking_history	El historial de tabaquismo también se considera un factor de riesgo para la diabetes y puede agravar las complicaciones asociadas con la diabetes.	Not current = No actualmente Ever = Siempre Never = Nunca Former = Anteriormente Current = Fumador activo

3. EXTRACCIÓN DE DATOS

3a.- Carga y exploración de datos

Se realizó un análisis exploratorio inicial con el fin de evaluar la composición del dataset y verificar la presencia de posibles problemas de calidad de datos, tales como valores nulos, duplicados, datos faltantes o categorías poco representativas. Los principales hallazgos fueron:

- El tamaño del dataset es de 100,000 instancias y no se encontraron valores nulos en ninguna columna.

```
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   gender               100000 non-null  object 
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64  
 3   heart_disease        100000 non-null  int64  
 4   smoking_history      100000 non-null  object 
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64  
 8   diabetes             100000 non-null  int64  
```

Figura 1. *Tabla con estructura del dataset obtenida con la función info() de la librería pandas*

- La distribución es 91,5% son personas **sin diabetes** y 8.5% son personas **con diabetes**.

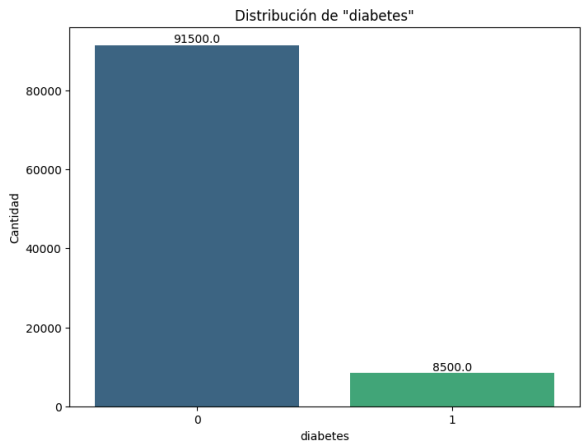

Figura 2. *Gráfica de distribución de la variable objetivo "diabetes"*

- En la variable categórica binaria "gender" se identificaron 18 instancias en la categoría "Other", lo que son únicamente el 0.00018% del total. Debido a su baja representatividad y para evitar introducir ruido, se decidió eliminarlos.

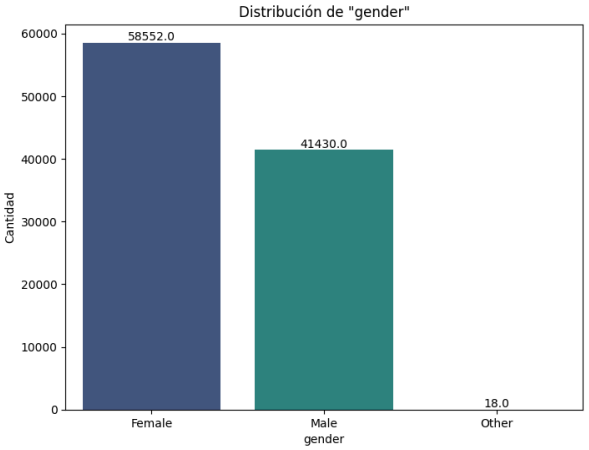

Figura 3*. Gráfica de distribución de la feature "gender"*

- En la variable categórica nominal "smoker" hay 35,816 instancias sin información. Se consideraron 3 opciones para manejar estos datos faltantes:
  - **Imputar**, que consistiría en asignar categorías basadas en una medida estadística; descartado por falta de evidencia confiable para hacerlo.
  - Mantenerlo como **otra categoría más**, sería opción si la ausencia de información representara un patrón relevante
  - **Eliminar** estas instancias para mantener el análisis sólo con seguros, que sea consistente y fiable.

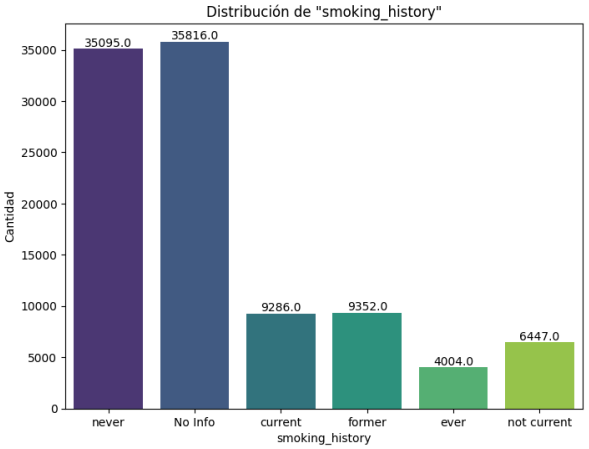

Figura 4. *Gráfica de distribución de la feature "smoking_history"*

Se optó por la eliminación ya que no mostraba relación con la variable objetivo si se mantuviera como categoría. Además debido al tamaño del dataset se puede prescindir de estos sin comprometer la cantidad adecuada de registros para entrenar y probar el modelo.

3b.- Evaluación de correlación

Para identificar las variables con mayor poder predictivo para la detección de diabetes, se realizó un análisis de correlación entre cada variable independiente y la variable dependiente. Este análisis ayuda a comprender la relevancia de cada característica y su potencial contribución al modelo.

blood_glucose_level	0.449698
HbA1c_level	0.438897
age	0.260850
bmi	0.204421
hypertension	0.192226
heart_disease	0.169614
smoking_history_former	0.079555
gender_Male	0.056997
smoking_history_never	0.050770
smoking_history_ever	0.006693
smoking_history_not current	0.002819

Figura 5. *Tabla de correlaciones con la variable dependiente (diabetes)*

Como se observa en la figura 5, las variables con mayor correlación con la condición diabética son el nivel de glucosa en sangre (0.45) y el nivel de HbA1c (0.44), lo cual es consistente con el conocimiento médico sobre los principales indicadores de esta enfermedad. En nivel de importancia siguen la edad (0.26), el índice de masa corporal (0.20), la hipertensión (0.19) y las enfermedades cardíacas (0.17). Por otro lado, las variables relacionadas con el historial de tabaquismo muestran correlaciones considerablemente más bajas, sugiriendo una influencia menor en la predicción de diabetes según los datos.

Los valores de correlación se pueden apreciar de manera más comprehensiva en el heatmap [Figura 6]. Este gráfico permite apreciar tanto la relación de las características con la condición diabética como la relación entre cada característica.

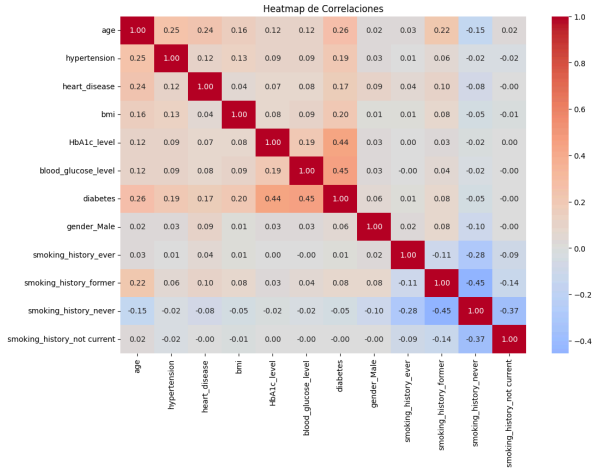

Figura 6. *Mapa de correlaciones de variables para predecir diabetes*

Este mapa revela relaciones interesantes entre variables, como:

- Una correlación moderada entre la edad y las condiciones médicas como hipertensión (0.25) y enfermedades cardíacas (0.24), lo que sugiere una asociación entre el envejecimiento y este padecimiento.
- Los niveles de glucosa en sangre y HbA1c presentan una correlación entre sí (0.19), algo consistente con el conocimiento médico, dado que ambos son indicadores relacionados con el metabolismo de la glucosa. [2]
- Las correlaciones negativas entre categorías de tabaquismo (como -0.45 entre *former* y *never*) son esperables porque son mutuamente excluyentes, es decir, pertenecer a una categoría implica no estar en otra. Esto se refleja como correlaciones negativas que no representan causalidad, sino la estructura de los datos categóricos transformados mediante one-hot encoding.

Estos hallazgos son valiosos para interpretar posteriormente los coeficientes del modelo entrenado y entender qué características contribuyen más significativamente a las predicciones.

4. TRANSFORMACIÓN DE DATOS

4a.- Estandarización de variables numéricas

Como parte del preprocesamiento, se aplicó estandarización z-score a las variables numéricas. Este procedimiento consiste en transformar cada valor restando la media de la variable y dividiendo entre su desviación estándar, lo que da como resultado una distribución con media cero y desviación estándar uno. Esta transformación no altera la relación entre los valores, pero sí permite que todas las variables estén en una escala comparable.[3]

$$X' = \frac{X - \mu}{\sigma}$$

Figura 7. *Fórmula de estandarización z-score*

La estandarización ayuda a evitar que variables con magnitudes mayores dominen el comportamiento del modelo, lo que puede afectar la interpretación de los coeficientes y la estabilidad del ajuste.

Las variables binarias se mantuvieron sin transformación, dado que ya se encuentran en una escala adecuada para el análisis.

4b.- Codificación de variables categóricas

One-Hot Encoding es otra técnica de preprocesamiento que se aplica específicamente a variables nominales (aquellas que no tienen una relación de orden entre las categorías) como el género, ya que codificarlas con números enteros podría sugerir una relación secuencial inexistente.

Consiste en crear una nueva columna para cada categoría posible de la variable, asignando un valor de 1 si la observación pertenece a esa categoría y 0 si no lo hace. [4]

Cabe mencionar que, para evitar multicolinealidad se codifican n-1 categorías. Esto ocurre porque si se incluyen todas las categorías como variables binarias, la suma de estas siempre será 1, creando una dependencia lineal entre ellas. Al eliminar una categoría y usarla como referencia, se rompe la dependencia, permitiendo que el modelo estime los efectos de las demás categorías en relación con la omitida. [5]

5. CONSTRUCCIÓN DEL MODELO

En esta sección se explican los algoritmos matemáticos implementados para realizar un modelo de regresión logística para clasificación binaria. Este modelo se basa en la transformación de una ecuación lineal mediante la función sigmoide para estimar probabilidades de pertenencia a una clase específica.

Además, en la programación de estos se aprovecha las capacidades de vectorización de la librería NumPy para optimizar algunas operaciones matemáticas. Esto buscando mejorar significativamente el rendimiento computacional de la construcción manual del modelo al procesar miles de registros simultáneamente y también hacer el código más conciso y legible.

5a.- Función de hipótesis

En la regresión logística, la función de hipótesis integra las variables independientes y sus parámetros para generar una predicción lineal. Esta predicción se transforma mediante la función sigmoide, que convierte cualquier valor real en una probabilidad entre 0 y 1. Así, la hipótesis permite estimar la probabilidad de que un individuo pertenezca a una clase específica (por ejemplo, tener o no diabetes), facilitando la clasificación binaria a partir de múltiples factores.

Matemáticamente, la función se expresa como:

$$h(x) = \sigma(w_1x_1 + w_2x_2 + \ldots + w\square x\square + b)$$

donde σ representa la función sigmoide:

$$\sigma(z) = 1 / (1 + e^{(-z)})$$

y z es la predicción lineal.

Se puede observar visualmente en la siguiente gráfica.

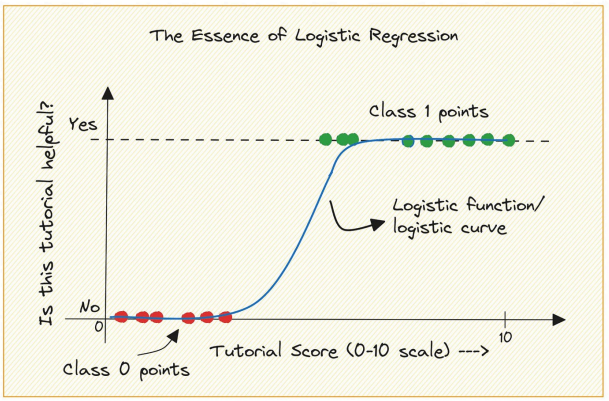

Figura 8. *La Esencia de la Regresión Logística [6]*

5b.- Función de pérdida: Cross Entropy

La función de pérdida utilizada es la entropía cruzada (cross entropy), que mide la discrepancia entre las probabilidades predichas por el modelo y los valores reales observados. Esta función penaliza fuertemente las predicciones incorrectas, propiciando que el modelo ajuste sus parámetros para minimizar el error y mejorar la precisión en la clasificación.

La fórmula para el cálculo es la siguiente:

$$J = -(1/N)\,\Sigma\,[y_i \log(a_i) + (1 - y_i)\log(1 - a_i)]$$

donde N es el número de muestras, y_i es el valor real, y a_i la probabilidad predicha para cada muestra.

$$-\begin{cases} \log a_i, & y_i = 1, \\ \log(1-a_i), & y_i = 0. \end{cases}$$

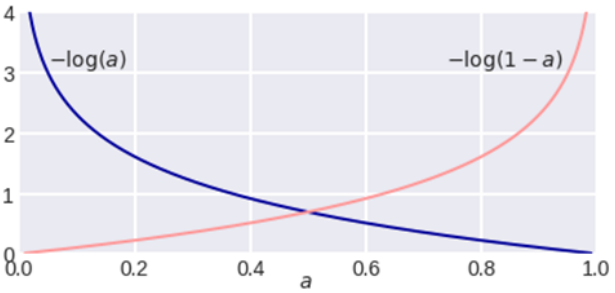

Figura 9. *Gráfica de función de pérdida con Cross Entropy [7]*

5c.- Función de optimización: Gradiente descendiente

El gradiente descendiente es el método de optimización empleado para ajustar los parámetros del modelo. Consiste en calcular la dirección en la que la función de

pérdida disminuye más rápidamente y actualizar los parámetros en esa dirección. Esta iteración permite que el modelo aprenda progresivamente, mejorando su capacidad predictiva en cada ciclo de entrenamiento.

Las fórmulas seguidas para actualiza los parámetros son:

$$w := w - \alpha \, (\partial J / \partial w)$$

$$b := b - \alpha \, (\partial J / \partial b)$$

donde α es la tasa de aprendizaje, y $\partial J/\partial w$ y $\partial J/\partial b$ son los gradientes calculados para los pesos y el sesgo, respectivamente. Hablaremos sobre estos parámetros a continuación.

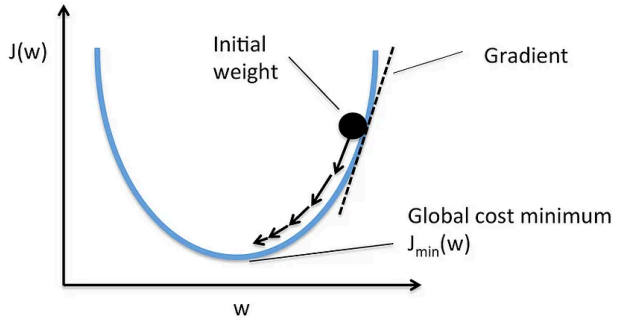

Figura 10. *Representación visual de aplicación del gradiente descendiente [8]*

5d.- Hiperparámetros para Regresión Logística

En el proceso de entrenamiento, tenemos dos hiperparámetros relevantes: el término de sesgo (bias) y la tasa de aprendizaje (learning rate). Su adecuada calibración permite lograr un desempeño óptimo que evite tanto el sobreajuste como la convergencia demasiado lenta.

El bias se recomienda inicializar en cero, permitiendo que el modelo comience sin preferencia por una clase y que se ajuste progresivamente durante el entrenamiento.

El learning rate se suele iniciar con un valor pequeño, como 0.1, para asegurar que los cambios en los parámetros sean graduales y evitar saltos bruscos que puedan dificultar la convergencia, como se observa en la Figura 11. Este valor se puede ajustar posteriormente dependiendo los resultados obtenidos, buscando un modelo tanto veloz como estable.

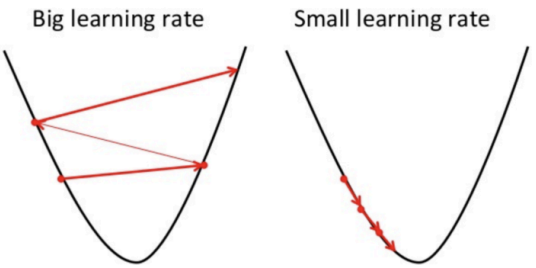

Figura 11. *Visualización de valores extremos para learning rate [9]*

6. ENTRENAMIENTO DEL MODELO

El proceso de entrenamiento busca ajustar los parámetros del modelo de regresión logística para que pueda predecir con precisión la probabilidad de que un individuo tenga diabetes. En esta implementación se utilizaron tres conjuntos de datos simultáneamente durante el entrenamiento.

6a.- División de Dataset en Conjuntos para Entrenamiento, Validación y Prueba

La división del conjunto de datos en subconjuntos de entrenamiento, validación y prueba permite evaluar el desempeño del modelo sobre datos no vistos, garantizando que los resultados sean representativos y generalizables. Cada conjunto cumple un propósito específico:

- Conjunto de Entrenamiento (60%)

Se utiliza para que el modelo aprenda los patrones y relaciones entre las características y las clases. Aquí se ajustan los parámetros (pesos y bias) para minimizar el error de predicción (pérdida).

- Conjunto de Validación (20%)

Permite evaluar el desempeño del modelo con datos que no ha visto durante el entrenamiento, pero no es la evaluación final, ayuda a:

  - Detectar problemas de sobreajuste.
  - Ajustar hiperparámetros como la tasa de aprendizaje.
  - Implementar técnicas como early stopping, que se explicará a continuación.

- Conjunto de Prueba (20%)

Sólo se utiliza una vez que el modelo está completamente entrenado, proporcionando una estimación imparcial de su capacidad de generalización.

En este caso, para garantizar que estos conjuntos sean representativos se realizó una selección aleatoria de los datos (con semilla 42 para reproducibilidad). Luego se mezclaron los índices antes de dividir por clases, para eliminar cualquier sesgo por el orden original. Finalmente, para la creación de los conjuntos se mantuvo la proporción entre las clases para que simularan la dinámica del conjunto completo eligiendo el porcentaje respectivo a cada conjunto por clase y uniendo estas selecciones.

6b.- Función de entrenamiento del modelo

En cada época de entrenamiento, el algoritmo implementado para crear el modelo:

1. Realiza predicciones en los tres conjuntos de datos (entrenamiento, validación y prueba).
2. Calcula y registra las métricas de pérdida y precisión para cada conjunto.
3. Detecta posibles signos de sobreajuste cuando la pérdida de validación aumenta mientras la de entrenamiento disminuye.
4. Aplica early stopping si es necesario.
5. Actualiza los pesos y el sesgo mediante gradiente descendiente usando sólo el conjunto de entrenamiento.

Los parámetros del modelo se actualizan a partir de los gradientes calculados, buscando minimizar el error y mejorar la precisión de las predicciones. El proceso se repite durante un número determinado de épocas, o hasta que se active el early stopping cuando el error de validación deja de mejorar. Durante todo el entrenamiento, se monitorean métricas como la pérdida y la precisión en los tres conjuntos para evaluar el desempeño, detectar sobreajuste y ajustar los hiperparámetros si es necesario. Así no sólo se mejorar el modelo, sino también se monitorea su capacidad de generalización a lo largo de todo el proceso.

6c.- Early Stopping como técnica de regularización

Esta técnica de regularización se implementa para monitorear la pérdida en el conjunto de validación durante el entrenamiento, buscando prevenir que el modelo “memorice” los datos de entrenamiento y pierda capacidad de generalizar.

La forma en que se aplica se basa en lo siguiente:

- Durante el entrenamiento, se monitorea la pérdida en el conjunto de validación.
- Cada vez que se calcula un modelo con mejor desempeño en validación, se guardan esos pesos y bias.
- Hay un registro de cuántas épocas consecutivas han pasado sin mejora significativa. Dado el tamaño del dataset, una mejora significativa se establece como una reducción de la pérdida de validación mayor a 1e-6.
- Si se alcanza un número predefinido de épocas sin mejora (parámetro de "paciencia", en este caso 20 épocas), se detiene el entrenamiento, aún si no se han completado las épocas previstas.
- Finalmente se recuperan los mejores parámetros encontrados durante todo el proceso.

7. EVALUACIÓN DEL MODELO

7a.- Matriz de confusión y métricas asociadas

Una matriz de confusión es una herramienta fundamental para evaluar el desempeño de un modelo de clasificación, como la regresión logística. [10]

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Figura 12. *Ejemplo de matriz de confusión [10]*

Esta matriz compara las predicciones del modelo con los valores reales, organizando los resultados en cuatro categorías: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). A partir de estos valores, se pueden calcular métricas clave, para este reporte usaré:

- Accuracy: Proporción de predicciones correctas. $(TP+TN)/total$
- Recall o sensibilidad: Capacidad del modelo para identificar correctamente los casos positivos. $TP/(TP + FP)$. En modelos donde se quieren detectar mayor cantidad de casos positivos como diagnósticos médicos o fraude se suele priorizar o buscar un balance con el Accuracy. [11]
- Especificidad: Capacidad del modelo para identificar correctamente casos negativos $TN/(TN + FP)$
- F1 Score: Media armónica entre precisión y recall, útil cuando se busca un balance entre ambas. $2TP/(2TP+FP+FN)$

Estas permiten entender. no sólo qué tan preciso es el modelo, sino también cómo se comporta frente a desequilibrios en los datos, como sucede en el dataset de este reporte.

7b.- Niveles de ajuste

El rendimiento de un modelo de machine learning está directamente relacionado con su capacidad para encontrar el equilibrio adecuado en su ajuste a los datos. Se evaluarán en esta ocasión los tres escenarios principales que pueden presentarse. Comprenderlos permite evaluar el desempeño del modelo para implementar técnicas para mejor rendimiento.

Antes de pasar a la tabla comparativa entre dichos escenarios hay que establecer que "valores/métricas óptimas" se refiere a valores de Pérdida (Loss) cercanos a 0 y Accuracy cercana a 1.

Escenario	Descripción del ajuste del modelo	Curvas de Pérdida (Loss) y Precisión (Accuracy)	Métricas de Matriz de confusión (Test)	Métricas reportadas en los conjuntos
Subajuste (Underfit-ting)	Demasiado simple que no captura patrones.	Loss alta estable; accuracy baja estable.	Muchos errores (FP y FN), Accuracy general y Recall bajos.	Métricas similares, pero bajas.
Ajuste ótpimo (Fitting)	Equilibrio entre patrones y puede generalizar.	Curvas convergente s a valores óptimos.	Valores altos y consistentes de todas las métricas.	Métricas similares y óptimas.
Sobre-ajuste (Overfit-ting)	Memoriza datos, incluyendo aquellos no significativos .	Divergencia entre curvas que se engrandece.	Accuracy menor que en el conjunto de train. Pero por sí sola, si no se tiene información de los demás conjuntos, no puede realizar un diagnóstico.	Métricas más óptimas en conjunto de entrenamient o que en validación y prueba.

Figura 13. *Indicadores de ajuste de modelo*

7c.- Evaluación de coeficientes

Significancia:

- P-value es un indicador de qué tan probable es que el efecto de una variable sea por azar, es por esto que se dice que un coeficiente es estadísticamente significativo si p-value es < 0.05.
- Un intervalo de confianza es el rango de valores en el que se estima que está el verdadero valor de un coeficiente, si no incluye 0 y es estrecho.

Magnitud y Signo:

- El valor absoluto del coeficiente indica que tan influyente es esa variable en la predicción.
- Si es positivo, la variable aumenta la probabilidad de diabetes.
- Si es negativo, la variable disminuye la probabilidad de diabetes.

7d.- Resultados Modelo de Regresión Logística

El modelo muestra características de un **ajuste óptimo** pues las curvas de Pérdida (Loss) y Accuracy convergen suavemente, se van estabilizando en conjunto, siendo casi idénticas y con métricas consistentes como se observa a continuación.

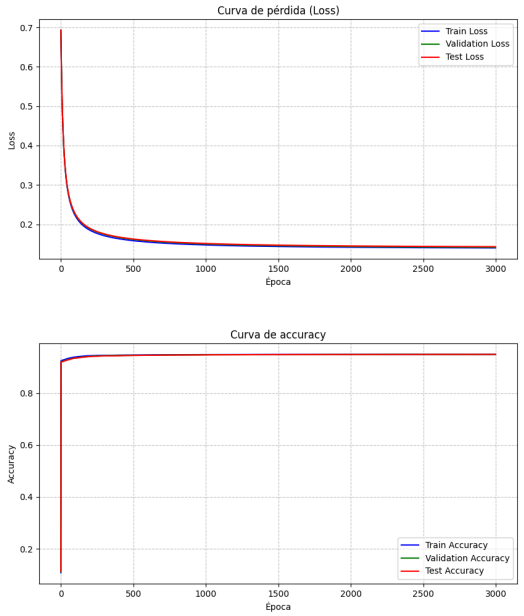

Figura 14. *Curvas de Pérdida (Loss) y Accuracy para modelo de Regresión Logística*

Conjunto	Loss	Accuracy
Train	0.14	94.88%
Validation	0.1414	94.86%
Test	0.1431	94.9%

Figura 15. *Métricas reportadas en conjuntos para modelo de Regresión Logística*

Para completar el análisis de desempeño, analizamos la matriz de confusión, que nos indica con el Recall que sólo se diagnostica correctamente el **63.14%** de todos los casos reales con diabetes, perdiendo cerca del 37% de los casos reales. Por otro lado, sí identifica correctamente el **99%** de los casos sin diabetes. Dados estos valores el F1 Score resulta en **73.9**, lo cuál sería bueno si los resultados no se inclinaran mayormente a los casos negativos. Aunque el modelo parece ser muy confiable, el recall en contextos médicos resultaría preocupante.

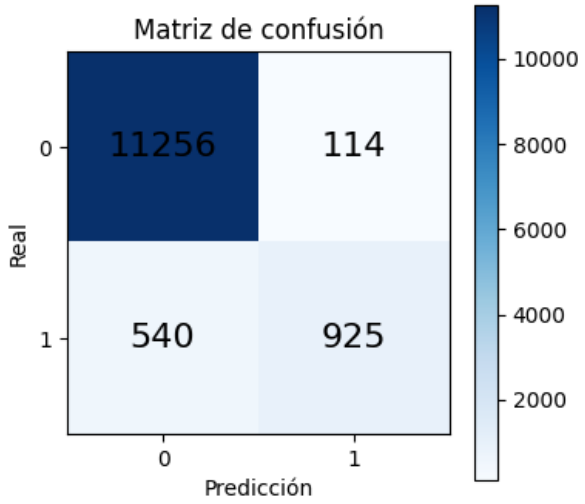

Figura 16. *Matriz de confusión para modelo de Regresión Logística*

Respecto a los coeficientes:

Casi todas las variables son estadísticamente significativas, excepto el sexo masculino **(p = 0.46)** que no muestra un efecto significativo sobre la diabetes, lo que se confirma por su intervalo de confianza que incluye el cero **[-0.08, 0.17]**.

HbA1c_level y *blood_glucose_level* son los predictores más fuerte de diabetes por su elevado coeficiente positivo **(coef = 2.21)** y **(coef = 1.29)**, respectivamente, además de los estrechos intervalos de confianza **[2.12, 2.31]** y **[1.22, 1.37]**.

Las categorías de historial de tabaquismo, principalmente no fumar, *smoking_history_never* **(coef = -0.67),** o dejar de fumar, *smoking_history_not current* **(coef = -0.59)**, está asociado con menor riesgo de diabetes comparados con la categoría de referencia, *smoking_history_current*.

La edad, hipertensión, y BMI incrementan moderadamente el riesgo.

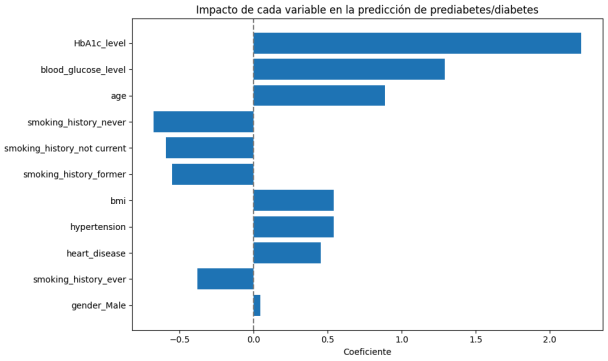

Figura 17. *Gráfica de coeficientes para modelo de Regresión Logística*

Feature	Coeficiente	p-value	¿Significante?	std error	Intervalo de confianza (95%)	
intercepto	-3.82	0.00	TRUE	0.04	-3.90	-3.73
age	0.89	0.00	TRUE	0.04	0.81	0.97
hypertension	0.54	0.00	TRUE	0.10	0.35	0.73
heart_disease	0.45	0.00	TRUE	0.14	0.18	0.73
bmi	0.54	0.00	TRUE	0.04	0.47	0.61
HbA1c_level	2.21	0.00	TRUE	0.05	2.12	2.31
blood_glucose_level	1.29	0.00	TRUE	0.04	1.22	1.37
gender_Male	0.05	0.46	FALSE	0.06	-0.08	0.17
smoking_history_ever	-0.38	0.02	TRUE	0.16	-0.69	-0.07
smoking_history_former	-0.55	0.00	TRUE	0.10	-0.74	-0.36
smoking_history_never	-0.67	0.00	TRUE	0.06	-0.79	-0.55
smoking_history_not current	-0.59	0.00	TRUE	0.13	-0.85	-0.34

Figura 18. *Tabla de evaluación de coeficientes para modelo de Regresión Logística*

8. IMPLEMENTACIÓN CON FRAMEWORK

8a.- Random Forest

El Random Forest es un algoritmo de aprendizaje automático supervisado que combina múltiples árboles de decisión para generar una predicción más precisa y robusta. [12]

El flujo que sigue su implementación es el siguiente:

1. Bootstrapping de datos: Cada árbol se entrena con una muestra aleatoria del conjunto de datos. Por lo que algunos registros se repiten dentro de una muestra y otros quedan fuera.
2. Feature bagging: Selección aleatoria de *features* a evaluar por nodo, buscará la mejor división entre las seleccionadas. Esto propicia la diversidad de árboles y evitando que todos se parezcan demasiado (baja la correlación), se explora la posibilidad de más y mejores patrones o conjuntos significativos de *features* para la predicción, favoreciendo la generalización.
3. Para clasificación, cada árbol da su predicción, se combinan y la más votada se asigna como resultado final.

Algunos contras podrían ser que presenta mayor complejidad que dificulta el entendimiento de cómo se llegó a la solución, a diferencia de un sólo árbol que es más interpretable. Y el uso intensivo de recursos que requiere más poder de cómputo, volviendo el

procesamiento es más lento si se usan grandes volúmenes de datos.[12]

Sin embargo, probablemente superará a la regresión logística porque las relaciones entre los *features* descritos y la diabetes suelen ser no lineales, porque cuenta con una mezcla de variables numéricas y categóricas con posibles patrones sutiles e interacciones complejas. Cabe recalcar que ya no se requiere estandarización previa, pues los valores extremos no afectan significativamente el rendimiento, ya que los árboles de decisión no se basan en magnitudes para tomar las decisiones.

8b.- Hiperparámetros

Los hiperparámetros en que se concentra este acercamiento a Random Forest son los siguientes [13]:

Hiperparámetro	Descripción	Valor default
n_estimators	Número de árboles en el bosque.	100, es un bosque de 100 árboles.
max_depth	Profundidad máxima de cada árbol (cuántos niveles puede tener).	None, no se limita la profundidad.
max_leaf_nodes	Número máximo de nodos hoja por árbol.	None, no hay límite de nodos hoja, permitiendo árboles con muchas hojas.
min_samples_split	Mínimo de muestras requeridas para dividir un nodo.	2, un nodo se divide con al menos 2 muestras, permite divisiones muy detalladas.
min_samples_leaf	Mínimo de muestras requeridas en una hoja.	1, Cada hoja puede contener al menos una muestra, lo que permite una división muy específica en los datos.
class_weight	Sirve para ajustar el peso o importancia de cada clase según su frecuencia.	None, el modelo tratará todas las clases como igualmente importantes.
n_jobs	Número de procesadores a usar.	'-1' significa usar todos los disponibles.
random_state	Semilla para la aleatoriedad, establecerla permite reproductividad en resultados.	42

Figura 19. *Tabla descriptiva de los hiperparámetros de un RandomForestClassifier, con la explicación de valores default*

8c.- Resultados del Modelo Random Forest con hiperparámetros default

Utilizando el framework Scikit-learn, se implementó el modelo RandomForestClassifier. Inicialmente, se evaluó su desempeño empleando el dstaset seleccionado y los hiperparámetros default, tal como se describió previamente. Y este fue el resultado.

El ajuste del modelo parece estar cayendo **levemente en sobreajuste** pues mientras que las métricas de Pérdida (Loss) y Accuracy en el conjunto de entrenamiento son casi perfectas, decaen en validación y prueba. Es altamente probable que el modelo esté memorizando datos y perdiera capacidad de generalizar nuevos.

Conjunto	Loss	Accuracy
Train	0.0257	99.97%
Validation	0.1610	96.24%
Test	0.1584	96.14%

Figura 20. *Métricas reportadas en conjuntos para modelo de Random Forest con hiperparámetros default*

Con la matriz de confusión podemos notar que si bien el Accuracy general incrementó respeto al modelo de regresión logística (con **96%**) , es porque esta nueva implementación identifica casi a la perfección los casos no diabéticos con una Especificidad cercana a **1**, pero sigue diagnosticando erradamente a 32% de personas con diabetes, dado que el Recall es de **0.68**. El F1 Sore muestra un balance de **0.8**, como se mencionó, ligado al alto desempeño identificando casos negativos.

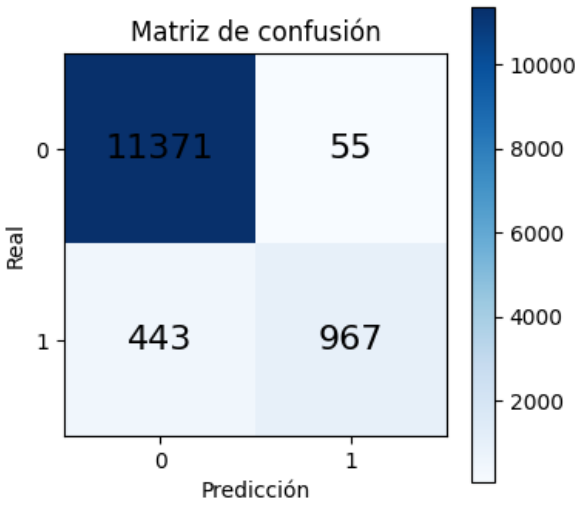

Figura 21. *Matriz de confusión para modelo de Random Forest con hiperparámetros default*

La curva ROC (Característica Operativa del Receptor) y su métrica AUC (Área Bajo la Curva) permiten evaluar

la capacidad del modelo para distinguir entre casos positivos y negativos, independientemente del umbral de decisión. Un valor alto de AUC indica que el modelo tiene buen poder discriminativo y generaliza correctamente [11], por lo que incluir esta métrica en el análisis es fundamental para validar el desempeño global y orientar mejoras en el modelo. Para este resultado, su valor cercano a 1 indica excelente capacidad discriminativa del modelo para distinguir entre clases.

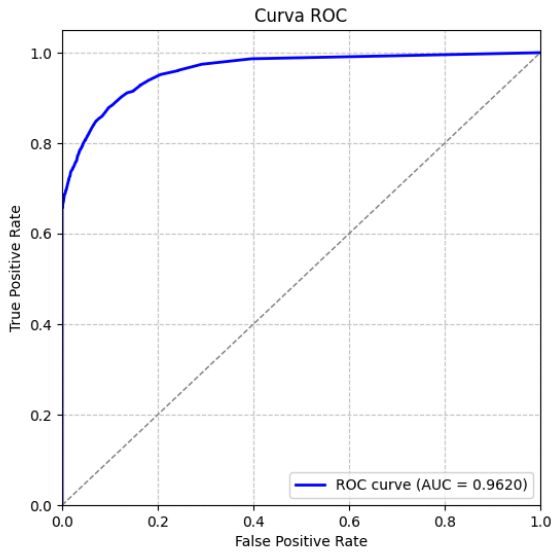

Figura 22. *Gráfica de curva ROC y AUC para modelo de Random Forest con hiperparámetros default*

Finalmente, de forma consistente con el modelo de Regresión Logística, los coeficientes a los que se les asigna mayor importancia son *HbA1c_level* y *blood_glucose_level*, lo que es consistente con el conocimiento clínico.

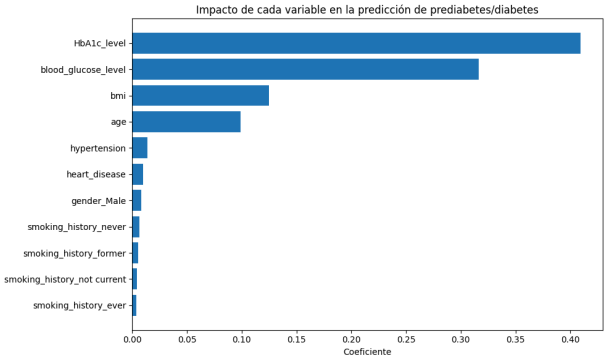

Figura 23. *Gráfica de coeficientes para modelo de Regresión Logística para modelo de Random Forest con hiperparámetros default*

9. MEJORA DE MODELO CON FRAMEWORK

9a.- Ajuste de hiperparámetros

Como técnica de regularización, de acuerdo con los ajustes y buscando mejorar el Recall (Accuracy de la clase minoritaria) se ajustarán los hiperparámetros.

Hiperparámetro	Mayor valor	Menor valor
n_estimators	Mejora la estabilidad y precisión del modelo, pero aumenta el tiempo en train.	Entrenamiento más rápido, puede resultar un modelo menos robusto.
max_depth	Árboles complejos, identifican más patrones pero hay riesgo de sobreajuste.	Árboles más simples y generales, puede evitar el sobreajuste pero también llevar a un subajuste si es muy bajo.
max_leaf_nodes	Permite árboles más complejos, que pueden sobreajustar si el valor es muy alto.	Limita la complejidad de cada árbol, forzando árboles más simples (reduce sobreajuste).
min_samples_split	Obliga a que las hojas tengan más datos antes de dividirse. Árboles pequeños y simples que pueden evitar sobreajuste pero sacrificar patrones.	Divisiones muy específicas, los árboles crecen hasta complejidades grandes, riesgo de sobreajuste.
min_samples_leaf	Hojas con más datos, árboles más generales. Menos propenso a sobreajuste.	Pocos datos por hoja lo que puede llevar a sobreajuste porque tal vez sólo tenga ejemplos de la clase mayoritaria.

Figura 24. *Tabla explicativa sobre impacto del cambio en hiperparámetros para RandomForestClassifier*

Configurar el **class_weight** con el indicador **"balanced"** tiene el siguiente efecto:

- En un dataset desbalanceado aumenta el peso de la clase menor para que los árboles no ignoren los pocos ejemplos de esta.
- Penalizar más los errores de la clase mayoritaria.
- Aplicar si se busca mejorar F1 de la clase menor.

En conclusión, si se quiere:

A. Detectar mejor la clase minoritaria, modificar class_weight/min_samples_split/min_samples_leaf.
B. Reduir o evitar sobreajuste, modificar max_dept y max_leaf_nodes.

C. Tener más árboles para entrenar el modelo, es conveniente elevar n_estimators.

Si bien el análisis del modelo anterior detectó un sobreajuste, lo primero que se buscó equilibrar fueron los pesos de la clase minoritaria y esto se mantuvo constante en ajustes futuros, puesto que se quiere encontrar un equilibrio entre la predicción de clases.

Además de esto, los patrones detectados en los resultados de las modificaciones de hiperparámetros visibles en la Figura 25 (Revisar sección **12. RECURSOS** al final del reporte), fueron:

- Modelos sin restricción de profundidad: muestran alta accuracy (>95%), pero recall bajo (≈0.69). Eso significa que dejan escapar muchos pacientes con diabetes, lo que sería clínicamente problemático.
- Modelos con profundidad limitada (max_depth=10–15, class_weight=balanced): bajan la accuracy a ≈90–93%, pero aumentan el recall a ≈0.89–0.9, detectando la gran mayoría de diabéticos aunque con más falsos positivos.
- Número de árboles (n_estimators): apenas cambia el rendimiento después de cierto punto. El verdadero ajuste se da modificando la profundidad y el balance de clases.
- Exclusión de variables: al quitar categorías de *smoking_history* o *gender* el desempeño no varía, lo que confirma que esas variables tienen poco peso predictivo.
- El aumento en la curva ROC-AUC muestra que el modelo distingue mejor entre personas con y sin diabetes, aunque la Accuracy no cambie mucho. Mejora su capacidad discriminativa global pues ordena más claramente a los pacientes por riesgo. Esto sugiere que se puede ajustar el umbral para priorizar Recall (detectar más diabéticos) sin perder tanta precisión.

9b.- Evaluación del modelo con framework e hiperparámetros ajustados

Si bien una vez introducido el balanceo de clases, los ajustes mostraron cambios menores, se optó por aquél que mejoraba tanto Recall como Especificidad.

Las métricas de Pérdida (Loss) y Accuracy son muy similares en entrenamiento, validación y prueba, lo que indica que el modelo generaliza bien, un **ajuste óptimo**.

Conjunto	Loss	Accuracy
Train	0.2032	90.79%
Validation	0.2044	90.56%
Test	0.2047	90.59%

Figura 26. *Métricas reportadas en conjuntos para modelo de Random Forest con hiperparámetros ajustados*

Analizando las métricas con la matriz de confusión, la Especificidad y Recall de **0.9** y **0.88** respectivamente indican que se identifican correcta y casi proporcionalmente los casos de diabetes y sin diabetes. F1 Score de **0.67** es consistente con el balance esperado.

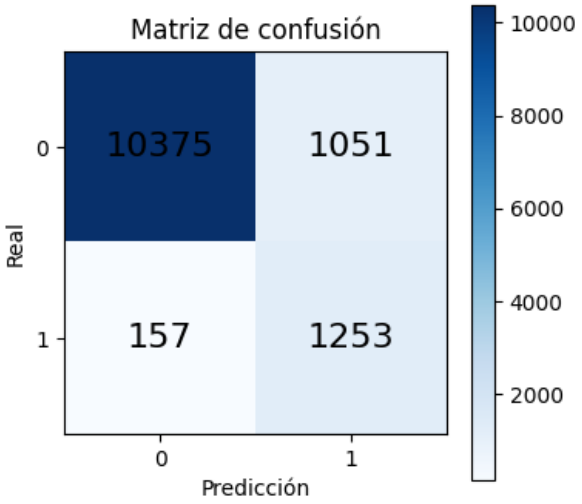

Figura 27. *Matriz de confusión para modelo de Random Forest con hiperparámetros ajustados*

La curva ROC y AUC de 0.97 indica una excelente capacidad para distinguir entre clases, independientemente del umbral de decisión que se implemente.

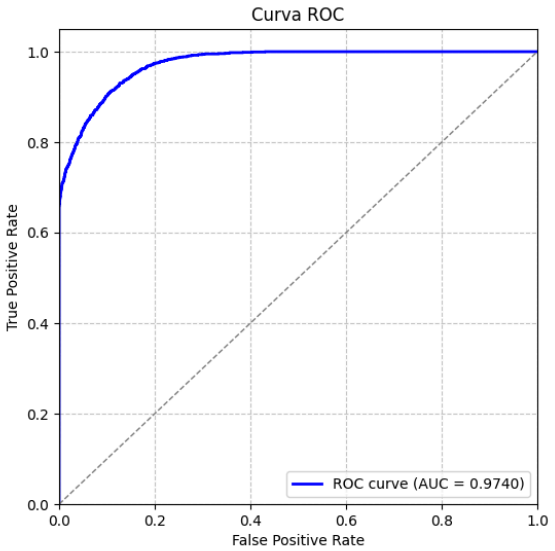

Figura 28. *Gráfica de curva ROC y AUC para modelo de Random Forest con hiperparámetros ajustados*

Consistente a lo evaluado en modelos previos, dada su correlación reportada y relevancia clínica, *HbA1c_level* y *blood_glucose_level* son los predictores más importantes de diabetes; las variables eliminadas (como otras categorías de *smoking_history* y *género*) tienen impacto mínimo en el modelo, pero ayudan a rescatar algunos TN.

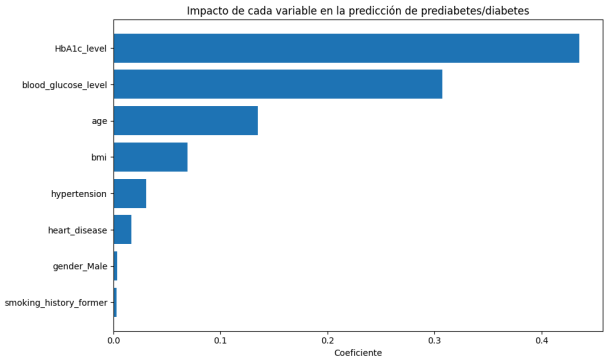

Figura 26. *Gráfica de coeficientes para modelo de Random Forest con hiperparámetros ajustados*

10. CONSIDERACIONES ADICIONALES

En el contexto de este reporte, es fundamental reconocer que el dataset analizado presenta un marcado desbalance de clases (≈90% sin diabetes frente a ≈10% con diabetes). Bajo estas condiciones, la métrica de Accuracy puede ser engañosa, ya que un modelo trivial que siempre prediga "no diabetes" alcanzaría una exactitud cercana al 90% sin dar valor clínico. Por esta razón, métricas como Recall y F1 Score son más relevantes, al reflejar la capacidad real del modelo para identificar correctamente a los pacientes diabéticos.

Los resultados muestran que, al limitar la profundidad de los árboles e incluir class_weight=balanced, se obtiene un Recall cercano al 90%, lo que permite identificar la gran mayoría de los casos positivos, aunque a costa de un mayor número de falsos positivos. En el ámbito médico, esta estrategia es preferible, puesto que el costo clínico de un falso negativo (no detectar a un paciente con diabetes) es mucho mayor que el de un falso positivo, el cual puede confirmarse con pruebas adicionales.[14]

En diferentes reportes consultados, se ha destacado que Random Forest logra desempeños sólidos en esta tarea, pero también se han propuesto mejoras como el uso de técnicas de rebalanceo (SMOTE o ADASYN), la optimización de umbrales de decisión sobre la curva Precision–Recall y la implementación de modelos de ensamble más avanzados como XGBoost. [14]

En conclusión, un sistema de predicción para diabetes idealmente se conforma de varios pasos, agregando a la detección automatizada con priorización del Recall como métrica principal, pruebas confirmatorias para los casos positivos.

11. REFERENCIAS

[1] *Diabetes prediction dataset.* (2023, 8 abril). https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data

[2] Sikaris, K. (2009). *The correlation of hemoglobin A1C to blood glucose.* Journal of Diabetes Science and Technology, 3(3), 429–438. https://doi.org/10.1177/193229680900300305

[3] Bhandari, A. (2025, 23 abril). *What is Feature Scaling and Why is it Important?* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

[4] GeeksforGeeks. (2025, 11 julio). *One hot encoding in machine learning.* GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/ml-one-hot-encoding/

[5] Sandaruwan, S. (2025, 21 mayo). *Dummy Variables and the Dummy Variable Trap.* Medium. https://medium.com/%40sriyantha1234567890/dummy-variables-and-the-dummy-variable-trap-27d950ede687

[6] Algorithm Alchemist. (2024, 2 septiembre). Understanding Logistic Regression: A Cornerstone of Classification in Data Science. Medium. https://blog.gopenai.com/understanding-logistic-regression-a-cornerstone-of-classification-in-data-science-d3f77d42f0d8

[7] Dyakonov, A. (2021, 15 febrero). *Log Loss Function*. Dasha. Dasha | Log Loss Function Explained by Experts

[8] Raschka, S. (2025, 16 septiembre). *What are gradient descent and stochastic gradient descent?* Sebastian Raschka, PhD. https://sebastianraschka.com/faq/docs/gradient-optimization.html

[9] Donges, N. (2024, 1 agosto). *Gradient Descent in Machine Learning: A Basic Introduction.* Built In. https://builtin.com/data-science/gradient-descent

[10] Markham, K. (2024, 4 junio). *Simple guide to confusion matrix terminology.* Data School. https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

[11] *Glosario sobre aprendizaje automático*. (s.f.). https://developers.google.com/machine-learning/glossary?authuser=3&hl=es-419#AUC

[12] *¿Qué es el bosque aleatorio?* (s.f) IBM. https://www.ibm.com/mx-es/think/topics/random-forest

[13] *RandomForestClassifier.* (s.f.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[14] Chang, V., Ganatra, M. A., Hall, K., Golightly, L., & Xu, Q. A. (2022). *An assessment of machine learning models and algorithms for early prediction and diagnosis of diabetes using health indicators.* Healthcare Analytics, 2, 100118. https://doi.org/10.1016/j.health.2022.100118

12. RECURSOS

Variables excluidas	class_weight	max_leaf_nodes	max_depth	n_estimators	min_samples_split	min_samples_leaf	ACC TRAIN	ACC VAL	ACC TEST	F1 (1)	Accuracy (1) / RECALL	Accuracy (0) / ESPECIFICIDAD	AUC ROC	Observación	Matriz de confusión
* smoking_history_former: 0.003087 * gender_Male: 0.003048 * smoking_history_not current: 0.000754 * smoking_history_ever: 0.000475 * smoking_history_never 0.001439	None	None	None	100	2	1	99.82	96.07	95.89	0.79	69.72	99.12	0.9612	Overfitting	[11325 101] [427 983]
	balanced	None	None	100	2	1	99.77	96.03	95.64	0.78	69.01	98.91	0.9605	Overfitting	[11301 125] [437 973]
	balanced	None	10	100	2	1	90.52	90.48	90.29	0.67	89.57	90.37	0.9737	Fitting, varios FP	[10326 1100] [147 1263]
	balanced	None	15	100	2	1	95.68	93.54	93.39	0.73	81.63	94.84	0.971	Leve overfitting	[10836 590] [259 1151]
	balanced	None	15	200	2	1	95.81	93.55	93.48	0.73	81.21	94.99	0.9712	Leve overfitting, menos TP	[10854 572] [265 1145]
	balanced	None	10	300	2	1	90.51	90.49	90.32	0.67	89.22	90.46	0.9737	Fitting, más TP, pero menos TN	[10336 1090] [152 1258]
	balanced	None	10	400	2	1	90.47	90.48	90.28	0.67	89.08	90.43	0.9736	Fitting	[10332 1094] [154 1256]
	balanced	**None**	**10**	**500**	**2**	**1**	**90.5**	**90.52**	**90.34**	**0.67**	**89.08**	**90.5**	**0.9737**	**Fitting**	**[10340 1086] [154 1256]**
	balanced	None	10	800	2	1	90.46	90.49	90.32	0.67	89.29	90.45	0.9737	Fitting	[10340 1091] [151 1259]
	balanced	None	10	1000	2	1	90.49	90.49	90.33	0.67	89.15	90.48	0.9737	Fitting	[10338 1088] [153 1257]
	balanced	None	10	2000	2	1	90.52	90.5	90.32	0.67	89.15	90.47	0.9737	Fitting	[10338 1089] [153 1257]
	balanced	None	10	500	2	5	90.42	90.43	90.38	0.67	89.22	90.52	0.9738	Fitting	[10343 1083] [152 1258]
	balanced	**None**	**10**	**500**	**2**	**6**	**90.45**	**90.51**	**90.4**	**0.67**	**89.22**	**90.55**	**0.9739**	**Fitting**	**[10346 1080] [152 1258]**
Ninguna	balanced	None	10	500	2	6	91.15	90.88	91.05	0.68	88.58	91.35	0.974	Fitting	[10438 988] [161 1249]
* smoking_history_not current: 0.000754 * smoking_history_ever: 0.000475	balanced	None	10	500	2	6	90.51	90.28	90.35	0.67	89.57	90.44	0.9739	Fitting	[10334 1092] [147 1263]
* smoking_history_not current: 0.000754 * smoking_history_ever: 0.000475 * smoking_history_never 0.001439	balanced	None	10	500	2	6	90.79	90.56	90.59	0.67	88.87	90.8	0.974	Fitting	[10375 1051] [157 1253]
	balanced	**None**	**10**	**500**	**8**	**6**	**90.79**	**90.56**	**90.59**	**0.67**	**88.87**	**90.8**	**0.974**	**Fitting**	**[10375 1051] [157 1253]**

Figura 25. *Tabla de métricas reportadas en diferentes modelos de Random Forest con hiperparámetros ajustados*